# Measuring self-consistency with simulations

Paper: J.C. Jackson, A.D. Redish (2003) “Detecting dynamical changes within a simulated neural ensemble using a measure of representational quality” *Network: Computation in Neural Systems*, 14:629-645.

Originally from: A.D. Redish (1999) Beyond the Cognitive Map: From Place Cells to Episodic Memory, MIT Press. *Appendix A*.

# Notes

The simulations are in the ANN directory.

These simulations are of a standard “ring attractor” network. A lot is hacked together and hardcoded, but it will serve our purposes for today. (For example, the code is written with 75 units. Changing that would be a pain.) The simulations were originally written in 1997 as a part of my PhD Thesis and included in the 1999 book as *Appendix A*.

The attr function runs the original code. You can try attr(1); attr(2); attr(3); attr(4); attr(6); attr(7); attr(8); There is no figure 5. I have cleaned it up trivially and included it as ANN. The ANN function takes an input of the structure   
{[i0, t0, s0], [i1, t1, s1],…}. You can include as many as you want in the list. If called without arguments, it runs a random input (equivalent to attr(1)). The ANN function builds an input structure of 100 times steps of a random input (same random input for those 100 timesteps) and then 200 timesteps of zeros and then adds in the inputs you specify in the function inputs. It will put a small exponential centered around each input (e.g. i0), and add that in for 100 timesteps starting at time t0, with strength s0.

If you call it without any outputs, it will not return any, but if you return outputs, it will output the activities of the units that you can use to display or calculate with.

# Task 1: decoding

## Goals

* Understand the properties of the ring attractor network
* The length of the vector mean is the decoded direction. Why?
* The length of the vector mean is the self-consistency of the network. Why?

## Steps

1. Run attr(1) through attr(8).
2. Test how the bump shifts as you increase the distance from the center of a weak correction input.
3. Test how the bump shifts as you increase the strength of a distal correction input.
4. Test how the final bump location changes as two comparisons approach each other.