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  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18 | from bmtk.utils.cell\_vars import CellVarsFile  from bmtk.analyzer.cell\_vars import \_get\_cell\_report  def get\_variable\_report(config\_file=None, report\_file=None, report\_name=None, variable=None, gid=None):  """Returns variable report for specified gid  Function will return the report for a specific cell's variable.  """  if report\_file is None:  report\_name, report\_file = \_get\_cell\_report(config\_file, report\_name)  var\_report = CellVarsFile(report\_file)  time\_steps = var\_report.time\_trace  return var\_report.data(gid=gid, var\_name=variable), time\_steps |

1. Things to note about this file:
   1. It’s heavily based on plot\_report in <https://github.com/AllenInstitute/bmtk/blob/develop/bmtk/analyzer/cell_vars.py>
   2. It can be further modified to return multiple variables or multiple cells etc. if you want
2. 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  3  4  5  6  7  8  9  10  11  12  13 | import matplotlib.pyplot as plt  from data import get\_variable\_report  var, time\_steps = get\_variable\_report(config\_file='simulation\_config.json', gid=0, variable="v")  plt.figure()  plt.plot(time\_steps,var)  plt.show() |

1. Things to note:
   1. This will plot a cell “0” voltage graph
   2. Data is directly accessible from the var variable
   3. The time\_steps variable length is determined by dt (resolution) and tstop (length) in your simulation\_config.json file
   4. 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  43  44  45  46  47  48  49  50  51  52  53 | "reports": {  "membrane\_report": {  "module": "membrane\_report",  "cells": "all",  "variable\_name": [  "cai",  "v"  ],  "file\_name": "cell\_vars.h5",  "sections": "soma"  } |