# MODTIE developers guide. v 1.11

Fred P. Davis, HHMI-JFRC
davisf@janelia.hhmi.org
http://pibase.janelia.org/modtie

October 7, 2010

## Introduction

MODTIE is a program that predicts binary and higher-order interactions among a set of protein sequences, based on similarity to template complexes of known structure. The routines to implement the method, benchmark it, and use it to make small-scale and genome-wide predictions are stored in a Perl library and called from short driver scripts. Here I describe the data files used by the program, and the layout of the core routines.

## Contents

**1 Data files** **1**
1.1 Static . . . 1
1.2 Dynamic . . . 2
1.3 Generated . . . 2

**2 Code layout** **2**
2.1 Run routines . . . 3
2.2 Core prediction routines . . . 3
2.3 Core method implementation routines . . . 3
2.4 Ancillary run modes . . . 4

## 1 Data files

### 1.1 Static

1. Statistical potentials (modtie_data/potentials)
2. Template interface list (modtie_data/templates)

3. PIBASE tables (modtie_data/pibase_tod, modtie_data/pibase_metatod)
4. Template domain PDB files (modtie_data/pibase_data)

### 1.2 Dynamic

These files are obtained through the MODBASE webserver as needed during a MODTIE run, and stored permanently:

1. Model PDB files. stored in location specified in modtie.pm:

   ```
   $modtie_specs->{local_modbase_models_dir}
   ```

2. Model alignment files

   ```
   $modtie_specs->{local_modbase_ali_dir}
   ```

Alternatively, these files can also be obtained through a local installation of the MODBASE mysql database (specify `-modbase_access local`). By default, the files are obtained via the website.

### 1.3 Generated

The following files are generated and stored during a MODTIE run, and can be reused in future runs:

- Target domains - PDB files containing the domains assigned in the target sequences. location specified in modtie.pm:

  ```
  $modtie_specs->{target_domains_dir}
  ```

- Target-template domain alignments - MODELLER SALIGN results for structural alignment of target and template domains. location specified in modtie.pm:

  ```
  $modtie_specs->{salign_ali_dir}
  ```

## 2 Code layout

The core routines are implemented in a Perl library (src/perl_api):

- modtie.pm - core prediction routines - I/O, processing, run logic
- SGE.pm - routines to interact with SGE compute cluster
- modtie/pibase.pm - routines to interact with PIBASE data files; most reused from PIBASE.
- modtie/complexes.pm - routines to predict higher-order complexes
- modtie/potential.pm - routines to build/use statistical potential
- modtie/yeast.pm - routines to assess yeast predictions

Several additional programs (src/auxil) are used to interact with PDB files, all part of the original PIBASE package. source code and o64 binaries are provided for the C programs.

- altloc_check - C program that checks if a PDB file has any alternative location field specified
- altloc_filter - Perl script that removes alternative locations from a PDB file.
- kdcontacts - C program that computes interatomic distances using a kd-trees algorithm
- subset_extractor - C program that extracts a chain/residue range from a PDB file

I describe the core modtie.pm routines and their functionality below.

## 2.1 Run routines

1. **runmodtie_modbase()** - Predicts inter- or intra-set interactions using homology models deposited in MODBASE. This routine requires either local or remote (default) access to model information, PDB files, and alignment files stored in MODBASE (`http://salilab.org/modbase`).
2. **runmodtie_scorecomplex()** - Scores the PDB file of a protein complex using the MODTIE statistical potential and domains defined in the input
3. **runmodtie_targetstrxs_template()** - Scores a putative complex given PDB files of the individual components and a template complex.

## 2.2 Core prediction routines

1. **model_2_domains()** - Assign SCOP domains to MODBASE models.
2. **seqid_2_domains()** - Assign SCOP domains to MODBASE sequences.
3. **seqid_2_domainarch()** - Compute SCOP domain architecture for MODBASE sequences.
4. **scoring_main()** - Main prediction routine: identifies candidate complexes, performs necessary alignments, and scores candidate complexes using the MODTIE statistical potentials.
5. **salign_targ_tmpl_domains()** - Uses the MODELLER SALIGN routine to align target and template domains.
6. **cut_domains()** - Extracts target domains from model PDB files.

## 2.3 Core method implementation routines

1. **extract_required_pibase_datafiles()** - Get the required datafiles (subsets_residues table-on-disk, bdp_residues table-on-disk, template domain PDB files).

   ```
   perl -e 'use modtie; modtie::extract_required_pibase_datafiles();' > pibase_datafile.log
   ```

2. **generate_interface_list()** - Generates template interface and interface cluster assignment lists by querying pibase.scop_interface_clusters.

```
perl -e 'use modtie; modtie::generate_interface_list();'
```

3. **buildpotential_count()** - Calculates and counts interatomic contacts to populate the statistical potential. The current statistical potential was built from interfaces in PIBASE v2005 with at least 1000 interatomic contacts at a distance threshold of 6.05 Å.
4. **buildpotential_postcalc()** - Builds the statistical potential from the contact counts.
5. **runmodtie_benchmark()** - Benchmarks the statistical potentials using complexes of known structure.
6. **runmodtie_roc()** - Builds receiver-operator curves characterizing statistical potential accuracies.

## 2.4 Ancillary run modes

1. **format_modbase_output()** - Format predicted complex results files for import into MODBASE.
2. **assess_yeast_results()** - Routines to filter yeast predictions using functional annotation, subcellular localization, and to benchmark the predictions against known complexes in MIPS, BIND, and Cellzome.