nature portfolio

Peer Review File

SARST2 high-throughput and resource-efficient protein structure alignment against massive databases

Corresponding Author: Professor Wei-Cheng Lo

This file contains all reviewer reports in order by version, followed by all author rebuttals in order by version.

Version 0:

Reviewer comments:

Reviewer #1

(Remarks to the Author)
The authors designed and develop SARST2, a fast protein structure alignment algorithm for searching large databases. Efficient structure alignment is an important topic. SARST2 converts protein structures into sequences of structural elements and compare structures using sequence alignments. The authors include multiple innovations to increase the speed and use Golang, a parallel computing language, to implement their algorithm. The manuscript is well-written. The software is easy to use and fast.
I have some minor comments:
1. They did not mention Foldseek, which is a similar software:
Van Kempen, Michel, et al. "Fast and accurate protein structure search with Foldseek." Nature Biotechnology 42.2 (2024): 243-246.
I think commenting the difference with FoldSeek or comparing with it is important.
2. The authors may want to also evaluate the alignment quality.

(Remarks on code availability)
The software is easy to use and fast.
1. This software only returns hits when the SEQRES records are preserved. The sequence information should be able to extracted from the structure itself.
2. It seems that the software can only return up to 500 hits at this moment.
3. Do the authors plan to regularly update the precompiled databases?

Reviewer #2

(Remarks to the Author)
This is a very interesting paper that describes a very effective and fast structure alignment algorithm. The work is on the whole well done and the results are quite impressive. However, there are a number of minor points that should be addressed prior to publication:

1. Precision/Recall are calculated whether the retrieved proteins are on the SCOP hit list. There are a number of problems with this: First of all, in the limit of lower sequence identity, some proteins have a low TM-score and obviously have a different fold than the majority members of the SCOP family, yet they are assigned to the same family in SCOP. what is the minimum sequence identity to assign a protein to a given SCOP family?In that regard, what TM-score cutoff is used to say that a protein is related to SCOP family members or not? Thus, some proteins should not be on in the SCOP list. Since this method first filters by sequence and SSE similarity, it might recover low homology proteins which actually have a different fold. Thus, another metric which should be used is the absolute value of the TM-score. Better structural alignments should have a higher TM-score of that is the metric of truth (of course different metrics will yield different results).

2. The TM-scores from SARST2 should be compared to those from TM-align in benchmarking cases. In that regard, there is

a better version of TM-align, fr-TM-align that often gives a higher TM-score than the original TM-align algorithm. Zhang has also implemented an improved version. Both are available from the Skolnick and Zhang websites. Is the latest version of TM-align being used?

Recommendation: Publish after minor revision.

(Remarks on code availability)

Reviewer #3

(Remarks to the Author)
authors update their SARST algorithm for protein structure alignment, increasing its speed and accuracy to the point that it could be effectively used to search through structural databases of millions of structures, such as now being available with AI structure prediction algorithms. While potentially interesting, the paper has a fatal flaw of not being compared to FoldSeek (PMID: 37156916), current king of the mountain in fast protein structure searched. Published almost a year ago (May 2023) it just cannot be ignored.

In addition, there are few other issues with the paper:
- authors often compare their program to BLAST or (sometimes) to PSI-BLAST. This comparison is both confusing and unfair. Confusing because they perform different tasks and run on different databases. Unfair, because the times given by the authors do not correspond to this reviewer experience. Searching a single sequence against a UniRef90 doesnt take "couple of hours", speed up with using a parallel version is almost linear with the number of processors. It is hard to avoid an impression that authors, perhaps unintentionally, use less than optimal options for BLAST, while spending a lot of time on optimizing their own program.
- authors do not provide much information about the alignment accuracy of their algorithm, especially on distant homologs or even analogs, those that are not detected by sequence methods. it would be interesting to know how many structurally similar proteins are being identified among those that cannot be seen by BLAST.
- critical information seems to be missing. Authors use two datasets, Qry400 and Qry200 for training and validating their algorithms, but I couldnt find their lists in the supplementatary material, which is not well described. Are these sets in the files 491475_0_data_set_686913_s93113.txt, 491475_0_data_set_686912_s93jj2.txt and 491475_0_data_set_686911_s93vv2.txt? there are no explanations in the package (or at least I couldnt find any)

(Remarks on code availability)
as far as I can tell, no code is available, neither in the review materials nor from the website. Only binary executables for several machines are included, so I cannot really evaluate the code.

Version 1:

Reviewer comments:

Reviewer #1

(Remarks to the Author)
The authors have addressed my questions.

A minor point:

- Please spell out SIMD and DP.

I have also evaluated the responses to Reviewer #3 and my related comments are as follows:
1. The foldseek comparison is done as it is also requested by me.

2. The abstract may want to mention foldseek also to strengthen the impact of this paper.

3. "may take an hour with one typical CPU (estimated by Table 2 of Ref3". Please be specific about what is one typical CPU.
4. Reviewer #3 asked about the remote homologs. The authors responded with the sequence identity: "For instance, the point on the far left of Fig 5a indicates that SARST2 reported a ~9% sequence identity for proteins receiving a 0–1% identity computed by BLAST." I am not sure about this. 9% sequence identity may mean nothing as it is too low any way. They may want to compare on AF3 structure similarities.

5. About the source code, I cannot open the links that they provided. Not sure it is because my hotel internet is terrible or not... Even if it is available, I do not think I will have time to evaluate the source code. So you may still consider another reviewer to check this.

6. Their GitHub page: https://github.com/NYCU-10lab/sarst seems missing lots of basic information… Basic instructions of usage are needed. It will be helpful to provide a test example. Add the reference to this paper will help people to cite it and understand its details. Foldseek GitHub serves as a good example. Somehow, google search cannot find this GitHub easily…

(Remarks on code availability)

Reviewer #2

(Remarks to the Author)
This revised manuscript has addressed all my previous concerns in a very comprehensive and thoughtful way. The resulting improvements further confirm my original opinion that this work definitely should be published in Nature Communications.

(Remarks on code availability)

Version 2:

Reviewer comments:

Reviewer #1

(Remarks to the Author)
The authors have addressed all my comments.

(Remarks on code availability)

NCOMMS-24-13170A

SARST2, a high-throughput protein structure alignment algorithm for searching massive databases

Dear Professors and Editor,

We would like to thank the anonymous reviewers for their careful reading and detailed comments, which have greatly enhanced this article. We are pleased that you find sufficient merit in this work to ask for an appropriately revised manuscript. We also feel very sorry for keeping you waiting so long. Some of the experiments were costly in terms of time and computation, and hardware damages caused by typhoons also dragged the revision. The manuscript, where newly added and extensively edited contents are marked with a light green background, has been modified according to the referees' comments, which are also answered as follows.

REVIEWER COMMENTS

Reviewer #1

Remarks to the Author:

The authors designed and develop SARST2, a fast protein structure alignment algorithm for searching large databases. Efficient structure alignment is an important topic. SARST2 converts protein structures into sequences of structural elements and compare structures using sequence alignments. The authors include multiple innovations to increase the speed and use Golang, a parallel computing language, to implement their algorithm. The manuscript is well-written. The software is easy to use and fast.

I have some minor comments:

1. They did not mention Foldseek, which is a similar software:

Van Kempen, Michel, et al. "Fast and accurate protein structure search with Foldseek." Nature Biotechnology 42.2 (2024): 243-246.

I think commenting the difference with FoldSeek or comparing with it is important.

Response:

We sincerely thank the reviewer for the positive comments and valuable suggestions, which greatly help us improve this work. This project started in 2021 when our preliminary literature survey showed that no structure alignment algorithm could

efficiently handle a massive database like the AlphaFold DB. Since then, we worked so hard to develop SARST2 and paid decreased attention to the survey of related works.

After reviewing Foldseek, we realized that, due to the delicate application of deep learning and acceleration by SIMD instructions, it might also be so efficient that its speed excelled a conventional sequence alignment search with BLAST. In addition to introducing Foldseek on Page 4 (Lines 101–107), we compared SARST2 with it in all major experiments (Figs. 2–5 and Supplementary Tables 1,2,7). The Subsection "Performance tests with the AlphaFold Protein Structure Database" has been extensively revised, with new experiments on SARST2 and Foldseek.

As shown in Fig. 2 (Page 6), assessed with the SCOP-2.07 dataset, the precision of SARST2 was 96.3%, slightly higher than Foldseek (95.9%). Fig. 3 and related main text on Pages 7–8 showed that using one CPU, SARST2 ran 1.4–3.3 times faster than Foldseek (Fig. 3a). When they both used 32 CPUs, SARST2 ran 7.1–10.5 times faster (Fig. 3b,c). The speed enhancement of SARST2 by parallel computation was 10.6 folds, higher than that of Foldseek (2.1–3.0 folds). The main text on Pages 11 (Line 345) – 13 (Line 384) and Fig. 4 demonstrate that when searching the AlphaFold DB with 32 Intel i9 CPUs, SARST2 cost 3.4 min, lower than Foldseek (18.6 min). Moreover, the memory and disk space requirements of SARST2 were several folds less than those of Foldseek.

In this work, we compared SARST2 with BLAST, partly trying to indicate that SARST2 broke the knowledge that structural alignments are slower than sequence alignments. In this regard, Foldseek was also a champion, for its speed and memory usage outperformed BLAST significantly. If the innovations of SARST2 and Foldseek can be combined through collaboration, structural alignment methods with superior efficiency would be created, bringing more benefits to biological and medical sciences.

2. The authors may want to also evaluate the alignment quality.

Response:

The reviewer is acknowledged for this constructive suggestion. SARST2 was defined as a search algorithm. Indeed, during the development, we focused more on its ability to distinguish the relative structural similarities among proteins than its absolute ability to align structures. Nevertheless, we are very interested in whether its alignment quality is comparable to state-of-the-art methods. Taking this opportunity, we made all-against-all pairwise alignments between the Qry400 dataset and all SCOP family-level homologs of those query proteins. To analyze the alignment quality of SARST2, we grouped the pairwise data according to the sequence identity computed by BLAST. Homolog pairs with BLAST identities in the range [x %, x+1 %) were collected in group x%. For every

group, SARST2, Foldseek, and TM-align were applied to align and compute the average sequence identities between homologs. Fig. 5a and the main text on Pages 14–15 indicate that the alignments by SARST2, Foldseek, and TM-align are better than those by BLAST at detecting the homology between far-related proteins. For proteins in identity groups <21%, the identities computed by SARST2 and Foldseek were significantly higher than those by BLAST and slightly higher than those by TM-align.

The distribution of pairwise sequence identities computed by BLAST, SARST2, Foldseek, and TM-align were also observed. As illustrated in Fig. 5b and the main text in Lines 424–431 on Page 15, the identity distribution of BLAST showed a peak at ~10%, while the peaks of TM-align, Foldseek, and SARST2 were around 17%, 21%, and 22%, respectively. On the plot, the distribution of TM-align went more to the right (the high identity direction) than BLAST. Foldseek went farther than TM-align, and SARST2 went the farthest, showing that SARST2 alignment was good at detecting the homology between proteins. For homologs with BLAST identities ≤ 30%, the Pearson correlation coefficient between the identities computed by SARST2 and those by BLAST was only 0.693. The coefficients between SARST2 and Foldseek and between SARST2 and TM-align were 0.859 and 0.882, indicating their comparability in alignment quality.

Many previous works demonstrated alignment quality through case studies on far-related proteins. Since SARST2 is a database search algorithm, we tried to assess the alignment quality with a large-scale approach. We hope these all-against-all pairwise alignment results are sufficient to show that SARST2 produces high-quality alignments when compared to state-of-the-art structure alignment algorithms.

Remarks on code availability:

The software is easy to use and fast.

1. This software only returns hits when the SEQRES records are preserved. The sequence information should be able to extracted from the structure itself.

Response:

We thank the reviewer for the comment, which reminded us that we should describe how the information was obtained from the structural files. Both the previous and current versions of the SARST2 programs extracted sequence information from the structure, just as the reviewer indicated. By decompressing Supplementary Software 1 from our previous submission, users can see that most of the demo structural files in the “dbs” folder contained only ATOM descriptions. No matter whether they contain SEQRES records, those files would produce the same results because they were processed merely based on the ATOM lines. For example, 101m.pdb (with SEQRES)

and 101mA.pdb (no SEQRES) would lead to the same alignments with 105mA.pdb. To help readers understand how we extract structural information, we have added a paragraph at the end of Subsection "Sources of experimental datasets" on Page 17.

2. It seems that the software can only return up to 500 hits at this moment.

Response:

We thank the reviewer for reminding us that the user experience of SARST2 could be improved. The hit list size is 500 by default but can be adjusted with the -brief and -detail arguments. When the program was executed without any arguments, a help message would describe the usage, including -brief ([int] Number of subject structures to show one-line summaries) and -detail ([int] Number of subject structures to show detailed alignment data). In the current submission, when the hit list is cut to 500 hits because the user does not set their own -brief and -detail values, the SARST2 program will show a clue at the end of the output to inform about those arguments.

3. Do the authors plan to regularly update the precompiled databases?

Response:

Sincerely, we thank the reviewer again for helping us improve this work. Yes, we will update the databases regularly. We will release them on our website and GitHub site. The precompiled PDB databases can be updated quarterly, and SCOP can be updated soon after new releases. As for the AlphaFold DB, Due to our limited resources, downloading an official release takes months, and we do not have sufficient space to store the raw files for the next release. We may make a pipeline to download and compile files simultaneously. The raw files can be discarded after compilation, and our small storage device shall hence be able to handle the regular AlphaFold updates.

Reviewer #2

Remarks to the Author:

This is a very interesting paper that describes a very effective and fast structure alignment algorithm. The work is on the whole well done and the results are quite impressive. However, there are a number of minor points that should be addressed prior to publication:

1. Precision/Recall are calculated whether the retrieved proteins are on the SCOP hit list. There are a number of problems with this: First of all, in the limit of lower sequence identity, some proteins have a low TM-score and obviously have a different fold than the majority members of the SCOP family, yet they are assigned to the same family in SCOP. what is the minimum sequence identity to assign a protein to a given SCOP family?

In that regard, what TM-score cutoff is used to say that a protein is related to SCOP family members or not?

Response:

We are very grateful to the reviewer for the positive comments and interesting questions, which stimulated us to learn more about our source materials and do more experiments to improve the quality of this work.

After reviewing SCOP's papers and websites, we found no specific minimum sequence identity threshold to assign a protein to a given family. SCOP specialists seemed to consider many factors other than sequence identity in manual inspection. For instance, whether the functional residues of a protein were well aligned in multiple alignments with its candidate family members. Sometimes, local sequence identities surrounding the functional sites might influence the classification more than global identities. Just like the reviewer mentioned, we knew that a protein visually having a different fold from some proteins might be assigned to the same family. Recently, we inspected many SCOP family pairs with visually low structure similarities. In most cases, they still had some common regions well superimposable. For example, d1m64a seems dissimilar from d1qdbc and d1fs9a. However, if we look at their core regions (residues 1–102 in d1m64a, 164–305 in d1qdbc, and 165–300 in d1fs9a), d1m64a is still well superimposed with these members of SCOP a.138.1.3.

Although knowing the (perhaps variable) identity thresholds SCOP specialists used would be difficult, we might draw a general picture. We have done all-against-all pairwise alignments between the Qry400 and SCOP-2.07 datasets. By extracting the alignments between homologs in the same family, we draw a plot of the sequence identity distributions computed by BLAST, SARST2, Foldseek, TM-align, and fr-TM-align (Fig. 5b). It shows that most SCOP family-level homologs had pairwise sequence identities >10% as aligned by SARST2, Foldseek, TM-align, and fr-TM-align. BLAST produced lower sequence identities, but most homologs were still in the >10% region. The main disagreements between these methods occurred in the ≤10% identity region, which might be the twilight zone where manual inspectors should pay more attention while assigning family classifications.

The distribution of TM-scores computed by SARST2 was also analyzed, and the results will be explained in the second part. We would first like to thank the reviewer for asking about the TM-score cutoff for detecting SCOP family-level homology. The SCOP team gave no clue about whether or how they used TM-score. However, we felt interested in whether the SARST2 TM-score could be applied to distinguishing proteins with different levels of homology.

We observed the difference in distribution between the TM-scores computed with SCOP family-level and superfamily-level homologs. Extended Data Fig. 3 shows that ~80% of the family-level homologs received a SARST2 TM-score ≥0.70. Meanwhile, only ~3% of the superfamily-level homologs received a SARST2 TM-score ≥0.70. Thus, a TM-score ≥0.7 might imply SCOP family-level homology between proteins. Related information has been added on Page 22 (Lines 692–703). A statement about this cutoff has been added to the manual of the -tmcut argument of SARST2.

Thus, some proteins should not be on in the SCOP list. Since this method first filters by sequence and SSE similarity, it might recover low homology proteins which actually have a different fold. Thus, another metric which should be used is the absolute value of the TM-score. Better structural alignments should have a higher TM-score of that is the metric of truth (of course different metrics will yield different results).

Response:

We thank the reviewer for this constructive suggestion, which bettered our software. SARST2 used the confidence score (equation (7)) to rank hits. The confidence score was the product of 1) the alignment score by dynamic programming, 2) the machine learning-generated probability score of a hit being a SCOP family-level homolog of the query protein, and 3) the TM-score. Just like the reviewer said, the absolute value of the TM-score can be applied. Other measures like sequence identity and RMSD may also be helpful in various situations.

Using the TM-score as the ranking metric, the information retrieval (IR) precision of SARST2 was 96.2%, close to that of the confidence score (96.3%). In our previous submission, the SARST2 program did not provide any argument to let the user decide how to rank the hits. Now, a new argument -orderby is added. Beyond the default, the user can change it to the TM-score, sequence identity, or RMSD.

The questions about SCOP and the feasibility of the hit ranking metric raised by the reviewer stimulated us to make a more challenging test for SARST2. CATH is another hierarchical structure database with a classification schema quite different from SCOP. We downloaded the 40% sequence identity non-redundant subset of CATH

(nrCATH40), which contained 31,885 proteins from 6,576 homologous superfamilies. Because of its difference from SCOP and the low homology between its proteins, nrCATH40 should be a good independent and challenging test dataset for SARST2, the machine learning model of which was trained with SCOP datasets.

From each of the three major CATH classes, we randomly selected 40 homologous superfamilies containing ≥20 proteins from nrCATH40 to be the query dataset, Qry120-CATH. An IR experiment showed that the precision of SARST2 lowered to 72%, and so did Foldseek and TM-align. The precision of BLAST sharply decreased to 26%. SARST2 exhibited compatible precision with state-of-the-art structural alignment methods in the face of such a challenging dataset, supporting the feasibility of the proposed confidence score as the hit-list sorting measure.

The revised manuscript describes the nrCATH40 experiment on Page 25 (Lines 794–814). Detailed results are exhibited in Supplementary Table 7. The entries of nrCATH40 and Qry120-CATH datasets are listed in Supplementary Data 4,5.

2. The TM-scores from SARST2 should be compared to those from TM-align in benchmarking cases. In that regard, there is a better version of TM-align, fr-TM-align that often gives a higher TM-score than the original TM-align algorithm. Zhang has also implemented an improved version. Both are available from the Skolnick and Zhang websites. Is the latest version of TM-align being used?

Response:

Thanks to the reviewer's suggestion, the distribution of TM-scores of SARST2 was also analyzed using the pairwise data mentioned above. TM-align and fr-TM-align were applied in this test, too. The results are summarized in the newly added Extended Data Fig. 3, and the related main text is on Page 22 (Lines 692–703).

The TM-align we used was not the latest but the v.20160521 Fortran edition. Although the Fortran edition was less versatile than the latter C++ versions (e.g., functional supports for mmCIF files, PyMOL, and US-align), our preliminary test showed they produced the same results for our SCOP query datasets. Notably, perhaps due to its simplicity, it ran slightly faster than the C++ versions. Since this work emphasized efficiency rather than functional versatility, we chose the Fortran edition.

Fr-TM-align was comparable to TM-align in IR precision (Fig. 2) but ran slower (Fig. 3). As the reviewer mentioned, after the fr-TM-align was published (in 2008), its TM-scoring scheme was implemented in the improved TM-align. The TM-align version we used was released in 2016; therefore, as shown in Extended Data Fig. 3, the TM-score distributions of fr-TM-align and TM-align were nearly identical.

As the figure shows, the TM-scores computed by SARST2 were slightly lower than those by TM-align and fr-TM-align. Nonetheless, their distributions were similar, and the Pearson correlation coefficients between SARST2 and TM-align and fr-TM-align were both 0.973.

Recommendation: Publish after minor revision.

Response:

We feel so grateful to the reviewer for instructing us to improve this work. TM-score played a key role in the accuracy of SARST2. The TM-score algorithm we implemented was adopted from TM-align v2016. Our implementation might not be the most efficient because the calculation was made with multiple loops. We will improve the TM-score code in the latter version, hoping that the overall speed of SARST2 can be enhanced.

Reviewer #3

Remarks to the Author:

authors update their SARST algorithm for protein structure alignment, increasing its speed and accuracy to the point that it could be effectively used to search through structural databases of millions of structures, such as now being available with AI structure prediction algorithms. While potentially interesting, the paper has a fatal flaw of not being compared to FoldSeek (PMID: 37156916), current king of the mountain in fast protein structure searched. Published almost a year ago (May 2023) it just cannot be ignored.

Response:

We would like to express our great acknowledgment to the reviewer for pointing out the weaknesses of our work. The work started in 2021 as Google-DeepMind implied their AlphaFold Database project. At that time, we did not find any structure alignment algorithm capable of efficiently handling millions of structures. After that, we worked hard on developing SARST2 and paid less attention to the literature survey. We did not know Foldseek before the previous submission. We sincerely thank the reviewer for letting us realize that keeping updated with the literature is what we always need to do.

After reviewing Foldseek, we admired so much its delicate application of deep learning to structure alignment and its advances in accelerating dynamic programming (DP) with SIMD instructions. We planned to use SIMD as the SARST2 project initiated. It was paused because managing those instructions with Golang was inconvenient, and we needed to design new DP procedures. We thank the reviewer for helping us learn

from Foldseek, in which brilliant SIMD DP algorithms have been proven feasible for structure alignment. Due to the complexity of managing SIMD with Golang, we could not implement them in the current SARST2. Nevertheless, we will do our best to use modern parallel computation technologies in the future.

In the revised **Introduction**, we have summarized the advances of Foldseek on Page 4 (Lines 101–107). Many experiments have been updated with Foldseek (Figs. 2–5 and Supplementary Tables 1,2,7). The Subsection "Performance tests with the AlphaFold Protein Structure Database" (Pages 11–13) was renewed with the database searching and formatting efficiency of SARST2 and Foldseek.

Fig. 2 exhibits that, assessed with the SCOP-2.07 dataset, the precision of SARST2 was 96.3%, slightly higher than Foldseek (95.9%). Two types of speed tests were described in this work, without parallelization (one CPU) or with parallelization (multiple CPUs). Using one CPU, SARST2 ran 1.4–3.3 times faster than Foldseek, depending on whether the clustered database function of Foldseek was enabled. With 32 Intel i9 CPUs, SARST2's speed enhancement was 10.6 folds, while Foldseek's was 2.1–3.0 folds. These results are described in Fig. 3 and the main text on Pages 7–8.

SARST2 also ran efficiently for larger datasets, as AlphaFold DB subsets with increasing sizes were applied to assess it. Searching the full AlphaFold DB with 32 i9 CPUs, the time cost by SARST2 was 3.4 min, lower than that of Foldseek (18.6 min). Moreover, the memory and disk space consumption of SARST2 were also lower. These large-scale tests are described in Fig. 4 and Pages 11–13.

In response to the reviewer, we made comparisons with the Foldseek. However, emphasis must be placed on the fact that we have no intention to compete with it or any other methods. We are a small, hardworking lab devoted to developing structural biology algorithms that we wish to help move biology and medical sciences forward. We learned so much from those excellent algorithms tested in this work. Without the light the authors have pointed ahead for us, we would struggle to make any progress.

In addition, there are few other issues with the paper:

- authors often compare their program to BLAST or (sometimes) to PSI-BLAST. This comparison is both confusing and unfair. Confusing because they perform different tasks and run on different databases.

Response:

The reviewer is thanked for reminding us that comparisons should focus more on structural alignment methods. SARST2 was compared with BLAST mainly because its previous versions, the SARST and *i*SARST, directly ran BLAST to align structural sequences. The only difference was the type of sequences applied – Ramachandran

structural sequences for SARSTs and amino acid sequences for the native BLAST. If the Ramachandran sequence was feasible for structural alignments, SARSTs should be more accurate than BLAST while searching SCOP. Since the previous SARSTs were compared with BLAST, we also did it here to show the improvement of SARST2 over its ancestors. As shown in Fig. 3, SARST1 and *i*SARST were slower than BLAST by >20 folds, but SARST2 ran 7 folds faster than BLAST. In addition, as we noticed that SARST2 was faster than BLAST, it occurred to us that SARST2 could be an example showing that structural alignments are not necessarily slower than sequence alignments. This idea became a key point of this article. Now, we have found that Foldseek is another good example of structural alignments being faster than BLAST sequence alignments, and we have clearly stated it in the revised manuscript (Line 209 on Page 7 and Lines 813–814 on Page 25).

We are sorry for the confusion we might have caused. In addition to the existing statements about previous SARSTs externally running BLAST (**Introduction**: Lines 112–113 on Page 4) and BLAST being assessed because it was our previous alignment engine (**Results**: Lines 182–184 on Page 7), we have added new statements in the legend of Fig. 5c on Page 14 and **Discussion** (Lines 432–435 on Page 15) to let the reader know more clearly why we made comparisons with BLAST.

Although SARST2 made structural while BLAST made sequence alignments, they were compared with the same fundamental data. Just like the BLAST website provides a sequence dataset of the Protein Data Bank for users to search against PDB structures, we also extracted the sequences from SCOP-2.07 and AlphaFold DB structures for BLAST to perform alignment search and be compared with other methods.

In addition, comparing structural alignment methods with BLAST may still be helpful because BLAST can serve as the performance baseline. As shown in Fig. 2, the precision of MODOKA, a recent structural alignment algorithm, was much lower than that of BLAST. Likewise, Figure 3 of the SARST paper (PMID: 17716377) and Figure 8 of the 3D-BLAST paper (PMID: 17335583) showed that some structure alignment search algorithms had lower accuracy than BLAST.

We must acknowledge the reviewer again for recommending Foldseek. Without it, the large-scale AlphaFold DB experiments would be done only for SARST2 and BLAST. Now, the SARST2 is compared with an excellent protein structural alignment search algorithm. We hope that the confusion can be sufficiently resolved.

Unfair, because the times given by the authors do not correspond to this reviewer experience. Searching a single sequence against a UniRef90 doesnt take "couple of hours", speed up with using a parallel version is almost linear with the number of

processors. It is hard to avoid an impression that authors, perhaps unintentionally, use less than optimal options for BLAST, while spending a lot of time on optimizing their own program.

Response:

We thank the reviewer for these instructive comments, which let us realize how misleading our statement of time cost was and how insufficient we provided information on parameter optimizations. The speed and running time we mentioned might be based on one or multiple CPUs. Due to the article length limit, we omitted this detail when making the "couple of hours" statement for PSI-BLAST. In previous work, we tested the time cost of running PSI-BLAST for seven well-known secondary structure prediction algorithms. As shown in Table 2 of Ref 3 (PubMed ID: 32603341), when using one Intel Xeon 3.33 GHz CPU, they took from 714 to 1,272 seconds to run PSI-BLAST against a UniRef-90 dataset of 38.2 million proteins. At the time of this study, UniRef-90 expanded to 172 million proteins. As estimated by the above data, running PSI-BLAST might take 3,215 to 5,727 seconds with the same hardware, meaning 0.9 to 1.6 hours. In the revised manuscript, we modified the statement as "may take an hour with one typical CPU (estimated by Table 2 of Ref[3])" on Page 3. In addition to notifying one CPU and citing the source of the estimate, we lowered the time to one hour because the average time cost of the above-mentioned Table 2 was 814 seconds for 38.2 million proteins, which would lead to 3,663 seconds for the 1.72-million-protein UniRef-90 dataset.

We sometimes described the speed based on one CPU because: 1) the computers of different researchers might have different numbers of CPUs, and one CPU can be a fair standard for comparison; 2) many of the assessed programs did not support parallel computation; 3) methods supporting parallel computation might have different parallelization efficiency (Fig. 3b,c), and using one CPU would be a common base for evaluations. We have verified that the number of CPUs was clearly stated for any speed/time data in the revised manuscript and Supplementary files.

As for the parameters, to develop reliable algorithms, we always tried to find the optimal settings for the assessed methods. The settings could not always be perfect, especially when there was some trade-off between accuracy and speed. In the previous manuscript, the parameters of BLAST were set according to our experiences in accurately finding far-related homologs because, in this work, we had to make sure all the SCOP family-level homologs for each query could be retrieved to reach 100% recall. Therefore, we used the options: word size = 2, scoring matrix = BLOSUM45, and gap opening/extension penalty = 10/3. Thanks to the reviewer's comments, we noticed that we forgot to show the parameter settings for BLAST. Moreover, Foldseek also provided

many parameters. We decided to apply a general procedure to find optimal settings for these database search methods.

Just as we developed all algorithms, “accuracy comes first before speed” was the principle when we met the trade-off in between. Supplementary Table 1 shows that we optimized the main parameters of BLAST and Foldseek one by one. For the starting parameter, accuracy was weighted more than speed. As the overall accuracy was settled, speed would be weighted more for the latter parameters.

The parameter optimizations took us much time; every test was repeated triply. Although we tried our best to find optimal parameter settings for the assessed methods, the settings might be the local rather than the global optimal. Finding the global optimal settings could be unrealistic because there were too many combinations, and the global optimal settings might vary from one dataset to another. We need to explain that we had no intention to optimize our algorithm solely; instead, we managed to compare different methods under optimal conditions. Based on the new parameter settings, the precision of BLAST was slightly lowered, but it ran two folds faster.

- authors do not provide much information about the alignment accuracy of their algorithm, especially on distant homologs or even analogs, those that are not detected by sequence methods. it would be interesting to know how many structurally similar proteins are being identified among those that cannot be seen by BLAST.

Response:

Thank you so much for the constructive comment, which gave us a good direction for improvement. All-against-all pairwise alignments between SCOP family-level homologs from the Qry400 and SCOP-2.07 datasets have been performed. Then, for each percent identity level of BLAST, we computed the average sequence identity of homologs according to SARST2 alignments. As shown in Fig. 5a, for protein pairs receiving <20% identities computed by BLAST, their identities by SARST2 were significantly higher. For instance, the point on the far left of Fig 5a indicates that SARST2 reported a ~9% sequence identity for proteins receiving a 0–1% identity computed by BLAST. These data revealed that SARST2 was more capable of identifying distant homologs than conventional sequence alignments. In this regard, the Foldseek and TM-align also outperformed BLAST. The average sequence identities computed by Foldseek and SARST2 were very similar and higher than those by TM-align.

The distributions of sequence identities computed based on SARST2, Foldseek, TM-align, and BLAST alignments were also observed. In Fig. 5b, if visually drawing a vertical line at the 10% sequence identity level and summing up the area under the curves to the left, we would find that 11.4% of SCOP family-level homologs received

a BLAST sequence identity ≤10%. Calculated similarly, only 3.0% of the homologs received a SARST2 sequence identity ≤10%. If we considered that homologs with sequence identities ≤10% were difficult cases for BLAST to see, there would be about 8.4% (11.4% – 3.0%) difficult cases identifiable by SARST2. Foldseek performed better than SARST2 by identifying 0.8% more difficult cases. For comparison, TM-align could recover 5.7% of cases difficult to see by BLAST.

- critical information seems to be missing. Authors use two datasets, Qry400 and Qry200 for training and validating their algorithms, but I couldnt find their lists in the supplementatary material, which is not well described. Are these sets in the files 491475_0_data_set_686913_s93113.txt, 491475_0_data_set_686912_s93jj2.txt and 491475_0_data_set_686911_s93vv2.txt? there are no explanations in the package (or at least I couldnt find any)

Response:

We thank the reviewer for this notification. The original files we submitted were named "SuppData3-Qry200.txt" and "SuppData4-Qry400.txt". Knowing that the submission system will rename files, in the first line of each Supplementary Data file, we have added a short description of its contents. In addition to the entry lists, we prepared the Qry200 and Qry400 PDB files on the download web page of SARST2.

Remarks on code availability:

as far as I can tell, no code is available, neither in the review materials nor from the website. Only binary executables for several machines are included, so I cannot really evaluate the code.

Response:

Sincerely, we thank the reviewer for reviewing this work and giving us helpful comments. The source code of SARST2 has been submitted as a Supplementary Code archive for peer review along with this revision. It can be compiled using an online Golang teaching service of our laboratory at http://10.life.nctu.edu.tw/Go .

Recently, our national information security policies have been modified to be very restrictive; sometimes, HTTP connections to our academic network from abroad may fail. To help the reviewer compile and test the source code, we have also established a testing SSH server equipped with necessary libraries and several experimental datasets with the following login information,

Host: 140.113.120.11
Username: sarst2
Password: NatureCommunications

SARST2 was written in Golang with our in-house Lo library, which consisted of packages like general, distributed, mathematics, scientific, and other computations. Because we are writing a book for this library and applying for patents for some of its packages, it would be inconvenient for us to make the source code of SARST2 public now. We will certainly release the source codes of SARST2 and Lo library in the future. At present, we can share the source code of SARST2 with the reviewer through the submitted Supplementary Code. The source code can be compiled using the Golang teaching service or the Linux SSH server.

The teaching website has clear instructions on compiling a Golang program based on the Lo library. The Linux SSH server utilizes a prebuilt Lo library to compile Golang programs. After logging in, the reviewer can find a quick manual file (README.txt) in the home directory, clearly describing how to compile SARST2. The latest SARST2 source code can be found on the server. Several experimental datasets have also been prepared over there. As a small laboratory with limited resources, we now have only three server computers for research. Although we have tried our best to clear one to let the reviewer have sufficient memory and disk space to test our programs, the server only has 24 CPU cores. Please forgive us that we are unable to offer the reviewer the same developmental environments of SARST2 to make tests due to the insufficiency of resources.

The reviewer can download Foldseek, TM-align, or other files to the server or transfer the recompiled SARST2 from the server to other places for testing. If the reviewer needs to compile some programs that require specific C/C++ or other libraries that have not been installed, please inform us through the Journal. We did not prepare third-party programs because we may not have the right to distribute them. Among the assessed algorithms, we particularly admire the advances accomplished by Foldseek. We would like to thank the reviewer again for suggesting that we study this brilliant algorithm. It will be our great honor if we have a chance in the future to establish collaboration with the Foldseek team.

NCOMMS-24-13170B

SARST2, a high-throughput protein structure alignment algorithm for searching massive databases

Dear Professors and Editor,

We sincerely thank the reviewers for their careful assessments and constructive suggestions, which have been invaluable in helping us refine the manuscript and clarify its broader relevance. We are especially grateful for the opportunity to submit a revised version and are encouraged by your recognition of the potential value of this work. We have addressed all reviewer comments in the updated manuscript, with the corresponding changes highlighted in light green for ease of reference. To meet the word limits of *Nature Communications*, we have also streamlined the overall text without altering any data or core arguments. We are truly thankful for your time and guidance, and respectfully present our point-by-point responses below.

REVIEWER COMMENTS

Reviewer #1

Remarks to the Author:

The authors have addressed my questions.

A minor point:

- Please spell out SIMD and DP.

Response:

We thank the reviewer once again for the insightful suggestions provided during the previous round of review, which greatly helped us optimize the algorithm and improve the manuscript.

Regarding the current comment, we appreciate the reminder that "SIMD" appeared only as an abbreviation in the original version. We have spelled it out in the revised manuscript as "Single Instruction, Multiple Data" (Page 4, Line 106).

As for "DP", the full term "dynamic programming" was mentioned once in the last paragraph of Page 3. However, we now realize that this single occurrence in the **Introduction** may not be sufficient for readers who start reading from the **Results**

section, where the abbreviation appears more frequently. To improve clarity, we have spelled it out again at the beginning of the **Results** (Page 5, Line 148). We are grateful to the reviewer for drawing our attention to this detail.

=====

I have also evaluated the responses to Reviewer #3 and my related comments are as follows:

1. The foldseek comparison is done as it is also requested by me.
2. The abstract may want to mention foldseek also to strengthen the impact of this paper.

Response:

We thank Reviewer #1 and Reviewer #3 for introducing us to Foldseek, an impressive and timely development that helped us refine our arguments and design more meaningful performance comparisons. We are especially grateful to Reviewer #1 for going above and beyond by evaluating the responses to their comments and those originally raised by Reviewer #3. This effectively involved reviewing two sets of materials, and we sincerely appreciate the extra time and effort this entailed.

While SARST2 showed stronger performance than Foldseek in several benchmark experiments, we fully recognize that Foldseek remains an important and rapidly evolving method. In our previous revision, we did not mention Foldseek in the **Abstract** for two main reasons. First, our goal was not to position SARST2 as a competitor to any specific algorithm but to highlight broader methodological progress in structure alignment. Second, the strict word limit of the **Abstract** constrained our ability to elaborate on comparative results, as we prioritized conveying the key message that structure alignment has now become faster than sequence alignment.

We now fully appreciate Reviewer #1's suggestion to enhance the visibility of SARST2 by explicitly referencing Foldseek. Given Foldseek's prominence and broad adoption, such context is indeed valuable. Accordingly, we revised the Abstract by streamlining the previous reference to BLAST and adding the sentence: *"In large-scale benchmarks, SARST2 outperformed state-of-the-art methods in accuracy, while completing AlphaFold Database searches significantly faster and with substantially less memory than BLAST and Foldseek."*

In parallel, we also updated the **Introduction** (Page 4, Lines 131–136) as follows: *"Tested with an answers-seeded AlphaFold DB using 32 Intel i9 processors, SARST2*

completed a 100% answer-recalled search in 3.4 minutes using 9.4 GiB memory. Foldseek and BLAST took 18.6 and 52.5 minutes, using 19.6 and 77.3 GiB memory, respectively. Moreover, storing AlphaFold DB requires 59.7 TiB disk space, but SARST2's grouped database formatting function reduces the requirement to only 0.5 TiB, compared to 1.7 TiB required by Foldseek".

Once again, we thank Reviewer #1 for the thoughtful feedback and kind support.

=====

3. "may take an hour with one typical CPU (estimated by Table 2 of Ref3". Please be specific about what is one typical CPU.

Response:

We are grateful to the reviewer for pointing out the lack of clarity in our original wording. To address this, we have now specified the CPU model in the manuscript (Page 3, Line 68): *"one typical CPU (estimated with a 3.33 GHz Intel Xeon processor based on Table 2 of Ref[3])"*.

For context, this specification is not merely illustrative. It refers to the actual server used in our laboratory when we conducted the work described in Ref[3]. Remarkably, we still rely on the same machines today. Most of them were purchased around 2010–2011 or handed down from retired colleagues. These aging servers also currently host the official websites for *i*SARST and SARST2.

Our intention was not to exaggerate the computation time required by BLAST but rather to reflect the practical limitations commonly faced by academic research groups working under constrained resources. The performance of SARST2 was therefore optimized under real-world conditions. We hope this explanation clarifies the rationale behind our benchmarking setup and highlights the importance of designing algorithms that remain efficient even in resource-limited environments.

=====

4. Reviewer #3 asked about the remote homologs. The authors responded with the sequence identity: "For instance, the point on the far left of Fig 5a indicates that SARST2 reported a ~9% sequence identity for proteins receiving a 0–1% identity computed by BLAST." I am not sure about this. 9% sequence identity may mean

nothing as it is too low any way. They may want to compare on AF3 structure similarities.

Response:

We appreciate the reviewer for this thoughtful and constructive suggestion. We agree that when the sequence identity between two proteins falls to 0–1 percent (as computed by BLAST), it becomes extremely difficult to determine whether any observed similarity is biologically meaningful. In our analysis, every query-subject pair was drawn from the same SCOP family, suggesting they likely share an evolutionary relationship. Based on this assumption, we reasoned that a more effective alignment method might recover slightly higher sequence identity, even in low-homology cases. Notwithstanding, we fully acknowledge that the 0–1% identity represents the extreme end of the spectrum. The statement, *"the point on the far left of Fig 5a indicates that SARST2 reported a ~9% sequence identity for proteins receiving a 0–1% identity computed by BLAST"*, was intended as an example to help interpret the figure, and it appeared only in our previous response letter, not in the manuscript text. The manuscript explicitly defined the "low-homology region" as the range of 1–20% sequence identity.

We appreciate Reviewer #1's suggestion to consider structural similarity as a more robust measure for remote homology detection. Although AlphaFold's pLDDT scores are useful for estimating local prediction confidence, they are not designed to compare different protein structures. For this reason, we relied on TM-score, a widely used metric supported by nearly all structural alignment tools included in this study.

In the revised manuscript, we added a new analysis to examine how the TM-scores produced by each algorithm vary across different sequence identity levels. As shown in the updated Fig. 5a, for SCOP family-level homologs with >25% identity, all methods, including BLAST, returned consistently high TM-scores. However, in the 1–20% range, the TM-scores produced by BLAST dropped substantially, whereas structure-based methods maintained stronger performance. This observation suggests that BLAST may fail to capture true homologous relationships under such conditions.

SARST2 and Foldseek produced similar TM-scores across all identity ranges, including the 1–20% region. This suggests that both methods are comparably effective in identifying remote homologs, even when sequence similarity is low. We now discuss these findings in the legend of Fig. 5a and the main text (Pages 14–15). We thank the reviewer again for raising this important point, which led to a more rigorous demonstration of SARST2's capacity to detect structural similarity in low-homology regions.

=====

5. About the source code, I cannot open the links that they provided. Not sure it is because my hotel internet is terrible or not... Even if it is available, I do not think I will have time to evaluate the source code. So you may still consider another reviewer to check this.

Response:

We are truly grateful that the reviewer took the time to evaluate our work while traveling. In addition to hosting SARST2 on our website, we had also uploaded a copy of the source code as "Supplementary Code" along with the previous revision through the *Nature Communications* peer review system. This file should be directly accessible via the submission portal.

Regarding the difficulty accessing our website, the issue may have been caused by temporary restrictions on international traffic imposed by our academic network, particularly for servers lacking a certified HTTPS certificate. These access limitations have become more common under recent national information security policies. We thank the reviewer for bringing this to our attention. In response, we have installed a verified HTTPS certificate for the SARST2 server, which we hope will improve international accessibility and help support broader dissemination of the software.

=====

6. Their GitHub page: https://github.com/NYCU-10lab/sarst seems missing lots of basic information… Basic instructions of usage are needed. It will be helpful to provide a test example. Add the reference to this paper will help people to cite it and understand its details. Foldseek GitHub serves as a good example. Somehow, google search cannot find this GitHub easily…

Response:

We thank the reviewer for this valuable suggestion, which has greatly helped us improve the accessibility and usability of the software. In response to your comments, we have updated the project website (https://10lab.ceb.nycu.edu.tw/sarst2) and the GitHub repository (https://github.com/NYCU-10lab/sarst). They now include detailed usage instructions, example datasets, and step-by-step tutorials to assist new users in getting started.

To support proper citation and ensure long-term availability, we have also archived the GitHub repository through Zenodo and obtained a DOI: 10.5281/zenodo.15758781.

This DOI has been added to the **Abstract** and **Code Availability** sections of the manuscript. We are also working to improve the discoverability of the GitHub page through public search engines.

We greatly appreciate the reviewer's thoughtful feedback and encouragement. Your suggestions have been instrumental in helping us strengthen the software's accessibility. We will continue refining SARST and exploring ways to broaden its practical applications. These efforts are intended not only to support the analysis of structural Big Data, but also to contribute to ongoing research in the biological and medical sciences.

=====

Reviewer #2

Remarks to the Author:

This revised manuscript has addressed all my previous concerns in a very comprehensive and thoughtful way. The resulting improvements further confirm my original opinion that this work definitely should be published in Nature Communications.

Response:

We sincerely thank the reviewer for the encouraging remarks and the insightful suggestions provided in the previous round of peer review. Your comments played a key role in shaping several important improvements to this work. In particular, your suggestion led us to analyze the TM-score distributions produced by SARST2 and to define a more objective threshold for identifying family-level homologs. Your recommendation also prompted us to benchmark SARST2 using the highly stringent 40% identity non-redundant CATH dataset. This allowed us to demonstrate that the algorithm's high accuracy is not simply the result of overfitting to the training data.

In this revision, we further expanded our analysis of TM-score behavior and uncovered additional insights that may inform future development of SARST. For example, TM-align reported higher TM-scores than SARST2 and Foldseek in the low-identity region, even when its alignments yielded lower sequence identities. While SARST2 and Foldseek incorporate sequence information into their algorithms, TM-align relies exclusively on Cα coordinates. Although the elevated TM-scores from TM-align in the low-homology region may not always indicate evolutionary relatedness, they may help highlight instances of convergent evolution. These observations suggest

that incorporating insights from TM-align could improve SARST's ability to detect such structural patterns in future work.

Once again, we are deeply grateful for your constructive feedback. Your comments have contributed significantly to both the scientific rigor and the overall scope of this study.