Fraunhofer-Institut für Fabrikbetrieb und
-automatisierung IFF

Logistics and Factory Systems Department

Time invariant Product features identification through Phylogenetic approach

Authors:
Sai Dheeraj Reddy, Tippani

July 28, 2022

Supervisor:
Dipl.-Phys. Bastian Sander, M.Sc.
Logistic and Factory Systems
Fraunhofer-Institut für Fabrikbetrieb und -automatisierung IFF
Sandtorstr. 22
39106 Magdeburg, Germany

Acknowledgements

I would like to express my deepest gratitude towards my supervisor Dipl.-Phys. Bastian Sander, M.Sc. It was his constant guidance that has steered the research in the current direction and I will always be thankful to him, for spending a significant amount of time guiding me through many challenges and many hindrances that have occurred during the research. And I thank him for immensely helping me in learning the research methodology.

Contents

1

Introduction and Motivation

1.1 Motivation

Identification of a product and material is one of the most fundamental step in every single stage of the production. Traditional way of identification is done by various methods. Some methods involve imprinting the associating ID or a code on the product’s surface and the product is identified by scanning or reading this unique ID. Other methods like RFID sensors are prevalent in most of the industries but they have their own set of drawbacks. Some of them include the issue of limited range and interference of surrounding materials on the performance, lower reliability when compared to bar code scanning, and also it costs more than other methods.

The most ideal version of any kind of identification must not involve any kind of contamination of the product or it’s surface in any manner. This project deals with researching such an ideal solution and problems that arise while solving one of the most ubiquitous problem in a production pipeline.

The idea is to utilise several kinds of ’Non Destructive Techniques’ and ’sensors’ to collect as much information as possible from a product. The information collected is consolidated to act as a feature set that can be used to uniquely identify the product. In other words, extracting the "DNA" of an object.

The biggest problem with such a setting is that once the object/product is put into usage in the environment where it is designed to be, it’s features/"DNA" might change due to environmental conditions it is subjected to. The challenge is to be able to account for such variations in the features and still be able to correctly identify the source of the object/product. In this project it is attempted to solve for such challenges by taking inspiration from the field of Biology.

1.2 Structure of this report

In Chapter 2, A detailed explanation of the ’Phylogeny based approach’ is given. It is attempted to give a clear reasoning for the choices and steps taken as detailed as possible for better understanding. The chapter also contains brief description of several of the concepts and corresponding terminology that will be used often in the coming chapters.

Chapter 3 is about the experimental setup and sensors used to obtain real world data of some sample materials. Further work will be focused around the experiment data collected. This chapter also discusses how the data collected is applied in the ’Phylogenetic analysis’ pipeline.

In Chapter 4, The pipeline for generating a maximum parsimonious tree and a detailed description of each of the involving steps.

In Chapter 5, it is attempted to analyse and understand the results obtained in the previous chapter. This step helps in building a robust model for identifying objects. Understanding the reasoning behind the results is crucial in scaling up the ’Phylogenetic analysis’ approach into a bigger model.

Many directions were not taken during the project for better utilisation of the time and other resources. Such potentially useful directions and future scope will be discussed briefly in chapter 6.

A brief conclusion for the project will be given in Chapter 7.

2
Phylogenetic approach

2.1 What is the aim?

The aim is to be able to Predict the object's ancestors based on the features collected from the object in the current time. In other words, to be able to identify if an object/product belongs to any known family of objects/products.

2.2 Why this approach?

Phylogenetics is a scientific methodology that deals with finding ancestral relationships between biological species based on information collected from organisms. There is a strong requirement in the field to have a robust mathematical technique in place in order to reliably find a relationship between two organisms. Phylogenetics is the name of the field that deals with such mathematical techniques that define relationships between organismsALLABY and WOODWARK (2004) ID et al. (2019)

A biologist will normally have morphological (Physical features) or molecular data (DNA/protein sequences) of a certain organism in the current/latest time and they use 'Phylogenetic techniques' to identify the ancestral history and relationships of this organism WALLACE (2011)

This is analogous to the use case of this project goal. Given we know the 'DNA' of the object, which in our case is feature information collected from the object through various kinds of sensors and instruments, theoretically speaking, we should be able to use such mathematically rigorous phylogenetic techniques to identify the ancestral history of this object, which is, it's identity. Usage of phylogenetic techniques for non-biological applications is being more more prominent. Many of such research ideas are an inspiration to this research FRAIX-BURNET et al. (2017) FRAIX-BURNET (2017) RETZLAFF and STADLER (2018)

2.3 Initial exploration

For ease of explanation, there are two kinds of Phylogenetic analyses. the variations is based on the kinds of features that are collected from the organisms. One is Morphological phylogeny, the other Molecular phylogeny. The difference between the two is pretty straight forward, Morphological phylogeny deals with data like the physical features of the organism and it's behaviour CHANG (2004). Molecular phylogeny deals with DNA, RNA, Protein sequences obtained from the organism's body or blood. Molecular data like DNA sequences need some pre-processing to be done to perform phylogenetic analysis. There are several tools that are freely available that can perform such pre-processing steps. MegaX is one of such softwares.

There are several databases to obtain molecular data of any kind of species that humanity has discovered. Some of such open source databases such as "Assembly", "NCBI", "ClinVar", "Genome", "GTR" etc.,

2.4 Mega X

Mega X is a software that can be used for performing most common "Bioinformatic" operations HALL (2013) KUMAR. This software can be used to collect sequence data from the renowned databases, perform alignment operations, choose an appropriate algorithms to perform a detailed phylogenetic analysis and the final result would be a tree structure, that is a visual representation of the relationship between three or more species. Fig 3.1 represents an extremely simplified version of a phylogenetic analysis pipeline. Fig 3.2 represents a typical phylogenetic analysis pipeline with selected options marked green in color.

A detailed explanation of Phylogenetic analysis and usage of MEGA X software is present in the document "phylogenetic_analysis_using_MEGAX.pdf"

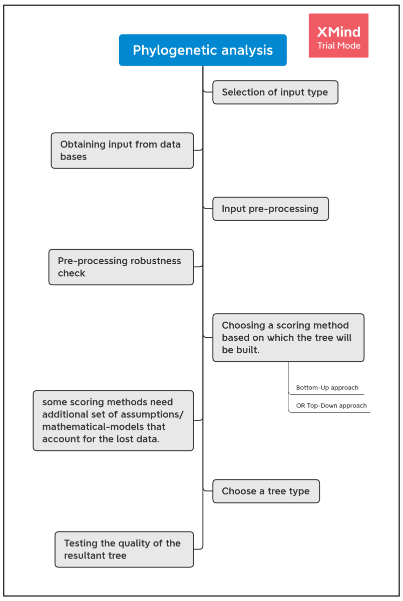

XMind
Trial Mode
Phylogenetic analysis
Selection of input type
Obtaining input from data bases
Input pre-processing
Pre-processing robustness check
Choosing a scoring method based on which the tree will be built.
Bottom-Up approach
OR Top-Down approach
some scoring methods need additional set of assumptions/ mathematical-models that account for the lost data.
Choose a tree type
Testing the quality of the resultant tree

Figure 2.1: Simplified Phylogenetic analysis pipeline**?**

2.5 Biopython

Biopython is a Python package that is built for performing 'Bioinformatics' operations like phylogenetics with ease CHANG et al. (2021). In this project this package is used extensively to perform phylogenetic analysis in all the stages.

Biopython makes the process of trial and error much smoother and faster because it is extremely easy to use and modular in nature. Using a preexisting rigid graphical user interface like MEGA X will become a limiting factor when the research demands frequent changes in the pipeline and analysis strategy. That is the primary reason for choosing Biopython as the major tool in this project.

Biopython is capable of performing sequence search from databases(like GENEBANK, NCBI, Entrez etc.,), searching for similar sequences from the databases given a query sequence (ClustalW, MUSCLE algorithms), pre processing (sequence alignments), tree generation and basic tree manipulation operations, visualisation and many more Bioinformatic analysis operations.

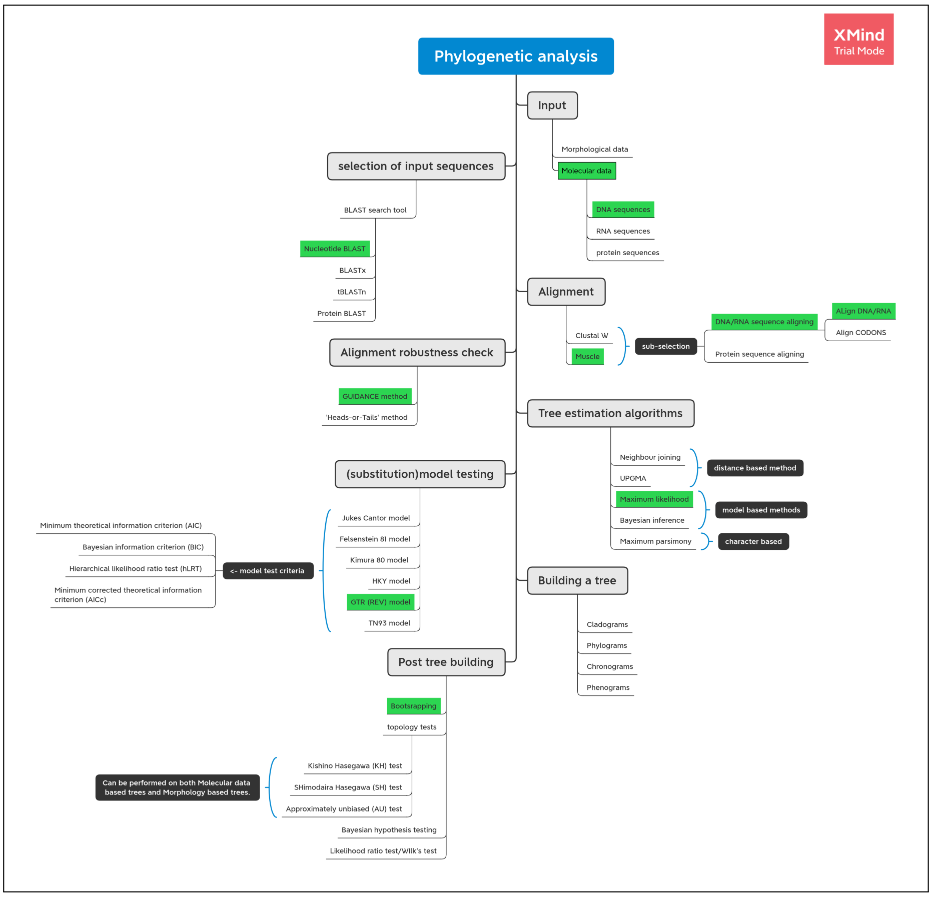


Figure 2.2: Typical Phylogenetic analysis pipeline**?**

As displayed in the fig 3.2, "Tree estimation algorithms" play an important role in the analysis. Because all the steps that come before this step can be considered as data preparation and pre-processing. The step of choosing the 'Tree estimation algorithm' will impact the final tree significantly. Although fig 3.2 is meant for explanation of the 'MEGA X' software, the pipeline is similar in case of any software or python package.

Some of the Tree estimation algorithms that Biopython supports are Neighbour Joining, UPGMA, Maximum Likelihood, Maximum Parsimony algorithms

Out of the above mentioned algorithms, Maximum parsimony is the only algorithm that is used for performing phylogenetics with both morphological data and molecular data.

2.6 Maximum Parsimony

Maximum parsimony algorithm will output a tree by minimising the number of evolutionary steps needed to explain a given aligned sequence or morphological data. This algorithm is the simplest of all in nature but it yields reasonable good results at greater speed when compared to the other algorithm YANG and RANNALA (2012).
In Biopython package, application of Maximum parsimony algorithm is simple and straight forward. But there is a problem that needs to be addressed, Biopython does not support morphological data types, this problem is circumvented by modifying existing substitution matrices in the Biopython library sourcecode.Fig 2.3 showcases the Maximum parsimony functionality in Biopython.

Biopython takes ".phy" as input. A .phy file contains molecular sequences of certain number of taxa(organisms). The output of the Parsimony tree constructor is most parsimonious phylogenetic tree.

To construct most parsimonious tree from just a molecular alignment data, Biopython needs to construct a distance matrix first, which in turn is dependent on the type of 'substitution model' we choose. Since substitution models do not make any sense when it comes to morphological data, this problem is circumvented by certain changes in the source code files. This will be discussed in greater detail in Chapter 6.

To build a parsimonious tree, the tree searcher algorithm (Nearest Neighbour Interchange algorithm)searches for the best possible tree in the tree space using the parsimony score assigned to each tree. Based on this score, the tree with least score is displayed as the most parsimonious tree. The scoring algorithm is called Fitch parsimony scoring algorithm. There are other variants available in Biopython, like, Sankoff parsimony scoring algorithm.

A tree searcher requires an initial tree to start the search from. Based on this initial tree, the task of searching huge and sparse tree space will be reduced significantly. This is the reason why Biopython needs to build a "Starting tree" or "Initiation tree" which is built using a distance based tree building algorithm called Neighbour Joining algorithm. Alternative to this algorithm is UPGMA (Unweighted Pair Group Method with Arithmetic

Mean) algorithm. To build an initial tree using a distance based algorithm, it needs a distance matrix. This distance matrix can only be obtained with a substitution matrix in case of Biopython even though we are dealing with morphological data in our case.

The type of data we are dealing with will be discussed in detail in the coming chapter. Chapter 5 will clarify the type of data that we will be needing to convert into a sequence in order to build a parsimonious tree in Biopython.

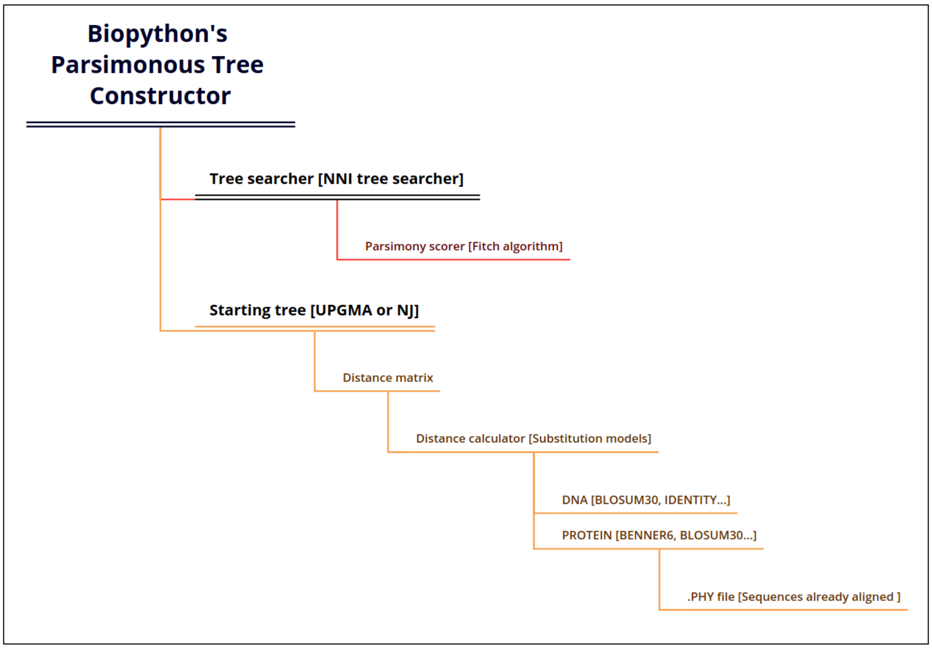

Biopython's Parsimonous Tree Constructor
Tree searcher [NNI tree searcher]
Parsimony scorer [Fitch algorithm]
Starting tree [UPGMA or NJ]
Distance matrix
Distance calculator [Substitution models]
DNA [BLOSUM30, IDENTITY...]
PROTEIN [BENNER6, BLOSUM30...]
.PHY file [Sequences already aligned]

Figure 2.3: Maximum parsimony pipeline in Biopython**?**

2.7 Role of Substitution matrices

A substitution matrix is a matrix format of displaying the evolutionary rate of change from one fundamental character to another. Each character signifies a specific protein type or certain nucleotide in DNA or RNA sequence AMIT ROY (2014). Some evolutionary changes are more common than others is the summary of a substitution matrix.

Such a matrix is necessary to be able to find the pairwise distance between bunch of taxonomies. Biologists usually choose amongst different types of substitution matrices based on the scale and purpose of the research being performed BENNET et al. (1994).

2.8 Role of 'initiation tree'

An initiation tree/ starting tree is the first tree that is built based on the substitution matrix and the distance matrix that is build using it. The step following the building of an initiation tree is a tree search algorithm which searches for the most parsimonious tree in the tree space. Initiation trees reduce the search space dramatically, making the parsimony method of building a phylogenetic tree and extremely fast way to build a phylogenetic tree.

3
Data collection

For the sake of application of phylogenetic algorithms on the real world data some measurements were taken of 16 samples of fabric material. Two instruments were used to perfomr this measurement on the sample materials, they are, Short wave Infrared Hyperspectroscopy camera (SWIR camera) and a "Visible and near infrared" Hyperspectroscopy camera (VNIR camera).

3.1 About the sensors

Hyper-spectroscopy cameras are extremely powerful cameras that are able to take picture in highly varying range of wavelengths of light as opposed to traditionally commercial cameras that are designed to capture only the visible range of light. Two of such cameras used in this project were,

3.1.1 Sensor1: Hyspex SWIR 384

This camera is capable of capturing light in the wavelength of 930-2500 nm. That is the reason for the name "Short-wave infrared" camera. This camera is used to capture the infrared behaviour of the surface of the material samples to a standard white-light. The reflectance of the materials are recorded by the SWIR camera.

3.1.2 Sensor2: Hyspex VNIR 1800

This camera deals with both visible and near infrared light with operating wavelength of light at 400 to 1000 nm. This camera will capture the reflectance behaviour of the material samples.

3.2 About the data

The analysis is performed on 16 different material samples. Each sample has a unique reflectance behaviour at different wavelengths of the incident white-light. Raw data looks like the Figure 4.2.

3.3 Post processing of data

In order to convert the raw numerical data into something that is useful for the phylogenetic tree generation application, several normalisation operations performed on each of the material sequence data. Such steps will be discussed in detail in the 4th chapter.

4 Pipeline

4.1 Raw data(Input):

The raw data is coming from the reflectance data collected from the Hyper spectroscopy cameras’. Raw data is in the form of a ’.csv’ file with incident light wavelengths and reflectance values of 16 different sample materials. Comprising of 4 fabric materials and each material with 4 different colors (Figure 4.1 and Figure 4.2). Figure 4.3 is a plot of raw data shown in Figure 4.2.

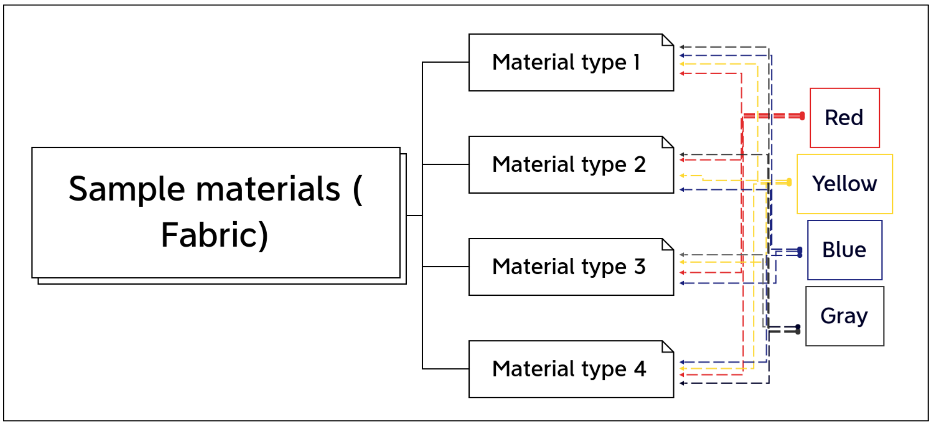
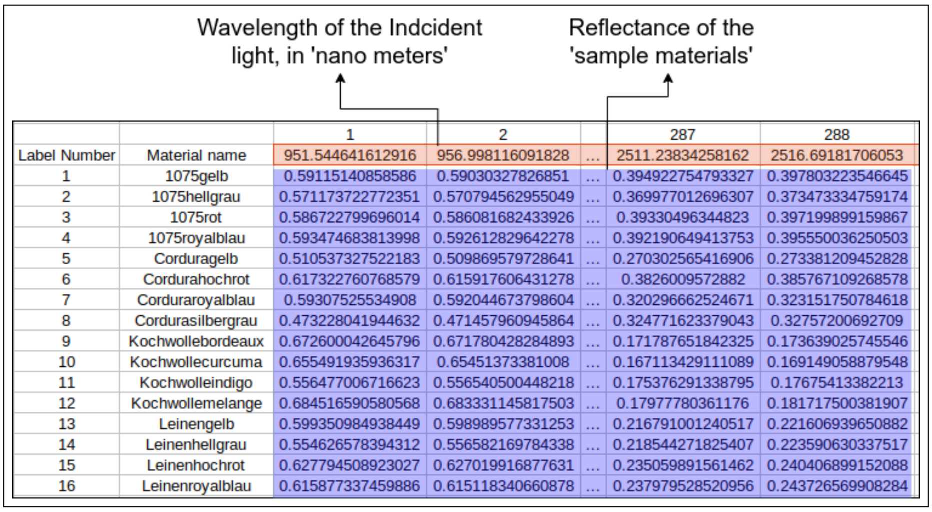


Figure 4.1: Sample materials



Wavelength of the Indcident light, in 'nano meters'

Reflectance of the 'sample materials'

		1	2		287	288
Label Number	Material name	951.544641612916	956.998116091828	...	2511.23834258162	2516.69181706053
1	1075gelb	0.59115140858586	0.59030327826851	...	0.394922754793327	0.397803223546645
2	1075hellgrau	0.571173722772351	0.570794562955049	...	0.369977012696307	0.373473334759174
3	1075rot	0.586722799696014	0.586081682433926	...	0.39330496344823	0.397199899159867
4	1075royalblau	0.593474683813998	0.592612829642278	...	0.392190649413753	0.395550036250503
5	Corduragelb	0.510537327522183	0.509869579728641	...	0.270302565416906	0.273381209452828
6	Cordurahochrot	0.617322760768579	0.615917606431278	...	0.3826009572882	0.385767109268578
7	Corduraroyalblau	0.59307525534908	0.592044673798604	...	0.320296662524671	0.323151750784618
8	Cordurasilbergrau	0.473228041944632	0.471457960945864	...	0.324771623379043	0.32757200692709
9	Kochwollebordeaux	0.672600042645796	0.671780428284893	...	0.171787651842325	0.173639025745546
10	Kochwollecurcuma	0.655491935936317	0.65451373381008	...	0.167113429111089	0.169149058879548
11	Kochwolleindigo	0.556477006716623	0.556540500448218	...	0.175376291338795	0.17675413382213
12	Kochwollemelange	0.684516590580568	0.683331145817503	...	0.17977780361176	0.181717500381907
13	Leinengelb	0.599350984938449	0.598989577331253	...	0.216791001240517	0.221606939650882
14	Leinenhellgrau	0.554626578394312	0.556582169784338	...	0.218544271825407	0.223590630337517
15	Leinenhochrot	0.627794508923027	0.627019916877631	...	0.235059891561462	0.240406899152088
16	Leinenroyalblau	0.615877337459886	0.615118340660878	...	0.237979528520956	0.243726569908284

Figure 4.2: Raw reflectance data looks like above

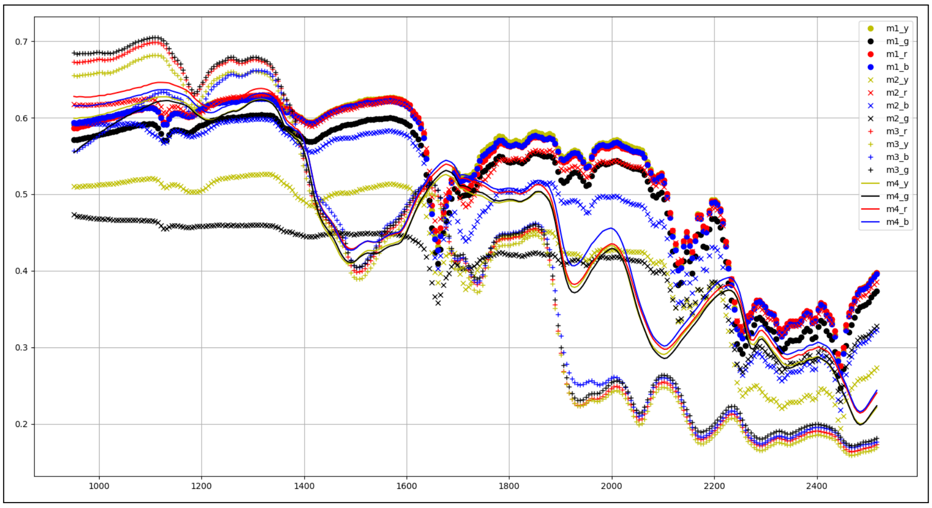


Figure 4.3: X-axis: Wavelength of incident light(900nm to 2500nm). Y-axis: Reflectance values (0-1)

4.2 Normalisation 1 - 'Independent material' wise:

Each material's reflectance sequence is min-max scaled between 0 and 1. This step is performed independent of other material's reflectance sequences. Figures 4.4, 4.5 and 4.6 are Comparison between unnormalised sequence, 'sample wise' independently normalised sequence and Globally normalised sequences, respectively. Global normalisation (normalisation with respect to 'maximum' and 'minimum' across the reflectance values of all 16 samples) did not yield significantly different results when compared to an unnormalised sequence, because, the maximum amongst all the materials is very close to 1 and the minimum amongst all the materials is very close to 0. That is the reason why Figures 4.4 and 4.6 appear very similar.

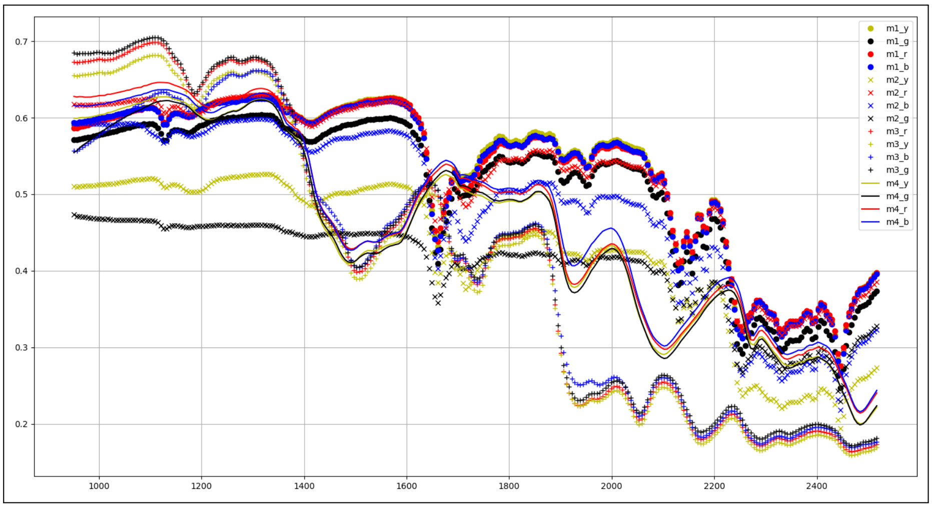


Figure 4.4: Unnormalised data. X-axis: Wavelength of incident light(900nm to 2500nm). Y-axis: Reflectance values (0-1)

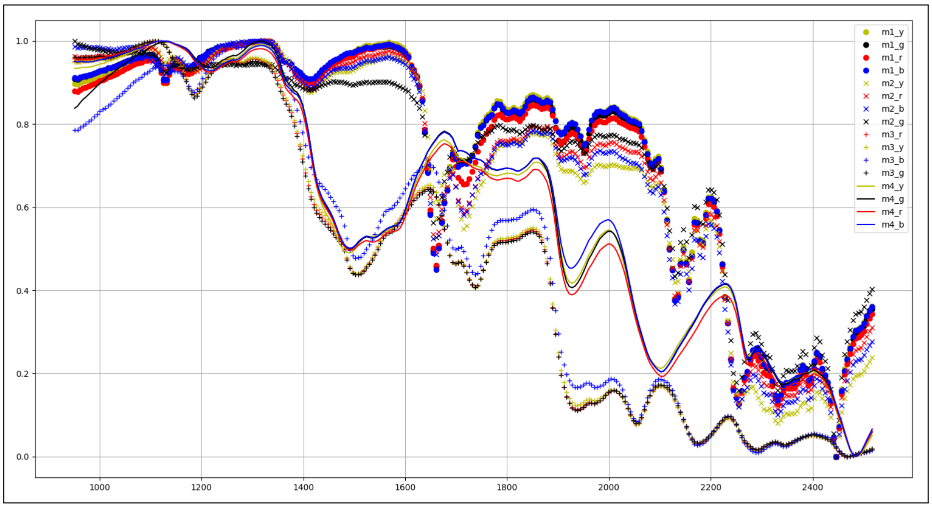


Figure 4.5: Material wise independently normalised data. X-axis: Wavelength of incident light(900nm to 2500nm). Y-axis: Reflectance values (0-1)

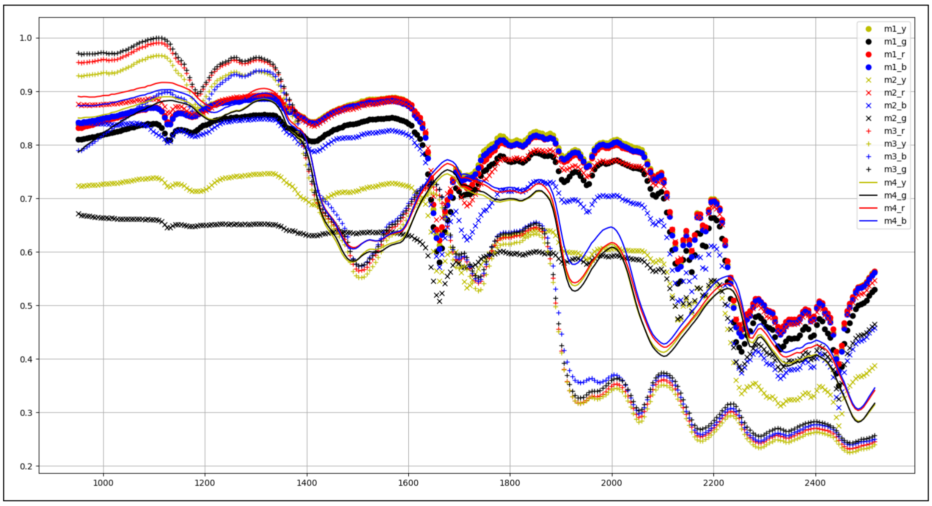
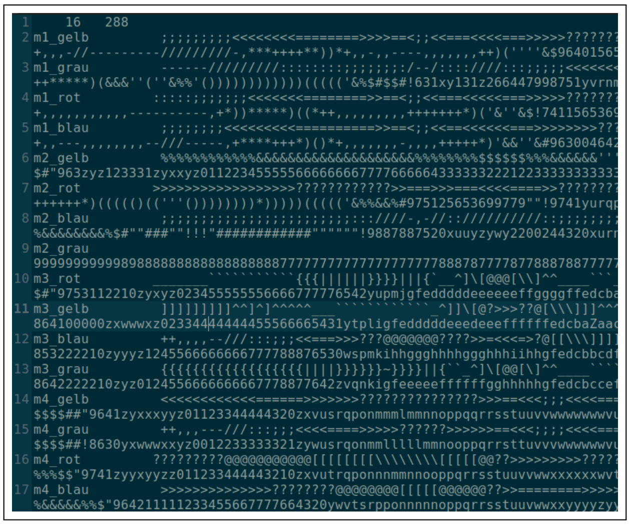


Figure 4.6: Global normalised data. X-axis: Wavelength of incident light(900nm to 2500nm). Y-axis: Reflectance values (0-1)

4.3 Numerical sequence to character sequences:

Phylogenetic algorithm expects the data to be in the format of characters. So, the reflectance values that range from 0 to 1 are mapped to ASCII characters that range from "!" to "~" as shown here:
!"#$%&'()*+,-./0123456789:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ
[\]^_‘abcdefghijklmnopqrstuvwxyz {|}~



```
    16   288
m1_gelb         ;;;;;;;;;<<<<<<<<<========>>>>>==<;;<<===<<<<===>>>>>??????
+,,,-//---------/////////-,***++++**))*+,,-,,----,,,,,,,++)(''''&$96401565
m1_grau         ------/////////::::::::;;;;;;;;:/--/::::////::::;;;;;<<<<<<<
++*****)(&&&''(''&%%'())))))))))))(((((('&%$#$$#!631xy131z266447998751yvrnm
m1_rot         ::::;;;;;;;;<<<<<<<=========>>==<;;<<===<<<<<===>>>>>??????
+,,,,,,,,,,,----------,+*))*****)((*++,,,,,,,,,+++++++*)('&''&$!7411565369
m1_blau         ;;;;;;;;<<<<<<<<<===========>>==<;;<<==<<<<<<===>>>>>>>>???
+,,---,,,,,,,,--///-----,+****+++*)()*+,,,,,,,-,,,,+++++*)'&&''&#963004642
m2_gelb         %%%%%%%%%%%%&&&&&&&&&&&&&&&&&&&&&&&%%%%%%%%%$$$$$$%%%&&&&&&''
$#"963zyz123331zyxxyz0112234555556666666677776666643333332221223333333333
m2_rot         >>>>>>>>>>>>>>>>>>>>????????????>>====>>====<<<<====>>???????
++++++*)((((()(('''())))))))*)))))(((((('&%%&&%#975125653699779""!9741yurqp
m2_blau         ;;;;;;;;;;;;;;;;;;;;;;;;;;;:::////-,-//::///////////:;;;;;;;;;
%&&&&&&&&&%$#""###""!!!"############""""""!9887887520xuuyzywy2200244320xurr
m2_grau
99999999999898888888888888888888888777777777777777777777888787777877888788877777
m3_rot         ________````````````{{{|||||||}}}}|||{`__^]\[@@@[\\]^^_____```_
$#"9753112210zyxyz02345555555566667777776542yupmjgfeddddeeeeeeffggggffedcba
m3_gelb         ]]]]]]]]]^^]^]^^^^^___`````````````_^]]\[@?>>>??@[\\\]]]^^^
864100000zxwwwxz02334444444455566666 5431ytpligfeddddeeedeeeffffffedcbaZaac
m3_blau         ++,,,,,--///:::;;;<<===>>>???@@@@@@@????>>=<<<=>?@[[\\\]]]]
853222210zyyyz12455666666667777888876530wspmkihhggghhhhggghhhiihhgfedcbbcdf
m3_grau         {{{{{{{{{{{{{{{{{{{{{||||}}}}}}}~}}}}||{``_^]\[@@[\]^^_____````
8642222210zyz012455666666666777887 7642zvqnkigfeeeeeffffffgghhhhhgfedcbccef
m4_gelb         <<<<<<<<<<<<<======>>>>>>>???????????????>>>==<<<;;;<<<<===
$$$$##"9641zyxxxyyz011233444444320zxvusrqponmmmlmmnnoppqqrrsstuuvvwwwwwwwvu
m4_grau         ++,,,,---///::::;;;<<<<====>>>>>??????>>>>>>==<<<;;;;<<<<===
$$$$##!8630yxwwwxxyz00122333333321zywusrqonmmllllllmmnooppqrrsttuvvvwwwwwwwvu
m4_rot         ?????????@@@@@@@@@@@[[[[[[[[\\\\\\\\\[[[[[@@??>>>>>>>>>>?????
%%%$$"9741zyyxyyzz011233444443210zxvutrqponnnmmnnooppqrrsstuuvvwwxxxxxxwvt
m4_blau         >>>>>>>>>>>>>>>???????@@@@@@@@[[[[@@@@@@??>>=========>>>>>
%&&&&&%%$"9642111112334556677777664320ywvtsrpponnnnnnoppqrrsstuuvwwxxyyyyzyy
```

Figure 4.7: Reflectance data mapped to 93 ascii characters

Figure 4.7 shows what it looks like when the numerical reflectance data is converted to ASCII characters.

4.4 Variable binning

The idea is to control the number of ASCII characters used to map the reflectance values based on the similarity between the material sequences. The input original reflectance sequence of 16 materials are divided into, say, 'k' sections. in each section pairwise DTW distance of each material with respect to every other material is calculated and summed. This sum corresponds to the "Overall Distance" of that 'section'.

- In a given section:
    - If the reflectance sequences are dramatically different to each other:
        - That means the "Overall Distance" is big and the number of characters that encode the numerical sequence should be small in number.
            - Why? Because, since the sequences are pretty dissimilar with respect to each other, not many characters are needed to differentiate.
    - else:
        - "Overall Distance" is a small number if reflectance sequences are indistinguishable each other. Implies, the number of characters that encode the numerical sequence should be a large number.
            - Why? because, since the sequences are pretty similar with respect to each other, encoding with more number of characters means capturing more detail.

4.5 Section cuts

The wavelength of incident light from the Hyper spectroscopy camera is in the range of ~900nm to ~2500nm. Reflectance behaviour response of the 16 sample materials (Figure 4.4) within this wavelength range can be subdivided in many ways as shown in Figures 4.8, 4.9, 4.10, 4.11 show some of the ways the total wavelength range can be sub divided into.

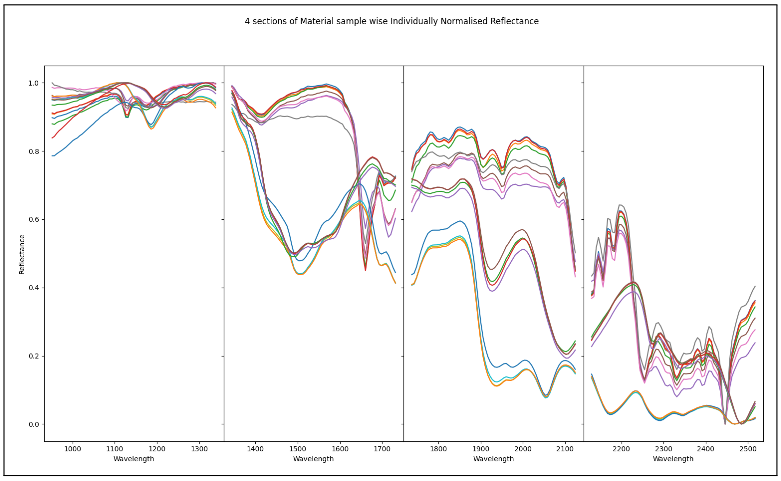


Figure 4.8: 4 sections of Material sample wise Normalised Reflectance

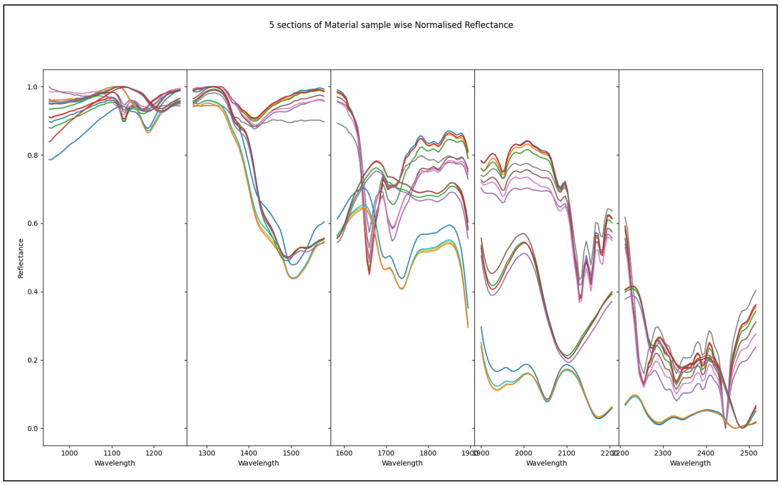


Figure 4.9: 5 sections of Material sample wise Normalised Reflectance

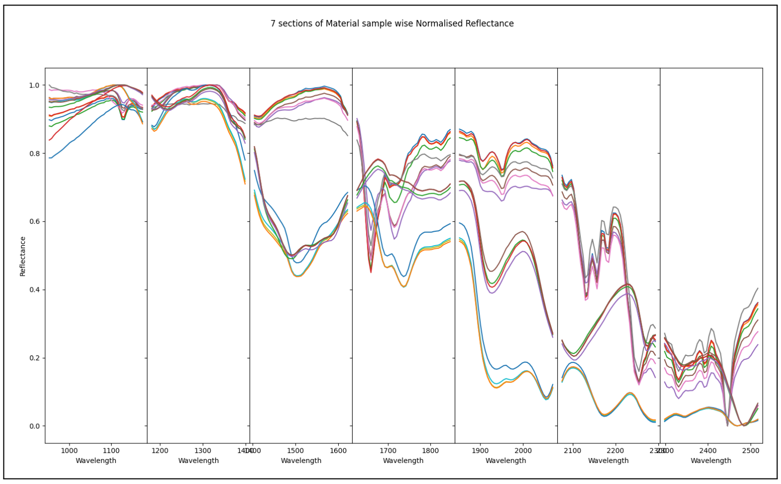


Figure 4.10: 7 sections of Material sample wise Normalised Reflectance

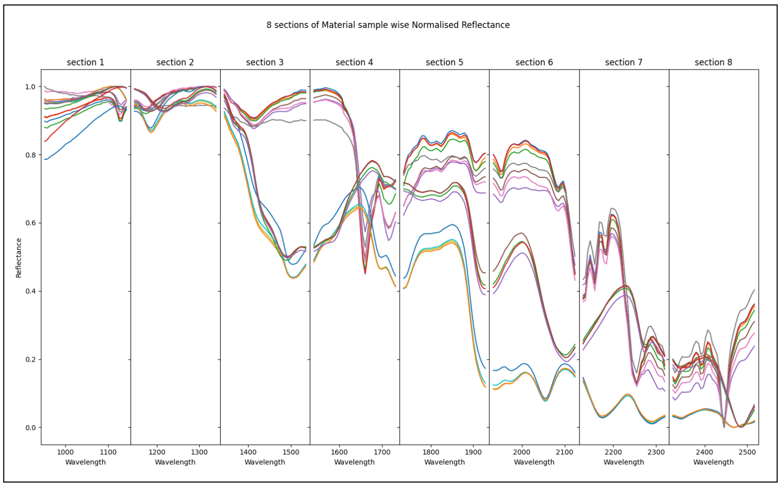


Figure 4.11: 8 sections of Material sample wise Normalised Reflectance

4.5.1 Normalisation 2 - Section wise

Max-min scaling/normalisation can be applied to the numerical reflectance sequence in each section. This ensures complete utilisation of all the characters during encoding of numerical sequence into character sequence. Figure 4.12 and Figure 4.13 illustrates section wise normalisation performed on a reflectance data that is already sample wise normalised. One can always perform section wise normalisation on an unnormalised or globally normalised data (Figure 4.4, 4.6). The additional character axis on the right of the plots in Figures 4.14 and 4.13 visualise the number of characters that will be utilised by each section. for example, in Figure 4.12 the first and the second sections from the left will be utilising only ~20% of the characters.

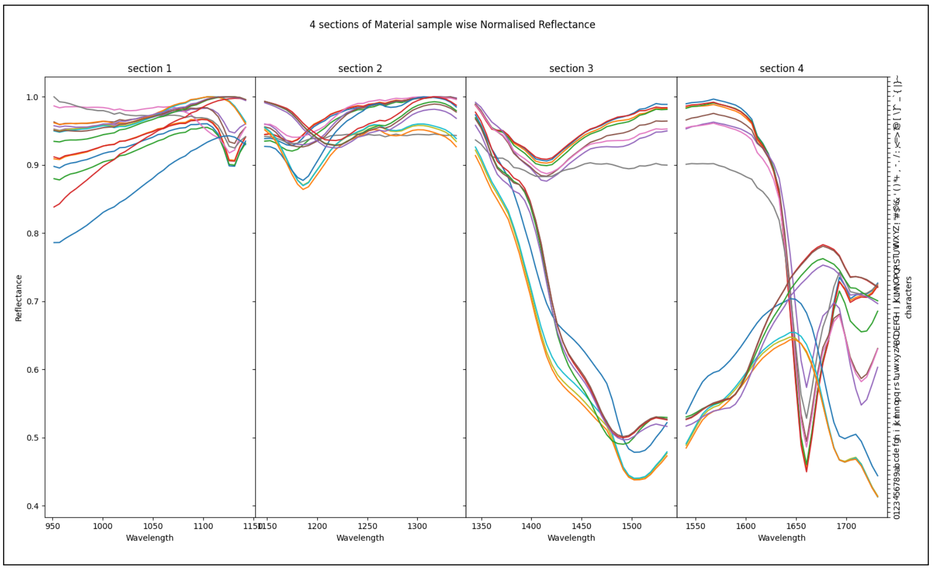


Figure 4.12: 4 sections of Material sample wise Normalised Reflectance data, along with 'Character'-axis

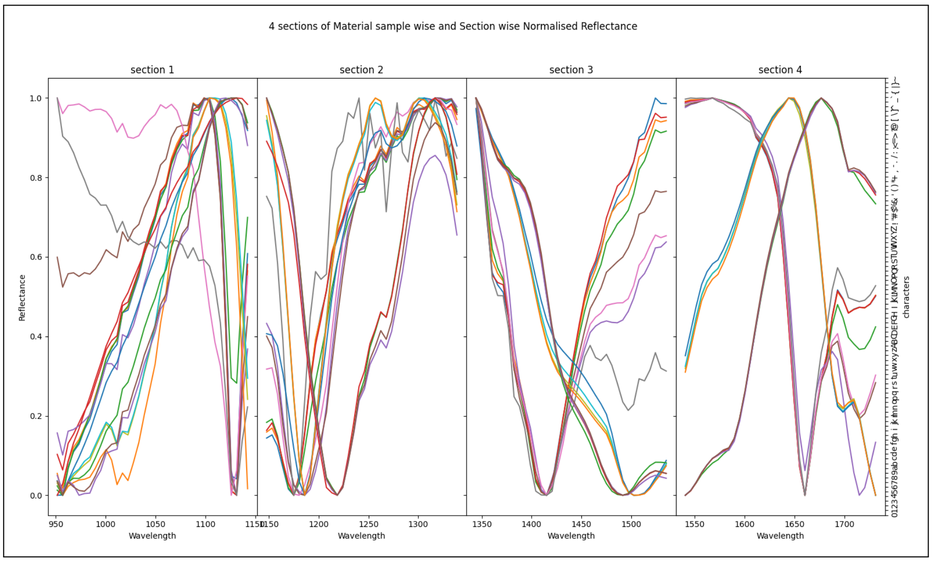


Figure 4.13: 4 sections of Material sample wise AND Section wise Normalised Reflectance data along with 'Character'-axis

4.6 Character sequence to ’.phy’ file conversion:

A ”.phy" file is an acceptable input for the phylogenetic Maximum parsimony algorithm. One ".phy" file must be generated for every section. Figure 4.7 is one of such ".phy" files.

4.7 Character sequence ’.phy’ to Maximum Parsimonious tree

Each .phy file is now the input to a tree building algorithm pipeline. Figure 4.14 shows a simplistic pipeline of building a Maximum parsimonious tree.

4.7.1 Tree building pipeline:

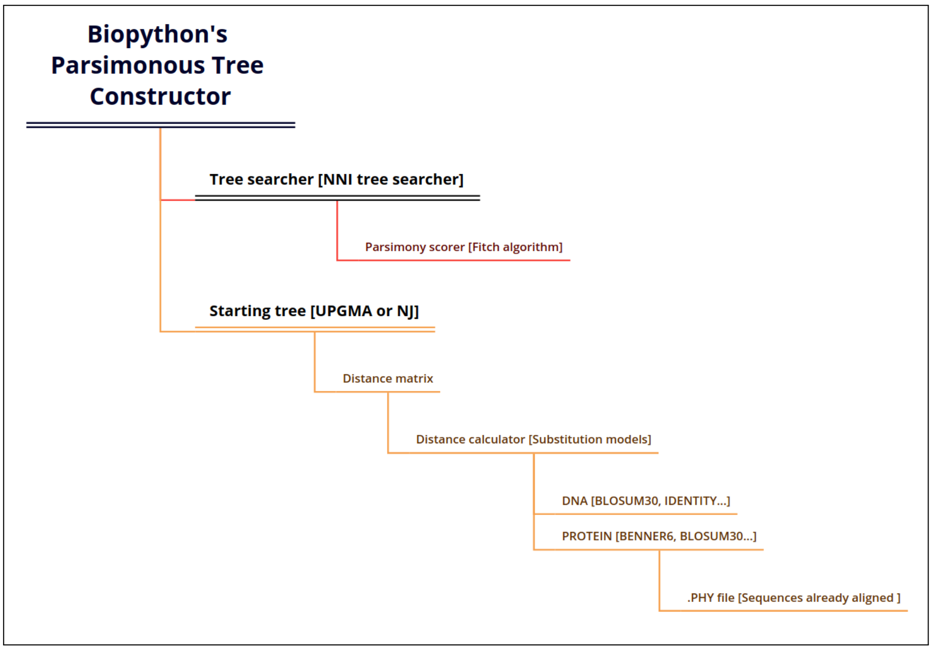

Biopython's Parsimonous Tree Constructor
Tree searcher [NNI tree searcher]
Parsimony scorer [Fitch algorithm]
Starting tree [UPGMA or NJ]
Distance matrix
Distance calculator [Substitution models]
DNA [BLOSUM30, IDENTITY...]
PROTEIN [BENNER6, BLOSUM30...]
.PHY file [Sequences already aligned]

Figure 4.14: Steps involved in building a ’Maximum Parsomony’ tree.

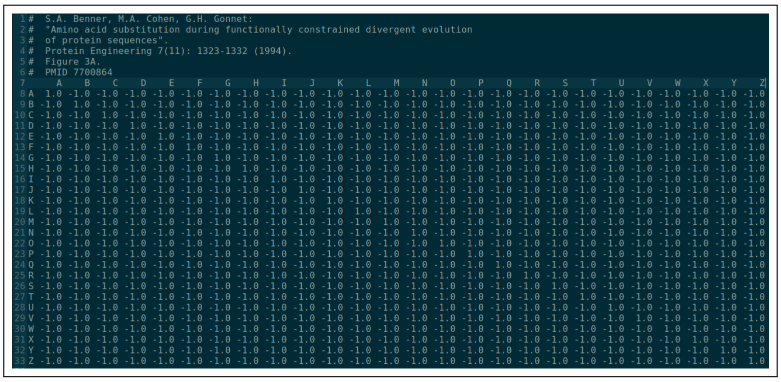
4.7.2 Input

’.phy’ file with character sequences of 16 sample materials (Figure 4.7).

4.7.3 Processing

Substitution Matrix

Benner93 - which is a substitution matrix that is obtained by modifying the original Benner6 substitution matrix. It contains 1's in the diagonal elements and -1's as every other element. i.e, any change is sequence is equally unappreciated. Figure 15 is the substitution matrix that is modified to contain only 1s and -1s.

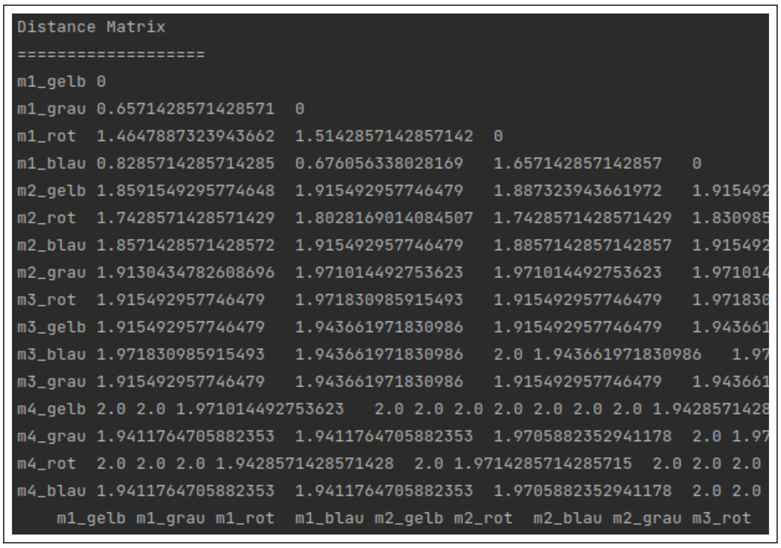


```
#  S.A. Benner, M.A. Cohen, G.H. Gonnet:
#  "Amino acid substitution during functionally constrained divergent evolution
#  of protein sequences".
#  Protein Engineering 7(11): 1323-1332 (1994).
#  Figure 3A.
#  PMID 7700864
     A    B    C    D    E    F    G    H    I    J    K    L    M    N    O    P    Q    R    S    T    U    V    W    X    Y    Z
A  1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0
B -1.0  1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0
C -1.0 -1.0  1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0
D -1.0 -1.0 -1.0  1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0
E -1.0 -1.0 -1.0 -1.0  1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0
F -1.0 -1.0 -1.0 -1.0 -1.0  1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0
G -1.0 -1.0 -1.0 -1.0 -1.0 -1.0  1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0
H -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0  1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0
I -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0  1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0
J -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0  1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0
K -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0  1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0
L -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0  1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0
M -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0  1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0
N -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0  1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0
O -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0  1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0
P -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0  1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0
Q -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0  1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0
R -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0  1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0
S -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0  1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0
T -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0  1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0
U -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0  1.0 -1.0 -1.0 -1.0 -1.0 -1.0
V -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0  1.0 -1.0 -1.0 -1.0 -1.0
W -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0  1.0 -1.0 -1.0 -1.0
X -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0  1.0 -1.0 -1.0
Y -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0  1.0 -1.0
Z -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0  1.0
```

Figure 4.15: Modified Substitution matrix

Distance Matrix

Biopython generates a distance matrix with the help of the defined substitution matrix and the input ".phy" file.



```
Distance Matrix
===================
m1_gelb 0
m1_grau 0.6571428571428571  0
m1_rot  1.4647887323943662  1.5142857142857142  0
m1_blau 0.8285714285714285  0.676056338028169   1.657142857142857   0
m2_gelb 1.8591549295774648  1.915492957746479   1.887323943661972   1.915492
m2_rot  1.7428571428571429  1.8028169014084507  1.7428571428571429  1.830985
m2_blau 1.8571428571428572  1.915492957746479   1.8857142857142857  1.915492
m2_grau 1.9130434782608696  1.971014492753623   1.971014492753623   1.971014
m3_rot  1.915492957746479   1.971830985915493   1.915492957746479   1.971830
m3_gelb 1.915492957746479   1.943661971830986   1.915492957746479   1.943661
m3_blau 1.971830985915493   1.943661971830986   2.0 1.943661971830986   1.97
m3_grau 1.915492957746479   1.943661971830986   1.915492957746479   1.943661
m4_gelb 2.0 2.0 1.971014492753623   2.0 2.0 2.0 2.0 2.0 2.0 2.0 1.9428571428
m4_grau 1.9411764705882353  1.9411764705882353  1.9705882352941178  2.0 1.97
m4_rot  2.0 2.0 2.0 1.9428571428571428  2.0 1.9714285714285715  2.0 2.0 2.0
m4_blau 1.9411764705882353  1.9411764705882353  1.9705882352941178  2.0 2.0
    m1_gelb m1_grau m1_rot  m1_blau m2_gelb m2_rot  m2_blau m2_grau m3_rot
```

Figure 4.16: Distance matrix

'Starting tree' Constructor

A start tree is constructed using the distance matrix and "Neighbour Joining" algorithm. there are alternatives such as UPGMA(Unweighted Pair Group Method with Arithmetic Mean) algorithm.

Tree searching algorithm

An algorithm called 'Nearest Neighbor Interchange' tree search algorithm is used to search for maximum parsimonious tree in the tree search space. It starts the search with the "Initiation tree/ understanding Starting tree". Tree searcher algorithms judge a given tree in the tree search space using parsimony scoring algorithms like "Fitch algorithm" and "Sankoff algorithm".

4.7.4 Resultant tree

A maximum parsimonious rooted tree is obtained.

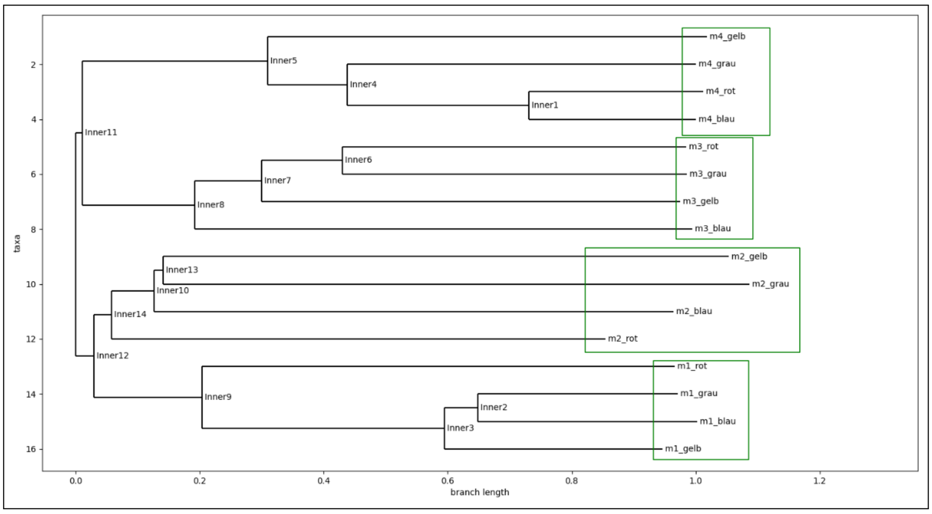


Figure 4.17: Rooted tree generated from 4th section of 4 section cut seen in Figure 4.13

4.7.5 Un-rooted Tree Representation

The rooted trees are rearranged and represented in the following form for better visualization of the clustering.

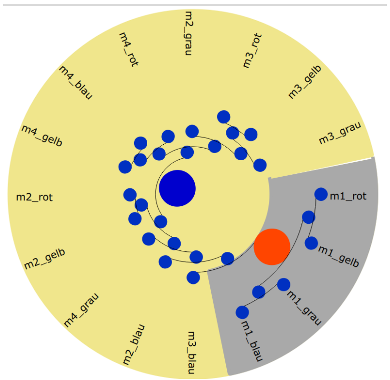


Figure 4.18: 1st of 4 Sections seen in Figure 4.13 have generated clustered trees as visualised here.

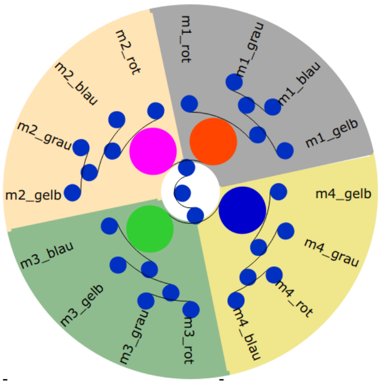


Figure 4.19: 4th of 4 Sections seen in Figure 4.13 have generated clustered trees as visualised here.

In the figures 4.18 and 4.19, the big nodes represent cluster heads, i.e, the cluster heads have 1 or more leaf nodes. In the next step called "Cluster check" the trees can be judged objectively AGHABOZORGI et al. (2015).

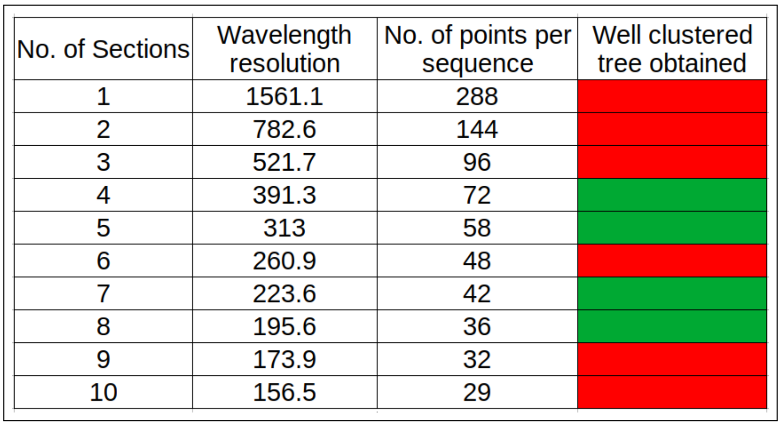
4.8 Output: Cluster Check

A systematic check is performed on the resultant maximum parsimonious tree where the system we have built can confirm whether the tree has successfully clustered the tree or not.
In this test, If at least one cluster head consists of leaf nodes that belong to more than 1 material type, The whole tree is labeled as "Bad Tree". If all the conditions are satisfied, It gives a "Good tree"

5
Analysis and interpretation of the results

Short-wave infrared (SWIR) and visible to near infrared (VNIR) hyper-spectroscopy camera data collection is discussed in Chapter 3. The pipeline discussed in Chapter 4 is applied on this reflectance behaviour data of the 16 material samples (Figure 4.1). We know apriori that, of the 16 samples, there are 4 types of materials and each material type has 4 samples. The aim as discussed in Chapter 2, is to build an algorithm/pipeline that will be able to correctly categorize/cluster the samples with their respective material types. Many approaches were explored in trial and error fashion to build a pipeline that can build a tree which can cluster each material types separately as displayed in Figure 4.17.

The pipeline discussed in Chapter 4 is one of such trial that yielded a tree that can cluster each of the 4 materials types well. In Chapter 4.5 it has been mentioned that the whole of SWIR reflectance behaviour data is sub divided into several equal sized sections. In previous approached a phylogenetic tree is built with the whole sequence of the data ranging from ~900nm to 1262500nm with out any kind of section cuts. In the current approach the reflectance sequence is sub-divided into several number of equally sized sections.



No. of Sections	Wavelength resolution	No. of points per sequence	Well clustered tree obtained
1	1561.1	288	🟥
2	782.6	144	🟥
3	521.7	96	🟥
4	391.3	72	🟩
5	313	58	🟩
6	260.9	48	🟥
7	223.6	42	🟩
8	195.6	36	🟩
9	173.9	32	🟥
10	156.5	29	🟥

Figure 5.1: Varying no. of sections and corresponding results

Figure 5.1 shows the number of ways in which SWIR data is sliced into. SWIR data contained of 288 reflectance behaviour points per material sample. The 288 reflectance behaviours were in response to incident light of wavelengths ranging from 900nm to 2500nm.

As mentioned previously and also in the Figure 5.1, the sequence with no sections cuts (No. of sections = 1), where the no. of points per sequence were 288, did not yield in a well clustered tree. Figure 5.3 shows only the detailed version of the first 5 section cuts. In Figure 5.3, the Incident light wavelength range that yielded a "well clustered tree" was in the range of 1900nm to 2200 nm Further section cuts were made with 2, 3, 4, 5, 6, 7, 8, 9, 10 sections. Out of the 55 (10+9+8...2+1) trees that were build from the various sections of varying sizes, only 4 sections have yielded in a 'well clustered tree'. All the 4 sections happen to be in a similar range of light wavelength (Figure 5.2) of ~2000 nanometers to ~2200 nanometers.

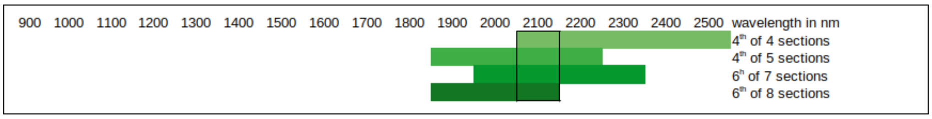
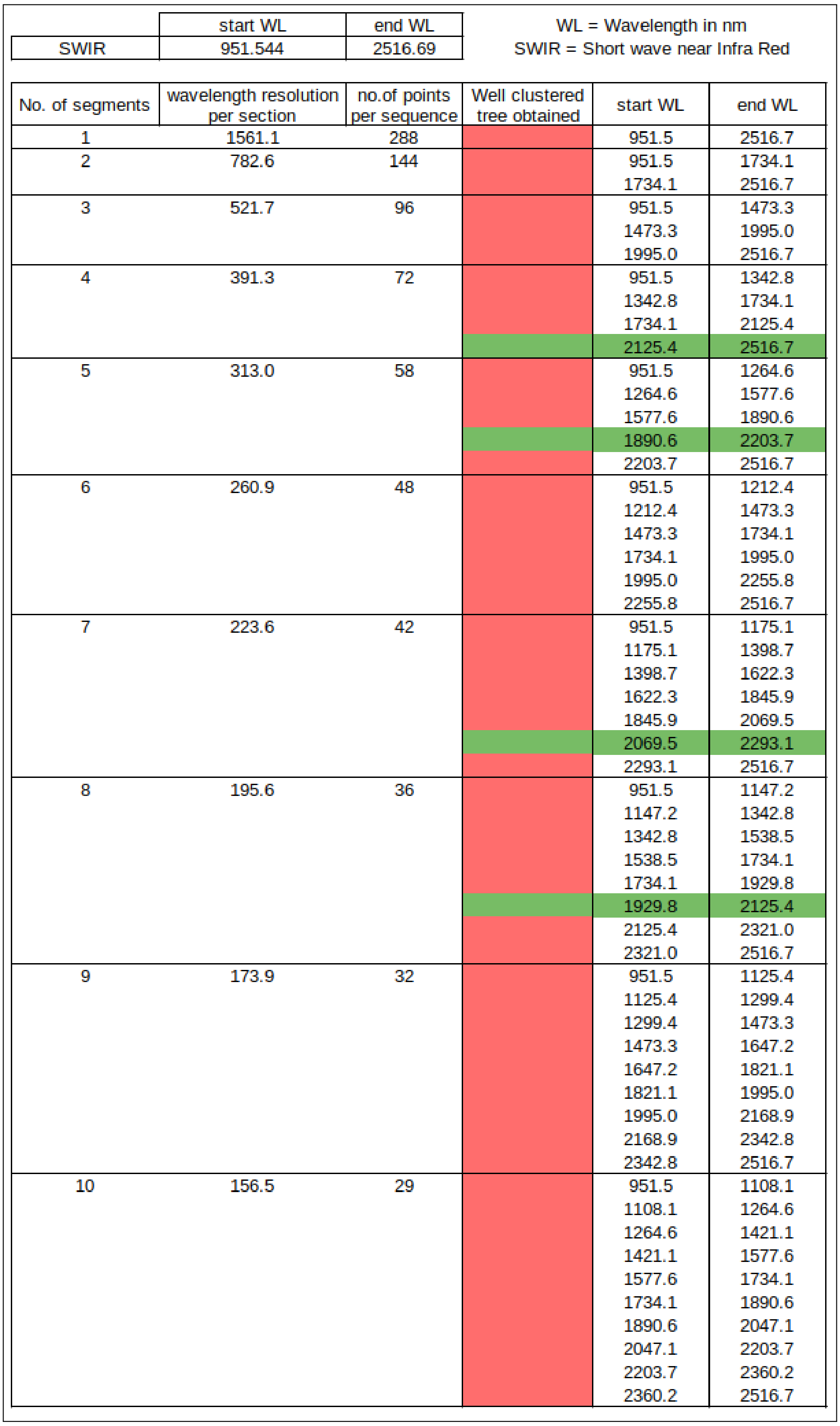


Figure 5.2: Common wavelength range of sections which yielded a 'well clustered tree'



	start WL	end WL
SWIR	951.544	2516.69

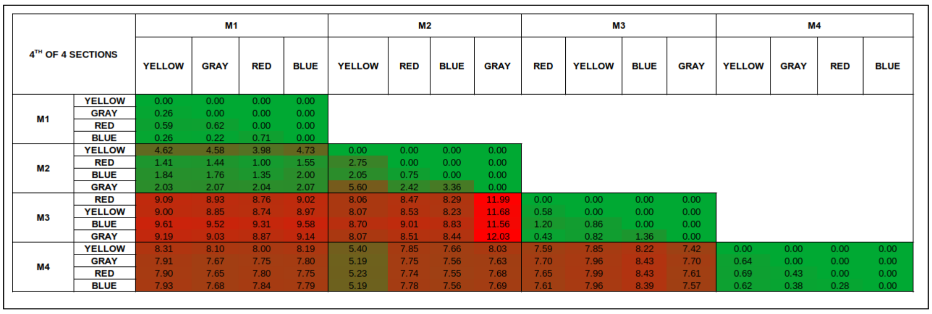
WL = Wavelength in nm
SWIR = Short wave near Infra Red

No. of segments	wavelength resolution per section	no.of points per sequence	Well clustered tree obtained	start WL	end WL
1	1561.1	288	🟥	951.5	2516.7
2	782.6	144	🟥	951.5	1734.1
			🟥	1734.1	2516.7
3	521.7	96	🟥	951.5	1473.3
			🟥	1473.3	1995.0
			🟥	1995.0	2516.7
4	391.3	72	🟥	951.5	1342.8
			🟥	1342.8	1734.1
			🟥	1734.1	2125.4
			🟩	2125.4	2516.7
5	313.0	58	🟥	951.5	1264.6
			🟥	1264.6	1577.6
			🟥	1577.6	1890.6
			🟩	1890.6	2203.7
			🟥	2203.7	2516.7
6	260.9	48	🟥	951.5	1212.4
			🟥	1212.4	1473.3
			🟥	1473.3	1734.1
			🟥	1734.1	1995.0
			🟥	1995.0	2255.8
			🟥	2255.8	2516.7
7	223.6	42	🟥	951.5	1175.1
			🟥	1175.1	1398.7
			🟥	1398.7	1622.3
			🟥	1622.3	1845.9
			🟥	1845.9	2069.5
			🟩	2069.5	2293.1
			🟥	2293.1	2516.7
8	195.6	36	🟥	951.5	1147.2
			🟥	1147.2	1342.8
			🟥	1342.8	1538.5
			🟥	1538.5	1734.1
			🟥	1734.1	1929.8
			🟩	1929.8	2125.4
			🟥	2125.4	2321.0
			🟥	2321.0	2516.7
9	173.9	32	🟥	951.5	1125.4
			🟥	1125.4	1299.4
			🟥	1299.4	1473.3
			🟥	1473.3	1647.2
			🟥	1647.2	1821.1
			🟥	1821.1	1995.0
			🟥	1995.0	2168.9
			🟥	2168.9	2342.8
			🟥	2342.8	2516.7
10	156.5	29	🟥	951.5	1108.1
			🟥	1108.1	1264.6
			🟥	1264.6	1421.1
			🟥	1421.1	1577.6
			🟥	1577.6	1734.1
			🟥	1734.1	1890.6
			🟥	1890.6	2047.1
			🟥	2047.1	2203.7
			🟥	2203.7	2360.2
			🟥	2360.2	2516.7

Figure 5.3: Detailed view of results of the section cuts for the first 10 section sizes

5.1 DTW pairwise distance matrices

In order to find what is special about the only wavelength range that most commonly yielded in a well clustered tree it is attempted to build pairwise distance matrices for each section. As discussed in Chapter 2.7 and Chapter 4.7.3, the role of distance matrix on the initial/starting tree is crucial. Initial tree is the necessary input for the tree search Nearest neighbour interchange(NNI) algorithm. Distance matrix in Biopython is built based on the substitution matrix, in our case, the substitution matrix is modified, as discussed in Chapter 4.7.3 and shown in Figure 4.15.
For the analysis of the results obtained after the section wise tree building, a new kind of distance matrix is calculated. This is called Dynamic Time Warping (DTW) algorithm, which is distance measuring algorithm that can compare between two sequences and assign a score for the amount of distance (inverse of similarity) between the two sequences.
A total of 55 (10+9...2+1) DTW distance matrices were generated for analysis. Figure 5.4 shows one of such distance matrices. The cells were color coded with respect to the DTW distance between the material samples. All the diagonal cell blocks happen to be coded 'Green' because the intra material distance(inverse of similarity) are small and are desirable. Meanwhile the inter-material sample distances (for example: between Material 1's samples and Material 3's samples) are in general big, hence they are coded in 'Red'. As one can see, such a color coded visualisation quickly helps in identifying the inter and intra material distances in a selected section.

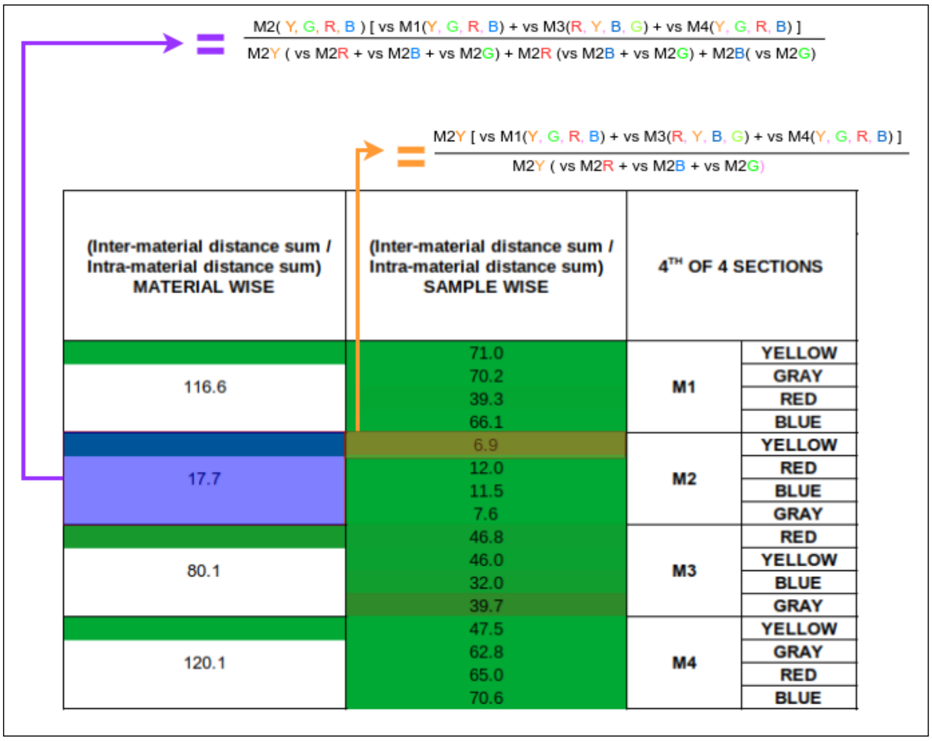


4TH OF 4 SECTIONS		M1				M2				M3				M4			
		YELLOW	GRAY	RED	BLUE	YELLOW	RED	BLUE	GRAY	RED	YELLOW	BLUE	GRAY	YELLOW	GRAY	RED	BLUE
M1	YELLOW	0.00	0.00	0.00	0.00												
	GRAY	0.26	0.00	0.00	0.00												
	RED	0.59	0.62	0.00	0.00												
	BLUE	0.26	0.22	0.71	0.00												
M2	YELLOW	4.62	4.58	3.98	4.73	0.00	0.00	0.00	0.00								
	RED	1.41	1.44	1.00	1.55	2.75	0.00	0.00	0.00								
	BLUE	1.84	1.76	1.35	2.00	2.05	0.75	0.00	0.00								
	GRAY	2.03	2.07	2.04	2.07	5.60	2.42	3.36	0.00								
M3	RED	9.09	8.93	8.76	9.02	8.06	8.47	8.29	11.99	0.00	0.00	0.00	0.00				
	YELLOW	9.00	8.85	8.74	8.97	8.07	8.53	8.23	11.68	0.58	0.00	0.00	0.00				
	BLUE	9.61	9.52	9.31	9.58	8.70	9.01	8.83	11.56	1.20	0.86	0.00	0.00				
	GRAY	9.19	9.03	8.87	9.14	8.07	8.51	8.44	12.03	0.43	0.82	1.36	0.00				
M4	YELLOW	8.31	8.10	8.00	8.19	5.40	7.85	7.66	8.03	7.59	7.85	8.22	7.42	0.00	0.00	0.00	0.00
	GRAY	7.91	7.67	7.75	7.80	5.19	7.75	7.56	7.63	7.70	7.96	8.43	7.70	0.64	0.00	0.00	0.00
	RED	7.90	7.65	7.80	7.75	5.23	7.74	7.55	7.68	7.65	7.99	8.43	7.61	0.69	0.43	0.00	0.00
	BLUE	7.93	7.68	7.84	7.79	5.19	7.78	7.56	7.69	7.61	7.96	8.39	7.57	0.62	0.38	0.28	0.00

Figure 5.4: DTW distance matrix of 4th of 4 sections

5.2 Further search for a pattern

For further detailed analysis, secondary distance metrics were extracted from the distance matrix. The two metrics are ratios between the sums of distances between the samples one is calculated material wise and the other is sample wise. This kind of metrics are extracted for each of the 55 distance matrices. Figure 5.5 displays one of such a distance matrix along with two kind of metrics and comparison between them.



$$= \frac{\text{M2(Y, G, R, B) [vs M1(Y, G, R, B) + vs M3(R, Y, B, G) + vs M4(Y, G, R, B)]}}{\text{M2Y (vs M2R + vs M2B + vs M2G) + M2R (vs M2B + vs M2G) + M2B(vs M2G)}}$$

$$= \frac{\text{M2Y [vs M1(Y, G, R, B) + vs M3(R, Y, B, G) + vs M4(Y, G, R, B)]}}{\text{M2Y (vs M2R + vs M2B + vs M2G)}}$$

(Inter-material distance sum / Intra-material distance sum) MATERIAL WISE	(Inter-material distance sum / Intra-material distance sum) SAMPLE WISE	4^{TH} OF 4 SECTIONS	
116.6	71.0	M1	YELLOW
	70.2		GRAY
	39.3		RED
	66.1		BLUE
17.7	6.9	M2	YELLOW
	12.0		RED
	11.5		BLUE
	7.6		GRAY
80.1	46.8	M3	RED
	46.0		YELLOW
	32.0		BLUE
	39.7		GRAY
120.1	47.5	M4	YELLOW
	62.8		GRAY
	65.0		RED
	70.6		BLUE

Figure 5.5: Material wise and Sample wise ratio of Inter-material and Intra-material distances

5.3 Material wise ratios

It is a ratio of "Sum of inter-Material distances" and "Sum of intra-Material distances".

5.3.1 Sum of inter-Material distances

This is a total sum of pairwise DTW distances between all samples of one material type versus all samples of rest of the materials.

5.3.2 Sum of intra-Material distances

This is the total sum of all pairwise DTW distances between samples of a given material. type.

5.4 Sample wise ratios

It is the ratio of "Sum of inter material sample distances" and " Sum of intra material sample distances"

5.4.1 Sum of inter material sample distances

This is a total sum of pairwise DTW distances between one sample of selected material type versus samples of all other material types.

5.4.2 Sum of intra material sample distances

This is a total sum of pairwise DTW distances between one sample of selected material type versus rest of the samples of the same material type.

5.5 Comparison of two metrics amongst all sections

Given the SWIR reflectance sequence is divided into k equal parts, a comparison study has been performed between each of the K equal parts. Figure 5.5 is a screen shot of 4th sections of 4 sections (i.e, k=4). Also, please refer to the document "segment_wise_tree_generation_data.xlxs"

5.6 Conclusion

The comparison of the two metrics that were mentioned earlier have did not display any kind of perceivable pattern that can predict the occurrence of a "well clustered tree". Hence there is still a need for systematic research in finding such a pattern.

6
Future scope

There are many ways to extend the research in this project. We will mention and briefly describe some of such potential extensions.

6.1 different types of Distance matrices and Substitution matrices

In the pipeline of generating a maximum parsimonious tree discussed in the chapter 4.7.3, building a distance matrix is one of the first steps involved in building a maximum parsimonious tree. In the current implementation the distance matrix, that is prerequisite for building a starting/initiation tree, is dependent on the substitution matrix.

6.1.1 Substitution matrix

One can modify the substitution matrix to have more variants of scores for representing rate of evolutionary change between two states. Different types of substitution matrices will directly impact the starting/initiation tree, because the starting/initiation tree is built based on the distance matrix that is obtained through the modified substitution matrix.

6.1.2 Distance matrix

Since the starting tree/initiation tree is a function of distance matrix, therefore, one can always use many other variants of distance algorithms that score pairwise distances between two sequences. One of such most extensively used distance algorithm is "Dynamic Time Warping" (DTW) algorithm. Lavenstein algorithm, Needleman Wunsch algorithm, Smith-

Waterman algorithm are some of the distance measuring algorithms that are suitable for this purpose.

6.2 A quantitative scoring/evaluation metric for a tree.

In the current implementation there is an algorithm that can automatically check if a certain tree is a well clustered tree or not. In the future this functionality can be extended to return a score for each tree with regards to how close the tree is to an "Ideal" well clustered tree.

7 Conclusion

Using the Phylogenetic approach to find the relationship between taxa is beginning to be applied outside of biological applications in recent times, especially in astrophysics and financial clustering and prediction areas of research. The attempt to apply Phylogenetic methodology to define a unique identity of an object based on it's inherent features is a novel application and there is still a tremendous amount of research opportunity to explore.

In this research study, it has been proven that a methodology is capable of inferring relationships that are true to the real world data. The only input necessary for this methodology is the feature information of the objects of interest and some basic hyper parameters like the choice of substitution matrices, tree search algorithms, parsimony tree scoring algorithms etc.,

For this pipeline to be robust enough to be put in actual real world application, there is still a lot of systematic research and analysis in many areas of the pipeline needed. The current development and results are a great starting point for further research in this direction.

A
Abbreviations and Notations

Acronyms

Acronym	Meaning
DTW	Dynamic time warping
VNIR	visible and near-infrared
NCBI	National Center for Biotechnology Information
SWIR	Short-wave infrared

B
List of Figures

C

Bibliography

[Allaby and Woodwark 2004] R. G. Allaby and M. Woodwark. **Phylogenetics in the bioinformatics culture of understanding**. pp. 128–146, 2004.

[Id et al. 2019] M. B. Id, N. M. Id, U. Mai, X. Jia and S. Mirarab. **TreeCluster : Clustering biological sequences using phylogenetic trees**. pp. 1–20, 2019.

[Wallace 2011] M. S. Wallace. **Morphology-based phylogenetic analysis of the treehopper tribe Smiliini (Hemiptera: Membracidae: Smiliinae), with reinstatement of the tribe Telamonini**. Zootaxa, Vol. 42(3047):1–42, 2011.

[Fraix-Burnet et al. 2017] D. Fraix-Burnet, M. D'Onofrio and P. Marziani. **Phylogenetic Analyses of Quasars and Galaxies**. Frontiers in Astronomy and Space Sciences, Vol. 4(October):1–6, 2017.

[Fraix-Burnet 2017] D. Fraix-Burnet. **Phylogenetic Tools in Astrophysics**. Wiley StatsRef: Statistics Reference Online, pp. 1–6, 2017.

[Retzlaff and Stadler 2018] N. Retzlaff and P. F. Stadler. **Phylogenetics beyond biology**. Theory in Biosciences, Vol. 137(2):133–143, 2018.

[Chang 2004] M. L. Chang. **Phylogenetic Analysis of Morphological Data (review)**. Human Biology, Vol. 76(1):165–168, 2004.

[Hall 2013] B. G. Hall. **Building phylogenetic trees from molecular data with MEGA**. Molecular Biology and Evolution, Vol. 30(5):1229–1235, 2013.

[KUMAR] S. P. Kumar. **Softwares For Phylogentic Analysis Survey of Software Programs Available For Phylogenetic Analysis Three Major Reasons for Using Phylogenetics**. ????

[CHANG et al. 2021] J. Chang, B. Chapman, I. Friedberg, T. Hamelryck, M. D. Hoon, P. Cock, T. Antao, E. Talevich and B. Wilczy. **Biopython Tutorial and Cookbook**. Vol. 2021, 2021.

[YANG and RANNALA 2012] Z. Yang and B. Rannala. **Molecular phylogenetics: Principles and practice**. Nature Reviews Genetics, Vol. 13(5):303–314, 2012.

[AMIT ROY 2014] S. R. Amit Roy. **Molecular Markers in Phylogenetic Studies-A Review**. Journal of Phylogenetics Evolutionary Biology, Vol. 02(02), 2014.

[BENNET et al. 1994] S. A. Bennet, M. A. Cohen and G. H. Gonnet. **Amino acid substitution during functionally constrained divergent evolution of protein sequences**. Protein Engineering, Design and Selection, Vol. 7(11):1323–1332, 1994.

[AGHABOZORGI et al. 2015] S. Aghabozorgi, A. Seyed and T. Y. Wah. **Time-series clustering – A decade review**. Information Systems, 2015.