HapFlow Manual

Authors: Mitchell J. Sullivan, Nathan L. Bachmann, Peter Timms and Adam Polkinghorne

For support please contact Mitchell Sullivan at: mjsull@gmail.com

Prepared by: Mitchell Sullivan and Nathan Bachmann

March, 2015

USER'S MANUAL

TABLE OF CONTENTS

1.0   GENERAL INFORMATION

1.0 GENERAL INFORMATION

1.1 Overview

HapFlow is a python application for visualising haplotypes present in sequencing data. It identifies variant profiles present and reads and creates an abstract visual representation of these profiles to make haplotypes easier to identify.

1.2 Description

HapFlow is a python application for visualising haplotypes present in sequencing data. It identifies variant profiles present and reads and creates an abstract visual representation of these profiles to make haplotypes easier to identify.

1.3 Acronyms and Abbreviations

SAM – Sequence Alignment Map (format)
BAM – Binary Alignment Map

1.4 Citing HapFlow

Please cite:
Sullivan MJ, Bachmann NL, Timms P & Polkinghorne A.
HapFlow: Visualising haplotypes in sequencing data
http://mjsull.github.io/HapFlow

2.0 Installation

2.0 INSTALLATION

2.1 OSX

1) Make sure PIP is installed by typing “pip” in Terminal.
If PIP is not installed on your system, it can be installed by entering the following command in Terminal.

```
% sudo easy_install pip
```

2) Install pysam. This is a required dependency for HapFlow. More information about pysam is available from its GitHub page (https://github.com/pysam-developers/pysam).

```
% sudo pip install pysam
```

3) Install canvasvg. This is a required dependency for HapFlow.

```
% sudo pip install canvasvg
```

4) HapFlow does not require installation and can run from any directory. Hapflow can be downloaded from its GitHub page.

```
% python Hapflow.py
```

2.2 Linux

1) Make sure PIP is installed by typing “pip” in Terminal.
If PIP is not installed on your system, it can be installed by entering the following command in Terminal.

```
% sudo easy_install pip
```

2) Install pysam. This is a required dependency for HapFlow. More information about pysam is available from its GitHub page (https://github.com/pysam-developers/pysam).

```
% sudo pip install pysam
```

3) Install canvasvg. This is a required dependency for HapFlow.

```
% sudo pip install canvasvg
```

4) HapFlow does not require installation and can run from any directory. Hapflow can be downloaded from its GitHub page.

```
% python Hapflow.py
```

3.0   GETTING STARTED

3.0 GETTING STARTED

3.1 Workflow

Hapflow uses a graphical user interface (GUI) for inputting and outputting files. The input files that Hapflow requires both a BAM file and VCF file.

This section describes the workflow for using Hapflow with paired-end Illumina reads by creating a sorted BAM file and VCF file.

1) The reads first need to be aligned against a reference genome to create a sorted BAM file. This can done using the read mapping BWA to create a SAM file:

```
% bwa index reference.fasta
% bwa aln read1.fastq > read1.sai
% bwa aln read2.fastq > read2.sai
% bwa sampe reference.fasta read1.sai read2.sai read1.fastq read2.fastq > aln.sam
```

Samtools can be used convert the SAM file to a sorted BAM file:

```
% samtools faidx reference.fasta
% samtools import reference.fasta.fai aln.sam aln.bam
% samtools sort aln.bam aln.sorted
```

2) Freebayes is the recommended software for creating a VCF file to be used with Haplflow

```
% freebayes –f reference.fasta aln.sorted.bam > aln.vcf
```

3) Launch Hapflow from the UNIX command line

```
% python Hapflow.py
```

4) Create the flow file using the sorted BAM and VCF file.

File -> Create flow file

5) Load the flow file.

File -> Load flow file

4.0 COMMAND OVERVIEW

4.0 COMMAND OVERVIEW

4.1 File Menu

The items in this menu deal with creating and loading flow files.

4.1.1 Create Flow File

Selecting “create flow file” will open a menu where the sorted BAM file and VCF file can be selected as well as designating a location to save the Flow file.

4.1.2 Load Flow file

Loads a flow file and visualizes the haplotype profile in the main window.

4.2 TOOLS MENU

4.2.1 Goto Base

Go to a user selected base position in the haplotype profile.

4.2.2 Create Image

Print out the contents of the current window as Scalar Vector Graphics (SVG) formatted image.

4.3 View Menu

4.3.1 Hide gapped

Hides the lines that represent gaps in the reads

4.3.2 Show gapped

Shows the lines that represent gaps in the reads

4.3.3 Stretch X

Extends x-axis of the display window.

4.3.4 Shrink X

Shrinks the x-axis of the display window.

4.3.5 Stretch Y

Extends y-axis of the display window.

4.3.6 Shrink Y

Shrinks the y-axis of the display window.

4.4 HELP

4.4.1 About

Provides version details about HapFlow.

4.4.1 Help

Links to the HapFlow Github page.

4.4.2 Citing HapFlow

Displays the citation details for HapFlow.

4.5 CONTEXT MENUS

Right clicking on a flow will bring up a variety of options.

4.5.1 Details

Provides the position in the alignment of the highlighted variant, the direction and number of the reads.

4.5.2 Get flow readnames

Outputs the name of the reads in the selected flow as a textfile.

4.5.3 Write flow to BAM

Outputs the read of the selected flow as a BAM file.

4.5.4 Get group readnames

Outputs the name of the reads in the selected group as a textfile.

4.5.5 Write group to BAM

Outputs the read of the selected group as a BAM file.

5.0 Tutorial

5.0 TUTORIALS

5.1 TUTORIAL 1: Simulated data from 3 strains

This tutorial will cover the approach for generating BAM and VCF for read data containing three *Chlamydia pecorum* strains and visualizing the haplotypes with HapFlow. In the tutorial folder are two FASTQ files containing simulated paired-end Illumina reads based on the complete genome of *C. pecorum* E58 at 20x coverage, *C. pecorum* PV3056 at 10x coverage and *C. pecorum* PV787 at 5x coverage mixed together to represent a mixed infection. The FASTQ files are called "mixed_3strains_R1.fastq" and "mixed_3strains_R2.fastq". Also included in the tutorial folder is a FASTA file of the complete genome *C. pecorum* W73, which will be used as a reference genome.

1) The BAM file can be created by aligning the simulated Illumina reads against the reference genome, *C. pecorum* W37 using BWA and Samtools.

```
% bwa index Cpecorum_W37.fasta
% bwa aln Cpecorum_W73.fasta mixed_3strains_R1.fastq > read1.sai
% bwa aln Cpecorum_W73.fasta mixed_3strains_R2.fastq > read2.sai
% bwa sampe Cpecorum_W73.fasta read1.sai read2.sai mixed_3strains_R1.fastq
mixed_3strains_R2.fastq > mixed_3strains_bwa.sam

% samtools faidx Cpecorum_W73.fasta
% samtools import Cpecorum_W73.fasta mixed_3strains_bwa.sam mixed_3strains_bwa.bam
% samtools sort mixed_3strains_bwa.bam mixed_3strains_bwa.sorted
% samtools index mixed_3strains_bwa.sorted.bam
```

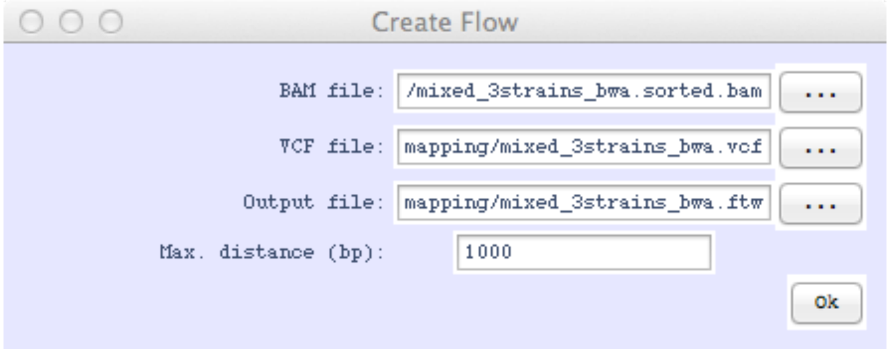
2) The VCF file can be created using Freebayes

```
% freebayes -f Cpecorum_W73.fasta mixed_3strains_bwa.sorted.bam > mixed_3strains_bwa.vcf
```

3) Launch HapFow.

```
% python HapFow.py
```

4) In the top menu bar, select File -> Create Flow File and load "mixed_3strains_bwa.sorted.bam" in the BAM file box and "mixed_3strains_bwa.vcf" for the VCF file box. Save the output as "mixed_3strains_bwa.ftw." This step may take a few minutes.



BAM file: /mixed_3strains_bwa.sorted.bam
VCF file: mapping/mixed_3strains_bwa.vcf
Output file: mapping/mixed_3strains_bwa.ftw
Max. distance (bp): 1000

Figure 1: The Create Flow menu is where the BAM and VCF file make the flow file.

5) Select File -> Load Flow File and load “mixed_3strains_bwa.ftw”. The following screen will appear in a few seconds. The x-axis can be extended by selecting View -> Stretch X.

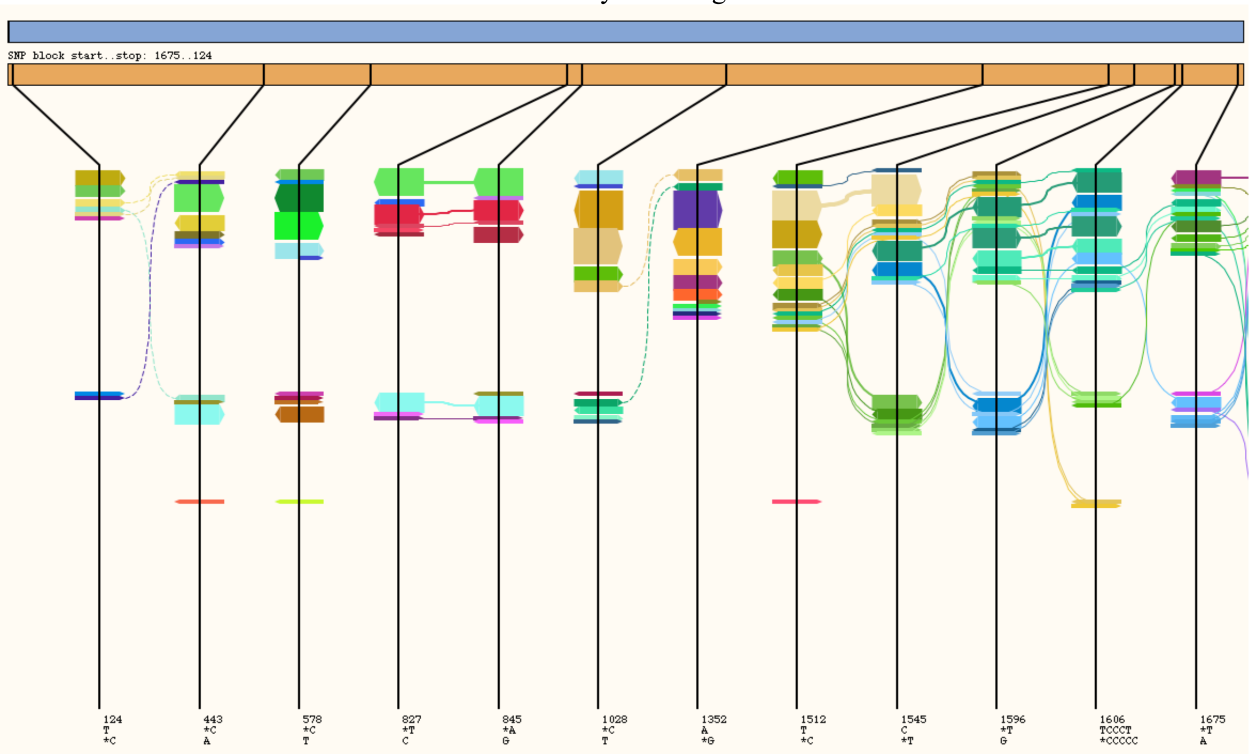


Figure 2: Hapflow diagram of simulated Illumina read data.

6) The orange rectangle with vertical lines represents where the variants are located within the displayed section of the genome, these lines are extended below and spaced an equal distance apart in the area where the flows are viewed. Each flow consisting of one or more reads is represented as one or more arrows overlapping each variant line that the reads of the flow align to. Width of the arrow represents the number of reads within that flow. A solid line joins variants on the same read of a pair.

7) Click on the arrows to highlight individual flows. Figure 3 show a section of the HapFlow profile where the three strains are visualised as flows. Figure 3A-C shows the flows containing a different combination of three single nucleotide polymorphisms (SNPs). Figure 3A represent *C. pecorum* E58 the most dominate strain (20x coverage) in the read data; hence why this flow has the thickest arrows of the three strains. Figure 3B show a flow associated with *C. pecorum* PV3056 the second most prevalent strain (10x coverage). Figure 3C marks the flow for *C. pecorum* P787, which is the least prevalent strain (5x coverage). Right click on the flow will display the options to retrieve the read names for flow or extract the reads as a BAM file.

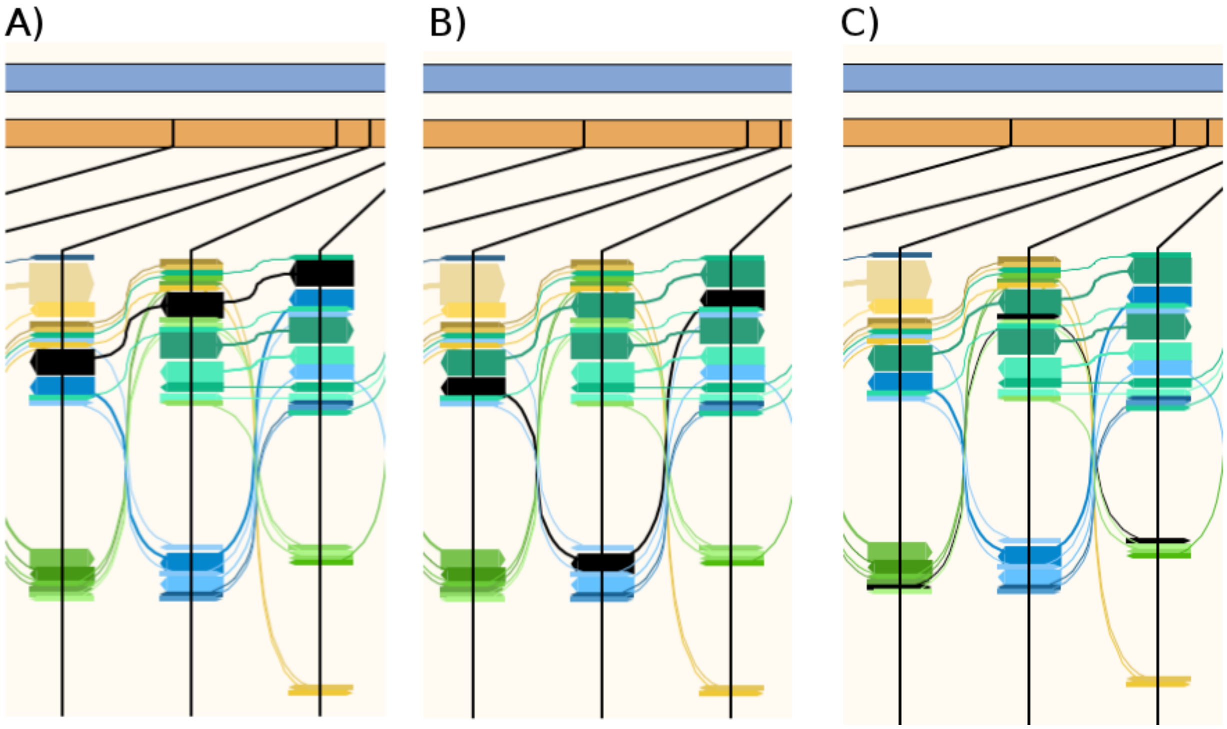


Figure 3: A section of Hapflow profile showing three SNPs (the black vertical lines in orange bar). Panel A-C shows three separate haplotypes (black arrows) representing the three *C. pecorum* strains. A) *C. pecorum* E58 flow. B) *C. pecorum* PV3056 flow. C) *C. pecorum* P787 flow.