1.0 Chapter 1: Introduction

1.1 Sequence Comparison

All the existing and the extinct genomes are the outcome of the copying process that happens each generation from the emergence of the first living cell approximately 3.8 billion years ago. However, this process was accompanied by mutations and recombination. The genetic variation thus generated permitted adaptation to different habitats, which resulted in the diversity of present and extinct organisms. Thus, the evolution of organisms or sequences can be envisaged as a branching process where every pair of organisms or sequences has a common ancestor at a varying depth of an emerging tree.

Evolution is an ancestral process which is not observable directly. Thus, we reconstruct this historical process from the available sequences in different sequences. First step is to compare different sequences of different species and finding similar regions between them (Domazet-Lošo & Mirjana, 2010). The result of this sequence comparison would be similarity scores which will help us to construct historical branching patterns.

Information from comparison of sequences can be used in other cases beside historical branching patterns (also known as phylogenies). Regions with high similarity even relatively distantly-related organisms usually mean alike biological functions or form. When two sequences have statistically considerable similarity, they will also have considerable structural similarity but the converse is not correct; there are many cases of similar structures that do not have significant similarity (Pearson & Wood, 2001). Therefore, study of functional and structural organization, evolutionary mechanisms and

evolutionary history of organisms all rely upon sequence comparison. This is the reason of sequence comparison importance as an essential tool in modern biology (Domazet-Lošo & Mirjana, 2010).

Calculation of biological sequence comparison needs widespread tools. Scientists have been come up with many computational and statistical methods in order to compare biological sequences in last ten years. To study the similarity or dissimilarity of sequences, there are two different bioinformatics methodologies which are alignment-based and alignment-free methods.

The alignment-based methods can be used among two sequences (it is called pairwise alignment) or multiple sequences (which is called MSA). In pairwise alignment, a pair of sequences is aligned, and in the multiple sequence alignment, more than two sequences are aligned. After computing the alignment, it still does not straightforwardly represent evolutionary distance. Particularly, the evolutionary distance among nucleotide sequences is the number of nucleotide substitutions per site. Therefore, the similarity score that has been derived from an alignment between a pair of sequences should be turned into evolutionary distance. There are two directions that alignment-free methods have been developed in them: methods that are based on the analysis of word frequencies between sequences, and the others are methods rely on information theory (Vinga & Almeida, 2003). In both methods the output will demonstrate the distance among sequences. The distance measures obtained by alignment-free methods can be converted to phylogenetic trees.

Phylogenetic trees symbolise evolutionary relationships among batches of species or biological sequences. The comparison of the topologies of two or more phylogenetic trees is used, for example, to estimate if tree partitions support a bootstrapping analysis or to compare other phylogenetic hypotheses or reconstructing a single species tree from individual gene trees (Marcet-Houben & Gabaldon, 2011).

After retrieving DNA Mitochondria of six species from NCBI, there are two types of comparison. One, by making alignment score matrix in alignment-based methods and other is comparing distance matrix between sequences which are alignment-free methods. Once an alignment is computed, it still does not directly reflect evolutionary distance. In particular, the evolutionary distance between nucleotide sequences is the number of nucleotide substitutions per site. Thus, the similarity score between a pair of sequences derived from an alignment should be transformed into evolutionary distance. Evolutionary distances can be further used to construct phylogenies. A phylogenetic (or evolutionary) tree is usually a bifurcating tree whose leaves represent sequences or organisms. Each internal node (a bifurcation in the tree) corresponds to a common ancestor of two or more entities (organisms or sequences) at the leaves of the tree. There are several methods for the reconstruction of a phylogenetic tree based on evolutionary distances between all sequence pairs.

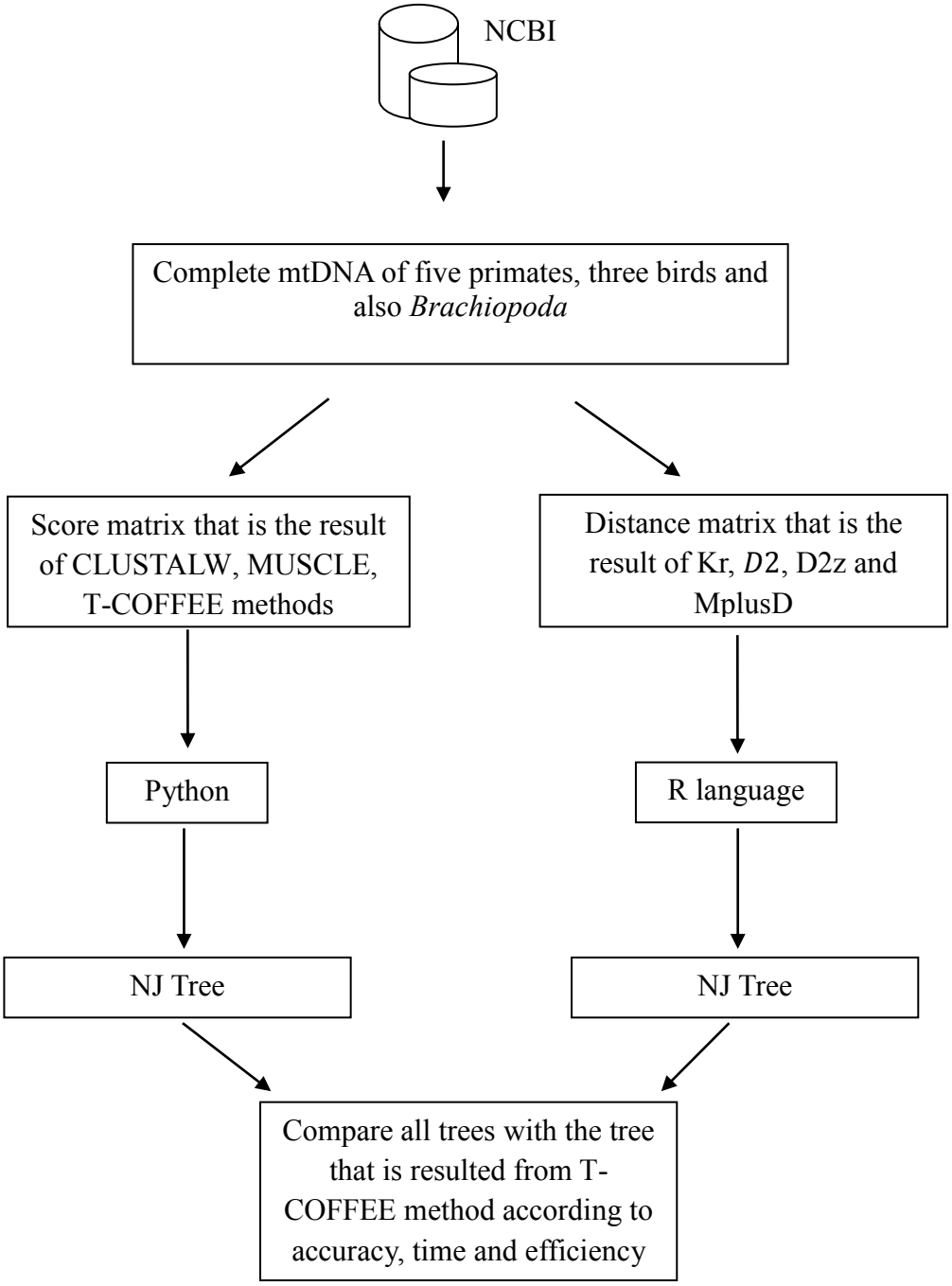

Figure 1.1. Data Flow

In this project the benchmark was the tree which produced by T-COFFEE because it is the most accurate method with the ability to incorporate heterogeneous types of information (Edgar, 2006).

1.2 Structure of Mitochondrial Genome

In this project I used the mitochondrial genome of 9 species, from two major animal groups(primates and birds) and a *Brachiopoda*. Comparing mitochondrial genome organizations enables to produce convincing phylogeny trees. Genome evolution can be traced by using mitochondrial systems (Boore, 1999). The mitochondrial is a tiny genome which is around 16 kbp in size. With some exceptions, Most of mitochondrial genomes contain 37 genes, 2 rRNA, 13 protein and 22 tRNA genes (Chinnery, Howell, Andrews, & Turnbull, 1999).

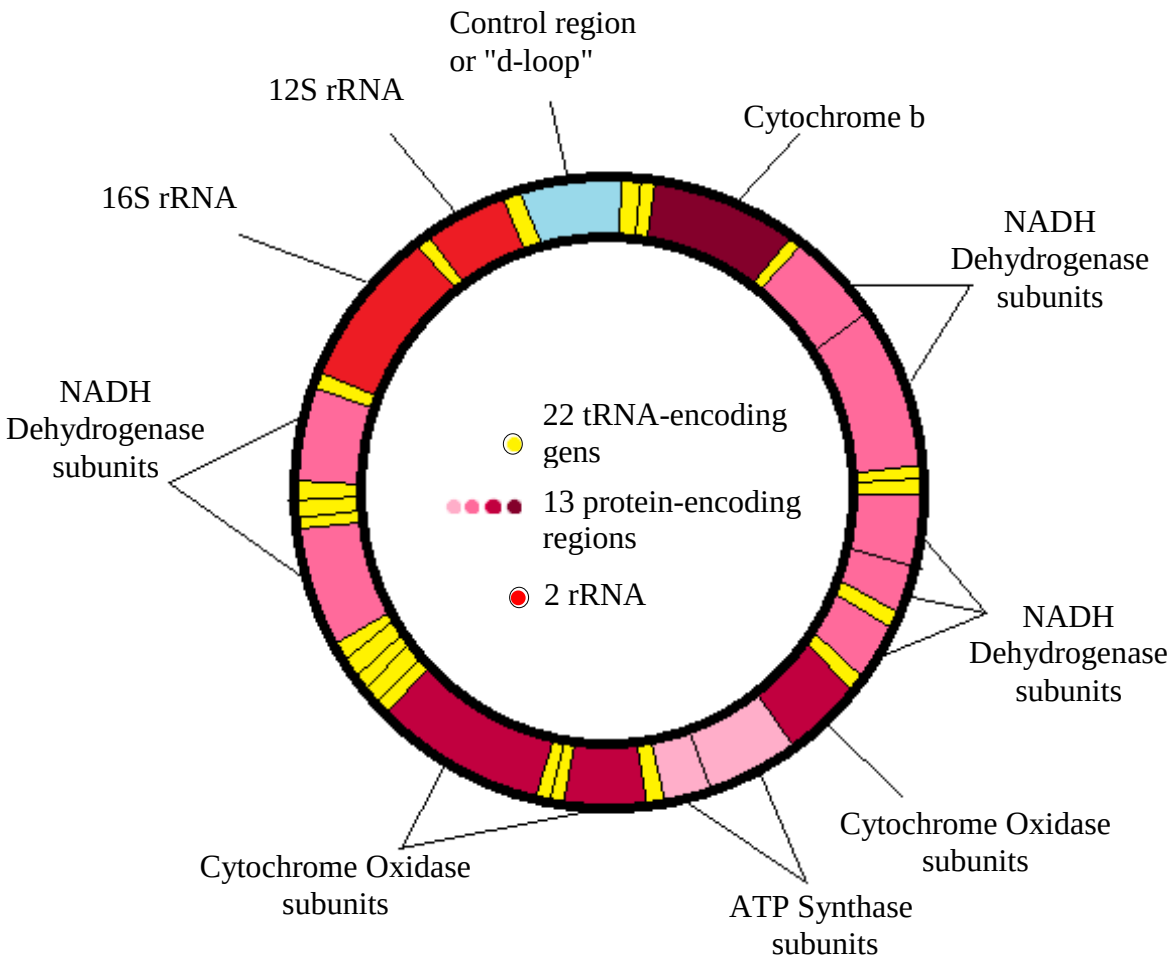

Figure 1.2. Human mitochondrial genome.

1.3 Research Objectives

Here, by comparison of several alignment-free methods with alignment-based methods, I aim to consider the speed of methods, finding out which of these two types of methods produce more accurate results. The other objective is to compare different alignment-based methods to figure out the level of efficiency through alignment-based methods, also comparing alignment-free methods which are D2, D2z, Kr and MplusD together to find out the most reliable one.

2.0 Chapter 2 Literature Review

2.1 Alignment-based methods

There have been lots of research and studies conducted on efficient methods for construction of multiple sequence alignment, since multiple sequence alignment is crucially important in computational biology. MSA has an important role model in two related areas of molecular biology which are discovering sub regions which are highly conserved through biological sequences and also collecting the evolutionary history of some species from their associated sequences. (Wang & Jiang, 1994).

According Zhang and Kahveci (2006) the query sequences that are being used as input in MSA tools are assumed to have an evolutionary relationship in such way that they share a lineage and are traced from a common ancestor. In order to find out about sequences` shared evolutionary origins, the results of MSA can be used in phylogenetic analysis. Two sequences can be aligned with maximum score that is obtainable by O (L^2) as time complexity and using dynamic programming. Here, length of the sequences is L. This was first proposed by Needleman and Wunsch (1970). On the other hand, this algorithm is able to be expanded to align N sequences, but needs O (L^N) time. In their study a different type of heuristic MSA algorithms have been developed. In which almost all of them are based on progressive application of pairwise alignment. By adding sequences one by one to existing alignment they create alignments of huge numbers of sequences.

Dynamic Programming DP has been diversely used in Multiple Sequence Alignment (MSA) problems. Meanwhile, Jiang & Su in 2008, concluded that in a situation of large number of sequences, multiple dimensional DP would suffers from

large storage and computational complexities. Hence, progressive pairwise DP has been used for MSA. Of all MSA methods, here, I am going to describe more about CLUSTALW, TCOFFEE and MUSCLE.

2.1.1 CLUSTALW

CLUSTALW (Thompson, Higgins, & Gibson, 1994) was proposed in 1994 and it became the best method of choice for most of the biologists. The reason was of course the efficient progress of CLUSTALW in alignment sensitivity together with speed in compare to other tools. Nowadays CLUSTALW has kept its place among biologists and still considered as very effective MSA program. But according to Edgar (2006) there has been no significant advancement have been made to the algorithm since 1994 and even several modern methods introduced which can achieve better performance in terms of speed, accuracy or both (Edgar, 2006).

As shown in Figure 2.3, three main stages consist in the basic CLUSTALW algorithm:

1. The pairs of all sequences are separately aligned to calculate a distance matrix, showing the divergence of each pair of sequences;
2. A tree is inferred using the distance matrix as input to NJ algorithm.
3. Base to the order of branching in the guide tree, sequences are progressively aligned.

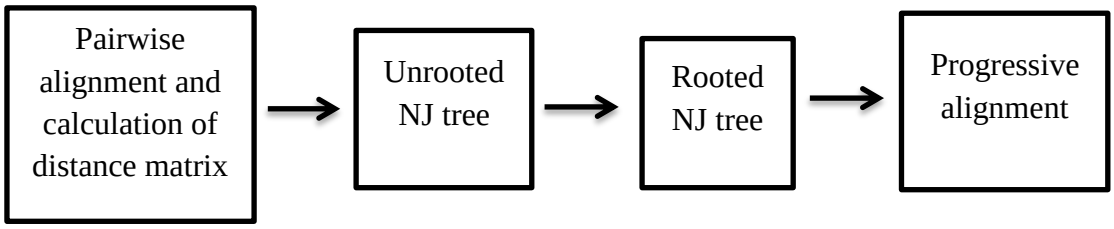

Figure 2.3. The basic procedure of CLUSTALW

Trelles(2001), classified CLUSTALW as a bioinformatics application which has semi-regular computational patterns (Trelles, 2001). On the other hand Li (2003) described that the algorithms are composed of both synchronous and asynchronous steps. In the pairwise alignment step, the pairwise distance can be measured by using fast approximate methods which enables a larger number of sequences to be aligned, on a microcomputer. The scores are calculated is in which that the number of k tuple minus a fixed penalty for each gap in the best alignment between two sequences. The calculation of scores is in the way that the number of identities in the most perfect alignment is divided by the number of compared residues (not including the position of gaps). Calculation of these two scores is by percentage of identity scores and later will change to distance matrixes by a simple transformation (division by 100 followed by subtraction from 1.0).

The distance matrix of first step will lead to trees which are used to finalize the multiple alignment process by using the NJ method. Later, un-rooted trees that has proportional branches length will calculate and find the divergence along each branch. Mid-point method places the root at a place that the centre of the branch lengths on each side of the root is equal. Other use of these trees is deriving a weight for each sequence.

Step three is the progressive alignment in which it utilises a series of pairwise alignments to align greater number of group sequences, base on the branching order in the guide tree. At every step a complete dynamic programming algorithm is applied by a weight matrix of residue and penalties for gap opening and extension. Every stage comprise of alignment of two alignments being there or sequences. Gaps which already exist in previous alignments maintain unchanged. New gaps are obtained at each step in basic alignments, get full gap opening and extension penalties, although they are achieved inside old gap places.

2.1.2 MUSCLE

MUSCLE can be described as a program for making multiple alignments of amino acid or nucleotide sequences. It can provide you with a range of options that enables the biologist with the choice of optimizing accuracy, speed, or some compromise between the two. By using kmer counting which is fast distance estimation in the algorithm, progressive alignment is applying a new function that it is called the log-expectation score and clarification by using tree dependent restricted partitioning (Edgar, 2004).

There are two distances measures that MUSCLE algorithm has used:

- kmer distance (used for unaligned pairs)
- Kimura distance (used for an aligned pair).

Kmer is a subsequence with the length of k, also can be called k-tuple. Those sequences that are related have more kmers in common compare to the other sequences. The kmer distance is built up from the fraction of kmers in common in a compressed alphabet. Since this measure does not need to have an alignment so it has an advantage

of much faster speed. In sequences of an aligned pair, we computed the pairwise identity then change to an additive distance approximate, implementing the Kimura correction for different alternatives at each site. Distance matrices are clustered using UPGMA, in which it has been found to provide some amended and improved results compare to neighbour-joining, expecting that neighbour-joining will come up with more accurate approximate of the evolutionary tree. This can be described by Edgar study conducted in 2004, in which he found that by considering that in progressive alignment, by aligning the two profiles that have lowest number of difference at each node the most precise accuracy can be achieved, although they are not evolutionary neighbours (Edgar, 2004). Figure 2.4 shows the steps of MUSCLE:

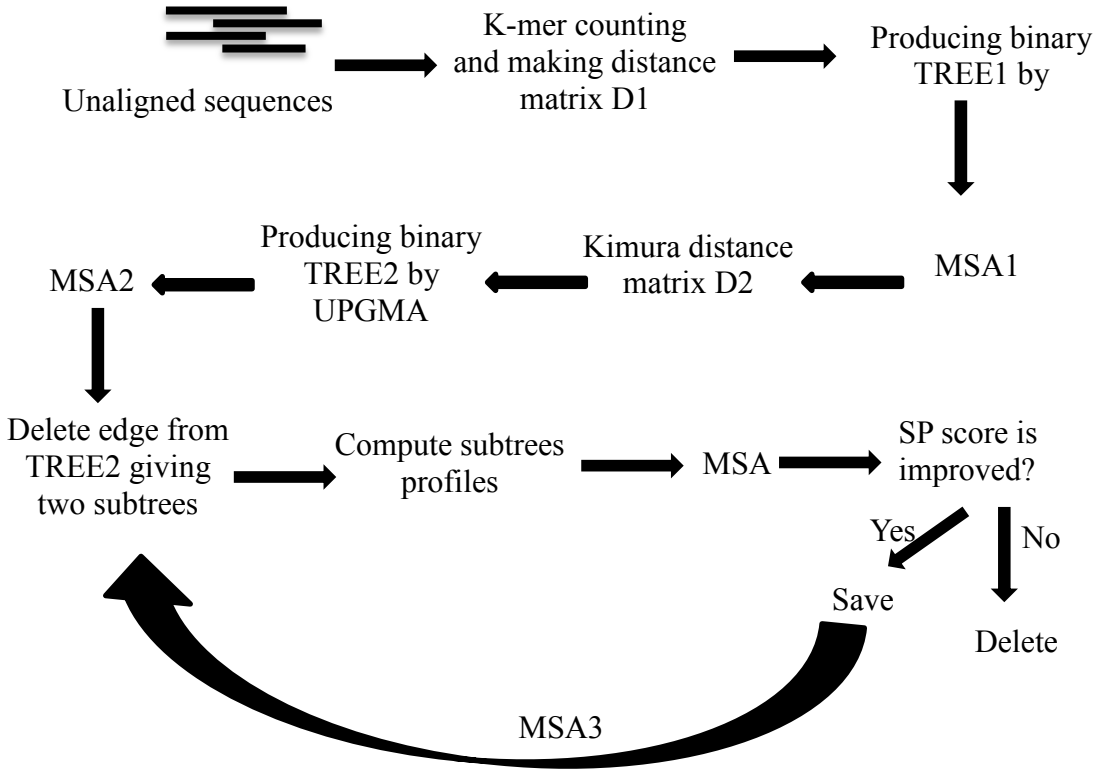

Figure 2.4. The flow of MUSCLE algorithm

Draft progressive is the first stage. Building a multiple alignment giving priority to faster speed rather than accuracy is the main goal of the first stage. In this stage;

1. Making distance matrix D1 by calculating the kmer distance for every single pair of input sequences.
2. Making binary tree TREE1 by clustering Matrix D1 using UPGMA.
3. A progressive alignment is built by considering the branching order of TREE1.

A profile is built from an input sequence in every single leaf. Nodes located in the tree are viewed in prefix order, in other words, parent right after their children. In every internal node, from two child profiles a pairwise alignment is constructed, providing a brand new profile which is given to that node. At the end these processes come up with a multiple alignment of all input sequences, MSA1, at the root.

Next section is improved progressive. Major root of error in the draft progressive stage is the approximate kmer distance measure produces in a suboptimal tree. Therefore MUSCLE algorithm re-produces the tree by using the Kimura distance, which is more precise in term of accurate but needs to have an alignment. So in this stage;

1. Distance matrix D2 is produced by calculating the Kimura distance for every pair of input sequences from MSA1.
2. Binary tree TREE2 is resulted by clustering matrix D2 using UPGMA.
3. A progressive alignment is resulted following TREE2 (same as first stage part 3) resulting multiple alignments MSA2. This is optimized by

calculating alignments just for sub-trees which its branching orders convert relative to TREE1.

Refinement is the last but not least stage. In this stage;

1. TREE2 is defining the edge (edges are seen in order of reducing the distance from the root).
2. TREE2 is reproduced into two sub-trees by removing the edge. The profile of the multiple alignments in every single sub-tree is calculated.
3. By re-aligning the two profiles new multiple alignments are produced.
4. In case of enhanced SP score, the new alignment is accepted, otherwise it is rejected.

2.1.3 T-COFFEE

A Multiple Sequence Alignment method which accommodates a fantastic enhancement in accuracy by using a little reduction in speed as compared to other used alternatives is so called T-Coffee. The method is mainly based on the popular progressive approach to multiple alignments but skips the major pitfalls resulted by the stingy nature of this algorithm. With T-Coffee we pre measure a data set of every single pair-wise alignments among the sequences. This enables us with a library of alignment data that can be used to lead the progressive alignment. Intermediate alignments which are based on the sequences to be aligned next, are based on the way that all of the sequences align with one another. This alignment data can be extracted from heterogeneous sources like a combination of alignment programs and/or structure superposition (Notredame, Higgins, & Heringa, 2000).

There are two main features in T-Coffee. First feature is that, T-Coffee enables a simple and elastic means of producing multiple alignments, by using heterogeneous data sources. All of the data from heterogeneous sources are fed to T-Coffee using a library of pair-wise alignments. Optimization method is considered to be the second main feature of T-Coffee, meaning that it is used to find the multiple alignments that best fit the pair-wise alignments in the input library.

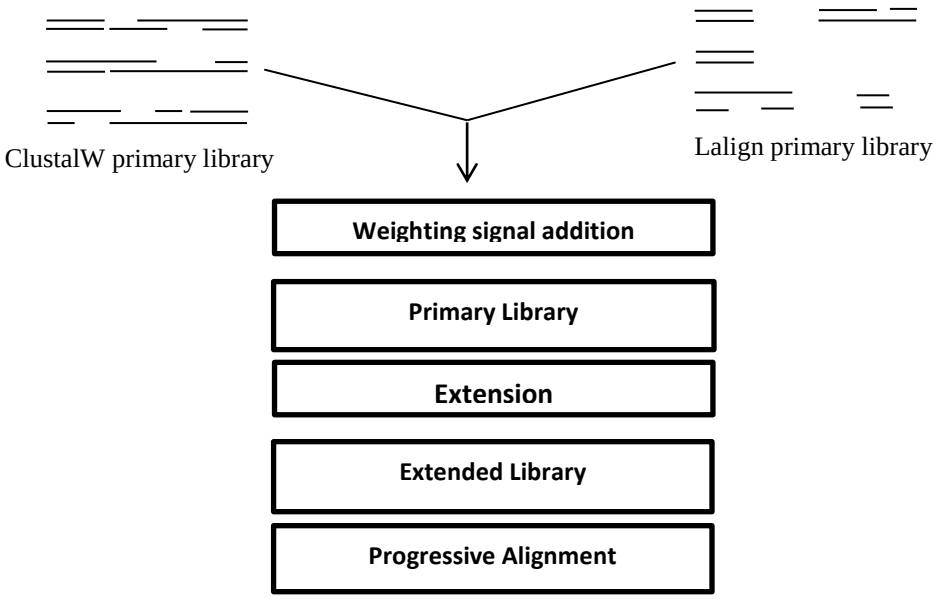

Figure 2.5. Basic procedure of T-Coffee

There are in total of five steps involved in T-Coffee process. The first step contains the generation of two sets of pairwise alignments, including one global and one local. The global alignments are produced by getting use of ClustalW on the sequences, two sequences at one time (default parameters; version 1.75). This will provide one full length alignment among every single pair of sequences. Local alignments on the other hand are the ten top scoring non-intersecting local alignments, among every single pair of sequences, putting together by using the Lalign program of the FASTA package with default parameters. Lalign is the FASTA implementation of the Sim program (Huang & Miller, 1991).

Determining the weights of the primary library is the second step in the process of T-Coffee. It is authorises a weight to every single pair of aligned residues in the main library. Best primary weights will be shown the correctness of a constraint. This weighting arrangement is very adequate for a previous consistency-based objective function.

In third step all of the libraries are combined. The goal is to combine local and global alignment information efficiently. This goal can be fulfilled by putting the ClustalW and Lalign primary libraries and add them together. In case of existence of any duplicated pair among the two libraries, it is put together and converted as a single entry in a way that its weight is the sum of the two weights. In other case, a brand new entry is built for the pair standing for the consideration. This primary library is able to directly be used to calculate a multiple sequence alignment. The worth of the information in the library is heightened by testing the continuity of every single pair of residues compare to residue pairs from all of the other alignments. For all of the pair of aligned residues existing is the library, we can assume a weight which can reflect the degree to which those residues align continently with residues from all the other sequences. This process is called library extension.

Library extension is fourth step. The problem is to fit a set of weighted constraints right into a multiple alignment. T-Coffee circumvents the complication by using a heuristic algorithm which is called library extension. The main goal is to merge information in a way that the final weight, for any pair of residues, reflects some of the information located in the whole library.

Progressive alignment strategy is the fifth and last step. In this step pair-wise alignments begin to make a distance matrix among all of the sequences, which is used

to build a guide tree by using the NJ method. The guide tree serves to guide the grouping of sequences during the multiple alignment process.

2.1.4 Limitations of Alignment-Based methods

Even though the algorithms have been used in alignment based methods appear acceptable, but by using large databases as input data, the computational load increases as a power function of the length of the sequences making its use unfeasible.

Many scoring systems have been proposed in recent decade, like amino acid substitution scoring matrices PAM (Daeyaert, Moereels, & Lewi, 1998) and BLOSUM (Henikoff & Henikoff, 1992) for protein alignment. This heuristic scoring systems demonstrates methodological short measurement in the access to sequence divergence, and also illustrates presumption of contiguity conservation among homologous potions. The interesting part is that no scoring diagram or chart in use will evaluate increasing the memory length (Vinga & Almeida, 2003).

The other limitation which is not often discussed is that heuristic solutions make it more difficult to evaluate the statistical relations of the resulting scores. Evaluations therefore are mostly in nature, implying that the global behaviour of these methods is unknown and confidence in there is judged purely on the basic of series of documented success and failure cases.

2.2 Alignment-free methods

Genetic recombination and, specifically, genetic shuffling are opposed with sequence comparisons by alignment, which considers conservation of contiguity between homologous segments. Few numbers of theoretical basics have been used to extract alignment-free methods that solve this drawback. Most of studies and researches have been conducted on alignment free sequence around the globe in past three decades which majority of the results have been published in the past decade years.

Vinga and Almeida(2003) proposed two major groups of methods. The first methods depended on word (oligomer) frequency. In the second types there is no need to resolve the sequence with fixed word length segments. First category is mainly based on the word frequency statistics, on the distances specified in a Cartesian space specified by the vectors of frequency, and as well as on the information details of frequency distribution. The second category uses Kolmogorov complexity and Chaos Theory. Alignment-free metrics are mostly being used as a predefine filters for alignment-based querying of huge applications. Studies conducted in the past few years are expanding their usage as a scale-independent methodology which is enables to recognise homology when loss of contiguity is above the possibility of alignment (Vinga & Almeida, 2003). Here, I briefly describe four alignment free methods which are Kr, D2, D2z and MplusD.

2.2.1 Kr

Kr (Domazet-Loso & Haubold, 2009) was produced as an alignment-free pair-wise distance measure in order to enable an efficient alignment-free method which can make biologically relevant evolutionary distances. According to DNA sequence evolutions of the Jukes-Cantor model, Kr can be explain and define as a predictor of the number of nucleotide substitutions per site. The rate of substitution predictor Kr is depended on the theory of pairwise the mean of the shortest unique substrings (shustring) which was basically defined to compare genomes. For example, consider two sequences, S1=ACCGT and S2=ACGGT, that we use as query and subject. At each of i positions in S1, the shortest substring S1[i..j] is determine that is absent from S2. As an example, S1 [1..3]=ACC is the briefest substring that started at the initial position in S1 that is absent from S2. This shustring`s length is 3 and in a compatible way we search for the lengths of the shustrings at every position in S1. The average function of these shustring lengths is Kr. The computation of lengths of shustring forms the main part of the *K*r calculation. Such lengths are the greatest looked up that used a suffix tree. In above example, the suffix tree of indexes sequences where is apart from the anchor *Si,* is a prefix of *Si*+1: *S*1 = A, *S*2=AC, *S*3=ACC and *S*4=ACCC. The used suffix *Si* [*j*..|*Si*|] is found in the suffix tree by linking the edge labels which are designated *i*,*j* from the root to a last node or leaf. Altogether, the found edge label by the path driving from the root to node *w* is called the path label of w. In our example, ACC is the path label of *w*1.

2.2.2 D2

The D2 method tries to compare the amount of likenesses among two biological sequences segments by using k-tuples (k-mers or k-words, k-grams). The D2 statistic is one of the most main employed statistics for sequence comparison regarding to k-tuples, which is depended on the joint of k-tuple content in the two sequences. Conceptually, when two sequences are mostly related, the k-tuple content of both two sequences are expected to be very likely similar. For example, two sequences, A = $A_1 A_2 \ldots A_n$ and B = $B_1 B_2 \ldots B_m$, are built of letters that are derived from a finite alphabet A of size d. For " a ϵ A" , let p_a shows the probability of a. For w = ($W_1, \ldots, W_1$) ϵ A_k, let:

$$X_w = \sum_{i=1}^{n} 1(A_i = w_1, \ldots, A_{i+k-1} = w_k) \qquad (1)$$

By counting the number of occurrences of A, and in the same way, Y_w calculates the number of w occurrences in B. In this formula n bar equals to, n – k + 1; in the same way, we use m bar which implies m – k + 1. Therefore D_2 is defined by equation 2. (Reinert, Chew, Sun, & Waterman, 2009)

$$D2 = \sum_{w \epsilon A^k} X_w Y_w \qquad (2)$$

The output of this method is a distance matrix in which the highest amount in each row will show the most similarity score.

2.2.3 D2z

D2z (Kantorovitz, Robinson, & Sinha, 2007) score is a score, derived from D2z algorithm in alignment-free sequence comparison. In this method all of the words with fixed length in two sequences frequencies compared together. The special and significant attribute of the D2z score is its comparability among sequence pairs carried from arbitrary foundation distributions. The use of the D2z score is proposed as such a 'normalized' measure that captures the statistical significance of the D2 score. It can be explained as the number of standard deviations by which the observed value of D2 deviates from its expected value under the background distribution. The calculation of D2z score is based on main assuming the generating of two sequences by Markov chains that may be dissimilar for the two sequences. Previously, the D2 statistic was introduced to be the number of k-word conforms among two sequences A and B, also containing flap overs. The computation is as:

$$D2 = \sum_{w \in A^k} X_w Y_w \quad (3)$$

The other way of studying D2 is by considering the central result of the words vectors counts in A and B. The computation of D2z score for the sequences A and B is as:

$$D2z\ (A, B) = \frac{D_2(A,B) - E(D_2)}{\sigma(D_2)} \quad (4)$$

In which E (D2) is the expectation and $\sigma(D2)$ is the standard deviation of D2 accordingly. D2z score calculates the number of standard deviations that the observed

value of D2 deviates from the mean. When the lengths of the sequences are large enough, the D2z statistic has a nearly standard normal.

2.2.4 MplusD

MplusD is alignment-free software which implements four statistical similarity measure proposed by Dai, Yang and Wang (2008) in order to calculate the similarity of a group of DNA sequences. It contains *rre,k,r* which is revised relative entropy, *wre.k.r*, weighted relative entropy, *S1.k.r* and *S2.k.r* which are symmetrical forms. The *rre.k.r* and the *S1.k.r*, are the statistical measurements between two different biological sequences according to Markov model. The *wre.k.r* and the *S2.k.r*, are the statistical measures between two different biological sequences according to Markov model and distributions of k-word. MplusD can be used in similarity search, evaluating the functionally which are related to sequences regulatory and building phylogenetic tree.

In the statistical similarity measure *wre.k.r* and *S*2.*k.r* by acquiring the *k*-word distributions into Markov model, here, a statistical model for every sequence is built. The similarity between two sequences can be found by calculating the log-likelihood difference between two sequences corresponding statistical models. Therefore, they can be simple alignment-free methods that concurs results reasonably.

2.2.5 Limitations of Alignment-Free methods

One of the limitations of alignment-free distances is that their relation to evolutionary events like substitutions is almost unknown (Haubold *et al*, 2009).

Other limitation of alignment-free sequence comparison is that frequencies of word or match lengths cannot be interpreted into mutation rates (Haubold *et al*, 2010).

2.3 Sequences GC content

Through a long range of genomic sequence, most often genes are characterized by having more GC-content in opposition to the basic GC-content for whole genome. There is an evidence of GC ratio, has illustrated that the coding sequence length is directly proportional to higher GC content. Therefore the longer the sequence, the higher GC bias (Pozzoli *et al*., 2008).

Remarkably, mammal and bird genomes are assembled into huge genomic areas (several hundreds of kilo bases) of related homogeneous base composition (that called isochores), fairly ranking from 30 percentages to 60 percentage of GC content (Bernardi, 2000). Some of important conditions of genome organization and evolution are reflected by isochore arrangement. Particularly, it has been illustrated that the GC content of isochores has relation with some other genomic features like length of gene density intron (Lander *et al*, 2001), time of replication (Watanabe *et al*, 2002), recombination (Kong, 2002), methylation pattern, and distribution of transposable elements. Therefore, finding the underlying mechanism that makes the evolution of isochores is an important issue in determining the organization of genomes (Meunier & Duret, 2004).