



Deposited via The University of Sheffield.

White Rose Research Online URL for this paper:

<https://eprints.whiterose.ac.uk/id/eprint/214815/>

Version: Published Version

Article:

Kariotis, S., Tan, P.F., Lu, H. et al. (2024) Omada: robust clustering of transcriptomes through multiple testing. *GigaScience*, 13 (2024). giae039. ISSN: 2047-217X

<https://doi.org/10.1093/gigascience/giae039>

Reuse

This article is distributed under the terms of the Creative Commons Attribution (CC BY) licence. This licence allows you to distribute, remix, tweak, and build upon the work, even commercially, as long as you credit the authors for the original work. More information and the full terms of the licence here:

<https://creativecommons.org/licenses/>

Takedown

If you consider content in White Rose Research Online to be in breach of UK law, please notify us by emailing eprints@whiterose.ac.uk including the URL of the record and the reason for the withdrawal request.

Omada: robust clustering of transcriptomes through multiple testing

Sokratis Kariotis^{1,2,3,*}, Pei Fang Tan^{1,2}, Haiping Lu^{1,4}, Christopher J. Rhodes^{1,3}, Martin R. Wilkins^{1,3}, Allan Lawrie^{1,3}, and Dennis Wang^{1,2,3,4,*}

¹Singapore Institute for Clinical Sciences, Agency for Science, Technology and Research (A*STAR), 30 Medical Dr, 117609, Singapore, Republic of Singapore

²Bioinformatics Institute, Agency for Science, Technology and Research (A*STAR), 30 Biopolis St, Matrix, 138671, Singapore, Republic of Singapore

³National Heart and Lung Institute, Imperial College London, Guy Scadding Building, Dovehouse St, SW3 6LY, London, United Kingdom

⁴Department of Computer Science, University of Sheffield, Regent Court, 211 Portobello, S1 4DP, Sheffield, United Kingdom

*Correspondence address. Dennis Wang, National Heart & Lung Institute, Imperial College London, W12 0BZ, United Kingdom. E-mail:

dennis.wang@imperial.ac.uk; Sokratis Kariotis, Bioinformatics Institute, Agency for Science, Technology and Research, 138671, Republic of Singapore.

E-mail: sokratis_kariotis@bii.a-star.edu.sg

Abstract

Background: Cohort studies increasingly collect biosamples for molecular profiling and are observing molecular heterogeneity. High-throughput RNA sequencing is providing large datasets capable of reflecting disease mechanisms. Clustering approaches have produced a number of tools to help dissect complex heterogeneous datasets, but selecting the appropriate method and parameters to perform exploratory clustering analysis of transcriptomic data requires deep understanding of machine learning and extensive computational experimentation. Tools that assist with such decisions without prior field knowledge are nonexistent. To address this, we have developed Omada, a suite of tools aiming to automate these processes and make robust unsupervised clustering of transcriptomic data more accessible through automated machine learning-based functions.

Findings: The efficiency of each tool was tested with 7 datasets characterized by different expression signal strengths to capture a wide spectrum of RNA expression datasets. Our toolkit's decisions reflected the real number of stable partitions in datasets where the subgroups are discernible. Within datasets with less clear biological distinctions, our tools either formed stable subgroups with different expression profiles and robust clinical associations or revealed signs of problematic data such as biased measurements.

Conclusions: In conclusion, Omada successfully automates the robust unsupervised clustering of transcriptomic data, making advanced analysis accessible and reliable even for those without extensive machine learning expertise. Implementation of Omada is available at <http://bioconductor.org/packages/omada/>.

Keywords: unsupervised learning, cluster analysis, gene expression, software toolkit

Introduction

The rapid development of next-generation sequencing boosted the quantitative analysis of gene expression in a variety of human tissues and organs [1, 2], generating valuable resources [3] for downstream investigative analysis. In recent years, such analyses aim to elucidate disease mechanisms [4] and construct genomic profiles [5] to explain diagnosis [6], prognosis, and treatment patterns. However, transcriptomic profiles can be heterogeneous due to several causes pertaining to technical biases that produce batch effects [7], cellular diversity [8], disease heterogeneity [9], and differences between individuals and populations [10, 11]. In turn, this heterogeneity hinders research efforts aiming to identify subpopulations within complex diseases by applying unsupervised machine learning techniques [12]. More specifically, clustering algorithms have been applied to high-dimensional data from transcriptomic profiling [13–15] and revealed novel molecular classes associated with different symptoms of disease [16–18]. Despite the intrinsic capabilities of these algorithms, a recurrent challenge is the insufficiency of default configurations to tailor these models precisely to unique datasets, thereby impeding the extraction of critical insights into the heterogeneity of the samples. This limitation underscores the complexity and hetero-

geneity inherent in biological data, which often eludes a one-size-fits-all approach to clustering. Empirical evidence from numerous comparisons and tests reveals that the algorithms exhibit variable efficacy across different datasets and disease contexts, accentuating the necessity for model tuning or optimization specific to the dataset in question, particularly with advanced methodologies [19–22]. The challenge is further compounded by the possibility that the samples could be from a spectrum of states and there are no optimal metrics to group them.

Most clustering packages offer implementations of 1 or more algorithms without considering the previous and subsequent steps of the clustering process, such as detecting useful features or estimating the number of subgroups in the data. Since clustering consists of multiple steps, it is important to carefully approach each step so that intermediate decisions are not arbitrary but based on appropriate clustering theory. Due to this diversity in methodologies, there is a need for a toolkit that guides nonexpert users through the different parameters of common clustering algorithms for their unsupervised analysis. One of the most important aspects of sample partitioning is the stability of the generated groups as unstable clusters, usually implying the lack of signal that should be present and driving the clustering of samples.

Received: July 28, 2023. Revised: February 9, 2024. Accepted: June 17, 2024

© The Author(s) 2024. Published by Oxford University Press GigaScience. This is an Open Access article distributed under the terms of the Creative Commons Attribution License (<https://creativecommons.org/licenses/by/4.0/>), which permits unrestricted reuse, distribution, and reproduction in any medium, provided the original work is properly cited.

Table 1: Comparison of clustering methodologies toward identifying heterogeneous patient groups. Five steps of the clustering process are reflected in the columns below. Feasibility analysis assesses the potential of the dataset to be used efficiently by clustering methods based on its dimensions. The clustering algorithm denotes the amount of algorithms available in the methodology. Feature selection allows the selection of the features that best discriminate samples during clustering. Cluster quality describes the methods/metrics used to validate the quality of the clusters produced. Number of clusters describes the approach a methodology follows to estimate the most probable number of different groups in the dataset.

	Feasibility	Clustering algorithm	Feature selection	Cluster quality	Number of clusters
Omada	Stability based	Multiple	Stability based	Internal indexes	Multiple index voting
pvclust [23]	—	Single	—	Bootstrap probability	Bootstrap probability
genieclust [24]	—	Single	—	Internal indexes	—
ClusterR [25]	—	Multiple	—	Single external index	Single index
pdfCluster [26]	—	Multiple	—	Internal indexes	—
clusterSim [27]	—	Single	—	Internal indexes	Single index
clustvarsel [28]	—	—	Gaussian based	—	—
VarSelLCM [29]	—	Multiple	BIC/MICL/AIC	—	BIC/MICL/AIC
FCPS [30]	—	Multiple	—	Internal indexes	Single index

Signals from genes across the transcriptome can vary across different tissues and diseases. Cluster instability can be caused inherently by the data points or by the type and application quality of a clustering technique. There lacks a robust approach that incorporates comprehensive assessment of multiple clustering methods and parameters for different sources of transcriptomic data.

Here we introduce Omada, a toolkit with multiple functions based on cluster stability and quality metrics that supports both experienced and inexperienced users from dataset exploration to the formation of the sample clusters. In contrast to other methodologies, we provided a flexible toolkit offering multiple metrics and algorithms for testing, as well as an end-to-end pipeline. Synthetic and real-world datasets from bulk tumor, single-cell sequencing, and whole blood have been used to test the general usability of Omada. While we cannot guarantee finding the optimal parameters, especially for healthy test samples where true clusters are unknown, we aim to provide users with guidelines based on quality metrics and empirical best practices, in order to ensure that their clustering results are robust and well justified.

Methods

A number of clustering methodologies are shown in Table 1 along with the steps of the clustering process they address.

This toolkit consists of a pipeline that takes in a gene expression matrix to identify transcriptomic subgroups of samples (Fig. 1 and Supplementary Fig. S1). Starting from a matrix of gene expression values (e.g., transcripts per million from RNA sequencing [RNA-seq]), the most suitable clustering method is chosen, followed by selecting the transcript features for clustering and determining the number of clusters and memberships.

Sample and gene expression preprocessing

A preprocessing step is recommended by the user before the application of these tools on any dataset to heighten the chances

of any underlying important signal to be discovered. Data biases and format can often drive clustering attempts to focus on discriminating data points based solely, or mostly, on known information producing no new insights irrespectively of the method used [21, 31]. To address this, it is recommended to attempt to remove/normalize any data points that might be introducing strong biases to allow the novel signal to be detected. Furthermore, numerical data may need to be normalized in order to account for potential misdirecting quantities (i.e., outliers) or specifically transformed to satisfy an algorithm's input criteria. Various normalizations can be used, each catering for specific data types (i.e., arcsine transformation fit RNA-seq data, as described in section "Test datasets"). It is worth noting that Omada (bio.tools/omada, RRID:SCR_025409) applies the arcsine transformation on the input data during the first step and the transformed data are carried over to subsequent steps. Data points or samples have to be filtered based on field knowledge to allow the data to answer specific scientific questions. Expression data should go through proper quality control depending on the manner of collection to identify outliers and remove unreliable data points. For microarrays, it is important to assess sample, hybridization, and overall signal qualities along with signal comparability and potential biases [32]. Array correlations through principal component analysis (PCA) and correlation plots should also be considered [32]. RNA-seq experiments also produce data that need to be controlled for potential trimming of adapter sequences, low-quality reads, uncalled bases, and contaminants by using a plethora of available tools [33, 34]. Quantitative PCR (qPCR)-generated data should be checked for abnormal amplification, positive and negative control samples, and control on PCR replicate variation and determine reference gene expression stability and deviating sample normalization factors [35]. Additionally, high-dimensional data such as the single-cell transcriptomic data should be properly quality-controlled where dying (or nonviable) cells and cells with empty droplets or multiplets should be removed. Together with normalization, scaling, clustering, and cell type-specific quality control, such data preprocessing steps ensure high-quality data to be

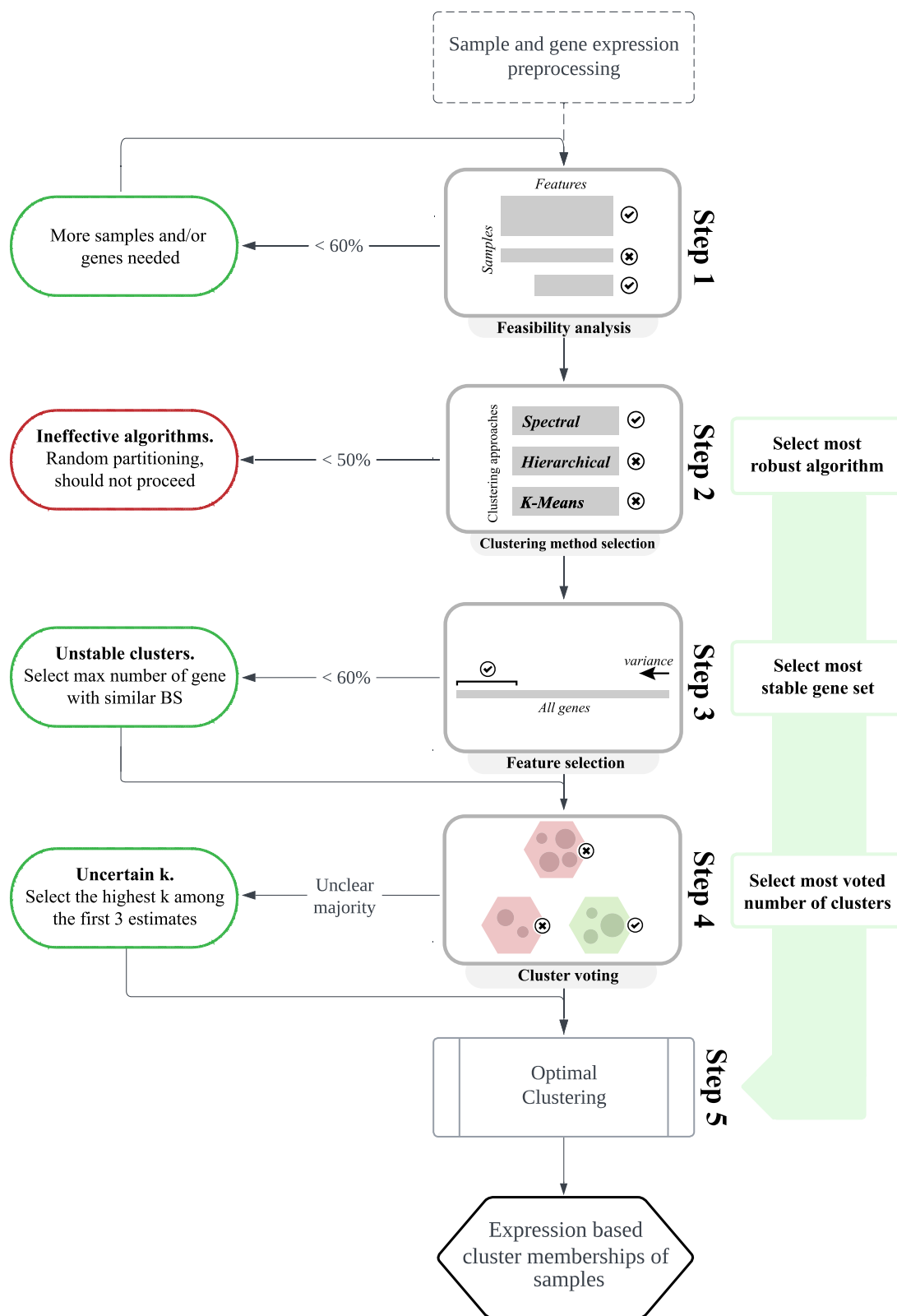


Figure 1: An overview of steps for discovering gene expression subgroups using Omada’s clustering tools. Processing and quality control should precede the application of Omada to ensure more accurate inputs. Step 1 of Omada allows a feasibility analysis to ensure that the input data are suitable for clustering. Step 2 selects the clustering methodology that provides the most consistent partitions for downstream analysis. In step 3, the genes that best discriminate samples and yield the most stable clusters are determined. Step 4 estimates the number of clusters (k) through majority voting by internal machine learning indexes. The final output of step 5 provides a cluster assignment to each sample driven solely by its expression profile. Percentage thresholds are recommended values that can be modified.

considered for further analysis. As for the number of genes, it is advised for larger gene sets (>1,000 genes) to filter down to the most variable ones (few hundreds) before the application of any function as genes that do not vary across samples do not contribute toward identifying heterogeneity. In this step, the filtering should not be very strict but rather should discard genes with zero or near-zero variance (i.e., housekeeping genes). In Omada, the main filtering is achieved in a later stage by identifying the set of genes that outputs the highest stability of generated clusters, as described in section “Stability-based assessment of feature sets.” Moreover, large gene sets require increased computational power and extended runtime without adding any real value due to the large number of nonuseful genes. Lastly, it is important to note that technical artifacts, such as sampling location or machine specifications, may drive clustering, causing the formation of very distinct clusters that can solely be attributed to relevant biases. It is very important for those cases to be identified and extracted insights should be disregarded as they do not reflect real signals or data trends.

Determining clustering potential

At the start of each study, we assess the suitability of the input dataset for clustering, using the Omada package prepipeline analysis, to ensure general dataset attributes do not influence the process (Fig. 1). The number of samples and features (e.g., genes), as well as the balance of the 2 dimensions, directly affects the capabilities of clustering methods to handle the dataset. An inadequate number of samples does not provide enough training power [36], while an overabundance of samples might clutter the provided information and confuse most methodologies [37]. Similarly, too few features can lead to weak clustering criteria and too many features might lead a methodology away from the features that can really differentiate between clusters of samples. Therefore, to estimate the feasibility of a clustering procedure on a specifically sized dataset, we rely on measurable metrics of cluster quality, such as stability. Clusters of high stability denote both a partitionable dataset as well as a dataset-suitable methodology [38]. The feasibility score of any dataset is a function of both dimensions as well as the number of classes requested. As such, if too many or a single class is requested of a relatively small dataset, the calculation will reflect low feasibility due to insufficient samples and/or features to form the desired classes. Omada’s functions `feasibilityAnalysis()` and `feasibility_analysis_based_on_data.R` can simulate datasets with dimensions similar to any dataset and calculate their stability across different numbers of clusters to determine whether such a dataset can be stable enough to be considered for clustering.

Simulating datasets

To assess the quality of the dataset to be used, our toolkit includes 2 functions for simulating datasets of different dimensionalities for stability assessment. We use those to understand the relation between the number of samples, genes, and cluster sizes. The first function simulates datasets, allowing for selecting the number of samples (n), genes (m), and clusters (c). Each cluster contains $\frac{n}{c}$ samples drawn from a normal distribution with a different mean and standard deviation. Each mean is drawn from a sequence of c evenly spaced integers that belong to the range $[5, c * 10]$. Each standard deviation is similarly drawn from a range of $[1, c * 2]$. To estimate the difference between distributions, we calculate the 2-sided Kolmogorov’s D statistic between each pair of distributions

representing the generated classes and plot the empirical cumulative distribution function (EDCF).

Subsequently, we calculate the stability of each k (number of clusters for a particular clustering run) using the `clusterboot` function in R package `fpc` v2.2–3. The number of clusters k to be considered belong to $k \in [\text{number of classes} - 2, \text{number of classes} + 2]$, with a minimum of $k = 2$. The maximum and average stabilities over all k are reported, providing a stability-based quality score that provides an insight in deciding whether a prospect dataset is suitable for a clustering study.

To assess the clustering feasibility of an existing dataset, this toolkit also provides a similar function that generates a simulated dataset based on an input dataset and the user’s estimation of the number of classes. The number of samples and genes equal those of the input dataset and its mean (m_{input}) and standard deviation (sd_{input}) affect those of each generated class within the dataset. Specifically, if $n \in (1, 2, 3, \dots)$ is the number of classes, each class mean (m_{class}) equals $m_{\text{input}} * 10 * n$ and each class standard deviation (sd_{class}) equals $sd_{\text{input}} * 2 * n$.

Intramethod clustering agreement

Unsupervised learning offers a multitude of methods to be applied on specific types of data due to their nature (e.g., numeric, binary) or underlying signal to be detected. Most studies employ widely used methods (e.g., hierarchical clustering) without exercising any kind of selection method that would point toward the most effective methodology. Selecting an appropriate approach requires extensive machine learning and data analysis knowledge coupled with tuning and testing of multiple different algorithms. To enable users without ML expertise to use the vast capabilities of this field and avoid limited efficiency of default methodologies, we present a clustering selection tool that offers an intelligent selection method with unbiased results through parameter randomization. The nature of this selection method allows any number of well-established unsupervised methods to be considered.

To address the lack of class labels and thus a performance measure in unsupervised models, we compare how consistently different approaches partition our data when 1 or more parameters change. As high consistency, we define the high agreement score calculated between different variations of a clustering algorithm. When 2 different clustering runs agree on the partitioning of the samples, they also show robustness since they do not randomly assign samples to subgroups but rather are driven by the underlying structure of the data.

We implemented a tool (Fig. 1) to calculate an average agreement score per clustering approach by comparing a number of runs within each of the 3 clustering approaches (hierarchical [39], k -means [40], spectral clustering [41]) using multiple parameters (kernels, measures, algorithms) specifically based on the dataset provided, which is already arcsine transformed from the first pipeline step. The aforementioned clustering methods were selected to represent fundamentally different approaches of partitioning data so as to cover several types of datasets. Hierarchical clustering is well suited for hierarchical data or data that can be represented as dendrograms, such as taxonomies [42]. K -means allows a different approach of partitioning samples, namely, centroid-based clustering, which is based on non-hierarchical clusters, vector quantization, and representing clusters with their centroids [42]. Spectral clustering adds a spatial component to k -means as it uses a similarity matrix to create a graph where clusters with irregular shapes are investigated [42].

Although the above approaches are not exhaustive, the modular nature of Omada allows for additional approaches to be added in subsequent versions.

The number of comparisons (c), between runs of the same approach, is an additional overarching parameter of this tool and contributes to the agreement score. Each comparison requires 2 variations of the same algorithm. These variations are expressed as clustering parameters such as distance measures or, in the case of hierarchical clustering, linkage type (Table 2). A difference in such parameters produces clustering models that, despite using the same algorithm, have different formulas to calculate clusters. If these different formulas agree on how to partition the data, then the algorithm using them demonstrates that it can capture an underlying signal in the data rather than randomly distributing data points. For each comparison, the parameters of the 2 runs are drawn from a predefined set (Table 2) selected randomly with replacement while not allowing the same parameters to be used within 1 comparison. As we have to decide on a dataset without prior training on similar data, the choice of multiple parameters is more reliable than selecting any single parameter. Additionally, in the interest of performance and computational time, we suggest 3 comparisons to be used instead of exhaustively using every parameter. Depending on c , we generate variations of the base clustering algorithms (package kernlab v0.9–29), along with the various distance measures and clustering categories they belong to. Within each pair of clustering runs, the agreement is calculated using the adjusted Rand Index (package fossil v0.3.7), the corrected-for-chance version of the original Rand index [43], which is based on the number of times any pair of points is partitioned in the same subgroup throughout different clustering runs. To calculate the agreement within each clustering algorithm (spectral, k -means, hierarchical), we are considering pairs of runs using the same algorithm but different parameters. For those pairs, the agreement is averaged across clustering runs and k number of clusters tested. The algorithm that presents the highest intramethod agreement over a logical range of clusters ($k \in [2, x]$) is noted as the most appropriate clustering of the samples based on a detected signal. A logical range of k is considered a set of successive k s (where $k \geq 2$) that is most probable to exist within our data, often determined by prior knowledge of the data, previous studies, or domain expertise. This selection procedure is mainly affected by the type and size of the data, leading similar datasets to opt for the same method due to the specific mathematical formulas within each algorithm.

Spectral clustering algorithm [41]

Given a set of points $S = \{s_1 \dots, s_n\}$ in \mathbb{R}^l that we want to cluster into k subsets:

1. Form the affinity matrix $A \in \mathbb{R}^{n \times n}$ defined by $A_{ij} = \exp(-\|s_i - s_j\| / 2\sigma^2)$ if $i \neq j$, and $A_{ii} = 0$
2. Define D to be the diagonal matrix (where $A(i, i) = 0$) whose (i, i) -element is the sum of A 's i -th row, and construct the matrix $L = D^{-1/2} A D^{-1/2}$
3. Find x_1, x_2, \dots, x_k , the k largest eigenvectors of L (chosen to be orthogonal to each other in the case of repeated eigenvalues), and form the matrix $X = [x_1 \ x_2 \ \dots \ x_k] \in n \times k$ by stacking the eigenvectors in columns
4. Form the matrix Y from X by normalizing each of X 's rows to have unit length (i.e., $Y_{ij} = X_{ij} / (\sum_j X_{ij}^2)^{1/2}$)
5. Treating each row of Y as a point in \mathbb{R}^k , cluster them into k clusters via K -means or any other algorithm (that attempts to minimize distortion)
6. Finally, assign the original point s_i to cluster j if and only if row i of the matrix Y was assigned to cluster j

Hierarchical clustering algorithm (average linkage)

Given a set of points $S = \{s_1 \dots, s_n\}$ that we want to cluster into k subsets:

1. Initialize with n clusters, each containing one data point (s_i)
2. Compute the between-cluster distance $D(r, s)$ as the between-object distance of the two data points in clusters r and s respectively, $r, s = 1, 2, \dots, n$. Let the square matrix $D = (D(r, s))$. Various distances can be used (euclidean, manhattan, canberra, minkowski, maximum).
3. Find the most similar pair of clusters r and s , such that $D(r, s)$ is minimum among all pairwise distances
4. Merge r and s to a new cluster t and compute the between-cluster distance $D(t, k)$ for any existing cluster $k \neq r, s$. Once the distances are obtained, delete the rows and columns corresponding to the old cluster r and s in the D matrix, as r and s do not exist anymore. Then add a new row and column in D corresponding to cluster t .
5. Repeat Step 3 a total of $n - 1$ times until there is only one cluster left (effectively minimizing the number of redundant clusters).
6. Decide on a point to cut the cluster tree created above so as to obtain the desirable number of clusters (k)

K -means [40]

K : kernel matrix, k : number of clusters, w : weights for each point, t_{max} : optional maximum number of iterations, $\{\pi_c^{(0)}\}_{c=1}^k$: optional initial clusters

1. If no initial clustering is given, initialize the k clusters $\pi_1^{(0)}, \dots, \pi_k^{(0)}$ (i.e., randomly). Set $t = 0$
2. For each a_i and every cluster c , compute $d(a_i, m_c) = K_{ii} - c$
3. Find $c^*(a_i) = \text{argmin}_c d(a_i, m_c)$, resolving ties arbitrarily. Compute the updated clusters as $\pi_c^{(t+1)} = \{a: c^*(a_i) = c\}$
4. If not converged or $t_{max} > t$, set $t = t + 1$ and go to Step 2; Otherwise, stop and output final clusters $\{\pi_c^{(t+1)}\}_{c=1}^k$

Feature set subsampling

While gene expression data provide measures on the thousands of transcripts in the transcriptome, not all of them may provide discriminative information on the samples and may not be useful for clustering. Moreover, most clustering algorithms are heavily affected by a large number of features both computationally due to input size and in performance due to misdirecting data noise [42]. A common strategy to select interesting and potentially useful RNA features is to measure their variance across samples and select the ones with the highest scores instead of those that are either housekeeping or do not differentiate in our context. In this tool, we exclude RNA features that remain stable across samples and are therefore unable to offer any discriminatory power to our unsupervised machine learning models. Furthermore, the exhaustive feature selection procedure incrementally considers all the genes in the feature set and takes into account the stability of all generated test clusters and number of cluster ranges. This step does not require any deep knowledge or filtering decisions by the user.

Based on this observation, our sample selection step, which is a part of the tool for bootstrap resampling of features presented in Figs. 1 and 2A, first ranks features in a descending order of variance (var function from the Stats R package) across samples, generating a list of the most variable features. Subsequently, multiple datasets of all samples and subsets of features are generated. All

Table 2: The clustering algorithms, their approach category, and the various distance measures tested

Clustering algorithms	Category	Distance measures/kernels	Additional parameters
K-means	Partitioning	Hartigan–Wong, Lloyd, Forgy, MacQueen	—
Hierarchical	Hierarchical	Euclidean, Manhattan, Minkowski, Canberra	Average, complete, median (linkage)
Spectral	Graph Theory	Rbfdot, Polydot, Tanhdot, Laplacedot, Vanilladot, Anovadot, Splinedot	—

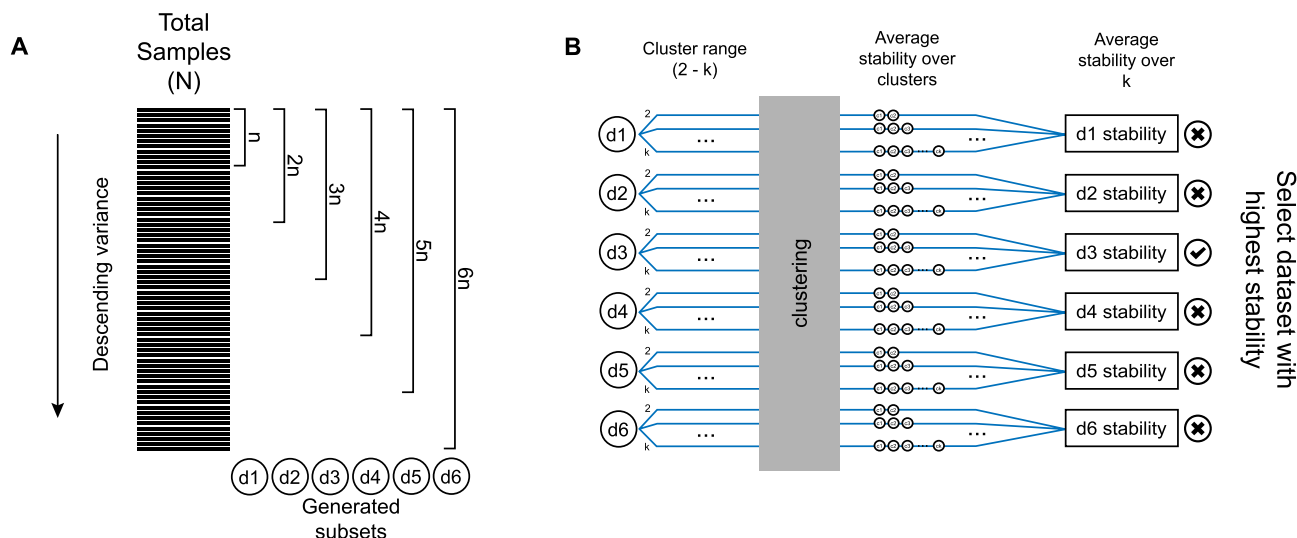


Figure 2: Sample selection overview. (A) Ranking of samples based on their variance across features and the subsequent generation of datasets of increasing size. (B) Calculation of the stability score of each generated dataset. Initially, we select a cluster range to run our clustering method for each dataset. After the clustering procedure, we calculate and average the stability over the generated clusters. Finally, we average the stabilities over k per dataset and determine a final stability score for each dataset. The features of the dataset with the highest stability are the ones that compose the most appropriate set for the downstream pipeline.

subsets draw a different number of features from the top of the variance list with replacement. The first dataset uses a relatively small number of features (n), depending on the total number of features (N) and the granularity of the result desired. The following datasets redraw from the initial list increasing the number of features by n , ending up with $\frac{N}{n}$ datasets.

Stability-based assessment of feature sets

To assess the suitability of each resampled feature set for our clustering, we measure the average stability of the clusters they generate per run when a clustering method is applied over a range of k s (Fig. 2B). First, the clustering range, where the stability of each dataset will be calculated, is selected. The lower end of this range is 2, the minimum possible number of clusters and the upper case is selected by the user. For each dataset, we generate the bootstrap stability for every k within range (clusterboot function in R package fpc v2.2–3 using nonparametric bootstrap, spectral clustering, and 25 resampling runs). To calculate each bootstrap stability score, the data are randomly sampled with replacement and clustered internally using a spectral approach. We then compute the Jaccard similarities between the original clusters and the most similar clusters in the resampled data. The above procedure results in a stability score for each k and each dataset. We then calculate the final stability of each dataset by averaging the stability over k . The genes that comprise the single dataset with the highest stability are the ones that compose the most appropriate set for the downstream analysis.

Choosing k number of clusters

Most clustering methods require the number of k clusters to be defined as a parameter before the application of the algorithm on the data. The lack of a concrete way to determine the real number of clusters in a dataset led many studies to base their estimation on field/prior knowledge or various estimation methods such as the Silhouette score [44]. However, each method favors different aspects of the generated clusters (i.e., how compact clusters are and how far apart cluster centers are) and therefore suits specific datasets and may introduce bias toward the selection of clusters. To encompass these different angles in one methodology, avoid the risk of selecting an ineffective index, and present a more general solution, this tool uses an ensemble learning approach (Fig. 1) where multiple internal cluster indexes contribute to the decision-making [45]. This approach prevents any bias from specific metrics and frees the user from making decisions on any specific metric and assumptions on the optimal number of clusters.

Initially, the value of the 15 indexes is calculated for each k within a cluster range of $[2, x]$, where x is a logical upper limit of the number of clusters realistic for our dataset (i.e., the present conditions within a dataset). The means over k are calculated per index and the optimal k is estimated by majority voting of the 14 means that evaluate the compactness and/or the distance between different subgroups. The selection of indexes can be found in Table 3. It is important to note that the most important aspect of determining k is minimum loss of information, which directs us

Table 3: The list of 15 internal indexes used to estimate the optimal number of clusters (k). All indexes are using different formulas to score a partitioning, measuring 1 or both of the following concepts: (a) how compact each cluster is and (b) how well the clusters separate. For each index, we present which value is preferred (min or max) and its source. For the formulas: k = number of clusters, n = number of data points, E^T = sum of the distances of all the points to the barycenter G of the entire dataset, E^W = sum of the distances of the points of each cluster to their barycenter, N_B = pairs constituted of points that do not belong to the same cluster, N_W = pairs constituted of points that belong to the same cluster, $N_T = N_W + N_B$, S_W = sum of the N_W distances between all the pairs of points inside each cluster, S_{MIN} = sum of the N_W smallest distances between all the pairs of points in the entire dataset, S_{MAX} = sum of the N_W largest distances between all the pairs of points in the entire dataset, S_B = sum of the between-cluster distances, α = weight equal to the value of average scattering of clusters obtained for the partition with the greatest number of clusters.

Internal index	Ideal	Formula	Source
Calinski–Harabasz	Max	$\frac{(\text{between cluster variation})}{(\text{within cluster variation})}$	[46]
Dunn	Max	$\frac{\min(\text{intercluster distance})}{\max(\text{distance between all pairs})}$	[47]
Pbm	Max	$\frac{1}{k} \times \frac{E^T}{E^W} \times D_B$	[48]
Tau	Max	$\frac{\text{concordant pairs} - \text{discordant pairs}}{\sqrt{N_B * N_W * \frac{N_T(N_T - 1)}{2}}}$	[49]
Gamma	Max	$\frac{\text{concordant pairs} - \text{discordant pairs}}{\text{concordant pairs} + \text{discordant pairs}}$	[50]
C index	Min	$\frac{S_W - S_{MIN}}{S_{MAX} - S_{MIN}}$	[51]
Davies–Bouldin	Min	$\frac{1}{n} * \sum_{i=1}^n \max(\frac{\text{clusters scatter difference}}{\text{cluster separation}})$	[52]
McClain–Rao	Min	$\frac{N_B}{N_W} * \frac{S_W}{S_B}$	[53]
sd_dis	Min	$a * (\text{avg scattering for clusters}) + \text{total separation between clusters}$	[54]
Ray–Turi	Min	$\frac{1}{n} * \frac{\text{within-cluster dispersion}}{\min \text{ of the sq. distances between all the cluster barycenters}}$	[55]
g_plus	Min	$\frac{2 * s^-}{N_T(N_T - 1)}$	[56]
Silhouette	Max	Average distance between clusters	[44]
s_dbw	Min	Mean dispersion of clusters + between cluster density	[57]
Compactness	Max	Intracluster distance	[58]
Connectivity	Max	The extent by which the items are placed in the same cluster as their nearest neighbors in the data space	—

to overestimate and not underestimate k [42] while interpreting the voting results. Furthermore, cases that present only a single k as the optimal number of clusters should be treated with caution in case they are a result of a biased dataset. Omada scores each k within the set range $[2, x]$ enabling the user to observe the most probable number of clusters according to the above criteria (function `clusterVoting()`). This prediction is agnostic to any other type of biological information except gene expression. However, after the clustering groups are formed, it is advised for the user to investigate whether they represent relevant to the specific case biological subgroups.

Optimal parameter tuning

Previous steps have selected the optimal method, number of features, and clusters. To perform the optimal clustering, we automate the selection of parameters for each method so that manual tuning is not required. Toward that goal, we use cluster stabilities to decide on the parameters (which depend on the specific algorithm, i.e., kernels in k -means and spectral clustering, linkage method in hierarchical clustering) selected by this toolkit. All available parameters (Table 2) participate in the selection procedure where we measure the average bootstrap stability of the clusters (`clusterboot` function in R package `fpc` v2.2–3) using the previously determined optimal k and feature set for each parameter. The parameter that produces the highest stability is used for the optimal clustering run.

Test datasets

Seven datasets were used to validate different capabilities of the Omada package. First, 2 datasets were simulated by Omada's functions. Function `feasibilityAnalysisDataBased()` was used to generate a multiclass dataset with 359 samples and 300 genes based on the contents and dimensions of the original RNA-seq data [18] and composed of 5 groups of samples drawn from 5 different distributions with means (5, 16, 27, 38, 50) and SD (1, 3, 5, 7, 10), representing the 5 classes. Function `feasibilityAnalysis()` simulated a single-class dataset of 100 samples and 100 genes drawn from a single distribution. For the multitissue pan-cancer dataset, we downloaded RNA-seq expression data for 2,244 samples and 253 genes representing 3 types of cancers: breast ($n = 1,084$), lung ($n = 566$), and colon/rectal ($n = 594$) downloaded through `cbioportal` [59] from The Cancer Genome Atlas (TCGA) PanCancer Atlas [60] (PANCAN). The mRNA expression was in the form of z-scores relative to normal samples where we applied an extra step of arcsine normalization. This type of transformation calculates a (proportional) pseudocount per gene instead of using 1 constant overall pseudocount, which creates a compression effect that also accommodates genes with low expression values. After filtering for tissue-specific genes [61] for the 3 cancer types, we retained 243 genes. An additional TCGA LUAD dataset was acquired from the GDC data portal [62]. Only patients with adenocarcinoma, not otherwise specified (NOS), whose primary tumor site was bronchus and lung with available transcripts per kilobase million (TPM) data, were included. No missing information on gen-

der, ethnicity, race, and vital status was allowed. To reduce complexity, we additionally removed patients with more than 1 TPM file. Altogether, 240 patients with unique TPM data were merged and genes with average TPM fewer than 2 or greater than 500 were removed, resulting in 16,088 genes remaining after quality control to be used for further downstream analysis. Next, we used a pulmonary arterial hypertension (PAH) dataset (25,955 genes) generated from 359 patient samples with an idiopathic (IPAH) or a heritable (HPAH) nature. The transcriptomic data can be found in the EGA (the European Genome-phenome Archive) database under accession code EGAS0000100553265 [63] (restricted access) and all preprocessing details and parameters used can be found in [18]. We also used an RNA dataset from the whole blood of 238 mothers during midgestation (26–28 weeks of pregnancy), where a whole transcriptome library was constructed based on Illumina's standard protocol and quantified using real-time PCR. Sequencing was based on an Illumina HiSeq 4000 system. Read counts were extracted from GEO (accession number GSE182409 [64]) and were then read into R and converted into TPM using the *convertCounts* function available in the *DGEobj.utils* package. For the purpose of clustering, we mapped the TPM dataset to the list of 24,070 genes used in the PAH dataset described in a previous section. Finally, we downloaded single-cell peripheral blood mononuclear cell (PBMC, 8k and 4k) data from a single healthy donor from the 10X genomics website in cellRanger output. UMI reads were merged and preprocessed using standard Seurat 4.3.0 [65] pipeline. Briefly, cells with percent mitochondria exceeding 5%, feature counts below 200 or exceeding 2500, and empty droplets or doublets were excluded from further analysis. This procedure aimed to identify and select the top 2,000 most variable genes for subsequent downstream analysis, which included data scaling and PCA analysis. The goal of these additional steps was to choose the top 10 principal components (PCs) capable of explaining at least 75% of the variation for clustering analysis. For cluster identification, we initially constructed a K-nearest neighbor (KNN) graph and a shared nearest neighbor (SNN) graph. Subsequently, the Louvain method was applied using default settings.

Cell types identification involved a consensus approach, combining insights from 4 different methods: the candidate marker method based on the Garnette database [66], RCAv2 (utilizing GlobalPanel), scType [67, 68] (leveraging the Immune system database), and DISCO [69] (focusing on blood tissue). Subsequently, cell type-specific quality control (QC) measures were implemented to eliminate any outlier cells within each identified cell type. In total, 9,303 samples with 19,171 features were retained, covering 7 major cell types. For the clustering analysis, we deliberately selected only 4 clusters, emphasizing those where the majority of cells predominated, for further in-depth analysis.

Evaluation of runtime and memory usage

Scalability performance was assessed by incrementally increasing the number of samples and features while maintaining the upper limit of clusters (k_{max}). Each combination of samples and features underwent 10 iterations, and the average runtime and memory usage were computed. Runtime measurements were conducted using the `sys.time()` function, capturing the time difference between the initiation and completion of Omada's pipeline. Concurrently, memory usage was evaluated using the `memory_used()` function from the "pryr" package in R. This comprehensive analysis provided insights into how adjustments in the parameters such as the number of samples and number of features can influence both the computational time and memory requirements of

Omada's processing. When benchmarking Omada on simulated datasets, the upper number of clusters was set to 3. Analyses were run on a Linux-based operating system, Intel 8268 48-core platform at 2.9 GHz Nvidia V-100.

Benchmarking Omada against other clustering approaches

To benchmark Omada against other clustering approaches, we performed clustering analyses using the algorithms (hierarchical, k-means, SOM, AP) used in [13] on the TCGA dataset (3 cancer types, LUAD, BRCA, COAD) as described in section "Test datasets." We extracted the resulting clusters for $k = 3$ and calculated the adjusted Rand Index to assess the similarity between the clustering results of Omada and each of the additional algorithms.

Results

Omada was applied to 7 diverse gene expression datasets to demonstrate its utility in guiding cluster analysis and identifying plausible subgroups of samples. Two datasets were simulated by our tools. The simulated dataset with multiple distinct classes was used to determine Omada's ability to accurately estimate k with reasonable stability when we know the existence of sample classes. In contrast, samples in the single-class simulated dataset were drawn from a single class and used to demonstrate the toolkit's ability to point toward the lack of sample subgroups by indicating inconclusive low scores throughout the analysis. A multitissue PANCAN dataset was introduced to assess Omada's capability to generate signal-based clusters that closely follow the tissue-specific patient sample distributions. A second TCGA LUAD dataset was also included to determine whether Omada can accurately recapitulate similar outcomes in terms of overall survival based on the cluster membership identified. To determine whether Omada can identify distinct heterogeneous subgroups from data without any prior classification information but potential present heterogeneity, we used a whole-blood RNA-seq dataset from patients with PAH [18]. In addition, implementation of the toolkit on a whole-blood expression dataset collected from women during pregnancy (GUSTO) was included to demonstrate a case with potential technical biases and no known subgroups since it is composed of healthy participants. Finally, single-cell PBMC data from a healthy donor were incorporated to assess Omada's capability in handling high-dimensional datasets. This addition served as a means to assess scalability performance in terms of both runtime and memory usage.

For the above, we measured its consistency on algorithm, feature, and number of clusters (k) selection and the stability of the generated clusters for a particular k ($stability_k$) and the average across k s ($stability_{avg}$). It is important to note that the value of this validation is derived from the fact that unstable clusters should not be interpreted as this instability comes from problematic data or an incorrect approach. However, cluster stability only provides a mechanistic way to assess the underlying data structure and further information is required to fully biologically validate the clusters [38].

Identifying known clusters

Single-class dataset: Homogeneously simulated dataset

To demonstrate Omada's ability to identify datasets without any present clusters where all patients belong to 1 class, we used the single-class simulated dataset (see "Test datasets" in Methods). All potential k of 2 or higher achieved low scores with

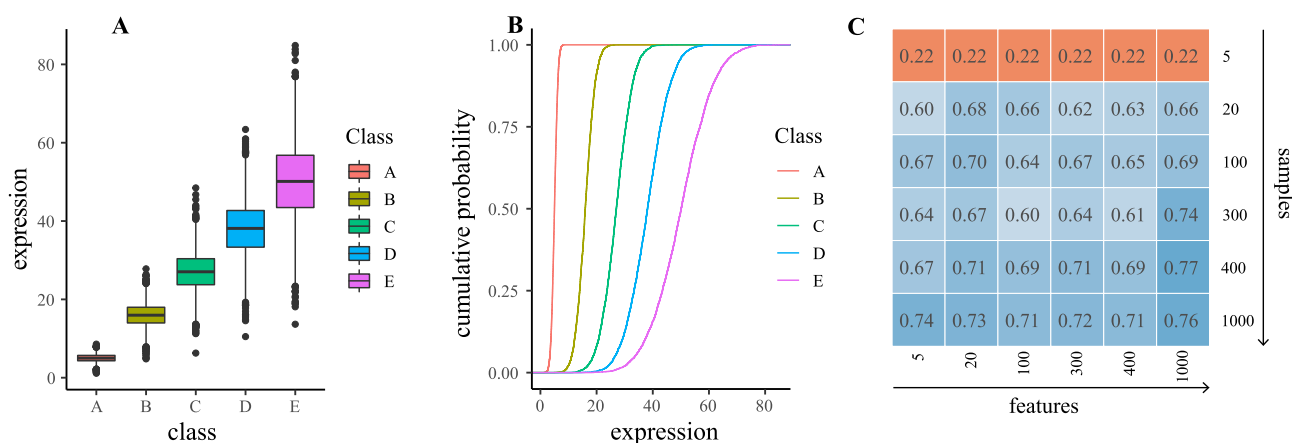


Figure 3: Performance criteria for single-class simulated dataset. The results demonstrate low scores for the majority of steps. (A) The average partition agreement of all 3 algorithms below the 52% mark, indicating very unstable clustering runs overall. Using spectral clustering (highest partition agreement) for all downstream steps, in (B) we observe that the stability of every possible subset of genes does not surpass 51.3% underlying overall unstable clusters. (C) Five clusters as the first estimate (voted by 8 metrics), significantly different from the 1 class this dataset contains (in this case, 14 scores were used as certain metrics are [infrequently] unable to produce a score and therefore omitted from voting).

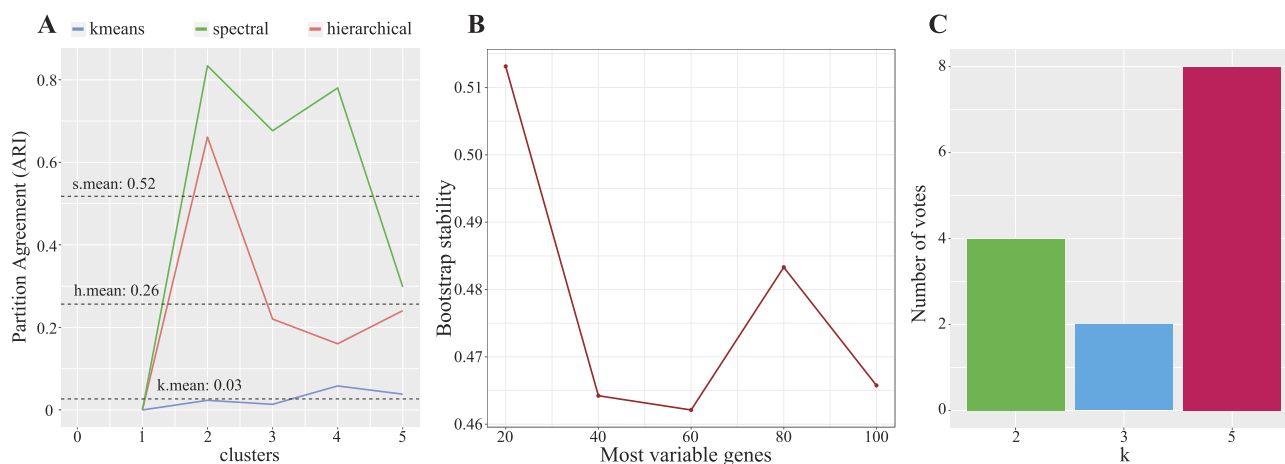


Figure 4: (A) Expression boxplots for the 5 clusters showing the means and standard deviations. (B) The cumulative probability (as calculated from the empirical cumulative distribution function) for the 5 clusters calculated by a 2-sided Kolmogorov–Smirnov test. (C) Average over- k stabilities for simulated datasets of increasing sample and gene numbers. A small number of samples consistently provides extremely unstable clusters (orange) while increasing both numbers consistently produces datasets that pass the stability threshold of 60% (blue).

average and maximum stabilities of 45% and 55%, respectively (Supplementary Table S1). It is recommended to avoid clustering analysis on such low score datasets and instead opt for scores of at least 60%. Ideally, stabilities of 80–90% are considered very strong [70], but the potential of several signals in transcriptomic data and the exploration across multiple k generally decreases the output stability to an acceptable threshold of 60–70%. Next, Fig. 3A shows the overall low partition consistencies (averaged over all tested k) for all algorithms with spectral average partition agreement of 52%, k -means average partition agreement of 3%, and hierarchical average partition agreement of 26%. With the best-performing algorithm showing an agreement of around 50%, we can assume that the tested algorithms are randomly assigning memberships; therefore, we cannot achieve a robust model with the current data. When using spectral clustering to select the most appropriate set of genes, the cluster stability rapidly dropped below 50% when using more than 20 genes (Fig. 3B), indicating that the algorithm got worse in assigning memberships as we considered more simulated genes. Finally, the ensemble voting step showed the majority of the votes supporting 5 clusters (Fig. 3C), a signif-

icant variation from the single simulated class of this dataset. In such unexpected outputs, one should examine the generated metric scores (Table 3; Supplementary Table S3).

Multiclass dataset: Five distinct simulated expression classes representing heterogeneity

Omada's basic function is to help identify samples that come from different sources and group together samples that come from the same source. Toward that end, we simulated a dataset with 5 sets of expression profiles with 50 samples each and 120 genes sourced from 5 unique distributions of expression data that represent heterogeneity within our samples. The means and standard deviations of each class are presented in Fig. 4A, depicting the expression differences. Additionally, the empirical cumulative distribution functions (ECDFs) of the 5 simulated classes (Fig. 4B) as well as the high average Kolmogorov–Smirnov distances ($D_{avg} = 88.3%$, Supplementary Table S2) show distinct differences between the distributions in respect to the expression in the simulated RNA-seq dataset. To demonstrate the effect of different sample and gene numbers, multiple datasets were simulated with an increas-

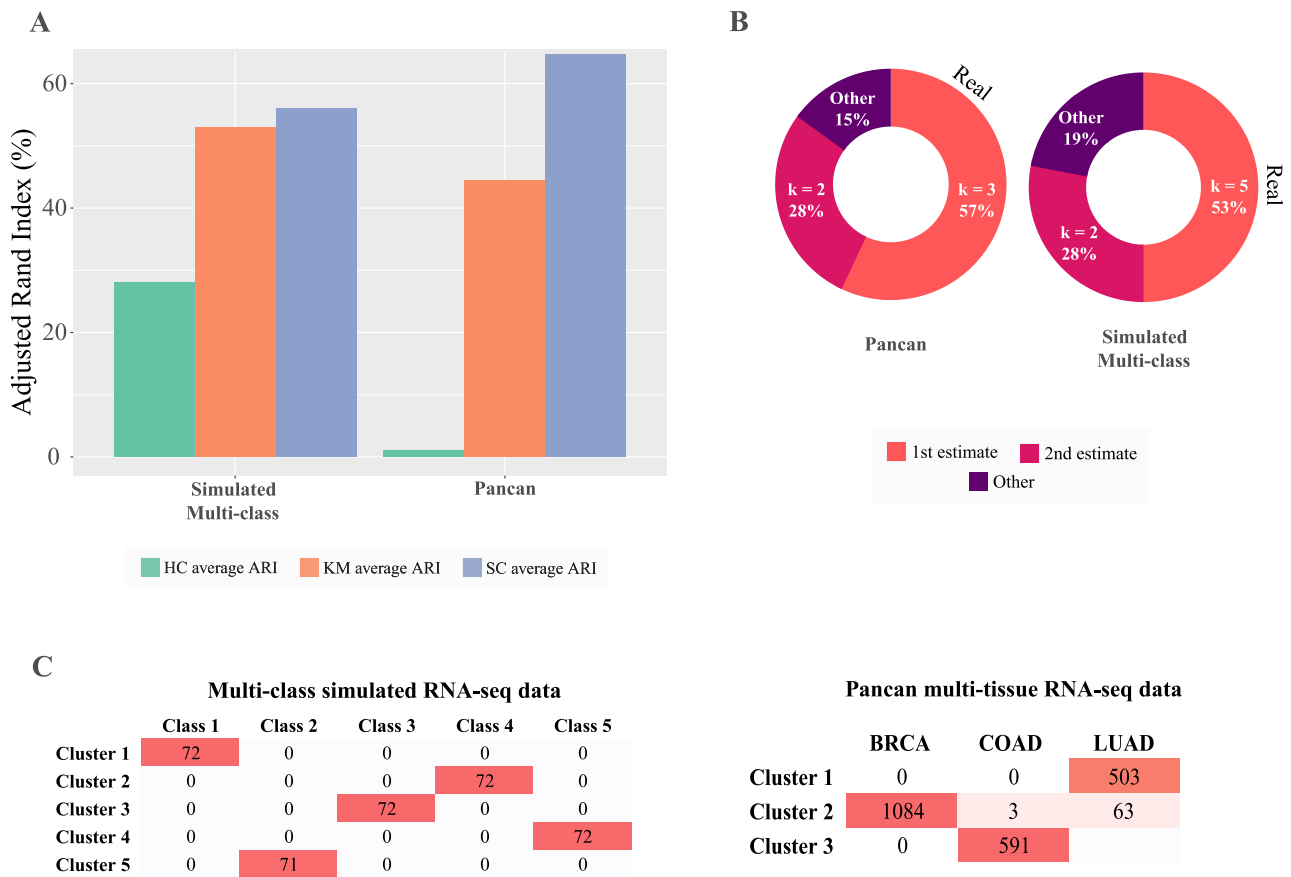


Figure 5: Performance criteria for 2 heterogeneous datasets, simulated multiclass and PANCAN dataset. The multiclass dataset contains artificial samples from 5 distinct clusters and the PANCAN dataset is composed of 3 different cancer types presenting biological heterogeneity. (A) The agreement between the predicted and true clusters (Adjusted Rand Index) from 3 different clustering algorithms (HC: hierarchical; KM: k-means; SC: spectral clustering) applied to the 2 datasets. (B) The real number of clusters for the dataset (black text) and the 2 most likely number of clusters k , with estimates of their percent probability. (C) The contingency tables of the combinations between generated clusters (first estimates) and real classes in the datasets. Darker red color intensity denotes higher frequency.

ing number of samples and genes (Fig. 4C). The calculated cluster stabilities, where each value represents the stability over a range of k and a specific number of samples and features, show 5 or fewer samples per class provide highly unstable and unreliable clusters. The minimum acceptable stability threshold of 60% was achieved with at least 20 samples and a reliable stability of 75% was achieved using 1,000 samples. The majority of metric scores are worse in testing single-class instead of multiclass simulations (Supplementary Table S3), especially in lower compactness and shorter distance between clusters.

To test the ability of the clustering tools to produce stable clusters in various contexts, we first apply them in sequence on strategically simulated data. The data are composed of distinct classes (based on class mean m_{class} and standard deviation sd_{class}), and due to that strong signal, our tools are expected to determine an accurate k with reasonable stability, scoring above 60%. To allow for a more direct comparison, we used a multiclass simulated dataset from random sampling of real RNA-seq data [18]. When considering ranges of k , we are using [2, 6] clusters to observe a broader range of results for comparison reasons. The clustering feasibility tool showed that the highest stability was 78% (Supplementary Table S4), providing a strong indication of stability across our clusters. Since we selected a limited range of $k \in [2, 6]$ where the stability should remain high, the averaged-over-every tested k -stability ($stability_{avg}$) of 72% indicates a dataset of adequate size and class definition to proceed to clustering analy-

sis. It should be noted that when large ranges of k are selected, the average stability will naturally decrease as the calculations will take into account k s much larger or smaller than the actual number of classes in the data. In such cases, the user can review the individual k stabilities generated as part of this tool to conclude whether those values are satisfying (i.e., a minimum of 60%). Next, we calculated the partitioning agreement of 3 clustering algorithms, and spectral clustering showed the highest average score of 56% (Fig. 5A and Supplementary Table S4). Partitioning agreement scores should be interpreted across algorithms applied on the same dataset rather than as absolute values, keeping in mind that a score below 50% represents a random partitioning and subsequently a nonrobust clustering. In the subsequent feature selection step, the highest average stability was registered when using all 300 features ($stability_{avg} = 78%$, Supplementary Table S4), not discarding any feature as they all demonstrated very similar variance due to the nature of the simulated data. Finally, 8 out of 15 internal metrics voted 5 clusters as the optimal number during the k estimation step (Fig. 5B), providing a confident estimation above 50%.

Discriminating cancer types from pan-cancer tissue expression data

An integral capability of Omada is to accurately stratify patients according to any biologically relevant signal present in

Table 4: The ARI of the top k estimate of various datasets with different numbers of cancer types

Algorithm	Real clusters		
	3	5	7
Hierarchical	0.55 ($k = 2$)	0.13 ($k = 2$)	0.19 ($k = 2$)
K-means	0.80 ($k = 3$)	0.54 ($k = 4$)	0.45 ($k = 6$)
Spectral	0.83 ($k = 3$)	0.84 ($k = 5$)	0.58 ($k = 6$)
M3C	0.78 ($k = 3$)	0.85 ($k = 5$)	0.63 ($k = 9$)

Table 5: The ARI of the top k estimate of various datasets with different numbers of simulated classes

Algorithm	Real clusters		
	3	5	7
Hierarchical	0.57 ($k = 2$)	0.9 ($k = 7$)	0.9 ($k = 9$)
K-means	1.00 ($k = 3$)	1.00 ($k = 5$)	0.79 ($k = 8$)
Spectral	0.57 ($k = 2$)	1.00 ($k = 5$)	1.00 ($k = 7$)
M3C	0.57 ($k = 2$)	0.78 ($k = 5$)	0.54 ($k = 4$)

expression data and detect differences stemming from genes, pathways, tissues, and so on. Real multitissue samples are often the focus of exploratory studies as they present cell-type differences but still unknown factors that may discriminate them. Using expression data from multiple cancer types (pan-cancer dataset as described in “Test datasets” in Methods), we expect our tools to identify clusters that are consistent with the samples’ tissues of origin. Due to the different types of tumors, we explored the potential cluster range of [2, 5] for each pipeline step. The clustering feasibility of the dataset (2,244 samples, 243 genes) presented an average stability of 88% and maximum stability of 100% (Supplementary Table S5), providing confidence for the downstream analysis. Spectral clustering showed the highest consistency (partition agreement_{avg} = 63% closely resembling the simulated multiclass dataset, Fig. 5A) and was therefore deemed as the most robust. In this example, hierarchical clustering showed high instability, as shown in Fig. 5A, demonstrating the importance of selecting the appropriate algorithm to create a robust model. According to our selection tool, all 243 genes produced the most stable set of clusters with a stability of 96% (Supplementary Table S5), which, coupled with the high algorithm robustness, indicated a model that most likely detects a signal in the data. Additionally, a very important observation is that all genes were deemed important to produce nearly perfectly stable clusters agreeing with the filtering of genes based on the cancer-type annotations we performed prior to this clustering analysis. The ensemble voting tool estimated our dataset to contain 3 clusters of samples with the support of 57% of the metrics (Fig. 5B). When comparing these results with the simulated 5-class dataset, both achieved higher certainty on the 5 clusters (>50%, Fig. 5B), reflecting the rigid differences between the clusters when dealing with cancer tissues and simulated classes. In the case of the PANCAN partitioning, the breast, lung and colon/rectal samples almost perfectly grouped in their respective clusters (Fig. 5C).

Omada also has the ability to supply additional information for assessing whether the existing dataset lacks signal for potential clustering. In this context, Omada was used to analyze separate TCGA LUAD whole-transcriptome data obtained from

PMID35664309 [62]. The objective was to compare the clustering and overall survival outcomes between Omada and the unsupervised clustering method employed by Pan et al. [62]. We restricted the dataset to 240 patients diagnosed with adenocarcinoma, NOS, and included 16,088 genes after quality control measures. We anticipate that Omada will categorize patients according to their survival or disease states. The feasibility of clustering demonstrated an average stability of 68%, with a maximum stability of 100%, instilling confidence for subsequent analysis. Spectral clustering exhibited the highest average partition agreement at 48%, with the peak agreement occurring when $k = 2$. Through cluster voting, we identified 45 patients in cluster 1 and 195 patients in cluster 2. To assess survival differences between the 2 identified clusters, we employed survival analysis using the Kaplan–Meier method. Remarkably, the identified clusters did not exhibit an overall disparity in survival, as indicated by a P value of 0.4 (Supplementary Fig. S4). This finding aligns with observations in Pan et al. [62]. A plausible explanation for this outcome may be attributed to a lack of signal in the data for clustering, possibly stemming from a low overall average partition agreement score, which falls below 50% based on spectral clustering.

Not all algorithms can detect the true number of clusters

We tested the ability of 4 algorithms to cluster samples that belong to 3 cancer types. As seen in Table 4, hierarchical clustering was not able to identify the 3 cancer types (Adjusted Rand Index [ARI] = 0.55) as it selected 2 clusters as most probable. On the contrary, spectral clustering separated the samples most accurately (ARI = 0.83) with a clear preference for 3 clusters (8/14 votes) closely followed by k -means (ARI = 0.80 and 5/14 votes). Consensus clustering followed with an ARI of 0.78 but a widest spread of estimated k as it selected 5 and 6 clusters as second and third choices, respectively. On the simulated dataset (Table 5), k -means estimated the correct number of clusters and managed to assign the memberships perfectly (ARI = 1). All other algorithms selected a different k and therefore presented a much lower ARI (as comparing a set of 2 clusters with a set of 3 clusters cannot

result in a high ARI). Both Omada and consensus clustering M3C suggest $k = 3$ as the second estimate. However, Omada already supplies the memberships for $k = 3$ as part of the k estimation achieving a perfect ARI of 1 if that partitioning is selected for all 3 algorithms.

RNA-seq data from diseased tissue with unknown heterogeneity

It is important for Omada to be able to robustly identify patient subgroups when heterogeneity for the cohort has not been previously characterized. We applied our tools on such a dataset (PAH dataset as described in “Test datasets” in Methods) to assess whether they can still produce stable clusters that differ in terms of expression profiles and other phenotypic measures. The feasibility for this dataset’s simulation showed an average stability of 61% and a maximum stability of 74% both acceptable to proceed with the clustering analysis (Supplementary Table S6). A notable 13% difference between average and maximum stability provides a positive indication that a specific k might prove significantly more stable downstream. The spectral clustering technique recorded the highest partition consistency (partition agreement_{avg} = 86% and partition agreement_{max} = 96%) when we examined each algorithm’s partition agreement for up to 10 clusters. The bootstrapping subset selection tool estimated the 300 most variable genes as the most stable clustering parameter with a maximum stability of 73% (Fig. 6A), showing an impressive reduction from the initial gene set (25,955) and ensuring the removal of a lot of data noise. According to the ensemble voting tool, 2 clusters were voted by 71% of the internal metrics, followed by $k = 3$ (14%) and $k = 5$ (7%). Despite the strong indication of 2 clusters, $k = 5$ was selected to prevent loss of information occurring when smaller embedded clusters are disregarded. As shown by the downstream analysis, fully presented in [18], selecting the higher k , even as a second estimate, allowed us to detect strong expression profiles. After considering cluster sizes, the 3 predominant subgroups showed significant differences in expression, immunity, and survival profiles as well as risk category distributions (Fig. 6B, C).

RNA-seq data from healthy whole blood tissue

Next we tested how Omada would discriminate samples from healthy individuals from a single tissue type. Generally, in studies based on a dataset with no discernible heterogeneity to be explored (i.e., a dataset without patients of dissimilar outcomes or controls), clustering algorithms may not be robust and may generate variable results. Useful partitioning might still be formed, such as unforeseen disease subgroups, but these observations must be validated. Toward that end, we used the GUSTO RNA whole-blood dataset of 238 mothers obtained during mid-gestation, as seen in “Test datasets” in Methods. During determining clustering potential, our simulated dataset showed stability_{avg} = 56% and stability_{max} = 59% (Supplementary Table S7), a similar low-stability score as in the simulated single class (45% and 55%). We examined a k -range of [2, 5] where spectral and k -means clustering showed very similar internal average partition agreements of 61% and 60% and very high maximum agreements of 93% and 88%, respectively. The extremely high agreement scores should be interpreted with caution as they might not reflect a very strong signal but an underlying bias that partitions samples in similar groups repeatedly, overpowering the parameter changes. The 50 most variable genes were estimated to produce the most stable clustering with maximum stability = 71% (Fig. 6A). Similarly to the agreement scores, a small number of genes driving the most

stable clusters (starting from 24,070 genes) might indicate either a strong expression signal or a preexisting bias. When estimating the number of clusters, 2 (46%) and 3 (40%) clusters were voted by the majority showing a general consensus. Considering all the above strong indications, we need to assess the dataset and the resulting subgroups for potential biases before relying on the cluster memberships. Toward that end and using clinical data, the association results show the dataset might be biased based on technical batches with sequencer machine (Fig. 6E) and flowID (Fig. 6F) presenting significant differences between clusters (1.39e-03 and 2.55e-06, respectively) with hospital location coming close to significance with $P = 0.072$ (Supplementary Table S8). Additional statistical tests and regression analysis with maternal and fetal physiological and clinical phenotypes did not show any association with the clusters. The expression profiles of the 2 clusters show visible differences (Fig. 6D) as do the t-distributed stochastic neighbor embedding (t-SNE) and PCA analyses (Fig. 6H) with the first principal component of the latter explaining 79% of the variance in the GUSTO dataset.

Scalability performance

To evaluate and benchmark Omada’s scalability, we used 2 distinct groups of datasets: simulated datasets (generated by Omada’s feasibilityAnalysis() function) and subsets of a high-dimensional single-cell PBMC dataset. For each dataset, we ran the Omada pipeline on a varying number of samples and features with each combination repeated 10 times. Average runtime and memory usage were recorded and presented in Supplementary Figs. S2–S3, as well as Supplementary Tables S9–S12.

Both types of datasets unveiled an increase in runtime and memory requirement as the number of cells (i.e., samples) analyzed increased while maintaining k_{max} and the number of features. A much smaller increase in time was observed when augmenting the number of features (i.e., genes).

When examining the simulated datasets (3 generated classes) setting 500 samples, we noticed a small runtime difference when using 100 and 1,000 features (4.2 and 4.8 minutes, respectively, Supplementary Table S9). Similarly, the difference in memory used was not substantial with 201 megabytes and 214 megabytes, respectively (Supplementary Table S10). As demonstrated, the increase in number of samples was the only factor that affected Omada’s runtime (Supplementary Fig. S2A) but not memory usage (Supplementary Fig. S2B). Neither of the above was greatly affected by the increase of features (Supplementary Fig. S2C, D).

In the single-cell PBMC data, the average time taken and memory consumption for processing 1,000 samples with 1,000 features amounted to approximately 11.0 minutes (Supplementary Table S11) and 915 megabytes (Supplementary Table S12), respectively. Likewise, the average time taken and memory consumption to analyze 1,000 samples and 1,000 features on the simulated dataset was 20.3 minutes and 220.8 megabytes. Across 10 iterations, memory usage remained consistent, with only slight variations in runtime (Supplementary Fig. S3). These variations can be attributed to the selection of fewer or more genes during the feature selection step. It is important to also note that depending on the sparsity/complexity of the input matrix, there may be variations in memory usage and computational time.

Benchmarking with other clustering algorithms

All tested algorithms except affinity propagation (AP) provided nearly identical subgroups, as shown in Supplementary Table S13. The generated subgroups were compared to the real cancer classes (Supplementary Table S14) with Omada, hierarchical,

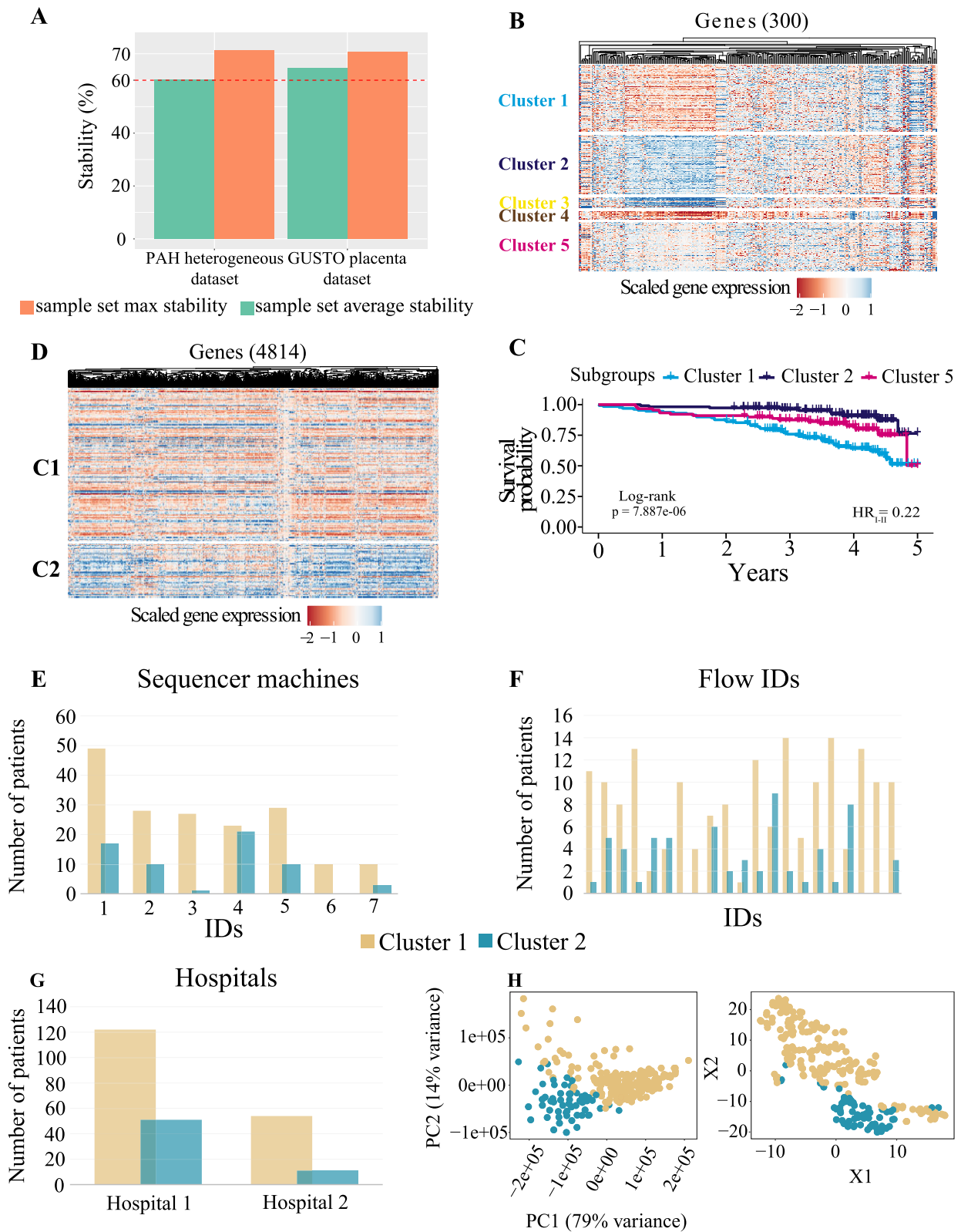


Figure 6: Performance criteria for PAH and GUSTO datasets, which have no known subgroups. (A) The average and max sample set stabilities (percentage) for both datasets. The red dashed line represents the threshold of a stable clustering (60%). The PAH RNA-seq dataset contains expression of IPAH patients with (B) showing the gene expression heatmap and (C) survival profiles for discovered subgroups (subgroups 3 and 4 are omitted as they were not used in downstream analysis due to their size). The GUSTO dataset contains expression from healthy maternal whole blood with (D) showing the gene expression heatmap. The following panels show the distribution of cluster members across (E) sequencer machines (chi-square $P = 1.39e-03$), (F) flow IDs (chi-square $P = 2.55e-06$), and (G) hospitals where the data were collected (chi-square $P = 0.072$). (H) t-SNE and PCA plots of the expression profiles with labeling of the 2 discovered subgroups.

k-means, and SOM demonstrating high accuracy in identifying the cancer types (~85%) and AP showing considerably lower performance (~37%). When applying Omada to a subset of the single-cell data containing 1,000 samples, wherein 250 cells were randomly selected from each of the 4 major cell types (B cells, monocytes, CD4 T cells, and CD8 T cells), along with 10,000 random features, spectral clustering revealed the highest agreement in partitioning when k was set to 2, resulting in a maximum partition agreement of 1. Subsequent cluster voting indicated that 57% of the votes supported a 2-cluster solution, while 28% supported a 3-cluster solution based on internal metrics. However, to avoid information loss, a k value of 3 was selected. Notably, the analysis revealed distinct cluster compositions: cluster 1 comprised cells of B-cell origin, cluster 2 had monocytes, and cluster 3 contained cells originating from T cells, including both CD4 and CD8 T cells. To better distinguish between CD4 and CD8 T cells, a larger dataset is required.

Discussion

Omada is designed to enable the assessment of multiple clustering solutions arising from transcriptomic studies of human samples. With the plurality of unsupervised methods available and their specialized nature, the selection of the most appropriate approach is a multifactor problem, and suggesting a general system of recommendation is impossible without considering each dataset in advance [21]. Many decisions on technical parameters are required from the user of clustering approaches in order to get a meaningful set of subgroups [71]. To assist with this problem, our toolkit initially assesses the potential of a target dataset and provides recommendations for the most appropriate clustering algorithm, gene set, and number of possible subgroups. Omada can also be run as a pipeline and, therefore, does not require deep prior knowledge of the algorithms, parameters, and metrics by the user. All outputs of the pipeline, intermediate and final, are observable, and each step is justified by multiple measures and indices representing widely used clustering techniques. We showed that a user of Omada can conduct analysis to assess the feasibility of applying different clustering algorithms and parameters to different RNA-seq datasets from healthy and diseased tissue samples.

Conducting unsupervised learning of expression datasets is often not a straightforward task as the true heterogeneity of the health condition or tissue of origin is unknown. No methods or metrics can give a definitive answer without validation from external biological or clinical insights, and therefore, the output of each algorithm has to be used with caution. Determining the dataset clustering potential is an indication rather than a clear sign that partitioning the dataset will yield informational subgroups [71]. Additionally, clustering can often contain nondeterministic steps (i.e., k -means initial cluster centers selection [40]) allowing for each function to behave slightly differently between similar runs. In order to evaluate the uncertainty and provide a robust set of tools, Omada has been applied to gene expression datasets where we have prior confidence in the number of clusters (simulated and cancer type data) and where we have little prior knowledge (GUSTO whole blood, PAH). Even in data from healthy tissue, such as from GUSTO whole blood, small technical biases can result in prominent clusters detected by Omada. It is also possible for Omada to hint toward the existence of a single homogeneous cluster by consistently revealing low partition agreement and stability scores across multiple functions, as demonstrated in our single-class datasets. Furthermore, Omada can help in selecting a small group of genes with potential partitioning capabilities as the fea-

ture selection step is expected to greatly reduce the number of genes, which in most cases count to thousands.

This toolkit currently encompasses common clustering techniques, metrics, and test datasets, but its modular nature allows its extension to more specific clustering problems, such as for single-cell and spatial data. Furthermore, the structure of this toolkit allows for additional algorithms and metrics to be added to the pool of clustering methods to be evaluated during an analysis run. Systematic empirical testing of algorithms using Omada's approach may result in more robust and better-justified clustering outputs for a variety of biomedical studies.

Availability of Supporting Source Code and Requirements

Project name: Omada

Project homepage: 10.18129/B9.bioc.omada, <https://github.com/BioSok/omada>

Operating system(s): Platform independent

Programming language: R

Other requirements: R-4.2

License: GPL-3

RRID:SCR_025409

bio.tools ID: omada

Additional Files

Supplementary Table S1. Simulated single-class dataset results, composed of 100 samples and 100 genes drawn from a single distribution.

Supplementary Table S2. Kolmogorov's D statistic for a simulated dataset containing 5 clusters (A, B, C, D, E). Each distance D has a P value of less than $2.2e-16$.

Supplementary Table S3. The scores of all internal indexes used to decide on the ensemble voting of the number of clusters for the multi- and single-class simulated datasets. The scores for the most voted k are presented for each dataset along with the ideal score (min/max) for each index.

Supplementary Table S4. Simulated multiclass dataset results, where distribution is represented by around 120 samples and 3 clusters are used as a default parameter.

Supplementary Table S5. Pancan multitissue RNA-seq dataset results using 3 cancer classes.

Supplementary Table S6. RNA-seq iPAH/HPAH dataset results as shown in [6].

Supplementary Table S7. Whole-blood RNA-seq dataset from pregnant mothers (GUSTO).

Supplementary Table S8. P values of chi-square statistical and generalized linear regression analysis results for clinical variables and maternal phenotypes from the GUSTO dataset.

Supplementary Table S9. Runtimes of Omada applied on simulated datasets with increasing numbers of samples and features.

Supplementary Table S10. Memory usage of Omada applied on simulated datasets with increasing numbers of samples and features.

Supplementary Table S11. Runtime of Omada applied on single-cell RNA-seq datasets.

Supplementary Table S12. Memory usage of Omada applied on single-cell RNA-seq datasets.

Supplementary Table S13. Partitioning agreement (adjusted Rand Index) between Omada and 3 clustering algorithms.

Supplementary Table S14. Partitioning agreement (adjusted Rand Index) between each clustering method and the real cancer types.

Supplementary Fig. S1. Supplementary Figure 1: Steps of the Omada pipeline (A) Calculate the partition agreement from multiple runs per clustering method, (B) For each subset of features calculate the average cluster stability. (C) Select the number of clusters k with the most quality metrics votes (D) Optionally, use a supervised method to extract the clinical signatures based on Omada's cluster memberships and additional clinical metadata

Supplementary Fig. S2. Evaluating scalability performance across different scenarios on multiple simulated datasets: (A) measuring runtime and (B) memory usage as the number of samples per features varies, and (C) measuring runtime and (D) memory usage as the number of features per samples varies. Black line represents average runtime and memory usage across samples or across features.

Supplementary Fig. S3. Evaluating scalability performance across different scenarios on a high-dimensional single-cell PBMC dataset: (A) measuring runtime and (B) memory usage as the number of samples per features varies, and (C) measuring runtime and (D) memory usage as the number of features per samples varies. Black line represents average runtime and memory usage across samples or across features.

Supplementary Fig. S4. Kaplan–Meier survival analysis on the 2 clusters generated by using Omada on TCGA LUAD whole-transcriptome data from PMID35664309.

Data Availability

The expression datasets used in this work can be accessed through the following sources: the 2 simulated, by Omada, datasets (single and multiclass) can be accessed and downloaded [72]. The Pan cancer tissue expression data can be accessed through [73–75]. A second TCGA LUAD dataset can be accessed through the Genomic Data Commons (GDC) Data Portal [76] under dbGAP study accession phs000178. The transcriptomic data used in this study can be accessed through the EGA (the European Genome-phenome Archive) database under accession code EGAS00001005532 [77]. In compliance with the ethics under which these data and samples have been collected, the transcriptomic data are available through restricted access for approved researchers who agree to the conditions of use (i.e., keeping them secure and only using them for approved purposes). To apply for access, please contact cohortcoordination@medschl.cam.ac.uk. You will receive an application form within 30 days. The “UK National PAH Cohort Study Data Access Committee” will review requests within 3 months of receipt of the completed application form and, if approved, provide details for access to the RNA-seq data stored at the EGA. All requesters must agree to the data access conditions found in EGA. The data used to generate statistics, plots, and figures are accessible through our interactive portal found in [78]. The GUSTO expression dataset is available in NCBI Gene Expression Omnibus [79] under the accession numbers GSE182409 (Corresponding Reviewer token number: qjollmeudnofnsv). The Single-cell PBMC 8k and 4k data are publicly available through [80]. Snapshots of our code and other data further supporting this work are openly available in the *GigaScience* repository, GigaDB [81].

List of Datasets Used

- Single-class simulated dataset (100 × 100): Simulated by using specific mean and standard deviation [72].

- Multiclass simulated dataset (359×300): Simulated by using specific means and standard deviations [72].
- Pan-cancer dataset (2,244 × 253): Sourced from 3 cancer type datasets: breast cancer [73], colorectal cancer [74], and lung cancer [75].
- Second TCGA-LUAD dataset (240 × 60,661) [76]: Primary tumor: Lung & Bronchus, Diagnosis: Adenocarcinoma, NOS.
- PAH dataset (359 × 300): can be found in the EGA (the European Genome-phenome Archive) database EGAS00001005532 [77]; needs permission from its Data Access Committee.
- GUSTO dataset (238 × 24,070): Can be found in the GEO database with accession number
- GSE182409.
- Single-cell PBMC dataset (8k and 4k from healthy donor) [80]: Cell Ranger output, pipeline version 2.1.0.

Abbreviations

AIC: Akaike information criterion; AP: affinity propagation; ARI: Adjusted Rand Index; BIC: Bayesian information criterion; BRCA: breast cancer; COAD: chronic obstructive airway disease; EDCF: empirical cumulative distribution function; i/H/PAH: idiopathic/hereditary/pulmonary arterial hypertension; KNN: k-nearest neighbor; LUAD: lung adenocarcinoma; MICL: minimum incremental coding length; mRNA: messenger RNA; PANCAN: Pancreatic Cancer Action Network; PCA: principal component analysis; qPCR: quantitative polymerase chain reaction; SOM: self-organizing map; TCGA: The Cancer Genome Atlas; TPM: transcripts per kilobase million; t-SNE: t-distributed stochastic neighbor embedding.

Acknowledgments

The GUSTO study group includes Allan Sheppard, Amutha Chinnadurai, Anne Eng Neo Goh, Anne Rifkin-Graboi, Anqi Qiu, Arijit Biswas, Bee Wah Lee, Birit Froukje Philipp Broekman, Boon Long Quah, Chai Kiat Chng, Cheryl Shufen Ngo, Choon Looi Bong, Christiani Jeyakumar Henry, Daniel Yam Thiam Goh, Doris Ngiuk Lan Loh, Fabian Kok Peng Yap, George Seow Heong Yeo, Helen Yu Chen, Hugo P. S. van Bever, Iliana Magiati, Inez Bik Yun Wong, Ivy Yee-Man Lau, Jeevesh Kapur, Jenny L. Richmond, Jerry Kok Yen Chan, Joanna Dawn Holbrook, Johan G. Eriksson, Joshua J. Gooley, Keith M. Godfrey, Kenneth Yung Chiang Kwek, Kok Hian Tan, Krishnamoorthy Naiduvaje, Leher Singh, Lin Lin Su, Lourdes Mary Daniel, Lynette Pei-Chi Shek, Marielle V. Fortier, Mark Hanson, Mary Foong-Fong Chong, Mary Rauff, Mei Chien Chua, Michael J. Meaney, Mya Thway Tint, Neerja Karnani, Ngee Lek, Oon Hoe Teoh, P. C. Wong, Peter David Gluckman, Pratibha Keshav Agarwal, Rob Martinus van Dam, Salome A. Rebello, Seang Mei Saw, Shang Chee Chong, Shirong Cai, Shu-E Soh, Sok Bee Lim, Stephen Chin-Ying Hsu, Victor Samuel Rajadurai, Walter Stunkel, Wee Meng Han, Wei Wei Pang, Yap Seng Chong, Yin Bun Cheung, Yiong Huak Chan, and Yung Seng Lee.

Author Contributions

S.K. and D.W. conceived the tools. S.K. undertook computational work and drafted the work with D.W. A.L., C.R., T.P.F., and M.W. participated in the data acquisition of the work. All authors revised it critically for important intellectual content and gave final approval of the version submitted for publication.

Funding

The UK National Cohort of Idiopathic and Heritable PAH is supported by grants from the British Heart Foundation (SP/12/12/29836 and SP/18/10/33975) and the UK Medical Research Council (MR/K020919/1). Additional samples from the Sheffield Teaching Hospitals Observational Study of Pulmonary Hypertension, Cardiovascular and other Respiratory Diseases were supported by British Heart Foundation (PG/11/116/29288). D.W. is supported by the Agency for Science Technology and Research (A*STAR) Use-Inspired Basic Research Award, Academy of Medical Sciences Professorship (APR7_1002) and the Engineering and Physical Sciences Research Council (EP/V029045/1). S.K. is supported by a Donald Heath PhD Studentship award and A*STAR Research Attachment Programme (ARAP) award.

Competing Interests

The authors declare that they have no competing interests.

References

1. Yu NY-L, Hallström BM, Fagerberg L, et al. Complementing tissue characterization by integrating transcriptome profiling from the Human Protein Atlas and from the FANTOM5 consortium. *Nucleic Acids Res* 2015;43:6787–98. <https://doi.org/10.1093/nar/gkxv608>.
2. Keen J, Moore H. The Genotype-Tissue Expression (GTEx) Project: linking clinical data with molecular analysis to advance personalized medicine. *J Pers Med* 2015;5:22–29. <https://doi.org/10.3390/jpm5010022>.
3. Uhlén M, Fagerberg L, Hallström BM, et al. Proteomics. Tissue-based map of the human proteome. *Science* 2015;347:1260419. <https://doi.org/10.1126/science.1260419>.
4. Wang L, Yu X, Wu C, et al. RNA sequencing-based longitudinal transcriptomic profiling gives novel insights into the disease mechanism of generalized pustular psoriasis. *BMC Med Genomics* 2018;11:52. <https://doi.org/10.1186/s12920-018-0369-3>.
5. Neff RA, Wang M, Vatansever S, et al. Molecular subtyping of Alzheimer's disease using RNA sequencing data reveals novel mechanisms and targets. *Sci Adv* 2021;7:eabb5398. <https://doi.org/10.1126/sciadv.abb5398>.
6. Saeidian AH, Youssefian L, Vahidnezhad H, et al. Research techniques made simple: whole-transcriptome sequencing by RNA-seq for diagnosis of monogenic disorders. *J Invest Dermatol* 2020;140:1117–26. e1. <https://doi.org/10.1016/j.jid.2020.02.032>.
7. Tran HTN, Ang KS, Chevrier M, et al. A benchmark of batch-effect correction methods for single-cell RNA sequencing data. *Genome Biol* 2020;21:12. <https://doi.org/10.1186/s13059-019-1850-9>.
8. Xing QR, Cipta NO, Hamashima K, et al. Unraveling heterogeneity in transcriptome and its regulation through single-cell multiomics technologies. *Front Genet* 2020;11:662. <https://doi.org/10.3389/fgene.2020.00662>.
9. Firth AL, Mandel J, Yuan JX-J. Idiopathic pulmonary arterial hypertension. *Dis Model Mech* 2010;3:268–73. <https://doi.org/10.1242/dmm.003616>.
10. Koirala B, Turkson-Ocran R-A, Baptiste D, et al. Heterogeneity of cardiovascular disease risk factors among Asian immigrants: insights from the 2010 to 2018 National Health Interview Survey. *J Am Heart Assoc* 2021;10:e020408. <https://doi.org/10.1161/JAHA.120.020408>.
11. Rivera-Andrade A, Luna MA. Trends and heterogeneity of cardiovascular disease and risk factors across Latin American and Caribbean countries. *Prog Cardiovasc Dis* 2014;57:276–85. <https://doi.org/10.1016/j.pcad.2014.09.004>.
12. Manchia M, Cullis J, Turecki G, et al. The impact of phenotypic and genetic heterogeneity on results of genome wide association studies of complex diseases. *PLoS One* 2013;8:e76295. <https://doi.org/10.1371/journal.pone.0076295>.
13. Vidman L, Källberg D, Rydén P. Cluster analysis on high dimensional RNA-seq data with applications to cancer research—an evaluation study. *PLoS One* 2019;14:e0219102. <https://doi.org/10.1371/journal.pone.0219102>.
14. Ren Z, Wang W, Li J. Identifying molecular subtypes in human colon cancer using gene expression and DNA methylation microarray data. *Int J Oncol* 2016;48:690–702. <https://doi.org/10.3892/ijo.2015.3263>.
15. Sotiriou C, Neo S-Y, Mcshane LM, et al. Breast cancer classification and prognosis based on gene expression profiles from a population-based study. *Proc Natl Acad Sci USA* 2003;100(18):10393–98. <https://doi.org/10.1073/pnas.1732912100>.
16. Lapointe J, Li C, Higgins JP, et al. Gene expression profiling identifies clinically relevant subtypes of prostate cancer. *Proc Natl Acad Sci USA* 2004;101:811–16. <https://doi.org/10.1073/pnas.0304146101>.
17. Wu F, Fan J, He Y, et al. Single-cell profiling of tumor heterogeneity and the microenvironment in advanced non-small cell lung cancer. *Nat Commun* 2021;12:2540. <https://doi.org/10.1038/s41467-021-22801-0>.
18. Kariotis S, Jammeh E, Swietlik EM, et al. Biological heterogeneity in idiopathic pulmonary arterial hypertension identified through unsupervised transcriptomic profiling of whole blood. *Nat Commun* 2021;12:7104. <https://doi.org/10.1038/s41467-021-27326-0>.
19. Xu D, Tian Y. A comprehensive survey of clustering algorithms. *Annals of Data Science* 2015;2:165–93. <https://doi.org/10.1007/s40745-015-0040-1>.
20. Reddy CK, Vinzamuri B. Survey of Partitional and Hierarchical Clustering Algorithms Data Clustering 2018;87–110. <https://doi.org/10.1201/9781315373515-4>.
21. Jamail I, Moussa A. Current state-of-the-art of clustering methods for gene expression data with RNA-seq. In: Travieso-Gonzalez CMed. Applications of Pattern Recognition. 167–169 Great Portland Street, London, W1W 5PF, United Kingdom: IntechOpen, 2020.
22. Ezugwu AE, Ikotun AM, Oyelade OO, et al. A comprehensive survey of clustering algorithms: state-of-the-art machine learning applications, taxonomy, challenges, and future research prospects. *Eng Appl Artif Intell* 2022;110:104743. <https://doi.org/10.1016/j.engappai.2022.104743>.
23. Suzuki R, Shimodaira H. Pvcust: an R package for assessing the uncertainty in hierarchical clustering. *Bioinformatics* 2006;22:1540–42. <https://doi.org/10.1093/bioinformatics/btl117>.
24. Gagolewski M. genieclust: fast and robust hierarchical clustering. *SoftwareX* 2021;15:100722. <https://doi.org/10.1016/j.softx.2021.100722>.
25. Mouselimis L, Sanderson C, Curtin R, et al. ClusterR: gaussian mixture models, k-means, mini-batch-kmeans, k-medoids and affinity propagation clustering, R Package. 2018. <https://mlampros.github.io/ClusterR/>.
26. Azzalini A, Menardi G. Clustering via nonparametric density estimation: the R package pdfCluster. *arXiv [statCO]* 2013.

- <https://cran.r-project.org/web/packages/pdfCluster/index.html>. Accessed 15 February 2023.
27. Walesiak M, Dudek A, Dudek MA. clusterSim package. <https://cran.r-project.org/web/packages/clusterSim/index.html>. Accessed 11 February 2011.
 28. Scrucca L, Raftery AE. clustvarsel: a package implementing variable selection for gaussian model-based clustering in R. *J Stat Softw* 2018;84:1–28. <https://doi.org/10.18637/jss.v084.i01>.
 29. Marbac M, Sedki M. VarSellCM: an R/C++ package for variable selection in model-based clustering of mixed-data with missing values. *Bioinformatics* 2019;35:1255–57. <https://doi.org/10.1093/bioinformatics/bty786>.
 30. Thrun MC, Ultsch A. Clustering benchmark datasets exploiting the fundamental clustering problems. *Data Brief* 2020;30:105501. <https://doi.org/10.1016/j.dib.2020.105501>.
 31. Wang C, Gao X, Liu J. Impact of data preprocessing on cell-type clustering based on single-cell RNA-seq data. *BMC Bioinf* 2020;21:440. <https://doi.org/10.1186/s12859-020-03797-8>.
 32. Eijssen LMT, Jaillard M, Adriaens ME, et al. User-friendly solutions for microarray quality control and pre-processing on ArrayAnalysis.Org. *Nucleic Acids Res* 2013;41:W71–76. <https://doi.org/10.1093/nar/gkt293>.
 33. Bittencourt S. FastQC: a quality control tool for high throughput sequence data. 2010. <https://www.bioinformatics.babraham.ac.uk/projects/fastqc/>. Accessed 13 January 2023.
 34. Bolger AM, Lohse M, Usadel B. Trimmomatic: a flexible trimmer for illumina sequence data. *Bioinformatics* 2014;30:2114–20. <https://doi.org/10.1093/bioinformatics/btu170>.
 35. D'haene B, Hellemans J. The importance of quality control during qPCR data analysis. *Int Drug Discov* 2010;1:18–31.
 36. Baccarella A, Williams CR, Parrish JZ, et al. Empirical assessment of the impact of sample number and read depth on RNA-seq analysis workflow performance. *BMC Bioinf* 2018;19:423. <https://doi.org/10.1186/s12859-018-2445-2>.
 37. Wang J, Yue S, Yu X, et al. An efficient data reduction method and its application to cluster analysis. *Neurocomputing* 2017;238:234–44. <https://doi.org/10.1016/j.neucom.2017.01.059>.
 38. Hennig C. Cluster-wise assessment of cluster stability. *Comput Stat Data Anal* 2007;52:258–71. <https://doi.org/10.1016/j.csda.2006.11.025>.
 39. Hartigan JA. *Clustering Algorithms*. New York, NY: John Wiley; 1975.
 40. Dhillon IS. *A Unified View of Kernel K-Means, Spectral Clustering and Graph Cuts*. Austin: Computer Science Department, University of Texas at Austin; 2004.
 41. Ng AY, Jordan MI, Weiss Y. On spectral clustering: analysis and an algorithm. In: Dietterich TG, Becker S, Ghahramani Z, eds. *Advances in Neural Information Processing Systems 14*. Cambridge, MA: MIT Press; 2002:849–56.
 42. Rodriguez MZ, Comin CH, Casanova D, et al. Clustering algorithms: a comparative approach. *PLoS One* 2019;14:e0210236. <https://doi.org/10.1371/journal.pone.0210236>.
 43. Rand WM. Objective criteria for the evaluation of clustering methods. *J Am Stat Assoc* 1971;66:846–50. <https://doi.org/10.1080/01621459.1971.10482356>.
 44. Rousseeuw PJ. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *J Comput Appl Math* 1987;20:53–65. [https://doi.org/10.1016/0377-0427\(87\)90125-7](https://doi.org/10.1016/0377-0427(87)90125-7).
 45. Polikar R. Ensemble based systems in decision making. *IEEE Circuits Syst Mag* 2006;6:21–45. <https://doi.org/10.1109/MCAS.2006.1688199>.
 46. Calinski T, Harabasz J. A dendrite method for cluster analysis. *Commun Stat Theory Methods* 1974;3:1–27. <https://doi.org/10.1080/03610927408827101>.
 47. Dunn† JC. Well-separated clusters and optimal fuzzy partitions. *J Cybern* 1974;4:95–104. <https://doi.org/10.1080/01969727408546059>.
 48. Pakhira MK, Bandyopadhyay S, Maulik U. Validity index for crisp and fuzzy clusters. *Pattern Recognit* 2004;37:487–501. <https://doi.org/10.1016/j.patcog.2003.06.005>.
 49. Kendall MG. A new measure of rank correlation. *Biometrika* 1938;30:81. <https://doi.org/10.2307/2332226>.
 50. Baker FB, Hubert LJ. Measuring the power of hierarchical cluster analysis. *J Am Stat Assoc* 1975;70:31–38. <https://doi.org/10.1080/01621459.1975.10480256>.
 51. Hubert LJ, Levin JR. A general statistical framework for assessing categorical clustering in free recall. *Psychol Bull* 1976;83:1072–80. <https://doi.org/10.1037//0033-2909.83.6.1072>.
 52. Davies DL, Bouldin DW. A cluster separation measure. *IEEE Trans Pattern Anal Mach Intell* 1979;PAMI-1:224–227. <https://doi.org/10.1109/tpami.1979.4766909>.
 53. McClain JO, Rao VR. CLUSTISZ: a program to test for the quality of clustering of a set of objects. *J Mark Res* 1975;12:456–60.
 54. Halkidi M, Batistakis Y, Vazirgiannis M. On clustering validation techniques. *J Intell Inf Syst* 2001;17:107–45. <https://doi.org/10.1023/A:1012801612483>.
 55. Ray S, Turi RH. Determination of number of clusters in k-means clustering and application in colour image segmentation. In: *Proceedings of the 4th International Conference on Advances in Pattern Recognition and Digital Techniques*. New Delhi, India: Narosa Publishing House; 1999:137–43.
 56. Rohlf FJ. Methods of comparing classifications. *Annu Rev Ecol Syst* 1974;5:101–13. <https://doi.org/10.1146/annurev.es.05.110174.000533>.
 57. Halkidi M, Vazirgiannis M. Clustering validity assessment: finding the optimal partitioning of a data set. In: *Proceedings 2001 IEEE International Conference on Data Mining*. <https://doi.org/10.1109/icdm.2001.989517>. Accessed 15 November 2023.
 58. Song Y. Class compactness for data clustering. In: *2010 IEEE International Conference on Information Reuse & Integration*. Las Vegas, NV, USA: IEEE; 2010:86–91. <https://doi.org/10.1109/IRI.2010.5558958>.
 59. cBioPortal for Cancer Genomics. <https://www.cbioportal.org/datasets>. Accessed 5 January 2022.
 60. Weinstein JN, Collisson EA, Mills GB, et al. The cancer genome atlas pan-cancer analysis project. *Nat Genet* 2013;45:1113–20. <https://doi.org/10.1038/ng.2764>.
 61. Kim P, Park A, Han G, et al. TISSGDB: tissue-specific gene database in cancer. *Nucleic Acids Res* 2018;46:D1031–38. <https://doi.org/10.1093/nar/gkx850>.
 62. Pan S, Meng H, Fan T, et al. Comprehensive analysis of programmed cell death signature in the prognosis, tumor microenvironment and drug sensitivity in lung adenocarcinoma. *Front Genet* 2022;13:900159. <https://doi.org/10.3389/fgene.2022.900159>.
 63. Kariotis S, Jammeh E. BioSok/spectral_clustering_of_IPAH: v1.0.1. 2021. <https://doi.org/10.5281/zenodo.5549872>. Accessed 5 October 2021.
 64. Pan H, Tan PF, Lim IY, et al. Integrative Multi-Omics database (iMOMdb) of Asian pregnant women. *Hum Mol Genet* 2022;31:3051–67. <https://doi.org/10.1093/hmg/ddac079>.
 65. Hao Y, Hao S, Andersen-Nissen E, et al. Integrated analysis of multimodal single-cell data. *Cell* 2021;184:3573–87. e29. <https://doi.org/10.1016/j.cell.2021.04.048>.

66. Pliner HA, Shendure J, Trapnell C. Supervised classification enables rapid annotation of cell atlases. *Nat Methods* 2019;16:983–86. <https://doi.org/10.1038/s41592-019-0535-3>.
67. Li H, Courtois ET, Sengupta D, et al. Reference component analysis of single-cell transcriptomes elucidates cellular heterogeneity in human colorectal tumors. *Nat Genet* 2017;49:708–18. <https://doi.org/10.1038/ng.3818>.
68. Schmidt F, Ranjan B, Lin QXX, et al. RCA2: a scalable supervised clustering algorithm that reduces batch effects in scRNA-seq data. *Nucleic Acids Res* 2021;49:8505–19. <https://doi.org/10.1093/nar/gkab632>.
69. Li M, Zhang X, Ang KS, et al. DISCO: a database of deeply integrated human single-cell omics data. *Nucleic Acids Res* 2022;50:D596–D602. <https://doi.org/10.1093/nar/gkab1020>.
70. Tibshirani R, Walther G. Cluster validation by prediction strength. *J Comput Graph Stat* 2005;14:511–28. <https://doi.org/10.1198/106186005X59243>.
71. Yin H, Aryani A, Petrie S, et al. A rapid review of clustering algorithms. *arXiv [csLG]* 2024. <https://doi.org/10.48550/arXiv.2401.07389>.
72. GitHub—BioSok/OmadaSimulatedDatasets. The two simulated datasets generated for Omada publication. GitHub <https://github.com/BioSok/OmadaSimulatedDatasets>. Accessed 10 August 2023.
73. cBioPortal for Cancer Genomics. https://www.cbioportal.org/study/summary?id=brca_tcga_pan_can_atlas_2018. Accessed 20 May 2022.
74. cBioPortal for Cancer Genomics. https://www.cbioportal.org/study/summary?id=coadread_tcga_pan_can_atlas_2018. Accessed 20 May 2022.
75. cBioPortal for Cancer Genomics. https://www.cbioportal.org/study/summary?id=luad_tcga_pan_can_atlas_2018. Accessed 20 May 2022.
76. Genomic Data Commons Data Portal. <https://portal.gdc.cancer.gov/projects/TCGA-LUAD>. Accessed 12 December 2023.
77. PAH sequencing study. <https://ega-archive.org/studies/EGAS0001005532>. Accessed 1 March 2024.
78. Sheffield University. <https://sheffield-university.shinyapps.io/pah-rnaseq-app/>. Accessed 11 May 2023.
79. GEO—NCBI. <https://www.ncbi.nlm.nih.gov/geo/>. Accessed 11 November 2023.
80. Mastering Biology to Advance Human Health. 10x Genomics. <https://www.10xgenomics.com/>. Accessed 14 December 2023.
81. Kariotis S, Tan PF, Lu H, et al. Supporting data for “Omada: Robust Clustering of Transcriptomes through Multiple Testing.” *GigaScience Database*. 2024. <https://doi.org/10.5524/102537>.