Stockholms universitet

Using Phylogenetic Contrast to Visualize Phylogenetic Signals With Sequence Logos

Användning av fylogenetiska kontrast för att visualisera fylogenetiska signaler med sekvenslogotyper

HaoLin Guo

Handledare: Lars Arvestad
Examinator: Marc Hellmuth
Inlämningsdatum: 2025-05-24

Kandidatuppsats i datalogi
Bachelor Thesis in Computer Science

Abstract

In bioinformatics, a common practice for visualizing patterns in sequence data for certain populations, for example, a group of DNA sequences, protein sequences, etc., is to generate a sequence logo that stacks existing sequence characters at sequence positions with different heights based on the frequency of each character at each position.
Traditional sequence logos visualize positional conservation in alignments (arranging the sequence characters and positions) but fail to account for phylogenetic non-independence, the evolutionary history of the sequences. Traditional sequence logos are generated solely on the basis of given sequence data, without accounting for the underlying connections between sequences of the sequence dataset. Additionally, some species or groups of species could be sampled less than other species, which will cause the pattern found in these species to be under-represented in the traditional sequence logos.
To address this, we investigate an approach that integrates phylogenetic trees into sequence logo generation by adapting principles from Phylogenetic Independent Contrasts (Felsenstein, 1985). Originally developed for continuous traits evolving under Brownian motion, PIC computes evolutionary contrasts between sister taxa, weighting sequences inversely by their shared branch lengths, aiming to correct for sampling bias, and converts correlated traits into independent contrasts.

Implemented as a Unix tool, picseqlogo takes a multiple sequence alignment file and a rooted phylogenetic tree file as input and outputs the sequence logo by visualizing the phylogenetically weighted symbol distribution. Compared to traditional sequence logos, the sequence logo generated by picseqlogo aims to preserve the evolutionary structure provided by the phylogenetic tree.

Sammanfattning

Inom bioinformatik är det vanligt att visualisera mönster i sekvensdata, till exempel DNA-sekvenser eller proteinsekvenser, genom att generera en sekvenslogo där tecken staplas vid varje sekvensposition med höjder proportionella mot varje teckens frekvens.
Traditionella sekvenslogon visualiserar konserverade positioner i en alignment (en ordnad uppställning av sekvenstecken och positioner) men tar inte hänsyn till fylogenetiskt beroende, d.v.s. sekvensernas evolutionära historia. Dessa logon genereras enbart baserat på given sekvensdata utan att beakta underliggande kopplingar mellan sekvenser i datamängden. Dessutom kan vissa arter eller artgrupper vara underrepresenterade i provtagningen, vilket leder till att mönster från dessa arter blir mindre synliga i traditionella sekvenslogon.
För att adressera detta problem undersöker vi en metod som integrerar fylogenetiska träd i genereringen av sekvenslogon genom att anpassa principer från Phylogenetic Independent Contrasts (PIC) (Felsenstein, 1985). Ursprungligen utvecklat för kontinuerliga egenskaper under Brownsk rörelse, beräknar PIC evolutionära kontraster mellan systertaxon och viktar sekvenser omvänt proportionellt mot deras delade grenlängder. Detta syftar till att korrigera för urvalsbias och omvandla fylogenetiskt korrelerade egenskaper till oberoende kontraster.

Metoden har implementerats som ett Unix-verktyg, picseqlogo, som tar en fil med multipla sekvensalignmenter och en fil med ett rott fylogenetiskt träd som indata och genererar en sekvenslogo baserad på den fylogenetiskt viktade symbolfördelningen. Jämfört med traditionella sekvenslogon syftar logon genererad av picseqlogo till att bevara den evolutionära strukturen som ges av det fylogenetiska trädet.

Contents

Page

1 Introduction

Sequence logos are a cornerstone of bioinformatics, visually representing positional conservation in biological sequences through stacked symbol heights proportional to their observed frequencies [1]. Widely used to identify functional motifs, binding sites, or evolutionary patterns, these logos assume statistical independence between sequences.

However, this assumption collapses in phylogenetically structured datasets, common in modern comparative genomics. For example, to find a peptide pattern in mice, dogs, and primates, it will be likely that the sequences of primates have the same or very similar versions of the pattern, while mice and dogs probably have more differences, but the pattern found in mice and dogs is drowned in the many samples from the primates. This misrepresentation in traditional sequence logo arises from treating sequences as independent observations, ignoring the shared evolutionary history encoded in phylogenetic trees [2].

Phylogenetic Independent Contrasts (PIC), introduced by Felsenstein, offer a solution by statistically disentangling phylogenetic non-independence through contrast calculations.

In this work, we propose a modified PIC framework for discrete sequence data that propagates the probabilities of ancestral symbols in a phylogenetic tree using branch lengths as evolutionary weights, bypassing contrast calculations while retaining PIC's algorithm structure to correct for sampling bias and the relations between sequences. By encoding sequences as positional frequency matrices and iteratively computing branch length weighted averages at internal nodes, we derive the root-state symbol distribution that reflects evolutionary, rather than sampling-driven conservation. This distribution is then visualized as a sequence logo, where symbol heights are adjusted to down-weight redundant evolutionary signals from densely sampled clades.

We implement this approach as a Unix tool picseqlogo that accepts a multiple sequence alignment file and a rooted tree file to generate phylogenetically corrected logos. Through case studies, we demonstrate how this approach resolves biases in datasets, recovering patterns obscured in traditional visualizations. Our method extends the utility of sequence logos to evolutionary contexts, offering a principled alternative to count-based approaches.

2 Background

This chapter provides an overview of the traditional sequence logo, what a sequence logo is and how it is generated; additionally, the essential terms included in the original PIC algorithm and our approach will also be discussed here to provide more context.

2.1 Sequence logo

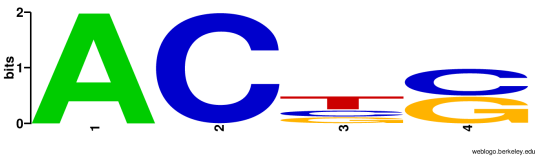


Figure 2.1: Sequence logo example

A sequence logo (Figure 2.1) is a graphical representation of conserved patterns in biological sequences (DNA, RNA, proteins, etc.) derived from a multiple sequence alignment [1], it visualizes two key features at each position in the sequences; the total height of each stack represents the positional sequence conservation (measured in bits), and the individual character heights within the stack reflect their relative frequencies at that position.

The sequence conservation [3] at a particular position R_{seq} is defined as the difference between the maximum possible entropy S_{max} and the entropy of the observed symbol distribution S_{obs}:

$$R_{seq} = S_{max} - S_{obs} = log_2(N) - (-\sum_{n=1}^{N} p_n \, log_2(p_n))$$

N is the number of distinct characters for the given sequence type, for DNA/RNA, $N = 4$ and for protein, $N = 20$, p_n is the observed frequency for character n at the particular sequence position.
The maximum possible entropy $S_{max} = log_2(4) = 2$bits for DNA/RNA sequence and $S_{max} = log_2(20) \approx 4.3$ bits for protein sequence. The individual character heights at each position will be $p_n \; R_{seq}$.
In tools created for generating sequence logos, a common way is to input the corresponding positional frequency matrix, with rows representing the sequence position and columns representing the possible character in the sequence, each entry then reflects the frequency of each character at each position for the whole sequence set.

2.1.1 Sequence logo example

For a small sequence set containing 4 DNA sequences: $S_a = ACTG, S_b = ACGG, S_c = ACTC, S_d = ACCC$, the corresponding positional frequency matrix M for this sequence set

will be:

$$M = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \end{array} \begin{array}{c} \begin{array}{cccc} A & C & T & G \end{array} \\ \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0.25 & 0.5 & 0.25 \\ 0 & 0.5 & 0 & 0.5 \end{bmatrix} \end{array}$$

Figure 2.1 is the sequence logo generated by this example sequence set $\{S_a, S_b, S_c, S_d\}$. With rows $1, 2, 3, 4$ and columns A, C, T, G representing positions and possible characters respectively, at position 1 and 2, character A and C covers the entirety of bits-axis since they have frequency 1, additionally, R_{seq} at position 3 and 4 is 0.5 and 1, respectively, so the sum of character heights will not cover the entirety of bits-axis.

2.2 Branch lengths in phylogenetic trees

A phylogenetic tree is a branching diagram that represents the evolutionary relationships among biological entities such as species, genes, or populations based on similarities and differences in their genetic or physical traits. These trees are fundamental to evolutionary biology, enabling researchers to infer historical patterns of divergence and identify shared ancestry [8].

Branch length as an attribute of phylogenetic trees encodes critical information about evolutionary divergence. Branch lengths quantify the expected amount of genetic change between nodes (ancestors) and their descendants, typically measured as substitutions per site [5]. For example, a branch length of 0.1 implies an average of 0.1 substitutions per site along that lineage, reflecting evolutionary time and/or rate heterogeneity across genomic regions [6].

The Phylogenetic Independent Contrasts (PIC) method relies on branch lengths to address phylogenetic non-independence in comparative studies, with branch lengths acting as variance weights, ensuring contrasts are standardized and statistically independent while also down-weighting redundant evolutionary signals during the process [7].

2.3 Contrast in PIC

In the original PIC framework, contrasts are standardized differences between the trait values of the sister nodes that account for their shared evolutionary history, since traits of closely related species are correlated due to their shared evolutionary history. These trait values at tree tips represent measurable characteristics of extant taxa in a phylogenetic tree, for example, morphological, physiological, or molecular features.

The contrasts transform trait data into statistically independent observations by taking the difference between trait values of sister nodes to allow parametric tests that assume the independence of data points [2].

3 Methods

This chapter provides an overview of the PIC method and algorithm introduced by Felsenstein, as well as how we adapted the algorithm in our work.

3.1 Phylogenetic Independent Contrast (PIC)

The Phylogenetic Independent Contrasts (PIC) method, introduced by Felsenstein (1985), is a comparative technique used to account for phylogenetic non-independence in trait evolution when testing hypotheses (e.g., correlations between traits).

PIC transforms trait values measured across species into independent contrasts at the internal nodes of a phylogenetic tree. These contrasts are statistically independent under a Brownian motion model of evolution, allowing valid hypothesis testing in regression or correlation studies[2].

The PIC algorithm assumes that the traits in the phylogenetic tree evolve via the Brownian motion model, indicating neutral drift and variance proportional to time. With a fully resolved (bifurcating) phylogenetic tree structure with known branch lengths, the branch lengths are used to represent evolutionary variance under the assumption of Brownian motion, since they accumulate with time and are a direct representation of the evolutionary time and divergence.

3.1.1 The PIC algorithm

Let X_i denote the trait value at node i, and v_i be the variance (branch length) leading to node i. As an iterative algorithm, PIC starts at the tips and moves iteratively toward the root.

For each pair of tips i and j, compute a contrast c_{ij}, the difference between two tip values that have expectation zero and variance proportional to v_i+v_j under the Brownian motion model.

$$c_{ij} = X_i - X_j$$

The contrast is then standardized by dividing by the square root of its variance (branch length).

$$s_{ij} = \frac{c_{ij}}{\sqrt{v_i + v_j}}$$

The trait value X_k for the parent node k of the pair of tips i and j is then estimated and assigned by the weighted average of X_i and X_j, with weights proportional to the inverses of the

variances v_i and v_j. The tips i and j are removed from the tree, leaving the parent node k as a tip.

$$X_k = \frac{\frac{X_i}{v_i} + \frac{X_j}{v_j}}{\frac{1}{v_i} + \frac{1}{v_j}} \tag{3.1}$$

The branch length v_k at the parent node k is then lengthened to address the error in estimating X_k.

$$v'_k = v_k + \frac{v_i v_j}{v_i + v_j} \tag{3.2}$$

The algorithm repeats $n-1$ times, where n denote the total number of tips in the phylogenetic tree, leading to $n-1$ contrasts when reaching the root of the phylogenetic tree. These contrasts are then used as statistically independent observations for studies that assume independence, which is not what we seek in this work.

3.1.2 Example

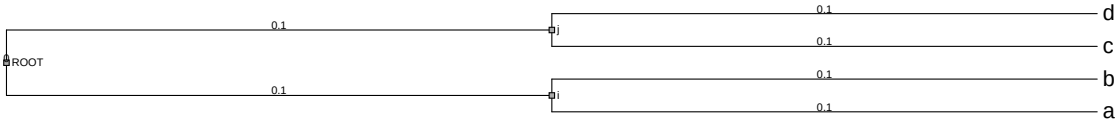


Figure 3.1: A simple tree example with tips a,b,c,d and length 0.1 for all branches between nodes.

A simple example tree (Figure 3.1) to apply the PIC algorithm to, with tips a, b, c, d and internal nodes i, j as parent nodes for pairs of nodes a, b and c, d, respectively, and a root r. All branch lengths between nodes in the example tree have length 0.1.

For pair of tips a, b, compute the contrast c_{ab}:

$$c_{ab} = X_a - X_b$$

The contrast is then standardized to recieve s_{ab}:

$$s_{ab} = \frac{c_{ab}}{\sqrt{v_a + v_b}} = \frac{c_{ab}}{\sqrt{0.2}}$$

Estimate the trait value X_i for the parent node i of tips a and b:

$$X_i = \frac{\frac{X_a}{v_a} + \frac{X_b}{v_b}}{\frac{1}{v_a} + \frac{1}{v_b}} = \frac{0.1X_a + 0.1X_b}{0.2} = 0.5X_a + 0.5X_b$$

Remove the pair of tips a, b and lengthen the branch length v_i:

$$v_i' = v_i + \frac{v_a v_b}{v_a + v_b} = 0.15$$

The algorithm is then repeated for the pair of tips c and d and their parent node j:

$$\begin{aligned} c_{cd} &= X_c - X_d \\ s_{cd} &= \frac{c_{cd}}{\sqrt{0.2}} \\ X_j &= 0.5X_c + 0.5X_d \\ v_j' &= 0.15 \end{aligned}$$

The algorithm repeats for the pair of tips i and j, which are former inner nodes, and their branch lengths are lengthened:

$$\begin{aligned} c_{ij} &= X_i - X_j = (0.5X_a + 0.5X_b) - (0.5X_c + 0.5X_d) \\ s_{ij} &= \frac{c_{ij}}{\sqrt{0.3}} \\ X_r &= \frac{0.15X_i + 0.15X_j}{0.3} = 0.25(X_a + X_b) + 0.25(X_c + X_d) \end{aligned}$$

The algorithm repeats 3 times and receives 3 contrasts, since the total number of tips is 4 in this example.

3.2 Our Approach

We adapt the idea of using evolutionary time/rate to estimate a weighted average from Felsenstein's PIC algorithm, and use the idea to estimate a weighted positional frequency matrix for the whole sequence dataset, to preserve the evolutionary information contained in the phylogenetic tree, and later visualize it in the sequence logo.

3.2.1 Implementation

Assume a rooted, bifurcating tree with branch lengths proportional to time or evolutionary rate, with corresponding aligned sequence strings for each tip.

Let S_i denote the corresponding sequence strings for each tip i, all with sequence length L; and let Σ denote the set of possible characters for the type of sequence strings (e.g., $\Sigma = \{A, C, T, G\}$ for DNA), with the number of possible characterc $C = |\Sigma|$.

Let v_i denote the branch length that leads to node i.

In the initial state, let M_i denote the trait value at the initial tips i; the trait value M_i is the positional frequency matrix with L rows and C columns, representing the corresponding sequence string.
Each row indexed by $l \in \{1, ..., L\}$ represents the sequence position in the corresponding sequence string, and each column is indexed by the possible characters $c \in \Sigma$. Each entry $M_i^{l,c} \in \{0, 1\}$ is either 1 if the character c is present at position l or 0 otherwise, in the initial state.

During the algorithm, the positional frequency matrix M_k for the parent nodes k will be estimated, until the algorithm estimates M_R for the root R, each entry of the estimated frequency matrix will have property $M_k^{l,c} \in [0, 1]$ instead of $\{0, 1\}$,

For each pair of tips i, j and their parent node k, the positional sequence matrix M_k is estimated using Equation 3.1 (Section 3.1.1), with M_k as X_k and M_i, M_j as X_i, X_j.

Remove the pair of tips i, j and leave the parent node k as a tip, the branch length v_k leading to the parent node k is then lengthened using Equation 3.2 (Section 3.1.1).

This algorithm repeats exactly $n - 1$ times, just as the original PIC algorithm, the positional sequence matrix M_R for root R of the tree is then estimated, with each entry $M_R^{l,c}$ representing the weighted probability of character c at position l for the data set as a whole. The sequence logo is then plotted by the 'WebLogo' package with X_R as input.

3.2.2 Example

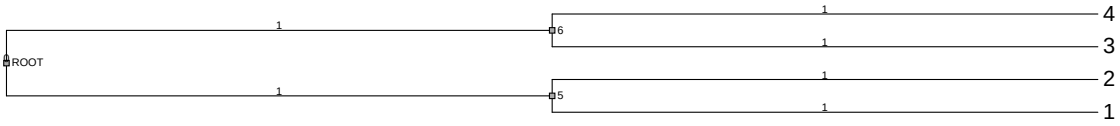


Figure 3.2: Example tree with 4 tips and branch length 1 between nodes.

A simple example for applying our modified algorithm to (Figure 3.2), with tips $\{1, 2, 3, 4\}$ and their corresponding DNA sequences $\{ACTG, GCTG, AGTG, CCCC\}$, additionally, all edges between nodes have branch length 1.
In this example, $\Sigma = \{A, C, T, G\}, L = 4$.

Map DNA sequences S_i to their corresponding tip trait value (positional frequency matrix) M_i:

$$S_1 = ACTG \Leftrightarrow \mathbf{M}_1 = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \end{array} \begin{array}{c} \begin{array}{cccc} A & C & T & G \end{array} \\ \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{array}$$

$$S_2 = GCTG \Leftrightarrow \mathbf{M}_2 = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \end{array} \begin{array}{c} \begin{array}{cccc} A & C & T & G \end{array} \\ \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{array}$$

$$S_3 = AGTG \Leftrightarrow \mathbf{M}_3 = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \end{array} \begin{array}{c} \begin{array}{cccc} A & C & T & G \end{array} \\ \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{array}$$

$$S_4 = CCCC \Leftrightarrow \mathbf{M}_4 = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \end{array} \begin{array}{c} \begin{array}{cccc} A & C & T & G \end{array} \\ \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \end{array}$$

Estimate the positional frequency matrix at the parent node 5 for the pair of tips 1 and 2:

$$\begin{aligned} M_5 &= \frac{\frac{M_1}{v_1} + \frac{M_2}{v_2}}{\frac{1}{v_1} + \frac{1}{v_2}} \\ &= \frac{\frac{M_1}{1} + \frac{M_2}{1}}{\frac{1}{1} + \frac{1}{1}} \\ &= \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \end{array} \begin{array}{c} \begin{array}{cccc} A & C & T & G \end{array} \\ \begin{bmatrix} 0.5 & 0 & 0 & 0.5 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{array} \end{aligned}$$

Lengthen the branch length v_5 at the parent node 5 :

$$\begin{aligned} v_5' &= \frac{v_1 v_2}{v_1 + v_2} + v_5 \\ &= \frac{1 \cdot 1}{1 + 1} + 1 \\ &= 1.5 \end{aligned}$$

The process repeats and receives:

$$M_6 = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \end{array} \begin{array}{c} \begin{array}{cccc} A & C & T & G \end{array} \\ \begin{bmatrix} 0.5 & 0.5 & 0 & 0 \\ 0 & 0.5 & 0 & 0.5 \\ 0 & 0.5 & 0.5 & 0 \\ 0 & 0.5 & 0 & 0.5 \end{bmatrix} \end{array}$$

$$v'_6 = 1.5$$

$$M_R = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \end{array} \begin{array}{c} \begin{array}{cccc} A & C & T & G \end{array} \\ \begin{bmatrix} 0.5 & 0.25 & 0 & 0.25 \\ 0 & 0.75 & 0 & 0.25 \\ 0 & 0.25 & 0.75 & 0 \\ 0 & 0.25 & 0 & 0.75 \end{bmatrix} \end{array}$$

The algorithm repeats 3 times to revive the estimated positional matrix M_R, which is then used as input to plot the sequence logo.

4 Experiments

This chapter presents how the visualization changes between the regular sequence logo and the adapted PIC applied sequence logo for different phylogenetic trees.

4.1 Sequence set

Label	Sequence									
S_a	A	A	A	A	A	C	A	A	C	Q
S_b	A	A	R	V	A	C	C	C	A	R
S_c	A	A	V	R	A	C	C	A	C	R
S_d	A	A	E	S	A	C	C	C	A	R
S_e	A	A	S	E	C	A	C	A	C	R
S_f	A	A	T	I	C	A	C	C	A	R
S_g	A	A	I	D	C	A	C	A	C	R
S_h	A	A	D	T	C	A	C	C	A	R

We created a simple DNA sequence set for testing such that $\Sigma = \{A, C, T, G\}, L = 10$, each sequence in the sequence set $\{S_a, S_b, S_c, ..., S_h\}$ corresponding to their tip in $\{a, b, c, ..., h\}$ at later sections.

4.2 Balanced tree

In this section, we create a balanced tree (Figure 4.1) with tips $\{a, b, c, d, e, f, g, h\}$ corresponding to the DNA sequences $\{S_a, S_b, S_c, ..., S_h\}$ and all edges between nodes have branch length 0.1.

As the sequence logos in Figure 4.2 indicate, applying our modified method to a balanced tree with identical branch lengths outputs an identical sequence logo as the regular sequence logo.

Since each pair of tips has the same branch length, they will be weighted the same, resulting in each of the estimated positional frequency matrices for their parent nodes containing exactly $\frac{1}{2}$

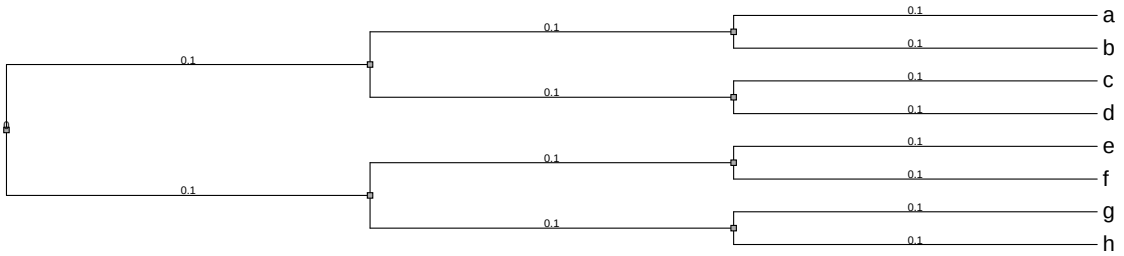


Figure 4.1: A balanced tree with 8 tips and branch length 0.1 between nodes.

(a) Regular sequence logo for the balanced tree.

(b) Sequence logo with modified PIC for the balanced tree.

Figure 4.2: The identical sequence logos generated for the example sequence set.

of the frequency of both tips.

The positional frequency matrix M_R estimated when reaching the root node will be the same as the positional frequency matrix received by counting the sequence characters.

4.3 Unbalanced tree

In this section, we create an unbalanced tree (Figure 4.3) that only has changes in branch lengths, with the same set of test sequences and tips. Due to the use of the same set of sequences, the regular sequence logo will remain unaffected by changes in the tree.

At the sequence positions 1 and 2 in the sequence logo generated with our modified method (Figure 4.4.b), the character heights of A are identical to the regular sequence logo (Figure 4.4.a), due to the fact that A has character frequency 1 at positions 1,2.

At the sequence position 7, C has frequency 0.875 and A has frequency 0.125, for the sequence position 10, R has frequency 0.875 and Q has frequency 0.125. Here, unlike the regular sequence

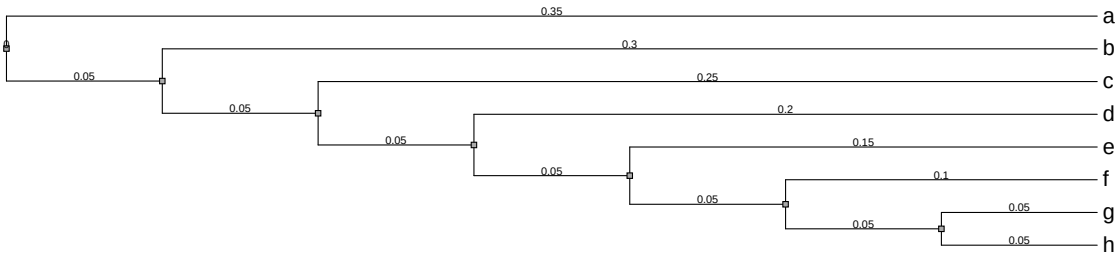


Figure 4.3: Unbalanced tree.

logo, our method emphasizes the evolutionary structure carried out by the phylogenetic tree and adjusts the character height of A and Q (A, S are character of S_a) since tip a forms a subtree by itself in this unbalanced tree.

At the sequence positions $3, 4$ and the sequence positions $5, 6, 8, 9$, the character heights of S_a are adjusted to retain the evolutionary structure of the phylogenetic tree, rather than generating the sequence logo solely based on the frequency of each character.

(a) Regular sequence logo stays unaffected by tree structure.

(b) Sequence logo with PIC for the unbalanced tree.

Figure 4.4: The sequence logos generated for the example sequence set with unbalanced trees.

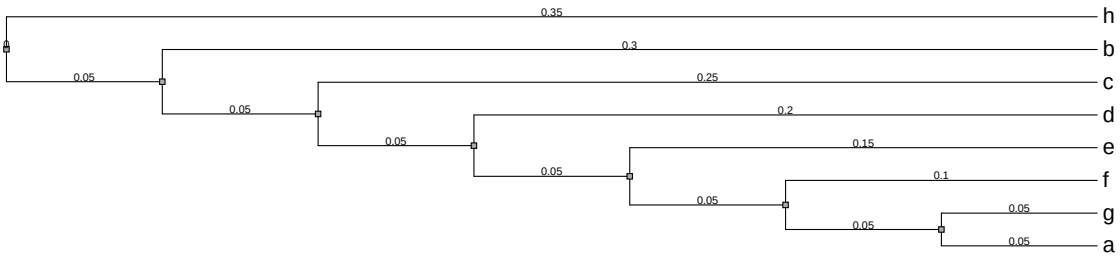
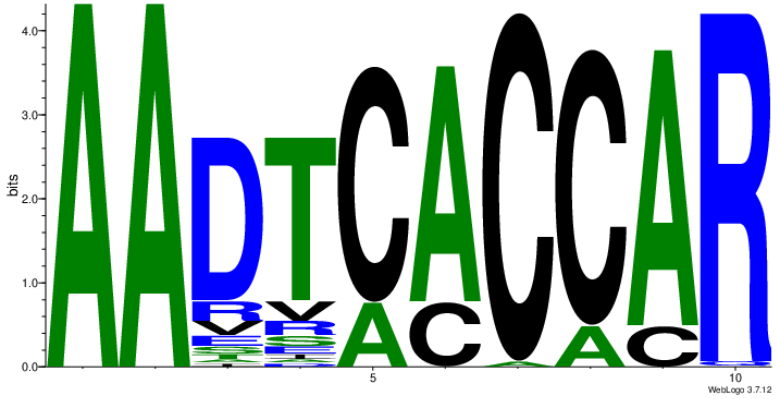


Figure 4.5: Modified unbalanced tree by swapping tips a and h.

4.3.1 Modified unbalanced tree

(a) Regular sequence logo stays unaffected by tree structure.



(b) Sequence logo for modified unbalanced tree with PIC.

Figure 4.6: The sequence logos generated for the example sequence set with modified unbalanced trees.

In this section, we use the same tree structure and branch lengths in the last section but swapped tips a and h (Figure 4.5) for further verification.

At the sequence positions $3, 4, 5, 6, 8, 9$ (Figure 4.6.b), we see the same pattern as in the previous

section, the character heights of S_h (D, T, C, A, C, A) are adjusted to retain the evolutionary structure since the tips a and h have changed locations.

The sequence positions $7, 10$ with a difference in the portion of character heights from the previous section (Figure 4.4.b) show that the frequency of characters (which are opposite to the position $7, 10$ from the previous section) also contributes to the heights of characters and is preserved in this modified PIC algorithm.

4.3.2 Unbalanced tree with different branch lengths

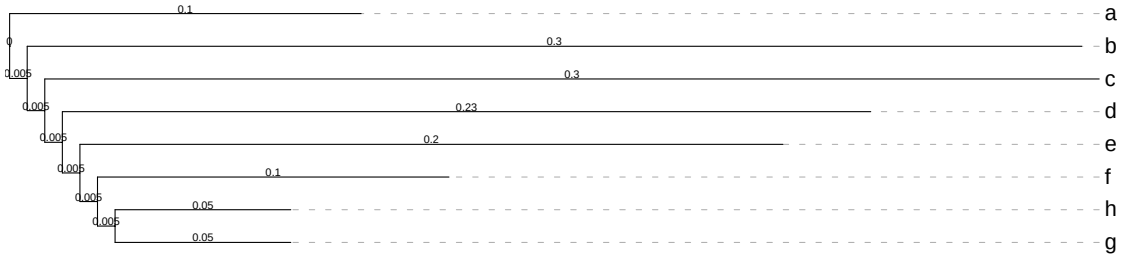


Figure 4.7: Unbalanced tree with different branch lengths.

In this section, we use the same unbalanced tree but with modifications in branch lengths instead (Figure 4.7).

The sequence logo (Figure 4.8.a) generated with PIC for this tree is generally close to the sequence logo for the unbalanced tree that has same root to tip branch length for all tips (Figure 4.1). With characters in the sequence S_a contributing most of the character heights, characters from other sequences also have slight changes in character heights since the changes in branch lengths.

(a) Sequence logo for unbalanced tree with PIC (Figure 4.4.b).

(b) Sequence logo for unbalanced tree (different branch lengths) with PIC.

Figure 4.8: The sequence logos generated with PIC for the unbalanced trees.

5 Result and Discussion

In this chapter, we apply the modified PIC to biological datasets and study the changes in the visualization of generated sequence logos.

5.1 PS00027

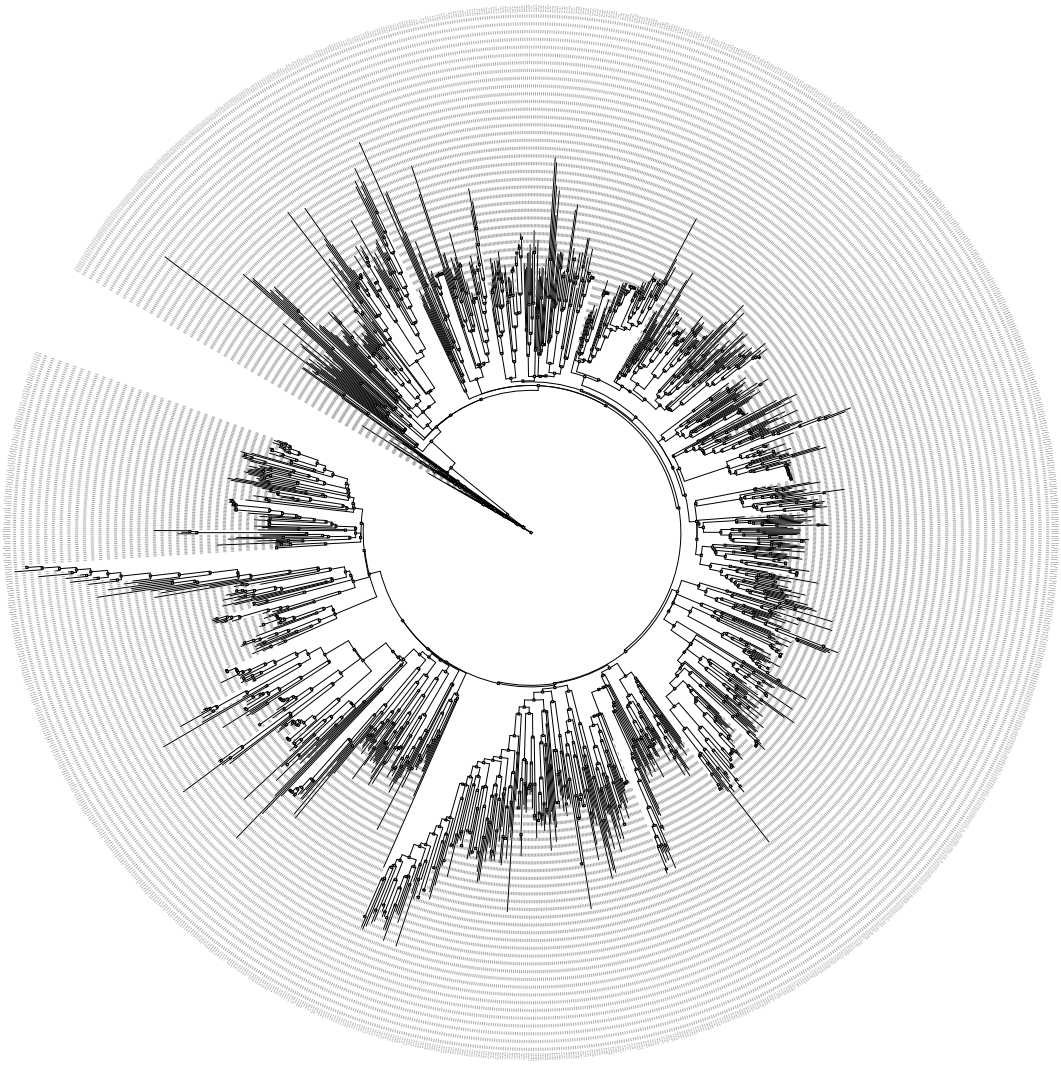


Figure 5.1: A phylogenetic tree with 1362 tips for dataset PS00027.

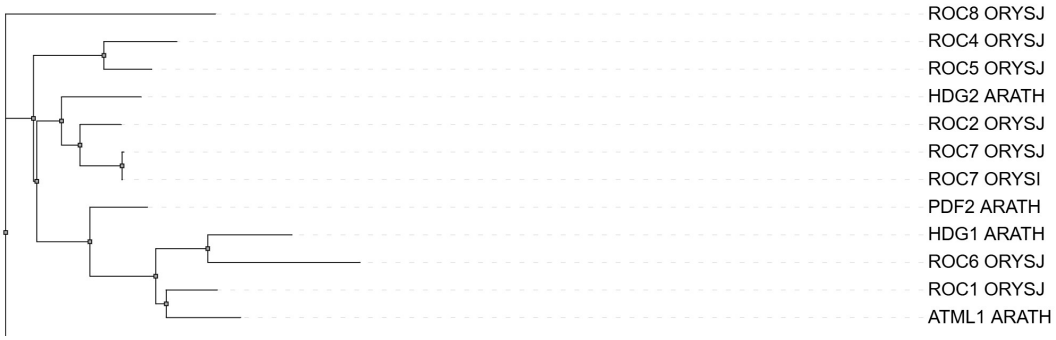


Figure 5.2: Two sub-trees formed by the two child nodes of the root, one with 1 tip and the other with 11 tips.

Prosite entry PS00027 [4] is a set of 1362 protein sequences describing a conserved ATP-binding domain found in ATPases and ATP-binding cassette (ABC) transporters.

The sequences from sequence set PS00027 are formed by the standard amino acid code, are as follows:

$$C = 20, \qquad \Sigma = \{A, C, D, E, F, G, H, I, K, L, M, N, P, Q, R, S, T, V, W, Y\}$$
$$L = 24.$$

With observations from examples in the previous chapter, it is worth noticing that this phylogenetic tree also forms smaller sub-trees that are close to the root and heavily influence the sequence logo generated by our algorithm.

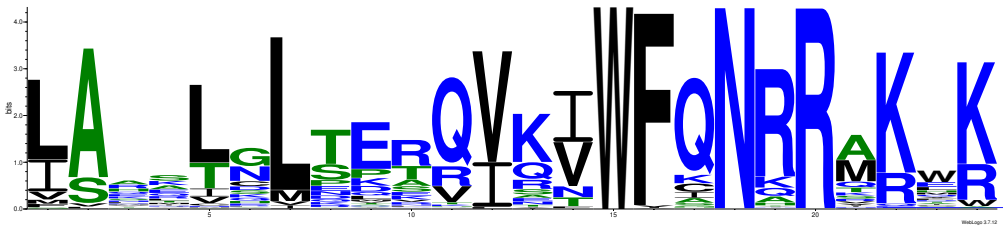
The root of the phylogenetic tree has three child nodes and forms three sub-trees, each with 1, 11, and 1350 tips, respectively (Figure 5.2).

Table 5.1: Sequences corresponding to the two sub-trees in Figure 5.2.

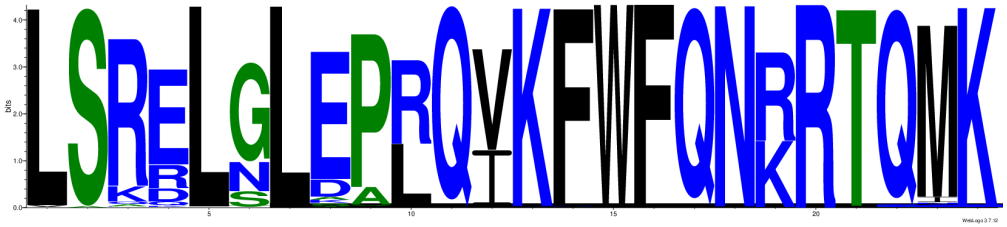
Sequence	**Position**																							
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
ROC8_ORYSJ	L	S	r	e	L	g	L	E	p	r	Q	I	K	F	W	F	q	N	r	r	t	q	m	K
ROC4_ORYSJ	L	S	k	r	L	g	L	E	p	r	Q	V	K	F	W	F	q	N	r	r	t	q	m	K
ROC5_ORYSJ	L	S	r	r	L	s	L	D	a	r	Q	V	K	F	W	F	q	N	r	r	t	q	m	K
HDG2_ARATH	L	S	r	e	L	n	L	E	p	l	Q	V	K	F	W	F	q	N	k	r	t	q	m	K
ROC2_ORYSJ	L	S	r	e	L	g	L	E	p	l	Q	V	K	F	W	F	q	N	k	r	t	q	m	K
ROC7_ORYSJ	L	S	r	e	L	g	L	E	p	l	Q	V	K	F	W	F	q	N	k	r	t	q	m	K
ROC7_ORYSI	L	S	r	e	L	g	L	E	p	l	Q	V	K	F	W	F	q	N	k	r	t	q	m	K
PDF2_ARATH	L	S	r	d	L	n	L	E	p	l	Q	V	K	F	W	F	q	N	k	r	t	q	m	K
HDG1_ARATH	L	S	r	r	L	n	L	D	p	r	Q	V	K	F	W	F	q	N	r	r	t	q	m	K
ROC6_ORYSJ	L	S	r	r	L	n	L	E	s	r	Q	V	K	F	W	F	q	N	r	r	t	q	m	K
ROC1_ORYSJ	L	S	r	e	L	g	L	E	p	l	Q	V	K	F	W	F	q	N	k	r	t	q	m	K
ATML1_ARATH	L	S	r	e	L	s	L	E	p	l	Q	V	K	F	W	F	q	N	k	r	t	q	m	K

Compared to the regular sequence logo (Figure 5.3.a), the sequence logo generated with PIC (Figure 5.3.b) is heavily influenced by the sequences (Table 5.1) of the two sub-trees.

This behavior of our approach, visualized in Figure 5.3.b follows what we observed in the previous chapter, where the character heights are adjusted to retain the evolutionary structure rather than solely depending on the frequency of characters. The sequence characters from the two sub-trees formed by the two child nodes of the root contributes significantly to the character heights at each position, the root of the phylogenetic tree should therefore be set with caution.



(a) Regular sequence logo for dataset PS00027.



(b) Sequence logo with PIC for dataset PS00027.

Figure 5.3: Visualization of the PS00027 dataset

5.2 PS00673

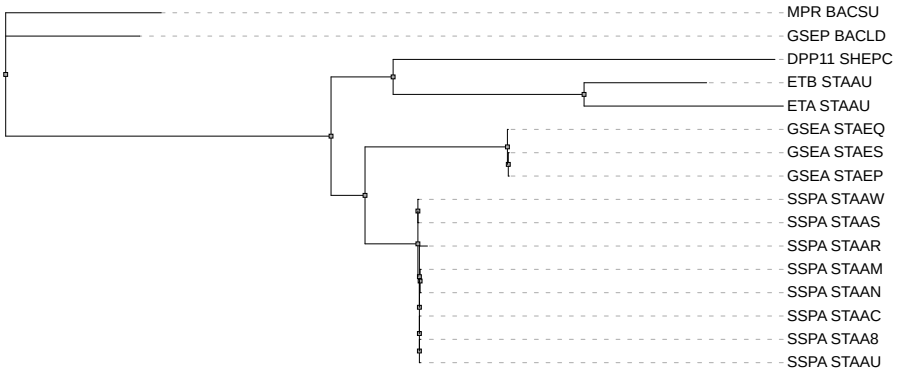


Figure 5.4: Phylogenetic tree for PS00673.

For the sequence set Prosite entry PS00673 [9] with 16 protein sequences of standard amino

acid, there are no clear differences in the generated sequence logos (Figure 5.5). At the sequence position 3, characters P and V have changes in frequency (characters at each position are ordered by their frequency) due to the structure of the phylogenetic tree.

(a) Regular sequence logo for dataset PS00673.

(b) Sequence logo with PIC for dataset PS00673.

Figure 5.5: Visualization of the PS00673 dataset

6 Conclusion

In this thesis, we explored an alternative way of generating sequence logos using our modified PIC algorithm, where we adapted Felsensteins technique and implemented sequence frequency matrix as traits in the original PIC algorithm.

This approach shows advantages in preserving the underlying evolutionary structure in the phylogenetic tree compared to the regular sequence logo, by using the evolutionary time/rate to weight the character frequencies. However, since this modified PIC algorithm uses a weighted average for estimation, in a pair of nodes with a significant difference in branch length, the subtree from the node with a shorter branch length could be over-represented. There will also be cases where the visualization will not see many changes.

Moving forward, future research could consider further studies on how the visualization changes with rerooting phylogenetic trees, which could be a solution to resolve the issue of over-weighting sub-trees closer to the root in some cases.

Bibliography

[1] Schneider, T.D., Stephens, R.M. (1990). Sequence logos: a new way to display consensus sequences. *Nucleic Acids Research*, 18(20), 6097–6100.

[2] Felsenstein, J. (1985). Phylogenies and the comparative method. *The American Naturalist*, 125(1), 1–15.

[3] Crooks, G.E., et al. (2004). WebLogo: A sequence logo generator. *Genome Research*, 14(6), 1188–1190.

[4] Prosite (2024). https://prosite.expasy.org/PS00027

[5] EMBL-EBI. https://www.ebi.ac.uk/training/online/courses/introduction-to-phylogenetics/what-is-a-phylogeny/aspects-of-phylogenies/branches/

[6] Rachel S Schwartz & Rachel L Mueller. 11 January 2010. Branch length estimation and divergence dating: estimates of error in Bayesian and maximum likelihood frameworks.

[7] Luke J. Harmon. Estimating Rates using Independent Contrasts.

[8] Inferring Phylogenies.—Joseph Felsenstein. 2003. Sinauer Associates, Sunderland, Massachusetts.

[9] Prosite (2024). https://prosite.expasy.org/PS00673

Datalogi
www.math.su.se

Beräkningsmatematik
Matematiska institutionen
Stockholms universitet
106 91 Stockholm