Useful Scripts

xml2png.py

Converts xml data file output from the Tecan Infinite m1000 Pro plate reader and allows for the quick visual inspection of raw absorbance and fluorescence data.

```
$ python python xml2png.py *.xml
```

quickmodel.py

Builds quick Bayesian model of both spectra and single wavelength two component binding experiments.

As input, it requires xml output files of the experiment form plate reader and a python script(*inputs.py*) that includes all experimental design details.

1. Run *calculate_L_stated_array.py* to generate ligand concentration array and copy it into *inputs.py*.
2. Construct *inputs.py* script based on experimental design.
3. Run`quickmodel.py`.

```
$ python quickmodel.py --inputs 'inputs' --type 'singlet' --nsamples 10000
```

inputs.py

inputs.py should be manually constructed to record experimental design details, following the layout of of *inputs_example.py* file. Ligand concentration array (*Lstated* section) can be constructed using *calculate_L_stated_array.py* script.

Sections of `inputs.py`

- *xml_file_path* : relative path to xml plate reader output files.
- *file_set* : option to group multiple experimental sets with a dictionary key.
- *ligand_order*: List of ligand names per each experiment set (one protein, one buffer row). If *skip* is specified as ligand name in this list, *quickmodel* analysis will skip the analysis of that experiment.
- *section* : Data section label of Tecan Infinite M1000 Pro plate reader as specified in its method.
- *wavelength* : Emission wavelength picked for analysis (nm).
- *Lstated* : Experimental value of ligand concentration (M), in NumPy array form. It can be constructed using *calculate_L_stated_array.py* script.
- *L_error* : Estimated % error in stated ligand concentrations.
- *Pstated* : Experimental value of protein concentration (M).
- *P_error* : Estimated % error in stated protein concentration.
- *assay_volume* : Volume of each assay sample (L).
- *well_area* : Area of microtiter plate well (cm^2)

calculate_L_stated_array.py

Generates Numpy array of stated ligand concentration (Lstated) for logarithmic or linear dilution along a row. This numpy array is necessary to construct *inputs.py* file for *quickmodel.py* analysis. Provide information on how ligand titration is constructed: number of wells in each titration (–*n_wells*), highest and lowest ligand concentrations in molar units (–h_conc and –l_conc), and serial dilution mode (–*dilution*, linear or logarithmic) as inputs.

```
$ python calculate_Lstated_array.py --n_wells 12 --h_conc 8e-06 --l_conc 2.53e-09 --
dilution logarithmic
```

The numpy array this script prints out must be directly copied to *Lstated* section of *inputs.py*.