How to run Arboretum-Hi-C

The commands to run the Arboretum-Hi-C are described in the README.txt file. Here we briefly go over the important input/output files. The usage of the program is:

```
./arboretum -s specorder.txt -e human2mouse.txt -k 10 -t tree.txt -c
config.txt -r none -o outs/ -m learn -b hESC -i uniform -p 0.8 -1 true
-2 true -w true
```

For more details you can use:

```
./arboretum --help
```

Here are the main input parameters:

1- List of species.

"`specorder.txt`" contains the list of cell lines/species, for example `data/arboretum_ins/specorder.txt` contains:

```
hESC
hIMR90
mESC
mCortex
```

2- List of aligned regions.

"`human2mouse.txt`" contains the list of orthologous regions. Each row correspond to one set of aligned regions, for example, these are the first 4 lines of `data/arboretum_ins/human2mouse.txt`:

```
1OG
OG1_1 hESC_chr1_2500000,hIMR90_chr1_2500000,mESC_chr4_154500000,mCortex_chr4_154500000
OG2_1 hESC_chr1_3500000,hIMR90_chr1_3500000,mESC_chr4_153500000,mCortex_chr4_153500000
OG3_1 hESC_chr1_5500000,hIMR90_chr1_5500000,mESC_chr4_152500000,mCortex_chr4_152500000
```

`OG[number]_1` indicates the ID of the orthologous group, and the next item is comma separated list of regions in each species. The region ID should be unique. We use the format `[cell line]_[chromosome]_[region midpoint]`, but you can use any alphanumeric string for region IDs as long as the IDs are unique.

3- Species tree.

"`tree.txt`" describes the relationship between cell lines/species. For example, the file `data/arboretum_ins/tree1.txt` contains (the first line is a comment):

```
#Child     LeftorRight     Parent
hESC  left  Anc3
hIMR90     right Anc3
mESC  left  Anc2
mCortex    right Anc2
```

```
Anc3  left  Anc1
Anc2  right Anc1
```

Which indicates hESC and hIMR90 are leafs of ancestral node Anc3, mESC and mCortex are leafs of ancestral node Anc2, and Anc2 and Anc3 are children of ancestral node Anc1.

4- Eigenvectors.

"`config.txt`" contains the list of input files. For example `data/arboretum_ins/config.txt` contains:

```
hESC  hESC.eigs.txt
hIMR90     hIMR90.eigs.txt
mESC  mESC.eigs.txt
mCortex    mCortex.eigs.txt
```

Where the first column correspond to cell line/species IDs (from `specorder.txt`) and the second column contains the location of input file for each cell line/species. Each input file is tab separated file where first column correspond to region ID (the same as `human2mouse.txt`) and the next *K* columns correspond the *K* eigenvectors for that cell line (See the Section below on "**Computing the eigenvectors**" scripts to obtain the eigen vectors). For example the first 3 lines of `data/arboretum_ins/hESC.eigs.txt` contains:

```
hESC_chr1_2500000     -0.030002  -0.014667  -0.021444  0.002710
     0.022303   0.040580   -0.006751  0.009828   0.031975   -0.012887
hESC_chr1_3500000     -0.031010  -0.014712  -0.023310  0.003993
     0.019556   0.042044   -0.005785  0.008594   0.032568   -0.014167
hESC_chr1_5500000     -0.031585  -0.012418  -0.026815  0.015564
     0.014714   0.043126   -0.019427  0.013534   0.035083   -0.011146
```

5- Outputs.

The output files would be saved in the output directory indicated by "`-o`" option. The main output files would be [cell line/species]_speciesspecnames_clusterassign.txt, for example `mESC_speciesspecnames_clusterassign.txt` would contain a tab separated list of regions and corresponding cluster IDs for mESC cell line:

```
mESC_chr1_108500000   0
mESC_chr1_122500000   0
mESC_chr1_123500000   0
```

Note that the region IDs would not be in the same order as the "`human2mouse.txt`" but you can use `python_scripts/sortCluster.py` script to sort them in the same order (usage was mentioned in README.txt). We suggest running the program multiple time and selecting the clusters corresponding to highest score (saved in `likelihood.txt`).

Defining the orthology

If the datasets are from the same cell liens, we do not need a mapping between regions and the `human2mouse.txt` will contain a list of regions, *e.g.*

```
OG1_1 CellLine1_Region1,CellLine2_Region1,CellLine3_Region1
OG2_1 CellLine1_Region2,CellLine2_Region2,CellLine3_Region2
…
```

Otherwise we need to define an orthology between regions of the different species. We can use `liftOver` software (from UCSC Genome Browser Utilities) to define the mapping. In this work, instead of using `liftOver`, we used blast to align regions from one species to the second species (and vice versa) and defined the reciprocal hits, to create a more conservative map. Please refer to the manuscript for more details.

Computing the eigenvectors

The script `matlab_scripts/getEigs.m` shows an example for computing the eigenvectors. To do this, we first need the normalized contact map corresponding to aligned regions (described in the previous section). The example files in `data/sub_mats/` are 1318x 1318 tab separated matrices corresponding to aligned 1M regions, reordered to match the order of regions in `human2mouse.txt`. From these matrices, the script creates the adjacency matrix with a specific measure of distance. Here we use positive Spearman correlation as a measure of similarity between regions, but any other measure of similarity can be used (*e.g.* log2 of normalized contact counts) to define the adjacency matrix. Next, the Laplacian of the adjacency matrix is calculated and the *K* eigenvectors of the Laplacian are used for clustering (*K* correspond to the number of clusters). We define the Laplacian as $L= D^{-0.5} A D^{-0.5}$ (where A is the adjacency matrix and D is the degree matrix) and used *K* eigenvectors corresponding to top *K* largest eigenvalues. Alternatively, we can define the Laplacian as $L=I- D^{-0.5} A D^{-0.5}$ and use *K* eigenvectors corresponding to smallest eigenvalues (von Luxburg, 2007). The `getEigs.m` script produces arbitrary region IDs, you can change the script to use correct region IDs, or use the script `python_scripts/convertNames.py` to convert the region names (usage described in README.txt).