# Experiment 1:

* Network is built with the connections defined in the diagram below. The number of cells per region in the model are as follows: EC – 30 (of which 20 are selected for any particular run); CA3 – Pyr 63, BC 8, OLM 8; DG – GC 384, DGb 32, DGh 32. So, there are a total of 7 cells types among all the regions.
* Inputs are presented to EC, with input synapses designed so that one spike input elicits one spike in EC cell.
  + For 5 iterations 10 cells are presented an input pattern. During this time Ach is at a high state in the synapses and learning is on. Weights of the synapse change with successive inputs, biologically this is equivalent to the introduction of a novel stimulus.
  + One input pattern consists of pulses at short 80ms intervals for 625ms – or 12.5 Hz. That is, a spike train at the upper range of theta. This results in 7 spikes per pattern input. So, this is the ‘encoding’ phase where the input pattern is ‘learned’ by adjusting the synapses using the standard learning rule for five iterations.
  + Starting iteration #6, ACH is turned down (1) and learning is turned off (0) in the synapses. The pattern remains the same for the first iteration to demonstrate response outside of the ‘encoding’ phase. This is the beginning of the ‘retrieval’ phase, i.e., checking what pattern emerges in CA3 (is ‘retrieved’) with a specific input pattern from EC. Specifically, this set of runs characterizes how much of the learned pattern #1 is retrieved by gradually dissimilar input patterns presented subsequently
  + Each of the following iterations (up to 15) will replace an additional cell until none of the original cells presented with the original input pattern are firing. The input pattern for each trial is again 7 EC cell spikes, but from different EC cells as cited.
  + Of the total 30 EC cells only 20 are used during any given simulation. By adjusting the initial seed we choose different sets of 20 each time, producing slightly different results to be averaged over several simulation runs.
* The time at which each CA3, DG, and EC cell spikes and its GID are recorded, threshold for counting the voltage increase as a spike being -15mV.
* Initially, all spike times with the associated cell GID are saved in a single spikes.csv file (each row being: time, GID) where 0 is the first GID and 556 is the last (556=total number of cells). By running the script generate\_legacy\_output.py we can separate the cell types out into multiple files for each cell type. (0-6 – a mapping created by Ali and used in his MATLAB scripts) for each cell recorded in all the regions – EC, CA3 and DG. This produces 7 files in the legacy folder, (each row being GID,time) where the GID starts from 0 for each cell type file. If the cell does not spike, it will not appear in this file, however every time it spikes an entry is made with its GID and time of spike.
* Correlation.m is run from Ali’s directory and the pattern separation / completion comparison graphic is displayed
* Correlation sets iteration 6 as the ground truth to be compared with the following iterations. For each of the trials from 7 to 16 (note: 7 EC spikes per trial):
  + Count the number of spikes per cell per type for EC, CA3e, and DG. If the cell does not spike 0 will be recorded. It is possible a receiving cell spikes more than the number of times it receives input.
  + Normalize this number between 0 and 1, using the maximum number in the file. So for each trial, there will be a vector each for EC, CA3e and DG with entries containing a raw number of times each cell spiked during that trial period of 625ms. A dot product is then performed with this vector (either EC, CA3e or DG) and the corresponding ground truth vector (iteration 6, as stated above).
  + Compute the dot product of the total number of spikes per region matrices to calculate the correlation between iterations, resulting in a single number per iteration for each cell type.
* We expect CA3 to remain highly correlated (completion) with the original input and DG to be less correlated (separation)
* MATLAB code steps for clarity (done for EC CA3e and DGg to produce correlation plot)
  + Open the directory containing spike information files
  + For each file containing information about the desired cell type (multiple for multiple seed runs)
    - Import the data
    - Parse the trial for each file
      * Given StimCount (16), number of cells in the region, end time, when each cell spies, reference number (6th trial)
      * For each trial compute the number of times each cell spikes
      * Normalize
        + For each trial
        + num spikes = num spikes/norm(num spikes)
      * Return a vector of trials compared to reference trial
        + Dot product between reference and each trial
    - Append correlation found to a master correlation (cor) variable per cell type
  + Find the mean of all of the correlations
  + Find the standard deviations of all of the correlations
  + Plot along a bounded line

# Running the Code:

1. **Generate the synapse JSON files** (./biophys\_components/synaptic\_models/) from Ali’s synapse property matrix (HummosEtAl2014/Inputs/Synapses.txt) (to be executed the first time only)

python generate\_synapses.py

1. **Generate the input spike times** (./input/exp0\_input\_new.csv)

python generate\_input.py

1. **Build the network** (output dir: ./network/ unless otherwise specified)

python bulid\_network.py

1. **Run the network** (output dir: ./output/ unless otherwise specified)

python bulid\_network.py

1. **Analyze the results** (requires bmtools [https://github.com/tjbanks/bmtools])

python –m bmtools.plot raster

1. **Correlation** (run in MATLAB)

Correlation('bmtk\_','../HummosBanks-bmtk/legacy/')

# Connection Diagram

