BRAT Tutorial (v. 1.0.0)

Joe Bemister-Buffington, Leslie Kuhn
Protein Structural Analysis and Design Lab, Michigan State University

Before your first use of BRAT

Make sure python2.7.5 or greater installed. If not check section 1 of the Quick Guide to BRAT for recommended install instructions. Section 2 of the Quick Guide describes BRAT input preparation and running BRAT. Section 3 of the Quick Guide describes the output from BRAT. This tutorial is designed to take you through a sample case for BRAT. This tutorial also allows you to compare your output to pre-computed output to ensure that your version of BRAT is functioning properly.

Preparing query and target pdb files and running DALI

The very first step to be done for input preparation is to gather or create your query and target pdb files. This can be done by downloading a pdb structure from the Protein Data Bank website at http://www.rcsb.org/pdb/home/home.do or you can create your own pdb file from any other source. For instance, if a homology model was done and put into pdb format. Once the pdb files are collected, make sure there is a header line present before any ATOM entries. This is necessary for DALI to establish the beginning of the pdb file. If this header line is not presented, DALI will ignore the first residue in your ATOM entries. The header line does not have to be extensive. It can match the format of a PDB structure file or it can be as simple as a line only containing “HEADER “ as shown below.

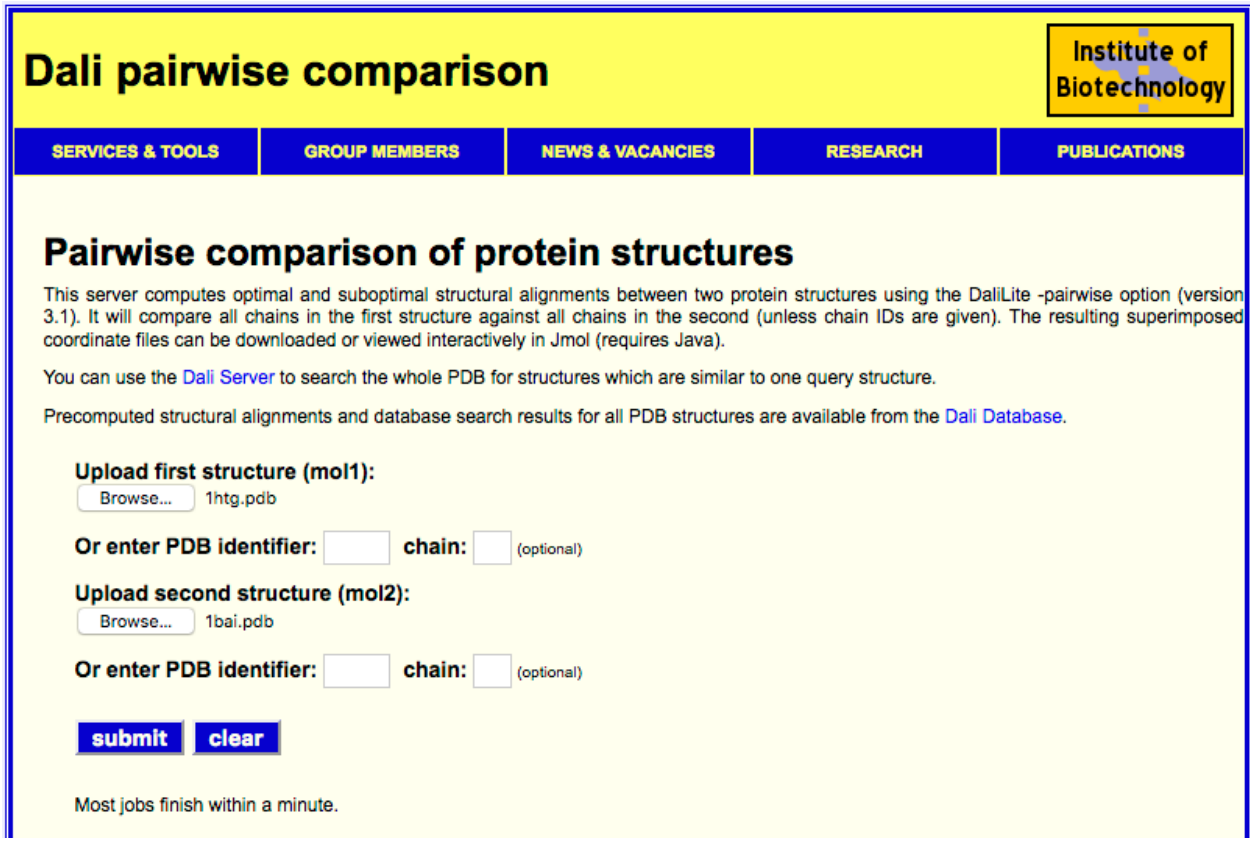
```
HEADER
ATOM   2040  N   TRP A  40       5.474  23.414  49.658  1.00 99.00
ATOM   2041  CA  TRP A  40       6.029  23.293  48.312  1.00 99.00
ATOM    246  C   TRP A  40       5.626  24.490  47.506  1.00 44.83
ATOM    247  O   TRP A  40       6.146  24.732  46.414  1.00 44.83
ATOM   2042  CB  TRP A  40       5.541  22.021  47.609  1.00 99.00
ATOM    159  CG  TRP A  40       6.366  20.855  47.961  1.00 39.60
```

Example of a simple query pdb file created by the user. Notice the HEADER line before the ATOM entries. This is important in order for DALI to consider all of the ATOM entries present in the pdb file.

For this tutorial we will be using the pdb files below:

QUERY: ~/Binding_Residue_Alignment_Tool/Input/sample_4/1htg.pdb
TARGET: ~/Binding_Residue_Alignment_Tool/Input/sample_4/1bai.pdb

Once your query and target pdb files are gathered or created, your next step is to run DALI. What DALI does is use the main chain atoms each protein and creates an alignment based on the angles and positioning of said atoms. Thus it's important that all of the main chain atoms are present in the pdb files. What BRAT requires is a DALI pairwise alignment which can be done at http://ekhidna.biocenter.helsinki.fi/dali_lite/start. It will ask for two structures (mol1 and mol2). You should upload the query pdb as mol1 and the target pdb as mol2. An example is shown below:



Example of a the DALI pairwise server with two pdb files uploaded for alignment.

After uploading your query and target file, submit for a DALI alignment. The process is really quick and will usually finish in under a minute. Once completed the current page will show the results as such:

Results

Parseable data

- mol1A
- mol1B

Query: mol1A

MOLECULE: HIV-1 PROTEASE;

Select neighbours (check boxes) for viewing as multiple structural alignment or 3D superimposition. The list of neighbours is sorted by Z-score. Similarities with a Z-score lower than 2 are spurious. Each neighbour has links to pairwise structural alignment with the query structure, and to the PDB format coordinate file where the neighbour is superimposed onto the query structure.

Structural Alignment | Expand gaps | 3D Superimposition (Jmol Applet) | Sequence Similarity Search | Reset Selection

Summary

	No:	Chain	Z	rmsd	lali	nres	%id	PDB	Description
☐	1:	mol2-A	14.3	1.8	99	125	39	PDB	MOLECULE: PROTEASE;
☐	2:	mol2-B	14.3	1.8	99	124	39	PDB	MOLECULE: PROTEASE;

Pairwise Structural Alignments

Notation: three-state secondary structure definitions by DSSP (reduced to H=helix, E=sheet, L=coil) are shown above the amino acid sequence. Structurally equivalent residues are in uppercase, structurally non-equivalent residues (e.g. in loops) are in lowercase. Amino acid identities are marked by vertical bars.

Example of the initial output page for a DALI pairwise alignment. The data we need for BRAT is found under the Parseable data link (boxed in for this tutorial) on the top left of the page.

Once you're at this page, click the Parseable data link to access the alignment data BRAT uses. The Parseable data link will take you to the image shown on the next page of this tutorial.

```
# Query: mol1A
# No:  Chain   Z    rmsd lali nres  %id PDB  Description
   1:  mol2-A 14.3  1.8   99   125   39   MOLECULE: PROTEASE;
   2:  mol2-B 14.3  1.8   99   124   39   MOLECULE: PROTEASE;

# Structural equivalences
   1: mol1-A mol2-A     1 -   6 <=>     1 -   6   (PRO    1  - TRP    6  <=> LEU    1  - GLU    6 )
   1: mol1-A mol2-A     7 -  16 <=>     9 -  18   (GLN    7  - GLY   16  <=> ASP    9  - THR   18 )
   1: mol1-A mol2-A    17 -  35 <=>    29 -  47   (GLY   17  - GLU   35  <=> SER   29  - GLU   47 )
   1: mol1-A mol2-A    36 -  45 <=>    50 -  59   (MET   36  - LYS   45  <=> TRP   50  - ALA   59 )
   1: mol1-A mol2-A    46 -  59 <=>    63 -  76   (MET   46  - TYR   59  <=> GLN   63  - SER   76 )
   1: mol1-A mol2-A    60 -  67 <=>    78 -  85   (ASP   60  - CYS   67  <=> ASP   78  - ILE   85 )
   1: mol1-A mol2-A    68 -  99 <=>    92 - 123   (GLY   68  - PHE   99  <=> GLU   92  - ASN  123 )
   2: mol1-A mol2-B     1 -   6 <=>     1 -   6   (PRO    1  - TRP    6  <=> LEU    1  - GLU    6 )
   2: mol1-A mol2-B     7 -  16 <=>     9 -  18   (GLN    7  - GLY   16  <=> ASP    9  - THR   18 )
   2: mol1-A mol2-B    17 -  35 <=>    29 -  47   (GLY   17  - GLU   35  <=> SER   29  - GLU   47 )
   2: mol1-A mol2-B    36 -  44 <=>    50 -  58   (MET   36  - PRO   44  <=> TRP   50  - GLU   58 )
   2: mol1-A mol2-B    45 -  59 <=>    62 -  76   (LYS   45  - TYR   59  <=> PRO   62  - SER   76 )
   2: mol1-A mol2-B    60 -  67 <=>    78 -  85   (ASP   60  - CYS   67  <=> ASP   78  - ILE   85 )
   2: mol1-A mol2-B    68 -  99 <=>    92 - 123   (GLY   68  - PHE   99  <=> GLU   92  - ASN  123 )

# Translation-rotation matrices
-matrix  "mol1-A mol2-A  U(1,.)   0.747749  0.600257  0.283837          -6.057237"
-matrix  "mol1-A mol2-A  U(2,.)  -0.663363  0.693803  0.280334           1.673261"
-matrix  "mol1-A mol2-A  U(3,.)  -0.028655 -0.397906  0.916978         -18.198193"
-matrix  "mol1-A mol2-B  U(1,.)  -0.739049  0.606153 -0.293913          12.783855"
-matrix  "mol1-A mol2-B  U(2,.)   0.672677  0.687495 -0.273599          20.260256"
-matrix  "mol1-A mol2-B  U(3,.)   0.036221 -0.399912 -0.915838          42.593422"

# Query: mol1B
# No:  Chain   Z    rmsd lali nres  %id PDB  Description
   1:  mol2-B 14.4  1.8   99   124   39   MOLECULE: PROTEASE;
   2:  mol2-A 14.3  1.8   99   125   39   MOLECULE: PROTEASE;

# Structural equivalences
   1: mol1-B mol2-B     1 -   6 <=>     1 -   6   (PRO    1  - TRP    6  <=> LEU    1  - GLU    6 )
   1: mol1-B mol2-B     7 -  16 <=>     9 -  18   (GLN    7  - GLY   16  <=> ASP    9  - THR   18 )
   1: mol1-B mol2-B    17 -  35 <=>    29 -  47   (GLY   17  - GLU   35  <=> SER   29  - GLU   47 )
   1: mol1-B mol2-B    36 -  44 <=>    50 -  58   (MET   36  - PRO   44  <=> TRP   50  - GLU   58 )
   1: mol1-B mol2-B    45 -  59 <=>    62 -  76   (LYS   45  - TYR   59  <=> PRO   62  - SER   76 )
   1: mol1-B mol2-B    60 -  67 <=>    78 -  85   (ASP   60  - CYS   67  <=> ASP   78  - ILE   85 )
   1: mol1-B mol2-B    68 -  99 <=>    92 - 123   (GLY   68  - PHE   99  <=> GLU   92  - ASN  123 )
   2: mol1-B mol2-A     1 -   6 <=>     1 -   6   (PRO    1  - TRP    6  <=> LEU    1  - GLU    6 )
   2: mol1-B mol2-A     7 -  16 <=>     9 -  18   (GLN    7  - GLY   16  <=> ASP    9  - THR   18 )
   2: mol1-B mol2-A    17 -  35 <=>    29 -  47   (GLY   17  - GLU   35  <=> SER   29  - GLU   47 )
   2: mol1-B mol2-A    36 -  45 <=>    50 -  59   (MET   36  - LYS   45  <=> TRP   50  - ALA   59 )
   2: mol1-B mol2-A    46 -  59 <=>    63 -  76   (MET   46  - TYR   59  <=> GLN   63  - SER   76 )
   2: mol1-B mol2-A    60 -  67 <=>    78 -  85   (ASP   60  - CYS   67  <=> ASP   78  - ILE   85 )
   2: mol1-B mol2-A    68 -  99 <=>    92 - 123   (GLY   68  - PHE   99  <=> GLU   92  - ASN  123 )

# Translation-rotation matrices
-matrix  "mol1-B mol2-B  U(1,.)   0.765180  0.567071  0.304844          -6.633084"
```

Example of the Parseable data from a DALI pairwise alignment. It's worth noting that this alignment has multiple chains and therefore the specific chain alignments are separated. DALI will align each chain from the query (mol1) individually onto each individual chain on the target (mol2).

The parseable data is sorted by the query chain aligned. For this specific case we have two chains so we will have two alignments. You should always choose the best Z score with the longest length of alignment. However, they may not necessarily have the same alignment number for each chain so it's important to identify which chains are being aligned. For instance, if the best scoring alignment is from mol1-A to mol2-A, then we shouldn't take the alignment for mol1-B to mol2-A as this would not be an accurate alignment of the query.

For this case we need to copy the structural equivalences data that is boxed in above. There's no need to include anything outside of what is boxed in, but it is a

good idea to save the data for reference. Once copied, paste the data into a text file. This file will become the alignment file for input into BRAT. After copying the parseable data write “molecule 1:” and “molecule 2:” for the first and second lines of the file respectively. This is where you will determine the 4 character label to be used for the alignment. The labels must be the last 4 characters of the first and second line. The file does not have a required name or suffix in order to be used in BRAT. The file should look like the file below.

```
molecule 1: 1htg
molecule 2: 1bai

# Structural equivalences
   1: mol1-A mol2-A     1 -   6 <=>     1 -   6   (PRO    1  - TRP    6  <=> LEU    1  - GLU    6 )
   1: mol1-A mol2-A     7 -  16 <=>     9 -  18   (GLN    7  - GLY   16  <=> ASP    9  - THR   18 )
   1: mol1-A mol2-A    17 -  35 <=>    29 -  47   (GLY   17  - GLU   35  <=> SER   29  - GLU   47 )
   1: mol1-A mol2-A    36 -  45 <=>    50 -  59   (MET   36  - LYS   45  <=> TRP   50  - ALA   59 )
   1: mol1-A mol2-A    46 -  59 <=>    63 -  76   (MET   46  - TYR   59  <=> GLN   63  - SER   76 )
   1: mol1-A mol2-A    60 -  67 <=>    78 -  85   (ASP   60  - CYS   67  <=> ASP   78  - ILE   85 )
   1: mol1-A mol2-A    68 -  99 <=>    92 - 123   (GLY   68  - PHE   99  <=> GLU   92  - ASN  123 )
   1: mol1-B mol2-B     1 -   6 <=>     1 -   6   (PRO    1  - TRP    6  <=> LEU    1  - GLU    6 )
   1: mol1-B mol2-B     7 -  16 <=>     9 -  18   (GLN    7  - GLY   16  <=> ASP    9  - THR   18 )
   1: mol1-B mol2-B    17 -  35 <=>    29 -  47   (GLY   17  - GLU   35  <=> SER   29  - GLU   47 )
   1: mol1-B mol2-B    36 -  44 <=>    50 -  58   (MET   36  - PRO   44  <=> TRP   50  - GLU   58 )
   1: mol1-B mol2-B    45 -  59 <=>    62 -  76   (LYS   45  - TYR   59  <=> PRO   62  - SER   76 )
   1: mol1-B mol2-B    60 -  67 <=>    78 -  85   (ASP   60  - CYS   67  <=> ASP   78  - ILE   85 )
   1: mol1-B mol2-B    68 -  99 <=>    92 - 123   (GLY   68  - PHE   99  <=> GLU   92  - ASN  123 )
```

Example of the alignment file created from a DALI pairwise alignment.

Deciding how to handle binding residue assignment

As mentioned in section 2 of the Quick Guide, there are 4 methods to identify the binding residues (or any other residues of interest). In this tutorial we will use 2 methods of binding residue identification. These will be a user created comma separated values (csv) file and the residues within 4.5 Angstroms of the ligand G37 number 300 of chain A. We’ll start with the creation of the csv file which is just a text format with commas separating the residues. These residues can be any residues present in the query regardless of if they are binding residues or not. This option gives the possibility of identifying residues of interest such as metal coordinating atoms, mutations, etc.. For this case we’ll say we’re interested in the alpha helix of chain A. The residues in this helix are arginine87, asparagine88, leucine89, and leucine90. In order for BRAT to properly identify these residues it needs the 3 character residue name (e.g. ASN for asparagine), the chain ID (A in this case), and the residue number (88 for asparagine). Push them together to look like the entries below.

```
ARGA87,ASNA88,LEUA89,LEUA90
```

There's no need for any other information in the file. Now that the csv file is complete we need to identify what ligand we want to use to identify binding residues. This can be done using pdb, pdbsum, pymol, etc.; but for the sake of time I have provided the ligand info. Now that we know the ligand we can input the needed details for BRAT to identify it's binding residues. BRAT requires the ligand ID, chain, and ligand number in the pdb. For this ligand the input will look like "G37,A,300".

Running BRAT

Now that we've prepared all of our input we are set to run BRAT. BRAT is a standalone python script thus running it from a shell is easy. To run it in a shell you must use the python command with the brat_main.py file. BRAT has a help option "-h" that will display all of the input options. If you wan't a more in depth description of each option check out the Quick Guide. So let's give running BRAT a try. Your inputs should look like:

```
corona:~ bemiste1$ python ~/Binding_Residue_Alignment_Tool/brat.py -q ~/Binding_
Residue_Alignment_Tool/Input/sample_4/1htg.pdb -t ~/Binding_Residue_Alignment_To
ol/Input/sample_4/1bai.pdb -a ~/Binding_Residue_Alignment_Tool/Input/sample_4/1h
tg_v_1bai_DALI.txt -b ~/Binding_Residue_Alignment_Tool/Input/sample_4/1htg_helix
_residues.txt -l G37,A,300
```

For every sample in the package there is a typescript indicating the inputs used for each sample. When running sample 4 BRAT should print a message stating that the default radius is used and the locations of the output files. All together, you'll see:

```
corona:~ bemiste1$ python ~/Binding_Residue_Alignment_Tool/brat.py -q ~/Binding_
Residue_Alignment_Tool/Input/sample_4/1htg.pdb -t ~/Binding_Residue_Alignment_To
ol/Input/sample_4/1bai.pdb -a ~/Binding_Residue_Alignment_Tool/Input/sample_4/1h
tg_v_1bai_DALI.txt -b ~/Binding_Residue_Alignment_Tool/Input/sample_4/1htg_helix
_residues.txt -l G37,A,300


User has requested BRAT's radius identification of binding residues
but did not specify a radius. The default 4.5 Angstrom radius will
be used.


Binding residues written to /Users/bemiste1/Binding_Residue_Alignment_Tool/Input
/sample_4/1htg_v_1bai_DALI_brat_res.csv
Results written to /Users/bemiste1/Binding_Residue_Alignment_Tool/Input/sample_4
/1htg_v_1bai_DALI_brat_aligned.html
```

For any questions about the output from BRAT, please check the Quick Guide.

August 12, 2016