10546–10562 *Nucleic Acids Research, 2018, Vol. 46, No. 20* *Published online 8 October 2018*
doi: 10.1093/nar/gky889

SURVEY AND SUMMARY

Multi-omic and multi-view clustering algorithms: review and cancer benchmark

Nimrod Rappoport and Ron Shamir*

Blavatnik School of Computer Science, Tel Aviv University, Tel Aviv, Israel

Received July 18, 2018; Revised September 17, 2018; Editorial Decision September 19, 2018; Accepted September 20, 2018

ABSTRACT

Recent high throughput experimental methods have been used to collect large biomedical omics datasets. Clustering of single omic datasets has proven invaluable for biological and medical research. The decreasing cost and development of additional high throughput methods now enable measurement of multi-omic data. Clustering multi-omic data has the potential to reveal further systems-level insights, but raises computational and biological challenges. Here, we review algorithms for multi-omics clustering, and discuss key issues in applying these algorithms. Our review covers methods developed specifically for omic data as well as generic multi-view methods developed in the machine learning community for joint clustering of multiple data types. In addition, using cancer data from TCGA, we perform an extensive benchmark spanning ten different cancer types, providing the first systematic comparison of leading multi-omics and multi-view clustering algorithms. The results highlight key issues regarding the use of single- versus multi-omics, the choice of clustering strategy, the power of generic multi-view methods and the use of approximated p-values for gauging solution quality. Due to the growing use of multi-omics data, we expect these issues to be important for future progress in the field.

INTRODUCTION

Deep sequencing and other high throughput methods measure a large number of molecular parameters in a single experiment. The measured parameters include DNA genome sequence (1), RNA expression (2,3), DNA methylation (4) etc. Each such kind of data is termed 'omic' (genomics, transcriptomics, methylomics, respectively). As costs decrease and technologies mature, larger and more diverse omic datasets are available.

Computational methods are imperative for analyzing such data. One fundamental analysis is clustering - finding coherent groups of samples in the data, such that samples within a group are similar, and samples in different groups are dissimilar (5). This analysis is often the first step done in data exploration. Clustering has many applications for biomedical research, such as discovering modules of co-regulated genes and finding subtypes of diseases in the context of precision medicine (6). Clustering is a highly researched computational problem, investigated by multiple scientific communities, and a myriad algorithms exist for this task.

While clustering each omic separately reveals patterns in the data, integrative clustering using several omics for the same set of samples has the potential to expose more fine-tuned structures that are not revealed by examining only a single data type. For example, cancer subtypes can be defined based on both gene expression and DNA methylation together. There are several reasons why a clustering based on multiple omics is desirable. First, Multi-omics clustering can reduce the effect of experimental and biological noise in the data. Second, different omics can reveal different cellular aspects, such as effects manifest at the genomic and epigenomic levels. Third, even within the same molecular aspect, each omic can contain data that are not present in other omics (e.g. mutation and copy number). Fourth, omics can represent data from different organismal levels, such as gene expression together with microbiome composition.

A problem akin to multi-omics clustering was investigated independently by the machine learning community, and is termed 'multi-view clustering' (see (7) and 'A Survey on Multi-View Clustering'). Multi-view clustering algorithms can be used to perform clustering of multi-omic data. In the past, methods developed within the machine learning community have proven useful in the analysis of biomedical

*To whom correspondence should be addressed. Tel: +972 3 640 5383; Fax: +972 3 640 5384; Email: rshamir@tau.ac.il datasets. However, by and large, multi-view clustering have not penetrated bioinformatics yet.

In this paper, we review methods for multi-omics clustering, and benchmark them on real cancer data. The data source is TCGA (The Cancer Genome Atlas) (8)—a large multi-omic repository of data on thousands of cancer patients. We survey both multi-omics and multi-view methods, with the goal of exposing computational biologists to these algorithms. Throughout this review, we use the terms *view* and *multi-view* instead of omic and multi-omics in the context of Machine Learning algorithms.

Several recent reviews discussed multi-omics integration. (9–11) review methods for multi-omics integration, and (12) review multi-omics clustering for cancer application. These reviews do not include a benchmark, and do not focus on multi-view clustering. (13) reviews only dimension reduction multi-omics methods. To the best of our knowledge, (14) is the only benchmark performed for multi-omics clustering, but it does not include machine learning methods. Furthermore, we believe the methods tested in the benchmark do not represent the current state of the art for multi omics clustering. Finally, (7) is a thorough review of multi-view methods, directed to the Machine Learning community. It does not discuss algorithms developed by the bioinformatics community, and does not cover biological applications.

REVIEW OF MULTI-OMICS CLUSTERING METHODS

We divide the methods into several categories based on their algorithmic approach. *Early integration* is the most simple approach. It concatenates omic matrices to form a single matrix with features from multiple omics, and applies single-omic clustering algorithms on that matrix. In *late integration*, each omic is clustered separately and the clustering solutions are integrated to obtain a single clustering solution. Other approaches try to build a model that incorporates all omics, and are collectively termed *intermediate integration*. Those include: (i) methods that integrate sample similarities, (ii) methods that use joint dimension reduction for the different omics datasets and (iii) methods that use statistical modeling of the data.

The categories we present here are not clear-cut, and some of the algorithms presented fit into more than one category. For example, iCluster (15) is an early integration approach that also uses probabilistic modeling to project the data to a lower dimension. The algorithms are described in the categories where we consider them to fit most.

Multi-omics clustering algorithms can also be distinguishable by the set of omics that they support. *General* algorithms support any kind of omics data, and are therefore easily extendible to novel future omics. *Omic specific* algorithms are tailored to a specific combination of data types, and can therefore utilize known biological relationships (e.g. the correlation between copy number and expression). A mixture of these two approaches is to perform feature learning in an omic specific way, but then cluster those features using general algorithms. For example, one can replace a gene expression omic with an omic that scores expression in cellular pathways, and thus take advantage of existing biological knowledge.

Throughout this review, we use the following notation: a multi-omic dataset contains M omics. n is the number of samples (or patients for medical datasets), p_m is the number of features in the m'th omics, and X^m is the n x p_m matrix with measurements from the m'th omic. X^m_{ij} is therefore the value of the j'th feature for the i'th patient in the m'th omic. $p = \Sigma_{m=1}^{M} p_m$ is the total number of features, and X is the $n \times p$ matrix formed by the concatenation of all X^m matrices. Throughout the paper, for a matrix A, we use A^t to designate its transpose, and consistently with the X^m notation, we use A^m for matrix indexing (and not for matrix powering). Additional notation is chosen to follow the original publications and common conventions.

Figure 1 summarizes pictorially the different approaches to multi-omics clustering. A summary table of the methods reviewed here is given in Table 1.

Early integration

Early integration is an approach that first concatenates all omic matrices, and then applies single-omic clustering algorithms on that concatenated matrix. It therefore enables the use of existing clustering algorithms. However, this approach has several drawbacks. First, without proper normalization, it may give more weight to omics with more features. Second, it does not consider the different distribution of data in the different omics. Finally, it increases the data dimension (the number of features), which is a challenge even in some single-omic datasets. When applying early integration algorithms designed specifically for multi-omics data, or when running single-omic methods on a concatenated matrix, these drawbacks must be addressed. Normalization of the features in different omics can assist in handling the different distributions, and feature selection can be used to decrease the dimension and to give different omics an equal prior opportunity to affect the results.

An additional way to handle the high dimension of the data is by using regularization, i.e. adding additional constraints to a problem to avoid overfitting (76). Specifically, LASSO (Least Absolute Shrinkage and Selection Operator) regularization creates models where the number of features with non-zero effect on the model is low (77), and regularization of the nuclear norm is often used to induce data sparsity. Indeed, LASSO regularization is used by iCluster (15) (reviewed in a later section), and LRACluster uses nuclear norm regularization (reviewed in this section). While any clustering algorithm can be applied using early integration, we highlight here algorithms that were specifically developed for this task.

LRACluster (16) uses a probabilistic model, where numeric, count and binary features have distributions determined by a latent representation of the samples Θ. For example, X^m_{ij} is distributed $\propto exp(-\frac{1}{2}(X^m_{ij} - \Theta^m_{ij})^2)$, where Θ^m is of the same dimensions as X^m. The latent representation matrix is encouraged to be of low rank, by adding a regularization on its nuclear norm. The objective function for the algorithm is $-\log(\text{model'slikelihood}) + \mu \cdot |\Theta|_*$ where Θ is the concatenation of all Θ^m matrices, and $| \cdot |_*$ is the nuclear norm. This objective is convex and provides a global optimal solution, which is found using a fast gradient-ascent algorithm. Θ is subsequently clustered using k-means. This

Table 1. Multi-omic clustering methods

Method	Description	Refs.	Implementation
Early integration			
LRAcluster•	Data originate from low rank matrix, omic data distributions modeled based on it	(16)	R
Structured sparsity	Linear transformation projects data into a cluster membership orthogonal matrix	(17)	Matlab
Alternate optimization			
MV k-means, MV EM	Alternating *k*-means and EM. Each iteration is done w.r.t. a different view	(18)	NA
Late integration			
COCA	Per omic clustering solutions integrated with hierarchical clustering	(19)	NA
Late fusion using latent models	Per omic clustering solutions integrated with PLSA	(20)	NA
PINS•	Integration of co-clustering patterns in different omics. The clusterings are based on perturbations to the data	(21)	R
Similarity-based methods			
Spectral clustering generalizations	Generalizations of similarity based spectral clustering to multi-omics data	(22–25)	Matlab
Spectral clustering with random walks	Generalizations of spectral clustering by random walks across similarity graphs	(26,27)	NA
SNF•	Integration of similarity networks by message passing	(28,29)	R, Matlab
rMKL-LPP•	DR using multiple kernel learning; similarities maintained in lower dimension	(30)	**
Dimension reduction			
General DR framework	General framework for integration with DR	(31)	NA
JIVE	Variation in data partitioned into joint and omic-specific	(32)	Matlab,R (33)
CCA•	DR to axes of max correlation between datasets. Generalizations: Bayesian, kernels, >2 omics, sparse solutions, deep learning, count data	(34–43), CCA for count data	R, two omics (44), R, multiple omics
PLS	DR to axes of max covariance between datasets. Generalizations: kernels, >2 omics, sparse solutions, partition into omic-specific and joint variation	(45–52)	R, two omics, Matlab, multiple omics
MCIA	DR to axes of max covariance between multi-omic datasets	(53)	R
NMF generalizations•	DR using generalizations of NMF to multi-omic data	(54–57), EquiNMF, (58,59)	MultiNMF (Matlab)
Matrix tri- factorization	DR. Each omic describes the relationship between two entities	(60)	NA
Convex methods	DR with convex objective functions, allowing unique optimum and efficient computation	(16,61,62)	Matlab
Low-rank tensor MV clustering	Factorization based on low-rank tensors	(63)	Matlab
Statistical methods			
iCluster/Plus/Bayes•	Data originate from low dimensional representation, which determines the distribution of the observed data	(15,64,65)	R
PARADIGM	Probabilistic model of cellular pathways using factor graphs	(66)	REST API
Disagreement between clusters	Methods based mainly on hierarchical Dirichlet processes; clustering in different omics need not agree	(67–71)	BCC (R)
Survival-based	Probabilistic model; patient survival data used in the clustering process	(72,73)	SBC (R)
Deep learning			
Deep learning methods	Neural networks used for integration. A variant of CCA, early integration and middle integration approaches	(37,74,75)	DeepCCA (Python)

DR: dimension reduction; EM: expectation maximization; MV: multi-view; PLSA: Probabilistic Latent Semantic Analysis; CCA: Canonical Correlation Analysis; PLS: Partial Least Squares; NMF: non-negative matrix factorization. •Methods included in the benchmark. Single-omic *k*-means and spectral clustering were also included in the benchmark. ** Available from the authors upon request.

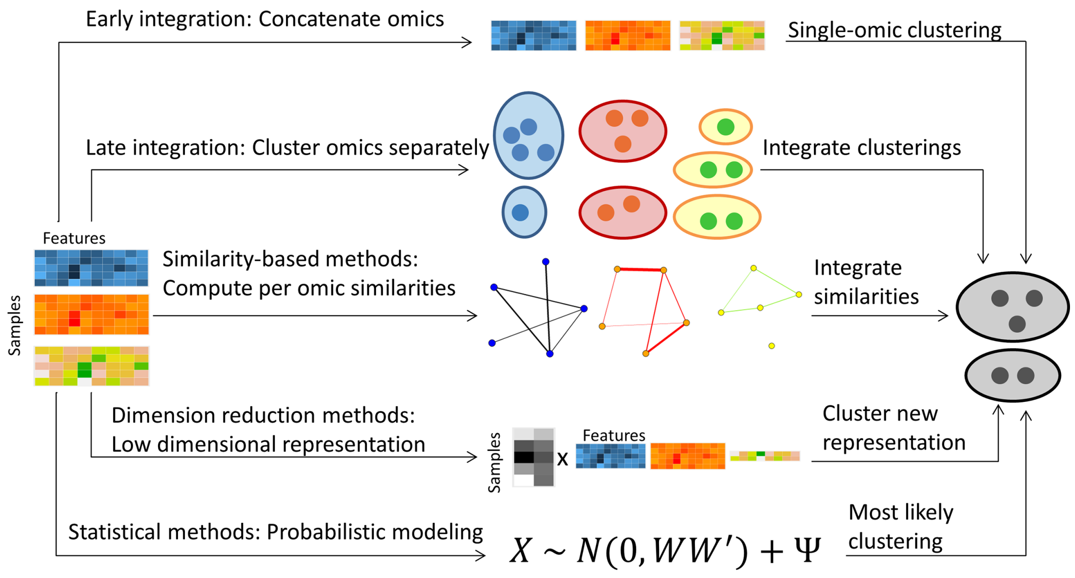

Early integration: Concatenate omics
Single-omic clustering
Late integration: Cluster omics separately
Integrate clusterings
Features
Samples
Similarity-based methods:
Compute per omic similarities
Integrate similarities
Dimension reduction methods:
Low dimensional representation
Samples
x
Features
Cluster new representation
Statistical methods: Probabilistic modeling
$X \sim N(0, WW') + \Psi$
Most likely clustering

Figure 1. Overview of multi-omics clustering approaches.

method was used to analyze pan-cancer TCGA data from eleven cancer types using four different omics, and to further find subtypes within these cancer types.

In (17), all omics are concatenated to a matrix X and the algorithm minimizes the following objective: $||XW + 1_n b^t - F||_2^2 + \gamma||W||_{G_1}$. W is a p x k projection matrix, F is an n x k cluster indicator matrix such that $F^t F = I_k$, 1_n is a column vector of length n of 1's, b is an intercept column vector of dimension k and γ is a scalar. The algorithm therefore seeks a linear transformation such the projected data are as close to a cluster indicator matrix as possible. That indicator matrix is subsequently used for clustering. The regularization term uses the G_1 norm, which is the l_2 norm for W entries associated with a specific cluster and view, summed over all views and clusters. Therefore, features that do not contribute to the structure of a cluster will be assigned with low coefficients in W.

Alternate optimization

Early research for integration of two views was performed in (78). This work improved classification accuracy for semi-supervised data with two views using an approach termed co-training, and inspired others to analyze multi-view data. One of the first attempts to perform multi-view clustering was (18). In this work, EM (expectation maximization) and k-means, which are widely used single-omic clustering algorithms, were adapted for multi-view clustering. Both EM and k-means are iterative algorithms, where each iteration improves the objective function value. The suggested multi-view versions perform optimization in each iteration with respect to a different omic in an alternating manner. This approach loses theoretical guarantees for convergence, but was found to outperform algorithms that use each view separately, and also naive early integration methods that cluster the concatenated matrix of the two views. Interestingly, (18) report improved results using the multi-view clustering algorithms on single-view datasets that were randomly split to simulate multi-view data. This was the first evidence for improved clustering using multiple views, and for the utility of a multi-view algorithm in clustering single-view data. While this work was very influential, other preliminary multi-view clustering methods (e.g. (22,31)) were since shown to achieve better results on datasets where the gold standard is known.

Late integration

Late integration is an approach that allows to use existing single-omic clustering algorithms on single-omic data. First, each omic is clustered separately using a single-omic algorithm. Different algorithms can be used for each omic. Then, the different clusterings are integrated. The strength of late integration lies in that any clustering algorithm can be used for each omic. Algorithms that are known to work well on a particular omic can therefore be used, without having to create a model that unifies all of these algorithms. However, by utilizing only clustering solutions in the integration phase we can lose signals that are weak in each omic separately.

COCA (19) was applied to pan-cancer TCGA data, to investigate how tumors from different tissues cluster, and whether the obtained clusters match the tissue of origin. The algorithm first clusters each omic separately, such that the m'th omic has c_m clusters. The clustering of sample i for omic m is encoded in a binary vector v_{im} of length c_m, where $v_{im}(j) = 1$ if i belongs to cluster j and 0 otherwise. The concatenation of the v_{im} vectors across all omics results in a binary cluster indicator vector for sample i. The $n \times c$ binary matrix B of these indicator vectors, where $c = \Sigma_{i=1}^{M} c_m$, is used as input to consensus clustering (79) to obtain the final clustering of the samples. Alternatively, in (20) a model based on Probabilistic Latent Semantic Analysis (80) was proposed for clustering B. These two methods allow any clustering algorithm to be used on each single omic, and therefore have an advantage when a method is known to perform well for a particular omic. Additionally, they can be used given the clustering solution only when the raw omic data are unavailable.

PINS (21) integrates clusters by examining their connectivity matrices for the different omics. Each such matrix S^m is a binary n x n matrix, where $S^m_{ij} = 1$ if patients i and j are clustered together in omic m, and 0 otherwise. These S^m matrices are averaged to obtain a single connectivity matrix, which is then clustered using different methods based on whether the different S^m matrices highly agree with each other or not. The obtained clusters are tested if they can be further split into smaller clusters. To determine the number of clusters for each omic and for the integrated clustering, perturbations are performed on the data by adding Gaussian noise to it, and the number of clusters is chosen such that the resulting clustering is robust to the perturbations. Unlike the previously presented late integration methods, PINS requires the original data and not only the clustering of each omic, since it performs perturbations to the data.

Several methods for ensemble clustering were developed over the years, and are reviewed in (81). While these were not originally developed for this purpose, they can be used for late multi-omics clustering as well.

Similarity-based methods

Similarity-based methods use similarities or distances between samples in order to cluster data. These methods compute the similarities between samples in each omic separately, and vary in the way these similarities are integrated. The integration step uses only similarity values. Since in current multi-omic datasets, the number of samples is much smaller than the number of features, these algorithms are usually faster than methods that consider all features while performing integration. However, in such methods it may be more difficult to interpret the output in terms of the original features. An additional advantage of similarity-based methods is that they can easily support diverse omic types, including categorical and ordinal data. Each omic only requires a definition of a similarity measure.

Spectral clustering generalizations. Spectral clustering (82) is a widely used similarity-based method for clustering single-view data. The objective function for single-view spectral clustering is $max_U trace(U^t LU)$ s.t. $U^t U = I$, where L is the Laplacian (83) of the similarity matrix of dimension $n \times n$, and U is of dimension $n \times k$, where k is the number of clusters in the data. Intuitively, it means that samples that are similar to one another have similar row vectors in U. This problem is solved by taking the k first eigenvectors of L (details vary between versions that use the normalized and the unnormalized graph Laplacian), and clustering them with a simple algorithm such as k-means. The spectral clustering objective was shown to be a relaxation of the discrete normalized cut in a graph, providing an intuitive explanation for the clustering. Several multi-view clustering algorithms are generalizations of spectral clustering.

An early extension to two views performs clustering by computing a new similarity matrix, using the two views' similarities (22). Denote by W_1 and W_2 the similarity matrices for the two views. Then the integrated similarity, W, is defined as $W_1 W_2$. Spectral clustering is performed on the block matrix

$$\begin{bmatrix} 0 & W \\ W^t & 0 \end{bmatrix}$$

Note that each eigenvector for this matrix is of length $2n$. Either half of the vector or an average of the two halves are used instead of the whole eigenvectors for clustering using k-means. Note that this method is limited in that it only supports two views.

(23) generalizes spectral clustering for more than two views. Instead of finding a global U matrix, a matrix U^m is defined for each omic. The optimization problem is:

$$max_{U^1,\ldots,U^M} \Sigma_m trace(U^{mt} L^m U^m)$$

$$+\lambda \cdot \text{Reg} \quad \text{s.t.}\ \forall m\ U^{mt} U^m = I.$$

L^m is the graph Laplacian for omic m and Reg is a regularization term equal to either $\Sigma_{m_1 \neq m_2} U^{m_1} U^{m_1 t} U^{m_2} U^{m_2 t}$ or $\Sigma_m U^m U^{mt} U^* U^{*t}$ with the additional constraint that U^* is an n x k matrix such that $U^{*t} U^* = I$.

Chikhi (24) uses a different formulation, which does not require a different U^m for each omic, but instead uses the same U for all matrices. The following objective function is used:

$$max_U \Sigma_m trace(U^t L^m U) \text{ s.t. } U^t U = I$$

This is equivalent to performing spectral clustering on the Laplacian $\Sigma_m L^m$. The obtained clusters are then further improved in a greedy manner, by changing the assignment of samples to clusters, while looking directly at the discrete normalized cut objective, rather than the continuous spectral clustering objective.

Li (25) suggests a runtime improvement over (23). Instead of looking at the similarity matrix for all the samples, a small set of 'representative' vectors, termed salient points, are calculated by running k-means on the concatenation of all omics and selecting the cluster centers. A similarity matrix is then computed between all these samples in the data and their s nearest salient points. Denote this similarity matrix for the m'th omic by W^m, and let Z^m be its normalization such that rows sum to 1. These matrices are of dimension $n \times$ the number of salient points. Next, the matrices

$$\begin{bmatrix} 0 & Z^m \\ Z^{mt} & 0 \end{bmatrix}$$

are given as input to an algorithm with the same objective as (24). This way, similarities are not computed between all pairs of samples.

The methods above differ in several ways. (23) allows each omic to have a different low dimensional representation, and has a parameter that controls the trade-off between how similar these representations are, and how similarities in the original data are maintained in U^m. Therefore, it allows to express cases where the omics are not assumed to have the same similarity structure (e.g., two samples can be similar in one omic but different in another). On the other hand, Chikhi (24) assumes the same similarity structure, and its greedy optimization step can result in an improved solution in such cases. (25) can be used when the number of samples is exceptionally large.

Zhou and Burges (26) views similarity matrices as networks, and examines random walks on these networks. Random walks define a stationary distribution on each network, which captures its similarity patterns (84). Since that stationary distribution is less noisy than the original similarity measures, Zhou and Burges (26) uses them instead to integrate the networks. Xia (27) also examines random walks on the networks, but argues that the stationary distribution in each network can still be noisy. Instead, the authors compute a consensus transition matrix, that has minimum total distance to the per-omic transition matrices and is of minimal rank. Random walks are highly related to spectral clustering; using a normalized variant of the graph's Laplacian in spectral clustering results in a solution in which random walks seldom cross between clusters (82). These random walk-based methods are currently competitive with other spectral clustering methods.

Similarity Network Fusion. SNF (Similarity Network Fusion) first constructs a similarity network for every omic separately (28,29). In each such network, the nodes are samples, and the edge weights measure the sample similarity. The networks are then fused together using an iterative procedure based on message passing (85). The similarity between samples is propagated between each node and its k nearest neighbors.

More formally, denote by $W^{(m)}$ the similarity matrix for the m'th omic. Initially a transition probability matrix between all samples is defined by:

$$P_1^{(m)}(i,\, j) = \begin{cases} \frac{W^{(m)}(i,j)}{2\Sigma_{k\neq i} W^{(m)}(i,k)}, & j \neq i \\ \frac{1}{2}, & j = i \end{cases}$$

and a transition porbability matrix between nearest neighbors is defined by:

$$S^{(m)}(i,\, j) = \begin{cases} \frac{W^{(m)}(i,j)}{\Sigma_{k\neq i} W^{(m)}(i,k)}, & j \in N_i \\ 0, & \text{otherwise} \end{cases}$$

where N_i are i's k nearest neighbors in the input X^m matrices. The P matrices are updated iteratively using message passing between the nearest neighbors: $P_{q+1}^{(m)} = S^{(m)} \frac{\Sigma_{k\neq m} P_q^{(k)}}{M-1} S^{(m)q}$ where $P_q^{(m)}$ is the matrix for omic m at iteration q. This process converges to a single similarity network, summarizing the similarity between samples across all omics. This network is partitioned using spectral clustering.

In (29), SNF is used on gene expression, methylation and miRNA expression data for several cancer subtypes from TCGA. In addition to partitioning the graph to obtain cancer sutbypes, the authors show that the fused network can be used for other computational tasks. For example, they show how to fit Cox proportional hazards (86), a model that predicts prognosis of patients, with a constraint such that similar patients in the integrated network will have similar predicted prognosis.

rMKL-LPP. Kernel functions implicitly map samples to a high (possibly infinite) dimension, and can efficiently measure similarity between the samples in that dimension. Multiple kernel learning uses several kernels (similarity measures), usually by linearly combining them, and is often used in supervised analysis. (30) developed rMKL-LPP (regularized Multiple Kernel Learning with Locality Preserving Projections), which uses multiple kernel learning in unsupervised settings. The algorithm performs dimension reduction on the input omics such that similarities (defined using multiple kernels) between each sample and its nearest neighbors are maintained in low dimension. This representation is subsequently clustered with k-means. rMKL-LPP allows the use of diverse kernel functions, and even multiple kernels per omic. A regularization term is added to the optimization problem to avoid overfitting. The authors run the algorithm on five cancer types from TCGA, and show that using multiple kernels per omic improves the prognostic value of the clustering, and that regularization improves robustness.

Dimension reduction-based methods

Dimension reduction-based methods assume the data have an intrinsic low dimensional representation, with that low dimension often corresponding to the number of clusters. The views that we observe are all transformations of that low dimensional data to a higher dimension, and the parameters for the transformation differ between views. This general formulation was proposed by (31), which suggest to minimize $\Sigma_{m=1}^{M} w_m l(X^m,\, f_m(B))$, where B is a matrix of dimension $n \times\ p$, f_m are the parametrized transformations, and w_m are weights for the different views, and l is a loss function. The work further provides an optimization algorithm when the f_m transformations are given by matrix multiplication. That is, $f_m(B) = BP^m$, and l is the squared Frobenius norm applied to $X^m - BP^m$. Once B is calculated, single-omic clustering algorithm can be applied to it. This general framework is widely used. Since the transformation is often assumed to be linear, many of the dimension reduction methods are based on matrix factorization. Dimension reduction methods work with real-valued data. Applying these methods to discrete binary or count data is technically possible but often inappropriate.

An advantage of linear dimension reduction methods is that they provide some interpretation for the dominant features in each cluster. For example, in the general framework just presented, each entry in the P^m matrix can be considered as the weight of a feature in a cluster. Such interpretation is missing from similarity-based methods, which ignore the original features once the similarities between samples were calculated. Therefore, dimension reduction methods may be useful when an association between clusters and features is needed.

JIVE. (32) assumes that the variation in each omic can be partitioned to a variation that is joint between all omics, and an omic-specific variation: $X^{m^t} = J^m + A^m + E^m$ where E^m are error terms. Let J and A be the concatenated J^m and A^m matrices, respectively. The model assumes that $JA^t = 0$, that is, the joint and omic specific variations are uncorrelated, and that $rank(J) = r$ and $rank(A_i) = r_i$ for each omic, so that the structure of each omic and the total joint variation are of low rank. In order for the weight of the different omics to be equal, the input omic matrices are normalized to have equal Frobenius norm. A penalty term is added to encourage variable sparsity. This method was applied to gene expression and miRNA data of Glioblastoma Multiforme brain tumors, and identified the joint variation between these omics.

Correlation and covariance-based. Two of the most widely used dimension reduction methods are Canonical Correlation Analysis (CCA) (34) and Partial Least Squares (PLS) (45). Given two omics X^1 and X^2, in CCA the goal is to find two projection vectors u^1 and u^2 of dimensions p_1 and p_2, such that the projected data has maximum correlation:

$$argmax_{u^1,u^2} corr(X^1u^1, X^2u^2)$$

These projections are called the first canonical variates, and are the axis with maximal correlation between the omics. The k'th pair of canonical variates, u_k^1 and u_k^2 are found such that correlation between $X^1u_k^1$ and $X^2u_k^2$ is maximal, given that the new pair is uncorrelated (that is, orthogonal) to the previous canonical variates. Chaudhuri *et al.* (87) proved and showed empirically that if the data originate from normal or log concave distributions, the canonical variates can be used to cluster the data. CCA was formulated in a probabilistic framework such that the optimization solutions are maximum likelihood estimates (88), and further extended to a Bayesian framework (35). An additional expansion to perform CCA in high dimension is Kernel CCA (36). A deep-learning based CCA method, DeepCCA, was recently developed (37). Rather than maximize the correlation between linear projections of the data, the projections are taken to be functions of the data calculated using neural networks, and the optimization process optimizes the parameters for these networks.

Solving CCA requires inversion of the covariance matrix for the two omics. Omics data usually have a higher number of features than samples, and these matrices are therefore not invertible. To apply CCA to omics data, and to increase the interpretability of CCA's results, sparsity regularization was added (38,39).

CCA supports only two views. Several works extend it to more than two views, including MCCA (39) which maximizes the sum of pairwise correlations between projections and CCA-RLS (40). Luo *et al.* (41) generalize CCA to tensors in order to support more than two views.

Another line of work on CCA, with high relevance for omics data, investigated relationships between the features while performing the dimension reduction. ssCCA (structure constrained sparse CCA) allows to incorporate into the model known relationships between features in one of the input omics, and force entries in the u^i vector for that view to be close for similar features. This model has been developed by (42) and utilized microbiome's phylogenies as the feature structure. Another model that considers relationship between features was developed in (43). In this work, rather than defining similarities between features, they are partitioned into groups. Regularization is performed such that both irrelevant groups and irrelevant features within relevant groups are removed from the model. Finally, Podosinnikova et. al, in 'Beyond CCA: Moment matching for multi-view models', extended CCA to support count data, which are common in biological datasets.

PLS also follows a linear dimension reduction model, but maximizes the covariance between the projections, rather than the correlation. More formally, given two omics X^1 and X^2, PLS computes a sequence of vectors u_k^1 and u_k^2 for $k = 1, 2, \ldots$ such that $cov(X^1u_k^1, X^2u_k^2)$ is maximal, given that ${u_k^1}^t u_k^1 = 1$, ${u_k^2}^t u_k^2 = 1$, and $cor(X^1u_k^1, X^1u_l^1) = 0$ for $l < k$. That is, new projections are not correlated with previous ones. PLS can be applied to data with more features than samples even without sparsity constraints. A sparse solution is nonetheless desirable, and one was developed (46,47). O2-PLS increases the interpretability of PLS by partitioning the variation in the datasets into joint variation between them, and variations that are specific for each dataset and that are not correlated with one another (48). While PLS and O2-PLS were originally developed for chemometrics, they were recently used for omics data as well (89,90). PLS was also extended to use the kernel framework (49), and a combined version of kernel PLS and O2 PLS was developed (50).

Like CCA, PLS was developed for two omics. MBPLS (Multi Block PLS) extends the model to more than two omics (91), and sMBPLS adds sparsity constraints. sMBPLS was developed specifically for omics data (51). It looks for a linear combination of projections of non-gene-expression omics that has maximal correlation with a projection of gene expression omic. An extension of O2PLS also exists for multi-view datasets (52).

Both CCA and PLS can be used in cases where high interpretability is wanted. The different u_k^1 and u_k^2 vector pairs are those along which the correlation (or covariance) between patients is maximal. They can therefore be used to associate between features from the different views.

An additional method that is based on maximizing covariance in low dimension is MCIA (53), an extension of co-inertia analysis to more than two omics (92). It aims to find projections for all the omics such that the sum of squared covariances with a global variation axis is maximal: $max_{u^m,v} \Sigma_{m=1}^{M} cov^2(X^m u^m, v)$. The projections of different omics can be used to evaluate the agreement between the different omics (the distance between projections reflects the level of disagreement between omics). Each of the projections can be used as a representation for clustering.

Non-negative Matrix Factorization. Non-negative Matrix Factorization (NMF) assumes that the data have an intrinsic low dimensional non-negative representation, and that a nonnegative matrix projects it to the observed omic (93). It is therefore only suitable for non-negative data. For a single omic, denote by k the low dimension. The formulation is $X \approx WH$, where X is the $n \times p$ observed omic matrix, W is $n \times k$ and H is $k \times p$. The objective function is $||X - WH||_2^2$, and it is minimized by updating W and H in an alternating manner, using multiplicative update rules, such that solutions remain non negative after each update (94). The low dimension representation W can be clustered using a simple single-omic algorithm. Like other dimension reduction methods, the W and H matrices can be used to better understand the weight of each feature in each cluster. The non-negativity constraint makes this weight more interpretable.

Several methods generalize this model to multi-omic data. MultiNMF (54) uses the following generalization: Each omic X^m is factorized into $W^m H^m$. This model is equivalent to performing NMF on each omic separately. Integration between the omics is done by adding a constraint that the W^m matrices are close to a 'consensus' matrix W^*. The objective function is therefore: $\Sigma_{m=1}^{M} || X^m - W^m H^m ||_2^2 + \lambda \Sigma_{m=1}^{M} || W^m - W^* ||_2^2$. Kalayeh *et al.* (55) generalizes this method to support weights for features' and samples' similarity. (56) extend MultiNMF by further requiring that the low dimensional representation W^* maintains similarities between samples (samples that are close in the original dimension must be close in W^*). This approach combines factorization and similarity-based methods.

Joint NMF (57) uses a different formulation, where a sample has the same low dimensional representation for all omics: $X^m \approx W H^m$. Note that by writing $X = WH$ where X and H are obtained by matrix concatenation, this model is equivalent to early integration. Joint NMF is not directly used for clustering. Rather, the data are reduced to a large dimension ($k = 200$) and high values in W and H^m are used to associate samples and features with modules that are termed 'md-modules'. The authors applied Joint NMF on miRNA, gene expression and methylation data from ovarian cancer patients, and showed that functional enrichment among features that are associated with md-modules that is more significant than the enrichment obtained in single-omic modules. In addition, patients in certain modules have significantly different prognosis compared to the rest of the patients. Much like (56) extends multiNMF, EquiNMF extends Joint NMF such that similarities in the original omics are maintained in lower dimension. (58) extends NMF to the case where different views can contain different samples, but constrains certain samples from different views to belong to the same cluster based on prior knowledge. Finally, PVC (59) performs partial multi-view clustering. In this setting, not all samples necessarily have measurements for all views.

The difference between MultiNMf and Joint NMF resembles the difference described previously between similarity-based methods. MultiNMF allows for different omics to have different representations, where the similarity between them is controlled by a parameter. It can therefore be used in cases where the different omics are not expected to have the same low dimensional representation.

Matrix tri-factorization. An alternative factorization approach presented in (60) is tri-matrix factorization. In this framework, each input omic is viewed as describing a relationship between two entities, which are its rows and columns. For example, in a dataset with two omics, gene expression and DNA methylation of patients, there are three entities which are the patients, the genes and the CpG loci. The gene expression matrix describes a relationship between patients and genes, while the methylation matrix describes a relationship between patients and CpG loci.

Each omic matrix R_{ij} of dimension $n_i \times n_j$ that describes the relationship between entities i and j is factorized as $R_{ij} = G_i S_{ij} G^t_j$, where G_i and G_j provide a low dimensional representation for entities i and j respectively and are of dimensions $n_i \times k_i$ and $n_j \times k_j$, and S_{ij} is an omic-specific matrix of dimension $k_i \times k_j$. As in NMF, the G_i matrices are non-negative. The same G_i matrix is used in all omics with entity i, and in this way data integration is achieved. In the above example, both the gene expression and DNA methylation omics will use the same G matrix to represent patients, but different matrices to represent genes and CpG loci. In this model, an additional matrix describing the relationship between genes and CpGs could optionally be used. This is a major advantage of matrix tri-factorization, as it allows to incorporate prior known relations between different entities, without changing the input omic matrices. (60) adds constraints to the formulation that can encourage entities to have similar representations. This framework was applied to diverse problems in bioinformatics, including in supervised settings: It was used to perform gene function prediction (60), and for patient survival regression (95).

Convex formulations. A drawback of most factorization-based methods is that their objective functions are not convex, and therefore optimization procedures do not necessarily reach a global optimum, and highly depend on initialization. One solution to this issue is by formulating dimension reduction as a convex problem. White *et al.* (61) relaxes CCA's conditions and defines a convex variant of it. Performance was assessed on reducing noise in images, but the method can also be used for clustering. However, like CCA, the method only supports two views. Guo (62) presents a different convex formulation for dimension reduction, for the general factorization framework presented earlier, which minimizes $\Sigma_{m=1}^{M} || X^m - BP^m ||_F^2 + \gamma || B ||_{2,1}$. $|| \cdot ||_{2,1}$ is the $l_{2,1}$ norm, namely the sum of the Euclidean norms of the matrix rows. This relaxation therefore supports multiple views. LRAcluster (16) also uses matrix factorization and has a convex objective function.

Tensor-based methods. A natural extension of factorization methods for multi-omic data is to use tensors, which are higher order matrices. One such method is developed in (63). This method writes each omic matrix as $X^m = Z^m X^m + E^m$, $diag(Z^m) = 0$, where Z^m is an n x n matrix and E^m are error matrices. The idea is that each sample in each omic can be represented as a linear combination of other samples (hence the $diag(Z^m) = 0$ constraint), and that its representation in that base (Z^m) can then be used for clustering. To integrate the different views, the different Z^m matrices are merged to a third-order tensor, Z. The objective function encourages Z to be sparse, and the E^m error matrices to have a small norm.

Statistical methods

Statistical methods model the probabilistic distribution of the data. Some of these methods view samples as originating from different clusters, where each cluster defines a distribution for the data, while other methods do not explicitly use the cluster structure in the model. An advantage of the statistical approach is that it allows to include biological knowledge as part of the model when determining the distribution functions. This can be done either using Bayesian priors or by choosing probabilistic functions, e.g.

using normal distribution for gene expression data. An additional advantage of statistical frameworks is their ability to make 'soft', probabilistic decisions. For example, a statistical method can not only assign a sample to a cluster, but can also determine the probability that the sample belongs to the cluster. For most formulations, parameter estimation is computationally hard, and different heuristics are used. Several models under the Bayesian framework allow for samples to belong to different clusters in different omics.

iCluster and iCluster+. iCluster ([15](#)) assumes that the data originate from a low dimension representation, which determines the cluster membership for each sample: $X^{m^t} = W^m Z + \epsilon^m$, where Z is a k x n matrix, W^m is an omic specific p_m x k matrix, k is the number of clusters and ϵ^m is a normally distributed noise matrix. This model resembles other dimension reduction models, but here the distribution of noise is made explicit. Under this model iCluster maximizes the likelihood of the observed data with an additional regularization for sparse W^m matrices. Optimization is performed using an EM-like algorithm, and subsequently k-means is run on the lower dimension representation of the data Z to get the final clustering assignments. iCluster was applied to breast and lung cancer, using gene expression and copy number variations. iCluster was also recently used to cluster more than ten thousand tumors from 33 cancers in a pan-cancer analysis (96). Note that by concatenating all W^m matrices to a single W matrix, and rewriting the model as $X^t = WZ + \epsilon$, iCluster can be viewed as an early integration approach.

iCluster's runtime grows fast with the number of features, and therefore feature selection is essential before using it, as was shown in (29). Shen *et al.* (15) only use genes located on one or two chromosomes in their analysis.

Since iCluster's model uses matrix multiplication, it requires real-values features. An extension called iCluster+ (64) includes different models for numeric, categorical and count data, but maintains the idea that data originate from a low dimension matrix Z. For categorical data, iCluster+ assumes the following model:

$$Pr(X_{ij}^m = c|z_i) = \frac{exp(\alpha_{jcm} + \beta_{jcm} \cdot z_i)}{\Sigma_l exp(\alpha_{jlm} + \beta_{jlm} \cdot z_i)}$$

while for numeric data the model remains linear with normal error:

$$x_{ijm} = \gamma_{jm} + \delta_{jm} \cdot z_i + \epsilon_{ijm}, \epsilon_{ijm} \sim N(0, \sigma_{jm}^2)$$

A regularization term encouraging sparse solution is added to the likelihood, and a Monte-Carlo Newton–Raphson algorithm is used to estimate parameters. The Z matrix is used as in iCluster for the clustering. The latest extension of iCluster, which builds on iCluster+, is iClusterBayes (65). This method replaces the regularization in iCluster+ with full Bayesian regularization. This replacement results in faster execution, since the algorithm no longer needs to fine tune parameters for iCluster+'s regularization.

PARADIGM. PARADIGM (66) is the most explicit approach to modeling cellular processes and the relations among different omics. For each sample and each cellular pathway, a factor graph that represents the state of different entities within that pathway is created. As a degenerate example, a pathway may include nodes representing the mRNA levels of each gene in that pathway, and nodes representing those genes' copy number. Each node in the factor graph can be either activated, nominal or deactivated, and the factor graph structure defines a distribution over these activation levels. For example, if a gene has high copy number it is more likely that it will be highly expressed. However, if a repressor for that gene is highly expressed, that gene is more likely to be deactivated. PARADIGM infers the activity of non-measured cellular entities to maximize the likelihood of the factor graph, and outputs an activity score for each entity per patient. These scores are used to cluster cancer patients from several tissues.

PARADIGM's model can be used for more than clustering. For example, PARADIGM-shift (97) predicts loss-of-function and gain-of-function mutations, by finding genes whose expression value as predicted based on upstream entities in the factor graph is different from their predicted expression value using downstream entities. However, PARADIGM relies heavily on known interactions, and requires specific modeling for each omic. It is also quite limited to the cellular level; For example, it is not clear how to incorporate into the model an omic describing the microbiome composition of each patient.

Combining omic-specific and global clustering. All the methods discussed so far assume that there exists a consistent clustering structure across the different omics, and that analyzing the clusters in an integrative way will reveal this structure more accurately than analyzing each omic separately. However, this is not necessarily the case for biomedical datasets. For example, it is not clear that the methylation and expression profiles of cancer tumors really represent the same underlying cluster structure. Rather, it is possible that each omic represents a somewhat different cluster structure. Several methods take this view point using Bayesian statistics.

Savage *et al.* (67) define a hierarchical Dirichlet process model, which supports clustering on two omics. Each sample can be either *fused* or *unfused*. Fused samples belong to the same cluster in both omics, while unfused samples can belong to different clusters in different omics. Patterns of fused and unfused samples reveal the concordance between the two datasets. This model is extended in PSDF (68) to include feature selection. Savage *et al.* (67) apply the model to cluster genes using gene expression and ChIP-chip data, while (68) clusters cancer patients using expression and copy number data.

In MDI (69) each sample can have different cluster assignments in different omics. However, a prior is given such that the stronger an association between two omics is, the more likely a sample will belong to the same cluster in these two omics. This association strength adjusts the prior clustering agreement between two omics. In addition to these priors, MDI's model uses Dirichlet mixture model, and explicitly represents the distribution of the data within each cluster and omic. Since samples can belong to different clusters in different omics, no global clustering solution is returned by the algorithm. Instead, the algorithm outputs sets of samples that tend to belong to the same cluster.

A different Bayesian formulation is given by BCC (70). Like MDI, BCC assumes a Dirichlet mixture model, where the data originate from a mixture of distributions. However, BCC does assume a global clustering solution, where each sample maps to a single cluster. Given that a sample belongs to a global cluster, its probability to belong to that cluster in each omic is high, but it can also belong to a different cluster in that omic. Parameters are estimated using Gibbs sampling (98). BCC was used on gene expression, DNA methylation, miRNA expression and RPPA data for breast cancer from TCGA.

Like MDI and BCC, Clusternomics (71) uses a Dirichlet mixture model. Clusternomics suggests two different formulations. In the first, each omic has a different clustering solution, and the global clusters are represented as the Cartesian product of clusters from each omic. This approach does not perform integration of the multi-omic datasets. In the second formulation, global clusters are explicitly mapped to omic-specific clusters. That way, not all possible combinations of clusters from different omics are considered as global clusters.

Survival-based clustering. One of the areas multi-omics clustering is widely used for is discovering disease subtypes. In this context, we may expect different disease subtypes to have a different prognosis, and this criterion is often used to assess clustering solutions. Ahmad and Fröhlich (72) develop a Bayesian model for multi-omics clustering that considers patient prognosis while clustering the data. Patients within a cluster have both similar feature distribution and similar prognosis. This approach is not entirely unsupervised, as it considers patient survival data, which are also used to assess the solutions. Coretto *et al.* (73) also develop a probabilistic clustering method that considers survival, and that supports a large number of features compared to (72), which only uses a few dozen features. As the survival data are used as input to the model, it is not surprising that this approach gives clusters with more significantly different survival than other approaches. This was demonstrated on Glioblastoma Multiforme data by (72) and for data from several cancer types by (73), both from TCGA.

Deep multi-view methods

A recent development in machine learning is the advent of deep learning algorithms (99). These algorithms use multi-layered neural networks to perform diverse computational tasks, and were found to improve performance in several fields such as image recognition (100) and text translation (101). Neural networks and deep learning have also proven useful for multi-view applications (102), including unsupervised feature learning (37), (103). Learned features can be used for clustering, as described earlier for DeepCCA. Deep learning is already used extensively for biomedical data analysis (104).

Recent deep learning uses for multi-omics data include (74) and (75). Chaudhary *et al.* (74) use an autoencoder, which is a deep learning method for dimension reduction. The authors ran it on RNA-seq, methylation and miRNA-seq data in order to cluster Hapatocellular Carcinoma patients. The architecture implements an early integration approach, concatenating the features from the different omics. The autoencoder outputs a representation for each patient. Features from this representation are tested for association with survival, and significantly associated features are used to cluster the patients. The clusters obtained have significantly different survival. This result is compared to a similar analysis using the original features, and features learned with PCA (Principal Component Analysis) rather than autoencoders. However, the analysis in this work is not unsupervised, since the feature selection is based on patient survival.

Liang *et al.* (75) use a different approach. They analyze expression, methylation and miRNA ovarian cancer data using Deep Belief Networks (105) which explicitly consider the multi-omic structure of the data. The architecture contains separate hidden layers, each having inputs from one omic, followed by layers that receive input from all the single-omic hidden layers, thus integrating the different omics. A 3D representation over $\{0, 1\}$ is learned for each patient, partitioning the patients into $2^3 = 8$ clusters. The clustering results are compared to k-means clustering on the concatenation of all omics, but not to other multi-omics clustering methods.

Deep learning algorithms usually require many samples and few features. They use a large number of parameters, which makes them prone to overfitting. Current multi-omic datasets have the opposite characteristics—they have many features and at least one order of magnitude less samples. The works presented here use only a few layers in their architectures to overcome this limitation, in comparison to the dozens of layers used by state-of-the-art architectures for imaging datasets. As the number of biomedical samples increases, deep multi-view learning algorithms might prove more beneficial for biomedical datasets.

BENCHMARK

In order to test the performance of multi-omics clustering methods, we compared nine algorithms on ten cancer types available from TCGA. We also compared the performance of the algorithms on each one of the single-omic datasets that make up the multi-omic datasets, for algorithms that are applicable to single-omic data. The nine algorithms were chosen to represent diverse approaches to multi-omics clustering. Within each approach, we chose methods with available software and clear usage guidelines (e.g. we chose PINS over COCA as a late integration method since COCA does not explicitly state how each single omic should be clustered), and that are widely used, so that a comparison of these methods will be most informative to the community. Three algorithms are early integration methods: LRAcluster, and k-means and spectral clustering on the omics concatenated into a single matrix. For similarity-based algorithms we used SNF and rMKL-LPP. For dimension reduction we used MCCA (39) and MultiNMF. We chose iClusterBayes as a statistical method, and PINS as a late integration approach.

The ten datasets contain cancer tumor multi-omics data, where each dataset is a different cancer type. All datasets contain three omics: gene expression, DNA methylation and miRNA expression. The number of patients range from 170 for AML to 621 for BIC. Full details on the datasets and cancer type acronyms appear in Supplementary File 2.

To assess the performance of a clustering solution, we used three metrics. First, we measured differential survival between the obtained clusters using the logrank test (106). Using this test as a metric assumes that if clusters of patients have significantly different survival, they are different in a biologically meaningful way. Second, we tested for the enrichment of clinical labels in the clusters. We chose six clinical labels for which we tested enrichment: gender, age at diagnosis, pathologic T, pathologic M, pathologic N and pathologic stage. The four latter parameters are discrete pathological parameters, measuring the progression of the tumor (T), metastases (M) and cancer in lymph nodes (N), and the total progression (pathologic stage). Enrichment for discrete parameters was calculated using the χ^2 test for independence, and for numeric parameters using Kruskal-Wallis test. Not all clinical parameters were available for all cancer types, so a total of 41 clinical parameters were available for testing. Finally, we recorded the runtime of each method. We did not consider in the assessment computational measures for clustering quality, such as heterogeneity, homogeneity or the silhouette score (107), since the different methods perform different normalization on the features (and some even perform feature selection). Full details about the survival and phenotype data appear in Supplementary File 2.

To derive a p-value for the logrank test, the χ^2 test for independence, and the Kruskal-Wallis test, the statistic for these three tests is assumed to have χ^2 distribution. However, for the logrank test and χ^2 test this approximation is not accurate for small sample sizes and unbalanced cluster sizes, especially for large values of the test statistic (this was shown for example in (108) for the logrank test in the case of two clusters). The p-values we report here are therefore estimated using permutation tests (i.e., we permuted the cluster labels between samples and used the test statistic to obtain an empirical p-value). We indeed observed large differences between the p-values based on permutation testing and based on the approximation, for both the logrank test and enrichment of clinical parameters. More details on the permutation tests appear in Supplementary File 1. After permutation testing, the p-values for the clinical labels were corrected for multiple hypotheses (since several labels were tested) using Bonferroni correction for each cancer type and method at significance level 0.05. Results for the statistical analyses are in Supplementary File 3.

We applied all nine methods to the ten multi-omics datasets, and to the thirty single-omic matrices comprising them. The only exceptions were MCCA, which we could not apply to single-omic data, and PINS, which crashed consistently on all BIC datasets[*]. All methods were run on a Windows machine, except for iCluster which was run on a Linux cluster utilizing up to 15 nodes in parallel. In general, we chose parameters for the methods as suggested by the authors. In case the authors suggested a parameter search, such search was performed, and the best solution was chosen as suggested by the authors, without considering the survival and clinical parameters that are used for assessment. The runtime we report for the methods includes the parameter search. The rationale is that the benchmark aims to record how a user would run the methods in terms of both results quality and total runtime. Details on hardware, data preprocessing and application of the methods appear in Supplementary File 1. Full clustering results appear in Supplementary File 4. All the processed raw data are available at http://acgt.cs.tau.ac.il/multi_omic_benchmark/download.html, and all software scripts used are available at https://github.com/Shamir-Lab/Multi-Omics-Cancer-Benchmark/.

Figure 2 depicts the performance of the benchmarked methods on the different cancer datasets, and Figures 3 and 4 summarize the performance for multi-omics data and for each single-omic separately across all cancer types. No algorithm consistently outperformed all others in either differential survival or enriched clinical parameters. With respect to survival, MCCA had the total best prognostic value (sum of -log10 p-values = 17.53), while MultiNMF was second (16.07) and LRACluster third (15.72). The sum of p-values can be biased due to outliers, so we also counted the number of datasets for which a method's solution obtains significantly different survival. These results are reported in Table 2. Here, with the exception of iClusterBayes, all methods that were developed for multi-omics or multi-view data had at least four cancer types with significantly different survival. MCCA and LRACluster had five. These cancer types are not identical for all the algorithms.

rMKL-LPP achieved the highest total number of significant clinical parameters, with 16 parameters. Spectral clustering came second with 14 and LRAcluster had 13. MCCA and MultiNMF, which had good results with respect to survival, had only 12 and 10 enriched parameters, respectively. rMKL-LPP did not outperform all other methods for all cancer types. For example, it had one enriched parameter for SKCM, while several other methods had two or three. We also considered the number of cancer types for which an algorithm had at least one enriched clinical label (Table 2). rMKL-LPP, spectral clustering, LRACluster and MCCA had enrichment in 8 cancer types, despite MCCA having a total of only 12 enriched parameters. Overall, rMKL-LPP outperformed all methods except MCCA, LRACluster and multiNMF with respect to both survival and clinical enrichment. MCCA, LRACluster and multiNMF had better prognostic value, but found less enriched clinical labels.

Each method determines the number of clusters for each dataset. These numbers are presented in Table 3. The numbers vary drastically among methods, from 2 or 3 (iCluster and MultiNMF) to more than 10 on average (MCCA). MCCA, LRACluster and rMKL-LPP partitioned the data into a relatively high number of clusters (average of 10.6, 9.4 and 6.7 respectively), and had good performance, which may indicate that clustering cancer patients into more clusters improves prognostic value and clinical significance. The

[*] Correction after publication: We performed all the benchmarks on a 64-bit computer, using the 32-bit version of R. In later tests we observed that PINS did not crash on 64-bit R, and it only crashed on 32-bit R due to insufficient memory. The clustering that PINS obtained on the breast cancer dataset had 4 enriched clinical parameters, and the p-value for the logrank test on that clustering was 0.05.).

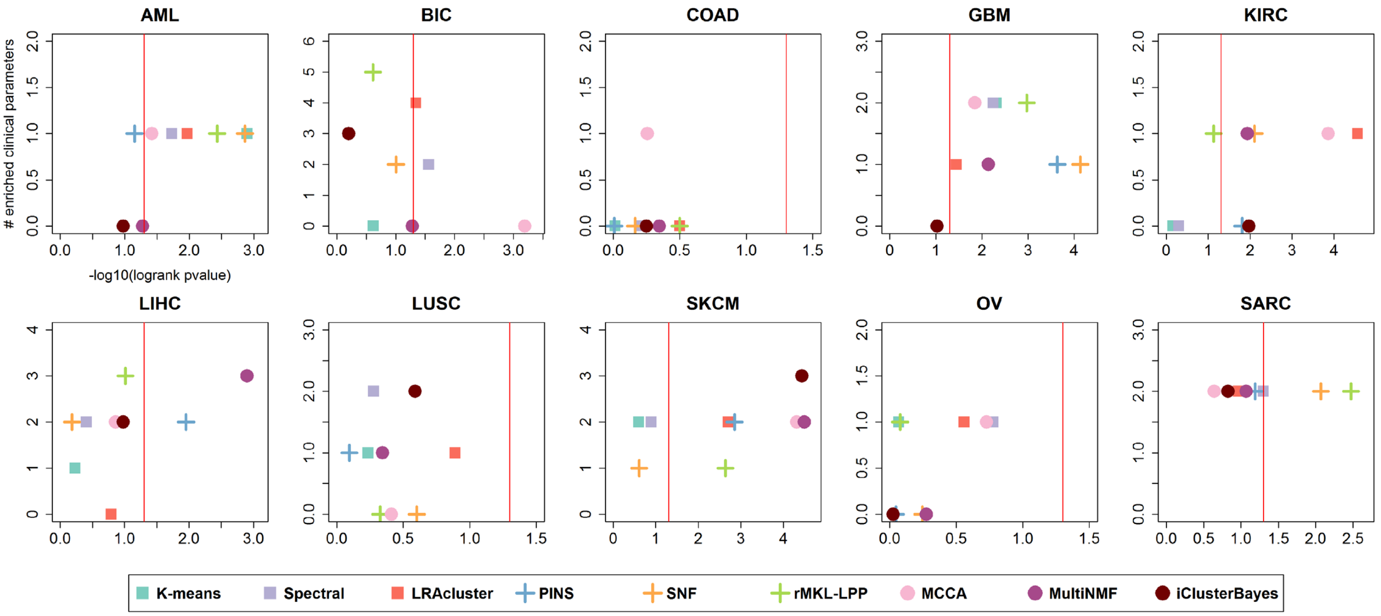


Figure 2. Performance of the algorithms on ten multi-omics cancer datasets. For each plot, the x-axis measures the differential survival between clusters (–$\log_{10}$ of logrank's test *P*-value), and the y-axis is the number of clinical parameters enriched in the clusters. Red vertical lines indicate the threshold for significantly different survival (*P*-value ≤ 0.05)

Table 2. Cancer types with significant results per algorithm

	k-means	Spectral	LRAcluster	PINS	SNF	rMKL-LPP	MCCA	MultiNMF	iClusterBayes
Significantly different survival	2	3	5	4	4	4	5	4	2
Significant clinical enrichment	7	8	8	6	7	8	8	6	5

For each benchmarked algorithm, the number of cancer subtypes for which its clustering had significantly different prognosis (first row) and had at least one enriched clinical label (second row) are shown.

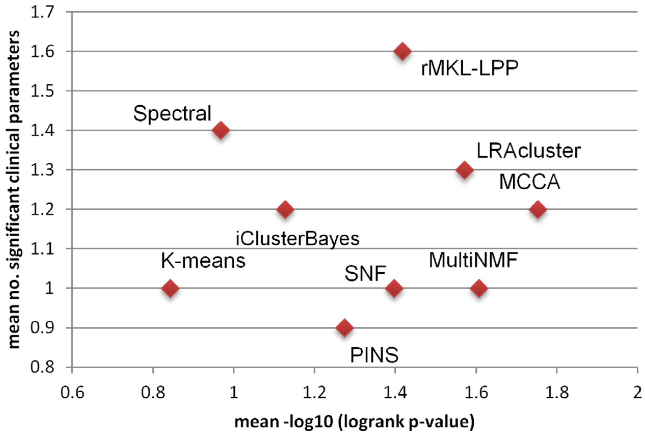


Figure 3. Mean performance of the algorithms on ten multi-omics cancer datasets. The x-axis measures the differential survival between clusters (mean –$\log_{10}$ of logrank's test *P*-value), and the y-axis is the mean number of clinical parameters enriched in the clusters.

higher number of clusters is controlled in the logrank and clinical enrichment tests by having more degrees of freedom for its χ^2 statistic.

The runtime of the different methods is reported in Table 4. Note that as mentioned earlier, iClusterBayes was run on a cluster, while the other methods were run on a desktop computer. All methods except for LRAcluster and iCluster took less than ten minutes per dataset on average. LRAcluster and iClusterBayes took about 56 and 72 minutes per dataset, respectively.

Figure 4 also shows the performance of the benchmarked methods for single-omic data. While several methods had worse performance on single-omic datasets, some achieved better performance. For example, the highest number of enriched clinical parameters for both single and multi-omic datasets (18) was achieved by rMKL-LPP on gene expression. The gene expression solution also had better prognostic value than the multi-omic solution.

To further test how analysis of single-omic datasets compares to multi-omic datasets, we chose for each dataset and method the single omic that gave the best results for survival and clinical enrichment. In this analysis, rMKL-LPP had both the highest total number of enriched clinical parameters (21), and the highest total survival significance (21.86). The runtime, number of clusters, and survival and clinical enrichment analysis for single-omic datasets appear in Supplementary Files 1 and 3. These results suggest that analysis of multi-omics data does not consistently provide better prognostic value and clinical significance compared to analysis of single-omic data alone, especially when different single-omics are used for each cancer types.

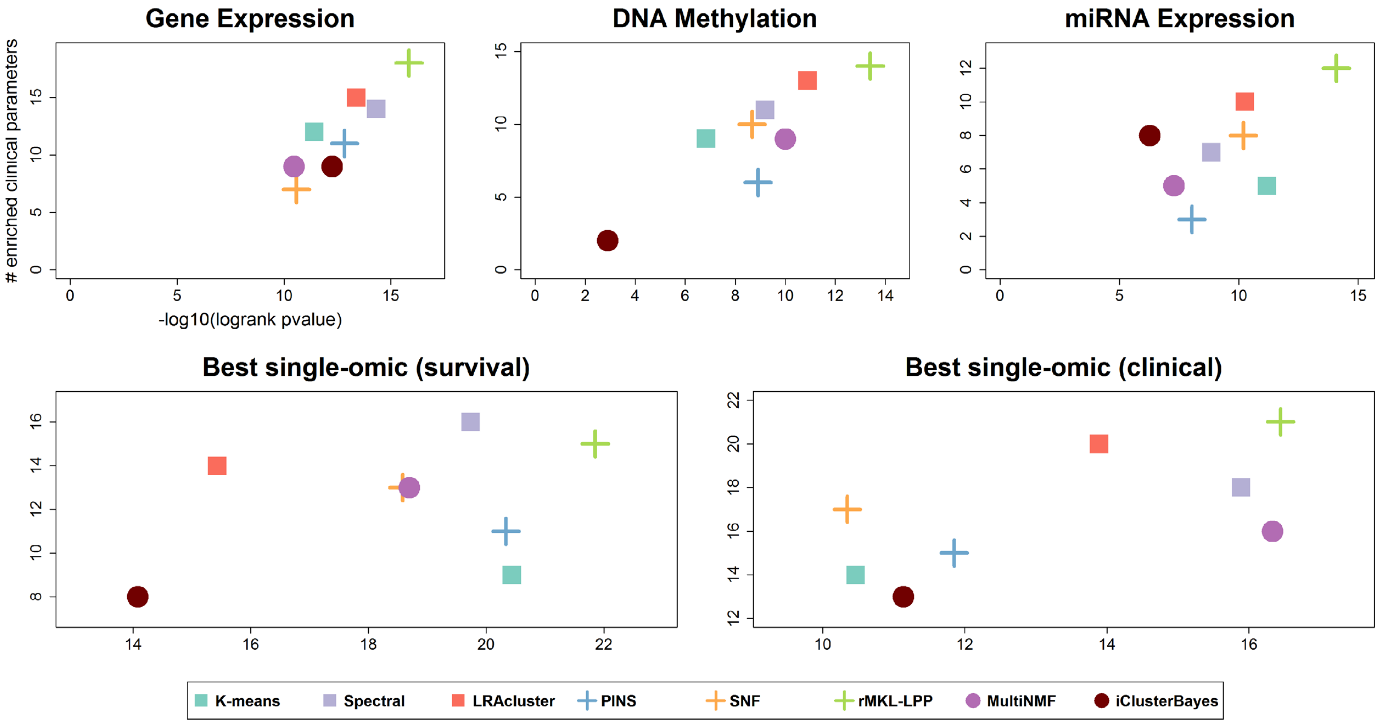


Figure 4. Summarized performance of the algorithms across ten cancer datasets. For each plot, the x-axis measures the total differential prognosis between clusters (sum across all datasets of $-\log_{10}$ of logrank's test *P*-value), and the y-axis is the total number of clinical parameters enriched in the clusters across all cancer types. (**A**–**C**) Results for single-omic datasets. (**D**) Results when each method uses the single omic that achieves the highest significance in survival. (**E**) Same with respect to enrichment of clinical labels.

Table 3. Number of clusters chosen by the benchmarked algorithms on ten multi-omics cancer datasets

	AML	BIC	COAD	GBM	KIRC	LIHC	LUSC	SKCM	OV	SARC	Means
K-means	5	2	2	5	2	2	2	2	2	2	2.6
Spectral	9	3	2	5	2	2	2	2	4	2	3.3
LRAcluster	7	7	5	11	3	12	12	15	9	13	9.4
PINS	4	NA	4	2	2	5	4	15	2	3	4.6
SNF	4	2	3	2	4	2	2	3	3	3	2.8
rMKL-LPP	6	7	6	6	11	6	6	7	6	6	6.7
MCCA	11	14	2	11	15	15	12	2	9	15	10.6
MultiNMF	2	2	2	3	2	3	2	2	2	2	2.2
iClusterBayes	2	3	2	2	2	3	2	2	2	2	2.2

The right column is the average number of clusters across all cancer types.

Table 4. Runtime in seconds of the algorithms on ten multi-omics cancer datasets

	AML	BIC	COAD	GBM	KIRC	LIHC	LUSC	SKCM	OV	SARC	Means
K-means	96	1306	153	212	102	407	444	723	303	191	394
Spectral	3	8	3	3	3	5	5	6	4	4	4
LRAcluster	957	11655	1405	1370	991	3959	3353	5892	2299	2004	3388
PINS	41	NA	112	115	59	125	228	317	214	113	147
SNF	5	42	7	7	6	14	13	21	9	8	13
rMKL-LPP	222	192	205	221	191	255	213	333	263	238	233
MCCA	12	43	12	13	13	26	25	25	19	16	20
MultiNMF	19	51	25	19	17	35	27	45	21	23	28
iClusterBayes*	2628	7832	3213	2569	2756	5195	4682	6077	4057	3969	4298

The right column is the average runtime across all cancer types. *For iClusterBayes numbers are elapsed time on a multi-core platform.

DISCUSSION

We have reviewed methods for multi-omics and multi-view clustering. In our tests on 10 cancer datasets, overall, rMKL-LPP performed best in terms of clinical enrichment, and outperformed all methods except MCCA and MultiNMF with respect to survival. The high performance of MCCA and MultiNMF is remarkable, as these are multi-view methods that were not specifically developed for omics data (though MCCA was applied to it).

Throughout this review we provided guidelines about the advantages and disadvantages of different approaches and algorithms. In the benchmark, no single method consistently outperformed all others on any of the assessment criteria. While some methods were shown to do well, we cannot conclude from this that they should be always preferred. We also could not identify one 'best' integration approach, but it is interesting to note that the top two performers with respect to survival were dimension reduction methods.

Careful consideration should be given when applying multi-view clustering methods to multi-omic data, since these data have characteristics that multi-view methods do not necessarily consider. The most prominent of these characteristics is the large number of features relative to the number of samples. For example, CCA inverts the covariance matrix of each omic. This matrix is not invertible when there are more features than samples, and sparsity regularization is necessary. Another feature of multi-omic data is the dependencies between features in different omics, but several multi-view algorithms assume conditional independence of the omics given the clustering structure. This dependency is rarely considered, since it greatly increases the complexity of models. An additional characteristic of current omic data types is that due to cellular regulation, they have an intrinsic lower dimensional representation. The characteristic is utilized by many methods.

In our benchmark, single-omic data alone sometimes gave better results than multi-omics data. This was intensified when for each algorithm the 'best' single-omic for each cancer type was chosen. These results question the current assumptions underlying multi-omics analysis in general and multi-omics clustering in particular.

Several approaches may lead to improved results for multi-omics analysis. First, methods that suggest different clusterings in different omics were developed and reviewed here, but were not included in the benchmark, since it is not clear how to compare algorithms that do not output a global clustering solution to those that do. These methods may be more sensitive to strong signals appearing in only some of the omics. Second, future algorithms can perform omic selection in the same manner that algorithms today perform feature selection. In the benchmark, we let each method choose a single-omic for each cancer type given the results of the analysis, which are usually not available for real data. Methods that filter omics with contradicting signals might obtain a clearer clustering. Finally, while some methods for multi-omics clustering incorporate prior biological knowledge, few of them incorporate knowledge regarding the relationship between omics, or between features in different omics. Several statistical methods include some form of biological modeling by describing the distribution of the omics, and MDI tunes the similarity of clustering solutions in different omics based on the omics similarity. However, these methods do not model the biological relationships between omics. A notable exception is PARADIGM, which formulates the relationships between different omics. However, it also requires accurate prior knowledge about biochemical consequences of interactions, which is often unavailable. Methods that model relations between omics might benefit from additional biological knowledge, even without modeling whole pathways. For example, one can incorporate in a model the fact that promoter methylation is anti-correlated with gene expression. As far as we know, such methods were only developed for copy-number variation and gene expression data (e.g. (109)), and not in the context of clustering.

We detected large differences between the p-values derived from the χ^2 approximation compared to the *P*-values derived from the permutation tests in the statistical tests we used. The differences were especially large due to the small sample size, small cluster sizes (in solutions with a high number of clusters) and due to a low number of events (high survival) for the logrank test. These p-values are used by single and multi-omic methods to assess their performance, and the logrank p-value is often the main argument for an algorithm's merit. The large differences between the *P*-values question the validity of analyses that are based on the χ^2 approximation, at least for TCGA data. Future work must use exact or permutation-based calculations of the *P*-value in datasets with similar characteristics to those used here for the benchmark.

The benchmark we performed is not without limitations. Gauging performance using patient survival is somewhat biased to known cancer subtypes, which may have been used in treatment decisions. Additionally, cancer subtypes that are biologically different may have similar survival. This is also true for enrichment of clinical parameters, although we attempted to choose parameters that would not lead to this bias. However, these measures are widely used for clustering assessment, including in the papers describing some of the benchmarked methods. Another limitation of the benchmark is that it only examines clustering, while some of the methods have additional goals and output. For example, in dimension reduction algorithms, the low dimensional data can be used to analyze features, and not only patients, e.g. by calculating axes of variation common to several omics. With respect to feature analysis, multi-omic algorithms can have an advantage over single-omic algorithms that we did not test. Finally, though we selected the parameters of each benchmarked method according to the guidelines given by the authors, judicious fine-tuning of the parameters may improve results.

DATA AVAILABILITY

All the processed raw data are available at http://acgt.cs.tau.ac.il/multi_omic_benchmark/download.html.

SUPPLEMENTARY DATA

Supplementary Data are available at NAR Online.

ACKNOWLEDGEMENTS

The results published here are based upon data generated by The Cancer Genome Atlas managed by the NCI and NHGRI. Information about TCGA can be found at http://cancergenome.nih.gov. We thank Nora K. Speicher for providing the rMKL-LPP tool and Ron Zeira for helpful comments.

FUNDING

United States–Israel Binational Science Foundation (BSF), Jerusalem, Israel and the United States National Science Foundation (NSF); Bella Walter Memorial Fund of the Israel Cancer Association (in part); Edmond J. Safra Center for Bioinformatics at Tel-Aviv University (to N.R.) (in part). Funding for open access charge: BSF–NSF and ICA grant listed under Funders (in part).
Conflict of interest statement. None declared.

REFERENCES

1. Goodwin,S., McPherson,J.D. and McCombie,W.R. (2016) Coming of age: ten years of next-generation sequencing technologies. *Nat. Rev. Genet.*, **17**, 333–351.
2. Ozsolak,F. and Milos,P.M. (2011) RNA sequencing: advances, challenges and opportunities. *Nat. Rev. Genet.*, **12**, 87–98.
3. Allison,D.B., Cui,X., Page,G.P. and Sabripour,M. (2006) Microarray data analysis: From disarray to consolidation and consensus. *Nat. Rev. Genet.*, **7**, 55–65.
4. Yong,W.-S., Hsu,F.-M. and Chen,P.-Y. (2016) Profiling genome-wide DNA methylation. *Epigenet. Chromatin*, **9**, 26.
5. Jain,A.K., Murty,M.N. and Flynn,P.J. (1999) Data clustering: a review. *ACM Comput. Surv.*, **31**, 264–323.
6. Prasad,V., Fojo,T. and Brada,M. (2016) Precision oncology: origins, optimism, and potential. *Lancet Oncol.*, **17**, e81–e86.
7. Zhao,J., Xie,X., Xu,X. and Sun,S. (2017) Multi-view learning overview: Recent progress and new challenges. *Information Fusion*, **38**, 43–54.
8. Network,T.C.G.A. (2008) Comprehensive genomic characterization defines human glioblastoma genes and core pathways. *Nature*, **455**, 1061–1068.
9. Huang,S., Chaudhary,K. and Garmire,L.X. (2017) More is better: recent progress in multi-omics data integration methods. *Front. Genet.*, **8**, 84.
10. Bersanelli,M., Mosca,E., Remondini,D., Giampieri,E., Sala,C., Castellani,G. and Milanesi,L. (2016) Methods for the integration of multi-omics data: mathematical aspects. *BMC Bioinformatics*, **17**, S15.
11. Li,Y., Wu,F.-X. and Ngom,A. (2016) A review on machine learning principles for multi-view biological data integration. *Brief. Bioinformatics*, 325–340.
12. Wang,D. and Gu,J. (2016) Integrative clustering methods of multi-omics data for molecule-based cancer classifications. *Quant. Biol.*, **4**, 58–67.
13. Meng,C., Zeleznik,O.A., Thallinger,G.G., Kuster,B., Gholami,A.M. and Culhane,A.C. (2016) Dimension reduction techniques for the integrative analysis of multi-omics data. *Brief. Bioinformatics*, **17**, 628–641.
14. Tini,G., Marchetti,L., Priami,C. and Scott-Boyer,M.-P. (2017) Multi-omics integration-a comparison of unsupervised clustering methodologies. *Brief. Bioinformatics*, doi:10.1093/bib/bbx167.
15. Shen,R., Olshen,A.B. and Ladanyi,M. (2009) Integrative clustering of multiple genomic data types using a joint latent variable model with application to breast and lung cancer subtype analysis. *Bioinformatics*, **25**, 2906–2912.
16. Wu,D., Wang,D., Zhang,M.Q. and Gu,J. (2015) Fast dimension reduction and integrative clustering of multi-omics data using low-rank approximation: Application to cancer molecular classification. *BMC Genomics*, **16**, 1022.
17. Wang,H., Nie,F. and Huang,H. (2013) Multi-view clustering and feature learning via structured sparsity. *Proc. ICML '13*, **28**, 352–360.
18. Bickel,S. and Scheffer,T. (2004) Multi-view clustering. *Proc. ICDM 2004*, 19–26.
19. Hoadley,K.A., Yau,C., Wolf,D.M., Cherniack,A.D., Tamborero,D., Ng,S., Leiserson,M.D., Niu,B., McLellan,M.D., Uzunangelov,V. *et al.* (2014) Multiplatform analysis of 12 cancer types reveals molecular classification within and across tissues of origin. *Cell*, **158**, 929–944.
20. Bruno,E. and Marchand-Maillet,S. (2009) Multiview clustering: A late fusion approach using latent models categories and subject descriptors. In: *Proc. ACM SIGIR '09*. ACM Press, NY, pp. 736–737.
21. Nguyen,T., Tagett,R., Diaz,D. and Draghici,S. (2017) A novel approach for data integration and disease subtyping. *Genome Res.*, **27**, 2025–2039.
22. de Sa,V.R. (2005) Spectral Clustering with Two Views. In: *Proceedings of the Workshop on Learning with Multiple Views, 22nd ICML*. pp. 20–27.
23. Kumar,A., Rai,P. and Daumé,H. III (2011) Co-regularized multi-view spectral clustering. In: *Proc. NIPS '11*. USA, pp. 1413–1421.
24. Chikhi,N.F. (2016) Multi-view clustering via spectral partitioning and local refinement. *Inform. Process. Manage.*, **52**, 618–627.
25. Li,Y., Nie,F., Huang,H. and Huang,J. (2015) Large-scale multi-view spectral clustering with bipartite graph. In: *Proc. AAAI 15*. pp. 2750–2756.
26. Zhou,D. and Burges,C.J.C. (2007) Spectral clustering and transductive learning with multiple views. In: *Proc. ICML '07*. pp. 1159–1166.
27. Xia,R., Pan,Y., Du,L. and Yin,J. (2014) Robust multi-view spectral clustering via low-rank and sparse decomposition. *AAAI Conf. Artif. Intell.*, 2149–2155.
28. Bo,Wang, Jiayan,Jiang, Wei,Wang, Zhi-Hua,Zhou and Zhuowen,Tu (2012) Unsupervised metric fusion by cross diffusion. In: *2012 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, pp. 2997–3004.
29. Wang,B., Mezlini,A.M., Demir,F., Fiume,M., Tu,Z., Brudno,M., Haibe-Kains,B. and Goldenberg,A. (2014) Similarity network fusion for aggregating data types on a genomic scale. *Nat. Methods*, **11**, 333–337.
30. Speicher,N.K. and Pfeifer,N. (2015) Integrating different data types by regularized unsupervised multiple kernel learning with application to cancer subtype discovery. *Bioinformatics*, **31**, i268–i275.
31. Long,B., Yu,P.S. and Zhang,Z.M. (2008) A General Model for Multiple View Unsupervised Learning. In: *Proceedings of the 2008 SIAM International Conference on Data Mining*. Society for Industrial and Applied Mathematics, Philadelphia, PA, pp. 822–833.
32. Lock,E.F., Hoadley,K.A., Marron,J.S. and Nobel,A.B. (2013) Joint and individual variation explained (JIVE) for integrated analysis of multiple data types. *Ann. Appl. Stat.*, **7**, 523–542.
33. O'Connell,M.J. and Lock,E.F. (2016) R. JIVE for exploration of multi-source molecular data. *Bioinformatics*, **32**, 2877–2879.
34. Hotelling,H. (1936) Relations between two sets of variates. *Biometrika*, **28**, 321.
35. Klami,A., Virtanen,S. and Kaski,S. (2013) Bayesian canonical correlation analysis. *J. Mach. Learn.*, **13**, 723–773.
36. Lai,P.L. and Fyfe,C. (2000) Kernel and Nonlinear Canonical Correlation Analysis. *Int. J. Neural Syst.*, **10**, 365–377.
37. Andrew,G., Arora,R., Bilmes,J. and Livescu,K. (2013) Deep canonical correlation analysis. In: *Proc. ICML '13*. Vol. **28**, pp. 1247–1255.
38. Parkhomenko,E., Tritchler,D. and Beyene,J. (2009) Sparse canonical correlation analysis with application to genomic data integration. *Stat. Applic. Genet. Mol. Biol.*, **8**, 1–34.
39. Witten,D.M. and Tibshirani,R.J. (2009) Extensions of sparse canonical correlation analysis with applications to genomic data. *Stat. Applic. Genet. Mol. Biol.*, **8**, Article28.
40. Vía,J., Santamaría,I. and Pérez,J. (2007) A learning algorithm for adaptive canonical correlation analysis of several data sets. *Neural Netw.*, **20**, 139–152.

41. Luo,Y., Tao,D., Ramamohanarao,K., Xu,C. and Wen,Y. (2016) Tensor canonical correlation analysis for multi-view dimension reduction. In: *Proc. ICDE 2016*. pp. 1460–1461.
42. Chen,J., Bushman,F.D., Lewis,J.D., Wu,G.D. and Li,H. (2013) Structure-constrained sparse canonical correlation analysis with an application to microbiome data analysis. *Biostatistics*, **14**, 244–58.
43. Lin,D., Zhang,J., Li,J., Calhoun,V.D., Deng,H.W. and Wang,Y.P. (2013) Group sparse canonical correlation analysis for genomic data integration. *BMC Bioinformatics*, **14**, 245.
44. Rohart,F., Gautier,B., Singh,A. and Lê Cao,K.-A. (2017) mixOmics: An R package for ‘omics feature selection and multiple data integration. *PLoS Computat. Biol.*, **13**, e1005752.
45. Wold,S., Sjöström,M. and Eriksson,L. (2001) PLS-regression: A basic tool of chemometrics. *Chemom. Intell. Lab. Syst.*, **58**, 109–130.
46. Lê Cao,K.-A., Rossouw,D., Robert-Granié,C. and Besse,P. (2008) A sparse PLS for variable selection when integrating omics data. *Stat.Applic. Genet.Mol. Biol.*, **7**, doi:10.2202/1544-6115.1390.
47. Lê Cao,K.-A., Martin,P.G., Robert-Granié,C. and Besse,P. (2009) Sparse canonical methods for biological data integration: application to a cross-platform study. *BMC Bioinformatics*, **10**, 34.
48. Trygg,J. (2002) O2-PLS for qualitative and quantitative analysis in multivariate calibration. *J. Chemometrics*, **16**, 283–293.
49. Rosipal,R., Trejo,L.J., Cristianini,N., Shawe-Taylor,J. and Williamson,B. (2001) Kernel partial least squares regression in reproducing kernel Hilbert space. *J. Mach. Learn. Res.*, **2**, 97–123.
50. Rantalainen,M., Bylesjö,M., Cloarec,O., Nicholson,J.K., Holmes,E. and Trygg,J. (2007) Kernel-based orthogonal projections to latent structures (K-OPLS). *J. Chemometrics*, **21**, 376–385.
51. Li,W., Zhang,S., Liu,C.-C. and Zhou,X.J. (2012) Identifying multi-layer gene regulatory modules from multi-dimensional genomic data. *Bioinformatics*, **28**, 2458–2466.
52. Löfstedt,T. and Trygg,J. (2011) OnPLS-a novel multiblock method for the modelling of predictive and orthogonal variation. *J. Chemometrics*, **25**, 441–455.
53. Meng,C., Kuster,B., Culhane,A.C. and Gholami,A. (2014) A multivariate approach to the integration of multi-omics datasets. *BMC Bioinformatics*, **15**, 162.
54. Liu,J., Wang,C., Gao,J. and Han,J. (2013) Multi-View Clustering via Joint Nonnegative Matrix Factorization. In: *Proc. ICDM ’13*. Society for Industrial and Applied Mathematics, Philadelphia, PA, pp. 252–260.
55. Kalayeh,M.M., Idrees,H. and Shah,M. (2014) NMF-KNN: Image annotation using weighted multi-view non-negative matrix factorization. In: *2014 IEEE Conference on Computer Vision and Pattern Recognition*. pp. 184–191.
56. Huang,J., Nie,F., Huang,H. and Ding,C. (2014) Robust Manifold Nonnegative Matrix Factorization. *ACM Trans. Knowledge Discov. Data*, **8**, 1–21.
57. Zhang,S., Liu,C.-C., Li,W., Shen,H., Laird,P.W. and Zhou,X.J. (2012) Discovery of multi-dimensional modules by integrative analysis of cancer genomic data. *Nucleic Acids Res.*, **40**, 9379–9391.
58. Zhang,X., Zong,L., Liu,X. and Yu,H. (2015) Constrained NMF-based multi-view clustering on unmapped data. In: *Proc. AAAI ’15*. Vol. **4**, pp. 3174–3180.
59. Li,S.-Y., Jiang,Y. and Zhou,Z.-H. (2014) Partial multi-view clustering. In: *Proc. AAAI ’14*. AAAI Press, pp. 1968–1974.
60. Žitnik,M. and Zupan,B. (2015) Data fusion by matrix factorization. *IEEE Trans. Pattern Anal. Mach. Intell.*, **37**, 41–53.
61. White,M., Yu,Y., Zhang,X. and Schuurmans,D. (2012) Convex multi-view subspace learning. In: *Proc. NIPS ’12*. USA, pp. 1673–1681.
62. Guo,Y. (2013) Convex subspace representation learning from multi-view data. *AAAI 2013*, 387–393.
63. Zhang,C., Fu,H., Liu,S., Liu,G. and Cao,X. (2015) Low-rank tensor constrained multiview subspace clustering. In: *Proc. ICCV ’15*. IEEE, pp. 1582–1590.
64. Mo,Q., Wang,S., Seshan,V.E., Olshen,A.B., Schultz,N., Sander,C., Powers,R.S., Ladanyi,M. and Shen,R. (2013) Pattern discovery and cancer gene identification in integrated cancer genomic data. *Proc. Natl. Acad. Sci. U.S.A.*, **110**, 4245–4250.
65. Mo,Q., Shen,R., Guo,C., Vannucci,M., Chan,K.S. and Hilsenbeck,S.G. (2018) A fully Bayesian latent variable model for integrative clustering analysis of multi-type omics data. *Biostatistics*, **19**, 71–86.
66. Vaske,C.J., Benz,S.C., Sanborn,J.Z., Earl,D., Szeto,C., Zhu,J., Haussler,D. and Stuart,J.M. (2010) Inference of patient-specific pathway activities from multi-dimensional cancer genomics data using PARADIGM. *Bioinformatics*, **26**, i237–i245.
67. Savage,R.S., Ghahramani,Z., Griffin,J.E., de la Cruz,B.J. and Wild,D.L. (2010) Discovering transcriptional modules by Bayesian data integration. *Bioinformatics*, **26**, i158–i167.
68. Yuan,Y., Savage,R.S. and Markowetz,F. (2011) Patient-specific data fusion defines prognostic cancer subtypes. *PLoS Comput. Biol.*, **7**, e1002227.
69. Kirk,P., Griffin,J.E., Savage,R.S., Ghahramani,Z. and Wild,D.L. (2012) Bayesian correlated clustering to integrate multiple datasets. *Bioinformatics*, **28**, 3290–3297.
70. Lock,E.F. and Dunson,D.B. (2013) Bayesian consensus clustering. *Bioinformatics*, **29**, 2610–2616.
71. Gabasova,E., Reid,J. and Wernisch,L. (2017) Clusternomics: Integrative context-dependent clustering for heterogeneous datasets. *PLOS Comput. Biol.*, **13**, e1005781.
72. Ahmad,A. and Fröhlich,H. (2017) Towards clinically more relevant dissection of patient heterogeneity via survival-based Bayesian clustering. *Bioinformatics*, **33**, 3558–3566.
73. Coretto,P., Serra,A. and Tagliaferri,R. (2018) Robust clustering of noisy high-dimensional gene expression data for patients subtyping. *Bioinformatics*, doi:10.1093/bioinformatics/bty502.
74. Chaudhary,K., Poirion,O.B., Lu,L. and Garmire,L.X. (2018) Deep learning-based multi-omics integration robustly predicts survival in liver cancer. *Clin. Cancer Res.*, **24**, 1248–1259.
75. Liang,M., Li,Z., Chen,T. and Zeng,J. (2015) Integrative data analysis of multi-platform cancer data with a multimodal deep learning approach. *IEEE/ACM Trans. Comput. Biol. Bioinformatics*, **12**, 928–937.
76. Bickel,P.J., Li,B., Tsybakov,A.B., van de Geer,S.A., Yu,B., Valdés,T., Rivero,C., Fan,J. and van der Vaart,A. (2006) Regularization in statistics. *Test*, **15**, 271–344.
77. Tibshirani,R. (1996) Regression Selection and Shrinkage via the Lasso. *J. R. Stat. Soc. B*, **58**, 267–288.
78. Blum,A. and Mitchell,T. (1998) Combining labeled and unlabeled data with co-training. In *Proc. COLT ’98*. ACM Press, NY, 92–100.
79. Monti,S., Tamayo,P., Mesirov,J. and Golub,T. (2003) Consensus clustering: a resampling-based method for class discovery and visualization of gene expression microarray data. *Mach. Learn.*, **52**, 91–118.
80. Hofmann,T. (1999) Probabilistic latent semantic analysis. In: *Proc. UAI ’99*. Morgan Kaufmann Publishers Inc., San Francisco, pp. 289–296.
81. Vega-Pons,S. and Ruiz-Shulcloper,J. (2011) A Survey of clustering ensemble algorithms. *Int. J. Pattern Recognit. Artif. Intell.*, **25**, 337–372.
82. von Luxburg,U. (2007) A tutorial on spectral clustering. *Stat. Comput.*, **17**, 395–416.
83. Mohar,B. (1991) The Laplacian spectrum of graphs. *Graph Theory Combinatorics Applic.*, **2**, 871–898.
84. Lo Asz,L. (1993) Random walks on graphs: a survey. *Combinatorics*, 1–46.
85. Pearl,J. (1988) *Probabilistic Reasoning in Intelligent Systems : Networks of Plausible Inference*. Morgan Kaufmann Publishers.
86. Cox,D.R. and Oakes,D. (1984) *Analysis of Survival Data*. Chapman and Hall
87. Chaudhuri,K., Kakade,S.M., Livescu,K. and Sridharan,K. (2009) Multi-view clustering via canonical correlation analysis. In: *Proc. ICML ’09*. pp. 1–8.
88. Bach,F.R. and Jordan,M.I. (2006) A probabilistic interpretation of canonical correlation analysis. *Dept. Statist. Univ. California Berkeley CA Tech. Rep.*, **688**, 1–11.
89. Bylesjö,M., Eriksson,D., Kusano,M., Moritz,T. and Trygg,J. (2007) Data integration in plant biology: The O2PLS method for combined modeling of transcript and metabolite data. *Plant J.*, **52**, 1181–1191.
90. el Bouhaddani,S., Houwing-Duistermaat,J., Salo,P., Perola,M., Jongbloed,G. and Uh,H.-W. (2016) Evaluation of O2PLS in omics data integration. *BMC Bioinformatics*, **17**, S11.
91. Hwang,D., Stephanopoulos,G. and Chan,C. (2004) Inverse modeling using multi-block PLS to determine the environmental conditions that provide optimal cellular function. *Bioinformatics*, **20**, 487–499.

92. Dray,S., Chessel,D. and Thioulouse,J. (2003) Co-inertia analysis and the linking of ecological data tables. *Ecology*, **84**, 3078–3089.
93. Seung,H.S. and Lee,D.D. (1999) Learning the parts of objects by non-negative matrix factorization. *Nature*, **401**, 788–791.
94. Lee,D.D. and Seung,H.S. (2001) Algorithms for non-negative matrix factorization. *Adv. Neural Inf. Proc. Syst.*, 535–541.
95. Žitnik,M. and Zupan,B. (2015) Survival regression by data fusion. *Syst. Biomed.*, **2**, 47–53.
96. Hoadley,A. *et al.* (2018) Cell-of-origin patterns dominate the molecular classification of 10,000 tumors from 33 types of cancer. *Cell*, **173**, 291–304.
97. Ng,S., Collisson,E.A., Sokolov,A., Goldstein,T., Gonzalez-Perez,A., Lopez-Bigas,N., Benz,C., Haussler,D. and Stuart,J.M. (2012) PARADIGM-SHIFT predicts the function of mutations in multiple cancers using pathway impact analysis. *Bioinformatics*, **28**, i640–i646.
98. Geman,S. and Geman,D. (1984) Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Trans. Pattern Anal. Mach. Intell., PAMI-6*, 721–741.
99. LeCun,Y., Bengio,Y. and Hinton,G. (2015) Deep learning. *Nature*, **521**, 436–444.
100. Krizhevsky,A., Sutskever,I. and Geoffrey,E. H. (2012) ImageNet classification with deep Convolutional neural Networks. In: *Proc. NIPS '12*. Vol. **1**, pp. 1097–1105.
101. Sutskever,I., Vinyals,O. and Le,Q.V. (2014) Sequence to sequence learning with neural networks. In: *Proc. NIPS'14*. MIT Press, Cambridge, pp. 3104–3112.
102. Ngiam,J., Khosla,A., Kim,M., Nam,J., Lee,H. and Ng,A.Y. (2011) Multimodal deep learning. *Proc. ICML '11*, 689–696.
103. Wang,W., Arora,R., Livescu,K. and Bilmes,J. (2016) On deep multi-view representation learning: objectives and optimization. *Proc. ICML '16*, 1083–1092.
104. Ching,T., Himmelstein,D.S., Beaulieu-Jones,B.K., Kalinin,A.A., Do,B.T., Way,G.P., Ferrero,E., Agapow,P.-M., Zietz,M., Hoffman,M.M. *et al.* (2018) Opportunities and obstacles for deep learning in biology and medicine. *J. R. Soc. Interface*, **15**, 20170387.
105. Hinton,G.E., Osindero,S. and Teh,Y.-W. (2006) A fast learning algorithm for deep belief nets. *Neural Comput.*, **18**, 1527–1554.
106. Hosmer,D.W., Lemeshow,S. and May,S. (2008) *Applied Survival Analysis: Regression Modeling of Time-to-Event Data*. Wiley-Interscience.
107. Rousseeuw,P.J. and Peter (1987) Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *J. Comput. Appl. Math.*, **20**, 53–65.
108. Vandin,F., Papoutsaki,A., Raphael,B.J. and Upfal,E. (2015) Accurate Computation of Survival Statistics in Genome-Wide Studies. *PLOS Comput. Biol.*, **11**, 1–18.
109. Aure,M.R., Steinfeld,I., Baumbusch,L.O., Liestøl,K., Lipson,D., Nyberg,S., Naume,B., Sahlberg,K.K., Kristensen,V.N., Børresen-Dale,A.-L. *et al.* (2013) Identifying in-trans process associated genes in breast cancer by integrated analysis of copy number and expression data. *PLoS ONE*, **8**, 1–15.

Supplementary Materials for “Multi-omic and multi-view clustering algorithms: Review and cancer benchmark”

Nimrod Rappoport and Ron Shamir*

The Blavatnik School of Computer Science, Tel Aviv University, Tel Aviv 69978, Israel

*To whom correspondence should be addressed. Tel: +972 3 640 5383; Fax: +972 3 640 5384;

Email: rshamir@tau.ac.il (RS)

Contents

Hardware

All experiments for timing the different methods were run on the following machine:

Windows 7 enterprise
Intel® Core™ i7-4770 CPU @ 3.4GHz
4 cores
32 GB RAM
64 bit operating system

The only exception were experiments for iClusterBayes, which were performed on a cluster:

Linux 4.9
72 CPUs, 2300 MHz each
756 GB RAM
64 bit operating system

Early Preprocessing

The TCGA datasets were preprocessed as follows: patients and features with more than 20% missing values were removed, and missing values were imputed using k nearest neighbor imputation. These data were further preprocessed by each benchmarked method separately, as described in this document. For methylation data, we selected the 5000 features with maximal variance in each dataset.

Survival and phenotype data were downloaded from Xena for all TCGA datasets.

Full details for the TCGA datasets used in the paper are in the tcga_datasets_metadata.xlsx file.

Benchmarked Methods and Software

Experiments were run on R-3.5.0.

For all methods, the number of clusters was selected between 2 and 15.

To determine the number of clusters for a method, or the dimension data are reduced to, we often used the “elbow method”. To choose the optimal elbow automatically rather than manually, we used as approximation the second derivative of a vector v:

$$v[i+1] + v[i-1] - 2v[i]$$

We took the index i that brings this expression to maximum or minimum (depending on whether v increases or decreases).

For all methods, we adhered to the guidelines for usage and parameter selection given by the developers. In some cases, where no information was given by the authors, we devised parameter selection methods. For example, MultiNMF does not discuss how to select the number of clusters in a dataset.

Kmeans

Preprocessing:

Sequence features were log transformed. The 2000 features with highest variance from gene-expression and methylation omics were selected. In the miRNA omic, features with zero variance were filtered. All features were then normalized to have zero mean and standard deviation 1.

Execution details:

kmeans R function was called with 60 different starting solution. For the multi-omic experiments, all matrices were concatenated. For single-omic experiments, the omic matrix was used as input.

Parameter selection:

The optimal number of clusters was chosen using the elbow method, where v(i) is the total within cluster sum of squares for a solution with i clusters.

Implementation:

The built-in kmeans implementation of R was used.

Spectral Clustering

Preprocessing:

Sequence features were log transformed. In the miRNA omic, features with zero variance were filtered. All features were then normalized to have zero mean and standard deviation 1.

Execution details:

For multi-omic experiments normalized matrices were concatenated to a single matrix. An affinity matrix for each omic was calculated using the functions dist2 and affinityMatrix from SNFtool package, with 20 neighbors and sigma=0.5. Spectral clustering was run on the affinity matrix with default parameters.

Parameter selection:

The optimal number of clusters was chosen using the rotation method [1] on the affinity matrix.

Implementation:

The functions dist2, affinityMatrix and spectralClustering from SNFtool version 2.3.0 were used.

LRAcluster

Preprocessing:

Sequence features were log transformed. In the miRNA omic, features with zero variance were filtered.

Execution details:

LRAcluster was run on the data as Gaussian data type, as performed for iClusterBayes, with default parameters. Kmeans with default arguments was used on the low dimensional representation. We also attempted not to log-transform sequence data, and treat counts as poisson data type, but this approach yielded worse results.

Parameter selection:

To determine the lower dimension of the data, LRAcluster was run for dimensions 1:15, and the dimension was selected using the elbow method where v(i) is the explained variance for solution using i dimensions (the "potential" attribute on LRAcluster's return value). To determine the number of clusters, we clustered the low dimensional data using k-means (with 60 different starting solutions) for k=2,…,15 and took the clustering with lowest silhouette score.

Implementation:

We used the LRAcluster from LRAcluster package version 1.0 downloaded from: http://bioinfo.au.tsinghua.edu.cn/member/jgu/lracluster/

The built-in kmeans implementation of R was used.

PINS

Preprocessing:

Sequence features were log transformed. In the miRNA omic, features with zero variance were filtered.

Execution details:

For multi-omic data, we used the function SubtypingOmicsData with default arguments. For single-omic data, we used the function PerturbationClustering with default arguments.

Parameter selection:

The number of clusters was automatically determined by the package. The maximum allowed number of clusters was set to 15.

Implementation:

We used the PINSPlus package version 1.0.1.

MultiNMF

Preprocessing:

Sequence features were log transformed. The 2000 features with highest variance from gene-expression and methylation omics were selected. In the miRNA omic, features with zero variance were filtered.

Execution details:

For single-omic data, we applied NMF on the matrix using Lee's optimization. For multi-omics data, all omics were normalized to have a total sum 1. **Parameter selection:**

The options argument, containing several parameters, was set as in the example given within the package.

To determine the dimension of the data and the number of clusters, we ran the method for k from 1 to 15. We used the elbow method where v(i) is the total variance of the data in the i-dimensional representation to determine the dimension. The number of clusters was set to be equal to the dimension.

Implementation:

For single-omic data, we used the R NMF package version 0.21.0. We used the method nmf with method='lee'.

We used the MultiNMF function from the Matlab package implemented by the author of the method. It was downloaded here: http://jialu.info/code/Code_multiNMF.zip.

rMKL-LPP

Preprocessing:

In the miRNA omic, features with zero variance were filtered. Features measured by RNA-seq and miRNA-seq were log transformed, and all features were then normalized to have zero mean and standard deviation 1. We also attempted not to perform the log transformation, and it yielded comparable results.

Execution details:

For each omic we created five similarity matrices, with kernel width = $10^{-6}, 10^{-3},\ 1,\ 10^{3},\ 10^{6}$. These were used as input to rMKL-LPP, with dimension 5 and 9 nearest neighbors.

Parameter selection:

The number of clusters was determined by rMKL-LPP.

Implementation:

An implementation of rMKL-LPP was provided by its authors.

IClusterBayes

Preprocessing:

Features measured by RNA-seq and miRNA-seq were log transformed. In the miRNA omic, features with zero variance were filtered. We also tried to only select the 2000 features with highest variance from gene-expression and methylation omics, but it yielded worse results.

Execution details:

iClusterBayes was run on the data as Gaussian data type, as performed in the analyses conducted by the authors, with default parameters.

Parameter selection:

To determine the lower dimension of the data, we used the dimension with maximal deviance ratio as defined by the authors. The number of clusters is the dimension + 1.

Implementation:

We used the method tune.iClusterBayes from the package iClusterPlus version 1.16.0. We have run iClusterBayes for every low dimension between 1 and 14 in parallel.

SNF

Preprocessing:

In the miRNA omic, features with zero variance were filtered. Sequence features were log transformed, and all features were then normalized to have zero mean and standard deviation 1. We also attempted not to perform the log transformation, but it yielded worse results.

Execution details:

An affinity matrix for each omic was calculated using the functions dist2 and affinityMatrix from SNFtool package, with the number of neighbors equals 1/10 of the number of samples and sigma=0.5. The matrices were integrated using the SNF method with the same number of neighbors and 30 iterations for multi-omic data. Spectral clustering was run on the integrated matrix with default parameters. For single-omic data, the affinity matrix was used as input to spectral clustering.

Parameter selection:

The optimal number of clusters was chosen using the rotation method [1] on the integrated matrix (for multi-omic data) or affinity matrix (for single omic data).

Implementation:

The functions dist2, affinityMatrix, SNF and spectralClustering from SNFtool version 2.3.0 were used.

MCCA

Preprocessing:

Sequence features were log transformed. The 2000 features with highest variance from gene-expression and methylation omics were selected. In the miRNA omic, features with zero variance were filtered. All features were then normalized to have zero mean and standard deviation 1.

Execution details:

For multi-omics data, we used MCCA to find the 15 first canonical variates, and used them to project the samples to a lower dimension. This lower dimension was clustered using kmeans. This method does not support single-omic data.

Parameter selection:

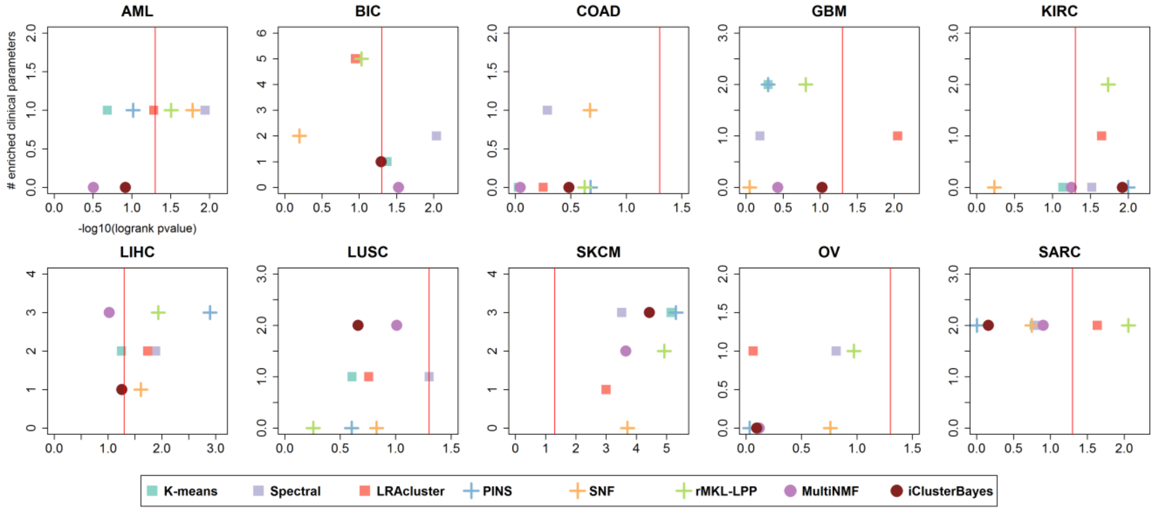
To determine the lower dimension of the data, the dimension was taken using the elbow method where v(i) is the sum of variances for i canonical variates. Given the dimension, the number of clusters was taken by clustering the low dimension representation using kmeans (with 30 different starting solutions) with k=2,…,15 and taking the clustering with lowest silhouette score.

Implementation:

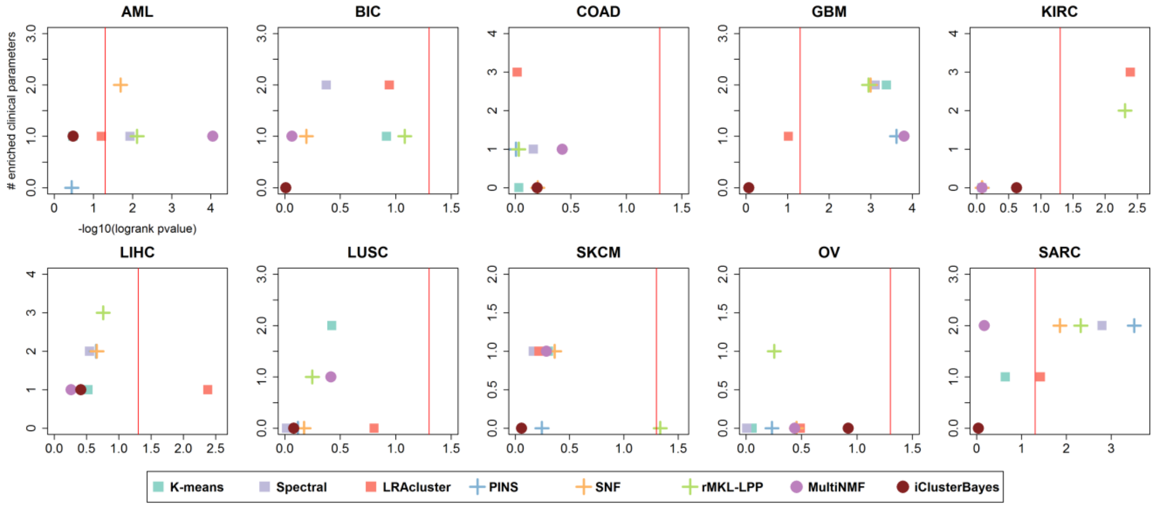
We used the function MultiCCA from the PMA package version 1.0.11. The built-in kmeans implementation of R was used.

Supplementary Figures

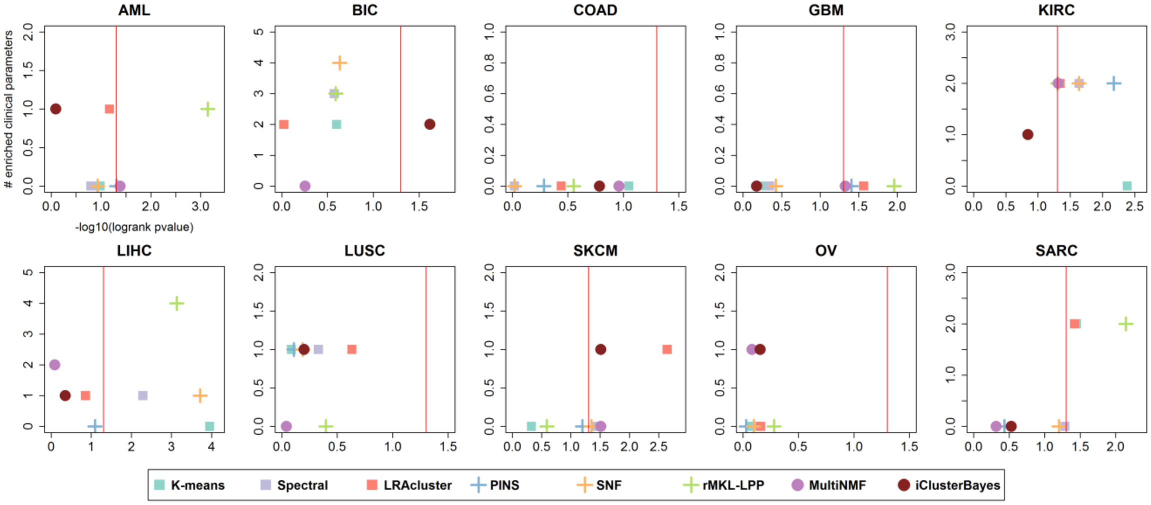
Supp. Figure 1 – survival analysis and clinical enrichment results on gene expression data for ten TCGA cancer datasets.



Supp. Figure 2 – survival analysis and clinical enrichment results on DNA methylation data for ten TCGA cancer datasets.



Supp. Figure 3 – survival analysis and clinical enrichment results on miRNA expression data for ten TCGA cancer datasets.


AML
BIC
COAD
GBM
KIRC
LIHC
LUSC
SKCM
OV
SARC
# enriched clinical parameters
-log10(logrank pvalue)
0.0
0.5
1.0
1.5
2.0
2.5
3.0
0
1
2
3
4
5
0.2
0.4
0.6
0.8
K-means
Spectral
LRAcluster
PINS
SNF
rMKL-LPP
MultiNMF
iClusterBayes

Permutation Tests

Originally, we used the R “survival” library, the chisq.test and kruskal.test R functions to perform the logrank, chi-square and Kruskal-Wallis tests respectively. However, we decided to perform permutation tests rather than use the chi-square approximation for these three tests.

To perform permutation tests, we randomly permuted the clustering assignments of the different samples. This fixed cluster sizes for the test. For the logrank test, the number of permutations we performed for each clustering solution was first $\min\left(\max\left(\frac{10}{original\ p\text{−}value}, 1e4\right), 1e6\right)$ and then another 1e5 permutations until the stopping condition was met. The stopping condition was having both the lower and upper ends of the 95% confidence interval for the p-value to be within 10% of its estimate, and such that the interval did not cross 0.05. We also stopped after a maximum of 2e7 iterations (which was only needed for p-values that are about 1e-5 such as for SKCM on gene expression data).

For the clinical enrichment tests, we continued on performing 1e3 permutations until the 95% confidence interval did not cross 0.05, up to a maximum of 1e5 iterations. This maximum number of iterations was only needed in case the p-value was extremely close to 0.05.

We have built exact confidence intervals for the p-value using the R method binom.test, when considering the number of permutations for which the chi-square statistic was greater or equal to the chi-square statistic of the original clustering, out of the total number of permutations performed.

We detected large differences between the p-values for the logrank test and the chi-square test for independence, when comparing the approximation-based and permutation-based p-values.

References

1. Zelnik-Manor, L. and Perona, P. (2004). Self-tuning spectral clustering. In *Proceedings of the 17th International Conference on Neural Information Processing Systems*, pages 1601–1608. MIT Press.