**Analysis Workflow**

Table of Contents

Required Programs & Data Formats……………………………………………………………………………………..2

Discovery Analysis

 Filtering Variants………………………………………………………………………………………………………3

 INTERSNP Interaction Analysis…………………………………………………………………………………4

 Identification of Significant Interactions………………………………………………………………….4

 Calculate LD: Within Pair………………………………………………………………………………………….5

 Calculate LD: Between Pair………………………………………………………………………………………5

Replication Analysis

 INTERSNP Replication Analysis……..…………………………………………………………………………6

Exploration of Confounding Processes

 Ceiling/Floor Effect………………………………………………………………………………………………….7

 Missing Genotypes…………………………………………………………………………………………………..7

Variants within the Probe Binding Site…………………………………………………………………….7

 Stratified Analysis…………………………………………………………………………………………………….8

 Population-specific LD……………………………………………………………………………………………..8

 D’ Filtering………………………………………………………………………………………………………………..8

 Conditional cis-eQTL analysis…………………………………………………………………………………..8

**Required Programs & Data Formats**

The analyses in this study required the following programs:

*INTERSNP*

The interaction test itself was performed using INTERSNP, which is available for download at <http://intersnp.meb.uni-bonn.de/>. The output files from INTERSNP were used for most subsequent analyses; consequently, the in-house scripts used to perform follow-up analyses are largely dependent on the INTERSNP output format. INTERSNP requires that input files be in PLINK files (PED/MAP, TPED/TFAM, or BED/BIM/BAM).

*PLINK*

PLINK was necessary for a variety of data filtering steps. It can be downloaded here: (<http://pngu.mgh.harvard.edu/~purcell/plink/>

*Bedtools*

This suite of tools was used to do a variety of filtering steps based upon genomic location. It can be downloaded here: <http://bedtools.readthedocs.org/en/latest/>

*R*

R, a commonly used language for statistical analyses, can be downloaded here: <https://www.r-project.org/>

We use the q-value package in order to calculate the FDR rate. It can be downloaded, with complete documentation, here: <http://bioconductor.org/packages/release/bioc/html/qvalue.html>

In addition to these tools, the authors used within-house scripts written in Python to perform a variety of analyses. While the purpose of each script is discussed in this document, more detailed information is provided by the script itself in most cases; simply run it without any arguments to see more. The majority of scripts also create a logfile, which provides detailed information on the specific run.

**Discovery Analysis**

*Filtering Variants*

Prior to performing any interaction testing, we imposed several filters to reduce the number of association tests performed. The below instructions assume there is one PED/MAP file containing all genetic data. This file is filtered for rare variation, then broken into individual PED/MAP files for each gene with expression data.

1. *Use INTERSNP to perform single-marker association tests.*

INTERSNP can run single-marker tests in addition to tests for interactions. In order to run INTERSNP, a ‘selection\_file’ must be provided. Extensive documentation for selection\_files and all their options is provided in the INTERSNP manual. An example selection\_file used in our analysis is shown below:

The script *single\_marker\_ciseqtl\_analysis.py* script generates this selection file and then runs INTERSNP. It can be run in parallel to decrease runtime.

1. *Identify nominal cis-eQTL from INTERSNP results.*

The script *identify\_cis\_eqtl.py* generates a file containing all variants passing a given significance threshold in INTERSNP’s single-marker analysis.

1. *Restrict PED/MAP file for each gene to just nominal cis-eQTL.*

Use the ‘extract’ argument in PLINK to reduce the PED/MAP files to just the list of nominal cis-eQTL identified above.

**Note:** In our analysis, nominal cis-eQTL had already been identified in our dataset by Veyrieras et al. (2007). We used their list of cis-eQTL variants.

*INTERSNP Interaction Analysis*

Once a PED/MAP file containing only nominal cis-eQTL for each gene has been generated, these files are provided to INTERSNP for interaction testing. INTERSNP requires a ‘selection file’ to run; extensive documentation of the selection files and all their options is provided in the INTERSNP manual. The selection file can be highly customized in order to fit your own analysis; the generic format of the selection file used in our analysis is shown below:

MAP A4GALT.map

PED A4GALT.ped

COVARIATEFILE covariates\_for\_intersnp.txt

OUTPUTNAME A4GALT

QT 1 // Quantitative Trait? 1=yes, 0=no

TWO\_MARKER 1 // Two-marker-analysis: 1=yes, 0=no

PRETEST 0 // Pretest: 1=yes, 0=no

TEST 6 // Test: 6 = GENOTYPIC INTERACTION, with both additive and dominant components included

SINGLE\_MARKER 2 // 2 = genotype test w. 2 df

PRINTTOP 500000

COVARIATES 1-4; // top four principle components

END

We provide the script *discovery\_interaction\_analysis.py,* which was used to generate the selection files used for this study. While it’s unlikely the exact same settings will be desired in another analysis, users can certainly alter the source code to tweak the selection file settings. Once the selection files are generated, the INTERSNP analysis is run.

We recommend that this step of the analysis be run in parallel if possible to reduce the runtime. This can be done by launching the script multiple times, each time with a distinct subset of genes provided.

*Identification of Significant Variants*

Once all interaction tests for all genes have been performed, the script *get\_interaction\_pvalue.py* extracts out the unadjusted p-value for each test. These p-values output as a list, correctly formatted to compute the FDR p-value in R.

The FDR correction is computed in R, using the qvalue package. To compute the FDR=5% threshold use the following commands:

p <- scan("input\_file\_name")

qobj <- qvalue(p)

max(qobj$pvalues[qobj$qvalues<=0.05])

This returns the p-value threshold corresponding to the FDR=5%. Provide this p-value to the script *get\_significant\_interactions.py*, which will scan all the INTERSNP interaction results to extract all interactions below whatever p-value threshold you specify.

*Calculate LD: Within pair*

At this stage, we computed the r2 and D’ between all variants identified as significantly interacting. This analysis utilizes PLINK to perform the calculation. The script *calc\_within\_pair\_ld.py* performs this analysis for all interacting variants. If desired, the user can specify r2 or D’ thresholds; interactions that do not pass this filter will not be returned. At this stage in our analysis, we imposed an r2 filter ≤ 0.6 and no D’ filter.

*Calculate LD: Between pairs*

Many of the significant SNP-SNP interactions may essentially be redundant due to LD; in other words, they many represent the same pair of interacting genomic loci. For two ieQTL models to be considered redundant in our analysis, each SNP within one significant ieQTL model had to be in high LD (r2 ≥ 0.9) with a SNP within the second ieQTL model, and vice versa. By using this criterion, the pairs were effectively correlated at r2 ≥ 0.8, the threshold typically used for tag-SNP selection.

The script *calc\_between\_pair\_ld.py* calculates the LD between all variants in a significant interactions for the same gene, computes the r2 between these variants, groups redundant pairs together, and selects the most significant interaction to represent the group. This script can be run in parallel if desired – an example slurm script is provided to illustrate how this can be done (*calc\_between\_pair\_ld.slurm).* If run in parallel, the run the script *merge\_results\_from\_calculate\_between\_pair\_ld.py* to combine the output from each partition.

**Replication Analysis**

The analyses required for the replication analysis are identical to those done in the discovery analysis; however, the restriction of the replication analysis to only interactions identified as significant in the discovery analysis requires a major difference in the INTERSNP analysis, which is discussed below.

*INTERSNP Replication Analysis*

In the discovery analysis, we wished to test all pairwise combinations of nominal cis-eQTL. In the replication analysis; however, we only wish to test very specific pairs of variants for significant interactions. INTERSNP permits you to do this by specifying a ‘combi file’ (described in detail in INTERSNP’s manual). This requires changes in the selection file, otherwise, the analysis is the same.

The steps necessary to run the replication analysis in a targeted fashion have all been implemented in the *replication\_interaction\_analysis.py* script. This script:

* Generates a combi file, which allows INTERSNP to only test the specified combinations rather than all possible pairs
* Generates a new selection file, which references this combi file
* Runs the INTERSNP analysis

Once INTERSNP has been run, use the *get\_significant\_interactions.py* to identify all interactions that pass a specified p-value threshold. In our analysis, we used a p-value threshold of 0.05, although a variety of other cut offs could be supplied.

**Exploration of Confounding Factors**

We then investigated the role of confounding processes on significant interactions identified in the discovery analysis. A separate analysis was conducted for each confounding factor, which are described below.

*Missing Genotypes*

It order to determine if missing genotype combinations were present, we restricted the PED/MAP files from the discovery analysis down to just the two variants of interest. We then recoded those variants into 0/1/2 encoding. We then checked to see if all combinations 00/01/02 etc. were observed. This analysis is implemented in the script *genotype\_check.py*, which calls PLINK for a variety of the analyses.

*Ceiling Floor effect.*

This is a two-step analysis. First, the estimates for the main effects in each interaction model have to be extracted from the INTERSNP files. This is implemented in the script *main\_effect\_extract.py*. Then, the output from this script and the *get\_significant\_interactions.py* script can be used in conjunction to run the *ceiling\_floor\_effect.py* script, which determines if the betas are consistent with the ceiling/floor effect.

*Variants within the Probe Binding Site.*

We needed to check if variants occurred within the probe binding site, as they may disrupt binding and consequently confound our analysis.

Several steps, outlined below, went into this analysis.

1. The sequence of the probe\_ids is included in documentation files for the microarray chip. We extracted the sequence of the probe\_id for each significant gene.
2. We then used BLAT to locate the sequences within the genome. If multiple alignments were observed for a probe, we performed a manual identified the most likely binding site (filtered based on quality score, length).
3. We then used these genomic coordinates to extract all variants within the binding sites identified in the 1000 Genomes project. This analysis was done for each probe using VCFtools, with the arguments --to-bp and --from-bp to get only the genomic regions of interest. The VCF files had been restricted to only individuals within our discovery analysis.
4. We then checked whether these variants were common or rare. This is implemented in the script *variants\_in\_probe\_check.py.*
5. If only rare variants were in the probe binding site, we then:
	1. Converted the VCF files to plink files, identified all individuals with rare variants, and removed these individuals from the analysis using PLINK. This is implemented in the *variants\_in\_probes\_remove\_ind\_with\_rare.py* script.
	2. Re-run the interaction analysis with the script *variants\_in\_probe\_interaction\_analysis.py*. This script uses the output from *variants\_in\_probe\_check.py* to identify genes with variants in the probe, then generate the significant combinations for those genes, and then perform the interaction analysis.
	3. Checked to see if the interaction remained significant. This is done with the *get\_significant\_interactions.py* script, which was previously discussed in the discovery analysis section.

*D’ Filtering.*

To calculate and filter out interaction based on D’, refer to the *Calculate LD: Within pair* portion of the Discovery Analysis section. The scripts described here can be used to filter based on either r2 or D’.

*Stratified analysis.*

This analysis exploits a feature of INTERSNP – if covariates are not provided for an individual, that individual is not included in the analysis. As a consequence, we didn’t need to generate new PED/MAP files here – all we had to do was to restrict the covariate file to individuals in each of the three discovery set populations. We then used the script *population\_stratification\_interactional\_analysis.py* to perform the interaction test in each population separately. We then used the *get\_significant\_interactions.py* script to extract all results (supply pvalue > 1) in each population.

*Population specific LD.*

Next, we calculated the LD between the two interacting variants in each of the discovery set populations separately using PLINK. This analysis is implemented in the script *calculate\_within\_pair\_ld\_population\_stratification.py*.

*Conditional cis-eQTL analysis.*

The conditional cis-eQTL analysis was a multi-step process. We first downloaded the 1000 Genomes genomic data. We then extracted out the data for individuals in our discovery dataset, and generated a ped/map file for each gene’s regulatory region. These files are provided.

We then ran a cis-eQTL single marker analysis. This is implemented with the script *single\_marker\_cis\_eqtl\_intersnp.py*. This script can be run in parallel if desired. The slurm script we used to do this is provided as *single\_marker\_cis\_eqtl\_intersnp.slurm*.

We then identified all cis-eQTL significant at a specified threshold. This is implemented in the script *identify\_cis\_eqtl.py*.

We then generated all combinations of cis-eQTL and pairs of interacting variants. This is necessary in order to reduce the number of tests performed by INTERSNP. This is implemented in the script *generate\_combinations\_for\_conditional\_analysis.py.*

Next, we ran the conditional analysis in INTERSNP. This is done by the script *conditional\_cis\_eqtl\_intersnp.py*. This script can be run in parallel, with each gene getting its own partition. The *conditional\_cis\_eqtl\_intersnp.slurm* script is a template for how to run this script in parallel.

To identify the conditional cis-eQTL that most mediated the interaction, run the script *conditional\_cis\_eqtl\_analysis\_results.py.*

These scripts are designed to be run in succession, and should take the output of the previous script without any necessary reformatting.