"Using DALI for protein structure comparison"

For the volume "Structural Bioinformatics" of the lab protocol series Methods in Molecular Biology, published by Springer Nature

Cover page

Liisa Holm

Institute of Biotechnology & Department of Biological Sciences, University of Helsinki, Finland

liisa.holm@helsinki.fi

Running head: Using DALI for protein structure comparison

Abstract

The exponential growth in the number of newly solved protein structures makes correlating and classifying the data an important task. Distance matrix alignment (Dali) is used routinely by crystallographers world-wide to screen the database of known structures for similarity to newly-determined structures. Dali is easily accessible through the web server (http://ekhidna.biocenter.helsinki.fi/dali). Alternatively, the program may be downloaded and pairwise comparisons performed locally on Linux computers.

Key Words

classification of protein folds; database searching; distance geometry; pattern recognition; protein structure alignment

Introduction

At the end of 2018, the Protein Data Bank (PDB) contained the structure of 300,000 protein chains. Nearly all proteins have structural similarities to other proteins. General similarities arise from principles of physics and chemistry that limit the number of ways in which a polypeptide chain can fold into a compact globule. Evolutionary relationships result in surprising similarities, which are even stronger than similarity due to convergence caused by physical principles. Comparing 3D structures may reveal biologically interesting similarities that are not detectable by comparing only sequences and may help to infer functional properties of hypothetical proteins. For example, the recent discovery of structural homology and a conserved Cys-Asp-His catalytic triad unified two previously uncharacterized effectors from *Legionella pneumophila* with the cycle inhibiting factors (cif) gene family, leading to mechanistic insights of host manipulation by this pathogenic bacterium [1].

Large proteins can be decomposed into semi-autonomous, globular folding units called domains. Domains are often evolutionarily mobile modules and may carry specific biological functions. Because a common

domain may be surrounded by completely unrelated domains, most structure comparison methods search for local similarities. A structural alignment defines a set of one-to-one correspondences between Cα atoms in two proteins. This is analogous to sequence alignment, except that the notion of similarity is much more complex between three-dimensional objects than between linear sequences. A large variety of scoring functions for structural similarity have been proposed [2]. The most important categories are (i) scoring functions based on the root mean square deviation (RMSD) of rigid-body superimposition and (ii) scoring functions allowing flexible superimposition or plastic deformations. Early works based on visual analysis of folds stressed the importance of plastic deformations in the evolution of protein structure. Dali's scoring function belongs to the latter category, and it has been shown to yield structural dendrograms that agree well with expert classifications [3-5].

The Dali method is based on a sensitive measure of geometrical similarities defined as a weighted sum of similarities of intramolecular distances [3]. Let's consider two proteins labeled A and B. The match of two substructures is evaluated using an additive similarity score *S* of the form:

Equation 1 $$S(A,B) = \sum_{i \in core} \sum_{j \in core} \varphi(i,j)$$

where *i* and *j* label residues, *core* is the common substructure, and φ is a similarity measure based on some pairwise relationship, here on the similarity of intramolecular Cα-Cα distances. Unmatched residues do not contribute to the overall score. For a given functional form of φ(i,j), the largest value of S corresponds to the optimal set of residue equivalences. The similarity measure needs to balance two contradictory requirements: maximizing the number of equivalenced residues and minimizing structural deviations. The use of relative rather than absolute deviations of equivalent distances is tolerant to the cumulative effect of gradual geometrical distortions. In Dali, the residue-pair score φ has the form:

Equation 2 $$\varphi(i,j) = (\theta - diff(i,j)) * env(d_{ij}^{*})$$

where the first term of the multiplication is the relative distance difference compared to a similarity threshold θ and the second term is an envelope function which downweights pairs in the long distance range. In Dali, the similarity threshold is set to θ = 0.2. The envelope is a Gaussian function $env(x) = e^{-(\frac{x}{R_0})^2}$ where R_0 = 20 Å, calibrated on the size of a typical domain. The relative distance difference $diff(i,j) = \frac{|d_{ij}^{A} - d_{ij}^{B}|}{d_{ij}^{*}}$, where d_{ij}^{A} and d_{ij}^{B} are intramolecular Cα-Cα distances in structure A and B, respectively, and their average is $d_{ij}^{*} = \frac{d_{ij}^{A} + d_{ij}^{B}}{2}$. Inserting the values of the constants, the resulting raw Dali score describing the structural similarity is given by:

Equation 3 $$S(A,B) = \sum_{i \in core} \sum_{j \in core} (0.2 - \frac{|d_{ij}^{A} - d_{ij}^{B}|}{d_{ij}^{*}})\, e^{-(\frac{d_{ij}^{*}}{20\,A})^2}$$,

The residue-pair scores (Figure 1) can get both positive and negative values, therefore the maximum of Equation 3 corresponds to a local alignment. Hydrogen bonded backbone segments in helices and sheets have a distance around 5 A. Here absolute distance deviations up to 1 A generate positive scores, while larger deviations incur a steeply increasing penalty. At 10 A distance, as found between helices or sheets in tertiary contact, a deviation up to 2 A still contributes positively, and larger deviations incur a mild penalty. The diameter of a typical domain is around 20 A. Beyond this distance, the score function is relatively insensitive to distance deviations. For example, two conformations of a two-domain structure, which are not superimposable as rigid bodies because of hinge rotation, can be structurally aligned by Dali since the similarity of local structure compensates for the downweighted deviations in interdomain distances.

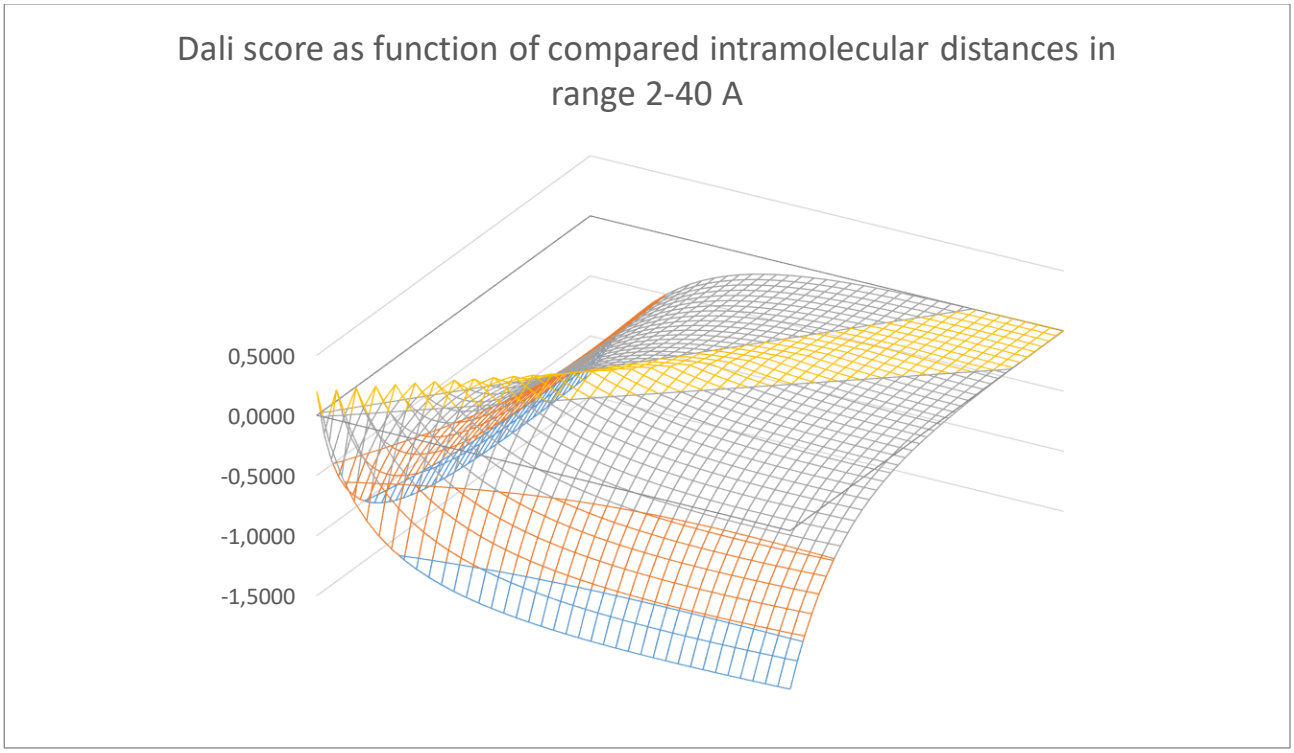

Figure 1. Pairwise distances contribute a positive score (yellow color) when the relative deviation is less than 20 %. An envelope function damps the contribution of longer distances.

For random pairwise comparison the expected Dali-score (Equation 3) increases with the number of residues in the compared proteins. In order to describe the statistical significance of a pairwise comparison score *S(A,B),* we use the Z-score defined as

Equation 4 $$Z(A,B) = \frac{S(A,B) - m(L)}{\sigma(L)}$$

The relation between the mean score m, standard deviation σ and the average length $L = \sqrt{L_A L_B}$ of two proteins was derived empirically from a large set of random pairs of structures. Fitting a polynomial gave the approximation:

Equation 5 $$m(L) \approx 7.95 + 0.71L - 2.59 \cdot 10^{-4} L^2 - 1.92 \cdot 10^{-6} L^3, \text{ if } L \leq 400$$

$$m(L) = m(400) + L - 400, if\ L > 400$$

For standard deviation, the empirical estimate was $\sigma(L) = 0.5 * m(L)$. The Z-score is computed for every possible pair of domains, and the highest value is reported as the Z-score of the protein pair [6]. Possible domains are determined by the Puu algorithm (Parser for protein Unfolding Units), which recursively cuts a structure into smaller compact substructures at the weakest interface [7].

Materials

The Dali method is available as a web service at http://ekhidna.biocenter.helsinki.fi/dali. The standalone version can be downloaded from http://ekhidna.biocenter.helsinki.fi/dali/README.html, which gives instructions for installation. The package contains two Perl wrapper scripts along with installation instructions in the README file as well as source code and sample input/output files. The program is designed to run under Linux. Compiling the source code requires Fortran-90 (e.g. gfortran) and C compilers. openmpi is optional. Standard Perl is required to execute the wrapper scripts. To install the programs, unpack the zip archive in a suitable directory, edit the path to the Dali home directory in the Makefile and follow the instructions.

The installed package contains two Perl scripts:

1. A script **import.pl** which must be used to convert PDB files to Dali's internal data format. This script handles the input PDB files which might contain multiple chains, passes them to the DSSP program for extracting the coordinates and defining secondary structure elements, reads the output of DSSP, prepares a hierarchical tree of folding units, and outputs a data file for each chain in the input PDB file (see Note 1).
2. The script **dali.pl** performs pairwise comparisons of a list of query structures to a list of target structures. The lists of query and target structures must be provided by the user (see Note 2).

Methods

3.1 Input file

The input structure must be a PDB format text file. The PDB format consist of records (lines) where the first six characters are a keyword and data follows in fixed-width columns. Dali uses data from the COMPND and ATOM records. Only the first model of an NMR ensemble is read in. The input structure must have

complete backbone atoms (C, CA, N, O), this requirement comes from the DSSP program used to parse PDB files. Though only CA coordinates are used in structural alignment, the DSSP step is necessary because also secondary structure assignments by DSSP are used as input to structure comparison. Chains shorter than 29 amino acids are ignored (see Note 3). The maximum throughput of the web server is 100 - 200 structure database searches per day. To apply the method on a larger number of structures, we advise the use of the standalone version.

3.2 Structure data parsing

The DSSP method [8] is used to parse Cα coordinates and to define secondary structure elements from the PDB file. The dsspCMBI implementation of DSSP is included in the standalone package. dsspCMBI is maintained at ftp://ftp.cmbi.ru.nl/pub/software/dssp/. The DSSP algorithm defines hydrogen bonds based on the dipole interaction of backbone amide and carbonyl groups. The interaction energy is modelled by a Coulomb potential between partial charges, which leads to a function of the angle and distance of the dipoles. Regular patterns of hydrogen bonds between runs of residues generate turns, helices, bridges, ladders and sheets. Dali uses secondary structure elements (helices, beta strands) to simplify structural alignment. Alignment is further simplified by using a tree of compact substructures to guide alignment identifying first local matches and then solving a combinatorial problem in building up larger clusters of matching substructures. The tree is generated by the Puu program [7]. The underlying physical concept is maximal interactions within each unit and minimal interaction between units (domains). In a simple harmonic approximation, interdomain dynamics is determined by the strength of the interface and the distribution of masses. The most likely domain decomposition involves units with the most correlated motion, or largest interdomain fluctuation time. The decomposition of a convoluted 3-D structure is complicated by the possibility that the chain can cross over several times between units. Grouping the residues by solving an eigenvalue problem for the contact matrix reduces the problem to a one-dimensional search for all reasonable trial bisections. Recursive bisection yields a tree of putative folding units. Simple physical criteria are used to identify units that could exist by themselves.

3.3 Pairwise comparison

Dali implements four structure alignment algorithms. In the standalone package these are available through the serialcompare / mpicompare programs, though in practice they are invoked through the wrapper script dali.pl.

1. The Soap algorithm [9] is used to align structures with few (see Note 4) secondary structure elements. Soap minimizes a “soap film” metric between two Cα traces superimposed in 3D space. The minimal surface area between the virtual backbones of two proteins is determined numerically using an iterative triangulation strategy. The first protein is then rotated and translated in space until the

smallest minimal surface is obtained. Such a technique yields the optimal structural superposition between two protein segments.

2. The Wolf algorithm is a very fast filter to identify obvious similarities [10]. It models secondary structure elements as vectors. Three points taken from an ordered pair of secondary structure elements (SSE) defines an internal coordinate frame. Here, the midpoint of the first SSE is the origin, the vector representing the first SSE aligns with the y axis and the midpoint of the second SSE is in the positive z-y half-plane. Each database structure is presented in the "poses" defined by all possible frames of SSE pairs. Testing all frames of the query structure allows counting the number of matching SSEs at nearby positions in all possible "poses" by a fast lookup procedure (see Note 5). The result is a ranked list of database structures which can be used as a filter in database search.

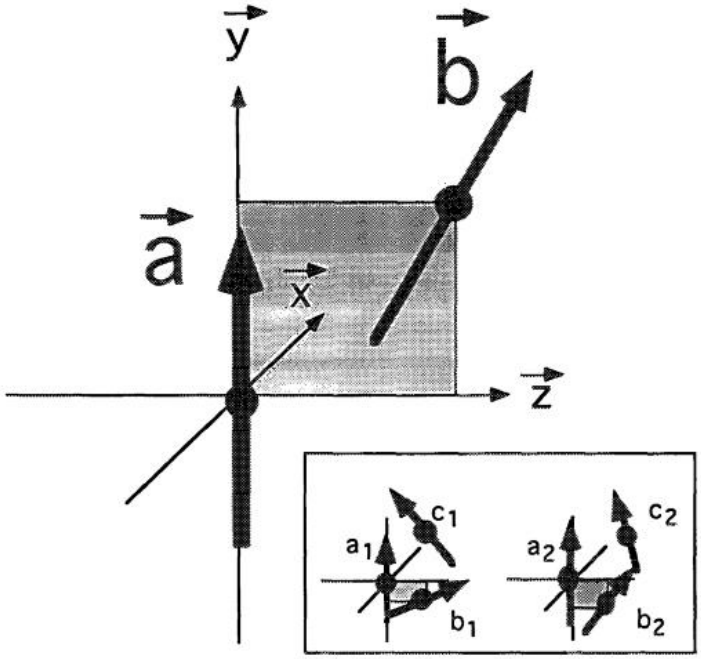

Figure 1: Coordinate system.
Protein structure is described as a set of *vectors* representing secondary structure elements (SSEs). An ordered pair of SSEs (*a* and *b*) defines a right-handed three-dimensional coordinate frame such that the midpoint of *a* is at the origin, the axis of *a* is along the positive y axis and the midpoint of *b* lies in the z-positive yz halfplane. It is required that the midpoint of *b* is not along the axis of *a*. (In practice, the singularity does not happen within machine precision.) The **inset** shows a comparison of the internal coordinate frames of two proteins (labelled 1 and 2): at the origin, the unit vectors $\hat{a}_1 \approx \hat{a}_2$ are the same by definition and in this case the other SSEs match approximately in their directions ($\hat{b}_1 \approx \hat{b}_2$ and $\hat{c}_1 \approx \hat{c}_2$) and in the position of the segment midpoints (filled circles) relative to the origin.

Figure 2. Coordinate system of Wolf method. EDITOR TO REDRAW!

3. Parsi is a sensitive branch-and-bound alignment algorithm [11]. The algorithm is guaranteed to deliver an exact solution to the subproblem of ungapped alignment of secondary structure elements (SSEs),

ignoring loops. Dali is based on a sum of pairs score. The score of an alignment involving *n* segments has *n* diagonal terms and *n(n-1)* off-diagonal terms in the summation. The off-diagonal dependencies pose a difficult combinatorial problem. The branch-and-bound algorithm overcomes the difficulty by initially pooling all possible segment-to-segment pairs as potential constituents of the optimal alignment. An upper bound of the total alignment score is given by the sum of the maxima of each of the n^2 terms independently of the others. More formally, the problem of optimizing the alignment score (Equation 1) over all possible alignments A→B can be rewritten using an indicator function $\mathbf{1}_{A\to B}$ as

Equation 6 $$S^*(A,B) = \max_{A\to B} \sum_{x=1}^{m} \sum_{y=1}^{m} \sum_{x'=1}^{n} \sum_{y'=1}^{n} \varphi(x \to x', y \to y) * \mathbf{1}_{A\to B}(x \to x') * \mathbf{1}_{A\to B}(y \to y')$$

where $$\mathbf{1}_X(a) = \begin{cases} 1, & if\ a\ is\ a\ member\ of\ set\ X \\ 0, & if\ a\ is\ not\ a\ member\ of\ set\ X \end{cases}$$

Here, the indicator function picks one-to-one correspondences defined by a given alignment A→B while all other terms are zero. The maximum is searched over all possible alignments of m residues in structure A and n residues in structure B, which is a hard combinatorial problem. A partition is a subset of the search space, which can contain many-to-many correspondences between residues in structure A and residues in structure A. An upper bound on the best one-to-one alignment score that is contained within a partition is given by

Equation 7 $$S^*(partition) \leq \sum_{x=1}^{m} \sum_{y=1}^{m} \max_{\substack{X'=1..n \\ y'=1..n}} (\varphi(x \to x', y \to y') * \mathbf{1}_{partition}(x \to x') * \mathbf{1}_{partition}(y \to y')).$$

The search space is recursively partitioned to derive tighter upper bounds for the subspaces. A binary partition moves one particular query-target segment pairing to one subspace and excludes it from the other. The algorithm terminates when the partition that has the highest upper bound corresponds to unique pairings of all segments.

4. All alignments generated by methods 1-3 above use different objective functions that only approximate the Dali score or exclude loops from the alignment. All alignments generated by methods 1-3 are therefore refined using a Monte Carlo algorithm (Dalicon) that aims to maximize the Dali score over the whole structures [3].

Interestingly, Dali has been shown to generate close to optimal solutions on a benchmark of small proteins [12].

3.4 Web server methods

The web server and standalone version use the same algorithms for structure comparison. However, the web server has search and data visualization options which are not included in the standalone package. The web server supports four types of comparison:

a. search query structure against the Protein Data Bank using heuristics and a knowledge base of pre-computed pairwise structure similarities
b. compare query structure against a representative subset of the Protein Data Bank using systematic pairwise comparison
c. perform pairwise comparison of a query structure against a set of target structures
d. perform all against all comparison of up to 64 structures

All methods are based on pairwise comparison. Methods b-d are available in the standalone version.

The search (method a) heuristically prunes the list of targets so that dissimilar target structures can be eliminated without explicit computation [13]. The elimination relies on a knowledge base of accumulated pairwise comparisons of structures in the PDB, which are represented as a graph. The nodes of the graph represent protein structures and edges represent structural alignments. The idea is that once a strong similarity to the query structure has been found, other structural neighbours can be collected by walks through the graph, provided that structurally similar proteins form a connected component in the graph. A cascade of comparison methods is used to try and find a strong similarity from the query structure to known structures with little computational effort. The cascade starts with sequence comparison followed by Wolf or Soap. When a strong similarity is found, the search switches to "walking" based on transitive alignments. If no strong similarity was found, the query structure is compared against a representative subset of PDB using Parsi. Finally, a sequence search of the structurally most similar targets identifies homologs not caught by the previous steps. The Z-score threshold for extending the walk is adjusted dynamically during the search. Edges with lower Z-score than the threshold are effectively removed from the structural similarity graph. There are only empirical rules for setting the threshold. Initially, it is set to the square root of the Z-score for the comparison of the query structure to itself. Subsequently, it is increased if there are many higher scoring targets. Specifically, the aim is to report the 100 highest scoring PDB90 representatives (PDB structures with less than 90% sequence identity to each other). Because small domains obtain smaller Z-scores than large domains, we recommend cutting multidomain structures and searching each domain separately.

3.5 Interpretation of the result

Like in sequence analysis, the goal of structural database searching is usually to identify homologous proteins which might provide clues to the function of the query protein. Homology means descent from a

common ancestor. We can infer homology from sequence or structural similarities that are so strong they would not be expected to have arisen by chance. The boundary between homologous and unrelated proteins varies from one family to another and there is no universally applicable Z-score cutoff to separate homologous from analogous (non-homologous) structures. As a rule of thumb, a Z-score above 20 means the two structures are definitely homologous, between 8 and 20 means the two are probably homologous, between 2 and 8 is a grey area, and a Z-Score below 2 is not significant. The wide grey zone is because the size of the proteins influences Z-scores - small structures will tend to have small Z-Scores, whereas a medium Z-score for very large structures need not imply a biologically interesting relationship. Fold type also has an effect – α/β proteins also usually have higher Z-scores than all-β proteins. For example, all ($\beta\alpha$)8-barrel folds are unified at Z-scores above 10. In contrast, a small avian polypeptide (PDB code 1ppt) contains only one helix and a proline-rich loop and gets a Z-score under 8 even in comparison to itself. In view of the Z-score, it is much more improbable to observe sixteen helices and strands arranged in a similar fold than to find a similar arrangement of just a helix and a loop.

Other criteria than the mere Z-score are often required to make a convincing case for homology. Structural dendrograms are useful in locating the boundary between homologous and analogous folds, the idea being that homologous proteins should be monophyletic and functionally similar [4]. Dali generates structural dendrograms from the matrix of pairwise Z-scores by average linkage clustering. Branch lengths in the dendrogram represent distances, which are modelled *ad hoc* as the difference of Z-scores.

Dali web server results (Figure 3) are linked to interactive sequence search and function assignment servers [14-15]. The structural alignments can be visualized as stacked sequence logos, where the logos are generated from sequence neighbors of the target protein and the alignment of the logos is based on the structure comparison. In particular enzyme super-families have sharp sequence signatures but binding domains can have very little sequence similarity. Without a sequence signature, it is harder to establish homology. Web server results additionally have a link labelled PDB for each target structure, which returns the coordinates superimposed onto the query structure for viewing in a molecular graphics program of your choice.

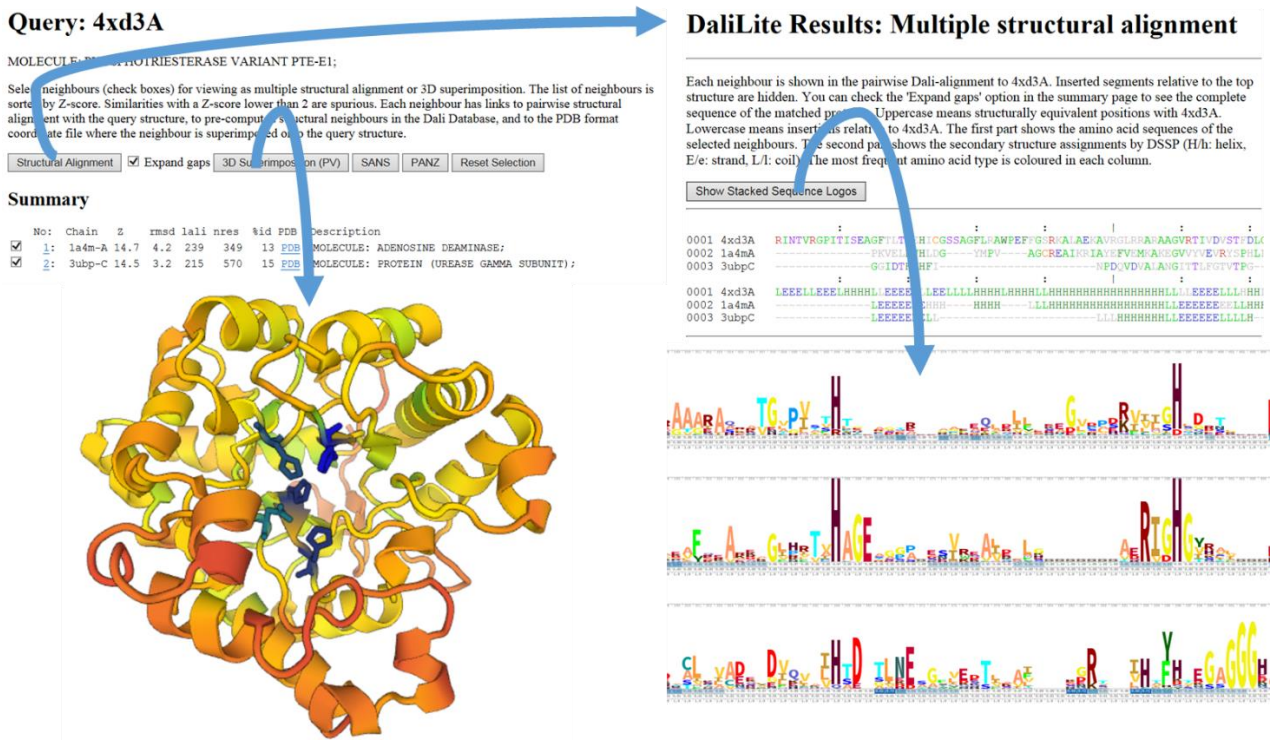

Figure 3. Outputs of the Dali web server, clockwise from top left: summary of similar structures ordered by Z-score, structural alignment showing amino acid sequences and secondary structure assigments, stacked sequence logos highlighting conserved structurally equivalent positions, 3-D structure view (here coloured by sequence conservation).

Besides the Z-score, Dali reports the RMSD and the number of equivalent residues (LALI), because they are traditional measures and often quoted as qualifiers of structural similarity. RMSD is a measure of the average deviation in distance between aligned Cα atoms in 3-D superimposition. For sequences sharing 50% identity, this should be around 1.0. Dali maximizes a geometrical similarity score, which is defined in terms of similarities of intra-molecular distances and is thus not primarily aiming to generate alignments with low RMSD. Numerous programs for structure comparison have been published over the last 30 years, based on a variety of similarity measures [2]. Consequently, method evaluations often assess the quality of structural alignments using a non-native yardstick, such as the popular RMSD measure. An alignment is 'better' if it has both smaller RMSD and larger LALI. If both RMSD and LALI are smaller or both are larger, it is not possible to establish an order between the alignments.

3.6 Dali comparison with the locally installed standalone version

The standalone version can be used for a PDB format file with one or multiple protein chains (identified by a letter in column 19). Compressed files with extension .gz and normal text files are accepted (see Note 6). There is generally no reason to change parameters from their default values which are hardcoded in the

standalone version. In a preparatory step, protein structure data must be imported by invoking the wrapper script import.pl. Normally, the script can be invoked as

```
perl –I $DALI_HOME/bin $DALI_HOME/bin/import.pl pdb1xg8ent.gz 1xg8 $DALI_DAT/
```

where environment variables $DALI_HOME and $DALI_DAT point to the installation directory and data directory, respectively. The first argument is the PDB file name and the second argument is the four-letter identifier to be used for the structure. The output is a data file for each chain the PDB file. Thus, the number of output files corresponds to the number of chains longer than 28 residues (see Note 3). The example above creates a file named 1xg8A.dat, where the chain identifier has been automatically appended to the four-letter structure identifier. The first line of a properly formed data file looks like Figure 4. If reading the coordinates failed, for any reason, you only find lots of zeros on the first line of the data file. Troubleshooting tips include installing the DSSP program from the distribution package and checking the format of the PDB entry (Figure 5).

```
>>>> 1xg8A  108    7    3    4  EHEHEEH
     |   |    |    |    |    |  └------  sequential order secondary structure elements
     |   |    |    |    |    └---------  number of beta-strands (E)
     |   |    |    |    └--------------  number of helices (H)
     |   |    |    └-------------------  number of secondary structure elements
     |   |    └------------------------  number of residues
     |   └-------------------------------  chain identifier (one character, appended automatically)
     └-----------------------------------  PDB identifier (four characters)
```

Figure 4. First line of a data file.

```
ATOM      1  N   GLY A   1       2.296  -9.636  18.253  1.00  0.00           N
ATOM      2  CA  GLY A   1       1.470  -9.017  17.255  1.00  0.00           C
ATOM      3  C   GLY A   1       0.448  -9.983  16.703  1.00  0.00           C
ATOM      4  O   GLY A   1       0.208 -11.066  17.345  1.00  0.00           O
```

Figure 5. Example of coordinate data in PDB format. The format has its roots in the era of punch cards when data fields had a fixed width and position.

Pairwise structure comparison is performed by invoking the wrapper script dali.pl as

```
perl –I $DALI_HOME/bin $DALI_HOME/bin/dali.pl –query query.list –db target.list –dat1 $DALI_DAT –dat2
$DALI_DAT
```

where query.list and target.list are text files containing one five-letter structure identifier per line. All query and target structures must have been imported beforehand using the script import.pl (see above). The output is a text file xxxxY.txt for each structure xxxxY in query.list (Y is the chain identifier, xxxx is the PDB entry identifier). The output is sorted in decreasing order of the Z-score (see Note 7). The results are presented in three blocks (Figure 6). The summary block lists statistical parameters for the matched target structures. The alignment block lists the aligned segments. The last block gives the translation-rotation matrices for rigid-body superimposition of target structures onto the query structure. UY+T is the best

approximation of X, where X and Y are the (x,y,z) coordinates of the query and target structure, respectively, U is the rotation matrix and T is the translation vector [16].

All-against-all structure comparison of a set of structures is invoked with the –matrix option as

perl –I $DALI_HOME/bin $DALI_HOME/bin/dali.pl –query query.list –matrix –dat1 $DALI_DAT

This option generates additional outputs named 'ordered' and 'newick_unrooted', which contain a matrix of pairwise Z-scores and a dendrogram in Newick format. Note that the matrix of Z-scores represents similarities between structures, whereas branch lengths of the Newick tree represent distances. Branch lengths are modelled *ad hoc* as the difference of Z-scores.

The wrapper scripts generate a number of intermediate results in the current work directory. The lock file dali.lock is created at the start of the job and is deleted automatically, when it completes successfully. If a file named dali.lock is present, you cannot start another Dali job in the same directory.

```
# Job: test
# Query: 1ppta
# No:  Chain   Z    rmsd lali nres  %id PDB  Description
   1:  1bba-A  3.6  1.8   33    36   39   MOLECULE: BOVINE PANCREATIC POLYPEPTIDE;
         |     |    |     |     |    └-  percent identity of aligned amino acids
         |     |    |     |     └------  number of residues in target structure
         |     |    |     └------------  number of aligned C-alpha atoms
         |     |    └------------------  root-mean-square deviation of aligned C-alpha atoms
         |     └-----------------------  Z-score
         └-------------------------------  target structure

# Structural equivalences
   1: 1ppt-A 1bba-A     1 -  33 <=>    1 -  33   (GLY    1  - ARG   33  <=> ALA    1  - ARG   33 )
      query  target   \_____________________/    \_____________________________________________/
                       query   <=>   target                 query   <=>    target
                       sequential numbering                 PDB residue numbering

# Translation-rotation matrices
-matrix  "1ppt-A 1bba-A  U(1,.)   0.631906 -0.761372 -0.144939          -0.890845"
-matrix  "1ppt-A 1bba-A  U(2,.)   0.512616  0.550832 -0.658642         -10.882093"
-matrix  "1ppt-A 1bba-A  U(3,.)   0.581308  0.341902  0.738366           4.946664"
                                  \_____________________________/         \________/
                                          rotation matrix U           translation vector T
```

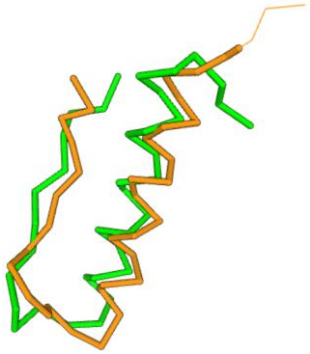

Figure 6. (top) Example output from pairwise comparison. Comments have been added in *italics*. (bottom) Cα traces of 1pptA (green) and 1bbaA (orange) in structural superimposition. Unaligned residues are shown by a thin line.

Notes

1. Dali handles each chain separately. Structure identifiers have a fixed length of five characters, where the last character is the chain identifier. Quaternary structure comparisons are not possible at present.
2. The dali.pl script has two parameters for data directories (DALIDATDIR_1 and DALIDATDIR_2). All query structures must be imported to DALIDATDIR_1. All target structures must be imported to DALIDATDIR_2. DALIDATDIR_1 and DALIDATDIR_2 can be identical, but usually DALIDATDIR_2 contains public structures imported from the Protein Data Bank (PDB) and DALIDATDIR_1 contains private structures.
3. The parameter $MINLEN in the Perl module mpidali.pm is set to 29 by default. The insulin peptide is accepted, shorter chains are rejected.
4. The parameter $MINSSE in the Perl module mpidali.pm is set to 3 by default. This means that structures with two or fewer SSEs are compared using the Soap method and structures with three or more SSEs are compared using Wolf or Parsi.
5. The Wolf algorithm uses three parameters. There is generally no reason to change the defaults. The parameters rcut and maxiter control the iterative refinement of superimposition. The pairing of C-alpha atoms from the query and target structure gets a positive score is their positional deviation is smaller than rcut, which is 4 A by default. Maxiter limits iterations to 10. The parameter neiborcutoff says that internal coordinate frames are generated using every pair of SSE vectors whose midpoints are closer than 12 A.
6. Structures of the Protein Data Bank can be mirrored using the command
   rsync –rlpt –v –z –delete –port=33444 rsync.wwpdb.org::ftp/ data/structures/divided/pdb/ $MIRRORDIR
   where environment variable $MIRRORDIR is the top level of the local structure data directory.
7. The amount of output from structure comparison is limited by the parameter $zcut in the Perl module mpidali.pm. $zcut is the minimum Z-score (default 2.0).

References

1. Valleau D, Quaile AT, Cui H, Xu X, Evdokima E, Chang C, Cuff ME, Urbanus ML, Houliston S, Arrowsmith CH, Ensminger AW, Savchenko A (2018) Discovery of Ubiquitin Deamidases in the Pathogenic Arsenal of *Legionella pneumophila*. Cell reports 23, 568-583.

2. Hasegawa H, Holm L (2009) Advances and pitfalls of protein structural alignment. Curr. Opin. Struct. Biol. 19, 381–389.

3. Holm L, Sander C (1993) Protein structure comparison by alignment of distance matrices. J. Mol. Biol. 233, 123-138.
4. Dietmann S, Holm L (2001) Identification of homology in protein structure classification. Nat. Struct. Biol. 8, 953–957.
5. Fox NK, Brenner SE, Chandonia JM (2014) SCOPE: Structural Classification of Proteins—extended, integrating SCOP and ASTRAL data and classification of new structures. Nucleic Acids Res. 42, D304–D309.
6. Holm L, Sander C (1998) Dictionary of recurrent domains in protein structures. Proteins 33, 88–96.

7. Holm L, Sander C (1994) Parser for protein folding units. Proteins 19, 256-268
8. Kabsch W, Sander C (1983) Dictionary of protein secondary structure: Pattern recognition of hydrogen bonded and geometrical features. Biopolymers 22, 2577-2637.
9. Falicov A, Cohen FE (1996) A surface of minimum area metric for the structural comparison of proteins. J. Mol. Biol. 258, 871-892.
10. Holm L, Sander C (1995) Fast protein structure database searches at 90 % reliability. ISMB 3, 179-187
11. Holm L, Sander C (1996) Mapping the protein universe. Science 273, 595-602.
12. Wohlers, i., Andonov, R., Klau, G.W. (2013) DALIX: optimal DALI protein structure alignment. IEEE/ACM Trans Comput Biol Bioinform 10, 26-36.

13. Holm L, Kääriäinen S, Rosenström P, Schenkel A (2008) Searching protein structure databases with DaliLite v.3. Bioinformatics 24, 2780-2781.

14. Somervuo P, Holm L (2015) SANSparallel: interactive homology search against Uniprot. Nucl. Acids Res. 43, W24-W29.
15. Petri Törönen, Alan Medlar, Liisa Holm (2018) PANNZER2: A rapid functional annotation webserver. Nucl. Acids Res. 46, W84-W88.
16. Kabsch, Wolfgang (1978). "A discussion of the solution for the best rotation to relate two sets of vectors". Acta Crystallographica. A34: 827–828