

Structural Bioinformatics

Combining co-evolution and secondary structure prediction to improve fragment library generation.

Saulo H. P. de Oliveira^{1,*}, and Charlotte M. Deane¹

¹Department of Statistics, University of Oxford, Oxford, OX1 3LB, United Kingdom

*To whom correspondence should be addressed.

Associate Editor: XXXXXXXX

Received on XXXXX; revised on XXXXX; accepted on XXXXX

Abstract

Motivation: Recent advances in co-evolution techniques have made possible the accurate prediction of protein structures in the absence of a template. Here, we provide a general approach that further utilizes co-evolution constraints to generate better fragment libraries for fragment-based protein structure prediction.

Results: We have compared five different fragment library generation programmes on three different data sets encompassing over 400 unique protein folds. We show that considering the secondary structure of the fragments when assembling these libraries provides a critical way to assess their usefulness to structure prediction. We then use co-evolution constraints to improve the fragment libraries by enriching them with fragments that satisfy constraints and discarding those that do not. These improved libraries have better precision and lead to consistently better modelling results.

Availability: Data is available for download from: <http://opig.stats.ox.ac.uk/resources>. Flib-Coevo is available for download from: <https://github.com/sauloho/Flib-Coevo>

Contact: saulo.deoliveira@dtc.ox.ac.uk

Supplementary information: Supplementary data are available at *Bioinformatics* online.

1 Introduction

The quality of protein structure prediction from sequence in the absence of a template is determined by the quality of its inputs (Moult *et al.*, 2016). The most successful approaches for template-free prediction use a library of fragments from known protein structures to restrict the conformational search space (Raman *et al.*, 2009; Moult *et al.*, 2016). When building this input library, it is possible to exclude near-native conformations from the search space altogether by selecting an incorrect set of fragments. If the input fragment libraries contains no near-native conformations, modelling is likely to fail. This is the case regardless of the efficiency of the search heuristics or the quality of the scoring functions used during modelling. Even with the accurate contact predictions now available (Jones *et al.*, 2014; Ovchinnikov *et al.*, 2017), modelling is unlikely to succeed in the presence of sufficiently accurate predicted contacts if the input fragments cannot represent near-native conformations correctly.

One potential concern when using a library of fragments to model a protein structure is that the target structure may have unique local conformations not observed in any other structure or fold. However, a test on a non-redundant set of 1,261 proteins showed that it was possible to

correctly reconstruct more than 99% of cases (less than 1 Å to native) using 4 to 16 residue-long fragments extracted from non-homologous protein structures (Baeten *et al.*, 2008).

In order to perform protein structure prediction, a set of fragments is selected to comprise the library based on the target sequence and its derived descriptors (e.g. predicted secondary structure, predicted torsion angles). Short fragments with the same sequence can have different structures (Zhou *et al.*, 2000) and local sequence similarity has been shown to have little to no correlation to local structure similarity (de Oliveira *et al.*, 2015). Therefore, methods that generate fragment libraries often struggle to identify structurally similar fragments and a large set of fragments is usually chosen to represent each target position. There does not appear to be a consensus as to the optimal number of fragments to be considered, but libraries usually contain tens to hundreds of fragments per target position (de Oliveira *et al.*, 2015; Gront *et al.*, 2011; Wang *et al.*, 2017).

Library quality can be assessed by the proportion of target residues that are correctly represented by at least one fragment (coverage). While a coverage of 100% is often seen as required for modelling success, it is not sufficient to guarantee accurate modelling. Although each fragment may be considered correct, the sum of small errors of each fragment can still lead to the exclusion of near-native conformations. Furthermore, the combinatorial problem persists; if certain target positions are only

correctly represented by a single fragment, then it is unlikely that near-native conformations will be sampled. As a way to ensure that a sufficient number of correct fragments is available for most positions, fragment library generators are also trained in terms of their precision, the proportion of correct fragments in the library. When methods output a varying number of fragments per position, this metric can give a misleading indication of performance. A previous study has shown that α -helical and, to a lesser extent, β -strand fragments present lower structural variability and, therefore, tend to be easier to predict (de Oliveira *et al.*, 2015). It is, therefore, possible to enhance the precision of a fragment library by outputting a large number of fragments for α -helical or β -sheet positions and a lower number of fragments for regions whose modelling is harder (e.g. loops).

Several fragment library generators have been published recently (de Oliveira *et al.*, 2015; Shen *et al.*, 2013; Wang *et al.*, 2017; Trevizani *et al.*, 2017; Bhattacharya *et al.*, 2016). Given the impact that fragment libraries have on the output of template-free structure prediction, we performed a large-scale comparison considering five of these methods, NNMake (Gront *et al.*, 2011), Flib (de Oliveira *et al.*, 2015), LRfragLib (Wang *et al.*, 2017), Profrager (Trevizani *et al.*, 2017), and FRAGSION (Bhattacharya *et al.*, 2016). Each programme was assessed in terms of their precision and coverage for three comprehensive data sets containing proteins with unique folds. Our findings show that considering the precision of fragments based on their predominant secondary structure is critical for achieving high quality libraries. The method LRfragLib showed the highest overall precision, but we found this was caused by its tendency to predict far more fragments in α -helical regions and far fewer in the harder to predict loop segments.

We also show that co-evolution can be used to enrich fragment libraries, thus improving the chances of modelling success. We compared the quality of fragments that satisfy co-evolution constraints (Jones *et al.*, 2014) to those that do not and show that bad-quality fragments can be excluded from the libraries in this fashion. We present a new protocol, Flib-Coevo for enriching fragment libraries with high quality co-evolution fragments in areas that are difficult to model. These fragments are then incorporated into existing fragment libraries leading to consistently better template-free modelling results.

2 Methods

2.1 Data Sets

Unique fold data set: We selected all the unique fold proteins at 40 % sequence identity from the 2.06 build of Astral (Fox *et al.*, 2014). A unique fold protein is defined as a case for which there is no other protein in the set in the same family, superfamily or fold according to the SCOPe classification (Fox *et al.*, 2014). We further reduced this set by considering only the proteins whose PDB_SEQRES is a perfect match to the fasta sequence described in the Astral database, resulting in a final set of 274 proteins (SI Table 1).

Multi-domain data set: we selected all proteins shorter than 150 residues and with exactly two domains according to their annotation in CATH (Orengo *et al.*, 1997). To avoid redundancy, we removed homologous sequences from this set using BLASTp (Altschul *et al.*, 1990) with standard parameters using a 30% sequence identity cut-off. In total, this multi-domain data set contains 111 non-homologous proteins (SI Table 2).

Modelling data set: we selected all proteins from the unique fold data set for which a multiple sequence alignment (MSA) with at least 500 sequences could be built. This smaller set contains 87 proteins (see SI Table 1).

CASP12 Modelling set: we selected 37 targets from CASP12 for which structures and adequate sequence information were available (SI Table 3). A smaller threshold of 50 sequences in the MSA was used to ensure that a sufficient number of targets was considered and to guarantee that an output for contact prediction was generated.

2.2 Fragment Library Generation

The methods Flib (de Oliveira *et al.*, 2015), LRfragLib (Wang *et al.*, 2017), Profrager (Trevizani *et al.*, 2017), and NNMake (Gront *et al.*, 2011) require secondary structure and torsion angle predictions to generate fragment libraries. We predicted secondary structure for the 274 proteins of the unique fold set using PSIPRED V4.0 (McGuffin *et al.*, 2000) with standard parameters. Identical secondary structure prediction was used as input for each method.

We used SPIDER2 (Heffernan *et al.*, 2015) with standard parameters to produce torsion angle predictions for the methods Flib, LRfragLib, and Profrager. For the software NNMake, we used torsion angle predictions as output by SPINE-X (Faraggi *et al.*, 2012) with standard parameters, as recommended by NNMake. The software FRAGSION (Bhattacharya *et al.*, 2016) does not require secondary structure prediction or torsion angle prediction as input.

The following versions of the software were used: Flib v1.02, LRfragLib v1.0, Profrager v1.2.0, NNMake (from Rosetta build 3.8) and FRAGSION v1.0. Standard parameters were used, as reported in (de Oliveira *et al.*, 2015; Wang *et al.*, 2017; Trevizani *et al.*, 2017; Gront *et al.*, 2011; Bhattacharya *et al.*, 2016).

Homologues were removed by each method before fragment library generation using appropriate input options. After fragment libraries were generated, we checked the output and removed any remaining homologues. Homologues were detected by BLAST+ version 2.5.0 (Altschul *et al.*, 1990) using an e-value cutoff of 0.05. Given that our data set had proteins with unique fold from the Astral database, this protocol is likely to identify all homologues.

2.3 Validation and Definitions

Fragments have been validated by computing their Root Mean Square Deviation to the native structure (RMSD). We have used a varying cutoff to ascertain whether a fragment correctly models the target's local structure. Fragment libraries were compared in terms of the following properties:

Average Fragment Length: The average length of the fragments in the library. We used the default setting of nine residues for NNMake. For Profrager and FRAGSION, we chose to output fragments of length six given that shorter fragments present lower RMSD and this was the minimum fragment length output across all methods.

Average Number of Fragments per Position: The total number of fragments in a library divided by the target's length.

Predicted Predominant Secondary Structure: we classify fragments according to their predicted predominant secondary structure. For the rest of this manuscript, when we refer to predominant secondary structure, we actually refer to the predicted secondary structure of the target. A predominant α or predominant β fragment is predicted to be comprised by at least 50% of α -helical or β -strand residues, respectively. A predominant loop residue is comprised by at least 50% residues that are neither part of an α -helix or a β -strand. The Other category describes fragments that do not have a predicted predominant secondary structure.

Proportion of fragments per secondary structure class: The proportion of fragments output for each of the four predicted secondary structure classes described above. This is calculated by dividing the number of fragments of a predicted predominant secondary structure class by the total number of fragments.

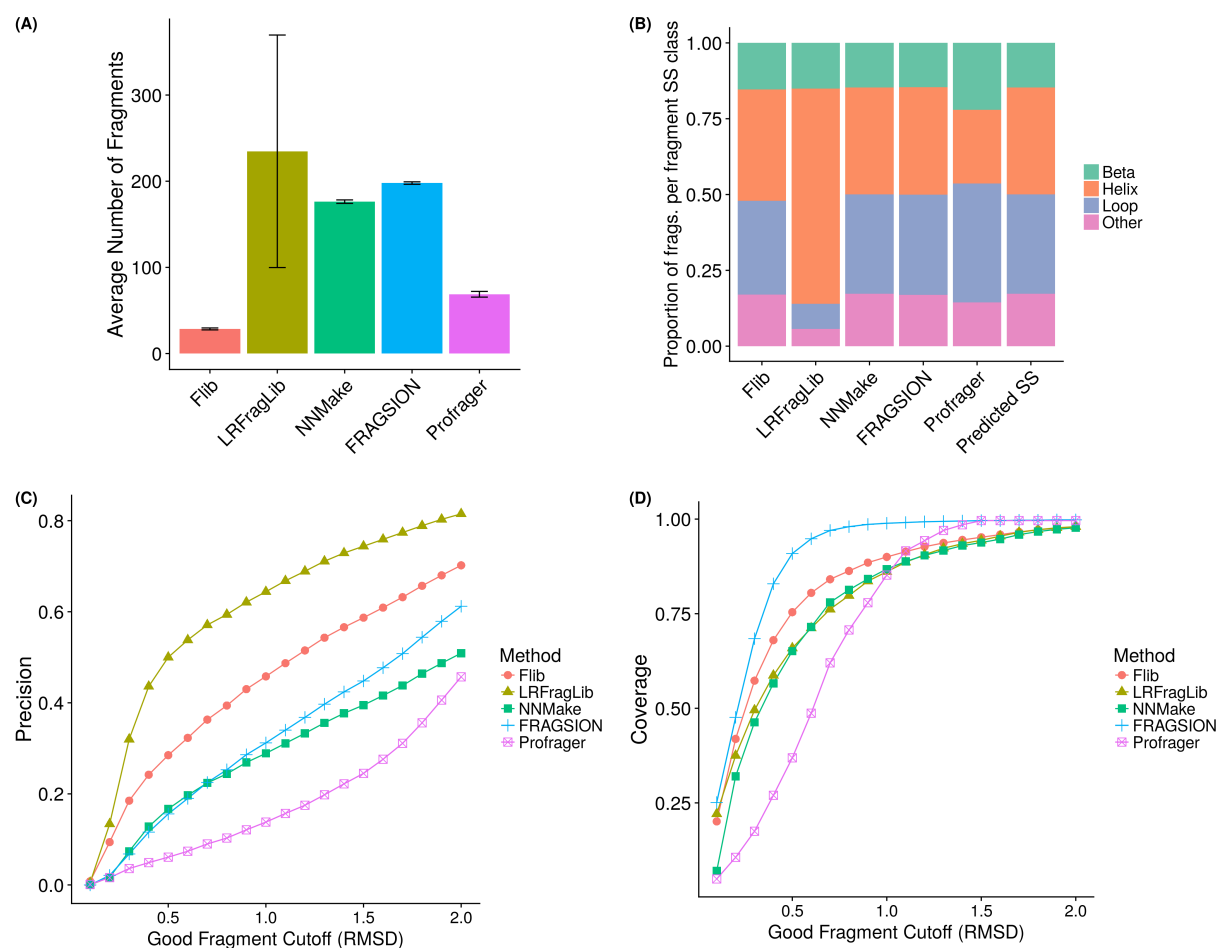


Fig. 1. Comparison of five different fragment library generators for 274 unique fold proteins. Methods were compared in terms of the average number of fragments per target position (A), the proportion of fragments output by each predicted predominant secondary structure type (B), precision (C), and coverage (D). Bars correspond to one standard deviation. Precision and coverage are shown based on a varying good fragment cutoff (x-axis).

Precision: the number of good fragments (fragments with an RMSD below our varying cutoff) divided by the total number of fragments in a library.

Coverage: the percentage of target residues represented by at least one good fragment in the fragment library.

2.4 Co-evolution Fragment Library Enrichment

For the 87 proteins in our modelling data set, we have used metaPSICOV v1.04 (Jones *et al.*, 2014) to predict contacts following the procedure described in (de Oliveira *et al.*, 2017a). Precision of predicted contacts are shown in SI Table 2. A contact is defined as two residues whose C- β s (C- α s in the case of Glycine) are less than 8Å apart. Analogously, a fragment that satisfies a predicted contact is defined as a fragment that contains a pair of residues predicted to be in contact and whose C- β s (C- α s in the case of Glycine) are less than 8Å apart. If the C- β s are more than 8Å apart, the fragment is said not to satisfy a contact predicted to occur within it.

For our co-evolution based protocol, Flib Co-evo, we removed a fragment if it had a predicted contact within it which was not satisfied. We then enriched the library by adding new fragments that satisfy predicted contacts (co-evolution fragments). No more than 30 fragments per position are added in this fashion. Many positions remain unchanged as there are no predicted contacts for their fragments.

2.5 Model Generation and Assessment

We have used our sequential approach to template-free structure prediction, SAINT2 (de Oliveira *et al.*, 2017b), to produce 1,000 decoys for the 87 targets in our modelling set using the fragment libraries output by Flib, LRFragLib, and by Flib-Coevo.

To assess model quality, we have calculated the TM-Score (Zhang and Skolnick, 2007) of the best model produced using each fragment library. Here, the best model is defined as the model with the highest TM-Score when compared to the native structure. Models were considered as having the correct topology if the TM-Score of the best model was greater than 0.5 (Xu and Zhang, 2010).

3 Results

3.1 Comparing fragment library generation software

Several protocols have been developed to generate fragment libraries for a target sequence for use in structure prediction (e.g. Gront *et al.*, 2011; Kalev and Habeck, 2011; Shen *et al.*, 2013; de Oliveira *et al.*, 2015; Wang *et al.*, 2017; Trevizani *et al.*, 2017; Bhattacharya *et al.*, 2016). Given the impact that the quality of fragment libraries has on the success rate of template-free modelling, it is important to determine which method

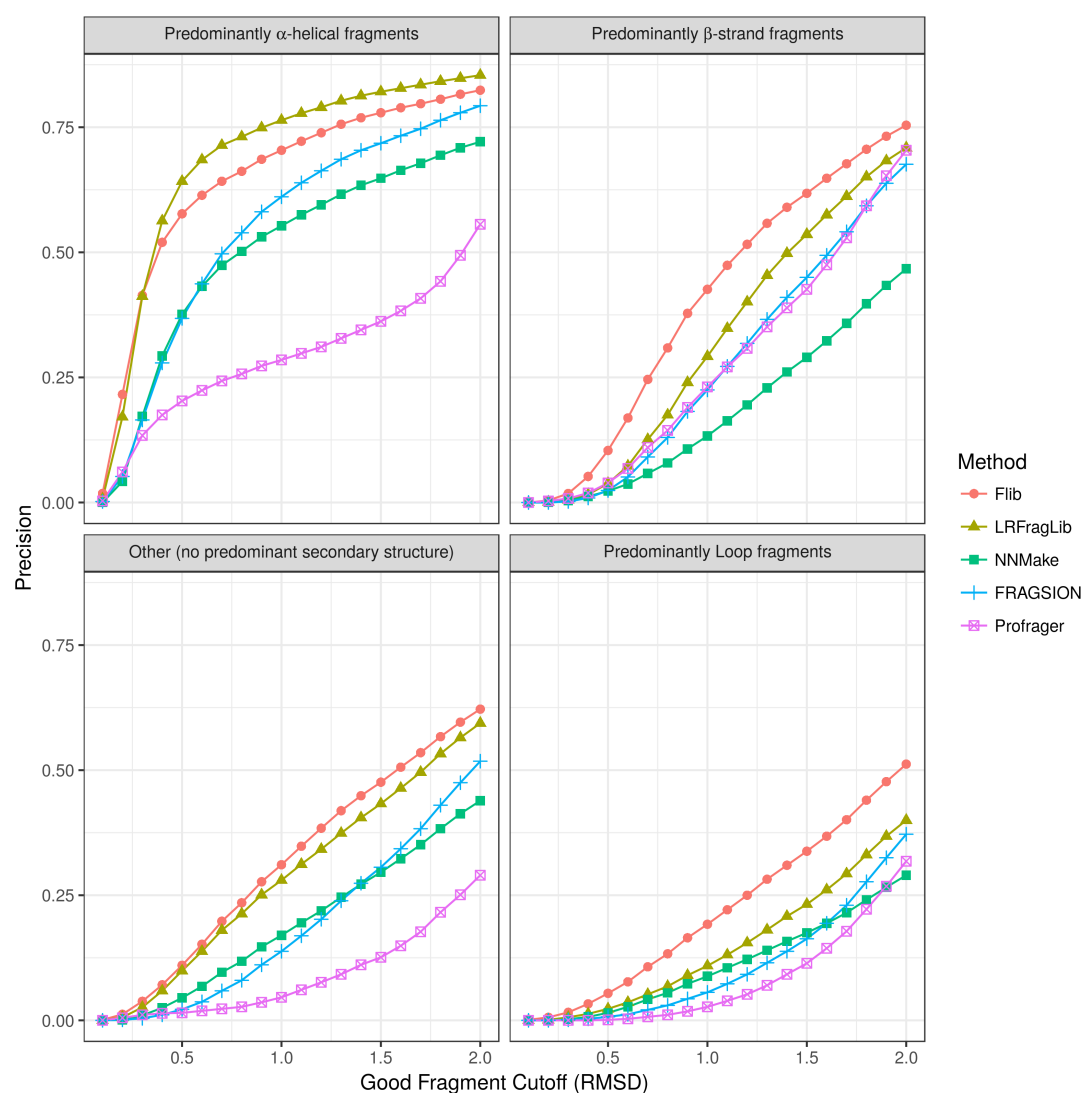


Fig. 2. Comparison of the precision according to predominant secondary structure obtained by five fragment library generators for the 274 proteins comprising our unique fold data set.

produces the most useful libraries. We have compared five fragment library generators on a set of 274 unique fold proteins.

When considering the average number of fragments output per position for our unique fold data set (Figure 1), the method Flib outputs fewer fragments per position than all of the methods considered ($\sim 29 \pm 0.7$) and the method LRFRagLib outputs the most fragments per target position ($\sim 249 \pm 119$). For the methods where the number of fragments per target position is defined at input, default settings were used. Our results also show that most methods tend to produce a relatively similar number of fragments across all target positions, presenting a standard deviation of less than one fragment per position (Figure 1). The only method that outputs a highly varying number of fragments per position is LRFRagLib, with a standard deviation of ~ 119 fragments per position.

Our results for the proportion of fragments output according to their predicted secondary structure class show that most methods present proportions similar to the predicted secondary structure of the targets. The method LRFRagLib outputs more than 50% of its fragments for positions that were predicted to be predominantly α -helical (Figure 1), even though fewer than 25% of positions were predicted to be helical. This constitutes a problem since α -helical fragments are easier to predict and therefore often

of good quality (de Oliveira *et al.*, 2015). In contrast, LRFRagLib outputs fewer fragments for the more difficult to predict Loop and Other positions (de Oliveira *et al.*, 2015).

There is no consensus as to the optimal fragment length, but it has been suggested that including fragments of varying length may be beneficial for modelling (Handl *et al.*, 2012). We have, therefore, assessed the average fragment length output by each method (SI Figure 1). Only two of the protocols produce fragments of varying length, Flib and LRFRagLib, these are on average ~ 6.8 and ~ 7.1 residues long, respectively.

We have also compared the methods in terms of their run time. All of the protocols considered present linear complexity, proportional to the length of the target. The methods FRAGSION and Profrager both operate under a minute per protein per core, whereas the other three methods take significantly longer to run. As an example, for the 100 residues-long protein 1BM8 (refer to SI Table 1), Flib, LRFRagLib, and NNMake took approximately 3 hours, 10 hours, and 12 hours per core respectively to produce an output. For the longest protein in our set (1M1C - 652 residues long), Flib, LRFRagLib, and NNMake took approximately 14 hours, 82 hours, and 63 hours per core respectively to produce an output. Although Flib can produce an output faster than NNMake and LRFRagLib

on a single core, unlike the other methods it currently does not have any parallelization capabilities. Two commonly used metrics to assess the quality of fragment library are precision and coverage (see Methods for more details). We calculated the average precision and coverage obtained by each protocol across all proteins in the unique fold data set. LRFRagLib achieves the highest precision (at 1Å, $\sim 0.64 \pm 0.27$). This is not surprising given that it outputs a significantly higher number of α -helical fragments when compared to other methods, which may lead to its precision being overestimated. Loop and Other fragments are inherently harder to predict accurately when compared to α -helical and, to a lesser extent, β -strand fragments (de Oliveira *et al.*, 2015). When a varying number of fragments is output per target position, precision can be overestimated if a larger number of fragments is output for regions that have less structural variability. Given this limitation, it has been suggested that a better way to assess fragment library quality is to consider the precision according to the predicted predominant secondary structure of the fragments (de Oliveira *et al.*, 2015). We have compared the precision across each of four predicted predominant secondary structure types for the 274 proteins in our unique fold data set (Figure 2). LRFRagLib obtained the highest precision for predominantly α -helical fragments, whereas Flib presented the highest precision for predominantly β -strand, predominantly Loop, and Other fragments. Therefore, LRFRagLib's global precision is inflated by the higher number of fragments for α -helical regions that it outputs. Amongst the methods that output a close-to-native proportion of fragments of each secondary structure type, Flib presents the highest overall precision (at 1Å, $\sim 0.46 \pm 0.18$).

To strengthen our comparison, we have also assessed the performance of each predictor for a multi-domain set of 111 non-homologous two domain proteins (SI Figures 2-3). For this set, Flib presented, on average, the lowest number of fragments per position ($\sim 29 \pm 1.5$) and LRFRagLib presented the highest number of fragments per position ($\sim 239 \pm 136$). Similar to the results produced for the previous set, all but one method produced a near-native proportion of fragments according to each secondary structure type (SI Figure 2). LRFRagLib output a higher number of fragments for α -helical positions in this set, once again suggesting that its precision is overestimated. When considering the precision according to the predominant secondary structure of the fragments (SI Figure 3), LRFRagLib presented the highest precision of α -helical fragments and Flib presented the highest precision for β -strand and loop fragments. For the fragments with no predominant secondary structure (Other), the precision of Flib and LRFRagLib was comparable.

We carried out a third comparison for a set of 37 targets from CASP12 (SI Figures 4-5). For the CASP12 set, Flib also presented the lowest number of fragments per position ($\sim 29 \pm 1.0$) and LRFRagLib the highest ($\sim 170 \pm 124$). LRFRagLib presented the highest overall precision (at 1Å, $\sim 0.60 \pm 0.31$), yet it output a disproportionately high number of α -helical fragments (SI Figure 4), as also shown for the other data sets. When considering the methods that output a close-to-native proportion of fragments of each secondary structure type, Flib presented the highest overall precision (at 1Å, $\sim 0.48 \pm 0.07$). When considering the precision according to the predominant predicted secondary structure of the fragments (SI Figure 5), LRFRagLib presented the highest precision for α -helical fragments and Flib presented the highest precision for β -strand, other and loop fragments. For this set, LRFRagLib presented a lower precision for other and loop fragments when compared to other methods.

When considering the coverage obtained by each predictor, the method FRAGSION obtained the highest coverage across all data sets (at 1Å, $\sim 0.99 \pm 0.03$). The other methods present similar coverages ranging from 0.85 (Profrager) to 0.90 (Flib), at 1Å.

A direct relationship between the quality of a fragment library as assessed in terms of precision and that of the final model is expected.

However, there may be deviations. To check for this, we compared the two most successful fragment library generators, Flib and LRFRagLib, in terms of the quality of the models produced using their fragment libraries (SI Figure 6). Models were generated using our fragment-based sequential protein structure prediction protocol, SAINT2 (de Oliveira *et al.*, 2017b). We produced two pools of 1,000 models for each of the 87 targets in the unique fold set that had enough sequence information for accurate contact prediction (more than 500 sequences in the MSA). Flib fragment libraries led to correct models (TM-Score > 0.5) being produced for 44 cases and LRFRagLib led to correct models being produced for 40 cases. When considering the best model, Flib models were better than LRFRagLib's for 30 of the 45 cases where a correct answer was produced.

We also performed a second comparison using our CASP12 set (SI Figure 7). Two pools of 1,000 models were generated using Flib and LRFRagLib fragment libraries and our modelling protocol SAINT2. For this set, Flib fragment libraries produced correct answers for 14 cases and LRFRagLib for 9 cases. The best model produced using Flib libraries was of better quality than the best model generated using LRFRagLib libraries for 30 out of 37 cases (12 out of the 14 cases with a correct answer).

Our results confirm previous findings that predominantly α -helical and, to a lesser extent, predominantly β -strand fragments are easier to predict. With the exception of Profrager, all the predictors achieve at least 50% precision for predominantly α -helical fragments and only Flib achieves close to 50% precision for predominantly β -strand fragments at a 1Å RMSD cutoff. Precision for predominantly loop fragments and fragments with no predominant secondary structure (other) continue to pose a prediction challenge. For predominantly loop fragments, Flib libraries were twice as precise when compared to the second best predictor (LRFRagLib). Overall, these findings suggest that Flib is better suited for usage in template-free modelling. However, even Flib obtained a precision for predominantly loop fragments below 25%, suggesting that modelling remains challenging in these regions.

3.2 Assessing the quality of fragments that satisfy co-evolution constraints

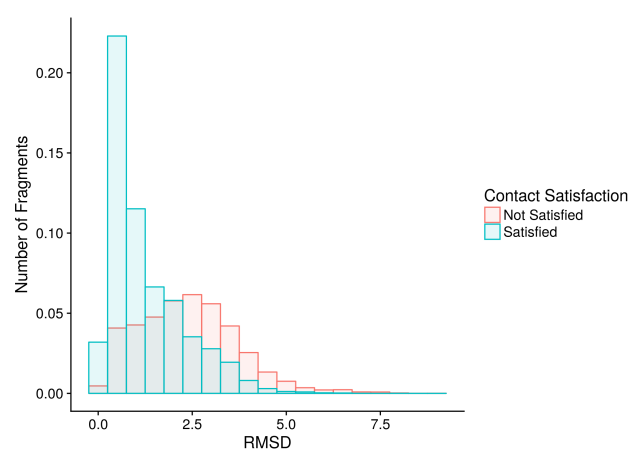


Fig. 3. Assessment of the RMSD of fragments that do or do not satisfy metaPSICOV predicted contacts for the 87 proteins in our unique Fold data set. Only fragments output by Flib which contain a predicted contact are shown.

Co-evolution can be used to accurately predict protein contacts based on a target sequence (Jones *et al.*, 2014; Ovchinnikov *et al.*, 2017). This provides a new set of structural descriptors that are derived from sequence and could be used during fragment library generation. In particular,

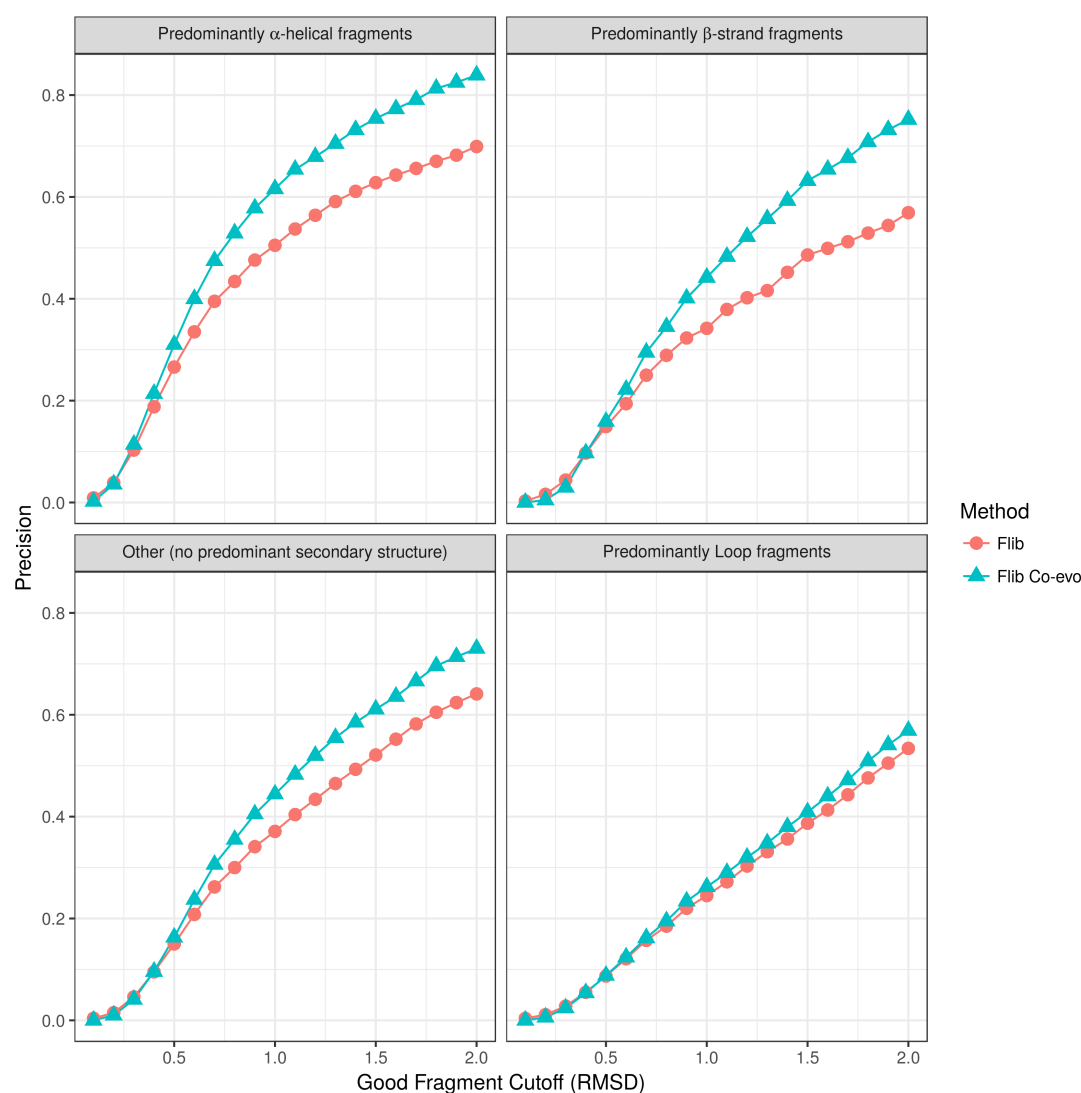


Fig. 4. Comparison of the precision according to predominant secondary structure of Flib libraries against the libraries enriched with co-evolution fragments (Flib-Coevo). Precision is shown based on a varying good fragment cutoff (x-axis).

predicted contacts could be used to improve the precision of fragments that are harder to predict (predominantly loop and other SS classes).

Here, we investigated if predicted contacts can be used to identify good fragments and further improve fragment libraries. We used metaPSICOV to generate contact predictions for the subset of targets in our unique fold data set for which a sufficient number of homologue sequences was available (>500 sequences). In total, 87 proteins were used for this comparison. We used the fragment libraries produced by Flib to assess the quality of the fragments that satisfied the predicted contacts against the ones that did not (Figure 3). For this assessment, we only considered the fragments for which at least one contact was predicted (where both residues predicted to be in contact fell within the fragment). Our results show that the RMSD of fragments that satisfy the predicted contacts is significantly lower than the ones that do not ($p\text{-value} < 2.2e^{-16}$). We have repeated these calculations using fragment libraries output by the other fragment library generators and obtained similar results (SI Figure 8). We have also assessed this relationship in terms of the sequence separation between the residues predicted to be in contact (SI Figure 9). Contacts predicted

between residues that are less than 12 residues apart are classified as short-range. Remaining predicted contacts are classified as medium-range. Our results show that fragments that satisfy predicted contacts tend to be of better quality regardless of the sequence separation between the residues, though the effect seems to be more pronounced for short-range contacts.

When considering the distribution of the fragments according to their secondary structure type (SI Figure 10), ~60% of the fragments that satisfy contacts are within 1\AA of the native structure regardless of the secondary structure of the fragment. This proportion increases to 80% when considering a more lenient cutoff of 2\AA . We observe distinct behaviours in terms of secondary structure for the fragments that do not satisfy the predicted contacts, with α -helical fragments presenting a lower RMSD when compared to the other secondary structure types ($p\text{-value} < 2.2e^{-16}$). Our results also show that more than 80% of predominantly β -strand fragments and Other fragments that do not satisfy predicted contacts are of poor quality ($\text{RMSD} > 1\text{\AA}$), suggesting these fragments should not be included in the fragment libraries. Removing these fragments from the libraries had no noticeable effect on fragment library coverage (SI Figure 11).

Our results show that predicted contacts can be used to ascertain whether fragments are of good quality. Predicated on this finding, we altered the Flib protocol to use co-evolution to enrich final libraries with good quality fragments.

We considered the top 100 fragments per position as output by Flib and selected the fragments that satisfied at least one predicted contact (co-evolution fragments). The co-evolution fragments are on average longer ($\sim 8.7 \pm 0.8$ residues) than the fragments Flib previously selected ($\sim 8.3 \pm 0.7$ residues). The proportion of co-evolution fragments in each predicted secondary structure class is similar to the predicted secondary structure for the positions considered (SI Figure 12).

For our final protocol, we removed fragments from Flib libraries that did not satisfy contacts predicted to occur within them. We then added to these libraries the co-evolution fragments described in the previous paragraph, obtaining a final fragment library (Flib-Coevo). No more than 30 co-evolution fragments were added per target position. We then compared the original fragment libraries produced by Flib against the ones produced by our new protocol, Flib-Coevo, for the 87 proteins in our unique fold data subset. Only fragments and positions that contained at least one predicted contact were considered. We find that the co-evolution-enriched libraries show higher precision and coverage when compared to the original Flib libraries (SI Figure 13). When considering the precision according to the predominant secondary structure class of the fragments (Figure 4), Flib-Coevo obtained a higher precision for all secondary structure classes, although the effect was less pronounced for predominantly loop fragments. These findings show that co-evolution can be used to enrich the precision of fragment libraries at no loss of coverage. Similar results were observed when comparing the precision of Flib-Coevo against Flib for the 37 proteins in our CASP12 set (SI Figure 14).

In order to assess the impact of the co-evolution-enriched fragment libraries on model generation, we used our fragment-based sequential protein structure prediction protocol, SAINT2 (de Oliveira *et al.*, 2017b), to produce 1,000 decoys for each of the 87 targets in the unique fold set that had enough sequence information for accurate contact prediction (more than 500 sequences in the MSA). We compared the models produced using the original Flib fragment libraries against the co-evolution enriched fragment libraries output by Flib-Coevo (SI Figure 15). Our results show that enriching fragment libraries leads to consistent modelling improvements for a large number of cases. To assess statistical significance of this finding, we compared the distributions of TM-Scores produced using Flib and Flib-Coevo for each target in our modelling set using a Kolmogorov-Smirnov test. The test supports that the null hypothesis (distribution of TM-Scores obtained by Flib-Coevo not greater than the distribution of TM-Scores obtained by Flib) should be rejected for approximately 40% of the cases ($p\text{-value} < 0.001$). For a list of all p -values, refer to SI Table 4.

4 Discussion

Fragment-based protein structure prediction has shown consistent improvements over the last few years (Moult *et al.*, 2016). One of its limitations however is that modelling success is still determined by the quality of its inputs. In particular, the quality of the fragment libraries used for modelling has a significant impact on the accuracy of the models produced and modelling fails when near-native conformations are not well represented by the fragments in the library. A number of protocols has been developed to produce libraries for the purpose of template-free protein structure prediction. Given the impact such libraries can have on the outcome of fragment-based modelling, it is imperative to determine which fragment library generators are more likely to lead to successful modelling. We present here an unbiased comparison of five different

fragment library generators for a data set comprised of 274 unique fold proteins and conclude that the method Flib produced the most useful libraries for template-free protein structure prediction.

Our unique fold data set was chosen to ensure that methods do not benefit from the presence of remote homologues in the structural database from which fragments are extracted. This is important to promote a more realistic template-free structure prediction scenario. To corroborate results for this set, we have extended our comparison to two additional data sets, one comprised of 111 multi-domain proteins and a second set containing 37 targets from CASP12. Overall, our results encompass over 400 distinct protein folds.

The number of fragments chosen to represent each target position varies across most methods and there is no consensus as to the ideal number of fragments that should be used (de Oliveira *et al.*, 2015; Gront *et al.*, 2011; Wang *et al.*, 2017). Using too many fragments may be detrimental as the combinatorics become unfavourable. On the other hand, while using fewer fragments leads to a reduced search space, near-native conformations may end up being excluded from the library. A compromise can be made by using as few fragments as possible while maintaining a near-complete coverage (the proportion of target residues represented by at least one good fragment). Despite presenting the lowest number of fragments per position, the method Flib obtained the second highest coverage across our unique fold data set. This suggests that near-native conformations are not being excluded from its output despite the low number of fragments output.

One potential pitfall of both precision and coverage, commonly used metrics to assess fragment library quality, is that they depend on fragment RMSD from native, which in turn is length-dependent. Therefore, these quality measures are inherently biased towards shorter fragments. It has been shown previously that there is a slight loss of precision and coverage as fragment length increases (de Oliveira *et al.*, 2015). To ensure that the comparisons presented here are not biased by fragment length, we calculated the RMSD distribution of different methods at different fragment lengths (SI Figure 16) and shown that the methods that are more precise tend to output fragments of lower RMSD regardless of the length of the fragment being considered.

While most methods output a similar number of fragments per target position across all residues, the method LRFRagLib outputs a significantly larger number of fragments for α -helical regions. This is particularly problematic given that α -helices present lower structural variability and are therefore easier to predict. By increasing the number of fragments output for these regions in comparison to more variable protein regions (e.g. protein loops), the overall precision for LRFRagLib is inflated.

When considering the precision depending on the predominant secondary structure of the fragments, the method Flib presented higher precision for 3 out of 4 secondary structure types. LRFRagLib presents higher precision for α -helical fragments. Our modelling results comparing Flib and LRFRagLib corroborate the relationship between the precision according to predominant secondary structure and final model quality, where Flib led to correct models being produced for more cases in both our modelling and CASP12 data sets and to better models being produced in general. The method FRAGSION presents rapid run speeds and the best coverage across all methods. FRAGSION could be used to generate fragment libraries for very large data sets.

We have shown that co-evolution can be used to improve the quality of fragment libraries. Fragments that satisfy the predicted contacts tend to be of good quality even when considering a stringent cutoff of 1Å. Interestingly, this is the case even though some of the predicted contacts are incorrect. More importantly, predicted contacts can be used to discard fragments without any impact on library quality. This finding is method-independent and, therefore, this exclusion could be incorporated into all existing fragment library generation software.

The number of targets for which this protocol can be used and the coverage of the co-evolution fragments are limited by the non-redundant sequence information available and, conversely, by the number of contact predictions considered (Jones *et al.*, 2014; Ovchinnikov *et al.*, 2017). For our analyses, we only considered targets with at least 500 non-redundant sequences present in their multiple sequence alignment, a cutoff that was suggested in (Jones *et al.*, 2014). It is likely that a more lenient cutoff for the number of non-redundant sequences could increase the number of targets for which our protocol leads to improvements. Given that we only consider predicted contacts with an estimated PPV > 0.5, we expect fewer predicted contacts to be output as the number of non-redundant sequences considered decreases. As co-evolution methods for contact prediction become more precise and the number of available sequences increases, the applicability of this protocol and the coverage of the co-evolution fragments are likely to increase.

It has been previously suggested that fragment libraries should be customised according to the target's secondary structure type (Abbass and Nebel, 2015; de Oliveira *et al.*, 2015). Our results show that the average precision of fragment library generators differs between each of the four main SCOP classes (SI Figure 17). This suggests that such a customisation may improve results further and could be explored in conjunction with the use of co-evolution information.

When enriching fragment libraries by identifying novel fragments that satisfy predicted contacts, we also observed an increase in the overall precision at no loss of coverage. We have used these findings to develop a new protocol, Flib-Coevo, which selects fragments that satisfy predicted contacts. This protocol can work in a method-independent fashion and can be used to supplement fragment libraries generated by other fragment library protocols. Our modelling results further corroborate that enriching fragment libraries in this fashion leads to better models being produced. However, the automatic selection of the best model is still an active field of research and, in practice, a sub-optimal model is likely to be selected. Improving modelling by use of co-evolution fragments does not guarantee that a better model will be selected. Even so, incremental improvements in the overall quality of models produced are likely to facilitate the selection of a good enough model, thus supporting the incorporation of co-evolution into fragment library generation as a viable technique to improve template-free modelling.

Acknowledgements

The authors would like to acknowledge the Oxford Protein Informatics Group for their intellectual input.

Funding

SHPdO and CMD have received funding from the Engineering and Physical Sciences Research Council (EP/G037280/1). The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

References

- Abbass, J., and Nebel, J.C. (2015). Customised fragments libraries for protein structure prediction based on structural class annotations. *BMC bioinformatics*, **16**(1), 136.
- Altschul, S. F., Gish, W., Miller, W., Myers, E. W., and Lipman, D. J. (1990). Basic local alignment search tool. *Journal of molecular biology*, **215**(3), 403–410.

- Baeten, L., Reumers, J., Tur, V., Stricher, F., Lenaerts, T., Serrano, L., Rousseau, F., and Schymkowitz, J. (2008). Reconstruction of protein backbones from the brix collection of canonical protein fragments. *PLoS Comput Biol*, **4**(5), e1000083.
- Bhattacharya, D., Adhikari, B., Li, J., and Cheng, J. (2016). Fragsion: ultra-fast protein fragment library generation by iohmm sampling. *Bioinformatics*, **32**(13), 2059–2061.
- de Oliveira, S. H., Shi, J., and Deane, C. M. (2015). Building a better fragment library for de novo protein structure prediction. *PLoS one*, **10**(4), e0123998.
- de Oliveira, S. H. P., Shi, J., and Deane, C. M. (2017a). Comparing co-evolution methods and their application to template-free protein structure prediction. *Bioinformatics*, **33**(3), 373–381.
- de Oliveira, S. H. P., Shi, J., and Deane, C. M. (2017b). Sequential search leads to faster, more efficient fragment-based de novo protein structure prediction. *Bioinformatics - In Press*.
- Faraggi, E., Zhang, T., Yang, Y., Kurgan, L., and Zhou, Y. (2012). Spine x: improving protein secondary structure prediction by multistep learning coupled with prediction of solvent accessible surface area and backbone torsion angles. *Journal of computational chemistry*, **33**(3), 259–267.
- Fox, N. K., Brenner, S. E., and Chandonia, J.-M. (2014). Scope: Structural classification of proteinsâ€”extended, integrating scop and astral data and classification of new structures. *Nucleic acids research*, **42**(D1), D304–D309.
- Gront, D., Kulp, D. W., Vernon, R. M., Strauss, C. E., and Baker, D. (2011). Generalized fragment picking in rosetta: design, protocols and applications. *PLoS one*, **6**(8), e23294.
- Handl, J., Knowles, J., Vernon, R., Baker, D., and Lovell, S. C. (2012). The dual role of fragments in fragment-assembly methods for de novo protein structure prediction. *Proteins: Structure, Function, and Bioinformatics*, **80**(2), 490–504.
- Heffernan, R., Paliwal, K., Lyons, J., Dehzangi, A., Sharma, A., Wang, J., Sattar, A., Yang, Y., and Zhou, Y. (2015). Improving prediction of secondary structure, local backbone angles, and solvent accessible surface area of proteins by iterative deep learning. *Scientific reports*, **5**.
- Jones, D. T., Singh, T., Kosciolk, T., and Tetchner, S. (2014). Metapsicov: combining coevolution methods for accurate prediction of contacts and long range hydrogen bonding in proteins. *Bioinformatics*, **31**(7), 999–1006.
- Kalev, I. and Habeck, M. (2011). Hhfrag: Hmm-based fragment detection using hhpred. *Bioinformatics*, **27**(22), 3110–3116.
- McGuffin, L. J., Bryson, K., and Jones, D. T. (2000). The psipred protein structure prediction server. *Bioinformatics*, **16**(4), 404–405.
- Moult, J., Fidelis, K., Kryshtafovych, A., Schwede, T., and Tramontano, A. (2016). Critical assessment of methods of protein structure prediction: Progress and new directions in round xi. *Proteins: Structure, Function, and Bioinformatics*, **84**(S1), 4–14.
- Orengo, C.A., Michie, A.D., Jones, S., Jones, D.T., Swindells, M.B., and Thornton, J.M. (1997). CATH—a hierarchic classification of protein domain structures. *Structure*, **5**(8), 1093–1109.
- Ovchinnikov, S., Park, H., Varghese, N., Huang, P.-S., Pavlopoulos, G. A., Kim, D. E., Kamisetty, H., Kyrpides, N. C., and Baker, D. (2017). Protein structure determination using metagenome sequence data. *Science*, **355**(6322), 294–298.
- Raman, S., Vernon, R., Thompson, J., Tyka, M., Sadreyev, R., Pei, J., Kim, D., Kellogg, E., DiMaio, F., Lange, O., *et al.* (2009). Structure prediction for casp8 with all-atom refinement using rosetta. *Proteins: Structure, Function, and Bioinformatics*, **77**(S9), 89–99.
- Shen, Y., Picord, G., Guyon, F., and Tuffery, P. (2013). Detecting protein candidate fragments using a structural alphabet profile comparison approach. *PLoS one*, **8**(11), e80493.
- Trevizani, R., Custódio, F. L., dos Santos, K. B., and Dardenne, L. E. (2017). Critical features of fragment libraries for protein structure prediction. *PLoS one*, **12**(1), e0170131.
- Wang, T., Yang, Y., Zhou, Y., and Gong, H. (2017). Lrfraglib: an effective algorithm to identify fragments for de novo protein structure prediction. *Bioinformatics*, **33**(5), 677–684.
- Xu, J. and Zhang, Y. (2010). How significant is a protein structure similarity with tm-score= 0.5? *Bioinformatics*, **26**(7), 889–895.
- Zhang, Y. and Skolnick, J. (2007). Scoring function for automated assessment of protein structure template quality. *Proteins: Structure, Function, and Bioinformatics*, **68**(4), 1020–1020.
- Zhou, X., Alber, F., Folkers, G., Gonnet, G. H., and Chelvanayagam, G. (2000). An analysis of the helix-to-strand transition between peptides with identical sequence. *Proteins: Structure, Function, and Bioinformatics*, **41**(2), 248–256.