
CryptoSite Documentation

Release master

Peter Cimermancic, Ben Webb

May 29, 2020

Contents

1	Installation	3
1.1	In the Sali Lab	3
1.2	Dependencies	3
1.3	Building	4
2	Basic usage	5
2.1	Overview	5
3	Indices and tables	7

CryptoSite is a computational tool for predicting the location of cryptic binding sites in proteins and protein complexes.

Contents:

1.1 In the Sali Lab

If you are working in the Sali lab, you don't need to build and install CryptoSite - it is already set up for you as a module. Just run `module load cryptosite` to load it.

1.2 Dependencies

All dependencies listed below are expected to be found in standard system paths. This may require setting `PYTHONPATH`, `PATH` and/or `LD_LIBRARY_PATH` environment variables. Note that Linux is the only platform for which all these dependencies are available, and so is the only platform on which CryptoSite currently functions.

- Python 2.6 or later (Python 3 should also be OK).
- MODELLER plus the SOAP-Protein library. The `soap_protein_od.hdf5` file needs to be placed into MODELLER's `modlib` directory.
- MUSCLE.
- DSSP. It is expected that the `mkdssp` binary is in the `PATH`.
- fpocket (version 2).
- PatchDock.
- IMP.
- NCBI BLAST+ plus a local copy of the UniProt database for it to search against. To make this local database, gunzip the `uniprot_trembl.fasta.gz` and `uniprot_sprot.fasta.gz` files available from the UniProt website and combine them into a single `uniprot` text file.
- USEARCH version 8.1 or later.
- Biopython.
- NumPy and SciPy.

- `scikit-learn`. Note that precisely version 0.12 is needed - other versions won't work.
- `AllosMod` is needed to run part of the protocol.
- `nose` is also needed to run the test suite (recommended but not essential).

In the Sali lab, running `module load modeller muscle dssp fpocket patch_dock imp blast+ usearch` will get all of these dependencies.

1.3 Building

Use `make PYTHON=python3` or `make PYTHON=python2` to build the library (depending on which version of Python you want to use). Use `make test` to test the library, and `make install` to install it. In most cases you will need to tell `make` where to install (if running on a Linux cluster, CryptoSite will need to be installed on a network-accessible filesystem) and where your local copy of UniProt is, with something like `make PREFIX=/shared/cryptosite UNIPROT=/database/uniprot install`. See `Makefile.include` for all `make` variables that can be configured.

Once installed, the CryptoSite protocol can be run by means of a command line tool (`cryptosite`). Each component of the protocol is also a Python package, which can be called directly from other Python software (via `import cryptosite`).

2.1 Overview

Running the basic protocol consists of these steps:

1. Create a set of input files specifying the structure to probe, and basic CryptoSite parameters.
2. Run `cryptosite setup` to calculate structural features and prepare input files for `AllosMod`.
3. Run `AllosMod` given the set of input files.
4. Calculate additional features using the `AllosMod` output (`cryptosite soap`, `cryptosite pockets`, `cryptosite am_bmi`, `cryptosite analysis`).
5. Gather together all features into a single file (`cryptosite gather`).
6. Predict cryptic binding sites with an SVM using the complete set of features as input (`cryptosite predict`).
7. Optionally visualize the results in `Chimera` (`cryptosite chimera`).

CHAPTER 3

Indices and tables

- `genindex`
- `modindex`
- `search`