JAG: Joint Association of Genetic Variants

V1.1 - Documentation

Esther Lips & Danielle Posthuma

Complex Traits Genetics

VU University Amsterdam, The Netherlands

ctglab.nl

October 6, 2014

Contents

Chapter 1

Getting started with JAG

JAG is a free open source tool to run gene-set analysis in GWAS data. It uses raw data as input and includes bot self-contained and competitive tests.

1.1 Requirements

JAG runs on UNIX/Linux, Mac OS X and Windows (from V1.1) operating systems. JAG requires Python v2.6 or higher, which can be downloaded via http://python.org/getit/. JAG further relies on PLINK (Purcell, et al., 2007) for genetic association tests and R for generating plots. PLINK can be downloaded from: http://pngu.mgh.harvard.edu/~purcell/plink/. R can be downloaded from http://cran.r-project.org/.

1.2 Installing JAG

JAG is a free, open source tool and can be downloaded from http://ctglab.nl/software/. The files included in the zipped archive are:

- example_data folder containing example.bed, example.bim, example.fam, example.set
- hg18 folder containing auxiliary gene- and snp location files in hg18 format
- hg19 folder containing auxiliary gene- and snp location files in hg19 format
- jagV* folder containing jag executable + subfolder containing jag source code

After download, unzip and copy all the files to your working directory. Since JAG is a command line tool, you need to open a terminal window and type commands at the prompt to perform analyses with JAG. JAG can be invoked by typing './jag' at the prompt, from the directory where JAG is installed.

It is also possible to invoke JAG by typing 'jag' (i.e. without the ./) at the prompt and run JAG from any directory by adding the JAG executable to the PATH variables. How to do this, depends on the environment you are working in. If you are unfamiliar with this, it is advised to consult your local system administrator. Usually, typing the following commands does the trick:

sudo cp -R path_to_jag_folder /usr/local/lib

sudo ln -s /usr/local/lib/jagV1.0/jag /usr/local/bin/jag

Or, alternatively try:

alias jag= [insert path to the jag executable]

1.3 Citing JAG

When using JAG, please cite both the software and the manuscript describing the methods.

Package: JAG (v*)

Author: Esther Lips, Maarten Kooyman

URL: http://ctglab.nl/software/jag/

Lips, ES, Kooyman M, de Leeuw C, Posthuma D JAG: a Computational Tool to Evaluate the Role of Gene-Sets in Complex Traits. submitted

1.4 Reporting problems, bugs and questions

We assume you first read the manual. If that does not solve your question, you can send us an e-mail in which you explain your specific problem and include:

- Used command
- Command line error reporting
- Log file
- The type of operating system you are using
- (part of the files) that you use as input

You can send your mail to e.s.lips@vu.nl or d.posthuma@vu.nl

Chapter 2

Resources for download

2.1 Example dataset

The example files can be found on our website, and contains HAPMAPIII CEU genotype data downloaded from the Hapmap FTP site (accession date: 13th of January, 2012); ftp://ftp.ncbi.nlm.nih.gov/hapmap/genotypes/2009-01_phaseIII/plink_format/.
The SNPs in this example dataset are mapped to the NCBI build 36 coordinates supplied by Hapmap. For example purposes, we filtered this dataset for SNPs that are included on the Illumina 1M array. Furthermore, we removed fifty-three non-founders and one individual due to heterozygosity on haploid genotypes. Subsequently we randomly assigned case/control status to the remaining 111 subjects (56 cases/55 controls).

The example data contains the following files:

```
example.bed  PLINK format binary PED file with genotype information
example.fam  PLINK format file describing individuals in the sample.
example.bim  PLINK format file describing all marker positions
example.set  file containing Entrez GeneIDs and set names of three randomly
             selected pathways from the KEGG database
```

2.2 Auxiliary files

```
hg18.gene.loc     File containing locations of genes in standard hg18 format
hg18.snp.loc      File containing locations of SNPs in standard hg18 format
hg19.gene.loc     File containing locations of genes in standard hg18 format
hg19.snp.loc      File containing locations of SNPs in standard hg18 format
```

2.3 Quickstart

Below is an example of the series of commands to perform self-contained and competitive analyses. This part is only meant as a quick reference. So, please read the manual for more information on each command.

1: Annotate SNPs to genes in specified gene-sets; obtain **`*.set.annot`** and **`*.allgenes.annot`** files:

```
jag --snp2gene example.set --gene_loc hg18.gene.loc --snp_loc hg18.snp.loc
```

2: Run self-contained gene-set analyses; obtain output in **`*.empp`** file:

```
jag --bfile example --perm 10000 --set jag.set.annot
```

3: Run competitive gene-set analysis for specific gene-set - matched for number of genes

3a: Draw matched random sets of genes, option: matched for ngenes (generates jag.draws_ngenes.set.annot):

```
jag --ndraw 100 --draw_ngenes set1 --set jag.set.annot --pool jag.allgenes.annot
```

3b: Run self-contained tests on randomly drawn, matched control sets (generates control_empp jag.draws_ngenes.P1.empp):

```
jag --bfile example --set jag.draws_ngenes.set.annot --perm 10000
```

3c: Calculate competitive P-value provides the competitive P-value in the log file: "Empirical P-value for competitive test for gene-set xxx = 0.xx"):

```
jag --orig_empp jag.merged.P1.empp --control_empp jag.draws_neff_genic.P1.empp -
-gene_set set1
```

Alternative series of commands where several sets of permutations are run simultaneously:

1: *as above*

2: Run self-contained gene-set analyses; obtain output in *merged.empp file:

```
jag --bfile example --perm 100 --set jag.set.annot &
jag --bfile example --perm 100 --set jag.set.annot &
..
jag --merge jag
```

3: Run competitive gene-set analysis for specific gene-set - matched for number of genes

3a: *as above*

3b: Run self-contained tests on randomly drawn, matched control sets (generates control_empp **jag.draws_ngenes.P1.empp**):

```
jag --bfile example --set jag.draws_ngenes.set.annot --perm 100 &
jag --bfile example --set jag.draws_ngenes.set.annot --perm 100 &
..
jag --merge jag.draws_ngenes
```

3c: *as above*

Chapter 3

Usage of JAG

This chapter is intended to provide an overview of the features included in JAG. The supplied examples given in this chapter are based on the example data that is provided on the JAG website.

3.1 Input files

JAG needs the following input files:

PLINK-format files.

(Please consult the PLINK documentation if you are not familiar with these file formats (http://pngu.mgh.harvard.edu/~purcell/plink/data.shtml#bed/))

`*.bim`

`*.bed`

`*.fam`

As JAG permutes phenotypes, it is crucial that the .fam file does not contain individuals with missing values on the phenotype of interest. If there are missing phenotypes, it is advised to run PLINK using **`--make-bed`** and **`--remove`** to remove these individuals before using JAG.

Standardized gene location file.

This file provides information on the location of genes, which is used to determine which SNPs belong to which gene(s). It should contain five columns, separated by a tab denoting Entrez GeneID, chromosome number, transcription start site (TSS) in bp, transcription end site (TES) in bp and chromosomal orientation. This file should not contain a header. Pre-compiled gene location files containing the coordinates of the protein-coding genes for both HG build 18/NCBI build 36.3 (**`hg18.gene.loc`**) and HG build 19/NCBI build 37.3 (**`hg19.gene.loc`**) are available on our website. The data of these genelists were retrieved from NCBI's ftp.

`*.gene.loc`

For example:

```
79501          1          58954          59871          +
729759         1          356723         358460         +
81399          1          610959         611897         -
148398         1          850984         869824         +
26155          1          869446         884542         -
339451         1          885830         890958         +
….
```

Standardized snp location file.

This file contains three columns separated by tab, denoting RS number, Chromosome, and Position (in bp). It has no header.

Files for all validated SNPs for dbSNPv130 (hg18.snp.loc, to be used in combination with **`hg18.gene.loc`**) and dbSNPv135 (hg19.snp.loc, to be used in combination with **`hg19.gene.loc`**) are distributed with JAG. These lists are retrieved from USCS database (http://genome.ucsc.edu/) and filtered for common SNPs for which validation status is not 'unknown'. Note: it is important that the snp IDs (rs number) used in the .bim file follow the same convention as the snp IDs in the snp.loc files.

`*.snp.loc`

For example:

```
rs10218492         1             691
rs10218493         1             767
rs10218527         1             790
rs10218495         1             944
rs9803797          1            1272
rs4849250          1            1809
...
```

A gene set file

A tab -delimited gene set file containing one Entrez GeneID per row and the name of the gene-set it is assigned to. This file should not contain a header. It is important that Entrez GeneIDs are used when using the provided .loc files for annotation. For memory saving purposes, it is advised to use short names for the gene-sets. The name should not contain spaces or tabs.

`*.set`

For example:

```
10000          hsa04010
100137049      hsa04010
10125          hsa04010
10235          hsa04010
10368          hsa04010
10369          hsa04010
….
```

3.2 SNP to gene annotation

The first step in running gene-set analysis with JAG is to map SNPs to the genes specified in the gene-sets. JAG maps SNPs to genes based on the provided transcription start site (TSS) and transcription end site (TES) of these genes. In addition, it is possible to map SNPs to genes within a user-specified 'regulatory region' around the gene boundaries with a maximum of 100Kb up- and/or downstream from the TSS and/or TES. Assuming all necessary files are in the working directory, the mapping of SNPs to genes can be called with the following command:

```
jag --snp2gene example.set --gene_loc hg18.gene.loc --snp_loc hg18.snp.loc
```

The **--snp2gene** option expects a parameter that specifies the location of the file with the gene-sets to annotate. The **--gene_loc** option expects the path to the location of the list of genes with their TSS and TES (gene.loc file). The **--snp_loc** option expects the location of the snplist (snp.loc file). In this example the output files will be saved with the default '*jag*' prefix. Adding the **--out** option to the command line allows specifying a custom prefix for output files.

By default, JAG does not map SNPs to a regulatory region. When a regulatory region around the gene is desired, the size of these regions (in Kb) can be specified with the **--up** and **--down** options. If a gene is oriented on the + strand the Kb's specified after **--up** will be subtracted from the TSS position and the Kb's indicated with **--down** will be added to the TES position of that gene, where in the case of a gene is oriented on the - strand the Kb's specified after **--up** will be added to the TES position and the Kb's after **--down** are subtracted from the TSS position. For example:

```
jag --snp2gene example.set --gene_loc hg18.gene.loc --snp_loc hg18.snp.loc --up 7.5 --
down 5
```

The latter command will generate the following output:

```
    ############################################################
    #                                                          #
    #     JAG  - Joint Association of Genetic variants - V1.1  #
    #            2012, Esther Lips & Danielle Posthuma         #
    #              GNU General Public License, v2              #
    #                                                          #
    #                Complex Trait Genetics Lab                #
    #               http://ctglab.nl/software/jag/             #
    #                                                          #
    ############################################################

Save logfile as [jag.log]

Analysis started: Wed Jul 11 13:59:38 2012

Used options:
     --snp2gene example.set
     --gene_loc hg18.gene.loc
     --up 7.5
```

```
     --down 5
     --snp_loc hg18.snp.loc

Found 19249 unique genes in hg18.gene.loc
Found 676 unique genes in example.set

No gene boundaries are found for:
geneID 100133583
geneID 2558
geneID 100506658
geneID 652614
geneID 3126
geneID 2567
geneID 3125
geneID 4950
geneID 100137049

Mapping SNPs to genes on chromosome 1
Mapping SNPs to genes on chromosome 2
Mapping SNPs to genes on chromosome 3
Mapping SNPs to genes on chromosome 4
Mapping SNPs to genes on chromosome 5
Mapping SNPs to genes on chromosome 6
Mapping SNPs to genes on chromosome 7
Mapping SNPs to genes on chromosome 8
Mapping SNPs to genes on chromosome 9
Mapping SNPs to genes on chromosome 10
Mapping SNPs to genes on chromosome 11
Mapping SNPs to genes on chromosome 12
Mapping SNPs to genes on chromosome 13
Mapping SNPs to genes on chromosome 14
Mapping SNPs to genes on chromosome 15
Mapping SNPs to genes on chromosome 16
Mapping SNPs to genes on chromosome 17
Mapping SNPs to genes on chromosome 18
Mapping SNPs to genes on chromosome 19
Mapping SNPs to genes on chromosome 20
Mapping SNPs to genes on chromosome 21
Mapping SNPs to genes on chromosome 22
Mapping SNPs to genes on chromosome X

Saving results...
Saved snps2allgenes file as jag.allgenes.annot
Saved snps2geneset file as jag.set.annot

Finished analysis: Wed Jul 11 14:00:39 2012
```

The results of snp2gene mapping are saved in two files;

- **jag.allgenes.annot**; includes the SNPs mapped to all the genes in '**hg18.gene.loc**'.
- **jag.set.annot**; includes the SNPs mapped to all the genes in '**example.set**' are saved, excluding the genes for which no boundaries were present in th ehg18.gene.loc file.

The two output files each contain three columns separated by a tab, denoting rsID, geneID and gene-setID:

```
rs12966656            1000 hsa04514
rs17498099            1000 hsa04514
rs17522784            1000 hsa04514
rs1148381             1000 hsa04514
```

The **jag.allgenes.annot** file has the set-name 'CODING_GENES' as 3rd column, and can be used in the competitive test to draw random sets of genes.

TIP: There is no upper or lower bound on the size of a gene-set although it is generally advised to keep gene-set size between 10-300 genes for interpretation purposes (Ramanan, et al., 2012)

TIP: When a large number of gene-sets are tested (>100) this may take a lot of memory and can slow down the analyses. In this case, it is advised to split up your analyses in different batches, i.e. gene-sets 1-100 in set file #1 and gene-sets 101-200 in set file # etc.

3.3 Self-contained test

3.3.1 Basic self-contained gene-set test

The self-contained gene-set analysis tests the null hypothesis that the gene-set is not associated with the trait. If association is partly due to population stratification, a self-contained test might be biased. To avoid spurious outcomes it is advised to carry out a genomic control correction or to run a multidimensional scaling analysis and control for population stratification using e.g. EIGENSTRAT2 derived ancestry informative principal components scores (Price, et al., 2006) . For statistically significant gene-sets it is advised to run additional competitive tests (see below).

The significance of the self-contained test in JAG is obtained by using permutation. JAG uses a multivariate SNP test statistic, which is calculated by summing the $-\log_{10}$ of the P-values obtained from single SNP association for all the SNPs in the gene-set(s). JAG invokes PLINK to run genome-wide association. By default, JAG uses the basic **`--assoc`** from PLINK, which is applicable to both case/control association analysis and quantitative trait analysis. However, more advanced association analyses are also possible and are described in Chapter 3.3.2.

The empirical P-value for a gene-set is calculated as follows: JAG first calculates and stores the $\mathbf{\Sigma}$-$\log_{10}$(P) across all phenotypes for each SNP based on the original dataset. Then for each permutation JAG also calculates the $\mathbf{\Sigma}$-$\log_{10}$(P). When finished permuting, JAG obtains the empirical P-value (P_{EMP}) for each SNP by dividing the number of times the $\mathbf{\Sigma}$-$\log_{10}$(P) from the permuted analyses exceeds or equals the $\mathbf{\Sigma}$-$\log_{10}$(P) from the original analysis (hits, H) by the number of permutations run (N).

The empirical P valued based on the Σ-$\log_{10}$(P) test statistic tests the hypothesis that the multivariate pattern of P-values of all SNPs in a gene-set is significantly different than what is expected under the null hypothesis of no association, given LD structure of SNPs.

JAG uses the added value of multiple SNPs and will be most powerful when multiple SNPs in a gene-set show at least some evidence of association. The following command line calls the self-contained test in JAG:

```
jag --bfile example --set jag.set.annot --perm 10
```

The **--bfile** specifies the prefix of the binary files including the genotype data (.bim, bed and fam files). The **--set** option specifies the path to the file which was generated in the snp2gene step and contains the SNP IDs and gene IDs of the gene-sets of interest. Option **--perm** specifies the number of permutations that need to be carried out for the self-contained test. For example purposes we only use 10 permutations here. Typically one would run at least 10,000 permutations. For multiple permutations it may be wise to reduce computation time by using a cluster computer and run multiple sets of permutations in parallel (see also Chapter 3.3.4).

The command above generates the following output:

```
    ###############################################################
    #                                                             #
    #     JAG  - Joint Association of Genetic variants - V1.1     #
    #            2012,  Esther Lips & Danielle Posthuma           #
    #              GNU General Public License, v2                 #
    #                                                             #
    #                Complex Trait Genetics Lab                   #
    #               http://ctglab.nl/software/jag/                #
    #                                                             #
    ###############################################################

Save logfile as [jag.log]

Analysis started: Wed Jul 11 14:53:05 2012

Used options:
     --bfile example
     --set jag.set.annot
     --perm 10

Running association analysis using PLINK with command:
--bfile example --assoc

Results of PLINK saved as jag.results.P1.assoc
Saved sumlog file as jag.P1.sumlog
Saved QQ plot from association analysis as jag.assoc_qq_plot.P1.pdf
Saved QQ plot(s) from gene set(s) as jag.qq-plot_all_sets.P1.pdf

Running permutations...
permutation 1 ready
permutation 2 ready
permutation 3 ready
permutation 4 ready
permutation 5 ready
permutation 6 ready
permutation 7 ready
permutation 8 ready
permutation 9 ready
permutation 10 ready

Saved permutation results as jag.IX6T1B0y.P1.perm
Saved empirical pvalues as jag.P1.empp
Saved distribution plot as jag.distribution_sumlogs.P1.pdf
```

```
Finished analysis: Wed Jul 11 14:54:41 2012
```

With the command given above, JAG runs an association analysis in PLINK for all the SNPs, and calculates the **Σ**-$\log_{10}$(P) for each of the three gene-sets in **jag.set.annot.** Subsequently, JAG permutes the phenotypes in the .fam file 10 times and runs the association test on each of the 10 permuted datasets. The fam file should not contain individuals that have a missing phenotype (e.g. status -9), to ensure that the same individuals (i.e. the same genotypes) are included in the permutations.

TIP: JAG uses a randomly generated seed for the permutation procedure. In some cases it might be useful to obtain exactly the same output as before, it is therefore also possible to set a fixed seed by using the **--seed** with an integer as an argument, or to use the seed numbers provided in **.perm** file. Use the following command to use a fixed seed:

--seed 1234

As a result from the self-contained test, JAG saves eight files (of which only the ***.log**, ***.P1.empp** and the ***.pdf** will be of interest to most users. Other files are generated as they may be of interest for advanced users):

- jag.log: this file contains the logfile

-jag.P1.empp contains the actual results from the self-contained test, and includes 9 columns denoting:

- Geneset: the name of the gene-set
- sumLogReal: the test-statistic (∑-$\log_{10}$(P)) of the original data
  nperm: the number of permutations performed
- emp_p: the empirical p-value that is calculated over permutations
- nSNP: the number of tested SNPs in the gene-set
- nGenes: the number of tested genes from the gene-set that contain tested SNPs
- var(Perms): the variance of the distribution of the test-statistic from the permutations
- mean(Perms): the mean of the test-statistics from the permutations
- nEff: the effective number of SNPs in the gene-set

The effective number of SNPs is based on the empirical distribution of the Σ-$\log_{10}$(P) under the null hypothesis of no association of the *N* permutations (see (Purcell, et al., 2009). Briefly, under the null hypothesis of no association,–$\log_{10}$(P) is distributed as 1/(2ln(10)) = 0.217 times a $\chi 2$ with 2 degrees of freedom. If all M SNPs are independent

then Σ–$\log_{10}$(P) has a mean of (0.217)(2M) and a variance of $(0.217)^2(4M) = 0.189M$. We define the effective number of SNPs (Neff) as

$$M_{eff} = \frac{M_{obs}[\sigma^2_{\exp,\sum -\log_{10}(p)} | SNPs_{ind}]}{\sigma^2_{emp,\sum -\log_{10}(p)}}$$

$$= \frac{M_{obs}[(0.217)^2(4M_{obs})]}{\sigma^2_{emp,\sum -\log_{10}(p)}}$$

$$= \frac{0.189M^2_{obs}}{\sigma^2_{emp,\sum -\log_{10}(p)}}$$

The expected mean and variance are calculated based on the number of SNPs that are summed to obtain the Σ-$\log_{10}$(P), and larger variance of the observed distribution than expected indicates dependency (i.e. due to LD) between included SNPs (see also (Lips, et al., 2011). The effective number of SNPs is used for competitive testing when matching randomly drawn gene-sets on the effective number of SNPs.

For the example dataset the **jag.P1.empp** is as follows:

Geneset	sumlogReal	nperm	emp_p	nSNP	nGenes	var(Perms)	mean(Perms)	nEff
hsa04010	3919.12	10	0.8	9195	264	14711.48	3982.08	1086
hsa04080	4108.02	10	0.0	8783	269	25331.12	3889.97	575
hsa04514	2922.94	10	0.2	6462	130	10844.74	2808.99	727

From the results above we can conclude that there is no evidence for association with gene-sets **hsa04010** and **hsa04514,** given their respective empirical P-values of 0.8 and 0.2. However, gene-set **hsa04080** has an empirical P-value of 0.0, indicating that in none of the 10 permutations the evidence for association was higher than in the original data, and that this gene-set is likely associated with the trait. Obviously, 10 permutations is not sufficient to draw any conclusion and for a real data analyses one would run 1,000 or 10,000 permutations.

- **jag.assoc_qq_plot.P1.pdf** contains a QQ plot for all the P-values of all SNPs genotyped in the sample. This plot can be used to inspect deviations of the single SNP p-values from expected under the null hypothesis of no association. In the **jag.qq-plot_all_sets.P1.pdf** file QQ plots are provided separately for each gene-set in a single .pdf file. These plots can be used to inspect deviation of the expected P-values within each of the gene-sets. Note that when 'R' is not installed, these plots are not generated.

-jag.distribution_sumlogs.P1.pdf provides a histogram of the distribution of the test-statistics of the permutations and the original data for each gene-set. The test-statistic for the original data is indicated with a red dotted line.

Other output files used internally by JAG:

- jag.results.P1.assoc: contains the results from the association test performed in PLINK. In case of using a quantitative trait, a file named **jag.results.P1.qassoc** is generated.

- jag.sumlog.P1.sumlog: contains the test statistic (i.e. the sum of the $-\log_{10}$ of all P-values assigned to a gene-set) for each gene-set, as well as the actual number of genes and SNPs tested and the value of the sum of the log of all P-values for those SNPs.

- **jag.JZFi61-P.P1.perm**, contains similar information but then from the permuted datasets. Specifically, it has *p* (+1 header) rows, and *g* (+1 seed number) columns, where *p* is the number of columns, and *g* is the number of tested gene-sets. For each gene-set the sum of the log of the P-values is given. The last column provides the used seed for the random number generator, allowing obtaining exactly the same result if needed. Note that the 'JZFi61-P' part is randomly generated and will be different each time you run JAG, except when you are using the **--seed**. In that case this part of the file naming is your seed number.

3.3.2. Using adjusted P-values

It is also possible to use the GC corrected P-values calculated by PLINK using option **--adjust** to the command (See PLINK manual for more information on adjusted P-values). This option generates the same output files as the basic self-contained test except for the results from the association test performed in PLINK, which will be saved as **jag.results.P1.assoc.adjusted**

An example of using the --adjust option in your command:

```
jag --bfile example --set jag.set.annot --perm 10 --adjust
```

NOTES: By using ***--out*** you can specify the prefix of the outfile, and by using ***--verbose***, the output of PLINK will also be printed to the screen/logfile.

3.3.3. Using alternate phenotype files

It is possible to use an alternate phenotype file by adding the **--pheno** option with the name of your alternate phenotype file as an argument. JAG will then run the gene-set analysis separately for each phenotype included in this file. Note that this alternate phenotype file should only contain individuals for which genotype data is available in the files indicated with **--bfile**. Furthermore, individuals with a missing phenotype (e.g. '-9') should be excluded from the alternate phenotype file.

An example command line to run a self-contained test including an alternate phenotype file is:

```
jag --bfile example --set jag.set.annot --perm 10 --pheno multi.pheno
```

3.3.4 Linear and logistic self-contained tests

By default, JAG runs a basic (**--assoc**) association test in PLINK when conducting a self-contained test. However, it is also possible to use the --linear or --logistic association tests in PLINK. Note however that these more complex types of analyses take more time to run in PLINK and therefore also in JAG.

Example command line for a case/control analysis (**--logistic**);

```
jag --bfile example --set jag.set.annot –logistic
```

Example command line a for a quantitative traits analysis (**--linear**) with an alternate phenotype file:

```
jag --bfile example --set jag.set --linear --pheno qt.pheno
```

3.3.5 Using a covariate file

It is also possible to adjust for covariates. When using a covariate file, the covariate(s) are permuted with the phenotypes, such that the relation between covariates and phenotypes remains the same and the genetic association test is always conducted on the same residuals. If sex is to be used as a covariate, it needs to be included in the covariate file, and should not be invoked with the option --sex

Including a covariate file, use:

```
jag --bfile example --set jag.set --covar mycov.cov --perm 10
```

NOTE: Files with covariates should only contain covariates for individuals for which genotype data is available in the files indicated with **--bfile**.

3.3.6 Gene based test

JAG also offers the possibility to perform a self-contained gene based test on all the genes in the file specified by **--set** by adding the option **--gene_based** to the self-contained test command. For example,

```
jag --bfile example --set jag.set.annot --gene_based --perm 10
```

```
    ############################################################
    #                                                          #
    #     JAG  - Joint Association of Genetic variants - V1.1  #
    #            2012,  Esther Lips & Danielle Posthuma        #
    #              GNU General Public License, v2              #
    #                                                          #
    #                Complex Trait Genetics Lab                #
    #               http://ctglab.nl/software/jag/             #
    #                                                          #
    ############################################################

Save logfile as [jag.gene_based.log]

Analysis started: Wed Jul 11 15:33:11 2012

Used options:
     --bfile example
     --set jag.set.annot
     --gene_based
     --perm 10

Running association analysis using PLINK with command:
--bfile example --assoc

Results of PLINK saved as jag.gene_based.results.P1.assoc
Saved sumlog file as jag.gene_based.P1.sumlog
Saved QQ plot from association analysis as jag.gene_based.assoc_qq_plot.P1.pdf
Saved QQ plot(s) from gene set(s) as jag.gene_based.qq-plot_all_sets.P1.pdf

Running permutations...
permutation 1 ready
permutation 2 ready
permutation 3 ready
permutation 4 ready
permutation 5 ready
permutation 6 ready
permutation 7 ready
permutation 8 ready
permutation 9 ready
permutation 10 ready

Saved permutation results as jag.gene_based.8BiiZwob.P1.perm
Saved empirical pvalues as jag.gene_based.P1.empp
Saved distribution plot as jag.gene_based.distribution_sumlogs.P1.pdf

Finished analysis: Wed Jul 11 15:34:45 2012
```

The files resulting files are equal to the files saved from a self-contained test performed on a gene-set, which are discussed in Chapter 3.2.1. The only difference is that the files from a gene-based test have 'gene_based' added to the prefix.

3.3.7 Parallel computing

A small number of permutations can be performed on the working computer. However, a large number of permutations need a lot of computation time. Therefore, it is useful to run the permutations in parallel on a computer cluster, and distribute multiple sets of permutations across different processors. When running on a computer cluster, it is advised to disable the calculation of the empirical P-value with the **`--no_emp`** flag and the generation of plots with **`--no_plots`**.

It is also advised to use the same prefix for all distributed, since the prefix (prefix.xx.perm, where xx is a randomly generated name) will be used in order to merge the results after all jobs are finished. When using the same prefix after **`--out`** the .empp file will be overwritten, but this is fine as the **`--merge`** function uses the files that contain the actual test-statistics (and that contain a random name in the filename). An example for which the results from multiple permuted resultfiles for a single prefix are merged, is given in the next command:

```
jag --merge myruns
```

With this function JAG will merge all output files starting with *jag.*.P1.perm* for phenotype 1 and saves them in **`myruns.merged.P1.perm`**. Subsequently JAG will use **`myruns.P1.sumlog`** to determine the number of hits and calculate the empirical P-values based on all permutations, which is saved in the file **`myruns.merged.P1.empp`**. In addition, a distribution plot is created, which includes a distribution of the test-statistics for all permutations, and saved as **`myruns.distribution_sumlogs.P1.pdf`**. When multiple phenotypes have been run, this function automatically merges all files within each available phenotype (P1, P2, etc) as long as they have a single prefix.

3.4 Competitive testing

3.4.1. Drawing random sets of genes

Competitive tests are robust against population stratification, and test whether a certain gene-set of interest is more strongly associated with the trait than a matched, random set of genes. These random sets can be matched with the gene-set of interest on the number of SNPs or on the number of genes. To fulfil both conditions is not feasible in practice as that would severely limit the pool of genes from which gene-sets can be drawn and create heavily dependence between the randomly drawn gene-sets, provided we want to draw at least 100 control gene-sets. Such dependency results in biased competitive P-values. JAG thus implements strategies that create random control gene-set either based on the effective number of SNPs or on the number of genes in the original gene-set.

In addition, random groups can be created from different pools, for example including all genic SNPs, all non-genic SNPs or all SNPs in genes expressed in brain. It is up to the user to change the pool from which genes or SNPs are to be drawn and thus to test different alternative hypothesis.

For every randomly drawn group, a self contained analysis needs to be run to obtain a self-contained empirical P-value which will then be evaluated against the self-contained empirical P from the original gene-set. Competitive testing thus comprises 3 steps:

1. Creating randomly drawn matched control gene-sets
2. Running self-contained analyses on the random control sets
3. Calculating the competitive P-value.

TIP: By default, JAG creates random draws from a pool excluding the SNPs located in the set on which the random draws are based. When desired, these SNPs can be included by adding **`--include`** to your command.

3.4.1.1. Random sets matched for the number of genes

In the next command, 50 randomly drawn gene-sets are generated that match the **`hsa04080`** gene-set for the number of genes:

```
jag --ndraw 50 --draw_ngenes hsa04080 --set jag.set.annot --pool jag.allgenes.annot
```

The option **--ndraw** specifies how many random gene-sets should be created. The option **--draw_ngenes** specifies that controls sets need to be matched on the same number of genes as in the target gene-set, which needs to be specified by name after this option. The option **--set** specifies the name of the file that includes the SNPIDs, geneIDs and genesetIDs. The option **--pool** specifies the name of the file containing the snp2gene mappings of the pool of SNPs and genes to be drawn from, for example an annotated list of all SNPs in all coding genes.

This command generates the following output on screen:

```
    ############################################################
    #                                                          #
    #     JAG  - Joint Association of Genetic variants - V1.1  #
    #             2012,  Esther Lips & Danielle Posthuma       #
    #               GNU General Public License, v2             #
    #                                                          #
    #                 Complex Trait Genetics Lab               #
    #                http://ctglab.nl/software/jag/            #
    #                                                          #
    ############################################################

Save logfile as [jag.log]

Analysis started: Wed Jul 11 15:53:59 2012

Used options:
     --ndraw 50
     --draw_ngenes hsa04080
     --set jag.set.annot
     --pool jag.allgenes.annot

Drawing 50 random gene sets...

Size of gene pool (--pool): 19238 unique genes

The geneset hsa04080 contains 270 genes

Minimum size of gene pool needed to draw 50 gene-sets (270*50): 13500

Saved random draws on number of genes as jag.draws_ngenes.set.annot

Finished analysis: Wed Jul 11 15:54:09 2012
```

The result of this command is saved in file **jag.draws_ngenes.set.annot**, which contains lists of SNPs included in each of the 50 draws. The file contains no header and three tab-delimited columns denoting SNPID, GeneID and Draw_number and will be used as input for the self-contained test over the random draws (See next section of this manual).

```
rs2721195          4796    Draw_1
rs2242268          4796    Draw_1
rs2242269          4796    Draw_1
rs2242264          4796    Draw_1
rs2242265          4796    Draw_1
rs4082352          4796    Draw_1
rs4082353          4796    Draw_1
rs741969           4796    Draw_1
…
```

Note that in this case we choose to generate 50 random sets of genes, as the pool of all coding genes is too small for drawing 100 sets of genes given the number of genes in the **hsa04080** gene-set. This number of random draws sets a lower limit to the competitive P-value: for 50 draws, the minimal P-value is < 1/50, or < .02 when no randomly drawn gene-set is more strongly associated than the original gene-set. For competitive testing, showing that the P-value is less than .05 is usually sufficient.

TIP: It is possible to use your own pool (**--pool**) to draw random gene--sets from, as long as the input files have the same format as described in this manual.

3.4.1.2 Random sets matched for the number of independent SNPs

It is also feasible to draw random sets of genes that are matched for the effective number of SNPs. In this case the competitive test tests the null hypothesis that there is 'no more evidence for association in the original gene-set than any other set of an equal effective number of SNPs'. For this purpose JAG calculates the effective number of SNPs per gene-set and draws an equal number of SNPs from a pruned SNPlist. SNP pruning is based on the input file of genotypes and JAG will invoke PLINK to do so, with command **--indep-pairwise 200 5 0.25** (See PLINK manual for more information on this). JAG accommodates drawing SNPs from independent SNPs within or outside genes or a combination of those two.

In the next example random sets of effective SNPs are generated from a pool of independent SNPs *within* genes, using the option **--draw_neff_genic**:

```
jag --ndraw 20 --bfile example --set jag.set.annot --draw_neff_genic hsa04080 --
neff_calc jag.P1.empp --pool jag.allgenes.annot
```

In this command 20 random gene-sets are drawn **(--ndraw)** from a set of genic SNPs **(--draw_neff_genic)** drawn from a pruned genotype file **(--bfile)**. The file given with **--pool** is generated in the pre-processing step of mapping SNPs to genes (see Chapter 3.2) and contains SNPs located in genes, including the SNPs within the regulatory region indicated with **--up** and **--down**. In this function, this file is used to determine whether independents SNPs are located within or outside a gene. The size of these draws is based on the number of independent SNPs for the gene-set given with the **--draw_neff_genic** command, which is extracted from the file given with the **--draw_neff_calc** option (which is the output file of the self-contained test in Chapter 3.2.2).

The command above gives an output like this:

```
############################################################
#                                                          #
#     JAG  - Joint Association of Genetic variants - V1.1    #
```

```
    #            2012,  Esther Lips & Danielle Posthuma          #
    #              GNU General Public License, v2                #
    #                                                            #
    #                Complex Trait Genetics Lab                  #
    #               http://ctglab.nl/software/jag/               #
    #                                                            #
    ##############################################################

Save logfile as [jag.log]

Analysis started: Wed Jul 11 16:05:59 2012

Used options:
     --ndraw 20
     --bfile example
     --set jag.set.annot
     --draw_neff_genic hsa04080
     --neff_calc jag.P1.empp
     --pool jag.allgenes.annot

Drawing random SNP sets (based on nEff)...

Performing LD based pruning...
Pruned independent SNPs on chromosome 1
Pruned independent SNPs on chromosome 2
Pruned independent SNPs on chromosome 3
Pruned independent SNPs on chromosome 4
Pruned independent SNPs on chromosome 5
Pruned independent SNPs on chromosome 6
Pruned independent SNPs on chromosome 7
Pruned independent SNPs on chromosome 8
Pruned independent SNPs on chromosome 9
Pruned independent SNPs on chromosome 10
Pruned independent SNPs on chromosome 11
Pruned independent SNPs on chromosome 12
Pruned independent SNPs on chromosome 13
Pruned independent SNPs on chromosome 14
Pruned independent SNPs on chromosome 15
Pruned independent SNPs on chromosome 16
Pruned independent SNPs on chromosome 17
Pruned independent SNPs on chromosome 18
Pruned independent SNPs on chromosome 19
Pruned independent SNPs on chromosome 20
Pruned independent SNPs on chromosome 21
Pruned independent SNPs on chromosome 22
Pruned independent SNPs on chromosome 23
Pruned independent SNPs on chromosome 25
Pruned independent SNPs on chromosome 26
Saved pruned SNP file as jag.prune.in

Drawing 20 x 2205 nEff SNPs from a pool of 63451 SNPs located within genes

Saved random draws on number of effective number of SNPS as
jag.draws_neff_genic.set.annot

Finished analysis: Wed Jul 11 16:18:26 2012
```

As a result from this command, a list of pruned SNPs will be saved as jag.prune.in. This file contains the independent SNPs from which the random sets of SNPs are drawn. This pruning as shown above is only performed in case no **jag.prune.in** exists. In cases this file does exist, JAG directly starts with generating the random sets.

Subsequently, a list with the independent SNPs selected for each draw is saved as **jag.draws_neff_genic.set.annot**. The tab-delimited file contains three columns denoting SNPID, GeneID and Draw_number and should be used for conducting the self-contained test over the draws.

```
rs285689              64377     Draw_1
rs7059099             1756      Draw_1
rs4281788             5573      Draw_1
rs4664453             2191      Draw_1
rs13355153            256987    Draw_1
rs10022693            10563     Draw_1
rs2130910             2195      Draw_1
rs3135093             57732     Draw_1
rs7858284             23189     Draw_1
…
```

In case that the desired number of draws times the number of effective SNPs of the given gene-set exceeds the available pool of independent SNPs, a warning is given by JAG and the analysis terminated.

As mentioned earlier, JAG also supplies the possibility to draw random sets of SNPs based on the number of independent SNPs outside genes by using option **--draw_neff_intergenic** or in- and outside genes by using option **--draw_neff_all**. Below examples are given for how to draw from these other pools of independent SNPs:

Random sets matched for the number of independent SNPs outside genes:

```
jag --ndraw 20 --bfile example --set jag.set.annot --draw_neff_intergenic hsa04080 --
neff_calc jag.P1.empp --pool jag.allgenes.annot
```

Random sets on number of independent SNPs in- and outside genes:

```
jag --ndraw 20 --bfile example --set jag.set.annot --draw_neff_all hsa04080 --
neff_calc jag.P1.empp --pool jag.allgenes.annot
```

TIP: It is possible to use your own set of pruned SNPs to draw random sets of independent SNPs, as long as the **.prune.in** file has the same format as described in this PLINK manual and has a prefix that equals your **–out** prefix.

3.4.2 Self-contained test on random sets

This step makes use of the self-contained test as previously described, but now on the randomly drawn sets. Compared to the self-contained analysis on real gene-sets, the files resulting from the self-contained test over the random sets will have an extra prefix referring to the drawing method, i.e. draws_ngenes, draws_neff_genic, draws_neff_nongenic, draws_neff_all.

Running the self-contained test on randomly drawn sets of genes is invoked by typing:

```
jag --bfile example --set jag.draws_ngenes.set.annot --perm 10
```

Which gives an output similar to when conducting a self-contained test on the original gene-set(s) as described in Chapter 3.3.1.

With a similar command the self-contained test can be conducted on random gene-sets, which are matched for number of effective SNPs.

Self-contained test for gene-sets matched for independent genic SNPs

```
jag --bfile example --set jag.draws_neff_genic.set.annot --perm 10
```

Self-contained test for gene-sets matched for independent SNPs in- and outside genic

```
jag --bfile example --set jag.draws_neff_all.set.annot --perm 10
```

Self-contained test for gene-sets matched for independent SNPs outside genes

```
jag --bfile example --set jag.draws_neff_nongenic.set.annot --perm 10
```

3.4.3 Calculate competitive P-value

When the self-contained test is performed over the randomly drawn gene-sets, the final step is to calculate the competitive P-value. In this simple test the self-contained empirical P-values of the randomly generated sets are compared with that of the original gene-set (hsa04080, in this example), which is an argument of the **--gene_set** option. The option **--orig_empp** is used to specify the file with self contained empirical P-values from the original gene-sets and **--control_empp** is used to specify the file with self contained empirical P-values from the control gene-sets.

```
jag --orig_empp jag.merged.P1.empp --control_empp jag.draws_neff_genic.P1.empp --
gene_set hsa04080
```

The following screen output is generated:

```
    ############################################################
    #                                                          #
    #     JAG  - Joint Association of Genetic variants - V1.1  #
    #            2012,  Esther Lips & Danielle Posthuma        #
    #              GNU General Public License, v2              #
    #                                                          #
    #                Complex Trait Genetics Lab                #
    #               http://ctglab.nl/software/jag/             #
    #                                                          #
    ############################################################

Save log file as [jag.log]

Analysis started: Thu Jul 12 11:03:03 2012

Used options:
     --orig_empp jag.empp.merged.P1.out
     --control_empp jag.draws_neff_genic.empp.P1.out
     --gene_set hsa04080

Empirical P-value for competitive test for gene-set hsa04080 = 0.12
```

```
Finished analysis: Thu Jul 12 11:03:03 2012
```

The competitive empirical P-value using matched control gene-sets drawn from all genic SNPs is 0.12, indicating that hsa04080 is not more strongly associated to the trait than any other set of randomly drawn genic SNPs.

Chapter 4

Frequently Asked Questions

Can I use my own SNP mapping?

Yes, you can. In that case you are also not restricted to use Entrez Gene identifiers. But note that we do not give support on making your own snp2gene mapping.

Can I use my own set of independent SNPs when drawing random sets of genes or SNPs?

Yes, you can as longs as the file of independent SNPs:

I) has the same format as a pruned set of SNPs is created in PLINK and

II) is named exactly as what you give with the **`--out`** option in your command line.

Chapter 5

Reference Tables

5.1 Options

Option	[Shortcut]	argument/default	Description
Basic output			
--out	[-o]	jag	Specify output filename
Annotate snps to genes			
--snp2gene		geneset_file	Specify file with genesets
--up		kb	Upstream window
--down		kb	Downstream window
--gene_loc		gene_file	Specify file with gene location
--snp_loc		snp_file	Specify file with SNPs
Self-contained test			
--set	[-g]	snp_set_file	Specify file with SNP list of genesets
--perm	[-m]	N	Number of permutations to run
--no_emp			Self-contained test is not performed
--gene_based			Use gene based test instead of gene-set test
--no_graph			Disable generation of R plots
--seed		seed	to specify a fixed seed
--merge		prefix	Prefix of files to merge
Supported PLINK options:			
--bfile	[-b]	jag	Specify .bed .bim. and .fam file
--pheno	[-p]	phenotype file	
--logistic			Test for disease traits
--linear			Test for quantitative traits
--covar		covarfile	Specify covariate file
--adjust			Use adjusted P-values for stratification
Random draws on ngenes			
--ndraw		N	Number of sets to be drawn
--group	[-g]	set_file	Specify file with SNP list of groups
--snp2gene		snp_list	List with snps for all genes
--draw_ngenes		set_name	Select draw on number of genes from group
Random draws on effective SNPs			
--bfile		jag	Specify .bed .bim. and .fam file
--ndraw		N	number of draws
--set		set_file	
--pool		snp_list	List with snps to draw from
--neff_calc		empp_file	Specify file in which # of effective SNPs is given
--draw_neff_all		set_name	
--draw_neff_intergenic		set_name	
--draw_neff_genic		set_name	
--include			include snps from orig set
Competitive test			
--gene_group		set_name	Specify name of set to run competitive test for
--orig_empp		orig_empp_file	Specify file with self-contained emp pvalue for original gen-set

--control_empp	control_empp_file	Specify file with self-contained emp pvalue for random draws

Other

--help	[-h]	Display list of options
--verbose	[-v]	Display PLINK output in screen/logfile

5.2 Output files

Filename	**Main associated command(s)**	**Description**
jag.log		Log file (always generated)
jag.results.P1.assoc	--assoc	Basic case/control association results (from PLINK)
jag.results.P1.qassoc	--assoc	Basic quantitative association results from PLINK
jag.results.P1.assoc.adjusted	--adjust	Adjusted P-values (from PLINK)
jag.results.P1.assoc.adjusted	--adjust	Adjusted P-values (from PLINK)
jag.results.P1.assoc.linear	--linear	Linear association P-values (from PLINK)
jag.results.P1.assoc.logistic	--logistic	Logistic association P-values (from PLINK)
jag.P1.sumlog	--perm	Test statistics for real data
jag.[unique_key].P1.perm	--perm	Test statistics for permuted data
jag.P1.empp	--perm	Self contained test results
jag.assoc_qq_plot.pdf	--perm	QQ plot of assocation results for all SNPs
jag.qq-plot_all_sets.P1.pdf	--perm	QQ plots of assocation results per gene set/gene
jag.distribution_sumlogs.P1.pdf	--perm	Distribution plot for test statistic of real and permuted data
jag.prune.in	--draw_neff_all --draw_neff_genic --draw_neff_intergenic	List of genic and non-genic independent SNPs
jag.merged.P1.perm	--merge	File with merged test-statistics
jag.draws_ngenes.set.annot	--draw_ngenes	Draws with draws on number of genes
jag.draws_neff_genic.set.annot	--draw_nef_genic	Draws on independent genic SNPs
jag.draws_neff_nongenic.set.annot	--draw_nongenic	Draws on independent nongenic SNPs
jag.draws_neff_all.set.annot	--draw_neff_all	Draws on all independent SNPs

Chapter 6

References

Lips, E.S., Cornelisse, L.N., Toonen, R.F., Min, J.L., Hultman, C.M., Holmans, P.A., O'Donovan, M.C., Purcell, S.M., Smit, A.B., Verhage, M., Sullivan, P.F., Visscher, P.M. and Posthuma, D. (2011) **Functional gene group analysis identifies synaptic gene groups as risk factor for schizophrenia**, *Mol Psychiatry*.

Price, A.L., Patterson, N.J., Plenge, R.M., Weinblatt, M.E., Shadick, N.A. and Reich, D. (2006) **Principal components analysis corrects for stratification in genome-wide association studies**, *Nat Genet*, **38**, 904-909.

Purcell, S., Neale, B., Todd-Brown, K., Thomas, L., Ferreira, M.A., Bender, D., Maller, J., Sklar, P., de Bakker, P.I., Daly, M.J. and Sham, P.C. (2007) **PLINK: a tool set for whole-genome association and population-based linkage analyses**, *Am J Hum Genet*, **81**, 559-575.

Purcell, S.M., Wray, N.R., Stone, J.L., Visscher, P.M., O'Donovan, M.C., Sullivan, P.F. and Sklar, P. (2009) **Common polygenic variation contributes to risk of schizophrenia and bipolar disorder**, *Nature*, **460**, 748-752.

Ramanan, V.K., Shen, L., Moore, J.H. and Saykin, A.J. (2012) **Pathway analysis of genomic data: concepts, methods, and prospects for future development**, *Trends Genet*, **28**, 323-332.