MANTEL: a tool for meta-analysis of genome-wide association studies

Authors:

Sara Pulit, Jessica van Setten, Paul de Bakker

Affiliations:

Division of Genetics, Brigham and Women's Hospital
Department of Medicine, Harvard Medical School
Program in Medical and Population Genetics, Broad Institute of Harvard and MIT

Contact:

Sara Pulit, pulit@broadinstitute.org
Paul de Bakker, debakker@broadinstitute.org

Citation:

P.I.W. de Bakker, M.A.R. Ferreira, X. Jia, B.M. Neale, S. Raychaudhuri, and B.F. Voight. Practical aspects of impuation-driven meta-analysis of genome-wide association studies. Human Molecular Genetics, 2008 Oct 15; 17; pp. R122-8

Summary:

Meta-analysis has allowed researchers to assemble large amounts of data across cohorts to better understand the genetics of common complex human traits. Below is a description of how to use the `MANTEL.pl` script to run a meta-analysis across multiple GWAS data sets. The process takes as input raw GWAS result data provided by multiple sources, and ends with a completed meta-analysis, outputting a list of independent, genome-wide significant SNPs. The procedure will address many of the practical issues that inherently arise from combining data across different studies.

PART I

--- RUNNING THE META-ANALYSIS ---

1. Data reformatting with `run_mantel.csh`

Data will most likely be received in a variety of formats. Reformatting each dataset so that the formats are identical is necessary to run `MANTEL.pl`.

The first section of `run_mantel.csh` reorganizes each raw data file into the following format (note that the exact handling of the raw data will change depending on the format in which you receive the data):

Col. 1 – SNP
Col. 2 – Chromosome
Col. 3 – Position
Col. 4 – Beta
Col. 5 – Standard error
Col. 6 – P-value
Col. 7 – Coded allele
Col. 8 – Allele 2
Col. 9 – Coded allele frequency
Col. 10 – Imputation quality score

As outlined in de Bakker, et al. (see citation), it is important to know the build number of each dataset to ensure concordance across datasets. When the data is reformatted, `run_mantel.csh` also automatically removes any SNPs for which SE is extremely large, association results are missing, allele frequency is 0 or 1, and/or imputation quality is above 1.1.

This step will produce a "`.dat`" file for each cohort.

2. Data QC with `run_mantel.csh`

Though cohort data has presumably been QC'd before a GWAS is run (to correct for sample or SNP missingness, relatedness, population stratification, and other possible confounders), data must also be QC'd before being used for meta-analysis.

After creating ".dat" files per cohort, `run_mantel.csh` creates 4 plots per dataset. The plots are:

1. `cohort.qqplot.pdf` – a standard QQ plot displaying expected vs. observed distribution of the data. The plot will also display lambda for this cohort.

2. `cohort.maf_qqplot.pdf` – also a QQ plot, but stratified by minor allele frequency. Can be used to identify SNPs of particular allele frequency that may not be behaving as expected.
3. `cohort.imp_qual_qqplot.pdf` – QQ plot stratified by imputation quality. Can be used to identify SNPs of particular imputation quality that may not be behaving as expected.
4. `cohort.obsexp_hist.pdf` – a histogram of imputation quality scores. Can help identify if a large proportion of SNPs are of poor imputation quality and should be removed from the dataset.

If any of the plots indicate that additional QC should be performed on the data (e.g. removal of very poorly imputed SNPs or removal of SNPs with frequency < 1%) then the previous steps in `run_mantel.csh` (that formatted the `.dat` files) can be adjusted to include these additional QC steps.

3. Converting rsIDs

rsIDs are not static and therefore can change from build to build. `run_mantel.csh` includes a step to ensure that all rsIDs are all from the same build (e.g. dbsnp129). This requires having a file, e.g. `dbsnp129_rsIDs.txt`, formatted as such:

Col. 1 – rsIDs currently in your dataset(s)
Col. 2 – rsIDs from build129 that correspond to the IDs in your dataset(s)

A file like this can easily be obtained from a resource such as SNAP (http://www.broadinstitute.org/mpg/snap/) using the "MAP SNP IDs" tool.

4. Manhattan plots

Once the data has been fully QC'd and all of the rsIDs have been mapped to the same build, `run_mantel.csh` compiles a list of all the unique rsIDs across all of the cohorts being used in the meta-analysis and then creates a Manhattan plot for each of the cohorts.

5. Preparing data for MANTEL

To shorten the time in which MANTEL runs, particularly if a meta-analysis is being performed over several million SNPs, `run_mantel.csh` reorders the SNP data from each cohort (based on the generic SNP list created in step 4 and then breaks up all the data into files of 125,000 SNPs.

When this step is performed, you will see files in your directory named `cohort1.129.all.txt.000`, `cohort1.129.all.txt.001` …

`cohort1.129.all.txt.xxx` for each cohort. Similarly, the list of unique rsIDs in your meta-analysis will be split into files containing 125,000 SNPs each.

Once this step is complete, MANTEL can be run.

6. Running MANTEL

I. Input files and directories

Before executing MANTEL, you will need to make sure that the MANTEL script can access the following files (these are outlined in detail at the beginning of the MANTEL script itself and in an index at the end of this documentation):

a. Your data, as prepared above
b. A file containing study-specific parameters (`*.params`)
c. A directory that MANTEL will write to, e.g. OUT/
d. A directory containing PLINK-format HapMap files and a file with minor allele frequencies taken from HapMap (created using --freq in PLINK)
e. A dbSNP reference file that annotates the alleles of each SNP
f. A gene reference file (e.g. taken from RefSeq) containing chromosome, gene name, and the positions of the start and stop of each gene

II. Running MANTEL - brief overview

1. Mantel will first read in all of the reference data you have provided it - the list of the SNPs that are being meta-analyzed, the study parameters, the dbSNP file, HapMap allele frequencies, and the list of genes.
2. MANTEL then prepares the output file to which it will write all meta-analysis results.
3. Iterating over each SNP and across all included cohorts, MANTEL will run the meta-analysis. First, it will check for strandedness and allele frequency issues by:
    a. checking that the alleles of each SNP match up with the dbSNP reference.
    b. checking that the allele frequencies match up with the HapMap frequencies (especially useful for A/T and C/G SNPs for which strandedness can be ambiguous).
4. MANTEL will compute the inverse variance-weighted z-score for each SNP (as well as a sample-size-weighted z-score and a z-score for the random effects model if the argument --random-effects has been passed to the script) as well as corresponding beta, standard error, and p-values.

5. MANTEL then annotates each SNP with additional information about that site (this information appears as additional columns in the output file):
   a. DIRECTIONS: the sign of beta in each contributing cohort, annotated as "." if the SNP is missing from a particular cohort.
   b. GENES_1000KB: List of genes 1000 kb from the SNP
   c. NEAREST_GENE: the gene closest to the SNP
   d. FUNCTION: functional annotation, taken from dbSNP
   e. CAVEAT: additional information about the SNP, e.g. if the SNP is an A/T SNP with allele frequency between 0.35 and 0.65 (indicating strandedness ambiguity)
6. Finally, MANTEL will print a variety of summary metrics to the stdout file, including number of SNPs used in the meta-analysis and counts of strandedness and allele annotation corrections that were made.

8. Output files from MANTEL

For each subset of 125,000 SNPs meta-analyzed, MANTEL automatically produces three files.

1. The standard out file (`.stdout`) : this file contains a variety of useful summary metrics, including how many samples appear in each cohort, the total number of samples in the meta-analysis and the number of informative SNPs provided by each cohort. The very end of the file will also indicate whether the script was "successfully finished."
2. The standard error file (`.stderr`): this file annotates SNPs that are skipped/removed and why (e.g. the SNP has more than two alleles, the alleles of the SNP are inconsistent with the dbSNP reference).
3. The results (`.out`): This file contains the actual meta-analysis results. Each line is a single SNP.

9. Concatenating the data

Finally, when the meta-analysis has been completed across all subsets of 125,000 SNPs, `run_mantel.csh` will concatenate the results together into a single file. From here, it is convenient to either a) correct out of range p-values (i.e. p-values that are 0 because the corresponding chi-square values were too large for Perl to compute an exact p-value) or b) apply genomic control to the data.

10. Correcting out-of-range p-values

For large chi-square values, Perl will compute a p-value of 0, making it necessary to correct these p-values using an alternate tool.

`run_mantel.csh` checks for p-values of 0 (for all three models - fixed effects, random effects, and sample weighted), selects their corresponding z-scores from the meta-analysis output, and calls the script `meta.R` to have the p-values recomputed. `run_mantel.csh` then replace the p-values of 0 with the newly computed values.

11. QQ plots

Now that we have recomputed p-values, we can make QQ plots of the meta-analysis results. `run_mantel.csh` calls the `qqplot.R` script to create the following QQ plots:

1. `zfixed.qqplot.pdf`: QQ plot for fixed effects model
2. `zsqrtn.qqplot.pdf`: QQ plot for sample-weighted model
3. `zrandom.qqplot.pdf`: QQ plot for random effects model

12. Histograms of effective sample size and contributing studies

`run_mantel.csh` will now create two histograms.

1. `hist_neff.pdf`: a histogram that displays the effective sample size of SNPs used in the meta-analysis.
2. `hist_k.pdf`: a histogram that displays the number of contributing studies per SNP used in the meta-analysis.

Once the QQ plots and histograms have been created, `run_mantel.csh` will now create a Manhattan plot (`manhattan.pdf`) of the meta-analysis results.

13. Applying genomic control

NOTE: Step 10 (correcting out-of-range p-values) need not be done if genomic control is being applied to the meta-analysis results. However, creating a QQ plot (step 11) is necessary in order to compute lambda for the meta-analysis so that lambda can be used to apply genomic control.

If genomic control is being used, `run_mantel.csh` calls the `lambda_correct.R` script to correct p-values according to the lambda of the study (passed to `lambda_correct.R` as a command line argument). The GC-corrected values are then merged with the final meta-analysis results in the file lambda_corrected.txt in the columns P_GC, SE_GC, and Z_GC.

`run_mantel.csh` will create a Manhattan plot of the results (`manhattan.GC.pdf`) once genomic control has been applied.

--- META-ANALYSIS COMPLETE ---

PART II

--- DOWNSTREAM ANALYSES ---

Once the meta-analysis is complete, you most likely want to create a list of independent genome-wide significant SNPs from your meta-analysis and create regional association plots around these index SNPs. `run_clumps.csh` is designed to run through these steps.

1. Setting parameters and clumping SNPs

At the beginning of `run_clumps.csh`, you must set your clumping parameters. The script defaults to a clumping p-value of genome-wide significance (p = 5e-8) and an r-squared threshold of 0.05 (for pruning based on linkage disequilibrium).

`run_clumps.csh` then takes the results file (set to the `lambda_corrected.txt` results, though non-GC corrected data can be used as well) and clumps based on the parameters, producing the file indexsnps.txt that lists the independent SNPs (rsIDs only) from the meta-analysis.

2. Results for independent hits

`run_clumps.csh` next re-runs the MANTEL script, this time producing verbose results (i.e. results on a cohort-by-cohort basis) for the independent hits and printing them to `clump.indexsnps.verbose.out`.

`run_clumps.csh` will also create a file, indexsnps.dbsnp.txt, that contains dbsnp annotation for the top hits as well as gene-annotations around the hits.

3. Regional association plots

Finally, `run_clumps.csh` points to the `assocplot.R` script in order to create regional association plots of the genome-wide significant SNPs.

`run_clumps.csh` will walk through a series of steps that parses linkage disequilibrium data and creates files necessary to produce the regional association plot.

`run_clumps.csh` will use HapMap data to compute LD around the top SNPs and produce the following files containing information about the top SNPs:

a. `rsID.txt` - information on your index SNP
b. `rsID.sorted.txt` - information on your index SNP, sorted by p-value

c. `rsID.locus.txt` - information about the locus around your SNP, including functional annotation
d. `rsID.snap.txt` - more locus information, also containing results from the meta-analysis
e. `rsID.snap.sorted.txt` - same as file above, but sorted by p-value
f. `rsID.assocplot.snps.dat` - parsed information to annotate SNPS and LD in the regional association plot
g. `rsID.assocplot.genes.dat` - information to annotate genes in the association plot

Then, manually entering information about the index SNP(s) into `run_clumps.csh` (see end of script), the `assocplot.R` script will create a regional association plot of the index SNP and SNPs around it, using p-values from the meta-analysis.

--- END OF DOWNSTREAM ANALYSES ---

PART III

--- INDEX OF SCRIPTS AND FILES ---

SCRIPTS (in order of usage, in SCRIPTS/)

1) *run_mantel.csh* (points to scripts 2-15)

Description: script that QCs raw data (including creating 3 QQ plots, an imputation quality score histogram and a Manhattan plot per cohort), prepares the data for use by MANTEL, runs the meta-analysis, and does some post-meta steps (QQ plot, Manhattan plot, correcting out-of-range p-values and applying genomic control if desired).

Usage: no command line arguments, but scripts 2-15 and all reference files should be in the same directory that run_mantel.csh is run from.

2) *qqplot.R*

Description: creates a standard QQ plot of observed vs. expected p-values. Calculates lambda. Possible data types: p-values (PVAL), z-scores (Z), chi-square values (CHISQ). Input file should not have a header line.

Usage: R CMD BATCH -CL -input_values -data_type -output.pdf qqplot.R

3) *qqplot_by_maf.R*

Description: creates QQ plot with observed/expected p-values stratified by the minor allele frequency of the corresponding SNP. Takes p-values only. Input file should not have a header line.

Usage: R CMD BATCH -CL -input_values -PVAL -output.pdf qqplot_by_maf.R

4) *qqplot_by_info.R*

Description: creates QQ plot with observed/expected p-values stratified by the imputation quality score of the corresponding SNP. Takes p-values only. Input file should not have a header line.

Usage: R CMD BATCH -CL -input_values -PVAL -output.pdf qqplot_by_INFO.R

5) *obsexp.R*

Description: creates a histogram of imputation quality scores for a given cohort.

Usage: R CMD BATCH -CL -input_values -output.pdf obsexp.R

6) *merge_tables.pl*

Description: merges two tables based on a common index column in both. The script prints all of the data from file2, and then prints all of the corresponding columns from file1, indexed on the chosen index column.

Usage: ./merge_tables.pl --file1 file1_name --file1 file2_name --index index_column

7) *uniquefy.pl*

Description: finds all unique values in a list.

Usage: ./uniquefy input_file

8) *manhattan_plot.R*

Description: creates a Manhattan plot from the data provided (chromosome, position, p-value).

Usage: R CMD BATCH -CL -input_data -output.pdf manhattan_plot.R

9) *MANTEL.pl*

Description: runs meta-analysis (see documentation and script for details).

Usage: ./MANTEL.pl --params params_file --snps snp_list --dbsnp dbsnp_file --freq HAPMAP.frq --genes refseq_file --dist distance_kb --out file.out --ext iteration --no-header [--random-effects]

10) *meta.R*

Description: Corrects out-of-range p-values that are computed as p=0 when the chi-square statistic is too large for Perl to compute an accurate p-value.

Usage: R CMD BATCH -CL -input_data -output_data meta.R

11) *histograms.R*

Description: creates histograms of a) the distribution of effective sample size, calculated per SNP, across all SNPs used in the meta-analysis and b) the number of contributing studies per SNP, across all SNPs used in the meta-analysis.

Usage: R CMD BATCH -CL -input_data -number_of_studes -output histograms.R

12) *lambda_correct.R*

Description: applies genomic control to the meta-analysis results (adjusting p-values based on the overall lambda of the study).

Usage: R CMD BATCH -CL -input_data -lambda -output lambda_correct.R

13) *parse_table.pl*

Description: allows you to parse a data table based on the columns you wish to extract from the table (by column name).

Usage: ./parse_table.pl --col Col_name1,Col_name2,Col_name2

14) *run_clumps.csh* (points to scripts 6, 9, 13, 15-18)

Description: takes meta-analysis results, clumps them into a list of independent genome-wide significant hits, produces cohort-by-cohort results on each independent hit, and produces regional association plots.

Usage: Set clumping p-values and r-squared threshold before running the script (see beginning of script). Script is currently set to use a clumping p-value of genome-wide significance (p = 5e-08) and to LD-prune at an r-squared threshold of 0.05.

There are no command line arguments, but all of the files needed to run run_mantel.csh are also necessary to run run_clumps.csh

15) *parse_loci.pl*

Description: parses information in a particular locus (i.e. around independent SNP hits).

Usage: ./parse_loci.pl --file meta_results.txt --snps index_hits_with_dbsnp-annotation --out outfile

16) *parse_clumps.pl*

Description: produces information about an index SNP and other SNPs in that locus.

Usage: ./parse_clumps.pl --file input_file --snp snp_rsID

17) *make_assocplot.pl*

Description: prepares all the datafiles necessary to run assocplot.R, which produces a regional association plot of your independent genome-wide significant hit(s).

Usage: ./make_assocplot.pl --locus input_data_about_locus --genes refseq_files.txt --snp snp_rsID --out output

18) *assocplot.R*

Description: creates a regional association plot using files created by make_assocplot.pl. Note that the command line argument "recomb/rate/directory" should be the name of the directory where the recombination rate files are stored (currently in RESOURCES/RECOMB_RATES/).

Usage: R CMD BATCH -CL -input_data.snps.dat -input_data.genes.dat -rsID -n_effective -p-value -locus_name -recomb/rate/directory/ -output.pdf assocplot.R

REFERENCE FILES (in REFERENCES/)

1) `rsID conversion file`

Description: file containing a column with rsIDs of SNPs in your meta-analysis and a column with the corresponding rsIDs for those SNPs in a single build (e.g. Build129).

Format:
SNP_meta SNP_BuildXXX

2) Statistics/ (note: found in home directory)

Description: package used by Perl in order to compute p-values.

3) `dbsnp129_hg18.txt`

Description: dbsnp file containing annotations for SNPs in dbSNP (corresponding to the build you are using for the meta-analysis).

Format:
Chromosome chromStart chromEnd SNP strand observed class func.

4) `reseq_genes.short.txt`

Description: file containing positions of RefSeq genes. The file does not have a header line.

Format:
Chromosome GeneStart GeneStop GeneName

5) `HAPMAP/`

Description: Directory containing HapMap information in Plink file format (`bim/bed/fam`) to be used during the meta-analysis as well as for downstream analyses (e.g. clumping). This directory should also contain a file with HapMap minor allele frequencies (*.`frq`) created using the HapMap files and the command --freq in PLINK.

6) `Params file`

Description: a file containing descriptive information about the cohorts being used in your meta-analysis. The file does not have a header line.

Format:
Cohort_name Lambda Sample_size Correction_factor data_file

9) `refseq_hg18.footprints.txt`

Description: a file used to annotate the regional association plots with genes near the SNP of interest.

Format:
GeneName Chromosome Strand GenStart GeneStop

10) `dbsnp129_hg18_excl_non_ref.txt`

Description: provides additional dbsnp annotation for your independent hits.

Format: identical to the dbsnp129_hg18.txt file.

11) `RECOMB_RATES/`

Description: this directory contains all of the files necessary to plot recombination rate on regional association plots. The directory needs to be passed as a command line argument to the script assocplot.R

Format:
Position COMBINED_rate(cM/Mb) Genetic_Map(cM)