Review of ‘Comparing sequences without using alignments: application to HIV/SIV subtyping’

Supervised by Prof. Mathieu Defrance (*Machine Learning Group - ULB*)
& Prof. Wim Vranken (*Bioengineering Sciences Department - VUB*)

Master’s in Computer Science

Tomás GUIJA VALIENTE
Benjamin OBERTHÜR

Abstract

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Contents

Notations

Notation	Meaning
$[\![\cdot, \cdot]\!] : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$	Integer interval, $[\![a, b]\!] = \{a, a+1, \ldots, b\}$

I. Introduction and summary of the article

This article describes the problem in sequence comparison of the fact that when we want to compare a large number of sequences together, some types of sequence alteration like insertion or deletion are poorly - if ever - handled by classic sequence alignment methods. Intuitive ideas to solve this problem without alignment would be to look at the nucleotids or amino acids frequency, but this methods is not really meaningful, as sequences with similar frequencies can be a lot different. A more sophisticated - and more working - is dealing with what [Did+07] calls N-words (in nowaday's litterature, we call them k-mers). With those, we can compute dissimilarities between sequences [KL94] which can help us show evolutionary relationships between sequences.

I.1 Local decoding method of order N

The first step of this analysis presented by [Did+07] is the local decoding method of order N, or N-local decoding.

Definition I..1 (N-words)

> In a sequence, a N-word is a **contiguous sub-sequence of size** N of the given sequence. The set of its N-words is the its sub-sequences of size N.

Example.

> The set of the 3-words of the sequence $AGTACGT$ is AGT, GTA, TAC, ACG, CGT.

Let $S = S_1 S_2 \ldots S_i \ldots S_{|S|}$ be a sequence, i a site (or index) of S. For a given $N \in \mathbb{N}^*$, we consider the set of N-words of S covering the site i.

Definition I..2 (Direct relation)

> Two sites are said **directly related** if they have the same position in two (or more) occurences of the same N-word.

Example.

> (We take the example on Figure 6a in [Did+07]). Let seq1 = CATTG TCCGC **T**GGAC CACAC and seq2 = CACT**T** GGACA CATAC CATGC. We consider the site 11 in seq1 and the site 5 in seq2 (bolded in their definitions), and look at the 5-words covering this site (contained in the sites colored in red).

C	C	G	C	**T**	G	G	A	C
C	C	G	C	T				
	C	G	C	T	G			
		G	C	T	G	G		
			C	T	G	G	A	
				T	**G**	**G**	**A**	**C**

C	A	C	T	**T**	G	G	A	C
C	A	C	T	T				
	A	C	T	T	G			
		C	T	T	G	G		
			T	T	G	G	A	
				T	**G**	**G**	**A**	**C**

> The 5-word TGGAC appears in both these sequences, and the site 11 in seq1 and 5 in seq2 are both in first position of the 5-word, so these two sites are **directly related**

Definition I..3 (Transitivity and transitive closure)

> Let $\mathcal{R}$ and $\mathcal{R}'$ be binary relations. $\mathcal{R}$ is said to be **transitive** if it respects the following property [Cau21]:
>
> $$a\mathcal{R}b \wedge b\mathcal{R}c \implies a\mathcal{R}c$$
>
> $\mathcal{R}'$ is the **transitive closure** of $\mathcal{R}$ if [Sch77]:
>
> $$\forall a,\, b;\; a\mathcal{R}b \implies a\mathcal{R}'b$$

> and
>
> $$\forall a,\, b,\, c;\; a\mathcal{R}b \wedge b\mathcal{R}c \implies a\mathcal{R}'c$$

Definition I..4

> We define the (simple) relation between two sites as the transitive closure of the direct relation. Therefore, we say that two sites are related of there is a (finite) chain of direct relations linking those sites

We can divide those sites in a partition of relation classes[1]. [Did+07] calls them N-classes.

Definition I..5 (N-classes)

> An N-class can be defined as follows (with a being the identifier of the N-class):
>
> $$C(a) = \{x \in [\![1,\, |S|]\!];\; x \text{ is related with } a\}$$

Therefore, by giving an unique identifier to each N-class, we can rename each site of the whole sequence by the nucleotid (or amino acid) followed by the identifier of the class it is in. Only, there can be sites that are not not related to any other site in any other sequence, making singleton N-classes. As it could be rapidly unreadable to have large number of identifiers, and having some that appear in only one site, we denote these sites only by there nucleotide or amino acid.

I.2 Dissimilarity matrix and clustering tree

Now that we have our new sequences divided in disjoint N-classes of sites, we want to compare each sequence. For that, we need to choose a mesure of (dis)-similarity. [Did+07] chose the one defined in [Did+06].

Let's defining the following notation. $|s|_x$ is the number of occurences of the identifier x in the sequence s. For each pair of sequence and identifier $(s,\, x)$, we compute the value of $|s|_x$.

Example.

> We consider the following rewriten sequences:

seq1	$C\,A\,T_0\,T_1\,G_0\,T_2\,C_0\,C_1\,G\,C_2\,T_3\,G_1\,G_2\,A_0\,C_3\,C_4\,A\,C\,A\,C\,C\,T_0\,T_1\,G_0\,T_2\,C_0\,C_1\,C\,T\,A$
seq2	$C_5\,A_1\,C_6\,T_4\,T_3\,G_1\,G_2\,A_0\,C_3\,A\,C\,A\,T\,A\,C\,C\,A\,T\,G\,C$
seq3	$C_5\,A_1\,C_6\,T_4\,T_3\,C\,T\,T\,T\,C\,C_2\,T_3\,G_1\,G_2\,A_0\,C_3\,C_4\,T\,C\,C$

Table 1: Rewriten sequence (picked from [Did+07])

	seq1	seq2	seq3
A_0	1	1	1
A_1	0	1	1
T_0	2	0	0
T_1	2	0	0
T_2	2	0	0
T_3	1	1	2
T_4	0	1	1
C_0	2	0	0
C_1	2	0	0
C_2	1	0	1
C_3	1	1	1
C_4	1	0	1
C_5	0	1	1
C_6	0	1	1
G_0	2	0	0
G_1	1	1	1
G_2	1	1	1

Table 2: Count of each identified site in each sequence

[1] The proof comes from the fact that the relation we described is an equivalence relation, i.e. it is reflexive, symmetric, and transitive

Then to compute the similarity between each sequences, we apply the following formula:

$$\text{sim}(\text{seq},\,\text{seq}') = \frac{\sum_x \min(|\text{seq}|_x,\,|\text{seq}'|_x)}{\min(|\text{seq}|,\,|\text{seq}'|)}$$

Example.

> for $i,\,j \in \{\text{seq1},\,\text{seq2},\,\text{seq3}\}$:
>
> $$\text{Sim} = (\text{sim}(i,\,j))_{i<j} = \begin{pmatrix} - & \frac{5}{20} & \frac{7}{20} \\ & - & \frac{9}{20} \\ & & - \end{pmatrix} = \begin{pmatrix} - & 0.25 & 0.35 \\ & - & 0.45 \\ & & - \end{pmatrix}$$

The dissimilarity (or distance) can easily be obtained by taking the complement to 1 of the similarity.

I.3 Clustering and trees

Now that we have our mesure of distance between sequences, we can perform an agglomerative hierarchical clustering [And73] that can give us an idea of a potential mutative evolution of HIV and SIV, that can be interpreted as subtypes of these viruses. However, in order to perform this clustering, we need a method to compute the distance between clusters containing multiple sequences. The article does not present one, so we add to design our own described in Subsection II.2

I.4 Bootstrapping

Finally, in order to have more confidence in the results, the authors use the statistical technique of 'bootstrapping' [Efr79]. In the sequence comparison case, the method proposed in [Fel80] was considered and adapted to local decoded sequences. The motivation to use this method is that the obtaind results may have some degree of uncertainty due to the noise of limitations in the data. Therefore, from the input data, we create an artificial set of sequence replicates. For a given sequence of the original data, we create an random replicate of the same size by randomly picking (sampling with replacement) and concatenating sites of the original data. After this artificial sets has been created, we can run the same analysis we made on the original set on our new set, and judge of the stability of our method with the new result.

II. Material & Method

II.1 Data Sets

We used two different sequence sets to run our experiments: one given from Gilles Didier, co-author of [Did+07], and another that we retrieved ourselves, both coming from the data base maintened by the 'Los Alamos National Laboratory'[2]. The first set contains 66 sequences writen prior to 2007 [3], and the second one has been created from the query to have 47 sequences sampled in 2016 in Germany (so that we look at mutations in a restrained place and time).

II.2 Cluster merging method

For our cluster merging method, we inspired ourselves of centroid linkage methods [DH+73]. For a given cluster $\mathcal{C}$ containing more than one sequence, we create an artificial sequence, of the size of the shortest sequence of the cluster, defined as follows:

$$\widetilde{\text{seq}}_i = \arg\max_{s_i;\, s\in\mathcal{C}} \frac{s_i}{|\mathcal{C}|}, \quad \forall i \in [\![1,\,n]\!],\, n = \min_{s\in\mathcal{C}} |s|$$

In a more natural language, for each site of the artificial sequence, we take the N-class appearing the most in the same site of the different sequences odf teh cluster. Obviously, there can be N-classes having the same frequency of occurence, therefore, we take the first element having the larger frequency. Then, to compute the distance between two clusters, we use the same formula as shown in Subsection I.2.

[2] `https://www.hiv.lanl.gov`

[3] The sequences presented in figure 1 of [Did+07], with 4 missing sequences

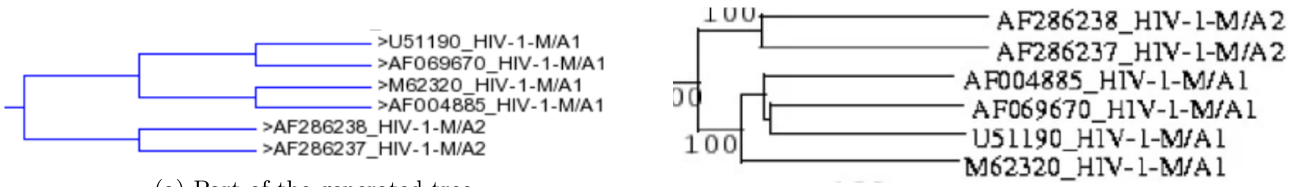
III. Experiments and Results

III.1 Original set of sequence

III.1.1 Comparison of the clustering trees

We applied the procedure described in the summary (Section I.). The obtained dissimilarity matrix of the set of sequences can be found in the `dissmatrix__66_sequences.txt` file in the repository, as it does not fit in a single to be readable. You can find the associated cluster tree of this dissimilarity matrix in Figure 3, in the Appendix, following the cluster tree of the original article in Figure 2.

When looking at both the figures, we can see that they are very alike. Let's take the following sequences in example:



(a) Part of the generated tree

(b) Part of the original tree

We can see on these parts of the trees that the only difference is that, on the generated tree, the sequences `M62320` and `AF004885` are 'siblings' (i.e. formed a cluster alone together at some point of the hierarchical clustering), where in the original tree, they are one 'generation' apart. These kind of changes can be explained by the fact that our clustering method are probably not the same, the one of the article not being presented, and by the fact that we miss 4 sequences from the original set.

The only potentially significant difference between our two trees is that, in ours, the sequence `M31113` is directly linked to the root, with no direct sibling, when it is a lot more deeply inserted in the original tree.

III.1.2 Bootstrapping

We performed 10 different bootstraps in order to see if our method is reliable. You can find the 10 trees at the following path: `Trees/Bootstraps_66_sequences`. All the trees have the same structure, with 3 blocks of sequences never mixing up between themselves, plus the sequence `M31113` that always has the same position in all the bootstrap trees. We can still notice some permutations between sequences in the different blocks. Those permutations stay in the deeper half of the tree for the 'large block'.

IV. Conclusion

Appendix

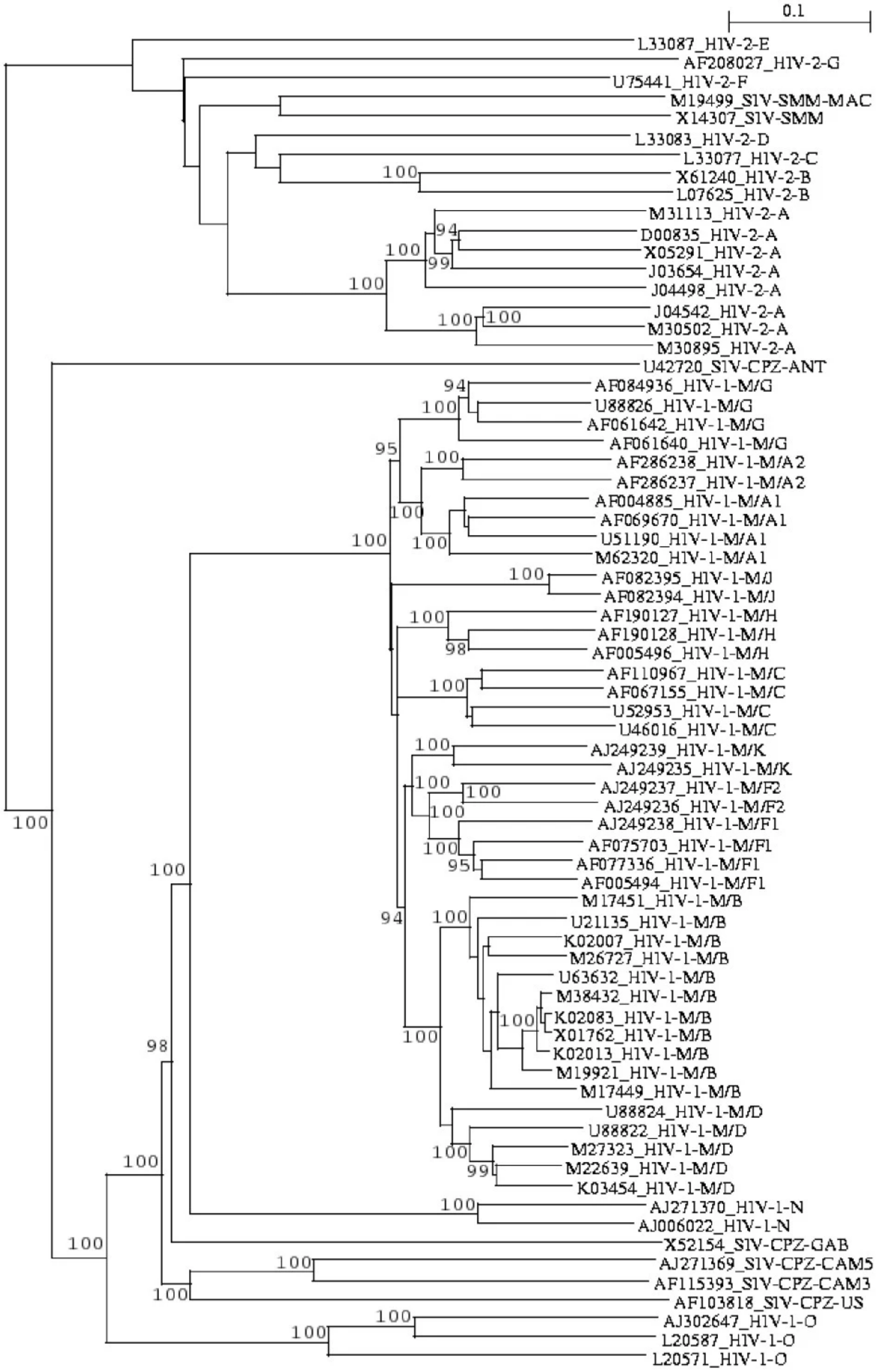


Figure 2: Original tree obtained from the dissimilarity matrix of the sequences of the original article, with $N = 15$

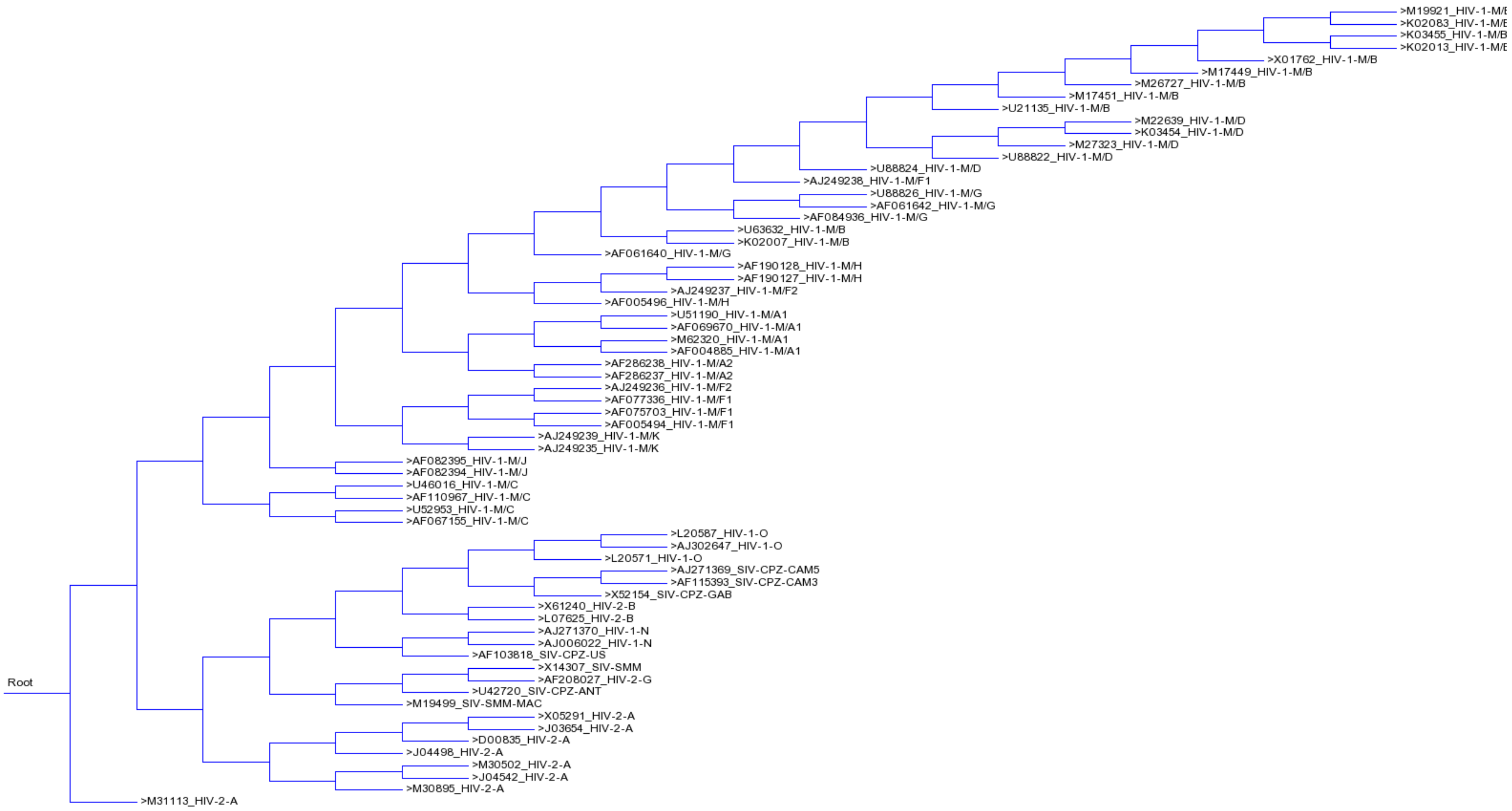


Figure 3: Tree obtained from the dissimilarity matrix of the sequences of the original article, with $N = 15$

References

[Cau21] Augustin-Louis Cauchy. *Cours d'analyse de l'École royale polytechnique. Analyse algébrique.* .1ère partie. Imprimerie royale, 1821.

[Sch77] Ernst Schröder. *Der Operationskreis des Logikkalkiils.* Ed. by Teubner-Verlag. Leipzig, 1877.

[And73] Michael R. Anderberg. "CHAPTER 6 - HIERARCHICAL CLUSTERING METHODS". In: *Cluster Analysis for Applications.* Ed. by Michael R. Anderberg. Probability and Mathematical Statistics: A Series of Monographs and Textbooks. Academic Press, 1973, pp. 131–155. DOI: https://doi.org/10.1016/B978-0-12-057650-0.50012-0. URL: https://www.sciencedirect.com/science/article/pii/B9780120576500500120.

[DH+73] Richard O Duda, Peter E Hart, et al. *Pattern classification and scene analysis.* Vol. 3. Wiley New York, 1973.

[Efr79] B. Efron. "Bootstrap Methods: Another Look at the Jackknife". en. In: *Ann. Statist.* 7.1 (1979), pp. 1–26. URL: http://dml.mathdoc.fr/item/1176344552.

[Fel80] J Felsenstein. "Confidence intervals on phylogenies: an approach using bootstrap". In: *Evolution* 39 (1980), pp. 1792–1797.

[KL94] Samuel Karlin and Istvan Ladunga. "Comparisons of eukaryotic genomic sequences." In: *Proceedings of the National Academy of Sciences* 91.26 (1994), pp. 12832–12836.

[Did+06] Gilles Didier et al. "Local Decoding of Sequences and Alignment-Free Comparison". In: *Journal of Computational Biology* 13.8 (2006). PMID: 17061922, pp. 1465–1476. DOI: 10.1089/cmb.2006.13.1465. eprint: https://doi.org/10.1089/cmb.2006.13.1465. URL: https://doi.org/10.1089/cmb.2006.13.1465.

[Did+07] Gilles Didier et al. "Comparing sequences without using alignments: application to HIV/SIV subtyping". In: *BMC Bioinformatics* 8.1 (Jan. 2007), p. 1. ISSN: 1471-2105. DOI: 10.1186/1471-2105-8-1. URL: https://doi.org/10.1186/1471-2105-8-1.