



Version 1.0, March 2006

Cristian I. Castillo-Davis
&
Wenxuan Zhong

Copyright © 2006 by Cristian I. Castillo-Davis and Wenxuan Zhong. This software package is provided "as is" without warranty of any kind. In no event shall the author be held responsible for any damage resulting from the use of this software. The program package, including source code, executables, example data sets, and this documentation, is distributed free of charge under the terms of the GNU General Public License (<http://www.gnu.org/copyleft/gpl.html>)

Suggested citation:

Ping Ma, Cristian I. Castillo-Davis, Wenxuan Zhong, Jun S. Liu. 2006.
A Data-Driven Clustering Method for Time Course Gene Expression Data.
Nucleic Acids Research. 34:1261-1269

<http://www.genemerge.bioteam.com/SSClust.html>

Wenxuan Zhong
Department of Statistics
Harvard University
1 Oxford Street
Cambridge, MA 02138, USA
Email: wenxuan@stat.harvard.edu

Introduction

Smoothing Spline Clustering (SSC) is a statistical method for clustering time-series gene expression data.

In particular, Smoothing Spline Clustering is useful for clustering genes in microarray experiments performed over several time-points, for example, over the course of development, a drug treatment, or other temporally based experiments.

SSCLUST is the **R** implementation of the SSC method. **R** is a free statistical programming language and environment.

Before Getting Started

All platforms

Download and install **R** for your operating system from:

<http://cran.us.r-project.org/>

There is a help manual here also to help you install **R** on your system.

Getting Started

Macintosh OS X

- (1) Download the program and save to a convenient location. Unpack the archive **SSClust1.0.tar.gz**. To unpack the file, open a "Terminal" window (Programs-> Utilities-> Terminal) and move to the SSClust folder using the command line command "cd" (change directory). Simply type "cd <directory>" to change directories). For example,

```
cd /hardrive/Desktop or cd SSClust.
```

Use `cd ..` to move up one folder.

- (2) Next, unpack the archive in the relevant folder using the command:

```
tar xvzf SSClust.tar.gz
```

A folder Called SSClust will be generated with all files needed in it.

- (3) Run **R** by typing R in the terminal window.

Linux

- (1) Download the SSCLUST to a convenient location. Unpack the archive **SSClust1.0.tar.gz** using

```
tar xvzf SSClust.tar.gz
```

A folder Called SSClust will be generated with all files needed in it.

- (2) Move to the SSClust folder using the `cd` command.
- (3) Run **R** by typing R in the terminal window.

Windows 95/98/NT/XP

- (1) Download and unzip the file **SSClust1.0.zip**.
A folder Called SSClust will be generated with all files needed in it.
- (2) Open R by double-clicking on the **R** icon

Installing required R Packages (all platforms)

When you have 'admin' or 'root' privileges

At the R command line (>) type the following:

```
> chooseCRANmirror()  
> install.packages(c("mvtnorm", "gss"))
```

When you DO NOT have 'admin' or 'root' privileges

At the R command line (>) type the following:

```
> chooseCRANmirror()  
> install.packages("mvtnorm", path)  
> install.packages("gss", path)
```

where *path* is the location were you wish to install the library locally.

For example:

```
install.packages("gss", 'C:/Documents and Settings/Cristian/Desktop/')  
install.packages("gss", 'HD/home/Sarah/')
```

TESTING THE INSTALLATION

Make the SSC directory the 'working directory' in **R** using the appropriate path with the *set working directory* (setwd) command:

```
> setwd("C:/Documents and Settings/Cristian/Desktop/SSC/")
```

On a PC you can also change the 'working directory' to **SSC** using the pull-down menu: File-> Change Dir -> Browse

To check if everything was installed correctly, run the test file "**SSClust.test.R**" by typing the following at the **R** command prompt.

```
> source("SSClust.test.R")
```

There should be no error messages and the output in the **R** window should look something like this:

```
[1] 1
[1] 2
[1] 3
[1] 4
Clusters = 4
BIC Score = 5680.25
```

Additionally, two image files will be created showing the raw expression curves, "raw_clusters.ps", and cluster mean curves with 95% confidence bands "cluster_mean_curves.ps", as well as a number of text files containing the names of all the genes in each cluster which are named cluster1.txt, cluster2.txt, ... clusterN.txt, and a file containing all the genes studied called "all_genes.txt." You can compare your results to those in the **Example** directory.

Running SSClust

MODIFYING THE 'MASTER CONTROL FILE'

SSCLUST is run in **R** on the command line. It uses a master control file that needs to be edited before you can analyze your data. The control file is also where the model parameters are specified and can be changed. These include:

nchain	- number of Markov chains used during the RCEM procedure, default is 5.
threshold	- RCEM threshold parameter c , default is 0.5.
nclust	- number of clusters

INPUT FILE FORMAT

The input file should be a simple tab delimited text file with headers for each column with no spaces in the heading names as follows (code missing data as "NA"):

Gene	time1	time2	time3	time4	time6	time7...
gene1	40.7	50.1	60.8	70.9	80.2	99.0
gene2	44.3	55.5	67.7	77.7	87.2	102.1
gene3	400.8	539.3	637.6	700.2	843.1	1560.3
gene4	323.4	453.3	45.8	67.2	43.9	332.9
...						

To specify the location of your expression data, type the following on the **R** command line, modifying the relevant path information and input file name:

```
> my.data <- read.table("/data/time.course.data.txt",  
header=T, row.names=1, sep="\t")
```

RUNNING SSCLUST

To run SSCLust simply type the following in **R**:

```
> source("SSClust.R")
```

CHOOSING THE OPTIMUM NUMBER OF CLUSTERS

SSCLUST is run once for each number of clusters, starting from the lowest to the highest, one at a time. Each time you run SSCLUST, be sure to save the cluster output files and record the BIC score.

Each time you run SSCLUST, increase the number of clusters by one by changing the value of **nclust** in the master control file and run SSCLUST again. Once you notice the BIC score start to rise for a while, stop. Determine what number of clusters yielded the lowest BIC score. This is the optimum number of clusters with respect to gene-to-cluster assignment and curve-fitting.

Understanding the Output

After each run of SSCLUST, two image files will be created, one showing the raw expression curves for all clusters and one showing the mean curves and 95% confidence bands for all clusters. Additionally, text files containing the names of all the genes in

each cluster will be generated; these can be used as input for functional genomic programs such as GeneMerge (<http://genemerge.bioteam.net/>).

<u>Output File</u>	<u>Description</u>
raw_clusters.ps*	-postscript file showing raw expression curves for each cluster.
cluster_mean_curves.ps*	- postscript file showing the mean curve (black) and 95% confidence bands (red) for each cluster.
cluster1.txt	- list of genes that belong to cluster 1
cluster2.txt	- list of genes that belong to cluster 2
cluster3.txt	- list of genes that belong to cluster 3
cluster4.txt	- list of genes that belong to cluster 4
cluster5.txt	- list of genes that belong to cluster 5
...	
all_genes.txt	- list of all the genes/objects that were clustered

* Adobe Acrobat Distiller can be used to convert .ps postscript files to PDF files or you can use an online ps-to-pdf service such as: <http://www.ps2pdf.com>.

OTHER OUTPUT INFORMATION

An R object file with name **output.Rdata** is generated each time SSCLUST is run. There are three R objects in this file: **output.clust** gives the membership labels for each gene/object. **output.center** gives the numerical value of the center mean curve for each cluster and **output.likeli** gives the log likelihood value for the current model.

Optimization

Remember that computation time increases with:

- **The number of genes being clustered.**
- **The number of RCEM chains run**
Increasing the number of RCEM chains will help prevent the algorithm from getting trapped in local maxima but increases computation time. The default is 5 chains.
- **The RCEM threshold value c .**
Increasing the RCEM threshold c will improve the speed of the algorithm at the expense of accuracy. The default value is 0.5.

You may want to experiment running SSCLUST with few RCEM chains and high c values, such as (0.5-0.7) when initially exploring your data.

Acknowledgements

Thanks to the **R** Development Core Team for the development of the **R** programming language and Professor Chong Gu for the development of the gss package. Special thanks to Ryan Jones for creating the super fresh SSC logo.