14

Clustering Analysis of SAGE Transcription Profiles Using a Poisson Approach

Haiyan Huang, Li Cai, and Wing H. Wong

Summary

To gain insights into the biological function and relevance of genes using serial analysis of gene expression (SAGE) transcription profiles, one essential method is to perform clustering analysis on a group genes with similar expression patterns. A successful clustering analysis depends on the use of effective distance or similarity measures. For this purpose, by considering the specific properties of SAGE technology, we modeled the SAGE data by Poisson statistics and developed two Poisson-based measures to assess similarity of gene expression profiles. By employing these two distances into a K-means clustering procedure, we further developed a software package to perform clustering analysis on SAGE data. The software implementing our Poisson-based algorithms can be downloaded from http://genome.dfci.harvard.edu/sager. Our algorithm is guaranteed to converge to a local maximum when Poisson likelihood-based measure is used. The results from simulation and experimental mouse retina data demonstrate that the Poisson-based distances are more appropriate and reliable for analyzing SAGE data compared to other commonly used distances or similarity measures.

Key Words: Clustering analysis; (Dis)similarity measures; Poisson statistics; K-means clustering procedure; SAGE data.

1. Introduction

Serial analysis of gene expression (SAGE), an effective technique for comprehensive gene expression profiling, has been employed in studies of a wide range of biological systems ***(1–5)***. Previous efforts to develop SAGE analysis methods have been focused primarily on extracting SAGE tags and

From: *Methods in Molecular Biology, vol. 387: Serial Analysis of Gene Expression: Digital Gene Expression Profiling*
Edited by: K. L. Nielsen © Humana Press Inc., Totowa, NJ

identifying differences in mRNA levels between two libraries ***(2,3,6–11)***. To gain additional insights into the biological function and relevance of genes from expression data, an established strategy is to perform clustering analysis, which is to search for patterns and group transcripts with similar expression profiles. This strategy has led to the fundamental question of how to measure the (dis)similarity of gene expression across multiple SAGE libraries. An effective distance or similarity measure ***(12)***, which takes into account the underlying biology and the nature of data, would be the basis for a successful clustering analysis. Commonly used distances or similarity measures include the *Pearson correlation coefficient* and *Euclidean distance*. Pearson correlation is used to detect the shape coherence of expression curves; Euclidian distance can be used when the data are normally distributed and the magnitude of expression matters. Other measures of relationships include *likelihood-based* approaches for measuring the probabilities of clusters of genes in Gaussian mixture modeling ***(13–15)***, etc. These measures have been proven useful in microarray expression data analysis. However, SAGE data are governed by different statistics; they are generated by sampling, which results in "counts." In this regard, clustering analysis of SAGE data should involve appropriate statistical methods that consider the specific properties of SAGE data.

In one of our previous studies ***(16)***, we assumed that the tag counts follow a Poisson distribution. This is a natural assumption considering that SAGE data are generated through a random sampling technique. Based on this assumption, two Poisson-based measures were developed to assess the similarity of tag count profiles across multiple SAGE libraries ***(16)***. One measure was defined based on Chi-square statistic, which evaluates the deviation of observed tag counts from expected counts in each cluster. This method was called *PoissonC*. The other measure was based on the log-likelihood of observed tag counts, which determines the cluster membership of a transcript by its observed counts' joint probability under the expected Poisson model in each cluster. This method was called *PoissonL*. A packaged clustering program with a modified K-means procedure and with the two measures implemented is available at http://genome.dfci.harvard.edu/sager.

In this chapter, we will introduce this Poisson-based SAGE clustering method and evaluate its performance by applying it to a simulation dataset and an experimental mouse retinal SAGE dataset. These additional applications to those described in Cai et al. ***(16)*** further demonstrate the advantages of the Poisson-based measures over Pearson correlation and Euclidean distance in terms of producing clusters of more biological relevance. We also verify that the Poisson

likelihood-based clustering algorithm *PoissonL* is guaranteed to converge to a local maximum of the Poisson likelihood function for observed data.

2. Materials

1. *Software:* online web application website as well as a Linux and Microsoft Windows software are available at http://genome.dfci.harvard.edu/sager.
2. *License agreement:* the program is copyrighted by Li Cai, Haiyan Huang, and other contributors, and is free for nonprofit academic use. It can be redistributed under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or any later version. This program is distributed in the hope that it will be useful for research purpose, but **without any warranty**.
3. *Data:* the file format is a classical expression matrix, with each row representing the counts for a single tag over multiple SAGE libraries and with each column representing the counts for all tags in a single library. The packaged program prefers tab-delimited format. The specific extensions supported by the packaged program are txt, xls, wk1, wk3, wk4, mdb, fp5, 123, and dat.
4. *Minimum computer hardware requirements:* the computer used to run this program should meet at least the following requirements: (1) at least 256 MB of RAM; (2) an at least 1-GHz CPU; (3) a hard drive with at least 500 MB of free disk space; (4) Microsoft Windows 9x/NT/ME/2000/XP or any Linux operating system.

3. Methods

In the following sections, we rationalize the Poisson assumption on SAGE data and provide a detailed description on the Poisson probability model, by which two Poisson-based similarity measures were defined (*see* **Note 1**). We also verify that the introduced clustering algorithm with the likelihood-based similarity measure is guaranteed to converge to a local maximum of the Poisson likelihood function. Finally, we present the application of the Poisson-based method to a simulation dataset and a real dataset.

3.1. Poisson Assumption

In an SAGE experiment, the tag extraction is performed on a set of transcripts that are sampled from a cell or tissue. As discussed in Man et al. ***(10)***, this sampling process is approximately equivalent to randomly taking a bag of colored balls from a big box. This randomness leads to an approximate multinomial distribution for the number of transcripts of different types for tag extraction ***(17)***. Moreover, as a result of the vast amount and numerous varied types of transcripts in a cell or tissue, the selection probability of a particular

type of transcript at each draw should be very small, which suggests that the tag counts of sampled transcripts of each type can be approximately Poisson distributed.

3.2. Probability Model

The above arguments suggest a Poisson-based probability model, which can be specified by the following two assumptions.

3.2.1. Assumption 1

$Y_i(t)$, the count of tag i in library t, are independent Poisson variables with parameters $\lambda_i(t)\theta_i$, where θ_i is the expected sum of counts of tag i over all libraries (unrelated to t), $\lambda_i(t)$ is the contribution of tag i in library t to the sum (θ_i) expressed in percentage, and the sum of $_i(t)$ over all libraries equals to 1.

Assumption 1 forms the basis of the probability model. By definition, θ_i reflects the gene general expression level, $\lambda_i(t)$ describes the expression changes across libraries, and $\lambda_i(t)\theta_i$ re-distributes the tag counts according to the expression profile $[\lambda_i(t)]$ with the sum of counts across libraries kept constant. The tags with similar $\lambda_i(t)$ over t (libraries) will be grouped together, because an established strategy for finding functionally related genes is to group genes with similar expression patterns ***(18)***. This motivates Assumption 2.

3.2.2. Assumption 2

The tags in the same cluster share a common profile of $\lambda_i(t)$ over t. The common profile is denoted by $\lambda = [\lambda(1), \lambda(2), \ldots, \lambda(T)]$, where T is the total number of libraries considered. λ then represents the cluster profile.

Now, let $\boldsymbol{Y}_i = [Y_i(1), \ldots, Y_i(T)]$ denote the vector of counts of tag i across T libraries. Then, under the above two assumptions, for a cluster consisting of tags 1, 2, …, m, the joint likelihood function for $\boldsymbol{Y}_1$, $\boldsymbol{Y}_2, \ldots, \boldsymbol{Y}_{\mathrm{m}}$ is

$$L(\lambda, \theta \mid \boldsymbol{Y}) \propto f(\boldsymbol{Y_1}, \ldots, \boldsymbol{Y_m} \mid \lambda, \theta_1, \ldots, \theta_m) = \prod_{i=1}^{m} \prod_{t=1}^{T} \frac{\exp(-\lambda(t)\theta_i)(\lambda(t)\theta_i)^{Y_i(t)}}{Y_i(t)!}. \quad (1)$$

The maximum likelihood estimates (MLEs) of λ and $\theta_1, \ldots, \theta_m$ are

$$\hat{\theta}_i = \sum_t Y_i(t), \text{ and } \hat{\lambda}(t) = \sum_{i=1}^{m} Y_i(t) \Big/ \sum_{i=1}^{m} \hat{\theta}_1 = \sum_{i=1}^{m} Y_i(t) \Big/ \sum_{i=1}^{m} \sum_t Y_i(t). \quad (2)$$

3.3. Two New Similarity Measures for Clustering Tags With Similar Expression Profiles

Given a cluster consisting of tags 1, . . ., *m*, the MLEs of parameters λ and θ_i in **eq. 2** provide the expected Poisson distributions for the tag counts in each cluster. This forms the basis of the definitions of the following two new measures, which evaluate how well a particular tag (gene) fits in each of the clusters.

3.3.1. Likelihood-Based Measure

It is natural to use the log-likelihood function log to evaluate how well the observed counts $(\boldsymbol{Y}_i|\lambda,\theta_i)$ fit the expected Poisson distributions. The larger the log-likelihood is, the more likely the observed counts are to be generated from the expected model. For a cluster consisting of tags 1, 2, . . ., *m*, the dispersion is defined as

$$L = -\log f(Y_1, \ldots, Y_m \mid \hat{\lambda}, \hat{\theta}) = \sum_{i=1}^{m} \sum_{t=1}^{T} (\hat{\lambda}(t)\hat{\theta}_i - Y_i(t)\log(\hat{\lambda}(t)\hat{\theta}_i) + \log(Y_i(t)!)). \quad (3)$$

The optimal partition of the genes into *k* distinct clusters can be obtained by minimizing the cluster dispersion $L_1 + L_2 + \ldots + L_k$.

3.3.2. Chi-Square Statistic-Based Measure

The Chi-square statistic can evaluate the deviation of observed counts from expected counts in each cluster. For a cluster consisting of tags 1, 2, . . ., *m*, the dispersion can be defined as

$$D = \sum_{i=1}^{m} \sum_{t=1}^{T} (Y_i(t) - \hat{\lambda}(t)\hat{\theta}_i)^2 \Big/ (\hat{\lambda}(t)\hat{\theta}_i). \quad (4)$$

The smaller *D* is, the tighter the cluster is. The optimal partition of the genes into *k* distinct clusters can be obtained by minimizing the cluster dispersion $D_1 + D_2 + \ldots + D_k$. Using the Chi-square statistic as a similarity measure, the penalty for deviation from large expected count is smaller than that for small expected count. This is consistent with the above likelihood-based measure because the variance of a Poisson variable equals its mean.

3.4. Clustering Procedure

Using the above two measures, Cai et al. ***(16)*** modified the K-means clustering algorithm to group tags with similar count profiles. The K-means clustering procedure ***(19)*** generates clusters by specifying a desired number of clusters, say, *K*, and then assigns each object to one of *K* clusters so as to

minimize a measure of dispersion within the clusters (*see* **Note 2**). We outline the algorithm from Cai et al. ***(16)*** as follows:

1. All SAGE tags are assigned at random to K sets. Estimate initial parameters $\theta_i^{(0)}$ and $\lambda_k^{(0)} = (\lambda_k^{(0)}(1), \ldots, \lambda_k^0(T))$ for each tag and each cluster by **eq. 2**.
2. In the $(b+1)$th iteration, assign each tag i to the cluster with minimum deviation from the expected model. The deviation is measured by either $L_{i,k}^{(b)} = -\log f(Y_i \mid \lambda_k^{(b)}, \theta_i^{(b)})$ or $D_{i,k}^{(b)} = \sum_t \left(Y_i(t) - \lambda_k^{(b)}(t)\theta_i^{(b)}\right)^2 \Big/ (\lambda_k^{(b)}(t)\theta_i^{(b)})$.
3. Set new cluster centers $\lambda_k^{(b+1)}$ by **eq. 2**.
4. Repeat **step 2** until convergence.

Let $c(i)$ denote the index of the cluster that tag i is assigned to. The above algorithm aims to minimizes the within-cluster dispersion $\sum_i L_{i,c(i)}$ or $\sum_i D_{i,c(i)}$. The algorithm using the likelihood-based measure L was called *PoissonL*, and the algorithm using the Chi-square based measure D was called *PoissonC*. We want to point out that *PoissonL* is guaranteed to converge to a local maximum of the joint likelihood function for the observed data under the assumed probability model. We present the proof below.

3.4.1. Lemma 3.4.1.

Each iteration in the *PoissonL* algorithm is guaranteed to increase the likelihood for the observed data under the assumed probability model, and thus the algorithm is guaranteed to converge to a local maximum of the likelihood function.

3.4.2. Proof of Lemma 3.4.1.

Under the Poisson model described under **Subheading 3.2.**, the tag count profiles $Y_1, Y_2, \ldots, Y_N$ are assumed to be independently generated from K different joint Poisson distributions, whereas the information on which and what model generates each tag count profile is unknown. Let y_i be the cluster label for tag i, and $\Theta = (\lambda_1, \ldots, \lambda_K, \theta_1, \ldots, \theta_N)$ be the model parameters with λ_K and θ_i defined as under **Subheading 3.2.** Then, the objective is to find the Θ and y_i that maximize

$$L(\Theta \mid X) = \prod_{i=1}^{N} f(\boldsymbol{Y}_i|\Theta) = \prod_{i=1}^{N} \prod_{k=1}^{K} f(\boldsymbol{Y}_i|\boldsymbol{\lambda}_k, \theta_i)^{I(y_i=k)}, \quad (5)$$

where $I(y_i = k)$ equals 1 when $y_i = k$ and 0 otherwise.

In the $(b+1)$th iteration of *PoissonL*, for $i = 1, .., N$ and $k = 1, .., K$, we estimate

$$y_i^{(b+1)} = \arg\min_k L_{i,k} = \arg\max_k f(Y_i|\lambda_k^{(b)}, \theta_i^{(b)}) \quad \text{(by step 2 of the algorithm), and} \quad (6)$$

$$\mathbf{\Theta}^{(b+1)} = \arg\max_{\Theta} \prod_{i=1}^{N} \prod_{k=1}^{K} f(\boldsymbol{Y}_i|\boldsymbol{\lambda}_k, \theta_i)^{\mathrm{I}(y_i^{(b+1)}=k)} \quad \text{(by step 3 of the algorithm).} \quad (7)$$

$$\begin{aligned} \text{Then,} \quad & \mathsf{L}(\mathbf{\Theta}^{(b+1)} \mid \mathsf{X}) = \prod_{i=1}^{N} \prod_{k=1}^{K} f(\boldsymbol{Y_i} \mid \boldsymbol{\lambda}_k^{(b+1)}, \theta_i^{(b+1)})^{\mathrm{I}(y_i^{(b+1)}=k)} \\ \text{(by (7))} \quad & \geq \prod_{i=1}^{N} \prod_{k=1}^{K} f(\boldsymbol{Y_i} \mid \boldsymbol{\lambda}_k^{(b)}, \theta_i^{(b)})^{\mathrm{I}(y_i^{(b+1)}=k)} \\ \text{(by (6))} \quad & \geq \prod_{i=1}^{N} \prod_{k=1}^{K} f(\boldsymbol{Y_i} \mid \boldsymbol{\lambda}_k^{(b)}, \theta_i^{(b)})^{\mathrm{I}(y_i^{(b)}=k)} = \mathsf{L}(\mathbf{\Theta}^{(b)} \mid \mathsf{X}), \end{aligned} \quad (8)$$

which means that each iteration in *PoissonL* is guaranteed to increase the likelihood for the observed data, and thus the algorithm is guaranteed to converge to a local maximum.

PoissonC and *PoissonL* differs at the step of updating $y_i^{(b+1)}$. In *PoissonC*, $y_i^{(b+1)} = \arg\min_k D_{i,k}^{(b)}$, under which $f(Y_i \mid \lambda_k^{(b)}, \theta_i^{(b)})^{1(y_i^{(b+1)}=k)} \geq f(\boldsymbol{Y}_i \mid \lambda_k^{(b)}, \theta_i^{(b)})^{1(y_i^{(b)}=k)}$ and therefore $\mathsf{L}(\Theta^{(b+1)} \mid \mathsf{X}) \geq \mathsf{L}(\Theta^{(b)} \mid \mathsf{X})$ may not hold because $D_{i,k}$ is not always monotone relative to the likelihood function. The nonmonotone domain is, however, vastly small. In practice, the nonmonotone domain is often sufficiently small and negligible for the considered dataset such that *PoissonC* agrees with *PoissonL* and converges to a local maximum. One big advantage of *PoissonC* compared to *PoissonL* is that it runs much faster based on the current version of program (*see* **Note 3**).

PoissonL is actually a specific version of Classification EM algorithm (CEM) ***(20)***. The objective likelihood function of CEM under the mixture Poisson model is

$$\mathbf{L}_{\mathrm{CEM}}(\mathbf{\Theta} \mid \mathsf{X}) = \prod_{i=1}^{N} \prod_{k=1}^{K} \left(f(Y_i \mid \Theta)^{\mathrm{I}(y_i=k)} f(\mathbf{y}_i = k \mid \mathbf{\Theta}) \right), \quad (9)$$

which is equivalent to **eq. 5** when the prior conditional probability of y_i given Θ is uniform (*see* **Note 4**).

3.5. Implementation

PoissonL and *PoissonC* are implemented in both C++ and Java. The implementation in C++ is based on the open source code of the C clustering Library provided by de Hoon et al. ***(21)*** (http://bonsai.ims.u-tokyo.ac.jp/~mdehoon/software/cluster/software.htm) (*see* **Note 5**). We developed a web-based application as well as Microsoft Windows and Linux versions of software to perform the clustering analysis. The software is available at http://genome.dfci.harvard.edu/sager.

3.6. Examples

Two examples are presented here to demonstrate the advantages of Poisson-based measures over other commonly used distance or similarity measures in analyzing SAGE or Poisson-like data. Because these examples are independent of the ones shown in Cai et al. ***(16)***, they can serve as an additional validation of the Poisson-based measures.

3.6.1. Example I: Clustering Results of Simulation Data

Data. The distributions used to generate the simulation dataset are described in **Table 1**. The simulation dataset consists of 46 vectors of dimension 5 with components independently generated from different Normal distributions. The mean (μ) and variance (σ^2) parameters of the normal distributions are constrained by $\sigma^2 = 3\,\mu$. This application evaluates the performance of our method on data with Poisson-like properties: variance increases with mean. Success in this dataset would shed light on more broad applications of our method.

In our simulation dataset, the 46 vectors belong to six groups (named A, B, C, D, E, and F) according to the Normal distributions from which they are generated. The six groups are of size 3, 6, 6, 9, 7, and 15, respectively. For comparison, we applied *PoissonC* together with *Eucli* (classical K-means clustering algorithm

Table 1
5-Dim Simulation Dataset With Normal Distributions $\sigma^2 = 3\mu$

Group ID		Mean parameters of the normal distributions (μ)				
Group A	a1 ~ a3	1	1	1	15	150
Group B	b1 ~ b6	15	1	1	1	150
Group C	c1 ~ c4	10	30	30	60	10
	c5 ~ c6	100	300	300	600	100
Group D	d1 ~ d7	200	70	70	10	10
	d8 ~ d9	2000	700	700	100	100
Group E	e1 ~ e5	210	120	10	10	10
	e6 ~ e7	2100	1200	100	100	100
Group F	f1 ~ f3	5	50	5	5	5
	f4 ~ f6	5	75	5	5	5
	f7 ~ f9	5	100	5	5	5
	f10 ~ f11	50	500	50	50	50
	f12 ~ f13	50	750	50	50	50
	f14 ~ f15	50	1000	50	50	50

using Euclidian distance) and *PearsonC* (K-means clustering procedure using Pearson correlation as similarity measure) to the simulated data. The clustering results from different methods are shown in **Fig. 1**. The simulation data is available at http://www.stat.berkeley.edu/users/hhuang/SAGE.html.

Results. In **Fig. 1**, only *PoissonC* has clustered the vectors perfectly into six groups. All of the other methods fail to correctly separate the vectors from Group A and Group B. *Eucli* works the worst when it is applied to unnormalized data. It fails to identify any of the six clusters. This is because Euclidian distance can be overly sensitive to the magnitude of changes. To reduce the magnitude effects, we further apply *Eucli* to the rescaled data. The rescaling is performed so that the sum of the components within each vector is set the same. The clustering result of *Eucli* on rescaled data is clearly better than the

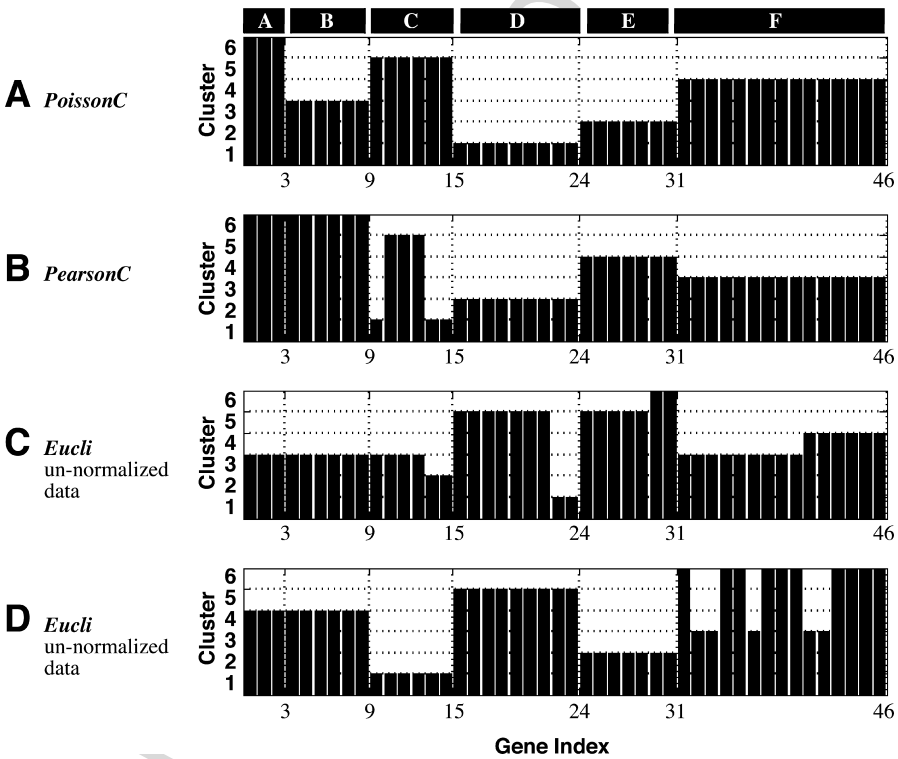

Fig. 1. Graphs of clustering results for simulation data. Horizontal axis represents the index of the 46 vectors, which belong to six groups (named A, B, C, D, E, and F) that are marked at the top of the figure. Vertical axis represents the index of the cluster that each vector has been assigned to by each algorithm.

result on unnormalized data. Groups C, D, and E have been correctly identified (*see* **Note 6**).

We perform an additional 100 replications of the above simulation. *PoissonC* correctly clusters 34 of the 100 replicate datasets. *Eucli*, on rescaled data, correctly clusters 2 of the 100 datasets whereas *PearsonC* or *Eucli*, on unnormalized data, never generates correct clusters.

We also want to point out that there is a small error in the simulation results presented in Table 1 and Fig. 1 of Cai et al. ***(16)***. For the data in Table 1, Fig. 1 reported a perfect clustering result by *PoissonC*, which is not correct. But the conclusion made from that example that *PoissonC* is superior to other methods is still valid because *PoissonC* has only wrongly clustered one tag.

3.6.2. Example II: Clustering Results of Experimental SAGE Data

For further validation, we apply *PoissonC, PearsonC*, and *Eucli* to a set of mouse retinal SAGE libraries.

Data. The raw mouse retinal data consists of 10 SAGE libraries (38,818 unique tags with tag counts $\geq$2) from developing retina taken at 2-d intervals, ranging from embryonic to postnatal and adult ***(16,22)***. One thousand four hundred sixty-seven of the 38,818 tags with counts $\geq$20 in at least one of the 10 libraries are selected (*see* **Note 7**). To effectively compare the clustering algorithms, a subset of 153 SAGE tags with known biological functions are further selected (*see* **Note 8**). These 153 tags fall into five clusters based on their biological function(s) (*see* **Table 2a**). One hundred twenty-five of these genes are developmental genes, which can be further grouped into four clusters by their expressions at different developmental stages. The other 28 genes are unrelated to the mouse retina development. This dataset is available at http://www.stat.berkeley.edu/users/hhuang/SAGE.html.

Results. PoissonC, PearsonC, and *Eucli* are applied to group these 153 tags into five clusters. Results show that the performance of *PoissonC* is superior to other methods (*see* **Table 2b**). We should also note that *PoissonC* is only

Table 2a
Functional Categorization of the 153 Mouse Retinal Tags (125 Developmental Genes; 28 Nondevelopmental Genes)

	Function Groups					
	Early I	Early II	Late I	Late II	Non-dev.	Total
Number of tags	32	34	32	27	28	153

Table 2b
Comparison of Algorithms on 153 Tags

Algorithm	# of tags in incorrect clusters	% of tags in incorrect clusters
PoissonC	22	14.4
Eucli on normalized data	36	23.5
PearsonC	26	17.0
Eucli	NA	NA

Clusters generated by *Eucli* were too messy.

slightly better than *PearsonC* in this application because the shapes of the gene expression curves are quite different from each other among these five clusters and the Pearson correlation can powerfully detect the shape coherence of curves.

Acknowledgments

The method described in this chapter is based on the original research paper published in Genome Biology ***(16)***. We thank Kyungpil Kim for help in generating the figure and tables.

Notes

1. The main advantage of the described method is that the newly designed measures consider both the magnitude and shape when comparing the expression patterns (λ represents the shape and θ represents the magnitude in our model), whereas Euclidian distance is focused only on the magnitude of changes and Pearson correlation is overly sensitive to the shape of the curve.
2. An unsolved issue in K-means clustering analysis is the estimation of K, the number of clusters. If K is unknown, starting with arbitrary, random K is a relatively poor method. Hartigan proposed a stage-wise method to determine the K value ***(19)***. However, when sporadic points are present in the dataset, Hartigan's method may fail. A recently introduced method, TightCluster ***(23)***, partially solves this problem by using a resampling scheme to sequentially attain tight and stable clusters in the order of decreasing stability. The Poisson based measures can be implemented in the TightCluster program to apply the TightCluster method to SAGE data.
3. We performed *PoissonL* and *PoissonC* similarly when applying them to many small simulation and experimental data sets. For large datasets, *PoissonC* should be more practical at this moment, as the current version of *PoissonL* (installed in the software package) is too slow. There is still much room for improving the *PoissonL* algorithm.

4. We can also derive an EM algorithm for fitting the mixture Poisson model. The associated objective likelihood is

$$\mathbf{L}_{\mathrm{EM}}(\Theta|\mathbf{X}) = \prod_{i=1}^{N} f(\boldsymbol{Y}_i|\boldsymbol{\Theta}) = \prod_{i=1}^{N} \left(\sum_{k=1}^{K} f(\boldsymbol{Y}_i|\boldsymbol{\Theta}) f(y_i = k|\boldsymbol{\Theta}) \right). \tag{10}$$

The E-step and M-step of the algorithm can be described as follows:
E-step: with the estimated $\boldsymbol{\Theta}^{(b)} = (\boldsymbol{\lambda}_1^{(b)}, \ldots, \boldsymbol{\lambda}_K^{(b)}, \theta_1^{(b)}, \ldots, \theta_N^{(b)})$, compute

$$\begin{aligned} &Q(\boldsymbol{\Theta}, \boldsymbol{\Theta}^{(b)}) \\ &\quad = E_{y_1,\ldots,y_N|\boldsymbol{Y}_1,\ldots,\boldsymbol{Y}_N,\boldsymbol{\Theta}^{(b)}} [\log f(\boldsymbol{Y}_1, \ldots, \boldsymbol{Y}_N, y_1, \ldots, y_N|\boldsymbol{\Theta})] \\ &\quad = \sum_{y_1,\ldots,y_N} \left(\log \prod_{i=1}^{N} f(\boldsymbol{Y}_i|\boldsymbol{\Theta}) \right) f(y_1, \ldots, y_N|\boldsymbol{Y}_1, \ldots, \boldsymbol{Y}_N, \boldsymbol{\Theta}^{(b)}) \\ &\quad = \sum_{y_1,\ldots,y_N} \left(\log \prod_{i=1}^{N} \left(\sum_{k=1}^{K} f(Y_i|\boldsymbol{\Theta}) f(\boldsymbol{y}_i = k|\boldsymbol{\Theta}) \right) \right) \left(\prod_{i=1}^{N} f(y_i|\boldsymbol{Y}_i, \boldsymbol{\Theta}^{(b)}) \right) \end{aligned} \tag{11}$$

M-step: find $\Theta^{(b+1)} = \arg\max_{\Theta} Q(\Theta, \Theta^{(b)})$.
Clearly, in the above EM algorithm, the objective likelihood function and therefore the optimal clustering results depend on the prior conditional probability of y_i given Θ. Preliminary simulation comparisons among *PoissonL, PoissonC*, and the EM algorithm show that they perform similarly. Further comparisons of these algorithms are ongoing.
5. The new measures were employed into a K-means clustering procedure to perform the analysis. The algorithm used for iteratively updating cluster assignments is an algorithm implemented in the C clustering library, which is publicly available ***(21)***. The algorithm terminates when no further reassignments take place. Because the convergent results of this algorithm are quite sensitive to the initial cluster assignments, usually, the algorithm should be run on many different initials to obtain an optimal result. The within-cluster dispersion should better be recorded to compare the results.
6. When the users are not confident about whether the data are Poisson-like or not, a good choice could be *Eucli* (K-means algorithm using Euclidian distance). Our experience tells that *Eucli* is quite stable and reliable when it is applied to data that are appropriately postnormalized according to the clustering purpose, i.e., the data can be rescaled to reduce the effects of magnitude if only the shape of expression pattern determines the clustering. Good measurement methods should consider both magnitude and shape of the expression patterns.
7. For clustering analysis, tags with only one count are usually excluded from analysis due to sequencing error problem. To select the potential most biologically relevant genes, tags with less than 2–10 counts can be excluded depending on how large the SAGE libraries are and how many total number of tags is intended to analyze.
8. Annotation of SAGE tags is through SAGEtag to UniGene mapping ***(24)***. The mapping is based on "SAGEmap_tag_ug-rel.Z" provided by the National center for Biotechnology Information (ftp://ftp.ncbi.nlm.nih.gov/pub/sage/map/), which contains all annotated SAGE tags mapping to UniGene clusters. However, there

are many ambiguities on the SAGE tag annotation. There are tag sequencing errors ***(25)***, and also the mapping between tags and genes can be nonunique. In one planned project, we propose to reduce this error by inferring the real expression level of genes from "weighted" counts of all mapped tags, where the weights can be determined by the available mapping quality information. An EM algorithm is feasible for this task.

References

1. Blackshaw, S., Fraioli, R. E., Furukawa. T., and Cepko, C. L. (2001) Comprehensive analysis of photoreceptor gene expression and the identification of candidate retinal disease genes. *Cell* **107**, 579–589.
2. Zhang, L., Zhou, W., Velculescu, V. E., et al. (1997) Gene expression profiles in normal and cancer cells. *Science* **276**, 1268–1272.
3. Velculescu, V. E., Zhang, L., Vogelstein, B., and Kinzler, K. W. (1995) Serial analysis of gene expression. *Science* **270**, 484–487.
4. Buckhaults, P., Zhang, Z., Chen, Y. C., et al. (2003) Identifying tumor origin using a gene expression-based classification map. *Cancer Res.* **63**, 4144–4149.
5. Porter, D., Weremowicz, S., Chin, K., et al. (2003) A neural survival factor is a candidate oncogene in breast cancer. *Proc Natl Acad Sci USA*. **100**, 10,931–10,936.
6. Margulies, E. H. and Innis, J. W. (2000) eSAGE: managing and analysing data generated with serial analysis of gene expression (SAGE). *Bioinformatics* **16**, 650–651.
7. van Ruissen, F., Jansen, B. J., de Jongh, G. J., van Vlijmen-Willems, I. M., and Schalkwijk, J. (2002) Differential gene expression in premalignant human epidermis revealed by cluster analysis of serial analysis of gene expression (SAGE) libraries. *FASEB J*. **16**, 246–248.
8. Audic, S. and Claverie, J. M. (1997) The significance of digital gene expression profiles. *Genome Res*. **7**, 986–995.
9. Madden, S. L., Galella, E. A., Zhu, J., Bertelsen, A. H., and Beaudry, G. A. (1997) SAGE transcript profiles for p53-dependent growth regulation. *Oncogene*, **15**, 1079–1085.
10. Man, M. Z., Wang, X., and Wang, Y. (2000) POWER_SAGE: comparing statistical tests for SAGE experiments. *Bioinformatics*. **16**, 953–959.
11. Blackshaw, S., Kuo, W. P., Park, P. J., et al. (2003) MicroSAGE is highly representative and reproducible but reveals major differences in gene expression among samples obtained from similar tissues. *Genome Biol.* **4**, R17.
12. Quackenbush, J. (2001) Computational analysis of microarray data. *Nat. Rev. Genet*. **2**, 418–427.
13. Fraley, C. (1998) Algorithms for model-based Gaussian hierarchical clustering. *SIAM Journal on Scientific Computing* **20**, 270–281.

14. Yeung, K. Y., Fraley, C., Murua, A., Raftery, A. E., and Ruzzo, W. L. (2001) Model-based clustering and data transformation for gene expression data. *Bioinformatics* **17**, 977–987.
15. Fraley, C. and Raftery, A. E. (2002) Model-based clustering, discriminant analysis and density estimation. *Journal of the American Statistical Association* **97**, 611–631.
16. Cai, L., Huang, H., Blackshaw, S., Liu, J. S., Cepko, C. L., and Wong, W. H. (2004) Clustering analysis of SAGE data using a Poisson approach. *Genome Biol.* **5**, R51.
17. Ewens, W. J. and Grant, G. R. (2001) *Statistical Methods in Bioinformatics*. Springer Verlag, Germany.
18. Eisen, M. B., Spellman, P. T., Brown, P. O., and Botstein, D. (1998) Cluster analysis and display of genome-wide expression patterns. *Proc. Natl. Acad. Sci. USA* **95**, 14,863–14,868.
19. Hartigan, J. (1975) *Clustering Algorithms*. Wiley, New York.
20. Celeux, G. and Govaert, G. (1992) A classification EM algorithm for clustering and two stochastic versions. *Computational Statistics & Data Analysis* **14**, 315–332.
21. de Hoon, M. J. L., Imoto, S., Nolan, J., and Miyano, S. (2004) Open source clustering software. *Bioinformatics* **20**, 1453–1454.
22. Blackshaw, S., Harpavat, S., Trimarchi, J., et al. (2004) Genomic analysis of mouse retinal development. *PLoS Biology* **2**, E247.
23. Tseng, G. C. and Wong, W. H. (2004) A resampling method for tight clustering: with an application to microarray analysis. *Biometrics* **61**, 10–16.
24. Lash, A. E., Tolstoshev, C. M., Wagner, L., et al. (2000) SAGEmap: a public gene expression resource. *Genome Res*. **10**, 1051–1060.
25. Beissbarth, T., Hyde, L., Smyth, G. K., et al. (2004) Statistical modeling of sequencing errors in SAGE libraries. *Bioinformatics*. **4(Suppl 20)** 1:I31–I39.