Running the Gierer 2D simulations

Stephen J Eglen

June 27, 2013

Contents

1 Introduction

The two-dimensional Gierer model was implemented in R by David Nissenbaum, with guidance from David Sterratt and Stephen Eglen. The package is available for installation only as a source package; please contact Stephen if you would like to run it under Windows. The `gierer` package depends on the `sjevor` package, a wrapper around Steve Fortune's Voronoi tesselation code. This package is available from `http://damtp.cam.ac.uk/user/sje30/r` and can be installed within R using:

```
install.packages(c("sjevor"), contriburl="http://damtp.cam.ac.uk/user/eglen/r/")
```

The gierer package is contained within the pipeline, and can be installed by the following shell commands:

```
cd /path/to/koulakov
R CMD INSTALL gierer
```

2 Running within R

After the *gierer* and *sjevor* packages have been installed, you can then run a self-contained example within R:

```
require(gierer)
example("Gierer-package")
```

The file `examples/runme.R` describes how to create a 'Gierer' object and then iterate the model. In particular, to create the simulation, we need two files, containing the locations and gradient information associated with each RGC and SC neuron. These data files (`rgc.file` and `sc.file`) are normally generated by the pipeline (see next section); the R package has example data files (both 500 and 2000 neurons) for testing within R.

As well as these two data files, there are four key parameters in `gierer.f` that can be given to construct a simulation:

Parameter	Default	Meaning
nterm	16	Number of terminals made by each RGC
epsilon	0.005	Growth rate for competitoin term
kappa	0	Decay rate for competition term
p0.fn	2	Choice of gradient function

There are several gradient functions implemented, as documented in the results file.

3 Running within the MATLAB pipeline

The R package integrates into the MATLAB pipeline. Before calling the gierer simulation, MATLAB generates two files `rgc.file` and `sc.file` containing the position and gradient information of each neuron. The location of these files, along with the required location of the output weight file, is written into a parameter file. This parameter file is then passed to an R script `runGierer2D.R`. The parameter file is treated as a regular R script, but typically will just contain variable assignments.

The R script `runGierer2D.R` reads in the paramter file, which must contain values for three variables: `rgc.file`, `sc.file`, `op.file`. You can also override default values for other key parameters; any trailing strings at the end of the call to start the pipeline are written as-is to the parameter file. For example, this call will change the default value of `p0.fn` and `epsilon`.

```
ComparePhenotypeModelling('Math5','run',12, 0, 0, 'Gierer2D', ...
                          'p0.fn=4', 'epsilon=0.005')
```

The R script generates plots of the map and the values of `rho` and `r` for each SC neuron. See the script `runGierer2D.R` for further details.

4 Notation

the notation used in the paper has changed slightly compared to the code, so that all models used consistent notation.

Code	Manuscript
kappa,k	eta
r(x,y)	c(x,y)

5 About this document

This pdf is generated from an Emacs Org file.