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

Christel Sirocchi[1][0000−0002−5011−3068], Martin Urschler[2][0000−0001−5792−3971], and Bastian Pfeifer[2][0000−0001−7035−9535]

[1] Department of Pure and Applied Sciences
University of Urbino, Italy
[2] Institute for Medical Informatics, Statistics and Documentation
Medical University of Graz, Austria

Abstract. Interpretable machine learning has emerged as central in leveraging artificial intelligence within high-stakes domains such as healthcare, where understanding the rationale behind model predictions is as critical as achieving high predictive accuracy. In this context, feature selection assumes a pivotal role in enhancing model interpretability by identifying the most important input features in black-box models. While random forests are frequently used in biomedicine for their remarkable performance on tabular datasets, the accuracy gained from aggregating decision trees comes at the expense of interpretability. Consequently, feature selection for enhancing interpretability in random forests has been extensively explored in supervised settings. However, its investigation in the unsupervised regime remains notably limited. To address this gap, the study introduces novel methods to construct feature graphs from unsupervised random forests and feature selection strategies to derive effective feature combinations from these graphs. Feature graphs are constructed for the entire dataset as well as individual clusters leveraging the parent-child node splits within the trees, such that feature centrality captures their relevance to the clustering task, while edge weights reflect the discriminating power of feature pairs. Graph-based feature selection methods are extensively evaluated on synthetic and benchmark datasets both in terms of their ability to reduce dimensionality while improving clustering performance, as well as to enhance model interpretability. An application on omics data for disease subtyping identifies the top features for each cluster, showcasing the potential of the proposed approach to enhance interpretability in clustering analyses and its utility in a real-world biomedical application.

Keywords: Feature selection · unsupervised random forest · interpretable machine learning · feature graphs · disease subtyping.

1 Introduction

Interpretable machine learning has become a predominant concern across diverse domains since understanding the reasoning behind model predictions is widely

considered at least as important as achieving high predictive accuracy [28]. In this context, decision trees stand out as interpretable models that offer 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 [13]. This advantage is particularly notable in domains like biomedicine, where datasets are commonly organised in tabular form. Hence, the challenge lies in balancing the interpretability of individual decision trees with the enhanced predictive power achieved through ensemble methods.

As machine learning techniques advance, so does the field of eXplainable AI (XAI) to offer methods for interpretable machine learning, aimed at uncovering the underlying explanatory factors of a black-box model's prediction. XAI has introduced various explanation techniques, ranging from interpretable models that approximate input-output relations to methods that highlight the most influential input features in black-box predictions [3,1]. In this context, feature selection efforts stand out as pivotal for improving model interpretability [6]. Feature selection for enhancing interpretability in random forests has been extensively explored in supervised settings [20], yet its investigation in the unsupervised regime remains limited.

Unsupervised random forests are highly effective in computing affinity matrices, which can be further analysed through clustering techniques, offering great potential for applications within the biomedical domain, particularly in disease subtyping. Disease subtyping employs advanced clustering algorithms to stratify patients based on shared characteristics, such as clinical features and molecular profiles [18]. This approach enables researchers to identify distinct subgroups within a disease, thereby enhancing the understanding of its underlying molecular complexity [27,25], ultimately facilitating personalised medical treatments. In this context, enhancing the largely unexplored interpretability of unsupervised random forests emerges as a critical pursuit, and is the focus of our work.

In this study, we improve the interpretability of unsupervised random forests by presenting a novel method for constructing feature graphs via leveraging the parent-child node splits within the trees. The feature graphs are constructed such that the centrality of features within the graph captures their relevance, and edges between features reflect the effectiveness of the feature pair in discriminating clusters. Feature graphs are constructed for the entire dataset or specific to each cluster, enabling the assessment of feature contributions within each cluster. Additionally, two feature selection strategies are introduced, a brute-force method and a greedy approach, aimed at selecting the top k features in the dataset. The study extensively evaluates the effectiveness of the proposed graph-building and graph-mining methods on both synthetic and benchmark datasets. Feature selection is evaluated on multiple fronts, both in terms of its ability to reduce dimensionality and improve performance by filtering features for model training, and also to enhance model interpretability by identifying the most relevant features for each cluster and overall. An application on omics data for disease subtyping identifies the top features for each cluster, demonstrating

the potential to enhance interpretability in clustering analyses and highlighting its utility in a real-world biomedical application.

The article is structured as follows. Section 2 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 Section 3, 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. Section 4 describes the synthetic datasets generated within the study and the benchmark datasets utilised, as well as the experiments conducted to evaluate the proposed methods. Section 5 presents and discusses the results of the investigation, showcasing a practical application of the method in the context of disease subtyping, and, finally, Section 6 concludes by summarising insights derived from the findings and outlining promising avenues for future research.

2 Previous Work

2.1 Feature importance in unsupervised learning

Unsupervised machine learning plays a predominant role in biomedical research, where clustering assists in grouping patients based on clinical or molecular features, supporting tasks such as disease subtyping or drug discovery for personalised medicine [8]. In disease subtyping, for instance, detected clusters typically consist of patients with different phenotypic characteristics, thereby requiring customised treatment. However, most of the existing disease subtyping methods do not infer the key biomarker characteristics for each cluster. A precise detection and selection of these features could enable clinicians to classify patients based on a small set of markers.

The field of XAI has contributed a plethora of explanation techniques to enhance model interpretability, for instance by identifying the most important input features in black-box predictions. However, research efforts aimed at improving interpretability in unsupervised models have been relatively limited, with a predominant focus on anomaly detection rather than clustering [14]. Most XAI efforts have resorted to recasting the unsupervised problem into a supervised one. For instance, a common approach involves converting the unsupervised model into a functionally equivalent supervised model able to replicate the cluster assignments of the original clustering model [14,19], so that available supervised XAI methods can be applied. However adopting a supervised perspective for an unsupervised problem risks compromising the inherent characteristics of unsupervised learning, such as uncovering hidden patterns or structures in data, particularly if the introduced labels do not accurately represent the underlying data structure. Therefore, the demand for interpretability in unsupervised models remains high, as the main purpose of applying these models is often to better understand the underlying data [19]. Identifying the features that most

contributed to a clustering assignment stands out as one of the most valuable explanations to ultimately provide actionable insights. In the context of disease subtyping, relevant features can help identify signature genes and potential biomarkers, enabling the development of personalised screening tests [9,30].

In unsupervised learning, methods for identifying relevant features have mainly been developed in the context of feature selection. These methods are often categorised based on their interaction with the clustering algorithm [11]. Filter methods for feature selection assess the relevance of each feature using statistical metrics such as correlation or variance, independently of the target variable. Features are then sorted or thresholded, and a subset is chosen for further analysis or model training. Filter methods are computationally efficient and scalable as they assess correlation and dependence among features without involving the model. However, they do not offer insights into how the model utilises features for clustering. In contrast, wrapper methods search for a feature subset such that the clustering algorithm, when trained on this subset, optimises a predefined criterion, thereby shedding light on the importance of features within the adopted clustering scheme [31]. However, wrapper methods pose several challenges, such as the requirement for an appropriate criterion to guide the search and high computational costs due to the need to retrain the model on numerous candidate feature subsets [12]. Additionally, these methods may overlook the context of feature interactions, potentially identifying confounded features that lack direct relevance to the task at hand. The selection of features that, while yielding good performance, do not reflect any meaningful relationship between features and do not support any mechanistic explanation for the problem under study, greatly reduces model interpretability.

To address this issue and to provide meaningful insights into feature importance, several strategies have emerged within the domain of tree ensembles. Some of these methods leverage existing graphs, while others compute feature graphs based on the data at hand. However, to the best of the authors' knowledge, all of these methods are adopted in a supervised setting.

2.2 Leveraging feature graphs with random forests

In the past decade, several studies, particularly within the realm of biomedicine, have leveraged networks 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 refine random forests by adjusting feature bagging (sub-sampling of features used for training each tree) and split feature selection (sub-sampling of features at each node during tree growth), or both. Such approaches proved effective, although they have been exclusively applied in a supervised setting.

First, Dutkowski et al. developed the Network-Guided Forest method, where feature bagging is guided by graph search on a given biological network and,

at each node, the split feature is chosen from the neighbours of the parent node [10]. Pan et al. [22] introduced the Mutual Information Network-guided random forests, which also select split features among the parent node's neighbours in the network, with probabilities proportional to corresponding edge weights. Andel et al. [2] proposed the Network-Constrained Forest (NCF), which begins by sampling a feature, potentially relevant to the studied phenomenon, as a seed and selects the remaining features from a probabilistic distribution over the network, favouring those closer to the seed. Petralia et al. [23] introduced iRafNet, an integrative random forest framework for gene regulatory network inference capable of incorporating information from diverse data sources. iRafNet derives weights from additional data sources such as protein-protein interactions, representing the prior belief of regulatory relationships for a given gene. Subsequently, using expression data as the primary source, random forests are trained to identify genes regulating the given gene, selecting features based on previously calculated weights. Wang et al. [33] presented Reweighted Random Survival Forests (RRSF) for survival prediction, which integrates the topological importance of genes assessed using directed random walk by selecting genes for node splitting based on their topological weight. Larsen et al. [17] introduced Grand Forest, where feature bagging involves selecting a node uniformly at random and growing a subgraph using breadth-first search. Split variable selection ensures that each split variable is chosen only from variables connected to the parent node. Pfeifer et al. [24] proposed Greedy Decision Forest subnetwork detection, which selects disease network modules based on multi-modal node features. The algorithm samples node features through random walks, generating decision trees from the visited nodes. Finally, Tian et al. [32] introduced Graph Random Forest (GRF), where the root node of each decision tree is determined through a data-driven approach, and features in the neighbourhood of any number of hops to the root node are considered to train each tree.

2.3 Building feature graphs from random forests

In the absence of available 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 graph-building approaches have been recently proposed. However, as for the methods presented in the previous section, they have been applied exclusively within supervised settings. Ruaud et al. [29] computed the importance of individual features and their pairwise interactions in tree ensemble models based on the interaction effects between pairs of variables on the response, presenting them as a feature network with the primary aim of enhancing model interpretation. Other recent studies have constructed feature graphs directly from the structure of supervised random forests for various applications. In these studies, each tree is represented as a graph, where features represented are vertices and parent-child relations are edges. These studies also assume that graph representations of decision trees can offer concise, interpretable summaries of relationships among features.

Kong et al. [16] presented a graph extractor module that builds an unweighted directed feature graph from a supervised random forest based on the splitting order of features in decision trees. The graph extractor is added to the graph-embedded deep feed-forward network (GEDFN) model proposed in a previous work [15], which integrates the feature graph as a hidden layer within deep neural networks, enabling the use of GEDFN even when an existing feature graph is not available. The primary aim here was to achieve an informative sparse structure model for n<<p scenarios, as in omics data analysis. Bayir et al. [4] converted each decision tree within a supervised forest model into a weighted directed feature graph and extracted key graph features to obtain a high-dimensional vector representation of each tree. Using TDA Mapper, they transformed these high-dimensional feature vectors into a topological cluster network of trees, with the goal of selecting representative decision trees for constructing smaller and more efficient random forests. Additionally, Cantao et al. [7] 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, generally 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.

3 Methods

3.1 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 the 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 children 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 $\mathrm{I}(v)$ 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 the Gini index are commonly utilised, leveraging the response vector y to inform the splitting. In unsupervised scenarios, recent developments have introduced unsupervised splitting criteria, including a splitting rule inspired by the Fixation Index in population genetics [35,26]. The Fixation Index computes the average pairwise distances between samples within and across groups formed by a node split. 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 is derived by counting the number of times each pair of samples appears in the same leaf node. The affinity matrix serves as input for distance-based clustering methods, such as Ward's linkage method [34,26].

The clustering algorithm 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., $c_i = c_h$.

3.2 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.,

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

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

3.3 Mining the feature graph

The feature graph built on the structure of a random forest trained in an unsupervised manner 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:

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

The constructed graph can also be leveraged to identify subsets of relevant features for feature selection purposes. To this aim, a brute-force and a greedy approach are proposed to identify subsets of vertices connected by heavy weights in the graph. 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-force 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.,

$$\mathrm{TW}(G') = \sum_{\{u_i,u_j\}\in E'} w_{ij} \in W' \quad \mathrm{AW}(G') = \frac{2TW(G')}{|V'|(|V'|-1)}$$

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.,

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

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 feature selection The top k features can 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:

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

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 $k*|E_X|$ operations, which approaches $k*d^2$ operations for dense graphs (which feature graphs often are). Consequently, this method exhibits polynomial computational complexity and can be efficiently implemented.

Algorithm 1 Top k greedy feature selection algorithm

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 [5]. 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.

4 Evaluation

In this evaluation section, we conduct a comprehensive analysis of the methodologies outlined in Section 3. Initially, we assess the effectiveness of the proposed

graph construction methods, which encompass four edge-building criteria. This evaluation entails a comparison across synthetic datasets, examining the correlation between node centrality and feature relevance, as well as the relationship between edge weight and the discriminatory power of feature pairs. While Section 4.1 analyses feature graphs constructed on the entire datasets, Section 4.2 focuses on cluster-specific feature graphs. Subsequently, we evaluate the proposed brute-force and greedy feature selection methods across synthetic datasets. These datasets comprise varying numbers of relevant and irrelevant features in Section 4.3, along with redundant features in Section 4.4. Furthermore, we evaluate the efficacy of the greedy feature selection method for dimensionality reduction purposes on benchmark datasets in Section 4.5. Finally, we demonstrate the utility of greedy feature selection for interpretability on an omics dataset in a disease subtyping application in Section 4.6.

4.1 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 relevant in discriminating clusters should exhibit higher out-degree centrality in the graph, indicating that out-degree centrality effectively captures the relative importance of features; (ii) pairs of features that, used together, are more effective in separating clusters should be connected by edges with greater weight, showing that edge weight accurately reflects the ability of the feature pair to discriminate clusters.

To evaluate these properties across the four edge-building criteria, two sets of synthetic datasets were generated. First, 30 synthetic datasets, each comprising 13 features (3 relevant and 10 irrelevant), were generated to form four clusters. In each cluster, 50 data points were distributed normally around predefined cluster centres, with a standard deviation of 0.2 to ensure well-defined clusters. For relevant features, data points were generated around distinct cluster centres ([1,0,0,0] for $V1$, [0,1,0,0] for $V2$, and [0,0,1,0] for $V3$), while for irrelevant features, data points were generated around the same cluster centres [0,0,0,0]. Then, 30 synthetic datasets of 13 features, 8 relevant and 5 irrelevant, were generated around four cluster centres so that different feature pairs could distinguish a varying number of clusters ranging from 1 to 4. For instance, values for features $V3$ and $V4$ were generated around cluster centres [1,0,0,0] and [0,1,1,1], so two globular clusters can be distinguished in the two-dimensional space defined by the two features, centred in (0,1) and (1,0). Figure 1 illustrates how data was generated so that different feature pairs had different discriminating power.

For these and all experiments conducted in this manuscript, random forests were trained in an unsupervised manner using the fixation index as a splitting rule, employing the uRF library [26]. The size of leaf nodes was set to 5, and the number of features sampled uniformly at random at each node was determined as the square root of the dimensionality of the feature space. Feature graphs were generated according to the four proposed edge-building criteria and evaluated in terms of vertex centralities and edge weights.

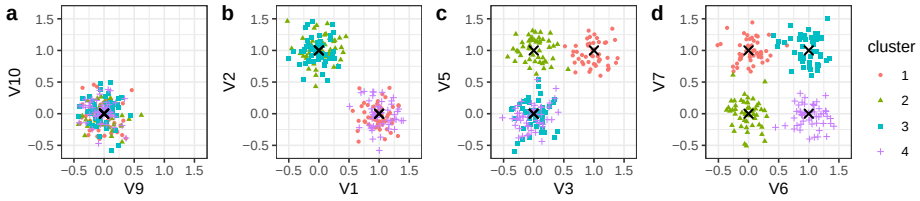


Fig. 1: Feature pairs discriminating (a) 1, (b) 2, (c) 3 and (d) 4 clusters.

4.2 Cluster-specific feature graphs

The subsequent evaluation aims to determine whether the generated cluster-specific feature graphs can effectively capture the role of each feature in discriminating individual clusters. Initially, a collection of 30 synthetic datasets, each containing 13 features, with 4 of them relevant and 9 irrelevant, was generated around four cluster centres. Data for the four relevant features were generated as detailed in Sec. 4.1, with data points distributed around the cluster centres [1,0,0,0], [0,1,0,0], [0,0,1,0], and [0,0,0,1]. This configuration ensured that each relevant feature could contribute to the discrimination of all clusters, while individually, each feature could effectively distinguish one cluster from all others. Subsequently, cluster-specific feature graphs were constructed for each cluster from forests trained on these datasets. Within each cluster-specific graph, the comparison of out-degree centrality was conducted on features categorised into three groups: cluster-specific (the feature capable of singularly identifying the cluster), sub-relevant(features collectively contributing to identifying the cluster), and irrelevant (features unable to discriminate the cluster).

4.3 Feature selection on synthetic datasets with relevant features

The proposed brute-force and greedy selection strategies were assessed on their ability to detect a varying number of relevant and irrelevant features. Synthetic datasets consisting of 13 features, with q relevant features ranging from 3 to 7, were generated around $q+1$ predefined cluster centres. The cluster centres were defined so that the i^{th} relevant feature generated data normally distributed around 1 for the $(i+1)^{th}$ cluster and 0 for the others. For instance, in the case of 3 relevant features, cluster centres were set as $[0,1,0,0]$ for $V1$, $[0,0,1,0]$ for $V2$, and $[0,0,0,1]$ for $V3$. This configuration ensured that all relevant features were necessary and equally important in differentiating clusters. For each configuration, 30 datasets were generated. Feature graphs were constructed from random forests trained on these datasets using the four edge-building criteria. The brute-force and greedy methods were then employed to select the top k features, with k ranging from 2 to 12, while recording the average weight of the subgraph induced by the selected features. The percentage of relevant features among the selected features was computed for each feature graph.

4.4 Feature selection on synthetic datasets with repetitive features

The proposed feature selection strategies for mining the constructed feature graphs were then evaluated on their ability to handle redundant features, by identifying combinations of features that provide complementary information. To validate this claim, synthetic datasets of 10 features, with the first 6 being relevant and the remaining 4 irrelevant, were generated around 4 cluster centres. In these datasets, pairs of relevant features were generated around identical cluster centres, thus providing overlapping information ([1,0,0,0] for V1 and V2, [0,1,0,0] for V3 and V4, [0,0,1,0] for V5 and V6). Consequently, any set of three features including either V1 or V2, V3 or V4, and V5 or V6 (amounting to 8 out of 120 possible feature combinations) can effectively distinguish all clusters. Feature graphs were constructed from random forests trained on these datasets according to the *sample* criteria. The average edge weight of every triad in the graph was evaluated to detect optimal feature combinations.

4.5 Feature selection on benchmark datasets

The proposed greedy feature selection approach, here denoted as the *graph-based* method, was compared against an alternative tree-based strategy, referred to as *impurity-based*. The *impurity-based* method utilises clustering predictions from an unsupervised forest to train a forest in a supervised manner using these predictions as the response variable while assessing feature importance based on impurity criteria. Features were subsequently ranked in descending order of impurity, and those with the highest impurity-based feature importance were selected. The performance of the two feature selection approaches was then compared by evaluating the quality of clustering predictions made by unsupervised random forests trained on the top k features selected by each method. The performance of Ward clustering on affinity matrices derived from the unsupervised random forests was quantified using the Adjusted Rand Index (ARI). Importantly, it should be noted that both feature selection strategies are unsupervised, as neither leverages ground truth information during model training.

The two feature selection approaches were evaluated on the 10 benchmark datasets detailed in Table 1. For each dataset, random forests consisting of 1000 trees were trained 30 times in an unsupervised manner to generate clustering predictions. Feature graphs were constructed from the structure of the forest for each iteration and then averaged to produce a final feature graph for the dataset. Feature importance was determined by computing the out-degree centrality of each feature in the graph. The clustering predictions were utilised to train 30 random forests of 1000 trees in a supervised manner, employing Gini impurity as the measure. The feature importance for each supervised forest was computed using impurity-corrected criteria [20], and the final feature importance was averaged across the 30 forests. The correlation between the feature importance obtained from the unsupervised and supervised approaches was evaluated using Pearson's correlation coefficient. Subsequently, the top k features were identified using the two methods: applying the greedy feature selection algorithm to the

feature graph, as per *graph-based feature selection*, and selecting the k features with the highest Gini impurity scores, according to *impurity-based feature selection*. Then, 30 random forests of 500 trees were trained for clustering on the top k features, with k ranging from 2 to the total number of features, and the quality of the clustering solutions was evaluated by computing the ARI.

Table 1: Benchmark Datasets

Datasets	#Samples	#Features	#Cluster	#Samples per Cluster
Iris	150	4	3	50,50,50
Liver	345	6	2	145,200
Ecoli	336	7	8	143,77,52,35,20,5,2,2
Breast cancer	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

4.6 Interpretable disease subtype discovery

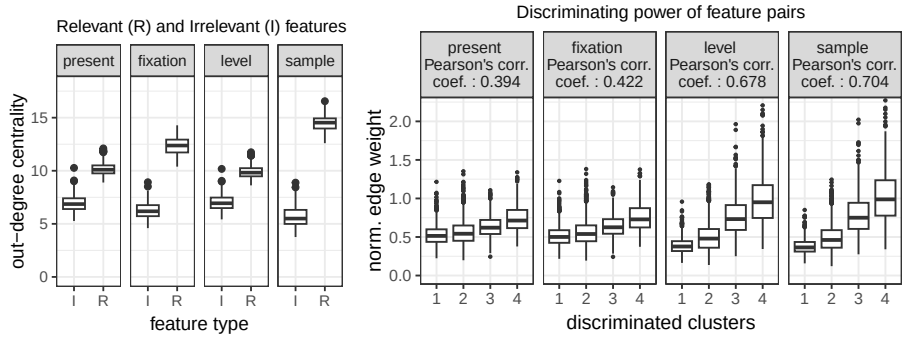
The graph-building and graph-mining methods proposed in this study were applied to disease subtyping. Unsupervised random forests were utilised to derive an affinity matrix from patients diagnosed with kidney cancer, which served as input for a clustering method. Subsequently, the inferred clusters were examined using the proposed methodologies, focusing on genes and gene interactions and their relevance to each patient cluster. Specifically, we demonstrate the practical utility of our approaches using gene expression data obtained from patients suffering from kidney cancer, as provided by [27]. The initial dimensions of the retrieved data table consisted of 606 patients and 20,531 genes. The obtained gene expression patterns were filtered by selecting the top 100 features with the highest variance and the bottom 100 features with the lowest variance. Following this pre-processing step we identified the top-15 relevant genes using the herein proposed greedy selection strategy (see Section 3.3). Based on this gene subset, we performed disease subtyping employing unsupervised random forests as introduced by Pfeifer et al. [26]. We then drew survival curves for the identified clusters and assessed differences using the Cox log-rank test, an inferential procedure for comparing event time distributions among independent (i.e., clustered) patient groups. Finally, we presented findings on genes, gene interactions, and their importance to each patient cluster.

5 Results and discussion

5.1 Evaluation on synthetic datasets

Out-degree centrality and edge weight The first experiment detailed in Section 4.1 reveals promising attributes of the constructed feature graph in terms of centrality and edge weight. Across all four edge-building criteria, the centrality of relevant features is consistently greater than that of irrelevant features, as evidenced in Figure 2 (a). T-tests with a significance threshold of 0.05 confirm that these differences are statistically significant (p-value $<$ e-16). Among all proposed criteria, the *sample* criterion emerges as the most effective in discriminating relevant and irrelevant features, followed by *fixation*. This suggests that the chosen centrality metric can effectively capture the importance of features in solving a clustering task.

The second experiment outlined in Section 4.1 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 Figure 2 (b). 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. These considerations motivate the proposed feature selection strategies.



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

Fig. 2: Evaluating out-degree centrality and edge weights of the feature graphs.

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 Figure 3. T-test confirmed the differences between the three groups of features to be significant for each examined criterion (p-value $<$ e-07). The fixation criterion emerged as the most adept at isolating the cluster-specific feature, while the sample approach 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 relevant sets of clusters.

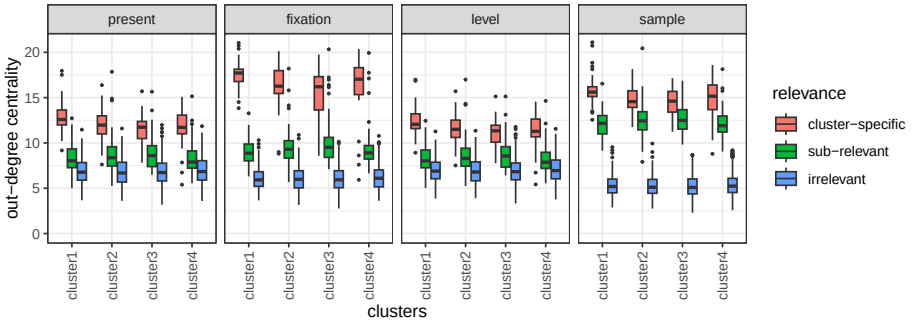


Fig. 3: Evaluating cluster-specific feature graphs.

Feature selection on synthetic datasets with relevant features Two proposed strategies for mining the constructed feature graphs, a brute force and a greedy approach, 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 can also provide valuable insights on the optimal number of features. 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 Figure 4, where a noticeable de-

cline in the average edge weight occurs right after selecting all relevant features, especially under *sample* or *fixation* edge-building criteria.

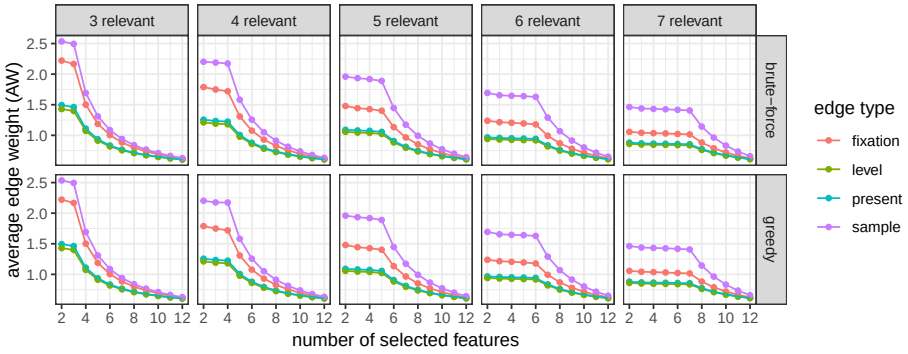
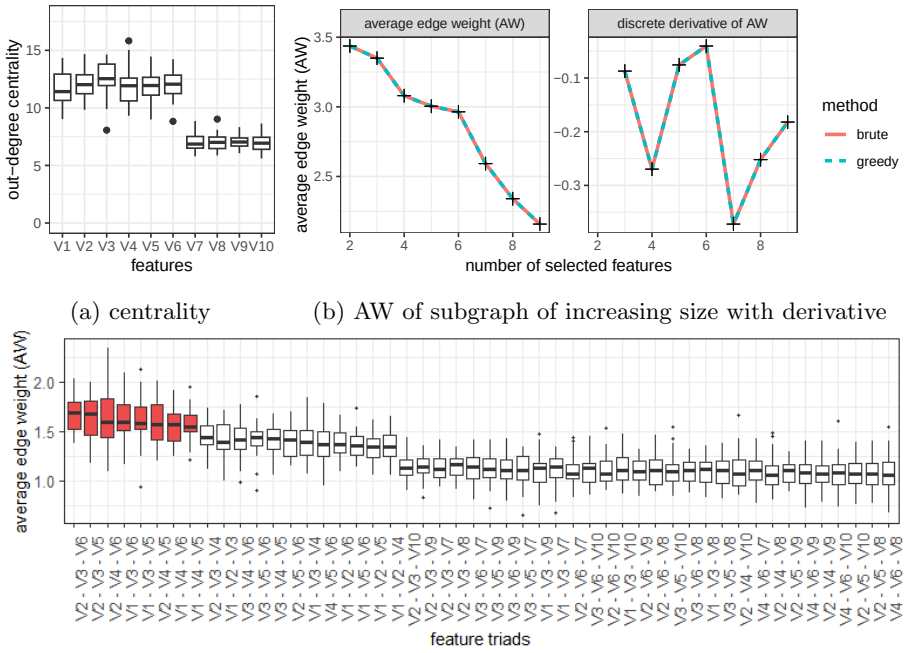


Fig. 4: Evaluating brute force and greedy feature selection approaches.

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 Figure 5 (a), 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 Figure 5 (c). When the number of features to select is unknown, exploring the graph structure with brute force and greedy approaches proves insightful. The two methods select the same features and experience two substantial drops in the average edge weight of the expanding feature set, as depicted in Figure 5 (b): 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.



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

(c) average edge weight of top 50 triads (8 effective feature combinations shown in red).

Fig. 5: Evaluating feature selection strategies in case of redundant features.

5.2 Evaluation on benchmark datasets

The effectiveness of the proposed methods in estimating feature importance and facilitating feature selection was evaluated on clustering (classification) benchmarks. Results in Figure 6 show a partial agreement between the feature importance computed from the feature graph as out-degree centrality and the impurity-based feature importance derived from supervised random forests. Pearson's correlation has positive coefficients across all datasets, with significant p-values for "Breast", "Glass", "Ionosphere", and "Lymphography". Unsupervised random forests trained on the top k features identified by the proposed greedy approach, and the top k features based on impurity-driven feature importance show that clustering performance is comparable for smaller datasets (which consist of a small set of curated features), while it is consistently superior with greedy selection for larger datasets (possibly including irrelevant features). Improved performance is particularly pronounced in datasets such as "Lymphography", "Parkinson", and "Sonar". Furthermore, it is notable that *graph-based feature selection* exhibits a higher degree of monotonicity in performance. Specifically, clustering performance tends to increase steadily with the number of features. In contrast, performance with *impurity-based feature selection* fluctuates with the number of features, as evident in the "Breast" dataset.

Establishing the optimal number of features is often challenging, and arbitrary cutoffs 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.

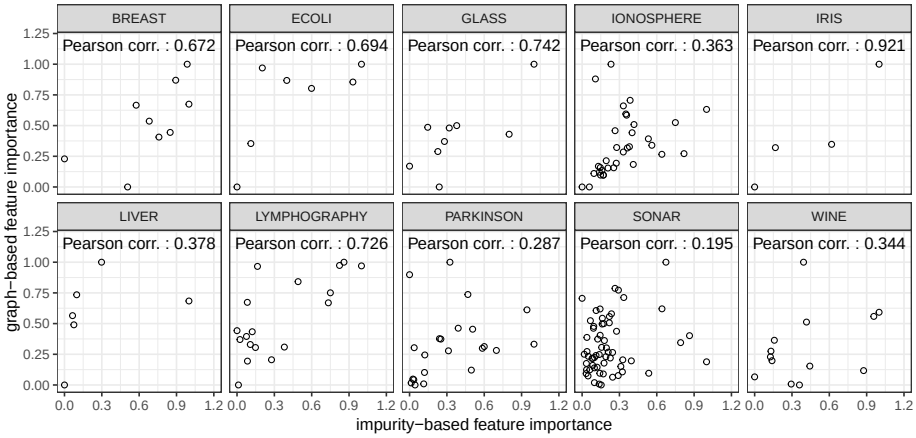
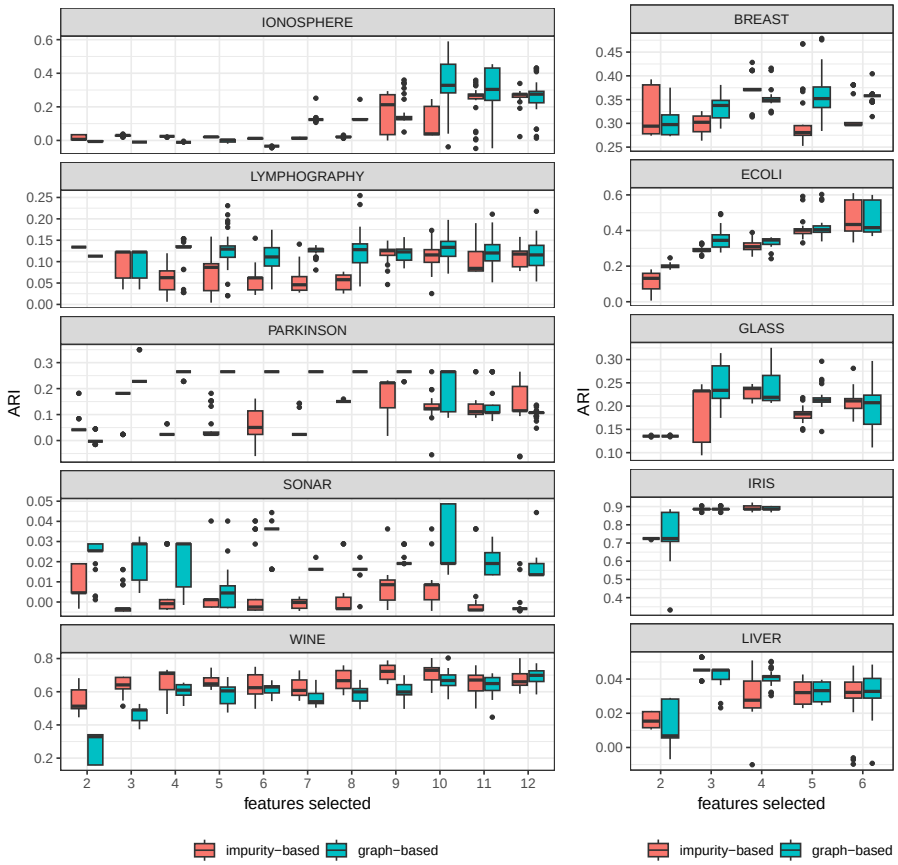


Fig. 6: Correlation between *graph-based* and *impurity-based* feature importance.

5.3 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 [27,25]. 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 associated with 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.

The application on kidney cancer subtyping reveals three patient clusters, which we further analyse in terms of medical time-to-event patterns. The survival curves are significantly different among the three clusters with a log-rank p-value of 0.009 (see Figure 8). Compared to cluster 1, the risk for a death outcome is significantly higher for patients assigned to cluster 2 and cluster 3. Moreover, Figure 8 shows the cluster-specific feature graphs, where the edges



(a) larger datasets (b) smaller datasets

Fig. 7: Clustering performance, evaluated as Adjusted Rand Index (ARI), of unsupervised random forests constructed on k selected features, with k ranging from 2 to 12 for larger datasets and from 2 to 6 for smaller datasets. Features are selected according to *graph-based* and *impurity-based* feature selection.

with weights in the upper 0.95 quantiles are displayed. Interestingly, cluster 1 and cluster 2 consist of the same number of edges with an intersection of only three edges, namely ABCA1-ABCA13, ABCA1-ZNF826, and ABCA13-ABCC2. This suggests that the remaining interactions are unique to the detected disease subtypes. The feature importance of genes in the feature graphs are displayed at the bottom of Figure 8. 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 [21]. However, further analysis is needed to draw meaningful medical conclusions.

6 Conclusion and future work

The study presents a novel method for constructing feature graphs from unsupervised random forests, applicable to both entire datasets and individual clusters. The constructed graphs exhibit desirable properties, with feature centrality reflecting their relevance to the clustering task and edge weights indicating 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 datasets demonstrates consistent behaviour between the greedy and brute-force methods, suggesting that, in the considered experimental setting, the greedy approximate algorithm also yields optimal solutions. Further evaluation on benchmark datasets reveals that the proposed greedy approach is more effective in selecting feature subsets than an impurity-based feature selection strategy, leading to superior clustering performance. Notably, the greedy approach also exhibits a higher degree of monotonicity in performance, with clustering performance increasing with the number of selected features. This added property of the proposed approach can mitigate the negative impact of selecting a suboptimal number of features. Moving forward, there are several avenues for further exploration. First, the proposed procedure for building feature graphs can be enhanced by defining alternative edge-building criteria, for instance, by combining sample and fixation strategies. Furthermore, the constructed graph offers ample opportunities for exploration through a diverse range of graph metrics, including eccentricity, clustering coefficients, and various centrality metrics. Additionally, graph mining algorithms such as graph search and community detection methods could unveil deeper insights into the structure of the feature graph. Finally, 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.

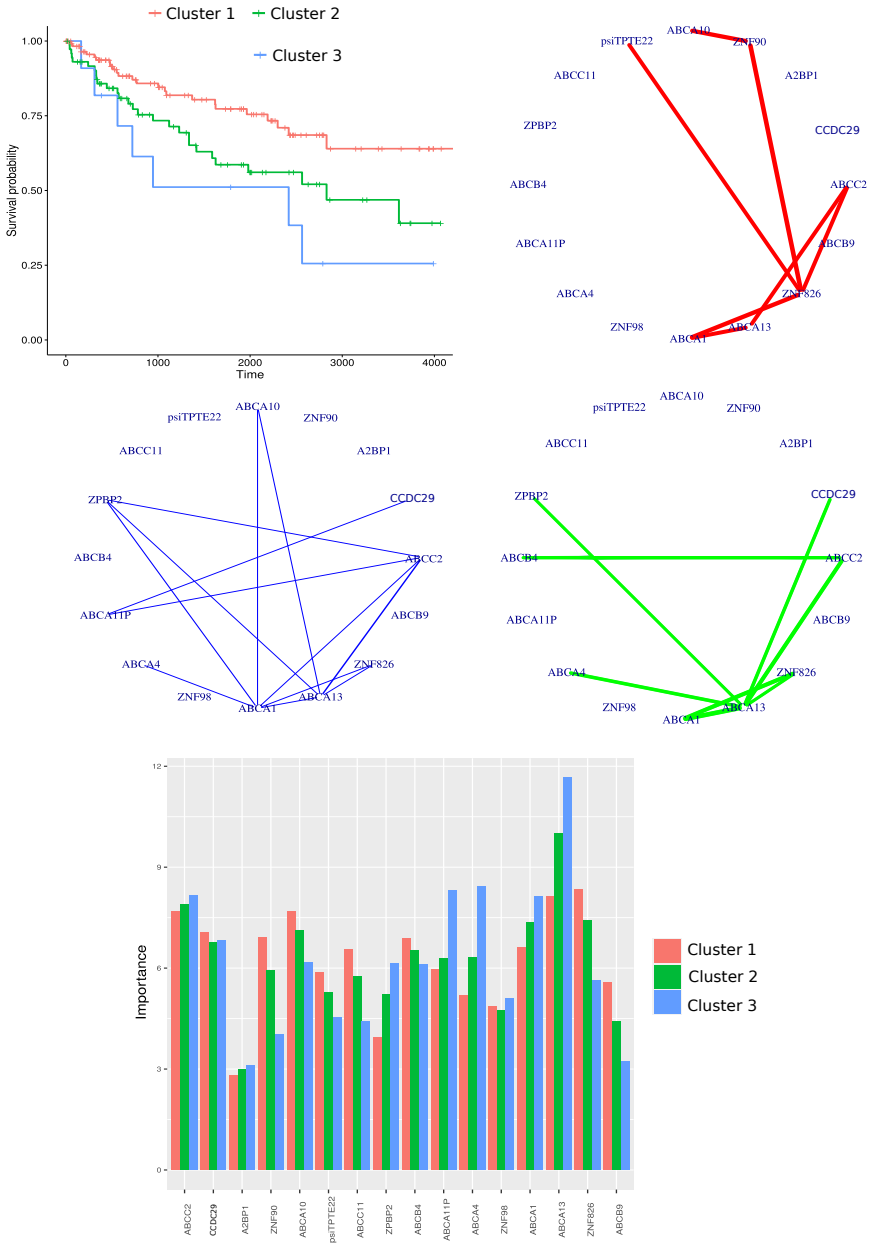


Fig. 8: Application on TCGA data for interpretable disease subtype discovery.

References

1. Ali, S., Akhlaq, F., Imran, A.S., Kastrati, Z., Daudpota, S.M., Moosa, M.: The enlightening role of explainable artificial intelligence in medical & healthcare domains: A systematic literature review. Computers in Biology and Medicine p. 107555 (2023)
2. Anděl, M., Kléma, J., Krejčík, Z.: Network-constrained forest for regularized classification of omics data. Methods **83**, 88–97 (2015)
3. Bach, S., Binder, A., Montavon, G., Klauschen, F., Müller, K.R., Samek, W.: On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. PLoS ONE **10**(7), e0130140 (2015)
4. Bayir, M.A., Shamsi, K., Kaynak, H., Akcora, C.G.: Topological forest. IEEE Access **10**, 131711–131721 (2022)
5. Bollobás, B.: Random graphs. Springer (1998)
6. Bolón-Canedo, V., Sánchez-Maroño, N., Alonso-Betanzos, A.: Recent advances and emerging challenges of feature selection in the context of big data. Knowledge-Based Systems **86**, 33–45 (2015)
7. Cantão, A.H., Macedo, A.A., Zhao, L., Baranauskas, J.A.: 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. pp. 330–343. Springer (2022)
8. Ciriello, G., Miller, M.L., Aksoy, B.A., Senbabaoglu, Y., Schultz, N., Sander, C.: Emerging landscape of oncogenic signatures across human cancers. Nature Genetics **45**(10), 1127–1133 (2013)
9. Crippa, V., Malighetti, F., Villa, M., Graudenzi, A., Piazza, R., Mologni, L., Ramazzotti, D.: Characterization of cancer subtypes associated with clinical outcomes by multi-omics integrative clustering. Computers in Biology and Medicine p. 107064 (2023)
10. Dutkowski, J., Ideker, T.: Protein networks as logic functions in development and cancer. PLoS Computational Biology **7**(9), e1002180 (2011)
11. Dy, J.G., Brodley, C.E.: Feature selection for unsupervised learning. The Journal of Machine Learning Research **5**(Aug), 845–889 (2004)
12. Elghazel, H., Aussem, A.: Unsupervised feature selection with ensemble learning. Machine Learning **98**, 157–180 (2015)
13. Grinsztajn, L., Oyallon, E., Varoquaux, G.: Why do tree-based models still outperform deep learning on typical tabular data? In: Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (2022)
14. Kauffmann, J., Esders, M., Ruff, L., Montavon, G., Samek, W., Müller, K.R.: From clustering to cluster explanations via neural networks. IEEE Transactions on Neural Networks and Learning Systems (2022)
15. Kong, Y., Yu, T.: A graph-embedded deep feedforward network for disease outcome classification and feature selection using gene expression data. Bioinformatics **34**(21), 3727–3737 (2018)
16. Kong, Y., Yu, T.: forgenet: a graph deep neural network model using tree-based ensemble classifiers for feature graph construction. Bioinformatics **36**(11), 3507–3515 (2020)
17. Larsen, S.J., Schmidt, H.H., Baumbach, J.: De novo and supervised endophenotyping using network-guided ensemble learning. Systems Medicine **3**(1), 8–21 (2020)
18. Leng, D., Zheng, L., Wen, Y., Zhang, Y., Wu, L., Wang, J., Wang, M., Zhang, Z., He, S., Bo, X.: A benchmark study of deep learning-based multi-omics data fusion methods for cancer. Genome Biology **23**(1), 1–32 (2022)

19. Montavon, G., Kauffmann, J., Samek, W., Müller, K.R.: Explaining the predictions of unsupervised learning models. In: International Workshop on Extending Explainable AI Beyond Deep Models and Classifiers. pp. 117–138. Springer (2020)
20. Nembrini, S., König, I.R., Wright, M.N.: The revival of the gini importance? Bioinformatics **34**(21), 3711–3718 (2018)
21. Nobili, S., Lapucci, A., Landini, I., Coronnello, M., Roviello, G., Mini, E.: Role of atp-binding cassette transporters in cancer initiation and progression. In: Seminars in Cancer Biology. vol. 60, pp. 72–95. Elsevier (2020)
22. Pan, Q., Hu, T., Malley, J.D., Andrew, A.S., Karagas, M.R., Moore, J.H.: 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. pp. 104–116. Springer (2013)
23. Petralia, F., Wang, P., Yang, J., Tu, Z.: Integrative random forest for gene regulatory network inference. Bioinformatics **31**(12), i197–i205 (2015)
24. Pfeifer, B., Baniecki, H., Saranti, A., Biecek, P., Holzinger, A.: Multi-omics disease module detection with an explainable greedy decision forest. Scientific Reports **12**(1), 16857 (2022)
25. Pfeifer, B., Bloice, M.D., Schimek, M.G.: Parea: multi-view ensemble clustering for cancer subtype discovery. Journal of Biomedical Informatics p. 104406 (2023)
26. Pfeifer, B., Sirocchi, C., Bloice, M.D., Kreuzthaler, M., Urschler, M.: Federated unsupervised random forest for privacy-preserving patient stratification. arXiv preprint arXiv:2401.16094 (2024)
27. Rappoport, N., Shamir, R.: Multi-omic and multi-view clustering algorithms: review and cancer benchmark. Nucleic Acids Research **46**(20), 10546–10562 (2018)
28. Reddy, S.: Explainability and artificial intelligence in medicine. The Lancet Digital Health **4**(4), e214–e215 (2022)
29. Ruaud, A., Pfister, N., Ley, R.E., Youngblut, N.D.: Interpreting tree ensemble machine learning models with endor. PLoS Computational Biology **18**(12), e1010714 (2022)
30. Sarkar, J.P., Saha, I., Sarkar, A., Maulik, U.: Machine learning integrated ensemble of feature selection methods followed by survival analysis for predicting breast cancer subtype specific mirna biomarkers. Computers in Biology and Medicine **131**, 104244 (2021)
31. Solorio-Fernández, S., Carrasco-Ochoa, J.A., Martínez-Trinidad, J.F.: A review of unsupervised feature selection methods. Artificial Intelligence Review **53**(2), 907–948 (2020)
32. Tian, L., Wu, W., Yu, T.: Graph random forest: A graph embedded algorithm for identifying highly connected important features. Biomolecules **13**(7), 1153 (2023)
33. Wang, W., Liu, W.: Integration of gene interaction information into a reweighted random survival forest approach for accurate survival prediction and survival biomarker discovery. Scientific Reports **8**(1), 13202 (2018)
34. Ward Jr, J.H.: Hierarchical grouping to optimize an objective function. Journal of the American statistical association **58**(301), 236–244 (1963)
35. Wright, S.: The genetical structure of populations. Annals of Eugenics **15**(1), 323–354 (1949)