

Analyzing at Scale the Effects of Optimal Global Sequence Alignment on Sequence Similarity Using a GPU-Optimized Implementation of the Needleman-Wunsch Algorithm and the SBERT Module [†]

Emilia Pardo ¹, Valko Milev ¹, Ventsislav Kolev ¹ and Maria Marinova ^{1,2,*} 

¹ Department Computer Systems and Technologies, Faculty of Electronics and Automation, Technical University Sofia Branch Plovdiv, Tsanko Diustabanov St., 25, 4000 Plovdiv, Bulgaria; epardo@tu-plovdiv.bg (E.P.); valko.krasimirov.milev@gmail.com (V.M.); venticolev@gmail.com (V.K.)

² Center of Competence “Smart Mechatronic, Eco-and Energy-Saving Systems and Technologies”, Technical University Sofia Branch Plovdiv, Tsanko Diustabanov St., 25, 4000 Plovdiv, Bulgaria

* Correspondence: m_marinova@tu-plovdiv.bg

[†] Presented at the 14th International Scientific Conference TechSys 2025—Engineering, Technology and Systems, Plovdiv, Bulgaria, 15–17 May 2025.

Abstract

Global sequence alignment remains a fundamental technique in bioinformatics, yet its large-scale application is often limited by computational demands. In this work, we explore the impact of optimal global sequence alignment on sequence similarity by combining a GPU-accelerated version of the Needleman–Wunsch algorithm with Sentence-BERT (SBERT), a transformer-based embedding model. This setup allows us to perform exhaustive alignments across large datasets while also capturing semantic similarities that traditional scoring schemes may miss. By comparing alignment scores with SBERT-derived similarity metrics, we examine where classical alignment aligns with—or diverges from—contextual similarity. Our results show that while there is general agreement between the two approaches, notable exceptions highlight cases where semantic meaning extends beyond structural alignment. The combination of efficient global alignment and deep semantic modeling offers new insights into the relationship between sequence form and function, opening the door for more nuanced similarity measures in both biological and textual sequence analysis.

Keywords: optimal global sequence alignment; machine learning sequence transformers; CUDA parallel programming; genome sequence similarity analysis



Academic Editors: Nikola G. Shakev, Valyo N. Nikolov, Nikolay R. Kakanakov and Sevil A. Ahmed-Shieva

Published: 4 August 2025

Citation: Pardo, E.; Milev, V.; Kolev, V.; Marinova, M. Analyzing at Scale the Effects of Optimal Global Sequence Alignment on Sequence Similarity Using a GPU-Optimized Implementation of the Needleman-Wunsch Algorithm and the SBERT Module. *Eng. Proc.* **2025**, *100*, 66. <https://doi.org/10.3390/engproc2025100066>

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Bioinformatics is an interdisciplinary field that combines computer science, mathematics, and statistics to analyze biological data. Computer science is used for the measurement and analysis of DNA, RNA, and proteins [1]. Mathematical and statistical methods, probability theory, machine learning (ML), and statistical modeling are used to analyze the significance of results [2]. These analyses are highly dependent on data storage, accessibility, and management. Another foundation of this multidisciplinary science is software algorithms and tools with diverse applications, such as sequence analysis, structure prediction, and data visualization. There are a large number of studies on genome assembly, gene

discovery, drug design, protein structure prediction, and, last but not least, protein and nucleotide sequence alignment.

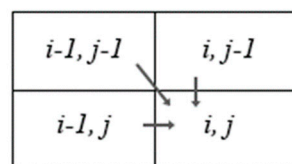
One of the most important questions to ask about a gene or protein is whether it is related to another gene or protein, such that the two share a common function. This is why sequence alignment is needed—to identify how the two are related, expressed in common regions or motifs.

DNA is a linear polymer made up of just four different bases or nucleotides. DNA and RNA comprise nucleotides that occur in different but chemically related forms. The four bases of DNA are the purines guanine (G) and adenosine (A), and the pyrimidines cytosine (C) and thymine (T). In most forms of the RNA molecule, there are also four bases, three of which are the same as in DNA, but the fourth is different—thymine in DNA is replaced by uracil (U) in RNA.

In the case of two sequences, if gaps are taken into account, a large number of alignments can be generated. We might conclude that there is an optimal alignment that is close to the ideal that could perfectly identify similarities in two sequences. In reality, however, there are many alternatives and variations that could potentially be misleading. Therefore, useful methods must produce alignments that can be compared in a meaningful way, and their degree of similarity must be estimated. Such methods are referred to as scoring methods, one of which is the Needleman–Wunsch algorithm [3]. A wavefront version of the Needleman–Wunsch algorithm has been developed using GPU acceleration in order to implement artificial intelligence.

2. Needleman–Wunsch Algorithm

The algorithm computes the optimal alignment between two sequences $S[1..n]$ and $T[1..m]$, which is the alignment with the maximum score [3,4]. It uses dynamic programming and takes exponential time $O(nm)$ [5,6]. In Figure 1 is shown main computation of the algorithm. Let us define $F(i,j)$ as the score of the optimal alignment between the two sequences S and T . Develop a recursive formula for $F(i,j)$ that depends on two cases:



$$F(i,j) = \max \begin{cases} F(i-1, j-1) + \delta(S[i], T[j]) \text{ match/mismatch} \\ F(i-1, j) + \delta(S[i], _) \text{ delete} \\ F(i, j-1) + \delta(_, T[j]) \text{ insert} \end{cases}$$

Figure 1. The three cases of the Needleman–Wunsch algorithm.

Either $i = 0$ or $j = 0$. For this case, the string is aligned with an empty string, meaning that there is an insertion or deletion.

Both $i > 0$ and $j > 0$.

For this case, we observe that in the best alignment between $S[1..i]$ and $T[1..j]$, the last pair of aligned characters should be either matched/mismatched, deleted or inserted [4,6,7]. To obtain the best possible score, we will choose the maximum of these three cases.

Following this principle, a matrix is filled. Figure 2 shows filled table F of two sample sequences, AAGC and AGT. The optimal alignment score is -1 , shown in the bottom right corner. The parameters of match, mismatch, and gap are set to 1, -1 , and -2 .

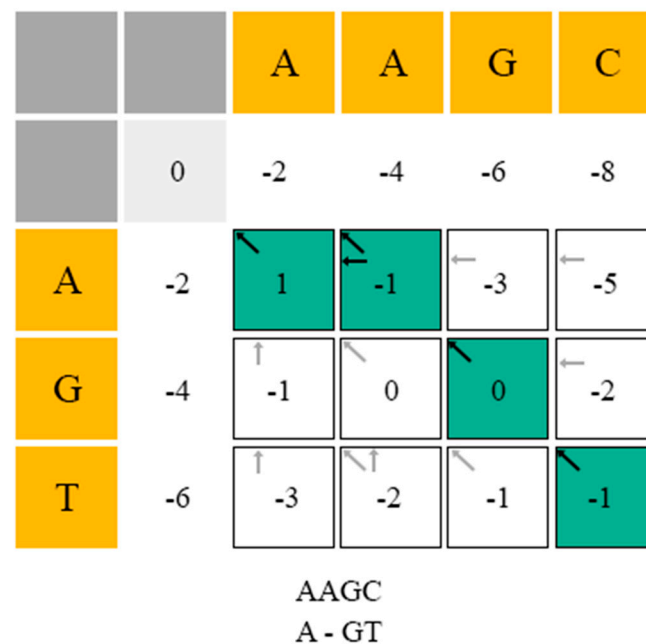


Figure 2. Needleman–Wunsch algorithm—score matrix and traceback logic.

3. Nvidia Ada GPU Architecture

The full AD102 GPU, based on NVIDIA's Ada Lovelace architecture, is engineered for high-performance computing, featuring 12 Graphics Processing Clusters (GPCs), 72 Texture Processing Clusters (TPCs), and 144 Streaming Multiprocessors (SMs). Its memory subsystem includes a 384-bit interface, comprising 12 32-bit memory controllers, which enables high memory bandwidth essential for data-intensive applications. The GPU also includes 288 FP64 cores (two per SM) for double-precision computing tasks.

Each SM in the AD10x lineup is highly integrated, containing 128 CUDA Cores for general-purpose parallel processing, one Ada third-generation RT Core for accelerated ray tracing, and four Ada fourth-generation Tensor Cores optimized for AI and matrix-heavy operations. Additionally, each SM features four Texture Units, a 256 KB register file, and 128 KB of L1/shared memory, which can be configured to optimize performance depending on workload requirements. This blend of high compute density, flexible memory architecture, and specialized cores positions the AD102 as a powerhouse for modern workloads spanning gaming, AI, and scientific computing.

The full AD102 GPU includes 18432 CUDA Cores, 144 RT Cores, 576 Tensor Cores, and 576 Texture Units. Ada's streaming multiprocessor contains a 128 KB Level 1 cache, which has a unified architecture that can be configured as either an L1 data cache or as shared memory, depending on the workload. The AD102 is equipped with a 98,304 KB L2 cache, which will impact all applications, but especially complex operations such as ray tracing.

4. GPU-Accelerated Parallel Needleman–Wunsch Algorithm

To run the algorithm in parallel on a GPU, we would have to utilize the Antidiagonal (Wavefront) approach due to the potential for race-condition-type issues that the algorithm inherently possesses [4,6,8]. As the calculation of the current matrix element requires the values of the top, left, and top-left diagonal matrix elements, the classic horizontal parsing

of the matrix would not be suitable for parallel calculation. However, if the matrix is parsed antidiagonally, each antidiagonal could be calculated in parallel as each element from the current antidiagonal has its data requirements met, given that the previous antidiagonal has been calculated. With this in mind, it would be possible to calculate each cell of the antidiagonal in parallel. The antidiagonal approach (shown in Figure 3) is specifically suitable for GPU implementation, as with the increasing length of each subsequent antidiagonal when approaching the middle of the matrix, more of the GPU's resources are being utilized and for bigger sequences, all of the thousands of CUDA cores are being utilized for the majority of the calculation period [9]. As GPUs are specialized to handle a huge amount of GFLOPS, redesigning the algorithm using the antidiagonal approach [10] is the optimal strategy for a parallel implementation on a GPU in order to maximize resource utilization.

		G	A	C	C	T	A
	0	2	4	6	8	10	12
G	2	0	2	4	6	8	10
T	4	2	1	3	5	6	8
A	6	4	2	2	4	6	6
C	8	6	4	2	2	4	6
T	10	8	6	4	3	2	4
T	12	10	8	6	5	3	3
T	14	12	10	8	7	5	4
A	16	14	12	10	9	7	5

Figure 3. Antidiagonal (wavefront) approach for calculating the score matrix.

After the developed solution has been tested successfully, it is important to use a profiling tool for in-depth diagnostics, even if the received results are expected and the reduction in execution time falls within the desired range. Nvidia provides advanced profiling tools such as Nsight Systems and Nsight Compute, which are capable of producing hundreds of GPU-specific metrics and multiple performance analysis panels, visualizations, and grids designed for both high-level and in-depth research.

Once profiling and optimization are complete, the algorithm is then scaled up to perform thousands of parallel alignments of sequences with lengths between 9000 and 10,000 characters. The results and alignment scores are fed to a pipeline and correlated with the results of the similarity evaluations produced by the machine learning sequence transformers. Based on this comparison, we conclude what the tendencies of the alignment algorithm are to impact the similarity of the sequences post-alignment.

5. SBERT-Based Vectorization and Similarity Scoring of Genes

To capture semantic similarity between gene descriptions, we employed the SBERT (Sentence-BERT) framework, a transformer-based model originally developed for natural language processing tasks [11]. Specifically, we used the sentence-transformers/all-MiniLM-L6-v2 model [12], which is among the most widely adopted SBERT variants due to its balance between performance and efficiency. This model encodes input sentences into dense 768-dimensional vectors in such a way that semantically similar inputs are mapped to nearby regions in the embedding space. This property enables effective comparison, clustering, and classification of sequence representations based on their contextual meaning rather than strict character-level similarity.

By representing each gene's descriptive sequence in this high-dimensional vector space, we can quantitatively assess the semantic closeness between any pair of gene descriptions. To compute similarity scores, we used cosine similarity, a standard metric for comparing high-dimensional vectors. Cosine similarity evaluates the cosine of the angle between two vectors, producing a score between -1 and 1 , where 1 indicates identical direction (maximum similarity), and lower values reflect increasing dissimilarity. In this context, genes that are functionally or contextually similar—regardless of exact wording—will yield higher cosine similarity scores.

This embedding-based approach allows us to move beyond traditional symbolic comparison and tap into latent semantic relationships encoded in natural language representations of genes. Furthermore, the encoded vectors serve as a foundation for downstream tasks such as clustering genes into functional groups or performing classification based on contextual features. The use of SBERT (Figure 4) thus provides a robust framework for modeling and quantifying gene similarity at a semantic level, complementing structural alignment approaches with a more nuanced, context-aware perspective.

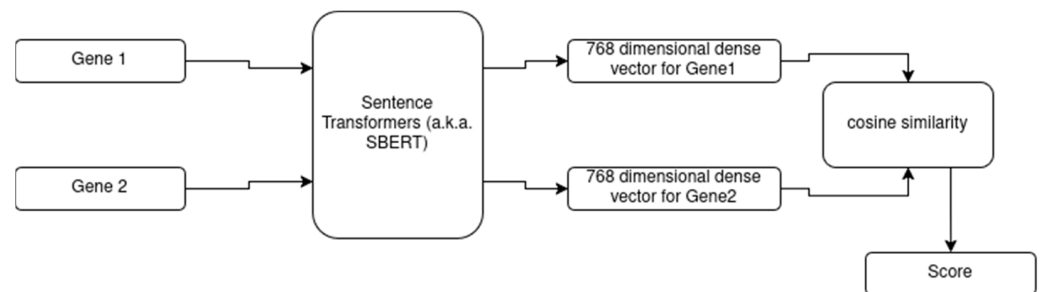


Figure 4. SBERT-based encoding of gene descriptions for semantic similarity analysis using cosine distance.

6. Results

This study assessed the impact of the Needleman–Wunsch global alignment algorithm on sequence similarity across four different sample sizes: 200, 500, 1000, and 2000 gene sequences. These samples consisted of random virus genomes, each with lengths ranging from 9000 to 10,000 characters [13]. For each sample size, we calculated the average similarity scores both prior to and following the alignment process, and subsequently computed the average increase in similarity. The results, summarized in Table 1, show a noticeable improvement in similarity after alignment for all sample sizes.

Table 1. Sample analysis results showing sequence similarity scores before and after applying the global alignment algorithm. Similarity scores were computed using the SBERT all-MiniLM-L6-v2 model to embed sequences into a semantic vector space. The alignment was performed using a CUDA-based parallel implementation of the Needleman–Wunsch algorithm, enabling efficient computation of globally aligned sequence similarities at scale.

Sample Size	Avg. Similarity Prior Alignment (%)	Avg. Similarity Post Alignment (%)	Avg. Similarity Increase (%)
200	84.44	90.90	6.46
500	84.78	89.99	5.21
1000	83.44	90.09	6.65
2000	90.83	96.30	5.47

As indicated by the data, the alignment process consistently increased the average similarity scores across all sample sizes tested. Specifically, for the smallest sample size of 200 sequences, the average similarity increased from 84.44% prior to alignment to 90.90%

post-alignment, resulting in an average increase of 6.46%. For the 500-sequence sample, the similarity scores rose from 84.78% to 89.99%, showing a smaller increase of 5.21%. For the 1000-sequence sample, the alignment resulted in an increase of 6.65%, with the average similarity going from 83.44% to 90.09%. The 2000-sequence sample showed the largest raw similarity scores both pre- and post-alignment, with an average similarity of 90.83% prior to alignment, which increased to 96.30%, resulting in a 5.47% improvement.

7. Conclusions

These results suggest that the alignment process led to a consistent improvement in similarity, regardless of sample size. Larger sample sizes generally yielded higher similarity scores both before and after alignment, likely due to the more extensive and diverse set of sequences representing the virus and bacteria genomes. However, the percentage increase in similarity was slightly more pronounced in smaller datasets. Specifically, the 200-sequence sample exhibited the largest relative increase (6.46%), while the 2000-sequence sample showed the smallest relative improvement (5.47%), despite starting with the highest pre-alignment similarity score.

These observations indicate that while the Needleman–Wunsch algorithm effectively improves similarity scores for all sample sizes, the magnitude of improvement appears to be more significant for smaller datasets. Larger sample sizes, on the other hand, might have already captured a significant amount of similarity due to their more extensive representation of the genomes. Nonetheless, even with the largest dataset, the algorithm provided substantial improvement in sequence similarity, confirming its robustness and efficiency in large-scale genomic comparisons.

In summary, this research presents the robustness of the existing algorithm combined with an innovative, optimized version of NW through parallelization to better cope with computational constraints in big data. Overall, this work contributes to the field of bioinformatics and computational sequence analysis in several key ways. First, it illustrates a scalable method for global alignment that makes exhaustive comparisons across large datasets computationally feasible. Second, it introduces a hybrid approach that bridges structural alignment and contextual embedding, enriching our understanding of sequence similarity from both biological and semantic perspectives. Finally, by revealing areas of divergence between alignment-based and embedding-based similarity, this study opens up new avenues for developing more nuanced and functionally relevant similarity measures in genomics and natural language processing alike.

Author Contributions: Conceptualization, E.P., V.M., V.K. and M.M.; methodology, E.P., V.M. and V.K.; software, E.P., V.M. and V.K.; validation, V.K., V.M. and M.M.; writing—original draft preparation, E.P. and V.K.; writing—review and editing, E.P.; visualization, E.P. and V.K.; supervision, M.M.; funding acquisition, M.M. All authors have read and agreed to the published version of the manuscript.

Funding: The authors gratefully acknowledge the financial support provided within the Technical University of Sofia, OP “Research, Innovation and Digitalization Programme for Intelligent Transformation 2021–2027”, Project No. BG16RFPR002-1.014-0005 Center of competence “Smart Mechatronics, Eco-and Energy Saving Systems and Technologies”.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data are included in the article.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Krastev, I.; Vetova, S.; Budakova, D. Protein sequence alignment analysis using “biosequence alignments” database and bioinformatics system. *AIP Conf. Proc.* **2024**, *3078*, 040004. [[CrossRef](#)]
2. Petrova, D. Applying protein structure comparison methods for studying SARS-CoV-2 spike protein. *AIP Conf. Proc.* **2024**, *2980*, 040004. [[CrossRef](#)]
3. Pardo, E. Overview of the Modern Approach of Sequence Alignment Algorithms. *AIP Conf. Proc.* **2024**, *3274*, 040006. [[CrossRef](#)]
4. Needleman, S.; Wunsch, C. A General Method Applicable to Search for Similarities in Amino Acid Sequence of 2 Proteins. *J. Mol. Biol.* **1970**, *48*, 443–453. [[CrossRef](#)] [[PubMed](#)]
5. Feng, B.; Gao, J. Distributed Parallel Needleman-Wunsch Algorithm on Heterogeneous Cluster System. In Proceedings of the International Conference on Network and Information Systems for Computers, Wuhan, China, 23–25 January 2015; pp. 358–361. [[CrossRef](#)]
6. Sung, W.-K. *Algorithms in Bioinformatics: A Practical Introduction*; Chapman & Hall/CRC Press: Boca Raton, FL, USA, 2010; pp. 29–51.
7. Gancheva, V.; Georgiev, I. Multithreaded Parallel Sequence Alignment Based on Needleman-Wunsch Algorithm. In Proceedings of the IEEE 19th International Conference on Bioinformatics and Bioengineering (BIBE), Athens, Greece, 28–30 October 2019; pp. 165–169. [[CrossRef](#)]
8. Muhamad, F.; Ahmad, R.B.; Asi, S.; Murad, M.N. Performance Analysis of Needleman-Wunsch Algorithm (Global) and Smith-Waterman Algorithm (Local) in Reducing Search Space and Time for DNA Sequence Alignment. *J. Phys. Conf. Ser.* **2018**, *1019*, 012085. [[CrossRef](#)]
9. Kolev, V.; Marinova, M.; Pardo, E. Optimal global sequence alignments of COVID-19 nucleotide sequences using the Needleman-Wunsch algorithm and program parallelization. *AIP Conf. Proc.* **2023**, 020004. [[CrossRef](#)]
10. CUDA Toolkit. Available online: <https://developer.nvidia.com/cuda-toolkit> (accessed on 19 June 2025).
11. Aguado-Puig, Q.; Marco-Sola, S.; Moure, J.C.; Castells-Rufas, D.; Alvarez, L.; Espinosa, A.; Moreto, M. Accelerating Edit-Distance Sequence Alignment on GPU Using the Wavefront Algorithm. *IEEE Access* **2022**, *10*, 63782–63796. [[CrossRef](#)]
12. Sentence Transformers Documentation. Available online: <https://www.sbert.net/2020> (accessed on 19 June 2025).
13. Sequence Transformers on Hugging Face. Available online: <https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2> (accessed on 19 June 2025).

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.