PETRI NETS FOR MULTISCALE SYSTEMS BIOLOGY

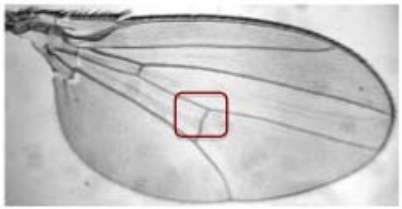
Monika Heiner

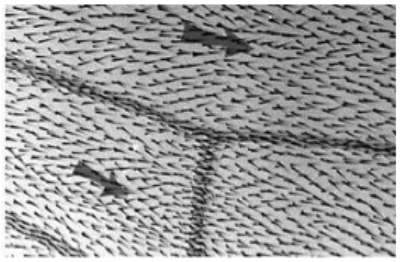
Brunel University, on sabbatical leave from

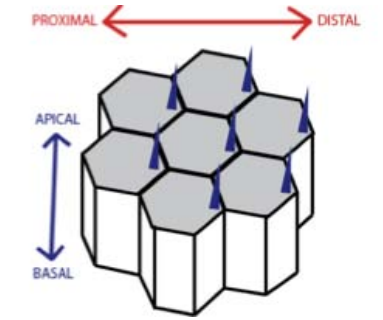
Brandenburg University of Technology Cottbus

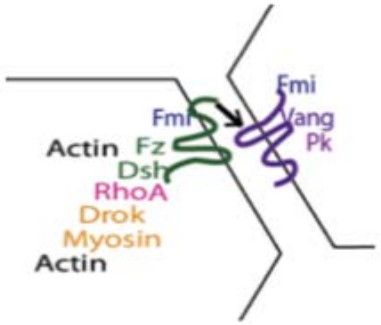
OUTLINE

- ***Prologue***
- **The Framework**
  - -> *(coloured) qualitative / stochastic / continuous Petri nets*
- **Coloured Petri Nets**
  - -> *Example 1: prey / predator systems*
  - -> *Example 2: repressilator*
- **Example 3: diffusion in SPACE - gradients**
  - -> *1D / 2D*
- **Example 4: multistrain cell colonies**
- **Summary**
- ***Epilogue***









PROLOGUE

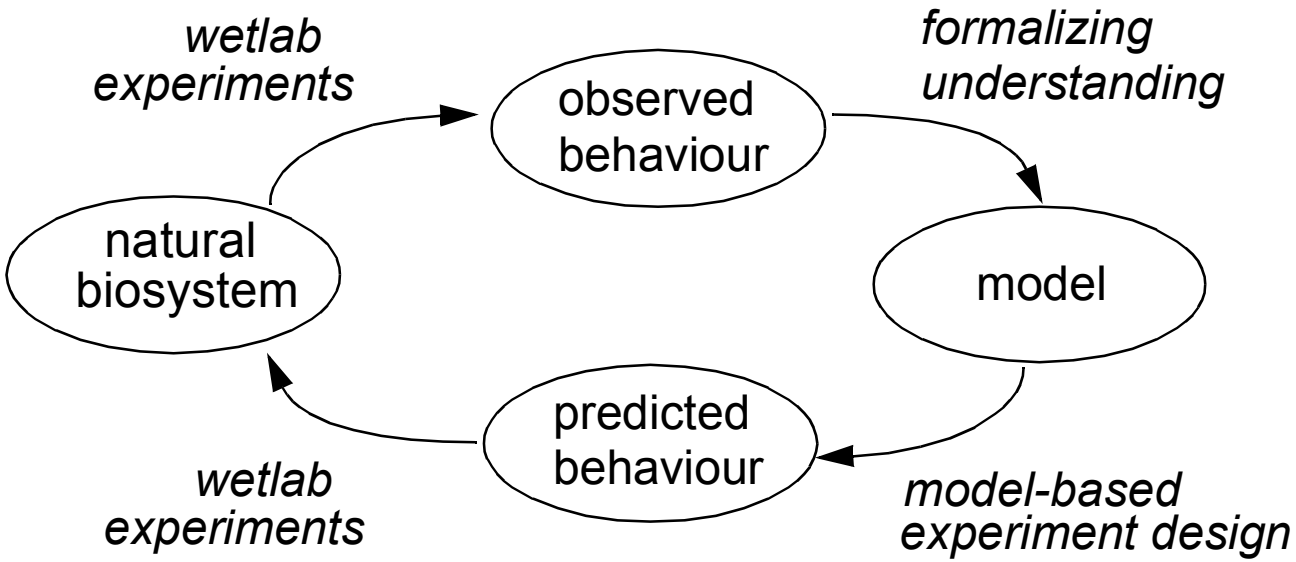
Spring in Messina Avenue

SUMMER AT KENNWOOD HOUSE

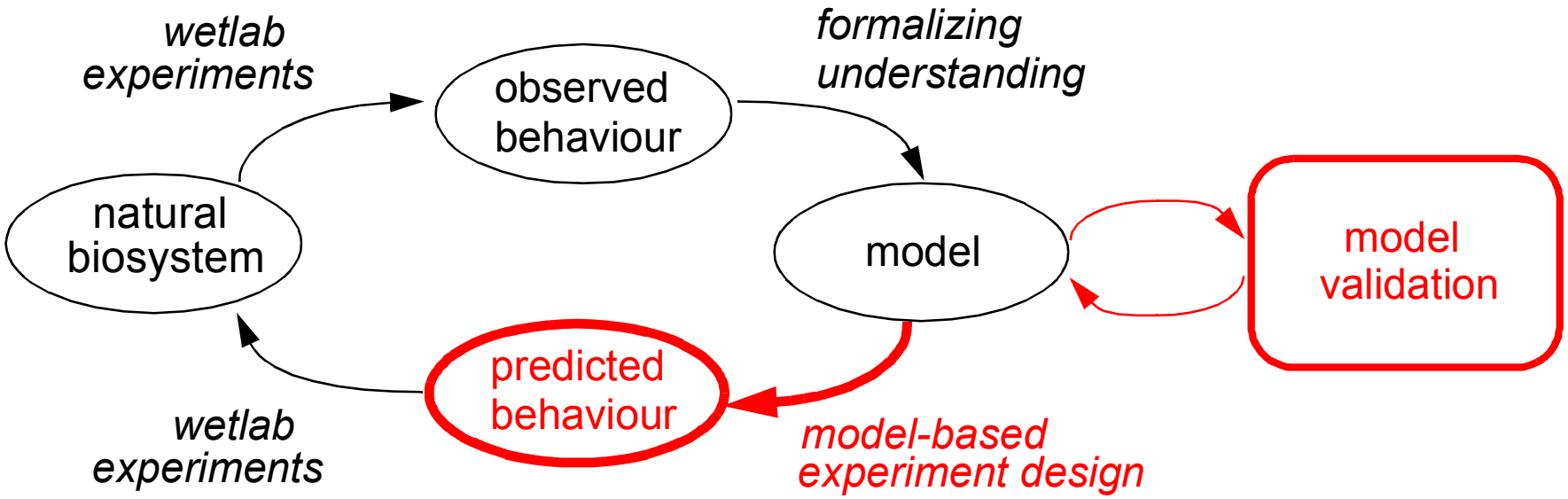
Autumn in Chelsea Physic Garden

THE FRAMEWORK

MODELLING = FORMAL KNOWLEDGE REPRESENTATION

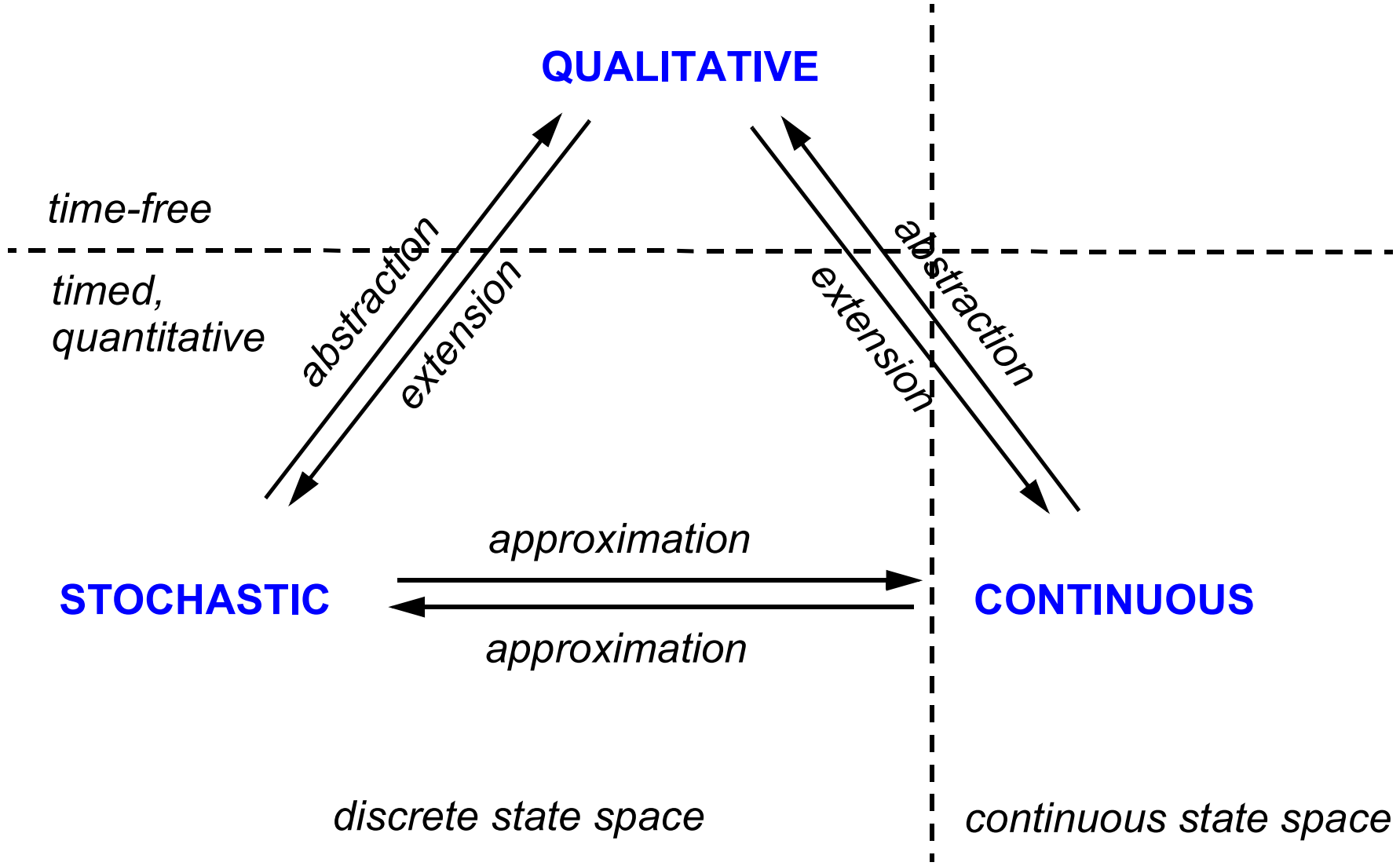


MODELLING = FORMAL KNOWLEDGE REPRESENTATION

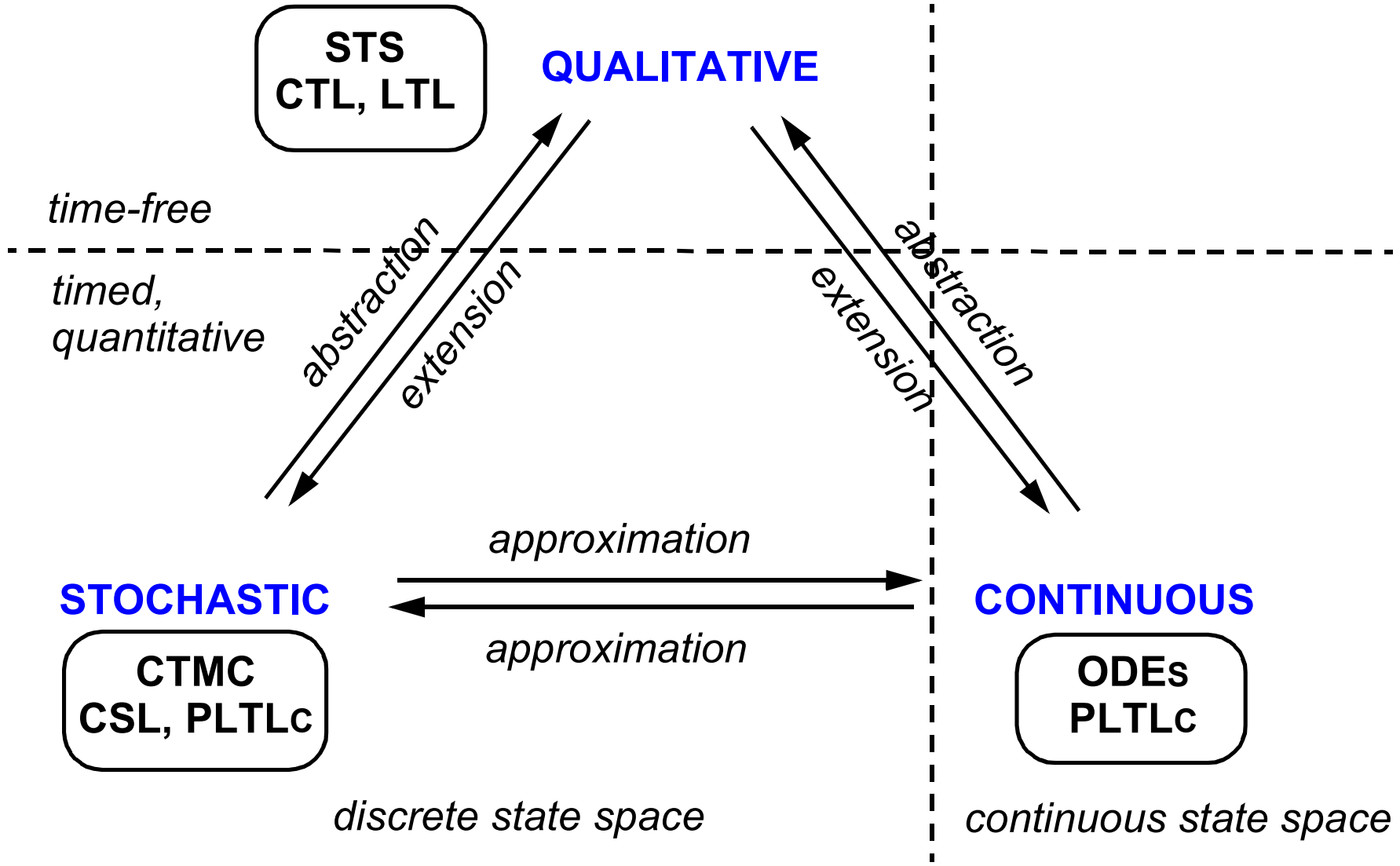


MODEL VALIDATION = CONFIDENCE INCREASE

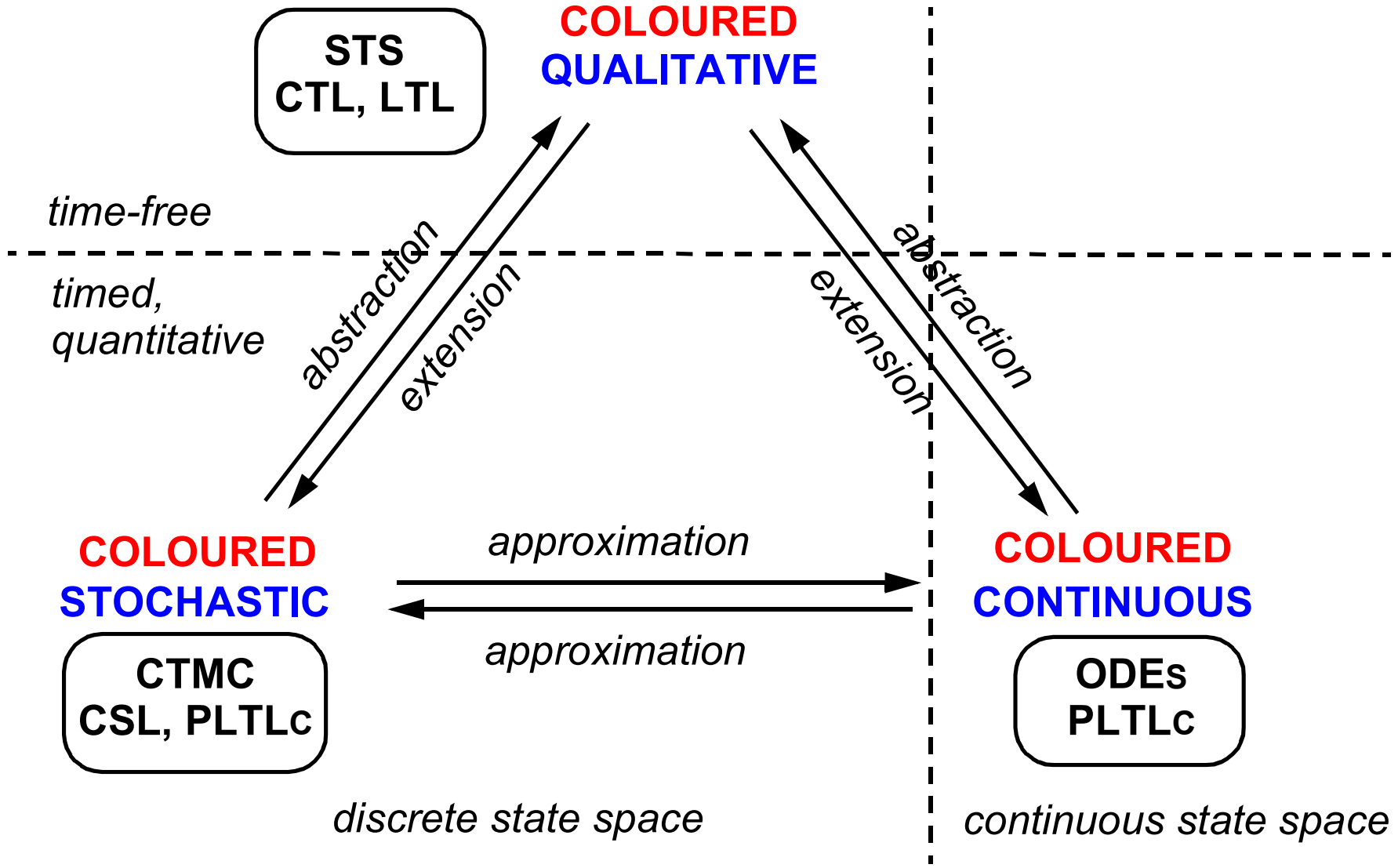
FRAMEWORK


QUALITATIVE
time-free
timed,
quantitative
abstraction
extension
abstraction
extension
approximation
STOCHASTIC
CONTINUOUS
approximation
discrete state space
continuous state space

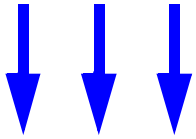
FRAMEWORK


STS
CTL, LTL
QUALITATIVE
time-free
timed,
quantitative
abstraction
extension
abstraction
extension
approximation
approximation
STOCHASTIC
CONTINUOUS
CTMC
CSL, PLTLc
ODEs
PLTLc
discrete state space
continuous state space

FRAMEWORK


STS
CTL, LTL
COLOURED
QUALITATIVE
time-free
timed,
quantitative
abstraction
extension
abstraction
extension
COLOURED
STOCHASTIC
approximation
approximation
COLOURED
CONTINUOUS
CTMC
CSL, PLTLc
ODEs
PLTLc
discrete state space
continuous state space

3 MODELS SHARING STRUCTURE

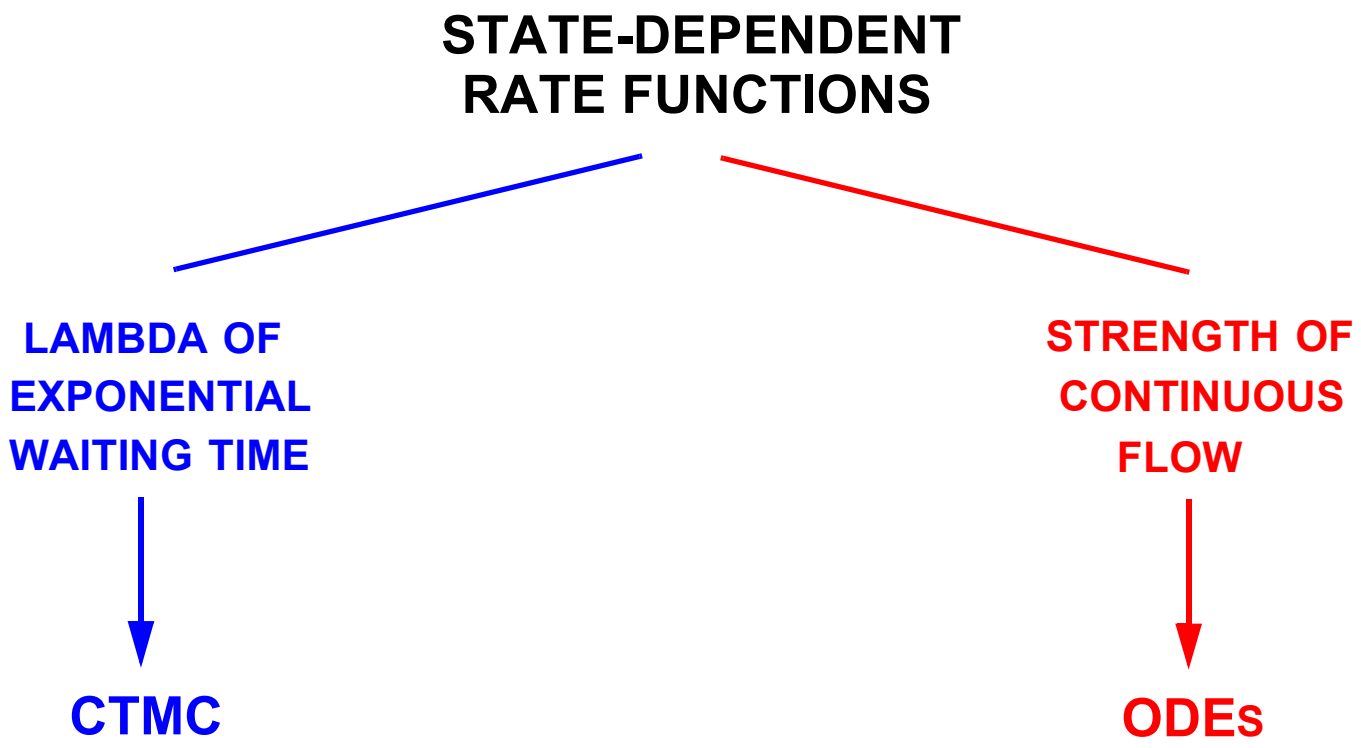


QUANTITATIVE MODEL = QUALITATIVE MODEL

+

RATE FUNCTIONS (KINETICS)

CRUCIAL POINT

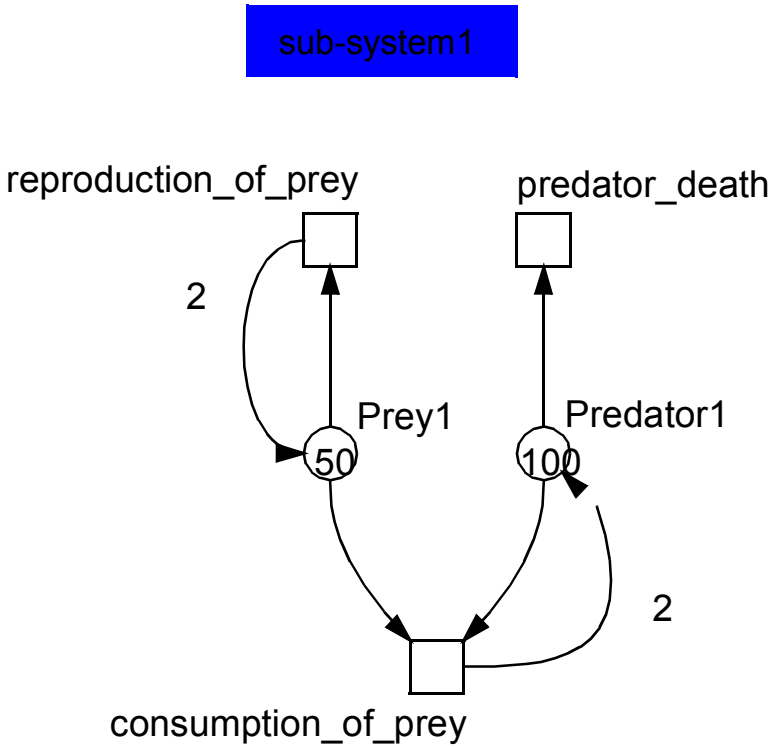


-> supported by, e.g., COPASI, Dizzy, ..., Snoopy

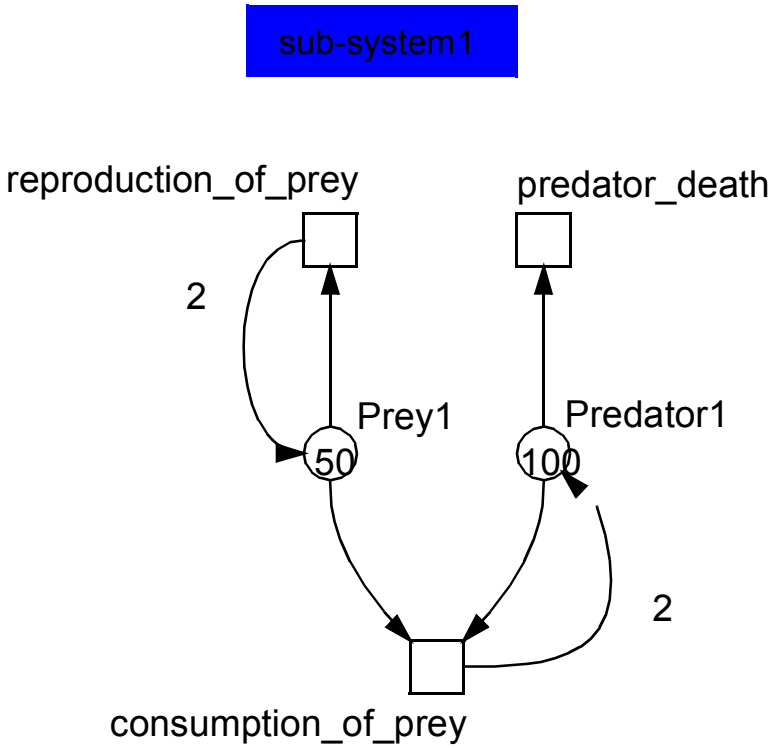
COLOURED PETRI NETS

monika.heiner@brunel.ac.uk

EX1: PREY - PREDATOR

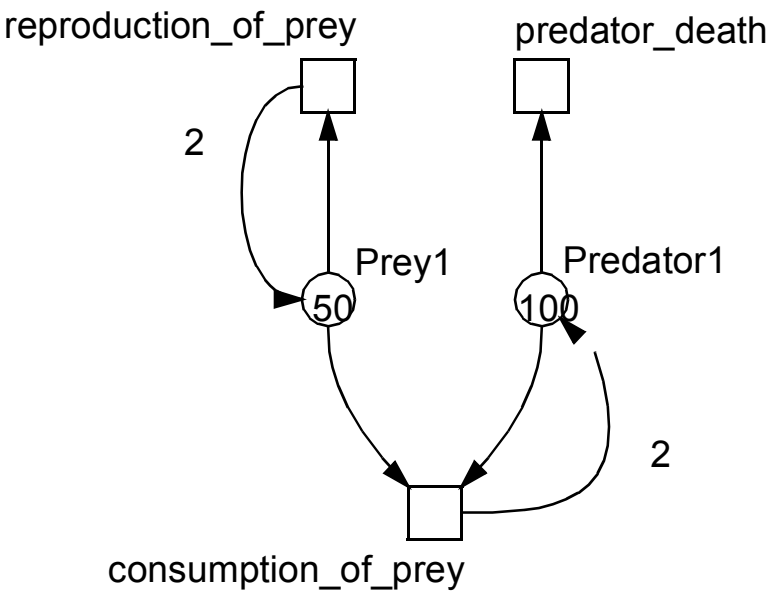


EX1: PREY - PREDATOR

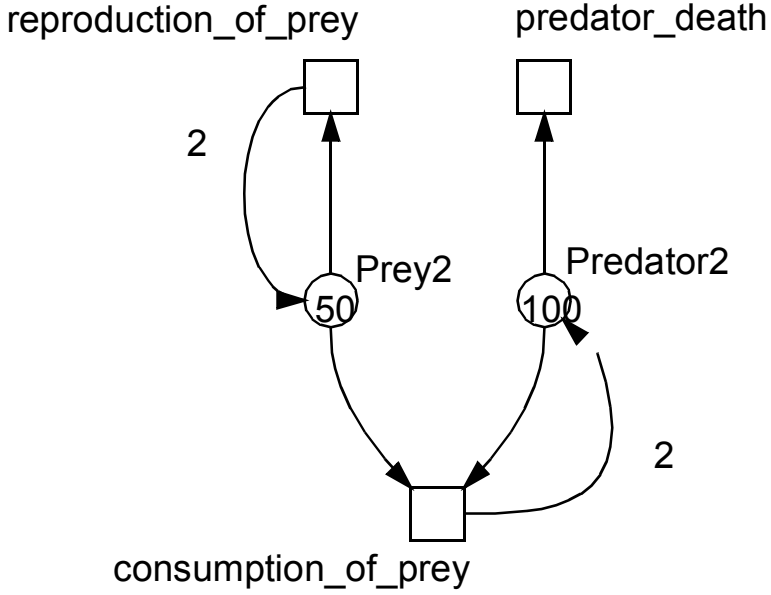


EX1: PREY - PREDATOR

sub-system1

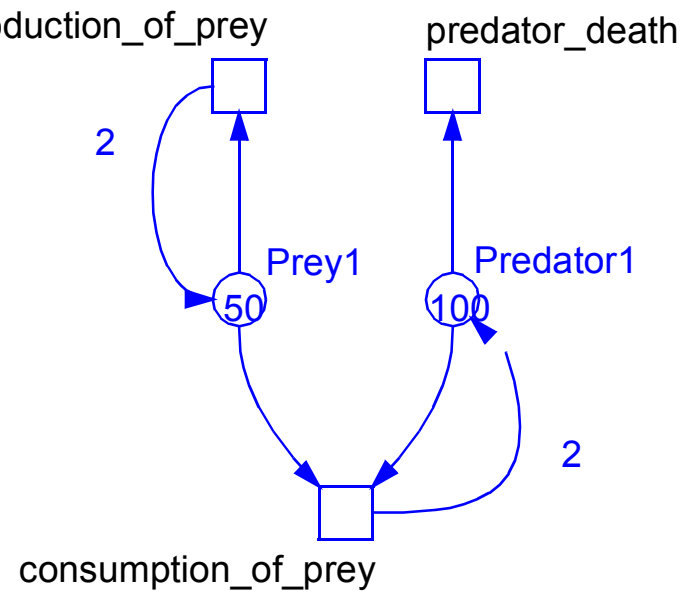


sub-system2

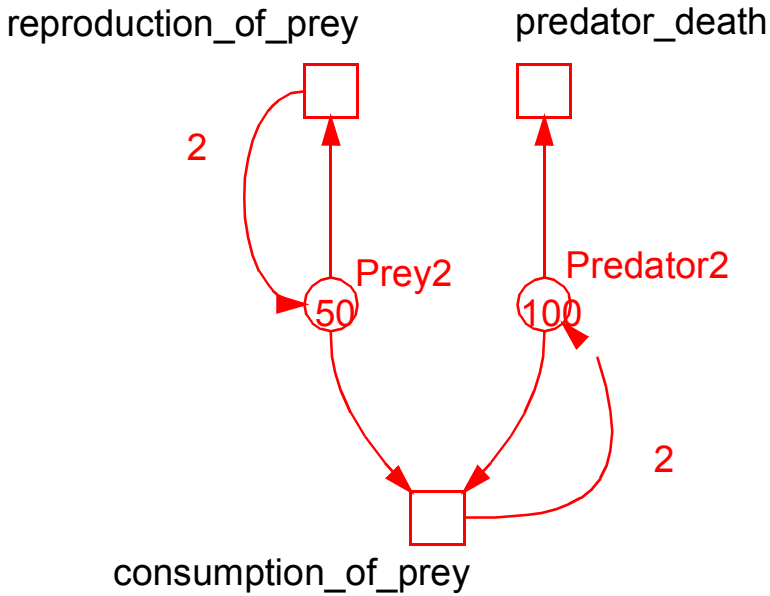


EX1: PREY - PREDATOR

sub-system1



sub-system2



EX1: PREY - PREDATOR

- **definitions**

  colourset *CS = 1-2;*

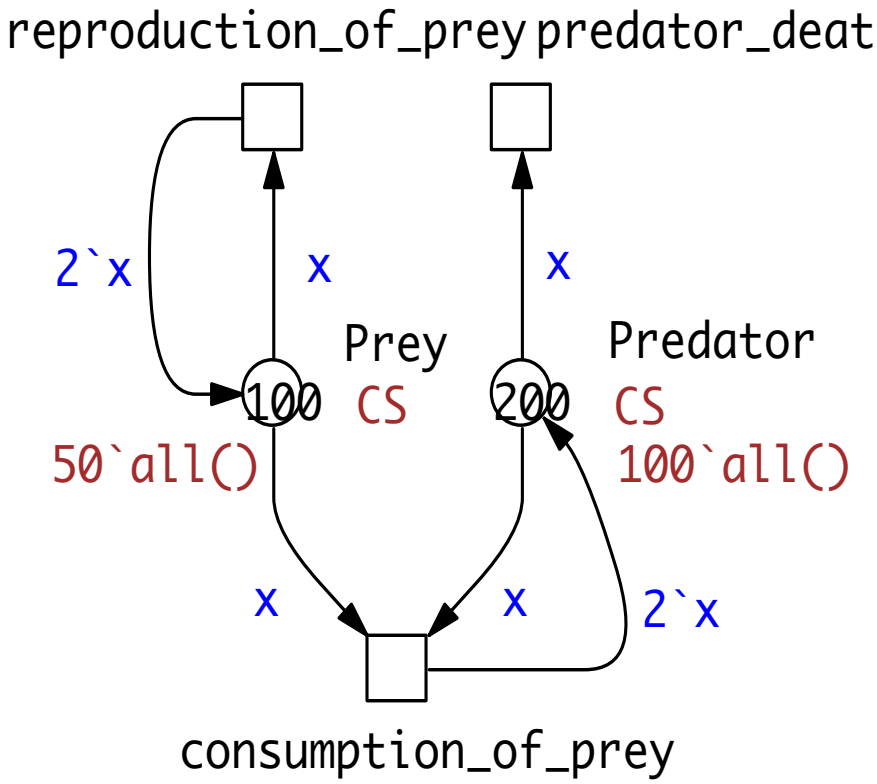
  var *x : CS;*

- **better:**

  const *SIZE = 2;*

  colourset *CS = 1-SIZE;*

  var *x : CS;*



EX1: PREY - PREDATOR

- **definitions**

  colourset *CS = 1-2;*

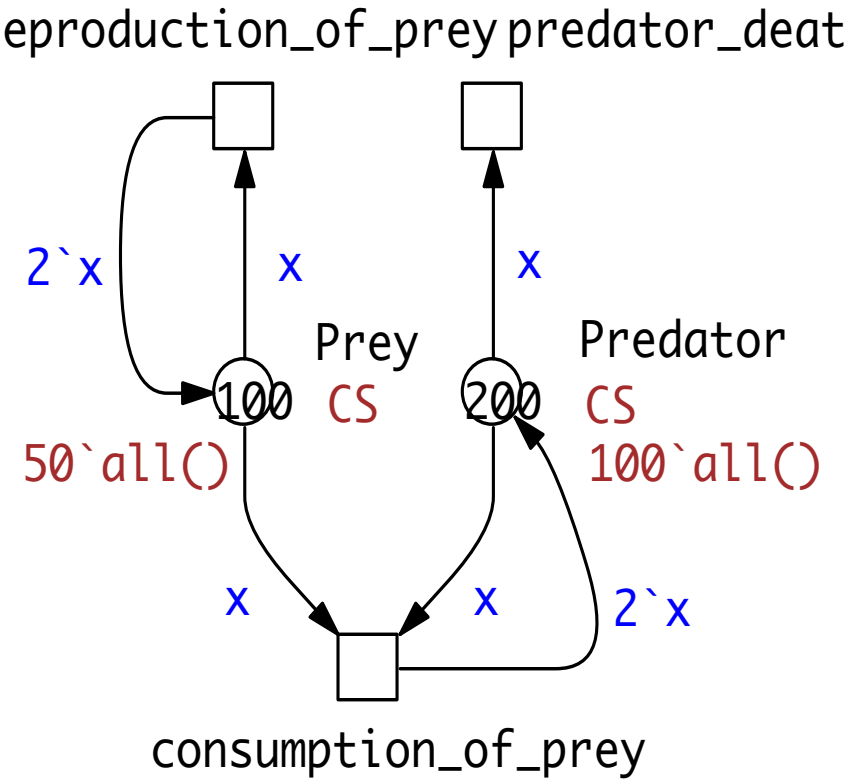
  var *x : CS;*

- **better:**

  const *SIZE = 2;*

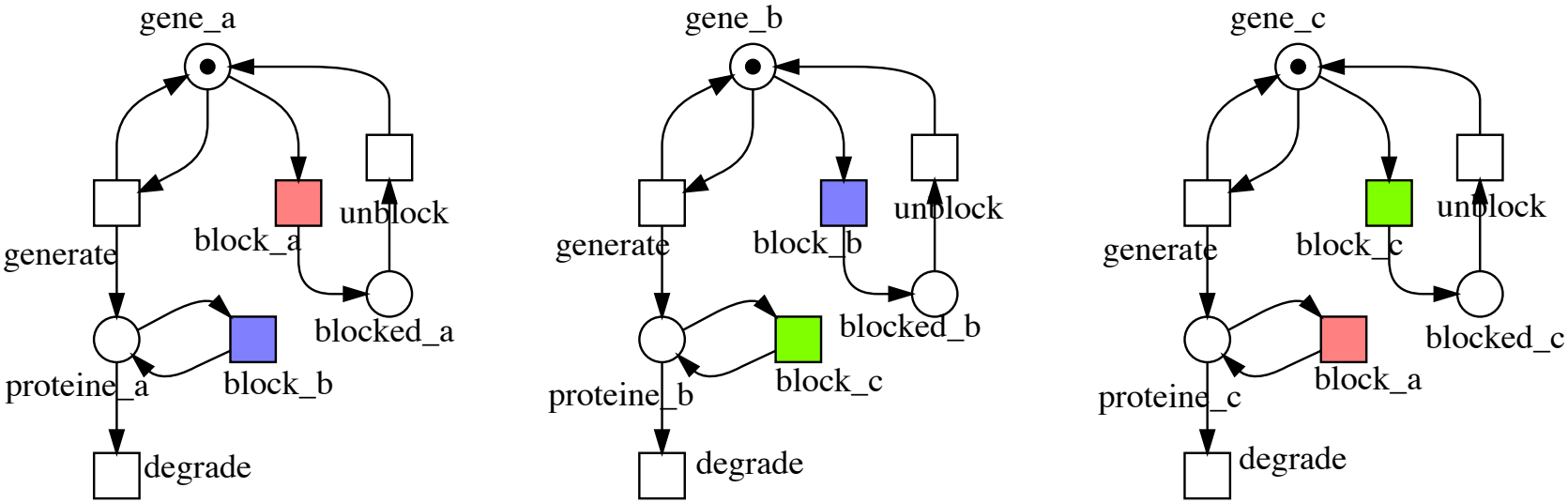
  colourset *CS = 1-SIZE;*

  var *x : CS;*

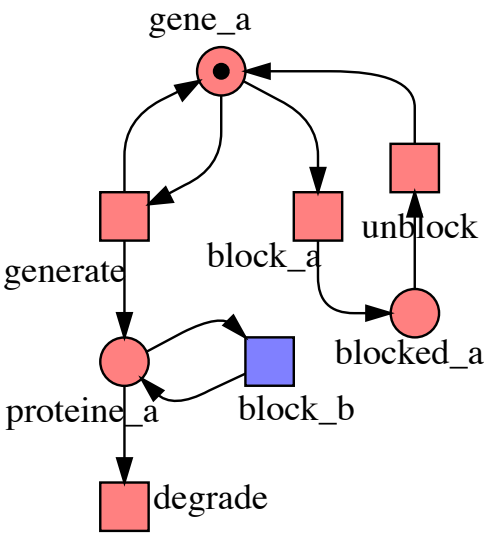


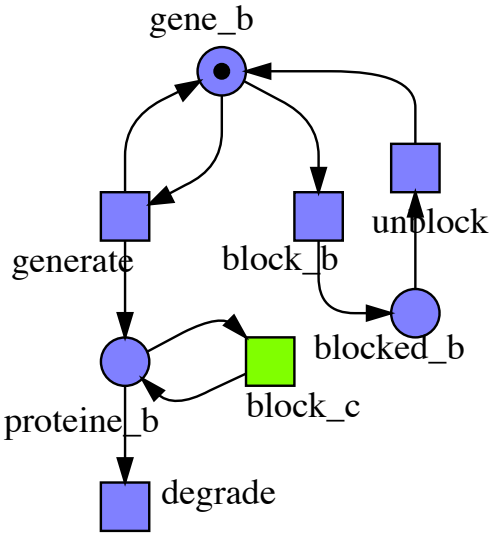
- **changing SIZE adapts the model to various scenarious**

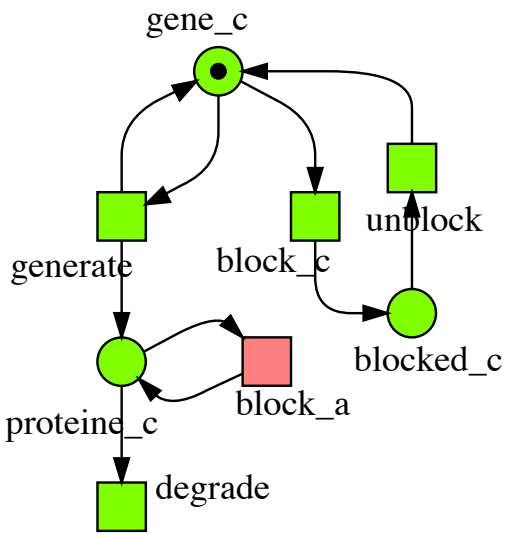
EX2: REPRESSILATOR



EX2: REPRESSILATOR





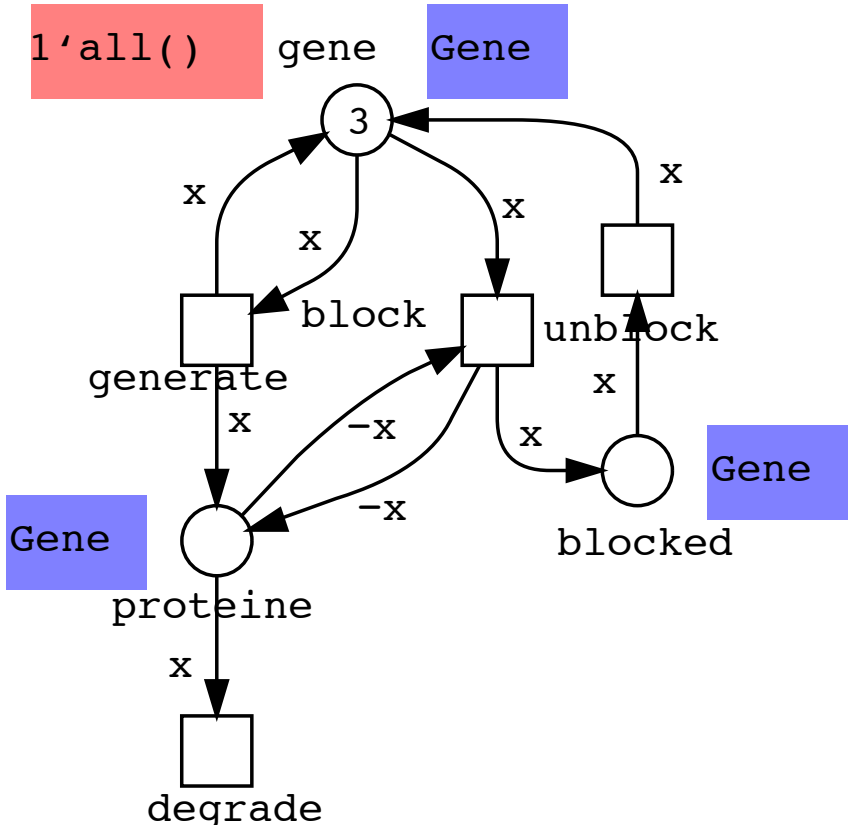


EX2: REPRESSILATOR

- **definitions**

  colorset *Gene = enum a-c;*

  var *x : Gene;*



EX2: REPRESSILATOR

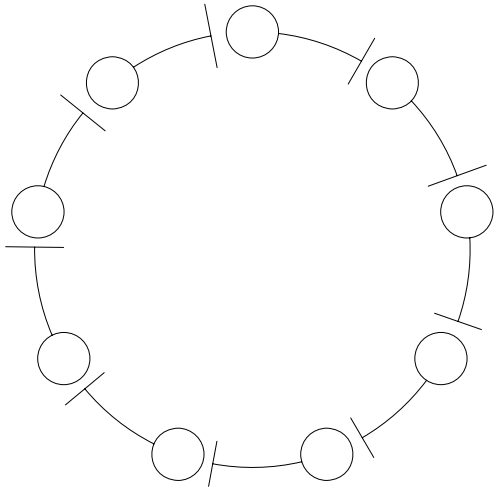
- **definitions**

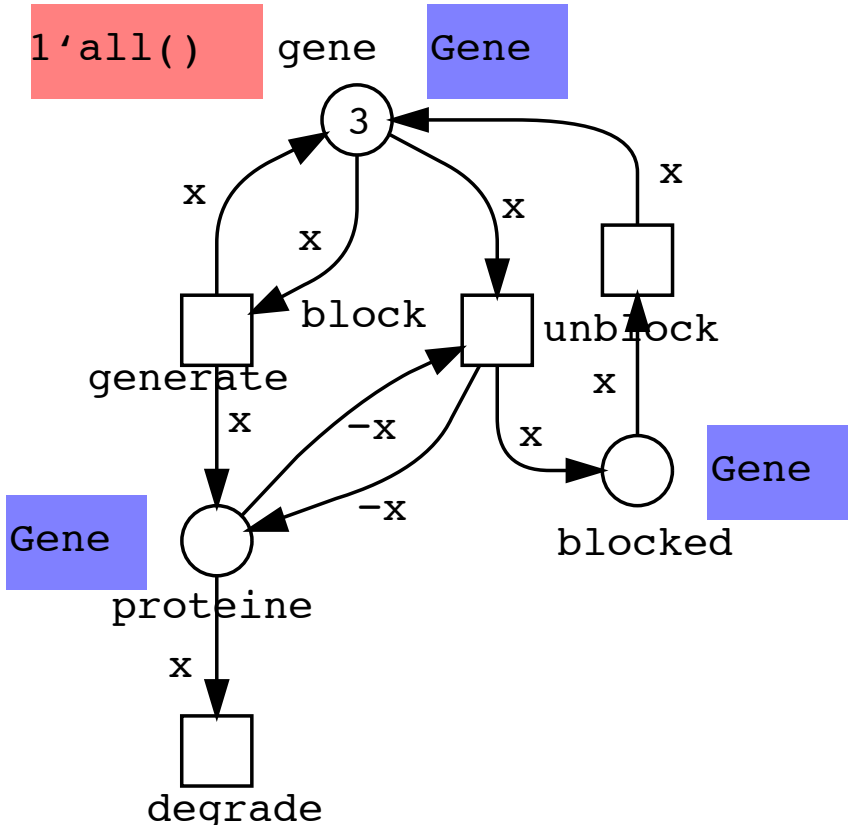
  colorset *Gene = enum a-c;*

  var *x : Gene;*

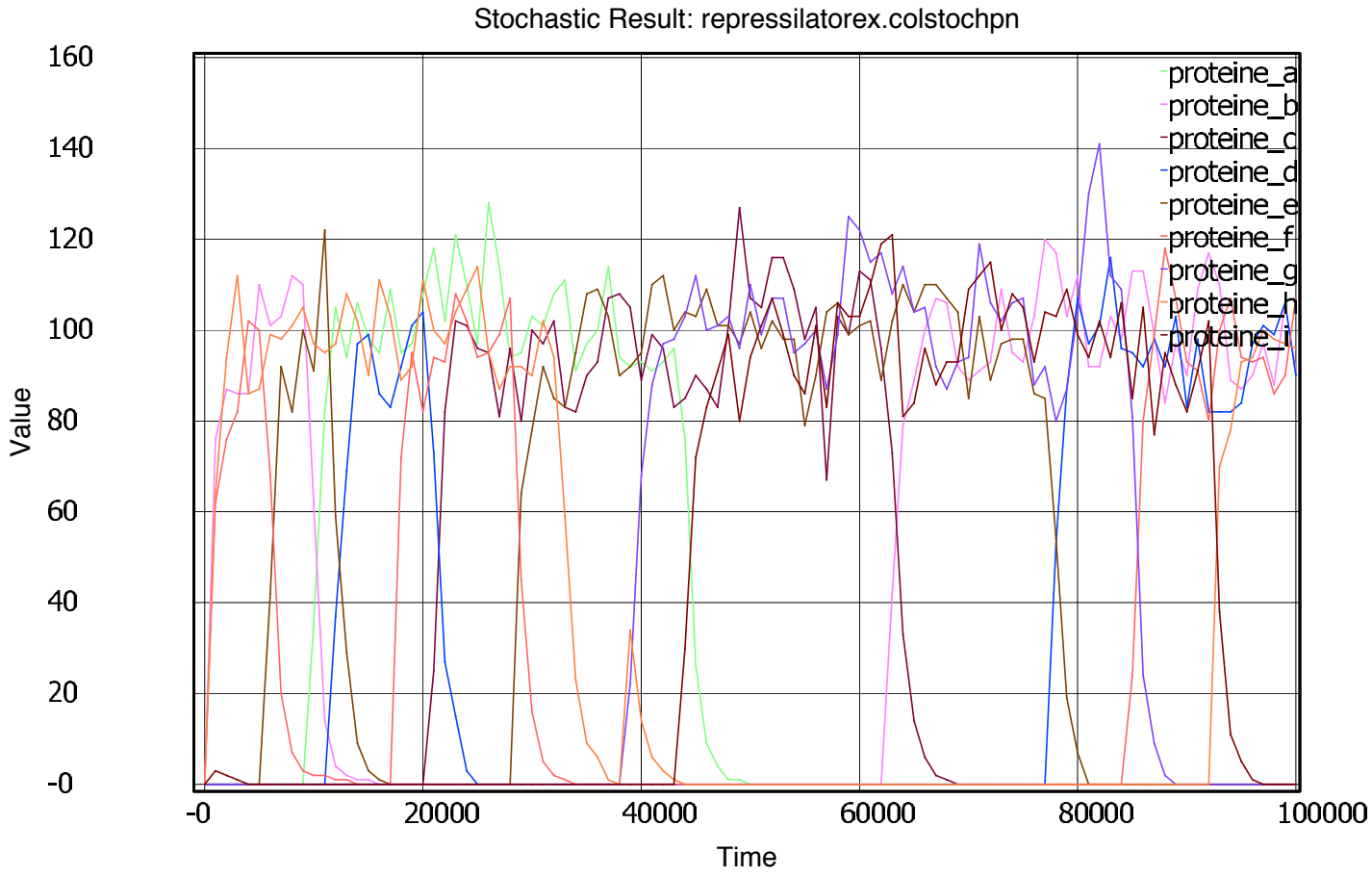
- **model scaling**

  colorset *Gene = enum a-i;*





EX2: REPRESSILATOR

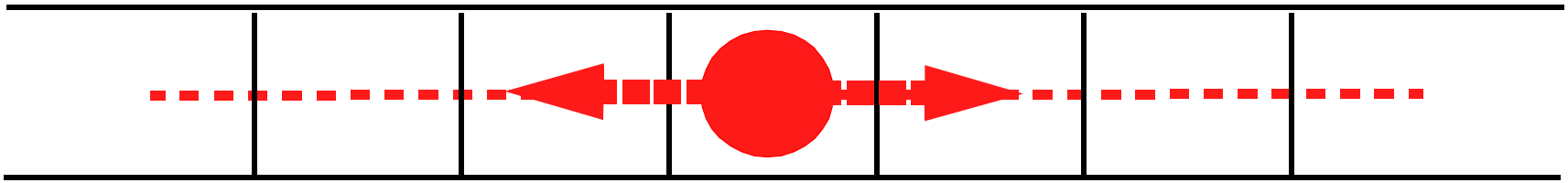


EXAMPLE:
DIFFUSION IN SPACE

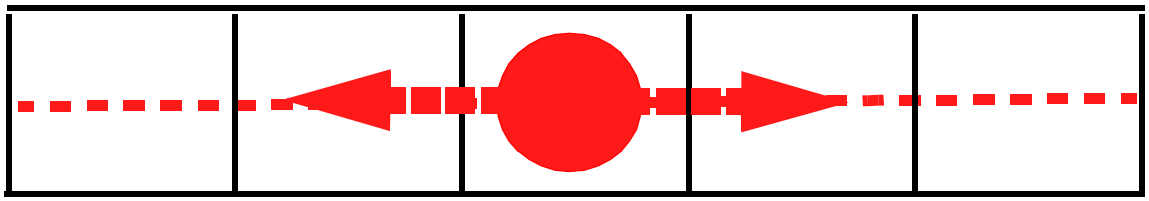
DISCRETE SPACE

Richmond, 13/09/2011

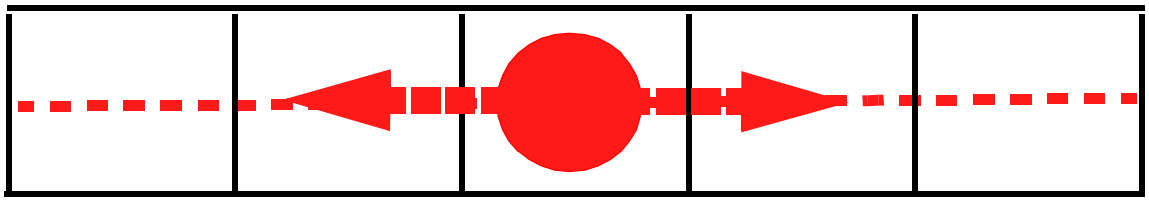
EX3: DIFFUSION - 1D

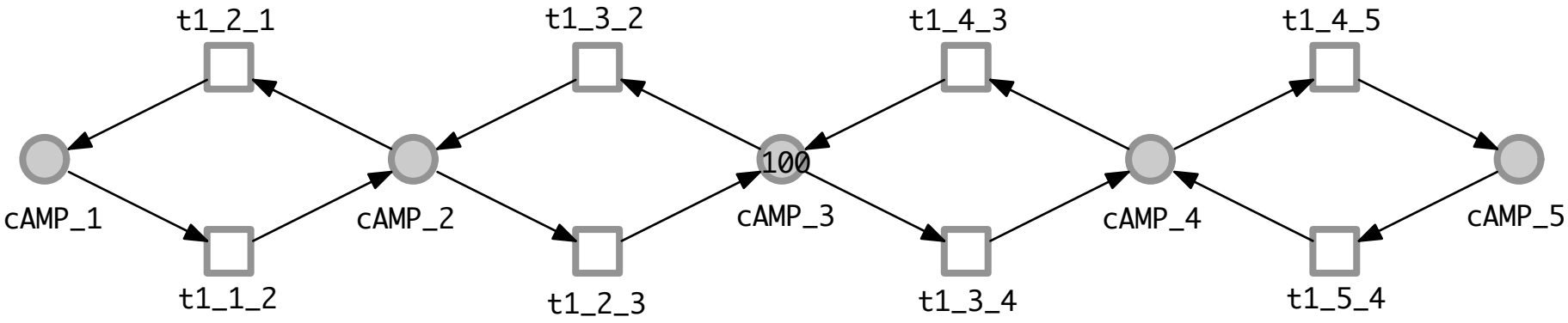


EX3: DIFFUSION - 1D



EX3: DIFFUSION - 1D





EX3: DIFFUSION - 1D

- **definitions**

```
const D1 = 5;            // grid size
colorset CS = 1-D1;      // grid positions
var x,y : CS;
```

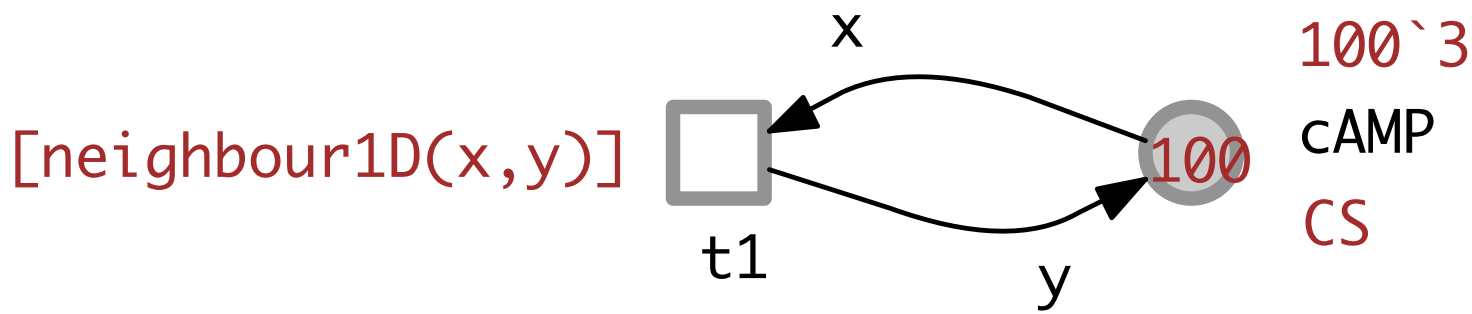
- **definitions**

```
const D1 = 5;              // grid size
colorset CS = 1-D1;        // grid positions
var x,y : CS;

function neighbour1D (CS x,a) bool:
  // a is neighbour of x
  ( a=x-1 | a=x+1)  & (1<=a) & (a<=D1);
```

Ex3: Diffusion - 1D

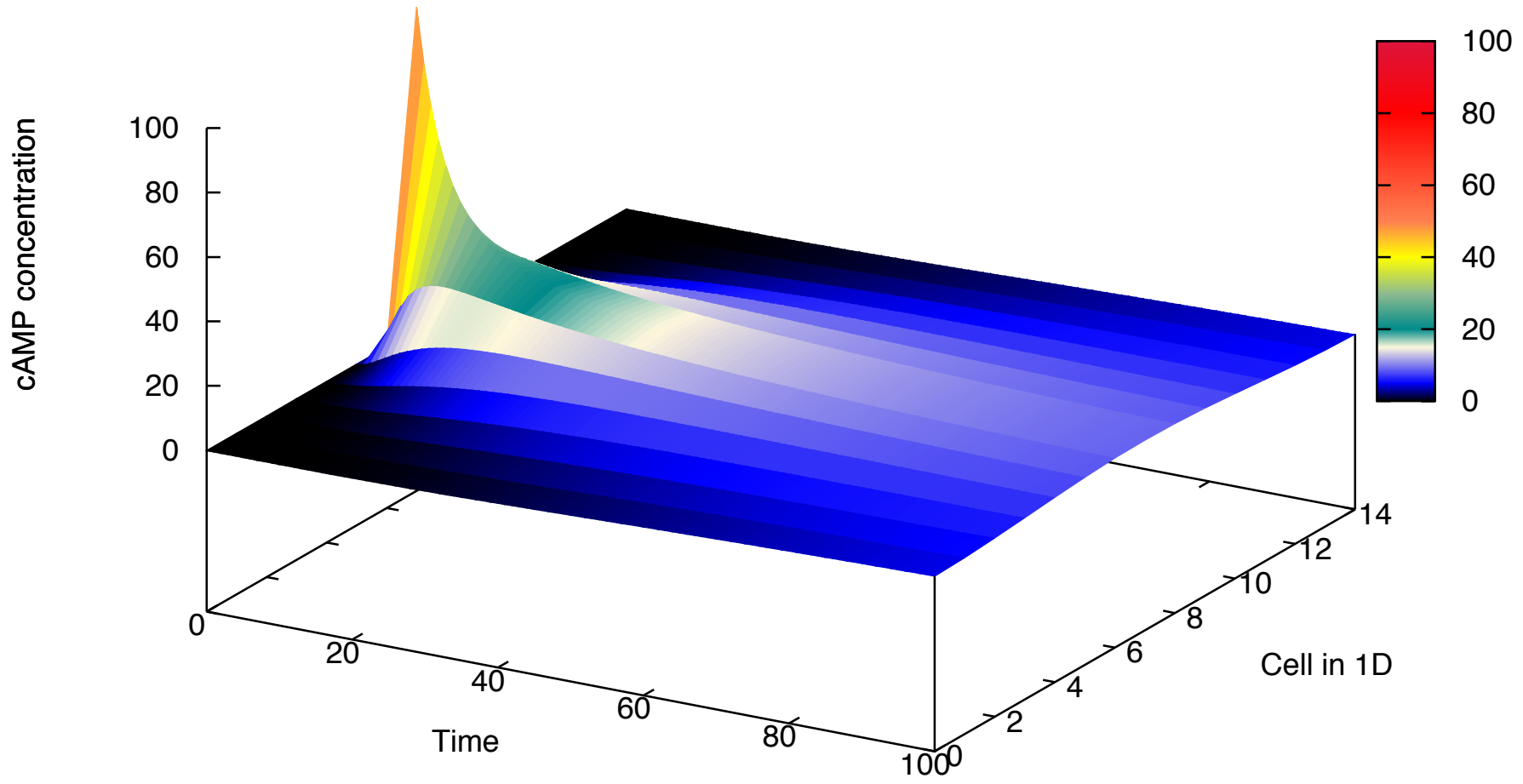
- **definitions**

```
const D1 = 5;              // grid size
colorset CS = 1-D1;        // grid positions
var x,y : CS;

function neighbour1D (CS x,a) bool:
  // a is neighbour of x
  ( a=x-1 | a=x+1)  & (1<=a) & (a<=D1);
```



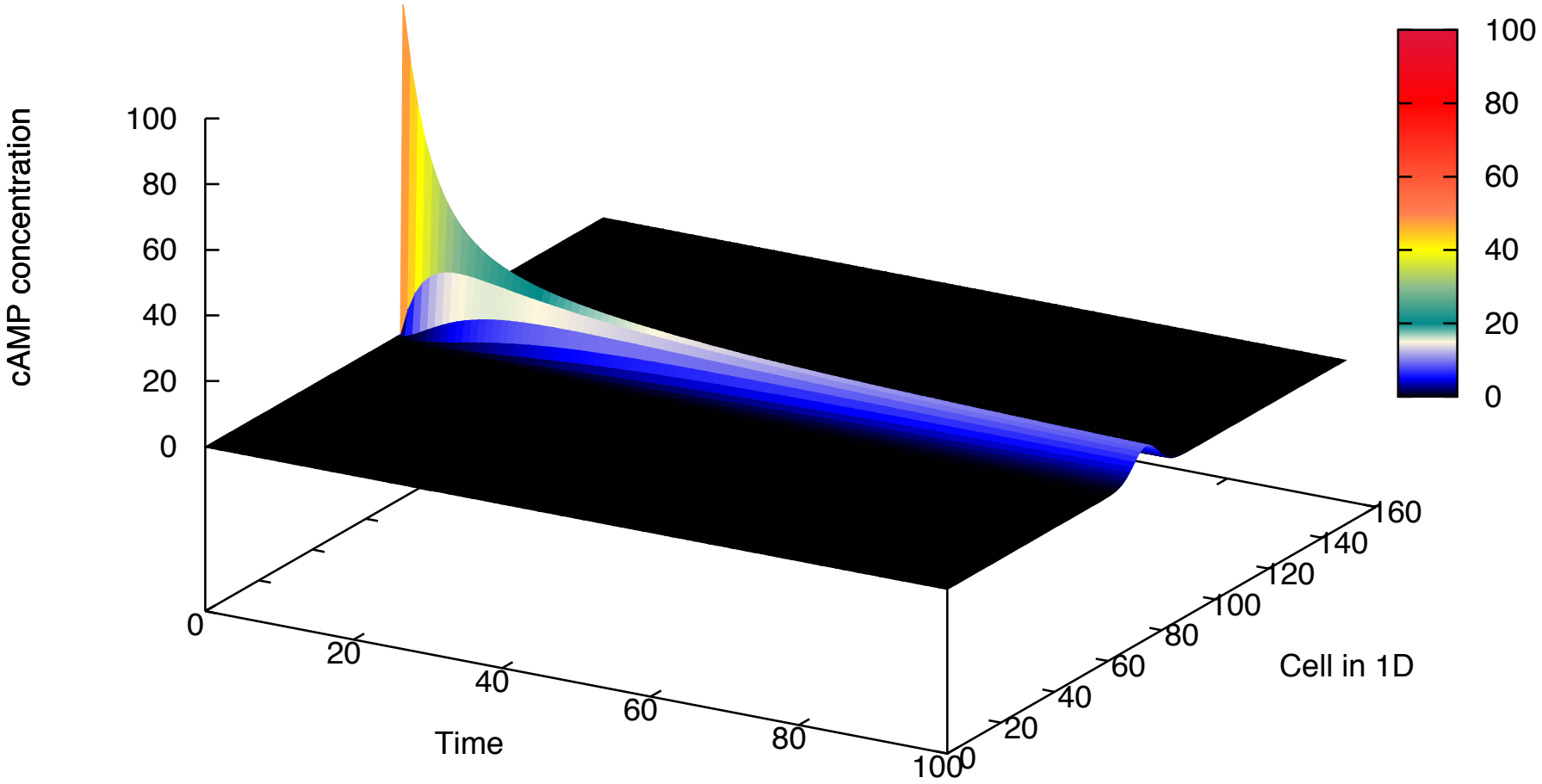
$$
\begin{aligned}
\frac{dc_1}{dt} &= k \cdot c_2 - k \cdot c_1 \\
\frac{dc_2}{dt} &= k \cdot c_1 + k \cdot c_3 - 2 \cdot k \cdot c_2 \\
\frac{dc_3}{dt} &= k \cdot c_2 + k \cdot c_4 - 2 \cdot k \cdot c_3 \\
\frac{dc_4}{dt} &= k \cdot c_3 + k \cdot c_5 - 2 \cdot k \cdot c_4 \\
\frac{dc_5}{dt} &= k \cdot c_4 - k \cdot c_5
\end{aligned}
$$

EX3: DIFFUSION - 1D



15 GRID POSITIONS

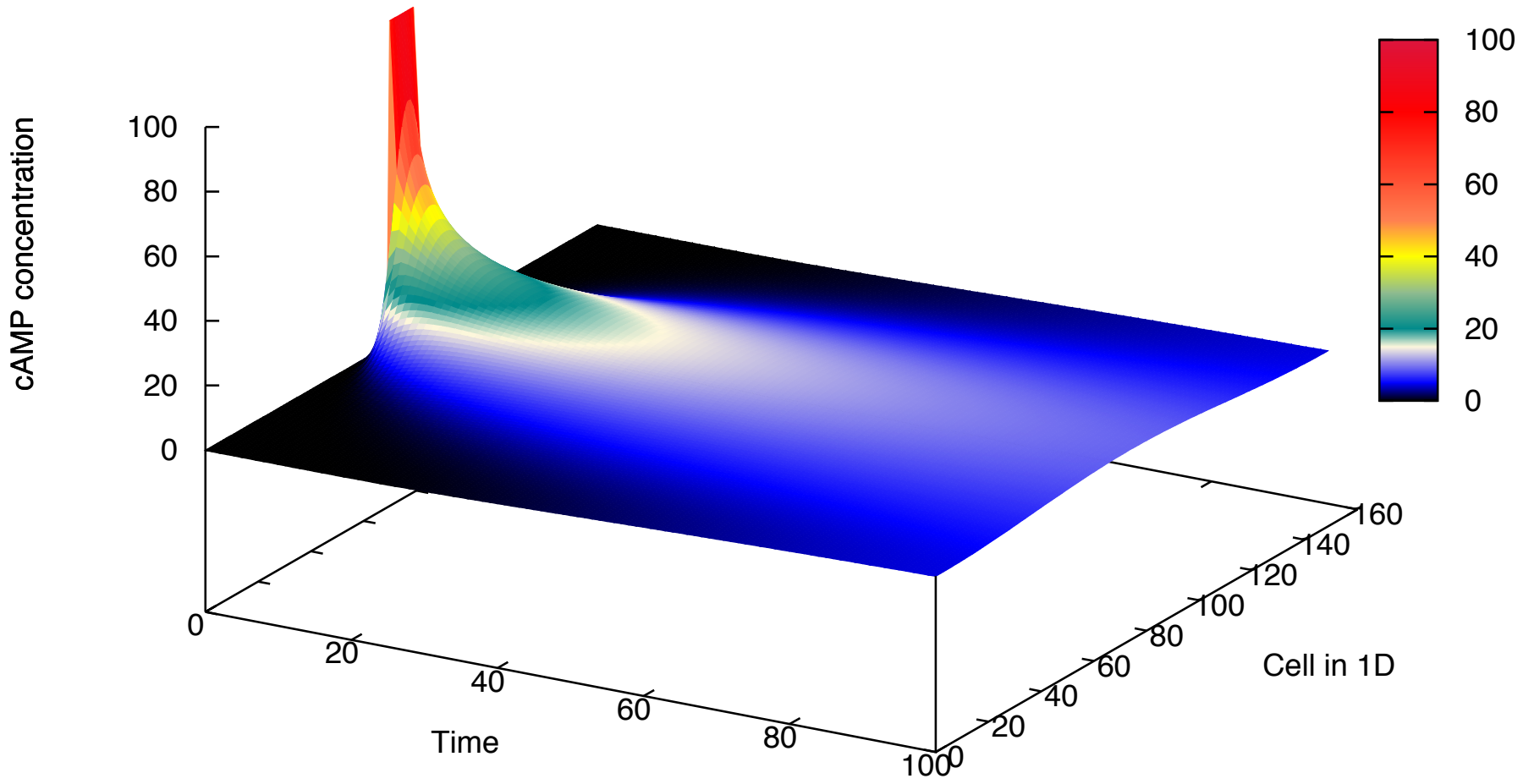
EX3: DIFFUSION - 1D



150 GRID POSITIONS, NO SCALING

monika.heiner@brunel.ac.uk

EX3: DIFFUSION - 1D

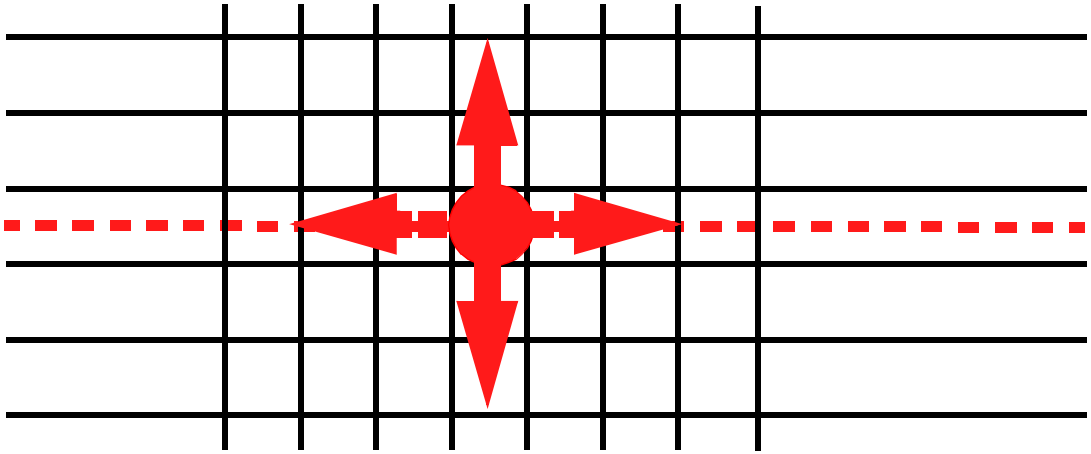


150 GRID POSITIONS, SCALING OF INITIAL MARKING AND RATES

monika.heiner@brunel.ac.uk

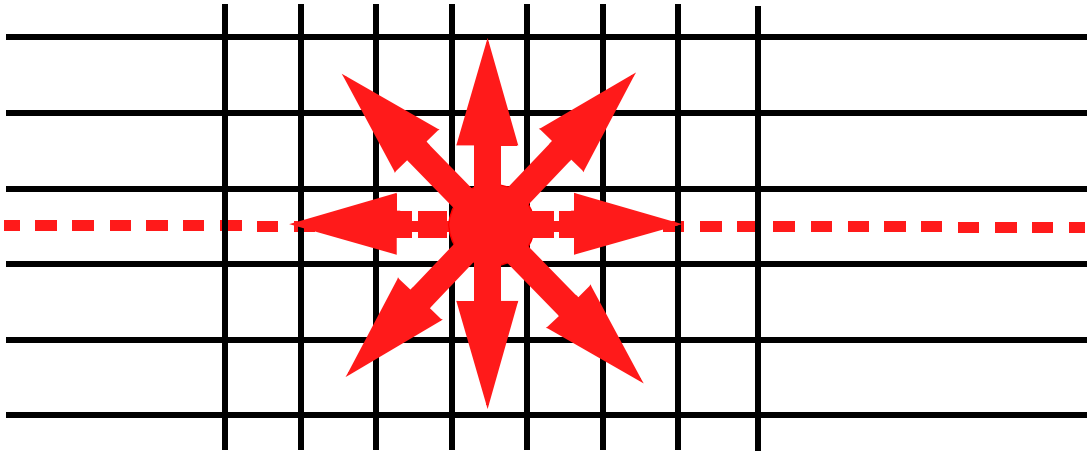
EX3: DIFFUSION - 2D

❑ SCHEME

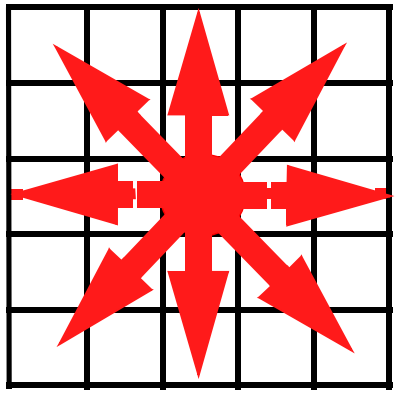


EX3: DIFFUSION - 2D

❑ SCHEME

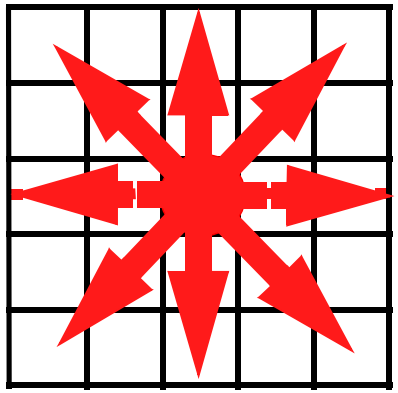


❑ SCHEME



EX3: DIFFUSION - 2D

SCHEME

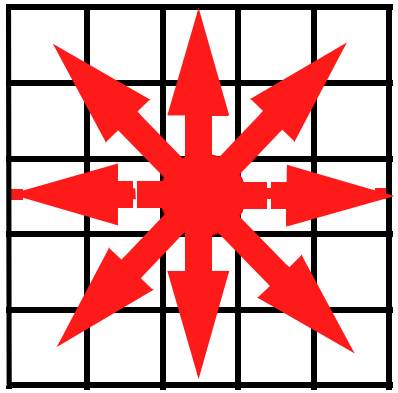


definitions

```
const D1 = 5;        // grid size first dimension
const D2 = 5;        // grid size second dimension
```

EX3: DIFFUSION - 2D

SCHEME

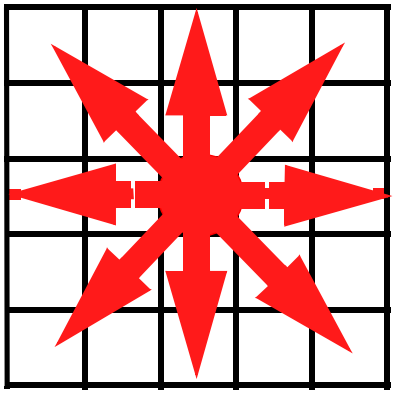


definitions

```
const D1 = 5;                    // grid size first dimension
const D2 = 5;                    // grid size second dimension

colorset CD1 = 1-D1;             // row index
colorset CD2 = 1-D2;             // column index
colorset Grid2D = CD1 x CD2;     // 2D grid
```

EX3: DIFFUSION - 2D

SCHEME

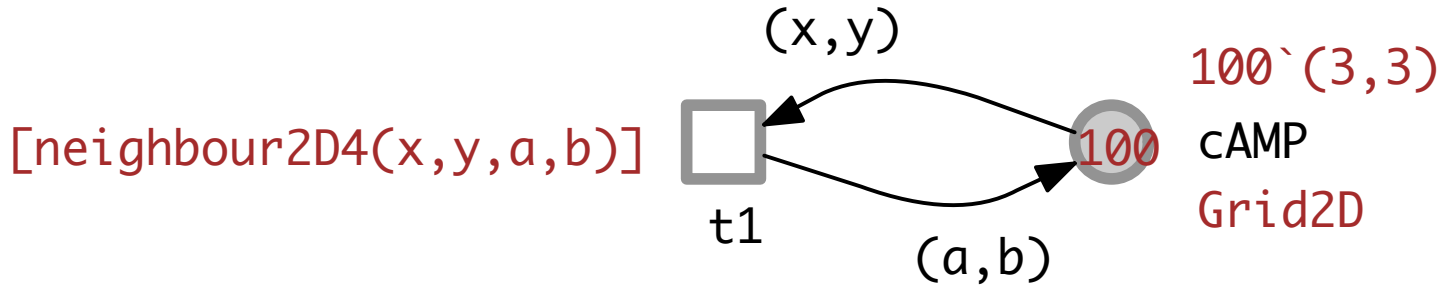


definitions

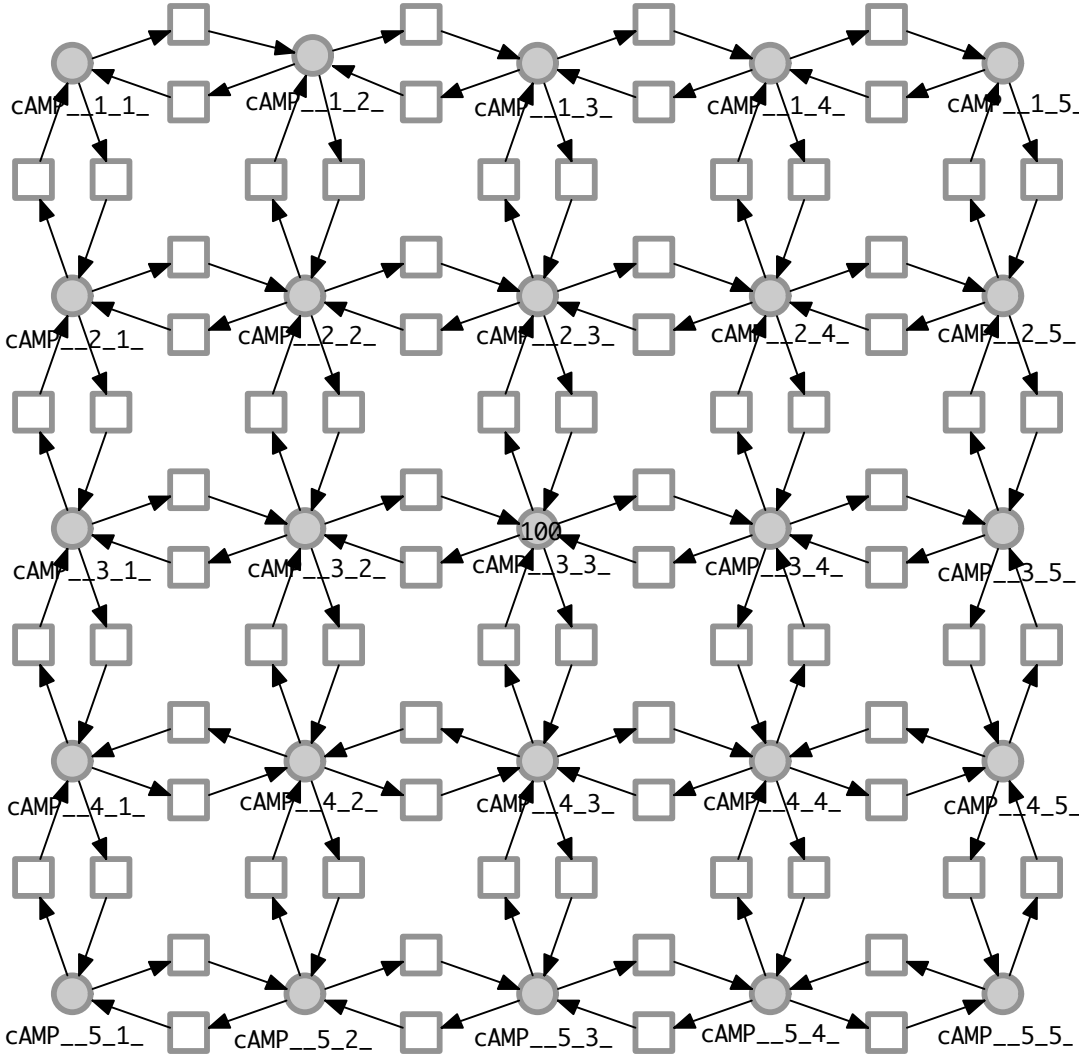
```
const D1 = 5;                    // grid size first dimension
const D2 = 5;                    // grid size second dimension

colorset CD1 = 1-D1;             // row index
colorset CD2 = 1-D2;             // column index
colorset Grid2D = CD1 x CD2;     // 2D grid

var x, a : CD1;
var y, b : CD2;
```

- **four neighbours**

```
function neighbour2D4 (CD1 x, CD2 y, CD1 a, CD2 b) bool:
  // (a,b) is one of the up to four neighbours of (x,y)
     (a=x & b=y-1) | (a=x & b=y+1)
  |  (b=y & a=x-1) | (b=y & a=x+1);
```



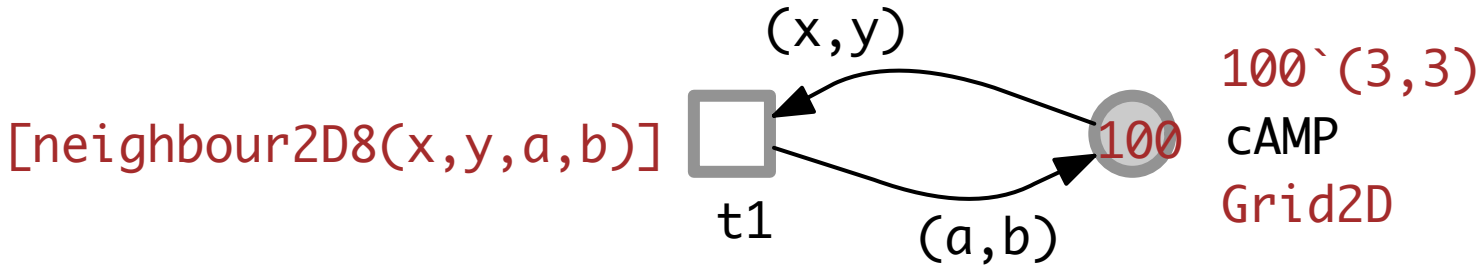
EX3: DIFFUSION - 2D4 NEIGHBOURHOOD



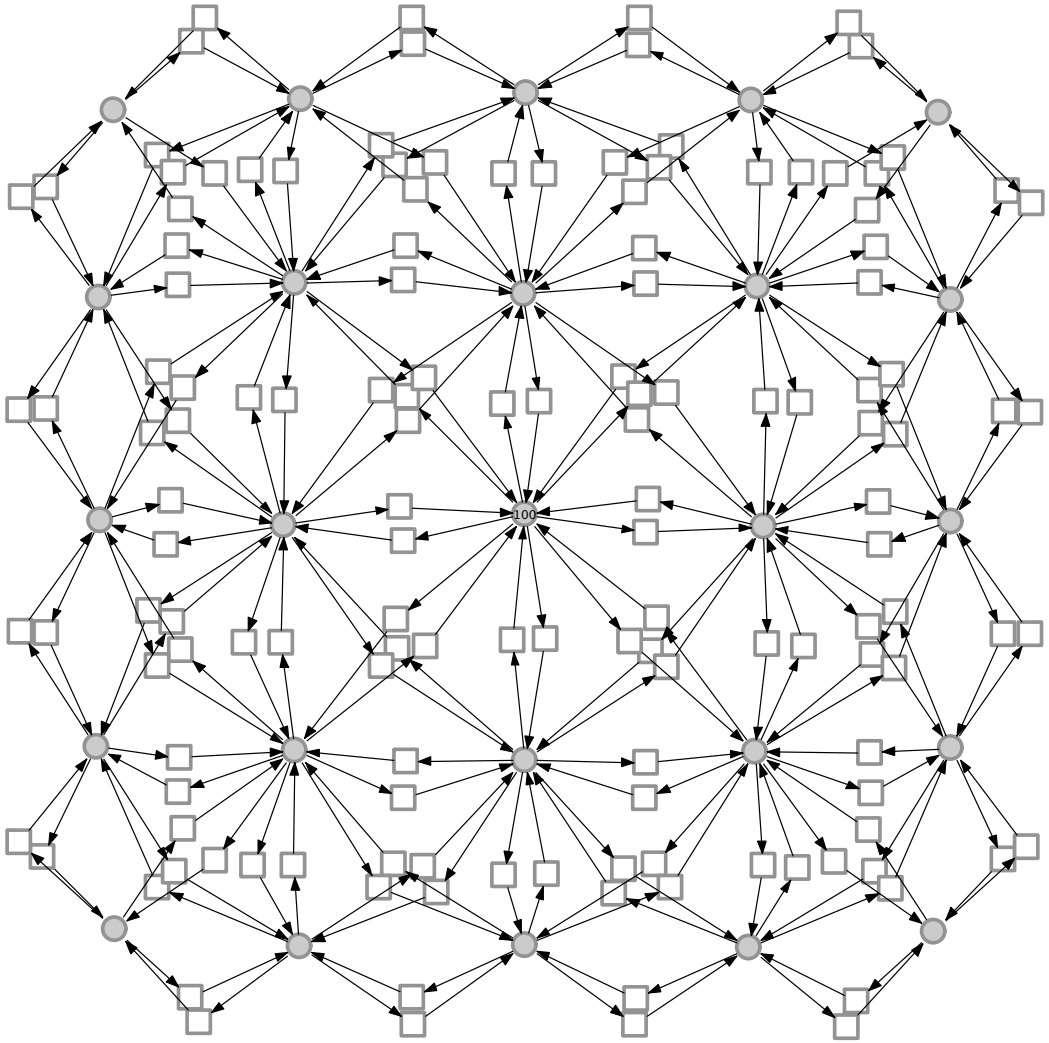
EX3: DIFFUSION - 2D8 NEIGHBOURHOOD

- **eight neighbours**

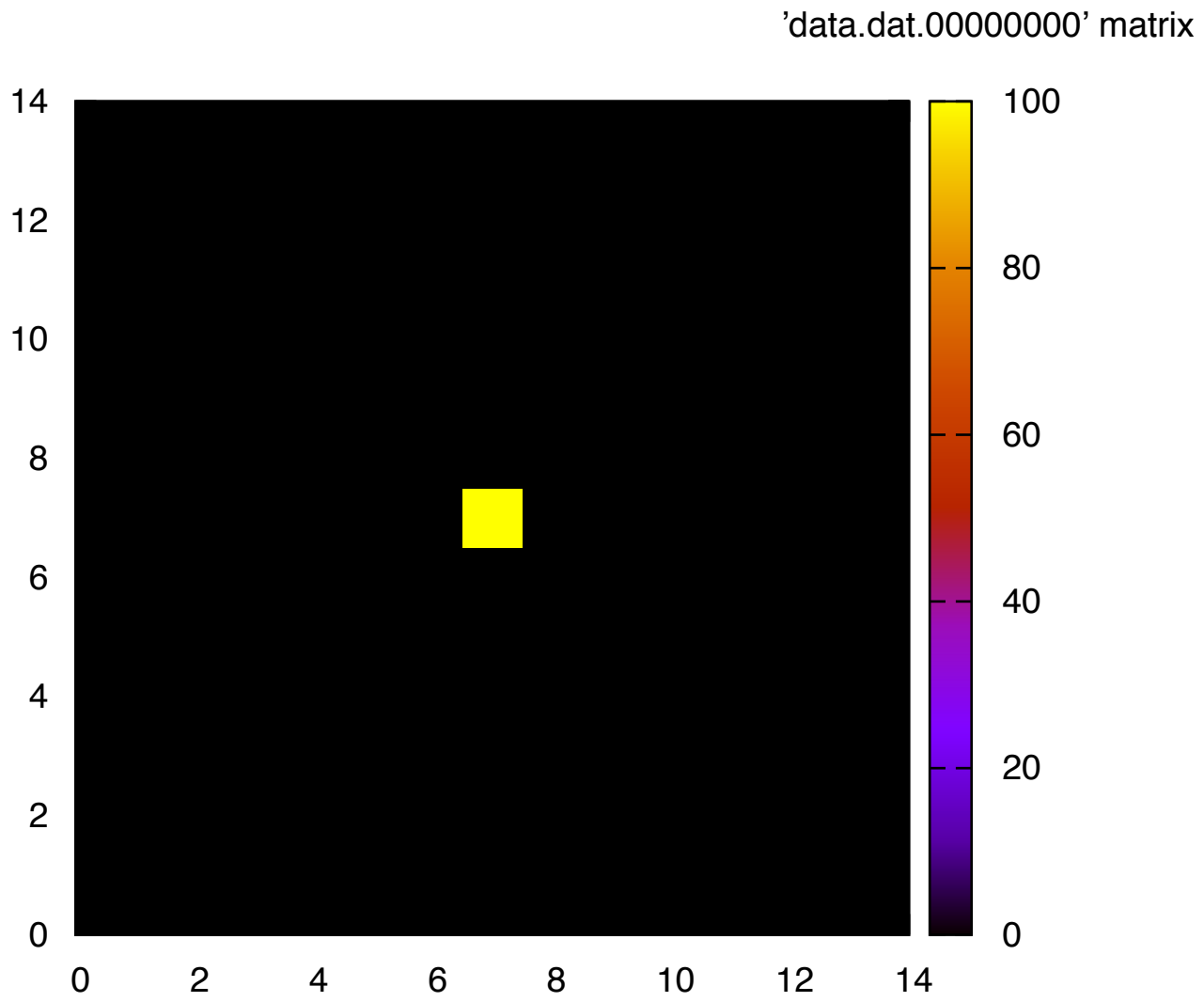
```
function neighbour2D8 (CD1 x, CD2 y, CD1 a, CD2 b) bool:
  // (a,b) is one of the up to eight neighbours of (x,y)
    (a=x-1 | a=x | a=x+1) & (b = y-1 | b=y | b=y+1)
  & (!(a=x & b=y))
  & (1<=a & a<=D1) & (1<=b & b<=D2);
```



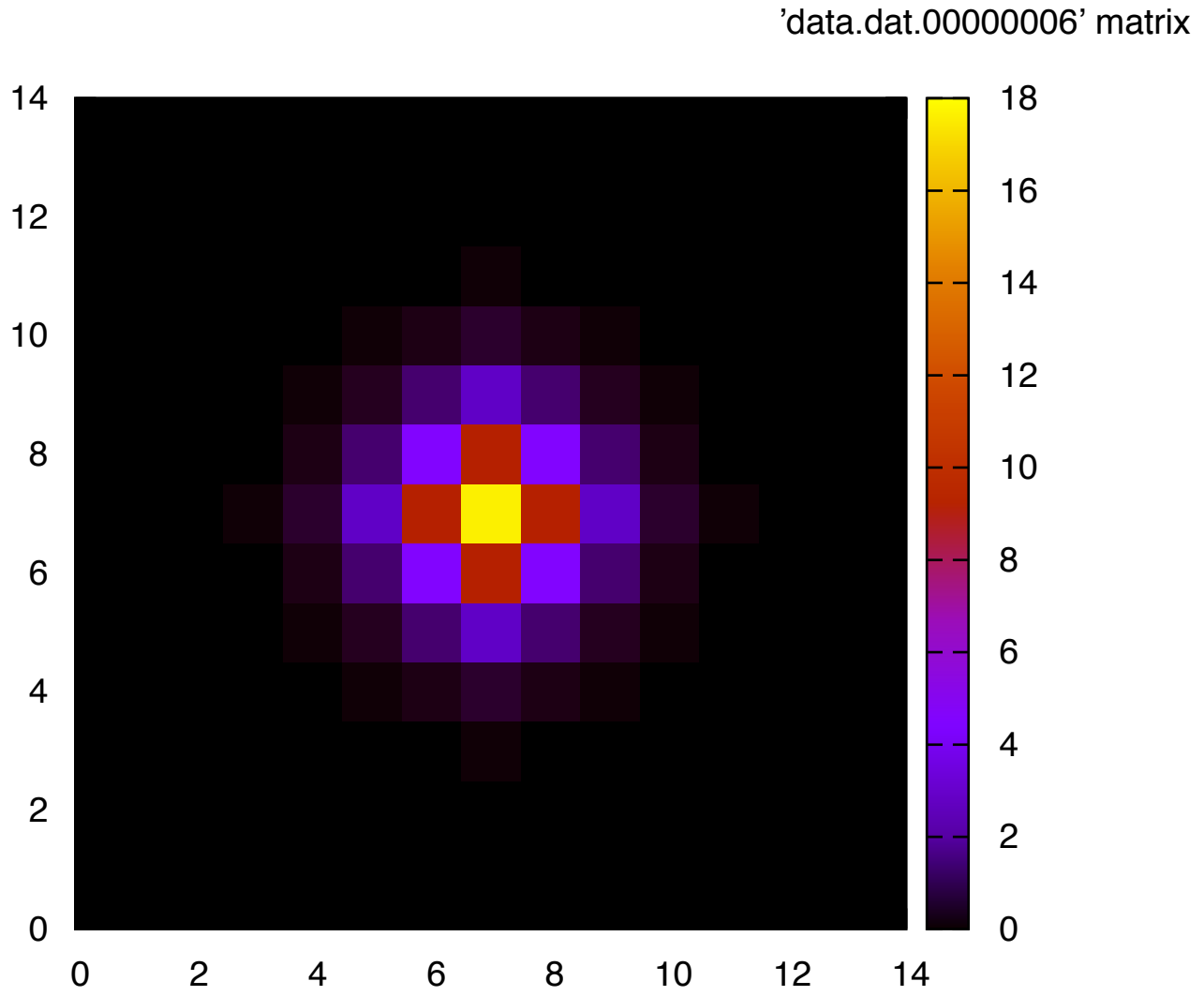
EX3: DIFFUSION - 2D8 NEIGHBOURHOOD



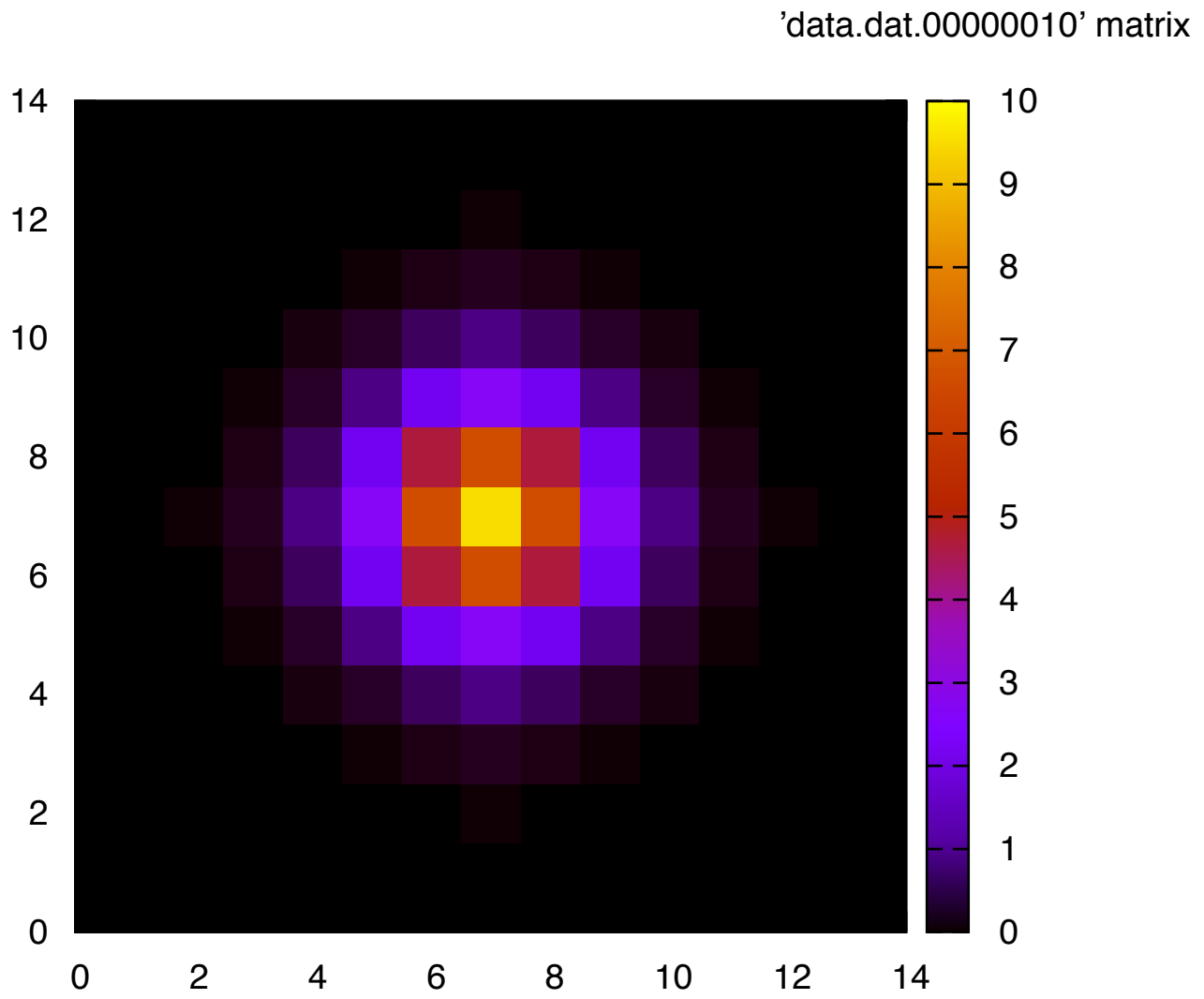
EX3: DIFFUSION - 2D4 NEIGHBOURHOOD, 15X15



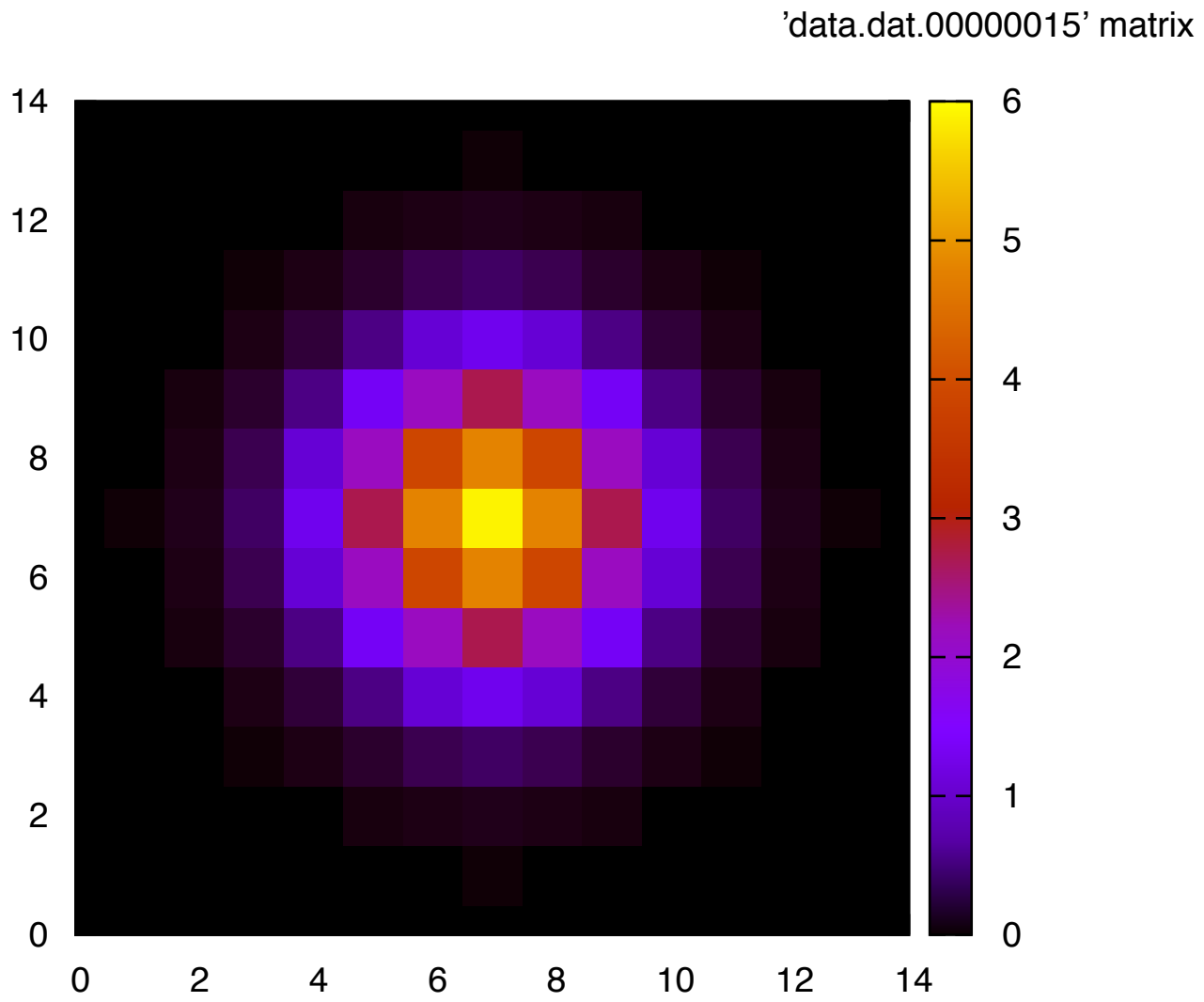
EX3: DIFFUSION - 2D4 NEIGHBOURHOOD, 15X15



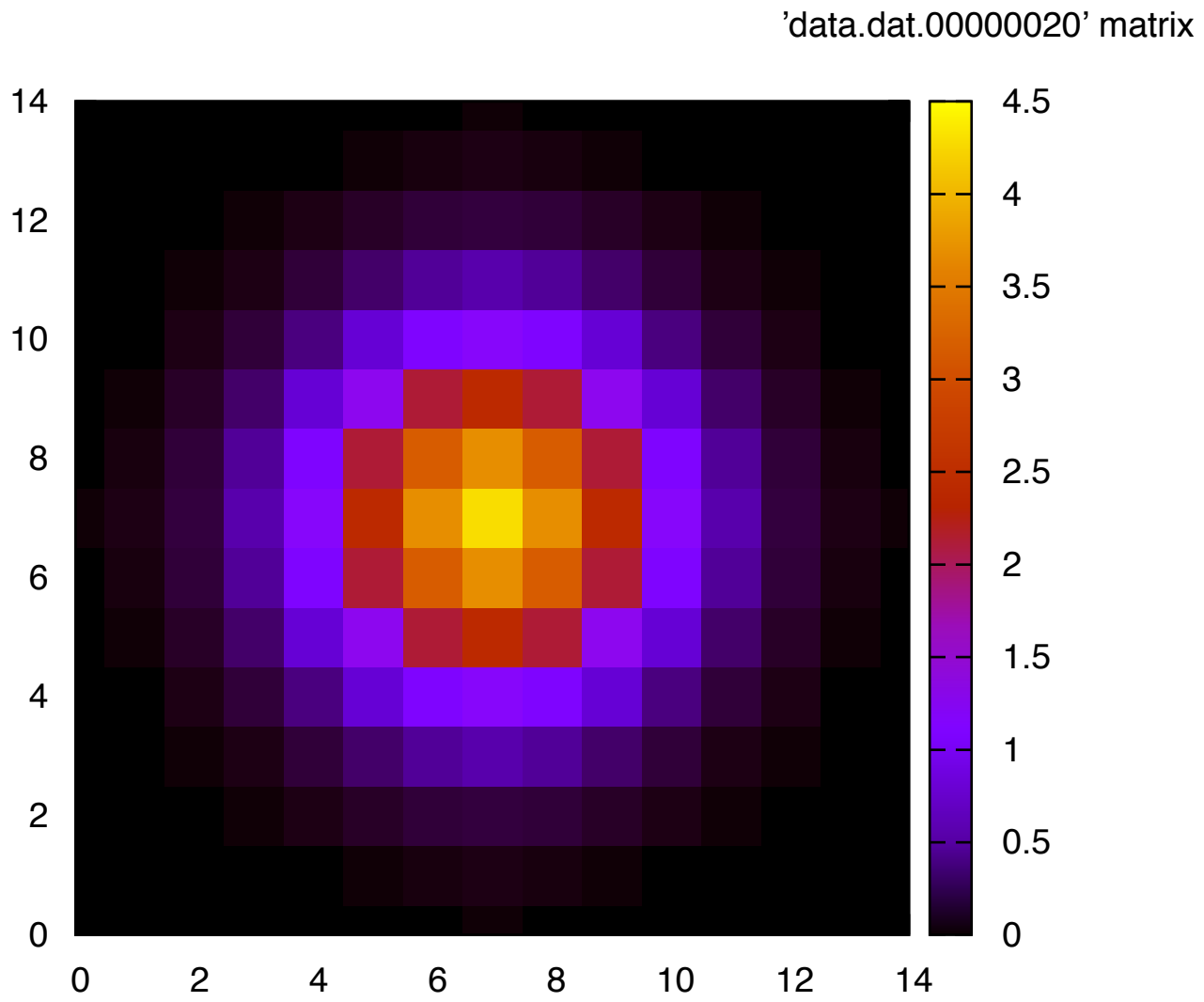
EX3: DIFFUSION - 2D4 NEIGHBOURHOOD, 15X15



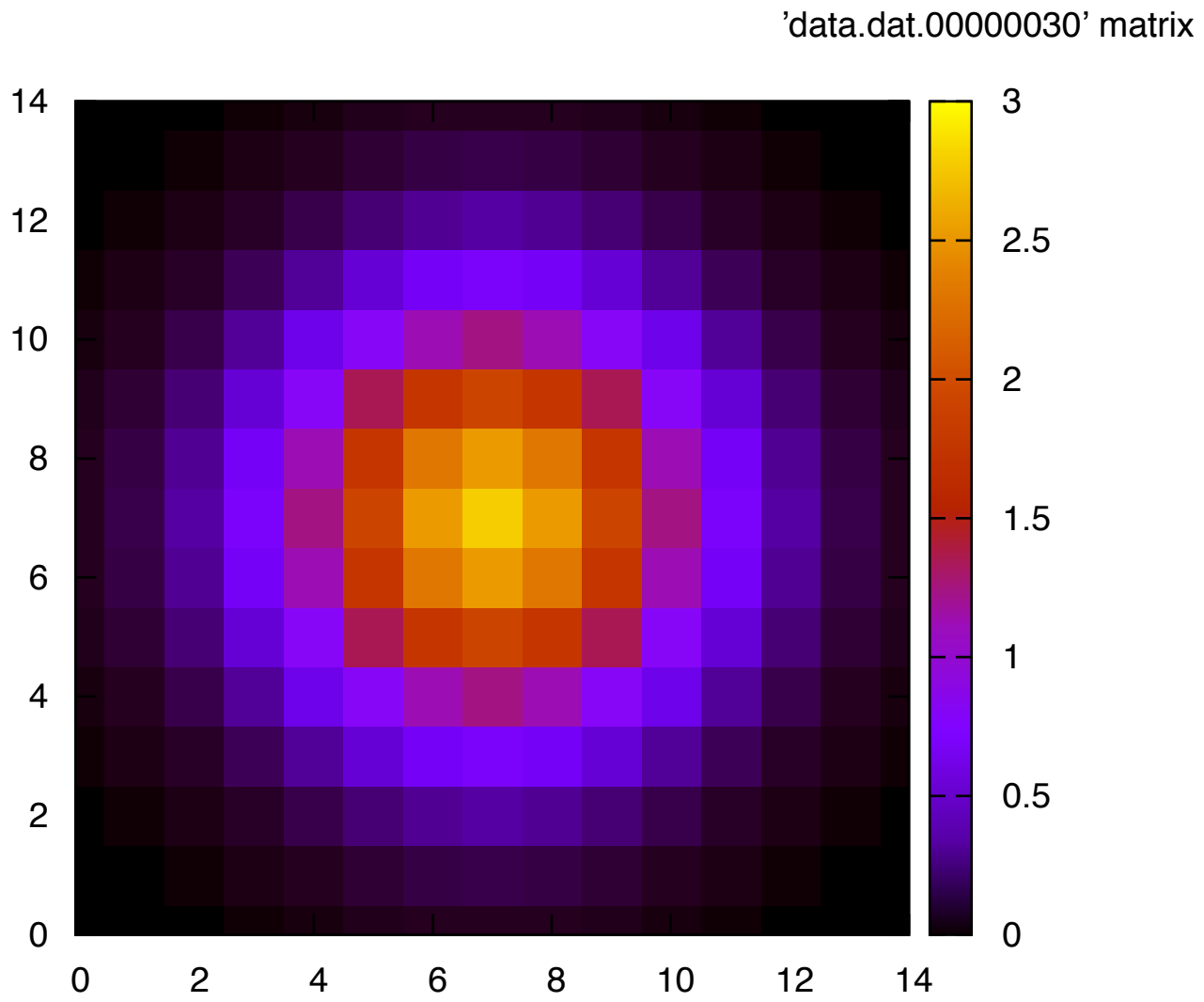
EX3: DIFFUSION - 2D4 NEIGHBOURHOOD, 15X15



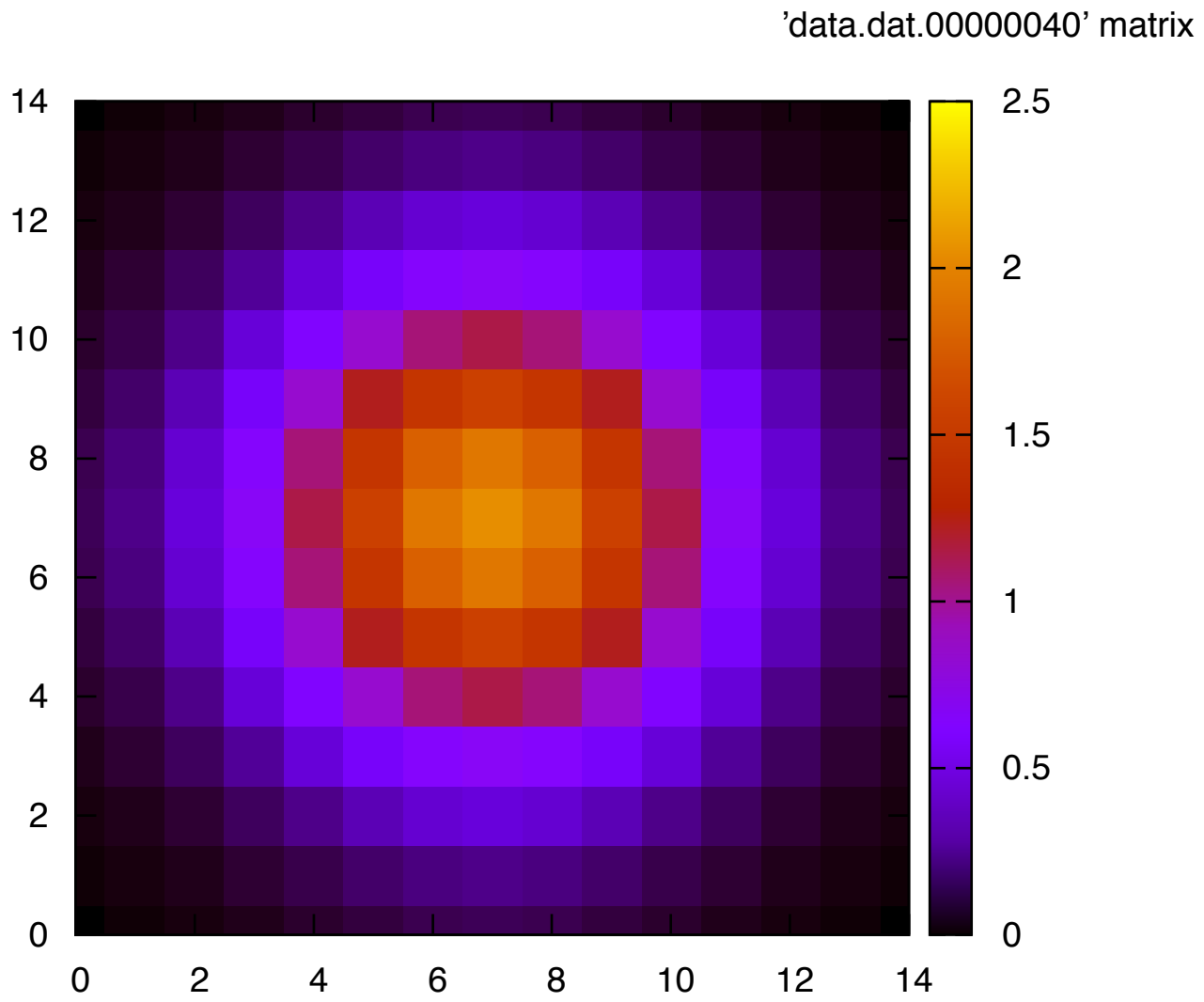
EX3: DIFFUSION - 2D4 NEIGHBOURHOOD, 15X15



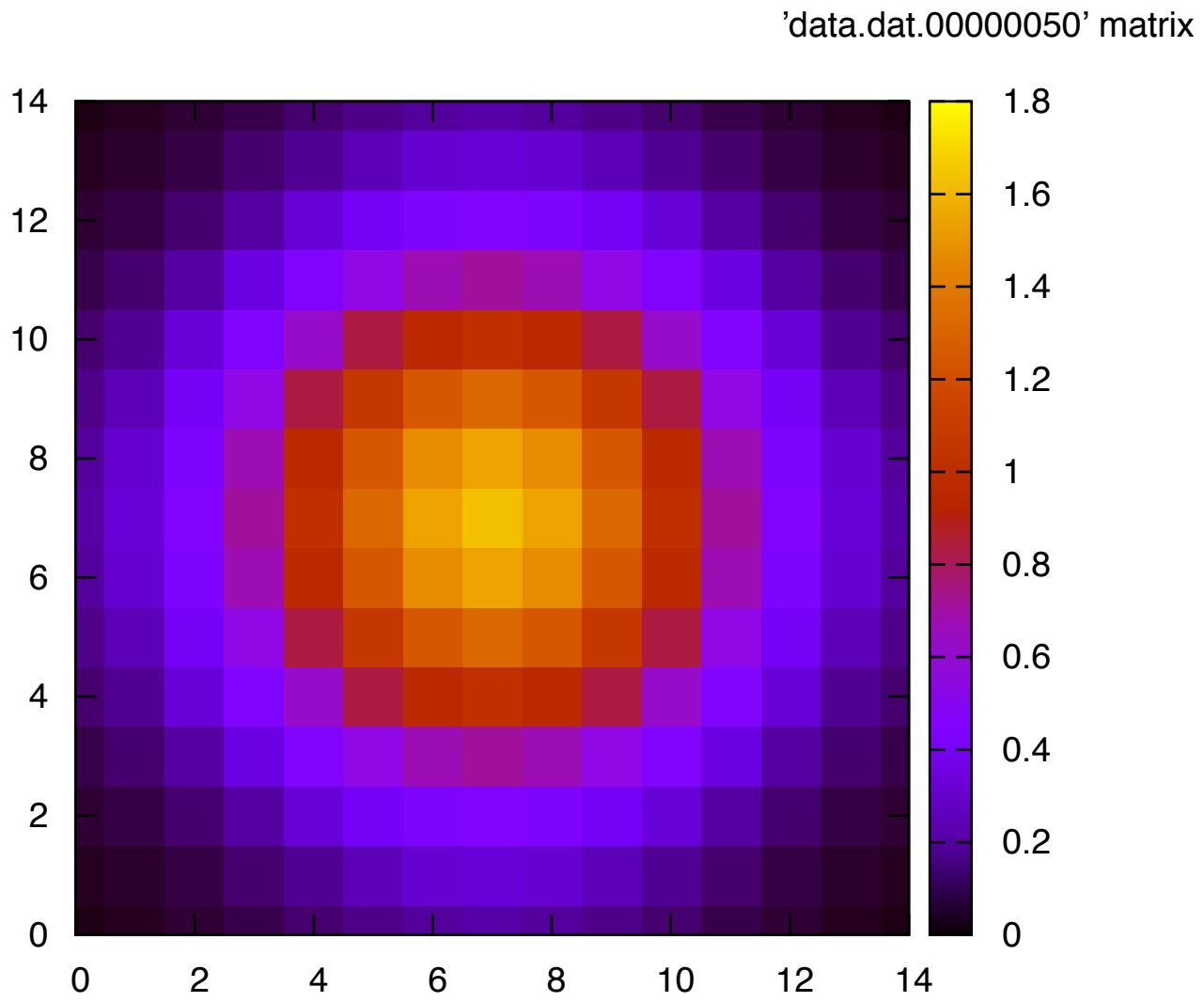
EX3: DIFFUSION - 2D4 NEIGHBOURHOOD, 15X15



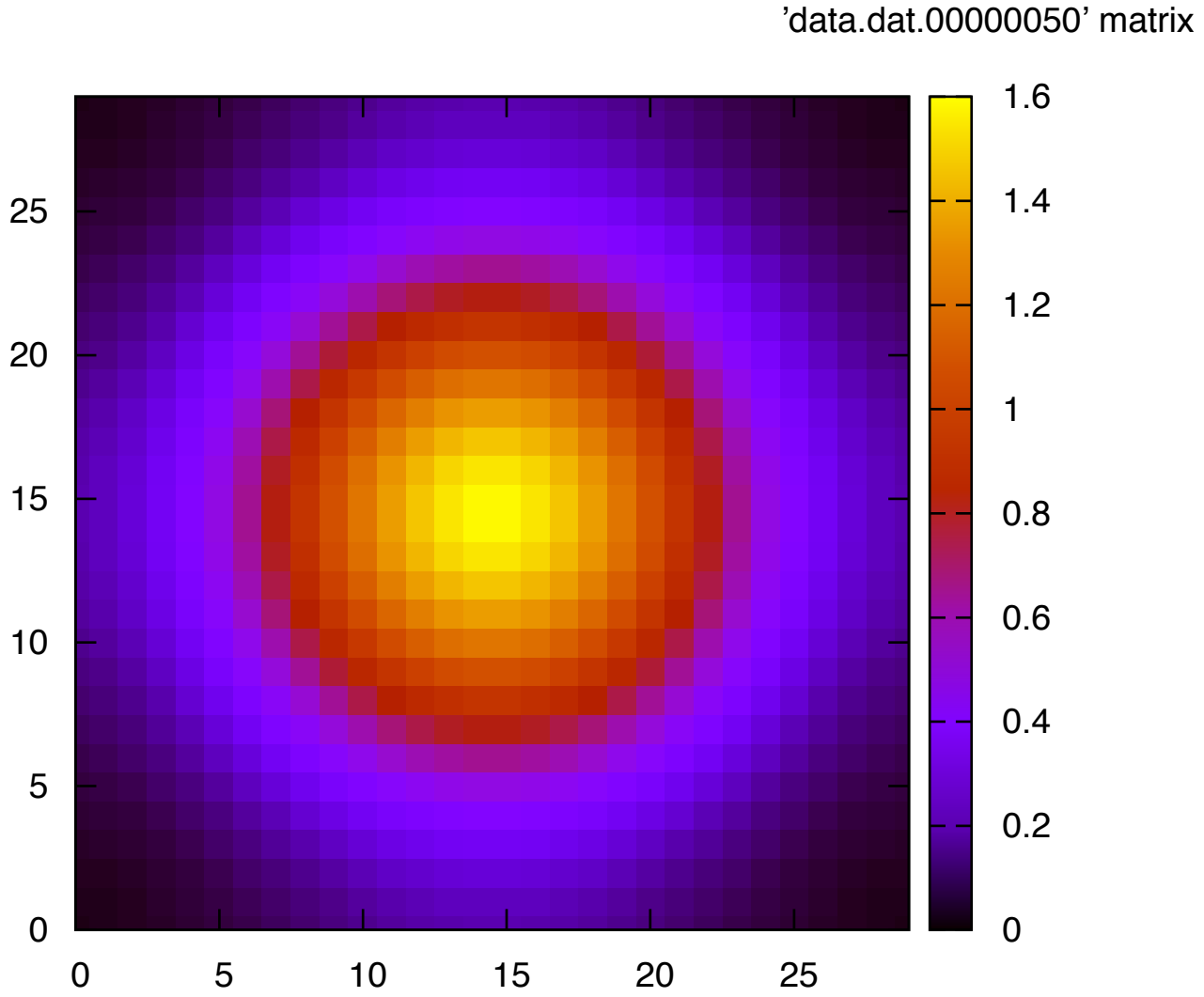
EX3: DIFFUSION - 2D4 NEIGHBOURHOOD, 15X15



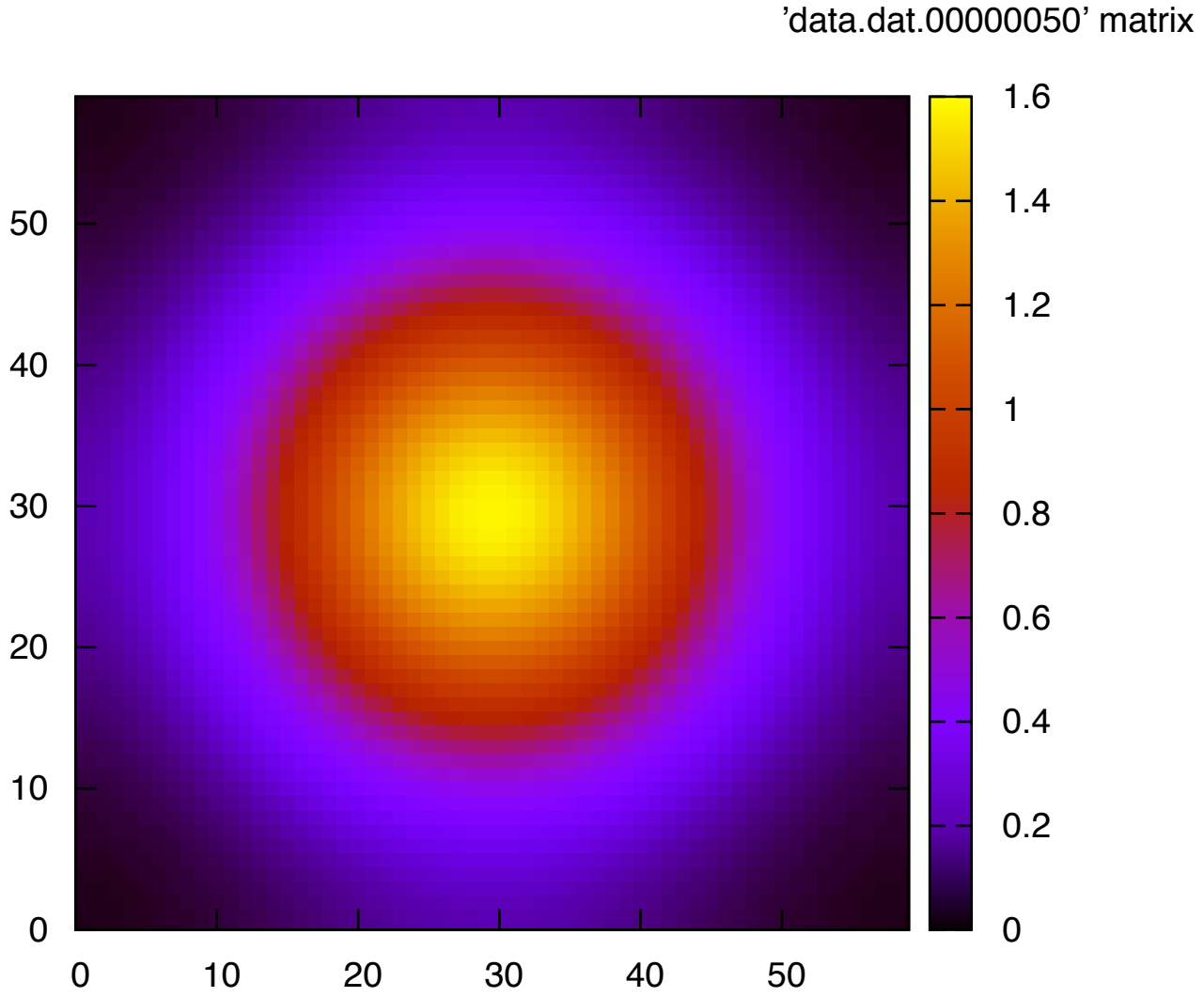
EX3: DIFFUSION - 2D4 NEIGHBOURHOOD, 15X15



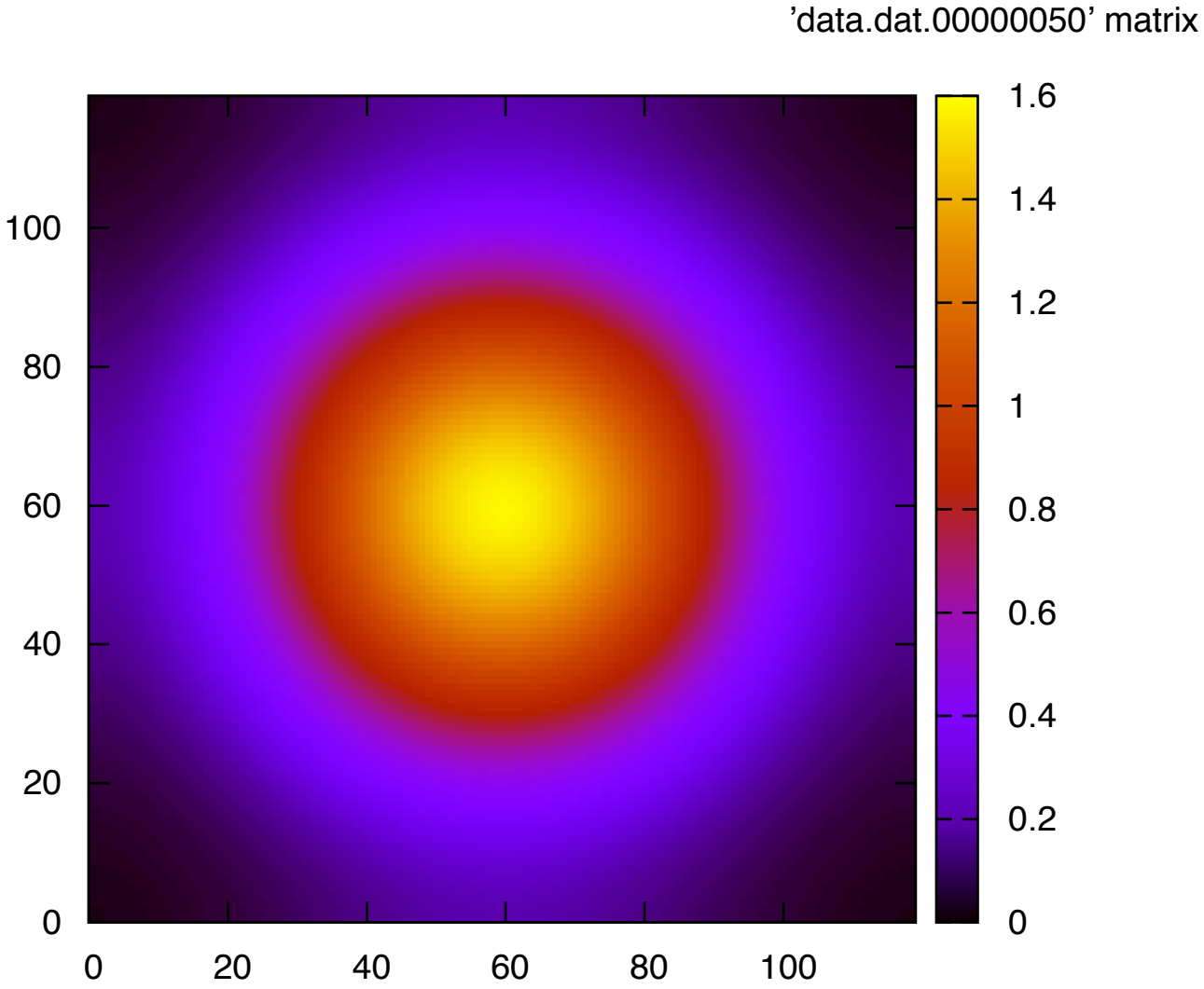
EX3: DIFFUSION - 2D4 NEIGHBOURHOOD, 30X30



EX3: DIFFUSION - 2D4 NEIGHBOURHOOD, 60X60



EX3: DIFFUSION - 2D4 NEIGHBOURHOOD, 120X120



EXAMPLE:

MULTISTRAIN CELL COLONIES

Microbiology (2003), **149**, 485–495

DOI 10.1099/mic.0.25807-0

Mutation rates: estimating phase variation rates when fitness differences are present and their impact on population structure

Nigel J. Saunders,[1]† E. Richard Moxon[1] and Mike B. Gravenor[2]

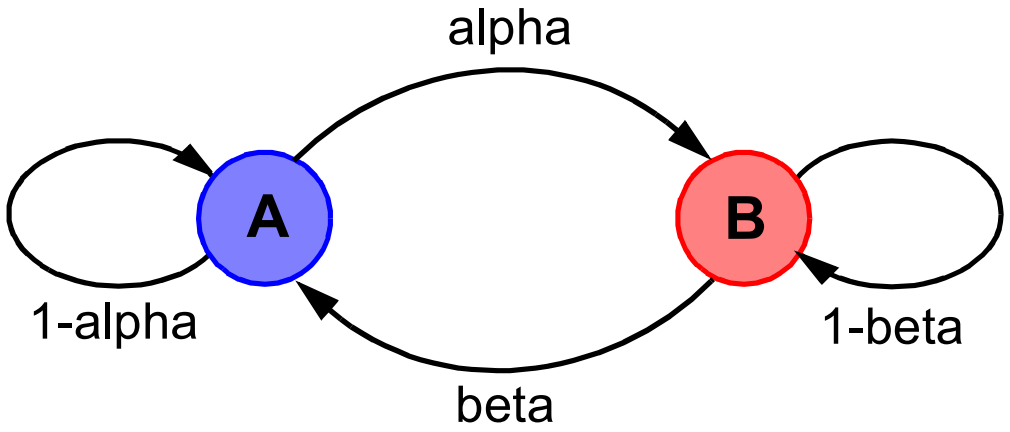
Correspondence
Nigel J. Saunders
saunders@molbiol.ox.ac.uk

[1]Molecular Infectious Diseases Group, Institute of Molecular Medicine, University of Oxford, Headington, Oxford OX3 9DS, UK

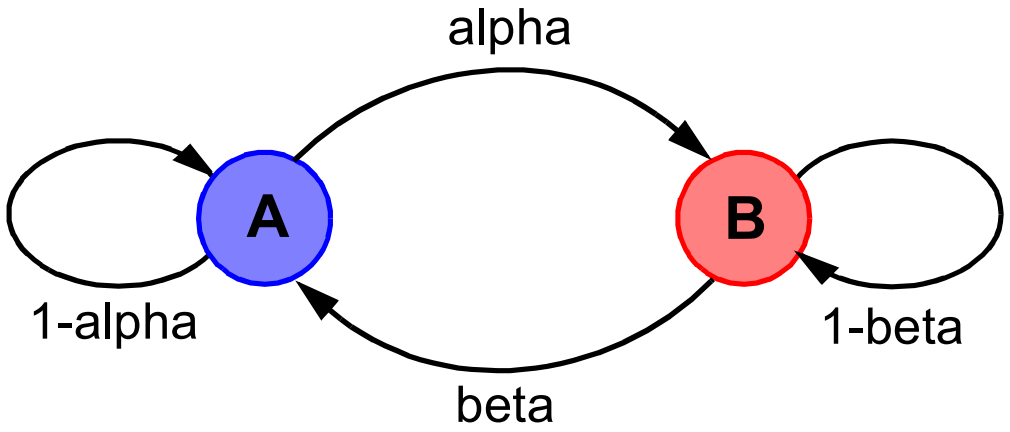
[2]Institute for Animal Health, Compton, Berkshire RG20 7NN, UK

Phase variation is a mechanism of ON–OFF switching that is widely utilized by bacterial pathogens. There is currently no standardization to how the rate of phase variation is determined experimentally.

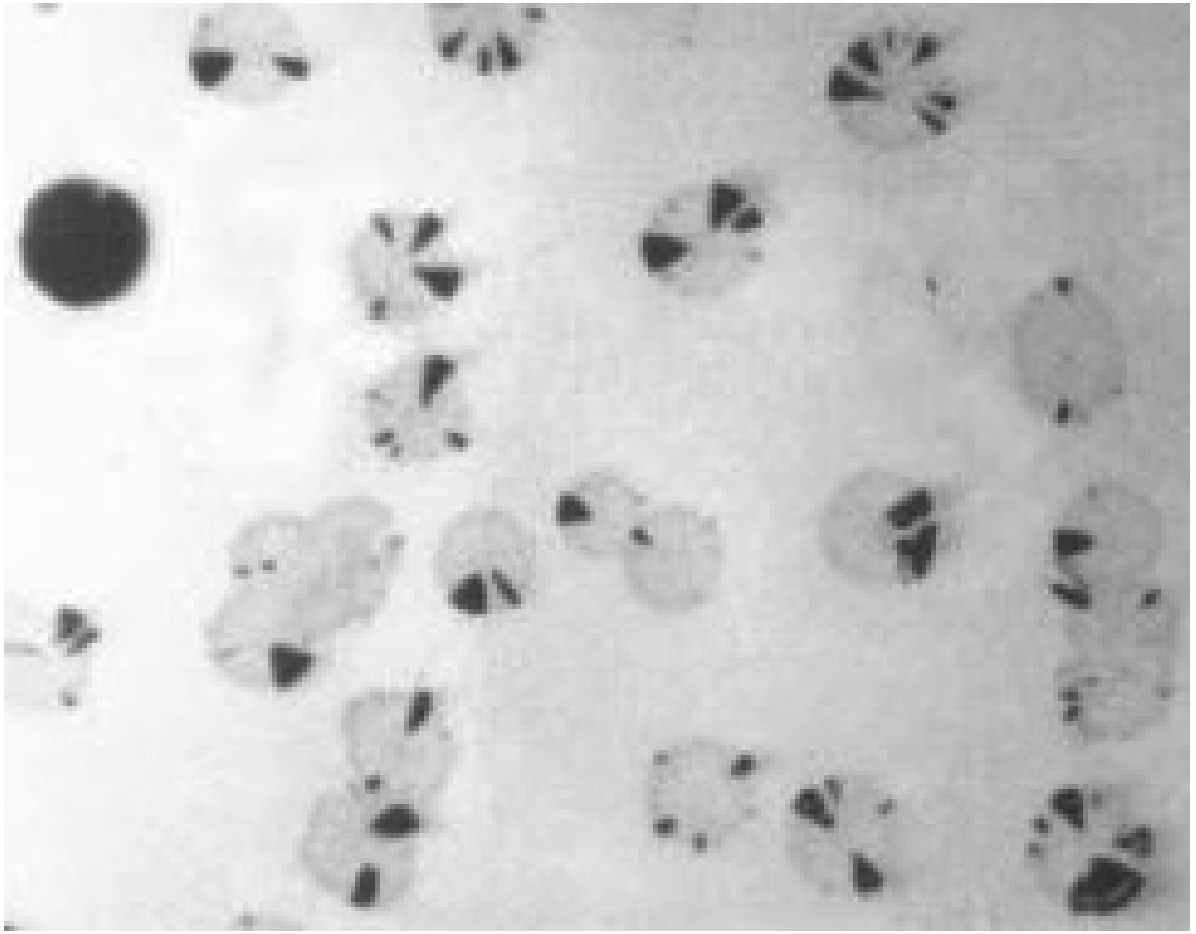
- **two cell types: phenotype A and B**
- **cell divide**
  - -> *cell division may involve mutation of the offspring*
  - -> *parent cell keeps its phenotype*



- **two cell types: phenotype A and B**
- **cell divide**
  - -> *cell division may involve mutation of the offspring*
  - -> *parent cell keeps its phenotype*
- **model parameters**
  - -> *alpha = beta - mutation rates*
  - -> *da, db - fitness of A, B*
  - -> *da/db - relative fitness*
- **output**
  - -> *total number of cells*
  - -> *proportion of A = A / (A + B)*
  - -> *proportion of B = B / (A + B)*

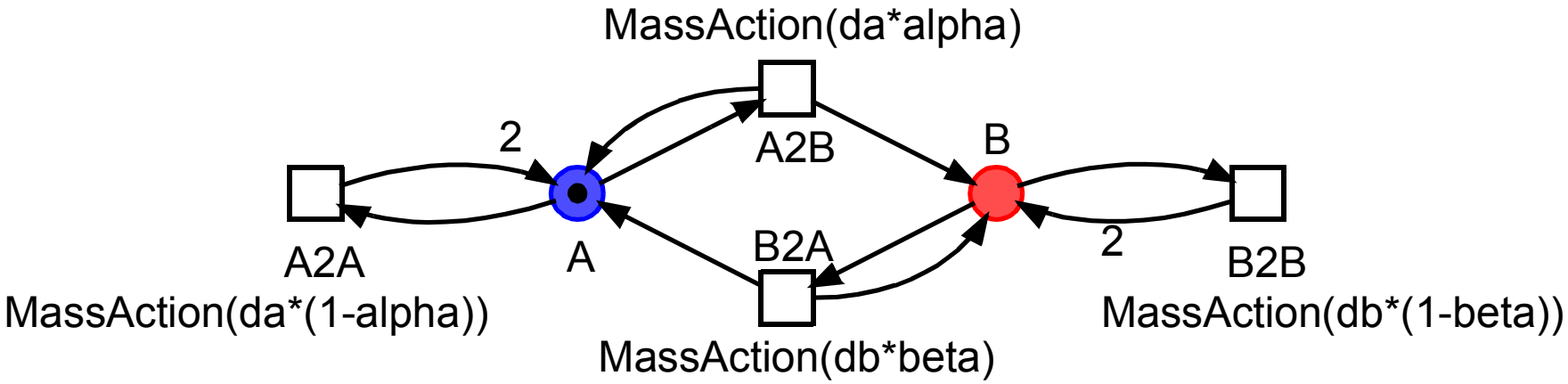


EX4: CELL COLONIES, WETLAB OBSERVATIONS

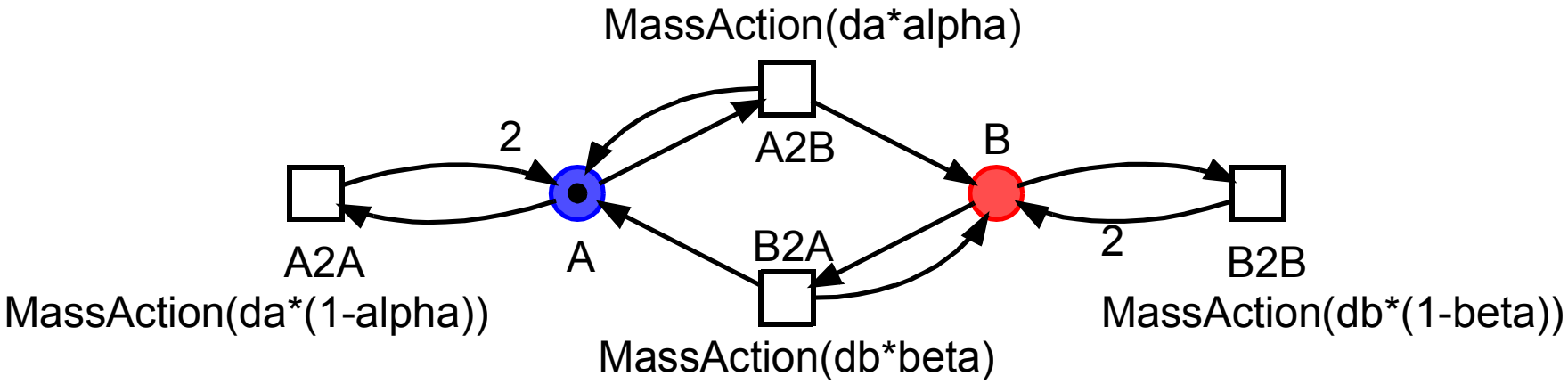


(courtesy of N Saunders)

EX4: CELL COLONIES, PETRI NET 1

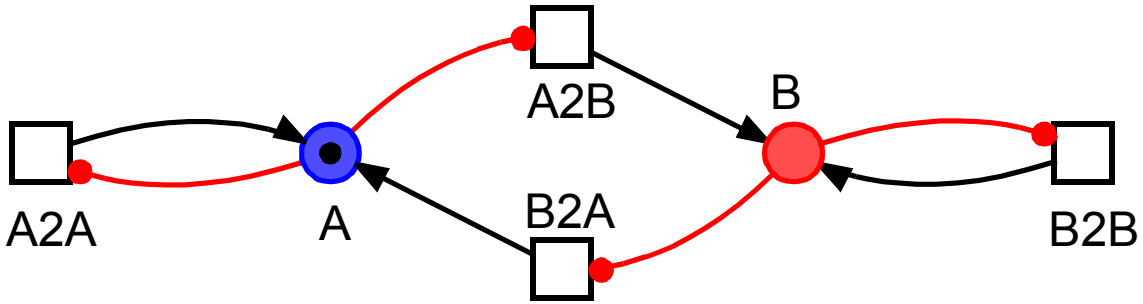


EX4: CELL COLONIES, PETRI NET 1

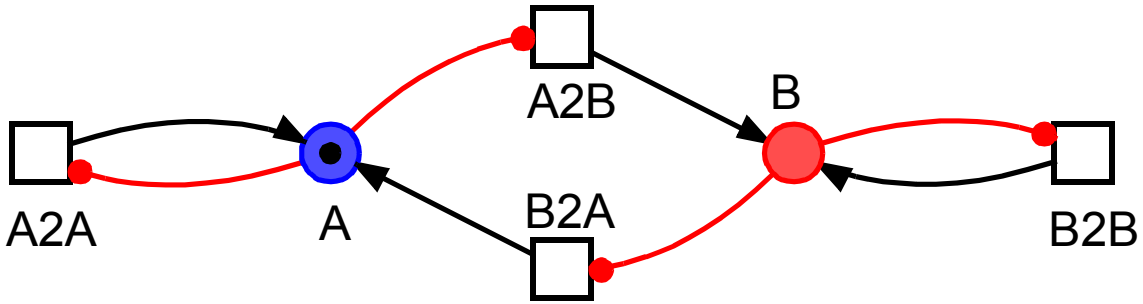


$$dA/dt = da * (1 - alpha) * A + db * beta * B$$
$$dB/dt = db * (1 - beta) * B \quad + da * alpha * A$$

EX4: CELL COLONIES, PETRI NET 2



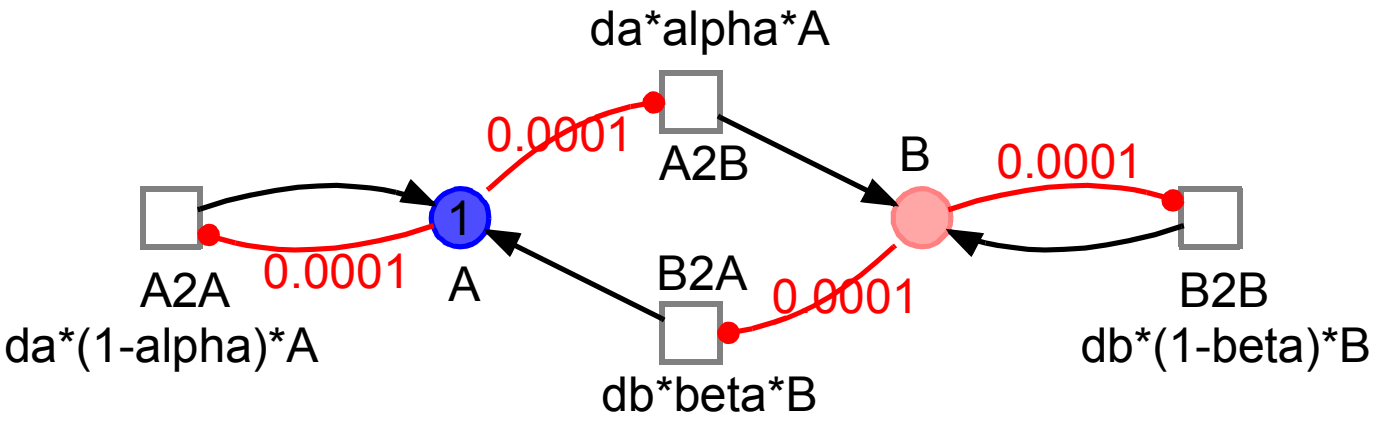
EX4: CELL COLONIES, PETRI NET 2



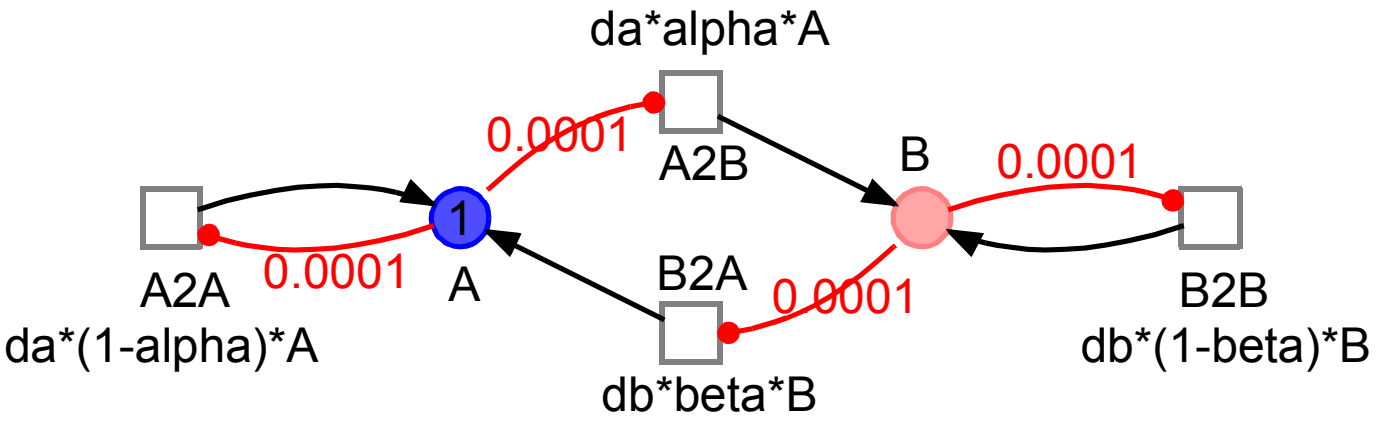
$$dA/dt = da * (1 - alpha) * A * read(1,A) + db * beta * B * read(1,B)$$

$$dB/dt = db * (1 - beta) * B * read(1,B) + da * alpha * A * read(1,A)$$

EX4: CELL COLONIES, PETRI NET 3



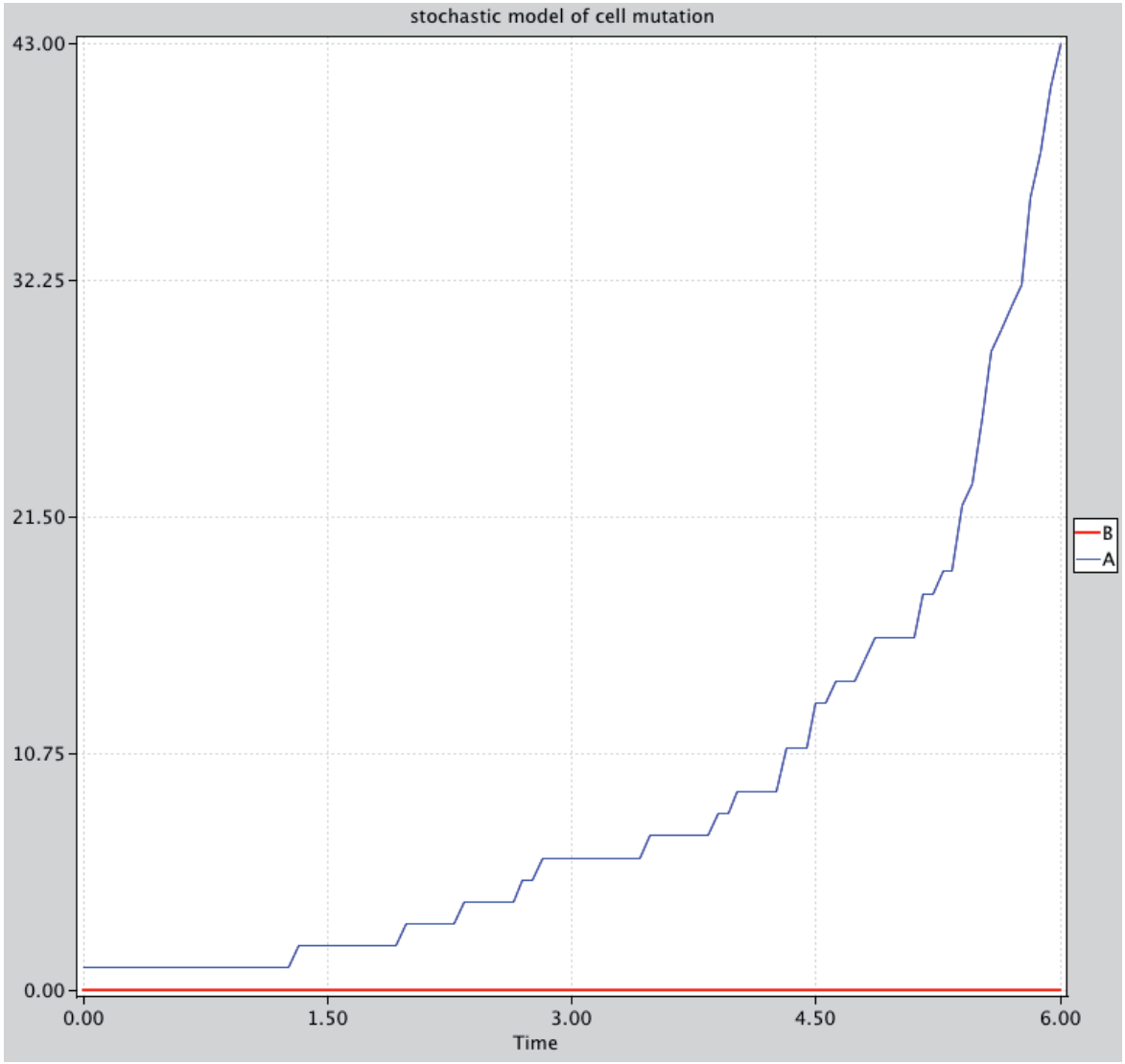
EX4: CELL COLONIES, PETRI NET 3



$$dA/dt = da * (1 - alpha) * A * read(0.0001,A) + db * beta * B * read(0.0001,B)$$

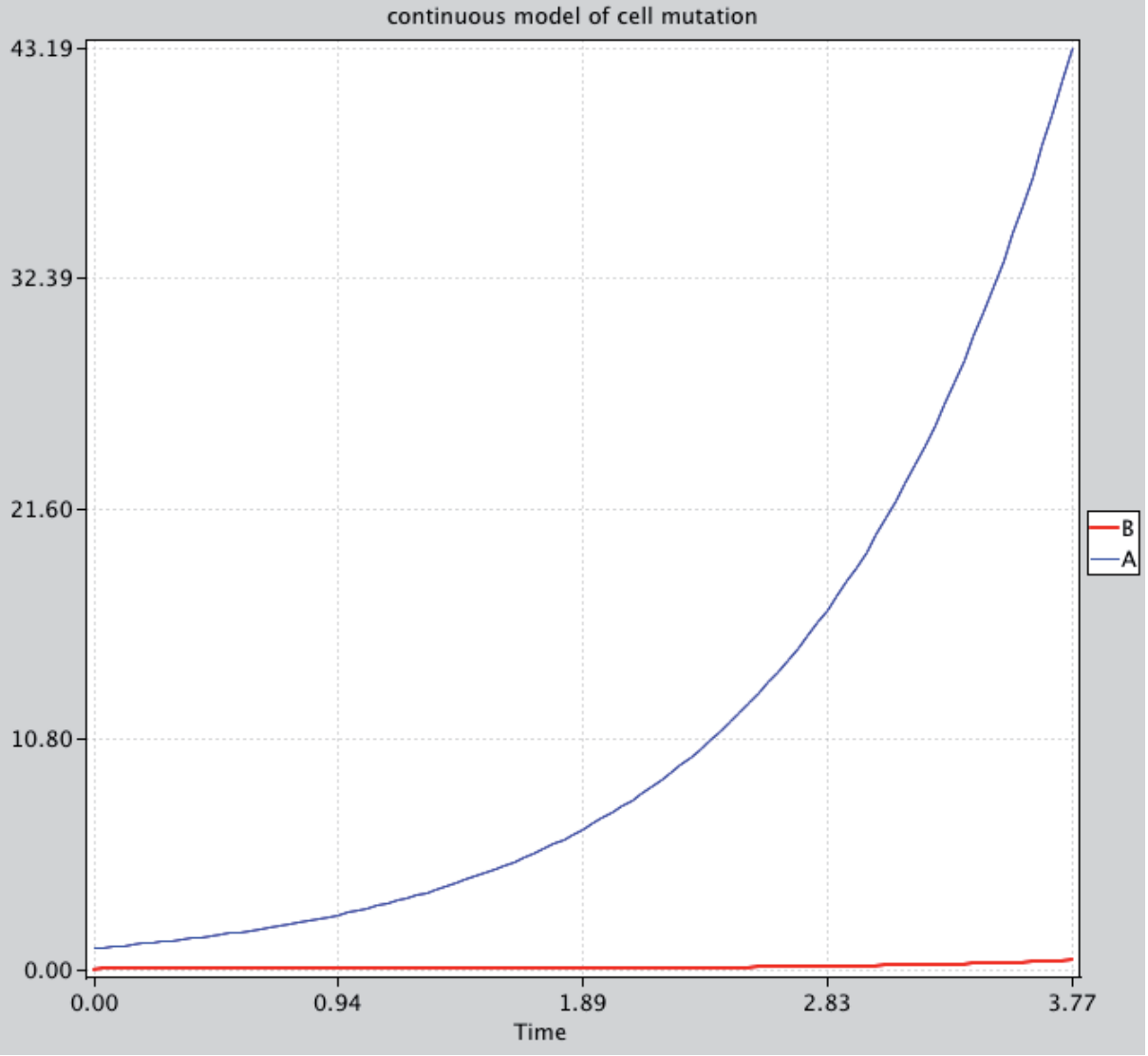
$$dB/dt = db * (1 - beta) * B * read(0.0001,B) + da * alpha * A * read(0.0001,A)$$

EX4: CELL COLONIES, STOCHASTIC PLOT



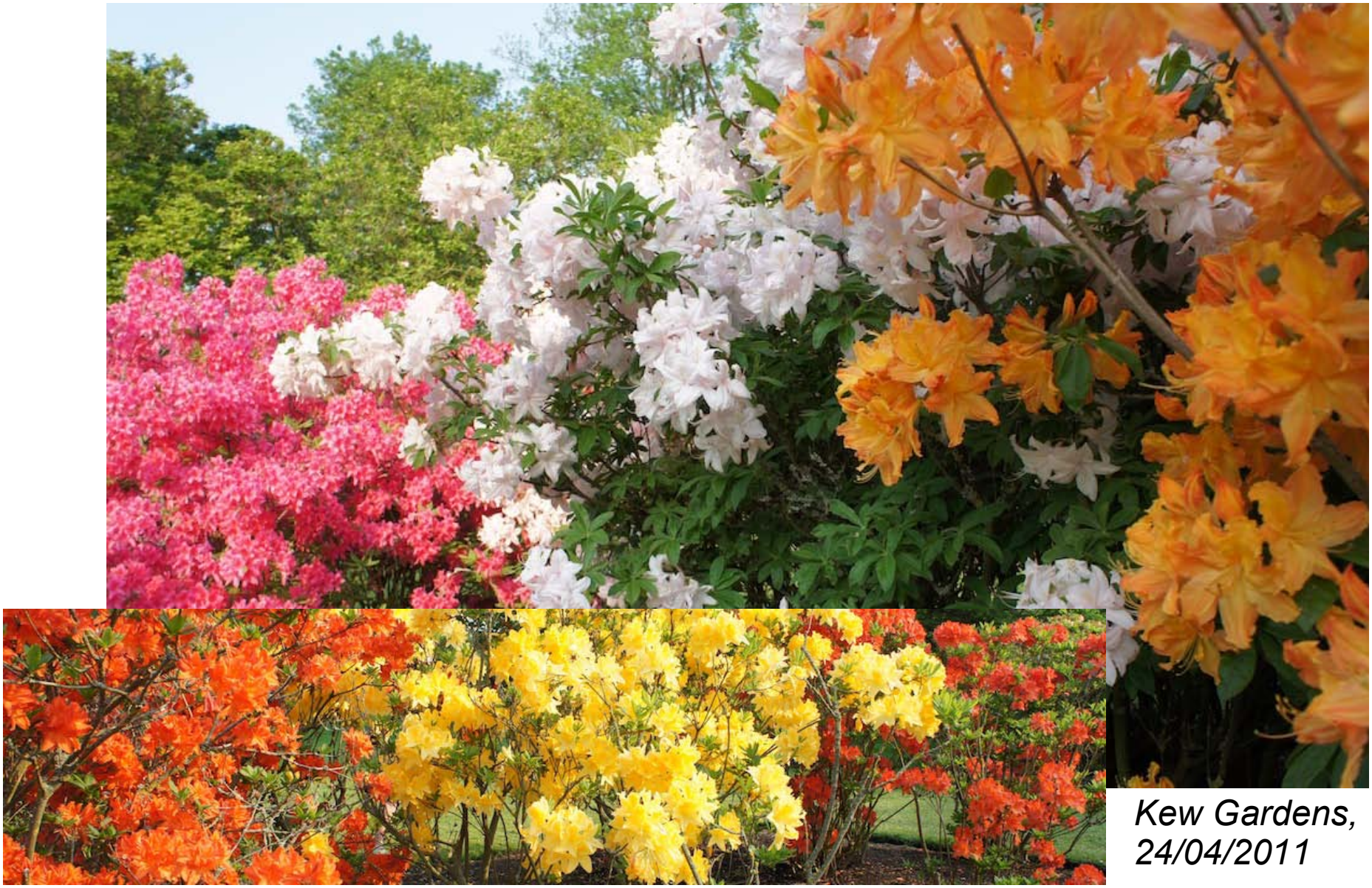
monika.heiner@brunel.ac.uk

EX4: CELL COLONIES, CONTINUOUS PLOT



monika.heiner@brunel.ac.uk

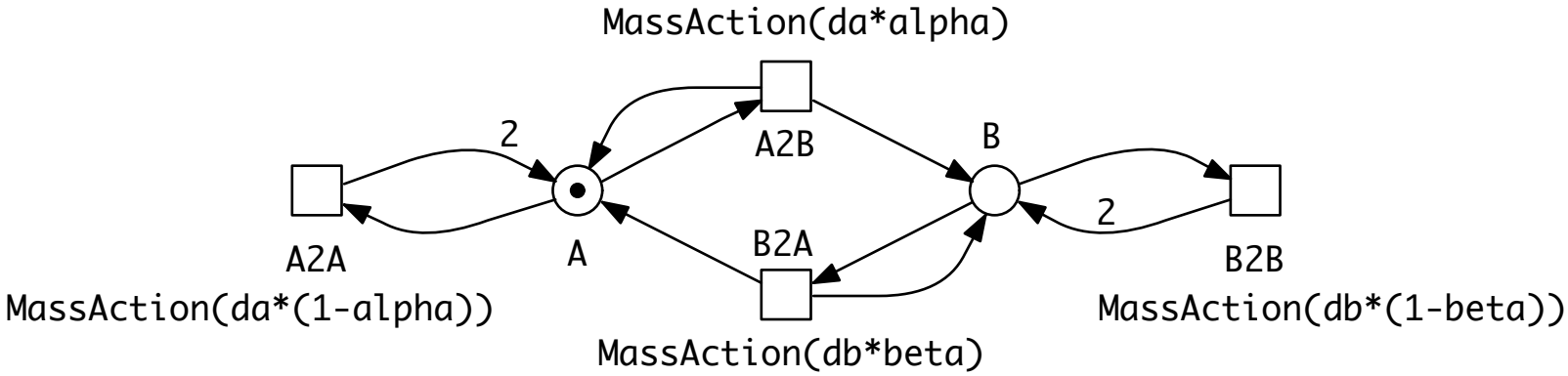
. . . AND THEN THERE WAS COLOUR

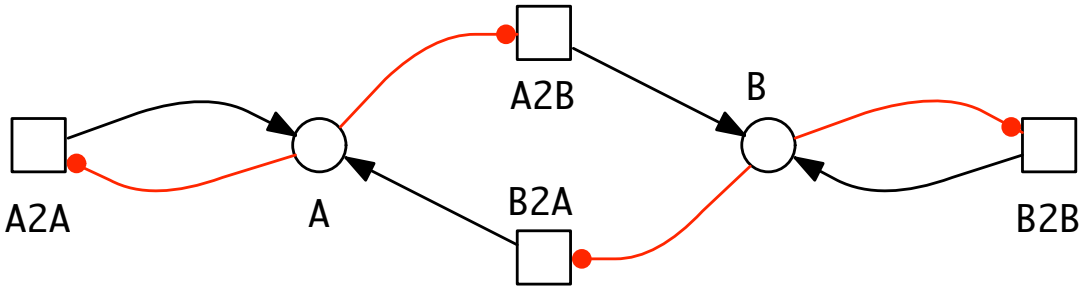


Kew Gardens, 24/04/2011

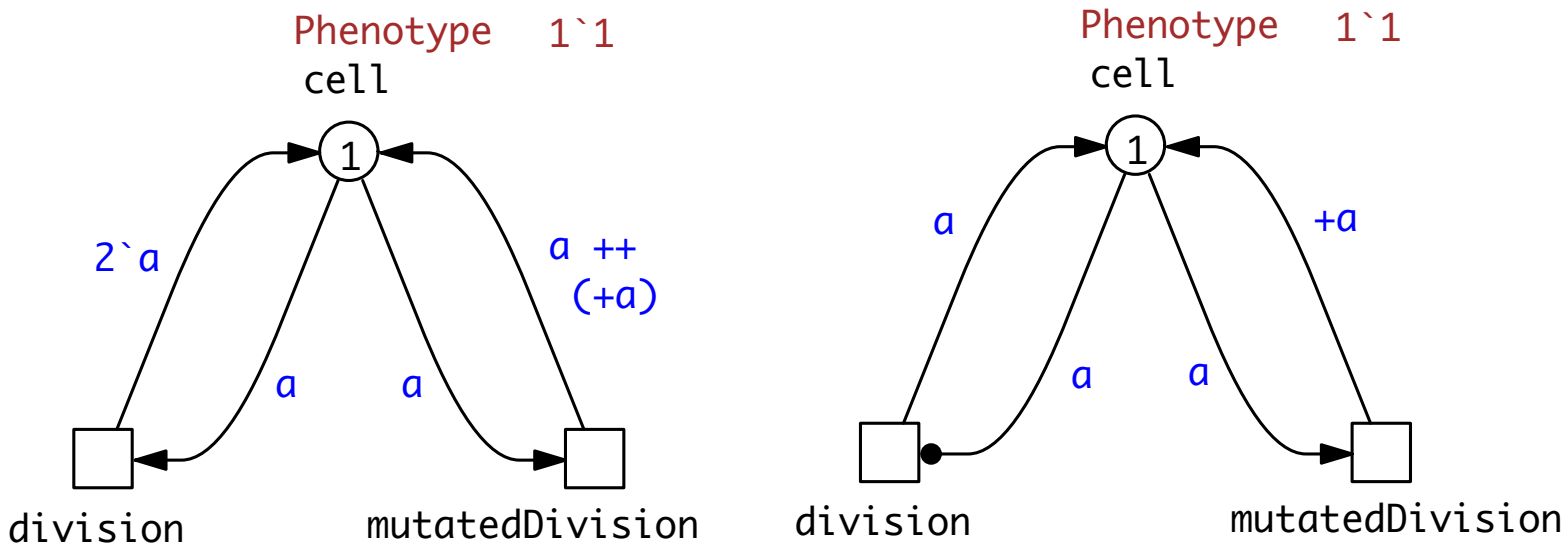
monika.heiner@brunel.ac.uk

EX4: CELL COLONIES, PLAIN PETRI NETS





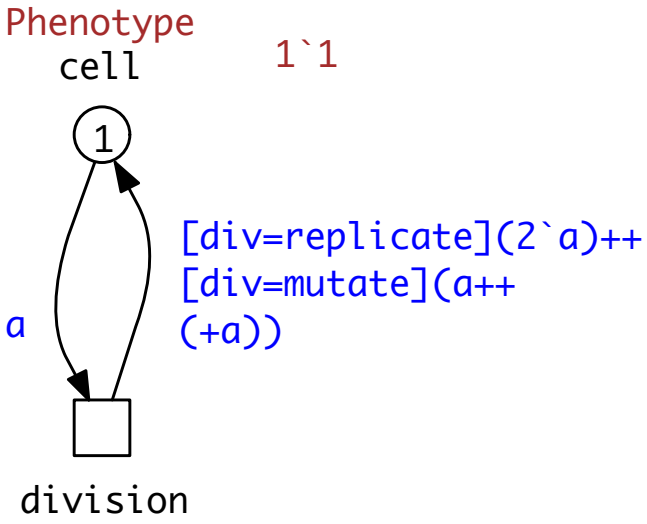
EX4: CELL COLONIES, FOLDING STEP 1

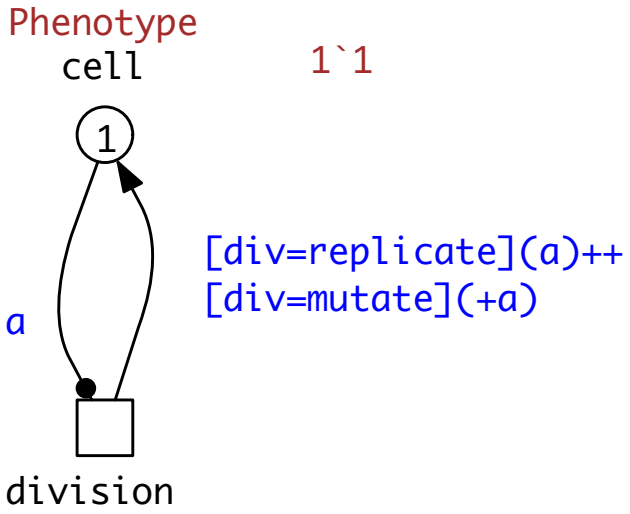



```
a=1 : da*(1-alpha)*cell
a=2 : db*(1-beta)*cell
```

```
a=1 : da*alpha*cell
a=2 : db*beta*cell
```

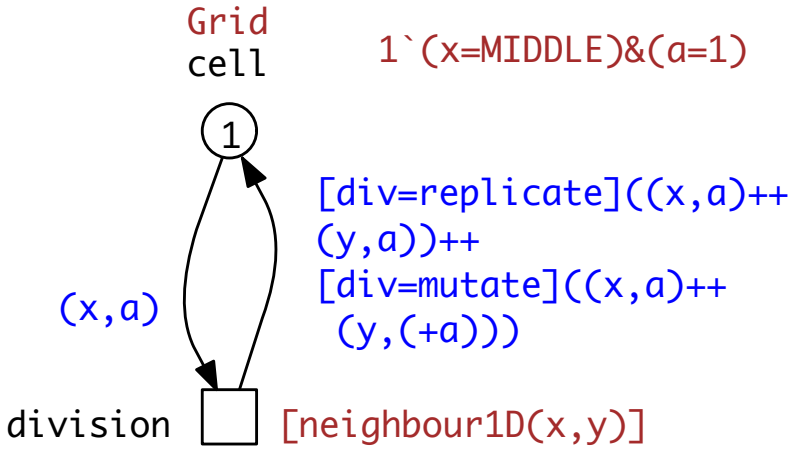
EX4: CELL COLONIES, FOLDING STEP 2

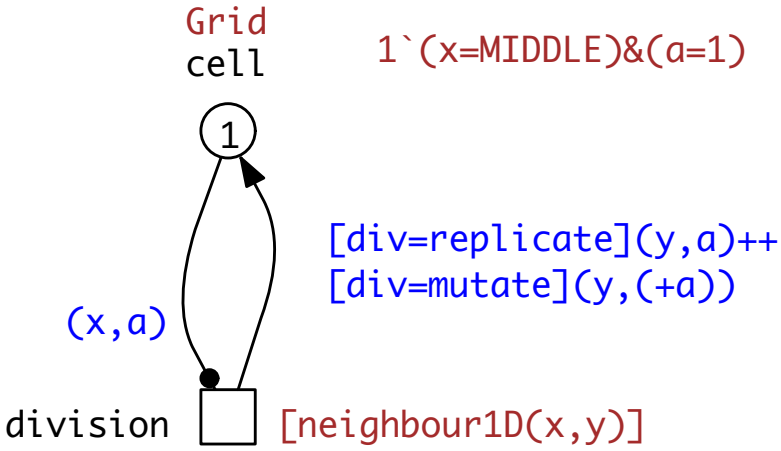





```
(a=1) & (div=replicate) : cell*da*(1-alpha)
(a=1) & (div=mutate) : cell*(da*alpha)
(a=2) & (div=replicate) : cell*(db*(1-beta))
(a=2) & (div=mutate) : cell*(db*beta)
```

EX4: CELL COLONIES, ADDING SPACE

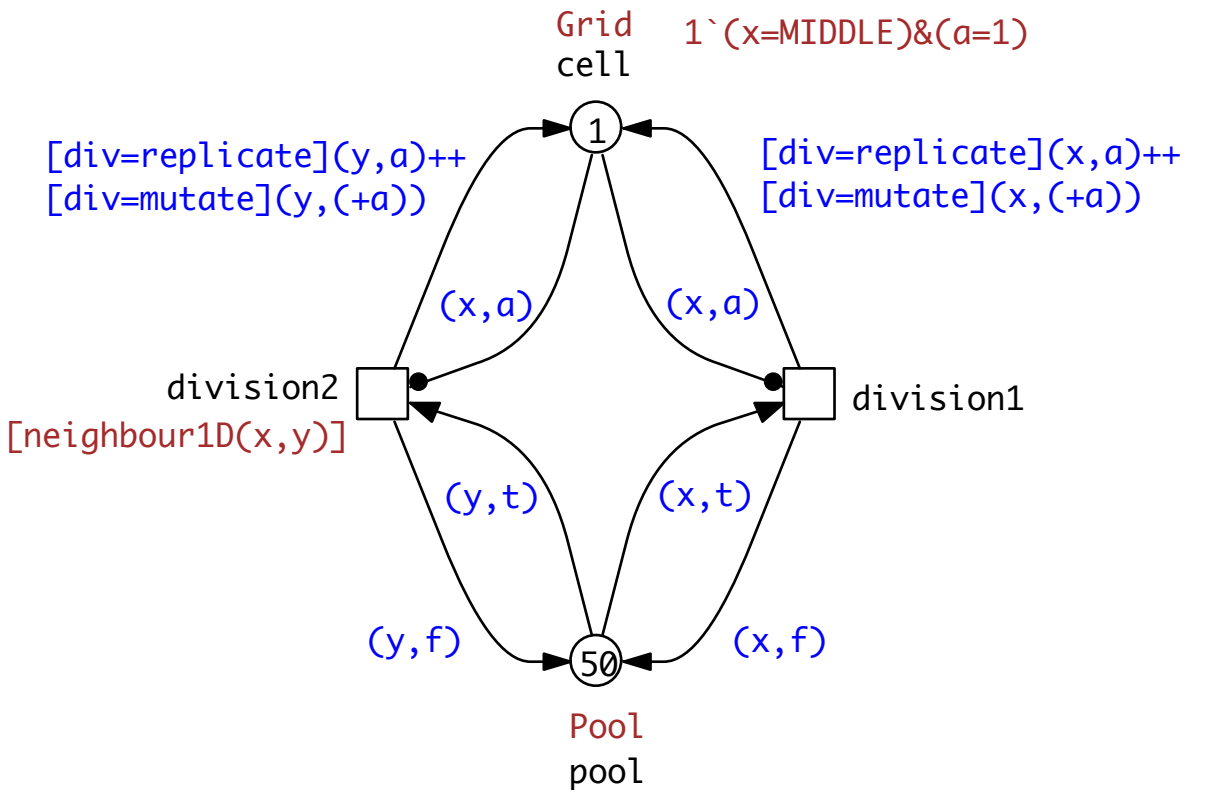





```
(a=1) & (div=replicate) : cell*da*(1-alpha)
(a=1) & (div=mutate) : cell*(da*alpha)
(a=2) & (div=replicate) : cell*(db*(1-beta))
(a=2) & (div=mutate) : cell*(db*beta)
```

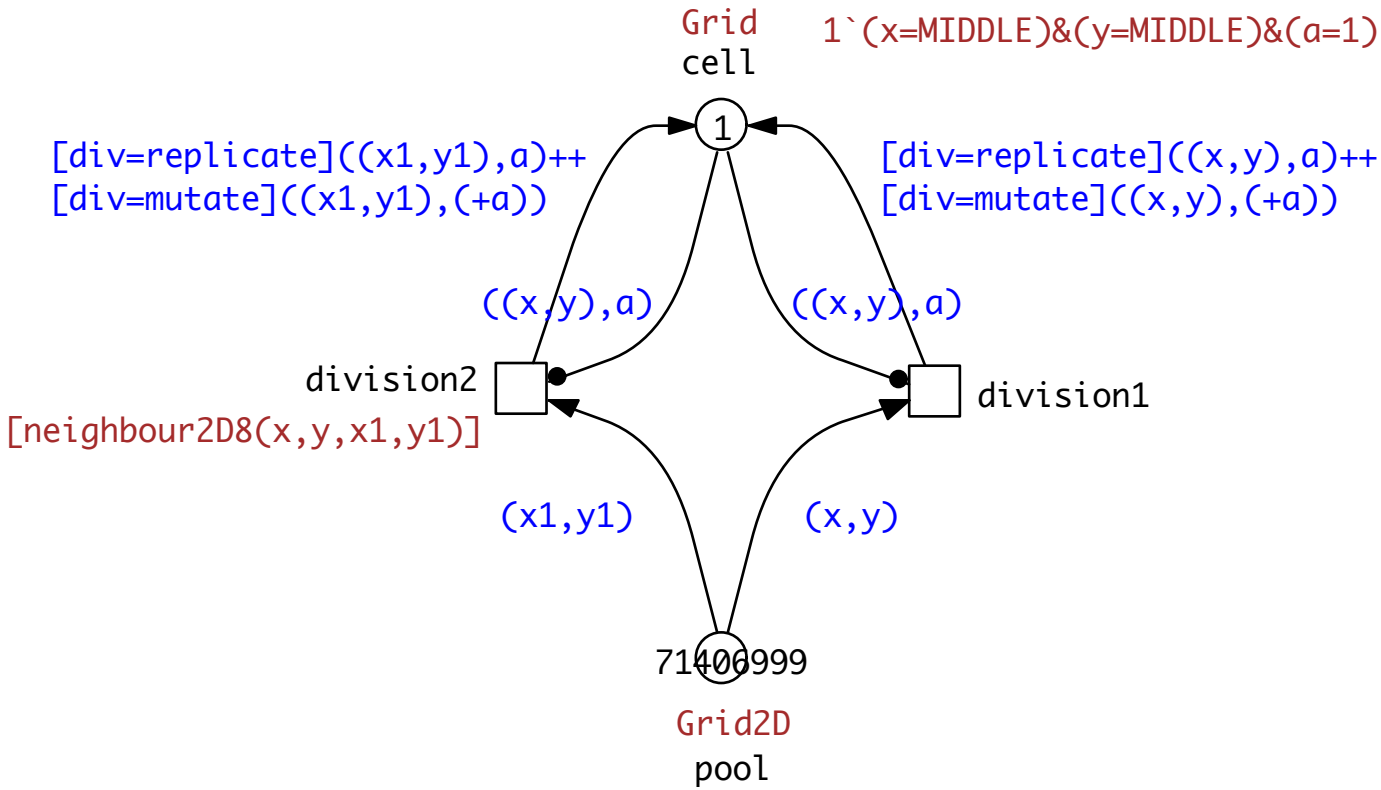
EX4: CELL COLONIES, CONTROLLING BIOFILM THICKNESS

EX4: CELL COLONIES, CONTROLLING SPREADING 1




```
POOLSIZE`(1<=x)&(x<=D1)&(x<>MIDDLE)&(ticket=t)++
1`(x=MIDDLE)&(ticket=f)++
9`(x=MIDDLE)&(ticket=t)
```

EX4: CELL COLONIES, CONTROLLING SPREADING 2




```
POOLSIZE`(1<=x&x<=D1) & (1<=y&y<=D2) & (x<>MIDDLE|y<>MIDDLE)++
POOLSIZE_1`(x=MIDDLE & y=MIDDLE)
```

EX4: CELL COLONIES, SOME DETAILS

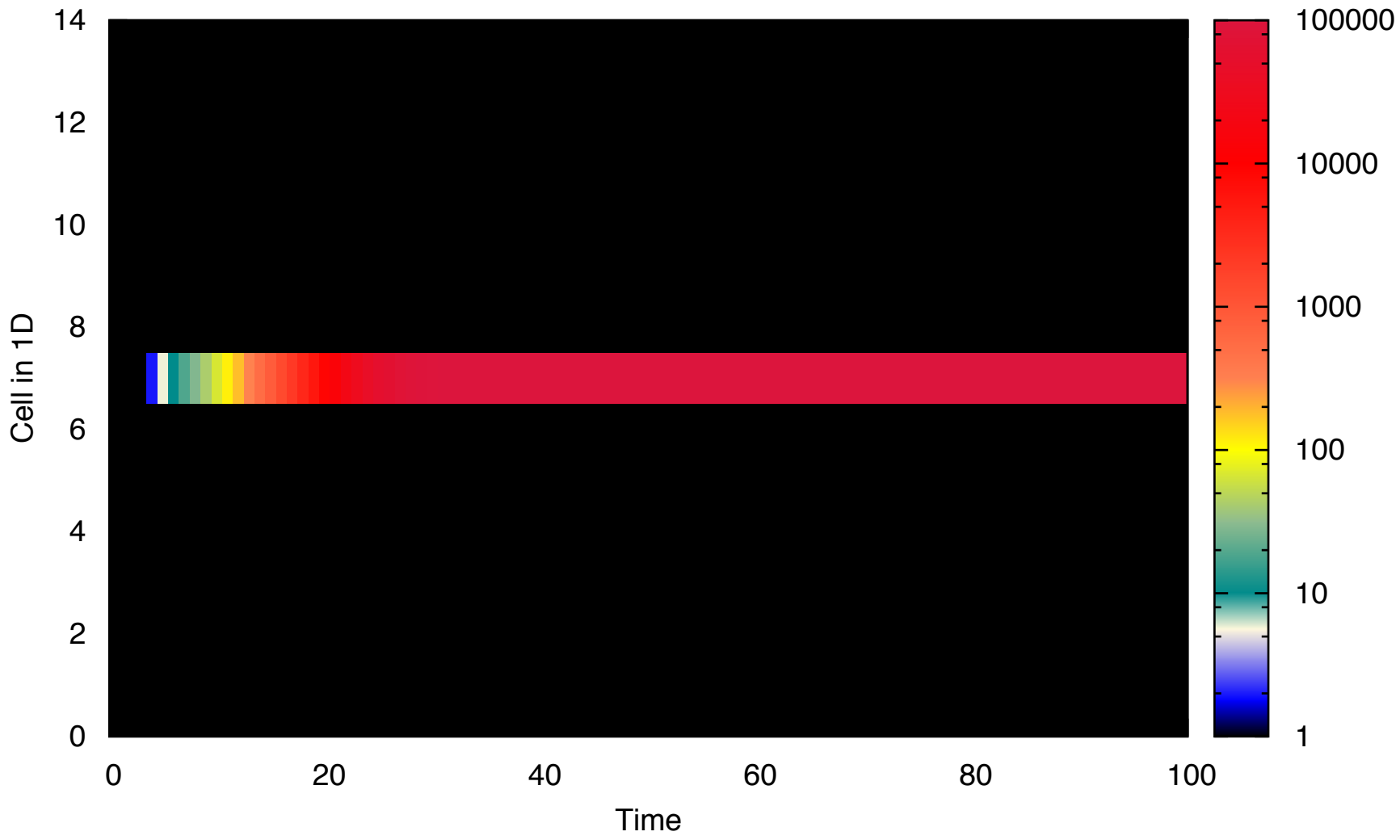
- **model assumptions**
  - -> *"If phase variation occurs, the progeny consists of one A and one B" (Saunders 2003)*
  - -> *It is always the mutant who goes to a neighbouring position, if any.*
  - -> *constant biofilm thickness (so far)*

- **colony size**
  - -> *25 generations: 33.5 E+06*
  - -> *26 generations: 67 E+06*

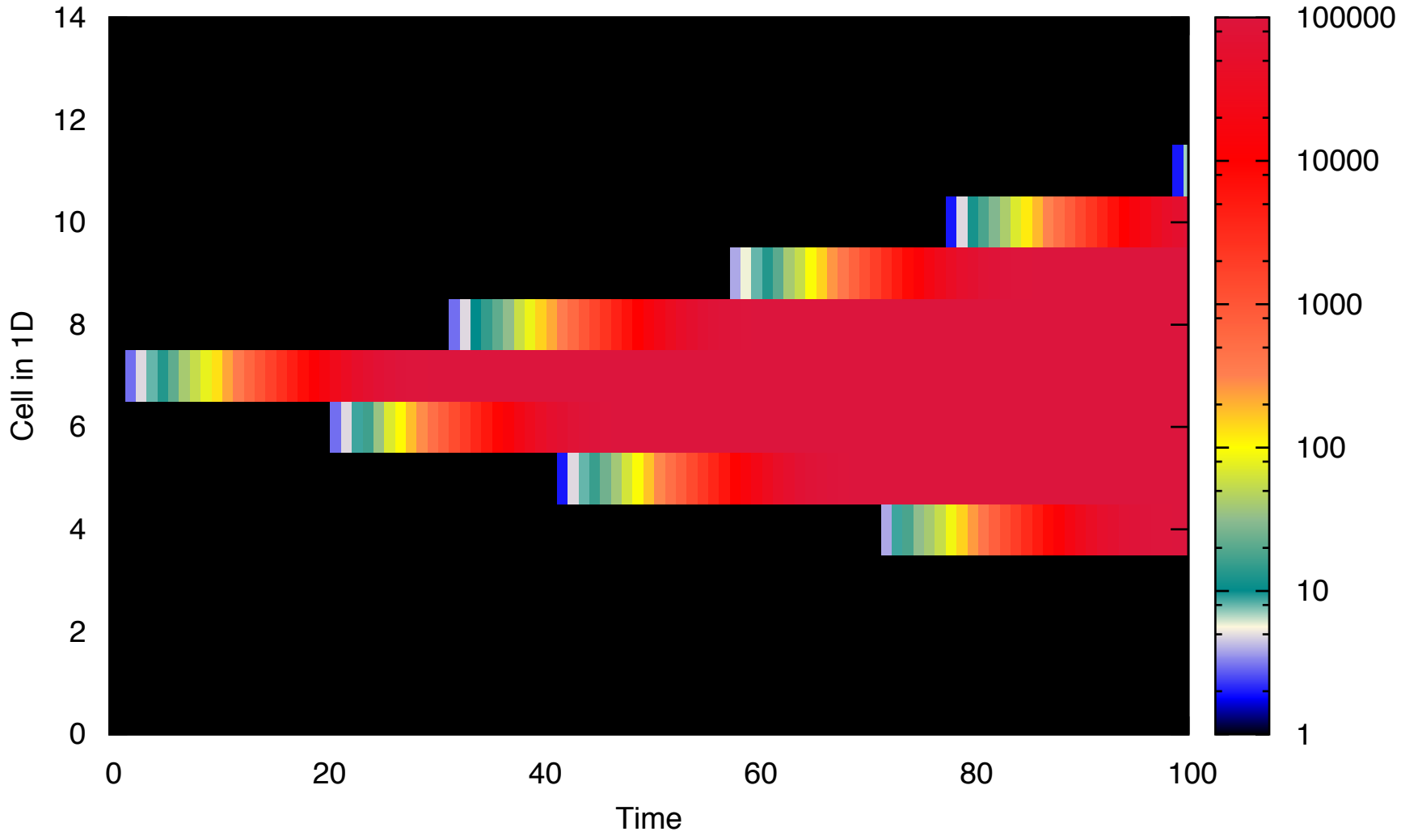
- **grid size**
  - -> *61x61 grid: 11,163 P / 131,044 T; unfolding: 152 sec;*
  - -> *101x101 grid: 30,603 P / 362,404 T; unfolding: 9 min;*

. . . SOME EXPERIMENTS

Ex4: 1D15 - Varying Mobility, Gamma = 100

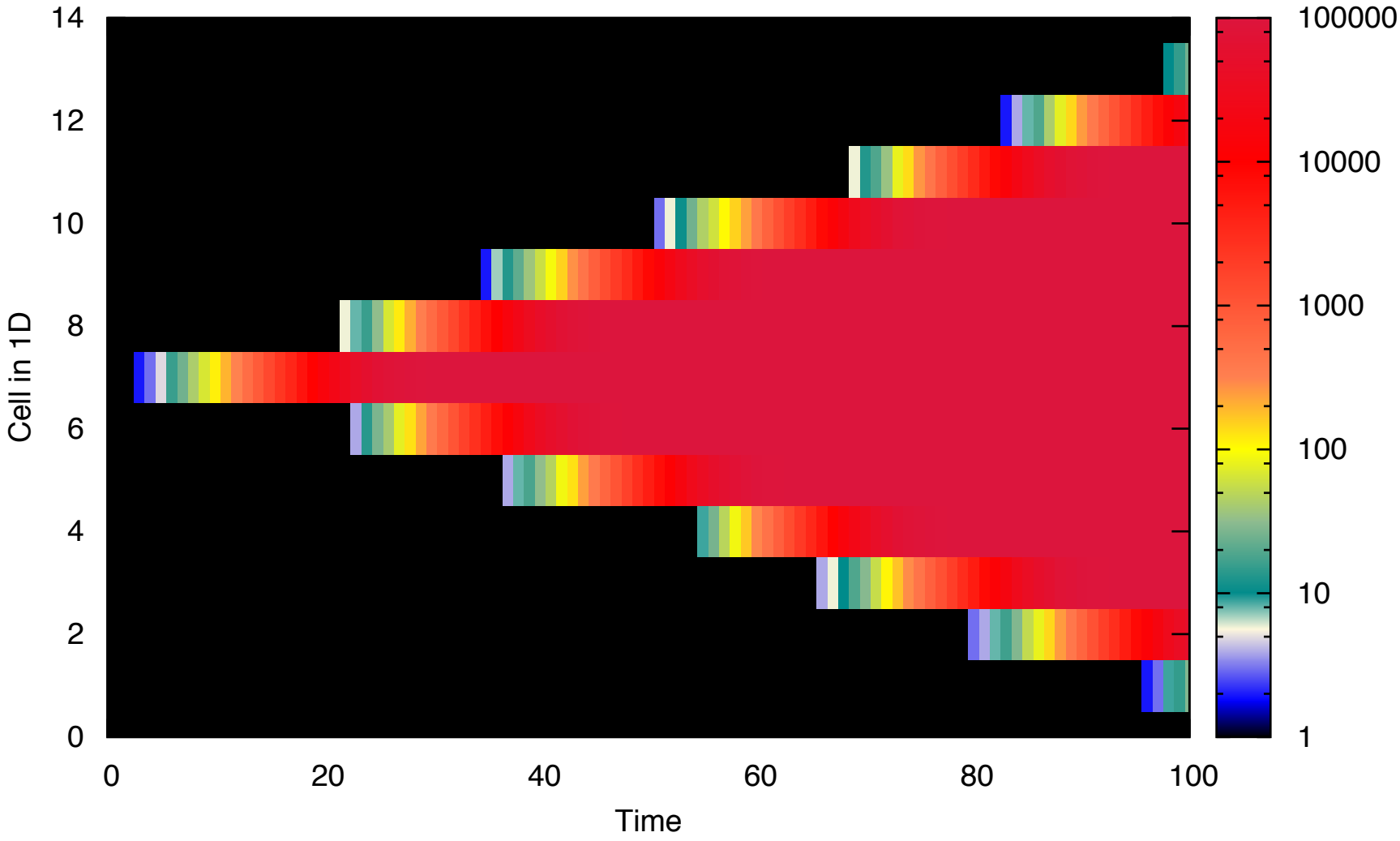


Ex4: 1D15 - Varying Mobility, Gamma = 99.999



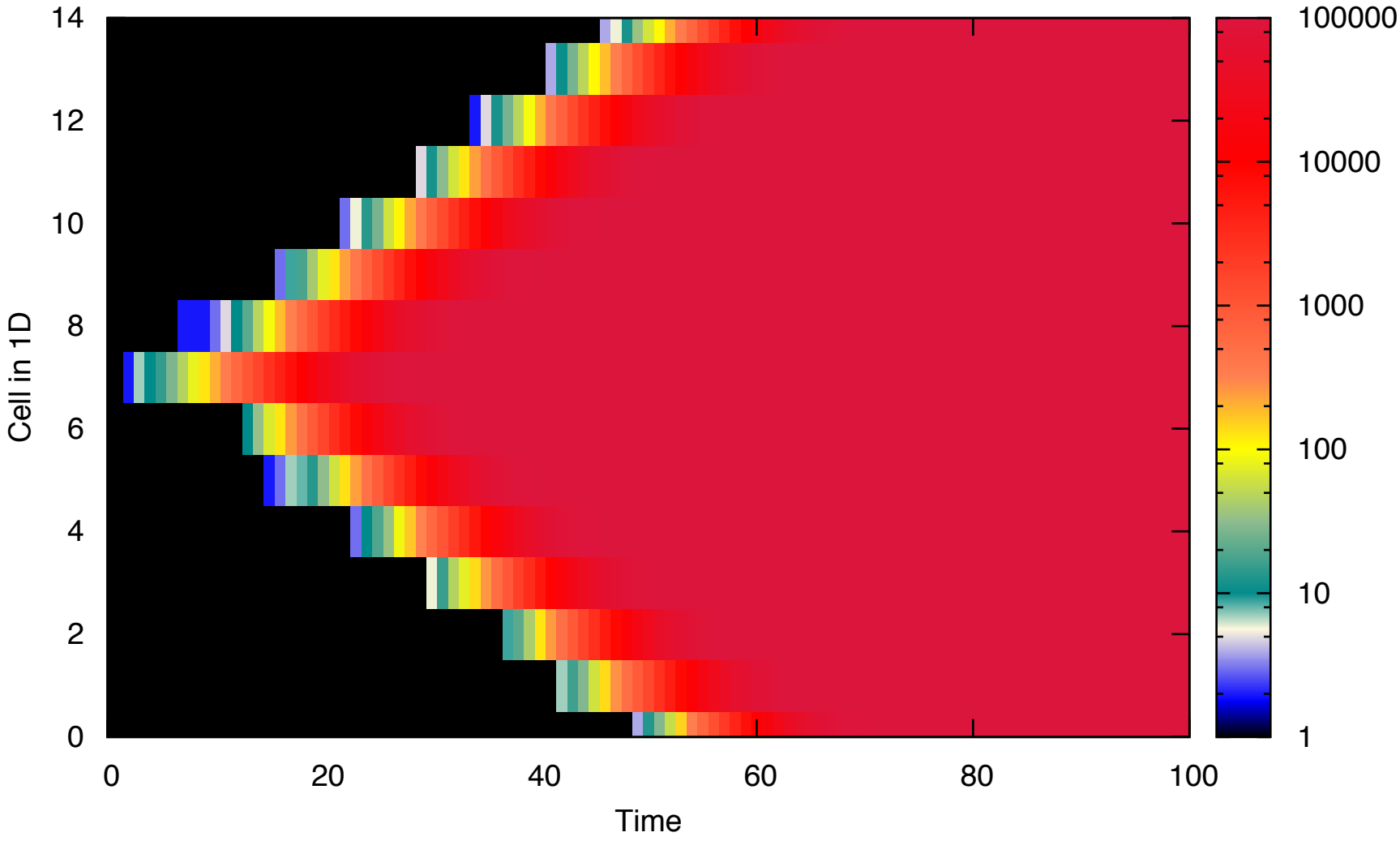
monika.heiner@brunel.ac.uk

Ex4: 1D15 - Varying Mobility, Gamma = 99.99



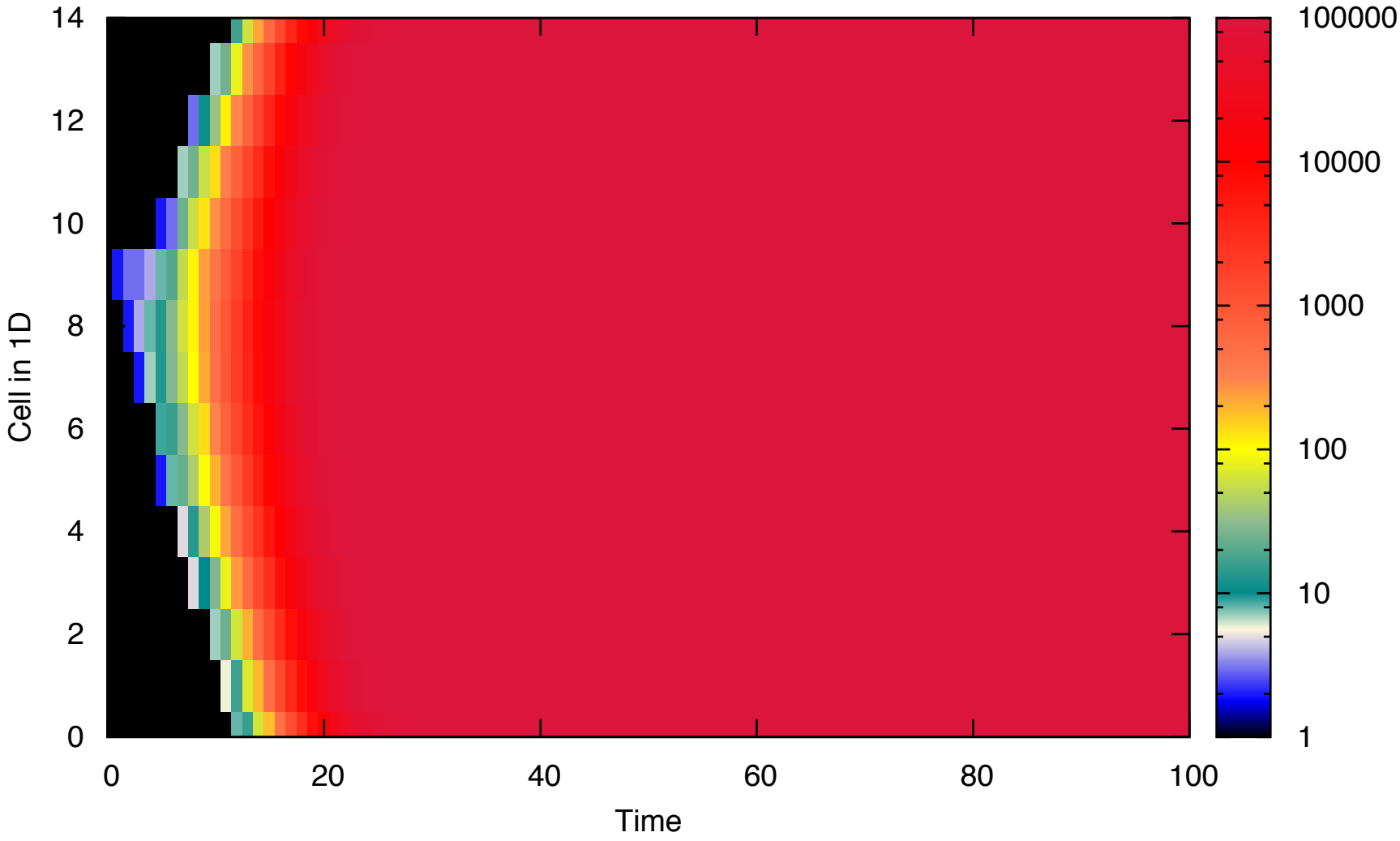
monika.heiner@brunel.ac.uk

Ex4: 1D15 - Varying Mobility, Gamma = 90



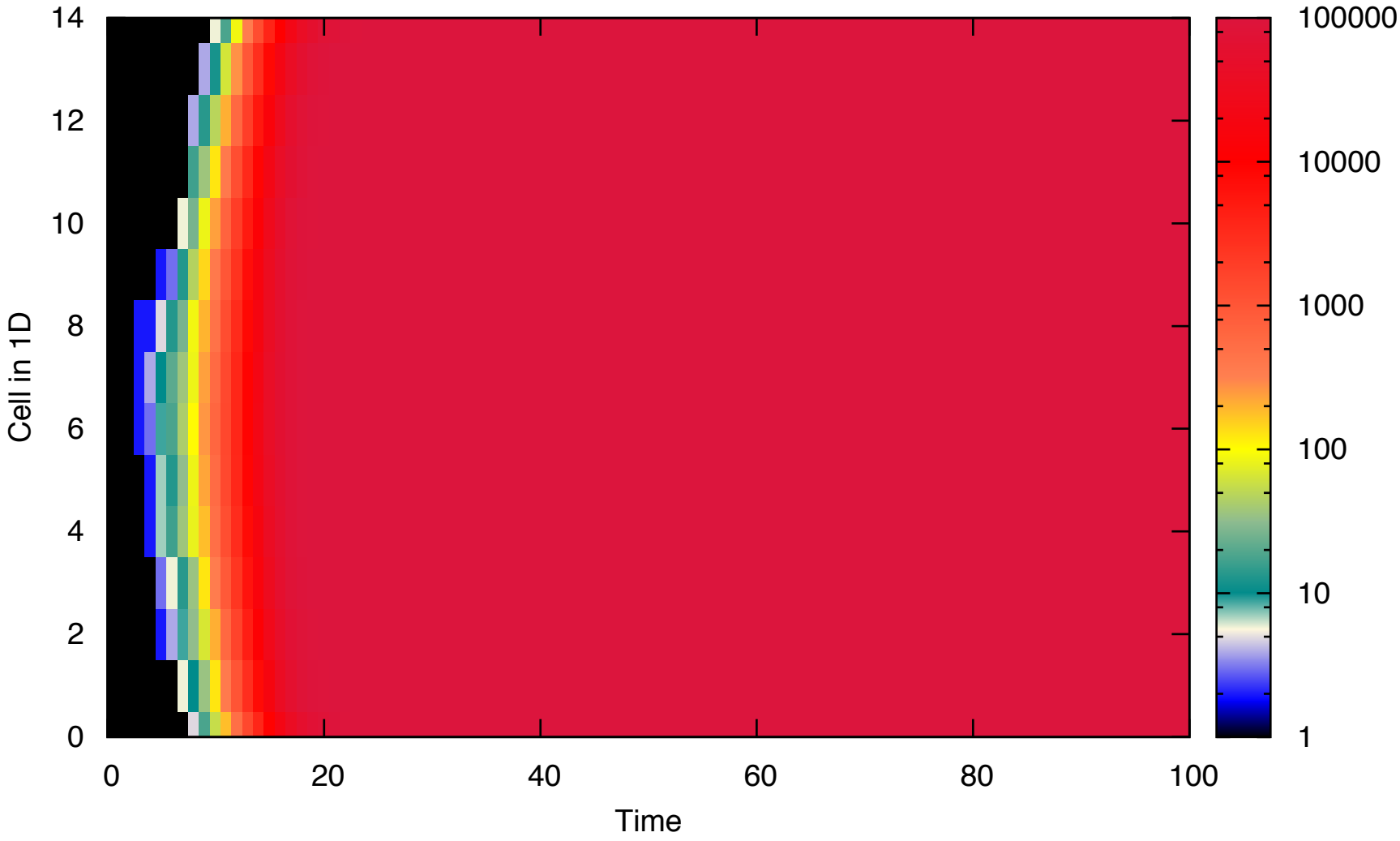
monika.heiner@brunel.ac.uk

Ex4: 1D15 - Varying Mobility, Gamma = 50



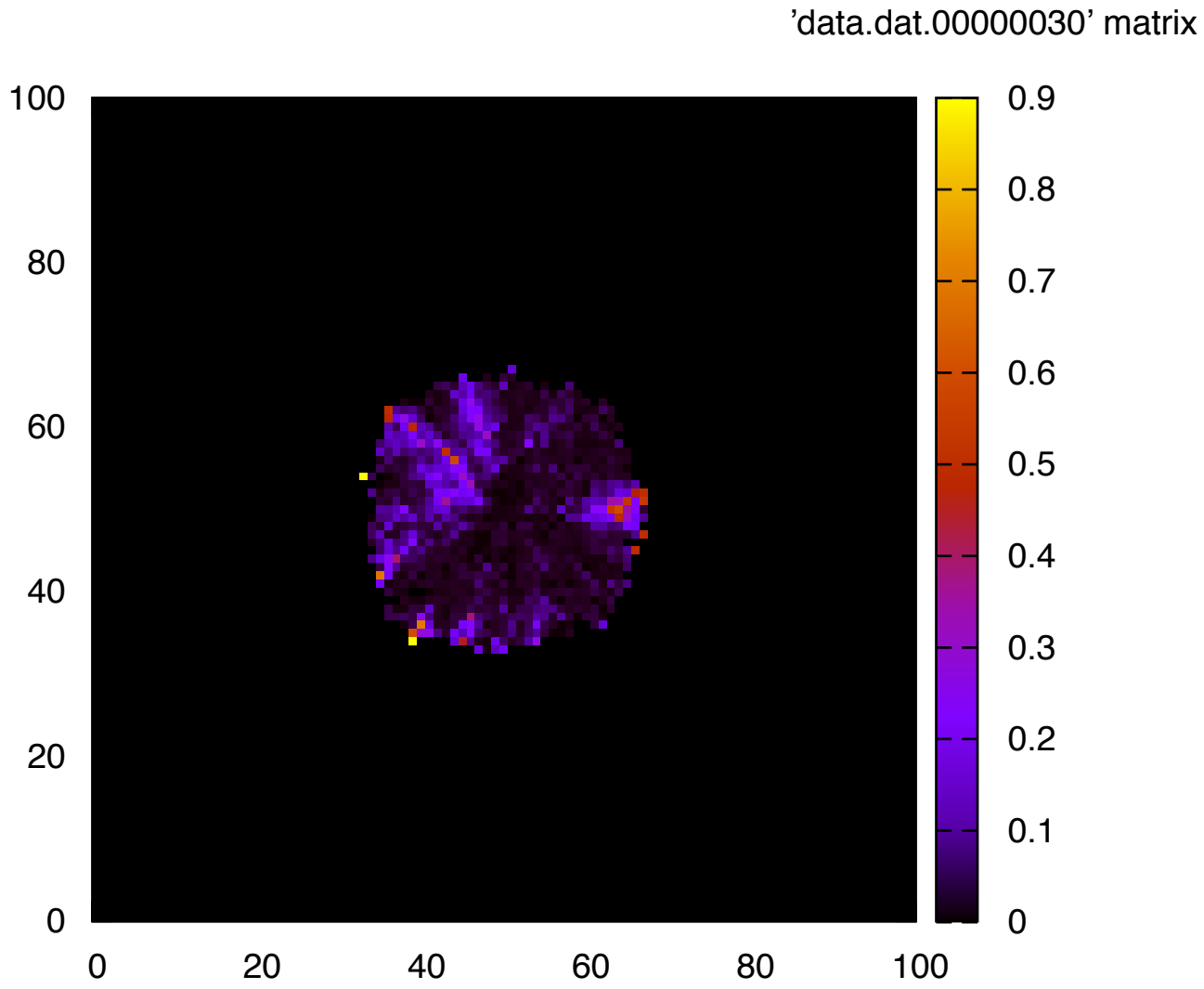
monika.heiner@brunel.ac.uk

Ex4: 1D15 - Varying Mobility, Gamma = 1

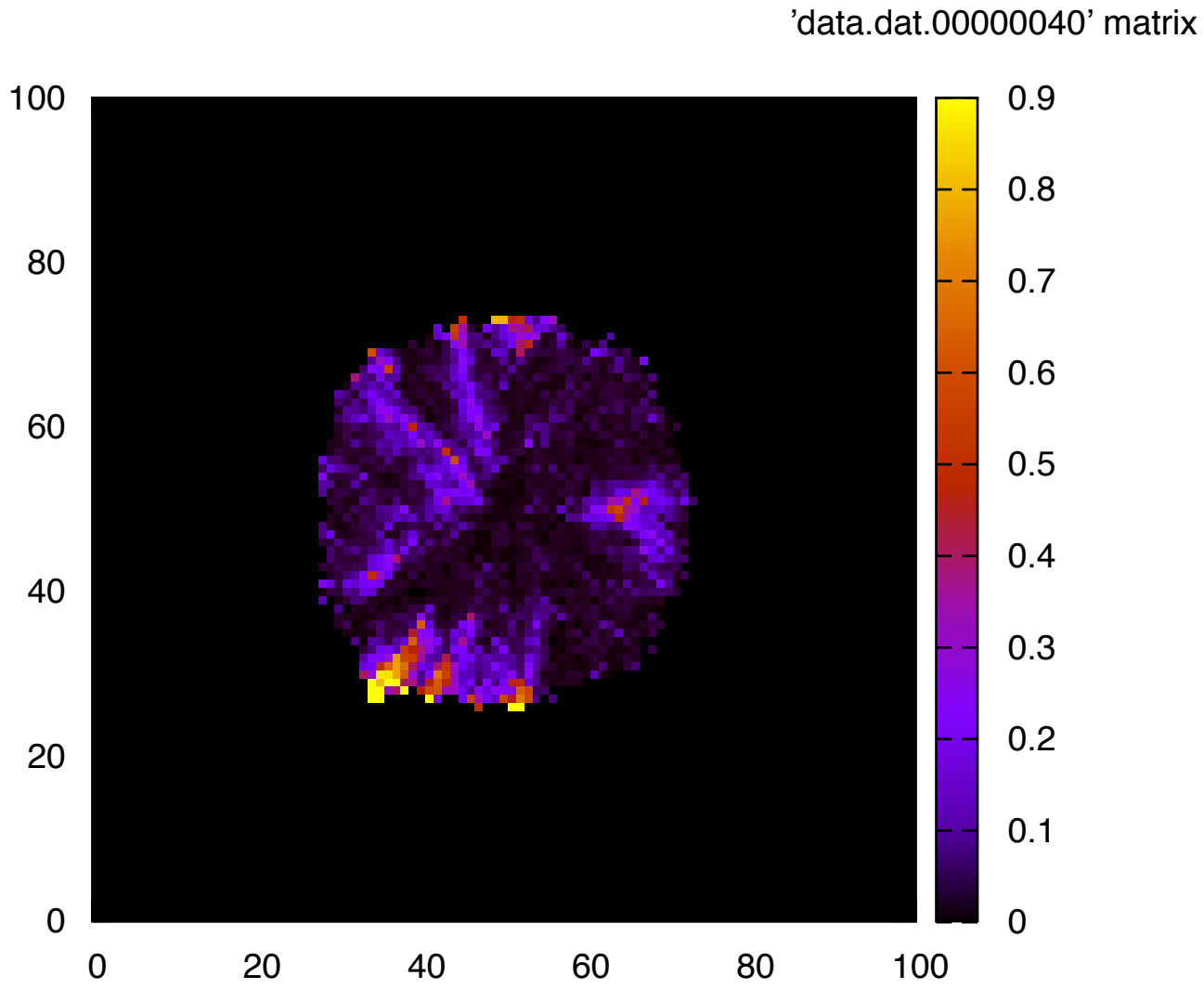


monika.heiner@brunel.ac.uk

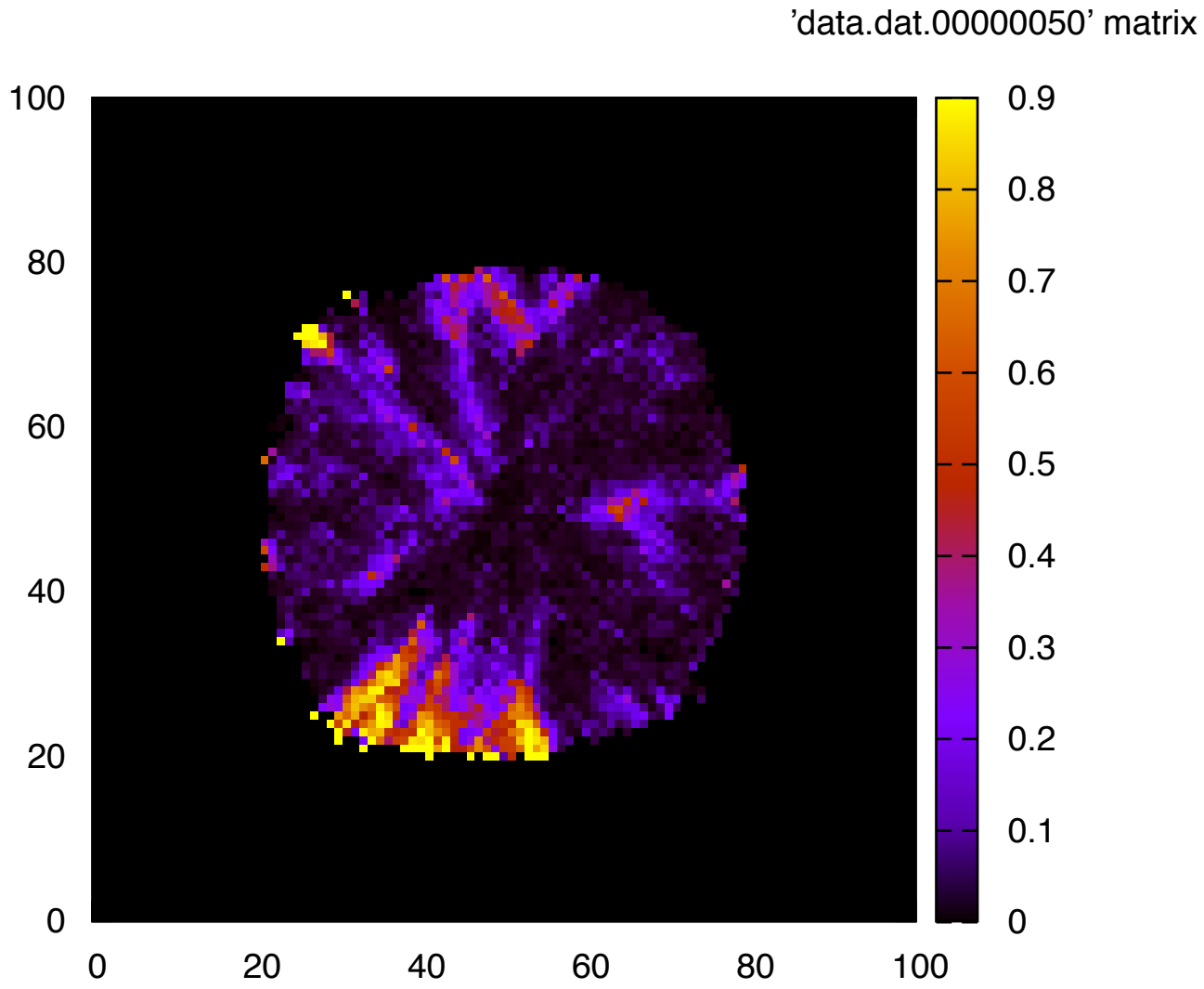
EX4: 2D - TRACE 1 (HIGH, F=1)



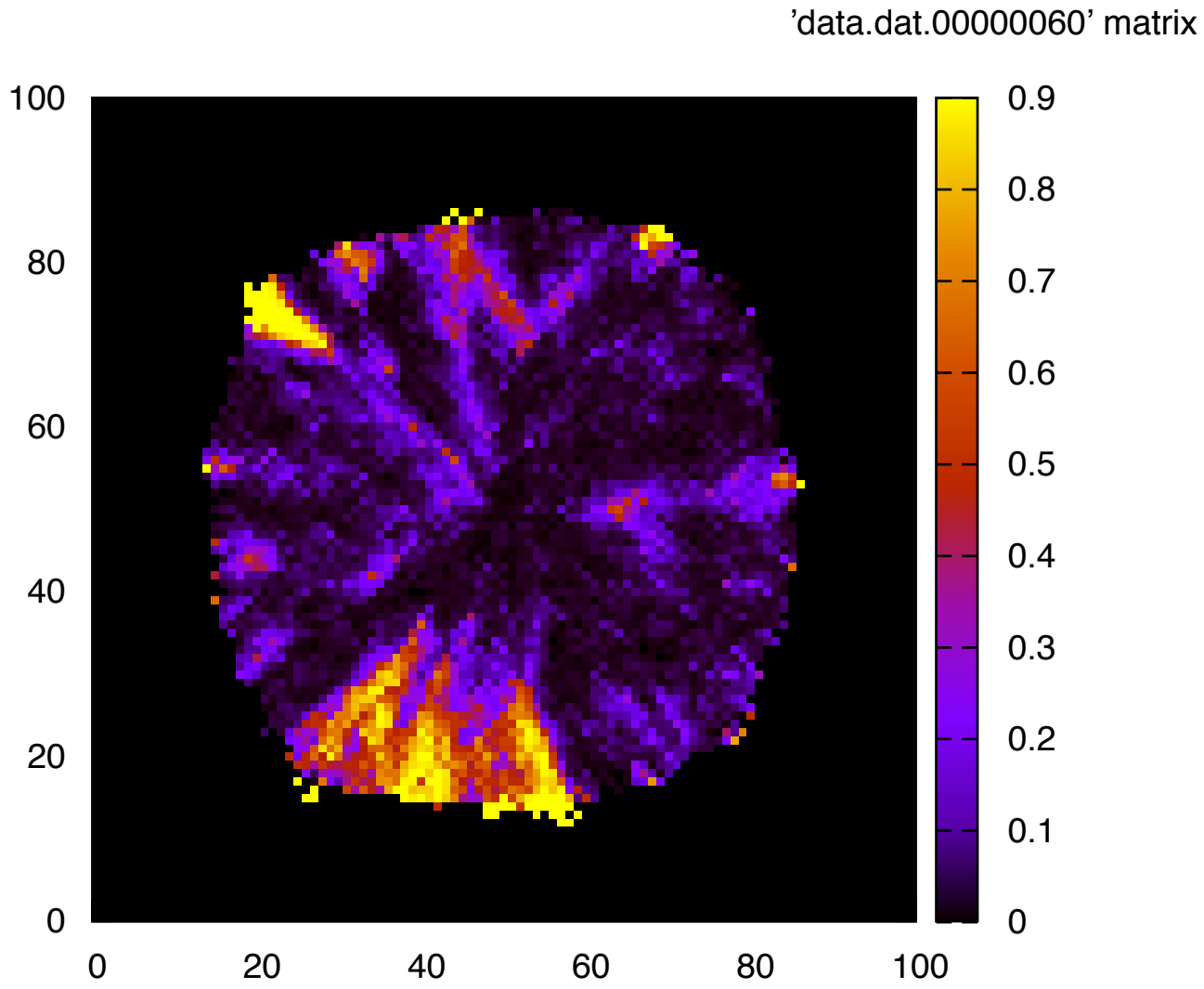
EX4: 2D - TRACE 1 (HIGH, F=1)



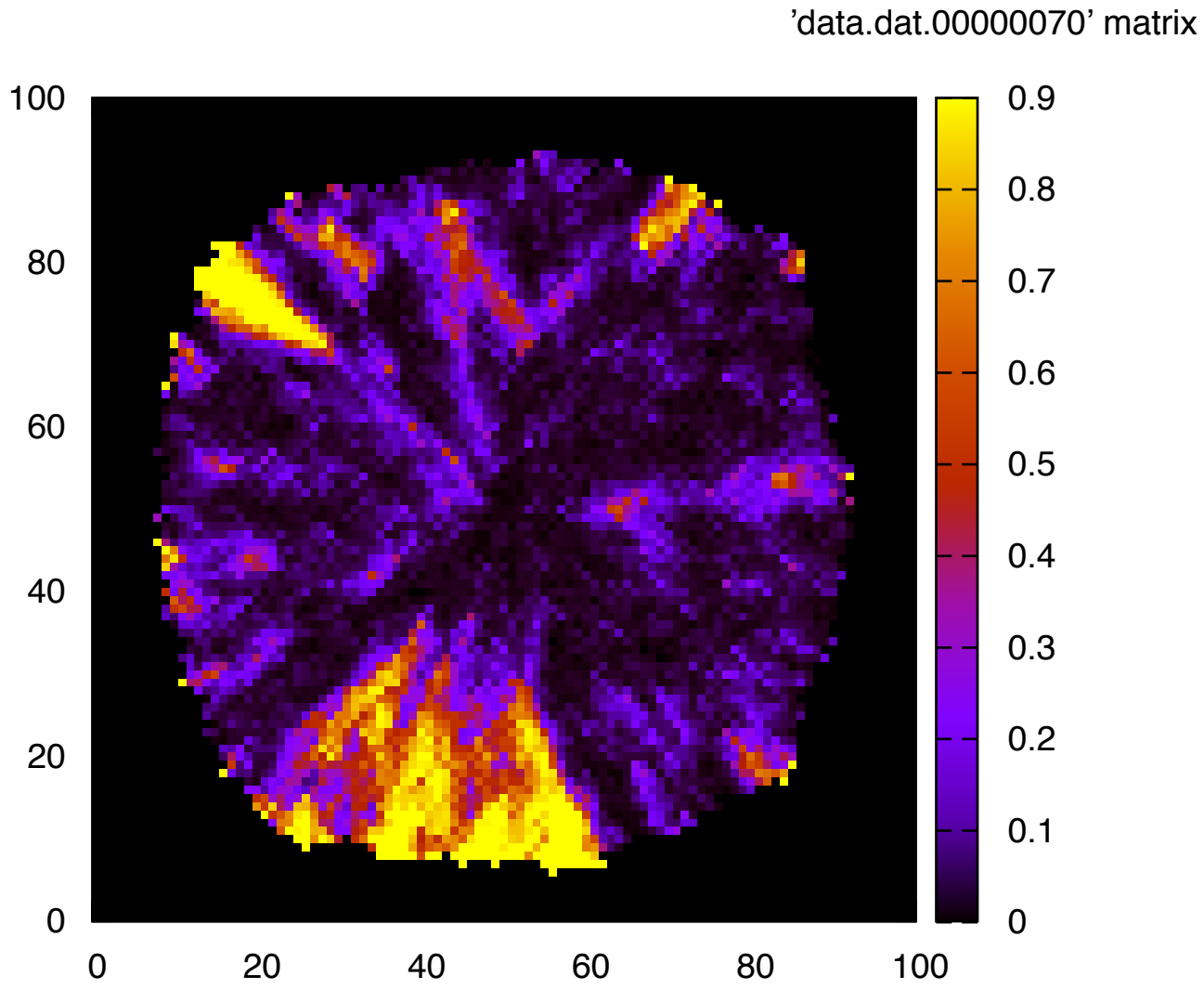
EX4: 2D - TRACE 1 (HIGH, F=1)



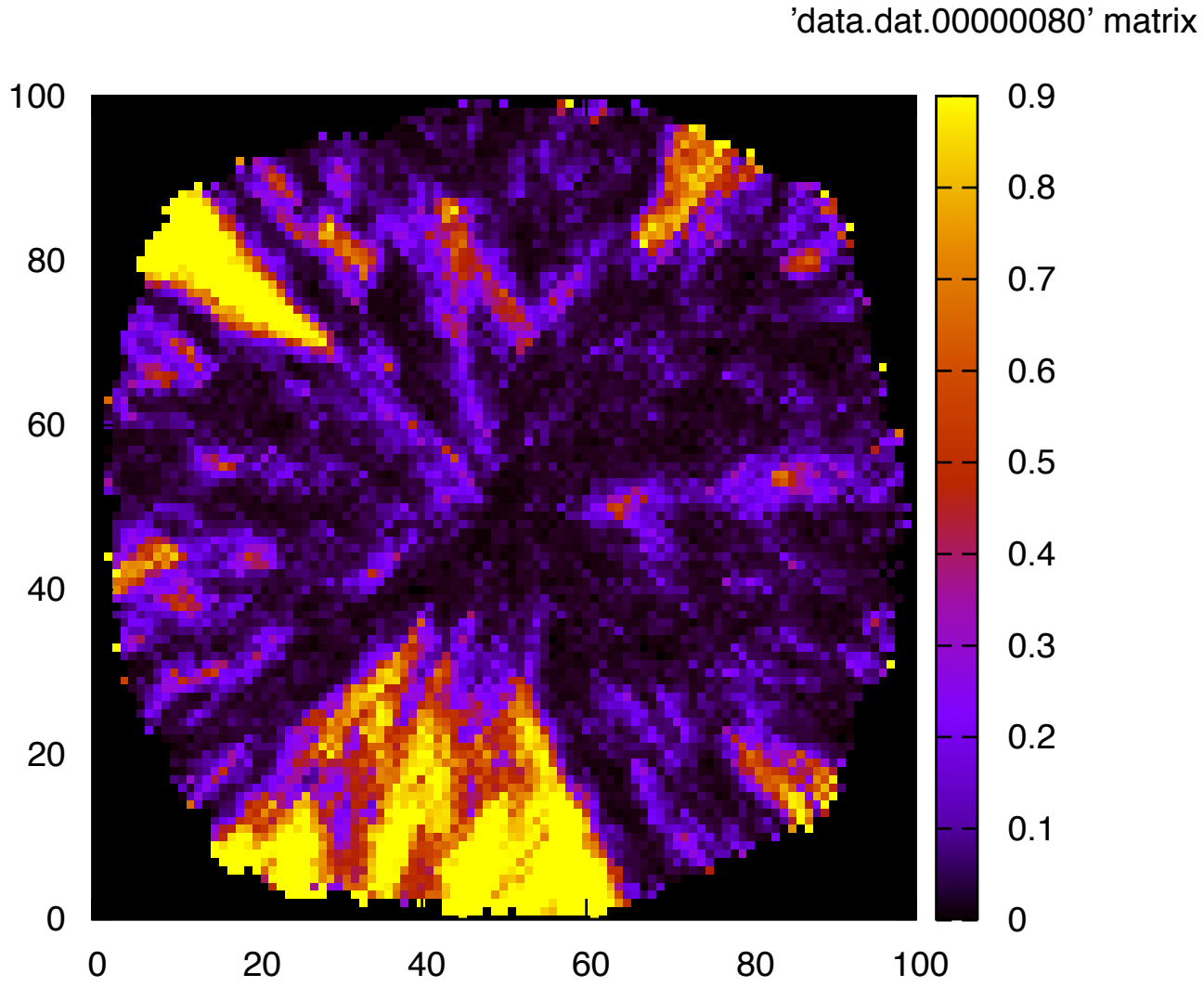
Ex4: 2D - Trace 1 (High, F=1)



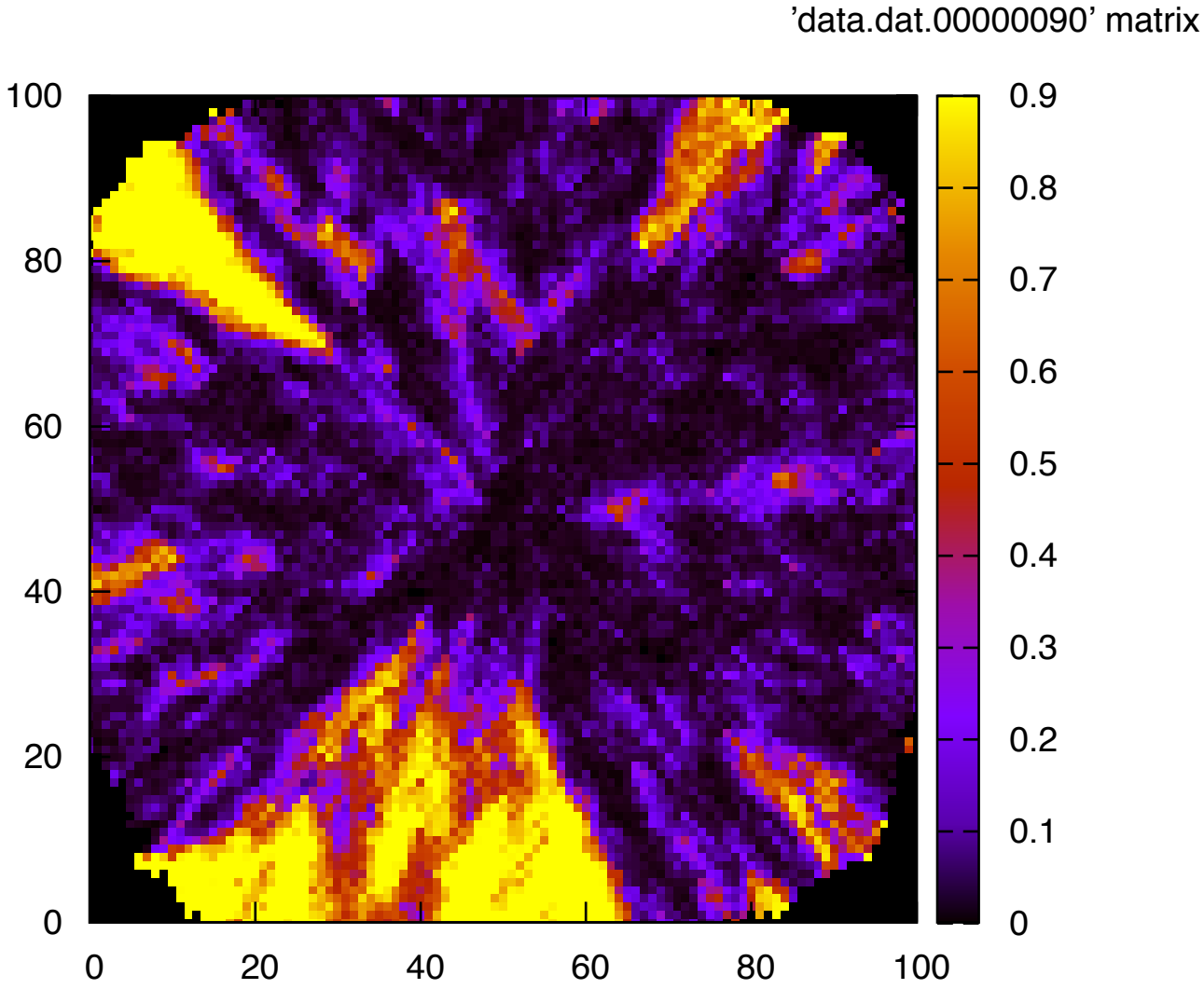
EX4: 2D - TRACE 1 (HIGH, F=1)



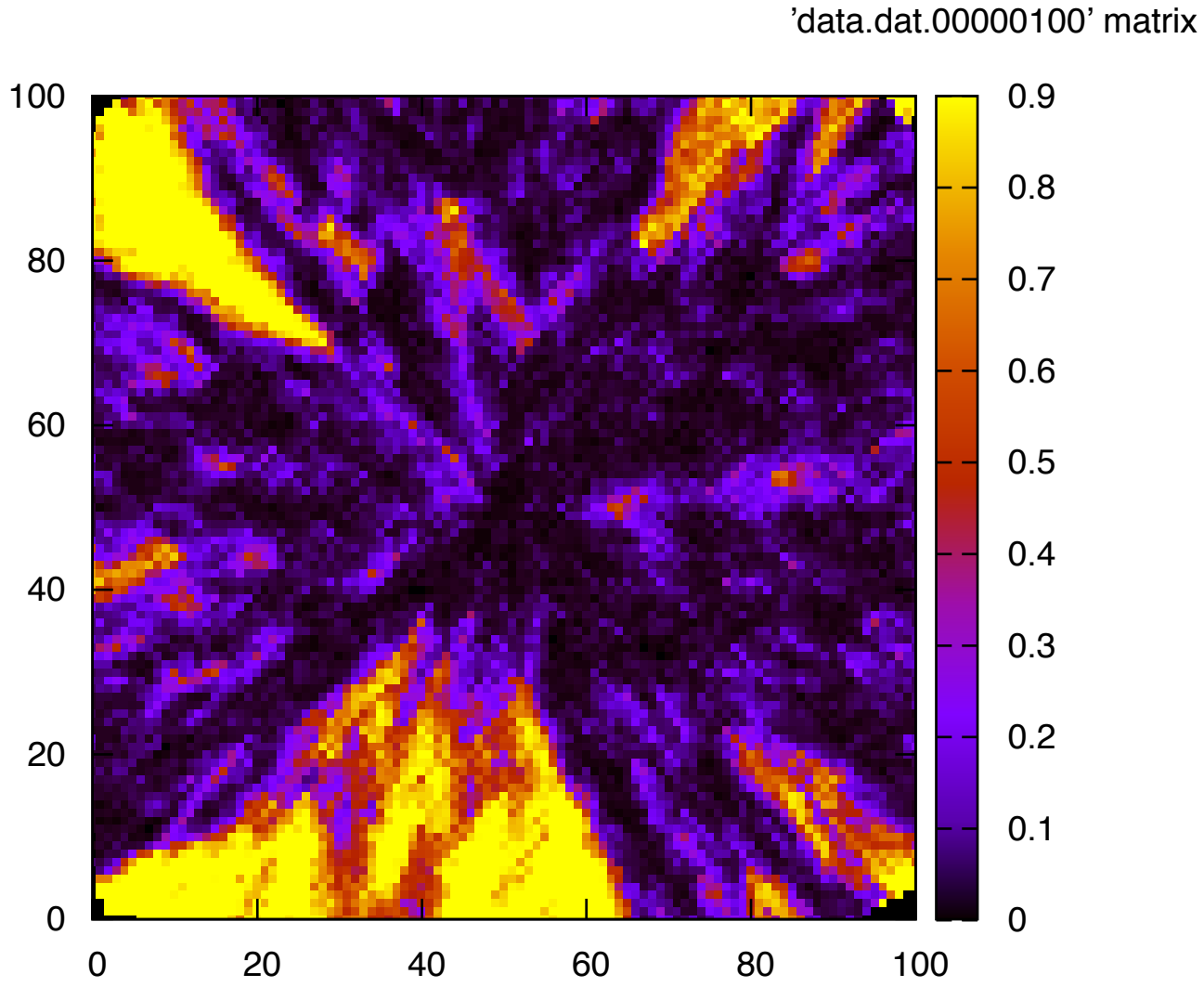
EX4: 2D - TRACE 1 (HIGH, F=1)



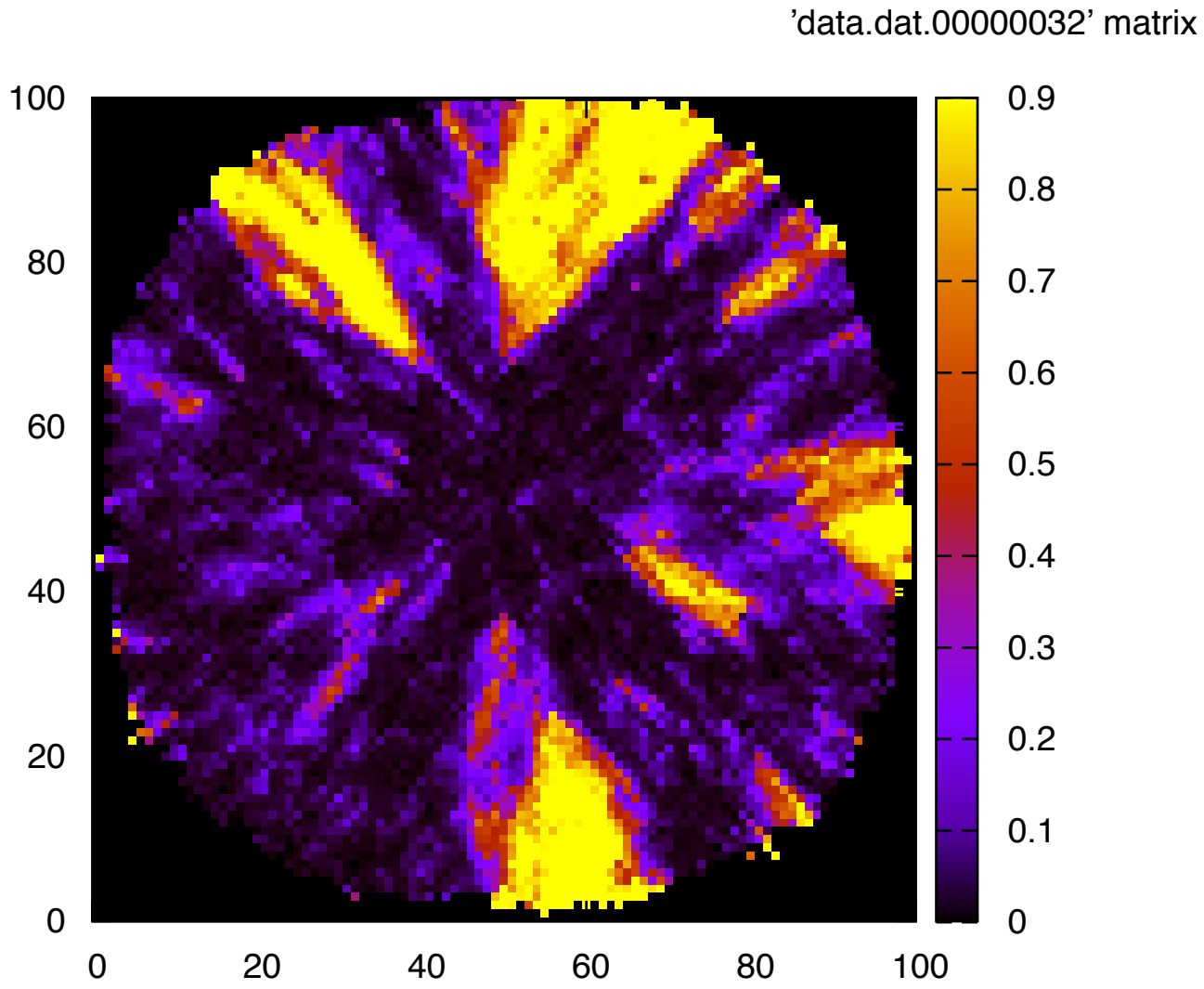
EX4: 2D - TRACE 1 (HIGH, F=1)



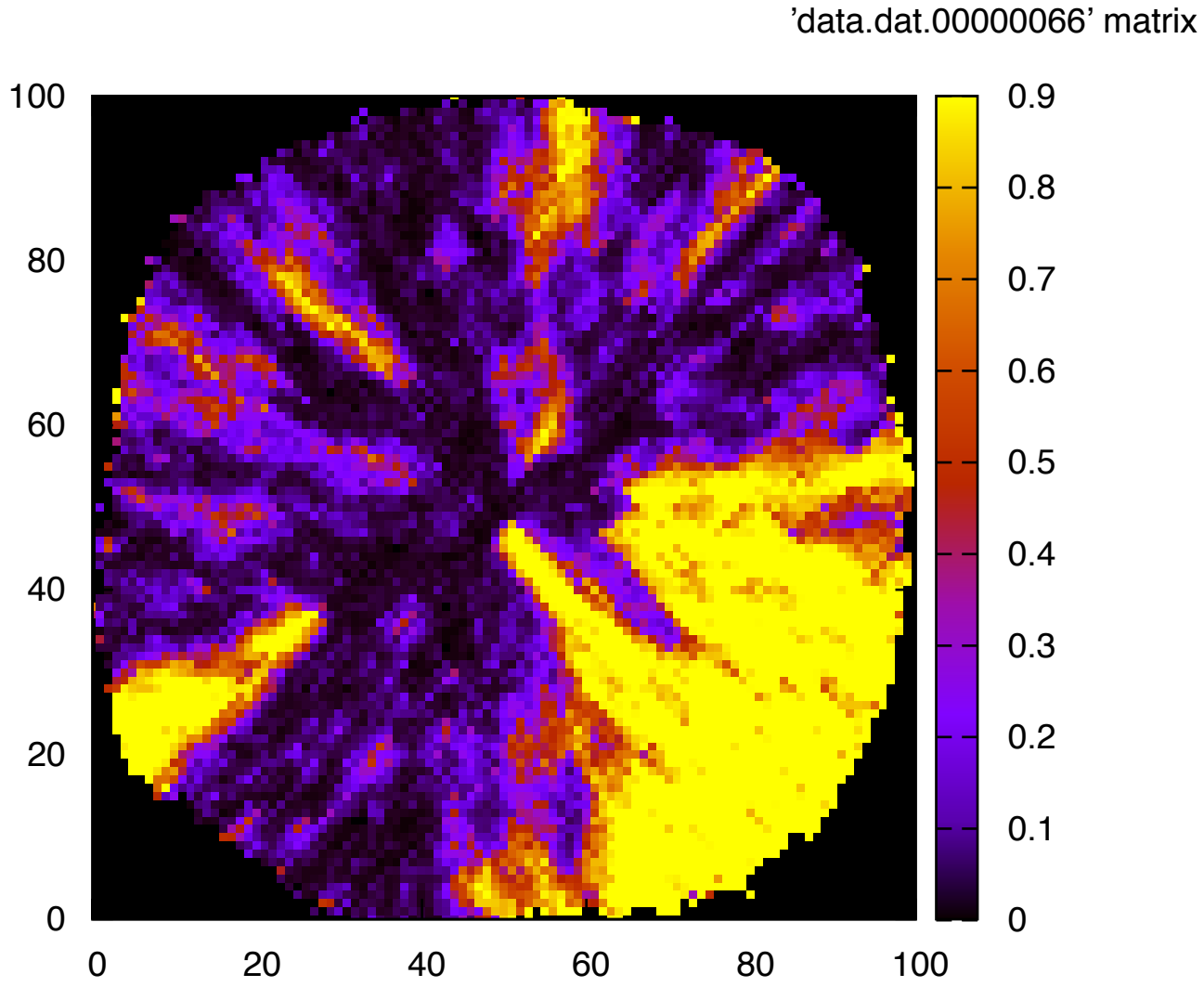
EX4: 2D - TRACE 1 (HIGH, F=1)



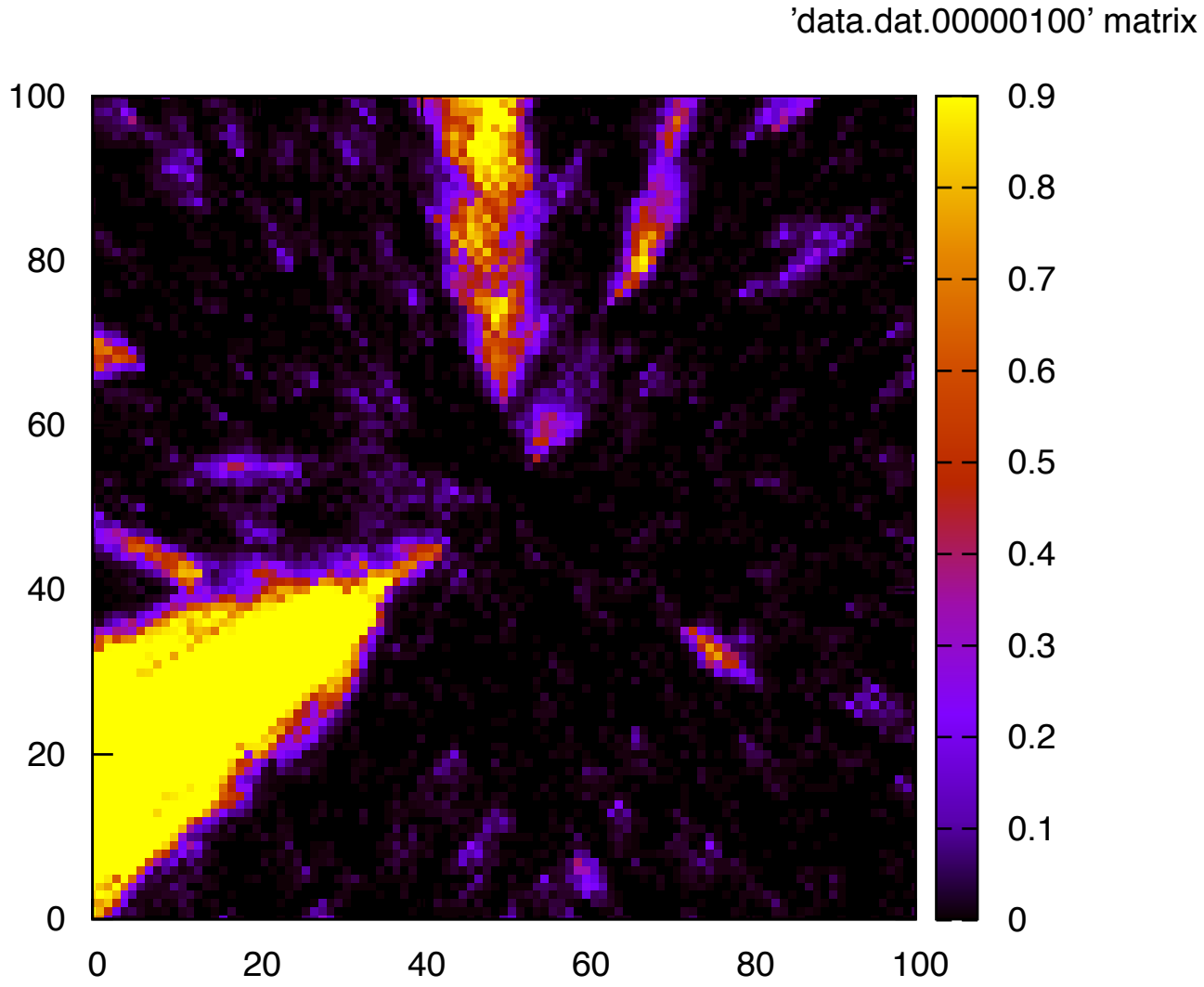
EX4: 2D - TRACE 2 (HIGH, F=1)



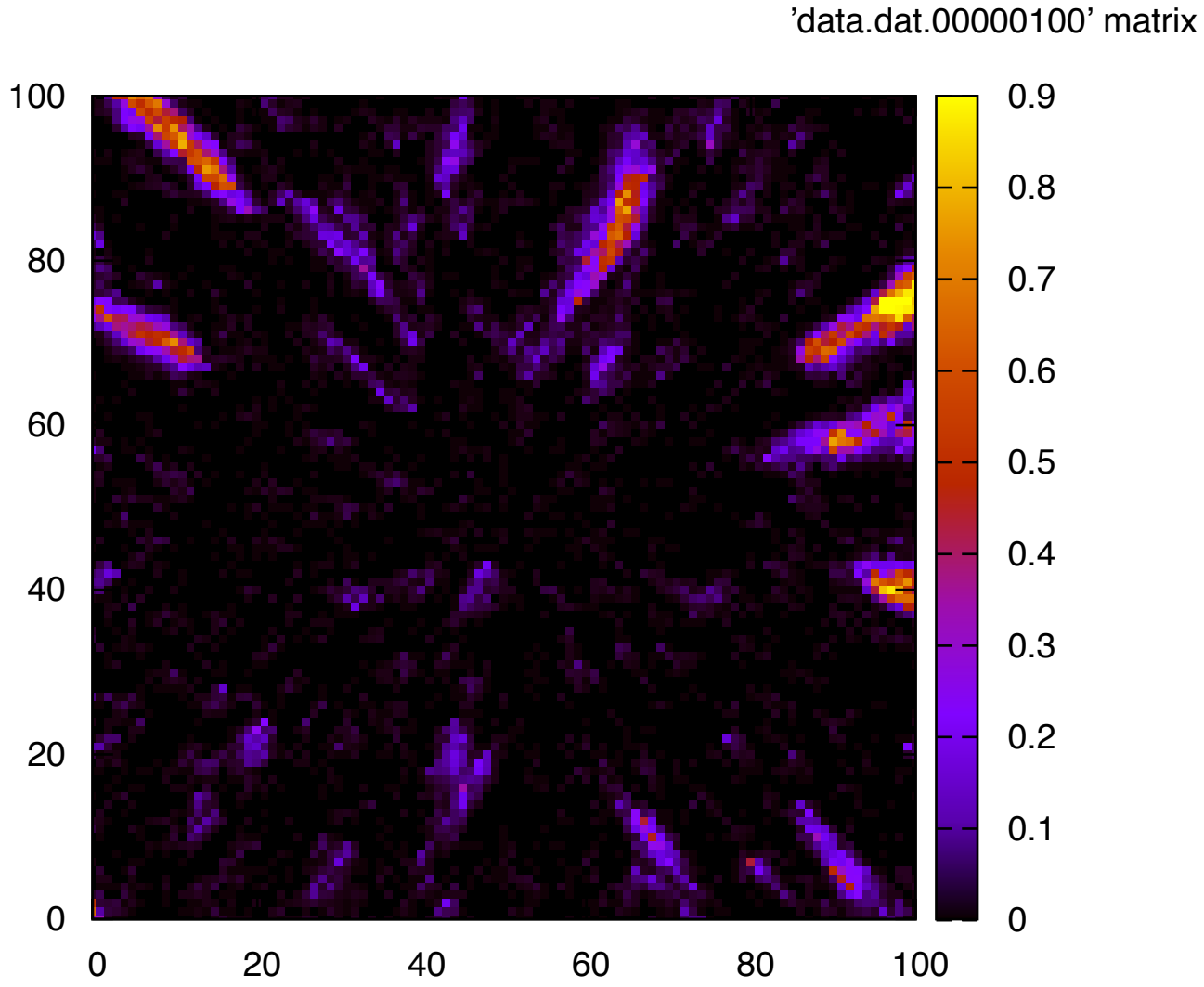
EX4: 2D - TRACE 3 (HIGH, F=1)



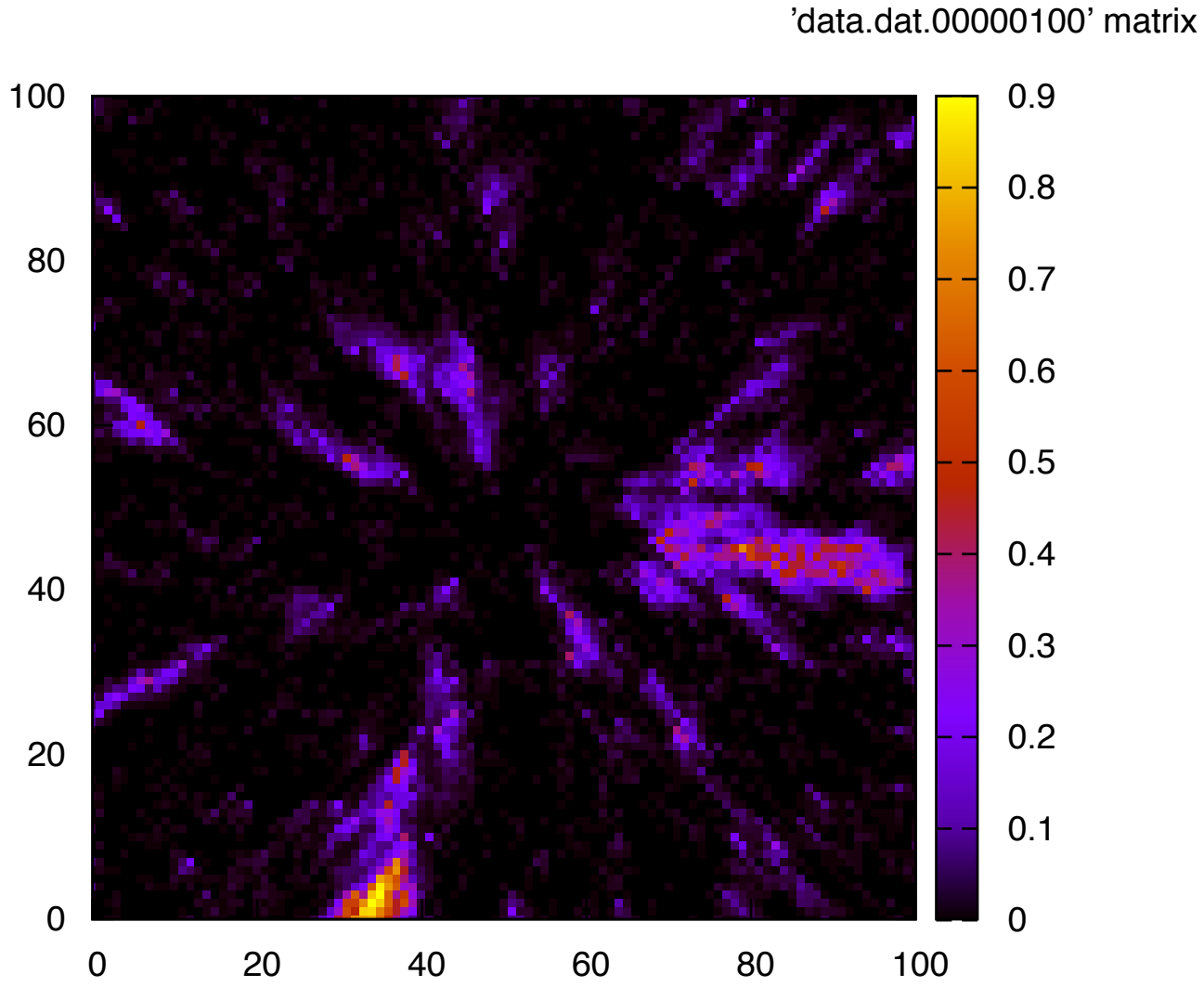
Ex4: 2D - Varying Fitness, Trace 1 (Medium, F=1)



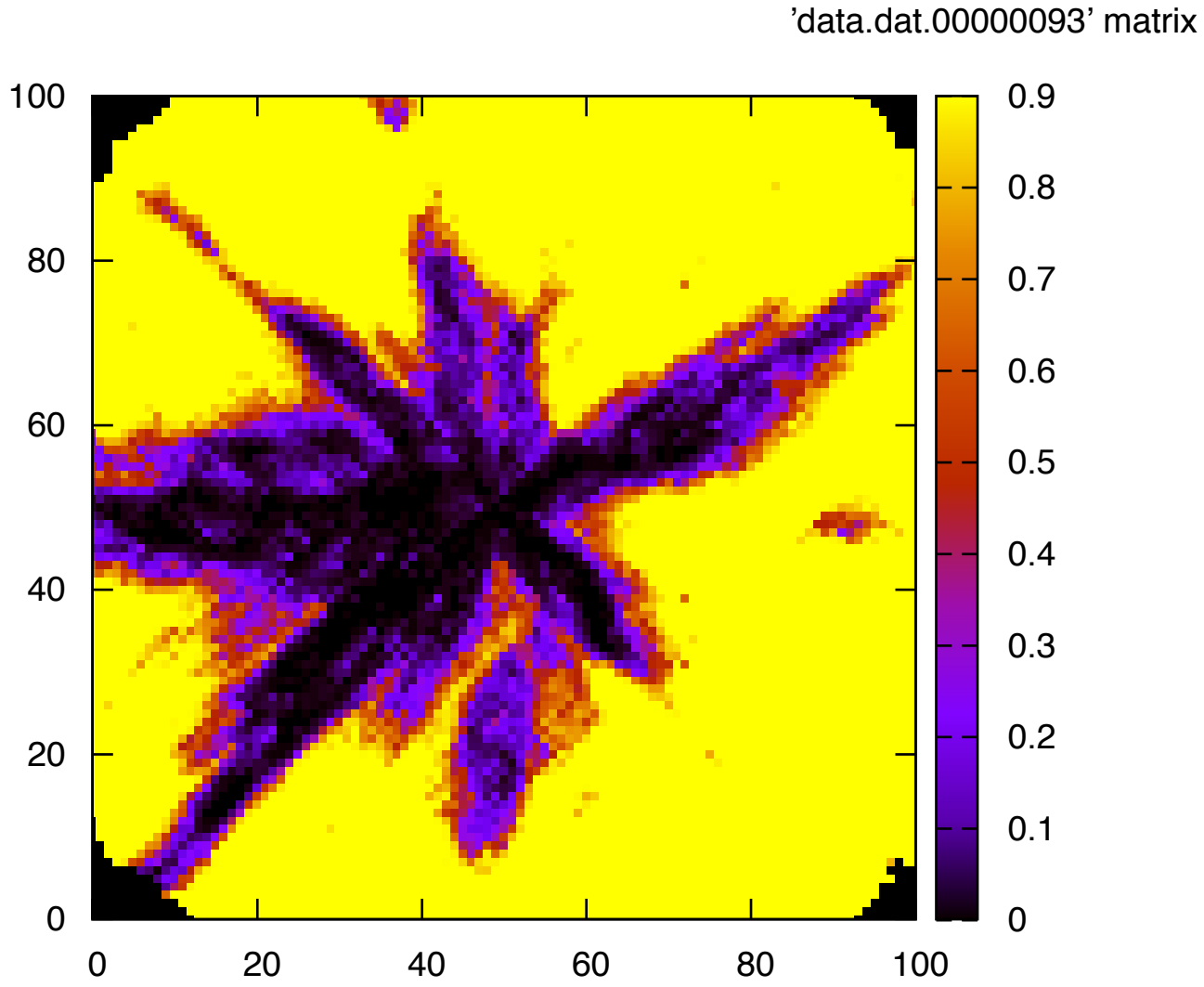
Ex4: 2D - Varying Fitness, Trace 2 (Medium, F=1)



Ex4: 2D - Varying Fitness, Trace 1 (Medium, F=0.99)



Ex4: 2D - Varying Fitness, Trace 1 (Medium, F=0.90)



WHAT NEXT?

- **how to analyse visual data?**
  - -> *auxiliary variables derived from model variables*
  - -> *clustering techniques*
  - -> *shape recognition*
  - -> *visual analytics*

- **use model to predict**
  - -> *mutation rates by measuring the mutation sectors,*
    or just the number of sectors?
  - -> *fitness by measuring angel of sectors*

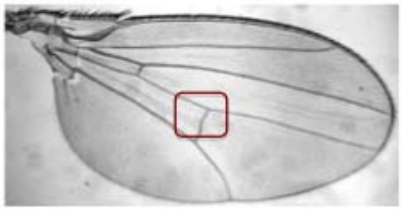
- **possible model extensions / variations**
  - -> *fine tuning of biofilm thickness*
  - -> *multiple gene on/off and their dependencies*
  - -> *log pedigree and/or mobility*

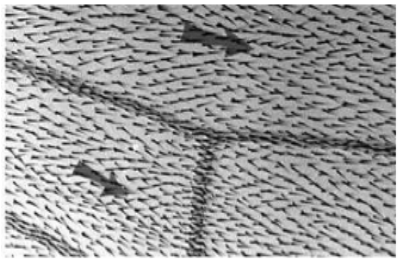
SUMMARY

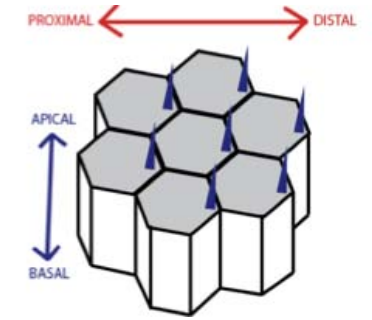
MULTISCALE CHALLENGES

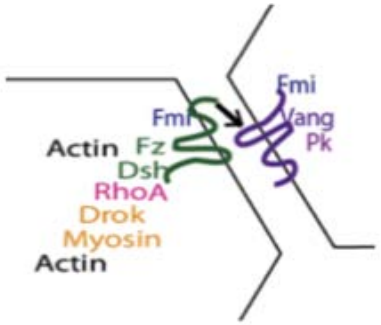
- repetition of components
- variation of components
- organisation of components
- communication between components
- mobility / motility
- replication / deletion of components
- hierarchical organisation of components

- dynamic grid size
- irregular / semi-regular organisation of components









monika.heiner@brunel.ac.uk

CREDITS

- **EPSRC Research Grant EP/I036168/1**
- **inspiring discussions**

  David Gilbert, Nigel Saunders
- **Snoopy + Charlie + Marcie development**

  Christian Rohr, Fei Liu, Mostafa Herayj

  Martin Schwarick, Jan Wegener
- **Gnuplots**

  Mary Ann Blätke, Daniele Maccagnola, Jan Wegener
- **further case studies**

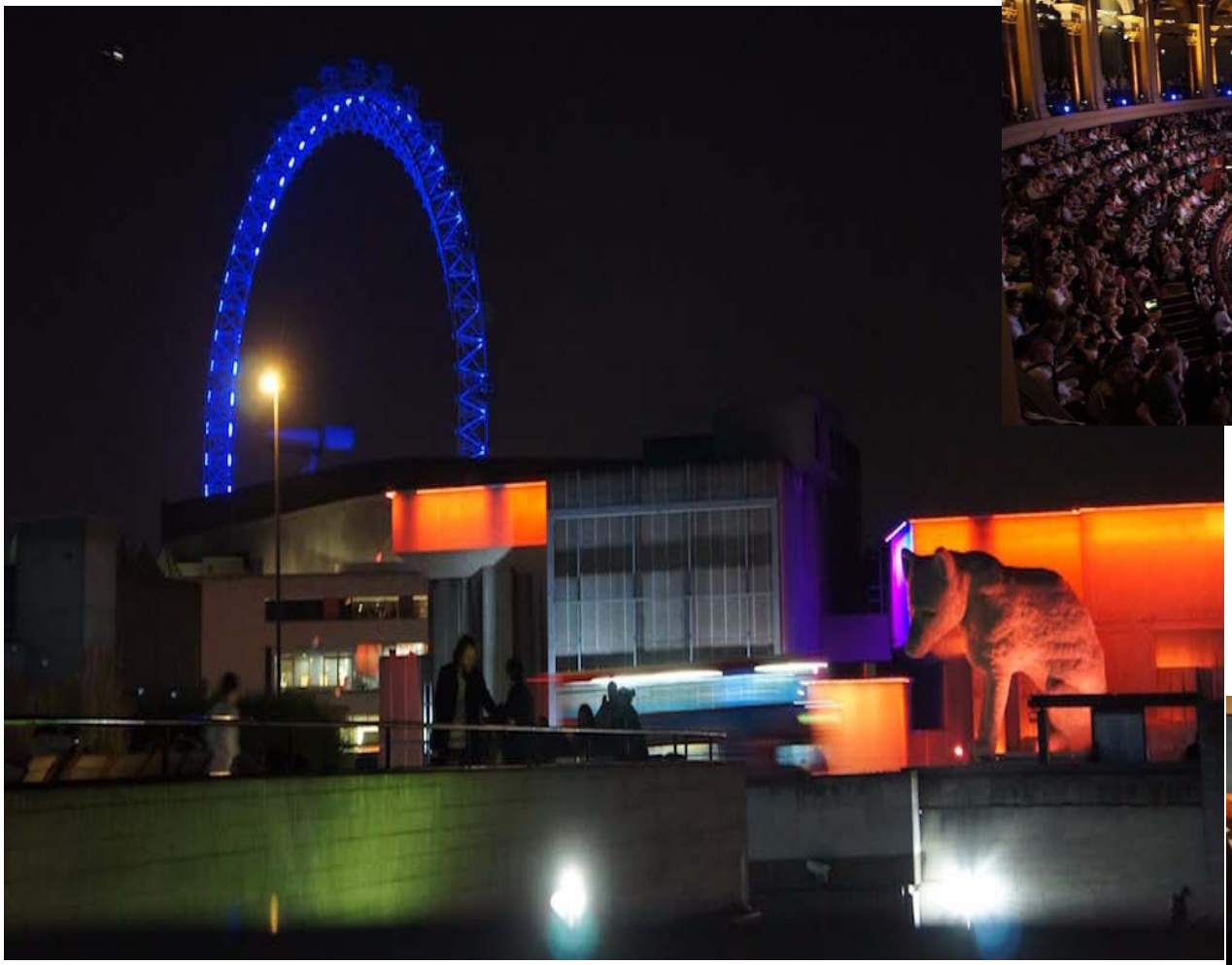
  Halo bacterium - Wolfgang Marwan

  PCP in fly wing - Pam Gao, David Tree

  Dictopat - team involving Andrzej Kierzek, Simon Hardy, ...

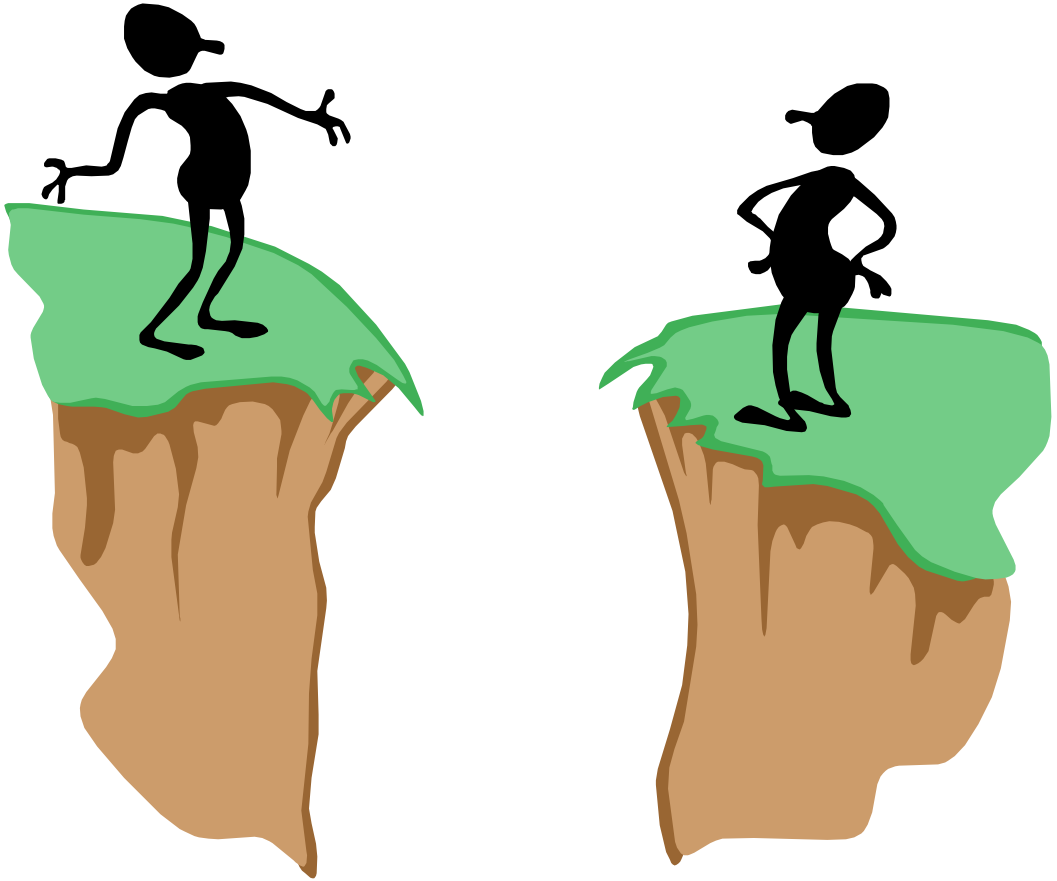
EPILOGUE

TIMEOUT



VARIETY IS THE SPICE OF LIFE

OUTLOOKS



THANKS !

HHTP://WWW-DSSZ.INFORMATIK.TU-COTTBUS.DE