10 – Simple data manipulation in BMTK

Instructions

This document assumes you have completed the necessary steps in 02-Single_Cell_Hoc_BMTK

In a previous guide you developed a single cell network using a single half-center oscillator cell. In this guide we'll extract some of the resulting data.

1. In the previous examples you plotted data using a single (`plot_report`) method call, but what if you want voltage or calcium traces for manipulation? We'll create a small library function you can use in any project to extract that data.

2. Copy the create two new files in your project directory (`data.py`) and (`__init__.py`)

3. Leave `__init__.py` empty but paste the following code into `data.py`:

data.py

```
1
2  from bmtk.utils.cell_vars import CellVarsFile
3  from bmtk.analyzer.cell_vars import _get_cell_report
4
5  def get_variable_report(config_file=None, report_file=None,
6  report_name=None, variable=None, gid=None):
7      """Returns variable report for specified gid
8      Function will return the report for a specific cell's variable.
9      """
10     if report_file is None:
11         report_name, report_file = _get_cell_report(config_file,
12 report_name)
13
14     var_report = CellVarsFile(report_file)
15     time_steps = var_report.time_trace
16
17     return var_report.data(gid=gid, var_name=variable), time_steps
18
```

4. Things to note about this file:
    a. It's heavily based on `plot_report` in https://github.com/AllenInstitute/bmtk/blob/develop/bmtk/analyzer/cell_vars.py
    b. It can be further modified to return multiple variables or multiple cells etc. if you want

5. To use your new function, you'll need to import it and supply the appropriate parameters. The following is an example usage script (located in the root of your project directory):

test_analysis_script.py

```
1
2  import matplotlib.pyplot as plt
3  from data import get_variable_report
4
5
6  var, time_steps =
7  get_variable_report(config_file='simulation_config.json', gid=0,
8  variable="v")
9
10 plt.figure()
11 plt.plot(time_steps,var)
12 plt.show()
13
```

6. Things to note:
    a. This will plot a cell "0" voltage graph
    b. Data is directly accessible from the `var` variable
    c. The `time_steps` variable length is determined by dt (resolution) and tstop (length) in your `simulation_config.json` file
    d. Any variable can be supplied to the `get_variable_report` method as long as it has been declared in your `simulation_config.json` as an output. Ex:

simulation_config.json

```
…
42   "reports": {
43       "membrane_report": {
44       "module": "membrane_report",
45       "cells": "all",
46       "variable_name": [
47         "cai",
48         "v"
49       ],
50       "file_name": "cell_vars.h5",
51       "sections": "soma"
52     }
53
```