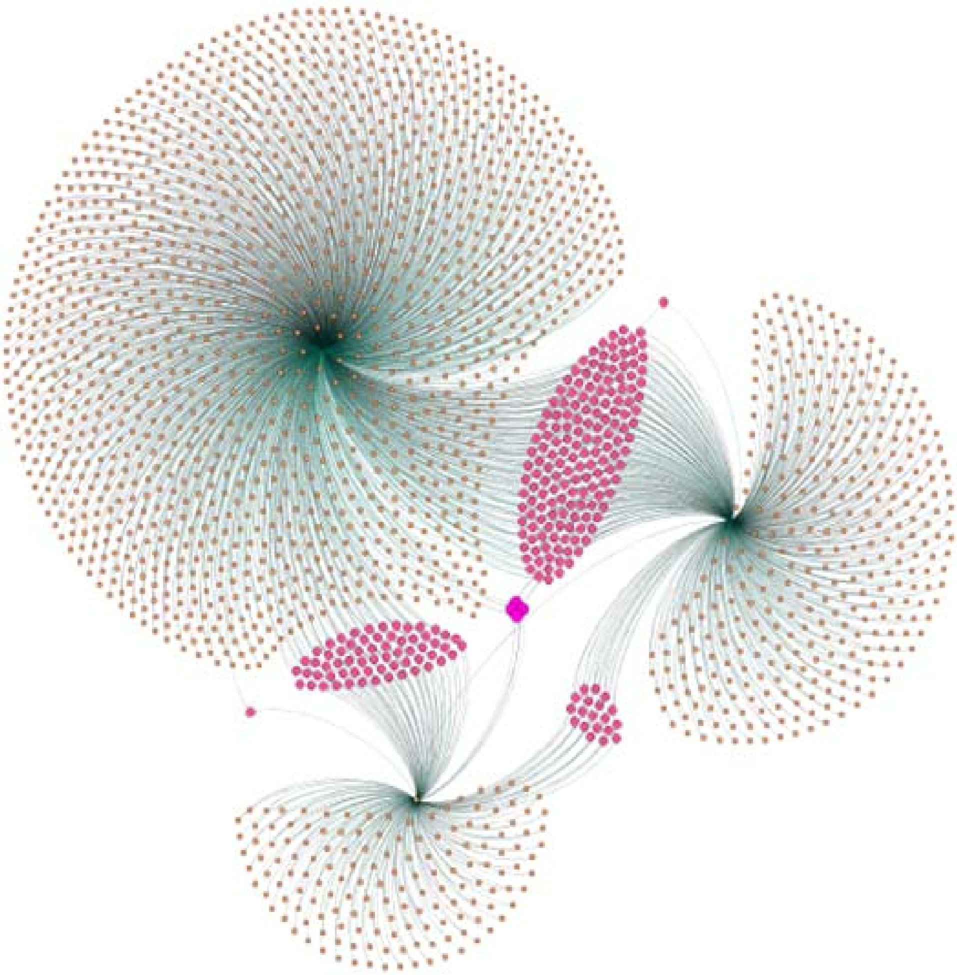
Graph layout

We experimented with existing graph layout software – Gephi. (http://gephi.github.io/) – these experiments are given first, with alternative views generated by our software afterwards.

Its use gave us some good ideas for layout for our problem which transferred to the final system. We ultimately built graph layout into our application to gain linked views between our existing visualizations and the graph layout. Gephi allowed edge weights that would influence layout, whereas in our implementation we can also fix edge lengths to computed distance in phase space.

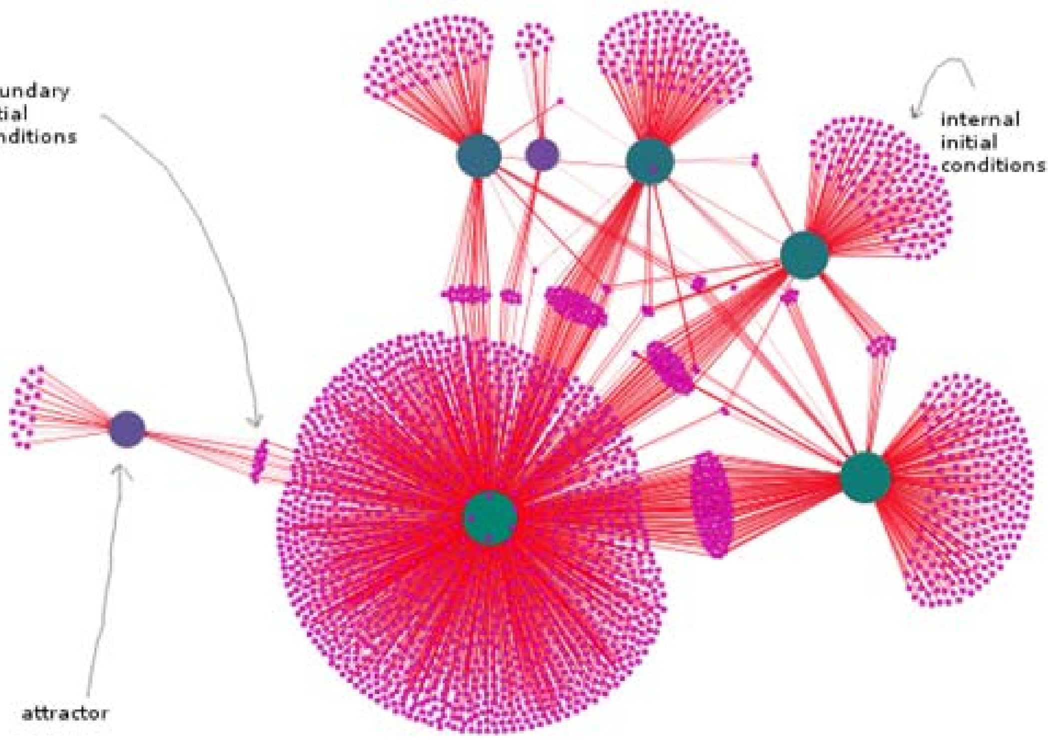
But here are the designs we used for inspiration along the way.

Experiment 1. Artificial data. Three attractors. Each state goes to one of the attractors in deterministic runs, but in the noisy runs it may go to one of the other attractors. The graph layout bunches states around their attractors, but bifurcating states are naturally pulled out between the shared attractors creating a visually obvious cluster. From this one visualisation we have the benefit of being able to see the relative size of basins, and the number of bifurcating points. We also see which attractor can "capture" states from another attractor. e.g. if we think a system is in the noisy area we must be aware that intervention (or lack of) could take us into either state. We could hover or click though on states to see the conditions. We could select many states, and see aggregate views (either through our multiple vector plots, or our spline based PCPs). This approach seems aesthetically pleasing due to the relaxed spacing.



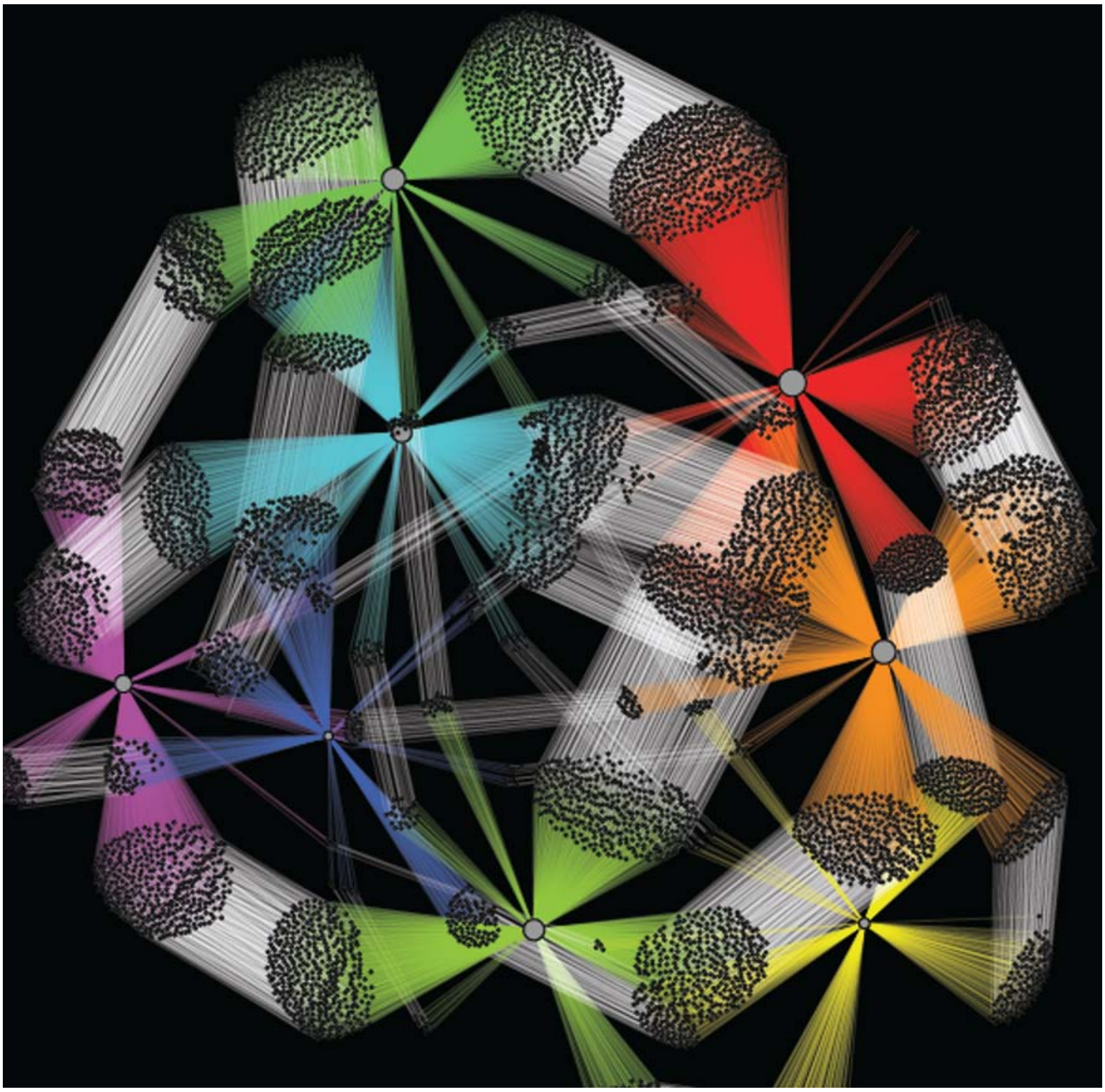
Experiment 2.

This time a retail model is applied to a hypothetical three retail center system (Figure 1 in the paper). In this case, with noisy runs from regular sampling over the hyperplane, the system results in 7 attractors. The situation and description is similar to the previous experiment.



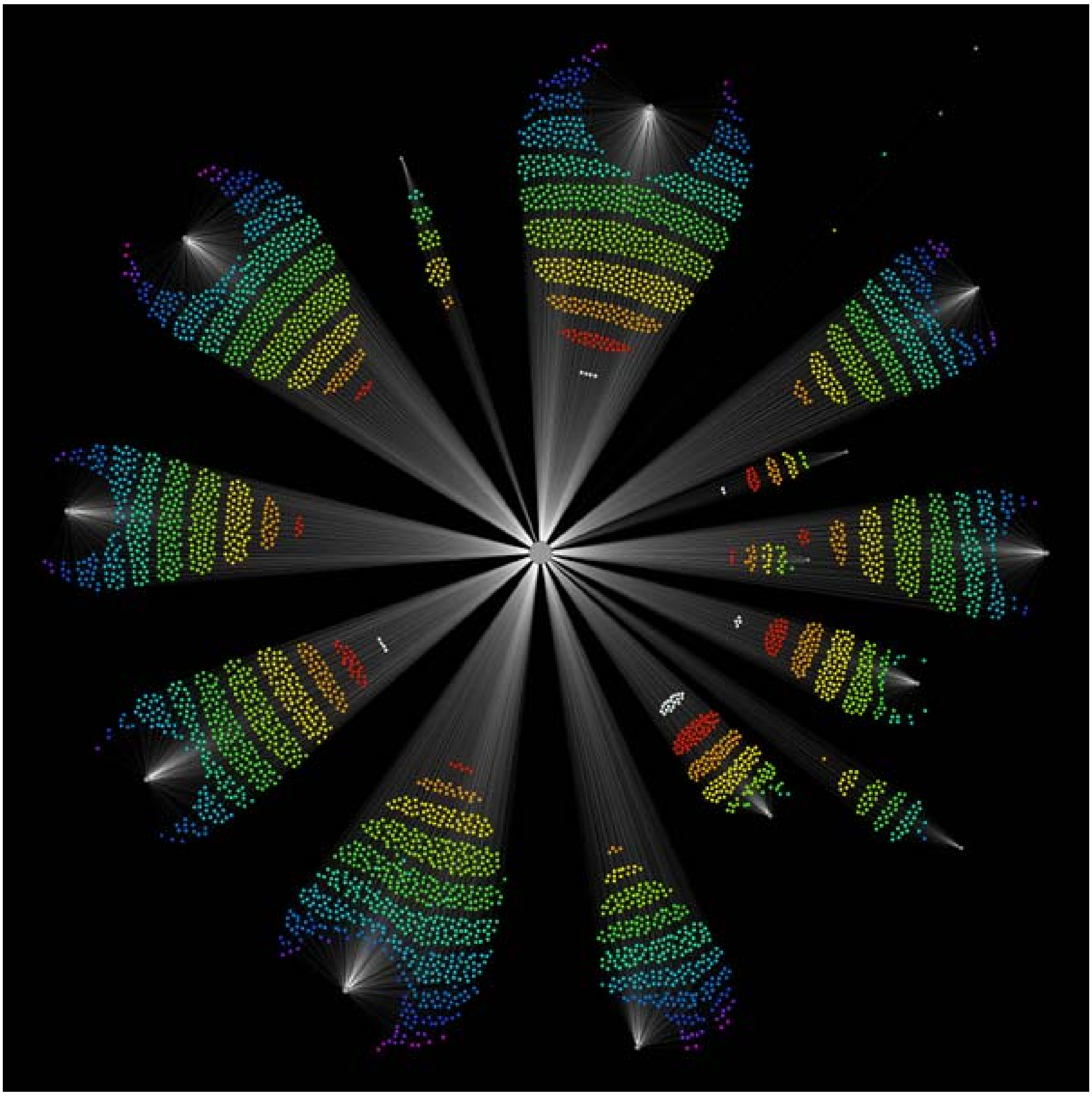
Experiment 3.

This time an 8D retail center model for South Yorkshire is created (as in the paper). At this stage we are having problems and layout issues with Gephi, so we only used 10% of the data. Each edge indicates a connection between a state and its attractor. Edges are colored categorically according to attractor. Again, the force directed layout shows this method is suitable for the reasons outlined above: Attractors popout, their size can be compared, bifurcating points popout (and their relative sizes) and with suitable selection and querying tools, the user has a powerful visualization tool for understanding their model.



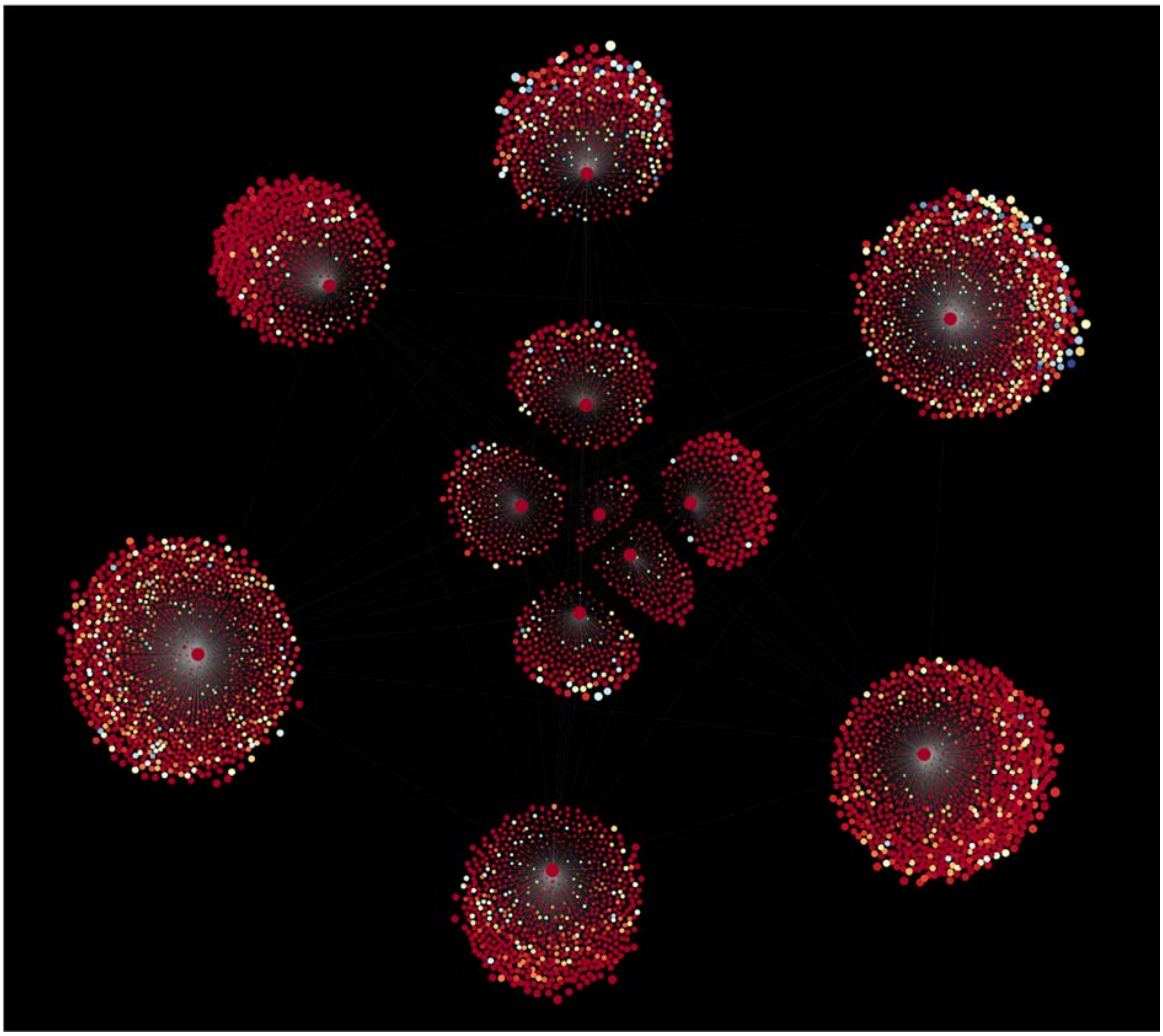
Experiment 4

This iteration has features we retained for the final approach taken in the paper. Color represents distance in phase space, but Gephi does not allow edge length to represent distance.

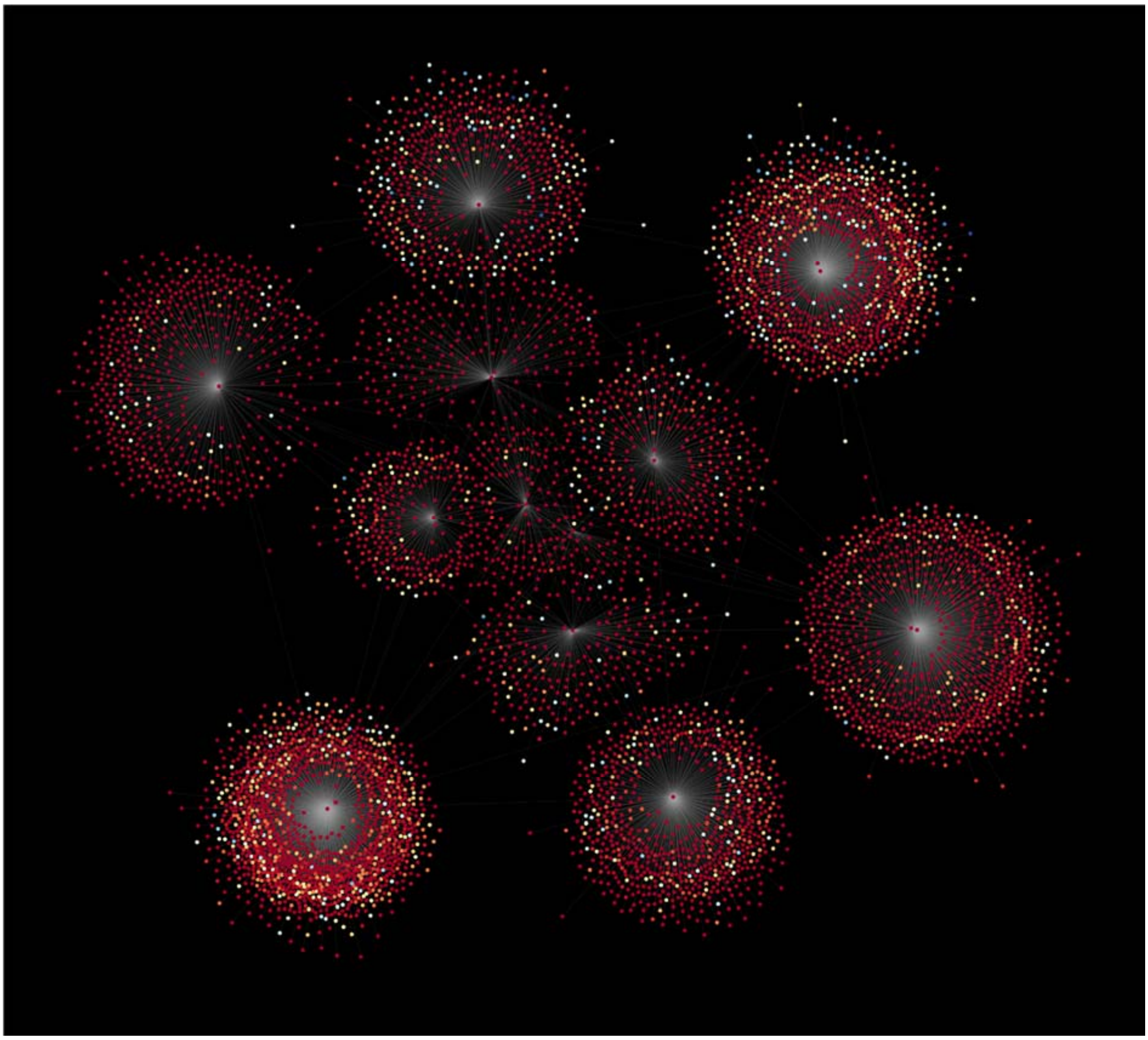


In our software, DynaMoVis we have options to alter the graph layout. We had to pick one layout for the paper, and we anticipate that reviewers may prefer an alternative.

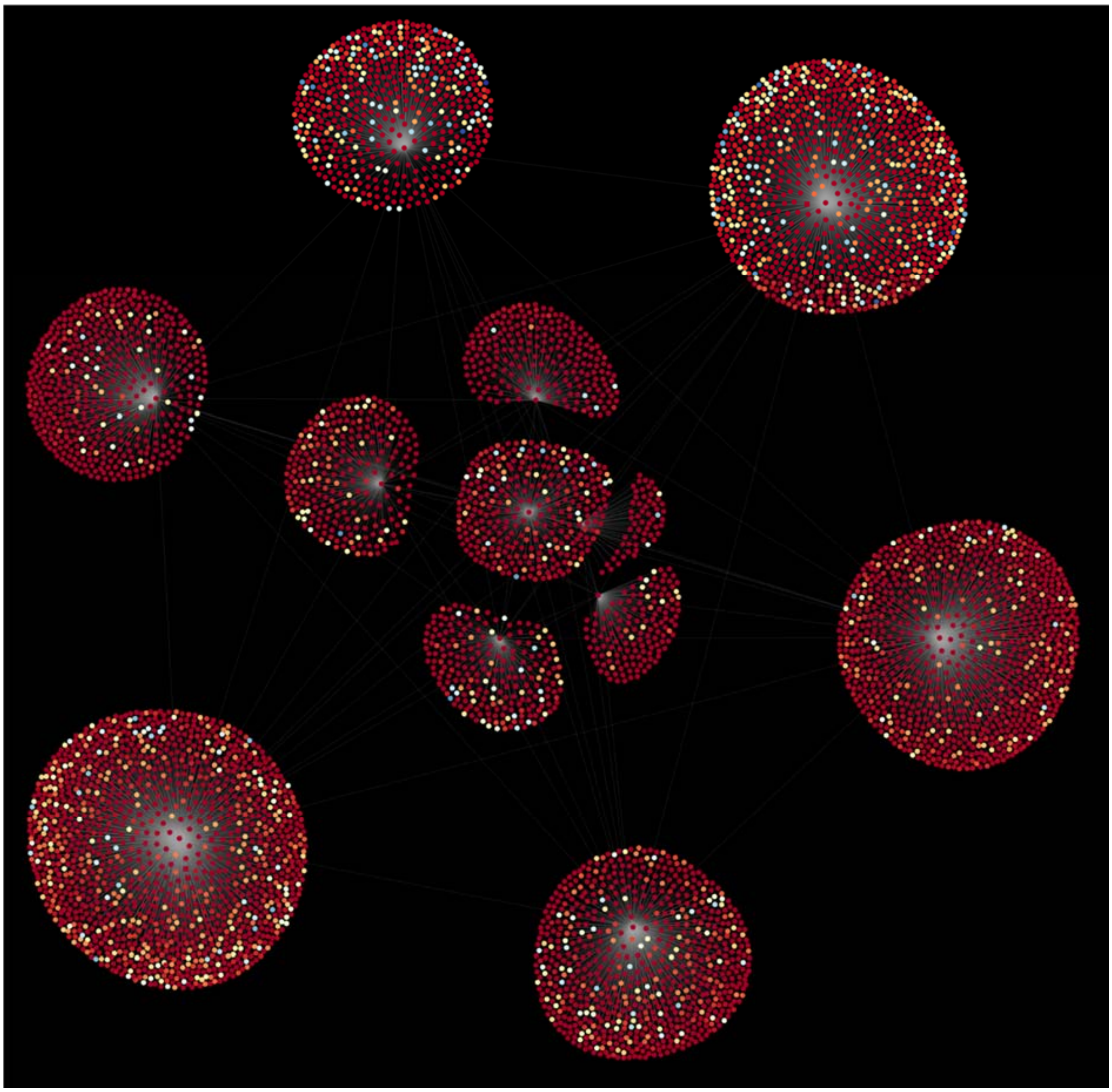
Colour, red to blue divergent Color Brewer scale, signifies the number of noise runs that converge to an alternative attractor compared to the deterministic run (blue=a lot of the variance runs change to a different attractor). The force directed layout clearly clusters basins of attraction. Edges are simply connections between the deterministic run and its attractor. We see that the attractor at the 2 o clock position is particularly sensitive to noise. Node size represents distance from attractor in nD space (attractors are all fixed node size unrelated to any variable). We therefore see that for that attractor, even nodes close to the attractor center potentially converge to a different attractor. Therefore this is not a stable basin.



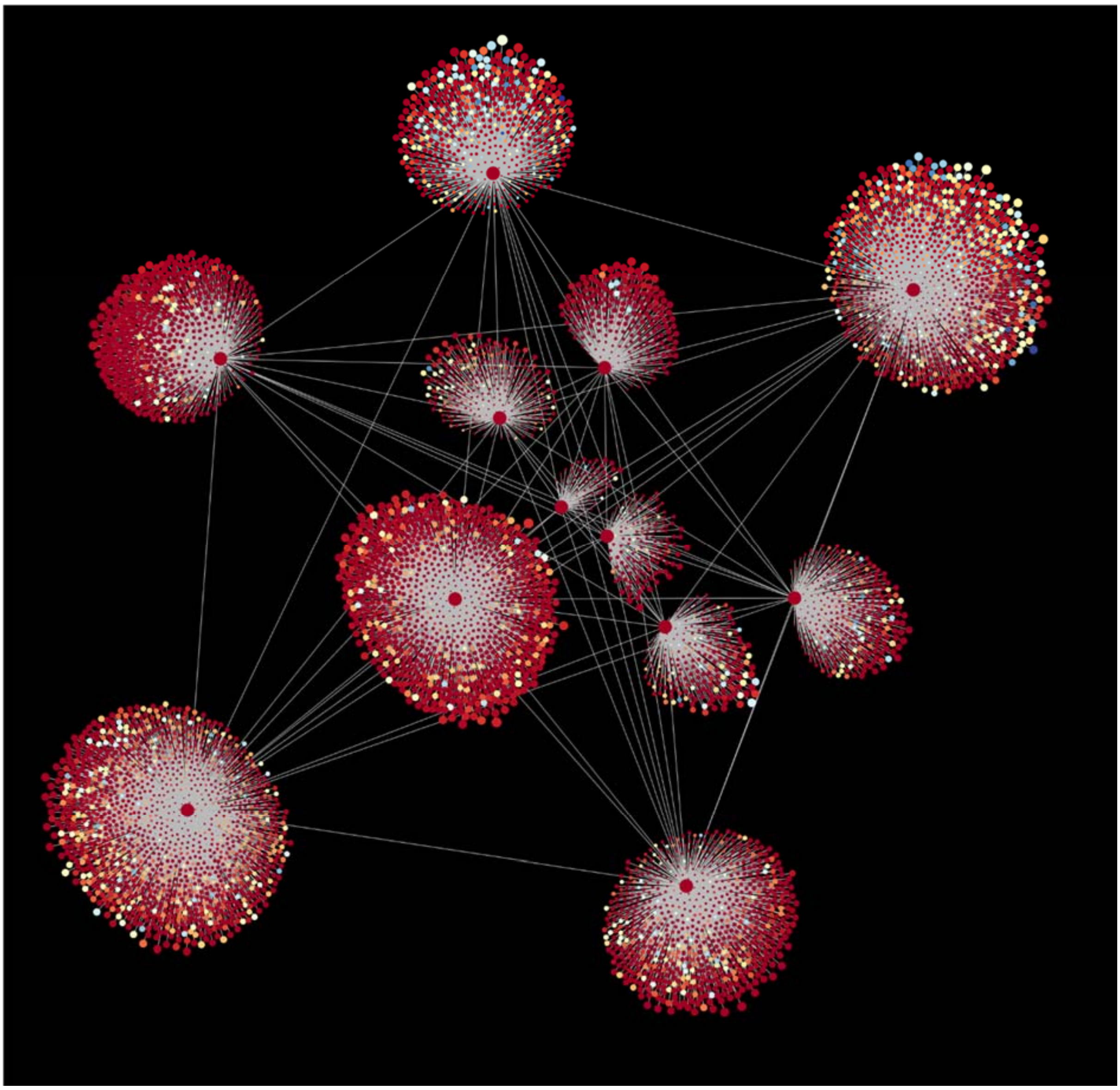
The next layout differs in that all nodes are the same size, but edge length is now distance from attractor in nD space. This perhaps gives a better impression of neighbouring basins. Encoding distance from attractor as node size as above may be difficult to cognitively integrate, but we decided to use the above image in the paper as it makes it obvious that interactive selections could be easier with distinct clusters.



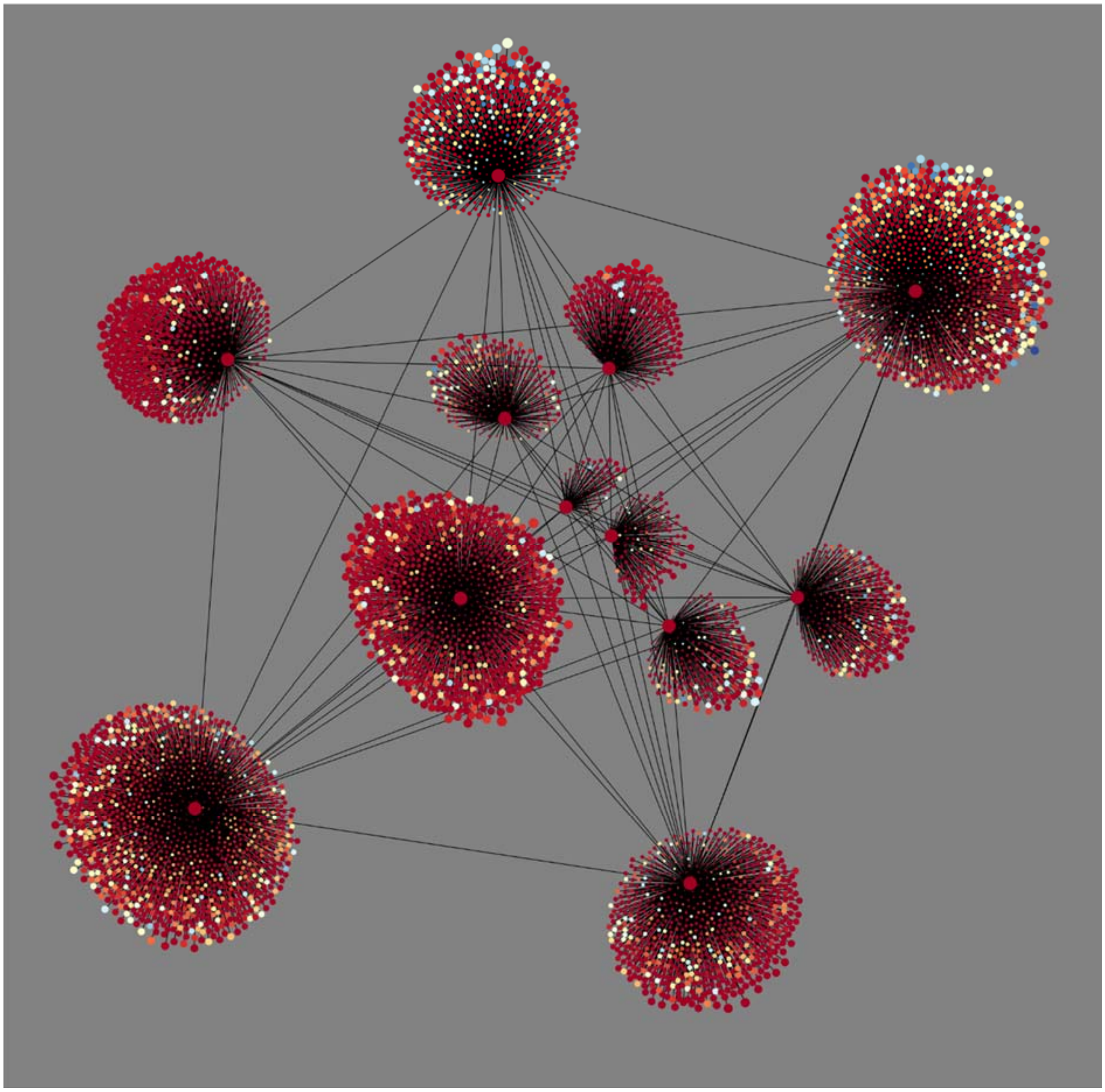
In this alternative distance from attractor is removed from any visual channel. Additionally nodes are relaxed leading to a more aesthetically pleasing approach like the Gephi examples we liked. Essentially, we can encode any of the data, such as distance from attractor, percentage of noisy runs converging to an alternative attractor, connections between an initial state and its attractor(s), to any of the visual channels – color, edge length, node size, and run force-directed layout (and with or without relaxation).



Edge line density can be turned up and down (but we preferred the low density, even though it omitted the attractor edges).



Background, foreground colours can be changed



Alongside the graph view we can bring up a view of the adjacency matrix. Clicking on a square in the adjacency matrix selects those two attractors and the shared nodes between them – i.e. nodes that bifurcate between those two attractors.