Bioinformatics, 35(15), 2019, 2665–2667
doi: 10.1093/bioinformatics/bty1003
Advance Access Publication Date: 18 December 2018
Applications Note

Sequence analysis

Pathogen–Host Analysis Tool (PHAT): an integrative platform to analyze next-generation sequencing data

Christopher M. Gibb[1,2,*,†], Robert Jackson[1,3,*,†], Sabah Mohammed[2], Jinan Fiaidhi[2] and Ingeborg Zehbe[1,4,5]

[1]Probe Development and Biomarker Exploration, Thunder Bay Regional Health Research Institute, [2]Department of Computer Science, [3]Biotechnology Program, [4]Department of Biology and [5]Northern Ontario School of Medicine, Lakehead University, Thunder Bay, Ontario, Canada

*To whom correspondence should be addressed.

Associate Editor: Bonnie Berger

[†]The authors wish it to be known that, in their opinion, the first two authors should be regarded as Joint First Authors.

Received on April 4, 2018; revised on November 28, 2018; editorial decision on December 3, 2018; accepted on December 12, 2018

Abstract

Summary: The Pathogen–Host Analysis Tool (PHAT) is an application for processing and analyzing next-generation sequencing (NGS) data as it relates to relationships between pathogens and their hosts. Unlike custom scripts and tedious pipeline programming, PHAT provides an integrative platform encompassing raw and aligned sequence and reference file input, quality control (QC) reporting, alignment and variant calling, linear and circular alignment viewing, and graphical and tabular output. This novel tool aims to be user-friendly for life scientists studying diverse pathogen–host relationships.

Availability and implementation: The project is available on GitHub (https://github.com/chgibb/PHAT) and includes convenient installers, as well as portable and source versions, for both Windows and Linux (Debian and RedHat). Up-to-date documentation for PHAT, including user guides and development notes, can be found at https://chgibb.github.io/PHATDocs/. We encourage users and developers to provide feedback (error reporting, suggestions and comments).

Contact: chris.gibb@outlook.com

1 Introduction

Analysis of pathogen data, especially of their genomes (Xiang *et al.*, 2007) via high-throughput or next-generation sequencing (NGS), is an essential endeavour to understanding intricate pathogen–host relationships. While the ease of producing NGS data has grown significantly, bottlenecks still exist in its processing and analysis. In particular, short-read alignment algorithms and the tools that implement them have matured to the point that they no longer represent the major hurdle in the data analysis process (Li and Homer, 2010). Instead, the availability of fast and user-friendly tools has become the limiting factor (Milne *et al.*, 2010). While there are excellent tools which perform one or several discrete functions in the same domain, e.g. Bowtie2 (Langmead and Salzberg, 2012) and SAMtools (Li *et al.*, 2009), all-in-one type platforms can offer a breadth of features that help address barrier-to-entry (i.e. the ease in which users can setup and perform analyses). Integrative multi-tool platforms such as Comparative Genomics (CoGe) (Lyons and Freeling, 2008), VirBase (Li *et al.*, 2014), Pathogen–Host Interaction Data Integration and Analysis System (PHIDIAS) (Xiang *et al.*, 2007), Galaxy (Afgan *et al.*, 2016) and Unipro UGENE (Okonechnikov *et al.*, 2012) exist, but they are often server or cloud-based. The infrastructure behind some of these projects, and their cloud-based nature, introduce roadblocks in the transfer of data to and from their servers (Li and Homer, 2010). One solution to such a limitation is to establish an onsite computational cluster. However, technical and infrastructure requirements may pose further barrier-to-entry for data analysis.

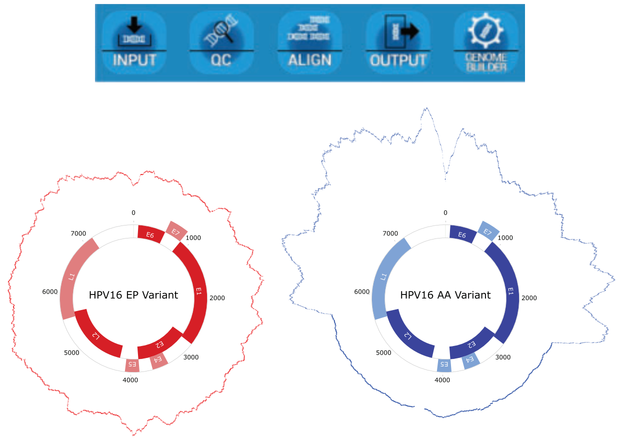

Fig. 1. Pathogen–Host Analysis Tool (PHAT) and visualization. The toolbar allows input of pathogen-containing NGS data and reference sequences, quality control with FastQC, alignment with Bowtie2 or HISAT2, SNP detection with VarScan2, visualization with pileup.js and the genome builder, as well as tabular output. Example HPV16 genome maps and coverage plots were generated to contrast viral variants: the European Prototype (EP) variant is episomal, whereas the Asian-American (AA) variant's coverage is disrupted by integration into host DNA (Jackson *et al.*, 2016)

We sought to develop the Pathogen–Host Analysis Tool (PHAT) to alleviate these issues by presenting an easy-to-setup and easy-to-use platform for life scientists conducting pathogen–host NGS analysis on common desktop computing hardware (e.g. Windows).

2 Features

Pathogen–host NGS analysis typically begins with high-throughput sequencing output files: experimentally relevant nucleic acid read information. PHAT is a platform for analyzing these data, with a focus on pathogen sequences within NGS data (Fig. 1). Reads are entered into PHAT as FASTQ files (Cock *et al.*, 2010), comprised of sequence reads with per base nucleotide identities and quality scores, or pre-aligned SAM/BAM files (Li *et al.*, 2009) generated via powerful cloud-based tools such as Galaxy (Afgan *et al.*, 2016). Quality control can be performed on individual files, with graphical reports generated. Reference genomes, recorded as FASTA files, must be indexed before they can be visualized or used for analysis. Once a pair of forward and reverse reads (paired FASTQ files) and a reference have been input, alignment can occur. PHAT also supports unpaired alignment and visualization of pre-aligned sequences.

The core functions of the PHAT platform as well as FASTQ quality control, sequence alignment, visualization, and its automated analyses are performed through well-known, established implementations. FastQC (https://www.bioinformatics.babraham.ac.uk/projects/fastqc) is used for quality control scoring. Sequence alignment is done by Bowtie2 (Langmead and Salzberg, 2012) or HISAT2 (Kim *et al.*, 2015), while linear alignment visualization is via pileup.js (Vanderkam *et al.*, 2016). Circular genomes are viewed with our enhancements to AngularPlasmid (http://angularplasmid.vixis.com/) which we make available as a new project called ngPlasmid (https://github.com/chgibb/ngPlasmid). Automated variant calling of single-nucleotide polymorphisms (SNPs) is by VarScan2 (Koboldt *et al.*, 2012).

The graphical user interface, based on GitHub's Electron project (https://electronjs.org/docs), operates in a client-server-based architecture. Each window acts as a client, communicating with a background server process. The server manages the saving and propagation of workspace data, as well as the generation of additional processes such as sequence alignment and quality control. This mechanism allows processes to act as threads, allowing the flow of data to and from the application window that invoked it and the created process itself. On systems with limited power, the server process limits the number of concurrently running processes and the amount of data propagated between windows to reduce memory and central processing unit (CPU) usage. We utilize an internal pipeline, spawning new processes as others end, passing data from one application window to another (e.g. alignment output). The server process, as well as the application windows themselves are implemented in Typescript. These windows can be conveniently undocked from the main toolbar.

3 Future work

With the development of PHAT, we aim to bring simple-to-use cross-platform NGS analysis to off-the-shelf hardware for life scientists studying pathogen–host relationships. In our own lab, we study human papillomavirus type 16 (HPV16) variants and their tumourigenicity in epithelia using NGS (Jackson *et al.*, 2016), but PHAT can be applied to a wide variety of pathogen–host relationships (e.g. genotyping of microbes such as viruses, bacteria, fungi and protozoans) from host NGS samples. To aid in our own experimental work, including analysis of HPV sequences within curated datasets (e.g. The Cancer Genome Atlas, TCGA), we are currently testing a viral-host integration detection feature in PHAT, with linkage to sequence databases. Additional features could include advanced alignment options as well as tools for further exploring pathogen–host interactions. We plan to actively develop, update and support PHAT based on user feedback and needs, with auto-updating features already included, in anticipation of building an active user and developer community.

Acknowledgements

The need for PHAT was conceived by RJ and IZ. Interface was designed by CMG and RJ, with programming by CMG. Manuscript writing was carried out by CMG, RJ, SM, JF and IZ. Intellectual property considerations made by SM and JF. Thanks to Zehbe Lab members for user testing, Dr. M. Togtema for feedback, as well as students M. Pynn, J. Braun, S. Liu, Z. Moorman and N. Catanzaro for improvements. We are thankful to the developers of open-source tools that are used in PHAT as well as GitHub and Reddit communities for helpful discussions.

Funding

This work was supported by Natural Sciences and Engineering Research Council of Canada (NSERC) grants to IZ (#RGPIN-2015-03855) and RJ (CGS-D #454402-2014). The funding body had no role in study design, data collection, data analysis and interpretation, or preparation of the manuscript.

Conflict of Interest: none declared.

References

Afgan,E. *et al.* (2016) The Galaxy platform for accessible, reproducible and collaborative biomedical analyses: 2016 update. *Nucleic Acids Res.*, **44**, W3–W10.

Cock,P.J. *et al.* (2010) The Sanger FASTQ file format for sequences with quality scores, and the Solexa/Illumina FASTQ variants. *Nucleic Acids Res.*, **38**, 1767–1771.

Jackson,R. *et al.* (2016) Functional variants of human papillomavirus type 16 demonstrate host genome integration and transcriptional alterations corresponding to their unique cancer epidemiology. *BMC Genomics*, **17**, 851.

Kim,D. *et al.* (2015) HISAT: a fast spliced aligner with low memory requirements. *Nat. Methods*, **12**, 357–360.

Koboldt,D.C. *et al.* (2012) VarScan 2: somatic mutation and copy number alteration discovery in cancer by exome sequencing. *Genome Res.*, **22**, 568–576.

Langmead,B. and Salzberg,S.L. (2012) Fast gapped-read alignment with Bowtie2. *Nat. Methods*, **9**, 357–359.

Li,H. and Homer,N. (2010) A survey of sequence alignment algorithms for next-generation sequencing. *Brief. Bioinform.*, **11**, 473–483.

Li,H. *et al.* (2009) The sequence alignment/map format and SAMtools. *Bioinformatics*, **25**, 2078–2079.

Li,Y. *et al.* (2015) ViRBase: a resource for virus-host ncRNA-associated interactions. *Nucleic Acids Res.*, **43**, D578–D582.

Lyons,E. and Freeling,M. (2008) How to usefully compare homologous plant genes and chromosomes as DNA sequences. *Plant J.*, **53**, 661–673.

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

Okonechnikov,K. *et al.* (2012) Unipro UGENE: a unified bioinformatics toolkit. *Bioinformatics*, **28**, 1166–1167.

Vanderkam,D. *et al.* (2016) pileup.js: a Javascript library for interactive and in-browser visualization of genomic data. *Bioinformatics*, **32**, 2378–2379.

Xiang,Z. *et al.* (2007) PHIDIAS: a pathogen–host interaction data integration and analysis system. *Genome Biol.*, **8**, R150.