Quick start

```
# set ISShome to directory where ISS_ProtSci was installed
ISShome=~/ISS_ProtSci-1
# run transitive closure search of AFDB2
$ISShome/1_search.csh -pdbfile <PDBFILE> -cd1 testA [ -minlali <integer> ] [ -zcut <float> ]
# annotate search results
$ISShome/2_annotate.csh -pdbfile <PDBFILE> -cd1 testA
# estimate recall using Pfam clan or family as positive controls
$ISShome/3_analyze.csh -clan <PFddddd|CLdddd> -cd1 testA [ -minlali <integer> ] [ -zcut <float> ]
```

Case study of a unique fold protein

Welcome to the ISS_ProtSci tutorial!

ISS_ProtSci (Liu et al., 2025) is a standalone tool designed for efficient structure-based searches of the AlphaFold Database version 2 (AFDB2). Its main goal is to comprehensively identify all instances of a fold represented by a given query structure. Key features include:

- **Accurate validation:** All hits are rigorously validated using DALI, effectively filtering out false positives that might be suggested by Foldseek.
- **High sensitivity:** By iteratively using new hits as anchors in successive search rounds, ISS_ProtSci surpasses Foldseek in detecting remote homologs while still maintaining practical runtimes.
- **Rich annotation:** Results are automatically annotated with Uniprot, Pfam, and alignment metadata, streamlining downstream analysis.

In this tutorial, we will walk through basic and advanced usage examples to help you get the most out of ISS_ProtSci. Pre-computed results can be downloaded from http://ekhidna2.biocenter.helsinki.fi/ISS_ProtSci/tut.tar.gz

Preparing your query

Create and navigate to your work directory. Results will be written to the work directory. We also set an environment variable `ISShome` which points to the directory where ISS_ProtSci was installed

```
mkdir -p tut
cd tut
```

```
export ISShome=~/ISS_ProtSci-1/
```

ISS_ProtSci requires a PDB formatted coordinate file as input. Experimental structures can be downloaded from the Protein Data Bank. Predicted models can be retrieved from AlphaFold Database. Here, we download the query structure (3H0N.pdb.gz) from the Protein Data Bank.

```
wget https://files.rcsb.org/download/3H0N.pdb.gz
```

It is always a good idea to first visualize the protein that you are working with. The query structure has two domains (Figure 1). We create separate queries for either domain.

```
zcat 3H0N.pdb.gz | egrep '^HETAT|^ATO' | gawk ' $6 < 139 ' > adom.pdb
zcat 3H0N.pdb.gz | egrep '^HETAT|^ATO' | grep -v HOH | gawk ' $6 > 138 && $6 & <= 184' > zinc.pdb
```

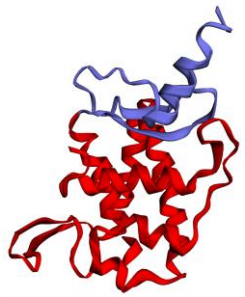

Figure 1: The query protein has two domains: an ABATE domain (residues 1-138, red) and a zinc finger domain (residues 139-184, blue). Rendered with EzMol.

Background information on the query protein

PDB entry 3h0n represents the crystal structure of one member of the ABATE domain family. The protein consists of a two-domain organisation, with the N-terminal domain presenting a new fold called the ABATE domain that may bind an as yet unknown ligand (PMID:20944211). Based on sequence comparisons, members of the ABATE domain family (PF07336) are found in *Actinomycetota* (*Streptomyces, Rhizobium, Ralstonia, Agrobacterium* and *Bradyrhizobium* species. The C-terminal domain forms a treble-clef zinc-finger that is likely to be involved in DNA binding, suggesting a role as stress-induced transcriptional regulator.

Pruning the query structure for effective search

ISS_ProtSci, using Dali, performs local structural alignments between the query structure and proteins in the target database (AFDB2). To ensure unambiguous results, it is beneficial to prune the query structure down to a single domain of interest. Note that while the query should ideally represent a single domain, the target proteins in AFDB2 may contain multiple domains, which are handled transparently to the user. By focusing on a single domain, the search process becomes more precise, improving the interpretability of the results. A key reason for pruning the query structure is to avoid ambiguity. When the query consists of a single domain, all database matches map to the same structural unit, ensuring consistency and simplifying both analysis and classification.

Search depth is determined by query coverage, making it an important consideration. Single-domain queries naturally allow for higher query coverage, leading to greater specificity and greater efficiency. If the query coverage threshold has been set too low, reaching transitive closure can take a very long time.

We recommend visual inspection and a text editor to extract the coordinates of the domain of interest from a PDB file.

Special considerations for AlphaFold models

Pruning is particularly important when using AlphaFold models as query structures. Although AlphaFold models are highly accurate in regions with high confidence scores, they often contain segments modelled with low confidence. These unreliable regions frequently appear as wide arcs in three-dimensional space, leading to spurious structural alignments that do not reflect true evolutionary or functional relationships. If low-confidence regions contribute to the alignment, they can artificially inflate alignment lengths, misguiding the search and reducing efficiency. Unreliable regions can be removed by “cropping” residues with low pLDDT values (e.g. pLDDT<60). pLDDT values are stored in the B-factors, written in the last floating point column of the PDB format. The example below shows the coordinates of an atom from an AlphaFold model with pLDDT 39.84:

```
ATOM      2  CA  MET A   1      24.941 -21.539  51.646  1.00 39.84           C
```

“Cropping” can be done manually using a text editor to remove lines from the PDB file.

Running a basic search with default parameters

ISS_ProtSci requires a PDB file and a run identifier as inputs. The run identifier is a prefix to all output files. Old output files are not overwritten, so either remove the old file or create a new, unique run identifier. The run identifier consists of four letters and the chain identifier from the PDB file.

Run a basic search with the full query protein and its constituent domains:

```
# full structure
$ISShome/1_search.csh -pdbfile 3H0N.pdb.gz -cd1 3h0nA
$ISShome/2_annotate.csh -pdbfile 3H0N.pdb.gz -cd1 3h0nA

# separate ABATE domain
$ISShome/1_search.csh -pdbfile adom.pdb -cd1 adomA
$ISShome/2_annotate.csh -pdbfile adom.pdb -cd1 adomA

# separate zinc finger domain
$ISShome/1_search.csh -pdbfile zinc.pdb -cd1 zincA
$ISShome/2_annotate.csh -pdbfile zinc.pdb -cd1 zincA
```

The search script echoes the default parameter values. The minimum Z-score is fixed at 2.0, since DALI does not report hits below this threshold. The default core size parameter is based on the assumption that the structural core consists of secondary structure elements and is set to the total number of residues assigned to α-helices or β-strands by the DSSP program. The transitive closure search expands to neighbours of structures that meet these criteria and terminates at those that do not. The minimum structural alignment length is 115 for 3h0nA, 91 for adomA, and 19 for zincA.

The search and annotation steps use separate scripts, because the annotation script processes files named like `3h0nA.AFDB2.dali.tsv`, which contain DALI outputs that may also be generated by protocols other than transitive closure. The main annotated result files are named like `3h0nA.AFDB2.pf.tsv`, including metadata such as Uniprot accession numbers, NCBI taxonomy identifiers, amino acid sequences, secondary structure assignments, and best Pfam match to the structurally aligned segment. The annotation step also runs Foldseek in a sensitive mode (--maxseqs 50000) for comparison with transitive closure, creating files named like `3h0nA.fsdirect.tsv` (direct Foldseek hits) and `3h0nA.fsdirect.dali.tsv` (DALI validated Foldseek hits).

Interpreting the results

Annotated hit list

The annotated outputs are tabular. The format is explained in the User manual. The TSV file has headers, and you could use Pandas or R to draw plots of selected variables. Below, we use Linux commands to extract specific data from the search with the **full protein** structure:

```
gawk ' $3 != "Query" ' 3h0nA.AFDB2.pf.tsv | cut -f 2,4,5,6,8,12,15,33 > x; { head -n 1 x; tail -n +2 x | sort -nrk 2; }
```

sbjct	z-score	rmsd	ali-length	seq-identity	description	species	pfam
jyshA	22.2	1.8	173	38	Zinc finger CGNR domain-containing protein	Nocardia brasiliensis (strain ATCC 700358 / HUJEG-1)	PF11706
jxr8A	17.7	2.6	170	25	Zinc finger CGNR domain-containing protein	Nocardia brasiliensis (strain ATCC 700358 / HUJEG-1)	PF11706
jwecA	16.5	2.4	159	29	Zinc finger CGNR domain-containing protein	Nocardia brasiliensis (strain ATCC 700358 / HUJEG-1)	PF11706
jz3pA	12.7	3.2	152	22	Zinc finger CGNR domain-containing protein	Nocardia brasiliensis (strain ATCC 700358 / HUJEG-1)	PF11706
jx8wA	12.5	3.4	156	27	Zinc finger CGNR domain-containing protein	Nocardia brasiliensis (strain ATCC 700358 / HUJEG-1)	PF11706
jys7A	12.1	3.7	156	22	Zinc finger CGNR domain-containing protein	Nocardia brasiliensis (strain ATCC 700358 / HUJEG-1)	PF11706
jtxeA	11.9	3.1	148	21	Zinc finger CGNR domain-containing protein	Nocardia brasiliensis (strain ATCC 700358 / HUJEG-1)	PF11706
jthcA	11.6	3.4	157	20	Zinc finger CGNR domain-containing protein	Nocardia brasiliensis (strain ATCC 700358 / HUJEG-1)	PF11706
jwwnA	11.1	4.8	159	16	Zinc finger CGNR domain-containing protein	Nocardia brasiliensis (strain ATCC 700358 / HUJEG-1)	PF11706
jucbA	10.9	4.4	157	22	Zinc finger CGNR domain-containing protein	Nocardia brasiliensis (strain ATCC 700358 / HUJEG-1)	PF11706
jwasA	8.6	4.5	144	21	Zinc finger CGNR domain-containing protein	Nocardia brasiliensis (strain ATCC 700358 / HUJEG-1)	PF11706
jxbxA	5.9	1.8	44	36	Zinc finger CGNR domain-containing protein	Nocardia brasiliensis (strain ATCC 700358 / HUJEG-1)	PF11706
gxuuA	4.2	10.7	95	8	Tetraspanin-11 Glycine max	NA	
bdqaA	4.0	4.2	86	8	Serine/threonine-protein kinase AtPK1/AtPK6	Zea mays	NA

We see that the twelve highest Z-scores are all zinc finger GCNR domain-containing proteins *Nocardia brasiliensis*, which represents *Actinomycetota* in AFDB2. AFDB2 is composed of Swissprot and selected proteomes, containing a total of one million proteins. The good hits share 16-38% sequence identity with the query protein. The output also includes hits that are shorter than the minimum core size parameter. They represent rejected hits that are neighbors to an accepted hit. Fortuitously, one of the short hits is a protein containing a single zinc finger (jxbxA).

We further notice that the good hits all belong to the CGNR zinc finger domain family (Pfam identifier PF11706) whereas the ABATE domain has Pfam identifier PF07336. *What's happening here?* ISS_ProtSci assumes that the query represents a single domain, and consequently it expects that annotating the sequence fragment of the structurally aligned region will match a single Pfam profile. The best matching Pfam profile (with lowest e-value) is used for annotation. Here, the PDB structure 3h0n contains two domains, the ABATE domain and the zinc finger domain. The zinc finger domain profile with its invariantly conserved cysteines wins.

The search using the separate **ABATE domain** yielded the same set of good hits as the search with the full protein. After excluding the zinc finger domain from the query, we now see one hit receive the expected Pfam annotation, PF07336. Since the Pfam profile is compared to an amino acid sequence concatenating only the structurally aligned positions, it is not surprising that several CGNR proteins fall below the Pfam profile's trusted cutoff. Z-scores are lower than for the full protein, reflecting the shorter structural alignment.

```
gawk ' $3 != "Query" ' adomA.AFDB2.pf.tsv | cut -f 2,4,5,6,8,12,15,33 > x; { head -n 1 x; tail -n +2 x | sort -nrk 2; }
```

```
sbjct   z-score rmsd    ali-length      seq-identity    description     species pfam
jyshA   15.9    1.6     126     30      Zinc finger CGNR domain-containing protein      Nocardia brasiliensis (strain ATCC 700358 / HUJEG-1)    PF07336
jxr8A   11.5    2.5     119     20      Zinc finger CGNR domain-containing protein      Nocardia brasiliensis (strain ATCC 700358 / HUJEG-1)    NA
jwecA   11.1    2.3     112     25      Zinc finger CGNR domain-containing protein      Nocardia brasiliensis (strain ATCC 700358 / HUJEG-1)    NA
jthcA   7.9     3.6     111     16      Zinc finger CGNR domain-containing protein      Nocardia brasiliensis (strain ATCC 700358 / HUJEG-1)    NA
jwwnA   7.0     3.5     111     12      Zinc finger CGNR domain-containing protein      Nocardia brasiliensis (strain ATCC 700358 / HUJEG-1)    NA
jx8wA   6.9     3.8     111     20      Zinc finger CGNR domain-containing protein      Nocardia brasiliensis (strain ATCC 700358 / HUJEG-1)    NA
jtxeA   6.8     3.0     104     13      Zinc finger CGNR domain-containing protein      Nocardia brasiliensis (strain ATCC 700358 / HUJEG-1)    NA
jys7A   6.6     4.3     106     14      Zinc finger CGNR domain-containing protein      Nocardia brasiliensis (strain ATCC 700358 / HUJEG-1)    NA
jz3pA   6.5     3.7     107     14      Zinc finger CGNR domain-containing protein      Nocardia brasiliensis (strain ATCC 700358 / HUJEG-1)    NA
jucbA   6.4     3.8     110     15      Zinc finger CGNR domain-containing protein      Nocardia brasiliensis (strain ATCC 700358 / HUJEG-1)    NA
jwasA   5.8     3.8     108     13      Zinc finger CGNR domain-containing protein      Nocardia brasiliensis (strain ATCC 700358 / HUJEG-1)    NA
is8dA   3.6     3.5     80      15      Bm12893 Brugia malayi   NA
ie8iA   2.7     3.8     90      8       None    None    NA
hu61A   2.6     6.7     49      4       Pectinesterase inhibitor domain-containing protein      Glycine max     NA
doqdA   2.5     4.7     60      8       MAGE domain-containing protein  Caenorhabditis elegans  NA
hgmmA   2.2     2.9     51      14      Pre-mRNA-processing protein 40C Glycine max     NA
```

The **zinc finger domain** search generated a large number of hits, so we only count the annotations to Pfam clans and families:

```
gawk ' $3 == "AFDB2" ' zincA.AFDB2.pf.tsv | cut -f 33,35 | sort | uniq -c
```

```
    281 NA       NA
    158 PF03884 CL0175
      1 PF05766 CL0263
     12 PF11706 PF11706
```

We recover the twelve hits to the familiar CGNR zinc finger domain family PF11706 and 158 matches to the YacG family (PF03884) which belongs to the TRASH clan (CL0175). TRASH-like domains contain well-conserved cysteine residues that are thought to be involved in metal coordination. These domains are thus expected to be involved in metal trafficking and heavy-metal resistance. It has been suggested that the members adopt a 'treble-clef' fold, with 3/4 beta strands preceding a C-terminal alpha helix (PMID:12713899). There are also 281 structural matches without Pfam annotation (NA).

Understanding DALI Z-scores

The results are sorted by DALI Z-scores, where larger values mean stronger structural similarity. The Z-score normalizes the geometric structural alignment score by accounting for domain sizes. Dali performs local structural alignment across the entire proteins. For Z-score calculation, it employs an

internal domain decomposition algorithm, assigning the match to the most representative pair of candidate domains. In this context, the subject domain refers to the domain assigned within the database protein. Lower Z-scores are indicative of partial matches between domains. Specifically, this occurs when the subject domain is contained within the query but is smaller, the query is contained within the subject domain but is smaller, or only a portion of either the query or subject domain aligns structurally.

It is useful to plot how the alignment length vs. Z-scores of the targets are distributed. Figure 2 reinforces that other families overlap the Z-score range of members of the Query protein's family PF11706.

```
# extract data
gawk ' $3== "AFDB2" ' zincA.AFDB2.pf.tsv | cut -f 4,6,33,35
```

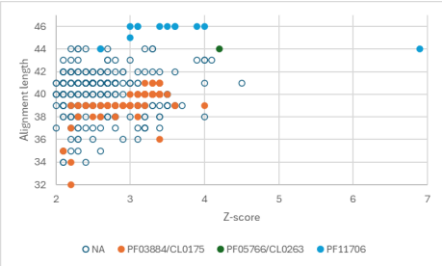

Figure 2: Distribution of structural alignment lengths against Z-score for the zinc finger domain. Data points are coloured according to Pfam annotations. Data is discretized because Z-scores are reported with one decimal.

Visualizing stacked alignments

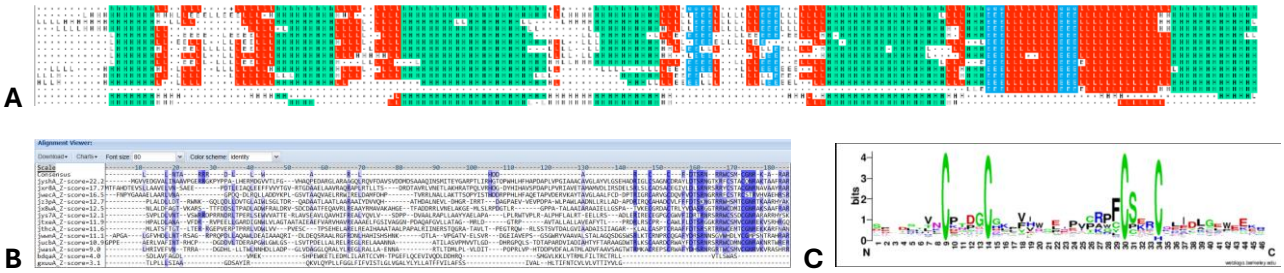

Figure 3: **A)** Stacked alignment view for 3h0nA query coloured by secondary structure assignments (alpha helix: green, beta strand: blue, loop: red). The top row represents consensus. The bottom two hits are unrelated to ABATE proteins. Rendered with MSAViewer, which supports zooming in and out of large alignments. **B)** Stacked alignment for 3h0nA query coloured by sequence identity. The zinc finger domain has invariant cysteines at positions 147, 152 and 172 and either cysteine or histidine at position 168. Rendered with VARDB. **C)** WebLogo representation for the large result set of the zinc finger domain query.

Columns "sequ-pileup" and "DSSP-pileup" of the output file contain projections of the amino acid sequences and secondary structure assignments, respectively, onto the query protein. These projections generate a stacked alignment, hiding insertions relative to the query protein. You can extract stacked alignments of amino acid sequences or secondary structure assignments into FASTA files. You can also filter rows of interest for further analysis by applying additional conditions on specific TSV columns.

```
# stacked amino acid sequences in FASTA format
awk -F'\t' '$3 == "AFDB2" { print ">" $2 " Z=" $4 "\n" $29 }' 3h0nA.AFDB2.tsv

# stacked three-state secondary structure assignments in FASTA format
awk -F'\t' '$3 == "AFDB2" { print ">" $2 " Z=" $4 "\n" $30 }' 3h0nA.AFDB2.tsv
```

Paste or upload these alignments into an online multiple alignment viewer, some of which are illustrated in Figure 3.

Superimposing hits onto the query structure

To explore the hits in 3D, the simplest approach is to use the pairwise alignment option on the Dali server's *pairwise alignment* option. Upload your query structure as the "first structure" and enter the DALI identifiers of selected hits as "second structures." This generates a 3D superposition in the PV viewer, along with links to Pfam annotations, function predictions, and sequence search tools.

Alternatively, you can transform the C-alpha coordinates of AFDB2 hits into the coordinate frame of the query structure using a provided utility script (Figure 4). This script takes the DALI identifier (sbjct) and the result file as inputs. Note that some precision is lost because the sbjct coordinates are retrieved from DALI's internal data files, which store coordinates rounded to one decimal place. Additionally, all residues are labeled as UNK.

```
$ISShome/scripts/ISS_sup.csh jxbxA zincA.AFDB2.pf.tsv
```

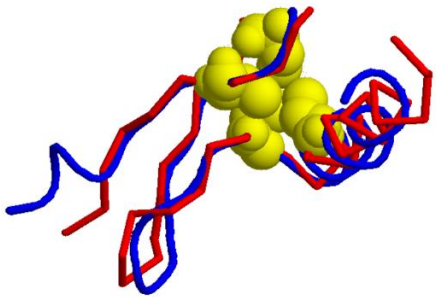

Figure 4: Superimposition of jxbxA (red) onto zinc.pdb (blue, cysteines in yellow CPK). Rendered with Rasmol. **Note:** DALI aligns structures based on the similarity of intramolecular distances rather than by minimizing RMSD. As a result, some alignments may display high RMSD values and poor superpositions. ISS_ProtSci currently does not support this, but you could improve the superposition by using information from the stacked alignment to weight structural equivalences differently at consistently aligned versus gappy positions.

Evaluating recall against positive controls

Pfam provides a curated classification of most UniProt sequences into families and clans (superfamilies), serving as a reference for assessing the recall of biologically meaningful relationships.

The annotation script, which was used above, reports Pfam profiles matching the aligned subsequences, helping identify potentially interesting families. The evaluation script, used here, is based on more accurate assignments in a precomputed database of Pfam matches to full protein sequences to check whether any hits contain a given Pfam family or clan.

The analysis script reports:

- **TRUE targets (TRUE)**:
  Instances listed in the reference classification (ground truth).
- **TRUE POSITIVES (TP)**:
  True targets successfully found in the search.
- **POSITIVES (P)**:
  All hits returned in the search output, regardless of correctness.

Filtering levels (indicated by suffixes _0, _1, and _2) are applied to both the ground truth and search results:

- **_0 (no filtering)**:
  No filters applied; the search is expected to fully reproduce the external reference classification. For transitive closure, P_0 is the number of DALI validation tests performed and TP_0 is undefined (reported as zero).

- **_1 (structural similarity filter)**:
  Only targets with structural similarity to the query structure are included.
  Note: Because all hits are validated by DaliLite, results with a Z-score below 2 are never reported.
- **_2 (common core filter)**:
  An additional filter is applied based on structural alignment length, ensuring that only results meeting the defined common core criteria are presented.

Interpreting the output, look for these key indicators:

- **TRUE_1 close to TRUE_0**: the query structure adequately represents the reference class.
- **TRUE_2 close to TRUE_1**: the common core size parameter is appropriate.
- **TP_2 close to TRUE_2:** high recall is achieved.
- **TP_2 close to P_2**: high precision is achieved.

Pfam family members are easy targets

There are nine members of the Pfam family PF07336 in AFDB2, all of which are successfully recalled by ISS_ProtSci (closure method) at every filtering level. In contrast, Foldseek recalls only six members, even at relaxed e-value cutoffs:

```
# run the analysis script, using PF07336 as reference
$ISShome/3_analyze.csh -cd1 adomA -clan PF07336 | grep EVAL
```

EVAL	method	TRUE_0	TP_0	P_0	TRUE_1	TP_1	P_1	TRUE_2	TP_2	P_2
EVAL	closure	9	0	16	9	9	16	9	9	11
EVAL	foldseek(e<10)	9	6	1516	9	6	9	9	6	8
EVAL	foldseek(e<1)	9	6	33	9	6	9	9	6	8
EVAL	foldseek(e<0.1)	9	5	6	9	5	6	9	5	6
EVAL	foldseek(e<0.01)	9	2	3	9	2	3	9	2	3
EVAL	foldseek(e<0.001)	9	1	2	9	1	2	9	1	2

There are two additional proteins in the P_2 set that meet the common core criteria but do not reach the trusted cutoff in a Pfam profile search required for inclusion in the TRUE_2 set.
However, all proteins in the P_2 set are annotated in Uniprot as “Zinc finger CGNR domain-containing protein”, indicating that they are likely correct hits.

Both transitive closure and Foldseek (at e-values below 0.1) successfully retrieve all twelve members of Pfam family PF11706. Here, filtering by Z-score or alignment length does not affect the results.

```
# run the analysis script, using PF11706 as reference
$ISShome/3_analyze.csh -cd1 zincA -clan PF11706 2> err | grep EVAL
```

EVAL	method	TRUE_0	TP_0	P_0	TRUE_1	TP_1	P_1	TRUE_2	TP_2	P_2
EVAL	closure	12	0	4048	12	12	452	12	12	452
EVAL	foldseek(e<10)	12	12	198	12	12	13	12	12	13
EVAL	foldseek(e<1)	12	12	13	12	12	13	12	12	13
EVAL	foldseek(e<0.1)	12	12	12	12	12	12	12	12	12
EVAL	foldseek(e<0.01)	12	5	5	12	5	5	12	5	5
EVAL	foldseek(e<0.001)	12	1	1	12	1	1	12	1	1

Since Pfam families are defined by sequence similarity, they serve as relatively easy targets. However, transitive closure identifies significantly more structurally similar positives than Foldseek. It is important to note that this test only evaluates positive controls—the biological significance of additional positive hits beyond the true-positive set remains uncertain.

Pfam clans are a more challenging test

Pfam clans (superfamilies) unify remotely related Pfam families. Below, using the TRASH clan (CL0175) as reference reveals a clear difference in favour of transitive closure over Foldseek.

```
# run the analysis script, using CL0175 as reference
$ISShome/3_analyze.csh -cd1 zincA -clan CL0175 2> err | grep EVAL
```

EVAL	method	TRUE_0	TP_0	P_0	TRUE_1	TP_1	P_1	TRUE_2	TP_2	P_2
EVAL	closure	1015	0	4048	231	180	452	231	180	452
EVAL	foldseek(e<10)	1015	93	198	231	0	13	231	0	13
EVAL	foldseek(e<1)	1015	0	13	231	0	13	231	0	13
EVAL	foldseek(e<0.1)	1015	0	12	231	0	12	231	0	12
EVAL	foldseek(e<0.01)	1015	0	5	231	0	5	231	0	5
EVAL	foldseek(e<0.001)	1015	0	1	231	0	1	231	0	1

We can make a number of observations. The TRASH clan as defined by Pfam profile searches includes TRUE_0 = 1015 proteins of AFDB2. However, only TRUE_1 = 231 of them are structurally similar to our query structure and thus findable by Dali, and the transitive closure search reached 180 of them. In contrast, Foldseek finds a weak match to 93 clan members (with e-value above 1) but none of them is structurally similar to the query structure (TP_1 = 0).

The common core size and Z-score cutoffs applied in the second filtering step (suffix _2) can be set by optional command-line arguments to the analysis script. For example, a stricter definition of the minimum common core size set to 36 brings the TP_2 count closer to TRUE_2, indicating good recall of the strongest structural similarities. Ramping up the Z-score cutoff (zcut) to 2.5 further increases the precision of reported hits in the P_2 set. Foldseek's recall remains zero.

```
# run analysis script with higher minlali
$ISShome/3_analyze.csh -cd1 zincA -clan CL0175 -minlali 36 2> err | grep EVAL | head -2
```

EVAL	method	TRUE_0	TP_0	P_0	TRUE_1	TP_1	P_1	TRUE_2	TP_2	P_2
EVAL	closure	1015	0	4048	231	180	452	**199**	**173**	**441**

```
# run analysis script with higher minlali and higher zcut
$ISShome/3_analyze.csh -cd1 zincA -clan CL0175 -minlali 36 -zcut 2.5 2> err | grep EVAL | head -2
```

EVAL	method	TRUE_0	TP_0	P_0	TRUE_1	TP_1	P_1	TRUE_2	TP_2	P_2
EVAL	closure	1015	0	4048	231	180	379	**167**	**162**	**299**

The "Advanced Topics" section discusses how to refine your search if recall or precision are unsatisfactory.

Implications

When inferring homology between proteins—especially in the absence of strong sequence similarity—structural evidence can be highly informative. The following criteria help support such claims using DALI structural alignments in our case study:

1. **DALI Z-Scores**
   Homologs typically show higher Z-scores than convergent folds, often forming nested clusters in structural space.
   - There was a clear gap in Z-scores between ABATE domains and unrelated proteins.
   - The range of Z-scores for the zinc finger domain overlapped with those of other zinc finger domain families, suggesting they are related

2. **Sequence Conservation in Core Regions**
   Remote homologs often retain conserved residues at key structural or functional positions.
   - The zinc finger domain showed conserved zinc binding residues.

3. **Consistent Structural Core**
   Homologs preserve a common structural core that is larger and more coherent than expected from background alignments.
   - The ABATE domain proteins consistently aligned six secondary structure elements: helices 1, 2, 3 and 5, as well as a beta hairpin.

4. **Functional and Contextual Clues (Optional)**
   Shared functions, cofactor binding, or domain context further support evolutionary relatedness.
   - The ABATE domain was found in a single proteome within AFDB2, suggesting a relatively late evolutionary innovation that recombines a sensor domain with a DNA-binding domain to control an unknown signalling pathway.

Advanced topics

Search with user-defined parameters

Leave some slack in Z-score threshold

If you have prior knowledge of the expected alignment length and Z-score ranges for true positives, you can set these as parameters in the transitive closure search to avoid spending time on irrelevant similarities. This approach is especially effective for filtering out widespread folding motifs embedded within a larger fold. Here, we illustrate the effect of such filtering using the zinc finger domain as a query. Using the commands below, the basic run (A) took 68 seconds and yielded 398 hits, while the stricter run (B) took 29 seconds and yielded 305 hits.

```
# Default search
$ISShome/1_search.csh -pdbfile zinc.pdb -cd1 zincA
# Stricter search
$ISShome/1_search.csh -pdbfile zinc.pdb -cd1 ztstA -minlali 40 -zcut 3.0
```

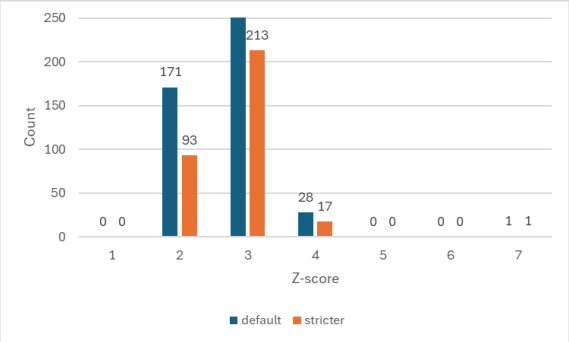

Figure 5: P_1 set of default and stricter search using a Z-score cutoff of 3.0 with the zinc finger domain as query.

Examining the Z-score distributions of the hits (Figure 5), we observe two main effects. First, distance in the fold space graph does not correlate smoothly with decreasing Z-scores. The stricter search misses some hits in the Z = 4 bin, indicating that the path from the first shell to these hits dips below the stricter Z-score threshold. Second, the P_1 sets from both searches include many rejected candidates, which were validated during the process of defining the boundary of the accepted region.

In conclusion, the default parameters usually provide a good starting point unless you have specific prior knowledge to guide further tuning.

Relaxed searches

Here, we perform relaxed searches, disabling structural alignment length as a termination criterion. This is not recommended for query structures that contain sizable, frequently recurring structural motifs, as the search would attempt to iterate over all of them. These results are used for method comparison in the next section.

```
# transitive closure search
$ISShome/1_search.csh -pdbfile 3H0N.pdb.gz -cd1 rlxfA -minlali 1
$ISShome/1_search.csh -pdbfile adom.pdb -cd1 rlxaA -minlali 1
$ISShome/1_search.csh -pdbfile zinc.pdb -cd1 rlxzA -minlali 1
# annotation
$ISShome/2_annotate.csh -pdbfile 3H0N.pdb.gz -cd1 rlxfA
$ISShome/2_annotate.csh -pdbfile adom.pdb -cd1 rlxaA
$ISShome/2_annotate.csh -pdbfile zinc.pdb -cd1 rlxzA
```

Systematic DALI search confirms the uniqueness of the ABATE domain fold

The ABATE domain protein is small and unique enough to make a systematic DALI search of the one million proteins in AFDB2 feasible. Here, we use run identifiers sysfA for the full protein query, systA for the ABATE domain query, and syszA for the zinc finger domain query. (For evaluation purposes, you could use the ISS_rundali.py script for constructing a positive control set for a subset of AFDB2, if no Pfam set is available.)

```
# search AFDB2.list in chunks
for c in {a..z}; do
  grep ^$c AFDB2.list > $c.list
  # full protein query
  python3 $ISShome/scripts/ISS_rundali.py --pdbfile 3H0N.pdb.gz --pdbid sysf -target $c.list --tsvfile sysfA.AFDB2_$c.dali.tsv
  # ABATE domain query
  python3 $ISShome/scripts/ISS_rundali.py --pdbfile adom.pdb --pdbid syst -target $c.list --tsvfile systA.AFDB2_$c.dali.tsv
  # zinc finger domain query
  python3 $ISShome/scripts/ISS_rundali.py --pdbfile zinc.pdb --pdbid sysz -target $c.list --tsvfile syszA.AFDB2_$c.dali.tsv
done
# merge chunks
for cd1 in sysf syst sysz; do
  cat $cd1.AFDB2_*.dali.tsv > $cd1.AFDB2.dali.tsv
done
# run annotation script
Python3 $ISShome/2_annotate.csh -pdbfile 3H0N.pdb.gz -cd1 sysfA
python3 $ISShome/2_annotate.csh -pdbfile adom.pdb -cd1 systA
python3 $ISShome/2_annotate.csh -pdbfile zinc.pdb -cd1 syszA
```

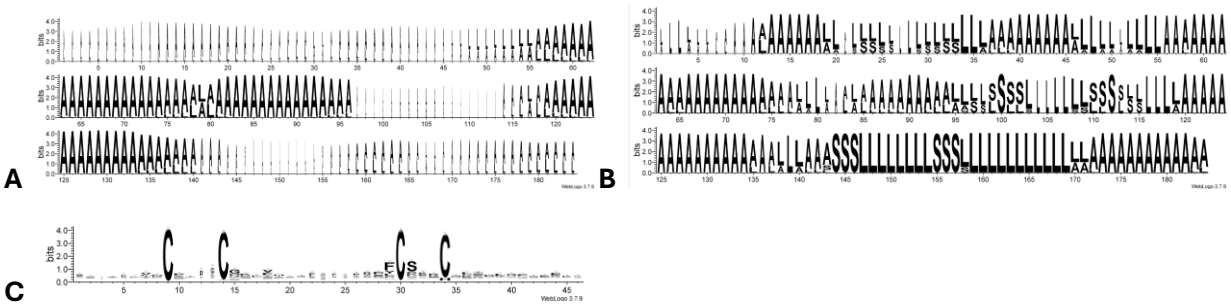

Figure 6: Repurposing WebLogo to display the secondary structure assignments of **(A)** all 7388 hits from a systematic DALI search of AFDB2, and **(B)** 11 members of PF11706 in AFDB2. Encoding: 'A' for alpha helix, 'S' for beta strand, 'L' for loop. The full 3h0nA structure was used as the query. A three-helix bundle motif is prominent in (A). **(C)** WebLogo of the 760 hits from a systematic DALI search of AFDB2 using the zinc finger domain as query. The conserved cysteines indicate that most of them are probable zinc binders.

The searches return a large number of hits matching the three-helical bundle formed by the third, fourth, and fifth helices of the ABATE domain (Figure 6A). However, in all zinc finger CGNR domain-containing proteins, it is the first, second, third, and fifth alpha helices — but not the fourth — that are

consistently aligned (Figure 6B). Filtering with a corresponding pattern in the systematic DALI results (using the full protein as the query) revealed two additional matches with low Z-scores.

```
cut -f 2,4,30 sysfA.AFDB2.pf.tsv | \
 grep '......HHHHHH.....................HHH[LH][LH][LH]............HHHHHH..HHHHHHHH.....................................................HHHHHH.HHHHH.....................................'
```

```
vex8A   2.6     LLLLL..LHHHHHHHHHH...HHHH..LLLL.HHHHHHHHHH....LLLH.HHHHHHHHHHHHHHHHHHHHHHHH.HHHLHHHHHHHHHH..HHHH..............HHHH.HHHHHHHHHHHHHHHHH.................HHHH..............LLLL
ebg4A   2.8     .LLLLL..LHHHHHHHHHH...HHLL.......HHHHHHHHHH...HLLLL.HHHHHHHHHHHHHHHHHHHHHHHH..LLLHHHHHHHHH..HHHH..............HHHH.HHHHHHHHHHHHHHHHH.....................................HHHHH
jwasA   8.6     HLLH......HHHHHHHHL..LLLL.....LLLLLLHHHHHHHHHLL..........HHHHHHHHHHHHHHHHHHHHL.........HHHLLLLEEEELLL...LLLEEEEL..LLHHHHHHHHHHHHHHHHHLLLHHHEEELLLLLLLEEELLLLLLLLLLLLHHHHHHHHHHHH
jucbA   10.9    LLLH......HHHHHHHLL.LLLL....LLLLLLLLLHHHHHHHLL..LLLLHHHHHHHHHHHHHHHHHHLLL.........HHHHHLLEEEELLL...LLLLLEEE.LLLLHHHHHHHHHHHHHHHHHLHHHEEELLLLLLLEEELLLLLLLLLLLLHHHHHHHHHHHH
jwwnA   11.1    ..LLLL....HHHHHHHHHL.LLLL.LLLLLLLLHHHHHHHHHHLLLL..LLLHHHHHHHHHHHHHHHHHHHHHLL.....LLLL..LLLLEE.EEEEL..LLLEEEEEE..LLHHHHHHHHHHHHHHHHHLLHHHEEELLLLLLLEEELLLLLLLLLLLLHHHHHHHHHHHH
jthcA   11.6    ..........LHHHHHHLL..EELL.LLLLEELLLHHHHHHHHH...LLLLL..LHHHHHHHHHHHHHHHHHHHLLHHHHHHHHHHHLLL.EEEEL..LLLLEEE..ELLLHHHHHHHHHHHHHHHHLLLL..EEEELLLLLLLEEELLLLLLLLLLLLHHHHHHHHHHHH
jtxeA   11.9    ..........LHHHHHHLL..LLLL..LLLLLLLLHHHHHHHHHLLHHHHHLLHHHHHHHHHHHHHHHHHHHHLLLL.LLHHHHHHHHHHH..LEEE.....LLEE.....HHHHHHHHHHHHHHHL...LHHHEEELL..LLLEEELLLLLLLLLLLLLHHHHHHHHHHHH
jys7A   12.1    ..........LHHHHHHLL..EELLLLLLEELLLHHHHHHHHLLLL.LLLLLHHHHHHHHHHHHHHHHHHH..LLLL..LLHHHHHHHHHHLLEELLL...LLLEELLL.HHHHHHHHH.HHHHL...LLHHHEEELLLLLLLEEELLLLLLLLLLLLHHHHHHHHHHHH
jx8wA   12.5    ..........LHHHHHHLL.EELLL..LLLEELLLHHHHHHHHHHLLLL.LLLLHHHHHHHHHHHHHHHHHHHHHLL..LLLHHHHHHHHLLL.EEEELLLLEE.....LLHH.HHHHHHHHHHHHHHLLL..LEEELLLLLLLEEELLLLLLLLLLLLHHHHHHHHHHHH
jz3pA   12.7    ..........LHHHHHLL..EEEE..LEEEELLLHHHHHHHHHLLLLL..LLLHHHHHHHHHHHHHHHHHHH.....LLLHHHHHH.LLLE.EEEE...LLEEEE.EELLLHH.HHHHHHHHHHHHHHH.LHHHEEELLLLLLLEEELLLLLLLLLLLLHHHHHHHHHLL
jwecA   16.5    ...LLLLHHHHHHHHHLL...........LLLL.LHHHHHHLLLLL.LLLLHHHHHHHHHHHHHHHHHHHLL.......LHHHHHHHHHLLLLEEELLLLLLEEELLLLLLHHHHHHHHHHHHHHHHH.LHHEEELLLLLLLEEELLLLLLLLLLLLHHHHHHHHHHHH
jxr8A   17.7    LLLLHHHHHHHHHHHHHH.LLLL.....LLLLLHHHHHHHHHLLLLL.LLLLHHHHHHHLLHHHHHHHLL....LHHHHHHHHHHHHLLLLEEELLL.LLEEELLLLLLHHHHHHHHHHHHHHHHHLLLHHHEEELLLLLLLEEELLLLLLLLLLLLHHHHHHHHHHHL
jyshA   22.2    ......LHHHHHHHHHHHHLLEEELLEEELLL.HHHHHHHHHHL...LLLLHHHHHHHHHHHHHHHHHHHHHLLHHHHHHHHHHHHLLLLEEELLLLLLEEELLLLLLHHHHHHHHHHHHHHHHLLLHHHEEELLLLLLLEEELLLLLLLLLLLLHHHHHHHHHHHH
```

The new hits, **ebg4A** and **vex8A**, are EF-hand domain-containing proteins with 27% sequence identity and a very high pairwise DALI Z-score of 31.3 (data not shown). The AlphaFold predictions reveal a structural motif of orthogonally packed helices that resembles the common core of ABATE domain-containing proteins, but it is embedded within a much larger domain (Figure 7). The overall domain fold is different from that of the ABATE domain, leading us to conclude that the ABATE domain fold is indeed unique.

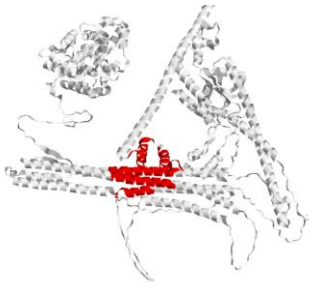

Figure 7: ABATE-like substructure in ebg4A highlighted in red.

Comparison of ISS_ProtSci, Foldseek and DALI

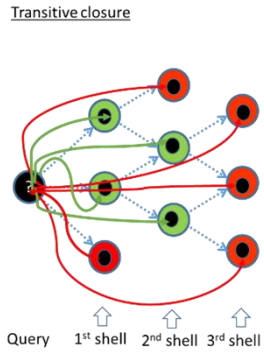

Figure 8: Schematic representation of the transitive closure search method. Dashed blue arrows represent neighbor relationships predicted by Foldseek. Solid arrows are color-coded based on the validation results: green indicates pairs that pass the filtering criteria (Dali Z-score and alignment length), while red indicates pairs that fail. The search terminates when no new candidates in a shell meet the validation criteria.

ISS_ProtSci aims to gather all instances of the fold represented by the query structure, assuming that the targets form a connected component within the underlying fold space graph (Figure 8). ISS_ProtSci can only reach targets that have a connecting path satisfying:

- the Foldseek e-value threshold of the underlying fold space graph, and
- the alignment length and Z-score criteria of DALI validation.

To test for **false negatives**, we compare the results of Foldseek, the transitive closure search with default parameters, the transitive closure search with relaxed parameters, and a systematic DALI search. The search results have been generated previously in this tutorial:

- Foldseek using –max-seqs 50000 to increase sensitivity: {3h0nA,adomA,zincA}.fsdirect.dali.tsv
- Transitive closure with default parameters: {3h0nA,adomA,zincA}.AFDB2.pf.tsv
- Transitive closure with relaxed parameters: {rlxfA,rlxtA,rlxzA}.AFDB2.pf.tsv

- Systematic DALI search: {sysfA,systA,syszA}.AFDB2.pf.tsv

There is a clear discrimination of the ABATE domain against database background along both the Z-score and alignment length axes (Figure 9A,C). Technically, both the full-protein and ABATE domain queries exhibit broken connectivity at low Z-scores, as the systematic search yields significantly more hits than the relaxed search (Figure 9B, D). However, these hits with Z-scores below 4 are noninformative, partial domain matches. In contrast, for the small zinc finger domain, all hits down to a Z-score of 2 are meaningful, as they share the zinc-binding site. Some hits identified by the systematic search are not recovered by transitive closure (Figure 9E–F), likely due to their proximity to the Z-score threshold and the non-metric nature of the similarity measures used by ISS_ProtSci (i.e., Foldseek e-values and DALI Z-scores).

In conclusion, while the continuity of the fold space graph remains the Achilles' heel of ISS_ProtSci, its efficient sampling strategy is sufficient to unify families of remote homologs—even when Foldseek reports statistically insignificant e-values.

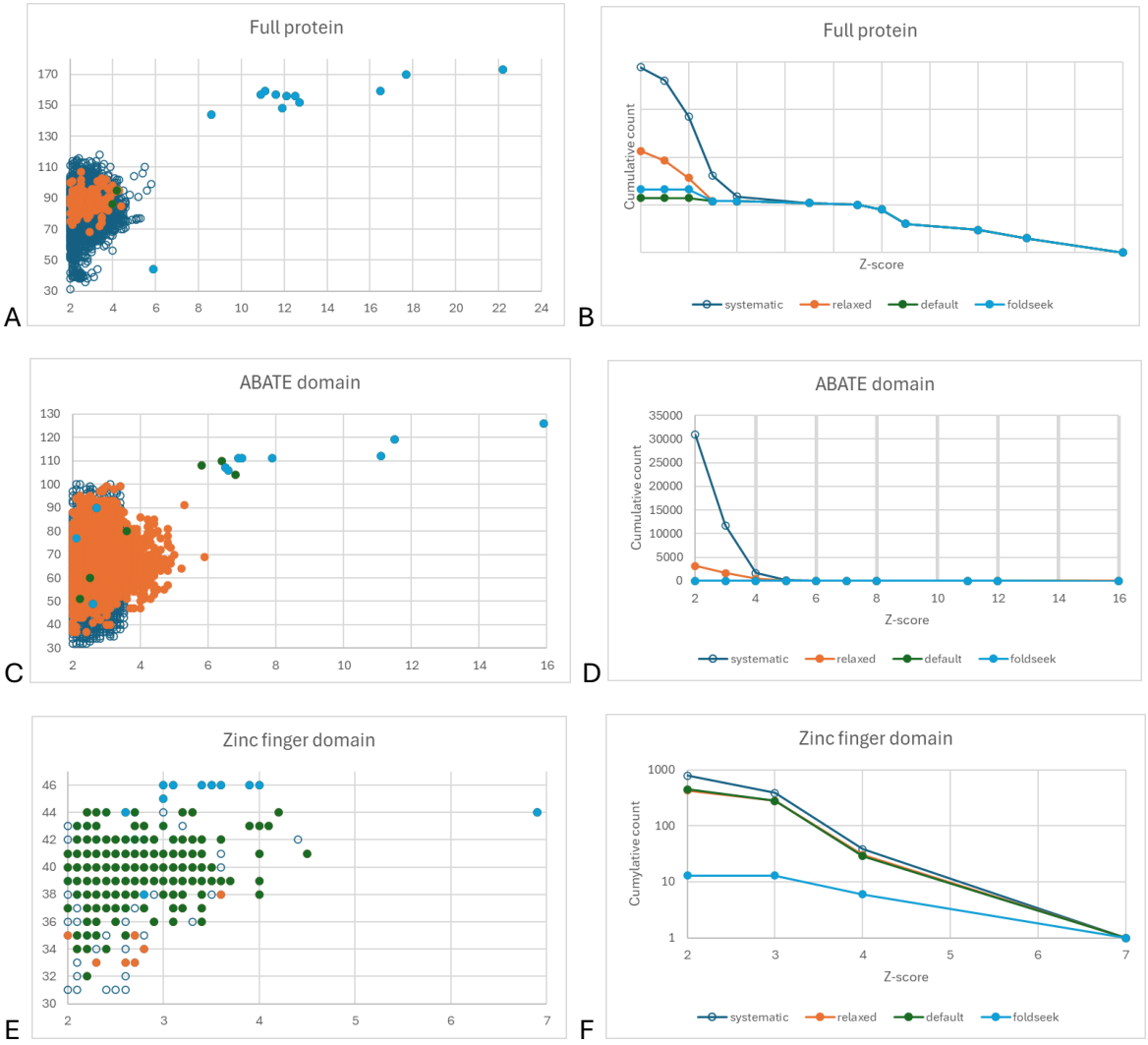

Figure 9: Alignment length plotted against Z-score **(A,C,E)** and **c**umulative counts of hits above a given Z-score threshold (**B,D,F)** reported by Foldseek, default transitive closure search, relaxed transitive closure search, and systematic DALI comparison.

Summary

In this tutorial, we explored how ISS_ProtSci can effectively detect remote homologs, using a case study that confirmed the uniqueness of the ABATE domain and linked the zinc finger domain to many other structurally similar zinc finger domain families.

Best practices to keep in mind:

- **Use domain-level queries** for more accurate and detailed annotations.
- **Default parameters** can be used for automated scripting and large-scale searches.
- **Include positive controls** to help fine-tune parameters for specific cases and improve sensitivity where needed.

By following these guidelines, you can get the most out of ISS_ProtSci for your structural search projects.

Cheatsheet

```
export ISShome=~/ISS_ProtSci-1/

# query structures
# wget https://files.rcsb.org/download/3H0N.pdb.gz
# zcat 3H0N.pdb.gz | egrep '^HETAT|^ATO' | gawk ' $6 < 139 ' > adom.pdb
# zcat 3H0N.pdb.gz | egrep '^HETAT|^ATO' | grep -v HOH | gawk ' $6 > 138 && $6 <= 184' > zinc.pdb

# basic search with defaults
# fuil structure
$ISShome/1_search.csh -pdbfile 3H0N.pdb.gz -cd1 3h0nA
$ISShome/2_annotate.csh -pdbfile 3H0N.pdb.gz -cd1 3h0nA

# separate ABATE domain
$ISShome/1_search.csh -pdbfile adom.pdb -cd1 adomA
$ISShome/2_annotate.csh -pdbfile adom.pdb -cd1 adomA

# separate zinc finger domain
$ISShome/1_search.csh -pdbfile zinc.pdb -cd1 zincA
$ISShome/2_annotate.csh -pdbfile zinc.pdb -cd1 zincA

# hit list
gawk ' $3 != "Query" ' 3h0nA.AFDB2.pf.tsv | cut -f 2,4,5,6,8,12,15,33 > x; { head -n 1 x; tail -n +2 x | sort -nrk 2; }
# stacked dssp
gawk ' $3 != "Query" ' 3h0nA.AFDB2.pf.tsv | cut -f 2,4,30 > x; { head -n 1 x; tail -n +2 x | sort -nrk 2; }

# fasta conversion
# stacked amino acid sequences
awk -F'\t' '$3 == "AFDB2" { print ">" $2 " Z=" $4 "\n" $29 }' 3h0nA.AFDB2.tsv

# stacked three-state secondary structure assignments
awk -F'\t' '$3 == "AFDB2" { print ">" $2 " Z=" $4 "\n" $30 }' 3h0nA.AFDB2.tsv

# 3-D superposition
$ISShome/scripts/ISS_sup.csh jxbxA zincA.AFDB2.pf.tsv

# ABATE hits
gawk ' $3 != "Query" ' adomA.AFDB2.pf.tsv | cut -f 2,4,5,6,8,12,15,33 > x; { head -n 1 x; tail -n +2 x | sort -nrk 2; }

# zinc finger hits
gawk ' $3 == "AFDB2" ' zincA.AFDB2.pf.tsv | cut -f 33,35 | sort | uniq -c
gawk ' $3 != "Query" ' zincA.AFDB2.pf.tsv | cut -f 2,4,5,6,8,11,12,15,33 > x; { head -n 1 x; tail -n +2 x | grep PF11706 | sort -nrk 2; }
# grouped by PF
# extract data
gawk ' $3== "AFDB2" ' zincA.AFDB2.pf.tsv | cut -f 4,6,33,35

# run the analysis script, using PF07336 as reference
$ISShome/3_analyze.csh -cd1 adomA -clan PF07336 | grep EVAL

# run the analysis script, using PF11706 as reference
$ISShome/3_analyze.csh -cd1 zincA -clan PF11706 2> err | grep EVAL

# run the analysis script, using CL0175 as reference
$ISShome/3_analyze.csh -cd1 zincA -clan CL0175 2> err | grep EVAL

# run analysis script with higher minlali
$ISShome/3_analyze.csh -cd1 zincA -clan CL0175 -minlali 36 2> err | grep EVAL | head -2

# run analysis script with higher minlali and higher zcut
$ISShome/3_analyze.csh -cd1 zincA -clan CL0175 -minlali 36 -zcut 2.5 2> err | grep EVAL | head -2

# Stricter search
$ISShome/1_search.csh -pdbfile zinc.pdb -cd1 ztstA -minlali 40 -zcut 3.0

# run relaxed transitive closure search
$ISShome/1_search.csh -pdbfile 3H0N.pdb.gz -cd1 rlxfA -minlali 1
$ISShome/1_search.csh -pdbfile adom.pdb -cd1 rlxaA -minlali 1
$ISShome/1_search.csh -pdbfile zinc.pdb -cd1 rlxzA -minlali 1
# annotation
$ISShome/2_annotate.csh -pdbfile 3H0N.pdb.gz -cd1 rlxfA
$ISShome/2_annotate.csh -pdbfile adom.pdb -cd1 rlxaA
$ISShome/2_annotate.csh -pdbfile zinc.pdb -cd1 rlxzA

# search AFDB2.list in chunks
for c in {a..z}; do
  grep ^$c AFDB2.list > $c.list
  # full protein query
  python3 $ISShome/scripts/ISS_rundali.py --pdbfile 3H0N.pdb.gz --pdbid sysf -target $c.list --tsvfile sysfA.AFDB2_$c.dali.tsv
  # ABATE domain query
  python3 $ISShome/scripts/ISS_rundali.py --pdbfile adom.pdb --pdbid syst -target $c.list --tsvfile systA.AFDB2_$c.dali.tsv
  # zinc finger domain query
  python3 $ISShome/scripts/ISS_rundali.py --pdbfile zinc.pdb --pdbid sysz -target $c.list --tsvfile syszA.AFDB2_$c.dali.tsv
done
# merge chunks
for cd1 in sysfA systA syszA; do
  cat $cd1.AFDB2_*.dali.tsv > $cd1.AFDB2.dali.tsv
done
# run annotation script
$ISShome/2_annotate.csh -pdbfile 3H0N.pdb.gz -cd1 sysfA
$ISShome/2_annotate.csh -pdbfile adom.pdb -cd1 systA
$ISShome/2_annotate.csh -pdbfile zinc.pdb -cd1 syszA

# orthogonal helix screen
cut -f 2,4,30 sysfA.AFDB2.pf.tsv | \
 grep '......HHHHH.....................HHH[LH][LH][LH].............HHHHH..HHHHHHHH.....................................................HHHHHH.HHHHH.....................................................'

# data for (z,lali) plots
cut -f 3,5 sysfA.AFDB2.dali.tsv |sort -n| uniq
```