HOW-TO GUIDE: Automating Fisher-muller diagrams in MATLAB

These MATLAB scripts were created by Katya Kosheleva, 10/12/2012, to automate the process of defining mutational cohorts and drawing Fisher-muller model diagrams for the 12 replicate adapting yeast populations through time. The scripts were modified by Kenneth Flynn, 05/30/2014, to accommodate different data sets and unique experimental designs.

Be starting, this guide assumes you have already downloaded and installed MATLAB and know how to set your current working directory to a specific folder that contains your data files and the following required scripts:

- time_series_import_KMF
- avetrajectories_KMF
- varycolor.m
- get_genotypes_KMF
- tight_subplot.m
- genotype_plots_KMF
- order_clusters_KMF
- ordered_cluster_plots_KMF
- jbfill.m
- muller_plots_KMF

Throughout the guide, commands that should be entered and run in MATLAB will appear as the following:

```
nests = nests(:, any(nests, 1))
```

Lastly, any graphs produced can be printed to an .eps file using the following:

```
print fileName.eps -depsc
```

Import temporal data from an xls excel file

time_series_import_KMF expects the data in a specific format. Columns must contain the following information in this order:

1- Population
2- Population number
3- Trajectory
4- Chromosome
5- Position
6- Class
7- Mutation

8- Gene
9- Amino Acid
10- Class
11- Amino
12- Nearest Downstream Gene
13- Distance
14- X- Value at X number of time points in generations.

Create a spreadsheet for each of your replicate populations following this format. Be sure to record the filename because we will need to direct MATLAB to this file for each replicate population.

For example, if your file is called 'all_nuclear_mutations_B1.xls', define names to be equal to this string by running the following:

```
names = ls('all_nuclear_mutations_B1.xls')
```

Also, be sure to define the following important variables:

sheets - as the name of the sheet the data can be found on. For most, this value will be 'Sheet1' since that is the default.

tNum - the number of time points your experiment utilizes not including 0

timepoints - the time points sampled including 0, typically in generations.

```
sheets = ['Sheet1']  %define Sheet the data can be found on

tNum = X %for example, my data had the following: 0, 102, 150, 264, 400, 450, 540. X = 7.

timepoints = [...] %for example, [0, 102, 150, 264, 400, 450, 540]

xUnits = ['generations'] %for example, generations, days, etc...
```

Start the import:

```
time_series_import_KMF
```

Once the import is complete, the script will clear the names and sheets variable and output **timeseries**. You should be able to open this variable and ensure that it looks correct based on this arrangement:

- 1- Population Number
- 2- Trajectory Number

- 3- Position on W303 Reference
- 4:X- Value at various timepoints

Once the import is complete, the next step is to sort mutations based on allele frequency, cluster them into cohorts and average their allele frequencies through time.

get_genotypes_KMF

You can visualize how well the clustering went with

genotype_plots_KMF

If you are satisfied with the clustering, the next step is to arrange the clusters in the order in which they occur throughout your experiment.

order_clusters_KMF

ordered_cluster_plots_KMF

These graphs provide a more detailed examination into the order of events by providing the likely order in which cohorts appeared and on which backgrounds. The following script I added can be run to export this information into excel for further examination.

gcluster2excel

This information is also used in the last step, drawing the Fisher-muller diagram. First, we must define some useful variables:

- **frequencies** - the frequencies of each *mutation or cluster of mutations* (not haplotypes)
- **nests** - the 'nesting array') describing the genealogical relationships among clones carrying these mutations

Then, we can draw the Fisher-muller diagrams.

frequencies = squeeze(genneststotal(1, any(squeeze(genneststotal(1, :, :)), 2), 2:trajSize))

nests = squeeze(genneststotal(1, any(squeeze(genneststotal(1, :, :)), 2), nameSize:end))

nests = nests(:, any(nests, 1))

muller_plots_KMF(frequencies, nests, timepoints)

To elaborate what these variables do, I will summarize how Katya explained it to me. The format of **nests** is described as follows:

> *'the last non-zero index of each row is the index of the mutation (i.e, just the number of the row). Everything before that describes the genetic background of the current mutation (or cluster of mutations).'*

Consider the following arrays examining the relationship between three mutations:

[1 0 0; 1 2 0; 3 0 0].	[1 0 0; 1 2 0; 1 2 3], etc.

On the left, mutation 2 occurs on the background of 1 while mutation 3 occurred independently on the wild-type background. On the right, the order is sequential; mutation 2 occurred on the background of 1 which accumulate mutation 3.

Example with data from our *Pseudomonas aeruginosa* B2 population:

names = 'all_nuclear_mutations_B2_2014-05-28.xlsx'

sheets = 'Sheet1'

tNum = 6

timepoints = [0 102 150 264 396 450 540]

time_series_import_KMF

get_genotypes_KMF

genotype_plots_KMF

order_clusters_KMF

ordered_cluster_plots_KMF

frequencies = squeeze(gennesttotal(1, any(squeeze(gennesttotal(1, :, :)), 2), 2:trajSize))

```
nests = squeeze(genneststotal(1, any(squeeze(genneststotal(1, :, :)), 2), nameSize:end))

nests = nests(:, any(nests, 1))

muller_plots_KMF(frequencies, nests, timepoints)
```