A feature ranking algorithm for clustering medical data

Shpigelman E[1], Shamir R[1]

[1] The Blavatnik School of Computer Science, Tel Aviv University, Tel-Aviv, Israel

Short Title: A feature ranking algorithm for clustering medical data

Addresses:

[1] Tel Aviv University, P.O. Box 39040, Tel Aviv 6997801, Israel

Corresponding Author:

Prof. Ron Shamir, rshamir@tau.ac.il, Tel Aviv University, P.O. Box 39040, Tel Aviv 6997801, Israel, phone number: +97236405383

Abstract

The availability of electronic medical records (EMR) data has grown dramatically in recent years, and clustering methods are often applied to them for a variety of purposes, including finding unknown subtypes of diseases. The abundance and redundancy of information in EMR data raises the need to identify and rank the features that are most relevant for clustering.

Here we propose FRIGATE, an ensemble feature ranking algorithm for clustering, which uses the concepts of Shapley value and Multiplicative Weights. FRIGATE derives the importance of features from multiple clustering solutions on sub-groups of features. For each clustering solution a small group of features is ranked in a Shapley-like framework, and multiplicative weights are applied to limit the randomness of their choice. FRIGATE outperforms previously suggested ensemble ranking algorithms, both in solution quality and in speed.

1. Introduction

In the past two decades, medical systems around the world underwent a major digitization revolution [1]. As a result, most of the personal medical information is now stored electronically, transforming the way medical research is conducted. Although medical data sharing has been slow [2], the number of clinical data sets available to researchers is growing [3]. Such resources include data sets that span a large range of clinical data types, such as MIMIC [4], [5], and some even offer a combination of genomic and medical information, e.g., the UK BioBank [6].

Medical data have some unique challenging characteristics. Firstly, some of them are of great magnitude. For example, in MIMIC-III alone there is information of 46,520 patients, with 753 different lab tests and 14567 different ICD-9 codes [4] (each test or code is called a feature). Another challenge is the data incompleteness. Medical data typically have high percentage of missing values even for frequently taken measurements [7].

A growing number of machine-learning studies attempted to respond to these challenges on medical data [8] and developed computational tools dedicated to analyzing them [9], [10]. One type of such machine-learning models is clustering, an unsupervised approach, that is used for the discovery of new subgroups of known diseases [11]–[13]. Here patients are partitioned into subgroups based on their feature similarity. Our research is focused on this type of problems.

A key challenge in medical research is the interpretability of the results. When finding new clusters in the data, we want to understand the most important features that distinguish them, in order to assign a clinical meaning to each cluster and obtain clinical insights. When dealing with large data sets with possibly thousands of features, this is challenging. Also, running the algorithms on huge data sets is computationally expensive and even prohibitive. For these reasons, feature selection algorithms, which seek the most important features for the clustering task, were proposed [14]. Our goal here is the development of such an algorithm that ranks all the features according to their importance to the clustering task, in a way that specifically fits medical data.

Many medical databases contain a large number of features. One way to deal with the large number is dimension reduction [15], but such procedures obscure the effect of individual features, which is crucial for medical insights. Another option is to use feature selection algorithms, which choose a subset of features that will create a sub matrix with “good” clusters. There are several feature selection methods for clustering algorithms [14], [16]. In recent years several ensemble feature ranking algorithms were suggested, which create an ensemble of clustering solutions on subsets of features and then use some metric to evaluate the contribution of each feature [17].These include FRMV [18] , FRCM [19], and FRSD [17]. These methods were shown to perform better than the traditional filter and wrapper methods, including on medical data sets [17]–[19]. Ensemble methods can also be used for choosing a subset of important features, in addition to ranking the full set of features [20]. Here we develop a new algorithm within the ensemble ranking framework. As we aim to work with medical information, we prefer to lose as little information as possible and thus rank the full set of features.

We introduce a new algorithm called FRIGATE (Feature Ranking In clustering using GAme ThEory), which uses two concepts from game theory. The first is motivated by Shapley value, a measure of the contribution of every player to the group in a cooperative game [21], [22]. In our case the players are the features and the “game” is clustering. Shapley values are widely used for feature evaluation in classification models [22] and so far were not used in clustering for feature selection or ranking. The second is Multiplicative Weights (MW) [23], a framework to improve the selection of players by iteratively selecting the players from a distribution based on their performance so far. In FRIGATE we use MW to guide the choice of features for each clustering solution and thus reduce the chance to choose features that proved to be insignificant. All previously presented ensemble algorithms choose subsets of features at random. To the best of our knowledge this is the first time that MW is adapted to feature selection for clustering.

The paper is organized as follows: we first present relevant background on clustering methods, ensemble feature ranking for clustering and relevant game theory concepts. Next, in the Methods section, we present the FRIGATE algorithm, describe the construction of simulated data and demonstrate a run of FRIGATE. In the Results section we measure the performance of FRIGATE and the extant ensemble algorithms both on simulated and on 11 different real genomics and EMR datasets. We conclude with a discussion of the results.

2. Background

In this chapter we describe the computational methods that will be used in the paper.

A fundamental, broadly used clustering algorithm for data with real-valued features is k-means [24]. Given the number of clusters k, it selects k points in R^d called centroids, assigns samples to the closest centroid and recomputes the new centroid of each resulting set. The process is iterated till convergence.

k-modes [25] is a variant of k-means for categorical data, namely, where feature values are discrete (two or more) categories. The Hamming distance is used as the distance metric instead of Euclidean distance. Here we used the k-modes implementation in [26]. k-prototypes [27] is an algorithm that clusters mixed data, i.e. data with both continuous and categorical features. The distance metric is:

$$d(x, y) = \sum_{i=1}^{p}(x_i - y_i)^2 + \gamma \sum_{i=p+1}^{m} \delta(x_i, y_i) \quad (1)$$

Where $x_1, \dots x_p$ are numerical variables, $x_{p+1}, \dots x_m$ are categorical variables, and δ is the Hamming distance function. The γ factor determines the relative contribution of the categorical features in comparison to the continuous features. k-prototypes was reported as one of the best performers in a recent benchmark of mixed-data clustering algorithms [28]. Here we used a k-prototype implementation in [26].

We now briefly describe extant ensemble feature ranking algorithms for clustering. In the **FRMV** algorithm [18], in each iteration of the algorithm a clustering solution is obtained for a subset of features, which are ranked based on some relevance measure (e.g. linear correlation). The final feature ranking is done according to the average rank. **FRCM** [19] was originally designed for genomic data. It does not require k as an input. For each run of k-means on a subset of features, k is selected uniformly at random from a prescribed range. Features are ranked based on a measure similar to Adjusted Rand Index [29] which measures the similarity between a consensus matrix for the clustering solutions, and a matrix for each feature, representing distances between pairs of samples for that feature. Finally, in the **FRSD** algorithm [17], in each iteration the algorithm randomly chooses a subset of the features, produces a clustering solution using k-means and ranks the selected features based on the change in the silhouette score [30] after shuffling the values of the feature. A prescribed range of k values is tested and the final score is based on the average rank of the iterations per k and over all values of k. Since no implementations were provided for the three algorithms, we implemented them as described in [17]–[19]. For FRMV we used the linear correlation as the relevance measure. For FRSD we used silhouette as implemented in Scikit learn [31]. In all cases we used k-means for clustering.

Shapley Values - In cooperative game theory, a set N of players can form coalitions. Each coalition $S \subset N$ has a value $g(S)$. According to the Shapley theory [21] the contribution of player i to group $S \cup \{i\}$ is defined as:

$$g(S \cup \{i\}) - g(S) \tag{2}$$

and the Shapley value of player i is a weighted average of its contributions over all possible Ss, i.e.:

$$\sum_{S \subseteq N \setminus \{i\}} \frac{|S|!(n-|S|-1)!}{n!} [g(S \cup \{i\}) - g(S)] \tag{3}$$

This value is widely used in supervised learning to measure the contribution of a feature to a prediction model, where for efficiency reasons it is usually evaluated using random permutations instead of enumerating all possible groups S [22]. To the best of our knowledge, Shapley values were not used to date in feature selection for clustering.

Multiplicative Weights - Multiplicative Weights (MW) is an algorithmic update method used in game theory and algorithm design. The motivation of MW [23] is to iteratively improve the decisions one makes by gradually favoring decisions that were proven to be right so far. In our case the decisions are the features selected and we use the Hedge update rule that was suggested by Arora et al. [23]:

$$w_i^{(t+1)} = w_i^{(t)} \cdot exp(-\eta m_i^t) \tag{4}$$

Where w_i^t is the weight of feature i at the t-th iteration, $\eta \leq 1$ is a constant parameter and m_i^t is the cost of feature i at iteration t. m_i^t is a value in the range $[-1,1]$ that reflects how good decision i was in iteration t, where higher positive values correspond to worse decisions and negative values correspond to good decisions that warrant an award instead of a cost. A common practice, also used in our implementation, is to use non-negative values only.

3. Methods

3.1 The FRIGATE algorithm

FRIGATE is a new ensemble feature ranking algorithm, which uses the Shapley value concept to find the most valuable features for clustering based on multiple runs of k-means (or k-prototypes/k-modes) algorithm.

To use the Shapley values in our context, the players are the features. We assume the number of clusters k is given, and use the total distance of samples to their cluster centroids as the objective function g:

$$g(S) = \sum_{j=1}^{k} \sum_{x \epsilon C_j} d^S \left(x, y_j\right) \qquad (5)$$

Here S is a set of features, k is the number of clusters, C_j is the set of samples included in cluster j and y_j is the centroid of cluster j. d^S is the distance function on the sample vectors restricted to the coordinates in S. We call $g(S)$ the *solution score*.

d^S and the clustering algorithm that we use will depend on the data types in S. If all the features are continuous then we use k-means for clustering and the Euclidean distance. When we have a mixture of categorical and continuous features, we will use k-prototype for clustering, and the corresponding distance function (Equation 1). If we have only categorical features, k-modes is used for clustering with d as the Hamming distance. We used k-means as implemented in Scikit learn [31] with 100 k-means++ initializations in each run.

Algorithm 1 presents the procedure for continuous features. In iteration t, the algorithm selects at random a subset of features and performs k-means on the corresponding submatrix $A^{(t)}$. Once a solution has been obtained, we calculate the contribution of feature i as the difference between that solution's score and the score obtained by the same clustering on the submatrix $A^{(t)}$ in which the values of feature i were randomly shuffled among the samples, keeping the rest unchanged. For the final ranking we use the average scores of the features.

```
Algorithm 1: FRIGATE

    Input: $\boldsymbol{A}$ - $m$x$n$ matrix of $m$ samples and $n$ features; $\boldsymbol{k}$ - number of clusters; $\boldsymbol{T}$ -
    number of iterations; $\boldsymbol{f}$ - fraction of features to use in each iteration
    Output: $\boldsymbol{R}$ – list of the $m$ features ordered by importance for clustering
1   $scores \leftarrow$ array of length $n$ for keeping score of each feature, initialized to 0s
2   $counts \leftarrow$ array of length $n$ for counting the times each feature is selected, initialized
    to 0s
3   for $t \leftarrow 1$ to $T$
4   |   $h \leftarrow$ a set of $q = \lceil f \cdot n \rceil$ randomly chosen features
5   |   $A^{(t)} \leftarrow$ a matrix of size $m$x$q$ with columns corresponding to $h$
6   |   Perform k-means on $A^{(t)}$
7   |   $I \leftarrow$ labels of the clustering solution
8   |   $g(h) \leftarrow$solution score of $A^{(t)}$ and $I$
9   |   for $v$ in $h$
10  |   |   $\hat{v} \leftarrow$ Shuffled version of $v$
11  |   |   $A_v^{(t)} \leftarrow$ a matrix identical to $A^{(t)}$ except having $\hat{v}$ instead of $v$
12  |   |   $g_v \leftarrow$ solution score of $A_v^{(t)}$ and $I$
13  |   |   $scores[v] \leftarrow scores[v] + (g_v - g(h))$
14  |   |   $counts[v] \leftarrow counts[v] + 1$
15  |   end
16  end
17  $scores \leftarrow scores/counts$
18  return the features sorted in decreasing order of scores
```

Note that FRSD can also be seen as a type of a Shapley-like algorithm with a function g that uses the silhouette. However, a main difference is that FRIGATE does not rank the features on every iteration and accumulates the ranks for the final score, as in FRSD and FRMV, but instead summarizes the raw scores. That way poor clustering solutions that are based on non-informative features will have large $g(h)$ values (line 8 in Algorithm 1) as well as large g_v values (line 12). This will limit the ability of these features to receive high scores, as they are calculated by subtracting the distance after shuffling the values of a feature from the original distance (line 13 in Algorithm 1). Thanks to these properties of $g(h)$ and g_v, we do not need to use an additional factor, as FRSD does with silhouette, to assess the quality of the clusters. It also reduces the number of calculations and improves the efficiency of the algorithm.

We now discuss runtime complexity, referring only to k-means for simplicity. The runtime of k-means is $O(m \cdot q \cdot k \cdot c)$ for m samples, q features, k clusters and up to c iterations. We sample in each FRIGATE iteration $q = f \cdot n$ features. For each k-means run we perform i initializations. Therefore, the runtime of the k-means executions in each iteration of FRIGATE is $O(mqkci)$. Other than k-means runs, in each iteration we shuffle the values of q features over the full cohort in $O(m)$ for each feature and recalculate the solution score d_v in $O(m)$. The overall runtime of an iteration is $O(mqkci + mq) = O(mqkci)$. Hence, the additional actions to test the contribution of each feature do not increase the asymptotic runtime. We perform T iterations, so the overall runtime is $O(mTqkci)$. As $q = f \cdot n$ with constant f we can write the runtime as: $O(mTnkci)$.

3.2 The FRIGATE-MW algorithm

MW offers a smarter way to choose the features in FRIGATE for each clustering solution instead of choosing them randomly. Algorithm 2 shows the version of FRIGATE that uses MW for continuous features, which we call FRIGATE-MW.

We define an n-long array L so that $L(i) = \frac{i-1}{n-1}$. At each iteration we rank the features by their scores so far and use the ranks and L to determine $m_i^{(t)}$ (see chapter 2). If the rank of feature i at iteration t is r then $m_i^{(t)} = L[r]$. The weights of features that were not selected in the iteration remain unchanged. For the next iteration we select features from distribution $\boldsymbol{p}^{(t)} = \{w_1^{(t)}/\Phi^{(\mathrm{t})}, \dots, w_N^{(t)}/\Phi^{(\mathrm{t})}\}$ where $\Phi^{(\mathrm{t})} = \sum_i w_i^{(t)}$ is the sum of weights at the t-th iteration. To the best of our knowledge, this is the first use of MW in feature selection for clustering.

Algorithm 2: FRIGATE-MW

```
Input: $A$ - $m$x$n$ matrix of $m$ samples and $n$ features; $k$ - number of clusters; $T$ - number of iterations; $f$ - fraction of features to use in each iteration; $\eta$ – a Multiplicative Weights parameter
Output: $R$ – list of the $m$ features ordered by importance for clustering
$scores \leftarrow$ array of length $n$ for keeping score of each feature, initialized to 0s
$counts \leftarrow$ array of length $n$ for counting the times each feature is selected, initialized to 0s
$weights \leftarrow$ array of length $n$ for keeping the weight of each feature, initialized to 1s
$L \leftarrow$ a static array of length $n$ for the costs used in Multiplicative Weights. $L = [0, \frac{1}{n-1}, \frac{2}{n-1} \dots, \frac{(n-2)}{n-1}, 1]$
for $t \leftarrow 1$ to $T$
    $P \leftarrow weights/sum(weights)$
    $h \leftarrow$ a set of $q = \lceil f \cdot n \rceil$ features chosen from the distribution $P$
    $A^{(t)} \leftarrow$ a sub matrix of size $mxq$ of $A$ with columns corresponding to $h$
    Perform k-means on $A^{(t)}$
    $I \leftarrow$ labels of the clustering solution
    $g(h) \leftarrow$the solution score of $A^{(t)}$ and $I$
    for $v$ in $h$
        $\hat{v} \leftarrow$ Shuffled version of $v$
        $A_v^{(t)} \leftarrow$ a matrix identical to $A^{(t)}$ except having $\hat{v}$ instead of $v$
        $g_v \leftarrow$the solution score of $A_v^{(t)}$ and $I$
        $scores[v] \leftarrow scores[v] + (g_v - g(h))$
        $counts[v] \leftarrow counts[v] + 1$
    end
    $ranks \leftarrow sort(scores/\ counts)$ // rank the features based on the scores so far
    for $v$ in $h$
        $r \leftarrow$ rank of $v$ in $ranks$
        $weights[v] = weights[v] \cdot \exp(-\eta \cdot L[r])$ // update the weight of $v$ according to eq. 14
end
$scores \leftarrow scores/counts$
return the features sorted in decreasing order of scores
```

In each iteration we update the weights of the q participating features in constant time for each feature and sort the array of weights in $O(n \cdot \log(n))$. The overhead of MW for each iteration is thus $O(n \cdot \log(n) + q) = O(n \cdot \log(n))$, since $q < n$. The total runtime of each iteration in FRIGATE-MW is: $O(mqkci + n \cdot \log(n))$. Therefore, the total runtime is: $O(mTqkci + Tn \cdot$

$\log(n)) = O\big(Tn(mkci + \log(n))\big)$. Altogether, the increase in the runtime over FRIGATE is not major. However, note that in FRIGATE-MW the iterations cannot be programmed to run in parallel, in contrast to FRIGATE.

For both variations of the algorithm we used $T = 2n$ and $f = 0.1$, and for FRIGATE-MW we used $\eta = 0.5$. For a detailed description of the parameter choice see Supplementary 2.

3.3 Simulation

We performed simulations in order to test the algorithms in situations where the true clustering and the informative features are known. The simulations were along the same lines of those described in [17]. The parameters of the simulation are:

- k – number of clusters
- c – number of samples in each cluster
- α – number of informative features
- β – number of non-informative features
- μ – distribution parameter
- σ – correlation coefficient between features

Simulating continuous data: For each cluster j, we construct c vectors of length $n = \alpha + \beta$ from multivariate normal distribution, where α features are sampled from a normal distribution with mean of $j \cdot \mu$ for $j\epsilon[0, \dots, k-1]$. The other β features are sampled from a normal distribution with mean 0 for all clusters and therefore represent the non-informative features. Thus, the mean vector of a sample in the j^{th} cluster is: $\mu_j = [(j \cdot \mu)_{\alpha\text{x}1}, 0_{\beta\text{x}1}]$.

Next, we define a covariance matrix, parameterized by σ, used to create correlations between the different features. The covariance matrix Σ is identical for all clusters:

$$\Sigma = (1-\sigma)\cdot I_{nxn} + \sigma \cdot 1_{nx1} \cdot 1_{nx1}^{T} \tag{6}$$

The $nx(k \cdot c)$ data matrix A then undergoes z-score normalization for each feature. This step is needed when working with many data types, especially in the medical domain as the values of different features can be of different magnitude.

Simulating mixed data: To build a simulation of mixed data we add three more parameters:

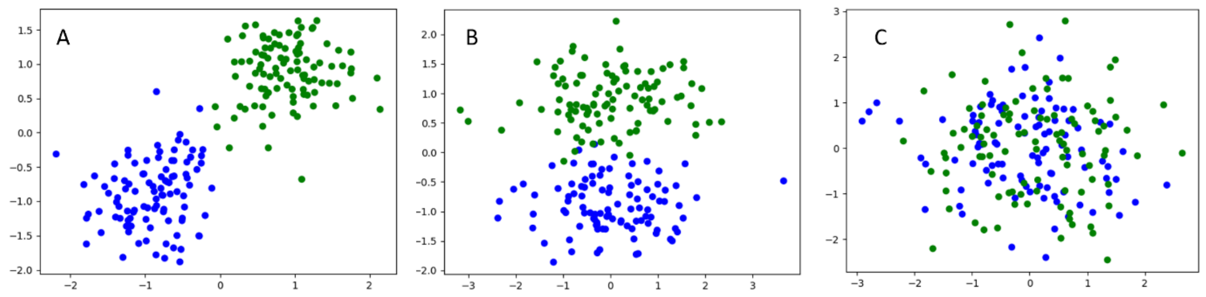
- $\alpha_{categorical}$ – number of informative categorical features
- $\beta_{categorical}$ – number of non-informative categorical features
- p – probability of choosing the right category

We assume that the categorical features have k categories, labeled $\{0,1,\dots,k-1\}$. For the informative features of a sample in the j^{th} cluster, we choose the value j with probability p and a value from $\{0,\dots,k-1\}\backslash\{j\}$ with probability $1-p$ where the value is chosen uniformly at random. For the non-informative features we choose a random value uniformly from $\{0,\dots,k-1\}$. The simulation of the continuous features is done as described before, and we concatenate the two matrixes into a single input matrix. In our simulation we used $p = 0.95$.

3.4 Demonstration of FRIGATE

For better understanding of the FRIGATE process, we demonstrate it graphically. We simulated data as described in section 3.3, with two continuous features, two clusters ($k = 2$), and 100 samples in each cluster, and simulation parameters $\mu = 4$, $\sigma = 0$. Figure 1 shows the data, where each axis is a feature and the samples are colored by cluster membership. We simulated three scenarios:

A. Both features are informative for the clustering solution (Figure 1A).
B. Only one feature is informative (Figure 1B)
C. Both features are not informative (Figure 1C).



D

Scenario	Solution Score	Scores after shuffling values of F1		Scores after shuffling values of F2	
		Solution	Feature	Solution	Feature
A	81.76	395.24	313.48	368.09	286.33
B	237.28	237.79	0.51	541.33	304.05
C	256.48	374.07	114.59	407.04	150.56

Figure 1. Illustrations of simulations with two clusters. In the simulation, there are 100 samples in each cluster, $\mu = 4$, $\sigma = 0$, and two features. A-C: features are represented by the axes. Each color represents a different cluster. A: both features are informative for clustering, B: only the feature represented by the y axis is informative, C: both features are not informative. D: Demonstration of FRIGATE iteration on the data of A-C. Solution score refers to line 8 in Algorithm 1, feature's score refers to line 13 in Algorithm 1. F1 is represented by the x-axis in Figure1 A-C and F2 is represented by the y-axis. We can see that the informative features received higher scores than the non-informative ones.

Next, we performed an iteration of the FRIGATE algorithm, using the centroids obtained from the clustering solution on the two features, to show the differences in scores in each scenario (Figure 1D):

A. When the two features were informative, the solution score (line 8 in Algorithm 1) was 81.76, and the scores of the features (line 13 in Algorithm 1) were 313.48 and 286.33. Both feature scores are high, and the difference can result from the randomness in shuffling the values (line 10 in Algorithm 1) or from the simulation that might have produced one feature that is more informative than the other.

B. When only one feature was informative, the solution score was 237.28, and the feature scores were 0.51 for the non-informative features and 304.05 for the informative feature.

C. When the two features were non-informative, the solution score was 256.48, and the feature scores were 117.59 and 150.57.

In Figure 2 we demonstrate graphically the iteration for scenario 2 (line 2 in Table 2). Figure 2A shows the results of k-means clustering of the data (line 6 in Algorithm 1), with a solution score of 237.28. Figure 2B shows the data after shuffling the values of the non-informative feature (line 10 in Algorithm 1). The shuffled data has an almost identical solution score of 237.79 (line 12 in Algorithm 1) and a feature score of 0.51. Figure 2C shows the sample locations after shuffling the values of the informative feature, which gives a new solution score of 541.33 and a feature score of 304.05.

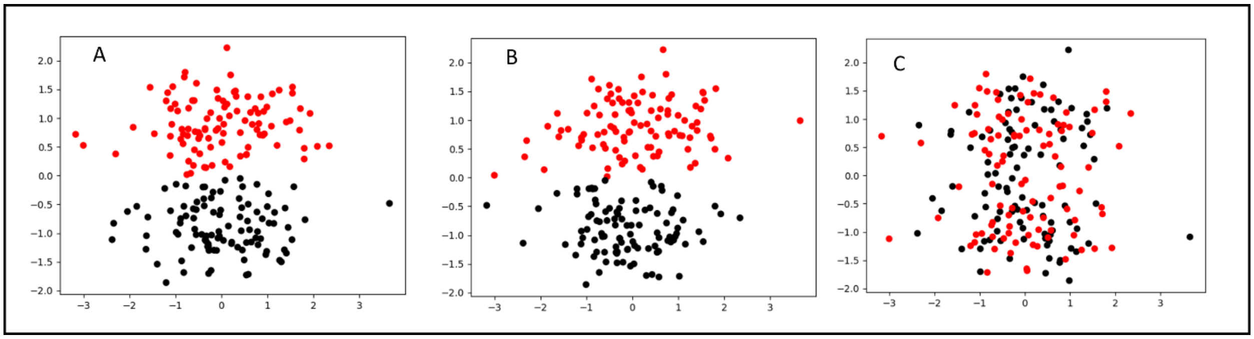


Figure 2. Illustrations of the different steps of FRIGATE for scenario 2 that is shown in Figure 1B, where one feature is informative (y axis) and one is non-informative (x axis). A – a clustering solution of the data (line 6 in Algorithm 1) colored by clusters labels. The solution score is 237.28 (line 8 in Algorithm 1). B- Results of shuffling the x coordinates, representing the non-informative feature (lines 10-13 in Algorithm 1). The solution score is similar to A and the feature's score is 0.51. C- Results of shuffling the y coordinates, representing the informative feature. An increase in the solution score led to a feature's score of 304.5.

The illustrations of scenarios A and C are given in Supplementary 3. In all scenarios the informative features scored much higher than the non-informative ones. Notice that the differences in scores are due to the initial solution score of each scenario – the poor results of scenario 3 already produced a relatively high solution score, so the ability of any feature to score high is limited.

3.5 Evaluation measures

When applied to a real dataset, each algorithm produces a ranking of the features. In our tests the truly informative features were unknown but the "true" clustering is known. We therefore applied the following procedure from [17]–[19] to evaluate the results. We ran k-means on the subset of the data containing only the j top ranked features. The clustering produced was compared to the true labels available for the dataset using the Adjusted Rand Index (ARI) [29]. The process was repeated with increasing values of j, for $j\epsilon[1, N]$ for N number of features. The rationale was that a better feature ranking will manifest a high ARI for smaller values of j, as it puts the most informative features at the top. The process was repeated ten times per algorithm.

The above measure gives a value for the top j features, and a separate value for each j. We developed two new scores that summarize the measure across all values of j, while giving higher weight to the features that rank higher.

Suppose M feature ranking algorithms are compared on the same dataset. For each j, we compute the ARI of each algorithm on the top j features that it selected, and rank the algorithms based on their scores, from 1 for the top performer to M. For simplicity of the description, we assume there are no ties. The *weighted rank* of algorithm a is defined as:

$$WR(a) = \frac{2}{N(N+1)}\sum_j (N - j + 1) * \left(\frac{M - rank(a,j) + 1}{M}\right) \quad (7)$$

Here $rank(a, j)$ is the rank of algorithm a on the top j features. Hence, the second factor in the sum ranges from 1 for the top ranked algorithm to $1/M$ for the worst ranked, and the first factor gives a different weight to each j, from N for the first feature to 1 for the last ranked. The factor $\frac{2}{N(N+1)}$ rescales the total sum to $[0,1]$.

The WR measure is relative and depends on the set of algorithms tested. We introduce a second measure for a single algorithm. The algorithm's ARI score is computed for each top j features and weighted as above. The *weighted ARI* of algorithm a is defined as:

$$WARI(a) = \frac{2}{N(N+1)}\sum_j (N - j + 1) * ARI(a,j) \quad (8)$$

where $ARI(a,j)$ is the ARI of algorithm a on the top j features. Hence, the range of the score is [-1,1] and higher scores are better.

Both scores can be generalized to handle ties and also situations where not all values of j are tested, e.g., when there are too many features.

4. Results

4.1 Algorithms Performance

We measured the performance of FRMV, FRSD, FRCM, FRIGATE and FRIGATE-MW on simulated and real data, including four genomic and seven EMR datasets. The number of clusters k in FRIGATE and for FRMV was chosen with the elbow method that we implemented as suggested in [32].

4.1.1Simulated Data

We simulated data with 200 samples and 100 features of which 20 are informative, divided into two or four equal-sized clusters ($k = \{2,4\}$), mean distances $\mu = \{0.5,1,2,4\}$ and feature correlation levels $\sigma = \{0,0.05,0.2, 0.5\}$. We ran the algorithms on data with and without z-score normalization. The *accurate recognition rate* is defined as the fraction of informative features in the top 20 ranked features. Results for $k = 4$ with $\mu = \{0.5,1\}$ are shown in Tables 1, and the other cases are found in Supplementary 5. In all cases, the elbow method chose $k = 2$. On normalized data FRCM performed best, and FRIGATE-MW second. On non-normalized data FRIGATE-MW was best. FRMV scored poorly in all cases. FRSD scored poorly in most normalized scenarios, while in most non-normalized scenarios it scored high. We can also see that in general smaller values of μ and k account for harder cases, and normalized data is more challenging than non-normalized data. The FRIGATE variations and FRCM are affected by the correlation levels, where high levels of correlation cause a drop in performance. We can see the major drop in performance of these algorithms for $\sigma \geq 0.2$. FRSD and to some extant FRMV show opposite behavior, where extreme levels of correlation lead to improved results. This is counter-intuitive, as high correlation levels are expected to cause higher similarities between all features, including pairs of informative and non-informative ones. FRSD and FRMV are also more affected by the structure of the data (k, μ, normalized. See Supplementary 5) in comparison to FRIGATE and FRCM (see Discussion).

It is worth mentioning that as 20% of the features were informative, a score below 0.2 accounts for performance worse than random ordering of features. FRMV repeatedly scored below 0.2, FRSD scored low for most of the normalized cases with low correlation levels, and FRIGATE scored below random levels in the extreme correlation setting. FRCM is the only algorithm that rarely dropped significantly below random levels (Supplementary 5).

parameters	$\mu = 0.5$ $\sigma = 0$	$\mu = 0.5$ $\sigma = 0.05$	$\mu = 0.5$ $\sigma = 0.2$	$\mu = 0.5$ $\sigma = 0.5$	$\mu = 1$ $\sigma = 0$	$\mu = 1$ $\sigma = 0.05$	$\mu = 1$ $\sigma = 0.2$	$\mu = 1$ $\sigma = 0.5$
normalized								
FRIGATE	0.98 ± 0.03	0.91 ± 0.07	0.46 ± 0.17	0.09 ± 0.08	$\mathbf{1 \pm 0}$	$\mathbf{1 \pm 0}$	0.97 ± 0.05	0.09 ± 0.08
FRIGATE-MW	$\mathbf{1 \pm 0}$	$\mathbf{1 \pm 0}$	0.62 ± 0.33	0.19 ± 0.15	$\mathbf{1 \pm 0}$	$\mathbf{1 \pm 0}$	$\mathbf{1 \pm 0}$	0.01 ± 0.02
FRCM	$\mathbf{1 \pm 0}$	$\mathbf{1 \pm 0}$	$\mathbf{0.72 \pm 0.15}$	0.35 ± 0.18	$\mathbf{1 \pm 0}$	$\mathbf{1 \pm 0}$	$\mathbf{1 \pm 0}$	$\mathbf{0.99 \pm 0.03}$
FRSD	0.06 ± 0.04	0.06 ± 0.04	0.11 ± 0.06	$\mathbf{0.38 \pm 0.17}$	0.01 ± 0.02	0 ± 0	0.04 ± 0.05	0.32 ± 0.08
FRMV	0.13 ± 0.16	0.13 ± 0.13	0.25 ± 0.16	0.16 ± 0.12	0.05 ± 0.1	0.03 ± 0.03	0.09 ± 0.12	0.06 ± 0.16
non-normalized								
FRIGATE	0.99 ± 0.02	0.98 ± 0.03	0.76 ± 0.12	0.7 ± 0.15	$\mathbf{1 \pm 0}$	$\mathbf{1 \pm 0}$	$\mathbf{1 \pm 0}$	$\mathbf{1 \pm 0}$
FRIGATE-MW	$\mathbf{1 \pm 0}$	$\mathbf{1 \pm 0.02}$	$\mathbf{0.94 \pm 0.08}$	0.31 ± 0.14	$\mathbf{1 \pm 0}$	$\mathbf{1 \pm 0}$	$\mathbf{1 \pm 0}$	$\mathbf{1 \pm 0}$
FRCM	$\mathbf{1 \pm 0}$	0.99 ± 0.02	0.82 ± 0.1	0.45 ± 0.2	$\mathbf{1 \pm 0}$	$\mathbf{1 \pm 0}$	$\mathbf{1 \pm 0}$	$\mathbf{1 \pm 0}$
FRSD	0.79 ± 0.06	0.74 ± 0.11	0.77 ± 0.07	$\mathbf{0.89 \pm 0.06}$	$\mathbf{1 \pm 0.02}$	$\mathbf{1 \pm 0}$	$\mathbf{1 \pm 0}$	$\mathbf{1 \pm 0}$
FRMV	0.2 ± 0.19	0.12 ± 0.2	0.23 ± 0.2	0.11 ± 0.08	0.02 ± 0.06	0.02 ± 0.04	0.13 ± 0.2	0.1 ± 0.15

Table 1. Performance on simulated data, with $k = 4$. In **bold** are the top performers.

4.1.2 Real Data

We tested the five algorithms on 11 real genomic and EMR datasets from different sources for which a known clustering was available or created by us. The datasets are described in Table 2.

Figure 3 shows the performance of the algorithms on four genomic databases [33]–[36] (datasets 1-4 in Table 2). These datasets were used in a benchmark of clustering [37]. They have a large number of features and a modest number of samples (about two orders of magnitude lower). Note that here we do know the true clustering but we do not know which and how many features are informative, but it is expected that many features do not carry information relevant to the clustering. In all cases the value chosen by the elbow method for FRIGATE and FRMV was $k = 2$.

The performance of both variations of FRIGATE and FRSD was comparable and generally good, reaching maximum ARI of 0.35-0.7 already with less than 100 features in most cases. FRSD performed markedly better than the other methods on dataset 3 (Figure 3C). FRCM performed poorly in most cases, with slow gradual increase in ARI. FRMV performed better than the others on dataset 2 (Figure 3B), and its results had a wide variance across repetitions in most cases. It is worth mentioning that the description of the FRMV algorithm in [18] was not clear, especially calculating linear correlation between continuous features and categorical cluster membership. This, as well as sampling features with replacement, can potentially create major variability between different runs of the algorithm.

No.	Source	Domain	Data Name	# of clusters	# of samples	# of features	Data type
1	[34], [37]	Genomic	Bredel-2005	3	50 (31,14,5)	1739	Continuous
2	[33], [37]	Genomic	Armstrong-2002-v2	3	72 (24,20,28)	2194	Continuous
3	[35], [37]	Genomic	Tomlins-2006	5	50 (27,20,32,13,12)	2315	Continuous
4	[36], [37]	Genomic	Nutt-2003-v1	4	50 (14,7,14,15)	1377	Continuous
5	MIMIC-III [4], [5]	EMR	Young cancer patients	2	161 (122,39)	70	Continuous

6	MIMIC-III [4], [5]	EMR	Young healthy patients	2	110 (84,26)	47	Continuous
7	MIMIC-III [4], [5]	EMR	Newborns	2	5286 (1534,3752)	29	Continuous
8	[38], [39]	EMR	Heart failure	2	169 (68,101)	77	Continuous
9	eICU [40], [41]	EMR	Intubated patients	2	441 (136, 305)	157 (87, 70)	Mixed
10	eICU [40], [41]	EMR	Short stay at ICU	2	570 (487, 83)	79 (59, 20)	Mixed
11	eICU [40], [41]	EMR	Young patients	2	232 (138, 94)	86 (72, 14)	Mixed

Table 2. Details of the real data sets used for the performance benchmark. The numbers in parentheses in column "# of samples" are the sizes of the clusters, and in the column "# of features" are the number of continuous and categorical features, respectively.

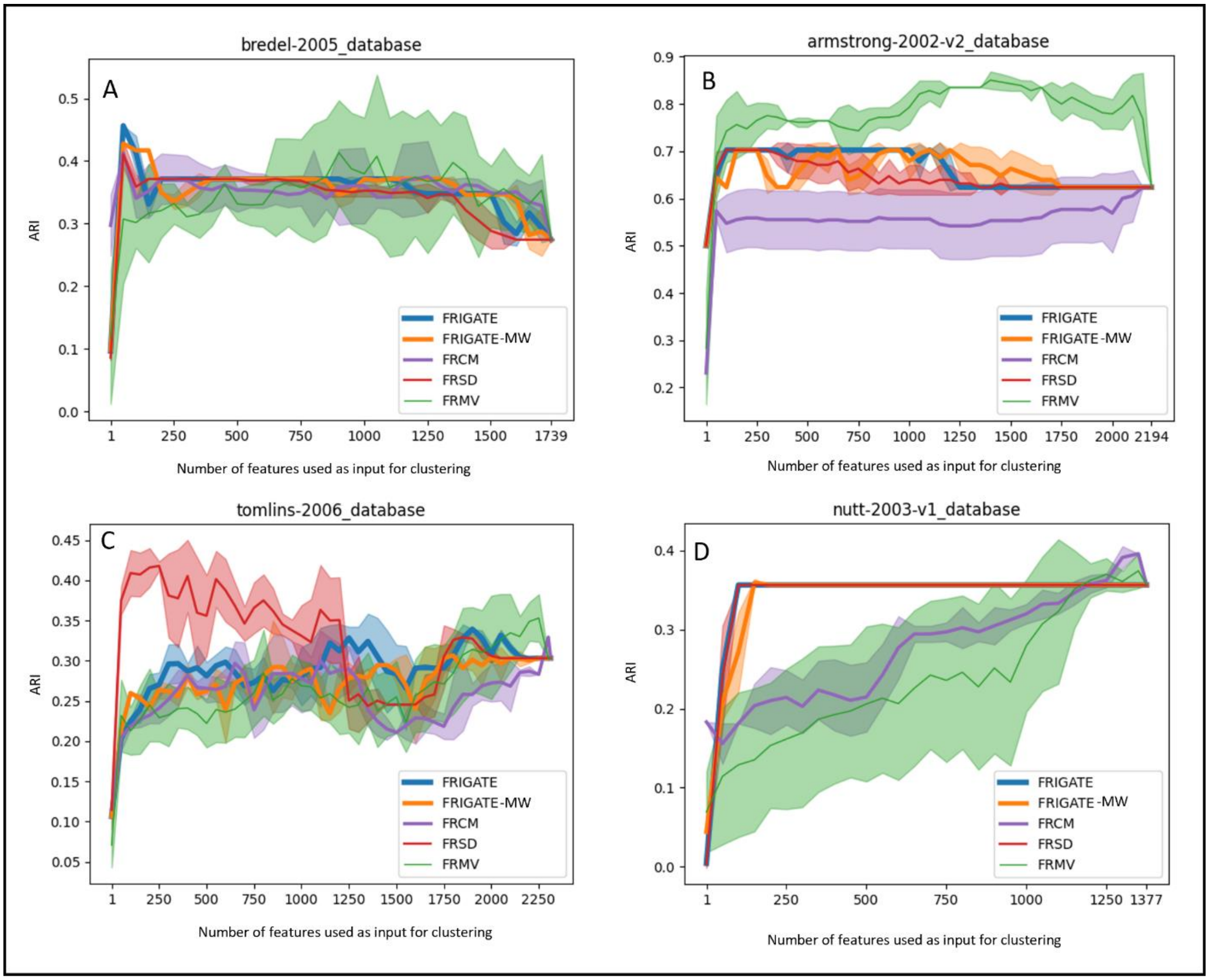


Figure 3. Performance of the tested algorithms on genomic datasets. The ranking produced by each algorithm was used to cluster the data with a growing number of features. The Y axis is the ARI score compared to the known clustering. The results are average of ten runs. The light-colored sleeve around each plot is ±1 std. A-D for datasets 1-4 in Table 2, respectively.

We created three EMR datasets from the MIMIC-III repository [4], [5] and three from the eICU repository [40], [41], both downloaded from PhysioNet [3] (datasets 5-7, 9-11 in Table 2). The input features used were continuous, containing lab tests ("labs"), age and length of stay in the hospital (days in MIMIC and minutes in eICU) and Apache score in eICU. For each lab, we included only the first measurement that was available for the patient during the ICU stay. For each patient we included data from a single ICU stay. For the MIMIC datasets ICD-9 diagnosis codes were extracted per ICU stay and used for labeling the patients. For the eICU datasets, diagnoses and Apache score parameters were used as categorical variables and for labeling. Labs that were missing in >70% of the cohort were removed. To remove potential outliers, we z-scored each continuous measurement across the cohort, and removed patients that had any lab with $|z - score| \geq 3$. We then applied the Iterative Imputer as implemented in [31] to the raw data to complete missing data and performed z-score normalization. The MIMIC cohorts that we constructed were:

1. Dataset 5 – patients that had a cancer ICD-9 diagnosis, aged 18-40. The data were divided into two clusters by length of stay: 122 patients who were discharged alive and spent less than 18 days in ICU, and 39 patients who either died during the ICU stay or stayed 18 days or more at the ICU. 70 features were recorded.
2. Dataset 6 – "healthy" patients: individuals aged 20-30 who did not have ICD-9 diagnosis of cancer, benign tumors, hypertension, cardiac disease, endocrine related disease, or hepatitis and stayed up to one day at ICU. They were divided into two clusters by sex: 84 males and 26 females. Here 47 features were recorded.
3. Datasets 7 – Newborns divided into two clusters: 1534 with jaundice and 3752 without jaundice, with 29 features.

The results on these datasets are shown in Figure 4A-C and summarized in Table 3. For Dataset 5 (Figure 4A), when using up to 50% of the ranked features FRIGATE performance was best.

With over 50% of features FRCM results were comparable. For Dataset 6 (Figure 4B) FRCM was best followed by FRIGATE. FRMV performed comparably to FRIGATE and FRSD performed worst. For Dataset 7 (Figure 4C) with up to 50% of features FRCM performed best. With 50% or more of the ranked features the results of FRIGATE and FRMV were comparable to FRCM or better. FRSD was the worst performer.

Dataset 8 consists of heart failure patients from Zigong Fourth People's Hospital [38], [39], also extracted from PhysioNet. This cohort was divided into two age groups: 68 patients of ages 29-49 and 101 patients of ages 89-100. We had 77 features in this cohort after removing features with >30% missing data, and used the Iterative Imputer for missing data. The results are shown in Figure 4D and Table 3. Here FRSD performed comparably to FRIGATE and even slightly better in some thresholds, with FRMV and FRCM performed much worse, with especially poor results in the first 40% of features. A full comparison among the results is found in Supplementary 6.

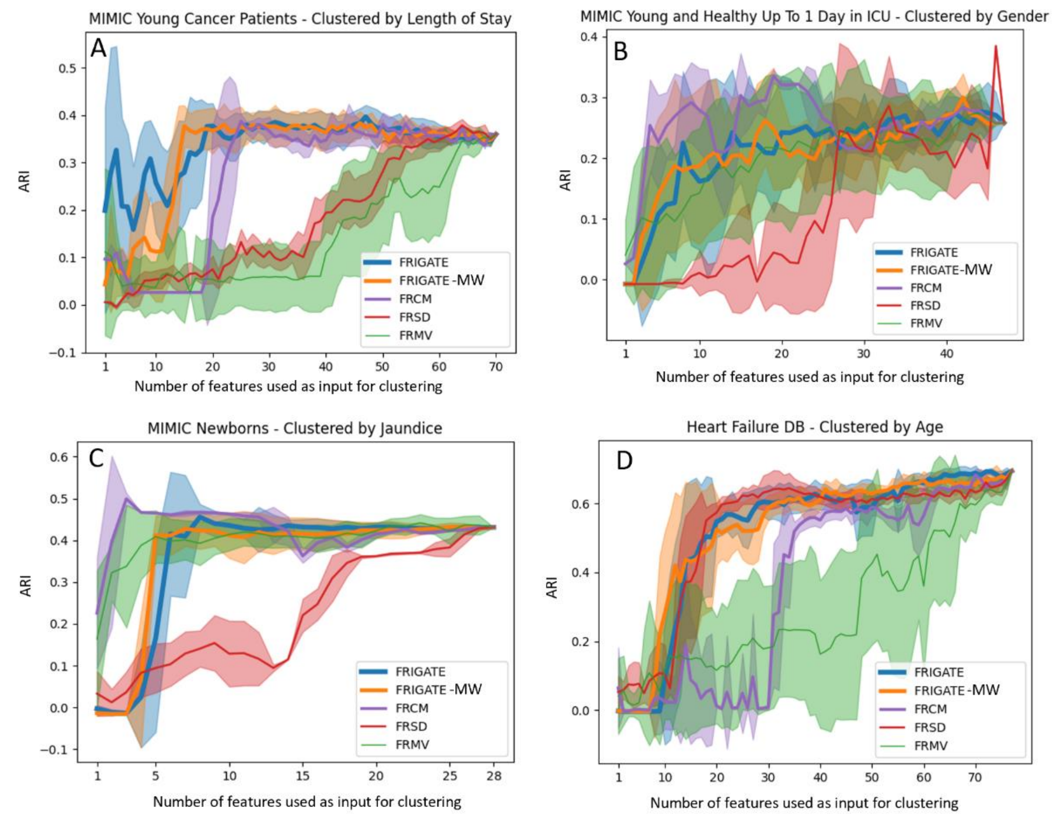


Figure 4. A-D: Performance on datasets 5-8 respectively. See Figure 3 for caption details.

Algorithm	Dataset 5		Dataset 6		Dataset 7		Dataset 8	
	ARI of top $\lceil 25\% \rceil$ features	ARI of top $\lceil 50\% \rceil$ features	ARI of top $\lceil 25\% \rceil$ features	ARI of top $\lceil 50\% \rceil$ features	ARI of top $\lceil 25\% \rceil$ features	ARI of top $\lceil 50\% \rceil$ features	ARI of top $\lceil 25\% \rceil$ features	ARI of top $\lceil 50\% \rceil$ features
FRIGATE	0.328 ± 0.105	0.372 ± 0.029	0.182 ± 0.149	0.237 ± 0.092	0.409 ± 0.146	$\mathbf{0.435 \pm 0.014}$	0.547 ± 0.045	$\mathbf{0.627 \pm 0.042}$
FRIGATE-MW	$\mathbf{0.37 \pm 0.042}$	$\mathbf{0.378 \pm 0.031}$	$\mathbf{0.21 \pm 0.119}$	0.198 ± 0.038	0.427 ± 0.042	0.417 ± 0.038	0.524 ± 0.103	0.601 ± 0.035
FRMV	$0.053 \pm 0.082^{*\dagger}$	$0.053 \pm 0.053^{*\dagger}$	0.181 ± 0.116	0.214 ± 0.104	0.401 ± 0.047	$0.41 \pm 0.027^{*}$	$0.116 \pm 0.154^{*\dagger}$	$0.221 \pm 0.233^{*\dagger}$
FRSD	$0.058 \pm 0.013^{*\dagger}$	$0.107 \pm 0.049^{*\dagger}$	$0.006 \pm 0.022^{*\dagger}$	$0.097 \pm 0.149^{*}$	$0.129 \pm 0.053^{*\dagger}$	$0.115 \pm 0.0^{*\dagger}$	$\mathbf{0.573 \pm 0.029}$	0.625 ± 0.025
FRCM	$0.026 \pm 0.0^{*\dagger}$	$0.35 \pm 0.027^{\dagger}$	$\mathbf{0.21 \pm 0.068}$	$\mathbf{0.291 \pm 0.054}^{\dagger}$	$\mathbf{0.465 \pm 0.007}$	0.422 ± 0.034	$0.012 \pm 0.004^{*\dagger}$	$0.559 \pm 0.024^{*\dagger}$

Table 3. Performance on Dataset 5-8. In **bold** are the top performers.

* - significant difference from FRIGATE, † - significant difference from FRIGATE-MW

The eICU cohorts that we constructed included Caucasian patients admitted directly to ICU with sex labels:

1. Dataset 9 – intubated patients aged 70 and above were divided according to status at discharge of "Alive", 305 patients, and "Expired", 136 patients. 87 continuous and 70 categorical features that had a value in at least 1% of the cohort were used.
2. Dataset 10 – patients who stayed up to one day in ICU, separated by age groups: 487 patients aged 18 to 80, and 83 patients aged 80 or older. 59 continuous and 20 categorical features that had a value in at least 5% of the cohort were used.
3. Dataset 11 – patients aged 18-30 separated by length of stay: 138 who stayed over 4.5 days (>6500 minutes) or expired, and 94 who stayed 4.5 days or less and were discharged alive. 72 continuous and 14 categorical features that had a value in at least 5% of the cohort were used.

The results for the eICU datasets are shown in Figure 5. Figures 5A, 5C, 5E compare all algorithms using the continuous features only. The same trends are observed – both versions of FRIAGTE and FRCM perform best, FRMV has a large variance in results and FRSD performs poorly.

We next used these datasets to test the ability to improve the results by adding categorical features. We tested different values of γ and looked for a change in the ARI of the full set of features in comparison to only using the continuous features (results not shown). A change in ARI means a different composition of the clusters caused by the categorical features. For $\gamma < 5$ in most cases there was no change in the composition of the clusters, and $\gamma > 6$ lead to a major decrease in ARI. We therefore chose $\gamma = 6$ in all cases. In most datasets we do not see an improvement, and in some cases more features were needed to reach high values of ARI. Overall, the categorical features did not improve the solution. Interestingly, in dataset 10 (Figure 5D) adding the categorical variables harmed the performance of FRIGATE-MW more than that of FRIAGATE.

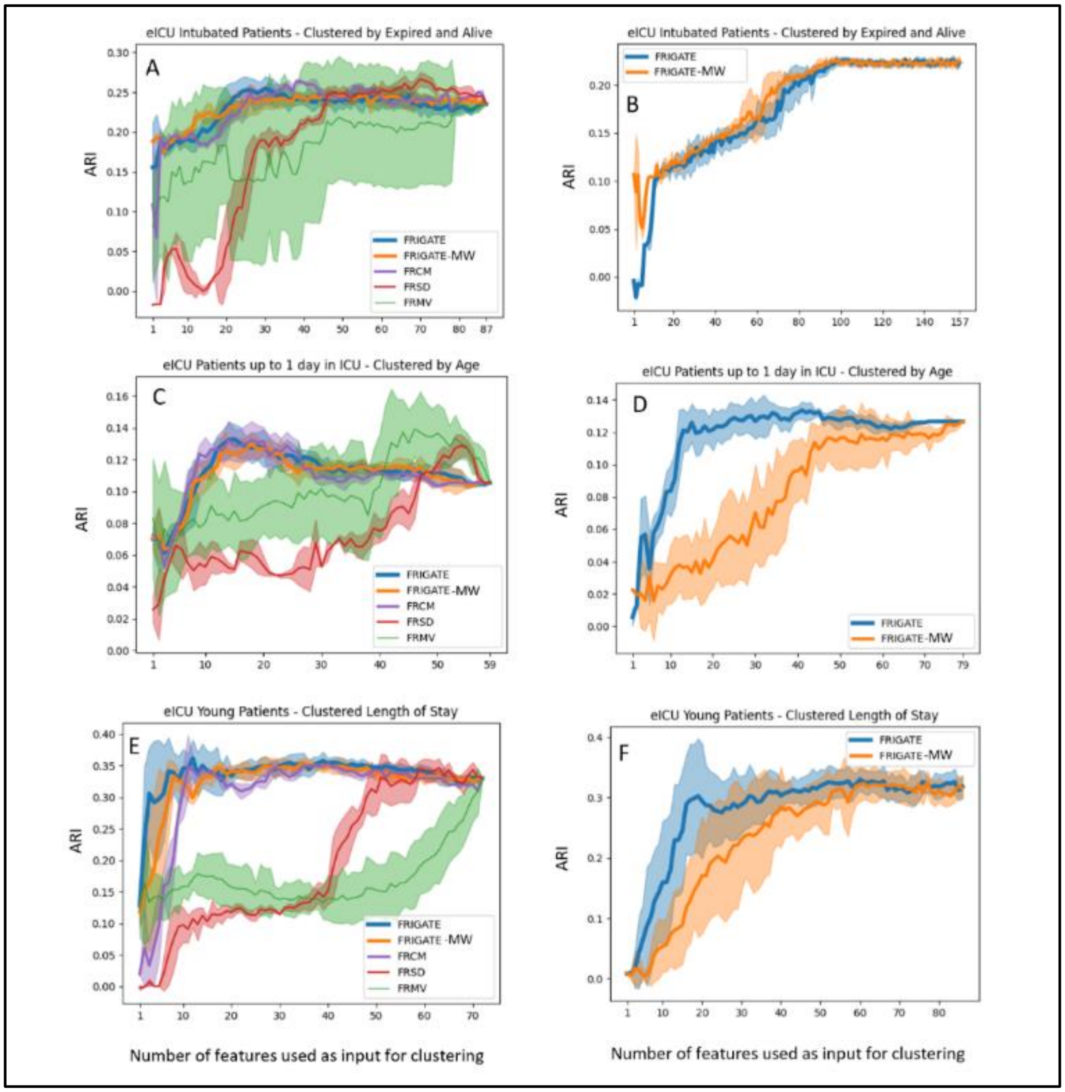


Figure 5. Performance on datasets 9-12 from the eICU repository. See Figure 3 captions for details. A-B: dataset 9, C-D: dataset 10, E-F: dataset 11. A, C, E show results when using only the continuous features of the datasets, and B, D, F show results for the mixed data.

In Table 4 we show the weighted rank (WR) and weighted ARI ($WARI$) scores of all algorithms for datasets 1-11. Apart from dataset 7 with the *WARI*, a variant of FRIGTAE is among the top two algorithms in all cases. In terms of *WR*, FRIGATE was best in the 4 cases and second in 4, and FRIGATE-MW was best in one and second in 6. In terms of WARI, FRIGATE was best in 4, second in 2, FRIGATE-MW best in 2 and second in 5 cases. FRCM was best in 3 and second in one case for both measures.

Dataset	1	2	3	4*	5	6*	7*	8	9	10	11
Weighted rank											
FRIGATE	**0.727**	0.662	0.678	**0.748**	**0.881**	0.613	0.657	0.649	0.668	0.667	**0.933**
FRIGATE-MW	0.699	0.582	0.565	0.735	0.807	0.606	0.522	0.682	**0.794**	0.751	0.752
FRMV	0.430	**0.951**	0.345	0.245	0.334	0.527	0.596	0.411	0.301	0.517	0.383
FRCM	0.514	0.195	0.468	0.429	0.519	**0.854**	**0.825**	0.358	0.789	**0.782**	0.625
FRSD	0.544	0.541	**0.879**	0.721	0.405	0.255	0.297	**0.800**	0.414	0.226	0.266
Weighted ARI											
FRIGATE	**0.347**	0.659	0.268	**0.312**	**0.314**	0.175	0.287	0.411	0.214	0.103	**0.327**
FRIGATE-MW	**0.347**	0.646	0.255	0.307	0.277	0.178	0.296	0.424	**0.223**	0.106	0.308
FRMV	0.321	**0.737**	0.236	0.179	0.085	0.166	0.367	0.185	0.161	0.089	0.152
FRCM	0.346	0.529	0.245	0.227	0.201	**0.236**	**0.412**	0.218	0.216	**0.108**	0.279
FRSD	0.339	0.641	**0.334**	0.311	0.105	0.058	0.143	**0.432**	0.126	0.058	0.128

Figure 4. Weighted rank and weighted ARI for the tested algorithms in datasets 1-11. In **bold** is the top performer for the dataset, underlined is the second best. * - Datasets where the top two performing algorithms were different for the two evaluation metrics.

4.2 Clinical Significance – Test Case

We wished to evaluate the clinical relevance of the leading chosen features to the target labels. We chose to focus on Dataset 6 as there is evidence for sex-based differences in lab tests [42]. We chose the twelve features that were available in both cohorts and according to [42] fulfil:

$$\frac{abs(x_{male} - x_{female})}{\max\,(x_{male}, x_{female})} \geq 0.1 \qquad (9)$$

where x_i is the mean value of feature x for sex i [42]. We call these the top features. A ranked list of all features according to FRIGATE and FRIGATE-MW and the top features are in Supplementary 7.

We performed a hypergeometric test between the 12 top ranked features according to FRIGATE and the top features from [42], and similarly for FRIGATE-MW. For FRIGATE-MW, six of the top ranked features were also top features in [42] giving a significant p-value of 0.034. For FRIGATE, five of the top twelve features were common with the top features of [42], which accounts to a non-significant p-value of 0.136.

We also calculated the p-value of the minimum hypergeometric score (mHG), as used in the DRIM algorithm [43], for calculating the significance without determining in advance the threshold for the hypergeometric test and accounting for multiple testing. For FRIGATE the mHG was obtained for 13 features, with p-value of 0.07. For FRIGATE-MW the threshold was 10 features with p-value of 0.01.

It is important to remember that [42] refers to seemingly healthy individuals, while Dataset 6 comprised of patients who spent in the ICU for up to one day, and some stayed overnight. That means that although the patients were young and did not require a major intervention, they still suffered from some medical condition. Indeed, the top feature in both versions of FRIGATE was “days in hospital” (more females stayed overnight, details not shown), which might suggest some correlation between the clusters and the medical condition, together with the correlation with sex.

4.3 Runtime comparison

Table 5 shows the runtimes on Databases 1-8 for the tested algorithms. The FRIGATE variants are slower on the genomic Datasets 1-4, which have many features and a few samples, but fast on the EMR datasets, which have less features. FRCM runs faster on Datasets 1-4, but when the number of samples grows its runtime increases sharply (Database 7).

The behavior of FRIGATE can be explained by the choice to set the number iterations depending on the number of features. However, this is a tunable parameter with a trade off with f, the number of features included per iteration (see Supplementary 2). FRCM, on the other hand, has a set number of iterations, and produces an mxm matrix for each feature, which is expensive both in runtime and in space. Note also the slowdown of FRSD on Dataset 7, which has thousands of patients.

Dataset no. /Algorithm	1	2	3	4	5	6	7	8
FRIGATE	1967.9 ± 28.2	4739.7 ± 80.2	5309.5 ± 136.3	1336.7 ± 21.4	49.8 ± 0.2	32.5 ± 0.1	286.8 ± 2.7	132.9 ± 1.1
FRIGATE-MW	2854.0 ± 11.9	4903.2 ± 45.6	5736.0 ± 35.084	1898.3 ± 6.4	40.9 ± 0.8	29.2 ± 1.2	296.3 ± 10.5	118.8 ± 1.3
FRCM	500.2 ± 6.2	1188.9 ± 14.3	2457.3 ± 31.4	411.7.2 ± 3.7	229.6 ± 2.5	111.2 ± 1.3	74215.4 ± 938.4	268.6 ± 1.8
FRSD	1159.6 ± 6.0	1898.3 ± 8.2	2808.4 ± 6.2	998.2 ± 11.1	1450.9 ± 12.3	922.8 ± 30.9	39716.9 ± 256.0	1484.0 ± 15.9
FRMV	101.0 ± 0.6	123.7 ± 0.6	133.6 ± 1.2	82.8 ± 0.7	21.6 ± 0.2	19.8 ± 0.3	653.9 ± 3.3	47.3 ± 1.0

Table 5. Runtime in seconds for Datasets 1-8. Results are mean±STD of five runs for Datasets 1-4 and of three runs for Datasets 5-8 (the number of repetitions was reduced as the total runtime was large).

5. Discussion

We presented here FRIGATE, a new ensemble feature ranking algorithm for clustering, aimed for clustering of medical data. To the best of our knowledge, this is the first use of MW within the feature ranking for clustering framework and the first explicit use of Shapley values for unsupervised feature selection. Unlike extant ensemble feature ranking algorithms, FRIGATE incorporates categorical and mixed data features. In tests on simulated and on real EMR datasets FRIGATE was the only algorithm that performed constantly well, and had an acceptable runtime in all cases.

The simulation results revealed interesting behaviors of the tested algorithms. FRSD and FRMV seem to improve, while FRIGATE and FRCM performed worse with higher correlation levels. Intuitively, it should be harder to set apart the informative features from the full set of features when high correlation levels are present. Our hypothesis is that enforcing extreme levels of correlation between all features shaped the data so that the differences between features are better captured by the changes in the silhouette score, which is incorporated in FRSD. This should be further addressed in future research.

When algorithms had accurate recognition rates below the random 0.2, the informative features tended to be recognized as non-informative. Indeed, at the bottom 20 features of FRIGATE on 10 simulation runs with: $k = 4,\ \mu = 2,\ \sigma = 0.05$ and normalized data, $68 \pm 24\%$ of the informative features were in the bottom 20%. This suggests that not only that FRIGATE did not recognize the informative features, but high levels of correlation make the algorithm recognize the informative features as the most non-informative. Although these levels of correlation are unrealistic, the behavior of the algorithm is not fully understood. Further research is needed to understand why the distance to centroids, which is objective function used by FRIGATE, was affected more dramatically for non-informative features when the correlation levels between all features were high.

FRIGATE and FRIGATE-MW had different behavior on simulated and real data. On simulated data, the two algorithms performed comparably, but when a difference was observed it was usually in favor of FRIGATE-MW. This suggests that MW has the potential to improve random selection of features in unsupervised tasks. On real data FRIGATE performed slightly better than FRIGATE-MW. However, although the algorithm was designed to work with mixed data, including categorical features did not improve the results. Future work should evaluate the possible contribution of MW to the ensemble framework, and more specifically, broaden the options for cost functions, which are a key factor in MW.

A limitation of FRIGATE compared to FRSD and FRCM is that the number of clusters k is needed as input. However, when testing different values of k on simulated data, the FRIGATE results were stable even when the input k was much larger than the real k (see Supplementary 8). Future research should test waiving the required input k. FRSD and FRCM are averaging their results over different values of k, but this method is currently not relevant for FRIGATE, as the solution score is affected by the number of clusters, and averaging over different values of k will probably be biased.

Our study has several limitations. We compared FRIGATE to three other algorithms for which code was not available. Their reported performance here is based on our implementation. This is mostly relevant to the runtime comparison. Other implementations may improve runtime for some of the tested algorithms.

A key limitation in the evaluation of EMR data was the validity of the clusters that we produced. Heterogenous cohorts like these of MIMIC and eICU may contain multiple overlapping subgroups, which may confound clustering attempts and their evaluation. Including mixed data where both the categorical and continuous features are relevant, was another challenging task. In our tests, adding the categorical features did improve the results, and in some cases harmed them. Also, all the datasets that we generated were partitioned into two clusters. More analysis is needed on medical datasets with mixed data and a larger number of clusters.

We used the elbow method for choosing k, the number of clusters. In all runs of both simulated and real data, the value $k = 2$ was chosen, even when the real number of clusters was higher. This is in line with a previous report [13]. Although we showed on simulated data that FRIGATE is unaffected by choosing the wrong k, there is a need for a better method to choose k.

6. Data availability

All real data used in this paper are from publicly available sources. See Table 2 for details.

7. Code availability

The code for FRMV, FRCM, and FRSD was not provided by their authors, and we reimplemented them. Their code, as well as the code for FRIGATE and FRIGATE-MW, are available in: https://github.com/Shamir-Lab/FRIGATE

8. 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.

9. Acknowledgements

This study was supported in part by a grant from the Tel Aviv University Center for AI and Data Science (TAD). ES was supported in part by a fellowship from the Edmond J. Safra Center for Bioinformatics at Tel-Aviv University.

10. References

[1] H. Atasoy, B. N. Greenwood, and J. S. Mccullough, “The digitization of patient care: a review of the effects of electronic health records on health care quality and utilization,” *Annual Review of Public Health Annu. Rev. Public Health*, vol. 13, no. 1, pp. 487–500, 2019, doi: 10.1146/annurev-publhealth.

[2] B. Fecher, S. Friesike, and M. Hebing, “What drives academic data sharing?,” *PLoS One*, vol. 10, no. 2, Feb. 2015, doi: 10.1371/journal.pone.0118053.

[3] A. Goldberger *et al.*, “PhysioBank, PhysioToolkit, and PhysioNet: Components of a new research resource for complex physiologic signals,” *Circulation [Online]*, vol. 101, no. 23, pp. e215–e220, 2000.

[4] A. Johnson, T. Pollard, and R. Mark, “ MIMIC-III Clinical Database (version 1.4),” PhysioNet.

[5] A. E. W. Johnson *et al.*, “MIMIC-III, a freely accessible critical care database,” *Sci Data*, vol. 3, May 2016, doi: 10.1038/sdata.2016.35.

[6] C. Sudlow *et al.*, “UK Biobank: an open access resource for identifying the causes of a wide range of complex diseases of middle and old age,” *PLoS Med*, vol. 12, no. 3, Mar. 2015, doi: 10.1371/journal.pmed.1001779.

[7] Y. Luo, “Evaluating the state of the art in missing data imputation for clinical data,” *Brief Bioinform*, vol. 23, no. 1, Jan. 2022, doi: 10.1093/bib/bbab489.

[8] A. Garg and V. Mago, “Role of machine learning in medical research: A survey,” *Computer Science Review*, vol. 40. Elsevier Ireland Ltd, May 01, 2021. doi: 10.1016/j.cosrev.2021.100370.

[9] M. M. Papathanasiou, M. Onel, I. Nascu, and E. N. Pistikopoulos, “Chapter 6 - Computational tools in the assistance of personalized healthcare,” *Computer Aided Chemical Engineering*, vol. 42, pp. 139–206, 2018.

[10] S. Khanmohammadi, N. Adibeig, and S. Shanehbandy, "An improved overlapping k-means clustering method for medical applications," *Expert Syst Appl*, vol. 67, pp. 12–18, Jan. 2017, doi: 10.1016/j.eswa.2016.09.025.

[11] Y. Wang *et al.*, "Unsupervised machine learning for the discovery of latent disease clusters and patient subgroups using electronic health records," *J Biomed Inform*, vol. 102, Feb. 2020, doi: 10.1016/j.jbi.2019.103364.

[12] G. Tosto, S. E. Monsell, S. E. Hawes, G. Bruno, and R. Mayeux, "Progression of extrapyramidal signs in Alzheimer's disease: clinical and neuropathological correlates," *Journal of Alzheimer's Disease*, vol. 49, no. 4, pp. 1085–1093, Jan. 2016, doi: 10.3233/JAD-150244.

[13] E. Shpigelman *et al.*, "Clustering of clinical and echocardiographic phenotypes of covid-19 patients," *Sci Rep*, vol. 13, no. 1, p. 8832, Dec. 2023, doi: 10.1038/s41598-023-35449-1.

[14] S. Solorio-Fernández, J. A. Carrasco-Ochoa, and J. F. Martínez-Trinidad, "A review of unsupervised feature selection methods," *Artif Intell Rev*, vol. 53, no. 2, pp. 907–948, Feb. 2020, doi: 10.1007/s10462-019-09682-y.

[15] K. Y. Yeung and W. L. Ruzzo, "Principal component analysis for clustering gene expression data," *Bioinformatics*, vol. 17, no. 9, pp. 763–774, 2001.

[16] J. G. Dy and C. E. Brodley, "Feature selection for unsupervised learning," *Journal of Machine Learning Research*, vol. 5, pp. 845–889, 2004.

[17] J. Yu, H. Zhong, and S. B. Kim, "An ensemble feature ranking algorithm for clustering analysis," *J Classif*, vol. 37, no. 2, pp. 462–489, Jul. 2020, doi: 10.1007/s00357-019-09330-8.

[18] Y. Hong, S. Kwong, Y. Chang, and Q. Ren, "Consensus unsupervised feature ranking from multiple views," *Pattern Recognit Lett*, vol. 29, no. 5, pp. 595–602, Apr. 2008, doi: 10.1016/j.patrec.2007.11.012.

[19] S. Zhang, H. S. Wong, Y. Shen, and D. Xie, "A new unsupervised feature ranking method for gene expression data based on consensus affinity," *IEEE/ACM Trans Comput Biol Bioinform*, vol. 9, no. 4, pp. 1257–1263, 2012, doi: 10.1109/TCBB.2012.34.

[20] D. Guan, W. Yuan, Y. K. Lee, K. Najeebullah, and M. K. Rasel, "A review of ensemble learning based feature selection," *IETE Technical Review (Institution of Electronics and Telecommunication Engineers, India)*, vol. 31, no. 3. Medknow Publications, pp. 190–198, 2014. doi: 10.1080/02564602.2014.906859.

[21] Shapley Loid S., "A value for n-person games," *Contributions to the Theory of Games*, pp. 307–317, 1953.

[22] S. Mukund and A. Najmi, "The many Shapley values for model explanation," *International conference on machine learning*, 2020.

[23] S. Arora, E. Hazan, and S. Kale, "The multiplicative weights update method: a meta-algorithm and applications," *Theory of Computing*, vol. 8, no. 1, pp. 121–164, 2012, doi: 10.4086/toc.2012.v008a006.

[24] J. McQueen, "Some methods for classification and analysis of multivariate observations," *Proc. 5th Berkeley Symp. Math. Statist. Prob.*, vol. 1, pp. 281–297, 1967.

[25] Z. Huang, "A fast clustering algorithm to cluster very large categorical data sets in data mining," *Dmdk*, vol. 3, no. 8, pp. 34–39, May 1997.

[26] N. J. de Vos, "kmodes categorical clustering library." Accessed: Jul. 10, 2023. [Online]. Available: https://github.com/nicodv/kmodes

[27] Z. Huang, "Extensions to the k-means algorithm for clustering large data sets with categorical values," *Data Min Knowl Discov*, vol. 12, pp. 283–304, 1998, doi: https://doi.org/10.1023/A:1009769707641.

[28] G. Preud'homme *et al.*, "Head-to-head comparison of clustering methods for heterogeneous data: a simulation-driven benchmark," *Sci Rep*, vol. 11, no. 1, Dec. 2021, doi: 10.1038/s41598-021-83340-8.

[29] L. Hubert and P. Arabie, "Comparing Partitions," *Journal of Classification 2, 193–218 ().*, vol. 2, pp. 193–218, Dec. 1985.

[30] P. J. Rousseeuw, "Silhouettes: a graphical aid to the interpretation and validation of cluster analysis," *J Comput Appl Math*, vol. 20, pp. 53–65, 1987.

[31] F. Pedregosa, V. Michel, and O. Grisel, "Scikit-learn: machine learning in python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[32] N. Rappoport and R. Shamir, "Multi-omic and multi-view clustering algorithms: review and cancer benchmark," *Nucleic Acids Res*, vol. 46, no. 20, pp. 10546–10562, Nov. 2018, doi: 10.1093/nar/gky889.

[33] S. A. Armstrong *et al.*, "MLL translocations specify a distinct gene expression profile that distinguishes a unique leukemia," *Nat Genet*, vol. 30, no. 1, pp. 41–47, 2002, doi: 10.1038/ng765.

[34] M. Bredel *et al.*, "Functional network analysis reveals extended gliomagenesis pathway maps and three novel MYC-interacting genes in human gliomas," *Cancer Res*, vol. 65, no. 19, pp. 8679–8689, Oct. 2005, doi: 10.1158/0008-5472.CAN-05-1204.

[35] S. A. Tomlins *et al.*, "Integrative molecular concept modeling of prostate cancer progression," *Nat Genet*, vol. 39, no. 1, pp. 41–51, Jan. 2007, doi: 10.1038/ng1935.

[36] C. L. Nutt *et al.*, "Gene expression-based classification of malignant gliomas correlates better with survival than histological classification," *Cancer Res*, vol. 63, no. 7, pp. 1602–1607, 2003.

[37] M. C. P. de Souto, I. G. Costa, D. S. A. de Araujo, T. B. Ludermir, and A. Schliep, "Clustering cancer gene expression data: A comparative study," *BMC Bioinformatics*, vol. 9, Nov. 2008, doi: 10.1186/1471-2105-9-497.

[38] Z. Zhang *et al.*, "Hospitalized patients with heart failure: integrating electronic healthcare records and external outcome data (version 1.3)," PhysioNet.

[39] Z. Zhang *et al.*, "Electronic healthcare records and external outcome data for hospitalized patients with heart failure," *Sci Data*, vol. 8, no. 1, Dec. 2021, doi: 10.1038/s41597-021-00835-9.

[40] T. Pollard, A. Johnson, J. Raffa, L. A. Celi, O. Badawi, and R. Mark, "eICU collaborative research database (version 2.0)," *PhysioNet*, 2019, doi: https://doi.org/10.13026/C2WM1R.

[41] T. J. Pollard, A. E. W. Johnson, J. D. Raffa, L. A. Celi, R. G. Mark, and O. Badawi, "The eICU collaborative research database, a freely available multi-center database for critical care research," *Sci Data*, vol. 5, Sep. 2018, doi: 10.1038/sdata.2018.178.

[42] N. M. Cohen *et al.*, "Personalized lab test models to quantify disease potentials in healthy individuals," *Nat Med*, vol. 27, no. 9, pp. 1582–1591, Sep. 2021, doi: 10.1038/s41591-021-01468-6.

[43] E. Eden, D. Lipson, S. Yogev, and Z. Yakhini, "Discovering motifs in ranked lists of DNA sequences," *PLoS Comput Biol*, vol. 3, no. 3, pp. 0508–0522, 2007, doi: 10.1371/journal.pcbi.0030039.