Applied Soft Computing Journal 93 (2020) 106421

Contents lists available at ScienceDirect

Applied Soft Computing Journal

journal homepage: www.elsevier.com/locate/asoc

Robust unsupervised dimensionality reduction based on feature clustering for single-cell imaging data

Daniel Peralta [*], Yvan Saeys

Department of Applied Mathematics, Computer Science and Statistics, Ghent University, Ghent, Belgium
Data Mining and Modeling for Biomedicine group, VIB Center for Inflammation Research, Ghent, Belgium

ARTICLE INFO

Article history:
Received 25 January 2019
Received in revised form 10 April 2020
Accepted 18 May 2020
Available online 23 May 2020

Keywords:
Feature clustering
Unsupervised dimensionality reduction
Single-cell data
Feature selection
Imaging data

ABSTRACT

Biological data, and in particular imaging data, have experienced an exponential growth in terms of volume and complexity in the last few years, raising new challenges in the field of machine learning. Unsupervised problems are of particular relevance, as the generation of labels for the data is often labor-intensive, expensive or simply not possible. However, interpretability of the data and the results is key to extract new valuable knowledge from the large-scale datasets that are studied. This highlights the necessity of adequate unsupervised dimensionality reduction techniques that can lower the computational workload necessary to process the dataset, while at the same time providing information on its structure. This paper describes a framework that brings together previous proposals on unsupervised feature clustering, with the goal of providing a scalable, interpretable and robust dimensionality reduction on single-cell imaging data. The framework integrates several inter-feature dissimilarity measures, clustering algorithms, quality criteria to select the best feature clustering, and dimensionality reduction methods that are built on the clustering. For each of these components, several approaches proposed in previous works have been tested and evaluated on three use cases coming from two different imaging datasets, highlighting the best-performing components. Affinity clustering is applied for feature clustering for the first time. The results were validated using statistical tests, showing that many of the combinations tested lowered the complexity of the datasets while maintaining or improving the accuracy yielded by classifiers applied on them. The analysis highlighted affinity clustering as the best algorithm for feature clustering, with median differences of up to 8.9% and 0.9% in accuracy with respect to FSFS and hierarchical clustering. Representation entropy obtained a median difference of 13.0% and 0.8% with respect to class separability and silhouette index, respectively, as a robust unsupervised criterion to select the cluster set. Dissimilarities based on Pearson's correlation performed slightly better than the alternatives, with a median improvement of 2.8% with respect to the cosine distance.

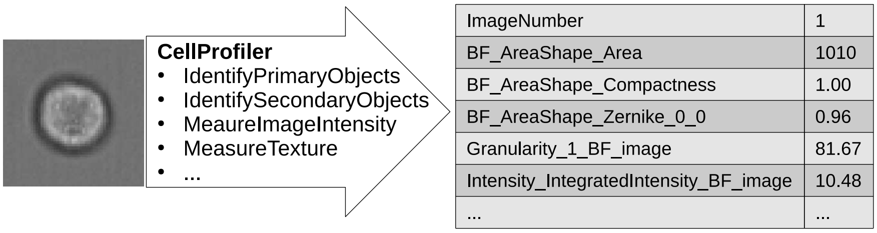
1. Introduction

Few scientific areas have experimented a growth such as computational biology has in the last few decades. As technology evolves, the volume, complexity and variety of biological data that are being generated have grown exponentially [1]. This has allowed to reach crucial breakthroughs, but the proper analysis of such amounts of data also poses a big challenge. Among the various types of data that can be extracted from biological samples, images have always been the focus of active research [2,3]. Various types of technologies allow capturing images at different scales, wavelengths and throughputs, such as microscopy [4], histology [5] and imaging flow cytometry [6].

Machine learning aims to extract valuable knowledge from raw data [7]. Such algorithms are typically categorized into two main groups: supervised methods, which require a label for each input instance and attempt to predict the label for previously unseen instances, and unsupervised methods, which aim to describe the intrinsic structure of the data without the use of labels. Most of these algorithms require the instances to be represented as a set of features; therefore, in the domain of imaging data, it is very common to first extract a vector of features from each image (Fig. 1) and then feed it to a machine learning algorithm. In a biological context, this feature extraction step is crucial for two main reasons. First, it will largely determine the performance of the subsequent analysis; a set of features that does not reflect the relevant characteristics of the image in relation to the problem will not yield optimal results. Second, it allows the experts to obtain an interpretable set of values that summarize an image,

[*] Corresponding author at: Department of Applied Mathematics, Computer Science and Statistics, Ghent University, Ghent, Belgium.
E-mail address: daniel.peralta@irc.vib-ugent.be (D. Peralta).

https://doi.org/10.1016/j.asoc.2020.106421
1568-4946/© 2020 Elsevier B.V. All rights reserved.



ImageNumber	1
BF_AreaShape_Area	1010
BF_AreaShape_Compactness	1.00
BF_AreaShape_Zernike_0_0	0.96
Granularity_1_BF_image	81.67
Intensity_IntegratedIntensity_BF_image	10.48
...	...

Fig. 1. Sketch of the feature extraction process.

which in turn allows them to highlight which properties of the images are relevant to the problem.

Modern feature extraction algorithms and software [8] enable the extraction of hundreds or even thousands of features per image, or even per individual element (such as a cell) within an image. The analysis of data at this scale by means of powerful multi-dimensional algorithms has consistently yielded good results and new insights [9,10]. However, interpretability is often crucial when solving a biological question. A very accurate classifier might be of no use if it is a black box from which no new biological knowledge can be gained. Therefore, it is also important to reduce the dimensionality of the feature vectors extracted from the images by eliminating noisy or redundant features, so as to maximize the performance of subsequently applied algorithms and to enable a consistent interpretation of their results [11]. Moreover, dimensionality reduction leads to a reduction of the computational workload of subsequently applied algorithms. This is especially true for features extracted from biological images, as many hundreds of highly correlated features can be extracted from a single image.

Among the various families of dimensionality reduction approaches, feature clustering [12] is especially appealing in terms of interpretability because the obtained clusters give additional information about which groups of features are similar. Then, these feature clusters are used as a starting point to reduce the dimensionality of the data by one of the three classic approaches: feature selection (selecting a subset of the original set of features), feature extraction (computing a new, smaller set of features by combining the original ones), and feature weighting (assigning weights to each feature according to its importance). Naturally, feature selection allows for a better interpretability because it maintains the original features, which often have a definite meaning or have been designed by experts. On the other hand, feature extraction is more flexible and allows for a more fine-grained elimination of redundancies in the features. One of the most common feature extraction approaches is Principal Component Analysis (PCA), which carries out a linear transformation of the data that maximizes the variance along each new coordinate. Even though feature weighting methods do not fall under a strict definition of dimensionality reduction, they can be used to incorporate information about the relevance of the features, leading to a simplification of the data processing by modifying the behavior of dissimilarity measures, as demonstrated by positive results in previous works [13–15].

Dimensionality reduction algorithms can be either supervised or unsupervised. A multitude of different approaches have been proposed in the scientific literature for the supervised case [16]. On the contrary, unsupervised feature selection did not receive much attention until much more recently [17]. Feature clustering has been used in few approaches so far, especially in the unsupervised case. The proposals in the literature have been shown to yield good results, but are usually aimed at specific problems – such as spectral data [18] or machinery fault detection [19] – and each proposal employs different clustering algorithms, dissimilarity measures, and criteria to select the number of clusters. A family of these proposals are based on entropy measures, which provide a statistically sound way of evaluating feature interdependencies, but at the cost of their discretization, which can be a complex process and can influence subsequent algorithms [20].

There is as yet no systematic study comparing the effect of each individual component of these pipelines on the overall behavior of the feature clustering. Also, some high-performing and efficient clustering algorithms (such as affinity clustering [21]) have never before been applied for feature clustering despite being suitable to the task. Furthermore, little attention is paid in the published literature to the robustness of the clustering methods. The robustness of a clustering algorithm – which we see here as the ability to return highly similar clusters in the case of small changes in the hyperparameters or the samples – is crucial for its applicability in previously unstudied problems. Finally, feature clustering is by construction especially suited to datasets where many features are highly similar to each other, because these features are grouped with each other and reduced [22]. This the case for features extracted from biological images without the specific guidance of a domain expert [8,9]. In such cases, many attributes are computed from each channel of the image regarding different aspects such as size, area, texture, etc., often varying the parameters of the extraction algorithms, which naturally leads to correlated (although complementary) features.

The goal of this paper is to identify how to perform a robust, interpretable dimensionality reduction by means of feature clustering in unsupervised problems. In this line, we present the following contributions:

- A flexible framework to carry out an interpretable dimensionality reduction by unsupervised feature clustering, which is then used to evaluate the various components of the process. For this purpose, we chose three well-known clustering techniques (affinity clustering, agglomerative hierarchical clustering and feature selection using feature similarity), which were evaluated in combination with several feature dissimilarity measures, which are in turn computed using a scalable parallel algorithm.
- Applying affinity clustering [21] on the feature clustering problem. To the best of the author's knowledge, this is the first application of affinity clustering in the feature domain, rather than in the instance domain.
- Comparing different unsupervised criteria to pick the best clustering among those yielded by different hyperparameter configurations for each clustering algorithm. These hyperparameters typically involve the dissimilarity measure and the number of clusters.
- An extensive experimental evaluation of the new sets of features thus obtained. The entire study was replicated on three use cases, built from two public single-cell imaging datasets with very different characteristics. One use case consists of the classification of cells according to their phase in the mitosis cycle using imaging flow cytometry data, the other two consist of a problem The study is done in both an unsupervised and a supervised context, to evaluate

the robustness of all the components of the dimensionality reduction process against variations in the samples and the hyperparameters of the algorithms.

- As a result, guidelines are given concerning the most robust combinations of all the evaluated components.

The results obtained in the experiments highlighted affinity clustering and representation entropy as the options which yielded the most robust clusterings, as well as the highest classification accuracy in a supervised setting. Statistically significant differences were observed between these components and the alternatives that were tested. The analysis of different dissimilarity measures revealed no clear option better than the rest, but did identify the cosine distance as the least performing measure. Dissimilarities based on Pearson's correlation performed slightly better than the rest, with no overall statistical significance.

This paper is structured as follows. Section 2 introduces the main published work about unsupervised feature clustering and single-cell image processing. Section 3 describes the framework designed in this paper for interpretable dimensionality reduction. Section 4 explains the experimental setup designed to test all considered components (namely dissimilarity, clustering, clustering selection criterion and dimensionality reduction method); the results of the experiments are shown and analyzed in Section 5. Finally, the conclusions of the study are presented in Section 6.

2. Background

This section provides an overview of the algorithms and main approaches for unsupervised feature clustering (Section 2.1) and the main types and characteristics of single-cell imaging data (Section 2.2).

2.1. Unsupervised feature clustering

Given a dataset composed of a set of features $\mathcal{A} = \{A_1, \dots, A_p\}$, feature clustering consists on partitioning them into a set of K disjoint clusters $\mathcal{C} = \{C_1, \dots, C_K\}$ such that $\bigcup_{k=1}^{K} C_k = \mathcal{A}$.

The most widely studied proposal in this field was described by Mitra et al. [12]. They proposed a feature clustering algorithm based on the Maximum Information Compression Index (MICI) that receives a single parameter, which determines the size of the biggest cluster. The first cluster is thus formed by choosing the most compact group of features of that size. Then, in every iteration the clusters are formed by grouping increasingly smaller sets of features until no further cluster can be made. Li et al. [23] follow a similar approach, grouping the features by a hierarchical clustering algorithm in combination with MICI.

Many of the more recent approaches use entropy-related similarity measures to compare the features. The method proposed by Bandyopadhyay et al. [24] computes the normalized mutual information between pairs of features to build a graph, and then obtains the densest subgraph to build the clusters. Zhou and Chan [25] describe an algorithm that uses the Maximal Information Coefficient between pairs of features as similarity measure, and then applies a K-modes algorithm. In Wang's approach [26], the mutual information between each pair of features is used to classify the features into irrelevant, weakly relevant and strongly relevant. Irrelevant features can be immediately removed from the dataset; then, a directed acyclic graph with the relevant features as nodes is used to determine which of them should be maintained. Each subgraph corresponds to a feature cluster. The main drawback of entropy-related measures is that they require the discretization of the continuous features of the dataset as an additional preprocessing step. This process can be complex and have a large influence in the behavior of subsequent algorithms [20].

Other proposals employ different dissimilarity measures. Pacheco et al. [19] propose the use of rough sets to characterize the similarity of the features, which are then efficiently clustered following a variant of K-means. However, the proposal is restricted to an online setting within the problem of fault severity classification of rotating machinery. Goswami et al. [27] carries out a feature clustering based on the correlation between the features, and use the entropy of each individual feature to eliminate irrelevant features. For the final feature set, a single feature is randomly picked from each cluster.

Despite the variety among the described methods, they all follow a similar scheme:

1. Computing the inter-feature dissimilarity
2. Grouping the features into clusters, either by a specific or general purpose clustering algorithm
3. Selecting one representative feature for each cluster

Moreover, in most of these methods a single hyperparameter controls the size of the cluster set obtained; therefore, from an external point of view, the search space of the problem is reduced from the combinatorial space of the feature subsets to the one-dimensional space of the algorithmic hyperparameters. The problem still remains as to which hyperparameter value yields the optimal performance for a given dataset, a matter that is usually not evaluated in the aforementioned works.

Finally, one important factor sets feature clustering methods apart from other unsupervised dimensionality reduction techniques: the only computation carried out at the instance level is the generation of the inter-feature dissimilarity matrix. Their computational complexity is linear on the number of instances, and quadratic on the number of features, rather than the other way around, which is the case for many instance-based or self-representation methods. This enables an interpretable reduction of the features within a reasonable computational time even for datasets with extremely large numbers of instances. Naturally, datasets with millions of features become intractable by these methods, but this is not the case with data extracted from single-cell images, where the number of features is typically around the order of a few hundreds.

2.2. Single-cell imaging data

Recent technological advances have brought an explosion of biological data, a trend that has been especially important in the area of imaging [1]. New types of images are being generated, covering a wide range of biological problems such as cancer detection [28], cell cycle classification [29] or compound activity prediction [30]. In particular, most of these technologies advance towards the generation of single-cell data [31], raising new challenges for machine learning to extract new knowledge from it.

The main types of imaging data that yield single-cell resolution are the following:

- **High-content screening (HCS)** [32] automates the process of preparing multiple samples in multi-well plates, and acquiring images of each sample by high-throughput microscopy. As a result, hundreds of thousands of images can be generated in a single screening study. Then, the images can be further processed to extract data at the single-cell level.
- **Imaging flow cytometry (IFC)** [6] allows researchers to extract low-resolution images of individual cells at a high-throughput rate as they sequentially pass between a laser beam and a sensor. The main advance with respect to classic flow cytometry technologies [33] – which produce a vector

of intensities of certain fluorescent markers for each cell – is the presence of morphological information, that reflects the distribution of the markers across the cell.

In both cases, biological samples might be stained with fluorescent markers prior to capturing of the image in order to highlight relevant proteins or structures in the cells. However, both technologies support the acquisition of so-called label-free samples, which do not contain any stains and yield pure morphological information about the cells [34].

When dealing with biological imaging data, the most common approach consists of using specific algorithms to extract a single vector of real-valued features from each image (or each cell within the image). In this context, CellProfiler [8] is among the most used libraries, with the advantage of being open source. CellProfiler provides multiple modules that support the entire pipeline for feature extraction from biological images. First, primary-level objects (typically nuclei) are detected and segmented; then, secondary-level objects (such as the cell edges around each nucleus) are in turn searched around each primary level object and used to segment the images into individual cells. Finally, multiple features are extracted for each cell describing aspects such as its area, size, texture, intensity or pixel correlation.

In this context, it is inevitable to refer to the enormous success obtained by Deep Learning [35] in multiple tasks associated with learning from raw images, avoiding the necessity of an explicit feature extraction process [28,36,37]. However, these are typically considered to be black boxes, and the elevated computational cost of their training often requires high performance Graphics Processing Unit (GPU) platforms. Despite the strong research current towards interpretability and feature attribution in such models [38], it is still widely accepted that more interpretable models are preferable for those problems in which black boxes are not desirable, or do not provide an additional accuracy [39].

3. A generic framework for unsupervised feature clustering

In this paper, we devise a workflow to carry out dimensionality reduction on large-scale imaging datasets in a robust, scalable and interpretable way. To achieve this goal, a parallel framework to compute inter-feature correlations and dissimilarities has been applied (Section 3.1). These dissimilarities can then be passed to several feature clustering algorithms (Section 3.2). Several unsupervised performance measures can be used to evaluate the clusters generated by different algorithms or parameter configurations (Section 3.3). Finally, three different dimensionality reduction alternatives from the feature clusters are described in Section 3.4. The basic scheme of the workflow is depicted in Fig. 2.

Note that, even though this framework has been designed to address the challenges of single-cell imaging data, it could be applied to other types of data with similar characteristics. In general, it can tackle dense datasets within the order of hundreds or thousands of features, with an arbitrarily large number of instances, and because it requires no subsampling it is robust against the presence of rare populations of instances, making it suitable to deal with imbalanced data [40].

3.1. Computing feature dissimilarities

As described in Section 2.1, unsupervised feature clustering algorithms that work on feature similarities are the most suited to large-scale problems because they do not involve pairwise instance comparisons. However, the computation of pairwise feature similarities can also be very time-consuming and even numerically unstable when the number of instances becomes very large [41].

To deal with these challenges, we have used a parallel framework[1] to compute first and second moment statistics of large datasets [42]. The average, and the covariance and correlation matrices of the features are computed by applying the general updating formulae described in [41], which provides a better numerical stability than the commonly used expressions to calculate those statistics.

The parallelization is carried out in a two-level way: Message Passing Interface (MPI)[2] is used to split the computation across several processes (typically, each process is run on a different computer), while OpenMP[3] manages multiple threads within each process. The instances of the dataset are split across the multiple processes and threads, so that each of these computes partial results on a different chunk of the dataset. The results are then aggregated by a master process.

Previously published papers on unsupervised feature selection use various similarity and dissimilarity measures to compare two features x_1 and x_2. In this study, several widely used measures are considered:

- **Pearson's distance** Eq. (1) is based on the well-known Pearson's correlation, and is defined in the $[0, 2]$ interval.
- **Variant of Pearson's distance** Eq. (2): this measure allows us to take into account that two inversely correlated features, which therefore lie at a very high Pearson's distance from each other, actually provide the same information. Therefore, the measure is re-defined so that positive and negative correlation have the same effect on the dissimilarity.
- **Maximal Information Compression Index (MICI)** Eq. (3) is defined in [12], and unlike Pearson's distance does not normalize by the standard deviation of the individual features.
- **Cosine distance** Eq. (4) is commonly used to measure the difference in the direction of vectors, rather than their modulus.

$$d_{\text{Pearson}}(x_1, x_2) = 1 - \rho(x_1, x_2) = 1 - \frac{\text{Cov}[x_1, x_2]}{\sigma_{x_1}\sigma_{x_2}} \tag{1}$$

$$d_{\text{PearsonA}}(x_1, x_2) = 1 - |\rho(x_1, x_2)| = 1 - \left|\frac{\text{Cov}[x_1, x_2]}{\sigma_{x_1}\sigma_{x_2}}\right| \tag{2}$$

$$d_{\text{MICI}}(x_1, x_2) = 0.5\left[\sigma_{x_1}^2 + \sigma_{x_2}^2 - \sqrt{(\sigma_{x_1}^2 - \sigma_{x_2}^2)^2 + 4\text{Cov}[x_1, x_2]^2}\right] \tag{3}$$

$$d_{\text{cosine}}(x_1, x_2) = \frac{x_1 \cdot x_2}{\|x_1\| \; \|x_2\|} \tag{4}$$

Note that, although the rest of these measures are distance metrics, the chosen variant of Pearson's distance does not comply with the self-identity axiom (that is, two completely anti-correlated features will have zero distance despite being different features). This is not a requirement for most feature clustering algorithms, which merely require an arbitrary dissimilarity measure between the features.

3.2. Feature clustering

Interpretability is one of the goals pursued in this study. Feature clustering groups features that are similar to each other,

[1] https://github.com/dperaltac/scalable-statistics.

[2] http://www.mpi-forum.org/.

[3] https://www.openmp.org/.

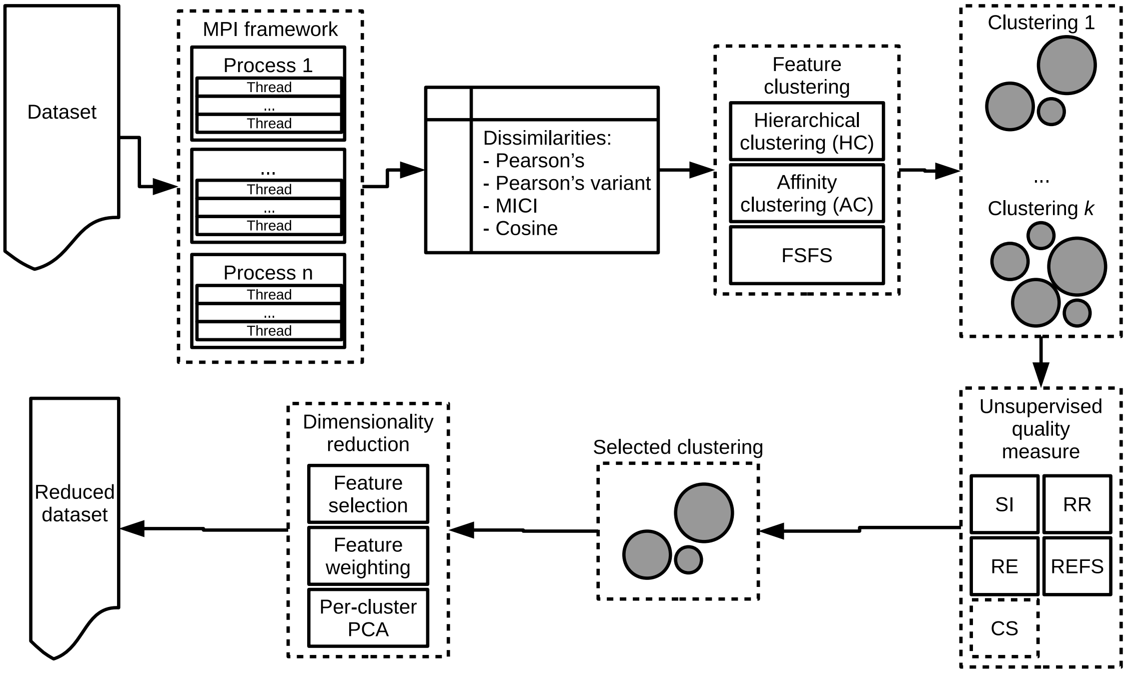


Fig. 2. Proposed workflow for unsupervised dimensionality reduction.

which provides useful information about which features are represented by each feature of the final, reduced dataset. In principle, any clustering algorithm that works on dissimilarity matrices is applicable to build clusters of features. In this study we have considered some of the most relevant ones, due to their efficiency, robustness and their wide use by the scientific community:

- **Hierarchical (agglomerative) clustering (HC)** starts by assigning every feature to a separate cluster. Then, in each iteration the two clusters that are the most similar to each other are joined into a single, bigger cluster. The procedure is carried on until all features belong to a single large cluster. In other words, this algorithm builds a dendrogram-like structure from the features, which in practice allows to select any number of clusters just by fixing the depth at which the dendrogram is cut. In this work, Ward's minimum variance method is used [43] to compare clusters with each other.
- **Affinity propagation clustering (AC)** [21] selects a set of representative features by a message-passing algorithm, which iteratively computes the availability of a feature to be a representative of a cluster along with the responsibility of a feature to be represented by another. Unlike for the agglomerative clustering, the number of clusters cannot be directly chosen, but is instead indirectly determined by a single parameter which sets the initialization of the exemplar preferences.
- **Feature selection using feature similarity (FSFS)** [12] receives a single parameter, which determines the size of the biggest cluster. This is the first cluster to be formed, by choosing the most compact group of features of that size. Then, in every iteration the clusters are formed by grouping increasingly smaller sets of features until no further cluster can be made.

3.3. *Performance measures*

Naturally, the parameters that are used when computing the feature clusters are critical to obtain a robust result. In a supervised context it is possible to compare the performance yielded by different clusterings or dimensionality reductions by evaluating the accuracy of a classifier trained on the reduced data, or by calculating the separability between instances from different classes.

In an unsupervised context, however, the use of such measures is not possible. It is necessary to use performance measures that can be computed from the structure of the data without any labels, which can then be used to select the more adequate parameters for each algorithms, or to fix the number of feature clusters that should be considered for a certain dataset. The following performance measures are considered in this study:

- **Silhouette index (SI)** [44]: this index is computed for every feature, and measures the ratio between intra-cluster and inter-cluster dissimilarity ($a(b)$ and $b(k)$, respectively) as shown in Eq. (5), yielding a value between -1 and 1. The average silhouette index across one clustering can then be used as a measure of quality of the clustering.

$$s(k) = \frac{b(k) - a(k)}{\max\{a(k), b(k)\}} \tag{5}$$

- **Representation entropy (RE and REFS)** [12]: it is defined in Eq. (6), where λ_j are the eigenvalues of the $d \times d$ covariance matrix of a feature set of size d. It reflects the level of redundancy present in a dataset. In this paper, we apply it in two different ways: on the one hand, the RE of each feature cluster should be as low as possible; on the other hand, the RE of the features obtained after the dimensionality reduction (we will note this as REFS) should be as high as possible.

$$\begin{aligned} \tilde{\lambda}_j &= \frac{\lambda_j}{\sum_{j=1}^{d} \lambda_j} \\ RE &= -\sum_{j=1}^{d} \tilde{\lambda}_j \log \tilde{\lambda}_j \end{aligned} \tag{6}$$

- **Redundancy rate (RR)**: it is the average correlation across all pairs of features Eq. (7).

$$RR = \frac{2}{d(d-1)} \sum_{i=1}^{d} \sum_{\substack{j=1 \\ i \neq j}}^{d} \rho_{i,j} \tag{7}$$

- **Variation of Information (VI)** [45]: it is a metric distance between two clusterings $\mathcal{C}$ and $\mathcal{C}'$. Each clustering is described in terms of a random variable $P(k)$ Eq. (8) which contains the proportion of features that are assigned to each cluster C_k. Similarly, $P(k, k')$ represents the joint probability that a feature belongs to cluster C_k in clustering $\mathcal{C}$ and to cluster $C'_{k'}$ in $\mathcal{C}'$ Eq. (9). The mutual information between these variables $I(\mathcal{C}, \mathcal{C}')$ can now be defined as shown in Eq. (10). Finally, the variation of information between $\mathcal{C}$ and $\mathcal{C}'$ is defined in Eq. (11), making use of the entropy function H.

$$P(k) = \frac{|C_k|}{p} \tag{8}$$

$$P(k, k') = \frac{|C_k \bigcap C'_{k'}|}{p} \tag{9}$$

$$I(\mathcal{C}, \mathcal{C}') = \sum_{k=1}^{K} \sum_{k'=1}^{K'} P(k, k') \log \frac{P(k, k')}{P(k)P(k')} \tag{10}$$

$$\begin{gathered} H(\mathcal{C}) = -\sum_{k=1}^{K} P(k) \log P(k) \\ VI(\mathcal{C}, \mathcal{C}') = H(\mathcal{C}) + H(\mathcal{C}') - 2I(\mathcal{C}, \mathcal{C}') \end{gathered} \tag{11}$$

- **Class separability (CS)**: among the various definitions of this supervised measure [46] we use the one shown in Eq. (12), where S_w and S_b are respectively the within-class and between-class scatter matrices, c is the number of classes, π_j is the proportion of instances belonging to class j, $Sigma_j$ is the covariance matrix within class j, μ_j is the mean vector for class j and M_0 is the mean vector across the entire dataset. This definition of the class separability is used, rather than the more general $trace(S_w^{-1} S_b)$, to avoid problems when S_w is singular, which happens relatively often when two very correlated features are selected in the set, or S_b is singular, which happens when the number of classes is lower than the number of features (because S_b is a linear combination of one scatter matrix per class).

$$\begin{gathered} S_w = \sum_{j=1}^{c} \pi_j \Sigma_j \\ S_b = \sum_{j=1}^{c} (\mu_j - M_0)(\mu_j - M_0)^T = \Sigma - S_w \\ CS = trace(S_b)/trace(S_w) \end{gathered} \tag{12}$$

Table 1 summarizes the measures described above showing the context in which they are applied. Note that, even though the ultimate goal is to evaluate the feature clusterings, some of the measures are defined to be applied on feature subsets.

3.4. Dimensionality reduction from feature clusters

Once the features have been grouped into clusters, it is necessary to produce a new set of features that will simplify the input dataset. This can be done in several ways, out of which we consider the following:

Table 1
Summary of the performance measures considered in the proposal.

	Unsupervised	Supervised
Feature clustering (per cluster)	SI, RE	
Feature clustering (per pair of clusterings)	VI	
Feature selection	REFS, RR	CS

- **Feature selection**: a single representative of each cluster is selected. For AC and FSFS, the clustering algorithm itself determines the representative of the clusters. For HC, we select the feature which is the closest to the cluster centroid; in other words, we select the feature that has a minimum average distance to the other features of the cluster.
- **Per-cluster Principal Component Analysis (PCA)**: it is also possible to carry out a feature extraction from the features within each cluster separately, so as to generate one new feature per cluster. In this paper, we applied PCA to extract the first principal component of each cluster. The resulting set of features will be less interpretable than after a mere feature selection where the original meaning of the features is kept. However, it is still possible to know which cluster originated each resulting feature, which yields a higher interpretability than applying a PCA transformation on the entire set of features.
- **Feature weighting**: this is a special case in which no dimensionality reduction is carried out; instead, the features are weighted according to the size of the cluster they belong to. The weight for each feature within a cluster C_k is $1/|C_k|$ so as to weight down large clusters that contain many features, so that the overall contribution of all clusters is approximately the same.

4. Experimental setup

This section describes the methodology followed to evaluate the described proposal (Section 4.1) along with the datasets that have been used for this purpose (Section 4.2).

4.1. Methodology

Throughout the paper, a stratified cross-validation scheme is used to evaluate the proposal in the presence of datasets with imbalanced classes [47]. The number of folds for each dataset has been selected so as to use the same setting as previously published works. For each of the obtained folds, the following steps were applied to obtain the results analyzed:

1. Data normalization. This step is dataset-dependent; details for particular datasets are provided in Section 4.2.
2. Save the full feature set
3. Compute a PCA dimensionality reduction from the full feature set
    (a) Compute PCA
    (b) Keep the principal components accounting for 99% of the variance
4. Evaluate the dimensionality reduction
    (a) Computation of the dissimilarities (4 measures)
    (b) Feature clustering (3 algorithms)
    (c) Selection of the best clustering (3 criteria)
    (d) Apply dimensionality reduction (3 methods):
        - Feature selection
        - Feature weighting
        - Per-group PCA

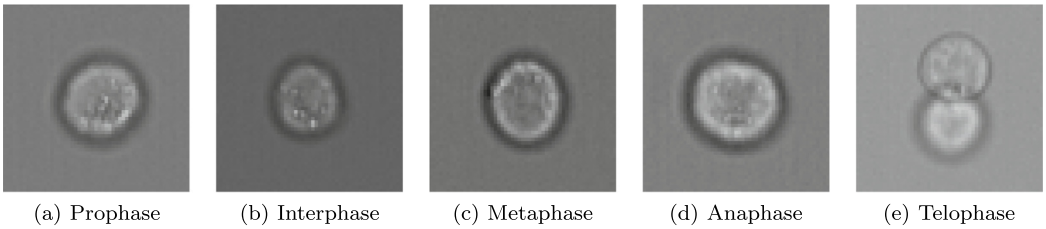

(a) Prophase (b) Interphase (c) Metaphase (d) Anaphase (e) Telophase

Fig. 3. Five classes of the JurkatIFC dataset.

5. For each of the datasets generated in steps 2, 3 and 4:
   (a) Train the classifier
   (b) Test the classifier

The results are analyzed both from an unsupervised and a supervised perspective. First, Section 5.1 will evaluate the robustness of the different feature clustering algorithms and dissimilarity measures tested in the experiments, along with the performance of the dimensionality reduction from an unsupervised point of view. Then, Section 5.2 will analyze the results obtained when multiple different classifiers are applied on the datasets after the dimensionality reduction procedure, therefore describing a supervised setting. These results are compared with those obtained when using the full feature set, and with a PCA reduction keeping the principal components that account for 99% of the total variance of the dataset.

4.2. Datasets

Throughout this study we used two public datasets to compare the behavior of the tested dimensionality reduction methodology. One of them is analyzed from two different points of view, thus producing three separate use cases.

4.2.1. JurkatIFC

This dataset is composed of 32 255 Jurkat cells measured by imaging flow cytometry and first introduced in [29]. The cells are labeled by their cycle phase, for a total of 7 different labels (Prophase, Metaphase, Anaphase, Telophase, G1, S and G2). However, in most works the 3 latter stages are grouped within a single Interphase label, thus producing a 5-class problem, which is the approach we also follow in the present paper (Fig. 3). We represent the dataset by using the features as extracted by the CellProfiler software [8]; in particular, only the 213 publicly available features[4] considered in [29] are taken into account; features referring to some non-informative metadata are removed. Each feature is normalized by subtracting its average and dividing by its standard deviation.

The main particularity of this dataset is that the classes are heavily imbalanced, in correspondence with the relative durations of the cell cycle phases. Thus, most instances belong to the Interphase class (31 542), while very few belong to Anaphase and Telophase (15 and 25, respectively). To avoid the problems that arise with imbalanced datasets [48] without restraining the experiments to a single approach, we applied two different preprocessing methods: Random Undersampling (RUS) [49] and random resampling with replacement. The experiments were carried out following a 10-fold cross validation procedure, as previously done by other authors [29,50,51].

4.2.2. BBBC021

The BBBC021 dataset [9] is part of the Broad Bioimage Benchmark Collection [52]. It contains high-throughput microscopy images of MCF-7 cells treated with different compounds at different concentrations, some examples of which are shown in Fig. 4. In compliance with previous papers that use this dataset, only the labeled images are used in this study. These refer to 103 compound-concentration combinations (38 different compounds, 1–7 concentrations) that are categorized into 12 mechanisms of action (MoA), in addition to the control wells. Furthermore, each of these compound-concentration combinations is replicated 3 times. At the single-cell level, the dataset contains 454 793 cells, whose publicly available CellProfiler features[5] have been used for this study. For the sake of reproducibility, the same 453 features used in [9] are used in this study.

In this dataset, every image contains multiple individual cells and its label refers to the image as a whole. This is a case of multiple instance data [53]. There are multiple approaches that can be followed to deal with multiple instance data; the most used one in single-cell contexts is *profiling*, which consists on summarizing all the cells of each sample into a single vector. In [9], where several profiling methods are compared using the BBBC021 dataset, the best performing method consisted on performing a factor analysis followed by averaging all cells within each sample. However, feature extraction is beyond the scope of this paper; therefore, in this study the profiling is carried out by averaging all cells of each sample, which is the most frequently applied method in the literature [54]. Then, the median of the profiles of the 3 replicates for each compound-concentration in the dataset is obtained, yielding the final representation for the data.

Both the normalization and the cross-validation have been applied as described in [9]. The data was linearly scaled so that the first percentile of the control cells was set to 0 and the 99th percentile was set to 1 for each plate. A leave-one-compound-out cross-validation procedure was applied, yielding a total of 38 folds.

5. Analysis of the experimental results

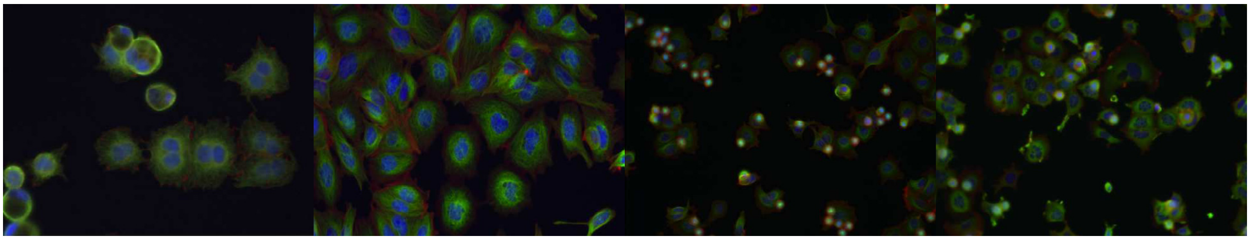
This section presents the results of the experiments carried out for this study. First, Section 5.1 analyzes the characteristics of the computed feature clusters from an unsupervised point of view. Then, Section 5.2 presents the results obtained when applying various classifiers on the datasets before and after dimensionality reduction. Due to the varied nature of the datasets, a separate analysis is carried out for each of them.

5.1. Feature clustering and dimensionality reduction

In this section, the behavior of the feature clustering algorithms and the dimensionality reduction is analyzed, focusing on the robustness of the clustering 5.1.1 and on several quality measures used to select the best clusterings 5.1.2.

[4] http://cellprofiler.org/imagingflowcytometry/ (visited on 2019-09-03).

[5] https://data.broadinstitute.org/bbbc/BBBC021/.



(a) Actin disrupter (b) Aurora kinase inhibitor (c) Monoaster (d) Tubulin destabilizer

Fig. 4. Four examples of the BBBC021 dataset.

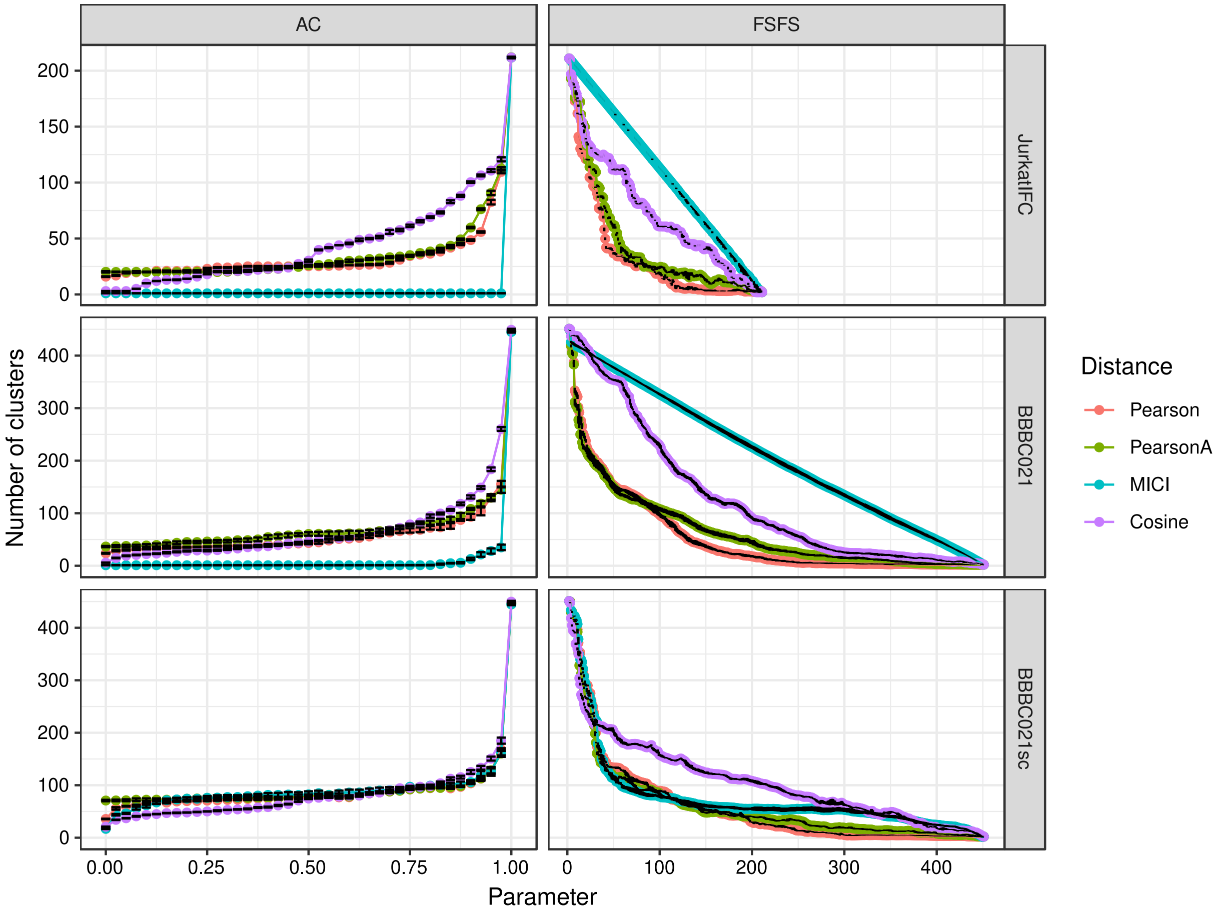


Fig. 5. Number of clusters obtained with different parameters for the feature clustering algorithms.

5.1.1. *Robustness of the feature clustering*

Fig. 5 shows the number of clusters obtained with the different feature clustering algorithms used, as a function of the specific parameters used for each algorithm. Vertical bars depict the standard deviation across the cross-validation folds. Note that HC is not shown, because the number of clusters is a parameter of the algorithm. For all tested algorithms, the number of clusters is almost monotonically determined by a single parameter with a very low variability across the different folds. The general behavior shown in the plots is similar for all clustering algorithms and distances, with the exception of MICI. This distance yields very few clusters when used with AC on BBBC021 and JurkatIFC; on the other hand, when used with FSFS on the same datasets, it always produces a single big cluster and encases the rest of the features into as many clusters of size 1. This behavior is inherent to the algorithm, as described in the original paper of the algorithm [12]; however, it might not yield optimal results when the goal is to achieve a significant dimensionality reduction.

Although Fig. 5 shows a very low variability of the number of clusters across the cross-validation folds, it is also relevant to determine whether these clusters contain different sets of features. Fig. 6 shows the variation of information (as defined in [45]) of the feature clusters across the cross-validation folds. It can be seen in the plots that the variability is higher for FSFS than for HC and AC. The most stable clusters are produced by HC, especially against small variations of the parameters, which was expected due to the nature of the algorithm which builds bigger clusters by joining pairs of smaller clusters. The different dissimilarity measures yield similar results in this case.

5.1.2. *Unsupervised performance measures: analysis and criteria for feature selection*

Figs. 7–10 show the variability of the five performance measures used in this study, as described in Section 3.3. In general, the behavior of the different performance measures is fairly similar for all datasets, clustering algorithms and distance measures.

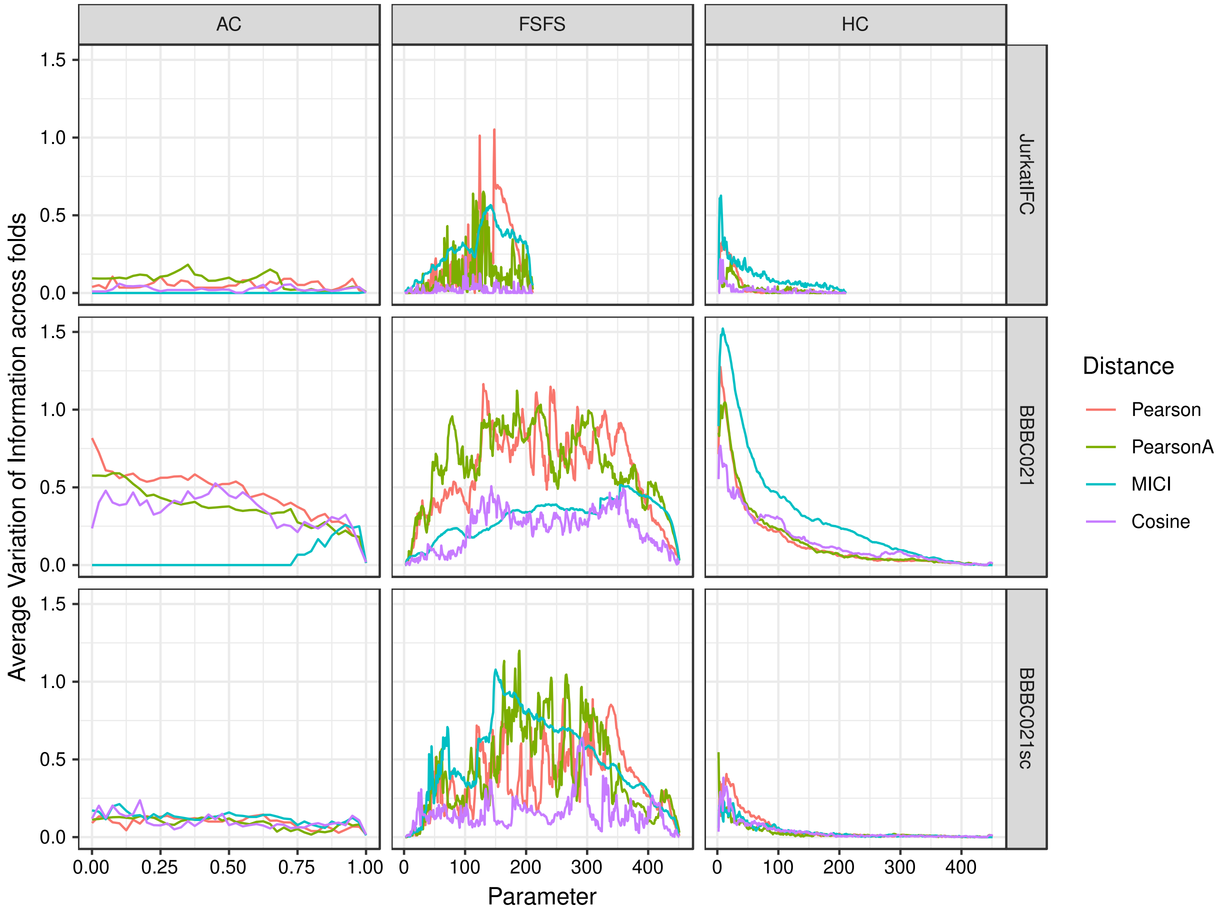


Fig. 6. Variation of information across cross-validation folds.

However, different measures behave differently with respect to the number of feature clusters.

The average silhouette index (Fig. 7) reaches its peak when a medium number of feature clusters are built. When the number of clusters is very high the SI converges to zero, reflecting that most clusters contain a single feature. FSFS produces lower SI values for all datasets; this is due to its tendency to build few large clusters along with many single-feature clusters. It is noteworthy that in some cases this produces negative values of the SI, indicating that some features do not belong to the cluster that is at the smallest average distance from them. A similar behavior is obtained when MICI is used, due to the dependency of this distance on the variance of the features, which causes features with low variance to lie at a small distance to all others, while features with high variance are far from all others.

The average intra-cluster representation entropy (Fig. 8), which measures the redundancy of the features within the same cluster, decreases as the number of clusters increases. The REFS shows a very similar behavior: it increases along with the number of clusters, indicating that the amount of information present in the set of selected features increases. However, it is noteworthy that this measure reaches a plateau or peak for low to average number of clusters, which reflects that, beyond that point, selected features only add redundancy to the dataset.

The redundancy rate (Fig. 9) turns out not to be very illustrative for the purpose of this study, because it maintains an almost constant value around zero. This might be due to the large amount of features in our datasets: even though there might be a high redundancy within different groups of features, the overall average redundancy is close to zero.

Finally, the class separability (Fig. 10) shows a more varied behavior. For a large number of clusters, the CS tends to stabilize; however, when there are few large feature clusters, it is possible to reach a better CS due to the elimination of highly redundant or noisy features. Note that CS values are much larger for BBBC021 than for the other datasets, which is an indication of their higher complexity.

When these performance measures are examined jointly, we can see that the optimal number of clusters is different in each case. Low numbers of clusters produce good median RE and CS, intermediate numbers produce the best SI and REFS. Therefore, although it is not possible to select a single optimal number of clusters, it appears that medium to heavy dimensionality reduction is able to simplify the structure of the data both from an unsupervised and a supervised point of view, which is a very desirable property in terms of interpretability of the data.

When comparing the performance obtained with different distance measures between the features, the first observation that can be made is the more erratic behavior of MICI. There are no big differences among the behavior of the other measures; however, in most cases the cosine distance produces results slightly less promising than those based on Pearson's correlation.

5.2. Classification

The analysis realized in the previous section highlighted SI and REFS are the most adequate criteria to fix a number of feature clusters, because they showed a stable behavior that peaked at a non-trivial number of clusters. In this section, we use these two criteria to reduce the dimensionality of the datasets, and we apply several classifiers to evaluate the performance yielded by

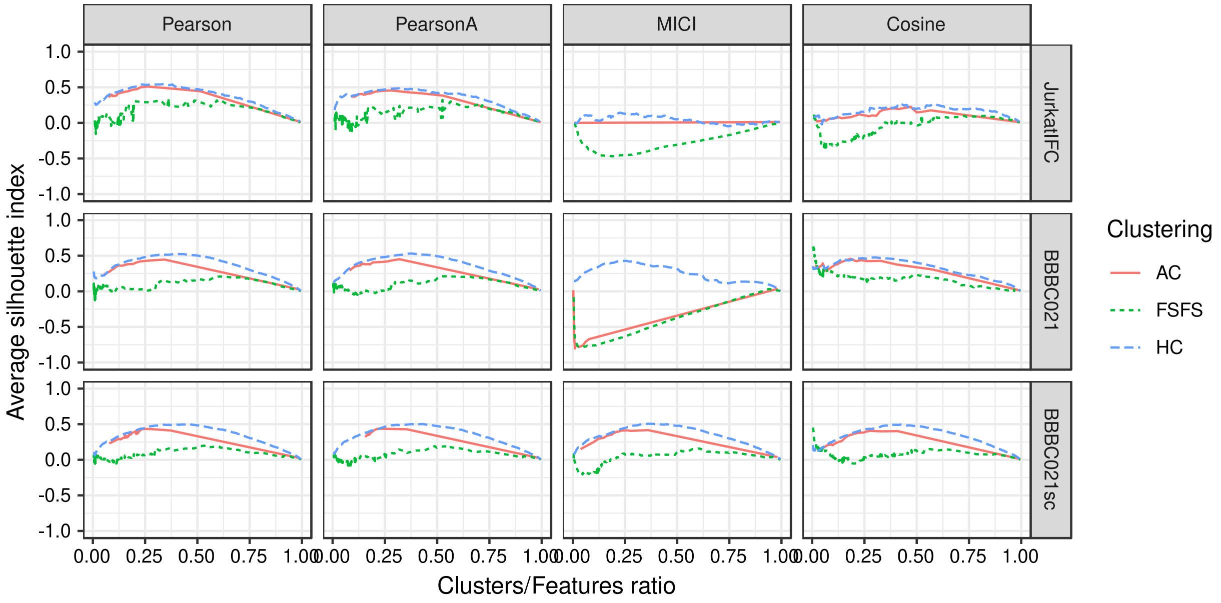


Fig. 7. Average silhouette index of the feature clusters.

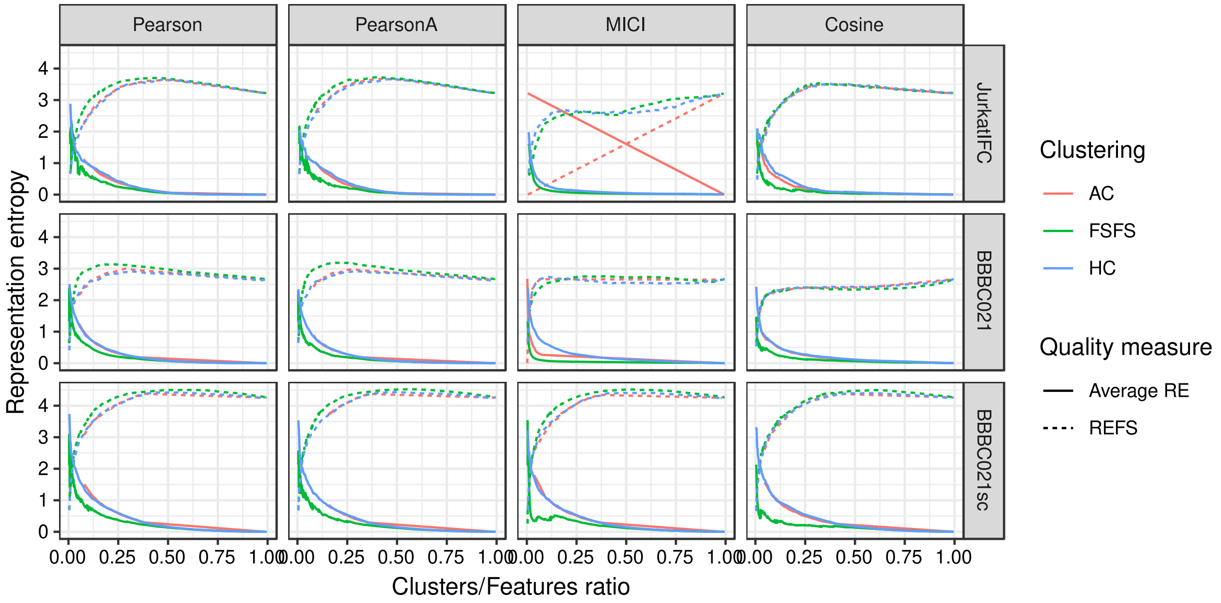


Fig. 8. Representation entropy measures.

the different studied combinations for dimensionality reduction. CS is also used as a selection criterion, in order to complete the analysis with a supervised measure. The balanced accuracy (that is, the average accuracy across all classes) is used as supervised performance metric for all datasets because of its compatibility with multi-class scenarios and its robustness to class imbalance.

The following classifiers are applied on all datasets:

- k-nearest neighbors (1NN) [55], with $k = 1$ and cosine distance, following the procedure applied in [9].
- Random Forest (RF) [56], with 1000 trees.
- Support Vector Machines (SVM) [57]: we use a linear kernel, which according to [58] performs better and faster than the well-known RBF kernel when working on high dimensional data.

In this section, the performance of the different tested dimensionality reduction schemes is evaluated on the three considered datasets: JurkatIFC (Section 5.2.1), BBBC021 (Section 5.2.2) and BBBC021sc (Section 5.2.3). The full tables displaying the balanced accuracy for all experiments are shown in Appendix.

5.2.1. *JurkatIFC: cell cycle classification from imaging flow cytometry data*

As explained in Section 4.2.1, the classes of this dataset are not balanced. When not properly handled, this imbalance can hinder the performance of machine learning algorithms that are applied on the data. In this study, we applied three different techniques to deal with the class imbalance:

- Random Undersampling (RUS) [49] consists on randomly removing instances of the majority classes.
- Random sampling with replacement produces a dataset of the same size as the input, with a uniform class distribution.

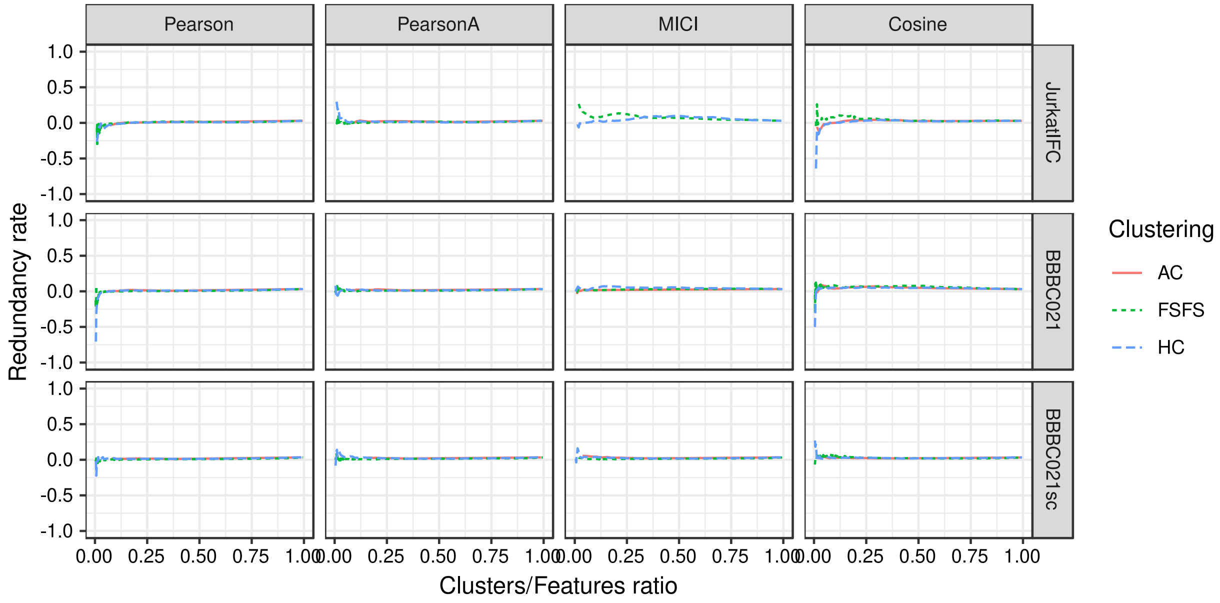


Fig. 9. Redundancy rate after feature selection.

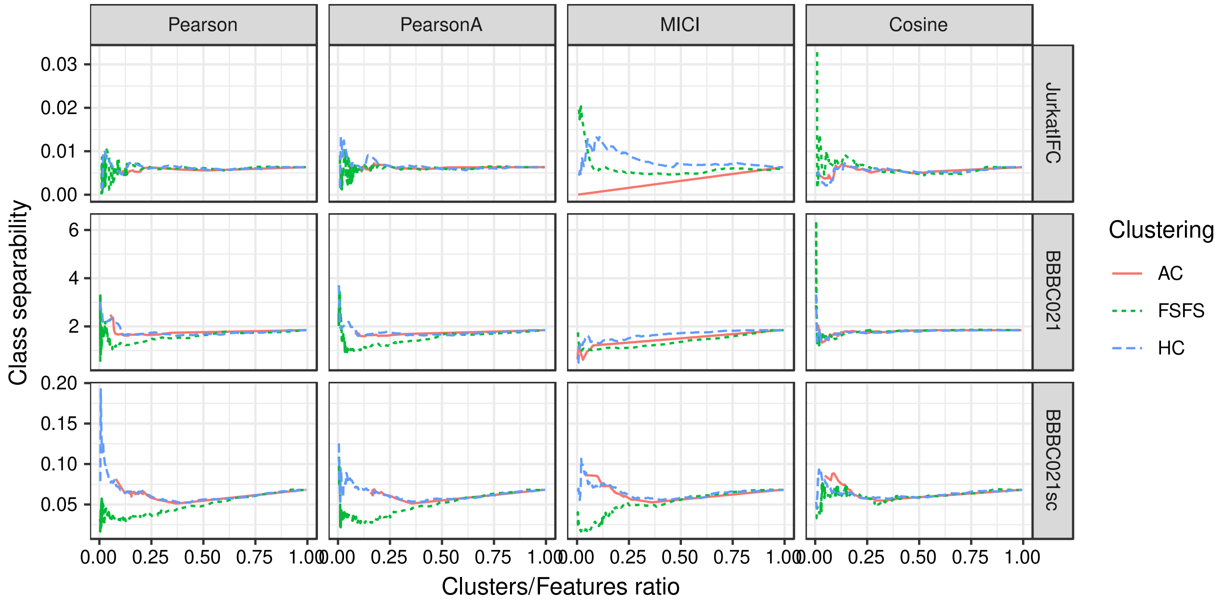


Fig. 10. Class separability after feature selection.

- RUSBoost [59] combines RUS with a boosting algorithm to efficiently train an ensemble of classifiers on multiple samples of the original dataset.

The first two techniques are applied as a preprocessing step prior to training of the classifiers, whereas RUSBoost encapsulates the preprocessing and the classifier training as a single process.

Fig. 11 shows the average accuracy obtained with all the combinations of feature clustering, inter-feature dissimilarity, quality measure, preprocessing and classifier. It can be clearly seen that the best results were obtained when using dissimilarities based on Pearson's correlation, both of which yielded an improvement for all classifiers. Moreover, when REFS was used as a criterion to fix the number of feature clusters, improvements were observed for all combinations of classifiers and distances, assessing the robustness of this approach. It is especially noteworthy that REFS, an unsupervised criterion, was able to yield results as good or even better than the supervised CS.

Table 2 shows the balanced accuracy obtained, restricted to those combinations involving PearsonA and REFS (the tables containing all results are provided in Appendix). The difference between the tested dimensionality reduction procedures is, in general, small, but there is some indication that AC performs better than the other tested clusterings.

Fig. 12 depicts the same accuracies shown in Fig. 11, this time grouped by dimensionality reduction method and classifier, as a function of the number of feature clusters that was judged as best by the different unsupervised quality criterion employed. The plots show how low number of features lead to a low accuracy, but as soon as the number of feature clusters reaches a certain size, the accuracy becomes equal or better than that obtained without any dimensionality reduction (which is depicted as dashed lines in the figure). This highlights why CS, which favors few large clusters, obtains a lower accuracy than SI and REFS. On the contrary, REFS tends to select clusterings

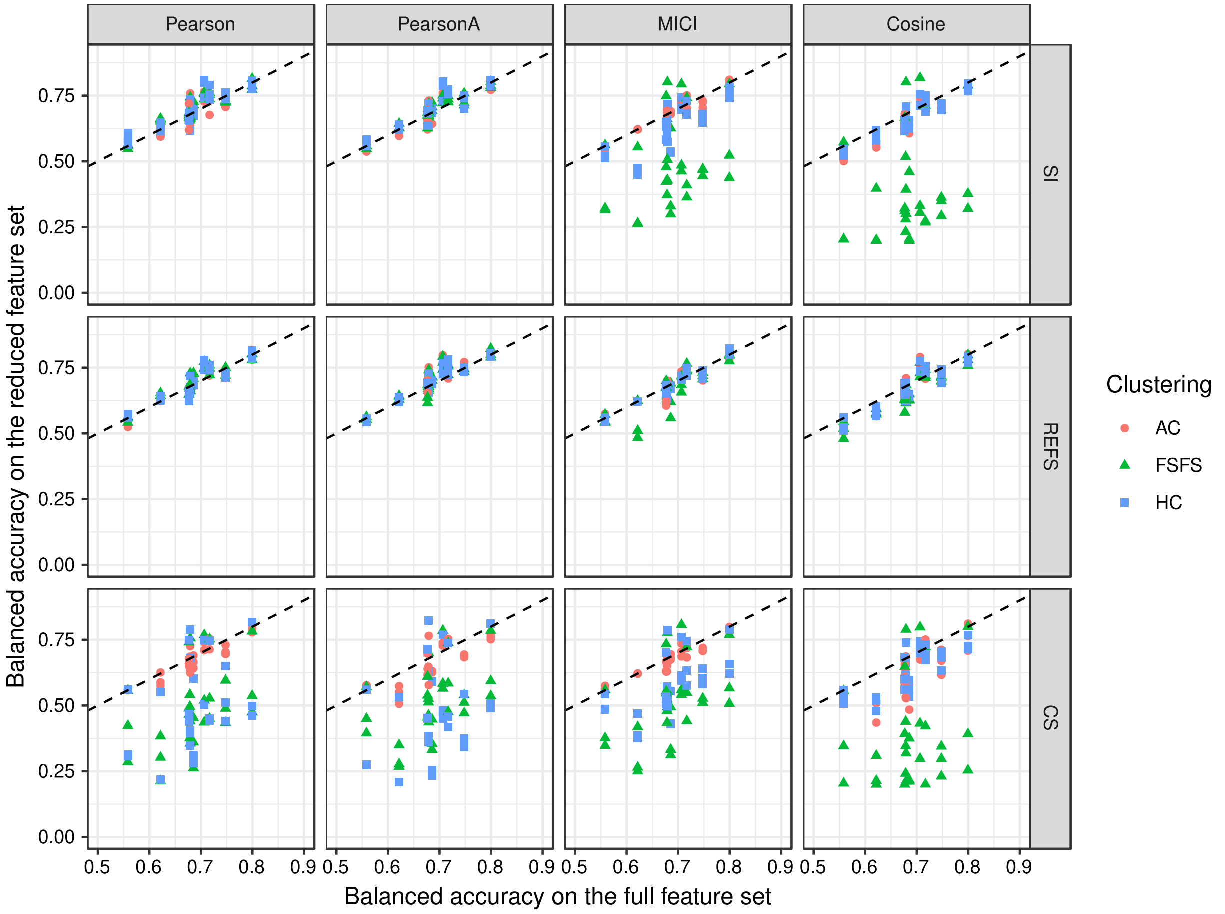


Fig. 11. Balanced accuracy on JurkatIFC. The dashed line depicts the identity function.

Table 2
Balanced accuracy for all classifiers, clustering algorithms and dimensionality reduction methods when using PearsonA dissimilarity and REFS as selection criterion. The highest accuracy in each line is bold-stressed.

Classifier	Preprocessing	All features	PCA (99% var.)	Affinity clustering			FSFS			Hierarchical clustering		
				FS	FW	PCA	FS	FW	PCA	FS	FW	PCA
1NN	None	0.62	0.62	**0.64**	**0.64**	**0.64**	**0.64**	**0.64**	0.63	**0.64**	0.63	0.62
1NN	RESAMPLE	0.69	0.68	0.70	0.70	0.69	0.71	**0.72**	0.70	0.70	0.69	0.69
1NN	RUS	0.68	0.66	0.69	0.68	0.69	0.68	0.66	0.67	**0.70**	0.66	0.66
RUSBoost	None	0.80	0.75	0.80	0.81	0.81	0.79	**0.82**	0.80	0.79	0.81	0.79
RandomForest	RESAMPLE	**0.56**	0.51	0.55	0.55	0.55	0.55	0.55	**0.56**	0.55	0.54	**0.56**
RandomForest	RUS	0.72	0.74	0.76	0.71	0.75	0.76	0.72	0.76	**0.78**	0.72	0.75
SVM-Linear	RESAMPLE	0.71	0.71	**0.80**	0.73	0.77	0.77	0.74	0.79	0.73	0.74	0.77
SVM-Linear	RUS	0.68	0.64	0.66	**0.70**	**0.70**	0.68	0.64	0.62	0.69	0.69	0.69
SVM-RBF	RESAMPLE	0.68	0.49	0.65	**0.75**	0.66	0.68	0.74	0.67	0.67	0.73	0.66
SVM-RBF	RUS	0.75	0.67	0.74	**0.77**	**0.77**	0.74	0.75	0.73	0.75	0.73	0.74

with an intermediate number of clusters, leading to a reasonable trade-off between accuracy and data reduction.

5.2.2. BBBC021: mechanism of action classification in high-throughput microscopy

It has already been stated in [29] that the results obtained on this dataset are better when linear combinations of the features are used instead of a subset of the original ones. Still, it is interesting to evaluate the behavior of our approach on this dataset, which has been used as a benchmark by several authors since its publication.

Fig. 13 shows the balanced accuracy obtained with all tested combinations of algorithms and measures, as a function of the number of clusters. There is a clear drop in the accuracy with respect to the results obtained for JurkatIFC, because BBBC021 is a complex multi-class problem, and after the aggregation of all the cells within each sample the final size of the dataset is of 103 instances. This reduces the stability of most machine learning algorithms, which become more prone to over-fitting. Consistent classifiers (such as k-NN and SVM [60]) produce a better accuracy as the number of training instances increases. It also becomes clear that, for this dataset, a heavy dimensionality reduction yielded a low accuracy. This is particularly the case for CS, which favors few feature clusters. However, most of the combinations that kept approximately one third of the features produced an accuracy as good as that of the original dataset and above that of PCA. Moreover, it can be seen that feature selection and per-group PCA produced very similar results.

The results of combining PearsonA and REFS (Table 3) also show less improvements from the dimensionality reduction with respect to the original set of features. Some of the classifiers show a behavior in accordance with the previous study in [29]

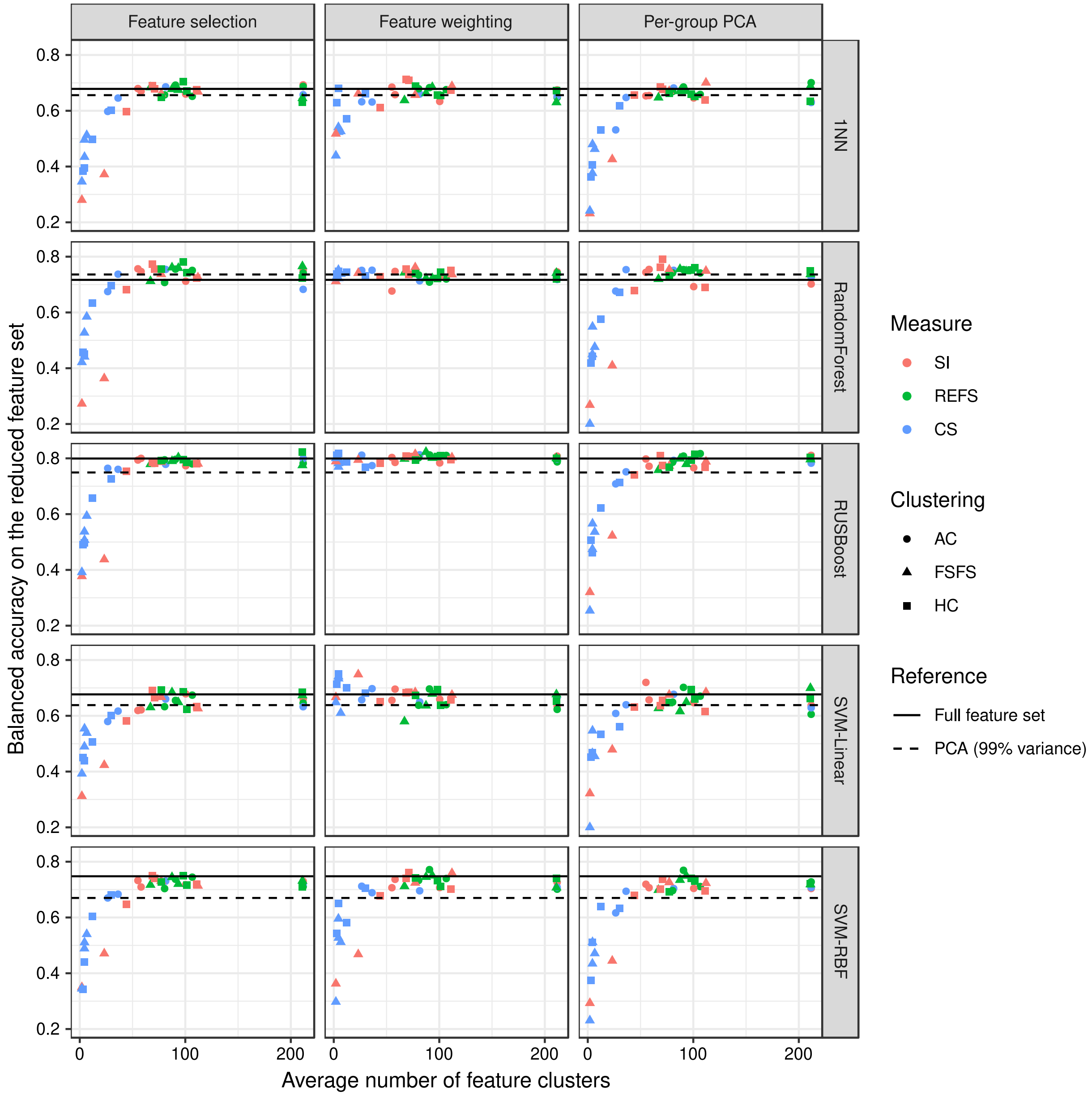


Fig. 12. Balanced accuracy on JurkatIFC after RUS, as a function of the number of feature clusters.

(which only used 1NN), in which PCA obtains better results than using the original features. However, the proposed dimensionality reduction can reach close results while reducing the total number of features.

5.2.3. BBBC021sc: mechanism of action classification in high-throughput microscopy at single-cell resolution

In order to delve deeper into the behavior of the different dimensionality reduction approaches, we have repeated the study on the BBBC021 dataset without aggregating the cells of each well. Therefore, in this case the goal is to classify each individual cell into the mechanism of action of the compound that has been applied to it. Obviously, this problem is much more difficult than when the cells are aggregated in wells. Many individual cells might not have been affected enough by the compounds so as to show any phenotypical changes, and the variability between cells – which disappears when the cells are aggregated – makes the classification even more difficult. Moreover, the three replicates of each sample which were median-aggregated in the standard BBBC021 are used separately in this case.

To the best of our knowledge, at the time of writing this manuscript there is only one published work that makes use of this dataset at the single-cell level [61]. The purpose of the authors was to evaluate transfer learning; therefore, they split the problem into two disjoint groups of treatments, therefore obtaining two separate 6-class problems. As a result, results presented in [61] are not comparable to our analysis, which consider the whole 12-class problem.

Fig. 14 depicts the balanced accuracy obtained on the BBBC021sc dataset with all the tested combinations. The values were much lower than those obtained with the aggregated dataset, which was obviously expected due to the much higher difficulty of classifying single cells. However, it can also be seen in the plots that some of the dimensionality reduction alternatives were able to increase the accuracy obtained, in a similar way as it was observed for JurkatIFC, instead of just maintaining the same accuracy values (as was generally the case for BBBC021). In particular, the performance of the SVM classifier was the most improved one, and both SI and REFS appeared to be robust criteria

Table 3
Balanced accuracy for all classifiers, clustering algorithms and dimensionality reduction methods when using PearsonA dissimilarity and REFS as selection criterion. The highest accuracy in each line is bold-stressed.

Classifier	All features	PCA (99% var.)	Affinity clustering			FSFS			Hierarchical clustering		
			FS	FW	PCA	FS	FW	PCA	FS	FW	PCA
1NN	0.82	**0.84**	0.76	0.66	0.74	0.68	0.62	0.65	0.76	0.72	0.73
RandomForest	0.81	0.70	0.77	0.81	0.80	0.77	0.80	0.74	**0.82**	0.81	0.80
SVM-Linear	**0.88**	0.86	0.83	0.82	0.87	0.83	0.76	0.77	0.86	0.81	0.86
SVM-RBF	0.72	**0.86**	0.69	0.03	0.64	0.52	0.01	0.44	0.70	0.02	0.67

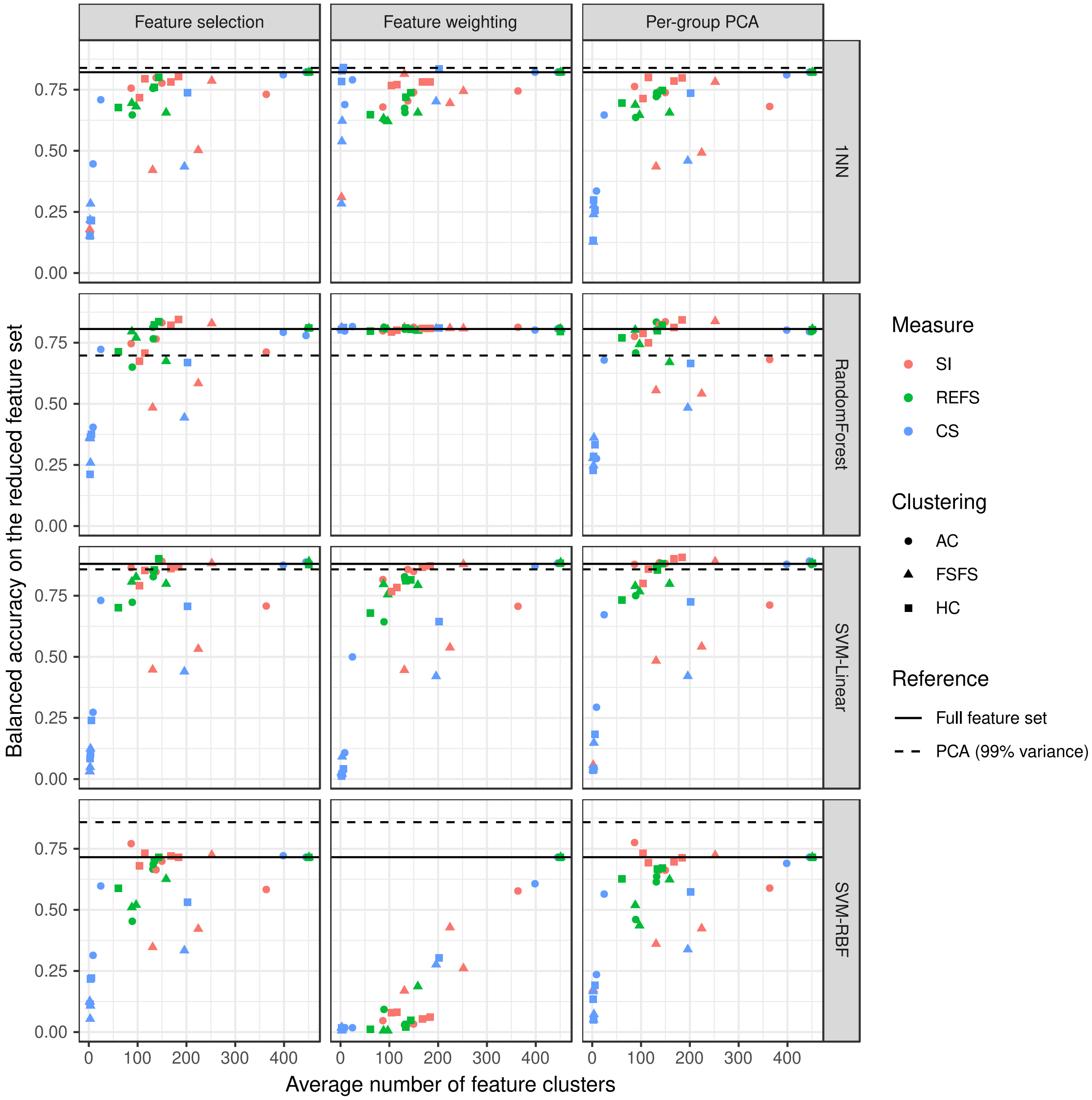


Fig. 13. Balanced accuracy on BBBC021 as a function of the number of feature clusters.

to select the feature clusters, as well as MICI and both Pearson's dissimilarities.

Complementarily, Table 4 show the results when using PearsonA and REFS. In accordance to the patterns shown in the figure, the dimensionality reduction reaches accuracies similar or higher to those using the original set of features.

It can also be seen that the peak accuracy was reached with about half the features with SVM, which enabled an improvement of the results obtained without any dimensionality reduction. It is also noteworthy that SVM reached a higher accuracy than both RandomForest and *k*-NN in this problem; therefore, the results obtained after the dimensionality reduction were the best of this use case. Both *k*-NN and RandomForest reach similar accuracy to the baseline when using about 25% of the features.

It is also observed, as for the previous datasets, that CS favored the creation of few large clusters, which deteriorated the final classification performance due to a too heavy dimensionality reduction. Therefore, REFS and SI yielded more acceptable results.

Table 4
Balanced accuracy for all classifiers, clustering algorithms and dimensionality reduction methods when using PearsonA dissimilarity and REFS as selection criterion. The highest accuracy in each line is bold-stressed.

Classifier	All features	PCA (99% var.)	Affinity clustering			FSFS			Hierarchical clustering		
			FS	FW	PCA	FS	FW	PCA	FS	FW	PCA
1NN	**0.26**	**0.26**	0.22	0.17	0.25	0.23	0.21	**0.26**	0.24	0.22	**0.26**
RandomForest	**0.20**	0.07	0.17	**0.20**	0.17	0.18	**0.20**	0.17	0.18	**0.20**	0.19
SVM-Linear	0.35	0.33	0.33	0.39	0.32	0.32	0.38	**0.40**	0.33	0.36	0.31

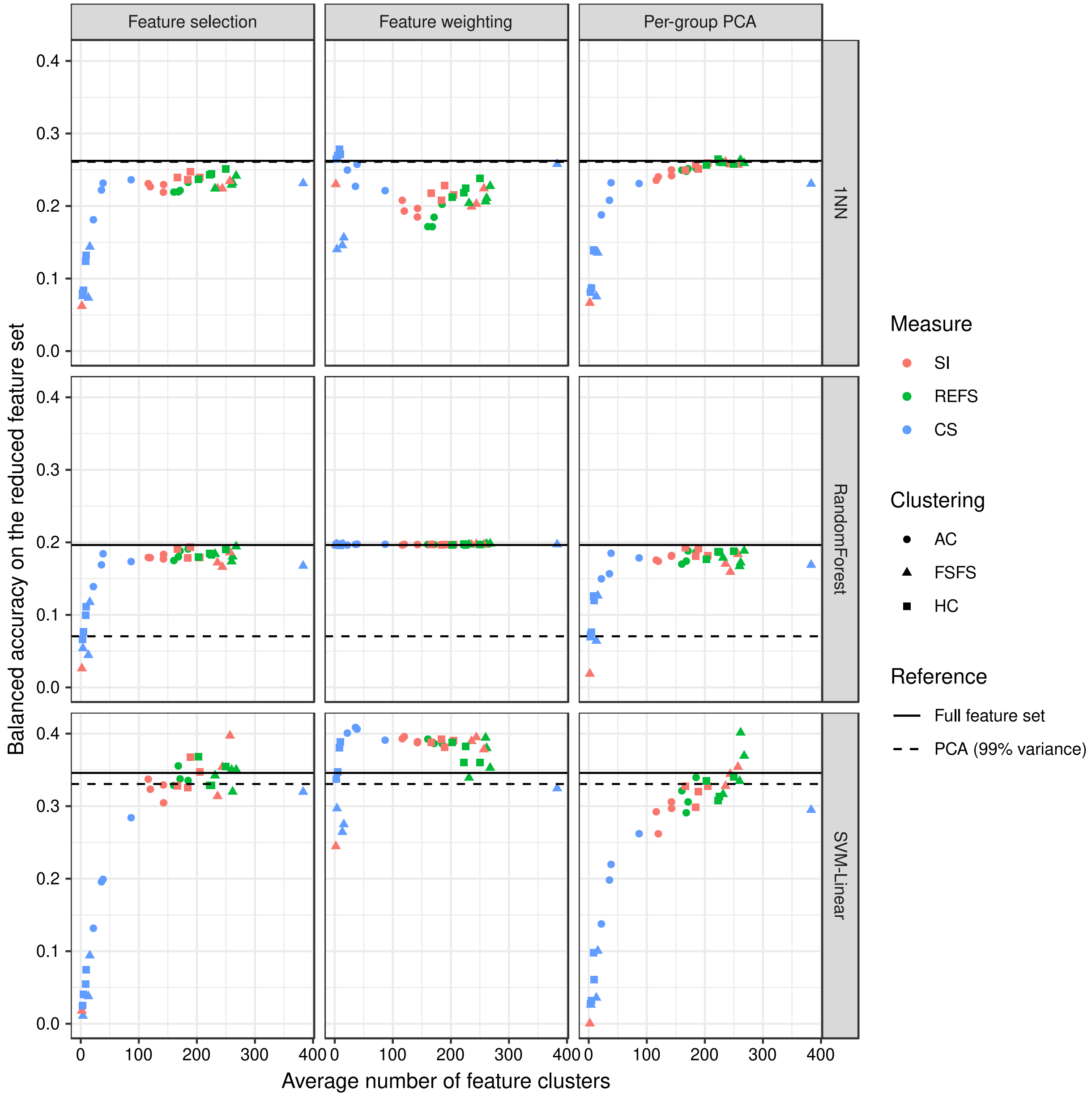


Fig. 14. Balanced accuracy on BBBC021sc as a function of the number of feature clusters. Horizontal lines show the accuracy of the classifiers without any dimensionality reduction.

5.3. Analysis with statistical tests

In order to provide a meaningful and robust evaluation of all the tested methods, it is necessary to go beyond the raw performance measures by carrying out statistical tests that allow us to determine which alternatives are consistently more suited to the task than others. The balanced accuracy is used throughout this section as the reference performance measure for the classification, because it can be used across datasets with different class imbalance ratios. Results with a p-value lower than 0.01 are bold-stressed in the tables throughout this section.

Tables 5–8 show the results of the pairwise Wilcoxon test when comparing the balanced accuracy obtained by different clustering algorithms, dissimilarities, clustering quality criteria

Table 5
Results of the Wilcoxon test for different feature clustering algorithms (Bonferroni-corrected p-values between parentheses).

	AC	FSFS
HC	**−0.0094** (6.89e−04)	**0.0278** (7.84e−18)
AC		**0.0890** (1.6e−33)

Table 6
Results of the Wilcoxon test for the dissimilarity measures (Bonferroni-corrected p-values between parentheses).

	MICI	Pearson	PearsonA
Cosine	**−0.0187** (8.79e−05)	**−0.028** (1.04e−13)	**−0.0243** (1.47e−11)
MICI		−0.0045 (> 1)	−0.0079 (0.105)
Pearson			0.0022 (> 1)

Table 7
Results of the Wilcoxon test for the clustering quality criteria (Bonferroni-corrected p-values between parentheses).

	REFS	SI
CS	**−0.1297** (3.65e−58)	**−0.0756** (1.08e−43)
REFS		**0.0076** (7.4e−05)

Table 8
Results of the Wilcoxon test for the feature selection algorithm (Bonferroni-corrected p-values between parentheses).

	Feature weighting	Per-cluster PCA
Feature selection	**−0.0229** (3.34e−13)	**0.004** (0.00758)
Feature weighting		**0.0273** (7.11e−15)

and dimensionality reduction procedures, respectively. This non-parametric test [62] determines whether two samples come from different distributions, where the null hypothesis is that the medians of the distributions of the samples are the same. A Bonferroni correction is also applied to the results of the test.

It can be seen that in many cases, statistically significant differences were observed between different alternatives. AC obtained slightly but consistently better results than HC, while FSFS was the worst ranked of the 3 tested algorithms. There were also some differences between dissimilarity measures: both measures based on Pearson's correlation were statistically similar, and were ranked above both MICI, which in turn performed better than the cosine distance. REFS was the best option as a criterion to select the best feature clusters, closely followed by SI. Interestingly, CS was ranked as the worst option despite being supervised; this highlights that the dataset manifold is more complex than the linearly separable structures that are assumed by this measure, and also the adequacy of the unsupervised criteria used in this paper. Finally, the feature weighting procedure yielded a better balanced accuracy than the feature selection and group-wise PCA thanks to the advantage of performing well even for a low number of feature clusters, as no features are removed from the dataset. Interestingly, feature selection consistently obtained slightly better results than per-group PCA, with a low *p*-value. This allows us to highlight feature selection as the preferred method, due to its better interpretability and lower computational requirements.

It is also necessary to compare the methods with the performance obtained on the original dataset, when no dimensionality reduction is applied. For this purpose, we applied Friedman's test [63] with Holm's correction to compare the results of the different combinations of clustering algorithm, dissimilarity, quality criterion and dimensionality reduction, when applied on the tested datasets using each classifier. The results that yielded a certain statistical significance ($p\text{-value} < 0.05$) are shown in Table 9, where it can be seen that all combinations actually damaged the performance with respect to using all features. In particular, all such combinations include either CS as selection criterion, or FSFS as clustering algorithm. Therefore, it can be inferred that for all combinations that do not include these elements, no significant differences have been found with respect to using the original dataset, despite the gain of time obtained after the dimensionality reduction.

6. Conclusions and future work

In this paper, we established a framework to evaluate multiple approaches for a robust unsupervised dimensionality reduction of large-scale datasets, focusing on the interpretability for biological single-cell imaging data. The elements conforming the dimensionality reduction were identified, including inter-feature dissimilarity measures, clustering algorithms, unsupervised quality metrics to evaluate the obtained clusters, and different dimensionality reduction procedures such as feature selection, feature weighting, and PCA. For each of these categories, the main strategies in the scientific literature were implemented and tested to evaluate their behavior in a general case.

The experiments carried out involved two recent imaging datasets of different types (namely imaging flow cytometry and

Table 9
Estimate of Friedman's test comparing each combination to the same dataset and classifier without dimensionality reduction.

Clustering	Distance	Quality measure	Dimensionality reduction	*p*-value	Summary statistic
FSFS	Cosine	SI	Per-cluster PCA	4.69e−09	0.1997
FSFS	Cosine	SI	Feature selection	6.68e−09	0.2175
FSFS	Cosine	CS	Per-cluster PCA	1.57e−08	0.1906
HC	PearsonA	CS	Per-cluster PCA	3.95e−08	0.2655
FSFS	Cosine	CS	Feature selection	4.73e−08	0.2611
FSFS	Pearson	CS	Per-cluster PCA	7.46e−08	0.2820
HC	PearsonA	CS	Feature selection	1.15e−07	0.2838
HC	Pearson	CS	Feature selection	1.54e−07	0.3005
HC	Pearson	CS	Per-cluster PCA	2.52e−07	0.3041
FSFS	PearsonA	CS	Per-cluster PCA	2.58e−07	0.3074
FSFS	Pearson	CS	Feature selection	4.78e−07	0.3236
FSFS	PearsonA	CS	Feature selection	5.94e−07	0.3327
HC	Cosine	CS	Feature selection	3.12e−05	0.4143
FSFS	MICI	CS	Feature selection	3.89e−05	0.4005
HC	Cosine	CS	Per-cluster PCA	4.07e−05	0.4080
FSFS	MICI	CS	Per-cluster PCA	5.34e−05	0.4172
HC	MICI	CS	Feature selection	1.21e−04	0.4869
HC	MICI	CS	Per-cluster PCA	1.57e−04	0.4899
FSFS	MICI	SI	Per-cluster PCA	3.46e−04	0.4040
FSFS	MICI	SI	Feature selection	3.51e−04	0.4005
AC	Cosine	CS	Per-cluster PCA	3.65e−04	0.4439
AC	Cosine	CS	Feature selection	3.78e−03	0.4858

Table A.10
Balanced accuracy obtained on JurkatIFC (Affinity clustering).

Distance	Measure	Dim. red.	No preprocessing		RESAMPLE				RUS			
			1NN	RUSBoost	1NN	RF	SVM-Lin.	SVM-RBF	1NN	RF	SVM-Lin.	SVM-RBF
Pearson	SI	FS	0.5943	0.7939	0.6864	0.5763	0.7645	0.6383	0.6796	0.7565	0.6191	0.7326
		FW	0.5945	0.8031	0.6791	0.5593	0.7270	0.7586	0.6848	0.6767	0.6555	0.7067
		PCA	0.6325	0.7977	0.6890	0.5869	0.7622	0.6238	0.6534	0.7437	0.7197	0.7192
	REFS	FS	0.6281	0.7801	0.6793	0.5380	0.7493	0.6752	0.6518	0.7509	0.6741	0.7446
		FW	0.6339	0.8104	0.6893	0.5601	0.7413	0.7178	0.6758	0.7200	0.6397	0.7396
		PCA	0.6232	0.8170	0.6898	0.5244	0.7492	0.6462	0.6586	0.7414	0.6704	0.7109
	CS	FS	0.5883	0.7785	0.6648	0.5588	0.7476	0.6333	0.6857	0.7543	0.6601	0.7306
		FW	0.5757	0.8127	0.6422	0.5582	0.7402	0.7254	0.6600	0.7136	0.6500	0.6957
		PCA	0.6257	0.7928	0.6904	0.5617	0.7107	0.6245	0.6819	0.7511	0.6771	0.7040
PearsonA	SI	FS	0.6268	0.8000	0.6975	0.5629	0.7492	0.6413	0.6701	0.7467	0.6212	0.7093
		FW	0.5969	0.7854	0.6425	0.5582	0.7607	0.7311	0.6578	0.7471	0.6956	0.7367
		PCA	0.6245	0.7717	0.6955	0.5371	0.7237	0.6342	0.6546	0.7543	0.6571	0.7068
	REFS	FS	0.6370	0.7964	0.7017	0.5498	0.7976	0.6506	0.6920	0.7553	0.6552	0.7371
		FW	0.6407	0.8123	0.6994	0.5469	0.7340	0.7514	0.6829	0.7088	0.6962	0.7716
		PCA	0.6370	0.8082	0.6857	0.5467	0.7700	0.6642	0.6853	0.7536	0.7020	0.7692
	CS	FS	0.5739	0.7606	0.6314	0.5751	0.7390	0.6218	0.6456	0.7372	0.6169	0.6835
		FW	0.5065	0.7741	0.5915	0.5614	0.7339	0.7652	0.6313	0.7515	0.6973	0.6889
		PCA	0.5509	0.7517	0.6258	0.5774	0.7274	0.5777	0.6475	0.7534	0.6396	0.6940
MICI	SI	FS	0.6212	0.7995	0.6910	0.5421	0.7213	0.6753	0.6929	0.7509	0.6595	0.7301
		FW	0.6216	0.8080	0.6790	0.5514	0.7289	0.6830	0.6750	0.7455	0.6355	0.7260
		PCA	0.6216	0.8107	0.6831	0.5579	0.7292	0.6702	0.6328	0.7020	0.6374	0.7040
	REFS	FS	0.6212	0.7821	0.6736	0.5743	0.7360	0.6906	0.6875	0.7412	0.6457	0.7108
		FW	0.6216	0.7873	0.6827	0.5738	0.7052	0.6812	0.6477	0.7416	0.6230	0.7014
		PCA	0.6216	0.7894	0.6791	0.5588	0.7015	0.6658	0.7008	0.7282	0.6054	0.7277
	CS	FS	0.6212	0.7989	0.6807	0.5583	0.6865	0.6799	0.6575	0.6828	0.6325	0.7196
		FW	0.6216	0.7982	0.6747	0.5750	0.7052	0.6823	0.6524	0.7182	0.6666	0.7082
		PCA	0.6216	0.7827	0.6948	0.5582	0.7353	0.6648	0.6297	0.7272	0.6288	0.7074
Cosine	SI	FS	0.5756	0.7735	0.6579	0.5079	0.7512	0.6476	0.6599	0.7120	0.6788	0.7174
		FW	0.5526	0.7836	0.6066	0.5481	0.7144	0.6598	0.6335	0.7265	0.6595	0.7073
		PCA	0.5652	0.7663	0.6368	0.5008	0.7398	0.6366	0.6459	0.6925	0.6472	0.7042
	REFS	FS	0.5834	0.7947	0.6403	0.5056	0.7908	0.6233	0.6570	0.7074	0.6327	0.7037
		FW	0.5649	0.8020	0.6302	0.5587	0.7571	0.7102	0.6790	0.7347	0.6377	0.7329
		PCA	0.5827	0.7860	0.6384	0.5184	0.7472	0.6332	0.6688	0.7418	0.6489	0.6963
	CS	FS	0.5092	0.7644	0.5477	0.5509	0.7178	0.5936	0.5976	0.6749	0.5795	0.6700
		FW	0.5191	0.8115	0.5808	0.5454	0.7302	0.6865	0.6322	0.7510	0.6573	0.7120
		PCA	0.4350	0.7085	0.4842	0.5055	0.6747	0.5275	0.5315	0.6766	0.6084	0.6166

Table A.11
Balanced accuracy obtained on JurkatIFC (FSFS).

Distance	Measure	Dim. red.	No preprocessing		RESAMPLE				RUS			
			1NN	RUSBoost	1NN	RF	SVM-Lin.	SVM-RBF	1NN	RF	SVM-Lin.	SVM-RBF
Pearson	SI	FS	0.6630	0.7864	0.7272	0.5481	0.7621	0.6635	0.6613	0.7371	0.6670	0.7304
		FW	0.6589	0.8150	0.7148	0.5594	0.7561	0.7436	0.6564	0.7615	0.6844	0.7244
		PCA	0.6526	0.7726	0.7148	0.5630	0.7371	0.6662	0.6704	0.7538	0.6753	0.7261
	REFS	FS	0.6430	0.8033	0.7295	0.5585	0.7484	0.6607	0.6746	0.7592	0.6486	0.7202
		FW	0.6538	0.8015	0.7272	0.5626	0.7450	0.7291	0.6847	0.7223	0.6787	0.7502
		PCA	0.6424	0.7793	0.7078	0.5416	0.7737	0.6630	0.6707	0.7473	0.6486	0.7494
	CS	FS	0.3026	0.5372	0.3605	0.4233	0.5197	0.4402	0.4958	0.5274	0.4896	0.4888
		FW	0.3834	0.7825	0.4533	0.5564	0.7682	0.7561	0.5412	0.7524	0.7411	0.5956
		PCA	0.2128	0.4745	0.2620	0.2852	0.4362	0.3563	0.3765	0.4502	0.4660	0.4345
PearsonA	SI	FS	0.6262	0.7805	0.7063	0.5476	0.7512	0.6746	0.6682	0.7254	0.6263	0.7137
		FW	0.6434	0.8031	0.6924	0.5578	0.7227	0.7190	0.6881	0.7362	0.6750	0.7590
		PCA	0.6273	0.7882	0.7237	0.5604	0.7401	0.6679	0.7007	0.7489	0.6846	0.7232
	REFS	FS	0.6387	0.7901	0.7132	0.5531	0.7701	0.6777	0.6776	0.7611	0.6842	0.7445
		FW	0.6431	0.8230	0.7156	0.5520	0.7434	0.7407	0.6630	0.7225	0.6365	0.7452
		PCA	0.6284	0.7991	0.7013	0.5628	0.7924	0.6688	0.6702	0.7554	0.6160	0.7345
	CS	FS	0.2679	0.5936	0.3323	0.4502	0.5651	0.4367	0.5126	0.5847	0.5385	0.5403
		FW	0.3497	0.7847	0.4475	0.5708	0.7853	0.6993	0.5265	0.7378	0.6103	0.5117
		PCA	0.2769	0.5363	0.3545	0.3950	0.5836	0.4499	0.4628	0.4760	0.4550	0.4714
MICI	SI	FS	0.2621	0.4376	0.3295	0.3217	0.4626	0.5069	0.3721	0.3637	0.4234	0.4709
		FW	0.5541	0.7945	0.6263	0.5618	0.7938	0.8019	0.6614	0.7412	0.7484	0.4680
		PCA	0.2649	0.5226	0.2992	0.3154	0.4850	0.4288	0.4260	0.4096	0.4788	0.4447
	REFS	FS	0.4840	0.7752	0.5587	0.5418	0.6564	0.6229	0.6451	0.7656	0.6732	0.7314

(continued on next page)

Table A.11 (*continued*).

Distance	Measure	Dim. red.	No preprocessing		RESAMPLE				RUS			
			1NN	RUSBoost	1NN	RF	SVM-Lin.	SVM-RBF	1NN	RF	SVM-Lin.	SVM-RBF
		FW	0.5107	0.8004	0.6200	0.5444	0.7176	0.6386	0.6303	0.7415	0.6769	0.7072
		PCA	0.6244	0.7961	0.6908	0.5717	0.6843	0.6887	0.6880	0.7367	0.6995	0.7188
	CS	FS	0.2646	0.5074	0.3117	0.3765	0.5414	0.4945	0.4344	0.4414	0.5542	0.5105
		FW	0.4185	0.7696	0.4940	0.5588	0.8073	0.7764	0.5239	0.7226	0.7336	0.5270
		PCA	0.2503	0.5661	0.3326	0.3477	0.5593	0.5529	0.4799	0.5489	0.5468	0.5116
Cosine	SI	FS	0.1988	0.3774	0.2041	0.2041	0.3053	0.3933	0.2802	0.2728	0.3120	0.3497
		FW	0.3972	0.7890	0.4601	0.5728	0.8179	0.8017	0.5177	0.7117	0.6667	0.3628
		PCA	0.2013	0.3201	0.1985	0.2038	0.3312	0.3013	0.2320	0.2682	0.3217	0.2929
	REFS	FS	0.5942	0.7791	0.6494	0.5188	0.7299	0.6234	0.6840	0.7123	0.6304	0.7169
		FW	0.5780	0.7981	0.6261	0.5439	0.7698	0.6849	0.6376	0.7428	0.5798	0.7119
		PCA	0.5727	0.7579	0.6288	0.4802	0.7147	0.6156	0.6472	0.7198	0.6272	0.6984
	CS	FS	0.2001	0.3918	0.2164	0.3460	0.4322	0.3190	0.3460	0.4216	0.3926	0.3448
		FW	0.3096	0.8003	0.3754	0.5560	0.7980	0.7894	0.4392	0.7227	0.6482	0.2975
		PCA	0.2143	0.2538	0.2115	0.2042	0.2980	0.2119	0.2416	0.2004	0.2001	0.2308

Table A.12
Balanced accuracy obtained on JurkatIFC (Hierarchical clustering).

Distance	Measure	Dim. red.	No preprocessing		RESAMPLE				RUS			
			1NN	RUSBoost	1NN	RF	SVM-Lin.	SVM-RBF	1NN	RF	SVM-Lin.	SVM-RBF
Pearson	SI	FS	0.6158	0.7833	0.6731	0.6068	0.7998	0.6761	0.6783	0.7551	0.6649	0.7410
		FW	0.6468	0.8066	0.6857	0.5648	0.7428	0.7358	0.7099	0.7355	0.6839	0.7609
		PCA	0.6456	0.7750	0.6984	0.5856	0.8093	0.6168	0.6773	0.7902	0.6558	0.7370
	REFS	FS	0.6426	0.7853	0.6982	0.5642	0.7408	0.6771	0.6701	0.7420	0.6227	0.7155
		FW	0.6342	0.8103	0.6832	0.5727	0.7546	0.7180	0.6541	0.7437	0.6380	0.7121
		PCA	0.6253	0.8149	0.6979	0.5598	0.7793	0.6517	0.6508	0.7607	0.6611	0.7302
	CS	FS	0.2174	0.4971	0.2801	0.3140	0.5057	0.3467	0.3954	0.4509	0.4392	0.4410
		FW	0.5503	0.8172	0.6025	0.5600	0.7496	0.7884	0.6801	0.7469	0.7499	0.6507
		PCA	0.2186	0.4623	0.3124	0.3074	0.5034	0.3938	0.4055	0.4432	0.4682	0.5111
PearsonA	SI	FS	0.6207	0.7862	0.6843	0.5755	0.8031	0.6558	0.6901	0.7739	0.6907	0.7503
		FW	0.6408	0.8081	0.7004	0.5591	0.7285	0.7198	0.7128	0.7557	0.6821	0.7408
		PCA	0.6407	0.8098	0.6939	0.5831	0.7882	0.6327	0.6860	0.7616	0.6356	0.7016
	REFS	FS	0.6374	0.7950	0.7017	0.5542	0.7263	0.6738	0.7041	0.7815	0.6862	0.7498
		FW	0.6342	0.8058	0.6899	0.5432	0.7387	0.7278	0.6557	0.7214	0.6944	0.7329
		PCA	0.6201	0.7932	0.6881	0.5561	0.7672	0.6631	0.6598	0.7516	0.6942	0.7404
	CS	FS	0.2085	0.4906	0.2549	0.2750	0.4805	0.3806	0.3838	0.4579	0.4502	0.3419
		FW	0.5330	0.8110	0.5914	0.5598	0.7692	0.8234	0.6293	0.7384	0.7138	0.5434
		PCA	0.2093	0.5072	0.2337	0.2729	0.4507	0.3608	0.3626	0.4187	0.4522	0.3745
MICI	SI	FS	0.4679	0.7533	0.5381	0.5204	0.7020	0.5736	0.5976	0.6812	0.5821	0.6478
		FW	0.4747	0.7826	0.5340	0.5559	0.7432	0.7174	0.6111	0.7287	0.6504	0.6777
		PCA	0.4503	0.7401	0.5359	0.5129	0.6966	0.5727	0.6568	0.6787	0.6316	0.6799
	REFS	FS	0.6215	0.8231	0.6800	0.5611	0.7050	0.6824	0.6307	0.7242	0.6837	0.7091
		FW	0.6212	0.7996	0.6690	0.5466	0.7165	0.6800	0.6720	0.7208	0.6524	0.7396
		PCA	0.6212	0.8026	0.6814	0.5587	0.7182	0.6706	0.6346	0.7496	0.6619	0.7205
	CS	FS	0.3765	0.6578	0.4296	0.4860	0.5954	0.5294	0.4977	0.6336	0.5066	0.6036
		FW	0.4687	0.7881	0.5557	0.5444	0.7598	0.7864	0.5712	0.7450	0.7008	0.5818
		PCA	0.3855	0.6218	0.4314	0.4843	0.6136	0.4915	0.5306	0.5757	0.5342	0.6400
Cosine	SI	FS	0.6182	0.7816	0.6580	0.5216	0.7566	0.6561	0.6763	0.7229	0.6333	0.7198
		FW	0.5799	0.7960	0.6251	0.5468	0.7116	0.7087	0.6734	0.7512	0.6565	0.7018
		PCA	0.5887	0.7678	0.6499	0.5331	0.7142	0.6337	0.6381	0.6897	0.6160	0.6958
	REFS	FS	0.6037	0.7927	0.6484	0.5113	0.7748	0.6317	0.6478	0.7556	0.6923	0.7268
		FW	0.5655	0.7931	0.6395	0.5611	0.7352	0.6943	0.6878	0.7391	0.6725	0.7424
		PCA	0.5837	0.7689	0.6492	0.5230	0.7393	0.6233	0.6626	0.7319	0.6483	0.6920
	CS	FS	0.5194	0.7256	0.5629	0.5078	0.6992	0.6083	0.6025	0.6954	0.6003	0.6815
		FW	0.5305	0.7686	0.5959	0.5597	0.7439	0.7394	0.6614	0.7315	0.6817	0.7055
		PCA	0.4785	0.7142	0.5336	0.5288	0.7344	0.5646	0.6186	0.6726	0.5604	0.6335

high-content screening) that are publicly available and used by other authors in the field. One of those datasets was evaluated both after profiling the samples (as is usually done by other authors) and also at the single-cell level, an experimental setting that had not been tested on this dataset before due to its large scale. All combinations of the previously described component of the pipeline were tested on several well-known classifiers in order to obtain sufficient results for a proper statistical analysis.

The results obtained in the experiments highlighted the higher robustness and accuracy yield of some of the components, such as Affinity Clustering as feature clustering algorithm, the Representation Entropy as a quality measure for feature clusterings, or inter-feature dissimilarities based on Pearson's correlation. It was also shown that, when the right combinations of elements were used, unsupervised dimensionality reduction was able to lower the complexity of the datasets without reducing the accuracy or even improving it in some cases.

A natural future research line following this paper involves the application of these same concepts to deep neural networks. A feature clustering such as proposed in this paper, combined with an adequate unsupervised quality criterion, could potentially enhance both the training process of the network and the posterior feature attribution. This approach could be valuable in the case of highly complex problems, for which neural networks obtain an accuracy much higher than other, more interpretable classifiers. Another line that immediately follows from our work is the development of quality criteria that would be directly embedded in the clustering algorithm. The results in this paper highlighted representation entropy as an especially promising robust criterion to evaluate clusterings; as such, embedding the computation of the RE directly within the clustering procedure could perhaps lead to enhanced feature clusters.

CRediT authorship contribution statement

Daniel Peralta: Methodology, Software, Validation, Writing - original draft. **Yvan Saeys:** Conceptualization, Methodology, Supervision, Writing - review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was supported by research grant IWT150865 Exaptation from the Flanders Innovation and Entrepreneurship agency, Belgium. D. Peralta is a Postdoctoral Fellow of the Research Foundation of Flanders (170303/12X1619N). Y. Saeys is an ISAC Marylou Ingram Scholar.

We would like to acknowledge the valuable contribution of all participants of the Exaptation project, namely Sofie van Gassen and Quentin Rouchon from Ghent University; Jörg Kurt Wegner, Emmanuel Gustin and Hugo Ceulemans from Janssen Pharmaceutica; Roel Wuyts and Charlotte Herzeel from imec; Jaak Simm, Adam Arany and Yves Moreau from KU Leuven, and Günter Klambauer, Maciej Kańduła and Sepp Hochreiter from the Johannes Kepler University.

Appendix. Balanced accuracy results

This section contains the full results used to produce the plots and statistical tests analyzed in Section 5. Table A.10 shows the balanced accuracy obtained on the JurkatIFC dataset when using affinity clustering along with all combinations of dissimilarities, quality measures, dimensionality reduction, preprocessing and classifiers. Tables A.11 and A.12 show similar results for FSFS and hierarchical clustering, respectively. Finally, Tables A.13 and A.14 show the balanced accuracy for BBBC021 and BBBC021_sc.

Table A.13
Balanced accuracy obtained on BBBC021.

Distance	Measure	Dim. red.	AC				FSFS				HC			
			1NN	RF	SVM-Lin.	SVM-RBF	1NN	RF	SVM-Lin.	SVM-RBF	1NN	RF	SVM-Lin.	SVM-RBF
Pearson	SI	FS	0.7770	0.8317	0.8909	0.6991	0.4212	0.4844	0.4476	0.3475	0.8041	0.8460	0.8686	0.7145
		FW	0.7395	0.8132	0.8492	0.0327	0.8146	0.8120	0.4459	0.1691	0.7818	0.8094	0.8719	0.0609
		PCA	0.7382	0.8356	0.8819	0.6622	0.4354	0.5550	0.4837	0.3609	0.7980	0.8437	0.9075	0.7124
	REFS	FS	0.7559	0.8131	0.8513	0.6665	0.6953	0.7960	0.8071	0.5105	0.7995	0.8369	0.9008	0.7145
		FW	0.6743	0.8053	0.8278	0.0292	0.6324	0.8075	0.7971	0.0060	0.7372	0.8042	0.8154	0.0476
		PCA	0.7220	0.8346	0.8696	0.6145	0.6879	0.8025	0.7896	0.5192	0.7468	0.8206	0.8815	0.6704
	CS	FS	0.7089	0.7226	0.7307	0.5978	0.2838	0.2588	0.1247	0.1078	0.2149	0.3754	0.2401	0.2209
		FW	0.7902	0.8160	0.4997	0.0179	0.6224	0.7994	0.0918	0.0179	0.8400	0.8121	0.0417	0.0179
		PCA	0.6465	0.6791	0.6723	0.5647	0.2767	0.3617	0.1481	0.0728	0.2562	0.3322	0.1825	0.1911
PearsonA	SI	FS	0.7992	0.7652	0.8477	0.6639	0.7863	0.8291	0.8825	0.7249	0.7818	0.8208	0.8600	0.7198
		FW	0.7043	0.8097	0.8573	0.0327	0.7443	0.8081	0.8789	0.2608	0.7818	0.8094	0.8666	0.0536
		PCA	0.7441	0.8252	0.8854	0.6619	0.7818	0.8379	0.8898	0.7238	0.7861	0.8128	0.9017	0.6975
	REFS	FS	0.7559	0.7660	0.8278	0.6855	0.6813	0.7703	0.8265	0.5203	0.7595	0.8223	0.8557	0.7041
		FW	0.6566	0.8095	0.8172	0.0319	0.6208	0.8037	0.7552	0.0060	0.7187	0.8096	0.8119	0.0208
		PCA	0.7372	0.8022	0.8715	0.6366	0.6464	0.7436	0.7677	0.4358	0.7349	0.8002	0.8561	0.6681
	CS	FS	0.8111	0.7926	0.8750	0.7215	0.2168	0.3585	0.0486	0.0539	0.2158	0.3638	0.1042	0.2166
		FW	0.8215	0.8019	0.8708	0.6069	0.5385	0.8111	0.0208	0.0060	0.8281	0.8032	0.0217	0.0179
		PCA	0.8111	0.8016	0.8788	0.6902	0.2403	0.2478	0.0498	0.0514	0.2984	0.2841	0.0430	0.0498
MICI	SI	FS	0.7311	0.7117	0.7079	0.5833	0.5026	0.5838	0.5324	0.4221	0.7947	0.7070	0.8538	0.7310
		FW	0.7449	0.8130	0.7067	0.5774	0.6949	0.8091	0.5378	0.4281	0.7714	0.8008	0.7835	0.0812
		PCA	0.6813	0.6813	0.7118	0.5890	0.4921	0.5416	0.5417	0.4239	0.8007	0.7497	0.8590	0.6935
	REFS	FS	0.6468	0.6501	0.7231	0.4533	0.6560	0.6746	0.7987	0.6260	0.6771	0.7129	0.7015	0.5879
		FW	0.6274	0.8134	0.6433	0.0927	0.6560	0.7983	0.7935	0.1870	0.6480	0.7967	0.6793	0.0119
		PCA	0.6363	0.7083	0.7501	0.4604	0.6560	0.6703	0.7984	0.6239	0.6960	0.7693	0.7314	0.6263
	CS	FS	0.8215	0.7795	0.8880	0.7155	0.4353	0.4436	0.4398	0.3339	0.7373	0.6684	0.7071	0.5311
		FW	0.8215	0.8067	0.8831	0.7155	0.7020	0.8075	0.4206	0.2764	0.8355	0.8112	0.6439	0.3036
		PCA	0.8215	0.7954	0.8918	0.7155	0.4593	0.4840	0.4211	0.3382	0.7355	0.6656	0.7254	0.5737
Cosine	SI	FS	0.7562	0.7462	0.8649	0.7708	0.1785	0.3610	0.0312	0.1236	0.7173	0.6751	0.7915	0.6794
		FW	0.6791	0.7984	0.8160	0.0466	0.3101	0.8114	0.0268	0.0179	0.7669	0.7946	0.7680	0.0785
		PCA	0.7631	0.7758	0.8785	0.7753	0.1276	0.2799	0.0572	0.1739	0.7147	0.7879	0.7997	0.7320
	REFS	FS	0.8215	0.8104	0.8864	0.7155	0.8215	0.8091	0.8899	0.7155	0.8215	0.8100	0.8781	0.7155
		FW	0.8215	0.8108	0.8839	0.7155	0.8215	0.8064	0.8861	0.7155	0.8215	0.7961	0.8824	0.7155
		PCA	0.8215	0.7953	0.8777	0.7155	0.8215	0.8067	0.8833	0.7155	0.8215	0.8031	0.8825	0.7155
	CS	FS	0.4464	0.4042	0.2731	0.3140	0.1526	0.3634	0.0312	0.1264	0.1523	0.2124	0.0833	0.1138
		FW	0.6891	0.7979	0.1074	0.0179	0.2842	0.8078	0.0268	0.0179	0.7839	0.8052	0.0119	0.0179
		PCA	0.3355	0.2762	0.2938	0.2352	0.1276	0.2776	0.0387	0.1666	0.1336	0.2285	0.0370	0.1340

Table A.14
Balanced accuracy obtained on BBBC021sc.

Distance	Measure	Dim. red.	AC			FSFS			HC		
			1NN	RF	SVM-Lin.	1NN	RF	SVM-Lin.	1NN	RF	SVM-Lin.
Pearson	SI	FS	0.2309	0.1791	0.3371	0.2241	0.1662	0.3541	0.2392	0.1794	0.3473
		FW	0.2079	0.1961	0.3929	0.2028	0.1976	0.3950	0.2152	0.1969	0.3902
		PCA	0.2355	0.1755	0.2923	0.2573	0.1592	0.3443	0.2591	0.1819	0.3277
	REFS	FS	0.2195	0.1800	0.3555	0.2292	0.1736	0.3500	0.2367	0.1799	0.3684
		FW	0.1715	0.1975	0.3876	0.2063	0.1967	0.3943	0.2125	0.1964	0.3880
		PCA	0.2478	0.1742	0.2909	0.2618	0.1671	0.3346	0.2569	0.1768	0.3349
	CS	FS	0.2220	0.1690	0.1957	0.0759	0.0539	0.0109	0.0792	0.0660	0.0249
		FW	0.2271	0.1971	0.4086	0.1403	0.1973	0.2969	0.2647	0.1964	0.3372
		PCA	0.2079	0.1567	0.1981	0.0854	0.0688	0.0261	0.0812	0.0722	0.0276
PearsonA	SI	FS	0.2267	0.1788	0.3233	0.2239	0.1722	0.3137	0.2363	0.1787	0.3253
		FW	0.1930	0.1973	0.3959	0.1991	0.1967	0.3896	0.2076	0.1967	0.3920
		PCA	0.2402	0.1739	0.2619	0.2608	0.1704	0.3277	0.2576	0.1807	0.2984
	REFS	FS	0.2192	0.1749	0.3285	0.2321	0.1804	0.3199	0.2429	0.1846	0.3284
		FW	0.1716	0.1972	0.3924	0.2109	0.1978	0.3800	0.2184	0.1976	0.3601
		PCA	0.2496	0.1700	0.3212	0.2634	0.1720	0.4014	0.2644	0.1869	0.3074
	CS	FS	0.2362	0.1735	0.2841	0.0734	0.0445	0.0379	0.0839	0.0765	0.0405
		FW	0.2212	0.1973	0.3910	0.1459	0.1970	0.2644	0.2699	0.1978	0.3472
		PCA	0.2310	0.1785	0.2620	0.0753	0.0642	0.0358	0.0866	0.0762	0.0318
MICI	SI	FS	0.2190	0.1770	0.3292	0.2345	0.1861	0.3971	0.2393	0.1911	0.3283
		FW	0.1848	0.1971	0.3875	0.2242	0.1970	0.3783	0.2177	0.1971	0.3881
		PCA	0.2498	0.1808	0.3059	0.2567	0.1837	0.3540	0.2495	0.1926	0.3272
	REFS	FS	0.2215	0.1882	0.3375	0.2240	0.1840	0.3422	0.2446	0.1830	0.3286
		FW	0.1845	0.1969	0.3865	0.2040	0.1970	0.3391	0.2244	0.1966	0.3825
		PCA	0.2514	0.1882	0.3059	0.2589	0.1787	0.3165	0.2608	0.1865	0.3132
	CS	FS	0.1811	0.1389	0.1317	0.2313	0.1677	0.3197	0.1321	0.1111	0.0744
		FW	0.2496	0.1961	0.4008	0.2579	0.1972	0.3246	0.2716	0.1956	0.3887
		PCA	0.1878	0.1498	0.1376	0.2306	0.1689	0.2948	0.1380	0.1201	0.0612
Cosine	SI	FS	0.2296	0.1835	0.3046	0.0621	0.0262	0.0179	0.2474	0.1934	0.3674
		FW	0.1967	0.1968	0.3889	0.2300	0.1969	0.2446	0.2280	0.1969	0.3812
		PCA	0.2418	0.1819	0.2969	0.0664	0.0187	0.0003	0.2513	0.1914	0.3199
	REFS	FS	0.2326	0.1909	0.3354	0.2417	0.1943	0.3502	0.2513	0.1902	0.3545
		FW	0.2024	0.1970	0.3861	0.2274	0.1982	0.3528	0.2382	0.1972	0.3600
		PCA	0.2520	0.1868	0.3396	0.2591	0.1884	0.3694	0.2577	0.1877	0.3403
	CS	FS	0.2315	0.1842	0.1991	0.1438	0.1176	0.0940	0.1237	0.0995	0.0549
		FW	0.2575	0.1976	0.4062	0.1565	0.1965	0.2749	0.2780	0.1964	0.3805
		PCA	0.2321	0.1850	0.2197	0.1356	0.1265	0.1006	0.1394	0.1257	0.0978

References

[1] J. Andreu-Perez, C.C.Y. Poon, R.D. Merrifield, S.T.C. Wong, G.-Z. Yang, Big data for health, IEEE J. Biomed. Health Inf. 19 (4) (2015) 1193–1208.

[2] K. Najarian, R. Splinter, Biomedical Signal and Image Processing, CRC Press, 2012.

[3] J. Ganesan, H.H. Inbarani, Hybrid tolerance rough set–firefly based supervised feature selection for MRI brain tumor image classification, Appl. Soft Comput. 46 (2016) 639–651.

[4] M.J. Sanderson, I. Smith, I. Parker, M.D. Bootman, Fluorescence microscopy, Cold Spring Harb. Protoc. (2014) 1042–1065.

[5] M. Gurcan, L. Boucheron, A. Can, A. Madabhushi, N. Rajpoot, B. Yener, Histopathological image analysis: A review, IEEE Rev. Biomed. Eng. 2 (2009) 147–171.

[6] N.S. Barteneva, E. Fasler-Kan, I.A. Vorobjev, Imaging flow cytometry, J. Histochem. Cytochem. 60 (10) (2012) 723–733.

[7] C.M. Bishop, Pattern Recognition and Machine Learning, Springer Science & Business Media, 2006.

[8] A.E. Carpenter, T.R. Jones, M.R. Lamprecht, C. Clarke, I.H. Kang, O. Friman, D.A. Guertin, J.H. Chang, R.A. Lindquist, J. Moffat, P. Golland, D.M. Sabatini, CellProfiler: Image analysis software for identifying and quantifying cell phenotypes, Genome Biol. 7 (10) (2006) R100.

[9] V. Ljosa, P.D. Caie, R. Ter Horst, K.L. Sokolnicki, E.L. Jenkins, S. Daya, M.E. Roberts, T.R. Jones, S. Singh, A. Genovesio, P.A. Clemons, N.O. Carragher, A.E. Carpenter, Comparison of methods for image-based profiling of cellular morphological responses to small-molecule treatment, J. Biomol. Screen. 18 (10) (2013) 1321–1329.

[10] S. Van Gassen, B. Callebaut, M.J. Van Helden, B.N. Lambrecht, P. Demeester, T. Dhaene, Y. Saeys, FlowSOM: Using self-organizing maps for visualization and interpretation of cytometry data, Cytometry A 87 (7) (2015) 636–645.

[11] Y. Saeys, I. Inza, P. Larrañaga, A review of feature selection techniques in bioinformatics, Bioinformatics 19 (2007) 2507–2517.

[12] P. Mitra, C. Murthy, S. Pal, Unsupervised feature selection using feature similarity, IEEE Trans. Pattern Anal. Mach. Intell. 24 (3) (2002) 301–312.

[13] D. Wettschereck, D.W. Aha, T. Mohri, A review and empirical evaluation of feature weighting methods for a class of lazy learning algorithms, Artif. Intell. Rev. 11 (1–5) (1997) 273–314.

[14] D. Peralta, S. del Río, S. Ramírez-Gallego, I. Triguero, J.M. Benitez, F. Herrera, Evolutionary feature selection for big data classification: A MapReduce approach, Math. Probl. Eng. 2015 (2015) 1–11.

[15] B. Jiang, F. Qiu, L. Wang, Multi-view clustering via simultaneous weighting on views and features, Appl. Soft Comput. 47 (2016) 304–315.

[16] I. Guyon, A. Elisseeff, An introduction to variable and feature selection, J. Mach. Learn. Res. 3 (2003) 1157–1182.

[17] Y. Kim, W.N. Street, F. Menczer, Feature selection in unsupervised learning via evolutionary search, in: Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM Press, New York, USA, 2000, pp. 365–369.

[18] S. Jones, L. Shao, A multigraph representation for improved unsupervised/semi-supervised learning of human actions, in: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2014, pp. 820–826.

[19] F. Pacheco, M. Cerrada, R.V. Sánchez, D. Cabrera, C. Li, J. Valente de Oliveira, Attribute clustering using rough set theory for feature selection in fault severity classification of rotating machinery, Expert Syst. Appl. 71 (2017) 69–86.

[20] S. Ramírez-Gallego, S. García, H. Mouriño-Talín, D. Martínez-Rego, V. Bolón-Canedo, A. Alonso-Betanzos, J.M. Benítez, F. Herrera, Data discretization: Taxonomy and big data challenge, Wiley Interdiscip. Rev.: Data Mining Knowl. Discov. 6 (1) (2016) 5–21.

[21] B.J. Frey, D. Dueck, Clustering by passing messages between data points, Science 315 (5814) (2007) 972–976.

[22] W.H. Au, K.C. Chan, A.K. Wong, Y. Wang, Attribute clustering for grouping, selection, and classification of gene expression data, IEEE/ACM Trans. Comput. Biol. Bioinform. 2 (2) (2005) 83–101.

[23] G. Li, X. Hu, X. Shen, X. Chen, Z. Li, A novel unsupervised feature selection method for bioinformatics data sets through feature clustering, in: IEEE International Conference on Granular Computing, 2008, pp. 41–47.

[24] S. Bandyopadhyay, T. Bhadra, P. Mitra, U. Maulik, Integration of dense subgraph finding with feature clustering for unsupervised feature selection, Pattern Recognit. Lett. 40 (1) (2014) 104–112.

[25] P.-Y. Zhou, K.C.C. Chan, An unsupervised attribute clustering algorithm for unsupervised feature selection, in: IEEE International Conference on Data Science and Advanced Analytics (DSAA), 2015.

[26] Y. Wang, Unsupervised representative feature selection algorithm based on information entropy and relevance analysis, IEEE Access 6 (2018) 45317–45324.

[27] S. Goswami, A.K. Das, A. Chakrabarti, B. Chakraborty, A feature cluster taxonomy based feature selection technique, Expert Syst. Appl. 79 (2017) 76–89.

[28] A. Esteva, B. Kuprel, R.A. Novoa, J. Ko, S.M. Swetter, H.M. Blau, S. Thrun, Dermatologist-level classification of skin cancer with deep neural networks, Nature 542 (7639) (2017) 115–118.

[29] T. Blasi, H. Hennig, H.D. Summers, F.J. Theis, J. Cerveira, J.O. Patterson, D. Davies, A. Filby, A.E. Carpenter, P. Rees, Label-free cell cycle analysis for high-throughput imaging flow cytometry, Nature Commun. 7 (2016) 10256.

[30] J. Simm, G. Klambauer, A. Arany, M. Steijaert, J.K. Wegner, E. Gustin, V. Chupakhin, Y.T. Chong, J. Vialard, P. Buijnsters, I. Velter, A. Vapirev, S. Singh, A.E. Carpenter, R. Wuyts, S. Hochreiter, Y. Moreau, H. Ceulemans, Repurposing high-throughput image assays enables biological activity prediction for drug discovery, Cell Chem. Biol. 25 (5) (2018) 611–618.

[31] H. Todorov, Y. Saeys, Computational approaches for high-throughput single-cell data analysis, FEBS J.

[32] R. Pepperkok, J. Ellenberg, High-throughput fluorescence microscopy for systems biology, Nat. Rev. Mol. Cell Biol. 7 (9) (2006) 690–696.

[33] H.M. Shapiro, Practical Flow Cytometry, John Wiley & Sons, Inc, Hoboken, NJ, USA, 2003.

[34] G. Gauglitz, G. Proll, Strategies for label-free optical detection, in: Advances in Biochemical Engineering/Biotechnology, Vol. 109, Springer Berlin Heidelberg, Berlin, Heidelberg, 2007, pp. 395–432.

[35] Y. LeCun, Y. Bengio, G. Hinton, Deep learning, Nature 521 (7553) (2015) 436–444.

[36] A. Krizhevsky, I. Sulskever, G.G.E. Hinton, Imagenet classification with deep convolutional neural networks, Adv. Neural Inf. Process. Syst. (2012) 1097–1105.

[37] D. Peralta, I. Triguero, S. García, Y. Saeys, J.M. Benitez, F. Herrera, On the use of convolutional neural networks for robust classification of multiple fingerprint captures, Int. J. Intell. Syst. 33 (1) (2018) 213–230.

[38] G. Montavon, W. Samek, K.R. Müller, Methods for interpreting and understanding deep neural networks, Digit. Signal Process. Rev. J. 73 (2018) 1–15.

[39] C. Rudin, Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead, Nat. Mach. Intell. 1 (5) (2019) 206–215.

[40] N. Japkowicz, S. Stephen, The class imbalance problem: A systematic study, Intell. Data Anal. 6 (5) (2002) 429–449.

[41] T. Chan, G. Golub, R. LeVeque, Updating formulae and a pairwise algorithm for computing sample variances, in: COMPSTAT 1982 5th Symposium, 1982, pp. 30–41.

[42] D. Peralta, Y. Saeys, Distributed, numerically stable distance and covariance computation with MPI for extremely large datasets, in: IEEE International BigData Congress, 2019, pp. 77–84.

[43] J.H. Ward, Hierarchical grouping to optimize an objective function, J. Amer. Statist. Assoc. 58 (301) (1963) 236–244.

[44] P.J. Rousseeuw, Silhouettes: A graphical aid to the interpretation and validation of cluster analysis, J. Comput. Appl. Math. 20 (C) (1987) 53–65.

[45] M. Meilă, Comparing clusterings—An information based distance, J. Multivariate Anal. 98 (5) (2007) 873–895.

[46] K. Fukunaga, Keinosuke, Introduction to Statistical Pattern Recognition, Academic Press, 1990.

[47] J.G. Moreno-Torres, J.A. Saez, F. Herrera, Study on the impact of partition-induced dataset shift on k-fold cross-validation, IEEE Trans. Neural Netw. Learn. Syst. 23 (8) (2012) 1304–1312.

[48] S. Maldonado, J. López, Dealing with high-dimensional class-imbalanced datasets: Embedded feature selection for SVM classification, Appl. Soft Comput. 67 (2018) 94–105.

[49] J. Laurikkala, Improving identification of difficult small classes by balancing class distribution, in: Proceedings 8th Conference on Artificial Intelligence in Medicine in Europe, Cascais, Portugal, 2001, pp. 63–66.

[50] H. Hennig, P. Rees, T. Blasi, L. Kamentsky, J. Hung, D. Dao, A.E. Carpenter, A. Filby, An open-source solution for advanced imaging flow cytometry data analysis using machine learning, Methods 112 (2017) 201–210.

[51] P. Eulenberg, N. Koehler, T. Blasi, A. Filby, A.E. Carpenter, P. Rees, F.J. Theis, F.A. Wolf, Reconstructing cell cycle and disease progression using deep learning, Nat. Com. 8 (1) (2017) art: 463.

[52] V. Ljosa, K.L. Sokolnicki, A.E. Carpenter, Annotated high-throughput microscopy image sets for validation, Nat. Methods 9 (7) (2012) 637.

[53] F. Herrera, S. Ventura, R. Bello, C. Cornelis, A. Zafra, D. Sánchez-Tarragó, S. Vluymans, Multiple Instance Learning. Foundations and Algorithms, Springer, 2016.

[54] J.C. Caicedo, S. Cooper, F. Heigwer, S. Warchal, P. Qiu, C. Molnar, A.S. Vasilevich, J.D. Barry, H.S. Bansal, O. Kraus, M. Wawer, L. Paavolainen, M.D. Herrmann, M. Rohban, J. Hung, H. Hennig, J. Concannon, I. Smith, P.A. Clemons, S. Singh, P. Rees, P. Horvath, R.G. Linington, A.E. Carpenter, Data-analysis strategies for image-based cell profiling, Nat. Methods 14 (9) (2017) 849–863.

[55] T.M. Cover, P.E. Hart, Nearest neighbor pattern classification, IEEE Trans. Inform. Theory 13 (1) (1967) 21–27.

[56] L. Breiman, Random forests, Mach. Learn. 45 (1) (2001) 5–32.

[57] M.M.A. Hearst, S.T.S. Dumais, E. Osman, J. Platt, B. Scholkopf, Support vector machines, IEEE Intell. Syst. Appl. 13 (4) (1998) 18–28.

[58] S.S. Keerthi, C.-J. Lin, Asymptotic behaviors of support vector machines with Gaussian kernel, Neural Comput. 15 (7) (2003) 1667–1689.

[59] C. Seiffert, T.M. Khoshgoftaar, J. Van Hulse, A. Napolitano, RUSBoost: A hybrid approach to alleviating class imbalance, IEEE Trans. Syst. Man Cybern. A 40 (1) (2010) 185–197.

[60] I. Steinwart, Support vector machines are universally consistent, J. Complexity 18 (3) (2002) 768–791.

[61] C. Kandaswamy, L.M. Silva, L.A. Alexandre, J.M. Santos, High-content analysis of breast cancer using single-cell deep transfer learning, J. Biomol. Screen. 21 (3) (2016) 252–259.

[62] F. Wilcoxon, Individual comparisons by ranking methods, Biom. Bull. 14 (1945) 80–83.

[63] M. Friedman, The use of ranks to avoid the assumption of normality implicit in the analysis of variance, J. Amer. Statist. Assoc. 32 (1937) 675–701.