# The Discordant R Package: A Novel Approach to Differential Correlation

Charlotte Siska and Katerina Kechris

May 10, 2016

May 10, 2016

## Contents

## 1 Introduction

Discordant is an R package that identifies pairs of features that correlate differently between phenotypic groups, with application to -omics datasets. *Discordant* uses a mixture model that bins molecular feature pairs based on their type of coexpression. More information on the algorithm can be found in Siska, et. al. The final output are posterior probabilities of differential correlation. This package can be used to determine differential correlation within one omics dataset or between two omics datasets (provided that both omics datasets were taken from the same samples). Also, the type of data can be any type of omics, such as metabolomics, transcriptomic, proteomics, etc. as long as the data are continuous (numerical) rather than discrete (categorical, count).

The functions in the Discordant package provide a simple pipeline for intermediate R users to determine differentially correlated pairs. The final output is a table of molecular feature pairs and their respective posterior probabilities. Functions have been written to allow flexibility for users in how they interpret results, which will be discussed further. Currently, the package only supports the comparison between two phenotypic groups (e.g., disease vs control, mutant vs wildtype).

## 2 Citing Discordant

Discordant is originally derived from the Concordant algorithm written by Lai, et. al. When citing Discordant, please also include Lai, et. al in references.

Lai, Y., Adam, B. -l., Podolsky, R., and She, J.-X. (2007). A mixture model approach to the tests of concordance and discordance between two large-scale experiments with two-sample groups. Bioinformatics 23, 12431250.

Siska C., Bowler R.P and Kechris K. (2015). The Discordant Method: A Novel Approach to Differential Correlation. Bioinformatics.

## 3 Discordant Algorithm

Using a three component mixture model and the EM algorithm, the model predicts if the correlation coefficients in phenotypic groups 1 and 2 for a

|  | 0 | - | + |
|---|---|---|---|
| 0 | 1 | 2 | 3 |
| - | 4 | 5 | 6 |
| + | 7 | 8 | 9 |

Table 1: Class Matrix

molecular feature pair are dissimilar. The correlation coefficients are generated for all possible molecular feature pairs between -omics A and -omics B and are transformed in to z scores using Fisher's tranformation. The three components are -, + and 0 which correspond respectively to a negative, positive or no correlation. Molecular features that have correlation coefficients in *different* components are considered *differentially* correlated, as opposed when correlation coefficients are in the *same* component they are *equivalently* correlated.

The class matrix (Table 1) are the classes that represent all possible paired-correlation scenarios. These scenarios are based off the components in the mixture models. Molecular features that have correlation coefficients in different components are considered differentially correlated, as opposed to when correlation coefficients are in the same component they are equivalently correlated. This can be visualized in the class matrix, where the rows represent the components for group 1 and the columns represent the components for group 2. The classes on the diagonal represent equivalent correlation (1, 5 and 9), and classes in the off-diagonal represent differential correlation (2, 3, 4, 6, 8).

After running the EM algorithm, we have 9 posterior probabilities for each molecular feature pair that correspond to the the 9 classes in the class matrix. Since we want to summarize the probability that the molecular feature pair is differentially correlated, we sum the posterior probabilities representing the off-diagonal classes in the class matrix.

## 4 Quick Workflow

### 4.1 Brief Introduction

Within omics refers to when the Discordant analysis is performed within one omics dataset. This means that all molecular features are analyzed to each other, rather than separating them by molecular type. This is mainly applicable to one omics dataset, such as a single microarray experiment.

Between -omics refers to when the Discordant analysis is performed with two -omics datasets. Molecular feature pairs analyzed are between the two -omics, i.e. transcript-protein, protein-metabolite, etc.

### 4.2 Required Inputs

**x** m by n matrix where m are features and n are samples. If only this matrix inputted into functions, a within -omics analysis is performed.

**y** m by n matrix where m are features and n are samples. Optional, will induce between -omics analysis. Samples must be matched with those in x.

**groups** vector containing 1s and 2s that correspond to the location of samples in the column of the matrix for group 1 and group 2. For example, the control group is group 1 and the experimental group 2, and the location of samples corresponding to the two groups matches the locations of 1s and 2s in the group vector

### 4.3 Example Run with Microarrays

Run with microarray data, using `TCGA_GBM_miRNA_microarray` and `TCGA_GBM_transcript_microa`

```
data(TCGA_GBM_miRNA_microarray)
data(TCGA_GBM_transcript_microarray)
groups <- c(rep(1,10), rep(2,10))

#Within -Omics

vectors <- createVectors(TCGA_GBM_transcript_microarray, \
      groups = groups)
```

```
result <- discordantRun(vectors$v1, vectors$v2, \
      TCGA_GBM_transcript_microarray)
resultTable <- makeTable(result$discordPPMatrix, \
      TCGA_GBM_transcript_microarray)

#Between -Omics

vectors <- createVectors(TCGA_GBM_miRNA_microarray, \
      TCGA_GBM_transcript_microarray, groups = groups)
result <- discordantRun(vectors$v1, vectors$v2, \
      TCGA_GBM_miRNA__microarray, \
      TCGA_GBM_transcript_microarray)
resultTable <- makeTable(result$discordPPMatrix, \
      TCGA_GBM_miRNA_microarray, \
      TCGA_GBM_transcript_microarray)
```

In addition to microarray datasets, test sequencing datasets are included, `TCGA_Breast_miRNASeq` and `TCGA_Breast_RNASeq` and voom-transformed datasets, `TCGA_Breast_miRNASeq_voom` and `TCGA_Breast_RNASeq_voom`.

## 5 Create Correlation Vectors

To run the Discordant algorithm correlation vectors respective to each group are necessary for input, which are easy to create using the `createVectors`. Each correlation coefficient represents the correlation between two molecular features. The molecular features depend if a within -omics or between -omics analysis has been performed. Correlation between molecular features in the same -omics dataset is within -omics, and correlation between molecular features in two different -omics datasets is between -omics. Whether or not within -omics or between -omics analysis is performed depends on whether one or two matrices are parameters for this function. `createVectors` has two outputs:

**v1** Correlation vector of molecular feature pairs corresponding to samples labeled 1 in group parameter.

**v2** Correlation vector of molecular feature pairs corresponding to samples labeled 2 in group parameter.

### 5.1 Correlation Metric

We also have included different options for correlation metrics. This argument is called `cor.method` and its default is `"spearman"`. Other options are `"pearson"`, `"bwmc"` and `"sparcc"`. For information and comparison of Spearman, Pearson and biweight midcorrelation (bwmc) please read this paper by Song et al.

The algorithm for SparCC was introduced by Friedman et al and the python tool is available online at bitbucket. We use R code written by Huaying Fang .

Example

```
vectors <- createVectors(TCGA_GBM_miRNA_microarray, \
      TCGA_GBM_transcript_microarray, groups = groups \
      cor.method = c("spearman"))
```

## 6 Run the Discordant Algorithm

The Discordant Algorithm is in the function `discordantRun` requires two correlation vectors and the original data. If the user wishes to generate their own correlation vector before inputting into the dataset, they can do so. However, the function will break if the dimensions of the datasets inserted do not match the correlation vector.

The posterior probability output of the Discordant algorithm are the summarized DC posterior probabilities (those in the off-diagonal of the class matrix described in Table 1). If the user wishes to observe more detailed conversation, alternative outputs are available. `discordantRun` has five outputs:

**discordPPMatrix** Matrix of summarized posterior probabilities where rows and columns reflect features. This value is necessary for `makeTable`.

**discordPPV** Vector of summarized posterior probabilities.

**class** Class with the highest posterior probability for each pair.

**classMatrix** Matrix of class with highest posterior probability where rows and columns reflect features.

**probMatrix** All posterior probabilities, where each row represents a feature pair.

**loglik** The log likelihood.

## 6.1 Subsampling

Subsampling is when independent feature pairs are drawn, ran through the EM algorithm to estimate parameters for a number of iterations, and then the average of these paramters are used to maximize posterior probabilities for all feature pairs. There are several arguments introduced so the subsampling option can be run to the user's preference. This option was introduced to make the Discordant method to run faster and also solve the independence assumption. Of course, it has its own set of issues which are explained in Siska, et al (submitted).

The argument `subsampling` must be set to `TRUE` for subsampling to be used. The number of independent feature pairs to be subsampled is determined by the argument `subSize` whose default value is the number of rows in `x`. The number of independent feature pairs must be less or equal to the number of features in `x` and `y`. The number of iterations to be run is set by the argument `iter`, whose default value is 100.

Example:

```
result <- discordantRun(vectors$v1, vectors$v2, \
      TCGA_GBM_transcript_microarray, \
      TCGA_GBM_miRNA_microarray, subsampling = TRUE, \
      iter = 200, subSize = 20)
```

## 6.2 Increase Component Size

There is now the option to increase component size from three to five in the mixutre model. Having five components instead of three in the mixture model allows the identification of feature pairs that have elevated differential correlation, or when there are associations in both groups in the same direction but one is more extreme. While this option introduces a new type of differential correlation, it does run longer and has less power than the three component mixture model.

The argument to use a five component mixture model instead of a three component model is `components` set to 5. The default is to run the three component mixture model.

Example:

```
result <- discordantRun(vectors$v1, vectors$v2, \
      TCGA_GBM_transcript_microarray, \
      TCGA_GBM_miRNA_microarray, components = 5)
```

## 7 Make Table to Summarize

To ease the user in determining the posterior probability for each pair, the function `makeTable` is included. The only parameters required is the matrix of summed up discordant posterior probabilities from `discordantRun` and the data matrices. Now it is possible to search posterior probabilities for all feature pairs, and retrieve the top feature pairs that meet a posterior probability threshold.