nature protocols

https://doi.org/10.1038/s41596-025-01189-x

Protocol

Check for updates

A graphic and command line protocol for quick and accurate comparisons of protein and nucleic acid structures with US-align

Chengxin Zhang[1,2,3], Lydia Freddolino[2,3] ✉ & Yang Zhang[4,5,6] ✉

Abstract

With the success of structural biology and the advancements in deep-learning-based structure predictions, rapid and accurate structural comparisons among macromolecular structures have become increasingly important in structural bioinformatics. US-align is a highly efficient, versatile, open-source program for sequential and nonsequential structure comparisons of proteins, RNAs and DNAs in pairwise and multiple alignment forms and applicable to both monomeric and multimeric complex structures. The core algorithm of US-align is built on a highly optimized, iterative superimposition and dynamic programming alignment process, guided with a unified and sequence length-independent scoring function, TM-score. The unique design of US-align not only ensures its high accuracy and speed compared with other state-of-the-art methods designed for specific alignment tasks but also makes it the only protocol that can be applied to multiple alignment tasks and allow a structural comparison across different molecular types, the latter of which is critical for template-based heteromolecular structure prediction and function annotations. Here we describe how to install and effectively utilize US-align as a command line tool, as an online web server, and as a plugin to commonly used molecular graphic systems such as PyMOL. US-align installation takes a few minutes to setup, while the actual alignment implementation can be completed typically within 1 s.

Key points

- US-align is a highly efficient, versatile and open-source program for the sequential and nonsequential structure comparisons of proteins, RNAs and DNAs in pairwise and multiple alignment forms. It is applicable to both monomeric and multimeric complex structures.
- Its unique design ensures high accuracy and speed compared with other state-of-the-art methods designed for specific alignment tasks and enables it to be applied to multiple alignment tasks, allowing structural comparison across different molecular types.

Key references

Zhang, C. et al. *Nat. Methods* **19**, 1109–1115 (2022): https://doi.org/10.1038/s41592-022-01585-1

Zhang, Y. et al. *Nucleic Acids Res.* **33**, 2302–2309 (2005): https://doi.org/10.1093/nar/gki524

Zhang, Y. et al. *Proteins* **57**, 702–710 (2004): https://doi.org/10.1002/prot.20264

[1]CAS Key Laboratory of Quantitative Engineering Biology, Shenzhen Institute of Synthetic Biology, Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen, China. [2]Gilbert S Omenn Department of Computational Medicine and Bioinformatics, University of Michigan, Ann Arbor, MI, USA. [3]Department of Biological Chemistry, University of Michigan, Ann Arbor, MI, USA. [4]Department of Computer Science, School of Computing, National University of Singapore, Singapore, Singapore. [5]Cancer Science Institute of Singapore, National University of Singapore, Singapore, Singapore. [6]Department of Biochemistry, School of Medicine, National University of Singapore, Singapore, Singapore. ✉e-mail: lydsf@umich.edu; zhang@zhanggroup.org

Introduction

Development of the Protocol

Molecule-level structure comparisons, including superimposition and alignment, are fundamental in structural biology and bioinformatics. Here, structural superimposition (for example, via optimization of the template modeling score (TM-score)[1–3] and root mean square deviation (r.m.s.d.)[4]) (see Box 1 for further details) refers to the optimal overlay of two known structures with a priori given comparison order (that is, residue-level correspondence) of sequences, while structural alignment (for example, TM-align[5] and Dali[6]) requires the determination of an optimal comparison order before the structural superimposition. Traditionally, different structure alignment tools have been developed for specific molecule types (proteins[5–9], DNAs[3] or RNAs[3,10–13]) and alignment tasks, including pairwise monomer[5–9] and oligomer complex alignments[14,15] and multiple structure alignment (MSTA) involving three or more structures[16–18]. The need to use different alignment programs and structure similarity metrics for different alignment applications creates both inconvenience for users and, more importantly, ambiguities in structural comparisons across different molecule types, which make it difficult to compare heteromolecular structures and make functional inferences across different molecule types. To address these issues, we developed the US-align

BOX 1

Mathematical definition of TM-score

In US-align, the main metric to quantify the similarity between a pair of protein structure is the TM-score[1]:

$$\mathrm{TM} = \frac{1}{L}\sum_{i=1}^{L_{\mathrm{ali}}} \frac{1}{1+(d_i/d_0)^2},$$

where L is the sequence length of the target protein, L_{ali} is the number of aligned residues and d_i is the distance between the Cα atoms of the ith aligned residue after the optimal TM-score superimposition. The normalization factor d_0 is a distance scale defined in a way that the TM-score is independent of the protein length:

$$d_0 = \begin{cases} 1.24\sqrt[3]{L-15} - 1.8, & \text{if } L > 21 \\ 0.5, & \text{if } L \le 21 \end{cases}.$$

The definition of TM-score for nucleic acids is similar, except that d_i is calculated between C3′ atoms and that the d_0 is defined by:

$$d_0 = \begin{cases} 0.6\sqrt{L-0.5} - 2.5, & \text{if } L \ge 30 \\ 0.7, & \text{if } 24 \le L \le 29 \\ 0.6, & \text{if } 20 \le L \le 23 \\ 0.5, & \text{if } 16 \le L \le 19 \\ 0.4, & \text{if } 12 \le L \le 15 \\ 0.3, & \text{if } L \le 11 \end{cases}$$

Based on extensive analysis of protein[2] and RNA[3] structure families, it was found that a TM-score ≥0.5 (or ≥0.45) corresponds to a protein (or RNA) pair with similar global structural topology. In addition to TM-score, US-align also reports the r.m.s.d. of the aligned residues:

$$\mathrm{r.m.s.d.} = \sqrt{\frac{1}{L_{\mathrm{ali}}}\sum_{i=1}^{L_{\mathrm{ali}}} d_i^2}.$$

While r.m.s.d. ranges from 0 to infinite, the TM-score ranges between 0 and 1, with a TM-score of 1 indicating a perfect match.

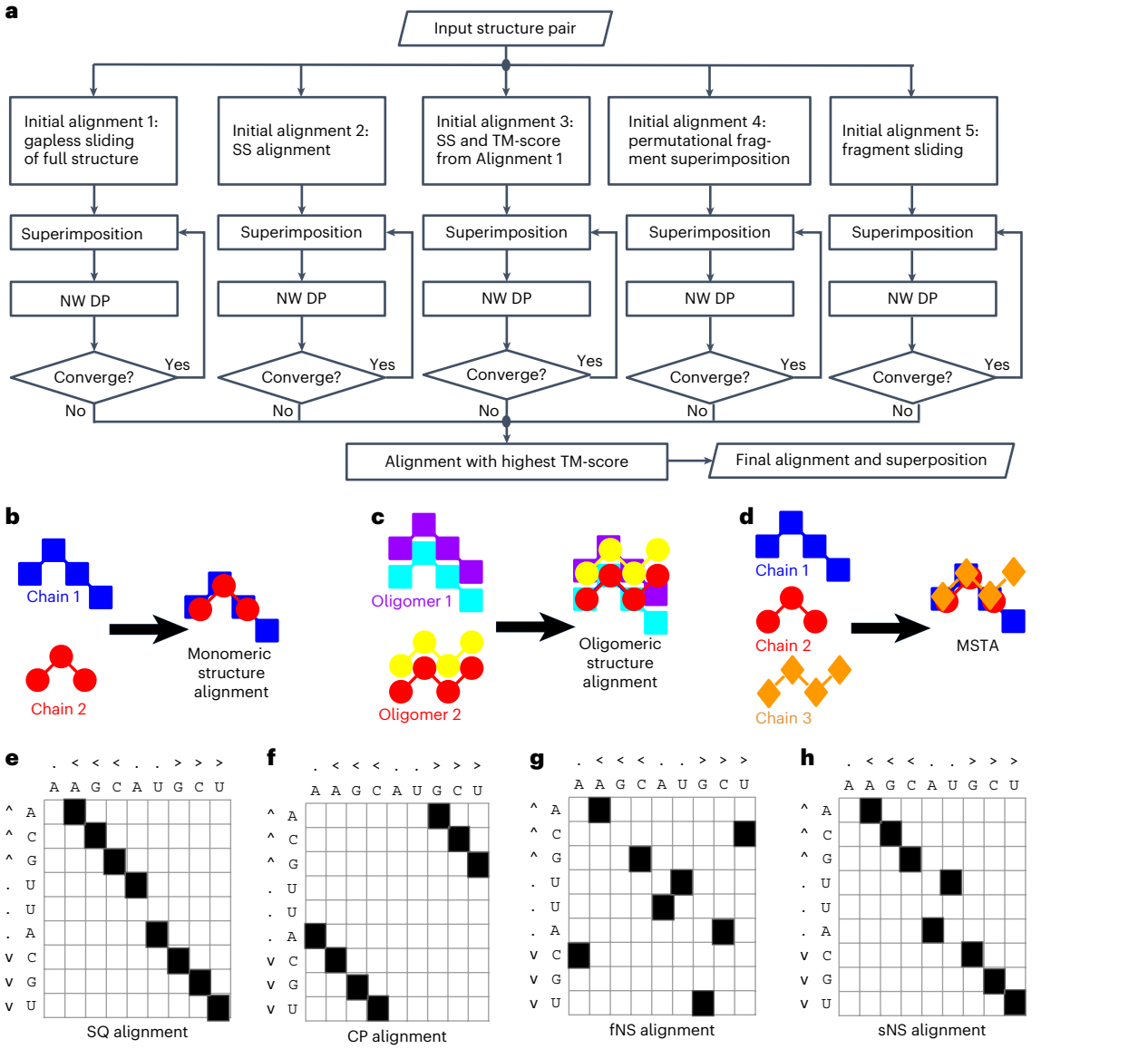

Fig. 1 | Diagrams of the US-align algorithm and its different alignment modes. a, The core iterative superimposition–alignment algorithm of US-align for different alignment tasks. 'Superimposition' refers to the rotation and translation of one structure over another structure to maximize the TM-score between the structure pair. 'Converge' refers to the end of superimposition–alignment iterations when the TM-score no longer changes. NW DP, Needleman–Wunsch global dynamic programming algorithm; SS, secondary structure. **b**–**d**, Different alignment tasks performed by US-align, including pairwise alignment of monomeric structures (**b**), pairwise alignment of oligomeric complexes (**c**) and MSTA of monomeric structures (**d**). **e**–**g**, Dot plots illustrating SQ (**e**), CP (**f**) and non-sequential (NS) (**g**) alignments of US-align, where each black square represents a pair of aligned residues. Here, the RNA SS (that is, base pairing pattern) of the input RNA pairs are shown in dot-bracket notation, where '.' indicates the unpaired nucleotides, while brackets '<' and '>' indicate the base pairs. In SQ alignment, if residue i and j in structure 1 aligns to residues m and n in structure 2, then $(i-j)(m-n) > 0$. In fNS, $(i-j)(m-n)$ can be either positive or negative. **h**, In sNS, $(i-j)(m-n)$ must be positive within a SS segment but can be either positive or negative outside a SS segment. In CP, the original termini of one structure were linked and new ones created elsewhere to align with another structure.

algorithm[19] (Fig. 1a) to extend and integrate all of the aforementioned structural alignment tasks for different biomolecule types (Fig. 1b–d) under a uniform scoring function (TM-score)[1,3] whose magnitude is statistically independent of sequence length. While US-align was originally developed for sequential structure alignment (Fig. 1e), we recently added nonsequential structure alignment functionalities to US-align[20] (Fig. 1f–h). Thus, US-align provides a universal framework for performing and scoring structural comparisons of biomacromolecules.

Applications of the method

US-align has been widely used in the community for macromolecular structure comparison, including the assessments of model qualities in various community-wide structure prediction challenges[21,22]. However, the application of US-align is not limited to mere structure comparisons, since accurate structural alignments can be employed for a variety of applications, including, for example, structure-based function annotation[23–25], protein–ligand interaction prediction[26–29], protein–protein and protein–RNA docking[12,30,31], protein domain structure assembly[32,33], evolution-based protein design[34], fragment-guided structure model refinement[35] and bioinformatics database curation[36–39].

Comparison of US-align with other methods

Although US-align is not the only method for any specific structure alignment tasks and not even the only program built on TM-score[9,12,15], it is often one of the fastest and most accurate tools available. In our recent benchmark for pairwise alignment of 637 single chain RNAs[19], for example, US-align achieves a 6–39% higher TM-score, while being at least ten times faster than the state-of-the-art RNA structure alignment programs such as RMalign[12], STAR3D[10], ARTS[11] and Rclick[13]. Furthermore, in a parallel benchmark for alignment of 31,951 pairs of protein domains[19], US-align achieves TM-scores that are on average 2–22% higher than those achieved by SPalign[9], Dali[6], MICAN[15] and SSM[8], while being at least 1.6 times faster than these competing programs.

While there are some programs[40,41] that run faster than US-align in database searching, the speed of these programs is mainly built on the prefiltering of a curated structure database that requires precalculated representations of the target structures in the library. Therefore, the programs are not designed or suitable for direct comparisons of unknown structures, like other mainstream structure alignment tools.

Expertise needed to implement the protocol

The US-align protocol contains three major functionalities: command line tool, online web server and a plugin to graphic systems, such as PyMOL. The usage of the web server and the PyMOL plugin for US-align does not require any command line usage and is intuitively straightforward for any structural biology users. However, the local installation and implementation of the US-align command line functions on the user's own computer requires familiarity with the basis of command line prompts in the appropriate operating system (OS) environments.

Limitations

Despite the versatility of US-align, it is not designed for flexible structure alignment, which is needed sometimes for matching and aligning homologous structure pairs where one of them involves large conformational changes, for example, in domain orientations. For such purposes, specifically designed flexible alignment tools, such as FATCAT[42] or Matt[18], can be used. Although US-align is available as a plugin for PyMOL, it currently does not include a plugin for other advanced molecular visualization and analysis software such as UCSF ChimeraX[43], a widely used tool in the cryo-EM and structural biology community. We plan to incorporate such plugins and other needed updates in future developments.

Experimental design

To illustrate pairwise monomer structure alignments, we mainly use three protein pairs. The first includes two myoglobins (Protein Data Bank (PDB) identifications (IDs): 101m and 1mba) that have a low sequence similarity (identity of 24.5%) and high structural similarity (TM-score of 0.85). The second pair includes two beta-glucanases (PDB ID: 1ajk and 2ayh), which are circular permutation (CP) proteins. The third pair contains a tRNA (PDB ID: 1evv) and a ribosome recycling factor protein (PDB ID: 1eh1), which represent a well-known case of molecular mimicry between protein and nucleic acid, with similar structure and function[44]. Similarly, for illustrating oligomer alignments, we chose a pair of hard-to-align octamers, including a mandelate racemase/muconate lactonizing enzyme (PDB ID: 4jhm) and an unknown

protein (PDB ID: 4iaj) with very low sequence similarity (identity of 5.3%) and modest structural similarity (TM-score of 0.54). For MSTA, we chose a set of tRNAs (PDB IDs: 1evv and 3am1) and a prohead RNA (pRNA, PDB ID: 6jxm) with varying degree of similarities (sequence identities ranging from 35.2% to 50.0%). These example files were chosen primarily for their challenging alignment characteristics and suitability for clear visualization of structural alignment models. However, users can apply US-align to any of the aforementioned structural alignment functions, provided the corresponding three-dimensional (3D) structures are available in PDB or mmCIF format.

Materials

Equipment

- A personal computer (in this study, all analysis is done on a Lenovo ThinkPad T14 Gen 2) with Internet connection that runs on one of the three major OSs (Microsoft Windows, Mac OS or Linux). In addition, Windows Subsystem for Linux is also supported. Only 64-bit OSs are fully supported

Software

To fully use all US-align functions (command line tool, webserver and PyMOL plugin), the following software is required:

- One of the major web browsers (for example, Google Chrome, Firefox, Safari and Microsoft Edge) that supports JavaScript is needed. JavaScript is used by the JSmol applet[45] in the output page of the US-align webserver to visualize the superimposed structures
- The g++ compiler (known to work with version 4.8.5 or later) or any other compiler compatible with C++ 98 or later, such as clang++ (known to work with version 12.0.5 or later) on Mac OS or mingw-w64 (known to work with version 9.3 or later) on Windows
- The PyMOL molecular graphics system[46] (version 1.7 or later). The PyMOL plugin for US-align supports PyMOL installed from the following sources:
  - Official installer from Schrödinger (https://pymol.org/2/)
  - PyMOL source code (https://github.com/schrodinger/pymol-open-source)
  - Package managers of Linux distributions (for example, apt-get and yum)
  - Unofficial Windows installer built by Christoph Gohlke from the Laboratory for Fluorescence Dynamics, University of California, Irvine (https://pymolwiki.org/index.php/Windows_Install#Open-Source_PyMOL)
  - PyMOL installed through conda (https://pymol.org/conda/)
    ▲ **CAUTION** US-align does not support the legacy PyMOL build distributed at SourceForge (https://sourceforge.net/projects/pymol/files/Legacy/), as it is too old (version 0.99).
- (Optional) Alternatively, the RasMol molecular graphics system[47], which is relatively lightweight and fast, especially when launched from a remote server connected through SSH, can be used if there is no need to use the PyMOL plugin for US-align. The newest version of US-align also supports UCSF ChimeraX[43]. Although other molecular graphic systems such as VMD[48] or Jmol[49] can also be used, they are not guaranteed to generate as consistent coloring schemes and artistic styles for US-align superimposed structures as PyMOL and RasMol

Data files

- The structure files used for illustrative purposes in this Protocol are listed in Table 1. While the table provides links to the PDB format structure files, US-align can also handle mmCIF format structure files with identical alignment results at comparable speed. On Unix-like OSs (for example, Linux and Mac OS), US-align can read both uncompressed structure files and gzip compressed files. In the latter case, gunzip and POSIX Process Control

Table 1 | Structure files used for illustrative purposes in this Protocol

PDB ID	Download link	Description
101m	https://files.rcsb.org/download/101m.pdb	Sperm whale myoglobin
1mba	https://files.rcsb.org/download/1mba.pdb	Slug sea hare myoglobin
4jhm	https://files.rcsb.org/download/4jhm.pdb1	*Pseudovibrio* mandelate racemase/muconate lactonizing enzyme
4iaj	https://files.rcsb.org/download/4iaj.pdb1	*Streptococcus pneumoniae* protein SP_1775
1eh1	https://files.rcsb.org/download/1eh1.pdb	*Thermus thermophilus* ribosome recycling factor
1evv	https://files.rcsb.org/download/1evv.pdb	Yeast tRNAs
6jxm	https://files.rcsb.org/download/6jxm.pdb	Phage prohead RNA
3am1	https://files.rcsb.org/download/3am1.pdb	*Methanocaldococcus jannaschii O*-phosphoseryl-tRNA kinase complexed with tRNA
1ajk	https://files.rcsb.org/download/1ajk.pdb	*Paenibacillus macerans* circularly permuted beta-glucanase
2ayh	https://files.rcsb.org/download/2ayh.pdb	*Bacillus* beta-glucanase
Non-redundant PDB	https://zhanggroup.org/library/PDB.tar.bz2	A nonredundant structure library consisting of protein domains and full protein chains from PDB with a pairwise sequence identity <70%. This weekly updated library is also used by I-TASSER as the template library (1.8 GB)
model.pdb, model2.pdb, native.pdb, align.txt, model_1l6hA.pdb, native_1l6hA.pdb, modelComplex.pdb, nativeComplex.pdb	https://zhanggroup.org/TM-score/help.zip	Example files to demonstrate the '-TMscore' option of US-align (628 KB)

in C++ (Wakely, J. (2020) https://pstreams.sourceforge.net/) are used for automated decompression. However, US-align can only read uncompressed files when running on Windows, which does not support POSIX

Procedure

Installing the US-align command line tool

● **TIMING 1 min**

▲ **CRITICAL** Steps 1–2 are not necessary if users only use the US-align web server and/or PyMOL plugin.

1. To attain the fastest execution speed for the US-align command line tool, we recommend compiling the US-align from the C++ source code as follows:

```
curl https://zhanggroup.org/US-align/bin/module/USalign.cpp -o USalign.
cppg++ -static -O3 -ffast-math -lm -o USalign USalign.cpp
```

Change g++ to clang++ or mingw-w64 based on the availability of C++ 98 compiler in the OS. Remove the '-static' flag when compiling the code on Mac OS, which does not support static executable. For ease of use, it is recommended to add this folder containing the executable file to the $PATH environment variable. For example, on Linux and Windows Subsystem for Linux, this can be achieved by:

```
echo "export PATH=$PWD:$PATH" >> ~/.bashrc
bash
```

▲ **CRITICAL STEP** Compilation of USalign.cpp by recent versions of clang++ may result in the following message: 'warning: 'sprintf' is deprecated:'. This is due to the usage of 'sprintf' rather than 'snprintf' for compatibility purposes. The user can safely ignore the warning message on sprintf deprecation.

◆ **TROUBLESHOOTING**

2. If the source code cannot be compiled (for example, due to lack of compiler for C++ 98), users can download the precompiled 64-bit binary executables of US-align for Windows, Mac OS (for both Intel and ARM architectures) and Linux at the bottom of https://zhanggroup.org/US-align.
   ◆ **TROUBLESHOOTING**

Installing the PyMOL plugin for US-align

● **TIMING 2 min**

▲ **CRITICAL** Steps 3–7 are not necessary if users only use the US-align web server and/or the command line tool.

3. Download the zip archive (on Windows or Linux) or tar archive (on Mac OS) for US-align precompiled for the OS (Windows, Mac OS or Linux) from the bottom of https://zhanggroup.org/US-align (Fig. 2a).
   ▲ **CAUTION** Do not decompress the zip or tar file for US-align before plugin installation. On Mac OS, the Safari browser will automatically decompress a downloaded zip file. In this case, you can either switch to a different web browser or suppress automatic unzipping at the Safari main menu via 'Safari' – 'Preference' – 'General' and uncheck 'Open safe files after downloading' before performing the download. Alternatively, rather than downloading the tar file through a web browser, download the tar file by command line:

   ```
   curl https://zhanggroup.org/US-align/bin/module/USalignMac.tar.gz -o
   USalignMac.tar.gz
   ```

4. Launch PyMOL. If your PyMOL was installed by the package manager of your Linux distributions (for example, apt-get, dnf and yum), you will need to launch it with sudo/root privileges so that the plugin will be available for all users after installation.
5. In the PyMOL main menu, select 'Plugin' – 'Plugin Manager' to launch the PyMOL plugin manager (Fig. 2b).
6. Under the 'Install New Plugin' tab, choose 'Install from local file' and select the zip or tar archive downloaded from Step 3 (Fig. 2c). This will automatically extract and install the '_init_.py' and 'USalign' files from the zip archive.
7. In the PyMOL plugin manager, verify the successful installation of the plugin by checking whether 'usalign' is listed under the 'Installed Plugin' tab (Fig. 2d).

Structure alignments using the US-align command line program

● **TIMING 35 min**

▲ **CRITICAL** The command line version provides the most comprehensive functionality of US-align. The following examples use US-align to perform pairwise alignments of monomeric and oligomer complex structures, MSTA of three or more structures and search of a query structure through a database, in both sequential and nonsequential alignments. In this section, unless mentioned otherwise, 'Usalign' in the command line should be replaced by the actual path to which the US-align command line tool is installed. As a summary, Table 2 lists the basic syntax and commonly used options of US-align, while the full list of available options can be viewed by running US-align using the '-h' option.

8. While pairwise alignment of two monomeric structures is the most basic usage of US-align, US-align includes also a convenient option for pairwise alignments among multiple structures. Use option A for pairwise alignment of two monomeric structures or option B for one-against-all database search and all-against-all structure comparisons.
   (A) **Pairwise alignment of two monomeric structures**
      (i) For example, align two myoglobins using the following command:

      ```
      USalign 101m.pdb 1mba.pdb
      ```

      The output summary of this run will include the file names, sequence lengths, TM-scores, alignment length, r.m.s.d. and sequence identity of the aligned region and the result of the pairwise alignment (Fig. 3a).

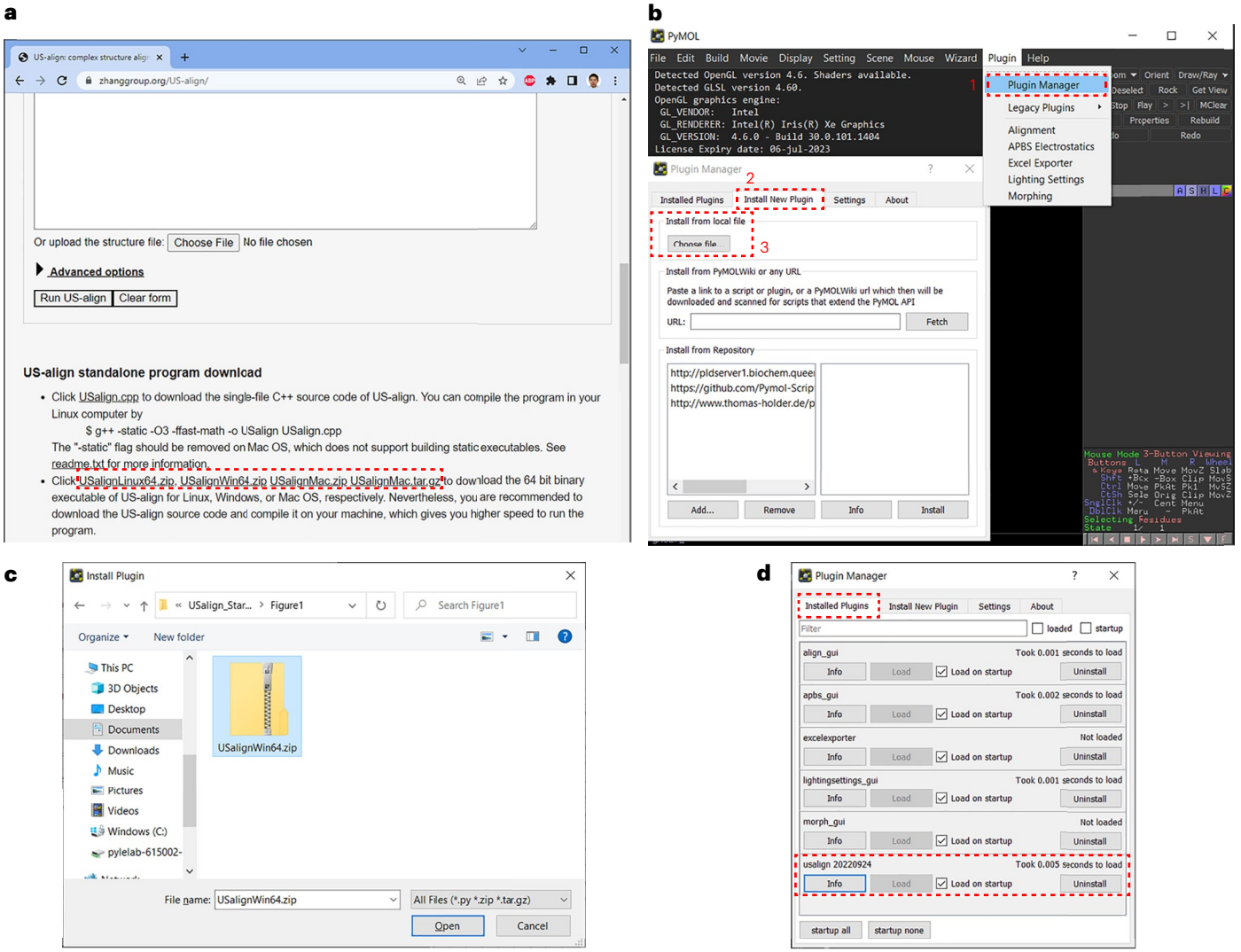

Fig. 2 | Installation of the PyMOL plugin for US-align. **a**, Download the OS-specific zip file with the link listed at the bottom of https://zhanggroup.org/US-align/. **b**, Launch the PyMOL plugin manager. **c**, Install the downloaded zip file. **d**, Verify successful installation of the plugin in the plugin manager.

(ii) By default, US-align only reads the first chain in a multichain complex; similarly, only the first model in a multimodel file is read. This can be changed using the '-ter' option. For example, for a pair of octamers, align each of the eight chains from the first octamer to each of the eight chains from the second octamer using the following command:

```
USalign 4iaj.pdb1 4jhm.pdb1 -ter 1
```

This will perform 8 × 8 = 64 pairwise alignments and report results for all 64 alignments. Here, '-ter 1' reads all chains from the first model. Meanwhile, '-ter 0' will read all chains from all models. Since both structure files have only one model for each chain, using '-ter 0' and '-ter 1' will generate the same alignment result in this example.
▲ **CAUTION** Note that this option is different from the oligomer complex structure alignment, which rotates two complex structures using a uniform rotation matrix, while this option aligns different chain pairs of the complex structures separately.

Table 2 | Common options of the US-align command line tool

Flag	Description	Option values
`-mm`	Alignment task	'`-mm 0`' aligns a pair of monomers (default) '`-mm 1`' aligns a pair of multi-chain oligomers '`-mm 3`' performs pairwise CP alignment '`-mm 4`' performs MSTA '`-mm 5`' and '`-mm 6`' performs pairwise fNS and sNS alignment, respectively
`-dir`	Perform all-against-all alignment or MSTA among a folder of structures	Without '`-dir`', the basic syntax of US-align is '`USalign Structure_1 Structure_2 [option]`', where `Structure_1` and `Structure_2` are the first and second structure in pairwise alignment, while `[option]` are optional flags With '`-dir`', the syntax is '`USalign -dir Folder list_file [option]`', where '`Folder`' is the folder containing the input structures. '`list_file`' is a text file listing the filenames (one file per line) for the input structures in "`Folder`"
`-mol`	The molecule type to align	'`-mol prot`' aligns the protein components '`-mol RNA`' aligns the DNAs and RNAs Default is to align both proteins and nucleic acids
`-atom`	The name of atom (four characters, including spaces) used to represent a residue	'`-atom`' default values are '`-atom " CA "`' for proteins and '`-atom " C3'"`' for nucleic acids, that is, Cα and C3′ atoms for amino acids and nucleotides, respectively '`-atom "PC4'"`' reads Cα atoms from amino acids and both P and C4′ atoms from nucleotides
`-TMscore`	Performs TM-score superimposition for structure pairs with known residue correspondence. (The same as '`-byresi`')	'`-TMscore 1`' assumes residue correspondence between two structures based on the residue sequence number '`-TMscore 2`' assumes residue correspondence based on residue sequence number and chain ID '`-TMscore 5`' or '`-seq`' establishes residue correspondence by sequence alignment '`-TMscore 6`' first derives optimal chain mapping between two oligomers, followed by superimposition for residue pairs with the same residue sequence number '`-TMscore 7`' derives chain mapping followed by superimposition based on sequence alignment
`-I`	Specifies the pairwise sequence alignment between the input structure pair	'`-I align.txt`' specifies the alignment through the FASTA file '`align.txt`'
`-ter`	The portion of input files to read	'`-ter 0`' reads all chains from all models '`-ter 1`' reads all chains from model 1 (default for '`-mm 1`', '`-mm 2`', '`-TMscore 2`', '`-TMscore 6`' or '`-TMscore 7`') '`-ter 2`' reads the first chain (default for all other cases)
`-o`	Generates the aligned/superimposed structure pair in PyMOL format	'`-o sup`' generates six files for the aligned structure pair in PyMOL format: '`sup.pdb`' is `Structure_1` superimposed onto `Structure_2` based on the alignment; '`sup.pml`' is the PyMOL script to show the Cα traces of the aligned region; '`sup_all.pml`' is the script for the Cα traces of the whole chains; '`sup_atm.pml`' is for all atoms of the aligned region; '`sup_atm_all.pml`' is for all atoms of the whole chains, excluding ligands; '`sup_atm_all_lig.pml`' is for all atoms of the whole structures, including ligands
`-rasmol`	Generates the aligned/superimposed structure pair in RasMol format	'`-rasmol sup`' generates five files for the aligned structure pair in RasMol format: '`sup`' is the RasMol script to show the Cα traces of the aligned region; '`sup_all`' is the script for the Cα traces of the whole chains; '`sup_atm`' is for all atoms of the aligned region; '`sup_atm_all`' is for all atoms of the whole chains, excluding ligands; '`sup_atm_all_lig`' is for all atoms of the whole structures, including ligands
`-chimerax`	Generates the aligned/superimposed structure pair in ChimeraX format	'`-chimerax sup`' generates six files for the aligned structure pair in ChimeraX format: '`sup.pdb`' is `Structure_1` superimposed onto `Structure_2` based on the alignment; '`sup.cxc`' is the ChimeraX script to show the Cα traces of the aligned region; '`sup_all.cxc`' is the script for the Cα traces of the whole chains; '`sup_atm.cxc`' is for all atoms of the aligned region; '`sup_atm_all.cxc`' is for all atoms of the whole chains, excluding ligands; '`sup_atm_all_lig.cxc`' is for all atoms of the whole structures, including ligands
`-m`	Outputs the rotation matrix for `Structure_1` relative to `Structure_2`	'`-m matrix.txt`' saves the rotation matrix to '`matrix.txt`' '`-m -`' prints to the rotation matrix to screen
`-fast`	Performs a fast but slightly less accurate alignment	'`-fast`' does not require additional option values
`-outfmt`	Format of the alignment output	'`-outfmt 0`' (default) prints both the metrics from the alignment (r.m.s.d., TM-scores and so on) and the full alignment '`-outfmt 2`' only prints the metrics in a compact table without the full alignment
`-dir1` and `-dir2`	Specify the folders of structures for `Structure_1` and `Structure_2`	For example, '`USalign -dir1 Folder1 list_file1 Structure_2`' aligns files listed by '`list_file1`' from '`Folder1`' to `Structure_2` '`USalign -dir1 Folder1 list_file1 -dir2 Folder2 list_file2`' aligns files listed by '`list_file1`' from '`Folder1`' to files listed by '`list_file2`' from '`Folder2`'
`-suffix`	The filename extension appended to the filenames listed by '`list_file1`', '`list_file2`' or '`list_file`' mentioned above	For example, '`USalign -dir1 Folder1 list_file1 -suffix.ent.gz 2ayh.pdb`', where '`list_file1`' has three rows: '`101m` `1mba` `1ajk`' will align '`Folder1/101m.ent.gz`', '`Folder1/1mba.ent.gz`' and '`Folder1/1ajk.ent.gz`' to '`2ayh.pdb`'

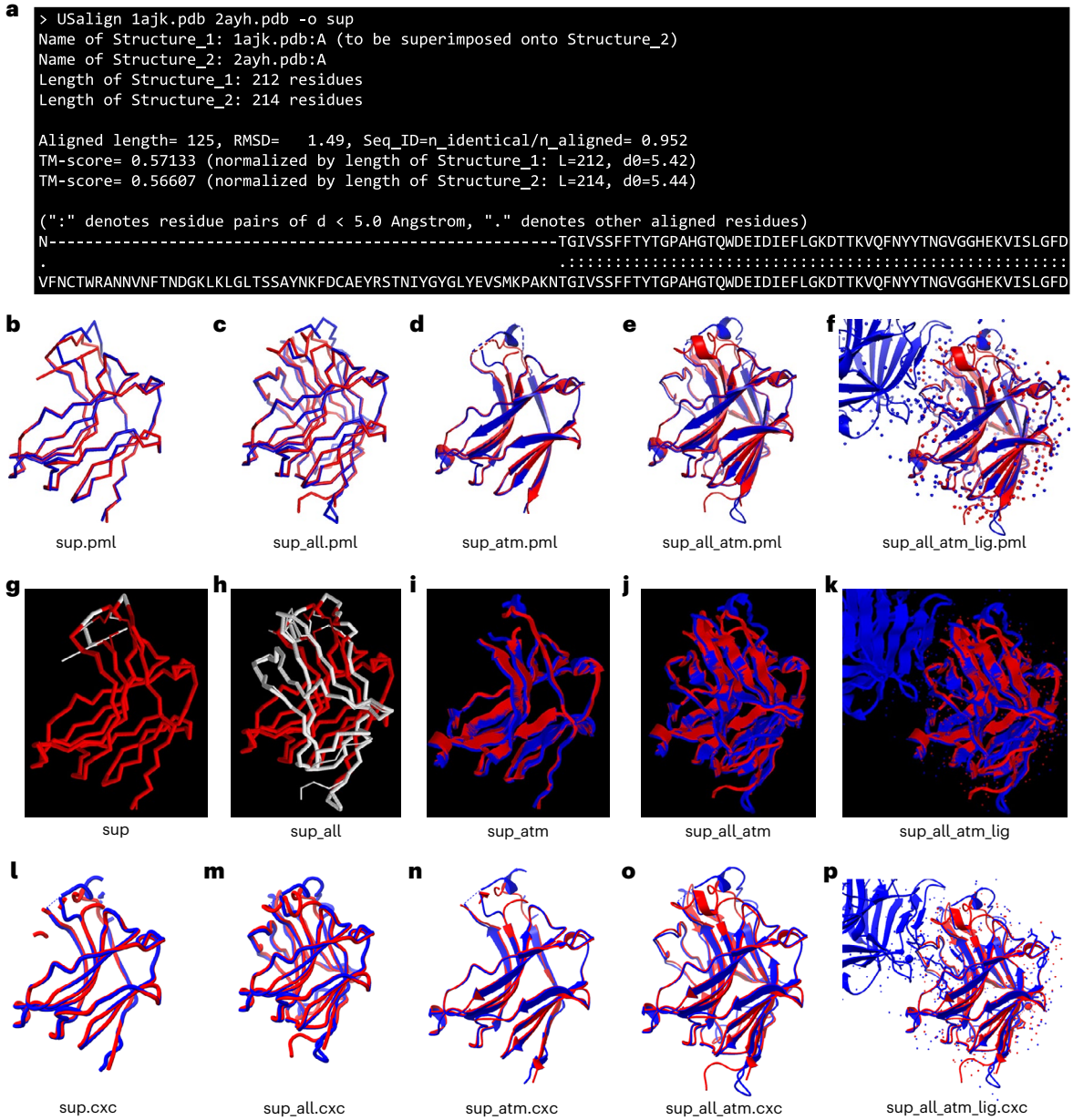

Fig. 3 | An illustrative example of pairwise monomer alignment by the US-align command line program. **a**, An excerpt of the structural alignment summary, including file name, sequence length, alignment length, r.m.s.d. of aligned region, sequence identity, TM-score and the structure-derived sequence alignment (trimmed due to limited space). **b**–**f**, The superimposed structures visualized by PyMOL. Structure 1 (PDB 1ajk) and structure 2 (PDB 2ayh) are shown in blue and red, respectively. **b**, The Cα trace of aligned region. **c**, The Cα trace of the full chain. **d**, The full atomic structure of the aligned region. **e**, The full atomic structure of the whole chain, excluding ligands and other chains. **f**, The full atomic structure of the whole structure, including ligands and other chains of the structure that are not in the alignment. **g**–**k**, The superimposed structures visualized by RasMol, with the Cα trace of aligned region (**g**), the Cα trace of the full chain (**h**), the full atomic structure of aligned region (**i**), the full atomic structure of the whole chain, excluding ligands and other chains (**j**) and the full atomic structure of the whole structure, including ligands and other chains of the structure that are not in the alignment (**k**). In **g** and **h**, the residue pairs aligned within 5 Å are shown in red and other residues are shown in white. In **i**, **j** and **k**, structure 1 (PDB 1ajk) and structure 2 (PDB 2ayh) are shown in blue and red, respectively. **l**–**p**, The superimposed structures visualized by UCSF ChimeraX, with the Cα trace of aligned region (**l**), the Cα trace of the full chain (**m**), the full atomic structure of aligned region (**n**), the full atomic structure of the whole chain, excluding ligands and other chains (**o**) and the full atomic structure of the whole structure, including ligands and other chains of the structure that are not in the alignment (**p**).

(iii) Sometimes, the structure may have both a protein and a nucleic acid component, but only the protein or only the nucleotide component is of interest for the alignment. In this case, only the component of interest should be used for alignment. This can be specified by the '-mol' option. For example, align the RNA component from a pair of RNA-protein complexes using the following command:

```
USalign 3am1.pdb 1evv.pdb -mol RNA
```

Here, '-mol RNA' only aligns the nucleic acid (RNA or DNA) components while '-mol prot' only aligns the protein components.

(iv) US-align can output the superimposed structures in different modes, which can be visualized by PyMOL or RasMol. To generate superimposed structures of the aligned regions in PyMOL format, use the '-o' option (Fig. 3b–f) as follows:

```
USalign 1ajk.pdb 2ayh.pdb -o sup
```

(v) Visualize the Cα trace of the aligned region (Fig. 3b) and the whole chain (Fig. 3c) in PyMOL as follows:

```
pymol -d @sup.pml # aligned regions
pymol -d @sup_all.pml # the whole chain
```

(vi) Visualize the full atomic structure of the aligned region (Fig. 3d), the whole chain (excluding small molecule ligands) (Fig. 3e) and the whole chain including ligands (Fig. 3f) in cartoons as follows:

```
pymol -d @sup_atm.pml # aligned regions
pymol -d @sup_all_atm.pml # the whole chain
pymol -d @sup_all_atm_lig.pml # including ligands
```

The above command also generates the 'sup.pdb' file which corresponds to full length 1ajk.pdb after superimposition.

(vii) On the other hand, use the '-rasmol' option to output superimposed structures in RasMol format (Fig. 3g–k) as follows:

```
USalign 101m.pdb 1mba.pdb -rasmol sup
```

(viii) View the Cα traces as follows:

```
rasmol -script sup # aligned region
rasmol -script sup_all # whole chain>
```

(ix) View full atomic structures as follows:

```
rasmol -script sup_atm # aligned region
rasmol -script sup_all_atm # whole chain
rasmol -script sup_all_atm_lig # including ligands
```

(x) Use a similar option '-chimerax' to output superimposed structures in UCSF ChimeraX format (Fig. 3l–p) as follows:

```
USalign 101m.pdb 1mba.pdb -chimerax sup
```

(xi) View the Cα traces as follows:

```
chimerax --script sup.cxc # aligned region
chimerax --script sup_all.cxc # whole chain
```

(xii) View full atomic structures as follows:

```
chimerax --script sup_atm.cxc # aligned region
chimerax --script sup_all_atm.cxc # whole chainchimerax --script
sup_all_atm_lig.cxc # including ligands
```

(xiii) Use the '-m' option of US-align to outpus the rotation matrix for the superimposition to a file (matrix.txt in the following example) as follows:

```
USalign 101m.pdb 1mba.pdb -m matrix.txt
```

(xiv) Alternatively, use '-m –' to print the rotation matrix to standard output (that is, print to screen) as follows:

```
USalign 101m.pdb 1mba.pdb -m -
```

(B) **Pairwise alignments among many structures**

(i) Perform a quick search of query protein 1mba.pdb, for example, through the I-TASSER template database (listed in the 'Data files' section above) consisting of 96,666 nonredundant protein chains and protein domains, as follows:

```
tar -xvf PDB.tar.bz2 # tarball for the I-TASSER library
USalign 1mba.pdb -dir2./PDB/./PDB/list -suffix.pdb -fast -outfmt
2 > search_result.txt
```

In this command, '-dir2./PDB/' means the second file in the pairwise alignment are found under the './PDB/' folder. './PDB/list' is a text file listing all protein structures to be compared, one protein per line. '-suffix.pdb' means the file name extension is '.pdb'. For example, if './PDB/list' has the following line: 8dceH, the corresponding full file name is './PDB/8dceH.pdb'. In case that the list contains a name '8dceH.pdb', US-align will use the actual name directly. The '-outfmt 2' option generates a compact summary table of results containing the file name, the TM-scores, r.m.s.d., sequence identity (normalized by length of protein 1, protein 2 and aligned region), sequence length and alignment length. Here, the full sequence alignments are omitted, to reduce the size of the output file 'search_result.txt'. The '-fast' uses a faster but slightly less accurate version of US-align that performs less iteration than the standard US-align algorithm. In this example, US-align with '-fast' option cost 32.5 min in total (or 0.02 s per alignment), which is three times faster than 93.3 min in total (or 0.06 s per alignment) by the default version when '-fast' is not used.

▲ **CAUTION** The above command may print a message such as 'WARNING!./PDB/96666.pdb does not exist'. This message is caused by the './PDB/list' file, whose first line shows the total number of entries in the list rather than an entry in the template database. This message can be safely ignored. Note that options '-o' or '-rasmol' will not work when performing database search or all-against-all alignment (Step 9).

(ii) Similarly, US-align can perform all-against-all alignment of all structure files within one folder using option '-dir'. For example, perform all-against-all alignment of the first 20 structures in the './PDB' folders mentioned above as follows:

```
head -20./PDB/list > list.top20.txt
USalign -dir./PDB/ list.top20.txt -suffix.pdb -fast -outfmt 2 >
all_against_all_result.txt
```

9. Perform pairwise alignment of oligomer structures (that is, complex structures with two or more chains) using US-align option '-mm 1' as follows:

```
USalign 4iaj.pdb1 4jhm.pdb1 -mm 1
```

▲ **CRITICAL STEP** By default, US-align with '-mm 1' reads all chains from the first model of each file (that is, '-ter 1'), which is suitable if the input PDB files for the complex structures are asymmetric units. If the input PDB files are biological units (that is, biological assemblies), the '-ter 0' option should be used to read all chains from all models, as different chains from the same biological assembly may come from different models of a PDB format biological assembly file.

10. MSTA (that is, alignment of three or more monomeric chains) in US-align corresponds to option '-mm 4'. For example, to align three RNAs (1evv, 6jxm and 3am1), create a text file named 'list.txt' listing one input per line and then perform MSTA among 1evv.pdb, 6jxm.pdb and 3am1.pdb, which are all located in the current folder './', using the following commands:

```
echo 1evv > list.txt
echo 6jxm >> list.txt
echo 3am1 >> list.txt
USalign -dir./ list -suffix.pdb -mm 4 -mol RNA
```

In this example, '-dir./' specifies the folder for input file. '-mol RNA' instructs US-align to only parse the RNA component, as 3am1.pdb includes both a protein and an RNA. MSTA also works with the '-o' option to generate PyMOL style output (Fig. 4a). Note that MSTA in this step is different from all-against-all alignment (as in Step8B(ii) above). For N input structures, all-against-all alignment generates $N(N-1)/2$ separate pairwise alignments, with each alignment containing two structures. On the other hand, MSTA generates a single alignment matrix that contains all N structures.

11. In addition to sequential alignment (SQ), which the above alignment tasks are built on, there are three types of non-SQs in US-align: CP (option A), fully non-SQ (fNS) (option B) and semi-non-SQs (sNS) (option C), as shown in Fig. 1f–h.

(A) **CP**

CP is a rearrangement of the sequence such that the original termini are linked, and a pair of new termini is created elsewhere (Fig. 1f). Perform CP alignment using option '-mm 3', as follows:

```
USalign 1ajk.pdb 2ayh.pdb -mm 3
```

An excerpt of the output is shown in Fig. 5 for this example.

(B) **fNS**

fNS alignment completely disregards the sequential connection of the residues and treats the atoms of a structure as a cloud of points (Fig. 1g). This is most suitable when aligning a protein against a nucleic acid to detect molecular mimicry. Perform fNS alignment using the following command:

```
USalign 1eh1.pdb 1evv.pdb -mm 5 -atom "PC4'"
```

Here, the '-atom' option specifies the atom used to represent a residue. By default, the C3′ atom and Cα atom are used to represent a nucleotide and an amino acid residue, respectively. This may not be suitable for full-atom protein–nucleic acid alignment, as a nucleotide contains many more atoms than an amino acid.

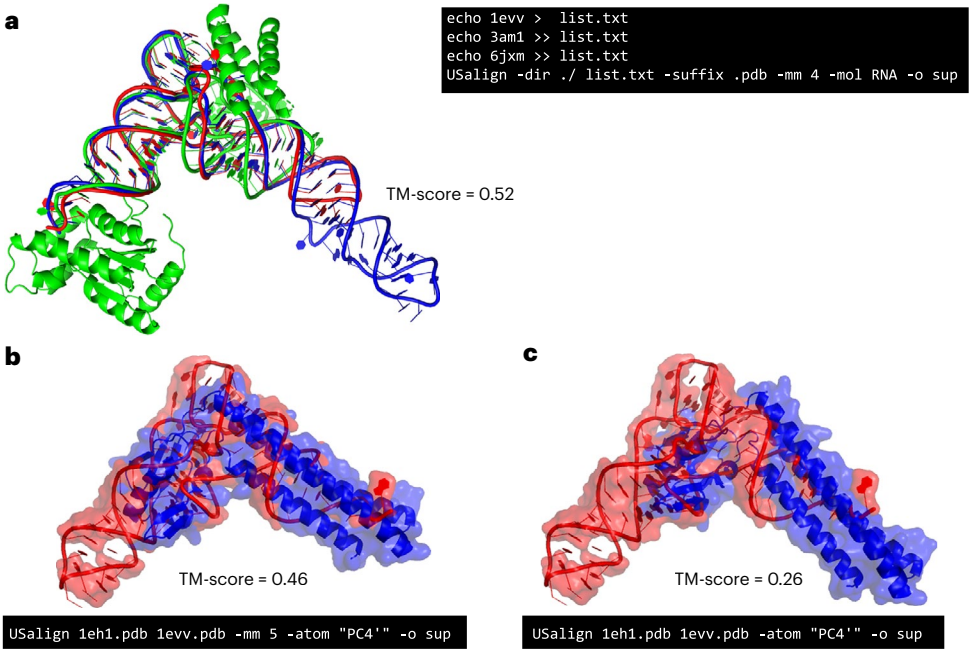

Fig. 4 | Illustrative examples of MSTA and fNS alignments by the US-align command line program. a, MSTA among three RNAs (PDB 1evv, 3am1 and 6jxm in red, green and blue, respectively). Since PDB 3am1 has both a protein component (a tRNA kinase) and a tRNA, option '-mol RNA' is used to specify the alignment of the RNA component only. **b**, The fNS alignment between a tRNA (red) and a ribosomal recycling factor protein (RRF, blue), where the latter mimics tRNA by binding to the tRNA binding site in ribosomes. **c**. SQ for the same tRNA-RRF with a much lower TM-score and less overlap between the envelops (semitransparent surfaces) of the structure pair.

Option '-atom `"PC4'"`' specifies the alignment of the Cα atoms of a protein with the P and C4′ atoms of a nucleic acid molecule. This makes the atomic structure of the protein and the nucleic acid more comparable, as Cα–Cα and P–C4′ distances between adjacent residues are both ~3.8 Å. The structural comparison results using fNS and SQs for the same structure pair are shown in Fig. 4b, c, respectively, showing that fNS alignment generates a better match both visually and in terms of the TM-score for this pair of RNA–protein mimicry.

(C) **sNS**

sNS alignment respects the sequential order within each secondary structure element (α-helix, β-strand or a strand in the nucleic acid double-helix) but disregards the sequential order outside the secondary structure element (Fig. 1h). Perform sNS alignment using the '-mm 6' option, as follows:

```
USalign 1ajk.pdb 2ayh.pdb -mm 6
```

In general, as illustrated in Fig. 1e–h, the alignment search scope in fNS is broader than that of sNS, which is broader than CP, and CP, in turn, is broader than SQ. Therefore, in terms of TM-score, SQ ≤ CP ≤ sNS ≤ fNS. For example, in the structure pair 1ajk and 2ayh, the TM-scores for SQ, CP, sNS and fNS alignments are 0.57, 0.94, 0.95 and 0.95, respectively. The differences in TM-scores primarily arise from variations in residue correspondences, while the overall structural superimpositions (that is, the rotation matrices) are identical across all four cases.

12. Structure superimposition of two structures with known alignment. While by default US-align searches through many structural alignments and returns TM-score of the best

```
Name of Structure_1: 1ajk.pdb:A (to be superimposed onto Structure_2)
Name of Structure_2: 2ayh.pdb:A
Length of Structure_1: 212 residues
Length of Structure_2: 214 residues

Aligned length= 207, RMSD=   1.50, Seq_ID=n_identical/n_aligned= 0.971
TM-score= 0.94331 (normalized by length of Structure_1: L=212, d0=5.42)
TM-score= 0.93466 (normalized by length of Structure_2: L=214, d0=5.44)
(You should use TM-score normalized by length of the reference structure)

(":" denotes residue pairs of d < 5.0 Angstrom, "." denotes other aligned residues)
QTGGSFFEPFNSYNSGTWEKADGYSNGGVFNCTWRANNVNFTNDGKLKLGLTSSAYNKFDCAEYRSTNIYGYGLYEVSMKPAK*N-TGIVSSFFTYTGPAHGTQWDEIDIEFLGKDTTKVQFNYYTNG
...::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::   .:::::::::::::::::::::::::::::::::::::::::
QTGGSFFEPFNSYNSGTWEKADGYSNGGVFNCTWRANNVNFTNDGKLKLGLTSSAYNKFDCAEYRSTNIYGYGLYEVSMKPAK--NTGIVSSFFTYTGPAHGTQWDEIDIEFLGKDTTKVQFNYYTNG

############### ###############
#Aligned atom 1 Aligned atom 2#
 CA  GLN A 132   CA  GLN A   1
 CA  THR A 133   CA  THR A   2
 CA  GLY A 134   CA  GLY A   3
 CA  GLY A 135   CA  GLY A   4
 CA  SER A 136   CA  SER A   5
       ...             ...
 CA  MET A 210   CA  MET A  79
 CA  LYS A 211   CA  LYS A  80
 CA  PRO A 212   CA  PRO A  81
 CA  ALA A 213   CA  ALA A  82
 CA  LYS A 214   CA  LYS A  83
###### Circular Permutation ###
 CA  THR A   2   CA  THR A  85
 CA  GLY A   3   CA  GLY A  86
 CA  ILE A   4   CA  ILE A  87
 CA  VAL A   5   CA  VAL A  88
 CA  SER A   6   CA  SER A  89
       ...             ...
 CA  LYS A 127   CA  LYS A 210
 CA  TYR A 128   CA  TYR A 211
 CA  THR A 129   CA  THR A 212
 CA  SER A 130   CA  SER A 213
 CA  ASN A 131   CA  ASN A 214
############### ###############
```

Fig. 5 | An illustrative example of CP alignment by the US-align command line program. The circularly permuted N- and C-terminal segments for structure 1 (PDB 1ajk) are highlighted in blue and cyan, while the circularly permuted N- and C-terminal segments for structure 2 (PDB 2ayh) are highlighted in red and orange, respectively. The superimposed structure (lower right) and the residue correspondence between 1ajk and 2ayh (lower left) are shown with some middle rows omitted (denoted by '…').

alignment, US-align can also be used to calculate the TM-score between two structures with a specific and known alignment, a function known as structural superimposition. The most frequent use of this type is to calculate the TM-score of the predicted model and native structure of the same protein in protein structure prediction. To illustrate this, we will use the example files from the 'help.zip' mentioned in Table 1. Perform TM-score calculation with the '-TMscore 1' option, as follows:

```
unzip help.zip
USalign help/model.pdb help/native.pdb -TMscore 1
```

This option establishes the residue correspondences between the two structures according to the residue sequence number, which was read from columns 12–27 of a PDB format file (or the _atom_site.auth_seq_id field of a mmCIF format file) (Fig. 6a). The '-TMscore' flag is interchangeable with the '-byresi' flag of the US-align program.

13. Although this section mainly focuses on the use of '-TMscore' for pairwise superimposition, the flag can also be used for pairwise superimposition of a folder of structures (options '-dir', '-dir1' or '-dir2'). For example, to perform one-against-all or all-against-all superimposition

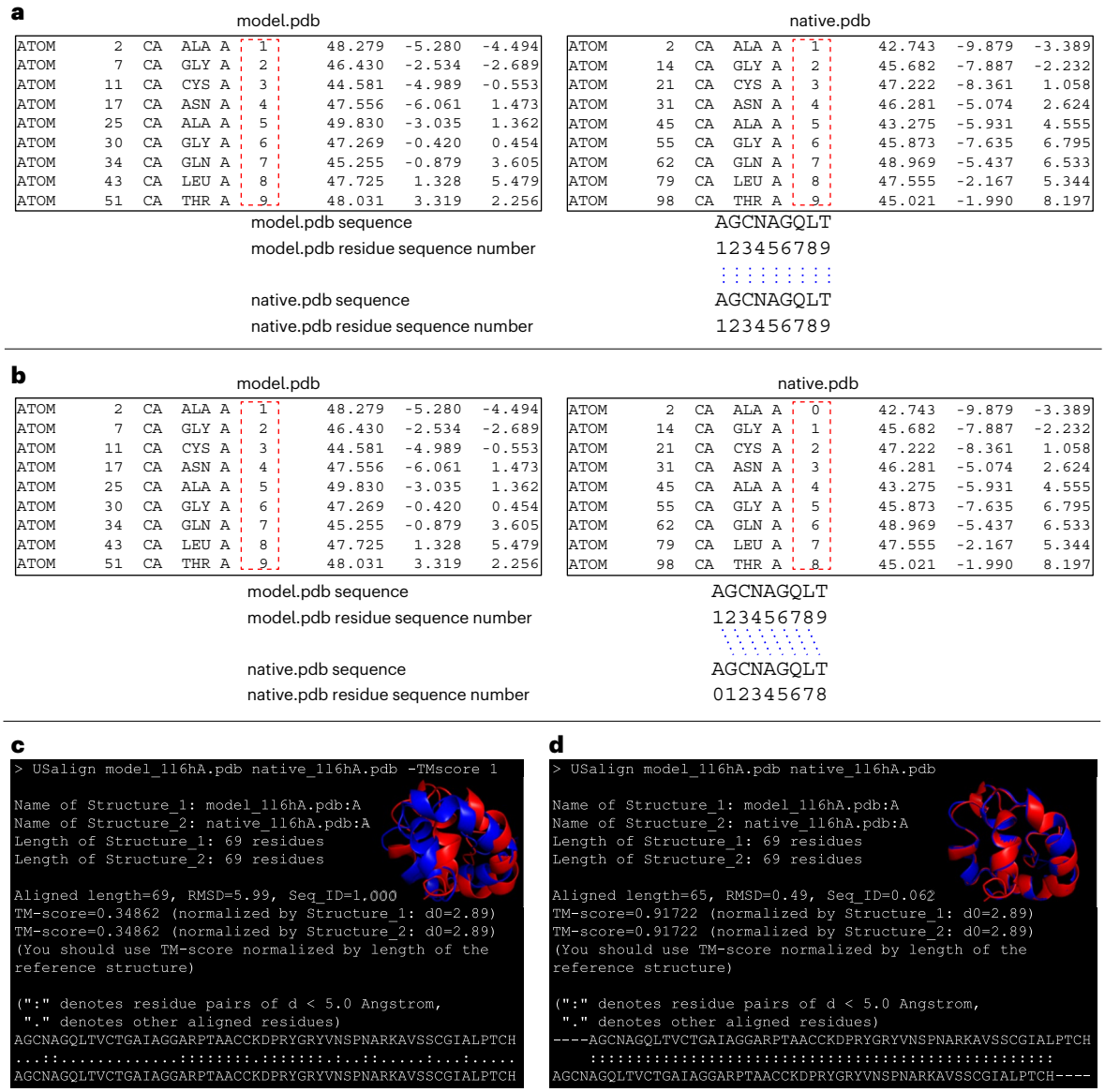

Fig. 6 | Illustration of TM-score calculations with different options of US-align. **a**, **b**, A calculation TM-score by structural superimposition using option '-TMscore 1', which compares the residue pairs with the same residue sequence number (that is, residue index) read from columns 23–27 (red dashed box) of a PDB format (or the _atom_site.auth_seq_id field of a mmCIF format file). **a**, The TM-score was correctly calculated because the residue sequence numbers in two files were correctly labeled, and therefore, correct residue pairs are compared (dotted lines). **b**, The TM-score was incorrectly calculated because there is a shift on the residue sequence number of the native structure. **c**, **d**, The difference between structural superimposition (with the '-TMscore' option) and structural alignment (without the '-TMscore' option) for the predicted (model_1l6hA.pdb) and experimental (native_1l6hA.pdb) structures of PDB 1l6h chain A. In this example, the model was predicted by a template-based modeling (TBM) approach which has the query-template alignment shifted by four residues compared with the best alignment. **c**, A superimposition with the '-TMscore 1' option reflects the incorrect modeling in TBM and reports a low TM-score of 0.349 and a high r.m.s.d. of 5.99 Å. **d**, A structural alignment without the '-TMscore' option by US-align performs a sequence-independent alignment based on structural similarity, reporting a much higher TM-score of 0.917 and lower r.m.s.d. of 0.49 Å with a shifted sequence alignment between the model and native.

among three different conformations (native.pdb, model.pdb and model2.pdb) in the folder of 'help/', use the following command:

```
USalign help/native.pdb -dir2 help/ help/list.txt -TMscore 1
USalign -dir help/ help/list.txt -TMscore 1
```

BOX 2

Differences in the TM-score reported by a default US-align run (structural alignment) and that from the '-TMscore' option (structure superimposition)

What is the difference between the TM-score calculated by Step 12 using the '-TMscore' options versus the default TM-score returned from US-align? The '-TMscore' option (Step 13) compares two structures based on a priori residue equivalency, for example, based on the residue sequence number in the PDB file, a predefined alignment file or the residue correspondence identified by a sequence alignment program; this is called 'structure superimposition'. On the other hand, US-align by default searches many possible alignments and reestablishes the optimal equivalent residues of two structures based on the structure similarity (regardless of the sequences) and then outputs the highest TM-score of the optimal alignment; this is called 'structural alignment'. The TM-score reported by US-align through structural alignment is generally higher than that by structural superimposition. An example to illustrate the difference between the two is shown in Fig. 6c,d.

An alternative approach to calculate the TM-score between model and native structure is to use the standard TM-score program, which is available at https://zhanggroup.org/TM-score/TMscore.cpp and includes all options described above in a similar way as US-align. The two approaches generate identical TM-score results, but we provide the options here for the convenient and versatile use of the US-align program.

▲ **CRITICAL STEP** This option cannot be used for MSTA, for example, in combination with option '-mm 4'.

▲ **CAUTION** This option supposes that the two structure files have consistent residue sequence numbers with the sequences. In case that the residue sequence numbers are inconsistent, the calculated TM-score using this option will be incorrect (Fig. 6b). Meanwhile, if there is clear residue sequence number correspondence between two structures, failure to specify the '-Tmscore' option can result in incorrect interpretation of structural similarities (Fig. 6d and Box 2).

14. If the residue sequence numbers of the two structures are inconsistent, the TM-score can be alternatively calculated through the '-I' option. Provide a FASTA format pairwise sequence alignment, called 'help/align.txt' in the following example, as follows:

```
>model.pdb
AGCNAGQLT
>native.pdb
AGCNAGQLT
```

Superimpose the two structures according to the sequence alignment specified by 'help/align.txt' using the '-I' option as follows:

```
USalign help/model.pdb help/native.pdb -I help/align.txt
```

15. Another approach is to use the '-TMscore 5' option, as follows. This option first establishes the residue correspondence by a simple sequence alignment, and then performs the TM-score superimposition based on established residue correspondence:

```
USalign help/model.pdb help/native.pdb -TMscore 5
```

The '-seq' option is the shorthand form of '-TMscore 5':

```
USalign help/model.pdb help/native.pdb -seq
```

Here, '-TMscore 5' is for superimposition between a pair of monomeric structures, and the equivalent option for a complex structure pair is '-TMscore 7' (Step 18).

▲ **CAUTION** While this option may seem handy and work well in most cases when the sequence similarity is obvious (for example, Fig. 6b), the automatically established sequence alignment may sometimes be different from the correct residue-level correspondence when the sequence similarity is low, and the sequence alignment is ambiguous. A manual confirmation is recommended when using this option.

16. Superimpose a pair of multichain complex structures, where residue correspondence is established by both the chain ID and the residue sequence number through option '-TMscore 2' as follows:

```
USalign help/modelComplex.pdb help/nativeComplex.pdb -TMscore 2
```

When running US-align with the flag '-TMscore 2', '-TMscore 6' and '-TMscore 7' (Steps 17–18 below), US-align will by default read all chains from the first model (that is, '-ter 1'). When superimposing biological assemblies with multiple models, set '-ter 0'.

17. If the chain ID correspondence is not available, use option '-TMscore 6' to calculate the TM-score of complexes. In this option, US-align first derives an optimal chain mapping, followed by TM-score superimposition for residues with the same residue sequence number in the mapped chain pair. This option is particularly useful for evaluation of structure prediction of symmetric complex structures that contain homo-chains, as follows:

```
USalign help/modelComplex.pdb help/nativeComplex.pdb -TMscore 6
```

In this example, the input structures are the prediction structure model and native structure of PDB 7yr6, which consist of one RNA (chain A) and four identical copies of the CsrA proteins (chains B, C, D and E). Despite identical chain labels, the optimal chain correspondences for chains A, B, C, D and E from the predicted structure are chains A, E, D, C and B of the native structure. Therefore, using the suboptimal chain correspondence derived from '-TMscore 2' results in TM-score of 0.41, which is worse than TM-score of 0.55 calculated with optimized chain correspondence by '-TMscore 6'.

18. If neither the chain ID correspondence nor the residue level correspondence is known, use option '-TMscore 7' to calculate the TM-score of two complexes. In this option, US-align also derives an optimal chain-to-chain mapping, followed by TM-score superimposition that is guided by sequence alignment of the mapped chain pair, as follows:

```
USalign help/modelComplex.pdb help/nativeComplex.pdb -TMscore 7
```

▲ **CAUTION** While this option may seem handy, the automatically established sequence alignment may be different from the interested correspondence sometime, similar to option '-TMscore 5'. A manual confirmation is recommended when using this option.

◆ **TROUBLESHOOTING**

Structure alignment using the US-align plugin in PyMOL

● **TIMING 1 min**

▲ **CRITICAL** The US-align plugin allows the alignment of structure objects within the PyMOL interface, which can considerably facilitate intuitive analysis of structures within PyMOL. Although the PyMOL plugin supports all pairwise alignment tasks in US-align, including monomeric alignment, oligomer alignments and non-SQ, options '-dir', '-dir1', '-dir2' and '-suffix' are not supported by this plugin. Therefore, the plugin does not yet support database search or MSTA. For such purposes (for example, database search and MSTA), the command line tool

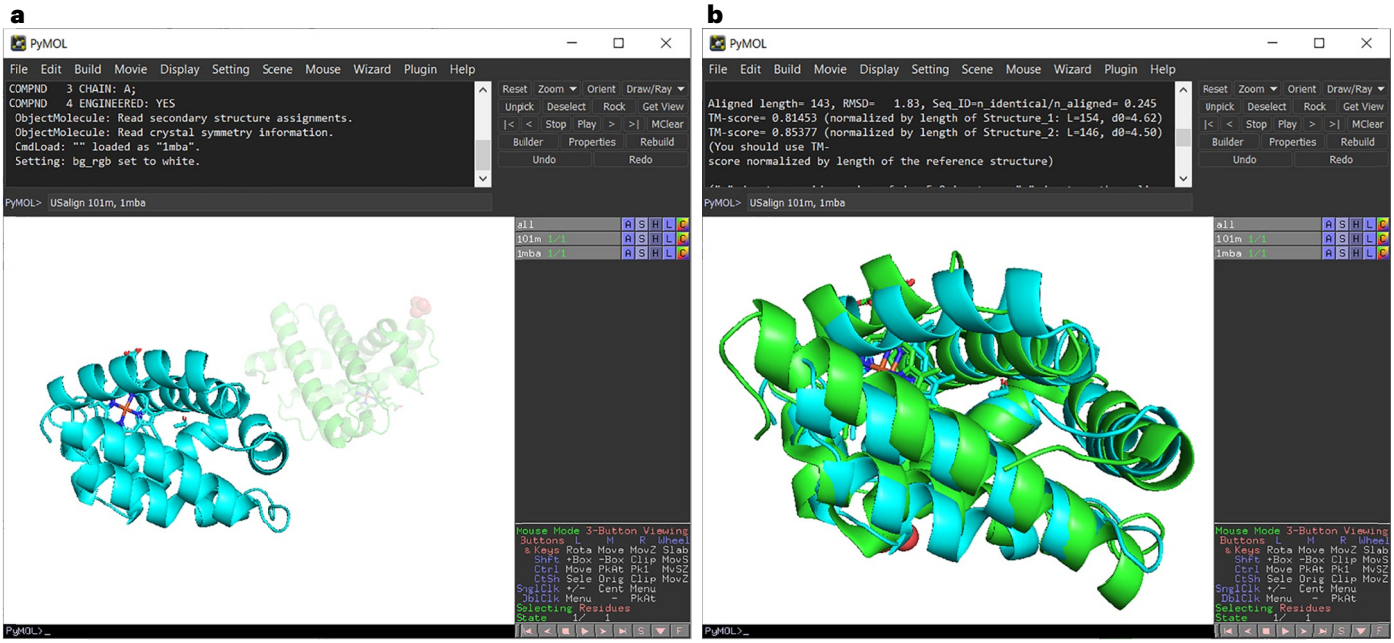

Fig. 7 | An example of pairwise alignment by the US-align PyMOL plugin. **a**, Before US-align alignment. **b**, After US-align alignment.

can be used, which outputs PyMOL scripts to display superimposed structures. Meanwhile, the plugin does not support changing the output format by '-o', '-m' or '-outfmt', as specific output format is required for interfacing with PyMOL.

19. To align 101m.pdb and 1mba.pdb in PyMOL, open the two structures into the same PyMOL session and use the following PyMOL command:

```
USalign 101m, 1mba
```

Here, '101m' and '1mba' are the PyMOL objects corresponding to the structures for alignment (Fig. 7).

▲ **CAUTION** In the above examples, although the original file names are 101m.pdb, 1mba.pdb, 4iaj.pdb1 and 4jhm.pdb1, the file name extensions '.pdb' and '.pdb1' should be excluded in the PyMOL command prompt because the PyMOL object names do not contain the file name extensions.

◆ **TROUBLESHOOTING**

20. The third argument of the 'USalign' command can be used to specify more advanced options of US-align. For example, to perform oligomer alignment (corresponding to US-align option '-mm 1') of 4iaj.pdb1 and 4jhm.pdb1, open the two files in PyMOL and run the following PyMOL command:

```
USalign 4iaj, 4jhm, " -mm 1 "
```

21. By default, the US-align plugin uses the precompiled US-align command line tool that is installed together with the plugin script. To use a US-align command line tool installed at a nonstandard location, use the 'exe' option of the PyMOL command. For example, to use the US-align command line tool installed at '/usr/bin/usalign/USalign' use the following command:

```
USalign 101m, 1mba, exe="/usr/bin/usalign/USalign"
USalign 4iaj, 4jhm, " -mm 1 ", exe="/usr/bin/usalign/USalign"
```

22. The '-mm' option also allow non-SQ after opening the corresponding pairs of PDB files in the same PyMOL session. Perform CP alignment with '-mm 3' as the third argument as follows:

```
USalign 1ajk, 2ayh, "-mm 3"
```

23. Or perform fNS and sNS alignments using '-mm 5' and '-mm 6', respectively:

```
USalign 1eh1, 1evv, "-mm 5 -atom "PC4'""
USalign 1ajk, 2ayh, "-mm 6"
```

The sNS alignment preserves the sequential order within each secondary structure element (helix and strand) and is therefore suitable for aligning structures of the same molecule type (for example, 1ajk and 2ayh are both proteins). The fNS alignment disregards all sequential order, even within secondary structure elements, and is therefore more suitable for aligning a protein against a nucleic acid (for example, 1eh1 and 1evv are protein and tRNA respectively). By default, US-align uses the C3′ and Cα atoms to represent a nucleotide and an amino acid residue, respectively. This may be a problem when aligning a protein to a nucleic acid, as the size of a nucleotide is typically larger than an amino acid. Therefore, in the above fNS example, two atoms (P and C4′) are used to represent a nucleotide while only one atom (Cα) is used to represent an amino acid, as specified by '-atom 'PC4''.

24. Via the '-TMscore' option, the plugin can perform TM-score-based superimposition for structure pairs with known residue correspondence, as illustrated below using example files 'model.pdb', 'native.pdb', 'modelComplex.pdb' and 'nativeComplex.pdb' extracted from help.zip. Perform TM-score calculation with the '-TMscore 1' option, where residue correspondence is established by the residue sequence number (columns 12–27 of a PDB format file (or the _atom_site.auth_seq_id field of a mmCIF format file) as follows:

```
USalign model_, native, "-TMscore 1"
```

Use option '-TMscore 2' to superimpose a pair of multi-chain oligomers, where residue correspondence is established by both the chain ID and the residue sequence number:

```
USalign modelComplex, nativeComplex, "-TMscore 2"
```

Use '-TMscore 5' or '-seq' option to perform a simple sequence alignment, based on which the TM-score superimposition is performed:

```
USalign model_, native, "-TMscore 5"
USalign model_, native, "-seq"
```

▲ **CAUTION** In the above examples, the PyMOL object name for model.pdb is 'model_' because 'model' is a keyword in PyMOL.

Structure alignment with the US-align web server

● **TIMING 1 min**

25. The US-align web server at https://zhanggroup.org/US-align/ has three tabs for pairwise (sequential and nonsequential) monomer alignment (option A), pairwise oligomer alignment (option B) and MSTA (option C).
    (A) **Monomer alignment**
        (i) Align a pair of monomeric chains through the default tab 'Monomer alignment'. Paste the contents of the PDB or mmCIF format structure files in the textboxes or upload the uncompressed files corresponding to structure 1 and structure 2. The user may modify 'Advanced options'. For simple pairwise structure alignment, no advanced options need to be set.

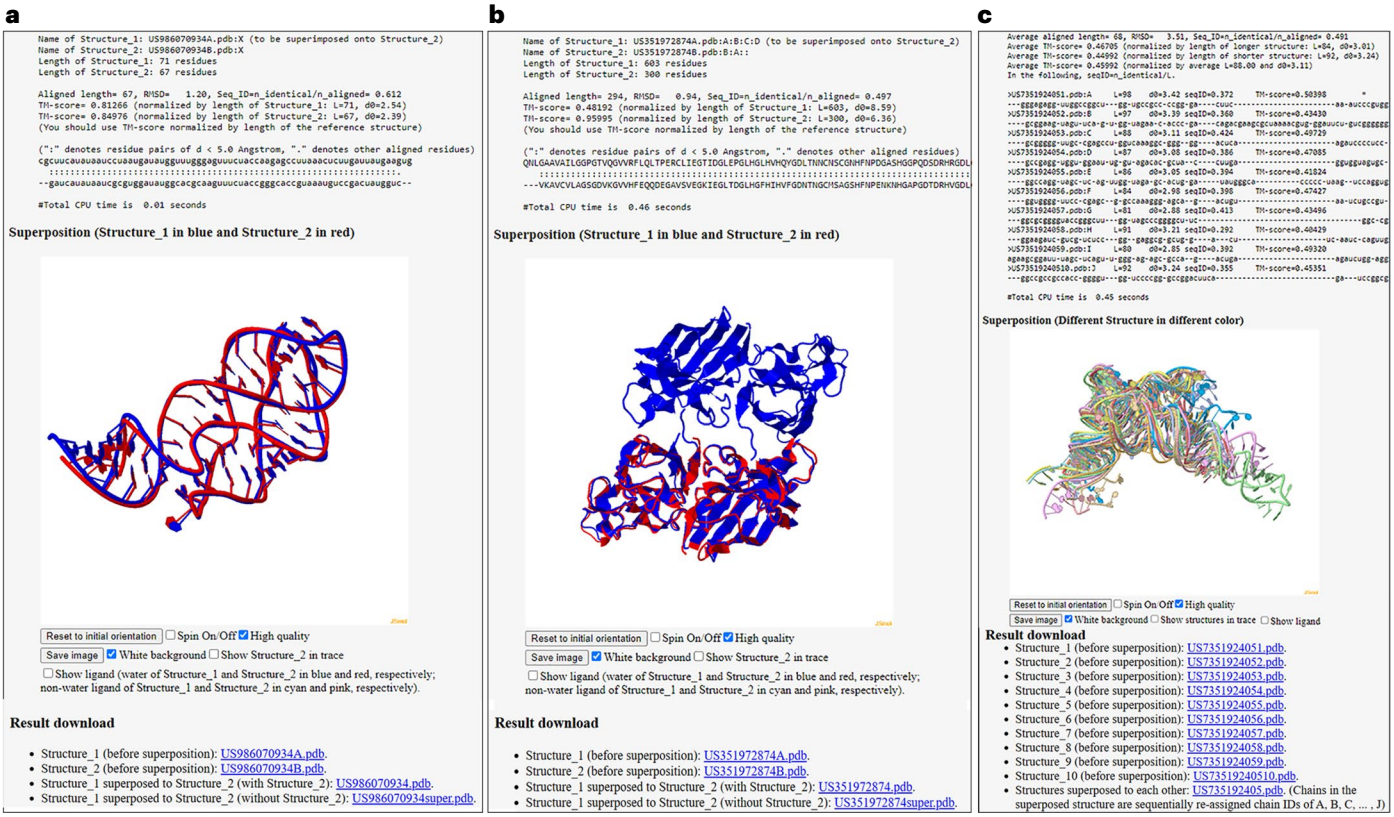

Fig. 8 | Example output of US-align web server. a, A pairwise monomer structure alignment of two riboswitches (PDB ID: 1y26 chain X and 4lx5 chain A). **b**, An oligomer complex structure alignment between human copper chaperone for superoxide dismutase and Cu,Zn superoxide dismutase B (PDB ID 1do5 and 1xso). **c**, MSTA of ten tRNAs. Each web page output includes a summary of the alignment at the top, followed by the superimposed structures shown by JSmol applet. Due to limited space, long sequence alignments are trimmed.

(ii) To superimpose two structures with known residue correspondence (corresponding to option '-TMscore' of the command line tool), select either 'assume correspondence between a pair of residues with the same residue index in the two structures' or 'perform superimposition based on local sequence alignment'.

(iii) To align only the protein or only the nucleic acid component of the input structure, change 'Molecule type' to either 'protein' or 'RNA/DNA', respectively (corresponding to option '-mol' of the command line tool).

(iv) To perform non-SQ, change the 'Sequence order dependency' option to 'Circular permutation alignment', 'Fully non-sequential (fNS) alignment' or 'Semi non-sequential (sNS) alignment' (corresponding to '-mm 3', '-mm 5' or '-mm 6', respectively).

(v) By default, US-align only reads the Cα atom of an amino acid or the C3′ atom of a nucleotide for alignment. The user may use a different nucleotide backbone atom through the 'Which backbone atom is used to represent a residue?' option (corresponding to option '-atom' of the command line tool). When aligning a protein against a nucleic acid, set this option to 'P (RNA/DNA), C4′ (RNA/DNA) and Cα (protein)'.

(vi) Finalize the submission through the 'Run US-align' button. Within a few seconds, the output webpage will be generated (Fig. 8a), which starts with the US-align command line output, including the r.m.s.d., TM-score and full alignment. This is followed by a JSmol applet[45] to show the aligned/superimposed structures where

the first and second structures are colored in blue and red, respectively. The spinning of the structure, the color of the background and the display of ligands can be toggled by the checkboxes beneath the applet. The last section contains four structures to download: the original structure 1, structure 2, structure 1 aligned/superimposed onto structure 2 (one file with and another file without structure 2).

(B) **Pairwise oligomer alignment**

(i) To align two multichain complex structures, switch to the 'Oligomer alignment' tab (corresponding to option '-mm 1' of the command line tool).

(ii) Paste the contents of the PDB or mmCIF format structure files in the textboxes or upload the uncompressed files corresponding to structure 1 and structure 2. Similar to the previous tab, this tab also contains advanced options for 'Molecule type' and 'Which backbone atom is used to represent a residue'. This tab also contains another advanced option for 'How many models to read from a multi-model structure file?' (corresponding to option '-ter' of the command line tool); choose 'Only read the first model' if the input files are asymmetric units or choose 'Read all models' if the input files are biological assembly. The output webpage (Fig. 8b) also consists of three sections: the US-align command line output, the JSmol applet for the aligned structure pair, and the aligned structures to download.

(C) **MSTA**

(i) To align at least three monomeric chains into a consensus alignment, switch to the 'Multiple structure alignment' tab (corresponding to option '-mm 4' of the command line tool). The webpage only has three textboxes to paste or upload the first three structures. To add more structures, use the 'Add structure' button to add more textboxes. Alternatively, instead of uploading input structures one-by-one, use the 'Alternatively, upload all structure files within a single zip archive' option at the bottom of the page to upload a zip file that contains all input structures. The output webpage (Fig. 8c) also consists of three sections: the US-align command line output, the JSmol applet for the aligned structures and the aligned structures to download. In the structure file containing all aligned structures, each of the ten input structures will have a different chain ID of A, B, C, ..., or J.

Troubleshooting

Troubleshooting advice can be found in Table 3.

Table 3 | Troubleshooting table

Step	Error message or problem	Possible reason	Solution
1	The command requires the command line developer tools. Would you like to install?	clang++ is unavailable on MacOS due to lack of Xcode command line tools	Install Xcode command line tools by the following command 'xcode-select –install'. It takes ~10 min to download and install, after which clang++ should be available
	g++: command not found	g++ is not available on your Linux distribution	Check that g++ is available by 'g++ --version'. If not available, install g++. On Ubuntu and Debian, this can be achieved by 'sudo apt install g++'. On Red Hat Enterprise Linux, Fedora, CentOS and Rocky Linux, install g++ by 'sudo dnf install gcc-c++' or 'sudo yum install gcc-c++'
2	'USalign' cannot be opened because the developer cannot be verified. macOS cannot verify that this app is free from malware	MacOS Gatekeeper automatically forbids binary executables downloaded from a third-party website	Compile US-align from source code (Step 1) or download the US-align program by command line rather than a web browser using the following command: 'curl https://zhanggroup.org/US-align/bin/module/USalignMac.zip -o USalignMac.zip; unzip USalignMac.zip'
8-18	Warning! Cannot parse file. Chain number 0	The file does not exist or (on Windows only) the file is compressed	Make sure the input file path is correct. On Windows, make sure the input file is not compressed, as US-align can only read gzip compressed files on Unix-like systems (for example, by Mac OS and Linux, including Windows Subsystem for Linux)

Table 3 (continued) | Troubleshooting table

Step	Error message or problem	Possible reason	Solution
19	Traceback (most recent call last): file '/Applications/PyMOL.app/Contents/lib/python3/site-packages/pmg_tk/startup/USalign/__init__.py', line 117, in usalign assert len(matrix) == 3 * 4 AssertionError	The file permission of the 'USalign' binary executable is changed during installation of the PyMOL plugin in Mac OS	On Mac OS, for some recent versions of PyMOL, the 'USalign' executable included in USalignMac.tar.gz may lose executable permission due to security policy. This can be fixed by running the following command in the PyMOL command line to reinstall a precompiled US-align executable with correct permission: '`conda install -c bioconda usalign`' If the issue persists, use the 'exe' option to specify the path to a working copy of the US-align binary executable as explained at the end of Step 21

Timing

Steps 1–2, installing the US-align command line tool: 1 min
Steps 3–7, installing the PyMOL plugin for US-align: 2min
Steps 8–18, structure alignment using the US-align command line program: 35 min (most of the time is spent on the database search (Step 8B), which takes 32.5 min)
Steps 19–24, structure alignment using the US-align plugin in PyMOL: 1 min
Step 25, structure alignment through the US-align web server: 1 min

Anticipated results

The outputs of the US-align command line tool, the PyMOL plugin and the web server all include a short summary table of structural alignment results that list the TM-score, the r.m.s.d. of aligned region and the structure-derived sequence alignment (for example, Fig. 3a). Moreover, for non-SQ, the correspondence between each pair of aligned residues is also listed in the table (for example, Fig. 5). The PyMOL plugin and the web server display the superimposed structures (Figs. 7 and 8), which can also be generated by the command line tool if the '-o' and/or '-rasmol' option is specified (Fig. 3).

Reporting summary

Further information on research design is available in the Nature Portfolio Reporting Summary linked to this article.

Data availability

PDB files for PDB IDs 101m, 1mba, 4jhm, 4iaj, 1eh1, 1evv, 6jxm, 3am1, 1ajk and 2ayh are available through https://rcsb.org. The non-redundant PDB library (that is, the I-TASSER template library) is updated on a weekly basis and available at https://zhanggroup.org/library/PDB.tar.bz2. Files to demonstrate the '-TMscore' option of US-align are available at https://zhanggroup.org/TM-score/help.zip.

Code availability

The US-align web server and source code are available at https://zhanggroup.org/US-align/. The code in this Protocol has been peer reviewed.

Received: 16 June 2024; Accepted: 1 April 2025;
Published online: 02 July 2025

References

1. Zhang, Y. & Skolnick, J. Scoring function for automated assessment of protein structure template quality. *Proteins* **57**, 702–710 (2004).
2. Xu, J. R. & Zhang, Y. How significant is a protein structure similarity with TM-score = 0.5? *Bioinformatics* **26**, 889–895 (2010).
3. Gong, S., Zhang, C. & Zhang, Y. RNA-align: quick and accurate alignment of RNA 3D structures based on size-independent TM-scoreRNA. *Bioinformatics* **35**, 4459–4461 (2019).
4. Kabsch, W. A solution for the best rotation to relate two sets of vectors. *Acta Crystallogr. A* **32**, 922–923 (1976).
5. Zhang, Y. & Skolnick, J. TM-align: a protein structure alignment algorithm based on the TM-score. *Nucleic Acids Res.* **33**, 2302–2309 (2005).
6. Holm, L. & Sander, C. Dali: a network tool for protein structure comparison. *Trends Biochem. Sci.* **20**, 478–480 (1995).
7. Shindyalov, I. N. & Bourne, P. E. Protein structure alignment by incremental combinatorial extension (CE) of the optimal path. *Protein Eng.* **11**, 739–747 (1998).
8. Krissinel, E. & Henrick, K. Secondary-structure matching (SSM), a new tool for fast protein structure alignment in three dimensions. *Acta Crystallogr. D* **60**, 2256–2268 (2004).
9. Yang, Y., Zhan, J., Zhao, H. & Zhou, Y. A new size-independent score for pairwise protein structure alignment and its application to structure classification and nucleic-acid binding prediction. *Proteins* **80**, 2080–2088 (2012).
10. Ge, P. & Zhang, S. STAR3D: a stack-based RNA 3D structural alignment tool. *Nucleic Acids Res.* **43**, e137 (2015).
11. Dror, O., Nussinov, R. & Wolfson, H. J. The ARTS web server for aligning RNA tertiary structures. *Nucleic Acids Res.* **34**, W412–W415 (2006).
12. Zheng, J., Xie, J., Hong, X. & Liu, S. RMalign: an RNA structural alignment tool based on a novel scoring function RMscore. *BMC Genomics* **20**, 276 (2019).
13. Nguyen, M. N., Sim, A. Y., Wan, Y., Madhusudhan, M. S. & Verma, C. Topology independent comparison of RNA 3D structures using the CLICK algorithm. *Nucleic Acids Res.* **45**, e5 (2017).
14. Mukherjee, S. & Zhang, Y. MM-align: a quick algorithm for aligning multiple-chain protein complex structures using iterative dynamic programming. *Nucleic Acids Res.* **37**, e83 (2009).
15. Minami, S., Sawada, K. & Chikenji, G. MICAN: a protein structure alignment algorithm that can handle multiple-chains, inverse alignments, Cα only models, alternative alignments, and non-sequential alignments. *BMC Bioinformatics* **14**, 24 (2013).
16. Dong, R., Peng, Z., Zhang, Y. & Yang, J. mTM-align: an algorithm for fast and accurate multiple protein structure alignment. *Bioinformatics* **34**, 1719–1725 (2018).
17. Konagurthu, A. S., Whisstock, J. C., Stuckey, P. J. & Lesk, A. M. MUSTANG: a multiple structural alignment algorithm. *Proteins* **64**, 559–574 (2006).
18. Menke, M., Berger, B. & Cowen, L. Matt: local flexibility aids protein multiple structure alignment. *PLoS Comput. Biol.* **4**, e10 (2008).
19. Zhang, C., Shine, M., Pyle, A. M. & Zhang, Y. US-align: universal structure alignments of proteins, nucleic acids, and macromolecular complexes. *Nat. Methods* **19**, 1109–1115 (2022).
20. Zhang, C. & Pyle, A. M. A unified approach to sequential and non-sequential structure alignment of proteins, RNAs, and DNAs. *iScience* https://doi.org/10.1016/j.isci.2022.105218 (2022).
21. Das, R. et al. Assessment of three-dimensional RNA structure prediction in CASP15. *Proteins* **91**, 1747–1770 (2023).
22. Studer, G., Tauriello, G. & Schwede, T. Assessment of the assessment—all about complexes. *Proteins* **91**, 1850–1860 (2023).
23. Zhang, C., Freddolino, P. L. & Zhang, Y. COFACTOR: improved protein function prediction by combining structure, sequence and protein–protein interaction information. *Nucleic Acids Res.* **45**, W291–W299 (2017).
24. Zhang, C. X., Zheng, W., Freddolino, P. L. & Zhang, Y. MetaGO: predicting gene ontology of non-homologous proteins through low-resolution protein structure prediction and protein protein network mapping. *J. Mol. Biol.* **430**, 2256–2265 (2018).
25. Laskowski, R. A., Watson, J. D. & Thornton, J. M. ProFunc: a server for predicting protein function from 3D structure. *Nucleic Acids Res.* **33**, W89–W93 (2005).
26. Yang, J., Roy, A. & Zhang, Y. Protein–ligand binding site recognition using complementary binding-specific substructure comparison and sequence profile alignment. *Bioinformatics* **29**, 2588–2595 (2013).
27. Roy, A. & Zhang, Y. Recognizing protein-ligand binding sites by global structural alignment and local geometry refinement. *Structure* **20**, 987–997 (2012).
28. Brylinski, M. & Skolnick, J. A threading-based method (FINDSITE) for ligand-binding site prediction and functional annotation. *Proc. Natl Acad. Sci. USA* **105**, 129–134 (2008).
29. Zhang, W., Bell, E. W., Yin, M. & Zhang, Y. EDock: blind protein-ligand docking by replica-exchange monte carlo simulation. *J. Cheminform.* **12**, 37 (2020).
30. Baspinar, A., Cukuroglu, E., Nussinov, R., Keskin, O. & Gursoy, A. PRISM: a web server and repository for prediction of protein–protein interactions and modeling their 3D complexes. *Nucleic Acids Res.* **42**, W285–W289 (2014).
31. Guerler, A., Govindarajoo, B. & Zhang, Y. Mapping monomeric threading to protein-protein structure prediction. *J. Chem. Inf. Model.* **53**, 717–725 (2013).
32. Zhou, X. G., Hu, J., Zhang, C. X., Zhang, G. J. & Zhang, Y. Assembling multidomain protein structures through analogous global structural alignments. *Proc. Natl Acad. Sci. USA* **116**, 15930–15938 (2019).
33. Zhou, X. et al. I-TASSER-MTD: a deep-learning-based platform for multi-domain protein structure and function prediction. *Nat. Protoc.* **17**, 2326–2353 (2022).
34. Pearce, R., Huang, X. Q., Setiawan, D. & Zhang, Y. EvoDesign: designing protein–protein binding interactions using evolutionary interface profiles in conjunction with an optimized physical energy function. *J. Mol. Biol.* **431**, 2467–2476 (2019).
35. Zhang, J., Liang, Y. & Zhang, Y. Atomic-level protein structure refinement using fragment-guided molecular dynamics conformation sampling. *Structure* **19**, 1784–1795 (2011).
36. Liu, Z., Zhang, C., Zhang, Q., Zhang, Y. & Yu, D.-J. TM-search: an efficient and effective tool for Protein Structure Database search. *J. Chem. Inf. Model.* **64**, 1043–1049 (2024).
37. Zhu, Y., Tong, C., Zhao, Z. & Lu, Z. MineProt: a stand-alone server for structural proteome curation. *Database* https://doi.org/10.1093/database/baad059 (2023).
38. Greene, L. H. et al. The CATH domain structure database: new protocols and classification levels give a more comprehensive resource for exploring evolution. *Nucleic Acids Res* **35**, D291–D297 (2007).
39. Zhang, C., Zhang, X., Freddolino, P. L. & Zhang, Y. BioLiP2: an updated structure database for biologically relevant ligand–protein interactions. *Nucleic Acids Res.* https://doi.org/10.1093/nar/gkad630 (2023).
40. van Kempen, M. et al. Fast and accurate protein structure search with Foldseek. *Nat. Biotechnol.* https://doi.org/10.1038/s41587-023-01773-0 (2023).
41. Yang, J. M. & Tung, C. H. Protein structure database search and evolutionary classification. *Nucleic Acids Res.* **34**, 3646–3659 (2006).
42. Li, Z., Jaroszewski, L., Iyer, M., Sedova, M. & Godzik, A. FATCAT 2.0: towards a better understanding of the structural diversity of proteins. *Nucleic Acids Res.* **48**, W60–W64 (2020).
43. Meng, E. C. et al. UCSF ChimeraX: tools for structure building and analysis. *Protein Sci.* **32**, e4792 (2023).
44. Selmer, M., Al-Karadaghi, S., Hirokawa, G., Kaji, A. & Liljas, A. Crystal structure of Thermotoga maritima ribosome recycling factor: a tRNA mimic. *Science* **286**, 2349–2352 (1999).
45. Hanson, R. M., Prilusky, J., Renjian, Z., Nakane, T. & Sussman, J. L. JSmol and the next-generation web-based representation of 3D molecular structure as applied to Proteopedia. *Isr. J. Chem.* **53**, 207–216 (2013).
46. DeLano, W. L. Pymol: an open-source molecular graphics tool. *CCP4 Newsletter Pro. Crystallogr.* **40**, 82–92 (2002).
47. Sayle, R. A. & Milnerwhite, E. J. Rasmol—biomolecular graphics for all. *Trends Biochem. Sci.* **20**, 374–376 (1995).
48. Humphrey, W., Dalke, A. & Schulten, K. VMD: visual molecular dynamics. *J. Mol. Graph.* **14**, 33–38 (1996).
49. Hanson, R. M. Jmol—a paradigm shift in crystallographic visualization. *J. Appl. Crystallogr.* **43**, 1250–1260 (2010).

Acknowledgements
The authors thank X. Wei and Z. Perry for technical assistances to compile US-align for Mac OS. This work used the Advanced Cyberinfrastructure Coordination Ecosystem: Services and Support (ACCESS) program, which is supported by National Science Foundation (2138259, 2138286, 2138307, 2137603 and 2138296). This work is supported in part by the National Institute of Allergy and Infectious Diseases (AI134678 to L.F. and Y.Z.), Ministry of Education (T1 251RES2309 to Y.Z.), and the National University of Singapore startup grants (WBS #A-8001129-00-00, #A-0010130-15-00, #A-8000974-00-00 to Y.Z.). The funders had no role in study design, data collection and analysis, decision to publish or preparation of the paper.

Author contributions
Y.Z. conceived the project. C.Z. developed the program and prepared the server. C.Z., L.F. and Y.Z. drafted the manuscript and approved the final version. All collaborators of this study who fulfilled the criteria for authorship inclusion required by *Nature Portfolio* journals have been included as authors. Roles and responsibilities were agreed among collaborators ahead of the research. Local and regional research relevant to this study is referenced.

Competing interests
The authors declare no competing interests.

Additional information
Supplementary information The online version contains supplementary material available at https://doi.org/10.1038/s41596-025-01189-x.

Correspondence and requests for materials should be addressed to Lydia Freddolino or Yang Zhang.

Peer review information *Nature Protocols* thanks John Dzimianski and the other, anonymous, reviewer(s) for their contribution to the peer review of this work.

nature portfolio

Corresponding author(s): Yang Zhang

Last updated by author(s): Jan 31, 2025

Reporting Summary

Nature Portfolio wishes to improve the reproducibility of the work that we publish. This form provides structure for consistency and transparency in reporting. For further information on Nature Portfolio policies, see our Editorial Policies and the Editorial Policy Checklist.

Statistics

For all statistical analyses, confirm that the following items are present in the figure legend, table legend, main text, or Methods section.

n/a	Confirmed
☒	☐ The exact sample size (*n*) for each experimental group/condition, given as a discrete number and unit of measurement
☒	☐ A statement on whether measurements were taken from distinct samples or whether the same sample was measured repeatedly
☒	☐ The statistical test(s) used AND whether they are one- or two-sided *Only common tests should be described solely by name; describe more complex techniques in the Methods section.*
☒	☐ A description of all covariates tested
☒	☐ A description of any assumptions or corrections, such as tests of normality and adjustment for multiple comparisons
☒	☐ A full description of the statistical parameters including central tendency (e.g. means) or other basic estimates (e.g. regression coefficient) AND variation (e.g. standard deviation) or associated estimates of uncertainty (e.g. confidence intervals)
☒	☐ For null hypothesis testing, the test statistic (e.g. *F*, *t*, *r*) with confidence intervals, effect sizes, degrees of freedom and *P* value noted *Give P values as exact values whenever suitable.*
☒	☐ For Bayesian analysis, information on the choice of priors and Markov chain Monte Carlo settings
☒	☐ For hierarchical and complex designs, identification of the appropriate level for tests and full reporting of outcomes
☒	☐ Estimates of effect sizes (e.g. Cohen's *d*, Pearson's *r*), indicating how they were calculated

Our web collection on statistics for biologists contains articles on many of the points above.

Software and code

Policy information about availability of computer code

Data collection	CURL (version 7.68.0) was used to download example files.
Data analysis	US-align (version 20241108) was used for structure alignment. The US-align web server and source code are available at https://zhanggroup.org/US-align/. PyMOL (version 2.1.0), RasMol (version 2.6.7.0) and UCSF ChimeraX (version 1.8) were used for structure visualization.

For manuscripts utilizing custom algorithms or software that are central to the research but not yet described in published literature, software must be made available to editors and reviewers. We strongly encourage code deposition in a community repository (e.g. GitHub). See the Nature Portfolio guidelines for submitting code & software for further information.

Data

Policy information about availability of data

All manuscripts must include a data availability statement. This statement should provide the following information, where applicable:

- Accession codes, unique identifiers, or web links for publicly available datasets
- A description of any restrictions on data availability
- For clinical datasets or third party data, please ensure that the statement adheres to our policy

PDB files for PDB IDs 101m, 1mba, 4jhm, 4iaj, 1eh1, 1evv, 6jxm, 3am1, 1ajk, and 2ayh are available through https://rcsb.org. The non-redundant PDB library (i.e.,

the I-TASSER template library) is updated on a weekly basis and available at https://zhanggroup.org/library/PDB.tar.bz2. Files to demonstrate the "-TMscore" option of US-align are available at https://zhanggroup.org/TM-score/help.zip.

Research involving human participants, their data, or biological material

Policy information about studies with human participants or human data. See also policy information about sex, gender (identity/presentation), and sexual orientation and race, ethnicity and racism.

Reporting on sex and gender	*Use the terms sex (biological attribute) and gender (shaped by social and cultural circumstances) carefully in order to avoid confusing both terms. Indicate if findings apply to only one sex or gender; describe whether sex and gender were considered in study design; whether sex and/or gender was determined based on self-reporting or assigned and methods used.* *Provide in the source data disaggregated sex and gender data, where this information has been collected, and if consent has been obtained for sharing of individual-level data; provide overall numbers in this Reporting Summary. Please state if this information has not been collected.* *Report sex- and gender-based analyses where performed, justify reasons for lack of sex- and gender-based analysis.*
Reporting on race, ethnicity, or other socially relevant groupings	*Please specify the socially constructed or socially relevant categorization variable(s) used in your manuscript and explain why they were used. Please note that such variables should not be used as proxies for other socially constructed/relevant variables (for example, race or ethnicity should not be used as a proxy for socioeconomic status).* *Provide clear definitions of the relevant terms used, how they were provided (by the participants/respondents, the researchers, or third parties), and the method(s) used to classify people into the different categories (e.g. self-report, census or administrative data, social media data, etc.)* *Please provide details about how you controlled for confounding variables in your analyses.*
Population characteristics	*Describe the covariate-relevant population characteristics of the human research participants (e.g. age, genotypic information, past and current diagnosis and treatment categories). If you filled out the behavioural & social sciences study design questions and have nothing to add here, write "See above."*
Recruitment	*Describe how participants were recruited. Outline any potential self-selection bias or other biases that may be present and how these are likely to impact results.*
Ethics oversight	*Identify the organization(s) that approved the study protocol.*

Note that full information on the approval of the study protocol must also be provided in the manuscript.

Field-specific reporting

Please select the one below that is the best fit for your research. If you are not sure, read the appropriate sections before making your selection.

☒ Life sciences ☐ Behavioural & social sciences ☐ Ecological, evolutionary & environmental sciences

For a reference copy of the document with all sections, see nature.com/documents/nr-reporting-summary-flat.pdf

Life sciences study design

All studies must disclose on these points even when the disclosure is negative.

Sample size	Only the minimal number of example cases necessary for demonstrating the functionality of the US-align programs are included.
Data exclusions	No data were excluded from the analysis.
Replication	Attempts to replicate the study were successful on Microsoft Windows (both natively and on Windows Subsystem for Linux), Linux and Mac OS.
Randomization	Randomization is not relevant to this study.
Blinding	Blinding was not relevant to this study.

Reporting for specific materials, systems and methods

We require information from authors about some types of materials, experimental systems and methods used in many studies. Here, indicate whether each material, system or method listed is relevant to your study. If you are not sure if a list item applies to your research, read the appropriate section before selecting a response.

Materials & experimental systems

n/a	Involved in the study
☒	☐ Antibodies
☒	☐ Eukaryotic cell lines
☒	☐ Palaeontology and archaeology
☒	☐ Animals and other organisms
☒	☐ Clinical data
☒	☐ Dual use research of concern
☒	☐ Plants

Methods

n/a	Involved in the study
☒	☐ ChIP-seq
☒	☐ Flow cytometry
☒	☐ MRI-based neuroimaging

Plants

Seed stocks

Report on the source of all seed stocks or other plant material used. If applicable, state the seed stock centre and catalogue number. If plant specimens were collected from the field, describe the collection location, date and sampling procedures.

Novel plant genotypes

Describe the methods by which all novel plant genotypes were produced. This includes those generated by transgenic approaches, gene editing, chemical/radiation-based mutagenesis and hybridization. For transgenic lines, describe the transformation method, the number of independent lines analyzed and the generation upon which experiments were performed. For gene-edited lines, describe the editor used, the endogenous sequence targeted for editing, the targeting guide RNA sequence (if applicable) and how the editor was applied.

Authentication

Describe any authentication procedures for each seed stock used or novel genotype generated. Describe any experiments used to assess the effect of a mutation and, where applicable, how potential secondary effects (e.g. second site T-DNA insertions, mosiacism, off-target gene editing) were examined.