dmGWAS_3.0: Edge-weighted dense module search for genome-wide association studies and gene expression profiles

Quan Wang[1], Hui Yu[1], Zhongming Zhao[1,2,3,4] and Peilin Jia[1,2]

[1]Department of Biomedical Informatics, [2]Center for Quantitative Sciences, [3]Department of Psychiatry, [4]Department of Cancer Biology, Vanderbilt University, Nashville, TN 37232, USA

February 16, 2015

1. Introduction

Over the past decade, genome-wide association studies (GWAS) have successfully uncovered many susceptibility loci for common diseases. However, the identified loci only explain a small portion of genetic risks (Jia, et al., 2011). It is challenging to uncover the remaining risky loci as their association signals are likely to be moderate or weak. One potential solution to this challenge is to incorporate other functional information, such as protein-protein interaction (PPI) or gene co-expression network, to investigate joint association signals beyond single markers (Jia, et al., 2012).

We have previously developed a network-assisted approach, dmGWAS, to address this problem (Jia*, et al.*, 2011). dmGWAS applies a greedy algorithm to search for dense modules in a PPI network that is node-weighted by GWAS signals. After its initial release, dmGWAS received much attention from the community and has become a popular tool for network-assisted GWAS analysis. However, dmGWAS ignores the edge information of the PPI network when expanding modules.

Therefore, we introduce an upgraded algorithm, EW_dmGWAS, to boost GWAS signals in a node- and edge-weighted PPI network. Differential gene co-expression (DGCE) is important genetic information, reflecting cellular dynamics and contributing to pathogenesis (Hou, et al., 2014). We thus utilize DGCE (i.e., the change of gene co-expression between case and control samples) in EW_dmGWAS to infer the weight of each edge and combine the association signals of its two nodes to assess the overall disease risk of network modules within the human PPI network. Our previous dmGWAS approach is implemented as R package ***dmGWAS_1.0*** and ***dmGWAS_2.X***. We thus continuously implement the algorithm of EW_dmGWAS as an R package named by ***dmGWAS_3.0***. ***dmGWAS_3.0*** has also been updated to cooperate with all the newest version of its depending packages, such as 'igraph', and is available at http://bioinfo.mc.vanderbilt.edu/dmGWAS.

2. Methods

In summary, EW_dmGWAS integrates GWAS signals and gene expression profiling to extract dense modules from the background PPI network. Node weights are derived from GWAS and edge weights are derived from gene expression profiling. The module score is a combination of node weight and edge weight. The aim of EW_dmGWAS is to identify modules with locally maximum scores. The workflow of EW_dmGWAS is described as follows.

Step 1. Defining node weight

Node weights are determined by the GWAS signals. EW_dmGWAS first map SNP p-values from GWAS onto gene-based p-values. Multiple tools are available for this step (Ballard*, et al.*, 2010). We recommend the convenient tool VEGAS (Liu*, et al.*, 2010). VEGAS applies the simulation from multivariate normal distribution to account for linkage disequilibrium. The gene-based p-values estimated by VEGAS are approximately uniformly distributed, and thus are appropriate to be transformed into standard normal distribution. We defined node weight by $nodeweight(v) = \varphi^{-1}(1-p)$, where p denotes the gene-based p-value of node v, and φ is the standard normal distribution function.

Step 2. Defining edge weight

We used the change of gene co-expression between case and control samples to infer edge weight. Specifically, let r_{case} and $r_{control}$ represent the Pearson's correlation coefficient (PCC) of gene expression in both case and control samples, and n_{case} and $n_{control}$ represent the sample size respectively. We first used the Fisher transformation [Equation (1)] and then Fisher's test of difference between two conditions [Equation (2)] to define a new statistic X:

$$F(x) = \frac{1}{2} ln \frac{1+x}{1-x}, \tag{1}$$

$$X = \frac{F(r_{case}) - F(r_{control})}{\sqrt{\frac{1}{n_{case}-3} + \frac{1}{n_{control}-3}}}. \tag{2}$$

The newly defined statistic X approximately follows standard normal distribution (Hou*, et al.*, 2014). We then defined edge weight as $edgeweight(e) = \varphi^{-1}\left[2 * \left(1 - \varphi(|X|)\right)\right]$.

Step 3. Defining module score

To quantitatively evaluate the density of high-weight nodes and edges within a module, we defined the module score S by

$$S = \lambda \frac{\sum_{e \in E} edgeweight(e)}{\sqrt{\#\ of\ E}} + (1-\lambda) \frac{\sum_{v \in V} nodeweight(v)}{\sqrt{\#\ of\ V}}, \tag{3}$$

where E and V represent the edges and nodes in the module respectively, and λ is a parameter to balance GWAS and gene expression signals.

Step 4. Module search

We performed a greedy algorithm to search for dense modules as follows.

(1) Assign a seed module M and calculate the module score S_m of M. At first, the seed module is a single gene.
(2) Examine the first order neighbors of M, and identify the neighbor node N_{max} that generates the maximum increment of module score.
(3) Add N_{max} to the current module M if the score increment is greater than $S_m \times r$, where r is a parameter to decide the magnitude of increment.
(4) Repeat steps 1-3 until no more neighbors can be added.

Step 5. Normalization of module score

In order to evaluate the significance of the identified modules, we used a randomization method to obtain the background distribution of module score. Specifically, for a module M with K nodes, we randomly generated a sub-network with the same size, and calculated the score $S_m(\pi)$ of this sub-network. We repeated this process 10,000 times and denoted the mean and standard deviation of $S_m(\pi)$ by μ and σ. The module score was normalized by $S_n = (S_m - \mu)/\sigma$, and S_n was used to determine the significance of the identified modules.

3. Example

Step 1. Reading data

Input file 1 - a list of genes with association p-values. The p-values are gene-based p-values, which can be estimated from GWAS SNP-level p-values (please refer to 'Methods' for details). For example,

```
> geneweight <- read.delim("gene_pvalues",as.is=T)
> head(geneweight,4)
     gene  weight
1 A3GALT2 0.04042
2 AADACL3 0.81300
3 AADACL4 0.56300
4   ABCA4 0.36200
```

Input file 2 - A PPI network in the format of protein interaction pair. For example,

```
> network <- read.delim("network.txt",as.is=T)
> head(network, 4)
  interactorA interactorB
1       SEPT6     SH3KBP1
2      ELAVL1       WAPAL
3       HPRT1        CUL5
4        TAF1       TAF15
```

Input file 3 – two gene expression matrices for case and control samples respectively. The first column is gene symbol, and the other columns indicate sample ids. Please make sure the two matrices have the same first column (i.e., the gene symbols are in the same order). For example,

```
> expr1 <- read.delim("case_expression",as.is=T)
> expr2 <- read.delim("control_expression",as.is=T)
> expr1[1:4,1:4]
  Gene_symbol  sample_1 sample_2 sample_3
1     HMGB1P1  6.377211 6.902405 6.715583
2   LOC155060  6.699311 6.540925 7.348245
3     HSPB1P1 10.444345 6.885489 9.393785
4      GTPBP6  8.442804 8.888129 8.188275
```

```
> expr2[1:4,1:4]
  Gene_symbol sample_1 sample_2 sample_3
1     HMGB1P1 6.840783 6.594853 6.469367
2   LOC155060 7.545505 7.904620 7.794645
3     HSPB1P1 7.476676 6.954855 7.383246
4      GTPBP6 8.608905 8.556181 8.824393
```

Step 2. Dense module search

One single function, ***dms***, performs all the analysis necessary for dense module search. The detail of the algorithm can be found in our manuscript (please see the References section on our web site). Six input parameters are necessary for the execution of this function:

geneweight: genes with association p-values read in step 1.
network: pair-wise PPI data read in step 1.
expr1 & expr2: two gene expression matrices read in step 1.
r: the cutoff for incensement during module expanding process. The score improvement of each step is required as passing $S_{m+1} > S_m \times (1 + r)$ for the inclusion of any neighborhood genes, where S_{m+1} is the module score by recruiting a neighborhood node.

λ: λ is a parameter between 0 and 1 to balance node and edge weights.

Box 1 explains how to choose values for *r* and λ. The command line to execute ***dms*** is:

```
> res.list <- dms(network, geneweight, expr1, expr2,  r=0.1, lambda=0.4)

> res.list <- dms(network, geneweight, expr1, expr2,  r=0.1, lambda="default" )

> res.list <- dms(network, geneweight, expr1=NULL, expr2=NULL, d=1, r=0.1)

> res.list <- dms(network, geneweight, expr1, expr2=NULL,  r=0.1, lambda="default" )
```

For λ, we provide two options. For users who have strong prior knowledge to balance GWAS and gene expression signals, they can directly provide a value between 0 and 1 (**the first command line**). For those who are not very sure on how to choose λ, EW_dmGWAS will estimate it automatically (**the second command line**; please see Box 1 and our manuscript for details on how to estimate λ).

dmGWAS_3.0 is compatible with the old versions, i.e., ***dmGWAS_1.0*** and ***dmGWAS_2.X***. Users can apply **the third command line** if they want to search for dense modules based on a background without edge weights. *d* is an integer to define the order of neighbor genes to be searched. *d* is always set up as 1 in ***dmGWAS_3.0***, but could be 1 or 2 in ***dmGWAS_1.0*** and ***dmGWAS_2.X***. Please refer to our website at http://bioinfo.mc.vanderbilt.edu/dmGWAS/dmGWAS_old.html for details.

In practical applications, users may sometimes do not have gene expression data for both case and control samples simultaneously. In ***dmGWAS_3.0,*** we provide the function to compute edge weights by using gene co-expression (GCE) when expression data is only available for one cohort. Users can use **the fourth command line** for this purpose. Please see Box 2 for details on how we estimate the edge weights when only one gene expression data set is available.

The resultant object, *res.list*, contains all the results, including the node-weighted network used for searching, the resultant dense modules and their component genes, the module score matrix, and the randomization data. A resultant file ***.RData** will also be generated for future recalling.

```
> names(res.list)
[1] "GWPI"                          "genesets.clear"
[3] "genesets.length.null.dis"      "genesets.length.null.stat"
[5] "module.score.matrix"           "ordered.module.score.matrix"
```

#GWPI, an object of igraph class, is the node- and edge-weighted PPI network.
#genesets.clear, a list, contains all the valid modules. The name of each record is the seed gene.
#genesets.length.null.dis, a list, contains the randomization data of for each size of modules.
#genesets.length.null.stat, a list, contains the statistic values of randomization data of for each size of modules.
#module.score.matrix, an object of matrix, contains data for each module: gene (seed gene), Sm (module score), Sn (normalized module score).
#ordered.module.score.matrix, ordered matrix of module.score.matrix based on Sn.

Step 3. Module selection

Modules are ranked and selected by the normalized module score S_n. Theoretically, and also based on our application, each gene has a local module; thus, there may be thousands of modules generated with extensive overlap between modules because of the complex structure of the human PPI network. As suggested in the original study (Jia*, et al.*, 2011), we usually chose the top modules for downstream analyses. This can be executed by calling function ***chooseModule***. The parameter *top* in ***chooseModule*** could be either a percentage (<1) or an integer (≥1). Fig. 1 shows an example of sub-network generated by function ***chooseModule.***

```
> selected <- chooseModule(res.list, top=0.01, plot=T)

> names(selected)

[1] "modules"    "subnetwork"
```

#modules, a list, contains all the selected modules. The name of each record is the seed gene.

#subnetwork, a sub-graph, contains all the nodes in the selected modules

```
> head(selected$modules,4)

$USP1
 [1] "CTNND1" "SRC"    "ERBB3"  "PTPN11" "ERBB2"  "USP1"   "MUC1"
 [8] "PTPN21"

$SLC9A2
 [1] "SRC"    "ERBB3"  "PTPN11" "ERBB2"  "JUP"    "MUC1"   "NRG1"

$EGLN1
 [1] "SRC"    "ERBB3"  "PTPN11" "ERBB2"  "JUP"    "MUC1"   "EGLN1"
 [8] "NRG1"   "PTPN21"

$ARRDC3
 [1] "SRC"    "ERBB3"  "PTPN11" "ERBB2"  "JUP"    "MUC1"   "ADRB2"
 [8] "GRK6"   "ARRDC3" "NRG1"   "PTPN21"

> selected$subnetwork

IGRAPH UNW- 129 195 --
+ attr: name (v/c), weight (v/n), weight (e/n)
```

Box 1. How to choose *r* and λ

The parameter *r* impedes restriction on the score of the module. When *r* is small, it imposes loose restriction during the module expanding process; thus, unrelated nodes and edges with lower weights (higher *P* values) might be included. On the other hand, when *r* is large, strict restriction is imposed and only those nodes and edges with very high weights (very low *P* values) could be included. As a result, it may miss some informative nodes that have moderate association *P* values. In our implementation, *r* is suggested to be 0.1, as it has been used in our previous versions of dmGWAS (Jia*, et al.*, 2011).

For λ, we provide two options. For users who have strong prior knowledge to balance GWAS and gene expression signals, they can directly provide a value between 0 and 1. For those who are not very sure on how to choose λ, it will be estimated automatically: EW_dmGWAS randomly extracts sub-networks 10,000 times from the background node- and edge-weighted PPI network, and compares the magnitude of edge weight part and node weight part by

$$mr = \left| \frac{\sum_{e \in E} edgeweight(e)}{\sqrt{\#\ of\ E}} \Big/ \frac{\sum_{v \in V} nodeweight(v)}{\sqrt{\#\ of\ V}} \right|, \quad (4)$$

where mr indicates magnitude ratio. λ is estimated as $1/(1 + median(mr))$.

In our application in a breast cancer data set, we compared the performance by using two user specified λ values (0.2 and 0.4) with the λ value estimated by EW_dmGWAS. As suggested in the original study (Jia*, et al.*, 2011), we chose the candidate genes residing in the top 1% of modules for evaluation. Table 1 lists the numbers of candidate genes and the overlap between CGC genes (http://cancer.sanger.ac.uk/cancergenome/projects/census/) and candidate genes under different λ. Table 2 lists the enriched KEGG pathways of candidate genes under different λ by WebGestalt (Zhang, et al., 2005).

Table 1. Summary of candidate genes identified under different λ in the breast cancer data set.

	dmGWAS	**EW_dmGWAS**		
λ	0	0.2	0.4	0.48 (default estimate)
# candidate genes	100	87	108	128
# overlap with the CGC genes	4	2	3	14

Table 2. Enriched KEGG pathways of candidate genes under different λ in the breast cancer data set.

Enriched KEGG pathway	**# of genes**	**Adjusted p-value***
dmGWAS (λ =0)		
Metabolic pathway	11	4.40×10^{-5}
EW_dmGWAS (λ =0.2)		
Metabolic pathway	11	1.38×10^{-5}
EW_dmGWAS (λ =0.4)		
No significant results	-	-
EW_dmGWAS (λ =0.48, default estimate)		
Pathways in cancer	10	8.22×10^{-7}
RIG-I-like receptor signaling pathway	6	1.11×10^{-6}
Neurotrophin signaling pathway	7	1.71×10^{-6}
Tight junction	7	2.23×10^{-6}
Hepatitis C	7	2.48×10^{-6}
ErbB signaling pathway	6	3.78×10^{-6}
Endocytosis	7	3.80×10^{-5}
Adherens junction	5	4.08×10^{-5}
GnRH signaling pathway	5	2.00×10^{-4}
Leukocyte transendothelial migration	5	4.00×10^{-4}
Focal adhesion	6	5.00×10^{-4}
Jak-STAT signaling pathway	5	1.50×10^{-3}
Calcium signaling pathway	5	3.00×10^{-3}
Chemokine signaling pathway	3	4.50×10^{-3}

*P-values were adjusted by Bonferroni correction.

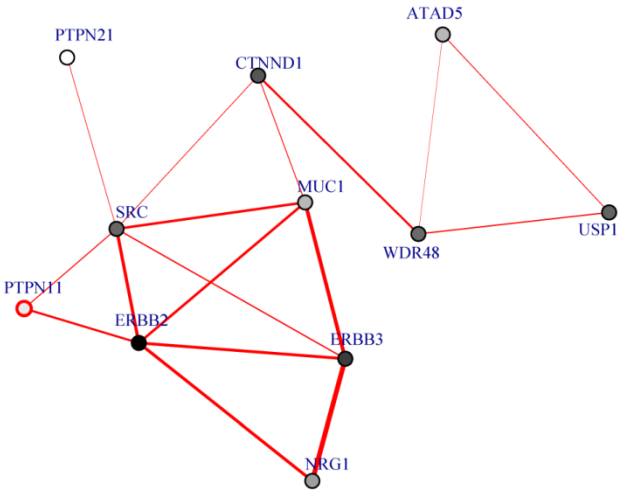

Fig. 1. An example of subnetwork generated by function ***chooseModule***. The color of nodes indicates the node weights. White color represents high weight. The width of edges indicates the edge weights. The more wider an edge is, the higher weight it has.

Box 2. Estimation of edge weights by using GCE

In practical applications, users may sometimes do not have gene expression data for both case and control samples simultaneously. In our algorithm, we provide the function to estimate edge weights based on GCE by only using one gene expression data set. Specifically, let r represent the Pearson's correlation coefficient (PCC) of expression for a pair of genes, and n represent the number of samples. We defined the new statistic as $T = r\sqrt{n-2}/\sqrt{1-r^2}$, and T approximately follows the student's t-distribution with $n-2$ degree. We then defined edge weight as $edgeweight(e) = {f_{n-2}}^{-1}\left[1-2*\left(1-f_{n-2}(|X|)\right)\right]$, where f_{n-2} is the cumulative distribution function of t-distribution with $n-2$ degree.

In our application in a schizophrenia (SCZ) data set, we compared the performance between DGCE and GCE (Please refer to our manuscript for the details of the SCZ data set). We had two gene expression profiles (one for case samples and one for control samples). Accordingly, we have the edge weights computed using GCE in the case samples only (referred to as GCE_case), the edge weights computed using GCE in the control samples only (referred to as GCE_control), and the edge weights computed using DGCE based on both case and control samples (referred to as DGCE). We applied the same greedy algorithm as provided by EW_dmGWAS using these edge weights respectively and obtained genes in the top 1% modules by each strategy. We used 38 SCZ core genes as a benchmark to evaluate the identified candidate genes (Jia*, et al.*, 2010). Tables 3-4 summarize the comparison of candidate genes and enriched pathways obtained using different edge weights. Overall, the candidate genes obtained using DGCE contained the highest proportion of SCZ core genes (Table 3) (all p-values < 0.05, binomial test). In terms of pathways (Table 4), two SCZ related pathways are enriched in the candidate genes identified by DGCE, including 'Endocytosis' and 'Neuroactive ligand-receptor interaction'. Recent studies have shown that 'Neuroactive ligand-receptor interaction' plays important roles in the antipsychotic treatment response (Adkins*, et al.*, 2012), while 'Endocytosis' has been implicated as the common pathophysiology underlying SCZ (Zhao*, et al.*, 2014). Although the candidate genes identified by using GCE_case are enriched in several SCZ related pathways, such as 'Neurotrophin signaling pathway' and 'Regulation of actin cytoskeleton', a number of other pathways are also enriched, but are not readily related to SCZ. In contrast, when the edge weights were computed using the control samples, little information could be identified. **Overall, the results implicated that DGCE is a more effective way to infer the edge weights. However, in our algorithm, we provide the function to compute edge weights based on GCE and allow users to explore different options in computing edge weights, especially when the expression data are not available for both case and control samples.**

Table 3. Summary of candidate genes identified by different edge weight strategies in the SCZ data set.

	GCE_case	**GCE_control**	**DGCE**
# candidate genes	181	114	65
# overlap with the 38 SCZ core genes	2	0	2

Table 4. Enriched KEGG pathways of candidate genes identified by different edge weight strategies in the SCZ data set.

Enriched KEGG pathway	# of genes	Adjusted p-value*
GCE_case		
Neurotrophin signaling pathway	8	1.44×10^{-6}
Phagosome	8	3.04×10^{-6}
Leukocyte transendothelial migration	7	4.32×10^{-6}
Jak-STAT signaling pathway	7	2.25×10^{-5}
Focal adhesion	7	5.90×10^{-5}
Tight junction	6	5.90×10^{-5}
VEGF signaling pathway	5	5.90×10^{-5}
Cell cycle	6	5.90×10^{-5}
Regulation of actin cytoskeleton	7	7.68×10^{-5}
Prostate cancer	5	7.68×10^{-5}
T cell receptor signaling pathway	5	2.00×10^{-4}
MAPK signaling pathway	7	2.00×10^{-4}
Oocyte meiosis	5	2.00×10^{-4}
Chemokine signaling pathway	6	3.00×10^{-4}
Osteoclast differentiation	5	3.00×10^{-4}
Hepatitis C	5	4.00×10^{-4}
Insulin signaling pathway	5	4.00×10^{-4}
Pathways in cancer	7	6.00×10^{-4}
Protein processing in endoplasmic reticulum	5	7.00×10^{-4}
GCE_control		
Ribosome	8	2.88×10^{-10}
DGCE		
Protein processing in endoplasmic recticulum	7	1.83×10^{-8}
Endocytosis	5	2.07×10^{-5}
Neuroactive ligand-receptor interaction	5	5.83×10^{-5}

*P-values were adjusted by Bonferroni correction.

References

Adkins, D.E.*, et al.* (2012) SNP-based analysis of neuroactive ligand-receptor interaction pathways implicates PGE2 as a novel mediator of antipsychotic treatment response: data from the CATIE study, *Schizophr Res*, **135**, 200-201.

Ballard, D.H., Cho, J. and Zhao, H. (2010) Comparisons of multi-marker association methods to detect association between a candidate region and disease, *Genet Epidemiol*, **34**, 201-212.

Hou, L.*, et al.* (2014) Guilt by rewiring: gene prioritization through network rewiring in genome wide association studies, *Hum Mol Genet*, **23**, 2780-2790.

Jia, P.*, et al.* (2010) SZGR: a comprehensive schizophrenia gene resource, *Mol Psychiatry*, **15**, 453-462.

Jia, P.*, et al.* (2012) Network-assisted investigation of combined causal signals from genome-wide association studies in schizophrenia, *PLoS Comput Biol*, **8**, e1002587.

Jia, P.*, et al.* (2011) dmGWAS: dense module searching for genome-wide association studies in protein-protein interaction networks, *Bioinformatics*, **27**, 95-102.

Liu, J.Z.*, et al.* (2010) A versatile gene-based test for genome-wide association studies, *Am J Hum Genet*, **87**, 139-145.

Zhang, B., Kirov, S. and Snoddy, J. (2005) WebGestalt: an integrated system for exploring gene sets in various biological contexts, *Nucleic Acids Res*, **33**, W741-748.

Zhao, Z.*, et al.* (2014) Transcriptome sequencing and genome-wide association analyses reveal lysosomal function and actin cytoskeleton remodeling in schizophrenia and bipolar disorder, *Mol Psychiatry*, Epub ahead of print.