
Tangram Suite Documentation

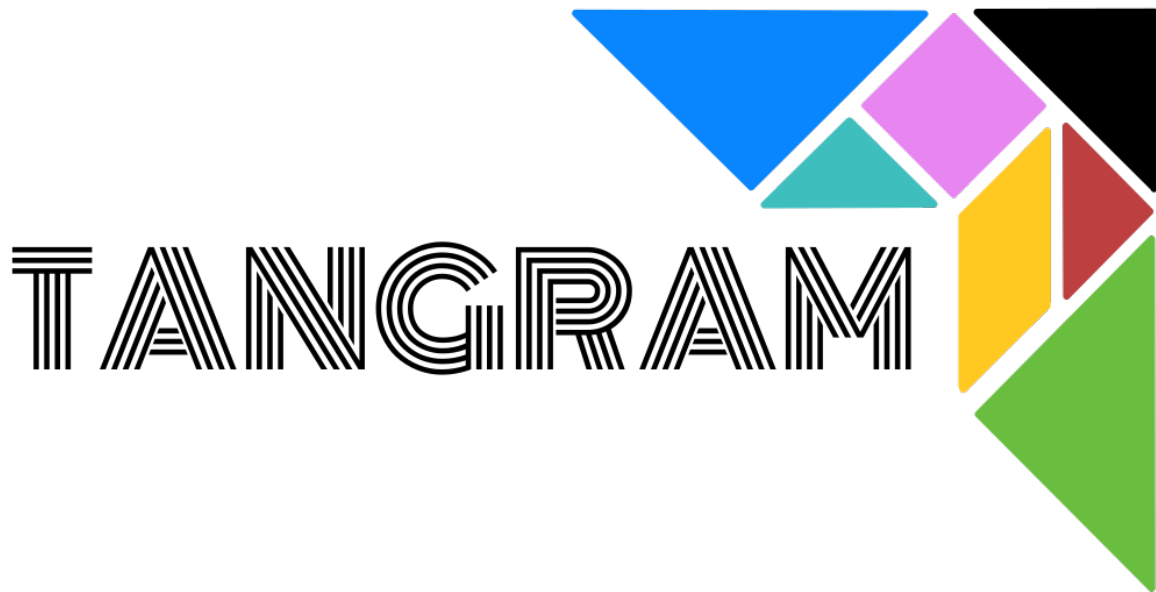
Release 0.0.8

InsiliChem

Mar 25, 2019

General instructions

1	What is Tangram Suite	3
2	How to install Tangram Suite	5
3	General usage	7
4	FAQ & Known issues	9
5	List of Tans	11
6	Tangram BondOrder	13
7	Tangram QMSetup	15
8	Tangram DummyAtoms	17
9	Tangram GAUDIView	19
10	Tangram MMSetup	21
11	Tangram NCIPLOT	23
12	Tangram NormalModes	25
13	Tangram OrbiTraj	27
14	Tangram PLIP	29
15	Tangram PoPMuSiCGUI	31
16	Tangram PropKaGUI	33
17	Tangram ReVina	35
18	Tangram SubAlign	37



It's composed of several independent graphical interfaces and commands for UCSF Chimera.

CHAPTER 1

What is Tangram Suite

In progress

How to install Tangram Suite

2.1 Install the full Tangram Suite (recommended)

1 - If you don't have UCSF Chimera installed, download the [latest stable copy](#) and install it with:

```
chmod +x chimera-*.bin && ./chimera-*.bin
```

2 - Download the latest installer from the [releases page](#) (Linux and MacOS only) and run it with:

```
bash tangram-*.sh
```

2.2 Using the conda meta-package

Instead of using the Bash installer, you can use conda (if you are already using it) to create a new environment with the `tangram` metapackage, which will handle all the dependencies. While in alpha, both `insilichem` channels are needed (the main one and also the `dev` label):

```
conda create -n tangram -c insilichem/label/dev -c insilichem -c conda-forge -c omnia_
↳-c rdkit tangram
```

2.3 Updating extensions

Each extension will check if there's a new release available every time you launch it. To update it, the installer provides you with a command that you can run with the `conda` environment already activated:

```
conda update -c insilichem/label/dev -c insilichem [-c additional channels]
↳<extension_name>
```

Note: More `-c` flags might be needed, depending on the requirements. Check each extension documentation page to see the needed conda channels.

For example, if you want to update *gaudiview*, you would write:

```
conda activate insilichem
conda update -c insilichem/label/dev -c insilichem gaudiview
```

2.4 Install only one specific extension

(Advanced users only)

Installing separate extension is not recommended but can be done.

```
conda install -c insilichem <package name>
```

Note: More `-c` flags might be needed, depending on the requirements. Check each extension documentation page to see the needed conda channels.

Once completed, tell UCSF Chimera to look for new extensions in the chosen environment. To do that, open UCSF Chimera and go to `Favorites> Add to Favorites/Toolbar`. In the newly opened dialog, specify the env location in the bottom box. In this case, the extensions location will be something like `$CONDA_PREFIX/lib/python2.7/site-packages`.

Note: The conda environment will always require `pychimera` to inject the conda dependencies.

CHAPTER 3

General usage

Once installed (check *Install the full Tangram Suite (recommended)*), some of the extensions will run without additional dependencies, which means they will be ready to use in your normal UCSF Chimera installation. However, more complex ones require 3rd party libraries (already provided with this installer, though!). To run these tricky ones, you must activate the `conda` environment that was created during the installation process. The installer will provide the needed steps, but you can also find them here (`$PREFIX` is the install location, `~/tangram` by default):

```
(...)  
Done! InsiliChem Tangram is now installed in your system.  
  
Some of the extensions will work out the box, but some others will  
require two (or three) additional steps:  
  
0) ONLY ONCE! If you haven't used conda before, you will have to run  
this command to enable Tangram's one:  
  
    echo ". $PREFIX/etc/profile.d/conda.sh" >> ~/.bashrc  
  
1) Activate Tangram environment:  
  
    conda activate $PREFIX  
  
2) Launch a patched UCSF Chimera instance with:  
  
    tangram # available in Bash and Fish  
    # use `pychimera --gui` in other shells  
  
(...)
```


CHAPTER 4

FAQ & Known issues

Tangram needs lots of external dependencies to run properly, and having them run on UCSF Chimera can be tricky if not done properly. To do that, we have built `pychimera`, on which Tangram is built. Before [raising an issue](#), please have a look at the [known issues](#) in its documentation.

- **Calculation setup**

- **MMSetup**: Setup MD calculations with OpenMM and ommprotocol [WIP]
- **QMSetup**: QM and QM/MM calculations setup

- **Visualization**

- **3D-SNFG**: Enable easy visualization of saccharydic residues
- **BondOrder**: Automatic bond order perception for UCSF Chimera [WIP]
- **GAUDIView**: Lightweight visualization of results coming from docking, conformational search or multi-objective optimization

- **Analysis**

- **NCIPlotGUI**: Straightforward interface to setup calculations for NCIPlot and visualize them
- **NormalModes**: Perform Normal Modes Analysis and view them directly on-screen
- **PLIPGUI**: Depict protein-ligand interactions, as calculated with PLIP
- **PoPMuSiCGUI**: Depict and apply the predictions made by PoPMuSiC calculations
- **PropKaGUI**: Analyze and depict the expected pKa values of protein residues with PropKa 3.1
- **SubAlign**: Align two, potentially different, molecules based on partial matches of substructures

- **Utilities & Patches**

- **DummyMetal**: A subtle modification to UCSF Chimera's MetalGeom extension to allow arbitrary elements to be placed at vacant positions, instead of just oxygens
- **OrbiTraj**: A subtle modification to UCSF Chimera's MD Movie extension to allow the visualization of volumetric data along a molecular trajectory
- **ReVina**: Resubmit failed AutoDock Vina jobs without reconfiguring the GUI

Tangram BondOrder

Automatic bond order perception for UCSF Chimera [WIP]

6.1 Features

While UCSF Chimera does have some kind of bond order perception that supports its internal atom typing engine (see [\[Chimera-users\] bond order](#)), this information is not available to the user (not even in the Python API). This extension tries to fill the gap by providing a graphic interface to perceive bond order with external programs and/or set them manually.

6.2 Usage

[WIP]

This extension is still under development. Documentation will be ready once basic functionality is achieved.

6.3 Requirements

- libtangram
- rdkit
- ambermini

```
conda install -c insilichem/label/dev -c insilichem -c rdkit -c omnia tangram_  
↳bondorder
```


QM and QM/MM calculations setup

7.1 Features

This extension will allow you to create Gaussian input files for QM and ONIOM (QM/MM) calculations. It's designed with simplicity in mind, but that does not mean the feature set is small:

- Configure any QM job allowed by Gaussian
- Customize the basis sets thanks to the included data from [BSE](#).
- Specify ModRedundant in Opt jobs
- Add any custom keywords directly in the GUI
- Set QM/MM jobs with ONIOM. The layers can be set easily with a dedicated dialog.
- Rapidly write replicas of the input file for different conformers/frames of your system/trajectory.

7.2 Usage

[WIP]

7.3 Requirements

- `libtangram` (*pip*-installable)

Note: `conda install -c insilichem/label/dev -c insilichem tangram_qmsetup`

Tangram DummyAtoms

Apply the [Cationic Dummy Atom](#) approach to parametrize metal-containing systems using the Ambertools suite

8.1 Features

- The extension will generate PRMTOP and INPRCD files for your system, compatible with Amber & OMMProtocol/OpenMM.
- Multiple atom sites allowed seamlessly.
- If this extension is installed, a reversible, subtle modification to UCSF Chimera's MetalGeom extension will allow arbitrary elements to be placed at vacant positions, instead of just oxygens.

8.2 Usage

[WIP]

8.3 Requirements

- libtangram, pychimera
- ambermini

```
conda install -c insilichem/label/dev -c insilichem -c rdkit -c omnia tangram_  
↳ dummyatoms
```

Note: This package might need that you launch UCSF Chimera with `pychimera --gui`

Tangram GAUDIView

Lighweight visualization of results coming from docking, conformational search or multiobjective optimization

9.1 Features

- Explore **GaudiMM** and **GOLD** results in the same GUI
- Lightweight and fast, it will lazy-load the molecules upon request
- Builtin commandline will perform any Chimera command upon row change (use the mouse or the keyboard arrows to move)
- Basic clustering functionality
- Use `Ctrl` and `Shift` to create multiple selections
- **Double-click to apply special modifications, depending on the job type**
 - If the **GOLD** job used rotamers, these will be applied on double-click
- Quit the dialog with `OK` to preserve the currently selected solutions on the Chimera canvas.

9.2 Usage

[WIP]

9.3 Requirements

- `libtangram`, `tkintertable`, `pyyaml`

```
conda install -c insilichem/label/dev -c insilichem tangram_bondorder
```


CHAPTER 10

Tangram MMSetup

Setup MD calculations with OpenMM and ommprotocol [WIP]

10.1 Features

[WIP]

10.2 Usage

[WIP]

10.3 Requirements

- libtangram, psychimera
- ommprotocol

```
conda install -c insilichem/label/dev -c insilichem -c omnia tangram_mmsetup
```

Note: This package might need that you launch UCSF Chimera with `psychimera --gui`

Straightforward interface to setup calculations for NCIPlot and visualize them

11.1 Features

[WIP]

11.2 Usage

[WIP]

11.3 Requirements

- `libtangram`, `pychimera`
- `nciplot`
- Optional: `cunci`, a CUDA-ready implementation of NCIPlot that runs faster.

```
conda install -c insilichem/label/dev -c insilichem -c tangram_nciplot
```

Note: This package might need that you launch UCSF Chimera with `pychimera --gui`

Tangram NormalModes

Perform Normal Modes Analysis and view them directly on-screen

12.1 Features

[WIP]

12.2 Usage

[WIP]

12.3 Requirements

- libtangram, prody, pychimera

```
conda install -c insilichem/label/dev -c insilichem tangram_normalmodes
```

Note: This package might need that you launch UCSF Chimera with `pychimera --gui`

A subtle modification to UCSF Chimera's MD Movie extension to allow the visualization of volumetric data along a molecular trajectory

13.1 Features

[WIP]

13.2 Usage

[WIP]

13.3 Requirements

- `libtangram` (*pip-installable*)

```
conda install -c insilichem/label/dev -c insilichem tangram_orbitraj
```


Depict protein-ligand interactions, as calculated with PLIP

Note: This package is not currently compatible with the Tangram suite due to incompatibilities with `rdkit` and `openbabel`. It should be installable separately, though.

14.1 Features

[WIP]

14.2 Usage

[WIP]

14.3 Requirements

- `libtangram`, `pychimera` (*pip*-installable)
- `openbabel` (*conda*-installable with `conda install -c openbabel openbabel`)

```
conda install -c insilichem/label/dev -c insilichem -c openbabel -c plip tangram_plip
```

Note: This package might need that you launch UCSF Chimera with `pychimera --gui`

Depict and apply the predictions made by PoPMuSiC calculations

15.1 Features

[WIP]

15.2 Usage

[WIP]

15.3 Requirements

- `libtangram` (*pip*-installable)

Note: `conda install -c insilichem/label/dev -c insilichem tangram_popmusicgui`

Analyze and depict the expected pKa values of protein residues with PropKa 3.1

16.1 Features

[WIP]

16.2 Usage

[WIP]

16.3 Requirements

- libtangram
- PropKa-3.1

Note: `conda install -c insilichem/label/dev -c insilichem tangram_propkagui`

Resubmit failed AutoDock Vina jobs without reconfiguring the GUI

17.1 Features

[WIP]

17.2 Usage

[WIP]

17.3 Requirements

- `libtangram` (*pip*-installable)

Note: `conda install -c insilichem/label/dev -c insilichem tangram_vinarelaunch`

Align two, potentially different, molecules based on partial matches of substructures

18.1 Features

[WIP]

18.2 Usage

In general:

```
subalign <reference> <probe> [methods <method>]
```

- `reference`, `probe`: must be Chimera selections (e.g. #0 or sel).
- **methods (optional): the alignment method(s) to use. If you specify several ones (separate by commas), `subalign` will try them in order until one works.**
 - `fast`: identify the best submatch of the naive one-to-one corresponding atom pairs.
 - `best` (default): iterate over all possible atom pair correspondences, trying `fast` over each one, and returns the one with best RMSD.
 - `o3a`: identifies atom pair correspondences by MM-like atom typing.
 - `com`: simply overlap centers of mass (great as a fallback method; e.g. `methods best, com`)

Note: The passed Chimera selections must correspond to a single molecule. If you only specify a subset of atoms, it will be expanded to the containing molecule. With protein-ligand systems, this can be costly. The current recommendation is to split the molecule in submodels beforehand with `split` (see [Chimera docs](#)) and then specify the desired submodel.

18.3 Requirements

- `libtangram`, `pychimera`
- `rdkit`

Note: `conda install -c insilichem/label/dev -c insilichem -c rdkit tangram_subalign`

Note: This package might need that you launch UCSF Chimera with `pychimera --gui`
