Bioinformatics, 32(3), 2016, 441–443
doi: 10.1093/bioinformatics/btv551
Advance Access Publication Date: 7 October 2015
Applications Note

Genome analysis

HapFlow: visualizing haplotypes in sequencing data

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

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

*To whom correspondence should be addressed.

Associate Editor: John Hancock

Received on May 31, 2015; revised on August 15, 2015; accepted on September 16, 2015

Abstract

Summary: HapFlow is a python application for visualizing 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.
Availability and implementation: 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

1 Introduction

The emergence of high-throughput sequencing has enabled new experimental approaches such as the sequencing of populations of bacteria. 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 analyze 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 visualizes 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 visualization 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, 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 sequencing artifacts, identify the minimum number of strains present in a sample and determine whether defining local or global haplotypes is possible using sequence data alone.

2 Implementation

HapFlow is a Python tool that uses the Tkinter windows system (https://wiki.python.org/moin/TkInter). It has been designed for bacterial sequencing datasets of all sizes that use short, paired-end or single-end reads. HapFlow is available as both a Python script and as a binary for Linux, OSX and Windows. 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 visualizing the flow file.

HapFlow-generator can be executed from the GUI or the command-line. It takes a VCF file (Danecek *et al.*, 2011) of called variants and an indexed BAM file (Li *et al.*, 2009) of aligned reads as input. Pysam (github.com/pysam-developers/pysam) is used to create a profile of alleles present in each read of the alignment. This profile consists of which allele is present at each variant the read aligns to, on which pair each allele is present and the direction of the read. A list of 'flows' is then written to the flow

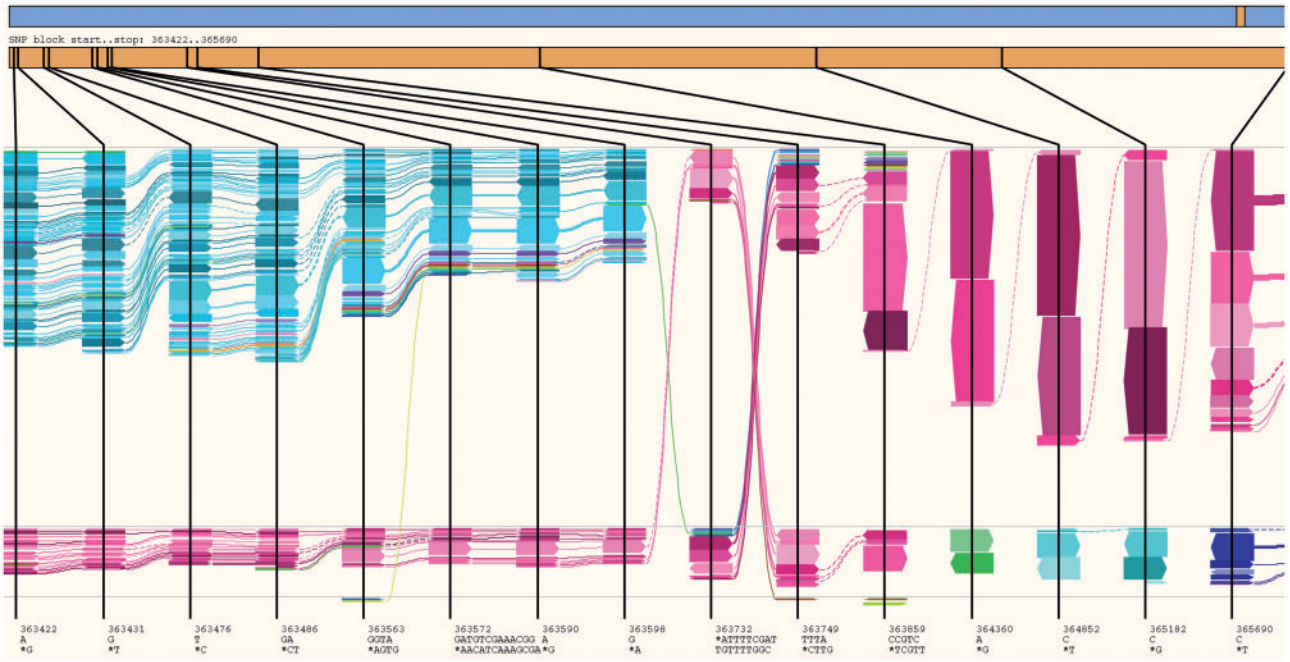

Fig. 1. HapFlow diagram of sequencing data from a urogenital tract infection in a Koala mapped to *C. pecorum*. Flows containing only the dominant variant group at the top, flows containing only the minor variant group at the in the second row, mixed flows switch between top and middle. The bottom row indicates flows containing an alignment or sequencing error. All reads with the most common variant at base 363,598 have the least common variant at base 363,732, similarly all reads with the least common variant at site base 363,598 have the most common variant at base 363,732. This pattern is repeated at sites C and D

file, where each 'flow' is a unique allele profile and a count of the reads with that profile. Flows are then assigned to groups. If all overlapping alleles of a flow are in consensus with a previously defined group, it is assigned to that group. Otherwise, it is assigned a new group number. These groups are used to colour flows when visualized. When multiple chromosomes are present in a BAM file, the user is prompted to select which chromosome to create a flow file for. A complete description of the flow file is included in the manual.

HapFlow-viewer displays the created flow file on the canvas of the GUI. A blue bar at the top of the screen represents the genome, an orange rectangle within the blue bar represents the portion of the genome currently being displayed. Underneath, an orange bar 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. At each variant, each flow covering the given variant is represented by an arrow, with the width of the arrow representing the number of reads the flow contains. The arrows representing each flow at adjacent variants are joined by lines, with solid lines representing variants covered by the same read of the flow and dashed lines representing variants covered by different reads in the pair. Arrows grouped at the top of the canvas represent the most frequently observed allele, the second group of flows represents the second most common allele and so on. The last group represents alleles not identified in the VCF file. This can be caused by potential sequencing, alignment or variant calling errors. All flows in a group are coloured using the same hue; saturation and lightness are randomly assigned. 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. Reads can then be separated into individual BAM files by selecting individual flows or group of flows, or by selecting all reads that conform to a user defined haplotype profile.

3 Case study

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 analyzed. *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 101 bp paired-end reads, as previously described (Bachmann *et al.*, 2015). These reads were then mapped back to a publicly available *C. pecorum* reference (E58) strain's genome using Bowtie 2 (Langmead and Salzberg, 2012). Variant calling was performed using FreeBayes (Garrison and Marth, 2012) and the resulting data was visualized using HapFlow. Two distinct strains were immediately identifiable in the HapFlow diagram. Variants are also close enough that local haplotypes can be inferred from read data. Importantly, several regions where read coverage in the dominant strain dropped below that of the minor strain were identified (Fig. 1). This was not unexpected as sequence capture is less efficient at cazpturing DNA in areas where the sequence of the strain varies significantly from the probe. This meant that any method of consensus calling that uses a majority call would result in a chimeric genome not representative of either strain. These insights can help with the design of an automated approach for assigning reads to strains. Alternatively, reads can be manually separated using HapFlow.

Funding

This work was financially 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.

Conflict of Interest: none declared.

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.*, **53**, 1573–1581.

Cohen,T. *et al*. (2012) Mixed-strain mycobacterium tuberculosis infections and the implications for tuberculosis treatment and control. *Clin. Microb. Rev.*, **25**, 708–719.

Danecek,P. *et al*. (2011) The variant call format and VCFtools. *Bioinformatics*, **27**, 2156–2158.

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**, 7549.

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.

Garrison,E. and Marth,G. (2012) Haplotype-based variant detection from short read sequencing. *arXiv preprint*, arXiv:1207.3907.

Gordon,D. and Green,P. (2013) Consed: a graphical editor for next-generation sequencing. *Bioinformatics*, **29**, 2936–2937.

Li,H. *et al*. (2009) The Sequence Alignment/Map format and SAMtools. *Bioinformatics*, **25**, 2078–2079.

Langmead,B. and Salzberg,S.L. (2012) Fast gapped-read alignment with Bowtie 2. *Nat. Methods*, **9**, 357–359.

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.