HapCompass Manual

Derek Aguiar
Brown University

June 13, 2014

Contents

1 Introduction

HapCompass is a software package for haplotype assembly of diploid, polyploid, and tumor genomes[1, 2, 3].

1.1 Description

HapCompass is a graph-theoretic algorithm for genome-wide haplotype assembly. HapCompass-Polyploidy and HapCompass-Tumor are the first models and algorithms to extend haplotype assembly beyond diploid genomes.

1.2 Download

HapCompass can be downloaded from `http://www.brown.edu/Research/Istrail_Lab/hapcompass.php`.

2 HapCompass options

2.1 Usage

usage: java [JVM options] -jar hapcompass.jar [OPTIONS]...

Example usage

java -Xmx1g -jar hapcompass.jar –bam example/example.bam –vcf example/62_ID0.txt -o example/out

2.2 Arguments

The parameters to HAPCOMPASS are given in `-<parameter> <value>` or `--<parameter>=<value>` pairs if there are arguments or `-<parameter>` or `--<parameter>` for boolean switches.

2.2.1 Required

```
-b,--bam <arg>          The BAM or SAM file to process. The
                        preferred input is a BAM file that is coordinate
                        sorted. If you require name sorted input files, the
                        --namesort option may be used in conjunction
                        with aname sorted BAM or SAM file. If the file is
                        name sorted, mated reads must has the same name.
-o,--output <arg>       Prefix to write output files to. This should
                        include a reference to the directory and a prefix
                        for files, for example: /home/me/myOutput_prefix
 -v,--vcf <arg>         The VCF file VCFFILE giving the variants. The VCF
                        file should follow the VCF 4.1 standard, that
                        is,it should have 8 tab delimited columns: #CHROM
                        POS ID REF ALT QUAL FILTER INFO followed by a
                        FORMAT column and 1 column representing the
                        individual's genotypes.
```

2.2.2 Optional

```
 -c,--cache              Enable caching of chunks of the input BAM file.
                         This option is off by default but can be used if
                         you are experiencing problems with the default
                         BAM loading and the BAM file is very large with
                         many very long insert size reads.
 -d,--debug              Turn on debugging. This option will produce an
                         output log file.
 -e,--export             Don't run the algorithm, just export data files.
 -f,--fragment <arg>     Set input fragment filename.
 -g,--downsample <arg>   Downsample an input BAM file.
 -i,--prim               Turn on the Prim-like algorithm for resolution of
                         the general chain graph for polyploid genomes.
 -j,--diverse            Force HapCompass to prefer edges in G_C that
                         include more unique haplotypes. This option
                         sometimes works well when the coverage is low and
                         the sequence read length is long.
 -l,--suffix <arg>       Defines a suffix to trim off read names. This was
                         added as a convinience for processing some files
                         produced by sff_extract.
 -n,--namesort           HapCompass will assume the input SAM/BAM files
                         are sorted by name
```

```
-p,--ploidy <arg>       Set the ploidy of the genome to be assembled.
                        Value must be 2 or higher. Any number above 2
                        will trigger usage of the polyploid algorithm.
                        Default is 2
-r,--reference <arg>    HapCompass will only produce results from this
                        reference name. This must match a VCF and SAM/BAM
                        reference name.
-s,--simulate <arg>     HapCompass will simulate reads with properties
                        given by the string:<number of reads>:<read
                        length>:<single base error substitution
                        probability>:<smallest base to sample>:<largest
                        base to sample>:<list of insert sizes seperated
                        by commas>:<list of insert sizes seperated by
                        commas>:<reference name to simulate reads
                        for>:<only produce good reads>Example:
                        1000:200:0.01:1:14000:3500,2500:1150,1250:chr1:tr
                        ue will simulate 1000 paired reads with 200 bases
                        for each read of the pairs, 1% error rate, in the
                        region of [1,14000], with insert sizes 3500 and
                        2500 and insert size standard deviations 1150 and
                        1250 respectively for reference chr1 and only
                        produce reads with 2 or more variants
-x,--evaluate <arg>     Provide a phasing filename which was output from
                        HapCompass for evaluation.
-y,--ilp                Output ILP file.
-z,--iterations <arg>   Number of iterations to run. The larger the value
                        the more accuracte the assembly. Default is 10
                        for ploidy = 2, and 1 for ploidy > 2.
```

2.2.3 HapCompass output

Several files will be output to the -o directory.

- `OUTPUT_PREFIX_<ALGORITHM>_solution.txt`

  Contains a list of blocks and phasings. A new block is shown by `BLOCK <start_position_of_block> <end_position> <start_snp_number> <end_snp_number> <score>` then a list of SNPs in the block...

```
<snp_id>    <snp_position>    <snp_number>    <hap_0_allele>    <hap_1_allele>
```

In the case of assembling more than 2 haplotypes, there will be additional columns for the extra haplotypes.

- `OUTPUT_PREFIX_reduced_representation.vcf` and `OUTPUT_PREFIX_reduced_representation.sam`

  A list of SAM reads and variants covering the blocks of variants in terms of the variant numbers instead of variant positions.

- `OUTPUT_PREFIX_reads.sam`

  A different representation of input reads in terms of true genomic positions.

- `OUTPUT_PREFIX_binning.txt`

  This file will only be produced if `-ploidy > 2`. Contains the pairwise variant phasings defined in the compass graph G_C.

- `OUTPUT_PREFIX_frags.txt`

  BAM input files are converted into an alternative representation that contains only the information required by HapCompass. This file will only contain the sequence reads that contain 2 or more variant alleles.

3 File formats

The HapCompass package includes sample files for all types of input.

3.1 Variants

HapCompass always requires specification of variants in the form of a VCF file. VCF specification can be found `https://github.com/amarcket/vcf_spec`.

3.2 Aligned sequence reads

There are two options for inputting sequence reads: BAM and fragment files.

A SAM file is a tab delimited file for specifying sequence reads and their properties. A BAM file is the SAM files binary version. BAM and SAM files may be manipulated using SAMTools. The BAM specification can be found `http://samtools.sourceforge.net/SAMv1.pdf`.

Fragment files are a **tab** delimited file format with reads separated by new lines. The columns are specified as follows:

column 1: number of read parts, e.g. 1 for a single read, 2 for two reads (paired sequencing when both ends are informative)

column 2: read name

column 3 and on: the read parts. The number of read parts is defined in column 1.

A read part is a **space** delimited list beginning with a reference name and then a set of one or more position(s), allele(s), base quality score(s) trios. The position is the 0-based position of the variant in the VCF file. E.g. the first variant is position 0, the second variant is position 1, and so on. The allele is the numerically encoded allele, 0 for reference, 1 for first alternative allele, and so on. The base quality score is the ASCII of Phred-scaled base quality+33. An example of a paired read is given below `<tab>` denotes a tab character:

```
2<tab>sequence_read_1<tab>chr1 0 1 D 1 1 D<tab>chr1 4 0 D 5 0 D 6 0 D
```

4 Caveats and assumptions

HapCompass, and haplotype assembly algorithms in general, requires a set of aligned sequence reads and variants calls. The polyploid algorithm requires knowledge of the number of homologous copies.

5 More information

5.1 What is haplotype assembly?

Standard genome sequencing workflows produce contiguous DNA segments of an unknown chromosomal origin. *De-novo* assemblies for genomes with two sets of chromosomes (*diploid*) or more (*polyploid*) produce consensus sequences in which the relative haplotype phase between variants is undetermined. The set of sequencing reads can be mapped to the phase-ambiguous reference genome and the diploid chromosome origin can be determined but, without knowledge of the haplotype sequences, reads cannot be mapped to the particular haploid chromosome sequence. However, sequence reads are derived from a single haploid fragment and thus provide valuable phase information when they contain two or more variants. The *haplotype assembly problem* aims to compute the haplotype sequences for each chromosome given a set of aligned sequence reads to the genome and variant information. The haplotype phase of variants is inferred from assembling overlapping sequence reads.

5.2 Author

Derek Aguiar earned his PhD from the Department of Computer Science at Brown University working under the direction of Sorin Istrail.

5.3 Citations

1. HapCompass for diploid genomes[1].
2. HapCompass for tumor and polyploid genomes[2, 3].

References

[1] Derek Aguiar and Sorin Istrail, *HAPCOMPASS: A fast cycle basis algorithm for accurate haplotype assembly of sequence data.* J. Comput. Biol., **19**, 2013

[2] Derek Aguiar and Sorin Istrail, *Haplotype assembly in polyploid genomes and identical by descent shared tracts.* Bioinformatics, Pacific Symposium of Biocomputing, **29**:352-360, 2013

[3] Derek Aguiar, Wendy S.W. Wong and Sorin Istrail, *Tumor haplotype assembly algorithms for cancer genomics.* Pacific Symposium of Biocomputing, 2014