Scatterplot clustering for the integrative analysis of expression and methylation data

M. Carme Ruiz de Villa, Francesc Carmona
Diego Arango del Corro, Alex Sánchez

2014-09-19

Resumen

En este trabajo se utilizan las técnicas de

Universitat de Barcelona

Dos Campus d'Excel·lència Internacional:
B:KC Barcelona Knowledge Campus
HUBC Health Universitat de Barcelona Campus

1. Métodos estadísticos

1.1. Gene-specific methylation on-off threshold

Methylation is often described as a binary on-off signal, and it is widely recognized that methylation represses gene expression. Typically, if a gene is controlled by its methylation, its expression is low when methylated. On the other hand, when unmethylated, its expression can be either high or low. Since measurements for methylation and expression are both continuous, a biaxial plot of these two signals will exhibit an L-shape pattern.

If we truly believe that methylation is binary, there are two implications:

1. the reflection point of the L-shape is an appropriate choice to binarize methylation data, and
2. conditioning on the binarized on-off methylation status, the continuous valued methylation data and expression data should be independent,

which motivates Liu(2012) to quantify the L-shape pattern using conditional mutual information (MI).

1.2. Conditional Mutual Information

Two questions: which genes exhibit L-shape, and what is the optimal threshold for binarizing methylation data for each L-shape gene.

To determine whether methylation and expression of a gene exhibit an L-shape, we compute the conditional Mutual Information (MI) for different choices of threshold to binarize the methylation data.

If we consider the continuous valued methylation and expression data as two random variables X and Y, and denote a nominal threshold as t, the conditional MI can be written as a weighted sum of MIs on the two sides of the threshold.

$$cMI(t) = I(X, Y|X > t)P(X > t) + I(X, Y|X \leq t)P(X \leq t)$$

1.3. Optimal threshold

When t is 0 or 1, cMI equals to the mutual information derived from all data points.
For an L-shape gene, as t moves from 0 to 1, $cMI(t)$ first decreases and then increases, and its value approaches zero when t coincides with the reflection point. Therefore,

The ratio $r = \frac{\text{mín}\{cMI(t)\}}{cMI(0)}$ for an L-shape gene is small, and $t^* = \text{argmin}\{cMI(t)\}$ is the optimal threshold for dichotomizing the methylation data of this gene.

1.4. Joint distribution estimator

To estimate the MI terms we use a kernel-based estimator, which constructs a joint probability distribution by applying a Gaussian kernel to each data point, and estimates the MI based on the joint distribution. The estimator is as follows:

$$I(X, Y) = \frac{1}{M} \sum_{i=1}^{M} \log \frac{M \sum_{j=1}^{M} e^{-\frac{1}{2h^2}((x_i - x_j)^2 + (y_i - y_j)^2)}}{\sum_{j=1}^{M} e^{-\frac{1}{2h^2}(x_i - x_j)^2} \sum_{j=1}^{M} e^{-\frac{1}{2h^2}(y_i - y_j)^2}}$$

where h is a tuning parameter for the kernel width and empirically set $h = 0{,}3$.

1.5. L-shapes

We filtered for L-shapes using a combination of three criteria:

- the ratio $r < 0{,}25$
- unconditioned MI $cMI(0) > 0{,}1$
- the median expression on the left side of the optimal threshold t^* is higher than the median expression on the right side.

The parameters are chosen according to a random permutation test (see Liu(2012)).

According to the above criteria, a total of 641 genes are selected to be L-shape genes.

2. El análisis

2.1. Los datos

A partir del archivo `Data_CorrelationsTots.csv` se monta el `data.frame` que contiene los datos.
Una vez limpio y escalados los datos de expresión, se ha grabado el conjunto de datos en el archivo `DataNet.RData` y así podemos cargar los datos directamente.
Las 30 primeras columnas son los datos de Metilación y las últimas 30 son datos de Expresión.

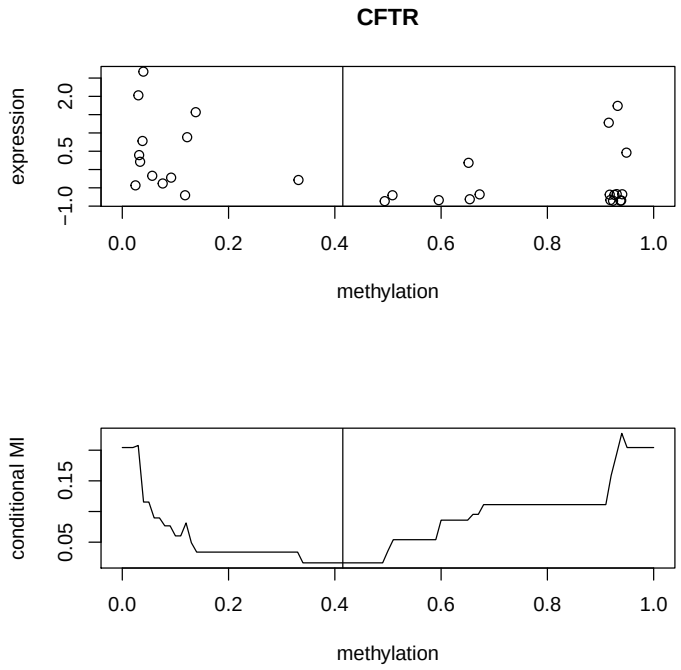
En primer lugar procedemos a eliminar de la base de datos aquellos genes que contienen algún valor *missing*. En concreto son 110. Si los eliminamos quedan 11746 genes remanentes en la base de datos.

2.2. Cálculo de la *conditional Mutual Information*

En primer lugar se han creado un par de funciones que calculan la *conditional Mutual Information* entre las muestras de metilación y expresión para cada gen. Con ellas se procede a la selección de los casos (genes) que verifican los dos primeros criterios. Los genes que verifican $r < 0{,}25$ son 5040 y los que además verifican que $cMI(0) > 0{,}1$ son 752.

A continuación procedemos a seleccionar los genes que han quedado con el tercer criterio.
Dichos genes seleccionados por los tres criterios estan en el archivo `genesSel.RData` y son 643.

En el siguiente gráfico vemos un ejemplo de gen seleccionado:



3. Clustering approach

Para que el análisis cluster no se pierda con muchos genes que no tienen la forma L deseada, en primer lugar procedemos a eliminar aquellos genes que De forma que nos quedamos con 473.
El siguiente paso consiste en calcular las correlaciones de Spearman para cada gen y seleccionar aquellos que tengan una correlación negativa significativa.
Hacemos una lista de matrices met/expr para cada gen.

3.1. Análisis cluster

3.1.1. Cálculo de los b-splines

Aplicamos la función `splines` a los datos ajustando con `degree=2` y obtenemos una lista con los coeficientes de los splines de cada gen.

3.1.2. Cluster analysis

Posteriormente se ha realizado un análisis de cluster jerárquico utilizando los coeficientes de los b-splines como variables para realizar la agrupación. El número de clusters óptimo es 4 con una distribución:

```
Hclustgroups0.5
 1  2  3  4
94 16 18  3
```

Seleccionamos los 4 clusters obtenidos cortando el árbol a 0.5

La representación de los genes de cada cluster se encuentra en los ficheros Cluster1...Cluster4 de la carpeta figuras.

Finalmente unimos los genes del cluster 1 y 2 ya que ambos corresponden al patrón de genes regulados por metilación tipo L. Para realizar la comparación de ambos métodos realizamos las siguientes intersecciones: Cluster1 con CMI (Tall1) y la unión de Cluster 1 y 2 con CMI (Tall2).

```
> genescluster1 <- names(Hclustgroups[Hclustgroups==1])
> genescluster2 <- names(Hclustgroups[Hclustgroups==2])
> genescriteri3 <- genes.sel
> tall1 <- intersect(genescriteri3,genescluster1)
> tall2 <- intersect(genescriteri3,union(genescluster1,genescluster2))
> gensComuns <- list(tall1=tall1,tall2=tall2)
> length(tall1)

[1] 75

> length(tall2)

[1] 87
```

El gráfico de los genes de las intersecciones se guardarán en un fichero aparte en la carpeta figuras.

```
pdf
  2

pdf
  2
```

Referencias

[1] R Development Core Team (2005). R: A language and environment for statistical computing, reference index version 2.14.0. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0, `http://www.R-project.org`