GRACy manual

Please meet GRACy

GRACy (Genome Reconstruction and Annotation of Cytomegalovirus) is a bioinformatics tool that can be used to analyse Illumina sequencing data of Human Cytomegalovirus (HCMV). The software is embedded in a GUI (graphic user interface), thus providing a user friendly environment which requires only limited bioinformatics knowledges.

In the current version, GRACy deals with the following tasks (described in more detail below): (a) read filtering, (b) genotyping, (c) de novo assembly, (d) annotation, (e) variant analysis and (f) read submission to public databases. However, additional modules may be added in the future and, in this regard, any feedback and suggestion from the users will be certainly taken into consideration. As many bioinformatics tools GRACy does not like spaces within file and folder names, so their use is not recommended.

Time to install the software

GRACy integrates novel algorithms with third parties' tools (e.g. aligners, de novo assemblers, variant callers, etc). This generally means that installation can get tricky. For this reason, the provided GRACy installer creates, with a single command, a new Anaconda environment where all the software and dependencies are installed with a specific version number. While keeping the installation easy, this method requires around 7Gb of free space on your hard disk. Ok, let's start.

After unzipping the GRACy package on your computer you can enter the GRACy folder and run the installer:

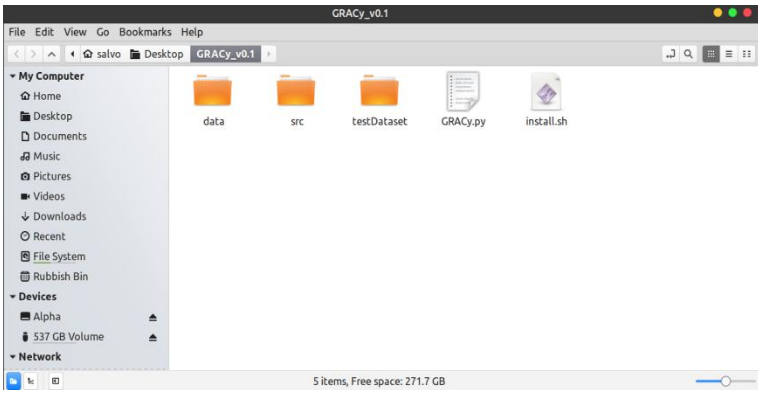
```
cd GRACy-0.1
bash install.sh
```

The installer will download and install all the required packages and for this reason an internet connection will be necessary. The entire process may take up to 20 minutes (or even more, depending on your internet connection).

Congratulations! You have installed GRACy on your computer!

What's inside?

After the installation is complete, you will find several files and folder in your GRACy-0.1 folder:



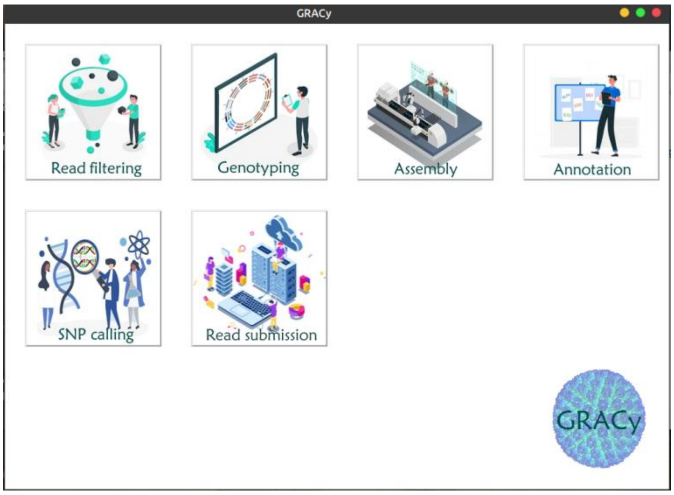
5 items, Free space: 271.7 GB

The data and src folders contain everything is needed to perform the tasks described below and should not be modified. The testDataset contains two paired end simulated read datasets that we will use to illustrate how the software works. Reads merlin_1.fastq and merlin_2.fastq represent reads from the HCMV reference genome, whereas merlinVar_1.fastq and merlinVar_2.fastq refer to a genome obtained by introducing in silico 4,000 SNPs to the merlin reference genome.

To launch GRACy, just open a terminal, navigate to the GRACy folder and type:

```
./GRACy.py
```

This could be the last time you use the terminal while working with GRACy. The main workplace will look like this:



Click on any of the task icons to start running some analysis with GRACy.

Filtering the reads.

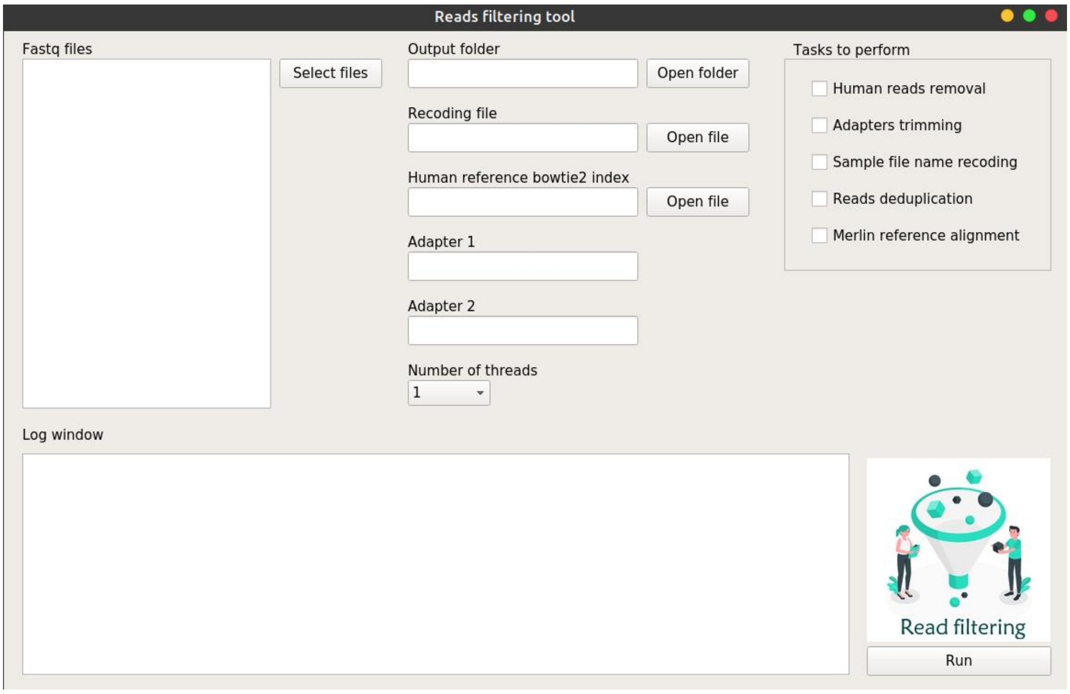
Before starting any kind of analysis, reads can be filtered according to the following criteria:

(1) Reads can be trimmed to remove sequencing adapters

(2) Reads containing human sequences (as expected when working with clinical samples for example) can be removed from the dataset. This is useful to remove bulky information that is not useful for the HCMV analysis, understanding the proportion of the reads that are attributable to HCMV and remove sensitive human data from your reads before submitting them to a public database.

(3) Duplicated fragments that may be produced by extensive use of PCR amplification during the DNA library preparation can be removed

When launching the reads filtering module, the following form will open:



Number of threads
1

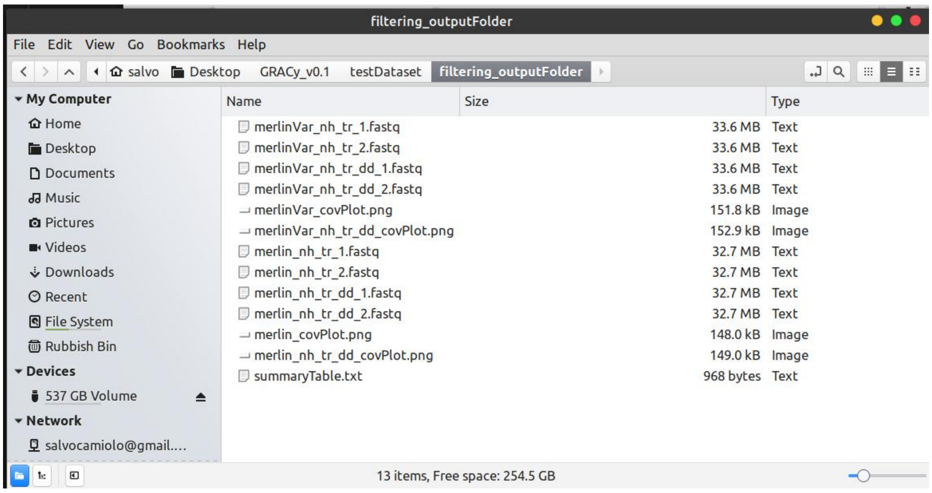
Let’s use it to perform a full filtering process while using the reads in the testFolder directory. Click on “Select files” and navigate to the testFolder and then Reads. You can select multiple files by pressing CTRL (or SHIFT) key while clicking. Let’s select all the files in the folder and press Ok.

Datasets are reported in the “Fastq file” text area (only the prefix of each read pair is reported). Since we want to go through all the filtering steps, let’s tick all the check boxes in the “Tasks to perform” box with the exception of “Recoding file” at the moment (we will see what this is for in a moment). Now we need to provide some additional information before clicking the “Run” button. We need to select an output folder where all the produced files will be stored, and a human bowtie2 index. The latter is necessary as we want to remove the human reads from our samples. A bowtie2 index is a group of files that are used to make the alignment on the human reference genome faster. Such files are quite big and can not be included with the GRACy installation. However you can download and extract a copy at the bowtie2 web page (http://bowtie-bio.sourceforge.net/bowtie2/manual.shtml) in the right section “Indexes” by selecting the file *H. sapiens*, NCBI GRCh38. The file GRACy needs is that ending with the _index.1.bt2, so select this file after clicking the “Open” button. Ok, we are all set. If you want, you can add your adapter sequences in the corresponding boxes (if not, GRACy will use the standard Illumina adapters) and select the number of threads you want to run the software on. Now click the “Run” button and the filtering process will start. In the log area you will see the steps that are performed and an arrow in the “Fastq file” area will indicate which datasets is being analysed.

If the Illumina sequencing machine provided file names that are not meaningful for you and you want to change them, you can do so by ticking the “Sample file name recoding” and providing a “Recoding file” that is a text tab formatted two column files with the original name and the new name being reported in the first and second column respectively.

The files that are reported in the output folder will contain suffixes in their name that are indicative of the steps that have been performed on the original dataset. More precisely, tr = adapter trimmed, _dd = deduplicated, nh = not human reads present. So if an output file has a name ending with _nh_tr_dd_1.fastq, that means that the original read file has been purged from the human reads, deduplicated and the adapter trimmed.

Since we also ticked the “Merlin reference alignment” check box, the filtered reads will be aligned to the merlin reference genome and a coverage plot will be produced. This plot allows to visualize how the reads are distributed along the genome, if some portion is underrepresented or if there is some peak of coverage that may be due to contamination. This is how the output folder should look like at the end:



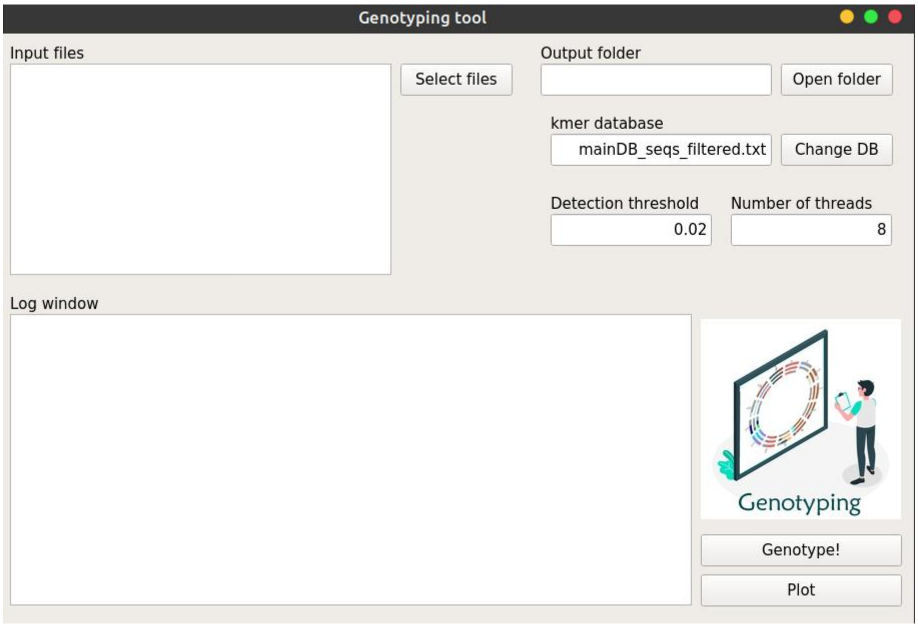
Name	Size	Type
merlinVar_nh_tr_1.fastq	33.6 MB	Text
merlinVar_nh_tr_2.fastq	33.6 MB	Text
merlinVar_nh_tr_dd_1.fastq	33.6 MB	Text
merlinVar_nh_tr_dd_2.fastq	33.6 MB	Text
merlinVar_covPlot.png	151.8 kB	Image
merlinVar_nh_tr_dd_covPlot.png	152.9 kB	Image
merlin_nh_tr_1.fastq	32.7 MB	Text
merlin_nh_tr_2.fastq	32.7 MB	Text
merlin_nh_tr_dd_1.fastq	32.7 MB	Text
merlin_nh_tr_dd_2.fastq	32.7 MB	Text
merlin_covPlot.png	148.0 kB	Image
merlin_nh_tr_dd_covPlot.png	149.0 kB	Image
summaryTable.txt	968 bytes	Text

13 items, Free space: 254.5 GB

The trimmed fastq reads that did not align to the human reference genome are reported, before and after the deduplication (nh_tr and nh_tr_dd respectively). The coverage plots are produced for these two datasets and a text summuryTable.txt reports the number of reads remaining after each filtering step together with the breadth of coverage on the merlin reference genome.

The genotyping tool

When selecting the Genotyping tool from the GRACy main workspace the following window will open:


Genotyping tool

Input files

Output folder

kmer database: mainDB_seqs_filtered.txt

Detection threshold: 0.02

Number of threads: 8

Log window

The genotyping tool is designed to look, within you reads, for motifs that are specific to reported genotypes of 13 HCMV hypervariable genes. To run this tool, just choose the reads you want to analyse as previously described for the filtering tool by clicking on the “Select files” button. You need to provide, as usual, an outputFolder where the produced files will be stored. Finally, you can specify the number of threads you want to use for the analysis and the detection threshold. The detection threshold refers to the minimum coverage a certain genotype must have in order to be reported in the output file, and it is calculated as a proportion of the average coverage on the entire genome. In brief, GRACy will align the reads to the merlin reference genome and calculate the average coverage by only considering the covered portions. The default detection threshold value of 0.02 implies that a genotype is reported only if confirmed by a number of reads equal or higher than the 2% of the average coverage on the merlin reference genome. The kmer database can also be specified. Although only the mainDB_filtered_seqs.txt database is available in the current GRACy version, support for the generation of custom produced kmer databases may be provided in the future. For now, we can leave it as it is.

Let’s give it a go. Click on the “Select files” button and select all the files in the testDatasets/reads folder. Choose an output folder and click the “Genotype” button.

After a couple of minutes the task is completed and you will find two files for each reads dataset in the output folder. The IDCard.txt file reports the genotype at each analysed hypervariable gene. There is a line for each gene that looks like this (taken from the merlin_1_IDCard.txt):

```
rl5a                    G1        1.0     121       2
```

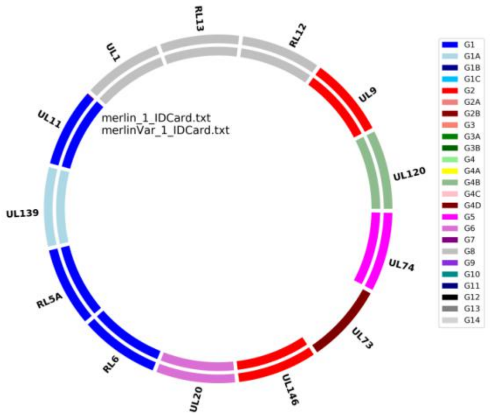
The first field is the gene name, the second is the detected genotype, the third is the proportion of the genotype, the forth is the number of reads that aligned on the gene and the fifth is the number of specific kmers that were found in the reads for that genotype. The four fields from 2 to 5 are repeated for each genotype present in the reads dataset for that particular gene (e.g. different genotypes are expected in samples originated by a multiple strain infection for example). In this specific case, for gene RL5A (field 1) the G2 genotype was detected (field 2) in 100% of the reads (field 3) as confirmed by 121 reads (field 4) that contained 2 (field 5) of the specific kmers used for the genotype of this gene/genotype combination.

An important note now on the interpretation of the result. If you multiply fields 3 and 4 you obtain the number of reads that aligned on the gene with a certain genotype, in the above example, that correspond to 121 as only G1 was detected for gene RL5A. However, have a look at gene UL11 in the same file:

```
ul11   G2  0.0080213903    374    10            G1     0.9919786096    374   26
```

It seems like our sample contains two genotypes at this locus. But if we multiply fields 3 and 4 for genotype G2 we obtain 0.008 * 374 = 3. So only 3 reads confirm this genotype, a number that is quite low to be considered reliable, as few low quality reads may match a genotype just by chance. In conclusion, always consider the number of reads calling a genotype before inferring its presence.

GRACy implements a plotting tool that allows the visualization of all the hypervariable gene genotypes in a donut plot. Several datasets can be reported in the same plot thus allowing an easy comparison at a gene level. After selecting the output folder click the “Plot” button. Select the two IDCard.txt files that have just been produced and click “Open”. You will be required to insert a name for the image file GRACy will produce and click “Ok”. Just insert a name without extension as GRACy will append the .png to the file name while generating the figure. In the output folder you will find the produced image that should look like the following:

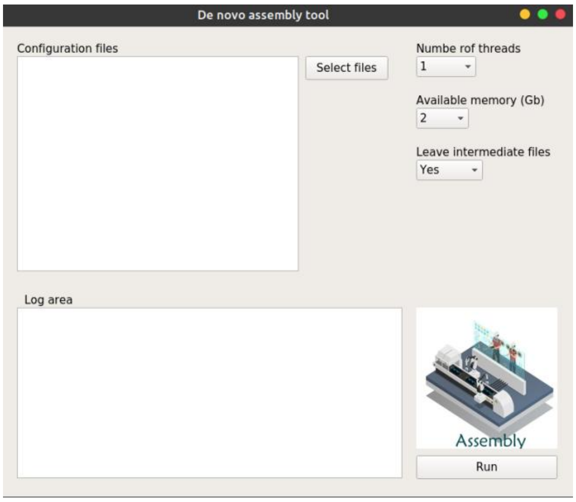


Inside the donut plot you see a list of the analysed datasets representing the plots from outward to the inward.

The two analysed dataset are both derived from the merlin reference genome and it is not surprising that colours for each gene are the same. However, dataset merlinVar was generated from a highly modified merlin genome (4000 SNPs) and in this situation no specific kmer was found for UL73. This gene feature few specific kmers that are not matched due to the introduced SNPs. Low viral loads or the usage of sub-optimal enrichment kits during DNA library preparation may also lead to missing segments in the genotyping plot.

The de novo assembly tool

This module can be used to reconstruct the HCMV genome sequence directly from the reads data. When clicking the Assembly button in the main GRACy workspace the following form will open:



De novo assembly tool

Configuration files

Numbe rof threads: 1

Available memory (Gb): 2

Leave intermediate files: Yes

Log area

The interface is simple and you have to only provide a list of configuration files, one for each dataset you want to perform the de novo assembly on. A copy of the configuration file (assembly.conf) is reported in the GRACy folder “data” of the distribution.

Create a new folder and make two copies of the assembly.conf file in it, let’s call them merlin_assembly.conf and merlinVar_assembly.conf). Now you can open these files with a text editor and start to modify it. The main (and possibly only) lines you need to modify are the line 1 where you insert the project (dataset) name (this will also be the output folder name for the intermediate files) and lines 2 to 5 where the full path of the reads you want to use should be reported. Line 2 and 3 (Read1_toAssemble and Read2_toAssemble) are the path to the reads 1 and 2 of the sample you want to run the assembly on. Line 4 and 5 (Read1_toFill and Read2_toFill) could be the path to the same reads or it can be used to include reads generated, for example, from other samples of the same patient. Briefly, GRACy will try to reconstruct the genome as one single scaffold covering the entire sequence. However, lack of reads in specific portions may cause gaps that are impossible to resolve (i.e. the GC rich ORI regions sometimes feature a lower coverage than the rest of the genome). If you have multiple samples for the same patient, you may consider concatenating them together and provide the path of these two joined files in line 4 and 5. For this example let’s just pretend to have one single sample.

Lines 7, 15,17,19,21 and 23 allow the user to switch on and off the different processes GRACy goes through during the assembly. Each step is performed in a specific folder and if you switch on only specific steps, GRACy will expect the folder (and relative files) for the previous step to be present in the output folder. Lines 8 to 13 allow the user to insert the parameters for the reads quality filtering step. Default values should be ok for most Illumina data

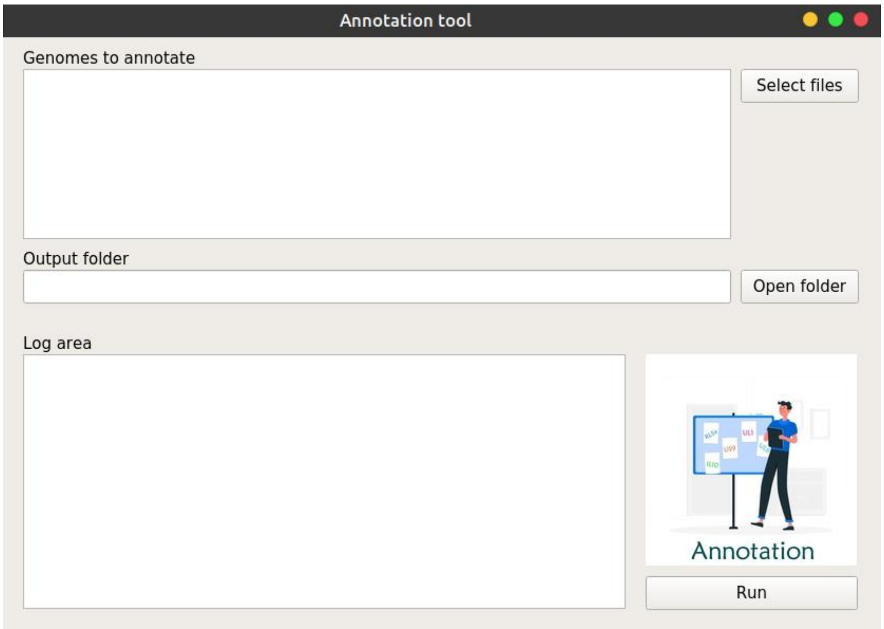
My configuration file for the merlin dataset will look like this (the two columns are tab separated):

```
Project_name                  merlin
Read1_toAssemble              /home/salvo/Desktop/GRACy_v0.1/testDataset/reads/merlin_1.fastq
Read2_toAssemble              /home/salvo/Desktop/GRACy_v0.1/testDataset/reads/merlin_2.fastq
Read1_toFill                  /home/salvo/Desktop/GRACy_v0.1/testDataset/reads/merlin_1.fastq
Read2_toFill                  /home/salvo/Desktop/GRACy_v0.1/testDataset/reads/merlin_2.fastq
#Perform reads quality filtering
Reads_quality_filter          yes
minQualMean                   28
trimLeft                      7
trimQualRight                 28
trimQualWindow                30
trimQualStep                  5
minLen                        50
#Perform denovo assembly
Denovo_assembly               yes
#Perform Scaffolding
Scaffolding                   yes
#Perform first consensus call
First_Consensus_call          yes
#Refine assembly
Refine_assembly               yes
#Create second consensus
Second_consensus_call         yes
```

I am using the original reads, but for clinical samples remember to use the filtered reads obtained with the reads filtering module. Now, modify the merlinVar_assembly.conf as shown above by using merlinVar as project name and the full path the merlinVar_1.fastq and merlinVar_2.fastq reads file. Click the “Select files” button and select the two configurations files. Select the number of threads you want to use and the amount of available memory (this is very important as if you specify more memory than you can actually allocate the software will generate an error and exit). You can leave the intermediate files drop-down menu to yes. Click the “Run” button and wait for the process to complete. This may require between 20 minutes and several hours depending on machine processor, the number of used reads and allocated threads and memory. The merlin dataset ran on an i7 computer on 2 threads / 2Gb memory in around 10 minutes. The produced genomic sequence, together with the folder with all the intermediate files, will be reported in the same folder where the configuration files were stored (the merlin genome was perfectly reconstructed!)

The annotation tool

Ok you assembled your genome from a clinical sample. Now it’s time to see where the genes are and investigate whether there is any disrupting mutation. The annotation tool performs exactly this task. When you open the tool from the GRACy main workspace the following window will open:



The only data you need to provide here is a list with the genome sequences you want to annotate and an output folder. Try to open the merlin_genome.fasta and the merlinVar_genome.fasta that you de novo assembled with the previous module and choose your output folder. Then click the “Run” button and wait until all the 160 annotated HCMV genes are searched within the provided genomic sequences. After the elaboration is finished you will find several files for each analysed genome in your output folder:

(1) The fasta file containing the coding sequences

(2) The fasta file containing the protein sequences

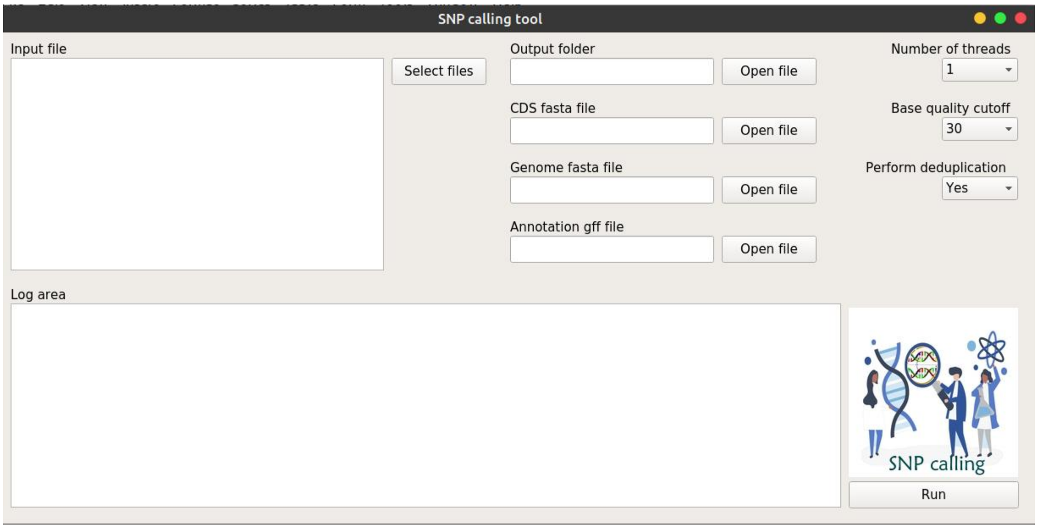
(3) The gff3 formatted annotation file

(4) The annotationWarning file

The annotationWarning file is something that deserves your attention after the annotation process is finished. In facts, for each protein GRACy looks for the best match within the genome and if either a valid start or stop codon are missing, it will look for these signals in the close proximity so that a longer or shorted protein may be generated. Moreover, after the coding sequence has been predicted, GRACy will scan it to be sure that it includes only complete triplets (e.g. no disrupting INDEL are present) while not containing any premature stop codon within it. The annotationWarning file reports for each gene the variations GRACy predicted (as compared to the deposited sequences the software use to identify the genes) and the presence of eventual disrupting mutations.

The SNP calling tool

This tool is useful to detect variants between two different strains or if you want to study the variation of the SNPs frequency in the same patient at different time points. When you click on the SNP calling module in the main GRACy workspace the following window will open:



Number of threads: 1

Base quality cutoff: 30

Perform deduplication: Yes

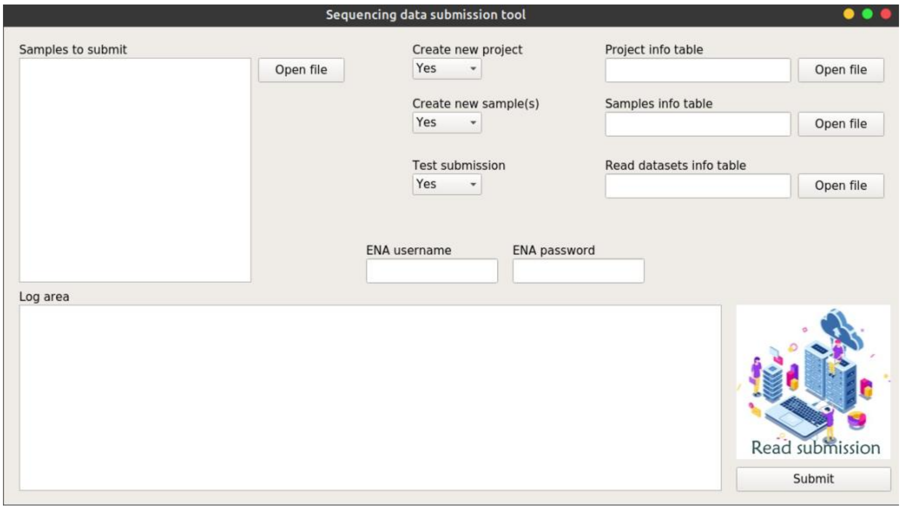
This module takes in input, as usual, Illumina fastq files and aligns the corresponding reads to a reference genome (this may be the genome of a strain you want to compare your reads to, or the consensus sequence of the patient genome you are studying). The alignment is followed by a deduplication step and the actual SNP calling. GRACy also assigns each SNP to the coding sequence it occurs in and output a table with the effect it has on the coded protein (e.g. it classifies SNPs in synonymous and non synonymous while computing the amino acid variation for the latter case).

Let’s try it out by calculating the SNPs for the merlinVar reads while using the merlin genome as the reference. Click on the “Select files” and select the reads merlinVar_1.fastq/merlinVar_2.fastq in the testDataset/reads folder. Now you need to select an output folder and, for the reference genome, a fasta file with the genome sequence, a fasta file with the coding sequences and a gff formatted annotation file. Since we just retrieved this data when we used the previous Annotation module let’s select the assembled merlin_genome.fasta, and the annotated merlin_genome.fasta_cds.fasta and merlin_genome.fasta_annotation.gff files. Finally, let’s select the number of threads we want to used, let’s leave the “Perform deduplication” drop-down menu to “Yes” (you can exclude this step although it is not recommended) and let’s leave the “Base quality cut-off” drop-down menu to 30 (this is the minimum phred quality a base should have in order to be considered during the SNP calling process). Now click the “Run” button and wait for the process to complete.

In the output folder you will find a file reporting all the found SNPs with their frequency (file _SNPfreq.txt) and a table with the effect on the protein sequences of all those SNPs that occurred in the coding sequences (file _SNPeffect.txt)

The read submission tool

This tool makes a little bit easier to submit your new HCMV reads to the European Nucleotide Archive (ENA) database. When you click on this module icon within the main GRACy workspace the following window will open:



Create new project: Yes
Create new sample(s): Yes
Test submission: Yes

It is important to note that the reads you want to submit needs to be already on ENA web space. In order to do that, you need to transfer them by using one of the methods reported at the ENA website https://ena-docs.readthedocs.io/en/latest/submit/fileprep/upload.html

After you reads have been successfully uploaded you can proceed to the submission process with GRACy. When submitting reads to ENA you may want to create a Sample for that dataset where other information may be added in the future (e.g. annotation, analysis etc). Moreover, when working with a big project, you may want to create a Project (aka Study) on ENA where all the Samples are kept together for easy access.

In the “data” folder of your GRACy distribution you will find the Excel formatted file ENA submission worksheet.xlsx . This file is divided in tabs each reporting a form you need to fill in order to create a new Project, new Samples (optional if your reads already have a project and sample registered in ENA) and new Reads (compulsory) with the info relative to the fastq file you want to submit. Fill the forms, and save them in a tab delimited text format on you computer.