**BIOINFORMATIC METHODS: ARGINASE I PAPER 2016**

Microarray and Pre-Processing

The Affymetrix GeneChip® Mouse Genome 430 2.0 Array was used for this analysis. Array hybridization was done by the UCLA Clinical Microarray Core using their protocols.

All pre-processing of raw data was done in R. Raw fluorescence data was stored as .CEL.gz files, which were loaded into R as an AffyBatch object (using the affy package in R). This object was then processed using Robust Multi-Array Average (RMA) normalization, which removed background ‘noise’ and converted data to a log2 scale. ComBat() in the sva R package was then applied to the normalized data to adjust for batch effects. Since all knock-out and wild-type mice were run in one hybridization batch, and all heterozygote and treated knock-out mice were run in the other hybridization batch, genotypic effects could not be fully differentiated from batch effects. Consequently, some ‘true’ genotypic expression effects were lost at this stage of the analysis when we removed batch effects.

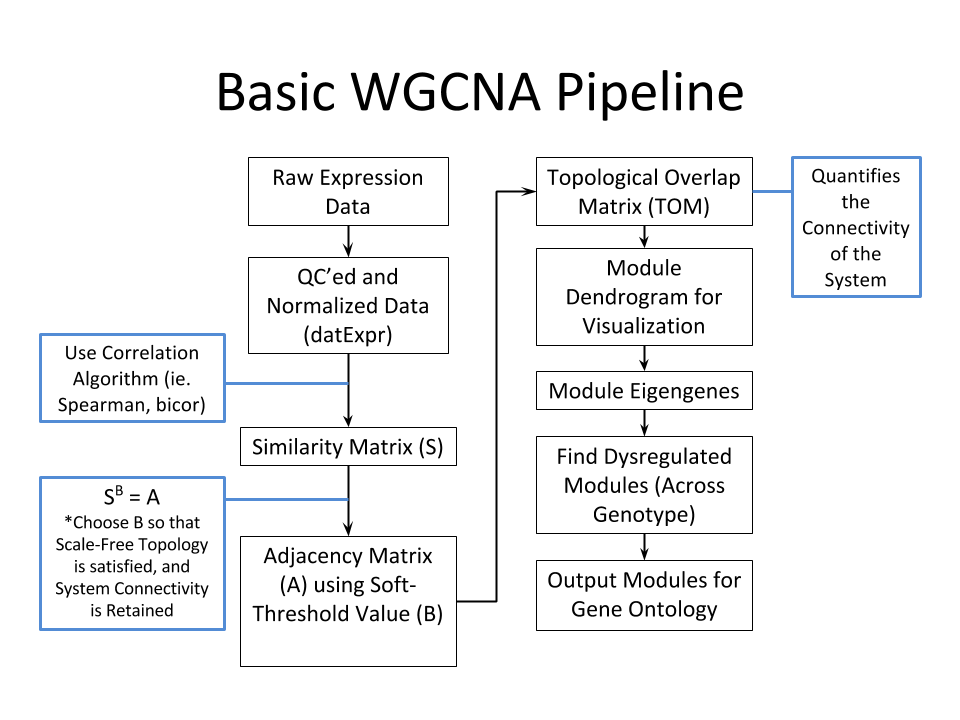
No outlier samples were found in the data (outlier investigation was done using a WGCNA sample network connectivity approach). Affymetrix probes were summarized and re-annotated with Ensembl gene names from the Ensembl July 2015 archive (Ensembl version 81). This annotation dataset was chosen because it most closely matched the time of the Affymetrix microarray manufacture date and following hybridization run. Finally, the dataset was filtered so that genes with a standard deviation in the 25th quantile or lower were removed. This step was performed to maximize the connectivity that could be found in our subsequent WGCNA analysis. This normalized, batch-adjusted, re-annotated, and filtered gene expression data matrix was our input for all later WGCNA and differential expression analyses.

The limma package in R was used for differential gene expression analysis. Our linear model only included genotype (wild-type, knock-out, heterozygote, and treated knock-out) since all other meta data categories (sex, species, etc.) were the same across samples or unknown. Differential expression p-values were calculated for each coefficient (knock-out, heterozygote, and treated knock-out) using topTable() in the limma package. Differentially expressed genes were those with an FDR adjusted p-value of less than 0.05.

WGCNA

We used Weighted Gene Co-Expression Network Analysis (WGCNA) to analyze our gene expression data. The first step in WGCNA is to input a gene expression matrix. Then, you create the symmetric gene co-expression, ‘similarity’, matrix (we used Spearman correlation as a co-expression measure). Next, you raise this matrix to a power (called the ‘soft-threshold’) to draw out the strong trends in co-expresion in the dataset. Finally, you input this matrix (called the adjacency matrix) into the Toplogical Overlap Matrix (TOM) algorithm. Each gene-pair entry of the TOM represents how similar the correlation of each gene (in the pair) with all other genes is. This measure is also called the 'connectivity' of each gene pair. This TOM undergoes average linkage hierarchical clustering to determine groups of highly connected genes, called gene modules. These gene modules represent groups of co-expressed genes. We can find the first principal component of each gene module (called the module eigengene) in the original gene expression matrix to summarize gene expression for each module. We use these modules and their respective eigengenes to look for dysregulation across genotypes and to determine which biological processes could be effected by the dysregulation we observe. See the Supplementary Figure 1 below for an overview of the WGCNA pipeline.

**Supplementary Figure 1: Basic WGCNA Pipeline**



All WGCNA calculations were done in R, using the WGCNA R package. In our WGCNA analysis, we chose a soft-threshold value of 18, since this soft-threshold value best accentuated the ‘scale-free topology’ (a small number of genes with very high connectivity) in our dataset. Our input expression matrix included 18,367 genes and our 12 samples. The following command was used to generate the TOM:

blockwiseModules(datExpr=tdatExpr, maxBlockSize = 19000,

power = 18, TOMType = "signed", minModuleSize =200,

reassignThreshold = 0, mergeCutHeight = 0.2,

numericLabels = TRUE, saveTOMs = TRUE,

saveTOMFileBase = "ArgMs\_TOM", verbose = 3, deepSplit = 4, pamStage = FALSE, networkType = "signed",

corType = "bicor", nThreads = 24)

After iterating through multiple values for minModuleSize, deepSplit, and cutHieght in order to optimize gene module creation (large and distinct modules), we settled on the following values:

minModuleSize = 200

deepSplit = 2

cutHeight = 0.1

?See supplementary figures 2-??? For dendrogram plots of WGCNA gene modules.

We obtained 20 distinct gene modules. Module eigengenes were obtained and then post-WGCNA anlaysis could begin.

Post-WGCNA Analysis

The module eigengene is the first principal component of the gene expression data within a gene module.

The following analyses were done with our gene modules and module eigengenes:

1. ANOVA test, Module Eigengene ~ Genotype
2. Boxplots of MEs to observe trends in expression across Genotype
3. Gene Ontology with GoElite on gene modules
4. pSI cell type marker enrichment in gene modules
5. Determination and subsequent investigation of hub genes (‘drivers’ of the gene module) by determining which genes in each module had the highest intra-modular connectivity (kIN) and highest correlation with the module eigengene (kME)
6. Fisher Over-Representation Analysis in gene modules of
   1. Differentially expressed genes
   2. Intellectual disability genes
   3. Blood-brain barrier CNS-specific genes
   4. Blood-brain barrier CNS developmentally up-regulated genes
   5. Blood-brain barrier CNS developmentally down-regulated genes
   6. Blood-brain barrier CNS-specific AND developmentally up-regulated genes
   7. Blood-brain barrier CNS-specific AND developmentally down-regulated genes
7. PPI enrichment in gene modules using DAPPLE

*(1) ANOVA test, Module Eigengene ~ Genotype*

An ANOVA test was done with the following linear model for each module eigengene (ME):

ME ~ Genotype

Wild-type was our baseline (wild-type coefficient = 0) group. FDR adjusted p-values were calculated from all twenty ANOVA p-values, and Tukey post-hoc p-values were calculated for gene modules with an adjusted ANOVA p-value of less than 0.05. Gene modules with an adjusted ANOVA p-value of less than 0.05 were considered differentially expressed across genotype, and the particular genotypes with a Tukey post-hoc p-value of less than 0.05 were considered the differentially expressed genotypes.

*(2) Boxplots of MEs to observe trends in expression across Genotype*

These plots simply show the distribution of the three samples from each genotype within each module eigengene. The black line in each box represents the median intensity sample. Each whisker is the lowest and highest intensity sample. Box edges extend to the 25th and 50th quantile.

*(3) Gene Ontology with GoElite on gene modules*

GoElite was used to find enrichment for gene ontology terms in our gene modules. Gene ontology terms with an enrichment z-score greater than two and a permuted p-value less than 0.05 were considered enriched terms. The top ten highest z-score gene ontology terms were chosen as the most ‘interesting’ terms and helped to guide our subsequent analyses in investigating the ANOVA-tagged differentially expressed gene modules.

*(4) pSI cell type marker enrichment in gene modules*

pSI in R was used to find enrichment of cell type markers in our gene modules. Zhang et al. mouse cortex gene expression data from isolated cell types was used to determine cell type markers. Cell type marker genes with a specificity index statistic less than or equal to 0.01 were considered marker genes. Fisher Over-Representation Analysis (ORA) was done with these pSI cell type marker genes to determine cell-type enrichments in modules. Cell-type enrichment p-values were bonferroni adjusted.

*(5) Determination and subsequent investigation of hub genes (‘drivers’ of the gene module) by determining which genes in each module had the highest intra-modular connectivity (kIN) and highest correlation with the module eigengene (kME)*

kIN, kME, and kME p-values were calculated with the following commands:

kME = signedKME(t(datExpr),MEs,corFnc="bicor")

kMEPvalue = as.data.frame(corPvalueStudent(as.matrix(kME), 12))

kIN = intramodularConnectivity.fromExpr(t(datExpr),colors,scaleByMax=TRUE,

corFnc="bicor",networkType="signed",power=18,

getWholeNetworkConnectivity = FALSE)

‘colors’ is the vector of module assignments for each gene in datExpr. The top 10-20 genes with the highest kIN and kME scores were considered ‘hub’ genes for each module. Differentially expressed modules’ hub genes were investigated in the literature to look for any relevant or interesting connections with the gene ontology we had determined or the Arg1 deficiency phenotype in general.

*(6) Fisher Over-Representation Analysis in gene modules*

Fisher’s ORA was done in R (function fisher.test) with each of the following group of genes (testing for enrichment of these groups in any of our gene modules):

1. Differentially expressed genes

* Differentially expressed genes between wild-type and knock-out mice, where genes with an FDR adjusted p-value of less than 0.05 were considered differentially expressed

1. Intellectual disability genes

* Genes associated with human intellectual disability, compiled in this Vissers et al. 2016 review

1. – (g) All Blood-brain barrier genes

* Compiled in Daneman and Barres et al. 2010, genes in the mouse blood-brain barrier transcriptome

*(7) PPI enrichment in gene modules using DAPPLE*

DAPPLE, freely available from the Broad institute, was used to look for PPI enrichment in our gene modules. A module was considered enriched for PPIs if any of its ‘NetStats’ had a p-value less than or equal to 0.05.