MAGMA joint modelling options and QC read-me (v1.07a)

This document provides a brief overview of the (application of) the different options for conditional, joint and interaction analysis added in version 1.07, and a manual for the supplemental R scripts provided to perform post-hoc QC and inspection of analysis results. A more detailed guide on performing and interpreting these analyses can be found in the Supplemental Materials of the *Conditional and interaction gene-set analysis reveals novel functional pathways for blood pressure* (De Leeuw, et al., Nature Communications (2018)) paper (linked on the MAGMA website). More information on running gene-level analysis, including conditional, joint and interaction analysis, is also provided in the main MAGMA manual (the below largely overlaps with the manual, and is intended to provide some additional information on usage of the different options).

Conditional, joint and interaction models

The --model flag provides a number of different options to specify multi-variable models, but all these models use the same general linear regression structure. The same models can often be specified using several of the available options, but some options are more convenient than others for particular purposes.

There are three general classes of options to specify models: *condition*, *joint* and *interaction*. The *condition* options can be used to specify variables that should be included in all of the analyses that will be performed, whereas the other two types of option are used to directly specify particular multi-variable models: with the *joint* options, multi-variable models containing arbitrary numbers of main effects can be specified; with the *interaction* options, pairs of variables to interact with each other can be specified.

The different *condition* options can be combined with each other and with any *joint* or *interaction* option, but only a single *joint* or *interaction* option can be used at a time. If no *joint* or *interaction* option is used, all variables in the active selection are analysed one at a time. By default this active selection consists of all variables available from the input files, excluding any used for the *condition* options. The *analyse* option of --model can be used to restrict the active selection, however. This can be useful in particular in combination with some of the *joint* and *interaction* options, many of which are based on the active selection in some way.

Conditional analysis options

Four options are available for conditional analysis: *condition*, *condition-hide*, *condition-residualize* and *condition-interaction*. The first two produce the same results, the only difference is between them is that for variables specified using *condition-hide* no output is generated in the results file. This can be useful for decluttering the results file, if the aim is only to condition on those variables and their parameters and p-values for these multi-variable models are not themselves of interest.

When using the *condition-residualize* option, all variables specified for it will be residualized out of the genetic association score, and this residualized score is then used in all other analyses as the outcome variable; this is the same way that corrections for internal covariates are processed (specified via the *correct* option for --model). This provides a stronger protection against confounding when analysing other variables correlated with these residualizing variables (all shared effect is assigned to the residualizing variables, rather than being divided amongst all the variables), though at the expense of

potentially overcorrecting and losing power as a result. A secondary advantage of the *condition-residualize* option is that it can cope with any level of collinearity. However, as with *condition-hide* no output is generated for these variables in the results file.

The last conditional analysis option is *condition-interaction*, which functions in essentially the same way as *condition* but conditions on a whole interaction (interaction term plus main effects). Interactions are specified in pairs of two variables, and any number of interactions can be conditioned on. The same variable is allowed to occur in multiple interactions specified.

Joint analysis options

Two options are available for joint analysis: *joint* and *joint-pairs*. The *joint* option is the more general option, this is used to select a file which contains the multi-variable model specifications. Each line corresponds to a separate model to analyse, which will contain all variables listed on that line (as well as any variables included via the various *condition* options; note that at present, any specified models containing conditioned-on variables are skipped). The models can contain any number of variables, provided that there are sufficient degrees of freedom in the data to fit the model.

As stated above as well, the *joint* analysis options are intended for specifying models containing unique combinations of variables. If you find yourself including the same variable in all models in the *joint* file, this would be more conveniently included using *condition* instead (though statistically it makes no difference in results).

The *joint-pairs* option is provided for convenience, for a scenario where analysis of all (or many) pairwise combinations of variables is desired. A common application of this would be when following up on the significant marginal associations of a gene-set analysis, using *joint-pairs* to investigate the overlap in associations of all the significant gene sets. The *joint-pairs* option will analyse all possible pairs of variables in the active selection, and it is therefore generally best used in conjunction with the *analyse* option to restrict that active selection.

Interaction analysis options

For interaction analysis, four options are available (plus *condition-interaction*): *interaction*, *interaction-pairs*, *interaction-each* and *interaction-all*. The *interaction* and *interaction-pairs* options function very similarly to the *joint* and *joint-pairs*: the first uses a file to specify specific interaction pairs (each line corresponds to an interaction to test, and should contain exactly two variables), the second is a shorthand option for analysing interactions between all possible (and valid) pairs of variables in the active selection. The *interaction-each* option is similar to *interaction-pairs*, but it specifies a particular list of variables to serve as interactors. For each variable on the specified list, all (valid) interactions between that variable and all variables in the active selection will be analysed.

With the *interaction-all* option, multiple interactions can be included in the same model. This option also specifies a list of variables to serve as interactors. For each variable in the active selection, a model is analysed containing all interactions between that variable and the variables on the specified list. Note that the analysis of a model will only be performed if all the interaction terms in it are valid (see below), and this option is therefore primarily useful when using it mostly with (continuous) covariates.

An example application (also used in the Nature Communications paper) is when analysing the effects of tissue-specific expression. When analysing tissue-specific expression, to ensure that the estimated effects are indeed tissue-specific it is necessary to condition on a general (across-tissue) measure of gene expression as well. Therefore, when analysing interactions between tissue-specific expression and gene sets, it is important to condition on the interaction between the general expression

and the gene set as well (as without it, the analysis could also be detecting interactions that are not specific to that tissue, ie. interactions between the gene set and the general expression measure).

Whenever a model includes an interaction, this means including both the interaction term itself as well as the corresponding main effects. For set by covariate interactions, the interaction is defined such that the main effect of the set corresponds to the difference in slope of the coefficient (for genes in the set vs outside it), measured at the gene-set mean of that covariate.

For any interaction, some additional requirements must also be met for them to be considered valid. For set by set interactions, this means that there must be sufficient (but not too much) overlap between the gene sets; for set by covariate interactions, the gene set must be sufficiently large. Consult the main MAGMA manual on how to change the default settings for sufficient size and overlap. At present, interaction analysis between continuous covariates is not available, and all interactions between pairs of such variables will therefore be considered invalid automatically.

Performing post hoc QC for gene sets and set by covariate interactions

For both gene sets and set by covariate interactions, it is important to perform post hoc checks of significant results, to ensure there is no undue influence of outliers. MAGMA will generate additional output to perform such checks, and a set of R scripts is provided to help facilitate these checks (R can be downloaded here: https://www.r-project.org/). This document is intended as a manual for using those R scripts only, as a companion to the more detailed rationale and guideline on the issue of outliers in this context that is provided in the Supplemental Materials of the Nature Communications paper.

NOTE: the posthoc_qc.r script requires output from MAGMA **v1.07a** or higher.

General functions

The QC functions are provided in the script file 'posthoc_qc.r', and can be loaded by copy-pasting them into your R window, using `source("posthoc_qc.r")` (provided the script file is in your current working directory), or using script loading functionality in your R interface. To provide an overview, the main (wrapper) functions are listed at the top of the file with a brief comment on their input and output.

All functionality is based on the per-gene output files generated by MAGMA for significant results (the *alpha* modifier of --model can be used to adjust the significance level that determines this output). These are the .gsa.genes.out and .gsa.sets.genes.out files for gene-set analysis, and .gsa.inter.genes.out for interaction analysis. Three loading functions are provided to load these files in R for use in the other provided functions: `load.sets`, `load.ss` and `load.sc`. These functions load the relevant files and perform some initial processing, then output an environment object containing the loaded results and information. Note that although the `load.ss` function is provided, at present no specific QC functions are at present available for set by set interaction analysis (in part because it does not share the susceptibility to outliers that set by covariate interactions have).

Having loaded some results (eg. `res = load.sets("my_analysis")`), a summary of the gene sets or interactions for which information is loaded can be obtained using `show.info(res)`, and specific entries can be filtered from the results object using the `filter.results` function. The latter can be useful if only some of the results are of interest, to declutter the PDF with output plots and speed up the process. As it is a standard R environment object, its contents can also be accessed directly in the normal ways: use `names(res)` to list the variables it contains and the $ operator to access them (eg. `res$info` to access the contents loaded from the .gsa.sets.genes.out or .gsa.inter.genes.out files).

QC for gene-set analysis results

The .gsa.sets.genes.out and .gsa.genes.out files can be used to check whether possible outlier effects may be unduly influencing the association of a gene set (note that these files are only generated if no *joint* or *interaction* --model options were used). This is done by means of a set-specific QQ-plot, which plots the expected and observed quantiles of the residual Z-scores (of the model not containing the gene set itself) of genes in the set, with the genes corresponding to the 25^{th}, 50^{th} and 75^{th} quantile marked in black.

The Supplement Materials of the Nature Communications paper provides a more extensive guideline on interpreting these plots, but in brief: ideally, the plot starts to deviate upwards from the diagonal early (ie. close to the plot origin), which would indicate that the level of association is consistently elevated for most or all of the genes in that set. By contrast, if deviation occurs much later this suggests that possibly only a small subset of genes in the set is responsible for its association, in which case the result is likely not actually representative for that gene set (eg. it may simply partially overlap with a different gene set that actually is relevant to the phenotype).

An upper 95% confidence band is also added to the plot, which is generated using a sampling procedure based on all genes in the analysis. Genes outside of the confidence band are marked in red, remaining genes in gray. This confidence region reflects how much a particular quantile is likely to deviate upwards from the diagonal by chance, which is useful in judging generally how strong the deviations are.

The QQ-plots can be generated using `plot.sets(res, "PREFIX")`, where the second argument is the prefix for the output file. This will generate a PDF containing QQ plots for all the gene sets in the `res` object. Note that running this function can take a while since a sampling procedure is used to generate the confidence bands. Both the number of permutations used for this as well as the percentage for the confidence band can be changed if desired.

QC for set by covariate interaction analysis results

Analysis of set by covariate interactions can be quite susceptible to outliers, as the interaction effect is driven by genes in the set only and it is possible that by chance it contains a small number of genes with unusually high (or low) values on both the covariate as well as the genetic association Z-scores. This can be a problem in particular for small gene sets, which is why it is prudent to restrict the analysis to interactions with larger sets only if possible.

Two strategies are provided for guarding against this issue, and it is recommended that both are used in conjunction to provide more information. The first strategy is to use interaction-specific scatter plots to identify and mark likely outliers, and to rerun the analyses with those outliers dropped from the gene set to see how much this impacts the results. The second strategy is to partition the gene set into smaller subsets based on the covariate, and analyse those subsets. This effectively dichotomizes the interaction, and therefore ameliorates the impact of outlying covariate values.

Having loaded a results object with the `load.sc` function, interaction-specific scatter plots can be generated using `plot.sc`. These show the residual Z-scores (of the model not containing the interaction term or corresponding main effects) plotted against the covariate for genes in the set, both standardized within the set. Genes marked as outliers are shown in red.

To use the first strategy, outliers must first be marked as such, which can be done in two ways. Using the `outliers` function, outliers can be marked using one of three simple criteria: based on univariate deviation (`sd.uni`), if genes are more than the specified distance from the origin on either the horizontal or vertical axis (due to the standardization, this distance is measured in within-set standard deviations); based on multivariate deviation (`sd.multi`), if genes are more than the specified

distance from the origin in the two-dimensional space; or using an iterative modification of `sd.multi` (this is the default setting).

For this iterative approach a gene is marked as an outliers if a) it is further than the specified distance from the plot origin and b) all other genes within distance of the gene (using the same distance parameter) that are closer to the plot origin, if any, are also outliers. This approach is therefore more selective than the `sd.multi` setting, as it will only mark relatively isolated (clusters of) genes as outliers, leaving genes more distant from the origin but with sufficient genes in between to 'connect' it alone.

Aside from the `outliers` function, outliers can also be marked (and unmarked) manually using `mark.outliers` and `unmark.outliers`. The recommended approach is to first use `outliers` as a first pass, inspect the scatter plots and if necessary modify the outlier selection using the manual functions (note that the `outliers` function will overwrite any existing outlier markings if applied again). To manually mark or unmark outliers the internal IDs of those genes must be provided. These can be obtained by inspecting the relevant entry in the results object, but more conveniently by using `plot.sc` with `show.ids=T`; with that setting, the dots in the scatter plots will be replaced by the IDs for the genes.

Once all outliers are appropriately marked, the `outlier.sets` function can be used to write them to a file for use with MAGMA. This function will create two files: a .sets file that contains the gene-set definitions with the outliers removed (by default it will also contain copies of the original sets for easy comparison), and a .model file that specified the pairs of interactions that need to be analysed. These files can then be used to rerun original analysis with the filtered gene sets. This reanalysis should use the same settings as the original analysis, and as such any gene sets that were used to condition on must be inserted into the .sets file generated by `outlier.sets`. This can be accomplished by using `read.setfile` to load the original gene-set definition file (and filter on required gene sets), and passing the output to `outlier.sets` which will include it in the .sets file. For the analysis, the .models file can be used with the *interaction* modifier of --model to run all the relevant interactions (unless *interaction-all* was originally used, in which case that should be used again instead).

The second strategy for guarding against outliers is the partitioning approach. Using this, genes in the gene set are ordered by their value on the covariate, and then partitioned into a specified number of subsets (four, by default). In case of interaction the level of gene association will vary throughout the set as a function of the covariate. For positive interactions, strongest genetic associations will then therefore be found in the higher partitions; for negative interactions, in the lower partitions.

These subsets are analysed one at a time, replacing the interaction and covariate main effect. As they are not dependent on the actual covariate values they are not sensitive to outlier values on it. Moreover, the analysis of the partitions can also offer additional information about the interaction itself, even if no outliers are present.

To run these analyses, the `partitioned.sets` function can be used to create the partitions and write them to a file for use in MAGMA. It will also create a .models file specifying all the models that will need to be run. For each partition two models will be specified: one for the partition on its own and one conditional on the full original gene set. The number of partitions to be used can be varied, but is set to four by default.

Using the created .sets file, the partitions can be analysed using the .models file with the *joint* modifier of --models. The same settings as in the original analysis, except for the *interaction* option, should be used. If other gene sets were conditioned on, these can again be included in the .sets file created by `partitioned.sets` by means of the `read.setfile` function.

At present, if *interaction-all* was originally used with more than one variable, there is no straightforward option in this follow-up analysis to condition on interactions that were also included in

the original analysis in this way. Options for this will be added in later updates, for now it is recommended that the variables used for *interaction-all* are conditioned on using *condition* (excluding the variable on which the partitioning is based; if partitions for multiple interactions are being analysed and the covariate involved in the interactions varies, these will need to be analysed in separate MAGMA runs).