Supplementary Materials

Text 1. Oligomeric state inference of different methods

Text 1.1 Parameter determination of POST

To determine the optimal weights in Equation (2), we empirically defined a range of values for each parameter, as shown in Table S7. Due to the large search space (~18,000 possible combinations), 15 grid points were randomly selected for optimization. The optimal weight combination was chosen as the grid point that resulted in the lowest average cross-entropy loss on the training set, as depicted in Figure S3A. In the final predictions, a probability threshold of 0.2 is applied to maximize the F1-score on the training set.

Text 1.2 Oligomeric state inference of individual methods

For each individual method, the putative templates are used to infer the oligomeric state. We first calculate the score for assigning state *k* according to the following formula

$$s_k = a \cdot s_{\mathrm{XX}}^k + b \cdot \frac{n_{\mathrm{XX}}^k}{N} \tag{S1}$$

where s_{XX}^k, n_{XX}^k are the maximum score and the total number of templates with oligomeric state k from POST-XX, respectively. N is the total number of used templates. In the training set, we determined that a and b are 7.5 and 3.5 for POST-DP, 7.5 and 3.5 for POST-PL, 8.5 and 5.5 for POST-HH, respectively (Figure S3B-D). Next, the probability of each state is calculated by exponentiating the scores and normalizing them by the sum of all exponents. Finally, the states with probability higher than a threshold (e.g., 0.15 for POST-DP, 0.2 for POST-HH, and 0.15 for POST-PL) are considered the predicted states.

Text 1.3 Oligomeric state inference of combining methods

To investigate the performance of different combinations of the methods POST-DP, POST-PL, and POST-HH, we design the following scoring function for any pair of methods.

$$s_k = a_1 \cdot s_{\mathrm{XX}}^k + a_2 \cdot s_{\mathrm{YY}}^k + b_1 \cdot \frac{n_{\mathrm{XX}}^k}{N} + b_2 \cdot \frac{n_{\mathrm{YY}}^k}{N} \tag{S2}$$

where $s_{\mathrm{XX}}{}^k$ ($s_{\mathrm{YY}}{}^k$), $n_{\mathrm{XX}}{}^k$ ($n_{\mathrm{YY}}{}^k$) are the maximum score and the total number of templates with oligomeric state k from POST-XX (POST-YY), respectively. N is the total number of used templates. The optimization of the weights a_1, a_2, b_1, b_2 on the training set follows the same procedure as in POST. For the combination of POST-DP and POST-HH methods, we determined the weights to be 4.5, 3, 2, and 4, respectively (as shown in Table S8). Similarly, for the combination of POST-DP and POST-PL methods, the weights were set to 3, 6, 1, and 5 (Table S9), and for the POST-HH and POST-PL methods, the weights were optimized to 2, 9, 4, and 6 (Table S10). Additionally, the corresponding probability thresholds for these combinations were set to 0.2, 0.2, and 0.15, respectively. These thresholds were established based on the training data to maximize the F1-score for each combination.

Text 1.4 Oligomeric state inference based on sequence similarity

We employ a nearest-neighbor approach based on sequence similarity to templates from our template library, which is denoted as **SeqTrans**. Specifically, we first calculate sequence identities between a given protein sequence and all templates in the library using MMseqs2 [1] with default parameters. The most similar template (i.e., the nearest neighbor) is then used to infer the oligomeric state of the query protein.

To investigate the impact of sequence identity between the query protein and the templates on prediction performance, we conducted evaluations under two conditions: (1) SeqTrans_Full: All templates in the template library are used. (2) SeqTrans_30: Templates with ⩾30% sequence identity to the query protein are excluded.

Text 1.5 AlphaFold2-based oligomeric state inference

For each protein sequence, AlphaFold v2.3.2[2] was run to model the protein in four different oligomeric states (i.e., monomer, homo-dimer, homo-trimer, and homo-tetramer). We used HHblits[3] to precompute the multiple sequence alignments (MSAs) for the target sequences, which were then trivially replicated repeatedly for homologous complexes. No templates were used, and no relaxation was performed. For monomer structures, all five "monomer" model parameters were applied, while complex structures were generated using all five "multimer" model parameters. This approach yielded 20 predictions for each test protein. During oligomeric state inference, if the maximum ipTM+pTM score for the dimer, trimer, and tetramer exceeds a threshold *C*, the state with the highest ipTM+pTM is predicted. If the maximum score is between 0.5 and *C*, and the pLDDT of the monomer is greater than 75, the prediction will be the monomer. If the pLDDT is less than or equal to 75, the state with the highest ipTM+pTM among the dimer, trimer, and tetramer will be selected. If all ipTM+pTM scores for the dimer, trimer, and tetramer are below 0.5, the monomer state is predicted. The results for different values of *C* are presented in Figure S4. In the final prediction, we chose *C* to be 0.8. Since the TS1220 dataset contains only monomers and dimers, we used AlphaFold2 to predict only the monomeric and dimeric structures for each protein to save time. For comparison, POST's outputs were also adjusted to include only monomers and dimers.

Text 2 Evaluation metrics

In this work, two types of evaluation metrics [4] are used to assess the performance of different methods. In this section, we list the detailed definition of these metrics.

Text 2.1 Sample-based metrics

Sample-based metrics include F1-score (F1), Accuracy (Acc), Precision (Pre) and Recall (Rec) as follows:

$$\begin{cases} F1 = \frac{1}{n}\sum_{i=1}^{n} \frac{2I_i}{T_i + P_i} \\ Acc = \frac{1}{n}\sum_{i=1}^{n} \frac{I_i}{U_i} \\ Pre = \frac{1}{n}\sum_{i=1}^{n} \frac{I_i}{P_i} \\ Rec = \frac{1}{n}\sum_{i=1}^{n} \frac{I_i}{T_i} \end{cases} \tag{S3}$$

Where n is the number of samples, T_i (P_i) is the number of real (predicted) labels for the i-th sample, I_i is the number of correctly predicted labels for the i-th sample, U_i is the total number of predicted and real labels (duplication removed) for the i-th sample.

Text 2.2 Label-based metrics

For the j-th label l_j, TP_j (FP_j) is the number of times label l_j is correctly (incorrectly) predicted, TN_j is the number of times label l_j is correctly not predicted, FN_j is the number of times label l_j is not predicted when it should have been. Then, the Matthews correlation coefficient (MCC) for label l_j can be defined based on the above four metrics:

$$MCC_j = \frac{TP_j \times TN_j - FP_j \times FN_j}{\sqrt{(TP_j + FP_j)(TP_j + FN_j)(TN_j + FP_j)(TN_j + FN_j)}} \tag{S4}$$

We also use precision, recall and F1-score to evaluate the performance of the prediction for each label l_j.

$$pre_j = \frac{TP_j}{TP_j + FP_j};\ rec_j = \frac{TP_j}{TP_j + FN_j};\ F1_j = \frac{2 \times pre_j \times rec_j}{pre_j + rec_j} \tag{S5}$$

Supplementary Figures and Tables

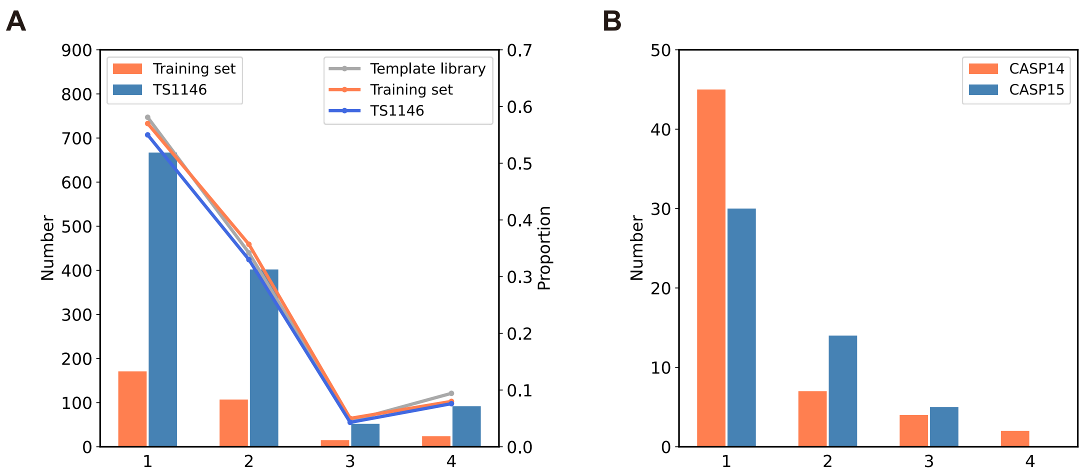

Figure S1. Distributions of the oligomeric states. (A) The training set and the TS1146 dataset collected by us. (B) The targets from CASP14 and CASP15. The bar and line charts represent the number and proportion of the four oligomeric states across different datasets, respectively. The x-axis represents the four oligomerization states (1: Monomer, 2: Dimer, 3: Trimer and 4: Tetramer).

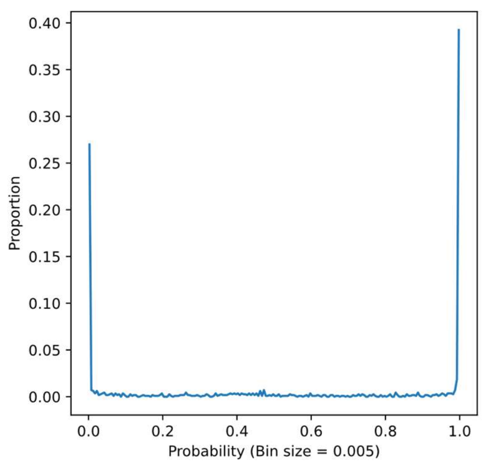

Figure S2. The distribution of probabilities predicted by POST on the dataset of 1134 proteins. The probability values are calculated based on the softmax function. The x-axis is divided into bins of width 0.005, ranging from 0 to 1. The y-axis represents the proportion of probabilities falling into each bin.

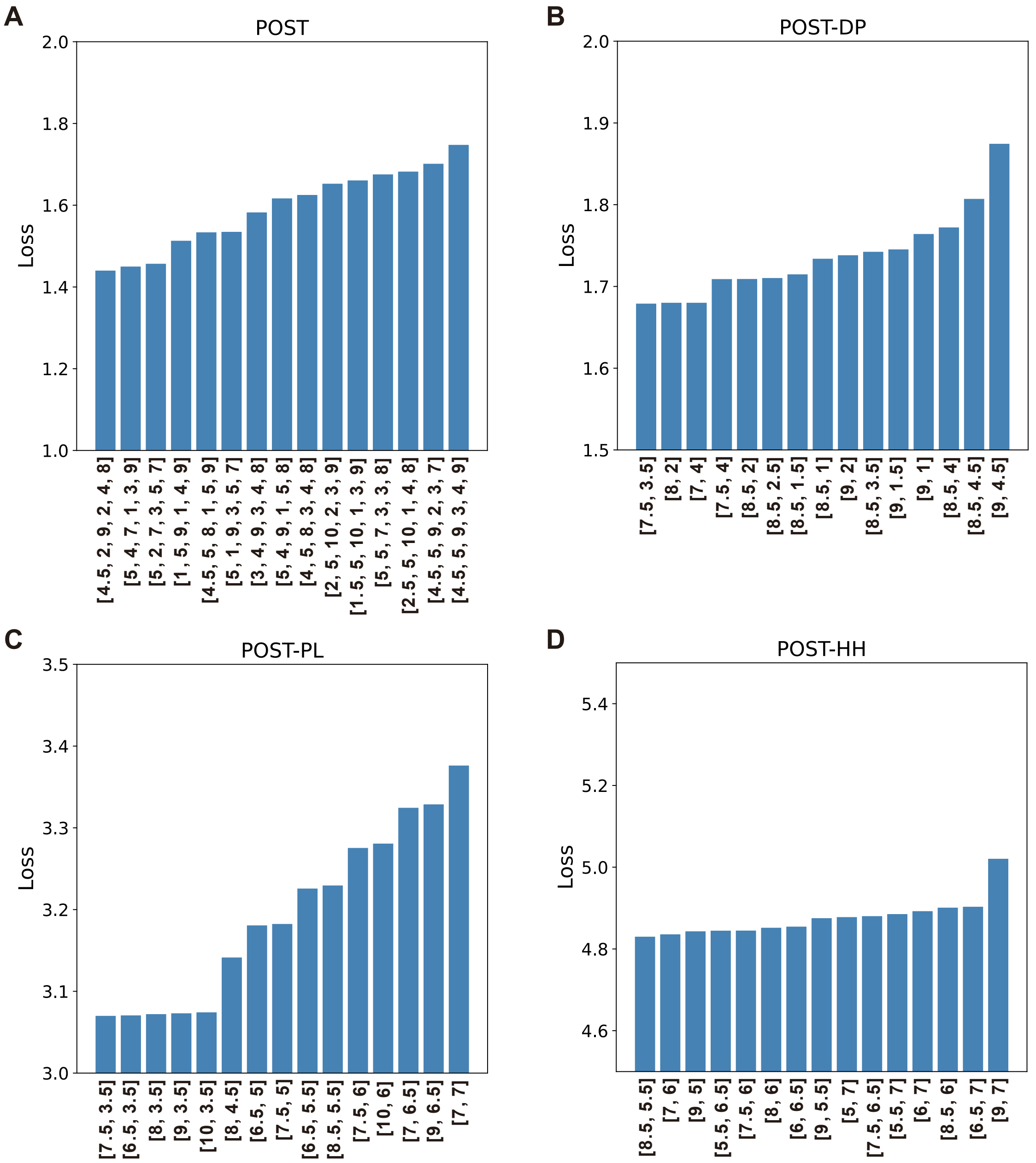

Figure S3. The cross-entropy losses for the 15 parameter combinations of the scoring function in POST, POST-DP, POST-PL, and POST-HH on the training set. The 15 parameter combinations in POST are randomly selected from all possible combinations within a defined range of values for six parameters. Similarly, POST-DP, POST-PL and POST-HH adopt the same approach as POST.

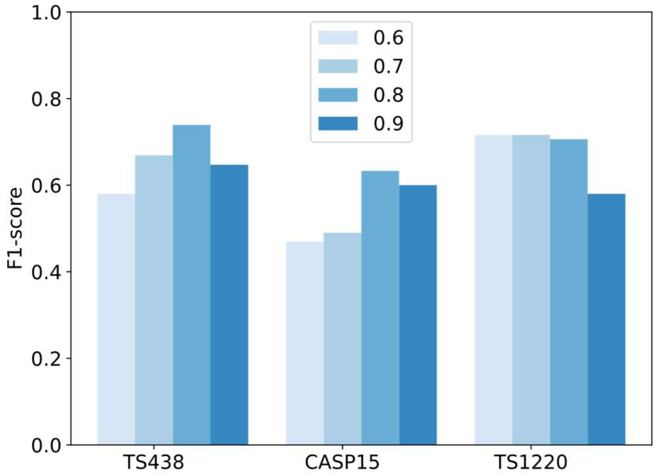

Figure S4. The F1-scores for different values of threshold *C* during AF2 oligomeric state inference on the TS438, CASP15 targets, and TS1220 datasets. In the final prediction, we chose 0.8 as the threshold. Since the different thresholds for monomer's pLDDT have little impact on the F1-score, we fixed it at 75.

Table S1. The *P*-values for the statistical tests (Mann-Whitney U-test) of the difference between POST and its component methods on the TS1146 dataset.

	POST-HH	POST-PL	POST-DP
POST-PL	2.56e-34		
POST-DP	2.56e-34	6.78e-12	
POST	2.56e-34	3.01e-31	2.78e-19

Table S2. F1-score values of POST and its component methods for individual oligomeric states on the TS1146 dataset.

Methods	Monomer	Dimer	Trimer	Tetramer
POST-HH	0.740	0.560	0.561	0.406
POST-PL	0.799	0.584	0.456	0.369
POST-DP	0.808	0.568	0.558	0.419
POST	0.815	0.613	0.602	0.410

Table S3. F1-score values of POST and other methods for individual oligomeric states on the CASP14 and CASP15 targets.

	Methods	Monomer	Dimer	Trimer	Tetramer
CASP14	**POST**	0.854	0.294	-	-
	POST-HH	0.694	0.143	-	-
	POST-PL	0.778	0.343	0.571	-
	POST-DP	0.848	0.245	-	-
	QUEEN	0.737	0.333	0.750	-
	DeepSub	0.485	0.114	0.889	-
CASP15	**POST**	0.722	0.385	0.286	-
	POST-HH	0.500	0.455	0.333	-
	POST-PL	0.696	0.489	0.600	-
	POST-DP	0.795	0.324	-	-
	QUEEN	0.615	0.250	0.500	-
	DeepSub	0.531	0.390	0.750	-

Table S4. Results of POST and AF2 for individual oligomeric states on the TS438, CASP15 targets, and TS1220 datasets.

Datasets	Methods	Label	Precision	Recall	F1-score
TS438	POST	1	0.730	0.918	0.813
		2	0.570	0.646	0.605
		3	0.909	0.526	0.667
		4	0.462	0.343	0.393
	AF2	1	0.783	0.860	0.820
		2	0.804	0.519	0.631
		3	0.542	0.684	0.605
		4	0.548	0.486	0.515
CASP15	POST	1	0.619	0.867	0.722
		2	0.417	0.357	0.385
		3	0.500	0.200	0.286
		4	-	-	-
	AF2	1	0.710	0.710	0.710
		2	0.700	0.538	0.610
		3	1	0.400	0.570
		4	-	-	-
TS1220	POST	1	0.743	0.689	0.715
		2	0.692	0.746	0.718
		3	-	-	-
		4	-	-	-
	AF2	1	0.704	0.750	0.726
		2	0.709	0.660	0.680
		3	-	-	-
		4	-	-	-

Table S5. Comparison of the run time between POST and other methods on 49 CASP15 targets.

Methods	Running time
POST-HH	3.28min
POST-PL	0.90min
POST-DP	4.46min
POST	8.64min
AF2	6.41h

† POST has two stages: template search and oligomeric state inference. During the template search stage, POST employs three individual methods to return templates sequentially. POST-DP runs on the CPUs with 60 processes, while POST-HH and POST-PL use default settings. POST's total runtime is approximately equal to the sum of the times taken by the three individual methods. For AlphaFold2, the total time is the sum of the modeling times for all four oligomeric states on an NVIDIA A100-PCIE-40GB GPU: 0.32 hours for monomers, 1.35 hours for dimers, 2.15 hours for trimers, and 2.59 hours for tetramers. Three targets were skipped due to modeling failures.

Table S6. Results of POST-DP and POST on the redundant and non-redundant libraries. The test is on the TS1146 dataset.

Methods	Library	F1-score	Accuracy	Precision	Recall
POST-DP	Non-redundant	**0.711**	0.666	0.684	0.784
	Redundant	0.709	0.671	0.690	0.767
POST	Non-redundant	**0.730**	0.693	0.711	0.790
	Redundant	0.727	0.683	0.701	0.800

Table S7. The empirically defined ranges for the six parameters of the scoring function in POST.

Parameter	Ranges
$\boldsymbol{a_1}$	[1, 1.5, 2, 2.5, 3, 3.5, 4, 4.5, 5]
$\boldsymbol{a_2}$	[1, 2, 3, 4, 5]
$\boldsymbol{a_3}$	[6, 7, 8, 9, 10]
$\boldsymbol{b_1}$	[1, 2, 3, 4, 5]
$\boldsymbol{b_2}$	[2, 3, 4, 5]
$\boldsymbol{b_3}$	[6, 7, 8, 9]

Table S8. Cross-entropy losses for the 15 weight combinations of the scoring function in the combination of POST-DP and POST-HH on the training set.

[a_1, a_2, b_1, b_2]	Loss	[a_1, a_2, b_1, b_2]	Loss
[4.5, 3, 2, 4]	1.217	[5, 4, 3, 2]	1.302
[4.5, 4, 2, 5]	1.238	[4.5, 5, 3, 2]	1.314
[4.5, 2.5, 3, 5]	1.246	[5.5, 4, 3, 1]	1.332
[4, 1.5, 4, 2]	1.254	[4, 2.5, 5, 3]	1.366
[5.5, 1.5, 1, 3]	1.257	[4.5, 2.5, 5, 2]	1.395
[4, 4.5, 3, 4]	1.268	[5.5, 4.5, 4, 4]	1.411
[5, 2.5, 3, 5]	1.274	[6, 2.5, 5, 2]	1.522
[6, 3, 1, 2]	1.289		

Table S9. Cross-entropy losses for the 15 weight combinations of the scoring function in the combination of POST-DP and POST-PL on the training set.

[a_1, a_2, b_1, b_2]	Loss	[a_1, a_2, b_1, b_2]	Loss
[3, 6, 1, 5]	1.140	[5.5, 4, 2, 3]	1.281
[1, 7, 3, 1]	1.151	[4, 6, 4, 4]	1.362
[3, 4, 3, 5]	1.161	[2, 9, 4, 3]	1.393
[2, 4, 4, 2]	1.177	[5, 6, 4, 6]	1.455
[3, 5.5, 3, 1]	1.189	[4, 9, 5, 3]	1.602
[1, 4, 5, 5]	1.215	[5.5, 9, 4, 4]	1.627
[4, 5.5, 2, 4]	1.226	[5.5, 7, 5, 2]	1.668
[5, 5.5, 1, 3]	1.249		

Table S10. Cross-entropy losses for the 15 weight combinations of the scoring function in the combination of POST-HH and POST-PL on the training set.

[a_1, a_2, b_1, b_2]	Loss	[a_1, a_2, b_1, b_2]	Loss
[2, 9, 4, 6]	2.507	[4.5, 5, 3, 9]	2.631
[4, 6, 5, 8]	2.661	[3, 6, 5, 8]	2.639
[4, 8, 4, 9]	2.694	[1, 9, 5, 9]	2.718
[6, 8, 5, 8]	2.887	[5, 4, 4, 7]	2.544
[5.5, 6, 3, 9]	2.672	[6, 9, 5, 6]	2.793
[5.5, 7, 3, 9]	2.694	[5, 8, 3, 8]	2.665
[4, 6, 2, 9]	2.569	[4, 4, 4, 7]	2.521
[5, 7, 5, 8]	2.776		

Supplementary References

1. Steinegger, M. and J. Söding, *MMseqs2 enables sensitive protein sequence searching for the analysis of massive data sets.* Nature Biotechnology, 2017. **35**(11): p. 1026-1028.
2. Evans, R., et al., *Protein complex prediction with AlphaFold-Multimer.* bioRxiv, 2022: p. 2021.10.04.463034.
3. Remmert, M., et al., *HHblits: lightning-fast iterative protein sequence searching by HMM-HMM alignment.* Nat Methods, 2011. **9**(2): p. 173-5.
4. Zhang, M.L. and Z.H. Zhou, *A Review on Multi-Label Learning Algorithms.* IEEE Transactions on Knowledge and Data Engineering, 2014. **26**(8): p. 1819-1837.