APPLICATION NOTE

NEFFy: A Versatile Tool for Computing the Number of Effective Sequences

Maryam Haghani,[1,*] Debswapna Bhattacharya[1] and T. M. Murali[1]

[1]Department of Computer Science, Virginia Tech, Blacksburg, 24061, VA, United States of America
[*]Corresponding author. haghani@vt.edu

Abstract

Motivation: A Multiple Sequence Alignment (MSA) contains fundamental evolutionary information that is useful in the prediction of structure and function of proteins and nucleic acids. The "Number of Effective Sequences" (NEFF) quantifies the diversity of sequences of an MSA. While several tools embed NEFF calculation with various options, none are standalone tools for this purpose, and they do not offer all the available options.
Results: We developed NEFFy, the first software package to integrate all these options and calculate NEFF across diverse MSA formats for proteins, RNAs, and DNAs. It surpasses existing tools in functionality without compromising computational efficiency and scalability. NEFFy also offers per-residue NEFF calculation and supports NEFF computation for MSAs of multimeric proteins, with the capability to be extended to DNAs and RNAs.
Availability and Implementation: NEFFy is released as open-source software under the GNU Public License v3.0. The source code in C++ and a Python wrapper are available at `https://github.com/Maryam-Haghani/NEFFy`. To ensure users can fully leverage these capabilities, comprehensive documentation and examples are provided at `https://Maryam-Haghani.github.io/NEFFy`.
Contact: Please contact the corresponding author at haghani@vt.edu.
Supplementary Information: Supplementary data are available at *Bioinformatics* online.

Key words: Multiple Sequence Alignment (MSA), Number of Effective Sequences (NEFF), Sequence diversity

Introduction

A Multiple Sequence Alignment (MSA) organizes a set of similar sequences by introducing gaps to ensure that all sequences are of the same length, l. The MSA contains each sequence in a row with l columns or positions, where each residue or gap occupies a distinct position. Computing an MSA involves maximizing the similarity in each position across the rows while minimizing the number of gaps. By uncovering evolutionarily conserved sequence patterns, regions of similarity that cannot be identified from a single sequence alone, MSAs are used in applications including contact map prediction (He et al., 2017; Wang et al., 2017; Adhikari et al., 2018; Liu et al., 2021), RNA and protein structure prediction (Jumper et al., 2021; Baek et al., 2021; Wang et al., 2023; Baek et al., 2024; Zheng et al., 2024), and protein function annotations (Zhang et al., 2017; Hu et al., 2022; Shao et al., 2024).

Classical MSA generation involves aligning a set of user-provided homologous sequences (Edgar, 2004; Notredame et al., 2000; Chenna et al., 2003). Recent applications construct an MSA starting from a single query sequence by searching large databases with iterative methods such as PSI-BLAST (Altschul et al., 1997) or HHblits (Remmert et al., 2012) to retrieve similar sequences and align them with the query. Recent advancements in DNA/RNA sequencing technology have expanded public databases, enabling the generation of MSAs with high sequence diversity (Wilke et al., 2016; Zhang et al., 2020). Such MSAs are generally believed to provide richer evolutionary and coevolutionary insights, and they can thereby enhance the effectiveness of models utilizing them for downstream tasks (Zheng et al., 2024). However, since MSAs can contain redundant sequences, the number of sequences by itself may not be an accurate reflection of their diversity. The concept of "Number of Effective Sequences", NEFF, addresses this redundancy and assesses the quality of an MSA. Higher NEFF values often indicate a more diverse and informative MSA, leading to enhanced accuracy in predicting contact maps and the tertiary structures of proteins or RNA molecules (Pearce and Zhang, 2021; Wu et al., 2019). For example, the accuracy of AlphaFold declines substantially when the NEFF value is below approximately 30 (Jumper et al., 2021). Additionally, for RNA structure prediction models such as trRosettaRNA, which utilize MSAs of RNAs as their input, prediction

Tool	Language	MSA Format							Similarity		Query Gaps	Gappy Position	Multi Alphabet	Non-Standard Residues	Validation
		a2m	a3m	sto	aln	fasta	clustal	pfam	sym.	asym.					
RaptorX (Källberg et al., 2012)	Python	✓	✓	✗	✗	✗	✗	✗	✗	✓	✗	✗	✗	✗	✗
Conkit (Simkovic et al., 2017)	Cython	✓	✓	✓	✗	✓	✓	✗	✓	✗	✗*	✗	✗	✗	✗
DeepMSA (Zhang et al., 2020)	C++	✗	✗	✗	✓	✗	✗	✗	✓	✓	✗	✗	✗	✗	✗
Gremlin (Kamisetty et al., 2013)	C++	✗	✗	✗	✓	✓	✗	✗	✓	✗	✗	✓	✓	✗	✗
rMSA (Zhang et al., 2023)	C++	✓	✗	✗	✓	✓	✗	✗	✓	✗	✗	✗	✗	✗	✗
NEFFy	C++ & Python	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

Table 1. An overview of NEFFy and other tools that incorporate NEFF calculation, highlighting their respective features (refer to Supplementary Section S2.1 for detailed information on each feature). **Language**: The programming language used to implement each tool. **MSA Format**: Formats used to represent aligned sequences in an MSA. **Similarity:** Methods for determining sequence similarity between pairs of sequences within the MSA. **Query Gaps**: Handling query gaps—defined as "gaps aligned to insertions"—can be customized during NEFF calculation based on user preference: either by removing them along with the corresponding positions in the aligned sequences or by keeping them intact. *`Conkit` can manage these gaps for the `a3m` format by offering two distinct options: `a3m-inserts` and `a3m`. **Gappy position:** Inspired by `Gremlin`, this option addresses positions with a gap frequency exceeding the desired gap threshold. **Alphabet**: Alphabet used to represent a biological sequence. Of the tools mentioned, `RaptorX` and `Conkit` do not explicitly define an alphabet, while `DeepMSA` and `rMSA` use the protein alphabet. In contrast, `Gremlin` supports both protein and RNA sequences (implicitly including DNA). NEFFy accommodates proteins, RNAs, and DNAs. **Non-standard Residues**: Residues outside the standard set of biological sequence residues are handled differently across tools. `RaptorX` and `Conkit` do not support these residues, while `rMSA` and `Gremlin` treat them as gaps. `DeepMSA` considers them as standard symbols in its symmetric version but treats them as gaps when calculating similarity cutoff in its asymmetric version. NEFFy offers users the flexibility to customize the handling of these residues according to their preference. **Validation**: Indicates if validation is provided for the MSA file before the NEFF calculation.

accuracy is correlated with NEFF (Wang et al., 2023), and for high-quality MSAs, these models can outperform other methods (Tarafder et al., 2024).

We introduce NEFFy, a fast and dedicated standalone tool for NEFF calculation. NEFFy is uniquely equipped to parse MSAs and calculate NEFF across a wide range of MSA formats for protein and nucleic acid sequences. It integrates all the features from earlier NEFF tools (see Table 1) and offers a set of new functionalities. NEFFy is developed in C++ for optimal performance and is also provided as a Python library that wraps the C++ executables. This approach enables seamless integration into Python-based workflows, simplifying use for a wider audience while preserving efficiency.

Method

The NEFF of an MSA M can be formulated as

$$\mathrm{NEFF}(M) = \left(\frac{1}{\sqrt{L}}\right)\frac{1}{1+\sum_{m=1,m\neq n}^{N} I[S_{m,n} \geq \theta]},$$

where L represents the length of the query sequence, N denotes the number of sequences in M, $S_{m,n}$ indicates the amount of sequence similarity between m-th and n-th sequences, θ is the similarity threshold (typically set to 0.8), and $I[]$ is the Iverson bracket, meaning that $I[S_{m,n} \geq \theta]$ equals 1 if $S_{m,n} \geq \theta$, and 0 otherwise.

This formula defines NEFF as the normalized sum of the weights of the sequences in the MSA. Specifically, each sequence i is assigned a weight of $\frac{1}{n_i}$, where n_i is the number of sequences (including itself) similar to it. This approach for calculating NEFF is widely employed in many contact prediction and MSA generation methods (Liu et al., 2021; Zhang et al., 2020; Wu et al., 2019; Li et al., 2021). Note that $\frac{1}{\sqrt{L}}$ is a normalization factor, which is a commonly used approach (Liu et al., 2021; Zhang et al., 2020; Li et al., 2021), although NEFFy offers other normalization options. We discuss alternative formulations of NEFF in Supplementary Section S5. In this work, we adopt the specified NEFF formula because of its broad acceptance in the structure prediction community and its proven ability to capture the evolutionary diversity present in MSAs.

Aligning with the given formulation, several tools incorporate NEFF calculation as part of their functionality (Zhang et al., 2020; Källberg et al., 2012; Simkovic et al., 2017; Kamisetty et al., 2013; Zhang et al., 2023). Additional

details about each are provided in Supplementary Section S3 of the supplementary file. However, none of these tools are exclusively designed for NEFF calculation and they typically offer NEFF as required for their primary tasks. Consequently, they do not offer the full range of NEFF-related capabilities. Table 1 compares these tools with NEFFy in terms of the features they provide. It highlights that NEFFy is the only solution compatible with a wide range of MSA formats and supports all the features provided by the other tools. Moreover, NEFFy introduces several new features:

1. **NEFF calculation for multiple MSAs**: Accepts multiple MSA input files, combines their sequences in the specified order, removes duplicates, and calculates the NEFF for the resulting merged MSA.
2. **Per-residue (column-wise) NEFF calculation**: Calculates NEFF for each position in the MSA by summing the weights of sequences containing a residue (i.e., non-gap character) at that position.
3. **NEFF for multimeric MSAs**: Computes NEFF for multimeric MSAs, which comprising multiple chains. NEFFy can process both homomers and heteromers. The user can specify the multimer format to enable NEFFy to automatically identify the appropriate blocks within the MSA and calculate NEFF values for each.
4. **MSA format conversion**: Converts MSA formats while preserving sequence integrity and annotations, with no need for user intervention.
5. **MSA validation**: Ensures that the input follows the specified format and contains only residues from the permitted character set for the given alphabet.

Supplementary Section S2 details each feature.

Results

We conducted an experiment using the CASP15 dataset, which consists of 93 targets, to compare the reliability and efficiency of NEFFy with other tools. To generate MSA files for these targets, we ran AlphaFold 2.3 locally, utilizing its default MSA generation pipeline. This process resulted in three MSA files per target, sourced from the Uniref90, Mgnify, and BFD datasets. In total, we generated 279 MSA files in `STO` and `A3M` formats. Given that certain tools do not support NEFF calculation in these formats, we employed NEFFy's built-in converter to convert the files into formats compatible with each tool. We then calculated the NEFF value for each of these MSAs. To do this, we ran NEFFy using the options specified by each tool and also used each tool directly to calculate NEFF (Supplementary Section S4.1). The results showed that NEFFy consistently produced NEFF values identical to those of `DeepMSA` and highly similar to those from other tools (Supplementary Figures S2-6).

To assess the computational efficiency of NEFFy relative to other tools, we used the MSA files from the previous analysis and recorded the execution time for each tool (Supplementary Section S4.2). NEFFy achieves the same efficiency as `DeepMSA` and `Gremlin` and significantly outperforms other tools. This makes it particularly ideal for processing deep MSAs of long query sequences (Supplementary Figure S7).

To evaluate the scalability of NEFFy with respect to MSA depth, we conducted an analysis by progressively increasing the MSA depth and measuring the corresponding execution times (Supplementary Section S4.3). The results show that NEFFy, along with `DeepMSA` and `Gremlin`, exhibits relatively constant execution times across all depths, demonstrating superior scalability and minimal sensitivity to increasing MSA depth. This indicates these methods are well-suited for efficiently handling deeper MSAs. In contrast, `RaptorX` and `Conkit` show a substantial increase in execution time as the depth increases, indicating that their performance is more sensitive to larger MSA depths and may struggle with higher computational loads at higher depths (Supplementary Figure S8). These findings provide valuable insights for selecting appropriate tools for MSA-based workflows depending on the computational resources available and the required scalability.

As a case study on multi-domain proteins, we explored the relationship between the NEFF values of individual domains and those of entire protein chains. Using 19 multi-domain proteins from the CASP15 dataset, we calculated NEFF values for the MSA of each domain and the entire target, using two separate sets of MSAs—one generated by AlphaFold (Jumper et al., 2021) and the other by RoseTTAFold (Baek et al., 2021) (Supplementary Section S4.4). To compare the relative NEFF values, we generated Grishin plots (Kinch et al., 2011), which display the correlation between the weighted sum of NEFF values for individual domains (y-axis) and the NEFF values for the full chain (x-axis). Our results showed that individual domains tend to have higher NEFF values than the complete protein chains in both the AlphaFold and RoseTTAFold-generated MSAs (Supplementary Figures S9 and S10).

In a recent study, Moussad et al. (Moussad et al., 2023) examined the prediction accuracies of individual domains and their overall packing in tertiary structures generated by AlphaFold and RoseTTAFold models for these 19 multi-domain targets using the Global Distance Test (GDT-TS) metric designed to compare the similarity between predicted and reference structures (Zemla, 2003). They found that individual domains were predicted with high accuracy, while the overall packing of these domains was less accurate. Our NEFF analysis for the MSAs used as input for these multi-domain targets reflects similar patterns, suggesting that the higher NEFF values observed for MSAs of individual domains may contribute to the improved accuracy of their predicted structures.

Conclusion

We present NEFFy, a fast and comprehensive tool for calculating the "Number of Effective Sequences" (NEFF) for a Multiple Sequence Alignment (MSA). By merging established features with innovative functionalities, NEFFy delivers an efficient and versatile solution for NEFF calculation, enhanced by its robust capabilities to convert and validate a wide range of MSA formats. Its implementation as a Python library further increases its accessibility and ease of use. Our experiments demonstrated that NEFFy is highly consistent with existing tools and is efficient, making it a valuable addition to the bioinformatics

toolkit for processing MSAs of proteins and nucleic acids. As foundational models continue to evolve (Sumi et al., 2024) and generate synthetic biological sequences, NEFFy can also be used to evaluate the effectiveness of evolutionary sequence search and alignment algorithms for proteins and nucleic acids (DNA and RNA) in comparison to synthetic sequences generated by these models. For future work, we will integrate pairwise NEFF calculations into NEFFy for heteromeric MSAs, enabling nuanced analysis of cases where some monomer pairs are accurately predicted while others are not.

Data Availability

The MSA files used for NEFF analysis of reliability and scalability are hosted on Zenodo at `https://doi.org/10.5281/zenodo.14210949`, while those for multi-domain analysis can be found at `https://doi.org/10.5281/zenodo.7682977`. The source code is archived at Zenodo: `https://doi.org/10.5281/zenodo.14908219`.

Acknowledgements

This work was partially supported by awards from the National Institute of General Medical Sciences (R35GM138146 to D.B.) and the National Science Foundation (DBI 2208679 to D.B. and CCF 2200045 to T.M.M.).

References

Adhikari, B., Hou, J., and Cheng, J. (2018). DNCON2: improved protein contact prediction using two-level deep convolutional neural networks. *Bioinformatics*, 34(9):1466–1472.

Altschul, S., Madden, T., Schäffer, A., Zhang, J., Zhang, Z., Miller, W., and Lipman, D. (1997). Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic acids research*, 25(17):3389–3402.

Baek, M., DiMaio, F., Anishchenko, I., Dauparas, J., Ovchinnikov, S., Lee, G., Wang, J., Cong, Q., Kinch, L., Schaeffer, R., et al. (2021). Accurate prediction of protein structures and interactions using a three-track neural network. *Science*, 373(6557):871–876.

Baek, M., McHugh, R., Anishchenko, I., Jiang, H., Baker, D., and DiMaio, F. (2024). Accurate prediction of protein–nucleic acid complexes using RoseTTAFoldNA. *Nature Methods*, 21(1):117–121.

Chenna, R., Sugawara, H., Koike, T., Lopez, R., Gibson, T. J., Higgins, D., and Thompson, J. (2003). Multiple sequence alignment with the Clustal series of programs. *Nucleic Acids Res*, 31(13):3497–3500.

Edgar, R. (2004). MUSCLE: multiple sequence alignment with high accuracy and high throughput. *Nucleic acids research*, 32(5):1792–1797.

He, B., Mortuza, S., Wang, Y., Shen, H., and Zhang, Y. (2017). NeBcon: protein contact map prediction using neural network training coupled with naïve Bayes classifiers. *Bioinformatics*, 33(15):2296–2306.

Hu, M., Yuan, F., Yang, K., Ju, F., Su, J., Wang, H., Yang, F., and Ding, Q. (2022). Exploring evolution-aware &-free protein language models as protein function predictors. *Advances in Neural Information Processing Systems*, 35:38873–38884.

Jumper, J., Evans, R., Pritzel, A., Green, T., Figurnov, M., Ronneberger, O., Tunyasuvunakool, K., Bates, R., Žídek, A., Potapenko, A., et al. (2021). Highly accurate protein structure prediction with AlphaFold. *Nature*, 596(7873):583–589.

Källberg, M., Wang, H., Wang, S., Peng, J., Wang, Z., Lu, H., and Xu, J. (2012). Template-based protein structure modeling using the RaptorX web server. *Nature Protocols*, 7(8):1511–1522.

Kamisetty, H., Ovchinnikov, S., and Baker, D. (2013). Assessing the utility of coevolution-based residue–residue contact predictions in a sequence-and structure-rich era. *Proceedings of the National Academy of Sciences*, 110(39):15674–15679.

Kinch, L., Shi, S., Cheng, H., Cong, Q., Pei, J., Mariani, V., Schwede, T., and Grishin, N. (2011). CASP9 target classification. *Proteins*, 79 Suppl 10(Suppl 10):21–36.

Li, Y., Zhang, C., Bell, E., Zheng, W., Zhou, X., Yu, D., and Zhang, Y. (2021). Deducing high-accuracy protein contact-maps from a triplet of coevolutionary matrices through deep residual convolutional networks. *PLoS computational biology*, 17(3):e1008865.

Liu, X., Jin, L., Gao, S., and Zhao, S. (2021). Protein contact map prediction using multiple sequence alignment dropout and consistency learning for sequences with less homologs. *Preprint at bioRxiv*.

Moussad, B., Roche, R., and Bhattacharya, D. (2023). The transformative power of transformers in protein structure prediction. *PNAS*, 120(32):e2303499120.

Notredame, C., Higgins, D., and Heringa, J. (2000). T-Coffee: A novel method for fast and accurate multiple sequence alignment. *Journal of molecular biology*, 302(1):205–217.

Pearce, R. and Zhang, Y. (2021). Toward the solution of the protein structure prediction problem. *Journal of Biological Chemistry*, 297(1).

Remmert, M., Biegert, A., Hauser, A., and Söding, J. (2012). HHblits: lightning-fast iterative protein sequence searching by HMM-HMM alignment. *Nature methods*, 9(2):173–175.

Shao, J., Chen, J., and Liu, B. (2024). ProFun-SOM: Protein Function Prediction for Specific Ontology Based on Multiple Sequence Alignment Reconstruction. *IEEE Transactions on Neural Networks and Learning Systems*.

Simkovic, F., Thomas, J., and Rigden, D. (2017). ConKit: a python interface to contact predictions. *Bioinformatics*, 33(14):2209–2211.

Sumi, S., Hamada, M., and Saito, H. (2024). Deep generative design of RNA family sequences. *Nature Methods*, 21(3):435–443.

Tarafder, S., Roche, R., and Bhattacharya, D. (2024). The landscape of RNA 3D structure modeling with transformer networks. *Biology Methods and Protocols*, 9(1).

Wang, S., Sun, S., Li, Z., Zhang, R., and Xu, J. (2017). Accurate de novo prediction of protein contact map by ultra-deep learning model. *PLoS computational biology*, 13(1):e1005324.

Wang, W., Feng, C., Han, R., Wang, Z., Ye, L., Du, Z., Wei, H., Zhang, F., Peng, Z., and Yang, J. (2023). trRosettaRNA: automated prediction of RNA 3D structure

with transformer network. *Nature Communications*, 14(1):7266.

Wilke, A., Bischof, J., Gerlach, W., Glass, E., Harrison, T., Keegan, K., Paczian, T., Trimble, W., Bagchi, S., Grama, A., et al. (2016). The MG-RAST metagenomics database and portal in 2015. *Nucleic Acids Res*, 44(D1):D590–D594.

Wu, T., Hou, J., Adhikari, B., and Cheng, J. (2019). Analysis of several key factors influencing deep learning-based inter-residue contact prediction. *Bioinformatics*, 36(4):1091–1098.

Zemla, A. (2003). LGA: A method for finding 3D similarities in protein structures. *Nucleic Acids Res*, 31:3370–3374.

Zhang, C., Freddolino, P., and Zhang, Y. (2017). COFACTOR: improved protein function prediction by combining structure, sequence and protein–protein interaction information. *Nucleic Acids Res*, 45(W1):W291–W299.

Zhang, C., Zhang, Y., and Pyle, A. (2023). rMSA: A Sequence Search and Alignment Algorithm to Improve RNA Structure Modeling. *Journal of Molecular Biology*, 435(14):167904.

Zhang, C., Zheng, W., Mortuza, S., Li, Y., and Zhang, Y. (2020). DeepMSA: constructing deep multiple sequence alignment to improve contact prediction and fold-recognition for distant-homology proteins. *Bioinformatics*, 36(7):2105–2112.

Zheng, W., Wuyun, Q., Li, Y., Zhang, C., Freddolino, P. L., and Zhang, Y. (2024). Improving deep learning protein monomer and complex structure prediction using DeepMSA2 with huge metagenomics data. *Nature Methods*, 21(2):279–289.