RERconverge Enrichment Functions Walkthrough

Amanda Kowalczyk

September 06, 2018

Contents

This walkthrough provides instructions for implementing the RERconverge package enrichment functions to identify gene pathways or groups of genes whose evolutionary rates shift in association with change in a trait. For information on how to download and install RERconverge, see the wiki. Source code and a quick start guide are available on github. A detailed walkthrough to produce the data used for this example is provided in the full walkthrough vignette.

Overview

This document describes how to run pathway enrichment analysis using RERconverge output and functions included with the RERconverge package. Enrichment analysis detects groups of genes that are evolving faster or slower with a phenotype of interest. In the RERconverge package, the enrichment function is implemented as a Wilcoxon Rank-Sum Test on a list of genes ranked based on their correlation statistics. It detects distribution shifts in groups of genes compared to all genes, and it thereby bypasses the need to set a foreground significance cutoff like some other enrichment methods.

Input to the enrichment function is the output from RERconverge correlation functions and pathways of interest with gene symbols (gene names).

Output is enrichment statistics for each pathway, including genes in the pathways and their ranks.

Extract Results from RERconverge Correlation Analysis

Enrichment analysis starts with the results from the *correlateWithContinuousPhenotype*, *correlateWithBinaryPhenotype*, or *getAllCor* functions in the RERconverge package. These results include Rho, p-value, and the Benjamini-Hochberg corrected p-value for the correlation between the relative evolutionary rates of each gene and the phenotype provided. These statistics are used to calculate enrichments.

In this case, we will start with the data from the continuous trait analysis described in the full walkthrough that used adult mass as the phenotype of interest in mammalian species. This walkthrough assumes that you have already installed RERconverge.

```
library(RERconverge)
data("RERresults")
```

Deriving a ranked gene list

We will perform our enrichment on Rho-signed negative log p-values from our correlation results. The *getStat* function converts our correlation results to these values and removes NA values.

```
library(RERconverge)
stats=getStat(res)
```

Import Pathway Annotations

Now that we have our gene statistics, we need pathway annotations. Download all curated gene sets, gene symbols (c2.all.v6.2.symbols.gmt) from GSEA-MSigDB as gmtfile.gmt. You must register for an account prior to downloading. The rest of the vignette expects "gmtfile.gmt" to be in the current working directory.

With the file in the appropriate place, simply use the *read.gmt* function to read in the annotation data.

```
annots=read.gmt("gmtfile.gmt")
```

Format Pathway Annotations

RERconverge enrichment functions expect pathways to be in named pathway-group lists contained within a list. A pathway-group is a group of similar pathways stored together (for example KEGG pathways might be one pathway-group and MGI pathways might be another pathway-group). Each pathway-group list contains a named list called *genesets* with each element of the list a list of gene names in a particular pathway, and the names of the elements the names of the pathways. The names of the genesets are also contained as a character vector in the second element of the pathway-group list named *geneset.names*. As an example of correctly-formatted annotations, you would have a list called *annotations* that contains another list called *canonicalpathways*. The *canonicalpathways* list contains a list named *genesets* and a vector named *geneset.names*. Each element of the *genesets* list is a list of gene names corresponding to a particular pathways, and the names of the elements in *genesets* are the names of the pathways. The *geneset.names* vector contains the names of the elements in *genesets*.

To convert our gmt file to the correct format, we simply need to put it inside another list; the *read.gmt* function automatically reads in gmt files in the format described above.

```
annotlist=list(annots)
names(annotlist)="MSigDBpathways"
```

Calculate Enrichment Using *fastwilcoxGMTAll*

We can now use *annotlist* and *stats* to calculate pathway enrichment. We will use the function *fastwilcoxGMTAll*, which calculates the estimated Wilcoxon Rank-Sum statistics based on the ranks of the genes in *stats*. The test essentially measures the distribution shift in ranks of genes of interest (each pathway) compared to the background rank distribution (ranks of all other genes included in the pathway-group annotations that are not part of the pathway of interest). We set *outputGeneVals* to true so the function returns names of the

genes in the pathway and their ranks. *num.g* specifies the minimum number of genes required to be present to run the enrichment (10 by default).

```
enrichment=fastwilcoxGMTall(stats, annotlist, outputGeneVals=T, num.g=10)
```

```
## 25 results for annotation set MSigDBpathways
```

Note that these results contain statistics for all pathways with the minimum number of genes specified. Very few pathways have a sufficient number of genes because we only calculated correlation statistics for 200 total genes.

Further Analysis and Visualization

You may visualize your pathway results in a variety of ways, including using **PathView** to overlay correlation colors on KEGG pathways and **igraph** to make pathway networks based on gene similarity among networks.

You may also calculate permutation p-values based on permuting the phenotype vector and rerunning the correlation and enrichment analyses many times to filter pathways that always tend to have significant enrichment values regardless of the phenotype input.