1. SPiGA PROJECT

Development of the Graphical User Interface SPiGA:

We have developed a number of tools for the analysis of variability of genome sequences (obtained using high-throughput methodologies). The objective of this interface is to make an application that put into a package all these programs, in order to facilitate the use to the users. The graphical Interface will simply collect the desired actions with their corresponding options. Then, the user will have the choice to save all the orders into a file (with extension .sh) or execute them directly into a terminal.

Our choice is to use Qt for developing the GUI.

We have designed the windows and screens that we want to show in the program (using Qt Designer). Now we need to give functionality to all these windows (what every menu and button do). The application must work in computers with macos and linux OS (Debian and Redhat architectures), at least.

What is necessary to do:

construct the application named SPiGA, including an icon that can be clicked directly. Give actions to all designed menus, buttons, etc. included in the windows of the program.

2. SPiGA: PROGRAMS TO CALL

In principle, SPiGA, is just agraphical interface. Associated programs can be called externally using the terminal window.

SPiGA has a repository in Github including the files we have up to now:

```
git clone https://github.com/CRAGENOMICA/SPiGA.git
```

The programs associated to SPiGA are these ones:

```
git clone https://github.com/CRAGENOMICA/mstatspop.git
git clone https://github.com/CRAGENOMICA/fastaconvtr.git
git clone https://github.com/CRAGENOMICA/weight4tfa.git
git clone https://github.com/CRAGENOMICA/gVCF2tFasta.git
git clone https://github.com/CRAGENOMICA/indexingtFasta.git
git clone https://github.com/CRAGENOMICA/concatenate_tFasta.git
git clone https://github.com/CRAGENOMICA/mergetFasta.git
git clone https://github.com/CRAGENOMICA/ms2geno.git
git clone https://github.com/CRAGENOMICA/DIGUP.git
git clone https://github.com/CRAGENOMICA/Tang_Rsb.git
```

3. FIRST PROTOTYPE: GSAW in QtPython

The first prototype is also in Github:. It may be useful for comparing some functions.

```
git clone https://github.com/CRAGENOMICA/GSAW.git
```

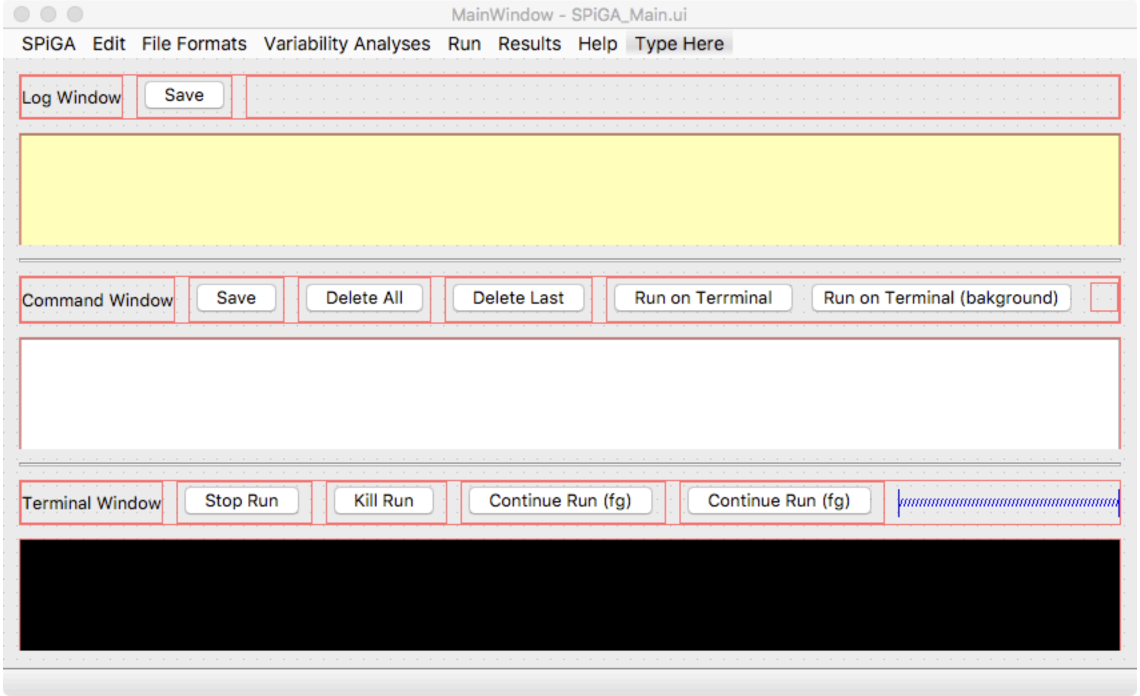
4. SPiGA: DESIGNED WINDOWS:

We have designed a number of graphical windows for SPiGA, using QtDesigner:

SPiGA_Main.ui
SPiGA_About.ui
SPiGA_fasta2tfasta.ui
SPiGA_fasta2ms.ui
SPiGA_tfasta2fasta.ui
SPiGA_tfasta2ms.ui
SPiGA_fasta2fasta.ui
SPiGA_gVCF2tfasta.ui
SPiGA_ms2geno.ui
SPiGA_Indexingtfasta.ui
SPiGA_Jointfasta.ui
SPiGA_weight4tfasta.ui
SPiGA_ResultsTable.ui
SPiGA_Contributors.ui
SPiGA_mstatspop4tfasta.ui
SPiGA_mstatspop4fasta.ui
SPiGA_mstatspop4ms.ui
SPiGA_Tang_Rsb.ui
SPiGA_DIGUP.ui

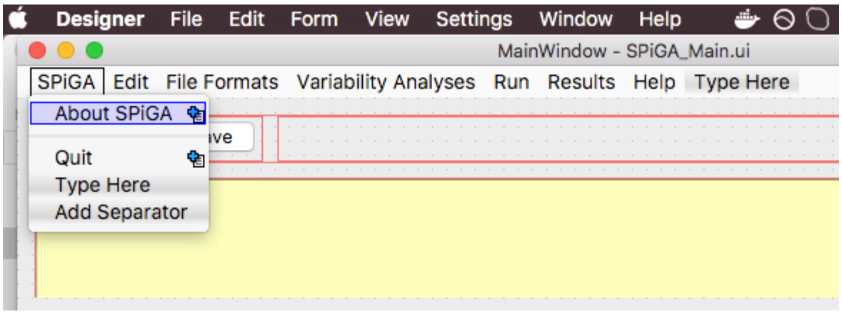
The **SPiGA_Main.ui** is the main page and will be only closed when the user Quits the program. It contains all the menus. The rest of windows are opened additionally to the Mai window.

4.1. SPiGA_Main Window:

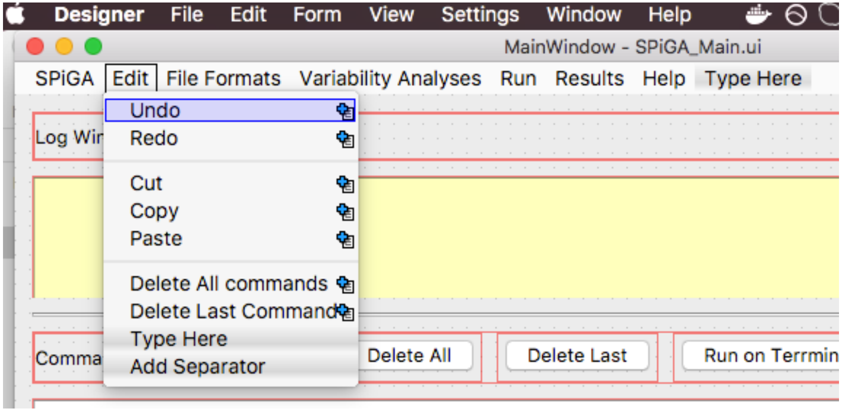


The main window is separated in a (i) log Window, (ii) Command window and (iii) Terminal Window and (iv) the Menu,

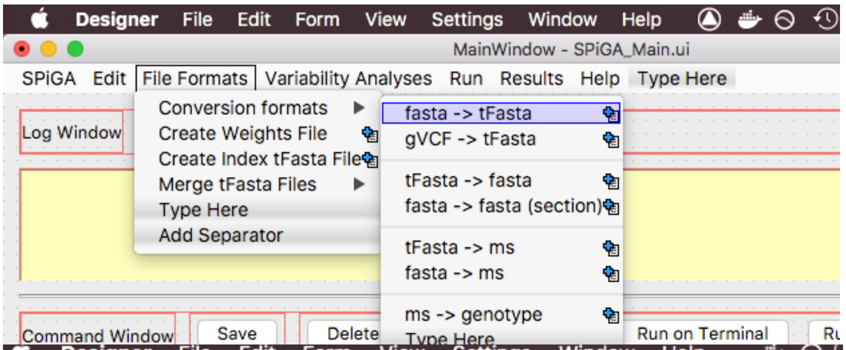
1. The log Window will contain a list of all the actions that the user will do with SPiGA (all those that implicate an action to the Command window), in sequential order. There is a button with the option to save the log file.
2. The Command Window will contain all the commands that the user defines using the menus. The user has the options to Save into a .sh file, delete all commands, delete only the last one and finally, the possibility to send to the Terminal window in foreground or in background, to be executed.
3. The Terminal Window will show the commands that run on Terminal (past and present). There are buttons to Stop (Ctrl Z), Kill (Ctrl C) and to continue the run (fg or bg).
4. The Menu contains a number of Basic functions (Copy, Paste, etc..) plus the way to access to all the rest of Windows.
   a. SPiGA Menu: contains the Window “SPiGA_About.ui” and the Command Quit.

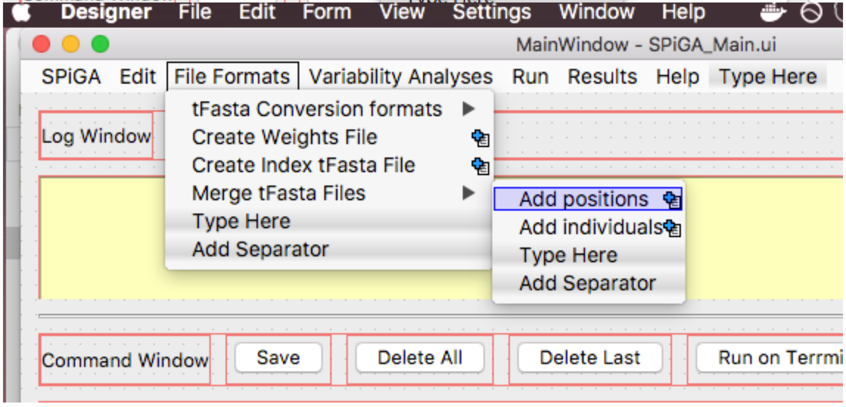

Designer File Edit Form View Settings Window Help
MainWindow - SPiGA_Main.ui
SPiGA Edit File Formats Variability Analyses Run Results Help Type Here
About SPiGA
Quit
Type Here
Add Separator
ve

b. Edit Menu: Options for Copy, Paste etc..

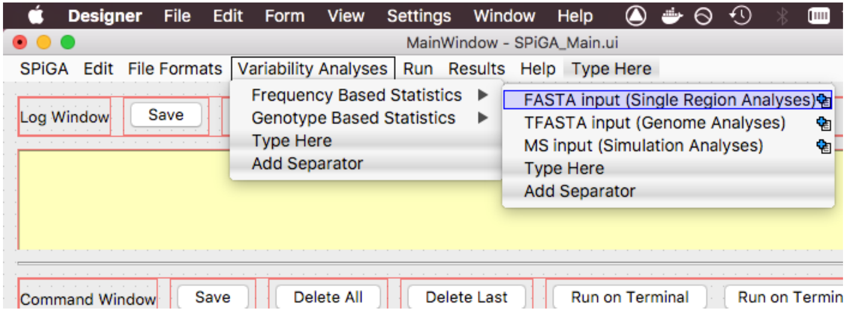

Designer File Edit Form View Settings Window Help
MainWindow - SPiGA_Main.ui
SPiGA Edit File Formats Variability Analyses Run Results Help Type Here
Undo
Redo
Cut
Copy
Paste
Delete All commands
Delete Last Command
Type Here
Add Separator
Log Wir
Comma
Delete All
Delete Last
Run on Termin

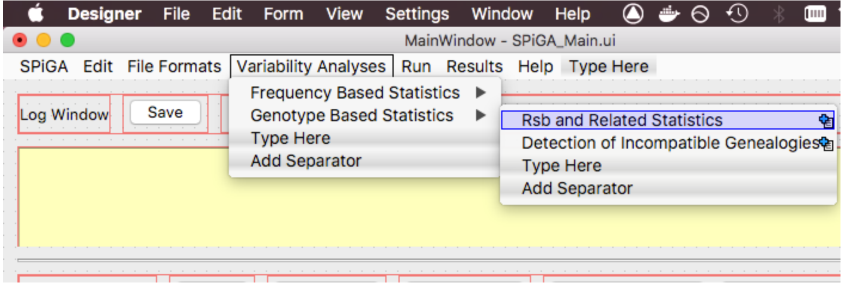
c. File Formats: Most of windows for commands are here: the submenu Conversion Formats has seven additional windows (example, SPiGA_fasta2tfasta.ui, SPiGA_gvcf2tfasta.ui, etc…), each one with different commands and flags. The other three, “Create weights File”, “Create Index tfasta File” and “Merge tfasta File”. This last one contains two additional windows as well.



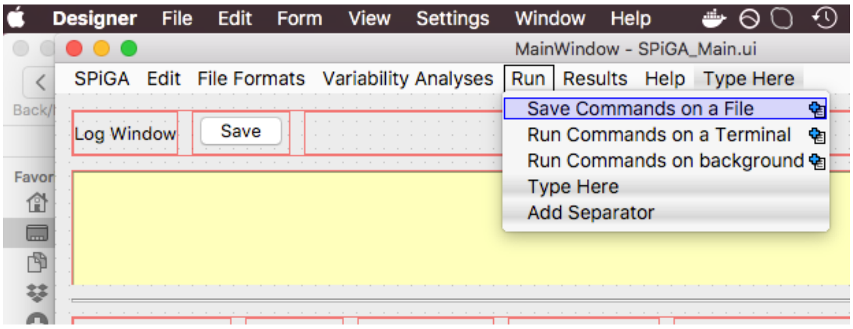


d. The main core of the analysis is the Menu “Variability analyses”. Right now, only contains two submenus (Frequency based statistics and Genotype based statistics). The analyses for Frequency contains three windows.The Anlyses for Genotype contains two windows.

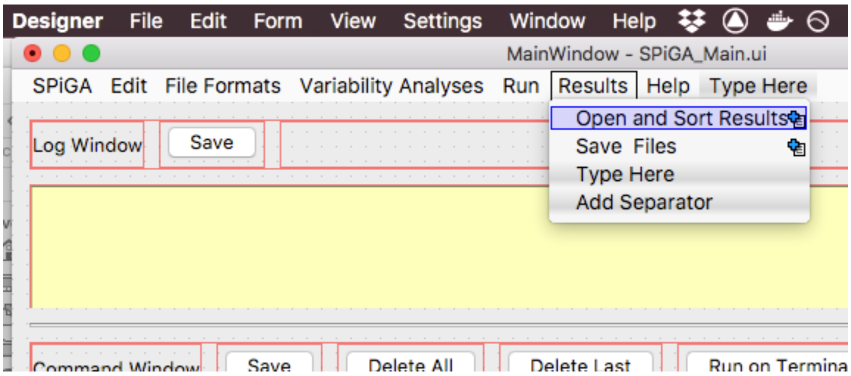

Designer File Edit Form View Settings Window Help
MainWindow - SPiGA_Main.ui
SPiGA Edit File Formats Variability Analyses Run Results Help Type Here
Frequency Based Statistics
Genotype Based Statistics
Type Here
Add Separator
FASTA input (Single Region Analyses)
TFASTA input (Genome Analyses)
MS input (Simulation Analyses)
Type Here
Add Separator
Log Window
Save
Command Window
Save
Delete All
Delete Last
Run on Terminal
Run on Termin


Designer File Edit Form View Settings Window Help
MainWindow - SPiGA_Main.ui
SPiGA Edit File Formats Variability Analyses Run Results Help Type Here
Frequency Based Statistics
Genotype Based Statistics
Type Here
Add Separator
Rsb and Related Statistics
Detection of Incompatible Genealogies
Type Here
Add Separator
Log Window
Save

e. The Run Menu is essentially the same that is in the Main window, located on the Command Window.

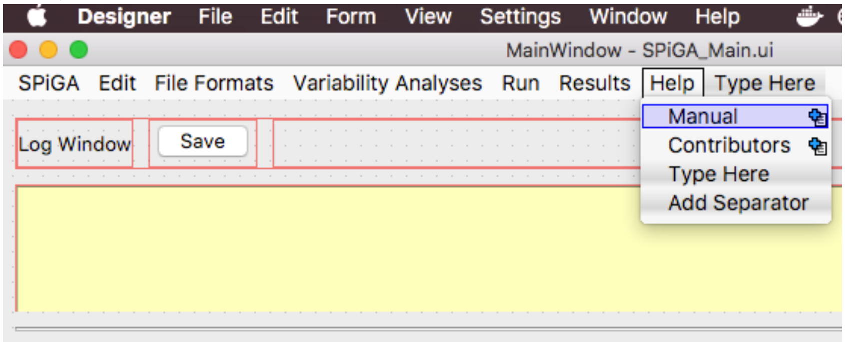

Designer File Edit Form View Settings Window Help
MainWindow - SPiGA_Main.ui
SPiGA Edit File Formats Variability Analyses Run Results Help Type Here
Save Commands on a File
Run Commands on a Terminal
Run Commands on background
Type Here
Add Separator
Log Window
Save

f. The Results Menu contains just a new window with commands about open text files, selection of columns (if tabulated) and saving new results.


Designer File Edit Form View Settings Window Help
MainWindow - SPiGA_Main.ui
SPiGA Edit File Formats Variability Analyses Run Results Help Type Here
Open and Sort Results
Save Files
Type Here
Add Separator
Log Window
Save
Command Window
Save
Delete All
Delete Last
Run on Termina

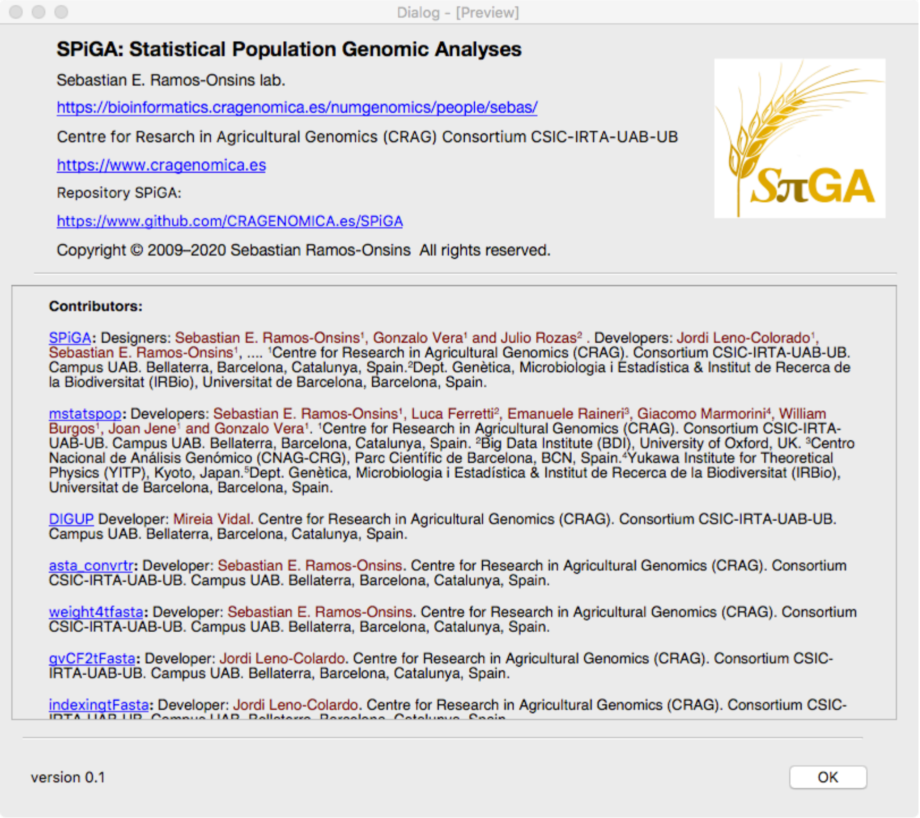
g. Finally, the Help Menu will contain a manual. We have constructed one for GSAW in LaTeX, and we will use one very similar for SPiGA.

Also, we will need a window for the final Contributors.



4.2. SPiGA_About.ui

Informative window. Close the Window when press OK.



SPiGA: Statistical Population Genomic Analyses

Sebastian E. Ramos-Onsins lab.

https://bioinformatics.cragenomica.es/numgenomics/people/sebas/

Centre for Resarch in Agricultural Genomics (CRAG) Consortium CSIC-IRTA-UAB-UB

https://www.cragenomica.es

Repository SPiGA:

https://www.github.com/CRAGENOMICA.es/SPiGA

Copyright © 2009–2020 Sebastian Ramos-Onsins All rights reserved.

Contributors:

SPiGA: Designers: Sebastian E. Ramos-Onsins[1], Gonzalo Vera[1] and Julio Rozas[2] . Developers: Jordi Leno-Colorado[1], Sebastian E. Ramos-Onsins[1], ... [1]Centre for Research in Agricultural Genomics (CRAG). Consortium CSIC-IRTA-UAB-UB. Campus UAB. Bellaterra, Barcelona, Catalunya, Spain.[2]Dept. Genètica, Microbiologia i Estadística & Institut de Recerca de la Biodiversitat (IRBio), Universitat de Barcelona, Barcelona, Spain.

mstatspop: Developers: Sebastian E. Ramos-Onsins[1], Luca Ferretti[2], Emanuele Raineri[3], Giacomo Marmorini[4], William Burgos[1], Joan Jene[1] and Gonzalo Vera[1]. [1]Centre for Research in Agricultural Genomics (CRAG). Consortium CSIC-IRTA-UAB-UB. Campus UAB. Bellaterra, Barcelona, Catalunya, Spain. [2]Big Data Institute (BDI), University of Oxford, UK. [3]Centro Nacional de Análisis Genómico (CNAG-CRG), Parc Científic de Barcelona, BCN, Spain.[4]Yukawa Institute for Theoretical Physics (YITP), Kyoto, Japan.[5]Dept. Genètica, Microbiologia i Estadística & Institut de Recerca de la Biodiversitat (IRBio), Universitat de Barcelona, Barcelona, Spain.

DIGUP Developer: Mireia Vidal. Centre for Research in Agricultural Genomics (CRAG). Consortium CSIC-IRTA-UAB-UB. Campus UAB. Bellaterra, Barcelona, Catalunya, Spain.

asta_convrtr: Developer: Sebastian E. Ramos-Onsins. Centre for Research in Agricultural Genomics (CRAG). Consortium CSIC-IRTA-UAB-UB. Campus UAB. Bellaterra, Barcelona, Catalunya, Spain.

weight4tfasta: Developer: Sebastian E. Ramos-Onsins. Centre for Research in Agricultural Genomics (CRAG). Consortium CSIC-IRTA-UAB-UB. Campus UAB. Bellaterra, Barcelona, Catalunya, Spain.

gvCF2tFasta: Developer: Jordi Leno-Colardo. Centre for Research in Agricultural Genomics (CRAG). Consortium CSIC-IRTA-UAB-UB. Campus UAB. Bellaterra, Barcelona, Catalunya, Spain.

indexingtFasta: Developer: Jordi Leno-Colardo. Centre for Research in Agricultural Genomics (CRAG). Consortium CSIC-IRTA-UAB-UB. Campus UAB. Bellaterra, Barcelona, Catalunya, Spain.

version 0.1

4.3. SPiGA_fasta2tfasta.ui

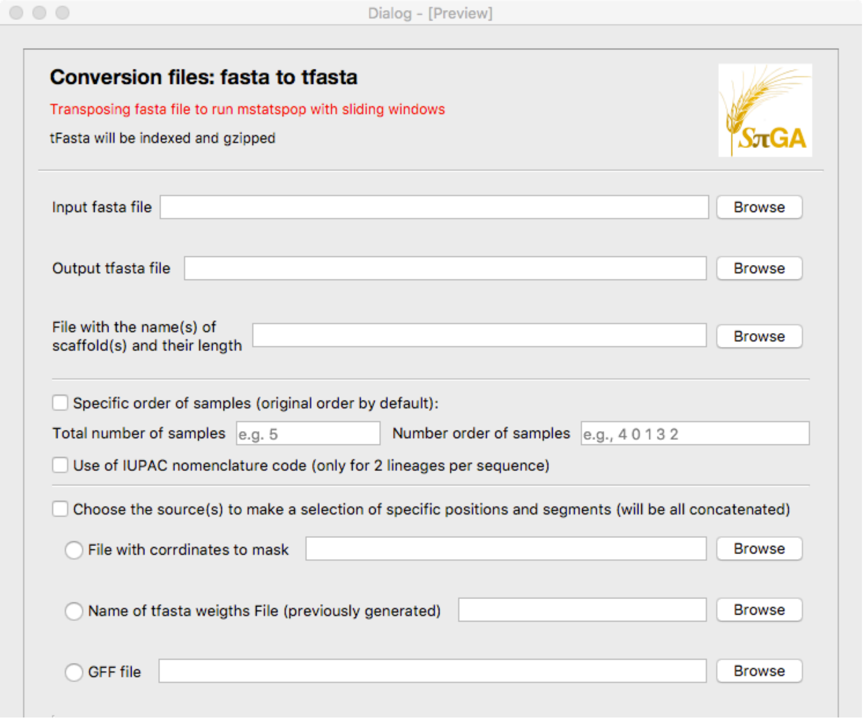
The command that the window will return is *fastaconvtr*. The flags are the following:

```
#fastaconvtr v0.1beta (20200613) Sebastian E. Ramos-Onsins.
Flags:
      -F [input format file: f (fasta), t (tfasta)]
      -i [path and name of the input file (text or gz indexed)]
      -f [output format file: t (tfasta), f (fasta), m (ms), 0(nothing)]
      -o [path and name of the output sequence file]
      -n [name of the file containing the name(s) of scaffold(s) and their length (separated by a tab), one per line
(ex. fai file)]
   OPTIONAL PARAMETERS:
      -h [help and exit]
      -P [define window lengths in 'physical' positions (1) or in 'effective' positions (0)]. DEFAULT: 1
      -O [#_nsam] [Reorder samples: number order of first sample, number 0 is the first sample] [second
sample] ...etc.
```

```
    -W [for ms and fasta outputs, file with the coordinates of each window: (one header plus nlines with init
end]
    -N [#_pops] [#samples_pop1] ... [#samples_popN] (necessary in case to indicate the outgroup population)
    -G [outgroup included (1) or not (0), last population (1/0)]. DEFAULT: 0
    -u [Missing counted (1) or not (0) in weights given GFF annotation]. DEFAULT: 0
    -m [masking regions: file indicating the start and the end of regions to be masked by Ns]
    -E [input file with weights for masking positions: include three columns with a header, first the physical
positions (1...end), second the weight for positions and third a boolean weight for the variant (eg. syn variant
but nsyn position)]
   Outputing ms format:
    -k [path and name of the output mask file for ms].
    -w [window size]. DEFAULT: Total_length
    -s [slide size]. DEFAULT: Total_length
   Inputing fasta format:
    -p [if fasta input, haplotype:1 (single sequence) genotype:2 (two mixed sequences in IUPAC). DEFAULT:
1
   Annotation file and weight options:
    -g [GFF_file]
       [add also: coding,noncoding,synonymous,nonsynonymous,silent, others (whatever annotated)]
       [if 'synonymous', 'nonsynonymous', 'silent' add: Genetic_Code:
Nuclear_Universal,mtDNA_Drosophila,mtDNA_Mammals,Other]
       [if 'Other', introduce the single letter code for the 64 triplets in the order UUU UUC UUA UUG ... etc.]
    -c [in case use coding regions, criteria to consider transcripts (max/min/first/long)]. DEFAULT: long
    -e [path and name of the output weights file (mandatory if included GFF)]
```

Here we use only the options related to fasta as input and tfasta as output:

The window sees like this:



Conversion files: fasta to tfasta

Transposing fasta file to run mstatspop with sliding windows

tFasta will be indexed and gzipped

Input fasta file [Browse]

Output tfasta file [Browse]

File with the name(s) of scaffold(s) and their length [Browse]

□ Specific order of samples (original order by default):

Total number of samples e.g. 5 Number order of samples e.g., 4 0 1 3 2

□ Use of IUPAC nomenclature code (only for 2 lineages per sequence)

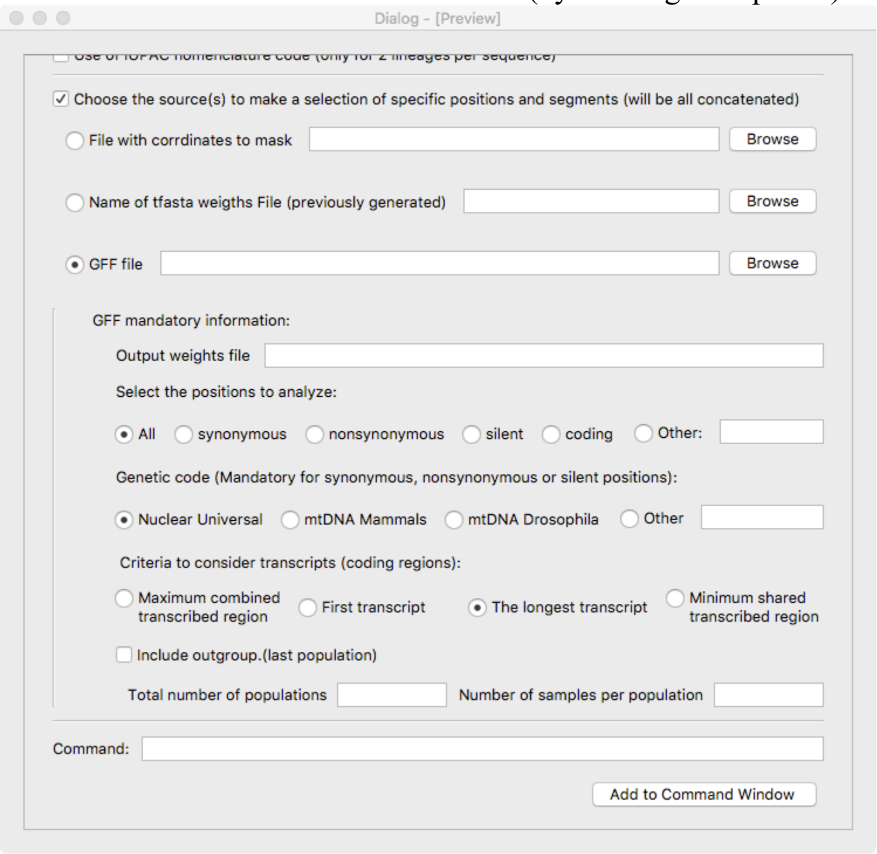
□ Choose the source(s) to make a selection of specific positions and segments (will be all concatenated)

○ File with corrdinates to mask [Browse]

○ Name of tfasta weigths File (previously generated) [Browse]

○ GFF file [Browse]

If you scroll to down, there are the GFF options, plus a visualization of the final command that the user has selected (by clicking the options):



Dialog - [Preview]

□ Use of IUPAC nomenclature code (only for 2 lineages per sequence)

☑ Choose the source(s) to make a selection of specific positions and segments (will be all concatenated)

○ File with corrdinates to mask [Browse]

○ Name of tfasta weigths File (previously generated) [Browse]

● GFF file [Browse]

GFF mandatory information:

Output weights file

Select the positions to analyze:

● All ○ synonymous ○ nonsynonymous ○ silent ○ coding ○ Other:

Genetic code (Mandatory for synonymous, nonsynonymous or silent positions):

● Nuclear Universal ○ mtDNA Mammals ○ mtDNA Drosophila ○ Other

Criteria to consider transcripts (coding regions):

○ Maximum combined transcribed region ○ First transcript ● The longest transcript ○ Minimum shared transcribed region

□ Include outgroup.(last population)

Total number of populations   Number of samples per population

Command:

[Add to Command Window]

The actions to do are the following: The first 4 are mandatory (if not defined, the "Add to Command Window" should not work).

Mandatory flags:

1. The initial flags for this window are: *fastaconvtr -F fasta -f tfasta*
2. Input fasta file: add the flag *-i [path.filename].* Add action to the button for browsing a file (usually extension .fa or .fasta, but not necessarily).
3. Output tfasta file: add the flag *-o [path.filename].* Also add the action to button for browsing.
4. File with the name(s) of scaffold(s) and their length: add the flag *-n [path.filename].* Also add the action to button for browsing.

Optional flags:

5. About optional flags, Specific order of samples: if checked, both QLineEdits should be filled. The number of values in the "number order of samples" (exemple, sample number 4, then sample number 0, etc..) must coincide with the "Total number of samples". Add flag *-O [Total_number_samples] [sample1] [sample2] …[sampleN]*.
6. Use of IUPAC nomenclature. Add flag *-p 1* if unchecked. If checked add flag *-p 2*.
7. If you choose to select specific regions (checking the check box), then, the following options should be visually available (before are visible but attenuated). There are three radio buttons that should be exclusive:

8. File with coordinates (within scaffolds) to mask. If checked, add flag -m [path.filename]. Also add the action to button for browsing.
9. File of weights (from a previous filtering). If checked, add flag -E [path.filename]. Also add the action to button for browsing.
10. GFF file. If this option is checked, the below options referring to GFF will be visually available. The user must add the name of the GFF file.
11. The name of the output weighting files. Add the flag -e [path.filename]. Also add the action to button for browsing.
12. Select one of the exclusive options (radio buttons, one must be checked). The possible values are "All" (default), "synonymous", "nonsynonymous", "silent", "coding" or other (whatever single word the user writes). If the option is not "synonymous", "nonsynonymous","silent" neither "All" add the flag -g [path.filename] [SelPositions]. If the the option is "All", add flag -g [path.filename].
13. In case the previous option is "synonymous", "nonsynonymous" or "silent", select one of the exclusive options (otherwise is not available). The possible values for Genetic Code are: "Nuclear_Universal" (default), "mtDNA_Drosophila", "mtDNA_Mammals", "Other". In case the user select other, 64 letters separated by spaces must be given. Add flag -g [path.filename] [SelPositions] [GeneticCode].
14. In the same way, if the previous option is available, this one will be also available. Select the criteria to consider transcripts. It is also an exclusive choice, like before. The options are: "max", "first", "long" (default) and "min", in this order. Add flag -c [max/first/long/min].
15. The weights can be calculated considering the presence of an outgroup species. In case considering an outgroup, add flag -G 1, if not add flag -G 0 (default).
16. This option is only mandatory if outgroup is defined (-G 1). Include the number of populations and the number of samples per population. Add flag -N [npops] [pop1] [pop2]…[popN].

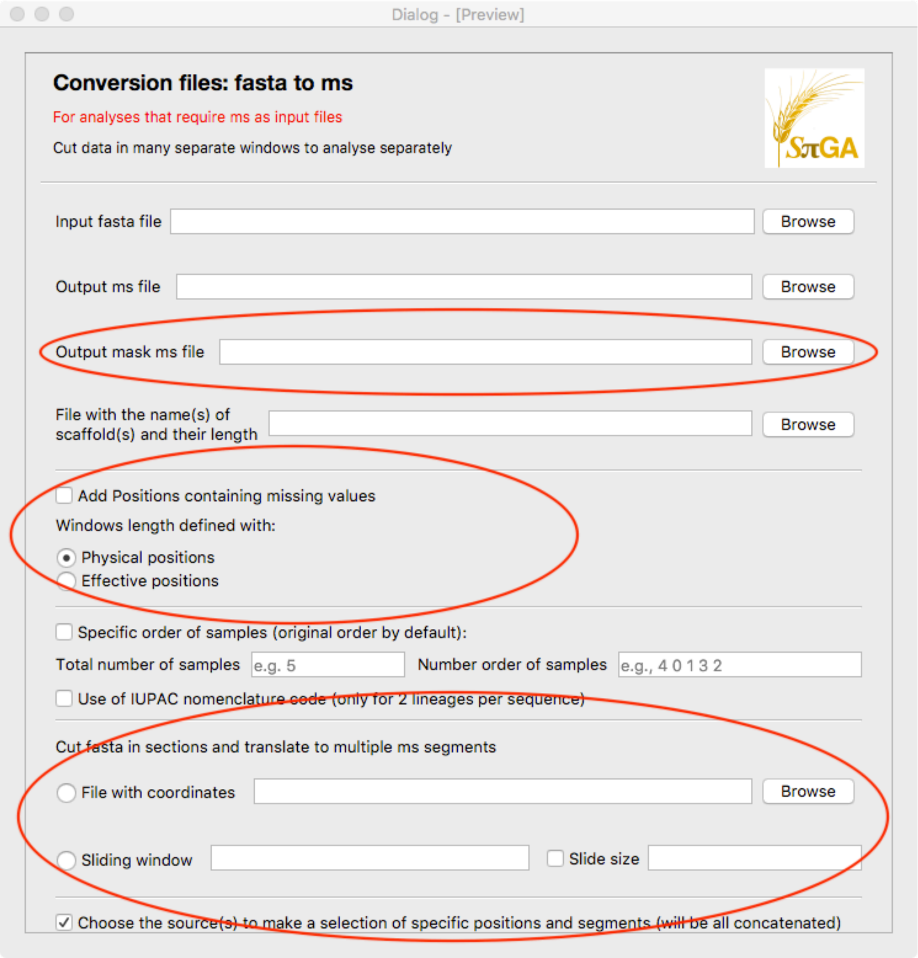
Command visualization

All these options can be seen at real time on the command line section below (the user can not edit). If the user accepts, just click the button "Add to Command Window" to accept this command. The window will close and the command will be visualized in the Command Window (and also registered in the log Window).

4.4. SPiGA_fasta2ms.ui.

This will also return a command from *fastaconvtr*. The window is very similar to previous, but the options are related to fasta as input and ms as output format. The initial Command for this window is: *fastaconvtr -F fasta -f ms*.

The windows contains the same options than before plus few more additional flags. See below.



Dialog - [Preview]

Conversion files: fasta to ms

For analyses that require ms as input files

Cut data in many separate windows to analyse separately

Input fasta file [ ] Browse

Output ms file [ ] Browse

Output mask ms file [ ] Browse

File with the name(s) of scaffold(s) and their length [ ] Browse

☐ Add Positions containing missing values

Windows length defined with:

● Physical positions

○ Effective positions

☐ Specific order of samples (original order by default):

Total number of samples [e.g. 5] Number order of samples [e.g., 4 0 1 3 2]

☐ Use of IUPAC nomenclature code (only for 2 lineages per sequence)

Cut fasta in sections and translate to multiple ms segments

○ File with coordinates [ ] Browse

○ Sliding window [ ] ☐ Slide size [ ]

☑ Choose the source(s) to make a selection of specific positions and segments (will be all concatenated)

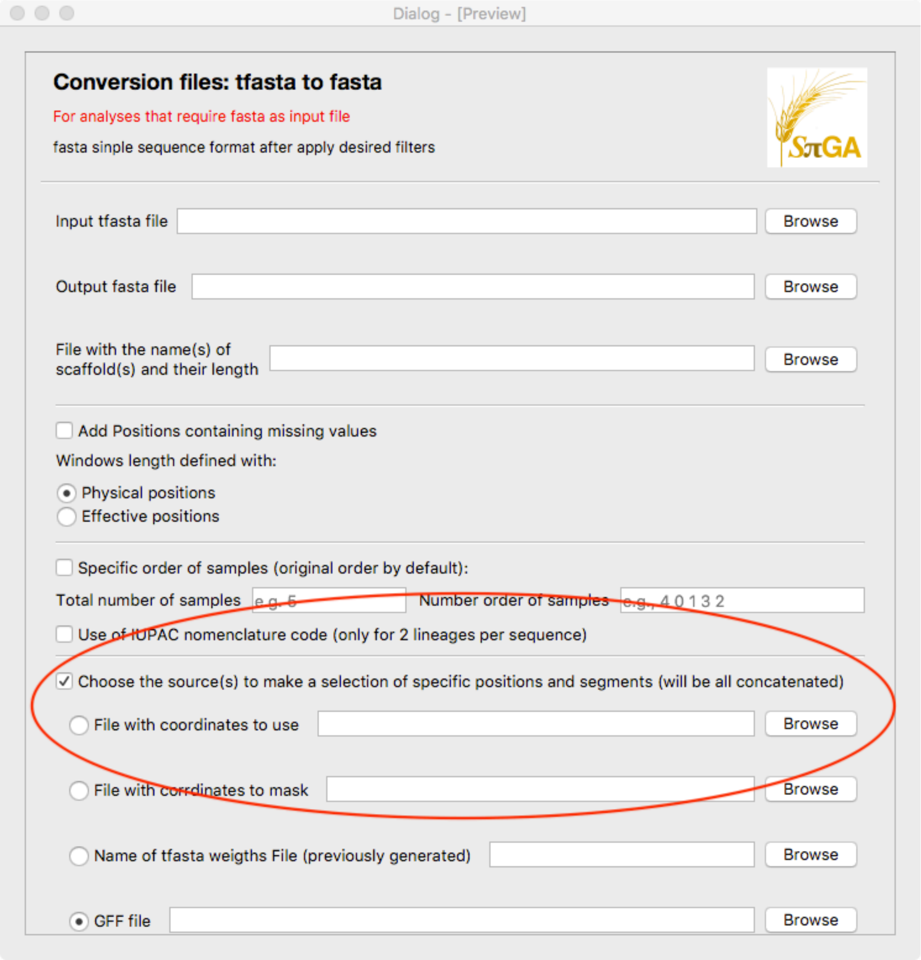
Additional flags (in comparison to SPiGA_fasta2tfasta.ui):

1. It is mandatory to include an Output mask ms file. Add flag *-k [path.filename]*. Also activate the browser.
2. Add positions containing missing values: If unchecked add flag *-u 0*, if checked add flag *-u 1*.
3. Windows length defined by: Physical positions or Effective positions. These are exclusive. If Physical is checked, add flag *-P 1*, if Effective, add flag *-P 0.*
4. Menu "Cut fasta in sections and translate to multiple ms segments": There are two exclusive options, "File with coordinates" and "Sliding windows".
   a. If checked the first one, add flag *-W [path.filename].*
   b. If checked Sliding windows, add flag *-w [number]*. Also, there is the option to add the size of window. Add flag *-s [number]*.

4.5. SPiGA_tFasta2fasta.ui

This will also return a command from *fastaconvtr*. The window is very similar to previous, but the options are related to tfasta as input and fasta as output format. The initial Command for this window is: *fastaconvtr -F tfasta -f fasta*.

The windows contains the same options than the one before with few modifications. See below.



Dialog - [Preview]

Conversion files: tfasta to fasta

For analyses that require fasta as input file

fasta sinple sequence format after apply desired filters

Input tfasta file [ ] Browse

Output fasta file [ ] Browse

File with the name(s) of scaffold(s) and their length [ ] Browse

□ Add Positions containing missing values

Windows length defined with:

● Physical positions
○ Effective positions

□ Specific order of samples (original order by default):

Total number of samples [e.g. 5] Number order of samples [e.g. 4 0 1 3 2]

□ Use of IUPAC nomenclature code (only for 2 lineages per sequence)

☑ Choose the source(s) to make a selection of specific positions and segments (will be all concatenated)

○ File with coordinates to use [ ] Browse

○ File with corrdinates to mask [ ] Browse

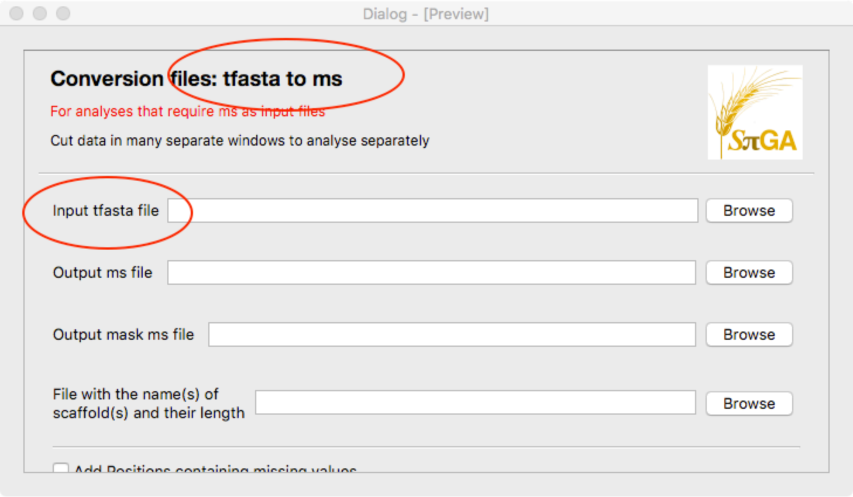
○ Name of tfasta weigths File (previously generated) [ ] Browse

● GFF file [ ] Browse

The option "choose the source(s) to make a selection…" has one more (and alsoexclusive) option in addition to the other three ("File with coordinates to use"). If the user choose this option, add flag *-E [path.filename]*.

4.6. SPiGA_tfasta2ms.ui

This window is exactly equal (same flags and structure) than SPiGA_fasta2ms.ui, except in the header of the page and the title of the input file. Here, the command is *fastaconvtr -F tfasta -f ms*.



Dialog - [Preview]

Conversion files: tfasta to ms

For analyses that require ms as input files

Cut data in many separate windows to analyse separately

Input tfasta file [ ] Browse

Output ms file [ ] Browse

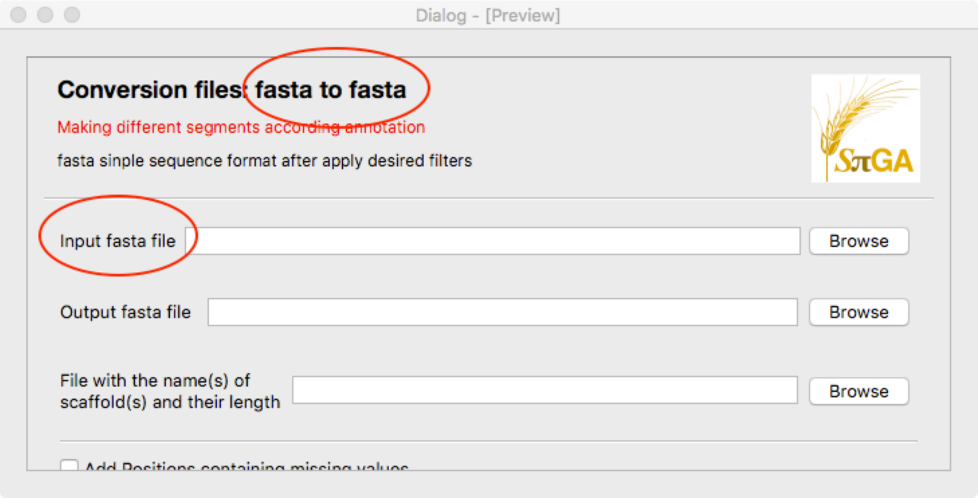
Output mask ms file [ ] Browse

File with the name(s) of scaffold(s) and their length [ ] Browse

□ Add Positions containing missing values

4.7. SPiGA_fasta2fasta.ui

This window is exactly equal (same flags and structure) than SPiGA_tfasta2fasta.ui, except in the header of the page and the title of the input file. Here, the command is *fastaconvtr -F fasta -f fasta*.



Conversion files: fasta to fasta

Making different segments according annotation

fasta sinple sequence format after apply desired filters

Input fasta file

Output fasta file

File with the name(s) of scaffold(s) and their length

□ Add Positions containing missing values

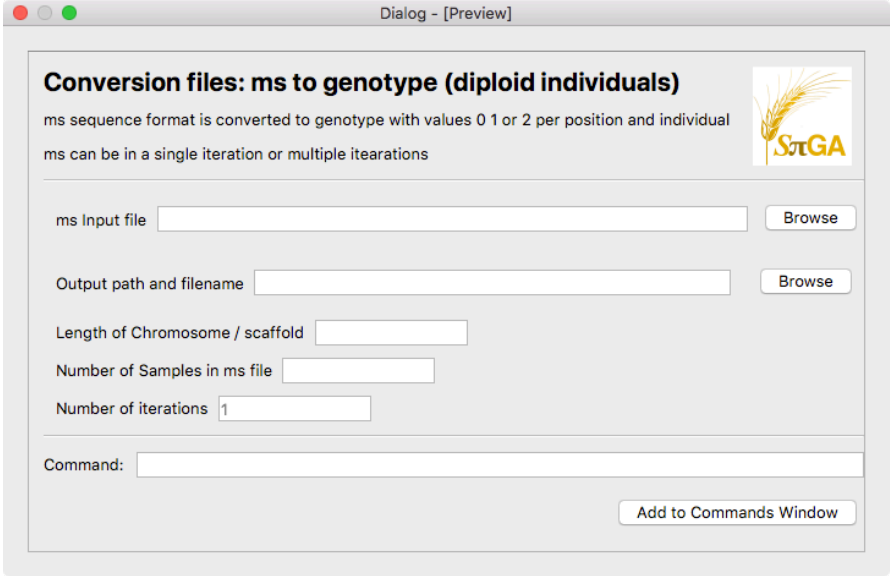
4.8. SPiGA_ms2geno.ui

The command that the window will return is *ms2geno*. The flags are the following:

```
ms2geno: Convert ms to genotype file
version 20200415

Usage:
ms2geno [chrom length] [nsam (2xind)] [iterations] [name_file OR stdin]
Results to stdout
```

The window sees like this:



Conversion files: ms to genotype (diploid individuals)

ms sequence format is converted to genotype with values 0 1 or 2 per position and individual

ms can be in a single iteration or multiple ítearations

ms Input file

Output path and filename

Length of Chromosome / scaffold

Number of Samples in ms file

Number of iterations 1

Command:

Here, the command does not contain flags, but values should be in order and separated by spaces:

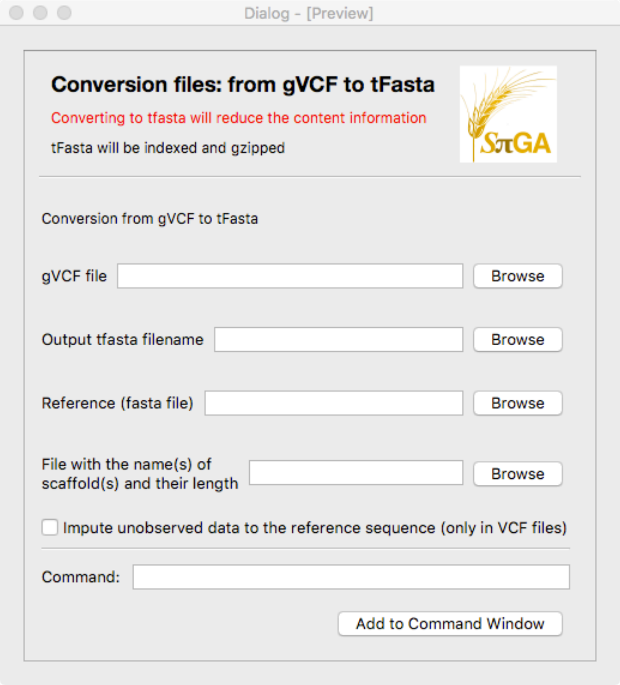
ms2geno [chrom length] [number of samples] [iterations] [path.input.filename] > [path.output.filename].

4.9. SPiGA_gVCF2tfasta.ui

The command that the window will return is *gVCF2tfasta*. The flags are the following:

```
VCF2tFasta
Version 0.2
Usage: ./gVCF2tFasta -v input.vcf(.gz) -r reference.fa(.gz) -o outputname -n chromosomes.txt
Structural Variants are considered as missing data (N)
Options:
    -h                  Help and exit
    -v                  Input VCF file
    -r                  Reference Fasta file
    -o                  Output compressed tFasta filename (without extension)
    -n                  File with chromosome(s) to convert and its length
    -i                  Imputation (Only use with VCF files, not gVCF files):
                                0 if missing data in VCF is equal to N in tFasta
                                1 if missing data in VCF is equal to reference fasta in tFasta
                                Default value is 0
```

The window sees like this:



Conversion files: from gVCF to tFasta

Converting to tfasta will reduce the content information

tFasta will be indexed and gzipped

SπGA
Conversion from gVCF to tFasta
gVCF file
Browse
Output tfasta filename
Browse
Reference (fasta file)
Browse
File with the name(s) of scaffold(s) and their length
Browse
Impute unobserved data to the reference sequence (only in VCF files)
Command:
Add to Command Window

The flags for *gVCF2tfasta* are the following:

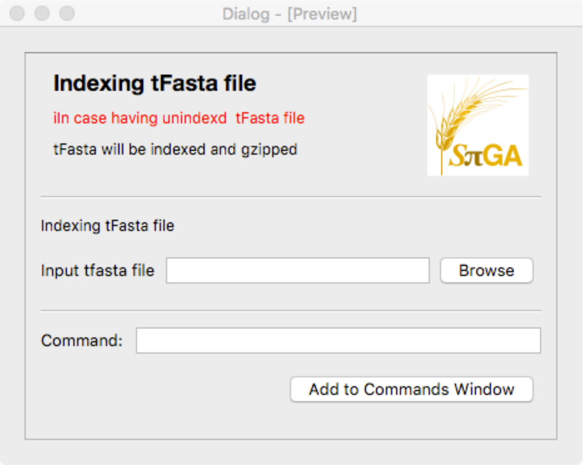
1. gVCF input file: flag *-v [path.filename]*.
2. Output file: flag *-o [path.filename]*.
3. Reference fasta: *-r [path.filename].*
4. File with names of scaffolds: flag *-n [path.filename]*.
5. Imputed unobserved data: If checked add flag *-i 1*, if unchecked add flag *-i 0*.

4.10. SPiGA_Indexingtfasta.ui

The command that the window will return is *indexingtfasta*. The flags are the following:

```
indexingtFasta
Usage: ./indexingtFasta -i file(.gz)
Options:
    -h          Help and exit
    -i          tFasta or Weights file (compressed or uncompressed) to index
```

The window sees like this:



Dialog - [Preview]

Indexing tFasta file

iIn case having unindexd tFasta file

tFasta will be indexed and gzipped

Indexing tFasta file

Input tfasta file

Command:

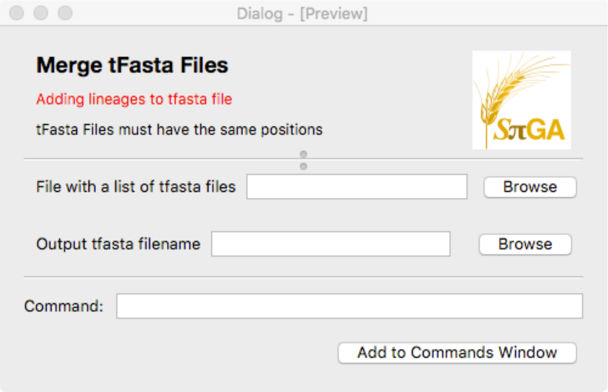
The flag for *indexingtfasta* is only the input file: *-i [path.filename]*.

4.11. SPiGA_Jointfasta.ui

The command that the window will return is *Merge_tFastas*. The flags are the following:

```
Merge_tFastas
Usage: ./merge_tfastas -i list_tFastas.txt -o outputname
Options:
        -h      Help and exit
        -i      List with the name of all tFasta to merge (with the path to each file if they were not in the
same directory of program)
        -o      Output compressed tFasta filename (without extension)
```

The window sees like this:



Dialog - [Preview]

Merge tFasta Files

Adding lineages to tfasta file

tFasta Files must have the same positions

File with a list of tfasta files

Output tfasta filename

Command:

The flags for *Merge_tFastas* are the following:

1. Input file: flag *-i [path.filename]*.
2. Output file: flag *-o [path.filename]*.

4.12. SPiGA_weight4tfasta.ui

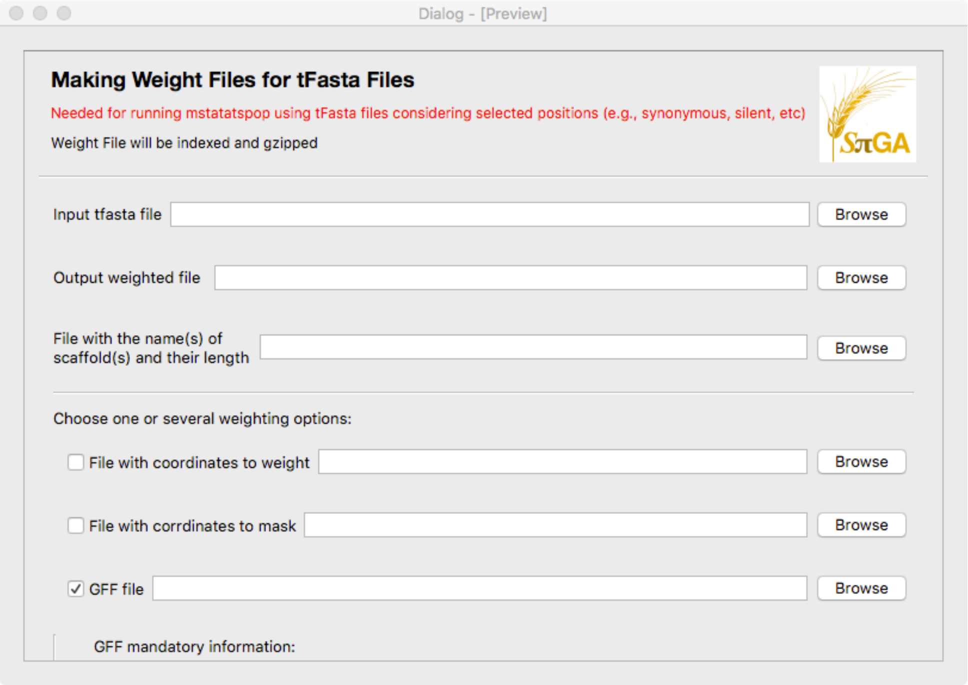
The command that the window will return is *weight4tfasta*. The flags are the following:

```
#weight4tfa v0.1beta (20200614) Sebastian E. Ramos-Onsins.

Flags:
     -i [path and name of the tfa file (gz file indexed)]
     -o [path and name of the output weighted file (will be ending with .gz)]
```

```
    -n [name of the file containing the name(s) of scaffold(s) and their length (separated by a tab), one per line (ex. fai file)]
  OPTIONAL PARAMETERS:
    -g [path of the GFF_file]
      [add also: coding,noncoding,synonymous,nonsynonymous,silent, others (whatever annotated)]
      [if 'synonymous', 'nonsynonymous', 'silent' add: Genetic_Code: Nuclear_Universal,mtDNA_Drosophila,mtDNA_Mammals,Other]
      [if 'Other', introduce the single letter code for the 64 triplets in the order UUU UUC UUA UUG ... etc.]
    -c [in case use coding regions, criteria to consider transcripts (max/min/first/long)]. DEFAULT: long
    -m [masking regions: file indicating the start and the end of regions to be masked by 0 weights]. DEFAULT: NONE
    -C [coordinates of regions: file indicating the start and the end of regions to be weighted (rest would be weighted as 0 if the file is included)]. DEFAULT: NONE
    -G [number of samples in the outgroup (if exist. Only allowed the last samples in the list)]. DEFAULT: 0
    -h [help and exit]
```

The window sees like this:



Making Weight Files for tFasta Files

Needed for running mstatatspop using tFasta files considering selected positions (e.g., synonymous, silent, etc)

Weight File will be indexed and gzipped

Input tfasta file

Output weighted file

File with the name(s) of scaffold(s) and their length

Choose one or several weighting options:

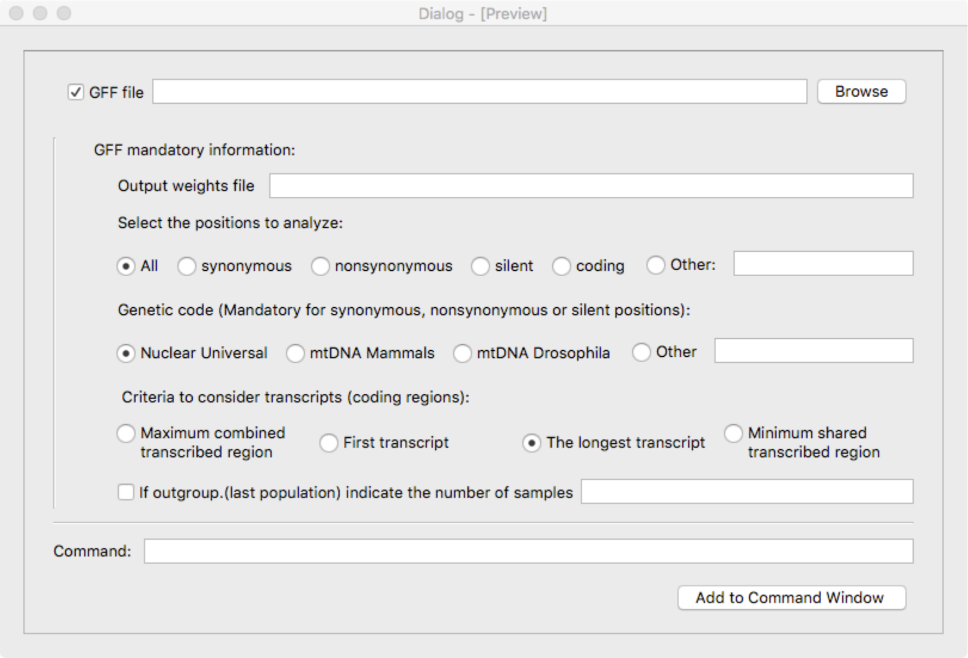
□ File with coordinates to weight

□ File with corrdinates to mask

☑ GFF file

GFF mandatory information:

If you scroll to down, there are the GFF options, plus a visualization of the final command that the user has selected (by clicking the options):



☑ GFF file

GFF mandatory information:

Output weights file

Select the positions to analyze:

● All ○ synonymous ○ nonsynonymous ○ silent ○ coding ○ Other:

Genetic code (Mandatory for synonymous, nonsynonymous or silent positions):

● Nuclear Universal ○ mtDNA Mammals ○ mtDNA Drosophila ○ Other

Criteria to consider transcripts (coding regions):

○ Maximum combined transcribed region ○ First transcript ● The longest transcript ○ Minimum shared transcribed region

□ If outgroup.(last population) indicate the number of samples

Command:

The actions to do are the following: The first 3 are mandatory (if not defined, the "Add to Command Window" should not work).

Mandatory flags:

1. Input tfasta file: add the flag *-i [path.filename].* Add action to the button for browsing a file (usually extension .fa or .fasta, but not necessarily).
2. Output weights file: add the flag *-o [path.filename].* Also add the action to button for browsing.
3. File with the name(s) of scaffold(s) and their length: add the flag *-n [path.filename].* Also add the action to button for browsing.

Optional flags:

4. Choose one or several weighting options:

   a. "File with coordinates to weight" . Add flag *-C[path.filename]*.

   b. "File with coordinates to mask", Add flag *-m[path.filename]*.

   c. "GFF File". Add flag *-g [path.filename]*. This flag has additional values, as shown previously . That is:

      i. Select one of the exclusive options (radio buttons, one must be checked). The possible values are "All" (default), "synonymous", "nonsynonymous", "silent", "coding" or other (whatever single word the user writes). If the option is not "synonymous", "nonsynonymous","silent" neither "All" add the flag -g [path.filename] [SelPositions]. If the the option is "All", add flag -g [path.filename].

      ii. In case the previous option is "synonymous", "nonsynonymous" or "silent", select one of the exclusive options (otherwise is not available). The possible values for Genetic Code are: "Nuclear_Universal" (default), "mtDNA_Drosophila", "mtDNA_Mammals", "Other". In case the user select other, 64 letters separated by spaces must be given. Add flag -g [path.filename] [SelPositions] [GeneticCode].

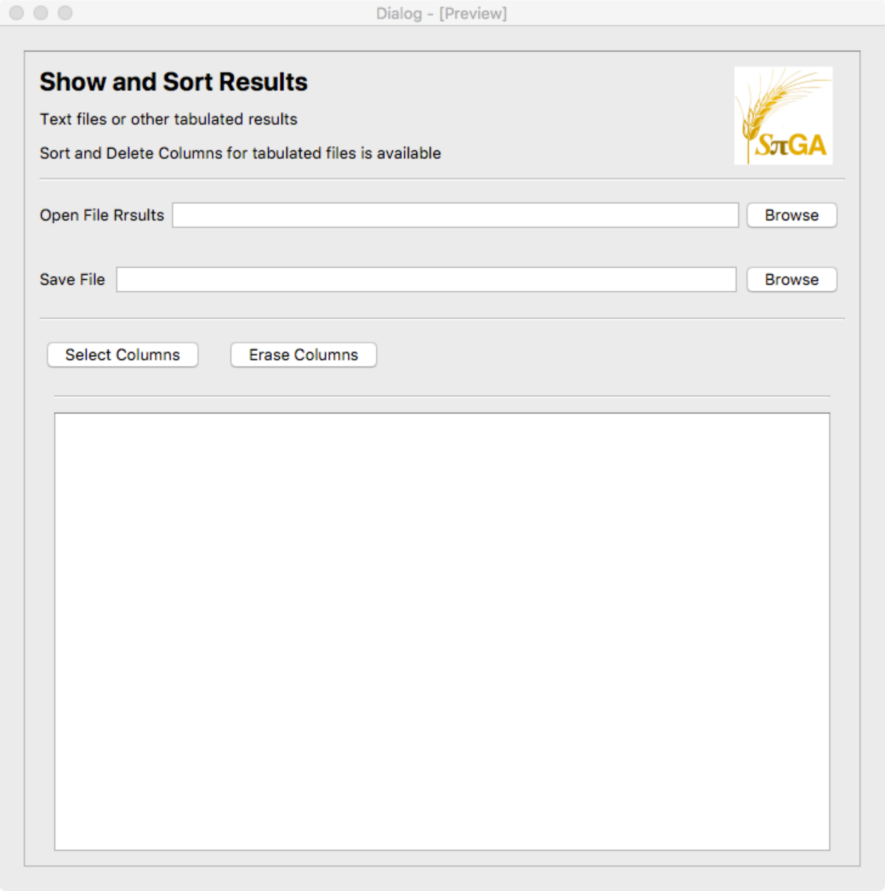
   d. In the same way, if the previous option is available, this one will be also available. Select the criteria to consider transcripts. It is also an exclusive choice, like before. The options are: "max", "first", "long" (default) and "min", in this order. Add flag -c [max/first/long/min].

   e. If outgroup, add the number of samples. Add flag *-G [number]*.

4.13. SPiGA_ResultsTable.ui

This window allows to see the text file results.

The window sees like this:



Dialog - [Preview]

Show and Sort Results

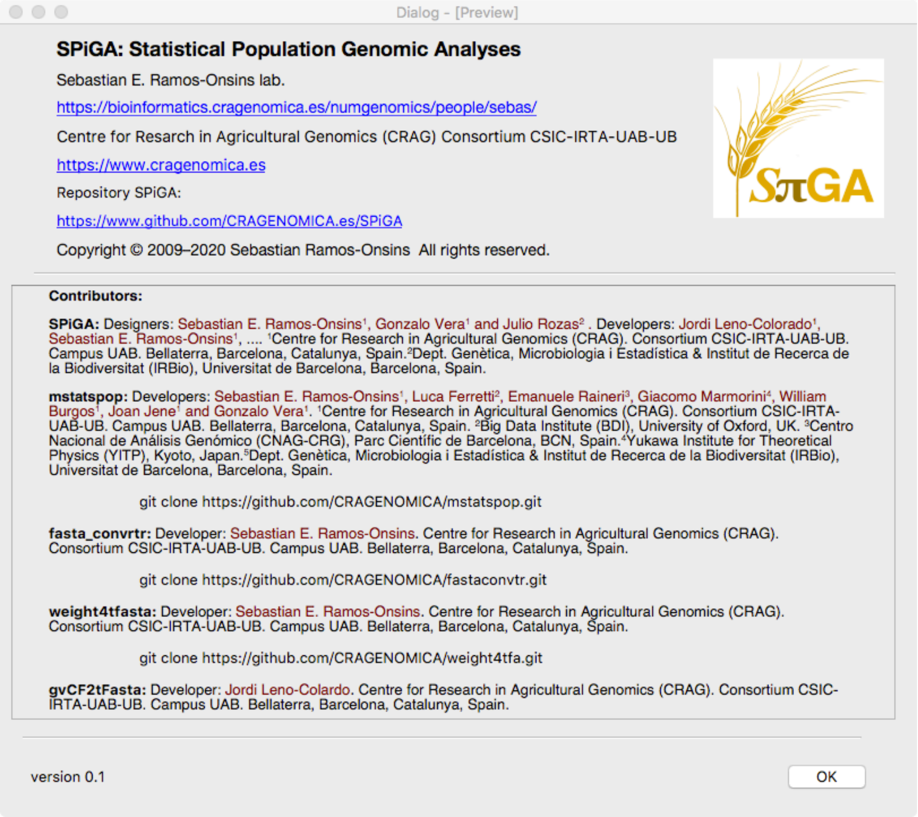
Text files or other tabulated results

Sort and Delete Columns for tabulated files is available

In this case, there is no associated program.
The user should be able to open a text file and be able to save it again (after modifications). Possible modifications (for tabulated files) are the number of columns, select and erase columns.

4.14. SPiGA_Contributors.ui

Informative window. Close the Window when press OK.



SPiGA: Statistical Population Genomic Analyses

Sebastian E. Ramos-Onsins lab.

https://bioinformatics.cragenomica.es/numgenomics/people/sebas/

Centre for Resarch in Agricultural Genomics (CRAG) Consortium CSIC-IRTA-UAB-UB

https://www.cragenomica.es

Repository SPiGA:

https://www.github.com/CRAGENOMICA.es/SPiGA

Copyright © 2009–2020 Sebastian Ramos-Onsins All rights reserved.

Contributors:

SPiGA: Designers: Sebastian E. Ramos-Onsins[1], Gonzalo Vera[1] and Julio Rozas[2] . Developers: Jordi Leno-Colorado[1], Sebastian E. Ramos-Onsins[1], ... [1]Centre for Research in Agricultural Genomics (CRAG). Consortium CSIC-IRTA-UAB-UB. Campus UAB. Bellaterra, Barcelona, Catalunya, Spain.[2]Dept. Genètica, Microbiologia i Estadística & Institut de Recerca de la Biodiversitat (IRBio), Universitat de Barcelona, Barcelona, Spain.

mstatspop: Developers: Sebastian E. Ramos-Onsins[1], Luca Ferretti[2], Emanuele Raineri[3], Giacomo Marmorini[4], William Burgos[1], Joan Jene[1] and Gonzalo Vera[1]. [1]Centre for Research in Agricultural Genomics (CRAG). Consortium CSIC-IRTA-UAB-UB. Campus UAB. Bellaterra, Barcelona, Catalunya, Spain. [2]Big Data Institute (BDI), University of Oxford, UK. [3]Centro Nacional de Análisis Genómico (CNAG-CRG), Parc Científic de Barcelona, BCN, Spain.[4]Yukawa Institute for Theoretical Physics (YITP), Kyoto, Japan.[5]Dept. Genètica, Microbiologia i Estadística & Institut de Recerca de la Biodiversitat (IRBio), Universitat de Barcelona, Barcelona, Spain.

```
git clone https://github.com/CRAGENOMICA/mstatspop.git
```

fasta_convrtr: Developer: Sebastian E. Ramos-Onsins. Centre for Research in Agricultural Genomics (CRAG). Consortium CSIC-IRTA-UAB-UB. Campus UAB. Bellaterra, Barcelona, Catalunya, Spain.

```
git clone https://github.com/CRAGENOMICA/fastaconvtr.git
```

weight4tfasta: Developer: Sebastian E. Ramos-Onsins. Centre for Research in Agricultural Genomics (CRAG). Consortium CSIC-IRTA-UAB-UB. Campus UAB. Bellaterra, Barcelona, Catalunya, Spain.

```
git clone https://github.com/CRAGENOMICA/weight4tfa.git
```

gvCF2tFasta: Developer: Jordi Leno-Colardo. Centre for Research in Agricultural Genomics (CRAG). Consortium CSIC-IRTA-UAB-UB. Campus UAB. Bellaterra, Barcelona, Catalunya, Spain.

version 0.1

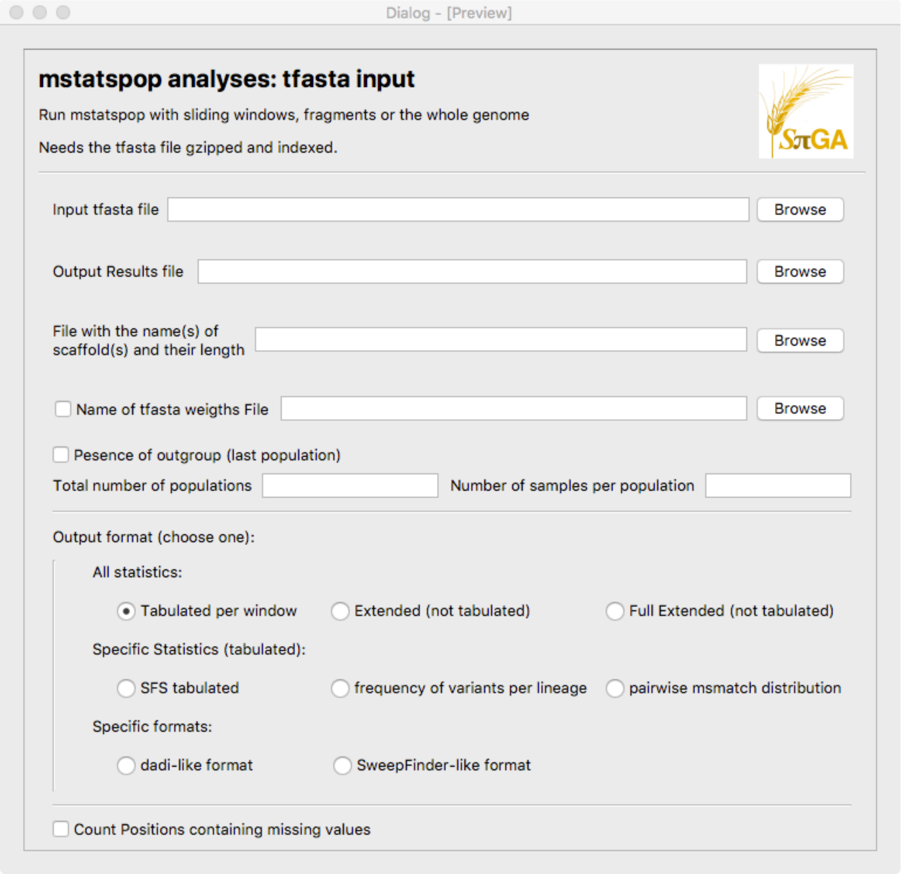
4.15. SPiGA_mstatspop4tfasta.ui

The analysis of data is performed with the code *mstatspop*. This program has numerous flags:

```
mstatspop v.0.1beta (20200331)
Sebastian E. Ramos-Onsins, Luca Ferretti, Emanuele Raineri, Giacomo Marmorini, William Burgos, Joan
Jene and Gonzalo Vera
Variability Analyses of multiple populations: Calculation and estimation of statistics and neutrality tests.
Flags:
   -f [input format file: ms, fasta OR tfa (gz file indexed)]
   -i [path and name of the input file]
   -o [output format file: 0 (extended),
                        1 (single line/window),
                        2 (single line SFS/window),
                        3 (dadi-like format),
                        4 (single line pairwise distribution)
                        5 (single line freq. variant per line/window)
                        6 (SNP genotype matrix)
                        7 (SweepFiinder format -only first pop-)
                       10 (full extended)]
   -N [#_pops] [#samples_pop1] ... [#samples_popN]
   -n [name of the file containing the name(s) of scaffold(s) and their length (separated by a tab), one per line
(ex. fai file)]
   -T [path and name of the output file]. DEFAULT stdout.
  OPTIONAL GENERAL PARAMETERS:
   -G [outgroup (0/1)] (last population). DEFAULT 0.
   -u [include unknown positions (0/1)].  DEFAULT 0.
   -A [Alternative Spectrum File (Only for Optimal Test): alternative_spectrum for each population (except
outg)
     File format: (average absolute values) header plus fr(0,1) fr(0,2) ... fr(0,n-1) theta(0)/nt,
     fr(1,1) fr(1,2) ... fr(1,n-1) theta(1)/nt...]
   -S [Null Spectrum File (only if -A is defined): null_spectrum for each population (except outg).
     (average absolute values) header plus fr(0,1) fr(0,2) ... fr(0,n-1) theta(0)/nt,
     fr(1,1) fr(1,2) ... fr(1,n-1) theta(1)/nt...]. DEFAULT SNM.
  Optional Parameters for fasta and tfa input files:
   -O [#_nsam] [number order of first sample, number 0 is the first sample] [second sample] ...etc. up to
nsamples.
        DEFAULT current order.
```

```
    -t [# permutations per window (H0: Fst=0). Only available with option -u 0]. DEFAULT 0.
    -s [seed]. DEFAULT 123456.
  PARAMETERS FOR TFASTA INPUT (-f tfa): 'SLIDING WINDOW ANALYSIS OF EMPIRICAL DATA'
    -w [window size].
     OR
    -W [file with the coordinates of each window [scaffold init end] (instead options -w and -z).
      DEFAULT one whole window.
   Optional:
    -z [slide size (must be a positive value)]. DEFAULT window size.
    -Z [first window size displacement [for comparing overlapped windows])]. DEFAULT 0.
    -Y [define window lengths in 'physical' positions (1) or in 'effective' positions (0)]. DEFAULT 1.
    -E [input file with weights for positions:
      include three columns with a header,
      first the physical positions (1...end),
      second the weight for positions and
      third a boolean weight for the variant (eg. syn variant in nsyn counts is 0.000)].
      DEFAULT all 1.000
  PARAMETERS FOR MS INPUT (-f ms):'SIMULATION ANALYSIS OF A SINGLE REGION'
    -l [length]
   Optional:
    -r [# ms iterations]. DEFAULT 1.
    -m [include mask_filename] DEFAULT -1 (all positions included).
      [mask_file format: 1st row with 'length' weights, next sample rows x lengths: missing 0, sequenced 1)].
      DEFAULT no mask.
    -v [ratio transitions/transversions]. DEFAULT 0.5.
    -F [force analysis to include outgroup (0/1) (0 in ms means ancestral)]. DEFAULT 0.
    -q [frequency of reverted mutation] (only with -F 1). DEFAULT 0.
  PARAMETERS FOR FASTA INPUT (-f fasta): 'WHOLE REGION ANALYSIS'
   Optional:
    -p [Number of lineages per sequence (1/2)]. DEFAULT 1.
    -g [GFF_file]
      [add also: coding,noncoding,synonymous,nonsynonymous,silent, others (or whatever annotated)]
      [if 'synonymous', 'nonsynonymous', 'silent' add: Genetic_Code:
Nuclear_Universal,mtDNA_Drosophila,mtDNA_Mammals,Other]
      [if 'Other', introduce the code for the 64 triplets in the order UUU UUC UUA UUG ... etc.].
      DEFAULT no annotation.
    -c [in case use coding regions, criteria to consider transcripts (max/min/first/long)]. DEFAULT long.
    -K [make a MASK file with the valid positions for this fasta. Useful for running ms simulations (1/0)].
DEFAULT 0.
  HELP:
    -h [help and exit]
```

The window sees like this:



Dialog - [Preview]

mstatspop analyses: tfasta input

Run mstatspop with sliding windows, fragments or the whole genome

Needs the tfasta file gzipped and indexed.

Input tfasta file [ ] Browse

Output Results file [ ] Browse

File with the name(s) of scaffold(s) and their length [ ] Browse

□ Name of tfasta weigths File [ ] Browse

□ Pesence of outgroup (last population)

Total number of populations [ ] Number of samples per population [ ]

Output format (choose one):

All statistics:

● Tabulated per window ○ Extended (not tabulated) ○ Full Extended (not tabulated)

Specific Statistics (tabulated):

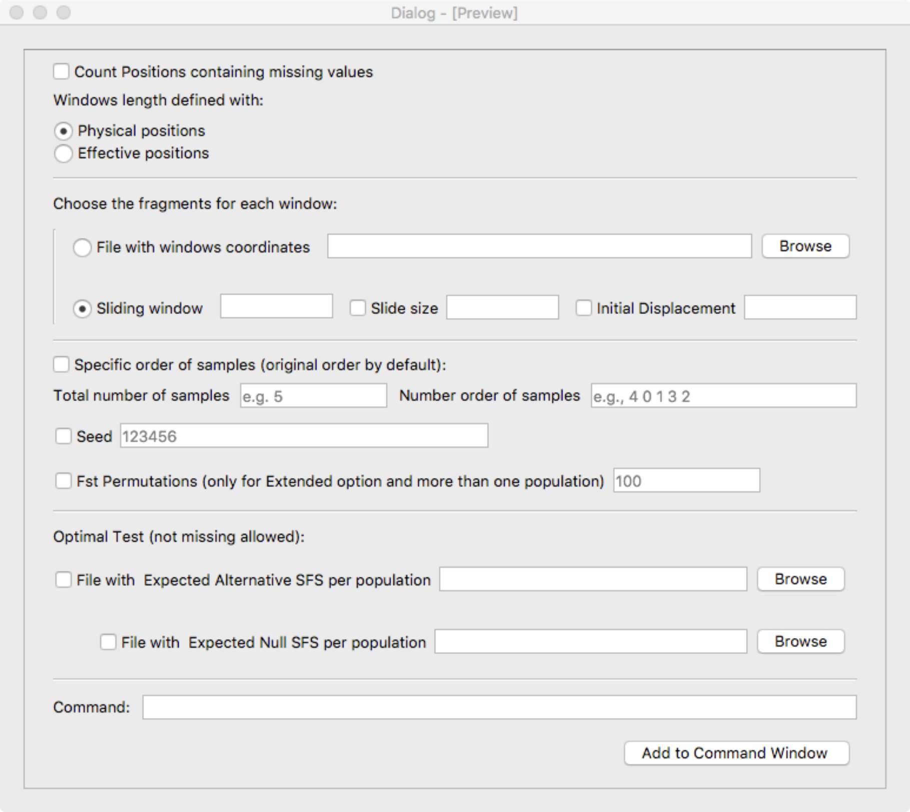
○ SFS tabulated ○ frequency of variants per lineage ○ pairwise msmatch distribution

Specific formats:

○ dadi-like format ○ SweepFinder-like format

□ Count Positions containing missing values

If you scroll to down you see:



Dialog - [Preview]

□ Count Positions containing missing values

Windows length defined with:

● Physical positions

○ Effective positions

Choose the fragments for each window:

○ File with windows coordinates [ ] Browse

● Sliding window [ ] □ Slide size [ ] □ Initial Displacement [ ]

□ Specific order of samples (original order by default):

Total number of samples [e.g. 5] Number order of samples [e.g., 4 0 1 3 2]

□ Seed [123456]

□ Fst Permutations (only for Extended option and more than one population) [100]

Optimal Test (not missing allowed):

□ File with Expected Alternative SFS per population [ ] Browse

□ File with Expected Null SFS per population [ ] Browse

Command: [ ]

Add to Command Window

The command for this window is *mstatspop -f tfa*.

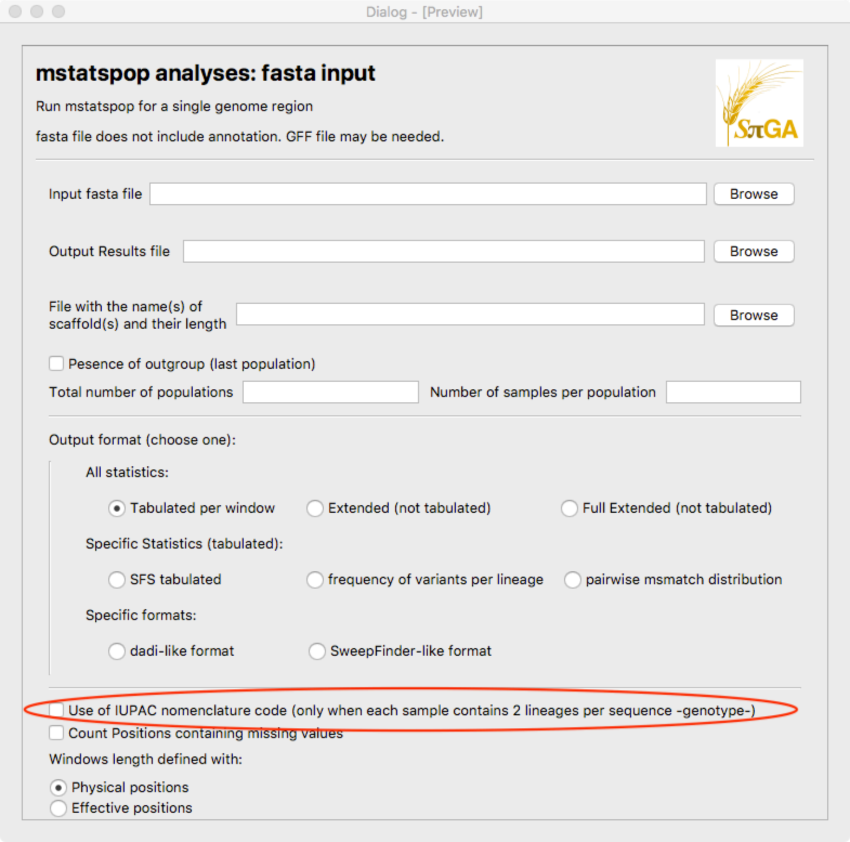
Mandatory and Optional flags:

1. Input tfasta file. Add flag *-i [path.filename]*. Mandatory.
2. Output filename. Add flag *-T [path.filename]*. Mandatory.
3. File with the name(s) of scaffold(s) and their length: add the flag *-n [path.filename].* Also add the action to button for browsing. Mandatory.
4. Input file of weights (Optional). If checked, add flag -E [path.filename]. Also add the action to button for browsing.
5. Outgroup presence. if checked, add flag *-G 1*, If not add flag *-G 0*.
6. Number of populations and number of samples at each population. Add flag *-N [npops] [sizepop1] [sizepop2] … [sizepopN]*. Mandatory.
7. Format output. Mandatory. There are 8 exclusive options: Flag is *-o [number]*.
    a. All statistics: Tabulated (-o 1), Extended (not tabulated) (-o 0), Full Extended (not tabulated) (-o 10).
    b. Specific data analysis: SFS tabulated (-o 2), dadi-like format (-o 3), pairwise distribution (-o 4), frequency variant per lineage (-o 5), SNP Genotype matrix (-o 6), SweepFinder-like format (only first pop) (-o 7).
8. Add positions containing missing values: If unchecked add flag *-u 0*, if checked add flag *-u 1*
9. Windows length defined by: Physical positions or Effective positions. These are exclusive. If Physical is checked, add flag *-Y 1*, if Effective, add flag *-Y 0.*
10. Menu for tfa format (Mandatory): "Choose the fragments for each window": There are two exclusive options, "File with coordinates" and "Sliding windows".
    a. If checked the first one, add flag *-W [path.filename].*
    b. If checked Sliding windows, add flag *-w [number]*. Also, there is the option to add the size of window. Add flag *-z [number]*. Finally, it is possible to make a displacement in the first window. If checked add flag *-Z [number]*.
11. New order of samples (Optional). The number of values in the "number order of samples" (example, sample number 4, then sample number 0, etc..) must coincide with the "Total number of samples". Add flag *-O [Total_number_samples] [sample1] [sample2] …[sampleN]*.
12. Seed number (Optional). Add flag *-s [number]*. Default is number 123456.
13. In case the option Extended (-o 0) or full extended (-o 10) and multiple populations are active (-N >1). The option "Fst Permutations" is active. Add flag *-t [number]*.
14. In case missing values is not selected (-u 0), two more options are available, Optimal test analysis become active: Include a file with the Expected Alternative SFS for each population. Add flag *-A [path.filename]*.
15. In case the previous option is checked, active the option to include a file with the expected SFS of each population for the Expected Null model. Add flag *-S [path.filename]*.

4.16. SPiGA_mstatspop4fasta.ui

This window uses the command *mstatspop -f fa*.

The window sees like this:



mstatspop analyses: fasta input

Run mstatspop for a single genome region

fasta file does not include annotation. GFF file may be needed.

Input fasta file

Output Results file

File with the name(s) of scaffold(s) and their length

□ Pesence of outgroup (last population)

Total number of populations | Number of samples per population

Output format (choose one):

All statistics:

● Tabulated per window ○ Extended (not tabulated) ○ Full Extended (not tabulated)

Specific Statistics (tabulated):

○ SFS tabulated ○ frequency of variants per lineage ○ pairwise msmatch distribution

Specific formats:

○ dadi-like format ○ SweepFinder-like format

□ Use of IUPAC nomenclature code (only when each sample contains 2 lineages per sequence -genotype-)

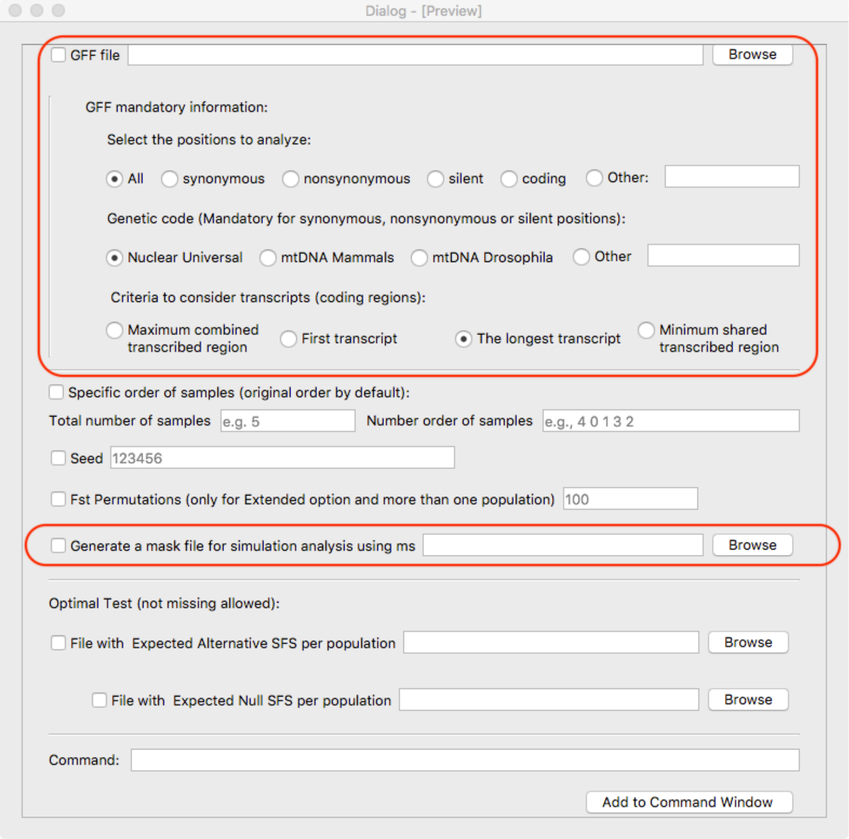
□ Count Positions containing missing values

Windows length defined with:

● Physical positions

○ Effective positions

If you scroll to down you see:



□ GFF file

GFF mandatory information:

Select the positions to analyze:

● All ○ synonymous ○ nonsynonymous ○ silent ○ coding ○ Other:

Genetic code (Mandatory for synonymous, nonsynonymous or silent positions):

● Nuclear Universal ○ mtDNA Mammals ○ mtDNA Drosophila ○ Other

Criteria to consider transcripts (coding regions):

○ Maximum combined transcribed region ○ First transcript ● The longest transcript ○ Minimum shared transcribed region

□ Specific order of samples (original order by default):

Total number of samples e.g. 5 | Number order of samples e.g., 4 0 1 3 2

□ Seed 123456

□ Fst Permutations (only for Extended option and more than one population) 100

□ Generate a mask file for simulation analysis using ms

Optimal Test (not missing allowed):

□ File with Expected Alternative SFS per population

□ File with Expected Null SFS per population

Command:

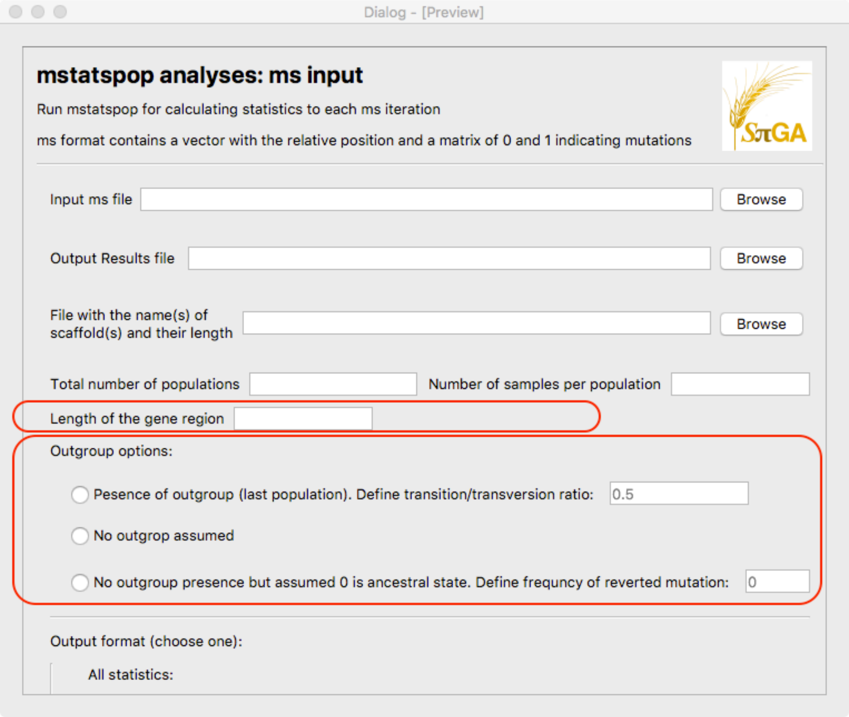
The flags are very similar to SPiGA_mstatspop4tfasta.ui but with some additional flags (see below) and other deleted flags ("Input file of weights", "Choose the fragments for each window"). The additional flags are the following:

Specific flags for fa format are:

1. Use of IUPAC nomenclature. Add flag *-p 1* if unchecked. If checked add flag *-p 2*.
2. Generate a mask file (useful for analyzing ms simulations with same missing data). If checked, add flag -K [path.filename]. Also add the action to button for browsing.
3. GFF file. If this option is checked, the below options referring to GFF will be visually available. The user must add the name of the GFF file.
    a. Select one of the exclusive options (radio buttons, one must be checked). The possible values are "All" (default), "synonymous", "nonsynonymous", "silent", "coding" or other (whatever single word the user writes). If the option is not "synonymous", "nonsynonymous","silent" neither "All" add the flag -g [path.filename] [SelPositions]. If the the option is "All", add flag -g [path.filename].
    b. In case the previous option is "synonymous", "nonsynonymous" or "silent", select one of the exclusive options (otherwise is not available). The possible values for Genetic Code are: "Nuclear_Universal" (default), "mtDNA_Drosophila", "mtDNA_Mammals", "Other". In case the user select other, 64 letters separated by spaces must be given. Add flag -g [path.filename] [SelPositions] [GeneticCode].
    c. In the same way, if the previous option is available, this one will be also available. Select the criteria to consider transcripts. It is also an exclusive choice, like before. The options are: "max", "first", "long" (default) and "min", in this order. Add flag -c [max/first/long/min].

4.17. SPiGA_mstatspop4ms.ui

This window uses the command *mstatspop -f ms*.
The window sees like this:



mstatspop analyses: ms input

Run mstatspop for calculating statistics to each ms iteration

ms format contains a vector with the relative position and a matrix of 0 and 1 indicating mutations

Input ms file [ ] Browse

Output Results file [ ] Browse

File with the name(s) of scaffold(s) and their length [ ] Browse

Total number of populations [ ] Number of samples per population [ ]

Length of the gene region [ ]

Outgroup options:

○ Pesence of outgroup (last population). Define transition/transversion ratio: 0.5

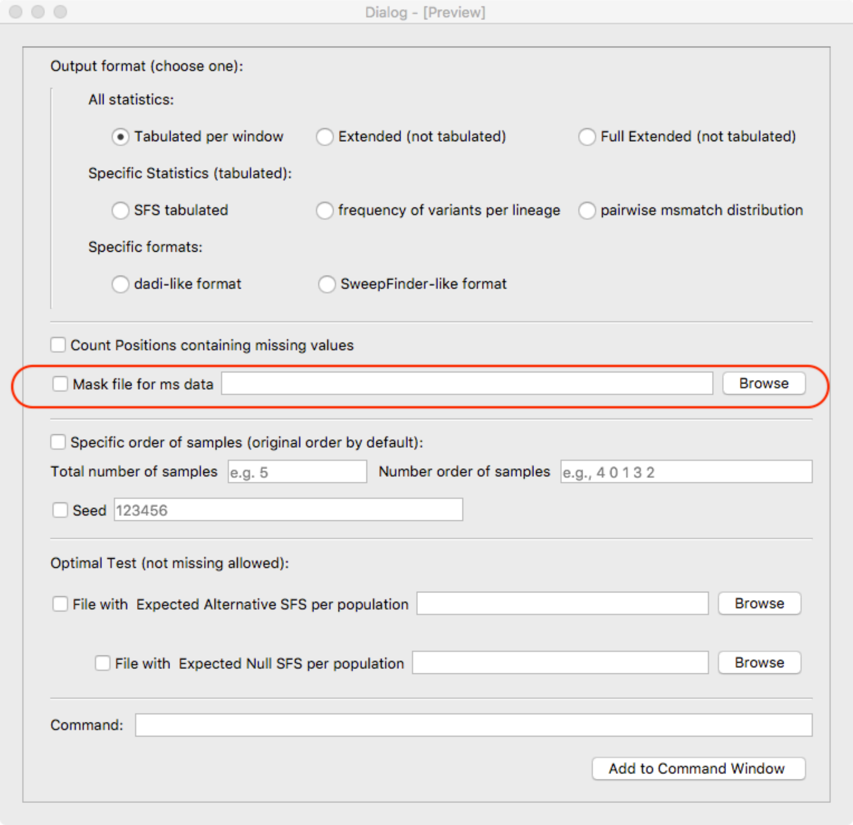
○ No outgrop assumed

○ No outgroup presence but assumed 0 is ancestral state. Define frequncy of reverted mutation: 0

Output format (choose one):

All statistics:

If you scroll to down you see:



Output format (choose one):

All statistics:

● Tabulated per window ○ Extended (not tabulated) ○ Full Extended (not tabulated)

Specific Statistics (tabulated):

○ SFS tabulated ○ frequency of variants per lineage ○ pairwise msmatch distribution

Specific formats:

○ dadi-like format ○ SweepFinder-like format

□ Count Positions containing missing values

□ Mask file for ms data [ ] Browse

□ Specific order of samples (original order by default):

Total number of samples e.g. 5 Number order of samples e.g., 4 0 1 3 2

□ Seed 123456

Optimal Test (not missing allowed):

□ File with Expected Alternative SFS per population [ ] Browse

□ File with Expected Null SFS per population [ ] Browse

Command: [ ]

The flags are very similar to SPiGA_mstatspop4tfasta.ui but with some additional flags and others deleted (“Input file of weights”, “Choose the fragments for each window”, “Fst Permutations”, “Physical/Effective Positions”…).

Specific additional flags for ms format are:

Mandatory flags:

1. Length of the fragment. Add flag *-l [number]*.

Optional flags:

2. Name of file for masking. Add flag *-m [filename]*.
3. Ratio transition/transversion. Add flag *-v [number]*.
4. Assume 0 value to be ancestral. If checked, add flag *-F 1*. If checked, also activate the Frequency of reverted mutation. Add *-q [number]*.

4.18. SPiGA_Tang_Rsb.ui

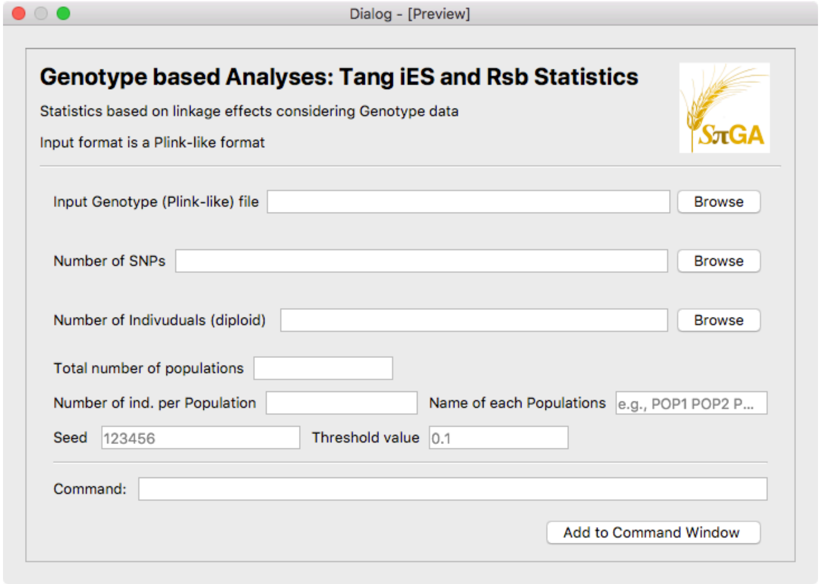
This window uses the command *Tang_stats*.

```
Software for calculating iES and Rsb statistics.
following Tang, Thornton & Stoneking, PloS Biology 2007.
verion 20200615

Usage:
Tang_stats [genotype filename (one chrom)] [number of SNPs] [number of indiv] [threshold value (eg=0.1)] [seed
(eg=123456)] [number pops] [size pop1] [size pop2] ... [size popN] [name_pop1] ... [name_popN]

Output file is automatically generated with the input filename plus '_Results_Tang.txt'
```

The window sees like this:



Dialog - [Preview]

Genotype based Analyses: Tang iES and Rsb Statistics

Statistics based on linkage effects considering Genotype data

Input format is a Plink-like format

Seed 123456

Threshold value 0.1

SπGA
Input Genotype (Plink-like) file
Browse
Number of SNPs
Browse
Number of Indivuduals (diploid)
Browse
Total number of populations
Number of ind. per Population
Name of each Populations
e.g., POP1 POP2 P...
Command:
Add to Command Window

Here, the command does not contain flags, but values should be in order and separated by spaces:

Tang_stats [path.input.filename] [number of SNPs] [number of indiv] [threshold value] [seed] [number pops] [size pop1] [size pop2] ... [size popN] [name_pop1] ... [name_popN]

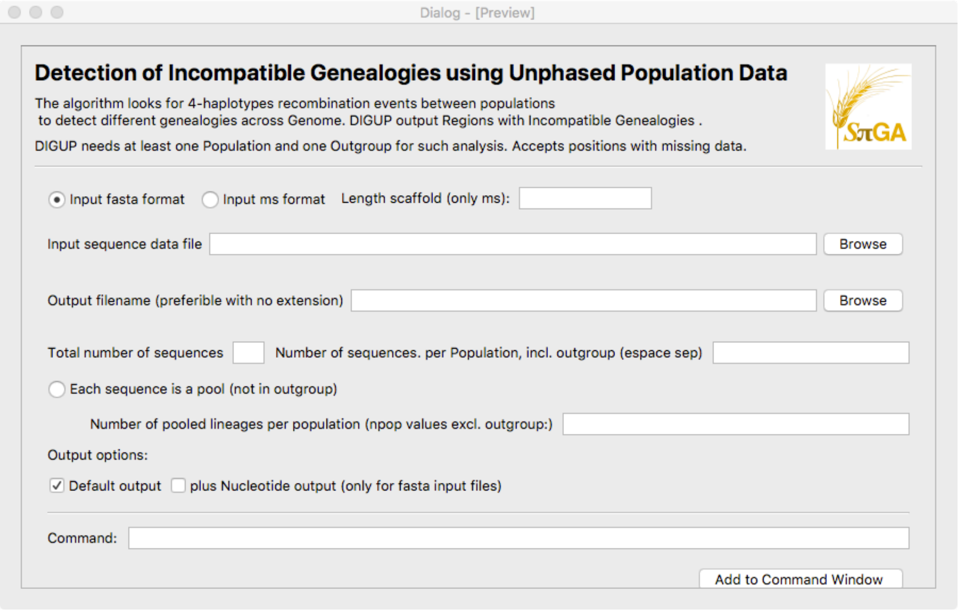
4.19. SPiGA_DIGUP.ui

This window uses the command *DIGUP*.

```
DIGUP code. Mireia Vidal v20200616:
Usage:
```

```
DIGUP [filename (fasta or ms formats)]
    -O   [Output filename (with no extension)]
    -ms (add flag if input is ms format)
    -o   [Type output: 1 (default) 2 (with nucleotides) 12 (both)]
    -n   [number of total sequences]
    -i   [sequences pop1] [sequences pop2] ... [sequences popN]
    -nt  [pooled lines pop1] [pooled lines pop2] ... [pooled lines popN]
    -l   [Sequence length of the ms format]
```

The window sees like this:



Dialog - [Preview]

Detection of Incompatible Genealogies using Unphased Population Data

The algorithm looks for 4-haplotypes recombination events between populations
to detect different genealogies across Genome. DIGUP output Regions with Incompatible Genealogies .

DIGUP needs at least one Population and one Outgroup for such analysis. Accepts positions with missing data.

● Input fasta format ○ Input ms format Length scaffold (only ms):

Input sequence data file

Output filename (preferible with no extension)

Total number of sequences Number of sequences. per Population, incl. outgroup (espace sep)

○ Each sequence is a pool (not in outgroup)

Number of pooled lineages per population (npop values excl. outgroup:)

Output options:

☑ Default output ☐ plus Nucleotide output (only for fasta input files)

Command:

The window contains the following:

1. Mandatory. Two exclusive buttons to select fasta or ms format. If fasta, do not add any flag (default). In case ms add flag *-ms* and activate the length of the scaffold (mandatory for this option). Add flag *-l [number]*.
2. Mandatory: The path and filename. Add *[path.filename]* as the first argument, with no flag letter.
3. Mandatory. Output filename (not extension recommended). Add flag *-O [path.filename]*.
4. Mandatory. Add the total number of sequences (flag *-n [number]*) and the number of sequences per population (*flag -i [numberpop1] [numberpop2]… [numberpopN]*).
5. Optional flag. If pool data activate number of sequences per population. add flag *-nt [numberpool1] [numberpool2]… [numberpoolN]*.
6. Output options. By default is the first option (flag *-o 1*). Extended option is flag *-o 2*. Both can be activated (Add flag *-o12*).

5. IN CASE WE HAVE TIME AND BUDGET: COMPILE GOR LINUX/MACOS ALL ADDITIONAL PROGRAMS

A number of programs are not compiling easily under some architectures. Specifically, all programs seem to compile well in Linux-Redhat and MacOS architectures, but some of them fail to compile in Linux-Debian (those programs made in C++, gVCF2tfasta, indexingtFasta and merge_tfasta).

6. IN CASE WE HAVE TIME AND BUDGET: REVISION AND INCLUSION OF tfaviewer

tfaviewer is a project for visualizing tfasta files (gzipped and indexed), together with their associated annotation file (GFF file).

The code is in Github:

```
git clone https://github.com/CRAGENOMICA/tfaviewer.git
```

The work in this step would be updating this code, in order to work correctly, and open it when is demanded by SPiGA.

7. TESTING

Most windows give as a result a command with flags that can be saved in a file or executed in the included Terminal. **The Commands will be also tested in deep for us or for other users in a next step**, to look for possible flag incompatibilities or bugs produced in the generation of commands.

About external programs, given that this interface only has an interaction with external software through the Terminal Window, only it is necessary to test that Terminal Window is operative: That is, the external programs (mstatspop, fastaconvtr, etc..) should be correctly executed from the terminal. Most programs have a help flag (-h) to test for running.

A number of example files for testing will be used once the interface is working (formats fasta, tfasta and ms, as well as files for coordinates, masks, GFF annotation file and size of scaffolds). A number of command examples with flags will be tested. We should be able to reproduce the same commands with the interface. We will use a list with commands and functions ready for testing once the interface is working.