PyADAPT Tutorial

ADAPT was originally writen in MATLAB and now we at CBio group would like to write it in python a.k.a, pyADAPT.

There are mainly three classes in the python package, `DataSet`, `Model`, and `ADAPT`. You can find their equivalences in the MATLAB project.

DataSet

DataSet class contains the experimental data we obtained from different sources. It is organized into a list of all the states and fluxes in the dataset. And It is able to produce a spline interpolation evaluated at different time intervals.

To instantiate DataSet, a raw data and a corresponding data description file should be provided.

```
data = DataSet(raw_data_path='data/toyModel/toyData.mat',
               data_specs_path='data/toyModel/toyData.yaml')
```

Since there's no way to standardize the way raw data is stored and defined, it is also possible for users to first read the raw data and the data specs as python dictionaries and the use them to instantiate the dataset.

Currently, this DataSet class meets all the needs of the toy model. For more complicated models which contains more than one measurement, such as the clamp model with four measurments, class `DataSets` will be a future solution.

Model

`Model` is a abstract class that provide a convenient interface to define an ADAPT style model. The goal is to keep the definition of models in both ADAPT and pythonic way. In order to define a new model `NewModel`, one should first inherit it from `NewModel(Model)`, add parameters, constants, states in `__init__` method and extend three important methods: `odefunc`, `reactions` and `inputs`. Check `pyADAPT.models.toy_model.ToyModel` for example.

odefunc

`odefunc` will be passed as an argument of `scipy.integrate.solve_ivp`. There's not much restriction on how `odefunc` should be implemented as long as the it takes time, initial values and paramters as the arguments. The `Model` class works by remembering the order of the states and parameters. Thus, `odefunc` should always return the derivatives in the same order as the initial values (`x0`).

reactions

Reactions are actually any arbitrary expression that uses the components of the model to obtain a value. In the MATLAB implementation, it can be either real reaction fluxes or a interesting intermediate value in the simulation that we would like to keep track of. In some models, the model outcome is fitted to the experimentally measured fluxes. I will implement this function when dealing with clamp model.

inputs

Inputs is not useful in the toy model but is important in the clamp model. It's function is overlapping with the constants in the model, so further inspectation is needed. I will work on it after I learn more about the clamp model.

ADAPT

ADAPT class contains all the procedure to perform an ADAPT simulation. It takes a model instance and a dataset instance as the argumnets. And it will calculate the parameter trajectory. It supports parallel computing with python multiprocessing module. In an ADAPT simulation, there are usually many iterations, class `ADAPT` is able to run multiple iterations at a time by assigning them to different proceses.