A k-mean-directions Algorithm for Fast Clustering of Data on the Sphere

Ranjan Maitra and Ivan P. Ramler [*]

Abstract

A k-means-type algorithm is proposed for efficiently clustering data constrained to lie on the surface of a p-dimensional unit sphere, or data that are mean-zero-unit-variance standardized observations such as those that occur when using Euclidean distance to cluster time-series gene expression data using a correlation metric. We also provide methodology to initialize the algorithm and to estimate the number of clusters in the dataset. Results from a detailed series of experiments show excellent performance, even with very large datasets. The methodology is applied to the analysis of the mitotic cell division cycle of budding yeast dataset of Cho, Campbell, Winzeler, Steinmetz, Conway, Widicka, Wolfsberg, Gabrielian, Landsman, Lockhart and Davis (1998). The entire dataset has not been analyzed previously, so our analysis provides an understanding for the complete set of genes acting in concert and differentially. We also use our methodology on the submitted abstracts of oral presentations made at the 2008 Joint Statistical Meetings (JSM) to identify similar topics. Our identified groups are both interpretable and distinct and the methodology provides a possible automated tool for efficient parallel scheduling of presentations at professional meetings.

KEYWORDS: directional, information retrieval, Langevin/von-Mises distribution, MC toolkit, microarrays, *spkmeans*

1. INTRODUCTION

Cluster analysis - an unsupervised method for grouping similar observations - is a common technique for analyzing multivariate datasets (Everitt, Landau and Leesem 2001; Fraley and Raftery 2002; Hartigan 1985; Kaufman and Rousseuw 1990; Kettenring 2006; Murtagh 1985; Ramey

[*] Ranjan Maitra is Associate Professor in the Department of Statistics at Iowa State University, Ames, IA 50011-1210, USA and Ivan P. Ramler is Visiting Assistant Professor in the Department of Mathematics, Computer Science and Statistics at St. Lawrence University, 23 Romoda Dr., Canton, NY 13617-1423, USA. This research was supported in part by the National Science Foundation CAREER Grant # DMS-0437555 and by the National Institutes of Health DC–0006740.

1985). Most methods can be classified as either hierarchical procedures or non-hierarchical partition-optimization algorithms. The former create tree-like structures and encompass both *agglomerative* methods (which start with individual objects as distinct groups, and proceed by combining the most similar ones) and *divisive* schemes that group all objects together at the start and subdivide them into subgroups that are most dissimilar from each other. Partitional algorithms usually employ locally optimal grouping strategies such as k-means (Hartigan and Wong 1979) or k-medoids (Kaufman and Rousseuw 1990), or probabilistically through mixture-model-based clustering (Celeux and Govaert 1995; Fraley and Raftery 1998; Fraley and Raftery 2002).

The iterative k-means algorithm proposed by MacQueen (1967) to find an optimal partition of n objects into K (given) groups such that the total sum of squared Euclidean distances from each observation to its closest group center is locally minimized is one of the oldest partitioning methods. Hartigan and Wong (1979) provided a simple and efficient implementation of k-means, when the distance metric for grouping is Euclidean. However, this algorithm may not be appropriate for other grouping similarity metrics. In particular, we consider two application areas in informatics, where either cosine similarity or correlation is the desired metric to identify similar observations.

1.1 Application to Informatics

This section illustrates scenarios in bioinformatics and information retrieval where a non-Euclidean similarity metric is most appropriate for grouping observations that are most alike. The first consists of clustering time-course microarray gene expression data, while the second considers clustering text documents.

1.1.1. Identifying Similarly-acting Genes in Microarray Gene Expression Time Series Data

The process by which a gene gets turned on to produce ribo-nucleic acid (RNA) and proteins is called *gene expression* and can be measured by the amount and/or activity of RNA. Related to this is *gene expression profiling* which measures the expression level of many genes in numerous cell types. This is typically achieved via microarray studies which involve analyzing gene expressions to provide insight into how cells respond to changing factors and needs `http://www.ncbi.nlm.nih.gov`. The functionality of some genes is well-established, but this is not true for many others. One approach to understanding gene functionality and characteristics is to determine groups of similarly-acting genes. Then properties of those genes that are not so well-established can be deduced by comparing their group memberships in relation to the well-understood ones. To this end, one may group gene expression profiles that have similar shapes or patterns over time, even though the magnitude of the expressions may differ greatly. Time series gene expression experiments account for about one-third of all microarray studies (Barrett, Suzek, Troup, Wilhite, Ngau, Ledoux, Rudnev, Lash, Fujibuchi and Edgar 2005) and cover a wide range of biological areas including applications in cell cycle (Cho et al. 1998; Zeeman, Tiessen, Pilling, Kato, Donald and Smith 2002; Whitfield, Sherlock, Saldanha, Murray, Ball, Alexander, Matese, Perou, Hurt, Brown and Botstein 2002; Giacomelli and Nicolini 2006). In this context, the correlation coefficient between two genes serves as an intuitive measure of the similarity between their activities (Eisen, Spellman, Brown and Botstein 1998).

The correlation similarity between two gene expression profiles has a connection with Euclidean distance in that its complement from unity is a scaled version of the squared Euclidean distance between the profiles' mean-zero-centered and unit-variance-standardized counterparts (see AppendixA). Thus clustering such data is equivalent to grouping these transformed observations that lie on the surface of a sphere and are orthogonal to the unit vector. In Section 4.1, we revisit the popular budding yeast gene expression dataset of Cho et al. (1998), analyzing it in its entirety to identify similar-acting genes. As analyzing the entire dataset presents a challenge to most clustering algorithms, most authors have typically only analyzed a subset of the data. There is some justification for this, in the sense that the subset represents a set of the most active genes, but it would also be helpful to get a better understanding on the genes in the entire dataset.

1.1.2. Clustering Text Documents The large amount of electronic text documents readily available presents a growing challenge for researchers. For instance, there are over twenty billion documents and webpages publicly available on the worldwide web. With such huge amounts of textual information, effectively grouping documents is a statistical task of potential practical interest. For example, it may be important to cluster documents for ready catalog and reference to interested parties. Given the huge numbers of documents, an automated approach to categorizing text documents is the only viable option.

Often text documents are processed by first listing all their unique words and then by removing both "high-frequency" and "low-frequency" words that provide little or no information in determining groups (Dhillon and Modha 2001). Thus a *bag of words* on which to categorize the documents is created. The text documents are then processed to form a *document-term frequency matrix,* where each row in the matrix represents a *document vector* containing the frequency of occurrence of each member in the bag of words. Weighting schemes are also used to improve separation between documents (Salton and Buckley 1988; Kolda 1997), with the most common ones normalizing each observation to lie on a high-dimensional unit sphere in an attempt to temper the effect of differing lengths of each document (Singhal, Buckley, Mitra and Salton 1996), resulting in document vectors of unit length each.

The result of standardizing the vectors is that the natural method of measuring similarity between documents is the inner product (also called cosine similarity) which is widely used in document clustering and information retrieval (Dhillon and Modha 2001; Frakes and Baeza-Yates 1992; Salton and McGill 1983). Thus, clustering according to this metric is again equivalent to grouping sphered data, but there is an additional complication: even medium-sized corpora can, after processing, have very high-dimensional document-term frequency matrices, representing a challenge for most clustering algorithms. We address such a dataset of great practical import to many statisticians in Section 4.2.

1.2 Background and Related Work

There are a few partitional methods that specifically address the issue of clustering spherically constrained data (Dhillon and Modha 2001; Banerjee, Dhillon, Ghosh and Sra 2005; Dortet-Bernadet and Wicker 2008). Banerjee et al. (2005) and Dortet-Bernadet and Wicker (2008) propose different mixture models, and employ an expectation-maximization (EM) approach. Unfortunately, in many situations, the EM algorithm is inapplicable to large-dimensional datasets. There is also a version of k-means proposed by Dhillon and Modha (2001) using cosine similarity. Called *spkmeans*, this algorithm replaces the Euclidean distance metric in the base k-means algorithm by cosine similarity. The algorithm does not inherit the properties of Hartigan and Wong (1979)'s efficient implementation of the k-means algorithm, and can be slow, potentially performing poorly in many datasets. Further, these algorithms are very sensitive to initial values, strategies for choosing which are often left unaddressed. Another difficult issue, unaddressed in Dhillon and Modha (2001) or in Banerjee et al. (2005), is in determining the optimal number of clusters (K) in a dataset. Dortet-Bernadet and Wicker (2008) study some methods (Akaike 1973; Akaike 1974; Schwarz 1978; Tibshirani, Walther and Hastie 2003) in this context, but their recommendation of Akaike (1974)'s AIC is hardly convincing, given the known tendency of AIC to overestimate the number of clusters by this criterion.

In this paper we propose a k-mean-directions algorithm, modifying the core elements of Hartigan and Wong's (1979) efficient k-means implementation to apply to sphered data. Our algorithm is general enough to incorporate the additional constraint of orthogonality to the unit vector, and thus extends to the situation of clustering using the correlation metric. Section 2 describes the modifications to the standard k-means algorithm of Hartigan and Wong (1979), develops an approach for initialization and proposes a method for estimating the number of clusters. The procedure is extensively evaluated in Section 3 through simulation experiments for varying dimensions and cluster separations. The time-course dataset on the budding yeast gene expression profiles and another on text documents are analyzed in detail in Section 4. We conclude with some discussion. Additionally, we provide an appendix which outlines the relationship between the correlation between two gene expression profiles and the squared Euclidean distance their standardized versions. We also have an online supplement providing further detailed experimental illustrations and performance evaluations. Sections, figures and tables in the supplement referred to in this paper are labeled with the prefix "S-".

2. METHODOLOGY

Let $\boldsymbol{X}_1, \boldsymbol{X}_2, \ldots, \boldsymbol{X}_n$ be p-variate observations from $\mathcal{S}^p = \{\boldsymbol{x} \in I\!R^{p+1} : \boldsymbol{x}'\boldsymbol{x} = 1\}$. At this point, we make no distinction between whether the observations are in $\mathcal{S}^p$ or $\mathcal{S}^p_{\perp\mathbf{1}} = \{\boldsymbol{x} \in \mathcal{S}^p : \boldsymbol{x}'\mathbf{1} = 0\}$. Our objective is to find class indicators $\zeta_1, \zeta_2, \ldots, \zeta_k$ such that given K, the objective function

$$Obj_K = \sum_{i=1}^{n} \sum_{k=1}^{K} I(\zeta_i = k)(1 - \boldsymbol{X}_i'\boldsymbol{\mu}_k) \tag{1}$$

is minimized. Here $\boldsymbol{\mu}_k$ represents the mean direction vector of the observations in the kth partition, lies in the same space ($\mathcal{S}^p$ or $\mathcal{S}^p_{\perp\mathbf{1}}$) as the $\boldsymbol{X}_i$s, and needs to be estimated, even though that is not necessarily the main goal of the exercise. Note that minimizing Obj_K is equivalent to minimizing $\sum_{i=1}^{n}\sum_{k=1}^{K} I(\zeta_i = k)(\boldsymbol{X}_i - \boldsymbol{\mu}_k)'(\boldsymbol{X}_i - \boldsymbol{\mu}_k)$ or maximizing $\sum_{i=1}^{n}\sum_{k=1}^{K} I(\zeta_i = k)\boldsymbol{X}_i'\boldsymbol{\mu}_k$. Additionally, K itself needs to be estimated, though we keep that issue aside until Section 2.3.

A model-based interpretation for the above is provided as follows: let $\boldsymbol{X}_i$ be independently distributed from a p-variate Langevin density $\mathcal{L}_p(\boldsymbol{\mu}_{\zeta_i}; \kappa)$ given by $f(\boldsymbol{x}) = c_p^{-1}(\kappa)\exp\{\kappa\boldsymbol{x}'\boldsymbol{\mu}_{\zeta_i}\}$ where κ is the common concentration parameter, and $c_p(\kappa) = \frac{\kappa^{p/2-1}}{(2\pi)^{p/2}I_{p/2-1}(\kappa)}$ when $\boldsymbol{X}_i$'s lie in $\mathcal{S}^p$ and where $I_\nu(\cdot)$ denotes the modified Bessel function of the first kind and order ν. (The constant of integration $c_p(\kappa)$ is appropriately modified when $\boldsymbol{X}_i$'s are in $\mathcal{S}^p_{\perp\mathbf{1}}$.) Under this setup, the joint likelihood for all the observations is provided by

$$L(\zeta_1,\zeta_2,\ldots,\zeta_n,\boldsymbol{\mu}_1,\boldsymbol{\mu}_2,\ldots,\boldsymbol{\mu}_k,\kappa;\boldsymbol{X}_1,\boldsymbol{X}_2,\ldots,\boldsymbol{X}_n) = c_p^{-n}(\kappa)\exp\left\{\kappa\sum_{i=1}^{n}\sum_{k=1}^{K} I(\zeta_i = k)\boldsymbol{X}_i'\boldsymbol{\mu}_k\right\}, \tag{2}$$

maximizing which – with respect to the $\boldsymbol{\mu}$'s and ζ's in the presence of the nuisance parameter κ – is equivalent to finding the class indicators and means minimizing (1). Note that (2) looks similar to the likelihood of the complete data in the case of mixtures-of-homogeneous-Langevins with uniform mixing proportions, but note that there, $\zeta_1,\zeta_2,\ldots,\zeta_n$ are the missing group indicator observations, not parameters as in our model. Also, in that case, parameter estimates for $\boldsymbol{\mu}$'s and κ are obtained and the focus of the problem is the maximization of the likelihood (with the help of the EM algorithm). The most likely class indicator for each observation is obtained *post hoc* by choosing the one that has maximum posterior probability calculated by fitting these parameter estimates. In the model-based interpretation provided for minimizing (1), ML estimates for ζs and $\boldsymbol{\mu}$ are jointly obtained maximizing (2). We now provide a ready modification of the k-means algorithm which makes iterative and local optimization of (1) possible.

2.1 The k-mean-directions Algorithm

For a given K and initial cluster centers $\{\hat{\boldsymbol{\mu}}_k; k = 1,\ldots,K\}$, the general strategy is to partition the dataset into K clusters, update the cluster mean-directions and iterate until convergence, which is to a local optimum of the objective function (1). In this context, note that when cluster memberships are provided, (1) is minimized at $\hat{\boldsymbol{\mu}}_k = \|\bar{\boldsymbol{X}}_k\|^{-1}\bar{\boldsymbol{X}}_k$, where $\bar{\boldsymbol{X}}_k = n_k^{-1}\sum_{i=1}^{n} I(\zeta_i = k)\boldsymbol{X}_i$ and $n_k = \sum_{i=1}^{n} I(\zeta_i = k)$ is the number of observations in the kth group. This is essentially the basis of the *spkmeans* algorithm of Dhillon and Modha (2001), which however, may be inadvisable in very large datasets and in cases when we need many runs (such as in the case when the number of clusters is also required to be estimated). Therefore, we develop a fast and computationally efficient algorithm, built in the same spirit as Hartigan and Wong (1979)'s suggestion for k-means which employs reductions and restricts recomputations to only when necessary. In doing so, we define a *live set* of clusters containing only those groups with a potential for reallocation among their observations. Potential reallocation of group members is itself effected in either the *optimal transfer* or the *quick transfer* stages. In the first case, calculations and reallocations are made with regard to all observations relative to clusters in the *live set*, while the quick transfer stage only checks for, and potentially updates, mean directions and memberships of recently reallocated groups. We provide the specifics of the algorithm next:

1. *Initial assignments and calculations.* Given K initializing cluster mean directions $\hat{\boldsymbol{\mu}}_1, \hat{\boldsymbol{\mu}}_2, \ldots, \hat{\boldsymbol{\mu}}_K$, find the two closest mean directions for each observation $\boldsymbol{X}_i, i = 1, 2, \ldots, n$, denoting the corresponding groups by C_{1i} and C_{2i} respectively. Assign $\boldsymbol{X}_i$ to the cluster C_{1i} (thus, the current $\zeta_i = C_{1i}$). Using this assignment, update the K mean-directions to be the mean directions of observations in each of the K classes. For $k = 1, 2, \ldots, K$, let $\nu_k^- = (n_k - 1)^2 - n_K^2 \|\bar{\boldsymbol{X}}_k\|^2 - 1$ and $\nu_k^+ = (n_k + 1)^2 - n_K^2 \|\bar{\boldsymbol{X}}_k\|^2 - 1$. These values are used in calculating the change in (1) when removing (ν_k^-) and adding (ν_k^+) an observation to and from a cluster. All clusters are in the *live set* at this stage.

2. The algorithm now moves to the iterative optimal and quick transfer stages and continues until the *live set* is empty.

3. *Membership in live set.* Initially, all clusters are in the *live set*. Any cluster whose membership and mean direction are updated get included in the live set. Further, if any cluster is updated in the previous quick transfer stage of Step 5, it belongs to the live set all through the next *optimal transfer stage*. Further, any cluster not updated in the last n optimal transfer steps exits the live set.

4. *Optimal transfer stage.* For each $\boldsymbol{X}_i, i = 1, 2, \ldots, n$, we calculate the maximum reduction in the objective function (1) by replacing ζ_i with another class (k_i, say). If ζ_i is in the live set, we consider all other classes as replacement candidates. Otherwise, we restrict attention only to classes in the live set. In either case, the maximum reduction is given by the quick calculation $\omega_i = (n_{k_i} + 1)(\nu_{k_i}^+ - 2n_{k_i} \bar{\boldsymbol{X}}'_{k_i} \boldsymbol{X}_i) - (n_{\zeta_i} - 1)(\nu_{\zeta_i}^- + 2n_{\zeta_i} \bar{\boldsymbol{X}}'_{\zeta_i} \boldsymbol{X}_i)$. If $\omega_i > 0$, then the only quantity to be updated is $C_{2i} = k_i$. Otherwise, reallocation takes place if $\omega_i < 0$ with $C_{1i} = k_i$, and the objective function, $\mu_{k_i}, \mu_{\zeta_i}, n_{k_i}, n_{\zeta_i}, \nu_{k_i}^+, \nu_{\zeta_i}^+, \nu_{k_i}^-$, and $\nu_{\zeta_i}^-$ are updated with the corresponding changes. Also C_{2i} and $\zeta_i = C_{1i}$ are updated and the old ζ_i and k_i are placed in the live set. We make one pass through the dataset at the optimal transfer stage unless the live set is empty, in which case the algorithm terminates.

5. *Quick transfer stage.* The quick transfer stage, as its name suggests is a quick pass, differing from the optimal transfer stage in that, it does not go through many potential candidate classes. Instead, for each observation $\boldsymbol{X}_i, i = 1, 2, \ldots, n$, it swaps ζ_i (equivalently, C_{1i})

with C_{2i} if either of the composition of these two clusters has changed in the last n steps and doing so leads to a negative ω_i (as defined above). The corresponding objective function as well as the associated n_k's, $\boldsymbol{\mu}_k$'s and ν^+'s and ν^-'s are also updated and both C_{1i} and C_{2i} enter the live set as mentioned in Step 3. We continue with the passes through the dataset until no quick transfers have happened for the last n stages.

Note that the swapping rule in Steps 4 and 5 follows from Mardia and Jupp (2000), pp 166 as the change in the objective function can be written in terms of $\|\bar{\boldsymbol{X}}\|$. An additional result is that (1) can be computed as $Obj_K = \sum_{k=1}^{K} n_k(1 - \|\bar{\boldsymbol{X}}_k\|)$ providing a quick way of obtaining the final value of (1) which will be used in estimating the number of clusters in Section 2.3.

The k-mean-directions is an iterative algorithm, finding local optima in the vicinity of its initialization, so starting values for mean directions need to be specified for it to proceed. We address a strategy for choosing starting values next.

2.2 Initialization of Cluster Centers

Initialization of iterative algorithms such as k-means can have tremendous impact on performance. Common methods for initializing k-means include randomly chosen starts or using hierarchical clustering to obtain K initial centers. While these can be used directly to initialize our k-mean-directions algorithm, they have been demonstrated to perform poorly for k-means in many situations by Maitra (2009). He also suggested a multi-staged deterministic initializing algorithm for finding initial values which finds a large number of local modes, classifying the observations and then choosing K representatives from the most widely-separated ones. This algorithm was the best performer in a majority of cases for several dimensions and numbers of clusters. The constrained structure of our dataset means that direct application of the iterative multi-stage method is not possible so we adapt the algorithm to this context next:

1. We use an approach similar to Maitra (2009) to obtain K initializing centers. Specifically, our first goal is to obtain local modes along each dimension of $\boldsymbol{X}$. To do so, use one-dimensional k-means initialized with $\lceil (K(p-1))^{1/(p-1)} \rceil$ equi-spaced quantiles, where $\lceil x \rceil$ denotes the smallest integer greater than or equal to x. The product set of the resulting one-dimensional modes are then projected back to lie on $\mathcal{S}^p$ (by dividing each element in the set by its norm). This set now forms a set of potential multivariate local modes that lie on $\mathcal{S}^p$. We prune this set by discarding all candidates that are not closest to any observations in $\boldsymbol{X}$.

These remaining K^* local modes are then used to initialize the k-mean-directions algorithm to produce K^* local modes in $\boldsymbol{X}$.

2. Obtain a representative from the K most widely separated modes. To do so, we use hierarchical clustering with single linkage on these K^* local modes and cut the resulting tree into K groups. We then classify each observation in $\boldsymbol{X}$ to one of these K groups by its closest Euclidean distance to one of these K centers. The mean directions of the resulting classifications are used to obtain K initial mean directions for $\boldsymbol{X}$.

The above algorithm can also be applied to the case when the observations $\boldsymbol{X}_i \in \mathcal{S}^p_{\perp \mathbf{1}}$. In this case, we find a $p \times (p-1)$ matrix $\boldsymbol{V}$ with columns orthogonal to $\mathbf{1}$ (the vector of ones) to project the data onto $\mathcal{S}^{p-1}$. $\boldsymbol{V}$ can be practically obtained by finding the first $p-1$ right singular vectors computed from the singular-value decomposition based on any p observations from $\mathcal{S}^p_{\perp \mathbf{1}}$. Let $\boldsymbol{U} = \boldsymbol{X}\boldsymbol{V}$ be the resulting projection of the data matrix. We continue with $\boldsymbol{U} \in \mathcal{S}^{p-1}$ in place of $\boldsymbol{X}$ in the above algorithm. Also, the initializing mean directions obtained in Step 2 are projected back onto $\mathcal{S}^p_{\perp \mathbf{1}}$ by using the $p \times (p-1)$ matrix $\boldsymbol{V}'$. Note that this entire exercise is possible because the objective function of (1) is invariant to transformation by the orthogonal matrix $\boldsymbol{V}'$.

In simulation experiments reported in the supplementary materials, we note that the above algorithm performs well in the majority of cases for higher dimensions with large number of clusters, but performance is worse for lower dimensions. Thus, in order to guard against underperformance, and for all dimensions, we supplement the chosen initial values with centers drawn from a hierarchically clustered solution with Ward's criterion (Ward 1963) as well as the best of R random starts. Specifically for the latter, we choose R sets of K randomly chosen (distinct) observations from $\boldsymbol{X}$. For each set, we classify the observations in $\boldsymbol{X}$ into K groups based on their closeness to these K values and calculate the corresponding value of (1). The set minimizing (1) is our proposed random start and is compared with the values of (1) obtained at the hierarchically clustered initializer and at the starting values provided by our multi-staged deterministic algorithm above. The initializing values which yield the lowest value of (1) is chosen as our starting value for the algorithm.

Note that the random starts approach as suggested above does not actually run the k-mean-directions algorithm above for each random start, but evaluates (1) at each combination, choosing the one that is optimal. This makes exploring a large number of initial values possible. Indeed, for our experiments in Section 3, we take $R = 1000$. Further, for high-dimensional datasets such as the text documents introduced in Section 1.1.2, the deterministic portion of the initializer can be very computationally taxing. In such situations, it may be more feasible to simply implement only the randomly selected starting directions discussed previously and for datasets with a smaller number of observations, supplement this with means obtained using hierarchical clustering with Ward's linkage, again choosing the set of centers that provides the minimal value of (1).

2.3 Estimating the Number of Clusters

We have hitherto assumed that the number of clusters, K, is known. As this is rarely true in practice, we need methods to determine an optimal estimate of K. While there has been a large body of work in this regard for k-means and other partitional algorithms, there has been almost no attention paid to this aspect even in the context of $spkmeans$. We have experimented with several adaptations of classical partitional algorithms (the Gap statistic, BIC, modified Marriott's criterion, etc.) for addressing this issue and have experimentally found our most promising pick to be an approach relating the largest relative change in the optimized objective function with an increase in number of clusters. Formally therefore, we look for the largest relative change in the final objective function at the termination of the k-mean-directions algorithm when we go from k to $k+1$ clusters. Thus, denoting Obj_k as the optimized (converged) value of (1), we choose $\hat{K} = k \in 2, \dots K_{T-1}$ that maximizes $\frac{Obj_{k+1}}{Obj_k} - \frac{Obj_k}{Obj_{k-1}}$. Additionally, and perhaps different from the classical non-transformed Euclidean-distance case, in some scenarios it may be necessary to determine if $K = 1$. This can be done by comparing Obj_1 to "Obj_0" which can be thought of the value of the objective function if no clusters are present (i.e., the observations are uniformly distributed in $\mathcal{S}^p$). In this scenario, one can use $E(Obj_0) = E_x(\sum_{i=1}^n 1 - X_i'\mu) = \sum_{i=1}^n E_{\theta_i}(1 - \cos\theta_i) = 2n$ to replace Obj_0. However, caution needs to be exercised when using this as in our experience, $E(Obj_0)$ tends to be much larger than Obj_1 when clusters are located in a subspace of $\mathcal{S}^p$.

The motivation behind this proposed method is that as K increases, the within-cluster variation goes down corresponding to an increase in the concentration of observations around the mean direction in each group. Further, the concentration should sharply increase when going from $K-1$ to K clusters, and should increase much more slowly when the number of clusters goes past the true K. If we assumed that the observations arose from a mixture of Langevins with common concentration parameter κ, an appropriate way of determining the largest relative change in the concentration would be to find $k \in 2, \dots K_T - 1$ that maximizes $\frac{\kappa_k}{\kappa_{k+1}} - \frac{\kappa_{k-1}}{\kappa_k}$. Then we can derive the maximum likelihood estimate for κ based on the high-concentration approximation

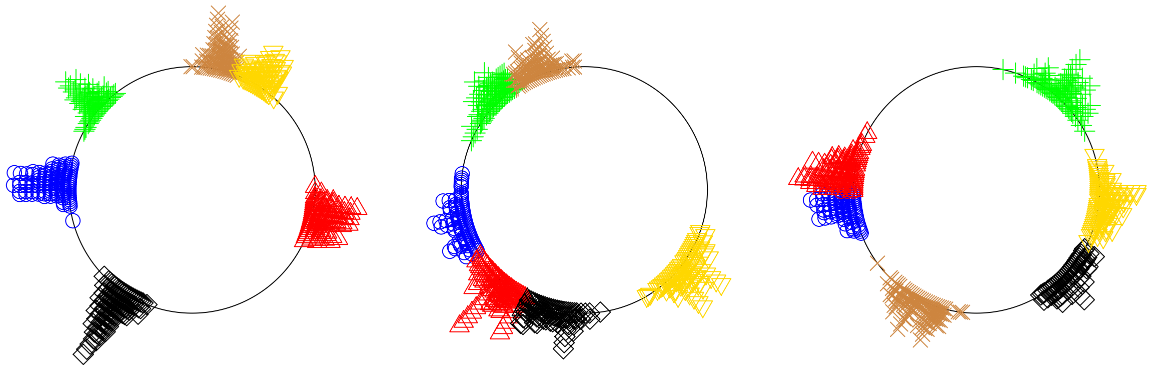
$2\kappa\left(n - \Sigma_{i=1}^{n}\boldsymbol{X}'_i\boldsymbol{\mu}\right) \dot{\sim} \chi^2_{n(p-1)}$ (Watson 1983) as $\hat{\kappa} = \frac{n(p-2)-2}{2\left(n - \Sigma_{i=1}^{n}\Sigma_{j=1}^{k}\zeta_{ij}\boldsymbol{X}'_i\boldsymbol{\mu}_l\right)} = \frac{n(p-2)-2}{Obj_k}$ where ζ_{ij} indicates the classification of observation i in cluster j. Finally, substituting $\hat{\kappa}$ and simplifying the previous ratio, we arrive at the proposed method involving only the objective function (1). We note further that the algorithm is fast and built entirely on the byproduct – final value of (1) – of the k-mean-directions algorithm. As such it is applicable to all datasets which can handle the k-mean-directions algorithm (i.e., spherically constrained datasets where Euclidean distances are the appropriate metric). This makes it practical to use in a large number of situations such as in the gene expressions and document clustering applications.

In this section therefore, we have proposed a fast and efficient iterative approach to clustering data on a sphere. We have also provided approaches to initializing the algorithm and to determining the optimal number of clusters. We now study the performance of each of the many aspects of the suggested algorithms.

3. EXPERIMENTAL EVALUATIONS

The proposed methodology was comprehensively evaluated through a large-scale series of simulation experiments. For brevity, we only report the overall performance of k-mean-directions with K estimated using the procedure from Section 2.3 and refer to the supplement for detailed descriptions on the actual performance of the k-mean-directions algorithm with K known (Section S-1.1), and its initialization (Section S-1-2). Additionally, in the supplement we also compare in detail performance of k-mean-directions to the standard k-means algorithm (Section S-1.3), as well as, a comparison of computing time between k-mean-directions and $spkmeans$ (Section S-1.4). We note here that as the number of dimensions increases, k-mean-directions performs faster than $spkmeans$. Our assessment is presented graphically for two-dimensional examples, and numerically via the adjusted Rand measure ($\mathcal{R}_a$) of Hubert and Arabie (1985) for 2, 6 and 10 dimensions. The experimental suite covered a collection of dimensions (p), number of true clusters (K), clustering difficulty (c) and number of observations (n).

Our simulation datasets were generated from an equi-proportioned mixture of K p-dimensional Langevins with common concentration parameter κ. Here, the Langevin distribution provides a natural reference distribution as under the assumption of a common concentration parameter, κ, the objective function (1) is closely related to the likelihood of a mixture of Langevins (see Section 2). The difficulty of a clustering problem is directly impacted by the overlap (or lack of separation) between clusters, so we use a modification of Dasgupta (1999) $c - separation$



(a) $c = 4, \mathcal{R}_a = 1.0$ (b) $c = 2, \mathcal{R}_a = 0.995$ (c) $c = 1, \mathcal{R}_a = 0.986$

Figure 1: Clustering via k-mean-directions on datasets with $K = 6$ clusters generated with (a) $c = 4$, (b) $c = 2$, and (c) $c = 1$. Color and character represent true and identified clusters, respectively.

modified for the Langevin distribution. Further, following Maitra (2009), we define a set of p-variate Langevin densities to be *exact–c–separated* if for every pair $\mathcal{L}_p(\boldsymbol{\mu}_i; \kappa_i)$ and $\mathcal{L}_p(\boldsymbol{\mu}_j; \kappa_j)$, $\|\boldsymbol{\mu}_i - \boldsymbol{\mu}_j\| \geq c\sqrt{1/\max(\kappa_i, \kappa_j)}$. Different choices of c give rise to different separations between clusters. In our experiments, we chose $c = 1, 2$ and 4 to represent clustering situations with poor, moderate and good separation, respectively. Additionally, although the algorithm is evaluated based solely on data lying in $\mathcal{S}^p$, we note that this setup is equivalent to evaluating k-mean-directions for data that lie in $\mathcal{S}^p_{\perp \mathbf{1}}$ for $p = 3, 7$ and 11 and contend that performance will be nearly identical, given the lower-dimensional projection into S^{p-1} using the $\boldsymbol{V}$ matrix of Section 2.2.

3.1 Illustrative Two-dimensional Experiments

For each of these experiments, $K = 6$ and $n = 500$. In all three cases, we correctly estimated K, and we display our clustering results by means of a stacked circular plot (Lund and Agostinelli 2007) in Figure 1 for the different levels of c. Performance is excellent throughout. The grouping was perfect ($\mathcal{R}_a = 1.0$) for $c = 4$, had one misclassification ($\mathcal{R}_a = 0.995$) for $c = 2$ and three misclassifications ($\mathcal{R}_a = 0.986$) for $c = 1$. Thus, there is some very minor degradation in performance with increasing difficulty of the clustering problem.

3.2 Large-sample Simulation Experiments

We have also conducted a series of large-sample higher-dimensional simulation experiments. Our primary objective for a k-mean-directions algorithm is computational efficiency and the ability

Table 1: Summary of k-mean-directions with estimated number of clusters ($\hat{K}$). Each cell contains the median $\hat{K}$ (top left), median adjusted Rand value ($\mathcal{R}_a$; top right), interquartile of $\hat{K}$ (bottom left) and interquarile of $\mathcal{R}_a$ (bottom right).

	$p = 2$					$p = 6$					$p = 10$				
		c					c					c			
		1.0		2.0			1.0		2.0			1.0		2.0	
	K	$\hat{K}$	$\mathcal{R}_a$	$\hat{K}$	$\mathcal{R}_a$	K	$\hat{K}$	$\mathcal{R}_a$	$\hat{K}$	$\mathcal{R}_a$	K	$\hat{K}$	$\mathcal{R}_a$	$\hat{K}$	$\mathcal{R}_a$
$n = 5,000$	3	3	0.949	3	0.997	3	3	0.946	3	0.992	3	3	0.926	3	0.992
		0	0.037	0	0.003		0	0.032	0	0.005		0	0.038	0	0.007
	6	6	0.928	6	0.997	6	6	0.784	6	0.993	8	8	0.769	8	0.954
		0	0.379	0	0.002		3	0.429	0	0.005		5	0.324	0	0.099
	12	12	0.820	13	0.959	12	12	0.897	12	0.988	20	20	0.729	20	0.979
		7	0.297	1	0.110		5	0.259	0	0.006		9	0.226	1	0.025
$n = 10,000$	3	3	0.956	3	0.998	3	3	0.955	3	0.995	3	3	0.923	3	0.994
		0	0.025	0	0.001		0	0.026	0	0.003		0	0.055	0	0.003
	6	6	0.960	6	0.998	6	6	0.786	6	0.995	8	8	0.761	8	0.989
		0	0.040	0	0.001		3	0.312	0	0.003		4	0.300	0	0.062
	12	12	0.882	13	0.952	12	12	0.896	12	0.993	20	21	0.729	20	0.991
		2	0.177	1	0.103		7	0.351	0	0.002		10	0.184	0	0.006
$n = 20,000$	3	3	0.941	3	0.998	3	3	0.959	3	0.994	3	3	0.924	3	0.995
		0	0.028	0	0.002		0	0.029	0	0.004		0	0.058	0	0.003
	6	6	0.950	6	0.998	6	6	0.809	6	0.996	8	9	0.779	8	0.994
		0	0.345	0	0.002		1	0.393	0	0.002		5	0.395	0	0.006
	12	12	0.912	13	0.945	12	12	0.951	12	0.996	20	22	0.773	20	0.992
		1	0.185	2	0.109		6	0.519	0	0.003		4	0.138	0	0.005

to handle large datasets which would otherwise be difficult to cluster using mixture modeling or other methods. So we evaluated performance with $(p, n, c) = \{2, 6, 10\} \times \{5000, 10000, 20000\} \times \{1, 2, 4\}$ with $K = \{3, 6, 12\}$ for $p = 2$ and 6 and $K = \{3, 8, 20\}$ for $p = 10$. For each case, we randomly generated 25 sets of parameters (cluster mean-directions and κ) according to the equal-proportioned K-Langevin-mixtures model, as above. In all, approximately 2,000 simulated datasets were used to assess the performance of k-mean-directions. With unknown K, we set K_T to 20 for the two- and six-dimensional experiments and $K_T = 40$ for $p = 10$. Further, for brevity, we report here only the summarized measures of performance over these 25 datasets for which K is unknown and also needs to be estimated, referring to the supplement for performance of k-mean-directions for when K is known, noting only that it is quite good in all scenarios even when separation of clusters is lower.

Table 1 summarizes the performance of our experiments for different settings and for K unknown. Note that K is almost always correctly estimated for $c = 4$ (and hence is not included in Table 1) as well as when $K = 3$ for both $c = 1$ and 2. For $c = 2$, K is estimated accurately in most cases with only a few instances where it overestimated the true number of clusters (eg., for $p = 2$ and $K = 12$ the median $\hat{K} = 13$). For these cases, $\mathcal{R}_a$ was slightly impacted, but overall performance was still very good. However, Table 1 indicates that for poor separation ($c = 1$) in higher dimensions, K is often incorrectly estimated (as noted by the larger interquartile ranges). Hence the resulting $\mathcal{R}_a$ will be negatively impacted, especially when K is underestimated. This is not surprising as the distinction between clusters becomes less apparent which makes estimating the number of clusters a difficult task. Overall, regardless of whether K is known or not, the results show that the proposed methodology is able to correctly identify groups quite well. Finally, we also note that the size of the datasets did not prove to be a limitation, showing that the computational efficiency of the k-mean-directions algorithm as developed comes in handy even in the context of large datasets.

3.3 Application to Very-high-dimensional Classification Dataset

Our final experiment was in identifying groups of documents in the well-known Classic3 dataset available online from `ftp://ftp.cs.cornell.edu/pub/smart`. This dataset is a well-known collection of 3,893 abstracts from CISI (1,460 articles on information retrieval), CRANFIELD (1,400 documents on aerodynamics), and MEDLINE (1,055 abstracts on medicine and related topics) databases consisting of over 24,000 unique words. Classic3 is often used to evaluate performance of text clustering/classification algorithms because it contains a known number of fairly well-separated groups (sources of abstracts). After processing the data to remove words appearing in less than 0.02% or more than 15% of the documents, 3,302 words remained and the resulting document vectors were each transformed to have unit L^2-norm. This processed dataset was used to evaluate performance of our k-mean-directions algorithm and our estimation method for the optimal number of clusters in that context. Since the dataset is severely high-dimensional, we initialized our k-mean-directions algorithm for each K at the best, in terms of lower value of (1), of the random starts method and hierarchical clustering with Ward's criterion for merging clusters. The optimal K was correctly estimated, using the method of Section 2.3, to be three. As seen in the confusion matrix summarizing the results in Table 2, the three clusters had about 99% of the documents correctly classified with only a few from each group being incorrectly clustered ($\mathcal{R}_a \approx 0.966$). Additionally, when compared to classifications derived from the mean-directions

Table 2: Confusion matrix for three clusters of Classic3. Columns represent the abstract origin and rows each represent one of the three identified clusters.

	CISI	CRANFIELD	MEDLINE
Cluster 1	1449	13	11
Cluster 2	8	2	1020
Cluster 3	3	1385	2

algorithm initialized from the true groupings and corresponding mean directions, 99.6% of the documents were correctly clustered ($\mathcal{R}_a \approx 0.989$). This indicates that the processed dataset has some minor overlap to its groups and shows that k-mean-directions, along with our methodology for choosing K, does an excellent job of grouping the documents when compared to what is ideally possible.

4. APPLICATION TO GENE EXPRESSION AND TEXT DATASETS

4.1 Mitotic Cell Division in Budding Yeast

The yeast cell cycle dataset *Saccharomyces cerevisia* (Cho et al. 1998) shows the expression levels over two cell cycles (or 17 time points) for 6,457 genes. These data contain the complete characterization of mRNA transcript levels during the mitotic cell division cycle which is comprised of several cell cycle phases including the phase where division does not occur (early and late G1, S and G2) as well as where mitotic (M) cell division occurs. Various subsets of this dataset have been analyzed to find groups of similarly-acting genes in several studies (Yeung, Fraley, Murua, Raftery and Ruzzo 2001; Banerjee et al. 2005; Dortet-Bernadet and Wicker 2008), but the entire dataset has itself never been completely analyzed. For example, Tavazoie, Hughes, Campbell, Cho and Church (1999) and Dortet-Bernadet and Wicker (2008) only consider the most variable 2,945 genes from 15 time points, while Yeung et al. (2001) and Banerjee et al. (2005) both separately consider even smaller subsets. Although each provide reasoning for analyzing only a subset (typically consisting of removing gene profiles believed to be "uninteresting" or those that have low expression/variability across all time points) it is unknown if there is any additional insight that would be provided if the entire set of genes could be analyzed. Our development of the k-mean-directions algorithm is to make clustering of huge datasets computationally practical and this dataset provides a natural scenario to apply our methodology.

As is typical in these studies, the original dataset was pre-processed before analysis. We first transformed each coordinate using quantile normalization (Boldstad, Irizarry, Astrand and Speed 2003) to reduce the differing variances and skewnesses in the gene profiles across all 17 time-points and then standardized each transformed gene expression profile to have zero-mean and unit L^2-norm. We then applied the k-mean-directions algorithm initialized using the best – in terms of lowest value of (1) – of the three strategies mentioned in Section 2.2 and used our methodology of Section 2.3 for estimating the number of clusters on the dataset to identify twenty-eight groups ranging in size from 171 to 315 profiles. The (standardized and transformed) cluster mean expression profiles, along with one standard error bars around the means for each of the 17 time-points are summarized in Figure 2. Since the results in Yeung et al. (2001) or Banerjee et al. (2005) are not presented in a form which readily permits comparisons with our groupings, we only compare our results here with those provided in Dortet-Bernadet and Wicker (2008), which estimated a similar number of clusters (twenty-six as opposed to the twenty-eight found here) in the reduced dataset using Akaike's Information Criterion (AIC). To facilitate easy cross-referencing with that article, we display our identified groups in Figure 2 in an order which provides the best visual match to that in Dortet-Bernadet and Wicker (2008).

Figure 2 shows that the resulting groups share many similarities with the results of Dortet-Bernadet and Wicker (2008) even though the expanded dataset was used. In particular, clusters numbered 1–6 have very similar mean profiles to the "cyclic gene clusters" of Dortet-Bernadet and Wicker (2008). These groups have two apparent yeast cycles in the mean expression profiles (time points 1–9 and 10–17). Clusters numbered 7–13 have similar trends to the "stress gene clusters" of Dortet-Bernadet and Wicker (2008), who named them as such due to the stress put on the cells from which they must recover. Many of the differences in our resulting clusters with those in Dortet-Bernadet and Wicker (2008) are in what the latter refer to as "miscellaneous gene clusters" as many have varying patterns that do not easily match up with the remaining clusters identified here (numbered 14–28). Further, several genes known to be involved in the cell cycle regulation in the G1 phase (CLN1, PCL1, NUD1 and SWE1) are all contained in cluster 2, while two others related to the G1 phase (CLB6 and SWI4) are in cluster 11. Additionally, cluster 1 contains other genes related to the M phase cell cycle regulation (CLB1 and CLB2). Similar results hold for the stress gene clusters as genes identified by Dortet-Bernadet and Wicker (2008) are identified in similar clusters here as well. Finally, we note that although we estimated a similar number of clusters as Dortet-Bernadet and Wicker (2008), we suspect that K is slightly overestimated as the observed c-separation between clusters numbered 13 and 15 was roughly 0.45 (which indicates very low separation). In addition to those two clusters, there were several other pairs with relatively low separation, including clusters 2 and 3 (with an observed c-separation of about 0.78).

In summary, it is apparent that k-mean-directions partitioned the entire dataset into clusters with similar interpretability as those obtained on a smaller dataset by Dortet-Bernadet and Wicker (2008). While the number and characteristics of groups identified are similar to those identified on the reduced dataset in that paper, we note that our results have been obtained using the entire dataset, without eliminating any genes, and the similarity of the identified clusters with Dortet-Bernadet and Wicker (2008) provides some assurance in our results. Further, by including the additional genes not used in the previous studies, further information is also provided into the functionality of those genes that have previously not been included in the cluster analysis.

4.2 Document Clustering

The Joint Statistical Meetings (JSM) is an annual international meeting of statisticians, with several thousand attendees and numerous technical sessions consisting of oral presentations. In 2008 for instance, there were 2,107 oral presentations made at the JSM in Denver, Colorado over four days. With such a large number of talks, it becomes necessary to have multiple sessions at one time: indeed, there were up to thirty-five sessions occurring at the same time at the 2008 JSM.

Oral presentations, ideally, should be scheduled in different sessions such that no two parallel sessions (happening at the same time) should be on the same kinds of topics. This would be of maximum benefit to attendees who are usually keen on attending presentations on similar topics of their specific interest, and would like related presentations to not take place in parallel. Typically each year, presenters submit their abstracts electronically at the beginning of February in the year of the JSM and request one or more sponsoring sections of the professional societies for placement of their presentations. Given the highly interdependent nature of the many aspects of the discipline of statistics, it is inconceivable that presentations can be neatly divided to disparate topics, if only done according to the sponsoring sections.

An alternative approach, which we propose and explore here is to use the text of the abstracts submitted by the presenters and to group them into as many clusters as the data may support. These clusters can then form the basis of parallel sessions. We study the potential for this automated approach using the methodology developed in this paper on the abstracts of the oral presentations at the 2008 JSM.

The collection of 2,107 abstracts consisting of 11,557 unique words forms our dataset. Words

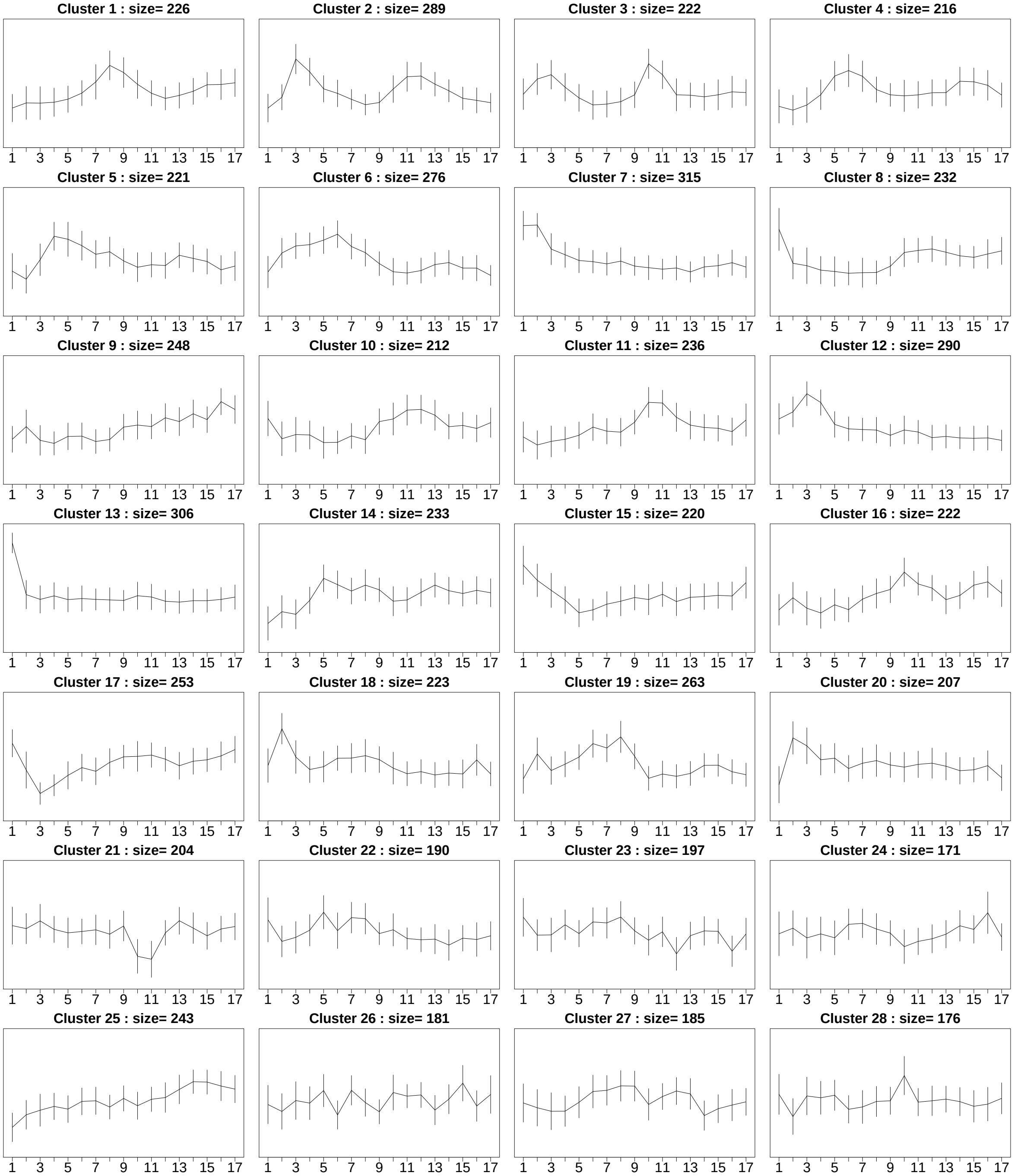


Figure 2: Standardized mean expression profiles (with standard error bars) of the 28 clusters found in the Yeast cell cycle data. Horizontal axis represents the time period of the experiment.

of very low and high frequencies were defined to be those appearing in less than four (0.02%) or more than 316 (about 15%) abstracts, respectively, and were removed from the lexicon using the MC toolkit of Dhillon and Modha (2001), leaving behind 3,762 remaining words. (Thus the dataset is 3,762-dimensional.) To emphasize words with low overall collection frequency, we used the commonly used *term frequency-inverse* weighting *i.e.*, the frequency of the jth term ($j = 1, \ldots, 3, 762$) in each document d_{ij} was weighted by $\log(2, 107/d_j)$, where d_j is the number of documents which contain the jth word. Each document vector was then normalized to be of unit length. The methodology from Section 2 was applied to the data initialized using the best of 1,000 random starts. The result was a total of 55 optimal clusters of presentations, ranging in size from 26 to 56 documents.

Figure 3 summarizes the cluster means of the five words with the highest weighted mean for each cluster. In all, there are 242 unique top-five words (each represented by a column in Figure 3) amongst the identified clusters with fifty-one of the top words being unique, while fifty-three of the second words differ from the top words. The third, fourth and fifth top words have 50, 43 and 45 unique words respectively. Thus, each cell in Figure 3 represents the weighted mean for one of the 242 words for a single cluster. The words are ordered such that the most frequent words in each cluster not already appearing in the display are grouped together and clusters are ordered according to their cardinality. From this, it is clear that the most of the groups were quite distinct as the "off-diagonal" cells in the plot displays very low intensities. Indeed, many of the clusters can be described by their top words. Using the frequency of occurrence of the top several words for each cluster, we were able to fairly uniquely identify each group of presentations. These identified groups of presentations, along with their cardinality, are represented in Figure 3.

Note that the groups identified very neatly fall into diverse sub-topics. The largest group of presentations have to do with the analysis of clinical trials, followed by those on variable selection, hypothesis testing, introductory statistics education, missing data and NHANES-related health surveys. At the other end, the smallest group of oral presentations was related to mortality studies, applications to economics and followed by financial statistics, applied stochastic processes, Bayesian Additive Regression Trees, Statistics in Sports and so on. A slightly more in-depth look at the top words in each cluster is available in Section S-2 of the supplementary materials, but we note that the resulting groups of presentations are both fairly distinct and interpretable.

The fifty-five clusters presented in Figure 3 indicate that there can be at the most fifty-five sessions in parallel for maximum benefit to attendees. The presentations in each group could

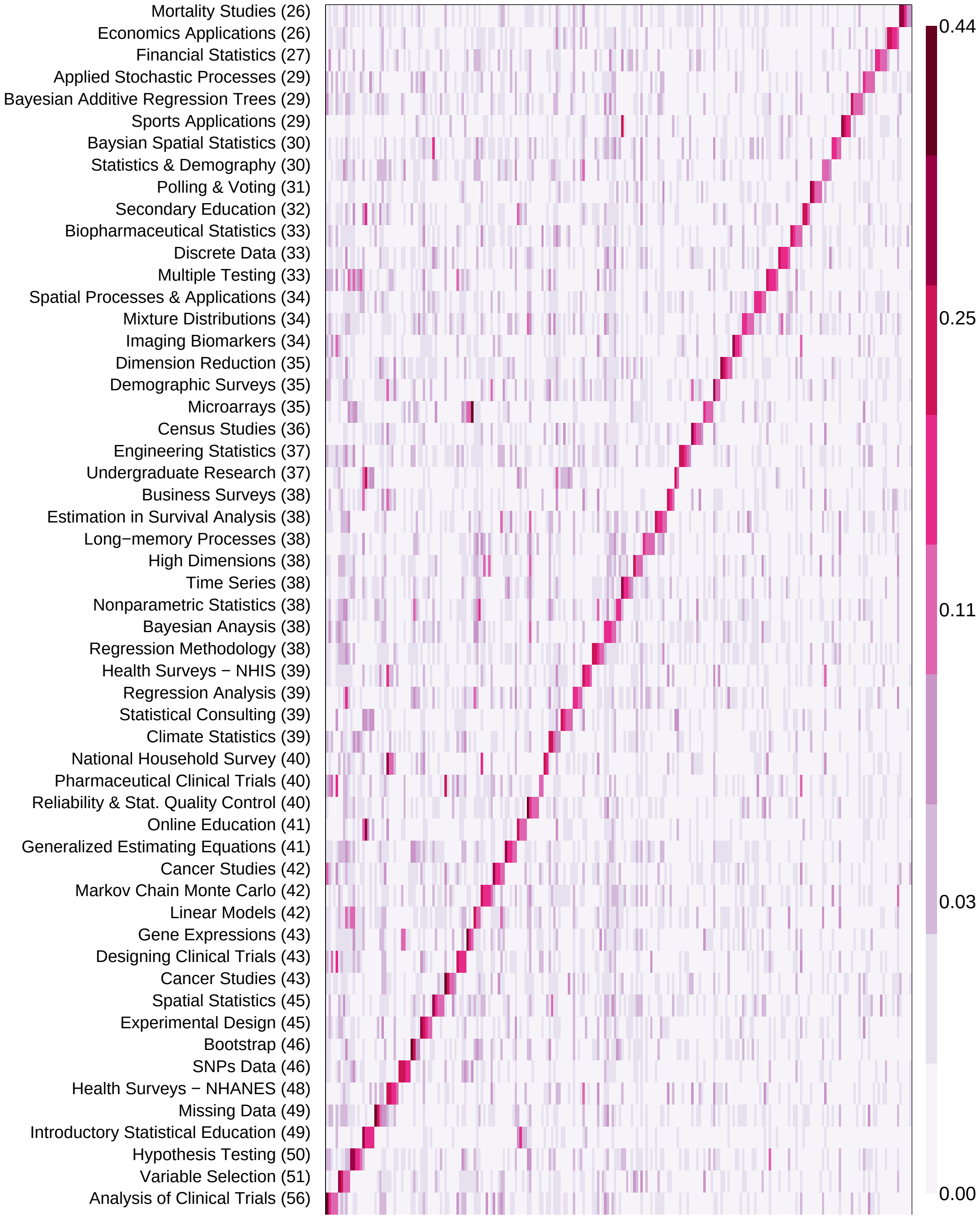


Figure 3: Means of the top five words for each of the 55 clusters in the 2008 JSM abstracts. Cluster labels (rows) identify the subject of the cluster with the number of abstracts in each cluster provided in parentheses. Each of the 242 terms are represented in the columns.

themselves be further grouped and made into more targeted serially-running sessions, taking care to ensure that these sub-grouped sessions never run in parallel with presentations in the other sub-categories of the same group. Further, note that our suggested finding of at most fifty-five concurrent sessions is many more than the up to thirty-five parallel sessions held in 2008, and suggests that the Meetings itself could have been shortened a little in duration to save costs for both organizing societies and attendees.

5. DISCUSSION

The main contribution of this paper is the development of a modification of the k-means algorithm of Hartigan and Wong (1979) for data constrained to lie on the circumference of a sphere. The suggested modified methodology maintains the frugality in computations that is the hallmark of its Hartigan and Wong (1979) k-means cousin and is able to take very large, severely high-dimensional datasets. Our algorithm is also an iterative scheme which requires initialization, for which we provide both a deterministic and a stochastic approach, and recommend proceeding with the solution that is best as the candidate initializers. We also provide a criterion for estimating the number of clusters that is based on the largest relative change in the locally optimized objective function. While ISO/ANSI compliant C software implementing k-mean directions and R code for all simulation datasets used in this paper are available from the supplementary materials section of the journal website or upon request, an R package is under development for public release. Further, while the methodology was developed in the context of k-mean-directions, both the initialization procedure and the criterion for estimating the number of clusters are general enough to extend to other clustering algorithms for directional data. Results on the simulated datasets are very promising. We have applied our methodology to analyze microarray gene expression data on the cell cycle of yeast as well as the collection of abstracts from the 2008 Joint Statistical Meetings. In the first, the goal was to compare results based on the entire dataset to those obtained in previous studies based on only a subset of the data. The resulting groups were very similar in interpretability to those of Dortet-Bernadet and Wicker (2008) but provide additional information on the yeast cell division process as the entire set of genes could be used in the analysis. The results arrived at in the JSM abstracts dataset consist of numerous interpretable groups that may provide help with the issue of assigning presentations to sessions at the conference.

A few points remain to be addressed. The first is to modify the algorithm to account for noise or scattered observations. One suggestion would be to develop an algorithm similar to the k-clips algorithm of Maitra and Ramler (2008) that accounts for such observations and is still computationally feasible. Certain aspects of the algorithm may be readily adapted, but determining how the algorithm defines scatter would need consideration. Another issue lies in clustering massive directional datasets; in this scenario, it may be possible to modify the multi-stage clustering approach of Maitra (2001). Thus while the methodology developed in this paper is an important tool contributing to clustering spherically constrained datasets, further issues remain that require additional attention.

ACKNOWLEDGMENTS

The authors thank Ron Wasserstein, Executive Director of the American Statistical Association, for providing the abstracts from the 2008 Joint Statistical Meetings. We thank the editor, an associate editor and three reviewers whose helpful suggestions and insightful comments greatly improved the quality of this manuscrpt.

SUPPLEMENTARY MATERIALS

Additional Investigations: Sections S-1–2 along with Figures S-1 and Tables S-1–6 are in the supplementary materials archive of the journal website.

Software: C code implementing this algorithm is available in the supplementary materials archive of the journal website.

Datasets: Datasets analyzed in this paper are provided in the supplementary materials archive of the journal website.

REFERENCES

Akaike, H. (1973), “Information theory and an extension of the maximum likelihood principle,” *Second international symposium on information theory*, pp. 267–281.

Akaike, H. (1974), “A new look at the statistical model identification,” *IEEE Transactions on Automatic Control*, 19, 716–723.

Banerjee, A., Dhillon, I. S., Ghosh, J., and Sra, S. (2005), “Clustering on the Unit Hypesphere using von Mises-Fisher Distributions,” *Journal of Machine Learning Research*, 6, 1345–1382.

Barrett, T., Suzek, T. O., Troup, D. B., Wilhite, S. E., Ngau, W. C., Ledoux, P., Rudnev, D., Lash, A., Fujibuchi, W., and Edgar, R. (2005), “NCBI GEO: mining millions of expression profiles–database and tools.,” *Nucleic Acids Res*, 33, D562–D566.

Boldstad, B. M., Irizarry, R. A., Astrand, M., and Speed, T. P. (2003), “A Comparison of Normalization Methods for High Density Oligonucleotide Array Data Based on Bias and Variance,” *Bioinformatics*, 19(2), 185–193.

Celeux, G., and Govaert (1995), “Gaussian parsimonious clustering models,” *Computational Statistics and Data Analysis*, 28, 781–93.

Cho, R. J., Campbell, M. J., Winzeler, E. A., Steinmetz, L., Conway, A., Widicka, L., Wolfsberg, T. G., Gabrielian, A. E., Landsman, D., Lockhart, D. J., and Davis, R. W. (1998), “A genome-wide transcriptional analysis of the mitotic cell,” *Molecular Cell*, 2, 65–73.

Dasgupta, S. (1999), Learning mixtures of Gaussians,, in *Proc. IEEE Symposium on Foundations of Computer Science*, New York, pp. 633–644.

Dhillon, I. S., and Modha, D. S. (2001), “Concept Decompositions for Large Sparse Text Data Using Clustering,” *Machine Learning*, 42, 143–175.

Dortet-Bernadet, J., and Wicker, N. (2008), “Model-based clustering on the unit sphere with an illustration using gene expression profiles,” *Biostatistics*, 9(1), 66–80.

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*, 95, 14863–14868.

Everitt, B. S., Landau, S., and Leesem, M. (2001), *Cluster Analysis (4th ed.)*, New York: Hodder Arnold.

Frakes, W. B., and Baeza-Yates, R. (1992), *Information Retrieval: Data Structures and Algorithms*, Englewood Cliffs, New Jersey: Prentice Hall.

Fraley, C., and Raftery, A. E. (1998), “How many clusters? Which cluster method? Answers via model-based cluster analysis,” *Computer Journal*, 41, 578–588.

Fraley, C., and Raftery, A. E. (2002), “Model-Based Clustering, Discriminant Analysis, and Density Estimation,” *Journal of the American Statistical Association*, 97, 611–631.

Giacomelli, L., and Nicolini, C. (2006), “Gene expression of human T lymphocytes cell cycle: Experimental and bioinformatic analysis,” *Cellular Biochemistry*, 99(5), 1326–1333.

Hartigan, J. (1985), “Statistical theory in clustering,” *Journal of Classification*, 2, 63–76.

Hartigan, J. A., and Wong, M. A. (1979), “A K-means clustering algorithm,” *Applied Statistics*, 28, 100–108.

Hubert, L., and Arabie, P. (1985), “Comparing partitions,” *Journal of Classification*, 2, 193–218.

Kaufman, L., and Rousseuw, P. J. (1990), *Finding Groups in Data*, New York: John Wiley and Sons, Inc.

Kettenring, J. R. (2006), “The practice of cluster analysis,” *Journal of classification*, 23, 3–30.

Kolda, T. G. (1997), Limited-Memory Matrix Methods with Applications, PhD thesis, The Applied Mathematics Program, University of Maryland, College Park, Maryland.

Lund, U., and Agostinelli, C. (2007), *circular: Circular Statistics*. R package version 0.3-8.

MacQueen, J. (1967), Some Methods for Classification and Analysis of Multivariate Observations,, in *Proc. Fifth Berkeley Symp. Math. Statistics and Probability*, Vol. 2, pp. 281–297.

Maitra, R. (2001), “Clustering massive datasets with applications to software metrics and tomography,” *Technometrics*, 43(3), 336–346.

Maitra, R. (2009), “Initializing Partition-Optimization Algorithms,” *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 6(1), 144–157.

Maitra, R., and Ramler, I. P. (2008), “Clustering in the Presence of Scatter,” *Biometrics*, preprint, 5 June 2008. doi:10.1111/j.1541-0420.2008.01064.x.

Mardia, K. V., and Jupp, P. E. (2000), *Directional Statistics*, New York: Wiley.

Murtagh, F. (1985), *Multi-dimensional clustering algorithms*, Berlin; New York: Springer-Verlag.

Ramey, D. B. (1985), “Nonparametric clustering techniques,” in *Encyclopedia of Statistical Science*, Vol. 6, New York: Wiley, pp. 318–319.

Salton, G., and Buckley, C. (1988), “Term-weigting approaches in automatic text retrieval,” *Ing. Process. Manage.*, 27(5), 513–523.

Salton, G., and McGill, M. J. (1983), *Introduction to modern retrieval*, New York: McGraw-Hill Book Company.

Schwarz, G. (1978), “Estimating the dimensions of a model,” *Annals of Statistics*, 6, 461–464.

Singhal, A., Buckley, C., Mitra, M., and Salton, G. (1996), “Pivoted document length normalization,” *ACM SIGIR’96*, pp. 21–29.

Tavazoie, S., Hughes, J. D., Campbell, M. J., Cho, R. J., and Church, G. M. (1999), “Systematic determination of genetic networks architecture,” *Nature Genetics*, 22, 281–285.

Tibshirani, R. J., Walther, G., and Hastie, T. J. (2003), “Estimating the number of clusters in a dataset via the gap statistic,” *Journal of the Royal Statistical Society*, 63(2), 411–423.

Ward, J. J. H. (1963), “Hierarchical Grouping to Optimize an Objective Function,” *Journal of the American Statistical Association*, 58, 236–244.

Watson, G. S. (1983), “Statistics on Spheres,” *University of Arkansas Lecture Notes in the Mathematical Sciences*, 6.

Whitfield, M. L., Sherlock, G., Saldanha, A. J., Murray, J. I., Ball, C. A., Alexander, K. E., Matese, J. C., Perou, C. M., Hurt, M. M., Brown, P. O., and Botstein, D. (2002), “Identification of Genes Periodically Expressed in the Human Cell Cycle and Their Expression in Tumors,” *Mol Biol Cell*, 13(6), 1977–2000.

Yeung, K. Y., Fraley, C., Murua, A., Raftery, A. E., and Ruzzo, W. L. (2001), “Modelbased clustering and data transformation for gene expression data,” *Bioinformatics*, 17, 977–987.

Zeeman, S. C., Tiessen, A., Pilling, E., Kato, K. L., Donald, A. M., and Smith, A. M. (2002), “Starch synthesis in Arabidopsis. Granule synthesis, composition, and structure,” *Plant Physiology*, 129, 516–529.

APPENDIX A. RELATIONSHIP BETWEEN CORRELATION SIMILARITY AND SQUARED EUCLIDEAN DISTANCE BETWEEN STANDARDIZED PROFILES

Let $\boldsymbol{W} = (W_1, W_2, \ldots, W_p)$ and $\boldsymbol{Z} = (Z_1, Z_2, \ldots, Z_p)$ be two p-dimensional gene expression profiles with corresponding means $\bar{W} = p^{-1}\sum_{i=1}^{p} W_i$, $\bar{Z} = p^{-1}\sum_{i=1}^{p} Z_i$ and variances $s_W^2 = p^{-1}\sum_{i=1}^{p}(W_i - \bar{W})^2$ and $s_Z^2 = p^{-1}\sum_{i=1}^{p}(Z_i - \bar{Z})^2$ respectively. Define the standardized profiles to be $\boldsymbol{U} = (U_1, U_2, \ldots, U_p)$ and $\boldsymbol{V} = (V_1, V_2, \ldots, V_p)$ where $U_i = (W_i - \bar{W})/s_W$ and $V_i = (Z_i - \bar{Z})/s_Z$ for $i = 1, 2, \ldots, p$. (Note that $p^{-\frac{1}{2}}\boldsymbol{U}$ and $p^{-\frac{1}{2}}\boldsymbol{V} \in \mathcal{S}^p_{\perp\mathbf{1}}$.)) Then the squared Euclidean distance $d^2(\boldsymbol{U}, \boldsymbol{V})$ between $\boldsymbol{U}$ and $\boldsymbol{V}$ is given by $d^2(\boldsymbol{U}, \boldsymbol{V}) = (\boldsymbol{U} - \boldsymbol{V})'(\boldsymbol{U} - \boldsymbol{V}) = 2 - 2\rho_{\boldsymbol{W},\boldsymbol{Z}}$ where $\rho_{\boldsymbol{W},\boldsymbol{Z}}$ is the correlation between $\boldsymbol{W}$ and $\boldsymbol{Z}$. Thus the squared Euclidean distance between the standardized profiles $\boldsymbol{U}$ and $\boldsymbol{V}$ is an affine transformation of the correlation between the untransformed gene expression profiles $\boldsymbol{W}$ and $\boldsymbol{Z}$.