Contents lists available at ScienceDirect

Computational and Structural Biotechnology Journal

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

Research article

3D clustering of gene expression data from systemic autoinflammatory diseases using self-organizing maps (Clust3D)

Orestis D. Papagiannopoulos [a], Vasileios C. Pezoulas [a], Costas Papaloukas [a,b,c], Dimitrios I. Fotiadis [a,c,*]

[a] *Unit of Medical Technology and Intelligent Information Systems, Dept. of Materials Science and Engineering, University of Ioannina, Ioannina GR45110, Greece*
[b] *Dept. of Biological Applications and Technology, University of Ioannina, Ioannina GR45110, Greece*
[c] *Institute of Biomedical Research, FORTH (Foundation for Research & Technology), Ioannina GR45110, Greece*

ARTICLE INFO

Keywords:
Systemic autoinflammatory diseases
Self-organizing maps
Gene expression
Timeseries
Classification

ABSTRACT

Background and objective: Systemic autoinflammatory diseases (SAIDs) are characterized by widespread inflammation, but for most of them there is a lack of specific biomarkers for accurate diagnosis. Although a number of machine learning algorithms have been used to analyze SAID datasets, aiding in the discovery of novel biomarkers, there is a growing recognition of the importance of SAID timeseries clustering, as it can capture the temporal dynamics of gene expression patterns.
Methodology: This paper proposes a novel clustering methodology to efficiently associate three-dimensional data. The algorithm utilizes competitive learning to create a self-organizing neural network and adjust neuron positions in time-dependent and high dimensional feature space in order to assign them as clustering centers. The quantitative evaluation of the clustering was based on well-known clustering indices. Furthermore, a differential expression analysis and classification pipeline was employed to assess the capability of the proposed methodology to extract more accurate pathway-specific genes from its clusters. For that, a comparative analysis was also conducted against a heuristic timeseries clustering method.
Results: The proposed methodology achieved better overall clustering indices scores and classification metrics using genes derived from its clusters. Notable cases include a threefold increase in the Calinski-Harabasz clustering index, a twofold improvement in the Davies–Bouldin clustering index and a $\sim$ 60% increase in the classification specificity score.
Conclusion: A novel clustering methodology was developed and applied on several gene expression timeseries datasets from systemic autoinflammatory diseases, and its ability to efficiently produce well separated clusters compared to existing heuristic methods was demonstrated.

1. Introduction

Systemic autoinflammatory diseases (SAIDs) refer to a collection of uncommon disorders that can affect individuals of any age. These conditions are characterized by widespread inflammation [1]. The physical symptoms of SAIDs primarily involve fever, skin rashes, joint pain, and swelling. The dysregulation of the innate immune system, often caused by genetic mutations, plays a significant role in these disorders [2,3]. Approximately 40–60% of patients with typical SAID symptoms face challenges in receiving a definitive diagnosis [4]. The diagnosis process usually involves a clinical evaluation, ruling out other potential disorders. Consequently, delays in diagnosis and inadequate treatment decisions are common for individuals with SAID-related conditions. One notable distinction between SAIDs and autoimmune diseases is the lack of specific biomarkers for diagnosing most of the SAIDs. Unlike autoimmune diseases where autoantibodies serve as diagnostic tools, SAIDs currently lack such defining markers. This further adds to the complexity of diagnosing SAIDs and underscores the need for continued research and advancements in the field to improve diagnostic accuracy and facilitate timely interventions for affected individuals.

Various computational techniques have been employed for the identification of biomarkers in SAIDs. These techniques leverage the

[*] Corresponding author at: Unit of Medical Technology and Intelligent Information Systems, Dept. of Materials Science and Engineering, University of Ioannina, Ioannina GR45110, Greece.
E-mail address: fotiadis@cc.uoi.gr (D.I. Fotiadis).

https://doi.org/10.1016/j.csbj.2024.05.003
Received 23 February 2024; Received in revised form 2 May 2024; Accepted 2 May 2024
Available online 13 May 2024

power of computational analysis and data processing to identify patterns, relationships, and potential markers that can aid in the diagnosis and understanding of SAIDs. In recent years, omics technologies, such as transcriptomics [5,6] and proteomics [7–11] have contributed to SAID biomarker discovery. These techniques generate large-scale molecular data that can be analyzed computationally to identify differentially expressed genes, proteins, or metabolites associated with SAIDs. Machine learning (ML) algorithms have also been utilized to analyze different SAID datasets. These computational methods can process large and complex datasets, extracting meaningful patterns and relationships that may contribute to the understanding and diagnosis of SAIDs. An important application of ML in SAID biomarker identification is feature selection [12,13]. With high-dimensional datasets, including genomic or omics data, it is crucial to identify the most informative features or variables associated with SAIDs. Such algorithms can evaluate the relevance and contribution of each feature and select a subset of features that are most discriminatory between SAID and non-SAID samples. ML can also be applied to identify potential novel biomarkers by leveraging unsupervised learning techniques [14,15]. These methods, such as clustering or dimensionality reduction algorithms like principal component analysis, can group samples based on shared characteristics or identify hidden patterns within the data. By exploring the resulting clusters or patterns, previously unrecognized subtypes, or molecular signatures of SAIDs can be uncovered, potentially leading to the discovery of new biomarkers.

While clustering techniques have been applied to SAIDs datasets, the research can be greatly improved with timeseries clustering techniques, considering SAID patients' gene expression data across multiple timepoints. SAIDs are dynamic conditions characterized by temporal variations in gene expression patterns, thus capturing the corresponding temporal dynamics is crucial for a comprehensive understanding of the disease progression and treatment response. Timeseries clustering could analyze gene expression profiles over time and identify distinct clusters which reflect different stages or responses within SAID patients. The existing literature offers a scarce selection of publicly available tools for clustering gene expression timeseries data. One such tool is TimeClust [16], which focuses on clustering genes according to their temporal expression profiles. Additionally, Clust [17] is a gene clustering software that aims to extract optimal co-expressed gene sets. However, these tools are primarily designed for gene-level clustering and do not specifically address the clustering of multiple patients utilizing the different timepoints. A heuristic approach has been employed in [14] to cluster timeseries gene expression data. The authors proposed a two-step methodology involving the utilization of a two-phase self-organizing map (SOM) to cluster timeseries data, considering each timepoint independently. However, we contend that this sequential application of SOMs across different timepoints introduces significant biases that may compromise the accuracy and comprehensiveness of the clustering results. Consequently, our objective is to establish a robust and unbiased methodology that surpasses these limitations. To this end, we have developed an innovative algorithm that capitalizes on the integration of the temporal dimension, aiming to overcome the aforementioned biases and enhance the overall clustering process.

Here, we present a novel algorithm designed for clustering 3D data (Clust3D), such as SAIDs gene expression timeseries. Our algorithm enables the clustering of patients based on their gene expression profiles at different timepoints. By grouping patients according to the similarity of their gene profile changes over time, this methodology provides a robust foundation for exploratory analysis. Notably, to the best of our knowledge, there is currently no publicly available tool or clustering algorithm specifically tailored for the simultaneous clustering of SAID patients across multiple time points. This work fills this gap by employing a self-adjusting neural network approach to effectively cluster SAID patients with time-related gene expressions. The aim of this study is to provide a novel computational framework that addresses the three-dimensional matrix structure and compare its clustering capabilities with the existing literature. This computational framework differentiates Clust3D from other time-addressing methods, like spatial-temporal clustering [18], where the spatial and temporal information is incorporated in the same two-dimensional matrix [19]. Hence, the distinguishing feature of this clustering methodology lies in its capability for efficient multi-timepoint and multi-dimensional clustering.

The subsequent sections first describe the design and implementation of the Clust3D algorithm. Then, a comparative analysis is performed between Clust3D, and the heuristic approach employed in [14]. The evaluation of the clustering outcomes from both methodologies is segregated into two distinct sections. Firstly, the clustering outputs are assessed by calculating various clustering indices, enabling a quantitative assessment. Secondly, a comprehensive workflow for differential expression (DE) analysis is employed to extract potential pathway-specific genes from the clusters. Additionally, a subsequent binary classification task is undertaken, to distinguish between disease and healthy samples. By leveraging this pipeline, the clustering performance of each methodology is evaluated based on classification metrics, employing the genes extracted from their respective clustermaps as classification features. Finally, in the last section, some prominent findings and remarks are discussed regarding the quantitative and qualitative analysis of the clustermaps of each methodology.

2. Materials and methods

2.1. Timeseries gene expression data

Clust3D was tested on three publicly available timeseries datasets from the Gene Expression Omnibus (GEO) [20], containing gene expression data from systemic autoinflammatory disease (SAID) patients. Specifically, GSE80060 [21] provides gene expression data of the whole blood of systemic juvenile idiopathic arthritis (SJIA) patients treated with canakinumab, or placebo and age matched healthy samples (148 disease samples and 22 healthy samples). This dataset contains two different time points corresponding to pre and post treatment. GSE97075 [22] provides gene expression data of hyperimmunoglobulin D syndrome (HIDS) patients with periodic fever syndrome treated with canakinumab (30 disease samples and 15 healthy samples). This dataset contains six time points corresponding to different treatment stages. Finally, GSE9863 [23] provides gene expression data for Kawasaki patients (60 disease samples and no healthy ones). It contains three different time points corresponding to different stages of the disease. To our knowledge, these are the only publicly available SAID timeseries datasets during the validation process of the proposed method (November 2023).

Clust3D works for timeseries data. As such, only the disease samples from each dataset are being clustered and not the healthy samples, since the datasets don't include temporal information about the healthy samples. Therefore, 148 samples (74 patients) and 54,675 genes were considered from GSE80060, 30 samples (5 patients) and 47,323 genes from GSE97075 and 60 samples (20 patients) and 37,632 genes fromGSE9863. Table 1 shows the number of timepoints, patients, GEO Sample Accessions (GSMs), and genes for each dataset.

Table 1
Number of timepoints, patients, GSMs, and genes eligible for clustering for each dataset.

Dataset	No. of timepoints (T)	No. of patients (P)	No. of GSMs (S)	No. of genes (G)
GSE80060	2	74	148	54,675
GSE97075	6	5	30	47,323
GSE9863	3	20	60	37,632

2.2. Design and implementation

The Clust3D framework is divided into five modules: input files, data preprocessing, neuron initialization, neural network training and clustering. The overall workflow is described below, while Clust3D's modules are shown in Fig. 1. Clust3D is implemented in Python using the external libraries of NumPy [24], pandas [25], scikit-learn [26] and matplotlib [27].

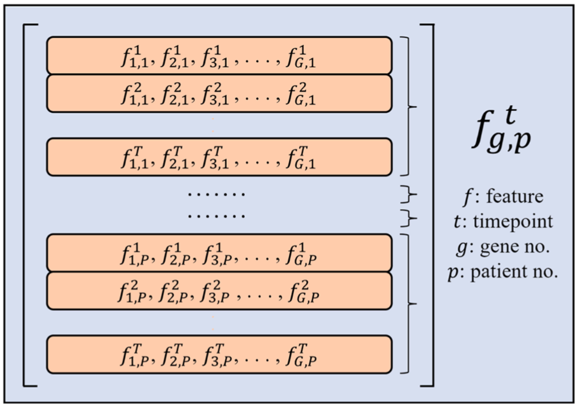
2.2.1. Input files

Two input files have to be provided to the Clust3D algorithm. The first one is the user-processed GEO Series Matrix File, which contains the GSMs at all timepoints. This is a $G \times S$ matrix, where G is the number of genes and S is the number of GSMs. The second one is the user-created correlation file, in which the patient labels and their corresponding GSMs at the different timepoints are specified. This is a $P \times T$ matrix, where P is the number of patients and T the number of timepoints.

2.2.2. Data preprocessing module

Clust3D creates the main 3D data matrix, based on the two previous files. This matrix contains all the information, retaining both the temporal and spatial dimensions for all patients. Contrary to trivial data formatting where each sample is represented as a vector, in Clust3D each sample (patient) is represented as a 2D matrix, with the time dimension (timepoints) being represented as rows and the spatial dimension (genes) as columns. The final size of the matrix is $P \times T \times G$ (Fig. 2), where P is the number of patients, T is the number of timepoints and G is the number of genes.

Following its creation, a default 0 - 1 scale normalization and a user-selected dimensionality reduction method is applied to the 3D matrix for efficient training. In Clust3D a variety of dimensionality reduction techniques is provided, with a Principal Component Analysis (PCA) with two components being the default, for the best noise-reduction and visualization. Additionally, Clust3D offers an automated selection of the number of PCA components using the mathematical elbow rule [28]. More specifically, the explained variance is calculated for a large number of components, and the optimal cut-off number is selected. This is the elbow point at approximately 45° to the x axis, on the normalized PCA explained variance plot. The PCA is performed once on the entirety of the dataset, which has been transformed to an S x G matrix for computing purposes. Due to this matrix containing all the timepoints, each timepoint has its features reduced (G'). This constitutes as a regular PCA application, applying its feature reducing effect on every timepoint. Following the PCA application, the matrix is transformed back to its original 3D shape (P x T x G'), with reduced features. No reduction in the time dimension is performed.



$f_{1,1}^1, f_{2,1}^1, f_{3,1}^1, \dots, f_{G,1}^1$
$f_{1,1}^2, f_{2,1}^2, f_{3,1}^2, \dots, f_{G,1}^2$
$f_{1,1}^T, f_{2,1}^T, f_{3,1}^T, \dots, f_{G,1}^T$
....
....
$f_{1,P}^1, f_{2,P}^1, f_{3,P}^1, \dots, f_{G,P}^1$
$f_{1,P}^2, f_{2,P}^2, f_{3,P}^2, \dots, f_{G,P}^2$
$f_{1,P}^T, f_{2,P}^T, f_{3,P}^T, \dots, f_{G,P}^T$
$f_{g,p}^{t}$
f: feature
t: timepoint
g: gene no.
p: patient no.

Fig. 2. Algebraic representation of the 3D data matrix. Indices t, g and p add up to T, G and P, respectively. Each curly bracket represents a patient.

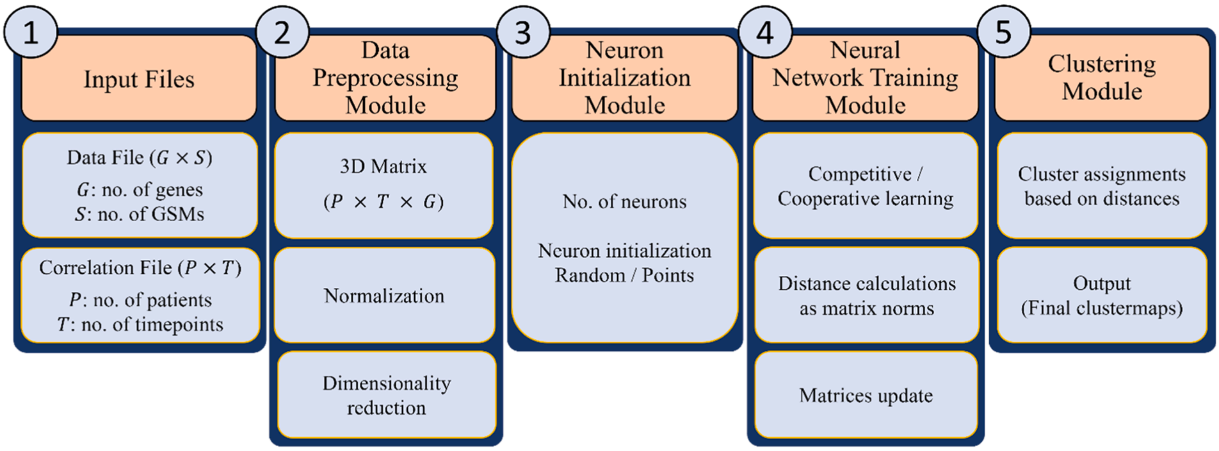
2.2.3. Neuron initialization module

Like in many well-established clustering algorithms, in Clust3D the user can specify the number of neurons (number of clusters) beforehand. Moreover, Clust3D provides the functionality of automatically selecting the optimal number of neurons, based on the elbow rule on the sum of squared error (SSE) plot. This is the algorithm's default behavior.

Self-organizing maps require neuron initialization for their training. Grid-like neuron initialization is not feasible with multiple timepoints, and as such a different approach has to be implemented. Clust3D offers the choice of random initialization of the neurons as 2D matrices, but also a data point-specific initialization. In the latter, it randomly selects existing data points and calculates their average in-between Euclidean distance. The combination with the highest average distance is selected as the chosen data points to initialize the neurons. This way, Clust3D initializes the neurons by utilizing the largest possible span in the time-related, high dimensionality space. The number of different combinations to be calculated can be dictated by the user, with higher numbers resulting in the minimization of the stochasticity and higher consistency.

2.2.4. Neural network training module

Regarding the training of the neural network, competitive learning [29] is employed. The Euclidean distance is first computed between an input sample and all the neurons. Then, the neuron that has the smallest distance to the sample is declared as the best matching unit (BMU) and its weights along with its nearest neighbor neurons (self-organizing) are re-adjusted to closer mimic the input sample (Fig. 3). The novelty here is the introduction of the matrix norms as distance concepts. Conventional distance metrics like the Euclidean, are typically calculated between



1
Input Files
Data File ($G \times S$)
G: no. of genes
S: no. of GSMs
Correlation File ($P \times T$)
P: no. of patients
T: no. of timepoints
2
Data Preprocessing Module
3D Matrix
($P \times T \times G$)
Normalization
Dimensionality reduction
3
Neuron Initialization Module
No. of neurons
Neuron initialization Random / Points
4
Neural Network Training Module
Competitive / Cooperative learning
Distance calculations as matrix norms
Matrices update
5
Clustering Module
Cluster assignments based on distances
Output (Final clustermaps)

Fig. 1. The modules of Clust3D.

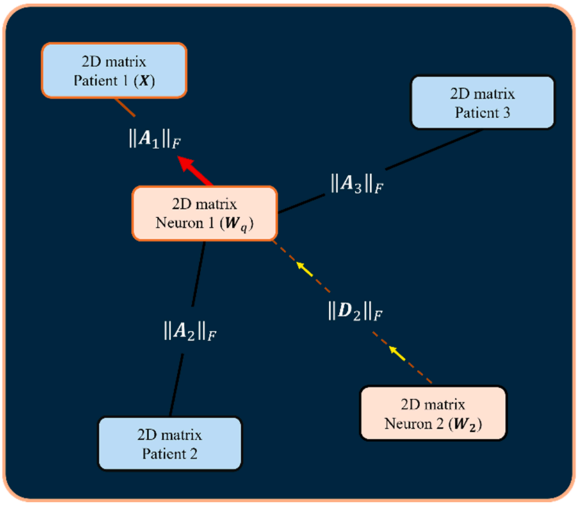


Fig. 3. Simplified example of Clust3D's neuron network training. All distances ($\|\boldsymbol{A}\|_F$, $\|\boldsymbol{D}\|_F$) detailed in 2.2.4 are shown. In this example, for the input sample ($\boldsymbol{X}$), the BMU ($\boldsymbol{W}_q$) is identified and then updated with (1) (red arrow). The remaining neighbor neuron ($\boldsymbol{W}_2$) is updated as well with (1) (yellow arrows). This simplified example refers to only one input sample ($\boldsymbol{X}$) and one iteration. The overall process is repeated for every input sample and every iteration.

vectors. In our case, where the data points are matrices, the distance between two data points is defined as the mathematical norm of the matrix of their differences. As such, Clust3D introduces the capability to train the neural network given the input samples and the neurons as matrices and not just as vectors, containing both the temporal and the spatial information. Thus, clustering can be implemented directly on the patients, given the different timepoints altogether, without the need for per phase clustering, as in [14].

The update function for the neurons is defined as:

$$\boldsymbol{W}_j(i+1) = \quad \boldsymbol{W}_j(i) + \boldsymbol{U}\left(\boldsymbol{W}_j, \boldsymbol{W}_q, \quad i\right) \quad y(i) \quad \left[\boldsymbol{X} - \boldsymbol{W}_q\right], \tag{1}$$

where $\boldsymbol{W}_j(i+1)$ is the matrix of neuron with index j at time $(i+1)$, with i being the current iteration, $\boldsymbol{X}$ is the matrix of the input sample, $\boldsymbol{W}_q$ is the BMU, and y is the learning rate. The BMU is the neuron with the least distance to the input matrix, which is calculated using the Frobenius norm of the matrix difference of the input matrix ($\boldsymbol{X}$) and each neuron ($\boldsymbol{W}_j$):

$$\left\|\boldsymbol{X} - \quad \boldsymbol{W}_j\right\|_F = \quad \|\boldsymbol{A}\|_F = \sqrt{\sum_{k=1}^{m}\sum_{l=1}^{n}|\boldsymbol{A}_{kl}|^2}, \tag{2}$$

where m and n correspond to the dimensions of matrix $\boldsymbol{A}$. The learning rate follows an exponential reduction:

$$y = \quad y_o \exp\left(\frac{-i}{t_1}\right), \tag{3}$$

where y_o is the initial learning rate and t_1 is a user defined constant which controls the exponential decrease of the learning rate. $\boldsymbol{U}\left(\boldsymbol{W}_j, \boldsymbol{W}_q, i\right)$ is the neighborhood function, which dictates the cooperation between neurons. It decreases exponentially and includes a reducing Gaussian distance function [14]:

$$\boldsymbol{U} \quad = \exp\left(\frac{- \quad {d_{jq}}^2}{2 \quad (\sigma_0 \quad \exp\left(\frac{-i\log(\sigma_0)}{t_2}\right))^2}\right), \tag{4}$$

where σ_0 is the standard deviation of the initial Euclidean distances of the randomly initiated neurons, t_2 is a user defined constant which controls the exponential decrease of the neighborhood function and lastly, d_{jq} is the Euclidean distance between a neighbor neuron and the BMU, which is calculated using the Frobenius norm of the neuron matrices difference, as in (2):

$$d_{jq} = \quad \left\|\boldsymbol{W}_j - \boldsymbol{W}_q\right\|_F = \quad \|\boldsymbol{D}\|_F = \sqrt{\sum_{k=1}^{m}\sum_{l=1}^{n}|\boldsymbol{D}_{kl}|^2}, \tag{5}$$

where m and n correspond to the dimensions of matrix $\boldsymbol{D}$, which are the same as matrix $\boldsymbol{A}$.

2.2.5. *Clustering module*

Lastly, after the neurons' algebraic adjustment in the multi-dimensional space, the Euclidean distances between them and the data points are calculated as in (2), to assign the cluster memberships based on the minimum distances. The calculation of Euclidean distances involves computing the straight-line distances between each neuron and the data points in the high-dimensional feature space. This distance metric allows for measuring the proximity or similarity between the neurons and the data points, helping to identify the closest matching neuron for each data point. By assigning cluster memberships based on the minimum distances, the algorithm groups data points together with the neurons that exhibit the least dissimilarity.

3. Experiments

This section is divided into two subsections. *3.1* details the implementation of a comparative method and *3.2* specifies the clustering evaluation metrics. In *3.2* firstly, the clustering indices and their implementations are described (*3.2.1*) and then, the DE and classification analysis is elaborated (3.2.2).

3.1. *Comparative method*

Due to the fact that TimeClust [16] and Clust [17] cluster genes and not patients, while no other algorithm exists in the literature to cluster 3D datasets, a related method had to be applied in order to be compared with Clust3D. Therefore, we utilized the heuristic approach proposed in [14]. The authors devised a two-step technique to cluster timeseries data by applying a two-phase self-organizing map (SOM). In the first phase, they divided a three-timepoint dataset into three single-timepoint datasets and clustered each one using SOMs. This way, they acquired a vector of clustering labels for each patient, with each element corresponding to a particular timepoint. At phase two, they used a final SOM on the $P \times L$ matrix containing the clustering labels, where P is the number of patients and L is the number of timepoints. Hence, they could heuristically cluster patients with multiple timepoint data.

Both the SOMbrero [30] and KMEANS [31] algorithms were chosen to compare the above two-step approach with Clust3D. For the GSE9863 dataset, the final clustermaps of [14] were used as SOMbrero's output, as the approach is exactly the same. For the other two datasets, SOMbrero's initial neuron grid was selected to be large enough (3 × 3) to ensure the automated membership convergence to fewer clusters. The KMEANS algorithm requires the user to predefine the number of clusters. The optimal number is selected based on the optimization of all three clustering indices presented in Section 3.2.1. Subsequently, for every dataset, the best possible clustermap from KMEANS is identified. Prior to clustering and evaluating all datasets and algorithms, a PCA with two principal components was utilized for this study, as it was the default method for feature reduction in [14].

3.2. *Evaluation procedure*

The clustermaps of Clust3D (implementation with the Clust3D algorithm) and the comparative method (implementation with the SOMbrero and KMEANS algorithms) were evaluated through a twofold assessment. Firstly, a comparative analysis was conducted employing the calculation of clustering indices. Secondly, a differential expression (DE) analysis workflow was used to extract potential pathway-specific genes from the three algorithms' clusters. A subsequent binary classification problem was also conducted between disease samples and healthy samples. Using this pipeline, each algorithm's clustering was assessed based on classification metrics, employing as features the genes extracted from their corresponding clusters.

3.2.1. Clustering indices

Traditional clustering indices work by calculating distances between vector data points. In our case, with the introduction of the time dimension, some modifications as to how these metrics are applied on matrices instead of vectors had to be implemented. To be thorough, we opted to study a variety of metrics, based on cluster variance, distance and similarity. Therefore, the Calinski-Harabasz index (CHI) [32], the Davies–Bouldin index (DBI) [33] and the Silhouette score (SS) [34] were used. For the CHI and the DBI, the source code of scikit's implementation [26] of those two indices was used with the appropriate alterations. For the SS, the corresponding mathematical formula was implemented in Python with the appropriate adjustments. Every clustering derived from this study is assessed with these clustering indices.

3.2.1.1. Calinski-Harabasz index. The CHI or the Variance Ratio Criterion is the score defined as the ratio of the sum of the between-cluster dispersion and the within-cluster dispersion. A higher index signifies dense and well-separated clusters. In mathematical notation the CHI is defined as:

$$\frac{\sum_{q=1}^{C} n_q \sum (m_q - \quad M)^2}{\sum_{q=1}^{C} \sum (D_q - \quad m_q)^2} \quad \frac{(P - C)}{\sum_{q=1}^{C} \sum (D_q - \quad m_q)^2 \quad (C-1)}, \tag{6}$$

where C is the number of clusters, P is the total number of data points (patients), q is the cluster index, n_q is the number of data points inside the cluster with index q, M is the overall mean 3D matrix of the whole dataset, m_q is the mean matrix of the cluster with index q and D_q is a 3D matrix with the data belonging to the cluster with index q. This metric can be applied as is in our case, but there is a difference as to how two of the factors are calculated. The first one is the $\sum (m_q - M)^2$ factor, in which the summation is done on all timepoint vectors, instead of just one. The second is the $\sum (D_q - \quad m_q)^2$ factor, in which the m_q matrix is subtracted from all matrices inside D_q instead of just a single subtraction.

3.2.1.2. Davies–Bouldin index. The DBI is defined as the average similarity measure of each cluster with its most similar cluster, with the similarity being the ratio of within-cluster distances, to between-cluster distances. Lower values indicate better clustering. In mathematical notation the DBI is defined as:

$$\frac{1}{C} \sum_{q=1}^{C} \max_{q \neq j} \left(\frac{\sum_{q=1}^{C} \quad \frac{1}{U} \sum_{u}^{U} \left\| W_u - m_q \right\|_F}{\left\| m_q - m_{j \neq q} \right\|_F} \right), \tag{7}$$

where C is the number of clusters, q is the cluster index, U is the number of data points in the cluster with index q, u is the data point index, m_q is the mean matrix of the cluster with index q, W_u is the data point with index u inside the cluster with index q, m_j is the mean matrix of the cluster with index $j \neq q$ and F symbolizes the Frobenius norm. The difference compared to the conventional implementation, is that the calculation of the pairwise distances of data points/centroids ($\left\| W_u - m_q \right\|_F$) and centroids/centroids ($\left\| m_q - m_j \right\|_F$) is based on the Frobenius norm using (2) and (5).

3.2.1.3. Silhouette score. The SS is used to evaluate the quality of clusters by measuring how similar a data point is to its own cluster, compared to the other clusters. The silhouette score ranges from − 1 to 1, where a score of − 1 indicates that the data point is assigned to the wrong cluster, 0 indicates that the data point is on the border between two clusters, and 1 indicates that the data point is well-matched to its own cluster. The SS is calculated for each data point in the dataset and represents the degree of cohesion and separation between clusters. In mathematical notation the SS is defined for every data point as:

$$\frac{b - a}{\max(b, \quad a)}, \tag{8}$$

where a is the mean distance between the data point and all other points in the same cluster and b is the mean distance between the data point and all other points in the next nearest cluster. More specifically, a is defined as:

$$\frac{1}{U} \sum_{u}^{U} \| W - W_u \|_F, \tag{9}$$

where W is the data point, U is the number of data points in the same cluster as W, u is the data point index, W_u is a data point with index u in the same cluster as W and F symbolizes the Frobenius norm. b is defined as:

$$\frac{1}{R} \sum_{r}^{R} \| W - W_r \|_F, \tag{10}$$

where R is the number of data points in the nearest cluster, r is the data point index and W_r is a data point with index r in the nearest cluster. The final SS of the whole clustering is the average SS of all data points. As per the DBI, the difference compared to the conventional implementation, is that the calculation of the pairwise distances of data points is based on the Frobenius norm using (2) and (5).

3.2.2. Differential expression analysis and classification

A pipeline consisting of a DE analysis and a classification step (Fig. 4) was designed to evaluate the robustness of Clust3D and the comparative method (implementation with KMEANS and SOMbrero), regarding their clustering outputs. The aim is to verify the good statistical separation of the derived clusters, by extracting cluster-differentiated genes that result in good classification metrics between disease and healthy samples. To extract the cluster-differentiated genes, R's DESeq2 algorithm [35] was used. DESeq2 accepts two groups of data as input. As such, for each cluster, a differential expression analysis is performed between the samples that belong to that particular cluster and all the samples from the rest of the clusters (Fig. 4). Thus, the ranked genes based on their p-adjusted values were extracted for each cluster. For simplicity and direct comparison, we selected only the top gene from each cluster, as these genes contribute the most to the mathematical distance between the clusters. As Fig. 4 shows, for every dataset, the DESeq2 algorithm ran n times, where n is the number of clusters, resulting in n top genes.

Classification was set up as a binary problem between disease and healthy samples, using the previously extracted top genes as the only classification features. Due to the fact that every sample (patient) has multiple gene counts vectors (one for each timepoint), every vector was regarded as a separate disease sample in the context of this classification. More specifically, for the GSE80060 dataset 148 disease samples and 22 healthy samples were considered. Likewise, for the GSE97075 dataset, 30 disease samples and 15 healthy samples were used. GSE9863 contains 60 disease samples but no healthy ones. Thus, as in [14], the samples from GSE47683 [36] were used instead, which refer to a different disease (67 renal-transplant patients), but on the same experimental platform (GPL6271). As sophisticated classification estimators, the Random Forest (RF) and the Gradient Boosting (GB) classifiers were selected. Additionally, the Support Vector Machine (SVM) and the

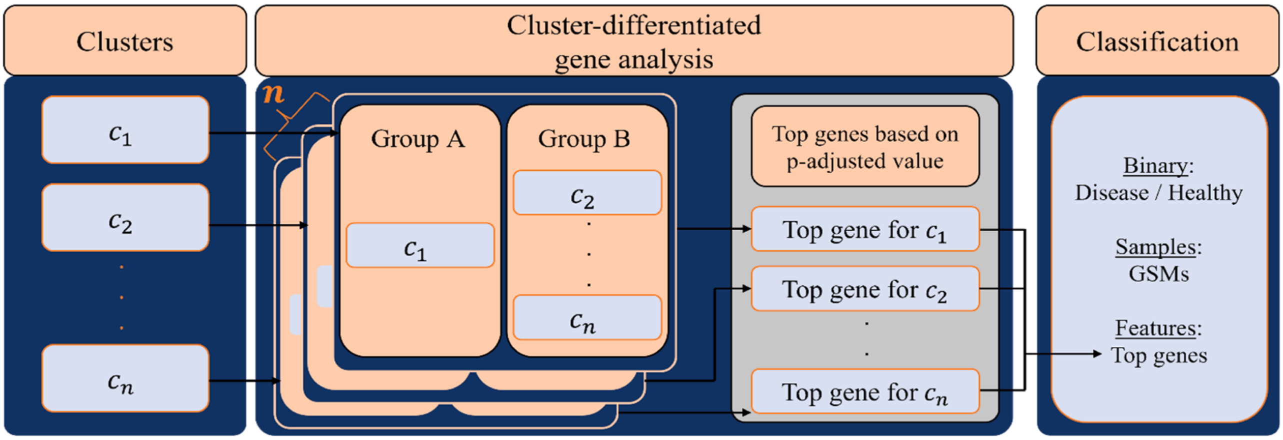


Fig. 4. The DE - classification pipeline.

Logistic Regression (LR) were also implemented. All classifiers were applied with their default parameter settings [24], except for SVM which was utilized with a polynomial kernel. The accuracy, sensitivity, specificity and Area Under the Curve (AUC) were chosen as the classification metrics. Finally, a five-fold stratified cross validation was implemented to address the class imbalance of the datasets.

4. Results

4.1. Clustermaps

The clustering analysis for the GSE80060 dataset resulted in four clusters for Clust3D, five clusters for SOMbrero and four clusters for KMEANS (Table S1 in Appendix). The clustermaps of Clust3D and KMEANS were almost identical except for three samples (2513, 3212 and 413). Likewise, the clustering analysis for the GSE97075 dataset resulted in three clusters for Clust3D, four clusters for SOMbrero and two clusters for KMEANS (Table S3 in Appendix). Even though this dataset contains very few samples, all the clustermaps were unique. Finally, the clustering analysis for the GSE9863 dataset resulted in four clusters for Clust3D, four clusters for SOMbrero [14] and in two clusters for KMEANS (Table S5 in Appendix). Similarly in this case, all the clustermaps were unique.

To provide a visual assessment of the cluster memberships considering all timepoints at once, scatter plots of the data points were created. To achieve this, the first principal component (PC1) was used at all timepoints as the axes. It should be noted, that even though the first principal component contributes the most to the explained variance and the scatter plot are indicative of that, the second principal component is not taken into consideration when plotting the scatter plots. For the GSE9863 dataset, Fig. 5 shows the scatter plots for all clustering algorithms, using PC1 in timepoints T1, T2 and T3. For the GSE80060 dataset, Fig. F1 in Appendix depicts the scatter plots for all clustering algorithms, using PC1 in timepoints T1 and T2. Finally, for the GSE97075 dataset, the visualization of the clustermaps is impractical due to the large number of timepoints (Table 1).

4.2. Clustering indices

Table 2 shows the clustering indices for the three datasets. Due to these indices reflecting a different aspect of clustering assessment (variance, distance, and similarity) the best result of each index is bolded. Fig. 6 shows the silhouette scores for each data point (patient) for all datasets and clustering algorithms. It offers a visual measure of confidence that each data point belongs to its designated cluster.

4.3. DE and classification

Tables 3–5 provide the classification metrics when using the top features (i.e., genes, as described in Section 3.2.1.1) from each algorithms' DE analysis. For the GSE80060 dataset, Table 3 shows the results using the top four features from Clust3D, the top five features from SOMbrero, and the top four features from KMEANS. Table S2 in Appendix shows the extracted features from each algorithm's DE analysis, along with their p-adjusted values. Likewise, Table 4 shows the results for the GSE97075 dataset using the top three features from Clust3D, the top four features from SOMbrero, and the top feature from KMEANS. Table S4 in Appendix shows the extracted features with their p-adjusted values. Due to DE scheme (Fig. 4) and the fact that KMEANS' clustering resulted in just two clusters, the same gene probe (*ILMN_2066060*) ranked first in both clusters. Finally, for the GSE9863 dataset, Table 5 shows the classification metrics when using the top four features from Clust3D, the top four features from SOMbrero, and the top five features from KMEANS. Table S6 in Appendix shows the extracted features and their p-adjusted values.

5. Discussion

We developed a novel clustering methodology that accepts 3D data as input. We tested it on three publicly available datasets and demonstrated its ability to produce well separated clusters. This assessment was made both through clustering indices and a DE - classification pipeline.

We used the Calinski-Harabasz index, the Davies–Bouldin index and the Silhouette score to evaluate the clustering output of Clust3D, compared to that of SOMbrero and KMEANS utilizing the. heuristic 2-step clustering framework. Since no clustering metrics exist for time-related clustering, modifications to those indices had to be made to address the inclusion of the temporal dimension, by substituting the calculation of vector distances with the Frobenius norm of the matrices. We acknowledge that such algebraic modifications can have an impact on the indices' results, and most likely differ confidence-wise from values that we are accustomed to in the general literature. However, due to the fact that every methodology is being assessed with the exact same metrics, there is no doubt about the improvement that Clust3D contributes.

In GSE80060, SOMbrero's results were subpar compared to Clust3D and KMEANS. Also, the clustermaps of Clust3D and KMEANS were almost identical, therefore the derived indices were very similar, however with a slight edge in favor of Clust3D. Fig. 6 shows the similarity in SS scores between Clust3D and KMEANS and their improvement over SOMbrero. In GSE97075, Clust3D achieved a CHI score that was more than threefold higher than those of SOMbrero and KMEANS and a BDI

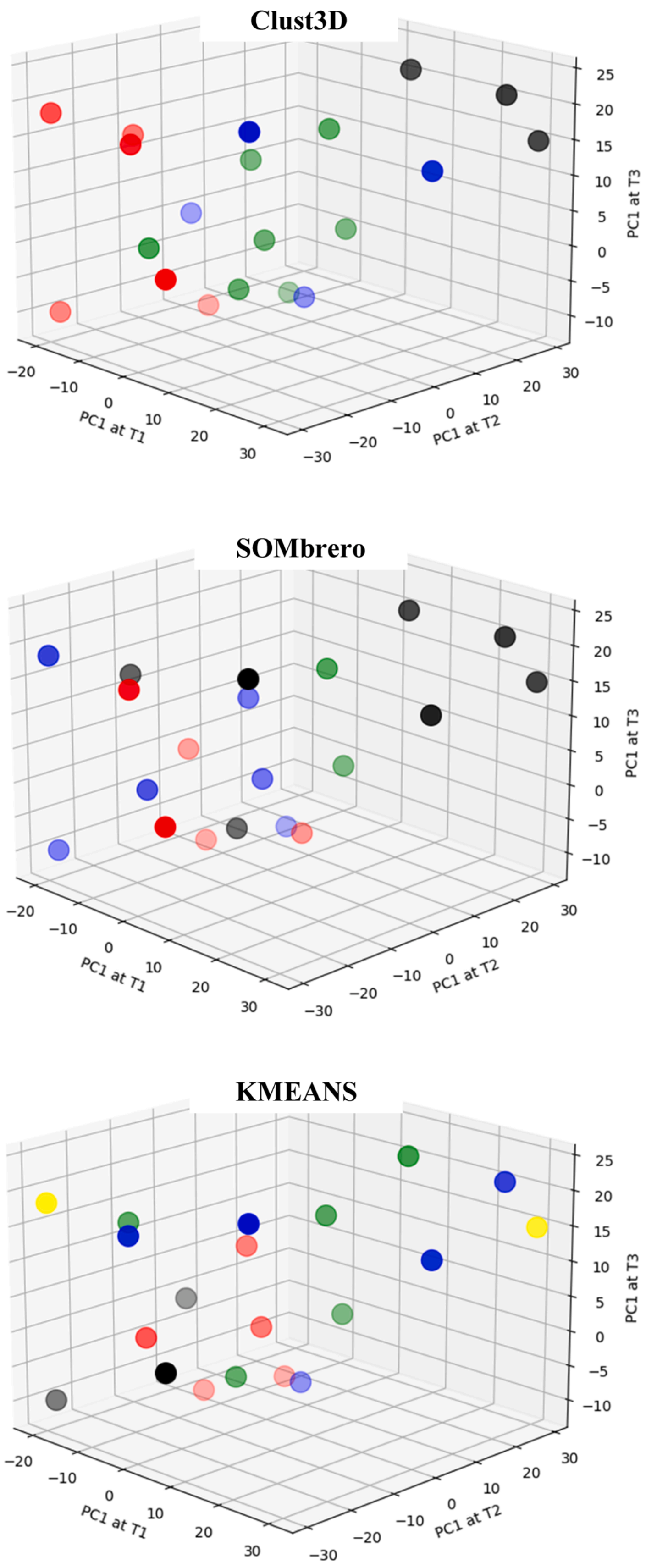


Fig. 5. Scatter plots of the data points (patients) in the GSE9863 dataset for the three clustering methods (Clust3D, SOMbrero and KMEANS). The axes consist of the first principal component of each data point at every timepoint (T1, T2 and T3). The colors differentiate the cluster members. Higher color saturation indicates a closer to the reader locus.

Table 2
Clustering indices for the three datasets. The arrows indicate what is the optimal score (higher or lower). The bold numbers refer to the best result of each index.

GSE80060			
Algorithm	CHI ↑	DBI ↓	SS ↑
Clust3D	**81.17**	**0.32**	**0.55**
SOMbrero	25.47	0.92	-0.12
KMEANS	77.62	0.35	0.54
GSE97075			
Clust3D	**7.61**	**0.22**	**0.17**
SOMbrero	2.04	0.51	-0.11
KMEANS	2.30	0.57	0.14
GSE9863			
Clust3D	**8.95**	**0.61**	**0.26**
SOMbrero	3.50	0.98	-0.02
KMEANS	2.50	1.63	-0.01

score more than twice as good. Contrary to the DBI and the CHI, the improvement of Clust3D in SS score was not that apparent (Fig. 6). Finally, in GSE9863, Clust3D achieved a more than twofold increase in CHI score, a > 60% increase in DB score, and it was the only algorithm that managed a positive SS score (Table 2). Interestingly, as the number of timepoints in the datasets under examination increases, so does the percentage improvement in the clustering indices of Clust3D in comparison to the other algorithms. These results demonstrate a favorable correlation between Clust3D's better index performance and an increase in timepoints. Based on variance, distance, and similarity metrics, our findings affirm that Clust3D offers a substantial clustering improvement in light of the examined datasets.

We also achieved improved clustering, in the context of the utilized DE- classification pipeline (Tables 3–5, Fig. 6). The classification based on Clust3D's analysis managed to achieve higher classification metrics in all datasets, compared to the metrics resulted from SOMbrero's and

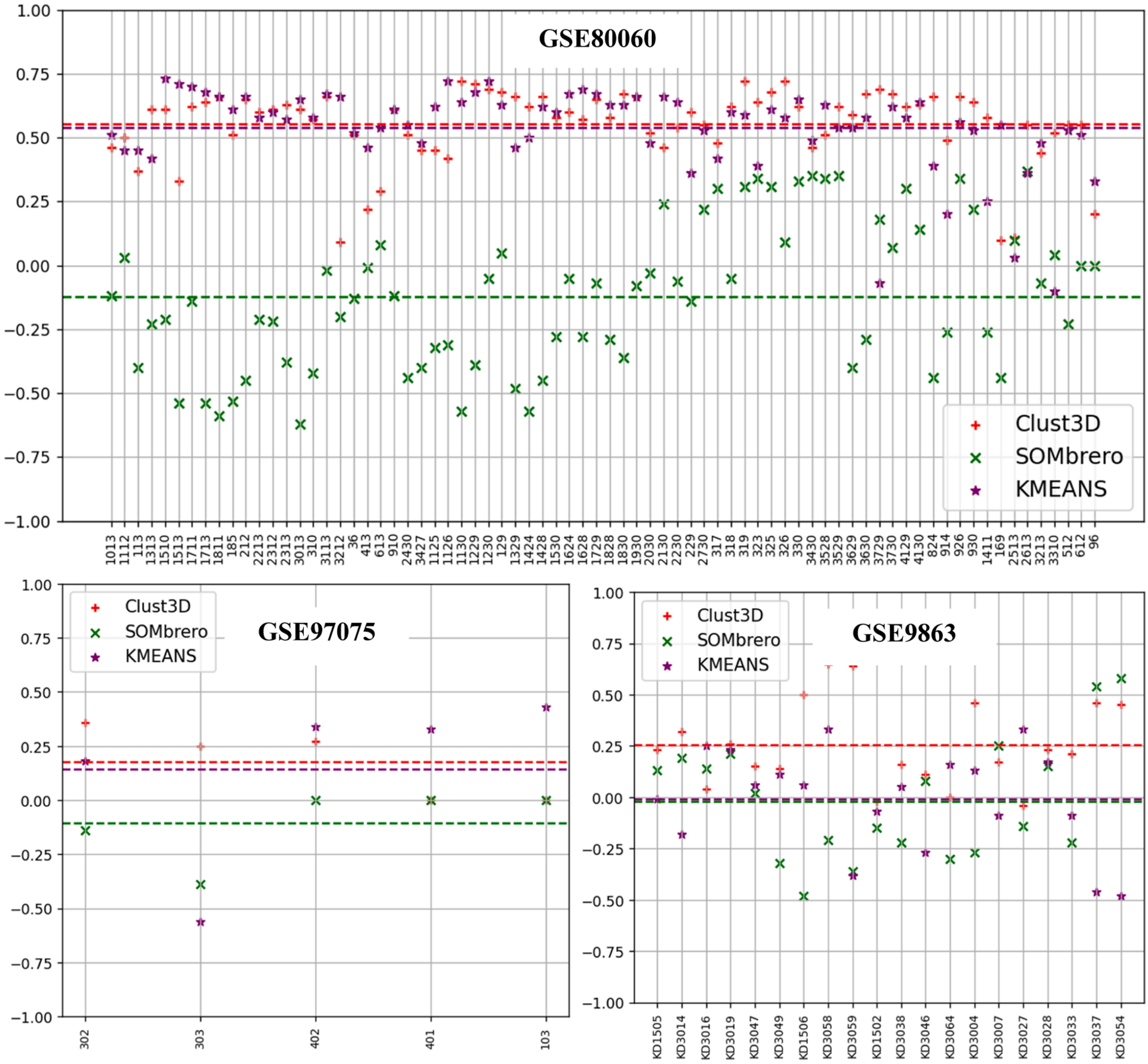


Fig. 6. Silhouette scores for every data point (patient) of each dataset for all clustering tools (Clust3D, SOMbrero and KMEANS). The higher the score, the higher the confidence of a correct cluster membership. Dotted lines represent the mean SS scores (Table 3) for each algorithm. Higher mean scores indicate better overall cluster separation.

Table 3
Classification metrics (mean and standard deviation) of the GSE80060 dataset analysis, based on the derived clusters of Clust3D, SOMbrero and KMEANS.

Clust3D (4 clusters)			
Classifier	Acc	Sens	Spec
GB	0.96 (0.04)	0.97 (0.04)	0.91 (0.11)
RF	0.96 (0.03)	0.98 (0.02)	0.83 (0.15)
SVM	0.91 (0.05)	0.95 (0.06)	0.67 (0.32)
LR	0.81 (0.01)	1.00 (0.00)	0.00 (0.00)
SOMbrero (5 clusters)			
Classifier	Acc	Sens	Spec
GB	0.88 (0.08)	0.96 (0.08)	0.37 (0.12)
RF	0.89 (0.05)	0.97 (0.06)	0.37 (0.12)
SVM	0.85 (0.02)	0.95 (0.03)	0.13 (0.11)
LR	0.81 (0.01)	1.00 (0.00)	0.00 (0.00)
KMEANS (4 clusters)			
Classifier	Acc	Sens	Spec
GB	0.91 (0.04)	0.96 (0.03)	0.57 (0.24)
RF	0.89 (0.02)	0.96 (0.01)	0.45 (0.25)
SVM	0.89 (0.02)	0.96 (0.04)	0.38 (0.34)
LR	0.81 (0.01)	1.00 (0.00)	0.00 (0.00)

Table 4
Classification metrics (mean and standard deviation) of the GSE97075 dataset analysis, based on the derived clusters of Clust3D, SOMbrero and KMEANS.

Clust3D (3 clusters)			
Classifier	Acc	Sens	Spec
GB	0.84 (0.05)	0.93 (0.08)	0.67 (0.22)
RF	0.87 (0.04)	0.93 (0.08)	0.73 (0.25)
SVM	0.69 (0.13)	0.73 (0.25)	0.60 (0.25)
LR	0.67 (0.0)	1.00 (0.00)	0.00 (0.00)
SOMbrero (4 clusters)			
Classifier	Acc	Sens	Spec
GB	0.68 (0.20)	0.77 (0.17)	0.53 (0.34)
RF	0.69 (0.04)	0.73 (0.08)	0.60 (0.13)
SVM	0.67 (0.12)	0.78 (0.13)	0.47 (0.16)
LR	0.73 (0.09)	0.97 (0.07)	0.27 (0.13)
KMEANS (2 clusters)			
Classifier	Acc	Sens	Spec
GB	0.60 (0.05)	0.60 (0.17)	0.60 (0.33)
RF	0.60 (0.05)	0.60 (0.17)	0.60 (0.33)
SVM	0.67 (0.0)	1.00 (0.00)	0.00 (0.00)
LR	0.67 (0.0)	1.00 (0.00)	0.00 (0.00)

Table 5
Classification metrics (mean and standard deviation) of the GSE9863 dataset analysis, based on the derived clusters of Clust3D, SOMbrero and KMEANS.

Clust3D (4 clusters)			
Classifier	Acc	Sens	Spec
GB	0.99 (0.02)	1.00 (0.00)	0.99 (0.03)
RF	0.99 (0.02)	1.00 (0.00)	0.99 (0.03)
SVM	0.99 (0.02)	0.98 (0.03)	1.00 (0.00)
LR	0.99 (0.02)	1.00 (0.00)	0.98 (0.03)
SOMbrero (4 clusters)			
Classifier	Acc	Sens	Spec
GB	0.97 (0.03)	0.98 (0.03)	0.95 (0.06)
RF	0.98 (0.03)	0.98 (0.03)	0.97 (0.06)
SVM	0.97 (0.03)	0.97 (0.04)	0.97 (0.04)
LR	0.98 (0.02)	0.98 (0.03)	0.99 (0.03)
KMEANS (5 clusters)			
Classifier	Acc	Sens	Spec
GB	0.91 (0.06)	0.90 (0.06)	0.93 (0.08)
RF	0.91 (0.04)	0.92 (0.05)	0.91 (0.09)
SVM	0.76 (0.05	0.97 (0.04)	0.57 (0.11)
LR	0.76 (0.09)	0.87 (0.13)	0.66 (0.22)

KMEANS' analyses. This indicates the better efficacy of our clustering algorithm to extract pathway-specific genes for further clinical analysis. The DE analysis was setup in such a way to further evaluate the quality of the clustering, by being applied between subgroups (clusters) of only the disease populations. Thereby, this study did not delve into the biological interpretability of the extracted features.

For the high sample sized GSE80060 dataset, we observed a $>$ 5% increase in accuracy and a significant improvement in specificity ($>$ 60%) for the RF and GB classifiers (Table 3). Fig. 7a shows the improvement in the AUC scores. While SVM did result in a similar percentage increase, the overall metrics were lower. LR resulted in the same non-satisfactory metrics (zero specificity) for all clustering pipelines. This result most likely stems from the lack of linear feature dependency. It is worth mentioning that even though the difference in clustering between Clust3D and KMEANS lies in just three samples, the different genes extracted from the DE analysis were responsible for the considerably improved classification metrics. These genes were substantially more potent in avoiding false positives compared to the genes extracted from SOMbrero's and KMEANS' analyses.

For the low sample sized GSE97075 dataset, the classification based on Clust3D's analysis resulted in a $>$ 26% higher accuracy, a $>$ 20% higher sensitivity and a $>$ 21% higher specificity (Table 4), for the RF and GB classifiers. Regardless of the low sample size and the stratified nature of the cross validation, which yielded higher standard deviations, Clust3D achieved better results in this context too. Fig. 7b shows the improvement in the AUC scores. The results from the SVM and LR were too inconsistent to extract robust conclusions. We suspect that this is a consequence of the adequate sample size.

Lastly, in the case of the large sample sized GSE9863 dataset, Clust3D's analysis once again outperformed the other methods, achieving higher classification metrics (for all classifiers), and even managing perfect sensitivity scores (Table 5). Fig. 7c shows the improvement in the AUC scores. Fig. 5 depicts the scatter plots of the members/samples for all clustering algorithms. Clust3D was able to create by far the most distinct and well-defined clusters across the three dimensions.

It is worth noting that there is a lack of correlation between the

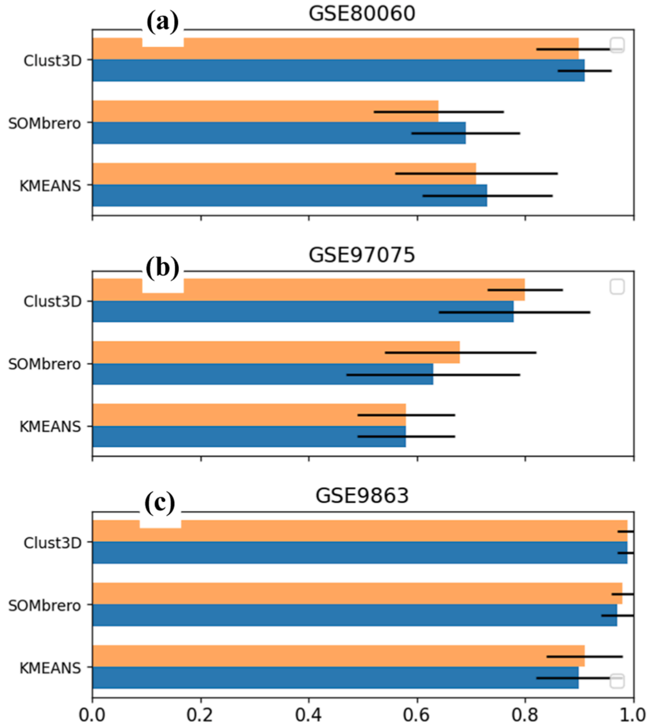


Fig. 7. Classification AUC scores based on the outputs from the three clustering algorithms (Clust3D, SOMbrero and KMEANS) for all datasets: (a) GSE80060, (b) GSE97075, and (c) GSE9863. In orange the scores from the GB classification estimator are depicted while in blue those from RF.

classification metrics obtained from the sets of features extracted using different algorithms and their corresponding p-adjusted values. Even though there are features extracted from SOMbrero's and KMEANS' clusters that have lower p-adjusted values than some of Clust3D's extracted features (Tables S2, S4 and S6), in no case did they result in higher classification metrics (Tables 3–5). Although these values reflect the statistical significance, they do not inherently capture the biological importance of the identified features. From the obtained results we can deduce that the approach of clustering patients based on their gene profile changes over time and the subsequent extraction of features that differentiate the temporal dynamics can potentially uncover key variables that represent complex biological interactions or pathway-level changes. These variables are likely to hold finer biological relevance and contribute to the improved classification performance observed in our study. In contrast, a per-timepoint clustering as in [14] is dictated by the gene expressions of each timepoint and not the temporal expression profile as a dynamic system. This can introduce bias related to the expression of genes at a single timepoint, contributing to seemingly higher statistical confidence when extracting the associated genes. Examining both the clustering indices (Table 2, Figs. 4–5) and the DE – classification pipeline results (Tables 3–5, Fig. 6), we make the case for a robust and consistent clustering algorithm, capable of clustering efficiently high-dimensional timeseries data, better than existing heuristic methods [14,30,31].

Clustering three-dimensional data using mainstream algorithms without heuristic approaches is achievable with the incorporation of a preprocessing step, that mainly involves the flattening or decomposition of one of the dimensions. Furthermore, innovative methods have been proposed that implement a third dimension into the clustering analysis using SOMs (3D-SOM). These methods utilize 3D neuron mapping to extend the capabilities of the SOM algorithm in relation to the structure arrangement of its output neurons. However, those clustering approaches are applied to two dimensional datasets [37,38]. On the contrary, this study proposes a novel way to directly cluster 3D data, exploiting the entire data structure. A clustering technique that takes into consideration the spatial and temporal aspects of a dataset, is the spatial-temporal clustering. This technique is employed to detect clusters or groups within datasets, characterized by their close spatial proximity and similar temporal patterns. This method proves valuable for analyzing datasets encompassing both spatial and temporal dimensions, such as environmental monitoring, crime, traffic, and epidemiological data [18]. Nevertheless, they typically use a two-dimensional matrix as input, usually corresponding to spatial coordinates (e.g. latitude and longitude) and temporal information (e.g. timepoints) [19]. Subsequently, such frameworks cannot address multiple samples (i.e. patients) with multiple features (i.e. genes) at different time intervals (timepoints). In order to implement the comparative method, SOMbrero and KMEANS were employed. SOMbrero was selected for its direct comparability [14], whereas KMEANS was chosen for its well-established status. The study's findings indicated that employing clustering with Clust3D yielded superior clustermaps and classification metrics compared to the comparative approach, which introduced biases by conducting ordinary clustering per timepoint. This inherent bias in per-timepoint clustering underscored the necessity for Clust3D. Consequently, the utilization of cutting-edge algorithms still would not address this issue, due to this bias being inherent in the heuristic method and not in the algorithms that implement it.

This study was primarily focused on autoinflammatory gene expression datasets. Nevertheless, Clust3D's neural network training design makes it applicable for clustering gene expressions timeseries from other diseases and even complex data structures with temporal dimensions, other than gene expressions. Potential applications of Clust3D could include exploratory analysis on clinical timeseries data from patients with infectious diseases (like COVID-19) or chronic disorders, and even biomarker identification studies by utilizing Clust3D's clustermaps alongside more disease-targeted DE analyses. Furthermore, population and epidemiological studies which heavily rely on timeseries cohorts, could benefit on a quality control basis, from the efficient and time-addressing clustering of Clust3D.

Nonetheless, it is important to acknowledge and address the limitations of Clust3D. First, the calculations utilize the Euclidean distance. Even though in this study it is applicable due to the applied PCA, more sophisticated datasets may require different distance metrics to address a higher number of features. Second, although this study tries to remain true to the mathematical equations of the clustering indices while also exploring a variety of evaluation concepts, the lack of well-established metrics in the literature for such a tool is apparent. Third, dimensional reduction of high-dimensional data such as gene expressions is a recurrent obstacle in the field of bioinformatics. While a reasonable effort was made in Clust3D to provide efficient techniques, still it is a very challenging task to capture the total explained variance of a high-dimensional dataset, which would be ideal in order to differentiate patients' profiles across timepoints. Lastly, another limitation is the inability of Clust3D to produce a fixed neuron grid in 3D space for the best clustering convergence. Whilst Clust3D employs resourceful techniques for optimizing the neuron initialization process (optimal neuron positions based on Euclidean distance dispersion), stochasticity still remains in the clustering.

Future work will focus on overcoming such limitations, improving existing features, and implement new modules. The incorporation of these modules will delve into the intricate connections within the three-dimensional data, such as variations across different timepoints, thereby establishing a stronger foundation. Finally, considering that this study provides a general computational framework, the capabilities and the benefits of Clust3D will be evaluated by using it extensively on a variety of datasets with intricate dimensions.

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

CRediT authorship contribution statement

Dimitrios Fotiadis: Funding acquisition, Methodology, Project administration, Resources, Supervision, Writing – review & editing. **Vasileios C. Pezoulas:** Formal analysis, Methodology, Writing – review & editing. **Orestis D. Papagiannopoulos:** Conceptualization, Formal analysis, Methodology, Software, Writing – original draft, Writing – review & editing. **Costas Papaloukas:** Conceptualization, Methodology, Project administration, Supervision, Writing – original draft, Writing – review & editing.

Data availability

Clust3D is available on GitHub (https://github.com/Orepap/Clust3D).

Acknowledgments

This project has received funding from the European Union's Horizon 2020 research and innovation program under grant agreement No 779295 (ImmunAID – Immunome project consortium for Auto-Inflammatory Disorders).

Appendix A. Supporting information

Supplementary data associated with this article can be found in the online version at doi:10.1016/j.csbj.2024.05.003.

References

[1] Efthimiou P, Paik P, Bielory L. Diagnosis and management of adult onset Still's disease. Ann Rheum Dis 2006;65(5):564–72.

[2] Betrains A, et al. Systemic autoinflammatory disease in adults. Autoimmun Rev 2021;20(4).

[3] Donato G, et al. Monogenic autoinflammatory diseases: state of the art and future perspectives. Int J Mol Sci 2021;22(12).

[4] Krainer J, Siebenhandl S, Weinhäusel A. Systemic autoinflammatory diseases. J Autoimmun 2020;109:102421.

[5] Wang J, et al. Low-ratio somatic NLRC4 mutation causes late-onset autoinflammatory disease. Ann Rheum Dis 2022.

[6] Zheng W, et al. Single-cell analyses highlight the proinflammatory contribution of C1q-high monocytes to Behçet's disease. Proc Natl Acad Sci 2022;119(26).

[7] Govaere O, et al. Transcriptomic profiling across the nonalcoholic fatty liver disease spectrum reveals gene signatures for steatohepatitis and fibrosis. Sci Transl Med 2020;12(572).

[8] Luo Y, et al. SOMAscan proteomics identifies serum biomarkers associated with liver fibrosis in patients with NASH. Hepatol Commun 2021;5(5):760–73.

[9] Chen G, et al. Serum proteome analysis of systemic JIA and related lung disease identifies distinct inflammatory programs and biomarkers. Arthritis Rheumatol 2022.

[10] Fong T, et al. Identification of plasma proteome signatures associated with surgery using SOMAscan. Ann Surg 2021;273(4):732.

[11] Begic E, et al. SOMAscan-based proteomic measurements of plasma brain natriuretic peptide are decreased in mild cognitive impairment and in Alzheimer's dementia patients. PLOS ONE 2019;14(2).

[12] Papagiannopoulos O, et al. Comparison of high-throughput technologies in the classification of adult-onset still's disease patients. 44th Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC). IEEE,; 2022.

[13] Papagiannopoulos O, et al. Comparison of proteomic approaches in autoinflammatory disease classification. IEEE-EMBS International Conference on Biomedical and Health Informatics (BHI). IEEE,; 2022.

[14] Pezoulas VC, et al. A computational workflow for the detection of candidate diagnostic biomarkers of Kawasaki disease using time-series gene expression data. Comput Struct Biotechnol J 2021;19:3058–68.

[15] Barturen G, et al. Moving towards a molecular taxonomy of autoimmune rheumatic diseases. Nat Rev Rheumatol 2018;14(2):75–93.

[16] Magni P, et al. TimeClust: a clustering tool for gene expression time series. Bioinformatics 2008;24(3):430–2.

[17] Abu-Jamous B, Kelly S. Clust: automatic extraction of optimal co-expressed gene clusters from gene expression data. Genome Biol 2018;19(1):1–11.

[18] Zhicheng S, Pun-Cheng LSC. Spatiotemporal data clustering: a survey of methods. ISPRS Int J geo-Inf 2019;8(3):112.

[19] Birant D, Kut A. ST-DBSCAN: an algorithm for clustering spatial–temporal data". Data Knowl Eng 2007;60(1):208–21.

[20] Brown G, et al. Gene: a gene-centered information resource at NCBI. Nucleic Acids Res 2015;43:D36–42.

[21] Brachat AH, et al. Early changes in gene expression and inflammatory proteins in systemic juvenile idiopathic arthritis patients on canakinumab therapy. Arthritis Res Ther 2017;19(1):1–10.

[22] Arostegui JI, et al. Open-label, phase II study to assess the efficacy and safety of canakinumab treatment in active hyperimmunoglobulinemia D with periodic fever syndrome. Arthritis Rheumatol 2017;69(8):1679–88.

[23] Popper SJ, et al. Gene-expression patterns reveal underlying biological processes in Kawasaki disease. Genome Biol 2007;8(12):1–12.

[24] R C, Harris, et al. Array programming with NumPy. Nature 2020;585:357–62.

[25] McKinney W. Data structures for statistical computing in Python. Proc 9th Python Sci Conf 2010;vol. 445(1).

[26] Pedregosa F, et al. Scikit-learn: machine learning in Python. J Mach Learn Res 2011;12:2825–30.

[27] Hunter JD. Matplotlib: a 2D graphics environment. Comput Sci Eng 2007;9(3): 90–5.

[28] Schreiber James B. Issues and recommendations for exploratory factor analysis and principal component analysis. Res Soc Adm Pharm 2021;17(5):1004–11.

[29] Kohonen T. The self-organizing map. Neurocomputing 1998;21:1–6.

[30] Boelaert Julien, et al. SOMbrero: an R package for numeric and non-numeric self-organizing maps. Advances in Self-Organizing Maps and Learning Vector Quantization. Cham: Springer,; 2014. p. 219–28.

[31] Lloyd S. Least squares quantization in PCM. IEEE Trans Inf Theory 1982;28(2): 129–37.

[32] Caliński T, Harabasz J. A dendrite method for cluster analysis. Commun Stat-Theory Methods 1974;3(1):1–27.

[33] Davies D, Bouldin D. A cluster separation measure. IEEE Trans Pattern Anal Mach Intell 1979;(2):224–7.

[34] Rousseeuw J. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. J Comput Appl Math 1987;20:53–65.

[35] Love MI, Huber W, Anders S. Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2. Genome Biol 2014;15:550.

[36] Brouard S, et al. Identification of a peripheral blood transcriptional biomarker panel associated with operational renal allograft tolerance. Proc Natl Acad Sci 2007;104:15448–53.

[37] Mohd Z, et al. Data clustering and topology preservation using 3d visualization of self organizing maps. Proc World Congr Eng 2012;Vol. 2.

[38] Xueyan Z, et al. 3D SOM Leaming And Neighborhood Algorithm. 2018 5th International Conference on Systems and Informatics (ICSAI). IEEE,; 2018.