SOLiD™ Analysis Tools (SAT)

User Guide

Applied Biosystems

For Research Use Only. Not for use in diagnostic procedures.

Information in this document is subject to change without notice. Applied Biosystems assumes no responsibility for any errors that may appear in this document.

APPLIED BIOSYSTEMS DISCLAIMS ALL WARRANTIES WITH RESPECT TO THIS DOCUMENT, EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO THOSE OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. IN NO EVENT SHALL APPLIED BIOSYSTEMS BE LIABLE, WHETHER IN CONTRACT, TORT, WARRANTY, OR UNDER ANY STATUTE OR ON ANY OTHER BASIS FOR SPECIAL, INCIDENTAL, INDIRECT, PUNITIVE, MULTIPLE OR CONSEQUENTIAL DAMAGES IN CONNECTION WITH OR ARISING FROM THIS DOCUMENT, INCLUDING, BUT NOT LIMITED TO, THE USE THEREOF.

Notice to Purchaser: License Disclaimer

Purchase of this software product alone does not imply any license under any process, instrument or other apparatus, system, composition, reagent or kit rights under patent claims owned or otherwise controlled by Applera Corporation, either expressly or by estoppel.

THIS IS A PRELIMINARY USER'S DOCUMENT OR PROTOCOL. THE COMPLETE LIMITED LICENSE STATEMENTS SPECIFYING THE PATENT RIGHTS CONVEYED WITH THIS SOLID™ SYSTEM (or SOFTWARE) WILL BE PROVIDED WITH YOUR FINAL USER'S DOCUMENTATION OR PROTOCOL, WHICH WILL BE SENT TO YOU UNDER SEPARATE COVER.

TRADEMARKS:

Part Number 4392959 Rev. B
April 2008

Contents

Preface

Safety Information

Chapter 1 Data Files and Processing

Chapter 2 SOLiD™ Analysis Tools Pipeline

Chapter 3 Analysis Stages and File Layout

Appendix A Examples of moap_analysis.ini Files

Appendix B Data Management

Appendix C Software Warranty Information

Glossary

Index

Preface

This section covers:

How to Use This Guide

Purpose of This Guide

This Guide provides:

- A high level overview of data processing with the SOLiD™ Experiment Tracking System (SETS) Software
- A description of the most important files generated
- A description of the SOLiD™ Analysis Tools (SAT) pipeline
- Locations of the files generated by analysis
- A summary of key aspects of color space.

Audience

This guide is intended for Advanced users of SOLiD™ Analysis Tools(SAT).

Assumptions

This guide assumes that your SOLiD™ System has been installed by an Applied Biosystems technical representative.

This guide also assumes that you have a working knowledge of the Microsoft® Windows® XP operating system and Linux.

Text Conventions

This guide uses the following conventions:

- `Code` text indicates user action or examples of code that appear in the software. For example:

  ```
  mate.pairs.run=1
  ```

- *Italic* text indicates new or important words and is also used for emphasis. For example:

  Before analyzing, *always* prepare fresh matrix.

- A right arrow symbol (▸) separates successive commands you select from a drop-down or shortcut menu. For example:

  Select **File ▸ Open ▸ Spot Set**.

  Right-click the sample row, then select **View Filter ▸ View All Runs**.

User Attention Words

Two user attention words appear in Applied Biosystems user documentation. Each word implies a particular level of observation or action as described below:

Note: – Provides information that may be of interest or help but is not critical to the use of the product.

IMPORTANT! – Provides information that is necessary for proper instrument operation, accurate chemistry kit use, or safe use of a chemical.

Examples of the user attention words appear below:

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

IMPORTANT! It is important to realize that these values are relative to the reference.

Safety Alert Words

Safety alert words may also appear in user documentation. For more information, see "Safety Alert Words" on page xii.

How to Obtain More Information

Related Documentation

The following related documents are shipped with the system:

- ***SOLiD^{TM} System v2.0 Site Preparation Guide*** **(PN 4386998)–** Describes how to ready your location for a SOLiD™ System installation.
- ***SOLiD^{TM} System v2.0 User Guide*** **(PN 4391587) –** Provides detailed descriptions of how to use the SOLiD™ System.
- ***SOLiD^{TM} System v2.0 SETS Software Getting Started Guide*** **(PN 4389302) –** Provides brief, step-by-step procedures for SOLiD™ SETS Software. It is designed to help you quickly learn to use the software.

Send Us Your Comments

Applied Biosystems welcomes your comments and suggestions for improving its user documents. You can e-mail your comments to:

techpubs@appliedbiosystems.com

IMPORTANT! The e-mail address above is only for submitting comments and suggestions relating to documentation. To order documents, download PDF files, or for help with a technical question, go to **http://www.appliedbiosystems.com**, then click the link for **Support**. (See "How to Obtain Support" below).

How to Obtain Support

For the latest services and support information for all locations, go to **http://www.appliedbiosystems.com**, then click the link for **Support**.

At the Support page, you can:

- Search through frequently asked questions (FAQs)
- Submit a question directly to Technical Support
- Order Applied Biosystems user documents, MSDSs, certificates of analysis, and other related documents
- Download PDF documents
- Obtain information about customer training
- Download software updates and patches

In addition, the Support page provides access to worldwide telephone and fax numbers to contact Applied Biosystems Technical Support and Sales facilities.

Safety Information

This section covers:

Safety Conventions Used in This Document

Safety Alert Words

Four safety alert words appear in Applied Biosystems user documentation at points in the document where you need to be aware of relevant hazards. Each alert word—**IMPORTANT, CAUTION, WARNING, DANGER**—implies a particular level of observation or action, as defined below:

Definitions

IMPORTANT! – Indicates information that is necessary for proper instrument operation, accurate chemistry kit use, or safe use of a chemical.

⚠ **CAUTION** – Indicates a potentially hazardous situation that, if not avoided, may result in minor or moderate injury. It may also be used to alert against unsafe practices.

⚠ **WARNING** – Indicates a potentially hazardous situation that, if not avoided, could result in death or serious injury.

⚠ **DANGER** – Indicates an imminently hazardous situation that, if not avoided, will result in death or serious injury. This signal word is to be limited to the most extreme situations.

Examples

The following examples show the use of safety alert words:

IMPORTANT! The sample name, run folder name, and path name, *combined*, can contain no more than 250 characters.

⚠ **CAUTION** **MUSCULOSKELETAL AND REPETITIVE MOTION HAZARD.** These hazards are caused by potential risk factors that include but are not limited to repetitive motion, awkward posture, forceful exertion, holding static unhealthy positions, contact pressure, and other workstation environmental factors.

⚠ **WARNING** Do not attempt to lift or move the computer or the monitor without the assistance of others. Depending on the weight of the computer and/or the monitor, moving them may require two or more people.

General Instrument Safety

⚠ **WARNING** **PHYSICAL INJURY HAZARD.** Use this product only as specified in this document. Using this instrument in a manner not specified by Applied Biosystems may result in personal injury or damage to the instrument.

Moving and Lifting Stand-Alone Computers and Monitors

⚠ **WARNING** Do not attempt to lift or move the computer or the monitor without the assistance of others. Depending on the weight of the computer and/or the monitor, moving them may require two or more people.

Things to consider before lifting the computer and/or the monitor:

- Make sure that you have a secure, comfortable grip on the computer or the monitor when lifting.
- Make sure that the path from where the object is to where it is being moved is clear of obstructions.
- Do not lift an object and twist your torso at the same time.
- Keep your spine in a good neutral position while lifting with your legs.
- Participants should coordinate lift and move intentions with each other before lifting and carrying.
- Instead of lifting the object from the packing box, carefully tilt the box on its side and hold it stationary while someone slides the contents out of the box.

Workstation Safety

Correct ergonomic configuration of your workstation can reduce or prevent effects such as fatigue, pain, and strain. Minimize or eliminate these effects by configuring your workstation to promote neutral or relaxed working positions.

⚠ **CAUTION** **MUSCULOSKELETAL AND REPETITIVE MOTION HAZARD.** These hazards are caused by potential risk factors that include but are not limited to repetitive motion, awkward posture, forceful exertion, holding static unhealthy positions, contact pressure, and other workstation environmental factors.

To minimize musculoskeletal and repetitive motion risks:

- Use equipment that comfortably supports you in neutral working positions and allows adequate accessibility to the keyboard, monitor, and mouse.
- Position the keyboard, mouse, and monitor to promote relaxed body and head postures.

Data Files and Processing 1

In This Chapter

This chapter covers:

Analysis Processes

SOLiD^{TM} System Analysis consists of these main processes:

- **Primary analysis** – Images for each cycle are analyzed. 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 aligned to the reference sequence. If the experiment is a mate-paired run, the system analyzes the mate pairs.

Note: The SAT pipeline discussed on page 2-2 is used to perform secondary analysis.

Color Space and Base Space

The final files generated by the SOLiD System are in base space (see gff file, consensus, and SNP files). To maximize the built-in error correction of two-base encoding, prior to final results all file analysis is conducted in color space.

Color Space Data

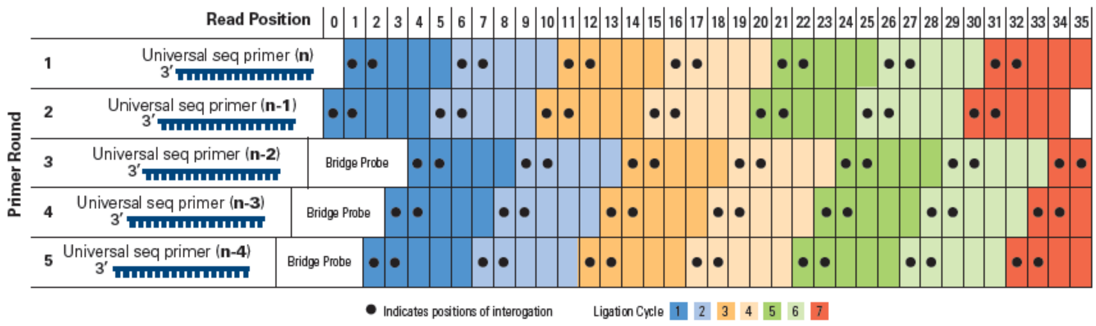
Rather than reading one base per cycle, the software measures information on two bases simultaneously, and in each cycle, calls one of four colors (color space call). Since each ligation measurement event measures 2 bases, all bases (except for the final base of a read) are interrogated twice, providing for an additional level of error correction.

Note: Users unfamiliar with color space should download the descriptions of two-base encoding from http://solid.appliedbiosystems.com.

Relationship Between Cycle and Base Position

The graph below 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 refers to a color space position.



Color Space Formats

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.

Note: Color space data are self-complementary: In some situations, when you might expect to see complemented data (for example, reverse), they appear the same. For example, AC = 1, TG = 1.

The different types of color-space data are:

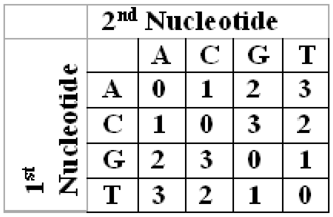
- **Processed color-space data**

  Processed color-space data 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 "xxx.gff File Format" on page 1-8 for more information.
- **Unprocessed color-space data**

  Unprocessed color-space data 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. It is included to disambiguate the first color call. See "xxxx.ma.tag_length.number_of_errors" on page 1-7 for more information.

Complementing Color Space Data

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



		2nd Nucleotide			
1st 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:

Reference = 2 3 2 2 **3** 1 2 3 1 1 3 1 2

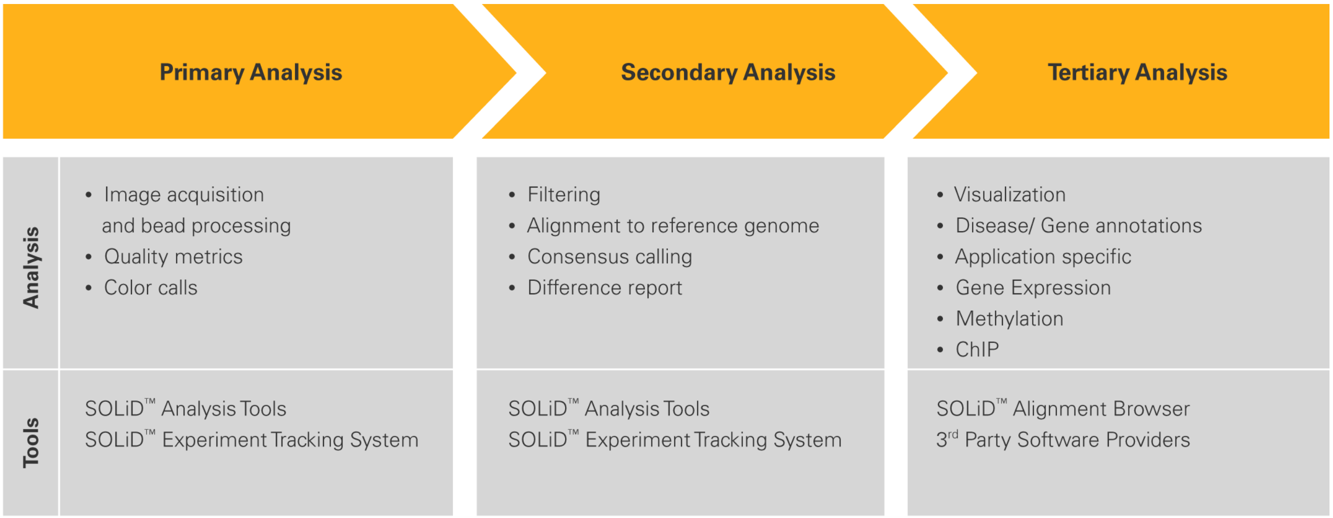
Observed = 2 3 2 2 **0** 1 2 3 1 1 3 1 2

Note: Notice the single color space error.

In this example, a single color space error occurs. 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 correction 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. Any single color error can likely be corrected, especially when using multiple aligned reads.

Processing

The flow of data through the SAT pipeline is shown in Figure 1-1.



	Primary Analysis	Secondary Analysis	Tertiary Analysis
Analysis	• Image acquisition and bead processing • Quality metrics • Color calls	• Filtering • Alignment to reference genome • Consensus calling • Difference report	• Visualization • Disease/ Gene annotations • Application specific • Gene Expression • Methylation • ChIP
Tools	SOLiD™ Analysis Tools SOLiD™ Experiment Tracking System	SOLiD™ Analysis Tools SOLiD™ Experiment Tracking System	SOLiD™ Alignment Browser 3rd Party Software Providers

Figure 1-1 Data flow through the SAT pipeline

Typically, a 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 and baselined, quality values are assigned, and color-call is assigned.

Results of primary analysis for each panel are stored in files with the .spch extension (Solid Panel Cache HDF5). The spch file is used as a cache; that is, primary analysis both writes results to the file and reads results from the file as analysis proceeds on a cycle-by-cycle basis. Data within the spch file is organized by cycle. When all cycles have been analyzed, a final processing step is run to convert the data from the cycle-based format to the "read-based" ASCII formats (.csfasta and .qual).

The spch file is in the HDF5 format. Details on the format are available at http://hdf.ncsa.uiuc.edu/products/hdf5/index.html.

Tools to view the contents and extract results are available at http://hdf.ncsa.uiuc.edu/hdf-java-html/hdfview/index.html.

Prior to secondary analysis, results are filtered by removing all tags with missing data. Filtering removes all incomplete beads and those reads with missing calls. The filtered data are then processed to generate the file *xxxx.csfsta*. As part of this process, a known base (from the adapter sequence) is prepended to the color-space data, allowing disambiguation of the first color call. 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 color-space data are used as the basis of all subsequent results. The data are then aligned in color space to the color space reference sequence. Note the following:

- Color space reference sequences are derived by converting the base-space reference sequence to color space (this is done internally; you supply their reference sequence in base space).
- All alignment is done in color space.
- All the alignments are with individual tags; the mate-paired information is not used in the initial alignments.
- 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 the tags where both the forward and reverse tags passed (both tags were matched to a reference sequence) are analyzed.

    2. Then, those mate pairs where only one of the tags was matched to the reference sequence (in color space) are reanalyzed, and the nonmatching tag is compared to the reference sequence.

  Because analysis 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.

Note: Where possible, all data are presented in a fasta-compatible format to facilitate use in a variety of applications. Final output files are in base space and relative to the submitted reference (e.g., consensus file, variation file).

SOLiD™ Analysis Tools—Primary Analysis

The key files generated by primary analysis are:

- **xxxx_sequence.csfasta**—the raw data from all beads
- **xxxx_sequence.QV.qual**—the quality values file

xxxx_sequence.csfasta

The .csfasta file is a color space fasta file that contains the color calls generated for each tag, with the last base of the primer prepended.

Its format is:

```
>TAG_ID
Color_space
```

For example:

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

This file contains all the data that passed filtering, for complete reads (for example, 35 base read has 35 bases). This file is the input to the alignment tool.

xxxx_sequence.QV.qual

The QV.qual files are FASTA-like files that list the quality values in sequence order (not cycle order).

```
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
```

Note: All primary data (with exception of images) is stored in the file xxxx.spch >. This file is in the hdf5 format. Tools to view and extract this data are available at http://hdf.ncsa.uiuc.edu/HDF5-FAQ.html. The data within the .spch file is organized by cycle. You can extract data in any format for your own analysis pipelines.

SOLiD™ Analysis Tools—Secondary Analysis

The results of secondary analysis are color space calls that have been matched to the reference sequence. These data are ready for use in an analytical context.

Key files include:

```
xxxx.sequence.csfasta—input to analysis
xxxx.sequence.csfasta.ma.xx.xx—initial output from alignment tool
F3toR3 file
xxxx.gff
consensus file
SNP file
xxx.bc.txt
```

In all files below, xxxx is the file name consisting of a run name along with extra information (i.e., spot number, or number of cycles).

xxxx.ma.tag_length.number_of_errors

File *xxxx.ma.25.3*

This file is the output from the alignment tool containing matching data, with tags the length of 25 bases and with up to 3 mismatches allowed.

Color space fasta file containing matches to the reference sequence:

```
>TAG_ID,LOCATION.ERRORS
SOLiD
>1_90_1917_R3
G3113123020101303220 3
>1_91_1943_R3,48653.1
G1311303132213331031 0
```

The term LOCATION.ERRORS describes the location of the read on the base space reference sequence (0-based) and the number of errors (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.

xxx.bc.txt

This file types is a 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 primer and the first color space call are replaced with the first base of the 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 ERRORS
BASE_CHANGES PLACEMENT
```

```
1_92_1875_R3    top      C0330101130332001221
    C0330101130333001221    1571875 1       13_23 unique
```

```
1_94_1682_R3    reverse 2133312212133221213T
2113312212133221213T    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`

xxx.gff File Format

Overview of .gff v1 Files

GFF (General Feature Format) is a record-based file format, where each line describes a single *feature* (in this case, a *read*) with a list of tab-delimited *fields* in a fixed order specified by the GFF specification. Because the format has been in widespread use for many years, there are many programs and bioinformatics libraries which support reading and writing to this type of file.

The default file type created by the SOLiD™ System is the .gff v1 file. The particular specialization of GFF used in the SAT pipeline was introduced to adapt the format to represent aligned color-space reads and allow for rapid visualization in a tool like the Apollo genome browser. Here is an excerpt from such a file:

```
3_6_737_F3 corona exon 180614 180632 0.000000-.
g=CGCTAcTATCATCGCGGTA;f=A;s=r6
```

- The first column is the read identifier for this read, with the suffix representing the primer set that this sequence is associated with.
- The second column is a track name for identifying what type of data is represented by this record. We use the track name "corona" to indicate reads from the SOLiD™ instrument.
- The third column is the type of feature represented by this record. The use of the feature name "exon" is a legacy implementation, to be replaced in the future – it is used to provide automatic display of mate-paired reads as multi-exon transcripts).

- – The fourth and fifth columns define the start and end position of this read on the genome.
- – The sixth column is an optional score, typically 0.0 in the current pipelines, which will be used in the future to denote a quality score.
- – The seventh column provides which orientation of the genome was used for this read alignment (either '+' or '-').
- – The eighth column represents the translational frame in the GFF format; it is not used in our implementation.
- – The ninth column contains key-value data, with key-value pairs separated by ';' and denoted by "key=value".

The keys which are used in our current implementation are:

- •g - provides the double-encoded sequence of the read. Double encoding refers to a representation of the 0,1,2,3 color-space calls as A,C,G,T. If the file has been annotated then lower-case bases represent differences from the reference.
- •f - provides the first base (in base space) of this read. This is the base after the primer base, after decoding the first color-space call.
- •s - provides color annotations for the SAB viewer. This is a comma-separated list of codes of the form "[ryb]##" where the first letter represents the color ('r'=red,'y'=yellow, etc...) and the number represents the position to be colored in the read sequence.
- •c - provides a condensed count of this sequence. If the same read sequence appears multiple times in an experiment, this provides a means for condensing the data and conveying information for how many counts were observed.

The GFF format is convenient for its standardization and widespread integration in genomics tools but it is not an ideal format for efficient storage or I/O access for gigabase sequencing experiments.

Note: Applied Biosystems will be moving to a more efficient and indexable binary format as the SOLiD™ platform evolves and as the community develops standards for storage of next-generation sequencing data.

Note: Use of double encoding to represent the reads is a legacy feature. The A,C,G,T characters found in this file do not correspond to actual bases but to the color calls.

Manual Creation of .gff v2 Files

Use the command:

```
gffv2_module.sh MainClass
```

Where MainClass is first MaToGff and then AnnotateChanges.

MaToGff is called with the output of the matched .csfasta file in the following manner (Usage):

```
MaToGff <maFile> [--convert=[unique|mapped|all]] [--first]
[--sort] [--qvs=<qvFile>] [--tempdir=<dirName>] > gffFile
```

MaToGff converts a FASTA file, maFile, of unmated reads to a SOLiD GFF v0.2 file.

```
--convert=unique
```

Output only reads that map uniquely to the reference

```
--convert=mapped
```

Output only reads that map anywhere to the reference

```
--convert=all
```

Output all input reads.

Note: --convert defaults to 'unique' if it is not specified on the command line.

```
--clear=<int>
```

The requested 'clear zone', used if --convert=unique.

Let u_k be the number of alignments in which the read mapped to the reference with k mismatches, and let k' be the least value k for which u_k != 0. Then the read maps uniquely if u_k' = 1 and u_k = 0 for all k in the interval [k'+1, k'+clear]. That is u_k' = 1, and 'clear' is the number of zeroes that must follow k' in u' (up to the end of vector u).

```
--qvs=<qvFile>
```

A file of quality values for the color calls in the reads.

For each read, MaToGff reports the position of the best alignment to the reference (i.e., the alignment with the least number of mismatches). If MaToGff finds more than one best position, it reports a position at random from a uniform distribution of the best candidates.

You can override this random behavior by setting the --first option, which causes MaToGff to report the first position instead, that is, to report the alignment with the least "start" position.

Note: FASTA and GFF reads are 2-bp encoded strings.

--tempdir is the directory that MaToGff should use for any temporary files that the system creates (defaults to \scratch). The system will delete those files, but not the directory, upon termination.

AnnotateChanges is called with the output gff file from MaToGff, then the base space reference file that was used in the read mapping portion of the analysis.

Usage:

```
AnnotateChanges <GFF_file> <FASTA_ref_file> [--tints=a[g[y[r]]]]
[--b] [--cn]
```

AnnotateChanges reads in a GFF file of color reads and a FASTA file that contains the reference base sequence to which the reads have been mapped. For each read, AnnotateChanges creates the 'r' (reference call at mismatch), 's' (mismatch annotations), and 'b' (base space representation) annotations.

Note: AnnotateChanges writes its result (a new GFF file) to Standard Out.

<GFF_File> is the name of the input, unannotated GFF file.

<FASTA_ref_File> is the name of the reference file.

--tints=agyr represents any number (four in this example) single-tint annotations.

- a = Isolated single-color mismatches(red)
- g = Color position that is consistent with an isolated one-base variant (e.g. a SNP)
- y = Color position that is consistent with an isolated two-base variant
- r = Color position that is consistent with an isolated three-base variant

and so on, for any number of adjacent base variants. You can use any tints for a,g,y, and r; it is their position that matters. You do not have to extend as far as 'r', nor do you have to stop there (if biologically realistic). The default is equivalent to -tints=agyr.

--b Requests the 'b' attribute, the base-sequence corresponding to the corrected color calls.

--cn Requests AnnotatedChanges to print the names of the contigs in the reference file. Format will be, for example:

```
##contig 5
```

Specification of .gff v2 Files

If a ‘#’ character appears anywhere on a line, the rest of that line is a *comment*. Ordinarily, an application reading the file ignores comments and *comment lines* (lines starting with ‘#’ after white space). Comment lines that begin with ‘##’ are called *metadata*, data that apply to all of the features in the file. These metadata, defined in Section 3.1, need not be ignored by all applications.

Metadata

The GFF format allows for the use of metadata in the form of ‘##’ comment lines, including the following standard GFF metadata tags:

`##solid-gff-version`

This is currently “`##solid-gff-version 0.2`”.

`##gff-version`

This is “`##gff-version 0.2`”.

`##source-version <source> <version text>`

A comment describing the version of the program that produced this file. There should be no spaces in the source and the version text. For an acceptable example:

```
##source-version Ma_to_gff.java v0.2
```

##date <date>

The date that the file was generated, in yyyy-mm-dd format. This example designates 1 October 2007:

```
##date 2007-10-01
```

##time <current time>

The local time, in 24 hour format, when the generating software wrote this line. For example,

```
## time 18:02:36
```

##Type <type>[<reference name>]

The type is set to **solid_read**. The reference name is the name of the reference to which all reads in this file were aligned.

##history <command line>

These comment lines allow recording of source information in addition to the program name, which is recorded by the ##source-version tag, and to record how previous processing steps lead to the data in the current SOLiD GFF v0.2 file. There is one history line for each processing step, with earlier steps appearing above later ones. It may not always be feasible to record all aspects of a command line, such as its file indirections, for example.

```
##history map Sample1_F3.csfastamyReference.fasta
##history Ma_to_gff.java --convert=unique –sort Sample1.ma
##history AnnotateChanges.java Sample1.gff myReference.fasta
##history filter_fasta.pl --noduplicates
--output=qvs.qual
##history AddQvs.java Sample1.gff qvs.qual
```

##color-code <code string>

This line specifies the color code used to generate *all* color reads in this file. The code-string is a comma-separated string of <motif>=<code> pairs. For example, the following entry specifies our current two-base encoding:

```
##color-code
AA=0,AC=1,AG=2,AT=3,CA=1,CC=0,CG=3,CT=2,GA=2,GC=3,GG=0,GT=1,TA=3,
TC=2,TG=1,TT=0
```

##primer base <code string>

A comma separated string of <primer set>=<base> pairs, each of which specifies the last base for each primer set in this file. For example,

```
##primer_base F3=T,R3=G
```

Fields

The feature lines specify one read each, with the following fields, as specified in the GFF standard.

`seqname` Field

The name is the bead identifier (panel_x_y) plus a suffix indicating the primer set id (either "_F3" or "_R3" currently). For example:

```
8_1727_1389_F3
```

`source` Field

This name is always `solid`.

`feature` Field

This name is always `read`.

`start` Field

The inclusive start point of the aligned read on the 1-based base-space reference sequence, indexed from 5′ to 3′. See 'end' for details.

`end` Field

The inclusive end point of the aligned read on the 1-based base-space reference sequence, indexed from 5′ to 3′.

By the GFF specification, start must be less than or equal to end. For example, if the read aligns to the forward strand of the reference (see strand field below), as in:

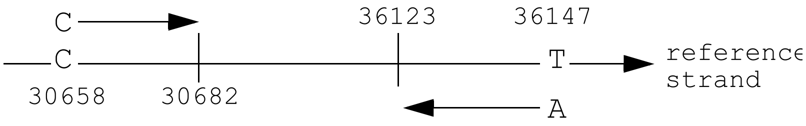
```
start    end      strand   [attributes]
30658    30682    +        g=C010311200313021323311032
```

The initial 'C' aligns with position 30658 of the reference strand, which should be a matching 'C'; the following '0', which corresponds to another 'C' (by the color code in §3.1.7), aligns with position 30659; and (the base corresponding to) the last '2' aligns with position 30682. See the diagram below.

If, on the other hand, the read aligns to the reverse strand of the reference, as in:

```
start    end      strand   [attributes]
36123    36147    -        g=A322121003310310232103022
```

Then the A aligns with position 36147 of the reference strand, which should be a complementing T; the first '3' after the initial 'A', which corresponds to a 'T' by the color code, aligns with position 36146 of the (forward) reference strand, and the last '2' aligns with position 36123.



score Field

A summary quality score for the read, recorded to one decimal place:

$$score = -10 \log_{10} P$$

where P is the average probability of error at any read position:

$$P = \frac{1}{n} \sum_{i=1}^{n} 10^{-QV_i/10}$$

and QV_i is the quality value of the color at read position i (see `q attribute`).

Note that 0 < score < 1, and that high scores correspond to high-quality reads.

strand Field

Either ‘+’, if the read aligns to the “forward,” “nominal,” or “given” strand (that is, the sequence in the database), or ‘-’ if the read aligns to the other strand.

frame Field

SOLiD does not use the frame and therefore always sets it to “.”.

attributes Field

This is a semi-colon separated list of key-value pairs of the form “key=value”. This implementation is slightly different than GFF v2 suggests (the ACEDB format), although the implementation of this field is somewhat free.

- `b` Field Attribute (Optional)

  The corrected base-space representation of the read. Constructed in three steps:

  1. Identify isolated and invalid mismatches (see **s**-attribute definitions) in the read, then replace them with the aligned reference color-calls.
  2. Convert the result to a DNA sequence and align it with the DNA reference.
  3. Replace mismatched nucleotides with the appropriate IUB code.

- `g` Field Attribute

  Required. The color space sequence for this read, written from 5′ (the bead end) to 3′. In addition, prepended to the sequence is the first (5′ most) nucleotide of the DNA target corresponding to the read. This attribute, together with the coding algorithm specified by the color-code metadata tag, allows any application to reconstruct the DNA sequence for the read.

  In addition, by referring to the ‘strand’ metadata tag (and assuming no errors in the data), the application can reconstruct the color-call subsequence and the DNA subsequence for either strand of the reference.

  Note: It is important to note that FASTA files generated by the instrument store their data in a different format. There, the sequence stores the last base of the primer, presumably the phase 5 primer, as its first base.

Fortunately, given the color-code tag, it is easy for any application to convert FASTA notation to GFF notation by reconstructing the first base of the read.

For example, the application converts the FASTA sequence T210033221 to the GFF sequence C10033221 by converting T2 to TC and dropping the primer base T.

The first color reported by 'g' is really the second color in the read from the instrument.

- `i` Field Attribute

  Optional. The 1-based index of the reference sequence. For example, i=3 says that this read is aligned to the third reference sequence in the reference name (see the ##type metadata tag). If this value is not specified, it defaults to 1.

- `p` Field Attribute

  Optional. This mappability measure counts the "effective number of hits" for this read onto the reference. A small value close to 1 indicates that the read is effectively unique in the reference. A higher value indicates that the read effectively matches at multiple locations. More formally, the field attributes have these definitions:

L Length of the read.

k Number of mismatches in an alignment. **Note**: $k \leq L$.

m Number of mismatches in *this* alignment.

N_k Number of reference positions where the read aligns with exactly ***k*** mismatches. **Note**: $N_k = 0$ for all $k < m$ (for an explanation, see "`end` Field" on page 1-13).

X_{Lk} The expected number of alignments with exactly *k* mismatches as a multiple of the expected number of alignments with 0 mismatches.

Then

$$X_{Lk} = 3^k \binom{L}{k} \text{ and } \boldsymbol{mappability} = X_{Lm} \sum_{k=m}^{L} \left(\frac{1}{X_{Lk}} N_k \right)$$

In the following examples, let L=25: For a read with exactly one alignment with no mismatches, m=0, and $N_m = 1$. Then $X_{L,m} = X_{25,\,0} = 1$ and *mappability* = 1, indicating an ideal unique match.

Even if the single alignment has two mismatches, then L=25, m=2, N_m=1, and mappability = 1 (the $X_{L,m}$s cancel out).

For two perfect alignments, L=25, m=0, and $N_m = 2$. Again, $X_{L,m} = X_{25,\,0} = 1$. But now mappability = 2, indicating an ambiguous match. In general, if the read maps perfectly to the reference in *n* places, the mappability measure is also *n*.

A final example shows how more complicated values arise: if $N_0 = 0$, $N_1 = 3$, and $N_2 = 25$, then mappability = 3.417. The three alignments with one mismatch contribute 3 and the 25, with two mismatches contribute the remaining 0.417 to the measure.

- **q** Field Attribute

  Optional quality values. A comma-separated list of quality values, one per color-call. Each quality value is an integer between 0 and 100, exclusive. The value -1 indicates a missing quality value for the corresponding color-call.

- **r** Field Attribute

  Optional reference call at mismatch. A comma-separated list of {position}_{ref_color} for all of the color-calls in the reference sequence that differ from the read sequence. Position is 1-based relative to the sequence specified in the "g" attribute (again, the prepended base has position 1 and the first color in 'g' has position 2.). This differs from the basechange format in that only the reference sequence call is provided and positions are 1-based and positive.

  For example, `r=18_3, 21_1` means that the reference value is 3 at position 18, and 1 at position 21.

- **s** Field Attribute

  Optional. This is a comma-separated string representing annotations on the sequence. The format is '{char}{position}' where {char} is a character representing the type of annotation (typically a formatting request to visualization software) and {position} is the position of this annotation in the read. The position is 1-based on the string recorded in the 'g' attribute. That is, the prepended base has position 1, and the first color has position 2. For example, "r5,y7,y8" means "format base 5 red, format base 7 yellow, and format base 8 yellow." SOLiD follows the convention that:

  – **r** (red) is an *isolated mismatch*; it is a mismatch, and neither the color-call on its left nor the call on its right is a mismatch.

  – **y** (yellow) is a *valid adjacent mismatch*; it is a mismatch and it, together with the adjacent mismatch on its left or right, could correspond to an isolated SNP.

  – **b** (blue) is an *invalid adjacent mismatch*; it is any mismatch other than one specified by red or yellow.

- **u** Field Attribute

  Optional. Mismatch count. This is a comma-separated list of non-negative integers. The i th number specifies the number of positions in the reference to which this read aligns with exactly i – 1 mismatches. For example, u=0,3,15 says that this read does not align to the reference anywhere with exactly 0 mismatches, but that it does align with one mismatch at 3 reference positions, and with exactly two mismatches at 15 reference positions. All unspecified mismatch counts are undefined. This example gives no information about the number of reference positions where the read aligns with exactly four mismatches.

Additional Semantics

Mate-paired data

For mate-paired reads, the reads are represented on two adjacent lines with the F3 read followed by the R3 read.

Line Order

The GFF standard allows lines in any order ("line order is not relevant"). The standard GFF files produced by the SOLiD system, however, should have metadata before feature lines, and feature lines should be sorted by increasing start position of the reads. In the case of mate-paired data, the reads should be sorted by increasing start position of the F3 read, and the F3 read and R3 read should remain adjacent.

Example

```
#gff-version 2
##source-version AnnotateChanges.java v0.2
##date 2007-08-28
##time 09:30:18
##Type solid_read DH10B_WithDup_FinalEdit
##color-code
AA=0,AC=1,AG=2,AT=3,CA=1,CC=0,CG=3,CT=2,GA=2,GC=3,GG=0,GT=1,TA=3,TC=2
,TG=1,TT=0
##generated-by "AnnotateChanges.java test-etc/modules/temp.gff test-
etc/modules/DH10B_WithDup_FinalEdit_validated.fasta
```

```
##hdr seqname     source   feature  start    end      score  strand      frame          [attributes]    [comments]
389_495_172_F3  solid    read     906843  906867  -1 +   g=T2121033030302201123013333;r=18_0,19_1,24_2;s=yl
                                                         8,y19,r24;u=0,0,0,1
389_595_202_F3  solid    read     912290  912314  -1 -   g=T3113211132100321303000120;u=1
```

This is an example of single-primer data which conforms to the specification described above. The reads have been annotated with reference sequence discrepancies, style annotations, and uniqueness counts. The example also shows a convenience "header" line (##hdr), which illustrates that not all comments, not even metadata, need come from this specification.

GFF v2

The GFF specification upon which this file is based can be found at http://www.sanger.ac.uk/Software/formats/GFF/GFF_Spec.shtml

F3_R3.mates

This file reports the paired tags (as a single line) and the category that the pairing falls into. The column headings are:

- Bead
- F3 Sequence
- R3 Sequence
- Number of F3 Mismatches
- Number of R3 Mismatches
- Total Mismatches
- F3 Position
- R3 Position
- Category

Note: *Category* refers to the orientation and alignment of the tags. It consists of three values: XYZ.

Table 1-1 Mate-pair descriptions for the Category Column Heading

# Mate-Pair XYZ	Description
AAA	Correct orientation + ordering and acceptable insert size
AAB	Correct orientation + ordering and small insert size
AAC	Correct orientation + ordering and large insert size
BAA	Incorrect orientation and acceptable insert size
BAB	Incorrect orientation and small insert size
BAC	Incorrect orientation and large insert size
ABA	Correct orientation, incorrect ordering, acceptable insert size
ABB	Correct orientation, incorrect ordering, small insert size
ABC	Correct orientation, incorrect ordering, large insert size

- The first value refers to orientation A = correct orientation; Both tags are on the same strand and read in the same direction. B = incorrect orientation; Tags are *not* reading in the same direction relative to one another.

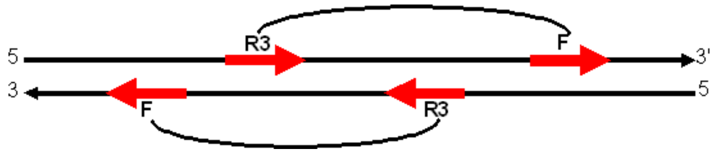


Figure 1-2 F3 and R3 showing correct orientations

- The second value refers to correct ordering. A = correct order; that is, reading from 5´ to 3´, the R3 read is first and the F3 is second (see above). B = incorrect order.
- The third value refers to the insert size (distance on reference genome between R3 and F3).
  - A = correct (within set parameters) insert size
  - B = smaller than expected size.
  - C = larger than expected insert size

IMPORTANT! These values are relative to the reference and therefore an individual mate-pair may not be truly good or bad. There may have been some structural variation relative to the supplied reference.

The TAG_ID of tags that match on the reverse strand are appended with _RC. The color-space data are preprocessed and therefore, the first base is the last base of the primer. Distance is the insert size between the paired tags. Bead errors is the sum of the errors in F3 and R3.

SOLiD™ Analysis Tools Pipeline 2

In This Chapter

This chapter covers:

SOLiD™ Analysis Tools (SAT) Pipeline Design

The SAT pipeline analysis system is a framework for organizing secondary and tertiary analysis pipelines into a single front end for convenient command-line access and automation. The SAT pipeline is used in the SOLiD™ analyzer computer system to automate secondary analysis and report generation after a run has been completed on the instrument.

While the SAT pipeline framework is written in Python, it is possible to use any analysis code which presents a command-line interface. The current pipelines call analysis modules that are written in C++, Perl, and Python, as well as common UNIX utilities and shell commands.

In developing the SAT pipeline, the following design attributes were considered important:

- **Modular** – SAT pipeline is hyper-modular in that not only can the individual pipelines be considered loosely coupled modules of code, but also each pipeline consists of loosely coupled command-line tools and scripts.
- **Extensible** – Addition of new analysis pipelines is relatively straightforward.
- **Testable** – The pipelines are amenable to unit tests and stand-alone integration tests.
- **Reentrant** – It is possible to reenter the end-to-end pipeline starting with any pipeline in the analysis flow.
- **Unified error handling and logging** – There is a unified mechanism for logging all of the events encountered by the pipelines. Errors are propagated to the caller through various mechanisms.

Use of SAT Pipeline

The SAT pipeline system supports two different modes of use: interactive analysis from the command line and automated analysis through the auto-analysis daemon system. Both modes are accessed through a command-line interface.

A simple, direct way to run the SAT pipeline from the command line is:

```
moap.py --logfile=moap.log moap_analysis.ini &> moap.err
```

Execution of this command reads parameters from a file named *moap_analysis.ini* and outputs a process log to a file named *moap.log*. Fatal error messages (as well as miscellaneous output from the analysis modules) are captured in a file named *moap.err*. For an example of a moap_analysis.ini file, see Appendix A, "Examples of moap_analysis.ini Files."

When this command is executed, the SAT pipeline process looks for a PBS server for submitting the analysis job. If no server can be found, the analysis is run on the current machine (this mode can be forced by using the `--block` command-line option).

Command Line Options

Table 2-1 Command line options

Parameter	Optional (yes or no)	Description
--appendLog	yes	Specifies that the process log file (--logfile) should be appended instead of overwritten.
--block	yes	If specified, then run the SAT pipeline process under the current shell on this computer.
--dir=...	yes	Specifies a working directory for generating temporary files. SAT pipeline will cd to this directory before starting analysis.
--logfile=...	yes	Specifies a file to contain the output of running the SAT pipeline processes and various useful logging output.
--help	yes	Display a short summary of the available command-line options and exit.
--version	yes	Display the version stamp of the analysis pipeline and exit.
--ini=...	yes	This is an alternate option for specifying a parameter file to through the command-line. If this parameter is not specified, SAT pipeline assumes the first non-option argument is a parameter file.
--queue=...	yes	Specifies a specific PBS queue for submitting the analysis job.

Parameter File

The parameter file specified to `moap.py` contains parameter choices which are specific to the requested analysis. The format of the file is a simple “key-value” properties file with lines of the form "*parameterName=value*" and comments delimited by lines that start with "#".

The following excerpt shows a partial example of such a file:

```
# Excerpt from a moap.ini file
import../../moap.ini
# run name for reports
run.base = SHIRAZ20070406_1
# version of analysis
version = V1
# not a pipeline-required parameter
match.results.dir =
/data/images/results/${run.base}/${version}/${spot}/s_analysis_00
01/s_matching
# The location of p_rawseq for the census and quality-values
pipelines
rawseq.dir = /data/images/results/${run.base}/${version}/rawseq
# name of this run for reports and directory names
run.name = ${run.base}_${spot:S1}_${version}
# root of analysis results (the pipeline doesn't read this)
base.dir = ..
# Whether or not this is mate-paired data
is.mates = 0
```

This example illustrates a mechanism that allows temporary variables to be assigned and substituted when assigning the values of parameters. Substitution allows the use of template files and simplifies the number of parameters that need to be changed for any particular analysis. A variable substitutes by referencing the variable within `${...}`. If a variable hasn't been assigned, a default value can be assigned by using the following notation `${variable:default_value}` (similar to bash shell).

Parameter assignments can be imported from other files using the **import** command. The syntax of this command is a single line with the text **import** [`iniFileName`] where [`iniFileName`] is the location of a parameters file for import.

The SAT pipeline first starts it searches for a file named `moaprc` in `${CORONAROOT}/etc/`. If this file exists, it is imported before any other parameters are assigned. If `~/.moaprc` exists, it will also be imported after the system-wide configuration file has been read.

Parameter assignment follows the *last-assignment-wins* model to allow the specific parameters applied in the command-line .ini file to override system defaults.

Installation

The SAT pipeline analysis system is pre-configured on the SOLiD™ analyzer computer system.

To allow a new user to access the SAT pipeline, either add the line:

`/etc/profile.d/corona.sh` to your `.bashrc`

or add the following lines to `.bash_profile`:

```
CORONAROOT=/share/apps/corona PATH=$PATH:${CORONAROOT}/bin
PYTHONPATH=${CORONAROOT}/lib/python${PYTHONPATH:+:$PYTHONPA
TH}
PERL5LIB=${CORONAROOT}/lib/python${PERL5LIB:+:$PERL5LIB}
```

where `CORONAROOT` is the root directory of the software installation (currently `/share/apps/corona`). Using this second method of configuration, the SAT pipeline can also be run outside of an instrument environment.

The SAT pipeline framework has only been run using Python version 2.3. Newer versions of Python are not likely to have any problems, but Applied Biosystems has not tested or confirmed compatibility.

Limitations

Not every analysis module in the SAT pipeline has been tested at various extremes of whole-genome resequencing (either large, that is, mammalian) genomes or large (> 3Gbp raw) data sets. The following caveats are extended around the current implementation of the SAT pipeline:

- **Large genomes** – Genome sizes larger than 100Mbp have not been examined using the entire set of SAT. The core matching algorithms can handle large mammalian genomes, and the Matching pipeline supports large genomes. It is unknown how the system will perform for the remaining pipelines, but it can be anticipated that most of the pipelines, except the common sequence pipeline, will break because of issues around implementation (Python) and available memory.
- **Large data sets** – Most of the pipeline processing is O(N) with respect to the number of reads (N), but arbitrarily large read sets can be handled, albeit with a concomitant slow-down in processing speed. A chief limitation in SAT is that the current implementation only works well on a single compute node--with some optimizations to take advantage of multi-core architectures. This limitation prevents the instrument cluster from fully taking advantage of SAT and limits the inherent scalability of analysis. In manual analysis, this limitation could be lifted by partitioning the read sets and calling smaller analysis jobs. This missing ingredient today is aggregation logic, which brings together the results of many smaller chunked analyses.

  Note: In the future, the SAT pipeline will be rewritten to internally take advantage of distributed processing without workarounds.

File Formats

GFF

For a description of the GFF file format, see **“Overview of .gff v1 Files” on page 1-8**.

Tab-delimited Text

Many of the raw analysis output files generated by the pipeline use a tab-delimited text format for representing tabular data. This format choice was made for its simplicity and simple integration with common analysis tools such as Matlab, R, gnuplot, and Excel. The GFF format is also tab-delimited text, but it is treated separately because it matches a well-defined specification, unlike the format for the various raw analysis files described here.

The particular tab-delimited format used by most of the analysis modules includes two non-standard extensions: header comments and column names. Here is an excerpt from a file containing these extensions:

```
        # Generated by countErrorsByPrefix.py -p
        0000100111001110011100111 -k 5 -d -5
        /share/reference/genomes/S_suis.dna F3_alignment.gff
        # 03/04/07  21:57:08
##ref    context   num errors        num occurrences    num frequency
AAAAA    117       563               0.207815
CAAAA    143       1779              0.080382
GAAAA    129       629               0.205087
TAAAA    75        667               0.112444
```

The first two lines are arbitrary comments delimited by #. The third line is a special line which provides column names for the columns. This line should start with ##. Any line starting with # should be treated as a comment line for the purposes of reading the data into other programs. The provision for column names allows the potential insertion and rearrangement of columns without breaking downstream code.

Note: File formats are still being assessed, and they will change with time (potentially moving to binary or XML format). All formats will be open, and as more standards evolve for ultra-high-throughput sequencing, they will be accommodated. Some of the data currently represented in this format will be moved to XML in the future. However, it may not make sense to move all data into an XML format, given the inefficiency of XML for large tabular data and the difficulty of applying standard UNIX command-line tools for simple data manipulation (unlike tab-delimited text or GFF).

Properties File

The Properties file format is a standard text file representation of key-value pairs. For more details about the format, see http://en.wikipedia.org/wiki/.properties.

Pipelines

The SAT pipeline is still evolving; some of the pipelines are experimental and provided as-is. Not all pipelines will be maintained in future releases and new pipelines may be added based on customer input.

Global Parameters

Parameters

Table 2-2 Global parameters

Parameter	Optional (yes or no)	Default Value	Description
run.name	no*	unknown	The name of this sequencing run.
reference.name	no*	N/A	The file name of the FASTA reference sequence. The standard configuration is set up to look for this file in a canonical location (reference.genomes.dir). Any arbitrary location.
primer.set	yes	F3	The name of the sequencing primer set (i.e., F3, R3, etc...) for this analysis.
version	yes	V1	The version label to use for this analysis.
sample.name	no*	S1	The name of the analyzed sample for this analysis.
read.length	no*	25	The nominal read length (number of cycles) for this analysis.
is.mates	yes	0	Whether or not this data should be analyzed as mate-paired data.
mismatch.level	yes	1	The maximum number of single-color mismatches to the reference when aligning to the reference sequence.
insert.start	yes	1700	The minimum insert size (in bp) to use when defining a “good” mate pair.

Table 2-2 Global parameters *(continued)*

Parameter	Optional (yes or no)	Default Value	Description
insert.end	yes	3200	The maximum insert size (in bp) to use when defining a “good” mate pair.
reference	no*	N/A	The full location of the FASTA-formatted reference sequence.
reference.de	yes	${reference.de .dir}/de ${reference .name}	The full location of a double-encoded translation of the reference sequence.
reference.analysis.dir	yes	./${reference .genomes .dir} /analysis	The location of a directory for keeping static one-time-only analyses related to a particular reference sequence.
f3.primer.base	yes	T	For the F3 tag, a sequencing run, this is the base underneath the F3 primer.
r3.primer .base	yes	G	For the R3 tag of a mate-pair run, this is the base underneath the R3 primer
reports.deploy	yes	0	Whether or not to deploy (copy) HTML reports to a single locally accessible directory.
reports.deploy.dir	yes	N/A	The location of a directory to deploy all HTML reports.

Matching Parameters

Parameters

Table 2-3 Matching parameters

Parameter	Optional (yes or no)	Default Value	Description
matching.run	yes	0	Whether or not to run the matching pipeline.
matching.tagfiles or *matching.tagfiles.dir*	no	N/A	One of these two parameters must be specified: – *Matching.tagfiles* is a comma-separated list of color space FASTA files to be matched against the genome. – *Matching.tagfiles.dir* is a directory containing color space FASTA files (*.*csfasta* files) to be matched against the genome.
matching.output.dir	yes	.	The location to place output files from this pipeline.
matching.repeat.count	yes	1000	Number of matches reported prior to cutting off search during the matching process.

The following global parameters are also used if available:

- reference
- read.length
- mismatch.level
- mask.positions

Outputs

The output files are placed in the directory location specified by *matching.output.dir*. This directory is created if it does not exist. .

- **`*F3*csfasta.ma.25.3`** – This is an example name for the matched FASTA file. In this case, the matched FASTA file would be for the F3 tag, matching 25 bases against the reference, tolerating three mismatch errors.

Single-Tag Parameters

Parameters

Table 2-4 Single-tag parameters

Parameters	Optional (yes or no)	Default Value	Description
single.tag.run	yes	0	
single.tag.tag	no*	F3	The name of the primer set (F3 or R3) used for sequencing this tag.
single.tag.tagfile or *single.tag.tagfile.dir*	no	N/A	One of these two parameters must be specified: *single.tag.tagfile* = a single *.cdfasta.ma.** file *single.tag.tagfile.dir* = the location of a directory that contains a single *.csfasta.ma.** file.
single.tag.reports	yes	0	Whether or not to generate HTML analysis reports.
single.tag.reports .dir	yes	html Report s	The location for outputting HTML analysis reports.
single.tag.errors.dir	yes	.	The location for outputting basecalling error statistics.
single.tag.coverage.dir	yes	.	The location for outputting depth of coverage statistics.
single.tag.panels.dir	yes	.	The location for outputting per-panel matching statistics.
single.tag.correlation.run	yes	0	Whether or not to run autocorrelation analysis.
single.tag.correlation.dir	yes	.	The location for outputting autocorrelation data.
single.tag.tagbias.dir	yes	.	The location for outputting tag-signature bias analysis.
single.tag.tetrad.dir	yes	.	The location for outputting tetrad analysis data.

The following global parameters are also used if available:

- reference
- reference.de
- read.length
- run.name
- sample.name
- f3.primer.base
- r3.primer.base
- reference.analysis.dir

- reports.deploy
- reports.deploy.dir
- basecall.dir

Outputs

The output files are placed in the directory locations specified by the output directory parameters listed above. If the directory locations do not exist, the system creates them.

In single.tag.reports.dir:

- **`*.html / *.png`** – These are the HTML and backing images which comprise the reports for single-tag analysis.

In *single.tag.errors.dir*:

- **`F3_positionErrors.txt`** – Tab-delimited text, one per primer set. Counts errors versus the reference as a function of the position in the read and the type of color substitution which caused the error.

In *single.tag.coverage.dir*:

- **`fwd_rev_coverage.txt`** – Tab-delimited text. Each line in this file corresponds to a base in the reference sequence. The first column is the number of reads covering this base on the forward strand (5´→ 3´). The second column is the number of reads covering this base on the reverse strand (3´→ 5´).
- **`coverageHist.txt`** – Tab-delimited text. Histogram data for the coverage across the reference. The first column is the level of coverage. The second column is the number of reference bases covered with this depth of coverage.
- **`condensedCoverage.txt`** – Tab-delimited text. The same data as fwd_rev_coverage.txt but adaptively binned to fit the reference into 10,000 evenly spaced bins. The coverage in each bin is determined by averaging across the bin.

In *single.tag.correlation.dir*:

- **`F3_autoCorrelation.txt`** – Tab-delimited text. This is the normalized auto-correlation (Pearson's *r* coefficient) of the color space calls against each other across all of the unfiltered reads. The first two columns specify the two base positions which are compared, the last column is the *r* coefficient. Examining the auto-correlation spectrum can help inform potential chemistry-related issues such as inefficient cleavage, primer contamination, mixed primers, and upstream library prep problems.

In *single.tag.panels.dir*:

- **`F3_panelStats.txt`** – Tab-delimited text. This file contains matching summary statistics for every panel that contained at least one read of usable sequence. The *panel* column is the panel number. The *nomatch* column is the number of beads which didn't match the genome. The *0 MM* column is the number of beads that matched the genome with zero mismatches. There are *MM* counts for every level of mismatch up to the maximum level of mismatch allowed in this analysis (mismatch.level). The *sum of matching* column is the number of beads which matched the genome at any mismatch level. The *fraction matching* is the *sum of matching* divided by the total number of beads (*nomatch + fraction matching*).

In *single.tag.tagbias.dir*:

- **F3_tagSignatureBias.txt** – Tab-delimited text. This is an analysis that examines the base content of the matched reads, looking for position-specific base imbalance. Such imbalance might be indicative of base-specific biases in various facets of the library preparation. The first column is the position of the read on the original reference sequence, relative to the matched position. The next four columns are the relative base composition at this position, normalized by the total occurrence of that particular base across all of the observed reads. Uniform base composition would correspond to a value of 1.0 for each of the four bases.

Basechange Parameters

IMPORTANT! Other pipelines currently have dependencies on this pipeline. Turning it off will cause certain single-tag analyses to fail and may subsequently cause the variation pipeline to fail .

Parameters

Table 2-5 Basechange parameters

Parameter	Optional (yes or no)	Default Value	Description
basechange.run	yes	0	Whether or not to run the basechange pipeline
basechange.tagfile or *basechange.output.dir*	yes	0	Whether or not to run the basechange pipeline
basechange.output.dir	yes	0	The location for outputting analysis files

Note: The global parameter ‘reference’ is also used if available.

Outputs

The output files are placed in the basechange.output.dir. This directory is created if it does not exist.

. The output includes `xxx.bc.txt` and `statistics.txt`.

In *basechange.output.dir*:

`xxxx.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 whether the tag was placed uniquely or randomly. Because this is processed color-space data in the SOLiD sequence, the last base of the primer and the first color space call are replaced with the first base of the 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 ERRORS BASE_CHANGES
PLACEMENT
```

```
1_92_1875_R3 top C0330101130332001221
C0330101130333001221 1571875 1 13_23 unique
1_94_1682_R3 reverse 2133312212133221213T
2113312212133221213T 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

A text file containing a summary of the matching statistics. Beads found is the number of beads initially identified. Many of these are not real 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 are unlikely to match the genome. Likewise, current single-round enrichment provides 60-80% amplicon positive beads. The amplicon empty beads are usually still found by the bead finder because they have a slight background signal.

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).

Paired-Tag Parameters

Parameters

Table 2-6 Paired-tag parameters

Parameter	Optional (yes or no)	Default Value	Description
mate.pairs.run	yes	0	Whether or not to run the paired-tag pipeline
mate.pairs.f3file or *mate.pairs.tagfile.dirs*	no	0	One of these two parameters must be specified
mate.pairs.correlation.run	yes	0	Whether or not to run autocorrelation analysis
mate.pairs.reports	yes	0	Whether or not to generate HTML analysis reports
mate.pairs.reports.dir	yes	0	The location for outputting HTML reports
mate.pairs.coverage.dir	yes	0	The location for outputting depth of coverage data
mate.pairs.errors.dir	yes	0	The location for outputting basecalling error data
mate.pairs.distances.dir	yes	0	The location for outputting mate-pair distance analysis
mate.pairs.panels.dir	yes	0	The location for outputting per-panel matching statistics
mate.pairs.correlation.dir	yes	0	The location for outputting autocorrelation data
mate.pairs.tagbias.dir	yes	0	The location for outputting tag-signature bias analysis
mate.pairs.tetrad.dir	yes	0	The location for outputting tetrad analysis data
mate.pairs.use.pairing	yes	0	Whether or not to use mate-pair rescue (the *pairing* binary)
mate.pairs.rescue.level	yes	0	Total number of mismatches across the F3 and R3 tags to allow when rescuing mate-pairs

The following global parameters are also used if available:

- reference
- reference.de
- read.length
- run.name
- sample.name
- f3.primer.base
- r3.primer.base
- reference.analysis.dir
- insert.start
- insert.end
- reports.deploy
- reports.deploy.dir
- basecall.dir

Outputs

The outputs of the mate-pairs pipeline are very similar to the outputs of the single-tag pipeline, with the addition of several analysis outputs that make sense only in the context of mate-pair resequencing. These output files are placed in the directory locations specified by the output directory parameters listed above. These locations are created if they do not exist.

The outputs in the `mate.pairs.error.dir, mate.pairs.correlation.dir, mate.pairs.panels.dir, mate.pairs.coverage.dir, mate.pairs.tetrad.dir, mate.pairs.reports.dir, and mate.pairs.tagbias.dir` directories are analogous to the outputs in the single-tag pipeline described above, with the extension that there exists a separate analysis file for each tag (primer set).

In addition, the `mate.pairs.distances.dir` directory contains:

- **`mateDistances.txt`** – Tab-delimited text. This file contains a single column of insert sizes in base pairs, one for each mate pair placed on the reference genome.

The `mate.pairs.reports.dir` directory contains additional files specific to mate-pair analysis, as well as the assortment of HTML and .png files associated with single-tag and mate-pair analysis:

- **`F3_R3_mates.report`** – Tab-delimited text. This file contains matching statistics on how many F3 reads match the genome versus R3 reads at various levels of mismatch tolerance. Missing tags and reads that do not match are also reported. For a mate-pair experiment, this report provides a comprehensive summary of what happened in the matching of each tag to the genome. The census report summarizes this report in an HTML table.

- **`annotateBadMates.report`** – Tab-delimited text. This file contains statistics counting how well the aligned mate pairs fell within mate-pair constraints (expected insert size, expected orientation with respect to each tag and the genome). Each mate pair is counted as a member of one of nine categories, depending on which constraints are satisfied. The census report summarizes this report in an HTML table.

Error-Prone Parameters

Parameters

Table 2-7 Error-prone parameters

Parameter	Optional (yes or no)	Default Value	Description
error.prone.run	yes	0	
error.output.dir	yes	0	
error.prone.reports	yes	0	
error.prone.reports.dir	yes	N/A	

The following global parameters are also used if available:

- reference
- reference.de
- read.length
- run.name
- sample.name
- f3.primer.base
- r3.primer.base
- reference.analysis.dir
- is.mates
- insert.start
- insert.end
- reports.deploy
- reports.deploy.dir

Outputs

Output files are placed in the *error.prone.output.dir* directory. The directory is not created if it does not exist.

Underneath this directory, sub-directories are created to contain the results of each sequence context combination that is examined. These directories are named as *{positions}_{probeType}_{tagName}* where *positions* is one of the following: *01*, *123456*, *inosine*. The *probeType* is either *12encoded* or *45encoded* and the *tagName* is either F3 or R3. The four possible *positions* combinations correspond to examining bases 1,2,3,4,5 under the probe, bases -1,1 under the probe, bases 1,2,3,4,5,6 under the probe, and bases 6,7,8 under the probe.

The set of positions which are examined is historical and can be cleaned up and made more efficient.

Under each of these directories are text files summarizing the frequency of basecalling error when presented with various sequence contexts (in base space on the reference genome):

- **kmerContext_err_vs_control.txt** – These files consist of tab-delimited text. The first column is the *k*-mer context on the reference genome. The next column is the number of errors that were observed when a probe sequenced this context. The third column is the number of color calls that were made when a probe sequenced this context. The fourth column is the frequency of error when this context was sequenced. The value of *k* is different depending on the number of bases in the particular context.

Coverage Bias Parameters

Parameters

Table 2-8 Coverage bias parameters

Parameter	Optional (yes or no)	Default Value	Description
coverage.bias.run	yes	0	
coverage.bias.window	yes	50	
coverage.bias.output.dir	yes	qc coverage	
coverage.bias.reports	yes	0	
coverage.bias.reports.dir	yes	qc reports	

The following global parameters are also used if available:

- reference
- read.length
- run.name
- sample.name
- reference.analysis.dir
- is.mates
- reports.deploy
- reports.deploy.dir

In addition, one of the two variables – *single.tag.coverage.dir* or *mate.pairs.coverage.dir* – must be set (see above).

Outputs

Output files are placed in the *coverage.bias.output.dir* directory. This directory is created if it does not exist.

- **`binned_coverage_masked.txt`** – Tab-delimited text. This file is similar to the *fwd_rev_coverage.txt* file, but the actual coverage has been replaced by a coverage category, determined by adaptively binning the observed coverage. The adaptive binning attempts to construct ten coverage bins, each containing a similar number of reference bases. The no-coverage bin is treated as a special case, so eleven bins are determined in the general case. The header of this file contains details on the bin choices. Coverage is only examined at locations in the genome that are sequenceable (uniquely placeable) for the given read length.
- **`[ACGT]_frequency_vs_coverage_masked.txt`** – Tab-delimited text. This file examines the forward and reverse coverage across the genome, using the coverage levels determined in the *binned_coverage_masked.txt* file, and reports a histogram of the local base composition for all of the bases covered by that coverage level. There are four files, one for each of the bases.

  Note: The sequenceability determination used in creating the coverage mask does not take into account the possibility of mate-pair coverage. Mate-pairs allow reference sequences to be significantly covered, but it is difficult to construct an automatic (and deterministic) judgment of mate-pair sequenceability due to the random variables involved (mean insert size and standard deviation).

Census Parameters

Parameters

Table 2-9 Census parameters

Parameter	Optional (yes or no)	Default Value	Description
census.run	yes	0	Whether or not to run the census pipeline
census.count.good	yes	0	
census.primer.match	yes	0	
census.primer.match.dir	yes	0	
census.primer.match.min.qv	yes	0	
census.primer.match.max.qv	yes	0	
census.primer.match.reference	yes	0	
census.reports	yes	0	
census.reports.dir	yes	0	

The following global parameters are also used if available:

- reference
- read.length
- run.name
- sample.name
- reference.analysis.dir
- is.mates
- reports.deploy
- reports.deploy.dir

In addition, one of the tagfile variables from the Single-tag or Paired-tag pipelines must be set, depending on whether the data is mate-paired or not.

Outputs

Output files are placed in the *census.primer.match.dir* directory. This directory is created if it does not exist.

- **`F3_highQuality_noMapping.fsta.ma.20.1`** – Matched color space FASTA. This file contains the high-quality reads that did not match the reference sequence but that matched the bead sequence instead.
- **`F3_highQualityReads_withCounts.txt`** – Tab-delimited text. This file contains a sorted list of all high-quality sequences, with a count of how many time that sequence was observed.
- **`primer.match.report`** – Tab-delimited text. Contains statistics of how many high-quality reads were found, how many did not match the reference sequence, and how many matched the bead sequence.

Common Sequence Parameters

Parameters

Table 2-10 Common sequence parameters

Parameter	Optional (yes or no)	Default Value	Description
common.sequence.run	yes	0	
common.sequence.tagfile.dir	yes	0	
common.sequence.output.dir	yes	qc commonSequence	
common.sequence.reports	yes	0	
common.sequence.reports.dir	yes	qc reports	

The following global parameters are also used if available:

- single.tag.tag
- read.length
- run.name
- is.mates
- reports.deploy
- reports.deploy.dir

Outputs

Output files are placed in the *common.sequence.output.dir* directory. This directory is created if it does not exist.

- **`10mer_prefix_counts.txt`** – Tab-delimited text. Contains a sorted list of all observed 10-mer prefixes for all reads, with the highest frequency prefixes reported at the top.
- **`5mer_prefix_counts.txt`** – Tab-delimited text. Contains a list of all observed 5-mer prefixes for all reads. This list is sorted by the prefix to facilitate anomaly detection when plotting this data.

Visualization Parameters

Parameters

Table 2-11 Visualization parameters

Parameter	Optional (yes or no)	Default Value	Description
visual.run	yes	0	Whether or not to run the visualization pipeline
visual.region.size	yes	0	
visual.output.dir	yes	0	
visual.reports	yes	0	
visual.reports.dir	yes	0	

The following global parameters are also used if available:

- run.name
- is.mates
- insert.start
- insert.end
- single.tag.tag
- reports.deploy
- reports.deploy.dir

Outputs

Output files are placed in the *visual.output.dir* directory. This directory is created if it does not exist.

- **`region_1_10000.gff`** – GFF. This file contains the GFF-formatted reads which map to a particular region in the genome. There is a similarly named file for each region in the genome. The region size is automatically determined (if not set by *visual.region.size*) to produce an optimally sized GFF file for loading into SOLiD™ Alignment Browser (~2MB today).

Variation Parameters

Parameters

Table 2-12 Variation parameters

Parameter	Optional (yes or no)	Default Value	Description
variation.run	yes	0	Whether or not to run the variation pipeline
variation.output.dir	yes	.	
variation.readlength	yes		
variation.splitlength	yes		

The following global parameters are also used if available:

- is.mates
- primer.set
- reference
- reference.dibase

Outputs

Outputs include:

```
bp_consensus_confirmed_sequence_with_Ns. fasta
xxx.snps_sorted.txt
```

bp_consensus_confirmed_sequence_with_Ns. fasta

This file contains the consensus sequence generated from alignment to the reference sequence. It is a fasta format sequence with a header and the sequence.

A consensus base is called for any base with >n coverage (default value for n is ?). If a base is not called, an N is recorded for that position. If a heterozygous SNP was called for a position in the SNP-consensus.txt file, it is recorded using the IUPAC code in the appropriate position. If heterozygosity was seen but the SNP probability is <?? then an N is recorded. Homozygote differences from the reference are recorded provided coverage is >n.

xxx.snps_sorted.txt

Column 1 (cov) — The number of reads covering the position with 2 color space calls (i.e., reads only partially covering the base with the first or last color space call of the read aren't counted)

Column 2 (ref) — The reference base at that position

Column 3 (consen) — The consensus base at that position (IUPAC codes are used for heterozygotes)

Columns 4 - 7 —- The reference base

Columns 8 - 10 — The consensus base

`score` = the weighted score assigned to the reference or consensus base based on all the reads that cover it with 2 color space calls (the weighted scores for each of the 16 types of dibase combinations sum to 1)

`conf` = the average confidence of each read that was used to generate the weighted score of the reference or consensus base

`F3` = number of F3 reads that agree with the reference / number of unique F3 molecules (start points) that agree with the reference

`R3` = number of R3 reads that agree with the reference / number of unique R3 molecules (start points) that agree with the reference

```
# cov  ref  consen   score    confi    F3    R3    score    conf     F3       R3       coord
1303   G    T        0.0007   0.7999   1/1   -/-   0.9675   0.9173   1259/68  -/-      223051
1884   A    G        0.0006   0.9618   1/1   -/-   0.9069   0.8698   1708/67  -/-      196597
988    T    C        0.0070   0.8756   7/5   -/-   0.9719   0.8832   959/67   -/-      228320
1051   T    C        0.0119   0.9105   12/5  -/-   0.935    0.8762   1023/66  -/-      437995
1183   G    C        0.0009   0.9230   1/1   -/-   0.9614   0.9147   1136/65  -/-      404144
1028   C    G        0.0010   0.9622   1/1   -/-   0.9602   0.8976   987/65   -/-      95856
```

Analysis Stages and File Layout 3

In This Chapter

This chapter covers:

Analysis Stages

Each SOLiD analysis comprises five job stages. The stages can be referred to by number or by name.

- Per Analysis:
  - (30) Post Focal Map. It finds bead positions.
  - (31) Post Scan Slide (a.k.a. Color-calling). There is 1 job (31) per ligation cycle (for example, F3_P1_03).
  - (32) Post Primer Set Primary (a.k.a. Filter Fasta). There is 1 job (32) per primer set (for example, F3 or R3). In this stage, information of all panels is aggregated into 1 fasta file.
  - (33) Post Primer Set Secondary (a.k.a. Moap-unpaired). There is 1 job (33) per primer set (for example, F3 or R3).
  - (34) Post Run Secondary (a.k.a. Run Analysis).

  There is a newly created analysis (also known as a *JobWorkflow*) for each sample every time it is submitted for analysis or reanalysis.
- Notation:

  $SampleResultFolder$ =
  /data/results/[instrumentName]/[runName]/[sampleName]
  /results
- Folder name:

  primary.[17digit timestamp] = “primary.” + the 17 digit date created timestamp of job (32)

  intermediate.[17digit timestamp] = “intermediate.” + the 17 digit date created timestamp of job (33)

  secondary.[17digit timestamp] = “secondary.” + the 17 digit date created timestamp of job (34)

The [17 digit timestamp] can be found in the “date_created” column of “analysis_job” table if you know how to get to the right flowcell_run/sample/job_workflow/analysis_job by their names and relationships.

Here is a very useful query to get all the analysis jobs for a run by name:

```
select sam.name, wf.name, aj.*
from flowcell_run run, sample_loading sl, sample sam, job_workflow
wf, analysis_job_workflow_link link, analysis_job aj
where
run.sample_loading_id = sl.sample_loading_id and
sl.sample_loading_id = sam.sample_loading_id and
sam.sample_id =  wf.sample_id and
wf.job_workflow_id = link.job_workflow_id and
link.analysis_job_id = aj.analysis_job_id and
run.name = 'DAEMON20070625_1' order by date_created
```

Results Folder for a Clean Run

Single-Tag

For each sample, the following folders are created:

```
$SampleResultFolder$/basecalls
$SampleResultFolder$/basecall_summary
$SampleResultFolder$/cycleplots
$SampleResultFolder$/traffic_lights
$SampleResultFolder$/hadesCycleplots
$SampleResultFolder$/primary.[17digit timestamp]
$SampleResultFolder$/intermediate.[17digit timestamp]
$SampleResultFolder$/secondary.[17digit timestamp]
```

Mate-Pair

For each sample, the following folders are created:

```
$SampleResultFolder$/basecalls
$SampleResultFolder$/basecall_summary
$SampleResultFolder$/cycleplots
$SampleResultFolder$/traffic_lights
$SampleResultFolder$/hadesCycleplots
$SampleResultFolder$/primary.[17digit timestamp]
$SampleResultFolder$/ primary.[17digit timestamp]
$SampleResultFolder$/intermediate.[17digit timestamp]
$SampleResultFolder$/intermediate.[17digit timestamp]
$SampleResultFolder$/secondary.[17digit timestamp]
```

Note: There are 2 primary.[17 digit timestamp] folders and 2 intermediate.[17 digit timestamp] folders in this case (since there are two primer sets).

SETS Secondary Re-Analysis

The re-analysis starts at job stage (33) – **`Moap-unpaired`**.

Single-Tag

For each submitted sample, the following folders are added:

```
$SampleResultFolder$/intermediate.[17digit timestamp]
$SampleResultFolder$/secondary.[17digit timestamp]
```

Mate-Pair

For each submitted sample, the following folders are added:

```
$SampleResultFolder$/intermediate.[17digit timestamp]
$SampleResultFolder$/intermediate.[17digit timestamp]
$SampleResultFolder$/secondary.[17digit timestamp]
```

SETS Primary Re-Analysis

The re-analysis starts at job stage (32)–**`Filter fasta`**.

Single-Tag

For each submitted sample, the following folders are added:

```
$SampleResultFolder$/primary.[17digit timestamp]
$SampleResultFolder$/intermediate.[17digit timestamp]
$SampleResultFolder$/secondary.[17digit timestamp]
```

Mate-Pair

For each submitted sample, the following folders are added:

```
$SampleResultFolder$/primary.[17digit timestamp]
$SampleResultFolder$/primary.[17digit timestamp]
$SampleResultFolder$/intermediate.[17digit timestamp]
$SampleResultFolder$/intermediate.[17digit timestamp]
$SampleResultFolder$/secondary.[17digit timestamp]
```

ICS Repeat Primers (Re-Analysis)

The re-analysis starts at job stage (31) – **`Color-calling`**.

Single-tag or Mate-pair run behave the same way in this case since ICS only allows to repeat primers of 1 primer set each time.

For each submitted sample, the following folders are added:

```
$SampleResultFolder$/primary.[17digit timestamp]
$SampleResultFolder$/intermediate.[17digit timestamp]
$SampleResultFolder$/secondary.[17digit timestamp]
And the following folders are modified: (new files for the new version
_Vx)
$SampleResultFolder$/basecalls
$SampleResultFolder$/basecall_summary
$SampleResultFolder$/cycleplots
$SampleResultFolder$/traffic_lights
$SampleResultFolder$/hadesCycleplots
[One HISTO file for each primerSet]
```

Examples of moap_analysis.ini Files A

In This Appendix

This appendix covers:

Note: These are sample pipeline analysis settings only. Modify these examples as appropriate.

Single Tag Run

Specify which pipeline should run. This example is for a single tag run.

```
common.sequence.run=1
census.primer.match=0
census.run=1
coverage.bias.run=1
visual.run=1
error.prone.run=1
single.tag.run=1
basechange.run=1
matching.run=1
mismatch.level=3
reference.name=DH10B_WithDup_FinalEdit_validated.fasta
analysis.settings.id=3
```

Analysis Job Parameters

If the slide is divided into several samples, the panel range for each sample should be specified.

```
instrument.name=R1a005
panels=871-1305
primer.set=F3
read.length=25
run.name=R1a005_20070622_2
sample.name=mySample
spots=3
```

Run Directories

The data and colorspace .fasta files are storedin run directories.

```
analysis.sample.dir=/data/results/${instrument.name}/${run.name}/${sa
mple.name}
analysis.run.dir=/data/results/${instrument.name}/${run.name}/${sampl
e.name}
analysis.results.dir=${analysis.run.dir}/results
```

Specify where the colorspace **`reads`** file is located. The file should end with .csfasta:

```
reads.result.dir=${analysis.run.dir}/results/primary.1/reads
read.dir=${analysis.run.dir}/results/primary.1/reads
```

The color call primary results are usually located in:

```
summary.dir=${analysis.run.dir}/results/basecall_summary
basecallfolder=${analysis.run.dir}/results/basecalls
cycleplots.dir=${analysis.run.dir}/results/cycleplots
```

Where to put the SAT pipeline secondary analysis results:

```
secondary.result.dir=${analysis.run.dir}/results/secondary01
```

Specify the location of the job folder for the run, including log files and .tmp files for the jobs:

```
analysis.work.dir=${analysis.run.dir}/jobs/secondary01.manual
```

Mate-Pair Run

Specify which pipeline should run. This example is for a mate-pair run.

```
census.run=1
insert.end=4200
insert.start=1800
mate.pairs.run=1
is.mates=1
coverage.bias.run=1
visual.run=1
error.prone.run=1
single.tag.run=0
basechange.run=0
matching.run=1
mismatch.level=3
reference.name=DH10B_WithDup_FinalEdit_validated.fasta
```

Analysis Job Parameters

```
instrument.name=R1a005
panels=1-435
primer.set=F3,R3
read.length=25
run.name=R1a00520070625_1
sample.name=DH10B_MP_Ctrl
spots=1
```

Run Directories

There are 2 dirs for F3 and R3 matching results.

```
analysis.run.dir=/data/results/${instrument.name}/${run.name}/${sampl
e.name}
analysis.results.dir=${analysis.run.dir}/results
summary.dir=${analysis.run.dir}/results/basecall_summary
basecallfolder=${analysis.run.dir}/results/basecalls
cycleplots.dir=${analysis.run.dir}/results/cycleplots
secondary.result.dir=${analysis.run.dir}/results/secondary.2007070408
0757718
analysis.sample.dir=${analysis.run.dir}
analysis.work.dir=${analysis.run.dir}/jobs/postRunSecondary.733
postPrimerSetSecondary.result.dir.1=${analysis.run.dir}/results/secon
dary.20070629013837921/s_matching
postPrimerSetSecondary.result.dir.2=${analysis.run.dir}/results/secon
dary.20070704080756734/s_matching
```

Data Management B

In This Appendix

This appendix covers:

Saving Data

This section describes data storage.

Raw image data

The data should be kept until the full completion of primary analysis is verified and primary analysis results are archived. After confirmation that the image files are no longer needed, these image files can be deleted for additional sequencing runs.

Primary analysis result data

The full primary analysis data can be copied onto an external removable USB drive. The data volume precludes long term storage on the SOLiD™ System.

Secondary analysis result data

The data can be copied onto external removable USB drives or through the network onto another computer server.

Approximate Storage Space for Primary Analysis Data

- **1 slide 1 tag** — Assumes that a full slide is used and 5 cycles for each sequencing primer, each slide has 2357 panels, 4 images per cycle, one for each dye. The storage required for primary analysis results (excluding images) is estimated 80-100GB for one quadrant and 360GB for one slide.
- **1 slide 2 tags** — With the same assumption, 720GB of storage space is required.
- **2 slides 1 tag** — With the same assumption, 720 GB of storage space is required.
- **2 slides 2 tags** — With the same assumption, 1.5 TB of storage space is required.

Note: These figures refer to storage of all secondary analysis files, including many intermediate files. If only key processed files (for example, .gff files, consensus files, and SNP files) are retained, space requirments will be in the 10's of GB.

Data Transfer Procedure

Requirements

- External USB hard disk drive (700 GB capacity).
- The drive should be formatted using UNIX systems as ext3 format.

Procedure

Connect USB drive to the USB port on the head node of Linux cluster.

1. If the drive does not mount automatically, check the log message to determine the device name for the drive:

   ```
   tail –f /var/log/messages
   ```

   or

   ```
   dmesg
   ```

2. If the drive shows up as `/dev/sdc`, then mount the drive as follows:

   ```
   mkdir/mnt/usb(to create a mount point)
   ```

   – This can be skipped to mount the drive under

   ```
   /media/usbdisk mount/dev/sdc/mnt/usb
   ```

   (This may need to be logged in as root), or

   – Mount/dev/sdc/media/usbdisk.

   If the drive mounts automatically, it will usually show up as `/media/usbdisk`

3. Copy directory named {run.name} on the Linux cluster to the USB drive using the following command:

   ```
   rsync -aup {run.name} /mnt/usb/
   ```

   or

   `rsync –aup {run.name} /media/usbdisk/` (depending on where the drive was mounted)

   Note: {run.name} = /data/results/{instrument.name}/{run.name}

   IMPORTANT! Perform the copy for each slide separately. If two slides and two tags were run, use two USB drives (700 GB each).

   The transfer rate of is approximately 10-20 MB/sec based on USB2.0 and 750 GB drive. For 300 GB of data, approximately 4 hour is needed to copy the data (150 GB / 120 min).

4. Once copy is complete, verify that the USB drive contains the copied data, then unmount the drive, and disconnect the USB drive.

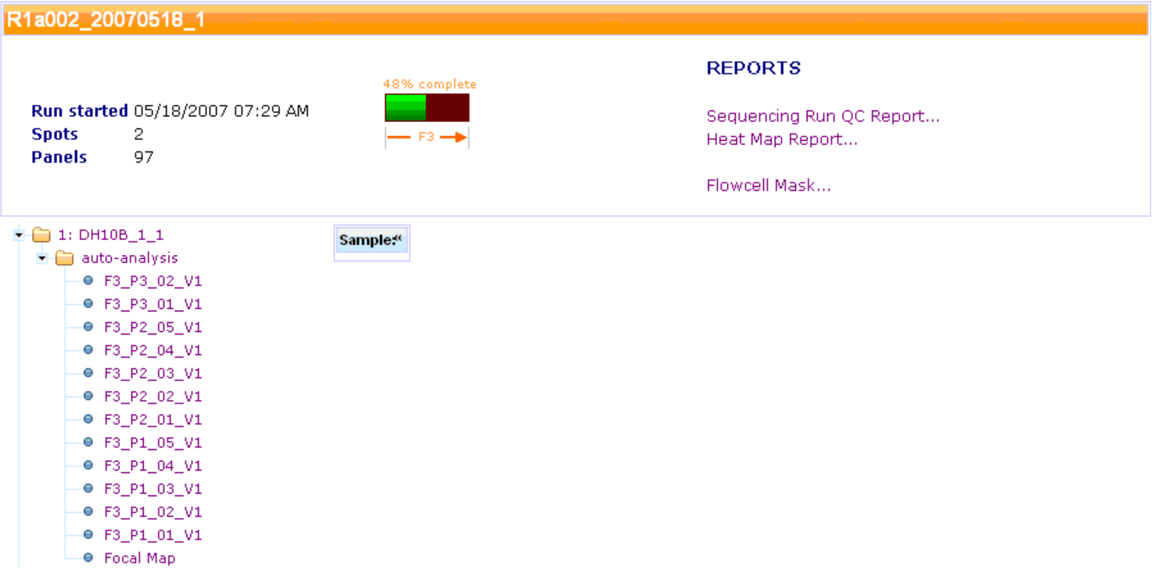
Procedures for Verifying Primary Analysis Status

To verify primary analysis status:

1. From SETS interface, under In-Progress or Most Recent Run or Recent Flowcell Runs, click the run name of interest.

   The numbered items in the data tree on the left side of the page relate to the individual samples on the slide.

2. Click one of the data tree sample numbers on the left of the page to see the Library file name, Spot number, Sample name, and Reports information for that particular sample.



R1a002_20070518_1

Run started 05/18/2007 07:29 AM
Spots 2
Panels 97

REPORTS

Sequencing Run QC Report...
Heat Map Report...

Flowcell Mask...

- 1: DH10B_1_1
  - auto-analysis
    - F3_P3_02_V1
    - F3_P3_01_V1
    - F3_P2_05_V1
    - F3_P2_04_V1
    - F3_P2_03_V1
    - F3_P2_02_V1
    - F3_P2_01_V1
    - F3_P1_05_V1
    - F3_P1_04_V1
    - F3_P1_03_V1
    - F3_P1_02_V1
    - F3_P1_01_V1
    - Focal Map

3. Click the drop-down arrow to the left of the sample folder to view individual analysis cycles for each sample.

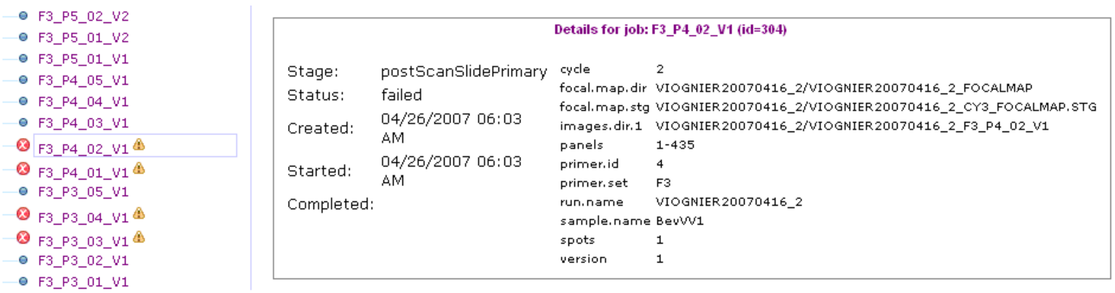
4. Check that all the analysis completed successfully. A blue dot before the analysis name means that the analysis completed successfully.

   Clicking the name of the analysis will show additional information about the analysis.

Troubleshooting Analysis Failure

If there is a triangle to the right of the analysis name and there is a red x before the analysis name, the analysis failed. To see details:

1. Click the name of the failed analysis. The following example shows 4 failed analyses: F3_P4_02_V1, F3_P4_01_V1, F3_P3_04_V1, and F3_P3_03_V1. The right portion of the figure shows the details of analysis F3_P4_02_V1

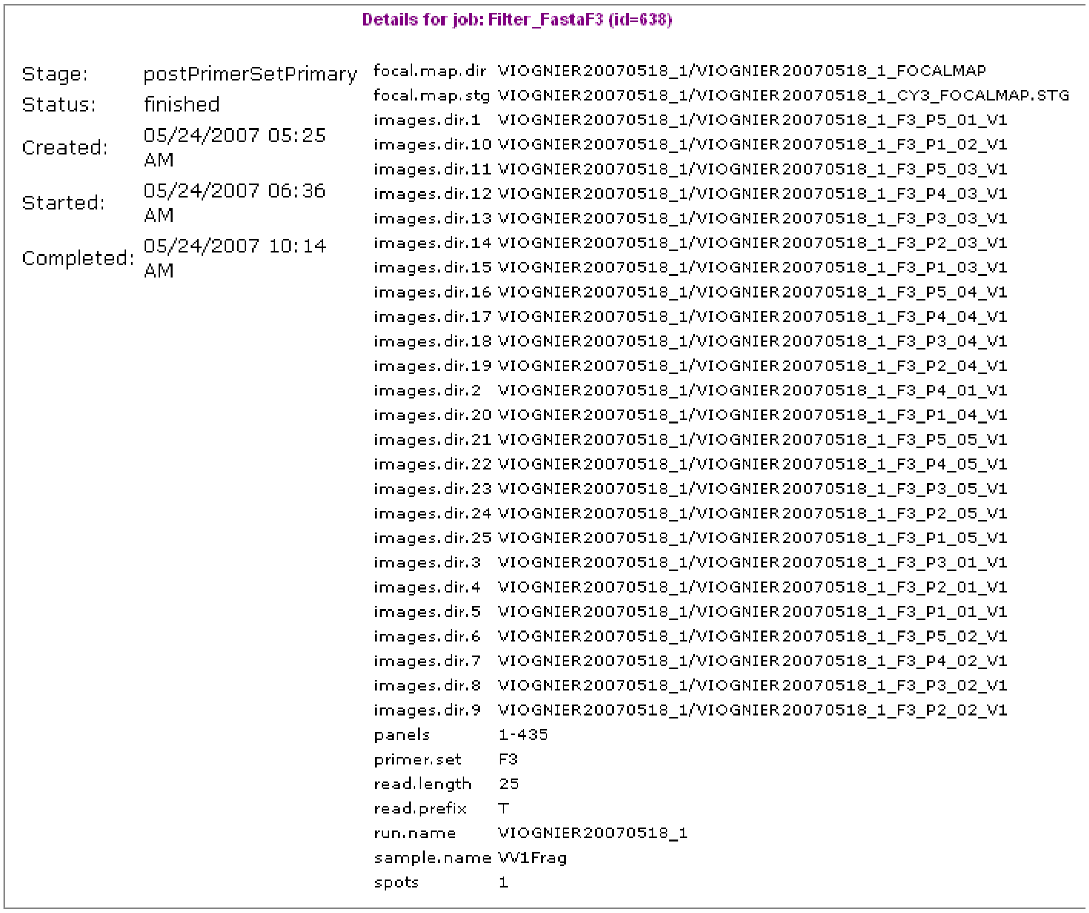


```
F3_P5_02_V2
F3_P5_01_V2
F3_P5_01_V1
F3_P4_05_V1
F3_P4_04_V1
F3_P4_03_V1
F3_P4_02_V1
F3_P4_01_V1
F3_P3_05_V1
F3_P3_04_V1
F3_P3_03_V1
F3_P3_02_V1
F3_P3_01_V1
```

Details for job: F3_P4_02_V1 (id=304)

```
Stage:      postScanSlidePrimary   cycle          2
Status:     failed                 focal.map.dir  VIOGNIER20070416_2/VIOGNIER20070416_2_FOCALMAP
Created:    04/26/2007 06:03       focal.map.stg  VIOGNIER20070416_2/VIOGNIER20070416_2_CY3_FOCALMAP.STG
            AM                     images.dir.1   VIOGNIER20070416_2/VIOGNIER20070416_2_F3_P4_02_V1
                                   panels         1-435
Started:    04/26/2007 06:03       primer.id      4
            AM                     primer.set     F3
Completed:                         run.name       VIOGNIER20070416_2
                                   sample.name    BevVV1
                                   spots          1
                                   version        1
```

2. Check Filter_FastaF3 and Filter_FastaR3 (if paired-end library) to see if analysis was successful and check the details for the analysis and each cycle to make sure they are correct.



Details for job: Filter_FastaF3 (id=638)

```
Stage:      postPrimerSetPrimary   focal.map.dir  VIOGNIER20070518_1/VIOGNIER20070518_1_FOCALMAP
Status:     finished               focal.map.stg  VIOGNIER20070518_1/VIOGNIER20070518_1_CY3_FOCALMAP.STG
Created:    05/24/2007 05:25       images.dir.1   VIOGNIER20070518_1/VIOGNIER20070518_1_F3_P5_01_V1
            AM                     images.dir.10  VIOGNIER20070518_1/VIOGNIER20070518_1_F3_P1_02_V1
Started:    05/24/2007 06:36       images.dir.11  VIOGNIER20070518_1/VIOGNIER20070518_1_F3_P5_03_V1
            AM                     images.dir.12  VIOGNIER20070518_1/VIOGNIER20070518_1_F3_P4_03_V1
Completed:  05/24/2007 10:14       images.dir.13  VIOGNIER20070518_1/VIOGNIER20070518_1_F3_P3_03_V1
            AM                     images.dir.14  VIOGNIER20070518_1/VIOGNIER20070518_1_F3_P2_03_V1
                                   images.dir.15  VIOGNIER20070518_1/VIOGNIER20070518_1_F3_P1_03_V1
                                   images.dir.16  VIOGNIER20070518_1/VIOGNIER20070518_1_F3_P5_04_V1
                                   images.dir.17  VIOGNIER20070518_1/VIOGNIER20070518_1_F3_P4_04_V1
                                   images.dir.18  VIOGNIER20070518_1/VIOGNIER20070518_1_F3_P3_04_V1
                                   images.dir.19  VIOGNIER20070518_1/VIOGNIER20070518_1_F3_P2_04_V1
                                   images.dir.2   VIOGNIER20070518_1/VIOGNIER20070518_1_F3_P4_01_V1
                                   images.dir.20  VIOGNIER20070518_1/VIOGNIER20070518_1_F3_P1_04_V1
                                   images.dir.21  VIOGNIER20070518_1/VIOGNIER20070518_1_F3_P5_05_V1
                                   images.dir.22  VIOGNIER20070518_1/VIOGNIER20070518_1_F3_P4_05_V1
                                   images.dir.23  VIOGNIER20070518_1/VIOGNIER20070518_1_F3_P3_05_V1
                                   images.dir.24  VIOGNIER20070518_1/VIOGNIER20070518_1_F3_P2_05_V1
                                   images.dir.25  VIOGNIER20070518_1/VIOGNIER20070518_1_F3_P1_05_V1
                                   images.dir.3   VIOGNIER20070518_1/VIOGNIER20070518_1_F3_P3_01_V1
                                   images.dir.4   VIOGNIER20070518_1/VIOGNIER20070518_1_F3_P2_01_V1
                                   images.dir.5   VIOGNIER20070518_1/VIOGNIER20070518_1_F3_P1_01_V1
                                   images.dir.6   VIOGNIER20070518_1/VIOGNIER20070518_1_F3_P5_02_V1
                                   images.dir.7   VIOGNIER20070518_1/VIOGNIER20070518_1_F3_P4_02_V1
                                   images.dir.8   VIOGNIER20070518_1/VIOGNIER20070518_1_F3_P3_02_V1
                                   images.dir.9   VIOGNIER20070518_1/VIOGNIER20070518_1_F3_P2_02_V1
                                   panels         1-435
                                   primer.set     F3
                                   read.length    25
                                   read.prefix    T
                                   run.name       VIOGNIER20070518_1
                                   sample.name    VV1Frag
                                   spots          1
```

The above example shows detailed information for one Filter_FastaF3 run. This run uses 5 primers (P1-P5) and for each primer, 5 cycles were used. This analysis used all 5 primers (P1-P5) and for each primer, 5 cycles were included in the analysis indicating no missing images in the analysis.

Sometimes, analysis may be reported to be successful (no red x or warning triangle next to the analysis name), but some of the primer/cycle combinations were not included in the analysis. Even though the analysis is successful, if the analysis is incomplete, the analysis should be treated as a failed analysis.

3. Log on to the instrument, and look in the directory {SampleResultsFolder}/primary.{17 digit timestamp}/reads, there should be a file named {run.name}_F3.stats for F3 tag, or {run.name}_R3.stats for R3 tag.

   This file contains information about the color space reads generation for each panel. It contains the number of reads and whether they have too many errors, too short or duplicates. At the bottom of the file, there should be a line beginning with Totals (### p): which summarizes number of panels found and percentage of error, short or duplicate reads. After this line, there should be a line beginning with Usable (### p) (##%): which summarizes how many panels were usable and percentage of erroneous, short, or duplicate reads among the usable panels.

   Note: If these 2 lines are missing, the primary analysis did not complete.

Procedure for Re-Analyzing Image (Primary Analysis)

Use this procedure to manually re-analyze the image to perform primary analysis if there was failure in the auto-primary analysis pipeline.

1. Make sure all the images are present under:

   ```
   /data/images/{run.name}
   ```

2. Issue the following command:

   ```
   jprimaryanalysis.sh --dir=<parameter file directory> --
   ini=<parameter file name>
   ```

   for example:

   ```
   jprimaryanalysis.sh --
   dir=/data/results/DAEMONAC2/LAST_MP_1245_041008/Sample1/jobs/
   postPrimerSetPrimary.3793 --ini=parameters.ini
   ```

Software Warranty Information C

In This Appendix

This appendix covers:

Computer Configuration

Applied Biosystems supplies or recommends certain configurations of computer hardware, software, and peripherals for use with its instrumentation.

Applied Biosystems reserves the right to decline support for or impose extra charges for supporting nonstandard computer configurations or components that have not been supplied or recommended by Applied Biosystems. Applied Biosystems also reserves the right to require that computer hardware and software be restored to the standard configuration prior to providing service or technical support. For systems that have built-in computers or processing units, installing unauthorized hardware or software may void the Warranty or Service Plan.

Limited Product Warranty

Notice to User

PLEASE READ THIS DOCUMENT CAREFULLY. THIS IS THE CONTRACT BETWEEN YOU AND APPLERA CORPORATION, ACTING THROUGH ITS APPLIED BIOSYSTEMS GROUP, REGARDING THE OPERATING SOFTWARE FOR YOUR APPLIED BIOSYSTEMS WORKSTATION OR OTHER INSTRUMENT AND BUNDLED SOFWARE INSTALLED WITH YOUR OPERATING SOFTWARE. THIS AGREEMENT CONTAINS WARRANTY AND LIABILITY DISCLAIMERS AND LIMITATIONS. YOUR INSTALLATION AND USE OF THE APPLIED BIOSYSTEMS SOFTWARE IS SUBJECT TO THE TERMS AND CONDITIONS CONTAINED IN THIS END USER SOFTWARE LICENSE AGREEMENT.

This Applied Biosystems End User License Agreement accompanies an Applied Biosystems® software product ("Software") and related explanatory materials ("Documentation"). The term "Software" also includes any upgrades, modified versions, updates, additions and copies of the Software licensed to you by Applied Biosystems. The term “Applied Biosystems,” as used in this License, means Applera Corporation, acting through its Applied Biosystems Group. The term “License” or “Agreement” means this End User Software License Agreement. The term “you” or “Licensee” means the purchaser of this license to use the Software.

Third Party Products

This Software includes software products licensed by the following third party provider(s): **Oracle Corporation.**

Title

Title, ownership rights and intellectual property rights in and to the Software and Documentation shall at all times remain with Applera Corporation and its subsidiaries, and their suppliers. All rights not specifically granted by this License, including Federal and international copyrights, are reserved by Applera Corporation or their respective owners.

Copyright

The Software, including its structure, organization, code, user interface and associated Documentation, is a proprietary product of Applera Corporation or its suppliers, and is protected by international laws of copyright. The law provides for civil and criminal penalties for anyone in violation of the laws of copyright.

License

Use of the Software

1. Subject to the terms and conditions of this Agreement, Applied Biosystems grants the purchaser of this product a non-exclusive license to install and use the Software on the computer shipped with the SOLiD System. You may transfer the Software to one or more additional computers or you may also request that Applied Biosystems install the Software for your use on your other computers. Following such request, you will make your facilities, computer and network available to an Applied Biosystems technician to perform such installation. Applied Biosystems reserves the right to charge a fee to cover the labor associated with such an installation. Such additional Software installed on your other computers or other computer networks shall be subject to the same non-exclusive license as the initial copy of Software obtained with the purchase of the foregoing product. Notwithstanding anything in this license agreement, you acknowledge that the Software may have not been tested on computer systems other than those sold by Applied Biosystems and that Applied Biosystems makes no warrantees, guarantees or assurances with regard to the Software's performance or data integrity on such computer systems and that use of the Software on such computer systems is at your own peril.

2. If the Software uses registration codes, access to the number of licensed copies of Software is controlled by a registration code. For example, if you have a registration code that enables you to use three copies of Software simultaneously, you cannot install the Software on more than three separate computers.

3. You may make one copy of the Software in machine-readable form solely for backup or archival purposes. You must reproduce on any such copy all copyright notices and any other proprietary legends found on the original. You may not make any other copies of the Software.

Restrictions

1. You may not copy, transfer, rent, modify, use or merge the Software, or the associated documentation, in whole or in part, except as expressly permitted in this Agreement.

2. You may not reverse assemble, decompile, or otherwise reverse engineer the Software.

3. You may not remove any proprietary, copyright, trade secret or warning legend from the Software or any Documentation.

4. You agree to comply fully with all export laws and restrictions and regulations of the United States or applicable foreign agencies or authorities. You agree that you will not export or reexport, directly or indirectly, the Software into any country prohibited by the United States Export Administration Act and the regulations thereunder or other applicable United States law.

5. You may not modify, sell, rent, transfer (except temporarily in the event of a computer malfunction), resell for profit, or distribute this license or the Software, or create derivative works based on the Software, or any part thereof or any interest therein. Notwithstanding the foregoing, if this Software is instrument operating software and if this Software does not include Oracle Corporation products, you may transfer this Software to a purchaser of the specific instrument in or for which this Software is installed in connection with any sale of

such instrument, provided that the transferee agrees to be bound by and to comply with the provisions of this Agreement. Oracle Corporation prohibits any transfer of Oracle Corporation products embedded in the Software. If this Software includes Oracle Corporation products, please contact Applied Biosystems to obtain replacement operating software in connection with any sale of the instrument. A re-licensing fee may be charged for any such replacement operating software.

Note: See "Third Party Products" on page C-2 to determine if the Software includes Oracle Corporation products. If Oracle Corporation is not listed, the Software does not include Oracle Corporation products.)

Trial

If this license is granted on a trial basis, you are hereby notified that license management software may be included to automatically cause the Software to cease functioning at the end of the trial period.

Termination

You may terminate this Agreement by discontinuing use of the Software, removing all copies from your computers and storage media, and returning the Software and Documentation, and all copies thereof, to Applied Biosystems. Applied Biosystems may terminate this Agreement if you fail to comply with all of its terms, in which case you agree to discontinue using the Software, remove all copies from your computers and storage media, and return the Software and Documentation, and all copies thereof, to Applied Biosystems.

U.S. Government End Users

The Software is a "commercial item," as that term is defined in 48 C.F.R. 2.101 (Oct. 1995), consisting of "commercial computer software" and "commercial computer software documentation," as such terms are used in 48 C.F.R. 12.212 (Sept. 1995). Consistent with 48 C.F.R. 12.212 and 48 C.F.R. 227.7202-1 through 227.7202-4 (June 1995), all U.S. Government End Users acquire the Software with only those rights set forth herein.

European Community End Users

If this Software is used within a country of the European Community, nothing in this Agreement shall be construed as restricting any rights available under the European Community Software Directive, O.J. Eur. Comm. (No. L. 122) 42 (1991).

Regulated Uses

You acknowledges that the Software has not been cleared, approved, registered or otherwise qualified (collectively, "Approval") by Applied Biosystems with any regulatory agency for use in diagnostic or therapeutic procedures, or for any other use requiring compliance with any federal or state law regulating diagnostic or therapeutic products, blood products, medical devices or any similar product (hereafter collectively referred to as "federal or state drug laws"). The Software may not be used for any purpose that would require any such Approval unless proper Approval is obtained. You agree that if you elect to use the Software for a purpose that would subject you or the Software to the jurisdiction of any federal or state drug laws, you will be solely responsible for obtaining any required Approvals and otherwise ensuring that your use of the Software complies with such laws.

Limited Warranty

Applied Biosystems warrants that for a period of ninety days from the beginning of the applicable warranty period (as described below), or for the designated warranty period if a different warranty period is designated as the warranty period for the Software in the current version of an instrument operating manual or catalog or in a specific written warranty including with and covering the Software, the Software will function substantially in accordance with the functions and features described in the Documentation delivered with the Software when properly installed, and that for a period of ninety days from the beginning of the applicable warranty period (as described below) the tapes, CDs, diskettes or other media bearing the Software will be free of defects in materials and workmanship under normal use.

The above warranties do not apply to defects resulting from misuse, neglect, or accident, including without limitation: operation outside of the environmental or use specifications, or not in conformance with the instructions for any instrument system, software, or accessories; improper or inadequate maintenance by the user; installation of software or interfacing, or use in combination with software or products not supplied or authorized by Applied Biosystems; and modification or repair of the products not authorized by Applied Biosystems.

Warranty Period

The applicable warranty period for software begins on the earlier of the date of installation or three (3) months from the date of shipment for software installed by Applied Biosystems' personnel. For software installed by the purchaser or anyone other than Applied Biosystems, the warranty period begins on the date the software is delivered to you. The applicable warranty period for media begins on the date the media is delivered to the purchaser.

APPLIED BIOSYSTEMS MAKES NO OTHER WARRANTIES OF ANY KIND WHATSOEVER, EXPRESS OR IMPLIED, WITH RESPECT TO THE SOFTWARE OR DOCUMENTATION, INCLUDING BUT NOT LIMITED TO WARRANTIES OF FITNESS FOR A PARTICULAR PURPOSE OR MERCHANTABILITY OR THAT THE SOFTWARE OR DOCUMENTATION IS NON-INFRINGING. ALL OTHER WARRANTIES ARE EXPRESSLY DISCLAIMED. WITHOUT LIMITING THE GENERALITY OF THE FOREGOING, APPLIED BIOSYSTEMS MAKES NO WARRANTIES THAT THE SOFTWARE WILL MEET YOUR REQUIREMENTS, THAT OPERATION OF THE LICENSED SOFTWARE WILL BE UNINTERRUPTED OR ERROR FREE OR WILL CONFORM EXACTLY TO THE DOCUMENTATION, OR THAT APPLIED BIOSYSTEMS WILL CORRECT ALL PROGRAM ERRORS. . APPLIED BIOSYSTEMS' SOLE LIABILITY AND RESPONSIBILITY FOR BREACH OF WARRANTY RELATING TO THE SOFTWARE OR DOCUMENTATION SHALL BE LIMITED, AT APPLIED BIOSYSTEMS' SOLE OPTION, TO (1) CORRECTION OF ANY ERROR IDENTIFIED TO APPLIED BIOSYSTEMS IN A WRITING FROM YOU IN A SUBSEQUENT RELEASE OF THE SOFTWARE, WHICH SHALL BE SUPPLIED TO YOU FREE OF CHARGE, (2) ACCEPTING A RETURN OF THE PRODUCT, AND REFUNDING THE PURCHASE PRICE UPON RETURN OF THE PRODUCT AND REMOVAL OF ALL COPIES OF THE SOFTWARE FROM YOUR COMPUTERS AND STORAGE DEVICES, (3) REPLACEMENT OF THE DEFECTIVE SOFTWARE WITH A FUNCTIONALLY EQUIVALENT PROGRAM AT NO CHARGE TO YOU, OR (4) PROVIDING A REASONABLE WORK AROUND WITHIN A

REASONABLE TIME. APPLIED BIOSYSTEMS SOLE LIABILITY AND RESPONSIBILITY UNDER THIS AGREEMENT FOR BREACH OF WARRANTY RELATING TO MEDIA IS THE REPLACEMENT OF DEFECTIVE MEDIA RETURNED WITHIN 90 DAYS OF THE DELIVERY DATE. THESE ARE YOUR SOLE AND EXCLUSIVE REMEDIES FOR ANY BREACH OF WARRANTY. WARRANTY CLAIMS MUST BE MADE WITHIN THE APPLICABLE WARRANTY PERIOD.

Limitation of Liability

IN NO EVENT SHALL APPLIED BIOSYSTEMS OR ITS SUPPLIERS BE RESPONSIBLE OR LIABLE, WHETHER IN CONTRACT, TORT, WARRANTY OR UNDER ANY STATUTE (INCLUDING WITHOUT LIMITATION ANY TRADE PRACTICE, UNFAIR COMPETITION OR OTHER STATUTE OF SIMILAR IMPORT) OR ON ANY OTHER BASIS FOR SPECIAL, INDIRECT, INCIDENTAL, MULTIPLE, PUNITIVE, OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE POSSESSION OR USE OF, OR THE INABILITY TO USE, THE SOFTWARE OR DOCUMENTATION, EVEN IF APPLIED BIOSYSTEMS IS ADVISED IN ADVANCE OF THE POSSIBILITY OF SUCH DAMAGES, INCLUDING WITHOUT LIMITATION DAMAGES ARISING FROM OR RELATED TO LOSS OF USE, LOSS OF DATA, DOWNTIME, OR FOR LOSS OF REVENUE, PROFITS, GOODWILL OR BUSINESS OR OTHER FINANCIAL LOSS. IN ANY CASE, THE ENTIRE LIABILITY OF APPLIED BIOSYSTEMS' AND ITS SUPPLIERS UNDER THIS LICENSE, OR ARISING OUT OF THE USE OF THE SOFTWARE, SHALL NOT EXCEED IN THE AGGREGATE THE PURCHASE PRICE OF THE PRODUCT.

SOME STATES, COUNTRIES OR JURISDICTIONS LIMIT THE SCOPE OF OR PRECLUDE LIMITATIONS OR EXCLUSION OF REMEDIES OR DAMAGES, OR OF LIABILITY, SUCH AS LIABILITY FOR GROSS NEGLIGENCE OR WILLFUL MISCONDUCT, AS OR TO THE EXTENT SET FORTH ABOVE, OR DO NOT ALLOW IMPLIED WARRANTIES TO BE EXCLUDED. IN SUCH STATES, COUNTRIES OR JURISDICTIONS, THE LIMITATION OR EXCLUSION OF WARRANTIES, REMEDIES, DAMAGES OR LIABILITY SET FORTH ABOVE MAY NOT APPLY TO YOU. HOWEVER, ALTHOUGH THEY SHALL NOT APPLY TO THE EXTENT PROHIBITED BY LAW, THEY SHALL APPLY TO THE FULLEST EXTENT PERMITTED BY LAW. YOU MAY ALSO HAVE OTHER RIGHTS THAT VARY BY STATE, COUNTRY OR OTHER JURISDICTION.

Sun Microsystems Sublicense Grant – Terms and Conditions

(This Section is based on the terms of Applied Biosystems' license agreements Sun Microsystems, Inc., and only applies where LICENSEE purchases a license to Software that includes Sun Microsystems, Inc. products, respectively).

Licensee's use of Sun Microsystems software is also subject to the additional following terms and conditions: Licensee agrees that, to the extent permitted by applicable law, Microsystems, Inc. (if Sun Microsystems software is included) shall each be a third-party beneficiary of this License.

General

This Agreement shall be governed by laws of the State of California, exclusive of its conflict of laws provisions. This Agreement shall not be governed by the United Nations Convention on Contracts for the International Sale of Goods. This Agreement contains the complete agreement between the parties with respect to the subject matter hereof, and supersedes all prior or contemporaneous agreements or understandings, whether oral or written. If any provision of this Agreement is held by a court of competent jurisdiction to be contrary to law, that provision will be enforced to the maximum extent permissible, and the remaining provisions of this Agreement will remain in full force and effect. The controlling language of this Agreement, and any proceedings relating to this Agreement, shall be English. You agree to bear any and all costs of translation, if necessary. The headings to the sections of this Agreement are used for convenience only and shall have no substantive meaning. All questions concerning this Agreement shall be directed to: Applied Biosystems, 850 Lincoln Centre Drive, Foster City, CA 94404-1128, Attention: Legal Department.

Unpublished rights reserved under the copyright laws of the United States.

Applera Corporation, 850 Lincoln Centre Drive, Foster City, CA 94404.

Applied Biosystems is a registered trademark of Applera Corporation or subsidiaries of Applera Corporation in the U.S. and certain other countries. Oracle is a registered trademark of Oracle Corporation. FioranoMQ is a trademark of Fiorano Software, Inc. All other trademarks are the sole property of their respective owners.

Glossary

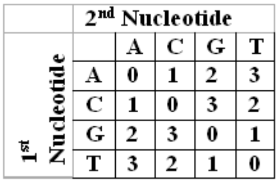
Color code

Four dye colors, encoded as follows:

- 6-FAM™ = 0†
- CY3™ = 1
- Texas Red® = 2
- CY5™ = 3

Color coding matrix

Matrix showing how the two nucleotides in the probe are encoded as a single color call. The following figure represents color coding:



		2nd Nucleotide			
		A	C	G	T
1st Nucleotide	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

Color space format

Method of presenting color-space data 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 dot (.). 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, consist of a numeric string prefixed by a single base. This base is the final base of the sequencing adapter, and it is not part of the target sequence.
- *Processed color-space*, referred to as *SOLID* or *Reference* in the 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).

Color space format *(cont'd)*

A quick way to recognize if data are pre- or post-processed is to look at the bases in all the reads from a single tag. If the bases are the same, then the data (notwithstanding some interesting tag applications) are probably unprocessed; 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 (T or G) files: xxxx.csfastaxxxx.ma	base = first base of sequence files: xxxx.bc.tab. xxxx.gff file

Color space data are self-complementary such 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 "Color Space and Base Space" on page 1-2 for more information.

Cycle Order Versus Sequence Order

Order in which data are generated (cycle) versus sequence order. The order in which the data are generated (cycle order) is different from the sequence order (with the exception of .gff files, which are in sequence order).

All data in the files containing `sequence` in the name have been transformed to sequence order, ready for alignment.

Values

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

- A dot (.) is used in color space to show that there is no call for this position. Although low values may have been measured, during clustering it was determined that no call could be made.
- A –1 in a set of calculated values (for example, N2S) means that there is no data for this point; that is, a dot (.) is in the color space for this position.
- A 0 (zero) in a set of calculated values (for example, N2S) means 0 (zero). The calculated result after rounding is 0.

Errors

Mismatches between the reference sequence in color space. As such, they are more correctly described as mismatches. They have not been edited using the extra information provided by two-base encoding.

Location

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

Panel

The region of the slide that the camera views. Returns on 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 panel numbers are unique within a slide regardless of how the slide is segmented (1,4,8 segments). The combination of panel and x,y coordinates uniquely identifies a bead.

SAT Pipeline

A framework for organizing secondary and tertiary analysis pipelines into a single front-end for convenient command-line access and automation. Any analysis code that deploys command-line interface can be used. The current pipelines call analysis modules written in C++, Perl, and Python, and used common UNIX utilities and shell commands.

Spot

When multiple samples are run on a single slide, each slide has its own spot.

Each sample has its own directory containing results. Each spot consists of a series of panels. All panels have unique numbers, that is, numbering is continuous across spots.

TAG_ID

A unique identifier for every tag, which consists of four components: *panel_xpixel_ypixel_tagtype*. For example, *1_567_321_R3* describes a bead in panel 1 at coordinates 56, 321 (X,Y) with the R3 tag (second tag in a mate pair).

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.

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 nth position of the tag is numbered $(n - 1)$ in the forward direction and $-(n - 1)$ in the reverse direction.

Index

Worldwide Sales and Support

Applied Biosystems vast distribution and service network, composed of highly trained support and applications personnel, reaches 150 countries on six continents. For sales office locations and technical support, please call our local office or refer to our Web site at **www.appliedbiosystems.com**.

Applera is committed to providing the world's leading technology and information for life scientists. Applera Corporation consists of the Applied Biosystems and Celera Genomics businesses.

Headquarters
850 Lincoln Centre Drive
Foster City, CA 94404 USA
Phone: +1 650.638.5800
Toll Free (In North America): +1 800.345.5224
Fax: +1 650.638.5884

04/2008

Applied Biosystems