HapFlow: Visualising haplotypes in sequencing data

Mitchell J. Sullivan, Nathan L. Bachmann, Peter Timms, Adam Polkinghorne*

Faculty of Science, Health, Education and Engineering, University of the Sunshine Coast,
Sippy Downs 4556, Australia

Running Head: HapFlow: Visualising haplotypes

*To whom correspondence should be addressed. Tel: +61 7 5456 5578; Email: apolking@usc.edu.au

Abstract

Summary: HapFlow is a python application for visualising haplotypes present in high-throughput sequencing data. HapFlow identifies nucleotide variant profiles in raw read sequences and creates an abstract visual representation of these profiles to make haplotypes easier to identify.

Availablity: HapFlow is freely available (under a GPL license) for download (for Mac OS X, Unix and Microsoft Windows) from github (http://mjsull.github.io/HapFlow).

Contact: apolking@usc.edu.au

PeerJ PrePrints | https://dx.doi.org/10.7287/peerj.preprints.895v1 | CC-BY 4.0 Open Access | rec: 14 Mar 2015, publ: 14 Mar 2015

Introduction

The emergence of high-throughput sequencing has enabled new experimental approaches such as the sequencing of bacterial populations. Infections frequently contain multiple strains of the same species (Darch, et al., 2015; Taylor, et al., 1995). This has important implications for detecting transmission events (Bachmann, et al., 2015) and determining treatment outcomes (Cohen, et al., 2012). Several methods have been developed to analyse mixed-strain populations. ShoRAH (Zagordi, et al., 2011) reconstructs a minimal set of global haplotypes and estimates the frequency of inferred haplotypes. It requires variants be dense enough to be linked by overlapping reads. A two-step maximum likelihood approach has also been described to identify the portion of infection rising from dominant and minor strains (Eyre, et al., 2013). This approach does not rely on variant density but is unable to infer local or global haplotypes. A tool that visualises haplotypes in sequencing data is needed to identify the best strategy for genomic analysis of multiple strains of the same bacteria within a sample.

Many excellent read alignment visualisation tools exist including Savant (Fiume, et al., 2010), Tablet (Milne, et al., 2010) and Consed (Gordon and Green, 2013). These tools arrange reads in a linear fashion with each read represented as a line, or row of bases. This layout is satisfactory for identifying variants or misaligned reads, however, it is not ideal for identifying haplotypes present in reads. Reads are packed tightly together making it difficult to determine whether distant variants are located on the same read pair. Additionally, reads are not grouped by haplotype making it difficult to identify how frequently a haplotype is represented in the sequencing data.

HapFlow addresses these problem by abstracting read alignment data to make the haplotypes present easier to identify. HapFlow can be used to help identify potential sites of recombination, identify the minimum number of strains present in a sample and determine whether defining local or global haplotypes is possible using sequence data alone.

Implementation

HapFlow is a python tool that uses the Tkinter windows system. It is available as a Python script or using the package manager PIP. It contains two parts: HapFlow-generator, a process for creating a flow file, which contains the count of reads with each haplotype profile and HapFlow-viewer, a tool for visualising the flow file.

HapFlow-generator can be executed from the GUI or the command-line. It takes a VCF file of called variants and an indexed BAM file of aligned reads as input. Pysam is used to create a profile of variants present in each read of the alignment. This profile consists of which variant or variants are present in the read, on which pair each variant is present and the direction of the read. If the variant profile is unique, a flow (profile of variants in a read) is created. If the flow already exists in another read, the count of the flow is incremented by one.

HapFlow-viewer displays the created flow file on the canvas of the GUI. An orange rectangle within the blue rectangle represents the portion of the genome currently being displayed. Underneath, an 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. Variants on the same read of a pair are joined by a solid line, variants on different reads of a pair are joined by a dotted line. Arrows grouped at the top of the canvas represent the most common variant, the second group of flows represents the second most common variant and so on. The last group represents potential sequencing, alignment or variant calling errors as they have sequence that has not been called as a variant. Information about the sequence of each variant is represented underneath the flows. The canvas is scrollable and zoomable allowing the user to easily navigate through whole genomes.

Results and discussion

To demonstrate the application of HapFlow, reads from the recent sequencing of a *Chlamydia pecorum* PCR-positive swab sample collected from the urogenital tract of a koala with mixed *C. pecorum* infections were analysed. *C. pecorum* DNA was extracted directly from the host cell contaminants using Sure-Select RNA probes and sequenced using an Illumina Hi-Seq to produce 101bp paired-end reads, as previously described (Bachmann, et al., 2015). These reads were then mapped back to E58 using Bowtie-2 and then variant calling was performed using FreeBayes. Exploration of the HapFlow diagram identified several regions in low complexity areas where non-chlamydial DNA had been captured. Importantly, several regions where read coverage in the dominant strain dropped below that of the minor strain were identified (Figure 1). This was not unexpected as sequence capture is less efficient at capturing DNA in areas where the sequence of the strain varies significantly from the probe. This meant that any method of consensus calling that relied on coverage would result in a chimeric genome not representative of either strain. Due to the proximity of variants, a linkage approach was used to determine the sequence for large regions of both strains.

Funding Information:
This work was supported by an Australian Research Council Discovery Grant (DP130102066) and a Queensland Department of Environment and Heritage Protection

Koala Research Grant (KRG18) awarded to AP and PT.

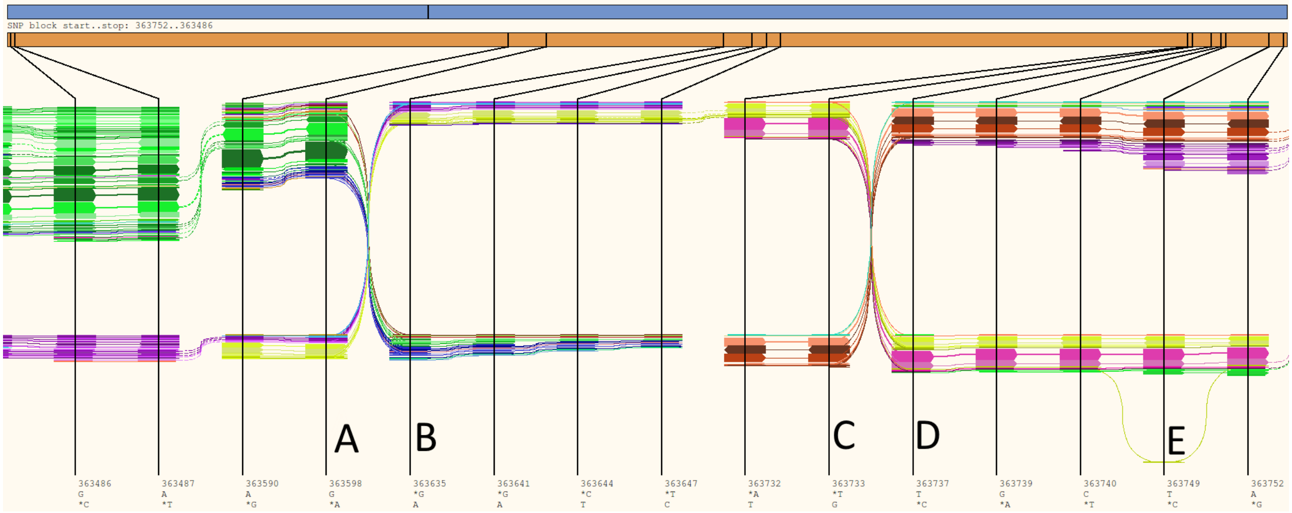

Figure 1: HapFlow diagram of sequencing data from a urogenital tract infection in a koala mapped to *C. pecorum*. Flows containing only the dominant variant group at the top, flows containing only the minor variant group in the second row while mixed flows switch between top and middle. A flow containing an alignment or sequencing error can be seen at site E. All reads with the most common variant at site A have the least common variant at site B. Similarly all reads with the least common variant at site A, have the most common variant at site B. This pattern is repeated at sites C and D.

References

Bachmann, N.L., *et al.* (2015) Culture-independent genome sequencing of clinical samples reveals an unexpected heterogeneity of infections by *Chlamydia pecorum*. *J Clin Microbiol* **5,** 1-9.
Cohen, T., *et al.* (2012) Mixed-strain mycobacterium tuberculosis infections and the implications for tuberculosis treatment and control. *Clin Microbiol Rev* **25,** 708-719.
Darch, S.E., *et al.* (2015) Recombination is a key driver of genomic and phenotypic diversity in a *Pseudomonas aeruginosa* population during cystic fibrosis infection. *Sci. Rep* **5,** 7649.
Eyre, D.W., *et al.* (2013) Detection of mixed infection from bacterial whole genome sequence data allows assessment of its role in *Clostridium difficile* transmission. *PLoS Comput Biol* **9,** e1003059.
Fiume, M., *et al.* (2010) Savant: genome browser for high-throughput sequencing data. *Bioinformatics* **26,** 1938-1944.
Gordon, D. and Green, P. (2013) Consed: a graphical editor for next-generation sequencing.

Bioinformatics **29,** 2936-2937.

Milne, I., *et al.* (2010) Tablet—next generation sequence assembly visualization. *Bioinformatics* **26,** 401-402.

Taylor, N.S., *et al.* (1995) Long-term colonization with single and multiple strains of Helicobacter pylori assessed by DNA fingerprinting. *J Clin Microbiol* **33,** 918-923.

Zagordi, O., *et al.* (2011) ShoRAH: estimating the genetic diversity of a mixed sample from next-generation sequencing data. *BMC Bioinformatics* **12,** 119.