MODTIE users guide. v 1.11

Fred P. Davis
fredpdavis@gmail.com
http://fredpdavis.com/modtie

August 26, 2015

Abstract

MODTIE predicts binary protein interactions and higher-order protein complexes from a set of protein sequences based on their similarity to template complexes of known structure. This document describes how to install and run MODTIE.

Contents

This version of MODTIE (v1.11) uses templates from PIBASE version 2010, released September 2010 (`http://fredpdavis.com/pibase`).

Part I
Installing MODTIE

1 Downloading

1.1 Software

The MODTIE package is available at `http://fredpdavis.com/modtie`. Obtain the latest release and unpack this file in the directory you'd like to install it (*eg*, /usr/local/software/modtie)

```
mv modtie_v1.11.tar.gz /usr/local/software/modtie
cd /usr/local/software/modtie
tar xvfz modtie_v1.11.tar.gz
```

1.2 Data files

Four data files need to be downloaded separately:

1. `https://zenodo.org/record/29594/files/modtie_data.modtie_v1.11.tgz` (55 MB)
2. `https://zenodo.org/record/29594/files/pibase2010_subsets_residues.modtie_v1.11.tar` (519 MB)
3. `https://zenodo.org/record/29594/files/pibase2010_bdp_residues.modtie_v1.11.tar` (680 MB)

4. https://zenodo.org/record/29594/files/pibase2010_data.modtie_v1.11.tar (425 MB)

Move the first file into the main directory and uncompress it there:

```
cd /usr/local/software/modtie
tar xvfz modtie_data.modtie_v1.11.tgz
```

Then move the remaining 3 files into the new modtie_data directory and uncompress them there:

```
cd modtie_data
tar xvf pibase2010_data.modtie_v1.11.tar
tar xvf pibase2010_bdp_residues.modtie_v1.11.tar
tar xvf pibase2010_subsets_residues.modtie_v1.11.tar
```

2 Installing

2.1 Prerequisite: wget

GNU wget is used to retrieve files from the MODBASE webserver. Most linux machines have this installed by default. wget is available from http://www.gnu.org/software/wget/

2.2 Prerequisite: MODELLER

The SALIGN module of MODELLER (Sali and Blundell, *J Mol. Biol.* 1993) is used to calculate structural alignments of target-template domains. MODELLER is available at http://salilab.org/modeller.

2.3 Prerequisite: PDB mirror

A local mirror of the PDB is necessary. This is easy to setup using the instructions at the wwPDB: http://www.wwpdb.org/downloads.html.

You want the PDB formatted version of the PDB (not the mmCIF or XML versions). Specifically, the directory ending in structures/divided/pdb/.

2.4 Compiling C programs

Several C programs are used by the program, and binaries are provided for x86_64/o64 CPUs. If you use a different architecture, recompile these programs by running make in the src/auxil directory:

```
cd /usr/local/software/modtie/src/auxil
make
```

2.5 Specifying local configuration details

Edit modtie.pm to specify your local configuration details, lines 108-129.

- Specify the installation location:

```
$modtie_specs->{root} => '/usr/local/software/modtie/';
```

- Specify the directory to store retrieved and generated files:

```
$modtie_specs->{runroot} = '/MY/FAVORITE/RUN/PATH' ;
```

- Specify the PDB mirror location.

```
$modtie_specs->{pdb_dir} = '/PDB_DIRECTORY/structures';
```

- Specify the MODELLER binary location:

```
$modtie_specs->{modeller_bin} = '/MODELLER_PATH/mod9v4';
```

- Specify the SGE compute cluster. The lines of interest start with:

```
$modtie_specs->{cluster}->{OPTION} = VALUE ;
```

1. `head_node` - this is the name of the machine from which jobs can be submitted; make sure you have passwordless ssh setup between your local machine and the cluster head node. If you can submit directly from your local machine, set this to blank: "".
2. `cluster_mode` - if 1, uses a cluster by default. if 0, will not.
3. `qstat_sleep` - how often to qstat the jobs (sec)
4. `priority` - the job priority to use in the submission
5. `numjobs` - maximum number of compute nodes to use.

2.6 Place the perl library in your PERL5LIB path

Place the directory containing modtie.pm (src/perl_api) in your PERL5LIB environment variable:
For example, if you run a csh or tcsh shell, add this to your .cshrc file:

```
setenv PERL5LIB {$PERL5LIB}:/usr/local/software/modtie/src/perl_api
```

For a bash shell, add this to your .bashrc:

```
PERL5LIB=/usr/local/software/modtie/src/perl_api:$PERL5LIB
export PERL5LIB
```

Part II
Running MODTIE

3 `runmodtie.modbase.pl` - Predict interactions between protein sequences

This program predicts interactions between sets of proteins using homology models built and stored in MODBASE `http://salilab.org/modbase`. This is useful for making genome-wide predictions of intra- or inter-species protein interactions.

3.1 What does this program do exactly?

This program performs the following steps, as indicated by the `message that the program prints to STDERR`:

1. Retrieve model information from MODBASE (∼30min for 3000 sequences)
2. Assign domain boundaries and classification to models (SGE parallel): `Running assign_model_domains`
3. Assign domains to sequences, by tiling together model domains (fast): `Running assign_seqid_domains`
4. Determine first residue number in each model PDB file (fast): `Running calc_model_pdbstart`
5. Compute domain architecture strings for each sequence (fast): `Running assign_seqid_domarch`
6. Generate initial candidate interaction list. Given model domains and template complexes, determine what candidate interactions are possible, and correspondingly what target domain PDB files must be generated and what pairs of target and template domains must be aligned. (fast): `Running candidate generator (ali, cut lister)`
7. Cut target domains out of model PDB files. (SGE parallel): `Cutting up target domains`
8. Align target to template domains (SGE parallel): `Running alignments`
9. Re-generate candidate interaction list taking into account the alignment results (fast): `Re-running candidate generator`
10. Score candidate binary interactions using the target-template alignments and the structure of the template complex. (SGE parallel): `Assessing binary interactions`
11. Determine higher-order complexes that are possible given the binary interactions that passed the filter above. (fast): `Building higher-order complexes`

3.2 Preparing input

The input to the program is one or two lists of MODBASE sequence identifiers (seq_id). There are a couple ways to get this:

1. Pre-built MODBASE models: MODBASE attempts to model most protein sequences available in public databases, so your sequences of interest are most likely in the database. You can convert database identifiers (*eg*, UniProt, ENSMBL) to seq_id identifiers at the MODBASE website: `https://modbase.compbio.ucsf.edu/modbase-cgi/sequence_utility.cgi`.
2. Fresh MODBASE models: To build fresh models for you sequences, make an account on the ModWeb server (`http://salilab.org/modweb`) and submit your sequences. The ModWeb server uses a cookie to authenticate your account; this cookie is necessary for MODTIE to access the resulting models. Grep the line containing modbase.compbio.ucsf.edu and user_name from your web browser's cookie file. For example, from firefox:

```
grep modbase.compbio.ucsf.edu
   .mozilla/firefox/6yf4vf8n.default/cookies.txt | grep
   user_name > my_modweb_cookie.txt
```

Store this line in a text file and specify its location with the `-modbase-cookies` option.

3.3 Running

To predict interactions among one set of proteins:

```
perl runmodtie.modpiperun.pl -seqid_set seqid_list.txt
```

To predict interactions between two sets of proteins - *eg*, for host–pathogen interactions:

```
perl runmodtie.modpiperun.pl -seqid_set1 seqid_list1.txt -seqidset2
   seqid_list2.txt
```

3.4 Options

- `-cluster_fl` <0|1> (optional)

  if 1 will run the query in parallel using an SGE computing cluster, as specified in modtie.pm. (Defaults to 1, cluster mode on).

- `-run MODPIPE_RUNNAME` (optional)
  `-run1 MODPIPE_RUNNAME` (optional)
  `-run2 MODPIPE_RUNNAME` (optional)

  if specified, will restrict the MODBASE data to a specific MODPIPE run: *eg*, human_2008

  use -run for intra-set runs; use -run1 and -run2 for inter-set runs

- `-modweb_cookie COOKIEFILE` (optional)

  Necessary to access MODBASE files generated by a ModWeb server submission. See **Preparing input**, above.

3.5 Options to restart incomplete runs

MODTIE stores output at the end of each step. In case a run dies before completion, it can be restarted using the results of the completed steps, so that these don't have to be rerun.

- `-seqid_setinfo seqid_setinfo_XXXXX.modtie` - sequence set information
- `-model_list model_list_XXXXX.modtie` - model information retrieved from MODBASE
- `-model_domains model_domains_out_XXXX.modtie` - model domain assignments
- `-seqid_domains seqid_domains_out_XXXX.modtie` - sequence domain assignments
- `-model_pdbstart model_pdbstart_out_XXXX.modtie` - model PDB files' first residue numbers
- `-seqid_domainarch seqid_domainarch_out_XXXX.modtie` - sequence domain architecture strings
- `-cutlist cutlist_out_XXXXX.modtie` - list of target domains to extract from model PDB files
- `-cuts_done 1` - specify if the domain cutting step finished properly

- `-alilist alilist_out_XXXXX.modtie` - list of alignments to perform
- `-ali_done 1` - specify if the alignment step finished properly
- `-postali_candilist postali_candilist_out_XXXXX.modtie` - candidate list of target domain interactions after considering alignment quality
- `-binscores binscores_out_XXXXX.modtie` - scoring results for candidate binary interactions

For example, I had to kill a run during the "Running candidate generator" step. The run was originally started with the following command:

```
perl /MODTIEPATH/runmodtie.modbase.pl -seqid_set
   get_mtuber_seqid.out -run mtuber 2>run_mtuber.$$.err
   >run_mtuber.$$.out
```

To restart the run, I specified all the intermediate files from the steps that had completed:

```
perl /MODTIEPATH/runmodtie.modbase.pl -seqid_set
   get_mtuber_seqid.out -run mtuber -model_domains
   model_domains_out_gCwr9.modtie -model_list
   model_list_0z00A.modtie -seqid_domainarch
   seqid_domainarch_out_klhZw.modtie -seqid_domains
   seqid_domains_out_z2R9Q.modtie -seqid_setinfo
   seqid_setinfo_2FHeR.modtie 2>run_mtuber.$$.err >run_mtuber.$$.out
```

3.6 Options for inter-set (host-pathogen) runs

The restart options are also useful for host–pathogen runs. For example, if the domain assignments were previously calculated during intra-set runs for the host and pathogen proteins, the corresponding files can be concatenated with a simple 'cat', and the location of the new merged file specified by the options above, *eg*, `-model_domains`. This would prevent redundant calculation of the domain assignments.

3.7 Output

All of the output files have header lines that describe the contents. The most useful output files are probably (1) the binscores file describing binary interactions and (2) the complexes file describing higher order complexes.

3.7.1 Binscores

Tab-delimited fields:

1. INTERFACE
2. template domain 1
3. template domain 2
4. seq_id 1

5. model_id 1
6. residue range 2
7. SCOP family domain 1
8. seq_id 2
9. model_id 2
10. residue range 2
11. SCOP family domain 2
12. number of template contacts
13. number of contacts aligned to target domains
14. statistical potential location
15. statistical potential type
16. statistical potential details
17. statistical potential distance cutoff
18. raw score
19. Z-score
20. Z-prime = Z-score - lowest Z-score of a true negative
21. Z-2 = Z-score - lowest observed Z-score
22. average raw score
23. lowest raw score
24. lowest raw score of a true negative
25. standard deviation of raw score
26. lowest Z-score
27. lowest Z-score of a true negative
28. false positives
29. RMSD of alignment between target and template domain 1
30. Number of equivalent positions between target and template domain 1
31. Number of residues in template domain 1
32. Number of residues in target domain 1

33. Number of residues identical between target and template domain 1
34. Number of template residues in binding site 1
35. Number of target residues in binding site 1
36. Number of residues identical between target and template binding site 1
37. RMSD of alignment between target and template domain 2
38. Number of equivalent positions between target and template domain 2
39. Number of residues in template domain 2
40. Number of residues in target domain 2
41. Number of residues identical between target and template domain 2
42. Number of template residues in binding site 2
43. Number of target residues in binding site 2
44. Number of residues identical between target and template binding site 2

3.7.2 Complexes

Two kinds of lines are generated: those that start with '#' that contain a summary of the complex, and the other lines that describe the details of the components.

Summary lines:

1. #compl
2. cid - numerical identifier unique to each complex
3. number of subunits (target domains)
4. average score of interfaces in the complex
5. maximum score of interfaces in the complex
6. list of template domains
7. original_cids - list of the original complex CID if this complex is a merger of other complexes.

Detail lines:

1. cid
2. subunit number
3. seq_id
4. model_id

5. resdue range
6. template domain
7. SCOP family
8. average score of interfaces involving this subunit
9. maximum score of interfaces involving this subunit

4 `runmodtie.targetstrxs_template.pl` - Score interaction between two protein structures given a template complex structure

This program scores the putative interaction between two target domains based on a template complex. It first calls MODELLER to align the target structures onto the corresponding domains of the template complex, calculates the putative interface contacts, and then scores them using the MODTIE statistical potential.

4.1 Input

Tab-delimited input specified on STDIN; each line specifies a target or template domain.

1. Complex name
2. 'template' or 'target'
3. Domain name - have to use same name for corresponding domain in template and target structures
4. PDB file name
5. Start residue number (leave blank to use the chain start)
6. End residue number (leave blank to use the chain end)
7. Chain identifier (leave blank if no chain identifier)

If the domain has multiple fragments, describe them in additional fields resembling 5-7. For example, fields 8-10 would describe the start/end/chain of the second domain fragment.

4.2 Output

Results are displayed to a file called `complexscores.XXXXX.out.modtie`, where `XXXXX` is a random string. Two kinds of lines are generated: COMPLEX lines describing the overall scores of a complex, and INTERFACE lines describing individual domain–domain interfaces.

The lines are tab-delimited with the following fields:
COMPLEX lines:

1. COMPLEX
2. complex_id

3. list of domains
4. statistical potential location
5. statistical potential type
6. statistical potential details
7. statistical potential distance cutoff
8. raw score
9. Z-score
10. Z-prime = Z-score - lowest Z-score of a true negative
11. Z-2 = Z-score - lowest observed Z-score
12. average raw score
13. lowest raw score
14. lowest raw score of a true negative
15. standard deviation of raw score
16. lowest Z-score
17. lowest Z-score of a true negative
18. False positives

INTERFACE lines:

1. INTERFACE
2. complex_id
3. domain 1
4. domain 2
5. statistical potential location
6. statistical potential type
7. statistical potential details
8. statistical potential distance cutoff
9. raw score
10. Z-score
11. Z-prime = Z-score - lowest Z-score of a true negative

12. Z-2 = Z-score - lowest observed Z-score
13. average raw score
14. lowest raw score
15. lowest raw score of a true negative
16. standard deviation
17. lowest Z-score
18. lowest Z-score of a true negative
19. False positives

5 `runmodtie.scorecomplex.pl` - Score the structure of a protein complex

This program takes as input a PDB file and domain definitions. It then scores each domain–domain interface in the file.

5.1 Input

Tab-delimited input provided through STDIN specifies the domain definitions:

1. PDB file name
2. Domain identifier
3. Start residue number (leave blank if at the chain start)
4. End residue number (leave blank if at the chain end)
5. Chain identifier (leave blank if no chain identifier)

To describe additional domains, or domain fragments, you can either add additional lines, or repeat fields resembling 2-5 (domain identifier/start/end/chain) at the end of the line.

5.2 Options

- `-out_scores_fn OUTPUT_FILE`

  Use this option to specify the output file name. By default, the results are displayed to a file named as `complexscores.XXXXX.out.modtie`, where `XXXXX` is a random string.

5.3 Output

Results are displayed to a file called `complexscores.XXXXX.out.modtie`, where `XXXXX` is a random string. Two kinds of lines are generated: COMPLEX lines describing the overall scores of a complex, and INTERFACE lines describing individual domain–domain interfaces.

The lines are tab-delimited with the following fields:
COMPLEX lines:

1. COMPLEX
2. PDB file name
3. list of domains
4. statistical potential location
5. statistical potential type
6. statistical potential details
7. statistical potential distance cutoff
8. raw score
9. Z-score
10. Z-prime = Z-score - lowest Z-score of a true negative
11. Z-2 = Z-score - lowest observed Z-score
12. average raw score
13. lowest raw score
14. lowest raw score of a true negative
15. standard deviation of raw score
16. lowest Z-score
17. lowest Z-score of a true negative
18. False positives

INTERFACE lines:

1. INTERFACE
2. PDB file name
3. domain 1
4. domain 2

5. statistical potential location
6. statistical potential type
7. statistical potential details
8. statistical potential distance cutoff
9. raw score
10. Z-score
11. number of interface contacts
12. Z-prime = Z-score - lowest Z-score of a true negative
13. Z-2 = Z-score - lowest observed Z-score
14. average raw score
15. lowest raw score
16. lowest raw score of a true negative
17. standard deviation of raw score
18. lowest Z-score
19. lowest Z-score of a true negative
20. False positives

6 Citing

- For interaction predictions within a single species, or to score protein complexes, please cite:
  Davis FP, Braberg H, Shen MY, Pieper U, Sali A, Madhusudhan MS. *Nucleic Acids Res* (2006) 34:2943-2952.
- For cross-species predictions, such as host–pathogen interactions, please cite:
  Davis FP, Barkan DT, Eswar N, McKerrow JH, Sali A. *Protein Sci* (2007) 16:2585-2596.