How To Use CORREP to Estimate Multivariate Correlation and Statistical Inference Procedures

Dongxiao Zhu

October 28, 2009

1 Introduction

OMICS data are increasingly available to biomedical researchers, and (biological) replications are more and more affordable for gene microarray experiments or proteomics experiments. The functional relationship between a pair of genes or proteins are often inferred by calculating correlation coefficient between their expression profiles. Classical correlation estimation techniques, such as Pearson correlation coefficient, do not explicitly take replicated data into account. As a result, biological replicates are often averaged before correlations are calculated. The averaging is not justified if there is poor concordance between samples and the variance in each sample is not similar. Based on our recently proposed multivariate correlation estimator, `CORREP` implements functions for estimating multivariate correlation for replicated OMICS data and statistical inference procedures. In this vignette I demo an non-trivial task accomplished using `CORREP`. First let's look at examples of replicated OMICS data and non-replicated OMICS data.

$$x_{11}, x_{12}, \ldots, x_{1n}$$

$$x_{21}, x_{22}, \ldots, x_{2n}$$

$$\ldots$$

$$x_{m_1 1}, x_{m_1 2}, \ldots, x_{m_1 n}$$

$$y_{11}, y_{12}, \ldots, y_{1n}$$

$$y_{21}, y_{22}, \ldots, y_{2n}$$

$$\ldots$$

$$y_{m_2 1}, y_{m_2 2}, \ldots, y_{m_2 n}$$

versus

$$x_1, x_2, \ldots, x_n$$

$$y_1, y_2, \ldots, y_n$$

In this toy example, X and Y are a pair of genes or proteins of interest. Using microarray or mass spectrum we were able to profile their expression over n biological conditions such as knockouts, overexpression or SiRNA treatments either with replication (upper example) or without replication (lower example). There are m_1 and m_2 replicates for X and Y respectively. It is often biological

interest how strong is the correlation between X and Y over n conditions and how significant the correlation is. Significant correlation between X and Y often indicates potential functional relevancy. In addition, 1-correlation are usually used to calculate distance matrix of a number of genes or proteins to perform hierarchical clustering. Therefore, correlation estimation is an important problem in functional genomics research.

For non-replicated OMICS data, estimating correlation is relatively trivial since there are plenty of established methods such as Pearson correlation coefficient and its non-parametric alternatives: Spearman's ρ and Kendall's τ. However it is not straightforward to estimate correlation from replicated OMICS data. A naive approach may be to average over replicates to transform the replicated data into non-replicated data. This approach might work for relative "clean" data in which the within-replicate correlation is relatively high. Unfortunately most OMICS data are noisy reflected partially by poor within-replicate correlation. The averaging for those "noisy" data is not justified. In order to properly estimate correlation of X and Y from replicated data, we must explicitly model the within-replicate correlation together with the between-replicate correlation. The next section briefly describes the models and methods for deriving the multivariate correlation estimator. For interested reader, please refer to our manuscript for more technical details [Zhu and Li, 2007]. Interested reader also refer to [Medvedovic *et al.*, 2004] for related Bayesian mixture model methods.

2 Methods

2.1 Pearson correlation estimator

Instead of averaging, we exploit all the replicated observations by assuming the data are i.i.d. samples from a multivariate normal distribution with a specified correlation matrix and a mean vector, i.e., $Z_j = (x_{j1}, \ldots, x_{jp}, y_{j1}, \ldots, y_{jq})^T$, $j = 1, 2, \ldots, n$, follows a $p+q$-variate normal distribution $N(\mu, \Sigma)$, where $\mu = \begin{bmatrix} \mu_x e_m \\ \mu_y e_m \end{bmatrix}$, $e_m = (1, \ldots, 1)^T$ is a $m \times 1$ vector, the correlation matrix Σ is a $(p+q) \times (p+q)$ matrix with structure:

$$\Sigma = \begin{pmatrix} 1 & \ldots & \rho_x & \rho & \ldots & \rho \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ \rho_x & \ldots & 1 & \rho & \ldots & \rho \\ \rho & \ldots & \rho & 1 & \ldots & \rho_y \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ \rho & \ldots & \rho & \rho_y & \ldots & 1 \end{pmatrix} = \begin{bmatrix} \Sigma_x & \Sigma_{xy} \\ \Sigma_{xy}^T & \Sigma_y \end{bmatrix}, \tag{1}$$

where the inter-molecule correlation ρ is the parameter of interest, and the intra-molecule correlation ρ_x or ρ_y are nuisance parameters. The ρ_x and ρ_y indicate the quality of replicates that high quality replicates tend to have high value, and *vice versa*. We employ three parameters: ρ, ρ_x and ρ_y to model the correlation structure of replicated omics data.

2.2 Multivariate correlation estimator

Assuming a multivariate normal model, the Maximum Likelihood Estimate (MLE) of ρ can be derived as follows (see Manuscript for more details):

$$\hat{\mu}_x = \frac{1}{n}\frac{1}{m}\sum_{j=1}^{n}\sum_{i=1}^{m} x_{ij} \tag{2}$$

Similarly,

$$\hat{\mu}_y = \frac{1}{n}\frac{1}{m}\sum_{j=1}^{n}\sum_{i=1}^{m} y_{ij} \tag{3}$$

therefore, $\hat{\mu} = \begin{bmatrix} \hat{\mu}_x e_m \\ \hat{\mu}_y e_m \end{bmatrix}$
The MLE of Σ is

$$\hat{\Sigma} = \frac{1}{n}\sum_{j=1}^{n}(Z_j - \hat{\mu})(Z_j - \hat{\mu})^T \tag{4}$$

To derive the MLE of ρ, the ideal method would be obtaining the likelihood explicitly as a function of ρ. However, this proved to be intractable in practice (see manscript for detailed discussion). Our approach is to use the average of the elements of $\hat{\Sigma}_{xy}$ estimated via Eq. 4:

$$\hat{\rho} = \text{Avg}(\hat{\Sigma}_{xy}) \tag{5}$$

The sample Pearson correlation coefficient can be also written into the following form:

$$cor(X, Y) = \frac{\sum_{j=1}^{n}(\bar{x}_j - \bar{x})(\bar{y}_j - \bar{y})}{(n-1)S_X S_Y}, \tag{6}$$

where S_X and S_Y are standard deviations of X and Y respectively. When there is no replicate ($m_1 = m_2 = 1$), the correlation matrix Σ is reduced to a 2 by 2 matrix with diagonal elements equal to 1 and off-diagonal elements equal to ρ. It is easy to show from Eq. 4 that

$$\hat{\rho} = \frac{\sum_{j=1}^{n}(\bar{x}_j - \bar{x})(\bar{y}_j - \bar{y})}{nS_X S_Y} \tag{7}$$

Hence we derive the connection between the two estimators when there is no replicate as follows:

$$\hat{\rho} = \frac{n-1}{n} cor. \tag{8}$$

2.3 Statistical inference procedures

For very small sample data, eg, $n < 4$, we recommend using all permutations to approximate the null distribution (see Manuscript for detail). For larger sample data, we provide a Likelihood Ratio (LR) test. For moderate to large sample data, we provide a LRT procedure for testing the hypothesis that the

multivariate correlation ρ vanishes. Under the multivariate normal distribution assumption, $Z_j \sim N(\mu, \Sigma)$, and we test the following hypothesis:

$$H_0 : Z \in N(\mu, \Sigma_0) \text{ versus } H_\alpha : Z \in N(\mu, \Sigma_1). \tag{9}$$

Here, both Σ_0 and Σ_1 are $(m_1 + m_2) \times (m_1 + m_2)$ matrices, where m_1 and m_2 are number of replicates for biomolecule X and Y and $\Sigma_0 = \begin{pmatrix} \Sigma_x & \mathbf{0}_{m_1} \\ \mathbf{0}_{m_2}^T & \Sigma_y \end{pmatrix}$, $\Sigma_1 = \begin{pmatrix} \Sigma_x & \Sigma_{xy} \\ \Sigma_{xy}^T & \Sigma_y \end{pmatrix}$, where Σ_x and Σ_y, with diagonal elements identity and all the other entries being ρ_x and ρ_y respectively. $\mathbf{0}_{m_1}$ is a $m_1 \times m_1$ zero matrix and $\mathbf{0}_{m_2}$ is a $m_2 \times m_2$ zero matrix, that is, under the null hypothesis, the intermolecule correlation ρ vanishes. Σ_{xy} is a $m_1 \times m_1$ matrix with all entries equal to ρ. Likewise Σ_{xy}^T is a $m_2 \times m_2$ matrix with all entries equal to ρ . The Likelihood Ratio (LR) statistic for testing the two different correlation structures can be derived as follows:

$$\wedge = \frac{|\hat{\Sigma}_0|^{-n/2} e^{-\frac{1}{2}\sum_{j=1}^{n}(Z_j-\hat{\mu})'(\hat{\Sigma}_0)^{-1}(Z_j-\hat{\mu})}}{|\hat{\Sigma}_1|^{-n/2} e^{-\frac{1}{2}\sum_{j=1}^{n}(Z_j-\hat{\mu})'(\hat{\Sigma}_1)^{-1}(Z_j-\hat{\mu})}}. \tag{10}$$

Note that for the test to be a true LRT, all the estimated quantities $\hat{(\cdot)}$ in the above formula should be MLE's. In Section 2.2, Eqs. 2, 3 and 4 give the formula of MLE's of the mean vector and the correlation matrix under H_α. The MLE of the correlation matrix under H_0 can be determined as $\hat{\Sigma}_0 = \begin{pmatrix} \hat{\Sigma}_x & O \\ O & \hat{\Sigma}_y \end{pmatrix}$, where

$$\hat{\Sigma}_x = \frac{1}{n}\sum_{j=1}^{n}(X_j - \hat{\mu}_x)(X_j - \hat{\mu}_x)', \tag{11}$$

$$\hat{\Sigma}_y = \frac{1}{n}\sum_{j=1}^{n}(Y_j - \hat{\mu}_y)(Y_j - \hat{\mu}_y)'. \tag{12}$$

The LR statistic, denoted by G^2, is therefore:

$$G^2 = -2\log\wedge = n[trM - \log|M| - (m_1 + m_2)], \tag{13}$$

where $M = (\hat{\Sigma}_0)^{-1}\hat{\Sigma}_1$. The LR statistic is asymptotically chi-square distributed with $2(m_1 * m_2)$ degrees of freedom under H_0.

3 Data Analysis Examples

In this example, we will analyze a subset of 205 genes whose expression were profiled using 4 replicates under 20 physiological/genetic conditions [Medvedovic *et al.*, 2004]. The whole data was initially reported in [Ideker *et al.*, 2000] We first estimate all pairwise correlation and then we use 1-correlation as distance measure to cluster these genes. Medvedovic et al, 2004 [Medvedovic *et al.*, 2004] has already classified the 205 genes into 4 functional groups according to their GO annotations. The four classes are:

- Biosynthesis; protein metabolism and modification
- Energy pathways; carbohydrate metabolism; catabolism
- Nucleobase, nucleoside, nucleotide and nucleic acid metabolism
- Transport

The membership of 205 genes were stored in the internal data "true.member". We were able to compare the performance of our multivariate correlation estimator versus Pearson correlation estimator through hierarchical clustering by comparing clustering results to the "external knowledge" above.

```
> library(CORREP)
> data(gal_all)
> gal_avg <- apply(gal_all, 1, function(x) c(mean(x[1:4]), mean(x[5:8]),
+     mean(x[9:12]), mean(x[13:16]), mean(x[17:20]), mean(x[21:24]),
+     mean(x[25:28]), mean(x[29:32]), mean(x[33:36]), mean(x[37:40]),
+     mean(x[41:44]), mean(x[45:48]), mean(x[49:52]), mean(x[53:56]),
+     mean(x[57:60]), mean(x[61:64]), mean(x[65:68]), mean(x[69:72]),
+     mean(x[73:76]), mean(x[77:80])))
> M1 <- cor(gal_avg)
```

The above code is to calculate a 205 by 205 correlation matrix using Pearson correlation coefficient implemented in R function `cor`. Note that we have to average over replicates before the straight-forward application of Pearson correlation coefficient can be done. As we mentioned in the paper, the data used in this example is relatively "clean" data (see the boxplot below for distribution of within-replicate correlation), which is not favorable condition to apply our estimator, however, the following clustering results show that it still significantly outperforms Pearson correlation coefficient.

```
> x <- gal_all[1, ]
> x <- cbind(t(x[1:4]), t(x[5:8]), t(x[9:12]), t(x[13:16]), t(x[17:20]),
+     t(x[21:24]), t(x[25:28]), t(x[29:32]), t(x[33:36]), t(x[37:40]),
+     t(x[41:44]), t(x[45:48]), t(x[49:52]), t(x[53:56]), t(x[57:60]),
+     t(x[61:64]), t(x[65:68]), t(x[69:72]), t(x[73:76]), t(x[77:80]))
> for (j in 2:205) {
+     y <- gal_all[j, ]
+     y <- cbind(t(y[1:4]), t(y[5:8]), t(y[9:12]), t(y[13:16]),
+         t(y[17:20]), t(y[21:24]), t(y[25:28]), t(y[29:32]), t(y[33:36]),
+         t(y[37:40]), t(y[41:44]), t(y[45:48]), t(y[49:52]), t(y[53:56]),
+         t(y[57:60]), t(y[61:64]), t(y[65:68]), t(y[69:72]), t(y[73:76]),
+         t(y[77:80]))
+     x <- rbind(x, y)
+ }
> boxplot(cor(x))
> rawdata <- x
> stddata <- apply(rawdata, 1, function(x) x/sd(x))
> stddata <- t(stddata)
```

The above code is to reshape the data to be compatible with functions implemented in this package. This step is not absolutely necessary if your data is

already in the right format, for example, columns correspond to conditions and rows correspond to (replicated) variables. For example, a 820 by 20 matrix for this data. **Moreover, data has to be standardized by making variance of each row (gene) equals to 1. The standardization is VERY important and must be followed**.

```
> M <- cor.balance(stddata, m = 4, G = 205)
```

The above code is to calculate 205 by 205 correlation matrix using the new multivariate correlation estimator implemented in R function `cor.balance`. Note that there are equivalent number of replicates for this data (balanced). For unbalanced data, R function `cor.unbalance` should be used.

```
> row.names(M) <- row.names(M1)
> colnames(M) <- colnames(M1)
```

We next use 1-correlation as distance measure to cluster the genes into 4 groups, and compare the consistency between these 4 groups and pre-defined 4 groups.

```
> M.rep <- 1 - M
> M.avg <- 1 - M1
> d.rep <- as.dist(M.rep)
> d.avg <- as.dist(M.avg)
```

The above code calculates distance matrix for hierarchical clustering using both Pearson correlation coefficient and multivariate correlation estimator.

```
> library(e1071)
> library(cluster)
> data(true.member)
> g.rep <- diana(d.rep)
> g.avg <- diana(d.avg)
```

The above code does hierarchical clustering in a top-down fashion ie, Diana, and classifies all 205 genes into a number of clusters. We recommend using top-down method in this case since we are interested in identifying a few (4) large clusters. For other data, if you are interested in identifying a lot of small clusters, we recommend using agglomerative hierarchical clustering methods. See below for examples.

```
> h.rep <- hclust(d.rep, method = "complete")
> h.avg <- hclust(d.avg, method = "complete")
```

The R function `hclust` can be applied to perform bottom-up hierarchical clustering. In addition to choosing a proper distance measure, we will need to determine a method to measure distance between objects such as are "complete", "average", "single". Option "average" seems to be robust in most of cases.

```
> member.rep.k4 <- cutree(g.rep, k = 4)
> member.avg.k4 <- cutree(g.avg, k = 4)
> classAgreement(table(member.avg.k4, as.matrix(true.member)))

$diag
[1] 0
```

```
$kappa
[1] -0.3751636

$rand
[1] 0.9437111

$crand
[1] 0.8792274
```

```
> classAgreement(table(member.rep.k4, as.matrix(true.member)))
```

```
$diag
[1] 0.06341463

$kappa
[1] -0.2491669

$rand
[1] 0.9751315

$crand
[1] 0.9473288
```

The above code retrieves the membership of 205 genes as determined by divisive clustering, and accesses consistency of the memberships with the "true" cluster membership (external knowledge) using adjusted RAND index. The function classAgreement that was used to calculate adjusted RAND index is from "e1071" package. It is obvious that clustering results based on multivariate correlation estimator are more consistent to the external knowledge (higher adjusted RAND index).

We can also examine the "quality" of clusters using, for example, silhouette plot. Here quality roughly means that ratio between within-cluster deviation and between-cluster deviation. Although it is statistically true that good quality clusters indicate better clustering performance, it is not necessarily true biologically. Good quality clusters are only a sufficient condition for good clustering performance but not a necessary condition.

```
> sil.rep4 <- silhouette(member.rep.k4, d.rep)
> sil.avg4 <- silhouette(member.avg.k4, d.avg)
> plot(sil.rep4, nmax = 80, cex.names = 0.5)
> plot(sil.avg4, nmax = 80, cex.names = 0.5)
```

References

[Medvedovic *et al.*, 2004] Medvedovic, M., Yeung, K.Y., Bumgarner, R.E. (2004) Bayesian mixtures for clustering replicated microarray data. *Bioinformatics*, **20**, 1222-1232.

[Ideker *et al.*, 2000] Ideker, T., Thorsson, V., Siegel, A.F. and Hood, L.E. 2000. Testing for differentially-expressed genes by maximum-likelihood analysis of microarray data. *J. Comput. Biol.*, **7**, 805-817.

[Zhu and Li, 2007] Zhu, D and Li, Y. 2007. Multivariate Correlation Estimator for Inferring Functional Relationships from Replicated 'OMICS' Data. Submitted.