

# SOLiD™ 3 System Application Documentation

## SOLiD BaseQV Tool

Document Version 1.0, 15 May 2009

### 1 Program

#### 1.1 Program Name

SOLiD BaseQV Tool

#### 1.2 Program Version

V1.0

### 2 Description

#### 2.1 Application

The SOLiD BaseQV software application converts a SOLiD™ Gff-format data file to SAM format.

#### 2.2 Description

This software takes Gff v.2 SOLiD™ reads as input and converts them to SAM format. It supports both mates and fragment-sorted Gff files. Using color space rules, it uses color quality values to predict base quality values, given the annotations generated by the base translation algorithm in the Gff file. Other features include generation of mate flags in SAM format, generation of MD and CIGAR tags, and ASCII-space quality values. For GFF mate-pair files sorted by position, automated split-and-merge of reads into chromosomes guarantees minimal memory usage. This also guarantees correct flags for each mate pair. The program supports parameters for SAM header settings and generates a Read Group (RG) tag for each of the reads. The color space reads and color quality values are all in 5' to 3' orientation; in contrast, the base reads and base quality values are represented on the forward genomic strand.

#### 2.3 Release Notes

This is a stable release with no known bugs. Because the GFF file contains only uniquely mapped reads or read pairs to SAM format, the SAM file also contains only uniquely mapped reads. Mapping quality values for all reads are set as high quality.

#### 2.4 Algorithm/Script Description

C++ code compiled/run from the command line.

## 2.4.1 Base QV Algorithm Description

Base quality values use the properties of color space error detection to predict the base accuracy. If the two colors on the sides of a read are both the same as the reference, or are “valid adjacent” color changes (for example, two colors consistent with a SNP), then the base quality values are the sum of the two color quality values. If color differences from the reference are not consistent with a SNP or are predicted to be unreliable, then the base quality value is the minimum of the relevant color quality values. Consider the example in which only one of the two colors that measure a base is different from the reference; this case is not consistent with a SNP.

Applied Biosystems has observed that the base quality values are accurate; base quality values have a linear relationship with the predicted base accuracy (as measured by human mismatches to the reference at non-dbSNP positions). Applied Biosystems has observed that base quality values can discriminate between true-positive and false-positive SNPs.

## 3 Installation instructions

A Makefile is provided to compile SOLiD BaseQV Tool from source code. Simply type:

```
make
```

to build the main executable (GffToSam). If you have an existing Corona Lite installation and want to include this tool, type:

```
make install PREFIX=$CORONAROOT
```

where CORONAROOT is the root of the Corona Lite installation.

SOLiD BaseQV has been built with Gnu g++ version 3.4.6.

## 4 Usage

### 4.1 Required input parameters

Parameter	Name / Description	Type / Range / Example
-i	Gff input file name or full path to gff input file	If the input ends with .gz, it is auto-decompressed using systems call to “gzip” library. It is finally converted and compressed back to the original file.  Usage: -i input.gff
-id	Unique identifier for reads file	A short, unique identifier should be added to the @RG header of SAM file, and to the RG tag of each alignment line.  Usage: -id S1
-sm	Sample name	Sample name, such as NA1907.  Usage: -sm NA19240

## 4.2 Optional input parameters

Parameter	Name / Description	Type / Range / Example
-o	Output file name	Full path to SAM output file. Default: 'Gff input file'.sam Usage: -o output.sam
-h	Help	Display parameter description. Usage: -h
-d	Run in debug mode	This parameter prints the status of program process. Usage -d
-m	Mates file	This parameter should be specified if no Gff header is present in the file. It is set as "true" for mates files and "false" for frag files. Usage -m t or -m f Default: -m f
-ms	'Mates-sorted' mates file	This parameter should be specified if no Gff header is present in the file. It is set as "true" for mates-sorted mates files, and "false" for position- (fragment) sorted mates files. Usage -ms t or -ms f Default: -ms f
-a	Ascii-33 format Quality values	This parameter allows the user to print integer quality values if set as "false". Because SAM format accepts quality input only in Ascii-33 format, this should be changed only for test/display purposes. Usage -a t or -a f Default: -a t
-c	Color tags in SAM file	CS (color sequence) and CQ (color quality) tags are added to SAM format. CS is the original color read sequence, and is not aligned to the genome. For reads that map to the negative strand, the color sequence can be reverse-complemented to generate the base translation that is shown on the forward genomic strand. Usage -c t or -c f Default: -c t

-maxqv	Max base QV	<p>Change this value to cap base quality values to the given maximum. For example, if SOLID™ color qualities range from 0 to 35, base quality values could range from 0 and 70. To ensure that base quality corresponds to a phred scale probability of error, the default maximum is set to 45.</p> <p>Usage: -maxqv 40</p> <p>Default: 45</p>
-gch	Grab Contig names from Header	<p>The default for this option is “false”, and contigs are named as the integer provided at “i=” tag of the gff file. If the user wants to auto-name the reference field in SAM according to information provided at Gff header (such as ## Contig 1 CHR1), this option should be set to “true”.</p> <p>Usage – gch t or – gch f</p> <p>Default: - gch t</p>
-flg	Gff Software Version Used (a.k.a First / Last Good)	<p>If version 0.7 of MatesToGff was used, set this flag to “false”. Otherwise, the default option supports MatesToGff version 0.8 or higher. There is also an automated version detection code, which overrides this setting if the version information is provided in the Gff header.</p> <p>Usage –flg t or –flg f</p> <p>Default: -flg t</p>
-rg	Read group identifier	<p>Add RG tags to every alignment.</p> <p>Usage: -rg t or –rg f</p> <p>Default: -rg t</p>
-male	Male sample	<p>Male sample and female sample reference names may differ. If the user wants to set reference names for specific samples in a different way, this parameter may be used. The code needs to be altered at <i>setRefIndex</i> (<i>string ref_index</i>) function of the code. If the contigs are named according to gender in Gff headers, this option is not necessary.</p> <p>Usage: -male</p>
-female	Female sample	<p>See male description above.</p> <p>Usage: -female</p>

Optional Header Tags	Name / Description	Type / Range / Example
-pu	Platform unit – solid slide name	Usage: -pu JOAN_20080121_1
-lb	Library name	Usage: -lb Library_name
-cn	Sequencing center where read is produced	Usage: -cn ABI_Foster_City
-pi	Median insert size	Usage: -pi 1500
-ds	Description	Usage: -ds “DESCRIPTION”
-dt	Date when run was produced	Usage: -dt 2008-01-21
-pl	Platform technology	Usage: -pl SOLID_v3

### 4.3 Usage Parameters: Is parameter order important?

No. Enter the parameters in any order.

### 4.4 Usage Example

```
./GffToSam -h // Display help
./GffToSam -i x.gff -o x.sam -id S1 -sm NA19507 // Debug mode
./GffToSam -i x.gff -o x.sam -id S1 -sm NA19507 // Regular run
./GffToSam -i x.gff -id S1 -sm NA19507 // Output name will be x.gff.sam
./GffToSam -i x.gff.gz -id S1 -sm NA19507 -pu JOAN_20080121_1 -gch true -male
-maxqv 40 // Provide more options to specialize final SAM
```

### 4.5 List of programs/scripts included

GffToSam

### 4.6 Other scripts called by this program

None.

### 4.7 Path constraints

Input arguments contain the full path of the expected input file. There are no constraints.

## 4.8 System Input Files

GFF v2 – sorted in frag order or mates order. Order and Mates/Frag file status must be specified in gff header. If gff header is missing, order can be specified by `-m` and `-ms` options. Otherwise, it is assumed to be a fragment file sorted by position.

## 4.9 Input File Versions supported

GFF v2 for the SOLiD™ System.

## 4.10 Additional Input Files

None.

## 4.11 Input File Comments

If there are reads that start at a “minus” position (because of a bug in an early version of the Corona rescue pipeline), both that read and its mate are not included as SAM records.

## 4.12 Output File(s)

SAM output is generated according to the regular SAM specification. You can find a description of the SAM format at this web address: <http://samtools.sourceforge.net/>

## 4.13 Output File(s) Comments

See the above web page. Optional SOLiD™ headers include CS and CQ. These are described in the SAM documentation.

# 5 Other

## 5.1 Development language

C++

## 5.2 Is PBS required?

No. All reads from a single run that map to the same chromosome must be present in a single file. Different chromosome gff's can be analyzed separately and merged using SamTools.

## 5.3 Comments

**Memory requirements:** Because the gff files are assumed to be sorted in position order, finding the mates is guaranteed by keeping minimal information on each read in memory. This is done per chromosome. Therefore, the program requires **20 bytes of memory per read per chromosome**. For example, if the largest chromosome in the file has 100 million reads, 2 GB allocatable memory is required. Otherwise, the program will crash.

Typically, different runs (separate gff files) should not be merged into one file. If a single contig (chromosome) contains as many reads that use up the above memory requirement, Applied Biosystems suggests the following strategy to convert to SAM format without any loss of data:

1. Generate a gff file in mates-sorted order.

2. Split this file into smaller chunks that each fit in memory. Ensure that each of the mates goes to the same file. Use GffToSam to generate separate SAM files with the same ID, tag, and other header options.
3. Merge the SAM files using Samtools or a similar tool.

## **5.4 Date**

May 15, 2009