snpToolkit

Mar 22, 2021

Contents

1 How to Install **3**

2 snpToolkit menu **5**

3 The explore command **7**

4 The annotate command **9**
4.1 Options . . . 9
4.2 Run it . . . 10
4.3 Outputs . . . 11

5 The viz command **15**
5.1 Visualize snptoolkit annotate command output files . . . 15

6 The combine command **19**
6.1 Options . . . 19
6.2 Running the combine command . . . 20
6.3 Find and Include Missing data . . . 22
6.4 Excluding SNPs . . . 26

7 The analyse command **29**
7.1 PCA . . . 30
7.2 UMAP . . . 31
7.3 Color mapping . . . 31

snpToolkit

snpToolkit is a computational framework written in Python 3. snpToolkit allows users to:

1. Visualize the content of their VCF files.
2. Filter SNPs based on multiple criteria:
    - Distance between SNPs
    - Coordinates of regions to exclude
    - Depth of coverage
    - Quality
    - The ratio corresponding to the number of reads that have the mutated allele / total number of reads at that particular position.
3. Annotate SNPs using genome annotation data provided within a genbank file.
4. Extract the distribution of all indels according to genome annotation.
5. Visualize and explore the annotated SNPs for all analyzed files.
6. Combine all snpToolkit output files generated using the annotate option and produce:
    - A table storing the distribution of all SNPs on each sample
    - A fasta file with all concatenated SNPs for each sample. such file can be used to build a phylogenetic tree.
7. Analyse your data using two dimentionality reduction methods: PCA and UMAP.

snpToolkit detects automatically if the input vcf files were generated using samtools mpileup, gatk HaplotypeCaller or freebayes. Vcf files can be in gzipped format or not.

CHAPTER 1

How to Install

The recommended way to install the most recent stable version of snpToolkit is:

```
pip install snptoolkit
```

Different python libraries will be installed:

- Biopython
- pysam
- pandas
- plotly
- dash
- tqdm
- coloredlogs

Note: If already installed, use pip install snptoolkit –upgrade

Install from source code:

```
git clone https://github.com/Amine-Namouchi/snpToolkit
cd snpToolkit
pip install .
```

CHAPTER 2

snpToolkit menu

```
$ snptoolkit -h
 usage: snptoolkit [-h] {explore,annotate,combine,viz,analyse} ...

     snpToolkit can takes vcf files, as well as bam files (optional) as inputs. The␣
↪vcf files could be generated using samtools/bcftools, gatk HaplotypeCaller or␣
↪freeBayes.
     Please visit https://snptoolkit.readthedocs.io/en/latest/index.html for more␣
↪information.


 positional arguments:
   {explore,annotate,combine,viz,analyse}
                         commands
     explore             Explore your vcf files before annotation
     annotate            Annotate one or multiple vcf files
     combine             Identify polymorphic sites and create distribution table and␣
↪alignment file in fasta format
     viz                 visualize snptoolkit output files
     analyse             analyse your SNPs data

 optional arguments:
   -h, --help            show this help message and exit
```

CHAPTER 3

The explore command

```
snptoolkit explore -h
usage: snptoolkit explore [-h] -i IDENTIFIER

optional arguments:
  -h, --help     show this help message and exit

snpToolkit explore required options:
  -i IDENTIFIER  Provide the input vcf files
```

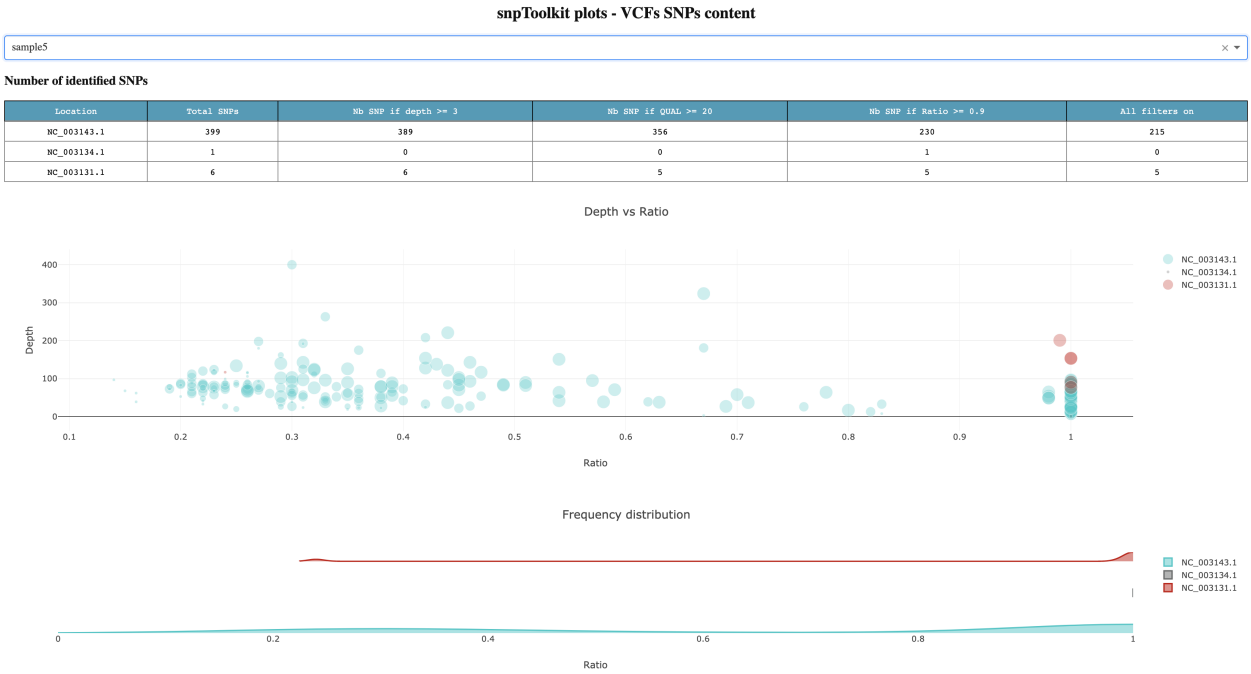
This command allows user to explore the SNPs on each of their vcf files.

The option -i allows to specify a common identifier in the vcf files names. If you want to explore all VCF files in a folder, you can use vcf as identifier as it is present in all vcf file names (usually filename.vcf.gz). On the contrary, if you have added in the filenames of your vcf files, for example, the years of isolation of each sample, you can use the year you want as identifier.

when you run the command:

```
$ snptoolkit explore -i vcf
[TIME][INFO] [snpToolkit is extracting your data and creating the different plots...]
progress: 100%|#######################################################################
↪##| 9/9 [00:00<00:00, 83.75it/s]
Dash is running on http://127.0.0.1:8050/

 * Serving Flask app "explore_snpToolkit" (lazy loading)
 * Environment: production
 * Running on http://127.0.0.1:8050/ (Press CTRL+C to quit)
```

snptoolkit will analyze all raw data on each VCF file in terms of SNPs and starts a web application that you access using the link mentioned above http://127.0.0.1:8050. For this example of 10 vcf files, it took less than a second to analyze all files. Figure 1 shows a screenshot of the generated dashboard to explore your data.



Location	Total SNPs	Nb SNP if depth >= 3	Nb SNP if QUAL >= 20	Nb SNP if Ratio >= 0.9	All filters on
NC_003143.1	399	389	356	230	215
NC_003134.1	1	0	0	1	0
NC_003131.1	6	6	5	5	5

Figure1

For sample 5 for example, we can see that the total number of SNPs in the chromosome NC_003143.1 is 399 SNPs. This is the total raw number. Lets detail each column of the first table:

- If we apply just the depth filter (-d) when using the option annotate (see below), only 10 SNPs will be excluded as they have a coverage less than 3.
- If we consider 20 as a cutoff for the quality of each SNPs, the number drop to 356 SNPs
- If we only consider those that have a ratio (nb reads with mutated allele/total number of reads on that position) 0.9, the number of SNPs drops to 230.
- If all filters are used: depth 3, QUAL 20 and ratio0.9, the number of filtered SNPs will be equal to 215.

For the case of *Yersinia pestis,* there are 3 plasmids. For sample 5, there are SNPs on plasmid NC_003134.1 and NC_003131.1

The first plot in Figure1 shows the distribution of all SNPs based on Ratio (x axis) and Depth (y axis). The size of each circle is proportional to the quality of each SNP. The second plot complement the first plot as it give you an idea about the proportion of SNPs for the chromosome and each of the plasmids. For the chromosome NC_004143.1, we can see that there is a small proportion of SNPs located between 0.2 and 0.4, but most of the SNPs has a high ratio 0.9.

To hide any of the data presented on each plot, you just need to select the name that you want.

CHAPTER 4

The annotate command

```
snptoolkit annotate -h
usage: snptoolkit annotate [-h] -i IDENTIFIER -g GENBANK [-p PROCESSORS] [-f␣
↪EXCLUDECLOSESNPS] [-q QUALITY] [-d DEPTH] [-r RATIO] [-e EXCLUDE]

optional arguments:
  -h, --help            show this help message and exit

snpToolkit annotate required options:
  -i IDENTIFIER         provide a specific identifier to recognize the file(s) to be␣
↪analyzed
  -g GENBANK            Pleae provide a genbank file

snpToolkit annotate additional options:
  -p PROCESSORS         number of vcf files to be annotated in parallel default value␣
↪[1]
  -f EXCLUDECLOSESNPS   exclude SNPs if the distance between them is lower then the␣
↪specified window size in bp
  -q QUALITY            quality score to consider as a cutoff for variant calling.␣
↪default value [20]
  -d DEPTH              minimum depth caverage. default value [3]
  -r RATIO              minimum ratio that correspond to the number of reads that has␣
↪the mutated allele / total depth in that particular position. default
                        value [0]
  -e EXCLUDE            provide a tab file with genomic regions to exclude in this␣
↪format: region start stop. region must correspond to the same name(s) of
                        chromsome and plasmids as in the genbank file
```

4.1 Options

This command allows to filter and annotate all SNPs from each selected VCF files. Only two options are required:

Option	Function
-i	The user need to specify a common identifier found on all VCF files he wants to analyze. If only one VCF file is to be analyzed, provide the file name. If all VCF files should be analyzed, the user needs to provide e.g vcf as all vcf files will have at the end .vcf.gz of .vcf.
-g	genbank file. The genbank file must include the fasta sequence for the chromosome and plasmids, if any. genbank files can be downloaded from NCBI.

Several options are additional and are needed to filter SNPs:

Option	Function
-f	To be able to exclude all SNPs that can be located in hotspot zones or short repeats, it is possible to specify an integer that will correspond to the minimum of distance between SNPs to be kept. if the distance between two SNPs is lower than the specified cutoff, both SNPs will be ignored.
-q	Quality score to consider as a cutoff for variant calling. The default value is 20.
-d	Minimum depth coverage. The default value is 3.
-r	r = M/M+R where **M** is the number of reads that carry the mutated allele and **R** is the is the number of reads that carry the reference allele. If not specified all SNPs will be taken into account.
-e	This is to specify a tab delimited file with the coordinates of the regions to be ignored when annotating SNPs.

If we take the example of the genbank used for this tutorial:

```
$ grep 'LOCUS' /Users/amine/Documents/tutorials/snptoolkit/GCF_000009065.1_
↪ASM906v1_genomic.gbff
LOCUS       NC_003143            4653728 bp    DNA     circular CON 20-MAR-
↪2020
LOCUS       NC_003131              70305 bp    DNA     circular CON 20-MAR-
↪2020
LOCUS       NC_003134              96210 bp    DNA     circular CON 20-MAR-
↪2020
LOCUS       NC_003132               9612 bp    DNA     circular CON 20-MAR-
↪2020
```

as you can see there is one chromosome NC_003143 and three plasmids: NC_003131, NC_003134 and NC_003132. The tab delimited file should look as follows:

```
NC_003143.1 4016    4079
NC_003143.1 7723    7758
NC_003143.1 11562   19149
NC_003143.1 25663   26698
```

If there are regions on the plasmids sequences you can also add them in the same file.

4.2 Run it

Now time to run the annotate command

```
$ snptoolkit annotate -i vcf -g GCF_000009065.1_ASM906v1_genomic.gbff -d 5 -q 30 -r 0.
↪9 -p 4
```

(continues on next page)

(continued from previous page)

```
[15:37:30] [INFO] [4 CPUs requested out of 8 detected on this machine]
[15:37:30] [INFO] [snpToolkit is filtering and annotating your SNPs]
100%|| 9/9 [00:01<00:00,  2.67it/s]
[15:37:32] [INFO] [snpToolkit output files will be located in folders
                 snpToolkit_SNPs_output_...
                 and snpToolkit_INDELS_output_...]
100%|| 9/9 [00:02<00:00,  3.95it/s]
```

4.3 Outputs

snpToolkit generates two folders with the date and time stamp, one for SNPs and one for indels:

```
├── snpToolkit_INDELS_output_...
│   ├── sample3_snpToolkit_indels.txt
│   ├── sample9_snpToolkit_indels.txt
│   ├── sample10_snpToolkit_indels.txt
│   ├── sample1_snpToolkit_indels.txt
│   ├── sample2_snpToolkit_indels.txt
│   ├── sample4_snpToolkit_indels.txt
│   ├── sample5_snpToolkit_indels.txt
│   ├── sample6_snpToolkit_indels.txt
│   ├── sample7_snpToolkit_indels.txt
│   └── sample8_snpToolkit_indels.txt
├── snpToolkit_SNPs_output_...
    ├── sample3_snpToolkit_SNPs.txt
    ├── sample9_snpToolkit_SNPs.txt
    ├── sample10_snpToolkit_SNPs.txt
    ├── sample1_snpToolkit_SNPs.txt
    ├── sample2_snpToolkit_SNPs.txt
    ├── sample4_snpToolkit_SNPs.txt
    ├── sample5_snpToolkit_SNPs.txt
    ├── sample6_snpToolkit_SNPs.txt
    ├── sample7_snpToolkit_SNPs.txt
    └── sample8_snpToolkit_SNPs.txt
```

All generated output files are tab delimited.

4.3.1 Example of SNP output file

```
##snpToolkit=__version__
##commandline= snptoolkit annotate -i vcf -g GCF_000009065.1_ASM906v1_genomic.gbff -d␣
↪5 -q 30 -r 0.9 -p 4
##VcfFile=sample5.vcf.gz
##Total number of SNPs before snpToolkit processing: 406
##The options -f and -e were not used
##Filtred SNPs. Among the 406 SNPs, the number of those with a quality score >= 30, a␣
↪depth >= 5 and a ratio >= 0.9 is: 218
##After mapping, SNPs were located in:
##NC_003131.1: Yersinia pestis CO92 plasmid pCD1, complete sequence 70305 bp
##NC_003143.1: Yersinia pestis CO92, complete genome 4653728 bp
##The mapped and annotated SNPs are distributed as follow:
##Location      Genes   RBS     tRNA    rRNA    ncRNA   Pseudogenes     intergenic     ␣
↪  Synonymous      NonSynonumous
```

(continues on next page)

(continued from previous page)

```
##SNPs in NC_003143.1: Yersinia pestis CO92, complete genome 4653728 bp 155     0
↪  0       1       0       0       57      54      101
##SNPs in NC_003131.1: Yersinia pestis CO92 plasmid pCD1, complete sequence 70305 bp
↪  2       0       0       0       0       0       3       1       1
##Syn=Synonymous NS=Non-Synonymous
##Coordinates   REF     SNP     Depth   Nb of reads REF Nb reads SNPs   Ratio
↪Quality Annotation      Product Orientation     Coordinates in gene     Ref codon
↪    SNP codon       Ref AA  SNP AA  Coordinates protein     Effect  Location
82      C       A       36      0       34      1.0     138.0   intergenic      .
↪  +       .       -       -       -       -       -       -       NC_003143.1:
↪Yersinia pestis CO92, complete genome 4653728 bp
130     G       C       28      0       27      1.0     144.0   intergenic      .
↪  +       .       -       -       -       -       -       -       NC_003143.1:
↪Yersinia pestis CO92, complete genome 4653728 bp
855     G       A       69      0       62      1.0     228.0   YPO_RS01010|asnC
↪  transcriptional regulator AsnC  -       411     ACC     AC[T]   T       T
↪137     Syn     NC_003143.1: Yersinia pestis CO92, complete genome 4653728 bp
```

The first lines of the snptoolkit file for SNPs contain a summary and useful information. The SNPs annotation is organized in tab delimited table. The columns of this table are:

Column name	Description
Coordinates	SNP coordinate
REF	Reference allele
SNP	New allele in analyzed sample
Depth	Total depth of coverage
Nb of reads REF	Number of reads with the reference allele
Nb reads SNPs	Number of reads with the new allele
Ratio	Nb reads SNPs/(Nb of reads REF+Nb reads SNPs)
Quality	Quality score
Annotation	Distribution within genes or intergenic
Product	Functional product of the gene
Orientation	Gene orientation
Coordinates in gene	Coordinate of the SNP within the gene
Ref codon	Reference codon, ACC in the example above
SNP codon	New codon, AC[T]
Ref AA	Amino Acid corresponding to reference codon
SNP AA	Amino Acid corresponding to new codon
Coordinates protein	Coordinate of the Amino Acid
Effect	Could be Synonymous (Syn) or Non-Synonymous (NS)
Location	ID of the chromosome and plasmids.

Warning: In the example above, the total depth for the first SNP is 36, while the number of reads that carry the reference allele plus the number of reads that carry the new allele is equal to 34. The VCF file corresponding to that sample is generated using samtools mpileup. By default, samtools mpileup applies Phred-scaled probability of a read base being misaligned, known as BAQ. As indicated in samtools documentation, this greatly helps to reduce false SNPs caused by misalignments. The total depth shown by snpToolkit is the raw depth taking into account all reads (column 4). However, the columns 5 and 6 show the number of reads with Phred-scaled probability. The ratio in column 7 is based only on column 5 and 6. I have made this decision to store as much information as possible from the original VCF file. If the VCF files where produced using samtools-mpileup with the option -B to skip Phred-scaled probability, you will not see such difference.

4.3.2 Example of INDELS output file

The indels output is in tab delimited format as follows:

```
55732   CCGGGGCGGGGCGGGGCGG CCGGGGCGGGGCGG  62  0   20  1.0 228.0   intergenic  .   .
↪  deletion    5   NC_003131.1: Yersinia pestis CO92 plasmid pCD1, complete sequence
↪70305 bp
35188   T   TTC 41  0   32  1.0 228.0   intergenic  .   .   insertion   2   NC_003134.
↪1: Yersinia pestis CO92 plasmid pMT1, complete sequence 96210 bp
73418   GAA GA  72  0   68  1.0 228.0   intergenic  .   .   deletion    1   NC_003134.
↪1: Yersinia pestis CO92 plasmid pMT1, complete sequence 96210 bp
16  AC  A   13  0   13  1.0 149.0   intergenic  .   .   deletion    1   NC_003143.1:
↪Yersinia pestis CO92, complete genome 4653728 bp
183029  CCAATAACAAT CCAATAACAATAACAAT   95  0   24  1.0 228.0   intergenic  .   .
↪insertion   6   NC_003143.1: Yersinia pestis CO92, complete genome 4653728 bp
266466  AGGGGGGGG   AGGGGGGGGG  40  1   25  0.96    66.0    CDS YPO_RS02340|YPO_
↪RS02340 EscV/YscV/HrcV family type III secretion system export apparatus protein
↪insertion   1   NC_003143.1: Yersinia pestis CO92, complete genome 4653728 bp
552919  TGGGGGGG    TGGGGGGGG   93  0   71  1.0 122.0   CDS YPO_RS03585|tssM    type
↪VI secretion system membrane subunit TssM  insertion   1   NC_003143.1: Yersinia
↪pestis CO92, complete genome 4653728 bp
581519  GTTCAATTCAATTCAAT   GTTCAATTCAATTCAATTCAAT  31  0   9   1.0 228.0
↪intergenic  .   .   insertion   5   NC_003143.1: Yersinia pestis CO92, complete
↪genome 4653728 bp
747924  AGGGGGGGG   AGGGGGGGGG  41  1   26  0.96    71.0    CDS YPO_RS04395|YPO_
↪RS04395 pseudopilin insertion   1   NC_003143.1: Yersinia pestis CO92, complete
↪genome 4653728 bp
813977  GC  GCCTGGCCATC 54  0   10  1.0 228.0   CDS YPO_RS04755|YPO_RS04755 DASS
↪family sodium-coupled anion symporter  insertion   9   NC_003143.1: Yersinia pestis
↪CO92, complete genome 4653728 bp
```

for the case of the position 266466 for example

```
266466  AGGGGGGGG   AGGGGGGGGG  40  1   25  0.96    66.0    CDS YPO_RS02340|YPO_
↪RS02340 EscV/YscV/HrcV family type III secretion system export apparatus protein
↪insertion   1   NC_003143.1: Yersinia pestis CO92, complete genome 4653728 bp
```

The different columns are:

Column number	Description
1	Coordinates (266466)
2	Reference (AGGGGGGGG)
3	Sample (AGGGGGGGGG)
4	Number of total reads (40)
5	Number of reads with reference sequence (1)
6	Number of reads with new sequence (25)
7	Ratio (0.96)
8	Quality score (66.0)
9	Location (CDS)
10	Gene or intergenic (YPO_RS02340|YPO_RS02340)
11	Gene product (EscV/YscV/HrcV family type III secretion system export apparatus protein)
12	Type of indel (insertion)
13	Number of nucleotide (1)
14	Sequence name (NC_003143.1: Yersinia pestis CO92, complete genome 4653728 bp)

Note: While snpToolkit annotate indels, it is important to be careful and check any indels you are interested in before to elaborate any hypothesis and conclusions.

CHAPTER 5

The viz command

```
$ snptoolkit viz -h
usage: snptoolkit viz [-h] [--dir DIRECTORY] [-p POLYMORPHIC_SITES] [-conf CONFIG]

optional arguments:
-h, --help            show this help message and exit

snpToolkit viz required options:
--dir DIRECTORY       provide the path of the directory containing snptoolkit SNPs␣
↪output files
-p POLYMORPHIC_SITES  provide the path of the polymorphic sites you want to analyze
-conf CONFIG          provide the path of the configuration file that contains the␣
↪information to use for data visualization
```

5.1 Visualize snptoolkit annotate command output files

```
 ── snpToolkit_SNPs_output_...
    ├── sample3_snpToolkit_SNPs.txt
    ├── sample9_snpToolkit_SNPs.txt
    ├── sample10_snpToolkit_SNPs.txt
    ├── sample1_snpToolkit_SNPs.txt
    ├── sample2_snpToolkit_SNPs.txt
    ├── sample4_snpToolkit_SNPs.txt
    ├── sample5_snpToolkit_SNPs.txt
    ├── sample6_snpToolkit_SNPs.txt
    ├── sample7_snpToolkit_SNPs.txt
    └── sample8_snpToolkit_SNPs.txt

$ snptoolkit viz --dir snpToolkit_SNPs_output_..
Dash is running on http://127.0.0.1:8050/

 * Serving Flask app "plot_snpToolkit_output" (lazy loading)
```

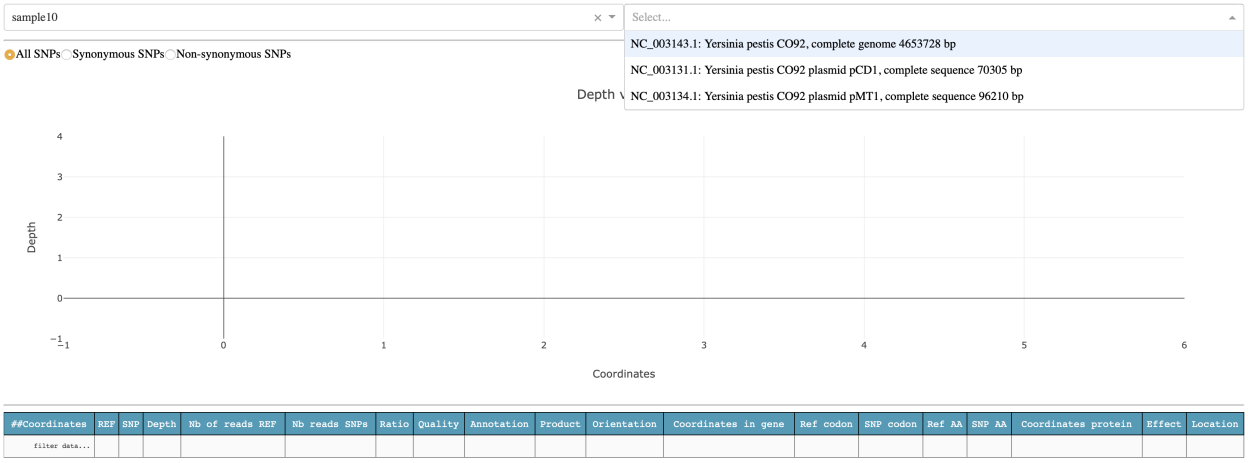
(continues on next page)

(continued from previous page)

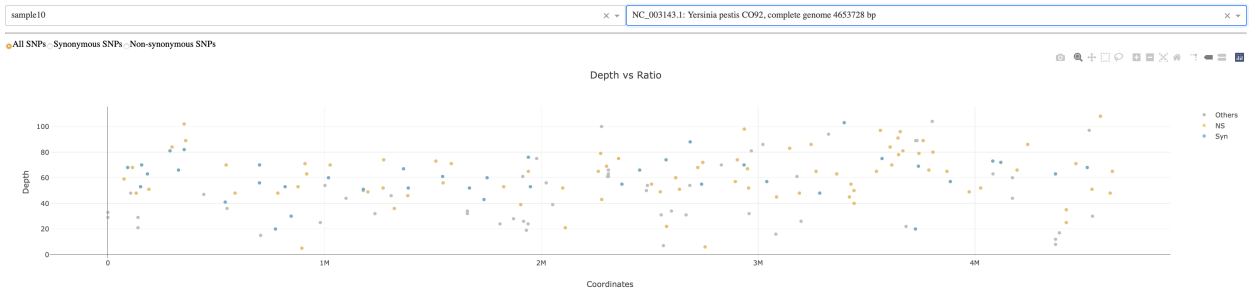
```
 * Environment: production
 * Debug mode: off
 * Running on http://127.0.0.1:8050/ (Press CTRL+C to quit)
```

To visualize all snpToolkit outputs, just access the url http://127.0.0.1:8050/.

The first time, you will see one sample selected, in this case sample10, and nothing in the plot and table below. Before to see anything you will need to select for which sequence you want to display the result. For sample10, SNPs where found in the chromosome of *Yersinis pestis* NC_003143.1 and two plasmids: NC_003131.1 and NC_003134.1



lets select the chromosome NC_003143.1



##Coordinates	REF	SNP	Ref codon	SNP codon	Ref AA	SNP AA	Effect	Annotation	Product	Orientation	Coordinates in gene	Coordinates protein	Depth	Nb of reads REF	Nb reads SNPs	Ratio	Quality
filter data...																	
82	C	A	-	-	-	-	-	intergenic	.	+	.	-	29	0	27	1	140
130	G	C	-	-	-	-	-	intergenic	.	+	.	-	33	0	29	1	148
74539	C	T	GCC	[A]CC	A	T	NS	YPO_RS01350|envC	murein hydrolase activator EnvC	-	361	121	59	0	47	1	228
91494	G	A	AAC	AA[T]	N	N	Syn	YPO_RS01440|YPO_RS01440	molybdate ABC transporter substrate-binding protein	-	15	5	68	0	56	1	228
105187	A	C	.	.	.	.		intergenic	.	+	.	.	48	0	46	1	146
113589	A	C	ATC	A[G]C	I	S	NS	YPO_RS01555|hslU	HslU--HslV peptidase ATPase subunit	-	956	319	68	0	63	1	228
130643	G	A	ACC	A[T]C	T	I	NS	YPO_RS01625|glpE	thiosulfate sulfurtransferase GlpE	-	20	7	48	0	47	1	228
138787	A	C	.	.	.	.		intergenic	.	+	.	.	29	0	28	1	145
138810	G	A	.	.	.	.		intergenic	.	+	.	.	21	0	19	1	101
150946	C	A	ACG	AC[T]	T	T	Syn	YPO_RS01705|pckA	phosphoenolpyruvate carboxykinase (ATP)	-	1572	524	53	0	46	1	228
155747	A	G	AGT	AG[C]	S	S	Syn	YPO_RS01725|YPO_RS01725	intracellular growth attenuator family protein	-	1419	473	70	0	65	1	228
182447	C	T	ATC	AT[T]	I	I	Syn	YPO_RS01830|YPO_RS01830	TonB-dependent copper receptor	+	1542	514	63	0	53	1	228
189227	C	T	CGT	[T]GT	R	C	NS	YPO_RS01855|YPO_RS01855	aminodeoxychorismate synthase component II	+	418	140	51	0	46	1	228
286528	T	A	CGA	CG[T]	R	R	Syn	YPO_RS02445|hutX	heme utilization cystosolic carrier protein HutX	-	102	34	81	0	76	1	228
296089	G	T	GAG	GA[T]	E	D	NS	YPO_RS02490|YPO_RS02490	tellurium resistance protein TerZ	+	453	151	84	0	73	1	228
325836	T	C	CCA	CC[G]	P	P	Syn	YPO_RS02605|YPO_RS02605	quinone oxidoreductase	-	330	110	66	0	63	1	228
351821	T	G	CAT	CA[G]	H	Q	NS	YPO_RS02710|aegA	formate-dependent uric acid utilization protein AegA	+	1851	617	102	0	96	1	228
351956	G	A	GAG	GA[A]	E	E	Syn	YPO_RS02710|aegA	formate-dependent uric acid utilization protein AegA	+	1986	662	82	0	67	1	228
358875	A	G	TTG	T[C]G	L	S	NS	YPO_RS02740|aspA	aspartate ammonia-lyase	-	1088	363	89	0	80	1	228
442439	T	C	.	.	.	.		intergenic	.	+	.	.	47	0	38	1	228

1 / 9

The plot shows the genomic distribution of all SNPs according to depth. By default, all SNPs are shown but you can select to visualize only Non-synonymous (orange), Synonymous (blue) and intergenic SNPs (grey). The table below the plot shows all relevant information retrieved from snptoolkit output file for each sample.

It is possible to filter the table using keywords on each column. In the example below, I used the keyword "transporter" in the column Product.

##Coordinates	REF	SNP	Ref codon	SNP codon	Ref AA	SNP AA	Effect	Annotation	Product	Orientation	Coordinates in gene	Coordinates protein	Depth	Nb of reads REF	Nb reads SNPs	Ratio	Quality
filter data...									transporter								
91494	G	A	AAC	AA[T]	N	N	Syn	YPO_RS01440|YPO_RS01440	molybdate ABC transporter substrate-binding protein	-	15	5	68	0	56	1	228
917155	A	G	AGT	[G]GT	S	G	NS	YPO_RS05115|YPO_RS05115	PTS sugar transporter subunit IIA	+	178	60	63	0	52	1	228
1512930	A	G	ACG	[G]CG	T	A	NS	YPO_RS07725|YPO_RS07725	DMT family transporter	+	544	182	73	0	69	1	228
1904871	T	C	ATC	A[C]C	I	T	NS	YPO_RS09340|yapJ	autotransporter outer membrane beta-barrel domain-containing protein	+	2072	691	39	0	34	1	134
2548551	G	T	GCT	G[A]T	A	D	NS	YPO_RS12325|YPO_RS12325	MFS transporter	-	590	197	49	0	46	1	228
2739149	C	A	ACC	AC[A]	T	T	Syn	YPO_RS13205|yfeA	iron/manganese ABC transporter substrate-binding protein YfeA	+	408	136	55	1	33	0.97	228
2903882	T	G	TGT	[G]GT	C	G	NS	YPO_RS13865|YPO_RS13865	sugar ABC transporter ATP-binding protein	+	1237	413	74	0	67	1	228
2934972	C	G	ACG	AC[C]	T	T	Syn	YPO_RS14020|YPO_RS14020	amino acid ABC transporter ATP-binding protein	-	126	42	70	0	68	1	228
2936268	G	A	CTT	[T]TT	L	F	NS	YPO_RS14030|YPO_RS14030	amino acid ABC transporter permease	-	244	82	98	1	84	0.99	228
3244204	A	G	GTC	G[C]C	V	A	NS	YPO_RS15475|YPO_RS15475	nickel/cobalt transporter	-	203	68	86	0	80	1	228
3397040	A	G	AAA	AA[G]	K	K	Syn	YPO_RS16190|acrD	multidrug efflux RND transporter permease AcrD	+	576	192	103	0	92	1	228
3787622	T	A	CAG	C[T]G	Q	L	NS	YPO_RS17930|fhuC	Fe3+-hydroxamate ABC transporter ATP-binding protein FhuC	-	458	153	66	0	56	1	228
4518401	G	A	AAC	AA[T]	N	N	Syn	YPO_RS21150|YPO_RS21150	MFS transporter	-	810	270	68	0	63	1	228
4540504	T	C	GTC	G[C]C	V	A	NS	YPO_RS21235|YPO_RS21235	iron chelate uptake ABC transporter family permease subunit	+	890	297	51	0	50	1	228

CHAPTER 6

The combine command

```
$ snptoolkit combine  -h
usage: snptoolkit combine [-h] --loc LOCATION [-r RATIO] [--bam BAMFILTER BAMFILTER␣
↪BAMFILTER] [--snps {ns,s,all,inter}] [-e EXCLUDE]

optional arguments:
  -h, --help            show this help message and exit

snpToolkit combine required options:
  --loc LOCATION   provide for example the name of the chromosome or plasmid you want␣
↪to create fasta alignemnt for

snpToolkit additional options:
  -r RATIO              new versus reference allele ratio to filter SNPs from␣
↪snpToolkit outputs. default [0]
  --bam BAMFILTER BAMFILTER BAMFILTER
                        provide the depth, ratio and the path to the folder␣
↪containing the bam files. eg. 3 0.9 path
  --snps {ns,s,all,inter}
                        Specify if you want to concatenate all SNPs or just␣
↪synonymous (s), non-synonymous (ns) or intergenic (inter) SNPs. default [all]
  -e EXCLUDE            Provide a yaml file with keywords and coordinates to be␣
↪excluded
```

6.1 Options

Option	Description
–loc	The name of chromosome or plasmid you want to concatenate the SNPs for. This can be found in the last coloumn of the output file of the annotate command

Several options are additional:

Option	Description
–bam	This option takes three parameters in the following order: depth ration path_to_bam_files. See below for more details
–snps	Type of SNPs to be concatenated. default [all]
-r	r = M/M+R where **M** is the number of reads that carry the mutated allele and **R** is the is the number of reads that carry the reference allele. If not specified all SNPs will be taken into account.
-e	This is to specify a yaml file with two arguments KEYWORDS and COORDINATES. See below for more details

The command combine should be run in the directory containing the snpToolkit output files generated using the annotate command.

```
├── sample1_snpToolkit_SNPs.txt
├── sample3_snpToolkit_SNPs.txt
├── sample5_snpToolkit_SNPs.txt
├── sample9_snpToolkit_SNPs.txt
├── sample10_snpToolkit_SNPs.txt
├── sample2_snpToolkit_SNPs.txt
├── sample4_snpToolkit_SNPs.txt
├── sample6_snpToolkit_SNPs.txt
├── sample7_snpToolkit_SNPs.txt
├── sample8_snpToolkit_SNPs.txt
```

6.2 Running the combine command

```
$ snptoolkit combine  --loc NC_003143.1
[09:33:38] [INFO] [Searching for polymorphic sites...]
[09:33:38] [INFO] [SNPs polymorphic sites distribution. Please wait...]
progress: 100%|#################################################################
↪#######| 490/490 [00:00<00:00, 20233.61it/s]
[09:33:38] [INFO] [Creating SNPs_polymorphic_sites.txt]
[09:33:38] [INFO] [Creating SNPs_alignment.fasta]
progress: 100%|#################################################################
↪#######| 10/10 [00:00<00:00, 4712.17it/s]
```

In the command above, the first step is to search for all polymorphic sites. A total number of 490 SNPs was found. As we didnt use the -r option, all SNPs were analyzed. The minimum ratio in this case will be 0.9

Note: It is important to remember that these snpToolkit output files were generated with the following command:

snptoolkit annotate -i vcf -g GCF_000009065.1_ASM906v1_genomic.gbff -d 5 -q 30 -r 0.9 -p 4

all annotated SNPs will have at least a depth of 5 and a ratio of 0.9.

Now lets run the command above with the option -r 1

```
$ snptoolkit combine  --loc NC_003143.1 -r 1
[10:06:27] [WARNING] [SNPs_polymorphic_sites.txt exists already and was created on␣
↪Thu Oct 22 09:44:43 2020. This file will be replaced.
                                          Press any key to continue or ctrl-c to␣
↪exit!]
```

(continues on next page)

(continued from previous page)

```
[10:06:28] [INFO] [Searching for polymorphic sites...]
[10:06:28] [INFO] [SNPs polymorphic sites distribution. Please wait...]
progress: 100%|#####################################################################
↪##########| 470/470 [00:00<00:00, 22782.49it/s]
[10:06:28] [INFO] [Creating SNPs_polymorphic_sites.txt]
[10:06:28] [INFO] [Creating SNPs_alignment.fasta]
progress: 100%|#####################################################################
↪##########| 10/10 [00:00<00:00, 9736.08it/s]
```

As the file SNPs_polymorphic_sites.txt exists already, snpToolkit warn you that you need to change the file name or it will be replaced by the new output file.

As we requested that for all SNPs, 100% of the reads must have the new allele, the number of polymorphic sites is now 470.

The Polymorphic sites output SNPs_polymorphic_sites.txt is as follows:

```
##snpToolkit=version
##commandline= snptoolkit combine --loc NC_003143.1 -r 1
##location=NC_003143.1
##Number of polymorphic sites= 470
##ID    Coordinates     REF     SNP     Location        Product Orientation
↪NucPosition     REF-codon       NEW-codon       REF-AA  NEW-AA  ProPostion
↪Type    sample10        sample9 sample8 sample7 sample6 sample5 sample4 sample2
↪sample3 sample1
snp1    82      C       A       intergenic      .       +       .       -       -
↪  -       -       -       -       1       1       1       1       1       1       1
↪       1       1       1
snp2    130     G       C       intergenic      .       +       .       -       -
↪  -       -       -       -       1       1       1       1       1       1       1
↪       1       1       1
snp3    855     G       A       YPO_RS01010|asnC        transcriptional regulator
↪AsnC  -       411     ACC     AC[T]   T       T       137     Syn     0       0
↪  0       0       1       1       1       0       0       0
snp4    18061   C       T       YPO_RS01090|YPO_RS01090 IS256 family transposase
↪  +       156     AAC     AA[T]   N       N       52      Syn     0       0       1
↪       1       1       1       1       0       0       0
snp5    21219   C       A       YPO_RS01110|YPO_RS01110 serine/threonine protein
↪kinase +       428     GCC     G[A]C   A       D       143     NS      0       0
↪   0       0       0       1       0       0       0       0
snp6    42303   C       T       YPO_RS01190|fabY        fatty acid biosynthesis
↪protein FabY   +       897     GTC     GT[T]   V       V       299     Syn     0
↪    0       0       0       0       0       0       0       1       0
snp7    61685   G       C       intergenic      .       +       64 bp from YPO_
↪RS01280|YPO_RS01280    .       .       .       .       .       .       0       0
↪     0       0       0       0       1       0       0       0
snp8    74539   C       T       YPO_RS01350|envC        murein hydrolase activator
↪EnvC -       361     GCC     [A]CC   A       T       121     NS      1       1
↪ 1       1       1       1       1       1       1       1
snp9    76590   C       T       intergenic      .       +       .       -       -
↪  -       -       -       -       0       0       0       0       0       0       0
↪       0       1       0
snp10   90931   T       A       YPO_RS01440|YPO_RS01440 molybdate ABC transporter
↪substrate-binding protein      -       578     CAG     C[T]G   Q       L       193
↪  NS      0       0       1       1       0       0       0       0       0       0
```

The first lines of this file contain a summary and useful information. The SNPs annotation is organized in tab delimited table. The columns of this table are:

Column name	Description
ID	Identifiier of the SNP
Coordinates	SNP coordinate
REF	Reference allele
SNP	New allele in analyzed sample
Locatio	location within the genome
Product	Functional product of the gene
Orientation	Gene orientation
NucPosition	gene Coordinate of the SNP within the gene
REF-codon	Reference codon
NEW-codon	New codon
Ref AA	Amino Acid corresponding to reference codon
SNP AA	Amino Acid corresponding to new codon
ProPostion	Coordinate of the Amino Acid
Type	Could be Synonymous (Syn) or Non-Synonymous (NS), or (.) for intergenic

After these columns, each column will represented one analyzed sample. The presence or absence of each SNP is represented by 1 or 0, respectively.

In addition to the SNPs_polymorphic_sites.txt, snpToolkit will also generates a fasta file SNPs_alignment.fasta containing the concatenation of all polymorphic sites on each sample.

```
$ grep '>' SNPs_alignment.fasta
>NC_003143.1
>sample10
>sample9
>sample8
>sample7
>sample6
>sample5
>sample4
>sample2
>sample3
>sample1
```

The first sequence is the reference sequence followed by the 10 samples used for this example

6.3 Find and Include Missing data

Lets now suppose that we have two ancient DNA samples that we have analyzed and generated the corresponding vcf files. When working with aDNA, usually not 100% of your genome is recovered. When looking for the distribution of all polymorphic sites within these aDNA, it is important to know if an SNP was not identified because for that position the aDNA is similar to the reference or because the region is not covered at all. To be able to identify such position, users have to provide the bam files of all samples for whom they want to account for missing data.

```
snptoolkit combine -r 0.9 --loc NC_003143.1 --bam 2 1.0 ../bam/
```

As you can see, you need just to specify one addition option '- -bam' with three parameter

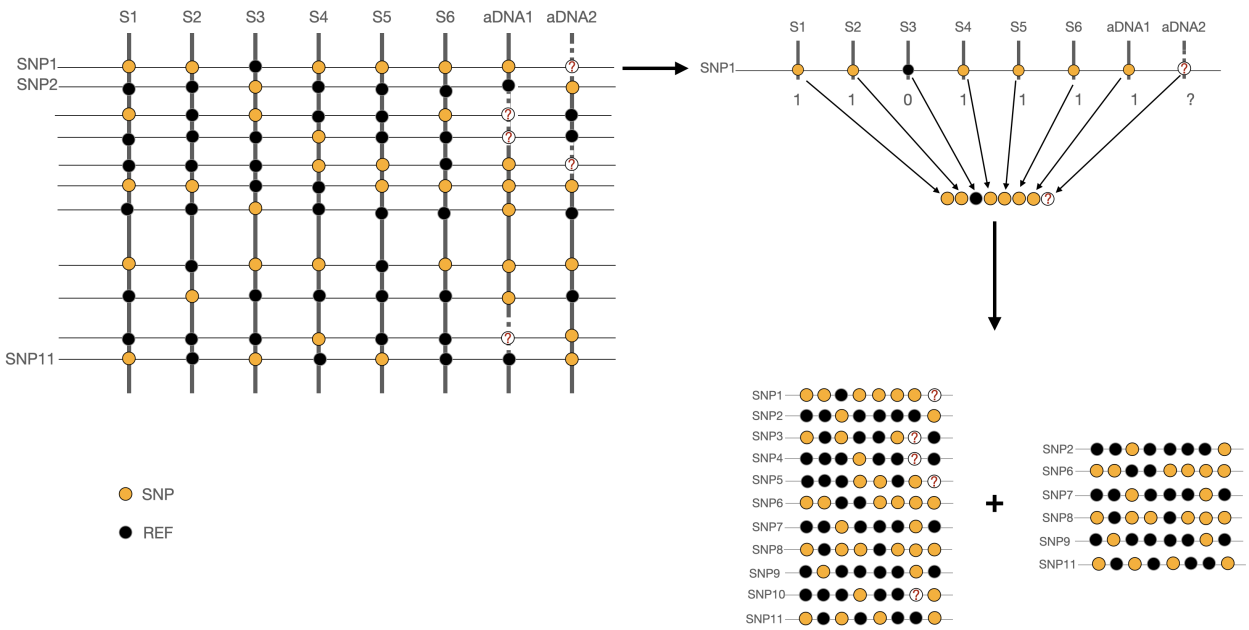
```
--bam 2 1.0 ../bam/
```

Parameter	Description
2	Minimum depth of coverage to consider. Here this depth is set to 2, which mean that at least 2 reads should be found
1.0	Number of reads with new allele / total number of reads ratio. For this example it is set to 1.0, which mean that the 2 reads should have the new allele
../bam	This is the path of the folder containing the bam files of all aDNA to be considered. The bam folder should also include the .bai files

Note: When using the –bam option, snptoolkit will create in total 4 output files:

- SNPs_polymorphic_sites.txt + SNPs_alignment.fasta
- SNPs_polymorphic_sites_clean.txt + SNPs_alignment.fasta

As described above, The first two files contains all SNPs found in all analysed samples including polymorphic sites where in some samples there is missing information indicated by a question mark. The second two files are a "clean" version of the two files described above in the sence that they don't contain any position where missing information is reported.




```
├── [ 19M ]  sampleY.bam
├── [ 33M ]  sampleX.bam
├── [ 14K ]  sampleX.bam.bai
└── [6.6K ]  sampleY.bam.bai
```

Now lets run the combine command on all snpToolkit output files generated using the annotate command:

```
├── [ 10K ]  sample9_snpToolkit_SNPs.txt
├── [ 10K ]  sampleY_snpToolkit_SNPs.txt
├── [ 32K ]  sample10_snpToolkit_SNPs.txt
├── [ 15K ]  sample1_snpToolkit_SNPs.txt
```

(continues on next page)

(continued from previous page)

```
├── [ 15K ]  sample2_snpToolkit_SNPs.txt
├── [ 12K ]  sample3_snpToolkit_SNPs.txt
├── [ 35K ]  sample4_snpToolkit_SNPs.txt
├── [ 36K ]  sample5_snpToolkit_SNPs.txt
├── [ 38K ]  sample6_snpToolkit_SNPs.txt
├── [ 37K ]  sample7_snpToolkit_SNPs.txt
├── [ 16K ]  sampleX_snpToolkit_SNPs.txt
├── [ 41K ]  sample8_snpToolkit_SNPs.txt
```

```
$ snptoolkit combine -r 0.9 --loc NC_003143.1 --bam 2 1.0 ../bam/
[10:45:48] [INFO] [Searching for polymorphic sites...]
[10:45:48] [INFO] [SNPs polymorphic sites distribution. Please wait...]
progress: 100%|#######################################################################
↪#######################################################################################
↪#################################| 505/505 [00:04<00:00, 112.91it/s]
[10:45:52] [INFO] [Creating SNPs_alignment.fasta]
progress: 100%|#######################################################################
↪#######################################################################################
↪#################################| 12/12 [00:00<00:00, 8558.35it/s]
[10:45:52] [INFO] [Creating SNPs_polymorphic_sites_clean.txt]
progress: 100%|#######################################################################
↪#######################################################################################
↪#################################| 375/375 [00:00<00:00, 183381.60it/s]
[10:45:52] [INFO] [Creating SNPs_alignment_clean.fasta]
progress: 100%|#######################################################################
↪#######################################################################################
↪#################################| 12/12 [00:00<00:00, 12738.96it/s]
```

By adding the two aDNA samples, the number of polymorphic sites has increased to 505. The new SNPs_polymorphic_sites.txt contains now the SNPs distribution for sampleX and sampleY.

```
##commandline= snptoolkit combine -r 0.9 --loc NC_003143.1 --bam 2 1.0 ../bam/
##location=NC_003143.1
##Number of polymorphic sites= 505
##ID    Coordinates     REF     SNP     Location        Product Orientation
↪NucPosition     REF-codon       NEW-codon       REF-AA  NEW-AA  ProPostion
↪Type    sample10        sampleX sampleY sample9 sample8 sample7 sample6 sample5
↪sample4 sample2 sample3 sample1
snp1    82      C       A       intergenic      .       +       .       -       -
↪  -       -       -       -       1       1       1       1       1       1       1
↪      1       1       1       1       1
snp2    130     G       C       intergenic      .       +       .       -       -
↪  -       -       -       -       1       1       1       1       1       1       1
↪      1       1       1       1       1
snp3    855     G       A       YPO_RS01010|asnC        transcriptional regulator
↪AsnC  -       411     ACC     AC[T]   T       T       137     Syn     0       0
↪  0       0       0       0       1       1       1       0       0       0
snp4    18061   C       T       YPO_RS01090|YPO_RS01090 IS256 family transposase
↪  +       156     AAC     AA[T]   N       N       52      Syn     0       ?       ?
↪      0       1       1       1       1       1       0       0       0
snp5    21219   C       A       YPO_RS01110|YPO_RS01110 serine/threonine protein
↪kinase +       428     GCC     G[A]C   A       D       143     NS      0       0
↪   0       0       0       0       0       1       0       0       0       0
snp6    29368   G       T       YPO_RS01140|hemN        oxygen-independent
↪coproporphyrinogen III oxidase         +       387     GTG     GT[T]   V       V
↪ 129     Syn     0       0       1       0       0       0       0       0       0
↪      0       0       0
```

(continues on next page)

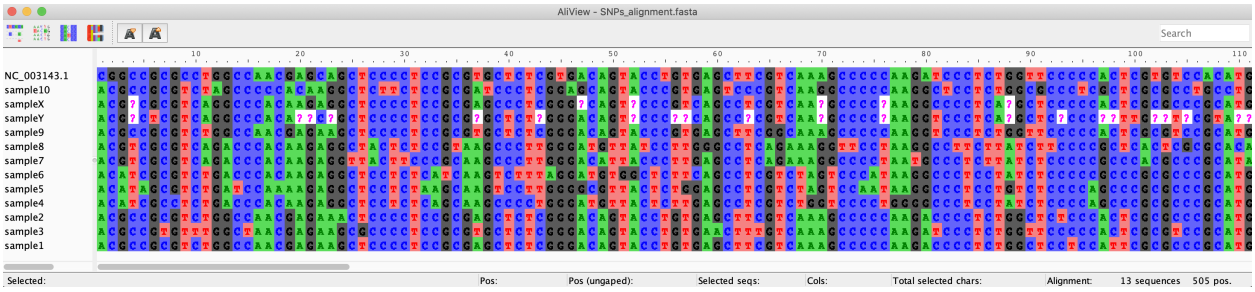
(continued from previous page)

```
snp7    42303   C       T       YPO_RS01190|fabY        fatty acid biosynthesis␣
↪protein FabY    +       897     GTC     GT[T]   V       V       299     Syn     0  ␣
↪    0       0       0       0       0       0       0       0       0       1     ␣
↪0
snp8    61685   G       C       intergenic      .       +       64 bp from YPO_
↪RS01280|YPO_RS01280      .       .       .       .       .       .       0       0 ␣
↪    0       0       0       0       0       0       1       0       0       0
snp9    74539   C       T       YPO_RS01350|envC        murein hydrolase activator␣
↪EnvC -       361     GCC     [A]CC   A       T       121     NS      1       1     ␣
↪ 1       1       1       1       1       1       1       1       1       1
```

For snp4, this SNP is considered as "?" as at position 18061 the criteria minimum 2 reads AND ratio 1.0 were not satisfied

```
##ID    Coordinates     REF     SNP     Location        Product Orientation     ␣
↪NucPosition     REF-codon       NEW-codon       REF-AA  NEW-AA  ProPostion     ␣
↪Type    sample10        sampleX sampleY sample9 sample8 sample7 sample6 sample5␣
↪sample4 sample2 sample3 sample1
snp4    18061   C       T       YPO_RS01090|YPO_RS01090 IS256 family transposase  ␣
↪  +       156     AAC     AA[T]   N       N       52      Syn     0       ?       ?␣
↪      0       1       1       1       1       1       0       0       0
```

Lets take a look now at the file SNPs_alignment.fasta:



The file SNPs_polymorphic_sites_clean.txt contains only 375 SNPs instead of 505 as 130 polymorphic sites contain missing information.

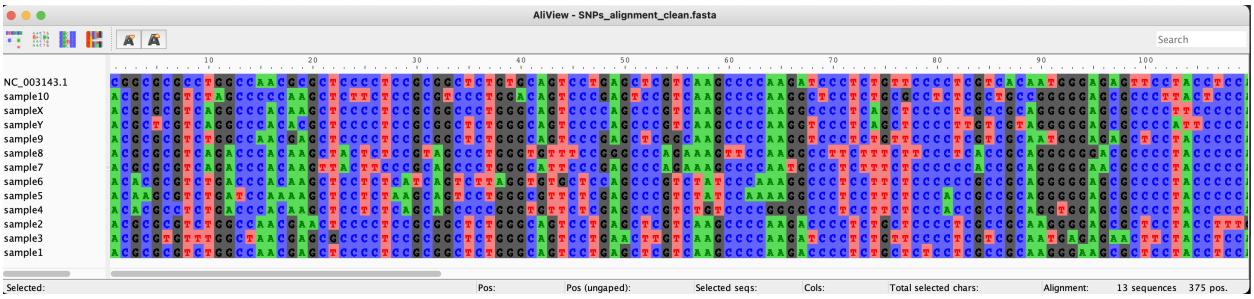
```
##commandline= snptoolkit combine -r 0.9 --loc NC_003143.1 --bam 2 1.0 ../bam/
##location=NC_003143.1
##Number of polymorphic sites= 375
##ID    Coordinates     REF     SNP     Location        Product Orientation     ␣
↪NucPosition     REF-codon       NEW-codon       REF-AA  NEW-AA  ProPostion     ␣
↪Type    sample10        sampleX sampleY sample9 sample8 sample7 sample6 sample5␣
↪sample4 sample2 sample3 sample1
snp1    82      C       A       intergenic      .       +       .       -       -  ␣
↪  -       -       -       -       1       1       1       1       1       1       1␣
↪      1       1       1       1       1
snp2    130     G       C       intergenic      .       +       .       -       -  ␣
↪  -       -       -       -       1       1       1       1       1       1       1␣
↪      1       1       1       1       1
snp3    855     G       A       YPO_RS01010|asnC        transcriptional regulator␣
↪AsnC  -       411     ACC     AC[T]   T       T       137     Syn     0       0  ␣
↪  0       0       0       0       1       1       1       0       0 0
snp4    21219   C       A       YPO_RS01110|YPO_RS01110 serine/threonine protein␣
↪kinase +       428     GCC     G[A]C   A       D       143     NS      0       0  ␣
↪    0       0       0       0       0       1       0       0       0 0
```

(continues on next page)

(continued from previous page)

```
snp5    29368   G       T       YPO_RS01140|hemN        oxygen-independent␣
↪coproporphyrinogen III oxidase       +       387     GTG     GT[T]   V       V       ␣
↪ 129     Syn     0       0       1       0       0       0       0       0 0       ␣
↪0       0       0
snp6    42303   C       T       YPO_RS01190|fabY        fatty acid biosynthesis␣
↪protein FabY    +       897     GTC     GT[T]   V       V       299     Syn     0   ␣
↪    0       0       0       0       0       0       0       0       0 1       0
snp7    61685   G       C       intergenic      .       +       64 bp from YPO_
↪RS01280|YPO_RS01280      .       .       .       .       .       .       0       0 ␣
↪     0       0       0       0       0       0       1       0       0 0
snp8    74539   C       T       YPO_RS01350|envC        murein hydrolase activator␣
↪EnvC -       361     GCC     [A|CC   A       T       121     NS      1       1     ␣
↪ 1       1       1       1       1       1       1       1       1 1
snp9    76590   C       T       intergenic      .       +       .       -       -     ␣
↪  -       -       -       -       0       0       0       0       0       0       0␣
↪       0       0       0       1       0
```

Lets take a look now at the file SNPs_alignment_clean.fasta that compared to SNPs_alignment.fasta does not contain polymorphic site with missing information.



6.4 Excluding SNPs

For the combine command it is possible also to provide a file that includes the SNPs to exclude with the option -e when searching for all polymorphic sites and building the fasta file. Compared to the annotate command, the file here must be in yaml format and not a tabulated format. Here is an example:

KEYWORDS: YPO3746;YPO3747

COORDINATES: 64265;7662

The yaml file has two arguments: KEYWORDS and COORDINATES. With the argument KEYWORDS you can specify genes names, a word within a product name, etc…

- If you specify a gene name, all SNPs within that gene will be excluded.
- If you specify for example 'cytochrome', all genes having the word cytochrome in their product will be ignored.

For the argument COORDINATES, it is straightforward, it will contains the list of SNP positions to be excluded.

The rational behind this choice of file structure is that after creating the files SNPs_polymorphic_sites.txt and SNPs_alignment.fasta, and making some downstream analyses like building phylogenetic trees and checking some SNPs for confirmation, it is possible to exclude for example SNPs that are in regions that included some repeats or a family of genes that has duplicated genes… such file will give all the flexibility for users.

Note: In the next version of snpToolkit, it will be possible instead of providing the bam files, to search back the original vcf files if only they include all the positions and not only variable sites:

- With samtools mpileup you can use the option -aa to output all positions, including unused reference sequences.
- With gatk haplotypeCaller you can use mode EMIT_ALL_SITES with the option -output-mode

CHAPTER 7

The analyse command

```
snptoolkit analyse -h
usage: snptoolkit analyse [-h] -p POLYMORPHIC_SITES [-c CONFIG]

optional arguments:
-h, --help            show this help message and exit

snpToolkit analyze required options:
-p POLYMORPHIC_SITES  provide the path of the polymorphic sites you want to analyze
-c CONFIG             provide the path of the configuration file that contains the␣
↪information to use for data visualization
```

The main goal of this analysis is to use two dimentionality reduction methods: PCA and UMAP to cluster all your samples based on the distibution of all identified polymorphic sites between them. Principal Component analysis (PCA) is a quite knowing method and is an unsupervised linear dimensionality reduction and data visualization technique. On the other hand, UMAP is a Uniform Manifold Approximation and Projection for Dimension Reduction. From a visualization point of view, PCA tries to preserve the global structure of the data while UMAP tries to preserve global and local structure. To apply both of these methods you need to provide as input the file **SNPs_polymorphic_sites.txt** generated with the snptoolkit combine command.

```
$ snptoolkit analyse -p SNPs_polymorphic_sites.txt
Dash is running on http://127.0.0.1:8050/

* Serving Flask app "plot_polySites_output" (lazy loading)
* Environment: production
   WARNING: This is a development server. Do not use it in a production deployment.
   Use a production WSGI server instead.
* Debug mode: off
* Running on http://127.0.0.1:8050/ (Press CTRL+C to quit)
```

Note: In case you used the option –bam with the snptoolkit combine command, two output files will be generated: SNPs_polymorphic_sites.txt and SNPs_polymorphic_sites_clean.txt. The file SNPs_polymorphic_sites_clean.txt does not contains any missing information indicated with a question mark "?" and should be used as input file for

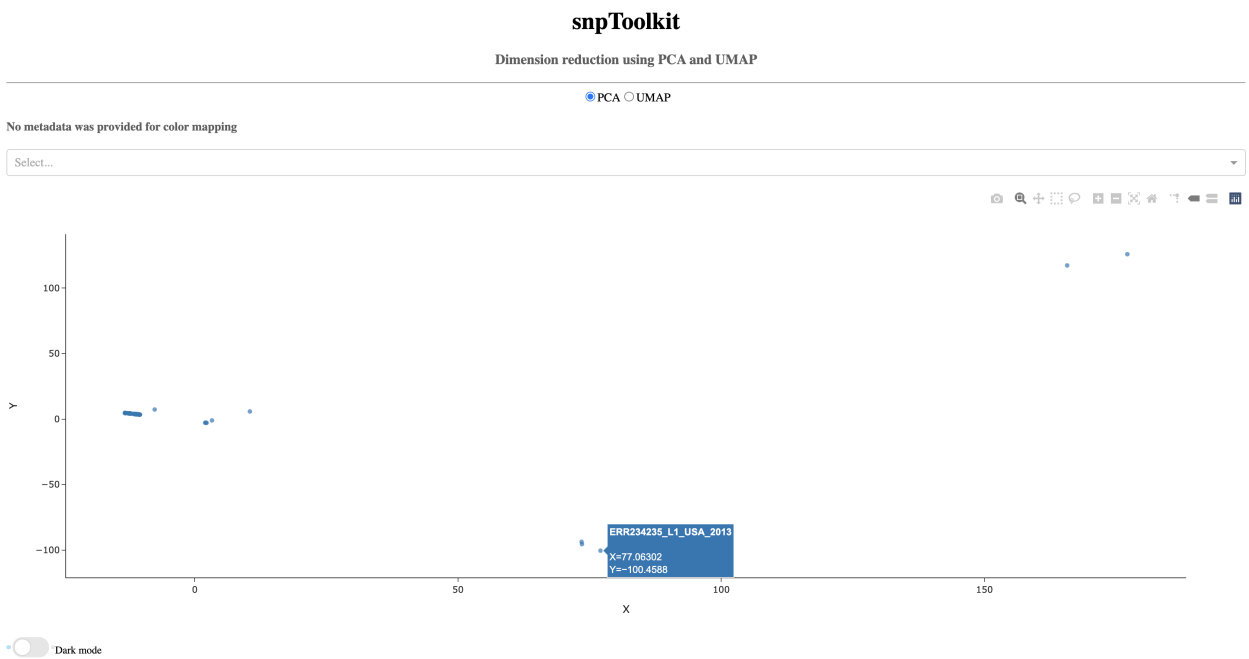
dimentionality reduction.

After running the command *snptoolkit analyse -p SNPs_polymorphic_sites.txt*, you can access your result following the link *http://127.0.0.1:8050/*.

Note: Please note that this step may take some time depending on the size of your data.

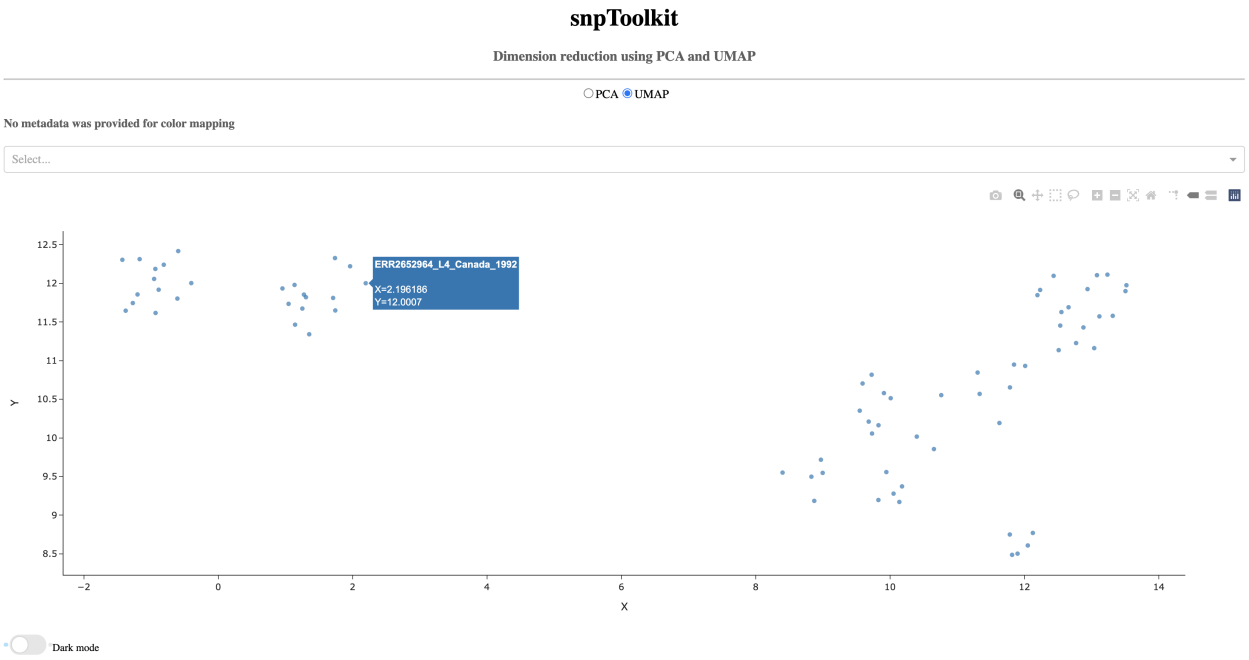
The result will be displayed as follows:

7.1 PCA



As you notice, when you hover of each dot, the name of the corresponding sample will be displayed.

7.2 UMAP



7.3 Color mapping

To take a better advantage of these two methods, it is possible to provide a configuration file that contains the metadata about the analyzed samples. This information will be used for color mapping which will make the visualization more comprehensive. The configuration file is a tab delimited file. Here is an example:

```
$ less metadata_file
        Lineage Rifampicin      Isoniazid       Pyrazinamide    Ethambutol
↪compensatory    Location        MDR
ERR760737_L4_Argentina_2006     L4      R       R       S       R       YES
↪Argentina       RR
ERR037537_L4_Malawi_0   L4      S       S       S       S       NO      Malawi  SS
ERR2652979_L4_Brazil_2004       L4      S       S       S       S       NO
↪Brazil  SS
ERR2652959_L4_Canada_2003       L4      S       S       S       S       NO
↪Canada  SS
ERR2653008_L4_Brazil_2004       L4      S       S       S       S       YES
↪Brazil  SS
ERR2652915_L4_USA_1999  L4      S       S       S       S       NO      USA     SS
ERR245833_L1_Malawi_0   L1      S       S       S       S       YES     Malawi  SS
ERR037471_L4_Malawi_0   L4      S       S       S       S       NO      Malawi  SS
ERR037549_L4_Malawi_0   L4      S       S       S       S       YES     Malawi  SS
ERR245675_L1_Malawi_0   L1      S       S       S       S       YES     Malawi  SS
ERR760755_L4_Argentina_2006     L4      R       R       S       R       YES
↪Argentina       RR
```

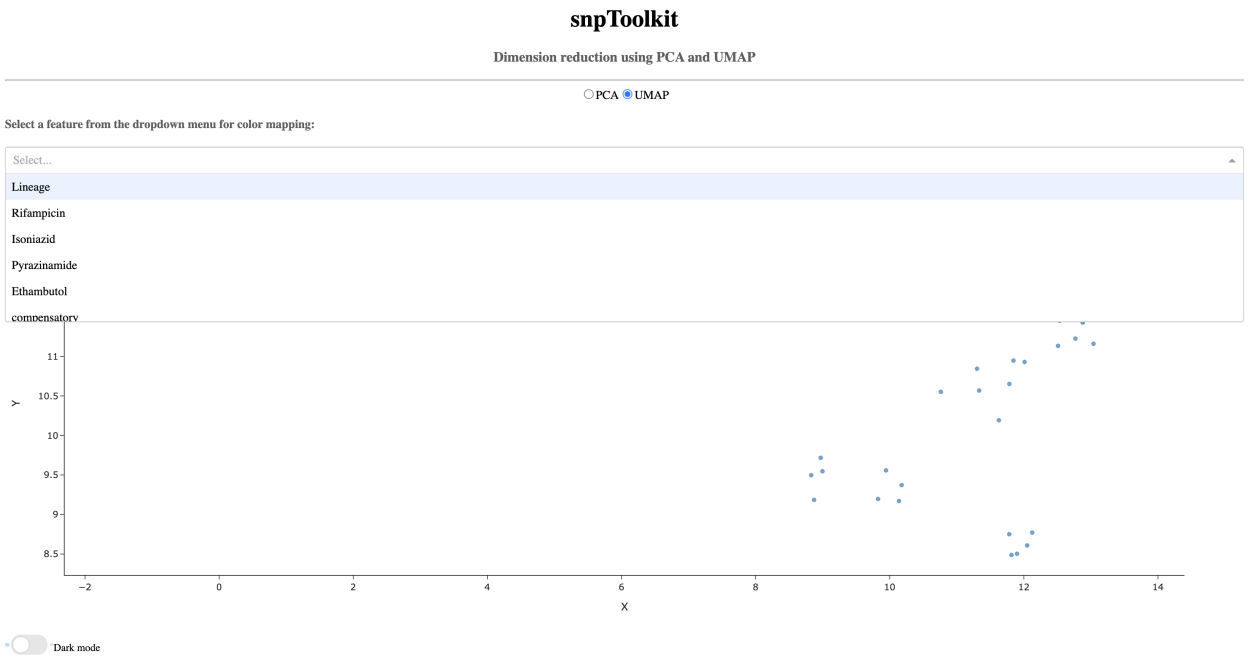
Note: Please note that the configuration file must contains all the samples that are present in the input file SNPs_polymorphic_sites.txt. In case not all the information is availble, you can just any label on the correspond-

ing cells e.g. NA for not availble.

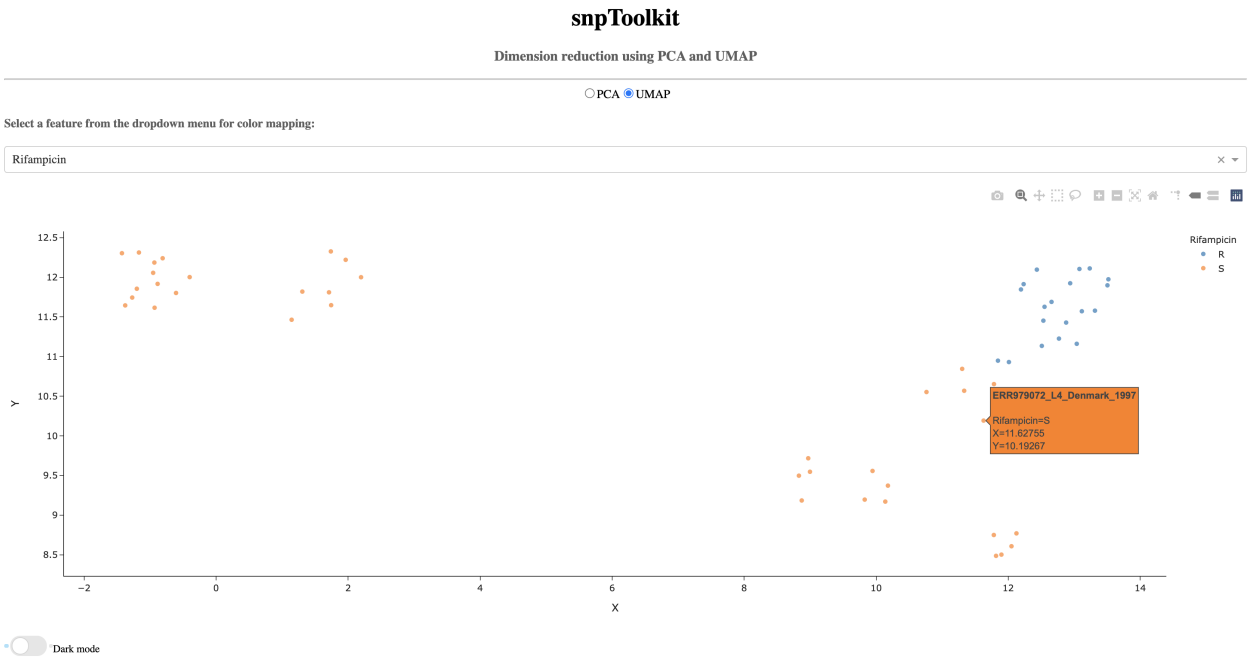
lets run the command analyse with the configuration file:

```
$ snptoolkit analyse -p SNPs_polymorphic_sites.txt -c metadata_file

Dash is running on http://127.0.0.1:8050/

* Serving Flask app "plot_polySites_output" (lazy loading)
* Environment: production
   WARNING: This is a development server. Do not use it in a production deployment.
   Use a production WSGI server instead.
* Debug mode: off
* Running on http://127.0.0.1:8050/ (Press CTRL+C to quit)
```

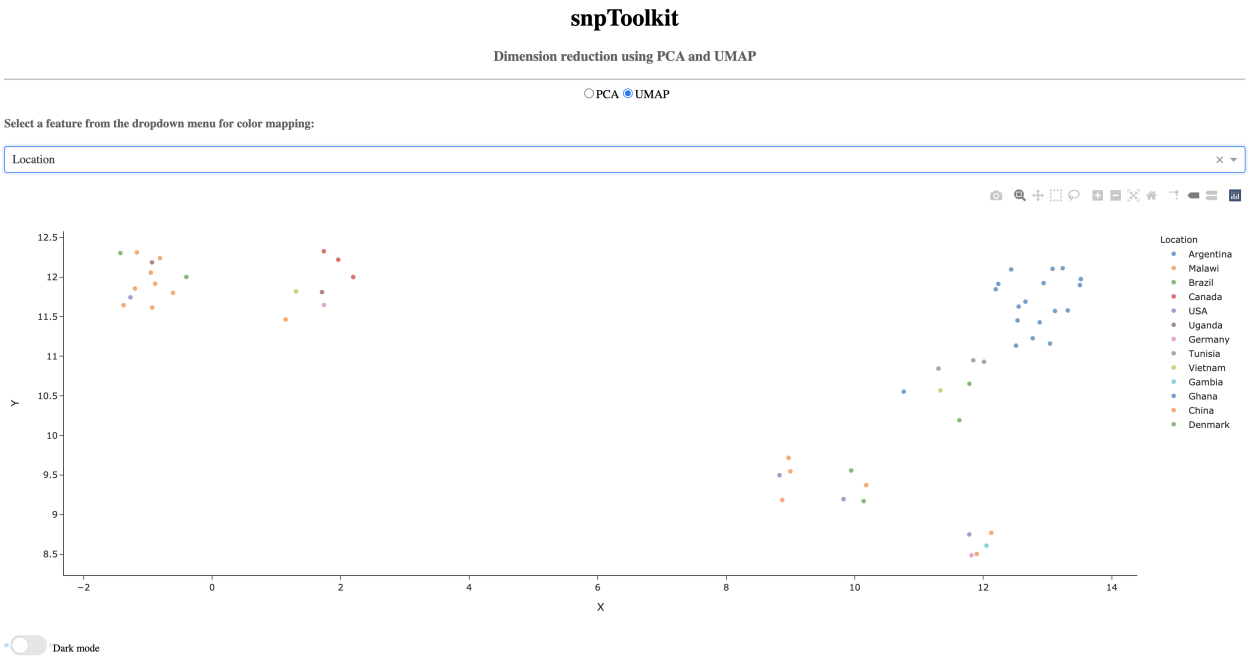
As you can see below, now the dropdown menu shows the list of features to use for coloring the different samples.



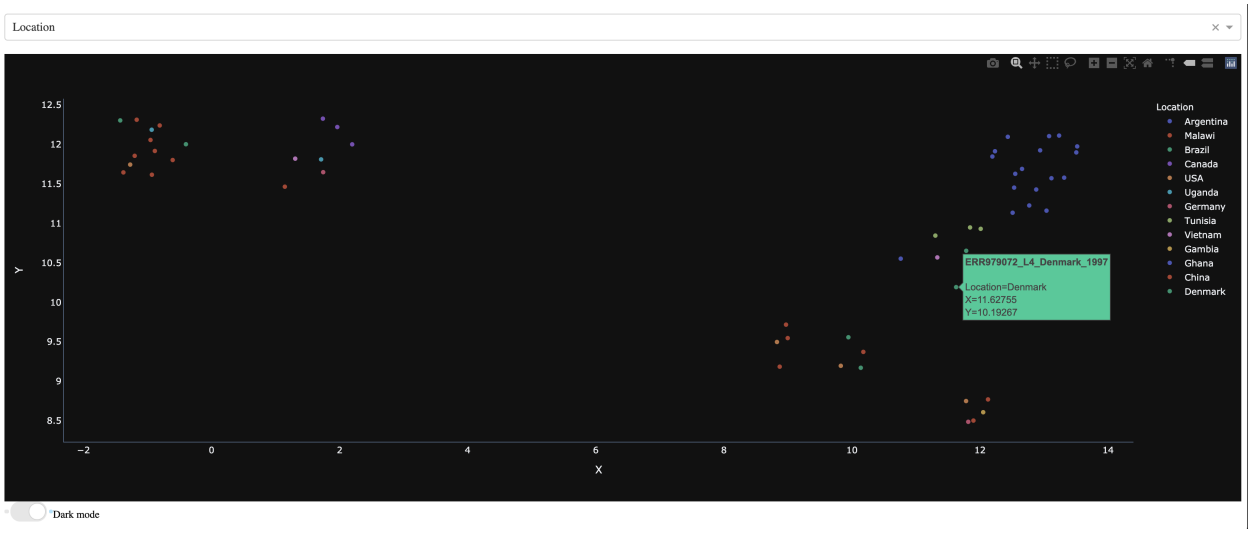
Now lets color the samples based on their resistance to rifampicin



Now lets color the samples based on their location



For those (like me) that like dark mode in general you can turn it on to get graphs with dark bakground.


Location
Location
Argentina
Malawi
Brazil
Canada
USA
Uganda
Germany
Tunisia
Vietnam
Gambia
Ghana
China
Denmark
ERR979072_L4_Denmark_1997
Location=Denmark
X=11.62755
Y=10.19267
Y
X
Dark mode