

# Scientific Programming Assignment 3

MPhil in Computational Biology

December 17, 2010

If there are errors found, I will update the assignment on the web at

<http://www.damtp.cam.ac.uk/user/sje30/teaching/r>

**Due date: 2011-01-19 23:45**

Please submit your report to the CamTools website as a PDF: NO OTHER FORMATS ARE ACCEPTED. Name your file `spa3_XXX.pdf`, where XXX is your CRSid. (For example, I would save my file as `spa3_sje30.pdf`).

Put a copy of your code (but nothing else) in the appendix of your report. To include R code in latex, see the example code at: <http://www.damtp.cam.ac.uk/user/sje30/teaching/r/rlistings>.

Your report must be a maximum of eight pages, excluding the appendix. This course work will consist of 30% towards your overall mark for this module.

## 1 Simulating spatial point patterns

Figure 1 shows the two-dimensional distribution of a set of points. Each point could correspond to the location of a neuron within a 2d tissue, or e.g. to the position of a tree within a forest. Point-patterns like this can be modelled with a “minimal distance” rule: points are positioned randomly subject to the constraint that no two points come closer than a minimal distance to each other.

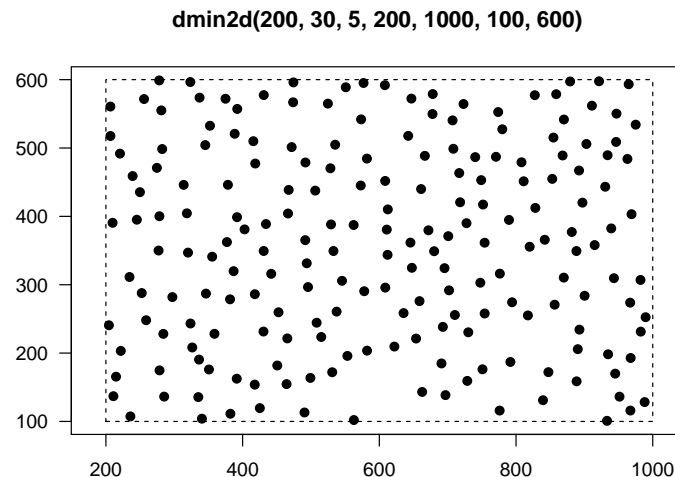


Figure 1: Example 2d spatial point pattern created using the  $d_{\min}$  model. The dotted line indicates the boundary of the sample region.

### 1.1 The $d_{\min}$ model

In the  $d_{\min}$  model each point has a circular exclusion zone surrounding it, which prevents neighbouring points from coming too close to it. To model a spatial pattern, we create a region **A** of the

same size as the real pattern. Initially this region is empty; points are added to **A** using a serial algorithm, positioning points one by one into the array until the number of points in the model region **A** matches the number in the real pattern (or until no more points can be added). To add one point into **A** we follow the following steps:

1. Generate a trial point and exclusion zone  $(x, y, d_{\min})$ . The position of the neuron  $(x, y)$  is determined by uniform sampling of **A**. The effective diameter of the exclusion zone,  $d_{\min}$  is drawn from a Normal distribution with fixed mean and standard deviation ( $\mu = m, \sigma = s$ ). The Normal distribution is truncated at some lower bound  $d_{\text{low}}$ , which here should be zero.
2. Find the distance of the trial point to all other points that have previously been accepted into **A**. The smallest of those distances is labelled  $d$ .
3. If  $d < d_{\min}$ , the trial point is too close to an existing point. The trial point is thus rejected, and you return to step 1 to generate another trial position. Otherwise, if  $d \geq d_{\min}$ , the trial point is accepted.

To model a spatial pattern, the only parameters required by this model are the mean and standard deviation of the exclusion zone.

## Part I: Build the $d_{\min}$ model [10 marks]

Implement the  $d_{\min}$  algorithm into a R function of the following form:

```
dmin2d <- function(n, m, s, xlo, xhi, ylo, yhi) {
  ## N: number of points to simulate
  ## m: mean of Normal distribution
  ## s: s.d. of Normal distribution
  ## xlo, xhi: possible range of X values.
  ## ylo, yhi: possible range of Y values.
}
```

Plot the output from your model when called with:

```
res <- dmin2d(200, 30, 5, 200, 1000, 100, 900)
```

Your  $d_{\min}$  algorithm should be stochastic: given the same parameter values, you should get different patterns returned.

## Part II: Evaluate regularity index [10 marks]

To quantitatively evaluate the regularity of a spatial pattern, we can calculate various measures, such as the regularity index (RI). For each point, measure the distance to its nearest-neighbour; the RI is then the mean of those distances divided by the s.d. of those distances.

1. Write a function that estimates the RI for a spatial point pattern.
2. The theoretical expectation for the RI for a set of points randomly positioned with no minimal distance constraint is around 1.91. Generate 1000 replicate simulations with the following parameters:

```
dmin2d(n=200, m=0, s=0, xlo=0, xhi=1000, ylo=0, yhi=1000)
```

Measure the RI of each pattern and report the 50th largest value. What is the utility of such a measure? How do your results vary as you vary the number of points ( $n$ ) in a pattern, or the geometry of the sample area (i.e. square regions versus rectangular)?

(Hint: You may need to write your `dmin2d` function so that when the “ $m$ ” argument is zero, the minimal distance constraint is ignored.)

### Part III: Fit the model to some data [10 marks]

Data file `spa3_real.dat` on the course web page contains the x,y locations of 238 points sampled in a region of size  $400 \times 400 \mu\text{m}^2$  (units are in  $\mu\text{m}$ ). Your task is to see which possible values of  $m, s$  in the  $d_{\min}$  model can generate patterns similar to the real data set.

To compare your model output with the observed pattern, for a given pair of parameters  $m, s$ , you should run your  $d_{\min}$  model 99 times and evaluate the RI of each simulated pattern. If the RI of the real pattern is  $x_1$  and the RI of  $n - 1$  ( $n=100$ ) simulated patterns are  $x_2 \dots x_n$ , then for each pattern  $i$  calculate a  $u_i$  score:

$$u_i = \text{abs} \left( x_i - \frac{1}{n-1} \sum_{j \neq i} x_j \right)$$

What does the u-score measure, and how can you use it to assess the quality of fit?

Present your evidence to show which  $m, s$  values can generate patterns that match the spatial properties of the real patterns. Describe your search procedure for finding suitable parameters.