Data File Formats

V2.1

Complete Genomics, Inc.
2071 Stierlin Court
Mountain View, CA, 94043

This page intentionally left blank

Complete Genomics

Table of Contents

1 Introduction

This document describes the directory structure and file formats for complete genome sequencing data delivered by Complete Genomics, Inc. (CGI) to customers and collaborators. The data includes sequence reads, their mappings to a reference human genome, and variations detected against the reference human genome.

Disclaimer of Warranties. COMPLETE GENOMICS, INC. PROVIDES THESE DATA IN GOOD FAITH TO THE RECIPIENT "AS IS." COMPLETE GENOMICS, INC. MAKES NO REPRESENTATION OR WARRANTY, EXPRESS OR IMPLIED, INCLUDING WITHOUT LIMITATION ANY IMPLIED WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE OR USE, OR ANY OTHER STATUTORY WARRANTY. COMPLETE GENOMICS, INC. ASSUMES NO LEGAL LIABILITY OR RESPONSIBILITY FOR ANY PURPOSE FOR WHICH THE DATA ARE USED.

Any permitted redistribution of the data should carry the Disclaimer of Warranties provided above.

Data file formats are expected to evolve over time. Backward compatibility of any new file format is not guaranteed.

2 Sequencing Approach

Complete Genomics' sequencing platform employs high-density DNA nanoarrays that are populated with DNA nanoballs (DNBs™) and base identification is performed using a non-sequential, unchained read technology, known as combinatorial probe-anchor ligation (cPAL™).

Complete Genomics' sequencing technology, including the library construction process and the ligation-based assay approach, is described in the Complete Genomics Technology Whitepaper, available in the "Resources" section of the Complete Genomics website (www.completegenomics.com). This section also describes the data structure, the nomenclature used, and the contents and organization of the data files.

Read Data Format

Each slide, containing an ultra-high density DNA nanoarray, is partitioned into several lanes. A field is a region within a lane that is imaged at one time; each field covers a two-dimensional array of spots on the slide, the vast majority of which are occupied by a single DNB. The DNB is a head-to-tail concatamer consisting of more than 200 copies of a circular DNA template comprised of genomic DNA and several synthetic adaptors. A library is a collection of these paired-end constructs processed together from genomic DNA and the known adaptors. Figure 1 depicts the architecture of the circular template and of the reads generated from a single four-adaptor DNB.

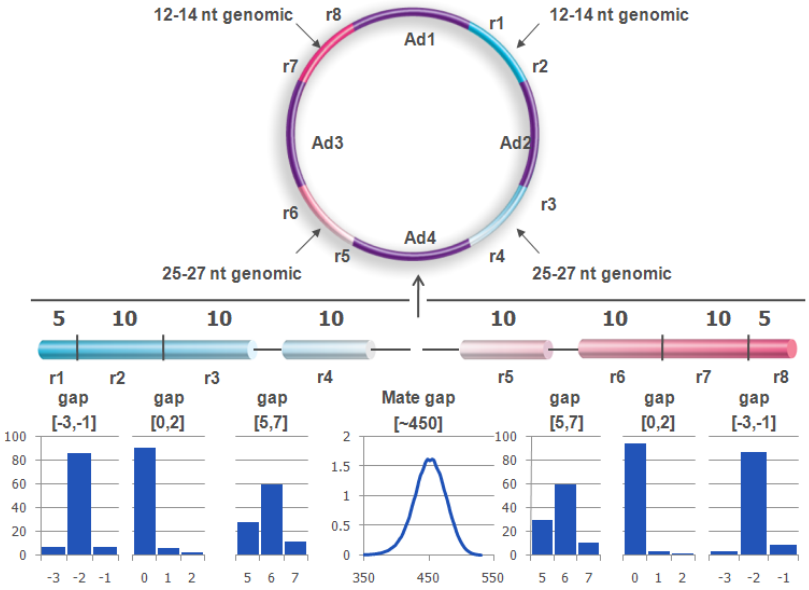

Figure 1 Gapped read structure

Each DNB consists of two paired reads, called half-DNBs, separated by a physical distance referred to as the mate gap. Within each half of the DNB, reads of genomic DNA are obtained from the ends of each adaptor (reads r1 – r4 correspond to one half-DNB and r5 – r8 to the other half-DNB in Figure 1). These reads do not include adaptor sequence. Neighboring reads within each half-DNB are proximal in genomic coordinates but may be separated from each other by small gaps (positive values, in bases), or may overlap one another (represented by negative values, in bases). The plot in the bottom-half of Figure 1 displays typical distributions for the gaps and overlaps associated with reads from a single, four-adaptor DNB. Actual gap distributions are empirically estimated from sampled data. DNB positions in output files refer to positions within an aggregation of the reads obtained from each DNB. In Figure 1, these are positions within the seventy bases (5 + 10 + 10 + 10 + 10 + 10 + 10 + 5) constructed by aggregating reads r1 – r8 in order of genomic position. Note that because proximal reads (such as r1 and r2 above) can overlap, two read positions may correspond to a single genomic location.

3 Directory structure on the hard drive (s)

Data for sequenced human genomes will be provided on one or more hard drives. The hard drives are formatted with the NTFS file system that can be read by a variety of operating systems. To install the hard drives, please refer to the documentation provided with them.

The data is stored in the folder structure that is shown in Table 1. Some of the files are signed using S/MIME technology to ensure data integrity, using the PKCS #7 secure message format specification (Public Key Cryptography Standards #7, published by RSA Security).

```
`-- Package [Single data delivery]
    |-- DOC [File format documentation and example scripts]
    |   |-- EXAMPLES
    |   |   |-- cgiMapLib.py
    |   |   |-- cgiMapReader.py
    |   |   |-- cgi2SAM.py
    |   |   `-- cgiGenomeDataTransformer.py
    |   `-- CGIFileFormat_0.2.pdf
    |
    |-- GS00011-DNA-C01 [single individual genome]
    |   |
    |   |-- ASM [data on assembled genome: variations, annotations]
    |   |   |-- REF [base-level coverage and reference scores, organized by chromosome]
    |   |   |   |-- coverageRefScore-4-GS000000123-ASM.tsv.gz
    |   |   |   |-- coverageRefScore-4-GS000000123-ASM.tsv.gz
    |   |   |   |-- coverageRefScore-4-GS000000123-ASM.tsv.gz
    |   |   |   `-- coverageRefScore-4-GS000000123-ASM.tsv.gz
    |   |   |
    |   |   |-- gene-GS000000123-ASM.tsv [genic annotation of variations]
    |   |   |-- dbSNPAnnotated-GS000000123-ASM.tsv [calls on dbSNP variations]
    |   |   |-- reg-GS000000123-ASM.tsv [block boundaries for called portions of the genome]
    |   |   |-- var-GS000000123-ASM.tsv [variations discovered with respect to reference genome]
    |   |   `-- summary-GS000000123-ASM.tsv [summary of assembly statistics]
    |   |
    |   `-- MAP [reads, scores, mappings, and associated data]
    |       |-- GS00009-FS3-L04 [section of data, currently one slide lane]
    |       |   |-- reads.tsv.gz [reads and scores for all reads]
    |       |   |-- lib_DNB_GS00021-CLS.tsv [DNB read architecture for this library]
    |       |   `-- mapping.tsv.gz [mappings to the reference genome for reads in reads.tsv.gz]
    |       |
    |       `-- GS00009-FS3-L05
    |           |-- reads.tsv.gz
    |           |-- lib_DNB_GS00021-CLS.tsv
    |           `-- mapping.tsv.gz
    |
    |-- README.txt [README accompanying data set]
    |-- manifest.all [manifest of files]
    `-- version [version of export format]
```

Table 1 The file hierarchy on the hard drives

3.1 Manifest.all

`manifest.all` is a `sha256sum`-compatible file into which all of the checksums for all files written to disk are recorded. This file is signed using S/MIME to assure that file corruption or modification can be detected.

3.2 Reads, Mapping and Assembly data

The data corresponding to a single genome is organized into three folders:

1) DOC – Documentation and example scripts
2) MAP – Reads, quality scores and alignments to the reference genome
3) ASM – Assembly of the complete genome: variations called, coverage, and annotations

The representation of reads, quality scores and alignments has been designed as a transfer format, dominated by considerations of simplicity and compactness. For some applications, this could result in increased cost in accessing particular subsets of interest within the data (see section "Association between ***mappings.tsv*** and ***reads.tsv***").

3.2.1 Header format

Each data file in the directory structure contains a header section that describes the contents of the file and provides associated metadata. Each header row begins with the hash character (#) followed by a tab-separated, key-value pair. The keys and their possible values are described below.

Key	Description	Allowed values
#TYPE	**Indicates the type of data contained in the file.**	**READS:** **reads file** **MAPPINGS:** **alignments of reads to the reference genome.** **LIB-DNB:** **description of the architecture of reads within DNBs in a library.** **REFMETRICS:** **reference scores (scores indicating the likelihood of the assembled genome being identical to the reference at each genomic position) and coverage information** **DBSNP-TO-CGI:** **information on loci annotated in dbSNP** **GENE-ANNOTATION:** **variations annotated with impact on RefSeq genes** **SUMMARY-REPORT:** **summary information on the assembled genome** **VAR-ANNOTATION:** **information on the assembled genome, expressed relative to the reference genome.**
#VERSION	**Version number of the file format, e.g. 1.0**	
#LIBRARY	**Identifier of the library that the DNBs were generated from**	
#SAMPLE	**Identifier of the sample that the library was created from**	
#LANE	**Identifier of the slide lane that the reads were extracted from**	
#CHROMOSOME	**Identifier of the chromosome that the reference score and coverage data apply to**	**1-22, M, X, Y**

Table 1: Header Metadata

The header section is followed by a single row of tab-separated column headers that begins with the "greater than" character '>'; followed by the data, also in a tab-separated format. An example from the ***lib_DNB_[library_name].tsv*** file is shown below:

```
#BUILD          1.4.0.10
#GENERATED_AT   2009-Sep-17 23:02:48.730999
#GENERATED_BY   exportTools
#LIBRARY        GS00321-CLS
#SAMPLE         GS00123-DNA-D01
#TYPE           LIB-DNB
#VERSION        0.2

>id     type    armId   indArm  objArm    min     max
0       read    0       0       0         5       5
1       gap     0       1       0         -3      -1
```

3.2.2 DOC Folder – Documentation and Examples

The DOC folder contains documentation (this file "***CGIFileFormat_0.2.pdf***") and example Python scripts.

3.2.3 MAP Folder – Reads and Mapping Data

The MAP folder contains reads, scores, and alignments to the reference genome for each DNB, organized by slide and lane. Each subfolder name is the identifier for the lane, for example "***GS08089-FS3-L01***" would represent data for the first lane (L01) of the slide "GS08089-FS3".

The following sections describe the files in each lane subdirectory within the MAP folder.

reads.tsv.gz:

A tab-delimited text file (compressed with ***gzip***) containing the reads and associated quality scores.

Name	Description	Text Format
flags	Mapping characteristics of the DNBs, represented in bits within an integer. Individual flags described below.	Integer (base 10), e.g. 8.
flag: LeftHalfDnbNoMatches	The left half of this DNB yielded no mappings to the reference genome.	0x01
flag: LeftHalfDnbMapOverflow	The left half of this DNB yielded a large number of mappings to the reference genome [indicative of highly repetitive sequence; mappings not tracked for this half-DNB].	0x02
flag: RightHalfDnbNoMatches	The right half of this DNB yielded no mappings to the reference genome.	0x04
flag: RightHalfDnbMapOverflow	The right half of this DNB yielded a large number of mappings to the reference genome [indicative of highly repetitive sequence; mappings not tracked for this half-DNB].	0x08
reads	The base calls read from a single DNB, in an order specified in **lib_DNB_<library_id>.tsv**. Base positions for which no information is available are denoted by 'N' in the "reads" field.	one character per base, not separated
scores	Quality scores for reads. Each score is a Phred-like transformation of the error probability associated with a single base read. Base positions for which no information is available are assigned a score of 0	one Ascii-33-encoded character per base, not separated.

Table 2 Reads file format description

A sample set of rows from a ***reads.tsv*** file is presented below for hypothetical DNBs of length 20, showing the Ascii-33-encoded, single-character quality scores. DNBs with the structure illustrated in Figure 1 would have 70 bases and corresponding scores, with the first 35 bases corresponding to the left half-DNB and the last 35 bases to the right half-DNB.

```
>flags     reads                 scores
1          AGTGAGACACCTGAGGGNGA  SXXX<NDUETSUBTMW]#\Z
3          AAATATATTTTGTAGTCNAG  PKMZH@+E6CN)KJ)[)#Z5
0          CTTCTCTGGTTTATTGTNTG  UXW6XTTP/R(0MST3[#],
```

The interpretations of all allowed values for the **flags** field are described below:

flags	0	1	2	4	5	6	8	9	10
LeftHalfDnbNoMatches		x			x			x	
LeftHalfDnbMapOverflow			x			x			x
RightHalfDnbNoMatches				x	x	x			
RightHalfDnbMapOverflow							x	x	x

A value of flags = 0 indicates that both arms of the DNB mapped to the reference genome. Values of 4 or 8 indicate that mappings are available only for the left arm; values of 1 or 2 indicate that mappings are available only for the right arm.

lib_DNB_[LIBRARY-NAME].tsv:

This is a tab-delimited, text file that describes the architecture of reads and gaps within all DNBs in the library. This information is useful in the interpretation of reads in ***reads.tsv***. The DNB is described as a series of objects of different types (reads, gaps, mate gap) sequentially following one another.

Column Name	Description	Text format
`id`	Position of the object within each DNB, numbered from 0 to n-1, where n is the number of objects (reads and gaps) within each DNB	int
`type`	Object type: currently one of "read","gap","mategap"	string
`armID`	Number of the half-DNB: 0-left, 1-right	int
`indArm`	0-based position of the object within an arm	int
`objArm`	0-based position of this object type within an arm, e.g. the second gap within the second arm has “1” for this field.	int
`min`	Minimum length in bases for the object. N.B. The minimum and maximum values for gaps given in this table are prior estimates. However, revised estimates are determined from the data within the Complete Genomics assembly software	int
`max`	Maximum length in bases for the object. Blank when maximum is not specified. N.B. The minimum and maximum values for gaps given in this table are prior estimates. However, revised estimates are determined from the data within the Complete Genomics assembly software	int

Table 3 Read structure file format description

An example of the ***libDNB-[LIBRARY-NAME].tsv*** file is shown below for the DNB architecture depicted in Figure 1:

```
>id     type    armID   indArm      objArm      min     max
0       read    0       0           0           5       5
1       gap     0       1           0           -3      -1
2       read    0       2           1           10      10
3       gap     0       3           1           0       0
4       read    0       4           2           10      10
5       gap     0       5           2           5       7
6       read    0       6           3           10      10
7       mategap 0       7           3           250
8       read    1       0           0           10      10
9       gap     1       1           0           5       7
10      read    1       2           1           10      10
11      gap     1       3           1           0       0
12      read    1       4           2           10      10
13      gap     1       5           2           -3      -1
14      read    1       6           3           5       5
```

mapping.tsv.gz:

This tab-separated, text file contains mapping information to the reference genome (compressed with ***gzip***) for the reads in ***reads.tsv.gz***. Each row corresponds to the alignment of a single half-DNB to the reference genome, with information on the most likely mate for this half-DNB. This file does not contain the bases and scores for each read. However, the mappings for each read are stored sequentially and in the same order as in ***reads.tsv.gz***. This format does not allow for random access to a genomic location, and retrieval of reads and mappings corresponding to one or several genomic regions would require a full scan of both files.

We provide example scripts for such a scan, also enabling translation into other formats such as SAM, in the DOC/EXAMPLES/ folder.

Column Name	Description	Text format
flags	Mapping characteristics encoded in bit fields, described below	integer
flag: LastDNBRecord	Set if the current mapping is last mapping record of the DNB	0x01
flag: side	The arm within the DNB that yielded this mapping. The left arm (i.e. first half of the bases in the *reads* column of ***reads.tsv.gz)*** is represented by 0; the right arm (i.e. second half of the bases in the *reads* column of ***reads.tsv.gz)*** is represented by 1., Right - 1	0x02
flag: strand	forward - 0, reverse - 1	0x04
Chromosome	Chromosome name in text: “1”,”2”,…,“22”,”X”,”Y” The pseudoautosomal regions for the sex chromosomes X and Y are represented by their coordintes oan chromosome X.	
offsetInChr	Starting coordinate on chromosome, 0-based (see section “Sequence Coordinate System” for more information).	
gap1 .. gap[n]	There are *n* tab-separated gap fields, where *n* is the number of gaps in the half-DNB as defined in ***lib_DNB_[LIBRARY-NAME].tsv.*** Currently n = 3, i.e. there are 3 gaps per half-DNB***.*** The column contains the length of each gap within the half-DNB. Gaps are listed in order of chromosomal position. Overlaps are represented as negative numbers.	integer
weight	Mapping weight. This is a Phred-like encoding of the probability that this half-DNB mapping is incorrect.	Ascii-33
mateRec	Zero-based index of the best mate for the current half-DNB, counting within the half-DNB mappings for the current DNB. Equals the index of the current mapping if no mate mappings are found.	integer

Table 4 Mapping file format description

The allowed values for the **flags** field in ***mappings.tsv.gz*** and their interpretation are shown below.

flags	LastDNBRecord	side	strand
0	no	Left	+
1	yes	Left	+
2	no	Right	+
3	yes	Right	+
4	no	Left	-
5	yes	Left	-
6	no	Right	-
7	yes	Right	-

A sample set of rows from a ***mappings.tsv.gz*** file is shown below:

```
>flags  chromosome    offsetInChr    gap1    gap2   gap3    weight  mateRec
0       18            54911965       -2      0      5       (       1
3       18            54912325       5       0      -3      (       0
0       7             92578954       -2      0      6       !       3
0       8             59803146       -2      0      6       !       4
4       19            19695620       4       0      -2      !       5
2       7             92579332       6       0      -3      !       0
2       8             59803538       6       0      -3      !       1
7       19            19695239       -3      0      6       !       2
4       7             101416273      6       1      -2      L       1
7       7             101415891      -2      0      5       L       0
5       8             85763053       5       0      -2      j       0
```

Note that in accordance with the column definitions, flags that are odd numbers signify the last mapping record for a DNB. Thus, in the above example, mappings for four DNBs are shown:

1. For the first DNB, there is one mapping available for each half-DNB, with both close to one another on chromosome 18. The `mateRec` field for the two half-DNB mappings is populated with 1 and 0 respectively, indicating that these two are best mates for one another. Based on the **flags** values of 0 and 3, it is shown that both half-DNBs map to the forward strand.
2. For the fourth and last DNB, there is only one mapping available. Based on flags = 5, it can be inferred that it is a mapping of the left half-DNB to the reverse strand of the reference genome. The offsetInChr field (representing the starting coordinate of the mapping, in zero-based half-open coordinates described in the section “Sequence coordinate system”) and gap fields are described with respect to the forward strand, however, and not in the order of the bases in reads.tsv.gz. That is, for the DNB architecture represented in Figure 1, the 35 bases in this reverse-strand-mapped, left half-DNB map to the right of offsetInChr, with contiguous reads of 10, 10, 10 and 5 bases separated by gaps of 5, 0 and -2 bases respectively (the last being an overlap of two bases). Because no mate mapping was found for this half-DNB, mateRec is populated with its own record position within the mappings for the DNB, which is 0.
3. The third DNB has one mapping available for each half-DNB on chromosome 7, both on the reverse strand based on the values of flags. Again, mateRec indicates that the two mappings are mated with one another.
4. The second DNB, represented in rows 3 – 8 of the example, has six, half-DNB mappings. The mateRec field values for these rows indicate that this DNB has three pairs of mated mappings on the genome: one each on chromosomes 7, 8 and 19. For example, the record numbers of the two chromosome 7 mappings within the set for this DNB are 0 and 3; the mateRec fields in these records are 3 and 0 respectively. The values of flags indicate that the first three rows (rows 3 – 5 in the example) correspond to the left half-DNB and the next three rows (rows 6 – 8 in the example) correspond to the right half-DNB; they also indicate that the chromosome 19 mappings are to the reverse strand.

Within DOCS/EXAMPLES, ***cgiMapReader.py*** is a Python script that processes the mapping files and extracts these pieces of information.

Association between mappings.tsv and reads.tsv:

DNB mappings in ***mappings.tsv*** are stored in the same order as records for DNBs in the ***reads.tsv*** file, allowing for an association between them. Within a DNB, all left-arm mappings precede right-arm mappings. The number of mapping records corresponding to each DNB is variable, and flags within the two files help to associate records within the two with each other.

The ***reads.tsv*** file includes read and score data for each DNB that passes basic quality filters. The flags corresponding to each DNB contain information on whether each of its constituent half-DNBs yielded mappings to the reference genome. There are three possibilities for each DNB:

1. If either LeftHalfDnbNoMatch or LeftHalfDnbMapOverflow is set to 1, no mapping records are expected for the left half-DNB in mappings.tsv.
2. If either RightHalfDnbNoMatch or RightHalfDnbMapOverflow is set to 1, no mapping records are expected for the right half-DNB in mappings.tsv.
3. The last half-DNB mapping record in mappings.tsv corresponding to this read will have the LastDNBRecord flag set to 1, indicating that the next mapping record corresponds to a new DNB.

Using the above rules, it is possible to scan the ***mappings.tsv*** and ***reads.csv*** files together, associating the mappings in ***mappings.tsv*** with reads and scores in ***reads.csv***. Mappings are associated with the next record in ***reads.csv*** following a record with the `LastDNBRecord` flag set to 1; however, records in ***reads.csv*** for which no mappings are expected, due to rules (1) and (2) above, are skipped. ***cgiMapReader.py*** is an example Python script provided with the data that implements such a scan.

3.2.4 ASM Folder - Assembly and variations identified

The files in this folder describe and annotate the genome assembly with respect to the reference genome. The ASM folder contains the primary results of the assembly within two files: ***var-[ASM-ID].tsv*** and ***reg-[ASM-ID].tsv.*** The file ***reg-[ASM-ID].tsv*** denotes blocks of the reference sequence that were either called identical to the reference, or in which specific variations were called; regions of the reference genome that were not called at all are not represented within this file. The file ***var-[ASM-ID].tsv*** includes a description of all loci where the assembled genome differs from the reference genome.

In addition to these files, annotations of the assembled sequence with respect to the dbSNP database, RefSeq transcripts, and protein sequences are included. Also included in the REF subdirectory are files containing supplementary information: the sequence coverage at each reference genomic position and a score indicating the likelihood of the genome being homozygous and identical to the reference at each position.

var-[ASM-ID].tsv

Variation records have the following fields:

Column #	Column Name	Description
1	locus	Identifier of a particular locus of variation
2	haplotype	Identifier for each haplotype at the variation locus. For diploid genomes, 1 or 2.
3	chromosome	Chromosome name in text: "1","2",...,"22","X","Y" The pseudoautosomal regions for the sex chromosomes X and Y are represented by their coordinates on chromosome X.
4	begin	Reference coordinate specifying the start of the variation (*not the locus)* using the half-open zero-based coordinate system. See section "Sequence Coordinate System" for more information.
5	end	Reference coordinate specifying the end of the variation (*not the locus)* using the half-open zero-based coordinate system. See section "Sequence Coordinate System" for more information.
6	varType	Type of variation, currently one of: **snp:** single-nucleotide polymorphism **ins:** insertion **del:** deletion **delins:** Substitution of reference bases with the bases in the allele column **'='** : no variation; the sequence is identical to the reference sequence on the indicated haplotype **ref-consistent**: when one or more bases are ambiguous, but the allele is potentially consistent with the reference **ref-inconsistent**: when one or more bases are ambiguous, but the allele is definitely inconsistent with the reference **no-call**: No call could be made for this allele
7	reference	The reference sequence for the locus of variation. Empty when varType is **ins**.
8	alleleSeq	The observed sequence at the locus of variation. Empty when varType is **del**. "**?**" is used to indicate 0 or more unknown bases within the sequence; "**N**" is used to indicate exactly one unknown base within the sequence.
9	totalScore	A score corresponding to a single variation and haplotype, representing the confidence in the call.
10	hapLink	Identifier that links a haplotype at one locus to haplotypes at other loci. Currently only populated for very proximate variations that were assembled together.
11	xRef	Field containing external variation identifiers, currently only populated for variations corroborated directly by dbSNP. Format: dbsnp:[rsID], with multiple entries separated by the semicolon (;).

Table 5 Variations block description

An example of a portion of the "***var-[ASM-ID].tsv"*** file is shown below:

>locus	haplotype	chromosome	begin	end	varType	reference	alleleSeq	totalScore	hapLink	xRef
976	1	1	835145	835146	snp	G	T	87		dbsnp:806721
976	2	1	835145	835146	snp	G	T	58		dbsnp:806721
977	1	1	835212	835215	=	GTC	GTC	36		
977	2	1	835212	835215	ref-consistent	GTC	?	36		
978	1	1	835363	835363	ins	G		47		
978	2	1	835363	835363	=			55		
979	1	1	836464	836465	del	T		57		
979	2	1	836464	836465	del	T		65		
980	1	1	838600	838601	=	C	C	120		
980	2	1	838600	838601	snp	C	T	479		
1043	1	1	849559	849563	=	ACGG	ACGG	65	779	
1043	1	1	849563	849564	snp	C	G	47	779	
1043	1	1	849564	849566	=	GT	GT	69	779	
1043	2	1	849559	849566	ref-consistent	ACGGCGT	?	780		
1044	1	1	849569	849570	=	C	C	47	779	
1044	2	1	849569	849570	ref-inconsistent	C	G?	45	780	

Notes:

1) The first set of variations (locus ID=976) is an example of a homozygous SNP call, where the reference sequence is a 'G' and the assembled genome has two copies of the 'T' allele. The confidence score for the existence of at least one 'T' allele is 87 and the confidence score for the existence of two 'T' alleles is 58. This variation has the dbSNP identifier "rs806721".
2) Variation ID 978 is an example of an insertion event in one of the haplotypes. An insertion of a 'G' is seen at position 835363 in haplotype 1, while haplotype 2 has the reference sequence, with a `vartype` of '='.
3) A homozygous deletion of a 'T' is found in variation ID 979 at position 836464, indicated by the calling of a 'del' variation in both haplotypes.
4) A heterozygous SNP 'C/T' call is found in variation ID 980, where reference shows a 'C" and the assembled genome has a 'C' allele in one haplotype and a 'T' in the other.
5) Variation ID 977 shows an example where only one of the two haplotypes is called. The assembled genome is identical to the reference (in this case, the bases 'GTC') on one haplotype, while the other allele could not be called due to competing alternate hypotheses that could not be adequately discriminated. The `alleleSeq` column shows '?' in this case. The type of allele is '***ref-consistent'***, which indicates that although the assembly software did not make a call for this region, it may be consistent with the reference sequence. It is possible that the other allele is 'GTC' as well, but there is not enough information in the data to support that conclusion.
6) An example of a '***ref-inconsistent***' call is shown for locus 1044. One haplotype of the assembled genome is identical to the reference (a 'C' at position 849569), but on the other haplotype the 'C' has been replaced by a 'G', and there is uncertainty about the insertion of more bases to the right of this one (indicated by '?').
7) The locus with ID = 1043 depicts a more complex situation, where there are three calls for one haplotype (1) and a 'ref-consistent' unresolved call for the other haplotype. There is only one variation call on haplotype 1 (a SNP at position 849564) but neither the length nor the composition of the sequence on the other haplotype could be reliably determined over this locus. This variation also has a value in the `haplink` column (780) which links this variation to variation with ID 1043. This indicates that these variations are in phase with one another.

reg-[ASM-ID].tsv

This file lists non-overlapping ranges of the reference genome and indicates if the region contains any variations or if the region is identical to the reference genome. Regions of the reference genome not covered by blocks in this file could not be resolved, and should be interpreted as 'no-calls' rather than as reference calls. Some of the reasons for these regions not being called are:

1) The region corresponds to an undefined block of sequence (a string of 'N') in the reference genome.
2) The region contains sequence that is sufficiently repetitive that it precludes unique mappings of Complete Genomics' paired-end reads to the reference genome.
3) Insufficient coverage over the region
4) Large variations overlapping the region.

The file contains a single row for each region and the format of the file is described in Table 6.

Column Name	Description
chromosome	Chromosome name in text: "1","2",...,"22","X","Y" The pseudoautosomal regions for the sex chromosomes X and Y are represented by their coordinates on chromosome X.
begin	reference coordinate specifying the start of the assembled block (see section "Sequence Coordinate System")
end	reference coordinate specifying the end of the assembled block (see section "Sequence Coordinate System")
ploidy	The ploidy of the block (=2 for diploid)
type	Indicates if the region contains a variation (value "variation") or is exactly the same as the reference genome (value "reference"). When the value of this field is "variation" one or more variations were found in this region. The variation(s) found in this region can be found in the accompanying file "***var-[ASM-ID].tsv***".

Table 6 Region file header description

A sample extract from a "regions" file is shown below.

```
>chromosome    begin   end     ploidy  type
1              960     972     2       reference
1              984     993     2       reference
1              1047    1059    2       variation
1              1085    1098    2       reference
```

Notes:

1) In the example above, the assembled sequence on chromosome 1 from position 960-972 is homozygous and identical to the reference genome

2) The region from positions 972-984 on chromosome 1 has not been called.

3) The region from 984-993 is, again, homozygous and identical to the reference genome.

4) The region from 1047-1059 contains one *or* more variations, the details of which are described in the file "***var-[ASM-ID].tsv***".

gene-[ASM-ID].tsv file

This tab-separated text file contains annotations on variations that fall within or near a RefSeq transcript. Each variation is annotated with its effect on the gene, such as frameshift, silent, nonsense mutations etc.

A description of the columns is provided in Table 7.

Column #	Column Name	Description
1	index	Identifier for this annotation
1	locus	Identifier for the locus. Identifier is the identifier from the Variations.csv file
2	haplotype	Identifier for each haplotype at the variation locus. For diploid chromosomes, 1 or 2.
3	chromosome	Chromosome name in text: "1","2",...,"22","X","Y" The pseudoautosomal Regions for the sex chromosomes X and Y are represented by their coordinates on chromosome X.
4	begin	Reference coordinates specifying the start of the variation (*not the locus).* Uses the half-open zero-based coordinate system. See section "Sequence Coordinate System" for more information.
5	end	Reference coordinates specifying the end of the variation (*not the locus).* Uses the half-open zero-based coordinate system. See section "Sequence Coordinate System" for more information.
6	variType	Type of variation, as reported in the Variations.csv file.
7	reference	The reference sequence at the locus of the variation. Empty when vartype is **ins**.
8	call	The observed sequence at the locus of the variation. Empty when variation is **del**. "**?**" is used to indicate 0 or more unknown bases within the sequence; "**N**" is used to indicate exactly one unknown base within the sequence.
9	xRef	Cross-reference to external identifier for variation. Currently populated for variations reported in dbSNP, release 129; indicated as, e.g. "dbSNP:rs12345".
9	geneId	EntrezGene identifier of the locus this variation falls in
10	mrnaAcc	Refseq mRNA accession number (versioned), e.g. NM_152486.2
11	proteinAcc	Refseq protein accession number (versioned), e.g. NP_689699.2
12	orientation	Orientation of the transcript with respect to the reference genome
15	exonCategory (category)	Category of region of the gene where this variation is located. Indicates the area of the locus this variation falls in. Can be "**EXON**", "**INTRON**", "**BEGIN**", "**END**" or "**UTR**". **BEGIN** or **END :** Indicates whether the variation falls inside the first two bases (DONOR) or last two bases (ACCEPTOR) of the intron
16	exon	Number indicating which exon or intron is affected by this variation (0-based, in order of chromosomal position of the exons)
17	codingRegion-Known	Indicates if a coding region is known for this transcript. Can be "**Y**" or "**N**"

18	aaCategory	Indicates the type of effect this variation has on the protein sequence. Currently empty or one of: **NO-CHANGE**: The sequence of this haplotype is identical to the canonical transcript sequence (which may or may not be identical to the reference sequence used in the assembly) **COMPATIBLE**: Synonymous. The DNA sequence for this transcript has changed, but there is no change in the protein sequence: the altered codon codes for the same amino acid **MISSENSE**: The DNA sequence for this transcript has changed and there is a change in the protein sequence as well, since the codon codes for a different amino acid. There is no change in size of the protein. **NONSENSE**: The DNA sequence for this transcript has changed and has resulted in a STOP codon (TGA, TAG or TAA), resulting in an early termination of the protein translation. **DELETE**: The DNA sequence for this transcript has changed and the length of the deletion is a multiple of 3, resulting in deletion of amino acids in the sequence in-frame, with no neighboring amino acids modified **INSERT**: The DNA sequence for this transcript has changed and the length of the insertion is a multiple of 3, resulting in the insertion of amino acids in the sequence in-frame, with no neighboring amino acids modified **DELETE+**: The DNA sequence for this transcript has changed and the length of the deletion is a multiple of 3, resulting in deletion of amino acids in the sequence in-frame, with one or two amino acids neighboring the deletion modified. **INSERT+**: The DNA sequence for this transcript has changed and the length of the insertion is a multiple of 3, resulting in the insertion of amino acids in the sequence in-frame, with one or two amino acids neighboring the insertion modified. **FRAMESHIFT**: The DNA sequence for this transcript has changed and has resulted in a frameshift for this protein. **NONSTOP**: The DNA sequence for this transcript has changed and has resulted in the change of a STOP codon (TGA, TAG or TAA) into a codon that codes for an amino acid, resulting in the continuation of the translation for this protein. **UNKNOWN**: Due to the fact that one or both alleles have no-calls (N or ?), it is not possible to determine the effect of the variation **UNDEFINED**: There is no known protein-coding region for the transcript
19	nucleotidePos	Start position of the variation in the mRNA. Counted from the start of the mRNA sequence (0 based)
20	proteinPos	Start position of the variation in the protein sequence. (0 based)
21	aaAnnot	Amino acid sequence for this allele ***before*** modification. Amino acid sequence is derived directly from the transcript sequence. It is **NOT** derived from the reference genome sequence used in the assembly since that may be different.
22	aaCall	Amino acid sequence for this allele ***after*** modification. Amino acid sequence is derived directly from the transcript sequence and modified. It is **NOT** derived from the reference genome sequence used in the assembly.
23	aaRef	Amino acid sequence for this allele ***before*** modification. This amino acid sequence **IS** derived from the reference genome sequence used in the assembly.

Table 7 Gene annotation file format description

dbSNPAnnotated-[ASM-ID].tsv

This file contains all dbSNP entries with fully-defined alleles (not unspecified large insertions and deletions) and the calls that were made for each of the locations in the genome being sequenced.

Column #	Column Name	Description
1	dbSnpId	Identifier for this dbSNP entry. Format is [DBNAME]:[ACC#]. DBNAME currently is "dbsnp" only and ACC# is the dbSNP identifier. (example: dbsnp:rs1167318)
2	alleles	Alleles for the dbSNP entry. (e.g. "C/T", "C/-", etc.)
3	chromosome	Chromosome name in text: "1","2",…,"22","X","Y" The pseudoautosomal regions for the sex chromosomes X and Y are represented by their coordinates on chromosome X.
4	begin	Reference coordinate specifying the start of the dbSNP entry. Uses the half-open zero-based coordinate system. See section "Sequence Coordinate System" for more information.
5	end	Reference coordinate specifying the end of the dbSNP entry. Uses the half-open zero-based coordinate system. See section "Sequence Coordinate System" for more information.
6	reference	The reference sequence at the locus of the variation.
7	found	Indicates whether the variation was located on the assembled genome.
8	locus	When the genome assembly resulted in a call different from the reference, then the locus ID from the variation file is given here, else blank. This field corresponds to column #1 of the variation file "var-[ASM-ID].csv"
9	zygosity	Indicates the zygosity of the call at this position. Can be "hom", "het" or empty, for homozygous, heterozygous and unknown respectively.
10	varType1	Indicates the type of variation at this location for the assembled genome for the first haplotype. Can be "=", "snp", "delins", "ins" and "del"
11	hap1	Sequence of the first haplotype
12	score1	Variation score of the first haplotype. (Empty in the case of a homozygous reference call)
13	varType2	Indicates the type of variation at this location for the assembled genome for the second haplotype. Can be "=", "snp", "delins", "ins" and "del"
14	hap2	Sequence of the second haplotype
15	score2	Variation score of the second haplotype. (Empty in the case of a homozygous reference call)
16	exactMatch	Indicates whether an exact match to the variation in dbSNP was detected. Partial matches are possible in the case of repeats, for instance, where the exact number of repeated copies in the database entry is not identical to the variation found. Value can be "Y" or "N"
17	locusContig	Chromosome name in text: "1","2",…,"22","X","Y" The pseudoautosomal Regions for the sex chromosomes X and Y are represented by their coordinates on chromosome X.
18	locusBegin	Reference coordinate specifying the start of the variation. Uses the half-open zero-based coordinate system. See section "Sequence Coordinate System" for more information. The pseudoautosomal Regions for the sex chromosomes X and Y are represented by their coordinates on chromosome X.
19	locusEnd	Reference coordinate specifying the end of the variation. Uses the half-open zero-based coordinate system. See section "Sequence Coordinate System" for more information. The pseudoautosomal Regions for the sex chromosomes X and Y are represented by their coordinates on chromosome X.

Table 8 Annotated dbSNP file format description

REF folder

The REF folder contains the coverage and reference score data for each base position of the reference genome. The data are split into several files, one corresponding to each chromosome. The coverage data represents the number of uniquely and fully mapped DNBs that overlap each base position – more precisely, it counts all full-DNB mappings that have a mapping weight ratio > 0.99 overlapping each position. The reference score is a measure of confidence that the base at that position is the same as the reference genome (homozygous reference). The reference score is computed based on an examination of several alternate hypotheses, including all heterozygous SNPs and some single-base insertions and deletions.

coverageRefScore-[chromosome-ID]-[ASM-ID].tsv.gz

The reference score and coverage files are organized by chromosome. The chromosome number is also represented in the header key "#CHROMOSOME".

[ASM-ID] in the file name is the assembly ID for this genome assembly.

The file consists of three columns as described in Table 9:

Column Name	Description
offset	0-based position within chromosome for the base
refScore	Reference score for the position. Positive values indicate greater confidence that the position is homozygous and identical to the reference genome.
coverage	Coverage of this position by unique, fully mapping reads (both arms map with expected order, orientation and separation, and the weight of this mapping indicates only one high-probability mapping)

Table 9 Coverage and reference score file format description

An example of a coverage/refScore file is provided below:

```
>offset   refScore   coverage
0         45         30
1         48         32
2         49         32
3         95         42
4         92         43
5         90         43
```

3.2.5 Sequence Coordinate System

Sequence positions in the mapping and variations files are represented in half-open, zero-based coordinates, which denote locations between successive reference base positions. A substitution or deletion of the second base (T) in the sequence of length 8 below would have a start position of 1 and an end position of 2. An insertion following the same second base would have both a start and end position of 2.

```
0 1 2 3 4 5 6 7 8
 A T A G G C T A
```

3.3 Example Scripts

Several example Python scripts are provided in the DOC/EXAMPLES folder. These demonstrate methods of extracting information from the data and transforming it to other formats. These Python scripts are described below.

3.3.1 cgiMapLib.py

This file provides a library of Python data structures that supports the processing of reads and mappings. It also includes data parsing routines. This library is used by all the other example scripts described below.

3.3.2 cgiMapReader.py

This Python script, executed from any location for a single lane within MAP/, processes the files ***reads.tsv.gz***, ***mappings.tsv.gz***, and ***lib_DNB_[LIBRARY-NAME].tsv***, associates the mappings in ***mappings.tsv.gz*** with corresponding reads and scores, and prints them out.

3.3.3 cgi2SAM.py

This Python script, executed for a single lane within MAP/, processes the files ***reads.tsv.gz***, ***mappings.tsv.gz***, and ***lib_DNB_[LIBRARY-NAME].tsv*** , associates the mappings in ***mappings.tsv.gz*** with corresponding reads and scores, and exports the data in the SAM format.

3.3.4 cgiGenomeDataTransformer.py

This is a high-level script that shows how a genome-scale conversion of mappings to an alternate format (based, e.g. on cgi2SAM.py) could be run in parallel. The specific example demonstrated is the conversion of all the reads, scores, and mappings for a genome to the binary BAM format, in conjunction with samtools for the conversion of the data from SAM to BAM. This script can be modified to accomplish other transformations of the data if desired.

3.4 Verifying the integrity of the files

For verification of the integrity of the manifest files, a certificate is required. This certificate can be downloaded from the Complete Genomics website. The certificate may be found under the “Resources” section on the main webpage or directly from the link below:

http://www.completegenomics.com/CGI-data-verification.pem

Alternatively, a certificate for verification of the integrity of the files may be obtained from VeriSign™ (https://digitalid.verisign.com/services/client/index.html) using the email address dev-support@completegenomics.com as the search criteria. From the download page, select the format that is compatible with your local verification system.

For verification purposes it may also be required that the CA certificates from VeriSign be available. These may be downloaded from:

https://knowledge.verisign.com/support/digital-id-support/index?page=content&id=S:SO6052&actp=search&searchid=1233261848732

Once the certificates are obtained, they may be used to verify the integrity of the manifest files with `openssl` or equivalent procedures. The manifest files are signed using S/MIME and the PKCS #7 secure message format specification.

Index

N

O

P

R

S

T

V

© Copyright 2009. Complete Genomics, Inc. All rights reserved. cPAL and DNB are trademarks of Complete Genomics, Inc. in the US and certain other countries. All other trademarks are the property of their respective owners.