Joint Likelihood Mapping 2 (JLIM_2.0)

JLIM is a cross-trait test of shared causal effect, which is described in Chun et al. 2017. JLIM tests whether two traits – main and secondary – are driven by shared causal effect or not. Typically, the main trait is a large GWAS study, and secondary trait can be an expression Quantitative Trait Loci (eQTL) association study. For main trait, JLIM takes only summary-level association statistics, but for secondary trait, it requires genotype-level data to generate permutation-based null distribution. JLIM is simultaneously released at Cotsapas lab github and Sunyaev lab website.

JLIM 2.0

Has the added functionality of obtaining association statistics for the secondary trait in a cohort specific manner and then combining them before running JLIM. It is designed for meta-analyses of secondary trait cohorts with matching ancestries. JLIM 2.0 is described in preprint.

Download

Code

- JLIM release v2

Example

- JLIM2 example data

Obstructive Sleep Apnea Data for Chun and Akle et al.

- AvSaO2:
  - discovery cohort association summary statistics
  - candidate genomic intervals
- MinSaO2:
  - discovery cohort association summary statistics
  - candidate genomic intervals
- AHI (rdi3p):
  - discovery cohort association summary statistics
  - candidate genomic intervals
- Event Duration:
  - discovery cohort association summary statistics
  - candidate genomic intervals

How to install

The core JLIM module is implemented as an R extension (**jlimR**). **jlimR** depends on **getopt** module. If it is not installed, **getopt** can be installed from CRAN by:

```
Rscript -e 'install.packages("getopt", repos="http://cran.r-project.org")'
```

After **getopt** has been installed, **jlimR** (included in the distribution file) can be installed by:

```
tar -zxvf JLIM_2.0_NE.tar.gz

R CMD INSTALL JLIM_2.0_NE.tar.gz
```

In case that it is preferred to install R extensions in your home directory (e.g. ~/R) instead of the default system path, please do the following instead:

```
Rscript -e 'install.packages("getopt", "~/R",
repos="http://cran.r-project.org")'

R CMD INSTALL -l ~/R jlimR_1.0.2.tar.gz
```

And then, add your local R library path to **R_LIBS** environment variable in .bashrc or .profile as:

```
export R_LIBS=~/R:$R_LIBS
```

The python based pre-processing step for JLIM 2 depends on numpy and scipy. Please check that your versions of the following packages are at least:

python 2.7.9

numpy 1.14.3

scipy 1.0.0

How to run JLIM on provided example

Example data

Please download and untar the JLIM_2_example.tar.gz in the same directory that the JLIM_2.0_NE.tar.gz was untared. The example/folder should be in parallel with the bin/ and R/ directories. In JLIM_2.0/example, we provide the following simulated dataset: Data from three simulated cohorts (A,B,C) in chromosome 1- Bimbam and map files for each cohort. A peaks file containing the mid point of each region in chromosome 1 that will be analyzed. A reference LD file corresponding to the two loci analyzed. An IndexSNP file (Height/Height_indexSNP.tsv) with both loci under analysis and two primary trait

statistics files (Height) containing only summary statistics. We also include a phenotype file, a samples file and a covariate file for each cohort.

To run the example after JLIM is installed, run all commands in the file CommandsExample.sh:

cd example

./CommandsExample.sh

Files needed to run JLIM

Bimbam file

One genotype file in BIMBAM format is needed per cohort, per chromosome analyzed, for the secondary trait. These files should have no header and should be gzipped. The separator used in the file should be provided (see bimbam.separator). Currently, JLIM does not allow missing genotypes.

Map file

The map (or info) file is a chromosome and cohort specific file with information on all variants present in the corresponding Bimbam file. It should have no header and it should be gzipped. The number of lines in the map file and bimbam file should match exactly, with each line corresponding to a variant. The separator used in the file should be provided (see map.separator)

Peaks file

This file contains a comma-separated list of genomic positions to be analyzed in the corresponding chromosome. These will be at the center of intervals of size (in bp) “interval.int” which will be analyzed using JLIM.

Samples file

Is a file with no header containing the sample names of all samples in the bimbamFile in the same order. These should match exactly. The sample names will be used to find the appropriate phenotypes and covariates. There should be one sample per row, and a single column

Secondary phenotype file

This file has samples names in the first column and their corresponding phenotype in the second column. Phenotypes that correspond to data sets where individual level data is provided are called secondary phenotypes, as opposed to primary phenotypes, where only summary statistics are needed. They should

either be quantitative values or “nan”, which will exclude the sample from the regression. The file should have a header and should be tab separated.

Covariate file

This is a tab separated file which has a header with covariate names. Rows are samples, columns are covariates. The first column (column 0) contains sample names. There should be no missing values (‘nan’) in any of the covariates or samples used in the regression. You are free to include samples and covariates not used in the regression in this file, as the samples used will be specified in the phenotype file and the covariates used will be specified in the argument covariates.list.

IndexSNP file

The indexSNP file is a tab-delimiated file with five columns: - CHR: chromosome - SNP: SNP ID of index SNP - BP: bp position of index SNP - STARTBP: start of interval around index SNP (bp) - ENDBP: end of interval around index SNP (bp)

CHR.*STARTBP*.*ENDBP* combination will be used as an interval identifier to locate files of primary association statistics, secondary association statistics, and reference LD. In each interval, the most associated SNP will be automatically picked based on primary association p-values, and then the analysis window will be set up to +/- 100kb around the most associated SNP. The JLIM analysis window will not extend over the original interval specificied in the indexSNP file. The exact bp position of index SNP does not matter for JLIM as it will pick the most associated SNP within the specified interval.

Primary trait summary statistics file

Primary trait file is named by *TraitName*.*CHR*.*STARTBP*.*ENDBP*.txt. It is space-delimited and has CHR, BP, and SNP. It also has to carry one of STAT, T, or P. If P is specified, the two-sided P-value will be transformed into Z-statistic. STAT or T will be approximated as Z-statistic.

Reference LD file

Reference LD files are provided one for each interval. It is a tab-delimited file without header. The file name is specified as locus.*CHR*.*STARTBP*.*ENDBP*.txt.gz. Each row is a marker, and it contains the following columns: CHROM, POS, ID, REF, ALT, QUAL, FILTER, and is followed by two alleles for each individual.

Intermediate files generated by JLIM

-data file

-snps file

-positions file

-dosage file

-assoc file

-vars file

-betas file

-permutation file

-meta.assoc file

Config file

The config file is generated by jlim_gencfg.sh, and contains the following column. JLIM will be executed on each row separately. The config file contains the following columns: - maintrID: name of main trait. Specified by –tr1-name. - chrom: chromosome - idxSNP: index SNP name (SNP in indexSNP file) - idxBP: index SNP pos (BP in indexSNP file) - idxP: P-value of association to primary trait at idxSNP - idx2BP: SNP that is most associated to primary trait (automatically selected by JLIM) - idx2P: P-value of association to primary trait at idx2BP - start: start of JLIM analysis window - end: end of JLIM analysis window - sectrID: name of secondary trait (“LCL” in the above example) - sectrsubID: sub-identifier of tested association in secondary trait folder (gene names in the above example) - minP2: smallest p-value of association to secondary trait within JLIM analysis window. By default, jlim_gencfg.sh excludes secondary traits with minP2 >= 0.01 from config file. The cut-off can be changed by –p-tr2-cutoff option.
- maintr: file path to main trait association file - sectr: file path to secondary trait association file - refld: file path to reference ld file - mainld: file path to in-sample ld of main trait cohort if specified (use refld if set to “.”) - secld: file path to in-sample ld of secondary trait cohort - perm: permutation file of secondary trait association

Output file

The JLIM out contains two columns: - STAT: JLIM statistic - p: p-value by permutation. The p-value of 0 means that the JLIM statistic is more extreme than permutation.

Running JLIM on your data

First, assemble all loci you want to analyze and create one peaks file per chromosome, with a comma-separated list of all the genomic coordinates of the focal SNPs. For each cohort obtain a samples file and for each cohort/chromosome, obtain a corresponding bimbam and map file. Make sure you prepare all files mentioned under *Files needed to run JLIM*

1) Generate the reference LD

We provide a sample script to extract LD info for non-Finnish Europeans from downloaded 1000 genomes project vcf files. For example, if the vcf files are present in /data/1000genomes/ftp/release/20130502:

```
fetch.refld0.EUR.pl /data/1000genomes/ftp/release/20130502/
primary_trait/indexSNP.tsv ld0/
```

2) For each cohort/chromosome, run the “CutBimbam.py” script

```
python CutBimbam.py bimbam_file map_file peaks_file samples_file output_string
bimbam_separator_string map_separator_string map_position_int Maf_float
interval_int
```

With the corresponding arguments:

- output_string - Should be cohort_name.chr, where chr is the chromosome number

This will be the beginning of the name of the output files: the snps and data files. Use the name of the cohort and the chromosome number separated by a “.” as in the example. You can include the folder where these files will be stored as a prefix (see example)

- bimbam_separator_string - This is the character that separates columns in the bimbam file.
- map_separator_string - This is the character that separates columns in the map (or info) file.
- map_position_int - In the map file, the column number that holds the genomic positions will be denoted by map_position_int . The first column corresponds to a 0

Optional arguments:

- Maf_float - Default 0.05. This is the minor allele frequency (MAF) cutoff . The script will eliminate all SNPS with a minor allele frequency below this cutoff
- interval_int - Default 200,000. This is the size of the interval (in BP), which will be analyzed by JLIM
- Output - The script will produce a ‘data’ and a ‘snps’ file for each peak in the chromosome. These will be named with the output_string and the chromosome coordinates as: output_string.start_BP.end_BP.snps.gz, where start_BP and end_BP are the chromosome coordinates of the boundaries of the interval analyzed. These will in turn be the coordinates of each focal SNP listed in the peaks file +- interval_int/2

3) Make directories

For each locus, make a directory where all files with meta-analyzed statistics will be stored, so that JLIM

can use the directory to obtain the permutation p-values. These folders should follow the naming format locus.chr.startBP.endBP. We recommend storing these folders in a directory with the secondary phenotype's name.

```
mkdir secondary_phenotype_name
mkdir secondary_phenotype_name/locus.chr.startBP.endBP
```

4) For each locus, run the "Makedosage.py" script:

This script will merge all genomic data in the different 'data' and 'snps' files to make a dosage file with the genotypes of all samples in the locus. It also makes locus-cohort specific position files.

```
python Makedosage.py cohort_number_int chr
secondary_phenotype_name/locus.chr.startBP.endBP/dosage_file cohortA_data_file
cohortA_snps_file cohortB_data_file cohortB_snps_file
```

With the corresponding arguments:

- cohort_number_int - The number of cohorts that will be meta-analyzed together
- chr- The chromosome number
- dosage_file - This file will have the genotypes of all samples in the locus, and will be used to calculate in-sample LD by JLIM. Make sure to include the appropriate path to where the file should be stored
- cohortA_data_file and cohortA_snps_file - list all data and snps files from the cohorts that will be meta analyzed together. The script expects cohort_number_int data files and cohort_number_int snps files

5) Run regressions

For each locus-cohort-secondary phenotype, run the "RunRegressions.py" script: This script will obtain summary statistics for each locus. It will generate permutation files with estimated effect sizes (betas_files) and estimated variances (vars_files), as well as individual summary statistics files (assoc_files)

```
python RunRegressions.py data_file snps_file samples_file
secondary_phenotype_file covariates_file regression_id_string covariates_list
permutation_number_int chr cohort_number_int
```

- data_file, snps_file - These files should be locus and cohort specific
- samples_file phenotypes_file covariates_file - These files should be cohort specific
- regression_id_string - This argument will be used in the naming of the outputs of this script. It

should include the cohort name, the locus ID (chr.start.end) and the name of the secondary phenotype included in the secondary phenotype file. It can also include a folder where the outputs will be stored. Ideally: cohort_name.chr.start_bp.end_bp. secondary_phenotype_name

- covariates_list - This is a comma-separated list, which indicates the covariates from the covariate_file that will be used in the regression. The left-most column (column 0) should correspond to the sample names. The first covariate column after the sample name would be column 1.

- permutation_number_int - The number of permutations that will be generated. This determines the lower bound and precision of the JLIM p-value.

- chr - The chromosome number

- cohort_number_int - The number of cohorts that will be meta-analyzed together

6) Meta analyze

For each locus-secondary phenotype, when all of the cohort specific regressions have been finished, run "METAmergecohorts.py" to combine the cohort specific statistics into meta-analyzed files. These will include a summary statistic association file (meta.assoc_file) and a permutation file (meta.dump.all)

```
python METAmergecohorts.py cohort_number_int assoc_file
secondary_phenotype_name/locus.chr.startBP.endBP
/meta_id_string beta_file_1 vars_file_1 betas_file_2 vars_file_2…..
```

- For example if two cohorts (A and B) are being meta analyzed:

```
python METAmergecohorts.py 2 CohortA_assoc_file
secondary_phenotype_name/locus.chr.startBP.endBP
/meta_id_string CohortA_beta_file CohortA_vars_file CohortB_beta_file
CohortB_vars_file
```

With the following arguments:

- cohort_number_int - The number of cohorts that will be meta-analyzed together

- CohortA_assoc_file - Use any assoc_file from the locus. The specific cohort that generated this file should be irrelevant.

- meta_id_string - This argument will be used in the naming of the outputs of this script. It should include the set of cohorts combined, the locus ID (chr.startBP.endBP) and the name of the secondary phenotype included in the phenotype file. Make sure to include the appropriate path to where the files should be stored

- CohortA_beta_file CohortA_vars_file - The beta and vars files generated from the Runregression.py script. Each beta file should proceed the var file from the same cohort. Pairs of beta and vars files

from all cohorts that will be meta-analyzed should be provided.

7) Generate a config file

After all the loci in the indexSNP file have been meta-analyzed, for each primary-secondary phenotype pair, run “jlim_gencfg.sh” Using the indexSNP file, jlim_gencfg.sh scans the provided secondary phenotype folder to generate a config file.

```
./jlim_gencfg.sh --tr1-name primary_phenotype_name --tr1-dir
primary_phenotype_folder --tr2-dir secondary_phenotype_folder --idxSNP-file
primary_phenotype_folder/indexSNP.tsv  --refld-dir ld0_folder --out
primary_phenotype_name. secondary_phenotype_name.cfg.tsv --tr2-genotype-
filetype dosage
```

- primary_phenotype_folder - This folder should contain the indexSNP file and all the primary trait summary statistics files which correspond to the loci listed in the indexSNP file
- secondary_phenotype_folder - This folder should contain all locus.chr.startBP.endBP folders generated in the steps 1-5 above.
- ld0_folder - This folder should contain reference LD files corresponding to all loci in the indexSNP file. Reference LD files are a tab-delimited file without a header. The file name is specified as locus.chr.startBP.endBP.txt.gz. Each row is a marker, and it contains the following columns: CHROM, POS, ID, REF, ALT, QUAL, FILTER, and is followed by two alleles for each individual.
- –tr2-genotype-filetype dosage - Include this flag so that JLIM knows to use a dosage file to calculate in-sample LD

8) Run JLIM

Finally, for each config file created (one per primary-secondary phenotype pair) run “run_jlim.sh”. This script runs JLIM in all loci in the config file, storing the statistics and p-values in a corresponding FIN file, and the logs in a LOG file

```
../run_jlim.sh primary_phenotype_name. secondary_phenotype_name.cfg.tsv
resolution_float primary_phenotype_name. secondary_phenotype_name.FIN.tsv >
primary_phenotype_name. secondary_phenotype_name.LOG.tsv
```

- resolution_float - This is the r2 resolution limit. The default value is 0.8

Excluding samples

If you need to exclude any samples, you should exclude them from the secondary phenotype file, or set their phenotype to “nan”. This file can only take quantitative values (or nan). The sample file should

match the genotypes in the bimbam file exactly, so no samples can be excluded from this file.