Supplementary Data

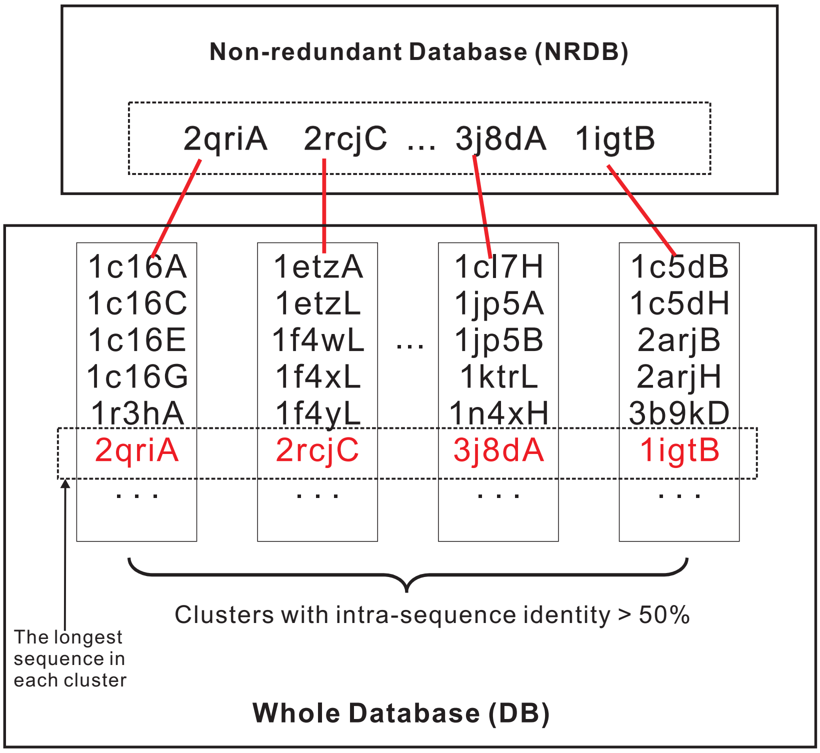

Figure S1. The relationship between the NRDB and DB.

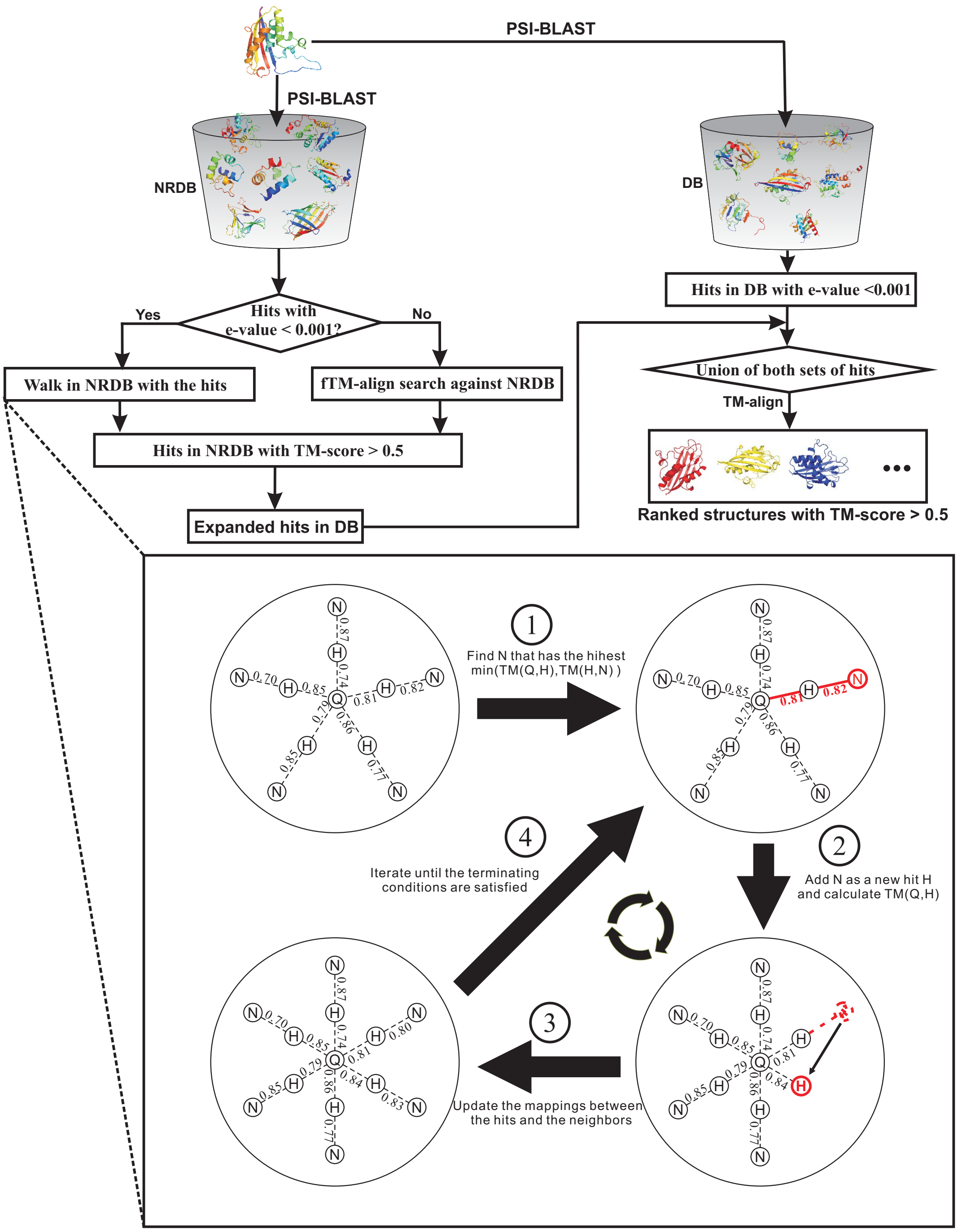
PSI-BLAST
PSI-BLAST
NRDB
DB
Hits with e-value < 0.001?
Yes
No
Walk in NRDB with the hits
fTM-align search against NRDB
Hits in NRDB with TM-score > 0.5
Expanded hits in DB
Hits in DB with e-value <0.001
Union of both sets of hits
TM-align
Ranked structures with TM-score > 0.5
1
Find N that has the hihest min(TM(Q,H),TM(H,N))
2
Add N as a new hit H and calculate TM(Q,H)
3
Update the mappings between the hits and the neighbors
4
Iterate until the terminating conditions are satisfied
Q
H
N
0.87
0.74
0.70
0.85
0.81
0.82
0.79
0.86
0.85
0.77
0.84
0.80
0.83

Figure S2. The overall architecture of the fast database search in the mTM-align server. The query sequence is compared with the sequences in NRDB by PSI-BLAST to find similar structures at e-value < 0.001. If hits are found, a procedure called 'walk' (shown at the bottom of the figure) is used to find more similar structures in NRDB. Otherwise, the query structure will be compared with all structures in NRDB by fTM-align, a fast version of TM-align, which works by reducing the number of iterations in TM-align. Then the identified structures from NRDB are expanded to other structures in DB which are in the same cluster of each identified structure. In addition, another search against the whole database (DB) is performed using PSI-BLAST at e-value of 0.001, which quickly detects all proteins with similar sequences to the query. These proteins are then combined with expanded search results from the NRDB.

After the combination, the query structure will be aligned to the structures in the combined set. Structures with TM-score > 0.5 will be included into the final list. Since the TM-score is normalized by the length of the query, all structures with short length (< half of the query length) are excluded at this stage to speed up the calculations.

The walk is an iterative method to find more similar structures in NRDB that are missed by PSI-BLAST. It starts from the hit set (denoted by *H*) from PSI-BLAST and will add one more structure into *H* in each iteration. First, the TM-score between the query and each hit in *H* is computed with TM-align. Then, the "closest neighbor" (not in *H*) of each hit (i.e., the one with the highest pre-calculated TM-score to the hit) is regarded as a new candidate to be added into *H*. The set of these neighbors are denoted by *N*. The purpose of the walk is to avoid performing the time-consuming structure alignment between the query and the structures in *N*. This is realized by estimating the TM-score $e(q, N_i)$ between the query and the *i*-th structure in *N* as the minimum of *TM-score*(q, H_i) and *TM-score*(N_i, H_i). After this, a new iteration is performed with new set *H*. The iteration stops when the following two conditions are satisfied: a specified number of hits (200) have been reported and the estimated TM-scores for all neighbors are less than 0.7.

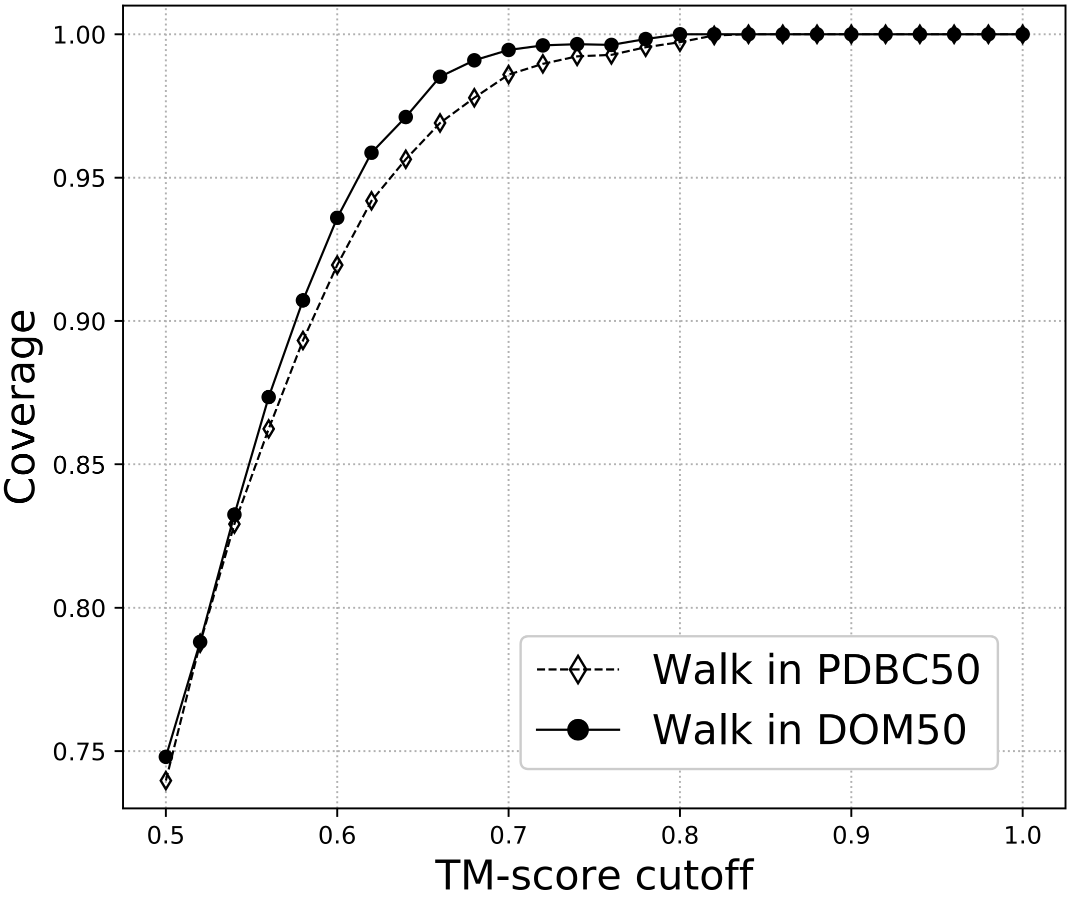

Figure S3. The performance of the walk algorithm, measured on the D500 dataset. The coverage is defined as

$$\mathrm{cov}(x) = \frac{\#\{t_i : \mathrm{TM}(t_i, q) \geq x;\ \text{and } 1 \leq i \leq N_{walk}\}}{\#\{t_i : \mathrm{TM}(t_i, q) \geq x;\ \text{and } 1 \leq i \leq N_{TM-align}\}}$$

where TM(t_i, q) is the TM-score between the template t_i and the query; x is a specified TM-score cutoff. N_{walk} is the total number of structures in NRDB detected based on walk; and $N_{TM\text{-}align}$ is the total number of the structures in NRDB detected by running TM-align directly.

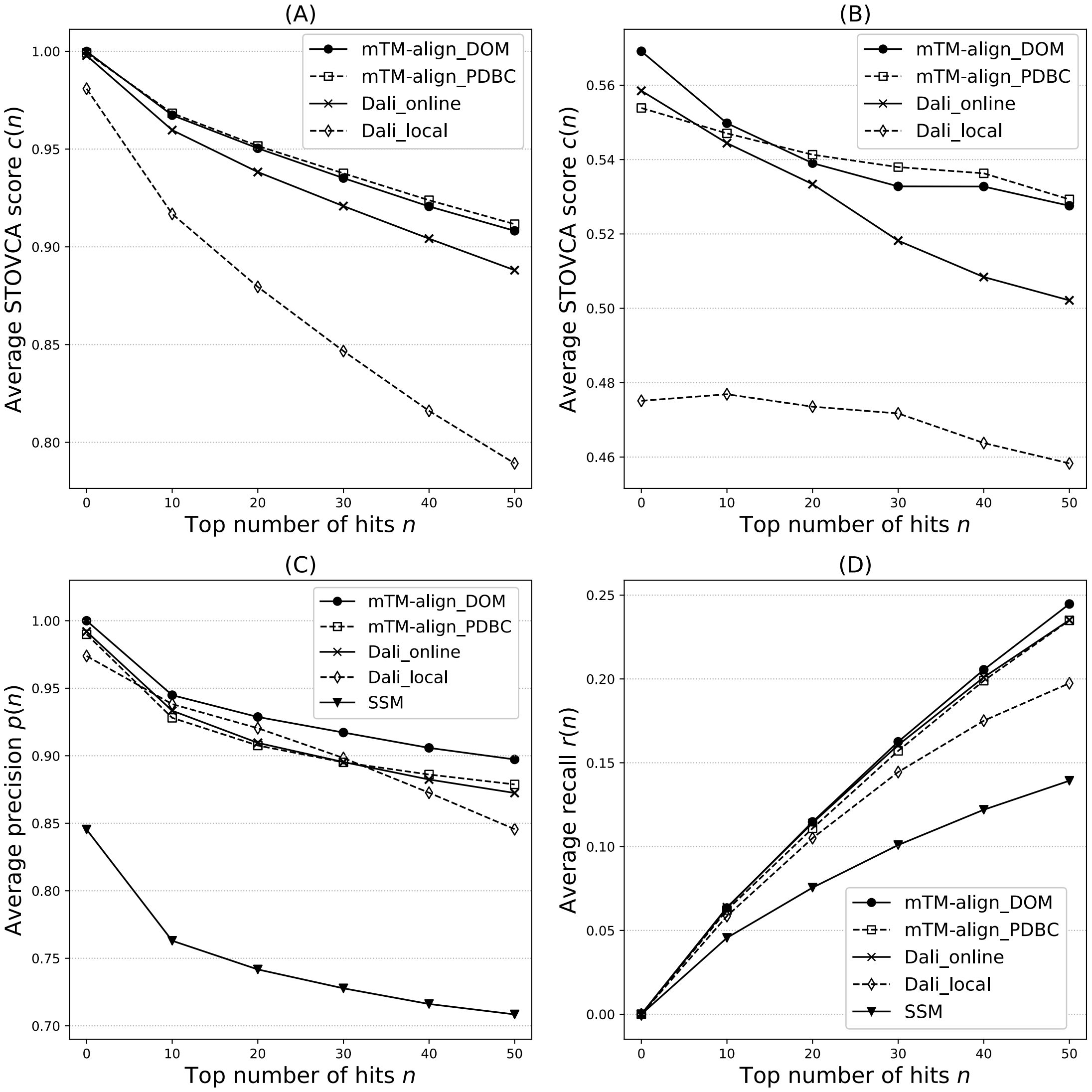

Figure S4. The zoomed version of Figure 2 for the range of $n \leq 50$.