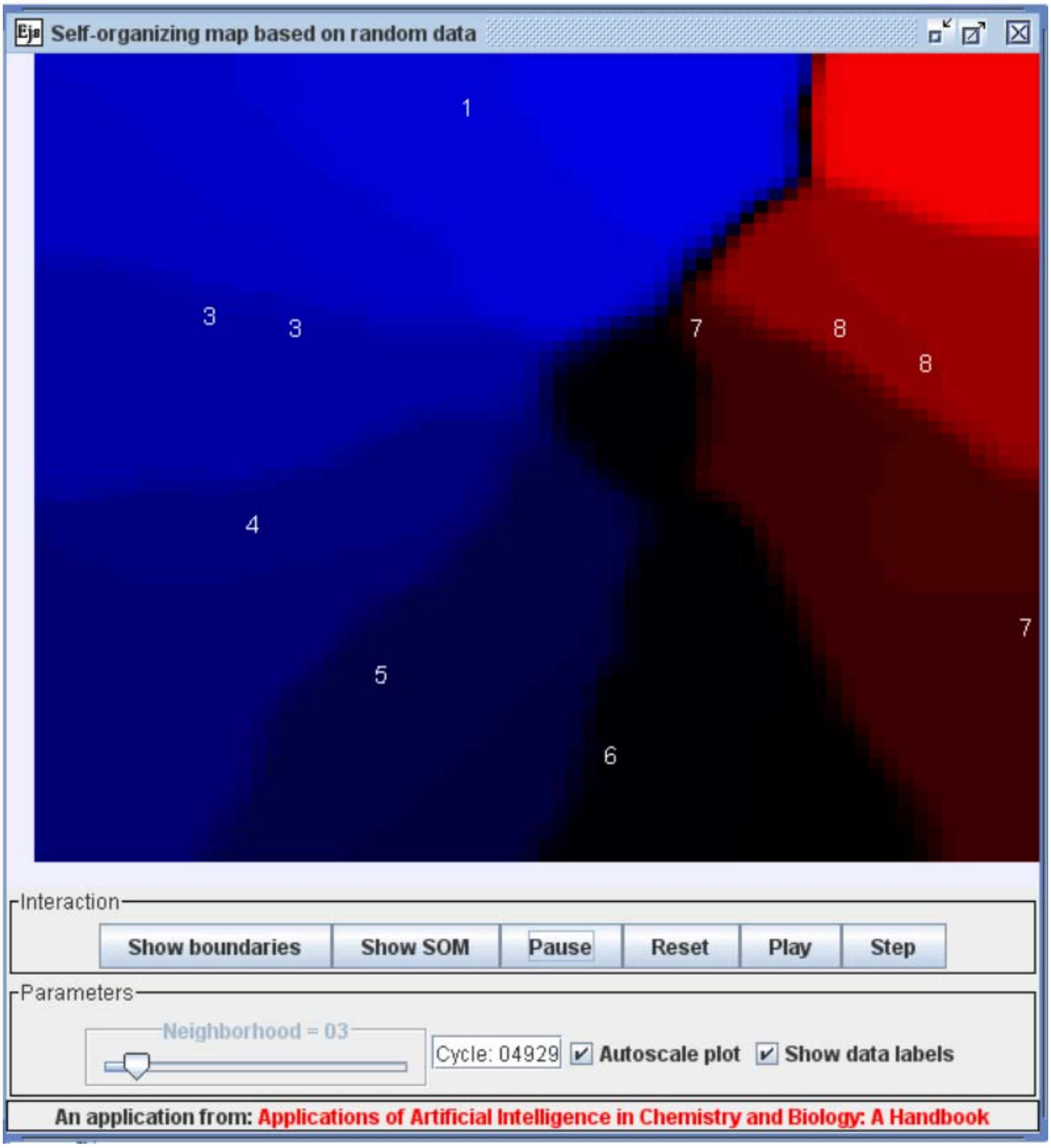
Simple SOM

This SOM handles two-dimensional data, but the data are a little more complicated that that used to create the untangling SOM.



In this instance, the data points fall in one of ten classes, created using the instructions:

```
for (i=0; i<10; i++)
{
    xpoint[i]=i*i;
    ypoint[i]=5.0+i;
}
```

When a datapoint is picked to be fed into the map, some random noise is added:

```
pointchoice=(int)(10.0*Math.random());
xp=xpoint[pointchoice]+Math.random();
yp=ypoint[pointchoice]+Math.random();
```

The SOM sorts out the data into classes, as shown above. The numbers show the class to which each data point belongs – the last ten data points that were fed into the map are identified. The SOM, after a period of training, effectively sorts the data without, of course, having been told the identity of the points it is reading.

Two types of display are available. **Show SOM** produces the type of display shown above, in which each node on the map is colored according to its weight. **Show boundaries** generates a display in which the color of a node is dependent upon how different the nodes of weight are in the immediate region of the node.

BUTTONS

Button	Function
Show boundaries	If this button is pressed, the display shows how different the weights are in the nodes immediately around each target node. This is done by finding the Euclidean distance of each node weight from the corresponding weight at the node. The mode of display emphasizes the regions on the map where the weights are changing most rapidly from one node to the next.
Show SOM	If this button is pressed, the display colors the different regions of the map according to the weights at each node (this is the display mode used for the map shown above).
Pause	Temporarily halt execution.

Reset	Restart the calculation from scratch; the neighborhood size is reset to the default value.
Play	Restart the program if it has previously been paused.
Step	Execute a single cycle of the program.

SLIDER

Slider	Default	Comment
Neighborhood	15	The neighborhood defines the size of the region around the BMU that is updated each cycle.

TICK BOXES

Box	Comment
Autoscale plot	If this is ticked (the default) the colors in the plot are automatically adjusted so that they cover the full range of values to be displayed. Thus, if the values to be plotted run from 15 to 83, the colors will be used so that one end of the range corresponds to the value 15, while the other corresponds to 83. If this box is unticked, the mapping of colors to values does not change as the range of values changes. This can sometimes give a clearer definition of boundaries in the boundaries plot, though at the expense of a possible loss of detail.

Show data labels	Each time a data point is fed into the map, a type number is associated with it. The points are created in initialization with the code:
	
	`for (i=0; i<10; i++)`
	`{`
	`    xpoint[i]=i*i;`
	`    ypoint[i]=5.0+i;`
	`}`
	
	And then picked in evolution with the code
	
	`pointchoice=(int)(10.0*Math.random());`
	`xp=xpoint[pointchoice]+Math.random();`
	`yp=ypoint[pointchoice]+Math.random();`

	Although `pointchoice`, the identity of the type of point, is not know to the algorithm, it can be shown on the display so that one can judge how well, or badly, the WSOM is learning to sort out the data.

Investigations

1. Default parameters

Run the simulation using the default parameters. What is the origin of the big splodges of color that appear rapidly as soon as the simulation starts?

2. Neighborhood size

The size of the neighborhood diminishes slowly as the simulation runs, but it can also be adjusted by hand using the slider. Restart the simulation and investigate how long it takes before meaningful separation into different regions for the different classes starts to appear using a small, medium or large neighborhood. Why is it likely to be advantageous to start the calculation with a large neighborhood instead of one of just two or three nodes? (Hint: consider the effect both of speed of calculation and also the quality of the finished map in terms of the number of different regions is it divided.)

3 Form of map

Restart the simulation and click on the Show boundaries button. Compare this form of display with the “normal” display. Are there advantages to using one rather than the other once the SOM has converged? Try adjusting the Neighbors for boundary plot variable; this is the size of the region around each target node that is taken into consideration when preparing the plot. Is it obvious what the effect of this parameter is on the plot qualitatively? What are the advantages or disadvantages of using a large or a small neighborhood for this type of plot?

4. Adjusting the display

You can change the appearance of the display by going to the **View** page of Ejs, left-clicking on the **Gridplot** in the left hand panel, selecting **Properties** and selecting a new colourmode.