Deriving BLOSUM-like matrices for protein-coding DNA

Alexander Temper

September 8, 2023

Abstract

Sequence Alignment algorithms rely on some form of scoring matrix, which can heavily influence their results. In this work, we try to derive scoring matrices for the deoxyribonucleic acid (DNA) of subsets of protein families for specific taxa akin to the BLOSUMx, which are derived for amino acid sequences.

Whilst the lack of proper benchmark data allows for no definitve quantitative statement about the performance of the derived matrices, we find that there is ample diversity among them to be further investigated.

Contents

1 Introduction

Sequence alignment is the task of pairing up the elements of (biological) sequences with the aim of finding conserved regions and homologous sequences anticipated to have some evolutionary relation. Sequence alignment algorithms are primarily used for constructing phylogenetic trees [3], searching for homologous sequences with BLAST [1], assembling full DNA strands from short reads, creating input features for machine learning algorithms like AlphaFold2 [13] and others [3]

Most alignment algorithms try to maximize a *scoring function*, which assigns a measure of supposed quality to a given alignment. The classical methods score alignments by summing up scores assigned to each pair of letters in the given alignment. The scores for the pairs of letters can be represented by a symmetric matrix — we call such matrices "*scoring matrices*".

Scoring matrices for amino acids have received significant attention and thus, there is guiding theory to their construction. However, nucleotide sequences have been neglected. In practice, when doing database searches with BLASTn, the scoring matrices are picked through intuition somewhat arbitrarily — this is the main motivation of this investigation, in which we derive BLOSUMs for DNA. Let us introduce the problem semi-formally.

Definition 1 *The following will be used throughout the paper:*

- *A* sequence s *is an ordered collection of letters from an alphabet* $\mathcal{L}$.
- *The letter '-' is called* indel *and represents gaps in the alignment, which typically occur due to mutations or sequencing errors.*
- *A (sequence) alignment is a matrix* $\mathbf{A} \in (\mathcal{L} + \{-\})^{n\times m}$, *where* n *is the number of aligned sequences and* m *is the length of the alignment. An alignment with* $n = 2$ *is called a* pairwise sequence alignment.
- *A* scoring function $\sigma : (\mathcal{L} + \{-\})^{n\times m} \to \mathbb{R}$ *maps an alignment to its score.*
- *A pairing of two nuleotides* a, b *in an alignment is called a* "match" *if* $a = b$, *a* "mismatch" *if* $a \neq b$ *and a* "gap" *if* $a =$ "-" *or* $b =$ "-".

The classical algorithms for pairwise sequence alignment use a scoring function of the form

$$\sigma_s(\mathbf{A}) = \sum_{j=0}^{m} s(\mathbf{A}_{1j}, \mathbf{A}_{2j}),$$

where $s : (\mathcal{L} + \{-\})^2 \to \mathbb{R}$ is a symmetric function evaluating pairs of letters, which can be represented by a matrix $\mathbf{S} \in \mathbb{Z}^{(\#\mathcal{L})+1\times(\#\mathcal{L})+1}$. Said matrix is the focus of this work.

Example 1 *Our alphabet of concern will be the 4 different nucleotide bases,* $\{\mathtt{A, C, G, T}\}$. *Two exemplary DNA sequences are* ACA *and* AAGA. *One possible alignment between them is*

$$\mathbf{A} = \begin{bmatrix} \mathtt{A} & \mathtt{C} & - & \mathtt{A} \\ \mathtt{A} & \mathtt{A} & \mathtt{G} & \mathtt{A} \end{bmatrix}.$$

An exemplary scoring matrix is

$$\mathbf{S} = \begin{array}{cccccc} & \mathtt{A} & \mathtt{C} & \mathtt{G} & \mathtt{T} & - \\ \mathtt{A} & 1 & -2 & -3 & 0 & 0 \\ \mathtt{C} & -2 & 0 & 0 & 0 & 0 \\ \mathtt{G} & -3 & 0 & 0 & 0 & -4 \\ \mathtt{T} & 0 & 0 & 0 & 0 & 0 \\ - & 0 & 0 & -4 & 0 & 0 \end{array}.$$

Using the above matrix,

$$\begin{aligned} \sigma_{\mathbf{S}}(\mathbf{A}) &= s(\mathtt{A},\mathtt{A}) + s(\mathtt{C},\mathtt{A}) + s(-,\mathtt{G}) + s(\mathtt{A},\mathtt{A}) && (1) \\ &= 1 - 2 - 4 + 1 && (2) \\ &= -6. && (3) \end{aligned}$$

1.1 BLOSUM

Famously, one family of scoring matrices are called BLOSUMs (**BLO**ck **SU**bstitution **M**atrices), first constructed by Henikoff and Henikoff [10] for aligning amino acid sequences. A wonderful explanation thereof was written by Eddy [6]. However, we shall briefly dive into the theoretical underpinnings of BLOSUMs here as well.

Underlying BLOSUMs is the equation

$$s(a,b) = 2\log_2 \frac{P(a,b)}{P(a)P(b)} \text{ where } a \in \mathcal{L}, b \in \mathcal{L}.$$

We can understand this as follows: We want to derive a score we assign for aligning the two letters a and b, $s(a,b)$. We can represent such a score by the odds of the probability of aligning a, b being observed, $P(a,b)$, versus the probability of the two being paired due to chance. In probablistic terms this is $P(a \cup b) = P(a)P(b)$.

Now, assuming we have a dataset with already aligned data we deem to be good, we can approximate $P(a,b)$ by counting how often a and b were aligned, and we can also approximate $P(a)P(b)$ by counting how often both a and b appear individually and multiplying the resulting two counts together.

In this fashion, the original BLOSUM was constructed from the BLOCKS database [11], which provided gapless multiple sequence alignments (MSAs) called BLOCKS, frrom which $P(a,b)$ and $P(a)$ and $P(b)$ were approximated in a frequentist fashion. Given the absence of gaps, the re- sulting scoring matrix does not include scores for gaps. We now follow up with a somewhat precise definition of a simplified version of the algorithm to compute the BLOSUM.

Definition 2 *A BLOCKS database* $\mathbf{Z}$ *is a sequence of MSAs of differing shapes, i.e.,* $\mathbf{Z} = (\mathbf{A}^k)_{k=1}^l$ *Let* $\mathbf{C} \in \mathbb{N}^{4\times 4}$ *be a symmetric matrix with counts of how often each nucleotide was aligned to another nucleotide in each column of each BLOCK. Let* $\mathbf{Q}$ *be the matrix of the relative frequencies of the observed pairs, i.e.,*

$$\mathbf{Q}_{i,j} = \mathbf{C}_{i,j} / \sum_i^4 \sum_j^i \mathbf{C}_{i,j} j$$

Further, we denote the relative frequencies of each nucleotide in $\mathbf{Z}$ *as* $\mathbf{p}$*, and it can be computed from* $\mathbf{Q}$ *via* $\mathbf{p}_i = \mathbf{Q}_{i,i} + \sum_{j\neq i, j=0}^4 \frac{\mathbf{Q}_{i,j}}{2}$.

Example 2 *Let* $\mathbf{Z}$ *consist of two BLOCKS:*

```
AA  AA
AC  AC.
AC  AC
```

For the first block in the first column, we have three pairs of `AA`*, and in the second column we have two pairs of* `AC` *and one pair of* `CC` *The second is identical, so the counts are identical. Assuming the indexes* $\mathtt{A} = 1, \mathtt{C} = 2, \mathtt{G} = 3, \mathtt{T} = 4$*, we get*

$$\mathbf{C} = \begin{bmatrix} 6 & 4 & 0 & 0 \\ 4 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

thus

$$\mathbf{Q} = \begin{bmatrix} 0.5 & 0.33 & 0 & 0 \\ 0.33 & 0.16 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

and

$$\mathbf{p} = \begin{bmatrix} 0.66 & 0.33 & 0 & 0 \end{bmatrix}.$$

Now the assumption is that, given that i is the index of a and j is the index of b,

$$s(a,b) \approx 2\log_2 \begin{cases} \frac{\mathbf{Q}_{i,j}}{2p_i p_j} & i \neq j, \\ \frac{\mathbf{Q}_{i,j}}{p_i^2} & i = j \end{cases}$$

— rounding these values to the nearest integer then gives us our scoring matrix $\mathbf{S}$.

Remark 1 *One might rightfully ask why we apply a binary logarithm and a doubling to the odds. The logarithm of odds, called* log-odds*, are fairly standard in statistics — odds are by definition positive, and opposing odds, i.e., those where the* $\frac{a}{b} < 1$*, span only* $(0,1)$*, whereas good odds have a range of* $[1,\infty)$*. Applying a logarithm counteracts this by making the function skew symmetric, i.e.,* $\frac{a}{b} \neq -\frac{b}{a}$ *but* $\log\frac{a}{b} = -\log\frac{b}{a}$.

To the best of our knowledge, there is no distinct reason why the number 2 is picked for the logarithm and the multiplier - we suspect this is due to its connection to information theory.

An important piece of the algorithm is still missing - the clustering of sequences within BLOCKS. We will introduce this more informally: we pick a similarity threshold x, we then check within each which sequences have $\geq x\%$ identity and then "average" the sequences transitively, i.e., if sequence i and sequence j are to be clustered and sequence j and sequence k as well, then sequence i, sequence j and sequence k are being clustered together. The count matrix $\mathbf{C}$ now includes fractional counts, and the rest of the computation remains the same. This is why the matrices are called BLOSUMx — a BLOSUM90 matrix is made from blocks where sequences of 90% similarity are clustered.

Example 3 *Let*

$$\mathbf{A} = \begin{matrix} \mathtt{AAC} \\ \mathtt{AGC} \\ \mathtt{AGG} \\ \mathtt{TTT} \end{matrix}$$

and $x = \frac{2}{3}$*. Sequences* $(1,2)$ *and* $(2,3)$ *exceed the similarity threshold, thus the clustered will be*

$$c(\mathbf{A}) = \begin{matrix} \mathtt{A}(\frac{2}{3}\mathtt{G}\frac{1}{3}\mathtt{A})(\frac{2}{3}\mathtt{C}\frac{1}{3}\mathtt{G}) \\ \mathtt{TTT} \end{matrix}$$

The resulting count matrix

$$\mathbf{C} = \begin{bmatrix} 0 & 0 & 0 & \frac{4}{3} \\ 0 & 0 & 0 & \frac{2}{3} \\ 0 & 0 & 0 & 1 \\ \frac{4}{3} & \frac{2}{3} & 1 & 0 \end{bmatrix}.$$

This amounts to the basic machinery of the algorithm to compute BLOSUMs.

Remark 2 *As mentioned in the introduction, this family of matrices has become a de-facto standard for aligning amino acid sequences - however, not so much for DNA. BLASTn, i.e., BLAST for nucleotide bases, currently uses a matrix where matches are rewarded with +2 and mismatches are penalized with -3 [1] very assumed generality that has motivated this paper.*

Remark 3 *Even for amino acid sequences, the reader might notice a certain bias within the original BLOSUMs — they only consider ungapped regions and are generalized to all domains. We thus recommend using a specialized BLOSUM for a given domain.*

Remark 4 *Interestingly enough, there have been mistakes in the original computation of the BLOSUMs, which are claimed to have been improving them quietly [20]. However, this might be attributed to the fact that the benchmarks today might be influenced from the BLOSUMs of the past.*

Remark 5 *The reader might notice that there is some form of circular reasoning going on around the computation of the BLOSUMs - good alignments are produced by using BLOSUMs and BLOSUMs are produced by good alignments. Indeed, the BLOCKS database itself is created with BLOSUMs, and the two iteratively create the other.*

First, the BLOCKS database is created from a scoring matrix with 0 mismatch score and 1 match score. Then, the first BLOSUM is created, which is then used to produce new BLOCKS. This process is then repeated an additional two times until the actual BLOSUM and BLOCKS database have been produced.

1.2 Related work

Most notably, the two families of classical scoring matrices are the PAM matrices [5] and the BLOSUMs. However, and this is also true for most other research around sequence alignment: there is a strong emphasis on studying the alignment of proteins over studying the alignment of DNA. To the best of our knowledge, nearly all benchmarks concern amino acid alignments [7, 19, 22, 23], which we suspect to to be one of the reasons that there is limited research focusing on DNA alignment. There have been attempts to create gold standards for RNA alignment [24], however, even there, investigations suggest an overrepresentation of tRNA, thus leading to a suboptimal benchmark [17]

The literature for DNA scoring matrices derived from data is sparse. Hamada et al. [8] derive scoring matrices from specific organisms sequenced by specific sequencers. Further, as they claim, different genes in different organisms have significantly differing rates of mutation, which is why general-purpose DNA scoring matrices might be a bad idea. Their main focus lies on recovering and mitigating the sequencing errors and projecting the differing GC contents of different species in the resulting matrix. They evaluate their data on simulated data, which consequentually makes strong assumptions. Thus, their findings are not too applicable to real world data. Moreover, their method performs only slightly better than 2 manually created scoring matrices, which we deem to be insufficient evidence to support any substantial claims.

2 Experiments and results

2.1 Method

We constructed a fully automated pipeline which takes as input an identification code of a protein family on InterPro and the similarity threshold x and computes the corresponding nBLOSUMx (nucleotide BLOSUMx). The source code can be found online, yet, here we give some details on the implementation and issues we faced.

First, we search the NCBI protein database for the given identification code and given taxon using eDirect using the query `<TAXON> [ORGN] AND <INTERPRO_CODE>`. The following protein families were selected from InterPro:

- IPR011014
- IPR002547
- IPR023584

This yields only a subset of the desired genes, since not every protein in the database is annotated with all protein family codes and its taxa. This method is, however, to the best of our knowledge, the fastest and most reliable way to download genes of a given protein family on InterPro — for the scope of this work, we deem this sufficient.

Next, the downloaded genes are preprocessed. Genes which contain letters besides `A, C, G, T` are removed. Afterwards, they are being sorted into bins, depending on their length, the rationale thereof being that most multiple sequence alignmeners assume that the sequences are of similar length. Further, bins consisting of too few sequences are being removed.

Unfortunately, the PROTOMAT [11] found online any longer. Thus, we create our own BLOCKS, which is definitely a major drawback and flaw of this work.

To do so, each bin is getting aligned by kalign [15] with its default settings. There was no specific rationale in picking kalign - it is user friendly, fast and reasonably accurate. Aligning the bins results in a multiple sequence alignment for each bin.

Thereafter, each MSA is looked at and gapless regions are selected. These will be the BLOCKS for the computation of the BLOSUMx. Subsequently, we compute the BLOSUM90s of the created BLOCKS - note that x was chosen somewhat arbitrarily, with the only reason that lower values would cluster too many sequences for narrower taxa. Also note that we do not iteratively refine the BLOCKS as noted in Remark 5 as we were unable to find a multiple sequence aligner that allows an easy specification of the scoring matrix.

2.2 Results & Discussion

Multiple nBLOSUM90s have been derived, and the main takeaway is that the they differ substantially from the +2/-3 matrix employed by BLASTn.

In the beginning, no filtering by taxa has been undertaken, leading to what was essentially the identity matrix, i.e., rewarding matches with +1 and not penalizing mismatches. We suspect this behavior due to the vast diversity of organisms within each protein family - said diversity implies large evolutionary distance between sequences of BLOCKS, leading to what is essentially noise within the multiple sequence alignments. It would be nearly impossible to align full protein families.

Constructing the BLOSUMs for specific taxa, however, lead to more refined scoring matrices which show some interesting patterns.

First and foremost, the different mismatch scores were rarely uniform, indicating that a general mismatch score might be suboptimal for aligning sequences of known taxa. Further, we noticed that match scores were typically similar, but not equal. This might be attributable to noise in the data, as often the diagonals were indeed repeating the same score.

Another observation to be made is that, transitions, i.e., mismatches of `A` $\iff$ `G` and `T` $\iff$ `C`, ended up receiving a smaller penalty than transversions, i.e., all other mismatches. There are models of evolution making this assumption [14], which is called the transition bias phenomenon, and finding this within derived scoring matrices is a supporting argument thereof.

Further, another observation is that the more diverse the examined taxa, the lower the resulting scores. As can be seen below, the matrices of animals and bacteria have the lowest standard deviation and norm (indicating low values in the matrix) — this does, however, not hold for plants, which are also largely diverse.

As a general theme, we would like to stress that the matter of large scale genetic data is very complex, and given the lack of a gold standard, our still reasonably small understanding of genetics and the scope of this work no substantial statements can or should be made.

Taxon	Code	# Seq.	$\sigma^2(\mathbf{S})$	$\|\mathbf{S}\|_2$
bacteria	IPR011014	3648	4.24	1.05
animals	IPR023584	201	4.58	1.14
bacteria	IPR002547	5043	6.32	1.50
bacteria	IPR023584	1453	7.00	1.69
plants	IPR023584	120	7.14	1.69
insects	IPR023584	42	7.14	1.73
animals	IPR002547	552	8.77	2.08
plants	IPR011014	64	9.59	2.18
insects	IPR002547	78	10.54	2.41
plants	IPR002547	246	11.27	2.49
reptiles	IPR002547	7	17.64	3.67
mammals	IPR002547	35	22.63	4.44

Table 1: Descriptive statistics about the derived nBLOSUM90s. "# Seq." is the number of sequences used for constructing each matrix, $\sigma^2(S)$ is the standard deviation of all 16 scores per matrix and $\|\mathbf{S}\|_2$ is the Frobenius norm.

			A	C	G	T
animals	IPR002547	A	2	-2	-1	-2
		C	-2	3	-2	-1
		G	-1	-2	3	-3
		T	-2	-1	-3	3
	IPR023584	A	1	-1	0	-1
		C	-1	2	-1	0
		G	0	-1	2	-1
		T	-1	0	-1	2
bacteria	IPR002547	A	2	-1	-1	-2
		C	-1	2	-1	-1
		G	-1	-1	2	-2
		T	-2	-1	-2	2
	IPR011014	A	2	-1	0	-1
		C	-1	1	-1	0
		G	0	-1	1	-1
		T	-1	0	-1	2
	IPR023584	A	2	-1	-1	-2
		C	-1	2	-2	0
		G	-1	-2	2	-2
		T	-2	0	-2	3
insects	IPR002547	A	2	-2	-1	-3
		C	-2	3	-3	-1
		G	-1	-3	3	-4
		T	-3	-1	-4	3
	IPR023584	A	1	-2	-1	-2
		C	-2	3	-1	0
		G	-1	-1	3	-2
		T	-2	0	-2	2
mammals	IPR002547	A	3	-7	-4	-8
		C	-7	4	-7	-2
		G	-4	-7	3	-7
		T	-8	-2	-7	4
plants	IPR002547	A	2	-3	-2	-3
		C	-3	3	-3	-1
		G	-2	-3	3	-4
		T	-3	-1	-4	3
	IPR011014	A	2	-2	-1	-3
		C	-2	3	-3	-1
		G	-1	-3	3	-3
		T	-3	-1	-3	2
	IPR023584	A	2	-1	-1	-2
		C	-1	3	-2	-1
		G	-1	-2	2	-2
		T	-2	-1	-2	2
reptiles	IPR002547	A	3	-5	-2	-7
		C	-5	3	-4	-2
		G	-2	-4	3	-6
		T	-7	-2	-6	4

Table 2: The derived nBLOSUM90s, per taxon and protein family

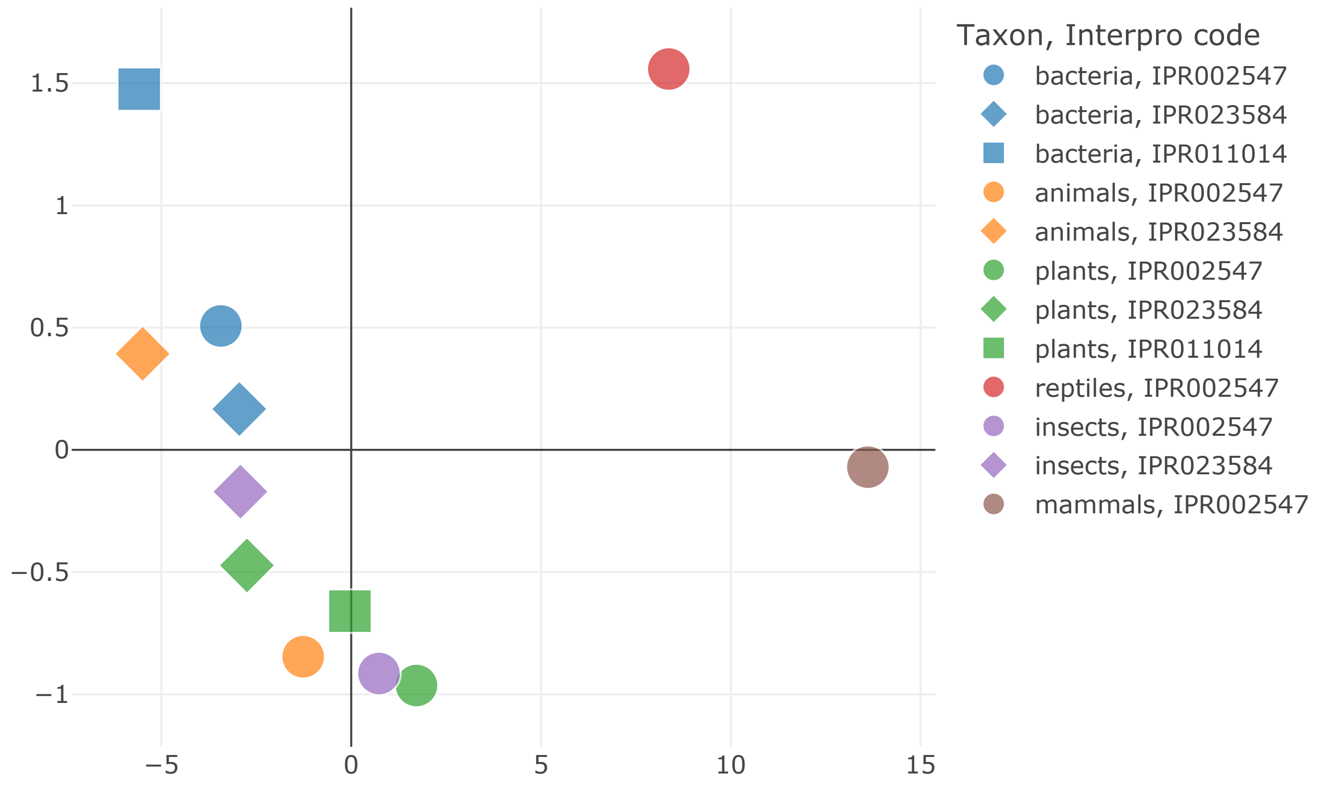

Figure 1: Various nBLOSUMs derived from the specific protein families for different subsets of species, projected to two dimensions using Principal Component Analysis [16] plotted with plotly [12] and scikit-learn [18]. The species is indicated by color and protein family by symbol.

Visually inspecting the downprojection of the BLOSUMs , we can also find hints for the fact that it is rather the species that should influence scoring matrices, rather than specific protein families. Note that the derived matrices for different protein families for a given taxa lie close together in the plot, whereas the matrices of some protein families spread rather far apart. This can also be seen inspecting the tables themselves.

3 Conclusion

We conclude with a few insights.

1. First and foremost, that the most needed advance in the field may be a reproducible and generally agreed upon consensus of what makes a good alignment for DNA, and a resulting benchmark dataset.
2. Further, whilst no quantitative superiority of derived scoring matrices for DNA has been established, observed differences between the currently employed matrix by BLASTn and the derived matrices (which do have some underlying theory as opposed to the empirically chosen match-/mismatch-scores used by BLASTn) give reason to further investigate data driven scoring matrices.
3. Moreover, the online user interface of BLAST should allow for user defined scoring matrices for, first to raise awarness of the arbitrariness of the current options and secondly to allow for further development of new nucleotide substitution matrices. We say this due to the findings of this work and 3.

References

[1] S. F. Altschul et al. "Basic local alignment search tool". In: *Journal of Molecular Biology* 215.3 (Oct. 5, 1990), pp. 403–410. ISSN: 0022-2836. DOI: 10.1016/S0022-2836(05)80360-2.

[2] *biopython/README.rst at d416809344f1e345fbabbdaca4dd6dcf441e53bd · biopython/biopython.* GitHub. URL: https://github.com/biopython/biopython/blob/d416809344f1e345fbabbdaca4dd6dcf441e53bd/README.rst (visited on 08/27/2023).

[3] Maria Chatzou et al. "Multiple sequence alignment modeling: methods and applications". In: *Briefings in Bioinformatics* 17.6 (Nov. 1, 2016), pp. 1009–1023. ISSN: 1467-5463. DOI: 10.1093/bib/bbv099. URL: https://doi.org/10.1093/bib/bbv099 (visited on 08/26/2023).

[4] Peter J. A. Cock et al. "Biopython: freely available Python tools for computational molecular biology and bioinformatics". In: *Bioinformatics* 25.11 (Mar. 2009), pp. 1422–1423. ISSN: 1367-4803. DOI: 10.1093/bioinformatics/btp163. eprint: https://academic.oup.com/bioinformatics/article-pdf/25/11/1422/48989335/bioinformatics_25_11_1422.pdf. URL: https://doi.org/10.1093/bioinformatics/btp163.

[5] M.O. Dayhoff, R.M. Schwartz, and B.C. Orcutt. "A Model of Evolutionary Change in Proteins". In: *A Model of Evolutionary Change in Proteins* 4 (Jan. 1, 1978), pp. 345–352.

[6] Sean R Eddy. "Where did the BLOSUM62 alignment score matrix come from?" In: *Nature Biotechnology* 22.8 (Aug. 2004), pp. 1035–1036. ISSN: 1087-0156, 1546-1696. DOI: 10.1038/nbt0804-1035. URL: https://www.nature.com/articles/nbt0804-1035 (visited on 08/05/2023).

[7] Paul P. Gardner, Andreas Wilm, and Stefan Washietl. "A benchmark of multiple sequence alignment programs upon structural RNAs". In: *Nucleic Acids Research* 33.8 (2005), pp. 2433–2439. ISSN: 1362-4962. DOI: 10.1093/nar/gki541.

[8] Michiaki Hamada et al. "Training alignment parameters for arbitrary sequencers with LAST-TRAIN". In: *Bioinformatics* 33.6 (Mar. 15, 2017), pp. 926–928. ISSN: 1367-4803. DOI: 10.1093/bioinformatics/btw742. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5351549/ (visited on 07/14/2023).

[9] Charles R. Harris et al. "Array programming with NumPy". In: *Nature* 585.7825 (Sept. 2020), pp. 357–362. DOI: 10.1038/s41586-020-2649-2. URL: https://doi.org/10.1038/s41586-020-2649-2.

[10] S Henikoff and J G Henikoff. "Amino acid substitution matrices from protein blocks." In: *Proceedings of the National Academy of Sciences* 89.22 (Nov. 15, 1992). Publisher: Proceedings of the National Academy of Sciences, pp. 10915–10919. DOI: 10.1073/pnas.89.22.10915. URL: https://www.pnas.org/doi/abs/10.1073/pnas.89.22.10915 (visited on 08/05/2023).

[11] S Henikoff and J G Henikoff. "Automated assembly of protein blocks for database searching." In: *Nucleic Acids Research* 19.23 (Dec. 11, 1991), pp. 6565–6572. ISSN: 0305-1048. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC329220/ (visited on 08/23/2023).

[12] Plotly Technologies Inc. *Collaborative data science.* 2015. URL: https://plot.ly.

[13] John Jumper et al. "Highly accurate protein structure prediction with AlphaFold". In: *Nature* 596.7873 (Aug. 2021). Number: 7873 Publisher: Nature Publishing Group, pp. 583–589. ISSN: 1476-4687. DOI: 10.1038/s41586-021-03819-2. URL: https://www.nature.com/articles/s41586-021-03819-2 (visited on 08/26/2023).

[14] M. Kimura. "Estimation of evolutionary distances between homologous nucleotide sequences". In: *Proceedings of the National Academy of Sciences of the United States of America* 78.1 (Jan. 1981), pp. 454–458. ISSN: 0027-8424. DOI: 10.1073/pnas.78.1.454.

[15] Timo Lassmann and Erik LL Sonnhammer. "Kalign – an accurate and fast multiple sequence alignment algorithm". In: *BMC Bioinformatics* 6.1 (Dec. 12, 2005), p. 298. ISSN: 1471-2105. DOI: 10.1186/1471-2105-6-298. URL: https://doi.org/10.1186/1471-2105-6-298 (visited on 08/25/2023).

[16] Jake Lever, Martin Krzywinski, and Naomi Altman. "Principal component analysis". In: *Nature Methods* 14.7 (July 1, 2017). Number: 7 Publisher: Nature Publishing Group, pp. 641–642. ISSN: 1548-7105. DOI: 10.1038/nmeth.4346. URL: https://www.nature.com/articles/nmeth.4346 (visited on 08/27/2023).

[17] Benedikt Löwes et al. "The BRaliBase dent—a tale of benchmark design and interpretation". In: *Briefings in Bioinformatics* 18.2 (Mar. 2017), pp. 306–311. ISSN: 1467-5463. DOI: 10.1093/bib/bbw022. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5444242/ (visited on 08/27/2023).

[18] F. Pedregosa et al. "Scikit-learn: Machine Learning in Python". In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.

[19] Muhammad Tariq Pervez et al. "Evaluating the Accuracy and Efficiency of Multiple Sequence Alignment Methods". In: *Evolutionary Bioinformatics Online* 10 (Dec. 7, 2014), pp. 205–217. ISSN: 1176-9343. DOI: 10.4137/EBO.S19199. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4267518/ (visited on 07/13/2023).

[20] Mark P. Styczynski et al. "BLOSUM62 miscalculations improve search performance". In: *Nature Biotechnology* 26.3 (Mar. 2008). Number: 3 Publisher: Nature Publishing Group, pp. 274–275. ISSN: 1546-1696. DOI: 10.1038/nbt0308-274. URL: https://www.nature.com/articles/nbt0308-274 (visited on 08/27/2023).

[21] The pandas development team. *pandas-dev/pandas: Pandas*. Version latest. Feb. 2020. DOI: 10.5281/zenodo.3509134. URL: https://doi.org/10.5281/zenodo.3509134.

[22] Julie D. Thompson et al. "BAliBASE 3.0: latest developments of the multiple sequence alignment benchmark". In: *Proteins* 61.1 (Oct. 1, 2005), pp. 127–136. ISSN: 1097-0134. DOI: 10.1002/prot.20527.

[23] Ivo Van Walle, Ignace Lasters, and Lode Wyns. "SABmark—a benchmark for sequence alignment that covers the entire known fold space". In: *Bioinformatics* 21.7 (Apr. 1, 2005), pp. 1267–1268. ISSN: 1367-4803. DOI: 10.1093/bioinformatics/bth493. URL: https://doi.org/10.1093/bioinformatics/bth493 (visited on 08/27/2023).

[24] Andreas Wilm, Indra Mainz, and Gerhard Steger. "An enhanced RNA alignment benchmark for sequence alignment programs". In: *Algorithms for Molecular Biology* 1 (Oct. 24, 2006), p. 19. ISSN: 1748-7188. DOI: 10.1186/1748-7188-1-19. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC1635699/ (visited on 08/27/2023).