Study and Analysis of Gene Expression Clustering with Gaussian Mixed Effects Models and Smoothing

Tejal Upadhyay[1] Dr Samir Patel[2]
tejal.upadhyay@nirmauni.ac.in Samir.Patel@sot.pdpu.ac.in
Department of Computer Science and Engineering, Institute of Technology, Nirma University, S G Highway, Ahmedabad, Gujarat, India
Department of Computer Science and Engineering, School of Technology, Pandit Deendayal Petrolium University, Raisan, Gandhinagar, Gujarat, India

Summary
A large number of longitudinal studies measuring gene expression aim to stratify the genes according to their differential temporal behaviors. Genes with similar expression patterns may reflect functional responses of biological relevance. However, these measurements come with intrinsic noise which makes their time series clustering a difficult task. Here, we have shown how to cluster such data with mixed effects models with nonparametric smoothing spline fitting and is able to robustly stratify genes by their complex time series patterns. The article has, besides the main clustering methods, a set of functionalities assisting the user to visualize and assess the clustering results, and to choose the optimal clustering solution. The first part is about the introduction to gene expression, how time series can be applied and how the clustering is important to gene expression. The Gaussian mixed effect model is also explain. The second part is about the related work already done with some references. The third part is about our own process and workflow with diagram. How the clustering is applied and diagrams of different cluster sets. The fourth part is about results and discussion, how the silhouette analysis is important and using 3 clusters and 4 clusters how the data sets look like. The fifth part is shown with applications of clustering effects, how the yeast data sets can be divided into clusters etc. The sixth part shows the methodology of mixed Gaussian effects and smoothing splines. The last part is about conclusion and references.

Keywords
Clustering, Silhouette Analysis, Gaussian mixed effect model, Smoothing Splines

1. Introduction

Gene expression is the process by which the instructions in our DNA are converted into a functional product, such as a protein or other molecules is called Gene Expression. To study in detail, a large number of longitudinal measuring gene expressions aim to stratify the genes according to their differential temporal behaviors. Different types of Classification and Clustering algorithms can be applied to gene expression values. Genes with similar expression patterns may reflect functional responses of biological relevance.

Time series data analysis can have sometimes intrinsic noise and as per the article [1] B -Spline is used to interpolate between two points. In this article we have used the method of mixed-effects models with nonparametric smoothing spline fitting and is able to robustly stratify genes by their complex time series patterns. It is a set of functionalities assisting the user to visualize and assess the clustering results, and to choose the optimal clustering solution [2].

Time series analysis for the fact that data points taken over time may have an internal structure such as auto correction, trend or seasonal variation that should be accounted for.

Gaussian mixture models provide an appealing tool for time series modelling. By embedding the time series to a higher-dimensional space, the density of the points can be estimated by a mixture model. The model can directly be used for short-to-medium term forecasting and missing value imputation. The modelling setup introduces some restrictions on the mixture model, which when appropriately taken into account result in a more accurate model. Experiments on time series forecasting show that including the constraints in the training phase particularly reduces the risk of overfitting in challenging situations with missing values or a large number of Gaussian components.

2. Related work

Gaussian Process methods have been applied to gene expression time-series with several aims, such as to infer transcription regulation [Honkela] and to find dynamic differential expression. Park and Choi [3] also proposed a method based on hierarchical Gaussian processes Their presentation is conceptually similar, but with the objective of saving some computation in performing Gaussian process regression, and Behseta et al. [4] proposed a hierarchical Gaussian process model with application to neuronal Poisson-process intensities

Clustering gene expression time series is an application which has attracted a lot of interest. Analysis of time series

Manuscript received May 5, 2020
Manuscript revised May 20, 2020

clusters is an important tool in exploring and understanding gene networks, whilst incorporating knowledge of the timeseries into the model has the potential to improve the ability of the method to discern clusters. Dunson [3] proposed a DP-GP model In Dunson's model, a series of GP functions are drawn from a DP-GP prior, and each observation is then assigned to one of the functions. However Dunson makes no use of structure in the model: observations differ from the latent function draws only by white noise. Rasmussen and Ghahramani [5] Also presented a method which combined GPs using DPs, but this method used a gating approach to produce a mixture of experts model.

3. The proposed workflow

Load the gene expression data into a data frame. Set clustering parameters and perform clustering for different configurations (e.g., numbers of clusters).

For a chosen clustering configuration, perform stability analysis and retrieve the optimal clustering solution

TMixClust provides additional functions which allow the user to obtain information about clusters by accessing the attributes of a TMixClust object, generating informative plots or generating a comprehensive clustering report.

3.1 Load Data:

First we are loading the data and defining a data frame which contains the time series gene expression data. A time series gene expression data set contains an ensemble of time series vectors, each time series being associated to a gene. The input data frame has a number of rows equal to the number of time series vectors and a number of columns equal to the number of time points. Row names correspond to the time series names (i.e., gene names), while column names represent time points names (e.g., "2h", "4h", etc.). The data frame can contain missing values.

By time series, we denote a vector of expression values of a gene measured at different, successive time points. X = {100, 200, 300, 400} is an example of a time series constituted by measurements at 4 time points.

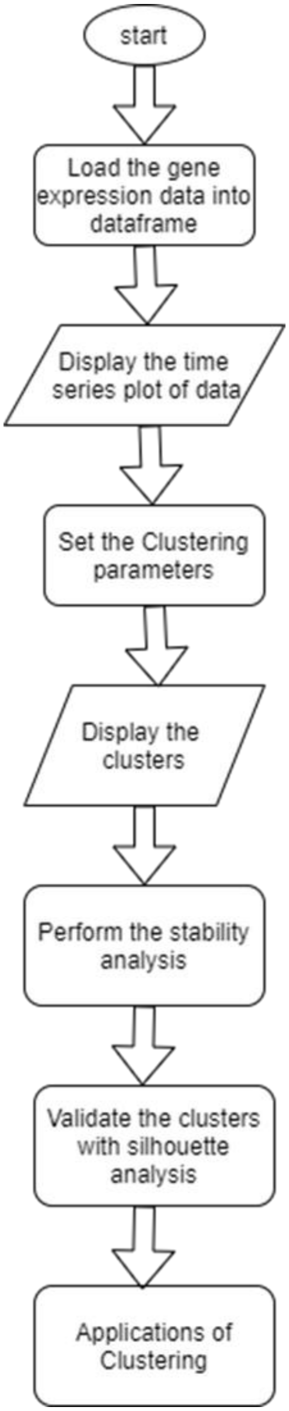
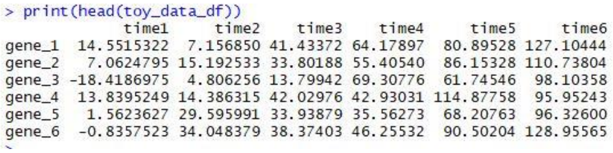

start
Load the gene expression data into dataframe
Display the time series plot of data
Set the Clustering parameters
Display the clusters
Perform the stability analysis
Validate the clusters with silhouette analysis
Applications of Clustering

Fig. 1 Block diagram of proposed approach

In this article, we have taken the toy_data_df which is simulated on TMixClust function. The data is taken at different 5 time slots with all gene values as follows:



```
> print(head(toy_data_df))
            time1     time2    time3    time4     time5     time6
gene_1  14.5515322  7.156850 41.43372 64.17897  80.89528 127.10444
gene_2   7.0624795 15.192533 33.80188 55.40540  86.15328 110.73804
gene_3 -18.4186975  4.806256 13.79942 69.30776  61.74546  98.10358
gene_4  13.8395249 14.386315 42.02976 42.93031 114.87758  95.95243
gene_5   1.5623627 29.595991 33.93879 35.56273  68.20763  96.32600
gene_6  -0.8357523 34.048379 38.37403 46.25532  90.50204 128.95565
>
```

Fig. 2 Head of toy_data_df

The function TMixClust contains a simulated data set, toy_data_df, with 91 time series. We have Loaded and plotted the time series data of toy_data_df

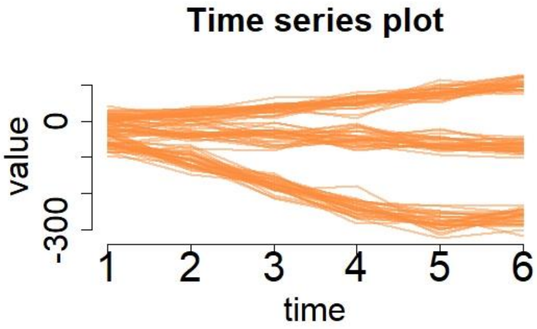


Fig. 3 Time series plot of toy_data_df

3.2 Clustering

We have divided the whole set into three parts and the clustering object is created. Depending on the input data, the clustering result may be different than the optimal solution. This behavior can be observed if the clustering operation is repeated several times [2]

For clustering we need just two arguments, dataframe and number of clusters. Depending on the data set the resultant clustering may be differed from the optimum clustering.

3.3 Optimal clustering selection

Clustering is done through the function TMixClust with several parameters. TMixClust is based on a statistical model where inference is made through the Expectation Maximization (EM) technique. When running the clustering algorithm, the EM procedure might get stuck in a local optimum. The local optimum solution has a lower likelihood and is suboptimal. It is therefore highly recommended to perform a stability analysis of the clustering in order to see how often the algorithm gets stuck in local optima and how different are these local optima from the best clustering solution.

Finally, we highly recommend to run several times the TMixClust function, in order to ensure that the global optimum solution is reached. The function also computes and plots the distribution of the Rand index [6] between each clustering solution and the global optimum solution. The Rand index quantifies the agreement probability between two clustering runs, also showing clustering stability.

The user can define the number of clustering runs (i.e, the number of times TMixClust algorithm is run on the same data, initial conditions and clustering configuration) and has the possibility to parallelize the runs by defining a number of computing cores. By default, the function uses 2 cores. 5 Clustering time series gene expression data with TMixClust [7] For example, we can repeat clustering on the previously presented simulated data for 10 times, for a number K=3 of clusters and using 3 cores, then plot the best clustering solution. For execution time reasons, we have stored the result of this analysis (commented analyse_stability command in the code below) in a pre-computed object available with the package TMixClust.

The function analyse_stability also produces a histogram of the Rand indexes corresponding to each clustering solution. For our example and straightforward simulated data, we have performed only 10 clustering runs. Depending on the size and complexity of the data, 10 runs might not be enough for attaining the global optimum so larger number of runs may improve the results.

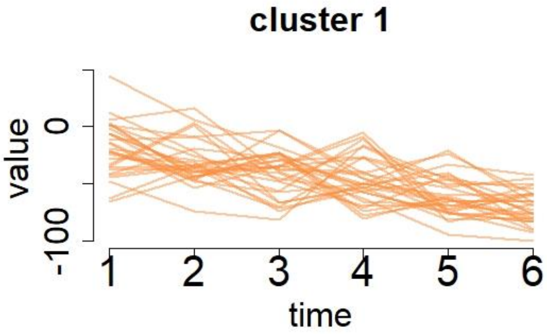


Fig. 4 Best Cluster 1

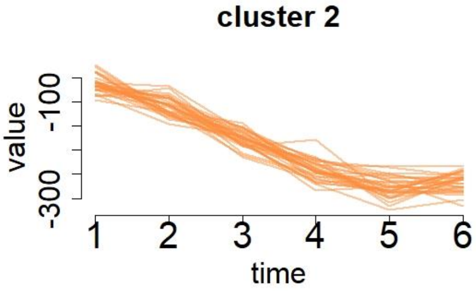


Fig. 5 Best Cluster 2

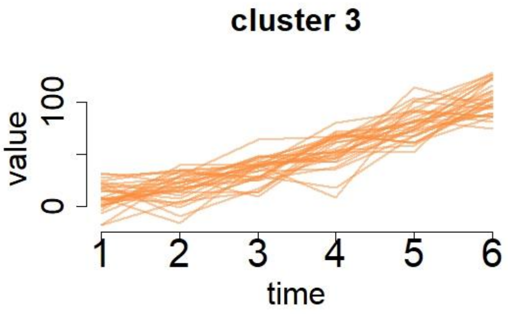


Fig. 6 Best Cluster 3

3.4 Clustering consistency with silhouette Analysis

Silhouette refers to a method of interpretation and validation of consistency within clusters of data. The technique provides a succinct graphical representation of how well each object has been classified.[13]
The silhouette value is a measure of how similar an object is to its own cluster (cohesion) compared to other clusters (separation). The silhouette ranges from −1 to +1, where a high value indicates that the object is well matched to its own cluster and poorly matched to neighboring clusters. If most objects have a high value, then the clustering configuration is appropriate. If many points have a low or negative value, then the clustering configuration may have too many or too few clusters.
The silhouette can be calculated with any distance metric, such as the Euclidean distance or the Manhattan distance. To assist the user in performing a qualitative analysis of different clustering configurations and choosing an adequate number of clusters, package TMixClust provides a tool based on the silhouette technique [8].
Here we have taken the silhouette width, which is a measure of how similar a data point is to the other points from the same cluster as opposed to the rest of the clusters. Therefore, a high average silhouette width indicates a good clustering cohesion. The most straightforward way to investigate silhouette widths for the data points in a clustering is through visualization of a silhouette plot.
This plot displays the distribution of silhouette coefficients calculated for each data point (in our case each time series) from every cluster. The user can generate a silhouette plot using the plot_silhouette function. For our simulated data, we can generate a silhouette plot for the previously obtained global optimum clustering solution for K=3

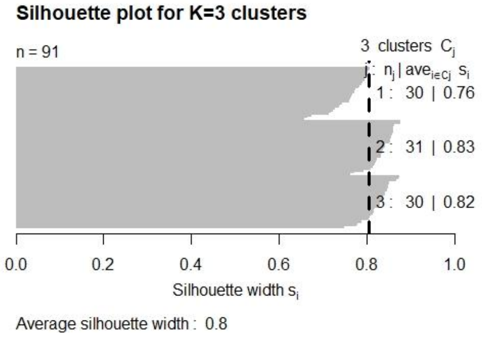


Fig. 7 Sihouette plot for 3 clusters

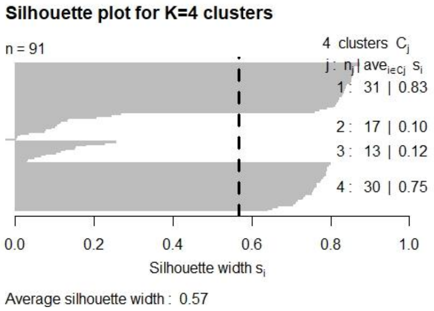


Fig. 8 Sihouette plot for 4 clusters

4. Results and Discussion

Generally, the larger the silhouette widths and the more data points with silhouette width above average, the better a clustering is defined. By comparing the silhouette plots, if we look at the average silhouette width (black dotted line) for K=4, we can clearly see how both the silhouette width and the proportion of data points above average width are less than for K=3, meaning that the clustering with K=4 is starting to over fit the data. The solution with K=3 is better. The user can in this way use the silhouette plot to choose the best number of clusters corresponding to the data.

5. Applications of TMixClust

We apply TMixClust to a real gene expression time series data set which records transcriptional changes during budding yeast cell-division cycle [9]. For our example, we use a subset of 125 time series measured at five different time points included in the package file yeast_time_series.txt. After running TMixClust with different numbers of clusters, investigating the silhouette plots and stability as presented in the previous section, we concluded that the main patterns of gene expression were best described by K=4 clusters. We have stored the TMixClust object containing the optimal clustering solution in the best_clust_yeast_obj object, available with the package. We can load the data, plot its time series, load the optimal clustering solution and plot the 4 identified clusters as following.

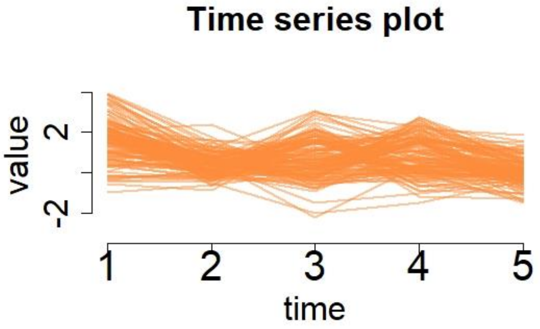


Fig. 9 Time series plot for yeast

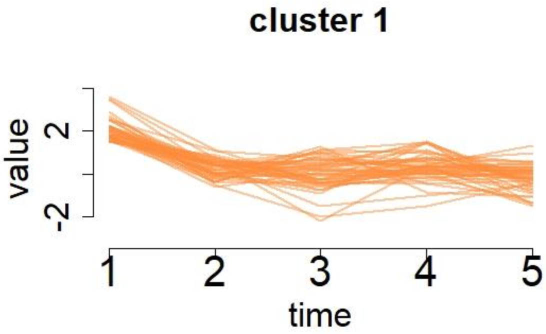


Fig. 10 Cluster 1 for Yeast

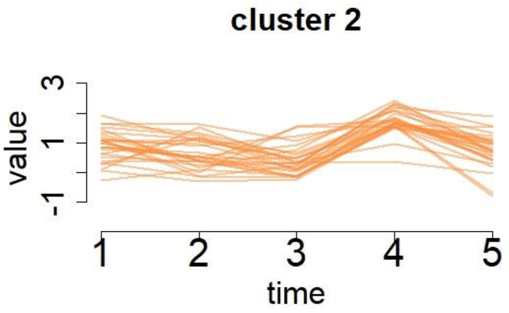


Fig. 11 Cluster 2 for yeast

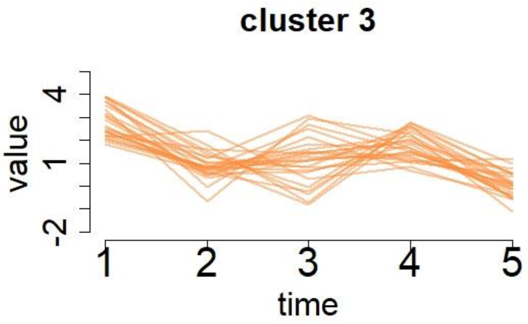


Fig. 12 Cluster 3 for yeast

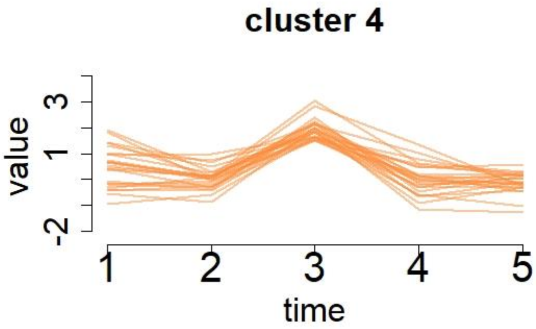


Fig. 13 Cluster 4 for yeast

6. Methodology of TMixClust

TMixClust uses the concepts described by [10] for clustering gene expression time series data within the Gaussian mixed-effects models framework with nonparametric smoothing spline estimation [11]. In the following, we provide a short description of these concepts.

6.1 Mixed effects model with embedded smoothing splines

Let X = {Xi}, 1 ≤ i ≤ N be a set of N gene expression observations, where each observation Xi is a gene expression time series with T time-points: Xi = {xi,1, ..., xi,T }.

The task is then to cluster the N observations into K groups based on their time series behavior.

6.2 Estimating model parameters

Within TMixClust, we use package gss in the following implemented EM learning scheme:

1. initialize clusters (e.g. with a K-means solution for speeding up convergence)
2. calculate data likelihood and repeat until convergence:
3. when convergence is reached, return maximum likelihood solution

E-step:	***M-step***
compute posterior probabilities	*maximize penalised likelihood score with package gss*
assign genes to clusters based on their posterior probabilities - compute mixing coefficients	*update model parameters*

Fig. 14 Table1: E-step and M-step

7. Conclusion

In this article, we have used a soft clustering method which employs mixes effects models with nonparametric smoothing spline fitting and is able to robustly stratify genes by their complex time series patterns. The results are besides the main clustering method, a set of functionalities assisting the user to visualize and assisting the user to visualize and assess the clustering results and to choose the most optimal clustering solution.

References

[1] H.Wang and J. E. Glover, "Noise analysis of time series data in gene regulatory networks," 2011 4th International Conference on Biomedical Engineering and Informatics (BMEI), Shanghai, 2011, pp. 1848-1852.

[2] Patil S.K., Mallick S., Chakraborty A., Das A. (2019) Informative Gene Selection Using Clustering and Gene Ontology. In: Abraham A., Dutta P., Mandal J., Bhattacharya A., Dutta S. (eds) Emerging Technologies in Data Mining and Information Security. Advances in Intelligent Systems and Computing, vol 813. Springer, Singapore

[3] Dunson. Nonparametric Bayes applications to biostatistics. In L. Hjort, C. Holmes, P. Muller, and S. Walker, editors, Bayesian Nonparametrics. Cambridge Univ Pr, 2010

[4] Behseta, R. E. Kass, and G. L. Wallstrom. Hierarchical models for assessing variability among functions. Biometrika, 92(2):419–434, [2005]

[5] Rasmussen and Z. Ghahramani. Infinite mixtures of gaussian process experts. Advances in neural information processing systems, 2:881–888, 2002.

[6] Lawrence Hubert and Phipps Arabie. Comparing partitions. Journal of Classification, 2(1):193–218, 1985. URL: http://dx.doi.org/10.1007/BF01908075, doi:10.1007/BF01908075.

[7] Alamuri, Madhavi, Bapi Raju Surampudi, and Atul Negi. "A survey of distance/similarity measures for categorical data." 2014 International joint conference on neural networks (IJCNN). IEEE, 2014.

[8] Peter J. Rousseeuw. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. Journal of Computational and Applied Mathematics, 20:53 – 65, 1987. URL: http://www.sciencedirect.com/science/article/pii/0377042787901257, doi:http://dx.doi.org/10.1016/0377-0427(87)90125-7

[9] Daniel F. Simola, Chantal Francis, Paul D. Sniegowski, and Junhyong Kim. Heterochronic evolution reveals modular timing changes in budding yeast transcriptomes. Genome Biology, 11(10):R105, 2010. URL: http://dx.doi.org/10.1186/gb-2010-11-10-r105, doi:10.1186/gb-2010-11-10-r105.

[10] Ping Ma, Cristian I. Castillo-Davis, Wenxuan Zhong, and Jun S. Liu. A data-driven clustering method for time course gene expression data. Nucleic Acids Res, 34(4):1261–1269, Mar 2006. 16510852[pmid]. URL: http://www.ncbi.nlm.nih.gov/pmc/articles/PMC1388097/, doi:10.1093/nar/gkl013.

[11] Chong Gu. Smoothing spline anova models: R package gss. Journal of Statistical Software, 58(1):1–25, 2014. URL: https://www.jstatsoft.org/index.php/jss/article/view/v058i05, doi:10.18637/jss.v058.i05.

[12] Monica Golumbeanu, Sebastien Desfarges, Celine Hernandez, Manfredo Quadroni, Sylvie Rato, Pejman Mohammadi, Amalio Telenti, Niko Beerenwinkel, and Angela Ciuffi. Dynamics of proteo-transcriptomic response to hiv-1 infection. in preparation, 2017.

[13] Peter J. Rousseeuw (1987). "Silhouettes: a Graphical Aid to the Interpretation and Validation of Cluster Analysis". Computational and Applied Mathematics. 20: 53–65. doi:10.1016/0377-0427(87)90125-7.

[14] www.bioconductor.org

[15] www.ncbi.nlm.nih.gov

First Author Prof Tejal Upadhyay has obtained her graduation degree from L D College of Engineering, Ahmedabad and Post graduate from Dharamsinh Desai University, Nadiad. She has published good publications in international journals and conferences. She has a more than 23 years of teaching experience to undergraduate and postgraduate students at an Engineering institutions. She is a

Life time membership of Indian Society for Technical Education and Computer Society of India. She is a student branch counselor (SBC) of CSI student branch, Nirma University since 2005.

Second Author "Dr. Samir B. Patel obtained his Ph.D. Degree from Nirma University in Computer Engineering in the month of October 2012. He has published more than 28 papers of National and International Repute. He is the author of one book. Before joining PDPU, he had worked as Principal, GMFE, Sr. Associate Professor at Nirma University, Assistant Prof. and Senior Lecturer at AESICS and before that as a lecturer and programmer cum lecturer at CPICA and PDPICA. He has a total 21 years of teaching and administrative experience. He is the reviewer of various journals of repute and is a life member of Computer Society of India and ISTE. "