Shrinkage Clustering: A Fast and Size-Constrained Algorithm for Biomedical Applications*

Chenyue W. Hu[1], Hanyang Li[2], and Amina A. Qutub[3]

1   Department of Bioengineering, Rice University, Houston, TX, USA
2   Department of Bioengineering, Rice University, Houston, TX, USA
3   Department of Bioengineering, Rice University, Houston, TX, USA
    aminaq@rice.edu

──── Abstract ────

Motivation: Many common clustering algorithms require a two-step process that limits their efficiency. The algorithms need to be performed repetitively and need to be implemented together with a model selection criterion, in order to determine both the number of clusters present in the data and the corresponding cluster memberships. As biomedical datasets increase in size and prevalence, there is a growing need for new methods that are more convenient to implement and are more computationally efficient. In addition, it is often essential to obtain clusters of sufficient sample size to make the clustering result meaningful and interpretable for subsequent analysis.

Results: We introduce *Shrinkage Clustering*, a novel clustering algorithm based on matrix factorization that simultaneously finds the optimal number of clusters while partitioning the data. We report its performances across multiple simulated and actual datasets, and demonstrate its strength in accuracy and speed in application to subtyping cancer and brain tissues. In addition, the algorithm offers a straightforward solution to clustering with cluster size constraints. Given its ease of implementation, computing efficiency and extensible structure, we believe *Shrinkage Clustering* can be applied broadly to solve biomedical clustering tasks especially when dealing with large datasets.

1998 ACM Subject Classification I.5.3 Clustering

Keywords and phrases Clustering, Matrix Factorization, Cancer Subtyping, Gene Expression

Digital Object Identifier 10.4230/LIPIcs.WABI.2017.11

1   Introduction

Cluster analysis is one of the most frequently used unsupervised machine learning methods in biomedicine. The task of clustering is to automatically uncover the natural groupings of a set of objects based on some known similarity relationships. Often employed as a first step in a series of biomedical data analyses, cluster analysis helps to identify distinct patterns in data and suggest classification of objects (e.g. genes, cells, tissue samples, patients) that are functionally similar or related. Typical applications of clustering include subtyping cancer based on gene expression levels [31], classifying protein subfamilies based on sequence similarities [5], distinguishing cell phenotypes based on morphological imaging metrics [26], and identifying disease phenotypes based on physiological and clinical information [22].

─────────────

17th International Workshop on Algorithms in Bioinformatics (WABI 2017).
Editors: Russell Schwartz and Knut Reinert; Article No. 11; pp. 11:1–11:13

Many algorithms have been developed over the years for cluster analysis [32], including hierarchical approaches [11] (e.g., *ward-linkage*, *single-linkage*) and partitional approaches that are centroid-based (e.g., *K-means* [15]), density-based (e.g., DBSCAN [7]), distribution-based (e.g., Gaussian mixture models [19]), or graph-based (e.g., Normalized Cut [25]). Notably, nonnegative matrix factorization (NMF) has received a lot of attention in application to cluster analysis, because of its ability to solve challenging pattern recognition problems and the flexibility of its framework [14]. NMF-based methods have been shown to be equivalent to a relaxed *K-means* clustering and Normalized Cut spectral clustering with particular cost functions [6], and NMF-based algorithms have been successfully applied to clustering biomedical data [4].

With few exceptions, most clustering algorithms group objects into a pre-determined number of clusters, and do not inherently look for the number of clusters in the data. Therefore, cluster evaluation measures are often employed and are coupled with clustering algorithms to select the optimal clustering solution from a series of solutions with varied cluster numbers. Commonly used model selection methods for clustering, which vary in cluster quality assessment criteria and sampling procedures, include *Silhouette* [24], *X-means* [23], *Gap Statistic* [29], *Consensus Clustering* [20], *Stability Selection* [13], and *Progeny Clustering* [10]. The drawbacks of coupling cluster evaluation with clustering algorithms include (i) computation burden, since the clustering needs to be performed with various cluster numbers and sometimes multiple times to assess the solution's stability; and (ii) implementation burden, since the integration can be laborious if algorithms are programmed in different languages or are available on different platforms.

Here, we propose a novel clustering algorithm *Shrinkage Clustering* based on symmetric nonnegative matrix factorization notions [12]. Specifically, we utilize unique properties of a hard clustering assignment matrix to simplify the matrix factorization problem and to design a fast algorithm that accomplishes the two tasks of determining the optimal cluster number and performing clustering in one. The *Shrinkage Clustering* algorithm is mathematically straightforward, computationally efficient, and structurally flexible. In addition, the flexible framework of the algorithm allows us to extend it to clustering applications with minimum cluster size constraints.

2   Methods

2.1   Problem Formulation

Let $X = \{X_1, \ldots, X_N\}$ be a finite set of N objects. The task of cluster analysis is to group objects that are similar to each other and separate those that are dissimilar to each other. The completion of a clustering task can be broken down to two steps: (i) deriving similarity relationships among all objects (e.g., Euclidean distance); (ii) clustering objects based on these relationships. The first step is sometimes omitted when the similarity relationships are directly provided as raw data, for example in the case of clustering genes based on their sequence similarities. Here, we assume that the similarity relationships were already derived and are available in the form of a similarity matrix $S_{N \times N}$, where $S_{ij} \in [0, 1]$ and $S_{ij} = S_{ji}$. In the similarity matrix, a larger S_{ij} represents more resemblance in pattern or closer proximity in space between X_i and X_j, and vice versa.

Suppose $A_{N \times K}$ is a clustering solution for objects with similarity relationships $S_{N \times N}$. Since we are only considering the case of hard clustering, we have $A_{ik} \in \{0, 1\}$ and $\sum_{k=1}^{K} A_{ik} = 1$. Specifically, K is the number of clusters obtained, and A_{ik} takes the value of 1 if X_i belongs to cluster k and takes the value of 0 if it does not. The product of A and its transpose

A^T represents a solution-based similarity relationship $\hat{S}$ (i.e. $\hat{S} = AA^T$), in which $\hat{S}_{ij}$ takes the value of 1 when X_i and X_j are in the same cluster and 0 otherwise. Unlike S_{ij} which can take continuous values between 0 and 1, $\hat{S}_{ij}$ is a binary representation of the similarity relationships indicated by the clustering solution. If a clustering solution is optimal, the solution-based similarity matrix $\hat{S}$ should be similar to the original similarity matrix S if not equal.

Based on this intuition, we formulate the clustering task mathematically as

$$
\min_A \qquad \|S - AA^T\|_F
$$
$$
\text{subject to} \quad A_{ik} \in \{0, 1\}, \quad \sum_{k=1}^{K} A_{ik} = 1, \quad \sum_{i=1}^{N} A_{ik} \neq 0 \ . \tag{1}
$$

The goal of clustering is therefore to find an optimal cluster assignment matrix A, which represents similarity relationships that best approximate the similarity matrix S derived from the data. The task of clustering is transformed into a matrix factorization problem, which can be readily solved by existing algorithms. However, most matrix factorization algorithms are generic (not tailored to solving special cases like Function 1), and are therefore computationally expensive.

2.2 Properties and Rationale

In this section, we explore some special properties of the objective Function 1 that lay the ground for *Shrinkage Clustering*. Unlike traditional matrix factorization problems, the solution A we are trying to obtain has special properties, i.e. $A_{ik} \in \{0, 1\}$ and $\sum_{k=1}^{K} A_{ik} = 1$. This binary property of A greatly simplifies the objective Function 1 as below.

$$
\min_A \|S - AA^T\|_F
$$
$$
= \min_A \sum_{i=1}^{N} \sum_{j=1}^{N} (S_{ij} - A_i \bullet A_j)^2
$$
$$
= \min_A \sum_{i=1}^{N} \Big(\sum_{j \in \{j | A_i = A_j\}} (S_{ij} - 1)^2 + \sum_{j \in \{j | A_i \neq A_j\}} S_{ij}^2 \Big)
$$
$$
= \min_A \sum_{i=1}^{N} \sum_{j \in \{j | A_i = A_j\}} (1 - 2S_{ij}) + \sum_{i=1}^{N} \sum_{j=1}^{N} S_{ij}^2
$$

Here, A_i represents the ith row of A, and the symbol $\bullet$ denotes the inner product of two vectors. Note that $A_i \bullet A_j$ take binary values of either 0 or 1, because $A_{ik} \in \{0, 1\}$ and $\sum_{k=1}^{K} A_{ik} = 1$. In addition, $\sum_{i=1}^{N} \sum_{j=1}^{N} S_{ij}^2$ is a constant that does not depend on the clustering solution A. Based on this simplification, we can reformulate the clustering problem as

$$
\min_A f(A) = \sum_{i=1}^{N} \sum_{j \in \{j | A_i = A_j\}} (1 - 2S_{ij}). \tag{2}
$$

Let's now consider how the value of the objective Function 2 changes when we change the cluster membership of an object X_i. Suppose we start with a clustering solution A, in which X_i belongs to cluster k ($A_{ik} = 1$). When we change the cluster membership of X_i from k to k' with the rest remaining the same, we would obtain a new clustering solution A',

in which $A'_{ik'} = 1$ and $A'_{ik} = 0$. Since S is symmetric (i.e. $S_{ij} = S_{ji}$), the value change of the objective Function 2 is

$$
\begin{aligned}
\triangle f_i &:= f(A') - f(A) \\
&= \sum_{j \in k'}(1 - 2S_{ij}) - \sum_{j \in k}(1 - 2S_{ij}) + \sum_{j \in k'}(1 - 2S_{ji}) - \sum_{j \in k}(1 - 2S_{ji}) \\
&= 2(\sum_{j \in k'}(1 - 2S_{ij}) - \sum_{j \in k}(1 - 2S_{ij})) \quad .
\end{aligned}
\tag{3}
$$

2.3   Shrinkage Clustering: Base Algorithm

Based on the simplified objective Function 2 and its properties with cluster changes (Function 3), we designed a greedy algorithm *Shrinkage Clustering* to rapidly look for a clustering solution A that factorizes a given similarity matrix S. As described in Algorithm 1, *Shrinkage Clustering* begins by randomly assigning objects to a sufficiently large number of initial clusters. During each iteration, the algorithm first removes any empty clusters generated from the previous iteration, a step that gradually shrinks the number of clusters; then it permutes the cluster membership of the object that most minimizes the objective function. The algorithm stops when the solution converges (i.e. no cluster membership permutation can further minimize the objective function), or when a pre-specified maximum number of iterations is reached. *Shrinkage Clustering* is guaranteed to converge to a local optimum as shown in Theorem 1. The main and advantageous feature of *Shrinkage Clustering* is that it shrinks the number of clusters while finding the clustering solution. During the process of permuting cluster memberships to minimize the objective function, clusters automatically collapse and become empty until the optimization process is stabilized and the optimal cluster memberships are found. The number of clusters remaining in the end is the optimal number of clusters, since it stabilizes the final solution. Therefore, *Shrinkage Clustering* achieves both tasks of (i) finding the optimal number of clusters and (ii) finding the clustering memberships.

▶ **Theorem 1.** *Shrinkage Clustering monotonically converges to a (local) optimum.*

Proof. We first demonstrate the monotonically decreasing property of the objective Function 2 in each iteration of the algorithm. There are two steps taken in each iteration: (i) removal of empty clusters; and (ii) permutation of cluster memberships. Step (i) does not change the value of the objective function, because the objective function only depends on non-empty clusters. On the other hand, step (ii) always lowers the objective function, since a cluster membership permutation is chosen based on its ability to achieve the greatest minimization of the objective function. Combing step (i) and (ii), it is obvious that the value of the objective function monotonically decreases with each iteration. Since $\|S - AA^T\|_F \geq 0$ and $\|S - AA^T\|_F = \sum_{i=1}^{N} \sum_{j \in \{j | A_i = A_j\}}(1 - 2S_{ij}) + \sum_{i=1}^{N} \sum_{j=1}^{N} S_{ij}^2$, the objective function has a lower bound of $-\sum_{i=1}^{N} \sum_{j=1}^{N} S_{ij}^2$. Therefore, a convergence to a (local) optimum is guaranteed, because the algorithm is monotonically decreasing with a lower bound.   ◀

2.4   Shrinkage Clustering with Cluster Size Constraints

It is well-known that *K-means* can generate empty clusters when clustering high-dimensional data with over 20 clusters, and *Hierarchical Clustering* often generate tiny clusters with few samples. In practice, clusters of too small a size can sometimes be full of outliers, and they are often not preferred in cluster interpretation since most statistical tests do not apply to

---

Algorithm 1 *Shrinkage Clustering: Base Algorithm*

Input: $S_{N \times N}$ (similarity matrix)

   K_0 (intial number of clusters)

Initialization:

a. Generate a random $A_{N \times K_0}$ (cluster assignment matrix)

b. Compute $\tilde{S} = 1 - 2S$

repeat

   1. Remove empty clusters:

      (a) Delete empty columns in A (i.e. $\{j | \sum_{i=1}^{N} A_{ij} = 0\}$)

   2. Permute the cluster membership that minimizes Function 2 the most:

      (a) Compute $M = \tilde{S}A$

      (b) Compute v by $v_i = \min_j M_{ij} - \sum_{j=1}^{K}(M \circ A)_{ij}$, where

         $\circ$ represents the element-wise product (Hadamard product)

      (c) Find the object $\bar{X}$ with the greatest optimization potential,

         i.e. $\bar{X} = \arg\min_i v_i$

      (d) Permute the membership of $\bar{X}$ to C', where $C' = \arg\min_j M_{\bar{X}j}$

until $\sum_{i=1}^{N} v_i = 0$ **or** reaching max number of iterations

Output: A (cluster assignment)

Algorithm 2 *Shrinkage Clustering with Cluster Size Constraints*

Additional Input: ω (minimum cluster size).

Updated Step 1:

   (a) Remove columns in A that contain too few objects, i.e. $\{j | \sum_{i=1}^{N} A_{ij} < \omega\}$

   (b) Reassign objects in these clusters to clusters with the greatest minimization

---

small sample sizes. Though extensions to *K-means* were proposed to solve this issue [3], the attempt to control cluster sizes has not been easy. In contrast, the flexibility and the structure of *Shrinkage Clustering* offers a straightforward and rapid solution to enforcing constraints on cluster sizes. To generate a clustering solution with each cluster containing at least ω objects, we can simply modify Step 1 of the iteration loop in Algorithm 1. Instead of removing empty clusters in the beginning of each iteration, we now remove clusters of sizes smaller than a pre-specified size ω. The base algorithm (Algorithm 1) can be viewed as a special case of $w = 0$ in the size-constrained *Shrinkage Clustering* algorithm.

3 Results

3.1 Experiments on Similarity Data

3.1.1 Testing with Simulated Similarity Matrices

We first use simulated similarity matrices to test the performance of *Shrinkage Clustering* and to examine its sensitivity to the initial parameters and noise. As a proof of concept, we generate a similarity matrix S directly from a known cluster assignment matrix A by $S = AA^T$. Here, the cluster assignment matrix $A_{100 \times 5}$ is randomly generated to consist of 100 objects grouped into 5 clusters with unequal cluster sizes (i.e. 15, 17, 20, 24 and 24 respectively). The similarity matrix $S_{100 \times 100}$ generated from the product of A and A^T therefore represents an ideal case, where there is no noise, since each entry of S only takes a binary value of either 0 or 1.

■ **Table 1** Clustering results of simulated similarity matrices with varying size constraints (ω), where C is the cluster generated by *Shrinkage Clustering*.

| True Label	$\omega = 0$					$\omega = 20$				$\omega = 25$	
	C1	C2	C3	C4	C5	C1	C2	C3	C4	C1	C2
Cluster 1	0	0	24	0	0	0	24	0	0	0	24
Cluster 2	15	0	0	0	0	15	0	0	0	15	0
Cluster 3	0	0	0	24	0	0	0	24	0	0	24
Cluster 4	0	17	0	0	0	17	0	0	0	17	0
Cluster 5	0	0	0	0	20	0	0	0	20	20	0

We apply *Shrinkage Clustering* to this simulated similarity matrix S with 20 initial random clusters and repeat the algorithm for 1000 times. Each run, the algorithm accurately generates 5 clusters with cluster assignments $\tilde{A}$ in perfect match with the true cluster assignments A (an example shown in Table 1 under $\omega = 0$), demonstrating the algorithm's ability to perfectly recover the cluster assignments in a non-noisy scenario. The shrinkage paths of the first 5 runs (Figure 1A) illustrate that most runs start around a number of 20 clusters, and all of them shrink down gradually to a final number of 5 clusters when the solution reaches an optimum. We then test whether the algorithm is sensitive to the initial number of clusters (K_0) by running it with K_0 ranging from 5 (true number of clusters) to 100 (maximum number of clusters). In each case, the true cluster structure is recovered perfectly, demonstrating the robustness of the algorithm to different initial cluster numbers. The shrinkage paths in Figure 1B clearly show that in spite of starting with various initial numbers of clusters, all paths converge to the same number of clusters at the end.

Next, we investigate the effects of size constraints on *Shrinkage Clustering*'s performance by varying ω from 1 to 5, 10, 20 and 25. The algorithm is repeated 50 times in each case. We find that as long as ω is smaller than the true minimum cluster size (i.e. 15), the size constrained algorithm can perfectly recover the true cluster assignments A in the same way as the base algorithm. Once ω exceeds the true minimum cluster size, clusters are forced to merge and therefore result in a smaller number of clusters (example clustering solutions of $\omega = 20$ and $\omega = 25$ shown in Table 1). In these cases, it is impossible to find the true cluster structure because the algorithm starts off with fewer clusters than the true number of clusters and it works uni-directionally (i.e. only shrinks). Besides enabling supervision on the cluster sizes, size-constrained *Shrinkage Clustering* is also computationally advantageous. Figure 1C shows that a larger ω results in fewer iterations needed for the algorithm to converge, and the effect reaches a plateau once ω reaches certain sizes (e.g. $\omega = 10$ in this case). The shrinkage paths (Figure 1D) show that it is the reduced number of iterations at the beginning of a run that speeds up the entire process of solution finding when ω is large.

In reality, it is rare to find a perfectly binary similarity matrix similar to what we generated from a known cluster assignment matrix. There is always a certain degree of noise clouding our observations. To investigate how much noise the algorithm can tolerate in the data, we add a layer of Gaussian noise over the simulated similarity matrix. Since $S_{ij} \in \{0, 1\}$, we create a new similarity matrix S^N containing noise defined by

$$S^N = \{ \begin{array}{ll} |\varepsilon| & \text{if } S_{ij} = 0 \\ 1 - |\varepsilon| & \text{if } S_{ij} = 1 \end{array},$$

where $\varepsilon \sim N(0, \sigma^2)$. Figure 1E illustrates the changes of the similarity distribution density as the standard deviation σ varies from 0 to 0.5. When $\sigma = 0$ (i.e. no noise), S^N is
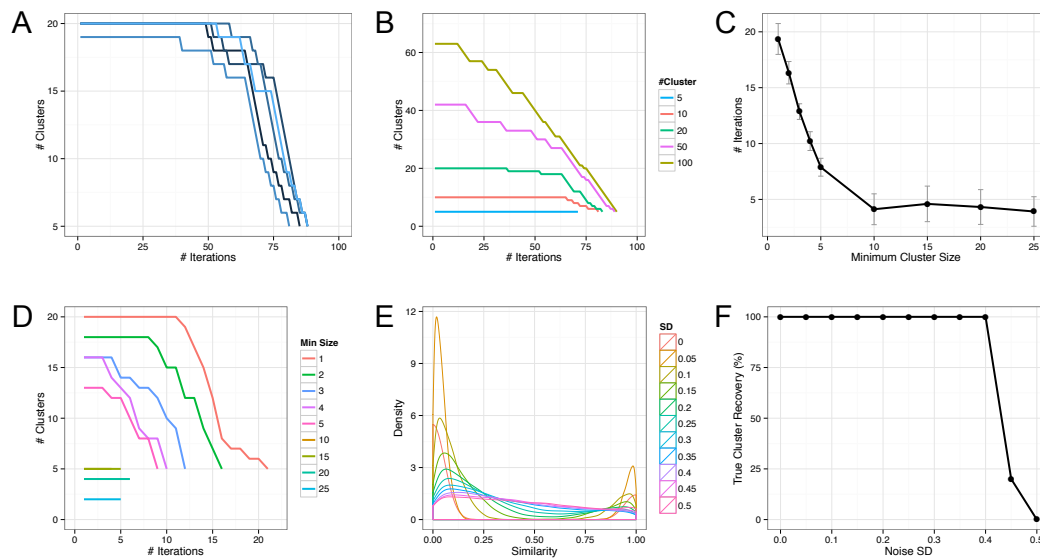
Figure 1 Testing results on simulated similarity data: (A) The first 5 shrinkage paths from running base algorithm with different initiations. (B) Example shrinkage paths with varying initial cluster numbers of 5, 10, 20, 50 and 100. (C) The average number of iterations spent with ω taking values of 1 to 5, 10, 15, 20 and 25. (D) Example shrinkage paths for ω of 1 to 5, 10, 15, 20 and 25 (path of $\omega = 10$ is in overlap with $\omega = 15$). (E) The similarity distribution for ε from 0 to 0.5. (F) The probability of successfully recovering the underlying cluster structure against different noise levels.

Bernoulli distributed. As σ becomes larger and larger, the bimodal shape is flattened by noise. When $\sigma = 0.5$, approximately 32% of the similarity relationships are reversed, and hence observations have been perturbed too much to infer the underlying cluster structure. The performances of *Shrinkage Clustering* in these noisy conditions are shown in Figure 1F. The algorithm proves to be quite robust against noise, as the true cluster structure is 100% recovered in all conditions except for when $\sigma > 0.4$.

3.1.2   Biological Case Study: TCGA Dataset

To illustrate the performance of *Shrinkage Clustering* on real biological similarity data, we apply the algorithm to subtyping tumors from the Cancer Genome Atlas (TCGA) dataset [27]. Derived from the TCGA database, the dataset includes 293 samples from 3 types of cancers, which are Breast Invasive Carcinoma (BRCA, 207 samples), Glioblastoma Multiforme (GBM, 67 samples) and Lung Squamous Cell Carcinoma (LUSC, 19 samples). The data is presented in the form of a similarity matrix, which integrates information from the gene expression levels, DNA methylation and copy number aberration. Since the similarity scores from the TCGA dataset are in general skew to 1, we first normalize the data by shifting its mean around 0.5 and by bounding values that are greater than 1 and smaller than 0 to 1 and 0 respectively. We then perform *Shrinkage Clustering* to cluster the cancer samples, the result of which is shown in comparison to the true cancer types (Table 2). We can see that the algorithm generates three clusters, successfully predicting the true number of cancer types contained in the data. The clustering assignments also demonstrate high accuracy, as the majority of samples are correctly clustered with samples of the same type.

■ **Table 2** Clustering results of the TCGA dataset, where the clustering assignments from *Shrinkage Clustering* are compared against the three known tumor types.

	Cluster 1	Cluster 2	Cluster 3
BRCA	3	204	0
GBM	0	0	67
LUSC	17	2	0

■ **Table 3** Performances of *Shrinkage Clustering* on Simulated, Iris and Wine data, where the clustering assignments are compared against the three simulated centers, three Iris species and three wine types respectively.

Simulated				Iris			Wine			
Center	C1	C2	C3	Species	C1	C2	Type	C1	C2	C3
$(-2, 2)$	0	49	1	*setosa*	50	0	1	0	59	0
$(-2, -2)$	0	1	49	*versicolor*	0	50	2	59	6	0
$(2, 0)$	50	0	0	*virginica*	0	50	3	0	6	48

3.2   Experiments on Feature-based Data

3.2.1   Testing with Simulated and Standardized Data

Since similarity matrices are not always available in most clustering applications, we now test the performance of *Shrinkage Clustering* using feature-based data that does not directly provide the similarity information between objects. To run *Shrinkage Clustering*, we first convert the data to a similarity matrix using $S = \exp(-(D(X)/\beta\sigma)^2)$, where $[D(X)]_{ij}$ is the Euclidean distance between X_i and X_j, σ is the standard deviation of $D(X)$, and $\beta = E(D(X)^2)/\sigma^2$. The same conversion method is used for all datasets in the rest of this paper.

As a proof of concept, we first generate a simulated three-cluster two-dimensional data set by sampling 50 points for each cluster from bivariate normal distributions with a common identity covariance matrix around centers at $(-2, 2)$, $(-2, 2)$ and $(0, 2)$ respectively. The clustering result from *Shrinkage Clustering* is shown in Table 3, where the algorithm successfully determines the existence of 3 clusters in the data and obtains a clustering solution with high accuracy.

Next, we test the performance of *Shrinkage Clustering* using two real data sets, the Iris [8] and the wine data [1], both of which are frequently used to test clustering algorithms; and they can be downloaded from the University of California Irvine (UCI) machine learning repository [2]. The clustering results from *Shrinkage Clustering* for both datasets are shown in Table 3, where the clustering assignments are compared to the true cluster memberships of the Iris and the wine samples respectively. In application to the wine data, *Shrinkage Clustering* successfully identifies a correct number of 3 wine types and produces highly accurate cluster memberships. For the Iris data, though the algorithm generates two instead of three clusters, the result is acceptable because the species *versicolor* and *virginica* are known to be hardly distinguishable given the features collected.

Table 4 Clustering Accuracy Comparison of the seven algorithms based on NMI (Normalized Mutual Information), Rand Index, F1 score and K (the optimal cluster number).

Type	Metric	Shrinkage	Spectral	K-means	Hierarchical	PAM	DBSCAN	Affinity
BCWD	NMI	0.50	0.29	0.46	0.09	0.50	0.29	0.27
	Rand	0.77	0.68	0.75	0.55	0.77	0.64	0.52
	F1	0.80	0.69	0.79	0.69	0.80	0.72	0.22
	K (2)	2	2	2	2	2	3	21
AIBT	NMI	0.56	0.20	0.58	0.17	0.54	0.31	0.40
	Rand	0.79	0.68	0.80	0.37	0.78	0.64	0.76
	F1	0.61	0.39	0.62	0.40	0.59	0.43	0.26
	K (4)	4	4	4	4	4	5	31

3.2.2 Biological Case Study 1: Breast Cancer Wisconsin Diagnostic (BCWD)

The BCWD dataset [28, 17] contains 569 breast cancer samples (357 benign and 212 malignant) with 30 characteristic features computed from a digitized image of a fine needle aspirate (FNA) of a breast mass. The dataset is available on the UCI machine learning repository [2] and is one of the most popularly tested dataset for clustering and classification. Here, we apply *Shrinkage Clustering* to the data and compare its performance against six commonly used clustering methods: *Spectral Clustering* [33], *K-means* [15], *Hierarchical Clustering* [11] (Ward's method [30]), *PAM* [16], *DBSCAN* [7] and *Affinity Propagation* [9]. Since *K-means*, *Spectral Clustering*, *Hierarchical Clustering* and *PAM* do not inherently determine the optimal cluster number and require the cluster number as an input, we first run these algorithms with cluster numbers from 2 to 10, and then use the mean *Silhouette* width as the criterion to select the optimal cluster number. The accuracy of all clustering solutions is evaluated using four metrics: Normalized Mutual Information (NMI) [18], Rand Index [18], F1 score [18], and the optimal cluster number (K).

The performance results (Table 4) show that *Shrinkage Clustering* correctly predicts a 2 cluster structure from the data and generates the clustering assignments with high accuracy. Notably, *Shrinkage Clustering* is the only algorithm that correctly estimates the optimal cluster number among the three algorithms (together with *DBSCAN* and *Affinity Propagation*) that inherently determine cluster numbers. When comparing the cluster assignments against the true cluster memberships, we can see that *Shrinkage Clustering* is one of the two best performers (with *PAM*) across all accuracy metrics.

3.2.3 Biological Case Study 2: Allen Institute Brain Tissue (AIBT)

The AIBT dataset [21] contains RNA sequencing data of 377 samples from four types of brain tissues, i.e. 99 samples of temporal cortex, 91 samples of parietal cortex, 93 samples of cortical white matter, and 94 samples hippocampus isolated by macro-dissection. For each sample, the expression levels of 50282 genes are included as features, and each feature is normalized to have a mean of 0 and a standard deviation of 1 prior to testing. In contrast to the previous case study, the AIBT data is much larger in size with significantly more features being measured. Therefore, this would be a great example to test both the accuracy and the speed of clustering algorithms in face of greater data sizes and higher dimensions.

Similar to the previous case study, we apply *Shrinkage Clustering* and the six commonly used clustering algorithms to the data, and use mean *Silhouette* width to select the optimal
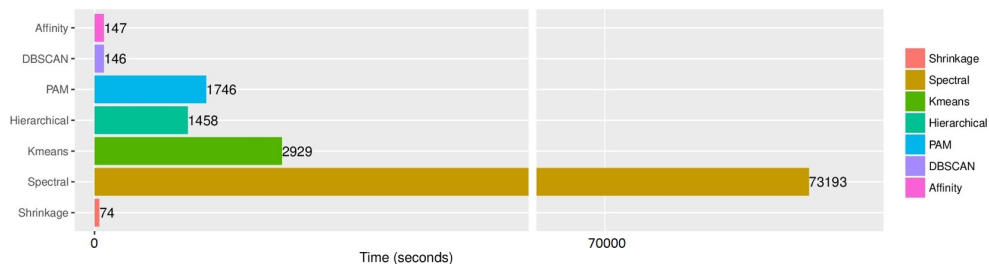
■ **Figure 2** Computation time of all clustering algorithms in application to the AIBT data.

cluster number for algorithms that do not inherently determine the cluster number. The performances of all seven algorithms measured across the four accuracy metrics (i.e. NMI, Rand, F1, K) are shown in Table 4. Among the three algorithms with built-in cluster number determination, *Shrinkage Clustering* is the only algorithm that accurately predicts a 4 cluster structure, whereas *DBSCAN* and *Affinity Propagation* both over-estimate the total number of clusters present in the data. In addition, *Shrinkage Clustering* is the second best performer among all seven algorithms in terms of clustering quality, with comparable accuracy to the top performer (*K-means*).

Next, we record and compare the speed of the seven algorithms for clustering the data. The speed comparison results, shown in Figure 2, demonstrate the unparalleled speed of *Shrinkage Clustering* compared to the rest of the algorithms. *Shrinkage Clustering* is twice as fast as *DBSCAN* and *Affinity Propagation*, and it is faster than *Hierachical*, *PAM* and *K-means* by more than 20 times. In particular, the same data that takes *Shrinkage Clustering* only 73 seconds to cluster can take *Spectral clustering* more than 20 hours.

4   Discussion

From the biological case studies, we showed that *Shrinkage Clustering* is computationally advantageous in speed with high clustering accuracy comparable to the best performers of the six commonly used clustering algorithms. This advantage in speed mainly comes from the fact that *Shrinkage Clustering* integrates the clustering of the data and the determination of the optimal cluster number into one seamless process, so the algorithm only needs to run once in order to complete the clustering task. In contrast, algorithms like *K-means*, *PAM* and *Spectral Clustering* perform clustering on a single cluster number basis, therefore they need to be repeatedly run for all cluster numbers of interest before a clustering evaluation method can be applied. Notably, the clustering evaluation method *Silhouette* that we used in this experiment does not perform any repetitive clustering validation and therefore is a much faster method compared to other commonly used methods that require repetitive validation [10]. This means that *Shrinkage Clustering* would have an even greater advantage in computation speed compared to the methods tested in this paper if we use a cluster evaluation method that has a repetitive nature (e.g. *Consensus Clustering*, *Gap Statistics*, *Stability Selection*).

One prominent feature of *Shrinkage Clustering* is its flexibility to solve semi-supervised clustering problems, i.e. its ability to add constraints to the minimum cluster size. The size constraints can help prevent generating empty or tiny clusters (which are often observed in *Hierarchical Clustering* and sometimes in *K-means* applications), and can produce clusters of sufficiently large sample sizes as required by the user. This is particularly useful when we need to perform subsequent statistical analyses based on the clustering solution, since clusters

of too small a size can make a statistical testing infeasible. For example, one application of cluster analysis in clinical studies is identifying subpopulations of cancer patients based on their gene expression levels, which is usually followed with a survival analysis to determine the prognostic value of the gene expression patterns. In this case, clusters that contain too few patients can hardly generate any significant or meaningful patient outcome comparison. In addition, it is difficult to take actions based on tiny patient clusters (e.g. in the context of designing clinical trials), since these clusters are hard to validate.

In summary, we developed a new NMF-based clustering method, *Shrinkage Clustering*, which shrinks the number of clusters to an optimum while simultaneously optimizing the cluster memberships. The algorithm performed with high accuracy on both simulated and actual data, exhibited excellent robustness to noise, and demonstrated superior speeds compared to some of the commonly used algorithms. The base algorithm has also been extended to accommodate requirements on minimum cluster sizes, which can be particularly beneficial to clinical studies and the general biomedical community. The size-constrained extension also illustrates the flexibility of the *Shrinkage Clustering* algorithm framework and the ease of adapting it towards semi-supervised learning tasks. Interesting future research directions include exploring *Shrinkage Clustering*'s capability to deal with missing data or incomplete similarity matrices, as well as extending the algorithm to handle semi-supervised clustering tasks with must-link and cannot-link constraints.

References

1 S. Aeberhard, D. Coomans, and O. De Vel. Comparison of classifiers in high dimensional settings. *Dept. Math. Statist., James Cook Univ., North Queensland, Australia, Tech. Rep*, (92-02), 1992.

2 Arthur Asuncion and David Newman. Uci machine learning repository, 2007.

3 P. S. Bradley, K. P. Bennett, and Ayhan Demiriz. Constrained k-means clustering. *Microsoft Research, Redmond*, pages 1–8, 2000.

4 Jean-Philippe Brunet, Pablo Tamayo, Todd R. Golub, and Jill P. Mesirov. Metagenes and molecular pattern discovery using matrix factorization. *Proceedings of the national academy of sciences*, 101(12):4164–4169, 2004.

5 Elisa Boari de Lima, Wagner Meira Júnior, and Raquel Cardoso de Melo-Minardi. Isofunctional protein subfamily detection using data integration and spectral clustering. *PLoS Comput Biol*, 12(6):e1005001, 2016.

6 Chris Ding, Xiaofeng He, and Horst D. Simon. On the equivalence of nonnegative matrix factorization and spectral clustering. In *Proceedings of the 2005 SIAM International Conference on Data Mining*, pages 606–610. SIAM, 2005.

7 Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *KDD*, volume 96, pages 226–231, 1996.

8 Ronald A. Fisher. The use of multiple measurements in taxonomic problems. *Annals of eugenics*, 7(2):179–188, 1936.

9 Brendan J. Frey and Delbert Dueck. Clustering by passing messages between data points. *science*, 315(5814):972–976, 2007.

10 Chenyue W. Hu, Steven M. Kornblau, John H. Slater, and Amina A. Qutub. Progeny clustering: A method to identify biological phenotypes. *Scientific reports*, 5, 2015.

11 Stephen C. Johnson. Hierarchical clustering schemes. *Psychometrika*, 32(3):241–254, 1967.

12 Da Kuang, Chris Ding, and Haesun Park. Symmetric nonnegative matrix factorization for graph clustering. In *Proceedings of the 2012 SIAM international conference on data mining*, pages 106–117. SIAM, 2012.

13   Tilman Lange, Volker Roth, Mikio L. Braun, and Joachim M. Buhmann. Stability-based validation of clustering solutions. *Neural computation*, 16(6):1299–1323, 2004.

14   Tao Li and Chris H. Q. Ding. Nonnegative matrix factorizations for clustering: A survey., 2013.

15   James MacQueen et al. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, pages 281–297. California, USA, 1967.

16   Martin Maechler, Peter Rousseeuw, Anja Struyf, Mia Hubert, and Kurt Hornik. Cluster: cluster analysis basics and extensions. *R package version*, 1(2):56, 2012.

17   Olvi L Mangasarian, W. Nick Street, and William H. Wolberg. Breast cancer diagnosis and prognosis via linear programming. *Operations Research*, 43(4):570–577, 1995.

18   Christopher D. Manning, Prabhakar Raghavan, Hinrich Schütze, et al. *Introduction to information retrieval*, volume 1. Cambridge University Press, 2008.

19   Geoffrey J. McLachlan and Kaye E. Basford. Mixture models. inference and applications to clustering. *Statistics: Textbooks and Monographs, New York: Dekker, 1988*, 1, 1988.

20   Stefano Monti, Pablo Tamayo, Jill Mesirov, and Todd Golub. Consensus clustering: a resampling-based method for class discovery and visualization of gene expression microarray data. *Machine learning*, 52(1-2):91–118, 2003.

21   Thomas J. Montine, Joshua A. Sonnen, Kathleen S. Montine, Paul K. Crane, and Eric B. Larson. Adult changes in thought study: dementia is an individually varying convergent syndrome with prevalent clinically silent diseases that may be modified by some commonly used therapeutics. *Current Alzheimer Research*, 9(6):718–723, 2012.

22   Wendy C. Moore, Deborah A. Meyers, Sally E. Wenzel, W. Gerald Teague, Huashi Li, Xingnan Li, Ralph D'Agostino Jr., Mario Castro, Douglas Curran-Everett, Anne M. Fitzpatrick, et al. Identification of asthma phenotypes using cluster analysis in the severe asthma research program. *American journal of respiratory and critical care medicine*, 181(4):315–323, 2010.

23   Dan Pelleg, Andrew W. Moore, et al. X-means: Extending K-means with Efficient Estimation of the Number of Clusters. In *ICML*, pages 727–734, 2000.

24   Peter J. Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 20:53–65, 1987.

25   Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(8):888–905, 2000.

26   John H Slater, James C. Culver, Byron L. Long, Chenyue W. Hu, Jingzhe Hu, Taylor F Birk, Amina A. Qutub, Mary E. Dickinson, and Jennifer L. West. Recapitulation and modulation of the cellular architecture of a user-chosen cell of interest using cell-derived, biomimetic patterning. *ACS nano*, 9(6):6128–6138, 2015.

27   Nora Speicher and Thomas Lengauer. *Towards the identification of cancer subtypes by integrative clustering of molecular data*. PhD thesis, Universität des Saarlandes Saarbrücken, 2012.

28   W. Nick Street, William H. Wolberg, and Olvi L. Mangasarian. Nuclear feature extraction for breast tumor diagnosis. In *IS&T/SPIE's Symposium on Electronic Imaging: Science and Technology*, pages 861–870. International Society for Optics and Photonics, 1993.

29   Robert Tibshirani, Guenther Walther, and Trevor Hastie. Estimating the number of clusters in a data set via the gap statistic. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 63(2):411–423, 2001.

30   Joe H. Ward Jr. Hierarchical grouping to optimize an objective function. *Journal of the American statistical association*, 58(301):236–244, 1963.

31 Pratyaksha Wirapati, Christos Sotiriou, Susanne Kunkel, Pierre Farmer, Sylvain Pradervand, Benjamin Haibe-Kains, Christine Desmedt, Michail Ignatiadis, Thierry Sengstag, Frédéric Schütz, et al. Meta-analysis of gene expression profiles in breast cancer: toward a unified understanding of breast cancer subtyping and prognosis signatures. *Breast Cancer Research*, 10(4):R65, 2008.

32 Christian Wiwie, Jan Baumbach, and Richard Röttger. Comparing the performance of biomedical clustering methods. *Nature Methods*, 12(11):1033–1038, 2015.

33 Achim Zeileis, Kurt Hornik, Alex Smola, and Alexandros Karatzoglou. kernlab-an S4 package for kernel methods in R. *Journal of statistical software*, 11(9):1–20, 2004.