Sweep deficit/enrichment pipeline

Important note: I developed the pipeline while working with human Ensembl Gene IDs. The pipeline recognizes gene IDs based on the first four letters of human Ensembl Gene IDs. Please just ENSG at the start of your own gene IDs.

If you are interested in collaborating or working independently on improving this pipeline (there is a lot of room for improvement), please contact me at denard@email.arizona.edu for help.

Installation:

To unzip the archive: tar -zxvf exdef_folder.tgz

This creates an exdef_folder that contains the whole pipeline. All the files needed to run the pipeline MUST be placed in exdef_folder without exception.

Then cd to exdef_folder: cd exdef_folder

Then to open the rights to use all the perl scripts: chmod +x ./*.pl

Note: It is up to the user to write their own scripts to prepare their own genes_set_file.txt and valid_file.txt, and to parse the output files in the test_outputs folder. Some scripts in the folder like make_valid_file.pl are my own personal scripts to create valid_file.txt based on my own specific needs, and they should not be used by other users. Again, the users can use their own small parsing scripts based on their own needs to prepare genes_set_file.txt and valid_file.txt. Another file that will be user-specific is distance_file.txt, as well as the gene coordinates file (currently in the folder this file is named ensemble_gene_coords_v69. The name of the coordinates file is one of the arguments to define in the exdef_input_parameters.txt file.

Usage:

Step 1: creation of file with the genes used for the test. The file contains the Ensembl IDs of the genes that are going to be used by the pipeline. One line in the file = one Ensembl gene ID. Use your own code to generate this file. In the provided folder the example file is valid_file.txt.

Step 2: create the file that defines the gene set of interest. Genes of interest (VIPs, disease genes, etc.) in the first column are flagged with “yes” in the second column. The other genes are flagged with “no” in the second column. Only the genes also present in the file of used genes (Step 1) will be used by the pipeline. In the folder the example file is genes_set_file.txt.

Step 3: for every gene, compute a file with the distance of every gene from the closest gene of interest (using gene genomic centers as reference points). If a gene is a gene of interest, then

the distance is zero. This is done to be able to choose control genes far enough from the genes of interest. The example file is distance_file.txt

Step 4: create a table file that contains all the confounding factors that the bootstrap test will control for. First column is Ensembl gene ID, next columns are each factor that will be matched. The example file is Factors_table.txt

Step 5: Create/Edit the input file that contains all the parameters of the pipeline run. The example file is `exdef_input_parameters_file.txt.`

-Sweep_files_prefix: the start of the name of the sweep rank files to test. In the example, five files with the iHS statistic results are used: all_ihsfreqabs_ranks_50kb, all_ihsfreqabs_ranks_100kb, all_ihsfreqabs_ranks_200kb, all_ihsfreqabs_ranks_500kb and all_ihsfreqabs_ranks_1000kb. These files provide the results of the ranking of genes according to the average of iHS (absolute values) in windows of the indicated sizes. In this case the prefix is all_ihsfreqabs_ranks.
The first column in a sweep file is the Ensembl gene IDs (if you are not using Ensembl human gene annotations, add ENSG to the start of your gene IDs). The following columns corresponding to gene ranks, for all the populations included, from the gene with the strongest sweep signal to the one with the lowest signal in each specific population. For example, the all_ihsfreqabs_ranks_"size" files have 27 columns with 26 corresponding to gene sweep ranks according to iHS in the 26 populations from 1000 Genomes phase 3.

-Sweep_sizes: the list of the different window sizes used to rank the sweeps. For the files described just above, **Sweep_sizes** is set to 50kb_100kb_200kb_500kb_1000kb.

-Pipeline_dir: path of the directory where all the pipeline is run. Make sure all the elements needed to run the pipeline are in this directory.

-Valid_genes_file: name of the file generated for step 1.

-Genes_set_file: name of the file with the genes of interest generated in step 2.

-Distance_file: name of the file with the distance from closest gene of interest.

-Factors_table: name of the file with the values for all the confounding factors.

-HGNC_file: file with HGNC names for Ensembl genes to be able to exclude HLA (extreme outliers that can bias results) and histone genes (notorious for abundant gene conversion). Replace this file with your own file if needed.

-Run_bootstrap: yes or no. Provides the option to not run the bootstrap test (no) if it was run already. If yes, the test is run and will overwrite the previous bootstrap results.

-Min_distance: minimal distance from genes of interest to choose control genes.

-Iterations_number: number of control gene sets to create per bootstrap test run. Has to be a multiple of 100.

-Flip: if there are more genes of interest than controls, flips the test to gain power and specificity if set to yes.

-Max_rep: the maximum number a control gene is allowed to be sampled on average across all sample iterations in the bootstrap test. The less the better as including more control genes instead of the same genes over and over again increases both power and specificity. Low Max_rep may lead to the bootstrap_test_script.pl to fail finding enough controls. Increase it if needed.

-Runs_number: total number of times to run the bootstrap test (see step 6 below). Typically, use Iterations_number*Runs_number = 10,000, but not more unless you have very large storage and computing power.

-Simult_runs: maximum number of runs running simultaneously. It depends on the total number of cores you have available.

-Tolerance_range: the range of departure from the average of confounding factors (see step 4) allowed during the matching of control genes to the gene set of interest in the bootstrap test. I typically use a range of 0.05, meaning that the average of confounding factors in the control sets has to be around the average in the gene set of interest, + or – 5%.

-Count_sweeps: yes or no. Instructs the script that quantifies the excess or deficit of sweeps to count either the number of sweeps (yes) or the number of genes in sweeps (no).

-Run_sweeps_count: yes or no. Runs the counting of sweeps or genes (yes) or not (no) if it was already done and does not need to be done again.

-Cluster_distance: maximum distance between genes to count them as overlapping the same selective sweep. A large value may count two neighboring sweeps as one.

-Gene_coords_file: file with gene coordinates, to notably assess the clustering of neighboring genes in the same sweep.

-Threshold_list: list of rank thresholds to use to assess deficit/enrichment. For example, the following list:
`2000,1500,1000,900,800,700,600,500,450,400,350,300,250,200,150,100,80,60,50,40,30,25,20,15,10.` Note that for the sake of speed, the pipeline does not allow thresholds higher than 2,000.

-Populations_list: list of populations as they are ordered in the sweep files. For example, the list of all 1000 Genomes phase 3 populations as they are ordered in the sweep files:

```
ACB,ASW,BEB,CDX,CEU,CLM,CHB,CHS,ESN,FIN,GBR,GIH,GWD,IBS,ITU,JPT,KHV,LWK,MSL,MXL,PEL,PJL,PUR,STU,TSI,YRI
```

It is the user's responsibility to have a sweep file with the corresponding populations as indicated in **Populations_list**.

-Groups_list: a list of groups in which the populations from **Population_list** fall. **Group_list** has the same number of elements as **Population_list**. For each population in **Population_list**, the corresponding element in **Group_list** is the name of the group that the corresponding population belongs to. For example, for the 26 populations of 1000 Genomes phase 3:

```
MIS,MIS,SAS,EAS,EUR,AMR,EAS,EAS,AFR,EUR,EUR,SAS,AFR,EUR,SAS,EAS,EAS,AFR,AFR,AMR,AMR,SAS,AMR,SAS,EUR,AFR
```

-Pop_interest: the population of particular interest to adjust the precision and speed of the script that counts sweeps (see step 7 below). Can be either a single population, or one of the groups defined in **Groups_list**, or all the populations in the sweep file (then **Pop_interest** is set to "All")

-Run_FDR: yes or no. Run the FDR or not.

-Run_shuffling: yes or no. Run the shuffling of genome for FDR analysis or not, in the case it has already been done (the same shuffled genomes may be reused for a different set of genes of interest, or for re-running the same test, for example if a previous attempt to run the test was interrupted, but the genome shuffling step had completed).

-FDR_number: number of times the genome will be shuffled and the bootstrap test re-run on shuffled genomes to estimate a false discovery rate that takes clustering into account. Caution, a large **FDR_number** will take a LOT of disk space. Due to a quirk of the current version of the pipeline, FDR_number has to be a multiple of 8.

-Shuffling_segments_number: number of segments in which to cut the genome. The segments are then randomly flipped and their order randomly permuted. This effectively shuffles the sweeps while conserving the clustering of genes within the same sweeps.

-Interrupted_FDR_run: yes or no. Specifies if you want to continue running a previously interrupted FDR analysis (yes), or if you want to start a whole new FDR analysis from scratch, in which case previously generated files for FDR present in the test_outputs folder will be removed (no).

Step 6: run in parallel the bootstrap test to create as many control gene sets as needed (typically 10,000 or less, see step 5 above **Iterations_number** and **Runs_number**).

Step 7: After the bootstrap test is done matching control gene sets with the set of genes of interest, a specific script counts sweeps that overlap all the genes in each gene set, in each included population, each group of populations defined in **Group_list**, and all populations considered together. A sweep represented in multiple populations of a group or of the complete population list is counted only once when computing the number of sweeps at the group-level or all populations-level, to avoid confusing the number of sweeps and the geographic spread of sweeps. This problem does not exist when looking at each population separately.
If a specific population or group, or all populations are the focus of interest, the script that counts sweeps takes as parameter the name of the population or group (**Pop_interest**), or All (for all populations) and explores the control sets depending on the first 100 random control sets. If a significant excess (P<=0.05) or significant deficit (P>=0.95) is found for at least one rank threshold, then the script explores 1,000 control sets. If there are thresholds with updated P-values below 0.002 (excess) or above 0.998 (deficit), then the script explores all the 10,000 control sets to get accurate p-values. Modulating the effort of exploration of the control set speeds up the analysis when it is clear that there are not very low or very high p-values to estimate and scanning all the control sets is not necessary to know that the test is not significant, or weakly so.

Step 8: When a significant excess or deficit is found at step 7, and if **Run_FDR** is set to yes, the pipeline starts the FDR analysis based on shuffling genome segments (see above). Then steps 6 and 7 are repeated, this time on the shuffled genomes as many times as specified by **FDR_number**. The script get_FDR.pl then estimates the FDR. Note **that** if the FDR analysis is interrupted for a reason or another, you can keep previous progress and restart when things stopped by setting the **Interrupted_run** parameter to yes.

Step 9: Estimation of the FDR. This is the most time consuming step, especially if the FDR analysis includes many iterations. Typically if the FDR is expected to be low, you may need to set **FDR_number** to 10,000 or even more. This will take time and a lot of hard disk space due to the storage of the shuffled genomes used for the FDR analysis.