CorGAT Documentation

Release 1.0.0

Matteo Chiara

Jan 26, 2023

Command line version

1 Prerequisites and usage 3

1.1 Mummer installation 4

1.2 Mummer installation MacOS X 4

1.3 Download of the Reference genome 5

2 Align to the reference genome 7

3 Functional annotation 9

3.1 Functional annotation: Important! 10

3.2 Functional annotation: adding functional elements! 11

4 Quickstart 13

5 Importing your data 15

6 Analysing your data 19

7 Multiple Fasta files 21

8 Using a single Multifasta file 23

9 Annotation 25

10 Installing the CorGAT Galaxy 27

11 How to use 29

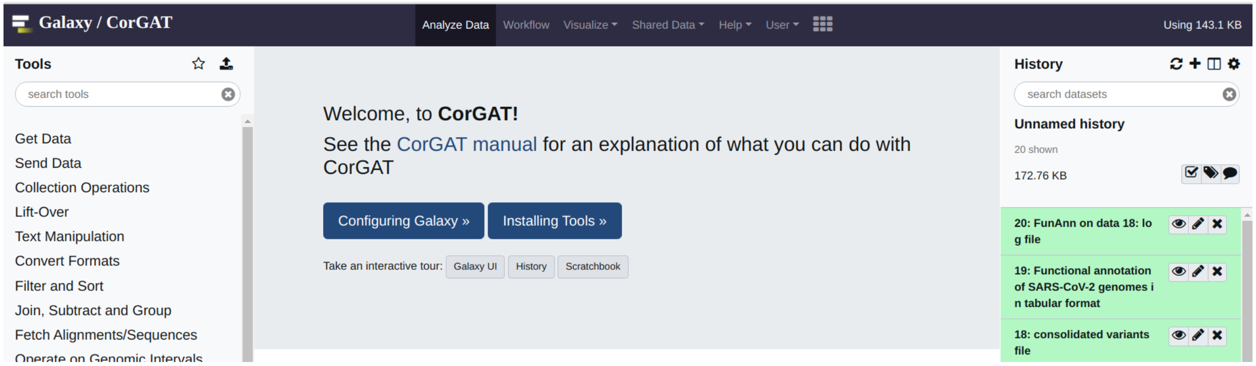
12 What to do next: 31

13 Galaxy dockers 33

14 Indices and tables 35

CorGAT is a collection of Perl utilities that can be used to align complete assemblies of SARS-CoV-2 genomes wih the reference genomic sequence, to obtain a list of polymorphic positions and to **annotate** genetic variants according to the method described in *Chiara et al 2020* to be published soon (hopefully). The manuscript is currently submitted and undergoing peer review.

This software package is composed of 2 very simple scripts and a collection of files with functional annotation data. Since the number of available SARS-CoV-2 genomic sequences is increasingly constantly, these files are regularly updated on a monthly basis. If you do not feel comfortable in installing/running CorGAT from the command line, you can find a Galaxy running the software at http://corgat.ba.infn.it/galaxy , or download a dockerized version of the Galaxy, with all the tools here.



Welcome, to **CorGAT!**

See the CorGAT manual for an explanation of what you can do with CorGAT

Take an interactive tour:

Unnamed history

20 shown

172.76 KB

20: FunAnn on data 18: log file

19: Functional annotation of SARS-CoV-2 genomes in tabular format

18: consolidated variants file

If you find any of this software useful for your work, please cite:

Chiara M, Horner DS, Gissi C, Pesole G. Comparative genomics provides an operational classification system and reveals early emergence and biased spatio-temporal distribution of SARS-CoV-2 bioRxiv 2020.06.26.172924; doi: https://doi.org/10.1101/2020.06.26.172924

and

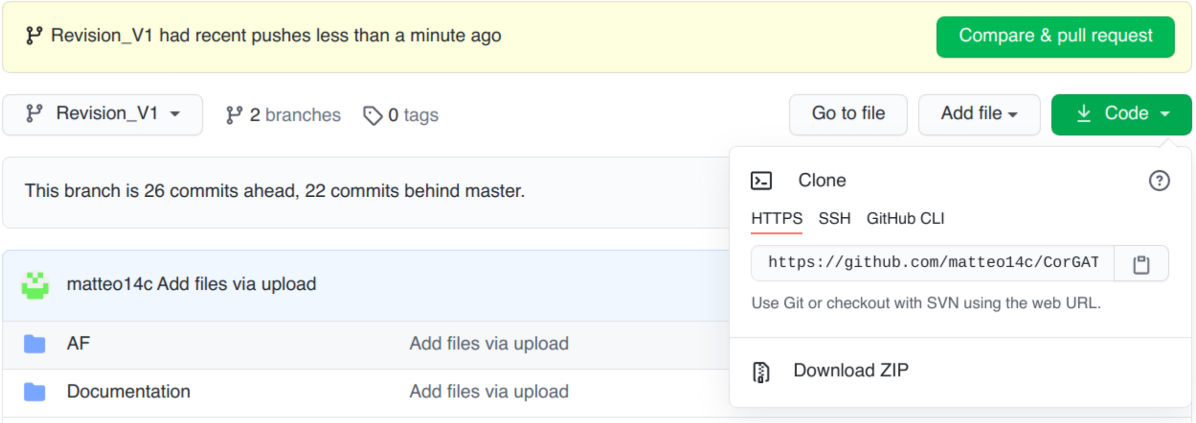
Chiara M, Zambelli F, Tangaro MA, Mandreoli P, Horner DS, Pesole G. CorGAT: a tool for the functional annotation of SARS-CoV-2 genomes. Under Peer review

If you find any issue with the software, please contact me, or report it on github.

CHAPTER 1

Prerequisites and usage

This software package is composed of 2 very simple scripts and a collection of files with functional annotation data. The only requirement is that you have an up to date installation (see below) of the Mummer package in your system and a copy of the reference genomic sequence, in fasta format. All the files (scripts, genomic sequences and accessory files) should be placed in the same folder. To install the command line version od CorGAT you can simply download the most recent version of the program, from the following [link]https://github.com/matteo14c/CorGAT/ . Click on code, and then on Download Zip, as illustrated in this Figure:



Revision_V1 had recent pushes less than a minute ago

This branch is 26 commits ahead, 22 commits behind master.

matteo14c Add files via upload

```
https://github.com/matteo14c/CorGAT
```

Use Git or checkout with SVN using the web URL.

At this point, you can place yoursef il the folder where the program was downloaded. For example if the default of your browser is the Downloads folder:

```
cd Downloads
```

you should see a zip archive named `CorGAT-Revision_V1.zip`. At this point to execute CorGAT you only need to unzip the archive and place yoursef in the CorGAT-Revision_V1 folded

```
unzip CorGAT-Revision_V1.zip
```

```
cd CorGAT-Revision_V1
```

1.1 Mummer installation

Please follow this link https://sourceforge.net/projects/mummer/files/ for detailed instruction on how to install and run Mummer. Please notice that after you have succesfully compiled all the executables by running:

```
make install
```

you will still need to place add these files to your executable PATH, either by adding/copying all the files to one of the directories already included in the PATH or by adding the whole mummer directory (where all the software was compiled) to the your PATH of executables. If for example all your executables are in a folder called "Mummer" in your home directory on a unix system you can symply run:

```
export PATH=~/Mummer:$PATH
```

1.2 Mummer installation MacOS X

Download Mummer at: https://sourceforge.net/projects/mummer/files/latest/download and extract the archive (tar.gz) file. Open up Terminal and:

```
tar xvzf MUMmer3.23.tar.gz
```

As explained in the INSTALL file, included in the Mummer package to build Mummer:

```
cd MUMmer3.23
make check
```

If make check does not report any error everything should be ok, then run:

```
make install
```

You should get something similar to this.

Now that mummer you have successfully built the binaries are, you need to add them to $PATH. Run the following command with your favourite text editor:

```
sudo vim /etc/paths
```

Enter your password, when prompted. Go to the bottom of the file, and enter the path you wish to add. For example, if you built Mummer in /Users/yourname/test/MuMmer3.23, add this to the file:

```
/usr/local/bin
/usr/bin
/bin
/usr/sbin
/sbin
/Users/yourname/test/MUMmer3.23
```

Save the file in vim

```
:wq
```

And finally you can test if everything is in place. Open a *NEW* terminal. To test if mummer is now in your PATH, run:

```
echo $PATH
```

You should see something like:

```
echo $PATH
/usr/local/opt/ruby/bin:/usr/local/bin:/usr/bin:/bin:/usr/sbin:/sbin:/Users/yourname/
↪test/MUMmer3.23
```

The Mummer package, and all its utilities are now available to be executed in your shell, and for CorGAT as well. For example, type “nucmer” to execute nucmer:

```
nucmer
USAGE: nucmer  [options]  <Reference>  <Query>

Try '/Users/marco/IBIOM-CNR/CorGAT/MUMmer3.23/nucmer -h' for more information.
```

1.3 Download of the Reference genome

The reference genome of SARS-CoV-2 can be found here.

On a unix system you can download this file using wget

```
wget https://ftp.ncbi.nlm.nih.gov/genomes/all/GCF/009/858/895/GCF_009858895.2_
↪ASM985889v3/GCF_009858895.2_ASM985889v3_genomic.fna.gz
```

followed by

```
gunzip GCF_009858895.2_ASM985889v3_genomic.fna.gz
```

Please notice that however the *align.pl* utility is going to download the file for you, if a copy of the reference genome is not found in the current folder. However, since the `wget` command is required this is supposed to work only unix and unix alike systems. *align.pl* will complain with an error if `wget` is not available in your system.

CHAPTER 2

Align to the reference genome

The helper script *align.pl* can be used to align a collection of genomic sequences to the reference assembly of SARS-CoV-2 and obtain a list of polymorphic positions. The script automates all the required steps. align.pl currently allows 3 different distinct methods to provide input files/sequences.

Inputs, alternatives:

1. Through a multifasta file: option –multi;
2. Through a list of file names: option –filelist;
3. By specifying a “suffix” that is common to all the names of the files that should be analyses: option –suffix;

All input files *MUST* be in the *same folder* from which the program is executed. A temporary directory will be created to store all the intermediate files and the alingment results for every file. The name of this temporary directory can be specified using the **–tmpdir option**. Please notice that this temporary directory, normally, will be deleted after the execution of align.pl. The **–clean option**, can be used to alter this behavior. If set to **F=FALSE** the temporary directory will not be deleted.

Please check the section *Prerequisites and usage* to obtain the reference genome sequence file. This file also needs to be in the same folder from which the program is executed (and yes **the same** where you have all the files). If the reference genome file is missing, *aling.pl* will try to download it from Genbank. Although this is supposed to work only for unix and unix alike systems (the *wget* command is required)

Finally the name of the output file can be specified by using the **–out option**. This defaults to **ALIGN_out.tsv**.

Once you have everything in place, to check if everything works you can simply run:

> perl align.pl

The help message, should be self-explanatory. You can try all the 3 different commands under the EXAMPLE section to test align.pl . Example input files are also provided in the main repository of CorGAT

```
perl align.pl --multi <apollo.fa>`  will align all the genomes contained in the␣
↪multifasta file named apollo.fa
```

```
perl align.pl --suffix fasta` will use all the files with the *.fasta suffix in the␣
↪current folder and finally
```

```
perl align.pl --filelist lfile` will align the files specified in lfile. One file per␣
 ↪line. Again, all files need to be in the current folder
```

For every genome you will obtain a file with the *.snps* extension, reporting all the polymorphism identified by nucmer. These files will be stored in the temporary directory, as specified by the –tmpdir option (default align.tmp). If the –clean option is set to **T (TRUE) however, this directory will be removed** after the execution of the program.

The final output consists in a simple tabular file (default name **ALIGN_out.tsv**) that lists genetic variants on the rows, and reports their presence (1) or absence (0) in the different genomes included in your analysis in the columns. This file provides the input for *annotate.pl*

CHAPTER 3

Functional annotation

The `annotate.pl` utility is used to perform functional annotation of SARS-CoV-2 variants. The program can be executed very easily, by running:

```
perl annotate.pl --in inputFile
```

This script is very simple to use. Only 3 parameters are accepted in input:

1. **–in** to specify the input file;
2. **–out** to set the name of the output file;
3. **–conf** to provide a configuration file;

> **Warning:** The configuration file, is nothing but a simple table that contains the name of the files that should be used to provide different types of functional annotations. A valid example of a configuration file is provided by *corgat.conf* as included in the current repo.(See below). Each row of this file is associated with a keyword (first column), to which the name of the file that should be used follows (second column). In particular:
>
> - `genetic -> specifies the name of the file with the genetic code`
> - `genome -> the name of the file with the reference genome sequence`
> - `annot -> a table, with the coordinates of functional genomic elements (see below)`
> - `hyphy -> the file used to provide annotation of variants under selection according to hyphy`
> - `AF -> the file with allele frequency data`
> - `EPI -> the files with annotations of predicted epitopes`

Since the number of publicly available genome sequences is increasing constantly over time, the hyphy and AF files are updated on a regular (monthly) basis. The corgat.conf file as provided in this repo is set to use the most up to date version of each file, denoted by suffix current.csv. Older versions are stored in the hyphy_data and AF folders respectively. Should you need to use an older version of the AF or hyphy annotations for any specific reason, you can simply modify your copy of corgat.conf accordingly. Average users however, should not need to edit this file.

The output consists in a simple table, delineated by <tab> (tabulations) and formatted as follows. If/when the docker or Galaxy version of this software are used, the output can be visualized directly in your browser:

Genomic position	Ref allele	Alt allele	Funct Elem annot	Allele Frequency	Epitopes annot	Selection annot	MFE annot
376	G	T	nsp1:c.111G>T,p.E37D,missense;orf1ab:c.111G>T,p.E37D,missense;	0.166	FGDSVEEVL,1,HLA-C*08:01	fel:false;meme:true;kind:positive	NA
29742	G	T	3'UTR:nc.G68T,NA,NA,NA;sl5:nc.G15T,NA,NA,N;	0.7635	NA	NA	mfe:-5.6;-4.76;-10.93;

Annotation of functional genomic elements, consists of 4 fields, separated by commas (**,**):

1. name of the element, followed by ":"
2. relative position (c.= coding, nc.=non coding)
3. amino acid change (NA if a non coding element)
4. predicted effect on protein (NA if a non coding element)

When a variant is overlapped by more than one element, multiple annotations are reported, separated by semicolumns (**;**)

Annotation of epitopes is according to https://doi.org/10.1038/s10038-020-0771-5 . The sequence of the epitope/epitopes is reported followed by the number and by the names of the HLAs that are predicted to recognize the epitope. Multiple annotations are separated by semicolumns (**;**).

For example in *FGDSVEEVL,1,HLA-C*08:01*, **FGDSVEEVL** is the sequence of the predicted epitope/epitopes, **1** and **HLA-C*08:01** indicate that the sequence is recognized by just 1 HLA, that is **HLA-C*08:01**.

Annotation of sites under selection is very simple: **fel:** is used to indicate if the site is under selection according to fel. Possible values are *true* or *false*. **meme** is the equivalent, but for the meme method. The **kind:** field indicates the type of selection: *positive* or *negative*.

The MFE annot column reports **predicted changes** in MFE (minimum free energy) for variants associated with secondary structure elements. Please notice that this annotation does not report the predicted MFE, but the **difference** between the MFE of the element based on the reference genome sequence, with the MFE calculated on the alternative sequence. Negative values indicate a descrease in MFE (a more stable structure). Positive values are suggestive of a less stable structure (increase in MFE). Three values are reported, representing respectively MFE of: *optimal secondary structure*, *the thermodynamic ensemble* and *the centroid secondary structure*. Obviusly there is no absolute cut-off for interpreting these results, however high shifts (>1 or <-1) in MFE might be suggestive of functional implications.

3.1 Functional annotation: Important!

Please notice, that to work properly `annotate.pl` needs to have access (read) several annotation files which provide the different types of functional annotations. If these files are not available, the program will exit with an error, complaining that one or more of the files are missing.

These files are *strictly required* and can be downloaded from the CorGAT Github repository. The repository itself is updated on a monthly basis. So it is *highly advised* that the latest version of the files should be downloaded *before* you perform your analyses. The Galaxy and docker versions of CorGAT are configured to use the most recent version of each file. The update process is handled automatically

Warning: All the files need to be (and normally are) in the **same folder** from which `annotate.pl` is executed.

The annotation files, all in simple text format include:

1. *genetic_code* -> 3 column file with the standard genetic code
2. *GCA_009858895.3_ASM985889v3_genomic.fna* -> the reference SARS-CoV-2 genome assembly sequence
3. *annot_table.pl* -> a 4 column tabular file with genomic coordinates of functional genomic elements
4. *AF_current.csv* -> tabular file with allele frequency data
5. *MFE_annot.csv* -> tabular file with Mininum Free Energy predictions for all the possible Single Nucleotide substitutions in secondary structure elements
6. *epitopes_annot.csv* -> tabular file with annotation of predicted epitopes
7. *hyphy_current.csv* -> tabular file with aa residues under selection according to meme/fel

Please see below for a brief guide that will help you to define additional functional elements in `annot_table.pl`.

3.2 Functional annotation: adding functional elements!

Functional genomic elements in the genome of SARS-CoV-2 are specified by a five columns tabular format file called `annot_table.pl`. This file can be used to specify additional functional elements and/or use a personalized annotation. The file has a very simple format: for every element, the first three columns specify respectively, the name of the element (column 1), the start (column 2) and the end coordinate (column 3) on the genome. The fourth column defines the functional class of the element. At the moment 4 different classes are supported:

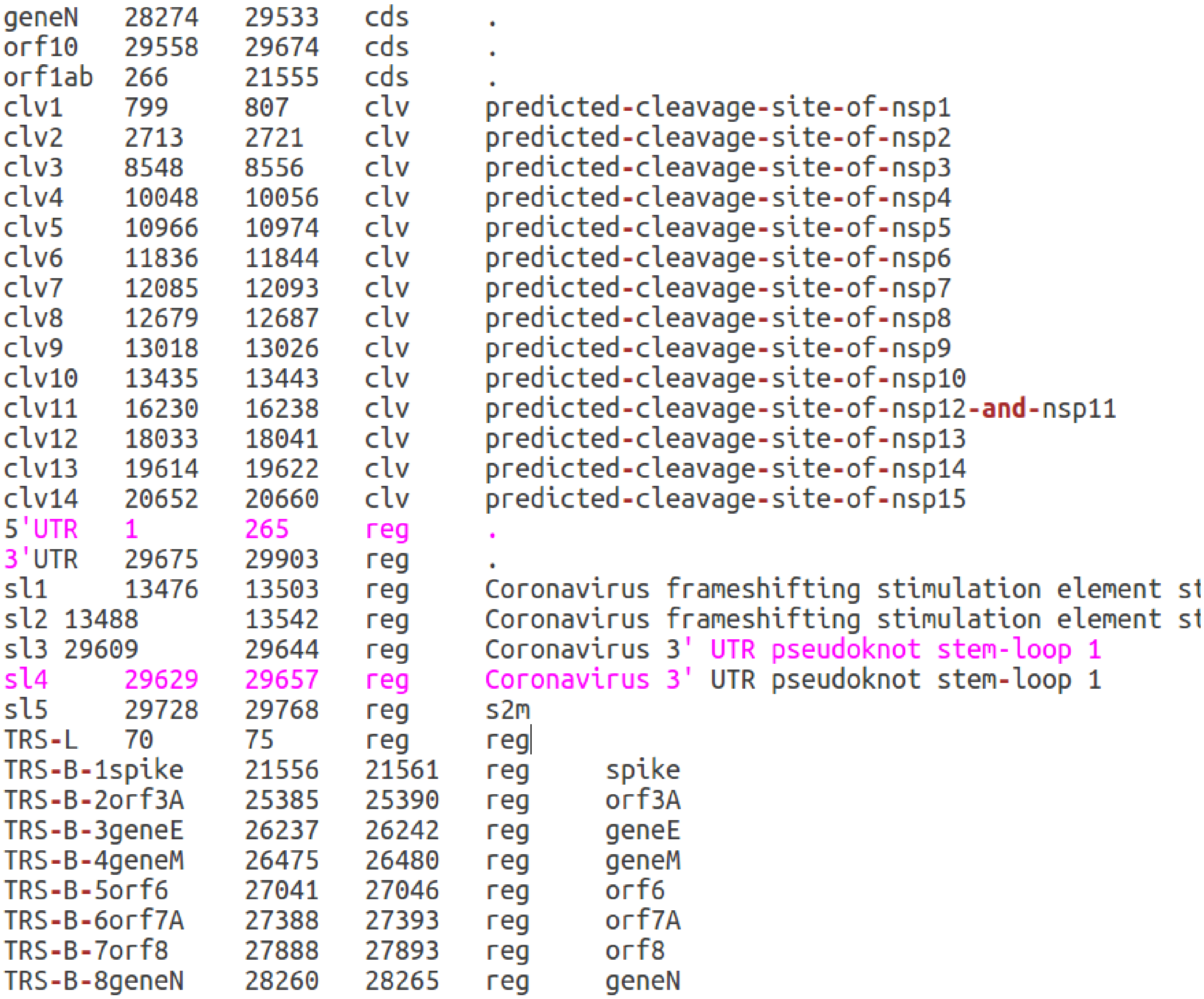
1. protein coding sequences (*cds*)
2. regulatory elements (*reg*)
3. cleavage sites of SARS-CoV-2 polyproteins (*clv*)
4. Sites associated with epigenetic modifications (*epi*)

Finally the fifth column is optional and contains additional comments and annotations.

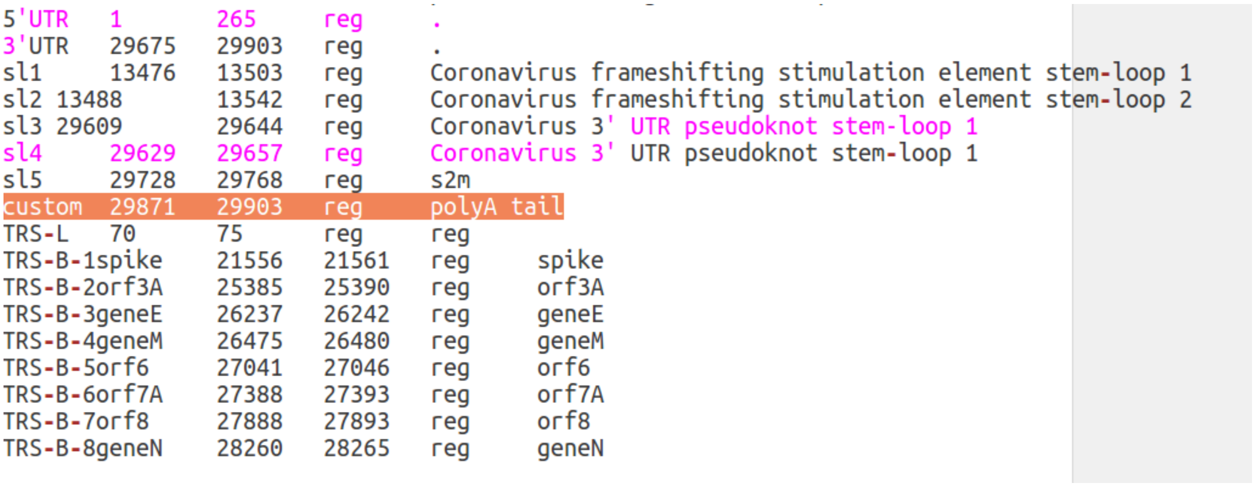
To add elements to `annot_table.pl` you need to open this file with your favourite text editor. First of all position yourself in the CorGAT directory (the directory that was created when you downloaded CorGAT from Github). You should see a file named `annot_table.pl` . Open this file with your favourite text editor. You should see something similat to this:

At this point any modification of the annotation file should be very simple. For example you can delete any element functional element by deleting the corresponding entry in this file. To add a novel element instead, you should add a line. As you can see from this example, where a custom annotation (custom) of the polyA tail of the genome as been added.

Please rememember that the different columns of this files are delineated by `tabulations`. Currently the Galaxy and docker version of CorGAT does not allow the specification of a custom `annot_table.pl` file.



```
geneN   28274   29533   cds     .
orf10   29558   29674   cds     .
orf1ab  266     21555   cds     .
clv1    799     807     clv     predicted-cleavage-site-of-nsp1
clv2    2713    2721    clv     predicted-cleavage-site-of-nsp2
clv3    8548    8556    clv     predicted-cleavage-site-of-nsp3
clv4    10048   10056   clv     predicted-cleavage-site-of-nsp4
clv5    10966   10974   clv     predicted-cleavage-site-of-nsp5
clv6    11836   11844   clv     predicted-cleavage-site-of-nsp6
clv7    12085   12093   clv     predicted-cleavage-site-of-nsp7
clv8    12679   12687   clv     predicted-cleavage-site-of-nsp8
clv9    13018   13026   clv     predicted-cleavage-site-of-nsp9
clv10   13435   13443   clv     predicted-cleavage-site-of-nsp10
clv11   16230   16238   clv     predicted-cleavage-site-of-nsp12-and-nsp11
clv12   18033   18041   clv     predicted-cleavage-site-of-nsp13
clv13   19614   19622   clv     predicted-cleavage-site-of-nsp14
clv14   20652   20660   clv     predicted-cleavage-site-of-nsp15
5'UTR   1       265     reg     .
3'UTR   29675   29903   reg     .
sl1     13476   13503   reg     Coronavirus frameshifting stimulation element st
sl2 13488       13542   reg     Coronavirus frameshifting stimulation element st
sl3 29609       29644   reg     Coronavirus 3' UTR pseudoknot stem-loop 1
sl4     29629   29657   reg     Coronavirus 3' UTR pseudoknot stem-loop 1
sl5     29728   29768   reg     s2m
TRS-L   70      75      reg     reg
TRS-B-1spike    21556   21561   reg     spike
TRS-B-2orf3A    25385   25390   reg     orf3A
TRS-B-3geneE    26237   26242   reg     geneE
TRS-B-4geneM    26475   26480   reg     geneM
TRS-B-5orf6     27041   27046   reg     orf6
TRS-B-6orf7A    27388   27393   reg     orf7A
TRS-B-7orf8     27888   27893   reg     orf8
TRS-B-8geneN    28260   28265   reg     geneN
```



```
5'UTR   1       265     reg     .
3'UTR   29675   29903   reg     .
sl1     13476   13503   reg     Coronavirus frameshifting stimulation element stem-loop 1
sl2 13488       13542   reg     Coronavirus frameshifting stimulation element stem-loop 2
sl3 29609       29644   reg     Coronavirus 3' UTR pseudoknot stem-loop 1
sl4     29629   29657   reg     Coronavirus 3' UTR pseudoknot stem-loop 1
sl5     29728   29768   reg     s2m
custom  29871   29903   reg     polyA tail
TRS-L   70      75      reg     reg
TRS-B-1spike    21556   21561   reg     spike
TRS-B-2orf3A    25385   25390   reg     orf3A
TRS-B-3geneE    26237   26242   reg     geneE
TRS-B-4geneM    26475   26480   reg     geneM
TRS-B-5orf6     27041   27046   reg     orf6
TRS-B-6orf7A    27388   27393   reg     orf7A
TRS-B-7orf8     27888   27893   reg     orf8
TRS-B-8geneN    28260   28265   reg     geneN
```

CHAPTER 4

Quickstart

To do all of the above:

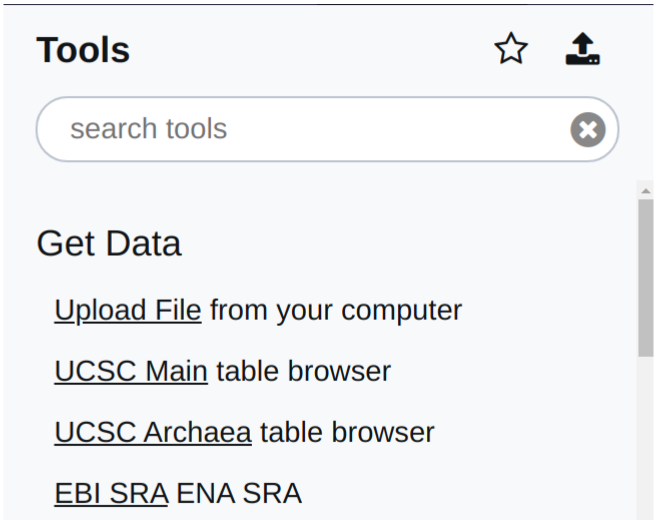
1. Put a multi fasta file of genome sequences in one folder.
2. download this repository.
3. run `perl align.pl --multi <your_fasta_file> --out <your_alignment_results>`.
4. run `perl annotate.pl --in <your_alignment_results> --out <funct_annot_output_file>`.
5. open the output file, and read the annotations.

CHAPTER 5

Importing your data

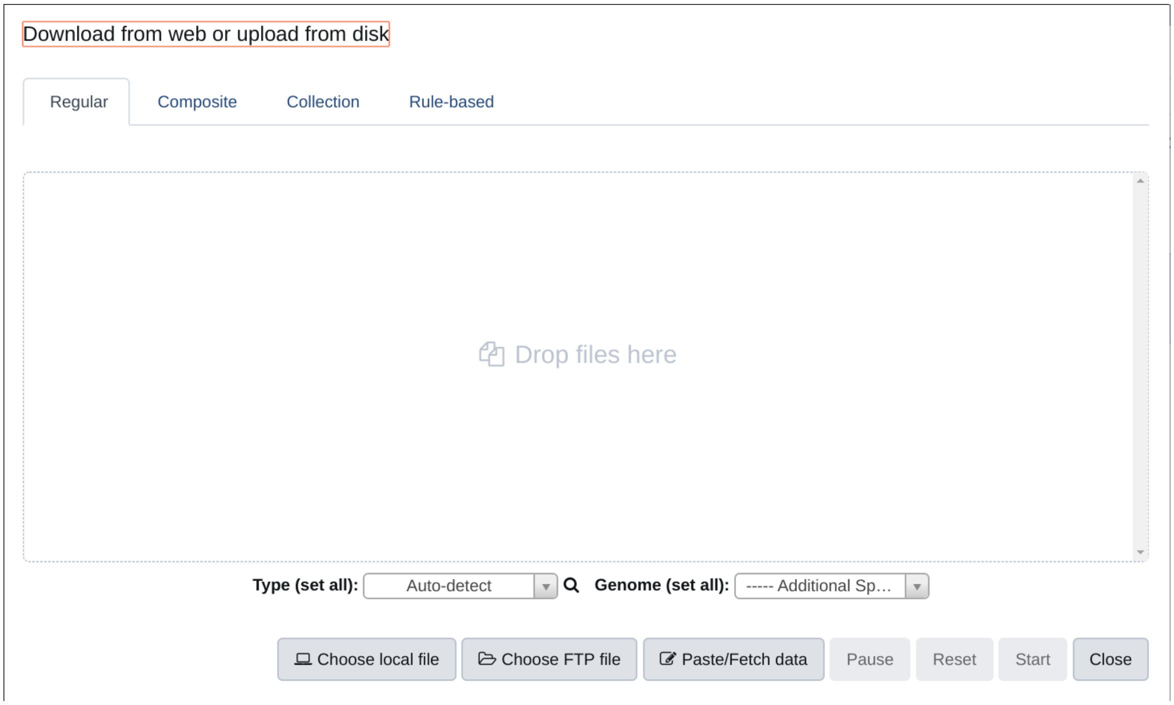
Warning: Please notice, this manual provide just a quick and simple reference for the usage of the Galaxy version of CorGAT. Please refer to https://galaxyproject.org/learn/ for a complete and accurate reference on how to use Galaxy

Before doing anything you are required to import your data into Galaxy. This operation is very simple and can be performed by using the `Upload file` menu, under `Get data`. As outlined in this figure:



You will be prompted with the following menu:

Select `Choose local file` and the folder on your system where you have your SARS-CoV-2 genome assemblies. These need to be in FASTA format. One genome per file. Please notice that the name specified in the header of your fasta will be used to identify each genome in all the subsequent steps of this analysis. Use sensible names, preferably avoid names containing strange characters or spaces. Select all the files that you want to upload to Galaxy. Multiple



Type (set all): Auto-detect

Genome (set all): ----- Additional Sp…

files can be selected at this time. Once you have selected all your files, you should obtain something that looks like the picture below. At that point hit start (the blue button). All the files will be imported in Galaxy.

Once your files are imported you should see something like the picture below, meaning that Galaxy is ready to analyse your data.

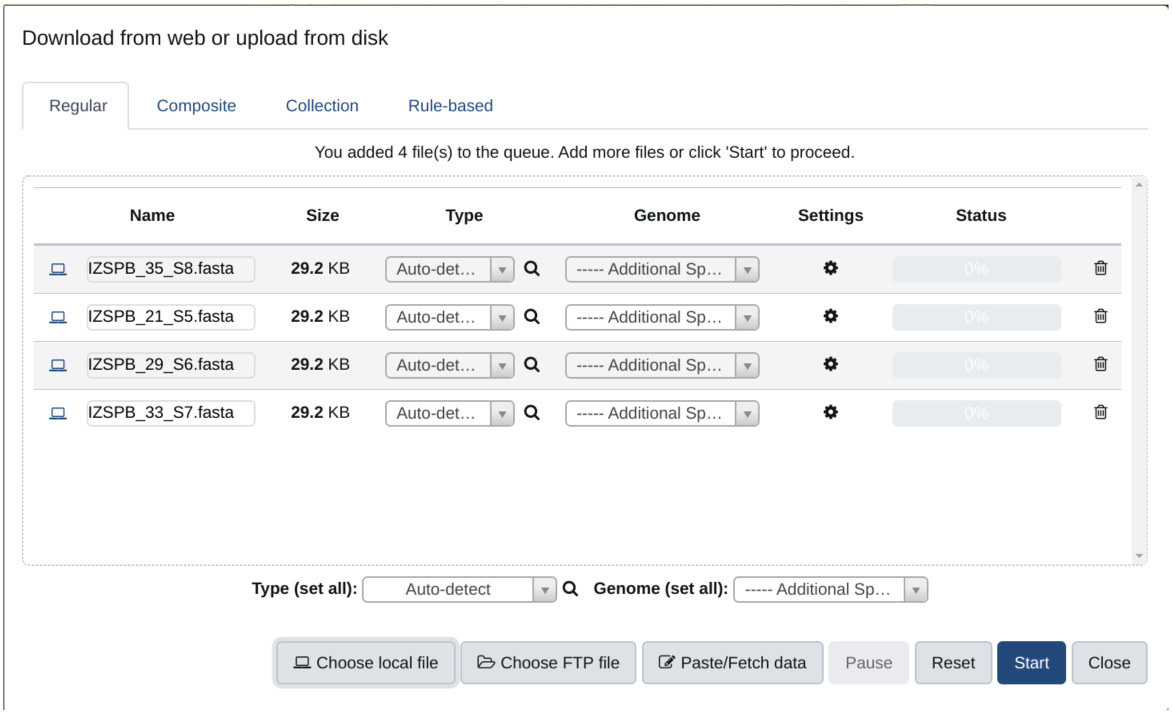
At this point before doing anything, you also need a copy of the reference genome.

This can be obtained from the following link at NCBI.

Alternatively, you can use the copy of the genome that is preloaded in CorGAT. For that you need to

1. navigate to Shared Data and then Data Libraries,
2. click on the library called SARS-CoV-2-REF
3. tick the file named GCA_009858895.3_ASM985889v3_genomic.fna (the only file in the library)
4. and then "export to history". The file must be exported as a dataset.

Please refer to the picture below for all of these operations.

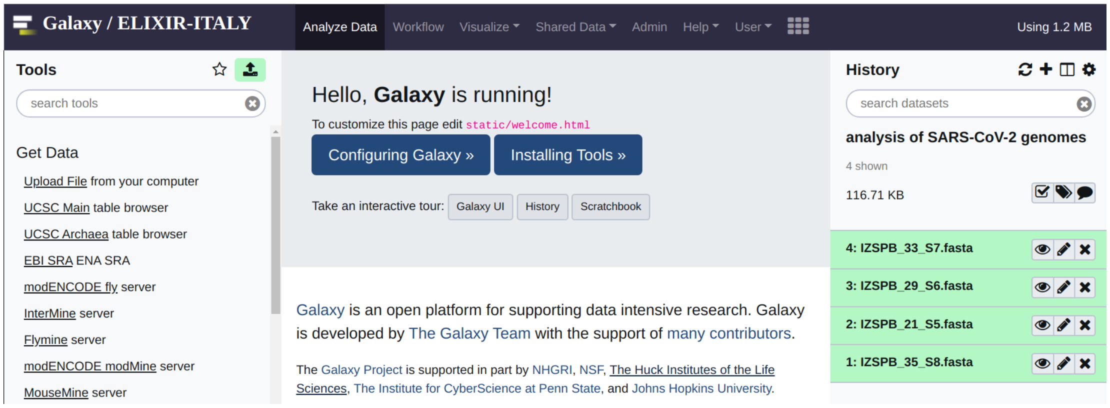


Download from web or upload from disk

You added 4 file(s) to the queue. Add more files or click 'Start' to proceed.

Name	Size	Type	Genome	Settings	Status
IZSPB_35_S8.fasta	29.2 KB	Auto-det…	----- Additional Sp…	⚙	0%
IZSPB_21_S5.fasta	29.2 KB	Auto-det…	----- Additional Sp…	⚙	0%
IZSPB_29_S6.fasta	29.2 KB	Auto-det…	----- Additional Sp…	⚙	0%
IZSPB_33_S7.fasta	29.2 KB	Auto-det…	----- Additional Sp…	⚙	0%

Type (set all): Auto-detect Genome (set all): ----- Additional Sp…



Hello, **Galaxy** is running!

To customize this page edit `static/welcome.html`

Take an interactive tour: Galaxy UI History Scratchbook

Galaxy is an open platform for supporting data intensive research. Galaxy is developed by The Galaxy Team with the support of many contributors.

The Galaxy Project is supported in part by NHGRI, NSF, The Huck Institutes of the Life Sciences, The Institute for CyberScience at Penn State, and Johns Hopkins University.

analysis of SARS-CoV-2 genomes

4 shown

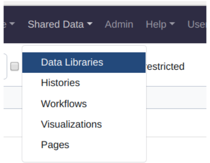
116.71 KB

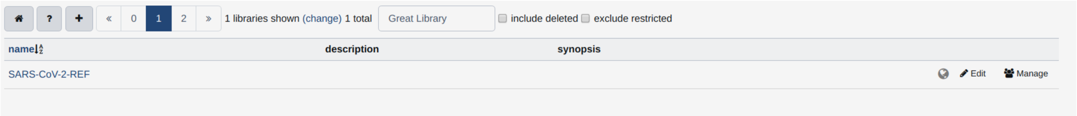
4: IZSPB_33_S7.fasta

3: IZSPB_29_S6.fasta

2: IZSPB_21_S5.fasta

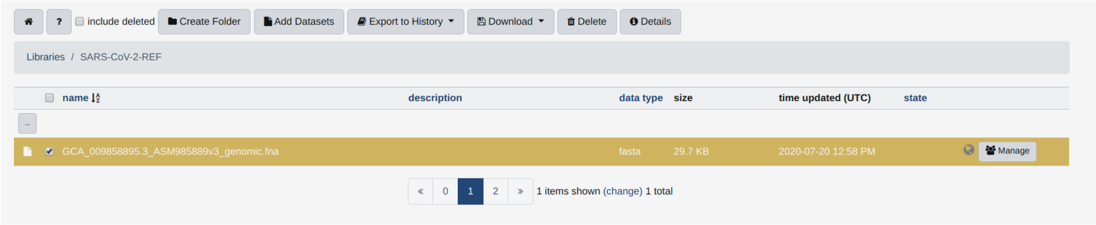
1: IZSPB_35_S8.fasta





1 libraries shown (change) 1 total

name	description	synopsis
SARS-CoV-2-REF		



	name	description	data type	size	time updated (UTC)	state
☑	GCA_009858895.3_ASM985889v3_genomic.fna		fasta	29.7 KB	2020-07-20 12:58 PM	

1 items shown (change) 1 total

CHAPTER 6

Analysing your data

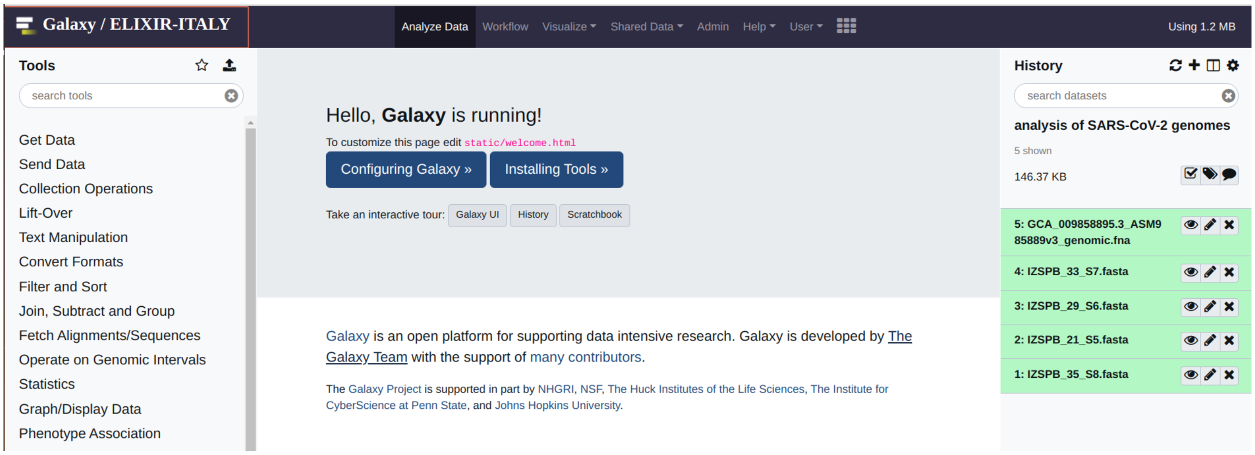
CHAPTER 7

Multiple Fasta files

Warning: Please notice, this manual provide just a quick and simple reference for the usage of the Galaxy version of CorGAT. Please refer to https://galaxyproject.org/learn/ for a complete and accurate reference on how to use Galaxy.

Once all the files have been imported, the analysis with CorGAT is very straightforward.

If everything was done according to the instruction provided in the first part of this manual, you should see something like this:



Hello, **Galaxy** is running!

To customize this page edit `static/welcome.html`

Galaxy is an open platform for supporting data intensive research. Galaxy is developed by The Galaxy Team with the support of many contributors.

The Galaxy Project is supported in part by NHGRI, NSF, The Huck Institutes of the Life Sciences, The Institute for CyberScience at Penn State, and Johns Hopkins University.

analysis of SARS-CoV-2 genomes

5 shown

146.37 KB

5: GCA_009858895.3_ASM985889v3_genomic.fna

4: IZSPB_33_S7.fasta

3: IZSPB_29_S6.fasta

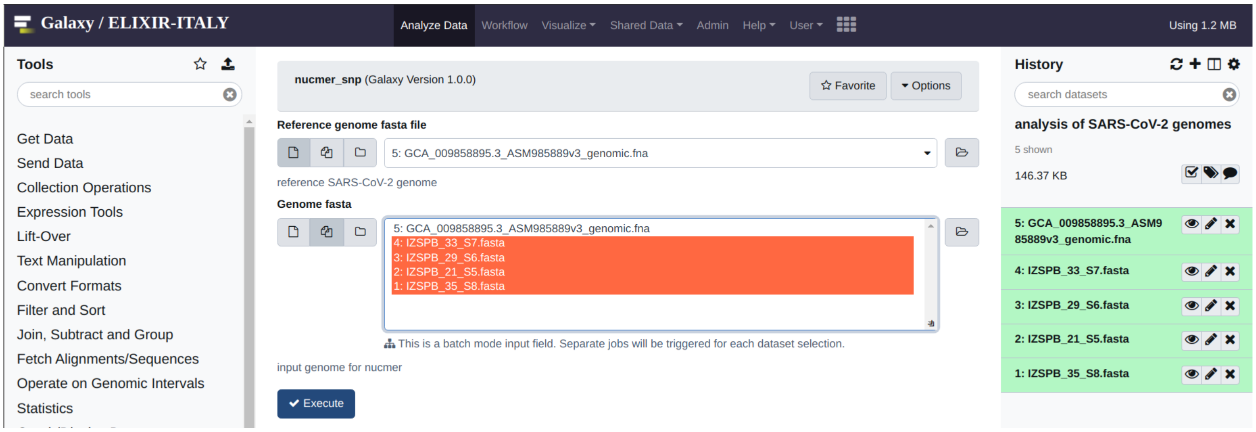
2: IZSPB_21_S5.fasta

1: IZSPB_35_S8.fasta

The first operation that you are required to do, is the alignment between your genome assemblies and the reference genome. This can be done by means of the “nucmer_snp” which is found under the “Coronavirus Genome Annotation Tool” menu. Simply click on the tool.

The interface is very simple: you are only required to indicate the reference (form on the top) and the “target” genome (form on the bottom). Multiple target genomes can be provided by clicking on the “multiple datasets icon”. Once all the “target genomes” have been selected, to run the analysis you can simply hit “Execute” (the blue button).

See below for an example:



nucmer_snp (Galaxy Version 1.0.0)

Reference genome fasta file

5: GCA_009858895.3_ASM985889v3_genomic.fna

reference SARS-CoV-2 genome

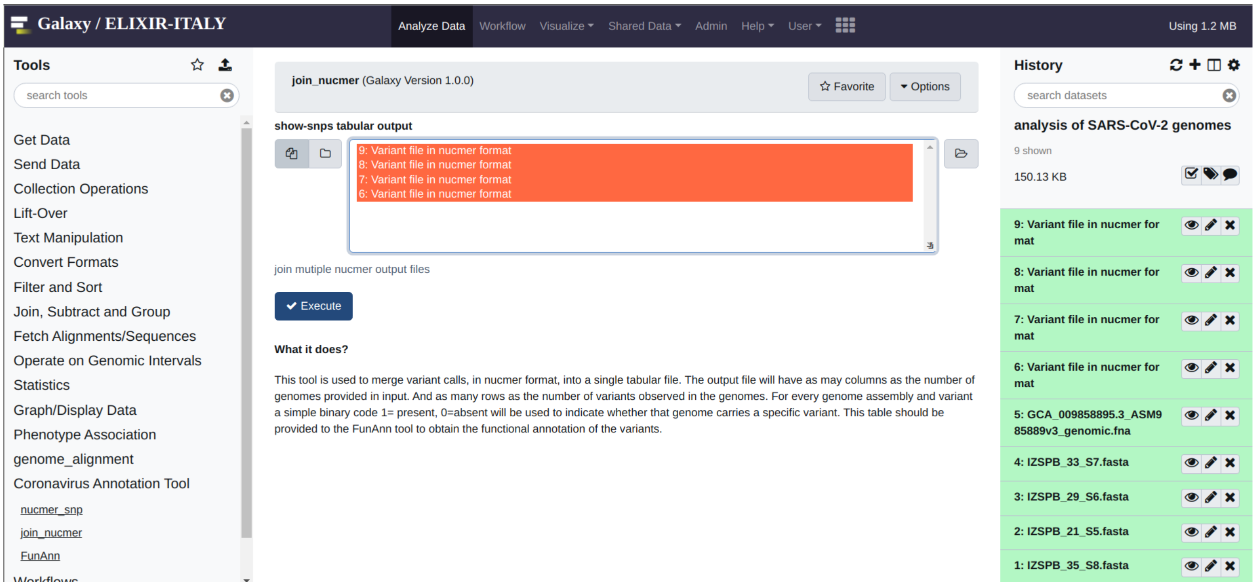
Genome fasta

5: GCA_009858895.3_ASM985889v3_genomic.fna
4: IZSPB_33_S7.fasta
3: IZSPB_29_S6.fasta
2: IZSPB_21_S5.fasta
1: IZSPB_35_S8.fasta

This is a batch mode input field. Separate jobs will be triggered for each dataset selection.

input genome for nucmer

After a brief while, you should obtain an output file for every input genome. These file need to be merged before performing the functional annotation of the variants. This operation is performed by applying the `join_nucmer` utility, again under `Coronavirus Genome Annotation Tool`. The interface of the tool is again very simple. All you need to do is to select the files that need to be merged from the form. And once ready, again hit execute.



join_nucmer (Galaxy Version 1.0.0)

show-snps tabular output

9: Variant file in nucmer format
8: Variant file in nucmer format
7: Variant file in nucmer format
6: Variant file in nucmer format

join mutiple nucmer output files

What it does?

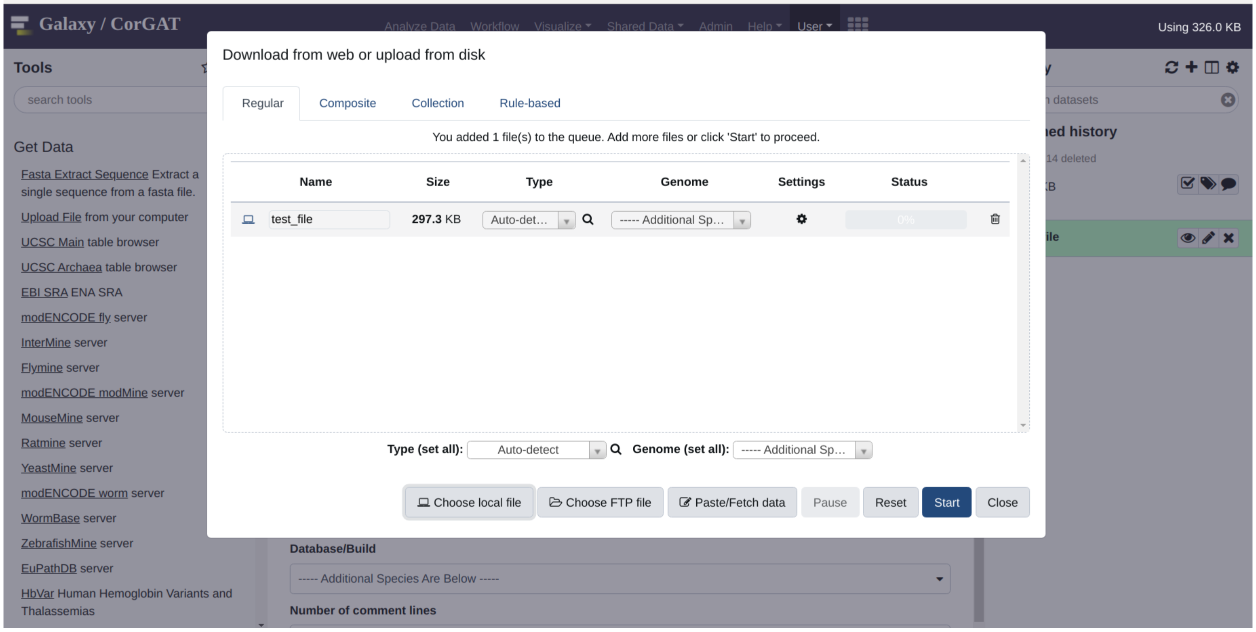
This tool is used to merge variant calls, in nucmer format, into a single tabular file. The output file will have as may columns as the number of genomes provided in input. And as many rows as the number of variants observed in the genomes. For every genome assembly and variant a simple binary code 1= present, 0=absent will be used to indicate whether that genome carries a specific variant. This table should be provided to the FunAnn tool to obtain the functional annotation of the variants.

The output will be a single file called `consolidate_variants`. This last file, will provide the input of the functional annotation tool, `FunAnn` which is found under the `Coronavirus Genome Annotation Tool` menu. The output consists in a tabular format file, where polymorphic positions are reported in the rows. And genomes, as indicated by their header in the respective fasta files, are reported in the columns. A value of 1 is used to indicate variants that are observed/present. Conversely a value of 0 zero indicate variants that are absent. Basically each column, represents the "haplotype" of the genome sequence of that particular genome.

CHAPTER 8

Using a single Multifasta file

Use the get data menue to upload a multifasta file. In this case, the file is simply called "test"



Download from web or upload from disk

You added 1 file(s) to the queue. Add more files or click 'Start' to proceed.

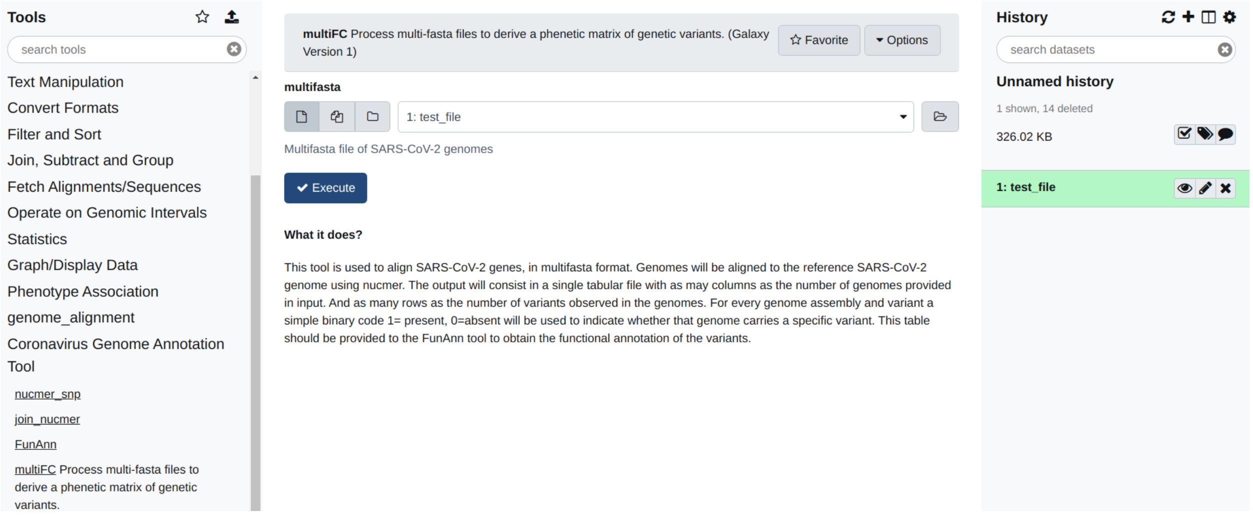
Name	Size	Type	Genome	Settings	Status
test_file	297.3 KB	Auto-det...	----- Additional Sp...		0%

Type (set all): Auto-detect Genome (set all): ----- Additional Sp...

Database/Build: ----- Additional Species Are Below -----

Number of comment lines

Select the *multiFC* utility under the "Coronavirus Genome Annotation Tool" menue. You should see something very similar to the figure you see below. This tools is very easy to use. You just need to provide a multifasta file as input. By default the tool will align all the sequences included in this file, with the reference assembly of the genome of SARS-CoV-2, and derive a phenetic matrix of presence/absence of polymorphic positions with the same file format as that produced by the `join_nucmer` utility. This file can be used to provide the input for `FunAnn` .



multiFC Process multi-fasta files to derive a phenetic matrix of genetic variants. (Galaxy Version 1)

multifasta

1: test_file

Multifasta file of SARS-CoV-2 genomes

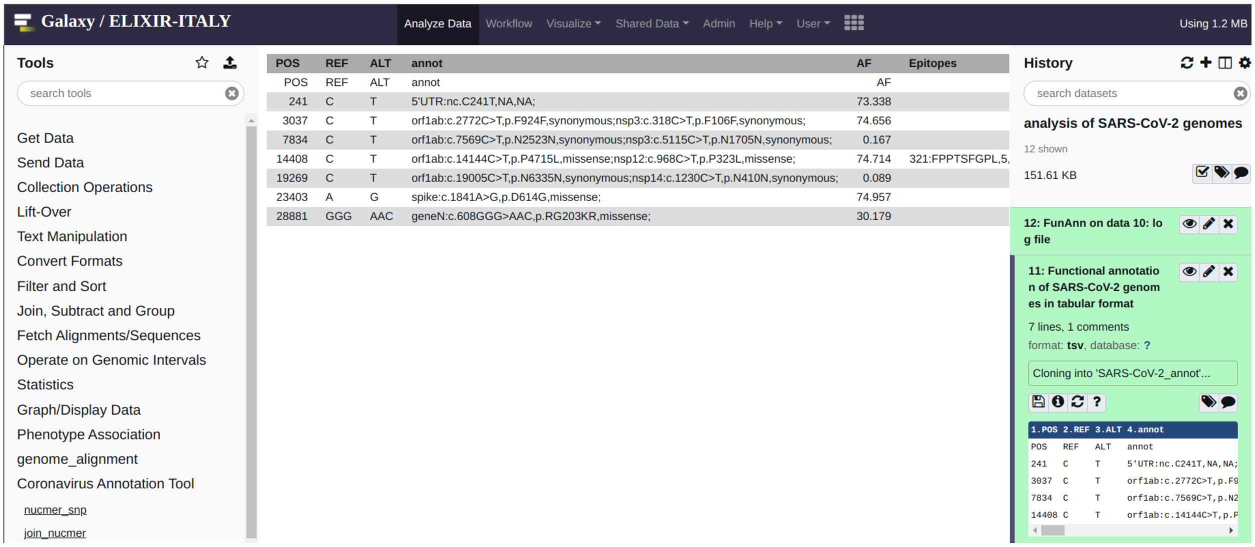
What it does?

This tool is used to align SARS-CoV-2 genes, in multifasta format. Genomes will be aligned to the reference SARS-CoV-2 genome using nucmer. The output will consist in a single tabular file with as may columns as the number of genomes provided in input. And as many rows as the number of variants observed in the genomes. For every genome assembly and variant a simple binary code 1= present, 0=absent will be used to indicate whether that genome carries a specific variant. This table should be provided to the FunAnn tool to obtain the functional annotation of the variants.

CHAPTER 9

Annotation

FunAnn takes only a single file as its input. This is the file created by `join_nucmer` or by `multiFC`. Please notice (above) that these files have the same format. To execute the functional annotation of the variants in your genome, click on the `FunAnn` tool and provide the correct input file. Then hit execute. You should obtain 2 output files. A log file (hopefully empty) which reports possible errors encountered in the execution of the software, and a tabular file with the annotations. If no errors files were encountered, you should see an output file that reads like this:



POS	REF	ALT	annot	AF	Epitopes
POS	REF	ALT	annot	AF	
241	C	T	5'UTR:nc.C241T,NA,NA;	73.338	
3037	C	T	orf1ab:c.2772C>T,p.F924F,synonymous;nsp3:c.318C>T,p.F106F,synonymous;	74.656	
7834	C	T	orf1ab:c.7569C>T,p.N2523N,synonymous;nsp3:c.5115C>T,p.N1705N,synonymous;	0.167	
14408	C	T	orf1ab:c.14144C>T,p.P4715L,missense;nsp12:c.968C>T,p.P323L,missense;	74.714	321:FPPTSFGPL,5,
19269	C	T	orf1ab:c.19005C>T,p.N6335N,synonymous;nsp14:c.1230C>T,p.N410N,synonymous;	0.089	
23403	A	G	spike:c.1841A>G,p.D614G,missense;	74.957	
28881	GGG	AAC	geneN:c.608GGG>AAC,p.RG203KR,missense;	30.179	

analysis of SARS-CoV-2 genomes

12 shown

151.61 KB

12: FunAnn on data 10: log file

11: Functional annotation of SARS-CoV-2 genomes in tabular format

7 lines, 1 comments

format: **tsv**, database: ?

Cloning into 'SARS-CoV-2_annot'...

```
1.POS 2.REF 3.ALT 4.annot
POS   REF   ALT   annot
241   C     T     5'UTR:nc.C241T,NA,NA;
3037  C     T     orf1ab:c.2772C>T,p.F9
7834  C     T     orf1ab:c.7569C>T,p.N2
14408 C     T     orf1ab:c.14144C>T,p.P
```

Congrats! If you have reached this point you should now be able to use CorGAT to annotate genomic variants in your SARS-CoV-2 genomes.

Please refer to the paper or this documentation for a more complete description of the functional annotations provided by CorGAT.

CHAPTER 10

Installing the CorGAT Galaxy

See here: CorGAT flavor for the Github repository

CHAPTER 11

How to use

- To install Docker follow this procedure.
- Run the container (i.e CorGAT) *docker run -d –privileged -p 8080:80 -p 8021:21 -p 8022:22 laniakea-cloud/galaxy_corgat:20.05*
- Log into Galaxy at http://localhost:8080 username: *admin@galaxy.org* passwd: *password*

CHAPTER 12

What to do next:

Now you have a local copy of the CorGAT Galaxy instance. Please refer to the CorGAT Galaxy manual. for tips and instructions on how to execute your analyses

CHAPTER 13

Galaxy dockers

For a more detailed refence on the usage and configuration of Docker based Galaxy instances see: https://github.com/bgruening/docker-galaxy-stable

CHAPTER 14

Indices and tables

- genindex
- modindex
- search