# Predicting the structure of large protein complexes using AlphaFold and sequential assembly

Patrick Bryant[1,2]*, Gabriele Pozzati[1,2], Wensi Zhu[1,2], Aditi Shenoy[1,2], Petras Kundrotas[1,3] and Arne Elofsson[1,2]

[1]Science for Life Laboratory, 172 21 Solna, Sweden
[2]Department of Biochemistry and Biophysics, Stockholm University, 106 91 Stockholm, Sweden
[3]Center for Computational Biology, The University of Kansas, Lawrence, KS 66047, USA

*Corresponding author, email: patrick.bryant@scilifelab.se

## Abstract

AlphaFold and AlphaFold-multimer can predict the structure of single- and multiple chain proteins with very high accuracy. However, predicting protein complexes with more than a handful of chains is still unfeasible, as the accuracy rapidly decreases with the number of chains and the protein size is limited by the memory on a GPU. Nevertheless, it might be possible to predict the structure of large complexes starting from predictions of subcomponents. Here, we take a graph traversal approach to assemble 175 protein complexes with 10-30 chains using predictions of subcomponents. We compute paths through a complex graph constructed of subcomponents using Monte Carlo Tree Search and assemble these in a stepwise fashion. Using subcomponents predicted from all possible trimeric interactions, 88 complexes (50%) are assembled to completion. We create a scoring function, mpDockQ, that can distinguish if assemblies are complete and predict their accuracy. Selecting complete complexes with TM-score ≥0.9 at FPR 10% using mpDockQ results in 23 complexes with a median TM-score of 0.92. The complete assembly protocol, starting from the sequences, is freely available at: https://gitlab.com/patrickbryant1/molpc

**Keywords**
Protein structure prediction
AlphaFold
Complex assembly
Markov Chain Tree Search

# Introduction

Large protein complexes govern many cellular processes, performing complicated tasks such as mRNA splicing[1], protein degradation[2] or assisting protein folding[3]. By incorporating protein-interaction information from many co-purification experiments, the human protein complex map, hu.MAP 2.0[4], provides a set of 4,779 complexes with more than two chains. However, only 83 of these complexes are present in PDB. There are only 372 structurally resolved human protein complexes with over two chains, and of the 3130 eukaryotic core complexes in CORUM[5] only 800 have homologous structures covering all chains in PDB, suggesting a gap in our structural knowledge of protein complexes.

In total, there are only 265 hetero and homomeric, non-redundant complexes in the PDB with 10-30 chains. Although it is unknown how many large complexes may exist, following the relationship between the known human complexes from hu.MAP and the structural coverage of these, one can extrapolate that there may indeed be a low structural coverage across different species.

There are at least three approaches[6] for modelling the structure of protein complexes, template-based modelling[7], shape complementarity docking[8] and integrative modelling[9,10]. Template-based modelling and docking methods have recently been shown to be outperformed by a combined fold and docking methodology using AlphaFold[11] for dimeric complexes, even if the bound form of each monomer is known[12]. Further, few docking programs handle more than two protein chains, i.e. these methods are not suitable for building large complexes with no close homology to known complexes. There is currently (to our knowledge) no available docking benchmark for complexes with more than two chains, and previous studies only report results on a few examples[13].

Assembling large protein complexes with integrative modelling generally requires electron density maps or other experimental information to guide the assembly process[9],[14]. This type of guided assembly is typically based on a Markov process[9] or Gaussian mixture models[15], where many different potential configurations are explored and scored. This process makes it possible to assemble complexes with up to 1000 protein chains[16]. However, obtaining electron density maps can be very difficult, as some protein complexes are hard to express, purify and crystallise. Still, many recent assemblies of large protein complexes exist, such as the human nuclear pore complex[17] and 26S proteasome[18].

The only deep learning method primarily designed to predict the structure of more than two protein chains is AlphaFold-multimer[19]. This method has been trained on proteins of up to nine chains or 1536 residues and can predict complexes of up to a few thousand residues, where memory limitations come into play. However, the performance declines rapidly for proteins with over two chains (Supplementary figures 1 and 2). Predicting the structure of larger complexes is thereby currently not feasible. An alternative approach could be to predict the structure of subcomponents of large complexes and then assemble them. We have earlier shown that it is possible to manually assemble large complexes from dimers in a few cases [20].

In vivo, all components of large protein complexes do not assemble simultaneously, but stepwise[21], due to the presence of homologous protein chains and potential interfaces that need to be buried before subsequent chains can be added. Here, we explore the limitations of AlphaFold for predicting protein complexes with 10-30 chains and create a graph-traversal algorithm that excludes overlapping interactions, making it possible to assemble large protein complexes in a stepwise fashion.

# Results and Discussion

## Complex assembly

To analyse the possibility of assembling large protein complexes, we extracted all high-resolution non-redundant complexes from the PDB with more than nine chains, not containing nucleic acids or interactions from different organisms (175 in total). We start by analysing the possibility to assemble these protein complexes if all pairs of interactions between protein chains are known. Using either AlphaFold-multimer[19] (AFM) or the FoldDock protocol[12] using AlphaFold[11] (AF), we predict the structure of all unique pairs of interacting protein chains as subcomponents and create assembly paths, described below, from these.

As an example, the assembly of 6ESQ (acetoacetyl-CoA thiolase/HMG-CoA synthase complex) is shown in Figure 1, using subcomponents predicted with AFM. The process starts from the two dimers, AC and CH, creating the trimer ACH through superposition using the chain C present in both dimers. Next, chain L is added through a connection with H (superposition using chain H); after that, chain J through a connection with L, this process then continues until the entire complex is assembled according to the outlined path.

During the first part of this paper, we assume that the interaction graph is known, i.e. we limit the assembly paths only to include interactions existing in the complex. Although this is a simplification, the number of assembly paths is still huge, and it could at least be theoretically possible to obtain this information from other types of experiments[22] or predictions[4]. Next, potential assembly paths are created by starting at a randomly selected chain and adding all possible connections through superposition. Often overlaps occur among the predictions due to imperfect subcomponents, resulting in that, e.g. atoms from chains B and C occupy the same spatial position in a given complex ABCD. Therefore, an assembly path is discontinued when over half of the alpha carbons from two different chains are within 5Å from each other. An assembly path is complete when all chains in a complex can be linked together. For 6ESQ, the assembly results in a model with a TM-score of 0.96.

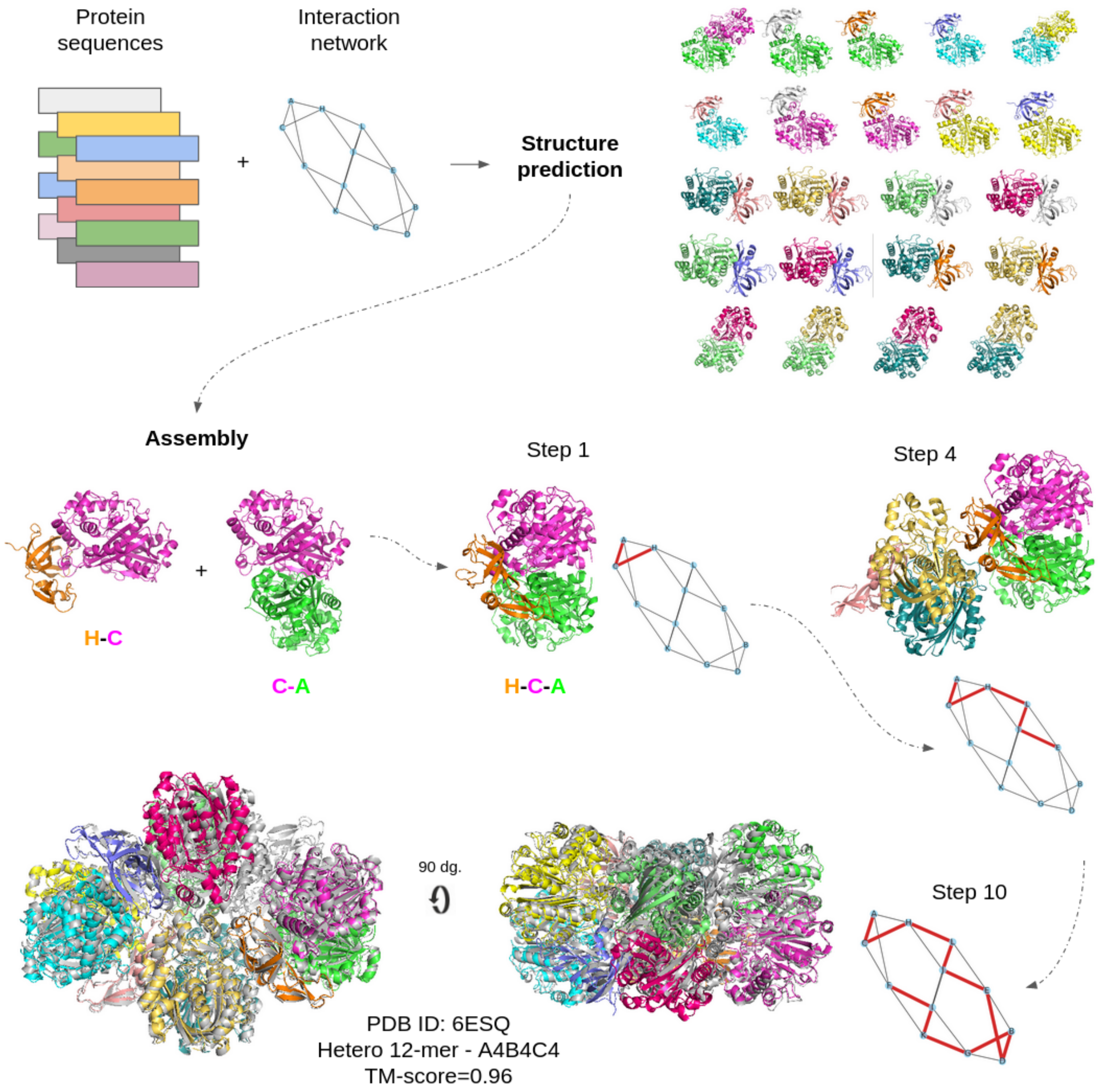


**Figure 1:** Assembly principle for the complex 6ESQ (acetoacetyl-CoA thiolase/HMG-CoA synthase complex). Starting from protein sequences from each chain and the interaction network, the structure of all interacting chains is predicted. From these predictions, an assembly path is constructed using the predictions as a guide. In each step, a new chain is added through a network edge resulting in a sequential construction of the complex. The taken path is outlined in red. The complete assembly is shown in overlap with the native complex (grey). The resulting TM-score is 0.96 using subcomponents from AFM (shown) and 0.92 using FoldDock (not shown).

## Monte Carlo Tree Search

Due to the high number of possible paths to explore, searching all paths is unfeasible. Therefore, we search for an optimal path using Monte Carlo Tree Search[23] (MCTS, Figure 2), which has been applied successfully to solve a variety of game-related problems[24,25]. Starting from a randomly selected chain (node); chains are added at random to expand the path, thereby creating new nodes. From these expansions, complete assemblies are simulated. Simulations are stopped when no additional subunits can be added, see methods. The simulated assemblies are scored by their cumulative mpDockQ (multiple-interface predicted DockQ; average interface plDDT times the logarithm of the number of interface contacts, Methods section) score, and the scores are backpropagated to yield support for the previous selections. The path with the most support is selected, creating a complex that is the most likely to be correct. Due to the statistical nature of the search procedure, no aspect of a specific complex is being “learned” in the backpropagation, which means that all 175 complexes can be used for the evaluation.

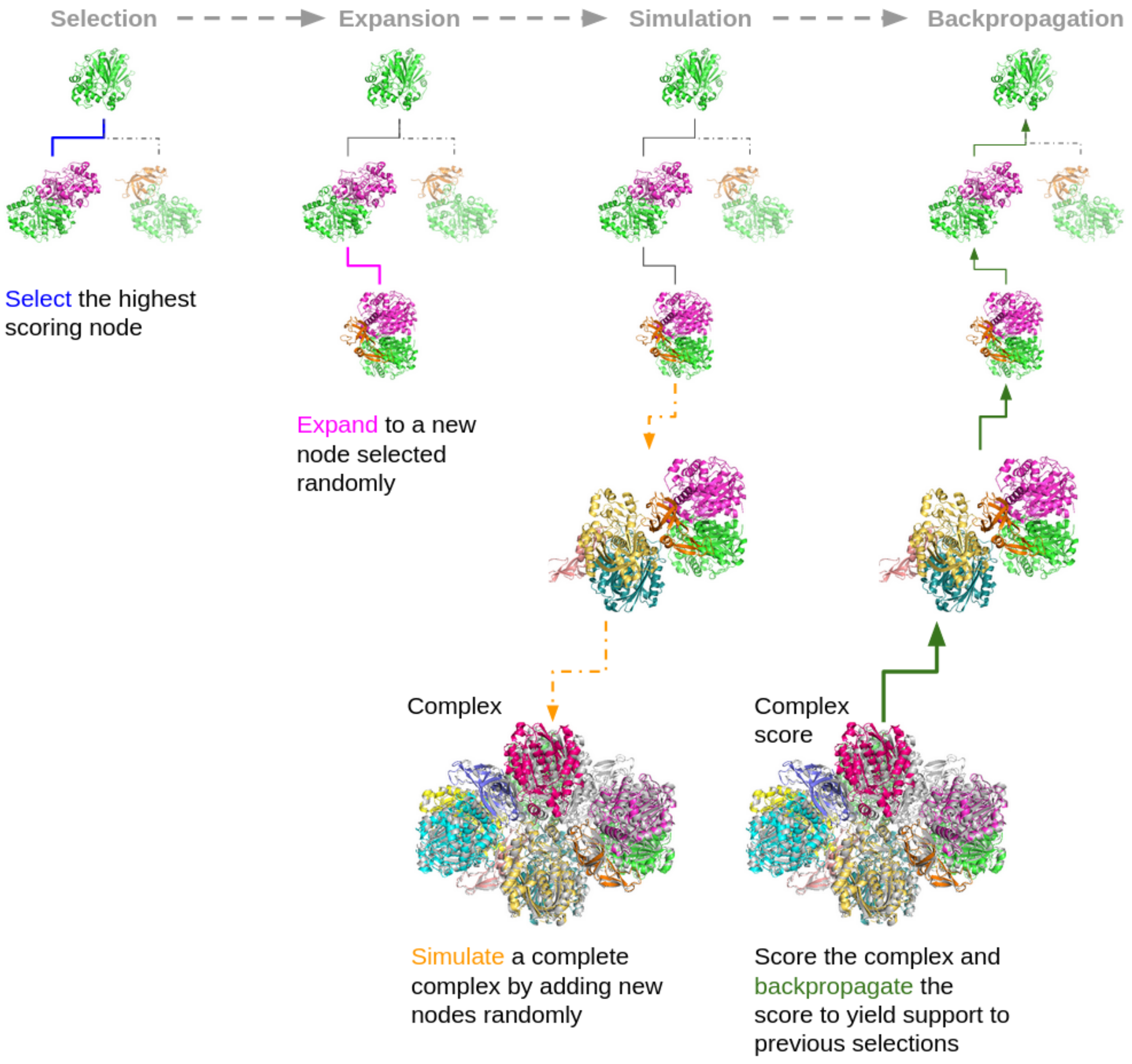

Selection
Expansion
Simulation
Backpropagation
Select the highest scoring node
Expand to a new node selected randomly
Complex
Simulate a complete complex by adding new nodes randomly
Complex score
Score the complex and backpropagate the score to yield support to previous selections

**Figure 2.** Monte Carlo Tree Search. Starting from a node (subcomplex), a new node is selected based on the previously backpropagated scores. From this node, a random node is

added (expansion). A complete assembly process is then simulated by adding nodes randomly until an entire complex is assembled or a stop caused by too much overlap is reached. The complex is scored and the score is backpropagated to all previous nodes, which yields support to the previous selections. The end result is that the nodes that are most likely to result in high scoring complexes are joined in a path containing all chains.

## AFM vs AF using pairwise interactions

Four and fifteen out of 175 complexes could be assembled to completion using known pairwise interactions with AFM and FoldDock (AF) respectively (Figure 3a). All assemblies based on FoldDock perform on par or better than those based on AFM. The results suggest that if a complete path can be found, it is likely to obtain a high TM-score (median=0.77) using the FoldDock pipeline.

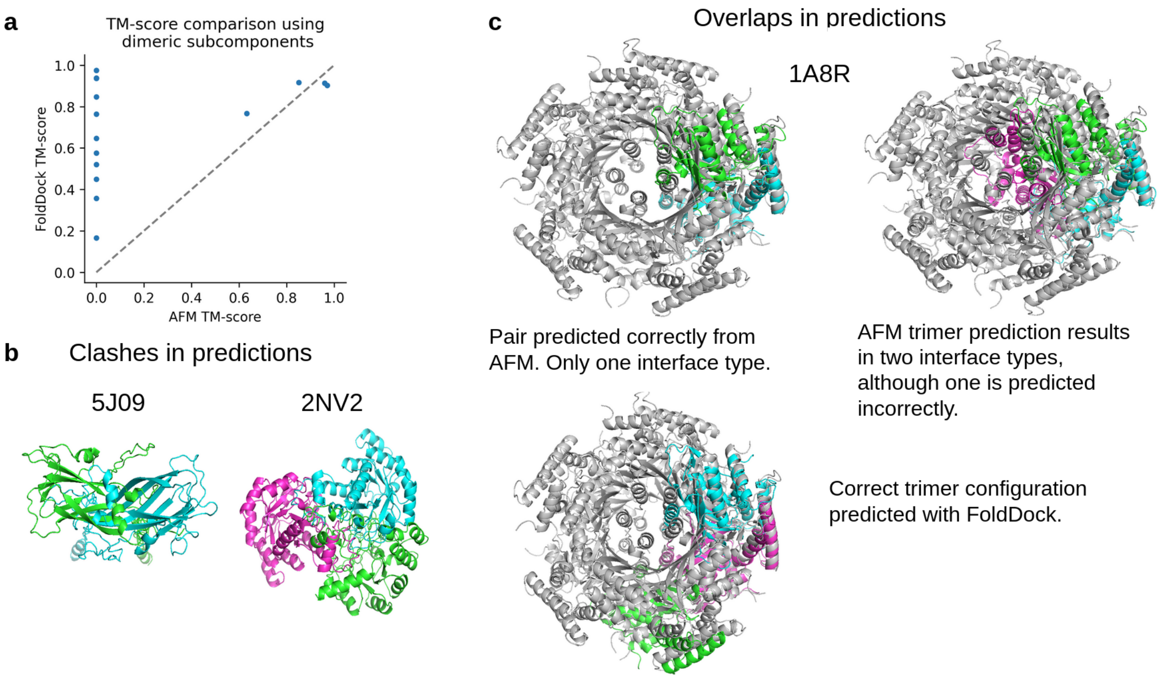
The AFM modelling pipeline often causes clashes (Figure 3b), resulting in atoms from different chains occupying the same positions. Defining clashes as atoms from different protein chains being within 1 Ångström from each other, 61.7% (403/653) of the AFM dimers contain clashes and 6.3% (41/656) for the FoldDock pipeline. This is likely the major reason that the assembly using AF is more likely to succeed.

For the unique trimers predicted with AFM, the clashes are even more frequent than for the dimers, 87.1% (1076/1236), while for the FoldDock pipeline, the corresponding fraction is 23.2% (520/2242). The high proportion of clashes and suboptimal performance obtained using AFM lead us to abandon this method and continue with only the FoldDock protocol in all subsequent analyses. This is also the reason why all possible trimers were not predicted with AFM.

## Limited conformational sampling in dimers

During assembly, the additive relative orientation of different protein chains can result in overlaps, due to predictions not being entirely correct. One cause of overlaps during the assembly process is due to that only the most stable conformation is favoured in the predictions, resulting in wrong interfaces in some dimers. As an example, we can investigate 1A8R, a homo 10-mer. When predicting unique pairwise interactions, only one type of dimeric conformations can be found, but in the complex, each chain has at least two different types of interactions with other chains This means that it is impossible to assemble the entire complex from the predicted dimers.

The overlapping interfaces can, here, be circumvented by predicting trimeric interactions, thereby generating alternative interfaces. In the case of 1A8R, the trimer is wrongly predicted for AFM (Figure 3c), resulting in the third chain (magenta) ending up within the complex (grey). This trimer prediction also contains clashes. If correctly predicted, the magenta chain should be above or below the green or cyan chains, as can be seen from the FoldDock prediction.



Clashes in predictions

Overlaps in predictions

Pair predicted correctly from AFM. Only one interface type.

AFM trimer prediction results in two interface types, although one is predicted incorrectly.

Correct trimer configuration predicted with FoldDock.

**Figure 3. a)** TM-scores for the **15** complexes that could be assembled to completion using pairwise interactions with AFM and FoldDock (AF) respectively **b)** Clashes in the predictions are shown for pairwise and trimeric interactions belonging to the complexes 5J09 and 2NV2 respectively. These overlaps are due to the fact that the AFM modelling pipeline does not consider clashes. **c)** The overlaps in the predictions are due to the most stable configuration being predicted. PDB ID 1A8R, is a homo 10-mer, containing only one unique chain (A10). This means that all interactions are between copies of this chain. When predicting pairwise interactions, only one conformation is found. This can be circumvented by predicting trimeric interactions. In this case, however, the trimer is wrongly predicted, resulting in the third chain (magenta) ending up within the complex (grey). Notable is that the magenta chain is also clashing with the two others. In the FoldDock prediction, the trimer configuration is predicted correctly.

## Complex assembly using trimeric interactions

Using the FoldDock protocol with AF, all known trimeric interactions were predicted for all complexes. From these, all dimeric interactions were extracted and assembly paths were constructed as previously. Out of 175 complexes, 58 (33 %) could be assembled to completion with a median TM-score of 0.80 (Figure 4a). In comparison with the guided dimer TM-scores (median=0.77 for 15 complexes), the guided trimer approach results in 46 additional complexes and a higher median TM-score, while three could not be assembled. When both approaches are successful, the results are similar.

In many cases, the exact interactions of all protein chains are not known, only that a set of chains interact [4]. After applying the method in protein complexes where we have assumed knowledge of interactions, we now turn to the more challenging (and realistic) problem of predicting the complexes without knowing interactions (no-knowledge approach). In addition to the problem of possibly incorrectly identified interacting pairs, this also increases the number of possible erroneous paths to be sampled. We find that 88/175 (50%) of structures can be assembled with a median TM-score of 0.51 (Figure 4a) using all possible trimeric interactions. 39 additional complexes are obtained, although 9 are missing compared to the known trimer approach. When both trimer approaches have complete assemblies (n=51), the median scores are 0.76 and 0.80 for the no-knowledge and known trimer approaches, respectively. To have knowledge of interactions thereby results in higher scores overall.

To analyse the possibility to distinguish when a complex is assembled to completion and has a high TM-score (≥0.9, n=22), we analyse the ROC curve (Figure 4b) as a function of the average interface plDDT (predicted lDDT from AF), the number interface residues, contacts and interactions between chains normalised with the number of chains in each complex and the average interface plDDT times the logarithm of the number of interface contacts. The plDDT $\cdot$ log(contacts) results in the highest AUC value (0.84) as well as higher TPRs at low FPRs, which is why it is preferred. We fit a sigmoidal curve using the plDDT $\cdot$ log(contacts) and the TM-score, creating the mpDockQ score (multiple-interface predicted DockQ, see Methods section). When the mpDockQ tends to be high, so does the TM-score and % completion of the complex (Figure 4c). This suggests that mpDockQ can be used to both select for when a complex is complete and how accurate it is.

Figure 4d shows examples of complexes assembled using all possible trimeric subcomponents selected at FPR 10% (TPR=55%, 23 complexes) with mpDockQ. Obtaining complete complexes with very high TM-scores (≥0.9) is the most important, as large complexes that are not entirely correct are not likely to provide biologically meaningful insights. The native and predicted complexes are in a structural superposition, portrayed in grey and coloured by chain, respectively. The median TM-scores for this selection are 0.92 and 0.92 using all 23 and only the complete (18) complexes in the selection, respectively. 1IWA, a heterodimeric complex with 16 chains has a TM-score of 0.93 and is completely assembled and so is 5OVS (homodimeric complex with 14 chains, TM-score=0.99) and 1DW9 (homodimeric complex with 10 chains, TM-score=0.96). The complex 1RBL (heterodimeric, 16 chains) was not assembled to completion, one chain is missing. The part that could be assembled has a good correspondence with the native structure (TM-score=0.96).

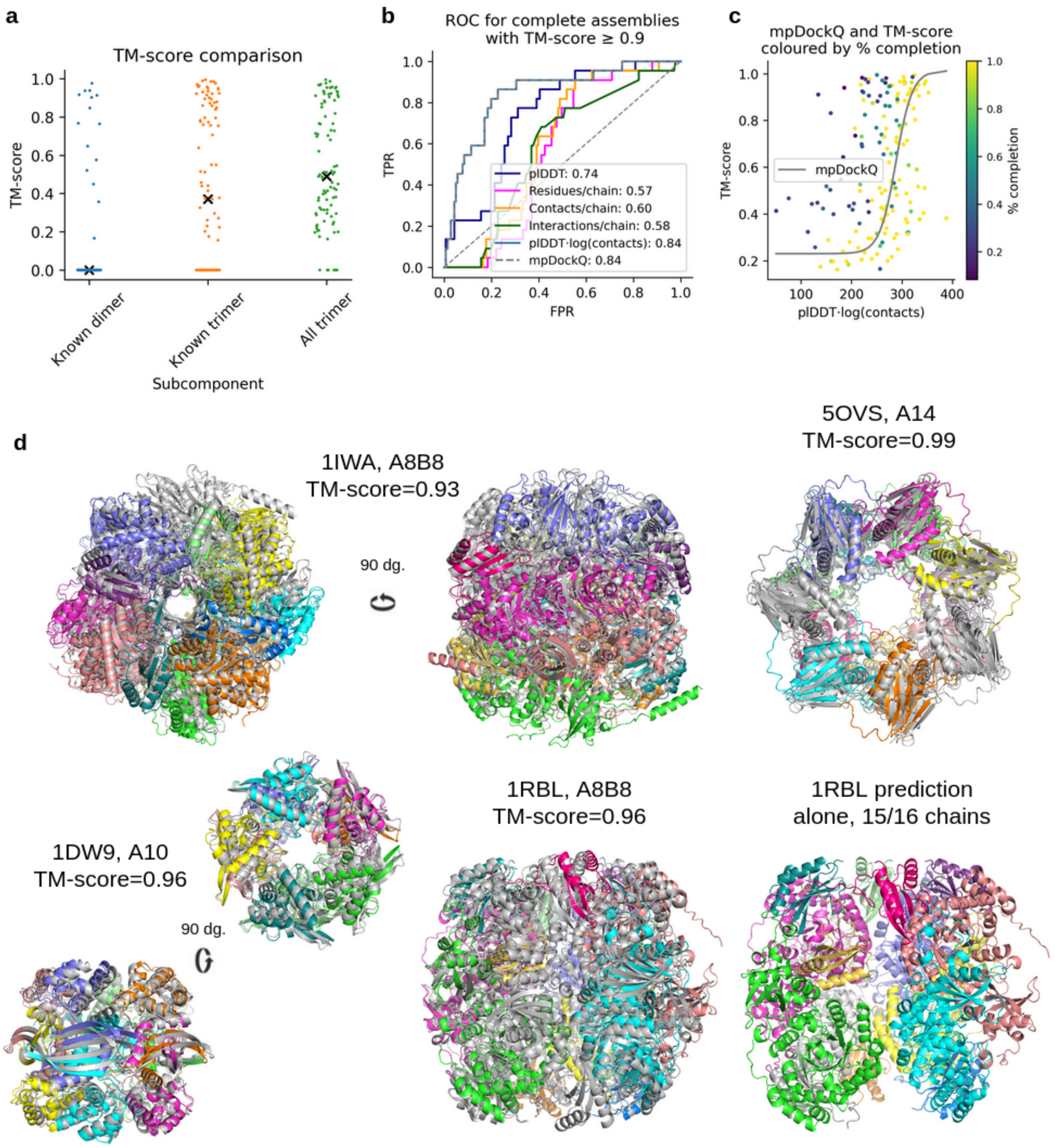


**Figure 4. a)** TM-scores for the complexes that could be assembled to completion using FoldDock (AF) and known dimeric, known trimeric and all trimeric subcomponents, respectively. The complete set of complexes from the three different approaches (n=97) is shown, with scores of zero representing missing complexes for each approach. The points display the TM-score of the individual complexes and the black “x” marks the median scores (0.00, 0.37 and 0.49 using known dimers, trimers and all trimers, respectively). The reason the median scores are low is due to the missing complexes between the approaches. Considering only the successful assemblies using known dimers, trimers and all trimers the median scores are 0.77, 0.80 and 0.51, respectively. **b)** Complex scoring using all trimers as subcomponents. ROC curve, where positives (n=22) are complete assemblies of TM-score ≥0.9, as a function of the average interface plDDT, the number of interface residues and contacts normalised with the number of chains in each complex, the average interface plDDT times the logarithm of the number of interface contacts and mpDockQ (see c). The

best separators are plDDT · log(contacts) and mpDockQ, both with AUC 0.84. **c)** TM-score vs the best separator in b), plDDT · log(contacts), coloured by the fraction of completion for the assemblies. The solid grey line represents a sigmoidal fit creating the mpDockQ score (see Methods section). When the mpDockQ tends to be high, so does the TM-score and % completion of the complex. This suggests that mpDockQ can be used to both select for when a complex is complete and how accurate it is. **d)** Examples of assembled complexes selected at FPR 10% (TPR=55%) with mpDockQ, using all trimeric interactions for assembly (the no-knowledge approach). The native and predicted complexes are in structural superposition, portrayed in grey and coloured by chain, respectively. 1IWA, a heterodimeric complex with 16 chains has a TM-score of 0.93 and is completely assembled and so is 5OVS (homodimeric complex with 14 chains, TM-score=0.99) and 1DW9 (homodimeric complex with 10 chains, TM-score=0.96). The complex 1RBL (heterodimeric, 16 chains) was not assembled to completion, one chain is missing. The part that could be assembled has a good correspondence with the native structure (TM-score=0.96).

## Aspects affecting the assembly

To answer why some complexes can be assembled with high accuracy and others not, we analyse the kingdom, the number of total chains, the oligomeric type (hetero or homomer), the number of effective sequences (Neff) and the subcomponent accuracy for each complex (Figure 5). We performed this analysis for the complexes assembled with known trimers due to the high redundancy of subcomponents in the blind approach. Bacteria is the most abundant kingdom and displays the most complete assemblies (29/85) with a median TM-score of 0.85 (Figure 5a). Eukaryota, Viruses and Archaea have 17/63, 8/12 and 4/15 with median TM-scores of 0.75, 0.44 and 0.92, respectively.

Most complete assemblies have fewer chains and are of homomeric type (Figures 5b and c), although the spread in TM-score is large. The TM-scores are higher for the complexes with higher (over 500) average Neff values, which corresponds well with findings for heterodimeric complexes[12] (Figure 5d). When analysing how far towards completion the assemblies go one finds that most complexes are 90-100% complete (Figure 5e). There appears to be a weak decreasing trend in TM-score with completion suggesting that smaller subcomplexes may be accurate, although the complete complex cannot be assembled. The average TM-score of the subcomponents (Figure 5f) provides the most evident explanation of when an assembled complex is accurate. When the subcomponents display high accuracy, so does the assembled complex.

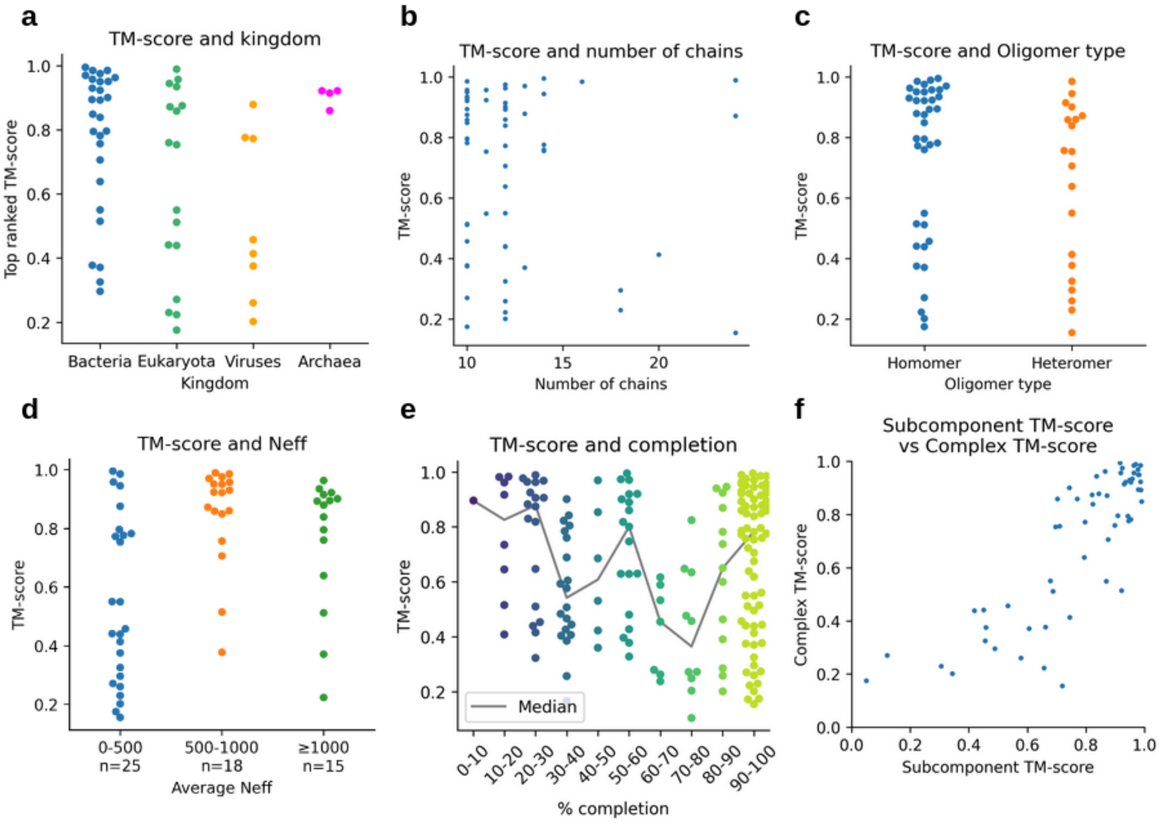


**Figure 5.** Analysis of the assemblies using known trimers.
**a)** TM-score per kingdom for the complete assemblies (n=58). Bacteria is the kingdom with the most complete assemblies (n=29) and reports a median TM-score of 0.85. Eukaryota (n=17), Viruses (n=8) and Archaea (n=4) have median TM-scores of 0.75, 0.44 and 0.92, respectively. **b)** TM-score vs the number of chains for the complete assemblies (n=58). Most complexes have fewer chains and the spread in TM-score is large. **c)** TM-score vs oligomer type, homomer (n=38 out of 114) or heteromer (n=20 out of 61), using complete assemblies. The homomeric complexes have a median TM-score of 0.86 and the heteromeric 0.73.
**d)** TM-score and Neff. The TM-scores are higher for the complexes with over 500 in average Neff value. **e)** TM-score and completion. The coloured points represent the scores within bins of 10% and the grey line shows the median for each bin. Most complexes are 90-100% complete and there appears to be a weak decreasing trend of the accuracy with completion.
**f)** Average TM-score of subcomponents vs TM-score of the whole complex for the complete assemblies (n=58). When the subcomponents display high accuracy, so does the assembled complex (SpearmanR=0.80).

## Conclusions and Limitations

To predict the structure of large complexes directly from sequence information is currently a difficult challenge. Here, we present a novel method that suggests that one possible approach is to predict subcomponents and assemble them into a larger complex. AlphaFold-multimer (AFM) is currently the only method primarily designed to predict the structure of more than one protein chain directly from sequence information without using templates. We show here that when predicting subcomponents with AFM, most of them contain clashes, resulting in that AFM is unable to be used in the pipeline. However, we expect this issue to be resolved in the future, making AFM predictions equally interchangeable. The FoldDock protocol based on AlphaFold (AF) is less affected by this issue. AF was not trained for this purpose either, yielding support to the robustness of this method.

More complexes are assembled to completion when using known trimeric subcomponents and the median TM-score is higher than with dimeric subcomponents. Assuming no knowledge of interactions with trimeric subcomponents results in the most complete complexes, although the median TM-score is lower. The created scoring function mpDockQ can distinguish if assemblies are complete and predict their accuracy, making this blind approach feasible. We find that when the subcomponents are accurately predicted using known trimers, so are the complete assemblies. This suggests it is possible to assemble complexes as long as their subcomponents are accurate.

Currently, not all trimers can be folded using two NVIDIA A100 Tensor Core GPUs with 40Gb of RAM. Roughly, the limit of AF (and AFM) on this computational platform appears to be 3000 residues, and 73/175 (42%) of all complexes are larger than that. Depending on the speed of computational development, an assembly approach to complex prediction may be needed even for proteins with much fewer than 10 chains.

## Future outlook

Here, we have shown that it is possible to assemble large complexes using only protein sequence information and stoichiometry. Modelling large complexes in parts and assembling them converts the problem of predicting large complexes to the prediction of their subcomponents. This suggests an exciting future where models of all components in entire cells, and eventually entire cells themselves, may be modelled.

One limitation for predicting protein complexes using the approach proposed here is stoichiometry. It is often not known how many copies there are of a protein in a given complex, a requirement for the assembly. Once this limitation is overcome either by computational or experimental studies of complexes, it will be possible to assemble many different protein complexes, possibly in novel configurations.

# Acknowledgements

Financial support: Swedish Research Council for Natural Science, grant No. VR-2016-06301 and Swedish E-science Research Centre and from Knut and Alice Wallenberg foundation. Computational resources: Swedish National Infrastructure for Computing, grants: SNIC 2021/5-297, SNIC 2021/6-197 and Berzelius-2021-29. All financial support and computational resources were received by A.E.

# Author contributions

PB designed and performed the studies. PB and AE performed the main analysis. GP, WZ, AS and PB set up the modelling infrastructure with AFM and analysed the performance of AFM for the smaller complexes of 3-6 chains. PK provided the smaller structures of 3-6 chains. PB wrote the first draft of the manuscript and prepared all figures which were later edited and improved by AE and PB. AE obtained funding.

# Competing interests

The authors have no competing interests.

# Availability

All information needed to repeat the study presented here as well as the pipeline itself is available at: https://gitlab.com/patrickbryant1/molpc. Large files and MSAs will be made available through a figshare repository.

# Methods

## Non-redundant complexes with 10-30 chains from the PDB

Since AlphaFold-multimer has a limit of 9 chains or 1536 residues [19] in its training and testing data, and there is no available method validated for modelling larger complexes, we obtained all complexes with 10-30 chains from the PDB to extend the current limit (Figure 6a). First, we selected all complexes not containing nucleic acids with ≤3Å resolution and experimental method X-ray crystallography or Electron Microscopy (1216). From these complexes, we require that all chains originate from the same organism (1027). We cluster all sequences from the complexes on 20% sequence identity using MMseqs2 (version edb8223d1ea07385ffe63d4f103af0eb12b2058e) [26] using this command:

```
MMseqs2 easy-cluster fastafile outname /tmp  --min-seq-id 0.2 -c
0.8 --cov-mode 1
```

Using clustering, we ensure that no complex has all of its clusters overlapping with any other. We keep the complexes that have the most clusters, resulting in that subcomponents of larger clusters are removed (265).  E.g. if the sequences from complex 1 map to clusters A, B and C and those of complex 2 map to clusters A, B, C, and D, then complex 2 will be kept and complex 1 excluded. After the clustering, we ensure that no complex contains any chain shorter than 50 residues (193 complexes), to remove protein-peptide interactions. We then download the first biological assembly[27] from each complex and check that the reported stoichiometry is correct and that the PDB files do not contain discontinuous chains, resulting in a total of 175 complexes. The distribution of the number of chains can be seen in Figure 6b. Most complexes have 10-12 chains.

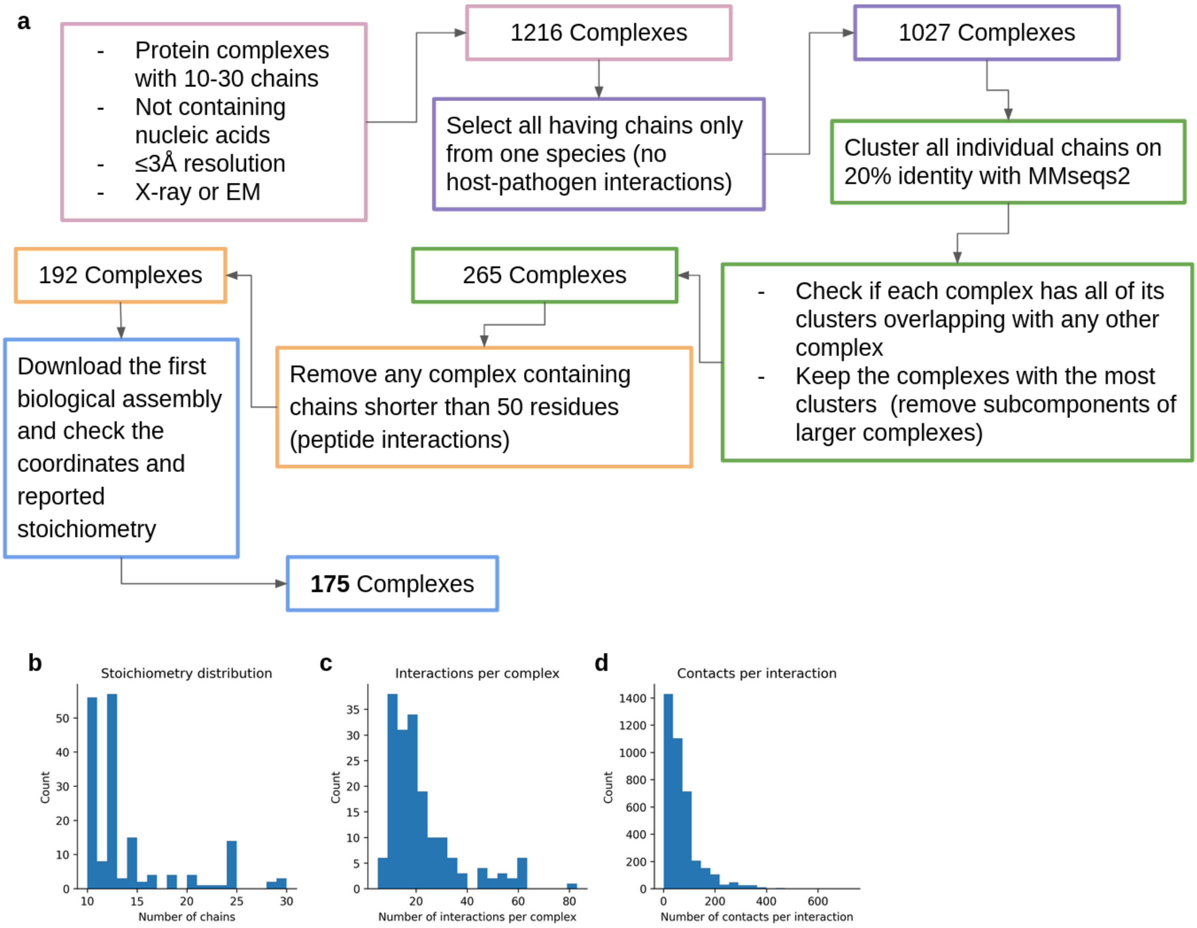

a
- Protein complexes with 10-30 chains
- Not containing nucleic acids
- ≤3Å resolution
- X-ray or EM
1216 Complexes
Select all having chains only from one species (no host-pathogen interactions)
1027 Complexes
Cluster all individual chains on 20% identity with MMseqs2
- Check if each complex has all of its clusters overlapping with any other complex
- Keep the complexes with the most clusters (remove subcomponents of larger complexes)
265 Complexes
Remove any complex containing chains shorter than 50 residues (peptide interactions)
192 Complexes
Download the first biological assembly and check the coordinates and reported stoichiometry
175 Complexes
b
Stoichiometry distribution
Count
0
10
20
30
40
50
10
15
20
25
30
Number of chains
c
Interactions per complex
Count
0
5
10
15
20
25
30
35
20
40
60
80
Number of interactions per complex
d
Contacts per interaction
Count
0
200
400
600
800
1000
1200
1400
0
200
400
600
Number of contacts per interaction

**Figure 6. a)** Outline of the data selection process.
**b)** Distribution of the number of chains for the 175 complexes. Most complexes have 10-12 chains **c)** Distribution of the number of interactions between all chains in a complex. On average there are 22 interactions per complex. **d)** Distribution of the number of contacts per interaction. On average there are 70 contacts per pair of interacting chains.

## Interaction network

To create interaction networks for the guided assembly of the complexes, interactions between different chains with CBs (CA for Glycine) within 8Å from each other were extracted. Interactions are defined when 10% of the beta carbons (alpha carbon for glycine) of the shortest of two different protein chains are within 8 Ångström from the other. On average, each interaction pair consists of 70 residue pairs and within each complex, there are 22 interacting pairs of chains (Figures 6c and 6d).

## Subcomponent and edge complexity

To assemble entire complexes, we predict all dimeric and trimeric interactions in a set of n chains.

The number of possible dimers follows:

$$D(n) = \frac{n!}{(n-2)!2!} = \frac{n(n-1)}{2} \qquad (i)$$

The number of possible trimers follows:

$$T(n) = \frac{n!}{(n-3)!3!} = \frac{n(n-1)(n-2)}{6} \qquad (ii)$$

From these dimers and trimers, we extract all edges (pairwise interactions). The number of edges in D(n) dimers is D(n) and in T(n) trimers:

$$E(n) = \frac{T(n)(T(n)-1)}{2} \qquad (iii)$$

## Structural predictions of dimeric and trimeric subcomponents

AlphaFold-multimer[19] was run with the standard settings. Four different MSAs are created by searching various databases with several genetic search programs. Using jackhmmer from HMMER3[28], three different MSAs are created through searching the databases Uniref90 v.2020_01[29], Uniprot v.2021_04[30] and MGnify v.2018_12[31]. The fourth MSA is created by searching the Big Fantastic Database[32] (BFD from https://bfd.mmseqs.com/) and uniclust30_2018_08[33] jointly with HHBlits[34] (from hh-suite v.3.0-beta.3 version 14/07/2017). By using the species (prokaryotes and eukaryotes) and genetic positional information (prokaryotes only), the results from the Uniprot search paired. All results from the other searches are instead block-diagonalized. All of the created MSAs (one paired and three block-diagonalized) are used to predict the structure of a protein complex.

The FoldDock protocol[12], based on AlphaFold[11], was run as well. This protocol creates two MSAs constructed from a single search with HHblits[34] version 3.1.0 against uniclust30_2018_08[33] using the options:

```
hhblits -E 0.001 -all -oa3m -n 2
```

The first of the two MSAs are constructed by extracting the organism identifiers (OX) from the resulting a3m file and pairing sequences using the top hit from each OX. The second is constructed by block diagonalizing the resulting a3m file. An extension to 3 chains was made here also, following the same pairing and block diagonalizing procedure as has been done for two chains. The folding was performed using AlphaFold model_1, 10 recycles and one ensemble structure. The  recycles refer to how many times the intermediate output is fed back into the network and the MSAs are resampled. The ensemble structure entails how many times the information within the network is processed before it is averaged.

The structural prediction was performed on two NVIDIA A100 Tensor Core GPUs each with 40 Gb of RAM. Three different sets of different subcomponents for the complexes were modelled, all known dimeric, all known trimeric and all possible trimeric subcomponents.

The unique dimer subcomponents of 653/656 and 656/656 could be predicted for AFM and FoldDock respectively. For FoldDock, 2242/2246 unique known trimers were predicted and for AFM 1236. The four that did not work using FoldDock had the error message “Cannot create a tensor proto whose content is larger than 2GB.”. The reason not all 2246 trimers were predicted using AFM is the high amount of clashes observed in the predicted ones (87%). Clashes result in unrealistic proteins due to the breaking of physical constraints, which led us to abandon this method. For the approach using all trimers, 8556/8561 unique subcomponents were successfully modelled. The five that did not work had the error message “Cannot create a tensor proto whose content is larger than 2GB.”.

## Path complexity

When considering all possible interactions in a complex, both dimeric and trimeric, one quickly realises that there are many possible paths that could connect all chains. Take the example of the maximum number of chains modelled here, 30. In the most extreme scenario, all of these are assumed to interact with each other. This means that starting at chain 1, it is possible to attach chain 2-30 (29 possibilities) and from these 28 possibilities for each node and so on.

If there are no overlapping interfaces in a complex of n’ nodes and E(n) edges, the number of unique paths that contain all nodes follow:

$$P(n) \ = n'^{(n'-2)},\ n' \geq 2 \qquad \textit{(iv)}$$

Note that n’ here are the number of nodes extracted from the predicted subcomponents, which are more than the number of unique nodes since e.g. the trimers ABC and ABD both contain the nodes A and B. Equation *(iv)* is exponential and thereby grows very fast. However, the overlaps will grow with the number of nodes as well, as it will be more likely to have overlapping interfaces with more edges.

According to equation *(ii),* there are $\frac{30(30-1)(30-2)}{6} = 4060$ possible trimers for a complex of 30 chains. For each trimer, there are three possible edges, resulting in $4060 \cdot 3=12180$ edges in total. This means that the number of effective nodes are more than the actual number of nodes. This is because e.g. chain A occurs many times in different trimers. E.g. ABC, ABD, ABE all have the possibility to have different interactions between A and B. Following equation *(iv)* there will be $30^{28} \approx 2.3 \cdot 10^{41}$ possible paths at the upper bound considering all dimers from 30 protein chains (and many more considering all trimers). This is a very large number that is not possible to search in a feasible amount of time with our available computational resources. However, it is very unlikely this number of paths has to be explored due to overlaps in the subassemblies.

When the subpaths that contain overlaps are excluded during assembly, the number of possible paths reduces quickly. Let’s assume there are only 3 possible interactions for each chain. Then the number of possible paths become much fewer, depending on how the network is connected. If all branches in a network contain unique chains (Figure **7**), there is in fact only one possible path that connects all chains. Still, there may be many possible paths to traverse to find this non-overlapping one that connects all chains. Therefore, we limit the number of paths searched at a given time point.

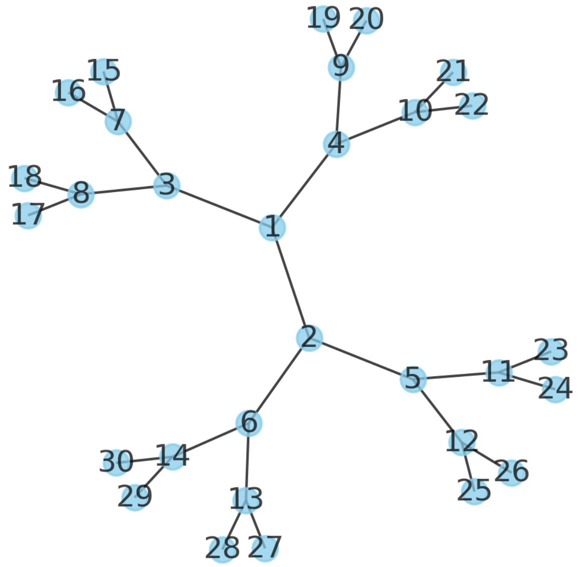


**Figure 7.** Branch network of 30 chains all connected to two other chains. There is only one path that connects all 30 chains (the network itself).

## Assembly procedure with Monte Carlo Tree Search

From the interactions in the predicted subcomponents, we add chains sequentially following a path through the interaction network (graph) constructed using Monte Carlo Tree Search (MCTS) [23]. MCTS applies a heuristic search method through a graph to find an optimal path (Figure 8). MCTS consists of four different steps named, selection, expansion, simulation and backpropagation. It has been shown that sampling random paths to completion from a certain node (simulation) informs the best action at a certain position. To add new chains to a path, we use BioPython's SVD Superimposer[35]. As an example, if two pairwise interactions are A-B and B-C, we assemble the complex A-B-C by superposing chain B from A-B and B-C and rotating the missing chain C to its correct relative position. The MCTS procedure is outlined accordingly:

1. Selection: start at a randomly chosen node 1 (e.g. chain A).
2. Expansion: obtain all edges $e^1$,..,$e^N$, deemed "children" to node 1 and create N different paths. Expand the new nodes added through the edges by randomly selecting new edges. If the new nodes do not have any edges, they are deemed "leaf nodes". In this case, the best scoring node according to equation (v) is selected and a new expansion is started from there. We expand all possibilities, ensuring convergence towards the best node selection at each position.
3. Simulation: add chains randomly to the path until the overlap criterium is obtained or the complex is complete. An overlap is defined as when over 50% of the alpha carbons in the shortest of two protein chains are within 5 Å from each other.
4. Backpropagation: score the simulated complex using equation (vi). Update all "parent nodes" with this score. The simulation and backpropagation together provide an estimate of how well the parent node performs in terms of creating a successful assembly path.

The best child nodes are selected using the upper confidence bound (UCB) accordingly:

$$UCB = V_i + 2\sqrt{\frac{\ln N}{ni}} \qquad (v)\ ,$$

Where $V_i$ is the average complex score (equation vi) of all nodes below node i, N is the number of times the parent node has been visited and $n_i$ is the number of times the node being scored has been visited. The MCTS procedure is continued until all chains are in complex or there are no more non-overlapping chains to add to the current path, after which the procedure is terminated.

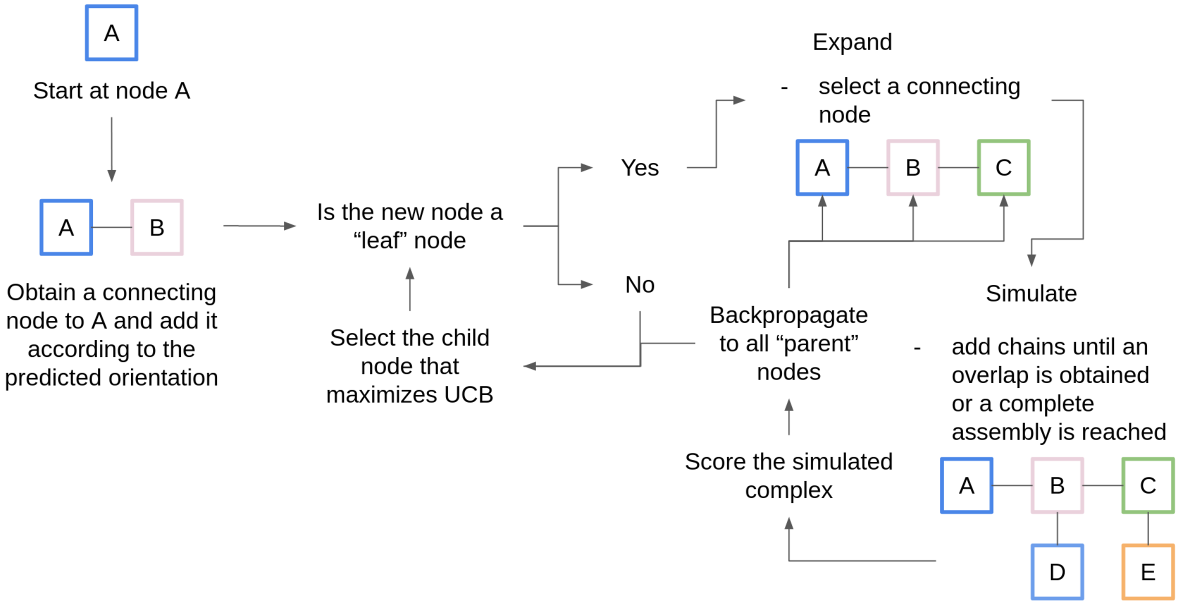

A
Start at node A
A
B
Obtain a connecting node to A and add it according to the predicted orientation
Is the new node a "leaf" node
Yes
No
Select the child node that maximizes UCB
Expand
- select a connecting node
A
B
C
Backpropagate to all "parent" nodes
Simulate
- add chains until an overlap is obtained or a complete assembly is reached
Score the simulated complex
A
B
C
D
E

**Figure 8.** Monte Carlo Tree Search (MCTS) procedure. Starting at node A, a connecting node (chain) is selected and added according to its predicted orientation. If this node is a "leaf" node (a node that has not been expanded before), an expansion is performed. During the expansion, a new node is added and from this an entire complex is simulated. The score from the simulation (equation vi) is backpropagated to all "parent" nodes of the expansion which is used to determine the UCB (equation v) and thus select the best possible path.

The complex 6LNI could not be assembled using trimers due to no interactions between the chains being present in the predictions. This protein is an amyloid protein and should thereby not occur naturally in the cell. For the assembly approach using all possible trimers, there are very many paths to assemble for some complexes. Ten additional complexes (1IRU, 1MFR, 1PCQ, 1S3Q, 1Z6O, 3OJ5, 6J0B, 6LQH, 6NHT and 6PYT) could not be assembled due to time constraints (48 hours on a 2.6 GHz processor).

## Scoring

We score the interfaces of the complexes being assembled in the MCTS using:

$$log_{10}(number\ of\ interface\ contacts) \cdot average\ interface\ plDDT \quad (vi)$$

, as done when calculating the pDockQ score[12]. This score for multiple interfaces, we deem “multiple-interface predicted DockQ” or mpDockQ. The interface contacts are taken as beta carbons (alpha carbons for Glycine) from two different protein chains being within 8Å from each other. These metrics are calculated for the entire interface of each chain, as in the DockQ[36] score for multiple interfaces. E.g. if chain A interacts with both chains B and C, the score is taken over both of these interfaces simultaneously. This is done for all interfaces and chains and summed over the entire complex. The complexes with the highest sums are favoured. Favouring complexes with higher scores, results in complexes with both larger interfaces and with more reliably predicted residues.

## Sigmoidal fit for mpDockQ

To create a continuous score for the multiple interface DockQ (mpDockQ), we fit a simple sigmoidal function towards the TM-score (Figure 4c) using the complete complexes assembled from trimeric subcomponents and “curve_fit” from SciPy v.1.4.1 [37] with the following sigmoidal equation:

$$mpDockQ = \frac{L}{1+e^{-k(x-x_0)}} + \mathrm{b} \qquad \text{(vii)}$$

, where x = average interface plDDT · log10(number of interface contacts) (equation vi) across all interfaces and we obtain L= 0.827, x0= 261.398, k= 0.036 and b= 0.221.

## Clashes

To analyse if the atoms from different chains in the same prediction overlap, we calculate the distance between all atoms in all chains in a given prediction. We count clashes as two atom positions from different chains being within 1Å from each other (the size of one hydrogen atom).

## MMalign

The DockQ[36] program is too slow to be run on large complexes if all interfaces are to be compared (minutes-hours for a single complex). Therefore, the program MMalign[38] is used to score entire complexes, as compared to the scoring of dimeric complexes with FoldDock previously[12]. MMalign performs optimal structural alignment between the model and native structures, computing a score (TM-score) normalised to be between zero and one, where one indicates a perfect match.

Since MMalign performs optimal structural superposition, it is also possible to evaluate models of different size. This is important since the predictions are based on full length protein sequences (and to score incomplete assemblies), while the PDB structures generally do not contain all residues from these, meaning that loops and other disordered regions are not present in the PDB structures. This also means that for most proteins, the score can never be 1, depending on how similar the SEQRES sequence is to the sequence present in the PDB structure. Since we assess the real sequences here, our approach represents a more realistic modelling scenario.

## Number of effective sequences

The number of effective sequences (Neff) is a measure of the information present in a multiple sequence alignment. To calculate the Neff, we clustered sequences from each MSA independently (both the paired and block diagonalized versions) at 62% sequence identity, following the rationale behind the BLOSUM62 matrix[39]. The clustering was performed using MMseqs2 version fcf52600801a73e95fd74068e1bb1afb437d719d [26]rs was used to indicate the Neff. MMseqs2 was run with the following command:

```
MMseqs2 easy-cluster msa outname /tmp  --min-seq-id 0.62 -c 0.8
--cov-mode 1
```

The clustering was done for all predicted subcomponents in each complex. To obtain a Neff score for each complex, we averaged the scores for all subcomponents.

## ROC curve

We create receiver operating characteristic (ROC) curves using the metrics average interface plDDT (predicted lDDT from AF), the number interface residues, contacts and interactions between chains normalised with the number of chains in each complex and the mpDockQ (multiple-interface predicted DockQ; average interface plDDT times the logarithm of the number of interface contacts). The positive examples are taken either as complete assemblies (when all native chains are present in an assembly) or being above the median TM-score (only for mpDockQ). The metrics are used to distinguish between true and false positives (TP and FP, respectively) by creating thresholds of all possible metric values. From the thresholding we calculate the true- and false positive rates:

$$TPR = \frac{TP}{TP+FN} \qquad \text{(viii)}$$

$$FPR = \frac{FP}{FP+TN} \qquad \text{(ix)}$$

Using the thresholds and corresponding TPR and FPR, the TPR is plotted against the FPR. This creates a ROC curve. For each metric the area under the ROC curve (AUC) is computed as:

$$AUC = \int_{x=0}^{1} TPR(\frac{1}{FPR(x)})dx \qquad \text{(x)}$$

## Non-redundant complexes from the PDB with 3-6 chains

The dataset of 3-6-mers was taken from the DOCKGROUND resource (dockground.compbio.ku.edu)[40] to analyse the relationship between accuracy and number of chains in complex using AlphaFold-multimer. This dataset consists of non-redundant PDB structures (available before October 2020) which have 3-6 protein chains (all longer than 30 amino acids) in both biological PDB units and do not have RNA or DNA. The extracted structures were further divided by stoichiometry. Redundancy was considered on the level of structural similarity of the quaternary structure quantified by the TM-score produced by MM-align[38]. Redundancy was removed within each stoichiometric group, separately, using a threshold of TM-score 0.6. In total, there are 1105 trimers, 678 tetramers, 210 pentamers and 531 hexamers. Supplementary figure 2 displays the distribution of the different types of oligomeric complexes.

## Human structures in the PDB

To analyse the number of available human PDB files (reported in the introduction), we downloaded all human entries from the PDB on 14th of October 2021 and counted the number of chains occurring in each entry. In total, there are 2649 human PDB files, 1557 with one chain, 720 with two and 372 entries with over two chains.

## Hu.MAP

To analyse the gap in complex structural knowledge for human proteins (introduction), all complexes with at least three chains from hu.MAP 2.0[4] were selected. hu.MAP is the result of a machine learning framework that identifies protein complexes using data from over 15000 mass spectrometry experiments. In total, there are 6956 complexes and 30572 protein chains, from 9962 unique genes. There are 4779 complexes with at least 3 chains, of which only 83 have all chains together in the same PDB entry.

# References

1. Will CL, Lührmann R. Spliceosome structure and function. Cold Spring Harb Perspect Biol. 2011;3. doi:10.1101/cshperspect.a003707

2. Tanaka K. The proteasome: overview of structure and functions. Proc Jpn Acad Ser B Phys Biol Sci. 2009;85: 12–36.

3. Ditzel L, Löwe J, Stock D, Stetter KO, Huber H, Huber R, et al. Crystal structure of the thermosome, the archaeal chaperonin and homolog of CCT. Cell. 1998;93: 125–138.

4. Drew K, Wallingford JB, Marcotte EM. hu.MAP 2.0: integration of over 15,000 proteomic experiments builds a global compendium of human multiprotein assemblies. Mol Syst Biol. 2021;17: e10016.

5. Giurgiu M, Reinhard J, Brauner B, Dunger-Kaltenbach I, Fobo G, Frishman G, et al. CORUM: the comprehensive resource of mammalian protein complexes-2019. Nucleic Acids Res. 2019;47. doi:10.1093/nar/gky973

6. Computational modeling of protein assemblies. Curr Opin Struct Biol. 2017;44: 179–189.

7. Fiser A. Template-Based Protein Structure Modeling. Methods Mol Biol. 2010;673: 73.

8. Sheng-You Huang XZ. MDockPP: A hierarchical approach for protein-protein docking and its application to CAPRI rounds 15–19. Proteins. 2010;78: 3096.

9. Russel D, Lasker K, Webb B, Velázquez-Muriel J, Tjioe E, Schneidman-Duhovny D, et al. Putting the pieces together: integrative modeling platform software for structure determination of macromolecular assemblies. PLoS Biol. 2012;10: e1001244.

10. Integrative Modelling of Biomolecular Complexes. J Mol Biol. 2020;432: 2861–2881.

11. Jumper J, Evans R, Pritzel A, Green T, Figurnov M, Ronneberger O, et al. Highly accurate protein structure prediction with AlphaFold. Nature. 2021;596: 583–589.

12. Bryant P, Pozzati G, Elofsson A. Improved prediction of protein-protein interactions using AlphaFold2. bioRxiv. 2021. p. 2021.09.15.460468. doi:10.1101/2021.09.15.460468

13. Christoffer C, Chen S, Bharadwaj V, Aderinwale T, Kumar V, Hormati M, et al. LZerD webserver for pairwise and multiple protein–protein docking. Nucleic Acids Res. 2021;49: W359–W365.

14. de Vries SJ, de Beauchêne IC, Schindler CEM, Zacharias M. Cryo-EM Data Are Superior to Contact and Interface Information in Integrative Modeling. Biophys J. 2016;110: 785.

15. Kawabata T. Multiple subunit fitting into a low-resolution density map of a macromolecular complex using a gaussian mixture model. Biophys J. 2008;95: 4643–4658.

16. Rantos V, Karius K, Kosinski J. Integrative structural modeling of macromolecular complexes using Assembline. Nat Protoc. 2021; 1–25.

17. Schuller AP, Wojtynek M, Mankus D, Tatli M, Kronenberg-Tenga R, Regmi SG, et al.

The cellular environment shapes the nuclear pore complex architecture. Nature. 2021;598: 667–671.

18. Huang X, Luan B, Wu J, Shi Y. An atomic structure of the human 26S proteasome. Nat Struct Mol Biol. 2016;23: 778–785.

19. Evans R, O'Neill M, Pritzel A, Antropova N, Senior A, Green T, et al. Protein complex prediction with AlphaFold-Multimer. bioRxiv. 2021. p. 2021.10.04.463034. doi:10.1101/2021.10.04.463034

20. Burke DF, Bryant P, Barrio-Hernandez I, Memon D, Pozzati G, Shenoy A, et al. Towards a structurally resolved human protein interaction network. bioRxiv. 2021. p. 2021.11.08.467664. doi:10.1101/2021.11.08.467664

21. Marsh JA, Hernández H, Hall Z, Ahnert SE, Perica T, Robinson CV, et al. Protein complexes are under evolutionary selection to assemble via ordered pathways. Cell. 2013;153: 461–470.

22. Luck K, Kim D-K, Lambourne L, Spirohn K, Begg BE, Bian W, et al. A reference map of the human binary protein interactome. Nature. 2020;580: 402–408.

23. Abramson B. The Expected-Outcome Model of Two-Player Games. PhD, COLUMBIA UNIVERSITY. 1987. Available: https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&ved=2ahUKEwidoJqS-bP2AhX0SfEDHX8oDcYQFnoECAYQAQ&url=https%3A%2F%2Facademiccommons.columbia.edu%2Fdoi%2F10.7916%2FD8TF05DD%2Fdownload&usg=AOvVaw1bn1Qo0xfmo_jmeTmvg1Oz

24. Silver D, Huang A, Maddison CJ, Guez A, Sifre L, van den Driessche G, et al. Mastering the game of Go with deep neural networks and tree search. Nature. 2016;529: 484–489.

25. Silver D, Hubert T, Schrittwieser J, Antonoglou I, Lai M, Guez A, et al. A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play. Science. 2018;362: 1140–1144.

26. Steinegger M, Söding J. MMseqs2 enables sensitive protein sequence searching for the analysis of massive data sets. Nat Biotechnol. 2017;35: 1026–1028.

27. Berman HM, Westbrook J, Feng Z, Gilliland G, Bhat TN, Weissig H, et al. The Protein Data Bank. Nucleic Acids Res. 2000;28: 235–242.

28. Eddy SR. Accelerated Profile HMM Searches. PLoS Computational Biology. 2011;7: e1002195.

29. Suzek BE, Huang H, McGarvey P, Mazumder R, Wu CH. UniRef: comprehensive and non-redundant UniProt reference clusters. Bioinformatics. 2007;23: 1282–1288.

30. UniProt Consortium. UniProt: the universal protein knowledgebase in 2021. Nucleic Acids Res. 2021;49: D480–D489.

31. Mitchell AL, Almeida A, Beracochea M, Boland M, Burgin J, Cochrane G, et al. MGnify: the microbiome analysis resource in 2020. Nucleic Acids Res. 2020;48: D570–D578.

32. Steinegger M, Mirdita M, Söding J. Protein-level assembly increases protein sequence recovery from metagenomic samples manyfold. Nature Methods. 2019;16: 603–606.

33. Mirdita M, von den Driesch L, Galiez C, Martin MJ, Söding J, Steinegger M. Uniclust

databases of clustered and deeply annotated protein sequences and alignments. Nucleic Acids Res. 2017;45: D170–D176.

34. Steinegger M, Meier M, Mirdita M, Vöhringer H, Haunsberger SJ, Söding J. HH-suite3 for fast remote homology detection and deep protein annotation. BMC Bioinformatics. 2019;20: 473.

35. Cock PJA, Antao T, Chang JT, Chapman BA, Cox CJ, Dalke A, et al. Biopython: freely available Python tools for computational molecular biology and bioinformatics. Bioinformatics. 2009;25: 1422–1423.

36. Basu S, Wallner B. DockQ: A Quality Measure for Protein-Protein Docking Models. PLoS One. 2016;11: e0161879.

37. Virtanen P, Gommers R, Oliphant TE, Haberland M, Reddy T, Cournapeau D, et al. SciPy 1.0: fundamental algorithms for scientific computing in Python. Nat Methods. 2020;17: 261–272.

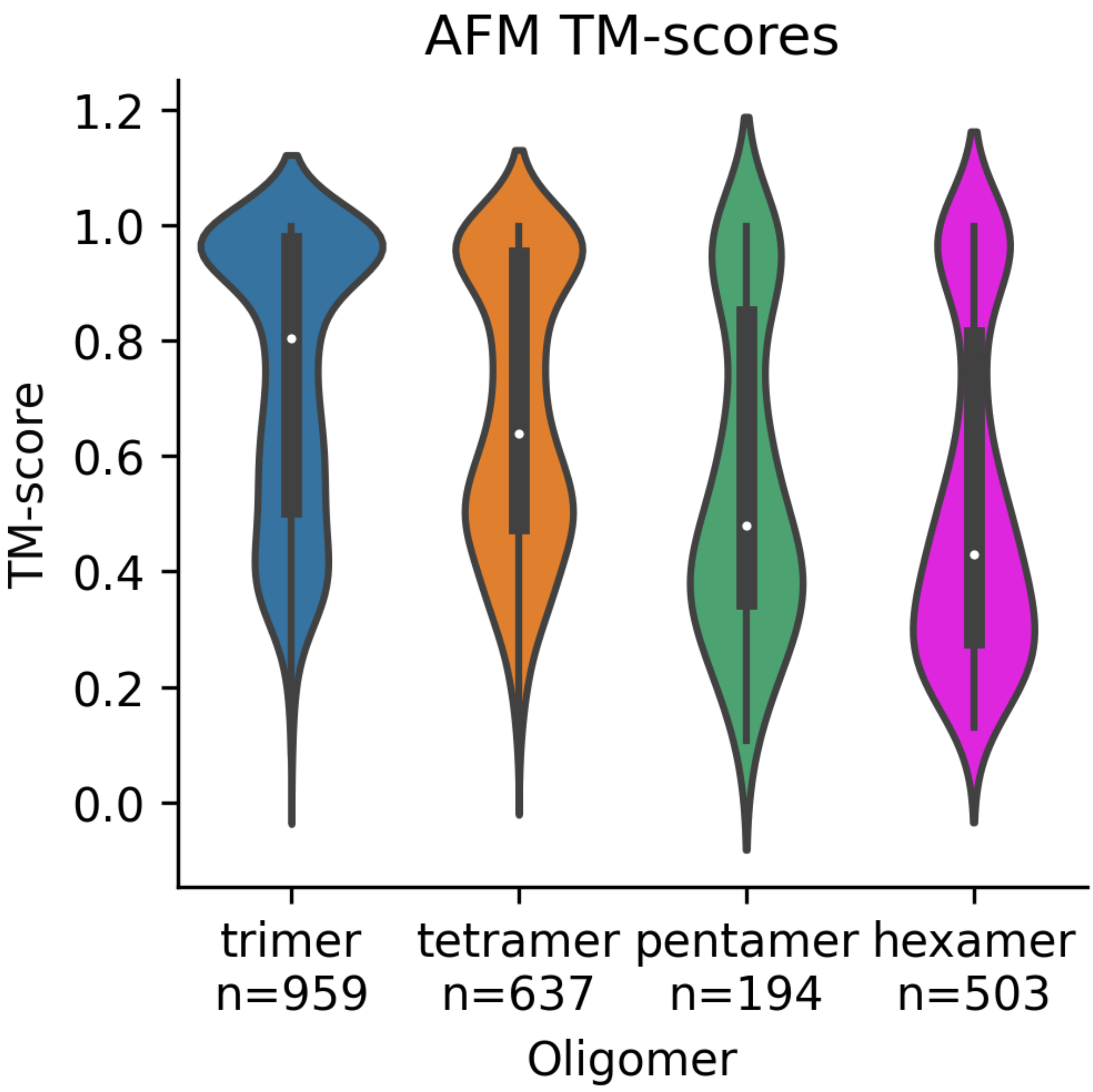
38. Mukherjee S, Zhang Y. MM-align: a quick algorithm for aligning multiple-chain protein complex structures using iterative dynamic programming. Nucleic Acids Res. 2009;37: e83–e83.

39. S Henikoff JGH. Amino acid substitution matrices from protein blocks. Proc Natl Acad Sci U S A. 1992;89: 10915.

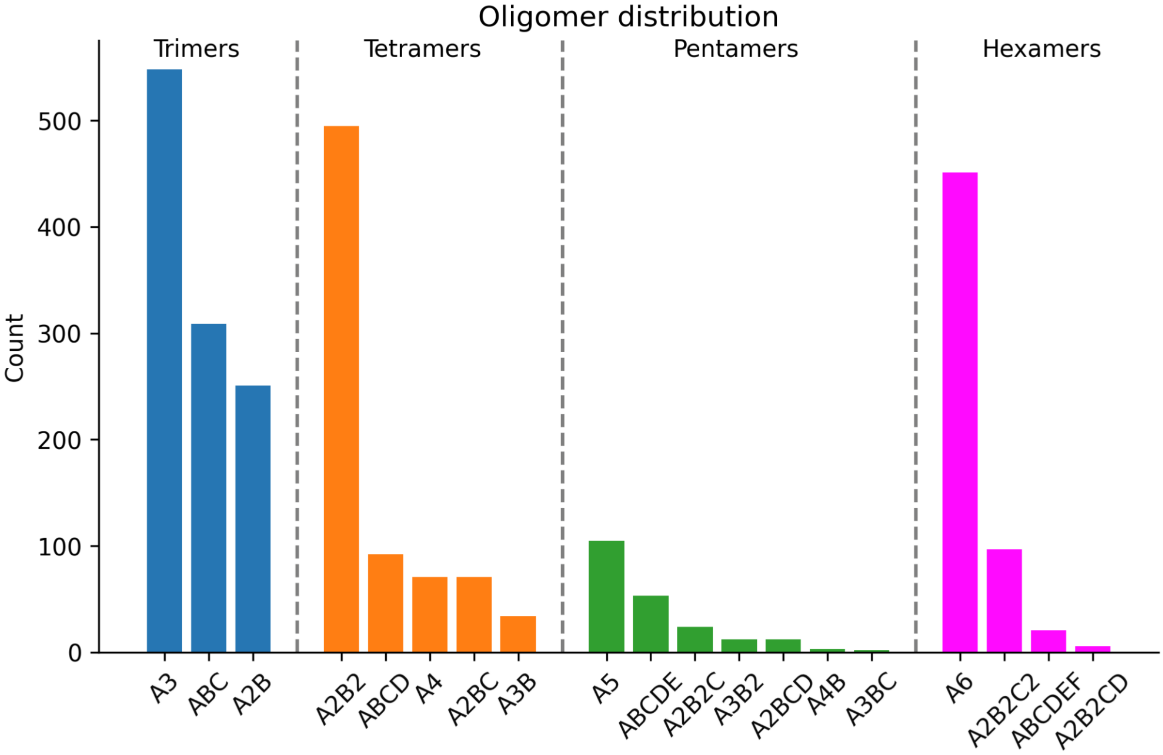
40. Kundrotas PJ, Kotthoff I, Choi SW, Copeland MM, Vakser IA. Dockground Tool for Development and Benchmarking of Protein Docking Procedures. Methods Mol Biol. 2020;2165: 289–300.

# Supplementary material

## Supplementary figures



**Supplementary Figure 1.** TM-score distributions for sets of non-redundant trimers, tetramers, pentamers and hexamers successfully modelled with AlphaFold-multimer . The scores decrease rapidly for oligomers with over three chains. The black boxes indicate the first and third quartiles of the data and the white dots the medians.



**Supplementary Figure 2.** Distribution of oligomers and their different cases for the complexes with 3-6 chains. A3 means three of the same chain, while ABC means three different chains and A2B two of the same and one different. All other namings follow the same convention. In total there are 1105 trimers, 678 tetramers, 210 pentamers and 531 hexamers.