

The LiquidAssociation Package

Yen-Yi Ho

October 17, 2016

1 Introduction

The LiquidAssociation package provides analytical methods to study three-way interactions. It incorporates methods to examine a particular kind of three-way interaction called liquid association (LA). The term liquid association was first proposed by ?). It describes the extend to which the correlation of a pair of variables depends on the value of a third variable. The term “liquid” was used in contrast with “solid” to emphasize that the association between a pair of variables changes according to the value of a third variable. Li first reported the existence of liquid association in the yeast cell cycle gene expression data set by ?). Furthermore, the liquid association phenomena could potentially be found in other kind of data as well. Building on the ground-breaking work by Li, ?) extended liquid association to accommodate more intricate co-dependencies among the 3 variables, calling the expanded statistic *generalized liquid association*, or GLA for short.

This software package provides functions to implement the estimation of liquid association through two approaches: the direct and the model-based estimation approach. For the model-based approach, we introduce the conditional normal model (CNM) and provided a generalized estimating equations (GEE)-based estimation procedure. In addition, we provide functions to perform hypothesis testing using direct estimate (sGLA) and model-based estimates (GEEb5) in this package.

2 Simple Usage

Here we present a typical work flow to investigate liquid association using a gene triplet data. We start with load the R package and the example data. In this package, we use the yeast cell cycle gene expression data by Spellman (1998). The data can be obtained through package *yeastCC*. The annotation package for the yeast experiment *org.Sc.sgd.db* can be obtained through Bioconductor.

```
> library(LiquidAssociation)
> library(yeastCC)
> library(org.Sc.sgd.db)
> data(spYCCES)
> lae <- spYCCES[, -(1:4)]
> ### get rid of the NA elements
> lae <- lae[apply(is.na(exprs(lae)), 1, sum) < ncol(lae)*0.3, ]
> probname <- rownames(exprs(lae))
> genes <- c("HTS1", "ATP1", "CYT1")
> geneMap <- unlist(mget(genes, revmap(org.Sc.sgd.GENENAME), ifnotfound=NA))
> whgene <- match(geneMap, probname)
```

After removing genes with high missing percentage, we keep 5721 genes for further analysis. We are interested in three genes: HTS1, ATP1, and CYT1 that are involved in the Yeast electron transport pathway. We would like to know whether the correlation of HTS1 and ATP1 gene can be modulated by the level of CYT1 gene.

```

> data<-t(exprs(lae[whgene,]))
> eSetdata<-lae[whgene,]
> data<-data[!is.na(data[,1]) & !is.na(data[,2]) & !is.na(data[,3]),]
> colnames(data)<-genes
> str(data)

num [1:66, 1:3] 0.3 -0.25 0.18 0.05 0.05 -0.16 0.26 0.14 0.03 0.01 ...
- attr(*, "dimnames")=List of 2
..$ : chr [1:66] "alpha_0" "alpha_7" "alpha_14" "alpha_21" ...
..$ : chr [1:3] "HTS1" "ATP1" "CYT1"

```

We notice that after removing missing observations, there are 66 observations left. We can use the *plotGLA* function to examine whether the correlation of the first two genes changes according to the level of the third gene. The *plotGLA* function produces scatter plot conditioning on the level of a third gene, X_3 . We can specify which column in the data to be the third gene using the argument *dim*. For example, we can specify HTS1 and ATP1 gene as the first two modulated genes, and CYT1 as the third controller gene by setting *dim*=3. The *cut* argument is used to specify the number of grid points over the third controller gene. In the *plotGLA* function, the input *data* can be in ExpressionSet class as well.

```

> plotGLA(data, cut=3, dim=3, pch=16, filen="GLAplot", save=TRUE)

```

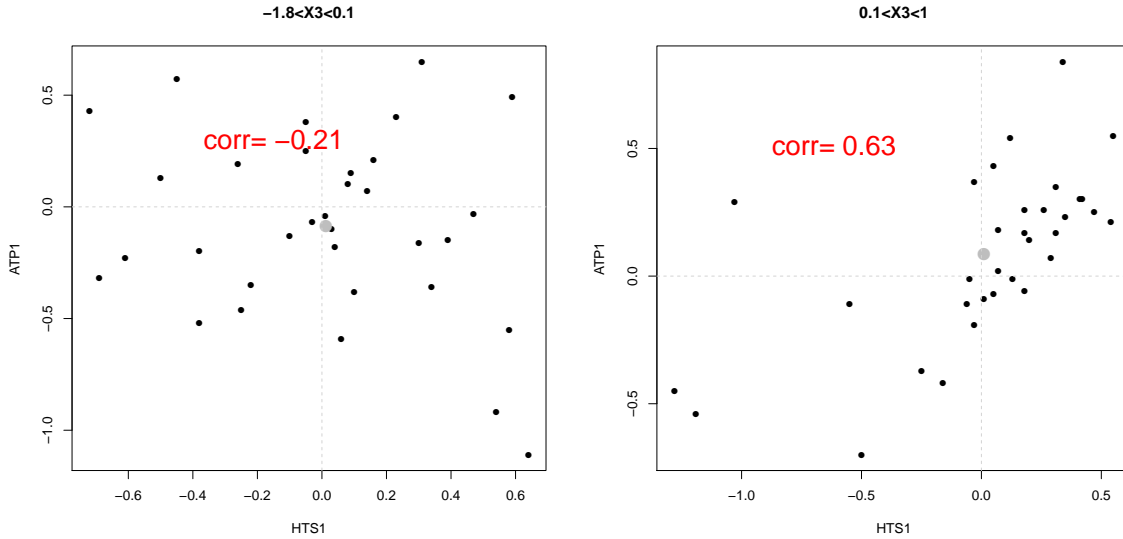


Figure 1: Conditional distributions of (HTS1, ATP1 | CYT1) according to the gene expression level of CYT1.

According to Figure ??, we find the evidence for the existence of liquid association among the triplet since the correlation of HTS1, and ATP1 gene changes from -0.21 to 0.63. We can further perform estimation and hypothesis testing to quantify the strength of liquid association. As described in Li (2002) and Ho et al (2009), the calculation of the liquid association measures assumes all three variable are standardized with mean 0 and variance 1 and the third gene follows normal distribution. Hence in this package, within the *GLA*, *LA*, *CNM.full*, *CNM.simple*, *getsGLA*, *getsLA* functions the standardization and normalization steps are performed internally (so that the three variables are with mean 0 and variance 1, and the third variable is normally distributed through quantile normalization).

We now use GLA static to robustly measure the magnitude of liquid association. The measure ranges from $-\sqrt{2/\pi} \approx -0.798$ to $\sqrt{2/\pi}$. When $GLA=0$, it means that the correlation of the two modulated genes does not change according to the level of the third controller gene, hence there is no evidence of LA. In addition, when $GLA > 0$, it indicates that the correlation of the first two genes increases with increasing

value of the third gene and vice versa. We use the function *GLA* to calculate the GLA estimate for a given triplet data as follows:

```
> LAest<-LA(data)
> GLAest<-rep(0,3)
> for ( dim in 1:3){
+   GLAest[dim]<-GLA(data, cut=4, dim=dim)
+ }
> LAest
```

```
LA(HTS1,ATP1,CYT1)
      0.5633369
```

```
> GLAest
```

```
[1] 0.4162624 0.3085295 0.3219779
```

The data argument in these function can also be in ExpressionSet class as follows:

```
> eSetGLA<-GLA(eSetdata, cut=4, dim=3, geneMap=geneMap)
> eSetGLA
```

```
GLA(HTS1,ATP1|CYT1)
      0.3219779
```

In the above example, we calculate three GLA estimates by sequentially changing the third controller gene. In addition, the three-product-moment estimator proposed by Li (2002) can also be calculated using the *LA* function as shown above. In the example, we find noticeable differences between the three-product-moment and GLA estimates, *LAest* and *GLAest*, respectively. As described in Ho (2009), the GLA estimator is more robust than *LA* in the sense that it could still correctly capture liquid association even when the marginal mean and variance depend on the third variable.

The second approach to estimate liquid association is using the model-based estimator. We fit the CNM using the function *CNM.full* as shown below. Furthermore, the CNM model is written as:

$$\begin{aligned} X_3 &\sim N(\mu_3, \sigma_3^2) \\ X_1, X_2 | X_3 &\sim N\left(\begin{pmatrix} \mu_1 \\ \mu_2 \end{pmatrix}, \Sigma\right). \end{aligned}$$

where $\Sigma = \begin{pmatrix} \sigma_1^2 & \rho\sigma_1\sigma_2 \\ \rho\sigma_1\sigma_2 & \sigma_2^2 \end{pmatrix}$. The mean vector (μ_1, μ_2) and variance matrix Σ depend on the level of X_3 as written below:

$$\begin{aligned} \mu_1 &= \beta_1 X_3, \\ \mu_2 &= \beta_2 X_3, \\ \log \sigma_1^2 &= \alpha_3 + \beta_3 X_3, \\ \log \sigma_2^2 &= \alpha_4 + \beta_4 X_3, \\ \log \left[\frac{1+\rho}{1-\rho} \right] &= \alpha_5 + \beta_5 X_3. \end{aligned}$$

```
> FitCNM.full<-CNM.full(data)
> FitCNM.full
```

```
Model: CNM(HTS1,ATP1|CYT1)
      estimates      san.se      wald      p value
a3 -0.01944285 0.20975976 0.008591625 9.261490e-01
```

```

a4 -0.27887616 0.16905326 2.721295365 9.901763e-02
a5 0.87944227 0.43552192 4.077506175 4.345775e-02
b1 -0.02107489 0.06517541 0.104559409 7.464253e-01
b2 0.42294929 0.07987766 28.036635027 1.190404e-07
b3 0.06814325 0.21511309 0.100348770 7.514115e-01
b4 -0.18465426 0.20408239 0.818667618 3.655700e-01
b5 1.32577318 0.49759016 7.098962665 7.712858e-03

```

The main parameter of interest in the CNM for examining the existence of liquid association is b_5 . As shown in the result above, the GEEb5 Wald test statistic is displayed in the 8th row 3rd column (GEEb5=7.1) with p value 0.008. We conclude there is statistically significant evidence that indicates the existence of liquid association for (HTS1, ATP1 | CYT1). We can also perform hypothesis testing using the direct estimate approach using the *getsGLA* function as follows:

```

> sGLAest<-getsGLA(data, boots=20, perm=50, cut=4, dim=3)
> sGLAest

      sGLA  p value
3.311775 0.000000

> sLAest<-getsLA(data,boots=20, perm=50)
> sLAest

      sLA  p value
5.183672 0.000000

```

where the argument *boots* specify the number of bootstrap iteration for estimating the bootstrap standard error. For demonstration purpose, we set boots to 20. The *perm* argument specifies the number of iterations to calculate permuted p value. In addition, we can also perform hypothesis testing using sLA statistic based on the three-product-moment estimator proposed by Li (2002). In real applications, sGLA is likely to be more robust than sLA. We draw similar conclusion using sGLA test statistic and GEEb5 with p value less than 0.01.

3 Extended Examples

In this section, we demonstrate an typical analysis procedure using the entire gene expression data set. In this example, we first filter out genes with small variances and keep 150 genes for further analysis.

```

> lae<-t(exprs(lae))
> V<-apply(lae, 2, var, na.rm=TRUE)
> ibig<-V > 0.5
> sum(ibig)

[1] 150

> big<-which(ibig)
> bigtriplet<-lae[,big]
> dim(bigtriplet)

[1] 73 150

```

We can annotation the ORF ID in the data set to their gene names as follows:

```

> x <- org.Sc.sgdGENENAME
> mappedgenes <- mappedkeys(x)
> xx <- as.list(x[mappedgenes])
> mapid<-names(xx)

```

```

> orfid<-colnames(bigtriplet)
> genename1<-xx[match(orfid, mapid)]
> imap<-which(sapply(genename1, length) !=1)
> bigtriplet<-bigtriplet[,-imap]
> colnames(bigtriplet)<-genename1[-imap]

```

After removing ORF probe set that can not be mapped to genes, a total number of 400995 possible triplet combinations that can be generated by the 135 genes. We now calculate the GLA estimates for all possible triplet combinations. Here, we demonstrate calculating GLA estimates for triplet combination #1 to #100.

```

> num<-choose(ncol(bigtriplet),3)
> pick<-t(combn(1:ncol(bigtriplet),3))
> GLAout<-matrix(0, nrow=100, 3)
> for ( i in 1:100){
+   dat1<-bigtriplet[,pick[i,]]
+   for ( dim in 1:3 ){
+     GLAout[i,dim]<-GLA(dat1, cut=4, dim=dim)
+   }
+ }

```

We would like to find triplets with large GLA estimates. For example, we can choose triplet with GLA greater than 0.2 as follows:

```

> GLAmax<-apply(abs(GLAout),1, max)
> imax<-which(GLAmax > 0.20)
> pickmax<-pick[imax,]
> GLAmax<-GLAout[imax,]
> trip.order<-t(apply(t(abs(GLAmax)), 2, order, decreasing=TRUE))

```

Furthermore, we perform hypothesis testing using the first triplet as demonstration. Based on the analysis result, we can not reject the null hypothesis that the liquid association is 0. We draw the same conclusion using GEEb5 and sGLA statistics.

```

> whtrip<-1
> data<-bigtriplet[,pickmax[whtrip,]]
> data<-data[,trip.order[whtrip,]]
> data<-data[!is.na(data[,1]) & !is.na(data[,2]) & !is.na(data[,3]),]
> data<-apply(data,2,qnorm2)
> data<-apply(data,2, stand)
> FitCNM1<-CNM.full(data)
> FitCNM1

```

Model: CNM(SEO1,RFA1|SSA1)

	estimates	san.se	wald	p value
a3	-0.02128987	0.1545360	0.01897957	0.890425061
a4	-0.13810219	0.1666190	0.68699255	0.407189215
a5	-0.32754023	0.2640596	1.53859884	0.214826486
b1	0.07713727	0.1149752	0.45011195	0.502281794
b2	-0.05877725	0.1083481	0.29429007	0.587484380
b3	-0.04416097	0.1449181	0.09286070	0.760571391
b4	-0.49537099	0.1761766	7.90614258	0.004926721
b5	-0.71803322	0.4028203	3.17735935	0.074665304

```

> sGLA1<-getsGLA(data, boots=20, perm=50, cut=4, dim=3)
> sGLA1

```

	sGLA	p value
	-1.209106	0.160000

```

>

```

4 Reference

5 Session Information

- R version 2.8.1 (2008-12-22), x86_64-unknown-linux-gnu
- Locale: LC_CTYPE=en_US.iso885915;LC_NUMERIC=C;LC_TIME=en_US.iso885915;LC_COLLATE=en_US.iso885915;LC_
- Base packages: base, datasets, graphics, grDevices, methods, stats, tools, utils
- Other packages: AnnotationDbi 1.4.2, Biobase 2.2.2, DBI 0.2-4, geepack 1.0-13, LiquidAssociation 1.0.4, org.Sc.sgd.db 2.2.6, RSQLite 0.7-1, yeastCC 1.2.6