

GeneSetScan (Version 0.02)

Scanning gene sets for associations in genome-wide association
studies

Daniel Schaid and Jason Sinnwell
Mayo Clinic
Division of Biomedical Statistics and Informatics
Rochester MN USA

July 2, 2014

Contents

1	Introduction	3
1.1	Description	3
1.2	Package Contents	3
1.3	Installation	3
1.4	System Requirements	4
2	Reference Files	4
3	SNP Scores	5
3.1	Definition	5
3.2	Example Scripts	7
4	Program Options	7
4.1	Complete List	7
4.2	Missing Values	9
4.3	Simulations	9
4.4	Output Files	9
4.5	Mapping Distance	9
5	Parameter Files	9
5.1	SNP-level	9
5.2	Gene-level	10
5.3	Gene Ontology	10
5.3.1	go-type	11
5.3.2	go-gene and go-edges	11
5.3.3	max-size	11
5.3.4	gene-set-level	11
5.4	KEGG	12
6	Results	13
6.1	Log File	13
6.2	HTML and CSV	13
6.2.1	SNP and Gene	13
6.2.2	Gene Sets	14
6.3	Map File	16
A	SNP to Gene Mapping	16
B	Gene Ontology Graph Edges	17
C	License	17

1 Introduction

1.1 Description

The GeneSetScan software offers a general approach to scan genome-wide SNP data for gene-set association analyses. The test statistic for a gene set is based on score statistics for generalized linear models, and takes advantage of the directed acyclic graph structure of the Gene Ontology to create gene-sets. The method can use other gene-set structures, such as the Kyoto Encyclopedia of Genes and Genomes (KEGG), or even user-defined sets. Our approach combines SNPs into genes, and genes into gene-sets, but assures that positive and negative effects on a trait do not cancel. To control for multiple testing of many gene-sets, we use an efficient computational strategy that accounts for linkage disequilibrium and correlations among genes and gene sets, and provides accurate step-down adjusted p-values for each gene-set.

1.2 Package Contents

The package contains the pre-compiled executable and reference data files to define gene sets. The most important step for the user is to create a file of SNP scores, and we provide scripts and documentation to guide you through that process. The package has the following directory structure, with a brief description of the sub-directories in *italics*:

- *INSTALL.txt*
- *README*
- **bin/** *binary executable*
- **refdata/** *reference files for mapping SNPs to gene sets, described in section 2*
- **doc/** *license and manual*
- **example/** *Example parameter files and example script to run the program*
- **SNPScores/** *scripts to create SNP scores, described in section 3*

1.3 Installation

We provide the pre-compiled GeneSetScan executable compiled for 64-bit Linux CentOS 4.3, which should work on most 64-bit Linux distributions. We can compile for other systems upon request. Make sure these steps are taken before running the program.

1. Copy the binary executable from GSS/bin/ to a location where it can be found by your shell commands. You could use the full path to GSS/bin/GeneSetScan to call the program, or copy it to a globally-available bin directory on your system.
2. Uncompress all reference files and place in GSS/data. The reference files were compressed using gzip on Linux. Use either gunzip or another de-compression program. The parameter files will tell the program where to find these files.

1.4 System Requirements

Memory and disk space usage are two issues to consider before running this program. First, we define "memory" as system memory needed while the program is running from start to finish, and we define "disk space" as the hard-disk space needed to store any file on your system.

We have tested GeneSetScan on various datasets that each were genotyped on 550K SNP chips, ranging from 1000 to 2500 subjects. The disk space needed for the complete SNP score file (see section 3) in those studies can exceed 10GB, and the memory needed to run the program can exceed 12GB. Because the machine language commands for programs compiled on 32-bit machines cannot index more than 4GB of memory, GeneSetScan should only be run on 64-bit machines with sufficient memory available during run-time. Table 1 contains a summary of the approximate disk and memory usage on analyses we have run.

N_{subj}	N_{SNP}	Disk	Memory
1000	550K	8GB	4GB
2500	550K	13GB	14GB

Table 1: Approximate Disk and Memory Usage

We estimate that the program will run with a reasonable amount of memory and consume moderate disk space for SNP chips up to 1 million SNPs. It will require more resource management to run this method on chips or imputed genotype sets that provide more coverage than 1 million SNPs, so we currently do not recommend using such large SNP sets.

Run-time is dictated by the number of simulations, with only a small portion attributable to other tasks. Run-time for 1000 simulations is about 1 hour, and there is a linear increase in time for more simulations with no additional memory cost.

2 Reference Files

We provide all files necessary for GeneSetScan to map SNPs by rsid to gene sets. We provide the following set of mapping reference files:

- **edges.csv** file defining the GO directed acyclic graph edges (release date: March 2011)
- **gene2go.human** list of genes mapping to GO terms (release date: March 2011)
- **hsa_pathway.list** list of genes mapping to KEGG human pathways (release date: March 2011)
- **gene_snp.B37.coding.dat** maps SNPs to genes using rsids of SNPs from HapMap reference panel updated to NCBI Build 37 positions, mapped to Build 37 gene start/stop position (HapMap-23a and Build GRCh37p2)

More details on how we created the above mapping files can be found in Appendices A and B.

3 SNP Scores

3.1 Definition

The GeneSetScan program requires an input SNP score file. This file contains the contribution of each subject and each SNP to the score statistic that tests associations of SNPs with a trait. The rows are for SNPs and the columns are for subjects. The first column is the SNP rsid, followed by the columns that are the scores per subject per SNP from a model fit, $S_{i,j} = (Y_{i,j} - \hat{Y}_{:,j}) * (G_{i,j} - \bar{G}_{i,.})$, where

- $\hat{Y}_{:,j}$ is the fitted value for the subject j from a logistic model that may or may not include adjusting covariates
- $G_{i,j}$ is the SNP dosage at SNP i for subject j, typically the minor allele count
- $\bar{G}_{i,.}$ is the mean SNP dosage over all subjects

The general format for the SNP score file is as follows:

```
rsid score1 score2 score3 score4 ...
```

Where an example looks like this:

```
rs100 0.152 -0.330 0.104 -0.225 ...
rs105 -999.0 0.951 -0.312 0.152 ...
rs222 0.443 0.051 -0.875 -0.342 ...
...
```

Missing values need to be numeric, yet different from the observed scores such that we can tell GeneSetScan to treat them as missings; setting missing values to -999.0 will work most of the time.

We provide a schema of the scripts we provide to create the SNP-Scores in figure 1, and the scripts themselves are described below.

SNP-Scores Scripts

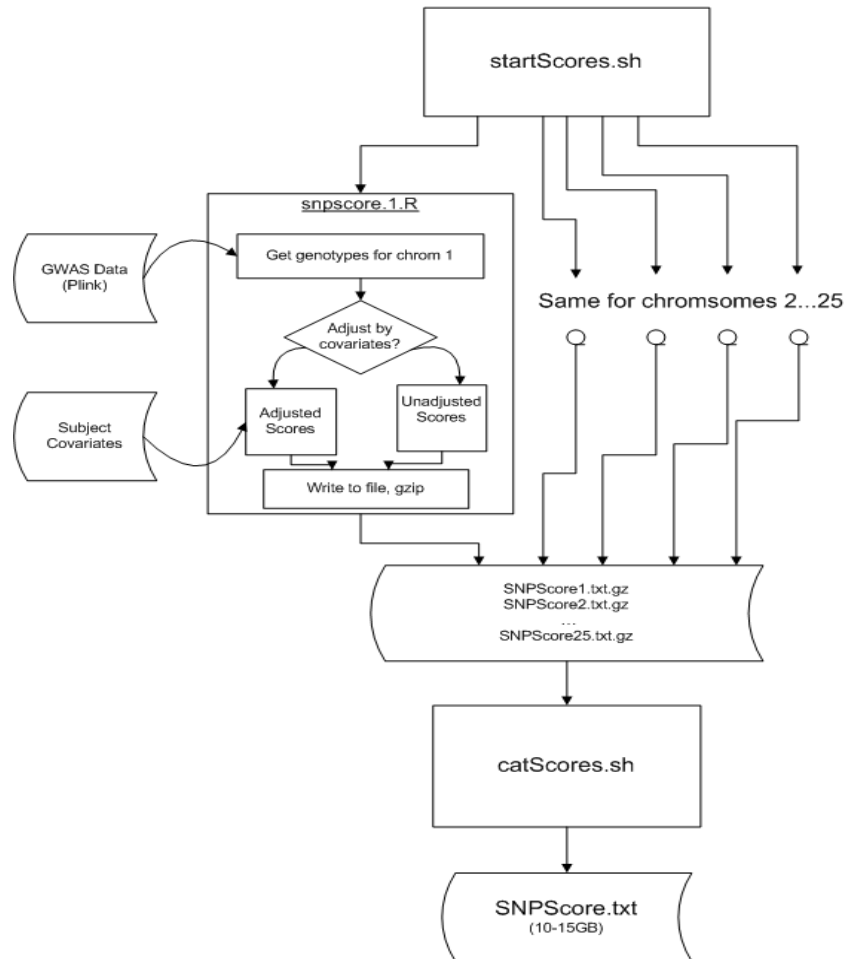


Figure 1: SNP-Score schema

3.2 Example Scripts

Users are free to use any program to create the SNP score file, but we provide some general scripts for computing SNP scores by chromosome using R[5], PLINK [6], and some shell scripting. The scripts we provide are set up to run a logistic regression in R for a binary case-control status.

- **unadjustedScores.R** an R script to compute SNP scores for a single chromosome, not adjusted for covariates. The script assumes that *plink* can be called via a system command.
- **adjustedScores.R** an R script to compute SNP scores for a single chromosome, adjusted for covariates. The script assumes that *plink* can be called via a system command, and requires some editing for the number of covariates to adjust for in the model.
- **startScores.sh** An example shell script that creates the SNP scores in separate R batch jobs for each chromosome.
- **catScores.sh** An example shell script that concatenates the SNP scores for each chromosome into one large SNP-score file. The individual chromosome SNP score files are large and can be re-created within a few hours, so we recommend deleting them once the large score file is created.

4 Program Options

4.1 Complete List

Program options can be viewed with the program help menu, invoked like this:

```
> GeneSetScan -h
```

OR

```
> GeneSetScan --help
```

Each option has either a short name preceded by one dash (“-”), or a long name preceded by two dashes, followed by a space and the program option (except flags like `--version/-v` and `--help/-h`). Below is the complete list of options, listed with the default and brief description.

```
Usage: -short, --long options with default values
       brief description

-h, --help
       display the help menu
-ge goEdges.dat, --go-edges goEdges.dat
       file of GO edges
-gs geneSNP.dat, --gene-snp geneSNP.dat
       file of gene-snp mapping
```

```

-gg goGene.dat, --go-gene goGene.dat
    file of go-gene mapping
-kg keggGene.dat, --kegg-gene keggGene.dat
    file of kegg-gene mapping
-sc score.dat, --score score.dat
    file of scores
-r zero, --replace zero
    replace missing scores(mean, zero)
-mv -999, --missval -999
    missing value code for scores
-m GO, --method GO
    method (GENE, GO, KEGG, SNP)
-gl sqrtmean, --gene-level sqrtmean
    gene-level scoring (max, mean, sqrtmean).
    Caution: max, mean are experimental
-gm nor, --gene-moments nor
    gene-level moments (nor, sim). Caution: sim is experimental
-gsl wtdmean, --gene-set-level wtdmean
    gene-set-level scoring (maxZ, meanZ, wtdmean).
    Caution: maxZ, meanZ are experimental
-gt all, --go-type all
    type of GO DAG (all, biological_process,
    cellular_component, molecular_function)
-d 50000, --dist 50000
    max absolute dist from SNP to gene start/stop, in bp distance
-ms 30, --max-size 30
    maximum no. genes in a set. Caution: Compute time increases with max-size
-ns 1000, --nsim 1000
    number of simualtions
-s -9, --seed -9
    integer for random number seed (set by clock if <= 0 or not specified)
-html output.html, --html-output output.html
    ouput file in html format
-csv output.csv, --csv-output output.csv
    ouput file in csv format
-mw, --map-write
    turn on writing of map output
-mf output.map, --map-output output.map
    output map file name
-par <file name here>, --parfile <file name here>
    file of options, one per line.
-v, --version
    0.01

```

All program options can be called on either the command line or a parameter file. The large number

of required options are managed easiest with a parameter file that is then input to GeneSetScan.

4.2 Missing Values

The `--missval` option tells GeneSetScan how to find missing input scores, and `--replace` tells it how to replace the missing value, namely, with zero or the mean score at that SNP.

4.3 Simulations

The `--nsim` option tells GeneSetScan how many simulations to perform, and `--seed` tells it the seed to start with so that simulations can be reproduced in separate runs.

4.4 Output Files

Output files are automatically written to default file names by the base name “output”, and the appropriate suffix. We recommend using `--html-output` and `--csv-output` to specify your own file names for the html and csv output files. A third output “mapping” file is optional and only applies to gene-set-level analyses. The `--map-write` option flag tells GeneSetScan to write the optional map file with information on how SNPs and genes map to gene-sets within the analysis. The `--map-output` option tells GeneSetScan the name you want for it. More details on the output files can be found in section 6.

4.5 Mapping Distance

The `--dist` option specifies how far outside the gene start/stop positions to allow SNPs to map to genes. The default setting is 50,000 base pairs, and GeneSetScan will allow `--dist` to be any number between 0 and 50,000 because the reference file that comes with the package was created with a maximum of 50,000 base pairs. If you need a larger range than 50,000 base pairs, we can create a different reference file upon request. More information on the SNPs mapping to genes can be found in Appendix A

5 Parameter Files

We discuss example parameter files in ascending order of set size: SNP, Gene, GO-Namespace, GO-All, and KEGG, which are given as they appear in the GSS/example directory. We highlight the arguments that are needed as the sets get more complex.

As described in section 4, the program options have a long and short name. All our examples show the long name, which are preceded by two dashes(-), followed by any number of white spaces, then the option setting. The options in all parameter files can be in any order, so we place the ones that change the least at the top, allowing us to focus on the ones that vary at the bottom.

5.1 SNP-level

SNP-based tests are commonly available in other programs, but we make it available because it is convenient to have a complete set of SNP, Gene, and Gene-Set association tests available in one program. Furthermore, the adjusted p-values are an advantage over other programs.

We see in the list below the first four options control the missing values and the simulations, which should not change much unless you want a different seed or more precise p-values. The score file is also required and will be used for all analyses on the same dataset. Next, we specify the names of the html and csv files specific to the analysis type so we can keep results separate when running multiple analyses. Finally, the option for method is set to SNP to tell GeneSetScan to run the SNP tests.

```
--replace zero
--missval -999.0
--nsim 1000
--seed 12887321
--score ../SNPScores/SNPScore.txt
--html-output ./results/snp.html
--csv-output ./results/snp.csv
--method SNP
```

5.2 Gene-level

For gene-level tests, we change method to *GENE*, change the names of the output files, and add three more options.

The first option added for gene-level analysis is *--gene-snp*, which tells GeneSetScan where to find the SNP to gene mapping file in refdata. The other added options tell the program how to score each gene. For the logistic model scoring of each gene, we set the *--gene-level* option to *sqrtmean* for the square-root of the mean of the SNP scores combined over a gene, for which we estimate the moments for the gene scores assuming a multivariate normal distribution, specified by *--gene-moments nor*.

```
--replace zero
--missval -999.0
--nsim 1000
--seed 12887321
--score ../SNPScores/SNPScore.txt
--html-output ./results/gene.html
--csv-output ./results/gene.csv
--method GENE
--gene-snp ../refdata/gene_snp.dat
--gene-level sqrtmean
--gene-moments nor
```

5.3 Gene Ontology

For the Gene Ontology gene sets we keep most of the options from the gene analysis setting with updates to output file names. We need to change *--method* to *GO* and specify which GO namespace to use. Then we add options to specify names of two more reference files, give GeneSetScan a maximum set size for gene sets, and finally, give instructions on scoring genes for gene sets. We describe these options below.

5.3.1 go-type

The `--go-type` option tells which of the three Gene Ontology namespaces to use: Biological Process, Molecular Function, and Cellular Component. Table 2 gives the GO Namespaces and their `--go-type` string that is recognized in the program, which includes ALL for terms from all three namespaces.

Namespace	--go-type
Biological Process	biological-process
Molecular Function	molecular-function
Cellular Component	cellular-component
ALL	all

Table 2: GO Namespaces `--go-type` names

5.3.2 go-gene and go-edges

In addition to the Gene-SNP file needed in the gene analysis above, we have to tell GeneSetScan where to find two GO-specific data files that we provide in `refdata`. The first is the `--go-gene` file that links Entrez Gene IDs to GO terms, which we created by subsetting to only those entries for the human species. The second GO-specific data file is the `--go-edges` file, which we pre-processed to define the GO nodes and edges that connect them. More information on the GO structure can be found in Appendix B.

5.3.3 max-size

Each namespace is a directed acyclic graph with the lowest level terms being the most specific and higher-level terms more general. At the higher levels the number of genes per set grows such that the tests combine information over so many genes that it is unlikely to have a meaningful result. Additionally, there is a computational cost for larger gene sets. For these reasons, the user can limit analyses to sets with no more than the number of genes specified by `max-size`. For all of the Gene Ontology gene sets, we recommend a `--max-size` of 100.

5.3.4 gene-set-level

We carry forward the gene-level scoring from the gene analysis, but we additionally need to tell GeneSetScan how to score genes within the gene sets. We have implemented the weighted mean, given by `--gene-set-level wtdmean`, of the gene-specific scores, where the weights are the inverse of the variance of the gene score. [[DAN PLEASE HELP CORRECT]].

```
--replace zero
--missval -999.0
--nsim 1000
--seed 12887321
--score ../SNPScores/SNPScore.txt
--html-output ./results/go.bp.html
```

```

--csv-output ./results/go.bp.csv
--gene-snp ../refdata/gene_snp.dat
--go-gene ../refdata/gene2go.human
--go-edges ../refdata/edges.csv
--method GO
--go-type biological-process
--max-size 100
--gene-level sqrtmean
--gene-moments nor
--gene-set-level wtdmean

```

To run analyses for the other GO namespaces, just change the *--go-type* option to one of the items listed in Table 2

5.4 KEGG

For analysis of gene sets defined by the Kyoto Encyclopedia of Genes and Genomes [3], we maintain most of the same options as described above with updates to the output file names to reflect the analysis method. The method option is changed to *KEGG*, and we only need to specify one file for the genes mapping to KEGG pathways by the *--kegg-gene* option for the refdata file called *hsa_pathway.list*

We change *--max-size* to 5000 because some KEGG sets are large. However, there are fewer KEGG gene sets overall, so we are able to capture all of them without too much computational cost. Last, we use the same gene-level and gene-set-level scoring as above.

```

--replace zero
--missval -999.0
--nsim 1000
--max-size 5000
--seed 12887321
--score ../SNPScores/SNPscore.txt
--html-output ./results/kegg.html
--csv-output ./results/kegg.csv
--method KEGG
--kegg-gene ../refdata/hsa_pathway.list
--gene-snp ../refdata/gene_snp.dat
--max-size 5000
--gene-level sqrtmean
--gene-moments nor
--gene-set-level wtdmean

```

The screenshot shows a web browser window with the title 'Results'. The main content is a section titled 'Gene-Level Analyses' which contains a table with the following data:

gene ID	no. SNPs	z-score	SNP-rank	p-value
200	61	5.359	7.284e+04	0.342
201	22	5.015	4.853e+04	0.626
202	29	5.013	9.38e+04	0.627
300	43	-2.555	2.882e+05	0.999

At the bottom of the window, there is a status bar that says 'Done'.

Figure 2: Gene Output

6 Results

6.1 Log File

All calls to the program generate the log information to the screen, which can be directed to a file with `>`, which we recommend because the log information is verbose and provides useful summaries of the number of SNPs, genes, and gene-sets that are used.

```
> GeneSetScan --par gene.par > results/gene.out
```

6.2 HTML and CSV

The GeneSetScan program writes the primary results to both a comma-separated and html file. The format of these files varies by the analysis method. As you will see, it is much more convenient to view and use the html file, but we also provide the csv files in standard formats so they can be parsed by other tools.

6.2.1 SNP and Gene

The SNP and Gene results are similar, in that there is only one line of output per test, so the csv file looks like the following frame, and the html file looks like Figure 2.

geneID,	num_snps,	score,	mean,	sd,	zscore,	mn_snp_rank,	pval
200,	61,	2.39,	0.964,	0.267,	5.36,	7.28e+04,	0.342
201,	22,	2.63,	0.943,	0.337,	5.02,	4.85e+04,	0.626
202,	29,	2.21,	0.969,	0.248,	5.01,	9.38e+04,	0.627
300,	43,	0.503,	0.982,	0.188,	-2.55,	2.88e+05,	0.999

6.2.2 Gene Sets

The output files for gene set results provide more information and thus have a more complex format. Because each gene set can have multiple set ids and multiple genes mapping to them, we provide all of the information in 3 columns of the csv file. The first column is the *setID*, which serves as an index to identify the lines for each set, starting at 1 with the set with the lowest p-value. The second column, *variable* is a text field saying what information is provided in the last column, labeled *value*. Figure 3 shows the html file with similar information, but condensed to a table with a row for each gene set, and active links to both the genes and gene sets.

```
setID, variable, value
1, num_genes, 3
1, score , 1.472
1, mean , 0.9367
1, sd , 0.1455
1, zscore , 3.68
1, mn_gene_rank , 7560
1, pval , 0.856
1, geneID, 101
1, geneID, 102
1, geneID, 103
1, GOID, GO:0010001
1, GOID, GO:0010002
2, num_genes, 4
2, score , 1.229
2, mean , 0.9509
2, sd , 0.08108
2, zscore , 3.432
2, mn_gene_rank , 7560
2, pval , 0.978
2, geneID, 1001
2, geneID, 1002
2, geneID, 1003
2, geneID, 1004
2, GOID, GO:00100003
```

Gene-Set-Scan Results

Gene-Set Analyses: GO-biological_process

gene set	no. genes	score	mean	sd	z-score	mn_gene-rank	p-value	genes in set	GO terms with set
1	3	1.472	0.9367	0.1455	3.68	7560	0.856	<ul style="list-style-type: none"> • 101 • 102 • 103 	<ul style="list-style-type: none"> • GO:0010001 • GO:0010002
2	4	1.229	0.9509	0.08108	3.432	7560	0.978	<ul style="list-style-type: none"> • 1001 • 1002 • 1003 • 1004 	<ul style="list-style-type: none"> • GO:0010003

Done

Figure 3: GO:Biological Process Output

6.3 Map File

An optional output file is a map file, which is only available for gene set analyses. The example below shows how the setID is the same as the setID in the csv file, and it gives which SNPs are mapped to the genes that go into the gene sets.

```
setID, gene, snp
1, 101, rs100001
1, 101, rs100002
1, 101, rs100003
1, 101, rs100004
1, 101, rs100005
1, 102, rs20001
1, 102, rs20002
1, 102, rs20003
1, 103, rs3001
1, 103, rs3002
1, 103, rs3003
2, 1001, rs4000001
2, 1001, rs4000002
2 1001, rs4000003
...
```

Appendix

A SNP to Gene Mapping

SNPs that are part of HapMap release 23a mapped to genes by the NCBI human genome build 37 positions. To create this file, we used a map file for HapMap panel available from Plink here:

<http://pngu.mgh.harvard.edu/~purcell/plink/res.shtml#hapmap>,

which we updated the SNP positions to NCBI human genome build 37.1 and mapped them to genes if they were within +/-50kb of the start/stop of any protein-coding gene. The gene start/stop positions are provided by seq_gene.md from this site:

ftp://ftp.ncbi.nih.gov/genomes/MapView/Homo_sapiens/sequence/BUILD.37.2/initial_release/seq_gene.md

where we use only the start/stop positions of non-pseudo genes for the Genome Reference Consortium (GRCh37.p2-Primary Assembly). The protein-coding status of genes is provided in Homo_sapiens.gene_info, from this site:

ftp://ftp.ncbi.nih.gov/gene/DATA/GENE_INFO/Mammalia/Homoe_sapiens.gene_info

phone: 507-284-0639
fax: 507-284-9542
email: schaid@mayo.edu

References

- [1] Schaid DJ, Sinnwell JP, Jenkins GD, McDonnell SK, Ingle JN, Kubo M, Goss PE, Costantino JP, Wickerham DL, Weinshilboum RM. **Using the gene ontology to scan multilevel gene sets for associations in genome wide association studies.** *Genet Epidemiol* 2011 Dec 7
- [2] The Gene Ontology <<http://www.geneontology.org>>
- [3] KEGG: Kyoto Encyclopedia of Genes and Genomes <<http://www.genom.jp/kegg>>
- [4] Lin DY. 2005. **An Efficient Monte Carlo approach to assessing statistical significance in genomic studies.** *Bioinformatics*, 21:781-787.
- [5] The R Project <<http://www.r-project.org>>
- [6] Purcell, Shaun. **Plink (version 1.07).** <<http://pngu.mgh.harvard.edu/purcell/plink/>>