# Robust continuous clustering

**Sohil Atul Shah**[a,1] **and Vladlen Koltun**[b]

[a]Department of Electrical and Computer Engineering, University of Maryland, College Park, MD 20740; and [b]Intel Labs, Santa Clara, CA 95054

Edited by David L. Donoho, Stanford University, Stanford, CA, and approved August 7, 2017 (received for review January 13, 2017)

**Clustering is a fundamental procedure in the analysis of scientific data. It is used ubiquitously across the sciences. Despite decades of research, existing clustering algorithms have limited effectiveness in high dimensions and often require tuning parameters for different domains and datasets. We present a clustering algorithm that achieves high accuracy across multiple domains and scales efficiently to high dimensions and large datasets. The presented algorithm optimizes a smooth continuous objective, which is based on robust statistics and allows heavily mixed clusters to be untangled. The continuous nature of the objective also allows clustering to be integrated as a module in end-to-end feature learning pipelines. We demonstrate this by extending the algorithm to perform joint clustering and dimensionality reduction by efficiently optimizing a continuous global objective. The presented approach is evaluated on large datasets of faces, handwritten digits, objects, newswire articles, sensor readings from the Space Shuttle, and protein expression levels. Our method achieves high accuracy across all datasets, outperforming the best prior algorithm by a factor of 3 in average rank.**

clustering | data analysis | unsupervised learning

Clustering is one of the fundamental experimental procedures in data analysis. It is used in virtually all natural and social sciences and has played a central role in biology, astronomy, psychology, medicine, and chemistry. Data-clustering algorithms have been developed for more than half a century (1). Significant advances in the last two decades include spectral clustering (2–4), generalizations of classic center-based methods (5, 6), mixture models (7, 8), mean shift (9), affinity propagation (10), subspace clustering (11–13), nonparametric methods (14, 15), and feature selection (16–20).

Despite these developments, no single algorithm has emerged to displace the $k$-means scheme and its variants (21). This is despite the known drawbacks of such center-based methods, including sensitivity to initialization, limited effectiveness in high-dimensional spaces, and the requirement that the number of clusters be set in advance. The endurance of these methods is in part due to their simplicity and in part due to difficulties associated with some of the new techniques, such as additional hyperparameters that need to be tuned, high computational cost, and varying effectiveness across domains. Consequently, scientists who analyze large high-dimensional datasets with unknown distribution must maintain and apply multiple different clustering algorithms in the hope that one will succeed. Books have been written to guide practitioners through the landscape of data-clustering techniques (22).

We present a clustering algorithm that is fast, easy to use, and effective in high dimensions. The algorithm optimizes a clear continuous objective, using standard numerical methods that scale to massive datasets. The number of clusters need not be known in advance.

The operation of the algorithm can be understood by contrasting it with other popular clustering techniques. In center-based algorithms such as $k$-means (1, 24), a small set of putative cluster centers is initialized from the data and then iteratively refined. In affinity propagation (10), data points communicate over a graph structure to elect a subset of the points as representatives. In the presented algorithm, each data point has a dedicated representative, initially located at the data point. Over the course of the algorithm, the representatives move and coalesce into easily separable clusters. The progress of the algorithm is visualized in Fig. 1.

Our formulation is based on recent convex relaxations for clustering (25, 26). However, our objective is deliberately not convex. We use redescending robust estimators that allow even heavily mixed clusters to be untangled by optimizing a single continuous objective. Despite the nonconvexity of the objective, the optimization can still be performed using standard linear least-squares solvers, which are highly efficient and scalable. Since the algorithm expresses clustering as optimization of a continuous objective based on robust estimation, we call it robust continuous clustering (RCC).

One of the characteristics of the presented formulation is that clustering is reduced to optimization of a continuous objective. This enables the integration of clustering in end-to-end feature learning pipelines. We demonstrate this by extending RCC to perform joint clustering and dimensionality reduction. The extended algorithm, called RCC-DR, learns an embedding of the data into a low-dimensional space in which it is clustered. Embedding and clustering are performed jointly, by an algorithm that optimizes a clear global objective.

We evaluate RCC and RCC-DR on a large number of datasets from a variety of domains. These include image datasets, document datasets, a dataset of sensor readings from the Space Shuttle, and a dataset of protein expression levels in mice. Experiments demonstrate that our method significantly outperforms prior state-of-the-art techniques. RCC-DR is particularly robust across datasets from different domains, outperforming the best prior algorithm by a factor of 3 in average rank.

## Formulation

We consider the problem of clustering a set of $n$ data points. The input is denoted by $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n]$, where $\mathbf{x}_i \in \mathbb{R}^D$. Our approach operates on a set of representatives $\mathbf{U} = [\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_n]$, where $\mathbf{u}_i \in \mathbb{R}^D$. The representatives $\mathbf{U}$ are initialized at the corresponding data points $\mathbf{X}$. The optimization operates on the representation $\mathbf{U}$, which coalesces to reveal the cluster structure latent in the data. Thus, the number of clusters

### Significance

**Clustering is a fundamental experimental procedure in data analysis. It is used in virtually all natural and social sciences and has played a central role in biology, astronomy, psychology, medicine, and chemistry. Despite the importance and ubiquity of clustering, existing algorithms suffer from a variety of drawbacks and no universal solution has emerged. We present a clustering algorithm that reliably achieves high accuracy across domains, handles high data dimensionality, and scales to large datasets. The algorithm optimizes a smooth global objective, using efficient numerical methods. Experiments demonstrate that our method outperforms state-of-the-art clustering algorithms by significant factors in multiple domains.**

Author contributions: S.A.S. and V.K. designed research, performed research, analyzed data, and wrote the paper.

The authors declare no conflict of interest.

This article is a PNAS Direct Submission.

Freely available online through the PNAS open access option.

[1]To whom correspondence should be addressed. Email: sohilas@umd.edu.

This article contains supporting information online at www.pnas.org/lookup/suppl/doi:10.1073/pnas.1700770114/-/DCSupplemental.

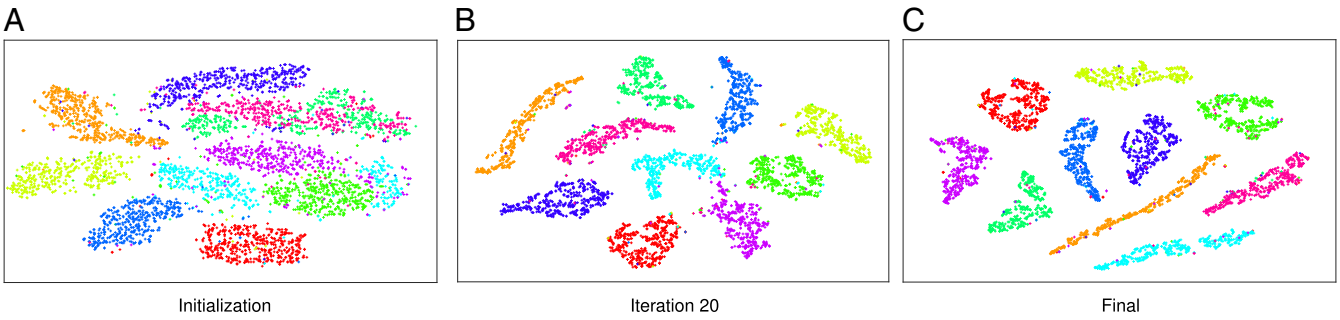


**Fig. 1.** RCC on the Modified National Institute of Standards and Technology (MNIST) dataset. Each data point $\mathbf{x}_i$ has a corresponding representative $\mathbf{u}_i$. The representatives are optimized to reveal the structure of the data. *A*–*C* visualize the representation $\mathbf{U}$ using the t-SNE algorithm (23). Ground-truth clusters are coded by color. (*A*) The initial state, $\mathbf{U} = \mathbf{X}$. (*B*) The representation $\mathbf{U}$ after 20 iterations of the optimization. (*C*) The final representation produced by the algorithm.

need not be known in advance. The optimization of $\mathbf{U}$ is illustrated in Fig. 1.

The RCC objective has the following form:

$$\mathbf{C}(\mathbf{U}) = \frac{1}{2}\sum_{i=1}^{n} \|\mathbf{x}_i - \mathbf{u}_i\|_2^2 + \frac{\lambda}{2}\sum_{(p,q)\in\mathcal{E}} w_{p,q}\,\rho\big(\|\mathbf{u}_p - \mathbf{u}_q\|_2\big). \quad [1]$$

Here $\mathcal{E}$ is the set of edges in a graph connecting the data points. The graph is constructed automatically from the data. We use mutual $k$-nearest neighbors (m-kNN) connectivity (27), which is more robust than commonly used kNN graphs. The weights $w_{p,q}$ balance the contribution of each data point to the pairwise terms and $\lambda$ balances the strength of different objective terms.

The function $\rho(\cdot)$ is a penalty on the regularization terms. The use of an appropriate robust penalty function $\rho$ is central to our method. Since we want representatives $\mathbf{u}_i$ of observations from the same latent cluster to collapse into a single point, a natural penalty would be the $\ell_0$ norm ($\rho(y) = [y \neq 0]$, where $[\cdot]$ is the Iverson bracket). However, this transforms the objective into an intractable combinatorial optimization problem. At another extreme, recent work has explored the use of convex penalties, such as the $\ell_1$ and $\ell_2$ norms (25, 26). This has the advantage of turning objective **1** into a convex optimization problem. However, convex functions—even the $\ell_1$ norm—have limited robustness to spurious edges in the connectivity structure $\mathcal{E}$, because the influence of a spurious pairwise term does not diminish as representatives move apart during the optimization. Given noisy real-world data, heavy contamination of the connectivity structure by connections across different underlying clusters is inevitable. Our method uses robust estimators to automatically prune spurious intercluster connections while maintaining veridical intracluster correspondences, all within a single continuous objective.

The second term in objective **1** is related to the mean shift objective (9). The RCC objective differs in that it includes an additional data term, uses a sparse (as opposed to a fully connected) connectivity structure, and is based on robust estimation.

Our approach is based on the duality between robust estimation and line processes (28). We introduce an auxiliary variable $l_{p,q}$ for each connection $(p, q) \in \mathcal{E}$ and optimize a joint objective over the representatives $\mathbf{U}$ and the line process $\mathbb{L} = \{l_{p,q}\}$:

$$\mathbf{C}(\mathbf{U},\mathbb{L}) = \frac{1}{2}\sum_{i=1}^{n} \|\mathbf{x}_i - \mathbf{u}_i\|_2^2 + \frac{\lambda}{2}\sum_{(p,q)\in\mathcal{E}} w_{p,q}\Big(l_{p,q}\|\mathbf{u}_p - \mathbf{u}_q\|_2^2 + \Psi(l_{p,q})\Big). \quad [2]$$

Here $\Psi(l_{p,q})$ is a penalty on ignoring a connection $(p, q)$: $\Psi(l_{p,q})$ tends to zero when the connection is active ($l_{p,q} \to 1$) and to one when the connection is disabled ($l_{p,q} \to 0$). A broad variety of robust estimators $\rho(\cdot)$ have corresponding penalty functions $\Psi(\cdot)$ such that objectives **1** and **2** are equivalent with respect to $\mathbf{U}$: Optimizing either of the two objectives yields the same set of representatives $\mathbf{U}$. This formulation is related to iteratively reweighted least squares (IRLS) (29), but is more flexible due to the explicit variables $\mathbb{L}$ and the ability to define additional terms over these variables.

Objective **2** can be optimized by any gradient-based method. However, its form enables efficient and scalable optimization by iterative solution of linear least-squares systems. This yields a general approach that can accommodate many robust nonconvex functions $\rho$, reduces clustering to the application of highly optimized off-the-shelf linear system solvers, and easily scales to datasets with hundreds of thousands of points in tens of thousands of dimensions. In comparison, recent work has considered a specific family of concave penalties and derived a computationally intensive majorization–minimization scheme for optimizing the objective in this special case (30). Our work provides a highly efficient general solution.

While the presented approach can accommodate many estimators in the same computationally efficient framework, our exposition and experiments use a form of the well-known Geman–McClure estimator (31),

$$\rho(y) = \frac{\mu y^2}{\mu + y^2}, \quad [3]$$

where $\mu$ is a scale parameter. The corresponding penalty function that makes objectives **1** and **2** equivalent with respect to $\mathbf{U}$ is

$$\Psi(l_{p,q}) = \mu\Big(\sqrt{l_{p,q}} - 1\Big)^2. \quad [4]$$

## Optimization

Objective **2** is biconvex on $(\mathbf{U}, \mathbb{L})$. When variables $\mathbf{U}$ are fixed, the individual pairwise terms decouple and the optimal value of each $l_{p,q}$ can be computed independently in closed form. When variables $\mathbb{L}$ are fixed, objective **2** turns into a linear least-squares problem. We exploit this special structure and optimize the objective by alternatingly updating the variable sets $\mathbf{U}$ and $\mathbb{L}$. As a block coordinate descent algorithm, this alternating minimization scheme provably converges.

When $\mathbf{U}$ are fixed, the optimal value of each $l_{p,q}$ is given by

$$l_{p,q} = \left(\frac{\mu}{\mu + \|\mathbf{u}_p - \mathbf{u}_q\|_2^2}\right)^2. \quad [5]$$

This can be verified by substituting Eq. **5** into Eq. **2**, which yields objective **1** with respect to $\mathbf{U}$.

When $\mathbb{L}$ are fixed, we can rewrite **[2]** in matrix form and obtain a simplified expression for solving $\mathbf{U}$,

$$\arg\min \frac{1}{2}\|\mathbf{X} - \mathbf{U}\|_F^2 + \frac{\lambda}{2}\sum_{(p,q)\in\mathcal{E}} w_{p,q}l_{p,q}\|\mathbf{U}(\mathbf{e}_p - \mathbf{e}_q)\|_2^2, \quad [6]$$

where $\mathbf{e}_i$ is an indicator vector with the $i$th element set to 1. This is a linear least-squares problem that can be efficiently solved using fast and scalable solvers. The linear least-squares formulation is given by

$$\mathbf{UM} = \mathbf{X}, \text{ where} \quad \mathbf{M} = \mathbf{I} + \lambda \sum_{(p,q)\in\mathcal{E}} w_{p,q} l_{p,q} (\mathbf{e}_p - \mathbf{e}_q)(\mathbf{e}_p - \mathbf{e}_q)^\top. \quad [7]$$

Here $\mathbf{I} \in \mathbb{R}^{n\times n}$ is the identity matrix. It is easy to prove that

$$\mathbf{A} \triangleq \sum_{(p,q)\in\mathcal{E}} w_{p,q} l_{p,q} (\mathbf{e}_p - \mathbf{e}_q)(\mathbf{e}_p - \mathbf{e}_q)^\top \quad [8]$$

is a Laplacian matrix and hence $\mathbf{M}$ is symmetric and positive semidefinite. As with any multigrid solver, each row of $\mathbf{U}$ in Eq. **7** can be solved independently and in parallel.

The RCC algorithm is summarized in *Algorithm 1: RCC*. Note that all updates of $\mathbf{U}$ and $\mathbb{L}$ optimize the same continuous global objective **2**.

The algorithm uses graduated nonconvexity (32). It begins with a locally convex approximation of the objective, obtained by setting $\mu$ such that the second derivative of the estimator is positive ($\ddot{\rho}(y) > 0$) over the relevant part of the domain. Over the iterations, $\mu$ is automatically decreased, gradually introducing nonconvexity into the objective. Under certain assumptions, such continuation schemes are known to attain solutions that are close to the global optimum (33).

The parameter $\lambda$ in the RCC objective **1** balances the strength of the data terms and pairwise terms. The reformulation of RCC as a linear least-squares problem enables setting $\lambda$ automatically. Specifically, Eq. **7** suggests that the data terms and pairwise terms can be balanced by setting

$$\lambda = \frac{\|\mathbf{X}\|_2}{\|\mathbf{A}\|_2}. \quad [9]$$

The value of $\lambda$ is updated automatically according to this formula after every update of $\mu$. An update involves computing only the largest eigenvalue of the Laplacian matrix $\mathbf{A}$. The spectral norm of $\mathbf{X}$ is precomputed at initialization and reused.

Additional details concerning Algorithm 1 are provided in *SI Methods*.

### Joint Clustering and Dimensionality Reduction

The RCC formulation can be interpreted as learning a graph-regularized embedding $\mathbf{U}$ of the data $\mathbf{X}$. In Algorithm 1 the dimensionality of the embedding $\mathbf{U}$ is the same as the dimensionality of the data $\mathbf{X}$. However, since RCC optimizes a continuous and differentiable objective, it can be used within end-to-end feature learning pipelines. We now demonstrate this by extending RCC to perform joint clustering and dimensionality reduction. Such joint optimization has been considered in recent work (34, 35). The algorithm we develop, RCC-DR, learns a linear mapping into a reduced space in which the data are clustered. The mapping is optimized as part of the clustering objective, yielding an embedding in which the data can be clustered most effectively. RCC-DR inherits the appealing properties of RCC: Clustering and dimensionality reduction are performed jointly by optimizing a clear continuous objective, the framework supports nonconvex robust estimators that can untangle mixed clusters, and optimization is performed by efficient and scalable numerical methods.

We begin by considering an initial formulation for the RCC-DR objective:

$$\mathbf{C}(\mathbf{U},\mathbf{Z},\mathbf{D}) = \|\mathbf{X} - \mathbf{DZ}\|_2^2 + \gamma \sum_{i=1}^{n} \|\mathbf{z}_i\|_1 + \nu \left( \sum_{i=1}^{n} \|\mathbf{z}_i - \mathbf{u}_i\|_2^2 + \frac{\lambda}{2} \sum_{(p,q)\in\mathcal{E}} w_{p,q} \rho \left( \|\mathbf{u}_p - \mathbf{u}_q\|_2 \right) \right). \quad [10]$$

Here $\mathbf{D} \in \mathbb{R}^{D\times d}$ is a dictionary, $\mathbf{z}_i \in \mathbb{R}^d$ is a sparse code corresponding to the $i^{\text{th}}$ data sample, and $\mathbf{u}_i \in \mathbb{R}^d$ is the low-dimensional embedding of $\mathbf{x}_i$. For a fixed $\mathbf{D}$, the parameter $\nu$ balances the data term in the sparse coding objective with the clustering objective in the reduced space. This initial formulation **10** is problematic because in the beginning of the optimization the representation $\mathbf{U}$ can be noisy due to spurious intercluster connections that have not yet been disabled. This had no effect on the convergence of the original RCC objective **1**, but in formulation **10** the contamination of $\mathbf{U}$ can infect the sparse coding system via $\mathbf{Z}$ and corrupt the dictionary $\mathbf{D}$. For this reason, we use a different formulation that has the added benefit of eliminating the parameter $\nu$:

$$\mathbf{C}(\mathbf{U},\mathbf{Z},\mathbf{D}) = \|\mathbf{X} - \mathbf{DZ}\|_2^2 + \gamma \sum_{i=1}^{n} \|\mathbf{z}_i\|_1 + \sum_{i=1}^{n} \rho_1 \left( \|\mathbf{z}_i - \mathbf{u}_i\|_2 \right) + \frac{\lambda}{2} \sum_{(p,q)\in\mathcal{E}} w_{p,q} \rho_2 \left( \|\mathbf{u}_p - \mathbf{u}_q\|_2 \right). \quad [11]$$

Here we replaced the $\ell_2$ penalty on the data term in the reduced space with a robust penalty. We use the Geman–McClure estimator **3** for both $\rho_1$ and $\rho_2$.

To optimize objective **11**, we introduce line processes $\mathbb{L}^1$ and $\mathbb{L}^2$ corresponding to the data and pairwise terms in the reduced space, respectively, and optimize a joint objective over $\mathbf{U}$, $\mathbf{Z}$, $\mathbf{D}$, $\mathbb{L}^1$, and $\mathbb{L}^2$. The optimization is performed by block coordinate descent over these groups of variables. The line processes $\mathbb{L}^1$ and $\mathbb{L}^2$ can be updated in closed form as in Eq. **5**. The variables $\mathbf{U}$ are updated by solving the linear system

$$\mathbf{UM}_{\text{dr}} = \mathbf{ZH}, \quad [12]$$

**Algorithm 1. RCC**

I: input: Data samples $\{\mathbf{x}\}_{i=1}^n$.
II: output: Cluster assignment $\{\hat{c}_i\}_{i=1}^n$.
III: Construct connectivity structure $\mathcal{E}$.
IV: Precompute $\chi = \|\mathbf{X}\|_2$, $w_{p,q}$, $\delta$.
V: Initialize $\mathbf{u}_i = \mathbf{x}_i$, $l_{p,q} = 1$, $\mu \gg \max \|\mathbf{x}_p - \mathbf{x}_q\|_2^2$, $\lambda = \frac{\chi}{\|\mathbf{A}\|_2}$.
VI: while $|\mathbf{C}^t - \mathbf{C}^{t-1}| < \varepsilon$ or $t <$ maxiterations do
VII: Update $l_{p,q}$ using Eq. **5** and $\mathbf{A}$ using Eq. **8**.
VIII: Update $\{\mathbf{u}_i\}_{i=1}^n$ using Eq. **7**.
IX: Every four iterations, update $\lambda = \frac{\chi}{\|\mathbf{A}\|_2}$, $\mu = \max\left(\frac{\mu}{2}, \frac{\delta}{2}\right)$.
X: Construct graph $\mathcal{G} = (\mathcal{V}, \mathcal{F})$ with $f_{p,q} = 1$ if $\|\mathbf{u}_p^* - \mathbf{u}_q^*\|_2 < \delta$.
XI: Output clusters given by the connected components of $\mathcal{G}$.

**Table 1. Datasets used in experiments**

| Name | Instances | Dimensions | Classes | Imbalance |
|---|---|---|---|---|
| MNIST (41) | 70,000 | 784 | 10 | ∼1 |
| Coil-100 (45) | 7,200 | 49,152 | 100 | 1 |
| YaleB (43) | 2,414 | 32,256 | 38 | 1 |
| YTF (44) | 10,036 | 9,075 | 40 | 13 |
| Reuters-21578 | 9,082 | 2,000 | 50 | 785 |
| RCV1 (38) | 10,000 | 2,000 | 4 | 6 |
| Pendigits (42) | 10,992 | 16 | 10 | ∼1 |
| Shuttle | 58,000 | 9 | 7 | 4,558 |
| Mice Protein (39) | 1,077 | 77 | 8 | ∼1 |

For each dataset, the number of instances, number of dimensions, number of ground-truth clusters, and the imbalance, defined as the ratio of the largest and smallest cardinalities of ground-truth clusters, are shown.

**Table 2. Accuracy of all algorithms on all datasets, measured by AMI**

| Dataset | *k*-means++ | GMM | Fuzzy | MS | AC-C | AC-W | N-Cuts | AP | Zell | SEC | LDMGI | GDL | PIC | RCC | RCC-DR |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| MNIST | 0.500 | 0.404 | 0.386 | 0.264 | NA | 0.679 | NA | 0.478 | NA | 0.469 | 0.761 | NA | NA | **0.893** | 0.828 |
| COIL-100 | 0.803 | 0.786 | 0.796 | 0.685 | 0.703 | 0.853 | 0.871 | 0.761 | 0.958 | 0.849 | 0.888 | 0.958 | **0.965** | 0.957 | 0.957 |
| YTF | 0.783 | 0.793 | 0.769 | 0.831 | 0.673 | 0.801 | 0.752 | 0.751 | 0.273 | 0.754 | 0.518 | 0.655 | 0.676 | 0.836 | **0.874** |
| YaleB | 0.615 | 0.591 | 0.066 | 0.091 | 0.445 | 0.767 | 0.928 | 0.700 | 0.905 | 0.849 | 0.945 | 0.924 | 0.941 | **0.975** | **0.974** |
| Reuters | 0.516 | 0.507 | 0.272 | 0.000 | 0.368 | 0.471 | 0.545 | 0.386 | 0.087 | 0.498 | 0.523 | 0.401 | 0.057 | **0.556** | 0.553 |
| RCV1 | 0.355 | 0.344 | 0.205 | 0.000 | 0.108 | 0.364 | 0.140 | 0.313 | 0.023 | 0.069 | 0.382 | 0.020 | 0.015 | 0.138 | **0.442** |
| Pendigits | 0.679 | 0.695 | 0.695 | 0.694 | 0.525 | 0.728 | 0.813 | 0.639 | 0.317 | 0.741 | 0.775 | 0.330 | 0.467 | 0.848 | **0.854** |
| Shuttle | 0.215 | 0.266 | 0.204 | 0.362 | NA | 0.291 | 0.000 | 0.322 | NA | 0.305 | **0.591** | NA | NA | 0.488 | 0.513 |
| Mice Protein | 0.425 | 0.385 | 0.417 | 0.534 | 0.315 | 0.525 | 0.536 | 0.554 | 0.428 | 0.537 | 0.527 | 0.400 | 0.394 | **0.649** | 0.638 |
| Rank | 7.8 | 8.6 | 9.9 | 9.9 | 12.4 | 6.3 | 6.3 | 8.1 | 10.4 | 7.2 | 4.9 | 9.9 | 10 | 2.4 | **1.6** |

For each dataset, the maximum AMI is highlighted in bold. Some prior algorithms did not scale to large datasets such as MNIST (70,000 data points in 784 dimensions). RCC or RCC-DR achieves the highest accuracy on seven of the nine datasets. RCC-DR achieves the highest or second-highest accuracy on eight of the nine datasets. The average rank of RCC-DR across datasets is lower by a multiplicative factor of 3 or more than the average rank of any prior algorithm. NA, not applicable.

where

$$\mathbf{M}_{\text{dr}} = \mathbf{H} + \lambda \sum_{(p,q)\in\mathcal{E}} w_{p,q} l_{p,q}^2 (\mathbf{e}_p - \mathbf{e}_q)(\mathbf{e}_p - \mathbf{e}_q)^\top \quad [13]$$

and $\mathbf{H}$ is a diagonal matrix with $h_{i,i} = l_i^1$.

The dictionary $\mathbf{D}$ and codes $\mathbf{Z}$ are initialized using principal component analysis (PCA). [The K-SVD algorithm can also be used for this purpose (36).] The variables $\mathbf{Z}$ are updated by accelerated proximal gradient-descent steps (37),

$$\bar{\mathbf{Z}} = \mathbf{Z}^t + \omega^t(\mathbf{Z}^t - \mathbf{Z}^{t-1}) \quad [14]$$

$$\mathbf{Z}^{t+1} = \mathbf{prox}_{\tau\gamma\|\cdot\|_1}\left(\bar{\mathbf{Z}} - \tau\left(\mathbf{D}^\top(-\mathbf{X} + \mathbf{D}\bar{\mathbf{Z}}) + (\bar{\mathbf{Z}} - \mathbf{U})\mathbf{H}\right)\right),$$

where $\tau = \frac{1}{\|\mathbf{D}^\top\mathbf{D}\|_2 + \|\mathbf{H}\|_2}$ and $\omega^t = \frac{t}{t+3}$. The $\mathbf{prox}_{\varepsilon\|\cdot\|_1}$ operator performs elementwise soft thresholding:

$$\mathbf{prox}_{\varepsilon\|\cdot\|_1}(v) = \text{sign}(v)\max\left(0, |v| - \varepsilon\right). \quad [15]$$

The variables $\mathbf{D}$ are updated using

$$\bar{\mathbf{D}} = \mathbf{X}\mathbf{Z}^\top\left(\mathbf{Z}\mathbf{Z}^\top + \beta\mathbf{I}\right)^{-1} \quad [16]$$

$$\mathbf{D}^{t+1} = \eta\mathbf{D}^t + (1-\eta)\bar{\mathbf{D}}, \quad [17]$$

where $\beta$ is a small regularization value set to $\beta = 10^{-4}\text{tr}(\mathbf{Z}\mathbf{Z}^\top)$.

A precise specification of the RCC-DR algorithm is provided in Algorithm S1.

## Experiments

**Datasets.** We have conducted experiments on datasets from multiple domains. The dimensionality of the data in the different datasets varies from 9 to just below 50,000. `Reuters-21578` is the classic benchmark for text classification, comprising 21,578 articles that appeared on the Reuters newswire in 1987. `RCV1` is a more recent benchmark of 800,000 manually categorized Reuters newswire articles (38). (Due to limited scalability of some prior algorithms, we use 10,000 random samples from RCV1.) `Shuttle` is a dataset from NASA that contains 58,000 multivariate measurements produced by sensors in the radiator subsystem of the Space Shuttle; these measurements are known to arise from seven different conditions of the radiators. `Mice Protein` is a dataset that consists of the expression levels of 77 proteins measured in the cerebral cortex of eight classes of control and trisomic mice (39). The last two datasets were obtained from the University of California, Irvine, machine-learning repository (40).

`MNIST` is the classic dataset of 70,000 hand-written digits (41). `Pendigits` is another well-known dataset of hand-written digits (42). The Extended Yale Face Database B (`YaleB`) contains images of faces of 28 human subjects (43). The YouTube Faces Database (`YTF`) contains videos of faces of different subjects (44); we use all video frames from the first 40 subjects sorted in chronological order. Columbia University Image Library (`COIL-100`) is a classic collection of color images of 100 objects, each imaged from 72 viewpoints (45). The datasets are summarized in Table 1.

**Baselines.** We compare RCC and RCC-DR to 13 baselines, which include widely known clustering algorithms as well as recent techniques that were reported to achieve state-of-the-art performance. Our baselines are *k*-means++ (24), Gaussian mixture models (GMM), fuzzy clustering, mean-shift clustering (MS) (9), two variants of agglomerative clustering (AC-Complete and AC-Ward), normalized cuts (N-Cuts) (2), affinity propagation (AP) (10), Zeta *l*-links (Zell) (46), spectral embedded clustering (SEC) (47), clustering using local discriminant models and global integration (LDMGI) (48), graph degree linkage (GDL) (49), and path integral clustering (PIC) (50). The parameter settings for the baselines are summarized in Table S1.

**Measures.** Normalized mutual information (NMI) has emerged as the standard measure for evaluating clustering accuracy in the machine-learning community (51). However, NMI is known to be biased in favor of fine-grained partitions. For this reason, we use adjusted mutual information (AMI), which removes this bias (52). This measure is defined as follows:

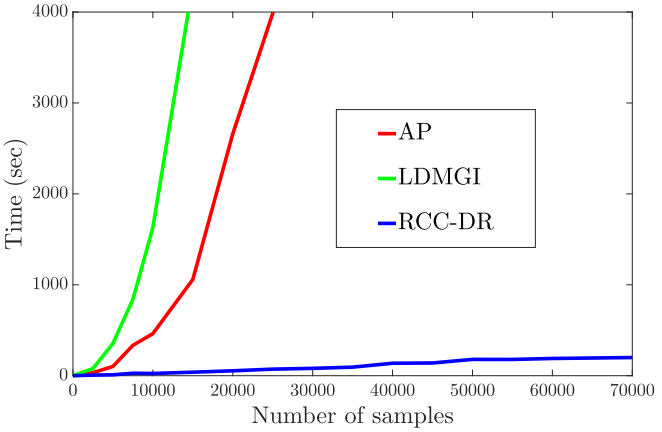


**Fig. 2.** Runtime comparison of RCC-DR with AP and LDMGI. Runtime is evaluated as a function of dataset size, using randomly sampled subsets of different sizes from the MNIST dataset.

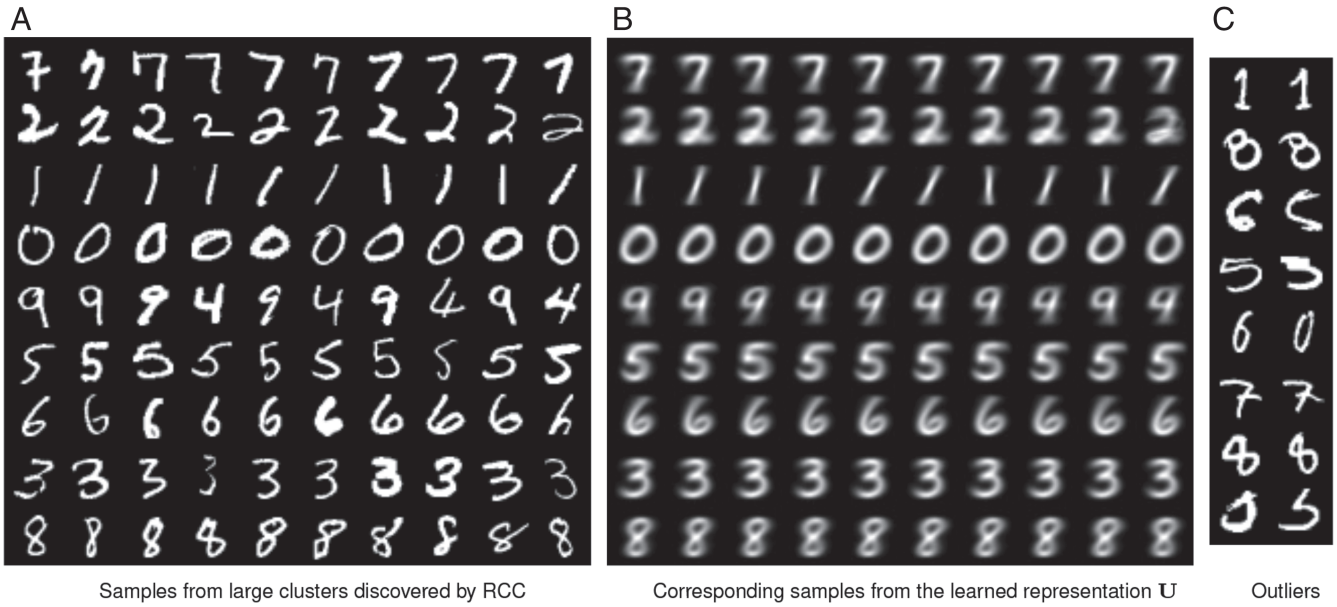


**Fig. 3.** Visualization of RCC output on the MNIST dataset. (*A*) Ten randomly sampled instances $\mathbf{x}_i$ from each large cluster discovered by RCC, one cluster per row. (*B*) Corresponding representatives $\mathbf{u}_i$ from the learned representation $\mathbf{U}$. (*C*) Two random samples from each of the small outlying clusters discovered by RCC.

$$\mathrm{AMI}(\mathbf{c}, \hat{\mathbf{c}}) = \frac{\mathrm{MI}(\mathbf{c}, \hat{\mathbf{c}}) - E[\mathrm{MI}(\mathbf{c}, \hat{\mathbf{c}})]}{\sqrt{H(\mathbf{c})H(\hat{\mathbf{c}})} - E[\mathrm{MI}(\mathbf{c}, \hat{\mathbf{c}})]}. \quad [18]$$

Here $H(\cdot)$ is the entropy, $\mathrm{MI}(\cdot, \cdot)$ is the mutual information, and $\mathbf{c}$ and $\hat{\mathbf{c}}$ are the two partitions being compared. For completeness, Table S2 provides an evaluation using the NMI measure.

**Results.** Results on all datasets are reported in Table 2. In addition to accuracy on each dataset, Table 2 also reports the average rank of each algorithm across datasets. For example, if an algorithm achieves the third-highest accuracy on half of the datasets and the fourth-highest one on the other half, its average rank is 3.5. If an algorithm did not yield a result on a dataset due to its size, that dataset is not taken into account in computing the average rank of the algorithm.

RCC or RCC-DR achieves the highest accuracy on seven of the nine datasets. RCC-DR achieves the highest or second-highest accuracy on eight of the nine datasets and RCC achieves the highest or second-highest accuracy on five datasets. The average rank of RCC-DR and RCC is 1.6 and 2.4, respectively. The best-performing prior algorithm, LDMGI, has an average rank of 4.9, three times higher than the rank of RCC-DR. This indicates that the performance of prior algorithms is not only lower than the performance of RCC and RCC-DR, it is also inconsistent, since no prior algorithm clearly leads the others across datasets. In contrast, the low average rank of RCC and RCC-DR indicates consistently high performance across datasets.

**Clustering Gene Expression Data.** We conducted an additional comprehensive evaluation on a large-scale benchmark that consists of more than 30 cancer gene expression datasets, collected for the purpose of evaluating clustering algorithms (53). The results are reported in Table S3. RCC-DR achieves the highest accuracy on eight of the datasets. Among the prior algorithms, affinity propagation achieves the highest accuracy on six of the datasets and all others on fewer. Overall, RCC-DR achieves the highest average AMI across the datasets.

**Running Time.** The execution time of RCC-DR optimization is visualized in Fig. 2. For reference, we also show the corresponding timings for affinity propagation, a well-known modern clustering algorithm (10), and LDMGI, the baseline that demonstrated the best performance across datasets (48). Fig. 2 shows the running time of each algorithm on randomly sampled subsets of the 784-dimensional MNIST dataset. We sample subsets of different sizes to evaluate runtime growth as a function of dataset size. Performance is measured on a workstation with an Intel Core i7-5960x CPU clocked at 3.0 GHz. RCC-DR clusters the whole MNIST dataset within 200 s, whereas affinity propagation takes 37 h and LDMGI takes 17 h for 40,000 points.

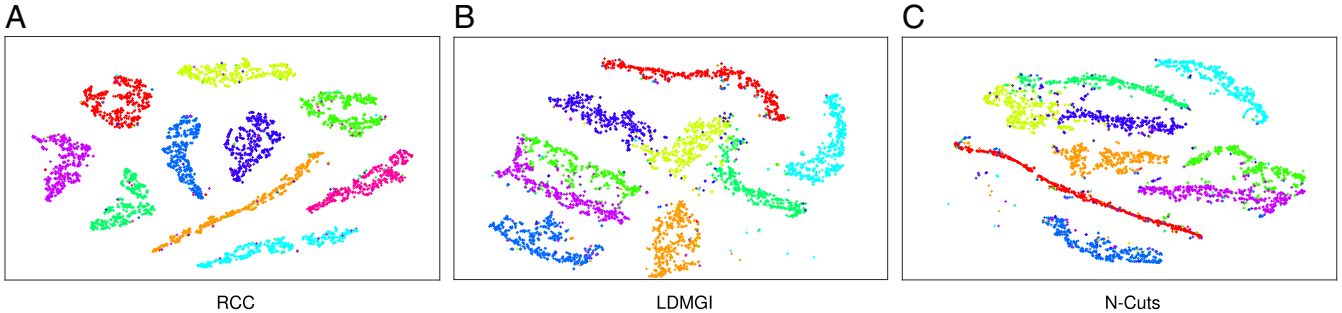


**Fig. 4.** (*A–C*) Visualization of the representations learned by RCC (*A*) and the best-performing prior algorithms, LDMGI (*B*) and N-Cuts (*C*). The algorithms are run on 5,000 randomly sampled instances from the MNIST dataset. The learned representations are visualized using t-SNE.

**Visualization.** We now qualitatively analyze the output of RCC by visualization. We use the MNIST dataset for this purpose. On this dataset, RCC identifies 17 clusters. Nine of these are large clusters with more than 6,000 instances each. The remaining 8 are small clusters that encapsulate outlying data points: Seven of these contain between 2 and 11 instances, and one contains 148 instances. Fig. 3*A* shows 10 randomly sampled data points $\mathbf{x}_i$ from each of the large clusters discovered by RCC. Their corresponding representatives $\mathbf{u}_i$ are shown in Fig. 3*B*. Fig. 3*C* shows 2 randomly sampled data points from each of the small outlying clusters. Additional visualization of RCC output on the Coil-100 dataset is shown in Fig. S3.

Fig. 4 compares the representation $\mathbf{U}$ learned by RCC to representations learned by the best-performing prior algorithms, LDMGI and N-Cuts. We use the MNIST dataset for this purpose and visualize the output of the algorithms on a subset of 5,000 randomly sampled instances from this dataset. Both of the prior algorithms construct Euclidean representations of the data, which can be visualized by dimensionality reduction. We use t-SNE (23) to visualize the representations discovered by the algorithms. As shown in Fig. 4, the representation discovered by RCC cleanly separates the different clusters by significant margins. In contrast, the prior algorithms fail to discover the structure of the data and leave some of the clusters intermixed.

## Discussion

We have presented a clustering algorithm that optimizes a continuous objective based on robust estimation. The objective is optimized using linear least-squares solvers, which scale to large high-dimensional datasets. The robust terms in the objective enable separation of entangled clusters, yielding high accuracy across datasets and domains.

The continuous form of the clustering objective allows it to be integrated into end-to-end feature learning pipelines. We have demonstrated this by extending the algorithm to perform joint clustering and dimensionality reduction.


1. MacQueen J (1967) Some methods for classification and analysis of multivariate observations. *Proc Berkeley Symp Math Stat Probab* 1:281–297.
2. Shi J, Malik J (2000) Normalized cuts and image segmentation. *PAMI* 22:888–905.
3. Ng AY, Jordan MI, Weiss Y (2002) On spectral clustering: Analysis and an algorithm. *Advances in Neural Information Processing Systems 14*, eds Dietterich TG, Becker S, Ghahramani Z (MIT Press, Cambridge, MA), Vol 2, pp 849–856.
4. von Luxburg U (2007) A tutorial on spectral clustering. *Stat Comput* 17:395–416.
5. Banerjee A, Merugu S, Dhillon IS, Ghosh J (2005) Clustering with Bregman divergences. *J Mach Learn Res* 6:1705–1749.
6. Teboulle M (2007) A unified continuous optimization framework for center-based clustering methods. *J Mach Learn Res* 8:65–102.
7. McLachlan G, Peel D (2000) *Finite Mixture Models* (Wiley, New York).
8. Fraley C, Raftery AE (2002) Model-based clustering, discriminant analysis, and density estimation. *J Am Stat Assoc* 97:611.
9. Comaniciu D, Meer P (2002) Mean shift: A robust approach toward feature space analysis. *Pattern Anal Mach Intell* 24:603–619.
10. Frey BJ, Dueck D (2007) Clustering by passing messages between data points. *Science* 315:972–976.
11. Vidal R (2011) Subspace clustering. *IEEE Signal Processing Mag* 28:52–68.
12. Elhamifar E, Vidal R (2013) Sparse subspace clustering: Algorithm, theory, and applications. *Pattern Anal Mach Intell* 35:2765–2781.
13. Soltanolkotabi M, Elhamifar E, Candès EJ (2014) Robust subspace clustering. *Ann Stat* 42:669.
14. Ben-Hur A, Horn D, Siegelmann HT, Vapnik V (2001) Support vector clustering. *J Mach Learn Res* 2:125–137.
15. Kulis B, Jordan MI (2012) Revisiting k-means: New algorithms via Bayesian nonparametrics. *Proceedings of the Twenty-Ninth International Conference on Machine Learning*, eds Langford J, Pineau J (Omnipress, Edinburgh), pp 1131–1138.
16. Friedman JH, Meulman JJ (2004) Clustering objects on subsets of attributes. *J R Stat Soc Ser B* 66:815–849.
17. Tadesse MG, Sha N, Vannucci M (2005) Bayesian variable selection in clustering high-dimensional data. *J Am Stat Assoc* 100:602–617.
18. Raftery AE, Dean N (2006) Variable selection for model-based clustering. *J Am Stat Assoc* 101:168–178.
19. Pan W, Shen X (2007) Penalized model-based clustering with application to variable selection. *J Mach Learn Res* 8:1145–1164.
20. Witten DM, Tibshirani R (2010) A framework for feature selection in clustering. *J Am Stat Assoc* 105:713–726.
21. Jain AK (2010) Data clustering: 50 years beyond K-means. *Pattern Recognition Lett* 31:651–666.
22. Everitt BS, Landau S, Leese M, Stahl D (2011) *Cluster Analysis* (Wiley, Chichester, UK), 5th Ed.
23. van der Maaten L, Hinton GE (2008) Visualizing high-dimensional data using t-SNE. *J Mach Learn Res* 9:2579–2605.
24. Arthur D, Vassilvitskii S (2007) k-means++: The advantages of careful seeding. *SODA '07 Proceedings of Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms* (Society for Industrial and Applied Mathematics, Philadelphia), pp 1027–1035.
25. Hocking T, Joulin A, Bach FR, Vert J (2011) Clusterpath: An algorithm for clustering using convex fusion penalties. *Proceedings of the Twenty-Eighth International Conference on Machine Learning*, eds Getoor L, Scheffer T (Omnipress, Bellevue, WA), pp 1–8.
26. Chi EC, Lange K (2015) Splitting methods for convex clustering. *J Comput Graphical Stat* 24:994–1013.
27. Brito M, Chávez E, Quiroz A, Yukich J (1997) Connectivity of the mutual k-nearest-neighbor graph in clustering and outlier detection. *Stat Probab Lett* 35:33–42.
28. Black MJ, Rangarajan A (1996) On the unification of line processes, outlier rejection, and robust statistics with applications in early vision. *Int J Comput Vis* 19:57–91.
29. Green PJ (1984) Iteratively reweighted least squares for maximum likelihood estimation, and some robust and resistant alternatives. *J R Stat Soc Ser B* 46:149–192.
30. Marchetti Y, Zhou Q (2014) Solution path clustering with adaptive concave penalty. *Electron J Stat* 8:1569–1603.
31. Geman S, McClure DE (1987) Statistical methods for tomographic image reconstruction. *Bull Int Stat Inst* 4:5–21.
32. Blake A, Zisserman A (1987) *Visual Reconstruction* (MIT Press, Cambridge, MA).
33. Mobahi H, Fisher III JW (2015) A theoretical analysis of optimization by Gaussian continuation. Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, eds Bonet B, Koenig S (AAAI Press, Palo Alto, CA), Vol 2, pp 1205–1211.
34. Wang Z, et al. (2015) A joint optimization framework of sparse coding and discriminative clustering. *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence*, eds Yang Q, Wooldridge M (AAAI Press, Palo Alto, CA), pp 3932–3938.
35. Flammarion N, Palanisamy B, Bach FR (2016) Robust discriminative clustering with sparse regularizers. arXiv:160808052.
36. Aharon M, Elad M, Bruckstein A (2006) K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation. *Trans Signal Process* 54:4311–4322.
37. Parikh N, Boyd SP (2014) Proximal algorithms. *Foundations and Trends in Optimization* 1:127–239.
38. Lewis DD, Yang Y, Rose TG, Li F (2004) RCV1: A new benchmark collection for text categorization research. *J Mach Learn Res* 5:361–397.
39. Higuera C, Gardiner KJ, Cios KJ (2015) Self-organizing feature maps identify proteins critical to learning in a mouse model of down syndrome. *PLoS ONE* 10:e0129126.
40. Lichman M (2013) UCI machine learning repository. Available at archive.ics.uci.edu/ml. Accessed December 6, 2016.
41. LeCun Y, Bottou L, Bengio Y, Haffner P (1998) Gradient-based learning applied to document recognition. *Proc IEEE* 86:2278–2324.
42. Alimoglu F, Alpaydin E (1997) Combining multiple representations and classifiers for pen-based handwritten digit recognition. *Proceedings of the Fourth International Conference on Document Analysis and Recognition* (IEEE Computer Society, Los Alamitos, CA), Vol 2, pp 637–640.
43. Georghiades AS, Belhumeur PN, Kriegman DJ (2001) From few to many: Illumination cone models for face recognition under variable lighting and pose. *PAMI* 23:643–660.
44. Wolf L, Hassner T, Maoz I (2011) Face recognition in unconstrained videos with matched background similarity. *Proceedings of IEEE CVPR 2011*, eds Felzenszwalb P., Forsyth D., and Fua P. (IEEE Computer Society, New York), Vol 1, pp 529–534.
45. Nene SA, Nayar SK, Murase H (1996) Columbia Object Image Library (COIL-100) (Columbia Univ., New York), Technical Report CUCS-006-96.
46. Zhao D, Tang X (2008) Cyclizing clusters via zeta function of a graph. *Advances in Neural Information Processing Systems 21*, eds Koller D, Schuurmans D, Bengio Y (MIT Press, Cambridge, MA), Vol 3, pp 1900–1907.
47. Nie F, Xu D, Tsang IW, Zhang C (2009) Spectral embedded clustering. *Proceedings of the Twenty-First International Joint Conference on Artificial Intelligence*, ed Boutilier C (AAAI Press, Palo Alto, CA), Vol 2, pp 1181–1186.
48. Yang Y, Xu D, Nie F, Yan S, Zhuang Y (2010) Image clustering using local discriminant models and global integration. *IEEE Trans Image Process* 19:2761–2773.
49. Zhang W, Wang X, Zhao D, Tang X (2012) Graph degree linkage: Agglomerative clustering on a directed graph. *Proceedings of the Twelfth European Conference on Computer Vision*, eds Fitzgibbon A, Lazebnik S, Perona P, Sato Y, Schmid C (Springer, Berlin), Vol 1, pp 428–441.
50. Zhang W, Zhao D, Wang X (2013) Agglomerative clustering via maximum incremental path integral. *Pattern Recognition* 46:3056–3065.
51. Strehl A, Ghosh J (2002) Cluster ensembles – A knowledge reuse framework for combining multiple partitions. *J Mach Learn Res* 3:583–617.
52. Vinh NX, Epps J, Bailey J (2010) Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance. *J Mach Learn Res* 11:2837–2854.
53. de Souto MC, Costa IG, de Araujo DS, Ludermir TB, Schliep A (2008) Clustering cancer gene expression data: A comparative study. *BMC Bioinformatics* 9:497.
54. Muja M, Lowe DG (2014) Scalable nearest neighbor algorithms for high dimensional data. *PAMI* 36:2227–2240.
55. Guennebaud G, et al. (2010) Eigen v3.3. Available at eigen.tuxfamily.org. Accessed November 28, 2016.