The GeneSurrounder package Vignette

Sahil Shah and Rosemary Braun <rbraun@northwestern.edu>

May 30, 2017

Availability

The GeneSurrounder package and its documentation are available on GitHub at https://github.com/sahildshah1.

Introduction

The GeneSurrounder package implements the method we previously developed [1] to identify disease-associated genes from expression data and an independent network model of cellular interactions. We developed GeneSurrounder to find the genes with neighbors on the network that are differentially expressed (with the magnitude of the differential expression decreasing with distance from the putative disease gene) and have correlated expression with the putative disease gene. Since the differential expression of the neighbors of a putative disease gene does not depend on their association with that gene, our algorithm consists of two tests that are run independently of each other (Table 1, Table 2). Their results are then combined to determine if the putative disease gene is a central candidate disease gene.

Example

In order to illustrate our method, we apply our algorithm to one study of high-vs-low grade ovarian cancer from the publicly available and curated collection curatedOvarianData (GEO accession GSE14764) [2]. We have constructed the global network model from KEGG pathways [3].

Load Data Set

Our algorithm uses the correlation between the expression of the genes, their differential expression, and their distances on the global network.

```
> load("../../data/CurOvGradeKEGGnets.RData")
> load("../../data/largestCompKEGGigraph.RData")
>
```

Source Functions

The functions that implement our method have to be sourced. The Observed.SI, Resample.SI,SumAbsCor functions implement the *Sphere of Influence* procedure and the Resample.DecayDE, Observed.DecayDE functions implement the *Decay of Differential Expression* procedure. The geneNIDG function calls these functions.

```
> library(pcaPP)
> library(igraph) # load largestCompKEGGigraph
> library(limma) #calcGeneStats()
> library(metap) #pFisher sumlog()
> source("../../R/calcCorMatrix.R")
> source("../../R/calcGeneTStats.R")
> source("../../R/calcAllPairsDistances.R")
> source("../../R/Observed.SI.R")
> source("../../R/Resample.SI.R")
> source("../../R/SumAbsCor.R")
> source("../../R/Resample.DecayDE.R")
> source("../../R/Observed.DecayDE.R")
> source("../../R/geneNIDG.R")
>
```

Apply Functions to Data

The correlation between the expression of the genes is calculated.

```
> CurOv_RankCorMatrix_GSE14764_eset <-
+ calcCorMatrix(exprMatrix = CurOvGradeKEGGnets[["GSE14764_eset"]]$expr,
+                           corMethod = "spearman",
+                           exprName = paste("CurOvGradeKEGGnets$","GSE14764_eset",sep=""))
>
```

The observed and resampled differential expression of the genes is calculated.

```
> # List of observed (vector) and resampled (resampling by gene matrix) t statistics
>
> intersectGeneNames = intersect(rownames(CurOvGradeKEGGnets[[2]]$expr),
+                                                             V(largestCompKEGGigraph)$name)
> expr = CurOvGradeKEGGnets[["GSE14764_eset"]]$expr
> classLabels = CurOvGradeKEGGnets[["GSE14764_eset"]]$grade
> #I can reduce the number of t tests by reducing the expr matrix to
> #only genes that are on the network.
> reducedExpr = expr[intersectGeneNames,]
> geneTStats = calcGeneTStats(reducedExpr,
+                                                          classLabels,
+                                                          numResamples = 1000)
>
```

The distances on the global network are calculated.

```
> CompKEGG_ShortestDistMatrix <-
+ calcAllPairsDistances(network = largestCompKEGGigraph,
+                                              directionPaths="all",
+                                              weightVector = NULL,
+                                              networkName = "largestCompKEGGigraph")
>
```

In this example, MCM2 (KEGG ID: hsa:4171) is the candidate disease gene.

```
> genes.assayedETnetwork <- intersect(
+         rownames(CurOv_RankCorMatrix_GSE14764_eset),
+         rownames(CompKEGG_ShortestDistMatrix))
> gene.id <- "hsa:4171"
>
```

The Sphere of Influence and Decay of Differential Procedures are run.

```
> geneNIDG.hsa4171 <-  geneNIDG(
+         gene.id = gene.id,
+         distance.matrix = CompKEGG_ShortestDistMatrix,
+         cor.matrix = CurOv_RankCorMatrix_GSE14764_eset,
+         geneStats.observed = geneTStats$observed,
+         perm.geneStats.matrix = geneTStats$resampled,
+         num.Sphere.resamples = 1000,
+         diameter = 34,
+         genes.assayedETnetwork = genes.assayedETnetwork)
>
>
```

The evidence from both procedures is combined.

```
> p.Fisher <- vapply( 1:34, function(index){
+
+         x <- sumlog( c(geneNIDG.hsa4171$p.Decay[index],
+                         geneNIDG.hsa4171$p.Sphere[index]) )
+
+         return(x$p)
+
+
+ },
+ numeric(1) )
> geneNIDG.hsa4171 <- cbind(geneNIDG.hsa4171,p.Fisher)
>
```

Description of the Output

Our method outputs a data frame.

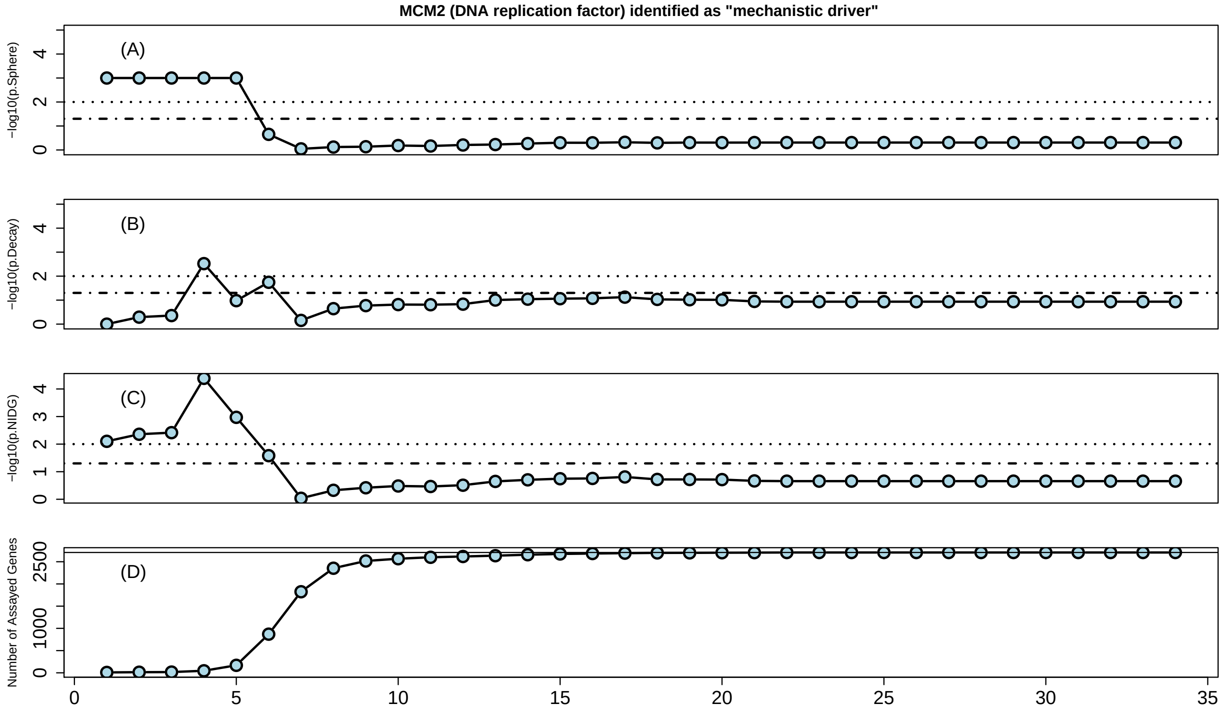
```
> str(geneNIDG.hsa4171)
```

```
'data.frame':        34 obs. of  8 variables:
 $ gene.id       : Factor w/ 1 level "hsa:4171": 1 1 1 1 1 1 1 1 1 1 ...
 $ radius        : int  1 2 3 4 5 6 7 8 9 10 ...
 $ size          : num  9 14 17 46 169 ...
 $ observed.tau_b: num  NaN -0.0156 -0.046 -0.4233 -0.0916 ...
 $ p.Decay       : num  1 0.511 0.439 0.003 0.105 0.018 0.699 0.226 0.169 0.153 ...
 $ observed.cor  : num  3.29 6.2 7.45 14.36 39.42 ...
 $ p.Sphere      : num  0.000999 0.000999 0.000999 0.000999 0.000999 ...
 $ p.Fisher      : num  7.90e-03 4.38e-03 3.83e-03 4.11e-05 1.07e-03 ...
```

```
>
>
```

We plot the results against every radius.

```
> source("../../R/plotRadiusVS.R")
> plotRadiusVS(geneNIDG.hsa4171)
>
```



Tables

```
Sphere of Influence

Input : Define the gene expression data and the network model
For putative driver gene $i$
    1. Calculate the Spearman rank correlation $\rho_k$ between the expression of gene $k$ and gene $i$ across all the conditions in the expression matrix for every other gene $k$: $\rho_1, \rho_2, \ldots, \rho_{N-1}$ (where $N$ is the number of genes both assayed and on the network)
    2. Calculate the total observed correlation between gene $i$ and its neighbors: $\sum_{k=1}^{M} |\rho_k|$ where the correlations $\rho_k$ are indexed so that the first $M$ genes are in the neighborhood of gene $i$
    3. Compute the distribution of total correlation under the null by generating $N_{\text{Sphere}}$ null total correlations. To generate each null total correlation:
        (a) Bootstrap from the correlations calculated in Step (1)
        (b) Preserve the gene indices's so that the first re-sampled correlation is assigned to gene $k = 1$, etc and recalculate the total correlation as in Step (2)
    4. Compute $p^i_{\text{Sphere}}$ by counting the the proportion of null sums greater than or equal to the observed sum. If $p^i_{\text{Sphere}} = 0$, set $p^i_{\text{Sphere}} = \frac{1}{N_{\text{Sphere}}+1}$
Output : Return $p^i_{\text{Sphere}}$
```

Table 1: The Sphere of Influence procedure tests if a putative driver gene is more correlated with its network neighbors than with a random set of genes.

```
Decay of Differential Expression

Input : Define the gene expression data and the network model
Calculate the moderated $t$-statistic $g_k$ between cases and controls of each gene: $g_1, g_2, \ldots, g_N$
For putative driver gene $i$
    1. Compute the observed discordance between differential expression and distance from gene $i$: $\tau_B(\{g_1, \ldots, g_M\}, \{d(1,i), \ldots, d(M,i)\}$ where the gene-level statistics are indexed so that the first $M$ genes are in the neighborhood of gene $i$, $\tau_B$ is the Kendall rank correlation, and $d(k,i)$ is the geodesic distance between gene $k$ and gene $i$
    2. Compute the distribution of the discordance under the null by generating $N_{\text{Decay}}$ null discordances. To generate each null discordance:
        (a) Permute the phenotype labels and recalculate the moderated $t$-statistics for each gene
        (b) Recompute the discordance as in Step (1)
    3. Compute $p^i_{\text{Decay}}$ by counting the the proportion of null discordances less than or equal to the observed discordance. If $p^i_{\text{Decay}} = 0$, set $p^i_{\text{Decay}} = \frac{1}{N_{\text{Decay}}+1}$
Output : Return $p^i_{\text{Decay}}$
```

Table 2: The Decay of Differential Expression procedure tests if the discordance between differential expression and distance from a putative driver gene is greater with the phenotype labels we observe than with a random permutation of the labels.

References

[1] Sahil Shah and Rosemary Braun. Network-based identification of candidate disease genes in expression data. *Forthcoming*, 2017.

[2] Benjamin Frederick Ganzfried, Markus Riester, Benjamin Haibe-Kains, Thomas Risch, Svitlana Tyekucheva, Ina Jazic, Xin Victoria Wang, Mahnaz Ahmadifar, Michael J. Birrer, Giovanni Parmigiani, Curtis Huttenhower, and Levi Waldron. curatedOvarianData: Clinically annotated data for the ovarian cancer transcriptome. *Database*, 2013:1–10, 2013.

[3] Minoru Kanehisa, Michihiro Araki, Susumu Goto, Masahiro Hattori, Mika Hirakawa, Masumi Itoh, Toshiaki Katayama, Shuichi Kawashima, Shujiro Okuda, Toshiaki Tokimatsu, and Yoshihiro Yamanishi. KEGG for linking genomes to life and the environment. *Nucleic Acids Research*, 36(SUPPL. 1):480–484, 2008.