

Consensus and variant calling practical

Introduction

In this session, you will learn about how to generate a consensus and look at variants within viral NGS datasets. We will then see how to interpret mutations and to check for drug resistance mutations (DRM) in the context of HIV.

After studying this tutorial section you should be able to:

- use tools to call variants based on reference genome
- investigate variants of interest ie. DRM

You need to have an alignment file (SAM/BAM file) and make sure that duplicates are removed and the file is sorted.

Software used

- samtools
- bcftools
- varscan

These are pretty standard softwares for consensus and variants calling. There are alternatives, for example the BBTools (<https://jgi.doe.gov/data-and-tools/bbtools/bb-tools-user-guide/>) offers a variant calling tool `callvariants.sh`

Before we start...

Let's create a directory for all our results:

```
mkdir variant_calling
cd variant_calling
```

If your computer is struggling, all results are saved in a file called `variant_calling_results.zip`

```
unzip ~/results/variant_calling_results.zip
```

Pileup file

For both calling consensus and variants we need to produce a pileup file. The SAMtools mpileup utility provides a summary of the coverage of mapped reads on a reference sequence at a single base pair resolution. In addition, the output of mpileup can be piped to BCFtools and Varscan to call genomic variants and consensus. If you type:

```
samtools mpileup
```

without any parameters, the usage and the parameters will be displayed. It is useful to use the -f parameter, which specifies the reference file, so that the actual nucleotide at a genomic location is printed out. The -s parameter is useful for outputting the mapping qualities of reads. I often use the -A parameter to keep anomalous read pairs.

```
samtools mpileup \  
-f /home/training/Course_Data/Ref.C.IN.95.95IN21068.AF067155.fa \  
/home/training/Course_Data/PG15-BW001347_without_duplicates.bam -s \  
> PG15-BW001347.pileup
```

Let's have a look at the pileup file we created:

```
#this command will print the first 2 lines of the pileup file  
sed -n '1,2p' PG15-BW001347.pileup
```

Each line represents a single genomic position and has seven columns:

1. Sequence name
2. 1-based coordinate
3. Reference base
4. Number of reads covering this position
5. Read bases
6. Base qualities
7. Alignment mapping qualities

Looking at our file, the first line is position 1 which on our reference was a A and was covered by 70 reads. The 5th column (read bases) contains information on whether a read base matched, mismatched, was inserted or deleted with respect to the reference. It also contains information about whether the read base was on the positive or negative strand with respect to the reference, and whether a read base was at the start or the end of a read. A period stands for a match to the reference base on the positive strand, a comma for a match on the negative strand. We generated paired-end reads, so we have reads on both positive and negative strand. The “^” signifies a base that was at the start of a read and the “\$” signifies a base that was at the end of a read.

Consensus

We can obtain the sequence of the most frequent nucleotides at each position, called consensus sequence.

We can pipe the output of SAMtools mpileup directly to BCFtools - this is a long command, but we are going

```
samtools mpileup \  
-uf /home/training/Course_Data/Ref.C.IN.95.95IN21068.AF067155.fa \  
/home/training/Course_Data/PG15-BW001347_without_duplicates.bam \  
| bcftools call -mv -Oz -o PG15-BW001347_calls.vcf.gz
```

the first two lines of the command are similar to the mpileup command we run before. Note that we now have the -u option to make the output go with bcftools. The bcftools command is calling variants i.e. difference from the reference. This is the first step of our consensus calling. We use the following parameters:

- -o write output to a file
- -Oz output type, zip in this case
- -v variants caller
- -m multiallelic caller

This command produces a file called PG15-BW001347_calls.vcf.gz. We are going to look at vcf files further a bit later on in this tutorial, but if you have time you can explore the file with:

```
zless PG15-BW001347_calls.vcf.gz
```

zless is a cool command, it works like the unix command "less" but for zip file!! Check out also zcat.

We now get the index of the vcf file with "tabix":

```
tabix PG15-BW001347_calls.vcf.gz
```

and finally we can call the consensus and get our sequence! Create consensus:

```
bcftools consensus PG15-BW001347_calls.vcf.gz -M N -p PG15-BW001347_ \  
-f ~/Course_Data/Ref.C.IN.95.95IN21068.AF067155.fa > PG15-BW001347.fa
```

- -M missing in this case "N"
- -p prefix to add to output sequence names
- -f reference fasta

What can you do with the consensus sequence?

The consensus sequence is the standard format for several phylogenetic analysis i.e. tree building, genetic distance, exploring transmission and so on. You can also check whether

your sequence contains any known drug resistance mutations. For that we can use the HIV Drug resistance database by Stanford University at this website:
<https://hivdb.stanford.edu/hivdb/by-sequences/>

```
cat PG15-BW001347.fa
```

copy your sequence or upload the fa file.

HIVdb Program
Genotypic Resistance Interpretation Algorithms

Sima version: 1.4.2 (last updated on 2019-11-01)
HIVdb version: 6.9.1 (last updated on 2019-10-25)

HIVdb accepts user-submitted protease, RT, and integrase sequences or mutations and returns inferred levels of resistance to the most commonly used protease, nucleoside, non-nucleoside, and integrase inhibitors. Its purpose is educational and as such it provides extensive comments and a highly transparent scoring system that is hyperlinked to data in the HIV Drug Resistance Database. A detailed description of the program as well as all updates is in the [Release Notes](#). A [web service](#) has been created to allow users to access HIVdb programmatically.

Sequences can be entered as plain text if just one sequence is entered. Sequences must be entered using the FASTA format if multiple sequences are entered. Sequences can be pasted in the text box or uploaded using the File Upload option. The upper limit is currently 1000 sequences containing 3000 nucleotides per sequence.

By default the program output will contain an HTML page with a navigation sidebar linking to the results for each sequence. Users selecting the spreadsheet output options, will obtain links to tab-delimited files containing tabular sequence summary data, tabular resistance summary data, and a formatted amino acid sequence alignment. Detailed descriptions of the HTML and spreadsheet output are explained in the [Release Notes](#).

Drug display options

By default, results will be shown for checked ARVs. Use checkboxes for additional ARVs. [Select all ARVs](#), [revert to default](#)

NRTI: ☒ ABC ☒ AZT ☒ FTC ☒ 3TC ☒ TDF ☐ B4T ☐ DDH
INSTI: ☒ RBC ☒ DTG ☒ EVG ☒ RAL
NNRTI: ☒ DOR ☒ EFV ☒ ETR ☒ NVP ☒ RPV
PI: ☒ ATV ☒ DRV ☒ LPV ☐ FPV ☐ IDV ☐ NFV ☐ SQV ☐ TPV

Input mutations | **Input sequences**

Header: (optional)

Upload text file: [Choose File](#) No file chosen

Here you can copy your sequence

Output options

☒ HTML ☐ Printable HTML ☐ Spreadsheets (TSV) ☐ XML

Click here to start the analysis

[Run](#) [Analyze](#)

Call minority variants

We can use a SNP calling software to look for variants, for example samtools/bcftools (as used before), Varscan, GATK and FreeBayes. Each one of these SNP callers make different assumptions about the reference genome and the reads, so each one of them is best suited for different situations.

Varscan (<http://varscan.sourceforge.net/using-varscan.html>) uses a simple method based on counting the number of reads for each alleles after appropriate thresholds for the sequencing and mapping qualities have been applied.

We are going to run varscan three times so we can get: - a file containing all position even those that do not differ from the reference - a file containing only the variants in tab format (I like this one because it's easy to check and can easily be read into R!) - a file containing only

the variants in vcf format

The input is the pileup file we created at the beginning of this practical!

This will show all position in the genome:

```
java -jar ~/software/varscan/VarScan.v2.4.4.jar mpileup2cns \  
PG15-BW001347.pileup --min-reads2 2 --min-avg-qual 20 \  
--min-var-freq 0.02 --p-value 0.01 \  
> PG15-BW001347_allpos_variants.tab
```

mpileup2cns command in VarScan

Common options:

- --min-coverage Minimum read depth at a position to make a call [8]
- --min-reads2 Minimum supporting reads at a position to call variants [2]
- --min-avg-qual Minimum base quality at a position to count a read [15]
- --min-var-freq Minimum variant allele frequency threshold [0.01]
- --p-value Default p-value threshold for calling variants [99e-02]

Explore the output: `~~~ less PG15-BW001347/allposvariants.tab ~~~`

OUTPUT:

- Tab-delimited consensus calls with the following columns:
- Chrom chromosome name
- Position position (1-based)
- Ref reference allele at this position
- Cons Consensus genotype of sample; */(var) indicates heterozygous
- Reads1 reads supporting reference allele
- Reads2 reads supporting variant allele
- VarFreq frequency of variant allele by read count
- Strands1 strands on which reference allele was observed
- Strands2 strands on which variant allele was observed
- Qual1 average base quality of reference-supporting read bases
- Qual2 average base quality of variant-supporting read bases
- Pvalue Significance of variant read count vs. expected baseline error
- MapQual1 Average map quality of ref reads (only useful if in pileup)
- MapQual2 Average map quality of var reads (only useful if in pileup)
- Reads1Plus Number of reference-supporting reads on + strand
- Reads1Minus Number of reference-supporting reads on - strand
- Reads2Plus Number of variant-supporting reads on + strand
- Reads2Minus Number of variant-supporting reads on - strand
- VarAllele Most frequent non-reference allele observed

This will save only the variants positions, using the flag "--variants 1"

```
java -jar ~/software/varscan/VarScan.v2.4.4.jar mpileup2cns \  
PG15-BW001347.pileup --min-reads2 2 --min-avg-qual 20 \  
--min-var-freq 0.02 --p-value 0.01 --variants 1 \  
> PG15-BW001347_variants.tab
```

Vcf files with the flag "--output-vcf 1"

```
java -jar ~/software/varscan/VarScan.v2.4.4.jar mpileup2cns \  
PG15-BW001347.pileup --min-reads2 2 --min-avg-qual 20 \  
--min-var-freq 0.02 --p-value 0.01 --variants 1 --output-vcf 1\  
> PG15-BW001347_variants.vcf
```

Let's explore the vcf file:

```
less PG15-BW001347_variants.vcf
```

Plot frequency distribution

We are going to use R to plot some distributions of the variants. To keep it simple we will use R from the command line, however I strongly suggest you use RStudio in the future.

Just type R and you will enter the R environment

```
R
```

We will use the package ggplot2 for plotting, which is part of the tidyverse tools. Tidyverse (<https://www.tidyverse.org/>) is a collection of R packages for data science and ggplot2 makes pretty plots, check it out: <https://www.r-graph-gallery.com/>

```
library(tidyverse)
```

We create a function to read the Varscan data

```
combined.df <- read.table("PG15-BW001347_variants.tab",  
header=TRUE,as.is=TRUE)  
df.het <- as.data.frame(matrix(unlist(strsplit(combined.df[,5],  
split=":")), ncol=6, byrow="T"), stringsAsFactors=F)  
all <- cbind(combined.df[,1:4], df.het[,1:5], combined.df[,6])  
  
colnames(all)[5]<-"Cons"  
colnames(all)[6]<-"Filter"  
colnames(all)[7]<-"Ref.count"  
colnames(all)[8]<-"Var.count"  
colnames(all)[9]<-"VarFreq"  
colnames(all)[10]<-"StrandFilter"
```

```
all$VarFreq <- readr::parse_number(all$VarFreq)
```

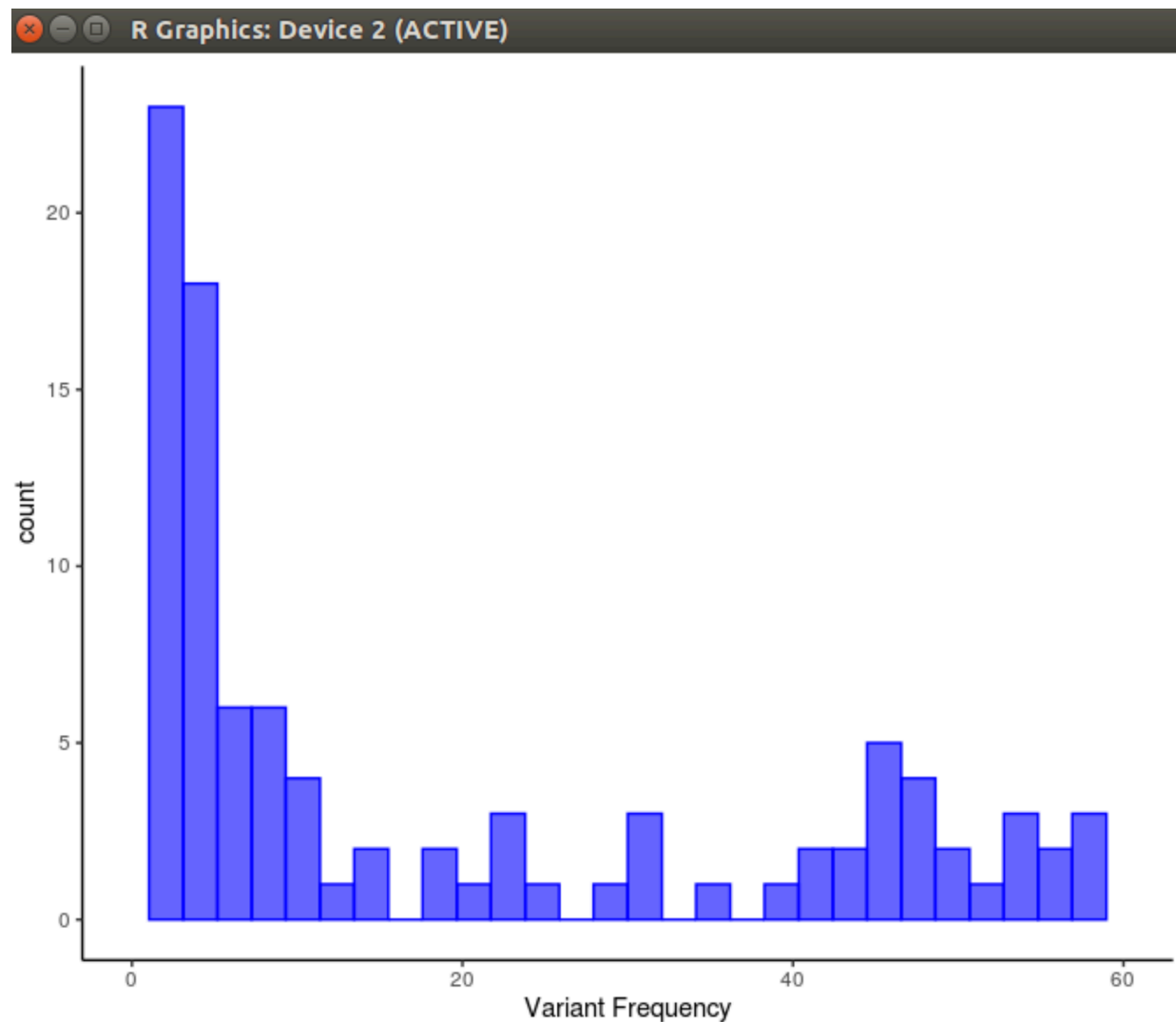
```
ggplot(all,aes(x=VarFreq)) +  
  geom_histogram()+  
  xlab("Variant Frequency")
```

A bit prettier

```
ggplot(all,aes(x=VarFreq)) +  
  geom_histogram(colour="blue", fill="blue",alpha=0.6)+  
  xlab("Variant Frequency") +  
  theme_classic()
```

Let's only plot the minority variants, those variants with frequency <60%

```
ggplot(all,aes(x=VarFreq)) +  
  geom_histogram(colour="blue", fill="blue",alpha=0.6)+  
  xlab("Variant Frequency") +  
  theme_classic() +  
  xlim(0,60)
```



This is showing you that there are other viral populations on top of the major one (which constitutes the consensus). If you have longitudinal samples, you can track the variants overtime and see whether new viral populations appear and/or if a minor one will end at the consensus level.

To exit R, type:

```
q()
```

then select "n"

Drug resistance mutations

Earlier in this session, we explore how to get drug resistance mutations from the consensus sequence. This is helpful, but often we have several viral populations coexisting in the same host. If any of this "quasi-species" contains a DRM we won't be able to see that at the consensus level, but we can use the information from the variants.

The Stanford DB has created a tool to obtain codon frequency from bam and fastq files and then you upload the codon table to their website.

Let's create a new directory for this:

```
mkdir drm  
cd drm
```

Let's copy a clean fastq files file for a sample:

```
cp ~/variant_calling/variant_calling_results/drm/PG15-BW001347.R*_val_*  
fastq2codfreq -p '(.+)\.R[12]\_val_[12]\.fq.gz$' -r '\1' \  
-1 '.R1_val_1.fq.gz' -2 '.R2_val_2.fq.gz' .
```

The easier version to run the program is `fastq2codfreq /path/to/folders/containing/fastq/files` (with file like `*L001R1001.fastq.gz` and `*L001R1002.fastq.gz`). You can specify your FASTQ file naming convention by passing `-p`, `-r`, `-1` and `-2` parameters to `fastq2codfreq`. Noted `-p` and `-r` are paired regular expression replacement. More info about this script can be found here: <https://hivdb.stanford.edu/page/codfreq/>

BE CAREFUL!!! There is also a tool for obtaining codfreq tables from sam/bam files.

However, you need to map to the correct sequence that HIV Stanford db is using (hint: type B)

This produces a file called "PG15-BW001347.codfreq" that you explore with:

```
less PG15-BW001347.codfreq
```


this is a codon frequency table which includes information about protease, RT, and/or integrase. CodFreq Files are tab- or comma-delimited text files containing the following 5 columns:

- Gene (PR, RT, or IN);
- Amino acid position;
- Number of reads for the position (coverage);
- Sequenced codon;
- Number of reads containing the codon.

Let's use the online database to explore the table:

Go to: <https://hivdb.stanford.edu/>

Click on **HIVdb-NGS (Beta)**:

Stanford University
HIV DRUG RESISTANCE DATABASE
A curated public database to represent, store and analyze HIV drug resistance data.

HOME GENOTYPE-RX GENOTYPE-PHENO GENOTYPE-CLINICAL HIVDB PROGRAM ABOUT HIVDB SUPPORT HIVDB!

SIERRA
Sierra 2.5.0
release notes / web service
Jan 24, 2020

HIVDB Algorithm
Version 8.9-1
Nov 1, 2019

HIVdb-NGS (Beta)
release notes
Oct 24, 2019

Reference Library: HIV-2 Resistance
A body of literatures reviewed, annotated and searchable
Sep 6, 2019

Reference Library: Dolutegravir Resistance
A body of literatures reviewed, annotated and searchable
Feb 1, 2019

Click here

HIVDB released on December 6, 2019
Query / Download

Genotype-treatment
ARV selection data comprising 169,458 protease, 178,901 RT and 23,390 integrase HIV-1 virus sequences from 190,341 persons; 1,022 protease, 802 RT and 340 integrase HIV-2 virus sequences from 1,110 persons.

Genotype-phenotype
Drug susceptibility data comprising 25,434 PI, 19,858 NRTI, 11,548 NNRTI and 4,907 INI susceptibility results from HIV-1 virus isolates

Genotype-clinical
Clinical outcome data comprising genotype, treatments, plasma HIV-1 RNA levels and CD4 counts from 14 clinical trials and >1500 Treatment-Change Episodes

References
1,704 references of genotype-treatment and/or genotype-phenotype data according to author-yr, including 181 references collected since 2019-01-01.
3,149 Genbank submission sets according to author-yr and submission title, including 34 new submissions from Genbank release on 2019-10-15.

Other Resources

Multi-Drug Resistant Panels
Database Mirror

REGA HIV-1 Subtyping Tool 3.0
HBVseq Program

HIVdb Program

Drug Resistance Summaries (Download PDF)
PIs NRTIs NNRTIs INSTIs

HIVdb NGS Program

HIVseq Program

HIValg Program

HIV-1 Genetic Variability for Drug Resistance

Single Genome Sequence Database

News & Updates

Surveillance Mutations

Reference Libraries

Point-of-Care / Essential Mutations

TCE

ART-AIDE

Publications

Upload PG15-BW001347.codfreq (remember this file is in /home/training/variant_calling/drm):



HIVdb-NGS (Beta)

[Release Notes](#)

Sierra version 3.0.0b2 (last updated on 2019-10-28)

HIVdb version 8.9-1 (last updated on 2019-10-25)

HIVdb-NGS (beta) accepts user-submitted protease, RT, and/or integrase [codon frequency tables \(CodFreq files\)](#) or [AAVF files](#) generated by the [HYDRA pipeline](#). Results are returned at 8 mutation detection thresholds. At each threshold the program quantifies the number of [unusual mutations](#) and the number of [signature APOBEC mutations](#). Thresholds with large numbers of unusual or signature APOBEC mutations are likely too low and pose unacceptable risks of identifying artifactual mutations caused either by machine error, PCR error, or G-to-A hypermutation.

By default, genotypic resistance interpretations are returned for the 20% threshold. A drop-down menu allows users to view results at lower thresholds. A multi-threshold mutation summary table is provided to help users identify thresholds to reduce the risk of identifying artifactual mutations. Genotypic resistance interpretations are suppressed at thresholds for which >1% of positions have an unusual mutation. The appropriate mutation detection threshold, however, cannot be identified with certainty in the absence of a sequencing protocol that uses unique molecular identifiers (UMIs) for each virus template.

Mutations detected at low thresholds are difficult to interpret because they are at increased risk of being artifactual and because few data have linked such mutations with an increased risk of virological failure to contemporary ART regimens. Nonetheless, the genotypic resistance interpretation of HIVdb-NGS beta version currently does not consider the threshold at which a mutation is detected. Like the main HIVdb program, the purpose of HIVdb-NGS beta is educational with regard to its genotypic resistance interpretations and quality control analysis.

Drug display options

By default, results will be shown for checked ARVs. Use checkboxes for additional ARVs. ([select all ARVs](#), [revert to default](#))

NRTI: ☒ ABC ☒ AZT ☒ FTC ☒ 3TC ☒ TDF ☐ D4T ☐ DDI

NNRTI: ☒ DOR ☒ EFV ☒ ETR ☒ NVP ☒ RPV

INSTI: ☒ BIC ☒ DTG ☒ EVG ☒ RAL

PI: ☒ ATV/r ☒ DRV/r ☒ LPV/r ☐ FPV/r ☐ IDV/r ☐ NFV ☐ SQV/r ☐ TPV/r

Input sequence reads

Upload file(s):

[Choose File](#)

No file chosen

[Load Example Data](#)


Drag and drop [CodFreq/AAVF files](#) here

Upload here the codfreq file

[Reset](#)

[Analyze](#)

We can change the read depth threshold (let's get everything for now) and the mutation detection threshold (let's change that to 2%):



Stanford University
HIV DRUG RESISTANCE DATABASE
A curated public database to represent, store and analyze HIV drug resistance data.

HOME
GENOTYPE-RX
GENOTYPE-PHENO
GENOTYPE-CLINICAL
HIVDB PROGRAM

ABOUT HIVDB
SUPPORT HIVDB!

Sequence reads summary

Sequence includes PR:	99 codon positions (1 ... 99)	SDRMs	Read Coverage
Sequence includes RT:	560 codon positions (1 ... 560)		
Sequence includes IN:	288 codon positions (1 ... 288)		
Median read depth:	32		
Subtype:	C (4.93%)		
Read depth threshold:	(all)		
Mutation detection threshold:	2%		

Change here!

Explore a bit the data, here an explanation if you are not familiar with the HIV Stanford db:

Sequence reads summary

This section describes the regions of HIV-1 PR, RT, and/or IN encompassed in the uploaded CodFreq file. The Read Depth describes the median number of reads encompassing each position (read coverage). The subtype is assessed using the consensus of all reads at the 20% mutation detection threshold. The methods of subtyping is similar to that used by the main HIVdb program and is described [here](#).

The Minimal Read Depth drop down menu allows users to instruct the program to ignore positions with reads below the specified depth. It will also prompt the program to return a warning reporting the number of codons containing reads below the specified depth. If the drop down menu is not used, the minimal read depth used by the program will be 1,000.

The Mutation Detection Threshold allows the user to specify the proportion of reads that must contain an amino acid for it to appear on the report. By default, this is set to 20%.

The Read Coverage button at the upper-right displays a figure which illustrates the read coverage across the pol regions encompassed by the CodFreq file.

Clicking on the SDRM button results in the addition of rows to the Sequence Read Summary

listing the surveillance DRMs for each gene included in the CFT. The complete list of PR, RT, and IN SDRMs and can be found on these pages: PR/RT, IN.

Multi-threshold mutation summary table

This section contains a table with the following columns:

- Mutation detection threshold: 20%, 10%, 5%, 2%, 1%, 0.5%, 0.2%, and 0.1%
- usual mutations: number of amino acid differences from the consensus B sequence excluding unusual mutations.
- unusual mutations: number of amino acids with a prevalence $<0.01\%$ in published direct PCR dideoxynucleotide (Sanger) group M sequences in HIVDB. Mutations that are known drug-resistance mutations (i.e., have a mutation penalty score) are excluded from this list even if their prevalence is $<0.01\%$. The procedure for identifying unusual mutations is defined here.
- DRMs: number of drug-resistance mutations defined as any mutation with a mutation penalty score (including polymorphic mutations and those with low scores).
- signature APOBEC mutations: # mutations suggestive of G-to-A hypermutation defined according to a procedure outlined here. The presence of ≥ 3 signature APOBEC DRMs in a pol sequence
- APOBEC-context DRMs: DRMs that could result from the activity of APOBEC-mediated G-to-A hypermutations. Mutation detection thresholds with large numbers of unusual or signature APOBEC mutations are likely too low and pose an unacceptable risk of identifying artifactual mutations caused either by machine error, PCR error, or G-to-A hypermutation. Genotypic resistance interpretations are suppressed for the 0.1% and 0.2% thresholds and for thresholds for which $>1\%$ of positions have an unusual mutation. Genotypic resistance interpretations are currently not suppressed for samples containing ≥ 3 signature APOBEC DRMs. However, APOBEC-context DRMs present at the same threshold are at risk of resulting from APOBEC-mediated G-to- A hypermutation and are likely of questionable clinical significance.

Warnings

Warnings appear to the right of the Mutation Statistics Table.

Low abundance mutations

The graph button at the upper-right produces a histogram showing the prevalence of all variants present in $<20\%$ of sequence reads. Grey histograms indicate usual mutations and red histograms indicate unusual mutations. A blue dot above the histogram indicates a drug-resistance mutation. The same mutation may be present below more than one histogram if it was encoded for by more than one codon. .

The spreadsheet button at the upper-right allows the user to download a spreadsheet listing all low-abundance codons in descending order of their prevalence within the NGS reads.

The spreadsheet contains 13 columns:

- Gene
- Amino acid position
- Number of reads for the position
- Sequenced codon
- Number of reads containing the codon
- The codon's amino acid translation
- The % of reads containing the codon
- The prevalence of the amino acid in HIVDB group M Sanger sequences
- The prevalence of the codon in HIVDB group M Sanger sequences
- DRM: Yes vs. empty field
- Unusual: Yes vs. empty field
- Signature APOBEC: Yes vs. empty field
- APOBEC-context DRM: Yes vs. empty field

Gene-specific drug-resistance interpretations

Interpretations are provided for mutations present above the mutation detection threshold provided the selected threshold was $\geq 0.5\%$ and fewer than 1% of positions had an unusual mutation. The appropriate mutation detection threshold, however, cannot be identified with certainty in the absence of a sequencing protocol that uses unique molecular identifiers (UMIs) for each virus template.

Mutations detected at low thresholds are difficult to interpret because they are at increased risk of being artifactual and because few data have linked such mutations with an increased risk of virological failure in persons receiving contemporary ART regimens. Nonetheless, the genotypic resistance interpretation of HIVdb-NGS beta version currently does not consider the threshold at which a mutation is detected. Like the main HIVdb program, the purpose of HIVdb-NGS beta is educational with regard to its genotypic resistance interpretations and quality control analysis. The procedure for developing the drug-resistance interpretation system is outlined here.

If you have time..

You can compare the DRM you obtained from the consensus sequence and the codfreq table. You can also run other samples to get more comfortable with the process!

Just a note on DRM and Stanford HIV db

What we used in this session it's a very convenient tool to get DRM in HIV directly from fastq file. Please remember that doing your own mapping, quality control, assembly is preferable when you use the sequence for other things rather than just DRM detection. For DRM investigation, fastq2codfreq maps to only a type of HIV missing a lot of information.

What about non-HIV DRM?

HIV is a well studied virus so lots of tools have been created to explore drug resistance mutations and analyse transmission. However, you might need to analyse other viruses which do not have these resources. What do you do? First don't despair! Second you can use the vcf file you learn to create. The steps are:

- map to the reference that provides information for DRM
- get the VCF file
- annotate the vcf file with a software like SnpEff <http://snpeff.sourceforge.net/SnpEff.html> or R biocoductor package Variant <https://bioconductor.org/packages/release/bioc/html/VariantAnnotation.html>
- look for the resistance mutations in your annotated file (you can write a quick R script to do the search)

Hint: if you are interested in HCMV (Human cytomegalovirus), talk to me! I have a cool pipeline doing exactly that! :)