BioData Mining

METHODOLOGY

Open Access

Feature graphs for interpretable unsupervised tree ensembles: centrality, interaction, and application in disease subtyping

Christel Sirocchi[1,2], Martin Urschler[3] and Bastian Pfeifer[3*]

*Correspondence:
bastian.pfeifer@medunigraz.at

[1] Department of Pure and Applied Sciences, University of Urbino, Urbino 61029, Italy
[2] Biomedical Network Science Lab, Department Artificial Intelligence in Biomedical Engineering, Friedrich-Alexander Universität Erlangen-Nürnberg, Erlangen 91052, Germany
[3] Institute for Medical Informatics, Statistics and Documentation, Medical University of Graz, Graz 8036, Austria

Abstract

Explainable and interpretable machine learning has emerged as essential in leveraging artificial intelligence within high-stakes domains such as healthcare to ensure transparency and trustworthiness. Feature importance analysis plays a crucial role in improving model interpretability by pinpointing the most relevant input features, particularly in disease subtyping applications, aimed at stratifying patients based on a small set of signature genes and biomarkers. While clustering methods, including unsupervised random forests, have demonstrated good performance, approaches for evaluating feature contributions in an unsupervised regime are notably scarce. To address this gap, we introduce a novel methodology to enhance the interpretability of unsupervised random forests by elucidating feature contributions through the construction of feature graphs, both over the entire dataset and individual clusters, that leverage parent-child node splits within the trees. Feature selection strategies to derive effective feature combinations from these graphs are presented and extensively evaluated on synthetic and benchmark datasets against state-of-the-art methods, standing out for performance, computational efficiency, reliability, versatility and ability to provide cluster-specific insights. In a disease subtyping application, clustering kidney cancer gene expression data over a feature subset selected with our approach reveals three patient groups with different survival outcomes. Cluster-specific analysis identifies distinctive feature contributions and interactions, essential for devising targeted interventions, conducting personalised risk assessments, and enhancing our understanding of the underlying molecular complexities.

Keywords: Feature selection, Unsupervised random forest, Interpretable machine learning, Feature graphs, Disease subtyping

Introduction

Interpretable machine learning has become a critical focus across diverse domains, as understanding the reasoning behind model predictions is widely regarded as at least as important as achieving high predictive accuracy [1]. To address this need, two main

approaches are adopted. Interpretability by design involves developing models that are inherently transparent in their decision-making process, while post-hoc interpretability seeks to explain the predictions of opaque models using techniques such as feature importance and counterfactual analysis [2]. In this context, decision trees stand out for their transparency in the decision-making process. However, their interpretability diminishes as they aggregate into tree ensembles. Random forests exhibit remarkable performance, particularly in tabular data, often outperforming deep learning techniques [3]. This advantage is especially notable in domains like biomedicine, where datasets are commonly organised in tabular form. Thus, the challenge lies in balancing the interpretability of individual decision trees with the enhanced predictive power of ensemble methods.

Random forests have also proven effective in unsupervised learning for computing affinity matrices, which can be further analysed through clustering techniques [4, 5]. This capability holds significant promise in biomedicine, particularly for disease subtyping, where advanced clustering algorithms stratify patients based on shared characteristics such as clinical features and molecular profiles [6]. Disease subtyping, powered by the collection of multi-omics data and the development of advanced tools leveraging these data sources [7, 8], enables researchers to identify distinct subgroups within a disease, thereby enhancing the understanding of its underlying molecular complexities [9, 10], ultimately facilitating personalised medical treatments. In these applications, uncovering the factors driving clustering decisions, both globally and within individual clusters, is crucial.

To leverage the performance of complex models such as random forests without sacrificing interpretability, the field of eXplainable AI (XAI) has advanced, offering methods to uncover the underlying explanatory factors of black-box model predictions. XAI has introduced various explanation techniques, ranging from interpretable models that approximate input-output relations to methods that highlight the most influential input features [11, 12]. Feature importance methods play a pivotal role in improving model interpretability, as they help identify relevant features and support feature selection [13]. However, while feature importance in random forests has been extensively studied in supervised settings [14], its exploration in unsupervised learning remains notably underexplored. Enhancing interpretability in unsupervised random forests is therefore a critical pursuit toward the practical applicability of these models and forms the focus of our research.

Novelty In our recent work [15], we proposed a novel split rule for training random forests in an unsupervised manner to derive affinity matrices for clustering. We explored its application to disease subtyping and multi-omic data analysis in a federated setting and highlighted the need for more advanced techniques to interpret these models. In this study, we address this gap by introducing a novel methodology centred on constructing and mining feature graphs. These graphs are designed to identify the most relevant features and feature combinations utilised by the model in the construction of affinity matrices.

Compared to existing methods for evaluating feature importance and performing feature selection in unsupervised learning [16–18], our framework offers several key

advantages. By constructing feature graphs, rather than computing feature importance scores, the proposed framework provides a more comprehensive feature representation, capturing both individual feature importance and feature combinations. Additionally, it supports the construction of both dataset-wide and cluster-specific feature graphs, mapping feature contributions to specific clusters. Because it leverages the structure of the trained unsupervised random forest directly rather than outcomes of clustering algorithms, the proposed method is agnostic to the specific clustering method applied to the affinity matrix, and therefore can be used in conjunction with hierarchical clustering [19] as well as spectral clustering [20], affinity propagation [21] and more [22, 23]. For this same reason, it does not require prior knowledge of the number of clusters-an important limitation in most existing methods-ensuring broader applicability to real-world datasets. Furthermore, our graph-mining strategy for feature selection demonstrates superior performance compared to state-of-the-art methods, both in terms of clustering accuracy, reliability, and computational efficiency, offering a versatile and efficient solution for feature selection in unsupervised settings. Overall, this methodology is particularly suited for disease subtyping applications, characterised by a need for cluster-specific insights and computational efficiency due to large data dimensionality.

To summarise, the main contributions of this paper are:

- A novel graph-building method for generating dataset-wide and cluster-specific feature graphs from unsupervised random forests, mapping feature contributions.
- A graph-mining strategy that selects relevant feature combinations from the constructed graphs, offering superior clustering performance at a lower computational cost.
- A case study on kidney cancer gene expression data for disease subtyping, demonstrating the value of the proposed approach in a real-world biomedical application.

Overview of the method In the proposed framework, we construct feature graphs by leveraging the parent-child node splits within unsupervised random forests. Each tree node is labelled with its corresponding split feature, and edges between parent-child nodes are weighted based on multiple criteria and aggregated to form feature graphs where nodes represent features, and edges connect pairs of features with weights equal to the sum of all corresponding edge weights across the trees. Feature importance is quantified using out-degree centrality, and feature selection is approached through two strategies: (1) a brute-force method termed *brute-graph*, with exponential complexity, selecting the subgraph of a given size with the highest total edge weight; (2) a greedy approach named *greedy-graph*, with polynomial complexity and competitive performance, which incrementally adds features that maximise the average edge weight with already selected features. The proposed graph-building and graph-mining methods are extensively evaluated on synthetic and benchmark datasets.

Organisation of the manuscript The article is structured as follows. Previous work section provides a comprehensive review of previous work on strategies for feature selection and model interpretability in unsupervised settings. Additionally, it outlines previous

efforts in leveraging existing feature graphs to enhance random forests or constructing feature graphs from tree-based models. In Methods section, the proposed methodology is introduced, beginning with a concise overview of unsupervised random forests and then detailing the method for constructing and mining feature graphs to estimate feature importance and perform feature selection. Evaluation section describes the synthetic datasets generated within the study and the benchmark datasets utilised, as well as the experiments conducted. Results and discussion section presents and discusses the results of the investigation, showcasing a practical application in disease subtyping, and, finally, Conclusion and future work section concludes by summarising our findings and outlining promising avenues for future research.

Previous work

In this section, we first review key strategies for deriving feature importance and performing feature selection in unsupervised settings, emphasising their limitations in addressing feature interactions (Feature importance in unsupervised learning section). We then explore approaches that use feature graphs to enhance the interpretability and biological relevance of random forest models (Leveraging feature graphs with random forests section). Finally, we examine data-driven methods for constructing feature graphs from tree-based models in supervised contexts (Building feature graphs from random forests section). This section highlights the need for efficient feature importance methods in unsupervised random forests and the potential of feature graphs to capture meaningful feature interactions, forming the basis for our approach.

Feature importance in unsupervised learning

Unsupervised machine learning plays a crucial role in biomedical research, where clustering assists in grouping patients based on clinical or molecular features, supporting tasks such as disease subtyping and drug discovery for personalised medicine [24]. In disease subtyping, identified clusters often comprise patients with distinct phenotypic characteristics, requiring customised treatment. Accurately selecting key features for each cluster enables clinicians to classify patients using a small set of signature genes and potential biomarkers, facilitating the development of personalised screening tests [25]. This is challenging due to the potentially high number of features (e.g., tens of thousands in omics data).

Therefore, in these applications, identifying the features that most contribute to clustering assignments is crucial for providing actionable insights. While the field of eXplainable Artificial Intelligence (XAI) has introduced numerous techniques to enhance model interpretability, research efforts in unsupervised learning remain relatively limited, with many XAI approaches recasting unsupervised problems into supervised ones [26]. For instance, Ismaili et al. [16] proposed a *classification-based* approach that employs clustering predictions as response variables to train a classifier, deriving feature importance for clustering from the classifier's feature importance (e.g., Gini impurity for random forests [27]). Although this method is versatile and computationally efficient, it assumes knowledge of the number of clusters and may misrepresent the underlying data structure if the cluster assignment is predicted incorrectly. In response to the need for feature importance metrics tailored to unsupervised learning, various

approaches have been proposed, though their applicability is often limited to specific clustering methods. For example, Pfaffel et al. [28] developed a feature importance strategy tailored to k-means clustering, while Zhu et al. [29] focused on model-based clustering. More akin to our approach, Cabezas et al. [18] offered a *phylogeny-based* framework for hierarchical clustering, providing a multi-resolution view of feature contributions by leveraging the dendrogram structure. This method does not require knowledge of the number of clusters; however, its computational complexity scales with the number of features, as it requires fitting an evolutionary model for each feature, making it impractical for high-dimensional datasets. Among model-agnostic feature importance methods, the *leave-one-variable-out* approach, proposed by Badih et al. [17], compares within-cluster heterogeneity by omitting each feature from the dataset. However, as in the *classification-based* approach, it assumes knowledge of the number of clusters. Similar to the *phylogeny-based* approach, it requires training a separate model for each feature, making it computationally demanding. Notably, none of the presented methodologies effectively considers feature combinations or provides a straightforward means to evaluate cluster-specific feature importance.

In the context of feature selection for unsupervised learning, methods have been categorised based on their interaction with clustering algorithms [30]. Filter methods assess feature relevance using statistical metrics, such as correlation or variance, and are computationally efficient but do not offer insights into how features contribute to clustering [31]. Conversely, wrapper methods search for a feature subset by training the clustering algorithm on various candidate subsets, offering insights into feature importance but posing challenges related to the selection of a predefined selection criterion and high computational costs [32]. Another notable limitation of methods from both groups, particularly univariate filter methods and sequential wrapper methods, is the tendency to overlook contextual feature interactions, resulting in the selection of confounded features that lack direct relevance to the task [31].

To address the lack of feature interaction context, several strategies have emerged within the domain of tree ensembles, some leveraging existing graphs, while others compute feature graphs from data. However, to our knowledge, all of these methods have been applied in a supervised setting.

Leveraging feature graphs with random forests

In the past decade, several studies, particularly in the realm of biomedicine, have leveraged graphs representing known relationships among features to improve the quality of random forest models. These efforts aimed to address various challenges, such as reducing over-fitting, and enhancing model robustness, data efficiency, interpretability, and coherence with prior knowledge. Most importantly, they sought to identify feature modules with greater biomedical relevance able to explain underlying physio-pathological mechanisms.

These methodologies modify random forests through two primary strategies. First, they refine feature bagging - the sub-sampling of features used for training each tree. For example, trees can be trained on feature sets sampled through random walks over a graph [33], or by selecting features from the neighbourhood of a seed node [34]. The second strategy focuses on enhancing split feature selection - the sub-sampling of features

at each node during tree growth. This can entail choosing features based on weights derived from biological networks [35], topological importance scores derived from random walks [36], probabilistic distribution over graphs [37], or simply from the neighbourhood of the parent node [38–40].

Together, these approaches demonstrate the potential of feature graphs to improve random forest models by enhancing their interpretability and aligning them with biological knowledge. Notably, all these approaches have been exclusively developed for supervised settings.

Building feature graphs from random forests

In the absence of feature graphs mapping known relationships among features, such graphs can be constructed by observing how the model utilises features in the learning phase. A few recent approaches have proposed graph-based representations for supervised tree ensembles, representing features as vertices and parent-child relationships as edges, offering concise, interpretable summaries of relationships among features.

Kong et al. [41] built an unweighted directed feature graph from a supervised random forest based on the splitting order of features in its decision trees, such that their graph-embedded deep feed-forward network (GEDFN) could be used in the absence of an existing feature graph. Bayir et al. [42], on the other hand, constructed a weighted directed feature graph from each decision tree in a supervised forest, derived high-dimensional vectors from these graphs, and transformed them into a topological cluster network to select representative decision trees and build more efficient random forests. Finally, Cantao et al. [43] represented random forest classifiers as weighted directed feature graphs and explored various centrality measures to rank features and perform effective feature selection.

These studies construct feature graphs from supervised random forests, primarily trained for classification tasks, by assigning unitary weight to each parent-child split and without considering leaf nodes. The resulting feature graphs are tailored to specific downstream tasks, such as feature selection, topological data analysis, or integration into neural architectures, without fully exploring their potential. In contrast, our study focuses on building feature graphs from random forests in unsupervised settings, an area that remains largely unexplored. It proposes diverse edge-building criteria for constructing both general and cluster-specific feature graphs and presents various strategies for leveraging these graphs to underscore the potential of using feature graphs for both feature selection and model interpretability.

Methods

Addressing the need for tailored feature importance methods in unsupervised learning and the rising application of random forests in clustering tasks, this article presents a novel approach for enhancing the interpretability of unsupervised random forests by building and mining feature graphs. A visual summary of the key steps in the proposed methodology is illustrated in Fig. 1.

The proposed method begins with training a random forest in an unsupervised manner, using a fixation index splitting rule [15] (detailed in Unsupervised random forests section). Then, each node in every tree of the forest is labelled with the

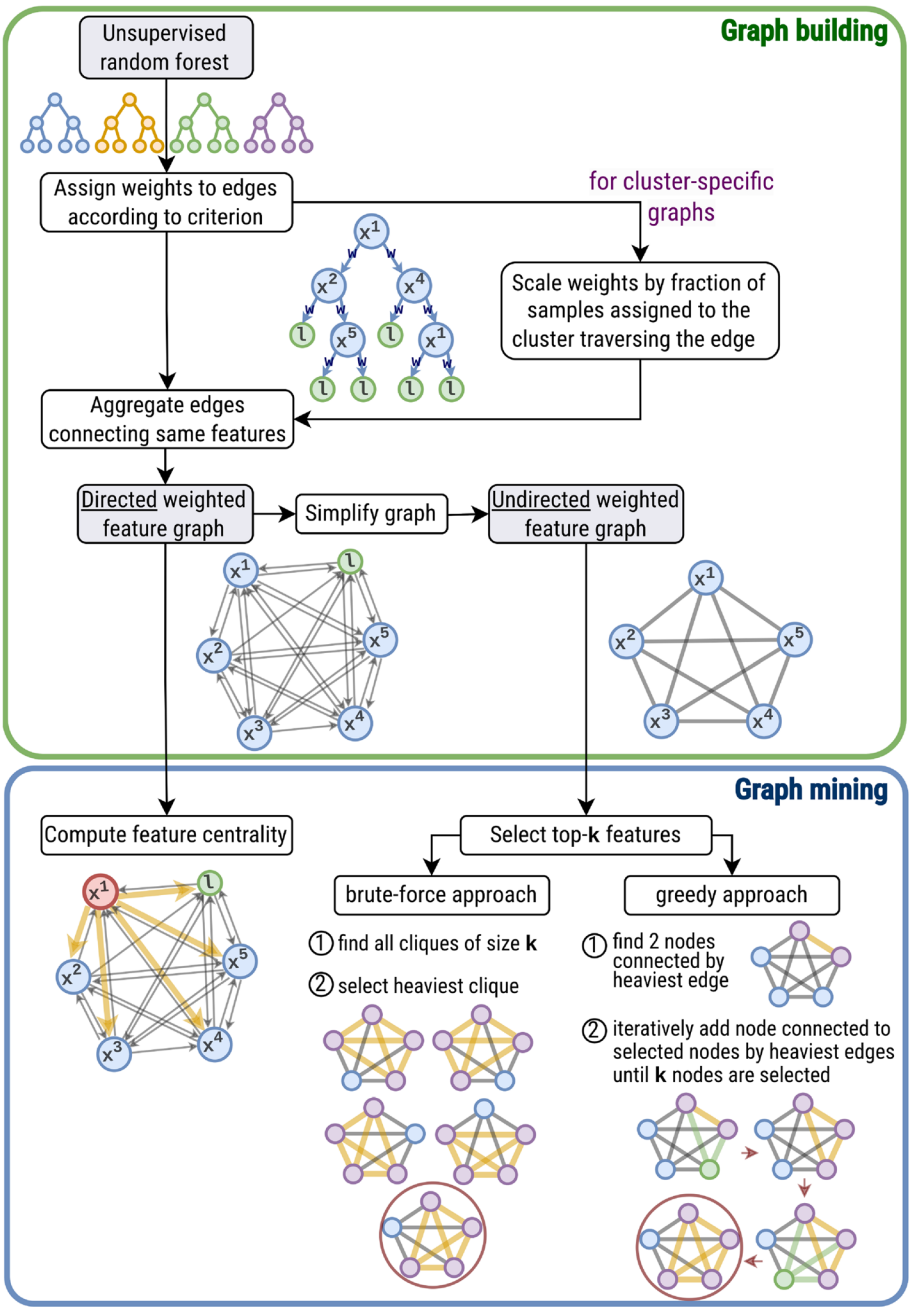


Fig. 1 Diagram outlining the steps of the proposed method

corresponding split feature, and each edge connecting parent-child nodes is assigned a weight based on one of four proposed criteria. A feature graph is then constructed where nodes represent features, and edges connect any two features with weights equal to the sum of all edge weights connecting those two features in the trees (as formalised in Building the feature graph section). In the resulting feature graphs, the centrality of features within the resulting feature graph captures their relevance to the clustering task, while the edges between features reflect the effectiveness of the feature pair in discriminating clusters, offering insights into feature interactions. Additionally, cluster-specific

feature graphs are constructed by scaling all edge weights by the proportion of data samples traversing the edge that is assigned to the considered cluster, thereby offering insights into feature importance within each cluster, which most available methods do not provide.

In the constructed feature graph, feature selection is carried out using two strategies (outlined in Mining the feature graph section). A brute-force method termed *brute-graph*, with exponential complexity, identifies the top k features by evaluating all connected subgraphs of size k, selecting the subgraph with the highest total edge weight. A greedy approach named *greedy-graph*, with polynomial complexity and comparable performance, first selects the two features connected by the heaviest edge and iteratively adds the feature that maximises the average weight of edges with already selected features. This feature selection approach is less computationally intensive than most available methods as it does not scale with the number of features. It is also more versatile as it can be applied to all clustering techniques leveraging an affinity matrix.

Overall, the proposed methodology is particularly suited for disease subtyping, where cluster-specific importance is often more relevant than overall importance, and where a small set of key features needs to be selected from a large number of features.

Unsupervised random forests

Random Forests are ensembles of decision trees, with each tree denoted as t, constructed independently of each other using a bootstrapped or subsampled dataset. The training dataset comprises n data points or samples x_i, with i from 1 to n, over the feature space X, where $X = \{x^1, x^2, ..., x^d\}$ and d is the dimensionality of the feature space, so that each x_i is a vector from R^d, and x_i^j denotes the value of feature j for sample i. Each tree t represents a recursive partitioning of the feature space X and each of its nodes $v \in t$ corresponds to a subset, typically a hyper-rectangle, in the feature space X, denoted as $R(v) \subset X$. The root node v_0 is the initial node of each decision tree and corresponds to the entire feature space, i.e. $R(v_0) = X$. The number of training samples falling into the hyper-rectangle $R(v)$ is denoted as $N(v)$.

Each tree t is grown using a recursive procedure which identifies two distinct subsets of nodes: the set of internal nodes $S(t)$ which apply a splitting rule to a partition of the feature space X, and the set of leaf nodes $L(t)$ which represent the terminal partitions of the feature space X and, in supervised settings, provide prediction values for regression tasks or class labels for classification tasks. A split at node v generates two child nodes v^{child} of node v, a left child v^{left} and a right child v^{right}, which divide $R(v)$ into two separate hyper-rectangles, $R(v^{\text{left}})$ and $R(v^{\text{right}})$, respectively. The depth of node v, denoted as $P(v)$, is defined as the length of the path from the root node v_0, so that $P(v_0) = 0$ and $P(v^{\text{left}}) = P(v^{\text{right}}) = P(v) + 1 \;\; \forall v \in t$.

The split at node v is decided in two steps. First, a subset $M(v)$ of features is chosen uniformly at random. Then, the optimal split feature $x(v) \in M(v)$ and split value $z(v) \in \mathbb{R}$ are determined by maximising the decrease in impurity given by:

$$\Delta_{\mathcal{I}}(v, x(v), z(v)) := \mathrm{I}(v) - \frac{N(v^{\text{left}})\mathrm{I}(v^{\text{left}}) - N(v^{\text{right}})\mathrm{I}(v^{\text{right}})}{N(v)} \tag{1}$$

where $I(v)$ is some measure of impurity. Thus, $\Delta_{\mathcal{I}}(v, x(v), z(v))$ indicates the decrease in impurity for node v due to $x(v)$ and $z(v)$.

In supervised contexts, impurity measures such as entropy and Gini index are commonly used as split criteria. In unsupervised scenarios, in the absence of a response vector y to guide data splitting at nodes, several strategies have been proposed [4]. One approach involves training a standard supervised forest to distinguish between original data (positive class) and synthetically generated data (negative class) [44, 45]. Another method leverages Extremely Randomized Trees, which rely on random splits of the data [46], while a third strategy involves training forests using unsupervised splitting criteria. For example, density random forests employ split rules based on density estimation, utilising either parametric or non-parametric methods [47, 48]. More recently, a splitting rule inspired by the fixation index from population genetics has been introduced [49]. The fixation index computes the average pairwise distances between samples within and across groups formed by a node split [15]. Formally:

$$\Delta_{\mathcal{F}}(v, x(v), z(v)) := \frac{(D_{\square}(v^{\text{left}}) + D_{\square}(v^{\text{right}}))/2}{D_{\nabla}(v)} \tag{2}$$

where

$$D_{\square}(v) := \frac{\sum_{i,h}(x_i(v) - x_h(v))^2}{N(v)(N(v)-1)} \quad \text{for} \quad i \neq h \tag{3}$$

and

$$D_{\nabla}(v) := \frac{\sum_{i,h}(x_i(v^{\text{left}}) - x_h(v^{\text{right}}))^2}{N(v^{\text{left}})N(v^{\text{right}})}, \tag{4}$$

where x is a given candidate feature and x_i, x_h refers to the specific value in sample i and sample h. Once the unsupervised random forest is built, an affinity matrix measuring between pairwise data points can be derived, for instance, by counting the number of times each pair of samples appears in the same leaf node. This affinity matrix can then serve as an input to any distance-based clustering algorithm, such as hierarchical clustering, affinity propagation, or spectral clustering [4]. Once the chosen clustering algorithm is applied, it generates a partition $\mathcal{C} = \{C_1, C_2, .., C_p\}$ of the data samples, where each C_g (with g ranging from 1 to p) represents an individual cluster, and p denotes the total number of identified clusters. For every sample x_i, the algorithm assigns a value c_i denoting its membership to a cluster C_g. Consequently, any two samples x_i and x_h belong to the same cluster $C_g \in \mathcal{C}$ if and only if their assigned membership values are equal, i.e., $c_i = c_h$.

Building the feature graph

Consider a random forest F trained on a dataset over the feature space X, comprising trees t made of nodes v, where each node is associated with a feature x or a leaf. A weighted directed graph G, denoted as $G(F) = (V, E, W)$, can be constructed from F, where:

- V is the set of vertices of the graph, representing the d features in X as well as l to denote leaf nodes, i.e., $V = \{x^1, x^2, \ldots, x^d\} \cup \{l\}$;
- E is the set of directed edges, where each edge is an ordered pair of vertices in V, i.e., $E = \{(u_i, u_j) \mid u_i, u_j \in V\}$;
- W is the set of weights assigned to each directed edge, indicating some measure of capacity or cost associated with traversing the edge, i.e., $W = \{w_{ij} \mid (u_i, u_j) \in E\}$.

The graph can also be represented by its adjacency matrix $A(G)$, where $A[u_i, u_j] = w_{ij}$. Given these premises, the proposed method constructs a feature graph from the structure of the random forest with features as nodes, and edges reflecting the occurrence of any two features labelling adjacent nodes (parent and child nodes) across all trees of the forest. The elements of the adjacency matrix are then defined as follows:

$$
\begin{aligned}
A[x^i, x^j] &= \sum_{t \in F} \sum_{\substack{v \in S(t) \\ x(v) = x^i \\ x(v^{\text{child}}) = x^j}} q(v, v^{\text{child}}) \quad \forall\, x^i, x^j \in X; \\
A[x^i, l] &= \sum_{t \in F} \sum_{\substack{v \in S(t) \\ x(v) = x^i \\ v^{\text{child}} \in L(t)}} q(v, v^{\text{child}}) \quad \forall\, x^i \in X; \\
A[l, x^i] &= 0 \quad \forall\, x^i \in X.
\end{aligned}
\tag{5}
$$

where $q(v, v^{\text{child}})$ is a quantity assigned to the node pair v and v^{child}, defined differently according to one of the following four edge-building criteria.

- Under the *present* criterion, $q(v, v^{\text{child}}) = 1$, thus counting the occurrences of any two features labelling adjacent nodes across all trees of the forest.
- According to the *fixation* criterion, $q(v, v^{\text{child}}) = \Delta_{\mathcal{F}}(v, x(v), z(v))$, where the contribution of each edge corresponds to the fixation index computed at that node, prioritising more effective splits.
- Under the *level* criterion, $q(v, v^{\text{child}}) = P(v^{\text{child}})^{-1}$, scaling the contribution of each edge by the depth of the node, giving more weight to splits closer to the root, which are assumed to have a greater impact on the outcome by affecting more samples.
- As per the *sample* criterion, $q(v, v^{\text{child}}) = \frac{N(v^{\text{child}})}{N(v_0)}$, with the contribution of each edge scaled by the number of data samples traversing that edge over the total number of samples, emphasising splits that affect more samples in the dataset.

The proposed approach can be tailored to construct cluster-specific feature graphs by scaling the edge weights (determined based on a specified criterion) by the proportion of data samples associated with a particular cluster assignment. This adjustment assigns greater importance to splits that predominantly affect samples assigned to a given cluster, under the assumption that if certain features effectively discriminate a particular cluster, samples assigned to that cluster will predominantly traverse paths in the tree containing those features.

To construct a cluster-specific graph, each quantity $q(v, v^{\text{child}})$ is multiplied by a factor $f(v, v^{\text{child}}, C)$, defined as

$$f(v, v^{\text{child}}, C) = \frac{|\{x_i \in v^{\text{child}} \wedge c_i = C\}|}{N(v^{\text{child}})} \tag{6}$$

where the numerator of the fraction computes the cardinality of the set of samples traversing the edge and assigned to a given cluster C. Notably, for every $q(v, v^{\text{child}})$, the cluster-specific multiplicative factors $f(v, v^{\text{child}}, C)$ sum to 1, i.e.,

$$\sum_{C \in \mathcal{C}} f(v, v^{\text{child}}, C) = 1 \;\; \forall\, v, v^{\text{child}} \in t, \; \forall\, t \in F\,. \tag{7}$$

Consequently, the constructed feature graph can be regarded as the sum of its cluster-specific feature graphs.

Mining the feature graph

The feature graph built based on the structure of a unsupervised random forest can be used to identify the features most effective in splitting data into clusters. In a weighted directed graph, the influence of a vertex can be measured using a variety of centrality measures, and out-degree centrality in particular. The weighted out-degree centrality $C_{\text{out}}^{\text{weighted}}(u_i)$ of a vertex u_i is the sum of the weights of edges outgoing from u_i:

$$C_{\text{out}}^{\text{weighted}}(u_i) = \sum_{u_j \in V} w_{ij}\,. \tag{8}$$

The constructed graph can also be leveraged to identify subsets of relevant features for feature selection purposes. To this end, we propose two approaches, a brute-force method and a greedy algorithm, referred to as *brute-graph* and *greedy-graph* respectively, to identify subsets of vertices in the graph that are connected by the heaviest edges. Both strategies consider edges connecting features while excluding l and its incident edges, as it represents terminal tree nodes not associated with any splitting feature. Additionally, they do not take into account self-edges or the direction of the edges. In this context, the weighted undirected graph $G_X = (V_X, E_X, W_X)$ is defined, induced by the feature set $\{x^1, x^2, \ldots, x^d\}$, with edge weights equal to the average of the weights of corresponding edges in the directed graph, i.e., $V_X = X$, $E_X = \{\{u_i, u_j\} \mid u_i, u_j \in V_X, u_i \neq u_j, (u_i, u_j) \in E\}$, and $W_X = \{\frac{w_{ij}+w_{ji}}{2} \mid u_i, u_j \in V_X, (u_i, u_j) \in E, w_{ij} \in W\}$.

Brute-graph feature selection

The top k features can be evaluated in a brute-force manner by inspecting all connected subgraphs of size k in G_X.

A subgraph $G' = (V', E', W')$ of size k is defined as the subgraph of G_X induced by the set of k vertices $V' \subseteq V_X$, containing all edges in E_X whose endpoints are in V', i.e. $E' = \{\{u_i, u_j\} \in E_X \mid u_i, u_j \in V'\}$. The resulting graph G' is said to be connected if, for every pair of vertices $u_i, u_j \in V'$, there exists a path in the graph from vertex u_i

to vertex u_j. Formally, G' is connected if and only if there is a sequence of vertices $u_1, u_2, \ldots, u_k \in V'$ such that for each i with $1 \leq i < k$, the edge $\{u_i, u_{i+1}\} \in E'$ exists.

The total weight (TW) of the subgraph G' is given by the sum of weights of all edges in E', and the average weight (AW) can be defined as the total weight divided by the maximum number of edges, i.e.,

$$\mathrm{TW}(G') = \sum_{\{u_i,u_j\}\in E'} w_{ij} \in W' \qquad \text{and} \qquad \mathrm{AW}(G') = \frac{2TW(G')}{|V'|(|V'|-1)}\,. \tag{9}$$

The top k features according to the proposed brute-force approach correspond to the vertex set $V' \subseteq V_X$ of size k inducing a subgraph G' of G_X that maximises the average (or total) edge weight, i.e.,

$$V' = \arg \max_{\substack{V' \subseteq V_X \\ |V'| = k \\ G' \text{ is connected}}} \mathrm{AW}(G')\,. \tag{10}$$

This approach faces computational limitations due to the growth in the number of possible subgraphs relative to the size of the feature set. Specifically, for d features, the maximum number of subgraphs of size k is determined by $\frac{d!}{k!(d-k)!}$. Consequently, this method exhibits exponential computational complexity and becomes infeasible for large feature spaces.

Greedy-graph feature selection

To overcome the exponential computational complexity, the top k features can also be determined in a greedy manner by initially selecting the two features associated with vertices in G_X that are connected by the heaviest edge. Subsequently, features are iteratively added by maximising the average weights of edges with the already selected features. Given a graph G_X, a vertex $u_i \in V_X$ and a vertex set $V' \subset V_X$ such that $u_i \notin V'$, the average weight of the new edges (AWN) connecting u and V' can be defined as:

$$\mathrm{AWN}(G_X, u_i, V') = \frac{1}{|V'|} \sum_{u_j \in V'} w_{ij} \in W_X\,. \tag{11}$$

The greedy procedure outlined in Algorithm 1 yields the top k features from the graph G_X, along with two weight lists: the average edge weight of the subgraph induced by the selected feature set and the average edge weight connecting the newly added feature to the previously selected features. These weight lists offer valuable insights into determining the optimal number of features. The algorithm operates by evaluating all neighbours of selected nodes at each iteration, resulting in approximately $k \cdot |E_X|$ operations, which approaches $k \cdot d^2$ operations for dense graphs (which feature graphs often are). Consequently, this method exhibits polynomial computational complexity and can be efficiently implemented.

Algorithm 1 Greedy-graph feature selection algorithm of top k features

Input: $G_X = (V_X, E_X, W_X)$, k
Output: V'
Initialise: $\mathcal{A} \leftarrow \emptyset$, $\mathcal{B} \leftarrow V_X$ {initialise added and to-add feature sets}
Initialise: $avg_n \leftarrow [\,]$, $avg \leftarrow [\,]$ {initialise lists for edge weight sum and avg}
$\{u_1, u_2\} = \arg\max_{u_i, u_j \in G_X} w_{ij} \in W_X$ {find vertices with max edge weight}
avg.append(w_{ij}), avg_n.append(w_{ij}) {update edge weight lists}
$\mathcal{A} \leftarrow \{u_1, u_2\} \cup \mathcal{A}$, $\mathcal{B} \leftarrow \mathcal{B} \setminus \{u_1, u_2\}^*$ {update feature sets}
while $|\mathcal{A}| <= k$ **do**
  $u_i = \arg\max_{u \in \mathcal{B}}$ AWN$(G_X, u, \mathcal{A})$ {find vertex with max avg edge weight}
  avg.append(AW$(G_X[\mathcal{A}])$) {update edge weight lists}
  avg_n.append(AWN$(G_X, u_i, \mathcal{A})$)
  $\mathcal{A} \leftarrow \{u_i\} \cup \mathcal{A}$, $\mathcal{B} \leftarrow \mathcal{B} \setminus \{u_i\}^*$ {update feature sets}
end while
return $\mathcal{A}$, avg, avg_n

For both feature selection strategies, all features are eligible for selection only if the constructed feature graph is connected. The probability of a graph being connected depends on the density of the graph, defined as the ratio of actual edges to possible edges, i.e., equivalent to $\frac{2|E_X|}{(|V_X|)(|V_X|-1)}$ for undirected graphs and $\frac{|E_X|}{(|V_X|)(|V_X|-1)}$ for directed graphs. As the density of edges increases, the probability of finding a path between any pair of vertices typically increases. In random graphs, there often exists a critical edge density above which the graph is almost surely connected [50]. The density of the feature graph is thus determined by the number of features ($|V_X|$) and the number of edges ($|E_X|$), which depend on the number of trees. To achieve a connected feature graph it is often sufficient to train a greater number of trees. Alternatively, both feature selection strategies can be modified so that, initially, the connected components of the graph are identified using a graph traversal algorithm. Then, the feature selection approach can be deployed concurrently in the largest connected components. In summary, while the proposed approaches rely on a connected feature graph, this assumption is often satisfied and generally attainable by training a greater number of trees. In the event of a disconnected graph, the algorithms can be adapted to operate on the largest connected components.

Evaluation

In this Evaluation section, we provide a comprehensive analysis of the methodologies outlined in Methods section for deriving feature graphs from unsupervised random forests and mining these graphs for feature selection. We begin by assessing the effectiveness of the proposed graph construction methods on synthetic datasets by exploring the relationship between centrality and feature relevance, as well as the correlation between edge weight and the discriminatory power of feature pairs. Specifically, Out-degree centrality and edge weight section examines feature graphs constructed from the entire dataset, while Cluster-specific feature graphs section focuses on cluster-specific feature graphs. Next, we evaluate our *brute-graph* and *greedy-graph* feature selection methods for mining feature graphs. We first evaluate these methods on synthetic datasets with varying numbers of relevant and irrelevant features in Feature selection on synthetic datasets with relevant features section, and with redundant features in Feature selection on synthetic datasets with repetitive features section. Then, we compare our methods

with state-of-the-art feature importance techniques on benchmark datasets in Feature selection on benchmark datasets section. Finally, we illustrate the utility of our *greedy-graph* method for interpretability through a disease subtyping application on an omics dataset in Interpretable disease subtype discovery section.

Throughout all experiments, unsupervised random forests were trained using the fixation index as a splitting rule, utilising the uRF library [15]. For reproducibility, the code used to generate the synthetic datasets and to replicate all experiments is available in the project's GitHub repository[1].

Out-degree centrality and edge weight

The initial evaluation aims to assess whether the generated feature graphs exhibit two desirable properties: (i) features that are more effective in discriminating clusters should exhibit higher out-degree centrality in the graph; (ii) pairs of features that, taken together, are more effective in separating clusters should be connected in the graph by edges with greater weight. To systematically evaluate these properties across the proposed four edge-building criteria, we generated two sets of synthetic datasets.

The first set was generated to comprise relevant features (those that aid in discriminating clusters) and irrelevant features (those that do not). This setup allows us to test whether, in the constructed feature graphs, the centrality of relevant features is greater than that of irrelevant features. Specifically, 30 synthetic datasets were generated, each containing 13 features (3 relevant, 10 irrelevant) with data points distributed across four clusters. Each cluster included 50 data points sampled from a normal distribution (standard deviation 0.2) around predefined centres (cluster centres A in Appendix A of supplementary material). In this arrangement, V_1, V_2, and V_3 are relevant for distinguishing clusters, while the others were irrelevant. Unsupervised random forests were trained on these datasets with a standard parameter configuration: 500 trees, a minimum leaf node size of 5, and a number of features sampled at each node set to the square root of the total number of features (here termed *sqrt*). Feature graphs were then constructed from the trained forests using each of the four proposed edge-building criteria: *present*, *level*, *fixation*, and *sample*. Out-degree centrality was calculated for all features, and the centralities of relevant and irrelevant features were compared using a t-test, with a p-value threshold indicating statistical significance set at 0.05. In this analysis, as for all other cases where the t-test is applied, the normality assumption was verified using the Shapiro-Wilk test.

The second set of synthetic datasets was designed to evaluate whether feature pairs with greater discriminative power are connected by heavier edge weights in the constructed feature graphs. We generated 30 datasets, each containing 13 features (8 relevant, 5 irrelevant) distributed across four clusters. For each cluster, 50 data points were sampled from a normal distribution (standard deviation 0.2) centred around cluster centres B (Appendix A, supplementary material), chosen such that different feature pairs could distinguish varying numbers of clusters (1 to 4). The discriminatory power of selected feature pairs in the generated synthetic datasets is illustrated in Fig. 2 for

[1] https://github.com/ChristelSirocchi/urf-graphs

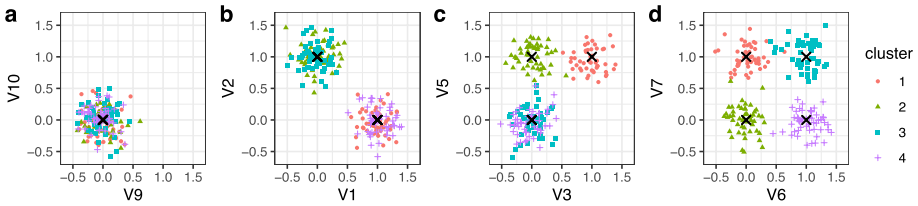


Fig. 2 Feature pairs discriminating **a** 1, **b** 2, **c** 3 and **d** 4 clusters

four representative pairs. For instance, the pair (V_1, V_2) has a discriminative power of 2, as two globular clusters can be distinguished in the two-dimensional space defined by these features, centred at (0, 1) and (1, 0) (Fig. 2b). In contrast, the pair (V_6, V_7) has a maximum discriminative power of 4, as all four clusters can be distinguished in their two-dimensional space, centred at (0, 1), (0, 0), (1, 1), and (1, 0) (Fig. 2d). Unsupervised random forests were trained on these datasets with the same parameter configuration of the previous experiment. We then constructed feature graphs and evaluated the relationship between the discriminative power of feature pairs and the weights of their connecting edges using Pearson's correlation, with statistical significance determined by a p-value threshold of 0.05.

To assess the influence of random forest hyper-parameters on the structure of the generated feature graphs, we repeated the analyses for both sets of synthetic datasets while varying the number of trees (from the set {10, 50, 100, 500, 1000}), the number of features sampled at each node, denoted as MTRY (from the set {1, 2, *sqrt*, n}, where n is the total number of features), and the minimum terminal node size denoted as *min bucket* (from the set {1, 2, 5, 10}).

Cluster-specific feature graphs

The subsequent evaluation aims to determine whether the cluster-specific feature graphs generated by our proposed approach can effectively capture the role of each feature in discriminating individual clusters. For each cluster, features are categorised into three groups: cluster-specific (features that can uniquely identify the cluster), sub-relevant (features that collectively aid in identifying the cluster), and irrelevant (features that do not discriminate the cluster). To systematically evaluate the ability of our cluster-specific feature graphs to differentiate among these categories, we generated synthetic datasets that include features representing each type. Specifically, we generated 30 synthetic datasets, each comprising 13 features (4 relevant and 9 irrelevant) as described in Out-degree centrality and edge weight section, but sampled around cluster centres C (Appendix A, supplementary material). According to this configuration, for each cluster, one of the relevant features is cluster-specific as it can uniquely distinguish the cluster from all others (e.g., V_1 for cluster 1). On the other hand, the other relevant features are sub-relevant, as they can distinguish the cluster but only when taken together (e.g., V_2, V_3, and V_4 for cluster 1). The remaining features (V_5 through V_{13}) are irrelevant for all clusters. Unsupervised random forests were trained on these datasets using 500 trees and *sqrt* feature bagging. Cluster-specific feature graphs were then constructed for each dataset,

cluster, and edge-building criterion. The out-degree centralities of features across the three categories were compared using t-tests, applying a significance threshold of 0.05.

Feature selection on synthetic datasets with relevant features

After confirming that the proposed graph-building strategies generate edge weights that reflect the discriminative power of feature combinations, we evaluated the effectiveness of the proposed *brute-graph* and *greedy-graph* algorithms in identifying relevant features based on edge weights. To this end, synthetic datasets with varying numbers of relevant features were generated. Specifically, datasets comprising 13 features were generated as described in Out-degree centrality and edge weight section, with the number of relevant features q ranging from 3 to 7. For each q, 30 datasets were generated, centred around the first $q + 1$ vectors from cluster centres D (Appendix A, supplementary material), designed to ensure that all relevant features contributed equally to differentiating the clusters across all experiments. Unsupervised random forests were trained on these datasets using the standard parameter configuration, and feature graphs were constructed based on the four edge-building criteria. The *brute-graph* and *greedy-graph* methods were then applied to select the top k features, with k from 2 to 12. During this selection process, we monitored the average edge weight of the subgraph induced by the selected features, under the assumption that a sudden decrease would indicate the inclusion of irrelevant features, thereby revealing the optimal number of features to select.

Cluster separability measures were computed as in [51] for increasing numbers of selected features (k from 2 to 12) and different configurations of relevant features (q from 3 to 7), with results averaged across iterations. Three cluster separability measures were computed and normalised to the [0,1] interval for easier comparison. *Silhouette Score* measures how well samples are clustered, comparing intra-cluster cohesion to inter-cluster separation, with values between -1 and 1. *Separability Index* represents the fraction of samples whose nearest neighbour belongs to the same cluster, with values between 0 and 1. *Hypothesis Margin* [52] computes the difference between the distance to a sample's nearest neighbour in the same cluster (*near-hit*) and the nearest neighbour in a different cluster (*near-miss*), averaged across all samples.

To evaluate the performance of our graph-based approach in identifying relevant features within higher-dimensional datasets, we extended this experimental setup by incorporating 90 and 490 additional irrelevant features, resulting in datasets of 103 and 503 features, respectively. Due to the computational complexity of the brute-force method, these experiments were performed exclusively using the greedy approach.

Feature selection on synthetic datasets with repetitive features

The proposed graph-mining strategies were evaluated for handling redundant features by identifying feature combinations that provide complementary information. To this end, 30 synthetic datasets with 10 features (6 relevant and 4 irrelevant) were generated around cluster centres E (Appendix A, supplementary material). In these datasets, pairs of relevant features, specifically (V_1, V_2), (V_3, V_4), and (V_5, V_6), are

Table 1 Description of benchmark datasets utilised in comparative analysis

Datasets	#Samples	#Features	#Cluster	Cluster
Iris	150	4	3	50,50,50
Liver disorders	345	5	2	169,176
Ecoli	336	7	8	143,77,52,35,20,5,2,2
Breast tissue	106	9	6	21,15,18,16,14,22
Glass	214	9	4	70,76,17,51
Wine	178	13	3	59,71,48
Lymphography	148	18	4	2,81,61,4
Parkinson	195	22	2	48,147
Ionosphere	351	34	2	225,126
Sonar	208	60	2	97,111

redundant, as they are generated around identical coordinates and thus provide overlapping information. Any set of three features including one from each pair can effectively distinguish all clusters. Feature graphs were constructed from unsupervised random forests trained on these datasets using standard parameter configuration and the *sample* edge-building criterion. The average edge weight of every triad within the graph was evaluated to identify optimal feature combinations.

Feature selection on benchmark datasets

The proposed *brute-graph* and *greedy-graph* methods were then benchmarked against three state-of-the-art methods, described in Feature importance in unsupervised learning section: *classification-based* [16], *phylogeny-based* [18], and *leave-one-variable-out* [17]. While *brute-graph* returns a feature subset and *greedy-graph* a feature ranking, the other three methods produce feature weights. Notably, all these strategies are unsupervised, as they do not rely on ground truth information. A comparative analysis of the five methods was conducted on 10 classification benchmark datasets[2] from the UCI Machine Learning repository[3], featured in established frameworks for benchmarking clustering tools [54] and comparative studies of feature selection methods [55]. These datasets vary in the number of features, samples, classes (clusters), and class sizes, as detailed in Table 1.

For each dataset, 30 unsupervised random forests with 500 trees and *sqrt* feature bagging were trained to generate affinity matrices, which were then used for clustering via Ward's linkage method [15, 56]. Feature selection strategies were applied as follows:

- In our approach, feature graphs were constructed from the unsupervised forests using the *sample* criterion. The 30 feature graphs were averaged to produce a final feature graph to which the *brute-graph* and *greedy-graph* algorithms were applied.
- In the *classification-based* approach, random forests were trained in a supervised manner using pseudo-labels derived from hierarchical clustering. Specifically, a random forest with 1000 trees and *sqrt* feature bagging was trained for each clustering

[2] in Liver disorders, the target was obtained by dichotomising the 6th column as in [53]

[3] https://archive.ics.uci.edu/

solution. Feature importance scores were calculated using the impurity-corrected criterion [14], and these scores were averaged across the 30 forests.

- In the *phylogeny-based* approach, phylogenetic feature importance was computed from the dendrograms generated by Ward's linkage, as in [18]. Importance scores were calculated for the 30 dendrograms and averaged. Similar to the *classification-based* method, this approach relies on the output of the clustering algorithm.
- In the *leave-one-variable-out* approach, the importance of each feature was computed by removing it from the dataset and calculating the within-cluster heterogeneity of the remaining features. This procedure was repeated 30 times and the results were averaged [17]. This method does not leverage the unsupervised random forest or the output of the clustering method but instead estimates importance by retraining forests on reduced datasets.

After applying feature selection with each of the five considered methods, 30 unsupervised random forests of 500 trees and *sqrt* feature bagging were trained on the top k features selected by each method, with k ranging from 2 to the total number of features (or a maximum of 12 for larger datasets). Clustering was then applied using Ward's linkage. The effectiveness of each feature selection method was evaluated by comparing the clustering quality on their top k features, measured using the Adjusted Rand Index (ARI), Normalised Mutual Information (NMI), and Fowlkes-Mallows Index (FMI) for increasing numbers of selected features. The monotonicity of clustering performance, defined as the tendency for performance to improve as the number of selected features increases, was estimated by computing a straightforward monotonicity score defined as follows. Given a sequence of w scores $S = [a_1, a_2, \ldots, a_w]$ representing ARI, NMI or FMI values computed for an increasing number of selected features, the monotonicity score is defined as

$$\text{monotonicity} = 1 - \frac{\sum_{i=1}^{w-1} \max(0, a_i - a_{i+1})}{\sum_{i=1}^{w-1} |a_i - a_{i+1}|} \tag{12}$$

where the numerator of the fraction calculates the total decrease in performance across the sequence while the denominator is the total variation in the sequence. The resulting score ranges from 0 to 1, where a score of 1 indicates that the performance consistently increases or remains stable as features are added.

The reliability of the feature selection methods, defined as the stability of the selection process across repeated applications, was assessed by comparing the feature rankings and feature subsets induced by each method over the 30 iterations. Feature ranking stability was evaluated using Spearman's rank correlation and a complemented Canberra distance (such that higher values indicate greater stability) for all methods except *brute-graph*, which produces a feature subset rather than a ranking. Feature subset stability was assessed for all methods using the Kuncheva Index as in [57].

To statistically compare the feature selection methods, differences in clustering performance (ARI, NMI, and FMI scores), monotonicity, and reliability (Spearman, Canberra, and Kuncheva index) were assessed using the Wilcoxon signed-rank test with Bonferroni correction for multiple comparisons, applying a threshold of 0.05 for statistical significance. Finally, computation time for each method was recorded over 30 runs on a

system with an Intel Core i7-7700HQ CPU (2.8–3.8 GHz), 32 GB of RAM, and a 512 GB SSD running Linux OS.

Interpretable disease subtype discovery

The graph-building and graph-mining methods proposed in this study were applied to a real-world disease subtyping problem, demonstrating their practical utility using gene expression data from kidney cancer patients. The study focused on clear cell renal cell carcinoma, the most common histological subtype of kidney cancer (accounting for 70%−80% of cases) and associated with the worst prognosis [58]. This cancer type has well-characterised molecular subtypes with established prognostic relevance [59]. In this context, our proposed method for characterising cluster-specific relevant features and their interactions can aid in the development of more precise molecular signatures for disease subtyping. Such advancements contribute to precision medicine by enhancing the prediction accuracy of patient outcomes and supporting the design of tailored treatment strategies. The analysis utilised gene expression data from the KIRC dataset, a widely used benchmark for studying this disease, available at The Cancer Genome Atlas[4] and the LinkedOmics platform[5]. The data was retrieved from the ACGT lab at Tel Aviv University[6], which provides a curated dataset repository proposed as a benchmark for omics clustering approaches [9].

The retrieved data table initially comprised 208 patients and 20,531 genes. Gene expression patterns were filtered to retain the top 100 features with the highest variance and the bottom 100 features with the lowest variance. Following this pre-processing, the top 15 relevant genes were identified using the proposed greedy selection strategy (see Mining the feature graph section). Based on this gene subset, we performed disease subtyping employing unsupervised random forests as in [15]. We then drew survival curves for the identified clusters and assessed differences using the Cox log-rank test [60], an inferential procedure for comparing event time distributions among independent (i.e., clustered) patient groups. Finally, we computed cluster-specific feature importance and feature graphs (see Building the feature graph section) and presented findings on genes, gene interactions, and their importance to each patient cluster. To validate the robustness of the findings and test the method on a higher-dimensional dataset, the procedure was repeated on a pre-processed gene expression dataset comprising 500 features, obtained by filtering the top 250 features with the highest variance and the bottom 250 with the lowest variance.

Results and discussion

Evaluation on synthetic datasets

Out-degree centrality and edge weight

The first experiment detailed in Out-degree centrality and edge weight section reveals promising attributes of the constructed feature graphs. Across all four edge-building criteria, the centrality of relevant features is consistently greater than that of irrelevant

[4] https://www.cancerimagingarchive.net/collection/tcga-kirc/

[5] https://www.linkedomics.org

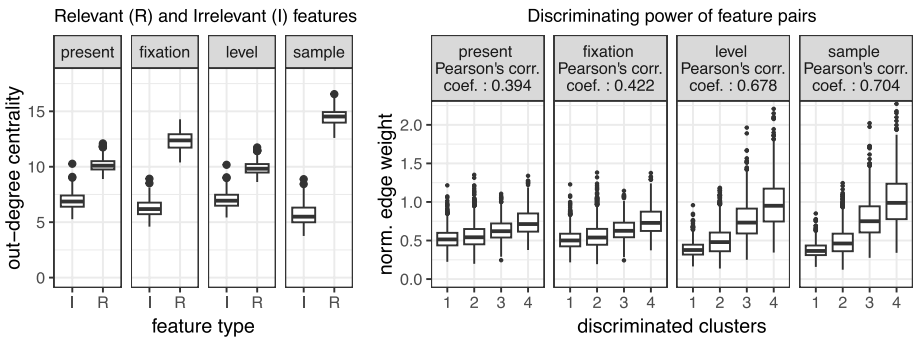
[6] http://acgt.cs.tau.ac.il/multi_omic_benchmark/download.html

features, as evidenced in Fig. 3a. T-tests confirm that these differences are statistically significant (p-value < 1e-16) across all criteria, suggesting that the chosen centrality metric can effectively capture feature importance in clustering tasks. Among the evaluated criteria, *sample* emerges as the most effective in discriminating relevant and irrelevant features, reporting the largest difference in centralities, followed by *fixation*, *present*, and *level*.

The second experiment outlined in Out-degree centrality and edge weight section investigates the correlation between the weight of an edge connecting any two features and the ability of those features to separate data into clusters, as illustrated in Fig. 3b. Across all edge-building criteria, there exists a significant positive correlation between the edge weight and the number of separated clusters, as quantified by Pearson's correlation coefficient. Once again, the *sample* criterion stands out as the most effective, recording the highest correlation coefficient, followed by *level*. Consequently, the edge weight between two features emerges as a reliable indicator of their effectiveness in cluster separation, suggesting that feature selection strategies should prioritise features connected by edges with heavy weights, motivating the proposed feature selection strategies.

The hyper-parameter analysis underscores the influence of parameter choice in model training on the properties of the resulting feature graphs. Figure 4 presents t-test results for centrality analysis and Pearson correlation tests for edge weight analysis across each edge-building criterion and hyper-parameter configuration.

In the centrality analysis (first row of the figure), the difference in centrality between relevant and irrelevant features is evaluated through the –log p-value of the t-test applied to the two groups of features. This difference is greatest when the number of features considered at each node for the best split (MTRY) is set to 2, followed by *sqrt*. On the other hand, configurations using the total number of features yield inferior results, underscoring the importance of feature sub-sampling in forest training. Notably, when a single feature is selected at random at each node, the relative position of features in the trees is uninformative, and the resulting feature graphs fail to differentiate between relevant and irrelevant features. Furthermore, the –log p-value generally increases with the number of trees, particularly from 10 to 500, before plateauing or decreasing at 1000 trees. This trend highlights the importance of selecting a number of trees that align with the complexity of the dataset.



(a) feature centrality (b) feature pair edge weights

Fig. 3 Evaluating out-degree centrality and edge weights of feature graphs

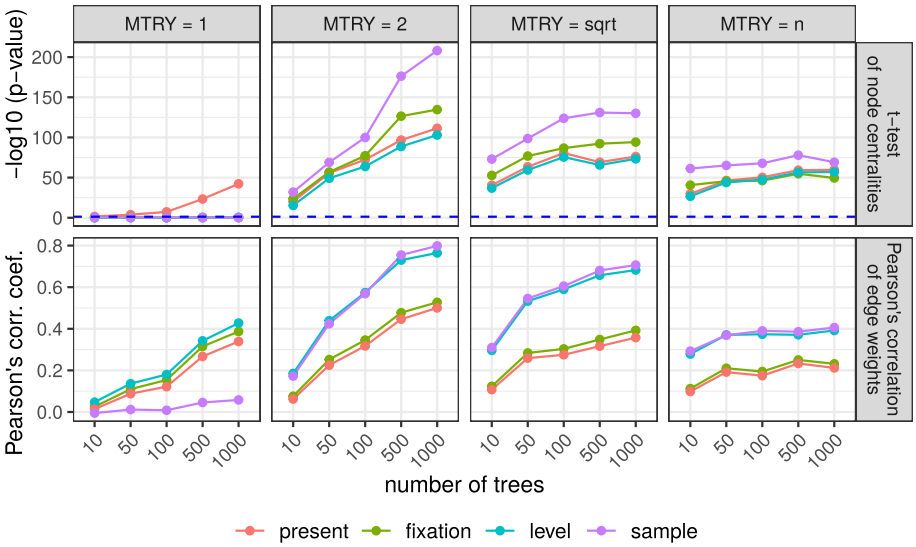


Fig. 4 Hyper-parameter analysis conducted on synthetic datasets, assessing the differences in centrality between relevant and irrelevant features (first row) and the correlation between edge weights and the discriminative power of feature pairs (second row) when varying the number of trees in the forest and the number of features considered at each node for the best split (MTRY). The blue horizontal lines in the first row indicate statistical significance at p=0.05

The edge weight analysis (second row of the figure) further corroborates that the best number of features sampled at each node-with the highest correlation between edge weights and the number of separated clusters-are *sqrt* and 2, with correlation coefficients increasing with the number of trees. Hyperparameter analysis including the minimum terminal node size (Supplementary Figure 1) indicates that bucket sizes of 5 and 10 yield feature graphs with more desirable properties. Based on these observations, subsequent experiments were conducted with feature sub-sampling set to *sqrt*, the minimum terminal node size set to 5, and the number of trees set to 500 or 1000, depending on the dataset complexity.

Cluster-specific feature graphs

Further experiments on synthetic datasets reveal that the proposed cluster-specific scaling factor, used in conjunction with one of the defined edge-building criteria, can generate feature graphs able to discriminate between cluster-specific features, sub-relevant features as well as irrelevant features. For every cluster-specific graph and under each edge-building criterion, the out-degree of the cluster-specific features exceeded that of sub-relevant features, which, in turn, was greater than that of irrelevant features, as illustrated in Fig. 5. T-test confirmed the differences between the three groups of features to be significant for each examined criterion (p-value < 1e-07). The *fixation* criterion emerged as the most adept at isolating the cluster-specific feature, while *sample* proved most effective at distinguishing between relevant and irrelevant features. The proposed approach presents a promising strategy for evaluating cluster-specific feature importance. Its inherent additive nature, which generates cluster-specific graphs that sum to the overall feature graph, also allows to easily compute feature graphs for sets of clusters.

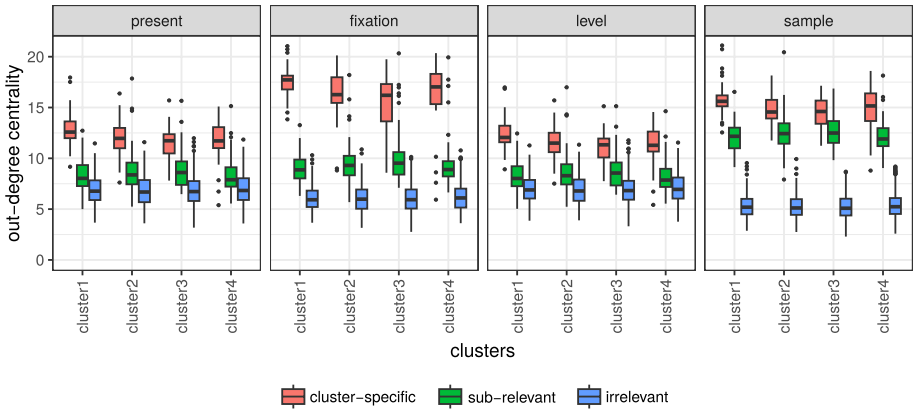


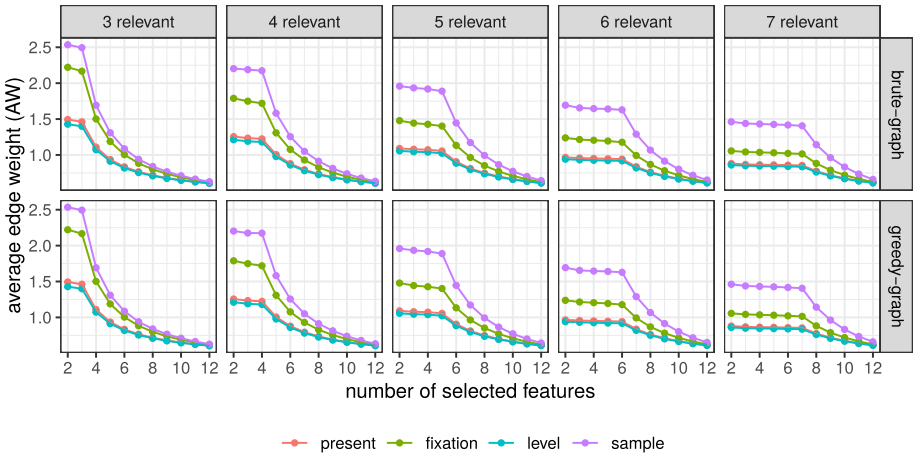
Fig. 5 Evaluating cluster-specific feature graphs

Feature selection on synthetic datasets with relevant features

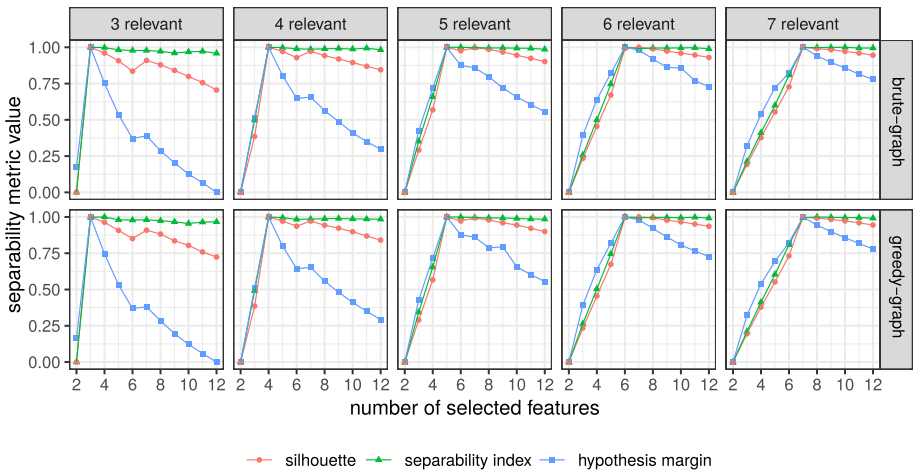
The two proposed strategies for mining the constructed feature graphs, *brute-graph* and *greedy-graph*, both focused on maximising the weight of edges connecting selected features, show promise in identifying the top k features in a clustering task. In experiments on feature graphs generated from synthetic datasets with a varying number of relevant features, both approaches consistently selected all relevant features before any irrelevant ones demonstrating comparable performance, although the brute-force approach has exponential computational complexity.

Evaluating the average weight of the subgraph induced by the selected features also provides valuable insights into the optimal number of features to select. In both graph-mining methods, the inclusion of an irrelevant feature reduces the average weight of the subgraph because relevant features typically connect through heavier edges, whereas irrelevant ones participate in weaker connections. Hence, a notable decrease in the average weight of the subgraph can indicate the inclusion of an irrelevant feature, suggesting that the optimal number of features has been reached. This is supported by results in Fig. 6, where a noticeable decline in the average edge weight occurs right after selecting all relevant features, especially under *sample* or *fixation* edge-building criteria. Separability metrics (shown in Fig. 6 for the *sample* criterion) were also computed for these synthetic datasets to validate the selection of relevant features by each method. As more features are selected with our selection methods, the value or the separability metric increases up until the exact predefined number of relevant features is selected and then it either decreases or plateaus.

These findings were further corroborated in experiments with higher-dimensional datasets of 103 and 503 features (Supplementary Figure 2). The *greedy-graph* approach consistently selected all relevant features before any irrelevant ones, with a drop in average edge weight after all relevant features had been selected, signalling the inclusion of irrelevant features.



(a) Average edge weight.

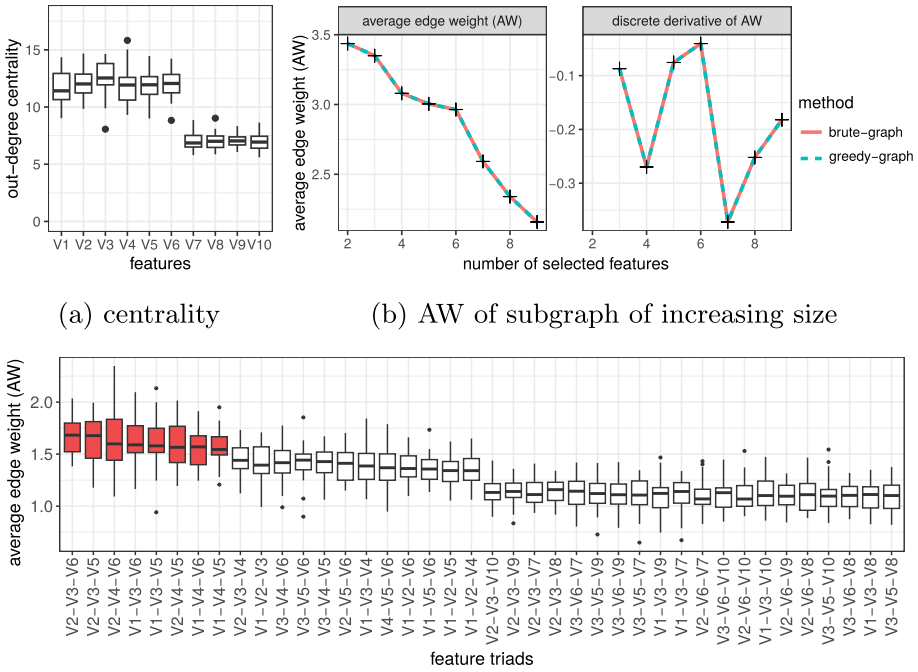


(b) Separability metrics.

Fig. 6 Evaluation of our *brute-graph* and *greedy-graph* feature selection approaches in synthetic datasets comprising average edge weight and separability metrics computed for varying number of features selected by each method

Feature selection on synthetic datasets with repetitive features

An alternative approach for feature selection involves ranking the top k features by their out-degree centrality. However, experiments conducted on synthetic data with redundant features show that the out-degree centrality is comparable for all relevant features, as illustrated in Fig. 7a, and alone it does not assist in identifying effective feature combinations. Conversely, graph-mining strategies evaluating the edge weight of subgraphs enable a more effective selection. Across all synthetic datasets, the heaviest triad consistently corresponded to one of the effective feature combinations, and the eight heaviest triads, on average, were exactly the eight correct feature combinations, as shown in Fig. 7c. When the number of features to select is unknown, exploring the graph structure with *brute-graph* and *greedy-graph* proves insightful. The two methods select the same features and experience two substantial drops in the average



(a) centrality (b) AW of subgraph of increasing size

(c) average edge weight of top 40 triads (8 effective combinations in red).

Fig. 7 Evaluating feature selection strategies in case of redundant features

edge weight of the expanding feature set, as depicted in Fig. 7b: the first after including three features, the minimum required to discriminate all clusters, and the second after reaching six features, the total number of relevant features.

Evaluation on benchmark datasets

The effectiveness of our proposed feature selection methods was validated through a comparative analysis with established techniques across 10 benchmark datasets. The average scores for ARI, NMI, and FMI are summarised in Table 2, alongside results from Wilcoxon signed-rank tests comparing our *greedy-graph* selection to each other method. Our findings show no significant differences between our *greedy-graph* and *brute-graph* methods, suggesting that the greedy variant achieves comparable performance but at a reduced computational cost. *Greedy-graph* demonstrates superior performance to *classification-based* and *phylogeny-based*, significantly outperforming these models on more datasets for all three performance metrics. It exhibits a more comparable performance to the *leave-one-variable-out* method, but still outperforms it with statistical significance 4 vs. 3 times for ARI and FMI, and 6 vs. 3 times for NMI.

The analysis of these metrics across a varying number of features selected, as illustrated in Fig. 8 for ARI, also reveals a higher degree of monotonicity in *greedy-graph* compared to other methods. Monotonicity scores, reported in Table 2, further confirm that *greedy-graph* achieves the highest average monotonicity among all evaluated methods, with statistical significance against *classification-based* and *phylogeny-based* methods. These results suggest that clustering performance with the proposed method tends to increase steadily with the number of features, while performance with the competing

Table 2 Adjusted Rand Index, Normalised Mutual Information, and Fowlkes-Mallows Index computed for our proposed *greedy-graph* and *brute-graph* feature selection methods, as well as *classification-based*, *phylogeny-based*, and *leave-one-variable-out* methods

	Greedy-graph	**Brute-graph**		**Classification-based**			**Phylogeny-based**			**Leave-one–variable-out**		
	mean	**mean**	***p*-value**	**mean**	***p*-value**		**mean**	***p*-value**		**mean**	***p*-value**	
Adjusted Rand Index												
Iris	0.8202	0.8201	1.000	0.8051	1.000		0.8074	1.000		0.8756	0.092	
Liver disorders	0.0328	0.0342	1.000	0.0307	1.000		0.0323	1.000		0.0108	<0.001	↑
Ecoli	0.3565	0.3540	1.000	0.3209	<0.001	↑	0.3420	0.867		0.3687	1.000	
Breast tissue	0.3455	0.3444	1.000	0.3273	<0.001	↑	0.3455	1.000		0.3524	1.000	
Glass	0.2183	0.2163	1.000	0.2047	0.012	↑	0.1921	<0.001	↑	0.1483	<0.001	↑
Wine	0.5778	0.5687	0.131	0.6534	<0.001	↓	0.4279	<0.001	↑	0.5055	<0.001	↑
Lymphography	0.1197	0.1186	1.000	0.0888	<0.001	↑	0.0770	<0.001	↑	0.0950	<0.001	↑
Parkinson	0.2087	0.2035	0.459	0.1028	<0.001	↑	0.0911	<0.001	↑	0.2905	<0.001	↓
Ionosphere	0.1253	-	-	0.0800	<0.001	↑	0.1125	1.000		0.1913	<0.001	↓
Sonar	0.0217	-	-	0.0030	<0.001	↑	0.0198	0.210		0.0432	<0.001	↓
Monotonicity	0.6806			0.6459	<0.001	↑	0.6051	<0.001	↑	0.6500	0.070	
Normalised Mutual Information												
Iris	0.8067	0.8072	1.000	0.7905	1.000		0.7923	1.000		0.8549	0.092	
Liver disorders	0.0295	0.0306	1.000	0.0270	1.000		0.0288	1.000		0.0142	<0.001	↑
Ecoli	0.4388	0.4381	1.000	0.4337	0.069		0.4440	1.000		0.3845	<0.001	↑
Breast tissue	0.5076	0.5092	1.000	0.4984	<0.001	↑	0.5258	<0.001	↓	0.5191	<0.001	↓
Glass	0.3065	0.3086	1.000	0.2942	<0.001	↑	0.2915	<0.001	↑	0.2141	<0.001	↑
Wine	0.5854	0.5824	1.000	0.6668	<0.001		0.4399	<0.001	↑	0.5245	<0.001	↑
Lymphography	0.1157	0.1167	1.000	0.0938	<0.001	↑	0.0974	<0.001	↑	0.1099	<0.001	↑
Parkinson	0.1155	0.1154	1.000	0.1325	<0.001		0.1217	1.000		0.1992	<0.001	↓
Ionosphere	0.1150	-	-	0.0727	<0.001	↑	0.1223	0.065		0.1942	<0.001	↓
Sonar	0.0577	-	-	0.0073	<0.001	↑	0.0169	<0.001	↑	0.0494	<0.001	↑
Monotonicity	0.6975			0.6393	<0.001	↑	0.6229	<0.001	↑	0.6930	1.000	

Table 2 (continued)

	Greedy-graph	Brute-graph		Classification-based			Phylogeny-based			Leave-one–variable-out		
	mean	**mean**	***p*-value**	**mean**	***p*-value**		**mean**	***p*-value**		**mean**	***p*-value**	
Fowlkes-Mallows Index												
Iris	0.8759	0.8799	1.0000	0.8712	1.000		0.8701	1.000		0.9062	1.000	
Liver disorders	0.5349	0.5352	1.0000	0.5302	1.000		0.5344	1.000		0.5488	<0.001	↓
Ecoli	0.5023	0.4998	1.0000	0.4797	<0.001	↑	0.5009	1.000		0.5758	<0.001	↓
Breast tissue	0.4611	0.4595	1.0000	0.4516	0.044	↑	0.4614	1.000		0.4605	1.000	
Glass	0.4558	0.4597	0.1105	0.4452	<0.001	↑	0.4347	<0.001	↑	0.4123	<0.001	↑
Wine	0.7165	0.7168	1.0000	0.7758	<0.001	↓	0.6307	<0.001	↑	0.6798	<0.001	↑
Lymphography	0.4603	0.4575	1.0000	0.4292	<0.001	↑	0.4094	<0.001	↑	0.4360	<0.001	↑
Parkinson	0.7215	0.7186	1.0000	0.6183	<0.001	↑	0.6177	<0.001	↑	0.7410	<0.001	↓
Ionosphere	0.6717	-	-	0.6752	0.145		0.7014	<0.001	↓	0.6774	0.824	
Sonar	0.5987	-	-	0.5298	<0.001	↑	0.5285	<0.001	↑	0.5588	<0.001	↑
Monotonicity	0.6291			0.5855	0.005	↑	0.5722	<0.001	↑	0.5955	0.004	↑

We also report *p*-values for Wilcoxon signed-rank tests with Bonferroni correction, evaluating our *greedy-graph* method against each other methods. Symbols ↑ (↓) indicate whether the *greedy-graph* method outperforms (underperforms) the other method with statistical significance. Missing values indicate cases where feature selection could not be applied due to excessive computation times (> 48 hours)

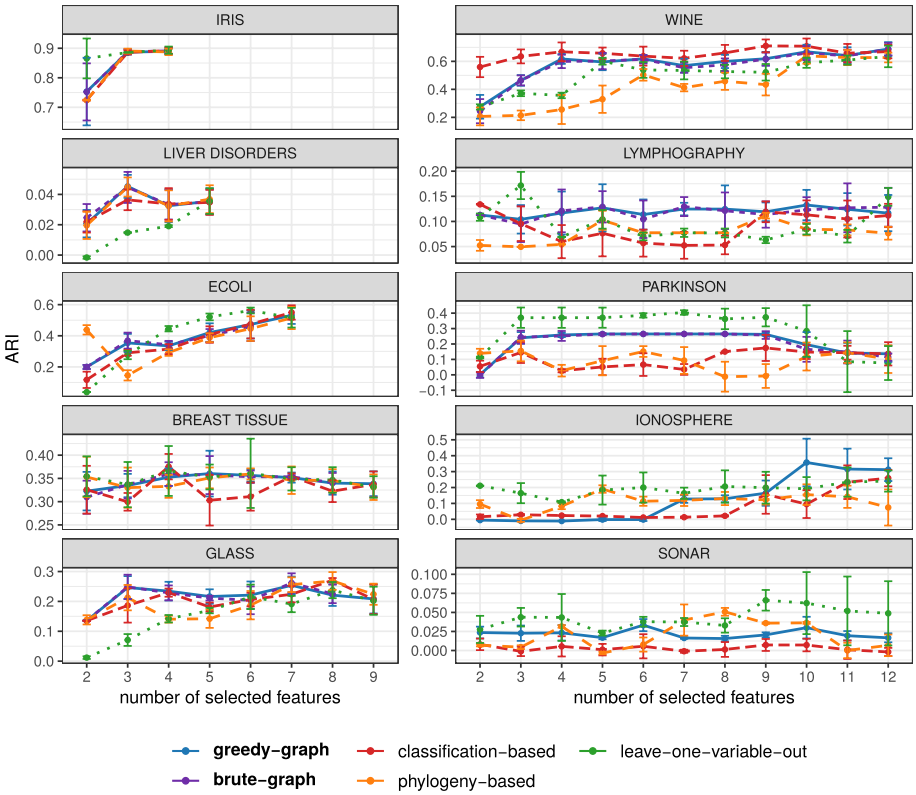


Fig. 8 Clustering performance, measured by the Adjusted Rand Index (ARI), of unsupervised random forests built on the top *k* features (where *k* ranges from 2 to 12) in 5 smaller datasets (left) and 5 larger datasets (right). Features are selected using our proposed *greedy-graph* and *brute-graph* approaches, along with three state-of-the-art methods: *classification-based*, *phylogeny-based*, and *leave-one-variable-out*

methods fluctuates with the number of features. Establishing the optimal number of features is often challenging, and arbitrary cut-offs are commonly applied. The observed monotonicity is an added benefit of the proposed approach, as it mitigates the negative impact of selecting a suboptimal number of features.

Stability evaluation of feature ranks and feature subsets induced by these methods over 30 iterations indicate that *greedy-graph* is the most reliable, with significantly higher Spearman's rank correlation than all other methods, as well as higher Canberra and Kuncheva indices compared to *classification-based* and *leave-one-variable-out*. High stability is another desirable property of feature selection methods, particularly valuable in a biomedical setting, where identifying a consistent subset of features is crucial and testing potential biomarkers is costly and time-consuming (Tables 3 and 4).

Computational times for the five methods are reported in Evaluation on benchmark datasets section. Our *greedy-graph* method consistently yields the fastest computation times, closely followed by the *classification-based* method. In contrast, the *phylogeny-based* and *leave-one-variable-out* report times that are 1 to 2 orders of magnitude greater than our *greedy-graph* for larger datasets. Although our brute-force algorithm is comparable to the greedy variant on smaller datasets, its computational demands escalate with larger datasets due to the exponential growth in clique

Table 3 Stability evaluation of feature subsets using the Kuncheva Index, and feature ranks using the complemented Canberra distance and Spearman's rank correlation for the proposed *Greedy-Graph* (GG) and *Brute-Graph* (BG) methods, as well as the *Classification-Based* (CB), *Phylogeny-Based* (PB), and *Leave-one-Variable-out* (LV) methods

	Kuncheva index					**Canberra distance**				**Spearman's rank correlation**			
	GG	**BG**	**CB**	**PB**	**LV**	**GG**	**CB**	**PB**	**LV**	**GG**	**CB**	**PB**	**LV**
Iris	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0.7977	1.0000	1.0000	1.0000	0.9191
Liver disorders	0.9642	0.9642	0.5840	1.0000	0.7676	0.9897	0.8670	1.0000	0.7267	0.9864	0.7818	1.0000	0.8184
Ecoli	0.9829	0.9829	1.0000	1.0000	0.7608	0.9848	0.8844	1.0000	0.9241	0.9920	0.9856	1.0000	0.8877
Breast tissue	0.9073	0.9073	0.6862	0.8802	0.6074	0.9414	0.7420	0.9350	0.8038	0.9725	0.8061	0.9637	0.7108
Glass	0.9637	0.9483	0.6966	0.9908	0.5323	0.9774	0.8250	0.9974	0.7712	0.9915	0.8350	0.9979	0.6327
Wine	0.9119	0.9100	0.7316	0.8726	0.2847	0.9324	0.8045	0.9118	0.6235	0.9827	0.8914	0.9672	0.2956
Lymphography	0.8838	0.8838	0.6469	0.9047	0.4852	0.9089	0.7782	0.9318	0.7251	0.9813	0.8344	0.9810	0.7482
Parkinson	0.9148	0.9024	0.4856	0.9081	0.1964	0.9632	0.7585	0.9371	0.6750	0.9883	0.5578	0.9893	0.4850
Ionosphere	0.9362	-	0.5952	0.9266	0.0639	0.9691	0.8158	0.9397	0.6554	0.9951	0.7825	0.9848	0.2718
Sonar	0.7814	-	0.3820	0.8340	0.2089	0.9259	0.7208	0.9226	0.7111	0.9879	0.4550	0.9756	0.5668
mean	0.9065	-	0.6289	0.9114	0.3761	0.9593	0.8196	0.9575	0.7414	0.9878	0.7930	0.9859	0.6336
p-value			<0.001	1.000	<0.001		<0.001	1.000	<0.001		<0.001	<0.001	<0.001

Averages across datasets are reported, along with *p*-values from Wilcoxon signed-rank tests with Bonferroni correction, comparing GG against all other methods

Table 4 Average and standard deviation of computation times (in seconds) over 30 iterations for all cases except that marked with an asterisk ([a]), based on 3 iterations

Dataset	Greedy-graph	Brute-graph	Classification-based	Phylogeny-based	Leave-one-variable-out
Iris	5.90 ± 0.38	5.46 ± 0.09	12.71 ± 0.45	26.10 ± 1.87	47.97 ± 3.93
Liver disorders	29.98 ± 0.23	29.00 ± 0.24	78.16 ± 2.80	113.98 ± 7.28	418.67 ± 15.89
Ecoli	15.41 ± 0.22	15.16 ± 0.12	66.98 ± 1.70	123.92 ± 11.87	420.39 ± 0.65
Breast tissue	4.57 ± 0.16	6.77 ± 0.07	6.78 ± 0.12	32.65 ± 3.65	57.09 ± 0.29
Glass	10.17 ± 0.31	12.07 ± 0.11	28.22 ± 0.55	66.23 ± 6.34	242.55 ± 0.52
Wine	7.84 ± 0.14	48.17 ± 0.57	20.41 ± 0.72	67.23 ± 7.50	235.36 ± 0.32
Lymphography	12.31 ± 0.16	1225.08 ± 24.41	12.28 ± 0.81	72.07 ± 9.03	210.52 ± 7.94
Parkinson	25.11 ± 0.36	21234.85 ± 776.63[a]	34.90 ± 2.08	112.03 ± 9.97	686.34 ± 10.71
Ionosphere	20.78 ± 0.23	-	131.05 ± 17.18	283.81 ± 14.39	4347.38 ± 345.35
Sonar	11.70 ± 0.16	-	51.69 ± 1.68	206.54 ± 1.67	2934.55 ± 68.99

Missing values indicate cases with excessive computation times (> 48 hours)

evaluations. For the Parkinson dataset, with 22 features, approximately 10^6 evaluations are made, requiring about 5 hours of computing time. In the Ionosphere dataset, comprising 34 features, 10^{10} evaluations are needed, while for the Sonar dataset, with 60 features, evaluations are 10^{18}. It follows that the algorithm quickly becomes unfeasible for non-small datasets without optimising the clique search.

Overall, our evaluation underscores the competitive accuracy and monotonicity of clustering results over features selected by our *greedy-graph* algorithm. It outperforms both the *phylogeny-based* and *classification-based* methods in terms of monotonicity and quality of clustering solutions. It produces feature ranks and feature subsets that are more stable than in *classification-based* and *leave-one-variable-out*, and it is more computationally efficient than *phylogeny-based* and *leave-one-variable-out*, making it more feasible for high-dimensional datasets. Compared to the *phylogeny-based* approach, which requires a built dendrogram, our method is more versatile, as it can be applied to any clustering technique that uses an affinity matrix, rather than being restricted to hierarchical clustering. Unlike the *classification-based* and *leave-one-variable-out* methods, which require cutting the dendrogram to determine cluster assignments, our approach does not assume prior knowledge of the number of clusters to construct whole-dataset feature graphs. Our graph-based approach offers an additional distinct advantage, as it relies solely on the structure of the unsupervised forest, independent of the clustering algorithm. This provides clearer insights into the features that define the affinity matrix. In contrast, competing methods depend on dendrograms or cluster assignments computed by the clustering algorithm applied to the affinity matrix, making it difficult to disentangle the contributions of features to the unsupervised random forests and the clustering algorithm. Finally, compared to all other methods, our approach offers deeper insights into feature contributions by capturing the effects of feature interactions. Most importantly, it can also construct cluster-specific feature graphs, identifying relevant features for distinct clusters, as we demonstrate in the next section.

Disease subtype discovery

Disease subtyping via clustering employs computational methods to group patients based on shared characteristics, such as clinical features, genetic profiles, or biomarker expression patterns, enabling a deeper understanding of disease complexity and offering potential for personalised medicine and improved patient outcomes [9, 10]. Understanding the varying importance of specific biomarkers and genes within risk clusters is crucial, underscoring the need for interpretability. Certain clusters may exhibit a higher relevance of particular genes or biomarkers, indicating their significance in understanding the risk for those subgroups. Identifying and prioritising these distinctive genetic factors within clusters is essential for devising targeted interventions and conducting personalised risk assessments. However, cluster interpretability is not fully addressed by previous work. The methods outlined in this work aim to bridge this gap by offering enhanced interpretability for unsupervised learning.

In the kidney cancer subtyping application, three patient clusters were identified and further evaluated through an analysis of medical time-to-event patterns: cluster 1 (123 patients), cluster 2 (74 patients), and cluster 3 (11 patients). The survival curves are significantly different among the three clusters with a log-rank p-value of 0.009 (see Fig. 9a). Compared to cluster 1, the risk for a death outcome is significantly higher for patients assigned to cluster 2 and cluster 3. Moreover, Fig. 9b shows the cluster-specific feature graphs, where the edges with weights in the upper 0.95 quantiles are displayed. Interestingly, cluster 1 and cluster 2 consist of similar number of edges with an intersection of only three edges, namely ABCA10-ABCA11P, ABCA13-ZNF826 and ABCA1-ZNF826. This suggests that the remaining interactions are unique to the detected disease subtypes. The feature importance of genes in the feature graphs is displayed in Fig. 9c. There is a notable difference in a gene called ABCA13, which is the most important gene in cluster 2 and cluster 3. In addition, a substantial difference between cluster 2 and cluster 3 can be observed in the genes ABCA11P and ABCA4, which are particularly important for cluster 3. The detected genes are part of the ATP Binding Cassette (ABC) transporters, extensively researched in cancer for their involvement in drug resistance. More recently, their impact on cancer cell biology has gained attention, with substantial evidence supporting their potential role across various stages of cancer development, encompassing susceptibility, initiation, progression, and metastasis [61]. However, further analysis is needed to draw meaningful medical conclusions.

Experiments conducted on a larger subset of genes (500 features, as opposed to 200), are presented in Supplementary Figure 3 and support the observations made in the initial analysis. The top 15 features identified by the method on this larger dataset effectively differentiate clusters, with significantly distinct survival curves for the three clusters (log-rank p-value of 0.04) and cluster assignment in partial agreement with that in the previous experiment. Among the top 15 features detected in the larger dataset, six (ABCA1, ZPBP2, ABCA4, ZNF826, ABCA13, and ABCA10) were also identified in the previous analysis, four of which are ABC transporter genes, recognised as highly relevant in the previous analysis. Feature importance scores confirm the role of ABCA13 in distinguishing clusters, particularly cluster 3. Cluster-specific feature graphs confirm the central roles of ABCA13, ABCA10, and ABCA4, showcasing both shared and unique feature interactions across clusters. Additionally, genes such as ACAN, ACOXL, ACSL6,

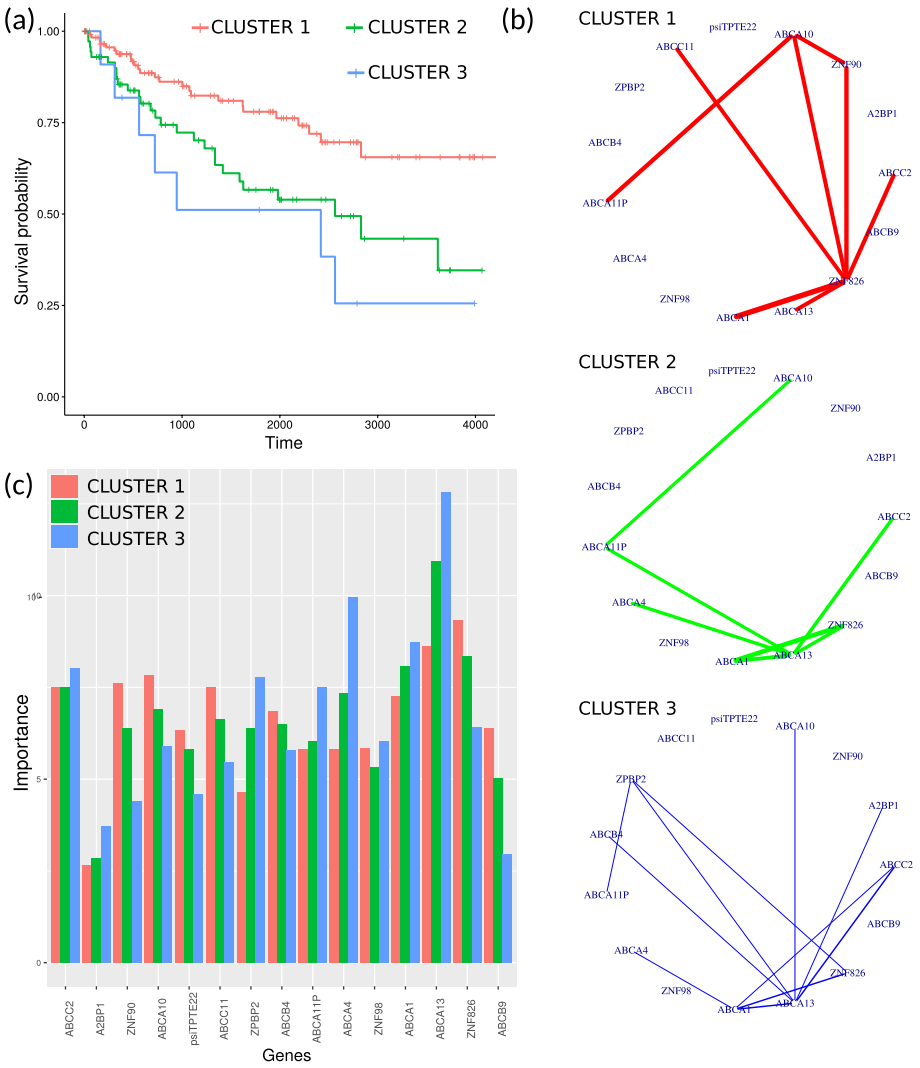


Fig. 9 Application to TCGA kidney cancer data for interpretable disease subtype discovery, preprocessed by selecting the top 250 and bottom 250 features based on variance: **a** survival curves, **b** cluster-specific feature graphs (displaying only edges with weights in the upper 0.95 quantiles), and **c** cluster-specific feature importance for the three identified patient subgroups

and ACCN4, which were not present in the initial subset of 200 features, emerge as important in the larger dataset and should be considered for further investigation.

Limitations of the proposed approach

While the proposed methodology proved effective in identifying relevant features and feature combinations in unsupervised random forests, several limitations are acknowledged. The proposed graph-building method does not directly capture *(a)* longer-range relationships, where pairs of features consistently appear at a certain distance from each other in the trees but are not in immediate proximity, and *(b)* higher-order relationships, involving sets of features that frequently occur in close proximity in the trees. To address the first point, we propose extending our graph-building strategy to account for paths, rather than only considering direct edges. In this extended approach, weights would be assigned to paths of a given length within the trees, and the graphs would be constructed

by aggregating these weights over pairs of features. Graphs built over different path lengths could then be combined. To address the second point, we propose modelling higher-order relationships as hyper-graphs. In this work, we deliberately focused on simple graphs to enhance interpretability and facilitate users in navigating feature relationships. However, when the primary objective shifts from interpretability toward feature selection, incorporating higher-order complexity through hyper-graphs could provide additional insights.

In the graph-mining phase, our greedy algorithm for feature selection assumes that the feature graph is connected to select the best feature from the whole set of available features at each iteration. While we suggested in Mining the feature graph section that increasing the number of trees in the random forest generally ensures graph connectivity, this may not always be feasible when computational resources are constrained and the number of features is very large. In such cases, alternative mining strategies that can efficiently handle disconnected graphs should be developed, and this will be a focus for future research.

Another notable limitation lies in how feature importance is conveyed. Our *greedy-graph* algorithm outputs a feature ranking, determined by the order in which features are selected, rather than an explicit feature importance score. While we suggested that feature centrality could serve as a proxy for importance, our experiments showed that there can be a disconnect between the features selected by the greedy algorithm and their corresponding centrality scores (see Fig. 7). To address this, future work will focus on developing an algorithm that can simultaneously and coherently provide a ranking and a graph-based feature importance score.

Conclusion and future work

This study introduces a novel methodology to enhance the interpretability of unsupervised random forests by elucidating feature contributions through the construction of feature graphs, applicable to both entire datasets and individual clusters. In these graphs, feature centrality reflects their relevance to the clustering task and edge weights indicate the discriminative ability of feature pairs. Additionally, two feature selection strategies are presented: a brute-force approach with exponential complexity and a greedy method with polynomial complexity. Extensive evaluation on synthetic and benchmark datasets demonstrates consistent behaviour between the two methods, suggesting that the greedy approximate algorithm also yields accurate solutions. Evaluation on benchmark datasets indicates that our *greedy-graph* approach outperforms three state-of-the-art feature selection strategies, resulting in superior clustering performance across most benchmarks at a lower computational cost. Importantly, *greedy-graph* exhibits a higher degree of monotonicity in performance, where clustering performance consistently improves with an increased number of selected features, mitigating the risk of selecting suboptimal feature subsets. It also demonstrates superior stability in feature ranks and selected feature subsets across multiple iterations, providing a robust solution for applications such as biomarker discovery, where further testing of relevant features is costly and time-intensive. Additionally, *greedy-graph* stands out for its versatility and computational efficiency, making it particularly suitable for biomedical applications characterised by high-dimensional data. A practical application to disease subtyping illustrates

its ability to offer precise insights into patient stratification through cluster-specific analysis.

Moving forward, there are several avenues for further exploration. First, the graph-building procedure can be enhanced by defining alternative edge-building criteria, for instance, by combining *sample* and *fixation* strategies. The constructed graph offers ample opportunities for exploration through a diverse range of graph metrics, including eccentricity, clustering coefficients, and various centrality metrics. Graph-mining algorithms such as graph search and community detection methods could unveil deeper insights into the structure of the feature graph. Moreover, while the current graph-building procedure only considers splitting features, enriching the graph representation to also include splitting values could provide additional context to the feature interactions captured within the graph. Lastly, the proposed methodologies will undergo systematic evaluation against a broader set of feature selection techniques and across multiple high-dimensional datasets. Their application to a variety of disease subtyping tasks beyond kidney cancer will be thoroughly explored, highlighting broader applicability and utility in diverse biomedical contexts.

Supplementary Information

The online version contains supplementary material available at https://doi.org/10.1186/s13040-025-00430-3.

Supplementary Material 1.

Authors' contributions

B.P. conceived of the project. C.S. developed the methods and conducted the experiments. B.P., M.U., and C.D. wrote the main manuscript text. All authors reviewed the manuscript.

Funding

No funding is reported.

Data availability

The source code for generating the synthetic data is available on GitHub (https://github.com/ChristelSirocchi/urf-graphs). The data used in the application can be found at https://www.linkedomics.org.

Code availability

The code necessary to replicate the experiments presented in the article can be found in the corresponding GitHub repository (https://github.com/ChristelSirocchi/urf-graphs).

Declarations

Ethics approval and consent to participate

Not applicable.

Consent for publication

Not applicable.

Competing interests

The authors declare no competing interests.

Received: 14 November 2024 Accepted: 5 February 2025
Published online: 15 February 2025

References

1. Reddy S. Explainability and artificial intelligence in medicine. Lancet Digit Health. 2022;4(4):e214–5.
2. Ciatto G, Sabbatini F, Agiollo A, Magnini M, Omicini A. Symbolic knowledge extraction and injection with sub-symbolic predictors: A systematic literature review. ACM Comput Surv. 2024;56(6):1–35.
3. Grinsztajn L, Oyallon E, Varoquaux G. Why do tree-based models still outperform deep learning on typical tabular data? Adv Neural Inf Process Syst. 2022;35:507–20.

4. Bicego M, Escolano F. On learning random forests for random forest-clustering. In: 2020 25th International Conference on Pattern Recognition (ICPR). IEEE Xplore; 2021. p. 3451–8.
5. Bicego M. DisRFC: a dissimilarity-based Random Forest Clustering approach. Pattern Recogn. 2023;133:109036.
6. Leng D, Zheng L, Wen Y, Zhang Y, Wu L, Wang J, et al. A benchmark study of deep learning-based multi-omics data fusion methods for cancer. Genome Biol. 2022;23(1):1–32.
7. Cai Y, Wang S. Deeply integrating latent consistent representations in high-noise multi-omics data for cancer subtyping. Brief Bioinforma. 2024;25(2):bbae061.
8. Xie M, Kuang Y, Song M, Bao E. Subtype-MGTP: a cancer subtype identification framework based on Multi-Omics translation. Bioinformatics. 2024;40(6):btae360.
9. Rappoport N, Shamir R. Multi-omic and multi-view clustering algorithms: review and cancer benchmark. Nucleic Acids Res. 2018;46(20):10546–62.
10. Pfeifer B, Bloice MD, Schimek MG. Parea: multi-view ensemble clustering for cancer subtype discovery. J Biomed Inform. 2023;143:104406.
11. Bach S, Binder A, Montavon G, Klauschen F, Müller KR, Samek W. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. PLoS ONE. 2015;10(7):e0130140.
12. Ali S, Akhlaq F, Imran AS, Kastrati Z, Daudpota SM, Moosa M. The enlightening role of explainable artificial intelligence in medical & healthcare domains: A systematic literature review. Comput Biol Med. 2023;166:107555.
13. Bolón-Canedo V, Sánchez-Maroño N, Alonso-Betanzos A. Recent advances and emerging challenges of feature selection in the context of big data. Knowl Based Syst. 2015;86:33–45.
14. Nembrini S, König IR, Wright MN. The revival of the Gini importance? Bioinformatics. 2018;34(21):3711–8.
15. Pfeifer B, Sirocchi C, Bloice MD, Kreuzthaler M, Urschler M. Federated unsupervised random forest for privacy-preserving patient stratification. Bioinformatics. 2024;40(Supplement_2):ii198–ii207.
16. Ismaili OA, Lemaire V, Cornuéjols A. A supervised methodology to measure the variables contribution to a clustering. In: Neural Information Processing: 21st International Conference, ICONIP 2014, Kuching, Malaysia, November 3-6, 2014. Proceedings, Part I 21. Springer Lecture Notes in Computer Science ((LNTCS,volume 8834)); 2014. p. 159–66.
17. Badih G, Pierre M, Laurent B. Assessing variable importance in clustering: a new method based on unsupervised binary decision trees. Comput Stat. 2019;34:301–21.
18. Cabezas LM, Izbicki R, Stern RB. Hierarchical clustering: Visualization, feature importance and model selection. Appl Soft Comput. 2023;141:110303.
19. Ran X, Xi Y, Lu Y, Wang X, Lu Z. Comprehensive survey on hierarchical clustering algorithms and the recent developments. Artif Intell Rev. 2023;56(8):8219–64.
20. Zhang X, Li J, Yu H. Local density adaptive similarity measurement for spectral clustering. Pattern Recogn Lett. 2011;32(2):352–8.
21. Sun L, Guo C. Incremental affinity propagation clustering based on message passing. IEEE Trans Knowl Data Eng. 2014;26(11):2731–44.
22. Azad A, Pavlopoulos GA, Ouzounis CA, Kyrpides NC, Buluç A. HipMCL: a high-performance parallel implementation of the Markov clustering algorithm for large-scale networks. Nucleic Acids Res. 2018;46(6):e33.
23. Rossi F. How many dissimilarity/kernel self organizing map variants do we need? In: Advances in Self-Organizing Maps and Learning Vector Quantization: Proceedings of the 10th International Workshop, WSOM 2014, Mittweida, Germany, July, 2-4, 2014. Advances in Intelligent Systems and Computing ((AISC,volume 295)) Springer; 2014. p. 3–23.
24. Ciriello G, Miller ML, Aksoy BA, Senbabaoglu Y, Schultz N, Sander C. Emerging landscape of oncogenic signatures across human cancers. Nat Genet. 2013;45(10):1127–33.
25. Crippa V, Malighetti F, Villa M, Graudenzi A, Piazza R, Mologni L, et al. Characterization of cancer subtypes associated with clinical outcomes by multi-omics integrative clustering. Comput Biol Med. 2023;162:107064.
26. Kauffmann J, Esders M, Ruff L, Montavon G, Samek W, Müller KR. From clustering to cluster explanations via neural networks. IEEE Trans Neural Netw Learn Syst. 2022;35(2):1926–40.
27. Breiman L. Random Forests. Mach Learn. 2001;45. https://doi.org/10.1023/A:1010933404324.
28. Pfaffel O. FeatureImpCluster: Feature importance for partitional clustering. R package version 0.1.5. 2021. https://CRAN.R-project.org/package=FeatureImpCluster. Accessed 4 Dec 2024.
29. Zhu X. Variable diagnostics in model-based clustering through variation partition. J Appl Stat. 2018;45(16):2888–905.
30. Dy JG, Brodley CE. Feature selection for unsupervised learning. J Mach Learn Res. 2004;5(Aug):845–89.
31. Solorio-Fernández S, Carrasco-Ochoa JA, Martínez-Trinidad JF. A systematic evaluation of filter Unsupervised Feature Selection methods. Expert Syst Appl. 2020;162:113745.
32. Elghazel H, Aussem A. Unsupervised feature selection with ensemble learning. Mach Learn. 2015;98:157–80.
33. Pfeifer B, Baniecki H, Saranti A, Biecek P, Holzinger A. Multi-omics disease module detection with an explainable Greedy Decision Forest. Sci Rep. 2022;12(1):16857.
34. Tian L, Wu W, Yu T. Graph Random Forest: A Graph Embedded Algorithm for Identifying Highly Connected Important Features. Biomolecules. 2023;13(7):1153.
35. Petralia F, Wang P, Yang J, Tu Z. Integrative random forest for gene regulatory network inference. Bioinformatics. 2015;31(12):i197–205.
36. Wang W, Liu W. Integration of gene interaction information into a reweighted random survival forest approach for accurate survival prediction and survival biomarker discovery. Sci Rep. 2018;8(1):13202.
37. Anděl M, Kléma J, Krejčík Z. Network-constrained forest for regularized classification of omics data. Methods. 2015;83:88–97.
38. Dutkowski J, Ideker T. Protein networks as logic functions in development and cancer. PLoS Comput Biol. 2011;7(9):e1002180.
39. Pan Q, Hu T, Malley JD, Andrew AS, Karagas MR, Moore JH. Supervising random forest using attribute interaction networks. In: Evolutionary Computation, Machine Learning and Data Mining in Bioinformatics: 11th European

Conference, EvoBIO 2013, Vienna, Austria, April 3-5, 2013. Proceedings 11. Springer Lecture Notes in Computer Science ((LNTCS,volume 7833)); 2013. p. 104–16.
40. Larsen SJ, Schmidt HH, Baumbach J. De novo and supervised endophenotyping using network-guided ensemble learning. Syst Med. 2020;3(1):8–21.
41. Kong Y, Yu T. forgeNet: a graph deep neural network model using tree-based ensemble classifiers for feature graph construction. Bioinformatics. 2020;36(11):3507–15.
42. Bayir MA, Shamsi K, Kaynak H, Akcora CG. Topological Forest. IEEE Access. 2022;10:131711–21.
43. Cantão AH, Macedo AA, Zhao L, Baranauskas JA. Feature Ranking from Random Forest Through Complex Network's Centrality Measures: A Robust Ranking Method Without Using Out-of-Bag Examples. In: European Conference on Advances in Databases and Information Systems. Springer; 2022. pp. 330–43.
44. Shi T, Horvath S. Unsupervised learning with random forest predictors. J Comput Graph Stat. 2006;15(1):118–38.
45. Zhu X, Change Loy C, Gong S. Constructing robust affinity graphs for spectral clustering. In: Proceedings of the IEEE conference on computer vision and pattern recognition. IEEE Xplore; 2014. p. 1450–7. https://doi.org/10.1109/CVPR.2014.188.
46. Geurts P, Ernst D, Wehenkel L. Extremely randomized trees. Mach Learn. 2006;63:3–42.
47. Criminisi A, Shotton J, Konukoglu E, et al. Decision forests: A unified framework for classification, regression, density estimation, manifold learning and semi-supervised learning. Found Trends Comput Graph Vis. 2012;7(2–3):81–227.
48. Pál D, Póczos B, Szepesvári C. Estimation of Rényi entropy and mutual information based on generalized nearest-neighbor graphs. Adv Neural Inf Process Syst. 2010;23. https://arxiv.org/abs/1003.1954.
49. Wright S. The genetical structure of populations. Ann Eugenics. 1949;15(1):323–54.
50. Bollobás B. Random graphs. New York: Springer; 1998.
51. Xu J, Ni J, Ke Y. A class-aware representation refinement framework for graph classification. Information Sciences. 2024;679:121061. https://doi.org/10.1016/j.ins.2024.121061.
52. Gilad-Bachrach R, Navot A, Tishby N. Margin based feature selection-theory and algorithms. In: Proceedings of the twenty-first international conference on Machine learning. 2004. p. 43.
53. Turney PD. Cost-sensitive classification: Empirical evaluation of a hybrid genetic decision tree induction algorithm. J Artif Intell Res. 1994;2:369–409.
54. Gagolewski M. A framework for benchmarking clustering algorithms. SoftwareX. 2022;20:101270.
55. Solorio-Fernández S, Carrasco-Ochoa JA, Martínez-Trinidad JF. A review of unsupervised feature selection methods. Artif Intell Rev. 2020;53(2):907–48.
56. Ward JH Jr. Hierarchical grouping to optimize an objective function. J Am Stat Assoc. 1963;58(301):236–44.
57. Barbieri MC, Grisci BI, Dorn M. Analysis and comparison of feature selection methods towards performance and stability. Expert Syst Appl. 2024;249, Part B:123667.
58. Hoefflin R, Harlander S, Schäfer S, Metzger P, Kuo F, Schönenberger D, et al. HIF-1α and HIF-2α differently regulate tumour development and inflammation of clear cell renal cell carcinoma in mice. Nat Commun. 2020;11(1):4111.
59. Purdue MP, Rhee J, Moore L, Gao X, Sun X, Kirk E, et al. Differences in risk factors for molecular subtypes of clear cell renal cell carcinoma. Int J Cancer. 2021;149(7):1448–54.
60. Bland JM, Altman DG. The logrank test. Bmj. 2004;328(7447):1073.
61. Nobili S, Lapucci A, Landini I, Coronnello M, Roviello G, Mini E. Role of ATP-binding cassette transporters in cancer initiation and progression. Semin Cancer Biol (Elsevier). 2020;60:72–95.

Publisher's Note