**Notes from 1 March 2016**

* 3 layers for a CA system: CA1 layer, CA2 layer, Interaction layer
* XOR complexity can be suppressed by ANDs
* Most simple neural networks are XOR gates
  + XOR is impossible to compute
  + Multiple layers, can be Turing complete and universal
* How to compute XOR?
  + Decreasing and increasing the number of outputs, by looking at the logic table
* Compare Boolean behavior to Kolmogorov complexity
* Gliders have highest probability to be Turing complete
  + How to test for this?
  + Turing complete (probably) if it can compute XOR and NAND (it will eventually reach any state given a large enough tape)
* What % of each trajectory repeats itself?
  + Static and dynamic metrics
* Game of life: Correlation between static and dynamic metrics
  + Add K as a feature in the vector
  + Can get interesting CA in a few steps of selection through a GA

Idea: Adapt Discovery

Understand XOR gates and how they interplay with other gates (min. Boolean expressions)

* Start with a random interaction rule, and select on it
  + CA1 with random rule and behavior metrics,
  + CA2 with random rule and behavior metrics,
  + Interaction rule with behavior metrics
  + Put these through a GA to get ones that are Turing complete
* 2 fitness parameters selecting simultaneously
  + How long are cycles?
  + Select on the longer cycles! Select those that have the longest cycles for the CA

To start: Use the app to calculate attractor sizes (cycle lengths)