Advanced methods and applications in phylogenetic tree generation and model interpretability

Natàlia Segura Alabart

Natàlia Segura Alabart

Advanced methods and applications in phylogenetic tree generation and model interpretability

DOCTORAL THESIS

UNIVERSITAT ROVIRA i VIRGILI

Tarragona, Spain

2024

Advanced methods and applications in phylogenetic tree generation and model interpretability

DOCTORAL THESIS

Supervised by

Dr. Francesc Serratosa

Dr. Alberto Fernández

Departament d'Enginyeria Informàtica i Matemàtiques

UNIVERSITAT ROVIRA i VIRGILI

Tarragona, Spain

2024

UNIVERSITAT
ROVIRA i VIRGILI

Escola Tècnica Superior d'Enginyeria
Departament d'Enginyeria Informàtica i Matemàtiques
Avinguda dels Països Catalans, 26
43007 Tarragona (Spain)
Tel. +34 977 559 703 `https://deim.urv.cat/`

In Tarragona, July, 2024.

I STATE that the present study, entitled "Advanced methods and applications in phylogenetic tree generation and model interpretability", presented by Mrs. Natàlia Segura Alabart for the award of the degree of Doctor, has been carried out under my supervision at the Department d'Enginyeria Informàtica i Matemàtiques of this university.

Signed by the doctoral thesis supervisors:

FRANCESC D'ASSIS SERRATOSA CASANELLES - DNI 35120787V

Digitally signed by FRANCESC D'ASSIS SERRATOSA CASANELLES - DNI 35120787V
Date: 2024.07.19 09:06:26 +02'00'

Dr. Francesc Serratosa

FERNANDEZ SABATER ALBERTO - 52600994V

Firmado digitalmente por FERNANDEZ SABATER ALBERTO - 52600994V
Fecha: 2024.07.19 12:39:47 +02'00'

Dr. Alberto Fernández

Acknowledgements

First and foremost, I would like to express my deepest gratitude to my supervisors, Francesc and Albert, for the opportunity to work with them over these years. I fondly remember the meetings we had at the beginning of my thesis, where we were trying to define the scope and direction of my research. Those early discussions were both challenging and enlightening, as we navigated through different ideas and perspectives. Their guidance and support have been invaluable, and I am truly thankful for their patience and expertise.

To Elena, my start was your almost end, but you have taught me so much about optimism, courage, and resilience. You are an inspiration that I have followed until now, and I am grateful to have ever met you.

To Yarkin, for our shared countless experiences, challenges, and love for sushi. Your support and companionship have been invaluable, and I am grateful for the journey we have taken together.

To Sarah, because doing a PhD can be lonely at times, even when we're working in the same lab, it's reassuring to know we're both navigating the madness in our heads together. I'm grateful for our weekly meetings where we cheer each other on, exchange ideas, and discuss improvements to our research. Your support has truly made this journey more enjoyable.

To Rodrigo, whose work-life balance and dedication I truly admire. Our weekly meetings became even more enriching with your presence. Your insights and enthusiasm have added so much value to our discussions, making the journey more rewarding.

To Carme and Susana, for our weekly graph meetings. Together with Francesc, Sarah, Rodrigo and Elena over the years, we explored countless topics, each bringing a unique perspective that enriched our discussions and deepened our understanding. Your insights have been invaluable, and I truly appreciate the collaborative spirit we've built together.

Thank you to all of you.

I thank Deirdre Cabooter and Alexander Kensert for allowing me to collaborate with them at University of Leuven (Belgium). The weekly meetings with them were highly inspiring, providing me with new research perspectives and ideas. I also thank all the members of the Pharmaceutical Analysis Research Unit for making me feel welcomed and supported during my mobility stay in Belgium, specially Marie, Emery, Getu, and Kris. I will always cherish the experience. It has not only shaped my research but also created lasting memories and connections that I hold dear.

I also want to thank my family and friends who have been encouraging me during these years.
Gràcies als meus pares Montse i Quim pel seu amor incondicional. Sempre m'heu recolzat en les decisions que he fet al llarg de la vida i meu ensenyat que amb esforç i constància es poden aconseguir coses meravelloses.
I want to thank my sister Maria who has always giving me her strong support. Your constant faith in my abilities and unwavering encouragement have been invaluable to me throughout this journey. I will not be who am I without you, thank you for always being there.
I want to thank my partner Gerard for sharing his life with me. We are achieving our dreams together.

The final acknowledgment I wish to express is to myself. We often overlook our own progress and achievements as we chase the next goal. I want to take a moment to recognize how proud I am of completing this PhD. It has always been a dream of mine, yet I hesitated to fully embrace it out of fear of failure. But I faced the challenge head-on, and through perseverance and dedication, I made it every step of the way. Today, I celebrate this accomplishment with a sense of pride and fulfillment.

Reaching out for help is an act of courage. – *Brené Brown*

This doctoral thesis has received funding from Universitat Rovira i Virgili through grant 2020PMF-PIPF-45 and from European Union through Erasmus+ Program.

Contents

Abstract

Phylogenetic trees are diagrams that represent the evolutionary relationships among various entities, such as biological species, based on the similarities and differences in their characteristics. In these diagrams, nodes represent individual elements or taxa, while branches represent the evolutionary connections between them. In this context, specific distance-based methods need to be defined or analyzed to determine how they create a phylogenetic tree and understand the evolutionary history they provide.

This thesis has two main goals. The first is to investigate different distance -based algorithms applied in the creation and analysis of phylogenetic trees, focusing on the implications of ties in proximity that can lead to ambiguities in binary phylogenetic tree structures. We analyze how these inexact trees affect our understanding of evolutionary relationships. The analysis developed in this thesis demonstrates that ties in proximity hinder the accurate representation of evolutionary histories, potentially misleading interpretations of phylogenetic relationships. Additionally, we propose a new method for generating phylogenetic trees that effectively addresses the ties in proximity problem, thereby enhancing the reliability of evolutionary inference.

The second focus of the thesis is the interpretability of Graph Convolutional Networks (GCNs), which are advanced deep learning models for graph-structured data. Despite their efficacy, GCNs are often criticized for their ”black box” nature, posing challenges in transparency and trust, especially in critical applications like healthcare. Saliency map generators (SMGs) offer post-hoc explanations for GCNs decisions by highlighting key features in the input data. We investigate the effectiveness of these saliency maps and propose metrics to evaluate their performance, thereby enhancing the interpretability of GCNs.

List of Figures

List of Tables

Nomenclature

Symbols

α_{ij}^{kc}	Gradient weights
A	Adjacency matrix
A^k	Feature map in a convolutional layer k
D_{ij}	Distance between OTUs i and j
D_{ik}	Distance between any OTU $i \in I$ and any other OTU $k \notin I$
D_{uk}	Distance between node u and OTU k
D_{uv}	Distance between node u and node v
E	Graph edges
E^c	Explanation map for classification
E_i	Explanation map for regression
G	Graph
I	Set of OTUs
L_{ij}^c	Class specific saliency map
L_{iu}	Branch length linking i and node u
R_{II^C}	Sum of distances between all the OTUs $i \in I$ and all the other OTUs $k \notin I$

R_{iI^C}	Sum of distances between an OTU $i \in I$ and all the other OTUs $k \notin I$
R_{II}	Sum of distances between all the OTUs in I
R_{iI}	Sum of distances between OTU $i \in I$ and all the other OTUs $i' \in I$, $i' \neq i$
R_{IJ}	Sum of distances between pairs of OTUs in I and J
R_{Ik}	Sum of distances between all the OTUs in I and OTU $k \notin I$
R_i	Sum of distances between OTU i and all other OTUs
S_{ij}	NJ computed distance matrix between each pair of OTUs i and j
W	Weight matrix of hidden layer
w_k^c	Weights for a feature map A^k and class c
X	Node feature matrix

Acronyms

CAM	Class Activation Mapping
CI	Coinfidence Interval
GCN	Graph Convolutional Network
MFNJ	MultiFurcating Neighbour-Joining
NJ	Neighbour-Joining
OTU	Operational Taxonomic Unit
ReLU	Rectified Linear Unit
RPLC	Reversed-Phase Liquid Chromatography
SMG	Saliency Map Generator
SSR	Simple Sequence Repeats
STM	Single Step Metrics
STR	Short Tandem Repeats
UPGMA	Unweighted Pair-Group Method with Arithmetic Mean

1
Introduction

This thesis presents two different topics: one is diverse algorithms applied to phylogenetic tree creation and analysis, and the other one is the use of saliency maps and saliency evaluation metrics in graph represented data.

Phylogenetics studies rely on phylogenetic trees to accurately represent evolutionary relationships among elements. A unique phylogenetic tree is crucial for correctly understanding these relationships. However, distance-based methods that produce phylogenetic trees, such as the Unweighted Pair-Group Method with Arithmetic Mean (UPGMA) and Neighbour-joining (NJ), often generate non-unique trees due to tied distances, leading to multiple possible phylogenetic trees for the same data. This lack of a single representation can result in ambiguity and misinterpretations in published research, which affects subsequent studies. Understanding both the impact of tied distances that lead to the creation of non-unique phylogenetic trees and developing methodologies to address these issues is essential. The aim of this thesis is to improve the accuracy and reliability of tree reconstruction algorithms to better handle tied distances, ensuring more consistent results, and to quantify the impact of non-unique phylogenetic trees in published research.

The first topic of this thesis is explained in Chapters 2 to 5. First of all, details about UPGMA and NJ and other fundamental concepts that are needed to understand the next three chapters of the manuscript are explained in Chapter 2. In the next two following chapters, we describe two methods for analyzing the impact of tied elements in phylogenetic trees using hierachical clustering algorithms. Chapter 3 focuses on the UPGMA method, while Chapter 4 examines the NJ algorithm. In Chapter 5, we present a new method using the NJ algorithm that effectively handles tied distances.

Simultaneously, Graph Convolutional Networks (GCNs) are powerful deep neural networls for learning representations from graph-structured data. However, their decision-making processes are often considered a ”black box” due to the intricate and multi-layered nature of their internal operations, which results in a lack of transparency and direct interpretability. This complexity makes it challenging to deduce, understand, and trust how the model arrives at specific decisions or predictions, particularly in real-world contexts that demand a high level of accountability and understanding such as healthcare. Saliency map generators (SMGs) have emerged to address these challenges by providing post-hoc explanations for GCNs. By highlighting the most important features in the input data, SMGs shed light on the decision-making processes of GCNs, improving trust in their predictions. The second aim of this thesis is to analyze the interpretability of graph-based neural networks using saliency maps, SMGs and metrics to evaluate the performance of SMGs.

The second topic of this thesis is explained in Chapters 6 and 7. The details about saliency maps, and other fundamental concepts that are needed to understand Chapter 7 are explained in Chapter 6. In Chapter 7, we describe a method to evaluate the faithfulness of the saliency maps explanations using evaluation metrics created for graph regression data.

At the end of the manuscript, Chapter 8 provides a summary of the overall conclusions for the thesis, and Chapter 9 shows a list of the publications produced during the progress of this thesis.

2

Methods in phylogeny

2.1 Introduction

The study of phylogeny involves understanding the evolutionary relationships between species using various methods. In this chapter we present an overview of the key techniques used in this field, focusing on molecular markers and distance-based clustering algorithms. Molecular markers, which are DNA sequence fragments, such as microsatellites, are highlighted for their effectiveness in identifying genetic variation. These markers provide the data needed to calculate genetic distances, which are then used to construct phylogenetic trees. Distance-based clustering methods, including the Unweighted Pair-Group Method with Arithmetic Mean (UPGMA) and the neighbour-joining (NJ) algorithm, are fundamental for building these trees. Additionally, it addresses the challenge of ties in proximity values, which can result in multiple possible phylogenetic trees, and explores advanced algorithmic solutions to address this issue.

2.2 Molecular markers in phylogenetic studies

Molecular markers are powerful tools to study genetic diversity. They can be used to identify and characterize the genetic variation (different genotypes) within and between species and populations (Ismail et al., 2016). Numerous molecular genetic markers are available for genetic variation studies: isozyme, directed amplification of minisatellite DNA (DAMD), random amplified polymorphic DNA (RAPD), amplified fragment length polymorphism (AFLP), inter-simple sequence repeat (ISSR), restriction fragment length polymorphism (RFLP) and microsatellite markers (Ismail et al., 2016; Williams et al., 1990; Powell et al., 1996). Among the different molecular markers in this manuscript, we will define and utilize microsatellite markers.

Microsatellites, also known as Short Tandem Repeats (STR) or Simple Sequence Repeats (SSR), are short fragments of DNA, between 2 to 6 base pairs, repeated in tandem and randomly inside the genome (Tautz, 1989). They are commonly used because they are highly reproducible, co-dominant and multiallelic molecular markers. Their highly polymorphism allows for precise discrimination between closely related genotypes, and they can be analyzed by a polymerase chain reaction (PCR) assays (Brondani et al., 1998; Ellegren, 2004; Vieira et al., 2016). Microsatellites have been used for clustering tasks, mainly in the Eukaryota domain, from animals (Ebrahimi et al., 2017; Aziz et al., 2020) to plants (Hormaza, 2002)) and fungi (Ates et al., 2019), and in the bacteria domain (Mohammad et al., 2017).

More specifically, microsatellite markers are used to measure the dissimilarity or distance between genotypes as a function of the proportion of shared alleles. Alleles are different forms of a gene found at the same locus (or location) on a chromosome. These markers are successful genetic tools due to the large number of alleles at a specific locus. Therefore, it often happens that different pairs of genotypes are separated by the same distance. For any of these clustering tasks, hierarchical clustering and distance-based methods are frequently used.

2.3 Distance-based methods for phylogenetic trees

Distance-based methods are techniques that utilize the proximity matrix of pairwise similarities or distances between elements to infer evolutionary relationships (Nei, 1978). In evolutionary studies, the graphical structure is represented as a phylogenetic tree to illustrate the evolutionary relationships of the species under study.

In the case of microsatellite markers, the similarity between any two genotypes is measured as the proportion of shared alleles, which can be computed using alternative distances susch as one minus the proportion of shared alleles or minus the logarithm of the proportion of shared alleles. Understanding these genetic similarities is essential for grouping organisms into taxa (singular: taxon), which are defined as groups of populations classified together based on shared characteristics and evolutionary relationships. For example, in a phylogenetic study of mammals, taxa could include species such as *Homo sapiens* (humans) and *Oryctolagus cuniculus* (rabbits).

These distance-based methods are categorized into two main types: hierarchical clustering methods and optimization methods (Lemey et al., 2009). Hierarchical methods like the Unweighted Pair-Group Method with Arithmetic Mean (UPGMA) construct trees by iteratively merging taxa based on average distances, while optimization methods such as neighbour-joining (NJ) seek to minimize evolutionary distances between taxa iteratively to construct phylogenetic trees (Han et al., 2010; Backeljau et al., 1996; Lance and Williams, 1966; Saitou and Nei, 1987; Gascuel and Steel, 2006). More details on UPGMA and NJ will be explained in section 2.3.2 and section 2.3.3, respectively.

2.3.1 Ties in proximity

In exploring genetic distances, it is notable that the number of shared alleles typically ranges between zero and the total number of alleles, but this number is usually relatively small. Consequently, it is not uncommon for distinct pairs of genotypes to have identical distance values. When there are identical similarity values between different pairs, either in the original distances or during the agglomeration process,

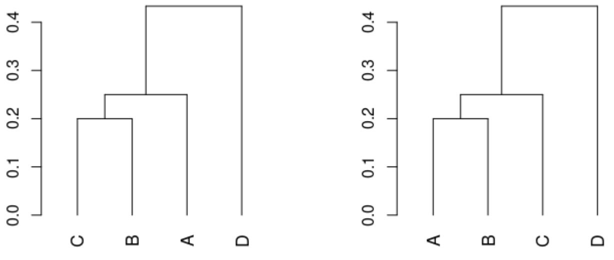


Figure 2.1. Example of two possible NJ phylogenetic trees for the same distance matrix.

phylogenetic tree reconstruction methods can generate more than one structurally different phylogenetic trees.

This problem arises because traditional algorithm creates internal nodes that are always dichotomies. An internal node of a phylogenetic tree is a dichotomy when the tree is rooted and the node is linked to two child subtrees, or when the tree is unrooted and three branches are connected to the node. If more branches are connected to an internal node, then we have a polytomy.

In all these cases in which multiple phylogenetic trees are possible, the reproducibility of the results is more difficult and their interpretation may be biased towards one of several possible solutions (McTavish et al., 2017; Podani, 1997; Segura-Alabart et al., 2022). This algorithmic property is known as the ties in proximity problem (Backeljau et al., 1996; Leal et al., 2016; Hart, 1983; MacCuish et al., 2001). For instance, Figure 2.1 shows an example where more than one phylogenetic tree is possible when the same distance separates genotype A from genotype B, as well as genotype B from genotype C; in this case, genotype B can cluster with either genotype A or genotype C. Additional ties may also appear due to the limited resolution (number of decimal digits) used to store the proximity matrix.

2.3.2 UPGMA algorithm

The UPGMA method is a hierarchical clustering approach that combines step-by-step the closest two clusters or elements into a higher-level cluster, a process also referred as agglomerative clustering. The distance between the new cluster and any other cluster is calculated as the arithmetic mean distance between elements in different clusters (Backeljau et al., 1996; Lance and Williams, 1966).

There are several options to perform hierarchical clustering, developed in different programming environments that return one of the possible binary phylogenetic trees. For instance, in R, there are the *hclust* (hierarchical clustering) function from the *stats* package (R Core Team, 2021b), and the *agnes* (agglomerative nesting) function from the *cluster* package (Maechler et al., 2021). In Python, there are the *AgglomerativeClustering* class from the *scikit-learn* package (Pedregosa et al., 2011), and the *linkage* function from the *scipy* package (Virtanen et al., 2020). And in Matlab, there is the *linkage* function (MATLAB, 2010).

UPGMA explained

The UPGMA algorithm builds a phylogenetic tree from a matrix of evolutionary distances, D_{ij}, between each pair of taxa i, j under study. In a phylogenetic tree, any taxon at a leaf and any internal node is referred to as an operational taxonomic unit (OTU).

The UPGMA algorithm operates iteratively to construct the phylogenetic tree. At each iteration, it selects the two taxa i, j with minimal D_{ij} from the distance matrix. These two taxa are considered to be the most similar or closely related at that iteration.

Let $I = \{i, j\}$ be a pair of selected OTUs with minimal D_{ij}. Then, the taxa i and j are clustered together to form a new internal node u. This new node represents the most recent common ancestor of i and j.

The distance between the new node u and any other OTU $k \neq i, j$ is calculated as follows:

$$D_{uk} = \frac{|x_i|\, D_{ik} + |x_j|\, D_{jk}}{|x_i| + |x_j|}, \tag{2.1}$$

where $|x_i|$ and $|x_j|$ are the number of elements in the cluster i and j, respectively. This weighted average calculation ensures that the new distances take into account the sizes of the clusters being merged. Note that if more than one pair of OTUs have the smallest D_{ij}, only one pair can be selected.

After computing the distances for the new node u, the algorithm updates the distance matrix by removing the distances associated with taxa i and j, and adding the new distances involving node u. This step effectively reduces the size of the distance matrix by one.

The process is repeated iteratively until all OTUs have been merged into a single cluster, resulting in the completion of the phylogenetic tree.

UPGMA variants

It exists several alternative versions of the UPGMA algorithm to try to solve the ties in proximity problem. To name a few, the use of a variable-group algorithm for agglomerative hierarchical clustering that yields a graphical representation known as multidendrogram, where more than two elements or clusters can be grouped when ties occur (Fernández and Gómez, 2007); the exploration of all possible binary phylogenetic trees to create a single phylogenetic tree that considers all possible combinations of elements, clustering them based on these combinations (Arnau et al., 2005); the use of pyramidal clustering, which allows cluster overlapping to create a unique solution by considering multiple levels of aggregation (Diday, 1987; Bertrand, 1995; Nicolaou et al., 2000); or the measure of the likelihood of clusters by counting cluster frequencies in the set of all possible binary phylogenetic trees resulting from ties (Leal et al., 2016).

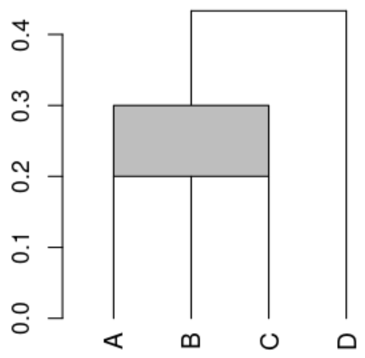


Figure 2.2. Representation of the multidendrogram derived from the data presented in Figure 2.1.

Multidendrogram

The multidendrogram is a variation of the UPGMA algorithm designed to address the ties in proximity problem by merging more than two elements or clusters simultaneously when ties occur during the clustering process.

Suppose that, in a specific iteration, two pairs of OTUs, i_1, i_2 and i_2, i_3, have the smallest D_{ij}; that is, $D_{i_1 i_2} = D_{i_2 i_3} = D_{min}$. In this case, the multidendrogram generates a new internal node u joining the set of three OTUs $I = \{i_1, i_2, i_3\}$.

More generally, let $I = \{i_1, i_2, \ldots, i_P\}$ be a set of OTUs to be clustered together, generating a new internal node u. The distance between the new OTU u and any other OTU $k \notin I$ is:

$$D_{uk} = \frac{\sum_{i \in I} |x_i| \, D_{ik}}{\sum_{i \in I} |x_i|}. \tag{2.2}$$

This equation generalizes the distance calculation from Equation(2.1) used in the standard UPGMA algorithm to handle multiple OTUs simultaneously.

When two new internal nodes u and v join two sets of OTUs $I = \{i_1, i_2, \ldots, i_P\}$

and $J = \{j_1, j_2, \ldots, j_Q\}$, respectively, the distance between the two clusters is:

$$D_{uv} = \frac{1}{|X_I|\,|X_J|} \sum_{i \in I} \sum_{j \in J} |x_i|\,|x_j|\,D_{ij}, \tag{2.3}$$

where $|X_I|$ and $|X_J|$ are the number of elements in the sets I and J, respectively.

This formulation allows the multidendrogram algorithm to effectively address the ties in proximity problem by merging multiple OTUs at once, ensuring that the clustering process is not biased by the order of input data. As a result, the multidendrogram approach provides a more robust and consistent method for phylogenetic tree reconstruction when ties in distances are present.

A graphical representation of a multidendrogram is shown in Figure 2.2. The shadowed region between heights 0.2 and 0.3 represents the interval between the respective values for the clusters. In this example, genotype B has the same minimal distance to both genotype A and genotype C. However, the distance between genotype A and genotype C is not minimal (also visible in Figure 2.1). This hetereogeneity of distances within the same cluster creates an area with varying heigth values. The mininum value corresponds to the smallest distance among all OTUs, while the maximum value corresponds to the distance separating genotype A from genotype C.

2.3.3 Neighbour-Joining (NJ) algorithm

The NJ algorithm is a distance-based method for building phylogenetic trees (Saitou and Nei, 1987; Studier and Kepplter, 1988). NJ is a greedy algorithm that combines the two closest clusters or elements into a parent cluster. When a given distance matrix satisfies the four-point condition, the NJ algorithm finds the correct tree for that distance matrix (Studier and Kepplter, 1988; Durbin et al., 1998; Buneman, 1974; Steel, 1992; Jiang et al., 2001; Cilibrasi and Vitányi, 2005). The four-point condition states that for any four taxa *A, B, C,* and *D*, $d_{AB} + d_{CD} \leq max(d_{AC} + d_{BD}, d_{AD} + d_{BC})$. Even when a given distance matrix does not satisfy the four-point condition, NJ is considered to return a good approximate tree (Atteson, 1999).

Therefore, the NJ algorithm is used by the scientific community to build phylogenetic trees, mainly due to its good accuracy for small to medium sets (Mailund and Pedersen, 2004).

NJ explained

The NJ algorithm constructs a phylogenetic tree based on a matrix of distances between each pair of taxa (organisms or groups). The process begins with all taxa arranged in a starlike tree, assuming no clusters. In each iteration, a score is calculated for each pair of taxa to determine their proximity. This score helps identify the pair of taxa that should be clustered together in the tree in that iteration.

Once the closest pair of taxa is identified, they are grouped together to form a new node. The distances from this new node to all other remaining taxa are then updated. This iterative process continues, with the two closest taxa being removed from the matrix and the new node being added, until only three taxa remain.

When the algorithm reaches the final three taxa, it connects them to complete the phylogenetic tree. This approach allows the NJ algorithm to construct a tree that reflects the evolutionary relationships between the taxa under study, based on their calculated distances. A more detailed explanation will be provided in Chapter 5.

NJ variants

Over the last years, the scientific community has developed several alternative versions of the NJ algorithm. To name just a few, there are algorithms that use heuristics to reduce their running time, making them suitable for large-scale applications: QuickTree (Howe et al., 2002), QuickJoin (Mailund and Pedersen, 2004), relaxed neighbor joining (Evans et al., 2006), and fast neighbor joining (Elias and Lagergren, 2009). Some algorithms try to recover the minimum evolution tree keeping track of several partial solutions along the execution of the algorithm and, thus, exploring a greater part of the tree space: generalized neighbor joining (Pearson et al., 1999), neighbor-joining maximum likelihood (Ota and Li, 2000), and multineighbor joining (Silva et al., 2005). And other possibilities include BIONJ (Gascuel,

1997) and weighted neighbor joining (Bruno et al., 2000), which consider differently long genetic distances than short ones, and MJOIN (Levy et al., 2005), which uses estimates of phylogenetic diversity rather than pairwise distances in the tree.

In order to allow for polytomies, one could use phylogenetic networks that, in spite of no longer trees, can present a unique network for a matrix of evolutionary distances (Bryant and Moulton, 2004). Another possibility could be to modify the NJ algorithm accordingly. As a matter of fact, the NJ algorithm itself is based on the simultaneous partitioning method by (Saitou, 1986), which considers all possible partitions of N OTUs into two clusters with m and n OTUs respectively ($m+n=N; m,n \geq 2$), and selects the best one. Unfortunately, considering all possible partitions into two clusters has the problem of combinatorial explosion (Saitou, 2018).

Several variants of the NJ algorithm can be used to avoid the ties in proximity problem. Live neighbor-joining (Telles et al., 2018) and extended neighbor-joining (Hong et al., 2021) both allow clusters with up to three elements.

3

Nonunique UPGMA phylogenies of microsatellite markers

3.1 Introduction

This chapter examines the occurrence of proximity ties in published UPGMA phylogenetic trees constructed using microsatellite markers, which can result in non-unique phylogenetic trees with multiple possible configurations. Binary phylogenetic trees, constrained to grouping elements in pairs, face challenges when tied distances occur between two or more elements. In these cases, these trees initially group two of the tied elements together, which leads to the other tied elements to be added in a posterior step or grouped in another cluster. In this way, the real genetic relationship between genotypes is not properly reflected in the phylogenetic tree. The presence of UPGMA binary phylogenetic trees with non-unique solutions in published articles can impact not only on the direct conclusions obtained in these publications, but also indirectly on the works based on these original publications.

We analyze the data in publications that had used the UPGMA method in phylogenetic studies on molecular markers using multidendrograms to detect tied distances and count the number of articles where more than one phylogenetic tree was possible. The analysis explained in this chapter has been presented in Segura-Alabart et al. (2022).

This chapter is organised as follows. Firstly, the proposed method is presented and explained in Section 3.2. Secondly, we show the experiments in Section 3.3 and, in the end, we present the conclusions of the chapter in Section 3.4.

3.2 The proposed method

We carried out the analysis of the articles in three steps. First, we set the search strategy by obtaining the population dataset of articles that used the UPGMA clustering method to classify microsatellite markers. Then we selected a sample dataset by filtering and analyzing a subset of the publications searching for specific features in them. Finally, we checked whether the selected articles had ties in their phylogenetic trees and counted the number of articles where the possible phylogenetic trees were non-unique.

3.2.1 Search strategy

We looked for scientific publications that used the UPGMA method on microsatellite markers, up to 2021, in the Scopus database. The search query used to retrieve the titles of articles was: 'UPGMA' AND ('microsatellite*' OR 'simple sequence repeat*' OR 'SSR' OR 'short tandem repeat*' OR 'STR'), where AND and OR are the standard boolean operators. We added the symbol * to some words to include the plural form of these words. We limited the search to words of the query present in the title, abstract or keywords, and the publication year up to 2021. We collected the following bibliometric information: document title, journal and year of publication. A total of 2255 articles had been published from 1995 to 2021 (27 years) and a total of 2239 articles remained after removing 16 duplicated records. That was the population dataset subject of this study.

3.2.2 Sample dataset

Figure 3.1 shows the flowchart of the dataset. We downloaded all bibliometric information corresponding to the selected articles and randomized the dataset to prevent a bias towards a specific year, subject area or alphabet order. Given the large number of publications included in the dataset ($n = 2239$), we selected a subset containing 20% of them. As a result, the initial sample dataset was composed of 454 articles. We excluded 62 articles not available. The remaining subset was composed

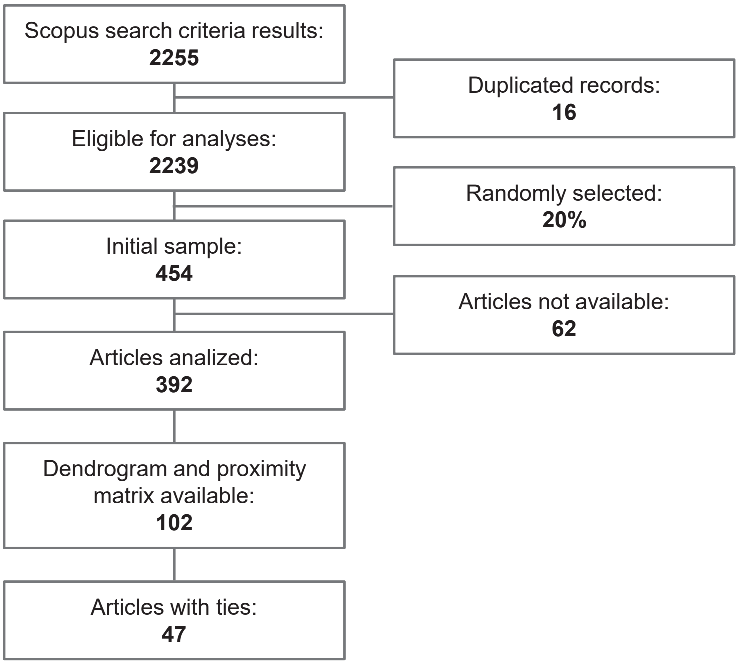


Figure 3.1. Flowchart of the elaboration of the dataset.

of 392 publications to analyze. From this subset, we only selected the articles that contain a phylogenetic tree and a matrix of proximity data, either similarities or distances, or a table describing the genetic profiles of all genotypes. In the cases where the table with the genetic profiles was provided, we computed a matrix with the proportion of shared alleles using the *adegenet* package (Jombart, 2008; Jombart and Ahmed, 2011) in R version 4.1.0 (R Core Team, 2021a). We rejected articles for further analysis if the proximity data matrix and the phylogenetic tree did not contain the same genotype information. In the end, we came up with a final sample dataset containing 102 articles (The complete list of articles, along with additional information, can be found in Segura-Alabart et al. (2022)).

3.2.3 Non-unique phylogenetic trees

We used the *mdendro* package in R to analyze the existence of ties in proximity and to create the corresponding multidendrograms (Fernández and Gómez, 2007, 2020). This package shows the location of any tie in a multidendrogram as a coloured rectangle that represents the variability or range between the minimum and the maximum distances separating any two of the constituent clusters, since it is possible that not all elements in a tie are separated by the same distance (Figure 3.2 A). We also used *Radatools 5.2* (Gómez and Fernández, 2021) to count the number of possible binary phylogenetic trees corresponding to a given matrix of proximity data. *Radatools* has the option of computing all possible binary phylogenetic trees as well as the unique multidendrogram. We chose the former option as we wanted to calculate the number of binary phylogenetic trees a specific article can have when there are tied clusters.

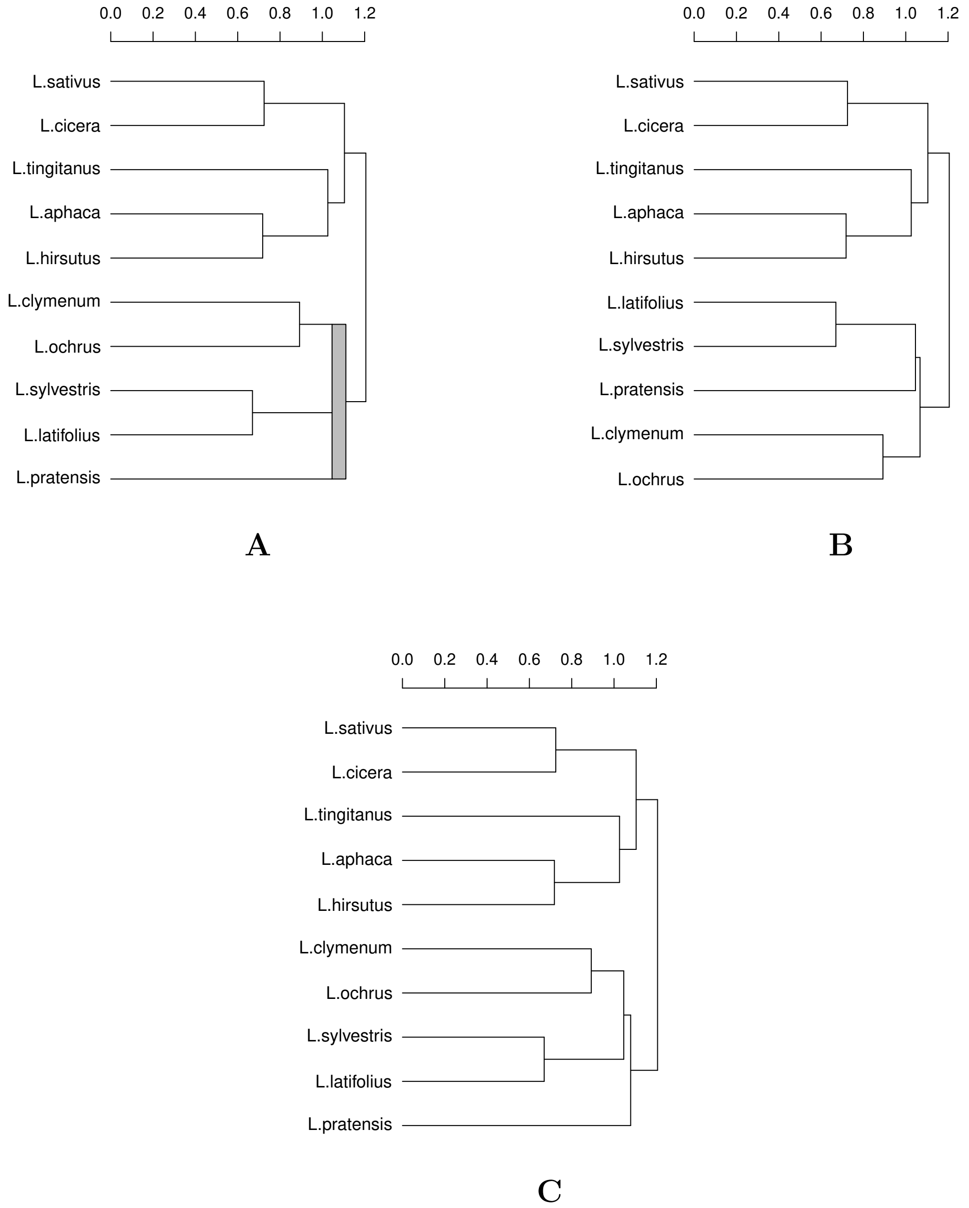


Figure 3.2. Phylogenetic tree of genetic distances between 10 individuals of *Lathyrus sativus* (Wang et al., 2015). (**A**) Multidendrogram created with the *mdendro* package, showing in gray the tied cluster grouping *L. clymenum*, *L. ochrus*, *L. sylvestris*, *L. latifolius* and *L. pratensis*, with a range band between the minimum distance (1.0459 units) and the maximum distance (1.1112 units) between all elements that compose the tie in proximity. (**B, C**) The two possible binary phylogenetic trees, where the last five elements are grouped differently.

3.3 Experiments

This section is divided as follows. First, we quantify the amount of publications that include at least one tie in their clustering process to reconstruct the corresponding phylogenetic tree, and we also asses the potential number of binary phylogenetic trees each article with ties can generate. Then, we examine the yearly number of articles published containing a UPGMA tree of microsatellite markers. We proceed by presenting the subject areas into which the articles are classified. Finally, we show two case studies illustrating different scenarios in binary phylogenetic trees generation: one yielding two possibilities and another yielding over 2.5 million possibilities.

3.3.1 Proportion of articles with ties in proximity

To count the number of publications that had at least one tie in the resulting phylogenetic tree, we took the proximity data from all the articles in our sample dataset and computed the corresponding multidendrogram. We found that in 47 out of the 102 articles analyzed there was more than one possible binary phylogenetic tree. This value corresponds to 46% of the articles, with a 95% confidence interval (CI) between 36% and 56%. Extrapolating this percentage to the total population of 2239 articles gives an estimate of 1032 articles (95% CI 816 – 1248 articles) with alternative solutions in the form of different binary phylogenetic trees. In such cases, employing a single arbitrary resolved phylogenetic tree out of the different possibilities can be misleading.

We were also interested in exploring the distribution of the number of binary phylogenetic trees resulting from the articles that had at least one tie in the resulting phylogenetic tree, see Figure 3.3. Most articles with ties had between 2 and 10 different binary phylogenetic trees (66%, i.e., 31 of all the articles with ties), followed by articles having between 11 and 100 different binary phylogenetic trees (13%, i.e., six of all the articles with ties). Remarkably, 11% of all the articles with ties (i.e., five articles) had more than 10000 different binary phylogenetic trees. These results are in good agreement with previous studies reporting that the occurrence of

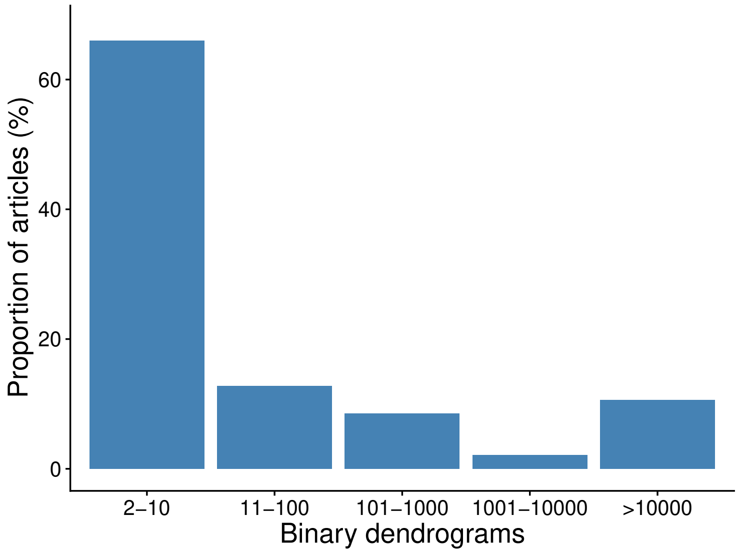


Figure 3.3. Distribution of the number of binary phylogenetic trees resulted from the articles that had at least one tie in the resulting phylogenetic tree ($n = 47$).

ties was responsible for more than one hundred thousand dendrograms (Leal et al., 2016), or even more than seven hundred million dendrograms (Gómez et al., 2013).

3.3.2 Analysis of publications per year

The publication year of the articles in our population dataset ranged from 1995 to 2021 (Figure 3.4). The majority of them were published after year 2000. Since 2009, more than 100 articles have been published yearly; and 2016 is the year with more published articles ($n = 158$) and for the last 10 years the number of publications has stabilized around 140 articles per year. Overall, the number of published articles shows a steady increase since the 2000s, indicating that this research area started to gain considerable attention. The reason for this increase may be 2-fold: on the one side, the existence of next-generation sequencing technologies that started a new era of genomics research with high throughput sequencing data and cheaper sequencing costs (Park and Kim, 2016), and on the other side, software packages to run phylogenetic tree algorithms in general, and the UPGMA method in particular, started to be more readily available at that time.

We are aware that this is just an underestimation of the real number of publications

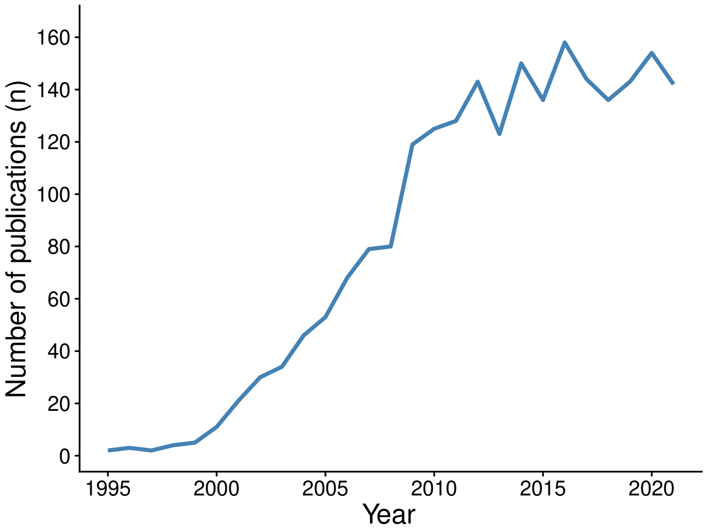


Figure 3.4. Number of articles published in Scopus from 1995 to 2021.

that contain an UPGMA tree of microsatellite markers. This is so because we did not take into consideration articles published in journals outside the Scopus Indexed Journal List. Also, because there are other articles in the Scopus database that contain an UPGMA tree of microsatellite markers, but they do not contain in their title, abstract or keywords, any of the words that we used as search criteria.

3.3.3 Distribution of subject areas

The 2239 articles were classified into 22 different subject areas. The two most common subject areas were Agricultural and Biological Sciences (46%), followed by Biochemistry, Genetics and Molecular Biology (29%) in second place (Figure 3.5). These two subjects constitute 75% of the total number of articles. It was expected that most of the articles were related to biological sciences or similar research areas as STR and SSR are tools frequently used in these areas. We grouped the 13 subject areas that constitute <1% of the total number of articles each into a category named 'Other' in Figure 3.5. For instance, Computer Science (Grishin and Grishin, 2002), Mathematics (Hariri et al., 2017) or Social Sciences (Li et al., 2020) are examples of research areas that are quite distinct from the previous ones. Such a variety of subject areas indicates that the clustering of microsatellite markers by UPGMA is widely used in many areas of scientific knowledge.

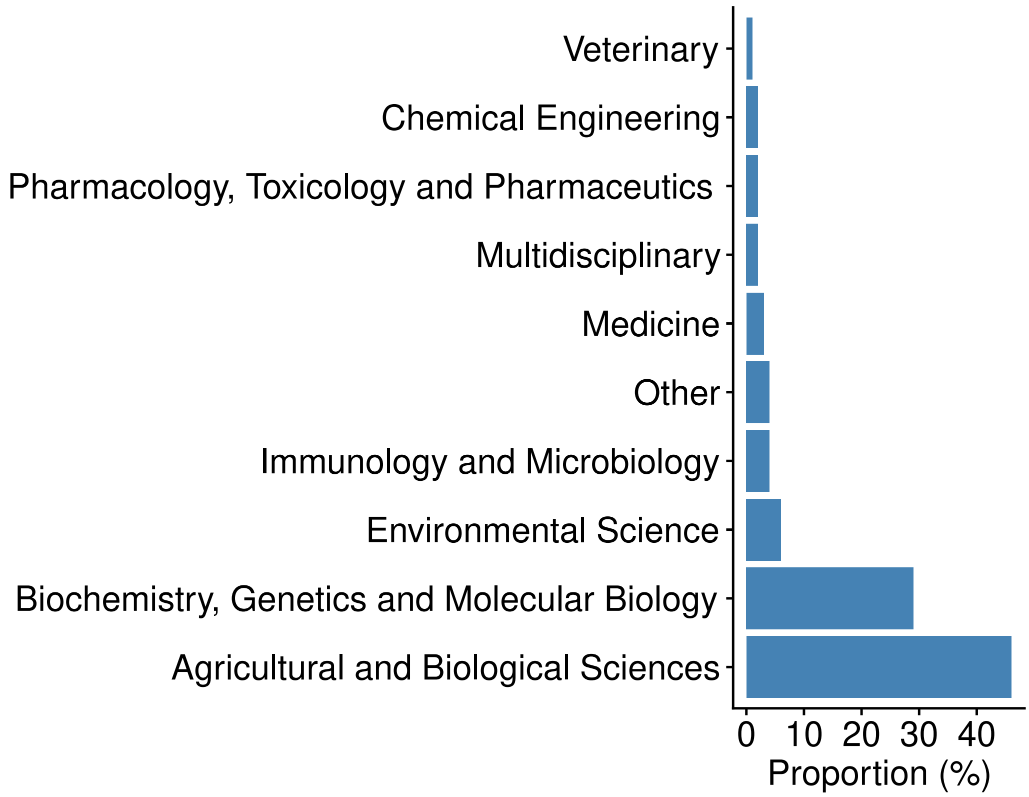


Figure 3.5. Articles classified by subject area. There are 13 subject areas that constitute less than 1% each, and they have been grouped together in the category named "Other".

3.3.4 Case studies

Among the 102 articles that we have analyzed from our sample dataset, we have selected two opposite cases to demonstrate how it is possible to obtain multiple different phylogenetic trees from the same dataset using the same clustering algorithm (UPGMA). The first example describes a case that generates 2 different binary phylogenetic trees, whereas the second example describes a case that generates more than 2.5 million different binary phylogenetic trees.

In the first case study, the authors analyze the genetic diversity among *Lathyrus sativus* (grasspea), also known as *L. sativus*, from its cultivated and wild relatives (Wang et al., 2015). The study has a total of 10 taxa, the number of microsatellite loci used is 30, and the distance matrix values have an accuracy of four decimal digits. The distance matrix values range from 0 to 2. The original data presents a tie between *L. pratensis* and two clusters of grasspea: one formed with *L. clymenum* and *L. ochrus*, and the other with *L. sylvestris* and *L. latifolius*. The corresponding multidendrogram is shown in Figure 3.2A, where the minimum and maximum distances between all cluster elements are 1.0459 and 1.1112, respectively. This tie is responsible for two different binary phylogenetic trees using the UPGMA method. We can describe this tie as happening in the middle of the phylogenetic tree as it is formed by taxa already in clusters. After its formation, the tied cluster will be grouped with the other five elements in the phylogenetic tree. This case clearly shows that tied distances can happen in any step of the clustering process.

In Figure 3.2B, we can observe one of the two possible binary phylogenetic trees, clustering first *L. pratensis* with the pair formed by *L. sylvestris* and *L. latifolius*. Then, a second cluster formed with *L. clymenum* and *L. ochrus* is added to the previous cluster containing three elements. This binary phylogenetic tree shown in Figure 3.2B is exactly the same that the authors of the study gave in their article in Wang et al. (2015). In Figure 3.2C, it is depicted the other possible binary phylogenetic tree for the same input data, where *L. clymenum* and *L. ochrus* are clustered first with *L. sylvestris* and *L. latifolius*. A fifth element, *L. pratensis*, is added then to the previous cluster containing four elements.

In the second case study, the authors analyze the genetic diversity of 22 chillies (*Capsicum annuum L.*) germplasm using four microsatellite markers (Hossain et al., 2014). The article provides a proximity matrix of similarity values with an accuracy of three decimal digits. The similarity matrix values range from 0 to 1. The original data presents three ties along with the resulting multidendrogram (see Figure 3.6A). These three ties are responsible for more than 2.5 million different binary phylogenetic trees using the UPGMA method (to be exact, 2655193 different binary phylogenetic trees). This second case study is a clear example that multiple ties can occur in the same phylogenetic tree. Note that the larger the data set, the more likely it is to have different binary phylogenetic trees (MacCuish et al., 2001).

In Figure 3.6, we can also observe two possible binary phylogenetic trees for the 22 chillies among the more than 2.5 million possibilities. The two selected phylogenetic trees have several remarkable differences between them. One clear difference, for instance, is that in Figure 3.6B *Comilla* is first clustered with *Sada_gol*, and the resulting cluster is merged with *Ruma*. On the contrary, in Figure 3.6C *Comilla* is first clustered with *Dhani*, and the resulting cluster is merged with *Sada_gol*. An even more outstanding difference is found between clusters (*Angoor*, *Shada*) and (*Boro*, *BD.2025*), that are directly clustered together in Figure 3.6C, whereas they only join at the root of the phylogenetic tree (minimum ultrametric similarity) in Figure 3.6B.

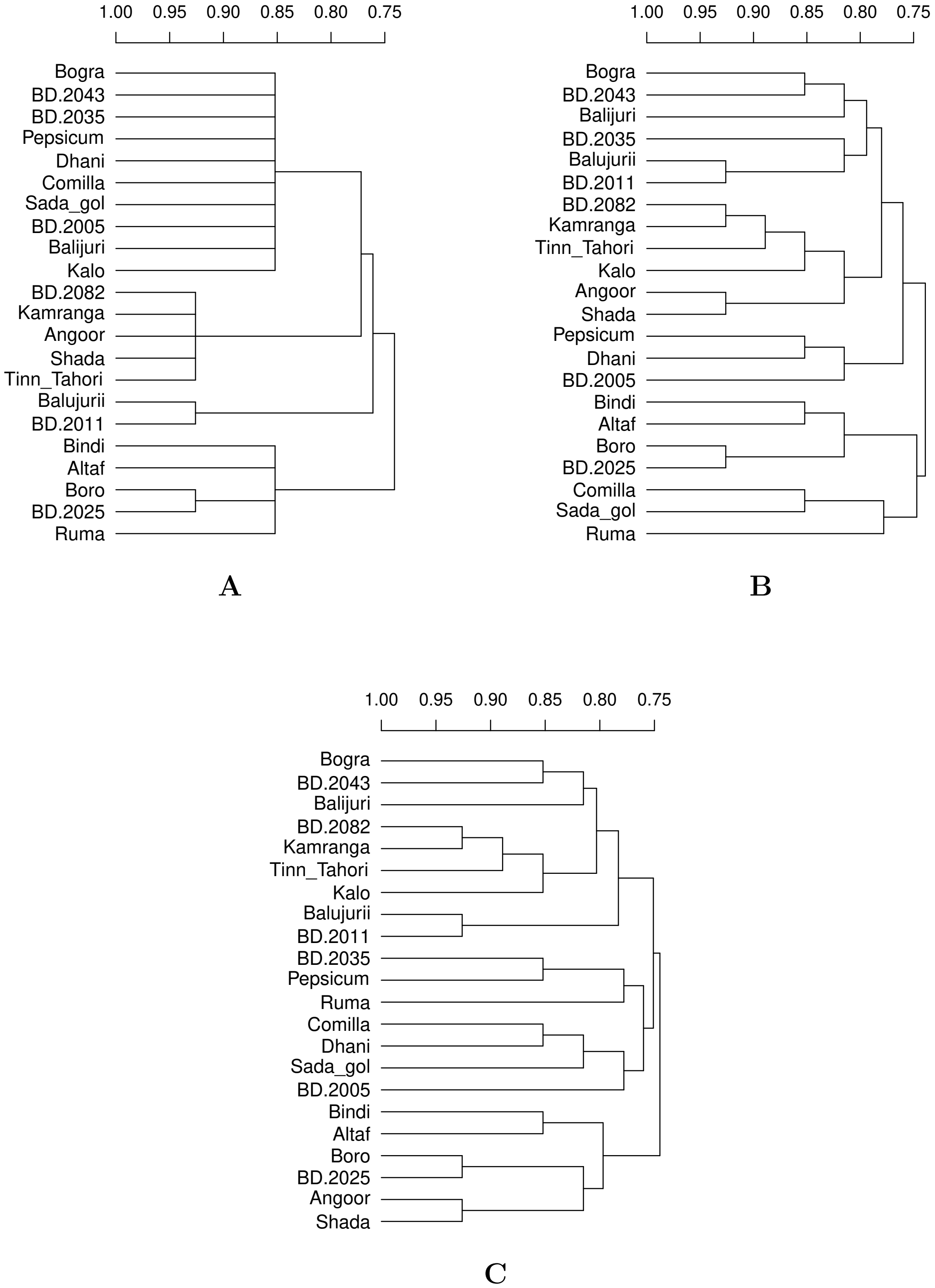


Figure 3.6. Phylogenetic tree of genetic similarity between 22 individuals of *Capsicum annuum L.* Hossain et al. (2014). (**A**) Multidendrogram showcasing the three different ties as a line joining more than two clusters, instead of a range band, for the sake of clarity. (**B, C**) Two possible binary phylogenetic trees among the more than 2.5 million available.

3.4 Conclusions

The method presented in this chapter analyzes the impact of tied elements in phylogenetic trees using the UPGMA algorithm.

The experimental validation shows that the 46% of the articles have at least one alternative solution to the published binary phylogenetic tree. In the dataset of 2239 articles, this would correspond to 1032 articles having at least one tie.

The potential implications that this finding uncovers need to be taken seriously into consideration because between one-third and up to one-half of the articles under consideration are affected by the ties in proximity problem.

While most articles with at least one tie had between 2 and 10 possible binary phylogenetic trees, there are instances where the number can exceed 10000 and even reach up to 2.5 million different binary phylogenetic trees.

The existence of articles containing UPGMA binary phylogenetic trees that are not unique solutions can have consequences not only on the direct conclusions obtained in these publications, but also indirectly on the works based on these original publications.

Ties in proximity affect more fields than the one analyzed here. We have shown that ties are not exclusive of biological sciences or similar research areas; instead, they can also occur in completely different research areas. Thus, such a wide range of research topics affected by ties is not exclusive of microsatellite data and experiments. Note that this problem is inherent in the methodology used to obtain binary phylogenetic trees.

4

A practical study of the proportion of non-unique neighbour-joining trees of microsatellite markers

4.1 Introduction

This chapter examines the non-uniqueness of the NJ algorithm when there are tied distances in the data. As mentioned before, NJ is an algorithm for reconstructing binary phylogenetic trees, which clusters two elements in each step, independently of whether there are more than two elements with the same distance. As a result, it cannot return multiple binary phylogenetic trees despite their possible existence, as it solely produces a single phylogenetic tree. This limitation went unnoticed by the authors of the publications in question. Consequently, the results and conclusions in published scientific papers that present a single phylogenetic tree generated by the NJ method may be biased or limited in scope.

We quantify the magnitude of the ties in proximity problem by conducting a statistical study of publications that contained NJ phylogenetic trees of microsatellite markers. The analysis presented in this chapter is similar to the one detailed in Chapter 3, but it differs by analyzing the ties in proximity problem using a different phylogenetic tree reconstruccion algorithm.

This chapter is organised as follows. Firstly, the proposed method is explained in Section 4.2. Secondly, we show the experiments in Section 4.3 and, in the end, we present the conclusions of the chapter in Section 4.4.

4.2 The proposed method

We conducted the examination of the articles in a three-step process. In the first section, we describe the search strategy implemented to acquire the population dataset of articles utilizing the NJ algorithm for microsatellite marker classification. Afterwards, we explain the procedure for generating the sample dataset. In the last section, we expound upon the analysis of ties within the phylogenetic trees and detail the methodology employed for quantifying the occurrences of non-unique binary phylogenetic trees across articles.

4.2.1 Search strategy

We analysed the scientific literature on microsatellite markers that used the NJ algorithm. We conducted a search for articles in the Scopus database up to the year 2022. The query used was similar to the one specified in Chapter 3. The search query employed to obtain the articles was: (”microsatellite” OR ”microsatellites” OR ”simple sequence repeat” OR ”simple sequence repeats” OR ”SSR” OR ”short tandem repeat” OR ”short tandem repeats” OR ”STR”) AND (”neighbor joining” OR ”neighbour joining” OR ”neighbor-joining” OR ”neighbour-joining”), where AND and OR are the standard boolean operators. We expanded the search scope by including plural forms and multiple word variations. We limited the search to include only the terms from the query found in the title, abstract, or keywords. We gathered bibliometric data, including the document title, journal name, year of publication and subject area. In total, we retrieved 1245 articles ranging from the year 1994 to year 2022 (29 years). After eliminating three duplicated records, the dataset analyzed in this study consisted of 1242 articles.

4.2.2 Sample dataset

Figure 4.1 describes the screening process of the articles included in the analysis. We downloaded the selected articles and randomized the dataset to prevent bias towards any of the collected bibliometric data, including specific years, subject areas,

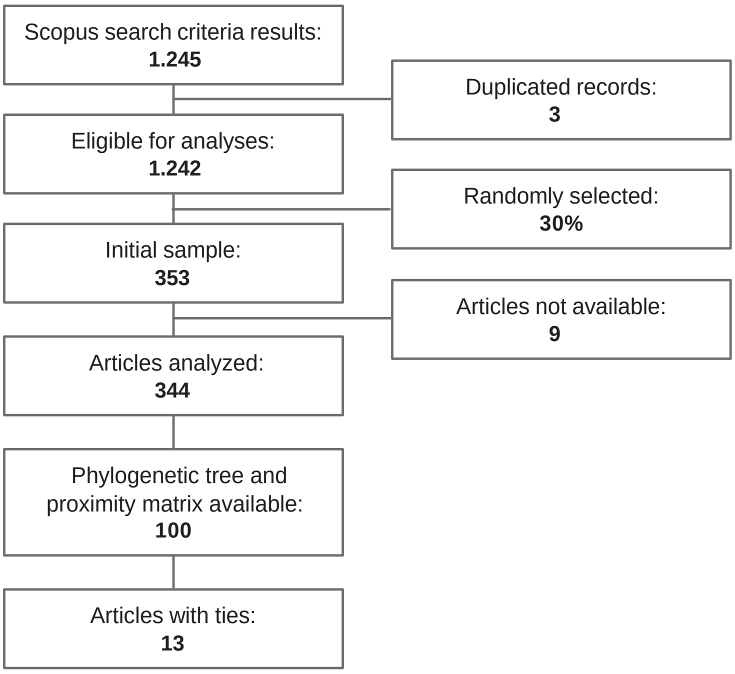


Figure 4.1. Flowchart detailing the screening process of the articles included in the dataset.

or alphabetical order. We worked with an initial subset of 353 articles, comprising roughly 30% of the dataset publications ($n = 1242$). We removed nine unavailable articles, leaving a subset of 344 publications. Within this subset, we specifically selected articles that contained either a phylogenetic tree and a matrix of proximity data (either similarities or distances), or a phylogenetic tree and a table detailing the genetic profiles of all genotypes, provided there were more than three genotypes. In those cases where only the genetic profile table was available, we calculated a matrix indicating the proportion of shared alleles using the *Adegenet* package (Jombart and Ahmed, 2011) in R version 4.1.0 (R Core Team, 2021a). We excluded articles if the proximity data matrix, whether given or calculated, did not match the genotype information in the phylogenetic tree, if the phylogenetic tree was not generated using the NJ algorithm, or if the provided information was incomplete.

Out of 344 articles analysed, only the 29% of them (100 articles) contained the necessary data to be reproducible (the complete list and additional information can be found at ASCLEPIUS-URV (2024)). Therefore, we had a final sample data set of 100 articles published in over 50 distinct journals. These journals adhere to varying data archiving guidelines, ranging from none to mandatory submission with accompanying documentation. This lack of standardised and mandatory data archiving policies across journals hinders data accessibility, reproducibility and restricts collaboration opportunities as well as scientific progress (Vines et al., 2013; Roche et al., 2014; McTavish et al., 2017). The issue of inadequate data availability is also discussed in Chapter 3, where statistical analysis reveals that approximately 26% of the analyzed articles featured correct data availability.

4.2.3 Tie analysis in phylogenetic tree reconstruction

We used the MultiFurcating Neighbour-Joining (MFNJ) algorithm to analyse and count the presence of ties. A detailed explanation of the MFNJ algorithm will be provided in Chapter 5; in essence, it efficiently groups any number of elements simultaneously, ensuring the generation of an unique phylogenetic tree.

We used the R packages *mphylo* to reconstruct the phylogenetic trees (Fernández, 2023), and *ggtree* to plot trees (Yu, 2020).

4.3 Experiments

This section is divided as follows. First, we quantify the amount of publications having at least one tie in their clustering process to reconstruct the corresponding phylogenetic tree. Then, we examine the yearly number of articles published containing a NJ tree of microsatellite markers, and we compare it with another agglomerative clustering algorithm such as UPGMA. We continue showing the topics that the articles are classified into. Finally, we show an example of a case study that yields two different binary phylogenetic trees.

4.3.1 Proportion of articles with ties

In order to count the publications with ties in their phylogenetic trees, we utilized the proximity data matrix extracted from every article in our sample dataset. Subsequently, we calculated their corresponding phylogenetic tree employing the NJ method and used the MFNJ algorithm to assess and quantify the occurrence of ties. We found that there was more than one possible binary phylogenetic tree in 13 out of the 100 articles analysed. That is the 13% of the articles, with a 95% CI ranging from 6% to 20%. Extending this percentage to the entire population, it yields an estimate of 161 articles (95% CI 74 – 248 articles) that may have alternative solutions in the form of different binary phylogenetic trees. The presence of multiple potential trees in a significant proportion of analyzed articles raises important considerations for downstream analyses. In such instances, relying on a single binary phylogenetic tree might be misleading.

4.3.2 Publications per year

Articles in our population dataset ranged from the year 1994 to the year 2022, see Figure 4.2. The search strategy utilised in this study might have unintentionally excluded some relevant articles. We acknowledge that this represents a conservative estimate of the overall count of publications featuring a NJ tree of microsatellite markers. This limitation arises from not incorporating articles published in journals beyond the Scopus Indexed Journal List. Moreover, there is a possibility that other articles within the Scopus database utilize an NJ tree with microsatellite markers, but they do not incorporate the specific terms we employed as search criteria within their titles, abstracts, or keywords.

We compare, under the same search strategy, the number of scientific publications utilizing both the NJ method and the UPGMA method, analysed in Chapter 3, because they are both agglomerative clustering methods. Although we observed a greater use of the UPGMA method over the years compared to the NJ method, both clustering methods have shown a steady increase since 2000 and reached a plateau in the number of publications in the last ten years. Approximately 70 articles have been published yearly for the NJ method and 100 articles for the UPGMA method. In the case of the NJ method, 2018 was the year with the highest number of publications ($n = 84$).

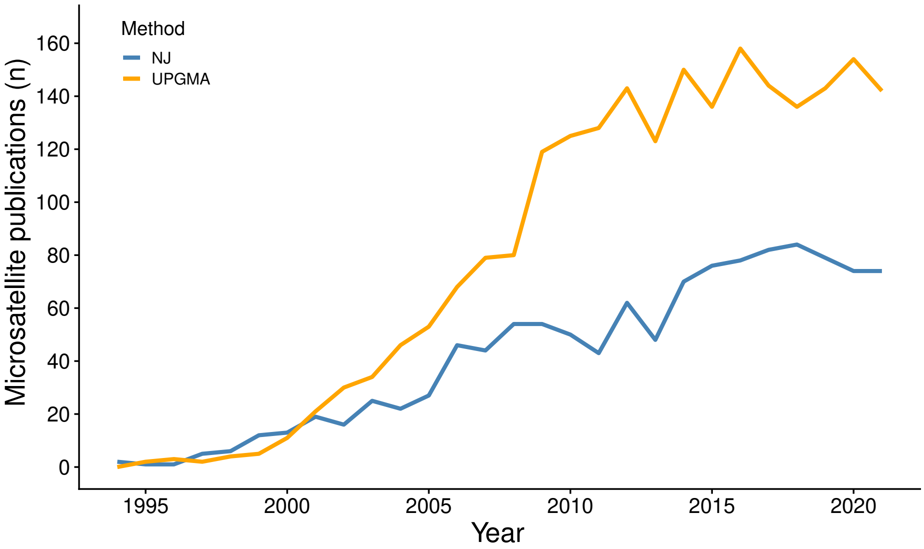


Figure 4.2. Comparison of the number of scientific publications in the Scopus database that used the NJ and the UPGMA methods on microsatellite markers from 1994 to 2022.

4.3.3 Publications per subject areas

The 1242 articles belong to 21 distinct subject areas. The predominant fields are Agricultural and Biological Sciences, comprising 42%, and Biochemistry, Genetics, and Molecular Biology, constituting 30% (Figure 4.3). Together, these two subjects account for 72% of the total article count. It was anticipated that a majority of the articles would pertain to life sciences or analogous subject areas, given the prevalent use of STR and SSR as tools within these fields. Subject areas with minimal representation, each contributing less than 1% to the total article count, have been grouped into a category labeled ”Other” in Figure 4.3. We can find examples in areas that differ significantly from the previous ones, such as Chemical Engineering (Ditta et al., 2018), Neuroscience (Li et al., 2018) and Computer Science (Chapal-Ilani et al., 2013).

4.3.4 Case study

We have selected a single case from the 100 articles analysed in our sample dataset to demonstrate how it is possible to generate distinct binary phylogenetic trees from the same data using the NJ algorithm. This example illustrates a scenario that

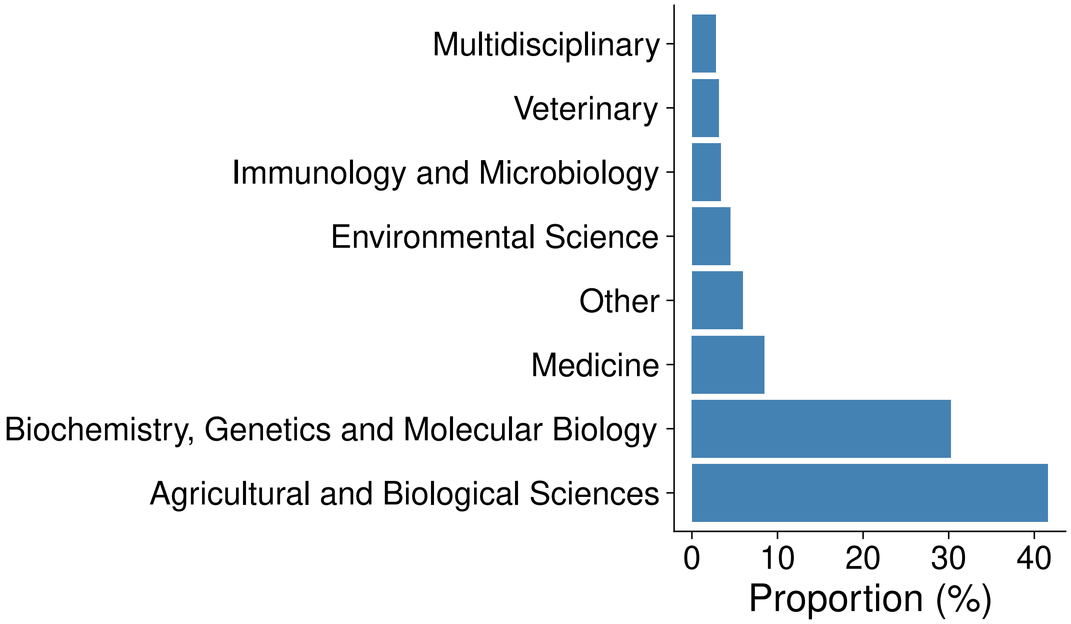
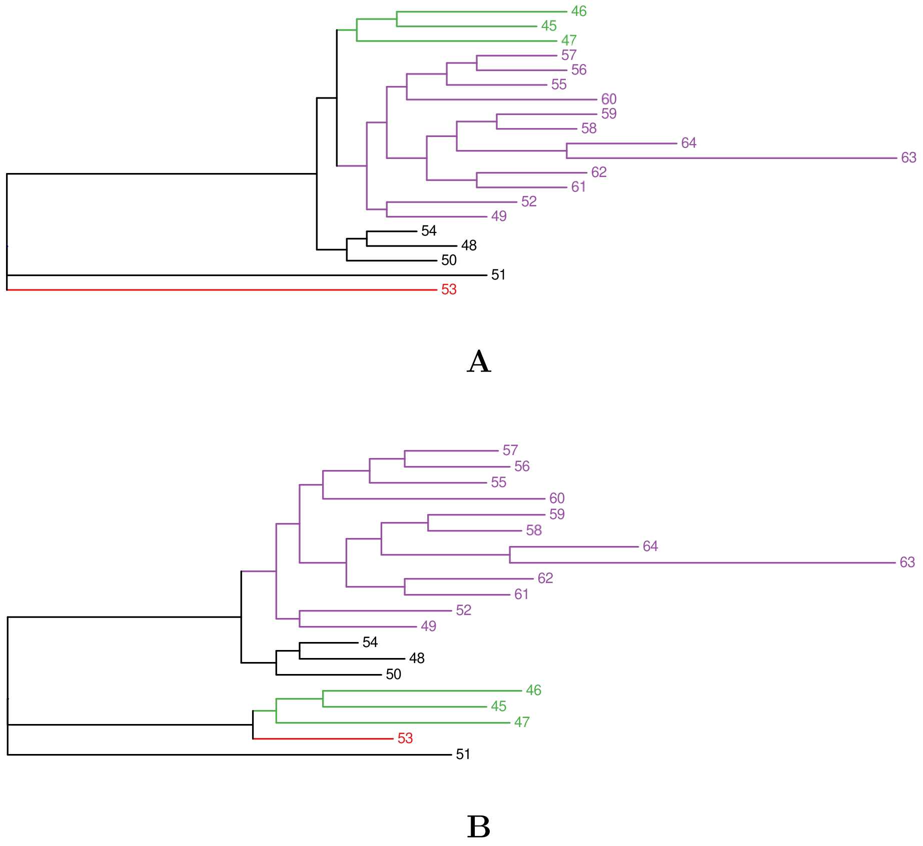


Figure 4.3. Articles classified by subject areas. The category named "Other" groups together 14 subject areas that constitute less than 1% each.

produces two distinct binary phylogenetic trees.

The authors of this study, Moiana et al. (2012), analysed the genetic diversity and population structure of the cultivated upland cotton (*Gossypium hirsutum L.*) (Figure 4.4). The study includes 20 taxa in total, assessing 27 microsatellite markers. The values in the distance matrix range from 0 to 0.73, with a precision of two decimal places. It exists a tie between cultivar 53 (depicted in red in Figure 4.4) and two clusters. The first cluster, referred to as subcluster 1, is formed by cultivars 49, 52, 55, 56, 57, 58, 59, 60, 61, 62, 63 and 64 (portrayed in purple in Figure 4.4). The second cluster, referred to as subcluster 2, is formed by cultivars 45, 46 and 47 (depicted in green in Figure 4.4). The cultivar names corresponding to the numbered identifiers are as follows: 45-BRS PEROBA, 46-BRS 7H, 47-ITA90, 48-BRS 8H, 49-BRS ARAÇÁ, 50-BRS PRECOCE, 51-BRS SUCUPIRA, 52-BRS 336, 53-BRS IPÊ, 54-BRS 286, 55-BRS CAMAÇARI, 56-ITA96, 57-BRS 335, 58-BRS ANTARES, 59-BRS 201, 60-BRS FACUAL, 61-BRS PRECOCE, 62-BRS CEDRO, 63-GIBANGA and 64-IMA CD05-8221.

Figure 4.4A represents one of the possible binary phylogenetic trees for the same data, where subclusters 1 and 2 are grouped together first. Figure 4.4B showcases



A

B

Figure 4.4. Phylogenetic trees of the genetic distances between 20 cultivars of upland cotton Moiana et al. (2012) using the NJ method. Distinct colours are employed to represent tied elements across the phylogenetic trees.

another binary phylogenetic tree for the same data, wherein subcluster 2 is first grouped with cultivar 53.

A notable difference between the two phylogenetic trees is the addition of cultivar 53. In Figure 4.4A, is clustered at the last step, while in Figure 4.4B, it is grouped with subcluster 2 in the middle of the phylogenetic tree.

4.4 Conclusions

The method presented in this chapter quantifies how frequently ties in proximity appear in published articles with microsatellite markers where the phylogenetic tree is generated using the NJ method and only show a binary phylogenetic tree.

The comparative analysis of scientific publications using the NJ and UPGMA methods reveals a notable historical preference for the UPGMA method in the context of microsatellite markers to study genetic diversity.

The majority of articles analyzed belonged to biological sciences, such as Agriculural and Biological Sciences and Biochemestry, Genetics and Molecular Biology.

The experimental validation shows that the 13% of the articles (95% CI 6 – 20%) possess at least one alternative binary phylogenetic tree to the published one. This corresponds to approximately 161 articles with at least one tie (95% CI 74 – 248 articles) in our dataset of 1242 articles. This result indicates that up to a fifth of the articles considered could be affected by the issue of ties in proximity.

5

The MultiFurcating Neighbor-Joining algorithm for reconstructing polytomic phylogenetic trees

5.1 Introduction

The motivation behind this study stems from addressing two significant challenges in distinct yet related fields: phylogenetic tree creation and the interpretability of graph-based neural networks. In phylogenetics, accurately representing evolutionary relationships is crucial, yet methods like UPGMA and Neighbour-Joining (NJ) often produce non-unique trees due to tied distances. These ambiguities can lead to misinterpretations in published research, affecting subsequent studies. This research aims to refine these algorithms, making them more robust and reliable, thus providing clearer insights into evolutionary patterns. Simultaneously, the rise of Graph Convolutional Networks (GCNs) in handling complex, graph-structured data has highlighted a critical need for understanding their decision-making processes. The inherent complexity and opacity of GCNs' mechanisms pose a barrier to their broader application and acceptance.

The aim of this thesis is to develop advanced methodologies to tackle these challenges, thereby enhancing both fields. For phylogenetics, this involves improving the accuracy and reliability of tree reconstruction algorithms to better handle tied distances and provide more consistent results. For GCNs, the focus is on employing saliency maps to demystify their decision-making processes, making these powerful tools more interpretable and trustworthy. The convergence of these aims not only advances theoretical knowledge but also delivers practical solutions that can be applied in real-world scenarios, enhancing both the precision of phylogenetic studies and the transparency of machine learning models.

In Chapter 2, we explored various alternative versions of the NJ algorithm, designed to enhance runtime efficiency, explore diverse tree spaces, and notably, address ties in proximity. Specifically, the variant addressing the ties in proximity problem only permits clusters with up to three elements, which limits their applicability in cases where clustering larger groups is necessary. This limitation and some discrepancies with the formulas used there motivated the development of a new NJ variant that generalizes the NJ algorithm.

This chapter introduces the multifurcating neighbour-joining (MFNJ) algorithm, developed during this thesis, that facilitates the simultaneous grouping of any number of elements. Additionally, the method enables the generation of a phylogenetic tree that can group multiple elements within the same cluster or across multiple clusters concurrently. This eliminates the need to run NJ multiple times with varing input orders to explore different potential phylogenetic trees. The method explained in this chapter has been presented in (Fernández et al., 2023).

This chapter is organised as follows. Firstly, Section 5.2 provides a detailed explanation of the NJ algorithm. Secondly, the proposed method is introduced and explained in Section 5.3. Thirdly, the experimental results are presented in Section 5.4. Finally, the conclusions are summarized at the end of the chapter in Section 5.5.

5.2 Background

In Chapter 2, we provided a general overview of the NJ algorithm. In this Section, we look into a more detailed explanation of the entire process, including its mathematical formulation.

5.2.1 The NJ algorithm

The NJ algorithm builds a phylogenetic tree from a matrix of evolutionary distances, D_{ij}, between each pair of taxa i, j under study.

Initially, the entire set of taxa is taken as the starting set of OTUs, arranged in a

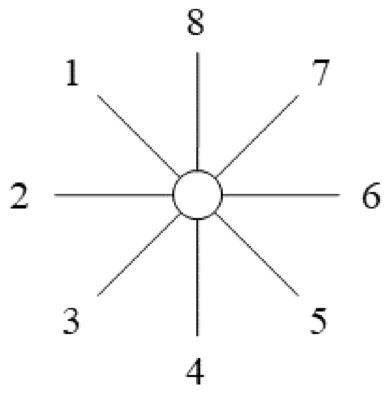


Figure 5.1. A starlike tree with no hierarchical structure.

starlike tree as in Figure 5.1, assuming that there is no clustering of OTUs. In each iteration of the algorithm, the values S_{ij} are calculated for each pair of OTUs i, j as follows:

$$S_{ij} = (N-2)D_{ij} - R_i - R_j, \tag{5.1}$$

where N is the current number of OTUs, and R_i is the sum of distances between OTU i and all the other OTUs:

$$R_i = \sum_k D_{ik}. \tag{5.2}$$

Note that Equation(5.1) is the one in Studier and Kepplter (1988), and minimizing it is equivalent to minimizing the sum of branch lengths of Saitou and Nei (1987) (Gascuel, 1994).

A pair of OTUs for which S_{ij} is the smallest is selected. Note that if more than one pair of OTUs have the smallest S_{ij}, only one pair is randomly selected. Let I ={i_1, i_2} be a pair of selected OTUs that minimize S_{ij}. Then, i_1 and i_2 are clustered together generating a new internal node u (Figure 5.2), and the distance between the new node u and any other OTU $k \neq i_1, i_2$ is calculated as follows:

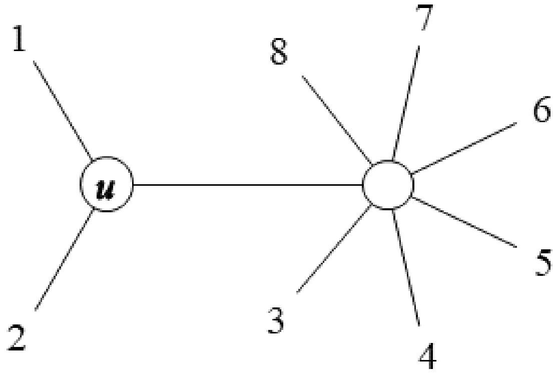


Figure 5.2. A tree with OTUs $I = \{1, 2\}$ joined to new node u.

$$D_{uk} = \frac{D_{i_1k} + D_{i_2k}}{2} - \frac{D_{i_1i_2}}{2}. \tag{5.3}$$

Equation (5.3), cited from Studier and Kepplter (1988), is equivalent to the one in Saitou and Nei (1987), both of which reconstruct the same tree (Gascuel, 1994). This equivalence holds despite their differing sources.

Finally, the length of the new branch linking i_1 and u is calculated as follows:

$$L_{i_1u} = \frac{D_{i_1i_2}}{2} + \frac{R_{i_1I^C}}{N-2} - \frac{R_{II^C}}{2(N-2)}, \tag{5.4}$$

where I^C is the complement of I, R_{iI^C} is the sum of distances between an OTU $i \in I$ and all the other OTUs $k \notin I$:

$$R_{iI^C} = \sum_{k \notin I} D_{ik}, \tag{5.5}$$

and R_{II^C} is the sum of distances between all the OTUs $i \in I$ and all the other OTUs $k \notin I$:

$$R_{II^C} = \sum_{i \in I} \sum_{k \notin I} D_{ik}, \tag{5.6}$$

L_{i_2u} can be obtained in the same way or simply subtracting L_{i_1u} from $D_{i_1i_2}$.

In each iteration, the two selected OTUs, i_1 and i_2, are removed from the distance matrix, D, and a new internal node u is added. The procedure ends when the current number of OTUs is equal to three, and there is only one possible unrooted tree. The branch length for each one of the last three OTUs, i_1, i_2 and i_3, is calculated as follows:

$$L_{i_1u} = \frac{D_{i_1i_2} + D_{i_1i_3} - D_{i_2i_3}}{2}. \tag{5.7}$$

5.3 The proposed method

5.3.1 MultiFurcating Neighbour-Joining

The method we propose, the MFNJ algorithm, generalizes the NJ algorithm. Both algorithms use Equation (5.1) to compute S_{ij} in the same way, where the two algorithms diverge is in the procedure for joining OTUs.

Suppose that, in a specific iteration, two pairs of OTUs, i_1, i_2 and i_2, i_3, have the smallest S_{ij}; that is, $S_{i_1i_2} = S_{i_2i_3} = S_{min}$. In this case, the NJ algorithm can only join one of these pairs of OTUs, i_1, i_2 or i_2, i_3, to generate a new internal node u, which pair is selected has consequences for the next steps of the NJ algorithm. In the MFNJ algorithm, given that both pairs of OTUs, i_1, i_2 and i_2, i_3, have i_2 in common, we propose to generate a new internal node u joining the set of three OTUs $I = \{i_1, i_2, i_3\}$.

Distance between an internal node and an OTU

More generally, let $I = \{i_1, i_2, \ldots, i_P\}$ be a set of OTUs to be clustered together generating a new internal node u. The distance between any OTU $i \in I$ and any other OTU $k \notin I$ can be separated in two parts (Figure 5.2):

$$D_{ik} = L_{iu} + D_{uk}. \tag{5.8}$$

Taking this equality for all the OTUs $i \in I$, the distance between the new node u and any OTU $k \notin I$ can be averaged as follows:

$$D_{uk} = \frac{1}{|I|} \sum_{i \in I} (D_{ik} - L_{iu}), \tag{5.9}$$

where $|I|$ is the number of OTUs to be joined to the internal node u. Now, using the equality given by Saitou and Nei (1987) for the sum of branch lengths of a star-shaped tree with central node u:

$$\sum_{i \in I} L_{iu} = \frac{R_{II}}{|I| - 1}, \tag{5.10}$$

where R_{II} is the sum of distances between all the OTUs in I:

$$R_{II} = \sum_{i \in I} \sum_{\substack{i' \in I \\ i' > i}} D_{ii'}, \tag{5.11}$$

we finally propose to generalize Equation (5.3) for the calculation of the distance D_{uk} between the new node u and any other OTU $k \notin I$ as follows:

$$D_{uk} = \frac{R_{Ik}}{|I|} - \frac{R_{II}}{|I|\,(|I| - 1)}, \tag{5.12}$$

where R_{Ik} is the sum of distances between all the OTUs in I and OTU $k \notin I$:

$$R_{Ik} = \sum_{i \in I} D_{ik}. \tag{5.13}$$

Distance between two internal nodes

As a matter of fact, there may be cases where more than one set of OTUs can be clustered during the same iteration of the algorithm, being these sets of OTUs disjoint sets. In these cases, when there are two new internal nodes u and v joining two disjoint sets of OTUs $I = \{i_1, i_2, \ldots, i_P\}$ and $J = \{j_1, j_2, \ldots, j_Q\}$, respectively, the distance between any OTU $i \in I$ and any other OTU $j \in J$ can be separated in three parts (Figure 5.3):

$$D_{ij} = L_{iu} + D_{uv} + L_{jv}. \tag{5.14}$$

Taking this equality for all the OTUs $i \in I$ and $j \in J$, the distance between the new nodes u and v can be averaged as follows:

$$D_{uv} = \frac{1}{|I|\,|J|} \sum_{i \in I} \sum_{j \in J} (D_{ij} - L_{iu} - L_{jv}), \tag{5.15}$$

which, using Equation (5.10), can be expressed as follows:

$$D_{uv} = \frac{R_{IJ}}{|I|\,|J|} - \frac{R_{II}}{|I|\,(|I|-1)} - \frac{R_{JJ}}{|J|\,(|J|-1)}, \tag{5.16}$$

where R_{IJ} is the sum of distances between pairs of OTUs in I and J:

$$R_{IJ} = \sum_{i \in I} \sum_{j \in J} D_{ij}, \tag{5.17}$$

and R_{II} and R_{JJ} are calculated using Equation (5.11).

Branch length when the complement of I is not empty

To generalize Equation (5.4), let u be a new internal node joining all the OTUs in $I = \{i_1, i_2, \ldots, i_P\}$. Given any OTU $i \in I$, when I^C is not empty we can sum the equality in Equation (5.8) for all the OTUs $k \notin I$:

$$\sum_{k \notin I} D_{ik} = (N - |I|) L_{iu} + \sum_{k \notin I} D_{uk}. \tag{5.18}$$

Using the definition given in Equation (5.5) and substituting D_{uk} with the expression in Equation (5.12), we obtain:

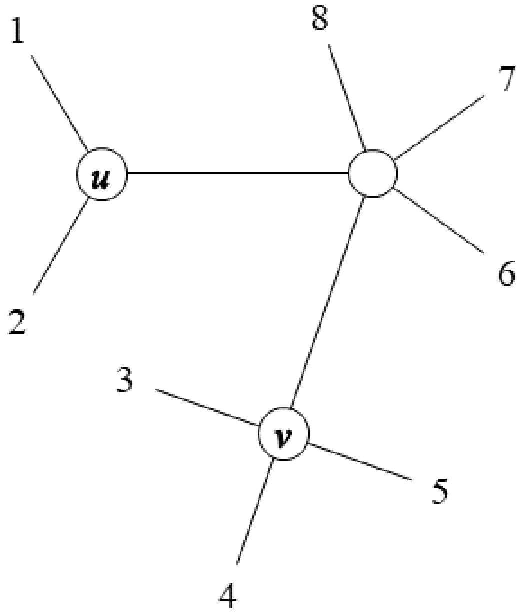


Figure 5.3. A tree with OTUs $I = \{1, 2\}$ joined to a new node u, and OTUs $J = \{3, 4, 5\}$ joined to another new node v, during the same iteration of the algorithm.

$$R_{iI^C} = (N - |I|)L_{iu} + \sum_{k \notin I} (\frac{R_{Ik}}{|I|} - \frac{R_{II}}{|I|\,(|I| - 1)}). \tag{5.19}$$

Now, we can use the definition given in Equation (5.6) and divide everything by $N - |I|$, obtaining:

$$\frac{R_{iI^C}}{N - |I|} = L_{iu} + \frac{R_{II^C}}{|I|\,(N - |I|)} - \frac{R_{II}}{|I|\,(|I| - 1)}, \tag{5.20}$$

which, rearranging terms, finally yields:

$$L_{iu} = \frac{R_{II}}{|I|\,(|I| - 1)} + \frac{R_{iI^C}}{N - |I|} - \frac{R_{II^C}}{|I|\,(N - |I|)}, \tag{5.21}$$

where R_{II}, R_{iI^C}, and R_{II^C} are defined in Equations (5.11), (5.5), and (5.6), respectively.

Branch length when the complement of I is empty

In case that all the remaining OTUs are clustered together in the same set I and, therefore, the set I^C is empty, then the new internal node u joins all the remaining OTUs, and the distance between any OTU $i \in I$ and any other OTU $i' \in I$, $i' \neq i$, can be separated as follows:

$$D_{ii'} = L_{iu} + L_{i'u}. \tag{5.22}$$

Summing this equality for all the OTUs $i' \in I$, $i' \neq i$, we obtain:

$$R_{iI} = (|I| - 1)L_{iu} + \sum_{i' \in I} L_{i'u} - L_{iu}, \tag{5.23}$$

where R_{iI} is the sum of distances between OTU $i \in I$ and all the other OTUs $i' \in I$, $i' \neq i$:

$$R_{iI} = \sum_{\substack{i' \in I \\ i' \neq i}} D_{ii'}. \tag{5.24}$$

Now, if we use Equation (5.10) for the sum of branch lengths of a starlike tree, we see that Equation (5.23) is equivalent to:

$$R_{iI} = (|I| - 2)L_{iu} + \frac{R_{II}}{|I| - 1}, \tag{5.25}$$

which, rearranging terms and dividing everything by $|I| - 2$,

$$L_{iu} = \frac{R_{iI}}{|I| - 2} - \frac{R_{II}}{(|I| - 1)(|I| - 2)}. \tag{5.26}$$

It is important to note here that both Equations (5.21) and (5.26) satisfy Equation (5.10) for the sum of branch lengths of a starlike tree.

In each iteration, all the OTUs in I are removed from the distance matrix, and the new node u is added. The procedure ends when all the remaining OTUs are clustered in the same set I and the set I^C is empty. If there are no polytomies, this will happen for sure when the number of remaining OTUs is equal to three. In this case, Equation (5.26) reduces exactly to Equation (5.7). As a matter of fact, when there are no polytomies, the MFNJ algorithm reconstructs the same phylogenetic trees as the NJ algorithm.

To the best of our knowledge, there are only two methods that addresss the ties in proximity problem: the live neighbour-joining and the extended neighbor-joining algorithms. Nevertheless, the formulas proposed in the extended neighbor-joining algorithm do not satisfy Equation (5.10) for the sum of branch lengths of a starlike tree. Both the live neighbor-joining and the extended neighbor-joining methods are limited in their ability to join only up to three OTUs to a new internal node. In contrast, the MFNJ algorithm is more versitale because Equations (5.12), (5.16), (5.21), and (5.26) can accommodate any number of OTUs.

5.4 Experiments

This section shows an example of the differences between the phylogenetic trees reconstructed by the NJ and the MFNJ algorithms using a specific distance matrix. In the case of the NJ algorithm, two possible phylogenetic trees are reconstructed. In the case of the MFNJ algorithm, only one phylogenetic tree is possible.

To do so, we used as input for both algorithms the matrix of distances given in Table 5.1. It is composed of the pairwise differences among mitochondrial DNA sequences of nine brown bears (*Ursus arctos* L.). We selected this case study because it had been previously used in one of the first articles that described the ties in proximity problem (Backeljau et al., 1996).

After four iterations of the NJ algorithm, *Kodiak*, *Captive*-3, *Captive*-5, *Grizzly*, and *Polar*-2 are clustered together in a subtree that we call *Subtree*-4 (colored in blue in Figure 5.4), and the other four bears remain nonclustered. At the fifth iteration of the algorithm, there is a tie between the pairs *Captive*-4 and *Subtree*-4, and *Subtree*-4 and Black, because their S_{ij} values are equal and the smallest. Since the

NJ algorithm cannot cluster three OTUs in a single step, two distinct phylogenetic trees are possible depending on the criterion used to break the tie. If *Captive*-4 and *Subtree*-4 are clustered first, then the phylogenetic tree in Figure 5.4A is obtained. However, if *Subtree*-4 and *Black* are clustered first, then the phylogenetic tree in Figure 5.4B is obtained.

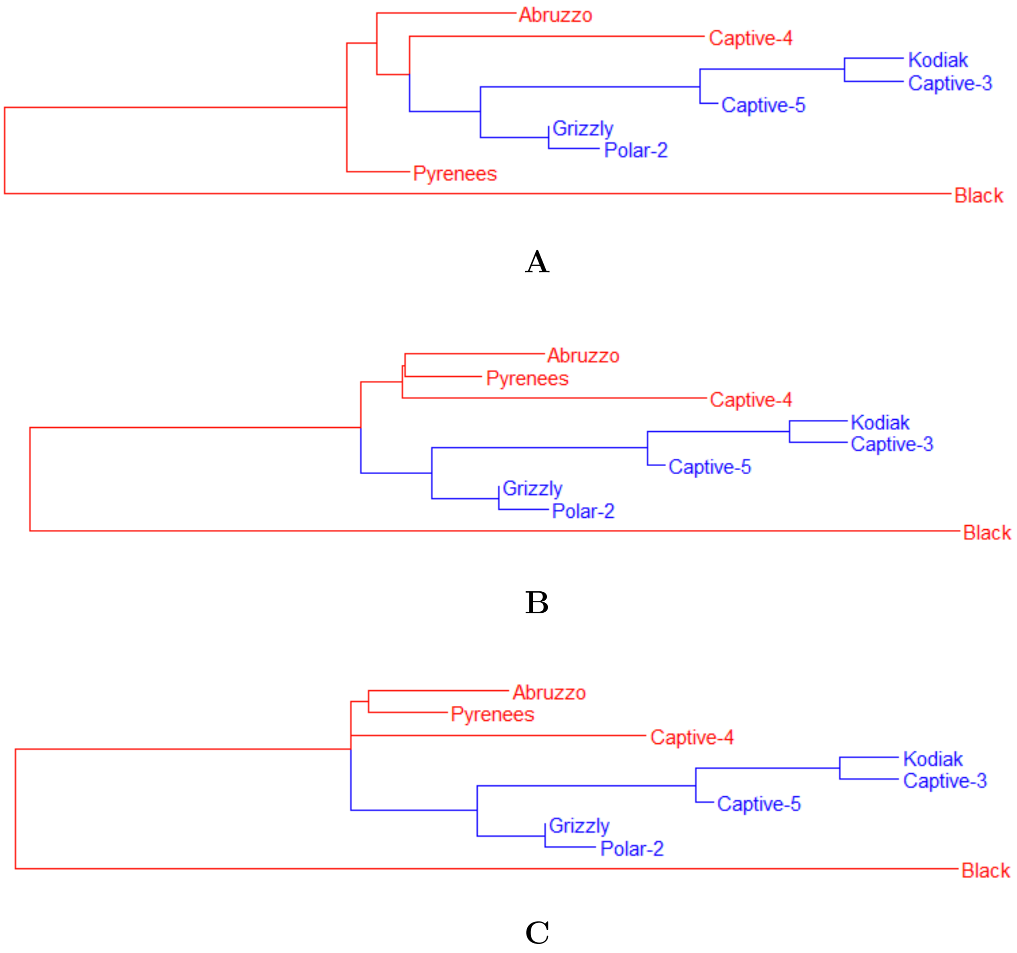


Figure 5.4. Phylogenetic trees obtained for the matrix of distances among bears given in Table 5.1. The trees have been plotted as rooted trees for convenience of comparison, where the longest branch has been placed at the root of each tree. At the fifth iteration of the algorithm, there is a tie between *Black*, *Captive*-4, and the subtree in blue. The bears in red are clustered during the last iterations of both the NJ and the MFNJ algorithms. **A**, **B** Two different dichotomic phylogenetic trees are possible when using the NJ algorithm. **C** A unique phylogenetic tree is possible when using the MFNJ algorithm, where a polytomy joining more than two subtrees can be observed.

When the MFNJ algorithm is used with the same dataset, the first iterations are identical to the NJ algorithm, until the tie is found at the fifth iteration. Then, the MFNJ algorithm clusters *Captive*-4, *Subtree*-4, and *Black* at the same time forming a polytomy. Figure 5.4C shows the complete phylogenetic tree reconstructed by the MFNJ algorithm. This multifurcating tree is uniquely determined, which guarantees the reproducibility of any study on it.

Table 5.1. Pairwise percentage differences among mitochondrial DNA sequences of nine brown bears (Randi et al., 1994).

	Abruzzo	Pyrenees	Kodiak	Captive-3	Captive-4	Captive-5	Grizzly	Polar-2
Pyrenees	1.3							
Kodiak	4.3	4.3						
Captive-3	4.3	4.3	0.7					
Captive-4	2.7	2.3	5.0	5.0				
Captive-5	3.0	3.0	1.3	1.3	3.7			
Grizzly	1.7	1.7	2.7	2.7	2.3	2.0		
Polar-2	2.0	2.0	3.0	3.0	2.7	2.3	0.3	
Black	8.7	8.0	10.0	10.0	10.0	8.7	9.0	9.4

5.5 Conclusions

We proposed the multifurcating neighbor-joining (MFNJ) algorithm for agglomerative hierachical clustering, which addresses ties in proximity problem. This algorithm is a generalization of the standard neighbor-joining (NJ) method.

We have generalized the definitions of distance between a cluster and any new OTU, as well as the distance between two clusters. Additionally, we have generalized the calculation of branch lengths linking new cluster elements.

Ties in the agglomerative process in phylogenetic trees can be visualized as lines connecting all clustered elements.

When there are no ties distances, MFNJ produces the same results as the NJ algorithm. However, when ties are present, MFNJ consistently yields a unique phylogenetic tree, regardless of the order of input data.

6

Methods in saliency maps

6.1 Introduction

Understanding the decision-making processes of Graph Convolutional Networks (GCNs), particularly in graph-structured data, remains a challenging yet crucial pursuit. While GCNs excel in learning meaningful representations from graphs, their decision-making processes are complex and not easaly interpretable. Saliency map generators (SMGs) have emerged to address these challenges. SMG are post-hoc explanation methods to understand the decision-making process of GCNs. Originally developed for image classification, SMGs like Grad-CAM and Grad-CAM++ have been adapted to uncover the importance of nodes and edges in GCNs. This chapter presents an overview of the basic concepts related to graphs, GCNs and the methodologies and metrics involved in using SMGs to enhance the interpretability of GCNs, shedding light on their decision-making processes and improving trust in their predictions.

6.2 Graphs

A graph is a mathematical data structure used to represent pairwise relationships between objects. These graphs are made up of nodes (also known as vertices) connected by edges (also known as links) (Figure 6.1). Graphs are used to model a wide array of structures such as computer networks, social networks, characters and letters, pixels in images, road maps, and chemical structures. In this manuscript, we will focus on and study graphs in the context of representing chemical structures. The chemical structure of a molecule is its spatial arrangement of its atoms and their chemical bonds. Molecules can be represented as graphs, where nodes represent the chemical atoms and edges represent the chemical bonds (Figure 6.2).

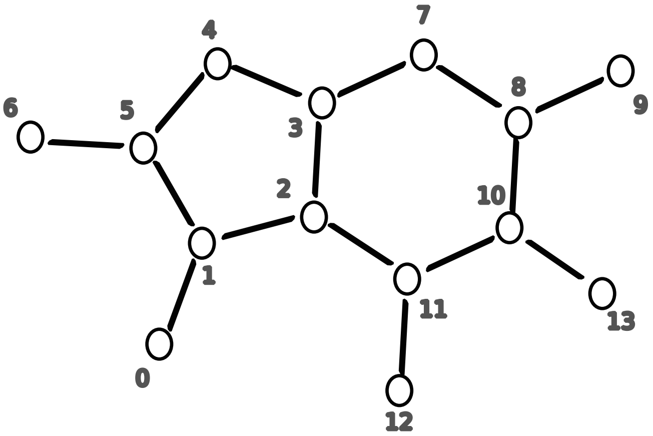


Figure 6.1. Example of a graph with 14 nodes and 15 edges.

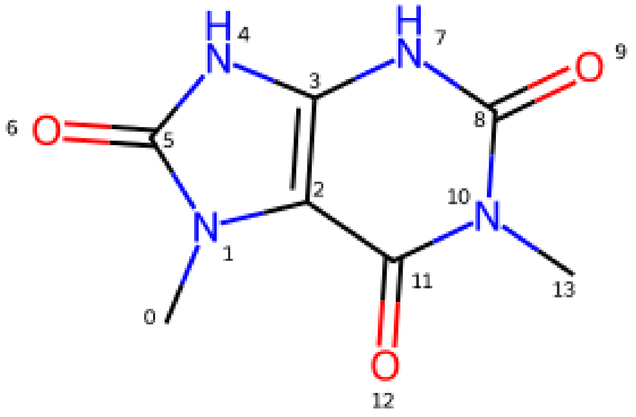


Figure 6.2. A graph representation of a chemical molecule, where nodes denote atoms and edges represent chemical bonds. Different colors indicate various elements: black for Carbon, red for Oxygen, and blue for Nitrogen. The graph is the same as the one in Figure 6.1.

6.3 Graph Convolutional Networks

Graph Convolutional Networks (GCNs) (Kipf and Welling, 2017) are deep neural models renowned for their ability to extract meaningful features in prediction tasks where data is represented by graphs.

We define a graph with attributes as a combination of a node feature matrix X , and an adjacency matrix A of dimensions R^{NxN}, where N is the number of nodes in the graph.

The graph is represented as $G = (X, E)$, where $|X| = N$ is the number of nodes and $|E| \leq N^E$ is the number of edges in the graph. The adjacency matrix $A \in R^{NxN}$ captures the relationships between nodes in the graph. Each entry A_{ij} indicates the presence or absence of an edge between nodes i and j. The propagation in a GCN is defined by the equation:

$$X^{l+1} = f(\hat{A}X^l W^l) \tag{6.1}$$

Where X^l represents the node features at layer l, W^l is the weight matrix of the hidden layer, $f(\cdot)$ represents an activation function, typically a non-linear function such as the rectified lienar unit (ReLU), and $\hat{A}$ is the normalized adjacency matrix, defined as:

$$\hat{A} = \tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}} \tag{6.2}$$

Where $\tilde{A}$ is the identity matrix of the adjacency matrix A and $\tilde{D}^{-\frac{1}{2}}$ is the inverse of the diagonal degree matrix of $\tilde{A}$.

6.4 Saliency Map Generators: Post-hoc Explanation Methods

Saliency map generators (SMG) are techniques initially applied to image classification, which indicate the areas of the input data that played an important role in the model decision (Gomez and Mouchère, 2023; Zhang et al., 2020). These areas are called saliency maps. Thus, they have become indispensable tools for interpreting the predictions of deep neural models (Bhambra et al., 2022). SMG, when adapted to GCNs, can be used for the interpretation of nodes and edges in regression or classification applications where the data is characterised by graphs (Pope et al., 2019; Kensert et al., 2021).

SMG are methods encompassed in post-hoc explanation methods, which are general approaches to generate explanations of the decisions made by any prediction model without requiring retraining. Specifically, for SMG, Class Activation Mapping (CAM) (Zhou et al., 2016) and its successors, such as Grad-CAM (Selvaraju et al., 2017) and Grad-CAM++ (Chattopadhay et al., 2018) are some of the most popular methods. Interestingly, Grad-CAM was adapted to generate the saliency maps for prediction models based on GCN in Pope et al. (2019). In the next subsections, we describe Grad-CAM and Grad-CAM++ in detail.

6.4.1 Grad-CAM

Grad-CAM also known as Gradient-weighted Class Activation Mapping is a SMG based on deep convolutional neural networks. Nevertheless, this method deduces the saliency map taking into consideration only the last layer of the classification model, in a similar way as CAM.

Formally, for any class c within a dataset, the gradient of the score for class c, y^c (prior to a softmax or any activation function), is calculated with respect to the activation of feature maps in a convolutional layer *k*, A^k. Subsequently, these resulting gradients are global average pooled across the width and height dimensions of the data (denoted as i and *j*, respectively). The weigths w_k^c for a feature map A^k and class c are:

$$w_k^c = \frac{1}{Z}\sum_i\sum_j \frac{\partial y^c}{\partial A_{ij}^k} \tag{6.3}$$

where Z is the number of elements in the feature map A^k. Finally, a weighted combination of w_k^c with A^k is performed to obtain the class-specific saliency map, L_{ij}^c:

$$L_{ij}^c = ReLU(\sum_k w_k^c A^k) \tag{6.4}$$

6.4.2 Grad-CAM++

Grad-CAM++ is a generalization of the Grad-CAM algorithm to improve the handling of multiple occurrences of a localized object in an image and improve the localization accuracy. The main difference is how the weights w_k^c for a feature map A^k and class c are computed. First, the gradient weights α_{ij}^{kc} are computed as:

$$\alpha_{ij}^{kc} = \frac{\frac{\partial^2 Y^c}{(\partial A_{ij}^k)^2}}{2\frac{\partial^2 Y^c}{(\partial A_{ij}^k)^2} + \sum_a \sum_b A_{ab}^k \frac{\partial^3 Y^c}{(\partial A_{ij}^k)^3}} \tag{6.5}$$

where *(i,j)* and *(a,b)* are iterators over the same A^k to avoid confusion. Then, the weights w_k^c are reformulated from Equation (6.3) as:

$$w_k^c = \sum_i \sum_j \alpha_{ij}^{kc} ReLU(\frac{\partial Y^c}{\partial A_{ij}^k}) \tag{6.6}$$

Finally, the weighted combination of w_k^c with A^k is used to obtain the class-specific saliency map L_{ij}^c, as defined by Equation (6.4).

6.5 Single Step Metrics for classification

Understanding the decision-making processes of deep learning models is a difficult task. This is because they are typically characterised by multiple layers of non-linear transformations, wherein the mapping between input and output is complex and not readily interpretable.

Various metrics exist to measure the correlation between the saliency map explanation from SMG and the model's prediction abilities given an image (Gomez and Mouchère, 2023; Chattopadhay et al., 2018; Jung and Oh, 2021). These metrics compare the model's predictions using the original input data to those using modified input data. The modifications are based on the saliency map generated from the original input data through a process known as masking.

The quality of the output depends on which part of the input data has been modified. If the modified data corresponds to the areas highlighted by the saliency map as important, the decrease in output quality should be more significant compared to modifications in areas deemed unimportant by the saliency map. The input data masking can be executed either in a single step or iteratively. This chapter focuses on the former case, known as Single Step Metrics (STM) (Gomez and Mouchère, 2023).

STM are evaluation metrics designed to assess the correctness of saliency maps generated by a SMG, such as Grad-CAM. These metrics are based on applying a usually slight modification to the input data and later measuring the consequential impact on the final prediction.

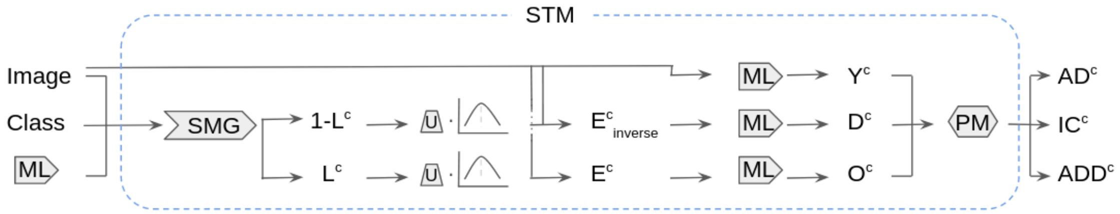


Figure 6.3. STM in image classification. ML stands for Machine Learning, SMG refers to the Saliency Map Generator, U denotes Upsampling and Normalization, and PM represents Performance Metrics.

Figure 6.3 provides an overview of the STM process for an image classification model. The process begins with creating the saliency map L^c using the SMG based on the input data or image i, a class c, and the model's trained weights. Next, with the input data or image and L^c, the explanation map E^c is computed. E^c involves a masking operation, where the saliency maps are point-wise multiplied with the original image I. This operation is defined as:

$$E^c = s(u(L^c)) \circ I \tag{6.7}$$

Where $u(\cdot)$ indicates the upsampling into the original data dimensions, $s(\cdot)$ is the min-max normalization function and $\circ$ is the Hadamard product. E^c only preserves the information of I in the pixels that are considered important and reduces the ones that are not. Then, using this modified image a new classification score O^c for a class c is predicted using the trained ML model. A higher value of O^c is expected to correspond to an increased confidence in the model's prediction of the image belonging to a specific class.

An inverse masking operation is also performed on I. This involves creating an inverse explanation map $E^c_{inverse}$ for a given class c, calculated by point-wise multiplication of the inverse of the saliency maps with the original image I, as follows:

$$E^c_{inverse} = [1 - s(u(L^c))] \circ I \tag{6.8}$$

In this case, $E^c_{inverse}$ preserves the information in the pixels of I that are not considered important for the final classification and reduces the information in the pixels that are important (i.e. *deletion*). Using this modified image, a new classification score, D^c, is predicted using the trained ML model.

Consequently, there are three different classification scores for the same image, depending on the masking applied: Y^c for no masking, O^c for direct masking, and D^c for inverse masking. Finally, these scores are used to evaluate the STM using specific performance metrics. The three performance metrics are Average drop percentage (AD), Increase in Confidence (IC), and Average Drop in Deletion percentage (ADD) (Chattopadhay et al., 2018; Jung and Oh, 2021), which are explained in section 6.6.

6.6 Performance metrics

6.6.1 Average drop (AD)

AD is the average percentage drop in the model's confidence for a given class c in an image classification value when comparing it to O^c, defined as:

$$AD = \frac{1}{N} \sum_{n=1}^{N} \frac{max(0, Y_i^c - O_i^c)}{Y_i^c} \times 100 \tag{6.9}$$

We use max to handle cases where Y_i^c is higher than O_i^c and N denotes the number of elements in the dataset. In case of equal output values, it can be deduced that the saliency map is insensitive to the model-learned parameters. The AD value is computed per image and then averaged over the entire dataset. A lower AD value indicates better performance.

6.6.2 Increase in Confidence (IC)

IC is complementary to the previous metric, AD. IC measures the number of times in the entire dataset that O_i^c is higher than Y_i^c. In other words, IC indicates how often the predicted classification value O_i^c is higher than the predicted value Y_i^c. This suggests that the model's learned parameters for classification using SMG are effective and relevant to the actual classification task. We define IC as follows:

$$IC = \frac{1}{N}\sum_{n=1}^{N} 1_{[Y_i^c < O_i^c]} \times 100, \tag{6.10}$$

where 1_x is an indicator function that returns 1 when the argument is true, $Y_i^c < O_i^c$. The IC value is computed per image and then averaged over the entire dataset. A higher IC value indicates better performance.

6.6.3 Average Drop in Deletion (ADD)

ADD modifies AD within a specific context by an inverse operation, $E^c_{inverse}$, to compute the classification score, D_i^c. It evaluates the average percentage drop in the model's confidence caused by the inverse masking on the final prediction. ADD can be defined as:

$$ADD = \frac{1}{N}\sum_{n=1}^{N} \frac{Y^c - D^c}{Y_c} \times 100 \tag{6.11}$$

The ADD value is computed per image and then averaged over the entire dataset. A higher ADD value indicates better performance.

7

Saliency maps in graph regression models

7.1 Introduction

Deep learning models are capable of making highly accurate predictions, but understanding how they arrive at these predictions and which parts of the input data are key to that prediction is challenging due to the intricate and layered nature of their structure. This lack of transparency makes it difficult to trust and validate the model's decisions, especially in critical applications such as healthcare, finance, and autonomous systems.

To address this issue, Saliency Map Generators (SMG) have been developed as a means to enhance the interpretability. SMG produces saliency maps that highlight the most important features in the input data that contribute to the model's predictions. By identifying these key features, saliency maps provide insights into the decision-making process of the model, making it easier to understand why certain predictions are made.

However, understanding saliency maps often requires additional analysis and domain-specific knowledge. Therefore, while saliency maps offer a valuable tool for improving model interpretability, their effective use depends on the user's expertise and the context in which they are applied.

This chapter introduces Single Step Metrics (STM) designed for graph inputs, with a focus on graph regression, to evaluate the performance of SMG. By comparing performance metrics with the insights provided by the saliency maps, we assess the efficacy of two SMG, Grad-CAM and Grad-CAM++, across various datasets. These metrics measure the effectiveness of the SMG in identifying the importance of each node in the regression results.

This chapter is organised as follows. Firstly, the proposed method is explained in Section 7.2. Secondly, we show the experiments in Section 7.3 and, in the end, we present the conclusions of the chapter in Section 7.4.

7.2 The proposed method

7.2.1 Model architecture

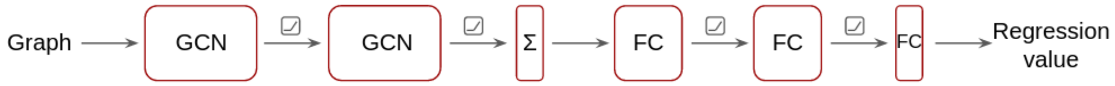


Figure 7.1. Model architecture.

The GCN model's architecture begins with an input layer that accepts graph structures. This is followed by two Graph Convolutional layers, which capture local neighbourhood information. A readout layer aggregates the graph-level information, which is then passed through three Fully Connected layers (Figure 7.1). The final output is a single value representing the regression value. Weights in the model are randomly initialized. The training process involves minimizing the mean squared error (MSE) between the predicted and actual global property using the Adam optimizer.

7.2.2 Single Step Metrics for graph regression

STM were initially designed for assessing image classification models, but they can be adapted to work with graph regression models.

Figure 7.2 provides an outline of the STM process for a graph regression model. This process is very similar to the one explained in Chapter 6 but two modifications have been incorporated. The first one involves the use of graphs as input data instead of images, while the second one implies the use of STM in regression models. Accordingly, we define the explanation map, E_i, for a given graph as the point-wise

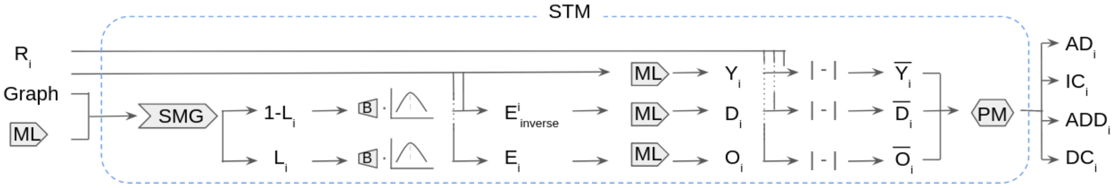


Figure 7.2. STM in graph regression. ML stands for Machine Learning, SMG refers to the Saliency Map Generator, B denotes Broadcasting and Normalization, R_i refers to the real value, and PM represents Performance Metrics.

multiplication of the saliency maps L_i with the node features of the original graph X_i:

$$E_i = b(|s(L_i)|) \circ X_i \tag{7.1}$$

where $|s(\cdot)|$ represents the normalization function applied to the saliency map values, transforming them to a range between 0 and 1 before being transformed into their absolute values, *b(·)* represents the broadcasting function into the original dimensions of X_i. In this way, each element of E_i represents the importance of a node, instead of a pixel. And the inverse explanation map $E^i_{inverse}$ as the point-wise multiplication of the inverse of the saliency maps with X_i as:

$$E^i_{inverse} = [1 - b(|s(L_i)|)] \circ X_i \tag{7.2}$$

Note that if we change the *b(·)* function into a *u(·)* function we can work with image data as E^c and $E^c_{inverse}$ of STM for image classification in Chapter 6 (Equations (6.7) and (6.8)).

Another change is the use of the absolute difference between the real value R_i and the predicted values for Y_i, O_i, and D_i ($\bar{Y}_i$, $\bar{O}_i$,$\bar{D}_i$, respectively), defined as follows:

$$\bar{W}_i = |R_i - W_i| \tag{7.3}$$

where W_i represents the predicted score. The reason is that in the context of classification models, a higher value of W_i is expected to correspond to an increased confidence in the model's prediction of the data belonging to a specific class. Contrarily, in regression models, the objective is to achieve a prediction the closest to the actual value. Therefore, using the actual predicted value to compare its performance is not meaningful.

7.2.3 Performance metrics

The performance metrics have been adapted to accommodate regression, and we have introduced a fourth metric, Drop in Confidence (DC). In each of these metrics, the goal is to attain higher values, indicating enhanced performance. Additionally, each metric is computed individually for every graph and subsequently averaged across the entire dataset.

Average drop (AD)

AD is the average increase in the model's confidence for the prediction value, $\bar{Y}_i$ when comparing it to $\bar{O}_i$, defined as:

$$AD = \frac{1}{N}\sum_{n=1}^{N}\frac{|\bar{Y}_i - \bar{O}_i|}{max(\bar{Y}_i, \bar{O}_i)} \tag{7.4}$$

We use *max* to handle cases where $\bar{Y}_i$ is less than $\bar{O}_i$, as well as when $\bar{Y}_i$ equals $\bar{O}_i$.

Increase in Confidence (IC)

IC quantifies the frequency with which $\bar{O}_i$ is lower than $\bar{Y}_i$ across the entirety of the dataset, defined as:

$$IC = \frac{1}{N}\sum_{n=1}^{N} 1_{[\bar{Y}_i > \bar{O}_i]} \tag{7.5}$$

where 1_x is an indicator function that returns 1 when the argument is true, $\bar{Y}_i > \bar{O}_i$.

Average Drop in Deletion (ADD)

ADD is the comparison of the prediction value, $\bar{Y}_i$, to $\bar{D}_i$, defined as:

$$ADD = \frac{1}{N}\sum_{n=1}^{N}\frac{max(0, \bar{D}_i - \bar{Y}_i)}{\bar{D}_i} \tag{7.6}$$

We use max to handle cases where $\bar{Y}_i$ is higher than $\bar{D}_i$, as well as when $\bar{Y}_i$ equals $\bar{O}_i$.

Drop in Confidence (DC)

DC is a new STM evaluation metric that complements ADD. This metric is designed to provide additional information similar to how IC complements AD. DC quantifies the frequency in the entire dataset where $\bar{Y}_i$ is lower than $\bar{D}_i$. We define DC as:

$$DC = \frac{1}{N}\sum_{n=1}^{N} 1_{[\bar{Y}_i < \bar{D}_i]} \tag{7.7}$$

where 1_x is an indicator function that returns 1 when the argument is true, $\bar{Y}_i < \bar{D}_i$.

7.3 Experiments

7.3.1 Datasets

To validate our approach, we applied it to three distinct chemical datasets, which were defined to test models that predict the retention time of various molecules. Each dataset consists of chemical compounds represented as graphs, derived from SMILE strings. In these graphs, nodes represent atoms and edges represent chemical bonds. The datasets used to validate our approach were obtained from Kensert et al. (2021).

Retention time refers to the time a compound remains in the stationary phase of a chromatography system before being detected, indicating its identity and concentration (Bouwmeester et al., 2019). The datasets used in this experiment are HILIC, RIKEN and SMRT. HILIC stands for Hydrophilic Interaction Liquid Chromatography, RIKEN refers to RIKEN plant specialized metabolome annotation, and SMRT stands for Small Molecule Retention Time. Both the RIKEN and SMRT datasets were acquired under Reversed-Phase Liquid Chromatography (RPLC) conditions. A summary of the three dataset characteristics including the number of molecules, the number of nodes and the retention time ranges in minutes is shown in Table 7.1

Table 7.1. Summary of dataset characteristics.

Database	Number of Molecules	Graph sizes	Retention time (min)
HILIC	1400	4-91	0.89-10.28
RIKEN	862	8-100	1.52-10.40
SMRT	77980	8-50	5.67-24.53

7.3.2 Architecture configuration

The input data, consisting of SMILE strings, was first converted into graph representations. In our experiments, we used two GCNs, each with 256 neurons, residual connections, and ReLU activation functions, followed by a readout layer. After that, three Fully Connected dense layers were employed, comprising 1024, 1024, and 1 neuron, respectively, all utilizing ReLU activation functions (Figure 7.1).

The training occured over 50 epochs for the HILIC and RIKEN datasets, and over 150 epochs for the SMRT dataset, utilizing batch sizes of 32 for the HILIC and RIKEN datasets, and 128 for the SMRT dataset. The training and validation was done using 90% of the data, with approximately 90% of that subset used for training and the remaining 10% for validation. The remaining 10% of the data was reserved for testing. We used the Adam optimizer and a learning rate scheduler that reduced the learning rate by a factor of 0.1 upon plateauing of validation loss, with a minimum learning rate of 10^{-6}. Early stopping was employed to halt training when no improvement in validation loss was observed for 10 consecutive epochs, with the best weights restored.

7.3.3 Predictive performance

To evaluate the overall performance of the model, the MSE, and coefficient of determination (R^2) were calculated on all datasets for both the training and testing sets, detailed in Table 7.2. The RIKEN dataset achieved a training MSE of 0.64 and a testing MSE of 0.70, indicating robust model performance in terms of MSE. In terms of R^2 values, both RIKEN and SMRT datasets indicate strong predictive capabilities. Specifically, the RIKEN dataset achieved an R^2 of 0.84 for training and 0.80 for testing, while the SMRT dataset achieved an R^2 of 0.88 for training and 0.84 for testing. The HILIC dataset also showed consistent performance, with a training MSE of 2.14 and a testing MSE of 2.11, and R^2 values of 0.71 and 0.69 for training and testing, respectively. Figure 7.3 shows the comparison between the predicted and actual retention time values across the entire dataset for each of the three datasets.

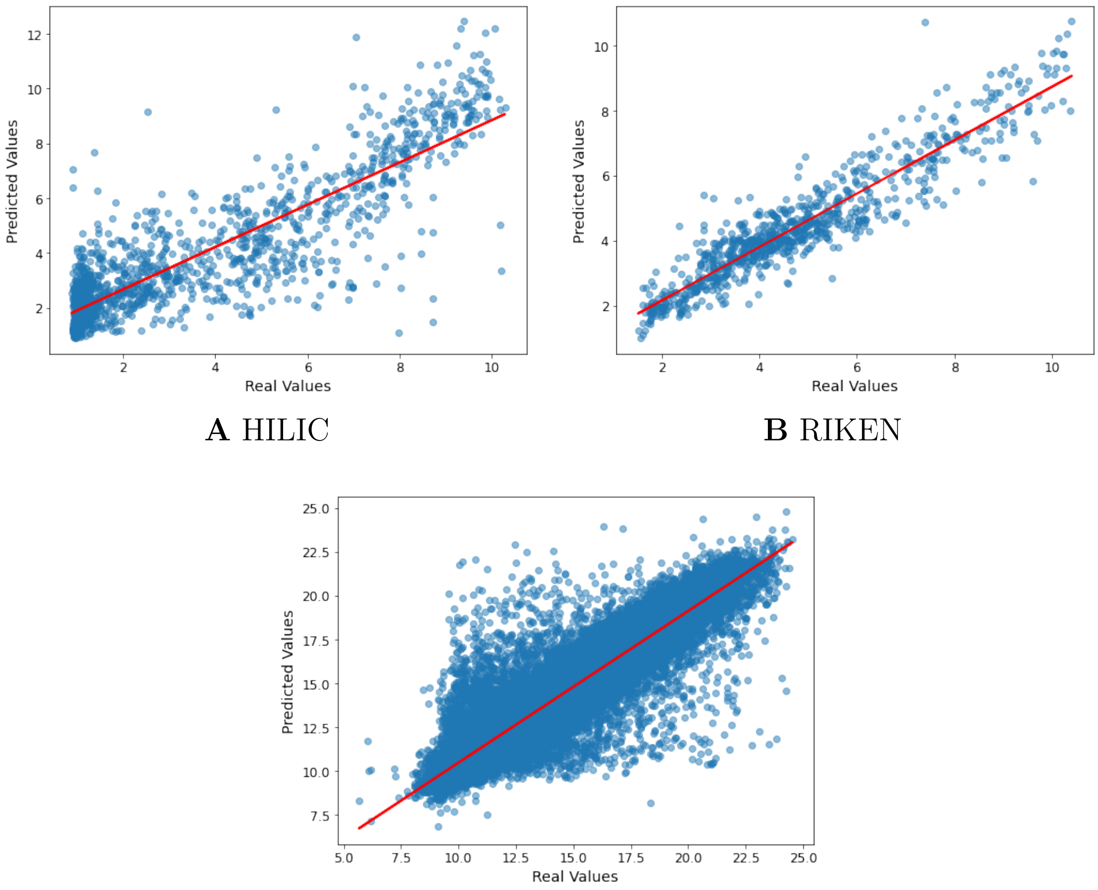
Table 7.2. MSE and R^2 of training and testing sets.

Dataset		**MSE**	**R^2**
HILIC	Train	2.14	0.71
	Test	2.11	0.69
RIKEN	Train	0.64	0.84
	Test	0.70	0.80
SMRT	Train	0.94	0.88
	Test	1.34	0.84

7.3.4 STM evaluation

We selected two molecules to visualize the saliency maps. As shown in Figure 7.4, the Grad-CAM SMG method for the HILIC dataset (graphs *a* and *g*) show inverted red and green regions compared to the other datasets (graphs *b, c,* and *h, i*) in the same row. Specifically, graph *a* and *g* highlights a positive contribution of the more polar atoms, whereas the other graphs in the same rows (RIKEN and SMRT datasets) show a negative contribution from these polar atoms. The remaining shared molecules exhibit the same behavior.

These findings align with the chemical insights derived from the saliency maps. This is because the chromatographic mechanism (RPLC) for the RIKEN and SMRT



A HILIC

B RIKEN

C SMRT

Figure 7.3. Comparison of predicted retention time (vertical axis) against real retention time (horizontal axis) for the HILIC, RIKEN, and SMRT datasets. The red dotted line represents the regression line of the data.

datasets operate on similar principles, while HILIC operates on opposite principles as such, polar functional groups will contribute positively to retention in HILIC.

In contrast, Grad-CAM++ highlights only the important regions, all shown in green, without indicating the direction of their importance. Grad-CAM++ was originally created to improve Grad-CAM in image recognition involving multiple objects in the same image. However, when Grad-CAM++ is applied to single chemical graphs, its perfmorance does not surpass that of Grad-CAM.

Understanding the predictions generated by deep learning models is crucial for their practical application in real-world contexts. Visualizations of saliency maps hold promise in improving model interpretability, but understanding these maps often requires additional analysis and domain-specific knowledge. By comparing the performance metrics with the insights provided by the saliency maps, we assess the efficacy of Grad-CAM and Grad-CAM++ across various datasets. Table 7.3 presents the evaluation metrics for the HILIC, RIKEN, and SMRT datasets using Grad-CAM and Grad-CAM++ SMG methods. The results indicate that Grad-CAM is generally more sensitive in capturing significant features across the datasest, as illustrated by the saliency maps. This consistency between the visual saliency maps and the SMG performance metrics provides an additional layer of analysis. Specifically, Grad-CAM exhibits higher IC, DC, and ADD values in the RIKEN and SMRT datasets. In contrast, for the HILIC dataset, Grad-CAM++ demonstrates a higher IC value, while the DC and ADD values are similar on the two methods.

Table 7.3. SMG performance metrics.

Database	SMG	IC	AD	DC	ADD
HILIC	**GradCAM**	0.14	0.72	0.63	0.39
	GradCAM++	0.41	0.49	0.64	0.42
RIKEN	**GradCAM**	0.17	0.74	0.97	0.80
	GradCAM++	0.02	0.82	0.79	0.56
SMRT	**GradCAM**	0.14	0.72	0.97	0.82
	GradCAM++	0.09	0.77	0.85	0.64

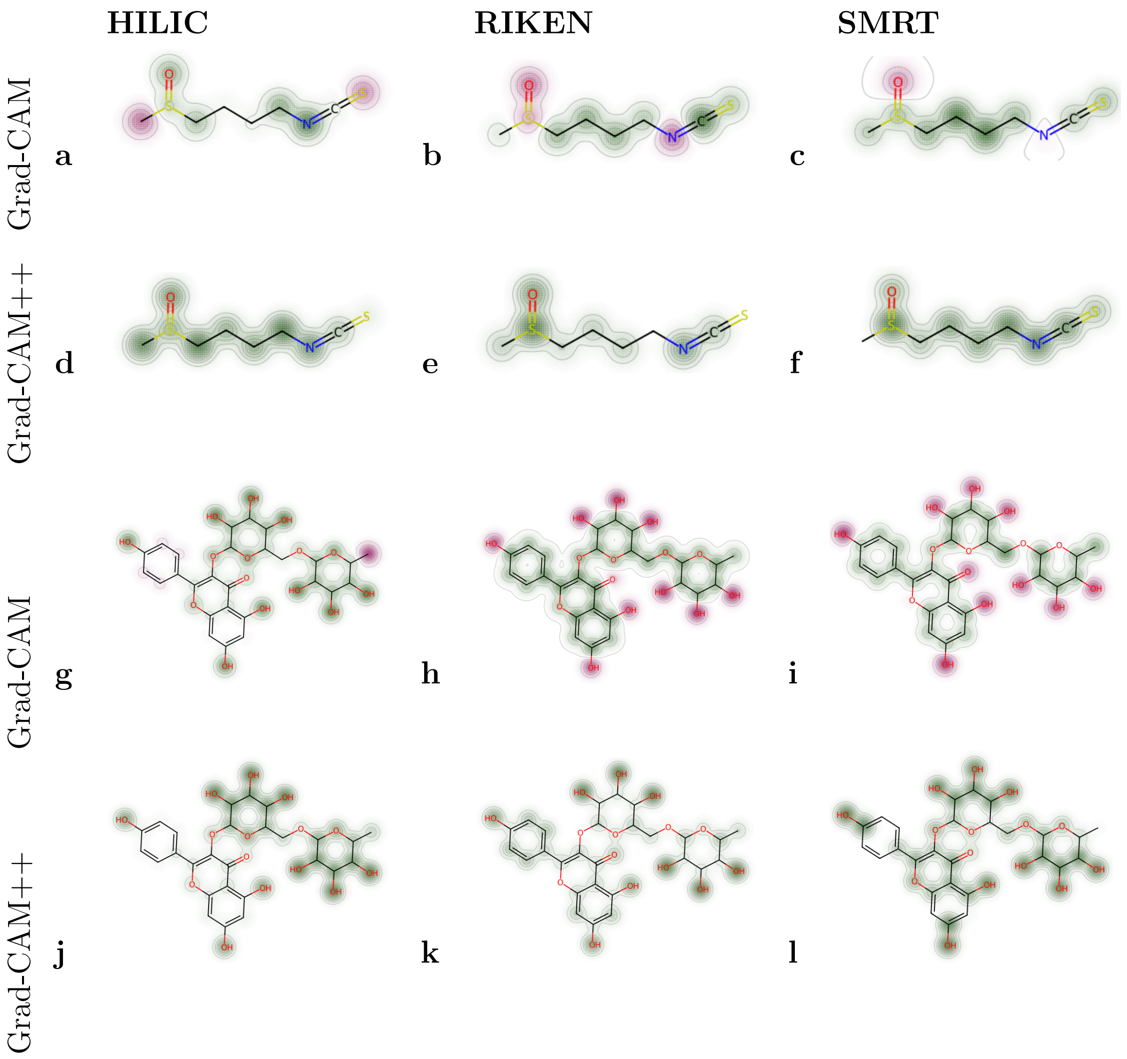


Figure 7.4. Saliency maps generated using Grad-CAM (first and third row) and GradCAM++ (second and fourth row) methods for HILIC (first column), RIKEN (second column), and SMRT (third column). The green regions indicate positive contributions and the red regions indicating negative contributions. The number of contour lines indicates the extent of influence, with a maximum of 10 contour lines per node. The first and second rows depict the molecule CS(=O)CCCCN=C=S, while the third and fourth rows depict the molecule CC1C(C(C(C(O1) OCC2C(C(C(C(O2)OC3=C(OC4=CC (=CC(=C4C3=O)O)O)C5= CC=C(C=C5)O)O)O)O)O)O)O.

7.4 Conclusions

The four Single Step Metrics (STM) presented in this chapter quantitatively evaluate the performance of Saliency Map Generators (SMG) in graph regression tasks. STM uniquely applies to regression studies, whether involving image or graph data.

The experimental validation shows that the results extracted from the STM are similar to the insights obtained from SMGs. This validation underscores the reliability of STM in assessing the performance of SMGs.

The comparison of Grad-CAM and Grad-CAM++ across three chemical datasets using STM demonstrates that Grad-CAM is more suited for analyzing graph-based chemical data, supported by evaluations from both SMG and STM.

From a practical point of view, these metrics offer the advantage of facilitating post-hoc analysis with minimal domain-specific knowledge required. This accessibility enhances their utility in interpreting model outputs in various applications.

8

General Conclusions

The general conclusions of this thesis are as follows:

The accurate representation of evolutionary relationships using phylogenetic trees is essential for clear and visual understanding of the elements under study.

Distance-based methods such as Unweighted Pair-Group Method with Arithmetic Mean (UPGMA) and Neighbour-joining (NJ) are widely used to generate phylogentic trees in various scientific fields, including biological and evolutionary sciences.

Both UPGMA and NJ suffer from the ties in proximity problem, which can lead to non-unique phylogenetic trees from a distance data matrix. This problem affects the reliability and interpretation of phylogenetic trees.

The issue of tied distances is not limited to biological sciences but extends to other research fields where distance-based methods are used.

The presence of non-unique phylogenetic trees can lead to ambiguities and misinterpretations in published research, impacting not only the conclusions of the original studies but also subsequent research based on these publications.

Experimental validation indicated that 46% of articles using the UPGMA method and 13% of articles using the NJ method had at least one alternative solution to the published binary phylogenetic tree. This indicates that a substantial proportion of scientific research is affected by te tied distances problem.

The development of the Multifurcating Neighbor-Joining (MFNJ) algorithm, a generalized version of NJ, addresses the tied distances problem in phylogenetic tree construction, ensuring consistent and unique tree outputs regardless of input order.

Graph Convolutional Networks (GCNs) excel in learning from graph-structured data by using node features and connections. This capability extends the utility of Deep Convolutional Networks beyond traditional data formats to graphs, allowing them to model complex relationships inherent in various fields.

In chemistry, GCNs facilitate predictive modeling and analysis by interpreting molecular structures as graphs.

Saliency Map Generators (SMGs), integrated with GCNs, play a key role in enhancing model interpretability by highlighting influential features like specific atoms or bonds in chemical structures, helping in understanding model predictions and decision-making processes.

Advances in SMGs, such as Grad-CAM and Grad-CAM++, have improved the interpretability of GCN outputs by visualizing feature importance in graph-based data, bridging the gap between model performance and actionable insights.

Single Step Metrics (STM) provide a robust quantitative assessment of SMGs in graph regression tasks, applicable to both image and graph data.

STM's minimal domain-specific knowledge requirement enhances its practical utility for interpreting deep learning model outputs across diverse applications.

9

List of publications and conferences

9.1 Publications

- Nonunique UPGMA clusterings of microsatellite markers.
  Natàlia Segura-Alabart, F. Serratosa, S. Gómez, A. Fernández.
  Briefings in Bioinformatics, 22 (5), 1-7. 2022.
  https://doi.org/10.1093/bib/bbac312

- The MultiFurcating Neighbor-Joining Algorithm for Reconstructing Polytomic Phylogenetic Trees.
  A. Fernández, **Natàlia Segura-Alabart**, F. Serratosa.
  Journal of Molecular Evolution, 91 (6), 773–779. 2023.
  https://doi.org/10.1007/s00239-023-10134-z

- A practical study of the proportion of non-unique neighbor-joining trees of microsatellite markers.

Natàlia Segura-Alabart, A. Fernández, F. Serratosa.

Computational and Structural Biotechnology Reports. 2024.

Under revision.

9.2 Conferences

- Nonunique UPGMA clusterings of microsatellite markers.

  Natàlia Segura-Alabart, F. Serratosa, S. Gómez, A. Fernández.

  X Jornada de Bioinformàtica i Genòmica. 2022.

  València, Spain.

  `https://scb.iec.cat/wp-content/uploads/2022/12/BookOfAbstracts.pdf`

- Prevalence of nonunique dendrograms in agglomerative clustering algorithms.

  Natàlia Segura-Alabart.

  8th URV Doctoral Workshop in Computer Science and Mathematics. 2023.

  Tarragona, Spain.

  `https://llibres.urv.cat/index.php/purv/catalog/book/566`

- Splitting Structural and Semantic Knowledge in Graph Autoencoders for Graph Regression.

  13th IAPR-TC15 International Workshop on Graph-Based Representations in Pattern Recognition (GbRPR)

  S. Fadlallah, **Natàlia Segura-Alabart**, C. Julià, F. Serratosa. 2023.

  Salerno, Italy.

  `https://link.springer.com/book/10.1007/978-3-031-42795-4`

- Use of saliency maps in chemestry graph regression models.

  Natàlia Segura-Alabart.

  9th URV Doctoral Workshop in Computer Science and Mathematics, 2024.

  Tarragona, Spain.

  To be published.

- Evaluation metrics in Saliency Maps applied to Graph Regression.

Natàlia Segura-Alabart, A. Fernández, A.Kensert, D. Cabooter, F. Serratosa.

20th Statistical Techniques in Pattern Recognition and Structural and Syntactic Pattern Recognition (S+SSPR), 2024.

Venice, Italy.

Under revision.

Bibliography

Arnau, V., Mars, S., and Marín, I. (2005). Iterative cluster analysis of protein interaction data. *Bioinformatics*, 21(3):364–378.

ASCLEPIUS-URV (2024). Supplementary material of a practical study of the proportion of non-unique neighbour-joining trees of microsatellite markers. `https://github.com/ASCLEPIUS-URV`. Accessed: 2024.

Ates, D., Altinok, H., Ozkuru, E., Ferik, F., Erdogmus, S., Can, C., and Tanyolac, M. (2019). Population structure and linkage disequilibrium in a large collection of Fusarium oxysporum strains analysed through iPBS markers. *J Phytopathol*, 167(10):576–590.

Atteson, K. (1999). The performance of neighbor-joining methods of phylogenetic reconstruction. *Algorithmica*, 25:251–278.

Aziz, D., Siraj, S., and Arshad, A. (2020). Genetic diversity of banana prawns Fenneropenaeus merguiensis in Malaysian waters using microsatellite markers. *J Environ Biol*, 41(5):1349–1357.

Backeljau, T., De Bruyn, L., De Wolf, H., Jordaens, K., Van Dongen, S., and Winnepenninckx, B. (1996). Multiple UPGMA and neighbor-joining trees and the performance of some computer packages. *Mol Biol Evol*, 13(2):309–313.

Bertrand, P. (1995). Structural properties of pyramidal clustering. In *Cox I, Hansen P, Julesz B (eds) Partitioning data sets. American Mathematical Society, Providence*, volume 19, pages 35–53. DIMACS Series in Discrete Mathematics and Theoretical Computer Science.

Bhambra, P., Joachimi, B., and Lahav, O. (2022). Explaining deep learning of galaxy morphology with saliency mapping. *Monthly Notices of the Royal Astronomical Society*, 511(4):5032–5041.

Bouwmeester, R., Martens, L., and Degroeve, S. (2019). Comprehensive and empirical evaluation of machine learning algorithms for small molecule lc retention time prediction. *Analytical Chemistry*, 91(5):3694–3703.

Brondani, R., Brondani, C., Tarchini, R., and Grattapaglia, D. (1998). Development, characterization and mapping of microsatellite markers in Eucalyptus grandis and E. urophylla. *Theor Appl Genet*, 97:816–827.

Bruno, W. J., Socci, N. D., and Halpern, A. L. (2000). Weighted neighbor joining: a likelihood-based approach to distance-based phylogeny reconstruction. *Molecular biology and evolution*, 17(1):189–197.

Bryant, D. and Moulton, V. (2004). Neighbor-Net: An Agglomerative Method for the Construction of Phylogenetic Networks. *Molecular Biology and Evolution*, 21(2):255–265.

Buneman, P. (1974). A note on the metric properties of trees. *J Comb Theory B*, 17(1):48–50.

Chapal-Ilani, N., Maruvka, Y., Spiro, A., Reizel, Y., Adar, R., Shlush, L., and Shapiro, E. (2013). Comparing algorithms that reconstruct cell lineage trees utilizing information on microsatellite mutations. *PLoS Comput Biol*, 9(11):1–17.

Chattopadhay, A., Sarkar, A., Howlader, P., and Balasubramanian, V. N. (2018). Grad-cam++: Generalized gradient-based visual explanations for deep convolutional networks. In *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 839–847.

Cilibrasi, R. and Vitányi, P. (2005). Clustering by compression. *IEEE Trans Inf Theory*, 51(4):1523–1545.

Diday, E. (1987). Orders and overlapping clusters by pyramids. *Rapports de Recherche*, 730.

Ditta, A., Zhou, Z., Cai, X., Wang, X., Okubazghi, K., Shehzad, M., Xu, Y., Hou, Y., Iqbal, M., Khan, M., Wang, K., and Liu, F. (2018). Assessment of genetic diversity, population structure, and evolutionary relationship of uncharacterized genes in a novel germplasm collection of diploid and allotetraploid Gossypium accessions using EST and genomic SSR markers. *Int J Mol Sci*, 19(8):2401.

Durbin, R., Eddy, S., Krogh, A., and Mitchison, G. (1998). *Biological sequence analysis: probabilistic models of proteins and nucleic acids*. Cambridge university press.

Ebrahimi, M., Mohammadabadi, M., and Esmailizadeh, A. (2017). Using microsatellite markers to analyze genetic diversity in 14 sheep types in Iran. *Arch Anim Breed*, 60(3):183–189.

Elias, I. and Lagergren, J. (2009). Fast neighbor joining. *Theoretical Computer Science*, 410:1993–2000.

Ellegren, H. (2004). Microsatellites: simple sequences with complex evolution. *Nat Rev Genet*, 5(6):435–445.

Evans, J., Sheneman, L., and Foster, J. (2006). Relaxed neighbor joining: A fast distance-based phylogenetic tree construction method. *Journal of Molecular Evolution*, 62:785–792.

Fernández, A. (2023). mphylo: multifurcated phylogenetic trees in R. Accessed 23 December 2023.

Fernández, A. and Gómez, S. (2007). Solving non-uniqueness in agglomerative hierarchical clustering using multidendrograms. *J Classif*, 25:43–65.

Fernández, A. and Gómez, S. (2020). Versatile linkage: a family of space-conserving strategies for agglomerative hierarchical clustering. *J Classif*, 37:584–597.

Fernández, A., Segura-Alabart, N., and Serratosa, F. (2023). The multifurcating neighbor-joining algorithm for reconstructing polytomic phylogenetic trees. *J Mol Evol*, (in press).

Gascuel, O. (1994). A note on Sattath and Tversky's, Saitou and Nei's, and Studier and Keppler's algorithms for inferring phylogenies from evolutionary distances. *Molecular Biology and Evolution*, 11(6):961–963.

Gascuel, O. (1997). Bionj: an improved version of the nj algorithm based on a simple model of sequence data. *Molecular biology and evolution*, 14(7):685–695.

Gascuel, O. and Steel, M. (2006). Neighbor-joining revealed. *Molecular Biology and Evolution*, 23:1997–2000.

Gomez, T. and Mouchère, H. (2023). Computing and evaluating saliency maps for image classification: a tutorial. *Journal of Electronic Imaging*, 32:020801.

Grishin, V. and Grishin, N. (2002). Euclidian space and grouping of biological objects. *Bioinformatics*, 18(11):1523–1534.

Gómez, S. and Fernández, A. (2021). Radatools 5.2: communities detection in complex networks and other tools.

Gómez, S., Fernández, A., Granell, C., and Arenas, A. (2013). Structural patterns in complex systems using multidendrograms. *Entropy*, 15(12):5464–5474.

Han, Z., Mo, Q., Liu, Y., and Zuo, M. (2010). Constructing taxonomy by hierarchical clustering in online social bookmarking. In *2010 International Conference on Educational and Information Technology (ICEIT 2010) Constructing*, volume 3, pages 47–51. IEEE.

Hariri, M., Chikmawati, T., and Hartana, A. (2017). Genetic diversity of Indigofera tinctoria L. in Java and Madura islands as natural batik dye based on intersimple sequence repeat markers. *J Math Fund Sci*, 49(2):105–115.

Hart, G. (1983). The occurrence of multiple UPGMA phenograms. *Numerical Taxonomy*, 1:254–258.

Hong, Y., Guo, M., and Wang, J. (2021). ENJ algorithm can construct triple phylogenetic trees. *Mol Ther Nucleic Acids*, 23(5):286–293.

Hormaza, J. (2002). Molecular characterization and similarity relationships among apricot (Prunus armeniaca L.) genotypes using simple sequence repeats. *Theor Appl Genet*, 104(2-3):321–328.

Hossain, S., Habiba, U., Bhuyan, S., Haque, M., Begum, S., and Hossain, D. (2014). DNA fingerprinting and genetic diversity analysis of chilli germplasm using microsatellite markers. *Biotechnology*, 13(4):174–180.

Howe, K., Bateman, A., and Durbin, R. (2002). Quicktree: building huge neighbour-joining trees of protein sequences. *Bioinformatics*, 18:1546–1547.

Ismail, N., Rafii, M., Mahmud, T., Hanafi, M., and Miah, G. (2016). Molecular markers: a potential resource for ginger genetic diversity studies. *Mol Biol Rep*, 43:1347–1358.

Jiang, T., Kearney, P., and Li, M. (2001). A polynomial time approximation scheme for inferring evolutionary trees from quartet topologies and its application. *SIAM J Comput*, 30(6):1942–1961.

Jombart, T. (2008). adegenet: a R package for the multivariate analysis of genetic markers. *Bioinformatics*, 24(11):1403–1405.

Jombart, T. and Ahmed, I. (2011). adegenet 1.3-1: new tools for the analysis of genome-wide SNP data. *Bioinformatics*, 27(21):3070–3071.

Jung, H. and Oh, Y. (2021). Towards better explanations of class activation mapping. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 1316–1324.

Kensert, A., Bouwmeester, R., Efthymiadis, K., Van Broeck, P., Desmet, G., and Cabooter, D. (2021). Graph convolutional networks for improved prediction and interpretability of chromatographic retention data. *Analytical Chemistry*, 93(47):15633–15641.

Kipf, T. N. and Welling, M. (2017). Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations*.

Lance, G. and Williams, W. (1966). A generalized sorting strategy for computer classifications. *Nature*, 212:218.

Leal, W., Llanos, E., Restrepo, G., Suárez, C., and Patarroyo, M. (2016). How frequently do clusters occur in hierarchical clustering analysis? A graph theoretical approach to studying ties in proximity. *J Cheminformatics*, 8(4).

Lemey, P., Salemi, M., and Vandamme, A.-M. (2009). *The Phylogenetic Handbook: A Practical Approach to Phylogenetic Analysis and Hypothesis Testing.* Cambridge University Press, 2 edition.

Levy, D., Yoshida, R., and Pachter, L. (2005). Beyond Pairwise Distances: Neighbor-Joining with Phylogenetic Diversity Estimates. *Molecular Biology and Evolution*, 23(3):491–498.

Li, H., Chappell, M., and Zhang, D. (2020). Assessing genetic diversity and population structure of Kalmia latifolia L. in the eastern United States: an essential step towards breeding for adaptability to southeastern environmental conditions. *Sustainability*, 12(19):8284.

Li, Z., Gichira, A., Wang, Q., and Chen, J. (2018). Genetic diversity and population structure of the endangered basal angiosperm Brasenia schreberi (Cabombaceae) in China. *PeerJ*, 6:e5296.

MacCuish, J., Nicolaou, C., and MacCuish, N. (2001). Ties in proximity and clustering compounds. *J Chem Inf Comput Sci*, 41:134–146.

Maechler, M., Rousseeuw, P., Struyf, A., Hubert, M., and Hornik, K. (2021). *cluster: cluster analysis basics and extensions.* R package version 4.2.0.

Mailund, T. and Pedersen, C. (2004). QuickJoin—fast neighbour-joining tree reconstruction. *Bioinformatics*, 20(17):3261–3262.

MATLAB (2010). *version 7.10.0 (R2010a).* The MathWorks Inc., Natick, Massachusetts.

McTavish, E., Drew, B., Redelings, B., and Cranston, K. (2017). How and why to build a unified tree of life. *BioEssays*, 39(11).

Mohammad, B., Daghistani, H., Jaouani, A., Abdel-Latif, S., and Kennes, C. (2017). Isolation and characterization of thermophilic bacteria from Jordanian hot springs: Bacillus licheniformis and Thermomonas hydrothermalis isolates as potential producers of thermostable enzymes. *Int J Microbiol*, 2017.

Moiana, L., Filho, P., Gonçalves-Vidigal, M., Lacanallo, G., Galván, M., De Carvalho, L., Maleia, M., Pacheco, C., Ribeiro, T., Neto, H., and Coimbra, G. (2012). Genetic diversity and population structure of cotton (Gossypium hirsutum L. race latifolium H.) using microsatellite markers. *Afr J Biotechnol*, 11(54):11640–11647.

Nei, M. (1978). Estimation of average heterozygosity and genetic distance from a small number of individuals. *Genetics*, 89(3):583–590.

Nicolaou, C., MacCuish, J., and Tamura, S. (2000). A new multi-domain clustering algorithm for lead discovery that exploits ties in proximities. In *Proceedings from the 13th European Symposium on Quantitative Structure–Activity Relationships*, pages 486–495. Prous Science, Barcelona.

Ota, S. and Li, W.-H. (2000). Njml: a hybrid algorithm for the neighbor-joining and maximum-likelihood methods. *Molecular Biology and Evolution*, 17(9):1401–1409.

Park, S. and Kim, J. (2016). Trends in next-generation sequencing and a new era for whole genome sequencing. *Int Neurourol J*, 20(2):76–83.

Pearson, W. R., Robins, G., and Zhang, T. (1999). Generalized neighbor-joining: more reliable phylogenetic tree reconstruction. *Molecular Biology and Evolution*, 16(6):806–816.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al. (2011). Scikit-learn: machine learning in Python. *J Mach Learn Res*, 12(Oct):2825–2830.

Podani, J. (1997). On the sensitivity of ordination and classification methods to variation in the input order of data. *J Veg Sci*, 8:153–156.

Pope, P. E., Kolouri, S., Rostami, M., Martin, C. E., and Hoffmann, H. (2019). Explainability methods for graph convolutional neural networks. In *2019 IEEE/CVF*

Conference on Computer Vision and Pattern Recognition (CVPR), pages 10764–10773.

Powell, W., Morgante, M., Andre, C., Hanafey, M., Vogel, J., Tingey, S., and Rafalski, A. (1996). The comparison of RFLP, RAPD, AFLP and SSR (microsatellite) markers for germplasm analysis. *Mol Breed*, 2:225–238.

R Core Team (2021a). *R: a language and environment for statistical computing.* R Foundation for Statistical Computing, Vienna, Austria. R version 4.1.0.

R Core Team (2021b). *The R stats package.* R package version 4.2.0.

Randi, E., Gentile, L., Boscagli, G., Huber, D., and Roth, H. (1994). Mitochondrial dna sequence divergence among some west european brown bear (ursus arctos l.) populations. lessons for conservation. *Heredity*, 73:480–489.

Roche, D., Lanfear, R., Binning, S., Haff, T., Schwanz, L., Cain, K., Kokko, H., Jennions, M., and Kruuk, L. (2014). Troubleshooting public data archiving: suggestions to increase participation. *PLoS Biol*, 12(1):e1001779.

Saitou, N. (1986). *Theoretical studies on the methods of reconstructing phylogenetic trees from DNA sequence data.* PhD thesis, University of Texas. Health Science Center at Houston. Graduate School of Biomedical Sciences.

Saitou, N. (2018). *Introduction to Evolutionary Genomics.* Springer, Cham.

Saitou, N. and Nei, M. (1987). The neighbor-joining method: a new method for reconstructing phylogenetic trees. *Mol Biol Evol*, 4(4):406–425.

Segura-Alabart, N., Serratosa, F., Gómez, S., and Fernández, A. (2022). Nonunique UPGMA clusterings of microsatellite markers. *Brief Bioinform*, 23(5):bbac312.

Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., and Batra, D. (2017). Grad-cam: Visual explanations from deep networks via gradient-based localization. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 618–626.

Silva, A., Villanueva, W., Knidel, H., Bonato, V., Reis, S., and Von Zuben, F. (2005). A multi-neighbor-joining approach for phylogenetic tree reconstruction and visualization. *Genetics and Molecular Research*, 4:525–534.

Steel, M. (1992). The complexity of reconstructing trees from qualitative characters and subtrees. *J Classif*, 9:91–116.

Studier, J. and Kepplter, K. (1988). A note on the neighbor-joining algorithm of Saitou and Nei. *Mol Biol Evol*, 5(6):729–731.

Tautz, D. (1989). Hypervariability of simple sequences as a general source for polymorphic DNA markers. *Nucleic Acids Res*, 17(16):6463–6471.

Telles, G., Araújo, G., Walter, M., Brigido, M., and Almeida, N. (2018). Live neighbor-joining. *BMC Bioinformatics*, 19:172.

Vieira, M., Santini, L., Diniz, A., and Munhoz, C. (2016). Microsatellite markers: what they mean and why they are so useful. *Genet Mol Biol*, 39(3):312–328.

Vines, T., Andrew, R., Bock, D., Franklin, M., Gilbert, K., Kane, N., Moore, J., Moyers, B., Renaut, S., Rennison, D., Veen, T., and Yeaman, S. (2013). Mandated data archiving greatly improves access to research data. *FASEB J*, 27(4):1304–1308.

Virtanen, P., Gommers, R., Oliphant, T., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., van der Walt, S., Brett, M., Wilson, J., Millman, K., Mayorov, N., Nelson, A., Jones, E., Kern, R., Larson, E., Carey, C., Polat, I., Feng, Y., Moore, E., VanderPlas, J., Laxalde, D., Perktold, J., Cimrman, R., Henriksen, I., Quintero, E., Harris, C., Archibald, A., Ribeiro, A., Pedregosa, F., van Mulbregt, P., and Contributors, S. . (2020). SciPy 1.0: fundamental algorithms for scientific computing in Python. *Nat Methods*, 17:261–272.

Wang, F., Yang, T., Burlyaeva, M., Li, L., Jiang, J., Fang, L., Redden, R., and Zong, X. (2015). Genetic diversity of grasspea and its relative species revealed by SSR markers. *PLoS One*, 10(3).

Williams, J., Kubelik, A., Livak, K., Rafalski, J., and Tingey, S. (1990). DNA polymorphisms amplified by arbitrary primers are useful as genetic markers. *Nucleic Acids Res*, 18(22):6531–6535.

Yu, G. (2020). Using ggtree to visualize data on tree-like structures. *Curr Protoc Bioinformatics*, 69:e96.

Zhang, J., Petitjean, C., Yger, F., and Ainouz, S. (2020). Explainability for regression cnn in fetal head circumference estimation from ultrasound images. In *Interpretable and Annotation-Efficient Learning for Medical Image Computing*, pages 73–82.

Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., and Torralba, A. (2016). Learning deep features for discriminative localization. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2921–2929.

UNIVERSITAT
ROVIRA i VIRGILI