bioRxiv preprint doi: https://doi.org/10.1101/2023.01.11.523286; this version posted January 15, 2023. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.

Sensitive inference of alignment-safe intervals from biodiverse protein sequence clusters using EMERALD

Andreas Grigorjew[1†], Artur Gynter[1†], Fernando Dias[1], Benjamin Buchfink[2], Hajk-Georg Drost[2*], and Alexandru I. Tomescu[1*]

[1]Department of Computer Science, University of Helsinki, Finland
[2]Computational Biology Group, Max Planck Institute for Biology, Tübingen, Germany
[†]Shared first-author contribution
[*]Shared last-author contribution

Abstract

Sequence alignments have become the foundation of life science research by unlocking biological mechanisms through protein comparisons. Despite its methodological success, most algorithmic innovation in the past decades focused on the optimal alignment problem, while often ignoring information derived from suboptimal solutions. The assumption that the score-derived optimal alignment represents the biologically most relevant choice has led many life scientists to accept this reduced dimension from thousands or millions of possible alignment configurations to one optimal alignment setting. However, we argue that one optimal alignment per pairwise sequence comparison may have been a reasonable approximation when dealing with very similar sequences, but is insufficient when aiming to capture the natural variation of the protein universe at tree-of-life scale. To overcome this alignment-sensitivity limitation, we propose the concept of pairwise alignment-safety as a way to explore the neighborhood of suboptimal alignment configurations when comparing divergent protein sequences. We show that by using alignment-safe intervals, it is possible to encode the defining structural features of proteins even when comparing highly divergent sequences. To demonstrate this, we present EMERALD, a dedicated command line tool able to infer alignment-safe sequence intervals from biodiverse protein sequence clusters. EMERALD effectively explores suboptimal alignment paths within the pairwise dynamic programming matrix and flags robust intervals that are shared across all suboptimal configurations. We apply EMERALD to clusters of 396k sequences generated from the Swiss-Prot database and show that alignment-safe intervals derived from the suboptimal alignment space are sufficient to capture the structural identity of biodiverse proteins even when comparing highly divergent clusters.

Keywords: ***Sequence Alignment, Dynamic Programming, Needleman-Wunsch Algorithm, Protein Folding, Suboptimal Alignments***

Correspondence: `alexandru.tomescu@helsinki.fi` and `hajk-georg.drost@tuebingen.mpg.de`

Introduction

When exploring the diversity of life, we tend to either reduce observations according to similar principles and patterns shared across lineages (comparative method) or aim to deduce the individual mechanistic function with a cause-and-effect-revealing experimental design (functional assessment). In genomics, such attempts translate into comparing genetic sequences according to the similarity of their DNA or protein composition (comparative genomics) or mechanistic analysis of three-dimensional structural conformations of proteins (functional genomics). Recent breakthroughs in protein structure prediction from primary sequence alone [Jumper et al., 2021, Baek et al., 2021] uncovered that integrating comparative and functional genomics into a predictive model can yield groundbreaking insights useful enough to guide mechanistic studies in molecular life sciences. Building on this integrative foundation of sequence comparison and protein structural prediction, we explore how the sequence diversity across the tree of life can be compressed into alignment-robust sequence regions while minimizing the loss of protein structural information.

To compare two biological sequences according to a predefined scoring-scheme, pairwise alignment methodologies have proven useful for various practical applications [Needleman and Wunsch, 1970, Smith and Waterman, 1981]. When constructing a pairwise alignment, a combinatorial space of possible alignment configurations is explored and a single optimal alignment setting is selected (based on the predefined scoring-scheme) and reported to equip experimenters with one plausible solution rather than overwhelming them with a wide range of possible solutions. While sufficient for many applications, including protein sequence similarity search, the reduction of comparison to one solution (even when optimal) can cause enormous information loss about biologically relevant, but suboptimal, alignment configurations, thereby systematically biasing the comparative method when applied at tree-of-life scale. One could argue that handling only optimal alignments is the most parsimonious approach to dealing with complexities when scaling to millions or even billions of pairwise sequence comparisons when organizing a diverse sequence space according to their pairwise identities. However, analogous to the concept of point estimates and confidence intervals in statistical parameter inference [Hastie et al., 2009], neglecting the goodness of fit for any application may result in unrealistic technical optima rather than focusing on quantifying the biological relevance (e.g. functional protein configuration) of reported alignment solutions.

bioRxiv preprint doi: https://doi.org/10.1101/2023.01.11.523286; this version posted January 15, 2023. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.

It seems therefore surprising that the experimental community has grown accustomed to interpret algorithmically derived optimal alignment solutions as biologically most relevant configuration of pairing similar proteins, although theory clearly states that such approximation may only be reasonable when comparing very similar sequences [Naor and Brutlag, 1994] and not when dealing with distant homologs.

We argue that by exploring the space of suboptimal alignment configurations using a quantification method able to capture stable positions across possible alignment paths (*alignment-safe intervals*), novel insights with greater biological relevance can be unveiled and quantified with particular relevance for protein structure evolution at tree-of-life scale.

Fortunately, previously collected evidence suggests that sufficiently screening the suboptimal alignment space for particular configurations that are biologically more relevant can in fact be achieved [Chen and Kihara, 2011, 2008]. Previous work, such as Jaroszewski et al. [2002], Sierk et al. [2010], Cline et al. [2002], has shown that there is a significant connection between the suboptimal alignment space and the structural alignment space and used this link to improve the accuracy of pairwise sequence alignments. It has also been shown that suboptimal alignments are often more accurate than strictly optimal ones and that they contain a high number of correct amino acid residue pairs [Chen and Kihara, 2011], indicating their use in protein structure prediction. Vingron and Argos [1990] showed that so-called *reliable regions*, defined in terms of a *robustness* measure of individual aligned amino acids can identify conserved and functionally relevant regions among two protein sequences. They demonstrate this functional relevance by validating that these conserved regions also correspond to aligned regions of their respective tertiary structures. In detail, Vingron and Argos [1990], Chao et al. [1993] introduce a *robustness* measure for a single pair of aligned symbols between two sequences to assess the difference between the optimal alignment score of the compared sequences without restrictions and the optimal alignment score *not* containing that aligned pair. A related approach suggested by Naor and Brutlag [1994] notes that the space of suboptimal alignments (whose score is within a difference Δ to the optimal solution) can reveal conserved regions when manually inspecting the "graphic representation" of these possible alignments. While this initial suggestion to explore the suboptimal alignment space yielded promising visual insights, no solution was given on how to automate or scale this approach to millions or billions of pairwise comparisons. Currently, most applications favour multiple sequence alignments (MSAs), Hidden Markov Model based approaches (HMMs), or protein language models (Lin et al. [2022]) when interested in sensitively locating conserved regions inside a set of sequences for large-scale phylogenetic applications (i.e., those positions or regions that remain unchanged in the phylogenetic tree). While providing reliable results in a molecular evolution context, calculating optimal MSAs or constructing relevant HMMs for deep homology searches scales exponentially with the number of sequences, which is computationally expensive and not suitable for tree-of-life applications. In addition, MSAs are designed to process highly similar sequences and often perform poorly when comparing divergent sequences (Penn et al. [2010], Levy Karin et al. [2019]).

Here, we overcome these limitations by introducing the application of *solution safety* [Tomescu and Medvedev, 2017], (Methods - Definition 1), for pairwise protein sequence alignments. With alignment safety we can explore the space of optimal *and* suboptimal alignment configurations (i.e. possible pairings of amino acids), and find entire intervals that are common to all or to a given proportion of alignment solutions. We implement this approach in a command line tool, EMERALD. Instead of forcing two (possibly very diverse) sequences into a single optimal alignment configuration, EMERALD embraces the diversity of possible alignment solutions, by revealing *alignment-safe intervals* of the two sequences which appear as conserved (and not even necessarily identical) in the entire space of optimal and suboptimal alignments (Figure 1). To demonstrate the effectiveness of this procedure for sizeable protein comparisons, we first cluster all protein sequences stored in the Swiss-Prot database [Consortium, 2021, Bairoch and Apweiler, 1997] using DIAMOND DeepClust [Buchfink et al., 2021] (Results and Methods) and apply EMERALD to each non-singleton cluster. With this comprehensive analysis, we aim to explore whether our method can provide a competitive solution to project alignment-safe primary-sequence intervals onto the structural conformation of proteins and thereby accelerate the biologically relevant exploration of sequence evolution and their corresponding structural divergence, a sought-after application known to fail with AlphaFold2 (failure to account for point mutations during *de novo* structure prediction) [Jumper et al., 2021].

Results

The main purpose of this study is to notify the genomics community about the advantages of exploring the suboptimal alignment space when dealing with comparisons of vastly divergent sequences where subsequent structural information will be used to make claims about protein sequence and fold evolution at tree-of-life scale. EMERALD allows to achieve this task by equipping users with an automated and scalable software solution to infer alignment-safe subsequences extracted from the suboptimal alignment neighborhood (Figure 1), which we demonstrate experimentally to correspond to conserved regions of the underlying protein structure (such as alpha-helix and beta-sheet, etc).

Defining Alignment-Safety for Pairwise Protein Sequence Alignments. Initially developed in the context of genome assembly [Tomescu and Medvedev, 2017], and later extended to other applications admitting multiple solutions, such as flow decompositions for RNA transcript assem-

bioRxiv preprint doi: https://doi.org/10.1101/2023.01.11.523286; this version posted January 15, 2023. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.

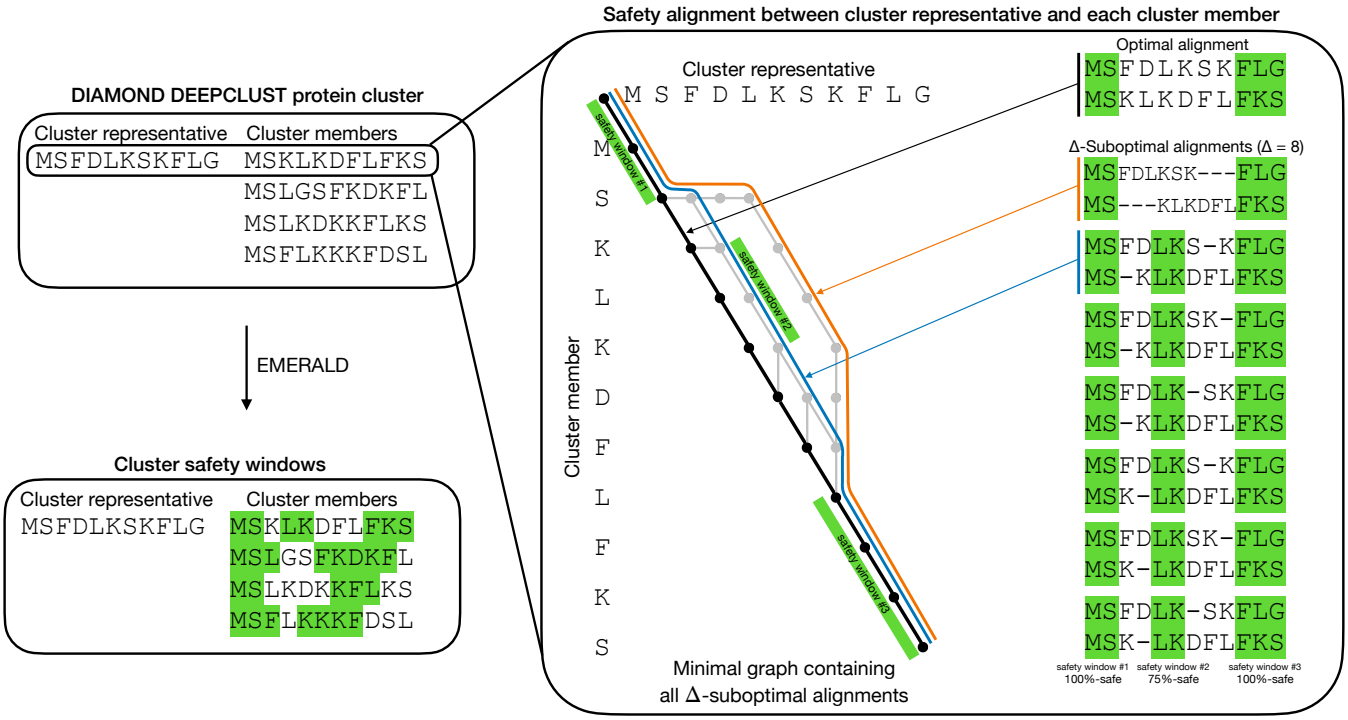


Figure 1: Schematic representation of EMERALD's safety window calculation of a DIAMOND DeepClust cluster containing 4 member sequences. EMERALD performs a pairwise global alignment between the cluster representative against each of the 4 cluster member sequences using affine gap costs and BLOSUM62 as substitution matrix. For the first sequence pair, the right-hand side illustrates the suboptimal alignment graph and their corresponding suboptimal alignment configurations between the two sequences listed as Δ-*suboptimal* alignments (An alignment is Δ-*suboptimal* if its score is not more than Δ smaller than the optimal score). The illustrated graph is one of minimum size to fulfil the property of including all Δ-suboptimal alignments [Naor and Brutlag, 1994] (here, we choose $\Delta = 8$). Source-to-sink paths in the graph correspond to suboptimal alignments; nodes and edges on the unique optimal alignment path are shown in black, while those configurations on a Δ-suboptimal path are illustrated in gray. The optimal alignment path is colour coded in black and the two top Δ-*suboptimal* alignment paths illustrated in orange and blue. For $\alpha = 0.75$ and $\Delta = 8$ we obtain three safety windows shown as green intervals. These three coloured safety windows correspond to subpaths contained in at least $\alpha = 0.75$ (i.e. 75%) of all source-to-sink paths (i.e., of all Δ-suboptimal alignments). Note that the middle safety window is not captured (i.e., contained) by the (unique) optimal alignment, in black, and is only revealed by the subgraph of all Δ-suboptimal alignments. Finally, we project the safety windows onto the cluster member (and cluster representative sequence) as explained in (Methods). This procedure is repeated for all possible pairwise comparisons between the representative sequence and the 4 members, thereby obtaining (α, Δ)-safety windows for each cluster member (bottom left).

bly [Khan et al., 2022], and RNA folding [Kiirala et al., 2019], *solution safety* identifies a set of partial solutions (e.g., an interval of an alignment) to infer *safe* positions or windows present in all optimal and Δ-suboptimal configurations to the problem. Using this definition, previous work by Vingron and Argos [1990], Chao et al. [1993], Naor and Brutlag [1994] can be reintroduced in the context of studying maximally safe partial alignments, whereby *maximal* denotes the property that such partial alignment cannot be extended left or right without losing the characteristic of being safe.

To achieve this property at scale and with biological relevance, we introduce novel computational aspects of the alignment-safety concept and implement these into the *C++* command line tool EMERALD (Methods). First, we generalize the notion of safety by defining a partial solution to be α-*safe* ($\alpha \in (0, 1]$) if it is present in at least a proportion α of all possible solutions. We further generalize α-safety to the space of Δ-suboptimal solutions from [Naor and Brutlag, 1994] that are at most Δ away from the optimal solution, by defining a joint-parameter (α, Δ) which captures partial solutions that appear in at least a proportion α of all Δ-suboptimal solutions. In other words, while Δ captures the boundaries of the suboptimal alignment space that a user wishes to explore, α regulates the quantile-range over all possible alignment solutions. The joint-parameter (α, Δ) then allows users to specify the suboptimal alignment space within a particular quantile range over all solutions that shall be explored. This approach allows us to address the fact that an (α, Δ)-safe partial solution is an interval of arbitrary alignment length (not only a single pair of aligned symbols, as is the case for the robustness measure from [Vingron and Argos, 1990]). Together, we denote all such (α, Δ)-safe intervals as the collection of *alignment-safe protein sequence intervals* or in short *safety windows*. Intuitively, Δ allows sufficient exploration across the suboptimal alignment space within an α range, thus enlarging the solution space, and leading to shorter safety-windows, while α relaxes the safety requirement by enforcing that only a α-fraction of the Δ-suboptimal alignment-configurations need to have the same amino acid to extend safety windows. We show that optimizing the α and Δ configuration can be a powerful tool for regulating the biological relevance safety-windows can capture in diverse protein sequences and their respective three-dimensional structures.

Additionally, we explore the biological signatures that can be captured when exploring the suboptimal alignment space

bioRxiv preprint doi: https://doi.org/10.1101/2023.01.11.523286; this version posted January 15, 2023. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.

using our alignment-safety approach. In order to assess biologically relevant outcomes from quantifying alignment-safety, we assume that the ability to retain structural information known to influence protein folding such as alpha-helices, beta-sheets, loops, etc while compressing the full primary protein sequence to alignment-safe windows is the 'ground truth' of biological relevance when comparing fold structures using primary sequences alone. For this purpose, we extracted the secondary structure from the Swiss-Prot database for each protein sequence using the command line tool Stride [Frishman and Argos, 1995]. For each residue in a sequence, Stride assigns a secondary structure type, including but not limited to: "AlphaHelix", "310Helix", "PiHelix", "Strand", "Bridge", "Coil", "TurnI' ", "TurnII' ", "TurnVIa", "TurnVIb", "TurnVIII", "TurnIV", "GammaClassic", "GammaInv" and "Turn". We further distinguish secondary structure types by placing them into two distinct categories: stable and not stable. Coil, Strand and Turn were labeled as not stable while the rest of the secondary structure types were labeled as stable. While this classification is naive in the first instance (for example, because there are well-known cases where Coils, Strands and Turns can be stable to fix certain protein conformations and vice-versa, alpha-helices and beta-sheets can be fairly flexible and unstable (Levy Karin et al. [2019], Bondos et al. [2021])), we use this distinction only to exemplify how users can categorize known protein-structural features into distinctive structural feature classes to benchmark the suboptimal alignment-space for their domain-specific application and quantification of biological relevance. The main motivation behind such categorization is to test which regions of a protein structure are robustly encoded in alignment-safe intervals. We envision that users make extensive use of this benchmarking setup to explore, quantify and test the biological relevance of harnessing the suboptimal alignment-space for encoding structural features and protein evolution of their (divergent) sequences of interest.

***In silico* experimental design to investigate the biological relevance of alternative suboptimal alignment configurations.** The focus of our experimental design is to test whether exploring the suboptimal alignment space when dealing with millions or billions of pairwise alignments spanning a comprehensive sequence diversity across the tree of life yields sufficiently more information (compared to a single optimal alignment configuration) to increase biologically relevant inference power when dealing with protein structure prediction and structural evolution tasks. To achieve this, we retrieved the Swiss-Prot database from [Bairoch and Apweiler, 1997] (June 2021), which represents a manually curated subset of the UniProtKB [Consortium, 2021] database. In the version of June 2021, Swiss-Prot contains approximately 560k protein sequences which can be retrieved as a FASTA-file. After data retrieval, we filter out protein sequences that did not have corresponding AlphaFold2 predicted three-dimensional structures [Jumper et al., 2021, Varadi et al., 2022]. We then clustered this dataset with DIAMOND DeepClust, a new sensitive deep-sequence-clustering method implemented into DIAMOND since version 2.1.0 [Buchfink et al., 2021]. We filter out cluster members with no stable bases and clusters of size 1, resulting in 15934 clusters and 396k sequences in total. To achieve a comprehensive overview of deep-homology associations between proteins across the full diversity intrinsic to the 396k Swiss-Prot sequences, clustering was carried out using a percent-identity threshold of 20% and 75% length coverage. This threshold was motivated mainly by two factors: 1) conformity with the twilight zone of protein evolution where sequence identity greater than 20-35% can still be reliably associated with structural similarity above this zone, while this association is 'breaking' otherwise [Rost, 1999]; and 2) to maximize the diversity of the protein sequence space across the tree of life. It is important to note that although the minimum pairwise identity threshold ensures that distant similarities are detected, some clusters can yield pairwise sequence compositions that have significantly higher identity-relations than the chosen threshold of 20%, since DIAMOND uses no upper identity threshold when clustering.

Next, we run EMERALD to calculate safety windows of all sequence and representative pairs of each cluster. To explore the influence of the suboptimal alignment space depth on capturing biologically relevant features, we test various parameter-combinations performed using three different alpha parameters (0.51, 0.75, 1) and seven delta parameters (0, 2, 4, 6, 8, 10, 15), values that are in the same order of magnitude as the BLOSUM62 metric [Henikoff and Henikoff, 1992], resulting in 21 safety window calculations per cluster (Figures 2 to 3). As cluster representative, we selected the cluster centroids reported by DIAMOND DeepClust's greedy set cover method. Finally, we benchmarked the CPU runtime and maximal memory consumption of these runs (Figure 4).

Benchmarking and analysis of stable versus unstable protein structural features that are encoded by alignment-safe windows. For quantification and comparison of safe intervals in the context of stable versus unstable bases, we utilise several benchmarking metrics: safety coverage, stable coverage, stable structure overlap and stable structure retention. Safety coverage is defined by the portion (%) of a protein sequence which is reported by EMERALD as (alignment-) safe, while stable coverage is the portion (%) of a sequence that is considered stable according to the distinct protein-structural features classification (user-defined) described in the previous section. We further classify a sequence position as *true positive (TP)* if it is both safe and stable, *false negative (FN)*, if it is not safe but stable, and *false positive (FP)* if it is safe but not stable. Based on these distinctions, we compute the metrics *stable structure retention (precision) = TP/(TP+FN)*, indicating the percentage of stable positions captured by safety windows, *stable structure overlap (recall) = TP/(TP+FP)*, indicating the proportion of safety windows that is also stable, and their combination, *F1-score*, which is the harmonic mean of the stable structure retention and overlap. This

bioRxiv preprint doi: https://doi.org/10.1101/2023.01.11.523286; this version posted January 15, 2023. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.

analytics tool allows users to benchmark according to their own distinction of structural classes when determining how much information is gained when incorporating the suboptimal alignment space to the sequence comparison task.

Figure 2 illustrates the stable structure retention and safety coverage benchmark for our 396k sequence dataset. Naively, we expect the stable structure retention and the safety coverage to be similar, as we are likely to cover e.g. 60% of all stable elements if 60% of the sequence is covered. Ideally, the stable structure retention would always be close to 100% with all varying safety coverages, meaning that safety corresponds perfectly to stable elements. Figure 2 further shows that as we consider sequences that are more dissimilar, safety coverage decreases, since we increase the space of optimal and suboptimal solutions.Figure 2 **b** shows the stable structure retention that is restricted to all sequences with a safety coverage of at most 70%, resulting in a decrease of the stable structure retention. In both, Figure 2 **a** and **b**, the average values of the stable structure retention exceeds the average value of the corresponding safety coverage, and in the identity range $[40\%, 70\%]$ the average stable structure retention is at 80% even when we restrict the sequences to have no more than 80% safety coverage. This result shows that the proportion of stable positions inside the safety windows is larger than the proportion of stable positions out of all the sequence (i.e., $(\text{stable} \cap \text{safe})/\text{safe} > \text{stable}/\text{sequence length}$).

Figure 2 **a** illustrates that the best tradeoff between safety coverage and stable structure retention is the identity range 40%-70% sequence identity. This range setting introduces the biggest gap between medians while ensuring that the stable structure retention does not drop below 70%. In Figure 2 **b** the median of the stable structure retention in the identity range of 40%-70% is at 80%, despite the safety coverage being restricted to be at most 80%. Furthermore, we analysed the stable structure retention over all combinations of parameters (Supplementary Figures 2 to 4). Overall, the stable structure retention is always slightly larger than the safety coverage. For $\delta = 0$ (i.e. we consider only optimal alignment parts) and all three α values $0.51, 0.75$ and 1.0, the safety coverage is close to 100% for identities of at least 20%, which implies that the substitution matrix is very good at finding a unique optimal alignment, which motivates exploring the suboptimal alignment space. As we predicted, decreasing α corresponds to increasing the stable structure retention and safety coverage, while increasing δ corresponds to decreasing them.

Over the whole dataset and all combinations of parameters, the stable structure overlap stays constant (Supplementary Figure 1) with the median at around 43%.

Figure 3 further explores the relationship between stable structure retention and safety coverage. It restricts the sequences in the x-axis to those with stable structure retention of at least x, plotting the safety coverage (on the left) and the proportion of considered sequences (on the right). Strikingly, in Figure 3 **a** we can see that for about 20% of all sequences, or of all sequences in the identity range $[40\% - 70\%]$ (turquoise, and green curves, respectively on the right for $x = 1.00$) EMERALD retains *all* their stable positions, with a safety coverage of only 15%. This illustrates that, in contrast to optimal alignment approaches, for lower identity bounds (dissimilar sequences) EMERALD manages to reveal structurally conserved intervals. Similarly, EMERALD achieves a stable structure retention of 80% for around 75% of all the sequences with a safety coverage of around 65%. In other words, EMERALD declares 65% of the sequence as safe, capturing 80% of all stable amino acids. This result further shows that by relaxing the stable structure retention criteria from 100% down to 80%, EMERALD is able to reduce the sequences to their safety intervals from full length to 65%. In Figure 3 **b**, we analogously restrict the sequences only to those of safety coverage of at most 80%. Here, for example, the green curve shows that around 10% of the sequences in the identity range 40% - 70% can be reduced to a safety coverage of nearly 5%, without losing the stable structural retention constraint of 100%. This suggests that EMERALD embraces the sequence diversity to narrow down the structurally-conserved context of a sequence.

Supplementary Figure 5 analyses the safety coverage and the F1 score, and it shows that, as we consider clusters with smaller post-computed identity values, F1-score has only a minor decrease, but safety coverage has a marked decrease. This indicates additionally that safety windows have a better ability to capture stable structural elements in clusters with smaller identity values.

Finally, we benchmarked EMERALD's run time and memory consumption (Figure 4) on a computing server with 32 cores (2x hyper-threading, 64 threads) and 512GB of RAM using the clustered 396k sequences from the Swiss-Prot database. Overall, none of our clusters ran more than 17 minutes, and, except for two clusters, they did not use more than 6.5GB of memory. While the time increases quadratically in the lengths of sequences, trivially it increases only linearly with the number of sequences in the cluster. This result illustrates that EMERALD can scale to millions of sequences and thousands of clusters to explore the protein universe across the tree of life.

Discussion

For the past decades pairwise sequence alignments have served silently and reliably as foundation of comparative and functional genomics applications. Algorithmic innovation focused on computing ever faster heuristics for retrieving optimal alignment solutions at scale or extending comparisons to multiple sequence alignments or Hidden Markov Model based statistical alignment approaches. However, only little innovation occurred in the pairwise alignment field on quantifying the biologically most relevant alignment configuration from a collection of up to millions of possible

bioRxiv preprint doi: https://doi.org/10.1101/2023.01.11.523286; this version posted January 15, 2023. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.

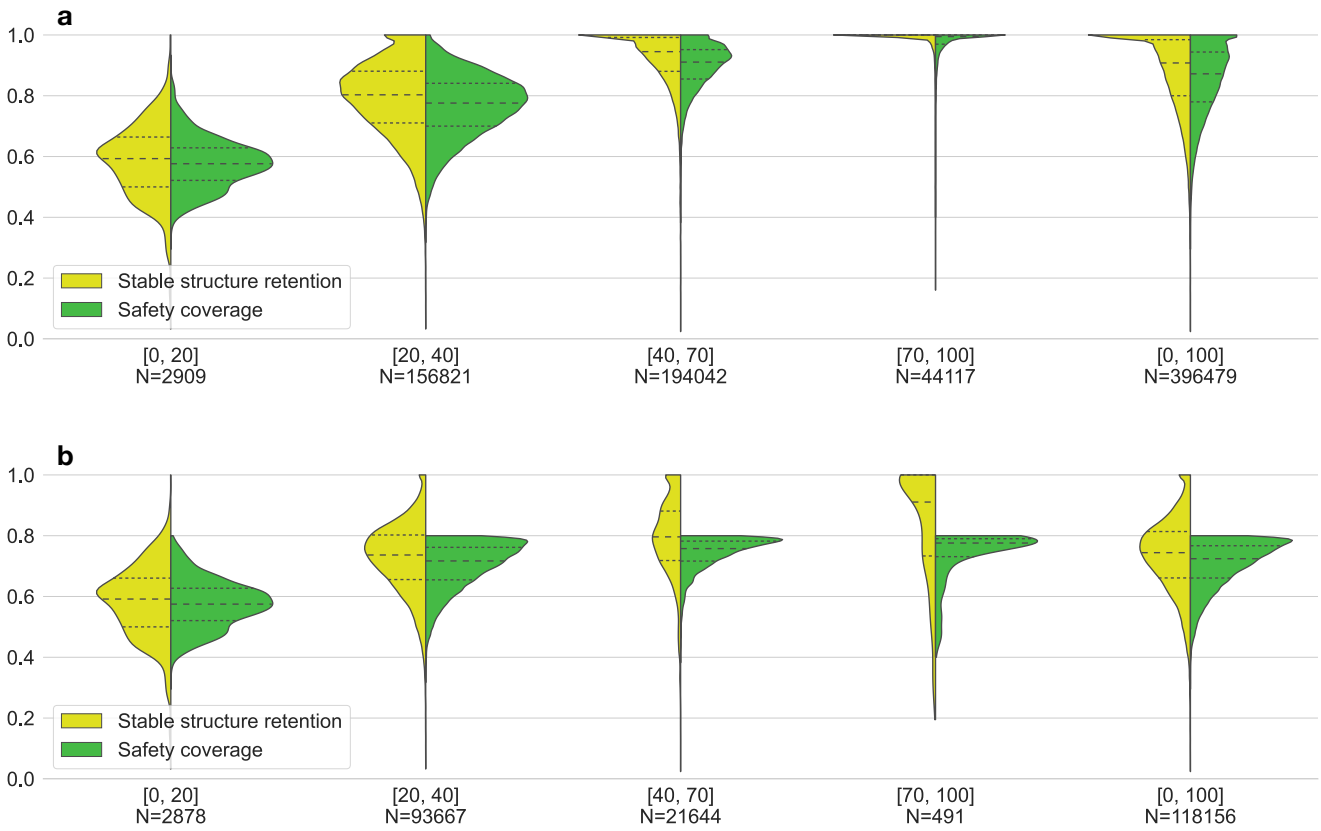


Figure 2: Comparing stable structure retention and safety coverage (y-axis) for several sequence identity ranges between cluster members against the cluster representative (x-axis) carried out with EMERALD parameters $\alpha = 0.75$ and $\Delta = 8$ on all obtained sequences, where N denotes the number of sequences in each identity range. The dashed lines indicate medians and the dotted lines the first and third quartiles. The safety coverage increases with higher identity ranges due to the smaller size of the suboptimal alignment space for high identity sequence pairs. Since the stable structure retention quantifies the proportion of stable amino acids that are alignment-safe, it increases alongside an increase in safety coverage. The high stable structure retention results indicate that EMERALD is indeed able to capture biologically relevant stable positions, as they make up a higher proportion when restricting the sequence to safety windows. **a** All sequences are included (i.e., with safety coverage of at most 100%). **b** Sequences restricted to those having safety coverage of maximum 80%.

(suboptimal) alignment solutions. While biological meaning in the context of alignment optimization is a vague concept, in the early days of comparative and functional genomics the ability to encode structural information of proteins was among the main applications underlying the benchmarking of biologically meaningful alignment configurations [Chen and Kihara, 2011, Hastie et al., 2009]. The consensus then was that focusing on optimal alignments is a reasonable heuristic when dealing with highly similar sequences, since the optimal alignment solution can indeed encode a good representation of conserved protein structure such as alpha-helix and beta sheets (Ranwez and Chantret [2020]). Still applied as a main assumption in functional genomics today when employing pairwise alignment searches, protein sequences are usually screened for high similarity across distant species or strains to test whether this retained sequence identity translates into structural conservation and potential functional similarity. Alternative approaches to increase alignment sensitivity such as MSAs or HMMs are often impractical when dealing with the sequence diversity and sheer scale of the hundreds of million of protein sequences now available across the tree of life.

In this study, we aimed to determine whether quantifying suboptimal alignment configurations in pairwise sequence alignments can significantly improve the sensitivity of identifying relationships between features of protein structure across the tree of life when analyzing a large diversity of protein sequence space and making comparisons between hundreds of thousands of species [Buchfink et al., 2021]. To approach this quest, we designed an *in silico* experiment to annotate all alignment-safe positions of the Swiss-Prot database. Using this annotation approach, we investigate how the quantification of the suboptimal alignment space can refine biologically relevant interpretations such as conserved protein structural features. We asked how information about suboptimal alignment configurations at scale can be harnessed to predict protein structural change when the proportion of alignment-safe positions in distant sequence alignments is reduced. Finally, we introduced *alignment-safety* as a new methodology to approach such questions and the command line tool EMERALD to implement our methodology at scale.

While previous work focused on assessing the quality or statistical robustness of optimal alignment configurations in comparison with a set of suboptimal alignment solutions (optimal alignment neighborhood) [Kschischo and Lässig, 1999,

bioRxiv preprint doi: https://doi.org/10.1101/2023.01.11.523286; this version posted January 15, 2023. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.

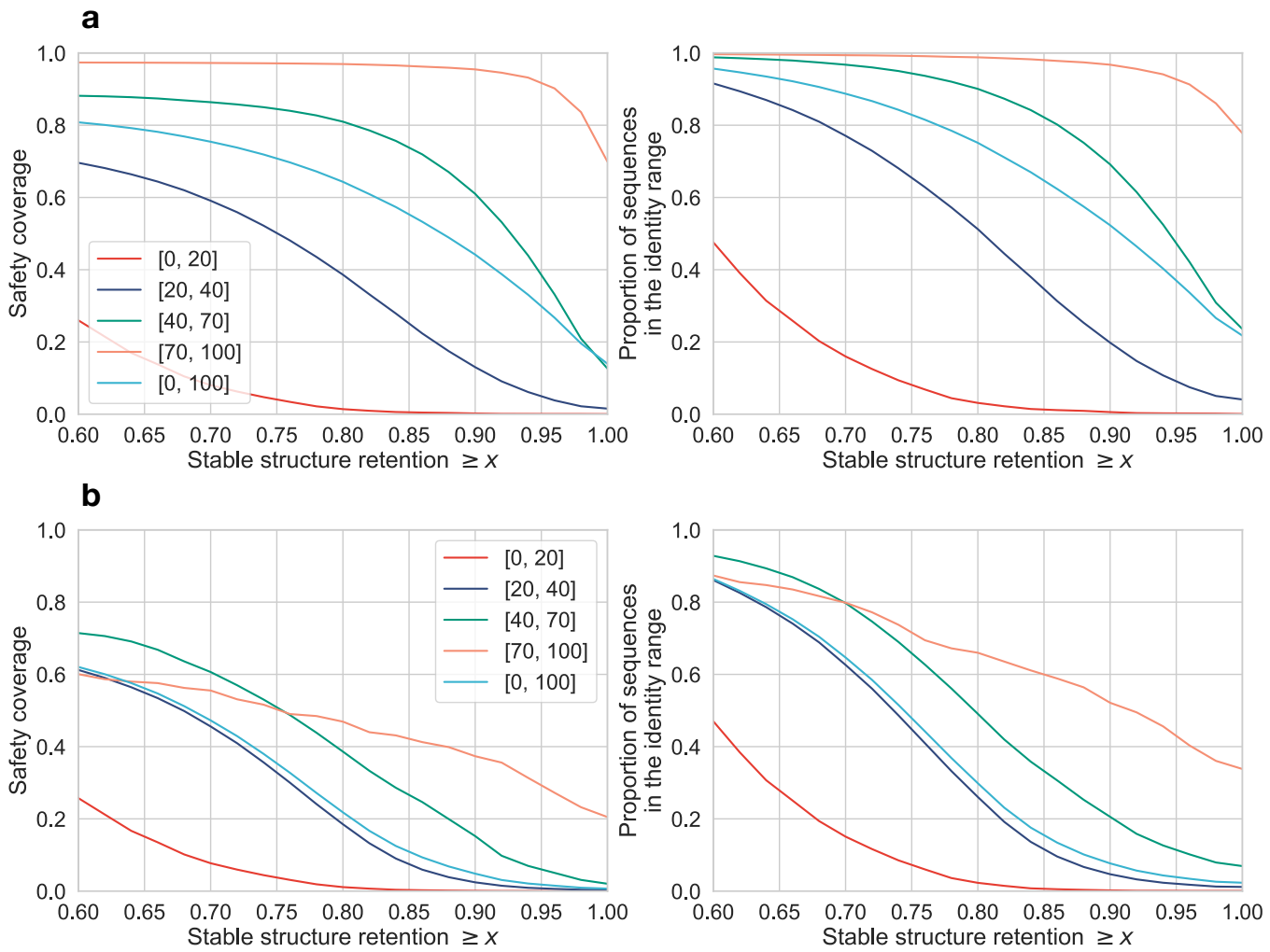


Figure 3: For a given threshold x, safety coverage of the sequences whose stable structure retention is at least x (on the left), and the proportion of the sequences whose stable structure retention is at least x (on the right). The results are split in the identity ranges from Figure 2. **a** All 396k sequences are included. **b** Restricted to sequences whose safety coverage is at most 80%. The turquoise curves, which cover sequences of all identities, show in **a** that around 20% of the sequences (right plot) have a stable structure retention of 100% and only 15% of safety coverage (left plot), while the green curves, which cover sequences from the identity range 40% - 70%, show in **b** that 10% out of the sequences with at most 80% safety coverage inside this identity range (right plot) have a stable structure retention of 100% and only a safety coverage of around 5% (left plot).

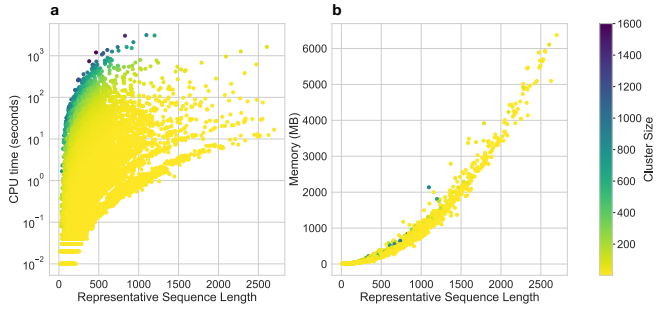


Figure 4: CPU runtime of EMERALD. (**a**) Computational run time for all DIAMOND DeepClust clusters generated from the filtered Swiss-Prot database using the threshold combination $\alpha = 0.75$ and $\Delta = 8$ and calculated on a single thread. Each dot corresponds to a protein sequence cluster and the colour of each dot indicates its number of corresponding member sequences. (**b**) Maximum memory consumption of EMERALD runs. For all trialed α and Δ parameter settings the average memory consumption for each cluster ranged between $205 - 207$ Mb and the average run time between $11.4 - 11.5$ seconds.

Schlosshauer and Ohlsson, 2002, Zhang and Marr, 1995] in the context of protein homology modeling or threading, our work significantly extends these concepts to associate protein sequence evolution with their respective structural change while scaling to millions of pairwise comparisons and trillions of suboptimal solutions at tree-of-life scale. We achieve this by inferring the robust amino acid positions across a range of suboptimal alignment-configurations. As a result, we learned that when attempting to relate the sequence biodiversity of the protein universe to the evolution of protein structure, a detailed inference and exploration of how alignment-safe positions are retained or lost throughout the tree of life can serve as robust proxy for how conserved structural features change over evolutionary time. We interpret this result such that alignment-safe positions seem to encode the backbone of protein structure and that further research is required to unveil the causal associations between alignment-safe positions and positions giving predictive signatures when inferring structural conservation above the twilight zone of protein evolution [Rost, 1999].

bioRxiv preprint doi: https://doi.org/10.1101/2023.01.11.523286; this version posted January 15, 2023. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.

Conclusions

We envision that studies exploring the suboptimal alignment space when comparing protein sequences pairwise across the protein universe will stimulate a renaissance of sequence alignment research attempting to address the remaining shortcomings of fold predictions such as dealing with disordered parts of proteins and incorporating diverse mutation patterns into fold evolution predictions. Our alignment-safety methodology and EMERALD software are designed to assist these efforts of associating new information gained from suboptimal alignments with biologically relevant phenotypes of protein structure in an evolutionary context. We designed our approach to be sufficiently fast and sensitive to scale to millions of sequences with various degrees of divergence to supply the data-intensive demands of the biosphere genomics era.

Methods

Conceptually, we build on the approach introduced by Naor and Brutlag [1994] and vastly extend its methodological depth and computational scalability for tree-of-life scale applications. In detail, the approach introduced by Naor and Brutlag [1994] is restricted to consider only aligned symbols that are part of *all* suboptimal alignments, ignoring pairs which are part of the optimal alignment itself. For example, suppose that an aligned pair is common to all the 1000 optimal alignments of score q, there is a single suboptimal alignment of score $q-1$ not containing the pair, and the next suboptimal alignments not containing the pair have score $q-100$. In this case, the single suboptimal alignment of score $q-1$ makes the pair "not conserved" under the approach of Naor and Brutlag [1994], and drastically decreases the robustness of the pair (from 100 to just 1), since the robustness measure cannot quantify the *proportion* of (sub-)alignments containing the pair. In addition, they define robustness independently for each aligned amino-acid pair thereby excluding the opportunity to quantify robust regions, which are often observed in natural settings. As a result, although Naor and Brutlag [1994] provide a first theoretical template to study suboptimal alignment spaces, they fail to deliver scalable algorithmic solutions, software tools, and biological validation to investigate the protein sequence diversity space in the context of robust - alignment safe regions for tree-of-life scale applications. In fact, their methodology does not exceed a manual analysis of graphical alignment representations with little potential for automation and efficient scaling.

Inference of Alignment-Safe Protein Sequence Windows. We formally introduce the calculation of alignment-safe intervals through the thresholded exploration of the suboptimal pairwise alignment space. Let A and B be two strings over an alphabet Σ of length n and m, respectively. In this study, we refer to an optimal alignment between A and B, when a particular alignment *maximizes* a scoring function based on an externally provided match/mismatch cost matrix. First, we consider scores provided by a amino acid substitution matrix (such as BLOSUM62 [Henikoff and Henikoff, 1992]) and solely focus on optimal alignments, before we introduce affine-linear gap costs in the subsection Introducing gap penalties [Myers and Miller, 1988] and the suboptimal alignment space to explain the necessary changes in our approach.

In sequence bioinformatics it is established that such optimal global alignments between A and B can be computed via dynamic programming [Needleman and Wunsch, 1970]. A common result is that alignments of maximum score are in bijection with maximum-weight (*optimal*) paths in the directed acyclic graph (*DAG*) which corresponds to this dynamic programming table. More formally, we can define the *alignment DAG of A and B* as $G(A,B)=(V,E)$, with $V=\{0,\ldots,n\}\times\{0,\ldots,m\}$, with each node (x,y) denoting three out-going edges in E to $(x+1,y),(x,y+1)$ and $(x+1,y+1)$ (and thus, the DAG has unique source $s=(0,0)$ and unique sink $t=(n,m)$). We denote paths in a graph by the notation $P=(v_1,v_2,\ldots,v_k)\in V^k$ and their restriction by $P[v_L..v_R]=(v_L,v_{L+1},\ldots,v_R)$. The first two of the edges correspond to a gap and the third edge corresponds to an alignment of the symbols $A[x]$ and $B[y]$. If we assign the scores of the substitution matrix as well as the gap score to the corresponding edges as weights, then finding an optimal alignment corresponds to finding a maximum-weight path from s to t (*s-t path*) in G. We define such score maximizing paths as *optimal*. We are then interested in discovering those *safe* partial alignments that are common to all optimal alignments.

We can further generalize the notion of safety by also considering a parameter $\alpha\in[0,1]$ and analogously explore α-safe partial alignments that appear in at least the proportion α of all optimal alignments. For $\alpha=1$, 1-safety coincides with safety, and considering $\alpha<1$ allows us to have potentially longer α-safe paths. We can define these notions formally based on the graph-centric definition of an optimal global alignment.

Definition 1: Let $\alpha\in[0,1]$, let A and B be two strings, and let $G(A,B)$ be the global alignment DAG connecting A and B. We denote that a path P in G is

- *safe*, if P is a subpath of all optimal s-t paths of G;
- *α-safe*, if P is a subpath of at least an α proportion of all optimal s-t paths of G.
- *maximally α-safe*, if it is not a subpath of a longer α-safe path.

We define the set of all maximal α-safe paths as *α-safety windows*. We will omit α when it is clear from the context.

Given the alignment DAG $G(A,B)$ that connects A and B, we define $G_0(A,B)=(V_0,E_0)$ as the unweighted minimal

bioRxiv preprint doi: https://doi.org/10.1101/2023.01.11.523286; this version posted January 15, 2023. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.

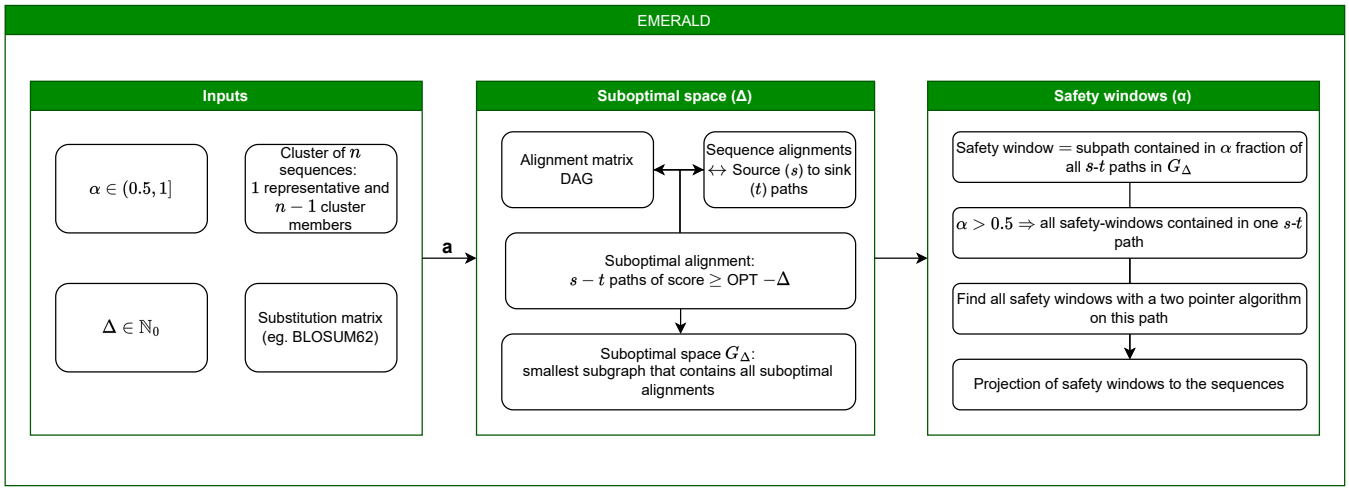


Figure 5: Conceptual overview of EMERALD's safety window calculation workflow. As input EMERALD receives a set of clusters in *fasta* format. For example, such protein sequence clusters can be generated using DIAMOND DeepClust or alternative clustering methods. Next, users can specify the scoring matrix (e.g. BLOSUM62) according to which optimal alignment configurations will be determined. Each cluster member sequence is then globally aligned against the cluster representative sequence (centroid) using the pairwise Needleman-Wunsch alignment algorithm. The resulting dynamic programming (DP) matrix of each pairwise comparison is then encoded as a graph data structure to search for optimal and suboptimal alignment paths according to the selected scoring matrix and the threshold configurations defining the suboptimal alignment space. Once all alignment-safe intervals are computed, EMERALD projects these safety intervals (safety windows) back to the representative sequence, thereby annotating the sequence intervals that are robust across all possible alignment configurations within the suboptimal alignment space.

subgraph of $G(A,B)$ to include all optimal s-t paths in $G(A,B)$ (thus s and t are nodes in both, $G(A,B)$ and $G_0(A,B)$). Backtracing an optimal solution in $G(A,B)$ can be done by taking any in-coming edges of any node (x,y) that is part of an optimal path. Due to the bijection property of G_0, we can ignore edge weights and focus on exploring only the set of all its s-t paths, since these are bijectively linked with the optimal alignments between A and B. We note that $G_0(A,B)$ is weakly connected (i.e., there is an undirected path between any two pairs of nodes, since by definition every node appears in some s-t path of $G_0(A,B)$). It is easy to notice that the safe edges of $G(A,B)$ are exactly the edges that when removed from $G_0(A,B)$ leaves $G_0(A,B)$ no longer sparsely connected. Such edges are also referred to as *bridges* of the undirected graph underlying $G_0(A,B)$ and they can be computed in linear time and in proportion to the size of $G_0(A,B)$ [Tarjan, 1974].

Here, we propose a refined approach to compute α-safe paths based on counting s-t paths in $G_0(A,B)$. For each node v, let $d(v)$ be the number of paths from v to t. We can calculate these numbers with the following recurrence:

$$d(v) = \begin{cases} 1, & \text{if } v = t, \\ \sum_{u \in \mathcal{N}^+(v)} d(u) & \text{otherwise,} \end{cases} \quad \textbf{(1)}$$

where $\mathcal{N}^+(v) = \{u \in V_0 \mid (v,u) \in E_0\}$. Since G_0 is a DAG, the recurrence is well-defined, and it stores the total number of s-t paths in $d(s)$. Analogously, we can also compute the number $d_r(u)$ of s-u paths for any $u \in V_0$. Given these two counts for any edge $e = (u,v) \in E_0$, we can define

$$p(e) := \frac{d_r(u) \cdot d(v)}{d(s)} \quad \textbf{(2)}$$

as the proportion of all s-t paths of G_0 that e is part of. We can also understand $p(e)$ as the probability of e appearing in an arbitrary s-t path of G_0. Likewise, given a path $P = (v_1, \ldots, v_k)$ of nodes $v_i \in V_0$, the proportion of s-t paths that P is part of is given by

$$p(P) := \frac{d_r(v_1) \cdot d(v_k)}{d(s)}. \quad \textbf{(3)}$$

Thus, P is α-safe if and only if $p(P) \geq \alpha$.

The following lemma shows that if $\alpha > 0.5$, then there exists an s-t path P^*, such that any α-safe path is a subpath of P^*. This not only simplifies the algorithm, but it also makes it computationally efficient since it guarantees that the number of maximal α-safe paths is proportional to the size of this path (i.e., $\mathcal{O}(n+m)$). Moreover, since P^* is an s-t path in G_0, it corresponds to an optimal alignment between A and B, and thus all safety windows can be reported as intervals of this alignment.

Lemma 1: Let A and B be two strings based on an amino acid alphabet Σ. If $\alpha \in (0.5, 1]$, then there is an s-t path P^* which contains all the α-safe paths of $G(A,B)$.

Proof: As defined above, for an edge e let $p(e) \in [0,1]$ denote the proportion of optimal paths it is part of. It is clear that

$$\sum_{e \in \delta^+(v)} p(e) \leq 1,$$

for all nodes v, where $\delta^+(v)$ is the set of all outgoing edges of v.

We assume the opposite, meaning that there are two α-safe edges e_1 and e_2 for which no common s-t path exists. Since by definition of α-safe edges, $p(e_1) > 0.5$ and $p(e_2) > 0.5$, we obtain the contradiction

$$\sum_{e \in \delta^+(s)} p(e) \geq p(e_1) + p(e_2) > 1,$$

bioRxiv preprint doi: https://doi.org/10.1101/2023.01.11.523286; this version posted January 15, 2023. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.

as the paths that cross e_1 are distinct of the paths that cross e_2 but all start at node s. Finally, since each pair of α-safe edges share a path, there must exist a single path P^* containing all α-safe edges. ■

Next, we give an approach on how to find such a path P^*.

Lemma 2: Let $G = (V, E)$ be a DAG with source s and sink t. Given a set of edges $A = \{e_1, e_2, \ldots, e_k\}$, we can find an s-t path that either contains all the edges in A or return the information that such path does not exist, in time $\mathcal{O}(|V| + |E|)$.

Proof: Let T be a topological order of all nodes in V and let A be sorted topologically by the tails of the edges. Assume we have a path from s to an edge $e_i = (u_i, v_i)$. The task is to determine a path to the edge $e_{i+1} = (u_{i+1}, v_{i+1})$. For any path P_{e_i} from v_i to u_{i+1} it must hold that $T[x] \in [T[v_i], T[u_{i+1}]] =: I\ (\subseteq V_0)$ for all nodes x contained in P_{e_i}. To achieve this, we perform a graph traversal (for example, depth-first search) in I and visit every node at most once. If u_{i+1} is not found after the graph search, it follows that the path does not exist. At the end of this procedure, if no edge in A is left, we just connect the path to t. ■

In Algorithm 1, we provide the pseudo-code to infer safety-windows. In lines 1 to 6, we construct the subgraph $G_0(A, B)$ of optimal paths, calculate the ratios $p(e)$ for all $e \in E$ and find a path P^* according to Lemma 1. Next, we calculate the safety-windows with a two-pointer algorithm on P^*. The iterator variable here is R (right pointer), and we use a second left pointer L with $L \leq R$. The idea is to keep the value L as small as possible, to maintain the safety-windows left maximal, while moving L up until the interval $[L, R]$ of P^* becomes α-safe. On line 13 we add the interval into the list of safety-windows if it is also right-maximal (line 12).

Theorem 1: Let $\alpha \in (0.5, 1]$. Given two strings A and B of length n and m, respectively, Algorithm 1 computes the safety windows of A and B in time $\mathcal{O}(n \cdot m)$, assuming constant-time arithmetic operations.

Proof: We analyse the runtime line by line. First, constructing $G_0(A, B)$ can be performed in $\mathcal{O}(n \cdot m)$ by the standard dynamic programming approach. As shown earlier, $d(v)$ overall can be calculated in linear time (here $\mathcal{O}(n \cdot m)$) for DAGs. Line 5 runs in constant time, so this for-loop also runs in linear time.

Next, in line 6 we find an s-t path P^* containing all α-safe edges, using the approach given in Lemma 2. Note that the path exists by Lemma 1.

Finally, we calculate the safety windows by iterating over P^* and store them in the set W. Both variables R and L require at most $\mathcal{O}(n+m)$ iterations and such W will at most contain $\mathcal{O}(n+m)$ safety windows.

Let $[\ell, r) \in W$. According to the choice of ℓ in line 10, such an interval is α-safe. It is maximal and thus a safety window, since neither $[\ell - 1, r) \in W$ nor $[\ell, r+1) \in W$.

Next, we show that all safety windows are in W. Let $[\ell, r)$ be a safety window. According to the iteration in line 9, R will eventually be equal to r. Consider the start of this iteration. If $L \leq \ell$, the iteration in line 10 will ensure that L will be equal

Algorithm 1: Pseudocode for inferring alignment-safety windows

Input : Two strings A and B, $\alpha \in (0.5, 1]$
Output: All α-safety windows of A and B, as subpath intervals in a path as in Lemmas 1 and 2 in $G(A, B)$

```
   // We define p(e), p(P) as in eq. (2,3)
1  Construct G_0(A,B) = (V_0, E_0);
2  for v ∈ V_0 do
3      Compute d(v) and d_r(v);    // As in eq. (1)
4  for e ∈ E_0 do
5      Mark e as α-safe if p(e) ≥ α;
6  Find one s-t path P* = (v_0, v_1, ..., v_{k-1}) in G_0(A,B)
     containing all α-safe edges;    // As in Lemma 2
7  W ← ∅;
8  L ← 0, R ← 0;
   // Find maximal safe paths in P*
9  while R < k do
10     while L < R and p(P*[v_L..v_R]) < α do
11         L ← L + 1;
12     if L < R and p(P*[v_L..v_{R+1}]) < α then
13         W ← W ∪ {[L, R]};
14     R ← R + 1;
15 return P* and W.
```

to ℓ by definition of safety windows and we insert $[\ell, r) = [L, R)$ into W. We now assume $L > \ell$. In previous iterations $R \leq r$, we must have increased L to make it larger than ℓ. But since $[\ell, r)$ is α-safe, all intervals $[\ell, R)$ for $R \leq r$ are α-safe, and we never increase L if $L = \ell$ and $R \leq r$. ■

Note that we assume that arithmetic operations run in constant time. If $G_0(A, B)$ contained all vertices in $G(A, B)$, then the number of paths will be equal to the exponentially growing Delannoy numbers [Banderier and Schwer, 2005] (i.e. if $G_0(A, B) = G(A, B)$, then $d(v) \in \mathcal{O}(5.8^n/\sqrt{n})$, as this value is iteratively the sum of three previous values). Though, even if $G_0(A, B)$ only consists of nodes close to the diagonal line, the d values can grow up to $\mathcal{O}(2^n)$.

Introducing gap penalties. So far, we only considered scores given by a substitution matrix $M[1..\sigma][1..\sigma]$ for σ characters as well as by gaps through insertions and deletions when determining (sub)optimal global alignments between two protein sequences. While accounting for gaps in a global pairwise alignment is necessary to retrieve biologically relevant alignment configurations, finding the right balance between gap introduction and similarity optimisation is an important consideration to make.

To achieve this, we generalise the gap scoring function to be affine-linear. This means that when creating a gap of length ℓ in the alignment, we obtain the score $\text{gap}_{p,g}(\ell) = p + \ell \cdot g$. The value $p \in \mathbb{Z}$ is the *gap penalty*, chosen in order to minimise the number of gaps in an optimal alignment. Before

bioRxiv preprint doi: https://doi.org/10.1101/2023.01.11.523286; this version posted January 15, 2023. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.

having introduced a gap score, we have set $p = 0$. To accommodate gap scores in our previously defined DAG G, we can slightly modify the graph to address the known issue that paths are then not in 1:1 correspondence to alignments anymore [Myers and Miller, 1988]. We replace each node (i,j) by three nodes $C(i,j), D(i,j)$ and $I(i,j)$. Nodes D and I correspond to being inside a gap, erasing the characters in A and B respectively, while node C is the default node, from which we can decide to either match two characters or to introduce a new gap. We thus create the following edges for C: $C(i,j) \to C(i+1,j+1)$ to match A_i with B_j, $C(i,j) \to D(i+1,j)$ to align A_i with a gap and $C(i,j) \to I(i,j+1)$ to align B_j with a gap. The last two of the edges also define the start of a gap. For D (and analogously for I), we create the following edges: $D(i,j) \to C(i,j)$ to close a gap and $D(i,j) \to D(i+1,j)$ to extend a gap. The new graph is an s-t DAG of size $\mathcal{O}(n \cdot m)$ which retains all properties to calculate safety-windows as previously described. Additionally to the substitution matrix and the gap score, we assign edge weights for closing and opening a gap, commonly chosen as 0 for closing and some negative integer for opening in order to punish creating too many gaps.

Extending our DAG approach to incorporate suboptimal alignments. We previously stated that the graph $G_0(A,B)$ has the property that all its s-t paths are optimal paths in $G(A,B)$. This gives us a compact data structure able to store exponentially many optimal paths. We now generalise the subgraph G_0 to contain also suboptimal paths. This is achieved by introducing a critical threshold Δ, which users can specify to either extend or narrow down the suboptimal alignment space. For a given $\Delta \geq 0$, an s-t path in $G(A,B)$ is said to be Δ-*suboptimal* if it is of weight at least $OPT - \Delta$, where OPT denotes the weight of an optimal path. We denote paths that appear in at least an α-proportion of all Δ-suboptimal paths in $G(A,B)$ as (α,Δ)-*safe* and define maximal (α,Δ)-safe paths as (α,Δ)-*safety windows*.
Following the notion of Naor and Brutlag [1994], we define $G_\Delta(A,B) = (V_\Delta, E_\Delta)$ to be the subgraph of $G(A,B)$ which is induced by the set of edges $e \in E$ such that there is a Δ-suboptimal s-t path in $G(A,B)$ crossing e. By definition, $G_\Delta(A,B)$ contains all Δ-suboptimal s-t paths in $G(A,B)$. Additionally, however, it may contain s-t paths that are not Δ-suboptimal. This implies that some paths of $G_\Delta(A,B)$ appear in at least a proportion α of s-t paths which are not necessarily (α,Δ)-safe paths (in $G(A,B)$).
Naor and Brutlag [1994] have shown that $G_\Delta(A,B)$ is the *smallest* subgraph of $G(A,B)$ that contains all Δ-suboptimal s-t paths. In fact, our data spanning tree-of-life scale protein sequence diversity does not indicate that these spurious non-suboptimal paths have any effect in practice. Naor and Brutlag [1994] argue that the only possible solution to capture only suboptimal paths would be to naively enumerate over all suboptimal paths, which is unfeasible, since in the worst case there can be exponentially many suboptimal paths of size $G(A,B)$. We address this limitation by approximating (α,Δ)-safe paths with those paths of $G_\Delta(A,B)$ appearing in at least a proportion α within its set of s-t paths. We compute these paths according to Algorithm 1, with the only difference that we start with $G_\Delta(A,B)$ in line 1 instead of $G_0(A,B)$.

To compute $G_\Delta(A,B)$, we proceed as in [Naor and Brutlag, 1994]: For all nodes $v \in V$, we find the weight $w(v)$ of an optimal v-t path of size $G(A,B)$ in linear time by traversing the nodes in reverse topological order, such that the weights of all out-neighbours of v are computed when we reach v. Similarly, we compute the weight $w^r(v)$ of optimal s-v paths for all $v \in V$. An edge $e = (u,v) \in E$ is now part of $G_\Delta(A,B)$ only if $w^r(u) + w(v) + w(e) \geq OPT - \Delta$, where $w(e)$ denotes the weight of the edge e. We can thus construct $G_\Delta(A,B)$ in linear time and proportional to the size of $G(A,B)$.

Implementing Alignment-Safety into the scientific software EMERALD. We designed EMERALD to efficiently implement our theoretical methodology to explore the suboptimal alignment space by inferring alignment-safe intervals when performing pairwise protein sequence comparisons. In detail, EMERALD accepts a protein sequence cluster in FASTA format of k sequences as input and returns $k-1$ safety-window sets in a custom safety window format. Our tool aligns all sequences in the cluster against a user-selected representative sequence of the cluster, which by default is the first sequence in the FASTA file, thereby maximising the score of the alignment. EMERALD by default uses BLOSUM62 as substitution matrix, however, any affine-linear gap cost function can be used via the command line parameters. For a cluster containing k sequences, the goal is to be able to compare the safety-windows of all $k-1$ pairs with each other. Thanks to Lemma 1, we can project the safety windows to the sequences. Node indices in the alignment graph are of the form (i,j) with $0 \leq i \leq n$ and $0 \leq j \leq m$, and such safety windows can be written as $[(i_1,j_1),(i_2,j_2)]$. For example, given the strings $S = "AB"$ and $T = "BC"$ and the alignment $"AB-"$ and $"-BC"$, a safety window of the first gap would be of the form $[(0,0),(1,0)]$. Even though by usual convention $T[0] = B$, this position is not part of the safety window. EMERALD then returns the interval $[0,1]$ for string S and $[0,0]$ for string T. In other words, the tuple indices of the nodes are *between* the string characters. This coincides with the fact that the node set is the product $\{0,\ldots,n\} \times \{0,\ldots,m\}$ (i.e. $(n+1)\cdot(m+1)$ nodes) and with the fact that we want to be able to include gaps in safety windows.

Data Access

EMERALD's source code can be accessed at `https://github.com/algbio/emerald`. All data preprocessing is implemented as a reproducible and parametrised Snakemake pipeline [Mölder et al., 2021] and computationally reproducible scripts can be found at `https://github.com/algbio/emerald-analysis`. All datasets, including the 396k protein sequences, DIAMOND-clusters, EMERALD-output, and annotation files can be re-

bioRxiv preprint doi: https://doi.org/10.1101/2023.01.11.523286; this version posted January 15, 2023. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.

trieved from figshare: https://doi.org/10.6084/m9.figshare.21720299.

Competing interest statement

The authors declare no competing interests.

ACKNOWLEDGEMENTS
This work was partially funded by the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No. 851093, SAFEBIO), and partially by the Academy of Finland (grants No. 322595, 328877, 352821). We thank V. Alva, H. Ashkenazy, C. Behr, A. Dobbelstein, F. Gabler, Lukas Maischak, K. K. Ullrich and D. Weigel for their interest and helpful discussions. H.G.D. and B.B. thank Detlef Weigel for ongoing support and sponsorship of their work. Finally, H.G.D. and B.B. acknowledge that their work was supported by the Max Planck Society.

Author contributions:
A.T. and H.G.D. designed and led this study. An.G. implemented EMERALD with feedback from Ar.G., F.D., B.B., A.T., and H.G.D. Data analysis was performed by An.G., Ar.G., F.D., B.B. Data interpretation was performed by A.T., H.G.D., An.G., Ar.G., F.D., and B.B. The manuscript was written by H.G.D., An.G., and A.T., with contributions from Ar.G., F.D., B.B.

Bibliography

Minkyung Baek, Frank DiMaio, Ivan Anishchenko, Justas Dauparas, Sergey Ovchinnikov, Gyu Rie Lee, Jue Wang, Qian Cong, Lisa N. Kinch, R. Dustin Schaeffer, Claudia Millán, Hahnbeom Park, Carson Adams, Caleb R. Glassman, Andy DeGiovanni, Jose H. Pereira, Andria V. Rodrigues, Alberdina A. van Dijk, Ana C. Ebrecht, Diederik J. Opperman, Theo Sagmeister, Christoph Buhlheller, Tea Pavkov-Keller, Manoj K. Rathinaswamy, Udit Dalwadi, Calvin K. Yip, John E. Burke, K. Christopher Garcia, Nick V. Grishin, Paul D. Adams, Randy J. Read, and David Baker. Accurate prediction of protein structures and interactions using a three-track neural network. *Science*, 373(6557):871–876, 2021. doi: 10.1126/science.abj8754. URL https://www.science.org/doi/abs/10.1126/science.abj8754.

Amos Bairoch and Rolf Apweiler. The swiss-prot protein sequence data bank and its supplement trembl. *Nucleic acids research*, 25(1):31–36, 1997.

Cyril Banderier and Sylviane Schwer. Why delannoy numbers? *Journal of Statistical Planning and Inference*, 135(1):40–54, nov 2005. doi: 10.1016/j.jspi.2005.02.004. URL https://doi.org/10.1016%2Fj.jspi.2005.02.004.

Sarah E Bondos, A Keith Dunker, and Vladimir N Uversky. On the roles of intrinsically disordered proteins and regions in cell communication and signaling, 2021.

Benjamin Buchfink, Klaus Reuter, and Hajk-Georg Drost. Sensitive protein alignments at tree-of-life scale using diamond. *Nature methods*, 18(4):366–368, 2021.

Kun-Mao Chao, Ross C. Hardison, and Webb Miller. Locating well-conserved regions within a pairwise alignment. *Bioinformatics*, 9(4):387–396, 08 1993. ISSN 1367-4803. doi: 10.1093/bioinformatics/9.4.387. URL https://doi.org/10.1093/bioinformatics/9.4.387.

Hao Chen and Daisuke Kihara. Estimating quality of template-based protein models by alignment stability. *Proteins: Structure, Function, and Bioinformatics*, 71(3):1255–1274, 2008.

Hao Chen and Daisuke Kihara. Effect of using suboptimal alignments in template-based protein structure prediction. *Proteins: Structure, Function, and Bioinformatics*, 79(1):315–334, 2011.

Melissa Cline, Richard Hughey, and Kevin Karplus. Predicting reliable regions in protein sequence alignments. *Bioinformatics*, 18(2):306–314, 2002.

The UniProt Consortium. Uniprot: the universal protein knowledgebase in 2021. *Nucleic acids research*, 49(D1):D480–D489, 2021.

Dmitrij Frishman and Patrick Argos. Knowledge-based protein secondary structure assignment. *Proteins: Structure, Function, and Bioinformatics*, 23(4):566–579, 1995.

Trevor Hastie, Robert Tibshirani, Jerome H Friedman, and Jerome H Friedman. *The elements of statistical learning: data mining, inference, and prediction*, volume 2. Springer, 2009.

Steven Henikoff and Jorja G Henikoff. Amino acid substitution matrices from protein blocks. *Proceedings of the National Academy of Sciences*, 89(22):10915–10919, 1992.

Lukasz Jaroszewski, Weizhong Li, and Adam Godzik. In search for more accurate alignments in the twilight zone. *Protein Science*, 11(7):1702–1713, 2002.

John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, et al. Highly accurate protein structure prediction with alphafold. *Nature*, 596(7873):583–589, 2021.

Shahbaz Khan, Milla Kortelainen, Manuel Cáceres, Lucia Williams, and Alexandru I. Tomescu. Improving rna assembly via safety and completeness in flow decompositions. *Journal of Computational Biology*, 29(12):1–18, 2022. doi: 10.1089/cmb.2022.0261. URL https://doi.org/10.1089/cmb.2022.0261.

Niko Kiirala, Leena Salmela, and Alexandru I. Tomescu. Safe and complete algorithms for dynamic programming problems, with an application to RNA folding. In Nadia Pisanti and Solon P. Pissis, editors, *30th Annual Symposium on Combinatorial Pattern Matching, CPM 2019, June 18-20, 2019, Pisa, Italy*, volume 128 of *LIPIcs*, pages 8:1–8:16. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019. doi: 10.4230/LIPIcs.CPM.2019.8. URL https://doi.org/10.4230/LIPIcs.CPM.2019.8.

Maik Kschischo and Michael Lässig. Finite-temperature sequence alignment. In *Biocomputing 2000*, pages 624–635. World Scientific, 1999.

Eli Levy Karin, Haim Ashkenazy, Jotun Hein, and Tal Pupko. A simulation-based approach to statistical alignment. *Systematic Biology*, 68(2):252–266, 2019.

Zeming Lin, Halil Akin, Roshan Rao, Brian Hie, Zhongkai Zhu, Wenting Lu, Nikita Smetanin, Robert Verkuil, Ori Kabeli, Yaniv Shmueli, et al. Evolutionary-scale prediction of atomic level protein structure with a language model. *bioRxiv*, 2022.

Felix Mölder, Kim Philipp Jablonski, Brice Letcher, Michael B Hall, Christopher H Tomkins-Tinch, Vanessa Sochat, Jan Forster, Soohyun Lee, Sven O Twardziok, Alexander Kanitz, et al. Sustainable data analysis with snakemake. *F1000Research*, 10, 2021.

Eugene W. Myers and Webb Miller. Optimal alignments in linear space. *Computer applications in the biosciences : CABIOS*, 4 1, 1988.

Dalit Naor and Douglas L Brutlag. On near-optimal alignments of biological sequences. *Journal of Computational Biology*, 1(4):349–366, 1994.

Saul B Needleman and Christian D Wunsch. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of molecular biology*, 48(3): 443–453, 1970.

Osnat Penn, Eyal Privman, Giddy Landan, Dan Graur, and Tal Pupko. An alignment confidence score capturing robustness to guide tree uncertainty. *Molecular biology and evolution*, 27(8): 1759–1767, 2010.

Vincent Ranwez and Nathalie Chantret. Strengths and limits of multiple sequence alignment and filtering methods, 2020.

Burkhard Rost. Twilight zone of protein sequence alignments. *Protein engineering*, 12(2):85–94, 1999.

Maximilian Schlosshauer and Mattias Ohlsson. A novel approach to local reliability of sequence alignments. *Bioinformatics*, 18(6):847–854, 2002.

Michael L Sierk, Michael E Smoot, Ellen J Bass, and William R Pearson. Improving pairwise sequence alignment accuracy using near-optimal protein sequence alignments. *BMC bioinformatics*, 11(1):1–15, 2010.

Temple F Smith and Michael S Waterman. Identification of common molecular subsequences. *Journal of molecular biology*, 147(1):195–197, 1981.

Robert Endre Tarjan. A note on finding the bridges of a graph. *Inf. Process. Lett.*, 2:160–161, 1974.

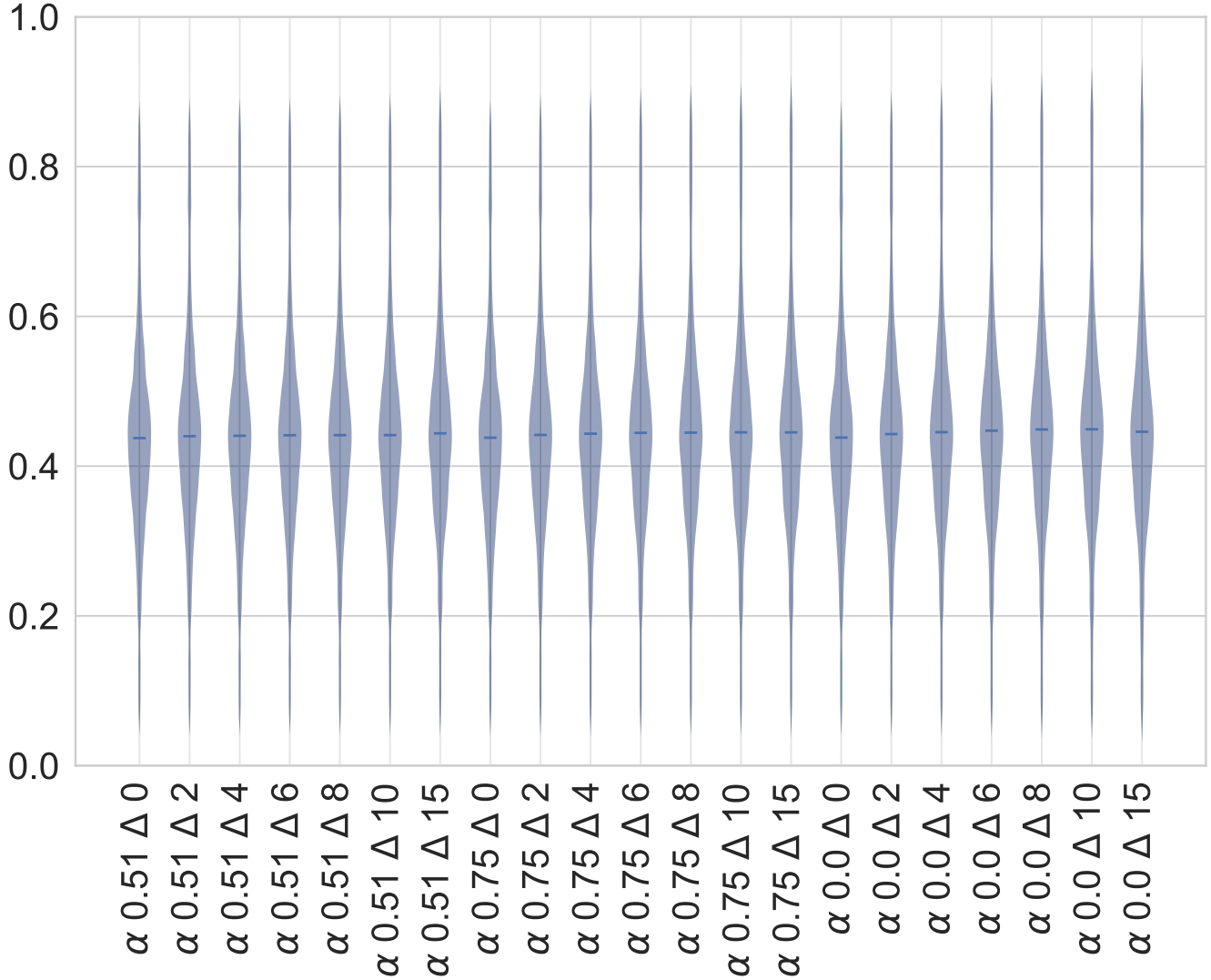
Alexandru I. Tomescu and Paul Medvedev. Safe and complete contig assembly through omnitigs. *J. Comput. Biol.*, 24(6):590–602, 2017. doi: 10.1089/cmb.2016.0141. URL https://doi.org/10.1089/cmb.2016.0141.

Mihaly Varadi, Stephen Anyango, Mandar Deshpande, Sreenath Nair, Cindy Natassia, Galabina Yordanova, David Yuan, Oana Stroe, Gemma Wood, Agata Laydon, et al. Alphafold protein structure database: Massively expanding the structural coverage of protein-sequence space with high-accuracy models. *Nucleic acids research*, 50(D1):D439–D444, 2022.

Martin Vingron and Patrick Argos. Determination of reliable regions in protein sequence alignments. *Protein Engineering, Design and Selection*, 3(7):565–569, 1990.

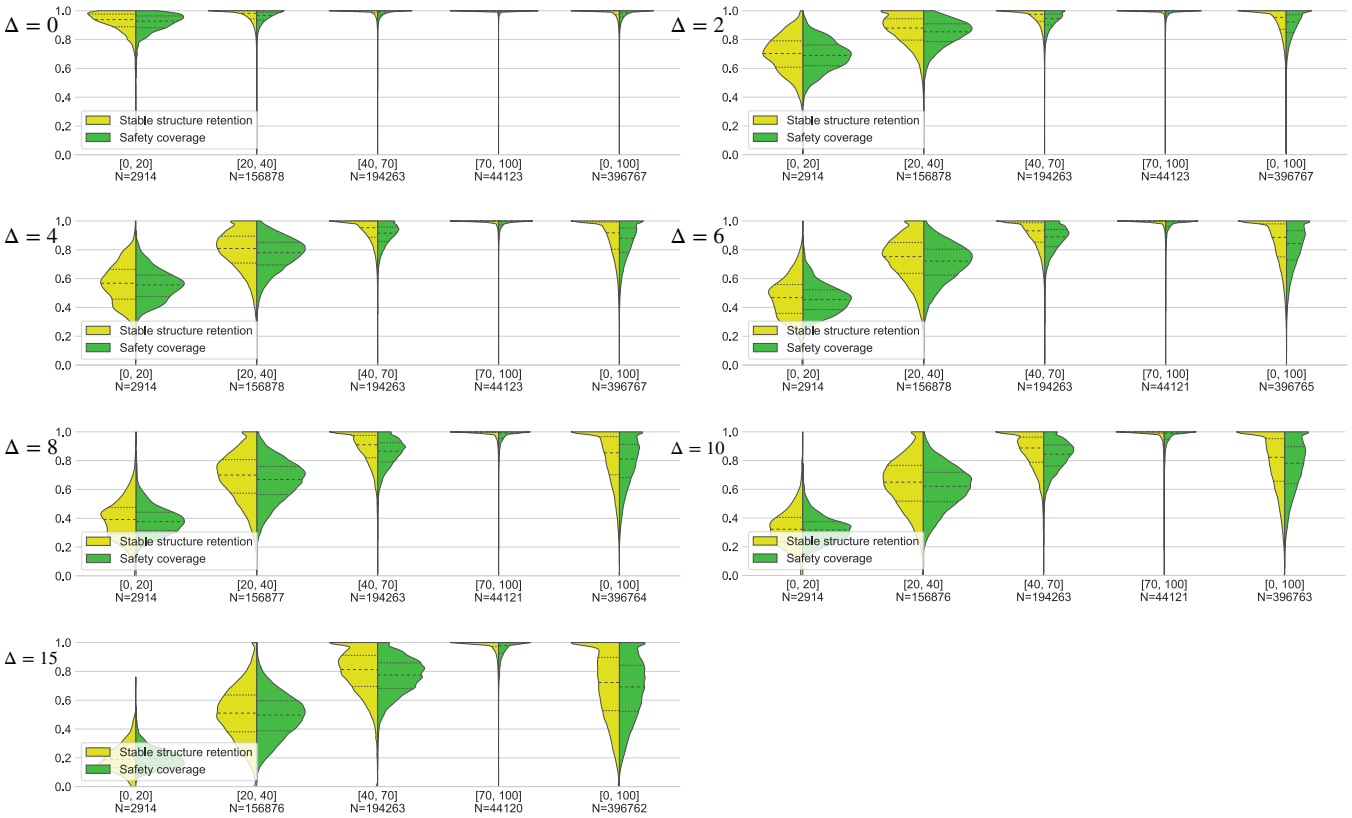
MQ Zhang and TG Marr. Alignment of molecular sequences seen as random path analysis. *Journal of theoretical Biology*, 174(2):119–129, 1995.

bioRxiv preprint doi: https://doi.org/10.1101/2023.01.11.523286; this version posted January 15, 2023. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.



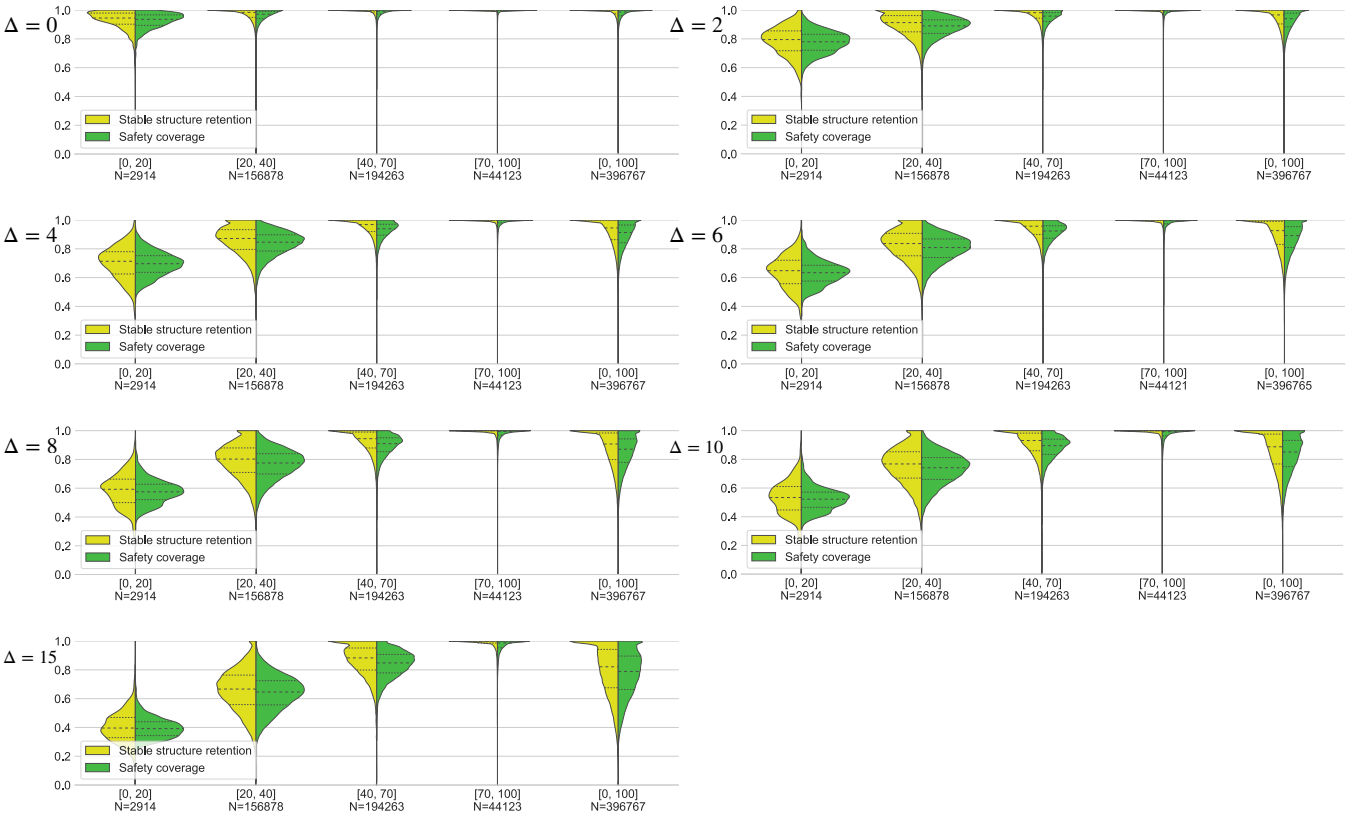
Supplementary Figure 1: Stable structure overlap for each pair of $\alpha = 0.51, 0.75, 1$ and $\Delta = 0, 2, 4, 6, 8, 10, 15$ on the whole dataset of 396k sequences. Some 16 pairs of aligned sequences had to be filtered out due to a safety coverage of 0. The stable structure overlap is throughout all pairs constant with the median at 43%. This is exactly the proportion of stable positions within α-safety.

bioRxiv preprint doi: https://doi.org/10.1101/2023.01.11.523286; this version posted January 15, 2023. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.



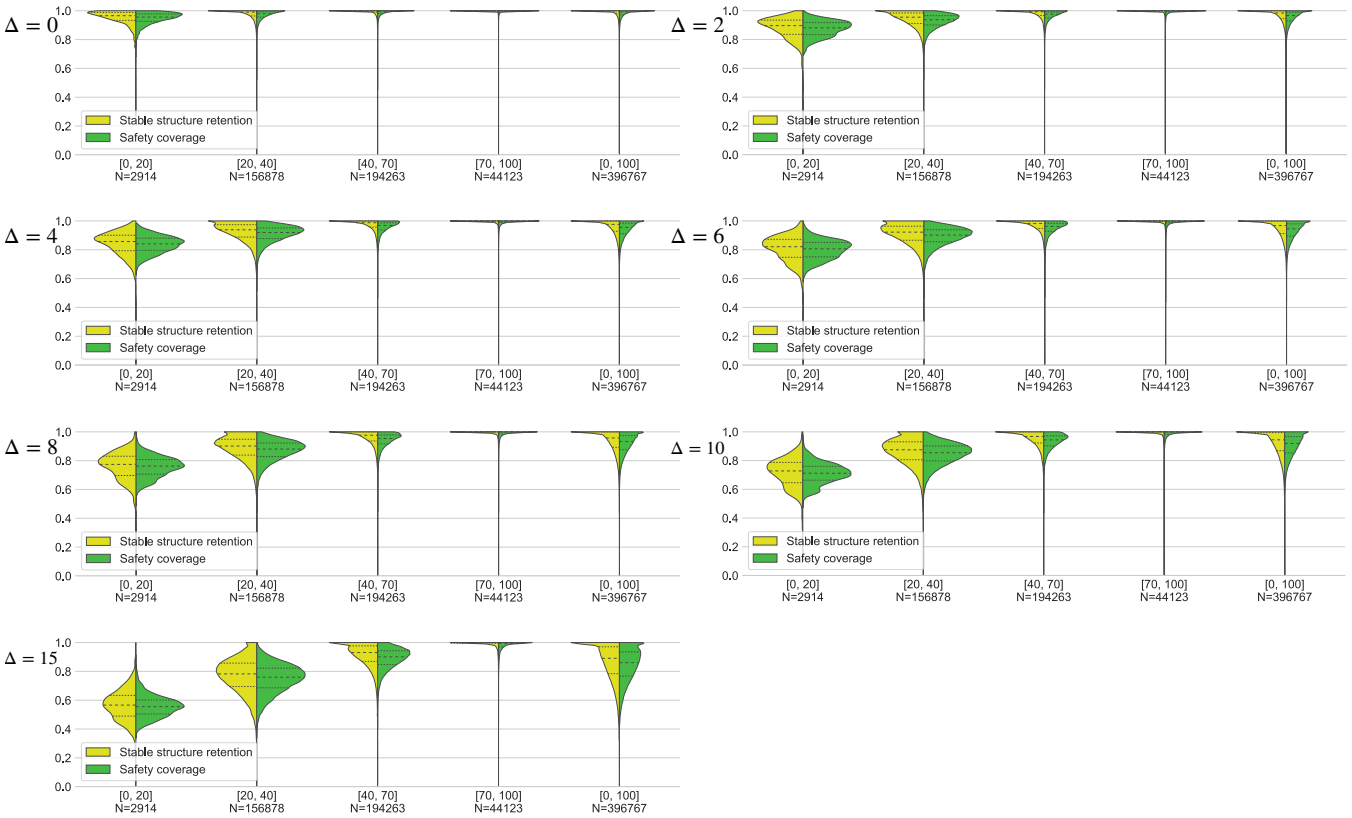
Supplementary Figure 2: Stable structure retention compared to safety coverage for $\alpha = 1$, $\Delta = 0, 2, 4, 6, 8, 10, 15$ and several identity ranges. The median and the quartiles are indicated in black.

bioRxiv preprint doi: https://doi.org/10.1101/2023.01.11.523286; this version posted January 15, 2023. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.



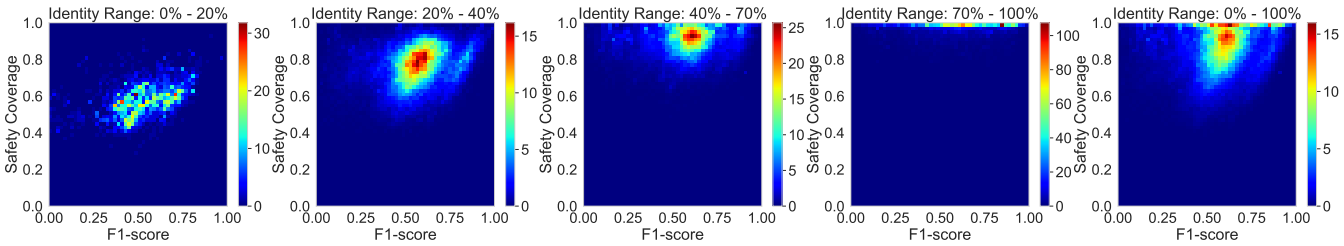
Supplementary Figure 3: Stable structure retention compared to safety coverage for $\alpha = 0.75$, $\Delta = 0, 2, 4, 6, 8, 10, 15$ and several identity ranges. The median and the quartiles are indicated in black.

bioRxiv preprint doi: https://doi.org/10.1101/2023.01.11.523286; this version posted January 15, 2023. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.



Supplementary Figure 4: Stable structure retention compared to safety coverage for $\alpha = 0.51$, $\Delta = 0, 2, 4, 6, 8, 10, 15$ and several identity ranges. The median and the quartiles are indicated in black.

bioRxiv preprint doi: https://doi.org/10.1101/2023.01.11.523286; this version posted January 15, 2023. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.



Supplementary Figure 5: F1-score for $\alpha = 0.75$, $\Delta = 8$, for sequences in clusters of varying identity ranges. We can see that the F1-score lies consistently in a range around 50%, while in the lower cluster identities the sequence is covered less by safety windows. This indicates that EMERALD performs better in the identity range 20-70%.