Published in final edited form as:

Methods Mol Biol. 2017 ; 1488: 531–549. doi:10.1007/978-1-4939-6427-7_26.

INTEGRATIVE ANALYSIS OF GENETIC, GENOMIC AND PHENOTYPIC DATA FOR ETHANOL BEHAVIORS: A NETWORK-BASED PIPELINE FOR IDENTIFYING MECHANISMS AND POTENTIAL DRUG TARGETS

James W. Bogenpohl[^], Kristin M. Mignogna⁺, Maren L. Smith⁺, and Michael F. Miles[^]

^{*}Department of Human and Molecular Genetics, Virginia Commonwealth University, Richmond, VA, 23298

⁺Department of Psychiatry, Virginia Commonwealth University, Richmond, VA, 23298

[^]Department of Pharmacology and Toxicology and the VCU Alcohol Research Center, Virginia Commonwealth University, Richmond, VA, 23298

Abstract

Complex behavioral traits, such as alcohol abuse, are caused by an interplay of genetic and environmental factors, producing deleterious functional adaptations in the central nervous system. The long-term behavioral consequences of such changes are of substantial cost to both the individual and society. Substantial progress has been made in the last two decades in understanding elements of brain mechanisms underlying responses to ethanol in animal models and risk factors for alcohol use disorder (AUD) in humans. However, treatments for AUD remain largely ineffective and few medications for this disease state have been licensed. Genome-wide genetic polymorphism analysis (GWAS) in humans, behavioral genetic studies in animal models and brain gene expression studies produced by microarrays or RNA-seq have the potential to produce non-biased and novel insight into the underlying neurobiology of AUD. However, the complexity of such information, both statistical and informational, has slowed progress toward identifying new targets for intervention in AUD.

This chapter describes one approach for integrating behavioral, genetic, and genomic information across animal model and human studies. The goal of this approach is to identify networks of genes functioning in the brain that are most relevant to the underlying mechanisms of a complex disease such as AUD. We illustrate an example of how genomic studies in animal models can be used to produce robust gene networks that have functional implications, and to integrate such animal model genomic data with human genetic studies such as GWAS for AUD. We describe several useful analysis tools for such studies: ComBAT, WGCNA and EW_dmGWAS. The end result of this analysis is a ranking of gene networks and identification of their cognate hub genes, which might provide eventual targets for future therapeutic development. Furthermore, this combined approach may also improve our understanding of basic mechanisms underlying gene x environmental interactions affecting brain functioning in health and disease.

Keywords

genomics; ethanol; alcoholism; genetics; mouse; brain; gene networks; bioinformatics

1. Introduction and background

Complex trait analysis describes the study of the vast majority of common diseases affecting humans. The most prevalent human diseases, such as heart disease, hypertension, cancer or addiction, are not caused by single genes or environmental events. Tackling the understanding of such complex diseases has been illusive given the interplay between multiple causal genetic and environmental factors. Traditional hypothesis based research, such as exploring the role of a single molecule or signaling event, has largely failed to produce breakthrough discoveries for the understanding of complex traits.

The advent of high-throughput genotyping and gene expression studies over the last 15 years has produced a deluge of data regarding complex traits. It was the hope at the advent of such technology that answers to seemingly unsolvable complex diseases would shortly be forthcoming [1]. Novel insights and non-biased hypotheses have indeed been generated though high-throughput genomic and genetic analyses. The recent discovery of over 100 loci significantly linked to the risk for schizophrenia is perhaps one of the most striking examples of such success [2]. However, even in this case, the step from genetically associated loci to causal mechanisms leading to new therapeutics is still in the distant future. Paradoxically, particularly in regard to genomic studies such as microarrays and RNA-Seq, the insights or causal relationships generated from the voluminous and complex data seem slow to come and extremely tedious to derive. Gene expression studies, particularly in exceedingly heterogeneous tissues such as the brain, may produce hundreds or thousands of genes with “significant” changes in their expression (e.g. see [3]). Approaches to derive meaning or actionable conclusions from such information have been varied but generally of low productivity in regard to the amount of effort expended. In part, this may be from an often one-dimensional approach to the analysis of complex multivariate data. Producing long lists of significant genes and then doing a simple over-representation analysis for functional biological categories has been commonly used and does not, by itself, identify potential key mechanistic points or potential drug targets. Recent application of gene network analysis, when combined with phenotypic, genetic and bioinformatics studies, has improved the identification of testable “hub genes” but expression correlations do not directly equate to causality associations.

Here we outline one approach to potentially identify causal relationships between gene networks derived from animal model studies and complex traits seen in human diseases such as alcohol use disorder. We describe the integration of genomic, genetic, and behavioral data across species to identify actionable targets underlying complex ethanol-related behaviors. While not presenting results on a single example project, we outline an approach and review existing literature for analysis of a complex phenotype such as behavioral responses to ethanol, including alcohol use disorders in humans.

2. Materials and Methods

The approach outlined below is somewhat hypothetical in terms of applicability to other experimental systems or diseases. We describe our actual procedures as performed to identify gene networks and candidate hub genes contributing to ethanol behavioral traits in animal model systems. These studies are designed to increase our understanding of the complex neurobiology underlying the disease of alcoholism in humans (now referred to as “alcohol use disorder”) and to identify potential new targets for therapeutic development. The protocol is described for studies with microarrays but RNA-seq results could be substituted with minimal change in the overall approach. We describe key elements in deriving gene networks from microarray genomic data and give several examples of how bioinformatics tools can be used to superimpose external database information onto derived networks so as to increase the likely yield of biologically informative networks. We do not discuss details of RNA isolation and generating probes for microarray analysis as these are described extensively elsewhere in the literature.

A. Experimental design

The approach discussed below assumes that an appropriate experimental design has been employed such that ethanol behavioral data and microarrays have been gathered on the same individual animals. Alternative designs are possible, but the gene network analysis by WGCNA (see below) requires sufficient numbers of samples (e.g. $n = 35$) to drive the formation of robust correlations across the network [4]. Thus, this method requires either sufficient numbers of animals from a single strain but across multiple doses/times (e.g. see [5]) or behavioral and genomic data across a genetic resource such as the BXD recombinant inbred panel derived from C57BL/6J (B6) and DBA2/J progenitors (e.g. see [3]). The goals with such designs are to ensure adequate variance in the data is derived from the biological phenotypes of interest and to directly correlate gene expression and behavior from the same animals. Extensive examples of ethanol behavioral studies (e.g. locomotor activity, anxiolytic-like behaviors, loss-of-righting reflex, withdrawal-induced seizures or ethanol consumption) across BXD strains are found in the GeneNetwork (www.genenetwork.org). Since analysis of behavioral genetic data within GeneNetwork is described elsewhere within this volume, we do not discuss it further here.

Deriving genomic and behavioral data across a large number of subjects is technically demanding and requires extensive planning in terms of experimental design so as to avoid uncontrolled or unknown environmental sources of variance in the data [6]. Such strategies are discussed further within the Notes section below. The keys to avoiding such “batch effects” are to avoid fluctuations in procedures (e.g. different staff doing behavioral or molecular analyses) and to perform a supervised randomization of the handling and treatment of animals and the processing of samples for RNA isolation and genomic analysis as much as possible. However, even with such measures, the possible presence of batch effects must be assessed and eliminated as described below.

B. Quality control and removal of batch effects from microarray data using ComBat

As mentioned above, one common challenge in analyzing genome-wide expression data (or any other large set of biological data) is dealing with batch effects. Non-biological or uncontrolled experimental variation is often introduced by the serial processing of batches of biological samples, or it can also be an issue when analyzing data collected across multiple experiments. In the case of large microarray experiments, factors could include sacrifice of animals at different times, sequential isolation of RNA in batches, or the hybridization and processing of arrays in batches. Even when the same experimenter has conducted experiment batches with the same equipment, subtle factors such as the age of reagents or the temperature of the laboratory can unknowingly introduce non-biological variation into the genomic data collected. Despite extreme care, even expert experimentalists can introduce non-biological variance into whole-genome expression data without their knowledge. For this reason, it is necessary to always assess for batch effects whenever completely parallel processing of samples is not possible.

The first step in eliminating batch effects is identifying them. Principle component analysis is an excellent way to do this and can be easily performed using the ggplot2 package for R (<https://cran.r-project.org/web/packages/ggplot2/index.html>). One must first identify all factors at which sample processing or data collection was possibly performed in a non-parallel manner. Once such factors have been identified, a simple and effective way to eliminate potential batch effects from microarray data is through use of the ComBat script for R (R Project for Statistical Computing <http://www.r-project.org/>). The ComBat method has been described in detail in a published report [7] and the software can be downloaded for free at <http://www.bu.edu/jlab/wp-assets/ComBat/Download.html> as a stand-alone or as part of the SVA package in the Bioconductor project (<http://www.bioconductor.org/packages/release/bioc/html/sva.html>) within the R statistical framework. The practical use of this method as a stand-alone is described below.

Protocol for PCA analysis and ComBat

1. Create a spreadsheet in which the first column is an identifier for each microarray, and each subsequent column contains information grouping the arrays into the experimental batches that were used in all non-parallel steps. You can also add columns containing other information, such as the binned age or weight of the animals used, cage location or perhaps their sex or treatment group. Each column should have a header describing the information within. Save this file as a tab delimited text file.
2. A second spreadsheet will contain the microarray expression data. The first column should contain the probe/gene IDs, and each subsequent column should contain the expression data for each microarray. Save this file as a tab delimited text file. Generally speaking, this array data has already been assessed for routine individual technical outliers (as may be caused by a degraded RNA sample) by use of standard low level microarray analysis metrics.

3. Run the following commands in R to perform principle component analysis. Lines marked with # indicate annotation lines rather than commands. This will create a .tiff image file containing a scatter plot of the first two principle components of your expression data, with each data point representing one array, colored by its batch. The below block of script is for coloring the data points by array batch, and in the hypothetical batches spreadsheet, the column header containing that information would be called “array.batch”. Repeat this block of code for each column in your batches spreadsheet, replacing “array.batch” with the relevant column header. The plots created will look similar to Figure 1A.

```
> library(ggplot2)
> expressiondata=read.csv("expressionspreadsheet.txt",
sep="\t")
> rownames(expressiondata)=expressiondata[,1]
> expressiondata=expressiondata[,-1]
#This calls the ggplot2 package and loads your expression
data. Bold italic text will be
#changed to the file name of your particular expression data
spreadsheet.
> pca=prcomp(t(expressiondata), scale=T)
> arraydata=read.csv("batchespreadsheet.txt",sep="\t")
#This runs the principle component analysis of your
expression data and loads your
#batch information spreadsheet, the name of which will
replace the bold italic text.
> array.batch=arraydata$array.batch
> scores=data.frame(array.batch, pca$x[,1:3])
> tiff("PCA by array batch.tiff", width=750, height=750)
> p=qplot(x=PC1, y=PC2, data=scores,
color=factor(array.batch), main="PCA plot by array batch")
> p + theme(text = element_text(size = 20)) + geom_point(size
= 5)
> dev.off()
```

4. Look at your principle component plots and determine which batch effects need to be corrected by examining the data structure. Ideal data structure looks like a homogenous, somewhat circular cloud of data points, as shown in panel B of the figure. Any grouping of data points apart from the whole should be addressed. Look for the batch category that best identifies each segregating group of data points by color.
5. Once it has been determined which batch factor needs to be addressed, create a sample information spreadsheet where the first column is headed “Array name” and contains the identifiers of your microarrays, and a

second column headed “Batch” which contains the relevant batch groupings. Do not put any extraneous information in this file. Save this spreadsheet as a tab delimited text file.

6. Run the following script to use ComBat for correcting an identified batch effect in your data. Make sure that the ComBat.R file you downloaded is in your working directory.

```
>Source( "ComBat.R" )
>ComBat( "expressionspreadsheet.txt",
"sampleinfospreadsheet.txt" )
```

The bold italic text should be replaced with the filenames of your expression data spreadsheet from step 2, and your sample information spreadsheet from step 5. A new file containing your adjusted data will automatically be written to your working directory. Instructions for more advanced usage of ComBat, as well as troubleshooting information can be found at <http://www.bu.edu/jlab/wp-assets/ComBat/Usage.html>.

7. Run principle component analysis again, repeating step 3 but substituting the adjusted data file name for the original expression data file name. Also, change the name of the output .tiff file so that your original plot is not overwritten. Ensure that the data structure is homogeneous, as shown in Figure 1B.

C. Network identification with weighted gene correlation network analysis

While multiple methods exist for clustering or network analysis of genomic data, the Weighted Gene Correlation Network Analysis algorithm (WGCNA) is a very widely used R software package to identify groups, known as modules, of correlated genes within microarray or other suitable data (Zhang and Horvath 2005). WGCNA is based on scale-free network topology, a model system that assumes a small number of highly connected nodes within a network. For transcriptomic data these nodes are referred to as ‘hub genes’. Due to their high connectivity, hub genes represent potential therapeutic targets to affect ethanol responsive gene expression in the brain and, potentially, ethanol behaviors. WGCNA involves multiple analysis steps that are outlined in detail in a series of R tutorials produced by the Horvath laboratory. These extensive tutorials and associated papers applying WGCNA are a major reason for the popularity of this approach. Instructions for obtaining the WGCNA R package and dependencies can be found at: <https://labs.genetics.ucla.edu/horvath/CoexpressionNetwork/Rpackages/WGCNA/>. A brief overview of a sample WGCNA analysis of microarray data from ethanol-treated mouse brain follows but the actual commands and options for the analysis can be found within the tutorials from the Horvath web site. Quite often, simple copy/pasting of those commands with changing file names or parameters is sufficient for carrying out an initial WGCNA analysis.

Protocol overview for WGCNA

1. An appropriate dataset must be chosen for WGCNA analysis. As with any clustering technique, it is essential to have substantial biological variation across samples and to have enough samples such that correlation networks have sufficient statistical power. Although firm thresholds are difficult to define, Iancu and colleagues, using a large microarray dataset from mouse striatum, determined that a $n = 35$ appeared optimal for defining network topology [4].
2. In order to identify networks with the most meaningful correlations, a variance filter is often first applied to e.g. a ComBat-corrected microarray dataset. This variance filter eliminates genes showing minimal variation in expression across samples. The median absolute deviation (MAD) is our preferred method for variance quantification:

$$MAD = \text{median}_i (|X_i - \text{median}_j (X_j)|).$$

A histogram of MAD values is plotted to identify the lower tail of variance. The number of genes included in network analysis may be limited by computer power. However, ideally, the proportion of variance should be calculated at regular intervals to determine a MAD threshold at which the majority of variation within the total data is included. All data below this threshold may be assumed to be noise, and excluded from further analysis.

3. The next step in WGCNA involves uploading gene expression data, and modeling to determine a soft-thresholding power at which the data structure best fits scale-free topology. This is done using the `pickSoftThreshold()` function as part of the WGCNA package. This function will output a table of scale-free fit metrics at various soft-thresholding powers (Table 1). We use the scale-free fit index ($SFT R^2$) as our primary measure of scale-free fit. A scale-free of 0.9 or greater is ideal, however, a scale-free fit of 0.75 or greater can be acceptable.
4. Network construction is the next step in WGCNA. Both manual and automatic network construction and module identification are outlined, in detail, in the WGCNA tutorials. During module identification, gene expression data is organized by topological overlap distance. This data can be visualized using a cluster dendrogram. One particular variable to pay attention to during module construction is the “deep-split”. Deep-split is used to fine-tune the sensitivity of module detection by adjusting the branch cutting threshold within the dendrogram. Multi-dimensional scale plots using first and second principal components as the x and y-axes are another useful way to visualize modules in order to identify optimal deep-split value. We consider an optimal deep-split value to be one where there is minimal spatial overlap between modules.

5. Both modules and individual genes can then be correlated to phenotype data. Individual gene correlations are performed using gene expression measures such as RMA values. Modules are correlated to phenotype based on their first principal component, known in WGCNA as the module eigengene. The module eigengene is a value that explains the majority of gene expression variance within each identified module. Phenotype data can include many variables from behavioral data to technical variables such as RNA quality index of each microarray sample. This network correlation analysis is one of the most powerful features of WGCNA. Ideally, phenotypic and genomic data are derived from the same individual animals. Due to the limited sample size often seen in microarray studies, we recommend using Spearman Rank rather than Pearson correlation in order to minimize the influence of outliers. An example of network correlations to phenotypic data is shown in Figure 2.
6. The identification of hub genes within modules is one of the final steps in WGCNA. Connectivity is usually the primary variable by which we identify hub genes. WGCNA produces several connectivity metrics with the command function: `intramodularConnectivity()`. For each gene, this command outputs its total connectivity within the network (`kTotal`), its connectivity within its assigned module (`kWithin`), the difference between `kTotal` and `kWithin` (`kOut`), and the difference between `kWithin` and `kOut` (`kDiff`). The `intramodularConnectivity()` function features an option to scale within module connectivity (`kWithin`) based on module size. We recommend scaling, as this eases identification of the most highly connected genes within their respective modules independent of module size.
7. Modules identified by WGCNA can be further interrogated for biological function enrichment using a wide variety of standard bioinformatics tools such as those for Gene Ontology or pathway analysis. Our laboratory frequently uses free web-based resources such as DAVID (<http://david.abcc.ncifcrf.gov>), ToppGene (<https://toppgene.cchmc.org>), WebGestalt (<http://bioinfo.vanderbilt.edu/webgestalt/>) or REVIGO (<http://revigo.irb.hr>) to identify and display over-represented functional categories. Additionally, module gene lists can be submitted to resources that construct networks based upon other biological information such as protein-protein interactions, published biological interaction, or transcription factor binding site analyses. Such tools include GeneMANIA (<http://genemania.org>) and the subscription-based Ingenuity Pathway Analysis (<http://www.ingenuity.com>). These tools can, in effect, validate the network structure of WGCNA-derived expression correlation networks.
8. WGCNA networks can be further validated and ranked based upon quantifying their overlap with other user defined gene lists obtained from differing biological contexts or from expression genetics datasets. For

example, we frequently interrogate WGCNA modules for overlap with public or our own gene sets within the GeneWeaver web-based application (www.geneweaver.org). Finally, WGCNA module genes can be interrogated for correlations in other expression datasets, phenotypic correlations, and conserved genetic regulators by the rich resources available within GeneNetwork (www.genenetwork.org).

D. Combined expression, protein-protein interaction and genetic networks using dmGWAS3

An ideal bioinformatics methodology for interrogating the significance and biological function of genomic expression data would be to superimpose other types of gene-gene interaction data or biological significance scores onto expression correlation networks. WGCNA, as described above, only produces expression correlation networks and phenotype/expression correlation. Several tools have been developed, however, to directly superimpose multiple types of biological interactions. These include GeneMANIA, mentioned above, where gene lists can be constructed into network via a combined scoring algorithm entailing mining multiple databases such as protein-protein interactions, other microarray expression correlations, and literature association. An additional resource whereby gene expression correlations can be combined with protein-protein interaction networks and genetic significance scores (such as p-values from human GWAS data), is the “edge-weighted dense-module searching of Genome Wide Association Studies” algorithm, or EW_dmGWAS, developed by Jia and colleagues at Vanderbilt University [8,9].

EW_dmGWAS (referred to henceforth as dmGWAS) is a method that allows integration of gene expression data, genotypic data, and protein-protein interaction data to find gene networks that are associated with complex traits. The tool seeks to solve the two simultaneous problems generated by under-powered and complex genomic or genetic datasets. First, is there an underlying biological structure to GWAS results, including those not reaching genome-wide significance levels? Second, are expression correlation networks (particularly those from model organisms) relevant to genetic loading in complex traits? Using gene pairs from a predefined protein-protein interaction (PPI) background network, the program forms edge weights between two genes, by comparing the correlation in expression between the two genes in the control versus the treated sample. Correlation changes between the two conditions result in increased weighting, thus informing the structure of the network for the biological trait of interest. dmGWAS then uses these values, in conjunction with GWAS p-values, to determine whether or not genes are part of the same network. The importance attributed to GWAS p-values, relative to that attributed to expression correlation differences, is determined by the parameter *lambda*, which can be user defined or estimated by the program. The estimation algorithm chooses a value that will attribute equal importance to each set of values. The program also allows the user to define the stringency of the inclusion choice (as determined by parameter *r*).

The program starts with a seed gene as a module, and expands each module by adding genes that are associated with the current module in the PPI network and increase the module score by a factor of *r*, one at a time. Each time a gene is queried, a new module score is calculated

from its node weight (GWAS p-value) and the weight (correlation difference) of each of its edges shared with genes already included in the module, using *lambda* to determine the relative reliance on node-weights and edge-weights. The expansion process is terminated when it reaches the point at which no additional genes increase the module score by a factor of *r*. The final module score is then standardized via z-score transformation with respect to a distribution of scores of 10,000 random modules of the same size. These scores can then be used to calculate the significance of each module. The resulting modules are networks of genes (nodes) whose protein products interact, have correlations (edges) differing between treatment and control groups, and have nodes enriched for genes with relatively low GWAS p-values. An example of results is shown in Figure 3.

Protocol overview for dmGWAS analysis

1. *Formatting genome wide association study (GWAS) data.* For this step, a human GWAS results file is needed that contains SNP names and raw p-values for the association of each SNP with a trait of interest. Because the nodes of the dmGWAS network will represent genes, as opposed to SNPs, gene-wise p-values need to be calculated from the raw SNP p-values. This can be accomplished by using programs like VEGAS2 (Versatile Gene-Based Association Study) [10] or KGG (Knowledge-based mining system for Genome-wide Genetic studies) [11]. A data frame should result that contains one column of human gene symbols and one column for gene-wise p-values.
2. *Formatting microarray data.* Data from of microarray results are needed for calculating edge weights between nodes (genes) in the dmGWAS networks. We typically employ RMA (Robust Multichip Average) [12] or similar expression values, obtained from tissue with a treatment and control group, and gene symbols associated with each probeset ID. The data used for the example network displayed in Figure 3 was obtained from a genomic study of acute ethanol (1.8 g/kg x 4 hours) expression responses in prefrontal cortex of BXD mouse strains [3]. Because the GWAS data contains human gene symbols, expression data gene symbols (mouse) were converted to human using resources such as SwissProt (<http://www.ebi.ac.uk/uniprot>). The mouse gene symbols and their human homologues are then merged with the microarray data frame by probeset ID. The resulting data frame should contain a column for each of the following: human gene symbols, mouse gene symbols, probeset IDs, and RMA values.
3. *Filtering microarray data.* In forming edge weights between two genes, dmGWAS compares the correlation in expression between the two genes in the control versus the treated (ethanol) sample. Some gene symbols in the microarray dataset will likely have multiple RMA value data values due to redundancies in probeset design. We thus reduce this complexity by using the probeset for a given gene having the largest average expression value across all microarray samples. Low expression value probesets/

genes are not removed from the array dataframe since they will not contribute appreciably to edge weighting although the cognate nodes may be vital to the formed dmGWAS network if they have high node values (low GWAS p-values). The filtered expression data frame should contain a column for each of the following: probesetIDs, mouse gene symbol, human gene symbol, and RMA expression values for each sample. Separate data frames will be needed for the control group and treatment group.

4. *Formatting protein-protein Interaction network data.* A predefined protein-protein interaction (PPI) network is used as a background network for the dmGWAS analysis. Although multiple such resources exist, the Protein Interaction Network Analysis (PINA) platform contains multiple curated public databases and interactions for human and five model organisms [13]. When using human GWAS data, a human PPI network is utilized. The data frame should contain two columns of protein IDs, with each row representing a pair-wise interaction. Protein IDs must be converted to human gene symbols to be matched with data in the gene expression and GWAS datasets. Such gene symbol cross-annotation data can be obtained from SwissProt (<http://www.ebi.ac.uk/uniprot>). The resulting final PPI data frame should contain two columns: one column for the gene symbol of the first member of an interaction pair, and a second column for gene symbols associated with the second member of each PPI pair.
5. *Checking data format for analysis.* Before you can run a dmGWAS analysis, it is critical for the data to be formatted in a specific manner, in order for the algorithms in the package to work. Four data frames are utilized by the dmGWAS command `dms()`, which will run the dense module search and output the result objects: GWAS, the two expression datasets, and the PPI data frame.
 - The GWAS data frame should contain one column of human gene symbols and one column of gene-wise GWAS p-values, in that order. This data frame will be entered in the “geneweight” argument of the dmGWAS command `dms()`.
 - The “exp1” and “exp2” arguments will take the treatment and the control expression data frames, respectively. The final data frames must contain one column of human gene symbols followed by n (number of samples) columns of RMA values associated with each gene.
 - The PPI network data frame must have two columns, with each row containing the human gene symbols associated with each protein in a pair-wise interaction. The number of rows should equal the number of pair-wise interactions in

your dataset. This data frame will be entered in the “network” argument.

- The program will discard information on any genes that are not present in all four data frames. Therefore, a gene must be present in all four datasets in order to be included in the analysis.

6. *Run the dms() command, insert the aforementioned data frames in their respective arguments, and define r and lambda. If you want to use the default r value (0.1), do not include the “r” argument in the command. If you want the program to estimate the lambda parameter, use lambda=“default”. Store the run as an object. This will save all of the output objects as a list. An example of the script to run a dmGWAS analysis and save output is thus:*

```
#Run dense module search and store results as dataframe
library(igraph)
library(dmGWAS)
library(tcltk)
res.list<-dms(network, geneweight, expr1, expr2, r=0.1,
lambda="default")
#res.list<-dms(network, geneweight, expr1, expr2, r=0.1, d=2,
lambda="default") #default is d=1
names(res.list)
save(res.list, file="Results_dmGWAS_filename.Rdata")
#GWPI: igraph class object, node and edge-weighted PPI network
#genesets.clear: list containing all valid modules; name of
each record=seed gene
#genesets.length.null.dis: list containing the randomization
data for each size of module
#genesets.length.null.stat: list containing the statistical
values of randomization data for each size of module
#module.score.matrix: matrix containing data for each module;
data = gene (seed gene), Sm (module score), Sn (normalized
module score)
#ordered.module.score.matrix: ordered matrix of
module.score.matrix based on Sn
#select the top (100x"top")% of modules with respect to
significance, and store as a dataframe
selected<-moduleChoose(res.list, top=0.005, plot=T)
Top0.5<-selected
save(Top0.5, file="TopPoint5Percent.Rdata")
```

7. *Reading the results.* The results list contains an object for each of the following: igraph object of your input, the resulting module names (names of the seed genes) and their raw and standardized scores, and the raw scores of the permuted random modules. If you use the dmGWAS_3.0 package, the lambda value will be stored in the name of a separate object that will be saved to your working directory automatically. This package will also output an object of module names and scores that is ordered by the standardized scores. The modules may overlap significantly. Identical modules are removed by the package, but no modules are merged based on overlap. It is up to the user to merge overlapping modules, if desired. Because the permuted module scores compose an approximately normal distribution, the standardized module scores can be treated as Z-scores, and therefore may be used to calculate p-values that represent module significance. If you choose to merge any modules, you must recalculate a standardized score based on a permutation of modules of the new merged module's size, in order to calculate its significance.

3. Results

The elements of a data analysis pipeline that are outlined in Methods will produce well normalized data suitable for network analysis by WGCNA and downstream bioinformatics studies, such as those produced by dmGWAS. The intended result of such studies is to identify networks of genes that can be ranked by their biological functions and perhaps by over-representation with other relevant genomic data or human GWAS results. The latter could greatly increase the biological value for both a given network and cognate “hub genes”, producing better and fewer targets for complicated and time-consuming validation in animal models (e.g. by gene targeting or pharmacological interventions) – with the eventual goal of defining candidates for new therapeutic measures in humans.

The importance of experimental design and eliminating batch effects is clearly demonstrated in Figure 1. Prior to ComBat normalization, the principle component analysis indicates at least three groupings of arrays within the data that do not correspond to known desired dependent variables (Figure 1A). In this case, these were produced by differing batches of animals in an experiment conducted over an extended period of time. Following ComBat correction for such a batch effect, the microarray data is much more evenly distributed with little apparent subdivision by any known dependent variable (Figure 1B).

The generation of scale-free networks by WGCNA analysis is a powerful tool for defining functional and regulatory groupings within complex sets of genomic data. The multi-step process that is well described by protocols from the Horvath laboratory produces modules that are logical choices for downstream interrogation in bioinformatics and biological validation experiments. Figure 2 demonstrates the power of this approach by displaying a module with remarkable correlation between the relative “connectivity” of genes within the module (x-axis) and the correlation of the expression of these genes with a theoretical quantitative behavioral phenotype of interest (y-axis). When such a striking correlation between module membership and trait correlation is seen, it serves to validate the module as

intimately involved in the phenotype of interest. More importantly, this analysis also portrays robust candidate genes for future studies, as being those in the top right corner of the Scattergram in Figure 2 – with the strongest intramodular correlations and correlation to the phenotype.

Finally, our laboratory has used genomic studies in mouse models for nearly 20 years as an effort to increase our mechanistic understanding of ethanol-responsive behaviors relevant to the development of alcohol use disorder in humans. The defining of ethanol-responsive gene networks in mouse models, however, does not necessarily directly translate to a role in human disease. Thus, the importance of methodologies such as dmGWAS is that they can directly superimpose animal model genomic data with human genetic results, within a known framework of protein interactions. As shown in Figure 3, such studies may therefore highlight particular aspects of networks and hub genes that have predictive validity for human disease.

4. Notes

1. The description on use of ComBat for correction of possible batch effects is brief and does not include the full details of the R module. Readers are referred to the source paper and the R module information for further information on the use of co-variables and other factors.
2. ComBat should be used with great care and only after extensive characterization of quality control measures across arrays or RNA-seq samples. RNA samples should have uniformly high quality measures such as RNA integrity numbers (RIN) on bioanalyzer studies prior to genomic analysis. Outliers can greatly modify the results of ComBat corrections and should be eliminated from the analysis if possible. Correction for an independent variable that seems unrelated to the dependent variable of interest may serve to eliminate expression patterns actually related to the trait of interest. For example, ComBat correction for age differences amongst experimental (male) animals might eliminate brain gene expression patterns related to testosterone levels.
3. WGCNA analysis can be greatly affected by inclusion of genes having marginal expression levels or no variance across the experimental samples. The trimming protocols that are suggested are thus critical to downstream analysis. Furthermore, efforts to decrease the number of genes included in the WGCNA analysis (for increasing processing speed) should be done with care since use of biased filters (such as statistical cutoffs) could significantly disrupt intramodular connectivity in some cases. A highly “responsive” network might actually disappear due to the lack of a few highly connected, but “unresponsive”, hub genes.
4. WGCNA module eigengene x phenotype correlations should only be done for phenotypes with sufficient numbers of subjects and data distribution. Correlation of continuous module eigengenes with discontinuous or

ordinal traits can produce markedly significant correlations of unclear significance.

5. In dmGWAS analysis, do not filter out genes based on GWAS p-values. dmGWAS is designed to incorporate the p-value of each gene in the decision to include or exclude the gene from a module. It is possible for a gene to play an important role in relevant networks, although genetic variation, specifically, may not contribute to the gene's association with the network. Protein-protein interactions and expression correlation changes might be more important drivers for inclusion of such a gene in a given network. Furthermore, the genetic variation in other genes associated with the same pathway may confer the relevance of the overall network. If such a gene with a poor GWAS p-value is central to the overall connectivity of a network, the module may be excluded from ranked results if the gene is excluded from the analysis.

Acknowledgments

This work was supported by grants from the National Institute on Alcohol Abuse and Alcoholism to Michael F. Miles (U01AA016667, P50AA022537) and Maren L. Smith (F31AA023134). Kristin M. Mignogna was partially supported by the VCU C. Kenneth and Dianne Wright Center for Clinical and Translational Research and a grant from the National Center for Advancing Translational Sciences (UL1TR000058). James W. Bogenpohl was supported by an Institutional Training Grant in Digestive and Liver Diseases from the National Institute of Diabetes and Digestive and Kidney Diseases (5T32DK007150).

References

1. Hughes TR, Marton MJ, Jones AR, Roberts CJ, Stoughton R, Armour CD, Bennett HA, Coffey E, Dai H, He YD, Kidd MJ, King AM, Meyer MR, Slade D, Lum PY, Stepaniants SB, Shoemaker DD, Gachotte D, Chakraburtt K, Simon J, Bard M, Friend SH. Functional discovery via a compendium of expression profiles. *Cell*. 2000; 102(1):109–126. [PubMed: 10929718]
2. Loh PR, Bhatia G, Gusev A, Finucane HK, Bulik-Sullivan BK, Pollack SJ, de Candia TR, Lee SH, Wray NR, Kendler KS, O'Donovan MC, Neale BM, Patterson N, Price AL. Schizophrenia Working Group of the Psychiatric Genomics C. Contrasting genetic architectures of schizophrenia and other complex diseases using fast variance-components analysis. *Nat Genet*. 2015; 47(12):1385–1392. DOI: 10.1038/ng.3431 [PubMed: 26523775]
3. Wolen AR, Phillips CA, Langston MA, Putman AH, Vorster PJ, Bruce NA, York TP, Williams RW, Miles MF. Genetic dissection of acute ethanol responsive gene networks in prefrontal cortex: functional and mechanistic implications. *PLoS One*. 2012; 7(4):e33575.doi: 10.1371/journal.pone.0033575 [PubMed: 22511924]
4. Iancu OD, Darakjian P, Malmanger B, Walter NA, McWeeney S, Hitzemann R. Gene networks and haloperidol-induced catalepsy. *Genes Brain Behav*. 2012; 11(1):29–37. DOI: 10.1111/j.1601-183X.2011.00736.x [PubMed: 21967164]
5. Smith ML, Lopez MF, Archer KJ, Wolen AR, Becker HC, Miles MF. Time-Course Analysis of Brain Regional Expression Network Responses to Chronic Intermittent Ethanol and Withdrawal: Implications for Mechanisms Underlying Excessive Ethanol Consumption. *PLoS One*. 2016; 11(1):e0146257.doi: 10.1371/journal.pone.0146257 [PubMed: 26730594]
6. Chesler EJ, Wilson SG, Lariviere WR, Rodriguez-Zas SL, Mogil JS. Identification and ranking of genetic and laboratory environment factors influencing a behavioral trait, thermal nociception, via computational analysis of a large data archive. *Neurosci Biobehav Rev*. 2002; 26(8):907–923. S0149763402001033 [pii]. [PubMed: 12667496]

7. Johnson WE, Li C, Rabinovic A. Adjusting batch effects in microarray expression data using empirical Bayes methods. *Biostatistics*. 2007; 8(1):118–127. DOI: 10.1093/biostatistics/kxj037 [PubMed: 16632515]
8. Jia P, Zheng S, Long J, Zheng W, Zhao Z. dmGWAS: dense module searching for genome-wide association studies in protein-protein interaction networks. *Bioinformatics*. 2011; 27(1):95–102. DOI: 10.1093/bioinformatics/btq615 [PubMed: 21045073]
9. Wang Q, Yu H, Zhao Z, Jia P. EW_dmGWAS: edge-weighted dense module search for genome-wide association studies and gene expression profiles. *Bioinformatics*. 2015; 31(15):2591–2594. DOI: 10.1093/bioinformatics/btv150 [PubMed: 25805723]
10. Mishra A, Macgregor S. VEGAS2: Software for More Flexible Gene-Based Testing. *Twin Res Hum Genet*. 2015; 18(1):86–91. DOI: 10.1017/thg.2014.79 [PubMed: 25518859]
11. Gui H, Li M, Sham PC, Cherny SS. Comparisons of seven algorithms for pathway analysis using the WTCCC Crohn's Disease dataset. *BMC Res Notes*. 2011; 4:386.doi: 10.1186/1756-0500-4-386 [PubMed: 21981765]
12. Irizarry RA, Bolstad BM, Collin F, Cope LM, Hobbs B, Speed TP. Summaries of Affymetrix GeneChip probe level data. *Nucleic Acids Res*. 2003; 31(4):e15. [PubMed: 12582260]
13. Cowley MJ, Pinese M, Kassahn KS, Waddell N, Pearson JV, Grimmond SM, Biankin AV, Hautaniemi S, Wu J. PINA v2.0: mining interactome modules. *Nucleic Acids Res*. 2012; 40(Database issue):D862–865. DOI: 10.1093/nar/gkr967 [PubMed: 22067443]

Figure 1. Principle component plots showing a batch effect of array processing batch (A), which is corrected following ComBat (B).

Figure 2.

Module membership correlation with gene significance. WGCNA analysis plot of member genes from one module. X-axis is module membership scoring where higher values represent genes with greater connectivity. Y-axis shows gene significance in terms of correlation of expression values versus a trait of interest. Genes with expression more highly correlated with trait of interest and showing higher connectivity (top right corner) are high value “hub genes”. High correlation of the module membership with gene significance strongly suggests this module is involved in biological mechanisms of the trait.

Table 1

Scale-free metrics from WGCNA analysis

Scale free metrics resulting from the function *pickSoftThreshold()* within WGCNA. Results show that a power of 5 results in acceptable scale-free fit index values (SFT.R.sq).

Power	SFT.R.sq	Slope	Truncated.R.sq	Mean.k.	Median.k.	Max.k.
1	0.028	0.345	0.456	746.978	761.733	1206.429
2	0.126	-0.597	0.843	254.498	250.843	573.545
3	0.340	-1.030	0.972	111.005	101.736	324.154
4	0.506	-1.422	0.973	56.536	47.248	202.307
5	0.681	-1.716	0.940	32.154	25.089	134.175
6	0.902	-1.497	0.962	19.906	14.498	94.774
7	0.921	-1.667	0.917	13.192	8.678	84.103
8	0.904	-1.724	0.876	9.249	5.394	76.315
9	0.859	-1.703	0.836	6.795	3.556	70.532
10	0.833	-1.663	0.831	5.194	2.382	65.752