SOLiD™ Data Format and File Definitions Guide

V 1.0 - Files deposited on 1/March/08

Overview

Runs

The files deposited are arranged by run, there are a total of 14 runs deposited. Each run is a single flow cell, SOLiD System has two flow cells the _1 or _2 refers to which flow cell data came from.

Run	Mate pair or Fragment Library	Insert size bases (sd)
BARB_20071114_1	Mate pair library 2 x 25 base reads	1,711 (316)
BARB_20071114_2	Mate pair library 2 x 25 base reads	1,179 (221)
CLARA_20071113_2	Mate pair library 2 x 25 base reads	600 (58)
CLARA_20071113_1B	Mate pair library 2 x 25 base reads	804 (184)
JOAN_20080104_1	Mate pair library 2 x 25 base reads	2841 (611)
JOAN_20080121_1	Mate pair library 2 x 25 base reads	3469 (697)
Florence_20080201	Mate pair library 2 x 25 base reads	2382 (726)
AMELIA_20071210_1	Fragment Library 50 base reads	
AMELIA_20071210_2	Fragment Library 50 base reads	
AMELIA_20071010_1	Fragment Library 45 base reads	
AMELIA_20071010_2	Fragment Library 50 base reads	
LIZ_20071107_1	Fragment Library 50 base reads	
CLARA_20080108B_1	Fragment Library 50 base reads	
CLARA_20080108_2	Fragment Library 50 base reads	

Files

For each run a set of files is produced

xxxxx _F3_QV.qual
xxxxx _F3.stats
xxxxx_ F3.csfasta
xxxxx _F3.all_chromosomes.csfasta.ma.25.2 {last two digits refer to
xxxxx _F3.all_chromosomes.unique.csfasta.ma.25.2 matching parameters}
xxxxx _F3/all_chromosomes.unique.stats.txt
xxxxx _F3/chr_unique/chr1.all_chromosomes.unique.bc.txt {one file per chromosome}

xxxxx_R3.csfasta

xxxxx _R3_QV.qual
xxxxx _R3.stats
xxxxx _R3.all_chromosomes.csfasta.ma.25.2 {last two digits refer to
xxxxx _R3.all_chromosomes.unique.csfasta.ma.25.2 matching parameters}
xxxxx _R3/all_chromosomes.unique.stats.txt
xxxxx _R3.csfasta.ma.25.2.25.2.chr17.bc.txt
xxxxx_R3/chr_unique/chr1.all_chromosomes.unique.bc.txt {one file per chromosome}

xxxxx /F3_R3.mates
xxxxx /F3_R3.mates.unique

grey = Mate pair runs only

Note: Detailed file descriptions start on page10

Possible uses of data

The deposited data is in color space. In order to allow mapping of the data with different parameters the *Color Space Mapping Tool* has been included in the distribution as a tar file. It is also available at info.appliedbiosystems.com/solidsoftwarecommunity (see Appendix 2 for description of use)
The xxxx.bc.txt files include data on position of the read and the mismatches in the file, Using this information, the distribution of the reads as well as locations of potential SNPs can be studied (a potential SNP will require two adjacent mismatches that meet the color space criteria)

The xxxx.mates files contain the information on the alignment positions of the mates as well as if the mates showed any aberrant behavior (see detailed description). This data set can be used to look for structural variation relative to the reference sequence

Reference sequence

The reference sequence used was hg18 downloaded from
http://hgdownload.cse.ucsc.edu/goldenPath/hg18/chromosomes/
This is the same as NCBI Build 36.2
http://www.ncbi.nlm.nih.gov/genome/guide/human/release_notes.html

Note only the normal chromosome data was downloaded and used for analysis (eg chr1.fa.gz)the random and alternative haplotype files were not used.

Color Space and Base Space

All files that are provided contain only color space data. Rather than reading one base per cycle, information on two bases is measured simultaneously, and in each cycle, one of four colors is called (color space call); this is the basis of color space. Users unfamiliar with color space should download the description of two-base encoding from *http://solid.appliedbiosytems.com*.

Processing

The flow of data through the SOLiD system pipeline is shown in Figure 1 below.

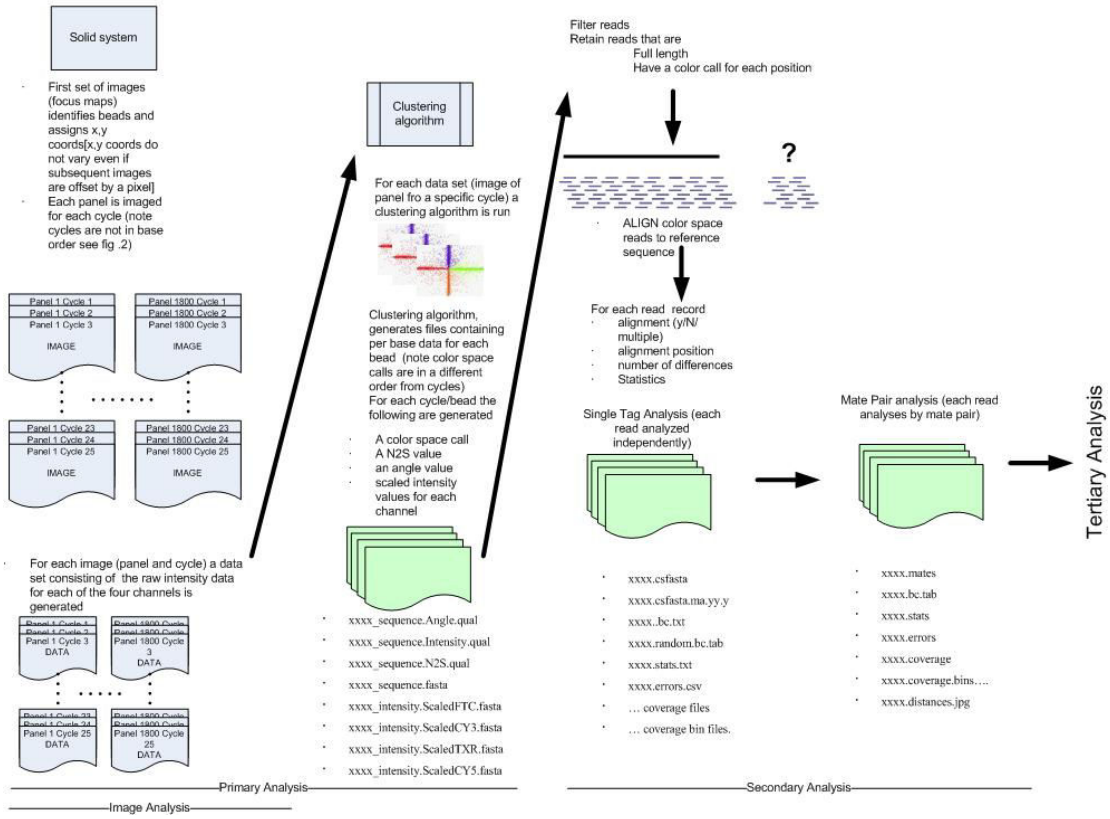

Figure 1. Data flow in the Applied Biosystems SOLiD™ System

Overview

The analysis process has three main parts:

- Image analysis: Images for each cycle are analyzed to produce raw intensity values for each channel and for each bead.
- Primary analysis: Data are clustered and normalized. For each tag, a sequential (sequence ordered) set of color space calls is produced. Normalization produces a set of quality metrics.
- Secondary Analysis: Data are filtered to remove poor-quality reads and mixed beads, then aligned to the reference sequence. If the experiment is a mate-paired run, analysis of the mate pairs is carried out.

Detail

In the data flow shown in Figure 1, each panel is imaged four times per cycle, once per channel. The color space calls are carried out on a by-cycle basis; each data point represents an individual bead with four intensity values. Beads are assigned a color-space

call using a clustering algorithm. As part of the clustering process, data are scaled, baseline, and color-call assigned.
These data are filtered by removing all tags with missing data. Filtering removes all incomplete (not full length) beads and those reads with missing calls. The filtered data are then processed to generate the file *xxxx.csfasta*. As part of this process, a known base (from the adapter sequence) is prepended to the color space data. Each read is treated separately, even if it is part of a mate-paired tag. The file contains the color space read for each tag, and the raw color space data are used to generate all subsequent results. The data are then aligned in color space to the color space reference sequence. (Color space reference sequences are derived by converting the base-space reference sequence to color space; all alignment is done in color space.)

- All the alignments are with individual tags; the mate-paired information is not used. -
- After the single reads are analyzed, a further round of analysis is performed in which the mate-pair information is used.
- The paired tags are analyzed in a two-step process:
  1) First analyzed are those tags where both the forward and reverse passed (i.e. both tags were matched to a reference sequence).
  2) Then, those mate pairs where only one of the tags was matched to the reference sequence (in color space) are reanalyzed, and the non-matching tag is compared to the reference sequence. Because the problem can be constrained by the knowledge of the allowed distances, this process allows alignment with more disagreements and the use of a more cpu-intensive alignment algorithm.

Where possible, all data are presented in a fasta-compatible format to facilitate use in a variety of applications. Also included are files in other formats that are used in the analysis pipeline process; this does lead to some duplication of data.

Glossary

1. **Color code**

   The four dye colors are encoded as follows:
   - 0 6-FAM™
   - 1 CY3
   - 2 Texas Red
   - 3 CY5

2. **Two base encoding matrix**

1st Nucleotide \ 2nd Nucleotide	A	C	G	T
A	**0**	**1**	**2**	**3**
C	**1**	**0**	**3**	**2**
G	**2**	**3**	**0**	**1**
T	**3**	**2**	**1**	**0**

   Examples:
   AA is encoded as 0
   CG is encoded as 3
   AACG is encoded as 0 1 3

3. **Spot**

   When multiple samples are run on a single array, each array has its own "spot". This is shown in the file name format S#. For example, S1 are files for Spot 1. Each spot consists of a series of panels. All panels have unique numbers, that is, numbering is continuous across spots.

4. **Panel**

   A panel is the region of the array that the camera views. Successive cycles result in four images for each panel (one per color). Beads are identified by position within a panel, that is, x,y coordinates are not unique values across panels. The combination of panel and x,y coordinates uniquely identifies a bead.

5. **TAG_ID**

   The Tag_ID is a unique identifier for every tag, which consists of four components: `panel_xpixel_ypixel_tagtype.`
   For example, `1_567_321_F3` describes a bead in panel 1 at coordinates 56, 321 (X,Y) with the F3 tag (first tag in a mate pair, only tag in a fragment run).

   Note: This Tag ID is used to describe the bead and its data in all the files.

   X,Y coordinates are fixed and derived from the initial focus map used to identify beads. Even if successive images differ by a pixel in alignment, the identified bead always has fixed X,Y coordinates. F3 and R3 are used to describe tag and orientation. F and R are 1st and 2nd tags in a construct (historically forward and reverse in a Sanger mate-paired library). The 3 specifies that it is 3′ to 5′

chemistry. Both reads are on the same strand going 3 to 5, which differs from traditional mate pairs with Sanger sequencing where the reads are 5 to 3 oriented and on opposite strands. NOTE: Sequence reported in the csfasta file is 5->3 reflecting the template strand not the synthesized strand!

6. Location

Location, given for processed files (see color space format below), describes the location of the prepended base on the base-space reference sequence.

In some cases the chromosome is prepended (csfasta file)
E.g. 6_7645363

Represents a location on Chromosome 6, at base 7645363

Color space format

The color space data are presented in three slightly different formats. In two of the formats, a base (a, c, g, or t) is appended to the color space calls. To understand the difference between these formats, be aware that:

- *Raw color space data* includes cycles where no base is called, shown as a "."No base is appended to color space data. These reads are removed on filtering and are available only in the file `xxxx_sequence.fasta.`
- *Unprocessed color space data* referred to as `color_space` in file descriptions, consists of a numeric string prefixed by a single base. This base is the final base of the sequencing adapter and is not part of the target sequence.

Processed color space, referred to as **SOLID** or **Reference** in file descriptions, consists of a numeric string prefixed (suffixed if reversed) by a single base. The base that precedes the numeric (color code) data is the first base of the actual sequence (in base space, not color space). See Appendix I for details.

A quick way to recognize if data are pre- or postprocessed is to look at the bases in all the reads from a single tag. If the bases are the same, then the data are (notwithstanding some interesting tag applications) probably unprocessed and, therefore, the base is the last base of the adapter. Also, preprocessed color space has n color space entries (+1 base), but the processed data have n −1 entries (+1 base).

Unprocessed Data	Processed Data
n color space calls	n −1 color space calls
1 base prefix	1 base prefix or suffix
base = last adapter base	base=first base of sequence
files:	files:
xxxx.csfasta	xxxx.bc.txt
xxxx.ma	

Note that color space data are self complementary so that in some situations when you expect to see complemented data (e.g., reverse) they appear the same because color space is self complementary (for example, AC = 1, TG = 1). See Appendix I for more information.

7. **Mismatches**

   Mismatches counts are mismatches to the reference sequence and have not been edited using the extra information provided by two-base encoding. All mismatches refer to mismatches between the reference sequence in color space.

8. **Zero and One Indexing**

   For preprocessed data, the first color space call is position 1, which refers to the transition between the last base of the adapter and the first base of the read. For processed data, the positions are 0-based so that the first position (the prepended base) is 0 on both strands. The n^{th} position of the tag is numbered (n −1) in the forward direction and – (n −1) in the reverse direction.

9. **Cycle order versus sequence order.**

   The order in which the data are generated (cycle order) is different from the sequence order. This is shown in Figure 2 (See Appendix I – Features of Color space). *All data in these files have been transformed to sequence order, ready for alignment*.

10. **Values**

    Most values in the files are obvious, but three need special explanation:

    - "." (dot) is used in color space to show that there is no call for this position. No data was collected.
    - −1 in a set of calculated values (for example, N2S) means that there is no data for this point (i.e., a "." is in the color space for this position.
    - 0 (zero) in a set of calculated values (for example, N2S) means 0 (zero). The calculated result after rounding is 0.

12 Unique

The term unique is used in several file names, the data in these files only includes sequences that had a single match to the genome

Detailed file descriptions

{Note some of the files have headers denoted by # that refer to the settings used for the analysis, these are not discussed below}

xxxx.csfasta

Color space fasta file which contains the color calls generated for each tag, containing the color space calls with the last base of the template strand adaptor prepended.

Its format is
`>TAG_ID`
`Color_space`

e.g.

```
>1_88_1830_R3
G32113123201300232320
>1_89_1562_R3
G23133131233333101320
```

This file contains all the data that passed filtering, including that which did not match the genome.

xxxx.csfasta.ma.tag_length.number_of_mismatches

File `xxxx.ma.25.2` represents a file containing matching data, with tags the length of 25 bases and with up to 2 mismatches allowed
File `xxxx.ma.45.4` represents a file containing matching data, with tags the length of 45 bases and with up to 4 mismatches allowed
File `xxxx.ma.50.2` represents a file containing matching data, with tags the length of 50 bases and with up to 2 mismatches allowed

Color space fasta file containing matches to the reference sequence:

```
>TAG_ID,LOCATION.MISMATCHES
SOLiD
```

```
>1_90_1917_R3
G31131230201013032203
>1_91_1943_R3,6_48653.1
G13113031322133310310
```

[note the chromosome is prepended to the location]

The term LOCATION.MISMATCHES describes the location of the read on the base space reference sequence (0-based) and the number of mismatches between the read and the reference sequence considering the first position in base space and the remaining

positions in color space. This is preprocessed data in which the last base of the primer is prepended to the color space sequence.

xxxx.unique.csfasta.ma.tag_length.number_of_mismatches

The same as the above (xxxx.csfasta.ma.tag_length.number_of_mismatches), but only reads that uniquely matched are included.

xxxx_.QV.qual

These are FASTA-like files that list the quality values in sequence order (not cycle order) for the color space calls (i.e. the QV maps to the csfasta file result and is calculated at the same time as the color call is made).

```
>TAG_ID
quality values
>97_2040_1850_F3
38 36 26 33 41 26 24 33 28 31 27 23 5 35 32 31 11 10 24 38 22 24 7 12
15 21 12 18 34 31 27 11 15 26 13 14 17 17 13 12 8 5 17 5 12
>97_2040_1898_F3
41 41 41 38 32 29 39 24 23 36 32 38 25 30 28 21 27 33 34 33 24 27 9 35
34 14 30 18 33 8 13 32 10 31 24 7 22 5 27 30 21 5 0 27 9
```

The QV value is calculated using a phred like score

$q = -10 \times \log10 (p)$

where q is the quality value and (p) is the predicted probability that the color call is incorrect

xxxx.unique.bc.txt

Tab-delimited file containing the base changes of the tags that match the genome. It includes all tags that match the genome uniquely as well as a single random placement of tags that match the genome in more than one location. The last column indicates if the tag was placed uniquely or randomly Because this is processed color space data in the SOLiD analyzer sequence, the last base of the template strand adaptor under the primer and the first color space call are replaced with the first base of the template strand tag in base space. . The reference sequence is in the same format with the first position in base space and the remaining positions in color space.

```
TAG_ID  STRAND    SOLiD     REFERENCE LOCATION
  MISMATCHES     BASE_CHANGES PLACEMENT   #_OF_PLACES


1_92_1875_R3    top     C0330101130332001221 C0330101130333001221
1571875 1       13_23      unique

1_94_1682_R3    reverse 21333122121332212l3T 21133122121332212l3T
1738072 1       -17_31     unique
```

The strand refers to the alignment on the provided reference sequence. When a tag matches the reverse strand, it is reversed, and the first position in base space is complemented. The location refers to the first position of the tag (the one in base space) in both top- and reverse-strand matches.

The base changes refer to the 0-based positions on the tag. The first position (the one in base space) is position 0 for tags on both strands.

BASE_CHANGES are in the format:

```
(position on tag)_(SOLiD)(REFERENCE)
```

Example: 14_02: At position 14 in the tag, SOLiD = 0, Reference = 2.
If there are multiple mismatches, each is comma separated (e.g.

```
12_01,17_10,22_12).
```

At position 12 in the tag, SOLiD = 0, Reference = 1
At position 17 in the tag, SOLiD = 1, Reference = 0
At position 22 in the tag, SOLiD = 1, Reference = 2

xxxx.stats.txt

The stats file contains a summary of the data for that run

```
122812090 total beads found
Total Beads
      0 mismatches 35876554 ( 29.21%)
      1 mismatches 16077243 ( 13.09%) 51953797 ( 42.30%)
      2 mismatches 14851735 ( 12.09%) 66805532 ( 54.40%)
Uniquely Placed Beads
      0 mismatches 23354456 ( 19.02%)
      1 mismatches 11102576 ( 9.04%) 34457032 ( 28.06%)
      2 mismatches 10464148 ( 8.52%) 44921180 ( 36.58%)

Errors within Uniquely Placed Tags
      Total Errors 32030826
      Single Errors 28439956 (88.79% of Total)
      Adjacent Errors 1795435 (11.21% of total)
      Valid 1151742 (7.19% of Total) (64.15% of Adjacent Errors)
      Invalid 643693 (4.02% of Total) (35.85% of Adjacent Errors)

Starting Points within Uniquely Placed Tags
      Number of Starting Points 42463277
      Average Number of Reads per Start Point 1.06
      Starting Points within Uniquely and Randomly Placed Tags
      Number of Starting Points 60918201
      Average Number of Reads per Start Point 1.10

Coverage of Uniquely Placed Tags
      Bases Not Covered 2252956407(73.14%)
```

.

Beads found is the number of beads initially identified. Many of these are not data producing beads. All percentages are referenced to this initial number. Depending on the emulsion dilution, 5–20% of the beads that have DNA on them are doublets that will not match the genome. Likewise, current single-round enrichment provides 80% amplicon positive beads. The amplicon empty beads are usually still found by the bead finder because they have a slight auto-fluorescense, these beads can be filtered based on QV. .
The matching statistics show the number of beads that mapped at each level of mismatch, the second column is the cumulative percentage. The Uniquely matching beads refers to sequences that had only one match to the genome
Starting points refers to the number of start points seen (If two reads started at the same base that would count as one starting point)
Adjacent errors (two consecutive mismatches) were split into valid (a change in color space that was permitted given the initial reference sequence marking a potential SNP) and invalid (a non-allowed change in color space that was permitted given the initial reference sequence, and thus not a SNP). For any two adjacent color space bases in the reference sequence, 3 color transitions are valid, and 6 are invalid.
Coverage of uniquely placed tags shows the number of bases of the reference genome that were not covered by at least one uniquely placed read.

F3_R3.mates

This file reports the paired tags (as a single line) and the category that pairing falls into

```
TAG_ID  F3_SEQUENCE     R3_SEQUENCE_F3_MISMATCHES      R3
MISMATCHES   TOTAL_MISMATCHES     F3_CHROMOSOME
   R3_CHROMOSOME   F3_POSITION     R3_POSITION     CATERGORY


173_1532_1532  T000323333033002000003231 32     G203110022030210301131230 0
1     0     1     1     1     -247196258     -247198167     AAA
2183_872_1369  T033121120311222101133322 3     G020002332000011303300211 02
2     1     3     1     1     -247196151     -247197397     AAA
```

- Category refers to the orientation and alignment of the tags it consists of three values e.g. AAA
    - The first value refers to orientation A= correct orientation (i.e. both tags are on the same strand and read in same direction) and B= incorrect orientation. (i.e. tags are not on the same strand)

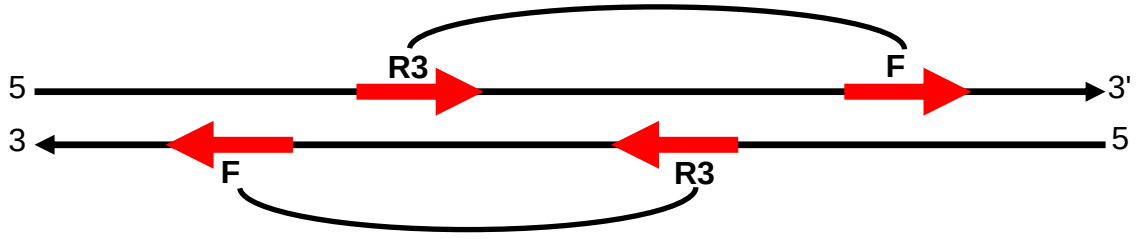

figure showing correct orientations

- The second value refers to correct ordering where A=correct order (i.e. reading from 5′ to 3′ the R3 read is first and the F3 is second (see above)) and B = incorrect order
- The third value refers to the insert (distance on reference genome between R3 and F3) size A= correct (within set parameters) insert size , B= smaller than expected insert size, C= larger than expected insert size. It is very important to realize that these values are relative to the reference, thus it does not mean that individual mate pair is truly good or bad it could mean there has been some rearrangement relative to the supplied reference.

Thus:

AAA: same reference, same strand, correct ordering, acceptable insert size
AAB: same reference, same strand, correct ordering, small insert size
AAC: same reference, same strand, correct ordering, large insert size
BAA: same reference, different strands, acceptable insert size
BAB: same reference, different strands, small insert size
BAC: same reference, different strands, large insert size
ABA: same reference, same strand, incorrect ordering, acceptable insert size
ABB: same reference, same strand, incorrect ordering, small insert size
ABC: same reference, same strand, incorrect ordering, large insert size
BBA: same reference, different strands, incorrect ordering, acceptable insert size
BBB: same reference, different strands, incorrect ordering, small insert size
BBC: same reference, different strands, incorrect ordering, large insert size
C**: different references

F3_R3.mates.unique

The same as F3_R3.mates but only amongst mate pairs with the same F3 and R3 start points, only one mate pair is included.

xxxx.errors.cv

The errors.cv file summarizes the observed mismatches by position in the read for all of the uniquely placed tags in the genome.

It consists of two comma separated columns

```
POSITION_MISMATCH   COUNT

2_30   34226
2_31   34524
```

2_32 26414
3_01 24530
3_02 32269
3_03 13946
3_10 39608
3_12 12073

The first column contains the zero based position on the read , followed by the observed and then the expected (reference) color, the next count is the number of times this mismatch was seen at this position, thus :

3_03 13946

Means at position 3 on the read (numbering starts at 0), 13946 times when the observed was a 0 (blue) a 3 (red) was expected.

There is a special case for the 0 position

0_AC 8790
0_TC 8941
0_CT 53109

Since the first color space call represents the last base space of the sequencing template strand adaptor and the first base of the target sequence it is possible to calculate the first base of the target sequence. By using this information all the data can be used (as this first color space call will not match the color space reference).
Thus 0_AC represents the first color space base and A was observed whilst C was expected.

It is important to note that this data is prior to application of the two base encoding rules. As the bulk of these mismatches will not correspond to allowable two base encoded changes it is easy to filter out the measurement errors (cases where the wrong color was called) from the biological interesting data (SNPs, indels or rearrangements).

Appendix I – Features of Color space

Relationship Between Cycle and Base Position

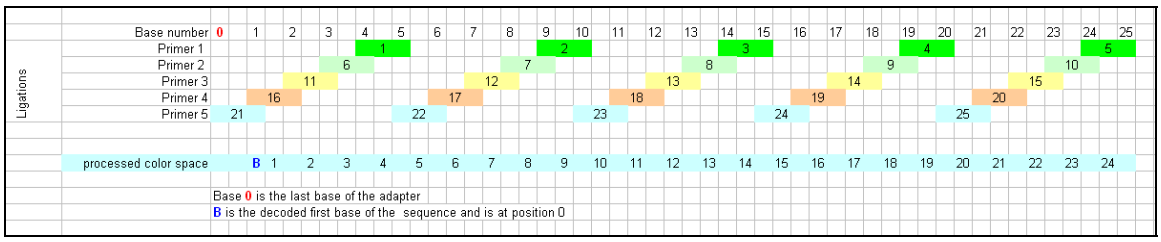

Figure 2. Relationship between ligations and sequence ordered data for a 25-base read

Figure 2 shows the relationship between color space, base position, and the sequencing chemistry. Base number refers to base position. Base 0 is the last base of the adapter and is not part of the target sequence. The five primer lines show the order in which the data was generated (0 indexed).

Note: In all files, only processed data has any reference to a color space position.

Complementing Color Space Data

Color space data are self-complementary as shown in the following matrix:

1st Nucleotide \ 2nd Nucleotide	A	C	G	T
A	0	1	2	3
C	1	0	3	2
G	2	3	0	1
T	3	2	1	0

Sequence

```
Base        A G C T C G T C G T G C A G
Color space 2 3 2 2 3 1 2 3 1 1 3 1 2
```

Complemented

```
Base        T C G A G C A G C A C G T C
Color space 2 3 2 2 3 1 2 3 1 1 3 1 2
```

Two-Base Encoding and Error Recognition

The error-checking abilities of the two-base encoding schemes have **not** been used to correct any of the data provided in these files.

Example:
A single color space error is seen, e.g.

```
Reference 2 3 2 2 3 1 2 3 1 1 3 1 2
Observed  2 3 2 2 0 1 2 3 1 1 3 1 2
```

In this example, the most likely explanation for the observed 0 is that it is a measurement error. Because a single color space change is not allowed, a change to one of the adjacent bases is needed for a real SNP. This requires multiple measurement errors, leaving the most likely explanation that the 0 is a 3. The fact that the two surrounding bases are the same as the reference is further evidence that correcting the 0 to a 3 is acceptable. This means that any single color error can likely be corrected, especially when using multiple aligned reads.

Appendix 2

Mapping tool

The Color Space Mapping Tool

It takes two input files

1. The color space reads in a .csfasta file (see p10)
2. A reference sequence in fasta format in base space (i.e, ACGT)

And outputs a

3. csfasta.ma file (see p10)

Color Space Mapping Tool readme

```
MATCHING READS TO A REFERENCE SEQUENCE


Two files are required to match (or align) reads to a reference
sequence:
      1. The color space reads in a .csfasta file
      2. A reference sequence in fasta format in base space (i.e, ACGT)



If the reference sequence is a multi-entry fasta file, it must first be
concatenated into a single sequence.

To concatenate a reference sequence, run concatenate_sequences.pl

      Usage: concatenate_sequences.pl -f <fasta_file> -o <outputfile> -
h <header>

      Example: perl concatenate_sequences.pl -f
my_multi_entry_fasta_file.fasta -o my_concatenated_fasta_file.fasta -h
what_i_want_the_fasta_header_of_my_new_sequence_to_be

Note: If the reference sequence is a single entry fasta file, the
concatenation step can be skipped and this file can be input directly
to fasta2match.pl
```

To match the reads against a reference sequence, run fasta2match.pl

```
      Usage: fasta2match.pl

         REQUIRED
         -g <genome_file>
         -r <reads_file>
         -d <output_directory>
         -t <tag_length>
         -e <number_of_errors>

         OPTIONAL
         -s <schema_file>
         -p <pattern: defaults to all 1's>
         -start <start: defaults to 0>
         -a <count adjacent errors as 1: 0 = no : 1 = valid adjacent
errors : 2 = all adjacent errors : defaults to 0>
         -z <maximum number of hits per tag: defaults to 1000>
         -ref <output the reference sequence of hits: 0 = no : 1 = yes
: defaults to 0>


      Examples:

            1. A standard use case would be to match reads of length 25
allowing up to 3 errors

                  perl fasta2match.pl -g
my_concatenated_fasta_file.fasta -r my_reads_file.csfasta -d 25_3 -t 25
-e 3


            2. To mask any of the positions in the reads so that they
will not be counted as an error, use the -p flag and indicate a 1 for
every position that is to be counted and a 0 for every position that is
not to be counted.  For example, to mask positions 16 and 21:

                  perl fasta2match.pl -g
my_concatenated_fasta_file.fasta -r my_reads_file.csfasta -d 25_3 -t 25
-e 3 -p 111111111111110111101111


            The -s, -start, -a, -z and -ref options are for special use
cases and are rarely used.
                  -s allows the user to input their own schema file for
matching rather than use one that is provided
                  -start refers to the coordinate of the first position
of the reference sequence
                  -a allows two adjacent errors or two valid adjacent
errors to be counted as one error
                  -z allows the user to define the maximum number of
hits that will be reported for each tag
                  -ref allows the user to have the reference sequence
at the location of each hit reported
```

Each read in the .csfasta file must be at least as long as the user indicated tag_length.

The locations of matches are zero-based and the dots in a concatenated sequence count as a position.

The system requirements for running the binary files called by fasta2match.pl are Linux x86_64, preferably a RedHat 4 or compatible system.

The matching algorithm internally converts the reference sequence to dibase encoding.
If you want to generate a dibase-encoded reference sequence, run convert_to_dibase_encoding.pl

Usage: convert_to_dibase_encoding.pl -s <sequence_file> -o <outputfile>

Example: perl convert_to_dibase_encoding.pl -s my_concatenated_fasta_file.fasta -o my_concatenated_fasta_file.dibase.fasta