Title

High-throughput pipeline for de-novo assembly and drug resistance mutations identification from Next-Generation Sequencing viral data of residual diagnostic samples.

Authors

Tiziano Gallo Cassarino[1], Daniel Frampton[1], Robert Sugar[2], Elijah Charles[2], Zisis Kozlakidis[1], Paul Kellam[1, 3]

[1] University College London, Division of Infection and Immunity, London, UK

[2] Health and Life Sciences, Intel Corporation, London, UK

[3] Wellcome Trust Sanger Institute, Hinxton, Cambridge, UK

Abstract

Motivation

Viral infections represent one of the most serious challenges to public health; the high genomic variation expressed by the viral population within an individual patient can lead the drug therapy to failure. Next-generation sequencing enables to identify viral quasi-species and to quantify the minority variants present in clinical samples; therefore it can be of direct benefit in terms of devising optimal treatment strategies for individual patients.

Method

Within the ICONIC (InfeCtion respONse through vIrus genomiCs) project, we developed an automated, portable and customisable high-throughput analysis pipeline to generate de-novo viral whole genomes and quantify minority variants from residual diagnostic samples. Our pipeline analyses Illumina short paired reads and can assemble either single or multiple segments viral genomes.

Results

The ICONIC pipeline was benchmarked on a dedicated High Performance Computing cluster using a pilot set of paired reads from 420 HIV clinical samples not filtered by viral load or amplification quality. The median genome length was 82% respect to the HIV-1 reference sequence (HXB2). The analysis lasted less than 10 hours, each sample took around 4 hours and required 5 GB of memory on average. The pipeline can be ported on a cluster or a single server through either an installation file or a Dockerfile.

Conclusions

It is technically possible for clinicians to obtain subtype information and a list of relevant Drug Resistance Mutations within three days of sample collection, therefore our pipeline can be used as a decision support tool towards more effective personalised treatments.

Availability

The pipeline and its documentation can be found on the GitHub repository https://github.com/ICONIC-UCL/pipeline.

Introduction

RNA viruses are intracellular parasites characterised by a high replication rate that allows them to generate a population of up to 10^{12} particles within the host organism [*"The population genetics and evolutionary epidemiology of RNA viruses" Nat Rev Microbiol. 2004*]. During this process, their polymerase proteins are very prone to transcription errors, so that RNA viruses have the highest mutation rate with respect to any other organisms. Given the short size of their genomes, ranging between 3 and 30 kilo bases, new mutant genomes are continuously generated and selected on the basis of their fitness to infect and to replicate within the host's cells [the population genetics and evolutionary epidemiology of RNA viruses]. Moreover, recombination of either pieces or segments of their genomes may increase the evolution dynamics and the genomic divergence of a viral population. One example is the Human Immunodeficiency Virus (HIV), in which the recombination is faster than the mutation rate [*"High rates of human immunodeficiency virus type 1 recombination: near-random segregation of markers one kilobase apart in one round of viral replication" J Virol. 2003*]. These and other features suggest that a viral population is actually made of an ensemble of related mutants that can be described as a quasi-species and on which the selective pressure influences all the viruses as a single unit.

Such high genomic variability allows the viral population to survive to the host's immune system and to antiviral agents, hindering the correct treatment of patients and making it difficult to find suitable drugs to eradicate the infection [*"Viral quasi-species and the problem of vaccine-escape and drug-resistant mutants" Prog Drug Res. 1997*].

Next-Generation Sequencing (NGS) coupled with bioinformatics analysis enables to detect genomic variants and to classify known and novel viruses without relying on the more expensive virus culturing and on labor-intensive Sanger sequencing [*"Deep sequencing: Becoming a critical tool in clinical virology" J Clin Virol. 2014*].

Applied to a clinical setting, NGS data can be applied on sequenced dignostic samples to identify pathogens by building genomes and to quantify low-level drug resistance mutations below the 15-20% frequency limit of the Sanger sequencing. These additional information can be used to improve the treatment of patients both for viral and bacterial infection.

Computational pipelines that employ NGS data have been developed to discover new pathogens [*"A cloud-compatible bioinformatics pipeline for ultrarapid pathogen identification from next-generation sequencing of clinical samples" Genome Res. 2014, "Identification of novel viruses using VirusHunter--an automated data analysis pipeline" PLoS One. 2013*], to identify viral quasispecies [*"QuRe: software for viral quasispecies reconstruction from next-generation sequencing data" Bioinformatics. 2012, "Viral Quasispecies Assembly via Maximal Clique Enumeration" PLoS Comput Biol. 2014*] and to detect genomic variants [*"LoFreq: a sequence-quality aware, ultra-sensitive variant caller for uncovering cell-population heterogeneity from high-throughput sequencing datasets" Nucleic Acids Res. 2012, "Viral* *population analysis and minority-variant detection using short read next-generation sequencing" Philos Trans R Soc Lond B Biol Sci. 2013, "Accurate single nucleotide variant detection in viral populations by combining probabilistic clustering with a statistical test of strand bias" BMC Genomics. 2013, "V-Phaser 2: variant inference for viral populations" BMC Genomics. 2013, "VirVarSeq: a low-frequency virus variant detection pipeline for Illumina sequencing using adaptive base-calling accuracy filtering" Bioinformatics. 2015*]. However, in order to improve patient treatment through these discoveries in clinical settings, it is also necessary for pipelines to be capable of both assembling de-novo genomes and reporting minority variants, in addition to being scalable for high-throughput analysis, customisable and portable to different software environments.

Therefore, we developed a computational pipeline for analysing NGS data of a target virus and that meets all those requirements; the input data are raw Illumina short paired reads from residual diagnostic samples and, as output, it reports the consensus genome and all minority variants present in the viral quasi-species.

The pipeline is part of the ICONIC project, which aims to make use of genomic data for providing a decision support approach towards a personalised treatment of patients for chronic infections, to guide hospital infection control responses and to inform surveillance and epidemiological responses to community outbreaks.

Methods

Pipeline Architecture

The pipeline is developed for analysing batches of sample reads on a Son of Grid Engine High Performance Computing (HPC) cluster and it is available either as an installation file or as a Dockerfile [https://www.docker.com/]. Either of these alternatives will automatically: (1) download the pipeline, its dependencies from the public GitHub repository and virus reference sequence databases, (2) configure the pipeline and (3) install it on the local appliance. Finally, the installation process makes the pipeline available as a bundle of modules *["Abstract Yourself With Modules", Proceedings of the Tenth Large Installation Systems Administration Conference (LISA '96) 1996.*], which can be used straight away.

Reads Analysis

The pipeline takes as input a viral name (e.g. "influenza" or "hiv") and a batch of short paired-end reads generated from one or more samples and it returns the consensus genome, the minority variants and a set of statistics for each analysed sample. It consists of functional units which allow either to run the whole pipeline starting from a given step or to only run a specific unit independently of the others. These functional units can be tuned through several input options and by means of a configuration file, which stores the database paths and/or other user-defined paths as static parameters.

Figure *1*: Workflow of the pipeline

After parsing the input parameters, the pipeline initialises and sends a job array to the HPC cluster, which runs the analysis of each sample reads in parallel to speed up the analysis. Each job array is composed of four distinct units.

The first pipeline unit takes as input the paired-end reads in a compressed fastq format and keep only the high quality non-contaminant pairs. The two files are passed to Trimmomatic (version 0.33) [*"Trimmomatic: A flexible trimmer for Illumina Sequence Data" Bioinformatics. 2014*], which removes or trims low quality reads (used with default settings, except for the sliding window quality cutoff of 30). The remaining read pairs are mapped against a decoy genome -- made of human chromosomes and of a reference viral genome -- in order to remove both unmapped reads and read pairs that map better to the host than to the pathogen. The mapping can be performed either with SMALT (version 0.7.6) [http://www.sanger.ac.uk/science/tools/smalt-0] or BWA (0.7.12) *["Aligning sequence reads, clone sequences and assembly contigs with BWA-MEM" arxiv.org 2013*]. Samtools (1.2) [*"The Sequence Alignment/Map format and SAMtools". Bioinformatics. 2009*] is used to manipulate and to extract data from the alignments and the read files. Whenever possible, external softwares are run in multithreading using all the available cores. FastQC (version 0.11.3) [http://www.bioinformatics.babraham.ac.uk/projects/fastqc/] is run both before and after this filtering step to perform some quality control checks and to spot possible sequencing biases. The results are saved in each sample's output folder for a later manual inspection of the read quality.

The second unit generates the sample consensus genome. In case no contigs are provided as input, they get created using the filtered read pairs by a de-novo assembler. In the current implementation, contigs are generated by the de-novo assembler "IVA" (version 1.0.0) [*"IVA: accurate de novo assembly of RNA virus genomes" Bioinformatics. 2015*]. Afterwards, contigs are aligned with blast+ (version 2.2.30) ["*BLAST+: architecture and applications" BMC Bioinformatics 2009*] against a local BLAST database with all the genomic sequences (identified by subtype and accession code) of the viral species specified as input, in order to find the sequence with the best match. Databases and decoy genomes can be added to the pipeline just by including their paths to the configuration file, thus allowing the pipeline to analyse different NGS viral data. To create a BLAST database. To create a draft genome using the contigs, these are aligned by LASTZ (1.07.73) [*Harris, R.S. (2007) Improved pairwise alignment of genomic DNA. Ph.D. Thesis*] to the previously found sequence or segments. The contigs with the highest read depth of coverage substitutes the corresponding aligned pieces of the reference, so that the resulting draft genome is made of contigs joined together by fragments of the reference. To generate a final consensus genome representing the nucleotide content of the sample, the reads are mapped iteratively to the genome and its bases are substituted at each position with the majority variants from the reads until the genome converges to a stable sequence, that is, there are no more substitutions to be made. Maximum 10 cycles are performed to avoid infinite iterations. In case less than 50 reads are aligned to a genomic position, the corresponding base in the consensus genome is substituted to N to indicate that there is not enough data to reliably assign a nucleotide.

In the third unit, the read pairs are mapped for the last time to the consensus genome and the minority variants at each genomic position are identified using Samtools. To reduce false positive variants, only those with a read depth of at least 100 and present in at least 50 reads are reported. These cutoffs were selected after manual inspection of test batches of samples and they ensure that at least 5 reads are present to call a variant. The minority variants are reported by binning them by their frequency with cutoffs of least 20%, 10%, 5% and 2% of reads depth, in accordance with the Public Health England recommendation [@Zisis: could you please add some references for these cutoffs?].

In the fourth unit, summary metrics and plots are generated for the whole consensus genome and for each genomic position; these metrics include genome length, read depth of coverage distribution, number of variants and strand bias. Optionally, it is possible to keep all the intermediate files created during the analysis but they use more disk space, between 3 to 5 Gigabytes for each sample, mainly due to the filtered reads, alignment and pileup files.

The pipeline produces two main logging files during the execution of the analysis. The first report stores all the calls to the external software (e.g. Samtools mpileup) with their memory usage and duration, while the second one can be tuned in the input option to trace all the steps and intermediate results during the analysis. Moreover, the pipeline reports the cause of any interruption of the analysis, for example when it stops because the assembly is not possible due to an insufficient number of reads; therefore, it is possible to inspect the analysis workflow to identify problematic samples.

Additional details on the pipeline options and parameters can be found in the software documentation or in the GitHub repository.

Results and Discussion

Comparison with public genomes

We selected a test batch of 39 publicly available sequences, with deposited reads, of the Human Respiratory Syncytial Virus (RSV), A and B subtypes, from the Sequence Read Archive (SRA). We created a local Blast database of reference sequences from full genomes available in GenBank (as of 2015-10-01). With respect to the publicly available genomes, those built with our pipeline were longer in 19 samples (49%), shorter only in 7 (18%) due to the small number of reads remained after the quality filtering step, and of comparable length in 13 samples (33%) (Details can be found in the Table S1 of the supplementary material). The median length of built genomes was 96% of the corresponding reference. Therefore, our pipeline can build consistently full-size genomes and also contribute to the deposited public sequences by improving their lengths.

Analysis of HIV clinical samples

As an example of the pipeline capabilities on sequencing data generated from clinical samples, we describe the results generated for two batches of paired reads from 420 HIV samples *(D. Frampton, T. Gallo Cassarino, Z. Kozlakidis, A. Hayward, P. Kellam, in preparation)* sequenced at the Wellcome Trust Sanger Institute (Cambridge) with an Illumina MiSeq machine. Using the sequences downloaded from the Los Alamos HIV Sequence Database (http://www.hiv.lanl.gov/content/sequence/HIV/mainpage.html) updated to the GenBank version of 2015-10-01, we generated a local Blast database of HIV sequences containing complete genomes for all HIV-1 subtypes. In case of successful de-novo assembly, for each sample analysed, we saved the consensus genome and the minority variants found in the reads. We aligned the pipeline-generated genomes against the HXB2 reference sequence (GenBank accession K03455) with Lastz (version 1.03.74) to measure their lengths, their coverage for each position and for each gene. As shown in Fig. 2, we observed that the genome coverage increases with the primer efficiency and the number of overlapping primer regions.

Figure *2*: Fraction of genomes aligned at each position of the reference sequence HXB2

The pipeline was able to generate clinical length genomes with a median coverage of 82%, independently of the sample virus subtype as predicted by COMET [*"COMET: adaptive context-based modeling for ultrafast HIV-1 subtype identification" Nucleic Acids Res. 2014*] (Fig. 3).

Figure *3* Distribution of the genome lengths built by the pipeline from the HIV sample reads, divided by the subtypes identified by COMET.

Coverage at the gene level also depends on primer lengths and efficiency. For example, complete Gag sequences are obtained for 81% of samples with a mean coverage of 85%, whereas only 45% generate complete sequences for the Env gene, with a mean coverage of 63% (Fig. 4).

Figure *4* Distribution of the gene coverage across the 420 HIV samples.

Given the observed gene coverage within the samples, co-screening for all drug-resistance mutations in protease, RT and integrase could be performed for about 75% of the analysed samples. In total, 68096 minority variants were called across all samples in 6191 positions (68% of the genome length) within the primer set boundaries. Variants were not uniformly distributed across the genomes; within Env gp120 they were significantly over-represented (p<0.001, using a binomial model as test). The number of genomes with variants at each position is illustrated in Figure 5.

Figure *5* Distribution of minority variants across the samples.

Performances

The analysis was performed using the University College London HPC cluster "Legion" on 124 dedicated Dell C6220 nodes, where each node can work as a 16 core Symmetric Multi-Processing device with 64 GB of RAM. Legion runs an operating system based on Red Hat Enterprise Linux 7 with the Son of Grid Engine batch scheduler.

The pipeline ran about 4 hours per sample on average. The longest step was the de-novo assembly, which is responsible for 90% of the pipeline execution time and increases with the nucleotide variability present in the reads.

Except for the scripts that build the consensus genome and manage the flow of the data through the pipeline, all the other software was run in multi-threading using all the available cores to decrease the computational time of the analysis.

The average peak memory usage, which depends on the number of sample reads stored in memory, reached about 10 GB during the filtering and mapping steps; however, the memory required during the rest of the analysis dropped below 2 GB.

Docker images have a negligible impact on the performances [The impact of Docker containers on the performance of genomic pipelines] and this holds true especially for our pipeline, since it is contained in a single Docker image.

Portability

The pipeline can be installed either through a Dockerfile or by means of a standard installation file on a GNU/Linux system. The former method can be used on a dedicated single-user appliance, while the latter is better suited for environments shared among multiple users in which Docker cannot be installed for security reasons, as typical of computational clusters in academia.

Availability

The pipeline and its documentation can be downloaded or cloned from the GitHub repository at https://github.com/ICONIC-UCL/pipeline

Conclusions

Motivated by the increasing applications of NGS on viral data as a method to improve the treatment of patients affected by viral diseases, we developed a high-throughput computational pipeline to assemble consensus genomes de-novo and to detect minority variants from reads of any single virus quasi-species. Our pipeline can be ported to different computational infrastructures, as found in clinical settings, since it can be easily installed on a cluster or on a single server running a GNU/Linux system, depending whether Docker is available.

We compared a set of public RSV sample sequences to the genome generated by our pipeline starting from the deposited reads. We were able to generate full genomes in most of the samples, with half of them being longer than the publicly available ones. We analysed two batches of HIV reads sequenced from diagnostic residual samples on a HPC cluster, to show that the pipeline can successfully assemble de-novo genomes. The analysis lasts only few hours and requires a reasonable amount of memory for each sample, thus capable of processing batches of hundreds samples overnight.

Therefore, our pipeline can be reliably utilised on sequencing data of any known virus, to generate de-novo full length viral genomes in real case scenarios, in which the sample quality is very variable and the virus subtype is unknown. These features empower our pipeline to be employed in clinical settings as a crucial decision support tool towards a personalised approach of patient treatment and an improved management of hospital infections.

Acknowledgments

We would like to thank Chiara Garattini for suggestions and useful discussions.

Figure 1: Workflow of the pipeline

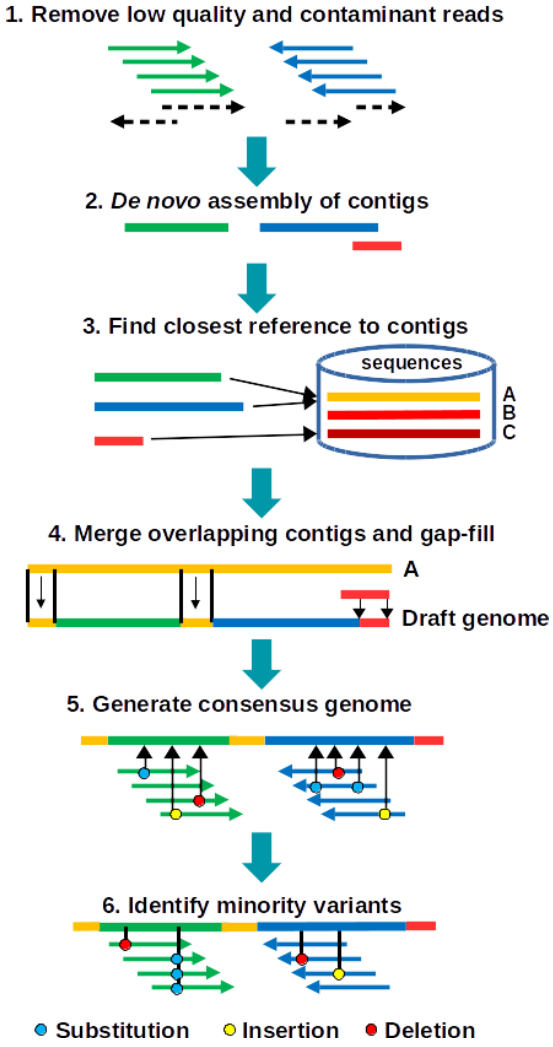

1. Remove low quality and contaminant reads
2. *De novo* assembly of contigs
3. Find closest reference to contigs
sequences
A
B
C
4. Merge overlapping contigs and gap-fill
A
Draft genome
5. Generate consensus genome
6. Identify minority variants
● Substitution
● Insertion
● Deletion

Figure 2: Fraction of genomes aligned at each position of the reference sequence HXB2

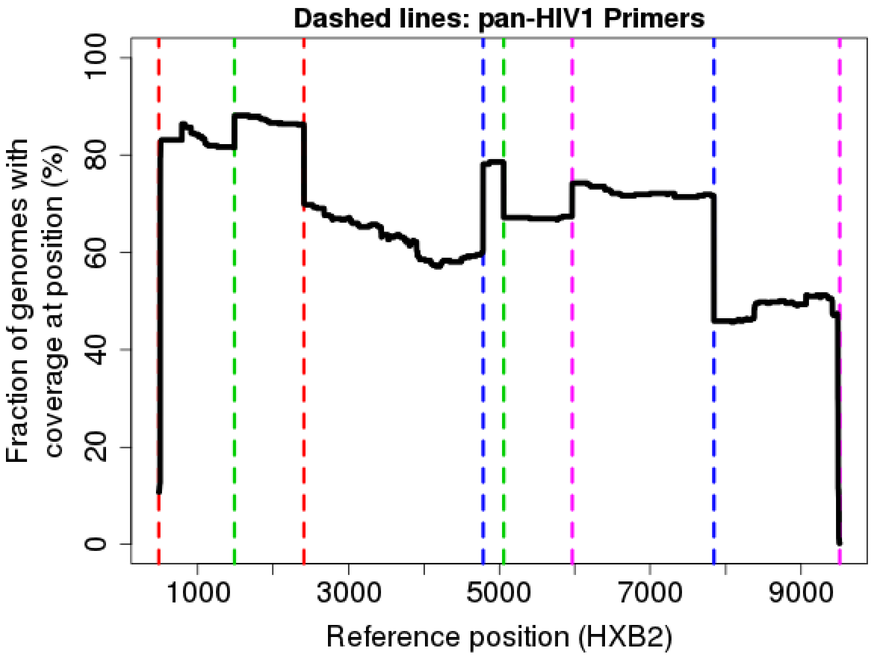


Figure 3: Distribution of the genome lengths built by the pipeline from the HIV sample reads, divided by the subtypes identified by COMET. NA indicates that there was not enough data to predict a subtype, while U1 and U2 show undefined subtypes.

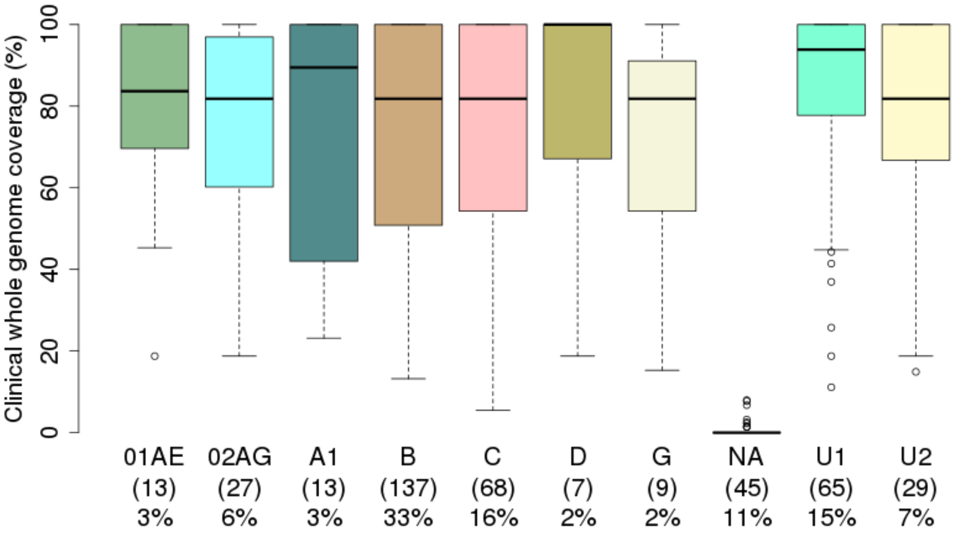


Figure 4: Distribution of the gene coverage across the 420 HIV samples.

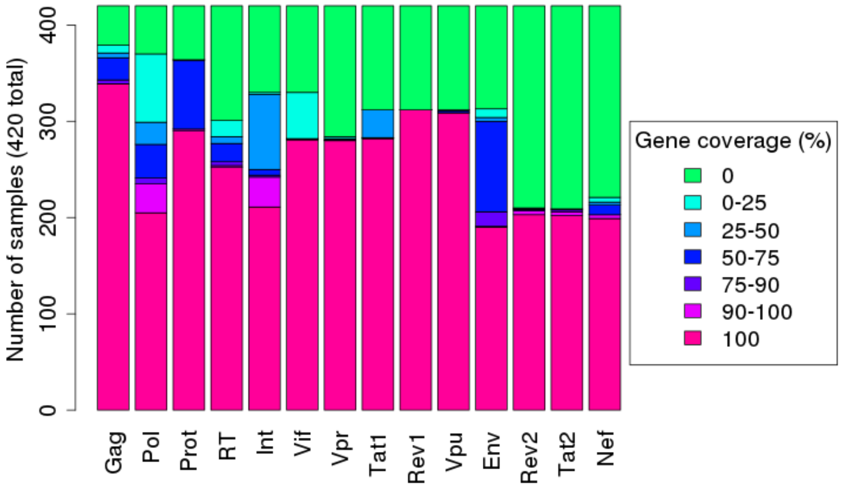
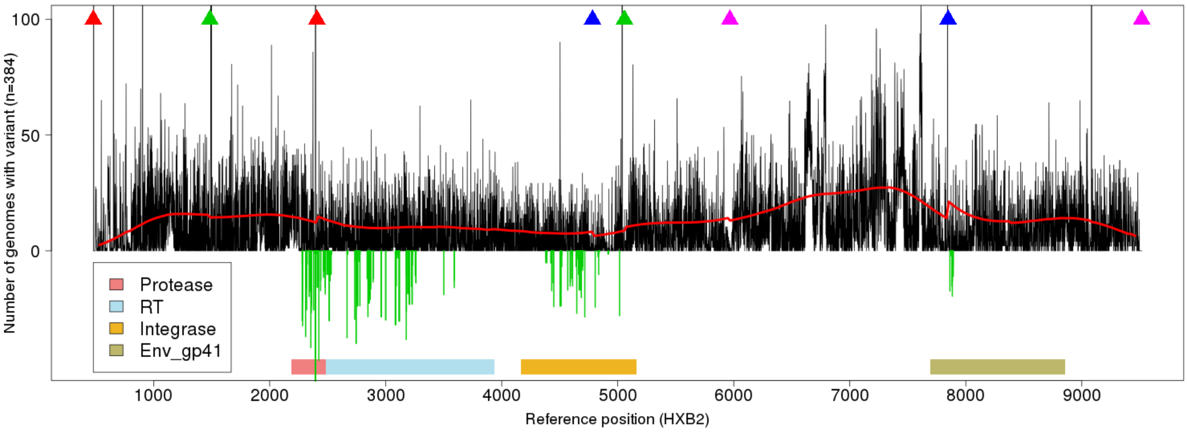


Figure 5: Distribution of minority variants across the samples. Black spikes indicate the number of minority variants; the relative frequency is shown with a red smoothed line. Primer binding sites are denoted by pairs of coloured triangles. Green spikes indicate the number of minority variants at known drug resistance mutation sites (PI, NRTI, NNRTI and FI), while the coloured bars at the bottom show the positions of the proteins Protease, RT, Integrase and gp41 within the Env gene.


Number of genomes with variant (n=384)
Protease
RT
Integrase
Env_gp41
Reference position (HXB2)