454 Sequencing System Software Manual
Version 2.9

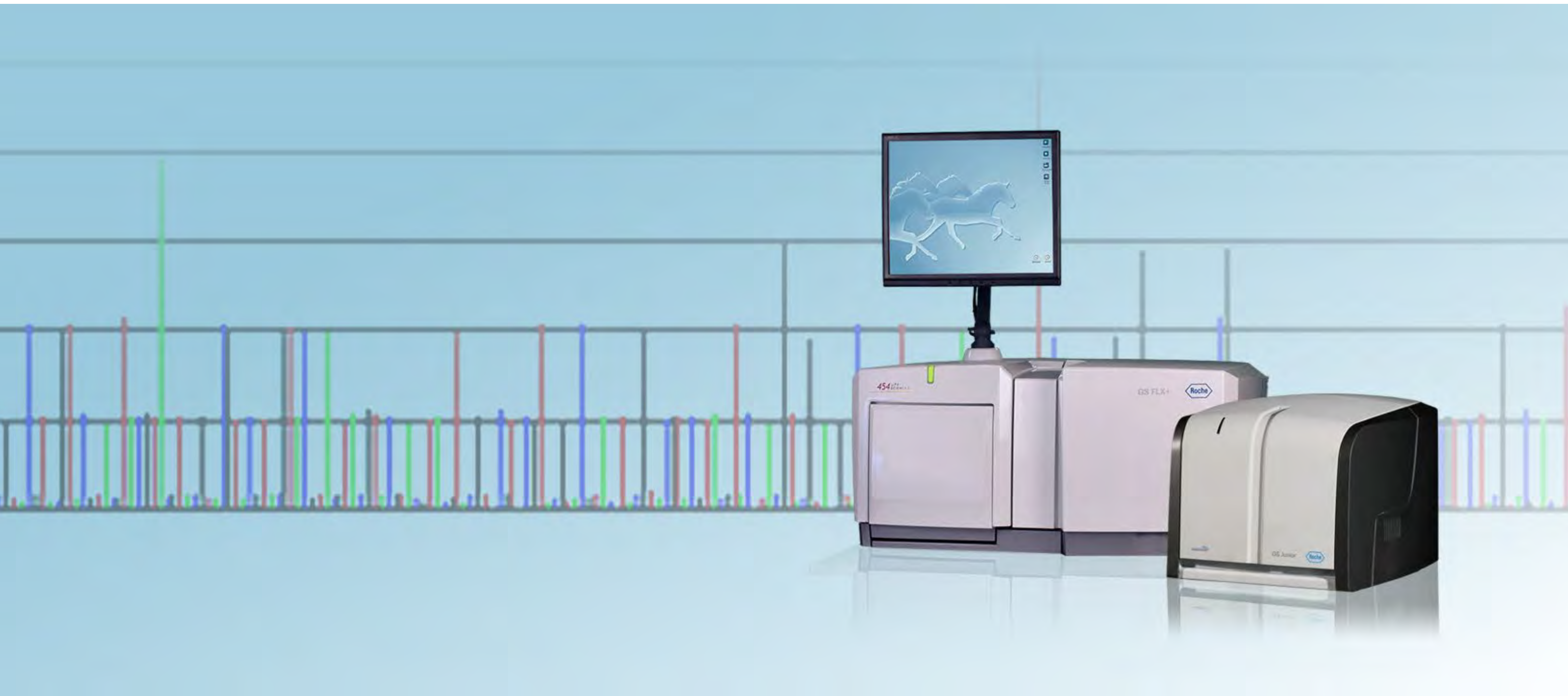
Part C:

GS De Novo Assembler, GS Reference Mapper, SFF Tools

June 2013

	Instrument	Kit
✓	GS Junior	Junior
✓	GS FLX+	XL+
✓	GS FLX+	XLR70
✓	GS FLX	XLR70

For life science research only. Not for use in diagnostic procedures.



GS De Novo Assembler, GS Reference Mapper, SFF Tools

1 GS *DE NOVO* ASSEMBLER

1.1 Overview of the GS *De Novo* Assembler

The GS *De Novo* Assembler application constructs *de novo* assemblies of the reads from one or more sequencing runs, using the "read flowgrams" (SFF files) as input. The GS *De Novo* Assembler software is an interactive application used to create assembly projects, add or remove reads from the project, specify project parameters, run the assembly algorithms on the project data, and view the output produced by the assembly computations. The application can be accessed *via* a Graphical User Interface (GUI) or from a command line interface (CLI).

Input data can come from one or several regions of one or several runs of interest. Additional read data can be imported from one or more external file formats, including FASTA and FASTQ. For example, long Sanger sequencing reads can be used to improve the construction of contigs and scaffolds. Assembly generates a consensus sequence of the sample DNA library, output as one or more contiguous sequences (contigs) in FASTA, ACE or consed files.

The GS *De Novo* Assembler allows the inclusion of one or more Paired End runs into the analysis, enabling the ordering and orientation of the assembled contigs from shotgun sequencing runs into scaffolds. This requires the preparation of a separate Paired End library from the same DNA material that was used to prepare the library for shotgun sequencing. In certain cases, the presence of reads from a separate shotgun library in the project is not strictly required, as Paired End reads can themselves be assembled with each other.

During the assembly process, the software:

- Identifies pairwise overlaps between reads
- Constructs multiple alignments of overlapping reads and divides or introduces breaks into the multiple alignments in regions where consistent differences are found between different sets of reads. (This step results in a preliminary set of "contigs" that represent the assembled reads.)
- Attempts to resolve branching structures between contigs
- Generates consensus basecalls of the contigs by using quality and flow signal information for each nucleotide flow included in the contigs' multiple alignments
- Outputs the contig consensus sequences and corresponding quality scores, along with an ACE file of the multiple alignments and assembly metrics files.

When Paired End data is available, the assembler performs these extra steps:

- Organizes the contigs into scaffolds using Paired End information to order and orient the contigs and to approximate the distance between contigs
- Outputs scaffolded consensus sequences and corresponding quality scores, ordered by size, along with an AGP file of the scaffolds and specific metrics tables.

Read overlaps and multiple alignments are made in "nucleotide" space while the consensus basecalling and quality value determination for contigs are performed in "flowspace". Work in flowspace allows the quality-weighted averaging of processed flow signals (a continuous variable) at each nucleotide flow of the sequencing run(s) and allows the use of information from the "negative flows", *i.e.* flows where no nucleotide incorporation is detected. The use of flowspace in determining the properties of the consensus sequence results in an improved accuracy for the final basecalls.

The assembler produces contigs from the multiple alignments of overlapping read sequences. The GUI (Graphical User Interface) provides tools that allow the contig consensus sequences, the multiple alignments of the reads that form the contig, and the flowgrams of these reads to be viewed interactively. The CLI (Command Line Interface) produces files containing the output produced by the assembly. The GUI can display the contents of these files.

- The GS *De Novo* Assembler application can be run from the attendant PC or a datarig when using the GS Junior Instrument.
- The GS *De Novo* Assembler application is not available on the GS FLX+ Instrument and must be run on a datarig for users of this instrument.
- Because of the inherent complexity of large genome assembly, we recommend contacting your Roche representative if you encounter any problems assembling any project where the expected genome size exceeds 500 Mbp.

The GS *De Novo* Assembler allows users to create, modify, and run assemblies in the form of projects. Both the GUI and command line interface (CLI) provide this functionality. Projects may be setup to assemble all reads at once. Alternatively, incremental operation allows additional reads to be added to an existing assembly. Results appear as output files using either the GUI or the CLI. The GUI provides a graphical interface to view many of the results from the assembly whether the project was assembled using the GUI or the CLI.

The GS *De Novo* Assembler application uses a folder on the file system to hold the assembly project information (whether the assembly of the reads, *i.e.* the computation, is carried out in project-based mode through the GUI application or through the newAssembly and related commands) and to hold the output files generated during and after the assembly computation.

The operation of the GUI is described in several of the subsequent sections. A description of the CLI is then presented followed by a discussion of transcriptome assembly. Finally, output files produced by the GS *De Novo* Assembler are described.

1.2 GS *De Novo* Assembler GUI

Assemblies can be performed or viewed using the Graphical User Interface application, gsAssembler, described in the following sections. The application includes graphical interfaces to:

- create new assembly projects
- open existing assembly projects
- Add/remove reads to/from assembly projects
- Modify assembly input, computation, and output parameters
- carry out an assembly computation
- view the results of a completed assembly
- view the progress and logging information of an assembly computation

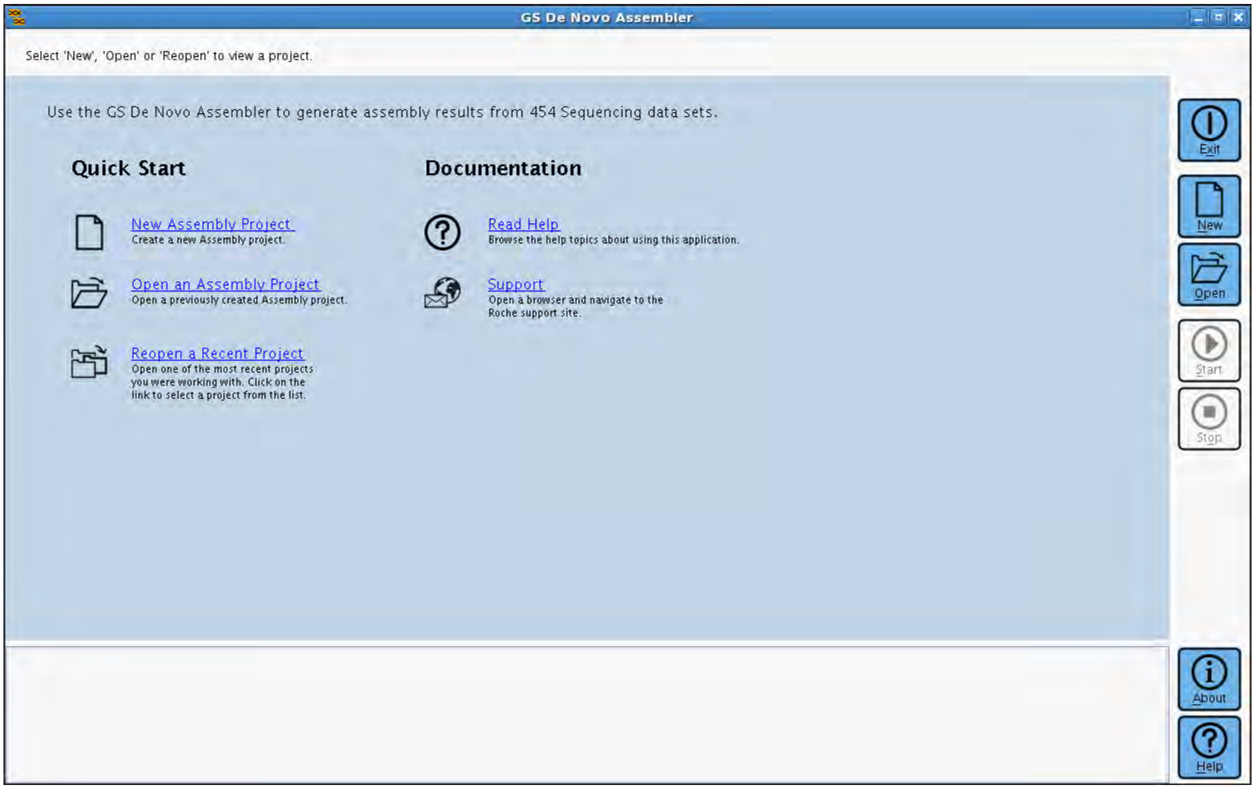
1.3 Launching the GS *De Novo* Assembler GUI Application

The GS *De Novo* Assembler GUI application is launched by double clicking on its desktop icon.

Alternatively, the following command can be used to launch the GS *De Novo* Assembler GUI application from a Linux terminal window on a GS Junior attendant PC or on a datarig, where the data analysis software package is installed:

```
gsAssembler
```

Once the GUI is launched (Figure 1), a user can open an existing project or create a new project.



GS De Novo Assembler

Select 'New', 'Open' or 'Reopen' to view a project.

Use the GS De Novo Assembler to generate assembly results from 454 Sequencing data sets.

Quick Start

New Assembly Project
Create a new Assembly project.

Open an Assembly Project
Open a previously created Assembly project.

Reopen a Recent Project
Open one of the most recent projects you were working with. Click on the link to select a project from the list.

Documentation

Read Help
Browse the help topics about using this application.

Support
Open a browser and navigate to the Roche support site.

Figure 1: Initial Interface of the gsAssembler.

1.3.1 The Main Buttons, Status Area and Progress Box

Seven main buttons are always visible on the toolbar, along the right-hand side of the GS *De Novo* Assembler's main window:

- The **Exit** button closes the GS *De Novo* Assembler application
- The **New** button allows one to create a new assembly project
- The **Open** button allows one to open an existing assembly project
- The **Start** button begins the assembly (computation) of an open assembly project
- The **Stop** button halts the execution of an ongoing assembly computation
- The **About** button shows the GS *De Novo* Assembler splash screen
- The **Help** button opens the GS *De Novo* Assembler section of the software manual.

The top of the main GS *De Novo* Assembler window contains a status area that displays a command hint, while the bottom panel is a progress box which remains blank until an assembly project has been opened.

The progress box at the bottom of the window can be resized by dragging its top frame separator. If the field is not visible, it is probably simply collapsed; you can view it by dragging up the edge of the collapsed frame.

1.3.2 The Quick Start and Documentation Text Links

There are 3 text links under the **Quick Start** column:

- The **New Assembly Project** text link creates a new assembly project. It performs the same action as the **New** button on the right side of the window.
- The **Open an Assembly Project** text link opens an existing project. It performs the same action as the **Open** button on the right side of the window.
- The **Reopen Recent Project** text link displays a dialog from which you can open a project on which you worked recently.

There are 2 text links under the **Documentation** column:

- The **Read Help** text link opens an electronic version of the software manual, *i.e.* it performs the same action as the **Help** button on the right side of the window.
- The **Support** text link opens the default internet browser and navigates to the Roche support site.

1.4 Opening a Project

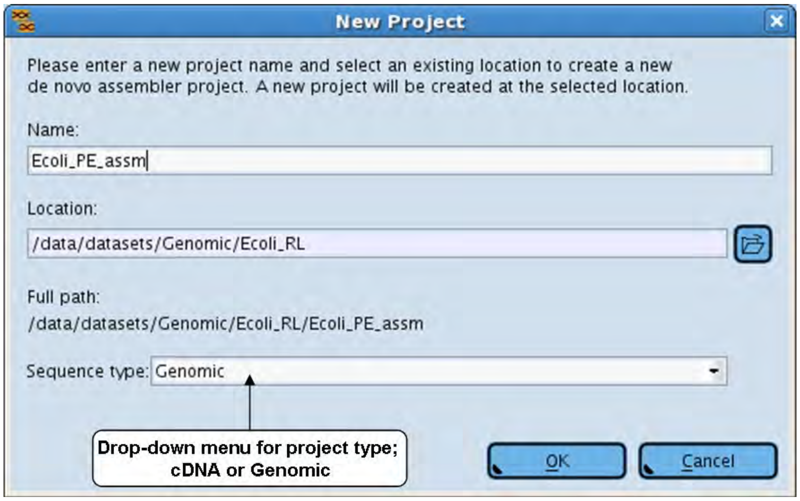
1.4.1 Creating a New Project

To create a new assembly project, either click on the **New Project** button in the right toolbar, or click on the "New Assembly Project" text button in the **Quick Start** column. This displays a dialog (Figure 2) in which you can specify the name and directory location for a new project.

Type a name for the new project in the **Name** text field. To specify the project location, either type the full path in the **Location** text field, or click on the **Open Project** button to the right of the text field and use the "Select Project Location" window (not shown) to navigate to the directory where you want to create the new project. The **Full Path** field changes as you update the **Name** and **Location** fields. When you are satisfied with the location and name for the new project, click on the **OK** button. You can save a project by clicking on the Yes button to the **Save** prompt.

You may save the project to your hard drive either by using the Exit button to exit the GS *De Novo* Assembler application (you will be prompted to save before the application actually exits) or by adding read data and running the project by clicking the Start button (*i.e.*, compute the assembly), in which case the project will automatically be saved prior to computation, as described in section 1.8.1, below.

The New Project dialog contains a drop-down menu (Figure 2) to specify the **Sequence type** indicating the type of DNA library sequenced, cDNA or Genomic. This choice of Sequence type determines many of the parameters the user will be allowed to modify in the project. Once a project is created the type cannot be changed.



New Project

Please enter a new project name and select an existing location to create a new de novo assembler project. A new project will be created at the selected location.

Name: Ecoli_PE_assm

Location: /data/datasets/Genomic/Ecoli_RL

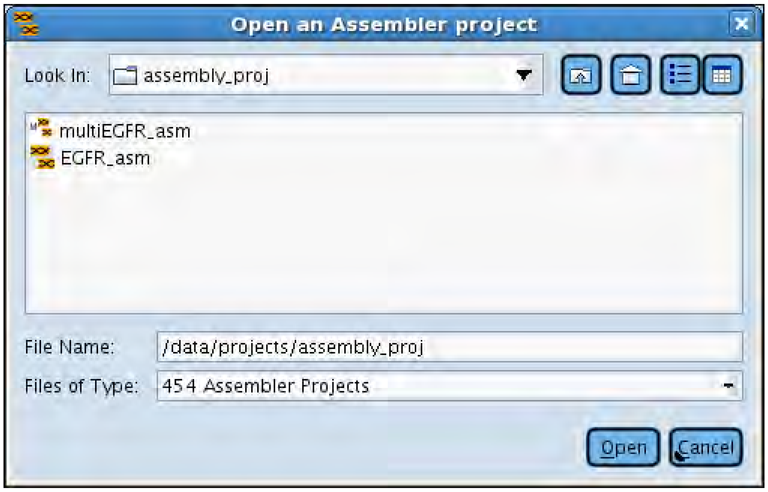
Full path: /data/datasets/Genomic/Ecoli_RL/Ecoli_PE_assm

Sequence type: Genomic

Figure 2: New Project dialog used to specify Sequence type.

1.4.2 Open an Existing Project

To open an existing assembly project, either click on the **Open Project** button in the right toolbar, or click on the "Open an Assembly Project" text button in the **Quick Start** column. This displays a dialog (Figure 3) in which you can select the name and specify the directory location of the project to be opened. By default, only 454 Assembly Projects will be displayed. You may choose to display all files by selecting "All Files" from the **"Files of Type"** dropdown menu.



Open an Assembler project

Look In: assembly_proj

multiEGFR_asm
EGFR_asm

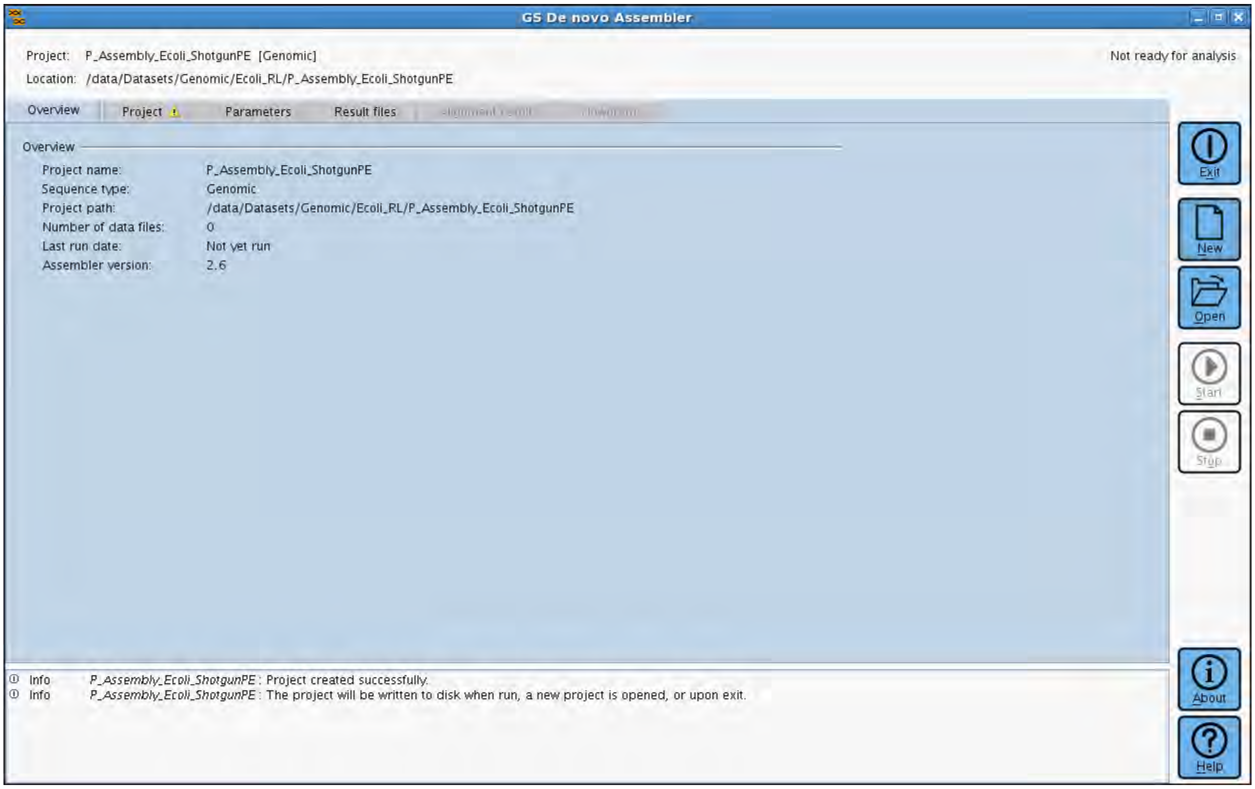
File Name: /data/projects/assembly_proj

Files of Type: 454 Assembler Projects

Figure 3: Open Assembly Project dialog. Note the distinctive icons for a multiplex project (the first icon, see Section 1.13) *vs.* an ordinary assembly project (the second icon).

1.5 View Project Summary with the Overview Tab

When a project opens, the Overview Tab is displayed (Figure 4). Other tabs can be selected with the mouse. Some tabs may remain inactive (grayed-out), until the type of information they require is available for the project. The Overview Tab's ***Project Summary Information*** area indicates the Sequence type along with other information that applies to the overall project. The data displayed on this tab is updated as new information becomes available, which occurs when data files are added or removed from the project and when a project computation completes.



Project: P_Assembly_Ecoli_ShotgunPE [Genomic]
Location: /data/Datasets/Genomic/Ecoli_RL/P_Assembly_Ecoli_ShotgunPE

Not ready for analysis

Overview

Project name:	P_Assembly_Ecoli_ShotgunPE
Sequence type:	Genomic
Project path:	/data/Datasets/Genomic/Ecoli_RL/P_Assembly_Ecoli_ShotgunPE
Number of data files:	0
Last run date:	Not yet run
Assembler version:	2.6

```
Info    P_Assembly_Ecoli_ShotgunPE : Project created successfully.
Info    P_Assembly_Ecoli_ShotgunPE : The project will be written to disk when run, a new project is opened, or upon exit.
```

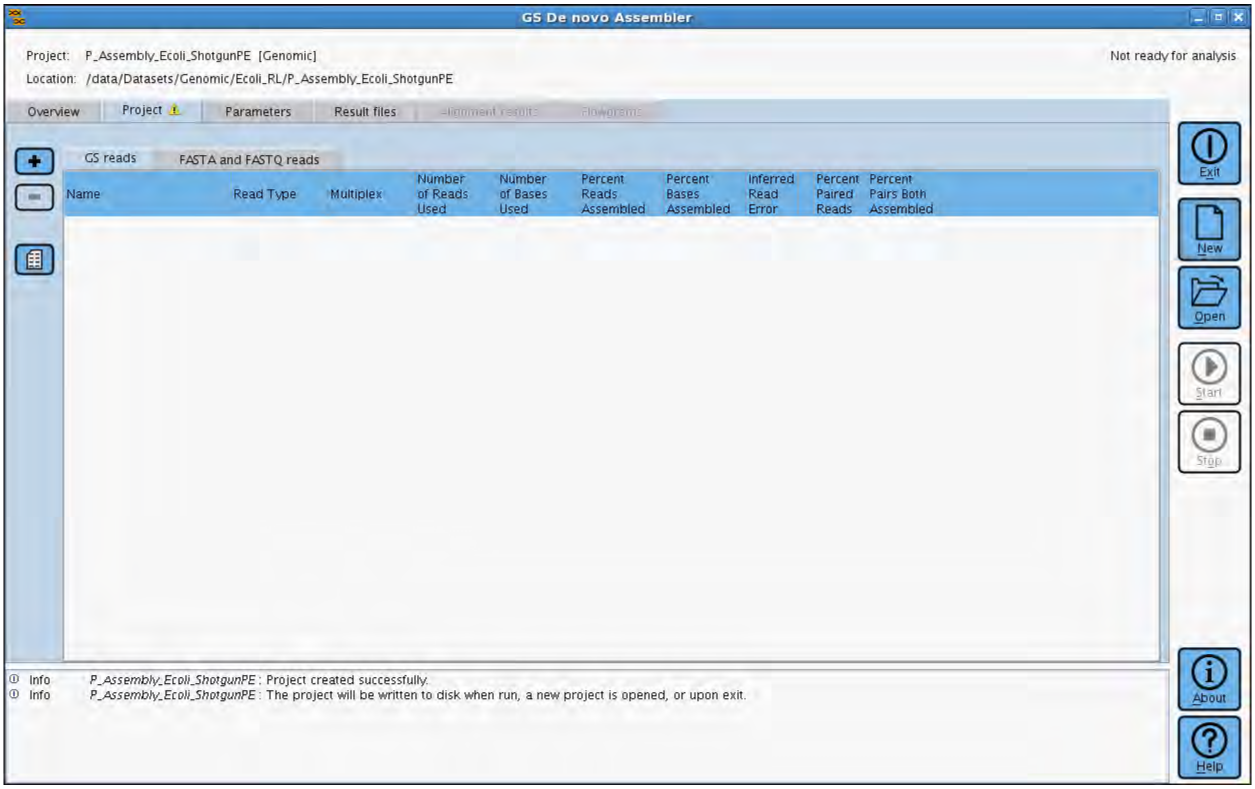
Figure 4: gsAssembler Overview Tab.

When a project is first created, the ***Computation status message*** in the upper right hand corner of the application window indicates "Not ready for Analysis" and the Start button is disabled because at least one Read Data file must be added to the project before an Assembly computation can be performed. Other errors in input parameters can also cause this message to appear. Once all such errors have been corrected and at least one Read Data file is added to the project and all options have valid values, the message will change to "Ready for analysis" and the Start button will become active.

1.6 Add/Remove Read Data with the Project Tab

Read Data files are added to a project on the Project tab of the main application window. There are two sub-tabs for adding read data: one for adding GS reads (GS Reads sub-tab) and one for adding non-GS reads, such as Sanger reads, in FASTA/FASTQ format (FASTA and FASTQ Reads sub-tab).

> The Start button in the toolbar, which is used to start an assembly, remains disabled until at least one Read Data file (either GS read or FASTA/FASTQ read) has been added to the project.



Project: P_Assembly_Ecoli_ShotgunPE [Genomic]

Location: /data/Datasets/Genomic/Ecoli_RL/P_Assembly_Ecoli_ShotgunPE

Not ready for analysis

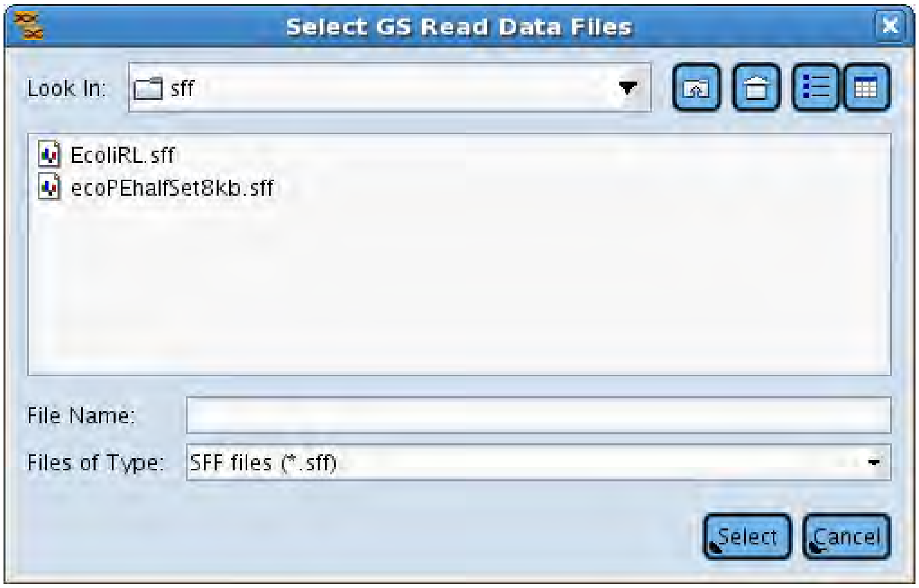
Name	Read Type	Multiplex	Number of Reads Used	Number of Bases Used	Percent Reads Assembled	Percent Bases Assembled	Inferred Read Error	Percent Paired Reads	Percent Pairs Both Assembled

```
Info    P_Assembly_Ecoli_ShotgunPE : Project created successfully.
Info    P_Assembly_Ecoli_ShotgunPE : The project will be written to disk when run, a new project is opened, or upon exit.
```

Figure 5: Project Tab of the gsAssembler (GS Reads sub-tab).

1.6.1 GS Reads, FASTA and FASTQ Reads Sub-Tabs

To add GS read data sets to a project, click on the GS reads sub-tab, then click the **Add** button ⊞ to open the "Select GS Read Data Files" window (Figure 6). This dialog window allows you to navigate and see the available read files. Read files already in the project are visible but grayed out. You can select multiple files in this window using the ctrl or shift keys while clicking with the mouse.



Look In: sff

EcoliRL.sff
ecoPEhalfSet8kb.sff

File Name:

Files of Type: SFF files (*.sff)

Figure 6: Select GS Read Data dialog.

Once the desired file(s) are selected, the "Set GS Read Data Attributes" dialog window opens. From this window (Figure 7) the GS Read file(s) can be explicitly specified as Paired End or non-Paired End, or the user can invoke auto-detection using the **Read Type Specification** dropdown. When the read type is known, it is advisable to specify it directly rather than use auto-detect as the auto-detect feature, on rare occasion, may fail to detect a Paired End file.

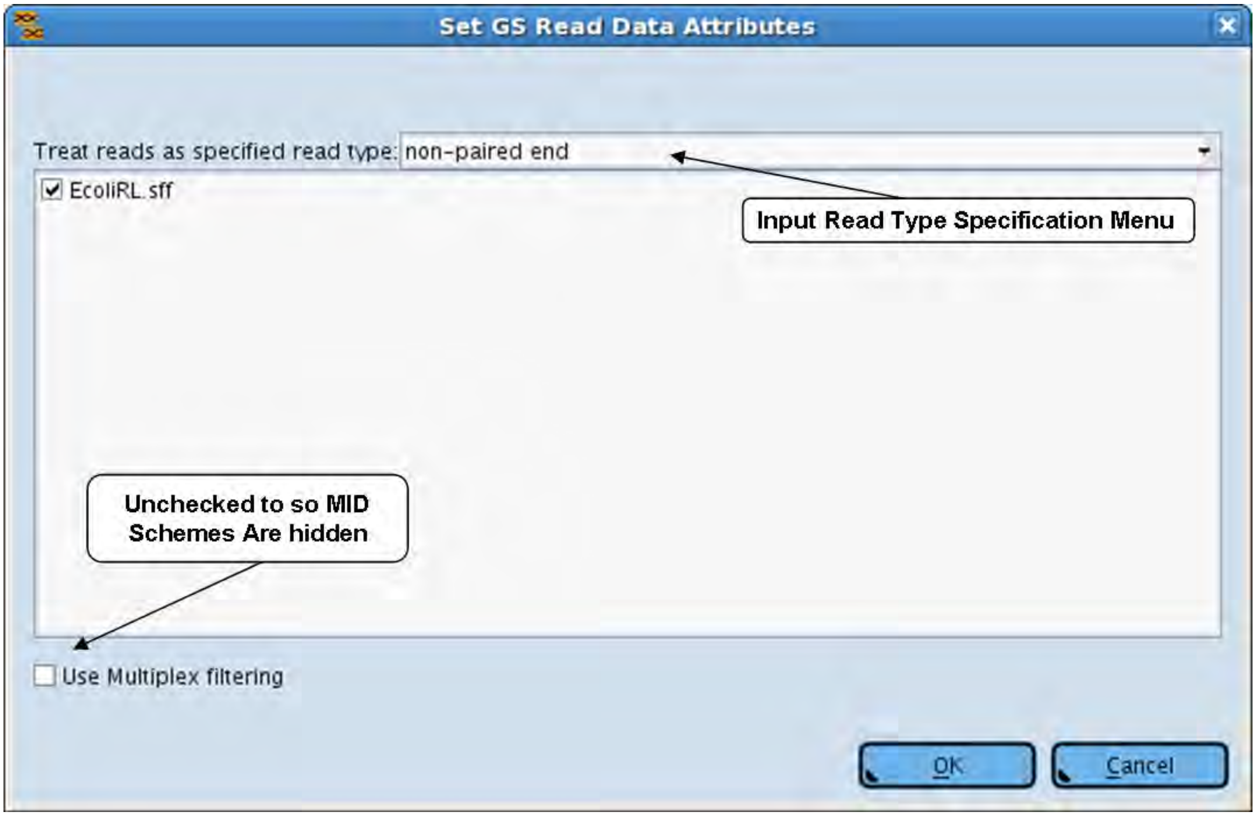

Set GS Read Data Attributes
Treat reads as specified read type: non-paired end
☑ EcoliRL.sff
Input Read Type Specification Menu
Unchecked to so MID Schemes Are hidden
□ Use Multiplex filtering
OK
Cancel

Figure 7: Set GS Read Data Attributes dialog.

When planning to run an incremental assembly, it is best to first add the shotgun reads, and then add reads from Paired End libraries with increasing insert spans. This is because longer contigs can be assembled using the longer shotgun reads. These longer contigs then have a better chance of having both ends of paired end reads aligned within them. This in turn allows more robust library span estimates to be made (see section 4.6).

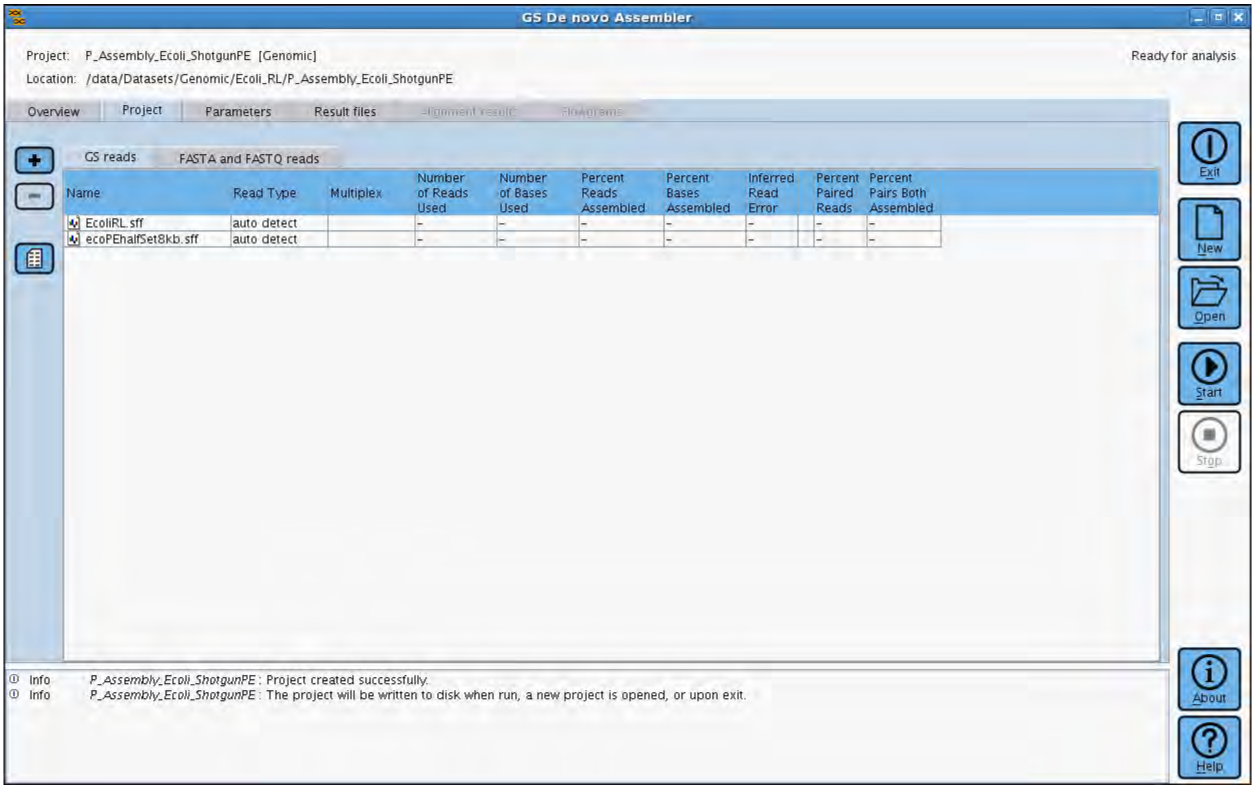
You can add any number of Read Data files to a project using the Add button, navigating to a folder, then selecting the files needed from the folder. You can even add multiple files that share the same name into a project, as long as they have different path locations in the file system (*e.g.* "/dir1/path1/reads.sff" and "/dir2/otherpath2/reads.sff"). In such a case, both files will be added to the Read Data list and displayed using the same name. To see the path to a file listed in the Read Data area of the main window (and the file's last modification date), hover the mouse over the filename of interest and a tooltip containing the file's path will be displayed. Files that have failed validation will have a red X left of the file name as an indication of failure. Pause the mouse cursor over the red X to bring up a tooltip explaining the problem encountered.

The FASTA and FASTQ Reads sub-tab is functionally similar to the GS Reads sub-tab for all project types. The only difference is that when a directory containing FASTA/FASTQ files is selected, the software examines the files in that directory to determine which files are FASTA/FASTQ files. This can take some time if there are many files in the directory, so a progress bar is displayed to show the progress of the search. See the Overview Manual, Section 2.2.2 for a discussion of the FASTQ format.

Order of addition of Read data may affect assembly results: In general, the reads should be added to an incremental assembly in the following order to achieve the best contigging and scaffolding results:

- Shotgun
- 3 kb Paired End
- 8 kb Paired End
- 20 kb Paired End

Except for the Name column and the Multiplex column (which contains comma-delimited lists of the MIDs associated with the file; see Section 1.6.3, below), all columns with run statistics are initially filled with dashes. These data are updated in the table each time the project runs to completion. For a project that has already been through an assembly computation, summary statistics relating to the usage of the reads in the assembly process is also listed, as shown in Figure 18. On the left hand side of the reads table is a data export button that can be used to write the reads table data in csv, tab-delimited or plain text formats.



Project: P_Assembly_Ecoli_ShotgunPE [Genomic]
Location: /data/Datasets/Genomic/Ecoli_RL/P_Assembly_Ecoli_ShotgunPE
Ready for analysis

Name	Read Type	Multiplex	Number of Reads Used	Number of Bases Used	Percent Reads Assembled	Percent Bases Assembled	Inferred Read Error	Percent Paired Reads	Percent Pairs Both Assembled
EcoliRL.sff	auto detect		–	–	–	–	–	–	–
ecoPEhalfSet8kb.sff	auto detect		–	–	–	–	–	–	–

```
Info    P_Assembly_Ecoli_ShotgunPE : Project created successfully.
Info    P_Assembly_Ecoli_ShotgunPE : The project will be written to disk when run, a new project is opened, or upon exit.
```

Figure 8: Project Tab of the gsAssembler (GS Reads sub-tab), after adding several read data files.

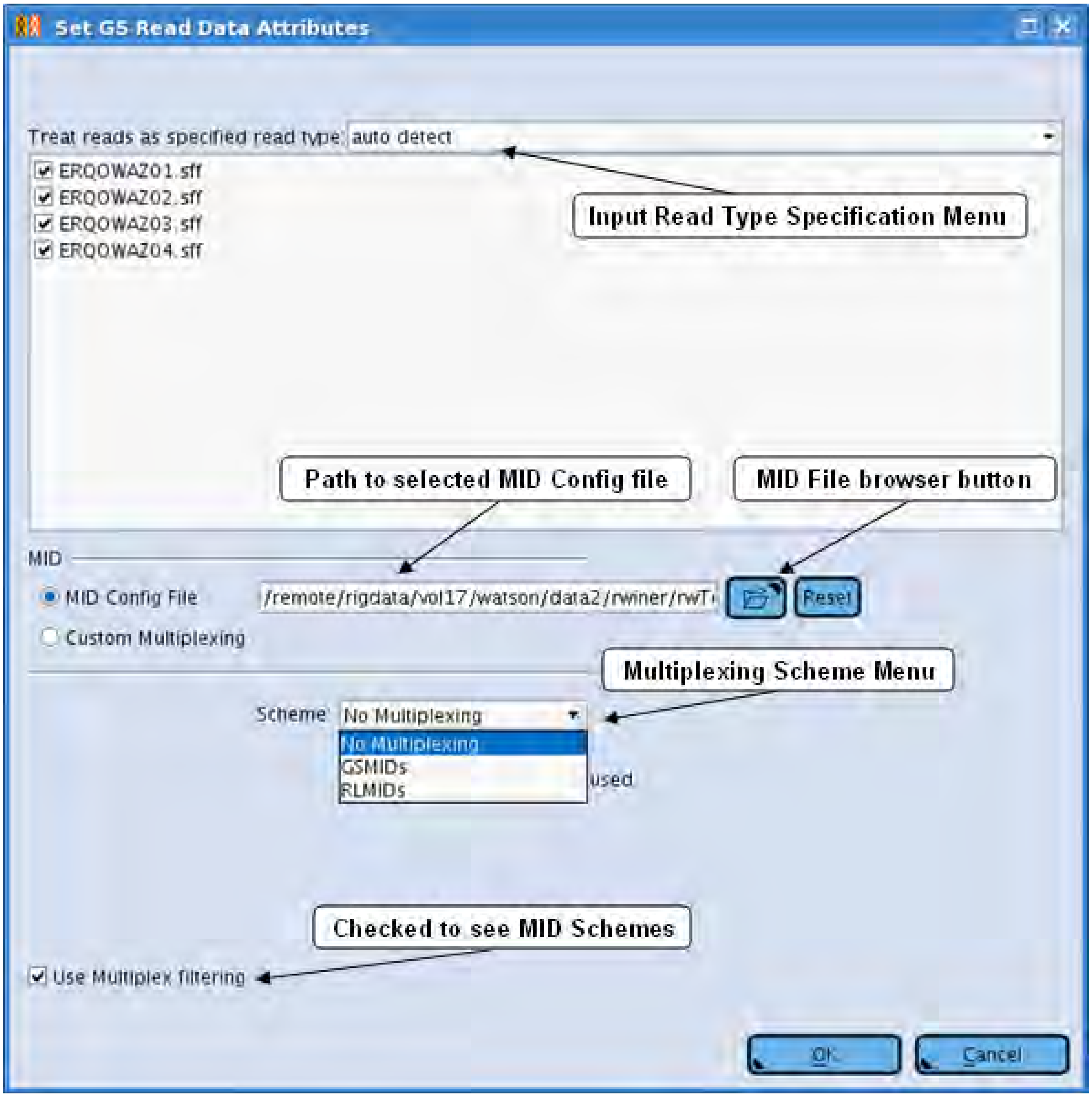
1.6.2 Removing Read Data from the Project Sub-Tabs

To delete read data from a project, click on the appropriate sub-tab, then click on a read file(s), then click the **Remove** button ⊟.

The **Remove** button removes the selected Read Data files from the list, but does not immediately remove the file(s) from the project's sff sub-directory or from any computed assembly results. Actual removal of the files from the project occurs only when the assembly is re-computed. When the files are removed, the reads they contain are removed from the alignments in which they occur.

1.6.3 Specifying Multiplex Identifiers (MIDs)

When the “Use Multiplex Filtering” checkbox is selected, special controls for MID filtering are displayed in the bottom half of the ”Set GS Read Data Attributes” window (Figure 9). Multiplex Identifiers (MIDs) allow you to design an experiment whereby multiple libraries are prepared using distinct MID tags and sequenced together, on the same PicoTiterPlate or PTP device. On the GS FLX+ system, multiple libraries can be sequenced together in the same region of a PTP device. (Use of MIDs can greatly improve the workflow and cost effectiveness of your experiment.)



Set GS Read Data Attributes

Treat reads as specified read type: auto detect

☑ ERQOWAZ01.sff
☑ ERQOWAZ02.sff
☑ ERQOWAZ03.sff
☑ ERQOWAZ04.sff

MID
● MID Config File /remote/rigdata/vol17/watson/data2/rwiner/rwT
○ Custom Multiplexing

Scheme: No Multiplexing

☑ Use Multiplex filtering

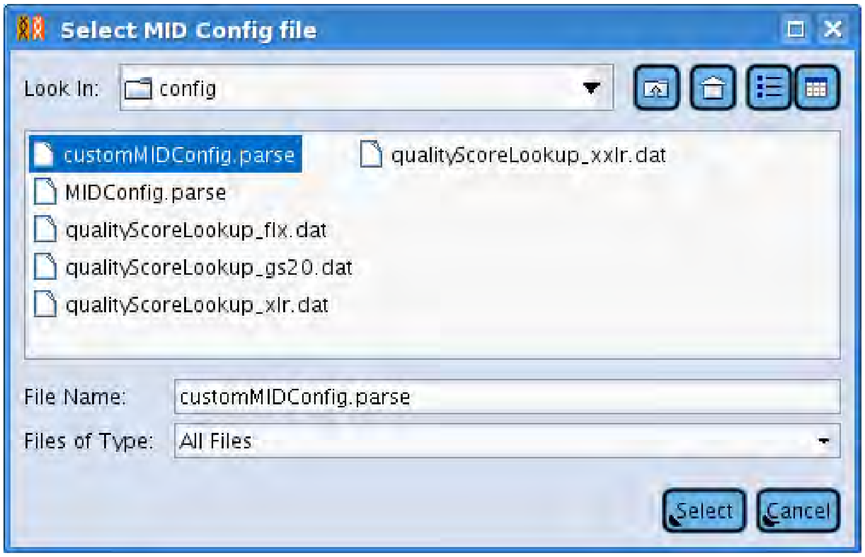
Input Read Type Specification Menu
Path to selected MID Config file
MID File browser button
Reset
Multiplexing Scheme Menu
No Multiplexing
GSMIDs
RLMIDs
used
Checked to see MID Schemes
OK
Cancel

Figure 9: Set GS Read Data Attributes dialog with the MID Scheme options expanded.

The MID controls on this window allow you to filter the reads in the selected Read Data files for inclusion in the Assembly project. Note, this filtering operation does not produce a different Assembly for each specified MID.

Rather, the reads of the SFF files are scanned for the presence of MIDs, and the reads containing the selected MIDs are made eligible for assembly in the project. The selected MID filtering criteria are associated with the sff files when the OK button is clicked (Figure 9). Any reads from these file(s) without the specified MIDs will be ignored. To produce independent Assemblies for different MIDs (or sets of MIDs), you must create independent Assembly projects for each of the desired MID subsets of the data.

MID filtering information is contained in 'MID schemes'. Available MID schemes may, in turn, be found in MID configuration files. The default MID configuration file supplied with the GS *De Novo* Assembler software is MIDConfig.parse (see Glossary). The MID schemes provided in this file are 'No Multiplexing', 'GSMIDs' for use of the original Genome Sequencer MID sets, and 'RLMIDs' for the Rapid Library Preparation Kit MIDs. An alternate configuration file can be specified by pressing the "browse for MID configuration file" button to the right of the "MID Config File" textbox (Figure 9). The "Select MID Config file" dialog (Figure 10) will then appear and can be used to navigate to the desired location. Choose the file to be used as the configuration file for the sff file(s) being imported and click the Select button. The MID schemes specified in the file selected will then populate the Scheme dropdown. Press the "Reset" button to set MIDConfig.parse (the default MID configuration file) as the configuration file to be used in the file import. 'Custom Multiplexing' can be selected if custom MIDs were used.


Select MID Config file
customMIDConfig.parse
qualityScoreLookup_xxlr.dat
MIDConfig.parse
qualityScoreLookup_flx.dat
qualityScoreLookup_gs20.dat
qualityScoreLookup_xlr.dat
Select
Cancel

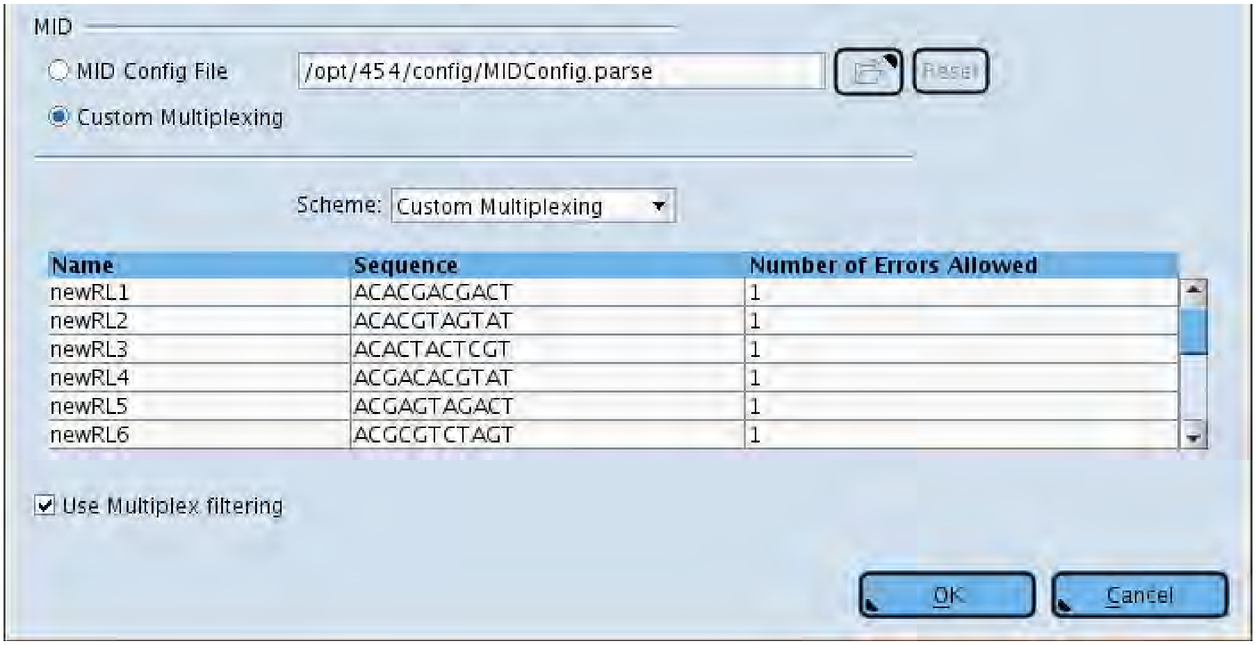
Look In: config

File Name: customMIDConfig.parse

Files of Type: All Files

Figure 10: Select MID Config file browser.

By default, no MID/Multiplexing scheme is associated with Read Data files. To use MIDs, first, specify the MID Config File, then select the desired MID Scheme (Figure 9) using the **Scheme** drop down menu. When an MID scheme is selected, a table of the MIDs present in that scheme is displayed, as shown in Figure 11. Use the checkboxes on the left to select or deselect the MIDs to be included in the assembly. Shift-clicking selects or deselects all MIDs in the scheme at once. The names, sequences and error limits of the MIDs selected will be used to filter the reads (from the Read Data sets selected in the top part of the window). The result is that only the reads that contain the selected MIDs, within the number of errors specified in the scheme, will be included in the project.



MID

○ MID Config File /opt/454/config/MIDConfig.parse

● Custom Multiplexing

Scheme: Custom Multiplexing

Name	Sequence	Number of Errors Allowed
newRL1	ACACGACGACT	1
newRL2	ACACGTAGTAT	1
newRL3	ACACTACTCGT	1
newRL4	ACGACACGTAT	1
newRL5	ACGAGTAGACT	1
newRL6	ACGCGTCTAGT	1

☑ Use Multiplex filtering

Figure 11: Set GS Read Data Attributes dialog with the RLMIDs Scheme active.

For more information about MIDs, see Section 4.5.

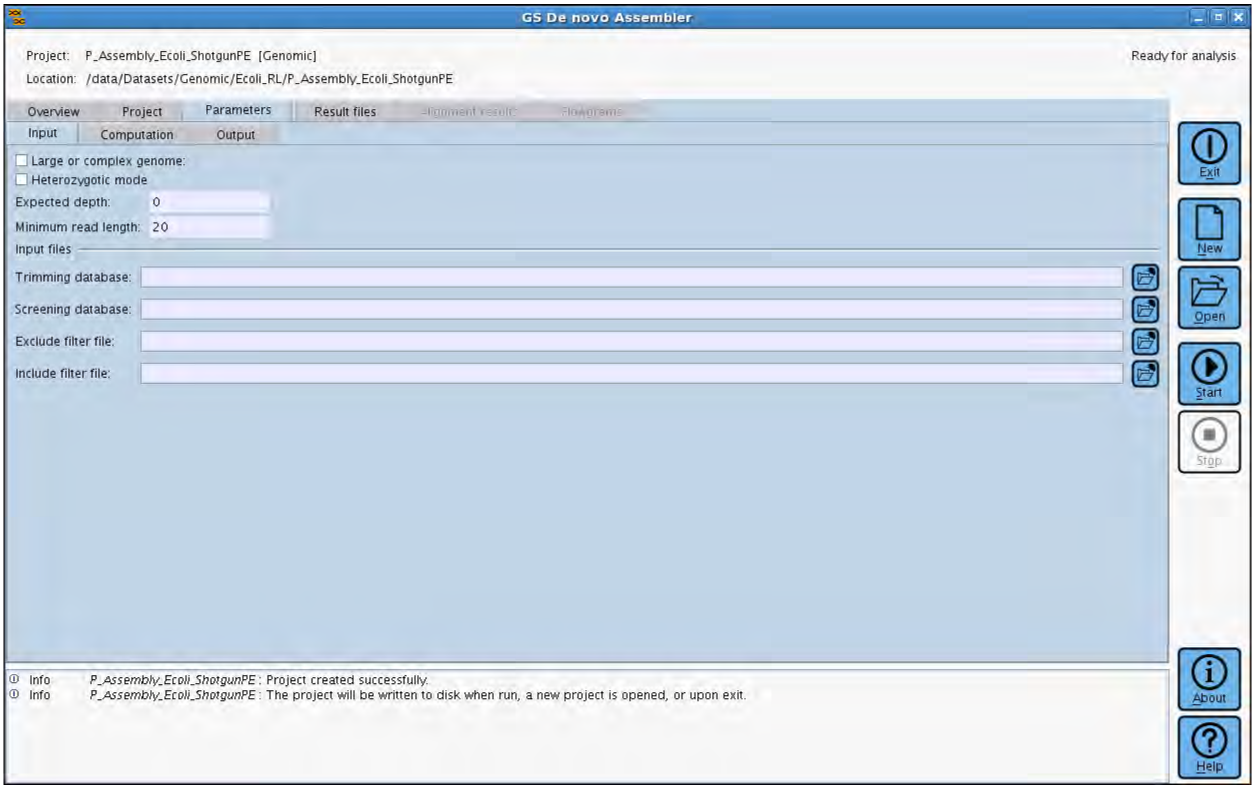
Once the MID scheme has been specified, click the **OK** button to add the selected data files (with MID filtration information) to the list of GS read data files (see Figure 9).

In certain circumstances, you may want to apply a different MID selection to different Read Data sets, within the same project. For example, you may have multiple libraries of the same organism, made with different MIDs (or with/without MIDs), that you want to assemble together. In this case, select the Read Data set(s) that need to be filtered with any given MID (or combination of MIDs), and click **OK** to add the MID-filtered data file(s) to the project. Back on the GS Reads sub-tab of the GS *De Novo* Assembler window (with the GS Reads sub-tab active) (Figure 5), click the **Add** button (⊞) again, and select other Read Data sets, to be filtered with different MIDs.

Incorrect MIDs may be assigned: In some cases, the MID(s) you select may be within the allowed number of errors of another MID, resulting in incorrect MID assignment. If this occurs, you should use the sfffile command (with the -s option) to split the reads with different MIDs into different files. The sfffile command, described in section 3.1, considers all MIDs in the MIDConfig.parse file, and will assign the MID(s) more accurately. The default MIDs and RLMIDs present in the MIDConfig.parse differ from each other by more than the allowed number of errors so this should occur very infrequently.

1.7 Customize Project with the Parameters Tab

The Parameters Tab is organized as three sub-tabs: Input, Computation and Output (Figure 12). The input parameters and output data options that can be specified are different for the Genomic and cDNA sequence types. Each sequence type will have its own set of parameters under the project sub-tabs.



Project: P_Assembly_Ecoli_ShotgunPE [Genomic]

Location: /data/Datasets/Genomic/Ecoli_RL/P_Assembly_Ecoli_ShotgunPE

Ready for analysis

□ Large or complex genome:

□ Heterozygotic mode

Expected depth: 0

Minimum read length: 20

Input files

Trimming database:

Screening database:

Exclude filter file:

Include filter file:

```
Info   P_Assembly_Ecoli_ShotgunPE : Project created successfully.
Info   P_Assembly_Ecoli_ShotgunPE : The project will be written to disk when run, a new project is opened, or upon exit.
```

Figure 12: gsAssembler Parameters tab, Input sub-tab for Genomic projects.

For a new project, the assembly parameters that take numerical values are initially set with their default values, as shown. If an invalid value is entered into any of these fields, a small red "X" appears in the lower left corner of the field and activates the "Not ready for analysis" warning in the upper right hand corner of the screen. Pausing the mouse over the red "X" reveals a tooltip indicating the allowed values for the parameter (see Section 1.12).

While default parameter values and settings are appropriate for most genomic and transcriptomic assemblies, you may find some options useful for specific projects.

1.7.1 Genomic Project Input Sub-Tab

The Input sub-tab for genomic projects is shown on Figure 12 (for cDNA projects, see Section 1.7.4). It allows you to adjust the following settings:

- The **Large or complex genome** (**-large**) option should be used when large or complex data sets (*i.e.* eukaryotic genomes) are being assembled. This will invoke algorithms especially suited for such data sets, allowing successful and speedy assembly. This option:
  - Skips assembly of high-copy count (> 500) repeats.
  - Sets the "seed count" to 2 during the assembly computation (see Section 1.7.2).
  - Skips the last level of detangling.
  - May result in 10% more contigs than might be obtained without this option.
  - Should not be used in cDNA assembly projects.
- The **Heterozygotic mode** (**-het**) option is used to specify that the project's read data is from a diploid or non-inbred organism. This prompts the assembler to adjust the algorithms it uses to reflect an increase in the expected variability in sequence identity. When two contigs can be joined by more than one path, ambiguity exists as to which path should be represented in the scaffold. The –het option instructs the software to select one path for inclusion in the scaffold and join the contigs through the chosen path. If –het is not used, the contigs won't be joined and if they are in a scaffold, a gap will appear between them in the scaffold.
- The **Expected depth** (**-e**) option allows you to specify the expected depth of coverage in the assembly. For high-depth assemblies (where the expected coverage of a position in the genome could reach hundreds or thousands of reads), this option can be used to filter out random-chance events that would be considered significant against a lower depth background.
  - Default: 0
  - Allowed values: 0 or greater, where a value of 0 tells the assembler to not use expected depth information in its computation.
- **The Minimum read length** (**-minlen**) option can be used to set the minimum read length used in assembly computations (default 20 bp). See Section 1.16.1.10 for details on the handling of reads under 50 bp.

The **input files** section of the tab allows you to specify the locations of several files that may be used in the assembly.

- An optional **Trimming database** (**-v | -vt**) is used to trim the ends of input reads (for cloning vectors, primers, adapters or other end sequences). Specify the path to a FASTA file of sequences to be used for this trimming (see Section 4.7).

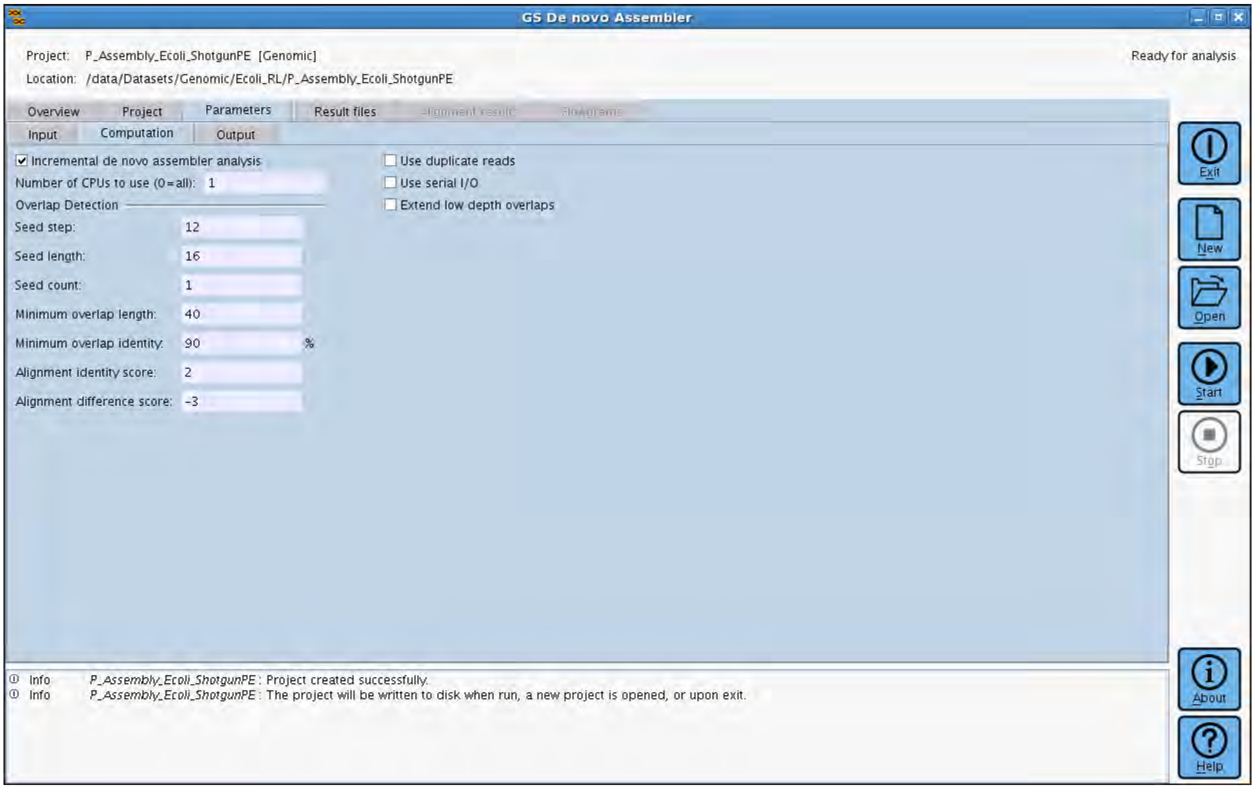
Please note that if you see warnings in your assembly, with regard to primer sequences at the 3' end, you may need to trim these sequences from your reads, by using the trimming database. You should trim the sequences if they appear to represent artifacts introduced during library preparation. It may take 3 or more iterations to completely get rid of all such sequences. You will need to make note of any additional warning that come up during the assembly, and add those to the FASTA file, for the next iteration.

- To use the **Screening database** (**-vs**) option, set the path to a FASTA file of sequences to be used to screen the input reads for contaminants. A read that almost completely aligns against a sequence in the screening database is removed so that it is not used in the computation; if at least 15 bases of a read do not align to the screening sequence, no action is taken (see Section 4.7).
- The **Include filter file** (**-fi**) and **exclude filter file** (**-fe**) options can be used to specify the path to a file of sequences to specifically include or exclude in the computation. The file format and functionality are the same as the –e and –i options of sfffile (see Section 3.1).

Long file paths will be displayed in full when you hover the mouse cursor over the text area.

1.7.2 Project Computation Sub-Tab

The Computation sub-tab is shown on Figure 13. It allows you to adjust the settings described below.



Project: P_Assembly_Ecoli_ShotgunPE [Genomic]
Location: /data/Datasets/Genomic/Ecoli_RL/P_Assembly_Ecoli_ShotgunPE
Ready for analysis

☑ Incremental de novo assembler analysis
Number of CPUs to use (0=all): 1
Overlap Detection
Seed step: 12
Seed length: 16
Seed count: 1
Minimum overlap length: 40
Minimum overlap identity: 90 %
Alignment identity score: 2
Alignment difference score: -3

□ Use duplicate reads
□ Use serial I/O
□ Extend low depth overlaps

```
Info    P_Assembly_Ecoli_ShotgunPE : Project created successfully.
Info    P_Assembly_Ecoli_ShotgunPE : The project will be written to disk when run, a new project is opened, or upon exit.
```

Figure 13: gsAssembler Parameters tab, Computation sub-tab for Genomic projects.

- **Incremental *De Novo* assembler analysis (-r)** – Results from computations performed with this option selected (checked) will serve as the basis for the assembly of additional reads. If a project computation is performed with this option deselected, all the project data will be assembled anew each time it is computed, *i.e.* forcing all alignments to be recalculated and overwriting any existing assembly results.
    - Default: selected (Note: This setting is not saved with a project, and is always reset to "selected" when a project is re-opened).
- The **number of CPUs (-cpu)** option can be given to limit the load of the software on the computing resources. The default is 1, which specifies that one of the CPUs on the computer should be used. The computation uses the CPU setting to create an approximate load on the computer, so the actual load may vary somewhat from this number, *i.e.*, using "-cpu 3" may cause a load from 2 to 4 on the system.

Currently, only the "Computing alignments…", "Mapping short reads to consensi…", and "Generating output…" phases have been parallelized (because they are the most computationally intensive phases for larger projects). Other phases of the computation will still use single threading. The parallel assembler only runs on shared memory, *i.e.* only CPUs with access to the same physical memory can be used.

The current memory footprint is 3 bytes per input (read) base. For example a 1 Gbp genome with a 20x coverage would need 60 GB of physical memory. Please note that the memory footprint can increase when working with a genome containing high-copy count repeat sequences, or when using multiple cpus.

- **Use duplicate reads (-ud)** – when checked, the Assembler will include duplicate reads when computing the consensus for a contig. Duplicate reads are a known potential artifact of the emPCR Amplification process which may occur, for example, when two or more beads are amplified in a single emulsion microreactor. In such cases, duplicate reads should be excluded from the computations, which is the default.
- **Use serial I/O (-sio)** – select this option if you are using many .sff files (> 4 million reads) in your project. The option instructs the software to prepare an intermediate read file in which the reads appear in the same order as they do in the alignments. For large projects, this can speed up the execution of the Assembler, especially during the Output Phase of its computation. For more information, see Section 4.4.
- **Extend low depth overlaps (-urt)** – select this option to instruct the assembler to extend contigs even when the depth falls to a single read. This option can be especially useful as a means to obtain more complete contigging across low-depth portions of genomic and transcriptomic assemblies. For more information, see Section 4.10 on the –urt option.

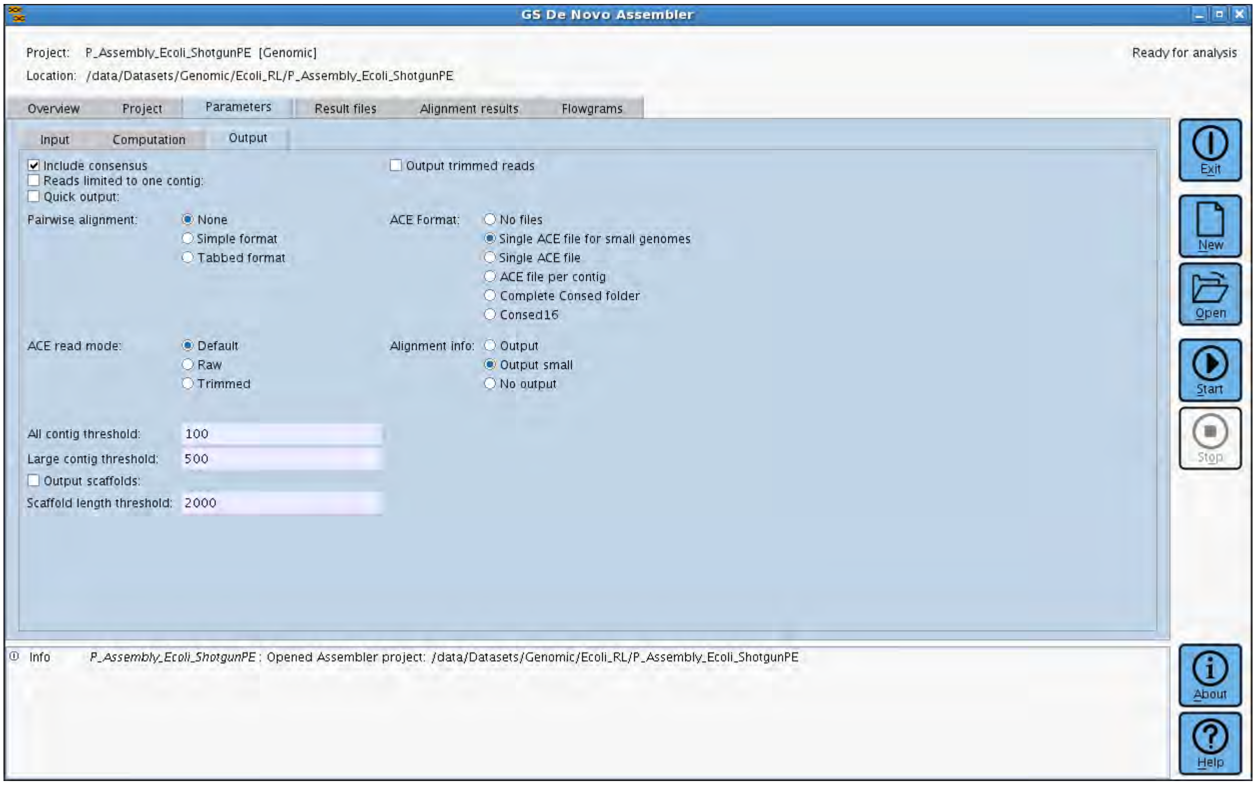
- **Overlap Detection** Parameters:

Parameter	Description	Default Value	Allowed Values
Seed step (-ss)	The number of bases between seed generation locations used in the exact k-mer matching part of the overlap detection.	12	>= 1
Seed length (-sl)	The number of bases used for each seed in the exact k-mer matching part of the overlap detection (*i.e.* the "k" value of the k-mer matching).	16	6 - 32
Seed count (-sc)	The number of seeds required in a window before an extension is made.	1	>= 1
Minimum overlap length (-ml)	The minimum length of overlaps used by the assembler for the pairwise alignment step. This value can be entered as either a number of bases or a percent of read length. To enter a percent, the % symbol must be used (*e.g.* 37%).	40	>= 1 (length) 1 – 100 (percent)
Minimum overlap identity (-mi)	The minimum percent identity of overlaps used by the assembler for the pairwise alignment step.	90	1 - 100
Alignment identity score (-ais)	When multiple overlaps are found, the per-overlap-column identity score used to sort the overlaps for use in the progressive alignment.	2	>= 0
Alignment difference score (-ads)	When multiple overlaps are found, the per-overlap-column difference score used to sort the overlaps for use in the progressive multi-alignment.	-3	<= 0

For a description of the overlap detection parameters for assembly, please see Section 4.2.

1.7.3 Genomic Project Output Sub-Tab

The Output sub-tab for genomic projects is shown on Figure 14 (for cDNA projects, see Section 1.7.5). It allows you to adjust the settings described below.



Project: P_Assembly_Ecoli_ShotgunPE [Genomic]
Location: /data/Datasets/Genomic/Ecoli_RL/P_Assembly_Ecoli_ShotgunPE

Ready for analysis

☑ Include consensus
□ Reads limited to one contig:
□ Quick output:

□ Output trimmed reads

Pairwise alignment: ● None ○ Simple format ○ Tabbed format

ACE Format: ○ No files ● Single ACE file for small genomes ○ Single ACE file ○ ACE file per contig ○ Complete Consed folder ○ Consed16

ACE read mode: ● Default ○ Raw ○ Trimmed

Alignment info: ○ Output ● Output small ○ No output

All contig threshold: 100
Large contig threshold: 500
□ Output scaffolds:
Scaffold length threshold: 2000

Info P_Assembly_Ecoli_ShotgunPE : Opened Assembler project: /data/Datasets/Genomic/Ecoli_RL/P_Assembly_Ecoli_ShotgunPE

Figure 14: gsAssembler Parameters tab, Output sub-tab for Genomic projects.

- **Include consensus (-no not specified)** – If this check box is selected, the application will generate all the output files, including the output files related to the generation of contigs and scaffold information. If it is not selected, the application will perform the assembly and will output files (the 454ReadStatus.txt and 454TrimStatus.txt files) associated with the creation of the multiple-alignments, but will not output files (the 454AllContigs.fna and .qual, 454LargeContigs.fna and .qual, scaffold files, *etc.*) or metrics (latter parts of the 454NewblerMetrics.txt file) involved with contig or consensus information. The file ContigGraph.txt is output, but is empty. If this option is unchecked, the assembly will run faster, providing an approximate idea of the quality of the data, based on the percentage of reads assembled.
  - Default: selected (Note: This setting is not persistent (saved with a project), and is always reset to "selected" when a project is re-opened.).

- The **Reads limited to one contig (-rip)** option tells the assembler to generate output where each read occurs in one and only one contig (*i.e.* read alignments are not allowed to span across multiple contigs). This creates output ace and consed files suitable for 3rd party software for further data analysis. The function operates after the contigs and scaffolds are created, by attempting to remove reads from the ends of contigs that contain sets of reads that branch to multiple contigs. When a read is found that extends from such a contig to another contig end that doesn't branch, the part of the read in the branched contig is removed and placed at the end of the part of the read in the non-branching contig.

Caution: the use of this option results in the loss of information, as connections between unique and repeat contigs are removed.

- The **Quick output** (**-qo**) option generates output faster than using the default setting by disabling the "Computing signals…" computation. The default behavior of the assembler requires that all reads be re-read so that signal and quality information can be used to compute consensus (see Section 1.1). If this option is selected, the accuracy of the consensus may be degraded (*i.e.*, more consensus errors may be generated) as the distribution statistics are not generated to assist in the basecalling step.

- **Output trimmed reads** (**-tr**) – select this option to output the read sequences to a .fna file after trimming has been performed (includes vector / primer / paired end linker / MIDs / low quality trimming). If available, the quality value for the output reads will be placed in a .qual file.

- **Output scaffolds** (**-scaffold**) – select this option to use 454ContigScaffolds.txt, 454Scaffolds.fna/qual and 454ScaffoldContigs.fna/qual as the standard assembly results instead of 454AllContigs.fna/qual and 454LargeContigs.fna/qual. Use –scaffold, with or without paired end data, to generate assembly results that are eligible to be submitted to NCBI. When -scaffold is selected, the software more aggressively attempts to close/fill gaps using additional information. For example, it will attempt to use portions of reads that were previously trimmed to search for overlaps between the ends of contigs. It will also attempt to place instances of repeats in the scaffolds. This will result in more complete contigging. Rather than representing a unique contig and a repeat contig independently, the two will be joined together.

- The **Ace Format** option determines whether or what form of ACE file(s) are output by the application. The default is "Single ACE file for small genomes". See section 1.16.1.5 for a description of the 454Contigs.ace file and of the ace and consed directories. By default, files consistent with version 17.0 or higher of consed are generated.
  - No files (-noace | -nobig) – no ACE file is generated.
  - Single ACE file for small genomes (default, no option specified) – a single ACE file is generated if fewer than 4 million reads are input to the assembly.
  - Single ACE file (-ace) – a single ACE file is generated containing the multiple alignments of all the contigs in the assembly, irrespective of the size of the genome.
  - ACE file per contig (-acedir) – a separate ACE file is generated for each contig in the assembly.
  - Complete Consed folder (-consed) – a "consed" folder is generated, containing all the directories and files necessary to display the data in the consed software.
  - Consed16 (-consed16) – This option generates files consistent with version 16.0 or earlier.

- The **Alignment info** option has 3 possible settings.
  - Output (-infoall) – turns on generation of the 454AlignmentInfo.tsv file.
  - No output (-noinfo) – suppresses the generation of the 454AlignmentInfo.tsv file.
  - Output small (-info) – the file will be generated unless there are more than 4 M reads or the assembly generates contigs with a total length in excess of 40 Mbp.

- **Pairwise alignment** – Determines whether the 454PairAlign.txt file is output; this file contains the overlaps used by the assembler. See section 1.16.1.8 for a description of the 454PairAlign.txt file.
    - None (-nop | -nobig) – the file is not generated (default).
    - Simple format (-pair) – the file contains a human-readable view of the alignments.
    - Tabbed format (-pt | -pairt) – the file contains tab-delimited lines of the overlaps.
- **Ace read mode** – When the ACE file is generated, output reads using either the raw, complete basecalled read or using just the trimmed portion of the reads (after low quality, vector and key trimming).
    - Default (-ad) is set to output trimmed reads for assembly.
    - Raw (-ar) is used to output the entire sequence of reads.
    - Trimmed (-at) is used to output trimmed reads.
- **All contig threshold (-a)** – The minimum number of bases for a contig to be output in the 454AllContigs.fna file.
    - Default: 100
- **Large contig threshold (-l)** – The minimum number of bases for a contig to be output in the 454LargeContigs.fna file.
    - Default: 500
- **Scaffold length threshold (-s)** – The minimum number of bases for a scaffold to be output.
    - Default: 2000

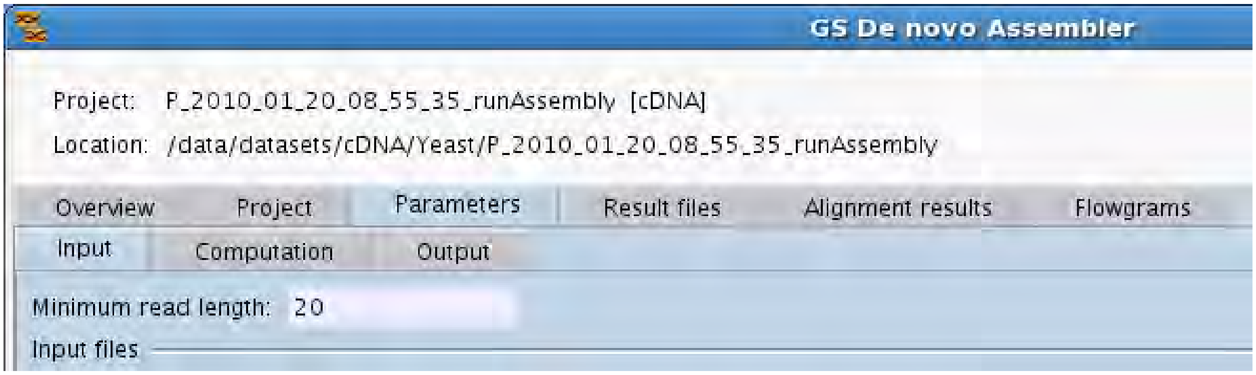
The 454AllContigs.fna file can be generated with the -a (all contigs length) threshold option, which filters out contigs whose lengths are less than the threshold. On rare occasions, the file may contain contigs whose lengths are less than this threshold. This is due to the two-phase nature by which contig consensi are determined. The –a setting is ignored in cDNA assembly computations.

For a full listing and description of the output options for assembly, please see Section 4.2.

1.7.4 cDNA Project Input Sub-Tab

The cDNA Assembly project parameters on the Input sub-tab are the same as those of Genomic Assembly projects (Section 1.7.1) with the exceptions noted below.

- cDNA Assembly projects do not have checkboxes for **Large or Complex Genome** (**-large**) or **Heterozygotic Mode** (**-het**), or a text field for entering **Expected Depth** (**-e**), as these are not relevant to cDNA projects (Figure 15).



GS De novo Assembler

Project: P_2010_01_20_08_55_35_runAssembly [cDNA]

Location: /data/datasets/cDNA/Yeast/P_2010_01_20_08_55_35_runAssembly

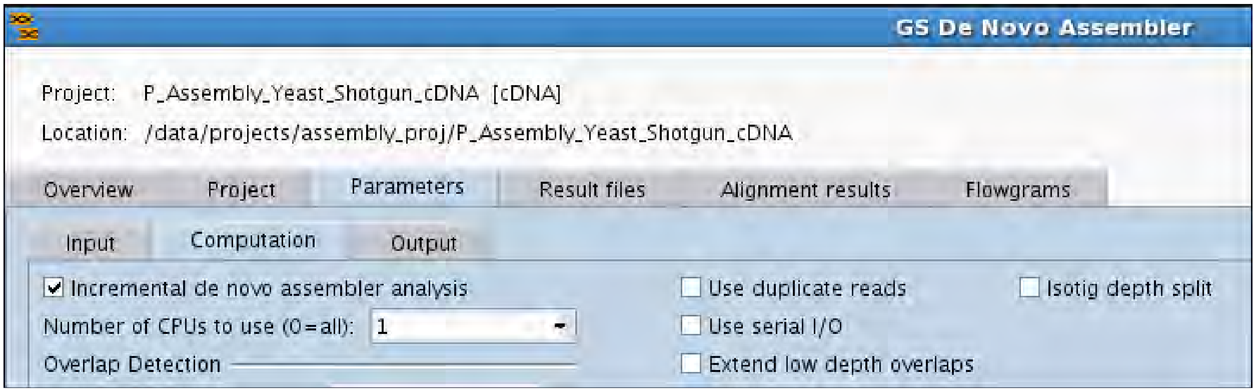
Minimum read length: 20

Figure 15: gsAssembler Parameters tab, Input sub-tab for cDNA projects (top-left corner).

1.7.5 cDNA Project Computation Sub-Tab

The cDNA Assembly project parameters (Figure 16) on the Computation sub-tab are the same as those of Genomic Assembly projects (Section 1.7.2) with the exception noted below.

- cDNA Assembly projects have an option to select the **Isotig depth split** (**-isplit**) option. (See 1.15 for a description of this option.).



GS De Novo Assembler

Project: P_Assembly_Yeast_Shotgun_cDNA [cDNA]

Location: /data/projects/assembly_proj/P_Assembly_Yeast_Shotgun_cDNA

☑ Incremental de novo assembler analysis

Number of CPUs to use (0=all): 1

□ Use duplicate reads

□ Use serial I/O

□ Extend low depth overlaps

□ Isotig depth split

Figure 16: gsAssembler Parameters tab, Computation sub-tab for cDNA projects (top).

1.7.6 cDNA Project Output Sub-Tab

The cDNA Assembly project parameters (Figure 17) on the Output sub-tab are the same as those of Genomic Assembly projects (Section 1.7.3) with the exceptions noted below.

- cDNA Assembly projects have options to specify the **Isotig threshold** (**-it**) and **Isogroup threshold** (**-ig**), as well as **Isotig contig count threshold** (**-icc**) and **Isotig contig length threshold** (**-icl**). (See 1.15 for a description of these thresholds.).
- cDNA Assembly projects do not have a checkboxes for **Scaffold length threshold** (**-s**), as this is not relevant to cDNA projects

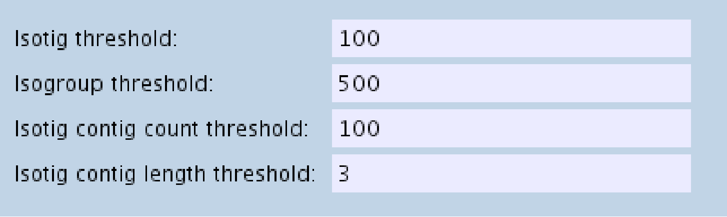

Isotig threshold: 100
Isogroup threshold: 500
Isotig contig count threshold: 100
Isotig contig length threshold: 3

Figure 17: gsAssembler Parameters tab, Output sub-tab, bottom-left corner, showing the Isotig and Isogroup fields specific to cDNA projects

1.8 Computing the Assembly

1.8.1 Computing an Assembly for the First Time

When at least one Read Data file has been added to the project and all parameters are within acceptable limits, the **Start** button becomes enabled and the "Ready for analysis" message appears in the upper right corner of the application window (Figure 14). The projects parameter settings are also saved when the **Start** button is clicked. Click the **Start** button to carry out the assembly computation of the current project, *i.e.* as defined in the Project and Parameters tabs. As the computation progresses, the **Start** button is disabled, periodic progress messages appear in the progress box, at the bottom of the application's main window, and the current stage of the assembly computation along with a progress bar are displayed in the upper right corner of the window. When the assembly computation is complete, the **Start** button is re-enabled and the message "Ready for analysis" appears again in the upper right corner of the application window.

Transcriptome and complex genome assemblies with high depth regions may not report progress for extended periods of time, even though the assembly is progressing normally. Do not stop the computation of complex projects simply because it has not reported progress for many hours, or even a full day.

1.8.2 Re-Computing an Assembly

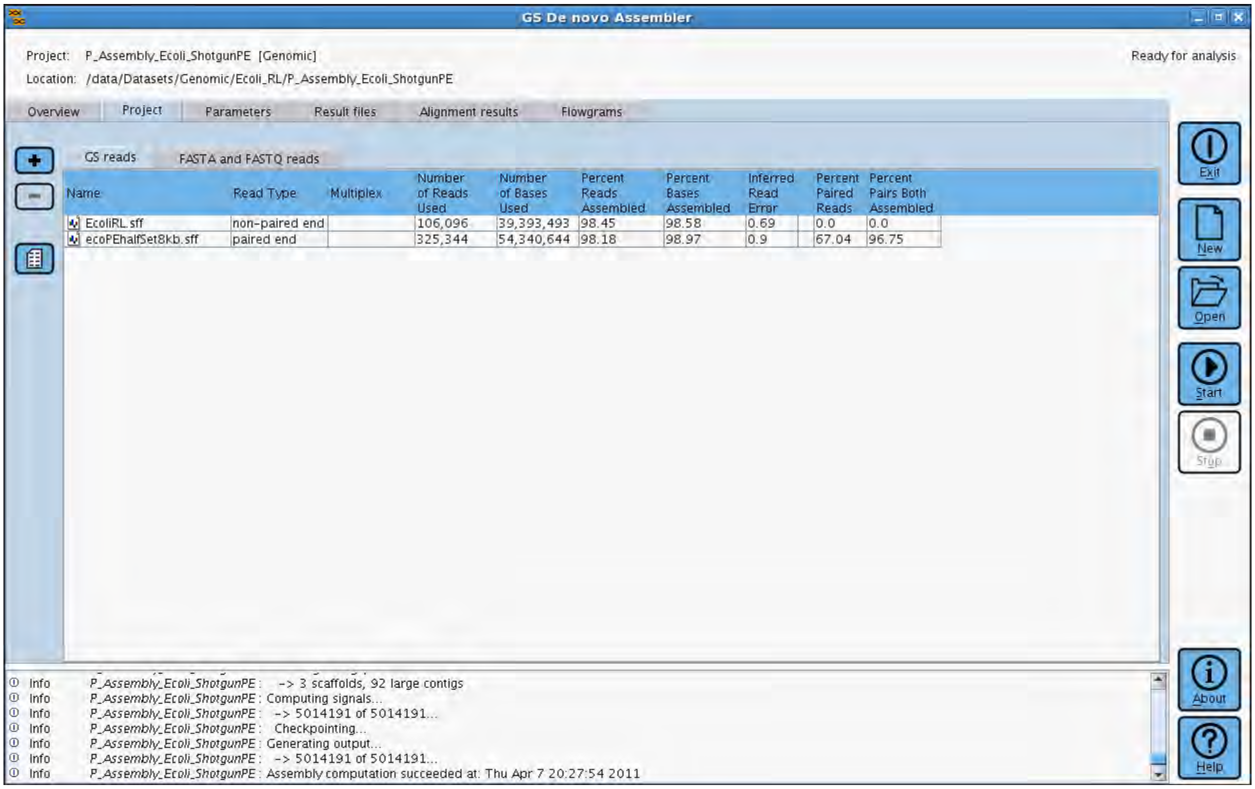
After an assembly has been performed on a project, the assembly computation can be repeated on that project, with the same Read Data or after addition or removal of one or more files, with the same assembly parameter settings or with different ones by pressing the **Start** button. If the “Incremental *De Novo* assembler analysis” checkbox on the Computation sub-tab of the Parameters tab is checked, this re-analysis will take much less time than the initial analysis, since only some data will be re-computed. Changes to some parameter settings (such as those on the Input and Computation sub-tabs of the Project Tab) will have no effect on reads already included in the prior assembly computation.

1.8.3 Stopping an Assembly Computation

Click on the **Stop** button while an assembly project computation is in progress to halt the computation. The effect may not be instantaneous, however. Any delay will be a function of the complexity of the assembly being performed and of the stage of the computation at the time it was interrupted. An informational message appears in the progress box at the bottom of the application’s main window, containing the text “Warning: stopRun called for this project. Stopping….” When the assembly computation has been halted, the **Stop** button is disabled and the **Start** button re-enabled.

1.8.4 Information Updated when Assembly Completes

After a successful assembly, the data for the various columns of the reads in the GS reads and the FASTA and FASTQ reads sub-tabs will be updated (Figure 18). Placing the mouse pointer over any cell in the table brings up a tool tip with additional information about that item. A successful assembly also updates data in the Overview tab. Project information is also saved when the computation completes.



Project: P_Assembly_Ecoli_ShotgunPE [Genomic]
Location: /data/Datasets/Genomic/Ecoli_RL/P_Assembly_Ecoli_ShotgunPE
Ready for analysis

Name	Read Type	Multiplex	Number of Reads Used	Number of Bases Used	Percent Reads Assembled	Percent Bases Assembled	Inferred Read Error	Percent Paired Reads	Percent Pairs Both Assembled
EcoliRL.sff	non-paired end		106,096	39,393,493	98.45	98.58	0.69	0.0	0.0
ecoPEhalfSet8kb.sff	paired end		325,344	54,340,644	98.18	98.97	0.9	67.04	96.75

```
Info    P_Assembly_Ecoli_ShotgunPE :   -> 3 scaffolds, 92 large contigs
Info    P_Assembly_Ecoli_ShotgunPE : Computing signals...
Info    P_Assembly_Ecoli_ShotgunPE :   -> 5014191 of 5014191...
Info    P_Assembly_Ecoli_ShotgunPE :   Checkpointing...
Info    P_Assembly_Ecoli_ShotgunPE : Generating output...
Info    P_Assembly_Ecoli_ShotgunPE :   -> 5014191 of 5014191...
Info    P_Assembly_Ecoli_ShotgunPE : Assembly computation succeeded at: Thu Apr 7 20:27:54 2011
```

Figure 18: Project Tab of the gsAssembler (GS Reads sub-tab), after completion of an assembly.

After an assembly project has been computed, the results of the assembly can be viewed on the Overview, Project, Results Files, Alignment Results, and Flowgram tabs of the GS *De Novo* Assembly application's main window.

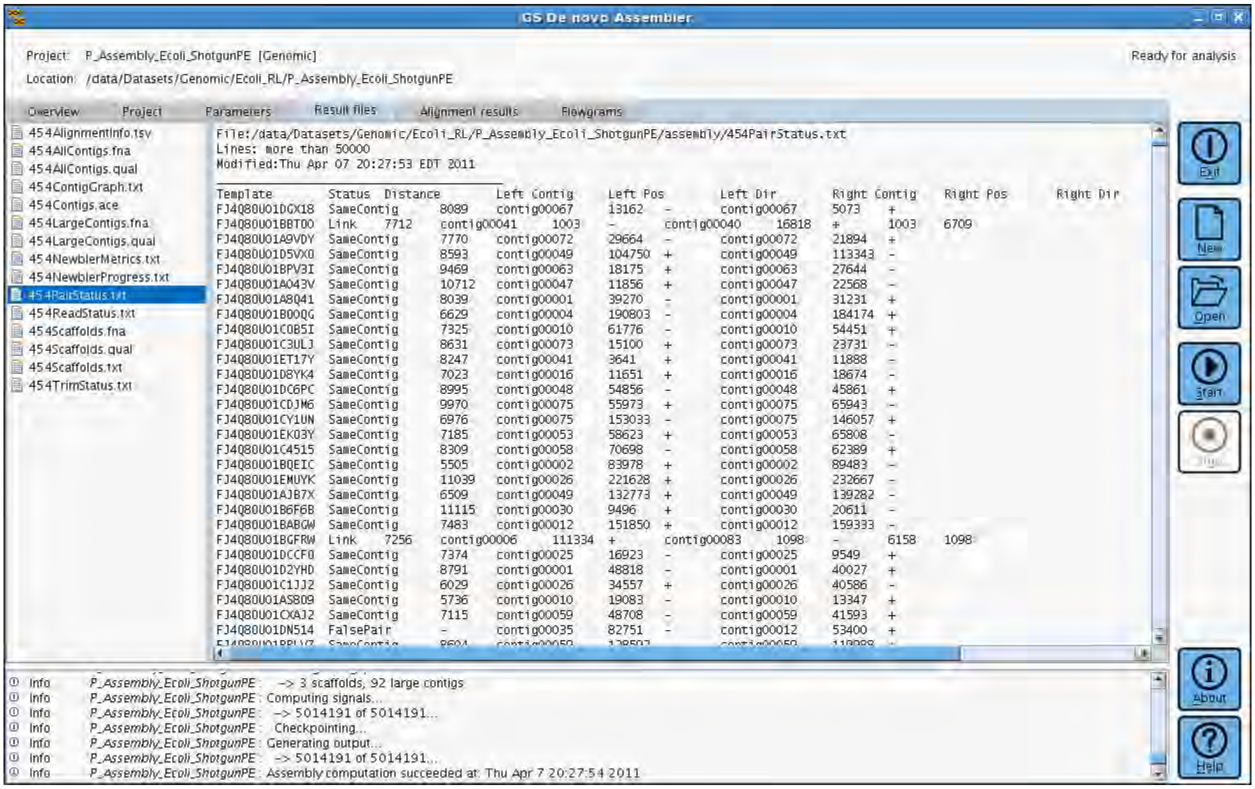
In some cases where large (greater than 100 Mbp) or complex genomes are being assembled, the computation may not indicate its progress for an extended period of time. When this happens, the GUI may incorrectly indicate that the computation has stalled. The computation should be given several more hours to complete if a large or complex genome is being assembled. In these cases, please note that the assembly is still running in the background. Using the top command will list the currently active Newbler Process.

1.9 Viewing Assembly Output with the Result Files Tab

The Result Files tab allows you to view the various metrics files generated by the GS *De Novo* Assembly software (described in detail in Section 1.16.1). Upper case and lower case letters are used to indicate basecall quality in DNA sequence files. Lower case letters in individual trimmed reads represent bases with a basecall accuracy of <99% (Q<20). In contigs lower case letters represent bases with a basecall accuracy of <99.99% (Q<40). Lower case letters are also used to indicate the bases that comprise the sequencing key when raw (untrimmed) SFF reads appear in FASTA or ACE formats.

Using the list in the left-hand panel of the tab, click on the name of the output file you want to examine; its content will be displayed on the right-hand panel of the tab (Figure 19).

> For very long files, such as the sequences of all contigs of a large assembled genome, this view displays the file truncated to 50,000 lines. When this occurs, a note to that effect specifies that it is so both at the beginning and at the end of the display. The file itself remains intact, however.



Project: P_Assembly_Ecoli_ShotgunPE [Genomic]
Location: /data/Datasets/Genomic/Ecoli_RL/P_Assembly_Ecoli_ShotgunPE
Ready for analysis

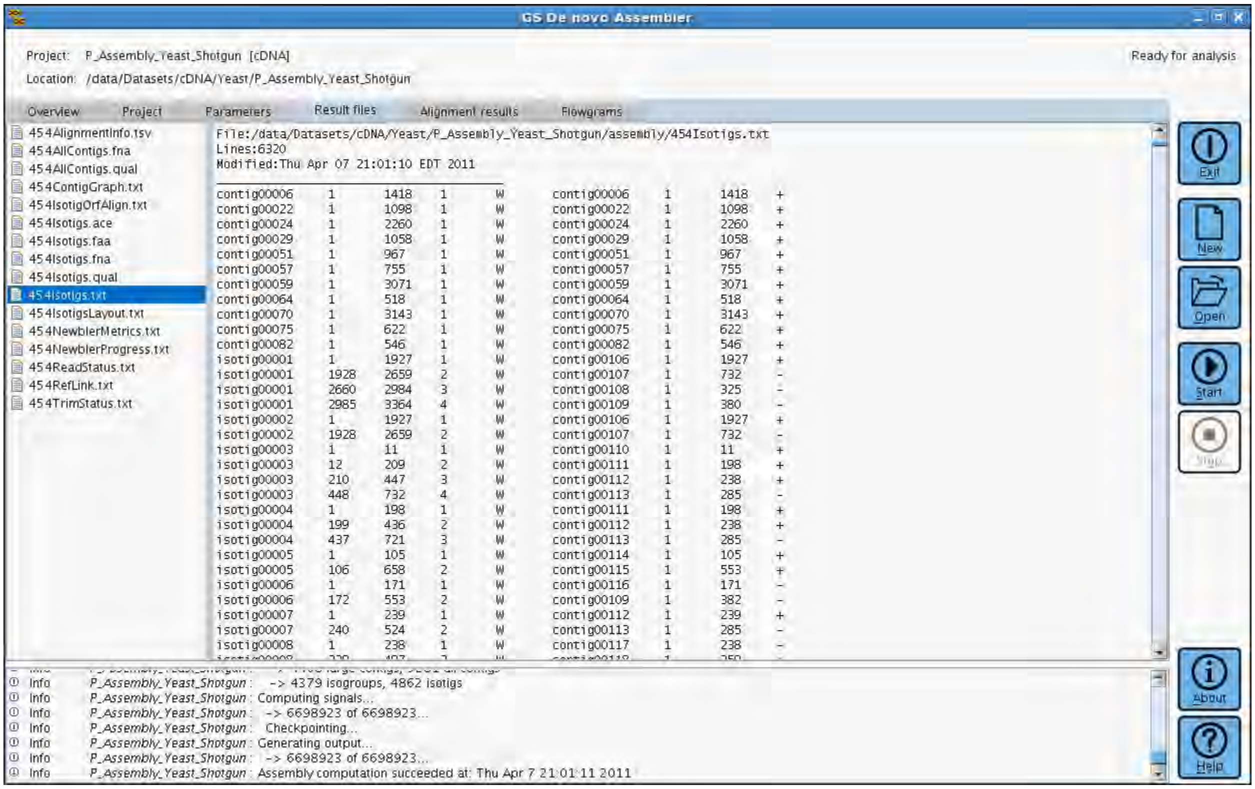
- 454AlignmentInfo.tsv
- 454AllContigs.fna
- 454AllContigs.qual
- 454ContigGraph.txt
- 454Contigs.ace
- 454LargeContigs.fna
- 454LargeContigs.qual
- 454NewblerMetrics.txt
- 454NewblerProgress.txt
- 454PairStatus.txt
- 454ReadStatus.txt
- 454Scaffolds.fna
- 454Scaffolds.qual
- 454Scaffolds.txt
- 454TrimStatus.txt

```
File:/data/Datasets/Genomic/Ecoli_RL/P_Assembly_Ecoli_ShotgunPE/assembly/454PairStatus.txt
Lines: more than 50000
Modified:Thu Apr 07 20:27:53 EDT 2011
________________________________________
Template         Status  Distance        Left Contig     Left Pos        Left Dir        Right Contig    Right Pos       Right Dir
FJ4Q80U01DGX18   SameContig      8089    contig00067     13162   -       contig00067     5073    +
FJ4Q80U01BBT00   Link    7712    contig00041     1003    -       contig00040     16818   +       1003    6709
FJ4Q80U01A9VDY   SameContig      7770    contig00072     29664   -       contig00072     21894   +
FJ4Q80U01D5VX0   SameContig      8593    contig00049     104750  +       contig00049     113343  -
FJ4Q80U01BPV3I   SameContig      9469    contig00063     18175   +       contig00063     27644   -
FJ4Q80U01A043V   SameContig      10712   contig00047     11856   +       contig00047     22568   -
FJ4Q80U01A8Q41   SameContig      8039    contig00001     39270   -       contig00001     31231   +
FJ4Q80U01B00QG   SameContig      6629    contig00004     190803  -       contig00004     184174  +
FJ4Q80U01C0B5I   SameContig      7325    contig00010     61776   -       contig00010     54451   +
FJ4Q80U01C3ULJ   SameContig      8631    contig00073     15100   +       contig00073     23731   -
FJ4Q80U01ET17Y   SameContig      8247    contig00041     3641    +       contig00041     11888   -
FJ4Q80U01D8YK4   SameContig      7023    contig00016     11651   +       contig00016     18674   -
FJ4Q80U01DC6PC   SameContig      8995    contig00048     54856   -       contig00048     45861   +
FJ4Q80U01CDJM6   SameContig      9970    contig00075     55973   +       contig00075     65943   -
FJ4Q80U01CY1UN   SameContig      6976    contig00075     153033  -       contig00075     146057  +
FJ4Q80U01EK03Y   SameContig      7185    contig00053     58623   +       contig00053     65808   -
FJ4Q80U01C4515   SameContig      8309    contig00058     70698   -       contig00058     62389   +
FJ4Q80U01BQEIC   SameContig      5505    contig00002     83978   +       contig00002     89483   -
FJ4Q80U01EMUYK   SameContig      11039   contig00026     221628  +       contig00026     232667  -
FJ4Q80U01AJB7X   SameContig      6509    contig00049     132773  +       contig00049     139282  -
FJ4Q80U01B6F6B   SameContig      11115   contig00030     9496    +       contig00030     20611   -
FJ4Q80U01BABGW   SameContig      7483    contig00012     151850  +       contig00012     159333  -
FJ4Q80U01BGFRW   Link    7256    contig00006     111334  +       contig00083     1098    -       6158    1098
FJ4Q80U01DCCF0   SameContig      7374    contig00025     16923   -       contig00025     9549    +
FJ4Q80U01D2YHD   SameContig      8791    contig00001     48818   -       contig00001     40027   +
FJ4Q80U01C1JJ2   SameContig      6029    contig00026     34557   +       contig00026     40586   -
FJ4Q80U01AS809   SameContig      5736    contig00010     19083   -       contig00010     13347   +
FJ4Q80U01CXAJ2   SameContig      7115    contig00059     48708   -       contig00059     41593   +
FJ4Q80U01DN514   FalsePair       -       contig00035     82751   -       contig00012     53400   +
```

```
Info    P_Assembly_Ecoli_ShotgunPE :   -> 3 scaffolds, 92 large contigs
Info    P_Assembly_Ecoli_ShotgunPE : Computing signals...
Info    P_Assembly_Ecoli_ShotgunPE :   -> 5014191 of 5014191...
Info    P_Assembly_Ecoli_ShotgunPE :   Checkpointing...
Info    P_Assembly_Ecoli_ShotgunPE : Generating output...
Info    P_Assembly_Ecoli_ShotgunPE :   -> 5014191 of 5014191...
Info    P_Assembly_Ecoli_ShotgunPE : Assembly computation succeeded at: Thu Apr 7 20:27:54 2011
```

Figure 19: gsAssembler Result Files tab for Genomic projects.

Figure 19 above shows the Result Files tab for a Genomic Assembly project. The result files for a cDNA project are listed below (Figure 20). The result file contents are described in detail in Section 1.16.1.



Project: P_Assembly_Yeast_Shotgun [cDNA]
Location: /data/Datasets/cDNA/Yeast/P_Assembly_Yeast_Shotgun

- 454AlignmentInfo.tsv
- 454AllContigs.fna
- 454AllContigs.qual
- 454ContigGraph.txt
- 454IsotigOrfAlign.txt
- 454Isotigs.ace
- 454Isotigs.faa
- 454Isotigs.fna
- 454Isotigs.qual
- 454Isotigs.txt
- 454IsotigsLayout.txt
- 454NewblerMetrics.txt
- 454NewblerProgress.txt
- 454ReadStatus.txt
- 454RefLink.txt
- 454TrimStatus.txt

```
File:/data/Datasets/cDNA/Yeast/P_Assembly_Yeast_Shotgun/assembly/454Isotigs.txt
Lines:6320
Modified:Thu Apr 07 21:01:10 EDT 2011
_______________________________________
contig00006    1      1418   1     W     contig00006    1     1418   +
contig00022    1      1098   1     W     contig00022    1     1098   +
contig00024    1      2260   1     W     contig00024    1     2260   +
contig00029    1      1058   1     W     contig00029    1     1058   +
contig00051    1      967    1     W     contig00051    1     967    +
contig00057    1      755    1     W     contig00057    1     755    +
contig00059    1      3071   1     W     contig00059    1     3071   +
contig00064    1      518    1     W     contig00064    1     518    +
contig00070    1      3143   1     W     contig00070    1     3143   +
contig00075    1      622    1     W     contig00075    1     622    +
contig00082    1      546    1     W     contig00082    1     546    +
isotig00001    1      1927   1     W     contig00106    1     1927   +
isotig00001    1928   2659   2     W     contig00107    1     732    -
isotig00001    2660   2984   3     W     contig00108    1     325    -
isotig00001    2985   3364   4     W     contig00109    1     380    -
isotig00002    1      1927   1     W     contig00106    1     1927   +
isotig00002    1928   2659   2     W     contig00107    1     732    -
isotig00003    1      11     1     W     contig00110    1     11     +
isotig00003    12     209    2     W     contig00111    1     198    +
isotig00003    210    447    3     W     contig00112    1     238    +
isotig00003    448    732    4     W     contig00113    1     285    -
isotig00004    1      198    1     W     contig00111    1     198    +
isotig00004    199    436    2     W     contig00112    1     238    +
isotig00004    437    721    3     W     contig00113    1     285    -
isotig00005    1      105    1     W     contig00114    1     105    +
isotig00005    106    658    2     W     contig00115    1     553    +
isotig00006    1      171    1     W     contig00116    1     171    -
isotig00006    172    553    2     W     contig00109    1     382    -
isotig00007    1      239    1     W     contig00112    1     239    +
isotig00007    240    524    2     W     contig00113    1     285    -
isotig00008    1      238    1     W     contig00117    1     238    -
```

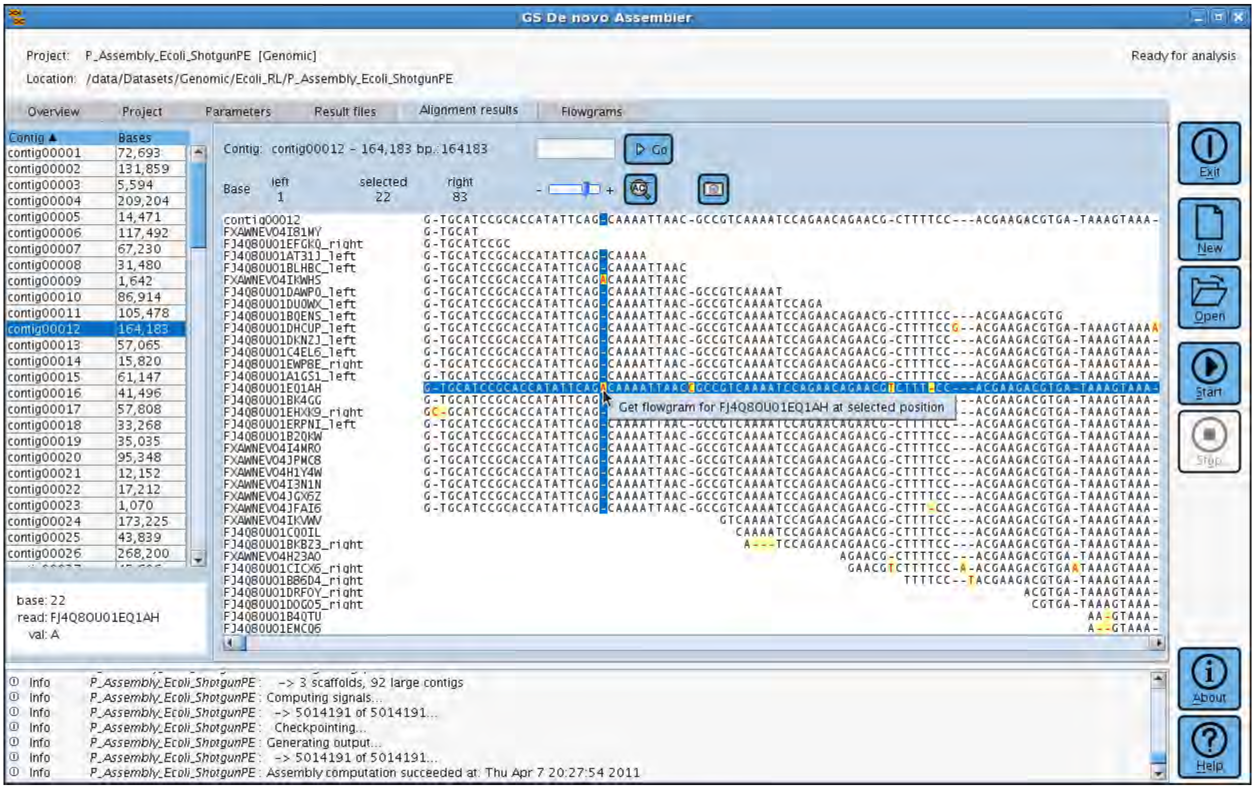
```
Info    P_Assembly_Yeast_Shotgun :   -> 4379 isogroups, 4862 isotigs
Info    P_Assembly_Yeast_Shotgun : Computing signals...
Info    P_Assembly_Yeast_Shotgun :   -> 6698923 of 6698923...
Info    P_Assembly_Yeast_Shotgun :   Checkpointing...
Info    P_Assembly_Yeast_Shotgun : Generating output...
Info    P_Assembly_Yeast_Shotgun :   -> 6698923 of 6698923...
Info    P_Assembly_Yeast_Shotgun : Assembly computation succeeded at: Thu Apr 7 21:01:11 2011
```

Figure 20: gsAssembler Results Files tab for cDNA projects.

1.10 Viewing Contigs with the Alignment Results Tab

Click on the Alignment Results tab to view the alignment data from the assembly computation (Figure 21). This tab shows the contigs and their underlying multiple alignments.

> Consensus sequence supported by a single read is considered unreliable. Regions of the assembly covered by only a single read are excluded from the assembly results. This exclusion could possibly result in contig shrinkage or breakage.



Project: P_Assembly_Ecoli_ShotgunPE [Genomic]

Location: /data/Datasets/Genomic/Ecoli_RL/P_Assembly_Ecoli_ShotgunPE

Ready for analysis

Contig ▲	Bases
contig00001	72,693
contig00002	131,859
contig00003	5,594
contig00004	209,204
contig00005	14,471
contig00006	117,492
contig00007	67,230
contig00008	31,480
contig00009	1,642
contig00010	86,914
contig00011	105,478
contig00012	164,183
contig00013	57,065
contig00014	15,820
contig00015	61,147
contig00016	41,496
contig00017	57,808
contig00018	33,268
contig00019	35,035
contig00020	95,348
contig00021	12,152
contig00022	17,212
contig00023	1,070
contig00024	173,225
contig00025	43,839
contig00026	268,200

base: 22
read: FJ4Q8OU01EQ1AH
val: A

Contig: contig00012 – 164,183 bp.:164183

Base	left	selected	right
	1	22	83

```
contig00012                G-TGCATCCGCACCATATTCAG-CAAAATTAAC-GCCGTCAAAATCCAGAACAGAACG-CTTTTCC---ACGAAGACGTGA-TAAAGTAAA-
FXAWNEV04I81MY             G-TGCAT
FJ4Q80U01EFGKQ_right       G-TGCATCCGC
FJ4Q80U01AT31J_left        G-TGCATCCGCACCATATTCAG-CAAAA
FJ4Q80U01BLHBC_left        G-TGCATCCGCACCATATTCAG-CAAAATTAAC
FXAWNEV04IKWHS             G-TGCATCCGCACCATATTCAGACAAAATTAAC
FJ4Q80U01DAWP0_left        G-TGCATCCGCACCATATTCAG-CAAAATTAAC-GCCGTCAAAAT
FJ4Q80U01DUOWX_left        G-TGCATCCGCACCATATTCAG-CAAAATTAAC-GCCGTCAAAATCCAGA
FJ4Q80U01BQENS_left        G-TGCATCCGCACCATATTCAG-CAAAATTAAC-GCCGTCAAAATCCAGAACAGAACG-CTTTTCC---ACGAAGACGTG
FJ4Q80U01DHCUP_left        G-TGCATCCGCACCATATTCAG-CAAAATTAAC-GCCGTCAAAATCCAGAACAGAACG-CTTTTCCG--ACGAAGACGTGA-TAAAGTAAAA
FJ4Q80U01DKNZJ_left        G-TGCATCCGCACCATATTCAG-CAAAATTAAC-GCCGTCAAAATCCAGAACAGAACG-CTTTTCC---ACGAAGACGTGA-TAAAGTAAA-
FJ4Q80U01C4EL6_left        G-TGCATCCGCACCATATTCAG-CAAAATTAAC-GCCGTCAAAATCCAGAACAGAACG-CTTTTCC---ACGAAGACGTGA-TAAAGTAAA-
FJ4Q80U01EWP8E_right       G-TGCATCCGCACCATATTCAG-CAAAATTAAC-GCCGTCAAAATCCAGAACAGAACG-CTTTTCC---ACGAAGACGTGA-TAAAGTAAA-
FJ4Q80U01A1GS1_left        G-TGCATCCGCACCATATTCAG-CAAAATTAAC-GCCGTCAAAATCCAGAACAGAACG-CTTTTCC---ACGAAGACGTGA-TAAAGTAAA-
FJ4Q80U01EQ1AH             G-TGCATCCGCACCATATTCAGACAAAATTAACCGCCGTCAAAATCCAGAACAGAACGTCTTT-CC---ACGAAGACGTGA-TAAAGTAAA-
FJ4Q80U01BK4GG             G-TGCATCCGCACCATATTCAG Get flowgram for FJ4Q8OU01EQ1AH at selected position --ACGAAGACGTGA-TAAAGTAAA-
FJ4Q80U01EHXK9_right       GC-GCATCCGCACCATATTCAG-                                             --ACGAAGACGTGA-TAAAGTAAA-
FJ4Q80U01ERPNI_left        G-TGCATCCGCACCATATTCAG-CAAAATTAAC-GCCGTCAAAATCCAGAACAGAACG-CTTTTCC---ACGAAGACGTGA-TAAAGTAAA-
FJ4Q80U01B2QKW             G-TGCATCCGCACCATATTCAG-CAAAATTAAC-GCCGTCAAAATCCAGAACAGAACG-CTTTTCC---ACGAAGACGTGA-TAAAGTAAA-
FXAWNEV04I4MR0             G-TGCATCCGCACCATATTCAG-CAAAATTAAC-GCCGTCAAAATCCAGAACAGAACG-CTTTTCC---ACGAAGACGTGA-TAAAGTAAA-
FXAWNEV04JPMC8             G-TGCATCCGCACCATATTCAG-CAAAATTAAC-GCCGTCAAAATCCAGAACAGAACG-CTTTTCC---ACGAAGACGTGA-TAAAGTAAA-
FXAWNEV04H1Y4W             G-TGCATCCGCACCATATTCAG-CAAAATTAAC-GCCGTCAAAATCCAGAACAGAACG-CTTTTCC---ACGAAGACGTGA-TAAAGTAAA-
FXAWNEV04I3N1N             G-TGCATCCGCACCATATTCAG-CAAAATTAAC-GCCGTCAAAATCCAGAACAGAACG-CTTTTCC---ACGAAGACGTGA-TAAAGTAAA-
FXAWNEV04JGX6Z             G-TGCATCCGCACCATATTCAG-CAAAATTAAC-GCCGTCAAAATCCAGAACAGAACG-CTTTTCC---ACGAAGACGTGA-TAAAGTAAA-
FXAWNEV04JFAI6             G-TGCATCCGCACCATATTCAG-CAAAATTAAC-GCCGTCAAAATCCAGAACAGAACG-CTTT-CC---ACGAAGACGTGA-TAAAGTAAA-
FXAWNEV04IKVWV                                                  GTCAAAATCCAGAACAGAACG-CTTTTCC---ACGAAGACGTGA-TAAAGTAAA-
FJ4Q80U01C00IL                                                    CAAAATCCAGAACAGAACG-CTTTTCC---ACGAAGACGTGA-TAAAGTAAA-
FJ4Q80U01BKBZ3_right                                               A---TCCAGAACAGAACG-CTTTTCC---ACGAAGACGTGA-TAAAGTAAA-
FXAWNEV04H23A0                                                                 AGAACG-CTTTTCC---ACGAAGACGTGA-TAAAGTAAA-
FJ4Q80U01CICX6_right                                                            GAACGTCTTTTCC-A-ACGAAGACGTGAATAAAGTAAA-
FJ4Q80U01B86D4_right                                                                   TTTTCC--TACGAAGACGTGA-TAAAGTAAA-
FJ4Q80U01DRF0Y_right                                                                                  ACGTGA-TAAAGTAAA-
FJ4Q80U01DOG05_right                                                                                   CGTGA-TAAAGTAAA-
FJ4Q80U01B4QTU                                                                                                AA-GTAAA-
FJ4Q80U01EMCQ6                                                                                                A--GTAAA-
```

```
Info    P_Assembly_Ecoli_ShotgunPE :   -> 3 scaffolds, 92 large contigs
Info    P_Assembly_Ecoli_ShotgunPE : Computing signals...
Info    P_Assembly_Ecoli_ShotgunPE :   -> 5014191 of 5014191...
Info    P_Assembly_Ecoli_ShotgunPE :   Checkpointing...
Info    P_Assembly_Ecoli_ShotgunPE : Generating output...
Info    P_Assembly_Ecoli_ShotgunPE :   -> 5014191 of 5014191...
Info    P_Assembly_Ecoli_ShotgunPE : Assembly computation succeeded at: Thu Apr 7 20:27:54 2011
```

Figure 21: gsAssembler Alignment Results Tab.

The left panel of the Alignment Results tab contains a list of the contigs produced by the assembly project. By default, the first contig on the list will be automatically selected.

The multi-alignment panel has the following features:

- The larger area, on white background, contains the multiple alignment proper
- The selected contig consensus sequence is shown on top of the panel, gapped for insertions in the aligned reads
- The reads underlying the contig's multiple alignment, gapped and showing bases that diverge from the consensus as red dashes on yellow background
- The names of the contig and reads in the multiple alignment appear in a column, on the left side of the multiple alignment area
- Scroll bars allow you to navigate the alignment manually:
- Clicking on the arrowheads on either side of a scroll bar scrolls the alignment by one base (horizontal) or one read (vertical) in the direction of the arrowhead
- Clicking on the light-colored area on either side of the horizontal scroll bar "handle" scrolls the alignment by 40 bases is the corresponding direction
- Clicking and dragging the "handle" of a scroll bar allows you to move larger distances rapidly
- Right clicking on a GS read will produce a "Get flowgram for … at selected position" menu item which, if selected, will activate the Flowgrams tab and display the flowgram for the read; centered on the flow corresponding to the base on which the user clicked to activate the option.

> This capability is not active for FASTA/FASTQ reads or the contig/consensus sequences themselves, for which no flowgrams are available.

- Above that area are some informational and navigational controls:
  - **Go To**: A data entry field and a **Go To** button allow you to navigate directly to a position of interest: enter a number not larger than the length of the selected contig in the data entry field, and click the **Go To** button.
  - A '**zoom to selected'** button can be used to zoom in, centering on the selected base. If no base is selected, it will use the base in the center of the current view as a default.
  - The **zoom-slider** tool allows for more or fewer bases to be viewed at a time in the main window. If the slider is set to all the way zoom out (-) then the individual bases are grayed out but the shape of the read alignment is still shown along with the positions of the differences highlighted in yellow (Figure 22).
  - **Camera icon**: saves an image of the part of the multiple alignment table currently visible on screen, to a file in .png format.

- The Alignment Results tab also features a “Mouse Tracker” area, in the left panel, below the list of contig sequences. If you pause the mouse pointer over a base in the multi-alignment, the following information is provided about that base:
    - **Base**: the position of that base relative to the selected contig sequence; gaps in the contig sequence are displayed as the following base of the contig
    - **Read**: the name of the read or contig to which this base belongs
    - **Val**: the “value” of the base under the mouse pointer, in that read, *i.e.* A, G, T, C

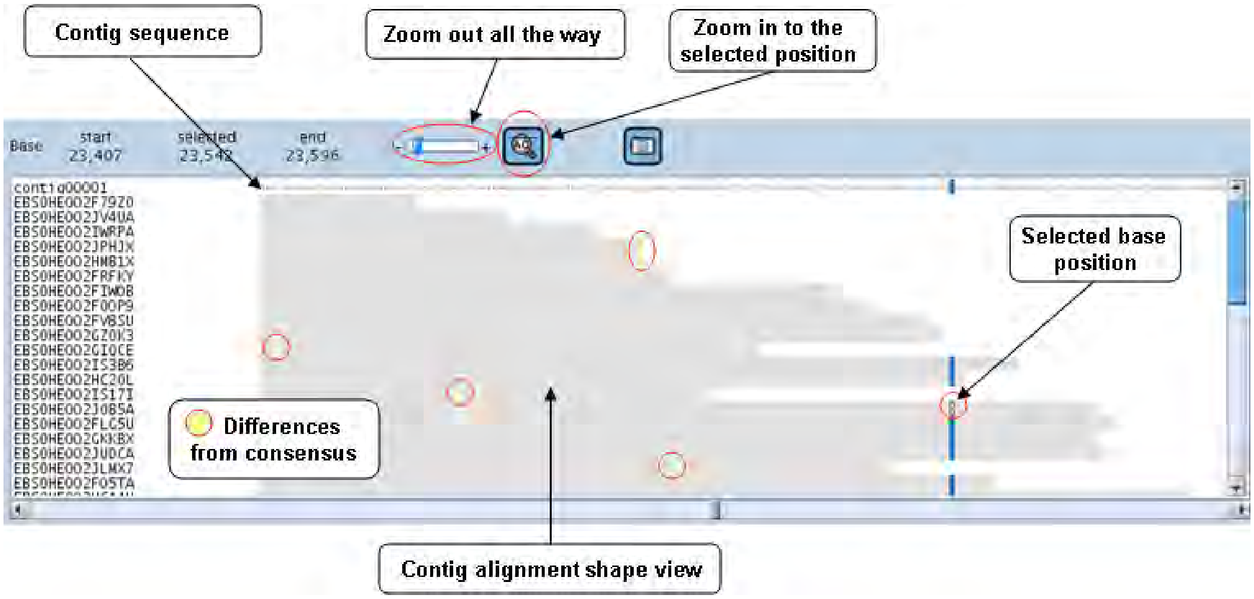
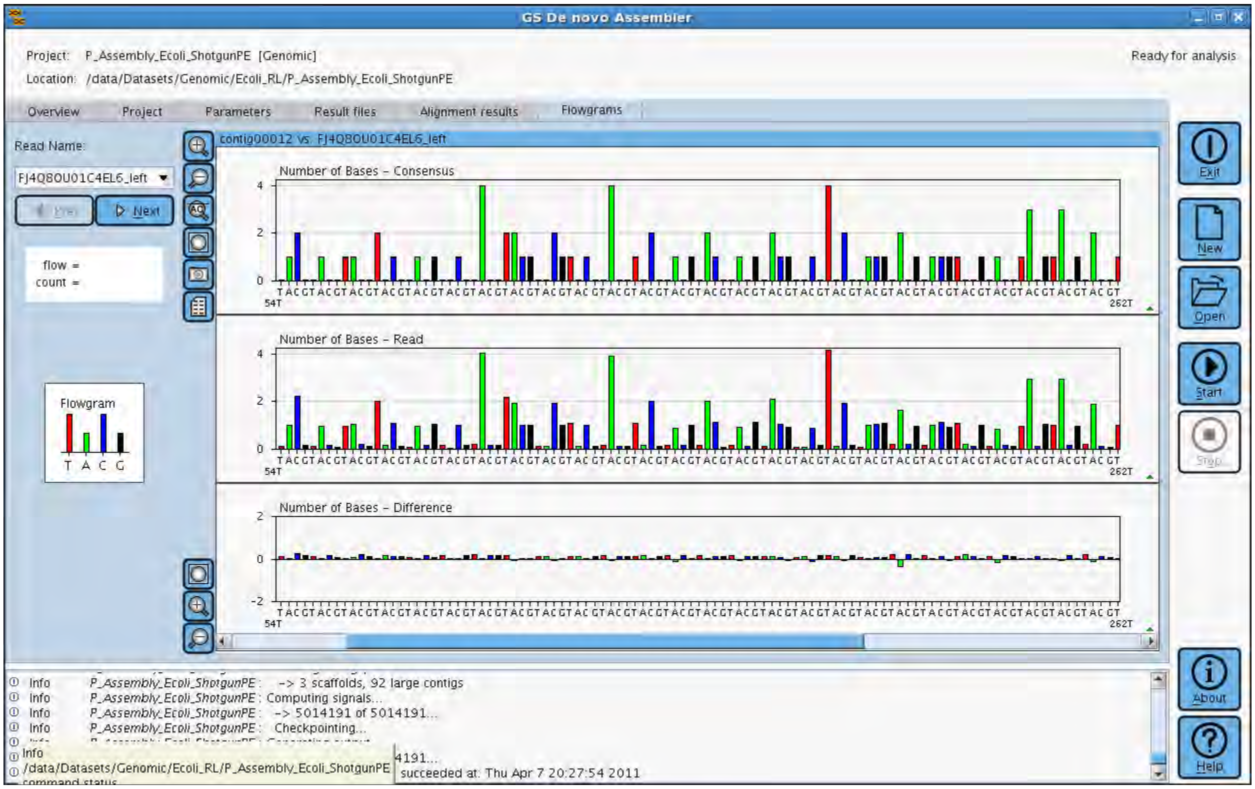


Figure 22: gsAssembler Alignment Results tab with the zoom slider all the way out.

1.11 The Flowgrams Tab

When viewing a multiple alignment display in the Alignment Results tab, any read in the multiple alignment can be selected in order to display its flowgram. Select the desired read, and then right-click on any base in the read of interest to open a contextual menu that contains a single item ("Get flowgram:.."; Figure 21). Selecting that item will display the flowgram for this read in the Flowgram tab (Figure 23). For additional information about using the **Flowgrams Tab**, see section 2.13.


GS De novo Assembler
Overview
Project
Parameters
Result files
Alignment results
Flowgrams
Read Name:
Prev
Next
Flowgram
T A C G
Exit
New
Open
Start
Stop
About
Help
contig00012 vs. FJ4Q8OU01C4EL6_left
Number of Bases – Consensus
Number of Bases – Read
Number of Bases – Difference
4
2
0
-2
TACGTACGTACGTACGTACGTACGTACGTACGTACGTACGTACGTACGTACGTACGTACGTACGTACGTACGTACGTACGTACGTACGTACGTACGTACGTACGT
54T
262T

Project: P_Assembly_Ecoli_ShotgunPE [Genomic]
Location: /data/Datasets/Genomic/Ecoli_RL/P_Assembly_Ecoli_ShotgunPE
Ready for analysis

Read Name: FJ4Q8OU01C4EL6_left

flow =
count =

```
Info    P_Assembly_Ecoli_ShotgunPE :    -> 3 scaffolds, 92 large contigs
Info    P_Assembly_Ecoli_ShotgunPE : Computing signals...
Info    P_Assembly_Ecoli_ShotgunPE :   -> 5014191 of 5014191...
Info    P_Assembly_Ecoli_ShotgunPE :   Checkpointing...
Info
/data/Datasets/Genomic/Ecoli_RL/P_Assembly_Ecoli_ShotgunPE    4191...
command status                                                 succeeded at: Thu Apr 7 20:27:54 2011
```

Figure 23: gsAssembler Flowgrams Tab.

1.12 Project Error Indicators

To assist the user in the proper setup of assembly, expansion of the error messaging is provided on the Parameters and Project tabs. Some examples are shown in section 4.12.

When a project is first created and no read files have been added yet, a warning icon on the Project Tab header will be displayed along with the status message 'Not ready for Analysis'.

1.13 Multiplex GS *De Novo* Assembler Projects

Multiplex Identifiers (MIDs) allow you to design an experiment whereby multiple libraries are prepared using distinct MID tags and sequenced together on the same PTP device (see Section 4.5). In an ordinary GS *De Novo* Assembler project, analysis can be restricted to a subset of reads that have been tagged with one or more specific MIDs (see Section 1.6.3), but a separate project must be created for each such set of MIDs. Moreover, although each ordinary MID project can be viewed separately, it is not possible to view summary statistics across all of the projects.

A multiplex GS *De Novo* Assembler project consists of a master project ( ) and a group of related ordinary projects called sample projects ( ). A sample is a named association of MIDs with read files that represents a subset of reads that should be analyzed separately, for example each in a series of microbial genomes to be assembled. The master project defines sample associations and project parameters during project setup. After computation, it provides summary read statistics across samples as well as drill-down access to the underlying results and reports for each of the individual samples.

The primary advantage of a multiplex assembly project over an ordinary project with MIDs is the substantial time savings in project setup across multiple samples. The primary restriction is that all of the sample projects must be computed using common project parameters.

The GS *De Novo* Assembler GUI is modified when a multiplex master project is created or opened. Similar to an ordinary assembly project, a multiplex master project has Overview, Project, Parameters and Result files tabs. However, the detail data displayed on the Alignment Results and Flowgrams tabs of an ordinary project are accessed in a multiplex master project by drilling down to individual sample projects in a hierarchical tree view on the Project tab. A new tab called Summary results displays unified read data statistics across all samples. Each individual sample project can also be opened as an ordinary GS *De Novo* Assembler project in order to view all of the underlying detail.

A multiplex GS *De Novo* Assembler project is set up and computed with only a few differences from an ordinary project (see Sections 1.4 through 1.8 for details on setting up an ordinary assembly project). Most of the differences in project setup are found on the Project tab, where samples are defined by the association of sample names with MID sequences and read files. In addition to the results generated for an ordinary project (see Sections 1.9 through 1.11 for details), a multiplex master project displays a unified summary of results across samples.

Multiplex projects were designed primarily for mapping projects, but still have considerable utility for assembly projects. For details on setting up and interpreting multiplex projects, see Section 2.15 for examples based on multiplex mapping projects.

1.14 The GS *De Novo* Assembler Command Line Interface

The GS *De Novo* Assembler CLI commands for one-step assembly of reads can be launched using the following commands:

```
runAssembly [options] [filedesc]
```

For incremental assembly *via* the CLI, a project directory structure is created to organize the files of the incremental build of the project data. The following commands are used for incremental assembly projects;

```
newAssembly [options: -cdna, -multi] projDir

addRun [options: -p, -lib, -mcf] [projDir] filedesc

removeRun [projDir] filedesc

runProject [options] [projDir]
```

For all of these commands;

- The arguments in brackets [ ] are optional.
- 'options' refers to the specific command options available.
- 'filedesc' is one of the following:
  - an sff filename
  - a [regionlist:]datadir
  - a read FASTA file
- any of the above prepended by an [MIDList@] specification where each "MIDList" is a multiplexing information string used to filter the set of file reads to be used in the assembly (If MIDs were used in the generation of the read data file, then an MIDList string must be specified in order for the assembler to properly handle the file's reads.) For details about using MIDs, see section 4.5.
- 'projDir' is the path of the data analysis project directory.

Some of the command line options for Assembly are mutually exclusive, for obvious functional reasons. See section 4.3 for a list of these options.

1.14.1 Working with Project Folders and Data Files

Since the assembly computation is often performed on a pool of sequencing runs (or Read Data files) rather than on any single run, the result files it generates are not deposited in a run folder. Two general cases exist.

- If the assembly is performed using the "one-step" command runAssembly, a folder with a 'P_' prefix (for 'P'roject folder) is created in the user's current working directory on the datarig at the time the application is launched, or written to a directory specified by the user on the command line, to contain these files. The name structure for this folder is as follows:

  P_yyyy_mm_dd_hh_min_sec_runAssembly

- For "incremental", or "project-based" assembly, using the GUI application or using the newAssembly and related commands, the output is placed in a "project" folder. A user can specify any name for a project folder; it will be recognized as a project folder by virtue of the "454Project.xml" file that will be automatically created within it. If a directory name is not specified on the newAssembly command line, the software will use the same default name as the runAssembly command (as above).

The incremental assembly folder contains additional folders and files that mark the folder as a project folder and that store configuration information and internal data for the GS *De Novo* Assembler application. A project folder is comprised of two sub-folders: an 'assembly' sub-folder which contains the project state and output files; and an 'sff' sub-folder containing the copies and/or symbolic links for the SFF files used as input to the project. The 454Project.xml file identifies the folder as a 454 project folder.

- The "-o" option can be specified on the command line to change the directory where the output files should be written. If the specified directory exists, the output files will be written to that directory. If the specified directory does not exist, the program will create it, if possible.

> When using the –o option with the runAssembly or newAssembly commands, the assembler will not overwrite the contents of the specified project directory if any exist. To force an overwrite of an existing directory, use the –force option with the runAssembly or newAssembly commands.

If a project or any of the files it uses need to be moved from one location to another, the project's configuration file (454AssemblyProject.xml) must be modified to reflect the new location. The changeRun command facilitates this process.

External Files

- GS Read Data files (SFF files) are not actually copied to the project directory. Rather, a symbolic link to the file is created and placed in the project's sff directory. Consequently, if the original file is moved or erased from the file system, the project will not operate correctly unless changeRun is used. This applies whether the files are added *via* the GUI or the command line.
- FASTA/FASTQ files are not actually copied to the project directory. Rather, the file path is simply recorded in the project setup (no symbolic link to the file is created). Consequently, if the original file is moved or erased from the file system, the project will not operate correctly unless changeRun is used. This applies to FASTA/FASTQ read files, whether added *via* the GUI or the command line:
  - FASTA/FASTQ reads files, including Sanger reads
- The changeRun command does <u>not</u> relink input files specified on the Input sub-tab of the GUI or using the command line. These files must be manually specified to complete the relinking process.
  - Trimming database file
  - Screening database file
  - Exclude filter file
  - Include filter file

1.14.1.1 The changeRun Command

The changeRun command can be used in either of two modes. It can be used to change the way reads are handled in an assembly, forcing reads in one or more files to be handled either as paired-end reads or non-paired reads. It can also be used to ensure the integrity of read file information in a project's configuration if the project or any of the read files it uses is moved. To change the handling of one or more read files, the changeRun command's syntax is

```
changeRun {-p | -np} projectDir readFile1, readFile2…
```

⚠ Warning: if you use changeRun to treat a Paired-End SFF file as a non-paired end file, the linkers will remain in the reads.

If you move a project to another computer or the read files are moved or no longer accessible by the project, changeRun can be used to inform the project of the new location of the reads:

```
changeRun [-verbose [-verbose]] [-relink] projectDir [sourcePath
destinationPath]
```

If you move a project to another computer or the read files (SFF, FASTA, FASTQ) are moved or no longer accessible by the project, changeRun can be used to inform the project of the new location of the reads. The changeRun command provides information about the project's current configuration at two levels of detail, using the –verbose option. It also allows you to modify the project's read file location(s) to point to valid file(s) using the -relink option. These two options can be used together in a variety of useful ways.

```
changeRun [-verbose [-verbose]] [-relink] projectDir [sourcePath
destinationPath]
```

Used in the absence of the –relink option, `changeRun –verbose` lists the read file(s) in the project's configuration and whether each file can be accessed, but without making any changes. Using `changeRun -verbose -verbose` expands the listing to include the read sequence accnos.

```
changeRun -verbose [-verbose] projectDir
```

Used in the absence of the –verbose option, `changeRun –relink` will change paths for all read files found in the sourcePath (if sourcePath exists in project's metadata) which are also found in the destinationPath location. The contents of the source/destination files are also compared to prevent relinking to incorrect (*i.e.* wrongly (re)named) read files.

Source and destination path names should not include file names. Also, wildcard characters are not allowed in the path names. The path arguments should be specified as they appear in the project's XML (configuration) file (*i.e.* full path to read file's physical location should be used).

```
changeRun –relink projectDir sourcePath destinationPath
```

Before actually changing a project's configuration, it is useful to assess the changes that need to be made. Used in combination with –verbose –verbose, -relink will test the "relinking process" before actually committing to it. The project's configuration isn't actually changed, but the integrity of each specified relink is reported.

Once all relink conflicts can be resolved, the –relink option can be used to perform the actual relinking. This can be done with or without feedback about the results (use of a single "–verbose" option provides feedback):

```
changeRun [-verbose]–relink projectDir sourcePath destinationPath
```

If a project using –consed (consed-compatible) output is moved, it will be necessary to recreate the sff_dir link to the sff folder. For a Linux system…

```
rm sff_dir
ln –s projSffDir sff_dir
```

1.14.2 One-Step Assembly: the runAssembly Command

If all the reads to be assembled are available at once, the GS *De Novo* Assembler application can process them with a single command, which has the following command line structure:

```
runAssembly [options] [MIDList@]filedesc…
```

…where:

- "[options]" are zero or more of the command line options (see sections 4.2),
- each "filedesc" is one of the following :
    - sfffilename or
    - [regionlist:]datadir or
    - readfastafile
- and each "MIDList" is a multiplexing information string used to filter the set of file reads to be used in the assembly (see section 4.5 for the format of the MIDList information). If MIDs were used in the generation of the data file, then a MIDList string must be specified in order for the assembler to properly handle the file's reads.

The runAssembly command is actually a wrapper program around the newAssembly and related commands used in incremental assembly (see section 1.14.3). After the incremental assembly command is completed, runAssembly then transforms the project files and folders into this one-step form (which more closely matches the output structure of older versions of the "Assembly" application). The actions taken are:

- All the assembly output files are moved up from the assembly sub-directory into the main folder
- All internal data files in the assembly sub-directory are deleted
- The 454AssemblyProject.xml and 454Project.xml files are deleted
- The assembly sub-directory is deleted

If the –**nrm** option is given, runAssembly does not perform this transformation, and the resulting folder can be used for further project-based assembly (the structure of the files/folders will match that of an incremental assembly project).

For data generated on the GS FLX+ system, the path given for any of the data directories may be prepended with an optional list of regions, separated from the path by a colon. For example:

```
1-3,4,6-8:D_yyyy_mm_dd_hh_min_sec_testuser_SignalProcessing
```

The list of regions must have the following format:

It must be a comma-separated list, ending with a colon.

- Each element of the list can either be a single region number or a dash-separated range of region numbers.
- Duplicate regions may be specified, and the regions may be specified in any order, but duplicates will be removed and the regions will be processed and reported in numerical order.
- No spaces or other characters are allowed in the string.

If the region list is not present for a data directory path, all regions of the run for that directory will be used in the assembly.

Any combination of explicit SFF files and data directory paths (with optional region lists) may be specified on the command line. For each data directory path given, the runAssembly command will read the existing SFF files in the “sff” sub-directory of the data directory. If SFF files are not present in a data directory (*e.g.* for a run whose data has been processed with a version of the 454 Sequencing system software anterior to 1.0.52), the signal processing step of the GS Run Processor application must be rerun on the Data Processing folder.

The path for any filename or data directory name can also be prepended with a multiplexing information string. Section 4.5 gives further information on using MIDs.

The data analysis software installation package contains a default MID configuration file (see Glossary). This file is read by the GS *De Novo* Assembler and used to match MID set names and MID names with their multiplexing information. Users can edit this file to add their own MID sets (following the format and syntax described in the file), or can copy this file to create their own separate MID configuration file (and then use the “-mcf” option to specify that as the MID configuration file to be used).

1.14.3 Incremental Assembly: the newAssembly and Related Commands

This section describes the main commands of the GS *De Novo* Assembler application, to be used when one or more runs are to be assembled as part of an assembly project. These commands allow you to add and remove runs over time and incrementally update the assembly. With these commands, the execution of the assembly algorithms and generation of the results can be controlled by command line options and configuration parameters (paralleling the equivalent controls in the GUI application). Such incremental assembly can be useful when, for example, you want to see intermediate results on existing sequencing runs to determine if you need to carry out further runs to reach a desired depth of coverage, or just to monitor the project. Incremental assembly is also useful if you simply wish to create output using different output parameter settings.

Four commands are used in an assembly project: newAssembly, addRun, removeRun and runProject. These are described in the sub-sections below.

1.14.3.1 The newAssembly Command

The newAssembly command is used to initiate an assembly project and set an assembly project folder to contain the project data (see Section 1.14.1). Its command line structure is:

```
newAssembly [option] [dir]
```

…where:

- `[option]` -cdna and –multi may be used; see Section 4.2
- `[dir]` is the optional name of the project directory

1.14.3.2 The addRun Command

The addRun command is used to add Read Data sets to existing assembly projects. Its command line structure is:

```
addRun [options] [projDir] [MIDList@]filedesc…
```

…where:

- "`[options]`" are zero or more of the command line options described in section 4.2
- each "`filedesc`" is one of the following:
  - ○ sfffilename or
  - ○ [regionlist:]datadir or
  - ○ readfastafile
- and each "MIDList" is a list of multiplexing information used to filter the set of file reads used in the assembly (see section 4.5 for the format of the MIDList information). If MIDs were used in the generation of the data file, then an MIDList string must be specified in order for the GS Reference *De Novo* Assembler to properly handle the file's reads.

- **Input read size constraints:** The reads input for the assembly computation must be shorter than 2000 bases per read (longer sequences are ignored) and longer than 50 bases (shorter sequences are ignored). When paired end 454 Reads (SFF reads) are part of the project, reads with lengths between the value of the minlen parameter and 50 bp will be mapped onto contigs formed by assembling longer reads during a later stage in the assembly.
- **Order of addition of Read data may affect assembly results:** In general, the reads should be added to an incremental assembly in the following order to achieve the best contigging and scaffolding results:
  - Shotgun
  - 3 kb Paired End
  - 8 kb Paired End
  - 20 kb Paired End

- The addRun command can be executed multiple times for an assembly project (in any combination with the removeRun and runProject commands). When addRun is executed, it adds (not "resets") the given runs/regions or SFF files to the list of sequencing data used in the project, *i.e.* it does not reset the list of sequence data. The reads used in the assembly (runProject, see section 1.14.3.4 below) are the union of the data from all executions of addRun for the project.
- The commands addRun, removeRun, and runProject can be executed in any combination and in any order, in an assembly project.

1.14.3.3 The removeRun Command

The removeRun command is used to remove Read Data sets from existing assembly projects. Its command line structure is:

```
removeRun [dir] (sffname or readfastafilepath)…
```

- This command is more conveniently carried out from the GUI application. When called from the command line, it requires the SFF file name(s) *given in the project sff sub-directory* or the FASTA/FASTQ file name(s) *given in the project configuration file 454AssemblyProject.xml*. These names may not match the original names of the corresponding files or may not be known to the user, especially if the software had to rename any of them to ensure name uniqueness. (The filenames of the SFF files in the sff sub-directory are assigned by the GS *De Novo* Assembler application to ensure uniqueness for all the files, while trying to preserve the original names when possible.)
- The execution of this command does not physically remove the file(s) from the project sff sub-directory or from any existing assembly. The file(s) and the reads they contain are only marked for removal by the removeRun command, and are actually removed only the next time the project is computed (*via* the runProject command).
- The commands addRun, removeRun, and runProject can be executed in any combination and in any order, in an assembly project.

1.14.3.4 The runProject Command

The runProject command performs the actual assembly computation for a project and generates the results of the assembly. Its command line structure is:

```
runProject [options] [projDir]
```

1.14.4 Multiplex Assembly Project Setup

A multiplex one-step assembly project is set up in exactly the same way as an ordinary one-step assembly project, except that–multi and –tsv options are used with runAssembly, and read data sets cannot be filtered by pre-pending an MID list to the filename. If read files have been specified within the samples.tsv file, any read files passed as an argument to runAssembly will be ignored.

```
runAssembly [multiplex options: -batchsize -mrerun] –multi –tsv samples.tsv
filedesc
```

There are two options specific to computing a multiplex project, including –batchsize (number of sample projects to be computed simultaneously) and -mrerun (controlling project re-computation).

A multiplex incremental assembly project is set up in exactly the same way as an ordinary incremental assembly project, except that the –multi option is used with the newAssembly command, and read data sets are added using the addSample command (Section 2.16.5.1) rather than the addRun command. The runProject command uses the same multiplex-specific computing options as runAssembly.

```
newAssembly [options: -cdna (-cref | -gref)] –multi projDir

addSample [multiplex options: -clear –datapath] –tsv samples.tsv projDir
[[MIDList@]filedesc]

runProject [multiplex options: -batchsize -mrerun] [projDir]
```

For all of these commands;

- The arguments in square brackets [ ] are optional.
- The '|' symbol means 'OR'.
- 'options' refers to the optional command arguments available.
- 'projDir' is the path of the data analysis project directory.
- 'filedesc' = (sfffile | fnafile | [regionlist:]analysisDir), referring to a read data file in one of the following formats:
  - an sfffilename
  - a [regionlist:]datadir
  - a readfastafile
  - a space-separated list of any of the above
- '[MIDList@]filedesc', where each 'MIDList' is a multiplexing information string used to filter the set of file reads to be used in the assembly (if MIDs were used in the generation of the read data file, then an MIDList string must be specified in order for the assembler to properly handle the file's reads.) For details about using MIDs, see section 4.5.

1.15 GS *De Novo* Assembler cDNA / Transcriptome Options

These descriptions are for options specific to cDNA / transcriptome assembly projects:

Option	Description
-cdna	cDNA / Transcriptome assembly - To specify a cDNA / transcriptome assembly project, use the -cdna option with the runAssembly or newAssembly commands.
-ig	Isogroup Threshold - Specifies the maximum number of contigs in an isogroup. If an isogroup has more contigs than this threshold, it will not be traversed for isotigs and its contigs will appear as contigs in the output files.
-it	Isotig Threshold - Specifies the maximum number of isotigs in an isogroup. When this threshold is reached, the isogroup traversal will stop and all of its contigs will appear as contigs in the output files.
-icc	Isotig Contig Count Threshold - Specifies the maximum number of contigs in one isotig. When this threshold is reached, the further traversal of a particular isotig in an isogroup will be stopped and the isotig's contigs might or might not appear as contigs in the output files, depending on whether that particular contig is part of another isotig in an isogroup; only the contigs which are not part of any isotig will be reported as contigs in the output files.
-icl	Isotig Contig Length Threshold - If any contig length is shorter than this threshold, the further traversal of a particular isotig in an isogroup will be stopped. The contig shorter than the icl threshold will be marked as such and reported in the output files. Other contigs from such an isotig might or might not appear as contigs in the output files, depending on whether each particular contig is part of another isotig in an isogroup; only the contigs which are not part of any isotig will be reported as contigs in the output files. Rarely, a contig with a reported length below this threshold will be traversed in an isotig. This is because the contig length estimated prior to final basecalling is the length tested against the –icl threshold. The reported length of the contig may change slightly when basecalling occurs. It is also possible for isotig traversal to stop at a contig whose reported length is above the –icl threshold.
-isplit	Initiate isotig traversal when depth spikes in alignments are found. These spikes will be treated as putative splice events so that traversal along the path will continue producing more isotigs.

For cDNA assembly, the “-a” is set to 0 regardless of the value that may be entered on the command-line.

If any of the above-mentioned thresholds are set to **zero**, it will not be taken into account during the computation. In such cases, some implicit thresholds and limits may still be applied as described below. Default thresholds for the computation are:

- ig = 500 contigs
- it = 100 isotigs
- icc = 100 contigs
- icl = 3 base pairs

Implicit thresholds and limits for the computation (imposed due to the recursive nature of traversal algorithm) are:

- Maximum recursion depth during isotig path traversal is **200**. This corresponds to the number of contigs in the isotig, *i.e.* the -icc threshold.
- Maximum isotig count during graph traversal is **10000**. This corresponds to the number of isotigs in the isogroup, *i.e.* the -it threshold.
- Cyclic graphs: recursive isotig path traversal will stop if cyclic isotig structures are detected, *i.e.* revisiting one contig which has already been part of the isotig being traversed. Such cyclic structures will be marked in the output files by assigning cyclic status for the first contig detected, and the cyclic isotig's contigs might or might not appear as contigs in the output files, depending on whether each particular contig is part of another non-cyclic isotig in an isogroup.
- Edge Threshold: is used when an edge's depth is less than 5% of the maximum of the "end" column depths of the two chords that the edge connects. "End" column means the alignment column of the end of the chord connected to the edge. Whenever an edge depth is less that this 5% threshold, isotig traversal ends and the chord that the edge leads to is left as a contig. These contigs are never removed from the isogroup.

The Large or complex genome option should not be used in cDNA assembly projects, and use of the -e and –het options is not recommended.

1.16 GS *De Novo* Assembler Output

1.16.1 Output File Specification

Output Files produced by the GS *De Novo* Assembler are described briefly below. GUI settings and command line options that control the content type of a file are given when applicable.

File name	Description	Assembly – Genomic Project	Assembly – cDNA Project	GUI Conditional output option	CLI Conditional output option
454AlignmentInfo.tsv	Tab-delimited file giving position-by-position consensus base and flow signal information. Output conditionally (using –info/-noinfo options or checkbox selection on GUI Parameters Tab Output Sub-tab) if there are more than 4M reads or the total length of assembled contigs exceeds 40Mbp.	X	X	Alignment info: Output Output small No output	-info (default) -noinfo
454AllContigs.fna	FASTA file of all the consensus basecalled contigs longer than 100 bases. The minimum length output can be changed by using the [-a #] option in the CLI or changing the All contig threshold in the GUI Parameters Tab, Output Sub-tab. Some contigs slightly shorter than this may be included due to the two-phase nature by which contig consensi are determined.	X	n/a	Parameters: All contig threshold	-a #
454AllContigs.qual	Corresponding Phred-equivalent quality scores for each base in the consensus contigs in 454AllContigs.fna.	X	X		
454Contigs.ace	ACE format file that can be loaded by third-party viewer programs that support the ACE format. The output can be a single file for the entire project or a folder containing individual files for each contig in the assembly.	X		ACE Format selection	-nobig, -ace, -consed, -noace, -ar, -at, -ad
454ContigGraph.txt	A text file giving the “contig graph” that describes the branching structure between contigs.	X	X		

File name	Description	Assembly – Genomic Project	Assembly – cDNA Project	GUI Conditional output option	CLI Conditional output option
454ContigScaffolds.txt	The 454ContigScaffolds.txt file is generated only with the –scaffold option in the CLI or checking "Output scaffolds" on the Output Parameters sub-tab. It is an AGP formatted file that gives you information about your contigs that are a part of a specific scaffold. Column 1 is the scaffold ID and Column 6 is the contig ID from the allContigs file. Column 5 is either a W (sequence) or N (gap). Some of the contig ID's will be found multiple times in different scaffolds indicating a repeat contig.	X			
454Isotigs.ace	ACE format file that can be loaded by third-party viewer programs that support the ACE format. The output can be a single file for the entire project or a folder containing individual files for each isotig and large contig in the assembly (only large contigs that aren't part of any isotigs appear as a separate entry in the ACE file).		X	ACE Format selection	-nobig -ace/acedir -consed -noace -consed16
454Isotigs.fna	FASTA file of all the isotig sequences traversed from the multiple alignment graph structure (*i.e.* from the isogroup) along with any large (longer than the Large contig Threshold number of bases) contigs that are not part of any isotig.		X		
454Isotigs.qual	Corresponding Phred-equivalent quality scores for each base in the isotigs and contigs in 454Isotigs.fna.		X		
454Isotigs.txt	An AGP file (NCBI's format for describing scaffolds of contigs) containing information describing the path through the contigs taken by each isotig in the isogroup		X		
454IsotigLayout.txt	A text file that gives a visual representation of how the contigs are laid along each isotig in the isogroup.		X		
454Isotigs.faa	FASTA file of all ORFs of 100 bp or more found in each isotig nucleotide sequence.		X		

File name	Description	Assembly – Genomic Project	Assembly – cDNA Project	GUI Conditional output option	CLI Conditional output option
454IsotigOrfAlign.txt	A text file containing nucleotide sequences and amino acid sequences for each ORF found in the isotigs,		X		
454LargeContigs.fna	FASTA file of all the "large" consensus basecalled contigs contained in 454AllContigs.fna (>500bp). The minimum length can be changed by using the [-l #] option in the CLI or changing the Large contig threshold in the GUI Parameters Tab, Output Sub-tab.	X		Parameter: Large contig threshold	-l #
454LargeContigs.qual	Corresponding Phred-equivalent quality scores for each base in the "large" consensus contigs in 454LargeContigs.fna.	X			
454NewblerMetrics.txt	File providing various assembly metrics, including the number of input runs and reads, the number and size of the large consensus contigs as well as all consensus contigs.	X	X		
454NewblerProgress.txt	A text log of the messages sent to standard output during the assembly computation.	X	X		
454PairAlign.txt	A text file giving the pairwise alignment(s) of the overlaps used in the assembly computation (only produced when using the –pair option [or –pairt option for the tab-delimited version of the file]).	X		Pairwise alignment selection	-pair, -pairt, -nobig
454PairStatus.txt	Tab-delimited text file providing a per-pair report of the location and status of how each paired end pair of reads were used in the assembly.	X			
454ReadStatus.txt	Tab-delimited text file providing a per-read report of the status of each read in the assembly. The 3' and 5' positions of each read's alignment within the contig are also reported.	X	X		-nobig

File name	Description	Assembly – Genomic Project	Assembly – cDNA Project	GUI Conditional output option	CLI Conditional output option
454ScaffoldContigs.fna	FASTA file that contains sequences without any gaps. This file is generated only with the –scaffold option specified in the CLI or checking "Output scaffolds" on the Output Parameters sub-tab. The ScaffoldContigs are the stretches of contiguous sequence within the scaffolds. These ScaffoldContigs may span multiple contigs from the allContigs file. A ScaffoldContig comprises segments, each of which corresponds to a contig in the allContigs file. The FASTA header contains sctg_XXX_YYY, where XXX is the segment number and the YYY is the scaffold to which the scaffoldContig belongs.	X			-scaffold
454ScaffoldContigs.qual	Corresponding Phred-equivalent quality scores for each nucleotide in the scaffolded consensus contigs in 454ScaffoldContigs.fna.	X			-scaffold
454Scaffolds.fna	FASTA file of the concatenated contig sequences that were scaffolded as a result of paired end analysis. The contigs are separated by a number of 'N' corresponding to the estimated size of the gap between them (but with a minimum of 25 N's to ensure the separation of the contigs).	X			
454Scaffolds.qual	Corresponding Phred-equivalent quality scores for each nucleotide in the scaffolded consensus contigs in 454Scaffolds.fna.	X			
454Scaffolds.txt	An AGP file (NCBI's format for describing scaffolds of contigs) containing the scaffold layout.	X			
454TagPairAlign.txt	A text file showing the pairwise alignments of short tag reads, which are not part of the standard overlap computation, but are mapped to the consensi in a later computation step (Section 1.16.1.10).	X		Pairwise alignment selection	-pair, -pairt

File name	Description	Assembly – Genomic Project	Assembly – cDNA Project	GUI Conditional output option	CLI Conditional output option
454TrimmedReads.fna	FASTA file of the trimmed reads used in the assembly	X	X	Parameter: Output trimmed reads	-tr
454TrimmedReads.qual	Corresponding Phred-equivalent quality scores for each base in the reads in 454TrimmedReads.fna	X	X	Parameter: Output trimmed reads	-tr
454TrimStatus.txt	Tab-delimited text file providing a per-read report of the original and revised trim points used in the assembly.	X			-nobig

Table 1: GS *De Novo* Assembly Output Files.

454NewblerProgress.txt: If runs are added incrementally and multiple executions of runProject occur, the output messages are appended to this file. If the “Incremental *De Novo* assembly analysis” checkbox option is not selected in the GUI application, or the “-r” option is given on the runProject command line, the GS *De Novo* Assembler deletes intermediate data and “restarts” the assembly computation, and the NewblerProgress.txt file is deleted and restarted as well.

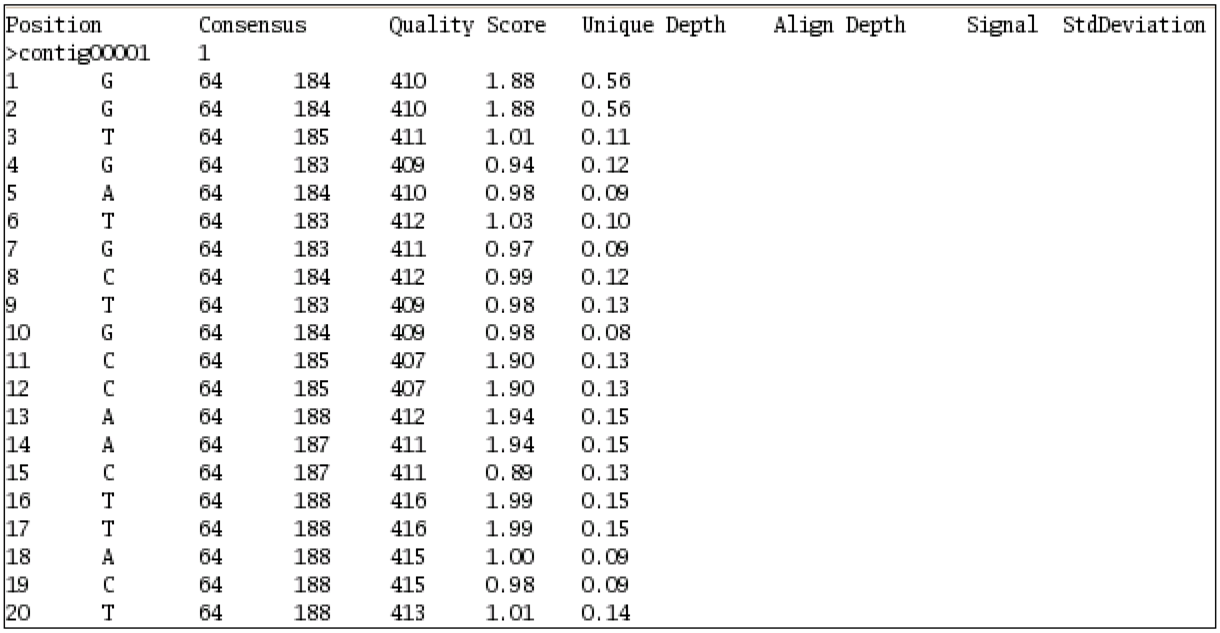
1.16.1.1 454AlignmentInfo.tsv

The 454AlignmentInfo.tsv file (Figure 24) contains position-by-position summary information about the consensus sequence for the contigs generated by the GS *De Novo* Assembler application, listed one nucleotide per line (in a tab-delimited format). Output conditionally (using the -info/-infoall/-noinfo options or the selection made on the GUI Parameters Tab Output Sub-tab). By default, this file is only output if there are fewer than 4 million input reads and the total length of assembled contigs is less than 40 Mbp. For larger projects, -info or -infoall or the corresponding GUI **Output** selection for **Alignment Info** must be used to generate this file.

The columns of each line contain the following information:

1. **Position** – the position in the contig
2. **Consensus** – the consensus nucleotide for that position in the contig
3. **Quality Score** – the quality score of the consensus base
4. **Unique Depth** – the number of non-duplicate reads that align at that position in the alignment
5. **Align Depth** – the number of reads (including duplicates) that align at that position in the alignment
6. **Signal** – the average signal of the read flowgrams, for the flows that correspond to that position in the alignment
7. **StdDeviation** – the standard deviation of the read flowgram signals at the corresponding flows

Prior to each region of lines for each contig, a header line beginning with a '>' displays the contig name. *e.g.*,



```
Position        Consensus       Quality Score   Unique Depth    Align Depth     Signal  StdDeviation
>contig00001    1
1       G       64      184     410     1.88    0.56
2       G       64      184     410     1.88    0.56
3       T       64      185     411     1.01    0.11
4       G       64      183     409     0.94    0.12
5       A       64      184     410     0.98    0.09
6       T       64      183     412     1.03    0.10
7       G       64      183     411     0.97    0.09
8       C       64      184     412     0.99    0.12
9       T       64      183     409     0.98    0.13
10      G       64      184     409     0.98    0.08
11      C       64      185     407     1.90    0.13
12      C       64      185     407     1.90    0.13
13      A       64      188     412     1.94    0.15
14      A       64      187     411     1.94    0.15
15      C       64      187     411     0.89    0.13
16      T       64      188     416     1.99    0.15
17      T       64      188     416     1.99    0.15
18      A       64      188     415     1.00    0.09
19      C       64      188     415     0.98    0.09
20      T       64      188     413     1.01    0.14
```

Figure 24: 454AlignmentInfo.tsv file portion example for an Assembly project.

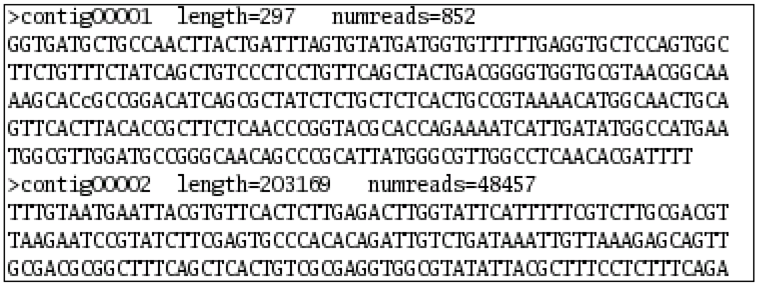
1.16.1.2 fna and qual files: 454AllContigs, 454LargeContigs, 454Scaffolds, 454ScaffoldContigs, 454TrimmedReads

These files contain the nucleotide sequences of all the contigs, large contigs, scaffolds, scaffoldContigs or trimmed reads (Figure 25) and associated nucleotide Quality Scores (Phred-equivalent; Figure 26) produced by the GS *De Novo* Assembler application. The AllContig and LargeContig minimum length thresholds are specified in the GUI or by CLI options described in Table 1, above.

The TrimmedReads output is generated by specifying the CLI option –tr or by checking the "Output trimmed reads" checkbox on the Parameters tab/Output sub-tab. Description lines for paired end entries will include the "library=", "template=", and "dir=" fields so they can be used as paired end input data for future projects.

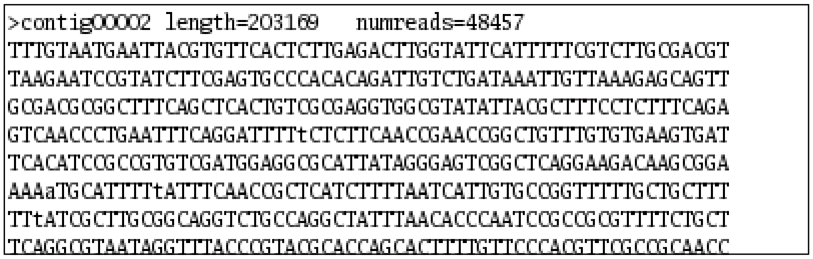
The scaffoldContigs output is generated by specifying the –scaffold option in the CLI or checking "Output scaffolds" on the Output Parameters sub-tab. ScaffoldContigs are the stretches of contiguous sequence within the scaffolds. These scaffoldContigs may span multiple contigs from the allContigs file.

A



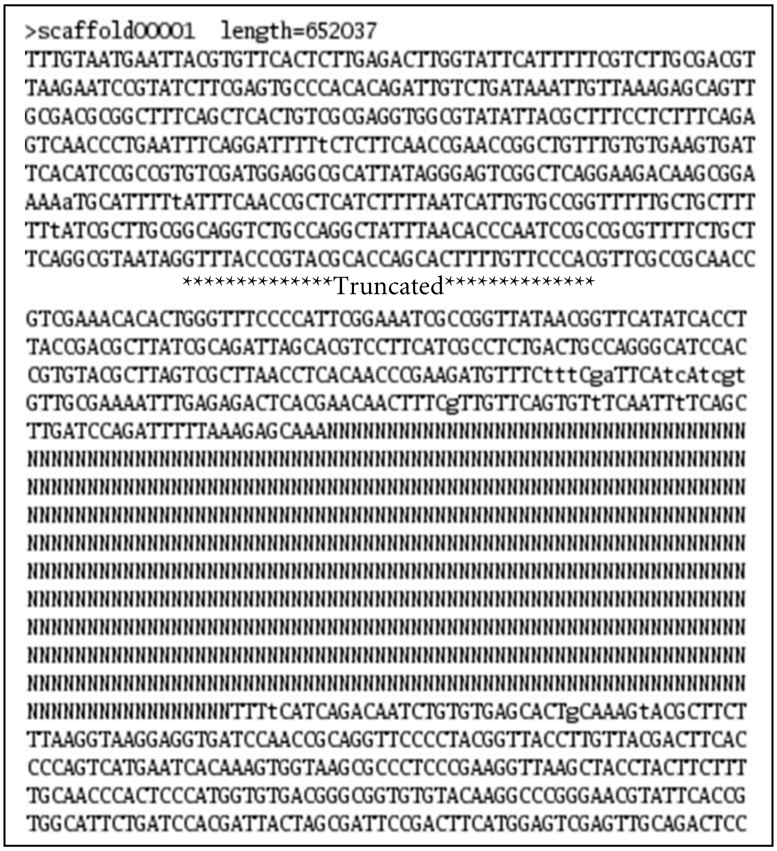
```
>contig00001  length=297   numreads=852
GGTGATGCTGCCAACTTACTGATTTAGTGTATGATGGTGTTTTTGAGGTGCTCCAGTGGC
TTCTGTTTCTATCAGCTGTCCCTCCTGTTCAGCTACTGACGGGGTGGTGCGTAACGGCAA
AAGCACcGCCGGACATCAGCGCTATCTCTGCTCTCACTGCCGTAAAACATGGCAACTGCA
GTTCACTTACACCGCTTCTCAACCCGGTACGCACCAGAAAATCATTGATATGGCCATGAA
TGGCGTTGGATGCCGGGCAACAGCCCGCATTATGGGCGTTGGCCTCAACACGATTTT
>contig00002  length=203169   numreads=48457
TTTGTAATGAATTACGTGTTCACTCTTGAGACTTGGTATTCATTTTTCGTCTTGCGACGT
TAAGAATCCGTATCTTCGAGTGCCCACACAGATTGTCTGATAAATTGTTAAAGAGCAGTT
GCGACGCGGCTTTCAGCTCACTGTCGCGAGGTGGCGTATATTACGCTTTCCTCTTTCAGA
```

B



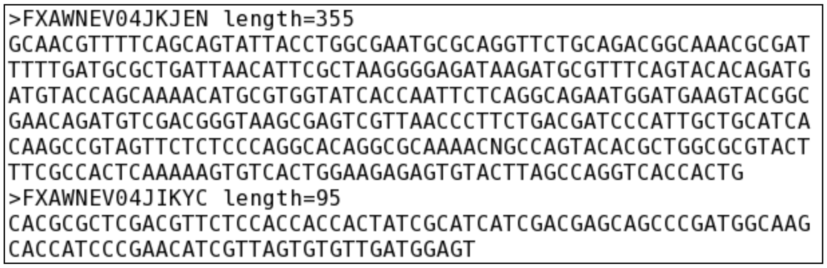
```
>contig00002 length=203169   numreads=48457
TTTGTAATGAATTACGTGTTCACTCTTGAGACTTGGTATTCATTTTTCGTCTTGCGACGT
TAAGAATCCGTATCTTCGAGTGCCCACACAGATTGTCTGATAAATTGTTAAAGAGCAGTT
GCGACGCGGCTTTCAGCTCACTGTCGCGAGGTGGCGTATATTACGCTTTCCTCTTTCAGA
GTCAACCCTGAATTTCAGGATTTTtCTCTTCAACCGAACCGGCTGTTTGTGTGAAGTGAT
TCACATCCGCCGTGTCGATGGAGGCGCATTATAGGGAGTCGGCTCAGGAAGACAAGCGGA
AAAaTGCATTTTtATTTCAACCGCTCATCTTTTAATCATTGTGCCGGTTTTTGCTGCTTT
TTtATCGCTTGCGGCAGGTCTGCCAGGCTATTTAACACCCAATCCGCCGCGTTTTCTGCT
TCAGGCGTAATAGGTTTACCCGTACGCACCAGCACTTTTGTTCCCACGTTCGCCGCAACC
```

C



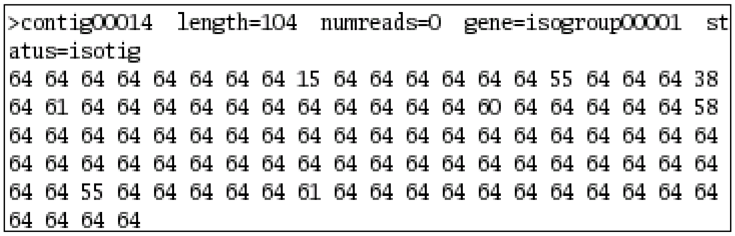
```
>scaffold00001  length=652037
TTTGTAATGAATTACGTGTTCACTCTTGAGACTTGGTATTCATTTTTCGTCTTGCGACGT
TAAGAATCCGTATCTTCGAGTGCCCACACAGATTGTCTGATAAATTGTTAAAGAGCAGTT
GCGACGCGGCTTTCAGCTCACTGTCGCGAGGTGGCGTATATTACGCTTTCCTCTTTCAGA
GTCAACCCTGAATTTCAGGATTTTtCTCTTCAACCGAACCGGCTGTTTGTGTGAAGTGAT
TCACATCCGCCGTGTCGATGGAGGCGCATTATAGGGAGTCGGCTCAGGAAGACAAGCGGA
AAAaTGCATTTTtATTTCAACCGCTCATCTTTTAATCATTGTGCCGGTTTTTGCTGCTTT
TTtATCGCTTGCGGCAGGTCTGCCAGGCTATTTAACACCCAATCCGCCGCGTTTTCTGCT
TCAGGCGTAATAGGTTTACCCGTACGCACCAGCACTTTTGTTCCCACGTTCGCCGCAACC
             *************Truncated*************
GTCGAAACACACTGGGTTTCCCCATTCGGAAATCGCCGGTTATAACGGTTCATATCACCT
TACCGACGCTTATCGCAGATTAGCACGTCCTTCATCGCCTCTGACTGCCAGGGCATCCAC
CGTGTACGCTTAGTCGCTTAACCTCACAACCCGAAGATGTTTCtttCgaTTCAtcAtcgt
GTTGCGAAAATTTGAGAGACTCACGAACAACTTTCgTTGTTCAGTGTtTCAATTtTCAGC
TTGATCCAGATTTTTAAAGAGCAAANNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNTTTtCATCAGACAATCTGTGTGAGCACTgCAAAGtACGCTTCT
TTAAGGTAAGGAGGTGATCCAACCGCAGGTTCCCCTACGGTTACCTTGTTACGACTTCAC
CCCAGTCATGAATCACAAAGTGGTAAGCGCCCTCCCGAAGGTTAAGCTACCTACTTCTTT
TGCAACCCACTCCCATGGTGTGACGGGCGGTGTGTACAAGGCCCGGGAACGTATTCACCG
TGGCATTCTGATCCACGATTACTAGCGATTCCGACTTCATGGAGTCGAGTTGCAGACTCC
```

D



```
>FXAWNEV04JKJEN length=355
GCAACGTTTTCAGCAGTATTACCTGGCGAATGCGCAGGTTCTGCAGACGGCAAACGCGAT
TTTTGATGCGCTGATTAACATTCGCTAAGGGGAGATAAGATGCGTTTCAGTACACAGATG
ATGTACCAGCAAAACATGCGTGGTATCACCAATTCTCAGGCAGAATGGATGAAGTACGGC
GAACAGATGTCGACGGGTAAGCGAGTCGTTAACCCTTCTGACGATCCCATTGCTGCATCA
CAAGCCGTAGTTCTCTCCCAGGCACAGGCGCAAAACNGCCAGTACACGCTGGCGCGTACT
TTCGCCACTCAAAAAGTGTCACTGGAAGAGAGTGTACTTAGCCAGGTCACCACTG
>FXAWNEV04JIKYC length=95
CACGCGCTCGACGTTCTCCACCACCACTATCGCATCATCGACGAGCAGCCCGATGGCAAG
CACCATCCCGAACATCGTTAGTGTGTTGATGGAGT
```

Figure 25: Portion examples of *.fna files for a genomic assembly project. (A) 454AllContigs.fna; (B) 454largeContigs.fna; (C) 454Scaffolds.fna with "Ns" signifying a gap; (D) 454TrimmedReads.fna.



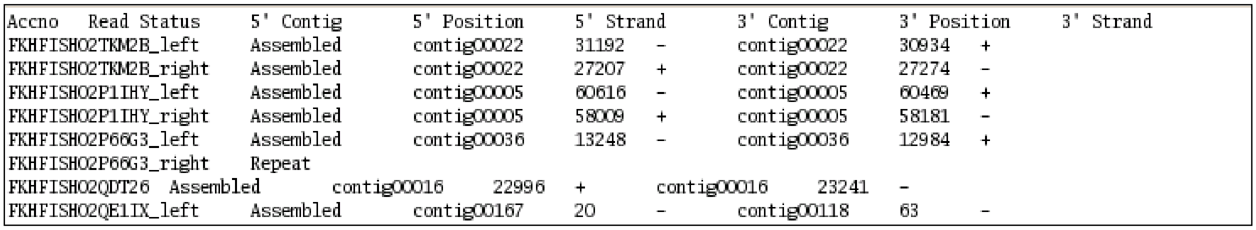
```
>contig00014  length=104  numreads=0  gene=isogroup00001  st
atus=isotig
64 64 64 64 64 64 64 64 15 64 64 64 64 64 64 55 64 64 64 38
64 61 64 64 64 64 64 64 64 64 64 64 64 60 64 64 64 64 64 58
64 64 64 64 64 64 64 64 64 64 64 64 64 64 64 64 64 64 64 64
64 64 64 64 64 64 64 64 64 64 64 64 64 64 64 64 64 64 64 64
64 64 55 64 64 64 64 64 61 64 64 64 64 64 64 64 64 64 64 64
64 64 64 64
```

Figure 26: 454AllContigs.qual file portion example for an Assembly project.

1.16.1.3 454ReadStatus.txt

The 454ReadStatus.txt file (Figure 27) contains the status identifiers for all the reads used in the assembly computation, plus the 3' and 5' positions for each assembled read's alignment within the contig results. The reads are listed one per line, in tab-delimited format. Each line contains the following information (these are the columns in the tab-delimited format):

1. **Accno** – Accession number of the input read. If this is a paired end read, the accno is followed by an underscore character and the mention "left" of "right", for which half of the pair this read comes from.
2. **Read Status** – status of the read in the assembly, which can be one of the following:
    a. **Assembled** – the read is fully incorporated into the assembly
    b. **PartiallyAssembled** – only part of the read was included in the assembly, the rest was deemed to have diverged sufficiently to not be included
    c. **Singleton** – the read did not overlap with any other reads in the input
    d. **Repeat** – the read was either:
        i. Inferred to be repetitive early in the assembly process. A read can be inferred to be repetitive if >70% of the read's seeds hit to at least 70 other reads Such reads are excluded from the assembly
        ii. Determined to partially overlap a contig. The portions of such reads that overlap unique contigs are still included in the assembly results
    e. **Outlier** – the read was identified by the GS *De Novo* Assembler as problematic, and was excluded from the final contigs (one explanation of these outliers are chimeric sequences, but sequences may be identified as outliers simply as an assembler artifact)
    f. **TooShort** – the trimmed read was too short to be used in the computation (shorter than 50 bases and longer than the value of the minlen parameter, unless 454 paired end reads are included in the data set, in which case, all reads at least "minlen" bases are used).
3. **5' Contig** – The accno of the contig in which the 5' end of the read's alignment begins.
4. **5' Position** – The position in the 5' contig where the 5' end of the read's alignment begins.
5. **5' Strand** – The orientation of the read's alignment relative to the 5' contig. A '+' indicates the alignment orientation of the read is the same as the orientation of the 5' contig. A '-' indicates the alignment orientation of the read is opposite to the orientation of the 5' contig.
6. **3' Contig** – The accno of the contig in which the 3' end of the read's alignment ends.
7. **3' Position** – The position in the 3' contig where the 3' end of the read's alignment ends.
8. **3' Strand** – The orientation of the read's alignment relative to the 3' contig. A '+' indicates the alignment orientation of the read is the same as the orientation of the 3' contig. A '-' indicates the alignment orientation of the read is opposite to the orientation of the 3' contig.



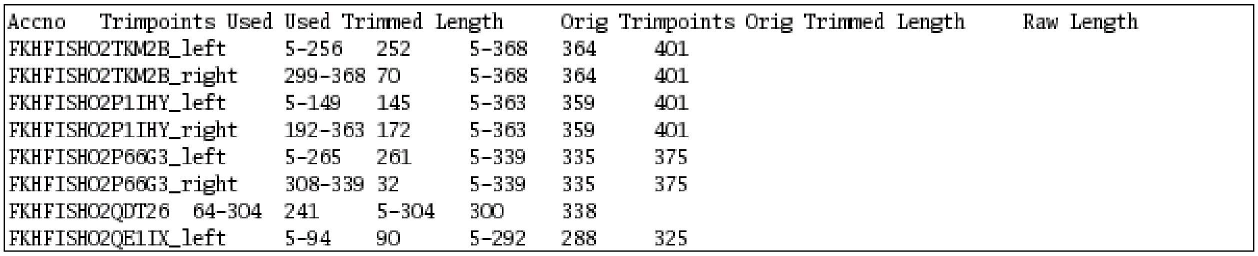
```
Accno   Read Status     5' Contig       5' Position     5' Strand       3' Contig       3' Position     3' Strand
FKHFISHO2TKM2B_left     Assembled       contig00022     31192   -       contig00022     30934   +
FKHFISHO2TKM2B_right    Assembled       contig00022     27207   +       contig00022     27274   -
FKHFISHO2P1IHY_left     Assembled       contig00005     60616   -       contig00005     60469   +
FKHFISHO2P1IHY_right    Assembled       contig00005     58009   +       contig00005     58181   -
FKHFISHO2P66G3_left     Assembled       contig00036     13248   -       contig00036     12984   +
FKHFISHO2P66G3_right    Repeat
FKHFISHO2QDT26  Assembled       contig00016     22996   +       contig00016     23241   -
FKHFISHO2QE1IX_left     Assembled       contig00167     20      -       contig00118     63      -
```

Figure 27: 454ReadStatus.txt file portion example for an Assembly project.

1.16.1.4 454TrimStatus.txt

This file contains a per-read report of the original and revised trimpoints used in the assembly, where the revised trimpoints include the effects of primer, vector and/or quality trimming of the original input reads (Figure 28). Each line contains the following information (these are the columns in the tab-delimited format):

1. **Accno** – accession number of the input read
2. **Trimpoints Used** – the final trimpoints used in the assembly, in #-# format
3. **Trimmed Length** – the final trimmed length of the read
4. **Orig. Trimpoints** – the original trimpoints of the read, found in the SFF or FASTA
5. **Orig. Trimmed Length** – the original trimmed length of the read
6. **Raw Length** – the length of the raw read (without any trimming)



```
Accno   Trimpoints Used Used Trimmed Length     Orig Trimpoints Orig Trimmed Length     Raw Length
FKHFISHO2TKM2B_left     5-256   252     5-368   364     401
FKHFISHO2TKM2B_right    299-368 70      5-368   364     401
FKHFISHO2P1IHY_left     5-149   145     5-363   359     401
FKHFISHO2P1IHY_right    192-363 172     5-363   359     401
FKHFISHO2P66G3_left     5-265   261     5-339   335     375
FKHFISHO2P66G3_right    308-339 32      5-339   335     375
FKHFISHO2QDT26  64-304  241     5-304   300     338
FKHFISHO2QE1IX_left     5-94    90      5-292   288     325
```

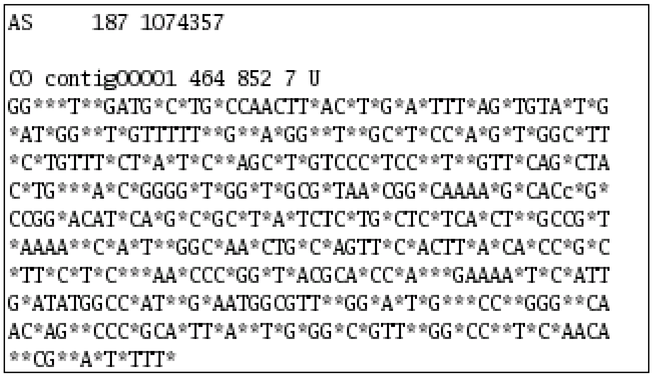
Figure 28: 454TrimStatus.txt file portion example.

1.16.1.5 454Contigs.ace or ace/ContigName.ace or consed/…

This viewer-ready genome file shows all the contigs contained in 454AllContigs.fna and allows the display of how the individual reads aligned to those contigs, in an ACE format file suitable for use in various third-party sequence finishing programs (Figure 29). (The freeware "clview" application can be downloaded from: http://compbio.dfci.harvard.edu/tgi/software/; a full description of the .ace file format can be found at: http://bozeman.mbt.washington.edu/consed/consed.html.) It should be noted, however, that such third-party viewing software will not be able to make full use of the flowspace assembly information available with 454 Sequencing reads, and that conversely some of the third-party program's functions (*e.g.* involving sequence chromatogram input) are not usable with 454 Sequencing data sets. Nonetheless, these programs may be useful to view and assess read characteristics and coverage depth in regions of interest.

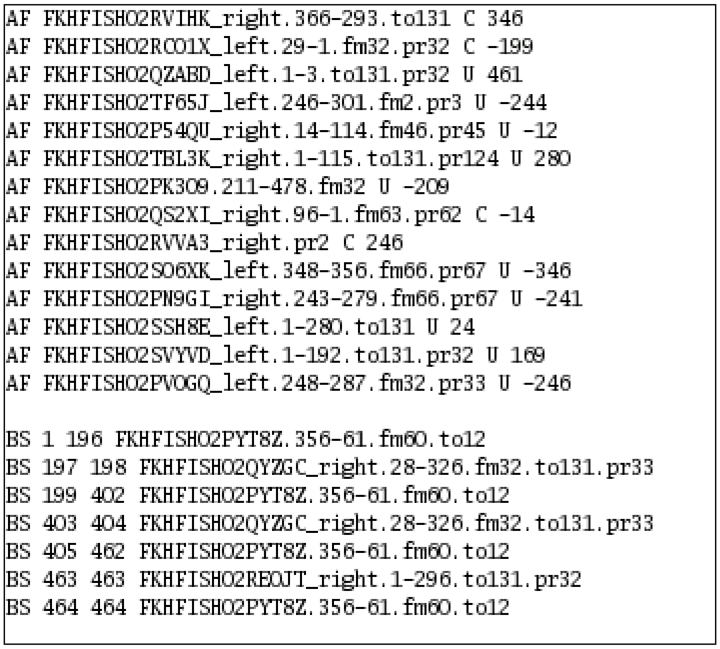
The data software analysis applications can also output the assembly results in a complete directory structure suitable for use as input to the consed software (see the above link for a description of this structure and its files). When the appropriate option is selected, a "consed" sub-directory is produced in the output (or project) directory for the computation. This directory contains the sub-directories, the ACE file and the PHD file, so that the functions of consed (viewing traces, editing reads, auto finishing) can be performed on the assembly. In order to integrate the 454 Sequencing reads (and their SFF files) into the consed data, an extra "sff_dir" directory is created in the consed sub-directory, and a number of consed options are automatically specified in this directory (see the "consed/edit_dir/.consedrc" file for the options specified by the generated structure). One option tells consed to use the "sff2scf" command to access any trace information requested by a user. See Section 3.3 for a description of the sff2scf command, and how it can be used to generate synthetic traces from SFF data, as well as provide a pass-through access to SCF data for Sanger reads.

A

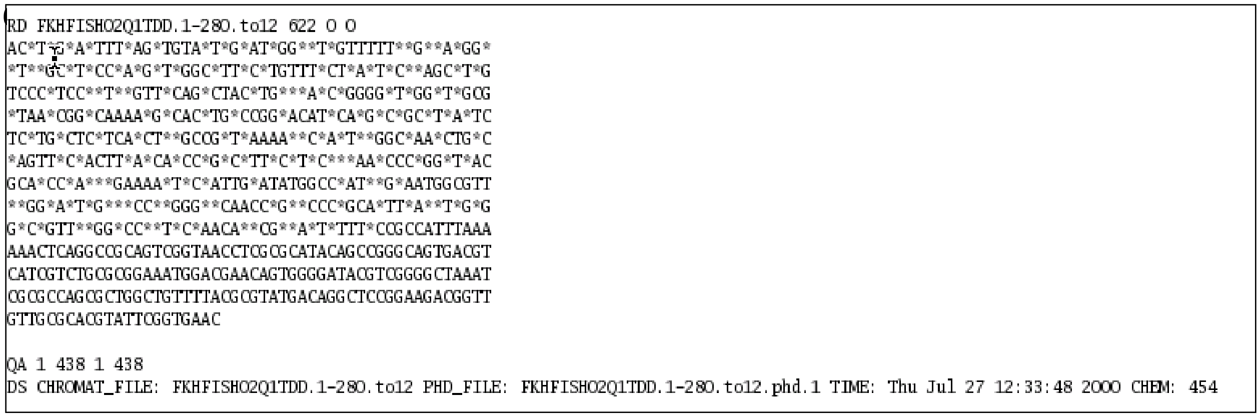


```
AS     187 1074357

CO contig00001 464 852 7 U
GG***T**GATG*C*TG*CCAACTT*AC*T*G*A*TTT*AG*TGTA*T*G
*AT*GG**T*GTTTTT**G**A*GG**T**GC*T*CC*A*G*T*GGC*TT
*C*TGTTT*CT*A*T*C**AGC*T*GTCCC*TCC**T**GTT*CAG*CTA
C*TG***A*C*GGGG*T*GG*T*GCG*TAA*CGG*CAAAA*G*CACc*G*
CCGG*ACAT*CA*G*C*GC*T*A*TCTC*TG*CTC*TCA*CT**GCCG*T
*AAAA**C*A*T**GGC*AA*CTG*C*AGTT*C*ACTT*A*CA*CC*G*C
*TT*C*T*C***AA*CCC*GG*T*ACGCA*CC*A***GAAAA*T*C*ATT
G*ATATGGCC*AT**G*AATGGCGTT**GG*A*T*G***CC**GGG**CA
AC*AG**CCC*GCA*TT*A**T*G*GG*C*GTT**GG*CC**T*C*AACA
**CG**A*T*TTT*
```

B



```
AF FKHFISH02RVIHK_right.366-293.to131 C 346
AF FKHFISH02RCO1X_left.29-1.fm32.pr32 C -199
AF FKHFISH02QZABD_left.1-3.to131.pr32 U 461
AF FKHFISH02TF65J_left.246-301.fm2.pr3 U -244
AF FKHFISH02P54QU_right.14-114.fm46.pr45 U -12
AF FKHFISH02TBL3K_right.1-115.to131.pr124 U 280
AF FKHFISH02PK309.211-478.fm32 U -209
AF FKHFISH02QS2XI_right.96-1.fm63.pr62 C -14
AF FKHFISH02RVVA3_right.pr2 C 246
AF FKHFISH02SO6XK_left.348-356.fm66.pr67 U -346
AF FKHFISH02PN9GI_right.243-279.fm66.pr67 U -241
AF FKHFISH02SSH8E_left.1-280.to131 U 24
AF FKHFISH02SVYVD_left.1-192.to131.pr32 U 169
AF FKHFISH02PVOGQ_left.248-287.fm32.pr33 U -246

BS 1 196 FKHFISH02PYT8Z.356-61.fm60.to12
BS 197 198 FKHFISH02QYZGC_right.28-326.fm32.to131.pr33
BS 199 402 FKHFISH02PYT8Z.356-61.fm60.to12
BS 403 404 FKHFISH02QYZGC_right.28-326.fm32.to131.pr33
BS 405 462 FKHFISH02PYT8Z.356-61.fm60.to12
BS 463 463 FKHFISH02REOJT_right.1-296.to131.pr32
BS 464 464 FKHFISH02PYT8Z.356-61.fm60.to12
```



```
RD FKHFISHO2Q1TDD.1-280.to12 622 0 0
AC*T*G*A*TTT*AG*TGTA*T*G*AT*GG**T*GTTTTT**G**A*GG*
*T**GC*T*CC*A*G*T*GGC*TT*C*TGTTT*CT*A*T*C**AGC*T*G
TCCC*TCC**T**GTT*CAG*CTAC*TG***A*C*GGGG*T*GG*T*GCG
*TAA*CGG*CAAAA*G*CAC*TG*CCGG*ACAT*CA*G*C*GC*T*A*TC
TC*TG*CTC*TCA*CT**GCCG*T*AAAA**C*A*T**GGC*AA*CTG*C
*AGTT*C*ACTT*A*CA*CC*G*C*TT*C*T*C***AA*CCC*GG*T*AC
GCA*CC*A***GAAAA*T*C*ATTG*ATATGGCC*AT**G*AATGGCGTT
**GG*A*T*G***CC**GGG**CAACC*G**CCC*GCA*TT*A**T*G*G
G*C*GTT**GG*CC**T*C*AACA**CG**A*T*TTT*CCGCCATTTAAA
AAACTCAGGCCGCAGTCGGTAACCTCGCGCATACAGCCGGGCAGTGACGT
CATCGTCTGCGCGGAAATGGACGAACAGTGGGGATACGTCGGGGCTAAAT
CGCGCCAGCGCTGGCTGTTTTACGCGTATGACAGGCTCCGGAAGACGGTT
GTTGCGCACGTATTCGGTGAAC

QA 1 438 1 438
DS CHROMAT_FILE: FKHFISHO2Q1TDD.1-280.to12 PHD_FILE: FKHFISHO2Q1TDD.1-280.to12.phd.1 TIME: Thu Jul 27 12:33:48 2000 CHEM: 454
```

Figure 29: 454Contigs.ace file portions example for a genomic DNA project with paired end reads. (A) Contig sequence and quality information; (B) Information mapping reads to contigs; (C) Read sequence and quality information.

The 454Contigs.ace file can be generated with the -a (all contigs length) threshold option, which filters out contigs whose lengths are less than the threshold. On rare occasions, the ace file may contain contigs whose lengths are less than this threshold. This is due to the two-phase nature by which contig consensi are determined. The –a setting is ignored in cDNA assembly computations.

The read information included in the ACE file produced by the GS *De Novo* Assembler application differs from traditional files in that a single read may appear in multiple contigs of the assembly. This occurs because the objective of the 454 Sequencing system software is to first identify and partition the repeat and non-repeat regions of the genome, and then output the consensus sequences for those regions. Therefore, if a single read spans the boundary between two contigs, and either one of these contigs consists of a repeat region, the read will be displayed in both the repeat contig and the non-repeat contig on the other side of the boundary.

To ensure compatibility with ACE file viewers, as well as to assist in the post-analysis of the assembly, the identifiers for the reads that appear in multiple places, and for paired end reads if present, are output with an additional suffix. Several suffixes may be added to the original read identifier:

- For 454 paired end reads, the two halves of the 454 read that constitute the sequences at the two ends of the original clone (from which the paired end read was generated) are marked by "_left" and "_right" suffixes.
- If only part of the trimmed read aligns in this contig (either because the read is only Partially Assembled, or the read is aligned with different sections in different contigs, because it spans a repeat region boundary, for example), the base position range of the region aligned in this contig is added, as in ".1-60" if the read has bases 1-60 aligned in the contig.
- If a read has sections aligning to different contigs, then a ".fm" or ".to" suffix (or both) is added to list the contig the read's alignment is "coming from" or "going to". The ".fm" suffix is added when the read's alignment continues into the 5' end of the contig, and the ".to" suffix is added for reads that lead to another contig off the 3' end of the contig. For example, if read "C3U5GNB01C40I3" appears in contigs 2 and 45, with bases 1-60 appearing in contig 2 and bases 61-248 appearing in contig 45, the entry in contig 2 will use the identifier "C3U5GNB01C40I3.1-60.to45" and the entry in contig 45 will be "C3U5GNB01C40I3.61-248.fm2".
- Paired end reads where both halves of the pair align into contigs will have a ".pr" suffix to list where the other half aligns in the assembly. For example, if reads "C3U5GNB01R8GSX_left" and "C3U5GNB01R8GSX_right" align in contigs 36 and 87, respectively, then the accessions will appear as "C3U5GNB01R8GSX_left.pr87" and "C3U5GNB01R8GSX_right.pr36" (assuming that their complete trimmed sequences aligned inside the two contigs, and did not span across multiple contigs).

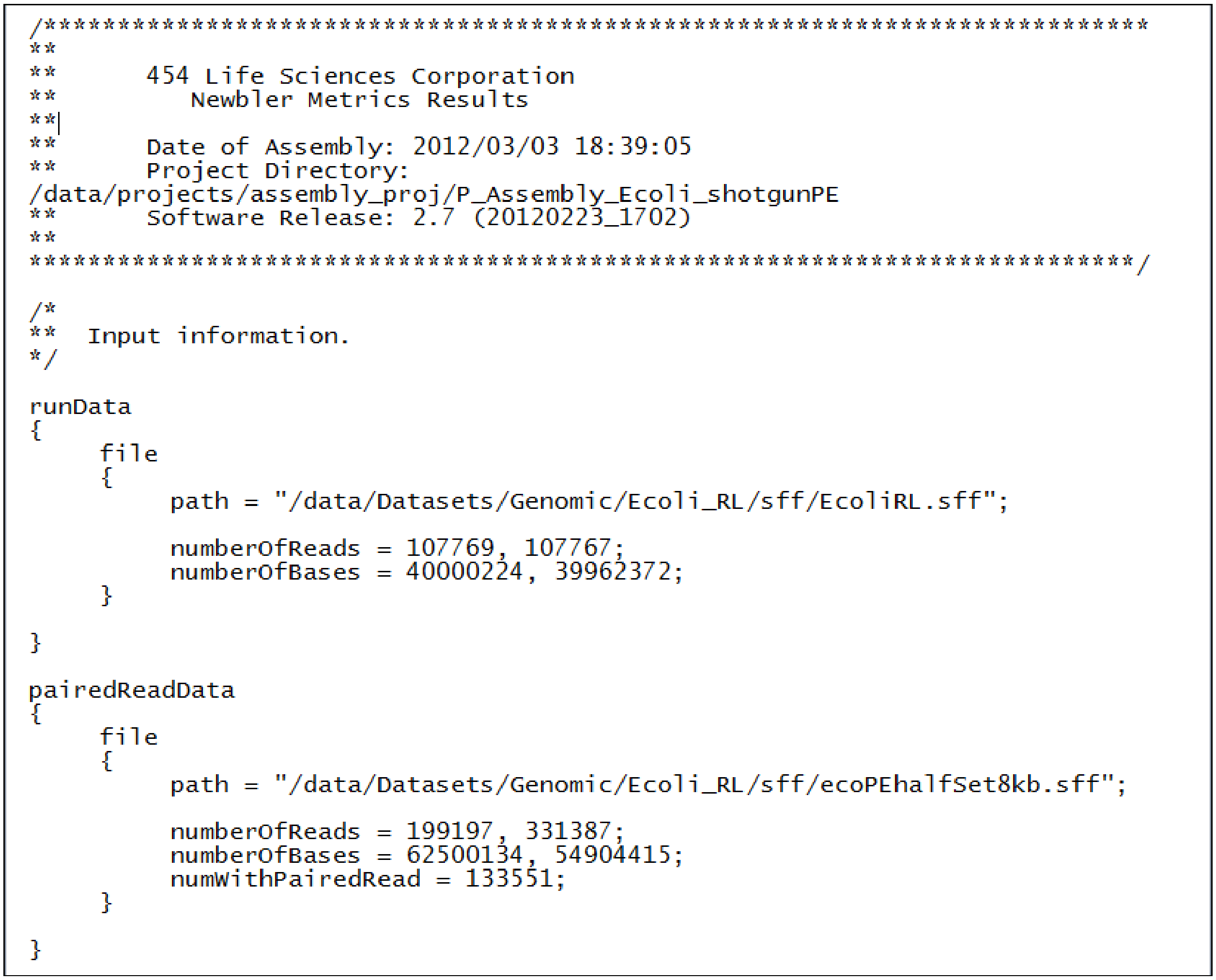
1.16.1.6 454NewblerMetrics.txt

The 454NewblerMetrics.txt file is a 454 parser file (see the General Overview section of this manual for a description of the file formats) that reports the key input, algorithmic and output metrics for the data analysis software applications. Each application that uses the Newbler algorithm creates a 454NewblerMetrics.txt file with relevant output. Below is a description of all the sections present in this file when produced by the GS *De Novo* Assembler, with their named groups and keywords.

Percentage metrics are calculated in two distinct ways, with the first value for each pair of percentages calculated based on the number of reads (or bases) used in the assembly computation, and the second percentage value calculated from the total number of physical reads. In the case of paired end reads in SFF files, the total number of reads in the file is approximated by counting the paired end reads twice. This prevents percentages greater than 100% from being reported.

1.16.1.6.1 Input information (Figure 30)

- **runData group** – contains information about the read data used in the analysis (both Sanger and non-Paired-End 454 Sequencing read files are reported on in this section) (not shown in Figure 30 since only paired end data files were used in this example).
- **pairedReadData group** – contains information about the paired end input data (paired end only; 454 Sequencing system and Sanger if any).

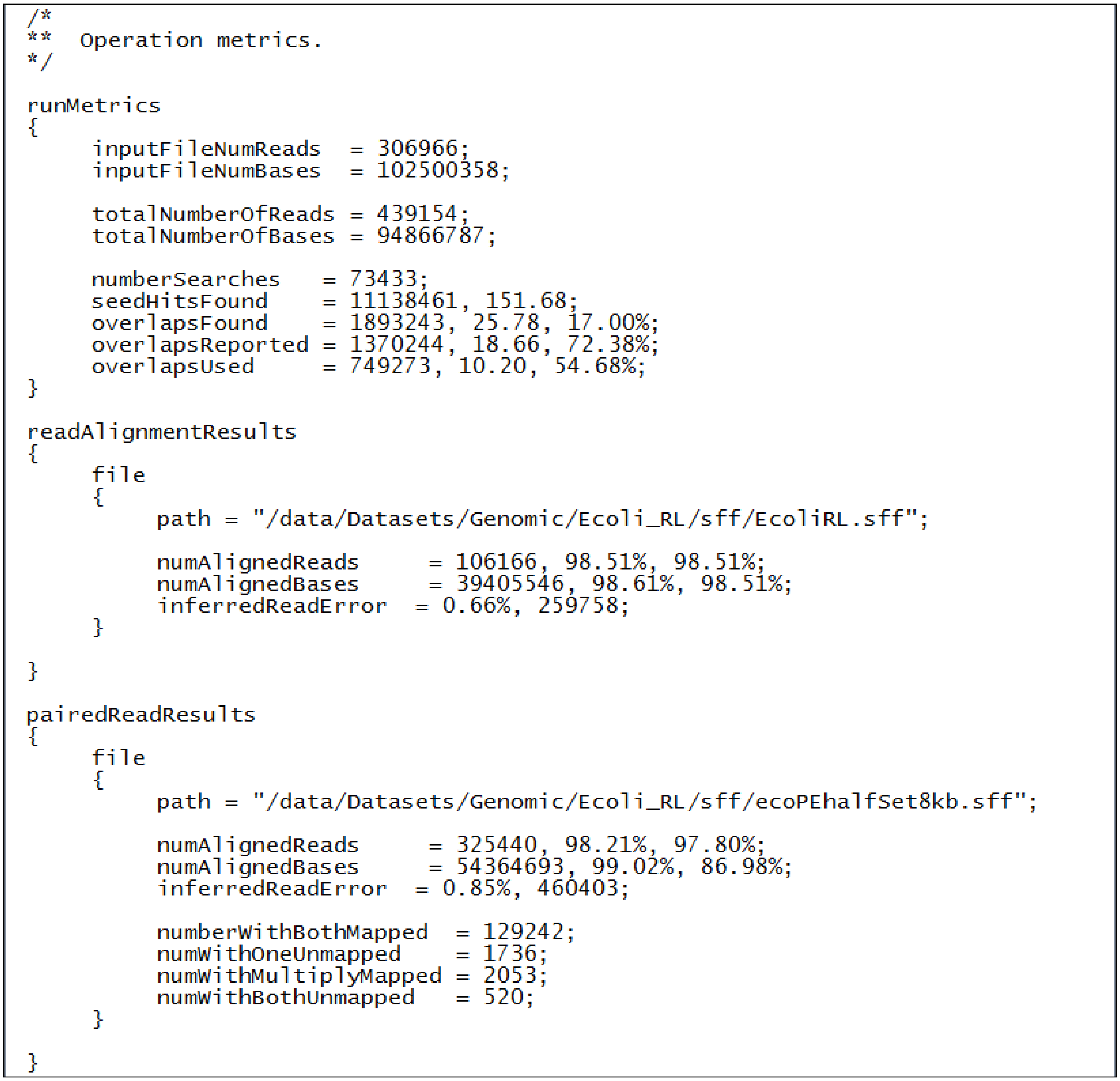


```
/***************************************************************************
**
**      454 Life Sciences Corporation
**         Newbler Metrics Results
**|
**      Date of Assembly: 2012/03/03 18:39:05
**      Project Directory:
/data/projects/assembly_proj/P_Assembly_Ecoli_shotgunPE
**      Software Release: 2.7 (20120223_1702)
**
***************************************************************************/

/*
**  Input information.
*/

runData
{
     file
     {
          path = "/data/Datasets/Genomic/Ecoli_RL/sff/EcoliRL.sff";

          numberOfReads = 107769, 107767;
          numberOfBases = 40000224, 39962372;
     }

}

pairedReadData
{
     file
     {
          path = "/data/Datasets/Genomic/Ecoli_RL/sff/ecoPEhalfSet8kb.sff";

          numberOfReads = 199197, 331387;
          numberOfBases = 62500134, 54904415;
          numWithPairedRead = 133551;
     }

}
```

Figure 30: 454NewblerMetrics file, runData and pairedReadData groups portion example.

1.16.1.6.2 Operation metrics (Figure 31)

- **runMetrics group** – contains information about the assembly computation.
- **readAlignmentResults group** – contains information about the alignments for each input file (SFF, FASTA/FASTQ, or run regions from wells file.
- **pairedReadResults group** – contains information about the paired end input data (paired end only; 454 Sequencing system and Sanger, if any).



```
/*
**  Operation metrics.
*/

runMetrics
{
     inputFileNumReads   = 306966;
     inputFileNumBases   = 102500358;

     totalNumberOfReads = 439154;
     totalNumberOfBases = 94866787;

     numberSearches    = 73433;
     seedHitsFound     = 11138461, 151.68;
     overlapsFound     = 1893243, 25.78, 17.00%;
     overlapsReported = 1370244, 18.66, 72.38%;
     overlapsUsed      = 749273, 10.20, 54.68%;
}

readAlignmentResults
{
     file
     {
          path = "/data/Datasets/Genomic/Ecoli_RL/sff/EcoliRL.sff";

          numAlignedReads     = 106166, 98.51%, 98.51%;
          numAlignedBases     = 39405546, 98.61%, 98.51%;
          inferredReadError  = 0.66%, 259758;
     }

}

pairedReadResults
{
     file
     {
          path = "/data/Datasets/Genomic/Ecoli_RL/sff/ecoPEhalfSet8kb.sff";

          numAlignedReads     = 325440, 98.21%, 97.80%;
          numAlignedBases     = 54364693, 99.02%, 86.98%;
          inferredReadError  = 0.85%, 460403;

          numberWithBothMapped   = 129242;
          numWithOneUnmapped     = 1736;
          numWithMultiplyMapped = 2053;
          numWithBothUnmapped    = 520;
     }

}
```

Figure 31: 454NewblerMetrics file, runMetrics, readAlignmentResults, and pairedReadResults groups portion example.

1.16.1.6.3 Consensus distribution information

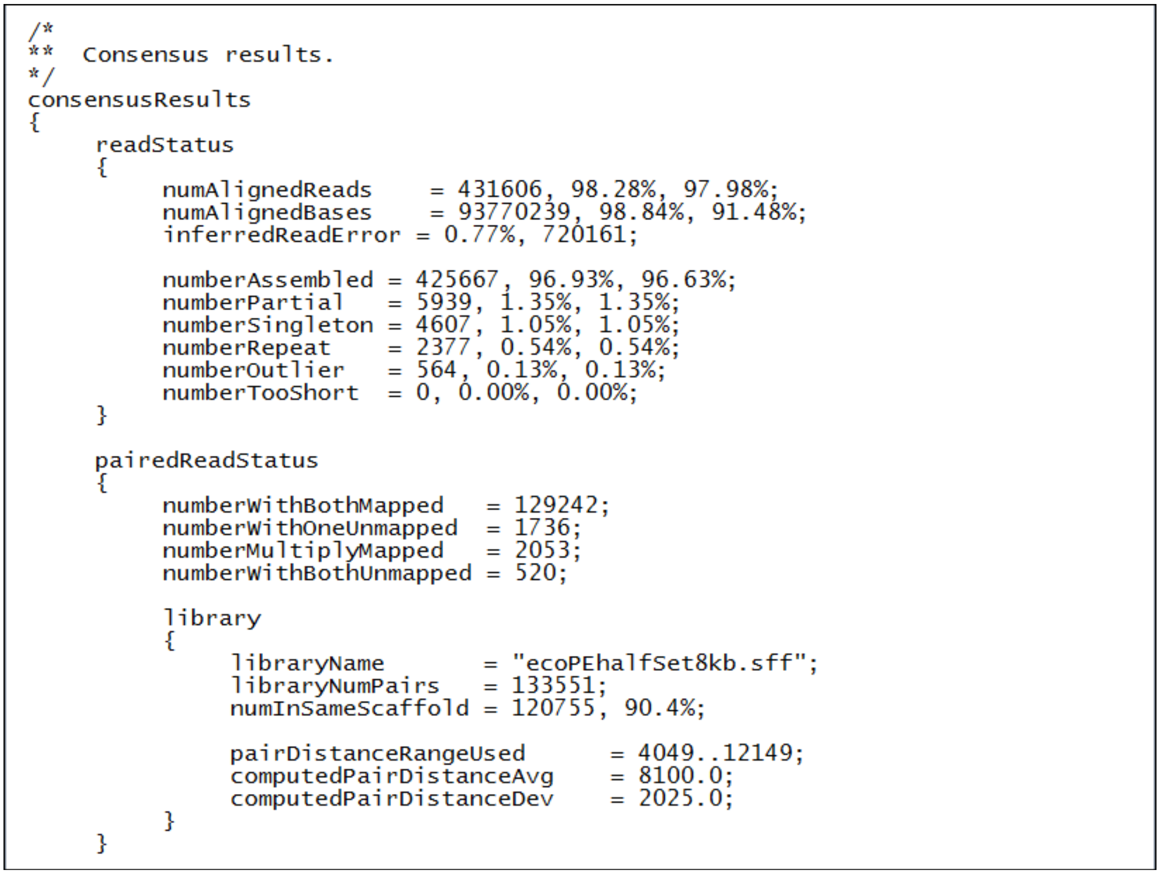
- **consensusDistribution group** – contains information about the consensus signals and basecalling thresholds
  - **fullDistribution**
  - **distributionPeaks**
  - **thresholdsUsed**

1.16.1.6.4 Alignment depths

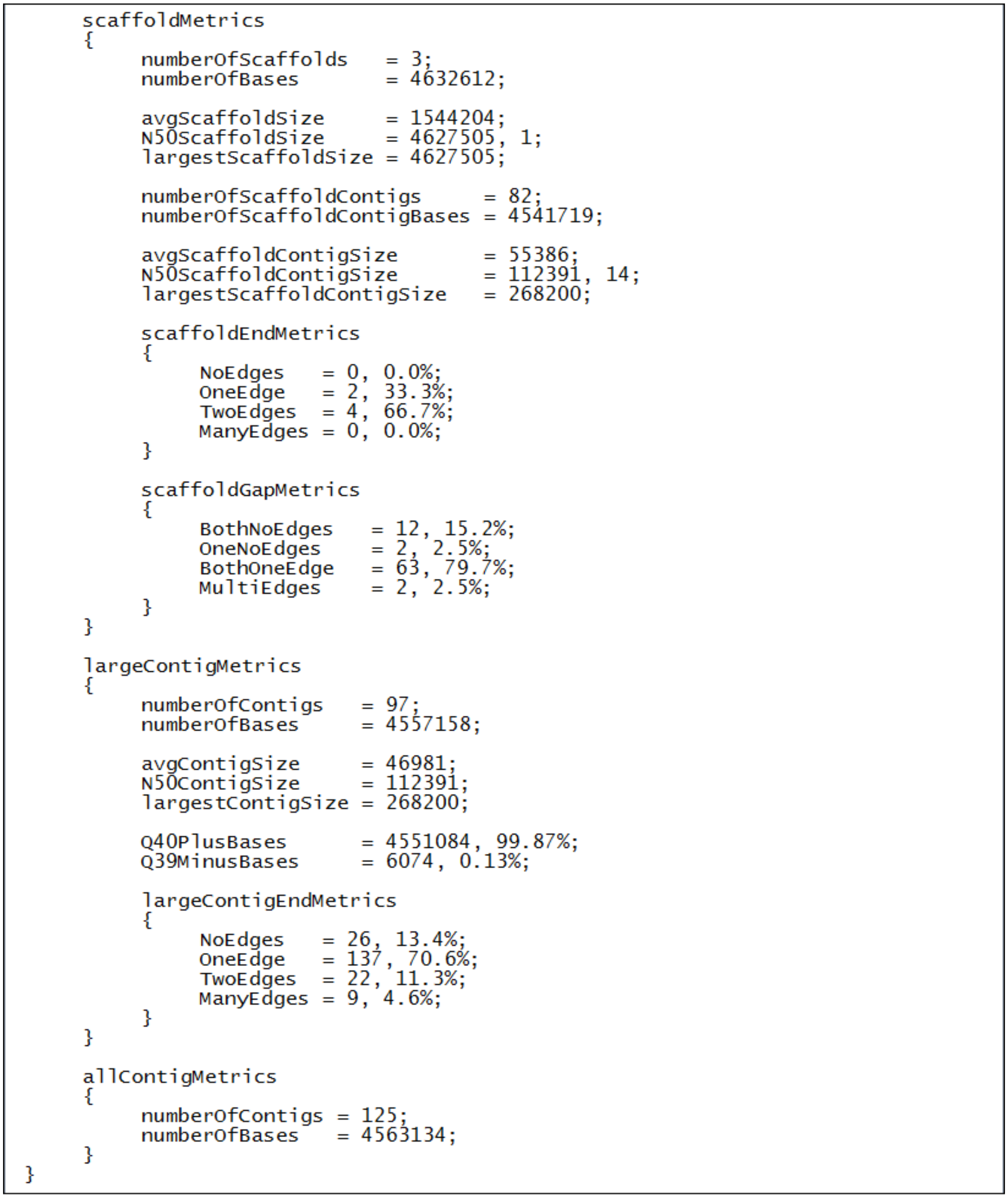
- **alignmentDepths group** – provides a histogram of the number of alignment positions at each coverage depth (including the gaps at all positions)

1.16.1.6.5 Consensus results (Figure 32).

- **consensusResults group** – contains summary information and statistics about reads, scaffolds, and contigs.
  - **readStatus** – summary information about the reads
  - **pairedReadStatus** – paired end library statistics (if paired end reads used).
  - **scaffoldMetrics** – scaffold statistics (if paired end reads are used).
  - **largeContigMetrics** – contig statistics for large contigs (longer than ‘largeContigThreshold’; default is 500 bp).
  - **allContigMetrics**– contig statistics for all contigs (default is 100 bp).



```
/*
**  Consensus results.
*/
consensusResults
{
     readStatus
     {
          numAlignedReads    = 431606, 98.28%, 97.98%;
          numAlignedBases    = 93770239, 98.84%, 91.48%;
          inferredReadError = 0.77%, 720161;

          numberAssembled = 425667, 96.93%, 96.63%;
          numberPartial   = 5939, 1.35%, 1.35%;
          numberSingleton = 4607, 1.05%, 1.05%;
          numberRepeat    = 2377, 0.54%, 0.54%;
          numberOutlier   = 564, 0.13%, 0.13%;
          numberTooShort  = 0, 0.00%, 0.00%;
     }

     pairedReadStatus
     {
          numberWithBothMapped   = 129242;
          numberWithOneUnmapped  = 1736;
          numberMultiplyMapped   = 2053;
          numberWithBothUnmapped = 520;

          library
          {
               libraryName        = "ecoPEhalfSet8kb.sff";
               libraryNumPairs    = 133551;
               numInSameScaffold = 120755, 90.4%;

               pairDistanceRangeUsed      = 4049..12149;
               computedPairDistanceAvg    = 8100.0;
               computedPairDistanceDev    = 2025.0;
          }
     }
```



```
     scaffoldMetrics
     {
          numberOfScaffolds    = 3;
          numberOfBases        = 4632612;

          avgScaffoldSize      = 1544204;
          N50ScaffoldSize      = 4627505, 1;
          largestScaffoldSize = 4627505;

          numberOfScaffoldContigs      = 82;
          numberOfScaffoldContigBases = 4541719;

          avgScaffoldContigSize        = 55386;
          N50ScaffoldContigSize        = 112391, 14;
          largestScaffoldContigSize    = 268200;

          scaffoldEndMetrics
          {
               NoEdges    = 0, 0.0%;
               OneEdge    = 2, 33.3%;
               TwoEdges   = 4, 66.7%;
               ManyEdges = 0, 0.0%;
          }

          scaffoldGapMetrics
          {
               BothNoEdges    = 12, 15.2%;
               OneNoEdges     = 2, 2.5%;
               BothOneEdge    = 63, 79.7%;
               MultiEdges     = 2, 2.5%;
          }
     }

     largeContigMetrics
     {
          numberOfContigs    = 97;
          numberOfBases      = 4557158;

          avgContigSize      = 46981;
          N50ContigSize      = 112391;
          largestContigSize = 268200;

          Q40PlusBases       = 4551084, 99.87%;
          Q39MinusBases      = 6074, 0.13%;

          largeContigEndMetrics
          {
               NoEdges    = 26, 13.4%;
               OneEdge    = 137, 70.6%;
               TwoEdges   = 22, 11.3%;
               ManyEdges = 9, 4.6%;
          }
     }

     allContigMetrics
     {
          numberOfContigs = 125;
          numberOfBases    = 4563134;
     }
}
```

Figure 32: 454NewblerMetrics file, consensusResults group example for a paired end genomic project.

Groups of reads that overlap from the end of one contig to the beginning of another form an edge between the contigs. The presence of many contigs with no edges may indicate insufficient sequencing depth.

1.16.1.7 454NewblerProgress.txt

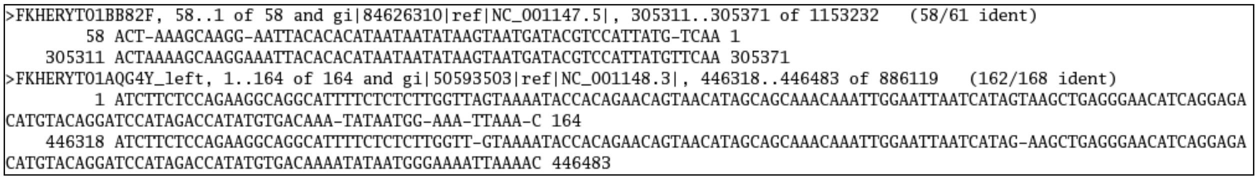
This file represents the text log of the messages sent to standard output by the runProject command (showing the progress of the execution of the assembly computation). If runs are added incrementally and multiple executions of runProject occur, the output messages are appended to this file. If the "Incremental *De Novo* assembler analysis" checkbox option is not selected in the GUI application, or if the "-r" option is given on the runProject command line, the GS *De Novo* Assembler deletes this file and "restarts" the assembly computation.

1.16.1.8 454PairAlign.txt

This file contains the pairwise alignments of the overlaps that were found during the assembly computation (Figure 33). By default, this file is not generated, but if the "-pair" or "-pairt" options are given on the runProject command line, it will be generated either in a human-readable text format ("-pair") or in tab-delimited format ("-pairt").

Each of the displayed alignments contains the following columns of information:

1. **QueryAccno** – accession number of the read used in the overlap detection search (the "query sequence")
2. **QueryStart** – starting position of the alignment in query sequence
3. **QueryEnd** – ending position of the alignment in query sequence
4. **QueryLength** – length of the query sequence
5. **SubjAccno** – accession number of the other read (the "subject sequence")
6. **SubjStart** – starting position of the alignment in subject sequence
7. **SubjEnd** – ending position of the alignment in subject sequence
8. **SubjLength** – length of the subject sequence
9. **NumIdent** – number of identities in the pairwise alignment, *i.e.* where query and subject characters match
10. **AlignLength** – the length of the pairwise alignment
11. **QueryAlign** – query alignment sequence
12. **SubjAlign** – subject alignment sequence



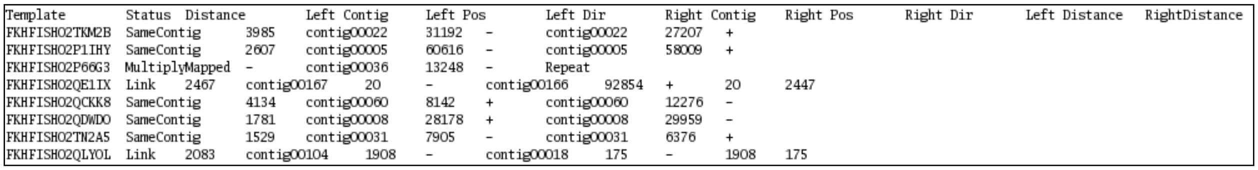
```
>FKHERYT01BB82F, 58..1 of 58 and gi|84626310|ref|NC_001147.5|, 305311..305371 of 1153232   (58/61 ident)
        58 ACT-AAAGCAAGG-AATTACACACATAATAATATAAGTAATGATACGTCCATTATG-TCAA 1
    305311 ACTAAAAGCAAGGAAATTACACACATAATAATATAAGTAATGATACGTCCATTATGTTCAA 305371
>FKHERYT01AQG4Y_left, 1..164 of 164 and gi|50593503|ref|NC_001148.3|, 446318..446483 of 886119   (162/168 ident)
         1 ATCTTCTCCAGAAGGCAGGCATTTTCTCTCTTGGTTAGTAAAATACCACAGAACAGTAACATAGCAGCAAACAAATTGGAATTAATCATAGTAAGCTGAGGGAACATCAGGAGA
CATGTACAGGATCCATAGACCATATGTGACAAA-TATAATGG-AAA-TTAAA-C 164
    446318 ATCTTCTCCAGAAGGCAGGCATTTTCTCTCTTGGTT-GTAAAATACCACAGAACAGTAACATAGCAGCAAACAAATTGGAATTAATCATAG-AAGCTGAGGGAACATCAGGAGA
CATGTACAGGATCCATAGACCATATGTGACAAAATATAATGGGAAAATTAAAAC 446483
```

Figure 33: 454PairAlign.txt file portion example for a genomic project with paired end reads.

1.16.1.9 454PairStatus.txt

This file contains the per-pair report of the location and status of how each paired end pair of reads was used in the assembly. Each line contains the following information (corresponding to the columns in the tab-delimited format; see also Figure 34):

1. **Template** – template string for the pair (this will be the original 454 accession for 454 paired end reads, and the "template" string for Sanger reads)
2. **Status** – the status of the pair in the assembly, with the following possible values:
   a. **BothUnmapped** – both halves of the pair were unmapped
   b. **OneUnmapped** – one of the reads in the pair was unmapped
   c. **MultiplyMapped** – one or both of the reads in the pair were marked as Repeat
   d. **SameContig** – both halves of the pair were assembled into the same contig, with the correct relative orientation, and are within the expected distance of each other
   e. **Link** – the halves were assembled to different contigs, and are near enough to the ends of those contigs that they could be used as a link in a scaffold
   f. **FalsePair** – the halves were assembled, but the orientation of the aligned contigs is inconsistent with a paired end pair or the distance between the halves is outside the expected distance
   g. **NoThreshold** – the halves were assembled, but there was an insufficient number of paired end reads to calculate statistically-significant distance thresholds.
3. **Distance** – for "SameContig" or "Link" pairs, the distance between the halves (where the "Link" distance is simply the sum of the distances from each half to the end of its respective contig).
4. **Left Contig** – the contig where the left half was assembled, or '-' if the read was Unmapped or Repeat. (If the read aligns across multiple contigs, the location of the first base in the read that is aligned is used, to denote where the end of the corresponding clone occurs.)
5. **Left Pos** – the position in the contig where the 5' end of the left half was assembled
6. **Left Dir** – the direction ('+' for the forward strand of the contig and '-' for reverse strand) in which the left half was assembled
7. **Right Contig** – the contig where the right half was assembled, or '-' if the read was Unmapped or Repeat. (If the read aligns across multiple contigs, the location of the first base of the read that is aligned is used, to denote where the end of the corresponding clone occurs.)
8. **Right Pos** – the position in the contig where the 3' end of the right half was assembled
9. **Right Dir** – the direction ('+' for the forward strand of the contig and '-' for reverse strand) in which the right half was assembled
10. **Left Distance** – the distance from the Left Pos to the respective end of the contig (for forward matches, this is the distance to the 3' end of the contig; for reverse matches, to the 5' end)
11. **Right Distance** – the distance from the Right Pos to the respective end of the contig (for forward matches, this is the distance to the 3' end of the contig; for reverse matches, to the 5' end).



```
Template        Status  Distance        Left Contig     Left Pos        Left Dir        Right Contig    Right Pos       Right Dir       Left Distance   RightDistance
FKHFISHO2TKM2B  SameContig      3985    contig00022     31192   -       contig00022     27207   +
FKHFISHO2P1IHY  SameContig      2607    contig00005     60616   -       contig00005     58009   +
FKHFISHO2P66G3  MultiplyMapped  -       contig00036     13248   -       Repeat
FKHFISHO2QE1IX  Link    2467    contig00167     20      -       contig00166     92854   +       20      2447
FKHFISHO2QCKK8  SameContig      4134    contig00060     8142    +       contig00060     12276   -
FKHFISHO2QDWDO  SameContig      1781    contig00008     28178   +       contig00008     29959   -
FKHFISHO2TN2A5  SameContig      1529    contig00031     7905    -       contig00031     6376    +
FKHFISHO2QLYOL  Link    2083    contig00104     1908    -       contig00018     175     -       1908    175
```

Figure 34: 454PairStatus.txt file portion example.

1.16.1.10 454TagPairAlign.txt

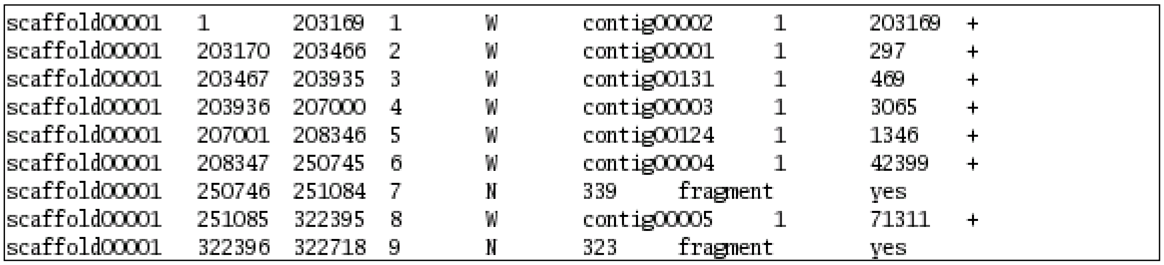
Reads shorter than 50 bp are rare in standard shotgun sequencing runs, and are not used to build contig consensus sequences using the standard overlap detection parameters (Section 1.7.2). Instead, these 'short tag' reads are mapped to the contig consensi in a later step of the computation, using more relaxed overlap detection parameters that allow mapping of shorter sequences. The shortest read mapped in this way is controlled by the -minlen parameter (default 20 bp). Thus, the default length range reported as short tag for standard shotgun runs is 20-49 bp. The -short option will assemble all reads down to the minimum overlap length limit defined by -ml (default 40 bp), which reduces the default range of short tag reads to 20-39 bp.

The most common short tag reads are derived by removing the linker sequence from paired end reads, generating the two half reads designated '_left' and '_right' (see Section 4.6). When paired end reads are detected, the behavior of the -short option is automatically implemented without requiring explicit use of this option.

The 454TagPairAlign.txt file reports the alignments of uniquely-mapped short tag reads against the contig consensi. This file is not generated by default, but output can be controlled from either the 'Pairwise alignment' option of the Output tab of the GUI or from the command line. It can be generated either in a human-readable text format ('Simple format', -pair) or in tab-delimited format ('Tabbed format', -pairt). The format of this file is identical to the 454PairAlign.txt file described in Section 1.16.1.8.

1.16.1.11 454Scaffolds.txt and 454ContigScaffolds.txt

The 454Scaffolds.txt file contains AGP-formatted information describing how the contigs included in the 454LargeContigs.fna file (see section 1.16.1.2) are scaffolded into the sequence scaffolds of 454Scaffolds.fna (section 1.16.1.2). A description of the AGP format can be found on NCBI's web site
(http://www.ncbi.nlm.nih.gov/genome/guide/Assembly/AGP_Specification.html).



```
scaffold00001   1       203169  1       W       contig00002     1       203169  +
scaffold00001   203170  203466  2       W       contig00001     1       297     +
scaffold00001   203467  203935  3       W       contig00131     1       469     +
scaffold00001   203936  207000  4       W       contig00003     1       3065    +
scaffold00001   207001  208346  5       W       contig00124     1       1346    +
scaffold00001   208347  250745  6       W       contig00004     1       42399   +
scaffold00001   250746  251084  7       N       339     fragment        yes
scaffold00001   251085  322395  8       W       contig00005     1       71311   +
scaffold00001   322396  322718  9       N       323     fragment        yes
```

Figure 35: 454Scaffolds.txt file portion example.

When –scaffold is selected, the contents of 454Scaffolds.txt will represent scaffoldContigs and gaps rather than the contigs found in the allContigs file. An additional file, 454ContigScaffolds.txt, will be produced that describes the scaffold layout (identical to the 454Scaffolds.txt file produced without the –scaffold option). The scaffoldContig names found in the 454Scaffolds.txt file represent the scaffoldContigs found in the 454ScaffoldContigs.fna and .qual files.

1.16.1.12 454ContigGraph.txt

The 454ContigGraph.txt file contains a graph-based description of the branching structure of an assembly's contigs, where nodes represent the contigs and edges between contigs give the branching structure. When paired end reads are included in the assembly, the file also shows scaffold edges, describing how the contigs have been linked together into scaffolds. The entries in the file are grouped by type, of which there are six, but not all projects will contain entries for every type. Two types of entries("S" and "P") are only present when paired end reads are included in the assembly.

The first group of entries describes the nodes (contigs) of the graph, and contains the following columns, separated by tab characters:

1. The number of the contig (*i.e.*, for contig "contig00002", this column would contain the number "2")
2. The full name of the contig (*e.g.*, "contig00002")
3. The length of the consensus for this contig
4. The average depth of the contig's multiple alignment

The second group of entries describes the edges of the graph, describing how the alignment of reads branch off in different directions (or converge together) from the end of a contig. Edges connect not contigs, but the ends of contigs (each contig has a 5' and 3' end, corresponding to the beginning (5') and end (3') of the contig multiple alignment). Different contigs are not oriented relative to each other, so an edge can connect the 5' end of contig00001 and the 5' end of contig00002 (or the 3' ends of two contigs). In this situation, reverse complementing one of the contigs, so that the edge is a 5' to 3' edge, presents the two contigs in a consistent orientation with each other. The columns for these entries are the following (separated by tab characters):

1. 'C' – the letter 'C' to mark this entry as a "contig edge"
2. The contig number on one side of the edge
3. Either "5'" or "3'" to denote which end of the contig
4. The contig number on the other side of the edge
5. Either "5'" or "3'" to denote which end of the contig
6. The depth of the multiple alignment that spans between the two contig ends (*i.e.*, the number of reads whose alignments crosses the two contigs ends).

For example, the line

```
C    1    5′    2    3′    21
```

describes an edge between the 5' end of contig00001 and the 3' end of contig00002, where the alignments of 21 reads occur partially at the 5' end of contig00001 and partially at the 3' end of contig00002. All edges are undirected edges.

If the assembly project includes paired end reads, the next group of entries describes the "scaffold edges" found when performing scaffolding. Only edges used in the scaffolding itself are listed (not edges representing all paired end data). The columns for these entries are the following (separated by tab characters):

1. 'S' – the letter 'S' to mark this entry as a "scaffold edge"
2. The contig number on one side of the edge
3. Either "5'" or "3'" to denote which end of the contig
4. The contig number on the other side of the edge
5. Either "5'" or "3'" to denote which end of the contig

For example, the line

```
S     6     3′     8     5′
```

describes an edge between the 3' end of contig00006 and the 5' end of contig00008. All edges are undirected edges.

The next group of entries ("I" entries") exist for most contigs shorter than 256 bp. The columns for this group are the following (separated by tab characters):

1. 'I' – the letter 'I' to mark this entry as a short contig
2. The contig number
3. The contig sequence, available only if the contig is a single-chord contig (Not all contigs shorter than 256 bp are single-chord contigs.)
4. Thru-flow Information. Thru-flows are sequences that start prior to the contig and that end after the contig. For the example below, there are two thru-flows. The first flow has 23 sequences that start at the 3' end of contig00025, continue at the 5' end of contig00024, flow out the 3' end of contig00024, flow into the 5' end of contig00712, and terminate in contig00712. The second flow has 48 reads that start at the 3' end of contig00025, continue at the 5' end of contig00024, flow out the 3' end of contig00024, flow into the 5' end of contig00838, and terminate in contig00838.

```
I     24    CAAGAGATTGGCTCTTCCAGCTTAAACG    23:25-3'..712-5'; 48:25-3'..838-5'
```

The next group of entries contain information about paired-end flows. The columns for this group are the following (separated by tab characters):

1. 'P'– the letter 'P' to mark this entry as a paired-end flow
2. The contig number
3. 5' Paired-End Flow information. This includes information for the paired-end reads "flowing" from the 5' end of the contig. For the first example below, there are 2 Paired-End reads flowing from the 5' end of contig00126 into contig00024. The estimated distance from the 5' end of contig00126 to the end of contig00024 into which the PE reads flow is 394.0 bp (there is no information about the end of contig00024 into which the PE reads flow).

   There are 7 Paired-End reads flowing from the 5' end of contig00126 into contig00047. The estimated distance from the 5' end of contig00126 to the end of contig00047 into which the Paired-End reads flow is 555.7 bp
4. 3' Paired-End Flow information. There are 6 Paired-End reads flowing from contig00126's 3' end into contig00046. The estimated distance from the 3' end of contig00126 to the end of contig00046 is 315.0 bp.

   The second example shows a contig with Paired-End reads flowing off its 5' end, but no PE reads flowing off the 3' end of contig00024.

Example 1:

```
P	126	24/2/394.0;47/7/555.7	46/6/315.0
```

Example 2:

```
P	127	172/3/491.7	-
```

The last group of entries contain information about Read flows. This information details the number of reads flowing from each end of the contig that end in other contigs. The columns for this group are the following (separated by tab characters):

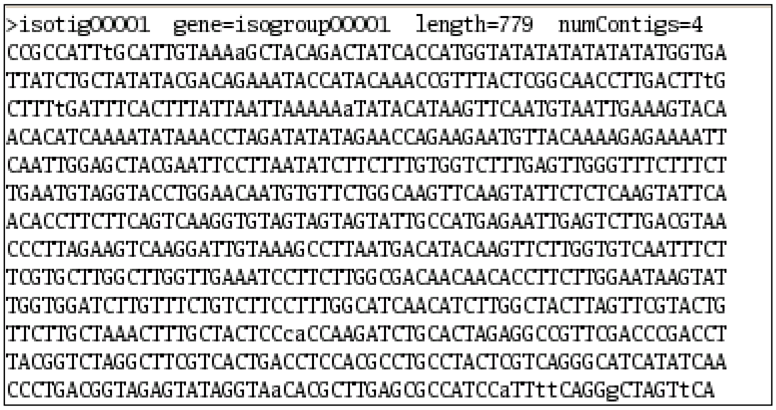
1. "F" – the letter 'F' to mark this entry as a Read flow
2. The contig number
3. 5' Read Flow information. This includes information for the reads "flowing" from the 5' end of the contig. For the example below, there are 67 reads flowing from the 5' end of contig00004 that end in contig00124. The average distance from the 5' end of contig00004 to the end of contig00124 into which the reads flow is 0.0 bp (the two contigs are right next to each other).
4. 3' Read Flow information. There are 46 reads flowing from contig00004's 3' end that end in contig00117 with an average distance of 0.0 bp between the ends of the contigs (again, they are next to each other). There are 3 reads flowing from contig00004's 3' end that terminate in contig00005. The average distance between the contig ends is 101.3 bp (it is likely in this case that the reads travel through contig00117 to get to contig00005 and that contig00117 is 101 bp long).

```
F	4	124/67/0.0	117/46/0.0;5/3/101.3
```

1.16.2 GS *De Novo* Assembler cDNA / Transcriptome Output

1.16.2.1 454Isotigs.fna, 454Isotigs.qual, and 454Isotigs.txt files

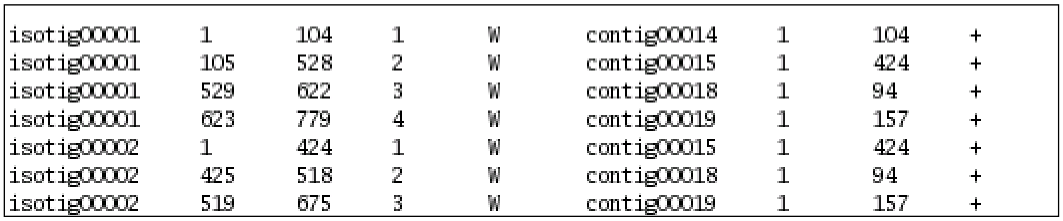
The files 454LargeContigs.fna and 454LargeContigs.qual (1.16.1.2) are not generated for a cDNA / transcriptome assembly project. However, the new files 454Isotigs.fna (Figure 36) and 454Isotigs.qual files are generated (in addition to the 454AllContigs.fna and 454AllContigs.qual files). Similar to contig files, they contain respectively FASTA sequences and their quality scores. These files contain the isotig sequences traversed from the multiple-alignment graph structure (*i.e.* from the isogroup), but they also can contain single contigs which haven't been traversed (and therefore aren't considered isotigs) because of thresholds, limits, cycles, *etc.*



```
>isotig00001  gene=isogroup00001  length=779  numContigs=4
CCGCCATTtGCATTGTAAAaGCTACAGACTATCACCATGGTATATATATATATATGGTGA
TTATCTGCTATATACGACAGAAATACCATACAAACCGTTTACTCGGCAACCTTGACTTtG
CTTTtGATTTCACTTTATTAATTAAAAAaTATACATAAGTTCAATGTAATTGAAAGTACA
ACACATCAAAATATAAACCTAGATATATAGAACCAGAAGAATGTTACAAAAGAGAAAATT
CAATTGGAGCTACGAATTCCTTAATATCTTCTTTGTGGTCTTTGAGTTGGGTTTCTTTCT
TGAATGTAGGTACCTGGAACAATGTGTTCTGGCAAGTTCAAGTATTCTCTCAAGTATTCA
ACACCTTCTTCAGTCAAGGTGTAGTAGTAGTATTGCCATGAGAATTGAGTCTTGACGTAA
CCCTTAGAAGTCAAGGATTGTAAAGCCTTAATGACATACAAGTTCTTGGTGTCAATTTCT
TCGTGCTTGGCTTGGTTGAAATCCTTCTTGGCGACAACAACACCTTCTTGGAATAAGTAT
TGGTGGATCTTGTTTCTGTCTTCCTTTGGCATCAACATCTTGGCTACTTAGTTCGTACTG
TTCTTGCTAAACTTTGCTACTCCcaCCAAGATCTGCACTAGAGGCCGTTCGACCCGACCT
TACGGTCTAGGCTTCGTCACTGACCTCCACGCCTGCCTACTCGTCAGGGCATCATATCAA
CCCTGACGGTAGAGTATAGGTAaCACGCTTGAGCGCCATCCaTTttCAGGgCTAGTtCA
```

Figure 36: Portion example of the 454Isotigs.fna file for a cDNA assembly project.

Also, a special file, 454Isotigs.txt is generated with content and format identical to the 454Scaffolds.txt file(Figure 37).



```
isotig00001    1      104    1      W      contig00014    1      104    +
isotig00001    105    528    2      W      contig00015    1      424    +
isotig00001    529    622    3      W      contig00018    1      94     +
isotig00001    623    779    4      W      contig00019    1      157    +
isotig00002    1      424    1      W      contig00015    1      424    +
isotig00002    425    518    2      W      contig00018    1      94     +
isotig00002    519    675    3      W      contig00019    1      157    +
```

Figure 37: Portion example of the 454Isotigs.txt file for a cDNA assembly project.

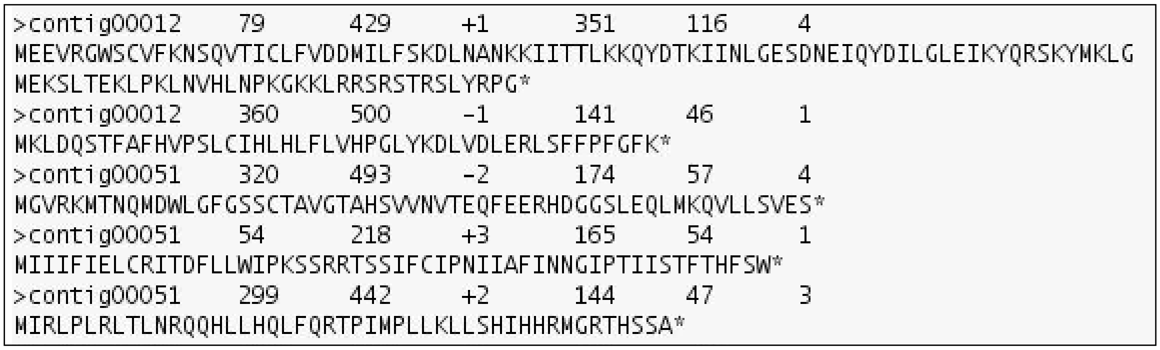
1.16.2.2 454Isotigs.faa and 454IsotigOrfAlign.txt

The 454Isotigs.faa file is a protein sequence FASTA file for all ORFs found in the given isotig nucleotide sequence, that are encoded by 100 or more nucleotides (Figure 38).

The amino acid sequences are sorted by length (for each isotig), and are preceded by a description line that consists of following tab-delimited information:

- **Isotig name** (as it appears in all other output files)
- **Isotig nucleotide start position** (1-based, inclusive)
- **Isotig nucleotide end position** (1-based, inclusive)
- **ORF frame** {-3, -2, -1, +1, +2, +3}
- **Nucleotide sequence length** (including stop codon, if present in the sequence)
- **Protein sequence length** (excluding stop codon, if present in the sequence)
- **Number of M (Methionine) codons**

> The data in the 454isotigs.faa file is based on the assumption that the isotig represents a complete, full-length transcript. Multiple open reading frames may be reported, but only one is correct. The longest ORF is most likely the correct one, but this must be verified by the user. The application does not evaluate the likelihood that the ORF is correct, for example by evaluating codon usage.

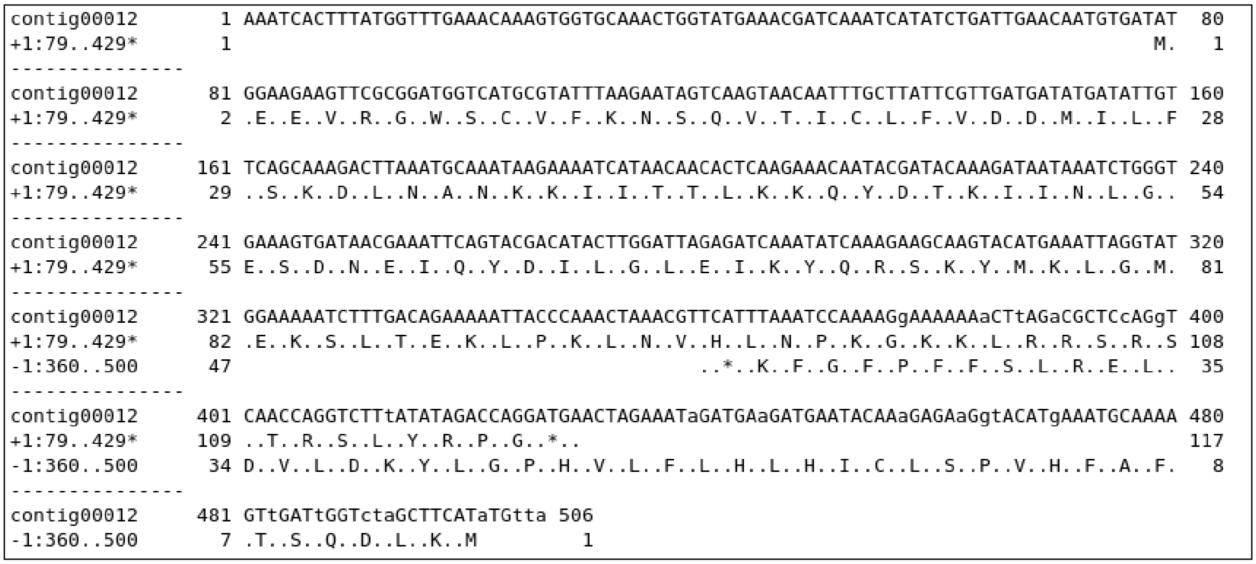


```
>contig00012    79      429     +1      351     116     4
MEEVRGWSCVFKNSQVTICLFVDDMILFSKDLNANKKIITTLKKQYDTKIINLGESDNEIQYDILGLEIKYQRSKYMKLG
MEKSLTEKLPKLNVHLNPKGKKLRRSRSTRSLYRPG*
>contig00012    360     500     -1      141     46      1
MKLDQSTFAFHVPSLCIHLHLFLVHPGLYKDLVDLERLSFFPFGFK*
>contig00051    320     493     -2      174     57      4
MGVRKMTNQMDWLGFGSSCTAVGTAHSVVNVTEQFEERHDGGSLEQLMKQVLLSVES*
>contig00051    54      218     +3      165     54      1
MIIIFIELCRITDFLLWIPKSSRRTSSIFCIPNIIAFINNGIPTIISTFTHFSW*
>contig00051    299     442     +2      144     47      3
MIRLPLRLTLNRQQHLLHQLFQRTPIMPLLKLLSHIHHRMGRTHSSA*
```

Figure 38: Portion example of the 454Isotigs.faa file for a cDNA assembly project.

The 454IsotigOrfAlign.txt file contains one or more lines for isotig nucleotide sequence, and each amino acid sequence encoded by a particular ORF found in the isotig (Figure 39). The isotig sequence is always first and amino acid sequences are sorted by start position in the isotig (or by end position for negative frames). Each line consists of following items:

- **Accno:** The isotig name or ORF info (frame:startBase..endbase); the longest ORF for each isotig ends with an asterisk "*"
- **First position of the current section:** nucleotide base position for an isotig, or amino acid position for an ORF sequence; a negative frame indicates it is greater than the last position for the current section
- **Sequence:** Section of nucleotide or amino acid sequence
- **Last position of the current section:** Nucleotide base position for isotig, or amino acid position for an ORF sequence; a negative frame indicates it is less than the first position for the current section



```
contig00012       1 AAATCACTTTATGGTTTGAAACAAAGTGGTGCAAACTGGTATGAAACGATCAAATCATATCTGATTGAACAATGTGATAT  80
+1:79..429*       1                                                                               M.   1
---------------
contig00012      81 GGAAGAAGTTCGCGGATGGTCATGCGTATTTAAGAATAGTCAAGTAACAATTTGCTTATTCGTTGATGATATGATATTGT 160
+1:79..429*       2 .E..E..V..R..G..W..S..C..V..F..K..N..S..Q..V..T..I..C..L..F..V..D..D..M..I..L..F  28
---------------
contig00012     161 TCAGCAAAGACTTAAATGCAAATAAGAAAATCATAACAACACTCAAGAAACAATACGATACAAAGATAATAAATCTGGGT 240
+1:79..429*      29 ..S..K..D..L..N..A..N..K..K..I..I..T..T..L..K..K..Q..Y..D..T..K..I..I..N..L..G..  54
---------------
contig00012     241 GAAAGTGATAACGAAATTCAGTACGACATACTTGGATTAGAGATCAAATATCAAAGAAGCAAGTACATGAAATTAGGTAT 320
+1:79..429*      55 E..S..D..N..E..I..Q..Y..D..I..L..G..L..E..I..K..Y..Q..R..S..K..Y..M..K..L..G..M.  81
---------------
contig00012     321 GGAAAAATCTTTGACAGAAAAATTACCCAAACTAAACGTTCATTTAAATCCAAAAGgAAAAAAaCTtAGaCGCTCcAGgT 400
+1:79..429*      82 .E..K..S..L..T..E..K..L..P..K..L..N..V..H..L..N..P..K..G..K..K..L..R..R..S..R..S 108
-1:360..500      47                                        ..*..K..F..G..F..P..F..F..S..L..R..E..L..  35
---------------
contig00012     401 CAACCAGGTCTTtATATAGACCAGGATGAACTAGAAATaGATGAaGATGAATACAAaGAGAaGgtACATgAAATGCAAAA 480
+1:79..429*     109 ..T..R..S..L..Y..R..P..G..*..                                                   117
-1:360..500      34 D..V..L..D..K..Y..L..G..P..H..V..L..F..L..H..L..H..I..C..L..S..P..V..H..F..A..F.   8
---------------
contig00012     481 GTtGATtGGTctaGCTTCATaTGtta 506
-1:360..500       7 .T..S..Q..D..L..K..M         1
```

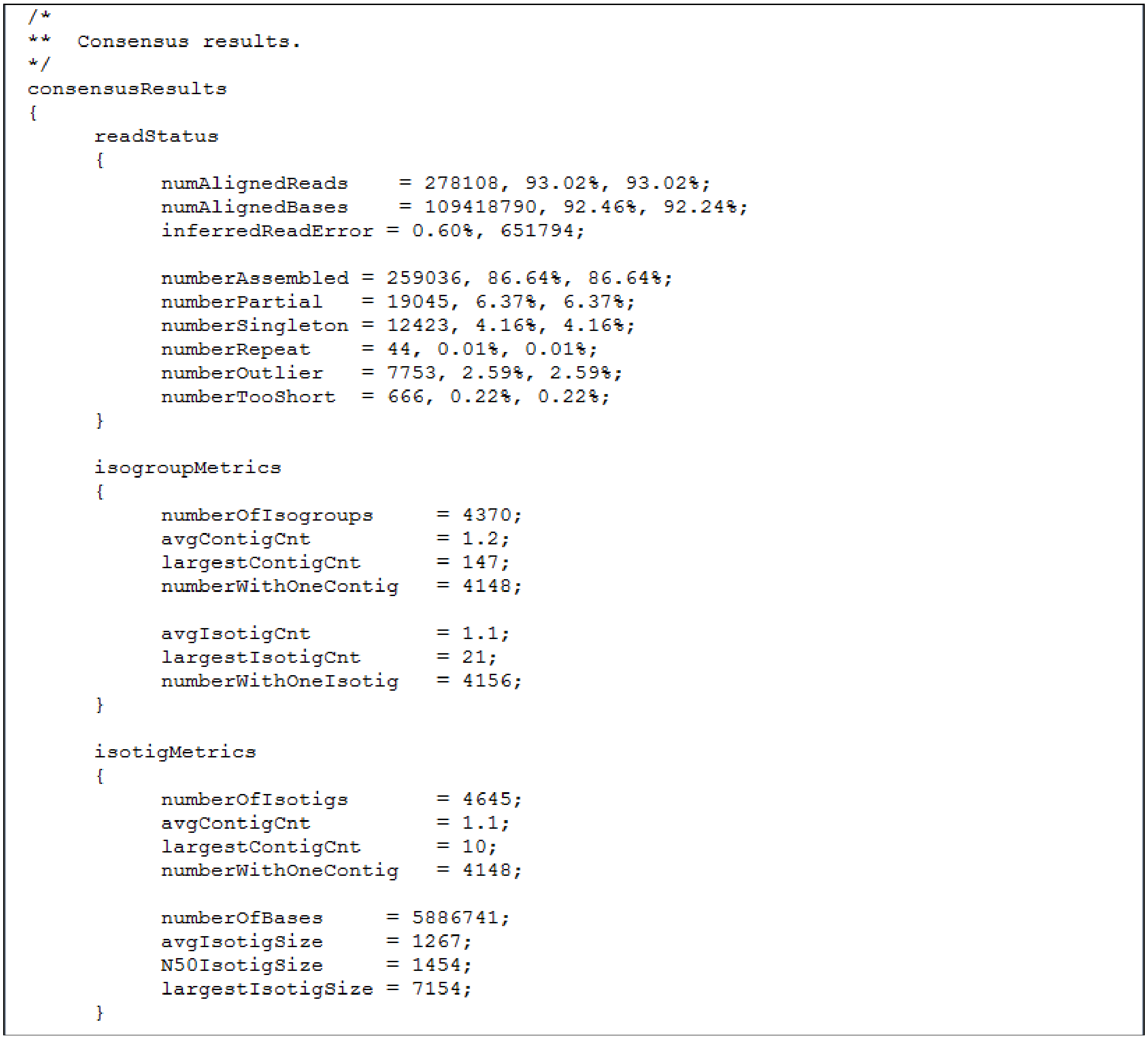
Figure 39: Portion example of the 454IsotigOrfAlign.txt file for a cDNA assembly project.

1.16.2.3 454Isotigs.ace and consed/…

The cDNA assembler produces an ACE file similar to that produced by the genomic assembler (see 1.16.1.5). Isotigs and large contigs that do not appear in any isotig are represented in the ACE file along with reads used to construct their multiple alignments.

1.16.2.4 454NewblerMetrics.txt

The consensusResults group of the 454NewblerMetrics file for cDNA projects displays output similar but not identical to those output by genomic projects in that it gives metrics for isogroups and isotigs in addition to the metrics for the large contigs and all contigs (Figure 40).

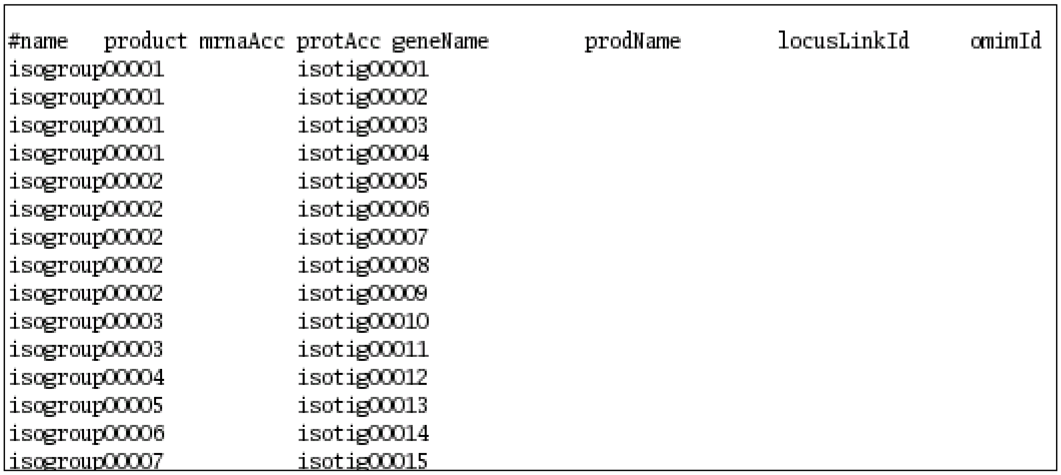


```
/*
**  Consensus results.
*/
consensusResults
{
	readStatus
	{
		numAlignedReads    = 278108, 93.02%, 93.02%;
		numAlignedBases    = 109418790, 92.46%, 92.24%;
		inferredReadError = 0.60%, 651794;

		numberAssembled = 259036, 86.64%, 86.64%;
		numberPartial   = 19045, 6.37%, 6.37%;
		numberSingleton = 12423, 4.16%, 4.16%;
		numberRepeat    = 44, 0.01%, 0.01%;
		numberOutlier   = 7753, 2.59%, 2.59%;
		numberTooShort  = 666, 0.22%, 0.22%;
	}

	isogroupMetrics
	{
		numberOfIsogroups     = 4370;
		avgContigCnt          = 1.2;
		largestContigCnt      = 147;
		numberWithOneContig   = 4148;

		avgIsotigCnt          = 1.1;
		largestIsotigCnt      = 21;
		numberWithOneIsotig   = 4156;
	}

	isotigMetrics
	{
		numberOfIsotigs       = 4645;
		avgContigCnt          = 1.1;
		largestContigCnt      = 10;
		numberWithOneContig   = 4148;

		numberOfBases     = 5886741;
		avgIsotigSize     = 1267;
		N50IsotigSize     = 1454;
		largestIsotigSize = 7154;
	}
```

Figure 40: 454NewblerMetrics file, consensusResults group example for a cDNA project.

1.16.2.5 454RefLink.txt

Another file specific to cDNA / transcriptome projects, 454RefLink.txt, is generated with a format like UCSC's refLink.txt files. This can be used as an annotation file for further mapping projects of the cDNA / transcriptome assembly products (isotigs and contigs).

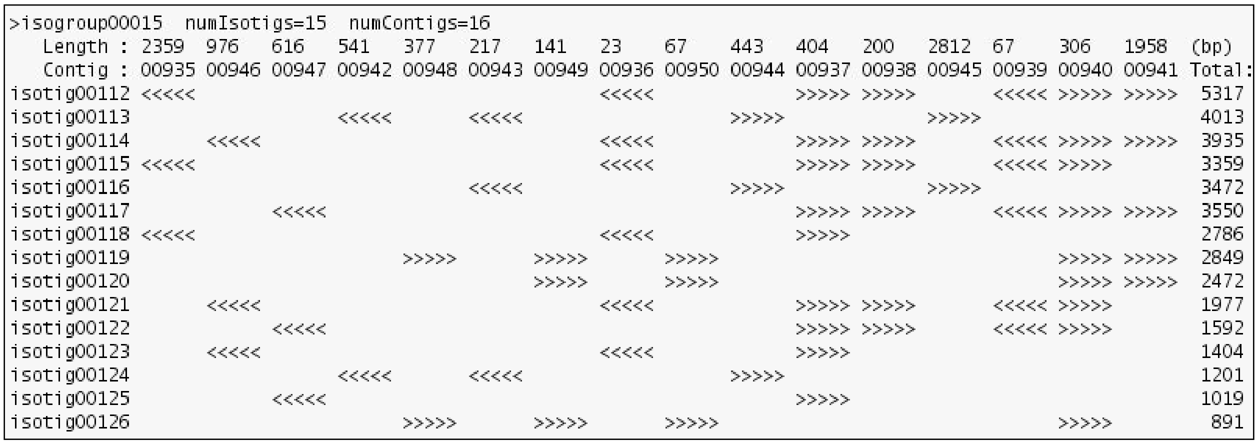


```
#name   product mrnaAcc protAcc geneName        prodName        locusLinkId     omimId
isogroup00001           isotig00001
isogroup00001           isotig00002
isogroup00001           isotig00003
isogroup00001           isotig00004
isogroup00002           isotig00005
isogroup00002           isotig00006
isogroup00002           isotig00007
isogroup00002           isotig00008
isogroup00002           isotig00009
isogroup00003           isotig00010
isogroup00003           isotig00011
isogroup00004           isotig00012
isogroup00005           isotig00013
isogroup00006           isotig00014
isogroup00007           isotig00015
```

Figure 41: 454RefLink.txt file example.

1.16.2.6 454IsotigsLayout.txt

Finally, the file 454IsotigsLayout.txt gives a visual representation of how the contigs are laid along each isotig in the isogroup:



```
>isogroup00015  numIsotigs=15  numContigs=16
   Length : 2359  976   616   541   377   217   141   23    67    443   404   200   2812  67    306   1958  (bp)
   Contig : 00935 00946 00947 00942 00948 00943 00949 00936 00950 00944 00937 00938 00945 00939 00940 00941 Total:
isotig00112 <<<<<                                     <<<<<             >>>>> >>>>>       <<<<< >>>>> >>>>>   5317
isotig00113                   <<<<<       <<<<<                   >>>>>             >>>>>                     4013
isotig00114       <<<<<                               <<<<<             >>>>> >>>>>       <<<<< >>>>> >>>>>   3935
isotig00115 <<<<<                                     <<<<<             >>>>> >>>>>       <<<<< >>>>>         3359
isotig00116                               <<<<<                   >>>>>             >>>>>                     3472
isotig00117             <<<<<                                           >>>>> >>>>>       <<<<< >>>>> >>>>>   3550
isotig00118 <<<<<                                     <<<<<             >>>>>                                 2786
isotig00119                         >>>>>       >>>>>       >>>>>                                 >>>>> >>>>>   2849
isotig00120                                     >>>>>       >>>>>                                 >>>>> >>>>>   2472
isotig00121       <<<<<                               <<<<<             >>>>> >>>>>       <<<<< >>>>>         1977
isotig00122             <<<<<                                           >>>>> >>>>>       <<<<< >>>>>         1592
isotig00123       <<<<<                               <<<<<             >>>>>                                 1404
isotig00124                   <<<<<       <<<<<                   >>>>>                                       1201
isotig00125             <<<<<                                           >>>>>                                 1019
isotig00126                         >>>>>       >>>>>       >>>>>                                 >>>>>          891
```

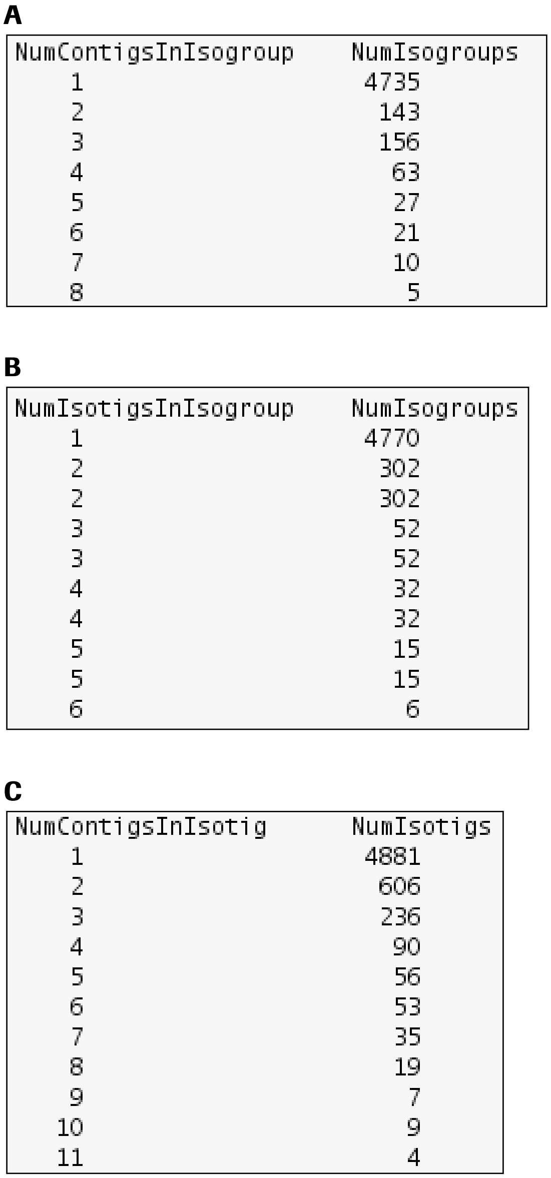
Figure 42: Partial view of upper portion of the 454IsotigsLayout.txt file

where ">>>>>" and "<<<<<" denote whether the particular contig was traversed from the 5' to the 3' end or vice - versa.

The lengths given in the top row are contig lengths, and the lengths on the right-most column are isotig lengths.

In addition, at the end of this file there are 5 histograms (Figure 43):

- counts of isogroup contigs
- counts of isogroup isotigs
- counts of isotig contigs
- lengths of contigs (binned into 100 bp bins)
- lengths of isotigs (binned into 100 bp bins)



A

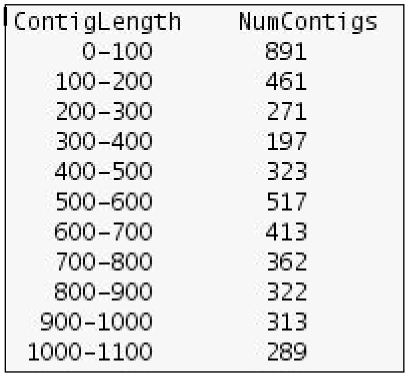
```
NumContigsInIsogroup    NumIsogroups
    1                     4735
    2                      143
    3                      156
    4                       63
    5                       27
    6                       21
    7                       10
    8                        5
```

B

```
NumIsotigsInIsogroup    NumIsogroups
    1                     4770
    2                      302
    2                      302
    3                       52
    3                       52
    4                       32
    4                       32
    5                       15
    5                       15
    6                        6
```

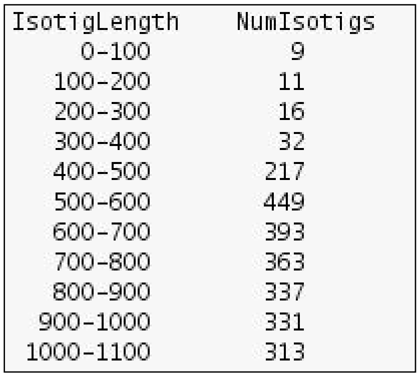
C

```
NumContigsInIsotig      NumIsotigs
    1                     4881
    2                      606
    3                      236
    4                       90
    5                       56
    6                       53
    7                       35
    8                       19
    9                        7
   10                        9
   11                        4
```



```
ContigLength    NumContigs
     0-100        891
   100-200        461
   200-300        271
   300-400        197
   400-500        323
   500-600        517
   600-700        413
   700-800        362
   800-900        322
  900-1000        313
 1000-1100        289
```

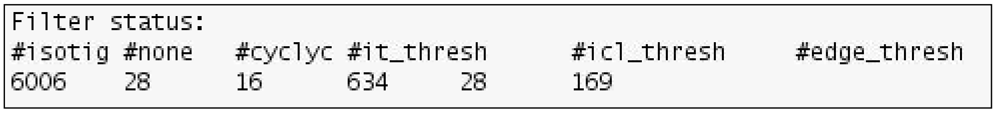
E



```
IsotigLength    NumIsotigs
     0-100          9
   100-200         11
   200-300         16
   300-400         32
   400-500        217
   500-600        449
   600-700        393
   700-800        363
   800-900        337
  900-1000        331
 1000-1100        313
```

Figure 43: Partial view of lower portion of the 454IsotigsLayout.txt file.

At the very end of the 454IsotigsLayout.txt file is information about how many regular isotigs are generated and how many non-traversed contigs remain. The status column gives the reason each contig was not traversed (thresholds, limits, cycles, *etc.*). For some contigs, the status can be "none", meaning they weren't reached at all because the traversal stopped before.



```
Filter status:
#isotig #none   #cyclyc #it_thresh      #icl_thresh     #edge_thresh
6006    28      16      634     28      169
```

Figure 44: The end of the 454IsotigsLayout.txt file.

2 GS Reference Mapper

2.1 Overview of the GS Reference Mapper

The GS Reference Mapper application aligns sequencing reads against a reference sequence consisting of one or more sequences or a GoldenPath genome, with or without associated annotations. The GS Reference Mapper software is an interactive application used to create mapping projects, add or remove reads from the project, specify reference sequences, annotations and other project parameters, run the mapping algorithms on the project data, and view the output produced by the mapping computations. The application can be accessed *via* a Graphical User Interface (GUI) or from a command line interface (CLI).

Input data can come from one or several regions of one or several runs of interest. Additional read data (for example from Sanger sequencing) can be imported from one or more external file formats, including FASTA and FASTQ. Mapping generates consensus sequences of the reads that align against the reference and also computes statistics for variations found in the reads, relative to the reference. Data are output to a variety of file formats, including FASTA, ACE, BAM or consed files.

The GS Reference Mapper application allows a user to:

- create mapping projects for genomic or cDNA reads
- add and remove read data sets from the project
- add or remove reference sequences and annotations
- specify mapping parameters
- specify special library preparations (such as the use of MIDs)
- run the mapping algorithms on the project data
- and view the output produced by the mapping computation

When the mapping algorithms run, the software performs the following operations:

- For each read, search for its best alignment to the reference sequence(s) (a read may align to multiple positions in the reference); this is done in 'nucleotide' space.
- Perform multiple alignments for the reads that align contiguously to the reference in order to form "contigs." From the contigs' multiple alignments, consensus basecall sequences are produced using the signals of the reads in the multiple alignments (performed in 'flowspace')
- Identify subsets of the reads that vary relative to the reference to form lists of putative variations (nucleotide differences and structural variants). For each putative variation, the reads supporting the variation will be in the subset.
- Evaluate these lists of putative variations to identify High-Confidence nucleotide differences (HCDiffs), structural variations (HCStructVars) and larger-scale structural rearrangements (HCStructRearrangements).
- Output the following information:
    - contig consensus sequence(s) and associated quality values
    - alignments of the reads to the reference
    - position-by-position metrics of the depth and consensus accuracy (quality values) for each position in the aligned reference
    - the positions and alignments of identified differences.

Read overlaps and multiple alignments are made in 'nucleotide' space while the consensus basecalling and quality value determination for contigs are performed in 'flowspace'. Work in flowspace allows the averaging of processed flow signals (a continuous variable) at each nucleotide flow of the sequencing run(s) and allows the use of information from the "negative flows", *i.e.* flows where no nucleotide incorporation is detected. The use of flowspace in determining the properties of the consensus sequence results in an improved accuracy for the final basecalls.

- The GS Reference Mapper application can be run from the attendant PC or a datarig when using the GS Junior Instrument.
- The GS Reference Mapper application is not available on the GS FLX+ Instrument and must be run on a datarig for users of this instrument.

The GS Reference Mapper allows users to create, modify, and run mapping projects. Both the GUI and command line interface (CLI) provide this functionality. Projects may be setup to map all reads at once. Alternatively, incremental operation allows additional reads to be added to an existing mapping project. Results appear as output files using either the GUI or the CLI. The GUI provides a graphical interface to view many of the results from the mapping computation whether the mapping was performed using the GUI or the CLI.

The GS Reference Mapper application uses a folder on the file system to hold the mapping project information (whether the mapping is performed through the GUI application or through the newMapping and related commands) and to hold the output files generated during the mapping computation. The contents of this folder and the names of the files generated by the application are the same for any mapping project or output folder.

The operation of the GUI is described in several of the subsequent sections. A description of the CLI is then presented followed by a discussion of transcriptome mapping. Finally, output files produced by the GS Reference Mapper are described.

2.2 GS Reference Mapper GUI

Mapping can be performed and the results can be viewed using the Graphical User Interface application, gsMapper, described in the next few sections. The application includes graphical interfaces to:

- create new mapping projects
- open existing mapping projects
- Add/remove reads to/from mapping projects
- Add/remove references to/from mapping projects
- Associate annotations with the reference(s)
- Modify mapping project and output parameters
- carry out a mapping computation on a project
- view the results of a completed mapping
- view the progress and logging information of a mapping computation

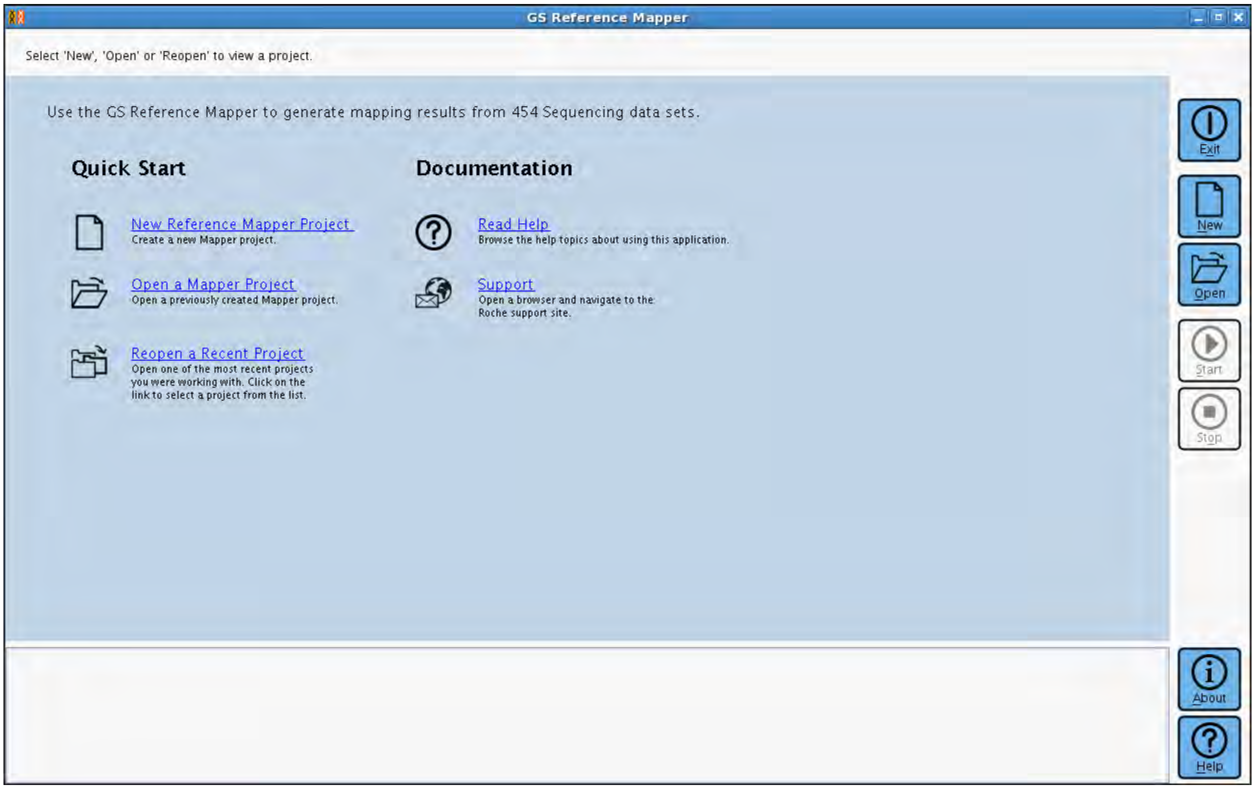
2.3 Launching the GS Reference Mapper GUI

The GS Reference Mapper GUI application is launched by double clicking on its desktop icon.

Alternatively, the following command can be used to launch the GS Reference Mapper GUI application from a Linux terminal window on a datarig where the data analysis software package is installed:

```
gsMapper
```

Once the GUI is launched (Figure 45), a user can open an existing project or create a new project.



Select 'New', 'Open' or 'Reopen' to view a project.

Use the GS Reference Mapper to generate mapping results from 454 Sequencing data sets.

Quick Start

New Reference Mapper Project
Create a new Mapper project.

Open a Mapper Project
Open a previously created Mapper project.

Reopen a Recent Project
Open one of the most recent projects you were working with. Click on the link to select a project from the list.

Documentation

Read Help
Browse the help topics about using this application.

Support
Open a browser and navigate to the Roche support site.

Figure 45: Initial Interface of the gsMapper.

2.3.1 The Main Buttons, Status Area and Progress Box

Seven main buttons are always visible on the toolbar, along the right-hand side of the GS Reference Mapper's main window:

- The **Exit** button closes the GS Reference Mapper application
- The **New** button allows you to create a new mapping project
- The **Open** button allows you to open an existing mapping project
- The **Start** button begins the mapping (computation) of an open mapping project
- The **Stop** button halts the execution of an ongoing mapping computation
- The **About** button shows the GS Reference Mapper splash screen
- The **Help** button opens the GS Reference Mapper section of the software manual.

The top of the main GS Reference Mapper window contains a status area that displays a command hint, while the bottom panel is a progress box which remains blank until a mapping project has been opened.

The progress box at the bottom of the window can be resized by dragging its top frame separator. If the field is not visible, it is probably simply collapsed; you can view it by dragging up the edge of the collapsed frame.

2.3.2 The Quick Start and Documentation Text Links

There are 3 text links under the **Quick Start** column:

- The **New Mapping Project** text link creates a new mapping project. It performs the same action as the **New** button on the right side of the window.
- The **Open a Mapping Project** text link opens an existing project. It performs the same action as the **Open** button on the right side of the window.
- The **Reopen Recent Project** text link displays a dialog from which you can open a project on which you worked recently.

There are 2 text links under the **Documentation** column:

- The **Read Help** text link opens an electronic version of the software manual , *i.e.* it performs the same action as the **Help** button on the right side of the window.
- The **Support** text link opens the default internet browser and navigates to the Roche support site.

2.4 Opening a Project

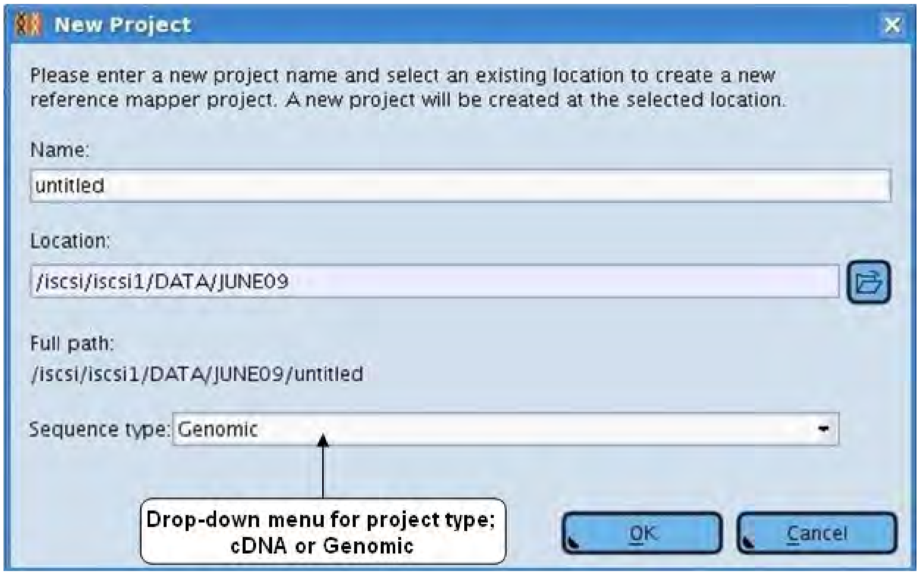
2.4.1 Creating a New Project

To create a new mapping project, either click on the **New Project** button in the right toolbar, or click on the "New Mapping Project" text button in the **Quick Start** column. This displays a dialog (Figure 46) in which you can specify the name and directory location for a new project.

Type a name for the new project in the **Name** text field. To specify the project location, either type the full path in the **Location** text field, or click on the **Open Project** button to the right of the text field and use the "Select Project Location" window (not shown) to navigate to the directory where you want to create the new project. The **Full Path** field changes as you update the **Name** and **Location** fields. When you are satisfied with the location and name for the new project, click on the **OK** button. You can save a project by clicking on the Yes button to the **Save** prompt.

You may save the project to your hard drive either by using the Exit button to exit the GS Mapper application (you will be prompted to save before the application actually exits) or by adding read data and running the project by clicking the Start button (*i.e.*, compute the mapping), in which case the project will automatically be saved prior to computation, as described in section 2.8.1, below.

The New Project Dialog contains a drop-down menu (Figure 46) to specify the **Sequence type** indicating the type of DNA library sequenced, cDNA or Genomic. This choice of Sequence type determines many of the parameters the user will be allowed to modify in the project. Once a project is created the type cannot be changed.


New Project

Please enter a new project name and select an existing location to create a new reference mapper project. A new project will be created at the selected location.

Name: untitled

Location: /iscsi/iscsi1/DATA/JUNE09

Full path: /iscsi/iscsi1/DATA/JUNE09/untitled

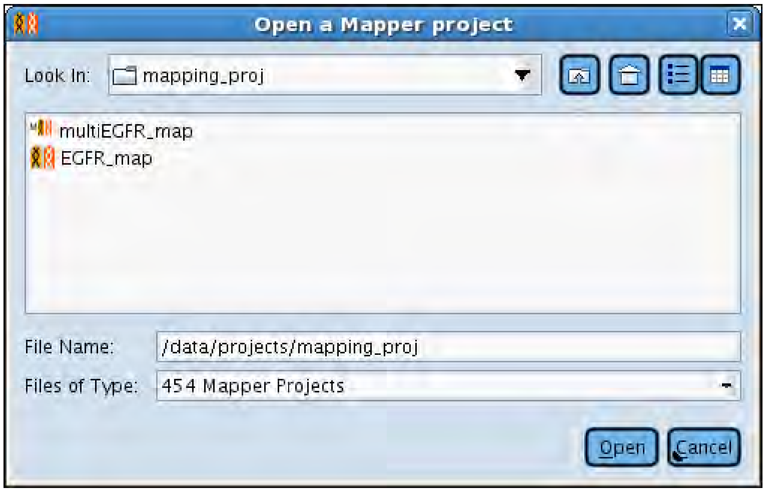
Sequence type: Genomic

Drop-down menu for project type; cDNA or Genomic

Figure 46: New Project dialog used to specify Sequence type.

2.4.2 Open an Existing Project

To open an existing mapping project, either click on the **Open Project** button in the right toolbar, or click on the "Open a Mapping Project" text button in the **Quick Start** column. This displays a dialog (Figure 47) in which you can select a name and specify the directory location of the project to be opened. By default, only 454 System Mapping Projects will be displayed. You may choose to display all files by selecting "All Files" from the **"Files of Type"** dropdown menu.


Open a Mapper project

Look In: mapping_proj

multiEGFR_map
EGFR_map

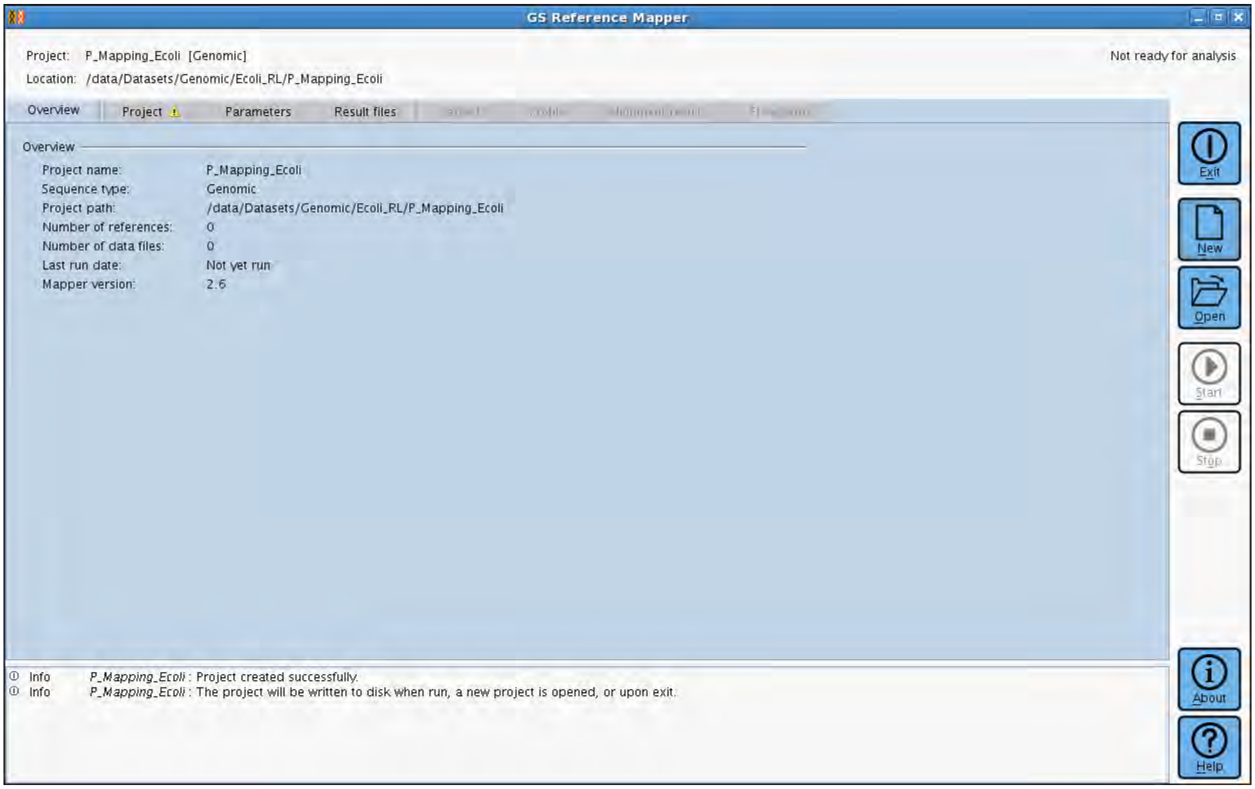
File Name: /data/projects/mapping_proj

Files of Type: 454 Mapper Projects

Figure 47: Open a Mapping Project dialog. Note the distinctive icons for a multiplex project (the first icon, see Section 2.15) *vs.* an ordinary mapping project (the second icon).

2.5 View Project Summary with the Overview Tab

When a project opens, the Overview Tab is displayed (Figure 48). Other tabs can be selected with the mouse. Some tabs may remain inactive (grayed-out), until the type of information they require is available for the project. The Overview Tab's ***Project Summary Information*** area indicates the Sequence type along with other information that applies to the overall project. The data displayed on this tab is updated as new information becomes available, which occurs when data and reference files are added or removed from the project and when the project computation completes.



Project: P_Mapping_Ecoli [Genomic]
Location: /data/Datasets/Genomic/Ecoli_RL/P_Mapping_Ecoli

Not ready for analysis

Overview

Project name: P_Mapping_Ecoli
Sequence type: Genomic
Project path: /data/Datasets/Genomic/Ecoli_RL/P_Mapping_Ecoli
Number of references: 0
Number of data files: 0
Last run date: Not yet run
Mapper version: 2.6

```
Info    P_Mapping_Ecoli : Project created successfully.
Info    P_Mapping_Ecoli : The project will be written to disk when run, a new project is opened, or upon exit.
```

Figure 48: gsMapper Overview Tab.

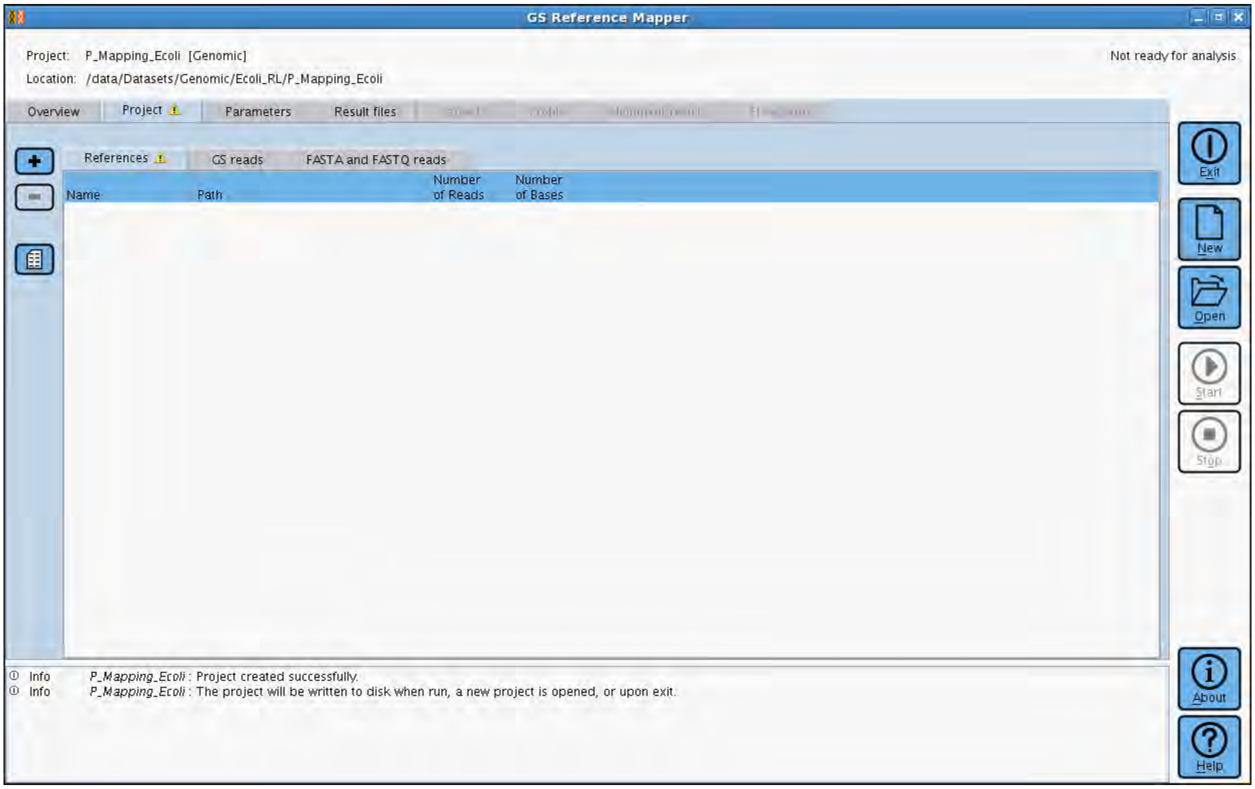
When a project is first created, the ***Computation status message*** in the upper right hand corner of the application window indicates "Not ready for analysis" and the Start button is disabled because at least one Read Data file and one Reference file must be added to the project before a Mapping computation can be performed. Other errors in input parameters can also cause this message to appear. Once all such errors have been corrected and at least one Read Data file and one Reference file are added to the project, the message will change to "Ready for analysis" and the Start button will become active.

2.6 Add/Remove Read Data and References with the Project Tab

Read Data and reference files are added to a project on the Project tab of the main application window (Figure 49). There are two sub-tabs for adding read data: one for adding GS reads ("GS Reads" sub-tab) and one for adding non-

GS reads, such as Sanger reads, in FASTA/FASTQ format ("FASTA and FASTQ Reads" sub-tab). In addition, the References sub-tab is used to add reference files to a project.

The **Start** button in the toolbar, which is used to start a mapping computation, remains disabled until at least one Read Data file (either GS read or FASTA/FASTQ read) and one Reference File have been added to the project.



Project: P_Mapping_Ecoli [Genomic]
Location: /data/Datasets/Genomic/Ecoli_RL/P_Mapping_Ecoli

Not ready for analysis

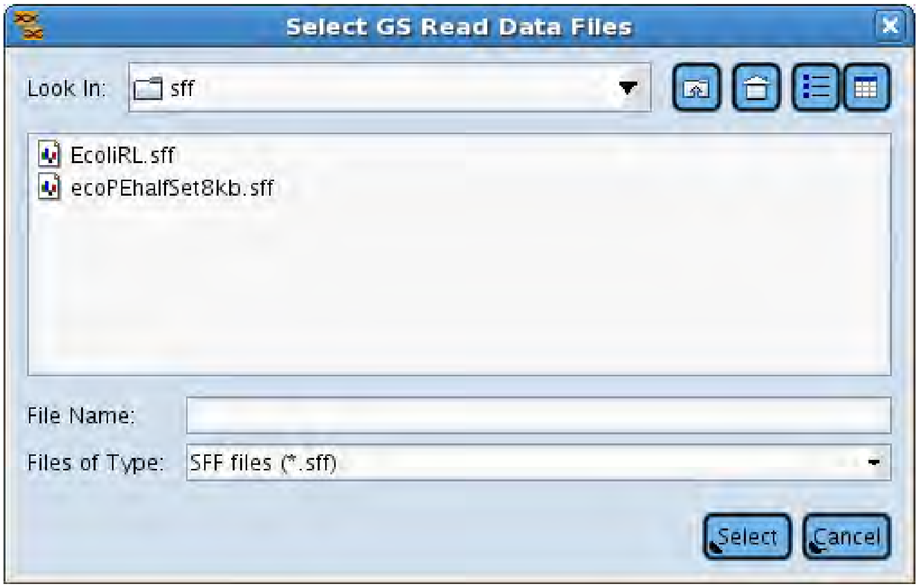
Name	Path	Number of Reads	Number of Bases

```
Info    P_Mapping_Ecoli : Project created successfully.
Info    P_Mapping_Ecoli : The project will be written to disk when run, a new project is opened, or upon exit.
```

Figure 49: Project Tab of the gsMapper (GS Reads sub-tab).

2.6.1 GS Reads, References, FASTA and FASTQ Reads Sub-Tabs

To add GS read data sets to a project, click on the GS reads sub-tab, then click the **Add** button ⊞ to open the "Select GS Read Files" window (Figure 50). This dialog window allows you to navigate and see the available read files. Read files already in the project are visible but grayed out. You can select multiple files in this window using the ctrl or shift keys while clicking with the mouse.



Select GS Read Data Files

Look In: sff

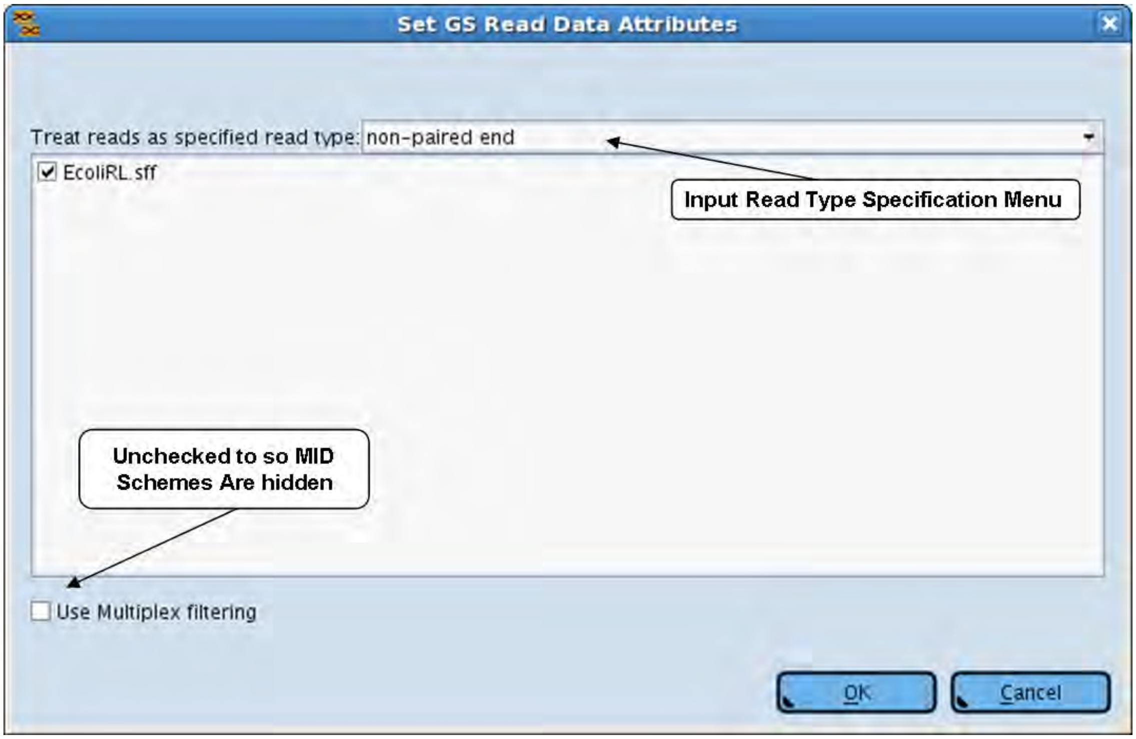
EcoliRL.sff
ecoPEhalfSet8kb.sff

File Name:

Files of Type: SFF files (*.sff)

Figure 50: Select GS Read Data dialog.

Once the desired file(s) are selected, the "Set GS Read Data Attributes" dialog window opens. From this window (Figure 51) the GS Read files can be explicitly specified as paired end or non-paired end or the user can invoke auto-detection (the default setting) using the **Read Type Specification** dropdown. When the read type is known, it is advisable to specify it directly rather than use auto-detect as the auto-detect feature, on rare occasion, may fail to detect a paired end file.



Set GS Read Data Attributes

Treat reads as specified read type: non-paired end

☑ EcoliRL.sff

□ Use Multiplex filtering

Input Read Type Specification Menu

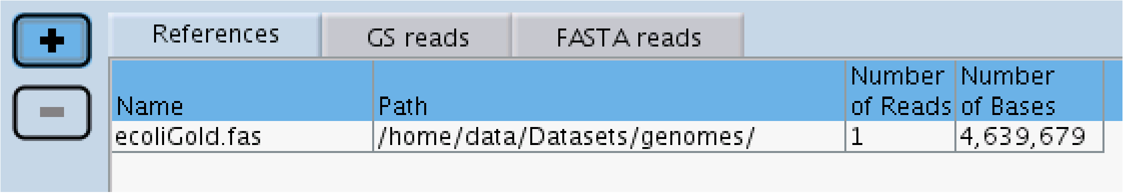
Unchecked to so MID Schemes Are hidden

Figure 51: Set GS Read Data Attributes dialog.

You can add any number of Read Data files to a project using the Add button, navigating to a folder, then selecting the files needed from the folder. You can even add multiple files that share the same name into a project, as long as they have different path locations in the file system (*e.g.* "/dir1/path1/reads.sff" and "/dir2/otherpath2/reads.sff"). In such a case, both files will be added to the Read Data list and displayed using the same name. To see the path to a file listed in the Read Data area of the main window (and the file's last modification date), hover the mouse over the filename of interest and a tooltip containing the file's path will be displayed. Files that have failed validation will have a red X left of the file name as an indication of failure. Pause the mouse cursor over the red X to bring up a tooltip explaining the problem encountered.

The FASTA and FASTQ Reads sub-tab is functionally similar to the GS Reads sub-tab for all project types. The only difference is that when a directory containing FASTA/FASTQ files is selected, the software examines the files in that directory to determine which files are FASTA/FASTQ files. This can take some time if there are many files in the directory, so a progress bar is displayed to show the progress of the search. See the Overview Manual, Section 2.2.2 for a discussion of the FASTQ format.

The References sub-tab is functionally similar to the Reads sub-tabs for both genomic and cDNA mapping projects. The difference is that instead of displaying all the available read data files when a directory selected, the software displays all recognized FASTA files. You may select one or more of these files. Reference files contain the DNA sequence(s) against which the reads in the Read Data files will be aligned. Once the mapping computation is run, the software determines the total number of reference sequences and bases in the file and includes this information in the two rightmost columns of the **References** tab (Figure 52).

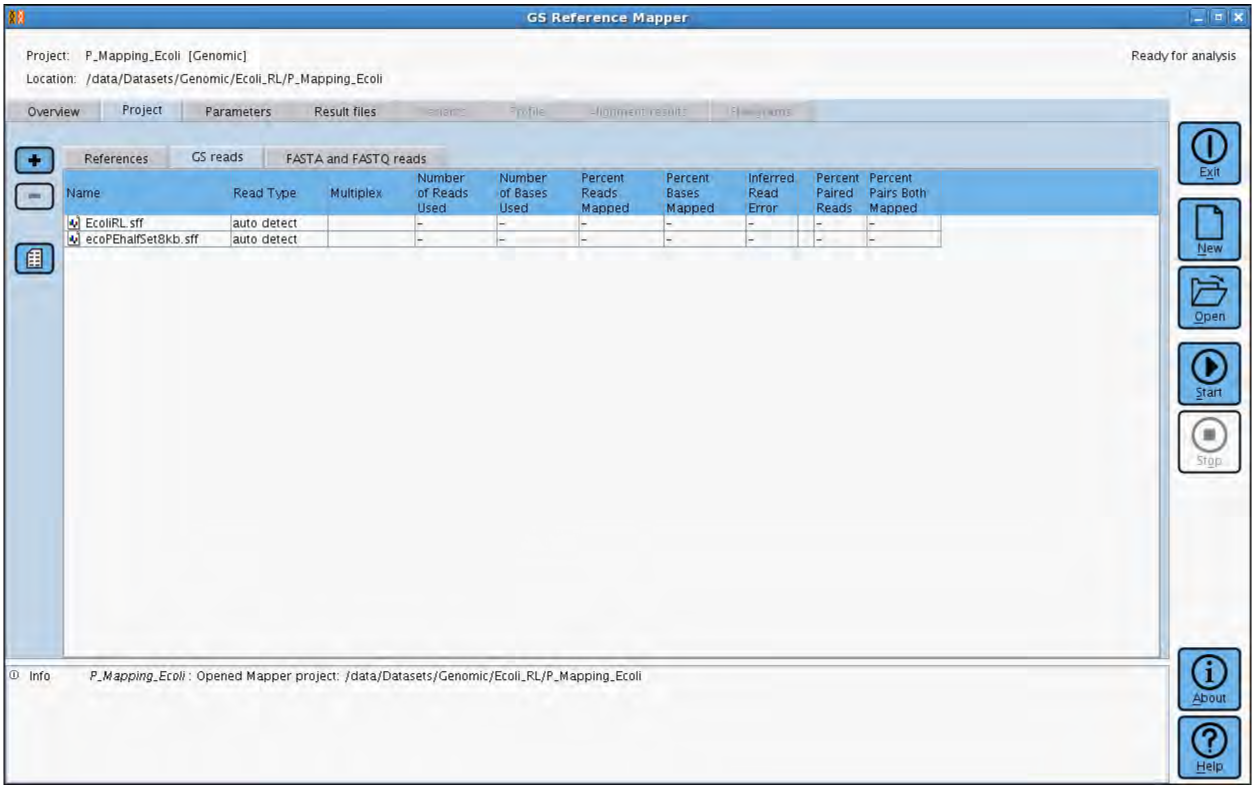


Name	Path	Number of Reads	Number of Bases
ecoliGold.fas	/home/data/Datasets/genomes/	1	4,639,679

Figure 52: gsMapper Project Tab References sub-tab.

Although in many cases the file name extension for FASTA and FASTQ files may be **.fna** and **.fastq**, respectively, there is no standardized extension for such files. If you cannot see your FASTA/FASTQ file, try setting the “Files of Type” field to “All Files”.

Except for the Name column and the Multiplex column (which contains comma-delimited lists of the MIDs associated with the file; see Section 2.6.3, below), all columns with run statistics are initially filled with dashes (Figure 53). These data are updated in the table each time the project runs to completion. For a project that has already been through a mapping computation, summary statistics relating to the usage of the reads in the mapping process are also listed, as shown in Figure 61. On the left hand side of the reads table is a data capture buttons that can be used to capture the reads table data in csv, tab-delimited or plain or text formats.



Project: P_Mapping_Ecoli [Genomic]
Location: /data/Datasets/Genomic/Ecoli_RL/P_Mapping_Ecoli

Name	Read Type	Multiplex	Number of Reads Used	Number of Bases Used	Percent Reads Mapped	Percent Bases Mapped	Inferred Read Error	Percent Paired Reads	Percent Pairs Both Mapped
EcoliRL.sff	auto detect		–	–	–	–	–	–	–
ecoPEhalfSet8kb.sff	auto detect		–	–	–	–	–	–	–

Info P_Mapping_Ecoli : Opened Mapper project: /data/Datasets/Genomic/Ecoli_RL/P_Mapping_Ecoli

Figure 53: Project Tab of the gsMapper (GS Reads sub-tab).

2.6.2 Removing Read or Reference Data from the project Sub-Tabs

To delete read of reference data from a project, click on the sub-tab that lists the reads or reference data, select the data file(s) to be removed, then click the **Remove** button − .

The **Remove** button removes the selected Reference and/or Read Data files from the list, but does not immediately remove the file(s) from the project's sff sub-directory or from any computed mapping results. Actual removal of the files from the project (and of the reads they contain from existing alignments), occurs only when the mapping is re-computed (see section2.8.2).

2.6.3 Specifying Multiplex Identifiers (MIDs)

When the Use Multiplex Filtering checkbox is selected, special controls for MID filtering are displayed in the bottom half of the "Set GS Read Data Attributes" window (Figure 54). Multiplex Identifiers (MIDs) allow you to design an experiment whereby multiple libraries are prepared using distinct MID tags and sequenced together, on the same PTP device. On the GS FLX+ system, multiple libraries can be sequenced together in the same region of a PTP device. (Use of MIDs can greatly improve the workflow and cost effectiveness of your experiment.)

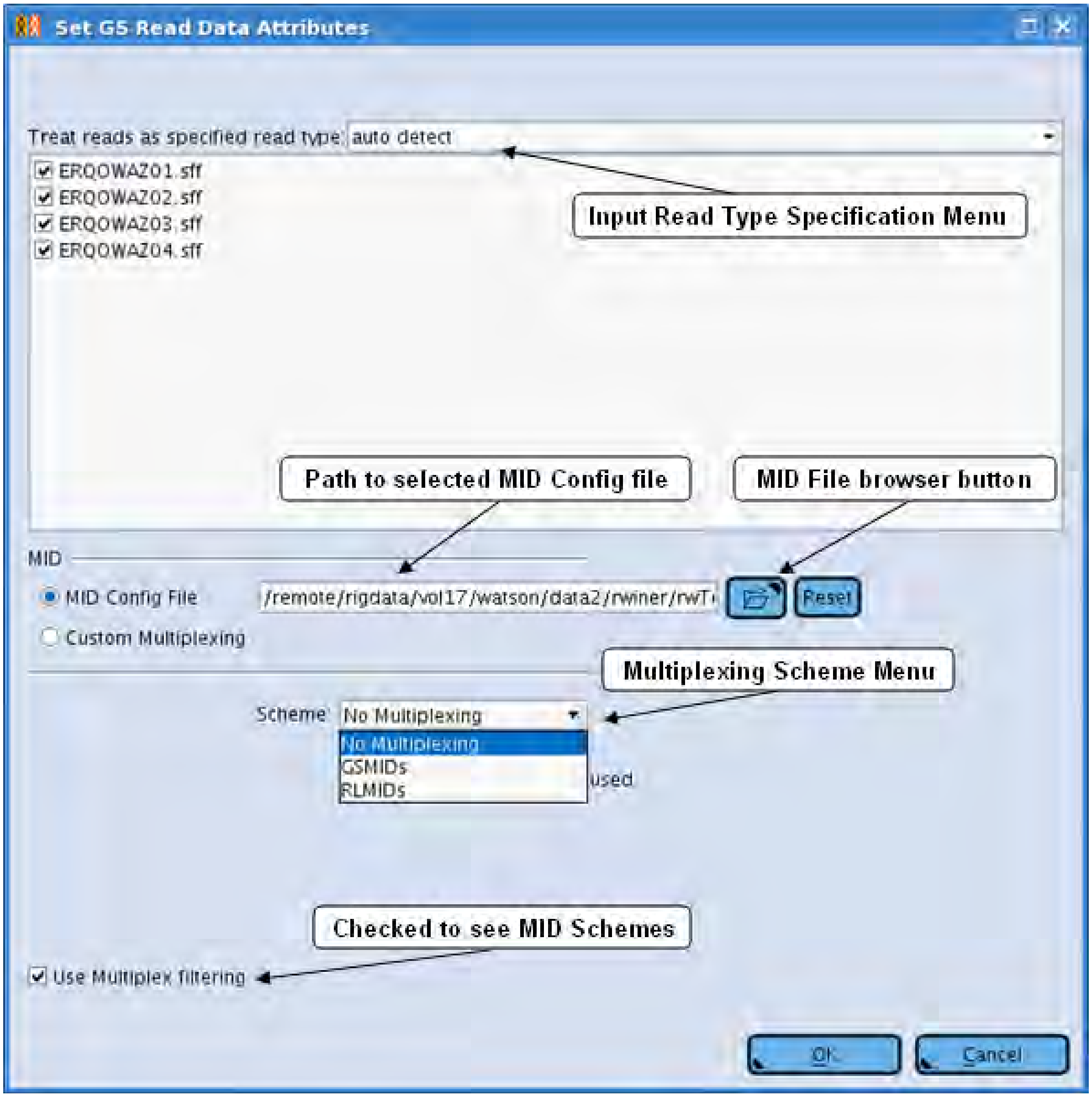
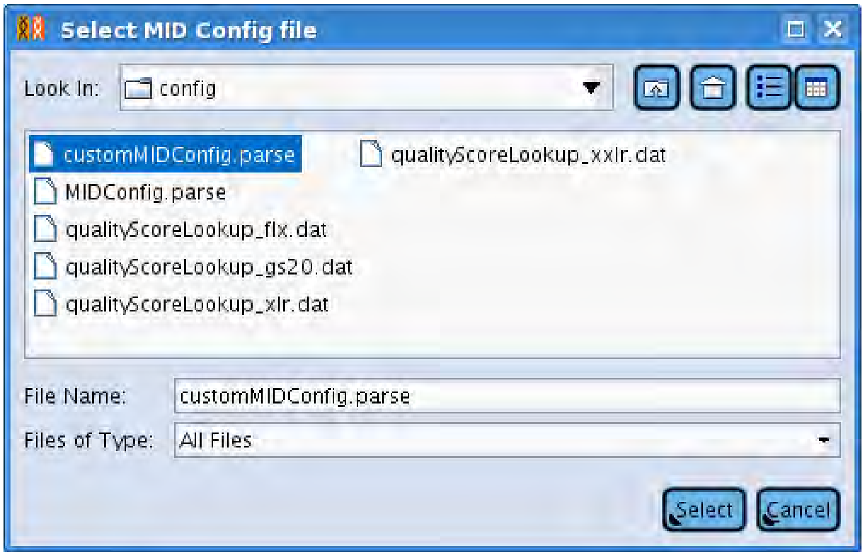

Set GS Read Data Attributes
Treat reads as specified read type: auto detect
☑ ERQOWAZ01.sff
☑ ERQOWAZ02.sff
☑ ERQOWAZ03.sff
☑ ERQOWAZ04.sff
Input Read Type Specification Menu
Path to selected MID Config file
MID File browser button
MID
● MID Config File
/remote/rigdata/vol17/watson/data2/rwiner/rwT
Reset
○ Custom Multiplexing
Multiplexing Scheme Menu
Scheme
No Multiplexing
No Multiplexing
GSMIDs
RLMIDs
used
Checked to see MID Schemes
☑ Use Multiplex filtering
OK
Cancel

Figure 54: Set GS Read Data Attributes dialog with the MID Scheme options expanded.

The MID controls on this window allow you to filter the reads in the selected Read Data files for inclusion in the Mapping project. Note, this filtering operation does not produce a different Mapping for each specified MID. Rather, the reads of the SFF files are scanned for the presence of MIDs, and the reads containing the selected MIDs are made eligible for mapping in the project. The selected MID filtering criteria are associated with the sff files when OK is pressed (Figure 54). Any reads from these file(s) without the specified MIDs will be ignored. To produce independent mappings for different MIDs (or sets of MIDs), you must create independent mapping projects for each of the desired MID subsets of the data.

MID filtering information is contained in 'MID schemes'. Available MID schemes may, in turn, be found in MID configuration files. The default MID configuration file supplied with the GS Reference Mapper software is MIDConfig.parse; the MID schemes provided in this file are 'No Multiplexing', 'GSMIDs' for use of the original Genome Sequencer MID sets, and 'RLMIDs' for the Rapid Library Preparation Kit MIDs. An alternate configuration file can be specified by pressing the "browse for MID configuration file" button to the right of the "MID Config File" textbox (Figure 54). The "Select MID Config file" dialog (Figure 55) will then appear and can be used to navigate to the desired location. Choose the file to be used as the configuration file for the sff file(s) being imported and click the Select button. The MID schemes specified in the file selected will then populate the Scheme dropdown. Press the "Reset" button to set MIDConfig.parse (the default MID configuration file) as the configuration file to be used in the file import. 'Custom Multiplexing' can be selected if custom MIDs were used.


Select MID Config file
customMIDConfig.parse
MIDConfig.parse
qualityScoreLookup_flx.dat
qualityScoreLookup_gs20.dat
qualityScoreLookup_xlr.dat
qualityScoreLookup_xxlr.dat
Select
Cancel

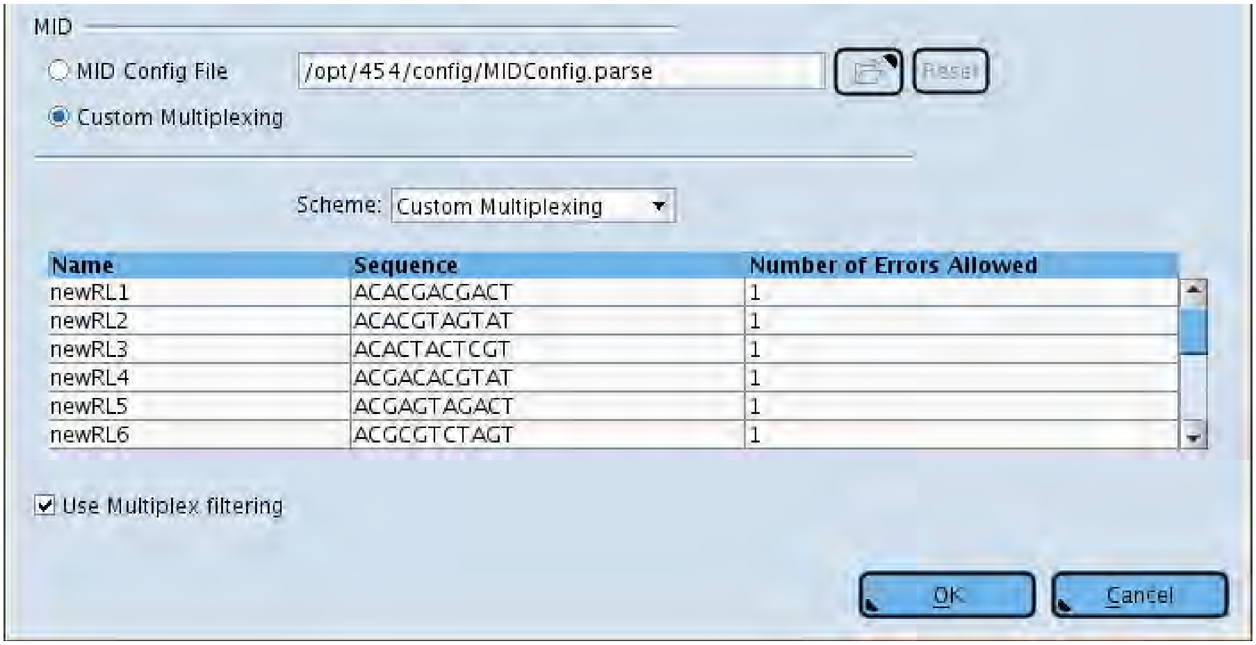
Look In: config

File Name: customMIDConfig.parse

Files of Type: All Files

Figure 55: Select MID Config file browser.

By default, no MID/Multiplexing scheme is associated with Read Data files. To use MIDs, first, specify the MID Config File, then select the desired MID Scheme (Figure 56) using the **Scheme** drop down menu. When an MID scheme is selected, a table of the MIDs present in that scheme is displayed, as shown in Figure 56. Use the checkboxes on the left to select or deselect the MIDs to be included in the mapping. Shift-clicking selects or deselects all MIDs in the scheme at once. The names, sequences and error limits of the MIDs selected will be used to filter the reads (from the Read Data sets selected in the top part of the window). The result is that only the reads that contain the selected MIDs, within the number of errors specified in the scheme, will be included in the project.



MID

○ MID Config File /opt/454/config/MIDConfig.parse

● Custom Multiplexing

Scheme: Custom Multiplexing ▾

Name	Sequence	Number of Errors Allowed
newRL1	ACACGACGACT	1
newRL2	ACACGTAGTAT	1
newRL3	ACACTACTCGT	1
newRL4	ACGACACGTAT	1
newRL5	ACGAGTAGACT	1
newRL6	ACGCGTCTAGT	1

☑ Use Multiplex filtering

Figure 56: Set GS Read Data Attributes dialog with the RLMIDs Scheme active.

For more information about MIDs, see Section 4.5.

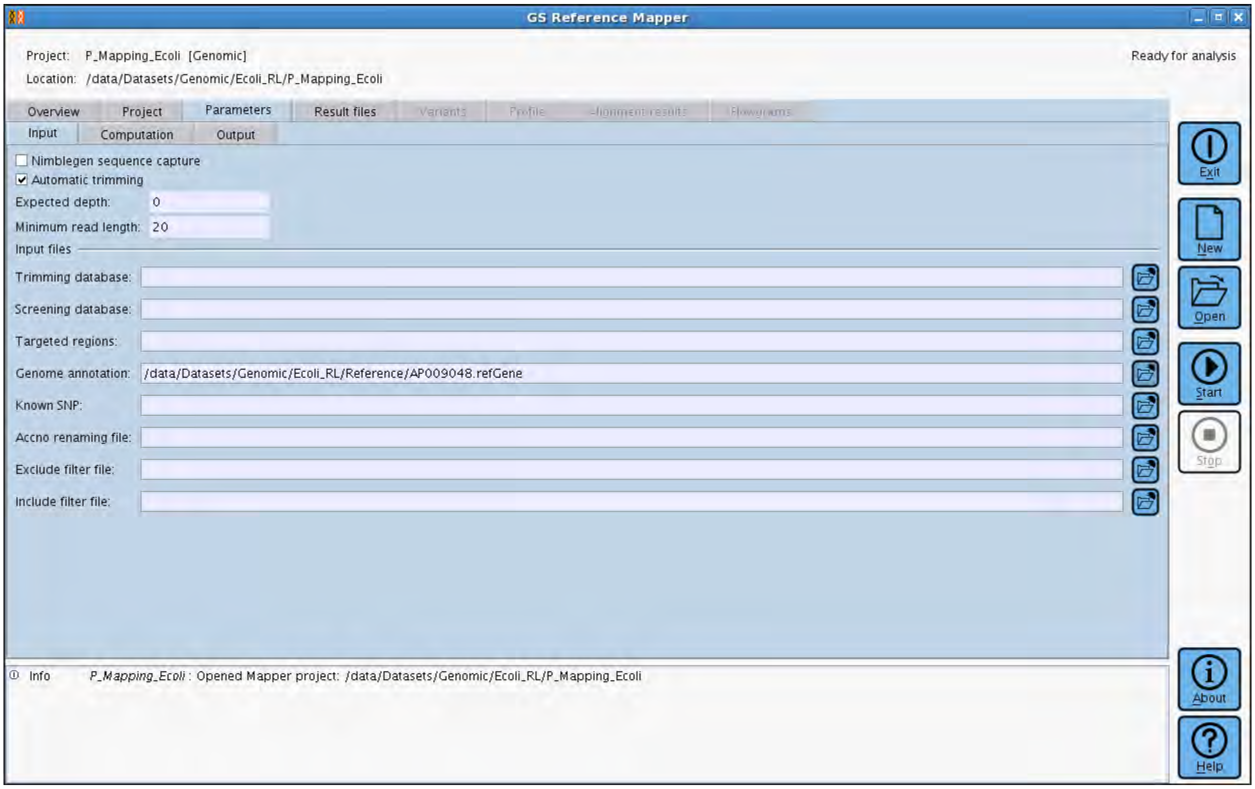
Once the MID scheme has been specified, click the **OK** button to add the selected data files (with MID filtration information) to the list of GS read data files (see Figure 54).

In certain circumstances, you may want to apply a different MID selection to different Read Data sets, within the same project. For example, you may have multiple libraries each with a different strain of the same species, made with different MIDs but sequenced in the same run, that you want to map together. In this case, select the Read Data set(s) that need to be filtered with any given MID (or combination of MIDs), and click **OK** to add the MID-filtered data file(s) to the project. Back on the GS Reads sub-tab of the GS Reference Mapper window (with the GS Reads sub-tab active, see Figure 53), click the **Add** button (+) again, and select other Read Data sets, to be filtered with different MIDs.

Incorrect MIDs may be assigned: In some cases, the MID(s) you select may be within the allowed number of errors of another MID, resulting in incorrect MID assignment. If this occurs, you should use the sfffile command (with the -s option) to split the reads with different MIDs into different files. The sfffile command, described in section 3.1, considers all MIDs in the MIDConfig.parse file, and will assign the MID(s) more accurately. The default MIDs and RLMIDs present in the MIDConfig.parse differ from each other by more than the allowed number of errors so this should occur very infrequently.

2.7 Customize Project with the Parameters Tab

The Parameters Tab is organized as three sub-tabs: Input, Computation and Output (Figure 57). The input parameters that can be specified are different for the Genomic and cDNA sequence types. Each sequence type will have its own set of parameters under the project Input sub-tab.



Project: P_Mapping_Ecoli [Genomic]
Location: /data/Datasets/Genomic/Ecoli_RL/P_Mapping_Ecoli
Ready for analysis

☐ Nimblegen sequence capture
☑ Automatic trimming
Expected depth: 0
Minimum read length: 20

Trimming database:
Screening database:
Targeted regions:
Genome annotation: /data/Datasets/Genomic/Ecoli_RL/Reference/AP009048.refGene
Known SNP:
Accno renaming file:
Exclude filter file:
Include filter file:

Info P_Mapping_Ecoli : Opened Mapper project: /data/Datasets/Genomic/Ecoli_RL/P_Mapping_Ecoli

Figure 57: gsMapper Parameters tab, Input sub-tab for Genomic projects.

For a new project, the mapping parameters are initially set to their defaults, as shown. If an invalid value is entered into any of these fields, an error indicator (red "X") appears in the lower left corner of the field as well as on the tab heading, and the "Not ready for analysis" warning in the upper right hand corner of the screen. Pausing the mouse over the error indicator reveals a tooltip indicating the allowed values for the parameter (see Section 2.14).

While default parameter values and settings are appropriate for most genomic and cDNA mapping projects, you may find some options useful for specific experiments.

2.7.1 Genomic Project Input Sub-Tab

The Input sub-tab for genomic projects is shown on Figure 57 (for cDNA projects, see Section 2.7.4). It allows you to adjust the following settings:

- When the **NimbleGen sequence capture** (**-n | -nimblegen**) checkbox is checked, the read data will automatically be primer-trimmed assuming NimbleGen Sequence Capture adapters are present at the beginning of the reads (this option defaults to unchecked). If the NimbleGen gSel protocol has been used, this option must be checked. There is no need to use this option with NimbleGen's optimized protocol for the GS FLX Titanium chemistry.

> NOTE on trimming in general: new reads added to a project are trimmed when the mapping computation is performed. The trimming is performed using the parameter settings on the Project Input sub-tab at the time the computation is run. Reads for which the mapping computation has already been run are not re-trimmed.

- If the **Automatic trimming** (**-trim**) checkbox is checked, the ends of new read files added to the project will be trimmed the next time a mapping computation is performed.
- **Expected depth** (**-e**) allows you to specify the number of reads expected to uniquely map to a single position of the reference genome. For high-depth mapping (where the expected coverage of a position in the genome could reach hundreds or thousands of reads), this option can be used to prevent low-frequency variations from being reported. If this option isn't used, many variations that are actually due to sequencing errors may be reported. The default value for this option is 0. A value of 0 or greater is allowed. If this option is set to a very low value (below 25), any variations that appear in only two reads may be output to the 454AllDiffs.txt file.
- **The Minimum read length** (**-minlen**) option can be used to set the minimum read length used in mapping computations (default 20 bp). See Section 2.18.1.10 for details on the handling of reads under 50 bp.

The **input files** section of the tab allows you to specify the locations of several files that may be used in the mapping.

- An optional **Trimming database** (**-v | -vt**) is used to trim the ends of input reads (for cloning vectors, primers, adapters or other end sequences). Specify the path to a FASTA file of sequences to be used for this trimming (see Section 4.8).
- To use the **Screening database** (**-vs**) option, set the path to a FASTA file of sequences to be used to screen the input reads for contaminants. A read that almost completely aligns against a sequence in the screening database is removed so that it is not used in the computation; if at least 15 bases of a read do not align to the screening sequence, no action is taken (see Section 4.8).
- The **Targeted regions** (**-reg**) option can be used to direct the GS Reference Mapper to only report output for regions of the reference you specify and the reads / contigs that overlap those regions. The reads are mapped against the complete reference, but the output files will only report on the specified regions. The typical use for this option may is with a SeqCap project. See Section 4.13.10 for more details on target regions and extended target regions. The string value can be
  - a "accno:#-#" string, *e.g.* "chr9:10549-30594" specifies bases 10549 to 30594 of the chr9 reference sequence.
  - the path to a file containing lines of "accno:#-#" strings (one per line).
  - the path to a GFF file created during a NimbleGen Sequence Capture experiment, describing the "primary_target_region" regions of the experiment. See appendix 4.10.1 for details of the GFF file format.
- If you are performing a mapping against an annotated reference, you may have to specify the locations of the annotation files (unless a GoldenPath reference is being used).
- To use the **Genome annotation** (**-annot**) option, enter the path to an annotation file describing the gene/coding-region annotations for the reference sequences, so that the variation detection (AllDiffs, HCDiffs, and Structural Variations) can report gene names and protein translations of any identified variations in gene regions. The format of this file must match that of the GoldenPath "refGene.txt" file. See appendix 4.10.2 for details of the refGene.txt file format.
- The **Known SNP** (**-snp**) option is used to specify the path to an annotation file describing the known SNP information for the reference sequences, so that the variation detection (HCDiffs tab) can link identified variations to the known SNP information. The format of this file must match that of the GoldenPath snp130.txt file. See appendix 4.10.3 for details of the snp130.txt file format.
- The **Accno renaming file** (**-accno**) option can be used to specify a renaming file to incorporate gene and transcript descriptions into the output files and/or to provide more meaningful names for genes and transcripts. For a description of the renaming file see Section 4.7.

The software does three things if no Genome annotations or known SNP paths are specified:

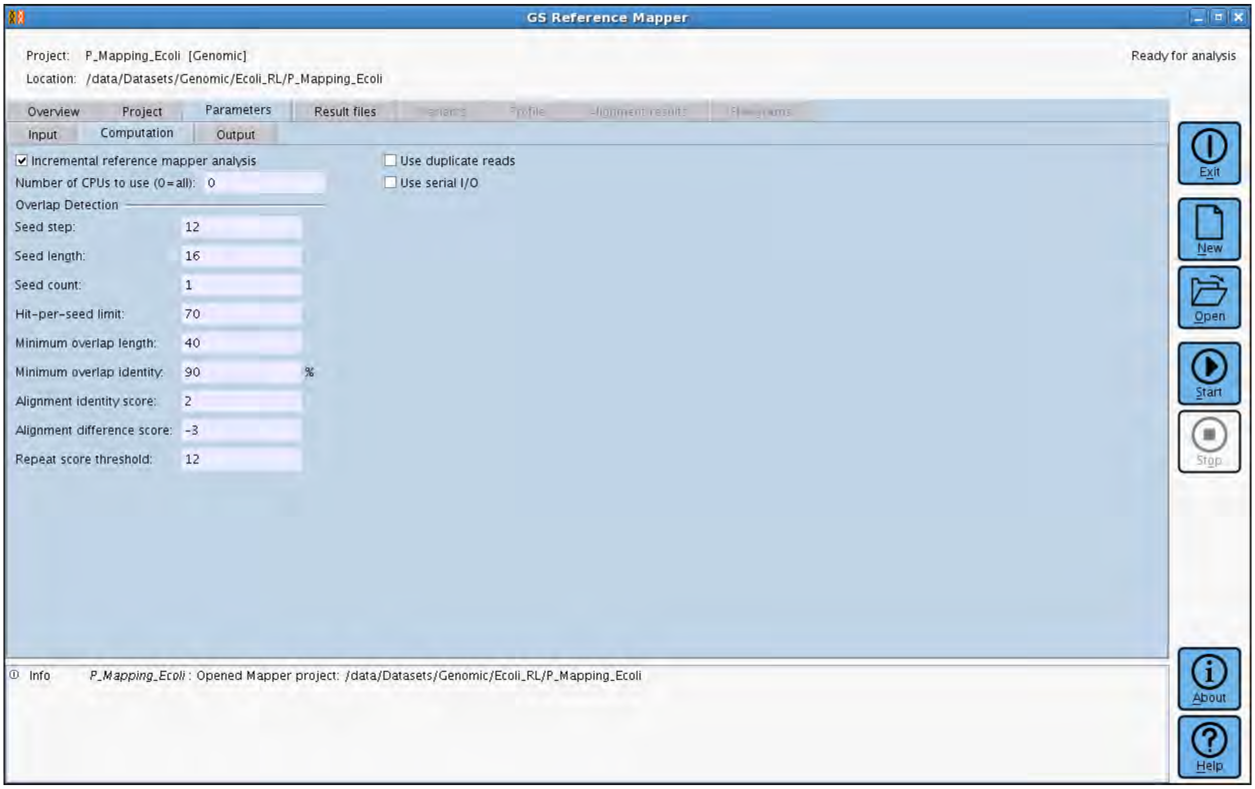
- If there is a single reference file, the software looks for a file with the same name as the reference, but with a .refGene/.refLink/.snp* suffix as the annotations.
- If the reference file(s) is (are all) in the same directory, the software looks in that directory for a "refGene.txt", "refLink.txt" or "snp*.txt" file for the annotation files.
- If the GOLDENPATH environment variable is set for a genomic mapping project, and the reference files are in a "chromosomes" directory inside the GOLDENPATH tree, the software looks for a "database" directory at the same level as the "chromosomes" directory, and then looks for the annotation files inside the database directory. As a consequence the situation may arise whereby a project gets annotated even if no annotation file has been specified in the parameters tab.

- The **Include filter file** (**-fi**) and **exclude filter file** (**-fe**) options can be used to specify a file of sequences to specifically include or exclude in the computation. The file format and functionality are the same as the –e and –i options of sfffile (see Section 3.1).

Long file paths will be displayed in full when you hover the mouse cursor over the text area.

2.7.2 Project Computation Sub-Tab

The Computation sub-tab is shown on Figure 58. It allows you to adjust the settings described below.



Project: P_Mapping_Ecoli [Genomic]
Location: /data/Datasets/Genomic/Ecoli_RL/P_Mapping_Ecoli
Ready for analysis

☑ Incremental reference mapper analysis
Number of CPUs to use (0=all): 0
□ Use duplicate reads
□ Use serial I/O

Overlap Detection
Seed step: 12
Seed length: 16
Seed count: 1
Hit-per-seed limit: 70
Minimum overlap length: 40
Minimum overlap identity: 90 %
Alignment identity score: 2
Alignment difference score: -3
Repeat score threshold: 12

Info P_Mapping_Ecoli : Opened Mapper project: /data/Datasets/Genomic/Ecoli_RL/P_Mapping_Ecoli

Figure 58: gsMapper Parameters tab, Computation sub-tab for Genomic projects.

- **Incremental reference mapper analysis** (**-r**) – Results from computations performed with this option selected (checked) will serve as the basis for the mapping of additional reads. If a project computation is performed with this option deselected, all the project data will be mapped anew each time it is computed, *i.e.* forcing all alignments to be recalculated and overwriting any existing results.
    - Default: selected (Note: This setting is not saved with a project, and is always reset to "selected" when a project is re-opened).
- The **number of CPUs** (**-cpu**) option can be given to limit the load of the software on the computing resources. The default is 0, which specifies that all of the CPUs on the computer should be used. The computation uses the CPU setting to create an approximate load on the computer, so the actual load may vary somewhat from this number, *i.e.*, using "-cpu 3" may cause a load from 2 to 4 on the system.

Currently, only the “Computing alignments…”, “Mapping short reads to reference …”, and “Generating output…” phases have been parallelized (because they are the most computationally intensive phases for larger projects). Other phases of the computation will still use single threading. Also, note that only CPUs which have access to the same physical memory can be used.

The current memory footprint is roughly 4 bytes per reference base plus 2 bytes per input (read) base. For example a 1 Gbp genome with a 20x coverage would need 44 GB of physical memory.

- **Use duplicate reads** (**-ud**) – when checked, the Mapper will include duplicate reads when computing the consensus for a contig and when computing frequency statistics for 454AllDiffs.txt and 454HCDifffs.txt. Duplicate reads are a known potential artifact of the emPCR Amplification process which may occur, for example, when two or more beads are amplified in a single emulsion microreactor. In such cases, duplicate reads should be excluded from the computation, which is the default. This option should be used in certain contexts, *e.g.* when analyzing an amplicon sequencing experiment using the GS Reference Mapper, where every read is a separate data point.
- **Use serial I/O** (**-sio**) – select this option if you are using many .sff files (> 4 million reads) in your project. The option instructs the software to prepare an intermediate read file in which the reads appear in the same order as they do in the alignments. For large projects, this can speed up the execution of the Mapper, especially during the Output Phase of its computation. For more information, see Section 4.4.

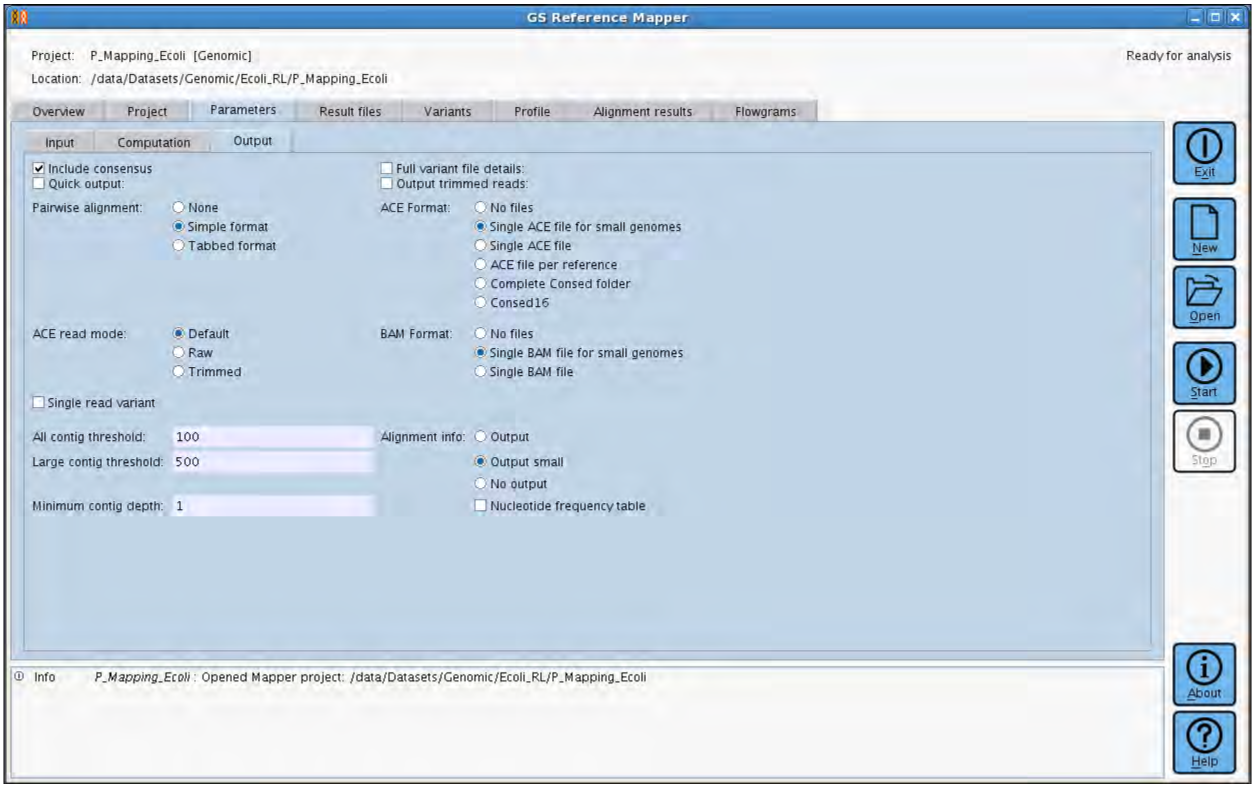
- **Overlap Detection** Parameters:

Parameter	Description	Default Value	Allowed Values
Seed step (-ss)	The number of bases between seed generation locations used in the exact k-mer matching part of the overlap detection.	12	>= 1
Seed length (-sl)	The number of bases used for each seed in the exact k-mer matching part of the overlap detection (*i.e.* the “k” value of the k-mer matching).	16	6 - 32
Seed count (-sc)	The number of seeds required in a window before an extension is made.	1	>= 1
Hit-per-seed limit (-hsl)	A filtering limit on the seeds found during the exact k-mer matching (if more than this number of seed matches are found for a lookup position in the query sequence, those seed matches are not tested as candidate alignments). If more than 70% of a read’s seeds have more than this number of seed hits, the read is not used and is classified as Repeat.	70	1 - 2000
Minimum overlap length (-ml)	The minimum length of overlaps used by the mapper. This value can be entered as either a number of bases or a percent of read length. To enter a percent, the % symbol must be used (*e.g.* 37%).	40	–>= 1 (length) 1 – 100 (percent)
Minimum overlap identity (-mi)	The minimum percent identity of overlaps used by the mapper.	90	1 - 100
Alignment identity score (-ais)	When overlaps are extended during an alignment, this is the per-overlap-column identity score used to calculate the overall score for the alignment.	2	>= 0
Alignment difference score (-ads)	When overlaps are extended during an alignment, this is the per-overlap-column difference score used to calculate the overall score for the alignment.	-3	<= 0
Repeat score threshold (-rst)	The threshold used to declare alignments “unique” or “multiple”. If an input read aligns to more than one position in the reference sequence(s), an alignment score is calculated for each alignment by summing the identity or difference score for each alignment column; if the best alignment score is greater than the score for any other alignment by more than this threshold value, it is marked as “uniquely mapped”. Otherwise, it is “multiply mapped” or “Repeat”.	12	>= 0

When a read is mapped, exact k-mer matches between the read and the reference yield a set of overlaps between the two. These overlaps are then extended into longer alignments. The best alignment is used to determine the mapping location of the read. In some cases, non-overlapping portions of a read may map to different loci in the reference(s). Together these mappings may form a chimeric best alignment for the read. Chimeric mappings favor alignments that are more closely placed along a single reference.

2.7.3 Project Output Sub-Tab

The Output sub-tab is shown on Figure 59. It allows you to adjust the settings described below.



Project: P_Mapping_Ecoli [Genomic]
Location: /data/Datasets/Genomic/Ecoli_RL/P_Mapping_Ecoli
Ready for analysis

☑ Include consensus
□ Quick output:
□ Full variant file details:
□ Output trimmed reads:

Pairwise alignment: ○ None ● Simple format ○ Tabbed format

ACE Format: ○ No files ● Single ACE file for small genomes ○ Single ACE file ○ ACE file per reference ○ Complete Consed folder ○ Consed16

ACE read mode: ● Default ○ Raw ○ Trimmed

BAM Format: ○ No files ● Single BAM file for small genomes ○ Single BAM file

□ Single read variant

All contig threshold: 100
Large contig threshold: 500
Minimum contig depth: 1

Alignment info: ○ Output ● Output small ○ No output
□ Nucleotide frequency table

Info P_Mapping_Ecoli : Opened Mapper project: /data/Datasets/Genomic/Ecoli_RL/P_Mapping_Ecoli

Figure 59: gsMapper Parameters tab, Output sub-tab for Genomic projects.

- **Include consensus (-no not specified)** – If this check box is selected, the application will generate all the output files, including the output files related to the generation of contigs and consensus sequence information. If it is not selected, the application will perform the mapping and will output files (454PairAlign.txt, 454ReadStatus (without locations in reference to what reads map) and 454TrimStatus) associated with the creation of the multiple-alignments, but will not output files (454AllContigs.fna and .qual, 454LargeContigs.fna and .qual, diff, rearrangement and variation files, *etc.*) or metrics (latter parts of 454NewblerMetrics.txt) involved with contig, consensus or variation information. If this option is unchecked, the mapping will run faster, providing an approximate idea of the quality of the data, based on the percentage of reads mapped.
  - Default: selected (Note: This value is not persistent (saved with a project), and is always reset to "selected" when a project is re-opened).
- **Quick output (-qo)** option generates output faster than using the default setting by disabling the "Computing signals…" computation. The default behavior of the mapper requires that all reads be re-read so that signal and quality information can be used to compute consensi (see Section 1.1). If this option is selected, the accuracy of the consensus may be degraded (*i.e.*, more consensus errors may be generated) as the distribution statistics are not generated to assist in the basecalling step.
- **Full variant file details (-fd) –** If this check box is selected, the application will output additional columns to the 454HCDiffs.txt and 454AllDiffs.txt files. See section 454AllDiffs.txt for details.

- **Output trimmed reads (-tr)** – select this option to output the read sequences to a .fna file after trimming has been performed (includes vector / primer / paired end linker / MIDs / low quality trimming). If available, the quality value for the output reads will be placed in a .qual file.

- The **Ace Format** option determines whether or what form of ACE file(s) are output by the application. The default is "Single ACE file for small genomes". See section 2.18.1.5 for a description of the 454Contigs.ace file and of the ace and consed directories. By default, files consistent with version 17.0 or higher of consed are generated.
  - No files (-noace | -nobig) – no ACE file is generated.
  - Single ACE file for small genomes (default, no option specified) – a single ACE file is generated if fewer than 4 million reads are input to the mapping and the reference is less than 40 Mbp.
  - Single ACE file (-ace) – a single ACE file is generated containing the multiple alignments of all contigs.
  - ACE file per contig (-acedir) – a separate ACE file is generated for each contig in the mapping.
  - Complete Consed folder (-consed) – a "consed" folder is generated, containing all the directories and files necessary to display the data in the consed software.
  - Consed16 (-consed16) –This option generates files consistent with version 16.0 or earlier.

- The **BAM Format** option
  - No files (-nobam) – no BAM file is generated.
  - Single BAM file for small genomes (default, no option specified) – a single BAM file is generated if fewer than 4 million reads are input to the mapping and the reference is less than 40 Mbp.
  - Single BAM file (-bam) – a single BAM file is generated containing the multiple alignments of all contigs.

- The **Alignment info** option has four possible settings to control the level of alignment information reported.
  - Output (-infoall) – turns on generation of the 454AlignmentInfo.tsv file.
  - No output (-noinfo) – suppresses the generation of the 454AlignmentInfo.tsv file.
  - Output small (-info) – the file will be generated unless there are more than 4 M reads or the mapping generates contigs with a total length in excess of 40 Mbp.
  - Nucleotide frequency table (-nft) – appends a column to the 454AlignmentInfo.tsv file that reports counts of bases (A,C,T,G,N) and gaps from forward and reverse reads at each position of the multiple alignments.

- **Pairwise alignment** – Determines whether the 454PairAlign.txt file is output; this file contains the overlaps used by the mapper. See section 2.18.1.8 for a description of the 454PairAlign.txt file.
  - None (-nop | -nobig) – the file is not generated (default).
  - Simple format (-pair) – the file contains a human-readable view of the alignments.
  - Tabbed format (-pt | -pairt) – the file contains tab-delimited lines of the overlaps.

- **Ace read mode** – When the ACE file is generated, output reads using either the raw, complete basecalled read or using just the trimmed portion of the reads (after low quality, vector and key trimming).
  - Default (-ad) is set to output raw for mapping.
  - Raw (-ar) is to output the entire sequence of reads.
  - Trimmed (-at) is to output trimmed reads.

- **Single read variant (-srv)** – By default, variations of at least 2 reads relative to the reference are output. If this option is selected, variations supported by only a single read are also included.

- **All contig threshold** (**-a**) – The minimum number of bases for a contig to be output in the 454AllContigs.fna file.
  - Default: 100
- **Large contig threshold** (-l) – The minimum number of bases for a contig to be output in the 454LargeContigs.fna file .
  - Default: 500
- **Minimum contig depth** (**-d**) – regions of references covered by at least this many reads are considered contigs for output purposes.

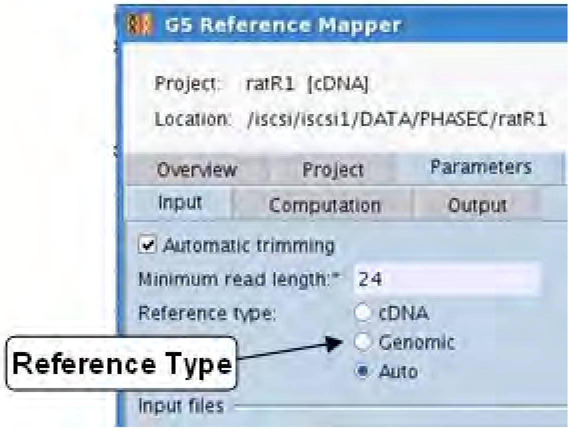
The 454AllContigs.fna file can be generated with the -a (all contigs length) threshold option, which filters out contigs whose lengths are less than the threshold. On rare occasions, the file may contain contigs whose lengths are less than this threshold. This is due to the two-phase nature by which contig consensi are determined. The –a setting is ignored in cDNA assembly computations.

For a full listing and description of the output options for mapping, see Section 4.2.

2.7.4 cDNA Project Input Sub-Tab

The cDNA Mapping project parameters on the Input sub-tab are the same as for Genomic mapping projects (Section 2.7.1) with the exceptions noted below.

- The NimbleGen sequence capture (-n | -nimblegen) check box is not available.
- cDNA Mapping contains a section for specifying the **Reference Type** as **cDNA** (**-cref**), **Genomic** (**-gref**), or **Auto** (default, no option specified).


GS Reference Mapper

Project: ratR1 [cDNA]
Location: /iscsi/iscsi1/DATA/PHASEC/ratR1

☑ Automatic trimming
Minimum read length:* 24
Reference type: ○ cDNA ○ Genomic ● Auto

Figure 60: gsMapper Parameters tab, Input sub-tab for cDNA projects.

The reference type (cDNA or Genomic) can be automatically detected under certain circumstances, thus removing the need to specify the reference type on the command line. The logic employed depends on whether or not the GOLDENPATH environmental variable is in use. See Appendix 5.4 for more details.

The other two sub-tabs of the Parameters tab are identical for Genomic and cDNA projects.

2.8 Computing the Mapping

2.8.1 Computing a Mapping for the First Time

When at least one Read Data file and one Reference file have been added to the project and all parameters are within acceptable limits, the **Start** button becomes enabled and the "Ready for analysis" message appears in the upper right corner of the application window (see Figure 59). The projects parameter settings are also saved when the **Start** button is clicked. Click the **Start** button to save the project parameter settings and carry out the mapping computation of the current project, *i.e.* as defined in the Project and Parameters tabs. As the computation progresses, the **Start** button is disabled, periodic progress messages appear in the progress box, at the bottom of the application's main window, and the current stage of the mapping computation along with a progress bar are displayed in the upper right corner of the window. When the mapping computation is complete, the **Start** button is re-enabled and the message "Ready for analysis" appears again in the upper right corner of the application window.

Mappings with highly-covered regions may not report progress for extended periods of time, even though the assembly is progressing normally. Do not stop the computation of complex projects simply because it has not reported progress for many hours, or even a full day.

2.8.2 Re-Computing a Mapping

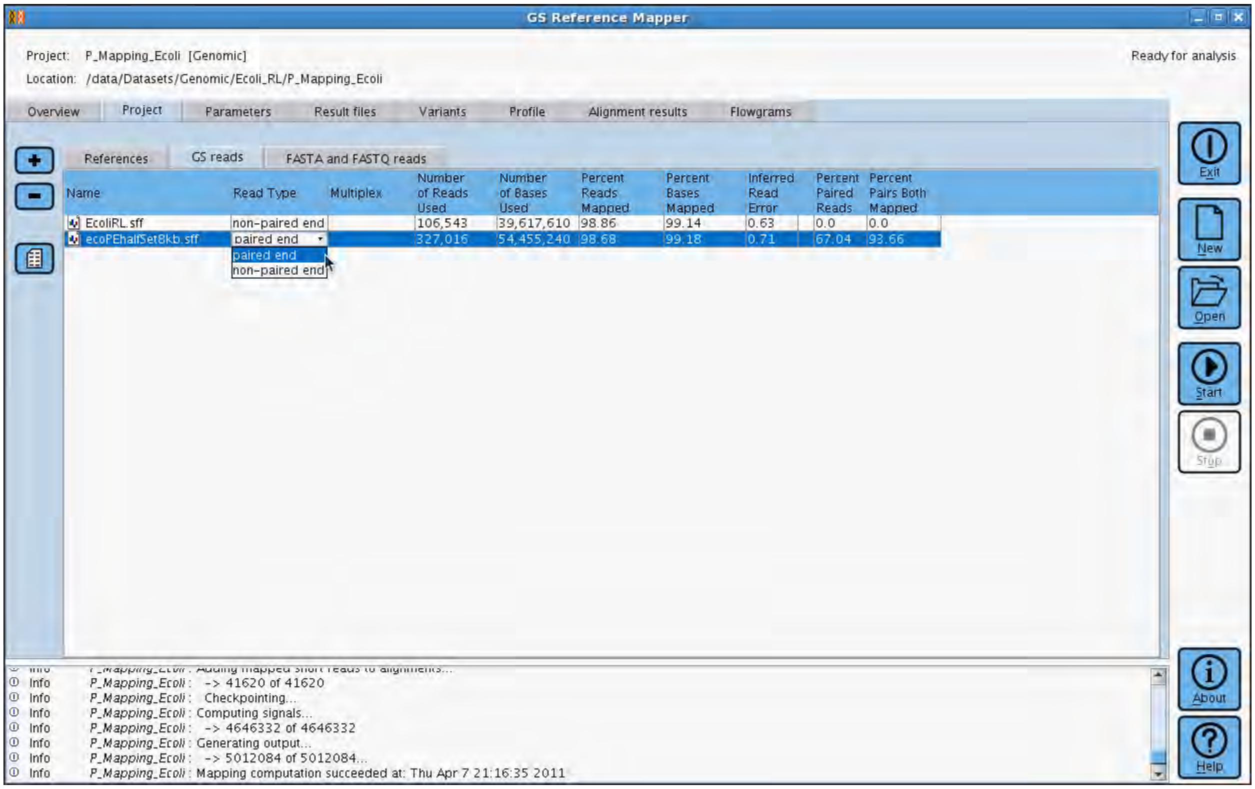
After a mapping has been performed on a project, the mapping computation can be repeated on that project - with the same Read Data or after addition or removal of one or more files, with the same mapping parameter settings or with different ones by pressing the **Start** button. If the "Incremental reference mapper analysis" checkbox on the Parameters tab is checked and no change has been made to the Reference, this computation will use the current parameters settings when mapping any reads new to the project; this re-analysis will take much less time than the initial analysis, since previously mapped reads will remain aligned to their existing reference locations. Changes to some parameter settings (such as those on the Input and Computation sub-tabs of the Project Tab) will have no effect on reads already included the prior mapping computation

2.8.3 Stopping a Mapping Computation

Click on the **Stop** button while a mapping project computation is in progress to halt the computation. The effect may not be instantaneous, however. Any delay will be a function of the complexity of the mapping being performed and of the stage of the computation at the time it was interrupted. An informational message appears in the progress box at the bottom of the application's main window, containing the text "Warning: stopRun called for this project. Stopping…." When the mapping computation has been halted, the **Stop** button is disabled and the **Start** button re-enabled.

2.8.4 Information Updated when Mapping Completes

After a successful mapping, the data for the various columns of the reads in the GS reads and FASTA and FASTQ reads sub-tabs will be updated (Figure 61). Placing the mouse pointer over any cell in the table brings up a tool tip with additional information about that item. A successful mapping also updates data in the Overview tab. Project information is also saved when the computation completes.



Project: P_Mapping_Ecoli [Genomic]
Location: /data/Datasets/Genomic/Ecoli_RL/P_Mapping_Ecoli

Name	Read Type	Multiplex	Number of Reads Used	Number of Bases Used	Percent Reads Mapped	Percent Bases Mapped	Inferred Read Error	Percent Paired Reads	Percent Pairs Both Mapped
EcoliRL.sff	non-paired end		106,543	39,617,610	98.86	99.14	0.63	0.0	0.0
ecoPEhalfSet8kb.sff	paired end		327,016	54,455,240	98.68	99.18	0.71	67.04	93.66

```
Info    P_Mapping_Ecoli : Adding mapped short reads to alignments...
Info    P_Mapping_Ecoli :   -> 41620 of 41620
Info    P_Mapping_Ecoli :   Checkpointing...
Info    P_Mapping_Ecoli : Computing signals...
Info    P_Mapping_Ecoli :   -> 4646332 of 4646332
Info    P_Mapping_Ecoli : Generating output...
Info    P_Mapping_Ecoli :   -> 5012084 of 5012084...
Info    P_Mapping_Ecoli : Mapping computation succeeded at: Thu Apr 7 21:16:35 2011
```

Figure 61: Project Tab of the gsMapper (GS Reads sub-tab), after completion of a mapping computation.

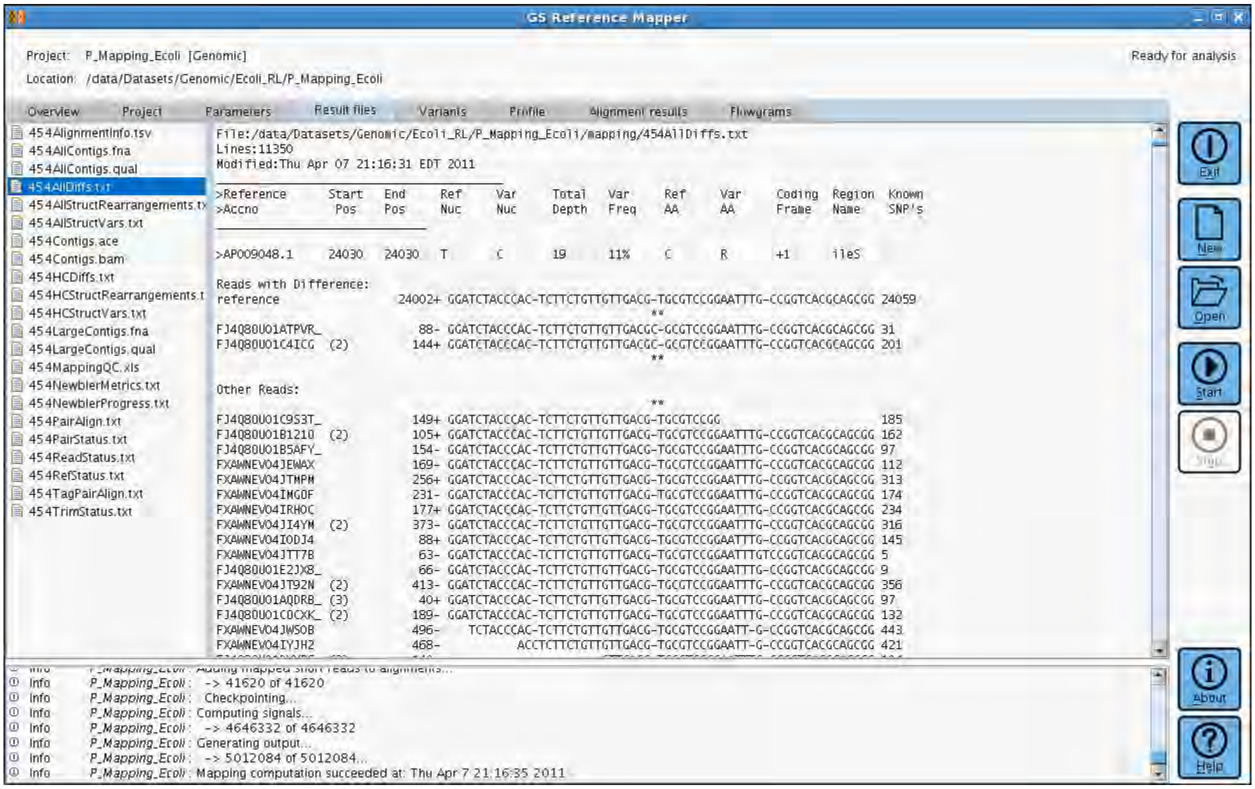
After a mapping project has been computed, the mapping results can be viewed on the Overview, Project, Results Files, Variants, Profile, Alignment Results, and Flowgram tabs of the Reference Mapper application's main window.

2.9 Viewing Mapping Output with the Result Files Tab

The Result Files tab allows you to view the various files generated by the GS Reference Mapper software (described in detail in 2.18.1). Upper case and lower case letters are used to indicate basecall quality in DNA sequence files. Lower case letters in individual trimmed reads represent bases with a basecall accuracy of <99% (Q<20). In contigs lower case letters are bases with a basecall accuracy of <99.99% (Q<40). Lower case letters are also used to indicate the bases that comprise the sequencing key when raw (untrimmed) SFF reads appear in FASTA or ACE formats.

Using the list in the left-hand panel of the tab, click on the name of the output file you want to examine; its content will be displayed on the right-hand panel of the tab (Figure 62).

For very long files, such as the sequences of all contigs mapped to a large genome, this view displays the file truncated to 50,000 lines. When this occurs, a note to that effect appears at both the beginning and at the end of the display. The file itself remains intact, however.



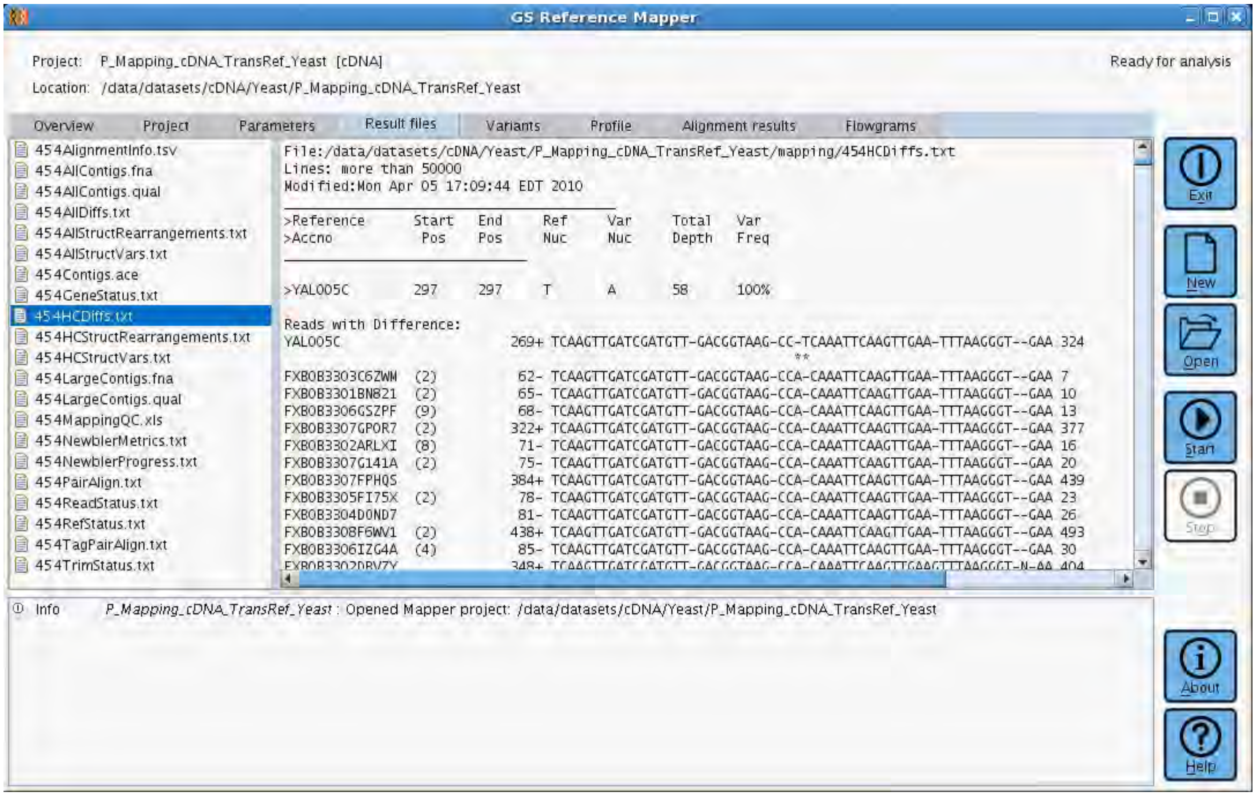
Project: P_Mapping_Ecoli [Genomic]
Location: /data/Datasets/Genomic/Ecoli_RL/P_Mapping_Ecoli

```
File:/data/Datasets/Genomic/Ecoli_RL/P_Mapping_Ecoli/mapping/454AllDiffs.txt
Lines:11350
Modified:Thu Apr 07 21:16:31 EDT 2011
_____________________________________________
>Reference       Start   End     Ref     Var     Total   Var     Ref     Var     Coding  Region  Known
>Accno            Pos    Pos     Nuc     Nuc     Depth   Freq    AA      AA      Frame   Name    SNP's
__________________________________

>AP009048.1     24030   24030   T       C       19      11%     C       R       +1      ileS

Reads with Difference:
reference                  24002+ GGATCTACCCAC-TCTTCTGTTGTTGACG-TGCGTCCGGAATTTG-CCGGTCACGCAGCGG 24059
                                                                 **
FJ4Q80U01ATPVR_               88- GGATCTACCCAC-TCTTCTGTTGTTGACGC-GCGTCCGGAATTTG-CCGGTCACGCAGCGG 31
FJ4Q80U01C4ICG  (2)          144+ GGATCTACCCAC-TCTTCTGTTGTTGACGC-GCGTCCGGAATTTG-CCGGTCACGCAGCGG 201
                                                                 **

Other Reads:
                                                                 **
FJ4Q80U01C9S3T_              149+ GGATCTACCCAC-TCTTCTGTTGTTGACG-TGCGTCCGG                       185
FJ4Q80U01B1210  (2)          105+ GGATCTACCCAC-TCTTCTGTTGTTGACG-TGCGTCCGGAATTTG-CCGGTCACGCAGCGG 162
FJ4Q80U01B5AFY_              154- GGATCTACCCAC-TCTTCTGTTGTTGACG-TGCGTCCGGAATTTG-CCGGTCACGCAGCGG 97
FXAWNEV04JEWAX               169- GGATCTACCCAC-TCTTCTGTTGTTGACG-TGCGTCCGGAATTTG-CCGGTCACGCAGCGG 112
FXAWNEV04JTMPM               256+ GGATCTACCCAC-TCTTCTGTTGTTGACG-TGCGTCCGGAATTTG-CCGGTCACGCAGCGG 313
FXAWNEV04IMGDF               231- GGATCTACCCAC-TCTTCTGTTGTTGACG-TGCGTCCGGAATTTG-CCGGTCACGCAGCGG 174
FXAWNEV04IRHOC               177+ GGATCTACCCAC-TCTTCTGTTGTTGACG-TGCGTCCGGAATTTG-CCGGTCACGCAGCGG 234
FXAWNEV04JI4YM  (2)          373- GGATCTACCCAC-TCTTCTGTTGTTGACG-TGCGTCCGGAATTTG-CCGGTCACGCAGCGG 316
FXAWNEV04IODJ4                88+ GGATCTACCCAC-TCTTCTGTTGTTGACG-TGCGTCCGGAATTTG-CCGGTCACGCAGCGG 145
FXAWNEV04JTT7B                63- GGATCTACCCAC-TCTTCTGTTGTTGACG-TGCGTCCGGAATTTGTCCGGTCACGCAGCGG 5
FJ4Q80U01E2JX8_               66- GGATCTACCCAC-TCTTCTGTTGTTGACG-TGCGTCCGGAATTTG-CCGGTCACGCAGCGG 9
FXAWNEV04JT92N  (2)          413- GGATCTACCCAC-TCTTCTGTTGTTGACG-TGCGTCCGGAATTTG-CCGGTCACGCAGCGG 356
FJ4Q80U01AQDRB_ (3)           40+ GGATCTACCCAC-TCTTCTGTTGTTGACG-TGCGTCCGGAATTTG-CCGGTCACGCAGCGG 97
FJ4Q80U01CDCXK_ (2)          189- GGATCTACCCAC-TCTTCTGTTGTTGACG-TGCGTCCGGAATTTG-CCGGTCACGCAGCGG 132
FXAWNEV04JWSOB               496-    TCTACCCAC-TCTTCTGTTGTTGACG-TGCGTCCGGAATT-G-CCGGTCACGCAGCGG 443
FXAWNEV04IYJH2               468-           ACCTCTTCTGTTGTTGACG-TGCGTCCGGAATT-G-CCGGTCACGCAGCGG 421
```

```
Info    P_Mapping_Ecoli : Adding mapped short reads to alignments...
Info    P_Mapping_Ecoli :   -> 41620 of 41620
Info    P_Mapping_Ecoli :   Checkpointing...
Info    P_Mapping_Ecoli : Computing signals...
Info    P_Mapping_Ecoli :   -> 4646332 of 4646332
Info    P_Mapping_Ecoli : Generating output...
Info    P_Mapping_Ecoli :   -> 5012084 of 5012084...
Info    P_Mapping_Ecoli : Mapping computation succeeded at: Thu Apr 7 21:16:35 2011
```

Figure 62: gsMapper Result Files tab for Genomic projects.

Figure 62 above shows the Result Files tab for a Genomic Mapping project. The result files for a cDNA project are shown below (Figure 63). The result file contents are described in detail in Section 2.18.1.



Project: P_Mapping_cDNA_TransRef_Yeast [cDNA]
Location: /data/datasets/cDNA/Yeast/P_Mapping_cDNA_TransRef_Yeast

Ready for analysis

454AlignmentInfo.tsv
454AllContigs.fna
454AllContigs.qual
454AllDiffs.txt
454AllStructRearrangements.txt
454AllStructVars.txt
454Contigs.ace
454GeneStatus.txt
454HCDiffs.txt
454HCStructRearrangements.txt
454HCStructVars.txt
454LargeContigs.fna
454LargeContigs.qual
454MappingQC.xls
454NewblerMetrics.txt
454NewblerProgress.txt
454PairAlign.txt
454ReadStatus.txt
454RefStatus.txt
454TagPairAlign.txt
454TrimStatus.txt

```
File:/data/datasets/cDNA/Yeast/P_Mapping_cDNA_TransRef_Yeast/mapping/454HCDiffs.txt
Lines: more than 50000
Modified:Mon Apr 05 17:09:44 EDT 2010
______________________________________________
>Reference       Start   End     Ref     Var     Total   Var
>Accno            Pos    Pos     Nuc     Nuc     Depth   Freq
_____________________________________


>YAL005C          297    297     T       A       58      100%

Reads with Difference:
YAL005C                         269+ TCAAGTTGATCGATGTT-GACGGTAAG-CC-TCAAATTCAAGTTGAA-TTTAAGGGT--GAA 324
                                                                **
FXBOB3303C6ZWM  (2)              62- TCAAGTTGATCGATGTT-GACGGTAAG-CCA-CAAATTCAAGTTGAA-TTTAAGGGT--GAA 7
FXBOB3301BN821  (2)              65- TCAAGTTGATCGATGTT-GACGGTAAG-CCA-CAAATTCAAGTTGAA-TTTAAGGGT--GAA 10
FXBOB3306GSZPF  (9)              68- TCAAGTTGATCGATGTT-GACGGTAAG-CCA-CAAATTCAAGTTGAA-TTTAAGGGT--GAA 13
FXBOB3307GPOR7  (2)             322+ TCAAGTTGATCGATGTT-GACGGTAAG-CCA-CAAATTCAAGTTGAA-TTTAAGGGT--GAA 377
FXBOB3302ARLXI  (8)              71- TCAAGTTGATCGATGTT-GACGGTAAG-CCA-CAAATTCAAGTTGAA-TTTAAGGGT--GAA 16
FXBOB3307G141A  (2)              75- TCAAGTTGATCGATGTT-GACGGTAAG-CCA-CAAATTCAAGTTGAA-TTTAAGGGT--GAA 20
FXBOB3307FPHQS                  384+ TCAAGTTGATCGATGTT-GACGGTAAG-CCA-CAAATTCAAGTTGAA-TTTAAGGGT--GAA 439
FXBOB3305FI75X  (2)              78- TCAAGTTGATCGATGTT-GACGGTAAG-CCA-CAAATTCAAGTTGAA-TTTAAGGGT--GAA 23
FXBOB3304D0ND7                   81- TCAAGTTGATCGATGTT-GACGGTAAG-CCA-CAAATTCAAGTTGAA-TTTAAGGGT--GAA 26
FXBOB3308F6WV1  (2)             438+ TCAAGTTGATCGATGTT-GACGGTAAG-CCA-CAAATTCAAGTTGAA-TTTAAGGGT--GAA 493
FXBOB3306IZG4A  (4)              85- TCAAGTTGATCGATGTT-GACGGTAAG-CCA-CAAATTCAAGTTGAA-TTTAAGGGT--GAA 30
FXBOB3302DBVZY                  348+ TCAAGTTGATCGATGTT-GACGGTAAG-CCA-CAAATTCAAGTTGAAGTTTAAGGGT-N-AA 404
```

ⓘ Info   *P_Mapping_cDNA_TransRef_Yeast* : Opened Mapper project: /data/datasets/cDNA/Yeast/P_Mapping_cDNA_TransRef_Yeast

Figure 63: gsMapper Results Files tab for cDNA projects.

2.10 Using the Variants Tab

There are three sub-tabs on the Variants tab that contain information about differences (relative to the reference) that are indicated by alignments of reads mapped to the reference: one reports on nucleotide differences and the others on structural variations. The sub-tabs contain tables of variations showing statistics, loci, and (optionally) annotations. Using the mouse, you can right-click any of the entries on the HC Diffs and HC Structural Variants sub-tabs and choose to view the alignment for the selected variation.

The sub-tabs are described in more detail in the sections below; they share the following common features:

- **Sorting:**
  - All columns are sortable by clicking on the column header. A small triangle appears to indicate the direction of the sort (see Figure 64 and Figure 66, below). Clicking on a column header of a column that is already sorted will reverse the order of the sort.
  - If multiple rows have the same value for the sorted column, these rows stay in the order in which they were before the sort, relative to one another. This allows for "nested sorting". For example, if you:
    - sort first based on "Start Position in Ref",
    - then sort based on "Reference Accession Number",
    - the result is a table where the high confidence differences are grouped by reference sequence, and the ones that belong to the same reference are displayed in order of their position on the reference sequence.

- **Summary tooltip** (mouse-over information):
  - For the HC Structural Rearrangement sub-tab, hovering the mouse cursor over any row provides a tooltip that indicates the type of rearrangement and position or length, depending on the type of variation.
  - Hovering the mouse cursor over any row in either of the other two sub-tabs provides a tooltip that summarizes the basic statistics of the high confidence difference described in that row (see Figure 64). In particular, the bases involved in the variation coupled with the total variation percentage, the depth of sequencing, and the separate statistics for both forward and reverse reads are presented.

- **Links to read alignment data:**
  - Right-clicking in the row for a particular HCDiff or HC Structural Variant opens a contextual menu with a single item: "Show Alignment". Selecting this loads the Alignment results tab with the alignment that supports the variation in question. The first base of the variation appears as the leftmost base in the view.
  - You can also select multiple variants, then right-click to load these variations into a list. The Alignment results tab will display the first variation in the list and the list itself will comprise the dropdown menu contents of the Positions of interest (Figure 66).
  - You can then scroll the alignment to examine the context of the variation.

- **Export of Variations Tables:**
  - You can export variations data, in its current sort order, in png, txt or csv format. The txt and csv export includes the complete table. The png format only includes the currently visible portion of the table.
  - The export dialog box is launched by clicking the 📄 button, for txt or csv format, or the 📷 button for png format. (See Figure 67)

Note on duplicate reads handling when computing Variants:

Duplicate reads of a single DNA input are a known potential artifact of the emPCR Amplification process which may occur, for example, when two or more beads are amplified in a single emulsion microreactor. In an attempt to compute more accurate variation percentages, the GS Reference Mapper, by default, groups individual reads into groups of duplicates (reads that start at the same base of the reference sequence). Groups of duplicate reads count as only one read in the computation of variants (HCDiffs and HC Structural Variants). However, the corresponding alignments on the Alignment results tab display all the reads whether or not they were considered duplicates. When grouping occurs, the variant summary tooltip provides the statistics for both grouped and for individual reads (see Figure 64).

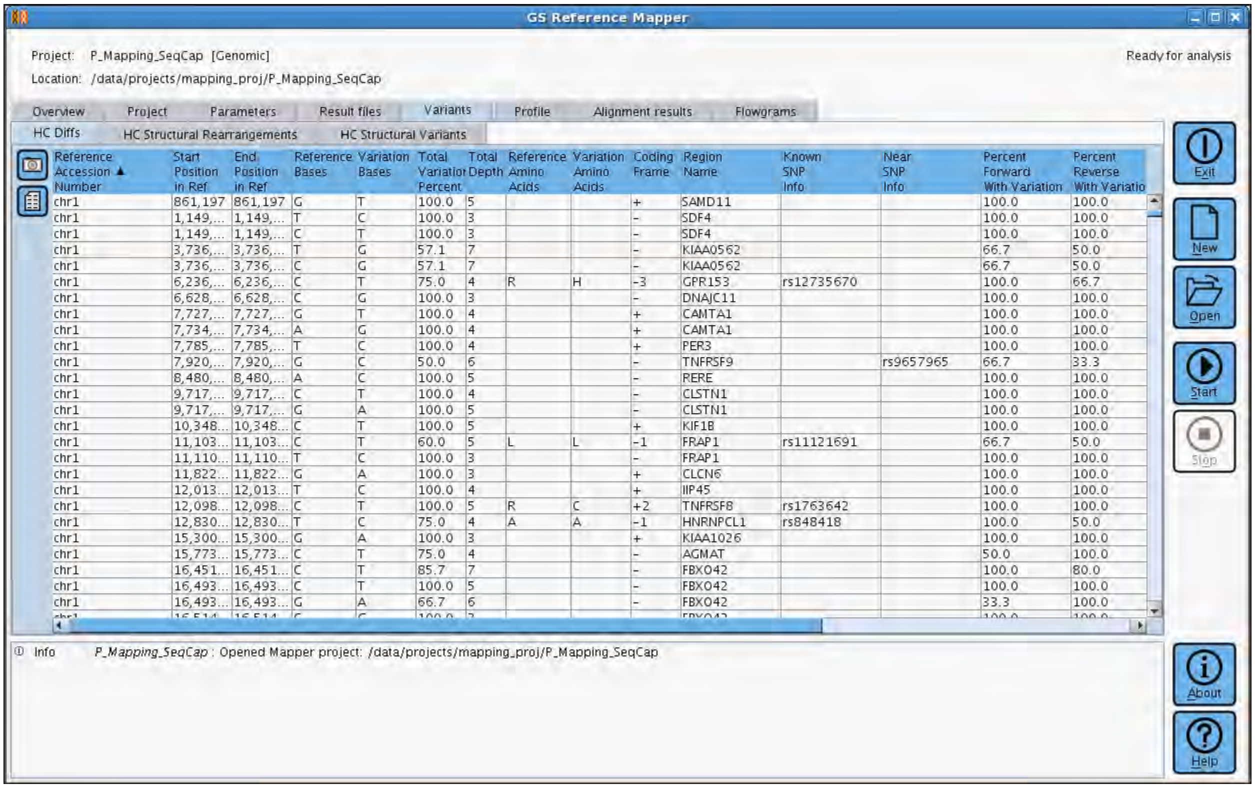
On the command line, the “-ud” option (see 4.2) instructs the GS Reference Mapper to treat all reads individually and to not perform this grouping operation. This option is available in the GUI, on the Parameters/Computation sub-tab. This should be used, for example, when using the GS Reference Mapper to analyze data from an amplicon sequencing experiment, where every read is a distinct data point.

2.10.1 The HC Diffs Sub-Tab

Information about local differences between reads and the reference appear in a table on the **HC Diffs** sub-tab. Explanations for each of the columns in the table are found in the description of the 454AllDiffs.txt file, in Section 2.18.1.13. Right-clicking on a row opens a contextual menu from which one can access the alignment that underlies the variation. Pausing the mouse over a row displays a tooltip providing information about that variation.

The tab can be divided conceptually into four main functional sections (Figure 64):

- The leftmost columns (from Reference Accession Number to Total Depth) contain general information on the high confidence differences (HCDiffs) that were found in the computed data, between the reads and the reference sequences.
- The center columns (from Reference Amino Acids to Known SNP Info) show gene annotation or known SNP file information on the regions of the HCDiffs. For such information to be available, Genome Annotation and Known SNP databases must have been specified (either on the Parameters Tab or by using a GoldenPath reference sequence).
- The columns from Percent Forward to Total Num Reverse Reads contain a detailed breakdown of the computation data shown on the left.
- The last column, Target Region Status, is filled if the project is run with a targeted regions file, and indicates whether the HCDiff is in a targeted region or in an extended region. HCDiffs outside extended target regions are not reported in this table.



Project: P_Mapping_SeqCap [Genomic]
Location: /data/projects/mapping_proj/P_Mapping_SeqCap

Ready for analysis

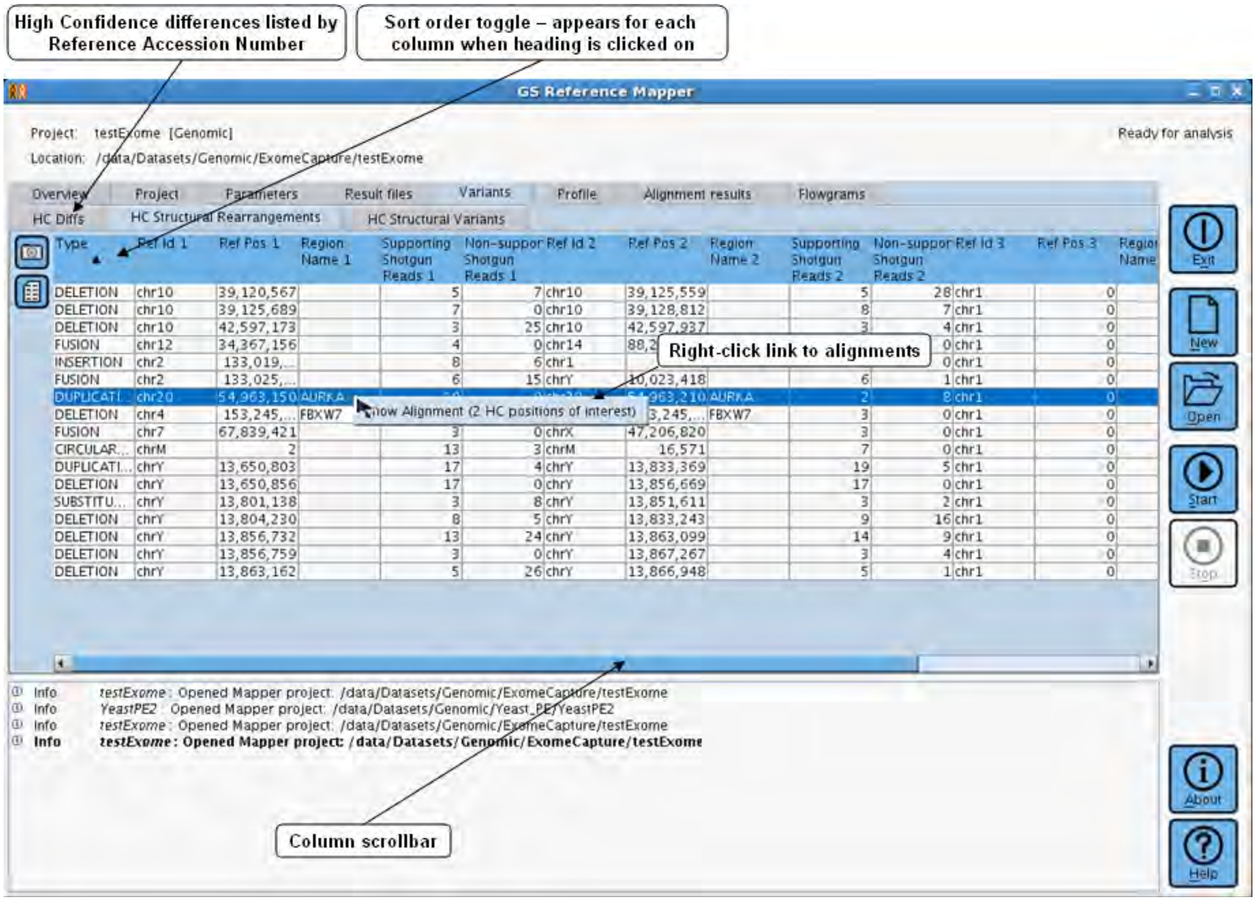
Reference Accession Number ▲	Start Position in Ref	End Position in Ref	Reference Bases	Variation Bases	Total Variatior Percent	Total Depth	Reference Amino Acids	Variation Amino Acids	Coding Frame	Region Name	Known SNP Info	Near SNP Info	Percent Forward With Variation	Percent Reverse With Variatio
chr1	861,197	861,197	G	T	100.0	5			+	SAMD11			100.0	100.0
chr1	1,149,...	1,149,...	T	C	100.0	3			-	SDF4			100.0	100.0
chr1	1,149,...	1,149,...	C	T	100.0	3			-	SDF4			100.0	100.0
chr1	3,736,...	3,736,...	T	G	57.1	7			-	KIAA0562			66.7	50.0
chr1	3,736,...	3,736,...	C	G	57.1	7			-	KIAA0562			66.7	50.0
chr1	6,236,...	6,236,...	C	T	75.0	4	R	H	-3	GPR153	rs12735670		100.0	66.7
chr1	6,628,...	6,628,...	C	G	100.0	3			-	DNAJC11			100.0	100.0
chr1	7,727,...	7,727,...	G	T	100.0	4			+	CAMTA1			100.0	100.0
chr1	7,734,...	7,734,...	A	G	100.0	4			+	CAMTA1			100.0	100.0
chr1	7,785,...	7,785,...	T	C	100.0	4			+	PER3			100.0	100.0
chr1	7,920,...	7,920,...	G	C	50.0	6			-	TNFRSF9		rs9657965	66.7	33.3
chr1	8,480,...	8,480,...	A	C	100.0	5			-	RERE			100.0	100.0
chr1	9,717,...	9,717,...	C	T	100.0	4			-	CLSTN1			100.0	100.0
chr1	9,717,...	9,717,...	G	A	100.0	5			-	CLSTN1			100.0	100.0
chr1	10,348...	10,348...	C	T	100.0	5			+	KIF1B			100.0	100.0
chr1	11,103...	11,103...	C	T	60.0	5	L	L	-1	FRAP1	rs11121691		66.7	50.0
chr1	11,110...	11,110...	T	C	100.0	3			-	FRAP1			100.0	100.0
chr1	11,822...	11,822...	G	A	100.0	3			+	CLCN6			100.0	100.0
chr1	12,013...	12,013...	T	C	100.0	4			+	IIP45			100.0	100.0
chr1	12,098...	12,098...	C	T	100.0	5	R	C	+2	TNFRSF8	rs1763642		100.0	100.0
chr1	12,830...	12,830...	T	C	75.0	4	A	A	-1	HNRNPCL1	rs848418		100.0	50.0
chr1	15,300...	15,300...	G	A	100.0	3			+	KIAA1026			100.0	100.0
chr1	15,773...	15,773...	C	T	75.0	4			-	AGMAT			50.0	100.0
chr1	16,451...	16,451...	C	T	85.7	7			-	FBXO42			100.0	80.0
chr1	16,493...	16,493...	C	T	100.0	5			-	FBXO42			100.0	100.0
chr1	16,493...	16,493...	G	A	66.7	6			-	FBXO42			33.3	100.0
chr1	16,514...	16,514...	C	C	100.0	3				FBXO42			100.0	100.0

ⓘ Info P_Mapping_SeqCap : Opened Mapper project: /data/projects/mapping_proj/P_Mapping_SeqCap

Figure 64: gsMapper Variants tab, HC Diffs sub-tab.

2.10.2 The HC Structural Rearrangements Sub-Tab

The HC Structural Rearrangements sub-tab (Figure 65) shows the biological classification that can be inferred from one or more entries listed on the HC Structural Variants sub-tab. The GS Reference Mapper software attempts to interpret and classify groups of structural variants in order to assign a higher-level rearrangement type to the group. Summary information about the supporting structural variants is provided on this sub-tab. Right-clicking on a row opens a contextual menu from which one can access the alignment that underlies the variation. Pausing the mouse over a row displays a tooltip providing the classification and either the reference position or the length of the rearrangement. Explanations for each of the columns in the table are found in the description of the 454AllStructRearrangements.txt file, in Section 2.18.1.16.


High Confidence differences listed by Reference Accession Number
Sort order toggle – appears for each column when heading is clicked on
GS Reference Mapper
Ready for analysis
Overview
Project
Parameters
Result files
Variants
Profile
Alignment results
Flowgrams
HC Diffs
HC Structural Rearrangements
HC Structural Variants
Right-click link to alignments
Show Alignment (2 HC positions of interest)
Column scrollbar
Exit
New
Open
Start
Stop
About
Help

Project: testExome [Genomic]
Location: /data/Datasets/Genomic/ExomeCapture/testExome

Type	Ref Id 1	Ref Pos 1	Region Name 1	Supporting Shotgun Reads 1	Non-suppor Shotgun Reads 1	Ref Id 2	Ref Pos 2	Region Name 2	Supporting Shotgun Reads 2	Non-suppor Shotgun Reads 2	Ref Id 3	Ref Pos 3	Regio Name
DELETION	chr10	39,120,567		5	7	chr10	39,125,559		5	28	chr1	0	
DELETION	chr10	39,125,689		7	0	chr10	39,128,812		8	7	chr1	0	
DELETION	chr10	42,597,173		3	25	chr10	42,597,937		3	4	chr1	0	
FUSION	chr12	34,367,156		4	0	chr14	88,2	[illegible]	[illegible]	0	chr1	0	
INSERTION	chr2	133,019,...		8	6	chr1	[illegible]	[illegible]	[illegible]	0	chr1	0	
FUSION	chr2	133,025,...		6	15	chrY	10,023,418		6	1	chr1	0	
DUPLICATI...	chr20	54,963,150	AURKA	[illegible]	[illegible]	[illegible]	54,963,210	AURKA	2	8	chr1	0	
DELETION	chr4	153,245,...	FBXW7	[illegible]	[illegible]	[illegible]	3,245,...	FBXW7	3	0	chr1	0	
FUSION	chr7	67,839,421		3	0	chrX	47,206,820		3	0	chr1	0	
CIRCULAR...	chrM	2		13	3	chrM	16,571		7	0	chr1	0	
DUPLICATI...	chrY	13,650,803		17	4	chrY	13,833,369		19	5	chr1	0	
DELETION	chrY	13,650,856		17	0	chrY	13,856,669		17	0	chr1	0	
SUBSTITU...	chrY	13,801,138		3	8	chrY	13,851,611		3	2	chr1	0	
DELETION	chrY	13,804,230		8	5	chrY	13,833,243		9	16	chr1	0	
DELETION	chrY	13,856,732		13	24	chrY	13,863,099		14	9	chr1	0	
DELETION	chrY	13,856,759		3	0	chrY	13,867,267		3	4	chr1	0	
DELETION	chrY	13,863,162		5	26	chrY	13,866,948		5	1	chr1	0	

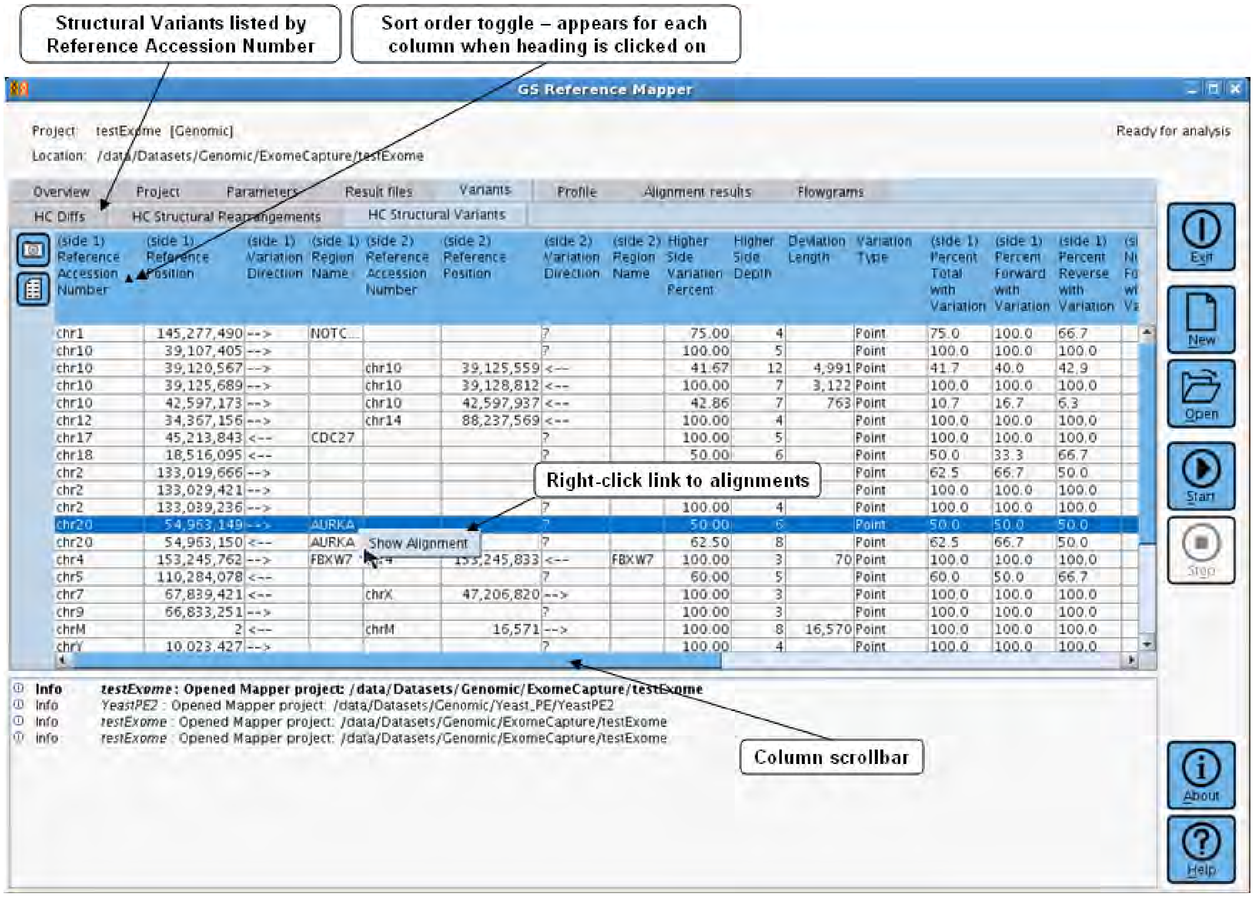
```
Info   testExome: Opened Mapper project: /data/Datasets/Genomic/ExomeCapture/testExome
Info   YeastPE2: Opened Mapper project: /data/Datasets/Genomic/Yeast_PE/YeastPE2
Info   testExome: Opened Mapper project: /data/Datasets/Genomic/ExomeCapture/testExome
Info   testExome: Opened Mapper project: /data/Datasets/Genomic/ExomeCapture/testExome
```

Figure 65: gsMapper Variants tab, HC Structural Rearrangements sub-tab.

2.10.3 The HC Structural Variants Sub-Tab

The HC Structural Variants sub-tab (Figure 66) shows locations of larger-scale changes relative to the reference that are indicated by a group of reads. Entries in the table are classified as either Rearrangement points or Rearrangement regions. See Section 2.18.1.15 for more details on these topics. Right-clicking on a row opens a contextual menu from which one can access the alignment that underlies the variation. Pausing the mouse over a row displays a tooltip providing information about that variation.

A

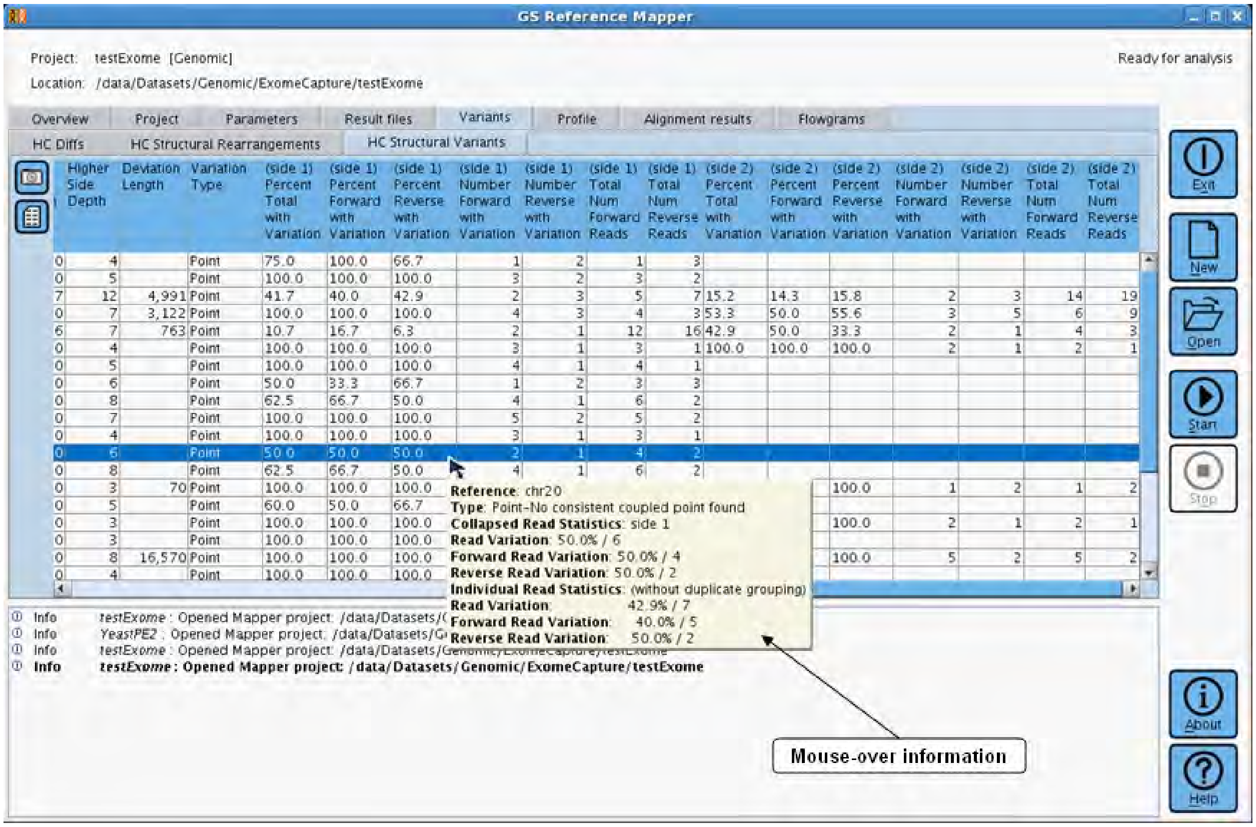


Project: testExome [Genomic]
Location: /data/Datasets/Genomic/ExomeCapture/testExome

Ready for analysis

(side 1) Reference Accession Number	(side 1) Reference Position	(side 1) Variation Direction	(side 1) Region Name	(side 2) Reference Accession Number	(side 2) Reference Position	(side 2) Variation Direction	(side 2) Region Name	Higher Side Variation Percent	Higher Side Depth	Deviation Length	Variation Type	(side 1) Percent Total with Variation	(side 1) Percent Forward with Variation	(side 1) Percent Reverse with Variation
chr1	145,277,490	-->	NOTC...			?		75.00	4		Point	75.0	100.0	66.7
chr10	39,107,405	-->				?		100.00	5		Point	100.0	100.0	100.0
chr10	39,120,567	-->		chr10	39,125,559	<--		41.67	12	4,991	Point	41.7	40.0	42.9
chr10	39,125,689	-->		chr10	39,128,812	<--		100.00	7	3,122	Point	100.0	100.0	100.0
chr10	42,597,173	-->		chr10	42,597,937	<--		42.86	7	763	Point	10.7	16.7	6.3
chr12	34,367,156	-->		chr14	88,237,569	<--		100.00	4		Point	100.0	100.0	100.0
chr17	45,213,843	<--	CDC27			?		100.00	5		Point	100.0	100.0	100.0
chr18	18,516,095	<--				?		50.00	6		Point	50.0	33.3	66.7
chr2	133,019,666	-->				[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	Point	62.5	66.7	50.0
chr2	133,029,421	-->				[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	Point	100.0	100.0	100.0
chr2	133,039,236	-->				?		100.00	4		Point	100.0	100.0	100.0
chr20	54,963,149	-->	AURKA			?		50.00	6		Point	50.0	50.0	50.0
chr20	54,963,150	<--	AURKA			?		62.50	8		Point	62.5	66.7	50.0
chr4	153,245,762	-->	FBXW7	chr4	153,245,833	<--	FBXW7	100.00	3	70	Point	100.0	100.0	100.0
chr5	110,284,078	<--				?		60.00	5		Point	60.0	50.0	66.7
chr7	67,839,421	<--		chrX	47,206,820	-->		100.00	3		Point	100.0	100.0	100.0
chr9	66,833,251	-->				?		100.00	3		Point	100.0	100.0	100.0
chrM	2	<--		chrM	16,571	-->		100.00	8	16,570	Point	100.0	100.0	100.0
chrY	10,023,427	-->				?		100.00	4		Point	100.0	100.0	100.0

```
Info   testExome: Opened Mapper project: /data/Datasets/Genomic/ExomeCapture/testExome
Info   YeastPE2 : Opened Mapper project: /data/Datasets/Genomic/Yeast_PE/YeastPE2
Info   testExome : Opened Mapper project: /data/Datasets/Genomic/ExomeCapture/testExome
Info   testExome : Opened Mapper project: /data/Datasets/Genomic/ExomeCapture/testExome
```



Project: testExome [Genomic]

Location: /data/Datasets/Genomic/ExomeCapture/testExome

	Higher Side Depth	Deviation Length	Variation Type	(side 1) Percent Total with Variation	(side 1) Percent Forward with Variation	(side 1) Percent Reverse with Variation	(side 1) Number Forward with Variation	(side 1) Number Reverse with Variation	(side 1) Total Num Forward Reads	(side 1) Total Num Reverse Reads	(side 2) Percent Total with Variation	(side 2) Percent Forward with Variation	(side 2) Percent Reverse with Variation	(side 2) Number Forward with Variation	(side 2) Number Reverse with Variation	(side 2) Total Num Forward Reads	(side 2) Total Num Reverse Reads
0	4		Point	75.0	100.0	66.7	1	2	1	3							
0	5		Point	100.0	100.0	100.0	3	2	3	2							
7	12	4,991	Point	41.7	40.0	42.9	2	3	5	7	15.2	14.3	15.8	2	3	14	19
0	7	3,122	Point	100.0	100.0	100.0	4	3	4	3	53.3	50.0	55.6	3	5	6	9
6	7	763	Point	10.7	16.7	6.3	2	1	12	16	42.9	50.0	33.3	2	1	4	3
0	4		Point	100.0	100.0	100.0	3	1	3	1	100.0	100.0	100.0	2	1	2	1
0	5		Point	100.0	100.0	100.0	4	1	4	1							
0	6		Point	50.0	33.3	66.7	1	2	3	3							
0	8		Point	62.5	66.7	50.0	4	1	6	2							
0	7		Point	100.0	100.0	100.0	5	2	5	2							
0	4		Point	100.0	100.0	100.0	3	1	3	1							
0	6		Point	50.0	50.0	50.0	2	1	4	2							
0	8		Point	62.5	66.7	50.0	4	1	6	2							
0	3	70	Point	100.0	100.0	100.0							100.0	1	2	1	2
0	5		Point	60.0	50.0	66.7											
0	3		Point	100.0	100.0	100.0							100.0	2	1	2	1
0	3		Point	100.0	100.0	100.0											
0	8	16,570	Point	100.0	100.0	100.0							100.0	5	2	5	2
0	4		Point	100.0	100.0	100.0											

Reference: chr20
Type: Point-No consistent coupled point found
Collapsed Read Statistics: side 1
Read Variation: 50.0% / 6
Forward Read Variation: 50.0% / 4
Reverse Read Variation: 50.0% / 2
Individual Read Statistics: (without duplicate grouping)
Read Variation: 42.9% / 7
Forward Read Variation: 40.0% / 5
Reverse Read Variation: 50.0% / 2

```
Info    testExome : Opened Mapper project: /data/Datasets/(
Info    YeastPE2 : Opened Mapper project: /data/Datasets/G
Info    testExome : Opened Mapper project: /data/Datasets/Genomic/ExomeCapture/testExome
Info    testExome : Opened Mapper project: /data/Datasets/Genomic/ExomeCapture/testExome
```

Figure 66: gsMapper Variants tab, HC Structural Variants sub-tab. A: Left columns; B: Right columns.

Note on handling paired end reads when computing Structural Variants:

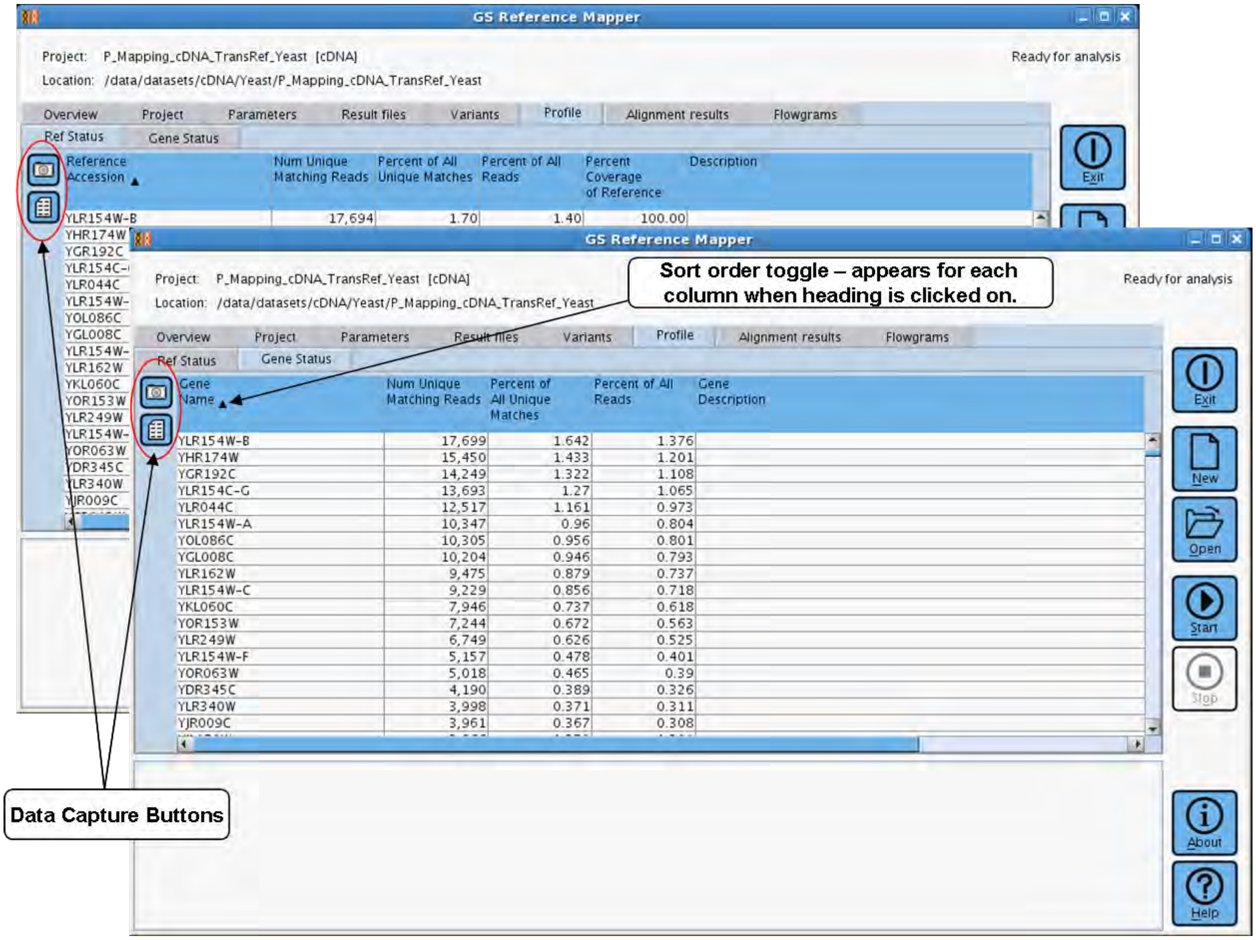
The detection of Rearrangement Regions depends on finding clusters of paired end reads considered to be False Pairs. When paired end libraries using different span distances are used (*e.g.* 3 kb and 8 kb or 20 kb libraries), more than one cluster may be found for the same structural variation.

2.11 Profile Tab

The Profile tab (Figure 67) contains sub-tabs for **Ref Status** and **Gene Status** (for cDNA Mapping projects only). These sub-tabs display statistics for reads mapping to each reference or gene. See Sections 2.18.1.12 and 2.18.2.1 for column descriptions.

Both sub-tabs contain tables that can be sorted:

- All columns are sortable by clicking on the column header. A small triangle appears to indicate the direction of the sort. Clicking on a column header of a column that is already sorted will reverse the order of the sort.
- If multiple rows have the same value for the sorted column, these rows stay in the order in which they were before the sort, relative to one another. This allows for “nested sorting”, as described in Section 2.10.



Project: P_Mapping_cDNA_TransRef_Yeast [cDNA]
Location: /data/datasets/cDNA/Yeast/P_Mapping_cDNA_TransRef_Yeast
Ready for analysis

Reference Accession ▲	Num Unique Matching Reads	Percent of All Unique Matches	Percent of All Reads	Percent Coverage of Reference	Description
YLR154W-B	17,694	1.70	1.40	100.00	
YHR174W					
YGR192C					
YLR154C-					
YLR044C					
YLR154W-					
YOL086C					
YGL008C					
YLR154W-					
YLR162W					
YKL060C					
YOR153W					
YLR249W					
YLR154W-					
YOR063W					
YDR345C					
YLR340W					
YJR009C					

Project: P_Mapping_cDNA_TransRef_Yeast [cDNA]
Location: /data/datasets/cDNA/Yeast/P_Mapping_cDNA_TransRef_Yeast
Ready for analysis

Gene Name ▲	Num Unique Matching Reads	Percent of All Unique Matches	Percent of All Reads	Gene Description
YLR154W-B	17,699	1.642	1.376	
YHR174W	15,450	1.433	1.201	
YGR192C	14,249	1.322	1.108	
YLR154C-G	13,693	1.27	1.065	
YLR044C	12,517	1.161	0.973	
YLR154W-A	10,347	0.96	0.804	
YOL086C	10,305	0.956	0.801	
YGL008C	10,204	0.946	0.793	
YLR162W	9,475	0.879	0.737	
YLR154W-C	9,229	0.856	0.718	
YKL060C	7,946	0.737	0.618	
YOR153W	7,244	0.672	0.563	
YLR249W	6,749	0.626	0.525	
YLR154W-F	5,157	0.478	0.401	
YOR063W	5,018	0.465	0.39	
YDR345C	4,190	0.389	0.326	
YLR340W	3,998	0.371	0.311	
YJR009C	3,961	0.367	0.308	

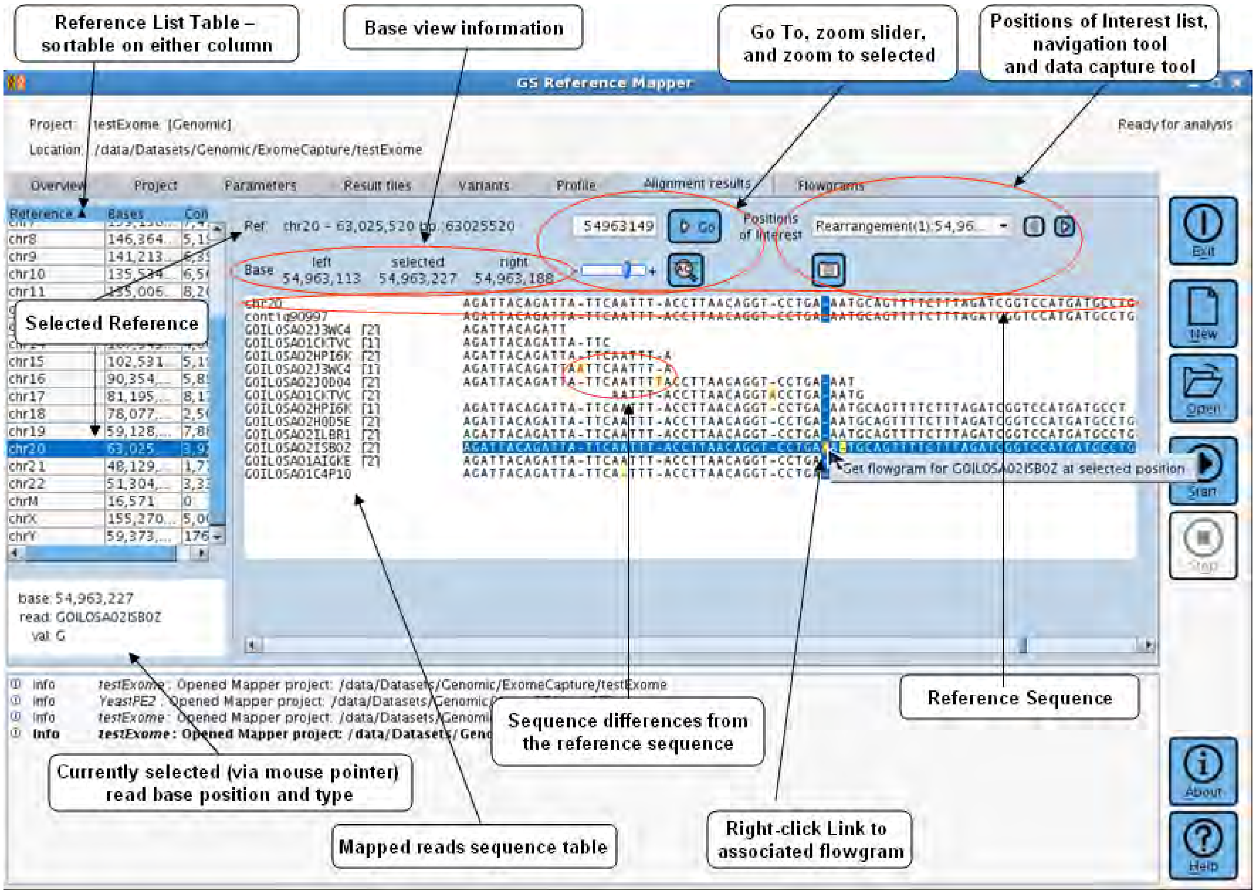
Sort order toggle – appears for each column when heading is clicked on.

Data Capture Buttons

Figure 67: gsMapper Profile Tab Ref Status and Gene Status Sub-tabs.

2.12 Viewing Reads Mapped to the Reference with the Alignment Results Tab

Click on the Alignment results tab to view the alignment data from the mapping computation (Figure 68). This tab shows the multiple alignments underlying the mapping of the reads to the reference.



Ready for analysis

Project: testExome [Genomic]
Location: /data/Datasets/Genomic/ExomeCapture/testExome

Reference	Bases	Con
chr8	146,364...	5,1
chr9	141,213...	6,3
chr10	135,534...	6,5
chr11	135,006...	8,2
chr15	102,531...	5,1
chr16	90,354,...	5,8
chr17	81,195,...	8,1
chr18	78,077,...	2,5
chr19	59,128,...	7,8
chr20	63,025,...	3,9
chr21	48,129,...	1,7
chr22	51,304,...	3,3
chrM	16,571	0
chrX	155,270...	5,0
chrY	59,373,...	176

Ref: chr20 - 63,025,520 bp.:63025520

Base	left	selected	right
	54,963,113	54,963,227	54,963,188

54963149

Rearrangement(1):54,96...

```
chr20                   AGATTACAGATTA-TTCAATTT-ACCTTAACAGGT-CCTGA-AATGCAGTTTTCTTTAGATCGGTCCATGATGCCTG
contig90997             AGATTACAGATTA-TTCAATTT-ACCTTAACAGGT-CCTGA-AATGCAGTTTTCTTTAGATCGGTCCATGATGCCTG
G0IL0SA02J3WC4  [2]     AGATTACAGATT
G0IL0SA01CKTVC  [1]     AGATTACAGATTA-TTC
G0IL0SA02HPI6K  [2]     AGATTACAGATTA-TTCAATTT-A
G0IL0SA02J3WC4  [1]     AGATTACAGATTAATTCAATTT-A
G0IL0SA02J0D04  [2]     AGATTACAGATTA-TTCAATTTTACCTTAACAGGT-CCTGA-AAT
G0IL0SA01CKTVC  [2]               AATTT-ACCTTAACAGGTACCTGA-AATG
G0IL0SA02HPI6K  [1]     AGATTACAGATTA-TTCAATTT-ACCTTAACAGGT-CCTGA-AATGCAGTTTTCTTTAGATCGGTCCATGATGCCT
G0IL0SA02H0D5E  [2]     AGATTACAGATTA-TTCAATTT-ACCTTAACAGGT-CCTGA-AATGCAGTTTTCTTTAGATCGGTCCATGATGCCTG
G0IL0SA02ILBR1  [2]     AGATTACAGATTA-TTCAATTT-ACCTTAACAGGT-CCTGA-AATGCAGTTTTCTTTAGATCGGTCCATGATGCCTG
G0IL0SA02ISB0Z  [2]     AGATTACAGATTA-TTCAATTT-ACCTTAACAGGT-CCTGAGAATGCAGTTTTCTTTAGATCGGTCCATGATGCCTG
G0IL0SA01AIGKE  [2]     AGATTACAGATTA-TTCAATTT-ACCTTAACAGGT-CCTGA
G0IL0SA01C4P10          AGATTACAGATTA-TTCA-TTT-ACCTTAACAGGT-CCTGA
```

Get flowgram for G0IL0SA02ISB0Z at selected position

base: 54,963,227
read: G0IL0SA02ISB0Z
val: G

```
Info   testExome : Opened Mapper project: /data/Datasets/Genomic/ExomeCapture/testExome
Info   YeastPE2 : Opened Mapper project: /data/Datasets/Genomic/
Info   testExome : Opened Mapper project: /data/Datasets/Genomi
Info   testExome : Opened Mapper project: /data/Datasets/Geno
```

Reference List Table – sortable on either column

Base view information

Go To, zoom slider, and zoom to selected

Positions of Interest list, navigation tool and data capture tool

Selected Reference

Reference Sequence

Sequence differences from the reference sequence

Currently selected (via mouse pointer) read base position and type

Mapped reads sequence table

Right-click Link to associated flowgram

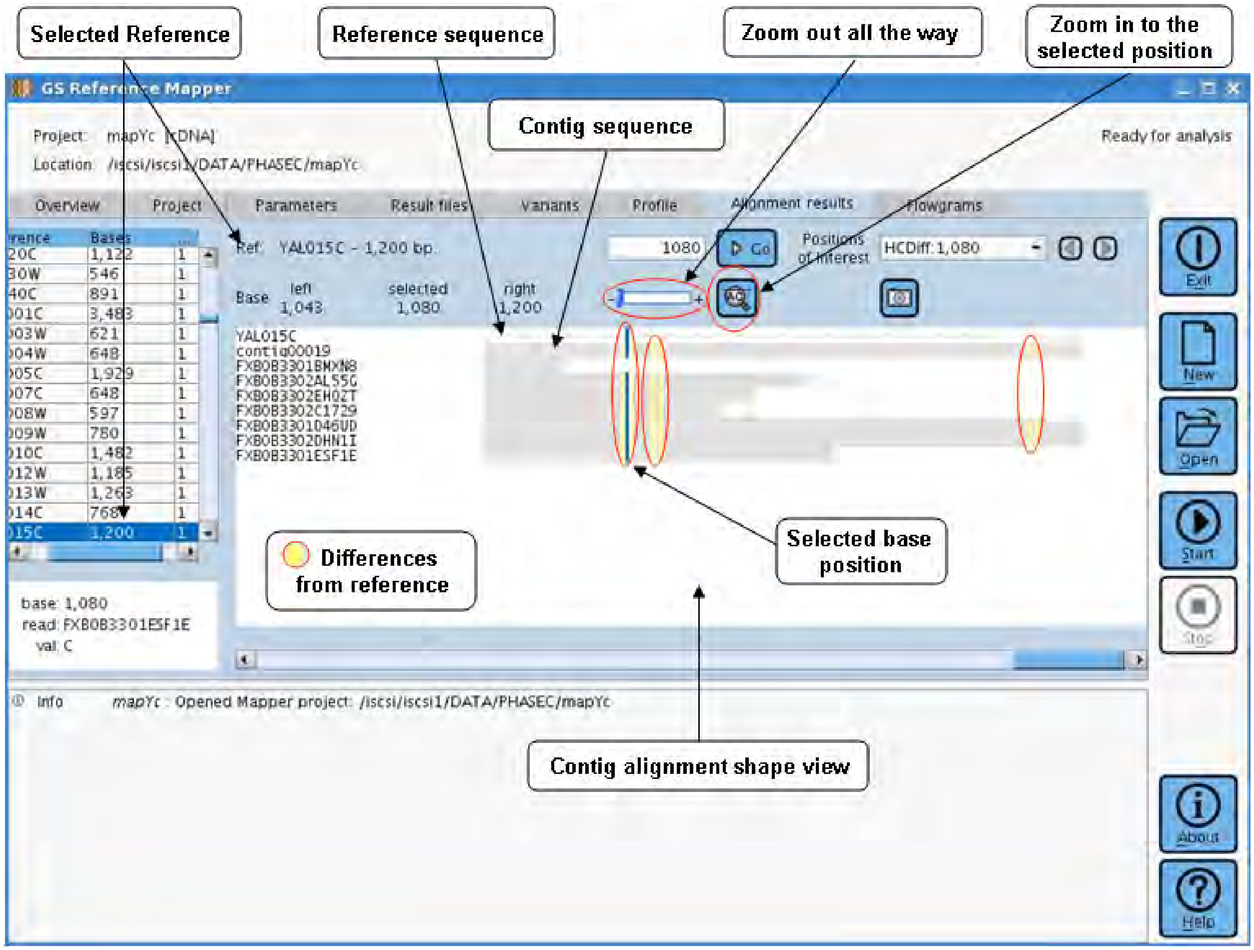
Figure 68: gsMapper Alignment results tab.

- The left-hand panel contains a sortable table of References.
- When an entry in that table is selected, the corresponding alignment results are displayed in the main window. The table columns include the accno, number of bases, and the number of contigs in each reference. Sorting by number of contigs (contiguous portions of the reference covered by at least a minimum number of reads) is helpful when the reads are expected to map to only a few of many reference sequences.
- The first row of the multiple alignment in the main window contains the reference sequence for the selected multiple alignment, gapped for insertions in the aligned reads.
- The successive lines display the individually aligned reads, gapped and showing bases that diverge from the consensus as red dashes on yellow background. The multi-nucleotide column display capability shows a SNP in a single column.
- Chimeric reads may participate in more than one multiple alignment. In order to distinguish the chimerically mapped segments of a read, a segment number between square brackets (*e.g.* [1], [2], etc) is appended to the accession number of the read in the alignment view.
- Scroll bars allow you to navigate the alignment manually:
- Clicking on the arrowheads on either side of a scroll bar scrolls the alignment by one base (horizontal) or one read (vertical) in the direction of the arrowhead
- Clicking on the light-colored area on either side of the horizontal scroll bar “handle” scrolls the alignment by 40 bases is the corresponding direction
- Clicking and dragging the “handle” of a scroll bar allows you to move larger distances rapidly
- For easy viewing, when a base position is selected, the base position column is highlighted so that individual variations may be easily identified.
- Right-clicking on a GS read will produce a “Get flowgram for … at selected position” menu item which, if selected, will activate the Flowgrams tab and display the flowgram for the read; centered on the flow corresponding to the base on which the user clicked to activate the option.

> This capability is not active for FASTA/FASTQ reads or the contig/consensus sequences themselves, for which no flowgrams are available.

- Above the Main window, the selected contig information is displayed. Below this is the base view information reporting the selected base and the left and right view positions in the current window.

- To the right of these information fields are tools for viewing and data capture of the main window alignments:
  - **Go To**: A data entry field and a **Go To** button allow you to navigate directly to a position of interest: enter a number not larger than the length of the selected contig in the data entry field, and click the **Go To** button.
  - The **zoom-slider** tool allows for more or fewer bases to be viewed at a time in the main window. If the slider is set to all the way zoom out (-) then the individual bases are grayed out but the shape of the read alignment is still shown along with the positions of the high quality differences highlighted in yellow. (Figure 69).
  - A '**zoom to selected**' button can be used to zoom in, centering on the selected base. If no base is selected, it will use the base in the center of the current view as a default.
  - **Camera icon**: saves an image of the part of the multi-alignment table currently visible on screen, to a file in .png format.



Project: mapYc [cDNA]
Location: /iscsi/iscsi1/DATA/PHASEC/mapYc

Ready for analysis

rence	Bases	...
20C	1,122	1
30W	546	1
40C	891	1
001C	3,483	1
003W	621	1
004W	648	1
005C	1,929	1
007C	648	1
008W	597	1
009W	780	1
010C	1,482	1
012W	1,185	1
013W	1,263	1
014C	768	1
015C	1,200	1

Ref: YAL015C - 1,200 bp

1080

Positions of interest: HCDiff: 1,080

Base left 1,043 selected 1,080 right 1,200

```
YAL015C
contig00019
FXB0B3301BMXN8
FXB0B3302AL55G
FXB0B3302EH0ZT
FXB0B3302C1729
FXB0B3301046UD
FXB0B3302DHN1I
FXB0B3301ESF1E
```

base: 1,080
read: FXB0B3301ESF1E
val: C

Info mapYc : Opened Mapper project: /iscsi/iscsi1/DATA/PHASEC/mapYc

Figure 69: gsMapper Zoom Slider - all the way out.

- The Alignment Results tab also features a "Mouse Tracker" area, in the left panel, below the list of reference sequences. If you pause the mouse pointer over a base in the multi-alignment, the following information is provided about that base:
  - **Base**: the position of that base relative to the selected reference sequence; gaps in the reference sequence are displayed as the following base of the reference
  - **Read**: the name of the read or contig to which this base belongs
  - **Val**: the "value" of the base under the mouse pointer, in that read, *i.e.* A, G, T, or C.

- Finally, a drop down menu of **Positions of Interest** exists near the upper right corner of the gsMapper Alignment results tab (Figure 70). This is populated with contig positions for a chosen reference or the position(s) from the HCDiff and HC Structural Variants tables. The Positions of interest list allows the user to "go to" base positions without having to type in the value and using the go to button. There are next and previous buttons to move through the list.

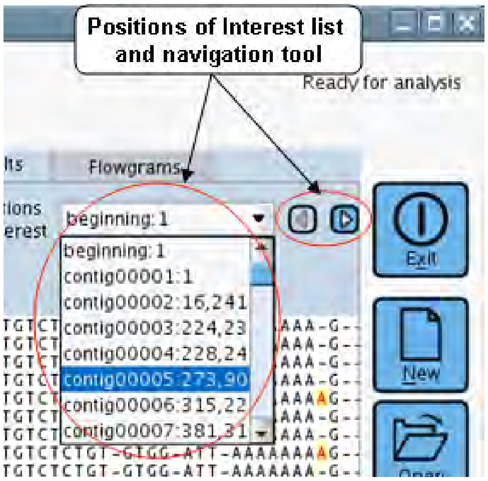
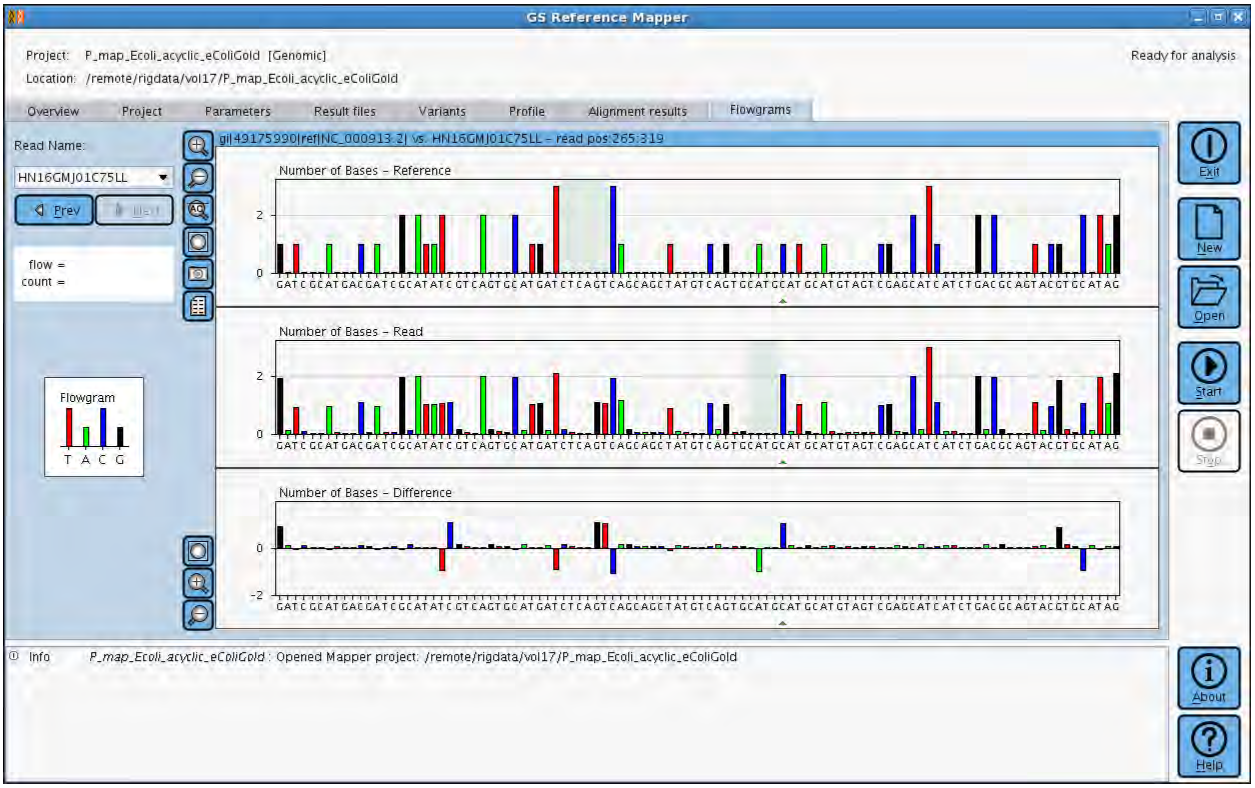


Figure 70: gsMapper Alignment Results Tab.

2.13 The Flowgrams Tab

When viewing a multiple alignment in the Alignment Results tab, the flowgram of any read can be displayed on the Flowgrams tab. Right-click on a base in a read of interest (*e.g.* at the location of an alignment difference), and select "Get flowgram for…" (Figure 68). This will display the selected read in the Flowgrams tab (Figure 71), with a small green triangle indicating the flow corresponding to the selected base.



Project: P_map_Ecoli_acyclic_eColiGold [Genomic]
Location: /remote/rigdata/vol17/P_map_Ecoli_acyclic_eColiGold

Ready for analysis

Read Name: HN16GMJ01C75LL

gi|49175990|ref|NC_000913.2| vs. HN16GMJ01C75LL – read pos:265:319

flow =
count =

Info P_map_Ecoli_acyclic_eColiGold : Opened Mapper project: /remote/rigdata/vol17/P_map_Ecoli_acyclic_eColiGold

Figure 71: gsMapper Flowgrams Tab, displaying acyclic flow pattern data with flowgram gaps.

The **Flowgrams tab** of GS Reference Mapper (and also GS *De Novo* Assembler) is divided into three panels.

- **Reference flowgram** – the top panel is an idealized (theoretical) flowgram for the selected reference sequence (or the consensus sequence when using GS *De Novo* Assembler).
- **Read flowgram** – the center panel is the flowgram for the selected read, displayed left-to-right for reads in the same orientation as the reference, or displayed right-to-left for reads in the opposite orientation.
- **Difference flowgram** – the bottom panel is a plot of the flow-by-flow differences between the two upper panels.

Reads sequenced in the opposite orientation to the reference are displayed as the reverse complement of the sequenced bases, reading from right-to-left. Interpretation of flowgrams often requires that you bear this in mind, especially for a read sequenced with an acyclic flow pattern. For example, the pattern of positive and negative flows might appear to be invalid unless you read them in the proper direction.

The top two panels display an alignment between the idealized flowgram of the reference and the actual flowgram of the read. The two flowgrams are aligned by minimizing the sum of the differences between the flowgram signals. The alignment algorithm introduces gray “flowgram gaps” in regions where the two flowgrams require a different number of flows to sequence through alignment differences at the location of a putative SNP or indel.

Two types of shaded flowgram gap are introduced, based on whether the theoretical reference flowgram requires fewer or more flows relative to the actual read flowgram. For cyclic flow patterns, flowgram gaps (also called cycle shifts) are introduced in one or more complete cycles.

- **Reference flowgram gap** – a block of negative flows identified in the reference flowgram that is associated with a putative SNP or indel that is ‘missing’ positive flows in the reference relative to the read. The shaded flows would have been classified as a dot (too many negative flows in a row) if the flowgram had been derived from an actual sequencing run. Subtracting these shaded flows results in a theoretical flow list that could have been used to generate the reference flowgram.
- **Read flowgram gap** – a block of negative flows inserted into the read flowgram that is associated with a putative SNP or indel that requires ‘extra’ positive flows in the reference relative to the read. The shaded flows are duplicated from the adjacent flow list to aid in alignment with the reference. Subtracting these shaded flows results in the actual flow list used to generate the read flowgram.

Dot – a block of negative flows (denoted as ‘N’ in a DNA sequence) that is ended by a positive flow of one of the nucleotides in the block, or started and ended by positive flows of the same nucleotide.

The difference flowgram, located in the bottom panel, highlights potential mapping variants or discrepancies from a reference (or consensus) sequence. Examining the read and difference flowgrams allows you to judge the validity of a homopolymer overcall or undercall by noting borderline signals and homopolymer order, bearing in mind that differences in homopolymer calls are intrinsically less certain as the length of the homopolymer increases. In general, differences involving only a single flow are less informative. Single nucleotide substitutions will always involve at least two flows; one each for the reference and read base. Other single nucleotide polymorphisms (SNPs) and small insertions or deletions (indels) may involve one or more flows, depending on the sequence context.

Differences that require a flowgram gap in order to align the read with the reference are particularly convincing, since such gaps would not be expected as a result of simple overcalling or undercalling, nor would they result from sequencing artifacts such as incomplete extension or carry forward (CAFIE errors). Thus, the association of flowgram gaps with alignment differences is an excellent way to increase confidence in potential variants (or discrepancies from consensus).

The Flowgram tab provides various navigation options and other features:

- The top-left corner of the tab provides two controls that allow the user to change the display to a different read:
    - A pull-down menu allows you to select from a list of all the reads in the contig to which the read currently displayed belongs (Figure 71), and which span the base position of the alignment that was used when initially launching the Flowgrams tab. That base position may be quickly updated by navigating back to the Alignment results tab and selecting a new alignment column (nucleotide in the sequence) by simply left-clicking in that column. Then switch back to the Flowgrams tab (by clicking on the tab). The drop down menu will be updated to include only those reads that have flowgrams and also which intersect with the new column of interest. The read previously displayed in the flowgram view will remain displayed (unless the new column of interest is not spanned by that read), but the green triangle will be updated to point to the flow corresponding to the nucleotide in the new column of interest. If the previously displayed read does not span the new column of interest, then the display will automatically go to the first read, topmost in the displayed alignment, which does span the column.
    - The pair of buttons, “Prev” and “Next”; allows you to change the display to the previous or the next read of the contig’s multiple alignment (in the order shown in the Alignment Results tab).
- You can also return to the Alignment Results tab and select a different read, from the alignment to the same or to any other contig.
- The plot navigation buttons appearing to the left of the flowgram have the following functions:

⊕	**Zoom in Y** – Zoom in by a factor of 1.5. For plots, this button zooms only the y-axis scale (use the **Zoom to labels** and **Freehand zooming** functions described below to zoom the x-axis).
⊖	**Zoom out Y** – Zoom out by a factor of 1.5. For plots, this will zoom only the y-axis scale and, unlike most zoom operations, this will zoom out past the data limits (to allow the user to get a better perspective of the data).
AC	**Zoom in labels** – This button zooms the x-axis of the flowgram so that the nucleotide/flow characters can fit below the axis.
◎	**Zoom fit** – Fit means ‘zoom all the way out.’ On plots, scale out to the limits of the data.
📷	**Output PNG** – Save a snapshot image of the current view to disk. This will open a dialog asking for a location and filename, and then will save a PNG image file at the location specified. The saved image contains only the visible region of the plot.
📄	**Output Text** – Save a tab (.txt) or comma-separated (.csv) text file of the data for this graphic. This will open a dialog asking for the location and filename to save the file. It then saves the data, along with summary information describing where the data came from. The data for all three plot subsections is saved to one file, with white space between subsection plot data.

- The lower panel (difference flowgram) has a separate set of zoom buttons (Zoom in Y, Zoom out Y, and Zom fit) because the scale of its Y axis (difference) is typically much smaller than that of the upper two panels.
- A mouse tracker area that shows the values for the nucleotide flow under the mouse pointer, when you pause it over the graph area:
- position of the nucleotide on the consensus or reference sequence,
- name of the nucleotide, and
- the “y” value on the plot

- number of bases added during that flow (idealized per the consensus [assembly] or reference sequence [mapping] or observed in the read), or
- the value of the "difference", where bases in the read that are absent in the consensus or reference are positive differences.
- There is also a free hand zoom function that allows you to zoom in to any area of a plot by drawing a box with the left mouse button. This function has the following unique features (not available in the corresponding GS Run Browser freehand zoom in functions) to facilitate the comparison between the three plots:
- zooming on the consensus/reference or on the read plots will apply to the Y axis of both those plots, and the X-axis of all three plots (including the difference plot).
- zooming on the difference plot zooms its Y axis alone, but zooms the X axis of all three plots (not shown).

2.14 Project Error Indicators

To assist the user in the proper setup of mapping, expansion of the error messaging is provided on the Parameters and Project tabs. Some examples are shown in the figures in section 4.12.

> When a project is first created and no read and reference files have yet been added, a warning icon on the Project Tab header will be displayed along with the status message 'Not ready for analysis'.

2.15 Multiplex GS Reference Mapper Projects

Multiplex Identifiers (MIDs) allow you to design an experiment whereby multiple libraries are prepared using distinct MID tags and sequenced together on the same PTP device (see Section 4.5). In an ordinary GS Reference Mapper project, analysis can be restricted to a subset of reads that have been tagged with one or more specific MIDs (see Section 2.6.3), but a separate project must be created for each such set of MIDs. Moreover, although each ordinary MID project can be viewed separately, it is not possible to view summary statistics across all of the projects.

A multiplex GS Reference Mapper project consists of a master project () and a group of related ordinary projects called sample projects (). A sample is a named association of MIDs with read files that represents a subset of reads that should be analyzed separately, for example each in a series of individual genomes to be mapped to a reference. The master project defines sample associations, reference sequence(s), and project parameters during project setup. After computation, it provides read and variation statistics across samples, and provides drill-down access to the underlying results and reports for each of the individual samples.

The two primary advantages of a multiplex mapping project over an ordinary project with MIDs are the substantial time savings in project setup and the ability to compare variants in unified reports across multiple samples. The primary restriction is that all of the sample projects must be computed using common reference and project parameters.

The GS Reference Mapper GUI is modified when a multiplex master project is created or opened. Similar to an ordinary mapping project, a multiplex master project has Overview, Project, Parameters and Result files tabs.

However, the detail data displayed on the Variants, Profile, Alignment Results, and Flowgrams tabs of an ordinary project are accessed in a multiplex master project by drilling down to individual sample projects in a hierarchical tree view on the Project tab. A new tab called Summary results displays unified read data and sequence variant statistics across all samples. Another new tab called Cross sample alignment simplifies comparison of sequence alignment across all of the samples in the multiplex project. Each individual sample project can also be opened as an ordinary GS Reference Mapper project in order to view all of the underlying detail.

2.15.1 Ordinary Project MIDs vs. Multiplex Project MIDs

An ordinary GS Reference Mapper project can filter a read file by MIDs when the read file is added to the project. The primary intended use for this feature is to allow reads identified by a single MID to be isolated from other MIDs in the read file for a mapping computation. However, it is also possible to combine (pool) reads containing several different MIDs together into a single analysis.

The primary intended use for a multiplex mapping project is to summarize results across a group of samples, where each sample is tagged by an MID sequence. In a single read file, the MID must uniquely identify the sample, but there are two very different scenarios when multiple read files are involved. In scenario #1, a sample is re-run using the original library preparation, so a given MID is associated with the same sample across multiple read files. In scenario #2, a panel of MIDs is associated with a different group of samples in each read file, so the MID is no longer a unique identifier of samples without also knowing the file of origin.

A multiplex project introduces the concept of “sample” in order to support each of these two major scenarios. In this context, a sample name is defined as the combination of an MID with one or more read files. Thus, a sample name ties together reads that share an MID across multiple read files (scenario #1), while distinguishing reads that share an MID across multiple read files (scenario #2). Note that in scenario #1, the MID names can serve as the sample names. This is the default when samples are configured automatically (see Section 2.15.3.1). In scenario #2, sample names must be unique. This is readily handled by appending a prefix or suffix to the automatically generated sample names (see Section 2.15.3.1), or by importing a pre-configured sample association file with sample names for each MID-read file combination (see Section 2.15.3.2).

Sample definition rules (GUI):

- All samples in a multiplex mapping project should use MIDs that were designed as part of the same compatible MID group.
- Sample names should not contain special characters (use only letters, numbers, underscores, and dashes).
- Sample names can be associated with multiple read files.
- Sample names can be associated with multiple MIDs.
- If a sample name is used repeatedly, the definitions will be merged.
- With multiple MIDs and multiple read files per sample, the sample names are associated with every combination of the cross between the multiple MIDs and multiple read files that were specified, even if only a subset of the combinations is actually associated with the sample.
- If two sample names share an MID-read file combination, an error will be reported and the project cannot be computed until the duplication is resolved.

It is possible for a multiplex project to associate multiple MIDs with a single sample name, for example to combine multiple MIDs within a single read file under a single sample name or to combine reads for a single sample that were associated with two different MIDs in two different library preparations. An ordinary mapping project can be used to filter a read file by multiple MIDs, which is the equivalent of defining multiple MIDs per sample.

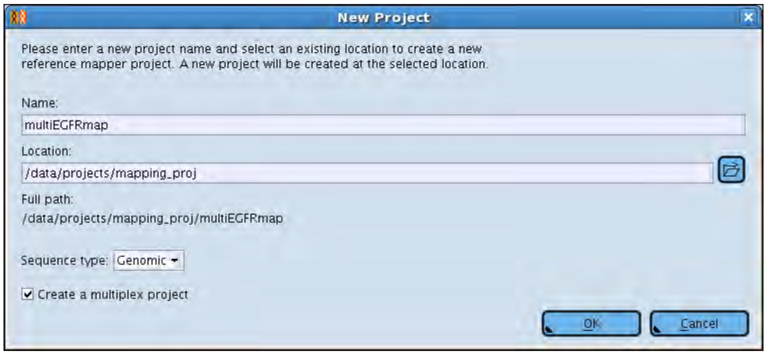
There are several ways to associate multiple MIDs with a single sample name.

- Rename multiple samples with the same name on the last screen of the Add Samples wizard (Section 2.15.3.1).
- Use a sample name on each of multiple passes through the Add Sample wizard, but associated with a different MID each time (Section 2.15.3.1).
- Import a sample association file that specifies multiple MIDs per sample name (Section 2.15.3.2).
- Specify multiple MIDs on the last screen of the Configure Sample wizard (Section 2.15.3.3).
- Create an ordinary mapping project and specify the MIDs of interest when adding the read file(s). This is equivalent to the use of addRun MID_List@sfffile from the CLI.
- Compute a multiplex project, open an individual sample project, add read files filtered by MID as above, re-compute the sample project and analyze the data in the sample project.

A multiplex GS Reference Mapper project is set up and computed with only a few differences from an ordinary project (see Sections 2.4 through 2.8 for details on setting up an ordinary mapping project). Most of the differences in project setup are found on the Project tab, where samples are defined by the association of sample names with MID sequences and read files.

2.15.2 Create a New Multiplex Mapping Project

Click on the **New Project** button in the toolbar on the right side of the screen, or click on the “New Mapping Project” text button in the **Quick Start** column to display a dialog in which you can specify the name and directory location for a new project (Figure 72). Check the “Create a multiplex project” checkbox to specify that the project being created is a multiplex master project.



Please enter a new project name and select an existing location to create a new reference mapper project. A new project will be created at the selected location.

Name:
multiEGFRmap

Location:
/data/projects/mapping_proj

Full path:
/data/projects/mapping_proj/multiEGFRmap

Sequence type: Genomic

☑ Create a multiplex project

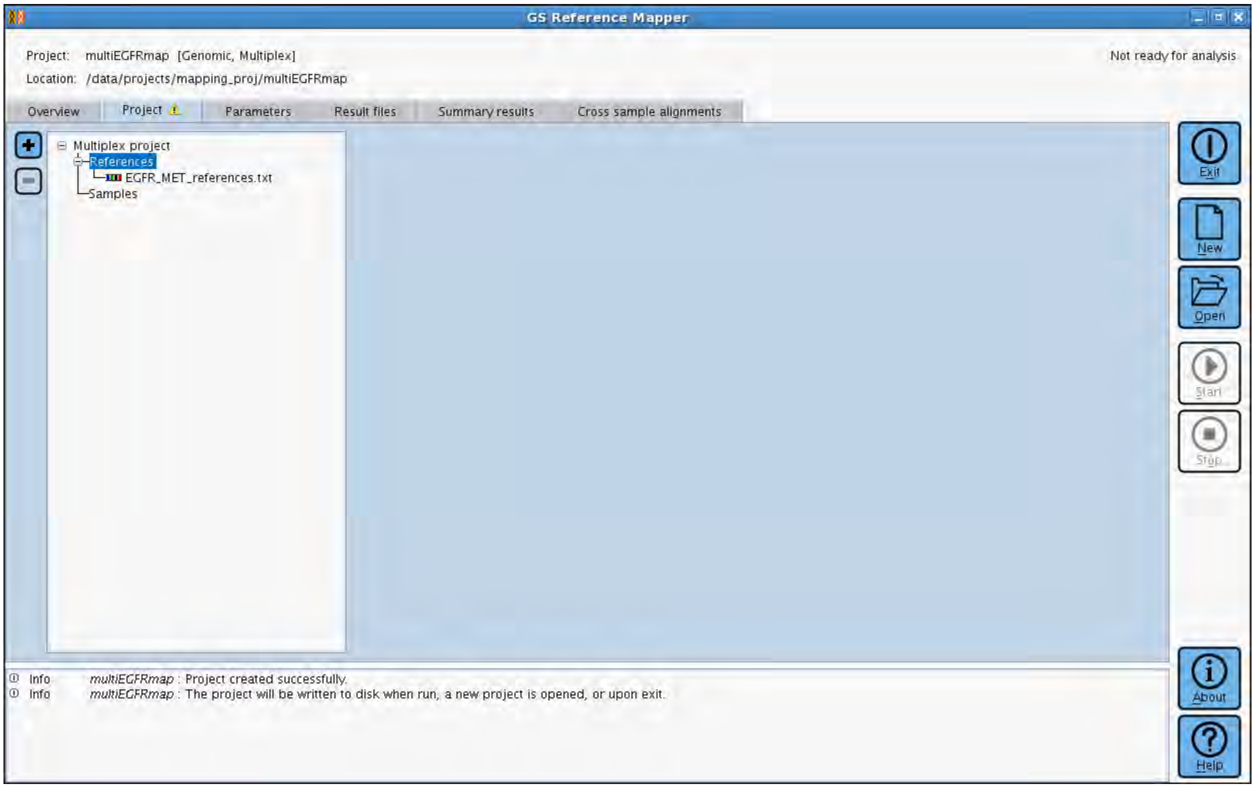
Figure 72: The New Project dialog for a multiplex project.

Note: ensure that no spaces are found anywhere in any path associated with a multiplexing project, including the path to SFF or reference files.

An abbreviated process for defining a multiplex mapping project is included below that emphasizes those aspects that are different from the process of defining an ordinary mapping project.

2.15.3 Add/Remove References and Samples

The Project tab in a multiplex project has a hierarchical tree view that displays all reference files and samples associated with the project. To specify the reference sequence, select References in the hierarchical tree, click the **Add reference file(s) to the project** button ⊞, and select one or more reference files for use by all of the sample projects. Once added, the reference file displays under References in the hierarchical tree on the left (Figure 73).



Project: multiEGFRmap [Genomic, Multiplex]
Location: /data/projects/mapping_proj/multiEGFRmap

Not ready for analysis

```
Info    multiEGFRmap : Project created successfully.
Info    multiEGFRmap : The project will be written to disk when run, a new project is opened, or upon exit.
```

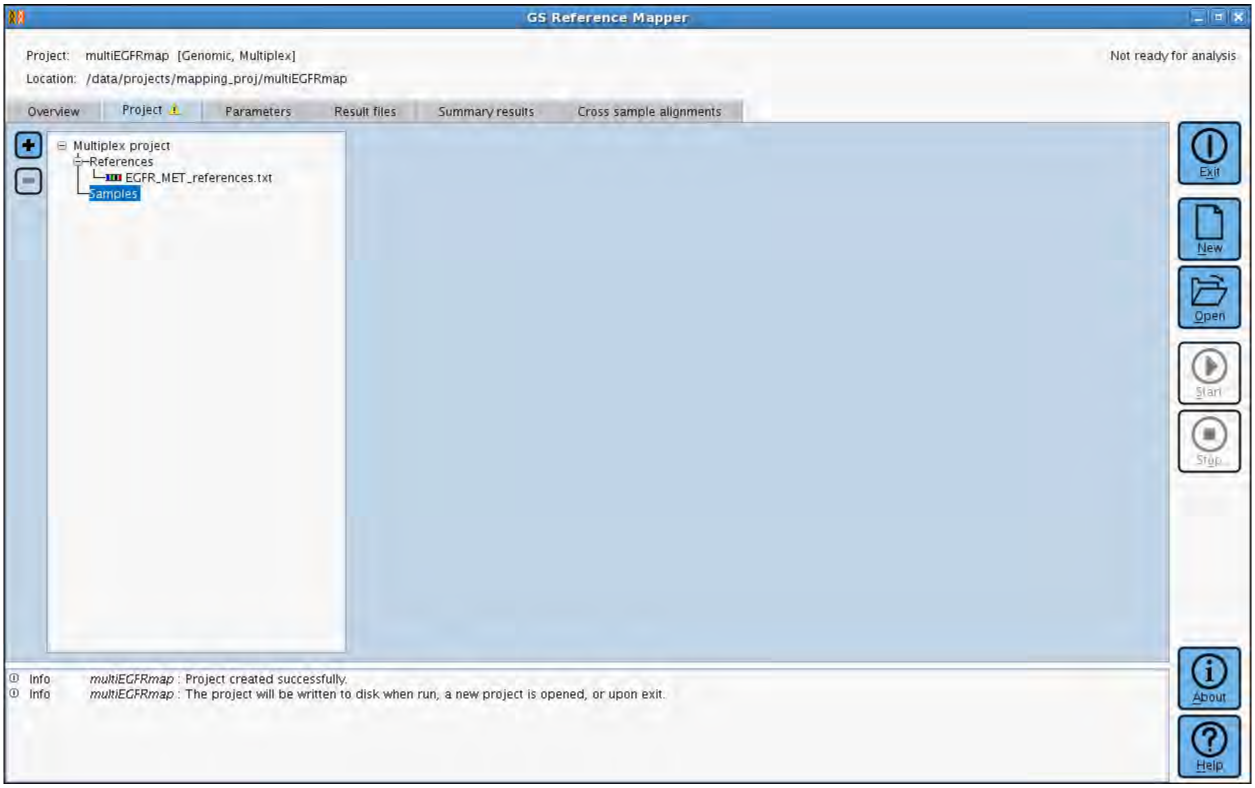
Figure 73: The Project tab, with a reference file added.

> All samples in a multiplex project are aligned against the single set of references that have been added to the project. However, it may be possible to map samples that align to different references by including the samples and their multiple references (*e.g.* for multiple genomes) in one combined multiplex master project.

There are three ways to add and configure samples using the Add Sample wizard ⊞, including: (1) sample names based on common MIDs defined in the default MID configuration file, (2) sample names based on custom MIDs defined in a custom MID configuration file, or (3) pre-configured sample names defined in a sample association file. Existing sample definitions can also be edited with the Configure Sample wizard ▦.

2.15.3.1 Configure Samples Automatically, Based on MID

The simplest way to define samples is by using one specific MID for each sample across one or more read files used in a multiplex project. On the Project tab (Figure 74), select Samples in the hierarchical tree and click the **Add Sample** wizard button +.



GS Reference Mapper

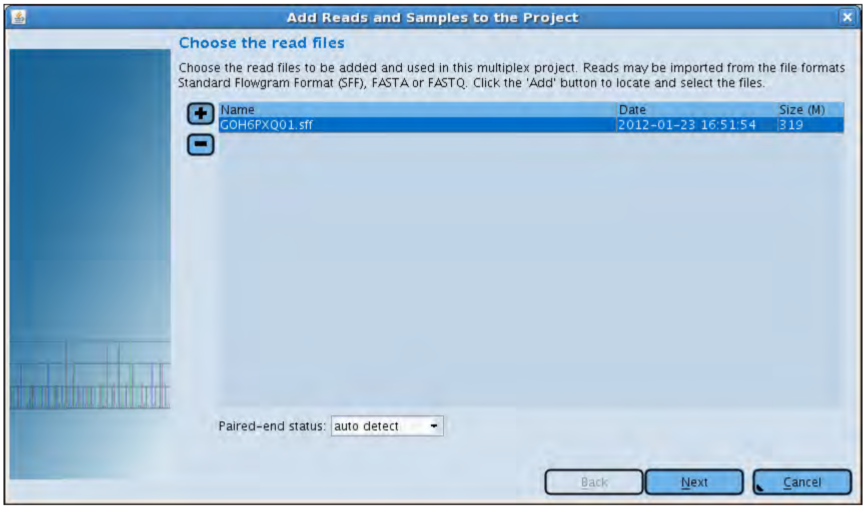
Project: multiEGFRmap [Genomic, Multiplex]
Location: /data/projects/mapping_proj/multiEGFRmap

Not ready for analysis

```
Info    multiEGFRmap : Project created successfully.
Info    multiEGFRmap : The project will be written to disk when run, a new project is opened, or upon exit.
```

Figure 74: The Project tab, with Samples selected in the hierarchical tree and ready to add new samples.

This opens a four step wizard to select and associate sample names with MIDs and read files. On the first screen of the wizard, select the read files to be associated with the sample by clicking the **Add read file(s)** button ⊞ and selecting one or more SFF, FASTA or FASTQ files (Figure 75). Add each of the read files that contains reads associated with the MIDs of interest. Click the Next button.



Add Reads and Samples to the Project

Choose the read files

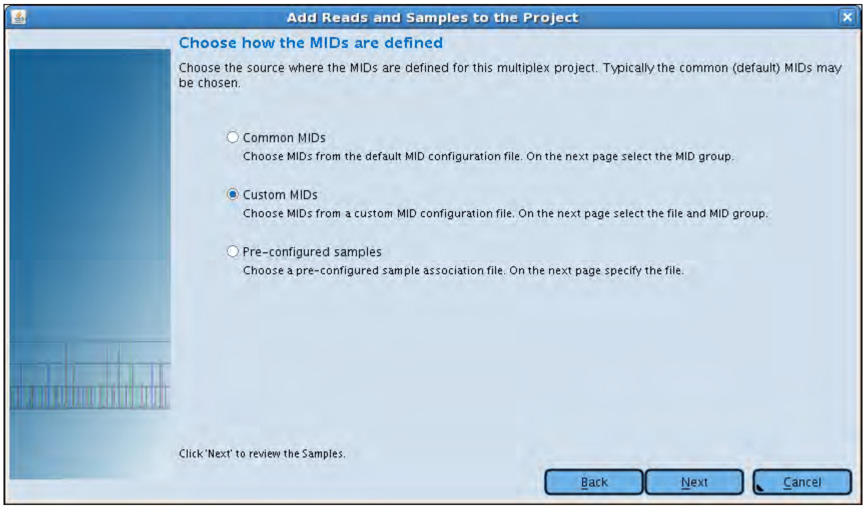
Choose the read files to be added and used in this multiplex project. Reads may be imported from the file formats Standard Flowgram Format (SFF), FASTA or FASTQ. Click the 'Add' button to locate and select the files.

Name	Date	Size (M)
GOH6PXQ01.sff	2012-01-23 16:51:54	319

Paired-end status: auto detect

Figure 75: The Add Sample wizard [Choose the read files], with a read file added.

Next, choose whether samples will be defined based on Common MIDs or Custom MIDs (Figure 76).



Add Reads and Samples to the Project

Choose how the MIDs are defined

Choose the source where the MIDs are defined for this multiplex project. Typically the common (default) MIDs may be chosen.

○ Common MIDs
Choose MIDs from the default MID configuration file. On the next page select the MID group.

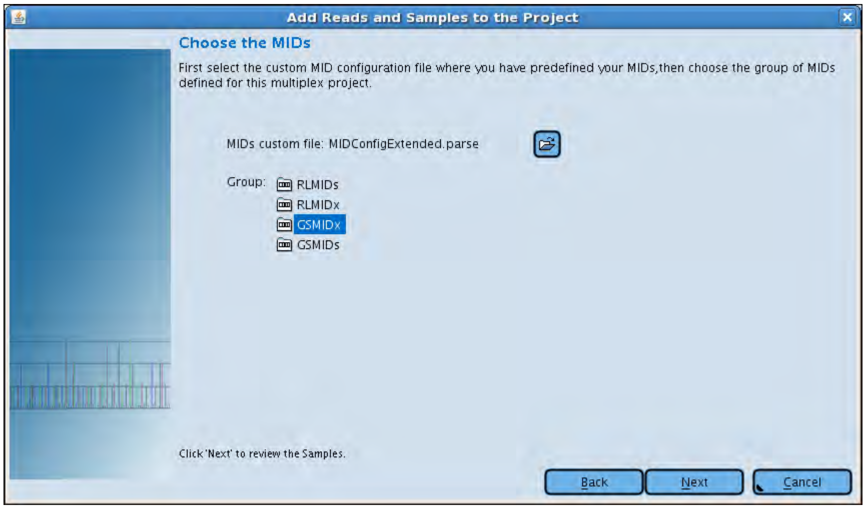
● Custom MIDs
Choose MIDs from a custom MID configuration file. On the next page select the file and MID group.

○ Pre-configured samples
Choose a pre-configured sample association file. On the next page specify the file.

Click 'Next' to review the Samples.

Figure 76: The Add Sample wizard [Choose how the MIDs are defined], with Custom MIDs selected.

Click the Next button, and select the MID group (*e.g.* GSMIDs or RLMIDs) that was associated with the sample at the time of library creation. The common MIDs are defined by the MIDConfig.parse file (see Glossary). If Custom MIDs was selected on the previous wizard screen, first click the **Browse MID config** button, choose the appropriate MID configuration file that defines the custom MIDs, and then select the appropriate MID group (Figure 77). Click the Next button.



Add Reads and Samples to the Project

Choose the MIDs

First select the custom MID configuration file where you have predefined your MIDs,then choose the group of MIDs defined for this multiplex project.

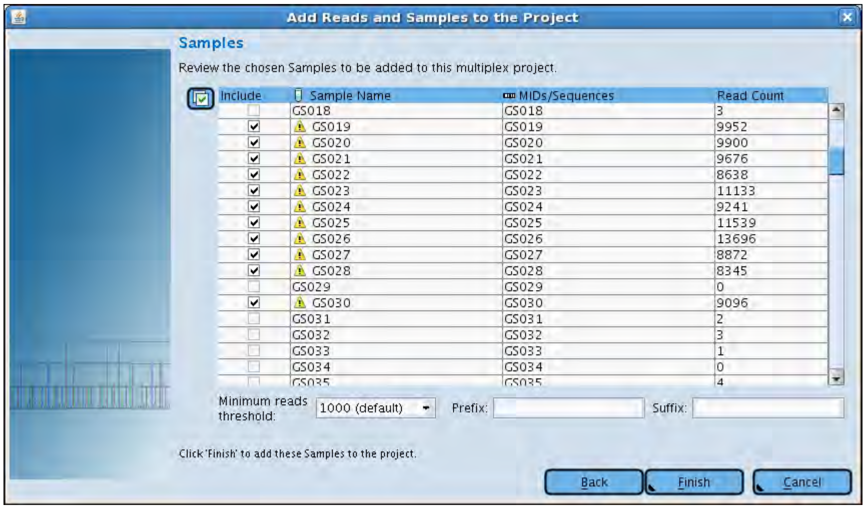
MIDs custom file: MIDConfigExtended.parse

Group:
- RLMIDs
- RLMIDx
- GSMIDx
- GSMIDs

Click 'Next' to review the Samples.

Figure 77: The Add Sample wizard [Choose the MIDs – Custom MIDs], custom MID configuration file and MID groups.

The fourth window of the wizard displays a table of sample names (the MID name, by default), a column displaying the MID name or sequence, and a count of the number of reads that appear to contain each MID across all selected read files (Figure 78). At the bottom of the wizard window is a Minimum reads threshold that allows you to disable samples that don't contain at least the minimum number of reads you want associated with each sample. This provides a very convenient way to identify the MIDs that are actually used in the project, and prevents the creation of a large number of sample projects that contain little or no data. Check one or more samples to be used by the multiplex project. At least one sample must exceed the Minimum reads threshold and be selected in order to finish the process of adding samples.



Add Reads and Samples to the Project

Samples

Review the chosen Samples to be added to this multiplex project.

Include	Sample Name	MIDs/Sequences	Read Count
□	GS018	GS018	3
☑	⚠ GS019	GS019	9952
☑	⚠ GS020	GS020	9900
☑	⚠ GS021	GS021	9676
☑	⚠ GS022	GS022	8638
☑	⚠ GS023	GS023	11133
☑	⚠ GS024	GS024	9241
☑	⚠ GS025	GS025	11539
☑	⚠ GS026	GS026	13696
☑	⚠ GS027	GS027	8872
☑	⚠ GS028	GS028	8345
□	GS029	GS029	0
☑	⚠ GS030	GS030	9096
□	GS031	GS031	2
□	GS032	GS032	3
□	GS033	GS033	1
□	GS034	GS034	0
□	GS035	GS035	4

Minimum reads threshold: 1000 (default) Prefix: Suffix:

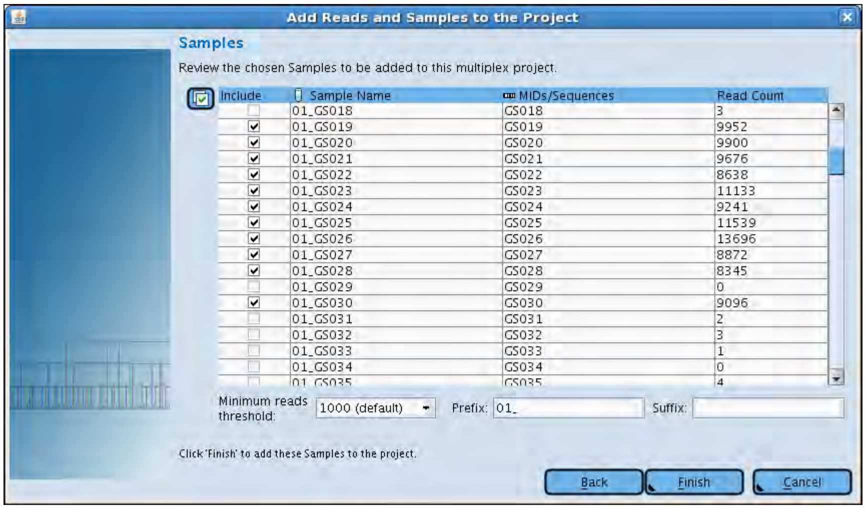
Click 'Finish' to add these Samples to the project.

Figure 78: The Add Sample wizard [Samples], with duplicate sample names).

If a sample name is already in use, a yellow triangle appears next to the sample name (Figure 78), with a tooltip that says "Sample is already in project, information will be merged.". If the sample definitions should be merged, simply click Finish. On the other hand, if the sample names should be unique, names can be edited by double-clicking and typing a new name. Alternatively, the entire group of sample names can be renamed in a single step by typing text into either the Prefix or Suffix box at the bottom of the window (Figure 79).

When a standard set of MIDs is used repeatedly with different groups of samples, for example in different runs on a GS Junior Instrument or in different regions of a PTP device, it is very important that each combination of MID and read file be given a distinctive sample name. The simplest way to accomplish this using the GUI is to make one pass through the Add Sample wizard for each region or group of runs associated with a single set of samples. For each group of samples, enter a prefix or suffix to identify the sample group, which will be added to the MID name.

Alternatively, you can import a series of sample association files with "real" sample names; one for each group of samples (see Section 2.15.3.2).



Samples

Review the chosen Samples to be added to this multiplex project.

Include	Sample Name	MIDs/Sequences	Read Count
□	01_GS018	GS018	3
☑	01_GS019	GS019	9952
☑	01_GS020	GS020	9900
☑	01_GS021	GS021	9676
☑	01_GS022	GS022	8638
☑	01_GS023	GS023	11133
☑	01_GS024	GS024	9241
☑	01_GS025	GS025	11539
☑	01_GS026	GS026	13696
☑	01_GS027	GS027	8872
☑	01_GS028	GS028	8345
□	01_GS029	GS029	0
☑	01_GS030	GS030	9096
□	01_GS031	GS031	2
□	01_GS032	GS032	3
□	01_GS033	GS033	1
□	01_GS034	GS034	0
□	01_GS035	GS035	4

Minimum reads threshold: 1000 (default)   Prefix: 01_   Suffix:

Click 'Finish' to add these Samples to the project.

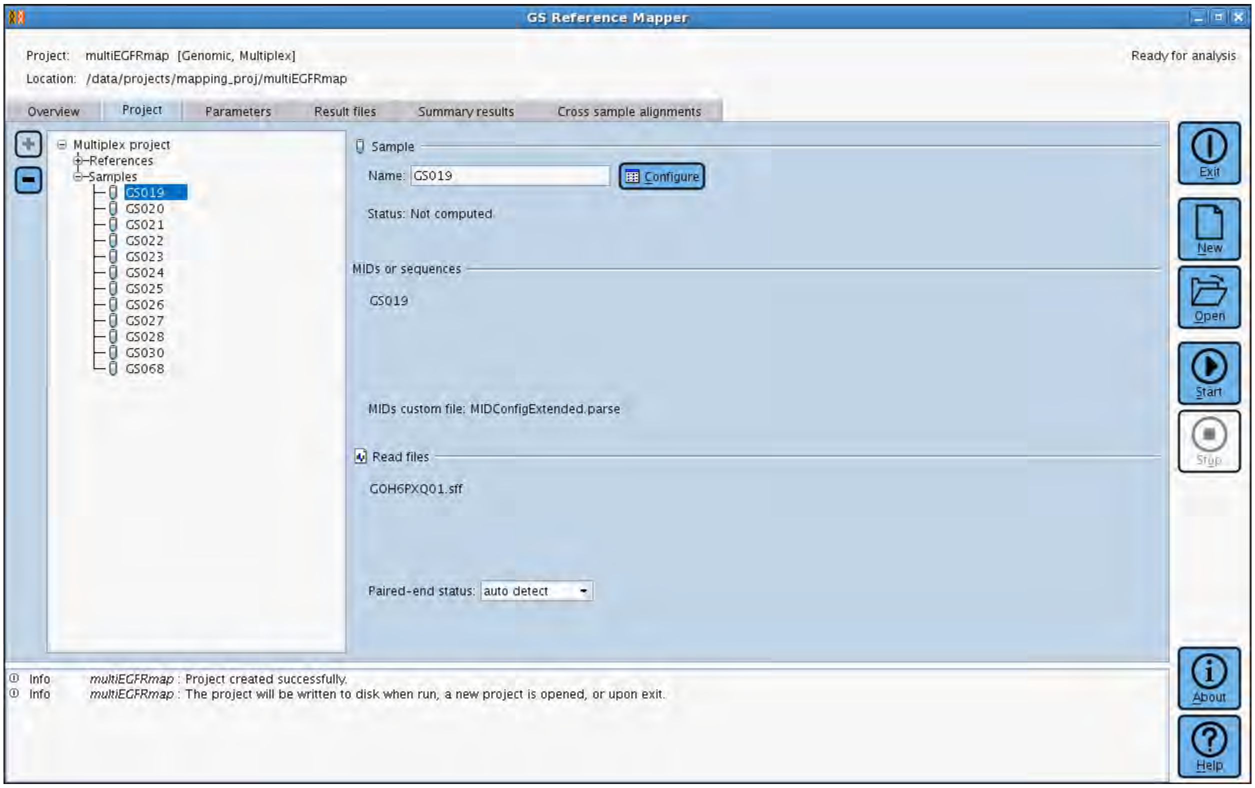
Figure 79: The Add Sample Wizard [Samples], rename samples.

The ability to merge sample definitions makes it possible to define multiple MIDs per sample name, either by making multiple passes through the wizard or by typing the same sample name for multiple MIDs on the fourth window of the wizard.

Be aware that when a sample name definition ends up with both multiple read files and multiple MIDs, the sample is associated with every combination of the cross between the multiple MIDs and multiple read files that were specified. This is true even if only a subset of the combinations was actually associated with the sample during library preparation.

If two samples end up with the same MID-read file combination, they will each be flagged in the hierarchical tree view using a red circle with a white "x" and a tooltip that says "MIDs – read files are duplicates of another sample". One or both of the samples with the duplicate MID-read file combination must be removed or edited before the project can be computed.

Click the Finish button to save the sample configuration. A summary of the sample configuration is displayed on the right side of the Project tab whenever the sample name is selected in the hierarchical tree on the left (Figure 80).



Project: multiEGFRmap [Genomic, Multiplex]
Location: /data/projects/mapping_proj/multiEGFRmap

Ready for analysis

Sample
Name: GS019
Status: Not computed

MIDs or sequences
GS019

MIDs custom file: MIDConfigExtended.parse

Read files
GOH6PXQ01.sff

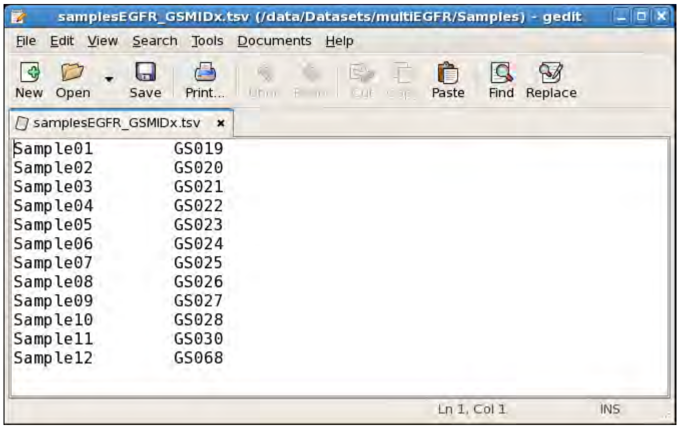
Paired-end status: auto detect

Info multiEGFRmap : Project created successfully.
Info multiEGFRmap : The project will be written to disk when run, a new project is opened, or upon exit.

Figure 80: Project tab, with sample configuration displayed.

2.15.3.2 Import Pre-Configured Samples

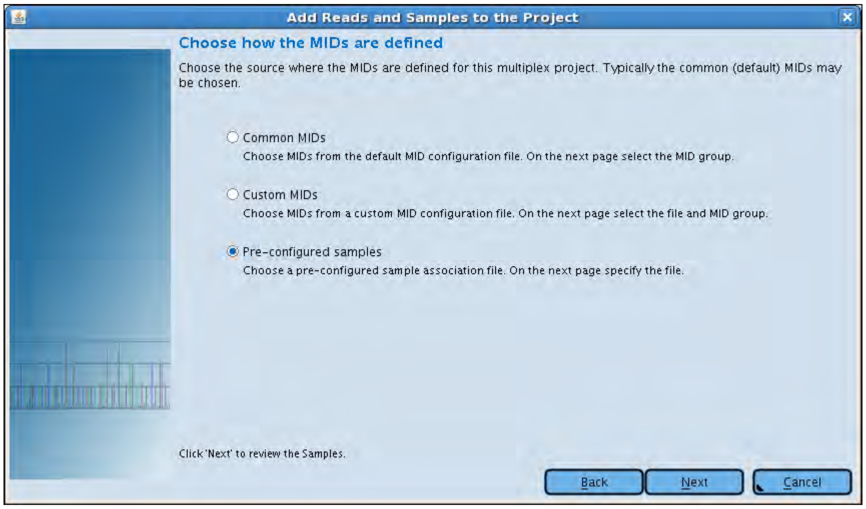
Samples can also be defined by importing a pre-configured sample association file, which provides greater flexibility in naming samples. Sample names and MIDs are stored in a two-column, tab-separated text file, which can be exported or saved from a LIMS, spreadsheet or text editor (Figure 81). The file imported by the GUI is a subset of the four-column sample association file used by the CLI addSample command (see Section 2.16.5.2 for details).


samplesEGFR_GSMIDx.tsv (/data/Datasets/multiEGFR/Samples) - gedit
File Edit View Search Tools Documents Help
New Open Save Print... Undo Redo Cut Copy Paste Find Replace
samplesEGFR_GSMIDx.tsv
Ln 1, Col 1
INS

```
Sample01        GS019
Sample02        GS020
Sample03        GS021
Sample04        GS022
Sample05        GS023
Sample06        GS024
Sample07        GS025
Sample08        GS026
Sample09        GS027
Sample10        GS028
Sample11        GS030
Sample12        GS068
```

Figure 81: Sample association file with two columns (sample name and MIDs).

As with the automatic sample configuration, select Samples in the hierarchical tree of the Project tab, click the **Add Sample** wizard button ⊞, add read files, and click the Next button. On the second screen of the wizard, choose Pre-configured samples (Figure 82), and click the Next button.



Choose how the MIDs are defined

Choose the source where the MIDs are defined for this multiplex project. Typically the common (default) MIDs may be chosen.

○ Common MIDs
Choose MIDs from the default MID configuration file. On the next page select the MID group.

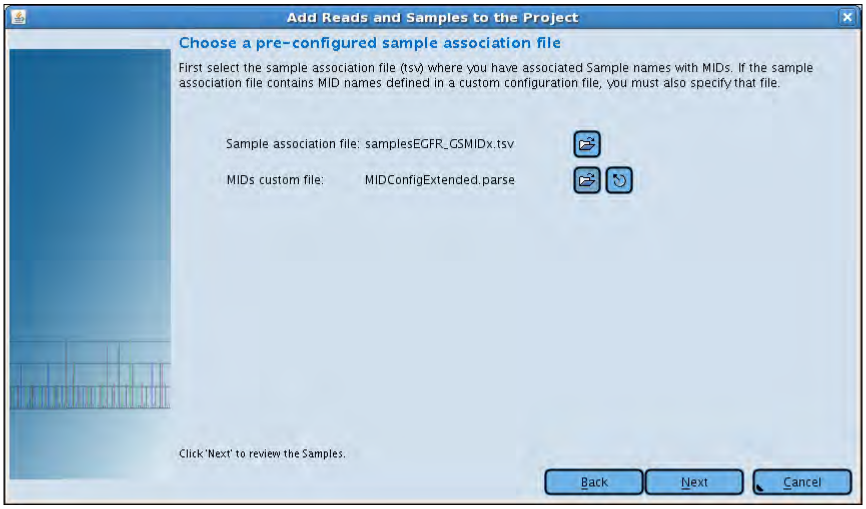
○ Custom MIDs
Choose MIDs from a custom MID configuration file. On the next page select the file and MID group.

● Pre-configured samples
Choose a pre-configured sample association file. On the next page specify the file.

Click 'Next' to review the Samples.

Figure 82: The Add Sample wizard [Choose how the MIDs are defined], with Pre-configured samples selected.

Click the **Browse sample association file** button 📂 and choose the sample association file to import (Figure 83). If MID sequences and allowed error counts are specified in the sample association file, no MID configuration file is necessary (because all relevant information if pulled from the sample association file). The same is true if the MID names in the sample association file are defined in the default MID configuration file. However, if the MID names are defined in a custom MID configuration file, you must select that file before continuing.



Choose a pre-configured sample association file

First select the sample association file (tsv) where you have associated Sample names with MIDs. If the sample association file contains MID names defined in a custom configuration file, you must also specify that file.

Sample association file: samplesEGFR_GSMIDx.tsv

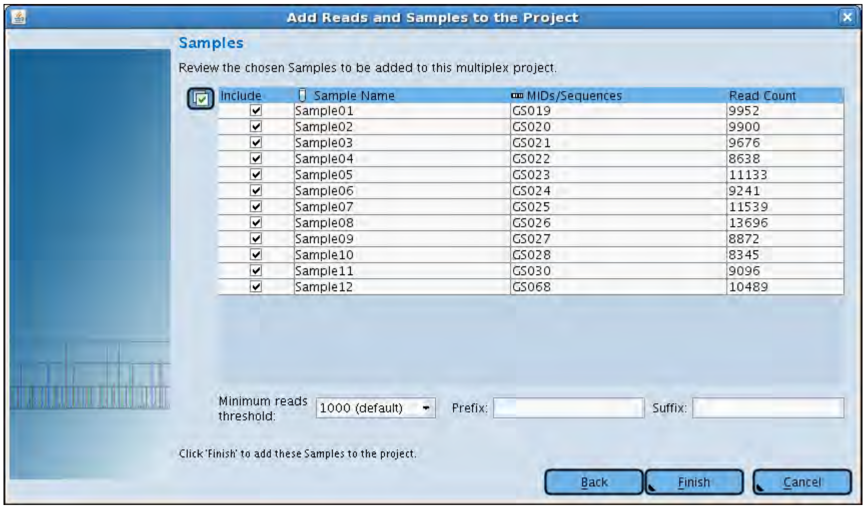
MIDs custom file: MIDConfigExtended.parse

Click 'Next' to review the Samples.

Figure 83: The Add Sample wizard [Choose a pre-configured sample association file], with custom MID configuration file.

It is not possible to select an MID group when a sample association file has been selected. The software will scan the selected (or default) MID configuration file for any matches to the MID names or sequences specified in the sample association file. Be sure that MID names and sequences in a custom configuration file are unique to avoid confusion. Also, never mix MIDs from incompatible MID groups into a single multiplex mapping project. MID groups such as GSMIDs and RSMIDs are designed with sufficient sequence "distance" between the individual MIDs to allow them to be recognized, even with substantial noise in the observed sequences.

The fourth window of the wizard works the same as with automatic sample configuration, except that the sample names and associated MIDs are pulled from the sample association file (Figure 84). Click the Finish button to save the sample configuration.



Samples

Review the chosen Samples to be added to this multiplex project.

Include	Sample Name	MIDs/Sequences	Read Count
☑	Sample01	GS019	9952
☑	Sample02	GS020	9900
☑	Sample03	GS021	9676
☑	Sample04	GS022	8638
☑	Sample05	GS023	11133
☑	Sample06	GS024	9241
☑	Sample07	GS025	11539
☑	Sample08	GS026	13696
☑	Sample09	GS027	8872
☑	Sample10	GS028	8345
☑	Sample11	GS030	9096
☑	Sample12	GS068	10489

Minimum reads threshold: 1000 (default)

Prefix:

Suffix:

Click 'Finish' to add these Samples to the project.

Figure 84: The Add Sample wizard [Samples], select samples to process.

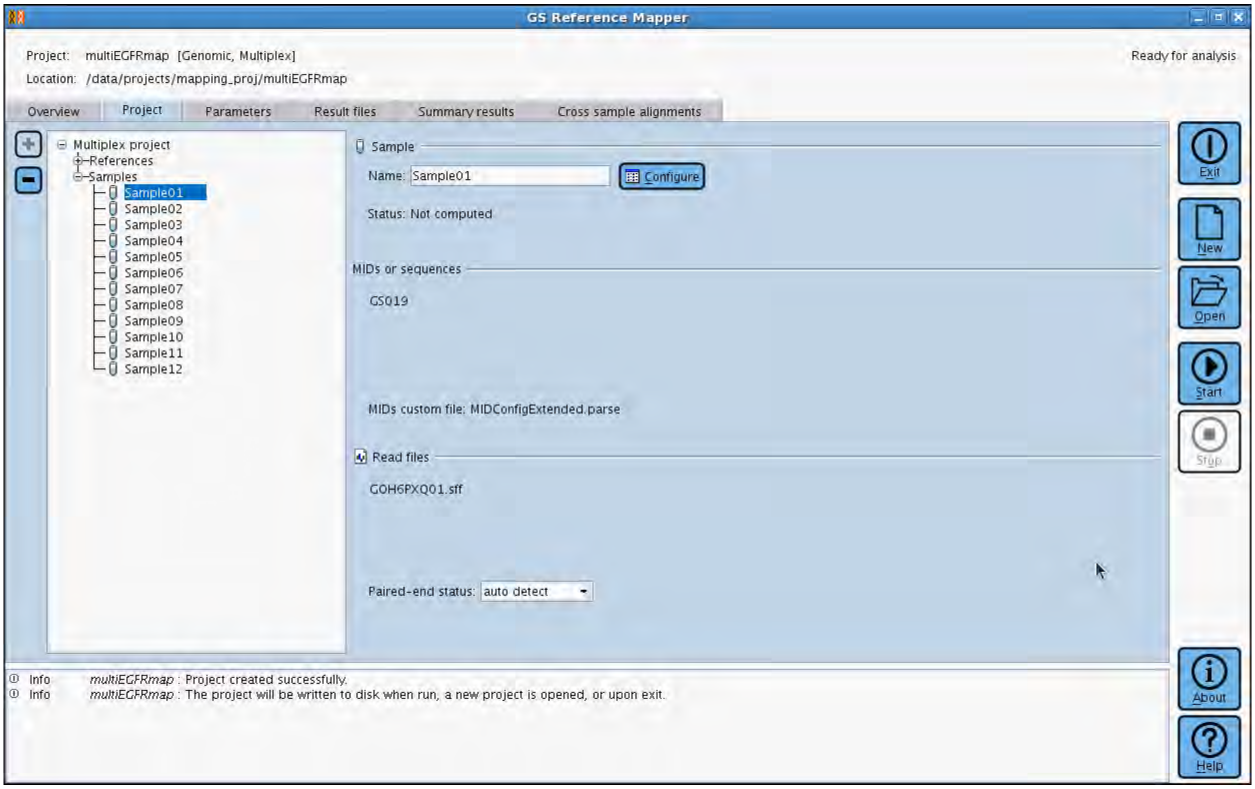
MID sequences can be defined directly within the sample association file by typing the sequence followed by "/#", where "#" is the maximum allowable errors. MIDs defined in this way will display "unknown" in the Read Count column on the fourth screen of the Add Sample wizard.

To use sample association files to rename samples from a panel or assay that re-uses MIDs across multiple runs, make one pass through the wizard for each run. Import the sample association file for that run to bring in the appropriate sample names.

This can also be accomplished using the CLI by creating a single four-column sample association file for the entire project, with the read file(s) for each sample stored in the third column.

2.15.3.3 Configure Existing Samples

Once a sample has been defined, the sample definition can be edited using the Configure Sample wizard. On the Project tab, select a sample name on the left and click the **Configure** button ▦ on the right (Figure 85).



GS Reference Mapper

Project: multiEGFRmap [Genomic, Multiplex]
Location: /data/projects/mapping_proj/multiEGFRmap

Ready for analysis

Overview | Project | Parameters | Result files | Summary results | Cross sample alignments

- Multiplex project
  - References
  - Samples
    - Sample01
    - Sample02
    - Sample03
    - Sample04
    - Sample05
    - Sample06
    - Sample07
    - Sample08
    - Sample09
    - Sample10
    - Sample11
    - Sample12

Sample

Name: Sample01 [Configure]

Status: Not computed

MIDs or sequences

GS019

MIDs custom file: MIDConfigExtended.parse

Read files

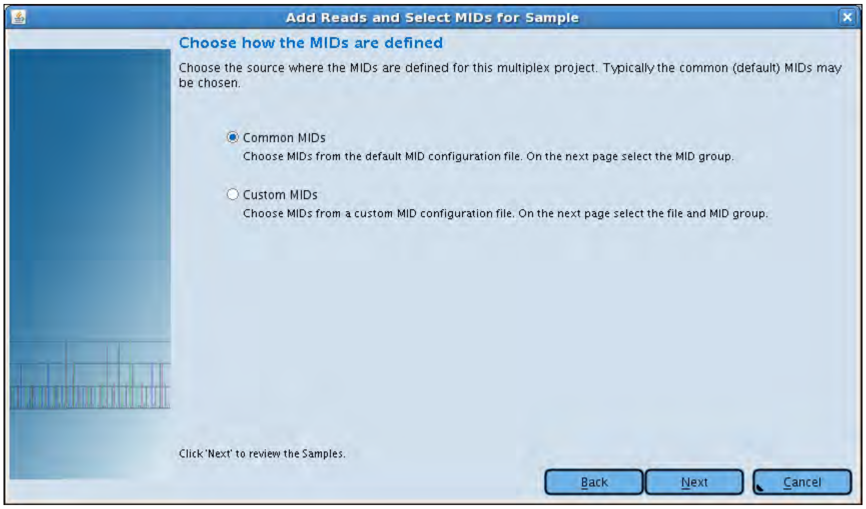
GOH6PXQ01.sff

Paired-end status: auto detect

Info *multiEGFRmap* : Project created successfully.
Info *multiEGFRmap* : The project will be written to disk when run, a new project is opened, or upon exit.

Figure 85: The Project tab, with a sample selected.

This opens a four step Configure Sample wizard. As with the Add Sample wizard, add or remove read files on the first screen of the wizard and click the Next button. On the second screen, select either Common MIDs or Custom MIDs (Figure 86).



Add Reads and Select MIDs for Sample

Choose how the MIDs are defined

Choose the source where the MIDs are defined for this multiplex project. Typically the common (default) MIDs may be chosen.

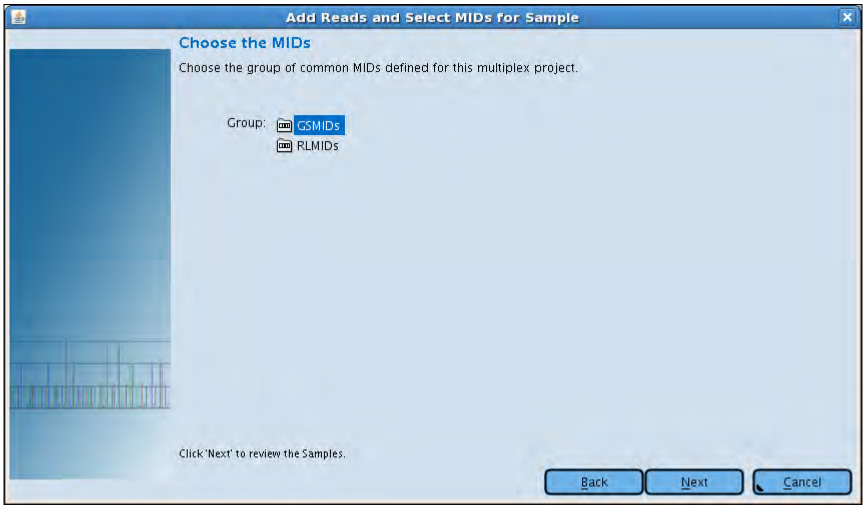
● Common MIDs
Choose MIDs from the default MID configuration file. On the next page select the MID group.

○ Custom MIDs
Choose MIDs from a custom MID configuration file. On the next page select the file and MID group.

Click 'Next' to review the Samples.

Figure 86: The Configure Sample wizard [Choose how the MIDs are defined], with Common MIDs selected.

Click the Next button, and select the MID group (*e.g.* GSMIDs or RLMIDs) that was associated with the sample at the time of library creation (Figure 87). If custom MID sequences were used, first click the **Browse MID config** button, choose the appropriate MID configuration file that defines the custom MIDs, and then select the MID group used. Click the Next button.



Add Reads and Select MIDs for Sample

Choose the MIDs

Choose the group of common MIDs defined for this multiplex project.

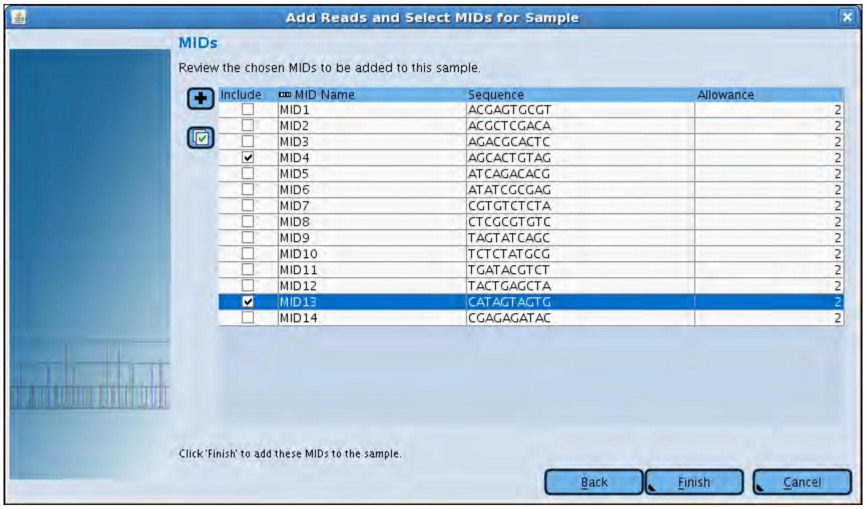
Group: GSMIDs
RLMIDs

Click 'Next' to review the Samples.

Figure 87: The Configure Sample wizard [Choose the MIDs - Common MIDs], using the default MID configuration file.

The fourth window of the wizard displays a table of the MID names and sequences from the selected MID group, along with the maximum number of allowable errors, as defined in the MID configuration file (Figure 88). Click the **Add MIDs** button ⊞ to add additional MIDs that are not defined in the selected MID configuration file (the MID sequence will be used as the MID name elsewhere in the project). Check the boxes adjacent to the multiple MIDs that should be combined together under the single sample name. Click the Finish button to save the edited sample configuration.

MID groups such as GSMIDs and RSMIDs are designed with sufficient sequence "distance" between the individual MIDs to allow them to be recognized, even with substantial noise in the observed sequences. Do not add MID sequences that are incompatible with the other MIDs defined by the selected MID configuration file and group used by the multiplex project.



MIDs

Review the chosen MIDs to be added to this sample.

Include	MID Name	Sequence	Allowance
□	MID1	ACGAGTGCGT	2
□	MID2	ACGCTCGACA	2
□	MID3	AGACGCACTC	2
☑	MID4	AGCACTGTAG	2
□	MID5	ATCAGACACG	2
□	MID6	ATATCGCGAG	2
□	MID7	CGTGTCTCTA	2
□	MID8	CTCGCGTGTC	2
□	MID9	TAGTATCAGC	2
□	MID10	TCTCTATGCG	2
□	MID11	TGATACGTCT	2
□	MID12	TACTGAGCTA	2
☑	MID13	CATAGTAGTG	2
□	MID14	CGAGAGATAC	2

Click 'Finish' to add these MIDs to the sample.

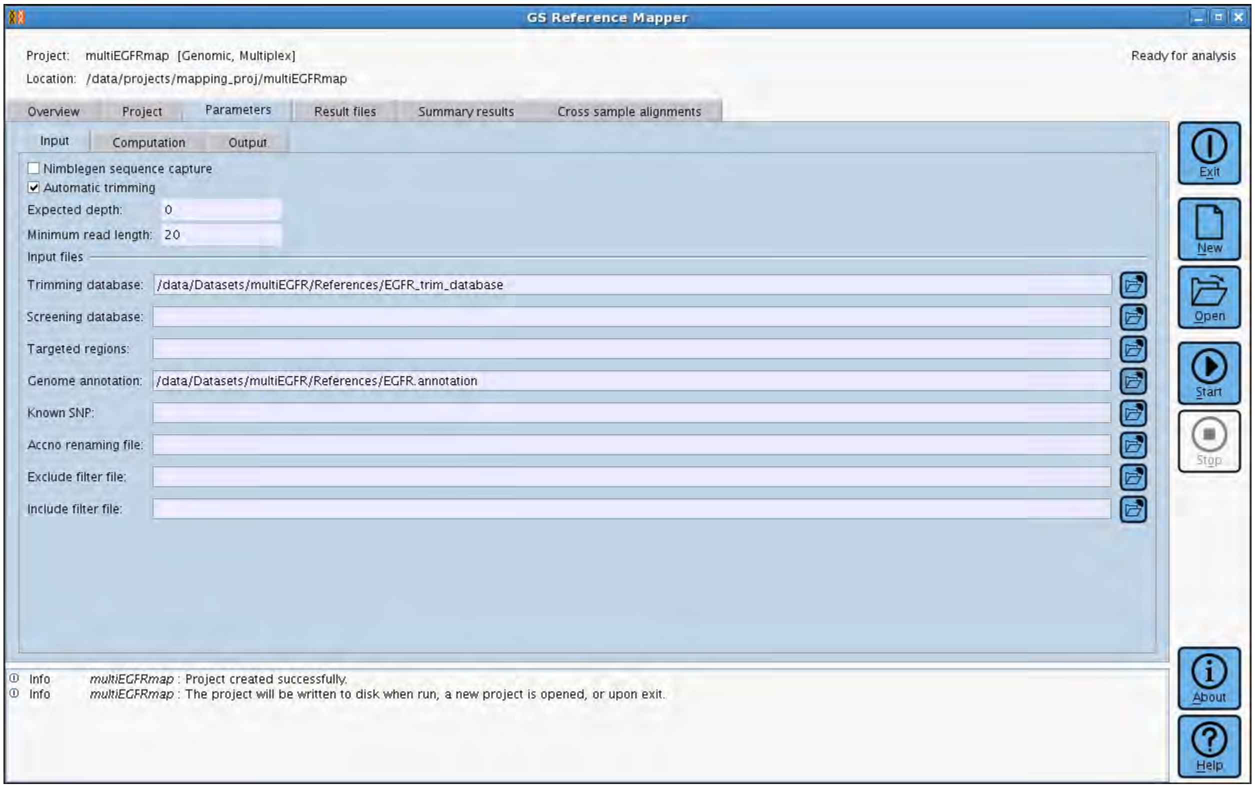
Figure 88: The Configure Sample wizard [MIDs], select multiple MIDs to associate with a sample.

Do not confuse this wizard screen with the final screen of the Add Samples wizard. The Configure Sample wizard allows you to associate multiple MIDs to a sample name or to add an additional MID sequence to those defined in the MID configuration file, but does not have the ability to filter MIDs by the number of reads.

2.15.4 Customize Multiplex Mapping Project with the Parameters Tab

2.15.4.1 Multiplex Project Input Sub-Tab

During multiplex project setup, the Input sub-tab is identical to that of an ordinary mapping project (Figure 88).



Project: multiEGFRmap [Genomic, Multiplex]
Location: /data/projects/mapping_proj/multiEGFRmap
Ready for analysis

□ Nimblegen sequence capture
☑ Automatic trimming
Expected depth: 0
Minimum read length: 20

Trimming database: /data/Datasets/multiEGFR/References/EGFR_trim_database
Screening database:
Targeted regions:
Genome annotation: /data/Datasets/multiEGFR/References/EGFR.annotation
Known SNP:
Accno renaming file:
Exclude filter file:
Include filter file:

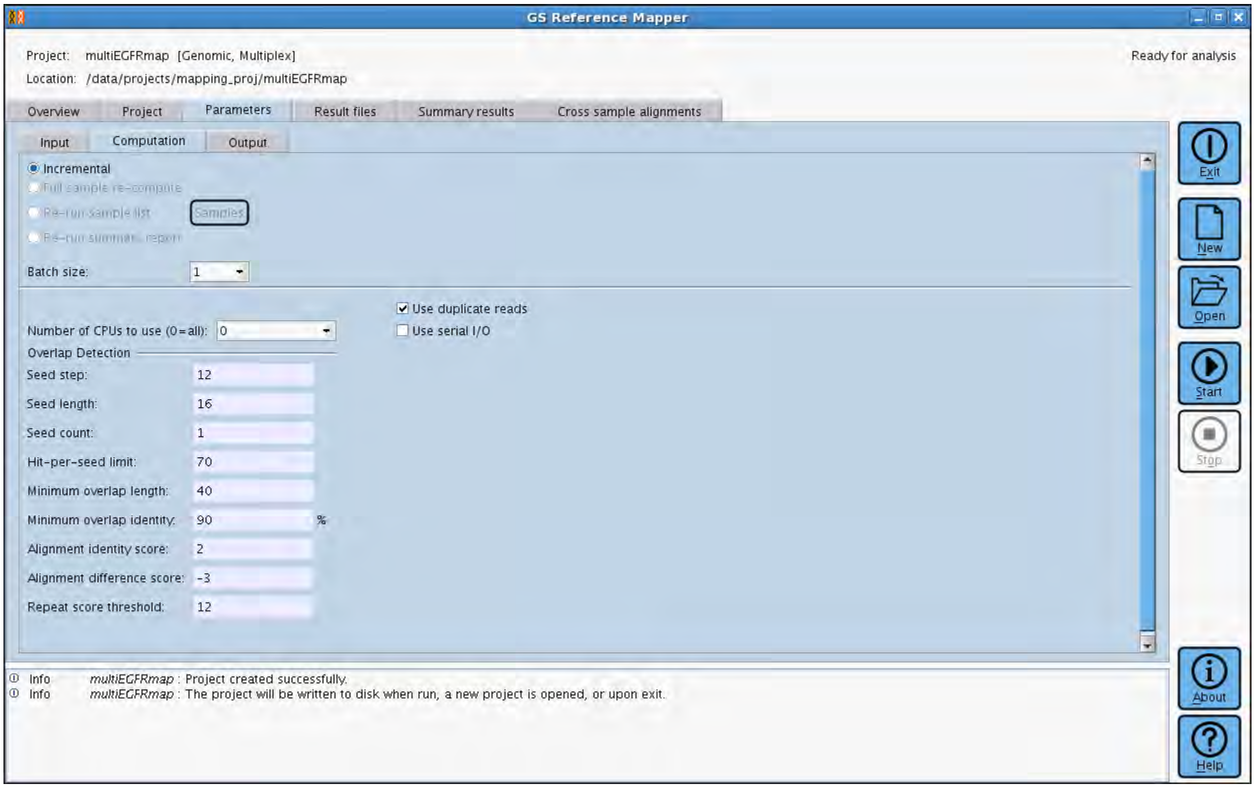
```
Info   multiEGFRmap : Project created successfully.
Info   multiEGFRmap : The project will be written to disk when run, a new project is opened, or upon exit.
```

Figure 89: Input sub-tab for a multiplex mapping project.

- Tip: Include a Genome annotation file to display additional information about known variants.
- Tip: Include a Trimming database file to trim primers, adapters, and other end sequences that would otherwise interfere with variant detection.

2.15.4.2 Multiplex Project Computation Sub-Tab

During multiplex project setup, there is one new option added to the Computation sub-tab (Figure 90).



Project: multiEGFRmap [Genomic, Multiplex]
Location: /data/projects/mapping_proj/multiEGFRmap
Ready for analysis

● Incremental
○ Full sample re-compute
○ Re-run sample list
○ Re-run summary report

Batch size: 1

☑ Use duplicate reads
☐ Use serial I/O

Number of CPUs to use (0=all): 0

Overlap Detection

Seed step:	12
Seed length:	16
Seed count:	1
Hit-per-seed limit:	70
Minimum overlap length:	40
Minimum overlap identity:	90 %
Alignment identity score:	2
Alignment difference score:	-3
Repeat score threshold:	12

```
Info    multiEGFRmap : Project created successfully.
Info    multiEGFRmap : The project will be written to disk when run, a new project is opened, or upon exit.
```

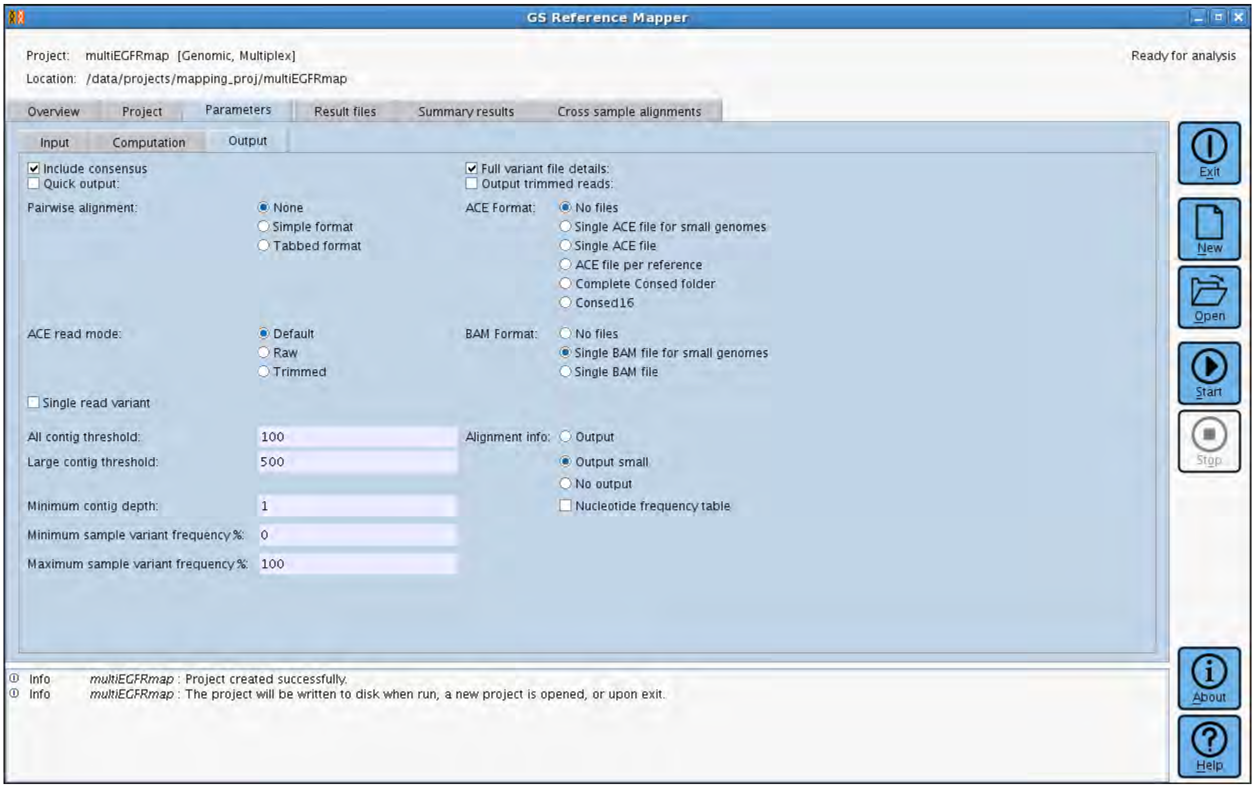
Figure 90: Computation sub-tab for a multiplex mapping project.

- **Batch size** (**-batchsize**) – Specifies how many sample projects will be run simultaneously.

- Tip: When analyzing high depth amplicon data in a multiplex mapping project, check "Use duplicate reads" (-ud) on the Computation sub-tab in order to view all reads instead of a smaller number of grouped reads.
- The "Number of CPUs to use" parameter refers to the number of CPUs per sample project. If used in conjunction with –batchsize, the number of CPUs may be reset so as not to exceed the total number available.

2.15.4.3 Multiplex Project Output Sub-Tab

During multiplex project setup, there are two new options added to the Output sub-tab (Figure 91).



GS Reference Mapper

Project: multiEGFRmap [Genomic, Multiplex]
Location: /data/projects/mapping_proj/multiEGFRmap

Ready for analysis

Overview | Project | Parameters | Result files | Summary results | Cross sample alignments

Input | Computation | Output

☑ Include consensus
□ Quick output:
☑ Full variant file details:
□ Output trimmed reads:

Pairwise alignment: ● None ○ Simple format ○ Tabbed format

ACE Format: ● No files ○ Single ACE file for small genomes ○ Single ACE file ○ ACE file per reference ○ Complete Consed folder ○ Consed16

ACE read mode: ● Default ○ Raw ○ Trimmed

BAM Format: ○ No files ● Single BAM file for small genomes ○ Single BAM file

□ Single read variant

All contig threshold: 100
Large contig threshold: 500
Minimum contig depth: 1
Minimum sample variant frequency %: 0
Maximum sample variant frequency %: 100

Alignment info: ○ Output ● Output small ○ No output
□ Nucleotide frequency table

Info *multiEGFRmap* : Project created successfully.
Info *multiEGFRmap* : The project will be written to disk when run, a new project is opened, or upon exit.

Figure 91: Output sub-tab for a multiplex mapping project.

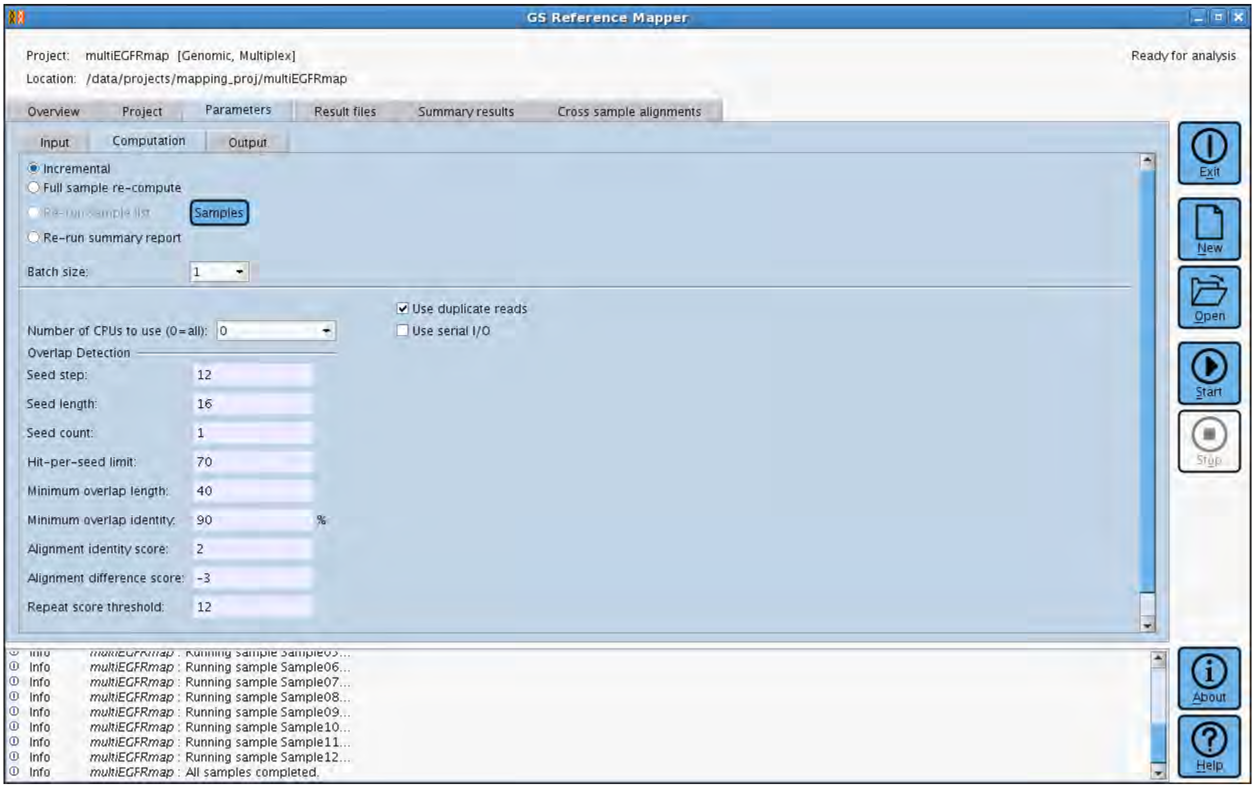
- **Minimum sample variant frequency percentage (-minsvf)** – Specifies the minimum variant frequency required to display a value in the All Variants and HC Variants sub-tabs of the Summary results tab and the 454AllSampleVariantFrequency.txt and 454HCSampleVariantFrequency.txt. This parameter has no impact on which variants are reported elsewhere or viewed in alignment results.
- **Maximum sample variant frequency percentage (-maxsvf)** – Specifies the maximum variant frequency required to display a value in the All Variants and HC Variants sub-tabs of the Summary results tab and the 454AllSampleVariantFrequency.txt and 454HCSampleVariantFrequency.txt. This parameter has no impact on which variants are reported elsewhere or viewed in alignment results.

- Tip: Check “Full variant file details” (-fd) on the Output sub-tab to include additional columns of data related to variants with sequence differences between samples and the reference that are defined in the Genome annotation file specified on the Input sub-tab. These will appear in the 454AllMultiplexReport.txt and 454HCMultiplexReport.txt, and on the related All Reports and HC Reports sub-tabs.

Changes to the parameters that control the range of variant frequency for display of variants in the All Variants and HC Variants sub-tabs will update when the project is re-computed using Re-run summary report (-mrerun report), as described in the next section (Section 2.15.5).

2.15.5 Re-Compute Multiplex Mapping Project

After a multiplex project has been computed, several new re-compute parameters become available on the Computation sub-tab (Figure 92).



Project: multiEGFRmap [Genomic, Multiplex]
Location: /data/projects/mapping_proj/multiEGFRmap
Ready for analysis

● Incremental
○ Full sample re-compute
○ Re-run sample list
○ Re-run summary report

Batch size: 1

☑ Use duplicate reads
□ Use serial I/O

Number of CPUs to use (0=all): 0

Overlap Detection

Seed step: 12
Seed length: 16
Seed count: 1
Hit-per-seed limit: 70
Minimum overlap length: 40
Minimum overlap identity: 90 %
Alignment identity score: 2
Alignment difference score: -3
Repeat score threshold: 12

```
Info    multiEGFRmap : Running sample Sample05...
Info    multiEGFRmap : Running sample Sample06...
Info    multiEGFRmap : Running sample Sample07...
Info    multiEGFRmap : Running sample Sample08...
Info    multiEGFRmap : Running sample Sample09...
Info    multiEGFRmap : Running sample Sample10...
Info    multiEGFRmap : Running sample Sample11...
Info    multiEGFRmap : Running sample Sample12...
Info    multiEGFRmap : All samples completed.
```

Figure 92: Computation sub-tab showing multiplex mapping project re-computation options.

- **Incremental** – Re-compute each sample project in incremental mode.
- **Full sample re-compute** – Re-compute each sample project in non-incremental mode, for example if a parameter needs to be changed across all sample projects.
- **Re-run sample list** (**-mrerun**) – Re-compute the selected list of sample projects in incremental mode.
- **Re-run summary report** (**-mrerun report**) – Re-generate all of the unified reports without re-computing the project, for example if the sample variant frequency thresholds have been changed.

Once a master multiplex project has been computed, references cannot be added or removed, and existing sample definitions cannot be edited. However, computed samples can be removed and new samples added.

2.15.6 View Unified Multiplex Mapping Result Files

A multiplex mapping project generates a series of unified result files containing the results of the multiplex mapping computation (see Table 2).

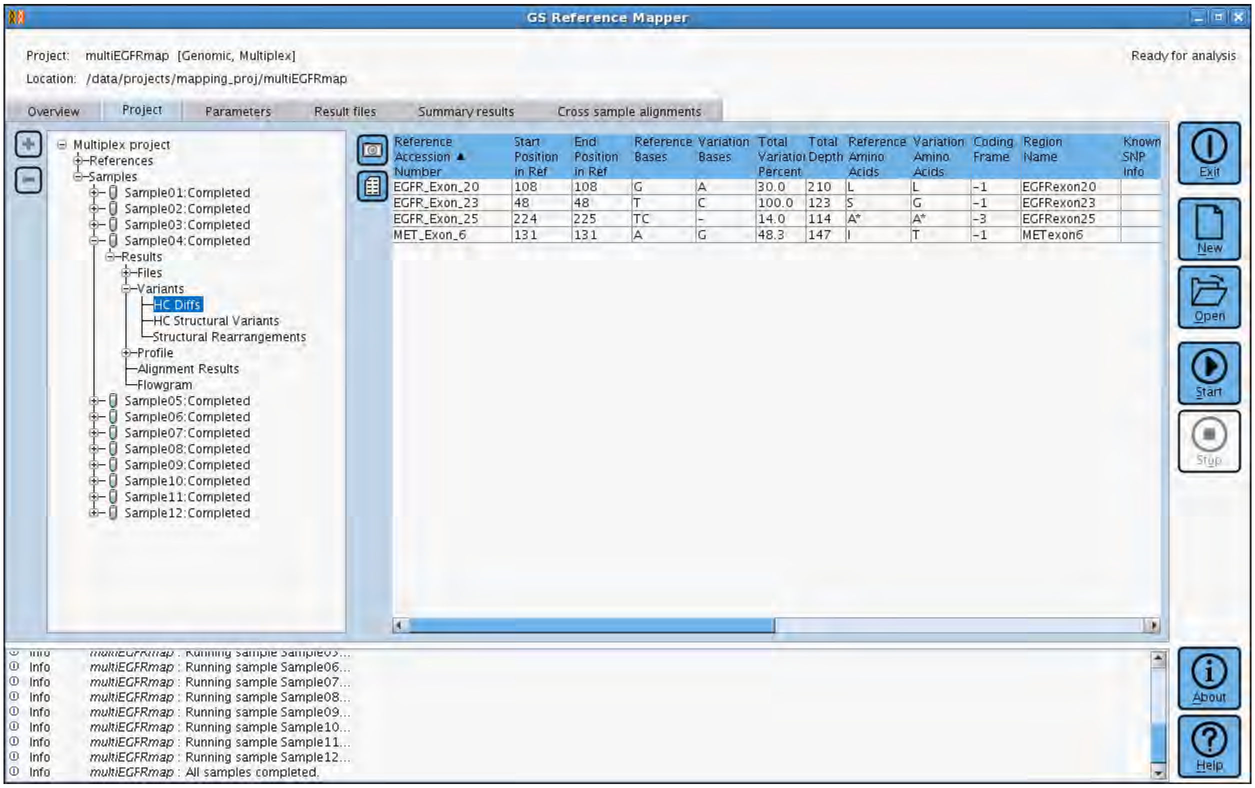
File name	Description
454AllMultiplexReport.txt	A consolidated report of the 454AllDiffs.txt summary lines for all sample projects, with variant summaries ordered by chromosome name and position. SNPs, insertion-deletion pairs, multi-homopolymer insertion or deletion regions, and single-base overcalls and undercalls are reported. These data are also displayed on the All Reports sub-tab.
454AllSampleVariantFrequency.txt	A pivoted version of the 454AllMultiplexReport.txt. Each sample has its own row, and each variant has its own column displaying frequency. These data are also displayed on the All Variants sub-tab.
454HCMultiplexReport.txt	A consolidated report of the high confidence variants of the 454HCDiffs.txt summary lines for all sample projects. These data are also displayed on the HC Reports sub-tab.
454HCSampleVariantFrequency.txt	A pivoted version of the 454HCMultiplexReport.txt. Each sample has its own row, and each variant has its own column displaying frequency. These data are also displayed on the HC Variants sub-tab.
454MultiplexMetrics.txt	A consolidated 454NewblerMetrics.txt report for all sample projects, including the number of input runs and reads, the number and size of the large consensus contigs and the number of all consensus contigs. Reference metrics are provided as well.
454NewblerProgress.txt	A text log of the messages sent to standard output during the multiplex mapping computation.

Table 2: GS Reference Mapper Multiplex Project Unified Result Files.

2.15.7 View Individual Sample Project Results

The multiplex master project does not have the full size Variants, Profile, Alignment Results, and Flowgrams tabs that are found in ordinary projects. However, the results from these tabs can still be displayed for each sample project from within the multiplex master project. Sample project results are accessed by drilling down to individual sample projects in a hierarchical tree view on the Project tab.

After a project has been computed, the hierarchical tree view on the Project tab includes additional levels of information (see Figure 93). Expand the tree view by clicking the small plus signs to view the results for a specific sample project. Each of the missing tabs (Variants, Profile, Alignment Results and Flowgram) and detail reports from the Result files tab can be viewed on the right side of the screen by clicking the appropriate level in the tree view on the left side of the screen.



Project: multiEGFRmap [Genomic, Multiplex]

Location: /data/projects/mapping_proj/multiEGFRmap

Ready for analysis

Reference Accession ▲ Number	Start Position in Ref	End Position in Ref	Reference Bases	Variation Bases	Total Variatio Percent	Total Depth	Reference Amino Acids	Variation Amino Acids	Coding Frame	Region Name	Known SNP Info
EGFR_Exon_20	108	108	G	A	30.0	210	L	L	-1	EGFRexon20	
EGFR_Exon_23	48	48	T	C	100.0	123	S	G	-1	EGFRexon23	
EGFR_Exon_25	224	225	TC	-	14.0	114	A*	A*	-3	EGFRexon25	
MET_Exon_6	131	131	A	G	48.3	147	I	T	-1	METexon6	

```
Info   multiEGFRmap : Running sample Sample05...
Info   multiEGFRmap : Running sample Sample06...
Info   multiEGFRmap : Running sample Sample07...
Info   multiEGFRmap : Running sample Sample08...
Info   multiEGFRmap : Running sample Sample09...
Info   multiEGFRmap : Running sample Sample10...
Info   multiEGFRmap : Running sample Sample11...
Info   multiEGFRmap : Running sample Sample12...
Info   multiEGFRmap : All samples completed.
```

Figure 93: Project tab hierarchical tree after computation, displaying drill-down sample project results.

2.15.8 View Multiplex Mapping Summary Results

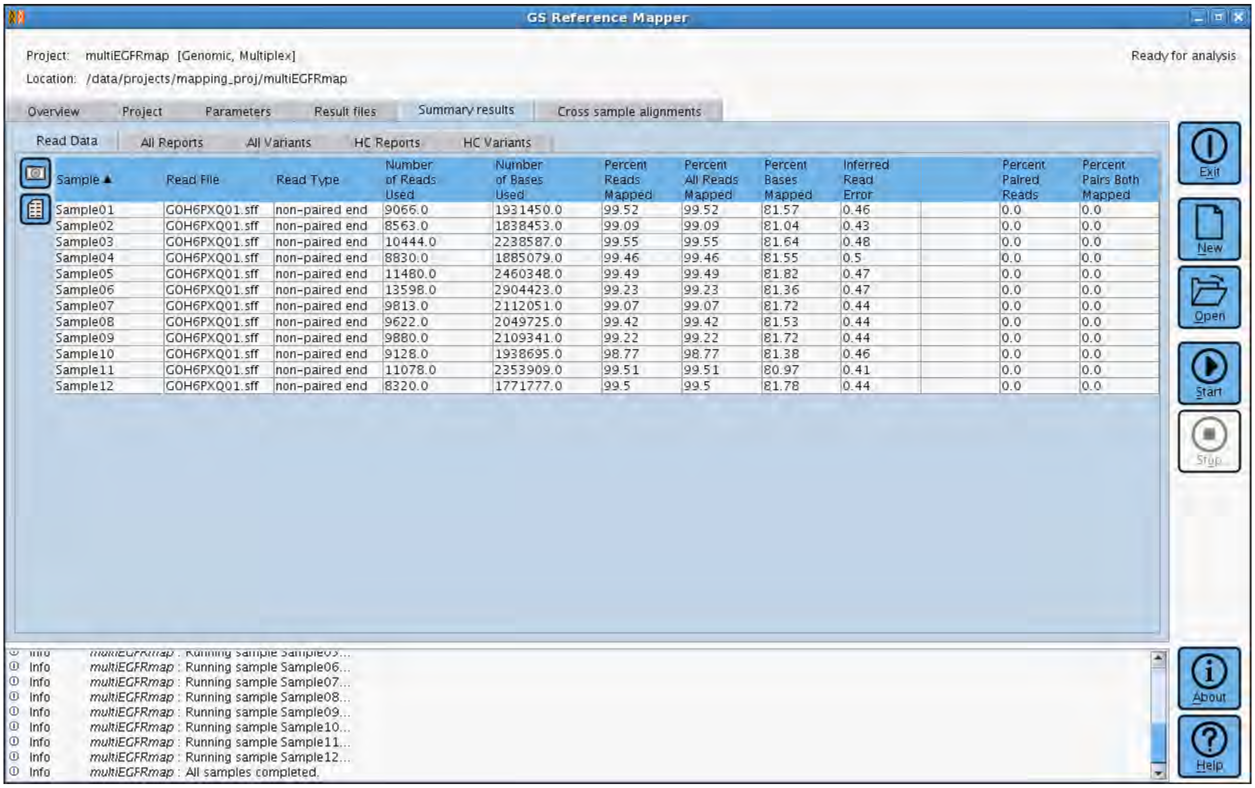
The true power of a multiplex mapping project is the ability to quickly compare mapping results across multiple samples. For example, HC Diffs can be viewed for all combinations of samples and reference position in a single table (HC Reports sub-tab), or variant frequencies can be viewed across all samples for each reference position (HC Variants sub-tab). Moreover, in addition to viewing read alignments for a single sample at multiple reference positions (Project tab hierarchical tree), you can view read alignments at a single reference position across multiple samples (Cross sample alignments sub-tab).

The two Reports sub-tabs and the two Variants sub-tabs are tabular views of the data stored in their equivalent plain text Result Files (see Table 2). They have the advantages of displaying the data in proper columns, sorting by any column of data, and scrolling in both dimensions. Moreover, the two Variants sub-tabs can be “flipped” (swapping rows and columns), which increases the flexibility in how the data are viewed.

The criteria used to report variants and high confidence (HC) variants are the same as with an ordinary mapping project, as described in Sections 2.18.1.13 and 2.18.1.14. In addition, multiplex projects have the ability to filter sample variant frequency reports by minimum and maximum thresholds, which are specified on the Parameters Output sub-tab. These thresholds hide variant frequency values outside of the allowed range in the two multiplex sample variant frequency result files found under the Result files tab and the related All Variants and HC Variants sub-tabs found under the Summary results tab. However, they do not remove empty columns or rows from these tables.

2.15.8.1 Summary Results – Read Data Sub-Tab

The Read Data sub-tab displays read statistics for each combination of Sample and read file (see Figure 94). This is a good place to start after a computation has finished.



Project: multiEGFRmap [Genomic, Multiplex]
Location: /data/projects/mapping_proj/multiEGFRmap

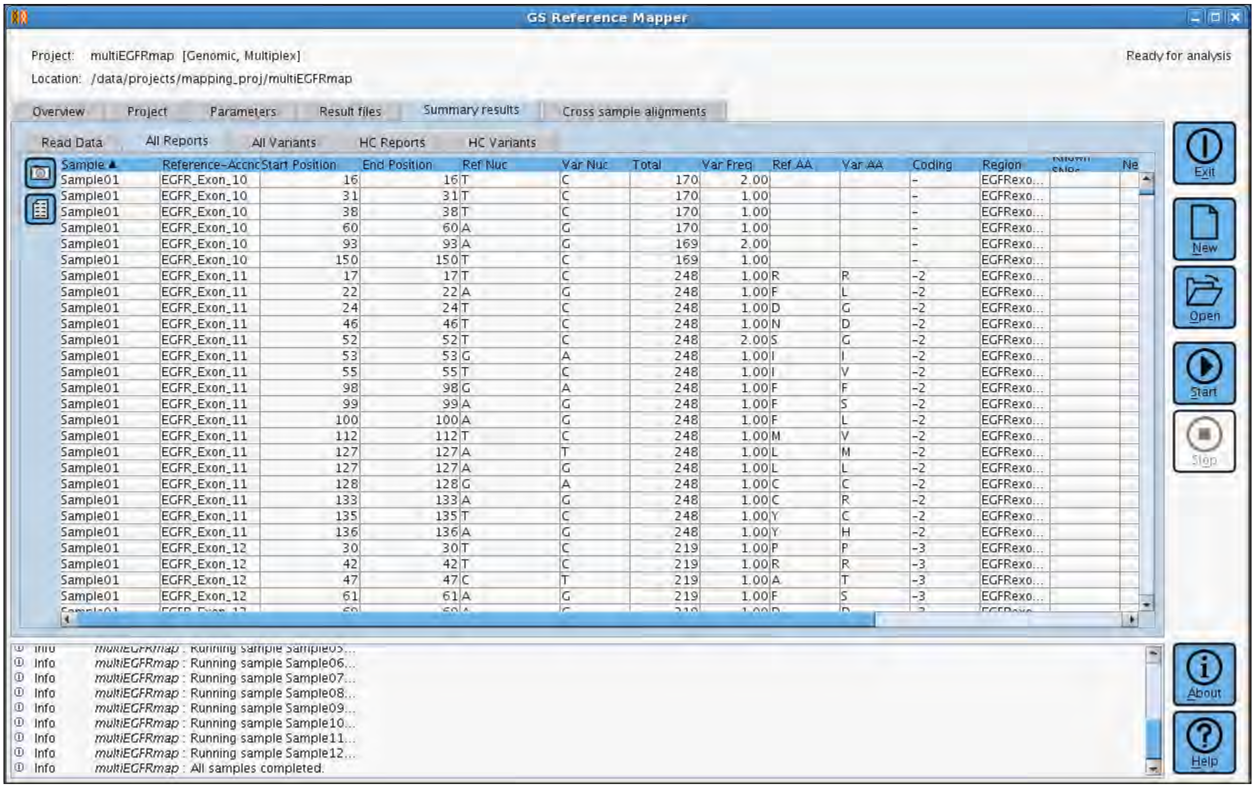
Sample ▲	Read File	Read Type	Number of Reads Used	Number of Bases Used	Percent Reads Mapped	Percent All Reads Mapped	Percent Bases Mapped	Inferred Read Error		Percent Paired Reads	Percent Pairs Both Mapped
Sample01	GOH6PXQ01.sff	non-paired end	9066.0	1931450.0	99.52	99.52	81.57	0.46		0.0	0.0
Sample02	GOH6PXQ01.sff	non-paired end	8563.0	1838453.0	99.09	99.09	81.04	0.43		0.0	0.0
Sample03	GOH6PXQ01.sff	non-paired end	10444.0	2238587.0	99.55	99.55	81.64	0.48		0.0	0.0
Sample04	GOH6PXQ01.sff	non-paired end	8830.0	1885079.0	99.46	99.46	81.55	0.5		0.0	0.0
Sample05	GOH6PXQ01.sff	non-paired end	11480.0	2460348.0	99.49	99.49	81.82	0.47		0.0	0.0
Sample06	GOH6PXQ01.sff	non-paired end	13598.0	2904423.0	99.23	99.23	81.36	0.47		0.0	0.0
Sample07	GOH6PXQ01.sff	non-paired end	9813.0	2112051.0	99.07	99.07	81.72	0.44		0.0	0.0
Sample08	GOH6PXQ01.sff	non-paired end	9622.0	2049725.0	99.42	99.42	81.53	0.44		0.0	0.0
Sample09	GOH6PXQ01.sff	non-paired end	9880.0	2109341.0	99.22	99.22	81.72	0.44		0.0	0.0
Sample10	GOH6PXQ01.sff	non-paired end	9128.0	1938695.0	98.77	98.77	81.38	0.46		0.0	0.0
Sample11	GOH6PXQ01.sff	non-paired end	11078.0	2353909.0	99.51	99.51	80.97	0.41		0.0	0.0
Sample12	GOH6PXQ01.sff	non-paired end	8320.0	1771777.0	99.5	99.5	81.78	0.44		0.0	0.0

```
Info    multiEGFRmap : Running sample Sample05...
Info    multiEGFRmap : Running sample Sample06...
Info    multiEGFRmap : Running sample Sample07...
Info    multiEGFRmap : Running sample Sample08...
Info    multiEGFRmap : Running sample Sample09...
Info    multiEGFRmap : Running sample Sample10...
Info    multiEGFRmap : Running sample Sample11...
Info    multiEGFRmap : Running sample Sample12...
Info    multiEGFRmap : All samples completed.
```

Figure 94: Multiplex mapping project Read Data sub-tab.

2.15.8.2 Summary Results – All Reports Sub-Tab

The All Reports sub-tab displays a list that combines all variants found in any of the AllDiffs files from the individual sample projects (see Figure 95). Each row is a combination of a sample name and a reference position where a difference was found between that sample and the reference. The table can be sorted by samples, reference position, start position, or any other column to aid in finding variants of interest. Select a single sample-variant of interest, and right-click>Show Sample Alignment to view the multiple read alignments at that reference position in the Project tab hierarchical tree (see Figure 98). The Positions of Interest drop down is populated with a list of all variants identified for that sample. Use the left and right arrow buttons to scroll through the variants one at a time.



Project: multiEGFRmap [Genomic, Multiplex]

Location: /data/projects/mapping_proj/multiEGFRmap

Ready for analysis

Sample ▲	Reference-Accnc	Start Position	End Position	Ref Nuc	Var Nuc	Total	Var Freq	Ref AA	Var AA	Coding	Region	Known SNPs	Ne
Sample01	EGFR_Exon_10	16	16	T	C	170	2.00			-	EGFRexo...		
Sample01	EGFR_Exon_10	31	31	T	C	170	1.00			-	EGFRexo...		
Sample01	EGFR_Exon_10	38	38	T	C	170	1.00			-	EGFRexo...		
Sample01	EGFR_Exon_10	60	60	A	G	170	1.00			-	EGFRexo...		
Sample01	EGFR_Exon_10	93	93	A	G	169	2.00			-	EGFRexo...		
Sample01	EGFR_Exon_10	150	150	T	C	169	1.00			-	EGFRexo...		
Sample01	EGFR_Exon_11	17	17	T	C	248	1.00	R	R	-2	EGFRexo...		
Sample01	EGFR_Exon_11	22	22	A	G	248	1.00	F	L	-2	EGFRexo...		
Sample01	EGFR_Exon_11	24	24	T	C	248	1.00	D	G	-2	EGFRexo...		
Sample01	EGFR_Exon_11	46	46	T	C	248	1.00	N	D	-2	EGFRexo...		
Sample01	EGFR_Exon_11	52	52	T	C	248	2.00	S	G	-2	EGFRexo...		
Sample01	EGFR_Exon_11	53	53	G	A	248	1.00	I	I	-2	EGFRexo...		
Sample01	EGFR_Exon_11	55	55	T	C	248	1.00	I	V	-2	EGFRexo...		
Sample01	EGFR_Exon_11	98	98	G	A	248	1.00	F	F	-2	EGFRexo...		
Sample01	EGFR_Exon_11	99	99	A	G	248	1.00	F	S	-2	EGFRexo...		
Sample01	EGFR_Exon_11	100	100	A	G	248	1.00	F	L	-2	EGFRexo...		
Sample01	EGFR_Exon_11	112	112	T	C	248	1.00	M	V	-2	EGFRexo...		
Sample01	EGFR_Exon_11	127	127	A	T	248	1.00	L	M	-2	EGFRexo...		
Sample01	EGFR_Exon_11	127	127	A	G	248	1.00	L	L	-2	EGFRexo...		
Sample01	EGFR_Exon_11	128	128	G	A	248	1.00	C	C	-2	EGFRexo...		
Sample01	EGFR_Exon_11	133	133	A	G	248	1.00	C	R	-2	EGFRexo...		
Sample01	EGFR_Exon_11	135	135	T	C	248	1.00	Y	C	-2	EGFRexo...		
Sample01	EGFR_Exon_11	136	136	A	G	248	1.00	Y	H	-2	EGFRexo...		
Sample01	EGFR_Exon_12	30	30	T	C	219	1.00	P	P	-3	EGFRexo...		
Sample01	EGFR_Exon_12	42	42	T	C	219	1.00	R	R	-3	EGFRexo...		
Sample01	EGFR_Exon_12	47	47	C	T	219	1.00	A	T	-3	EGFRexo...		
Sample01	EGFR_Exon_12	61	61	A	G	219	1.00	F	S	-3	EGFRexo...		

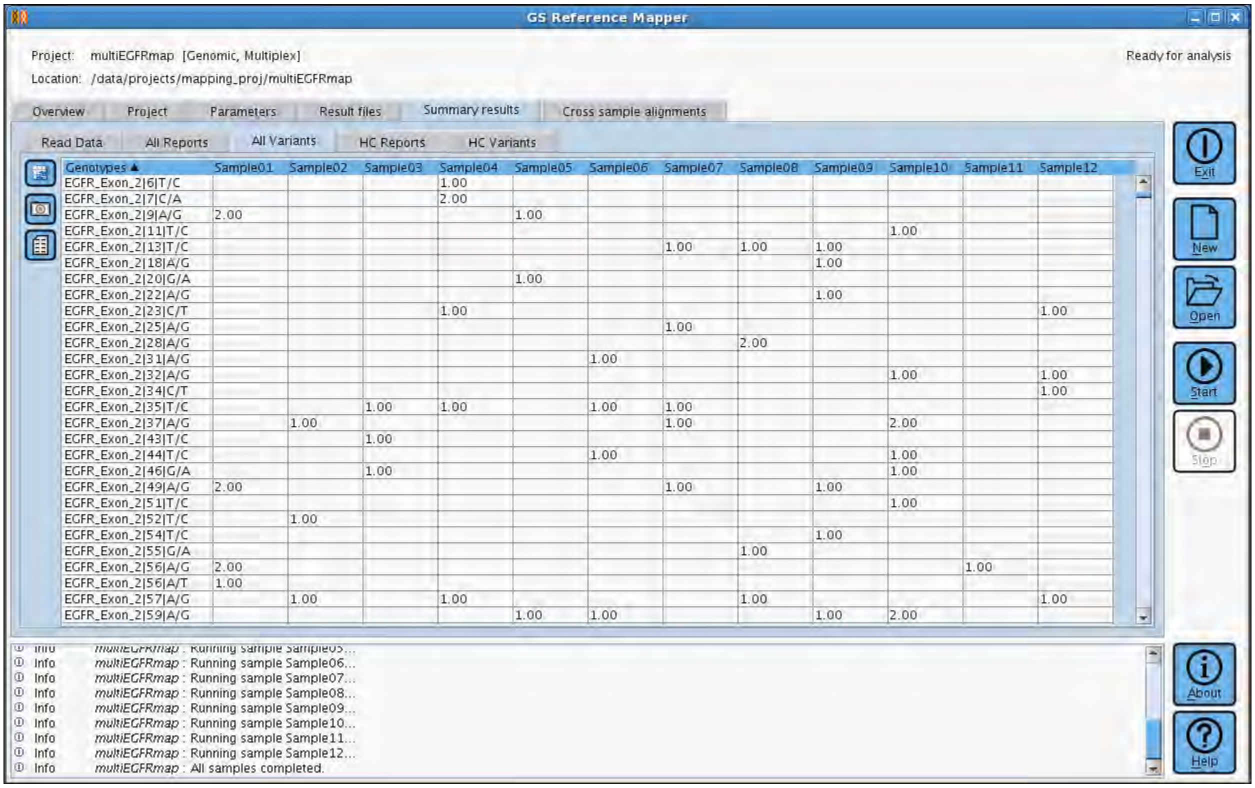
```
ⓘ Info    multiEGFRmap : Running sample Sample05...
ⓘ Info    multiEGFRmap : Running sample Sample06...
ⓘ Info    multiEGFRmap : Running sample Sample07...
ⓘ Info    multiEGFRmap : Running sample Sample08...
ⓘ Info    multiEGFRmap : Running sample Sample09...
ⓘ Info    multiEGFRmap : Running sample Sample10...
ⓘ Info    multiEGFRmap : Running sample Sample11...
ⓘ Info    multiEGFRmap : Running sample Sample12...
ⓘ Info    multiEGFRmap : All samples completed.
```

Figure 95: Multiplex mapping project All Reports sub-tab.

2.15.8.3 Summary Results – All Variants Sub-Tab

The All Variants sub-tab displays the variant frequency for each observed variant by sample, with each sample in a column (see Figure 96). Variants are sorted by reference position, which makes them easy to locate. Click the Flip table button ⇅ to pivot the table, displaying samples in rows and variants in columns.

The sample variant frequency thresholds are useful for honing in on variants of interest based on their frequency of occurrence. By default, this filtration is turned off by setting the minimum and maximum values to 0% and 100%, respectively. Sample-variant combinations with variant frequencies outside of the defined range will still display a cell, but without a variant frequency value. To change the threshold values used in this table, set the Minimum and/or Maximum sample variant frequency percentage on the Output sub-tab (see Section 2.15.4) and then Re-run summary report on the Computation sub-tab (see Section 2.15.5).



Project: multiEGFRmap [Genomic, Multiplex]
Location: /data/projects/mapping_proj/multiEGFRmap

Ready for analysis

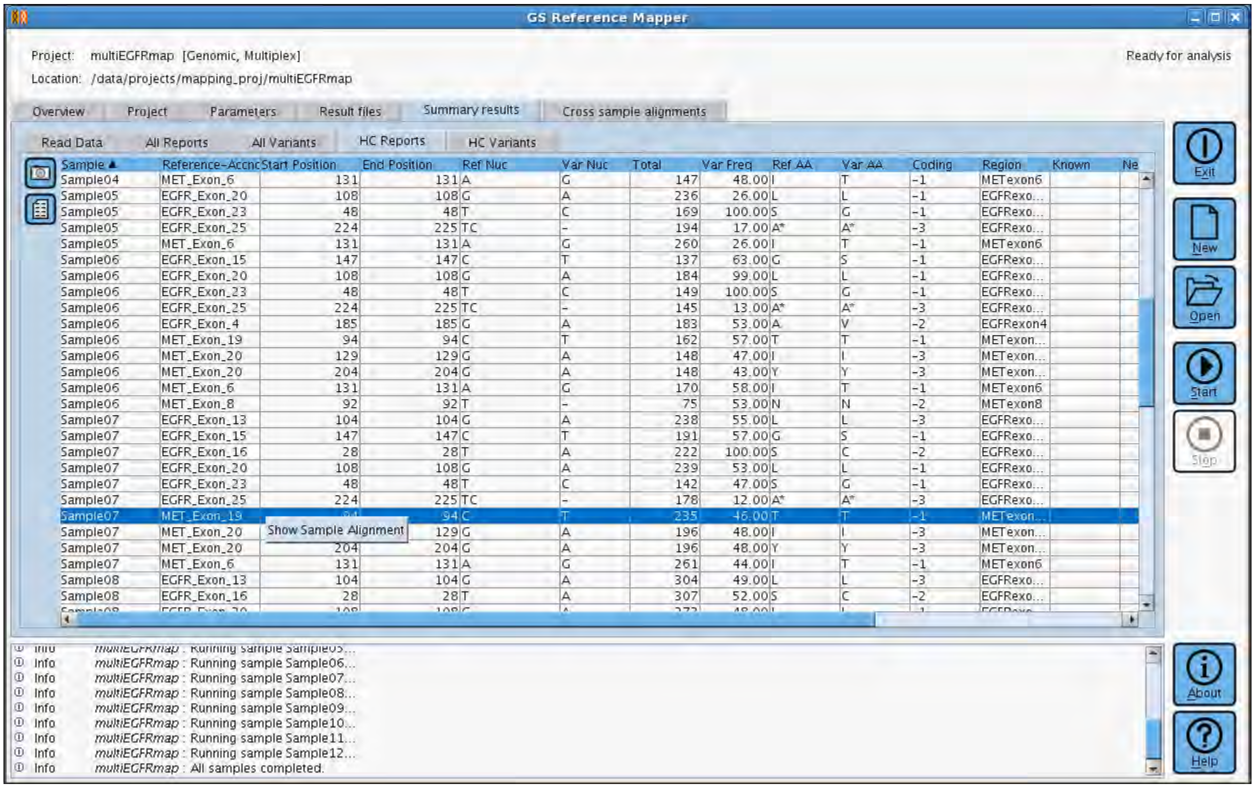
Genotypes ▲	Sample01	Sample02	Sample03	Sample04	Sample05	Sample06	Sample07	Sample08	Sample09	Sample10	Sample11	Sample12
EGFR_Exon_2|6|T/C				1.00								
EGFR_Exon_2|7|C/A				2.00								
EGFR_Exon_2|9|A/G	2.00				1.00							
EGFR_Exon_2|11|T/C										1.00		
EGFR_Exon_2|13|T/C							1.00	1.00	1.00			
EGFR_Exon_2|18|A/G									1.00			
EGFR_Exon_2|20|G/A					1.00							
EGFR_Exon_2|22|A/G									1.00			
EGFR_Exon_2|23|C/T				1.00								1.00
EGFR_Exon_2|25|A/G							1.00					
EGFR_Exon_2|28|A/G								2.00				
EGFR_Exon_2|31|A/G						1.00						
EGFR_Exon_2|32|A/G										1.00		1.00
EGFR_Exon_2|34|C/T												1.00
EGFR_Exon_2|35|T/C			1.00	1.00		1.00	1.00					
EGFR_Exon_2|37|A/G		1.00					1.00			2.00		
EGFR_Exon_2|43|T/C			1.00									
EGFR_Exon_2|44|T/C						1.00				1.00		
EGFR_Exon_2|46|G/A			1.00							1.00		
EGFR_Exon_2|49|A/G	2.00						1.00		1.00			
EGFR_Exon_2|51|T/C										1.00		
EGFR_Exon_2|52|T/C		1.00										
EGFR_Exon_2|54|T/C									1.00			
EGFR_Exon_2|55|G/A								1.00				
EGFR_Exon_2|56|A/G	2.00										1.00	
EGFR_Exon_2|56|A/T	1.00											
EGFR_Exon_2|57|A/G		1.00		1.00				1.00				1.00
EGFR_Exon_2|59|A/G					1.00	1.00			1.00	2.00		

```
ⓘ Info    multiEGFRmap : Running sample Sample05...
ⓘ Info    multiEGFRmap : Running sample Sample06...
ⓘ Info    multiEGFRmap : Running sample Sample07...
ⓘ Info    multiEGFRmap : Running sample Sample08...
ⓘ Info    multiEGFRmap : Running sample Sample09...
ⓘ Info    multiEGFRmap : Running sample Sample10...
ⓘ Info    multiEGFRmap : Running sample Sample11...
ⓘ Info    multiEGFRmap : Running sample Sample12...
ⓘ Info    multiEGFRmap : All samples completed.
```

Figure 96: Multiplex mapping project All Variants sub-tab. The Minimum sample variant frequency percentage was left at the default value of 0% on the Output sub-tab, so all reported variants display a value rather than an empty cell (the empty cells under this circumstance represent samples that have no variants reported at the reference position).

2.15.8.4 Summary Results – HC Reports Sub-Tab

The HC Reports sub-tab displays the same information as the All Reports sub-tab, but for only those reference positions with high confidence variants (see Figure 97).



Project: multiEGFRmap [Genomic, Multiplex]
Location: /data/projects/mapping_proj/multiEGFRmap
Ready for analysis

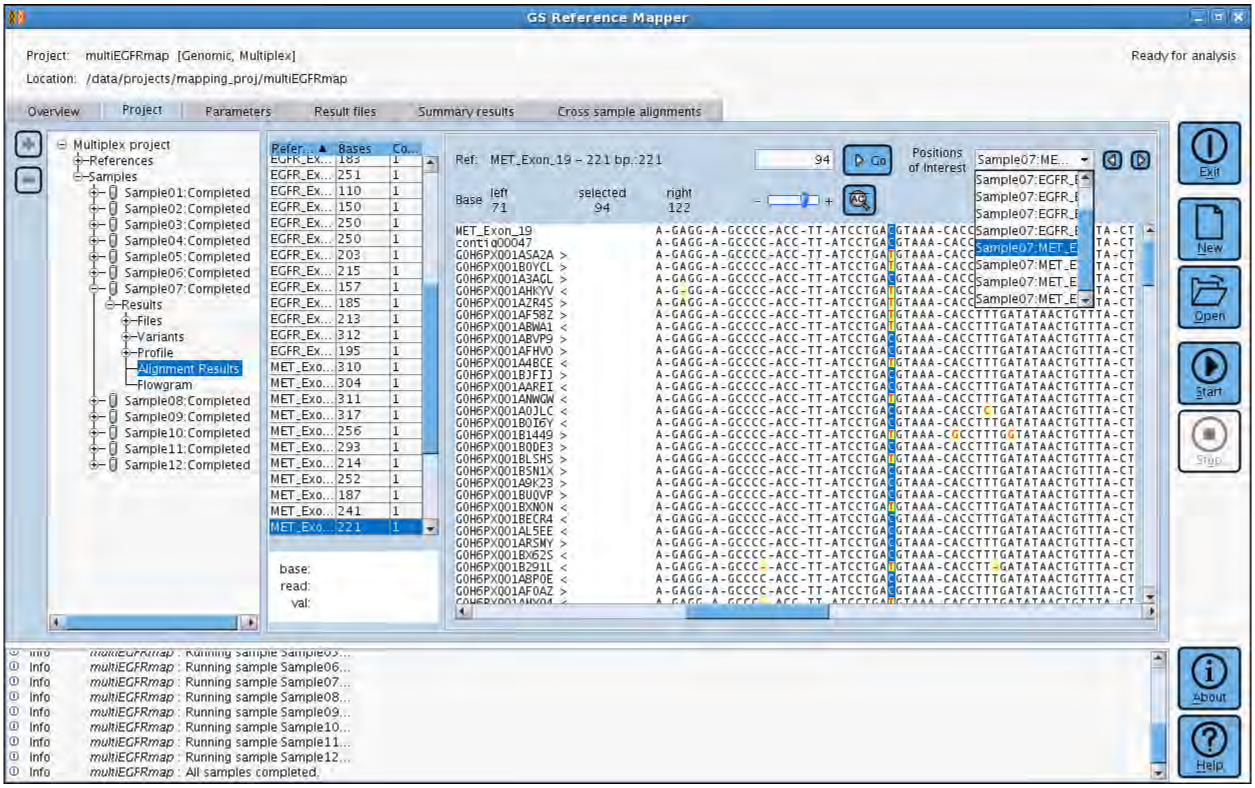
Sample ▲	Reference-Accnc	Start Position	End Position	Ref Nuc	Var Nuc	Total	Var Freq	Ref AA	Var AA	Coding	Region	Known	Ne
Sample04	MET_Exon_6	131	131	A	G	147	48.00	I	T	-1	METexon6		
Sample05	EGFR_Exon_20	108	108	G	A	236	26.00	L	L	-1	EGFRexo...		
Sample05	EGFR_Exon_23	48	48	T	C	169	100.00	S	G	-1	EGFRexo...		
Sample05	EGFR_Exon_25	224	225	TC	-	194	17.00	A*	A*	-3	EGFRexo...		
Sample05	MET_Exon_6	131	131	A	G	260	26.00	I	T	-1	METexon6		
Sample06	EGFR_Exon_15	147	147	C	T	137	63.00	G	S	-1	EGFRexo...		
Sample06	EGFR_Exon_20	108	108	G	A	184	99.00	L	L	-1	EGFRexo...		
Sample06	EGFR_Exon_23	48	48	T	C	149	100.00	S	G	-1	EGFRexo...		
Sample06	EGFR_Exon_25	224	225	TC	-	145	13.00	A*	A*	-3	EGFRexo...		
Sample06	EGFR_Exon_4	185	185	G	A	183	53.00	A	V	-2	EGFRexon4		
Sample06	MET_Exon_19	94	94	C	T	162	57.00	T	T	-1	METexon...		
Sample06	MET_Exon_20	129	129	G	A	148	47.00	I	I	-3	METexon...		
Sample06	MET_Exon_20	204	204	G	A	148	43.00	Y	Y	-3	METexon...		
Sample06	MET_Exon_6	131	131	A	G	170	58.00	I	T	-1	METexon6		
Sample06	MET_Exon_8	92	92	T	-	75	53.00	N	N	-2	METexon8		
Sample07	EGFR_Exon_13	104	104	G	A	238	55.00	L	L	-3	EGFRexo...		
Sample07	EGFR_Exon_15	147	147	C	T	191	57.00	G	S	-1	EGFRexo...		
Sample07	EGFR_Exon_16	28	28	T	A	222	100.00	S	C	-2	EGFRexo...		
Sample07	EGFR_Exon_20	108	108	G	A	239	53.00	L	L	-1	EGFRexo...		
Sample07	EGFR_Exon_23	48	48	T	C	142	47.00	S	G	-1	EGFRexo...		
Sample07	EGFR_Exon_25	224	225	TC	-	178	12.00	A*	A*	-3	EGFRexo...		
Sample07	MET_Exon_19	94	94	C	T	235	46.00	T	T	-1	METexon...		
Sample07	MET_Exon_20		129	G	A	196	48.00	I	I	-3	METexon...		
Sample07	MET_Exon_20	204	204	G	A	196	48.00	Y	Y	-3	METexon...		
Sample07	MET_Exon_6	131	131	A	G	261	44.00	I	T	-1	METexon6		
Sample08	EGFR_Exon_13	104	104	G	A	304	49.00	L	L	-3	EGFRexo...		
Sample08	EGFR_Exon_16	28	28	T	A	307	52.00	S	C	-2	EGFRexo...		

```
Info   multiEGFRmap : Running sample Sample05...
Info   multiEGFRmap : Running sample Sample06...
Info   multiEGFRmap : Running sample Sample07...
Info   multiEGFRmap : Running sample Sample08...
Info   multiEGFRmap : Running sample Sample09...
Info   multiEGFRmap : Running sample Sample10...
Info   multiEGFRmap : Running sample Sample11...
Info   multiEGFRmap : Running sample Sample12...
Info   multiEGFRmap : All samples completed.
```

Figure 97: Multiplex mapping project HC Reports sub-tab.

Select a variant and right-click>Show Sample Alignment to view the multiple read alignments at that reference position in the Project tab hierarchical tree.

The Positions of Interest drop down is populated with a list of all HC Diffs identified for that sample (see Figure 98). Use the left and right arrow buttons to scroll through the variants one at a time.



Project: multiEGFRmap [Genomic, Multiplex]
Location: /data/projects/mapping_proj/multiEGFRmap
Ready for analysis

```
⊟ Multiplex project
  ⊞-References
  ⊟-Samples
    ⊞- Sample01:Completed
    ⊞- Sample02:Completed
    ⊞- Sample03:Completed
    ⊞- Sample04:Completed
    ⊞- Sample05:Completed
    ⊞- Sample06:Completed
    ⊟- Sample07:Completed
       ⊟-Results
         ⊞-Files
         ⊞-Variants
         ⊞-Profile
          -Alignment Results
          -Flowgram
    ⊞- Sample08:Completed
    ⊞- Sample09:Completed
    ⊞- Sample10:Completed
    ⊞- Sample11:Completed
    ⊞- Sample12:Completed
```

Refer... ▲	Bases	Co...
EGFR_Ex...	183	1
EGFR_Ex...	251	1
EGFR_Ex...	110	1
EGFR_Ex...	150	1
EGFR_Ex...	250	1
EGFR_Ex...	250	1
EGFR_Ex...	203	1
EGFR_Ex...	215	1
EGFR_Ex...	157	1
EGFR_Ex...	185	1
EGFR_Ex...	213	1
EGFR_Ex...	312	1
EGFR_Ex...	195	1
MET_Exo...	310	1
MET_Exo...	304	1
MET_Exo...	311	1
MET_Exo...	317	1
MET_Exo...	256	1
MET_Exo...	293	1
MET_Exo...	214	1
MET_Exo...	252	1
MET_Exo...	187	1
MET_Exo...	241	1
MET_Exo...	221	1

base:
read:
val:

Ref: MET_Exon_19 – 221 bp.:221   94

Positions of Interest: Sample07:ME...

- Sample07:EGFR_E
- Sample07:EGFR_E
- Sample07:EGFR_E
- Sample07:EGFR_E
- Sample07:MET_E
- Sample07:MET_E
- Sample07:MET_E
- Sample07:MET_E

Base	left	selected	right
	71	94	122

```
MET_Exon_19          A-GAGG-A-GCCCC-ACC-TT-ATCCTGACGTAAA-CACC              TA-CT
contig00047          A-GAGG-A-GCCCC-ACC-TT-ATCCTGACGTAAA-CACC              TA-CT
G0H6PX001ASA2A >     A-GAGG-A-GCCCC-ACC-TT-ATCCTGATGTAAA-CACC              TA-CT
G0H6PX001B0YCL >     A-GAGG-A-GCCCC-ACC-TT-ATCCTGATGTAAA-CACC              TA-CT
G0H6PX001A3AGL >     A-GAGG-A-GCCCC-ACC-TT-ATCCTGACGTAAA-CACC              TA-CT
G0H6PX001AHKYV <     A-G-GG-A-GCCCC-ACC-TT-ATCCTGATGTAAA-CACC              TA-CT
G0H6PX001AZR4S >     A-GAGG-A-GCCCC-ACC-TT-ATCCTGATGTAAA-CACC              TA-CT
G0H6PX001AF58Z >     A-GAGG-A-GCCCC-ACC-TT-ATCCTGATGTAAA-CACCTTTGATATAACTGTTTA-CT
G0H6PX001ABWA1 <     A-GAGG-A-GCCCC-ACC-TT-ATCCTGATGTAAA-CACCTTTGATATAACTGTTTA-CT
G0H6PX001ABVP9 >     A-GAGG-A-GCCCC-ACC-TT-ATCCTGACGTAAA-CACCTTTGATATAACTGTTTA-CT
G0H6PX001AFHVO >     A-GAGG-A-GCCCC-ACC-TT-ATCCTGACGTAAA-CACCTTTGATATAACTGTTTA-CT
G0H6PX001A4BCE <     A-GAGG-A-GCCCC-ACC-TT-ATCCTGATGTAAA-CACCTTTGATATAACTGTTTA-CT
G0H6PX001BJFIJ >     A-GAGG-A-GCCCC-ACC-TT-ATCCTGACGTAAA-CACCTTTGATATAACTGTTTA-CT
G0H6PX001AAREI <     A-GAGG-A-GCCCC-ACC-TT-ATCCTGACGTAAA-CACCTTTGATATAACTGTTTA-CT
G0H6PX001ANWGW <     A-GAGG-A-GCCCC-ACC-TT-ATCCTGATGTAAA-CACCTTTGATATAACTGTTTA-CT
G0H6PX001A0JLC <     A-GAGG-A-GCCCC-ACC-TT-ATCCTGACGTAAA-CACCTCTGATATAACTGTTTA-CT
G0H6PX001B0I6Y <     A-GAGG-A-GCCCC-ACC-TT-ATCCTGACGTAAA-CACCTTTGATATAACTGTTTA-CT
G0H6PX001B1449 >     A-GAGG-A-GCCCC-ACC-TT-ATCCTGATGTAAA-CGCCTTTGGTATAACTGTTTA-CT
G0H6PX001B0DE3 >     A-GAGG-A-GCCCC-ACC-TT-ATCCTGACGTAAA-CACCTTTGATATAACTGTTTA-CT
G0H6PX001BLSHS >     A-GAGG-A-GCCCC-ACC-TT-ATCCTGATGTAAA-CACCTTTGATATAACTGTTTA-CT
G0H6PX001BSN1X >     A-GAGG-A-GCCCC-ACC-TT-ATCCTGACGTAAA-CACCTTTGATATAACTGTTTA-CT
G0H6PX001A9K23 >     A-GAGG-A-GCCCC-ACC-TT-ATCCTGACGTAAA-CACCTTTGATATAACTGTTTA-CT
G0H6PX001BUQVP >     A-GAGG-A-GCCCC-ACC-TT-ATCCTGACGTAAA-CACCTTTGATATAACTGTTTA-CT
G0H6PX001BXNON <     A-GAGG-A-GCCCC-ACC-TT-ATCCTGATGTAAA-CACCTTTGATATAACTGTTTA-CT
G0H6PX001BECR4 <     A-GAGG-A-GCCCC-ACC-TT-ATCCTGACGTAAA-CACCTTTGATATAACTGTTTA-CT
G0H6PX001AL5EE <     A-GAGG-A-GCCCC-ACC-TT-ATCCTGACGTAAA-CACCTTTGATATAACTGTTTA-CT
G0H6PX001ARSMY >     A-GAGG-A-GCCCC-ACC-TT-ATCCTGACGTAAA-CACCTTTGATATAACTGTTTA-CT
G0H6PX001BX62S <     A-GAGG-A-GCCCC-ACC-TT-ATCCTGACGTAAA-CACCTTTGATATAACTGTTTA-CT
G0H6PX001B291L <     A-GAGG-A-GCCC--ACC-TT-ATCCTGATGTAAA-CACCTT-GATATAACTGTTTA-CT
G0H6PX001A8P0E <     A-GAGG-A-GCCCC-ACC-TT-ATCCTGACGTAAA-CACCTTTGATATAACTGTTTA-CT
G0H6PX001AF0AZ >     A-GAGG-A-GCCCC-ACC-TT-ATCCTGACGTAAA-CACCTTTGATATAACTGTTTA-CT
```

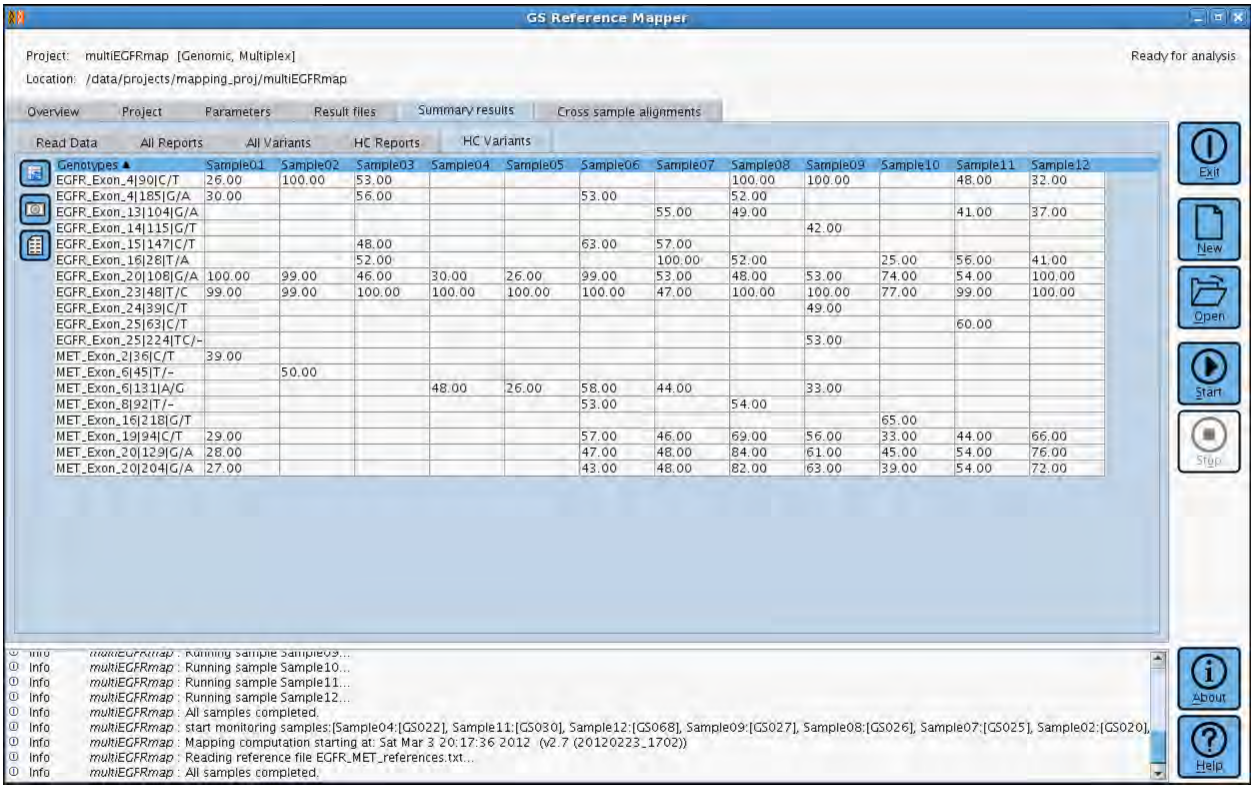
```
ⓘ Info   multiEGFRmap : Running sample Sample05...
ⓘ Info   multiEGFRmap : Running sample Sample06...
ⓘ Info   multiEGFRmap : Running sample Sample07...
ⓘ Info   multiEGFRmap : Running sample Sample08...
ⓘ Info   multiEGFRmap : Running sample Sample09...
ⓘ Info   multiEGFRmap : Running sample Sample10...
ⓘ Info   multiEGFRmap : Running sample Sample11...
ⓘ Info   multiEGFRmap : Running sample Sample12...
ⓘ Info   multiEGFRmap : All samples completed.
```

Figure 98: Multiple read alignments for a specific sample across HC Diffs.

2.15.8.5 Summary Results – HC Variants Sub-Tab

The HC Variants sub-tab displays the same information as the All Variants sub-tab, but for only those reference positions with high confidence differences (see Figure 99).

The sample variant frequency thresholds are useful for honing in on variants of interest based on their frequency of occurrence. By default, this filtration is turned off by setting the minimum and maximum values to 0% and 100%, respectively. Sample-variant combinations with variant frequencies outside of the defined range will still display a cell, but without a variant frequency value. To change the threshold values used in this table, set the Minimum and/or Maximum sample variant frequency percentage on the Output sub-tab (see Section 2.15.4) and then Re-run summary report on the Computation sub-tab (see Section 2.15.5).



Project: multiEGFRmap [Genomic, Multiplex]
Location: /data/projects/mapping_proj/multiEGFRmap

Genotypes ▲	Sample01	Sample02	Sample03	Sample04	Sample05	Sample06	Sample07	Sample08	Sample09	Sample10	Sample11	Sample12
EGFR_Exon_4|90|C/T	26.00	100.00	53.00					100.00	100.00		48.00	32.00
EGFR_Exon_4|185|G/A	30.00		56.00			53.00		52.00				
EGFR_Exon_13|104|G/A							55.00	49.00			41.00	37.00
EGFR_Exon_14|115|G/T									42.00			
EGFR_Exon_15|147|C/T			48.00			63.00	57.00					
EGFR_Exon_16|28|T/A			52.00				100.00	52.00		25.00	56.00	41.00
EGFR_Exon_20|108|G/A	100.00	99.00	46.00	30.00	26.00	99.00	53.00	48.00	53.00	74.00	54.00	100.00
EGFR_Exon_23|48|T/C	99.00	99.00	100.00	100.00	100.00	100.00	47.00	100.00	100.00	77.00	99.00	100.00
EGFR_Exon_24|39|C/T									49.00			
EGFR_Exon_25|63|C/T											60.00	
EGFR_Exon_25|224|TC/-									53.00			
MET_Exon_2|36|C/T	39.00											
MET_Exon_6|45|T/-		50.00										
MET_Exon_6|131|A/G				48.00	26.00	58.00	44.00		33.00			
MET_Exon_8|92|T/-						53.00		54.00				
MET_Exon_16|218|G/T										65.00		
MET_Exon_19|94|C/T	29.00					57.00	46.00	69.00	56.00	33.00	44.00	66.00
MET_Exon_20|129|G/A	28.00					47.00	48.00	84.00	61.00	45.00	54.00	76.00
MET_Exon_20|204|G/A	27.00					43.00	48.00	82.00	63.00	39.00	54.00	72.00

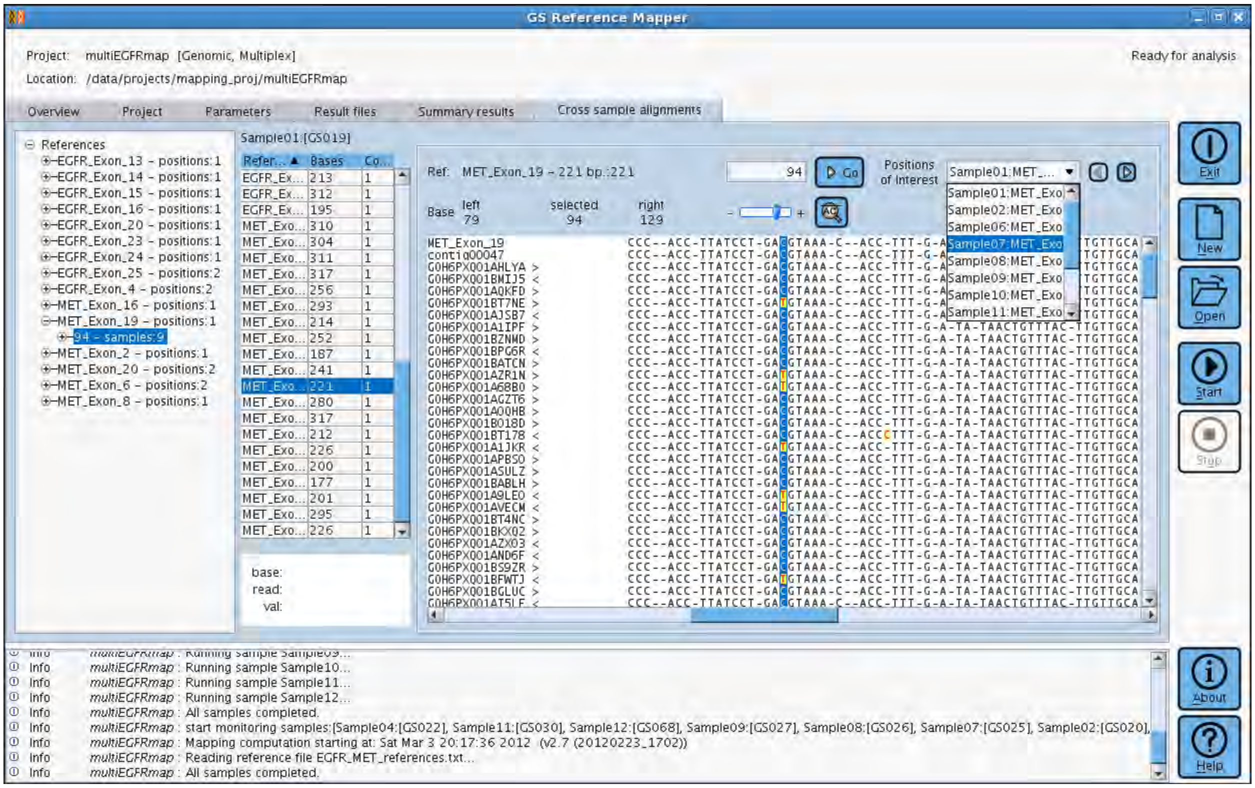
```
ⓘ Info    multiEGFRmap : Running sample Sample09...
ⓘ Info    multiEGFRmap : Running sample Sample10...
ⓘ Info    multiEGFRmap : Running sample Sample11...
ⓘ Info    multiEGFRmap : Running sample Sample12...
ⓘ Info    multiEGFRmap : All samples completed.
ⓘ Info    multiEGFRmap : start monitoring samples:[Sample04:[GS022], Sample11:[GS030], Sample12:[GS068], Sample09:[GS027], Sample08:[GS026], Sample07:[GS025], Sample02:[GS020],
ⓘ Info    multiEGFRmap : Mapping computation starting at: Sat Mar 3 20:17:36 2012  (v2.7 (20120223_1702))
ⓘ Info    multiEGFRmap : Reading reference file EGFR_MET_references.txt...
ⓘ Info    multiEGFRmap : All samples completed.
```

Figure 99: Multiplex mapping project HC Variants sub-tab. Only variants with a variant frequency exceeding the 25% value set for the Minimum sample variant frequency percentage on the Output sub-tab are displayed.

2.15.9 Cross Sample Alignments Tab

The Cross sample alignments tab displays multiple read alignments at a given reference position, one sample at a time (see Figure 100). A hierarchical tree is displayed in the left panel of the screen that lists each HC Diff. Click the plus sign to expand the tree to display a count of samples that have an HC Diff at that position (a second click on the next plus sign expands the tree to show the individual sample names).

Select the level of the tree that displays the sample count in order to display read alignments at that reference position, with a list of sample-reference position combinations listed in the Positions of interest drop down. Use the left and right arrow buttons to the right of the Positions of interest drop down to move through the samples one at a time.



Project: multiEGFRmap [Genomic, Multiplex]
Location: /data/projects/mapping_proj/multiEGFRmap

Ready for analysis

References
- EGFR_Exon_13 – positions:1
- EGFR_Exon_14 – positions:1
- EGFR_Exon_15 – positions:1
- EGFR_Exon_16 – positions:1
- EGFR_Exon_20 – positions:1
- EGFR_Exon_23 – positions:1
- EGFR_Exon_24 – positions:1
- EGFR_Exon_25 – positions:2
- EGFR_Exon_4 – positions:2
- MET_Exon_16 – positions:1
- MET_Exon_19 – positions:1
  - 94 – samples:9
- MET_Exon_2 – positions:1
- MET_Exon_20 – positions:2
- MET_Exon_6 – positions:2
- MET_Exon_8 – positions:1

Sample01:[GS019]

Refer... ▲	Bases	Co...
EGFR_Ex...	213	1
EGFR_Ex...	312	1
EGFR_Ex...	195	1
MET_Exo...	310	1
MET_Exo...	304	1
MET_Exo...	311	1
MET_Exo...	317	1
MET_Exo...	256	1
MET_Exo...	293	1
MET_Exo...	214	1
MET_Exo...	252	1
MET_Exo...	187	1
MET_Exo...	241	1
MET_Exo...	221	1
MET_Exo...	280	1
MET_Exo...	317	1
MET_Exo...	212	1
MET_Exo...	226	1
MET_Exo...	200	1
MET_Exo...	177	1
MET_Exo...	201	1
MET_Exo...	295	1
MET_Exo...	226	1

base:
read:
val:

Ref: MET_Exon_19 – 221 bp.:221

94

Positions of Interest: Sample01:MET_...

- Sample01:MET_Exo
- Sample02:MET_Exo
- Sample06:MET_Exo
- Sample07:MET_Exo
- Sample08:MET_Exo
- Sample09:MET_Exo
- Sample10:MET_Exo
- Sample11:MET_Exo

Base left 79 selected 94 right 129

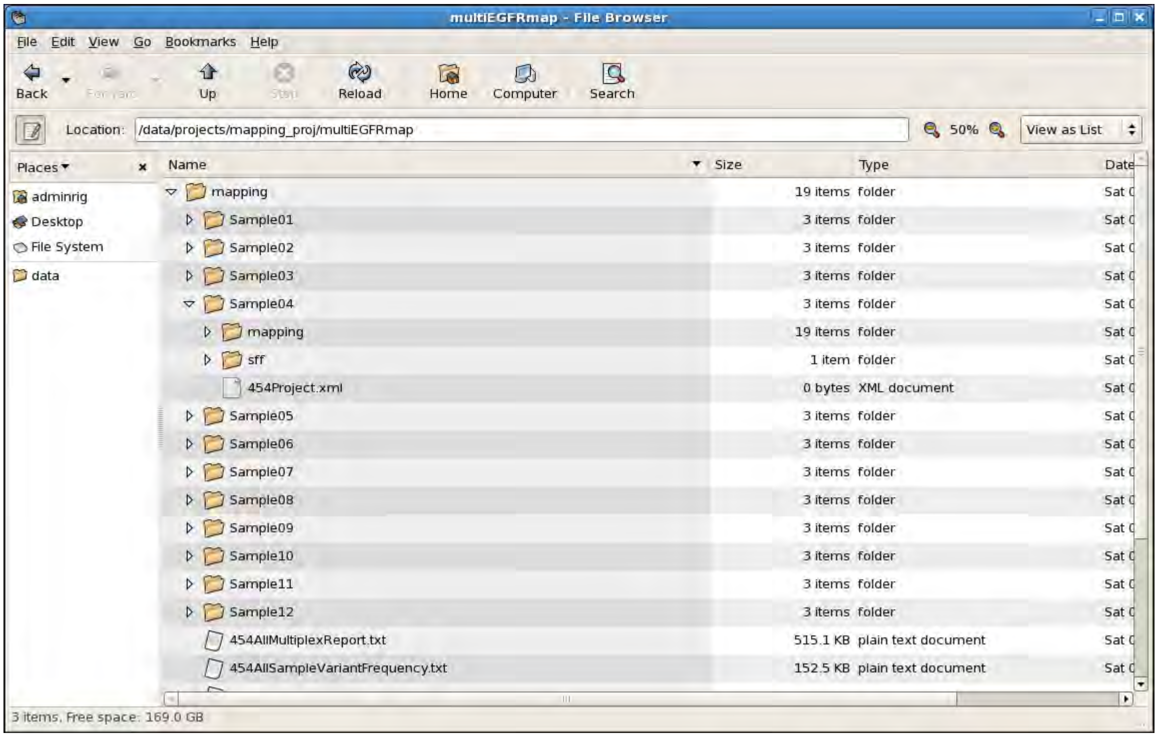
```
MET_Exon_19          CCC--ACC-TTATCCT-GACGTAAA-C--ACC-TTT-G-A               TGTTGCA
contig00047          CCC--ACC-TTATCCT-GACGTAAA-C--ACC-TTT-G-A               TGTTGCA
G0H6PX001AHLYA >     CCC--ACC-TTATCCT-GACGTAAA-C--ACC-TTT-G-A               TGTTGCA
G0H6PX001BMIJ5 <     CCC--ACC-TTATCCT-GACGTAAA-C--ACC-TTT-G-A               TGTTGCA
G0H6PX001AQKFD >     CCC--ACC-TTATCCT-GACGTAAA-C--ACC-TTT-G-A               TGTTGCA
G0H6PX001BT7NE >     CCC--ACC-TTATCCT-GATGTAAA-C--ACC-TTT-G-A               TGTTGCA
G0H6PX001AJSB7 <     CCC--ACC-TTATCCT-GACGTAAA-C--ACC-TTT-G-A               TGTTGCA
G0H6PX001A1IPF >     CCC--ACC-TTATCCT-GACGTAAA-C--ACC-TTT-G-A-TA-TAACTGTTTAC-TTGTTGCA
G0H6PX001BZNMD >     CCC--ACC-TTATCCT-GACGTAAA-C--ACC-TTT-G-A-TA-TAACTGTTTAC-TTGTTGCA
G0H6PX001BPG6R <     CCC--ACC-TTATCCT-GACGTAAA-C--ACC-TTT-G-A-TA-TAACTGTTTAC-TTGTTGCA
G0H6PX001BATCN >     CCC--ACC-TTATCCT-GACGTAAA-C--ACC-TTT-G-A-TA-TAACTGTTTAC-TTGTTGCA
G0H6PX001AZR1N >     CCC--ACC-TTATCCT-GATGTAAA-C--ACC-TTT-G-A-TA-TAACTGTTTAC-TTGTTGCA
G0H6PX001A68B0 >     CCC--ACC-TTATCCT-GATGTAAA-C--ACC-TTT-G-A-TA-TAACTGTTTAC-TTGTTGCA
G0H6PX001AGZT6 >     CCC--ACC-TTATCCT-GACGTAAA-C--ACC-TTT-G-A-TA-TAACTGTTTAC-TTGTTGCA
G0H6PX001A0QHB >     CCC--ACC-TTATCCT-GACGTAAA-C--ACC-TTT-G-A-TA-TAACTGTTTAC-TTGTTGCA
G0H6PX001B018D >     CCC--ACC-TTATCCT-GACGTAAA-C--ACC-TTT-G-A-TA-TAACTGTTTAC-TTGTTGCA
G0H6PX001BT178 <     CCC--ACC-TTATCCT-GACGTAAA-C--ACCCTTT-G-A-TA-TAACTGTTTAC-TTGTTGCA
G0H6PX001A1JKR <     CCC--ACC-TTATCCT-GATGTAAA-C--ACC-TTT-G-A-TA-TAACTGTTTAC-TTGTTGCA
G0H6PX001APBS0 >     CCC--ACC-TTATCCT-GACGTAAA-C--ACC-TTT-G-A-TA-TAACTGTTTAC-TTGTTGCA
G0H6PX001ASULZ >     CCC--ACC-TTATCCT-GACGTAAA-C--ACC-TTT-G-A-TA-TAACTGTTTAC-TTGTTGCA
G0H6PX001BABLH >     CCC--ACC-TTATCCT-GACGTAAA-C--ACC-TTT-G-A-TA-TAACTGTTTAC-TTGTTGCA
G0H6PX001A9LE0 <     CCC--ACC-TTATCCT-GATGTAAA-C--ACC-TTT-G-A-TA-TAACTGTTTAC-TTGTTGCA
G0H6PX001AVECM <     CCC--ACC-TTATCCT-GATGTAAA-C--ACC-TTT-G-A-TA-TAACTGTTTAC-TTGTTGCA
G0H6PX001BT4NC >     CCC--ACC-TTATCCT-GACGTAAA-C--ACC-TTT-G-A-TA-TAACTGTTTAC-TTGTTGCA
G0H6PX001BKX02 >     CCC--ACC-TTATCCT-GACGTAAA-C--ACC-TTT-G-A-TA-TAACTGTTTAC-TTGTTGCA
G0H6PX001AZX03 <     CCC--ACC-TTATCCT-GACGTAAA-C--ACC-TTT-G-A-TA-TAACTGTTTAC-TTGTTGCA
G0H6PX001AND6F <     CCC--ACC-TTATCCT-GACGTAAA-C--ACC-TTT-G-A-TA-TAACTGTTTAC-TTGTTGCA
G0H6PX001BS9ZR >     CCC--ACC-TTATCCT-GACGTAAA-C--ACC-TTT-G-A-TA-TAACTGTTTAC-TTGTTGCA
G0H6PX001BFWTJ <     CCC--ACC-TTATCCT-GATGTAAA-C--ACC-TTT-G-A-TA-TAACTGTTTAC-TTGTTGCA
G0H6PX001BGLUC >     CCC--ACC-TTATCCT-GACGTAAA-C--ACC-TTT-G-A-TA-TAACTGTTTAC-TTGTTGCA
G0H6PX001AT5LF <     CCC--ACC-TTATCCT-GACGTAAA-C--ACC-TTT-G-A-TA-TAACTGTTTAC-TTGTTGCA
```

```
Info    multiEGFRmap : Running sample Sample09...
Info    multiEGFRmap : Running sample Sample10...
Info    multiEGFRmap : Running sample Sample11...
Info    multiEGFRmap : Running sample Sample12...
Info    multiEGFRmap : All samples completed.
Info    multiEGFRmap : start monitoring samples:[Sample04:[GS022], Sample11:[GS030], Sample12:[GS068], Sample09:[GS027], Sample08:[GS026], Sample07:[GS025], Sample02:[GS020],
Info    multiEGFRmap : Mapping computation starting at: Sat Mar 3 20:17:36 2012  (v2.7 (20120223_1702))
Info    multiEGFRmap : Reading reference file EGFR_MET_references.txt...
Info    multiEGFRmap : All samples completed.
```

Figure 100: Multiplex mapping project Cross sample alignments tab showing multiple read alignments for a specific HC Diff across multiple samples.

2.15.10 Open Individual Sample Projects

A multiplex mapping project actually generates a series of ordinary mapping projects; one for each sample. These sample projects are located inside of the "mapping" folder of the multiplex master project (see Figure 101). Individual sample projects can be opened in order to view mapping results in the standard way, to add read files filtered by multiple MIDs, to re-compute individual projects with individualized settings, *etc.*



Location: /data/projects/mapping_proj/multiEGFRmap

Name	Size	Type	Date
mapping	19 items	folder	Sat
Sample01	3 items	folder	Sat
Sample02	3 items	folder	Sat
Sample03	3 items	folder	Sat
Sample04	3 items	folder	Sat
mapping	19 items	folder	Sat
sff	1 item	folder	Sat
454Project.xml	0 bytes	XML document	Sat
Sample05	3 items	folder	Sat
Sample06	3 items	folder	Sat
Sample07	3 items	folder	Sat
Sample08	3 items	folder	Sat
Sample09	3 items	folder	Sat
Sample10	3 items	folder	Sat
Sample11	3 items	folder	Sat
Sample12	3 items	folder	Sat
454AllMultiplexReport.txt	515.1 KB	plain text document	Sat
454AllSampleVariantFrequency.txt	152.5 KB	plain text document	Sat

3 items, Free space: 169.0 GB

Figure 101: Multiplex mapping project directory, showing associated sample projects.

- Do not open a sample project at the same time that its multiplex master project is open.
- Modifying individual sample projects may lead to inconsistencies in the multiplex master project.

2.16 GS Reference Mapper Command Line Interface

The GS Reference Mapper CLI command for one-step mapping of reads can be launched using the following command:

```
runMapping [options] refdesc [filedesc]
```

For incremental mapping *via* the CLI, a project directory structure is created to organize the files of the incremental build of the project data. The following commands are used for incremental mapping projects:

```
newMapping projDir

addRun [options: -p, -lib, -mcf] [projDir] filedesc

removeRun [projDir] filedesc

setRef [options: -gref, -cref, -random] [projDir] refdesc

runProject [options] [projDir]
```

For all of these commands;

- The arguments in brackets [ ] are optional.
- 'options' refers to the specific command options available.
- 'filedesc' is one of the following:
  - an sfffilename
  - a [regionlist:]datadir
  - a readfastafile
- any of the above prepended by an [MIDList@] specification where each "MIDList" is a multiplexing information string used to filter the set of file reads to be used in the mapping (If MIDs were used in the generation of the read data file, then an MIDList string must be specified in order for the mapper to properly handle the file's reads.) For details about using MIDs, see section 4.5.
- 'refdesc' is a path/name to a reference sequence file in FASTA format.
- 'projDir' is the path of the data analysis project directory.

Some of the command line options for Mapping are mutually exclusive, for obvious functional reasons. See section 4.3 for a list of these options.

2.16.1 Working with Project Folders and Data Files

Since the mapping computation is often performed on a pool of sequencing runs (or Read Data files) rather than on any single run, the result files it generates are not deposited in a run folder. Two general cases exist.

- If the mapping is performed using the "one-step" command runMapping, a folder with a 'P_' prefix (for 'P'ost-Run Analysis) is created in the user's current working directory on the datarig at the time the application is launched, or written to a directory specified by the user on the command line (or its GUI equivalent), to contain these files. The name structure for this folder is as follows:

  P_yyyy_mm_dd_hh_min_sec_runMapping

- For "incremental", or "project-based" mapping, using the GUI application or using the newMapping and related commands, the output is placed in a "project" folder. A user can specify any name for a project folder; it will be recognized as a project folder by virtue of the "454Project.xml" file that will be automatically created within it. If a directory name is not specified using the newMapping command line, the software will use the same default name as the runMapping command (as above).

The incremental mapping folder contains additional folders and files that mark the folder as a project folder and that store configuration information and internal data for the GS Reference Mapper application. A project folder is comprised of two sub-folders: a 'mapping' sub-folder which contains the project state and output files; and an 'sff' sub-folder containing the copies and/or symbolic links for the SFF files used as input to the project. The 454Project.xml file identifies the folder as a 454 project folder.

- The "-o" option can be specified on the command line to change the directory where the output files should be written. If the specified directory exists, the output files will be written to that directory. If the specified directory does not exist, the program will create it, if possible.

When using the –o option with the runMapping or newMapping command, the mapper will not overwrite the contents of the specified project directory if any exist. To force an overwrite of an existing directory, use the –force option with the runMapping or newMapping commands.

If a project or any of the files it uses need to be moved from one location to another, the project's configuration file (454MappingProject.xml) must be modified to reflect the new location. The changeRun and changeRef commands facilitate this process.

External Files

- GS Read Data files (SFF files) are not actually copied to the project directory. Rather, a symbolic link to the file is created and placed in the project's sff directory. Consequently, if the original file is moved or erased from the file system, the project will not operate correctly unless changeRun is used. This applies whether the files are added *via* the GUI or the command line.
- FASTA/FASTQ files are not actually copied to the project directory. Rather, the file path is simply recorded in the project setup (no symbolic link to the file is created). Consequently, if the original file is moved or erased from the file system, the project will not operate correctly unless changeRun and/or changeRef are used. This applies to FASTA/FASTQ read files and reference files, whether added *via* the GUI or the command line:
  - FASTA/FASTQ reads files, including Sanger reads
  - Reference files
- The changeRun and changeRef commands do not relink input files specified on the Input sub-tab of the GUI or using the command line. These files must be manually specified to complete the relinking process.
  - Trimming database file
  - Screening database file
  - Targeted regions file
  - *Genome annotation file
  - *Known SNP file
  - Accno renaming file
  - Exclude filter file
  - Include filter file

*Note that Genome annotation and Known SNP files are automatically relinked by changeRef when GOLDENPATH is used.

2.16.1.1 The changeRun Command

The changeRun command can be used in either of two modes. It can be used to change the way reads are handled in a mapping, forcing reads in one or more files to be handled either as paired-end reads or non-paired reads. It can also be used to ensure the integrity of read file information in a project's configuration if the project or any of the read files it uses is moved. To change the handling of one or more read files, the changeRun command's syntax is

```
changeRun {-p | -np} projectDir readFile1, readFile2…
```

⚠ Warning: if you use changeRun to treat a Paired-End SFF file as a non-paired end file, the linkers will remain in the reads.

If you move a project to another computer or the read files are moved or no longer accessible by the project, changeRun can be used to inform the project of the new location of the reads:

```
changeRun [-verbose [-verbose]] [-relink] projectDir [sourcePath
destinationPath]
```

If you move a project to another computer or the read files (SFF, FASTA, FASTQ) are moved or no longer accessible by the project, changeRun can be used to inform the project of the new location of the reads. The changeRun command can be used to provide information about the project's current configuration at two levels of detail, using the –verbose option. It also allows you to modify the project's read file location(s) to point to valid file(s) using the -relink option. These two options can be used together in a variety of useful ways.

```
changeRun [-verbose [-verbose]] [-relink] projectDir [sourcePath
destinationPath]
```

Used in the absence of the –relink option, `changeRun –verbose` lists the read file(s) in the project's configuration and whether each file can be accessed, but without making any changes. Using `changeRun -verbose -verbose` expands the listing to include the read sequence accnos.

```
changeRun -verbose [-verbose] projectDir
```

Used in the absence of the –verbose option, `changeRun –relink` will change paths for all read files found in the sourcePath (if sourcePath exists in project's metadata) which are also found in the destinationPath location. The contents of the source/destination files are also compared to prevent relinking to incorrect (*i.e.* wrongly (re)named) read files.

Source and destination path names should not include file names. Also, wildcard characters are not allowed in the path names. The path arguments should be specified as they appear in the project's XML (configuration) file (*i.e.* full path to read file's physical location should be used).

```
changeRun –relink projectDir sourcePath destinationPath
```

Before actually changing a project's configuration, it is useful to assess the changes that need to be made. Used in combination with –verbose –verbose, -relink will test the "relinking process" before actually committing to it. The project's configuration isn't actually changed, but the integrity of each specified relink is reported.

Once all relink conflicts can be resolved, the –relink option can be used to perform the actual relinking. This can be done with or without feedback about the results (use of a single "–verbose" option provides feedback):

```
changeRun [-verbose]–relink projectDir sourcePath destinationPath
```

If a project using –consed (consed-compatible) output is moved, it will be necessary to recreate the sff_dir link to the sff folder. For a Linux system…

```
rm sff_dir
ln –s projSffDir sff_dir
```

2.16.1.2 The changeRef Command

If an incremental mapping project or reference sequence is moved to a new location, changeRef can be used to ensure the integrity of the project's reference information. The changeRef command can be used to provide information about the project's current reference configuration at two levels of detail, using the –verbose option. It also allows you to modify the project's reference location(s) to point to valid file(s) using the –relink option. These two options can be used together in a variety of useful ways.

```
changeRef [-verbose [-verbose]] [-relink] projectDir [sourcePath
destinationPath]
```

Used in the absence of the –relink option, `changeRef –verbose` lists the reference file(s) in the project's configuration and whether each file can be accessed, but without making any changes. Using `changeRef -verbose -verbose` expands the listing to include the reference sequence accnos.

```
changeRef -verbose [-verbose] projectDir
```

Used in the absence of the –verbose option, `changeRef –relink` will change paths for all reference sequence files found in the sourcePath (if sourcePath exists in project's metadata) which are also found in the destinationPath location. The contents of the source/destination files are also compared to prevent relinking to incorrect (*i.e.* wrongly (re)named) reference sequence files.

Source and destination path names should not include file names. Also, wildcard characters are not allowed in the path names. The paths arguments should be specified as they appear in the project's XML (configuration) file (*i.e.* full path to reference sequence file's physical location should be used).

```
changeRef –relink projectDir sourcePath destinationPath
```

Before actually changing a project's configuration, it is useful to assess the changes that need to be made. Used in combination with –verbose –verbose, -relink will test the "relinking process" before actually committing to it. The project's configuration isn't actually changed, but the integrity of each specified relink is reported.

Once all relink conflicts can be resolved, the –relink option can be used to perform the actual relinking. This can be done with or without feedback about the results (use of a single "–verbose" option provides feedback):

```
changeRef [–verbose]–relink projectDir sourcePath destinationPath
```

2.16.2 One-Step Mapping: the runMapping Command

If all the reads to be mapped to the reference sequence are available at once, the GS Reference Mapper application can process them with a single command, which has the following command line structure:

```
runMapping [options] refdesc [MIDList@]filedesc…
```

…where:

- "`[options]`" are zero or more of the command line options (listed below),
- the "`refdesc`" is one of the following:
  - reffasta – a FASTA file containing the reference sequence(s),
  - refdirname – a name or path for a directory containing the reference sequence (all FASTA files in the directory will be used as the reference),
  - refgenome – a GoldenPath genome name (see the fourth Note, below),
- `–ref` (reffasta | refdirname)… -read
- the string "`-ref`", followed by multiple FASTA file names or directories, followed by the string "`-read`" (to mark the beginning of the read data files).
- each "`filedesc`" is one of the following:
  - sfffilename or
  - [regionlist:]datadir or
  - readfastafile
- and each "`MIDList`" is a multiplexing information string used to filter the set of file reads to be used in the mapping (see section 4.5 for the format of the MIDList information). If MIDs were used in the generation of the data file, then an MIDList string must be specified in order for the GS Reference Mapper to properly handle the file's reads.

The runMapping command is actually a wrapper program around the newMapping and related commands used in incremental mapping (see section 2.16.3). After the incremental mapping command is completed, runMapping then transforms the project files and folders into this one-step form (which more closely matches the output structure of older versions of the "Mapping" application). The actions taken are:

- All the mapping output files are moved up from the mapping sub-directory into the main folder
- All internal data files in the mapping sub-directory are deleted
- The 454MappingProject.xml and 454Project.xml files are deleted
- The mapping sub-directory is deleted

If the –nrm option is given, runMapping does not perform this transformation, and the resulting folder can be used for further project-based mapping (the structure of the files/folders will match that of an incremental mapping project).

For data generated on the GS FLX+ system: The path given for any of the data directories may be prepended with an optional list of regions, separated from the path by a colon. For example:

```
1-3,4,6-8:D_yyyy_mm_dd_hh_min_sec_testuser_SignalProcessing
```

The list of regions must have the following format:

- It must be a comma-separated list, ending with a colon.
- Each element of the list can either be a single region number or a dash-separated range of region numbers.
- Duplicate regions may be specified, and the regions may be specified in any order, but duplicates will be removed and the regions will be processed and reported in numerical order.
- No spaces or other characters are allowed in the string.

If the region list is not present for a data directory path, all regions of the run for that directory will be used in the mapping.

Any combination of explicit SFF files and data directory paths (with optional region lists) may be specified on the command line. For each data directory path given, the runMapping command will read the existing SFF files in the "sff" sub-directory of the data directory. If SFF files are not present in a data directory (*e.g.* for a run whose data has been processed with a version of the 454 Sequencing system software anterior to 1.0.52), the signal processing step of the GS Run Processor application must be rerun on the Data Processing folder.

The path for any filename or data directory name can also be prepended with a multiplexing information string. Section 4.5 gives further information on using MIDs.

The data analysis software installation contains a default MID configuration file, found by default at Installation_path/454/config/MIDConfig.parse. This file is read by the GS Reference Mapper and used to match MID set names and MID names with their multiplexing information. Users can edit this file to add their own MID sets (following the format and syntax described in the file), or can copy this file to create their own separate MID configuration file (and then use the "-mcf" option to specify that as the MID configuration file to be used).

Contrary to the situation with incremental mapping (see the setRef command, section 2.16.3.2), you can use only one reference FASTA file, directory or GoldenPath genome name with the runMapping command, unless the –ref/-read options are included to separate the reference files from the read data files. All the files can still contain multiple reference sequences.

The GS Reference Mapper is aware of the folder/file structure of the UCSC GoldenPath genome databases (most specifically the human genome and the other major eukaryotic genome databases). If one or more of those databases are downloaded onto the datarig, and the user sets a GOLDENPATH environment variable to the root directory containing those databases, then the genome name can be given as the reference for runMapping (*i.e.*, the command "runMapping hg18 reads.sff" will map reads.sff against the GoldenPath hg18 human genome). The GS Reference Mapper will automatically read the chromosome FASTA files, the gene annotation file and the known SNP information file.

When given a GoldenPath genome (in genome mapping mode only, not for cDNA mapping), only the main chromosome files will be used as the reference by default. The GS Reference Mapper will not read the additional "_random" or "_hap" files contained in the chromosome directory. In order to include the "_random" and "_hap" files, the "-random" option must be given to the "setRef" or "runMapping" command.

Specifically, the Mapping application looks for the reference FASTA files in the "chromosomes" sub-directory, and files named "refGene.txt" and "snp###.txt" (*i.e.*, a file like "snp130.txt" that begins with "snp", has three digits, then ends with ".txt") in the "database" sub-directory. This matches the organization of the human genome database. Other GoldenPath databases (which don't match this organization) can be used by the GS Reference Mapper application if the files are modified to match this organization. See the UCSC GoldenPath "Downloads" page for information on how their genome databases can be downloaded (http://hgdownload.cse.ucsc.edu/downloads.html).

In addition to automatically reading the gene/coding-region annotation and known SNP files when the reference is a GoldenPath genome, the GS Reference Mapper application automatically searches for specific annotation/SNP files for any reference. If the reference files are located in a directory named "chromosomes", the GS Reference Mapper will look for refGene.txt and snp###.txt files in a neighboring "database" directory (inside the same parent directory as "chromosomes"). The GS Reference Mapper also searches the same directory as the reference FASTA files for a refGene.txt and "snp###.txt". Finally, if the reference consists of a single FASTA file, then the software will look for files with ".refGene" and ".snp" suffixes (*e.g.*, if the reference file is "mygenome.fna", then files "mygenome.refGene" and "mygenome.snp" will be tried.

To disable this automated reading of the annotation/SNP files, the -noannot and -nosnp options must be included as part of the command line options.

2.16.3 Incremental Mapping: the newMapping and Related Commands

This section describes the main commands of the GS Reference Mapper application, to be used when one or more runs are to be mapped against a reference sequence or database, as part of a mapping project. These commands allow you to add runs over time and incrementally align them to the reference; and to change the reference sequence and restart mapping without losing the data set of runs. With these commands, the execution of the mapping and generation of the results can be controlled by command line options and configuration parameters (paralleling the equivalent controls in the GUI application). Such incremental mapping can be useful when, for example, you want to see intermediate mapping results on existing sequencing runs to determine if you need to carry out further runs to reach a desired depth of coverage, or just to monitor the project. Incremental mapping is also useful if you simply wish to create output using different output parameter settings.

Five commands are used in a mapping project: newMapping, setRef, addRun, removeRun and runProject. These are described in the subsections below.

2.16.3.1 The newMapping Command

The newMapping command is used to initiate a mapping project and set up a mapping project folder to contain the project data (see Section 2.16.1). Its command line structure is:

```
newMapping [option] [projDir]
```

…where:

- `[option]` -cdna and/or –multi may be used; see Section 4.2
- `[projDir]` is the optional name of the project directory

2.16.3.2 The setRef Command

The setRef command is used to assign the reference sequence(s) to which the reads from the sequencing run(s) included in the project will be mapped. Its command line structure is:

```
setRef [projDir] refdesc…
```

…where each “`refdesc`” is one of the following:

- `reffasta`: a FASTA file containing the reference sequence(s),
- refdirname: a name or path for a directory containing the reference sequence (all FASTA files in the directory will be used in the reference),
- refgenome: a GoldenPath genome name (see the third Note in the runMapping Section 2.16.2, above).

2.16.3.3 The addRun Command

The addRun command is used to add Read Data sets to existing mapping projects. Its command line structure is:

```
addRun [options] [MIDList@]filedesc…
```

…where:

- "`[options]`" are zero or more of the command line options described in section 4.2
- each "`filedesc`" is one of the following:
  - `sfffilename` or
  - `[regionlist:]datadir` or
  - `readfastafile`

Input read size constraints: The reads input for the mapping computation must be shorter than 2000 bases per read (longer sequences are ignored) and longer than 50 bases (shorter sequences are ignored). When paired end 454 reads (SFF reads) are part of the project, reads between the value of the minlen parameter and 50 bp long will be mapped onto contigs formed by mapping longer reads to the reference during a later stage in the mapping process.

- The addRun command can be executed multiple times for a mapping project (in any combination with the removeRun and runProject commands). When addRun is executed, it adds (not "resets") the given runs/regions or SFF files to the list of sequencing data used in the project. (It does not reset the list of sequence data). The reads used in the mapping (runProject, see section 2.16.3.5 below) are the union of the data from all executions of addRun for the project.
- The commands addRun, removeRun, and runProject can be executed in any combination and in any order, in a mapping project.

2.16.3.4 The removeRun Command

The removeRun command is used to remove Read Data sets from existing mapping projects. Its command line structure is:

```
removeRun [projDir] (sffname or readfastafilepath)…
```

- This command is more conveniently carried out from the GUI application. When called from the command line, it requires the SFF file name(s) *given in the project sff sub-directory* or the FASTA/FASTQ file name(s) *given in the project configuration file 454MappingProject.xml*. These names may not match the original names of the corresponding files or may not be known to the user, especially if the software had to rename any of them to ensure name uniqueness. The filenames of the SFF files in the sff sub-directory are assigned by the GS Reference Mapper application to ensure uniqueness for all the files, while trying to preserve the original names when possible.
- The execution of this command does not physically remove the file(s) from the project sff sub-directory or from any existing mapping. The file(s) and the reads they contain are only marked for removal by the removeRun command, and are actually removed only the next time the project is computed (*via* the runProject command).
- The commands addRun, removeRun, and runProject can be executed in any combination and in any order, in a mapping project.

2.16.3.5 The runProject Command

The runProject command performs the actual mapping computation for a project and generates the results of the mapping. Its command line structure is:

```
runProject [options] [projDir]
```

2.16.4 Genetic Code Specification

The -gencode option can be used to specify various types of genetic codes (bacterial, mitochondrial, eukaryotic, *etc.*) for predicting protein translation products during variant calling. The genetic code description can be specified in either a multi-line format (64 lines, with one codon per line) or a single-line format (64 characters, with one codon per character).

```
runMapping -gencode codedesc refdesc [MIDList@]filedesc…
```

2.16.4.1 Multi-line Genetic Code Description

The multi-line genetic code description has one row for each codon, and each row includes a space-separated codon, single letter amino acid abbreviation, and optional three letter amino acid abbreviation (empty or blank lines are ignored).

```
TTC F Phe
TTA L Leu
TTG L Leu
TCT S Ser
```

2.16.4.2 Single-line Genetic Code Description

The single-line genetic code description is a single line containing a 64 character sequence representing the genetic code, http://www.ncbi.nlm.nih.gov/Taxonomy/taxonomyhome.html/index.cgi?chapter=cgencodes, for example:

```
FFLLSSSSYY**CC*WLLLLPPPPHHQQRRRRIIIMTTTTNNKKSSRRVVVVAAAADDEEGGGG
```

The single-line description may either be specified in a file or entered directly on the command line.

2.16.5 Multiplex Mapping Project Setup

A multiplex one-step mapping project is set up in exactly the same way as an ordinary one-step mapping project, except that–multi and –tsv options are used with runMapping, and read data sets cannot be filtered by pre-pending an MID list to the filename. If read files have been specified within the samples.tsv file, any read files passed as an argument to runMapping will be ignored.

```
runMapping [multiplex options: -batchsize –maxsvf –minsvf –mrerun] –multi
–tsv samples.tsv refdesc filedesc
```

There are several options specific to computing a multiplex project, including –batchsize (number of sample projects to be computed simultaneously), -maxsvf and -minsvf (controlling the range of reported variant frequencies) and -mrerun (controlling project re-computation).

A multiplex incremental mapping project is set up in exactly the same way as an ordinary incremental mapping project, except that the –multi option is used with the newMapping command, and read data sets are added using the addSample command (Section 2.16.5.1) rather than the addRun command. The runProject command uses the same multiplex-specific computing options as runMapping.

```
newMapping [options: -cdna (-cref | -gref)] –multi projDir

setRef [options: (-gref | -cref) –random] [projDir] refdesc

addSample [multiplex options: -clear –datapath] –tsv samples.tsv projDir
[[MIDList@]filedesc]

runProject [multiplex options: -batchsize –maxsvf -minsvf -mrerun] [projDir]
```

For all of these commands;

- The arguments in square brackets [ ] are optional.
- The '|' symbol means 'OR'.
- 'options' refers to the optional command arguments available.
- 'projDir' is the path of the data analysis project directory.
- 'refdesc' is a path/name to a reference sequence file in FASTA format.
- 'filedesc' = (sfffile | fnafile | [regionlist:]analysisDir), referring to a read data file in one of the following formats:
    - an sfffilename
    - a [regionlist:]datadir
    - a readfastafile
    - a space-separated list of any of the above
- '[MIDList@]filedesc', where each 'MIDList' is a multiplexing information string used to filter the set of file reads to be used in the mapping (if MIDs were used in the generation of the read data file, then an MIDList string must be specified in order for the mapper to properly handle the file's reads.) For details about using MIDs, see section 4.5.

2.16.5.1 The addSample Command

The addSample command is used in place of addRun to add read files to a multiplex project and to define the relationship between sample name and MIDs in use. If you attempt to use addRun in a multiplex project, you will receive an error that says "Error: This a multiplex project. The addSample command must be used instead of addRun.".

```
addSample [options: (-p | -np) –lib (-mcf | -custom) –clear -datapath]
–tsv samples.tsv projDir [[MIDList@]filedesc]
```

… where:

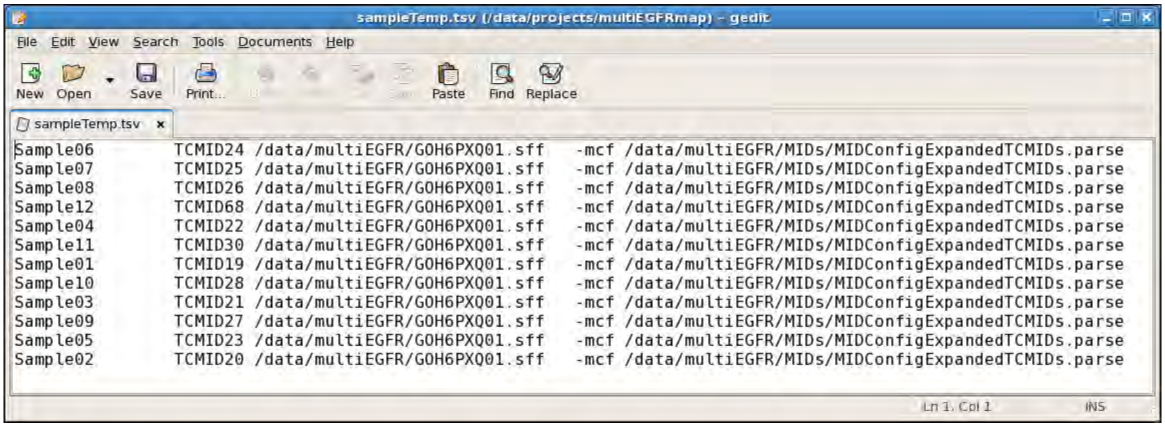
- The arguments in square brackets [ ] are optional.
- The '|' symbol means 'OR'.
- 'options' refers to the optional command arguments available.
- 'projDir' is the path of the data analysis project directory.
- 'filedesc' = (sfffile | fnafile | [regionlist:]analysisDir), referring to a read data file in one of the following formats:
    - an sfffilename
    - a [regionlist:]datadir
    - a readfastafile
    - a space-separated list of any of the above
- '[MIDList@]filedesc', where each 'MIDList' is a multiplexing information string used to filter the set of file reads to be used in the mapping (if MIDs were used in the generation of the read data file, then an MIDList string must be specified in order for the mapper to properly handle the file's reads.) For details about using MIDs, see section 4.5.

The –tsv option is used to specify a sample association file (see Section 2.16.5.2), the –clear option removes the sample association file from a multiplex project, and the –datapath option specifies an optional datapath to a directory of reads in a multiplex project.

If read files have been specified within the sample association file, any read files passed as an argument to addSample will be ignored.

2.16.5.2 Sample Association Files (-tsv option)

A sample association file can consist of two, three or four columns (see Figure 81 for a two column example, and see Figure 102 for a four column example). The first two columns (sample name and MID name/sequence) are imported by the GUI, but the optional third column (read files) and fourth column (addRun options such as –mcf or -p) are used only by the CLI command addSample. If a sample is associated with multiple read files, they are listed in the third column separated by commas.


sampleTemp.tsv (/data/projects/multiEGFRmap) - gedit
File Edit View Search Tools Documents Help
New Open Save Print... Paste Find Replace
sampleTemp.tsv
Ln 1, Col 1
INS

```
Sample06        TCMID24 /data/multiEGFR/GOH6PXQ01.sff   -mcf /data/multiEGFR/MIDs/MIDConfigExpandedTCMIDs.parse
Sample07        TCMID25 /data/multiEGFR/GOH6PXQ01.sff   -mcf /data/multiEGFR/MIDs/MIDConfigExpandedTCMIDs.parse
Sample08        TCMID26 /data/multiEGFR/GOH6PXQ01.sff   -mcf /data/multiEGFR/MIDs/MIDConfigExpandedTCMIDs.parse
Sample12        TCMID68 /data/multiEGFR/GOH6PXQ01.sff   -mcf /data/multiEGFR/MIDs/MIDConfigExpandedTCMIDs.parse
Sample04        TCMID22 /data/multiEGFR/GOH6PXQ01.sff   -mcf /data/multiEGFR/MIDs/MIDConfigExpandedTCMIDs.parse
Sample11        TCMID30 /data/multiEGFR/GOH6PXQ01.sff   -mcf /data/multiEGFR/MIDs/MIDConfigExpandedTCMIDs.parse
Sample01        TCMID19 /data/multiEGFR/GOH6PXQ01.sff   -mcf /data/multiEGFR/MIDs/MIDConfigExpandedTCMIDs.parse
Sample10        TCMID28 /data/multiEGFR/GOH6PXQ01.sff   -mcf /data/multiEGFR/MIDs/MIDConfigExpandedTCMIDs.parse
Sample03        TCMID21 /data/multiEGFR/GOH6PXQ01.sff   -mcf /data/multiEGFR/MIDs/MIDConfigExpandedTCMIDs.parse
Sample09        TCMID27 /data/multiEGFR/GOH6PXQ01.sff   -mcf /data/multiEGFR/MIDs/MIDConfigExpandedTCMIDs.parse
Sample05        TCMID23 /data/multiEGFR/GOH6PXQ01.sff   -mcf /data/multiEGFR/MIDs/MIDConfigExpandedTCMIDs.parse
Sample02        TCMID20 /data/multiEGFR/GOH6PXQ01.sff   -mcf /data/multiEGFR/MIDs/MIDConfigExpandedTCMIDs.parse
```

Figure 102: Sample association file with four columns. The four columns hold sample names, MID name(s) or sequence(s), read file(s), and addRun options, respectively.

The sample association file also supports the special case where multiple MIDs are associated with an individual sample name. Multiple MIDs are specified by separating the MID names or sequences with commas in the second column of the sample association file.

Sample definition rules (CLI):

- All samples in a multiplex mapping project should use MIDs that were designed as part of the same compatible MID group.
- Sample names should not contain special characters (use only letters, numbers, underscores, and dashes).
- Sample names can be associated with multiple read files (separated by commas).
- Sample names can be associated with multiple MID names (separated by commas).
- Sample names can be associated with multiple MID sequences (separated by commas).

  ```
  ACGAGTGCGT/2,ACGCTCGACA/2,AGACGCACTC/2 specifies three MIDs with
  two allowed errors each
  ```
- A sample definition must be specified on a single line of the sample association file. An error will be generated if a sample name is listed twice in the sample association file.
- With multiple MIDs and multiple read files per sample, the sample names are associated with every combination of the cross between the multiple MIDs and multiple read files that were specified, even if only a subset of the combinations is actually associated with the sample.

  ```
  Sample1 = MID1-readfile01.sff, MID2-readfile02.sff
  =>
  Sample1   MID1,MID2   readfile01.sff,readfile02.sff
  =>
  Sample1 definition = MID1-readfile01.sff, MID1-readfile02.sff,
  MID2-readfile01.sff, MID2-readfile02.sff
  ```
- An error will be generated if an MID-read file combination is associated with multiple sample names in a sample association file.

2.17 GS Reference Mapper cDNA / Transcriptome Options

These descriptions are for options specific to cDNA / transcriptome mapping projects.

cDNA / transcriptome mapping option "**-cdna**": To specify a cDNA / transcriptome mapping project, use the **-cdna** option with the runMapping or newMapping commands.

cDNA reference option “**-cref**”: To specify a cDNA (transcript) reference sequence for the mapping project, use the **-cref** option on the command line for the setRef command or with the runMapping, newMapping, or runProject commands.

cDNA reference option “**-gref**”: To specify a gDNA (genomic DNA) reference sequence for the mapping project, use the **-gref** option on the command line for the setRef command or with the runMapping, newMapping, or runProject commands.

Renaming file option **"-accno"**: To specify a tab-delimited file containing annotation data, use the –**accno** option (see more details on the use of this option in the next section).

refGene file option "**-annot**": To specify a tab-delimited file containing gene, transcript, and exon information, use the **-annot** option (see more details on the use of this option are in the next section).

The reference type (cDNA or gDNA) can be automatically detected under certain circumstances, thus removing the need to specify the reference type on the command line. The logic employed depends on whether or not the GOLDENPATH environmental variable is in use. See Appendix 5.4.

2.17.1 Annotation Input Files for Mapping Transcriptomes

Annotation files specified using the '-annot' option in the CLI for mapping to cDNA / transcriptome sequence data can be of several different types. They are described in Table 3, below.

Source	Reference	Description
GoldenPath annotation data	Mapping to a cDNA ref	If a project is using the GoldenPath reference file named refMrna.fa or a compatible reference file (*i.e.*, one using the NCBI style accession numbers), GoldenPath annotation data can be incorporated into the project by setting up the required directory structure (the same as used for working with SNP data) that contains the required GoldenPath database files and setting the GOLDENPATH environment variable to the path culminating in the "bigZips" directory of the GoldenPath directory structure. The only required GoldenPath annotation file is refLink.txt. The GoldenPath annotation file productName.txt, if present, can be used to incorporate gene-level descriptions into the output files.
GoldenPath annotation data	Mapping to a gDNA Ref	Annotation information for genes, transcripts, and exons can be obtained from the GoldenPath annotation file named refGene.txt. This gives transcript start and stop positions, coding sequence stop and start positions and exon boundaries for each gene.
Reference sequence file	all	In cases where GoldenPath annotation data is unavailable, annotations may be obtained from the description lines for each reference sequence in reference file. The tag/value pair "gene=*geneName"* must be present on the description line. If "gene=*geneName"* is not present, transcript and/or gene descriptions can only be incorporated into the output by using a renaming file (see below).
renaming file	all	A renaming file can be used to either provide more meaningful names for genes and/or transcripts and/or to incorporate transcript and gene descriptions into the output files. The structure of a renaming file is as follows: OLD_ID<TAB>NEW_ID<TAB>DESCRIPTION, where OLD_ID is either the transcript identifier or gene name that currently exists in the project or incoming annotation data, and NEW_ID is the new identifier you want to map to the old identifier. DESCRIPTION is the transcript- or gene-level description you want to map to the NEW_ID. Note that it is *not* necessary to rename a gene or transcript if all you want to do is incorporate gene- or transcript-level descriptions into the output. In this scenario, the OLD_ID and NEW_ID would be identical and would correspond to the appropriate description. Note also that it is possible to specify *both* transcript- and gene-level descriptions in the same renaming file. The way to do this is to put the transcript-level descriptions on the same lines as the corresponding transcript identifiers and the gene-level descriptions on the same lines as the corresponding gene identifiers.

Table 3: Annotation input files for mapping cDNA / transcriptome data sets.

2.18 GS Reference Mapper Output

2.18.1 Output File Descriptions

Output Files produced by the GS Reference Mapper are described briefly below. GUI settings and command line options that control the content type of a file are given when applicable.

File name	Description	Mapping – Genomic Project	Mapping- cDNA Project	GUI Conditional output option	CLI Conditional output option
454AlignmentInfo.tsv	Tab-delimited file giving position-by-position consensus base and flow signal information. Output conditionally (using –info/-noinfo options or checkbox selection on GUI Parameters Tab Output Sub-tab) if there are more than 4M reads or the total length of reference sequences exceeds 40Mbp.	X	X	Alignment info selection: Output Output small No output N/A	-info (default) -noinfo -infoall
454AllContigs.fna	FASTA file of all the consensus basecalled contigs longer than 100 bases. The minimum length output can be changed by using the [-a #] option in the CLI or changing the All contig threshold in the GUI Parameters Tab, Output Sub-tab. Some contigs slightly shorter than this may be included due to the two-phase nature by which contig consensi are determined.	X	X	Parameters: All contig threshold	-a #
454AllContigs.qual	Corresponding Phred-equivalent quality scores for each base in the consensus contigs in 454AllContigs.fna.	X	X		
454AllDiffs.txt	This file contains the list of variations (of at least 2 reads) relative to the reference sequence or to other reads aligned at a specific location. SNPs, insertion-deletion pairs, multi-homopolymer insertion or deletion regions, and single-base overcalls and undercalls are reported.	X	X	Single read variant Selection. If used, single read variations are also included	-srv Single reads variations are also included if this option is used

File name	Description	Mapping – Genomic Project	Mapping- cDNA Project	GUI Conditional output option	CLI Conditional output option
454AllFusionVariantTable.txt	Mapping information for the chimeric reads composing the gene-to-gene and gene-to-non-gene fusion events found in the structural rearrangement files (particularly useful for mapping transcriptomic data to a genomic reference).	X	X	n/a	-nosv disables
454AllStructRearrangements.txt	This file provides explicitly labeled classifications and information for the types and coordinates of all structural rearrangements observed in the reads of the data set analyzed, relative to the reference. See section 2.18.1.16 for details	X	X	n/a	-nosv disables
454AllStructVars.txt	A text file containing a section listing the rearrangement points, followed by a section listing the rearrangement regions. (The associated data describing rearrangement points and rearrangement regions output is found in Section 2.18.1.15.)	X	X	n/a	-nosv disables
454Contigs.ace	ACE format file that can be loaded by third-party viewer programs that support the ACE format. The output can be a single file for the entire project or a folder containing individual files for each contig in the mapping.	X		ACE Format Selection Ace read mode Selection	-nobig, -ace, -consed, -noace, -ar, -at, -ad, -consed16
454Contigs.bam	BAM format file that can be loaded by third-party viewer programs that support the BAM format. The output is a single file containing all multiple alignments for the contig.	X		BAM Format	-nobam -bam
454GeneStatus.txt	A file reporting statistics on the number of reads mapped exclusively to each gene (for cDNA mapping projects only)		X		
454HCDiffs.txt	This file contains the list of high confidence variations relative to the reference sequence or to other reads aligned at a specific location. See section 2.18.1.14 for details	X	X		

File name	Description	Mapping – Genomic Project	Mapping- cDNA Project	GUI Conditional output option	CLI Conditional output option
454HCFusionVariantTable.txt	Mapping information for the chimeric reads composing the high confidence gene-to-gene and gene-to-non-gene fusion events found in the structural rearrangement files (particularly useful for mapping transcriptomic data to a genomic reference).	X	X	n/a	-nosv disables
454HCStructRearrangements.txt	This file provides explicitly labeled classifications and information for the types and coordinates of high confidence structural rearrangements observed in the reads of the data set analyzed, relative to the reference. See section 2.18.1.16 for details	X	X	n/a	-nosv disables
454HCStructVars.txt	A text file containing a section listing the high confidence rearrangement points, followed by a section listing the high confidence rearrangement regions. (The associated data describing rearrangement points and rearrangement regions output is found in section2.18.1.15.) The GS Reference Mapper application uses a combination of flow signal information, quality score information and variant type information to determine if a variant is High-Confidence.	X	X	n/a	-nosv disables
454LargeContigs.fna	FASTA file of all the "large" consensus basecalled contigs contained in 454AllContigs.fna (>500bp). This can be changed by using the [-l #] option in the CLI or changing the Large contig threshold in the GUI Parameters Tab, Output Sub-tab.	X		Parameter: Large contig threshold	-l #
454LargeContigs.qual	Corresponding Phred-equivalent quality scores for each base in the "large" consensus contigs in 454LargeContigs.fna.	X			
454TrimmedReads.fna	FASTA file of the trimmed reads used in the mapping	X	X	Parameter: Output trimmed reads	-tr

File name	Description	Mapping – Genomic Project	Mapping- cDNA Project	GUI Conditional output option	CLI Conditional output option
454TrimmedReads.qual	Corresponding Phred-equivalent quality scores for each base in the reads in 454TrimmedReads.fna	X	X	Parameter: Output trimmed reads	-tr
454MappingQC.xls	This file reports metrics calculated using the mapping results. The columns reported differ for genomic and cDNA projects and are described in sections 2.18.1.11 and 2.18.2.2.	X	X		
454NewblerMetrics.txt	File providing various mapping metrics, including the number of input runs and reads, the number and size of the large consensus contigs and the number of all consensus contigs. Reference metrics are provided as well.	X	X		
454NewblerProgress.txt	A text log of the messages sent to standard output during the mapping computation.	X	X		
454PairAlign.txt	A text file giving the pairwise alignment(s) of the overlaps used in the mapping computation (only produced when using the -pair option [or -pt or -pairt option for the tab-delimited version of the file]).	X	X	Pairwise alignment selection	-pair, -pt or -pairt,
454ReadStatus.txt	Tab-delimited text file providing a per-read report of the status of each read in the mapping. The mapped location of all uniquely mapped reads is also reported. If the –reg option is used to specify regions of a reference, such as for a NimbleGen Sequence Capture experiment, then the per-read 'in-region' and 'out-of-region' status is also given.	X	X		-nobig
454RefStatus.txt	A file reporting the statistical information on the number or reads mapping to each reference sequence.	X	X		

File name	Description	Mapping – Genomic Project	Mapping- cDNA Project	GUI Conditional output option	CLI Conditional output option
454TagPairAlign.txt	A text file showing the pairwise alignments of short tag reads, which are not part of the standard mapping computation, but are mapped to the reference in a later computation step (Section 2.18.1.10).	X	X	Pairwise alignment selection	-pair, -pt or -pairt
454TrimStatus.txt	Tab-delimited text file providing a per-read report of the original and revised trimpoints used in the mapping.	X	X		-nobig

Table 4: GS Reference Mapper Output Files.

NewblerProgress.txt: If runs are added incrementally and multiple executions of runProject occur, the output messages are appended to this file. If the "Incremental Reference Mapper Analysis" checkbox option is not selected in the GUI application, or the "-r" option is given on the runProject command line, the GS Reference Mapper deletes intermediate data and "restarts" the mapping computation, and the NewblerProgress.txt file is deleted and restarted as well.

2.18.1.1 454AlignmentInfo.tsv

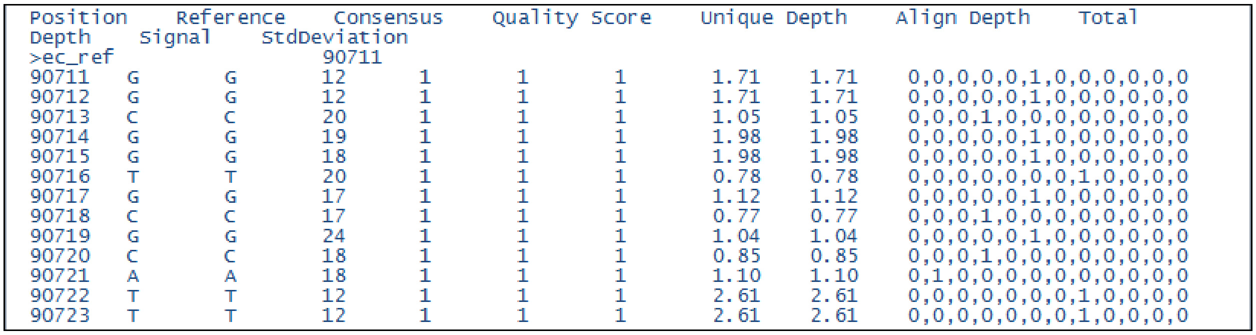
The 454AlignmentInfo.tsv file (Figure 103) contains position-by-position summary information about the consensus sequence for the contigs generated by the GS Reference Mapper application, listed one nucleotide per line (in a tab-delimited format). Lines are generated for any position of the reference for which any of the reported Depths is greater than 0. The 454AlignmentInfo.tsv file is output conditionally (depending on the **-info**/**-infoall**/**-noinfo** options or the selection made on the GUI Parameters Tab Output Sub-tab). By default, this file is only output if there are fewer than 4 million input reads and the total length of reference sequences is less than 40Mbp. For larger projects, **-info** or **-infoall** or the corresponding GUI **Output** selection for **Alignment Info** must be used to generate this file. From the command line, the **-infoall** option is used to tell the mapper to report all lines of the 454AlignmentInfo.tsv file, even if there is no coverage. (This option guarantees that every reference position is reported in the 454AlignmentInfo.tsv file, with the exception that if regions are specified in the mapper, only the regions are reported (but every position in the regions will be reported). From the command line, the **–nft** option or the corresponding GUI **Output** selection for **Alignment Info** is used to add a column to the 454AlignmentInfo.tsv file that represents nucleotide and gap counts of multiple alignments at each position.

The columns of each line contain the following information:

1. **Position** – the position in the reference.
2. **Consensus** – the consensus nucleotide for that position in the reference.
3. **Quality Score** – the quality score of the consensus base.
4. **Unique Depth** – the number of non-duplicate, uniquely mapping reads that align at that location.
5. **Align Depth** – the number of uniquely mapping reads aligned at that location.
6. **Total Depth** – an estimated unique plus repeat mapping depth at that location, where the repeat depth is estimated. The estimate is made by randomly assigning each repeat read to one of its assigned locations and incrementing the existing count for that location.
7. **Signal** – the average signal of the read flowgrams, for the flows that correspond to that position in the alignment.
8. **StdDeviation** – the standard deviation of the read flowgram signals at the corresponding flows.
9. **Region Status** (with –**reg** option) – identifies each mapped base as "IN" if they map within the target regions or "EXT" if they map in the extended target region, but not in the target regions (see Section 4.13.10 for more details). This column is not present without the –**reg** option.
10. **Nucleotide Frequency table** (with **–nft** option) – A comma-delimited list of base and gap counts at each position of the reference (note that the column has no header). The values in the list are the forward and reverse read counts of read bases (nucleotides) at each position, listed in the following order:

    A fwd. **A** rev. **C** fwd. **C** rev. **G** fwd. **G** rev. **T** fwd. **T** rev. **N** fwd, **N** rev, **gap** (-) fwd, **gap** (-) rev.

Prior to lines for each contig, a header line beginning with a '>' displays the Reference sequence accno, as shown in Figure 103.

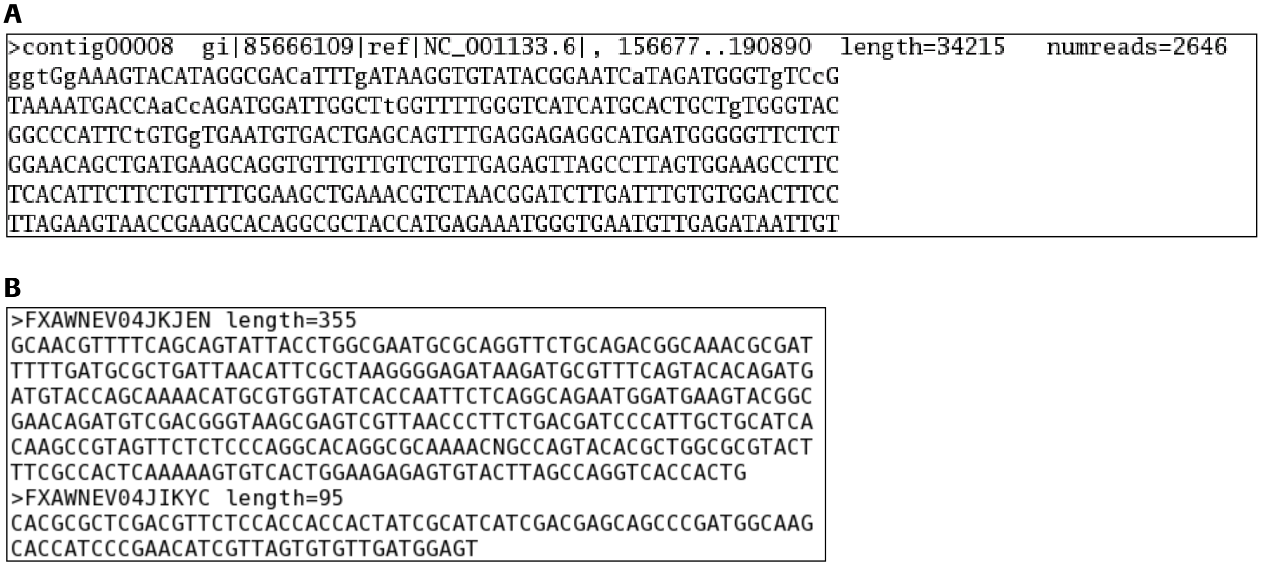


```
Position    Reference    Consensus    Quality Score    Unique Depth    Align Depth    Total
Depth    Signal    StdDeviation
>ec_ref                 90711
90711   G       G       12      1       1       1       1.71    1.71    0,0,0,0,0,1,0,0,0,0,0,0
90712   G       G       12      1       1       1       1.71    1.71    0,0,0,0,0,1,0,0,0,0,0,0
90713   C       C       20      1       1       1       1.05    1.05    0,0,0,1,0,0,0,0,0,0,0,0
90714   G       G       19      1       1       1       1.98    1.98    0,0,0,0,0,1,0,0,0,0,0,0
90715   G       G       18      1       1       1       1.98    1.98    0,0,0,0,0,1,0,0,0,0,0,0
90716   T       T       20      1       1       1       0.78    0.78    0,0,0,0,0,0,0,1,0,0,0,0
90717   G       G       17      1       1       1       1.12    1.12    0,0,0,0,0,1,0,0,0,0,0,0
90718   C       C       17      1       1       1       0.77    0.77    0,0,0,1,0,0,0,0,0,0,0,0
90719   G       G       24      1       1       1       1.04    1.04    0,0,0,0,0,1,0,0,0,0,0,0
90720   C       C       18      1       1       1       0.85    0.85    0,0,0,1,0,0,0,0,0,0,0,0
90721   A       A       18      1       1       1       1.10    1.10    0,1,0,0,0,0,0,0,0,0,0,0
90722   T       T       12      1       1       1       2.61    2.61    0,0,0,0,0,0,0,1,0,0,0,0
90723   T       T       12      1       1       1       2.61    2.61    0,0,0,0,0,0,0,1,0,0,0,0
```

Figure 103: 454AlignmentInfo.tsv file portion example.

2.18.1.2 fna and qual files: 454AllContigs, 454LargeContigs, 454TrimmedReads

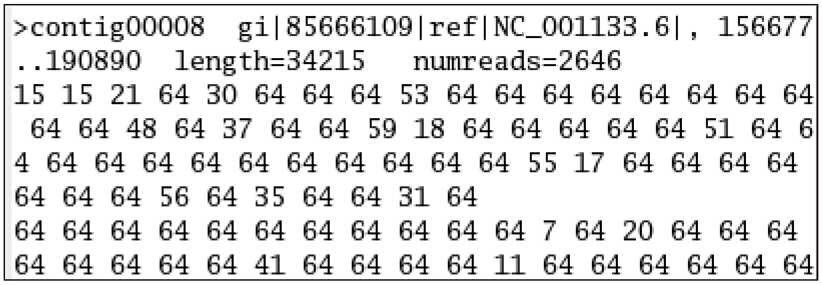
These files contain the nucleotide sequences of all the contigs and trimmed reads (Figure 104), and associated nucleotide Quality Scores (Phred-equivalent; Figure 105) produced by the GS Reference Mapper application. The AllContig and LargeContig output lengths are specified in the GUI or by CLI options described in Table 4, above. The TrimmedReads output is generated by specifying the CLI option –tr or by checking the “Output trimmed reads” checkbox on the Parameters tab/Output sub-tab. With the –reg option, the output is restricted to reads in the extended target regions (see Section 4.13.10 for more details).



```
>contig00008  gi|85666109|ref|NC_001133.6|, 156677..190890  length=34215   numreads=2646
ggtGgAAAGTACATAGGCGACaTTTgATAAGGTGTATACGGAATCaTAGATGGGTgTCcG
TAAAATGACCAaCcAGATGGATTGGCTtGGTTTTGGGTCATCATGCACTGCTgTGGGTAC
GGCCCATTCtGTGgTGAATGTGACTGAGCAGTTTGAGGAGAGGCATGATGGGGGTTCTCT
GGAACAGCTGATGAAGCAGGTGTTGTTGTCTGTTGAGAGTTAGCCTTAGTGGAAGCCTTC
TCACATTCTTCTGTTTTGGAAGCTGAAACGTCTAACGGATCTTGATTTGTGTGGACTTCC
TTAGAAGTAACCGAAGCACAGGCGCTACCATGAGAAATGGGTGAATGTTGAGATAATTGT
```

```
>FXAWNEV04JKJEN length=355
GCAACGTTTTCAGCAGTATTACCTGGCGAATGCGCAGGTTCTGCAGACGGCAAACGCGAT
TTTTGATGCGCTGATTAACATTCGCTAAGGGGAGATAAGATGCGTTTCAGTACACAGATG
ATGTACCAGCAAAACATGCGTGGTATCACCAATTCTCAGGCAGAATGGATGAAGTACGGC
GAACAGATGTCGACGGGTAAGCGAGTCGTTAACCCTTCTGACGATCCCATTGCTGCATCA
CAAGCCGTAGTTCTCTCCCAGGCACAGGCGCAAAACNGCCAGTACACGCTGGCGCGTACT
TTCGCCACTCAAAAAGTGTCACTGGAAGAGAGTGTACTTAGCCAGGTCACCACTG
>FXAWNEV04JIKYC length=95
CACGCGCTCGACGTTCTCCACCACCACTATCGCATCATCGACGAGCAGCCCGATGGCAAG
CACCATCCCGAACATCGTTAGTGTGTTGATGGAGT
```

Figure 104: Portion examples of *.fna files for a genomic mapping project. (A) 454AllContigs.fna; (B) 454TrimmedReads.fna.



```
>contig00008  gi|85666109|ref|NC_001133.6|, 156677
..190890  length=34215   numreads=2646
15 15 21 64 30 64 64 64 53 64 64 64 64 64 64 64 64
 64 64 48 64 37 64 64 59 18 64 64 64 64 64 51 64 6
4 64 64 64 64 64 64 64 64 64 64 55 17 64 64 64 64
64 64 64 56 64 35 64 64 31 64
64 64 64 64 64 64 64 64 64 64 64 7 64 20 64 64 64
64 64 64 64 64 41 64 64 64 64 11 64 64 64 64 64 64
```

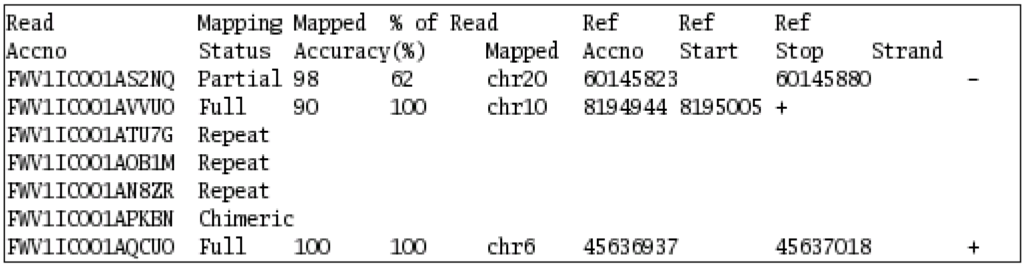
Figure 105: 454AllContigs.qual file portion example.

2.18.1.3 454ReadStatus.txt

The 454ReadStatus.txt file (Figure 106) contains the status identifiers for all the reads used in the mapping computation, plus the position for each mapped read's alignment within the reference (unless the mapping status is "chimeric"). The reads are listed one per line, in tab-delimited format. Furthermore, if the "-**reg**" option is given (to specify a set of regions of the reference, such as in a NimbleGen sequence capture experiment), then the per-read "InRegion", "InExtRegion", or "OutOfRegion" status is given, to describe which reads aligned in the target regions, in the regions flanking the target regions, and which ones aligned elsewhere in the genome (see Section 4.13.10).

The following columns are reported:

1. **Read Accno** – Accession number of the input read.
2. **Mapping Status** – status of the read in the mapping, which can be one of the following:
   a. **Full** – the read is fully aligned to the reference (every base).
   b. **Partial** – only part of the read aligned to the reference.
   c. **Chimeric** – part of the read aligned to one location on the reference and a different part of the read aligned to a different reference or to a distant location on the same reference.
   d. **Repeat** – the read aligned equally well to multiple locations in the reference.
   e. **Unmapped** – the read did not align to the reference.
   f. **TooShort** – the trimmed read was too short to be used in the computation (shorter than 50 bases and longer than minlen bases, unless 454 paired end reads are included in the data set, in which case, all reads at least "minlen" bases are used and 454NewblerMetrics.txt will report the value of numberTooShort as 0 since any shotgun reads at least as long as the minimum read length will be used in the mapping).
3. **Mapped Accuracy** – the percentage identity of the alignment, rounded to the nearest whole number (reads with 'Full' and 'Partial' status only).
4. **% of Read Mapped** – the percentage of the read that occurs in the alignment (reads with 'Full' or 'Partial' status only).
5. **Ref Accno** – the accno of the reference sequence to which the read is aligned.
6. **Ref Start** – the position in the reference sequence where the read's alignment begins.
7. **Ref Stop** – the position in the reference sequence where the read's alignment ends.
8. **Strand** – the orientation of the read's alignment relative to the reference sequence. A '+' indicates the alignment orientation of the read is the same as the orientation of the reference. A '-' indicates the alignment orientation of the read is opposite to the orientation of the reference.
9. **Region Status** (with –**reg** option) – indicates whether or not the read intersects a target region as defined by the parameter given with the –**reg** option: 'InRegion' means that the read intersects a target region, 'InExtRegion' means that the read is in the extended target region but not in the target region, and 'OutOfRegion' means that the read does not intersect any extended target regions. This column is not present without the –**reg** option.



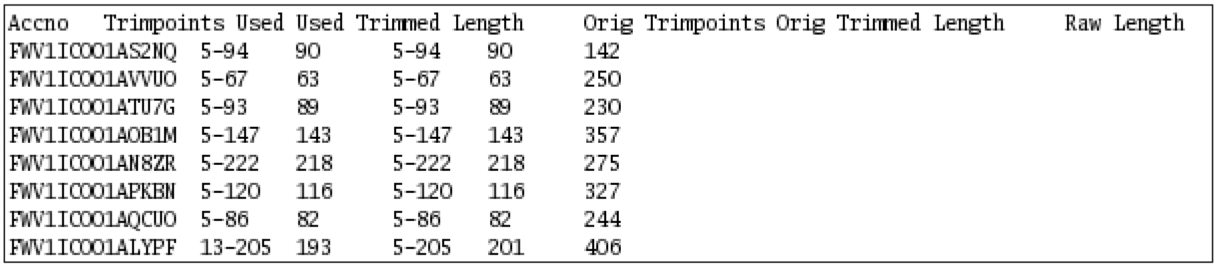
```
Read            Mapping Mapped  % of Read       Ref     Ref     Ref
Accno           Status  Accuracy(%)     Mapped  Accno   Start   Stop    Strand
FWV1IC001AS2NQ  Partial 98      62      chr20   60145823        60145880        -
FWV1IC001AVVUO  Full    90      100     chr10   8194944 8195005 +
FWV1IC001ATU7G  Repeat
FWV1IC001AOB1M  Repeat
FWV1IC001AN8ZR  Repeat
FWV1IC001APKBN  Chimeric
FWV1IC001AQCUO  Full    100     100     chr6    45636937        45637018        +
```

Figure 106: 454ReadStatus.txt file portion example for a Mapping Project (-reg option not used).

2.18.1.4 454TrimStatus.txt

This file contains a per-read report of the original and revised trimpoints used in the mapping project, where the revised trimpoints include the effects of primer, vector and/or quality trimming of the original input reads (Figure 107). Each line contains the following information (these are the columns in the tab-delimited format):

1. **Accno** – accession number of the input read.
2. **Trimpoints Used** – the final trimpoints used in the mapping, in #-# format.
3. **Trimmed Length** – the final trimmed length of the read.
4. **Orig. Trimpoints** – the original trimpoints of the read, found in the SFF or FASTA file.
5. **Orig. Trimmed Length** – the original trimmed length of the read.
6. **Raw Length** – the length of the raw read (without any trimming).



```
Accno   Trimpoints Used Used Trimmed Length     Orig Trimpoints Orig Trimmed Length     Raw Length
FWV1IC001AS2NQ  5-94    90      5-94    90      142
FWV1IC001AVVUO  5-67    63      5-67    63      250
FWV1IC001ATU7G  5-93    89      5-93    89      230
FWV1IC001AOB1M  5-147   143     5-147   143     357
FWV1IC001AN8ZR  5-222   218     5-222   218     275
FWV1IC001APKBN  5-120   116     5-120   116     327
FWV1IC001AQCUO  5-86    82      5-86    82      244
FWV1IC001ALYPF  13-205  193     5-205   201     406
```

Figure 107: 454TrimStatus.txt file portion example.

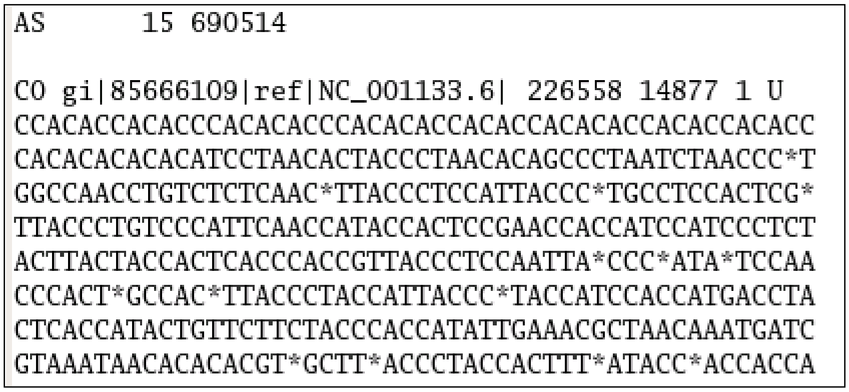
2.18.1.5 454Contigs.ace or ace/ContigName.ace or consed/…

This viewer-ready genome file shows all the reference sequences to which reads mapped. The file allows the display of how the individual reads aligned to those reference sequences, in an ACE format file suitable for use in various third-party sequence finishing programs (Figure 108). (The freeware "clview" application can be downloaded from: http://compbio.dfci.harvard.edu/tgi/software/; a full description of the .ace file format can be found at: http://bozeman.mbt.washington.edu/consed/consed.html.) It should be noted, however, that such third-party viewing software will not be able to make full use of the flowspace mapping information available with 454

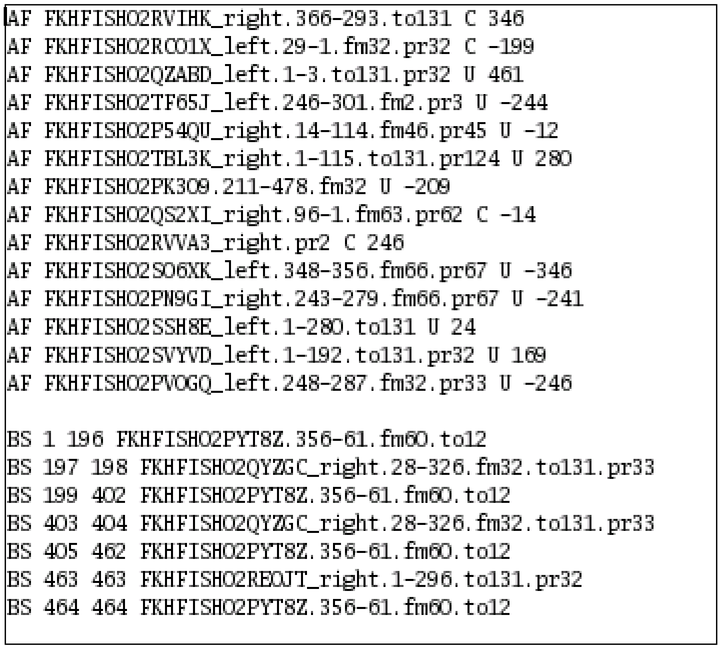
Sequencing reads, and that conversely some of the third-party program's functions (*e.g.* involving sequence chromatogram input) are not usable with 454 Sequencing data sets. Nonetheless, these programs may be useful to view and assess read characteristics and coverage depth in regions of interest.

The data software analysis applications can also output the mapping results in a complete directory structure suitable for use as input to the consed software (see the above link for a description of this structure and its files). When the appropriate option is selected, a "consed" sub-directory is produced in the output (or project) directory for the computation. This directory contains the sub-directories, the ACE file and the PHD file, so that the functions of consed (viewing traces, editing reads, auto finishing) can be performed on the mapping. In order to integrate the 454 Sequencing reads (and their SFF files) into the consed data, an extra "sff_dir" directory is created in the consed sub-directory, and a number of consed options are automatically specified in this directory (see the "consed/edit_dir/.consedrc" file for the options specified by the generated structure). One option tells consed to use the "sff2scf" command to access any trace information requested by a user. See Section 3.3 for a description of the sff2scf command, and how it can be used to generate synthetic traces from SFF data, as well as provide a pass-through access to SCF data for Sanger reads.

A

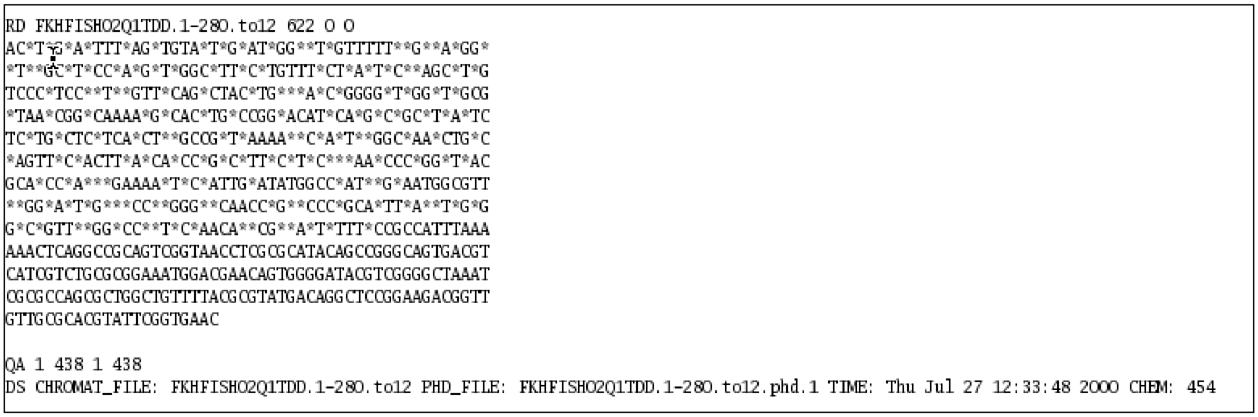


```
AS      15 690514

CO gi|85666109|ref|NC_001133.6| 226558 14877 1 U
CCACACCACACCCACACACCCACACACCACACCACACACCACACCACACC
CACACACACACATCCTAACACTACCCTAACACAGCCCTAATCTAACCC*T
GGCCAACCTGTCTCTCAAC*TTACCCTCCATTACCC*TGCCTCCACTCG*
TTACCCTGTCCCATTCAACCATACCACTCCGAACCACCATCCATCCCTCT
ACTTACTACCACTCACCCACCGTTACCCTCCAATTA*CCC*ATA*TCCAA
CCCACT*GCCAC*TTACCCTACCATTACCC*TACCATCCACCATGACCTA
CTCACCATACTGTTCTTCTACCCACCATATTGAAACGCTAACAAATGATC
GTAAATAACACACACGT*GCTT*ACCCTACCACTTT*ATACC*ACCACCA
```



```
AF FKHFISHO2RVIHK_right.366-293.to131 C 346
AF FKHFISHO2RCO1X_left.29-1.fm32.pr32 C -199
AF FKHFISHO2QZABD_left.1-3.to131.pr32 U 461
AF FKHFISHO2TF65J_left.246-301.fm2.pr3 U -244
AF FKHFISHO2P54QU_right.14-114.fm46.pr45 U -12
AF FKHFISHO2TBL3K_right.1-115.to131.pr124 U 280
AF FKHFISHO2PK3O9.211-478.fm32 U -209
AF FKHFISHO2QS2XI_right.96-1.fm63.pr62 C -14
AF FKHFISHO2RVVA3_right.pr2 C 246
AF FKHFISHO2SO6XK_left.348-356.fm66.pr67 U -346
AF FKHFISHO2PN9GI_right.243-279.fm66.pr67 U -241
AF FKHFISHO2SSH8E_left.1-280.to131 U 24
AF FKHFISHO2SVYVD_left.1-192.to131.pr32 U 169
AF FKHFISHO2PVOGQ_left.248-287.fm32.pr33 U -246

BS 1 196 FKHFISHO2PYT8Z.356-61.fm60.to12
BS 197 198 FKHFISHO2QYZGC_right.28-326.fm32.to131.pr33
BS 199 402 FKHFISHO2PYT8Z.356-61.fm60.to12
BS 403 404 FKHFISHO2QYZGC_right.28-326.fm32.to131.pr33
BS 405 462 FKHFISHO2PYT8Z.356-61.fm60.to12
BS 463 463 FKHFISHO2REOJT_right.1-296.to131.pr32
BS 464 464 FKHFISHO2PYT8Z.356-61.fm60.to12
```

C



```
RD FKHFISHO2Q1TDD.1-280.to12 622 0 0
AC*T*G*A*TTT*AG*TGTA*T*G*AT*GG**T*GTTTTT**G**A*GG*
*T**GC*T*CC*A*G*T*GGC*TT*C*TGTTT*CT*A*T*C**AGC*T*G
TCCC*TCC**T**GTT*CAG*CTAC*TG***A*C*GGGG*T*GG*T*GCG
*TAA*CGG*CAAAA*G*CAC*TG*CCGG*ACAT*CA*G*C*GC*T*A*TC
TC*TG*CTC*TCA*CT**GCCG*T*AAAA**C*A*T**GGC*AA*CTG*C
*AGTT*C*ACTT*A*CA*CC*G*C*TT*C*T*C***AA*CCC*GG*T*AC
GCA*CC*A***GAAAA*T*C*ATTG*ATATGGCC*AT**G*AATGGCGTT
**GG*A*T*G***CC**GGG**CAACC*G**CCC*GCA*TT*A**T*G*G
G*C*GTT**GG*CC**T*C*AACA**CG**A*T*TTT*CCGCCATTTAAA
AAACTCAGGCCGCAGTCGGTAACCTCGCGCATACAGCCGGGCAGTGACGT
CATCGTCTGCGCGGAAATGGACGAACAGTGGGGATACGTCGGGGCTAAAT
CGCGCCAGCGCTGGCTGTTTTACGCGTATGACAGGCTCCGGAAGACGGTT
GTTGCGCACGTATTCGGTGAAC

QA 1 438 1 438
DS CHROMAT_FILE: FKHFISHO2Q1TDD.1-280.to12 PHD_FILE: FKHFISHO2Q1TDD.1-280.to12.phd.1 TIME: Thu Jul 27 12:33:48 2000 CHEM: 454
```

Figure 108: 454Contigs.ace file portion portions example for a genomic DNA project with paired end reads. (A) Contig sequence and quality information; (B) Information mapping reads to contigs; (C) Read sequence and quality information.

The structure of the ACE file produced by the GS Reference Mapper application differs from the traditional assembly ACE files because it is displaying the sequence information within the context of the given reference sequence(s). Each individual sequence in the reference to which at least one read maps is output as a "contig" in the ACE file (so that the alignments of the 454 reads to each reference sequence can be viewed relative to the reference sequence bases in the "contig alignment viewers" that most third-party ACE file viewers contain). The 454 reads and FASTA/FASTQ reads, as well as the contigs generated by the GS Reference Mapper application, are output in the ACE file as "reads", again to organize all the data relative to the reference.

To ensure compatibility with ACE file viewers, as well as to assist in the post-analysis of the mapping, the identifiers for reads whose alignments are split and paired end reads (if present), are output with an additional suffix. Several suffixes may be added to the original read identifier:

- For 454 paired end reads, the two halves of the 454 read that constitute the sequences at the two ends of the original clone (from which the paired end read was generated) are marked by "_left" and "_right" suffixes.
- If only part of the trimmed read aligns in this contig (either because the read is only Partially Mapped, or the read is aligned with different sections in different contigs due to a structural variation, for example), the base position range of the region aligned in this contig is added, as in ".1-60" if the read has bases 1-60 aligned in the contig.

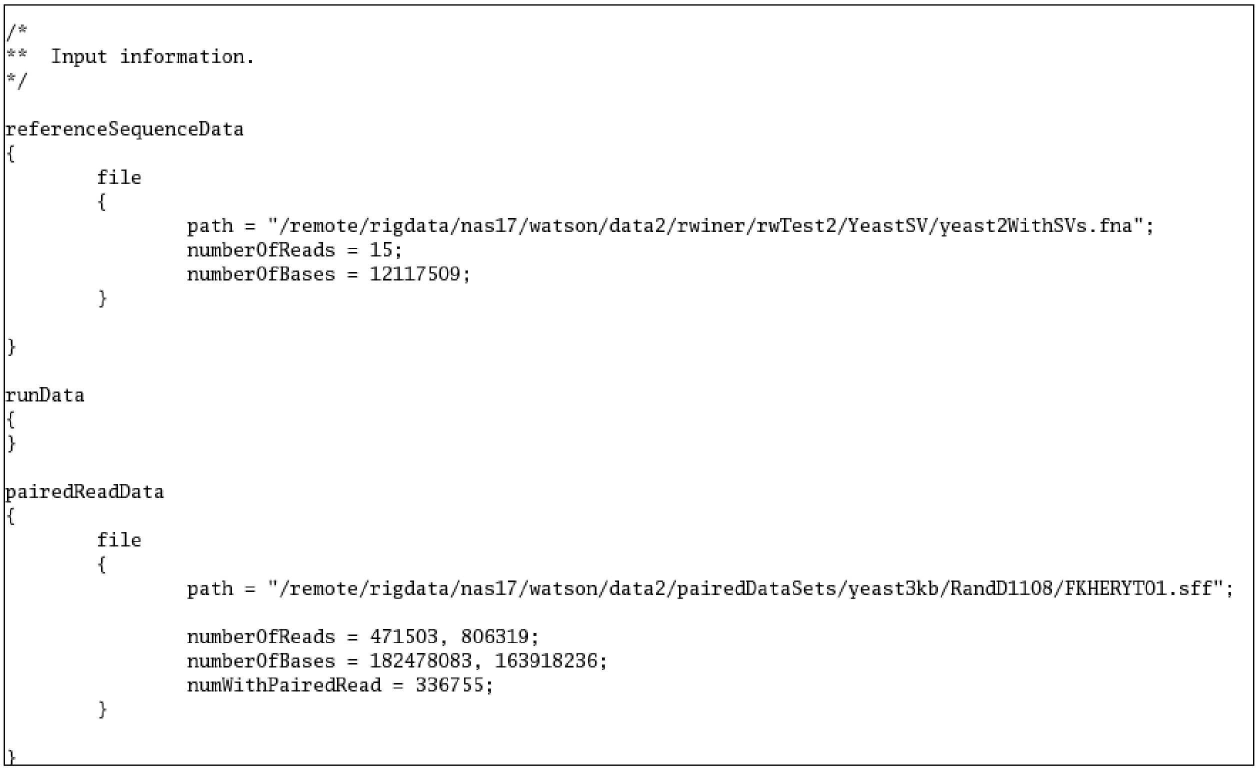
2.18.1.6 454NewblerMetrics.txt

The 454NewblerMetrics.txt file is a 454 parser (see the General Overview section of this manual for a description of the file formats) file that reports the key input, algorithmic and output metrics for the data analysis software applications. Each application that uses the Newbler algorithm creates a 454NewblerMetrics.txt file with relevant output. Below is a description of all the sections present in this file when produced by the GS Reference Mapper, with their named groups and keywords.

Percentage metrics are calculated in two distinct ways, with the first percentage value of each pair calculated based on the number of reads (or bases) used in the mapping computation, and the second percentage value calculated from the total number of physical reads (*i.e.* number of sequences in the read files).

2.18.1.6.1 Input information (Figure 109)

- **referenceSequenceData group** – contains information about the reference sequence file(s).
- **runData group** – contains information about the read data used in the analysis (both Sanger and non-Paired-End 454 Sequencing read files are reported on in this section; not shown since only paired end data files were used in this example).
- **pairedReadData group** – contains information about the paired end input data [paired end only; 454 Sequencing reads only (not Sanger reads)].



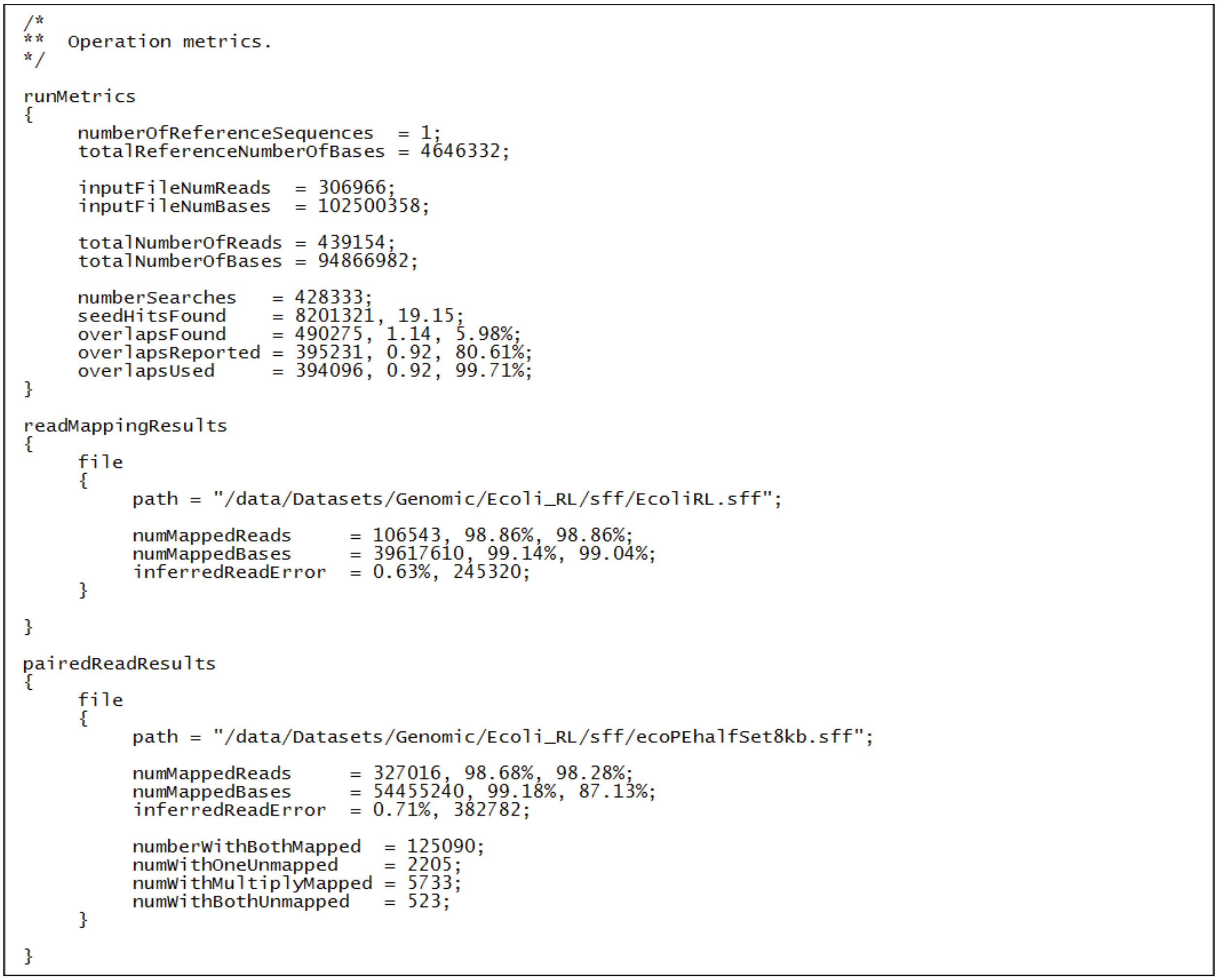
```
/*
**  Input information.
*/

referenceSequenceData
{
        file
        {
                path = "/remote/rigdata/nas17/watson/data2/rwiner/rwTest2/YeastSV/yeast2WithSVs.fna";
                numberOfReads = 15;
                numberOfBases = 12117509;
        }

}

runData
{
}

pairedReadData
{
        file
        {
                path = "/remote/rigdata/nas17/watson/data2/pairedDataSets/yeast3kb/RandD1108/FKHERYT01.sff";

                numberOfReads = 471503, 806319;
                numberOfBases = 182478083, 163918236;
                numWithPairedRead = 336755;
        }

}
```

Figure 109: 454NewblerMetrics file, referenceSequenceData and runData groups portion example.

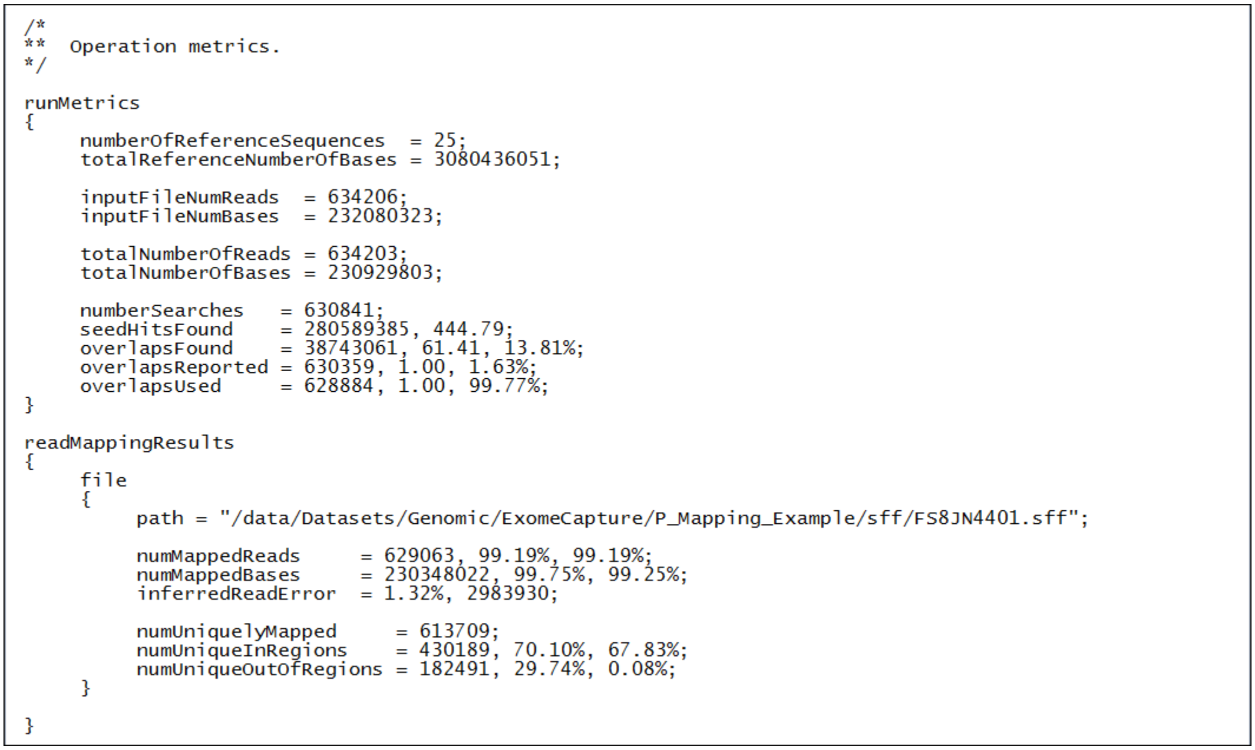
2.18.1.6.2 Operation metrics (Figure 110)

- **runMetrics group** – contains information about the mapping computation.
- **readMappingResults group** – contains information about the mapping process for each input file [SFF, FASTA/FASTQ (including Sanger paired end), or run regions from wells file; not shown on Figure 110 since only paired end data files were used in this example]. In the case of mapping performed with a region file, metrics are also provided for reads mapping uniquely in regions and out of regions. Any read whose mapping overlaps a region by at least one base will be included in NumUniqueInRegions. Other uniquely-mapping reads are included in NumUniqueOutOfRegions". The total of these two categories is reported as NumUniquelyMapped.
- **pairedReadResults group** – contains information about the paired end input data (paired end only; GS Junior and GS FLX+ systems) (Figure 110).

A



```
/*
**  Operation metrics.
*/

runMetrics
{
     numberOfReferenceSequences  = 1;
     totalReferenceNumberOfBases = 4646332;

     inputFileNumReads  = 306966;
     inputFileNumBases  = 102500358;

     totalNumberOfReads = 439154;
     totalNumberOfBases = 94866982;

     numberSearches   = 428333;
     seedHitsFound    = 8201321, 19.15;
     overlapsFound    = 490275, 1.14, 5.98%;
     overlapsReported = 395231, 0.92, 80.61%;
     overlapsUsed     = 394096, 0.92, 99.71%;
}

readMappingResults
{
     file
     {
          path = "/data/Datasets/Genomic/Ecoli_RL/sff/EcoliRL.sff";

          numMappedReads     = 106543, 98.86%, 98.86%;
          numMappedBases     = 39617610, 99.14%, 99.04%;
          inferredReadError  = 0.63%, 245320;
     }

}

pairedReadResults
{
     file
     {
          path = "/data/Datasets/Genomic/Ecoli_RL/sff/ecoPEhalfSet8kb.sff";

          numMappedReads     = 327016, 98.68%, 98.28%;
          numMappedBases     = 54455240, 99.18%, 87.13%;
          inferredReadError  = 0.71%, 382782;

          numberWithBothMapped   = 125090;
          numWithOneUnmapped     = 2205;
          numWithMultiplyMapped  = 5733;
          numWithBothUnmapped    = 523;
     }

}
```

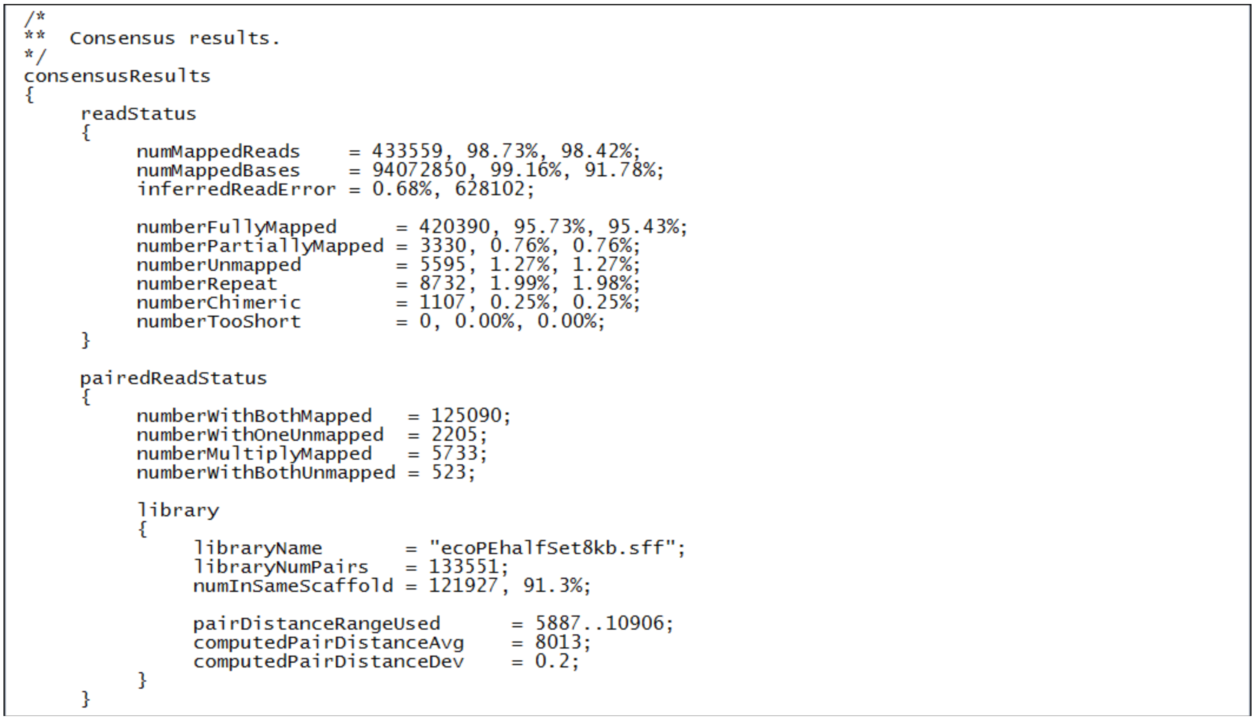


```
/*
**  Operation metrics.
*/

runMetrics
{
     numberOfReferenceSequences  = 25;
     totalReferenceNumberOfBases = 3080436051;

     inputFileNumReads  = 634206;
     inputFileNumBases  = 232080323;

     totalNumberOfReads = 634203;
     totalNumberOfBases = 230929803;

     numberSearches   = 630841;
     seedHitsFound    = 280589385, 444.79;
     overlapsFound    = 38743061, 61.41, 13.81%;
     overlapsReported = 630359, 1.00, 1.63%;
     overlapsUsed     = 628884, 1.00, 99.77%;
}

readMappingResults
{
     file
     {
          path = "/data/Datasets/Genomic/ExomeCapture/P_Mapping_Example/sff/FS8JN4401.sff";

          numMappedReads     = 629063, 99.19%, 99.19%;
          numMappedBases     = 230348022, 99.75%, 99.25%;
          inferredReadError  = 1.32%, 2983930;

          numUniquelyMapped     = 613709;
          numUniqueInRegions    = 430189, 70.10%, 67.83%;
          numUniqueOutOfRegions = 182491, 29.74%, 0.08%;
     }

}
```

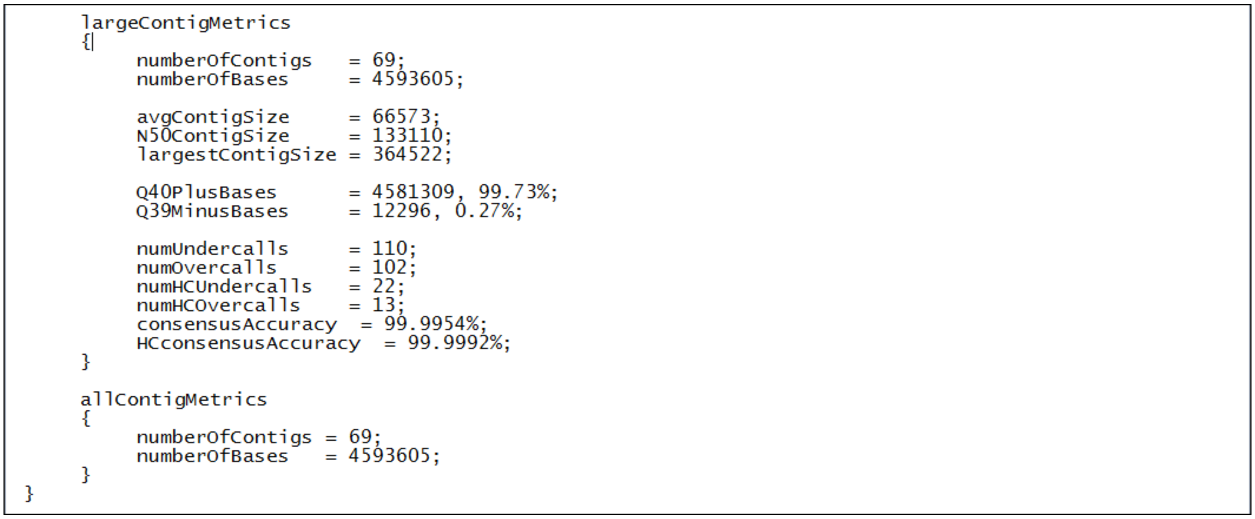
Figure 110: 454NewblerMetrics file, runMetrics and readMappingResults groups portion example. (A) With paired end data. (B) With a region file.

2.18.1.6.3 Consensus results (Figure 111)

- **consensusDistribution group** – contains information about the consensus signals and basecalling thresholds.
  - **fullDistribution**
  - **distributionPeaks**
  - **thresholdsUsed**
- **consensusResults group** – contains summary information and statistics about reads, scaffolds, and contigs.
  - **readStatus** – summary information about the reads.
  - **pairedReadStatus** – paired end library statistics (if paired end reads used).
  - **scaffoldMetrics** – scaffold statistics (if paired end reads are used).
  - **largeContigMetrics** – contig statistics for large contigs (longer than 'largeContigThreshold'; default is 500 bp).
  - **allContigMetrics**– contig statistics for all contigs (default is 100 bp).



```
/*
**  Consensus results.
*/
consensusResults
{
     readStatus
     {
          numMappedReads     = 433559, 98.73%, 98.42%;
          numMappedBases     = 94072850, 99.16%, 91.78%;
          inferredReadError = 0.68%, 628102;

          numberFullyMapped      = 420390, 95.73%, 95.43%;
          numberPartiallyMapped = 3330, 0.76%, 0.76%;
          numberUnmapped         = 5595, 1.27%, 1.27%;
          numberRepeat           = 8732, 1.99%, 1.98%;
          numberChimeric         = 1107, 0.25%, 0.25%;
          numberTooShort         = 0, 0.00%, 0.00%;
     }

     pairedReadStatus
     {
          numberWithBothMapped    = 125090;
          numberWithOneUnmapped   = 2205;
          numberMultiplyMapped    = 5733;
          numberWithBothUnmapped = 523;

          library
          {
               libraryName        = "ecoPEhalfSet8kb.sff";
               libraryNumPairs    = 133551;
               numInSameScaffold = 121927, 91.3%;

               pairDistanceRangeUsed      = 5887..10906;
               computedPairDistanceAvg    = 8013;
               computedPairDistanceDev    = 0.2;
          }
     }
```



```
      largeContigMetrics
      {|
           numberOfContigs   = 69;
           numberOfBases     = 4593605;

           avgContigSize     = 66573;
           N50ContigSize     = 133110;
           largestContigSize = 364522;

           Q40PlusBases      = 4581309, 99.73%;
           Q39MinusBases     = 12296, 0.27%;

           numUndercalls     = 110;
           numOvercalls      = 102;
           numHCUndercalls   = 22;
           numHCOvercalls    = 13;
           consensusAccuracy  = 99.9954%;
           HCconsensusAccuracy  = 99.9992%;
      }

      allContigMetrics
      {
           numberOfContigs = 69;
           numberOfBases   = 4593605;
      }
 }
```

Figure 111: 454NewblerMetrics file, consensusResults group example for a paired end genomic project.

The ConsensusResults section contains an entry line for chimeric reads. These are reads split into two or more segments, where the segments map to non-consecutive positions of the reference. A typical mapping of genomic DNA reads to a genomic DNA reference will have less than 1% chimeric reads. However, in some cases, such as when mapping cDNA reads to a genomic reference, the percentage of reads demonstrating this trait can be quite large. See Section 2.18.1.17 for details on the 454HCFusionVariantTable.txt and 454AllFusionVariantTable.txt report files.

2.18.1.7 454NewblerProgress.txt

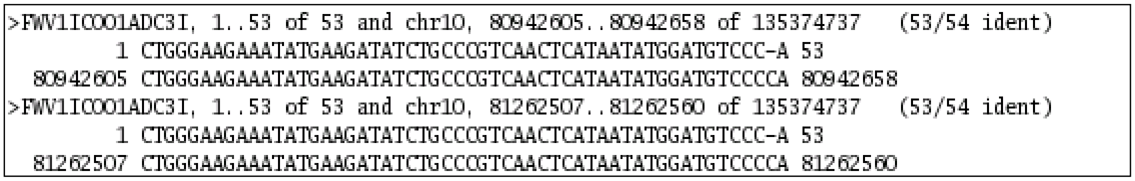
This file represents the text log of the messages sent to standard output by the runProject or runMapping command (showing the progress of the execution of the mapping computation). If runs are added incrementally and multiple executions of runProject occur, the output messages are appended to this file. If the “Incremental reference mapping analysis” checkbox option is not selected in the GUI application, or the “-r” option is given on the runProject command line, the GS Reference Mapper deletes intermediate data and “restarts” the mapping computation, and this file is deleted and restarted as well.

2.18.1.8 454PairAlign.txt

This file contains the pairwise alignments of the overlaps that were found during the mapping computation (Figure 112). By default, this file is not generated, but if the “-pair” or “-pairt” options are given on the runProject command line, it will be generated either in a human-readable text format (“-pair”) or in tab-delimited format (“-pairt”).

Each of the displayed alignments contains the following information (these are the columns in the tab-delimited format):

1. **QueryAccno** – accession number of the read used in the overlap detection search (the “query sequence”).
2. **QueryStart** – starting position of the alignment in query sequence.
3. **QueryEnd** – ending position of the alignment in query sequence.
4. **QueryLength** – length of the query sequence.
5. **SubjAccno** – accession number of the other read (the “subject sequence”).
6. **SubjStart** – starting position of the alignment in subject sequence.
7. **SubjEnd** – ending position of the alignment in subject sequence.
8. **SubjLength** – length of the subject sequence.
9. **NumIdent** – number of identities in the pairwise alignment, *i.e.* where query and subject characters match.
10. **AlignLength** – the length of the pairwise alignment.
11. **QueryAlign** – query alignment sequence.
12. **SubjAlign** – subject alignment sequence.



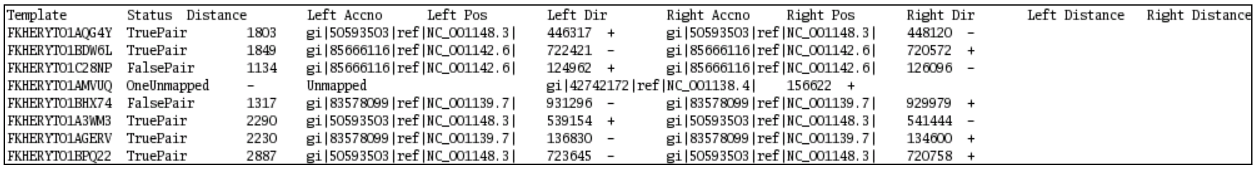
```
>FWV1ICO01ADC3I, 1..53 of 53 and chr10, 80942605..80942658 of 135374737   (53/54 ident)
        1 CTGGGAAGAAATATGAAGATATCTGCCCGTCAACTCATAATATGGATGTCCC-A 53
  80942605 CTGGGAAGAAATATGAAGATATCTGCCCGTCAACTCATAATATGGATGTCCCCA 80942658
>FWV1ICO01ADC3I, 1..53 of 53 and chr10, 81262507..81262560 of 135374737   (53/54 ident)
        1 CTGGGAAGAAATATGAAGATATCTGCCCGTCAACTCATAATATGGATGTCCC-A 53
  81262507 CTGGGAAGAAATATGAAGATATCTGCCCGTCAACTCATAATATGGATGTCCCCA 81262560
```

Figure 112: 454PairAlign.txt file portion example.

2.18.1.9 454PairStatus.txt

This file contains the per-pair report of the location and status of how each paired end pair of reads used in the mapping. Each line contains the following information (these are the columns in the tab-delimited format):

1. **Template** – template string for the pair (this will be the original 454 accession for 454 paired end reads, and the "template" string for Sanger reads).
2. **Status** – the status of the pair in the mapping, with the following possible values:
    a. **BothUnmapped** – both halves of the pair were unmapped.
    b. **OneUnmapped** – one of the reads in the pair was unmapped.
    c. **MultiplyMapped** – one or both of the reads in the pair were marked as Repeat.
    d. **TruePair** – both halves of the pair were mapped into the same reference sequence, with the correct relative orientation, and are within the expected distance of each other.
    e. **FalsePair** – the halves were mapped to the same reference sequence, but the orientation of their alignment is inconsistent with a paired end pair or the distance between the halves is outside the expected distance.
3. **Distance** – for "TruePair" or "FalsePair" pairs, the distance between the halves.
4. **Left Contig** – the contig where the left half was mapped, or "-" if the read was Unmapped or Repeat.
5. **Left Pos** – the position in the contig where the 5' end of the left half was mapped.
6. **Left Dir** – the direction ('+' for the forward strand of the reference sequence and '-' for reverse strand) in which the left half was mapped.
7. **Right Contig** – the contig where the right half was mapped, or "-" if the read was Unmapped or Repeat.
8. **Right Pos** – the position in the contig where the 3' end of the right half was mapped.
9. **Right Dir** – the direction ('+' for the forward strand of the reference sequence and '-' for reverse strand) in which the right half was mapped.
10. **Left Distance** – the distance from the Left Pos to the respective end of the reference sequence (for forward matches, this is the distance to the 3' end of the sequence; for reverse matches, to the 5' end).
11. **Right Distance** – the distance from the Right Pos to the respective end of the reference sequence (for forward matches, this is the distance to the 3' end of the sequence; for reverse matches, to the 5' end).



```
Template        Status  Distance        Left Accno      Left Pos        Left Dir        Right Accno     Right Pos       Right Dir       Left Distance   Right Distance
FKHERYTO1AQG4Y  TruePair        1803    gi|50593503|ref|NC_001148.3|    446317  +       gi|50593503|ref|NC_001148.3|    448120  -
FKHERYTO1BDW6L  TruePair        1849    gi|85666116|ref|NC_001142.6|    722421  -       gi|85666116|ref|NC_001142.6|    720572  +
FKHERYTO1C28NP  FalsePair       1134    gi|85666116|ref|NC_001142.6|    124962  +       gi|85666116|ref|NC_001142.6|    126096  -
FKHERYTO1AMVUQ  OneUnmapped     -       Unmapped                        gi|42742172|ref|NC_001138.4|    156622  +
FKHERYTO1BHX74  FalsePair       1317    gi|83578099|ref|NC_001139.7|    931296  -       gi|83578099|ref|NC_001139.7|    929979  +
FKHERYTO1A3WM3  TruePair        2290    gi|50593503|ref|NC_001148.3|    539154  +       gi|50593503|ref|NC_001148.3|    541444  -
FKHERYTO1AGERV  TruePair        2230    gi|83578099|ref|NC_001139.7|    136830  -       gi|83578099|ref|NC_001139.7|    134600  +
FKHERYTO1BPQ22  TruePair        2887    gi|50593503|ref|NC_001148.3|    723645  -       gi|50593503|ref|NC_001148.3|    720758  +
```

Figure 113: 454PairStatus.txt file portion example.

2.18.1.10 454TagPairAlign.txt

Reads shorter than 50 bp are rare in standard shotgun sequencing runs, and are not used to map to the reference using the standard overlap detection parameters (Section 2.7.2). Instead, these 'short tag' reads are mapped to the reference in a later step of the computation, using different overlap detection parameters that allow more efficient mapping of shorter sequences. The shortest read mapped in this way is controlled by the -minlen parameter (default 20 bp). Thus, the default length range reported as short tag for standard shotgun runs is 20-49 bp. The -short option will map all reads down to the sum of the seed length plus seed step (default 16 + 12 = 28 bp), which reduces the default range of short tag reads to 20-27 bp.

The most common short tag reads are derived by removing the linker sequence from paired end reads, generating the two half reads designated '_left' and '_right' (see Section 4.6). When paired end reads are detected, the behavior of the -short option is automatically implemented without requiring explicit use of this option.

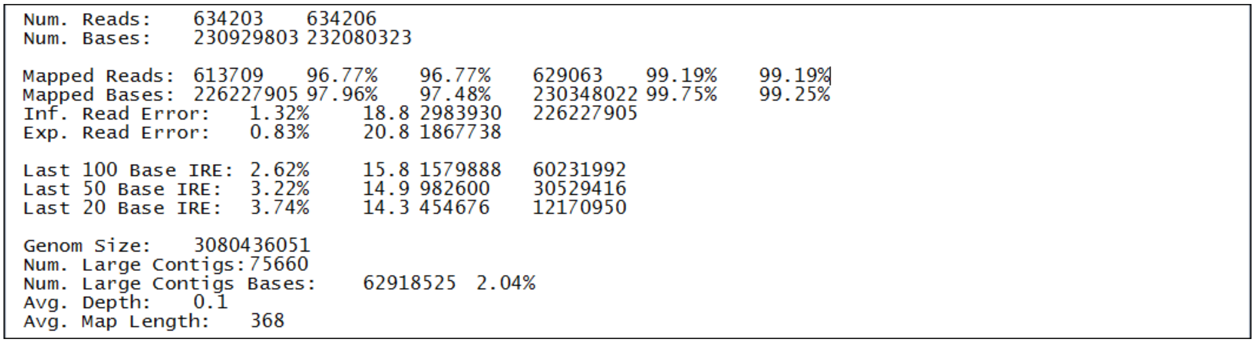
The 454TagPairAlign.txt file reports the alignments of uniquely-mapped short tag reads against the reference. This file is not generated by default, but output can be controlled from either the 'Pairwise alignment' option of the Output tab of the GUI or from the command line. It can be generated either in a human-readable text format ('Simple format', -pair) or in tab-delimited format ('Tabbed format', -pairt). The format of this file is identical to the 454PairAlign.txt file described in Section 2.18.1.8.

2.18.1.11 454MappingQC.xls

This file contains a number of detailed metrics regarding the mapping results; its format and structure are intended for reading by MS Excel or a similar spreadsheet program, so that the metrics can be easily visualized using Excel's charting/graphing tools. The file is in tab-delimited format, and contains six main sections (Note: The section titles given here are not displayed in the file.)

Percentage metrics are calculated in two distinct ways, with the first percentage value of each pair calculated based on the number of reads (or bases) used in the mapping computation, and the second percentage value calculated from the total number of physical reads (*i.e.* number of sequences in the read files).

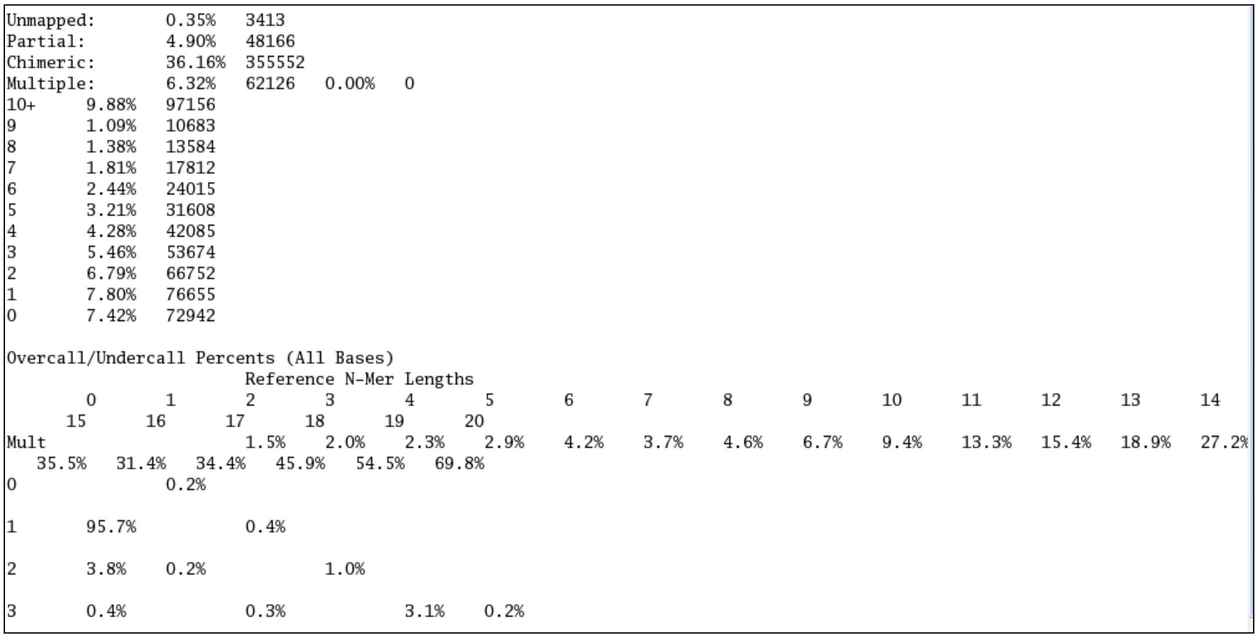
1. Summary Statistics (Figure 114)
    a. **Num. Reads** – the number of input reads used in the mapping computation followed by the number of physical reads.
    b. **Num. Bases** – the number of bases in the input reads followed by the number of bases in the physical reads.
    c. **Mapped Reads** – the number and percentage of reads that uniquely mapped to the reference, followed by the number and percentage of reads that uniquely or multiply mapped.
    d. **Mapped Bases** – the number and percentage of bases that uniquely mapped to the reference, followed by the number and percentage of reads that uniquely or multiply mapped.
    e. **Inf. Read Error** – the "inferred read error" percentage and quality score (calculated as the number of read alignment differences over the number of mapped bases), along with the counts of the number of read alignment differences and mapped bases.
    f. **Exp. Read Error** – the expected read error computed from the input read quality scores, given as a percentage, quality score and expected number of alignment differences. This is computed by summing the expected number of errors for each quality score value (*i.e.* number of bases with a quality score times the accuracy rate of that quality score).
    g. **Last 100 Base IRE** – the "inferred read error" numbers, using only the last (3') 100 bases of each read.
    h. **Last 50 Base IRE** – the "inferred read error" numbers, using only the last (3') 50 bases of each read.
    i. **Last 20 Base IRE** – the "inferred read error" numbers, using only the last (3') 20 bases of each read.
    j. **Genome Size** – the number of bases in the reference.
    k. **Num. Large Contigs** – the number of large contigs reported in the 454LargeContigs.fna file.
    l. **Num. Large Contig Bases** – the number of bases in the large contigs.
    m. **Avg. Depth** – the average alignment depth (*i.e.* how many reads aligned to each position of the reference).
    n. **Avg. Map Length** – the average length of the alignment of a read (the read's "map length").



```
Num. Reads:   634203    634206
Num. Bases:   230929803 232080323

Mapped Reads: 613709    96.77%    96.77%   629063    99.19%   99.19%
Mapped Bases: 226227905 97.96%    97.48%   230348022 99.75%   99.25%
Inf. Read Error:   1.32%     18.8 2983930  226227905
Exp. Read Error:   0.83%     20.8 1867738

Last 100 Base IRE: 2.62%     15.8 1579888  60231992
Last 50 Base IRE:  3.22%     14.9 982600   30529416
Last 20 Base IRE:  3.74%     14.3 454676   12170950

Genom Size:   3080436051
Num. Large Contigs:75660
Num. Large Contigs Bases:    62918525  2.04%
Avg. Depth:   0.1
Avg. Map Length:   368
```

Figure 114: 454MappingQC, summary statistics portion.

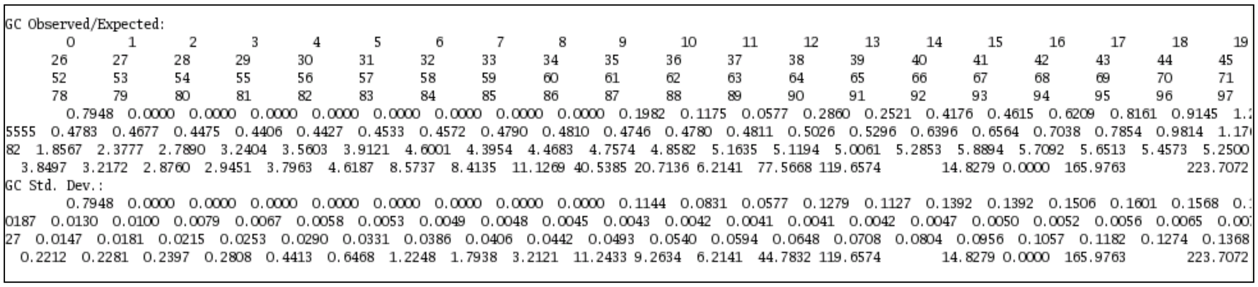
2. Read Error Histogram (Figure 115)
    a. This section breaks down the number of reads based on their read status and/or number of alignment differences, and displays percentages and counts per category.
    b. Reads that did not map to the reference ("Unmapped"), reads where only part of the sequence mapped to the reference ("Partial") and reads that mapped to multiple locations in the reference ("Multiple") are displayed as one category each.
    c. Reads that mapped fully to the reference (meaning every base of the read occurred in the alignment) are then divided by the number of alignment differences found, and percentages and number of reads having 0, 1, 2, …, 9 or 10+ (10 or more) errors are shown.
3. Overcall/Undercall Tables (Figure 115)
    a. These two tables display the percentages and numbers of homopolymer accuracy, using the read alignments to count when the read contains the same or different homopolymer length as compared to the reference.
    b. The rows are the read homopolymer length and the columns are the reference homopolymer length.
    c. When scanning the read alignments, the reference sequence homopolymer is first determined, then the alignment of the read to that homopolymer is evaluated and counted.
        i. Some reads may have multiple homopolymers aligned to a single reference homopolymer (like "ACA" align to "AAA"). These are counted and displayed separately on a "Mult" row.
        ii. The percentages shown in the first overcall/undercall table are given as a percentage of the column (*e.g.* what percent of the time at a reference 5-mers did the read have a 4-mer). Also, the percent table does not show the percentage of the correct alignments (*e.g.* 5-mer to 5-mer), nor does it show percentages less than 0.1% (in order to highlight the overcall/undercall trend).
        iii. Below the counts table, a "%Ident" row displays the percentages for each reference n-mer where the read was called correctly (*i.e.* its homopolymer length matched the reference).



```
Unmapped:       0.35%   3413
Partial:        4.90%   48166
Chimeric:       36.16%  355552
Multiple:       6.32%   62126   0.00%   0
10+     9.88%   97156
9       1.09%   10683
8       1.38%   13584
7       1.81%   17812
6       2.44%   24015
5       3.21%   31608
4       4.28%   42085
3       5.46%   53674
2       6.79%   66752
1       7.80%   76655
0       7.42%   72942

Overcall/Undercall Percents (All Bases)
                        Reference N-Mer Lengths
        0       1       2       3       4       5       6       7       8       9       10      11      12      13      14
      15      16      17      18      19      20
Mult                    1.5%    2.0%    2.3%    2.9%    4.2%    3.7%    4.6%    6.7%    9.4%    13.3%   15.4%   18.9%   27.2%
   35.5%   31.4%   34.4%   45.9%   54.5%   69.8%
0               0.2%

1       95.7%           0.4%

2       3.8%    0.2%            1.0%

3       0.4%            0.3%            3.1%    0.2%
```

Figure 115: 454MappingQC, Read Error Histogram and Overcall/Undercall Tables portion.

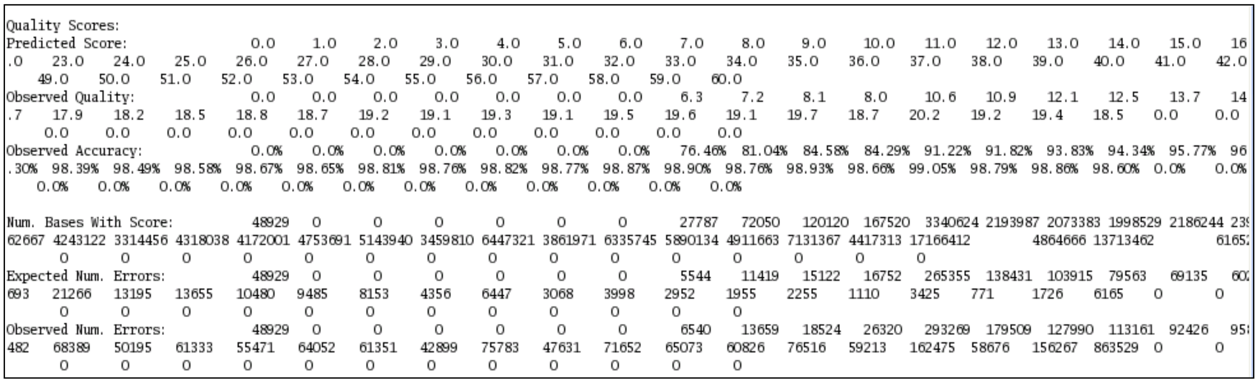
4. GC Content Information (Figure 116)
   a. **GC Observed/Expected** – the two lines below this display the GC content percentages (from 0 to 100) and the observed over expected mapping depth. This is calculated by first counting the number of reads with particular GC content and counting the GC content of all windows of the reference (where the window length is one half the average read length). Then the two counts (read and reference) for a specific GC content value are divided by the read/reference totals to compute the percentage of the reads/references with that GC content. The observed/expected value is the ratio of those two percentages.
   b. **GC Std. Dev.** – this is the standard deviation of the GC Observed/Expected (based on the sampling at that GC content value). The values on this line are useful for setting the "Y Error Bars" information in Excel, if an "XY (Scatter)" chart is made using the GC Observed/Expected two lines as the source data. This line can then be used as the "+" and "-" data of the "Custom" Error amount, found inside the "Y Error Bars" tab of the "Format Data Series" dialog box).



```
GC Observed/Expected:
        0       1       2       3       4       5       6       7       8       9       10      11      12      13      14      15      16      17      18      19
       26      27      28      29      30      31      32      33      34      35      36      37      38      39      40      41      42      43      44      45
       52      53      54      55      56      57      58      59      60      61      62      63      64      65      66      67      68      69      70      71
       78      79      80      81      82      83      84      85      86      87      88      89      90      91      92      93      94      95      96      97
          0.7948  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.1982  0.1175  0.0577  0.2860  0.2521  0.4176  0.4615  0.6209  0.8161  0.9145  1.2
5555  0.4783  0.4677  0.4475  0.4406  0.4427  0.4533  0.4572  0.4790  0.4810  0.4746  0.4780  0.4811  0.5026  0.5296  0.6396  0.6564  0.7038  0.7854  0.9814  1.17
82  1.8567  2.3777  2.7890  3.2404  3.5603  3.9121  4.6001  4.3954  4.4683  4.7574  4.8582  5.1635  5.1194  5.0061  5.2853  5.8894  5.7092  5.6513  5.4573  5.2500
  3.8497  3.2172  2.8760  2.9451  3.7963  4.6187  8.5737  8.4135  11.1269 40.5385 20.7136 6.2141  77.5668 119.6574        14.8279 0.0000  165.9763        223.7072
GC Std. Dev.:
          0.7948  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.1144  0.0831  0.0577  0.1279  0.1127  0.1392  0.1392  0.1506  0.1601  0.1568  0.1
0187  0.0130  0.0100  0.0079  0.0067  0.0058  0.0053  0.0049  0.0048  0.0045  0.0043  0.0042  0.0041  0.0041  0.0042  0.0047  0.0050  0.0052  0.0056  0.0065  0.00
27  0.0147  0.0181  0.0215  0.0253  0.0290  0.0331  0.0386  0.0406  0.0442  0.0493  0.0540  0.0594  0.0648  0.0708  0.0804  0.0956  0.1057  0.1182  0.1274  0.1368
  0.2212  0.2281  0.2397  0.2808  0.4413  0.6468  1.2248  1.7938  3.2121  11.2433 9.2634  6.2141  44.7832 119.6574        14.8279 0.0000  165.9763        223.7072
```

Figure 116: 454MappingQC, GC content information portion.

5. Quality Score Information (Figure 117)
   a. **Predicted Score** – the quality score values, from 0 to 60.
   b. **Observed Quality** – the observed quality score obtained from the read alignments (computed as "Observed Num. Errors" over "Num. Bases With Score" values, see below).
   c. **Observed Accuracy** – the observed quality score expressed as an accuracy percentage.
   d. **Num. Bases With Score** – the number of mapped bases having the Predicted Score (only mapped bases are used, because they can be evaluated for accuracy).
   e. **Expected Num. Errors** – the expected number of errors for a quality score, given the number of mapped bases with that quality score.
   f. **Observed Num. Errors** – the number of bases which did not match in the read alignment (*i.e.* the alignment column containing that base was not an identity).



```
Quality Scores:
Predicted Score:              0.0     1.0     2.0     3.0     4.0     5.0     6.0     7.0     8.0     9.0     10.0    11.0    12.0    13.0    14.0    15.0    16
.0    23.0    24.0    25.0    26.0    27.0    28.0    29.0    30.0    31.0    32.0    33.0    34.0    35.0    36.0    37.0    38.0    39.0    40.0    41.0    42.0
   49.0    50.0    51.0    52.0    53.0    54.0    55.0    56.0    57.0    58.0    59.0    60.0
Observed Quality:             0.0     0.0     0.0     0.0     0.0     0.0     0.0     6.3     7.2     8.1     8.0     10.6    10.9    12.1    12.5    13.7    14
.7    17.9    18.2    18.5    18.8    18.7    19.2    19.1    19.3    19.1    19.5    19.6    19.1    19.7    18.7    20.2    19.2    19.4    18.5    0.0     0.0
    0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0
Observed Accuracy:            0.0%    0.0%    0.0%    0.0%    0.0%    0.0%    0.0%    76.46%  81.04%  84.58%  84.29%  91.22%  91.82%  93.83%  94.34%  95.77%  96
.30%  98.39%  98.49%  98.58%  98.67%  98.65%  98.81%  98.76%  98.82%  98.77%  98.87%  98.90%  98.76%  98.93%  98.66%  99.05%  98.79%  98.86%  98.60%  0.0%    0.0%
   0.0%    0.0%    0.0%    0.0%    0.0%    0.0%    0.0%    0.0%    0.0%    0.0%    0.0%    0.0%

Num. Bases With Score:        48929   0       0       0       0       0       0       27787   72050   120120  167520  3340624 2193987 2073383 1998529 2186244 23
62667 4243122 3314456 4318038 4172001 4753691 5143940 3459810 6447321 3861971 6335745 5890134 4911663 7131367 4417313 17166412        4864666 13713462        6165
      0       0       0       0       0       0       0       0       0       0       0       0       0       0       0
Expected Num. Errors:         48929   0       0       0       0       0       0       5544    11419   15122   16752   265355  138431  103915  79563   69135   60
693   21266   13195   13655   10480   9485    8153    4356    6447    3068    3998    2952    1955    2255    1110    3425    771     1726    6165    0       0
      0       0       0       0       0       0       0       0       0       0       0       0
Observed Num. Errors:         48929   0       0       0       0       0       0       6540    13659   18524   26320   293269  179509  127990  113161  92426   95
482   68389   50195   61333   55471   64052   61351   42899   75783   47631   71652   65073   60826   76516   59213   162475  58676   156267  863529  0       0
      0       0       0       0       0       0       0       0       0       0       0       0
```

Figure 117: 454MappingQC, Quality score information portion.

6. Histogram/Window-based Information. This final section contains the data for larger histograms and data sets, and is organized column-wise, instead of the row-wise layout of the previous sections. It contains three sub-sections:
   a. **Read Length and Mapped Length Histograms** – histograms showing the number of reads of each read length (the "Read Length Histogram" column), the number of reads at each length of the aligned regions per the reference sequence, *i.e.* counting only the read bases in the alignment, not the alignment length (the "Mapped Length Histogram" column), and the depth or count of reads at each position ("Read Count Histogram).
   b. **Errors by Base Position** – plot values showing position-by-position errors in the reads, *i.e.*, the proportion of errors at the N'th base across all the reads. This section displays the 'Errors by Base Position' (% error rate, observed & predicted quality) and the 'Cumul. Errors by Base Position' (cumulative % error rate and cumulative quality).
   c. **Note:** if an alignment column contains a gap in the read, that is counted as an error at the previous base position (*i.e.*, any alignment gaps between base 5 and 6 in a read are counted as errors at position 5).
   d. Cross-Reference Depth and GC Information.

      The last six columns of this section contain region-by-region statistics of the alignments across the reference, where the reference is evenly divided into 1000 regions.

      Important Note: This division of the reference into regions has no understanding of repeat regions, and simply reports on the alignments of the uniquely mapping reads. Since repeat reads are not aligned to the reference, the values in this column will count repeat regions as unaligned regions.

      i. The first column displays the **position** in the reference sequence at the center of the region.
      ii. **Avg. Depth** – the average alignment depth in the region.
      iii. **Min. Depth** – the minimum alignment depth in the region.
      iv. **Max. Depth** – the maximum alignment depth in the region.
      v. **Depth Score** – a score that is indicative of the shallowness of the alignment in the region. Each alignment column in the region is given a score of "max(0, 4-depth)" where "depth" is the alignment depth of the column. The Depth Score for a region is the sum of the column scores. This score is a very sensitive metric for use in resequencing projects, in order to gauge when enough sequencing has been performed (and the addition of more reads will not fill in any more unaligned or shallowly aligned regions of the reference).
      vi. **GC** – the average GC content of the region.

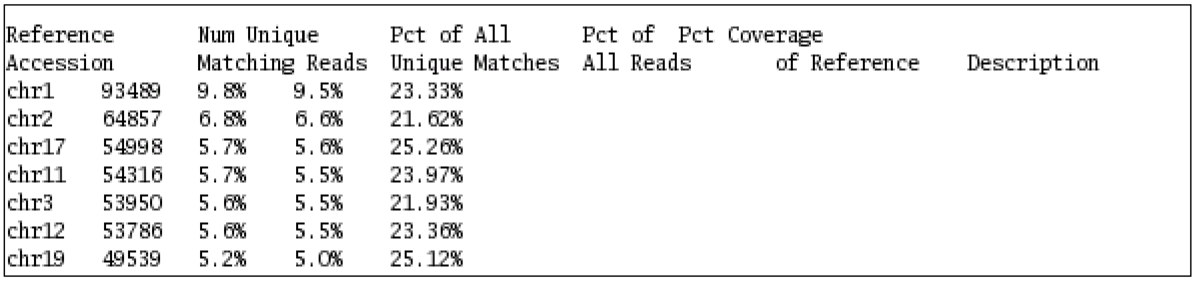
```
                                Errors by Base Position         Cumul. Errors by Base Position
                                                                                                Read
Length Histogram        Mapped Length Histogram         Read Count Histogram            % Error Rate
        Obs. Quality    Pred. Quality   Avg. Quality    Cum. % Error Rate       Cum. Quality
        Flow Base       Per-Flow Errors         Per-Flow Error Rate     Per-Flow Quality
        Accno   Position        Avg. Unique Depth       Avg. Align Depth        Avg. Total Depth
        Min. Align Depth        Max. Align Depth        Depth Score     GC Content
1       0       0       134779          0.20%   27                      0.20%   27
                                gi|49175990|ref|NC_000913.2|    773     9.9     10.2    10.2    4
        16      0       52.1
2       0       0       134780          0.09%   30.5                    0.15%   28.4
                                gi|49175990|ref|NC_000913.2|    2320    13.5    15.8    15.8    10
        22      0       51.8
3       0       0       134781          0.13%   28.7    30.6    38.1    0.14%   28.5
                                gi|49175990|ref|NC_000913.2|    3867    14.5    15.2    15.2    5
        23      0       54.6
4       0       0       134781          0.17%   27.6    29.5    37.6    0.15%   28.2
                                gi|49175990|ref|NC_000913.2|    5414    17.1    19      19      13
        24      0       49.5
5       0       0       134783          0.17%   27.6    28.7    37.3    0.15%   28.1
                                gi|49175990|ref|NC_000913.2|    6961    10.7    11.4    11.4    6
        17      0       53.4
```

Figure 118: 454MappingQC, Histogram/Window-based Information portion.

2.18.1.12 454RefStatus.txt

This file gives statistical information regarding the number of reads that mapped to each reference sequence. It has six fields:

1. **Reference Accession** – accession number of a reference sequence
2. **Num Unique Matching Reads** – how many reads mapped uniquely to the reference. To be considered unique, a given portion of a read may only map to a single reference location. If a portion of a read maps to multiple reference locations (or multiple transcript variants of the same gene in the case of cDNA mapping projects), the read is considered to be a repeat..
3. **Pct of All Unique Matches** – the number of reads mapping uniquely to an individual reference sequence divided by the total number of reads that mapped uniquely to any reference sequence.
4. **Pct of All Reads** - number of reads that mapped uniquely to this reference divided by the total number of reads in this mapping project.
5. **Pct Coverage of Reference** – number of reference bases covered by at least one uniquely mapping read divided by the total number of bases in this reference.
6. **Description** – reference description obtained from the renaming file or annotation files.



```
Reference       Num Unique      Pct of All      Pct of  Pct Coverage
Accession       Matching Reads  Unique Matches  All Reads       of Reference    Description
chr1    93489   9.8%    9.5%    23.33%
chr2    64857   6.8%    6.6%    21.62%
chr17   54998   5.7%    5.6%    25.26%
chr11   54316   5.7%    5.5%    23.97%
chr3    53950   5.6%    5.5%    21.93%
chr12   53786   5.6%    5.5%    23.36%
chr19   49539   5.2%    5.0%    25.12%
```

Figure 119: 454RefStatus file example.

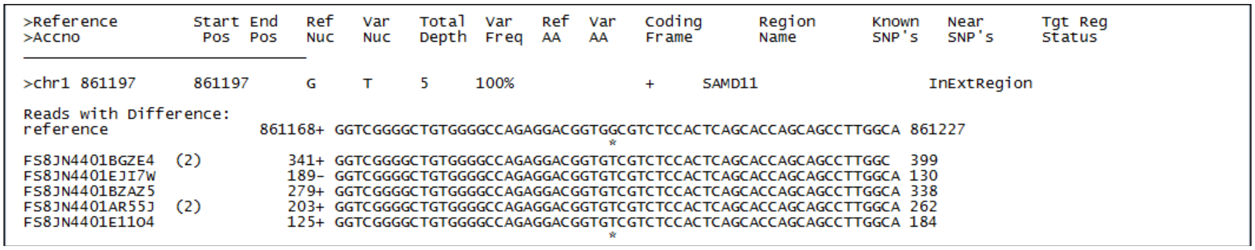
2.18.1.13 454AllDiffs.txt

This file contains the list of variations where at least 2 reads differ either from the reference sequence or from other reads aligned at a specific location. SNPs, insertion-deletion pairs, multi-homopolymer insertion or deletion regions, and single-base overcalls and undercalls are reported. Also, in order for a difference to be identified and reported, there must be at least two non-duplicate reads that (1) show the difference, (2) have at least 5 bases on both sides of the difference, and (3) have few other isolated sequence differences in the read. In addition, if the -e option is used to set an expected depth, then there must be at least 5% of that depth in differing reads. Finally, for single-base overcalls or undercalls to be reported, they must have a flow signal distribution that differs from the signal distribution of the reads matching the reference (*i.e.*, not all overcalls and undercalls are reported as variations). Once the difference is identified, all reads that fully span the difference location and have at least 5 additional flanking nucleotides on both sides are used in reporting the difference. For a sample 454AllDiffs.txt file, see Figure 120.

The file consists of one tab-delimited summary line for each difference, plus a multiple alignment of the reads spanning the difference location. Each of the difference summary lines begin with a '>' character, so the summary list of differences can be extracted from the file using the command "fgrep '>' 454AllDiffs.txt". The full summary lines contain up to eighteen columns of information, of which the first seven columns are always output, columns eight through thirteen are output if gene annotations or known SNP information is given to the GS Reference Mapper (or the **–fd** option is given), columns fourteen through seventeen are output only if the **–fd** option is given to the GS Reference Mapper, and column eighteen is output only if the **–reg** option is given. The summary lines contain the following columns when both **–fd** and **–reg** are given:

1. **Reference Accno** - the accession number of the reference sequence in which the difference was detected.
2. **Start Pos** - the start position within the reference sequence, where the difference occurs.
3. **End Pos** - the end position within the reference sequence, where the difference occurs.
4. **Ref Nuc** - the reference nucleotide sequence at the difference location.
5. **Var Nuc** - the differing nucleotide sequence at the difference location.
6. **Total Depth** - the total number of reads that fully span the difference location.
7. **Var Freq** - the percentage of different reads versus total reads that fully span the difference location.
8. **Ref AA** - the reference amino acid sequence at the difference location, if it occurs within the coding region of an annotated gene.
9. **Var AA** - the differing amino acid sequence at the difference location, if it occurs within the coding region of an annotated gene.
10. **Coding Frame** - the reading frame {-3, -2, -1, +1, +2, +3} if the difference occurs within the coding region of an annotated gene, or only the orientation {'+', '-'} if the difference occurs within a non-coding region of the gene (the column remains empty if the difference lies outside of a known gene).
11. **Region name** - the gene name at the difference location, if it occurs within the region of an annotated gene.
12. **Known SNP's** - a SNP ID from the dbSNP (database of know/verified SNPs and indels) that matches the location (within two bases), the reference base(s), and the variant base(s) of an observed difference.
13. **Near SNP's** - a SNP ID from the dbSNP (database of know/verified SNPs and indels) that is near (within ten bases of) the location of an observed difference, regardless of what substitution is observed.
14. **# Fwd w/ Var** (with **–fd** option only) - the number of forward reads that include the difference.
15. **# Rev w/ Var** (with **–fd** option only) - the number of reverse reads that include the difference.
16. **# Fwd Total** (with **–fd** option only) - the total number of forward reads that fully span the difference location.
17. **# Rev Total** (with **–fd** option only) - the total number of reverse reads that fully span the difference location.
18. **Tgt Region Status** (with –**reg** option only) – identifies each difference as "InRegion" if they map within the target regions or "InExtRegion" if they map in the extended target region, but not in the target regions (see Section 4.13.10 for more details).

In version 2.7 and later, the gsMapper application verifies that a Known SNP has an identical base substitution, although it might differ slightly in location.



```
>Reference        Start End   Ref   Var   Total  Var   Ref  Var   Coding     Region     Known   Near     Tgt Reg
>Accno             Pos  Pos   Nuc   Nuc   Depth  Freq  AA   AA    Frame      Name       SNP's   SNP's    Status
______________________________

>chr1 861197      861197      G     T     5      100%             +      SAMD11                 InExtRegion

Reads with Difference:
reference                861168+ GGTCGGGGCTGTGGGGCCAGAGGACGGTGGCGTCTCCACTCAGCACCAGCAGCCTTGGCA 861227
                                                              *
FS8JN4401BGZE4  (2)         341+ GGTCGGGGCTGTGGGGCCAGAGGACGGTGTCGTCTCCACTCAGCACCAGCAGCCTTGGC  399
FS8JN4401EJI7W              189- GGTCGGGGCTGTGGGGCCAGAGGACGGTGTCGTCTCCACTCAGCACCAGCAGCCTTGGCA 130
FS8JN4401BZAZ5              279+ GGTCGGGGCTGTGGGGCCAGAGGACGGTGTCGTCTCCACTCAGCACCAGCAGCCTTGGCA 338
FS8JN4401AR55J  (2)         203+ GGTCGGGGCTGTGGGGCCAGAGGACGGTGTCGTCTCCACTCAGCACCAGCAGCCTTGGCA 262
FS8JN4401E1104              125+ GGTCGGGGCTGTGGGGCCAGAGGACGGTGTCGTCTCCACTCAGCACCAGCAGCCTTGGCA 184
                                                              *
```

Figure 120: 454AllDiffs.txt portion example.

The multiple alignment for a difference shows the difference location plus approximately 30 bases on either side. The alignment is divided into two sections: the first section displays the reads found to contain the difference and the second, the reads not found to have the difference. Each line of the alignment displays the following information for a read:

1. The identifier for the read.
2. The number of duplicate reads whose alignment matches this read (when there are duplicates of the read).
3. The position of the read's first base displayed in the alignment region.
4. The orientation of the read in the displayed alignment ("+" is forward orientation, "-" is reverse-complement orientation, relative to the reference).
5. The aligned bases of the read.
6. The position of the read's last base displayed in the alignment region.

If the difference is a homopolymer undercall or overcall, the alignment rows are followed by a histogram depicting the signal distribution for the homopolymer

2.18.1.14 454HCDiffs.txt

This file contains the same type of information as the 454AllDiffs.txt file (section 2.18.1.13, above), but restricted to the "High-Confidence" differences. The GS Reference Mapper application uses a combination of flow signal information, quality score information and difference type information to determine if a difference is High-Confidence. The general rules are:

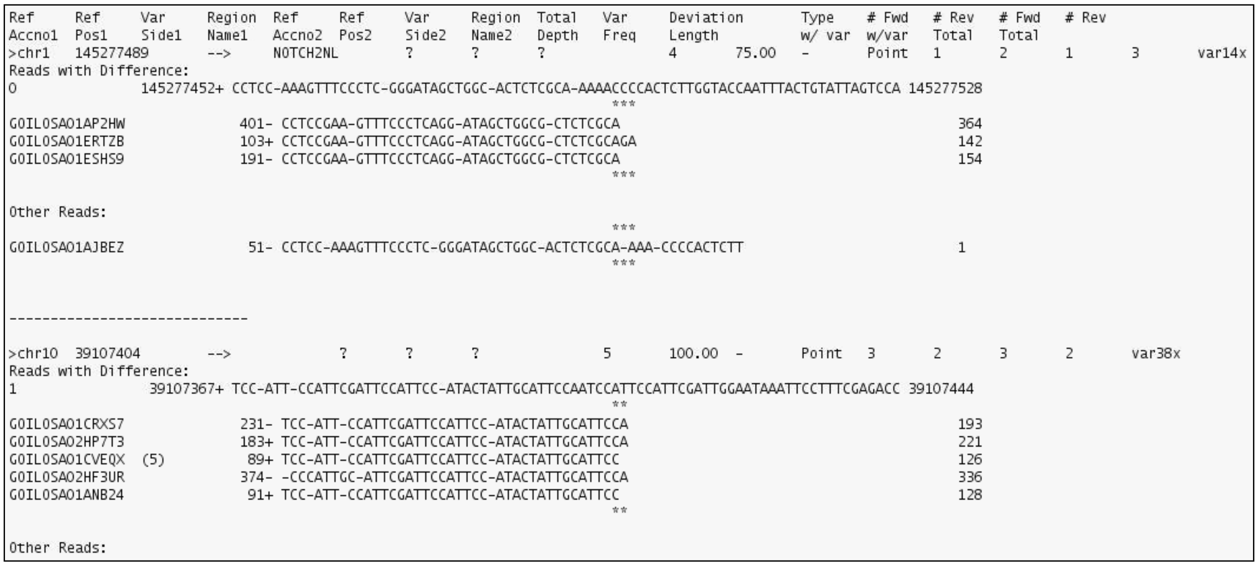
- There must be at least 3 non-duplicate reads with the difference, unless the -e option is specified, in which case at least 10% of the expected depth must have the difference
- There must be both forward and reverse reads showing the difference, unless there are at least 7 reads with quality scores over 20 (or 30 if the difference involves a 5-mer or higher)
- If the difference is a single-base overcall or undercall, then the reads with the difference must form the consensus of the sequenced reads (*i.e.*, at that location, the overall consensus must differ from the reference) and the signal distribution of the differing reads must vary from the matching reads (and the number of bases in that homopolymer of the reference).

2.18.1.15 454HCStructVars.txt and 454AllStructVars.txt

These two files show rearrangement points and rearrangement regions observed in the reads of the data set analyzed, relative to the reference. The structure of the two files is the same, but 454AllStructVars.txt uses a lower threshold for the frequency or variant reads relative to non-variant reads than 454HCStructVars.txt for reporting purposes.

The columns found on the summary line for each variation are described below. Some columns (the Region Name columns) require that annotations be supplied to the project. Some other columns are output only if the "**–fd"** option is specified.

1. **Ref Accno1** – the accession number of the reference sequence on one side of the variation.
2. **Ref Pos1** – the reference position on one side of the variation.
3. **Var Side1** – a direction arrow "-->" or "<--" describing the direction of the variation on the reference (*e.g.*, the 3' end of the reads or paired-end clones that diverge from the reference occur in this direction).
4. **Region Name1** – the gene name, or annotated region name, covering the location in the reference denoted by Ref Accno1 and Ref Pos1 (Gene or Region annotation must be included in the project for this field to contain a value).
5. **Ref Accno2** – the accession number of the reference sequence on the other side of the variation, if known. (If only one side of the variation is known, a question mark is given here).
6. **Ref Pos2** – the reference position on the other side of the variation, or a question mark if only one side is known.
7. **Var Side2** – a direction arrow "-->" or "<--" for the direction on the other side of the variation, or a question mark if only one side is known.
8. **Region Name2** – the gene name, or annotated region name, covering the location in the reference denoted by Ref Accno2 and Ref Pos2 (Gene or Region annotation must be included in the project for this field to contain a value).
9. **Total Depth** – the number of reads (for rearrangement points) or pairs (for rearrangement regions) covering the variation location(s).
10. **Var Freq** – the percentage of the reads/pairs that support the variation.
11. **Deviation Length** – if both sides of the variation occur on the same reference, this is the distance between the two variation locations.
12. **Type** – the string "Point" or "Region" to denote whether the rearrangement is a rearrangement point identified by split-read alignments or a rearrangement region identified by paired-end reads.
13. **# Fwd w/ var** – number of reads on the forward orientation that contain the variation (requires -**fd** option).
14. **# Rev w/ var** – number of reads on the reverse orientation that contain the variation (requires -**fd** option).
15. **# Fwd Total** – total number of reads in the forward orientation that map to this area of the reference (requires -**fd** option).
16. **# Rev Total** – total number of reads in the reverse orientation that map to this area of the reference (requires -**fd** option).
17. **Var ID** (no heading in file) – a field that identifies each individual variation, in the format "Var#x", where # is the ID number.



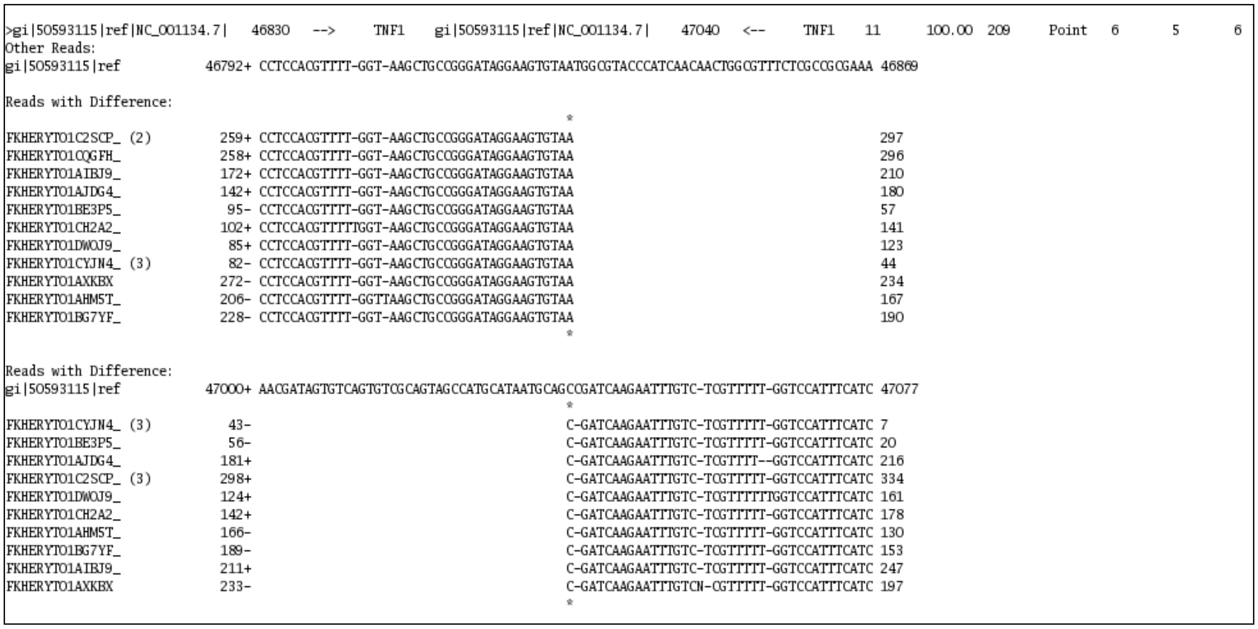
```
Ref     Ref     Var     Region  Ref     Ref     Var     Region  Total   Var     Deviation       Type    # Fwd   # Rev   # Fwd   # Rev
Accno1  Pos1    Side1   Name1   Accno2  Pos2    Side2   Name2   Depth   Freq    Length          w/ var  w/var   Total   Total
>chr1   145277489       -->     NOTCH2NL        ?       ?       ?               4       75.00   -       Point   1       2       1       3       var14x
Reads with Difference:
0               145277452+ CCTCC-AAAGTTTCCCTC-GGGATAGCTGGC-ACTCTCGCA-AAAACCCCACTCTTGGTACCAATTTACTGTATTAGTCCA 145277528
                                                                         ***
G0IL0SA01AP2HW              401- CCTCCGAA-GTTTCCCTCAGG-ATAGCTGGCG-CTCTCGCA                                         364
G0IL0SA01ERTZB              103+ CCTCCGAA-GTTTCCCTCAGG-ATAGCTGGCG-CTCTCGCAGA                                       142
G0IL0SA01ESHS9              191- CCTCCGAA-GTTTCCCTCAGG-ATAGCTGGCG-CTCTCGCA                                         154
                                                                         ***

Other Reads:
                                                                         ***
G0IL0SA01AJBEZ               51- CCTCC-AAAGTTTCCCTC-GGGATAGCTGGC-ACTCTCGCA-AAA-CCCCACTCTT                          1
                                                                         ***


-----------------------------

>chr10  39107404        -->             ?       ?       ?               5       100.00  -       Point   3       2       3       2       var38x
Reads with Difference:
1                39107367+ TCC-ATT-CCATTCGATTCCATTCC-ATACTATTGCATTCCAATCCATTCCATTCGATTGGAATAAATTCCTTTCGAGACC 39107444
                                                                         **
G0IL0SA01CRXS7              231- TCC-ATT-CCATTCGATTCCATTCC-ATACTATTGCATTCCA                                        193
G0IL0SA02HP7T3              183+ TCC-ATT-CCATTCGATTCCATTCC-ATACTATTGCATTCCA                                        221
G0IL0SA01CVEQX  (5)          89+ TCC-ATT-CCATTCGATTCCATTCC-ATACTATTGCATTCC                                         126
G0IL0SA02HF3UR              374- -CCCATTGC-ATTCGATTCCATTCC-ATACTATTGCATTCCA                                        336
G0IL0SA01ANB24               91+ TCC-ATT-CCATTCGATTCCATTCC-ATACTATTGCATTCC                                         128
                                                                         **

Other Reads:
```

Figure 121: 454HCStructVars.txt portion example of rearrangement region.

2.18.1.15.1 Rearrangement Points

A coupled rearrangement point is found to have both a beginning and ending. A single rearrangement point has only one or the other.

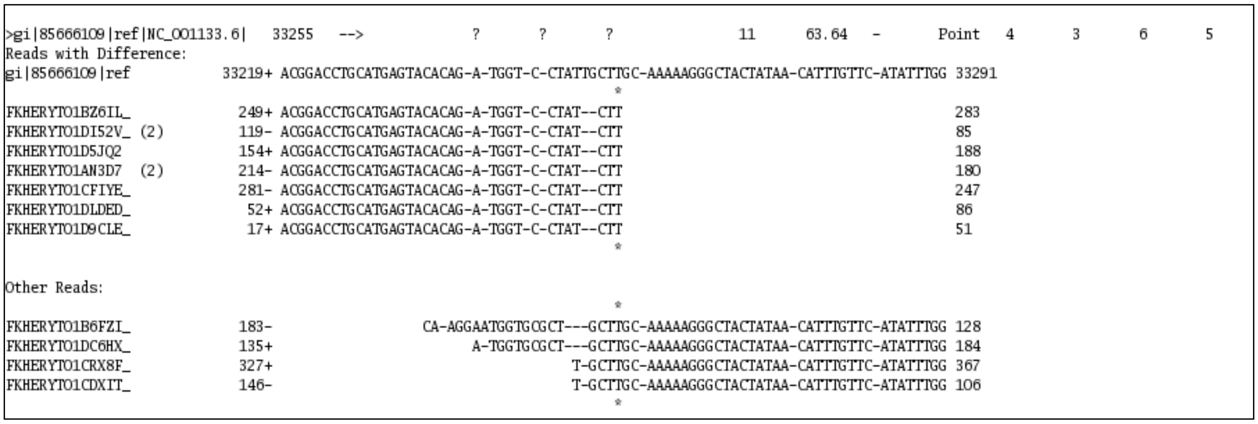
In the 454HCStructVars.txt file, coupled rearrangement points are displayed twice. The beginning is inverted, such that reads that do not contain the variation are displayed first, followed by the reads that do. The ends are in typical order, showing the reads with the variation before those without it. An example is below:



```
>gi|50593115|ref|NC_001134.7|   46830    -->      TNF1    gi|50593115|ref|NC_001134.7|    47040    <--     TNF1    11      100.00  209     Point   6       5       6
Other Reads:
gi|50593115|ref          46792+ CCTCCACGTTTT-GGT-AAGCTGCCGGGATAGGAAGTGTAATGGCGTACCCATCAACAACTGGCGTTTCTCGCCGCGAAA 46869

Reads with Difference:
                                                                        *

FKHERYTO1C2SCP_ (2)        259+ CCTCCACGTTTT-GGT-AAGCTGCCGGGATAGGAAGTGTAA                                        297
FKHERYTO1CQGFH_            258+ CCTCCACGTTTT-GGT-AAGCTGCCGGGATAGGAAGTGTAA                                        296
FKHERYTO1AIBJ9_            172+ CCTCCACGTTTT-GGT-AAGCTGCCGGGATAGGAAGTGTAA                                        210
FKHERYTO1AJDG4_            142+ CCTCCACGTTTT-GGT-AAGCTGCCGGGATAGGAAGTGTAA                                        180
FKHERYTO1BE3P5_             95- CCTCCACGTTTT-GGT-AAGCTGCCGGGATAGGAAGTGTAA                                        57
FKHERYTO1CH2A2_            102+ CCTCCACGTTTTTGGT-AAGCTGCCGGGATAGGAAGTGTAA                                        141
FKHERYTO1DWOJ9_             85+ CCTCCACGTTTT-GGT-AAGCTGCCGGGATAGGAAGTGTAA                                        123
FKHERYTO1CYJN4_ (3)         82- CCTCCACGTTTT-GGT-AAGCTGCCGGGATAGGAAGTGTAA                                        44
FKHERYTO1AXKBX             272- CCTCCACGTTTT-GGT-AAGCTGCCGGGATAGGAAGTGTAA                                        234
FKHERYTO1AHM5T_            206- CCTCCACGTTTT-GGTTAAGCTGCCGGGATAGGAAGTGTAA                                        167
FKHERYTO1BG7YF_            228- CCTCCACGTTTT-GGT-AAGCTGCCGGGATAGGAAGTGTAA                                        190
                                                                        *


Reads with Difference:
gi|50593115|ref          47000+ AACGATAGTGTCAGTGTCGCAGTAGCCATGCATAATGCAGCCGATCAAGAATTTGTC-TCGTTTTT-GGTCCATTTCATC 47077
                                                                        *

FKHERYTO1CYJN4_ (3)         43-                                         C-GATCAAGAATTTGTC-TCGTTTTT-GGTCCATTTCATC 7
FKHERYTO1BE3P5_             56-                                         C-GATCAAGAATTTGTC-TCGTTTTT-GGTCCATTTCATC 20
FKHERYTO1AJDG4_            181+                                         C-GATCAAGAATTTGTC-TCGTTTT--GGTCCATTTCATC 216
FKHERYTO1C2SCP_ (3)        298+                                         C-GATCAAGAATTTGTC-TCGTTTTT-GGTCCATTTCATC 334
FKHERYTO1DWOJ9_            124+                                         C-GATCAAGAATTTGTC-TCGTTTTTTGGTCCATTTCATC 161
FKHERYTO1CH2A2_            142+                                         C-GATCAAGAATTTGTC-TCGTTTTT-GGTCCATTTCATC 178
FKHERYTO1AHM5T_            166-                                         C-GATCAAGAATTTGTC-TCGTTTTT-GGTCCATTTCATC 130
FKHERYTO1BG7YF_            189-                                         C-GATCAAGAATTTGTC-TCGTTTTT-GGTCCATTTCATC 153
FKHERYTO1AIBJ9_            211+                                         C-GATCAAGAATTTGTC-TCGTTTTT-GGTCCATTTCATC 247
FKHERYTO1AXKBX             233-                                         C-GATCAAGAATTTGTCN-CGTTTTT-GGTCCATTTCATC 197
                                                                        *
```

Figure 122: 454HCStructVars.txt example of coupled rearrangement point.

For single rearrangement points, the unknown end is represented with a series of questions marks ("???") for the reference, min pos, and max pos, as seen below:



```
>gi|85666109|ref|NC_001133.6|   33255    -->            ?        ?        ?              11      63.64    -      Point   4        3        6        5
Reads with Difference:
gi|85666109|ref          33219+ ACGGACCTGCATGAGTACACAG-A-TGGT-C-CTATTGCTTGC-AAAAAGGGCTACTATAA-CATTTGTTC-ATATTTGG 33291
                                                                        *

FKHERYTO1BZ6IL_            249+ ACGGACCTGCATGAGTACACAG-A-TGGT-C-CTAT--CTT                                        283
FKHERYTO1DI52V_ (2)        119- ACGGACCTGCATGAGTACACAG-A-TGGT-C-CTAT--CTT                                        85
FKHERYTO1D5JQ2             154+ ACGGACCTGCATGAGTACACAG-A-TGGT-C-CTAT--CTT                                        188
FKHERYTO1AN3D7  (2)        214- ACGGACCTGCATGAGTACACAG-A-TGGT-C-CTAT--CTT                                        180
FKHERYTO1CFIYE_            281- ACGGACCTGCATGAGTACACAG-A-TGGT-C-CTAT--CTT                                        247
FKHERYTO1DLDED_             52+ ACGGACCTGCATGAGTACACAG-A-TGGT-C-CTAT--CTT                                        86
FKHERYTO1D9CLE_             17+ ACGGACCTGCATGAGTACACAG-A-TGGT-C-CTAT--CTT                                        51
                                                                        *


Other Reads:
                                                                        *

FKHERYTO1B6FZI_            183-                  CA-AGGAATGGTGCGCT---GCTTGC-AAAAAGGGCTACTATAA-CATTTGTTC-ATATTTGG 128
FKHERYTO1DC6HX_            135+                        A-TGGTGCGCT---GCTTGC-AAAAAGGGCTACTATAA-CATTTGTTC-ATATTTGG 184
FKHERYTO1CRX8F_            327+                                    T-GCTTGC-AAAAAGGGCTACTATAA-CATTTGTTC-ATATTTGG 367
FKHERYTO1CDXIT_            146-                                    T-GCTTGC-AAAAAGGGCTACTATAA-CATTTGTTC-ATATTTGG 106
                                                                        *
```

Figure 123: 454HCStructVars.txt example of a rearrangement point.

2.18.1.15.2 Rearrangement Regions

Rearrangement Regions are represented by clusters of paired-end reads that are considered False Pairs (see section 4.6.4 for a brief discussion of True Pair vs. False Pair reads). The detection of Rearrangement Regions depends on finding clusters of False Pairs of paired end reads near each other. When a cluster shows a consistent deviation from the reference (varying either in size or expected orientation), a rearrangement region is reported. An example rearrangement region is shown in Figure 121.

In the above examples, the +/- refers to whether the lowest location was the left-half of the pair (signaling that the read was in the same strand as the reference) or the right-half of the pair (signaling that the read was in the opposite strand of the reference). The arrows give the read direction for each half of the pair.

2.18.1.16 454HCStructRearrangements.txt / 454AllStructRearrangements

These two files show structural rearrangements observed in the reads of the data set analyzed, relative to the reference. The structure of the two files is the same, but 454AllStructRearrangements.txt includes all found rearrangements, while 454HCStructRearrangements includes only those that are high confidence.

The files will contain one entry block for each rearrangement. The exact specifications for the entry blocks vary by rearrangement type and are detailed in Appendix 4.13. Each block will follow a general format as follows:

- **Label** – states which type of rearrangements was found. The type can be a deletion, insertion, substitution, inversion, tandem duplication, interspersed duplication, translocation, fusion, exon splice, or circular genome. Duplications and translocations can also be inverted.
- **Reference accno** – for intra-chromosomal rearrangements, there will only be one reference. Inter-chromosomal rearrangements will span two.
- **Reference positions** – The rearrangements will involve one, two, or three points on the reference(s), depending on their type.
- **Length** – (where applicable): Most but not all rearrangements will have an associated length. See the individual specifications detailed in Appendix 4.13.
- **Confidence** – Low or High. Confidence is considered High if at least one of the individual variations comprising the given rearrangements is also High. If none of the variations are high confidence, the rearrangements will be marked as Low confidence, and will only appear in the 454AllStructRearrangements file.
- **Support and context** – The number of supporting and non-supporting shotgun reads at each involved positions will be shown where available, along with the reference context of that position. Note that rearrangements supported only by paired end but no shotgun reads at a given position will have no support information available at that position.
- **Paired End lengths** – The deviation length of supporting paired end reads from their expected library lengths will be shown. If there are less than 10 such reads, their lengths will be listed. If there are 10 or more, their lengths will be shown in a histogram. There will be a separate listing/histogram for groups of paired ends mapping to different strands.
- **Individual Variation IDs** – This is a list of the varID numbers of the individual variations in the `454AllStructVars.txt` and `454HCStructVars.txt` file that make up the given rearrangements. Note that a rearrangement can and usually will be made up of more than one variation. Note also that in the `454HCStructVars.txt` and `454AllStructVars.txt` file, the summary line for every variation has been appended with a field in the format “Var#x”, where # is the var ID number.

2.18.1.17 454HCFusionVariantTable.txt / 454AllFusionVariantTable.txt

These two files show show mapping information for all of the chimeric reads that compose the gene-to-gene and gene-to-non-gene fusion events found in the structural rearrangement files. The structure of the two files is the same, but 454AllFusionVariantTable.txt uses a lower threshold for the frequency or variant reads relative to non-variant reads than HCFusionVariantTable.txt for reporting purposes.

Each of the displayed alignments contains the following information (these are the columns in the tab-delimited format):

1. **VarID** – the Variation ID number of the corresponding variation event in the 454HCStructuralRearrangments.txt or 454AllStructuralRearrangments.txt file.
2. **Accno** – the accession number of the chimeric read.
3. **RefChr1** – the reference chromosome to which the first segment maps.
4. **RefPos1** – the position on the reference chromosome to which the first segment maps.
5. **RegionName1** – the name of the gene (if any) to which the first segment maps.
6. **QueryStart1** – the start position for the first segment on the chimeric read.
7. **QueryEnd1** – the end position for the first segment on the chimeric read.
8. **RefChr2** – the reference chromosome to which the second segment maps.
9. **RefPos2** – the position on the reference chromosome to which the second segment maps.
10. **RegionName2** – the name of the gene (if any) to which the second segment maps.
11. **QueryStart2** – the start position for the second segment on the chimeric read.
12. **QueryEnd2** – the end position for the second segment on the chimeric read.
13. **Sequence** – the read sequence, with the two chimeric fusion segments separated by a vertical bar '|' character.

```
VarID  Accno  RefChr1      RefPos1      RegionName1  QueryStart1  QueryEnd1
       RefChr2      RefPos2      RegionName2  QueryStart2  QueryEnd2    Sequence
791    HDTLPLX01A3T7W      chr1   9634875      SLC25A33     256    193    chr1
       159887916     TAGLN2 191    1      GTTTAGGGCTAAGCATAGTGGGGTATCTAATCCCAGTTTGGGTCTT
AGCTATTGTGTGTTCAGA|CCATGGTCTGGGGCTTGAGGAAGATGAGTTTGTTGATTTTAAAATAAAGA
1045   HDTLPLX01CTTZR      chr1   11807602     AGTRAP 1      43     chr1   11808472
       44     239    GTGGGCTGTGGCTCAGCGGGACTCCATCGACGCCATAAGCATG|TTTCTGGGTGGCTTGCTGGCCA
CCATCTTCCTGGACATCGTGCACATCAGCATCTTCTACCCGCGGGTCAGCCTCACGGACACGGGCCGCTTTGGCGTGGGCATGGCC
ATCCTCAGCTTGCTGCTCAAGCCGCTCTCCTGCTGCTTCGTCTACCACATGTACCGGGAGCGCGGGGGTGAGCTCCTGGTCCACA
1045   HDTLPLX02FUGT6      chr1   11807602     AGTRAP 427    321    chr1   11808472
       320    146    CATGCTTATGGCGTCGATGGAGTCCCGCTGAGCCACAGCCCACACGCCCAAGGCCAGGATGGTGAA
GTTGGCCCAGGCATAGGAGCCTGAGAATACAATGCAGCCCC|CTGATGTGCACGATGTCCAGGAAGATGGTGGCCA
```

Figure 124: 454HCFusionVariantTable.txt portion example of fusion variant report.

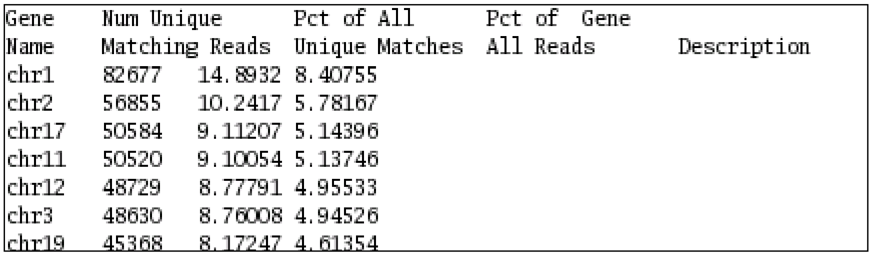
2.18.2 GS Reference Mapper cDNA / Transcriptome Output

Output for cDNA / transcriptome mapping is found in the 454RefStatus.txt file (see Section 2.18.1.12) and the 454GeneStatus.txt file (below). There is also relevant information in the 454MappingQC.xls file (Section 2.18.1.11), but with certain differences described in Section 2.18.2.2. Fusion variant mapping is described in the 454HCFusionVariantTable.txt and 454AllFusionVariantTable.txt files (Section 2.18.1.17).

2.18.2.1 454GeneStatus.txt

This gives statistical information regarding the number of reads that mapped exclusively to each gene. It has five fields:

1. **Gene Name** – the name is the gene's identifier in the related annotation database
2. **Num Unique Matching Reads** – the number of reads mapping exclusively to one or more of the transcript variants for each gene. Note that reads mapping equally well to more than one transcript variant for a gene are still considered Repeats in terms of their final ReadStatus, but are included in this statistic if they only map to variants of a single gene.
3. **Pct of all unique matches** – the number of reads exclusively mapped to the gene (as described above) divided by the total number of reads that mapped exclusively to any of the reference sequences
4. **Pct of All Reads** – the number of reads exclusively mapped to the gene (as described above) divided by the total number of reads in this mapping project
5. **Description** – the description obtained from the annotation data or renaming file



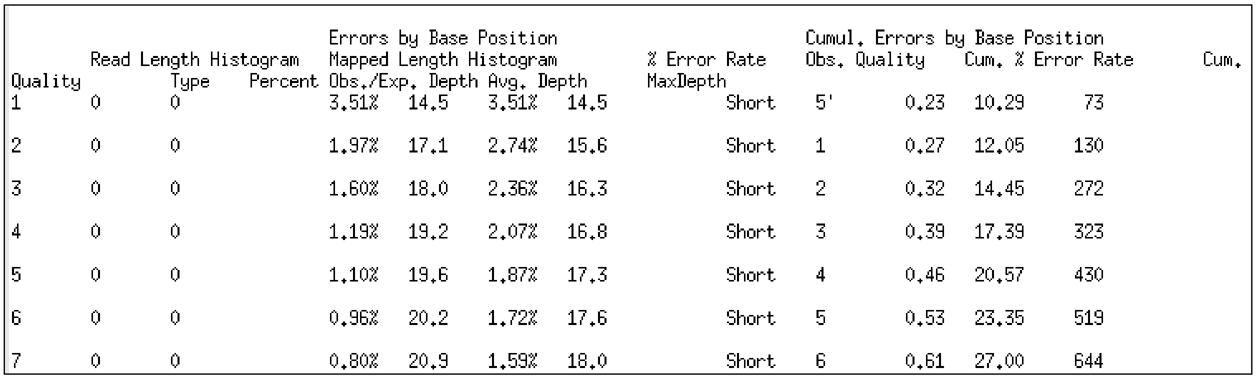
```
Gene     Num Unique      Pct of All      Pct of  Gene
Name     Matching Reads  Unique Matches  All Reads        Description
chr1     82677    14.8932 8.40755
chr2     56855    10.2417 5.78167
chr17    50584    9.11207 5.14396
chr11    50520    9.10054 5.13746
chr12    48729    8.77791 4.95533
chr3     48630    8.76008 4.94526
chr19    45368    8.17247 4.61354
```

Figure 125: 454GeneStatus.txt file example.

2.18.2.2 454MappingQC.xls

The following six columns present in the cumulative errors by base position table for Genomic mapping projects (see Section 2.18.1.11) do not appear when mapping cDNA reads to a cDNA reference: **accno**, **position**, **Avg Unique Depth**, **Min Unique Depth**, **Max Unique Depth**, and **Depth Score**. In their place, the following five columns are reported:

1. **Type** – cDNA references are grouped into three categories: Short, Medium or Long. References less than 1500 base pairs long are categorized as "Short". References greater than or equal to 1500 base pairs long but less than 3500 base pairs are categorized as "Medium". References equal to or greater than 3500 base pairs are categorized as "Long".
2. **Percent** – this column is a histogram of averaged length percentiles for all cDNA references of a given Type
3. **Obs./Exp. Depth** – number of reference bases within the percentile that were covered by reads, divided by the total number of reference bases in the percentile
4. **Avg. Depth** – average of the number of reads mapping to each reference in the given length percentile
5. **MaxDepth** – maximum number of reads mapping to a single reference in the given length percentile



```
                                Errors by Base Position                          Cumul. Errors by Base Position
         Read Length Histogram  Mapped Length Histogram          % Error Rate    Obs. Quality    Cum. % Error Rate       Cum.
Quality           Type    Percent Obs./Exp. Depth Avg. Depth     MaxDepth
1        0        0              3.51%    14.5     3.51%   14.5          Short     5'         0.23   10.29      73

2        0        0              1.97%    17.1     2.74%   15.6          Short     1          0.27   12.05     130

3        0        0              1.60%    18.0     2.36%   16.3          Short     2          0.32   14.45     272

4        0        0              1.19%    19.2     2.07%   16.8          Short     3          0.39   17.39     323

5        0        0              1.10%    19.6     1.87%   17.3          Short     4          0.46   20.57     430

6        0        0              0.96%    20.2     1.72%   17.6          Short     5          0.53   23.35     519

7        0        0              0.80%    20.9     1.59%   18.0          Short     6          0.61   27.00     644
```

Figure 126: 454MappingQC.xls file example for mapping to a cDNA reference.

3 SFF Tools Commands

The 454 Sequencing system software package contains five programs related to the handling of Standard Flowgram Format (SFF) files or other read data. These programs allow you to:

- combine files into the Standard Flowgram Format (SFF) and filter reads present in SFF files [sfffile]
- access SFF file information [sffinfo]
- dynamically generate an SCF trace file suitable for display by the consed software [sff2scf]
- prepare FASTA files with the necessary annotations for use by the GS *De Novo* Assembler and GS Reference Mapper [fnafile]
- rescore older SFF files (generated with software versions prior to v. 1.1.03) using the new phred-based quality scores [sffrescore]

These tools are all evoked from the Linux command line. The descriptions below assume that the reader is familiar with the GS Junior and GS FLX+ systems data and formats, including the SFF file format. For more details on the SFF and other file formats used in the 454 Sequencing system software, see the General Overview section of this manual.

3.1 sfffile

The sfffile command manipulates the content of SFF files, for example, to pool the results of multiple regions or sequencing runs into a single SFF file. The reads written to the new SFF file can be filtered using inclusion and exclusion lists of accession numbers, and their sequence trimming points or flowgram lengths can be adjusted.

The sfffile command uses the following syntax:

```
sfffile [options...] [MIDList@](sfffile | datadir)
```

where “`MIDList`” is a multiplexing information string used to filter the set of file reads output. (See the Appendix 4.5 for details on the use of MIDs in the data analysis.)

Command	**Description**
sfffile	Constructs a single SFF file containing the reads from a list of SFF files and/or data sets from older sequencing runs.

Option / Argument	**Description**
-o output	The default output of the sfffile program is a single SFF file containing all the reads of the SFF files / sequencing runs given in argument on the command line, output to the file 454Reads.sff. The “-o” option allows the user to specify a different filename.
-r	This option re-generates the phred-based quality scores for each of the input reads using the current quality scoring table, and overwrites the existing quality scores with these new quality scores in the output file.
-i accnofile	By default, all reads in the inputs are written to the output SFF file. If the “-i” (Include only these reads in output) or “-e” (Exclude these reads from output) options are used, the accession numbers given in the files specified will change the reads that are output. (If both -i and -e are used, the output will include the reads that are in the -i file and not in the -e file.) Those files should list the accessions one per line, where the accession should be the first word on the line after an optional ‘>’ (*i.e.*, if the line begins with ‘>’, that character is skipped, and the following characters up to the first whitespace character is read as the accession). These options are cumulative, *i.e.* if multiple -i options are given, the reads included will be the union of the -i accession lists.
-e accnofile	
-t trimfile or -tr trimfile	These two options adjust the trim points for some or all reads in the output SFF file. The specified “trimfile” should contain one or more lines consisting of (1) a read accession number, (2) a starting trimpoint, and (3) an ending trimpoint, separated by whitespace characters or where the trimpoints are separated by a dash (*e.g.*, “accno 12 543” or “accno 12-543”). The trimpoint values are 1-based positions that denote the first and last base of the trimmed region (*e.g.* for a read 800 bases in length, the lines above specify that bases 1-11 and 544-800 should be ignored, and bases 12-543 form the trimmed region of the read). A value of 0 specifies that the beginning or end of the read should be used (*e.g.* for a read 800 bases in length, the line “accno 12 0” sets the trimmed region to 12-800). The –t option will merge the given trimpoints with any existing trim points for the input read, writing the largest starting trimpoint and smallest ending trimpoint into the output SFF file. By contrast, the –tr option will “reset” the trimpoints, using only the trimpoint information occurring in this file. Note: The use of this option does not limit the reads that will be written into the output SFF file. To only output the reads in the trim file, the –i option must also be used, with this file as its argument.

Option / Argument	Description
-c # or -gsflx+ or –xlr or -gsxlr	These options change the length of the flowgrams written to the output SFF file, shortening or lengthening the flowgrams (and associated basecalled sequence) to the given number of cycles in the argument. The -c option allows an arbitrary number of cycles to be specified. If a flowgram is lengthened, 0.0 flowgram values will be added to the end of the flowgram, and no bases will be called for those flowgram values. If a flowgram is shortened, the ending flows, and all basecalls made from those flows, will be removed from the read's entry. Reads whose flowgram length matches the specified number of cycles will not be altered. Acyclic flow pattern reads will be padded with a number of flows required to bring the total flow count to four times the value specified with the –c option. Note: If a flowgram and its read are shortened, the removed data does not exist in, and is not recoverable from, the output SFF file.
-s groupname	This option "splits" the input reads into separate output files named "454Reads.MID1.sff", *etc.*, based on the closest match to MID sequences occurring at the beginning of the reads. The MID sequence and maximum number of errors are also reported. The argument to the option is an MID group name, occurring in the MID configuration file specified by the –mcf option. By default, only files with one or more reads will be output, but this can be controlled with the -mrt option. Using –s without specifying a group name may result in a handful of reads being misassigned to MIDs in other groups defined in the configuration file, which might be remedied by re-filtering using the -rf option.
-si groupname	This option reports counts of reads that share an MID, similar to –s, but without actually splitting the input SFF file.
-both	Use both 5' and 3' MID tags for MID splitting by MID using -s or -si.
-rf	Re-filter (-s) or re-count (-si) the file's reads based on the majority of MIDs found in the first pass (works only with standard GS and RL schemas).
-mrt	Specify the minimum number of reads that must be detected for each MID in order to be counted (-si) or to create a new SFF file (-s).
[-mcf *filename*]	This option specifies a different MID configuration file to be used by the command for decoding the multiplex information appearing on the command line.
[-pick #] or [-pickb #]	This option tells sfffile to generate an output file containing a certain number of bases, by randomly "picking" reads from the input. The argument can be a number, optionally followed by either a 'k' or 'm' to specify thousands or millions of bases, respectively. For example, "-pick 3000000" and "-pick 3m" both tell sfffile to pick random reads from the input and generate an output file containing 3 million bases. Note: Reads are not broken to achieve exactly the number of bases specified, so the actual number of bases in the output file may differ slightly from the desired number.

Option / Argument	Description
[-pickr #]	This option tells sfffile to generate an output file containing a certain number of reads, by randomly "picking" reads from the input. The argument number can be followed by a 'k' or 'm' to specify thousands or millions of reads, respectively. Reads in any input SFF files are merged directly into the single output SFF file generated by this command. If SFF files are not present in a Data Processing folder (*e.g.* for a run whose data has been processed with a version of the 454 Sequencing system software anterior to 1.0.52), the signal processing step of the GS Run Processor application must be rerun on the Data Processing folder, and can then merged into the output SFF file generated by the sfffile command. Input "D_..." directories (with data generated on a GS FLX+ Instrument) may be prepended with a list of regions separated by a colon. For example, "1,3-5,7:R_dir/D_dir" tells sfffile to use regions 1, 3, 4, 5 and 7 of R_dir/D_dir. An optional multiplexing information string can be prepended to each file/data-directory argument.
[-nmft]	By default, a "manifest" listing the set of sequencing runs that were the source of the reads in an SFF file is written into the index section of the SFF file (to provide an explicit traceback to the origin of the reads). This includes the run name (the R_... name), the Data Processing name (the D_... name) and the full path to the Data Processing directory used in the conversion from run data to SFF file(s). The "-nmft" option prevents the propagation of the manifest from the input SFF files into the output SFF file.
[--help]	Displays a usage line and short description of the command.
[--version]	Displays the current software version of the command.
[MIDList@] (sfffile | datadir)	Path to one or more SFF files and/or Data Processing directories ("D_..."). Reads in any input SFF files or the SFF files in the Data Processing directories are merged directly into the single output SFF file generated by this command. Input "D_..." directories (with data generated on a GS FLX+ Instrument) may be prepended with a list of regions separated by a colon. For example, "1,3-5,7:R_dir/D_dir" tells sfffile to use regions 1, 3, 4, 5 and 7 of R_dir/D_dir. An optional multiplexing information string can be prepended to each file/data-directory argument (see Appendix 4.5).

3.1.1 Using sfffile to Merge Multiple Read Files

One of the most common uses of the sfffile program is to collect and organize sequences in terms of a user's projects, instead of the standard timestamp- and run-based organization generated by the GS Junior and GS FLX+ instruments. For example, multiple sff files can easily be combined into a single "mygenome.sff" file using

```
sfffile -o mygenome.sff path_to_file1.sff path_to_file2.sff path_to_file3.sff
```

This file could then be given to the GS *De Novo* Assembler and GS Reference Mapper software, or carved and divided into separate pieces (see below), while still preserving the traceback information to the source data, since:

- The universal accessions for each read encode the run timestamp, the region and the X,Y location of the read, and all but ensure uniqueness for the accessions.
- The manifest of the SFF file provides the mapping from the run prefix of the universal accessions to the actual Run and Data Processing directories used to generate the read. When universal accessions are given to the reads, the manifest is kept updated through multiple invocations of sfffile (as best it can).

Thus, if a particular read needs investigation, the user can invoke the sffinfo command (see section 3.2, below) for that read to see the traceback information, including the source Run and Data Processing directories. The GS Run Browser (see Part B of this Reference Manual) could be used to evaluate the read within the context of the run [where the Find location feature on the Wells tab allows entry of a universal accession and will then center the image and place a marker at the read's X,Y location].

Each SFF file stores the list of underlying flows (defined by the flow pattern) in a common header. When merging multiple files, the resulting SFF file must include a flow list that applies to each of the reads in the merged file. For each pair being merged, only SFF files that share the flow list of the shorter read length file can be merged, the merged file uses the flow list of the longer read length file, and shorter reads are padded with empty flows so that all reads are the same length. Longer lists of files are merged sequentially.

Note that the -c option no longer needs to be used before merging SFF files that share a cyclic flow pattern, but differ in the number of cycles.

3.1.2 Using sfffile to Extract Read Subsets

Another common use of sfffile is to extract subsets of the reads from a sequencing Run or a set of sequencing runs, to perform further processing. Using the –i and –e options, one can construct an SFF file containing only the reads of interest, and then use that for further processing, *e.g.* input it into to the GS *De Novo* Assembler or GS Reference Mapper applications. For example, a user can extract the singletons from an assembly (when performing, say, survey sequencing) by executing the following commands (which assume that the current working directory is the output directory of the assembly):

```
grep Singleton 454ReadStatus.txt > singles.txt

sfffile -o singles.sff -i singles.txt sff/*

sffinfo -s singles.sff > singles.fna
```

The Linux grep command extracts the lines in the 454ReadStatus.txt file that list the accession numbers of the singleton reads, then the –i option in the sfffile command is used to signify that the command should "include" the reads in the new SFF file. These reads are then output to FASTA format using sffinfo.

Or, after running the GS Reference Mapper application, you can retrieve the reads that did not map to your reference sequence (so that, for example, you can give them as input to the GS *De Novo* Assembler software to see if there are any contigs that are part of what you sequenced but are not part of the reference) by using the following commands (which assume that the current working directory is the output directory of the assembly):

```
grep Unmapped 454ReadStatus.txt > unmapped.txt

sfffile -o unmapped.sff –i unmapped.txt sff/*

runAssembly unmapped.sff
```

The Linux grep command extracts the lines in the 454ReadStatus.txt file that list the accession numbers of the unmapped reads, then the –i option in the sfffile command is used to signify that the command should "include" the reads in the new SFF file. (Similarly, there is a –e option to "exclude" reads from the output file.)

The -i and -e options cannot take a list of files, so to specify the inclusion or exclusion of multiple files on the command, you must precede each with an -i or -e option.

3.2 sffinfo

The sffinfo command extracts read information from an SFF file, and reports it in text form. It can be used for generating the FASTA and quality score files of the read sequences, which can be given to standard bioinformatics tools. The sffinfo command uses the following syntax:

```
    sffinfo [options...] [- | sfffile] [accno...]
```

Command	Description
sffinfo	The sffinfo command extracts read information from an SFF file, and reports it in text form. By default, a text summary of all the read information is output, but the output can be limited to only the sequences, quality scores, flowgrams or manifest. All output is written to standard output. If trimming occurs, sffinfo will exclude zero-length reads from the output, and will report the trimmed (usable) length.

Option / Argument	Description
-a or -accno	This option limits the output to only the accession numbers.
-s or -seq	This option limits the output to only the sequences.
-q or -qual	This option limits the output to only the quality scores.
-f or -flow	This option limits the output to only the flowgrams.
-c or –cheader	This option limits the output to only the common header information, which provides a quick summary of the contents of the SFF file.
-t or -tab	The default format for the sequences, quality scores and flowgram is FASTA. This option changes this to tab-delimited lines.
-n or -notrim	By default, the trimmed data is output. The -notrim option changes this to output the untrimmed sequences or quality scores.
-m or -mft	The command does not output the manifest by default. This option outputs the manifest text, if it exists in the SFF file. In this case, the -tab and -notrim options and the accnos on the command line are ignored.
[--help]	Displays a usage line and short description of the command.
[--version]	Displays the current software version of the command.
- | sfffile	Path to the SFF file whose data will be output. If this is replaced by a “-” on the command line, then standard input is read for the SFF contents.
accno...	List of the accession numbers of the reads whose data will be output. If no accnos are given on the command line, the information from all reads in the file are output. If an SFF file is specified and it contains an sfffile-created index, then an index-based lookup is used and the reads are output in the order they appear on the command line. If not, the complete SFF file is scanned and the reads are output in the order they appear in the file.

Only one of -accno, -seq, -qual, -flow or -mft may be specified: the program uses the last one specified on the command line.

3.3 sff2scf

The sff2scf command converts the flowgram information from a read's SFF entry into an SCF file, creating a synthetic "trace" for the read. In this software version, it has been written mainly for program-triggered execution by the consed software, and minimal support is provided for direct command line use. (The main reasons for this are that the command converts a 1000 byte SFF entry into a 45,000 byte SCF file, and that the synthetic trace generated cannot be used by other software, like polyphred, which expect a trace from a Sanger read.)

The sff2scf command uses the following syntax:

```
sff2scf locationstring [outputfile]
```

Command	Description
sff2scf	The sff2scf command extracts one read's information from an SFF file and converts it into an SCF file (or performs "call throughs" to access SCF data for Sanger reads).

Option / Argument	Description
[--help]	Displays a usage line and short description of the command.
[--version]	Displays the current software version of the command.
locationstring	A "locationstring" that tells sff2scf what path or what command to use to access the read's trace information. See below for a description of the format of this string.
outputfile	The path to where the SCF file should be written. If no outputfile is given, the output is written to "/tmp/locationstring" (the location that consed expects to read the data).

The locationstring argument can take one of 4 forms, which allow the sff2scf command to fully support accessing SFF and SCF data for the reads that may appear in an assembly or mapping output from the 454 Sequencing system. The forms are the following:

1. If the locationstring begins with "sff:", then it specifies an SFF file and read accession to get the SFF data, and will cause the generation of a synthetic SCF trace file. The format of the locationstring should be either "sff:*path*:*accno*" or "sff:-f:*path*:*accno*" and is processed using the following rules:
    a. Any instance of the string ":_:" in the path will be translated into the character "/", so that the locationstring can be used as a simple filename when executed by consed even though it encodes a path through the directory structure.
    b. If the path begins with the character "/" (after rule a), it is treated as the absolute path to the SFF file.
    c. If the path does not begin with "/", both "path" and "../sff_dir/path" are checked for the existence of a file (the "../sff_dir/path" format is the form used by the GS *De Novo* Assembler and the GS Reference Mapper applications when generating a full consed directory structure).
    d. The command "sffinfo truepath accno" is executed to access the read's SFF data (where "truepath" is the path determined above).
    e. If the "-f" string appears immediately following "sff:", the read's flowgram is reverse-complemented (or "flipped") prior to the generation of the SCF file. (This occurs when the read is the left half of a 454 paired end read. The consed software (and other software) assume a directionality of paired end reads, where the clone that was used to generate the paired end read is assumed to extend off the 3' end of both reads in the pair. In the processing of 454 paired end reads, this is not the case for the left half of the paired end read (the clone extends from the 5' end of this half). To support consed and the standard view of paired end reads, the GS *De Novo* Assembler reverse-complements all left halves of paired end reads, and so this command must support the reverse-complementing in the generation of the SCF trace).
2. If the locationstring begins with "file:", it is treated as specifying the path to an SCF file, of the form "file:*path*", and is processed using the following rules:
    a. Any instance of the string ":_:" in the path will be translated into the character "/", so that the locationstring can be used as a simple filename when executed by consed even though it encodes a path through the directory structure.
    b. If the path begins with the character "/" (after rule a), it is treated as the absolute path to the SCF file.
    c. If the path does not begin with "/", both "path" and "../chromat_dir/path" are checked for the existence of a file.
    d. The bytes of the file (determined from above) are copied into the output file without change.
3. If the locationstring begins with "cmd:" it is treated as specifying a command string to be executed to access the SCF contents, is assumed to have the form "cmd:*commandline*", and is processed using the following rules:
    a. Any instance of the string ":_:" in the command line will be translated into the character "/", so that the locationstring can be used as a simple filename when executed by consed even though it encodes a path through the directory structure.
    b. After the above, any colon ":" will be translated into a space " ", so that the locationstring can be used as a CHROMAT string, which cannot contain whitespace, in the ACE and PHD files used by consed (and will be the strings passed to sff2scf file by consed).
    c. The resulting command line string is executed using the proper command, and the standard output of the command is copied into the output file without change.
4. Otherwise, the locationstring is treated as a path to an SCF file, and processed using the following rules:
    a. If the path begins with "/", that is treated as the absolute path to the file
    b. If the path does not begin with "/", both "path" and "../chromat_dir/path" are checked for the existence of a file, where the "../chromat_dir/path" form exists to support the addition of new reads when running consed itself (the default operation of consed, looking in the ../chromat_dir directory to find the SCF file for a read).
    c. The bytes of the file (as determined above) are copied into the output file.

The complex syntax of the location strings is needed because when sff2scf is used within consed, (1) it is responsible for accessing the traces for all the reads in the assembly (454 Sequencing reads with SFF data, Sanger reads with SCF files, or commands that generate SCF data); (2) it is only given a single argument (the “CHROMAT” string in the ACE file or PHD file, which cannot contain whitespace) and must produce a file named by that same argument, and (3) that argument string must be a valid simple filename (not a path or command), even if the source of the data is a full path or command string.

3.4 fnafile

The fnafile command provides similar functionality for FASTA files as the sfffile command provides for SFF files, and in addition provides a mechanism to easily generate and add annotation strings to the description lines of reads in the file. The command constructs a single FASTA file containing the reads from a list of FASTA, PHD or SCF files, or directories containing FASTA, PHD or SCF files. The reads written to the new FASTA file can be filtered using inclusion and exclusion lists of accession numbers, and their sequence trimming points or annotation strings can be adjusted. This command is used to pool the results of many Sanger reads into a single FASTA file, to simplify further handling of the combined data.

The fnafile command uses the following syntax:

```
fnafile [options] (fastafile or PHDfile or SCFfile or directory)...
```

In version 2.7 and later, fnafile supports MID filtering and splitting in the same way as the sfffile command, although trimming points included in the FASTA file description line are ignored.

Command	Description
fnafile	Constructs a single FASTA file (plus associated quality score file) containing the reads from a list of FASTA files (possibly with associated quality score files), PHD files, SCF files or directories containing FASTA, PHD or SCF files.

Option / Argument	Description
-o output	The "-o" option allows the user to specify a different filename for the output fna and qual files.
-i accnofile	By default, all reads in the inputs are written to the output file. If the "-i" (Include only these reads in output) or "-e" (Exclude these reads from output) options are used, the accession numbers given in the files specified will change the reads that are output. The associated accnofile files should list the accession numbers one per line. These options are cumulative, *i.e.* if multiple -i options are given, the reads included will be the union of the -i accession lists.
-e accnofile	
-w line width	Flag to specify the width of each line of sequence in an output fna file. Use a value of '0' to output the sequence on a single line.
-t trimfile or -tr trimfile	These two options adjust the trim points for some or all reads in the output file. The specified "trimfile" should contain one or more lines consisting of (1) a read accession number, (2) a starting trimpoint and (3) an ending trimpoint, separated by whitespace characters or where the trimpoints are separated by a dash (*e.g.*, "accno 12 543" or "accno 12-543"). The trimpoint values are 1-based position values that denote the first and last base of the trimmed region (*e.g.* for a read 800 bases in length, the example above specify that bases 1-11 and 544-800 should be ignored, and bases 12-543 form the trimmed region of the read). A value of 0 specifies that the beginning or end of the read should be used (*e.g.* for a read 800 bases in length, the line "accno 12 0" sets the trimmed region to 12-800). The –t option will merge the given trimpoints with any existing trim points for the input read, writing the largest starting trimpoint and smallest ending trimpoint into the output. The –tr option will "reset" the trimpoints, using only the trimpoint information occurring in this file. Note: The use of this option does not limit the reads that will be written into the output file. To only output the reads whose accessions appear in the trim file, the –i option must also be used, with the same file as its argument.

Option / Argument	Description
-af filename	This option specifies a file that contains adjustments to the description line for some or all reads. Each line should contain the adjustments for a single read, and the line must begin with the accession number of the read. The text that follows the accession number (and subsequent whitespace characters) can take one of two forms. (1) If the text to the right of the accession number does not begin with a '>', it is appended to the description line for the read. For example, the line DJS045A03F template=DJS045A03 dir=F library=DJS045 will append the text "template=DJS045A03 dir=F library=DJS045" to the end of the current description line for read "DJS045A03F". (2) If the text to the right of the accession number does begins with a '>', it will completely replace the description line for the input read, *including changing the accession number for the read*. For example, the line gi|tl|00103402131 >DJS045A03F template=DJS045A03... will change the description line for input read "gi|tl|00103402131" to be ">DJS045A03F template=DJS045A03…" in the output FASTA file that is written (and effectively change the accession number of the read for any program which uses the output FASTA file). This feature is useful for translating less descriptive read accession numbers (such as the NCBI Trace Archive accessions) into more descriptive read accession numbers (such as the genome project accession numbers, which encode library, plate-well and direction in the accession numbers).
-ac command	This option specifies a command to be executed for each input read, to determine adjustments to make to the read's description line in the output FASTA file. The fnafile command will execute the command string "*command accno* '*description*'" for each read, where "*command*" is the value of the –ac option, "*accno*" is the first non-whitespace string on the input read's description line, and "*description*" is the input read's full description line (including the accno). The command processes each line of the file individually and writes the result to standard output. The output should be formatted as described above for the "text after the accession" for the –af option and will be processed by the fnafile command the same way. If no line of output is written, the input read's description line will remain unchanged.
-s groupname	This option "splits" the input FASTA file reads into separate output files named "454Reads.MID1.fna", *etc.*, based on the closest match to MID sequences occurring at the beginning of the reads. The MID sequence and maximum number of errors are also reported. The argument to the option is an MID group name, occurring in the MID configuration file specified by the –mcf option. By default, only files with one or more reads will be output, but this can be controlled with the -mrt option. Using –s without specifying a group name may result in a handful of reads being misassigned to MIDs in other groups defined in the configuration file, which might be remedied by re-filtering using the -rf option.
-si groupname	This option reports counts of FASTA file reads that share an MID, similar to –s, but without actually splitting the input FASTA file.
-both	Use both 5' and 3' MID tags for MID splitting by MID using -s or -si.

Option / Argument	**Description**
-rf	Re-filter (-s) or re-count (-si) the file's reads based on the majority of MIDs found in the first pass (works only with standard GS and RL schemas).
-mrt	Specify the minimum number of reads that must be detected for each MID in order to be counted (-si) or to create a new FASTA file (-s).
[-mcf *filename*]	This option specifies a different MID configuration file to be used by the command for decoding the multiplex information appearing on the command line.
[--help]	Displays a usage line and short description of the command.
[--version]	Displays the current software version of the command.
fastafile or PHD file or SCF file or directory	Path to a FASTA file, PHD file, SCF file or a directory containing FASTA, PHD or SCF files. The reads in any input files are merged directly into the single output FASTA file generated by this command. If a directory is given, all the files it contains are read, and any files in the directory that are FASTA, PHD or SCF are processed and their reads are included in the output. Note: This command is *not* aware of consed's phd_dir versioning of reads. If a phd_dir directory is given that contains multiple versions of the PHD file for a single read, all versions will be read and used by the command, and multiple versions of those reads will appear in the output FASTA file. The same holds true if a phd.ball file and individual PHD files are given on the command line.

3.5 sffrescore

The sffrescore command can be used to rewrite existing SFF files with the new Phred-like quality scores. It recursively searches through a list of files or directories, identifies all the SFF files, determines which SFF files contain the older quality scores, runs the sfffile command (with the –r option) to generate a new SFF file with the new quality scores, then overwrites the existing SFF file with the new file. The effect is that the reads, flowgrams and basecalls in the files are left unchanged, but the files now contain new quality scores.

The sffrescore command uses the following syntax:

```
sffrescore [-f] (file | directory)...
```

Command	**Description**
sffrescore	The sffrescore command recursively searches through a list of files or directories, and rescores any SFF file that contains the older quality scores. Any file rescored is rewritten in place with a new version of the SFF file.

Option / Argument	**Description**
-f	This option forces the rescoring of all SFF files (by default, any SFF file found to already have the new scores are not rewritten).
file or directory...	List of the files and directories to be processed. The command will recursively search through the directories to find all of the SFF files in the directory trees.

4 GS *DE NOVO* ASSEMBLER AND GS REFERENCE MAPPER APPENDICES

4.1 Commands for Assembly and Mapping

The CLI commands for assembly and mapping are given in the table below. See Section 1.14 for details about assembly commands, Section 2.16 for details about mapping commands, and Section 4.2 for details about command options and parameters.

Command	Description	Usage
addRun Section 1.14.3.2 Section 2.16.3.3	Command to add Read Data sets to an existing incremental assembly or mapping project. Not used with multiplex projects.	`addRun [options: (-p | -np) -lib (-mcf | -custom)] [projDir] [MIDList@]filedesc`
addSample Section 2.16.5.1	Command to add Read Data sets to an existing multiplex assembly or mapping project, along with the relationship between samples and MIDs (specified in a sample association .tsv file).	`addSample [options: (-p | -np) –lib (-mcf | -custom) –clear -datapath] –tsv samples.tsv projDir [[MIDList@]filedesc]`
changeRef Section 2.16.1.2	Command to change the path of an incremental mapping project's reference sequence(s), for example after a project has been moved from one computer to another one, or the network location where the original paths to the reference sequence(s) are no longer accessible.	`changeRef [options: -verbose [-verbose] -relink] projDir [sourcePath destinationPath]`

changeRun
Section 1.14.1.1
Section 2.16.1.1
	Command with two modes:
	[1] to change pairing info on an incremental assembly or mapping project's read files,
	or
	[2] to change the path of an incremental assembly or mapping project's read files, for example after a project has been moved from one computer to another one, or the network location where the original paths to the read files are no longer accessible.
		`changeRun (-p | -np) [options: n/a] projDir [MIDList@]filedesc`
		or
		`changeRun [options: -verbose -relink] projDir [sourcePath destinationPath]`

newAssembly Section 1.14.3.1	Command to initiate an incremental assembly project and set up an assembly project folder to contain the project data; used when multiple runs must be incrementally assembled over time.	`newAssembly [options: -cdna -multi] projDir`
newMapping Section 2.16.3.1	Command to initiate an incremental mapping project and set up a mapping project folder to contain the project data; used when multiple runs must be incrementally aligned over time or to change the reference sequence.	`newMapping [options: -cdna (–cref | -gref) –multi] projDir`

Command	Description	Usage
removeRun Section 1.14.3.3 Section 2.16.3.4	Command to remove Read Data sets from existing incremental assembly or mapping projects. Requires full path to non-SFF read files, but only needs the file name of files in the project SFF directory,	`removeRun [options: n/a] [projDir] (sffname | readfastafilepath | readfastqfilepath)`
runAssembly Section 1.14.2	Command for single shot (one step) assembly of reads; used when all reads are available at once. For multiplex projects, the MIDList filtering of read files is not supported.	`runAssembly [options] [MIDList@]filedesc`

runMapping
Section 2.16.2
	Command for single shot (one step) mapping of reads; used when all reads are available at once. For multiplex projects, the MIDList filtering of read files is not supported.	`runMapping [options]`
		`[-ref refdesc1 refdesc2 …] refdesc`
		`[-read [MIDList1@]filedesc1`
		`[MIDList1@]filedesc2…] [MIDList@]filedesc`

runProject Section 1.14.3.4 Section 2.16.3.5	Command to perform the actual computation of an incremental assembly or mapping project.	`runProject [options] [projDir]`
setRef Section 2.16.3.2	Command to assign the reference sequence(s) in an incremental mapping project.	`setRef [options: (-gref | -cref) -random] [projDir] refdesc`
stopRun	Command to stop computation of a project. There may be a delay between the time this command is used and the point at which the software actually acts on the command.	`stopRun [options: n/a] [projDir]`

Table 5: Commands for Assembly and Mapping. Unless otherwise noted, all commands work with both multiplex and ordinary (non-multiplex) projects.

For all of these commands;

- The arguments in square brackets [ ] are optional.
- The '|' symbol means 'OR'.
- 'options' refers to the optional command arguments available.
- 'projDir' is the path of the data analysis project directory.
- 'refdesc' is a path/name to a reference sequence file in FASTA format.
- 'filedesc' = (sfffile | fnafile | [regionlist:]analysisDir), referring to a read data file in one of the following formats:
    - an sfffilename
    - a [regionlist:]datadir
    - a readfastafile
    - a space-separated list of any of the above
- '[MIDList@]filedesc', where each 'MIDList' is a multiplexing information string used to filter the set of file reads to be used in the mapping (if MIDs were used in the generation of the read data file, then an MIDList string must be specified in order for the mapper to properly handle the file's reads.) For details about using MIDs, see section 4.5.

4.2 Options for Assembly and Mapping Commands

The options for the CLI assembly and mapping commands and the corresponding GUI options for assembly and mapping are given in the table below. See Section 4.1 for details about the mapping and assembly CLI commands that use these options.

Option	Parameter [Default]	Description	Persistent	runAssembly	runMapping	runProject	Other CLI Commands	GUI	GUI Parameter
--help	none	Flag to display command usage.		X	X	A M		A M	Help button
--version	none	Flag to display command version.		X	X	A M		A M	About button
-a	bases >=0 [100] [0 for cDNA assembly]	Flag to specify minimum length of multiple alignments of contigs to be included in 454AllContigs.fna. Some contigs slightly shorter than this may be included since contig consensi are computed after contigs are flagged for inclusion in the 454AllContigs.fna file. When the consensi are computed, some alignment columns may not yield consensus bases. For cDNA assembly, this value is set to 0 and can't be changed.	X	X	X	A M		A M	**Output**: All contig threshold [value]
-accno	filedesc	Flag to specify an accno renaming file, to allow reference accession numbers and descriptions to be changed in the generation of the output (has no effect with genomic projects).	X		X	M		M	**Input**: Accno renaming file [filedesc]
-ace	none	Flag to specify single ace file generation. The default behavior (with no option specified) is to output a single ACE file for small genomes (fewer than 4 million reads and a reference less than 40 Mbp). Related options include -acedir, -consed, -consed16, -noace, and -nobig.	X	X	X	A M		A M	**Output**: Ace Format [Single ACE file]

Option	Parameter [Default]	Description	Persistent	runAssembly	runMapping	runProject	Other CLI Commands	GUI	GUI Parameter
-acedir	none	Flag to generate ace file subdirectory with an ace file for each contig output. The default behavior (with no option specified) is to output a single ACE file for small genomes (fewer than 4 million reads and a reference less than 40 Mbp). Related options include -ace, -consed, -consed16, -noace, and -nobig.	X	X	X	A M		A M	**Output**: Ace Format [Ace file per contig] or [Ace file per reference]
-ad	none	Flag to output default reads in ACE or consed folder. Default for assembly = trimmed Default for mapping = raw Related options include -ar and -at.	X			A M		A M	**Output**: Ace read mode [Default]
-ads	score any integer [-3]	Flag to set the alignment difference score parameter.	X	X	X	A M		A M	**Computation**: Alignment difference score [value]
-ais	score any integer [2]	Flag to set the alignment identity score parameter.	X	X	X	A M		A M	**Computation**: Alignment identity score [value]
-annot	filedesc	Flag for optional file with gene-coding/region annotation for reference sequence(s) for gene name and protein translation info in Variant GUI tab.	X		X	M		M	**Input**: Genome Annotation [filedesc]
-ar	none	Flag to output full raw read sequences in ace or consed folder. Related options include -ad and -at.	X	X	X	A M		A M	**Output**: Ace read mode [Raw]
-at	none	Flag to output only trimmed sequence reads in ace or consed folder. Related options include -ad and -ar.	X	X	X	A M		A M	**Output**: Ace read mode [Trimmed]

Option	Parameter [Default]	Description	Persistent	runAssembly	runMapping	runProject	Other CLI Commands	GUI	GUI Parameter
-bam	none	Flag to generate bam file containing the multiple alignments for the contigs. The default behavior (with no option specified) is to output a single BAM file for small genomes (fewer than 4 million reads and a reference less than 40 Mbp). Related options include -nobam and –nobig.	X		X	M		M	**Output**: BAM Output [Single BAM file]
-bamsub	none	Flag to specify bam adjacent indel substitution.		X	X	A M			
-batchsize (multiplex only)	CPUs 1 - # avail [1]	Flag to specify the number of sample projects to be computed simultaneously for a multiplexing project.		X	X	A M		A M	**Computation**: Batch size [value]
-cdna (cDNA only)	none	Flag for transcriptome projects. Use without an option to specify a genomic project.		X	X		newAssembly newMapping	A M	**New Project dialog**: Sequence type [cDNA]
-clear (multiplex only)	none	Flag to remove the sample association .tsv file from a multiplex project.					addSample		
-consed	none	Flag to generate a consed file subdirectory and place all consed necessary file and folders within (consed V17.0). The default behavior (with no option specified) is to output a single ACE file for small genomes (fewer than 4 million reads and a reference less than 40 Mbp). Related options include -ace, -acedir, -consed16, -noace, and -nobig.	X	X	X	A M		A M	**Output**: Ace Format [Complete Consed folder]

Option	Parameter [Default]	Description	Persistent	runAssembly	runMapping	runProject	Other CLI Commands	GUI	GUI Parameter
-consed16	none	Flag to generate output compatible with versions of consed older than V17.0. (-consed is compatible with version 17.0). The default behavior (with no option specified) is to output a single ACE file for small genomes (fewer than 4 million reads and a reference less than 40 Mbp). Related options include -ace, -acedir, -consed, -noace, and -nobig.	X	X	X	A M		A M	**Output**: Ace Format [Consed16]
-cpu	CPUs 0 - # avail [0]	Flag to specify number of CPUs to use for each project (or sample project in a multiplexing project). Default of '0' means use all. If used in conjunction with –batchsize, the number of CPUs may be reset so as not to exceed the total number available.	X	X	X	A M		A M	**Computation**: Number of CPUs to use [value]
-cref	none	Flag for cDNA reference sequence. Use no option for auto-detection of reference sequence type.	X		X	M	newMapping setRef	M	**Input**: Reference type [cDNA]
-custom	List of custom MIDs	Flag to specify a custom MID group.					addRun addSample	A M	**Read Data Attributes dialog**: Custom Multiplexing [MID info]
-d	reads >=1 [1]	Flag to set the minimum contig depth.	X		X	M		M	**Output**: Minimum contig depth [value]
-datapath (multiplex only)	directory	Flag to specify an optional datapath to a directory of reads in a multiplex project.	X				addSample		
-e	reads >=0 [0]	Flag to specify expected depth of data, needed if depth >50, 0 resets to default (ignore).	X	X	X	A M		A M	**Input**: Expected depth [value]

Option	Parameter [Default]	Description	Persistent	runAssembly	runMapping	runProject	Other CLI Commands	GUI	GUI Parameter
-fd	none	Flag to specify full variant file details.			X	M		M	**Output**: Full variant file details [checked]
-fe	filedesc	Flag for exclude filter file to be specified.	X	X	X	A M		A M	**Input**: Exclude filter file [filedesc]
-fi	filedesc	Flag for include filter file to be specified.	X	X	X	A M		A M	**Input**: Include filter file [filedesc]
-force	none	Flag required to overwrite an existing project directory.		X	X				
-gencode	filedesc | 64 character description	Flag to specify various types of genetic codes (bacterial, mitochondrial, eukaryotic, *etc.*) for predicting protein translation products during variant calling.			X	M			
-gref (cDNA only)	none	Flag for genomic reference sequence. Use no option for auto-detection of reference sequence type. This is the default for genomic projects, and therefore does not need to be specified.	X		X	M	newMapping setRef	M	**Input**: Reference type [Genomic]
-het	none	Flag to enable Heterozygotic mode, which causes the assembler to choose a path between 2 contigs when there is ambiguity regarding the path that should be taken. By default, the assembler will not choose a path when ambiguity exists.	X	X		A		A	**Input**: Heterozygotic mode [checked]
-hsl	score 1 – 2000 [70]	Flag to set the Hit-per-seed limit parameter (new default value optimized for newer algorithms, with values >2000 re-set to 70).	X		X	M		M	**Computation**: Hit-per-seed limit [value]

Option	Parameter [Default]	Description	Persistent	runAssembly	runMapping	runProject	Other CLI Commands	GUI	GUI Parameter
-icc (cDNA only)	contigs 1 – 200 [100]	Flag to specify the maximum number of contigs in an isotig; corresponds to the recursion depth during graph traversal.	X	X		A		A	**Output**: Isotig contig count threshold [value]
-icl (cDNA only)	bases >=0 [3]	Flag to specify the minimum length a contig must be to be part of an isotig. Some contigs slightly shorter than this may be included in isotigs due to the two-phase nature by which contig consensi are determined.	X	X		A		A	**Output**: Isotig contig length threshold [value]
-ig (cDNA only)	contigs >=1 [500]	Flag to specify the maximum number of contigs in an isogroup.	X	X		A		A	**Output**: Isogroup threshold [value]
-info	none	Flag to output to 454AlignmentInfo.tsv file. Related options include -infoall, -nft, and -noinfo.	X	X	X	A M		A M	**Output**: Alignment info [Output small]
-infoall	none	Flag to output to 454AlignmentInfo.tsv file including 0-coverage positions. Related options include -info, -nft, and -noinfo.	X	X	X	A M		A M	**Output**: Alignment info [Output]
-isplit (cDNA only)	none	Initiate isotig traversal when depth spikes in alignments are found. These spikes will be treated as putative splice events so that traversal along the path will continue producing more isotigs.	X	X		A		A	**Computation**: Isotig depth split [checked]
-it (cDNA only)	isotigs 1 – 10,000 [100]	Flag to specify the maximum number of isotigs in an isogroup.	X	X		A		A	**Output**: Isotig threshold [value]
-l	bases > all contig threshold [500]	Flag to set the minimum length for contig to appear in 454LargeContigs.fna.	X	X	X	A M		A M	**Output**: Large contig threshold [value]

Option	Parameter [Default]	Description	Persistent	runAssembly	runMapping	runProject	Other CLI Commands	GUI	GUI Parameter
-large	none	Flag to enable large genome assembly mode. This option should not be used in cDNA assembly projects.	X	X		A		A	**Input**: Large or complex genome [checked]
-lib	libName	Flag to define the default paired end library. It can be used to group together the paired end reads from different files, so that they are all used together in calculating a mean distance between the halves of each paired end pair of reads. (By default, each SFF file is treated as a separate library, and a separate mean paired end distance is calculated for each.).					addRun addSample		
-m	none	Flag to keep sequence data in memory to speed up cpu time (requires large CPU memory).		X	X	A M			
-maxsvf (multiplex only)	percentage 0 -100 [100]	Flag to control the maximum variant frequency values displayed in a multiplex project variant frequency reports and sub-tabs.	X		X	M		M	**Output**: Maximum sample variant frequency % [value]
-mcf	filedesc	Flag to specify a non-default MID config file to be used for decoding multiplex information appearing on the command line.		X	X	A M	addRun addSample	A M	**Read Data Attributes dialog**: MID Config File: Import button [filedesc]
-mi	percentage 1 - 100 [90]	Flag to set the minimum overlap identity parameter. A value of 100 matches only identical sequences. Values below 50 may result in unpredictable behavior, with very low values (*i.e.* 1) essentially matching all reads.	X	X	X	A M		A M	**Computation**: Minimum overlap identity [value]

Option	Parameter [Default]	Description	Persistent	runAssembly	runMapping	runProject	Other CLI Commands	GUI	GUI Parameter
-minlen	bases 15 – 45 [20]	Flag to set the minimum read length used in assembly and mapping computations. Reads of this length will generally be classified as short tag reads, which are treated separately from longer reads in computations (see Sections 1.16.1.10 and 2.18.1.10).	X	X	X	A M		A M	**Input**: Minimum read length [value]
-minsvf (multiplex only)	percentage 0 -100 [0]	Flag to control the minimum variant frequency values displayed in multiplex project variant frequency reports and sub-tabs.	X		X	M		M	**Output**: Minimum sample variant frequency % [value]
-ml	bases ->=1 [40] or percentage 1% – 100% [90%]	Flag to set the minimum overlap length parameter, as either a minimum length in bases or a percentage of read length.	X	X	X	A M		A M	**Computation**: Minimum overlap length [value]
-mrerun (multiplex only)	comma-delimited sample project list or “report”	Flag to re-compute a multiplex project or re-generate the unified multiplex reports.		X	X	A M		A M	**Computation**: [Re-run sample list] **Computation**: [Re-run summary report]
-multi (multiplex only)	none	Flag to specify creation of a multiplex project.	X	X	X		createProject newAssembly newMapping	A M	**New Project dialog**: Create a multiplex project [checked]
-n {-notrim}	none	Flag to turn off default quality and primer trimming of input reads, for example to QC reads.	X	X	X	A M		M	**Input**: Automatic Trimming [unchecked]

Option	Parameter [Default]	Description	Persistent	runAssembly	runMapping	runProject	Other CLI Commands	GUI	GUI Parameter
-nft	none	Flag to append a column to the 454AlignmentInfo.tsv file that reports counts of bases (A,C,T,G,N) and gaps from forward and reverse reads at each position of the multiple alignments. Related options include -info, -infoall, and -noinfo.			X	M		M	**Output**: Alignment info: Nucleotide frequency table [checked]
-nimblegen {-n}	none	Flag to specify reads should be primer-trimmed using the early NimbleGen Sequence Capture protocol's primer sequence .	X		X	M		M	**Computation**: Nimblegen sequence capture [checked]
-no	none	Flag to specify no output of most files. The output will include files associated with the creation of the multiple alignments, but will not include files or metrics involved with contig, consensus or variation information.		X	X	A M		A M	**Output**: Include consensus [unchecked]
-noaccno	none	Flag to deactivate use of an accno renaming file.	X		X	M		M	**Input**: Accno renaming file [empty]
-noace	none	Flag to specify no ace file generation. The default behavior (with no option specified) is to output a single ACE file for small genomes (fewer than 4 million reads and a reference less than 40 Mbp). Related options include -ace, -acedir, -consed, -consed16, and -nobig.	X	X	X	A M		A M	**Output**: Ace Format [No files]
-noannot	none	Flag to disable the automatic reading of annotation file.	X		X	M		M	**Input**: Genome annotation [empty]

Option	Parameter [Default]	Description	Persistent	runAssembly	runMapping	runProject	Other CLI Commands	GUI	GUI Parameter
-nobam	none	Flag to suppress bam file generation. The default behavior (with no option specified) is to output a single BAM file for small genomes (fewer than 4 million reads and a reference less than 40 Mbp). Related options include -bam and –nobig.	X		X	M		M	**Output**: BAM Output [No files]
-nobig	none	Flag to skip output of large files; ACE format files, 454PairAlign.txt, 454AlignmentInfo.tsv, 454TrimStatus.txt, 454ReadStatus.txt		X	X	A M		A M	**Output**: Pairwise alignment [None] **Output**: Ace Format [No files]
-nofe	none	Flag to disable use of exclude filter file.	X	X	X	A M		A M	**Input**: Exclude filter file [empty]
-nofi	none	Flag to disable use of include filter file.	X	X	X	A M		A M	**Input**: Include filter file [empty]
-nohet	none	Flag to disable Heterozygotic mode.	X	X		A		A	**Input**: Heterozygotic mode [unchecked]
-noinfo	none	Flag to suppress output of the 454AlignmentInfo.tsv file. Related options include -info, -infoall, and -nft.	X	X	X	A M		A M	**Output**: Alignment info [No output]
-noisplit (cDNA only)	none	Do not initiate isotig traversal when depth spikes in alignments are found. These spikes will be treated as putative splice events so that traversal along the path will continue producing more isotigs.	X	X		A		A	**Computation**: Isotig depth split [unchecked]

Option	Parameter [Default]	Description	Persistent	runAssembly	runMapping	runProject	Other CLI Commands	GUI	GUI Parameter
-nolarge	none	Flag to disable large genome assembly mode.	X	X		A		A	**Input**: Large or complex genome [unchecked]
-nonimblegen {-non}	none	Flag to turn off NimbleGen mapping mode.	X		X	M		M	**Computation**: Nimblegen sequence capture [unchecked]
-nop	none	Flag to turn off generation of the 454PairAlign.txt file. Related options include -nobig, -pair, and -pairt.	X	X	X	A M		A M	**Output**: Pairwise alignment [None]
-nor	none	Flag to turn off the automatic rescore function for read quality scores.		X	X	A M			
-noreg	none	Flag to turn off sequence region based generation of output.	X			M		M	**Input**; Targeted regions [empty]
-norip	none	Flag to turn off the output each read in a single contig.	X	X		A		A	**Output**: Reads limited to one contig [unchecked]
-nosersvf	none	Flag to hide empty rows in sample variant frequency tables generated by multiplex mapper.	X	X	X	A M			
-nosio	none	Flag to turn off –sio, -siom or –siod. Required to turn off serial I/O between consecutive runs of the same project.	X	X	X			A M	**Computation**: Use serial I/O [unchecked]
-nosnp	none	Flag to disable the automatic reading of known SNP files.	X		X	M		M	**Input**; Known SNP [empty]
-nosrv	none	Flag to disable single read variant output.	X		X	M		M	**Output**: Single read variant [unchecked]

Option	Parameter [Default]	Description	Persistent	runAssembly	runMapping	runProject	Other CLI Commands	GUI	GUI Parameter
-nosv	none	Flag to disable structural variant detection.			X	M			
-notrim {-n}	none	Flag to turn off default quality and primer trimming of input reads, for example to QC reads.	X	X	X	A M		M	**Input**: Automatic Trimming [unchecked]
-novs	none	Flag to turn off a vector screening database used in an earlier part of the computation.	X			A M		A M	**Input**; Screening database [empty]
-novt {-nov}	none	Flag to turn off a vector trimming database FASTA file for cloning vectors, primers, adaptors and other end sequences specified in an earlier part of the computation.	X			A M		A M	**Input**; Trimming database [empty]
-np	filedesc	Flag to specify SFF data in filedesc comes from non-paired end sequencing. Auto-detect will recognize a paired end file by the presence of the paired end linker sequence in at least 25% of the first 500 reads or 25% of all the reads if there are fewer than 500 reads in the file.		X	X		addRun addSample changeRun	A M	**Read Data Attributes dialog**: Treat reads as specified read type [non-paired end]
-nrm	none	Flag to specify 'no removal' of meta data files which are needed for GUI viewing of data.		X	X				
-numn	count of N's any integer [5]	Flag to specify the threshold for number of Ns in a FASTA or FASTQ read (a value of "0" ignores). Reads are first trimmed from both 5' and 3' ends up to the first occurrence of two adjacent called bases. Reads are rejected if they contain X consecutive N's (where X is a parameter >0) or a series of X N's with no more than one intervening called base (where X is a parameter <0).		X	X	A M			
-o	directory	Flag to open specified directory.		X	X				

Option	Parameter [Default]	Description	Persistent	runAssembly	runMapping	runProject	Other CLI Commands	GUI	GUI Parameter
-p	filedesc	Flag to specify SFF data in filedesc comes from paired end sequencing. In the absence of the –p option, Auto-detect will recognize a paired end file by the presence of the paired end linker sequence in at least 25% of the first 500 reads or 25% of all the reads if there are fewer than 500 reads in the file.	X	X	X		addRun addSample changeRun	A M	**Read Data Attributes dialog**: Treat reads as specified read type [paired end]
-pair	none	Flag to output pairwise alignment of short tag reads in txt file. Related options include -nobig, -nop, and -pairt.	X	X	X	A M		A M	**Output**: Pairwise alignment [Simple format]
-pairt {-pt}	none	Flag to output pairwise alignment of short tag reads in tab delimited file. Related options include -nobig, -nop, and -pair.	X	X	X	A M		A M	**Output**: Pairwise alignment [Tabbed format]
-pickb {-pick}	bases	Flag to generate an output file containing a specified number of bases, by randomly 'picking' reads from the input. The argument can be a number, optionally followed by either a 'k' or 'm' to specify thousands or millions of bases, respectively. Note: Reads are not broken to achieve exactly the number of bases specified, so the actual number of bases in the output file may differ slightly from the desired number. Note: This option will re-compute the project using all selected reads (non-incremental mode).		X	X	A M			

Option	Parameter [Default]	Description	Persistent	runAssembly	runMapping	runProject	Other CLI Commands	GUI	GUI Parameter
-pickr	reads	Flag to generate an output file containing a specified number of reads, by randomly 'picking' reads from the input. The argument can be a number, optionally followed by either a 'k' or 'm' to specify thousands or millions of reads, respectively. An optional multiplexing information string can be prepended to each file/data-directory argument. Note: This option will re-compute the project using all selected reads (non-incremental mode).		X	X	A M			
-qo	none	Flag to generate quick output for mapping and assembly. Disables signal distribution computation for calling consensus sequences and can decrease accuracy.		X	X	A M		A M	**Output**: Quick output [checked]
-r	none	Flag to restart the computation (non-multiplex projects, only).				A M		A M	**Computation**: Incremental *de novo* assembler analysis or Incremental reference mapper analysis [checked]
-random	none	Flag to allow GOLDENPATH _random and _hap references to be automatically used.			X		setRef		
-read	filedescr(s)	Flag to separate read files.			X				
-ref	filedescr(s)	Flag to separate reference files.			X				
-reg	accno | regstring | DARFile	Flag to only output specified sequence region of the reference and aligned reads.	X		X	M		M	**Input**: Targeted Regions [accno | regstring | DARFile]

Option	Parameter [Default]	Description	Persistent	runAssembly	runMapping	runProject	Other CLI Commands	GUI	GUI Parameter
-relink	none	Flag to relink file path info, rather than changing pairing info.					changeRef changeRun		
-rescore	none	Flag to rescore an SFF file using the quality score lookup table corresponding to the number of Flows for reads in the SFF file.		X	X	X			
-rip	none	Flag to output each read in only one contig.	X	X		A		A	**Output**: Reads limited to one contig [checked]
-rst	score >=0 [12]	Flag to set the repeat score threshold parameter.	X		X	M		M	**Computation**: Repeat score threshold [value]
-s	bases >=0 [2000]	Flag to specify the minimum scaffold size threshold.		X		A		A	**Output**: Scaffold length threshold [value]
-sc	seeds >=1 [1]	Flag to set the seed count parameter.	X	X	X	A M		A M	**Computation**: Seed count [value]
-scaffold	none	Flag to enable gap-filling when low-quality portions of reads at the ends of scaffolded contigs indicate an overlap. Also fills scaffold gaps when the same short-tandem repeat is found at the ends of contigs leading into either side of the gap.		X		A		A	**Output**: Output scaffolds [checked]
-sersvf	none	Flag to show empty rows in sample variant frequency tables generated by multiplex mapper.	X	X	X	A M			

Option	Parameter [Default]	Description	Persistent	runAssembly	runMapping	runProject	Other CLI Commands	GUI	GUI Parameter
-short	none	Force assembly of reads with a length between the value set by -ml (default 40 bp) and 49 bp, equivalent to the behavior when paired end reads are present. May be useful for hybrid assemblies using short Illumina reads.		X	X	X			
-sio	none	Flag to invoke serial I/O. Use for projects with > 4 million reads.	X	X	X	A M		A M	**Computation**: Use serial I/O [checked]
-siod	none	Flag to invoke an alternate implementation of serial I/O that minimizes the disk space requirement.	X	X	X				
-siom	RAM (GB) 1 – 1000	Flag to invoke serial I/O and specify the size of the memory footprint to be used.	X	X	X				
-sl	bases 6 – 32 [16]	Flag to set the seed length parameter.	X	X	X	A M		A M	**Computation**: Seed Length [value]
-snp	filedesc	Flag for optional file with known SNP information for the reference sequence(s) for linking to identified variations on the HCDiffs GUI tab.	X		X	M		M	**Input**: Known SNP [filedesc]
-ss	bases >=1 [12]	Flag to set the seed step parameter.	X	X	X	A M		A M	**Computation** Seed step [value]
-srv	none	Flag to enable single read variant output.	X		X	M		M	**Output**: Single read variant [checked]

Option	Parameter [Default]	Description	Persistent	runAssembly	runMapping	runProject	Other CLI Commands	GUI	GUI Parameter
-sveg	filedesc	Flag to specify a structural variation excluded genes file, which consists of gene pairs to be excluded from the 454AllFusionVariantTable.txt and 454HCFusionVariantTable.txt files. Each line in the file contains two gene names, separated by a tab.			X	M			
-tr	none	Flag to output 454TrimmedReads.fna & 454TrimmedReads.qual.		X	X	A M		A M	**Output**: Output trimmed reads [checked]
-trim	none	Flag to turn on default quality and primer trimming of input reads.	X	X	X	A M		M	**Input**: Automatic Trimming [checked]
-tsv (multiplex only)	filedesc	Flag to specify the sample association file, which contains sample configuration information for a multiplex project (association of sample name with MIDs and read files).	X	X	X		addSample	A M	**Add sample wizard**: Sample association file [filedesc]
-ud	none	Flag to treat each read separately, with no grouping into duplicates.		X	X	A M		A M	**Computation**: Use duplicate reads [checked]
-urt	none	Extend contigs using the ends (tips) of single reads.		X		A		A	**Computation**: Extend low depth overlap [checked]
-verbose	none	Flag for increasing the amount of information logged to stdout or 454NewblerProgress.txt file (*e.g.* timestamp, machine name, user, CLI options used, *etc.*).	X	X	X		changeRef changeRun		
-vs	fastafile	Flag to specify a vector screening database FASTA file of read sequences that indicate contaminants. Read is screened if it matches completely, partial matches are not screened.	X	X	X	A M		A M	**Input**: Screening database [filedesc]

Option	Parameter [Default]	Description	Persistent	runAssembly	runMapping	runProject	Other CLI Commands	GUI	GUI Parameter
-vt {-v}	fastafile	Flag to specify a vector trimming database FASTA file for cloning vectors, primers, adaptors and other end sequences.	X	X	X	A M		A M	**Input**; Trimming database [filedesc]

Table 6: Mapping and Assembly options. Where appropriate, the Parameter column includes the nature of the parameter, the range of allowed values, and the default value in square brackets. An "X" in the Persistence column indicates that the parameter value is stored with the project, and will be used automatically the next time the project is opened (sometimes referred to as being "sticky"). In the runProject and GUI columns, "A" refers to an assembly project and "M" refers to a mapping project. The GUI Parameter column describes where within the GS *De Novo* Assembler or GS Reference Mapper GUI applications the equivalent option parameters are controlled, most often on one of the Parameters sub-tabs (location displayed in bold). The value of the parameter is displayed in square brackets.

4.3 Mutually Exclusive Options for Assembly and Mapping Commands

Mutually exclusive options of the assembly and mapping CLI commands are:

- -no or –nobig ***with*** –ace, -acedir, -consed, -pair, -pairt
- -ace, -acedir, -noace, -consed
- -ar, -at
- -a #, -g
- -annot, -noannot
- -nimblegen (-n), -nonimblegen (-non)
- -pair, -pairt (-pt)
- -reg #, -noreg
- -sio, -siom, -siod, -nosio
- -snp, -nosnp
- -trim, -notrim
- -v, -nov
- -vs, -novs
- -vt, -novt

4.4 Serial I/O

4.4.1 Overview

The serial I/O options invoke an optimized method of processing the read files during the later phases of the assembly or mapping processes. In the assembler, the order of sequences required in these later phases is determined by the structure of the scaffolds created. In the mapper, the order of sequences required in is the last phase is determined by the order of the references to which the reads map. In both cases, the order is different than the order of the sequences in the input read files. By default, random access I/O is used to obtain the sequences from the read files on disk while computing signals and producing the output files. That is, read operations will be scattered across all of the input read files. For projects using a small number of input read files this is an adequate approach. However, for larger projects (*e.g. de novo* assembly of large genomes) there may be hundreds of read files and tens of millions of sequences. Consequently, the inefficiency of random access file I/O becomes magnified many times and results in prohibitively long execution times for the I/O-intensive, later part of the assembly/mapping operation.

The approach used in serial I/O is to iterate over the scaffolds or references to create a list of the required sequences in the order in which they will be accessed. This list is then used to create a single read file containing all the reads in the order specified by the list. Doing so allows the use of more efficient sequential I/O operations when the sequences are actually read from disk. The costs of this approach are the memory footprint and disk space (up to three times the space for the input read files) required to implement the algorithms used in creating the sequential read file. For sufficiently large projects, these costs are far outweighed by the greater efficiency of the I/O operations enabled by this approach.

4.4.2 Options for Serial I/O

4.4.2.1 The –sio Option

The most basic option is **–sio**. This option invokes the default serial I/O implementation and turns off the other options related to serial I/O (**–siom**, **–siod**) which are discussed below. The memory footprint required for this option is the number of reads in the project (up to one million) times 8 KB. That is, if your project contains less than one million reads the memory footprint will be the number of reads in your project times 8 KB. The maximum memory footprint required will be 8 GB (one million times 8 KB). The disk space required will be three times the space required for the original read files in your project. This is because during the creation of the single sequential read file, temporary files will be created that contain sequences from the original input read files sorted in the order in which they will be required when the sequences are actually read. These temporary files will be merged into a single file. Thus, at one point in the process the input sequence data will be represented three times on the disk: once in the original files, once in the temporary files and once in the sequential read file. At the end of the merge process the temporary files will be deleted.

The sequential read file is named **.SequentialReads** and is located in the output directory for the project. This file is deleted when execution of the project has completed.

4.4.2.2 The –siom Option

The **–siom** option allows the user to invoke serial I/O and specify an allowable memory footprint greater than the baseline requirement (see above). The option takes a numeric argument (in gigabytes) with a minimum of 1 and a maximum of 1000. Otherwise, the behavior is the same as for the default serial I/O functionality. It is the responsibility of the end user to specify a value that corresponds to the amount of free, physical RAM that actually exists on the computer.

4.4.2.3 The –siod Option

The **–siod** option invokes an alternative implementation of the serial I/O functionality that minimizes the disk space requirement. This option takes no argument. This alternative implementation processes the input read files incrementally and creates a smaller sequential read file to be is used to compute signals during the Contig Phase or produce output files during the Output Phase. The temporary files and sequential read file used in each increment are discarded before the next increment. This minimizes both the disk space and memory footprint requirement. The performance of this implementation is comparable to the default serial I/O implementation.

4.4.2.4 The –nosio Option

The options for serial I/O are “sticky”, meaning that they persist between consecutive runs of the same project. To turn off serial I/O when re-running a project, use the **–nosio** option. This option turns off the **–sio**, **–siom**, and **–siod** options.

4.4.3 Guidelines for Use

The serial I./O functionality is intended for use on large projects with many read files (greater than 10 million reads), for which the reduction in the time required to compute signals and create the output files justifies the time and resources required to create the sequential read file. Large genome assembly projects represent the best use of the serial I/O functionality. Sufficient hardware resources (RAM and disk space) must be available. Smaller scale projects may still benefit from using serial I/O, but the potential gains are much less and if these projects are run on platforms lacking sufficient resources, the use of serial I/O may actually degrade performance.

4.5 Multiplex Identifiers

4.5.1 The Use of Multiplex Identifiers in the Data Analysis (MIDs)

The path for any filename or data directory name can be pre-pended with a multiplexing information string, separated from the filename/directory-string by an ”@” sign. Multiplexing information strings should be used when multiple samples have been prepared using different initial “tag” sequences (such as those provided by the GS Multiplex Identifiers (MID) Kits), then mixed together for sequencing. The software uses these tag sequences to segregate the reads from each sample, by matching the initial bases of the reads to the known tag sequences used in the preparation of the libraries. If a multiplex string is given, sfffile will automatically match the 5’ bases of each read against the multiplex sequences, and will only use the reads that match the specified sequences (and reset the trim point of those reads so that the multiplex sequence is trimmed off the output read sequence).

Three examples of multiplex information strings are the following:

- mid2@myreads.sff
- GSMIDs:mid1,mid4,mid8@/home/xxx/morereads.sff
- aattctc/1@1-3,4,6-8:D_2005_01_01_01_01_01_testuser_runImagePipe

The first example gives “myreads.sff” as the input SFF file and tells sfffile to only use reads whose initial 5’ sequence matches the “mid2” MID (as listed in the MID configuration file). The second example tells sfffile to use the file “/home/xxx/morereads.sff” and filter that file, using only the reads starting with the mid1, mid4 and mid8 sequences, as defined in the “GSMIDs” MID set. The third example tells sfffile to read regions 1, 2, 3, 4, 6, 7 and 8 of the “D_2005_...” sequencing run, but only use the reads whose 5’ end matches the DNA sequence “AATTCTC” with 1 error or less. (Note that regions only apply to data produced by a GS FLX+ Instrument).

The format of the multiplexing information string is the following:

```
[groupname:]mid[/#](,mid[/#]…)@
```

where

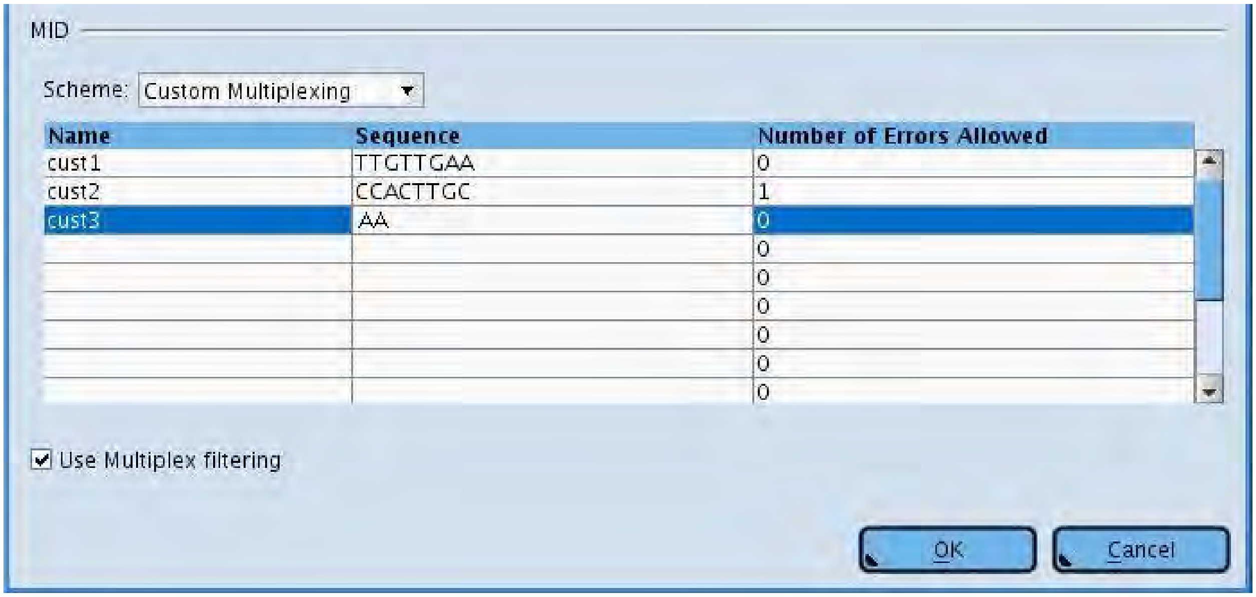
- the "`groupname`" is an optional name of an MID set found in the MID configuration file and can be given in uppercase or lowercase letters (the matching is case-insensitive)
- each "`mid`" is either an MID name found in the configuration file or a DNA sequence
- each "#" is an optional allowed number of errors

No spaces are allowed in the multiplexing information string, and no colons, slashes or "@" signs are allowed in the MID set names or MID names.

The default MID configuration file is MIDConfig.parse (see Glossary). This file is read by sfffile and used to match MID set names and MID names with their multiplexing information. Users can edit this file to add their own MID sets (following the format and syntax described in the file), or can copy this file locally and create their own custom MID configuration file and then use the '-mcf filedesc' option to specify the newly created MID configuration file.

The list of MID groups provided in the Scheme drop down menu is read from the selected MID configuration file (by default, this is the MIDConfig.parse file). For pre-existing MID schemes, the names and sequences of the MIDs, and the number of errors allowed, cannot be changed in the Select GS Read Data window. To modify these values, or to add new user-specified MID schemes, edit the MIDConfig.parse file using any text editor. See section 4.5.2 to view the default MIDConfig.parse file and for a description of how to edit it.

You can also create new MIDs directly in the Select GS Read Data window, by selecting the "Custom Multiplexing" option. In this case, a table will appear and new MID names, sequences, and error limits can be typed into the table (see Figure 127). However, these names, sequences and error limits will only apply to the sff files being imported.



Scheme: Custom Multiplexing

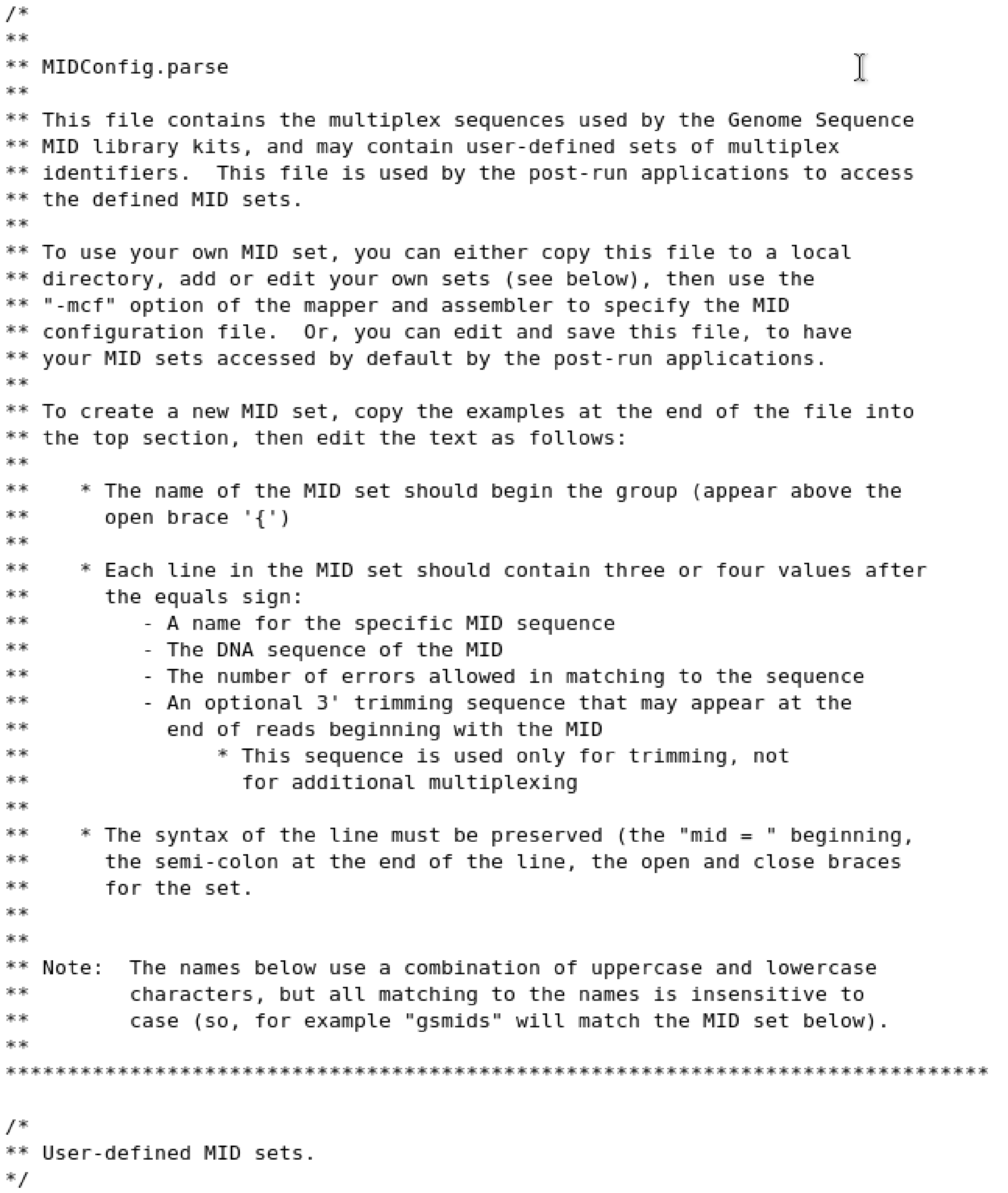
Name	Sequence	Number of Errors Allowed
cust1	TTGTTGAA	0
cust2	CCACTTGC	1
cust3	AA	0
		0
		0
		0
		0
		0
		0

☑ Use Multiplex filtering

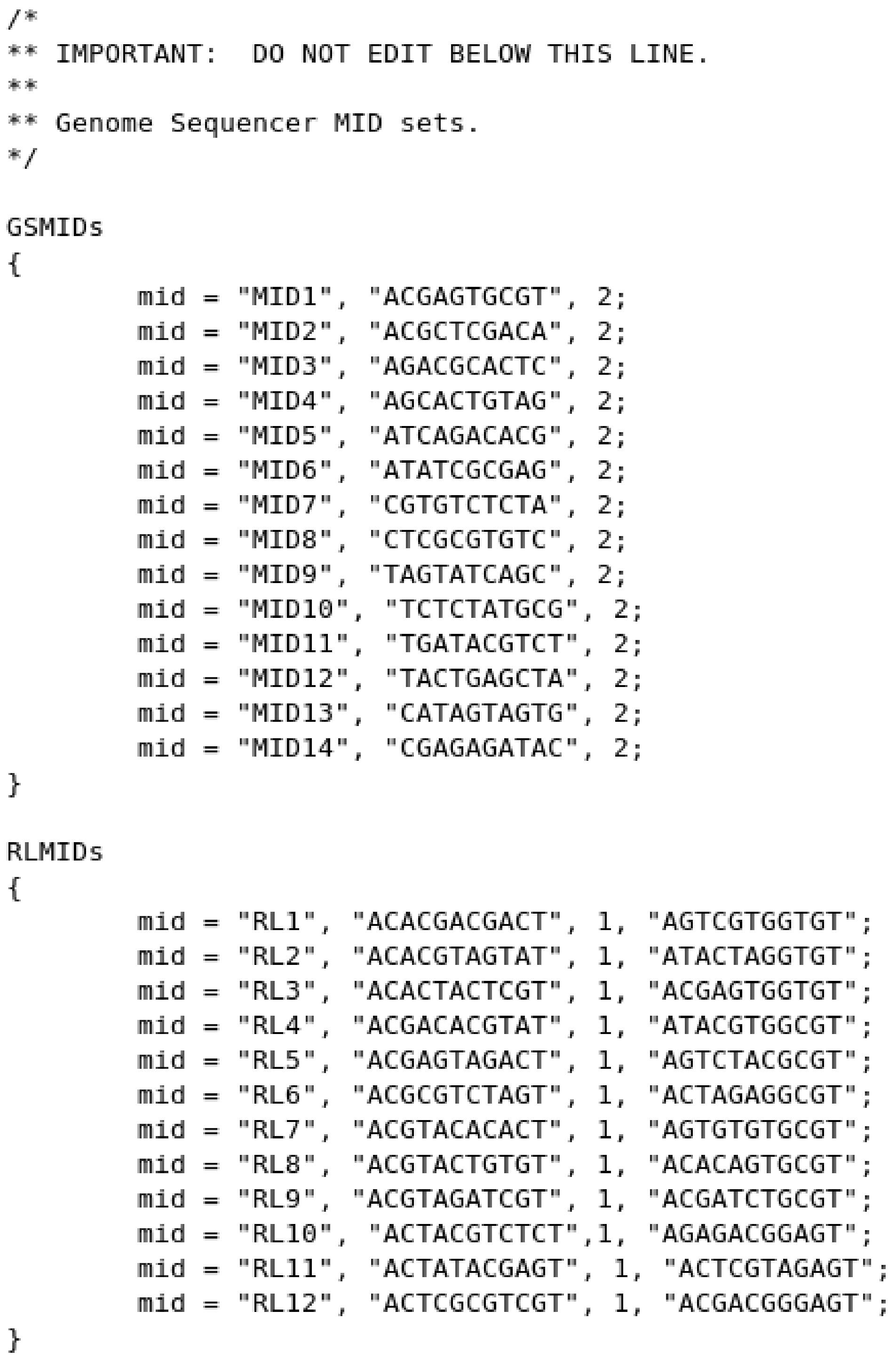
Figure 127: GS Read Import Dialog Rapid Library MID Support.

There is no formal limit to the number or length of custom MIDs that one can define in an Assembly or Mapping project. However, efficient de-multiplexing requires that the MID sequences be divergent enough that the software will be able to differentiate them from one another (within the "number of errors allowed" specified). The 454 Sequencing system software comes with 2 MID schemes, GSMIDs and RLMIDs. 14 MIDs (of which MIDs 1 to 12 match the "MID Adaptors" available in kit form) are listed for GSMIDs. 12 MIDs are available as RLMIDs.

4.5.2 The MIDConfig.parse File



```
/*
**
** MIDConfig.parse
**
** This file contains the multiplex sequences used by the Genome Sequence
** MID library kits, and may contain user-defined sets of multiplex
** identifiers.  This file is used by the post-run applications to access
** the defined MID sets.
**
** To use your own MID set, you can either copy this file to a local
** directory, add or edit your own sets (see below), then use the
** "-mcf" option of the mapper and assembler to specify the MID
** configuration file.  Or, you can edit and save this file, to have
** your MID sets accessed by default by the post-run applications.
**
** To create a new MID set, copy the examples at the end of the file into
** the top section, then edit the text as follows:
**
**    * The name of the MID set should begin the group (appear above the
**      open brace '{')
**
**    * Each line in the MID set should contain three or four values after
**      the equals sign:
**         - A name for the specific MID sequence
**         - The DNA sequence of the MID
**         - The number of errors allowed in matching to the sequence
**         - An optional 3' trimming sequence that may appear at the
**           end of reads beginning with the MID
**               * This sequence is used only for trimming, not
**                 for additional multiplexing
**
**    * The syntax of the line must be preserved (the "mid = " beginning,
**      the semi-colon at the end of the line, the open and close braces
**      for the set.
**
**
** Note:  The names below use a combination of uppercase and lowercase
**        characters, but all matching to the names is insensitive to
**        case (so, for example "gsmids" will match the MID set below).
**
*******************************************************************************

/*
** User-defined MID sets.
*/
```



```
/*
** IMPORTANT:  DO NOT EDIT BELOW THIS LINE.
**
** Genome Sequencer MID sets.
*/

GSMIDs
{
        mid = "MID1", "ACGAGTGCGT", 2;
        mid = "MID2", "ACGCTCGACA", 2;
        mid = "MID3", "AGACGCACTC", 2;
        mid = "MID4", "AGCACTGTAG", 2;
        mid = "MID5", "ATCAGACACG", 2;
        mid = "MID6", "ATATCGCGAG", 2;
        mid = "MID7", "CGTGTCTCTA", 2;
        mid = "MID8", "CTCGCGTGTC", 2;
        mid = "MID9", "TAGTATCAGC", 2;
        mid = "MID10", "TCTCTATGCG", 2;
        mid = "MID11", "TGATACGTCT", 2;
        mid = "MID12", "TACTGAGCTA", 2;
        mid = "MID13", "CATAGTAGTG", 2;
        mid = "MID14", "CGAGAGATAC", 2;
}

RLMIDs
{
        mid = "RL1", "ACACGACGACT", 1, "AGTCGTGGTGT";
        mid = "RL2", "ACACGTAGTAT", 1, "ATACTAGGTGT";
        mid = "RL3", "ACACTACTCGT", 1, "ACGAGTGGTGT";
        mid = "RL4", "ACGACACGTAT", 1, "ATACGTGGCGT";
        mid = "RL5", "ACGAGTAGACT", 1, "AGTCTACGCGT";
        mid = "RL6", "ACGCGTCTAGT", 1, "ACTAGAGGCGT";
        mid = "RL7", "ACGTACACACT", 1, "AGTGTGTGCGT";
        mid = "RL8", "ACGTACTGTGT", 1, "ACACAGTGCGT";
        mid = "RL9", "ACGTAGATCGT", 1, "ACGATCTGCGT";
        mid = "RL10", "ACTACGTCTCT",1, "AGAGACGGAGT";
        mid = "RL11", "ACTATACGAGT", 1, "ACTCGTAGAGT";
        mid = "RL12", "ACTCGCGTCGT", 1, "ACGACGGGAGT";
}
```

4.6 Paired End Libraries in the 454 Sequencing System

4.6.1 Paired End Read Naming Convention

In traditional Sanger paired end sequencing, the ends of a fragment are sequenced using Forward and Reverse Primers (see Figure 128). In the 454 Sequencing system, by contrast, paired end sequencing is performed by first circularizing the fragment using a linker to join the 2 ends of the fragment of sample DNA. The circularized fragment is then nebulized, and the nebulized fragments containing the linker are then sequenced (see Figure 129).

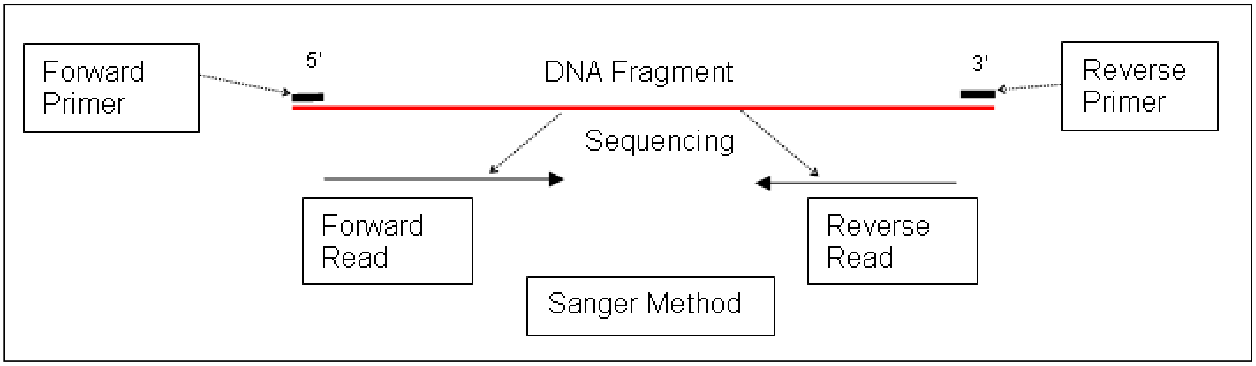


Figure 128: Sanger paired end reads

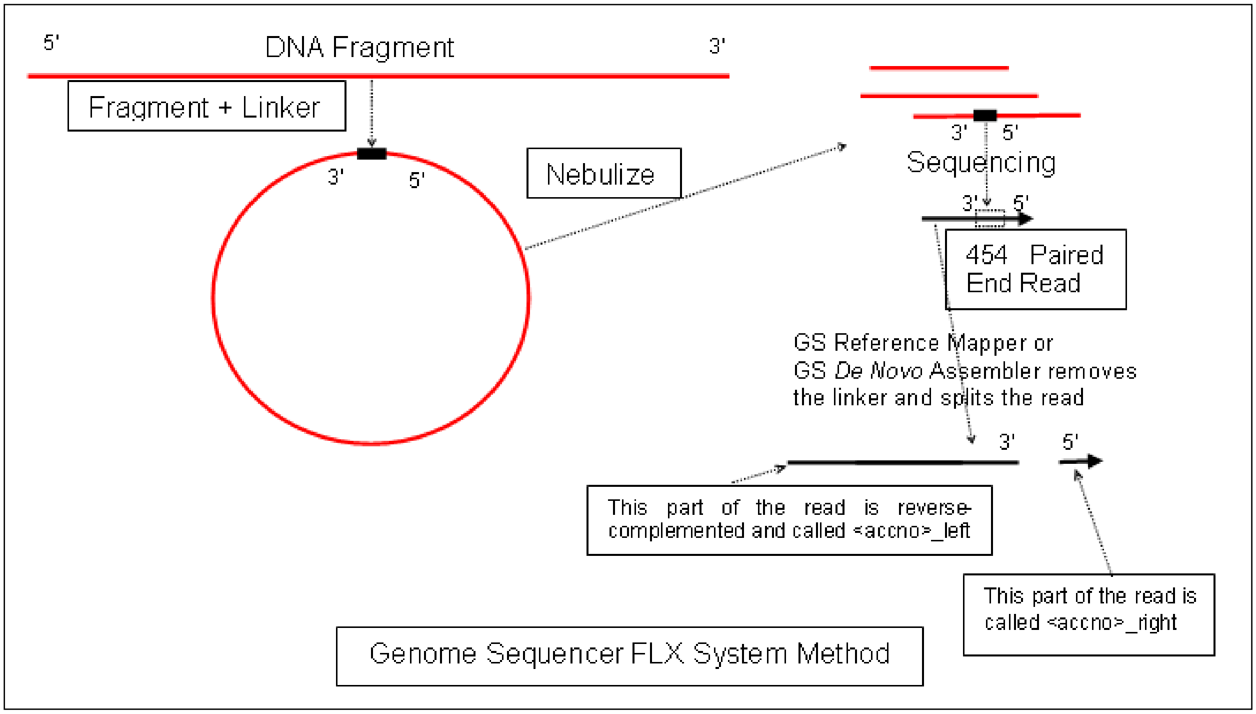

5'
DNA Fragment
3'
Fragment + Linker
3'
5'
Nebulize
3'
5'
Sequencing
3'
5'
454 Paired End Read
GS Reference Mapper or
GS De Novo Assembler removes
the linker and splits the read
3'
5'
This part of the read is reverse-complemented and called <accno>_left
This part of the read is called <accno>_right
Genome Sequencer FLX System Method

Figure 129: 454 Sequencing system paired end reads.

The GS *De Novo* Assembler or GS Reference Mapper application takes paired end reads that have been basecalled, removes the linker sequence, and creates two “half” reads for computation and reporting purposes. The half of the read corresponding to the reverse read in the Sanger method (coming from the 3'-end of the original linear fragment) is reverse-complemented to match the orientation one would obtain by extending the Reverse primer in the Sanger method; this half read is labeled with the original read's accno and given the extension '_left'. The half of the read corresponding to the forward read in the Sanger method is left in its original orientation and labeled with the original read's accno and given the extension '_right'.

4.6.2 Paired End Reads Shorter Than 50 bp

Reads shorter than 50 bp are handled differently than longer reads (see Sections 1.16.1.10 and 2.18.1.10 for details). When paired end reads are detected, processing is automatically equivalent to using the -short option, without having to explicitly specify the option. In an assembly project, all reads of at least the minimum overlap length (default 40 bp) are included in the process of generating contig consensus sequences, including those between 40 and 49 bp that normally would not be used. In a mapping project, all reads that are at least as long as the sum of the seed length plus seed step (default 16 + 12 = 28 bp) are mapped against the reference using the standard overlap detection parameters. Under both circumstances, shorter reads are treated as short tag reads.

4.6.3 Paired End Library Span Estimation

Estimates of the distance spanned by paired end reads in a library are made when at least 8 consistent mate pairs are found that align to the same contig or scaffold. Both halves of a paired end read must align to the same contig with the expected directionality (the read halves 3' ends point toward each other, after reverse-complementation of the left half). Statistics for the distance between mated pairs are kept for each library. As additional scaffolds are formed, statistics for additional paired end reads become available and the library span is re-estimated. Paired End reads whose halves are too far away from the mean of the distribution and those whose halves don't have the expected relative orientation are excluded from the span distance calculation. The estimate is less robust when either little paired end information for a library is available or when very few contigs are significantly longer than the actual library span (in the latter case, the estimated span may be significantly lower than the actual span).

4.6.4 True Pairs and False Pairs

"True Pairs" follow the normal rules for paired end reads, *i.e.* they are within the expected library distance from each other, map to the same reference sequence or align within the same assembled contig or scaffold, and map or align in opposite orientations.

False Pairs are paired end reads whose ends either:

- map to different reference sequences or align in contigs that are in different scaffolds
- map to locations on a single reference sequence or assembled contig with a distance outside the expected library span
- have an unexpected relative orientation

When there are too few paired end reads to obtain a robust estimate of the paired end library span, assembled (but not mapped) reads are classified as NoThreshold.

4.7 Trimming and Screening Files

Trimming and screening files use the same format (FASTA) and can even contain the same information. Each entry is simply a unique description line followed by one or more lines of sequence text. Only upper- and lower-case A, C, G, T, and N characters should be used. The N's are expected to correspond to low quality sequence and will NOT be considered a match to any read character except an N. Do not use X's or N's as masking characters. An example follows:

```
>VectorSeq1
TCCGATCGTACGATGATCGATCGATCGGATCGATCGAT
GGCTACTTAGGGCTATAAAACCCATG
>VectorSeq2
GGCTAGATTATTAGCCCCTAGATAAACCTTTTTAGCTAG
TCGATCGGATCGATCGATCATG
```

4.8 Annotation File Formats

4.8.1 GFF File Format

The GFF file must meet the following requirements:

- fields must be tab-delimited
- each row in the file must contain 8 or 9 fields
- the first field must contain a chromosome (*e.g.* chr13) or an accession number (accno) of a reference contig already added to this project.
- the second field must contain the string "NGS"
- the third field must contain the string "primary_target_region"
- the fourth and fifth fields must contain integer values that represent the start and end positions of a target region.
- Fields 6 through 9 are ignored

4.8.2 refGene.txt File Format

The GoldenPath file refGene.txt is a tab-delimited text file consisting of 16 fields. All fields are required, including the bin field. The value of the bin field is ignored, but must be present.

Field	Description
bin	Indexing field
name	Name of gene
chrom	Reference sequence chromosome or scaffold
strand	+ or - for strand
txStart	Transcription start position
txEnd	Transcription end position
cdsStart	Coding region start position
cdsEnd	Coding region end position
exonCount	Number of exons
exonStarts	Comma-separated list of exon start positions
exonEnds	Comma-separated list of exon end positions
id	Unique identifier
name2	Alternate name
cdsStartStat	'none', 'unk', 'incmpl' or 'cmpl'
cdsEndStat	'none', 'unk', 'incmpl' or 'cmpl'
exonFrames	Exon frame offsets {0,1,2}, or {-1} if no frame for exon

4.8.3 snp130.txt File Format

The GoldenPath file snp130.txt is a tab-delimited text file consisting of 18 fields. All fields are required, including the bin field. The value of the bin field is ignored, but must be present.

Field	Description
bin	Indexing field
chrom	Reference sequence chromosome or scaffold
chromStart	Start position in chrom
chromEnd	End position in chrom
name	Reference SNP identifier or Affy SNP name
score	Not used
strand	+ or - for strand
refNCBI	Reference genomic from dbSNP
refUCSC	Reference genomic from nib lookup
observed	The sequences of the observed alleles from rs-fasta files
molType	'unknown', 'genomic' or 'cDNA'
class	'unknown', 'single','in-del', 'het','microsatellite', 'named', 'mixed', 'mnp', 'insertion' or 'deletion'
valid	'unknown', 'by-cluster', 'by-frequency', 'by-submitter', 'by-2hit-2allele' or 'by-hapmap'
avHet	The average heterozygosity from all observations
avHetSE	The Standard Error for the average heterozygosity
func	The functional category of the SNP (coding-synon, coding-nonsynon, intron, *etc.*)
locType	'range', 'exact', 'between', 'rangeInsertion', 'rangeSubstitution' or 'rangeDeletion'
weight	The quality of the alignment

4.9 Accno Renaming File

Scenario 1: Mapping a transcript to a gene name and adding an optional description.

Example of when this would be used: When there is no annotation file but the reference file contains gene identifiers (gene=geneName) in the header line.

YAL001C TFC3 "Largest of six subunits of the RNA polymerase III transcription initiation factor complex (TFIIIC); part of the TauB domain of TFIIIC that binds DNA at the BoxB promoter sites of tRNA and similar genes; cooperates with Tfc6p in DNA binding"

Explanation: YAL001C is the transcript name. TFC3 is the gene name. The quoted text is the description (in this case used for both the transcript and the gene).

Scenario 2: Adding a transcript level description.

Example of when this would be used: When there is no annotation file or the annotation file does not contain a description for one or more transcripts.

YAL011W YAL011W "Protein of unknown function, component of the SWR1 complex, which exchanges histone variant H2AZ (Htz1p) for chromatin-bound histone H2A; required for formation of nuclear-associated array of smooth endoplasmic reticulum known as karmellae"

Explanation: YAL011W is the transcript name. The quoted text is the transcript description. In this case we do not want or need to map the transcript name to a gene name.

Scenario 3: Adding a gene level description.

Example of when this would be used: When there is no annotation file or the annotation file does not contain a description for one or more genes.

CYS3 CYS3 "Cystathionine gamma-lyase, catalyzes one of the two reactions involved in the transsulfuration pathway that yields cysteine from homocysteine with the intermediary formation of cystathionine"

Explanation: CYS3 is the gene name. The quoted text is the gene description. In this case we do not want or need to rename the gene.

Scenario 4: Renaming a gene.

Example of when this would be used: A cDNA Assembly is the reference and there is external data available to map isogroup names to genes.

isogroup00001 geneName geneDescription (optional)

Explanation: isogroup00001 is the reference transcript name. geneName is the name of a gene that has been determined to correspond to the reference transcript. The geneDescription is optional.

4.10 The –urt option: Use Read Tips

The "-urt" option can be used to improve contigging in low depth portions of assemblies. While designed primarily to help recover more complete representations for rare transcripts in cDNA assemblies, genomic assemblies may also benefit from this option since it helps extending contigs to the overhanging tips of the reads at the end of multiple alignments.

The -urt option performs the following 2 operations:

- When a single read is found to overhang past the end of a contig, the assembler tries to extend the contig to the "tip" (end) of the read which extends unaligned at the end of the contig (Figure 130 A and B).
- It calls contigs through regions of barely overlapping reads, where read depth is 1 (Figure 130 C).

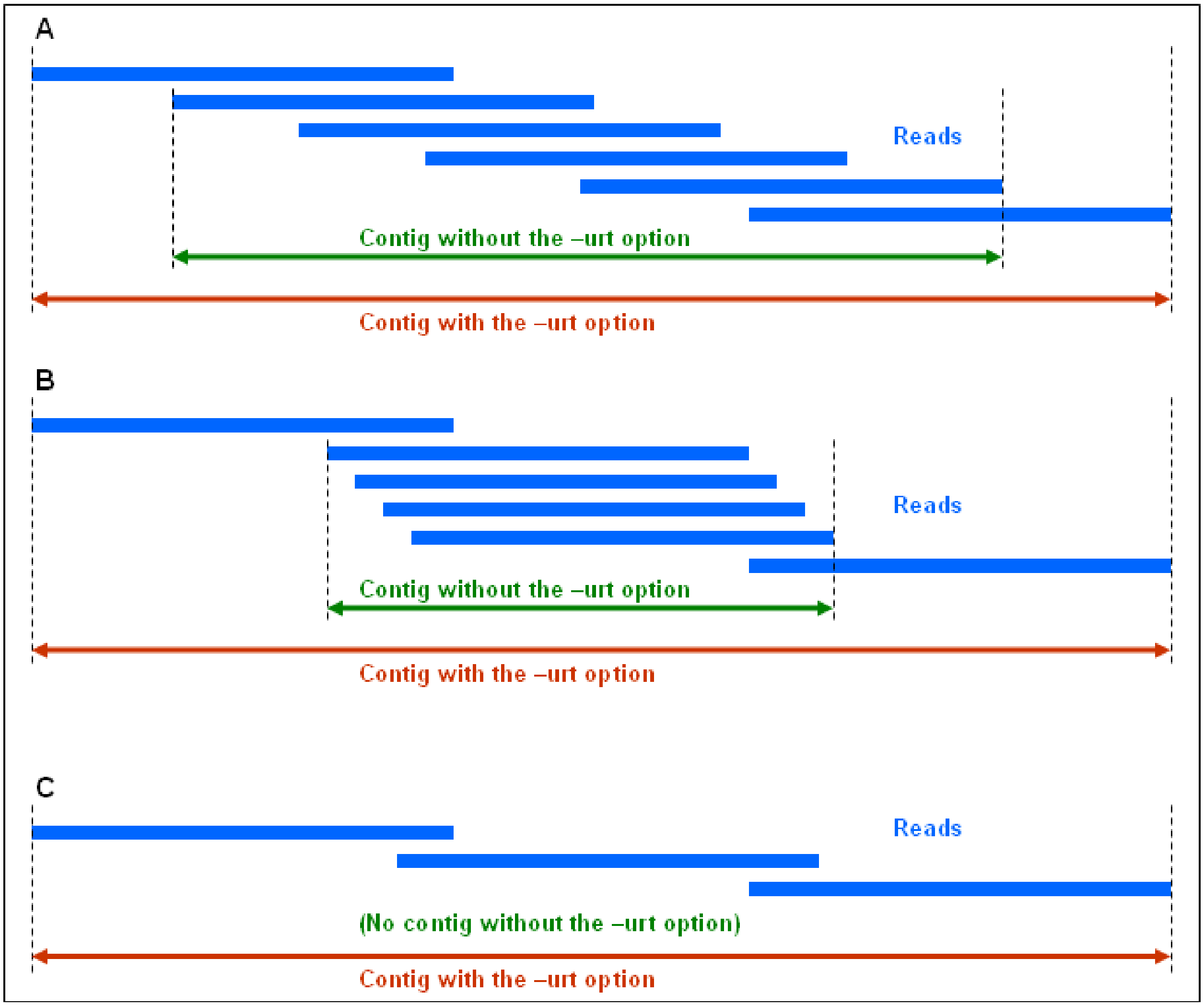

A
Reads
Contig without the –urt option
Contig with the –urt option
B
Reads
Contig without the –urt option
Contig with the –urt option
C
Reads
(No contig without the –urt option)
Contig with the –urt option

Figure 130: Effects of the –urt option on assemblies.

For cDNA assemblies, use of this option usually results in a reduction in the number of singletons while increasing the number of fully and partially aligned reads. Also, the number of isogroups and isotigs generally increases because –urt generates additional contigs from reads previously declared to be singletons. For genome that contain closely packed genes, however, this option can cause isotig and isogroup fusions, so the option should be used with caution in such cases. Also, with the lack of coverage for certain loci, this option can cause isogroup fragmentation: while some of the reads from a low coverage locus are assembled, assemblies may be broken into multiple isogroups because there are not enough reads to cover the whole locus (typically with each isogroup containing a single low-depth contig), thereby increasing overall fragmentation. Another side effect of using this option on data sets with incomplete coverage is a drop in average contig length and N50 statistics (again, due to the creation of new, shorter, low-depth contigs which cover only portions of certain loci).

Note that the -urt option is not recorded and/or persisted in the project's xml file. Therefore, it must be specified on the command line or selected in the GUI (Extend low depth overlaps check box) each time the computation is run.

4.11 Assembling with FASTA Reads Obtained Using the Sanger Sequencing Method

Introducing reads obtained using the Sanger sequencing method (Sanger reads) into an assembly or mapping can be more complicated than including 454 Sequencing reads, because paired end information, trimming information and vector (and other non-sample) information is often not directly associated with the read sequences. To utilize the full capabilities of the GS *De Novo* Assembler or the GS Reference Mapper, the FASTA files of input Sanger reads must be prepared so that they provide the software with the paired end, trimming and non-sample information (see the Overview for a description of FASTQ files, which can simplify this process). This section describes how the data analysis software applications may use Sanger read data FASTA files.

4.11.1 Simple Sanger Reads

In the simplest case, the data analysis software applications can take FASTA files and assemble or map the sequences they contain directly, without any preparation. In this case, it treats all the reads present in the FASTA files as single ended, shotgun reads, and assumes that the sequence has been trimmed for quality and for vector, primer, adapter, linker, *etc.* sequences (all non-sample sequence). The GS *De Novo* Assembler or the GS Reference Mapper will use all the sequence data present in the file, and produce or map contigs based on those sequences.

4.11.2 Sanger Reads with available Quality Scores

If there are quality score files associated with the FASTA files, the data analysis software applications will automatically read the quality scores for the reads, and use them in the assembly or mapping and consensus calling. The GS *De Novo* Assembler or the GS Reference Mapper will look for a file that begins with the same file name or file prefix as the FASTA file but ending with ".qual".

4.11.3 Sanger Read Annotations

Additional information about the Sanger reads can be specified using "*name=value*" annotation strings on the description line of each sequence in the FASTA file. The data analysis software applications look for and uses the following annotation strings:

1. **template** – the paired end template string for this read (paired end reads are matched by having the same template string)
2. **dir** – values "F", "R", "fwd" or "rev" giving the direction of the paired end read
3. **library** – the name of the library that generated this paired end read (all paired end reads are grouped by library name for the determination of expected pair distance)
4. **trim** – the trimmed region of the sequence, given as "#-#"
5. **scf** – the path or "command string" to use to access the SCF file for the read
6. **phd** – the path or "command string" to use to access the PHD file for the read

For example, the description line:

```
>DJS045A03F template=DJS054A03 dir=F library=DJS045 trim=12-543
```

tells the data analysis software that this read (accession number "DJS045A03F") is a paired end read whose template is "DJS045A03", is a forward read from library "DJS045", and bases 12-543 should be used in the assembly.

The data analysis software looks for the six "*name=*" strings on the description line, and then takes the text from the "=" to the next whitespace character as the "*value*" of the annotation string. So, other text besides annotation strings can appear on the description line, and no whitespace may appear in the value of an annotation string.

4.11.3.1 The "template", "dir", and " library" Annotations

The first three annotation strings, template, dir and library, are used to specify the paired end information for a read; only reads with template, dir and library annotations will be treated as paired end reads. The data analysis software performs exact string matching of the template and library annotations to determine which reads come from the same template or library. Multiple reads may have the same template, dir and library information: they all will be assembled or mapped but only the best aligned sequence (by aligned length or number of differences from the consensus) will be used further. There is no requirement that the accession number for the read encode the paired end information, and the data analysis software will not try to "parse" the accession number for any information. However, if you use a naming convention for encoding the paired end information in the accession numbers for reads, the fnafile command can be very useful for translating that encoded information into the annotation strings that the data analysis software can then read. (See Section 3.4 for a description of this use.)

4.11.3.2 The "trim" Annotation

The trim annotation describes the region of the sequence that the data analysis software should use for assembly or mapping (with the assumption that the rest of the sequence is non-sample or low quality). For a read 800 bases in length and a trim annotation string "trim=12-543", for example, the assembler will ignore bases 1-11 and 544-800, and only use bases 12-543 in the assembly or mapping. If either number in the trim annotation string is 0, the beginning or end of the read will be treated as the trimming point (*i.e.*, for the same example read as above, "trim=12-0" would tell the assembler to use bases 12-800 of the read in the assembly).

This trim annotation should combine the results of all of the sources of trimming that may occur (low quality, vector, primer, adapter, linker, *etc.*), so that the data analysis software is given just the sequence region that represents the bases to be included in the assembly or mapping. The "-v" option can be used with the runAssembly, runMapping and runProject commands to specify a FASTA file containing sequences to be trimmed: each read will then be screened against this database, and the ends of reads that match the sequences included will be trimmed off.

4.11.3.3 The "scf" and "phd" Annotations

The scf and phd annotations provide a link back to the "trace" information for the read, and are only used when generating the full consed directory structure (when the -consed option is used). The value string for scf or phd annotations can take one of two forms, either as a simple path to the SCF or PHD file, or as a "command string" to be executed to access the SCF or PHD file contents. If the value string does not begin with "cmd:", it is treated as a path. If the value string begins with "cmd:", it should have the format "cmd:*commandline*", where any whitespace in the *commandline* portion of the command should be replaced with colons.

Any command specified in the scf or phd annotation strings must, when executed, write the PHD or SCF contents to standard output. The command's standard output is read, and those bytes are written or processed as needed, by the assembler/mapper and/or the sff2scf command (the programs in the current software that require access to the PHD or SCF contents).

4.11.4 Recommended Method for Including Sanger Reads

The best method for including Sanger reads into an assembly is to first run phred with vector screening (or an equivalent basecaller that generates a FASTA file with vector sequence marked as 'X' or 'N', plus a quality score file). The ".fasta.screen" and "fasta.screen.qual" files produced by phred contain the screening and trimming information needed by the GS *De Novo* Assembler or the GS Reference Mapper (and it is setup to properly process those reads and the marking and quality information).

There are two ways to include the pairing information for the reads, each of which requires some Perl scripting, because the paired end information is typically encoded in the accession numbers of the reads. The simplest way is to write a Perl script like the one below. This script is designed to:

- handle a paired read naming convention where there is only one paired end library;
- where the forward and reverse reads have suffixes ".f1" or ".r1"; and
- where the paired end reads can be distinguished from follow-up finishing reads by having only alphanumeric characters before the ".f1" or ".r1"):

```
#!/usr/bin/perl

if ($ARGV[0] =~ /^(\w*)\.([rf])1$/) {

   print "template=$1 dir=$2 library=pairlib\n";

}
```

where "`$ARGV[0]`" will be the accession number for the read, when used in the procedure below.

The regular expression in this script states "`^(\w*)\.([rf])1$`" matches a word of any length, "`\w*`", followed by a period, "`\.`", followed by either an '`r`' or an '`f`', followed by a '`1`'. The parentheses are there to then extract the characters that matched the word and matched the '`r`' or '`f`' and place them into the "`$1`" and "`$2`" parameters (so that they can be used in the print statement on the next line). For a local naming convention, customize the Perl regular expression to extract out the template, library and forward/reverse direction strings from the accession, and output them appropriately (see Section 4.11.3) for a description of this output.

If this script is named "accnoPair", then the following procedure can be used to incorporate Sanger reads:

1. Run phred with vector screening to produce the "fasta.screen" file (plus fasta.screen.qual).
2. Run the command "fnafile -o sanger.fna -ac accnoPair fasta.screen" to attach the paired end information to each read, where fnafile calls the accnoPair program for each read, then adds the output of that command to the read's description line.
3. Give sanger.fna to the assembler or mapper, where it will read and use the pairing information.

The one drawback to this method is that the execution of the Perl script for each read causes the fnafile program to take a couple of minutes to convert 15,000-20,000 reads. An advanced script and slightly different procedure that runs the attachments faster can be used:

If the following Perl script is generated (called "detPairs"):

```
#!/usr/bin/perl

die "Usage: detPairs screenfile\n" unless defined($ARGV[0]); open(FILE,
$ARGV[0]) or die "Error:  Unable to open file: $ARGV[0]\n";

while (<FILE>) {

    if (/^\>(\S*)/) {

        my $accno = $1;

        if ($accno =~ /^(\w*)\.([rf])1$/) {

            print "$accno template=$1 dir=$2 library=pairlib\n";

        }

    }

}
```

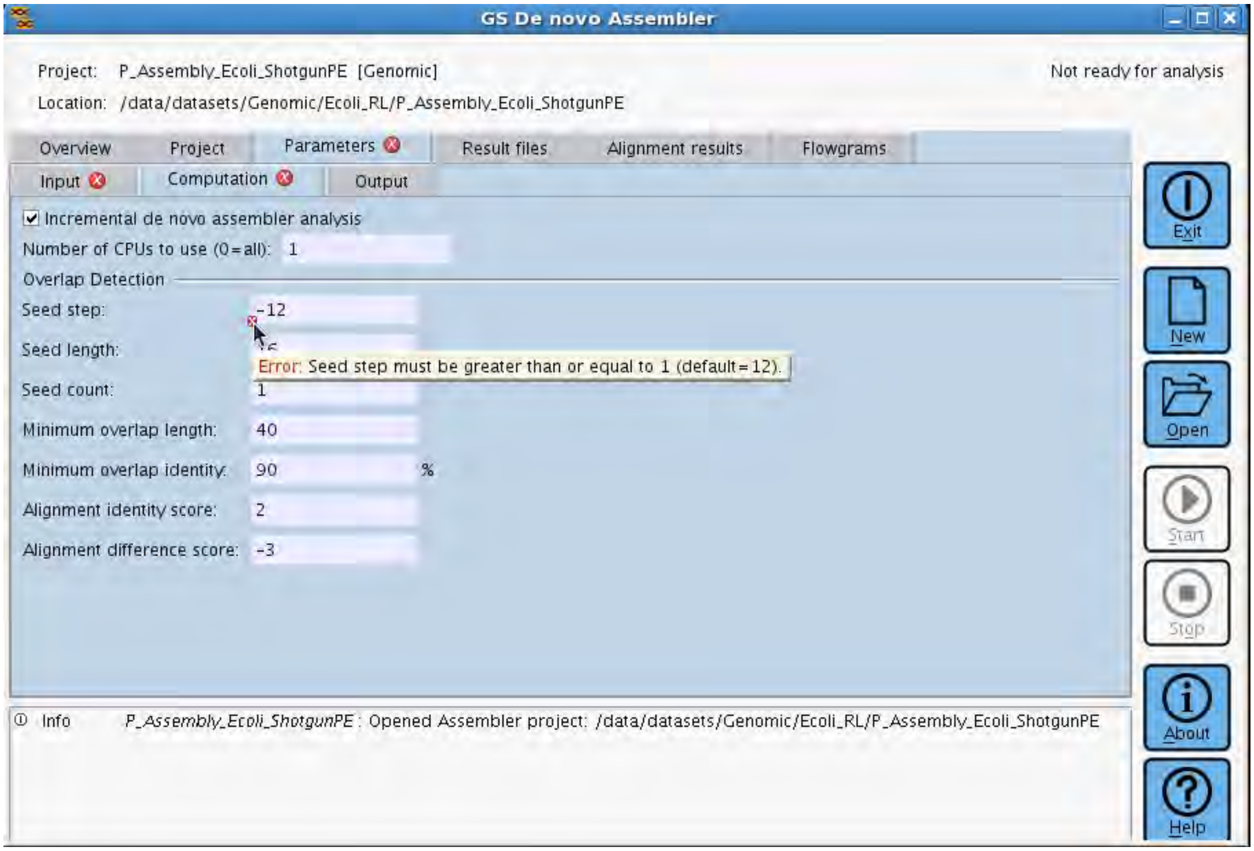
then the following procedure can be used to incorporate Sanger reads:

1. Run phred with vector screening to get the fasta.screen file.
2. Run the command "detPairs fasta.screen > sanger.acc" to get a file with the pairing information.
3. Run the command "fnafile -o sanger.fna -af sanger.acc fasta.screen".
4. Give the sanger.fna file to the assembler or mapper, where it will read and use the pairing information.

For both of these scripts, see Section 3.4 for a description of the command line structure and arguments of fnafile.

4.12 Project Error Indicators

Descriptive warning and error messages are available on the Parameters and Project tabs. Errors are indicated on the tabs and sub-tabs as red circles with and 'x' both on the tab headings and on the field on the tab producing the error. Placing the mouse over the field indicated reveals a descriptive tooltip of the error including, when available, the default value for the field. A list of the errors to be addressed can also be viewed *via* a tooltip when hovering the mouse over the status message in the upper right hand corner of the main application window. When the errors have been addressed, the status message will read 'Ready for Analysis'. Some examples are shown in the figures below.



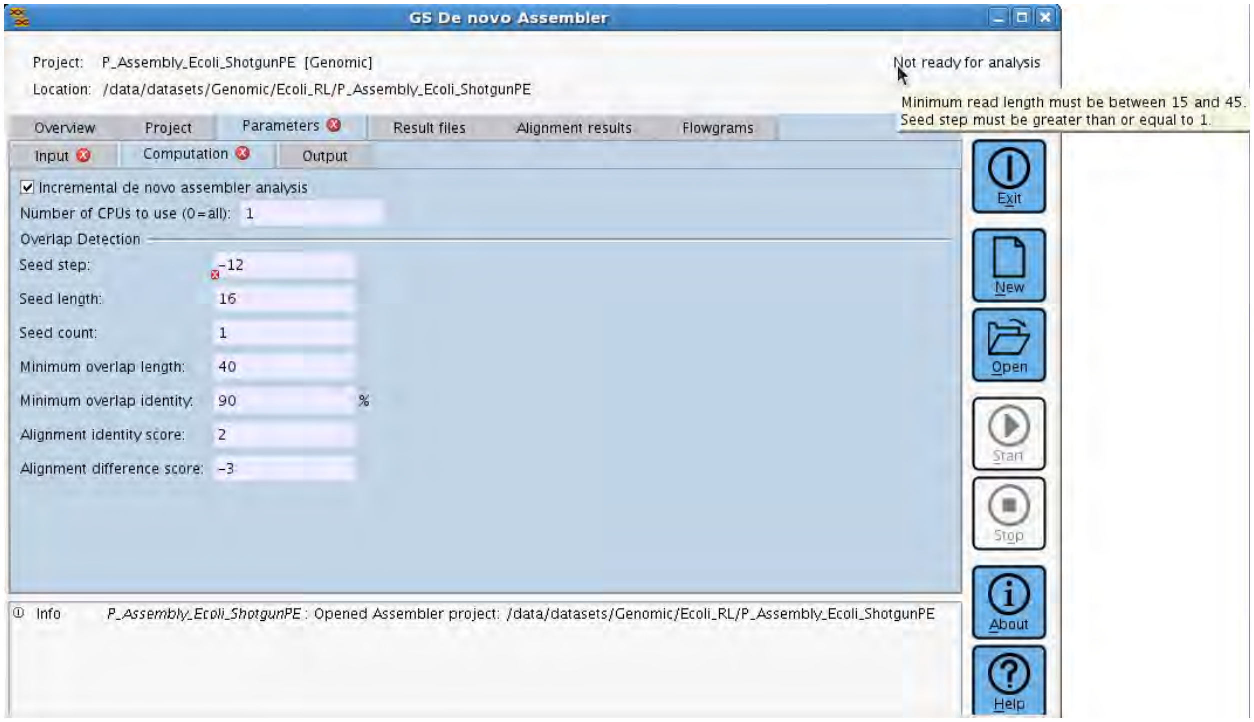
Project: P_Assembly_Ecoli_ShotgunPE [Genomic]
Location: /data/datasets/Genomic/Ecoli_RL/P_Assembly_Ecoli_ShotgunPE

Not ready for analysis

☑ Incremental de novo assembler analysis
Number of CPUs to use (0=all): 1
Overlap Detection
Seed step: -12
Seed length: 16
Error: Seed step must be greater than or equal to 1 (default=12).
Seed count: 1
Minimum overlap length: 40
Minimum overlap identity: 90 %
Alignment identity score: 2
Alignment difference score: -3

Info P_Assembly_Ecoli_ShotgunPE : Opened Assembler project: /data/datasets/Genomic/Ecoli_RL/P_Assembly_Ecoli_ShotgunPE

Figure 131: Error messages on the Tab headings and associated tab fields.



GS De novo Assembler

Project: P_Assembly_Ecoli_ShotgunPE [Genomic]

Location: /data/datasets/Genomic/Ecoli_RL/P_Assembly_Ecoli_ShotgunPE

Not ready for analysis

Minimum read length must be between 15 and 45.
Seed step must be greater than or equal to 1.

☑ Incremental de novo assembler analysis

Number of CPUs to use (0=all): 1

Overlap Detection

Seed step: -12

Seed length: 16

Seed count: 1

Minimum overlap length: 40

Minimum overlap identity: 90 %

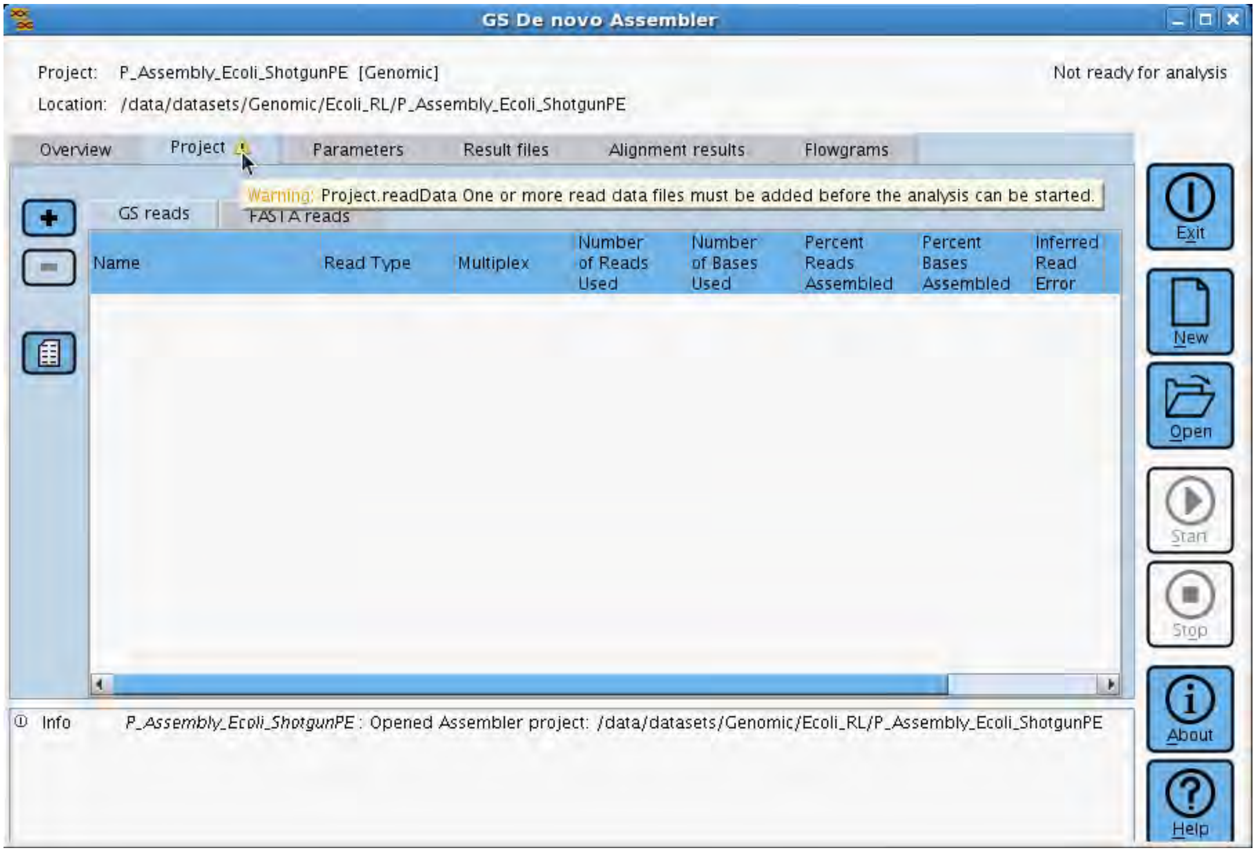
Alignment identity score: 2

Alignment difference score: -3

Info P_Assembly_Ecoli_ShotgunPE : Opened Assembler project: /data/datasets/Genomic/Ecoli_RL/P_Assembly_Ecoli_ShotgunPE

Figure 132: Error messages listed as tooltip on mouse-over of the status message.

When a project is first created and no read files (or read or reference files, for a mapping project) have yet been added, a warning icon on the Project Tab header will be displayed along with the status message 'Not ready for Analysis'



Project: P_Assembly_Ecoli_ShotgunPE [Genomic]
Location: /data/datasets/Genomic/Ecoli_RL/P_Assembly_Ecoli_ShotgunPE

Not ready for analysis

Warning: Project.readData One or more read data files must be added before the analysis can be started.

Name	Read Type	Multiplex	Number of Reads Used	Number of Bases Used	Percent Reads Assembled	Percent Bases Assembled	Inferred Read Error

Info P_Assembly_Ecoli_ShotgunPE : Opened Assembler project: /data/datasets/Genomic/Ecoli_RL/P_Assembly_Ecoli_ShotgunPE

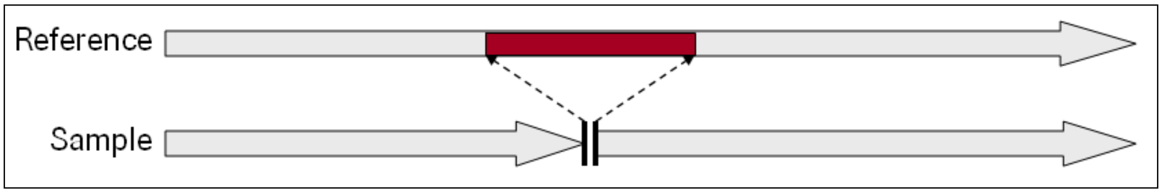
Figure 133: Warning message displayed before read data is added to a project.

4.13 Types of Structural Rearrangements

There are twelve types of rearrangements that may be included in the output files 454HCStructRearrangements.txt and 454AllStructRearrangements.txt: deletions, insertions, substitutions, inversions, tandem duplications, interspersed duplications, translocations, fusions, and exon splices. Tandem duplications, interspersed duplications, and translocations can also be inverted, making the last three rearrangement types. Details for each are given below.

4.13.1 Deletion

Deletions involve only one subject, with two positions for the beginning and end. They can be supported by chimeric shotgun reads or stretched mate pairs.

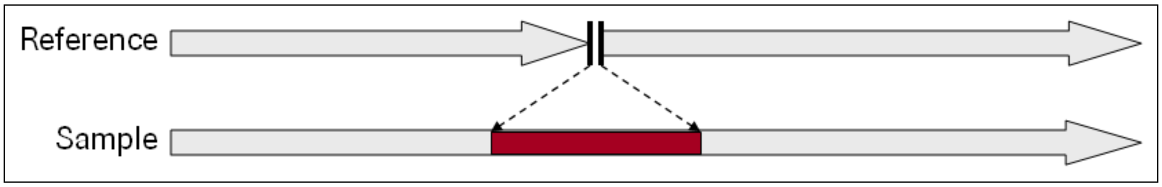


Example output:

```
======================================================================
DELETION
SubjAccno      SubjPos1   RegionName1     SubjPos2  RegionName2     Length  Confidence
gi|6322016|ref|NC_001141.1|     107240          117741          10501   High


Consensus: gi|6322016|ref|          107205+
AATCACAATAGGCAACTCTAAATCATCGTCCAATGTCACAGGAATGGTGTCGGTTGGTTCTGGTGCAAGCAACGT 107279
                                   *
# supporting shotgun reads, pos 107240: 33
# non-supporting shotgun reads, pos 107240: 0

Consensus: gi|6322016|ref|          117701+
GCTGAACAACAGGCGAAAGTGGATGCTGAGAAACTGCTGGTTGGGTTATTGGATCTGAACATAAAACAACCATAGTCT 117778
                                        *

# supporting shotgun reads, pos 117741: 36
# non-supporting shotgun reads, pos 117741: 0


17 mate pairs forward, 5' to 3', coverage = 17, frequency = 100.00
From position 107193 to position 118269
+9930   2
+10198  0
+10466  3
+10734  2
+11002  5
+11270  1
+11538  1
+11806  1
+12074  1
+12342  1


Individual Variation IDs: 241, 242

======================================================================
```

4.13.2 Insertion

Insertions involve only one position on one subject. In this example, the insertion is supported only by shotgun reads.

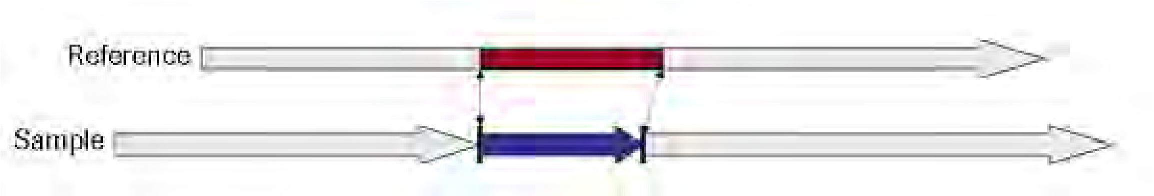


Example output:

```
======================================================================
INSERTION
subjAccno       SubjPos RegionName      Length  Confidence
gi|6322016|ref|NC_001141.1|     275943          -       High

Consensus: gi|6322016|ref|          275907+
TTAAGTGATGAAAAGGCTTTTGAACCCAAATCACACTTGGCCTGTATCTTGGGAGGATTGTGCAACTCGTACG 275979
                                                                     *

# supporting shotgun reads, pos 275943: 53
# non-supporting shotgun reads, pos 275943: 0


Individual Variation IDs: 267, 268

======================================================================
```

4.13.3 Substitution

Substitutions involve only one subject, with two positions for the beginning and end. They are a combination of a deletion and an insertion, such that X subject bases are replaced by Y sample bases. Substitutions are supported only by shotgun reads.

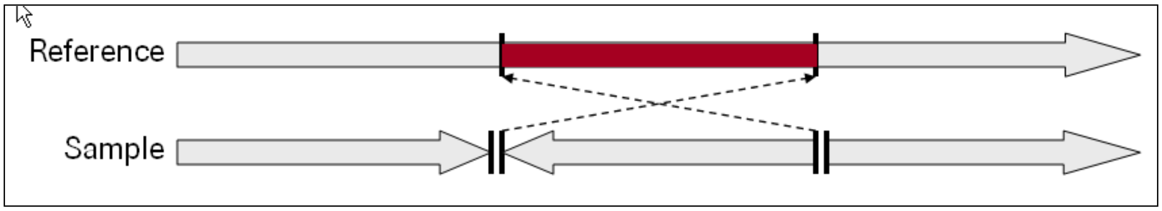


Example output:

```
======================================================================
SUBSTITUTION
SubjAccno    SubjPos1      RegionName1  SubjPos2      RegionName2  Length
       Confidence   M_genitalium 174010        174128       117    High


Consensus:                   173975+
TGTTAGGTGGGCAGTTTGACTGGGGCGGTCGCCTCCTAAAAGGTAACGGAGGCGCACAAAGGTACCTTCA 174044
                                   *
# supporting shotgun reads, pos 174010: 5
# non-supporting shotgun reads, pos 174010: 15

Consensus:                   174088+
CTTGACTGTGAGACTTACAGGTCGAACAGGTGAGAAATCAGGTCATAGTGATCCGGTGGTTCAGTATGGAATGGC 174162
                                     ****

# supporting shotgun reads, pos 174128: 11
# non-supporting shotgun reads, pos 174128: 7


Individual Variation IDs: 1, 2, 3
```

4.13.4 Inversion

Inversions can be supported by both paired end and shotgun reads. Only one subject is involved, and two positions are given for the start and end.

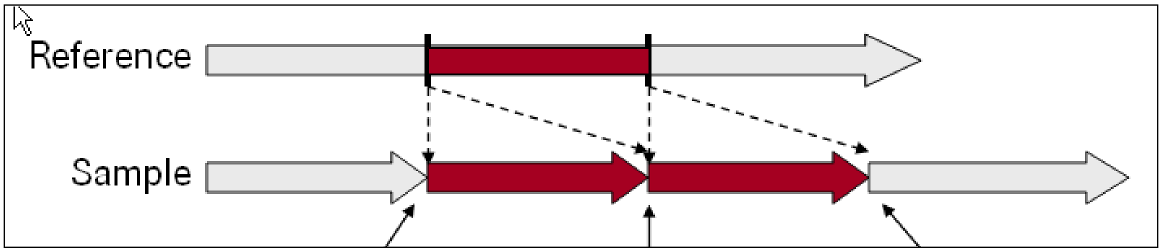


Example output:

```
======================================================================
INVERSION
SubjAccno     SubjPos1     RegionName1     SubjPos2    RegionName2  Length  Confidence
nebCjejuniGold-1.seq    148715          239878          91163   High


59 mate pairs inverted, 5' to 5', coverage = 59, frequency = 100.00
From position 147691 to position 238819
+84129  2
+84923  1
+85717  7
+86511  6
+87305  13
+88099  8
+88893  15
+89687  4
+90481  2
+91275  1


68 mate pairs inverted, 3' to 3', coverage = 68, frequency = 94.44
From position 148715 to position 239878
+84266  2
+85039  3
+85812  10
+86585  12
+87358  10
+88131  13
+88904  10
+89677  6
+90450  0
+91223  2


Individual Variation IDs: 32, 33

======================================================================
```

4.13.5 Duplication, Tandem (May or May Not Be Inverted)

Tandem duplications are only on one subject. The reference, start, and end of the duplicated segment are given, along with the length.

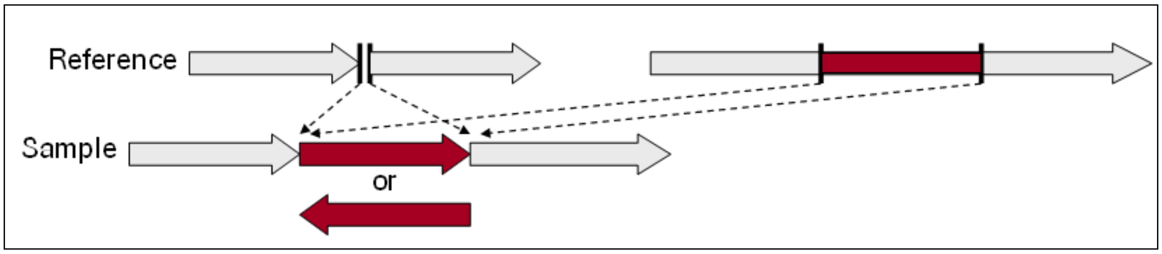


Example output:

```
DUPLICATION, TANDEM, INVERTED
SubjAccno     SubjPos1      RegionName1   SubjPos2    RegionName2   Length  Confidence
gi|85666114|ref|NC_001136.7|    1406931         1427860         20929   High

Consensus: gi|85666114|ref         1406891+
GTCTTGTTTCGTGGACCTTGCACTTCGTAGATTCATCATGTTCTATACGTTTTTCATTTCCGCATTCTAAACCCAAAAGA 1406970
                                        *
# supporting shotgun reads, pos 1406931: 7
# non-supporting shotgun reads, pos 1406931: 10

Consensus: gi|85666114|ref         1427820+
GGGCTTGCGTCATGGACCAGGATTGAGAATCTAAAGAATCCTTCATATTTTCTAACTGTTTACGTTTATTTAAAATCTTC 1427899
                                        *

# supporting shotgun reads, pos 1427860: 7
# non-supporting shotgun reads, pos 1427860: 0


7 mate pairs reverse, 3' to 5', coverage = 7, frequency = 70.00
From position 1407092 to position 1427633
+16255
+16563
+16594
+16672
+16722
+16736
+16880

8 mate pairs inverted, 3' to 3', coverage = 8, frequency = 88.89
From position 1407295 to position 1427939
+16618
+17518
+17782
+18954
+19048
+19092
+19102
+20346

Individual Variation IDs: 328, 329, 330, 332

======================================================================
```

4.13.6 Duplication, Interspersed (May or May Not Be Inverted)

Interspersed duplications involve a segment of the genome being copied to another positions of the genome. The reference of origin, start position, end position, and length of the segment are given, along with its destination ("To") reference and start position. The segment's end position in the reference of origin is not listed in the summary, but may appear below if it is supported by shotgun reads. The origin and destination references may or may not be the same.



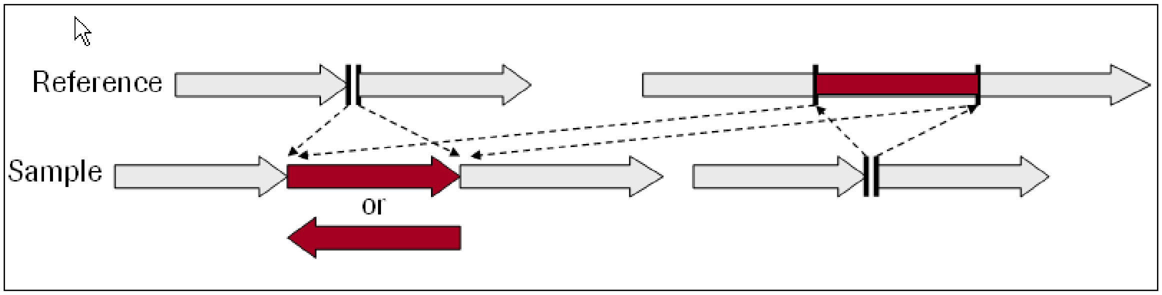
Example output:

```
======================================================================
DUPLICATION, INTERSPERSED, INVERTED
OrigSubjAccno   OrigSubjStart   RegionName1     OrigSubjEnd    RegionName2     ToSubjAccno
ToSubjPos       RegionName3     Length  Confidence
gi|85666114|ref|NC_001136.7|    139931          167891         gi|85666114|ref|NC_001136.7|
1330001         27960   High

Consensus: gi|85666114|ref          139891+
TGAAAAAATTGCCGTTGTCCCACAAGGCGGTAACACGGGGTTGGTAGGTGGTTCTGTGCCCATTTTTGATGAATTAATT 139969
                                                            *

# supporting shotgun reads, pos 139931: 3
# non-supporting shotgun reads, pos 139931: 5

# supporting shotgun reads, pos 167891: 5
# non-supporting shotgun reads, pos 167891: 8

Consensus: gi|85666114|ref         1329961+
TCCAAGAGATGCATACAGAACCAGAGATGCTCCACGTGAAAGATCACCAACCAGGTAAGCCATTTATATAGTTGAGAAAA 1330040
                                        *


# supporting shotgun reads, pos 1330001: 8
# non-supporting shotgun reads, pos 1330001: 0

6 mate pairs inverted, 3' to 3', coverage = 6, frequency = 100.00
From position 140564 to position 1330083
+1185913
+1186047
+1186653
+1187282
+1188394
+1189059

11 mate pairs inverted, 5' to 5', coverage = 11, frequency = 100.00
From position 167891 to position 1329830
+1157647        1
+1158077        0
+1158507        1
+1158937        2
+1159367        0
+1159797        1
+1160227        2
+1160657        1
+1161087        1
```

```
+1161517        2

Individual Variation IDs: 73, 74, 101, 102

======================================================================
```

4.13.7 Translocation (May or May Not Be Inverted)

A translocation is similar to an interspersed duplication, but instead of a segment of the genome being copied to another place in the genome, it is deleted from its original place and moved to the new location. The original reference, start position, end position, and length of the segment are given, along with its target reference and target start position. The original segment's end position is not listed in the summary, but may appear below if it is supported by shotgun reads. The origin and target reference may or may not be the same.

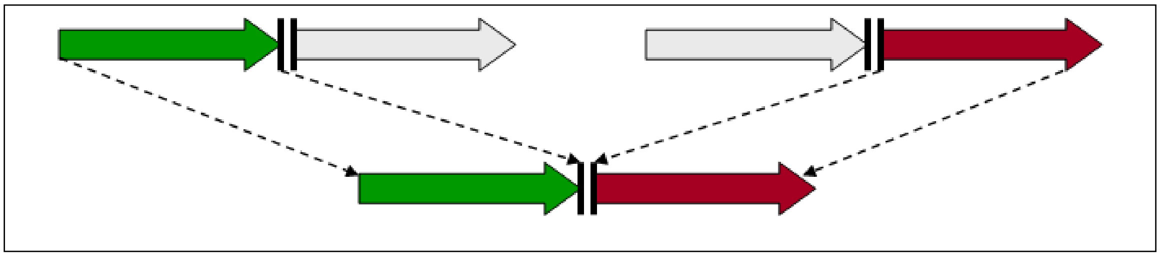


Example output:

```
       ======================================================================
TRANSLOCATION
OrigSubjAccno   OrigSubjStart   RegionName1     OrigSubjEnd     RegionName2
ToSubjAccno     ToSubjPos       RegionName3     Length  Confidence
gi|6322016|ref|NC_001141.1|     70001           97999
gi|44829554|ref|NC_001145.2|    105000          27998   High
Consensus: gi|6322016|ref|           69961+
AACCTAAATCGCGCCTTATCATCTTGGTAGAAAGTAAACCATCCACTTTAGGCATCTGGACATCCATGAAAATCATATTAT
70041
                                        **
# supporting shotgun reads, pos 70001: 7
# non-supporting shotgun reads, pos 70001: 0
Consensus: gi|6322016|ref|           97959+
CCCGCACTGTTTGGTGTTTCGCAAGGTGCCTTATATTTTGCAGTTTACGATACCTTAAAGCAAAGAAAATTGCGACGGAA 98038
                                        *
# supporting shotgun reads, pos 97999: 6
# non-supporting shotgun reads, pos 97999: 0
Consensus: gi|44829554|ref          104960+
CTATGCCACATCTGTACAAGGGGGTTCGTGAGACAAGAGCATTTAAAACGACATCAAAGAGCACATACGAACGAGAAACC
105039
                                        *
# supporting shotgun reads, pos 105000: 7
# non-supporting shotgun reads, pos 105000: 0
Individual Variation IDs: 337, 338, 339
======================================================================
```

4.13.8 Fusion

A fusion is detected by chimeric or paired end reads connecting two reference chromosomes or two genes. The output file provides the reference, position and gene name at each side of the connection.

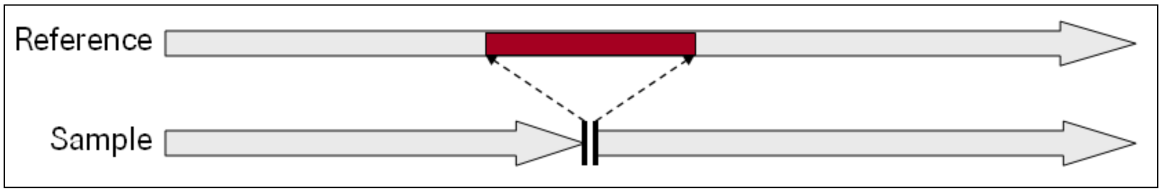


Example output:

```
======================================================================
FUSION
subjAccno1     subjPos1     RegionName1     subjAccno2     subjPos2     RegionName2
Confidence          chr9     20350776     MLL     chr11     117859950     MLLT3     High

Consensus: chr9                   20350740+
CTGGTTGTTGTTGGTTTTTAGTAAGGGTGGTGGAGGTTCGTGATGTAGGGGTGAAGAAGCAGAACTG 20350806
                                    *

# supporting shotgun reads, pos 20350776: 7
# non-supporting shotgun reads, pos 20350776: 30

Consensus: chr11                 117859918+
CTGTATTGCAGCCTAGGCAACAAAGCAAGACCCAGTCTCTTTTAAAAAAAAATTCAAAG 117859976
                                *


# supporting shotgun reads, pos 117859950: 8
# non-supporting shotgun reads, pos 117859950: 26


Individual Variation IDs: 23, 24

======================================================================
```

4.13.9 Exon Splice

Exon splices are essentially deletions, but they occur only in projects mapping cDNA reads to a genomic reference, and only if both ends of the deletion are within the same gene. They are supported by chimeric shotgun reads. We do not currently use paired end reads for cDNA projects.



Example Output:

```
======================================================================
EXON SPLICE
SubjAccno      SubjPos1    RegionName1    SubjPos2     RegionName2  Length  Confidence
chr10   276051  ZMYND11 276833  ZMYND11 782     High


Consensus: chr10                    276011+
CTGACATTGCGAGGATGCTATATAAAGACACATGTCATGAGGTACTATTCATTGCCCAATAGTTATACTCTTTCTATAAC
276090
                                        *
# supporting shotgun reads, pos 276051: 3
# non-supporting shotgun reads, pos 276051: 0

Consensus: chr10                    276793+
ATATTTTATGTAGGCTAAAGTAGTTTCTTTTCTTTTTCAGCTGGATGAACTGCAGCTTTGCAAGAATTGCTTTTACTTG 276871
                                        *

# supporting shotgun reads, pos 276833: 3
# non-supporting shotgun reads, pos 276833: 0


Individual Variation IDs: 7

======================================================================
```

4.13.10 Circular Genome

4.14 Target Regions and Extended Target Regions

For targeted resequencing mapping projects, target regions are defined as part of the experimental design. In the GS Reference Mapper, these targeted regions are specified *via* the Input files section, on the Input sub-tab of the Parameters tab (Section 2.7.1).

Many targeted resequencing protocols will produce reads that extend beyond the actual target region. The sequence information from the areas just outside the target region boundaries is often useful, *e.g.* the intronic regions flanking exons in an exome experiment. The GS Reference Mapper reports these sequences as belonging to “extended target regions”.

Specifically, target regions and extended target regions are defined as follows (Figure 134):

- **Target regions:** defined in an input file provided to the project *via* the Input sub-tab of the Parameters tab (or on the command line, using the –reg option). All reads and contigs that overlap a target region, and sequencing variants therein, are labeled “InRegion”.
- **Extended target regions:** sum of the targeted regions and all contigs that overlap the targeted regions. The reads within these contigs that DO NOT overlap a target region, and sequencing variants therein, are labeled “InExtRegion”.
- Any read that does not overlap with an extended target region, and any sequencing variant therein, is labeled “OutOfRegion”.

This principle applies whenever target regions are specified on the Input sub-tab, whether the DNA sample has been physically selected as part of the library preparation procedure.

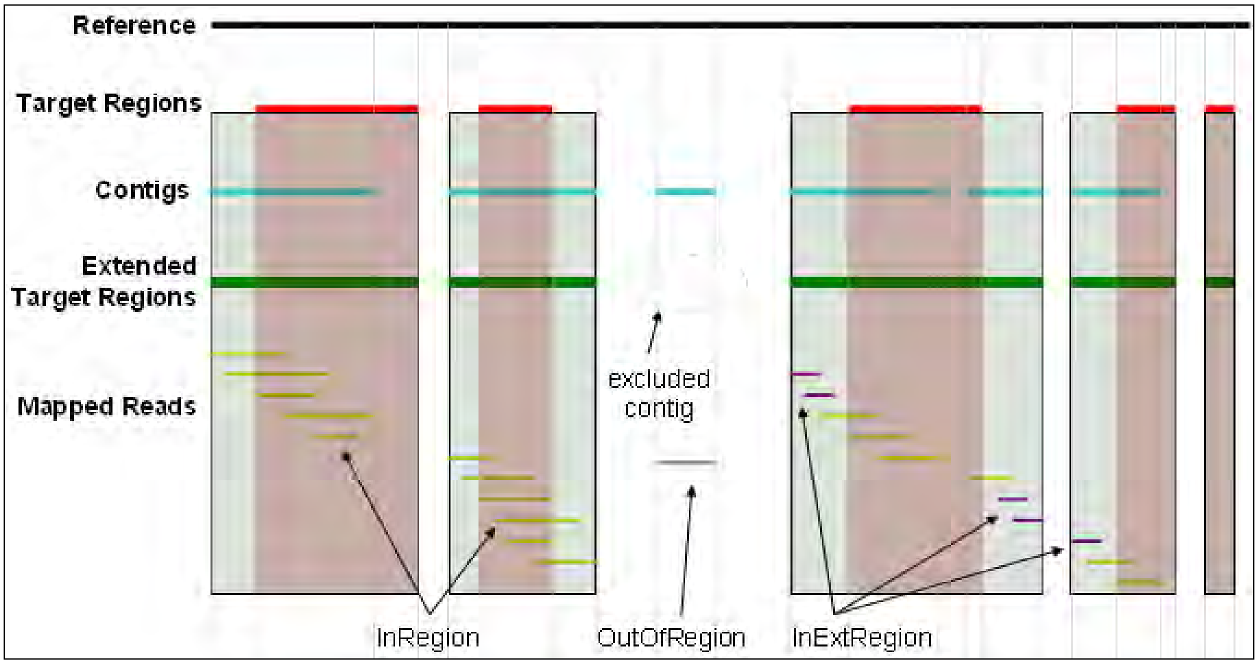


Figure 134: Target regions, extended target regions, and out of target regions.

5 cDNA / Transcriptome Sequencing Appendix

5.1 Introduction to cDNA Sequencing Analysis

cDNA is produced from the RNA (often, mRNA) in a sample. The cDNA is then sequenced resulting in reads that represent the original sample's mRNA. These reads can then be mapped against one or more references to gain information about the representation of individual mRNAs and genes from which the mRNAs are transcribed and their level of variation in the original sample. Alternatively, the reads can be assembled together to discover novel transcripts and splice variants for those mRNAs represented by multiple overlapping reads.

5.2 Transcriptome Assembly Concepts

5.2.1 Definitions

5.2.1.1 Isogroup

An isogroup is a collection of contigs containing reads that imply connections between them. A discussion of the assembly process (see Section 1.1) explains how breaks can be introduced into the multiple alignments of overlapping reads, leading to branching structures between them. After attempting to resolve the branching structures, the Transcriptome Assembler groups all contigs whose branches could not be resolved into collections called isogroups. Using rules described in the following section, the assembler traverses the various paths through the contigs in an isogroup to produce the set of isotigs that gets reported. All possible paths through the contigs in an isogroup are traversed unless one or more thresholds is reached (see Section 5.2.2.2).

5.2.1.2 Isotig

An isotig is meant to be analogous to an individual transcript. Different isotigs from a given isogroup can be inferred splice-variants. The reported isotigs are the putative transcripts that can be constructed using overlapping reads provided as input to the assembler. Connections between contigs in an isogroup are represented by sequences (reads) that have alignments diverging consistently towards two or more different contigs (see Figure 135) or by a depth spike (Section 5.2.2.2). Traversal from the start contig to the end contig or from the end contig to the start contig should yield the same but reverse-complemented isotig sequence.

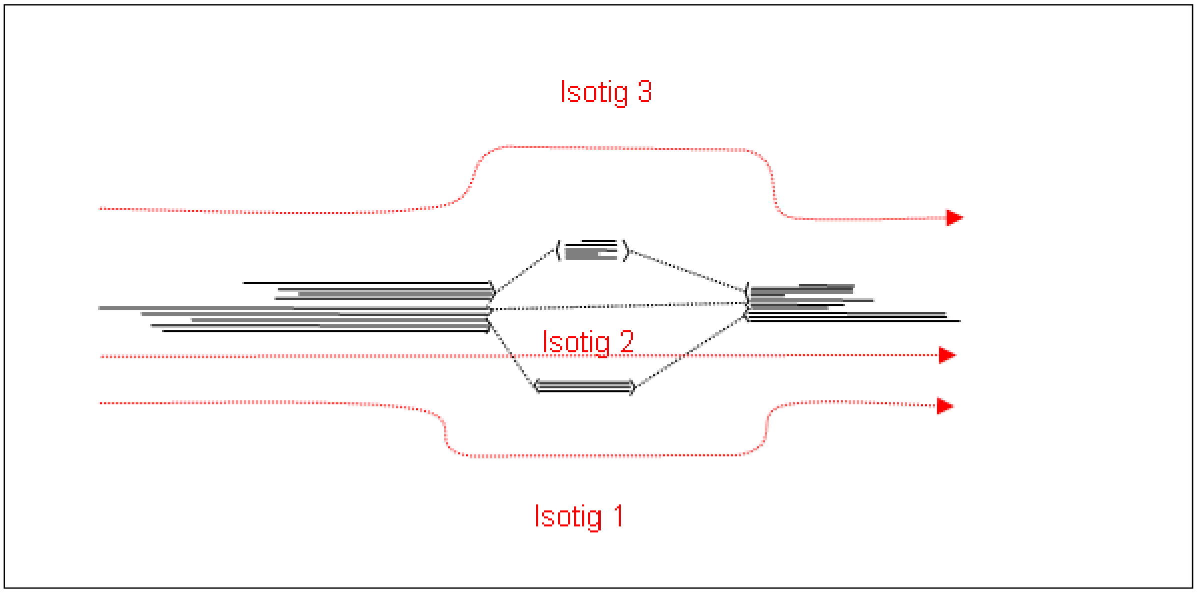


Figure 135: Traversal of contigs in an isogroup.

While many reads may contain poly-A tails, these tails are trimmed off prior to assembling the reads. Presently, the assembler ignores the fact that poly-A tails existed, so the orientation of reads in the assembly cannot be determined. Because of this lack of directionality, an isotig may be output as the reverse-complement of the biological transcript it represents. Contigs forming an isotig may be thought of as exons. This is not strictly correct, however, since untranslated regions (UTRs) and introns (in the case of primary transcripts) may exists in the reads generated from the sample.

5.2.2 Rules for Path Traversal of Contigs in an Isogroup

5.2.2.1 Path Initiation

Contigs lacking reads connecting them to other contigs on one or both ends are used to initiate the traversal of an individual isotig. When a contig has no reads connecting it to any other contigs, it may become an isotig composed of a single contig (see Figure 136).

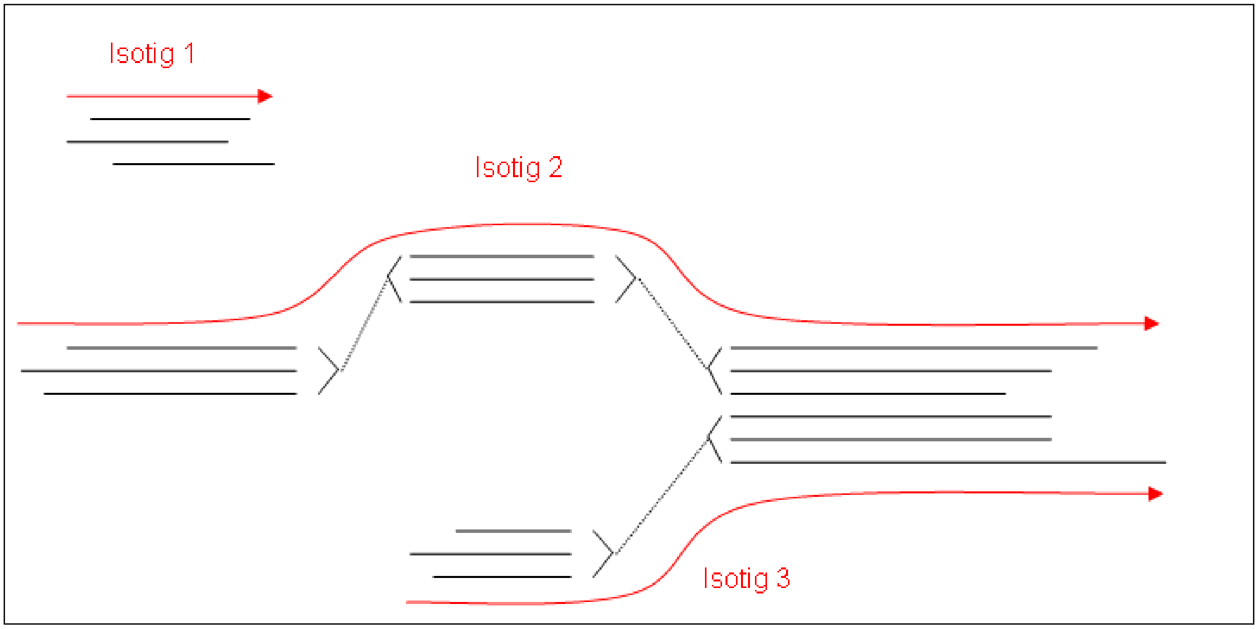


Figure 136: Isotig initiation.

If the –isplit option is used, isotigs will also be initiated when a depth “spike” of at least 50% between one alignment column and the next is found. When a “spike” in depth is found while traversing an isotig path through the alignment for a contig (see Section 5.2.2.2), the alignment column in which the relative depth change occurs is marked for future use as an initiation point for further isotig traversal. The isotig path up to that point is reported as an isotig, and the original isotig’s path continues until it terminates. The marked initiation point is then used as a starting contig and all paths starting from it are reported until they are exhausted.

5.2.2.2 Spike Detection

Identification of a spike along a path occurs when all of the following conditions are met at the end of a contig where a spike is detected:

- The alignment depth is at least 10 reads
- A minimum of 20% of the aligned reads must be in the opposite orientation relative to the more abundant orientation of the aligned reads
- A spike may not occur within 10 bases of an already detected spike
- A change in alignment depth between one alignment column and the next of at least 50% signals the location of a "spike".

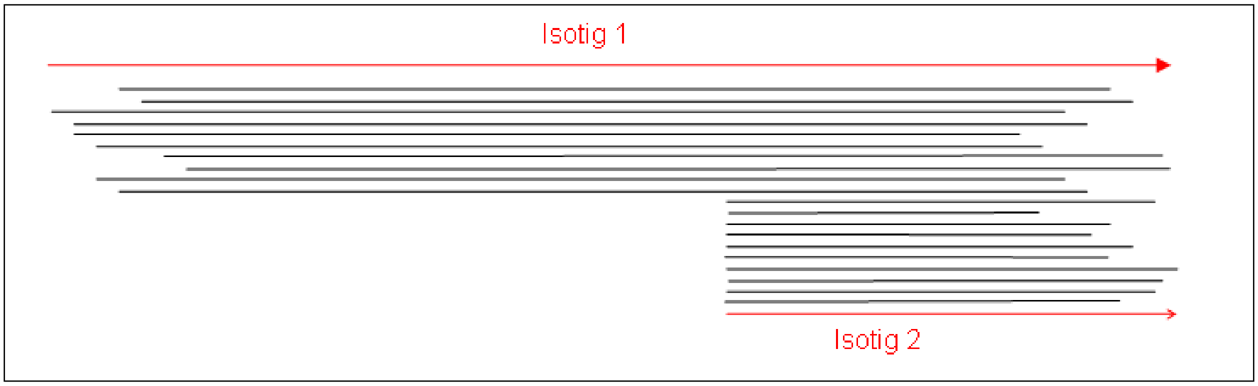


Figure 137: Spike identification.

5.2.2.3 Path Extension

An isotig path that starts at an initiating contig (call this Contig I) will be extended by following reads that consistently lead to another contig. Once a terminating condition is found on a particular path, the isotig is reported, and then all other paths starting from Contig I are reported until they are exhausted. The process used to explore all possible paths is called "recursion." Extension along a particular path will continue until a terminating condition is met (see Section 5.2.2.4).

5.2.2.4 Path Termination

The end of an isotig is found when any of the following conditions occur:

- No reads are found that extend from the contig currently at the end of the isotig path
- The number of reads connecting two contigs is less than 5% of the alignment depth of either
- The **Isotig Contig Count Threshold** is reached. In this case, the further traversal of a particular isotig in an isogroup will be stopped.
- A contig is reached whose length is below the **Isotig Contig Length Threshold**. If a contig is reached with a length shorter than this threshold, the further traversal of a particular isotig in an isogroup will be stopped. The contig shorter than the icl threshold will be marked as such and reported in the output files.
- A cyclic path is encountered. Recursive path traversal will stop if cyclic structures are detected, *i.e.* revisiting one contig which has already been included in an earlier part of the isotig being traversed. Such cyclic structures will be marked in the output files by assigning cyclic status for the first contig detected.

5.2.2.5 Other Rules that Prevent Path Traversal

- If the number of contigs in an isogroup exceeds the **Isogroup Threshold**, the paths in the isogroup will not be traversed to report isotigs.
- When the number of isotigs in an isogroup is found to exceed the **Isotig Threshold**, further traversal of isotig paths for the group stops.

5.3 Transcriptome Mapping Concepts

5.3.1 Mapping cDNA Reads to a Transcriptomic Reference

Reads derived from cDNA samples can be mapped to a collection of reference sequences representing the set of transcripts that might be found in the sample. Often this collection of reference sequences is a list of known or putative transcripts (such as refMrna.fa from UCSC). Some of the applications that can be facilitated by such a mapping include the detection of novel SNPs within known transcripts and the profiling of expression levels for known transcripts and genes.

Many of the reads from cDNA sample may map uniquely to only one transcript. Uniquely mapped reads can be used for SNP detection and for profiling transcript levels. However, because some exons may be shared by many transcripts (*e.g.* splice-variants from a single gene), reads covering such exons may map equally well to many transcripts. In such cases, it may not be possible to assign the reads to any single transcript reference. Nonetheless, when mapping to transcriptomic references, it is often possible to determine the number of reads that map to a single gene even when reads cannot be mapped uniquely to a single transcript produced by the gene.

5.3.2 Mapping cDNA Reads to a Genomic Reference

When mapping reads from cDNA samples to genomic reference sequences, it is often possible to detect splice variants that might not be apparent when mapping to a transcriptomic reference. Breaks in the alignments of individual reads against a region of the genome often indicate exon boundaries. Also, while one may be able to map only parts of reads derived from novel splice variants to a set of known transcripts, mapping against the genomic reference allows the identification of genomic regions not previously known to exist as part of the transcriptome. Mapping to a genomic reference only allows per-gene coverage statistics to be reported while mapping to a transcriptomic reference can facilitate the profiling of individual transcripts.

5.4 Reference Type Auto Detection

For cDNA mapping projects, the reference type (cDNA or gDNA) can be automatically detected under certain circumstances thus removing the need to specify the reference type on the command line. The logic employed depends on whether or not the GOLDENPATH environmental variable is in use.

5.4.1 Using GOLDENPATH

5.4.1.1 Genomic Reference

When the GOLDENPATH environment variable is in use, the reference should not be specified when a genomic (gDNA) reference type is being used. The software will automatically find the reference and annotation files in the GOLDENPATH directory structure. When using the GUI, use the default setting for the reference type (Auto). On the command line, ***do not*** use the **–gref** argument with **setRef**, **runProject,** or **runMapping.**

Example:

```
setRef hg18
```

5.4.1.2 cDNA Reference

When a transcriptome (cDNA) reference type is being used in conjunction with the GOLDENPATH environment variable, auto-detection of the reference type is not possible. You must use the **–cref** argument as an option to **setRef**, **runProject** or **newbler** or select the corresponding choice on the GUI.

Example:

```
setRef –cref hg18
```

5.4.2 Not Using GOLDENPATH

5.4.2.1 Reference and Annotation Files Are in the Same Directory

In this case, the software will automatically detect the reference type based on the annotation file that is present in the directory where the reference file is located. The following rules apply:

- The presence of an annotation file named **refGene.txt** tells the software that the reference type is genomic (gDNA), irrespective of the presence of any other annotation files.
- The presence of an annotation file named **refLink.txt** tells the software that the reference type is transcriptome (cDNA), ***UNLESS*** an annotation file named **refGene.txt** is also present.

5.4.2.2 Reference and Annotation Files Are in Different Directories

In this case, you must specify the location of the annotation files by using the **–annot** option or the corresponding choice in the GUI. Specifying **refLink.txt** tells the software that the reference type is transcriptome (cDNA). Specifying **refGene.txt** tells the software that the reference type is genomic (gDNA).

5.4.2.3 Obtaining Annotations from the Reference File

In some cases, separate annotation files are not available, but annotation data is included in the header line(s) of the reference file. By default, the software will assume the reference type is genomic. However, if the annotation data contains the following name/value pair:

gene=geneName

the software will set the reference type to transcriptome (cDNA).

Some reference files (*e.g.*, FlyBase) explicitly declare the type of the reference by including the name/value pair **type=*referenceType.*** In these cases, the software will set the reference type to transcriptome if the specified reference type matches the string **mRNA**. Any other value, or the absence of the key **type** in the annotation data, will result in the software setting the reference type to genomic.

GLOSSARY

A

Accno – accession number of the input read. If this is a paired end read, the accno is followed by an underscore character and the mention "left" of "right", for which half of the pair this read comes from.

Assembly – computation performed by the GS *De Novo* Assembler. See GS *De Novo* Assembler.

C

cDNA project – a project that contains cDNA data. See Project.

Command Line Interface (CLI) – software commands run from the command line prompt.

Contig– contiguous multiple alignment.

Contig graph – description of branching structure between contigs.

D

Dot – a block of negative flows (denoted as 'N' in a DNA sequence) that is ended by a positive flow of one of the nucleotides in the block, or started and ended by positive flows of the same nucleotide.

F

Flow – during a sequencing run, nucleotides are flowed sequentially across the PTP device, one at a time, as controlled by the run script. When the flowed nucleotide is complementary to the next homopolymer (including a single nucleotide) on the DNA template in any given well, the polymerase extends the nascent DNA strand in that well. Addition of one or more nucleotide(s) release(s) a corresponding number of pyrophosphate (PPi) molecules. One molecule of ATP is synthesized for each PPi released, causing a flash of light (signal) whose intensity is proportional to the number of nucleotides incorporated.

- **Key flows** – the first few nucleotide flows needed to sequence the library and control sequence keys. For the flow order 'TACG' and the key 'TCAG', the key flows would be **T**-A-**C**-G-T-**A**-C-**G** (incorporation in bold) and consist of eight flows.
- **Negative flow** – a well-specific attribute denoting a nucleotide flow where no signal is detected and thus no nucleotide incorporation is assumed.
- **Positive flow** – a well-specific attribute denoting a nucleotide flow where signal is detected and the intensity of the signal is related to the number of nucleotides incorporated.

Flow cycle – an invariant flow set consisting of exactly four nucleotide flows, repeating in a specific flow order.

Flow list – the series of nucleotide flows during a sequencing run, as specified by the run script.

Flow order – the repeated sequence of nucleotides flowed during each flow set of a cyclic flow pattern sequencing run, generally 'TACG'.

Flow pattern – the pattern of nucleotide flows during a sequencing run, as determined by the choice of run script.

- Cyclic flow pattern – a pattern of nucleotide flows characterized by a repeated cycle of four nucleotide flows, with each cycle (flow set) defined by a specific flow order.
- Acyclic flow pattern – a pattern of nucleotide flows characterized by a pattern that is not cyclic.

Flow set – the smallest group of nucleotide flows at any point in a flow list that includes at least one flow of each of the four nucleotides, with the simplest case being a four nucleotide flow cycle in a cyclic flow pattern.

Flowgram – series of values representing the results of a 454 Sequencing run. Provides the input data for the GS *De Novo* Assembler and the GS Reference Mapper.

Flowgram gap – adjustments introduced to generate a common flowgram flow list that can be used to display the relative alignment between read and reference/consensus flowgrams; also known as a cycle shift when referring to cyclic flow pattern reads.

- **Reference flowgram gap** – a block of negative flows identified in the reference flowgram that is associated with a putative SNP or indel that is 'missing' positive flows in the reference relative to the read. The shaded flows would have been classified as a dot (too many negative flows in a row) if the flowgram had been derived from an actual sequencing run. Subtracting these shaded flows results in a theoretical flow list that could have been used to generate the reference flowgram.
- **Read flowgram gap** – a block of negative flows inserted into the read flowgram that is associated with a putative SNP or indel that requires 'extra' positive flows in the reference relative to the read. The shaded flows are duplicated from the adjacent flow list to aid in alignment with the reference. Subtracting these shaded flows results in the actual flow list used to generate the read flowgram.

fna file – Phred-equivalent file that contains the nucleotide sequences of all of the contigs and associated nucleotide Quality Scores.

G

GS *De Novo* Assembler – software that identifies pairwise overlaps between reads, constructs multiple alignments of overlapping reads, and divides or introduces breaks into the multiple alignments in regions where consistent differences are found between different sets of reads. (This step results in a preliminary set of "contigs" that represent the assembled reads.) After attempting to resolve branching structures between contigs, the software generates consensus basecalls of the contigs by using quality and flow signal information for each nucleotide flow included in the contigs' multiple alignments. Output includes the contig consensus sequences and corresponding quality scores, and an ACE file of the multiple alignments and assembly metrics files.

GS Reference Mapper – software that aligns sequencing reads against reference sequences (consisting of one or more sequences or a GoldenPath genome), with or without associated annotations. Mapping generates consensus sequences of the reads that align against the reference and also computes statistics for variations found in the reads, relative to the reference.

Genomic project – a project that contains genomic assembly data.

GS reads – result of the sequencing process that provides input to an assembly project. Read types can be paired-end or non-Paired end. The Read Type Specification auto-detects unknown read type data.

H

HC diffs – High Confidence difference. The GS Reference Mapper application uses a combination of flow signal information, quality score information and difference type information to determine if a difference is High-Confidence.

I

Isogroup – represents the Transcriptome Assembler groups of contigs that are connected in a multiple alignment graph structure.

M

MIDConfig.parse – the default MID configuration file used by GS *De Novo* Assembler and GS Reference Mapper, and contains MID names, sequences, and the allowed number of errors. The MIDConfig.parse file is found in *installDir*/apps/gsSeqTools/config/MIDConfig.parse, where *installDir* is the main software installation path (*e.g.* /usr/local/rig/ on the GS FLX+ instrument or /opt/454/ on the GS Junior attendant PC). GS Amplicon Variant Analyzer using a family of scripts to populate project MIDs. These scripts are found in *installDir*/apps/amplicons/config/lib/.

Multiplex Identifier (MID) – a unique identifier that is attached a DNA library to identify the library to which an individual read belongs. Allows multiple libraries tagged with different MIDs to be sequenced together, within an individual PTP device.

N

Newbler Assembler – software package for *De Novo* DNA sequence assembly that is run by the GS Assembler graphical user interface or the Command Line Interface (CLI).

P

Path extension – in transcriptome assembly, an isotig path that starts at an initiating contig and continues until a terminating condition is found.

Path termination – in transcriptome assembly, represents the end of an isotig.

Path traversal – in transcriptome assembly, analysis of branching of contigs in an isogroup.

Project – provides an interface for creating, modifying, and running assemblies using the GS *De Novo* Assembler. The project folder in the file system contains the assembly project files. Users can run projects from the command line interface or using a graphical user interface (GUI).

Q

Qual file - Phred-equivalent file that contains the nucleotide sequences of all of the contigs and associated nucleotide Quality Scores

R

Read status – status of the read in the assembly, which can be: Assembled, PartiallyAssembled, Singleton, Repeat, Outlier or Too Short.

Rearrangement points – a single structural variation relative to the reference sequence.

Rearrangement regions – when a cluster of paired-end reads that are considered False Pairs shows a consistent deviation from the reference (varying either in size or expected orientation), a rearrangement region is reported.

Reference data – data files that contain reference DNA sequences for comparison to flowgrams.

S

Scaffolds – a series of contigs that are in the right order but not necessarily connected in one continuous stretch of sequence. The GS *De Novo* Assembler uses paired end information calculate the approximate the distance between the contigs.

SFF file – Standard Flowgram Format file - output data from the 454 Sequencing system.

SFF Tools – Software tools for managing SFF files.

Short tag reads – short reads that are processed with different overlap detection parameters that allow more efficient mapping of shorter sequences relative to the standard overlap detection parameters. During assembly, short tag reads are not involved in building contig consensus sequences, but are instead mapped to these contig consensi for increased depth of coverage.

Structural variations – larger-scale changes relative to the reference that are indicated by a group of reads. Entries in the table are classified as either Rearrangement points or Rearrangement regions.

T

Transcriptomic project – a project that contains cDNA data. See Project.

INDEX

Published by
454 Life Sciences Corp.
A Roche Company
Branford, CT 06405
USA

Notice to Purchaser
For patent license limitations for individual products please refer to: **www.technical-support.roche.com**.

For life science research only. Not for use in diagnostic procedures.

(9) 0613