FROM PETRI NETS TO PARTIAL DIFFERENTIAL EQUATIONS

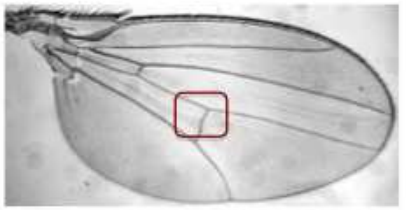
- SPATIAL MODELLING IN SYSTEMS BIOLOGY -

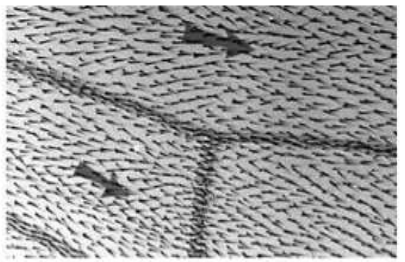
Monika Heiner

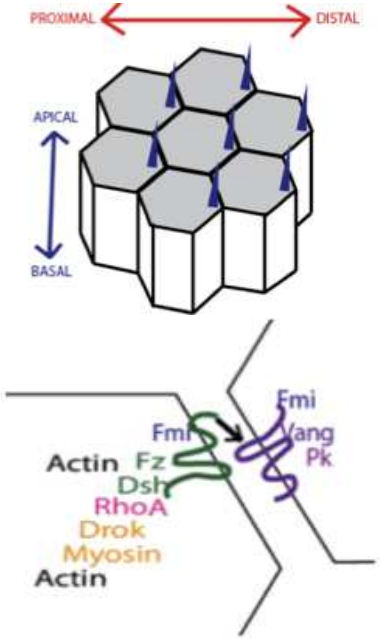
Brandenburg University of Technology
Computer Science Institute

OUTLINE

- **FRAMEWORK**
  - -> *unifying four paradigms: QPN - SPN - CPN - HPN*
- **OUR TOOLBOX**
  - -> *Snoopy / Marcie / Charlie*
- **MODELLING BIO PETRI NETS**
  - -> *composition from standard components*
  - -> *bottom-up (reverse engineering)*
  - -> *genome-controlled model generation*
- **COLOURED PETRI NETS**
  - -> *colouring space -> PDE*
  - -> *phase variation*
  - -> *planar cell polarity*







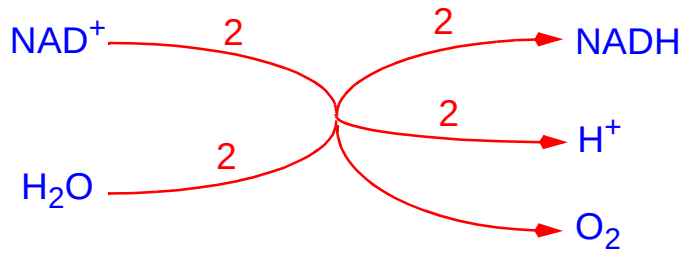
THE PETRI NET FRAMEWORK

monika.heiner@b-tu.de

. . .

ARE NETWORKS OF BIOCHEMICAL REACTIONS

$$2\ NAD^+ + 2\ H_2O \rightarrow 2\ NADH + 2\ H^+ + O_2$$



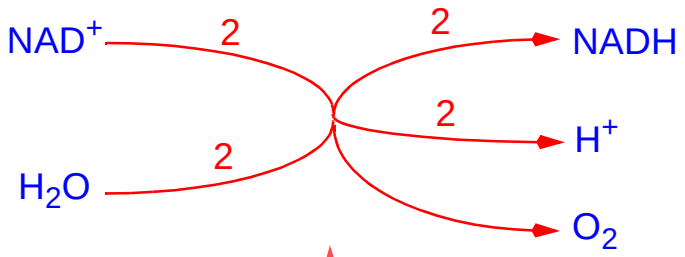
. . .

ARE NETWORKS OF BIOCHEMICAL REACTIONS

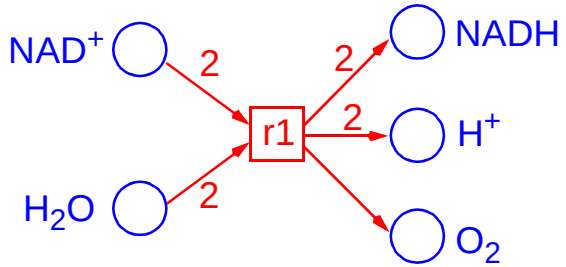
. . .

NATURALLY EXPRESSIBLE AS PETRI NETS

$2\ NAD^+ + 2\ H_2O \rightarrow 2\ NADH + 2\ H^+ + O_2$

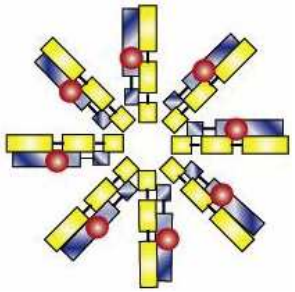


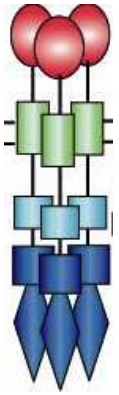
hyper-arcs

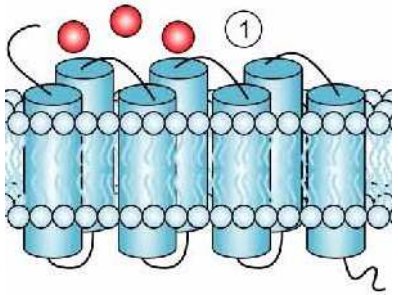


PLACES, TRANSITIONS - SOME BIO INTERPRETATIONS

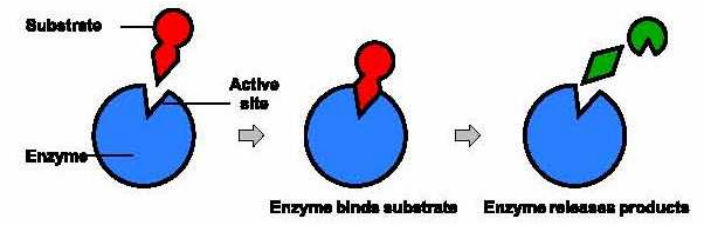
- **places -> model variables**
  - -> *(bio-) chemical compounds*
  - -> *proteins*
  - -> *protein conformations*
  - -> *complexes*
  - -> *genes, . . . , etc.*
  - *. . . in different locations*





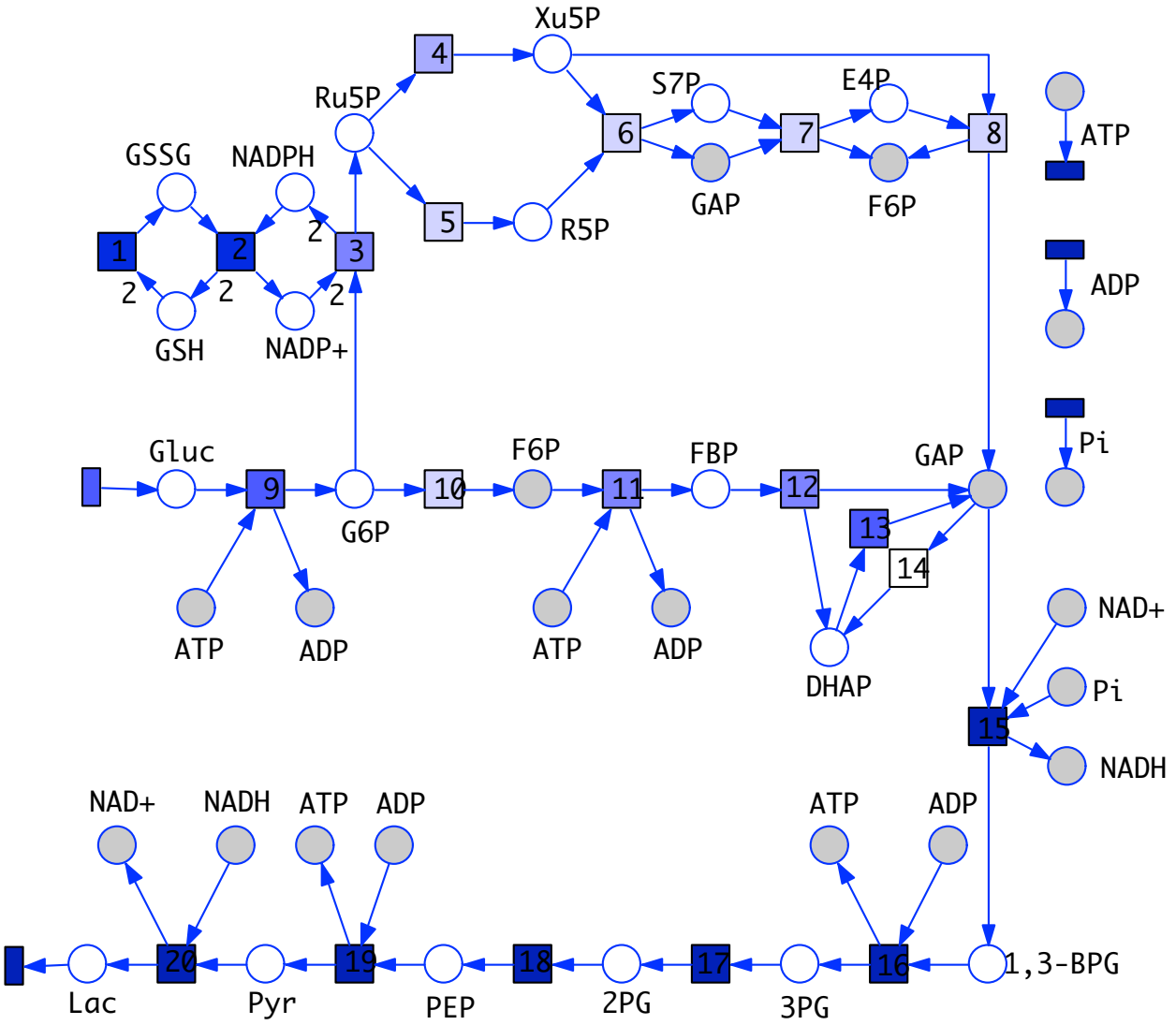


- **transitions -> atomic events**
  - -> *(stoichiometric) chemical reaction*
  - -> *complexation / decomplexation*
  - -> *phosphorylation / dephosphorylation*
  - -> *conformational change*
  - -> *transport step, . . . , etc.*
  - *. . . in different locations*

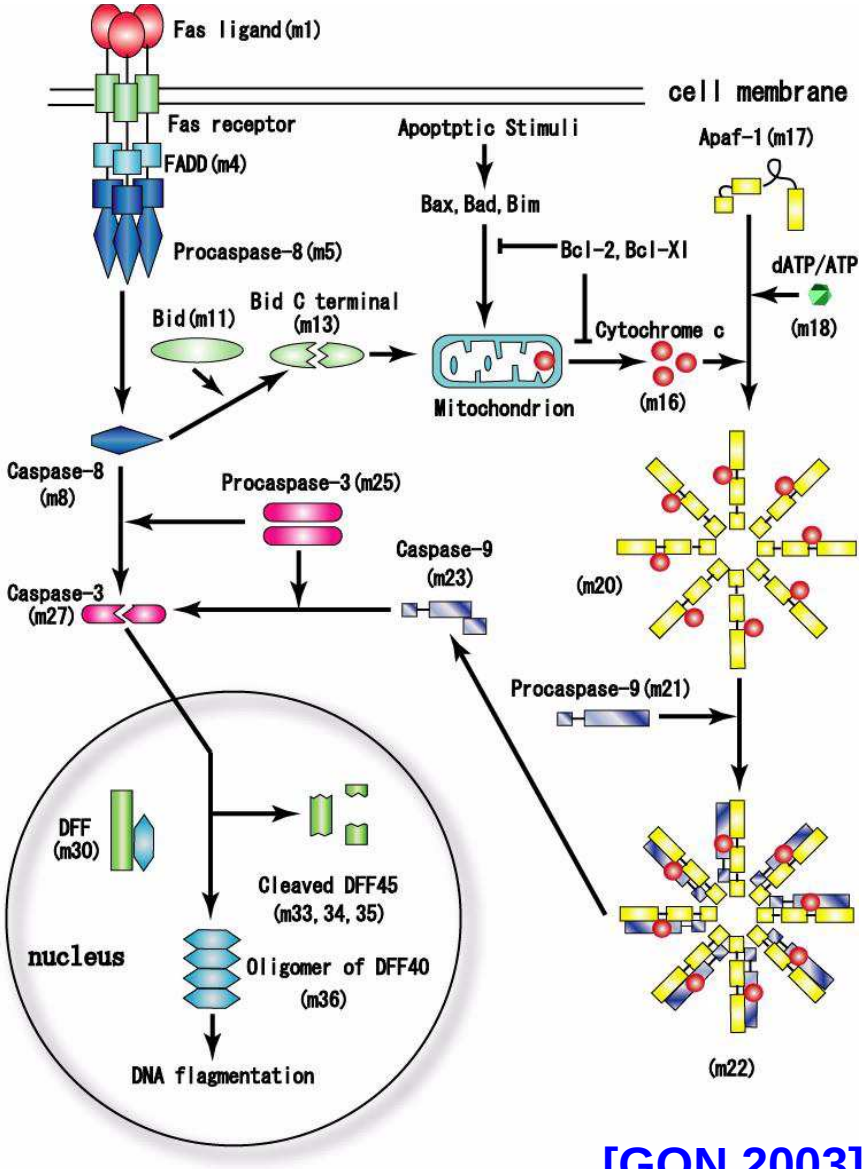


Ex1 - Glycolysis and Pentose Phosphate Pathway

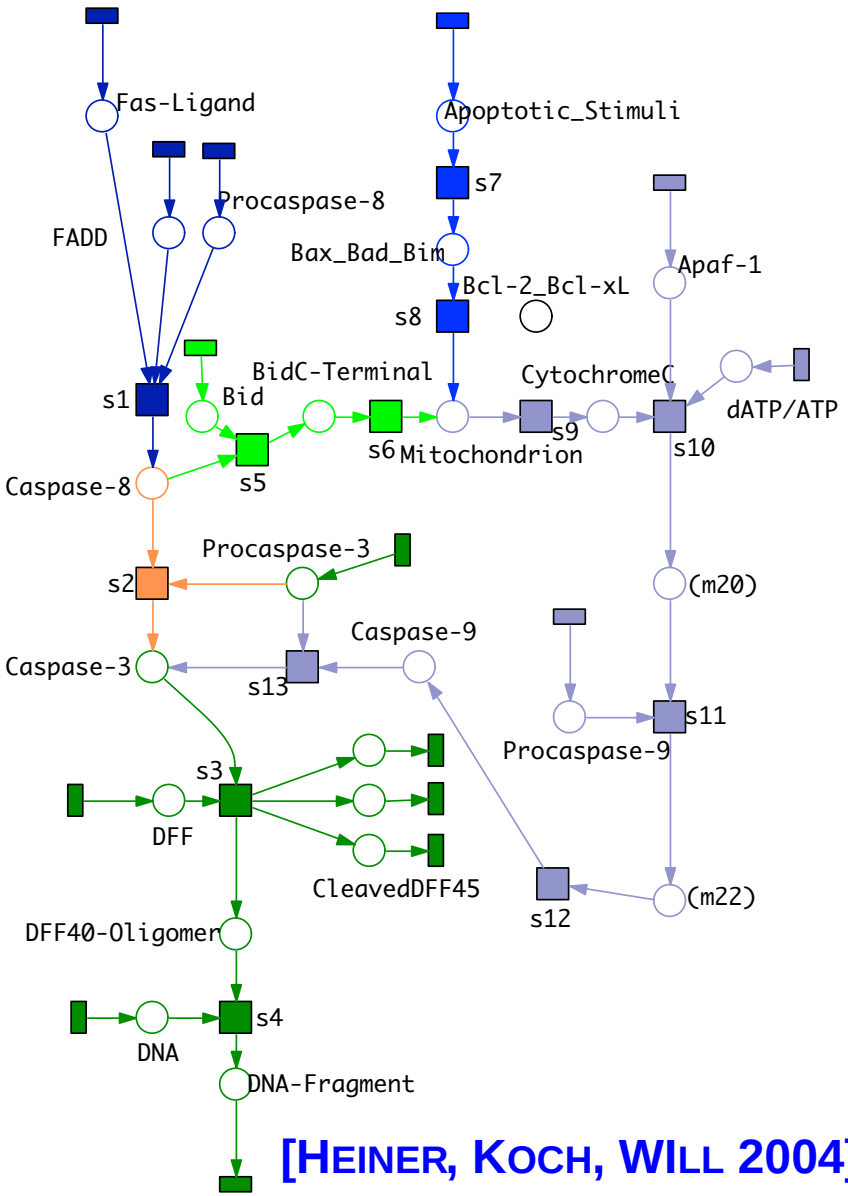
[Reddy 1993]
[Heiner 1998]



EX2 - APOPTOSIS IN MAMMALIAN CELLS

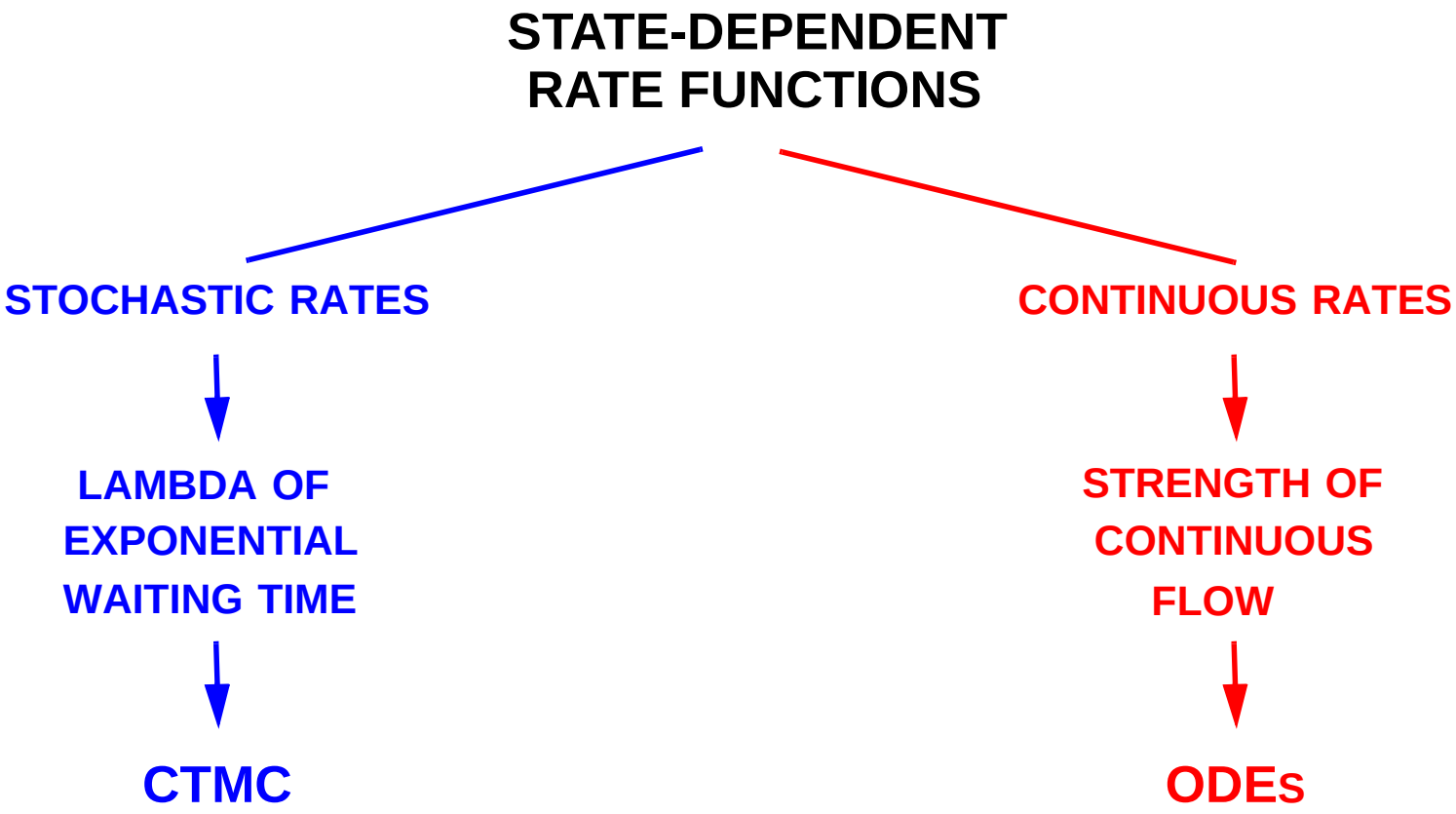


[GON 2003]



[HEINER, KOCH, WILL 2004]

STATE-DEPENDENT RATE FUNCTIONS



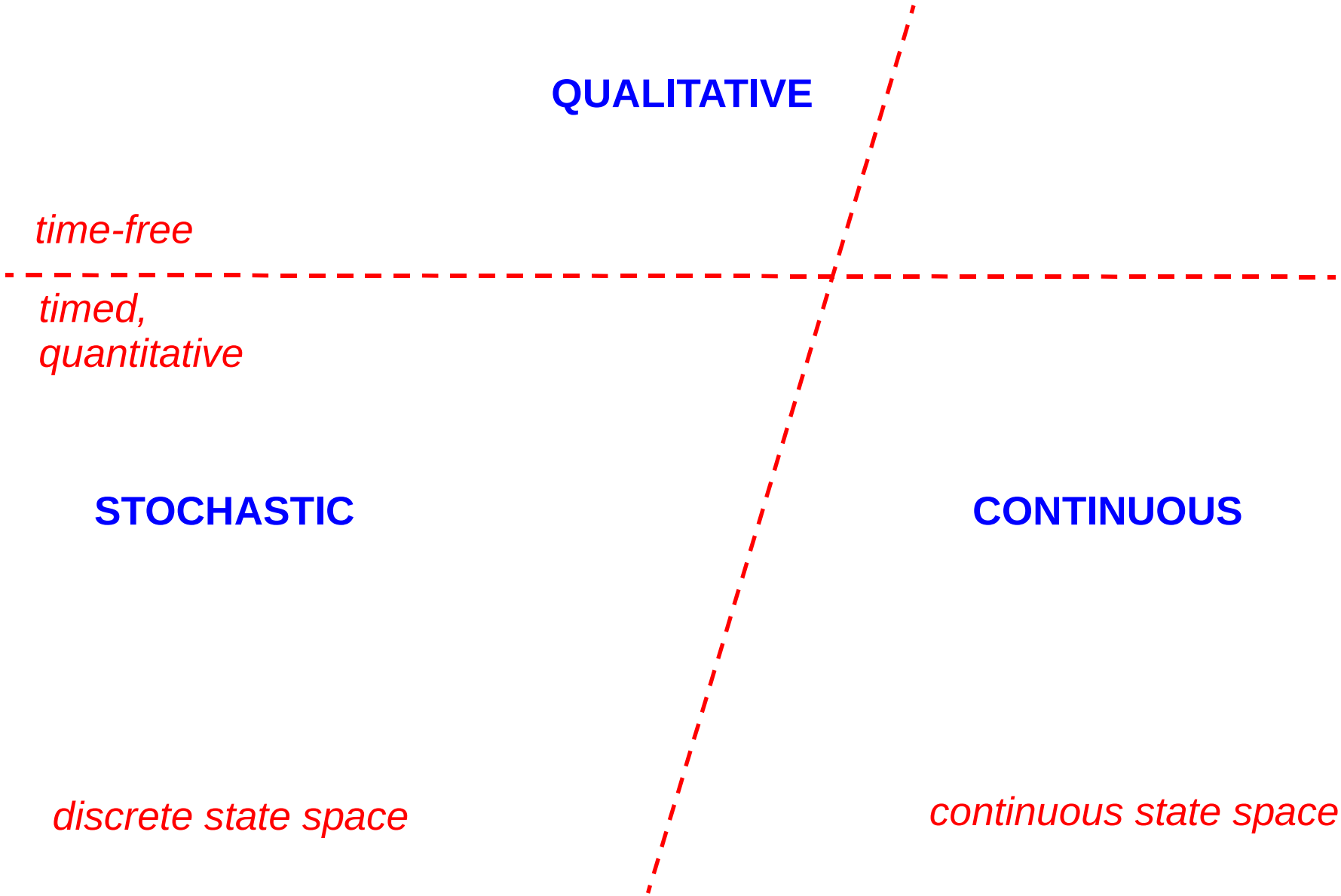
-> supported by, e.g., COPASI, Dizzy, ..., Snoopy

QUALITATIVE

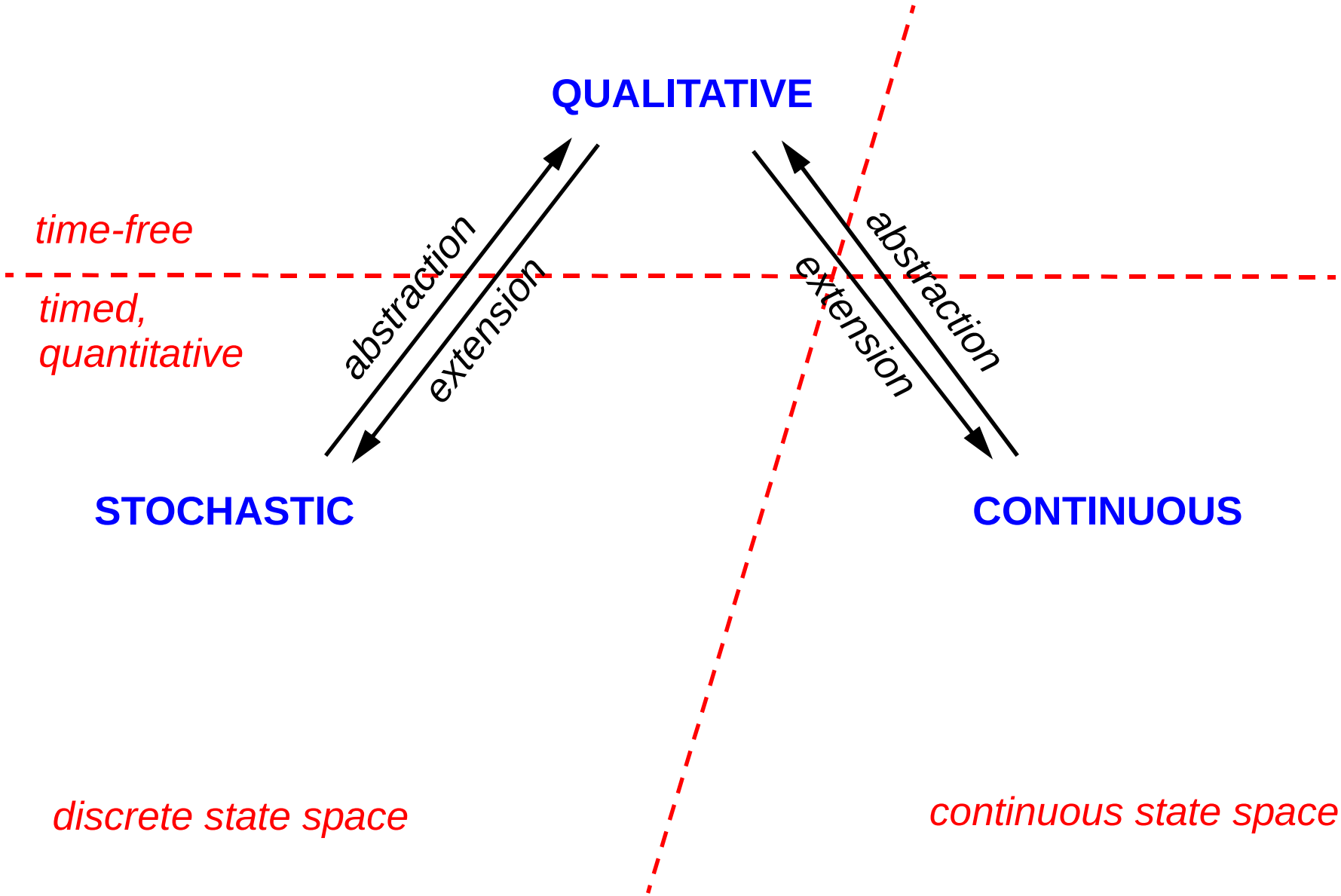
STOCHASTIC

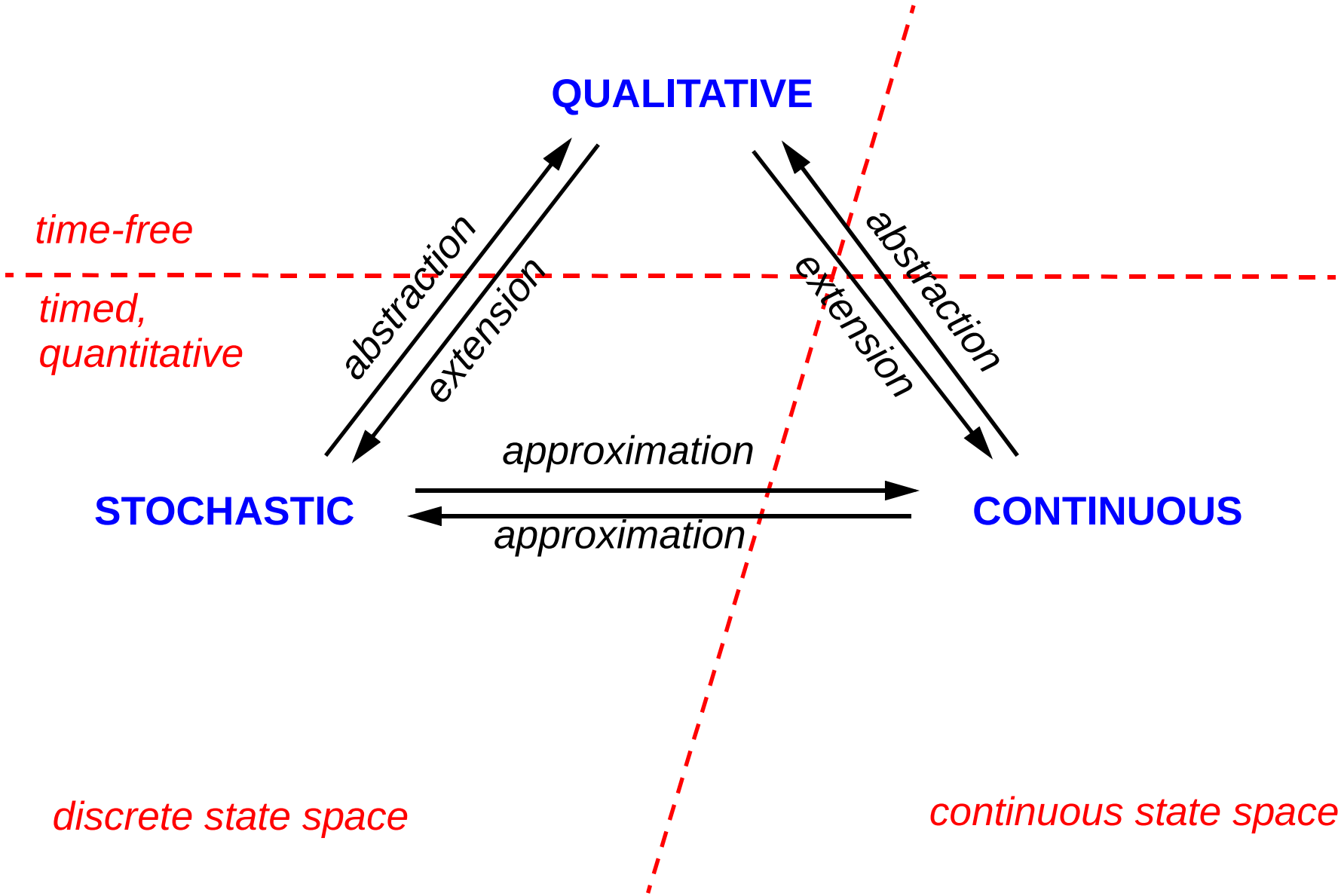
CONTINUOUS

FRAMEWORK 2007

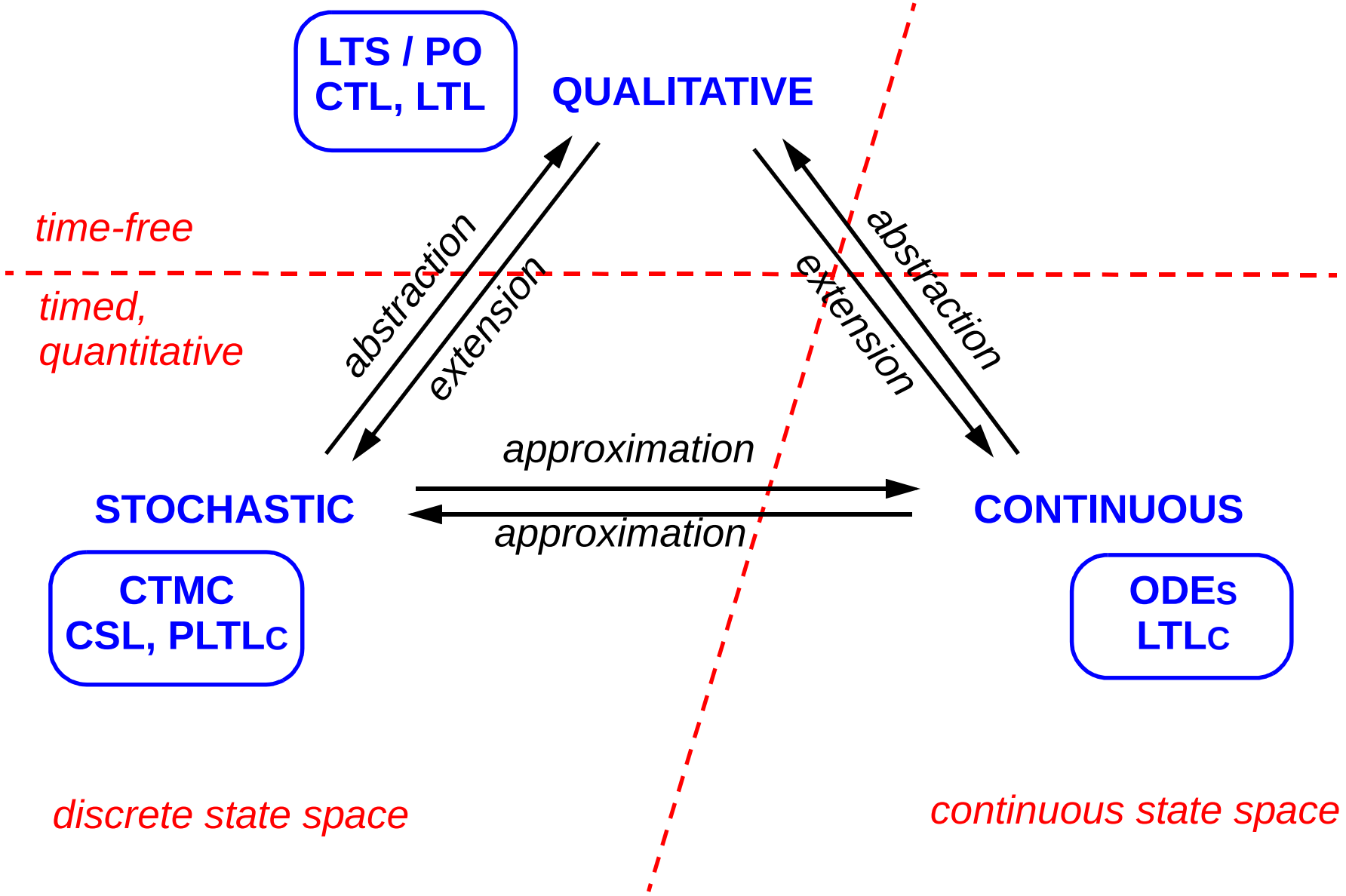


FRAMEWORK 2007

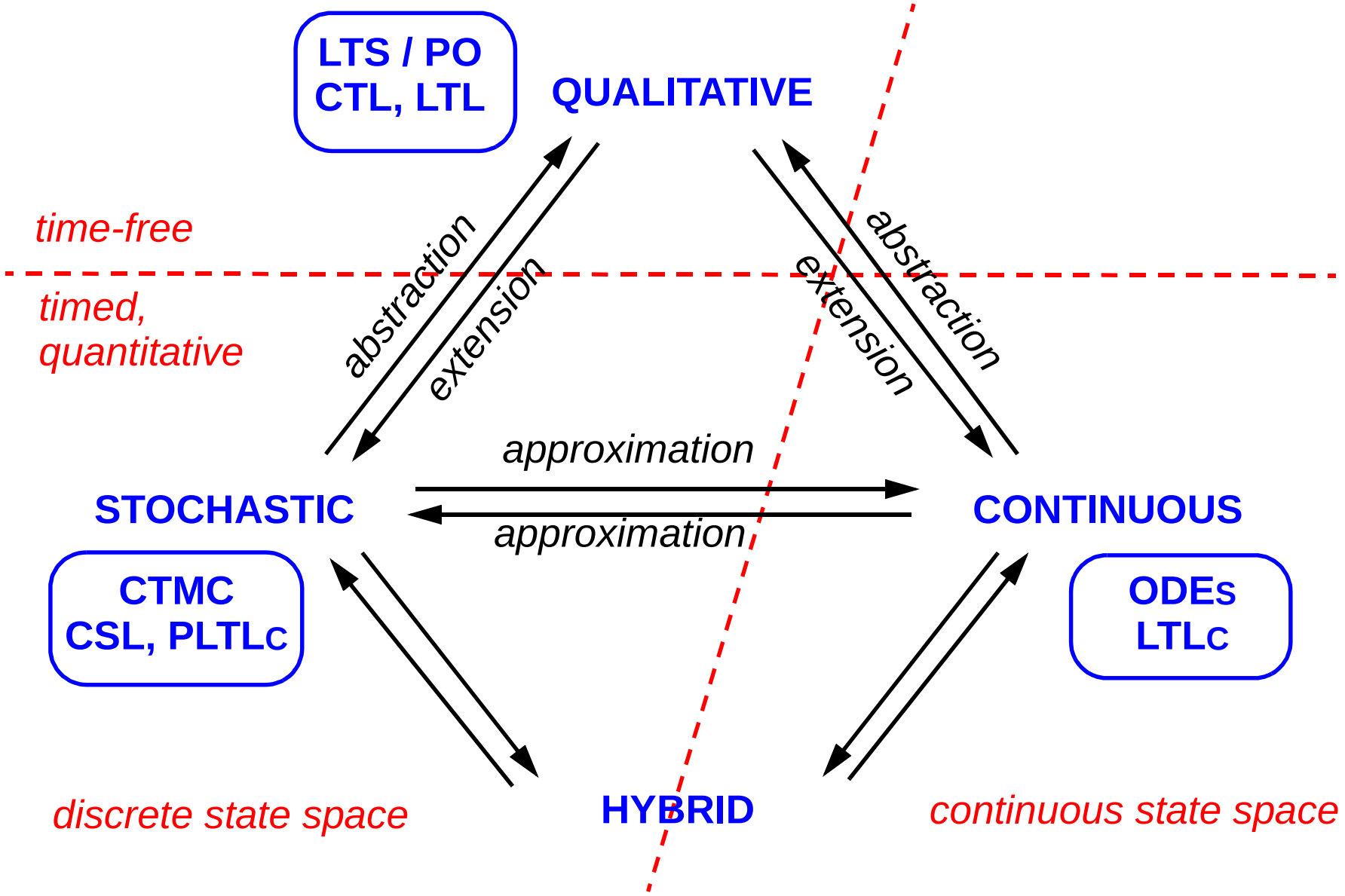



QUALITATIVE
time-free
timed,
quantitative
abstraction
extension
abstraction
extension
approximation
approximation
STOCHASTIC
CONTINUOUS
discrete state space
continuous state space

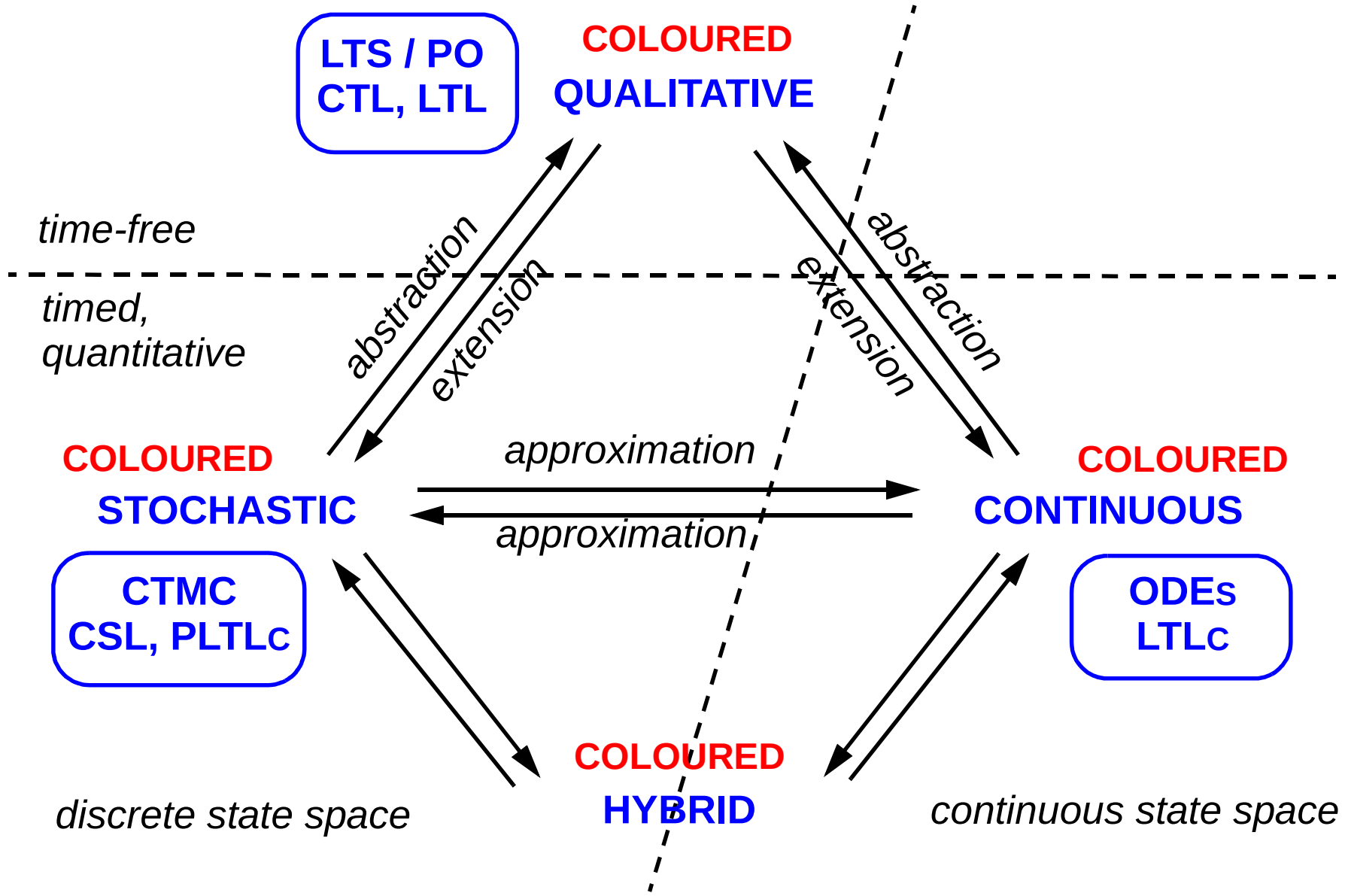
FRAMEWORK 2007



FRAMEWORK 2010

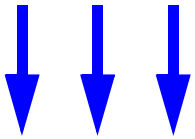


COLOURED FRAMEWORK 2011



4x2

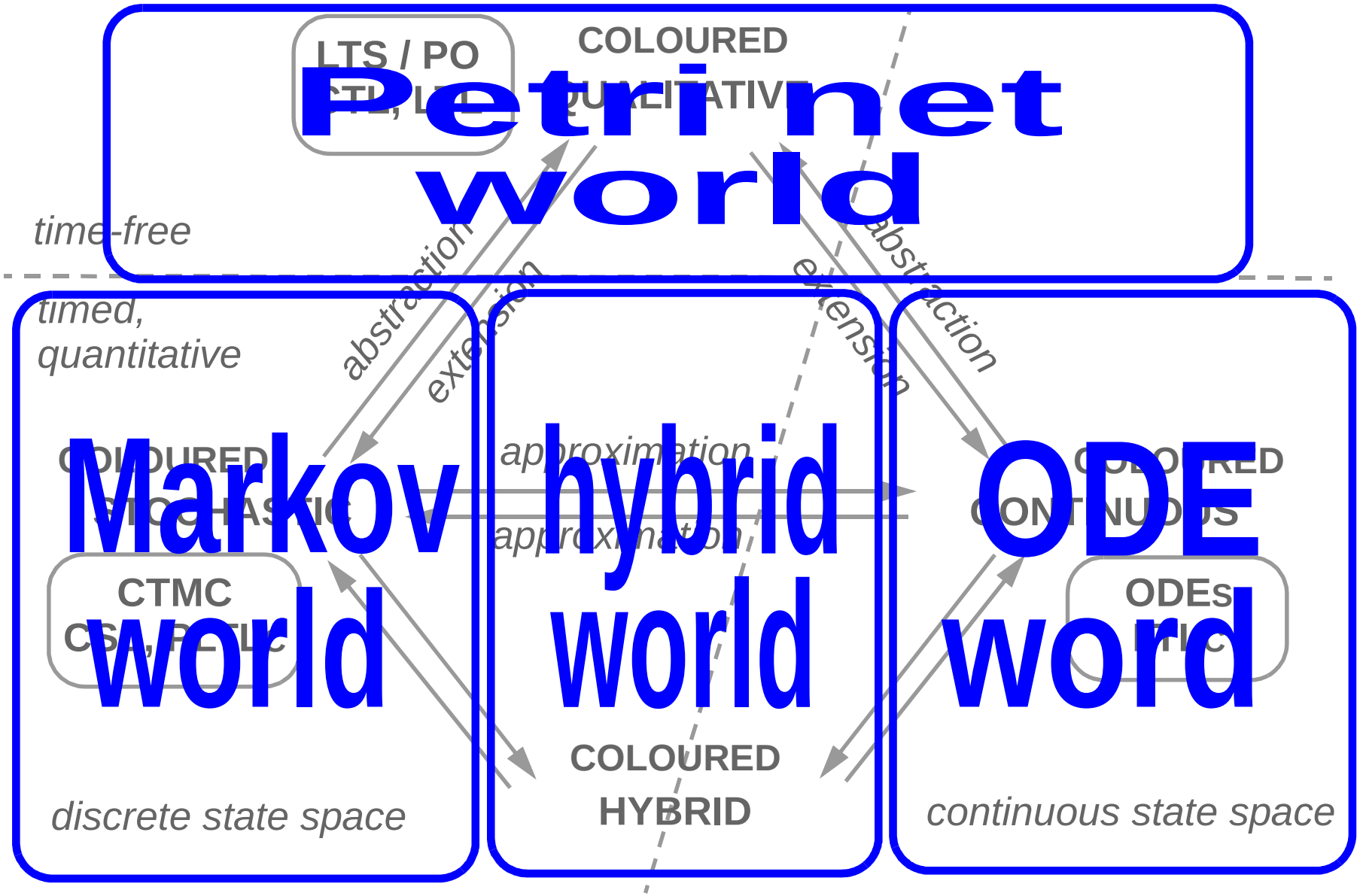
MODELS SHARING STRUCTURE



QUANTITATIVE MODEL = QUALITATIVE MODEL

+

RATE FUNCTIONS (KINETICS)


Petri net world
LTS / PO
COLOURED
time-free
timed, quantitative
abstraction
extension
approximation
approximation
extension
abstraction
Markov world
hybrid world
ODE world
CTMC
ODEs
COLOURED HYBRID
discrete state space
continuous state space

Our Toolbox

OUR TOOL BOX

- **SNOOPY**

  -> *modelling and animation/simulation of hierarchical graphs,*
  e.g. various Petri net classes, e.g. PN, XPN, SPN, XSPN, CPN, ...

- **S4**

  -> *standalone, computational steering server*

OUR TOOL BOX

- **SNOOPY**
  - -> *modelling and animation/simulation of hierarchical graphs,*
    e.g. various Petri net classes, e.g. PN, XPN, SPN, XSPN, CPN, ...
- **S4**
  - -> *standalone, computational steering server*
- **CHARLIE**
  - -> *PN, XPN, Time/Timed Petri nets (TPN)*
  - -> *mostly standard analysis techniques of Petri net theory*
- **MARCIE**
  - -> *PN, XPN, SPN, XSPN, SRN*
  - -> *symbolic and simulative model checking*

OUR TOOL BOX

- **SNOOPY**
  - -> *modelling and animation/simulation of hierarchical graphs,*
    e.g. various Petri net classes, e.g. PN, XPN, SPN, XSPN, CPN, ...
- **S4**
  - -> *standalone, computational steering server*
- **CHARLIE**
  - -> *PN, XPN, Time/Timed Petri nets (TPN)*
  - -> *mostly standard analysis techniques of Petri net theory*
- **MARCIE**
  - -> *PN, XPN, SPN, XSPN, SRN*
  - -> *symbolic and simulative model checking*
- **Patty**
  - -> *animation via web browser*

OUR TOOL BOX

- **SNOOPY**
  - -> *modelling and animation/simulation of hierarchical graphs,*
    e.g. various Petri net classes, ... XPN, SPN, XSPN, CPN, ...
- **S4**
  - -> *standalone, computational steering server*
- **CHARLIE**
  - -> *PN, XPN, Time/Timed Petri nets (TPN)*
  - -> *mostly standard analysis techniques of Petri net theory*
- **MARCIE**
  - -> *PN, XPN, SPN, XSPN, SRN*
  - -> *symbolic and simulative model checking*
- **Patty**
  - -> *animation via web browser*

SBML import/export

EXPORT TO MATAB AND

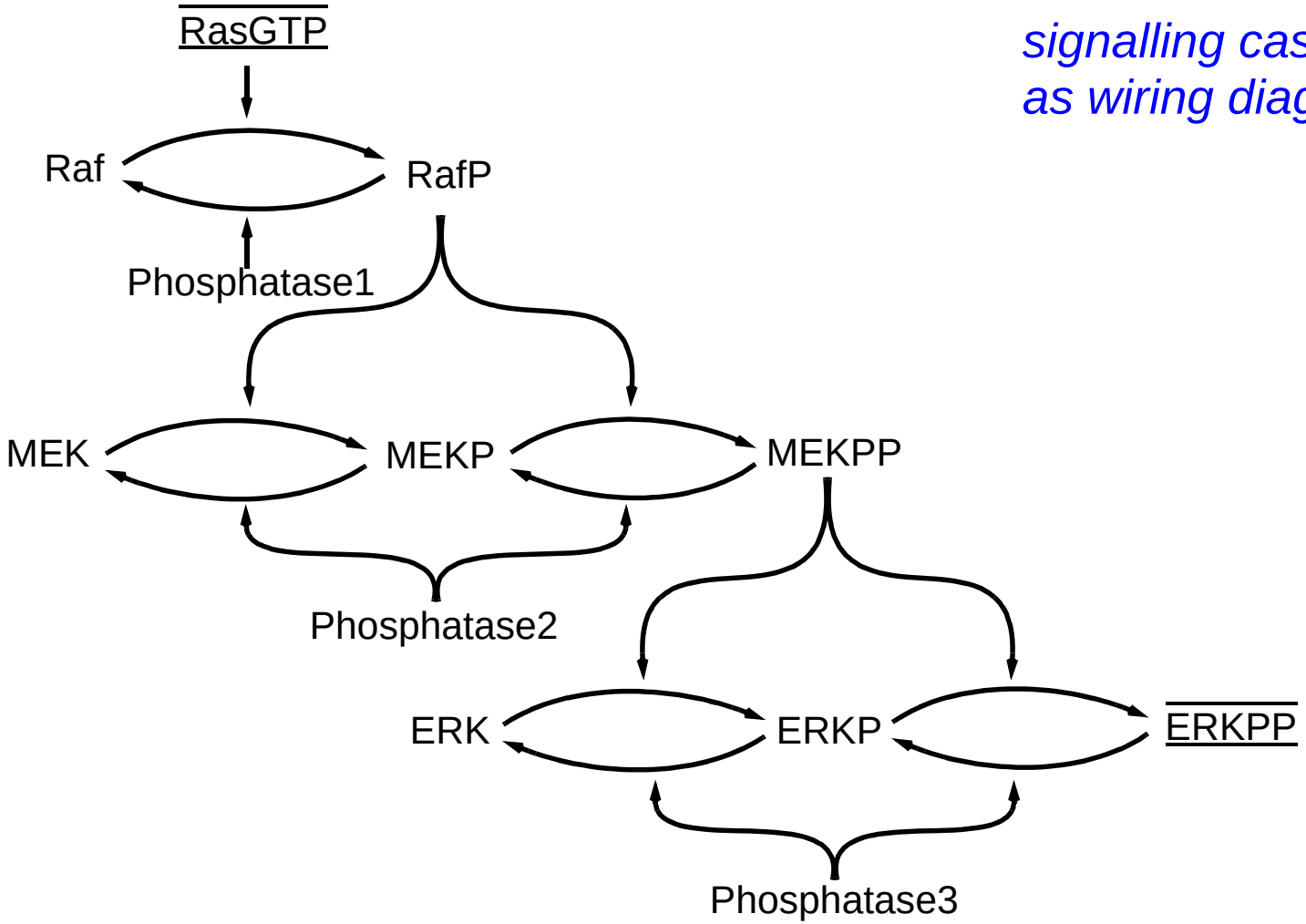
MANY OTHER TOOLS

MODELLING BIO (PETRI) NETS

-

THREE APPROACHES

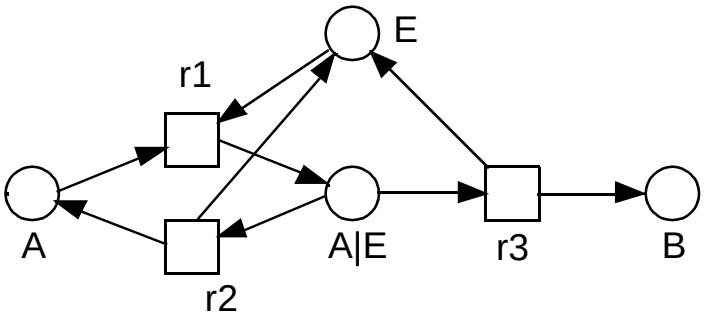
APPROACH 1



signalling cascade as wiring diagram

BASIC STRUCTURE

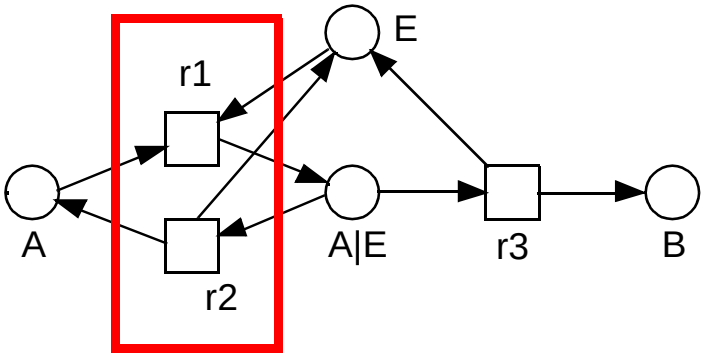
A <E--> A|E --> B



enzymatic reaction,
mass-action kinetics

BASIC STRUCTURE

A <E--> A|E --> B

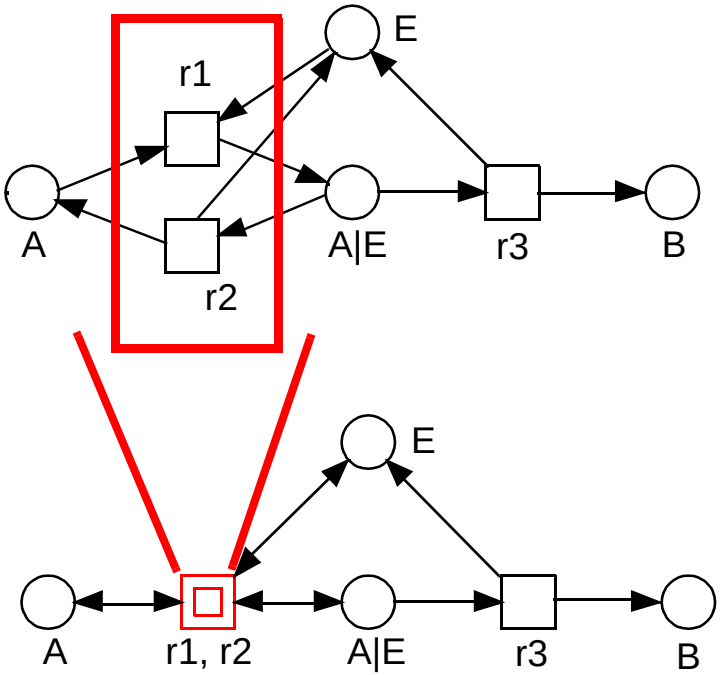


enzymatic reaction,
mass-action kinetics

BASIC STRUCTURE

A <E--> A|E --> B

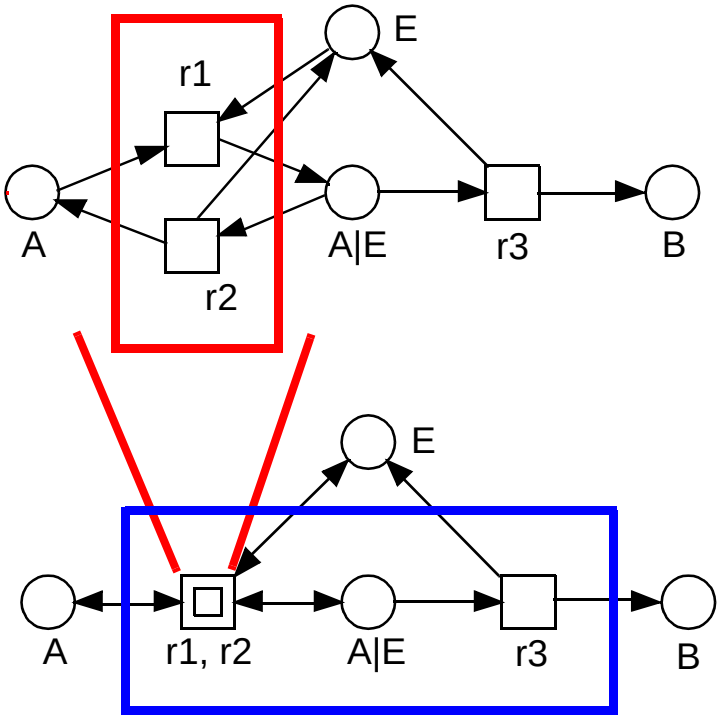
enzymatic reaction,
mass-action kinetics



BASIC STRUCTURE

$$A \xleftrightarrow{E} A|E \rightarrow B$$

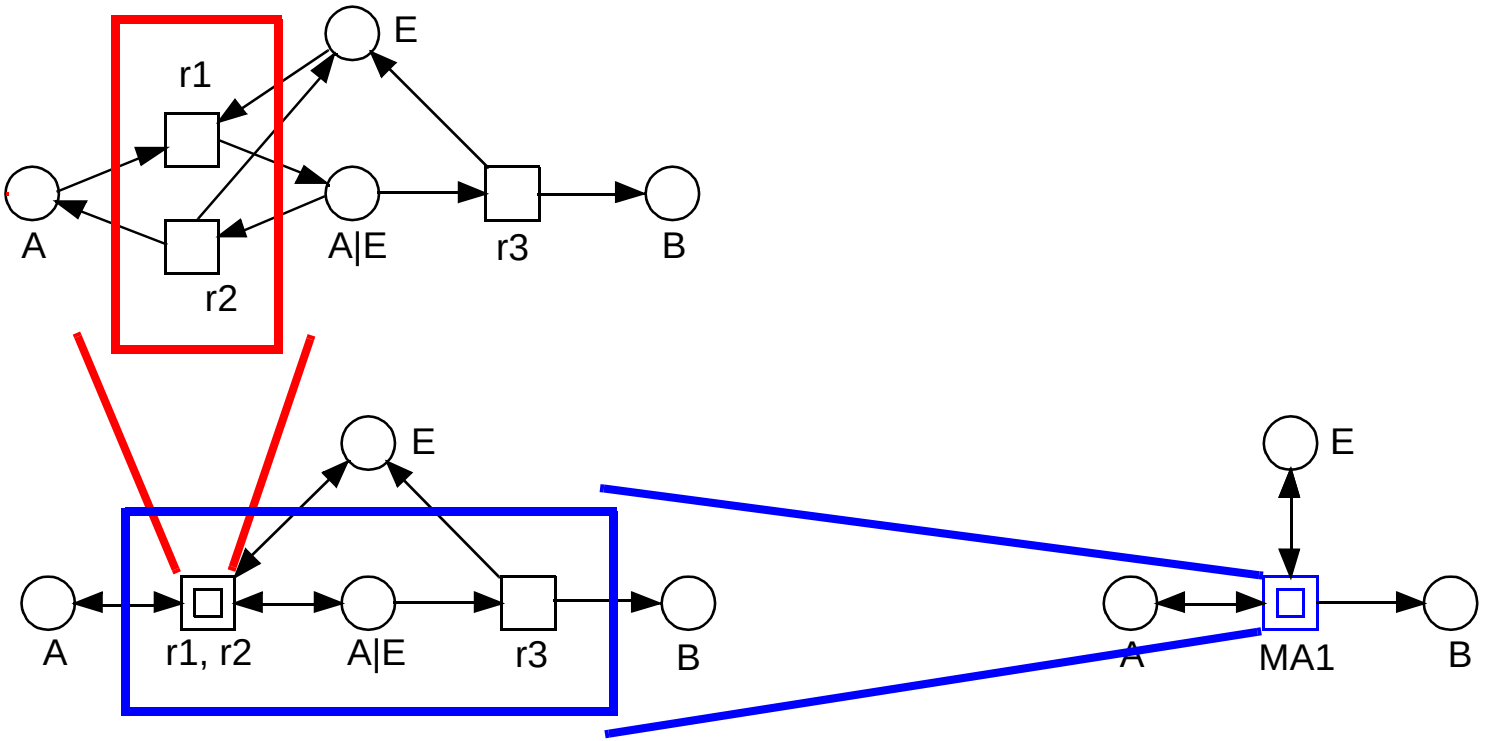
enzymatic reaction,
mass-action kinetics



BASIC STRUCTURE

A <E--> A|E --> B

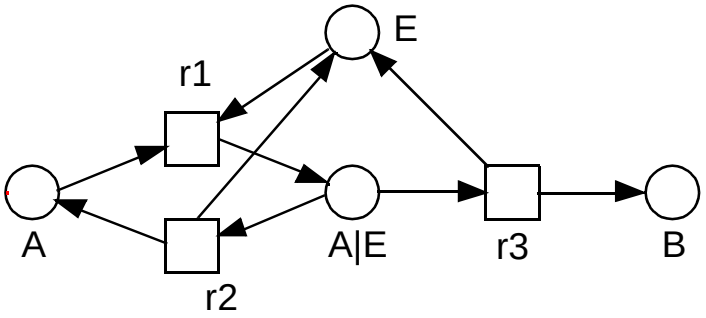
enzymatic reaction,
mass-action kinetics

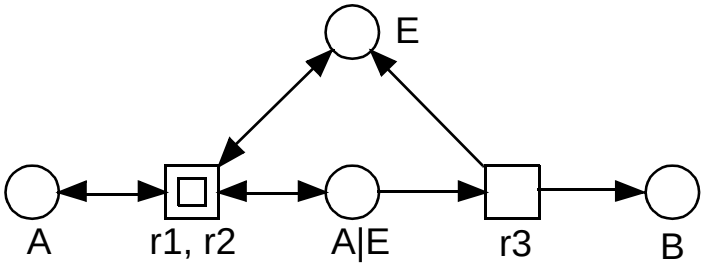


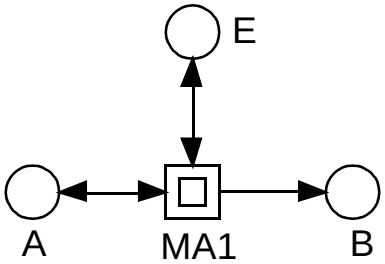
BASIC STRUCTURE

$$A \xleftarrow{E} A|E \rightarrow B$$

enzymatic reaction,
mass-action kinetics

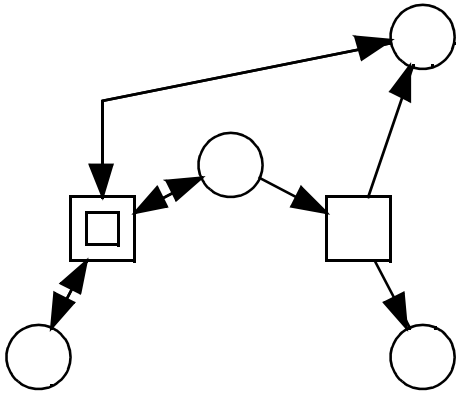




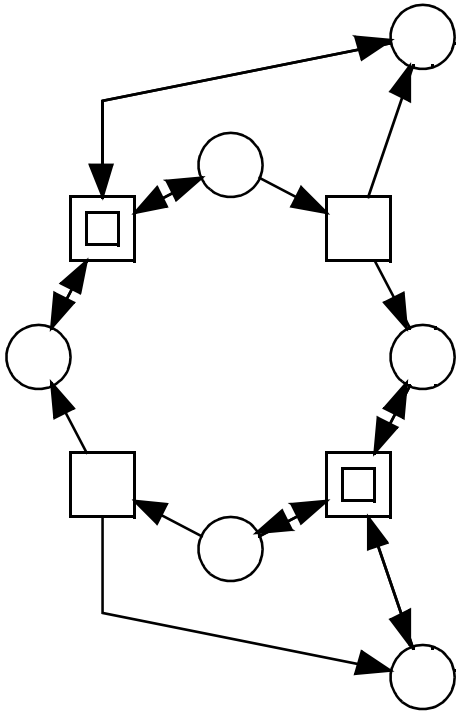


NET COMPOSITION FROM BUILDING BLOCKS

SINGLE
MASS-ACTION STEP

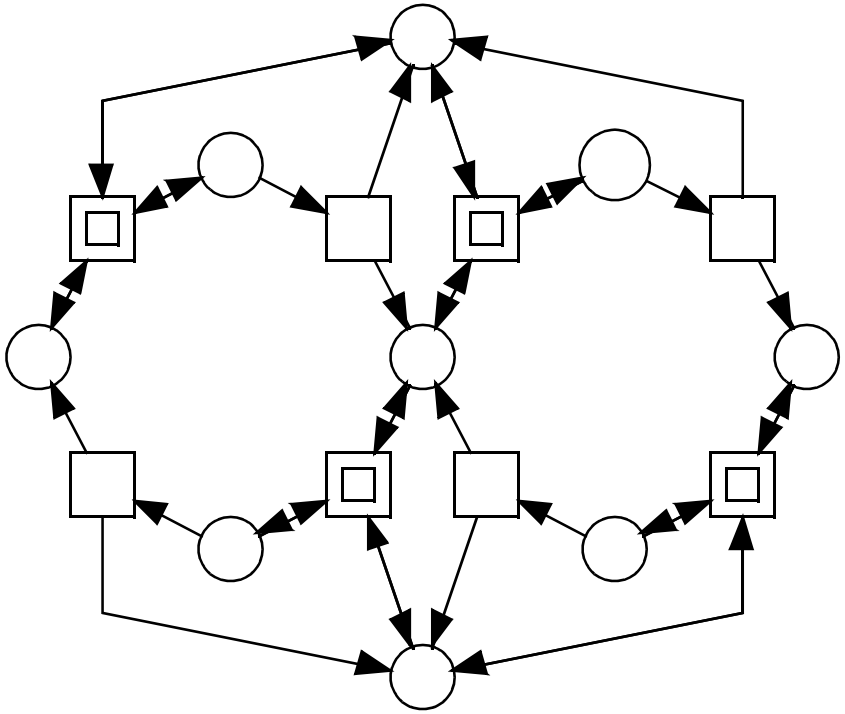


NET COMPOSITION FROM BUILDING BLOCKS

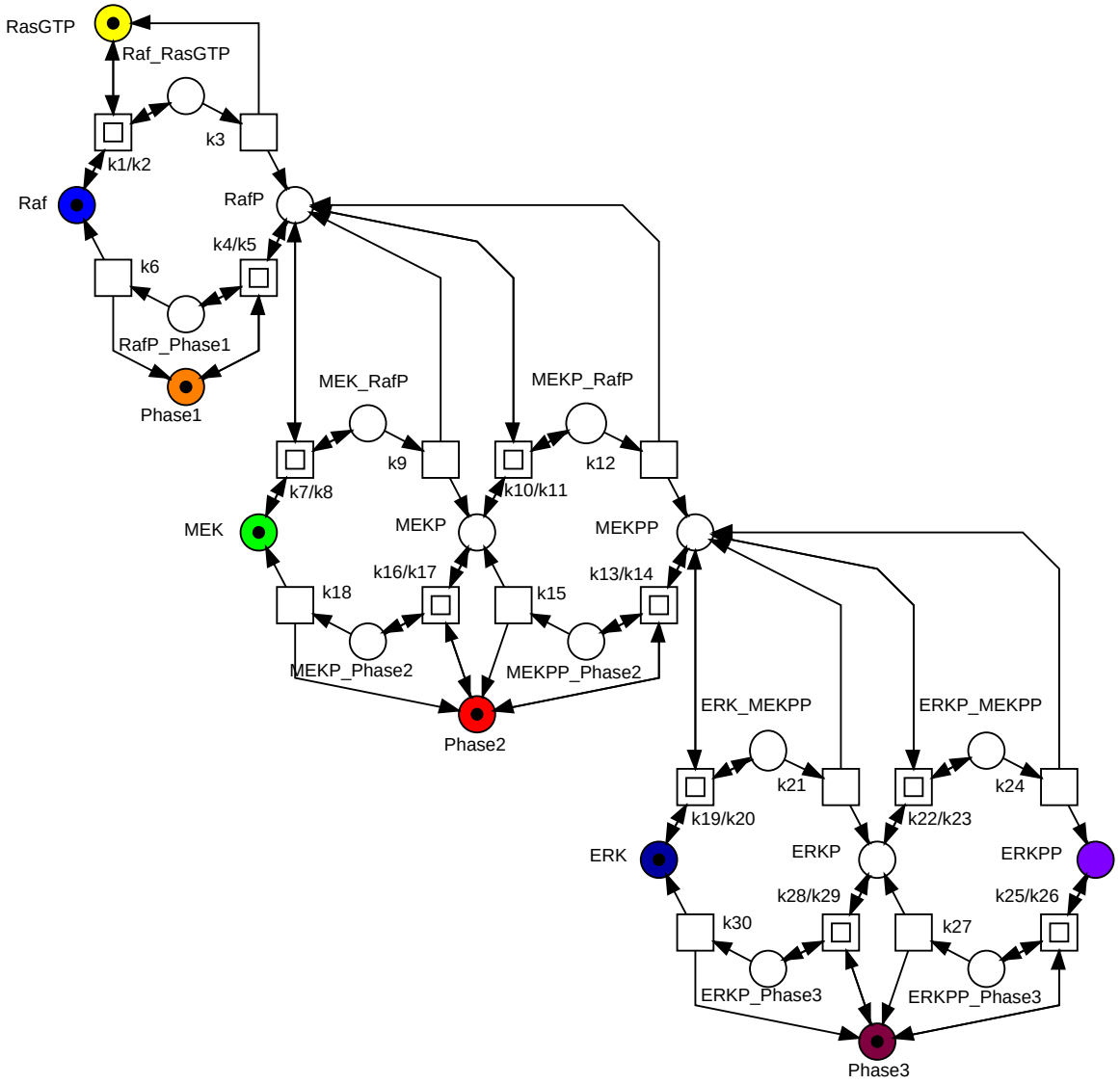


SINGLE
PHOSPHORYLATION / DEPHOSPHORYLATION

DOUBLE PHOSPHORYLATION / DEPHOSPHORYLATION



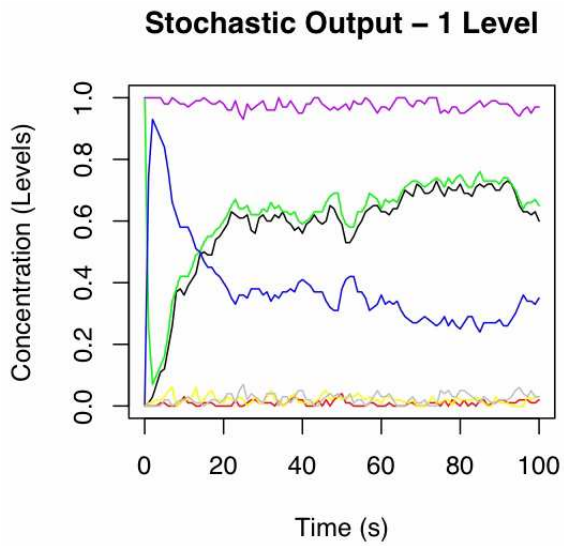
SIGNALLING CASCADE AS PETRI NET

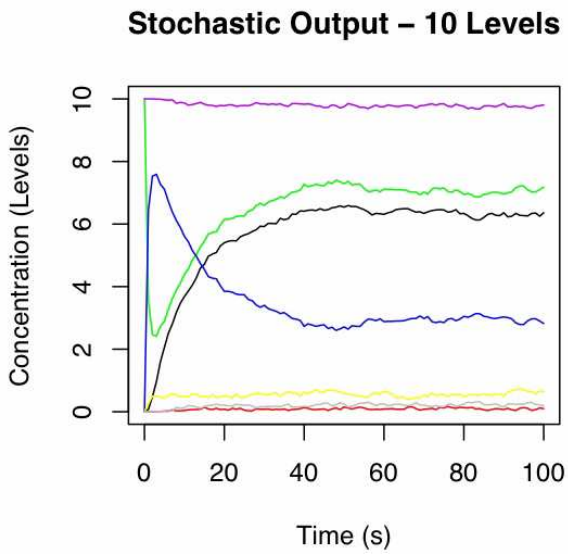


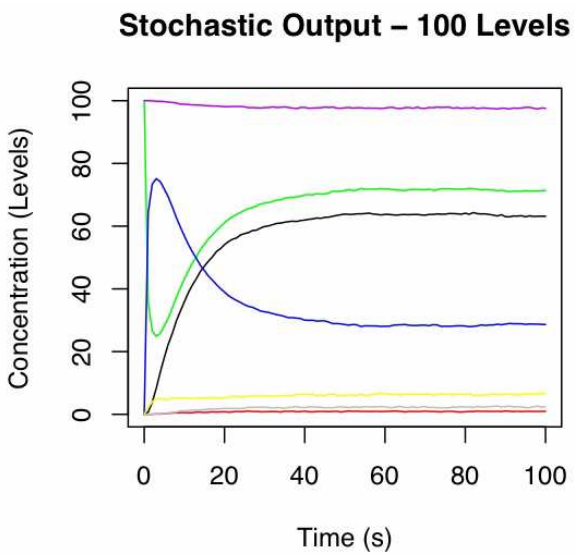
[GILBERT, HEINER, LEHRACK 2007]

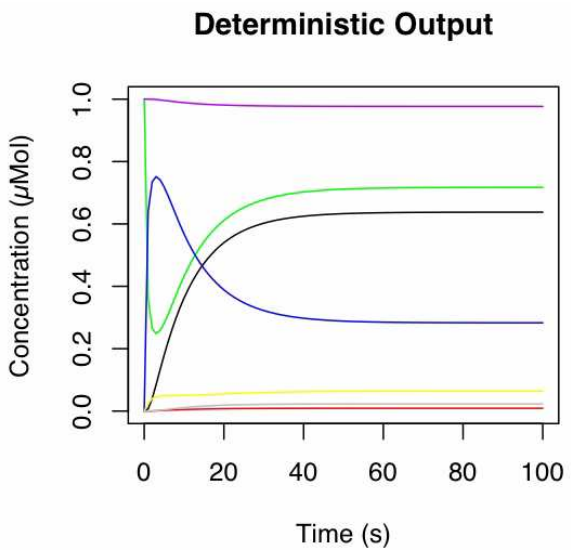
[HEINER, GILBERT, DONALDSON 2008]

SIGNALLING CASCADE, SIMULATIONS









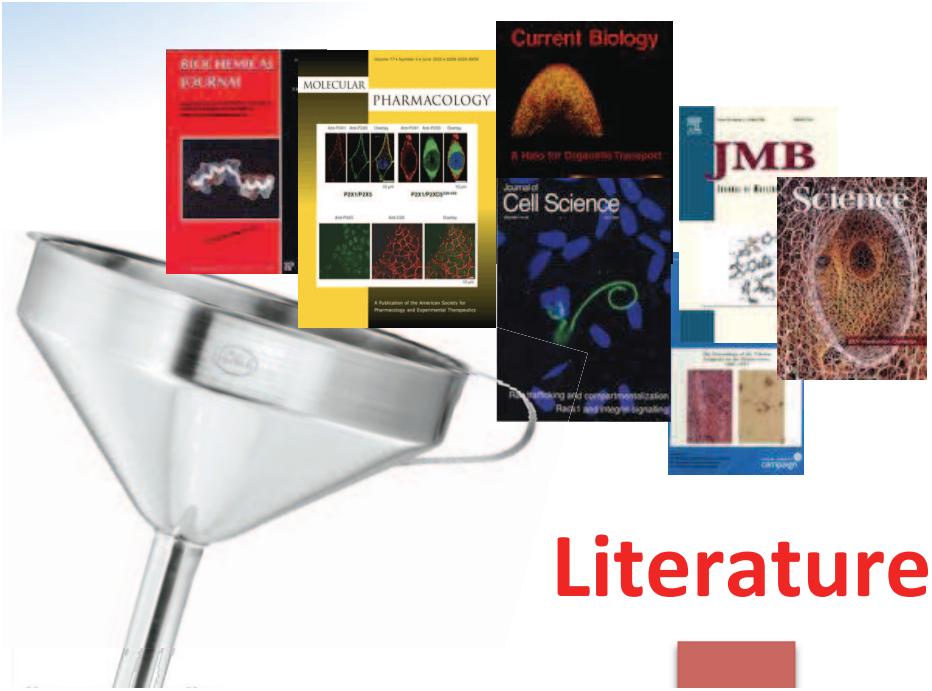
signalling cascade as wiring diagram

[GILBERT, HEINER, LEHRACK 2007]

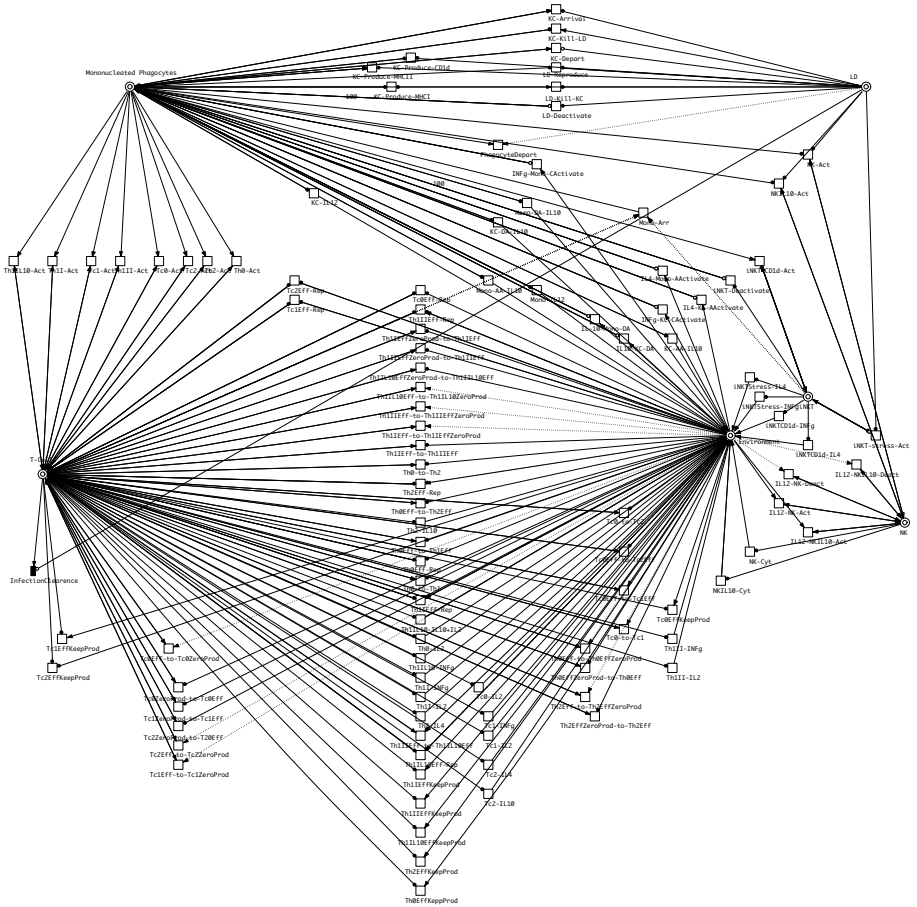
[HEINER, GILBERT, DONALDSON 2008]

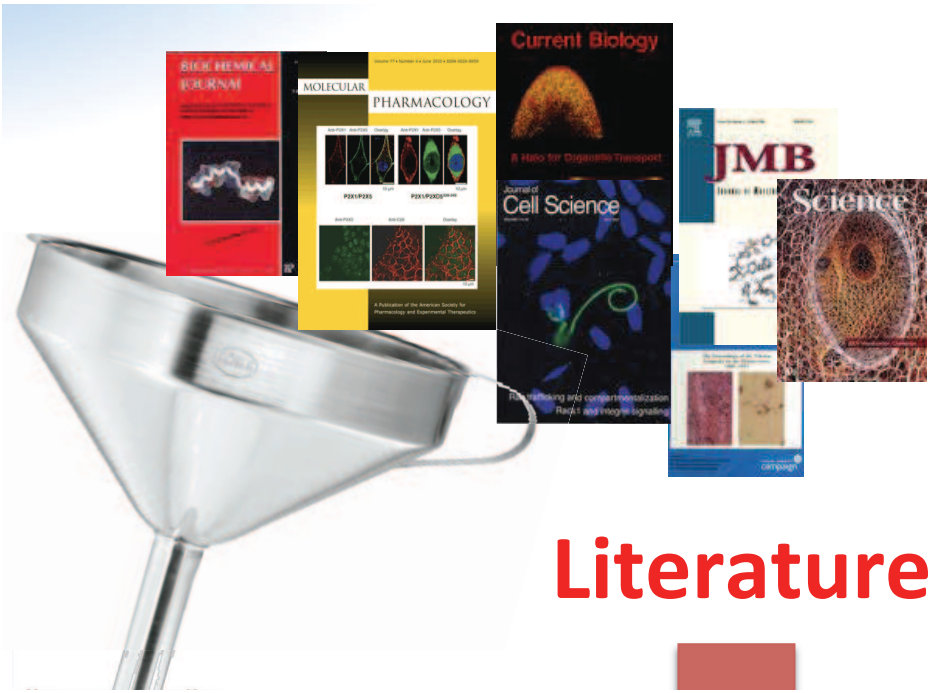
APPROACH 2

FROM LITERATURE TO MODELS



FROM LITERATURE TO MODELS



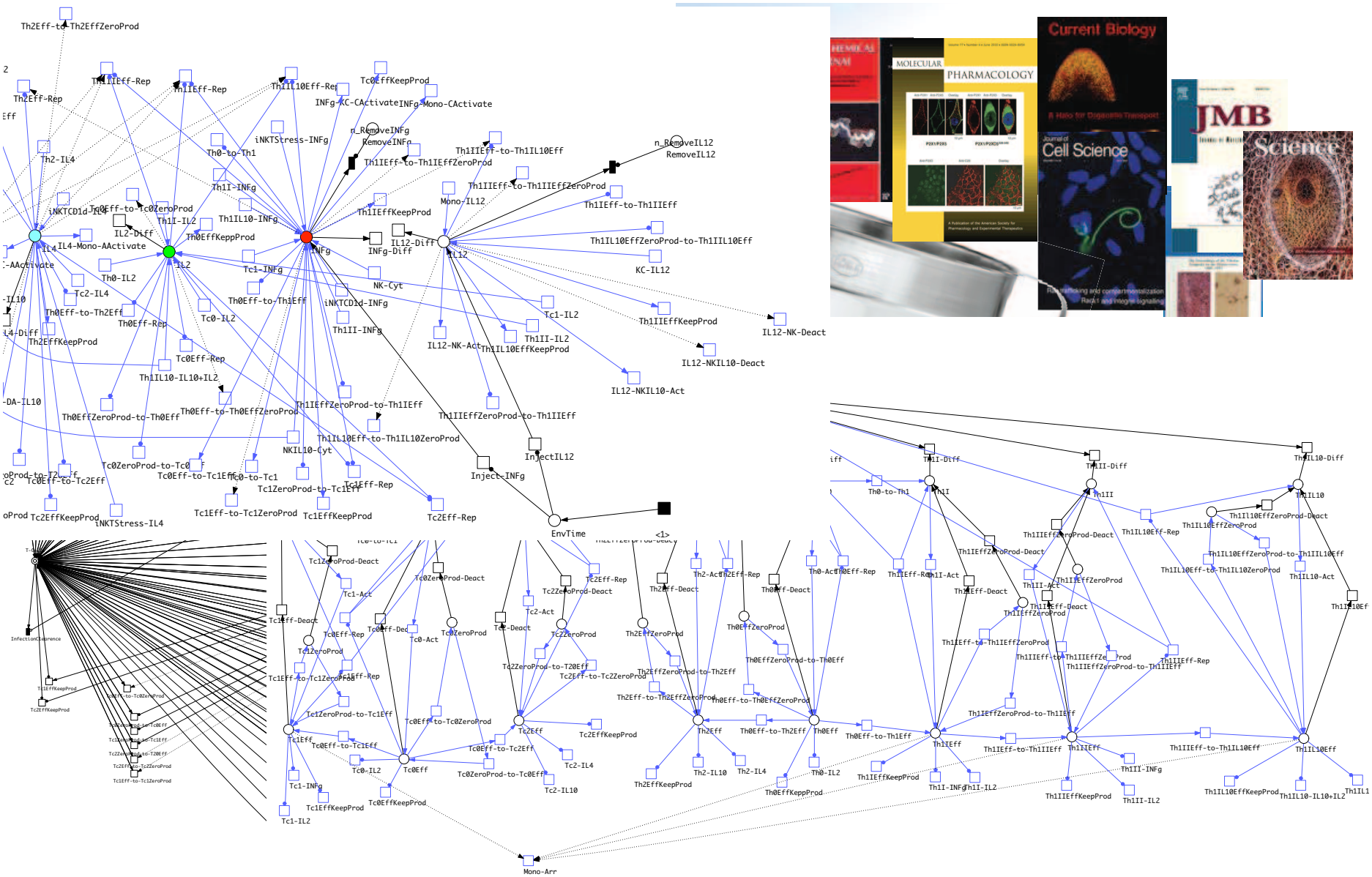


Literature

FROM LITERATURE TO MODELS

monika.heiner@b-tu.de

FROM LITERATURE TO MODELS

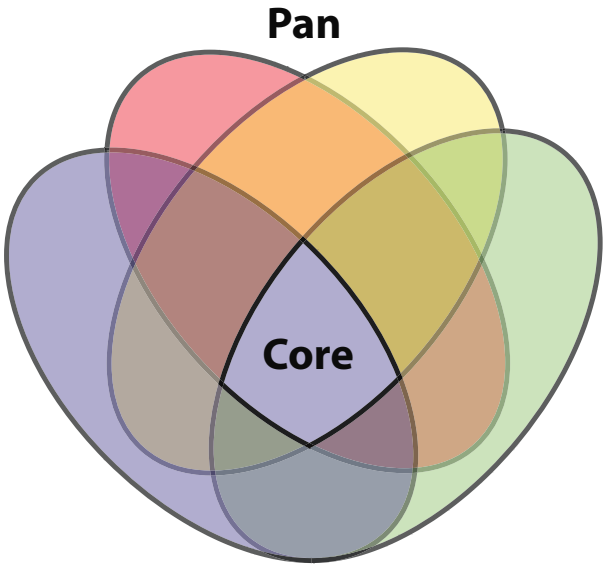


APPROACH 3

FROM GENOMES TO MODELS

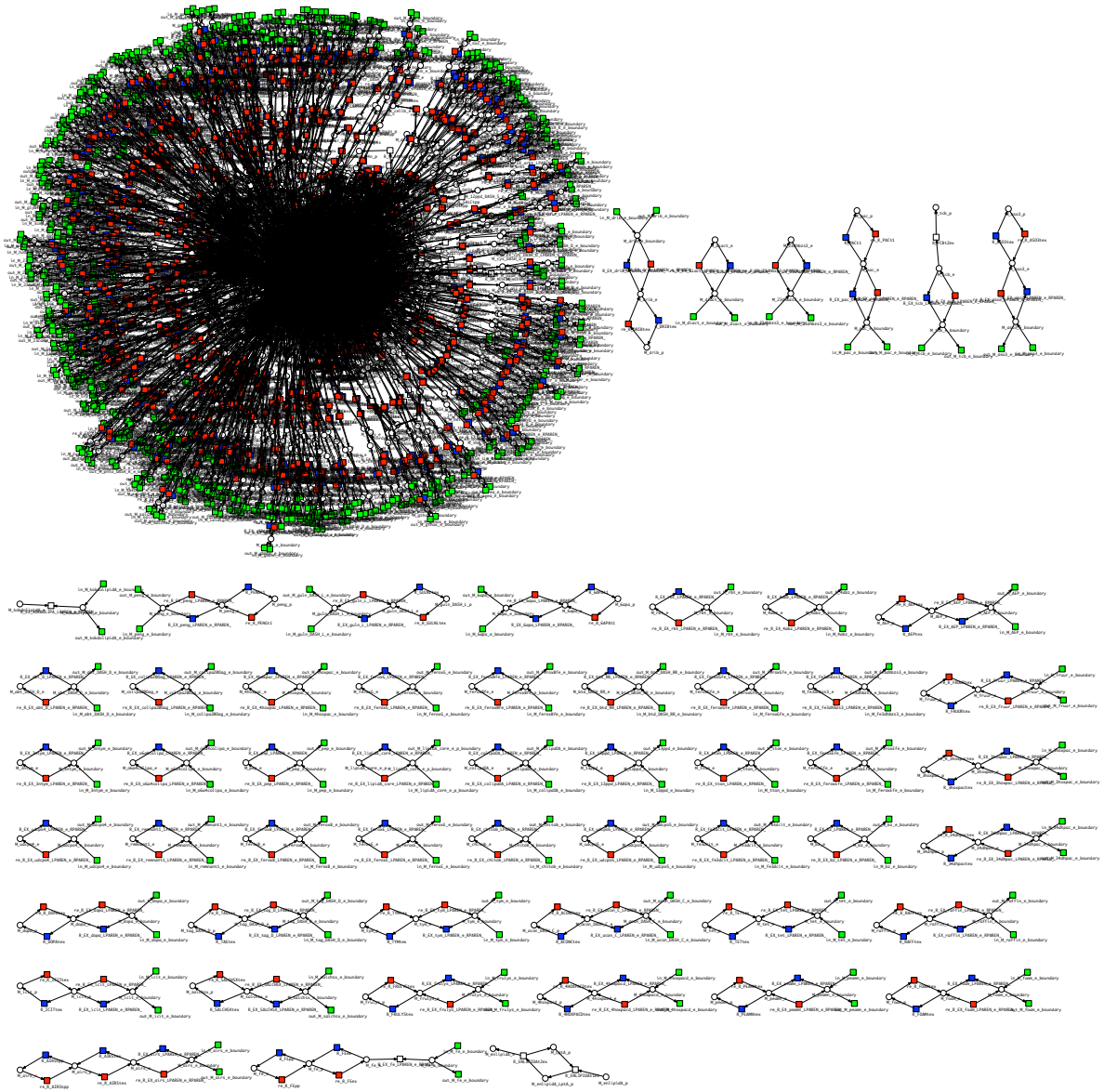
- **synthetic systems biology**
  - -> *knowledge-based design*
  - -> *creation of new living systems (in our case: bacteria)*
  - -> *with novel exploitable/translational applications*

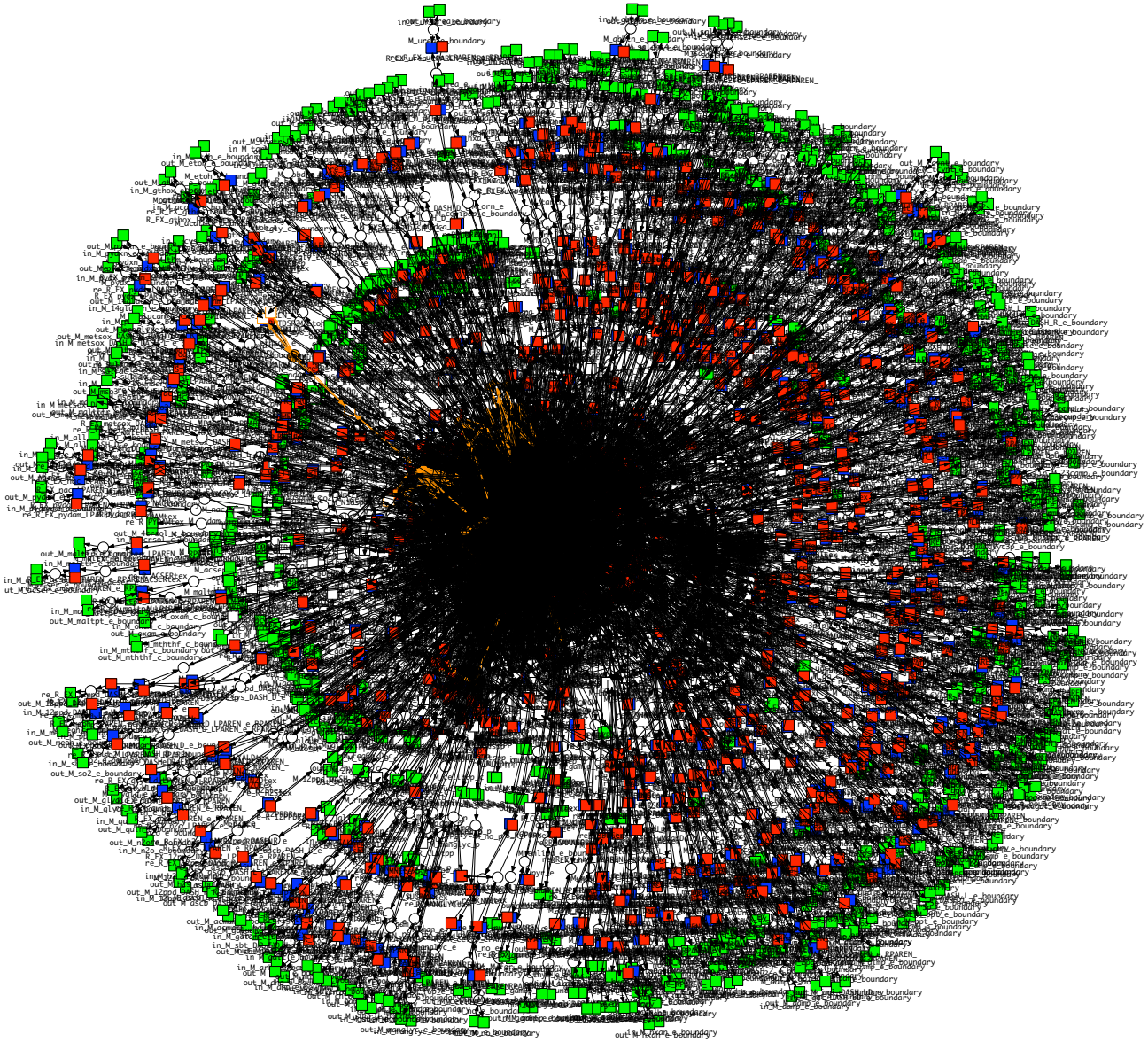
- **engineer bacteria to act as little factories for,, e.g.,**
  - -> *energy / drug production*
  - -> *environmental sensors*

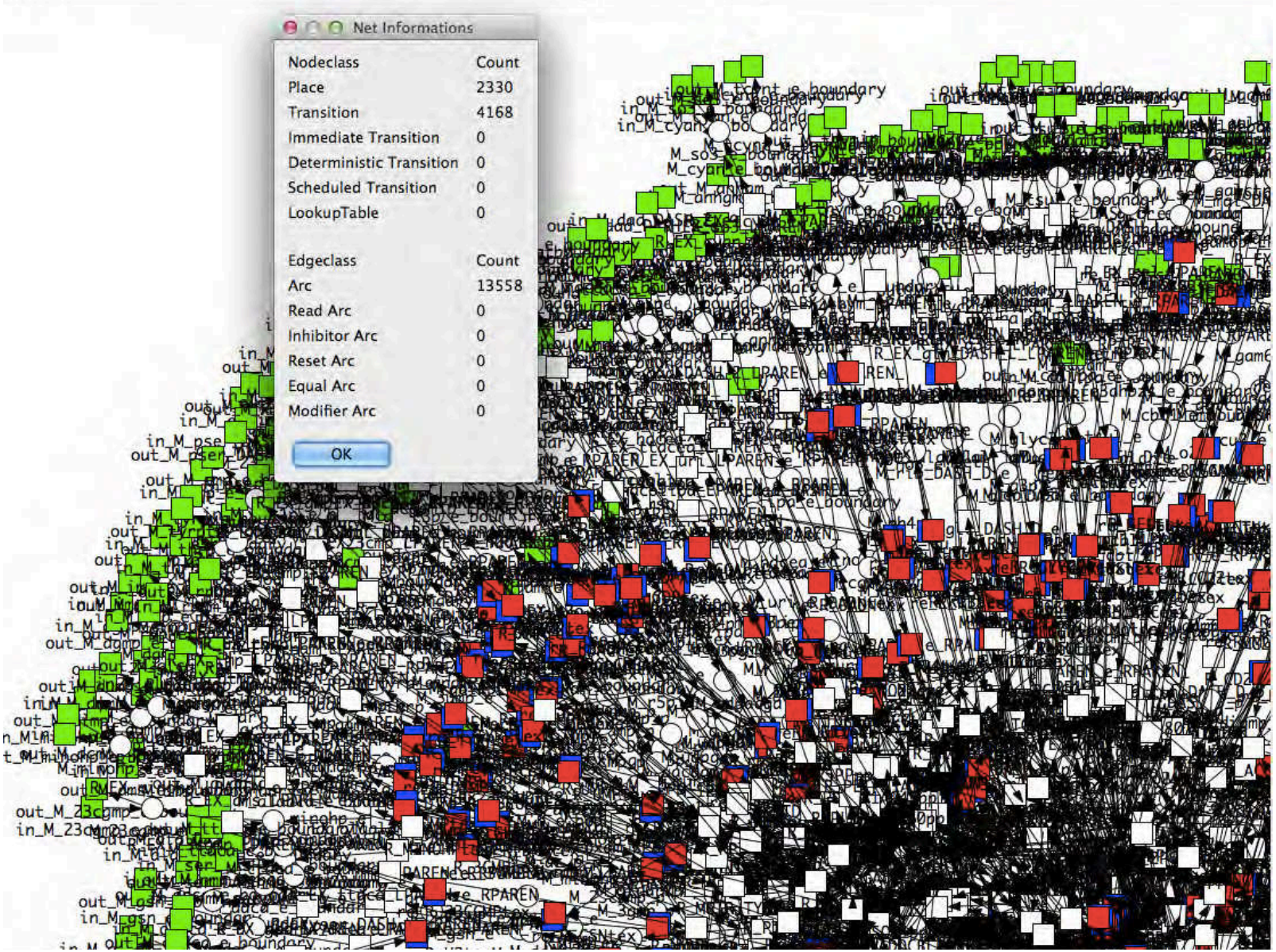


- **model organisms (e.g., E. coli)**
  - -> *several hundred ‚complete' genomic sequences*
  - -> *each individual genome: 4,000 - 5,500 genes,*

- **generating strain-specific (metabolic) models**
  - -> *computational metabolic models*
  - -> *estimated model size:*
    1,800 (reversible) reactions, 1,700 (compartemental) metabolites [Monk 2013]






Net Informations
Nodeclass Count
Place 2330
Transition 4168
Immediate Transition 0
Deterministic Transition 0
Scheduled Transition 0
LookupTable 0
Edgeclass Count
Arc 13558
Read Arc 0
Inhibitor Arc 0
Reset Arc 0
Equal Arc 0
Modifier Arc 0
OK

HOW TO SURVIVE LARGER MODELS ?

HOW TO SURVIVE LARGER MODELS ?

-> MODULAR MODELLING

HOW TO SURVIVE LARGER MODELS ?

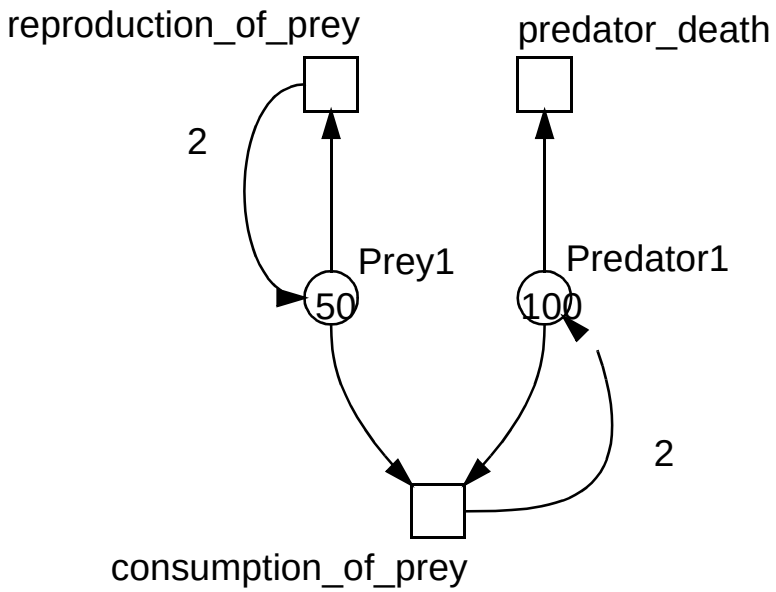
-> MODULAR MODELLING

-> COLOURED PETRI NETS

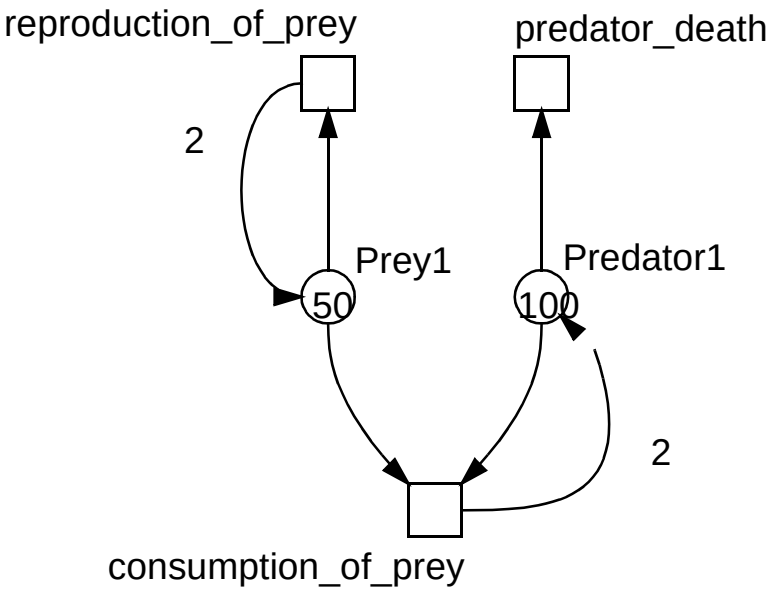
. . . AND THEN THERE WAS COLOUR

Kew Gardens, 24/04/2011

EX: PREY - PREDATOR

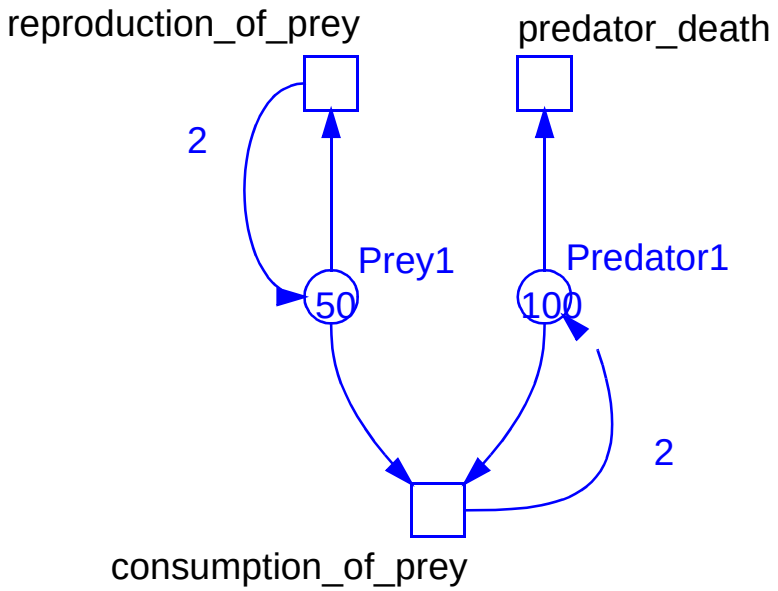


EX: PREY - PREDATOR

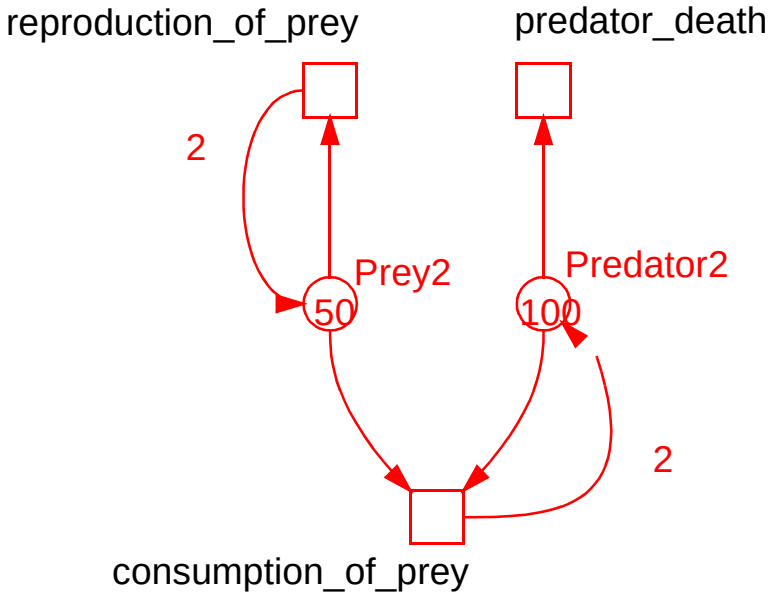


EX: PREY - PREDATOR

sub-system1



sub-system2



EX: PREY - PREDATOR

- **definitions**

  colourset *CS = 1-2;*

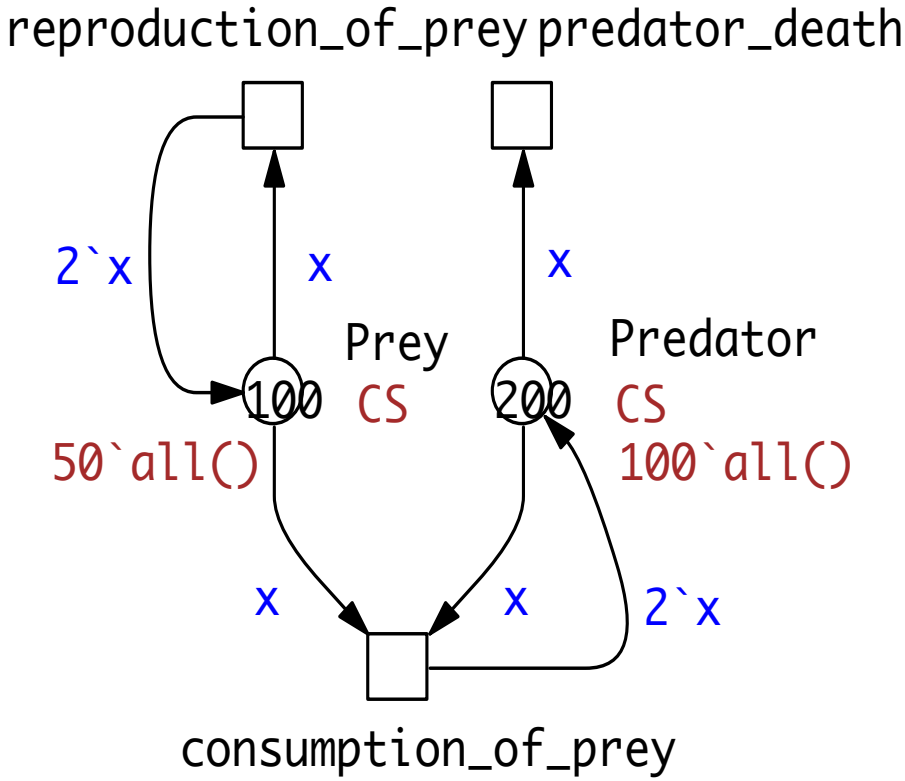
  var *x : CS;*

- **better:**

  const *SIZE = 2;*

  colourset *CS = 1-SIZE;*

  var *x : CS;*



EX: PREY - PREDATOR

- **definitions**

  colourset *CS = 1-2;*

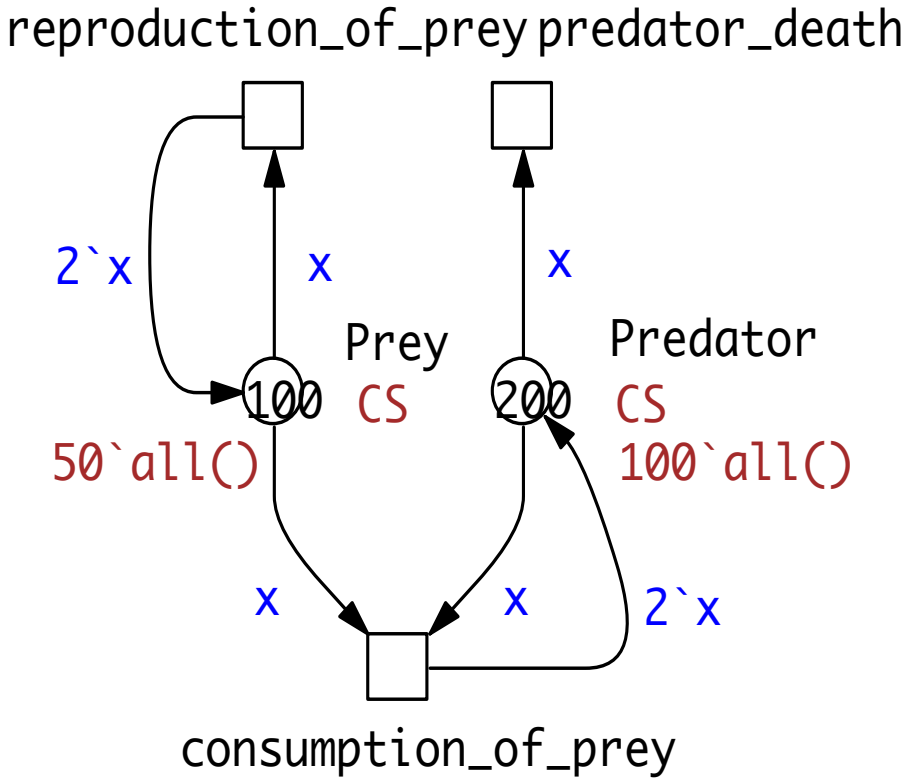
  var *x : CS;*

- **better:**

  const *SIZE = 2;*

  colourset *CS = 1-SIZE;*

  var *x : CS;*



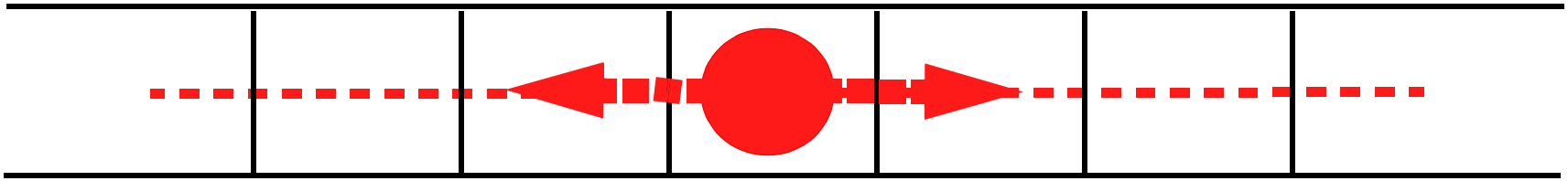
- **changing SIZE adapts the model to various scenarious**

EXAMPLE 1:

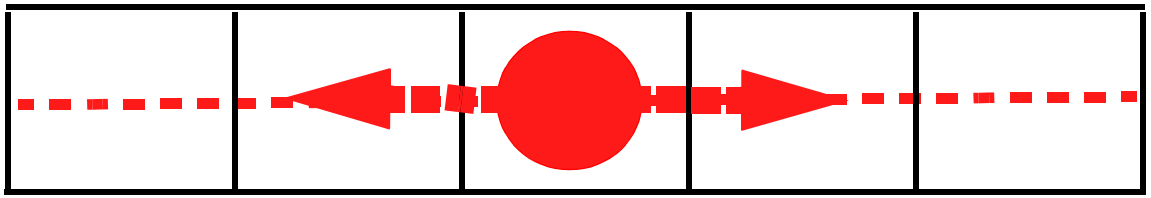
DIFFUSION IN SPACE

Richmond, 13/09/2011

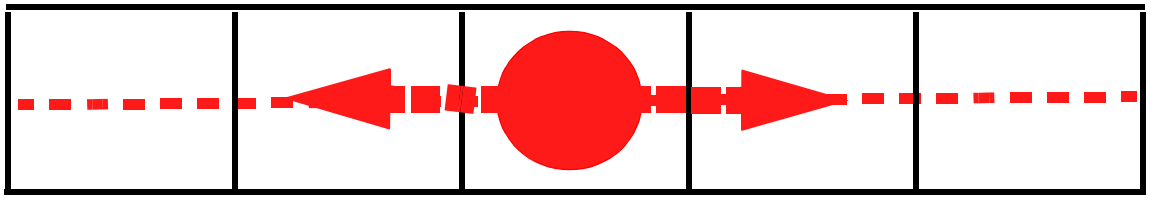
EX1: DIFFUSION - 1D

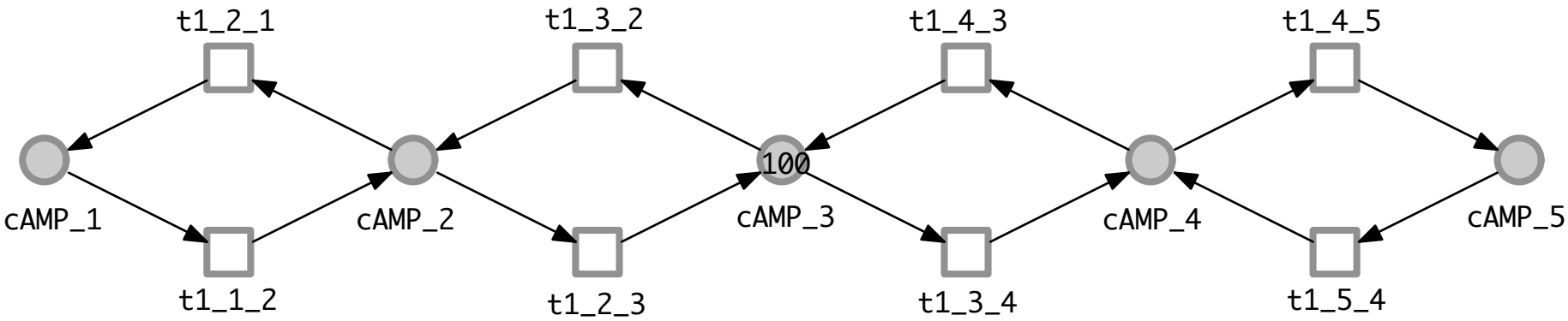


EX1: DIFFUSION - 1D



EX1: DIFFUSION - 1D





EX1: DIFFUSION - 1D

- **definitions**

```
const D1 = 5;              // grid size
const MIDDLE = D1/2;
colorset CS = 1-D1;        // grid positions
var x,y : CS;
```

EX1: DIFFUSION - 1D

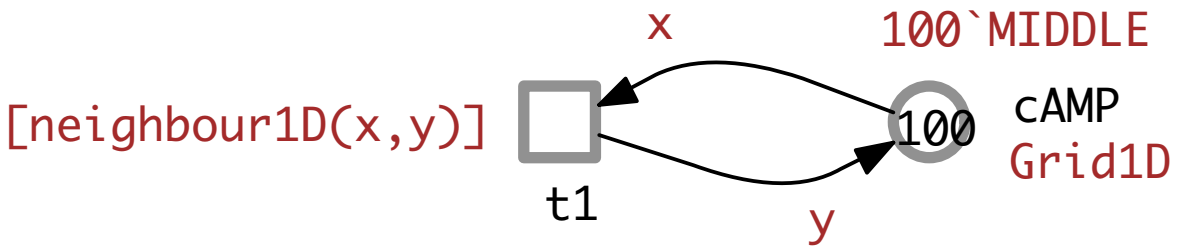
- **definitions**

```
const D1 = 5;              // grid size
const MIDDLE = D1/2;
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);
```

EX1: DIFFUSION - 1D

- **definitions**

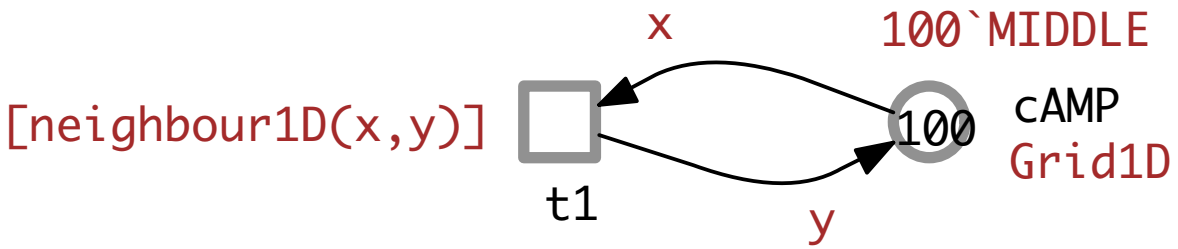
```
const D1 = 5;              // grid size
const MIDDLE = D1/2;
colorset CS = 1-D1;        // grid positions
var x,y : CS;

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



EX1: DIFFUSION - 1D

- **definitions**

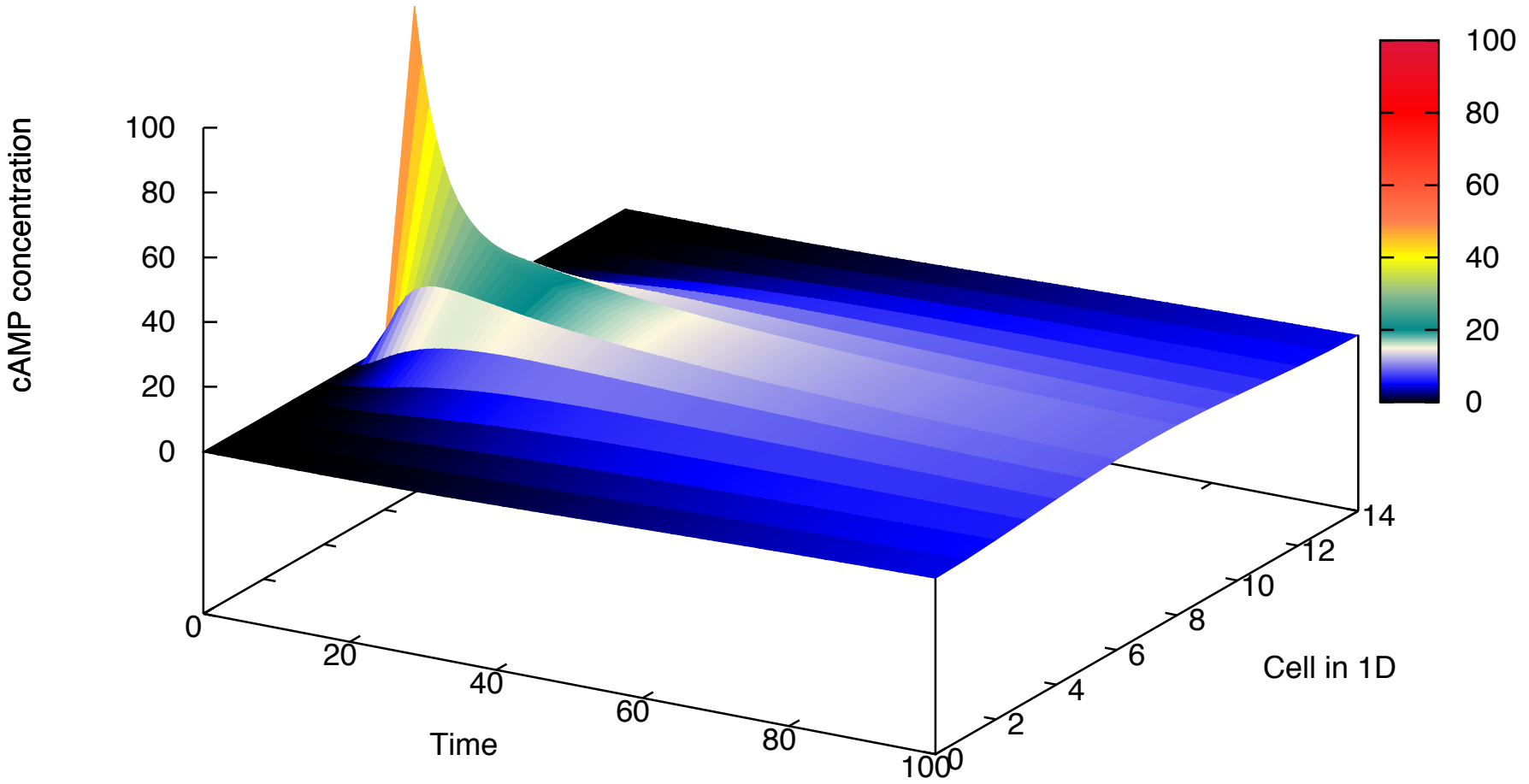
```
const D1 = 5;              // grid size
const MIDDLE = D1/2;
colorset CS = 1-D1;        // grid positions
var x,y : CS;

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



- **movement = changing colour**

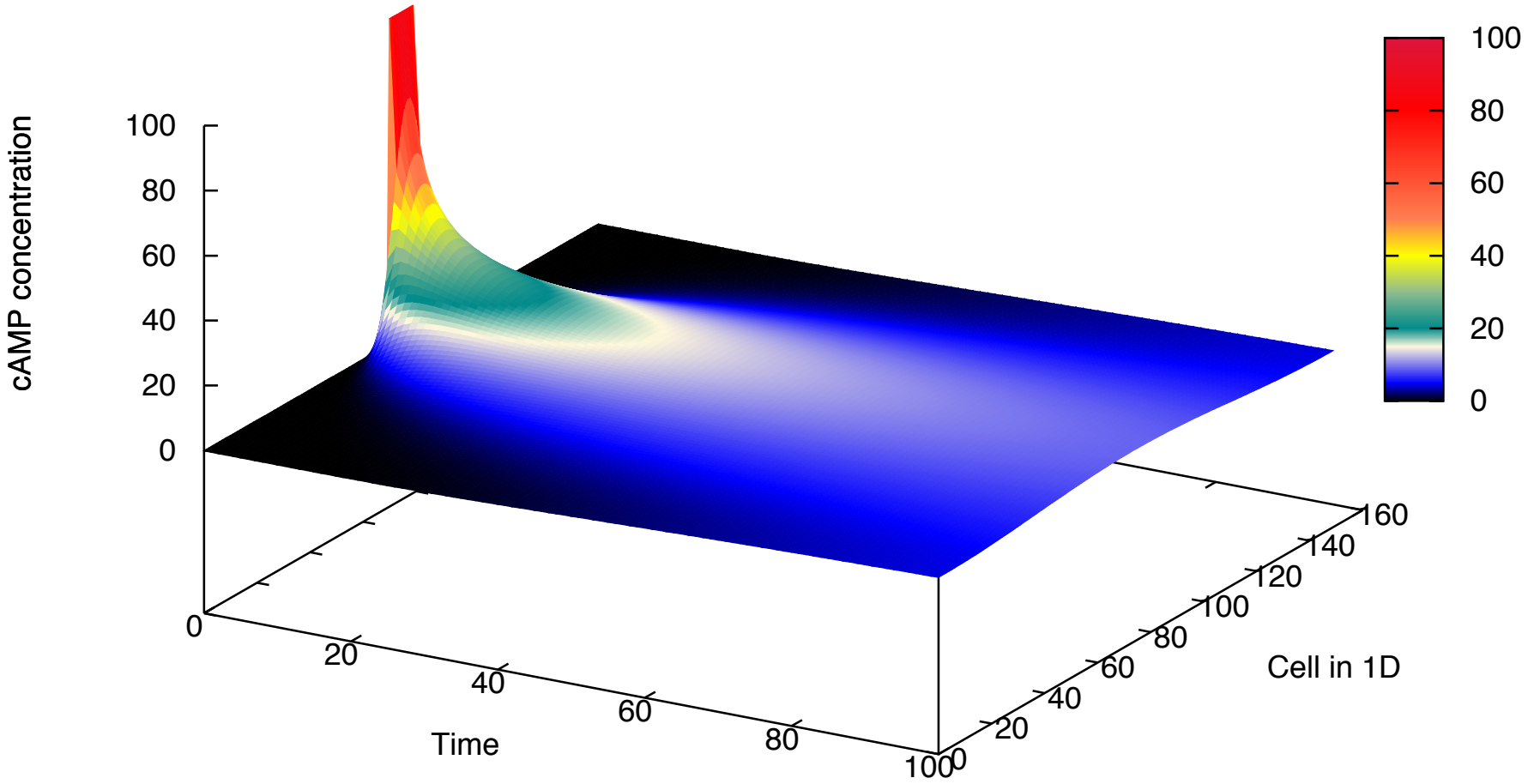
$$
\begin{aligned}
\frac{dc_1}{dt} &= k \cdot c_2 - k \cdot c_1 \\
\frac{dc_2}{dt} &= k \cdot c_1 + k \cdot c_3 - 2 \cdot k \cdot c_2 \\
\frac{dc_3}{dt} &= k \cdot c_2 + k \cdot c_4 - 2 \cdot k \cdot c_3 \\
\frac{dc_4}{dt} &= k \cdot c_3 + k \cdot c_5 - 2 \cdot k \cdot c_4 \\
\frac{dc_5}{dt} &= k \cdot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

EX1: DIFFUSION - 1D



15 GRID POSITIONS

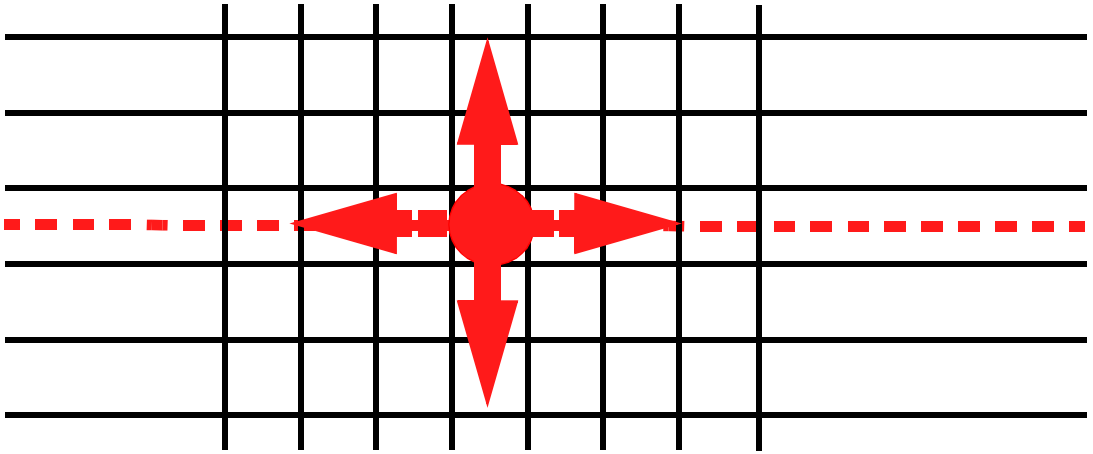
EX1: DIFFUSION - 1D



150 GRID POSITIONS, SCALING OF INITIAL MARKING AND RATES

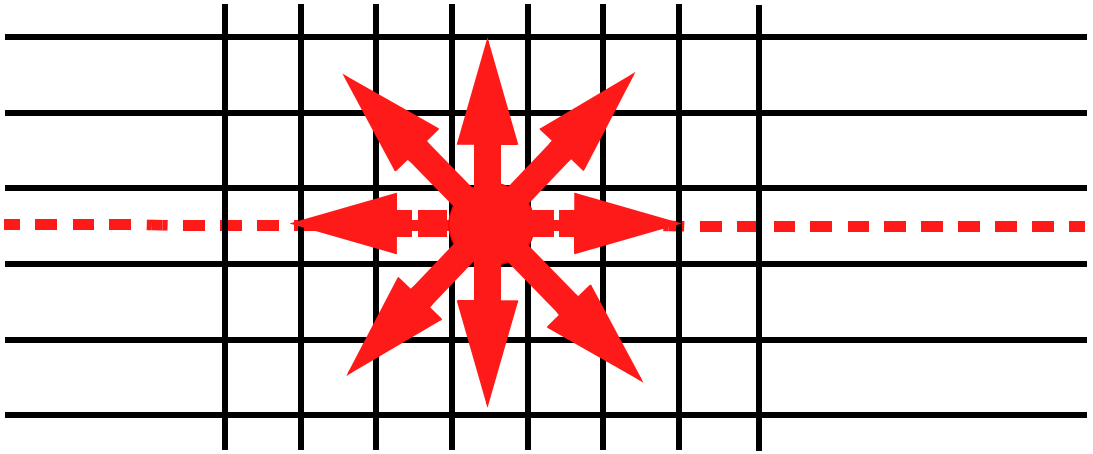
EX1: DIFFUSION - 2D

SCHEME

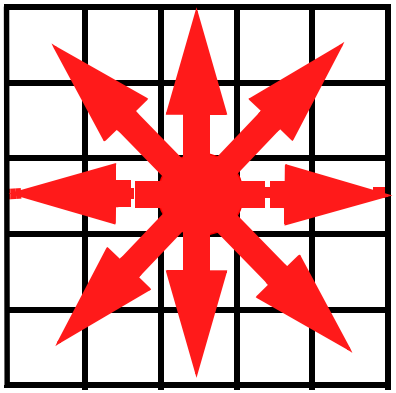


EX1: DIFFUSION - 2D

SCHEME

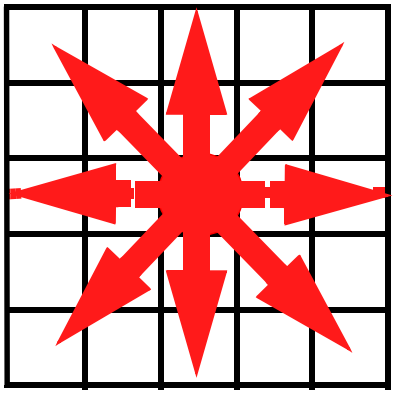


SCHEME



EX1: DIFFUSION - 2D

SCHEME

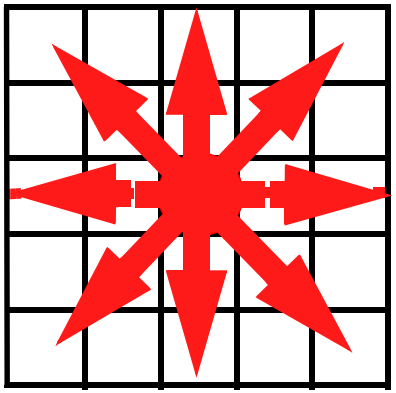


definitions

const *D1 = 5;* *// grid size first dimension*

const *D2 = D1;* *// grid size second dimension*

SCHEME

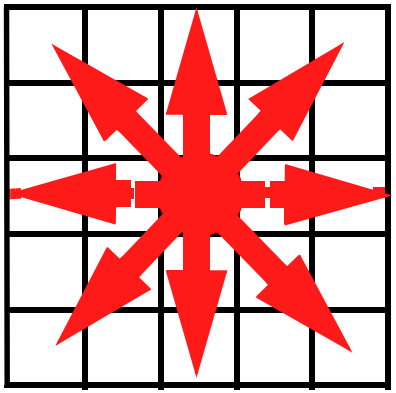


definitions

```
const D1 = 5;                    // grid size first dimension
const D2 = D1;                   // grid size second dimension
const MIDDLE = D1/2;

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

EX1: DIFFUSION - 2D

SCHEME



definitions

```
const D1 = 5;                    // grid size first dimension
const D2 = D1;                   // grid size second dimension
const MIDDLE = D1/2;

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;
```

EX1: DIFFUSION - 2D4 NEIGHBOURHOOD

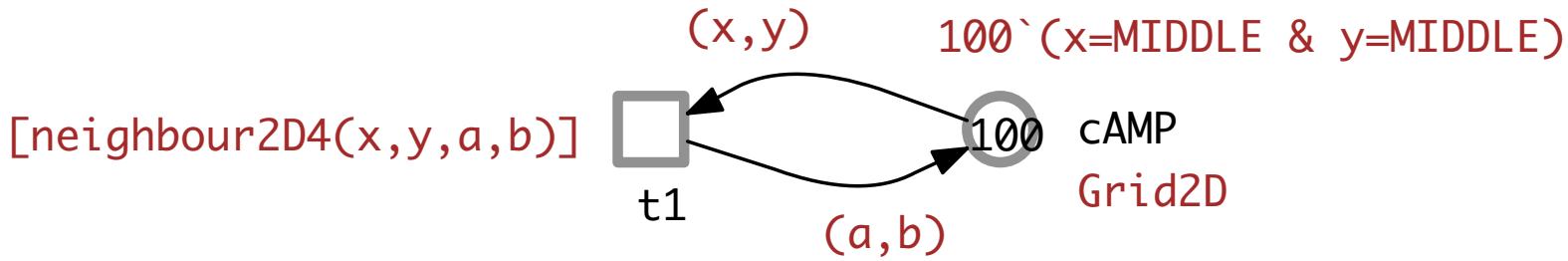
four neighbours

function *neighbour2D4 (CD1 x, CD2 y, CD1 a, CD2 b)* ***bool****:*

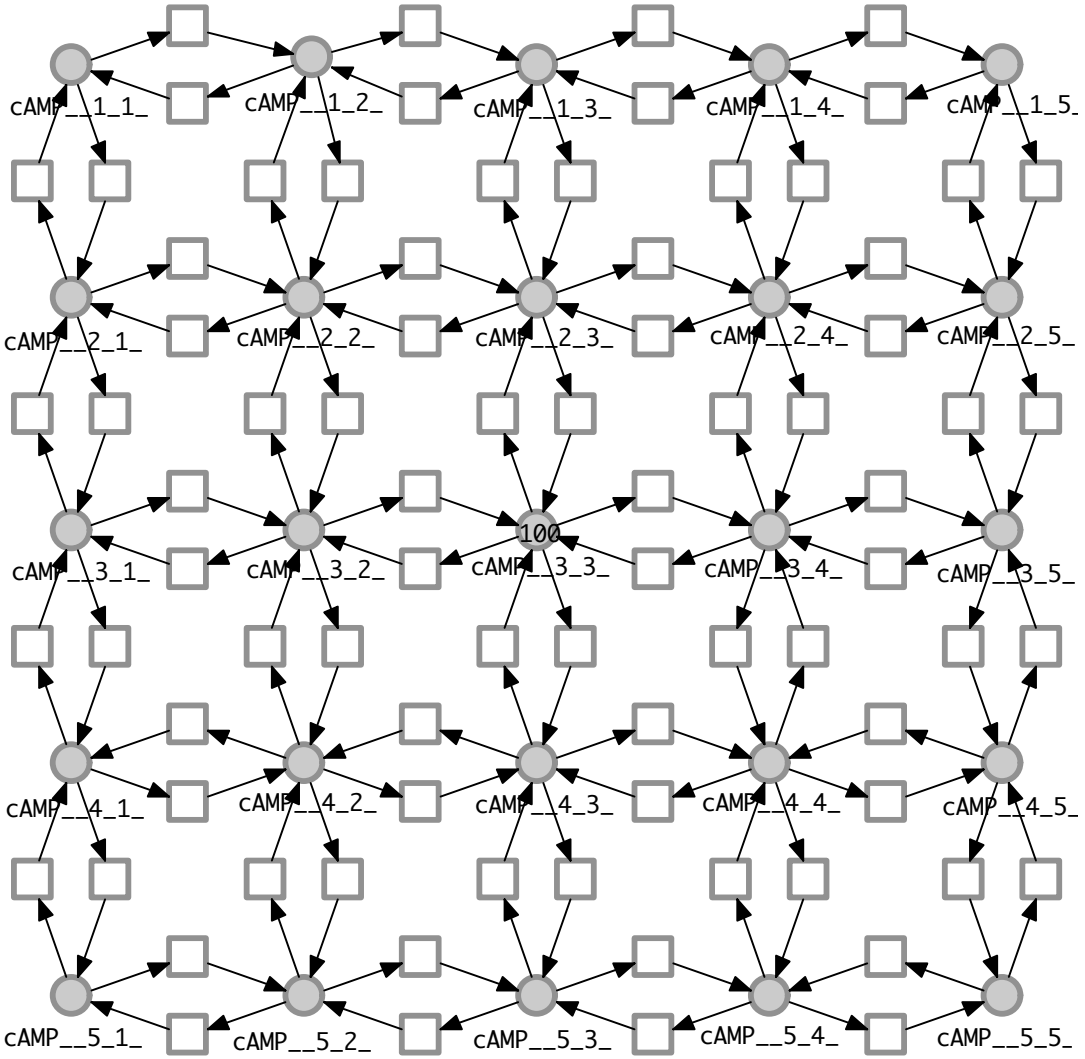
// (a,b) is one of the up to four neighbours of (x,y)

(a=x & b=y-1) | (a=x & b=y+1)

| (b=y & a=x-1) | (b=y & a=x+1);



EX1: DIFFUSION - 2D4 NEIGHBOURHOOD



EX1: DIFFUSION - 2D8 NEIGHBOURHOOD

- **eight neighbours**

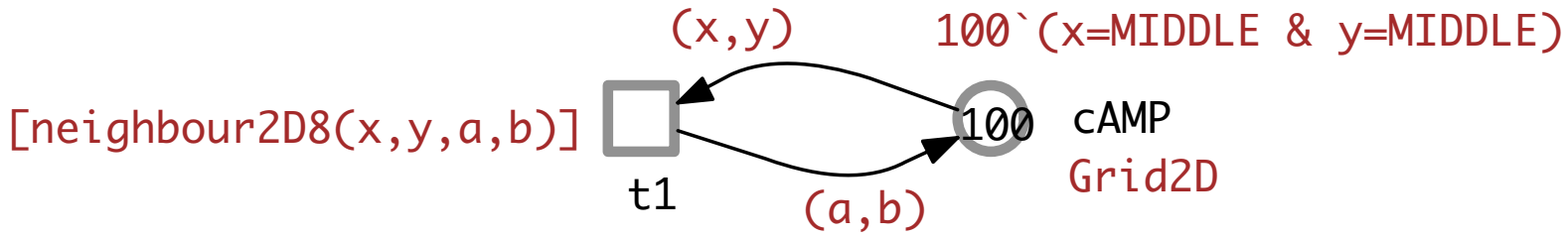
function *neighbour2D8 (CD1 x, CD2 y, CD1 a, CD2 b)* ***bool****:*

// (a,b) is one of the up to eight neighbours of (x,y)

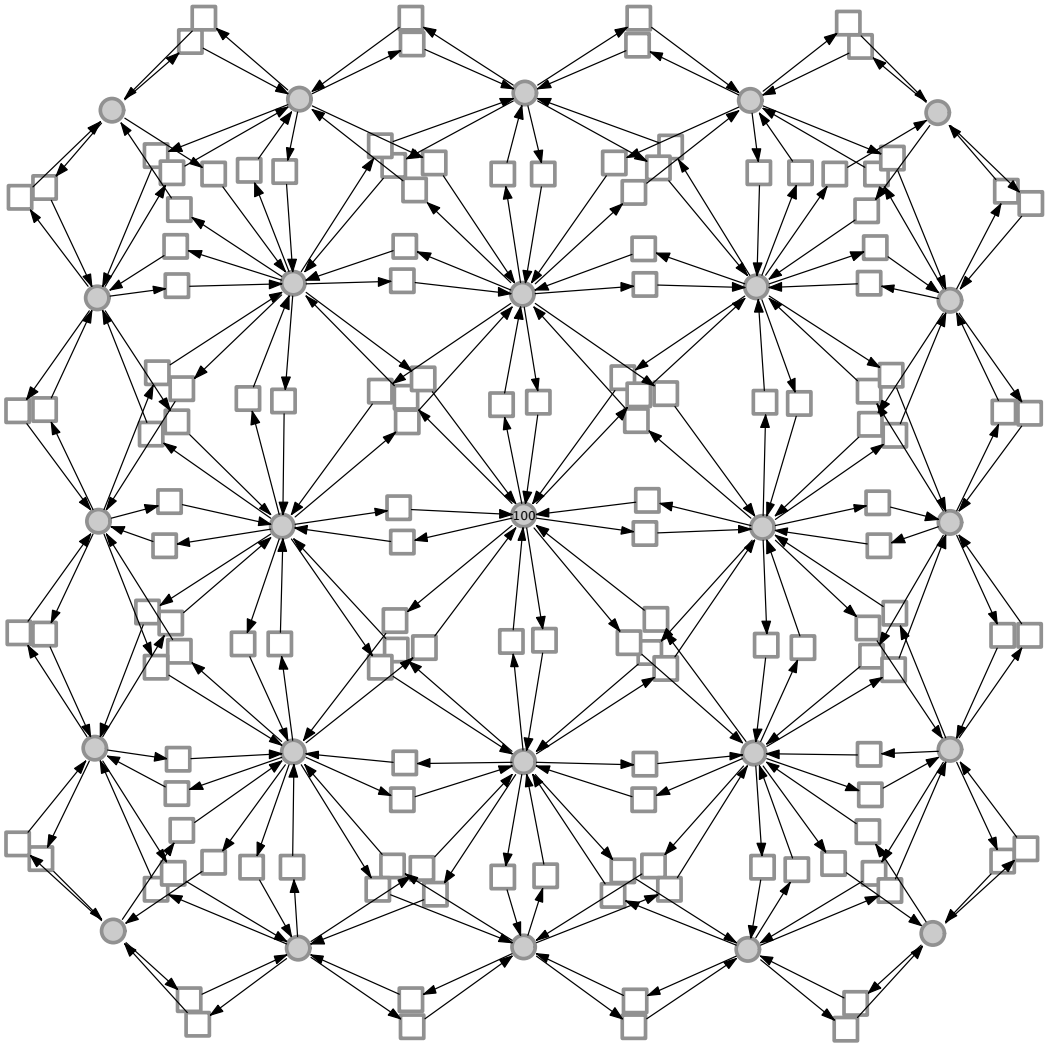
(a=x-1 | a=x | a=x+1) & (b = y-1 | b=y | b=y+1)

& (!(a=x & b=y))

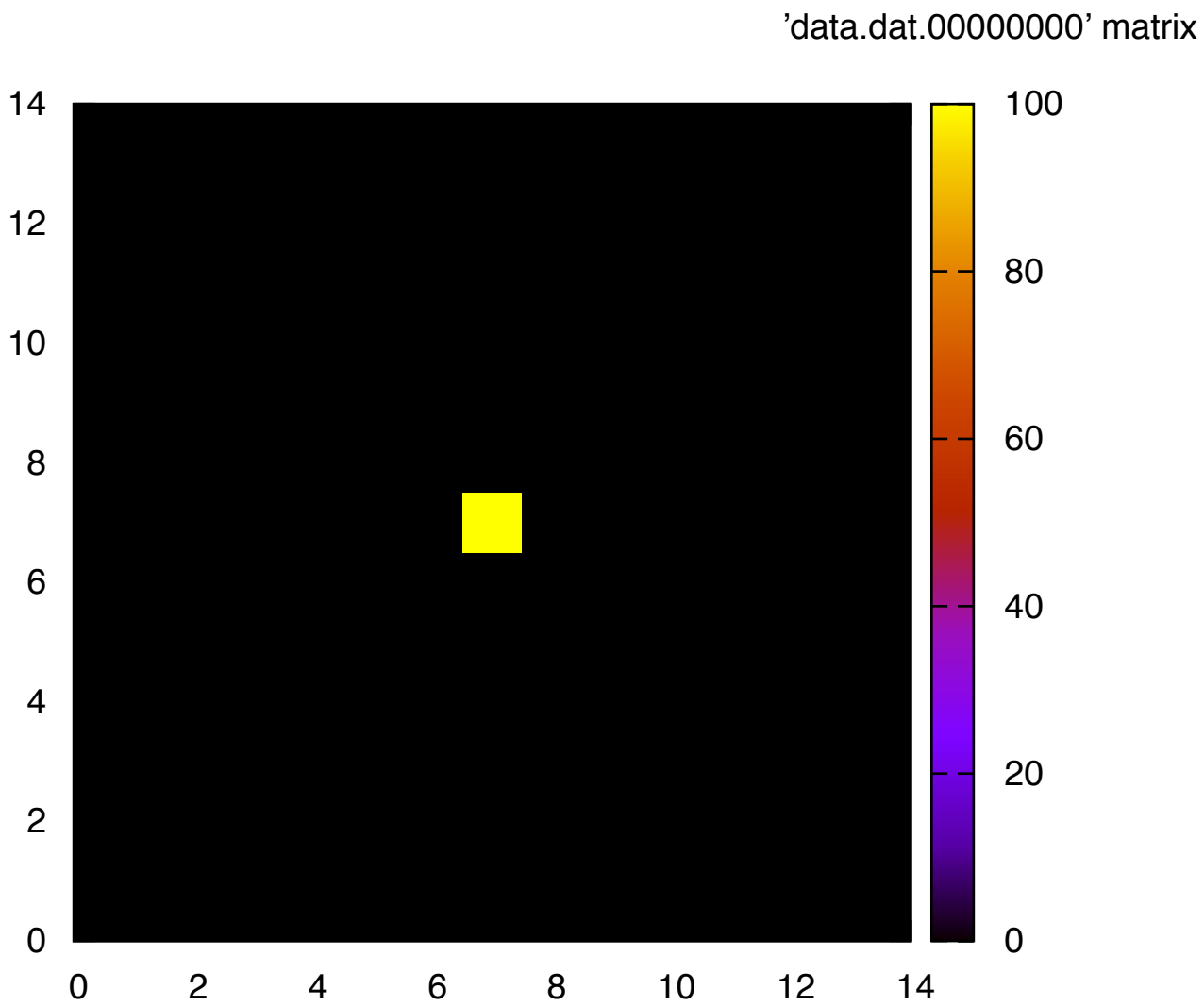
& (1<=a & a<=D1) & (1<=b & b<=D2);



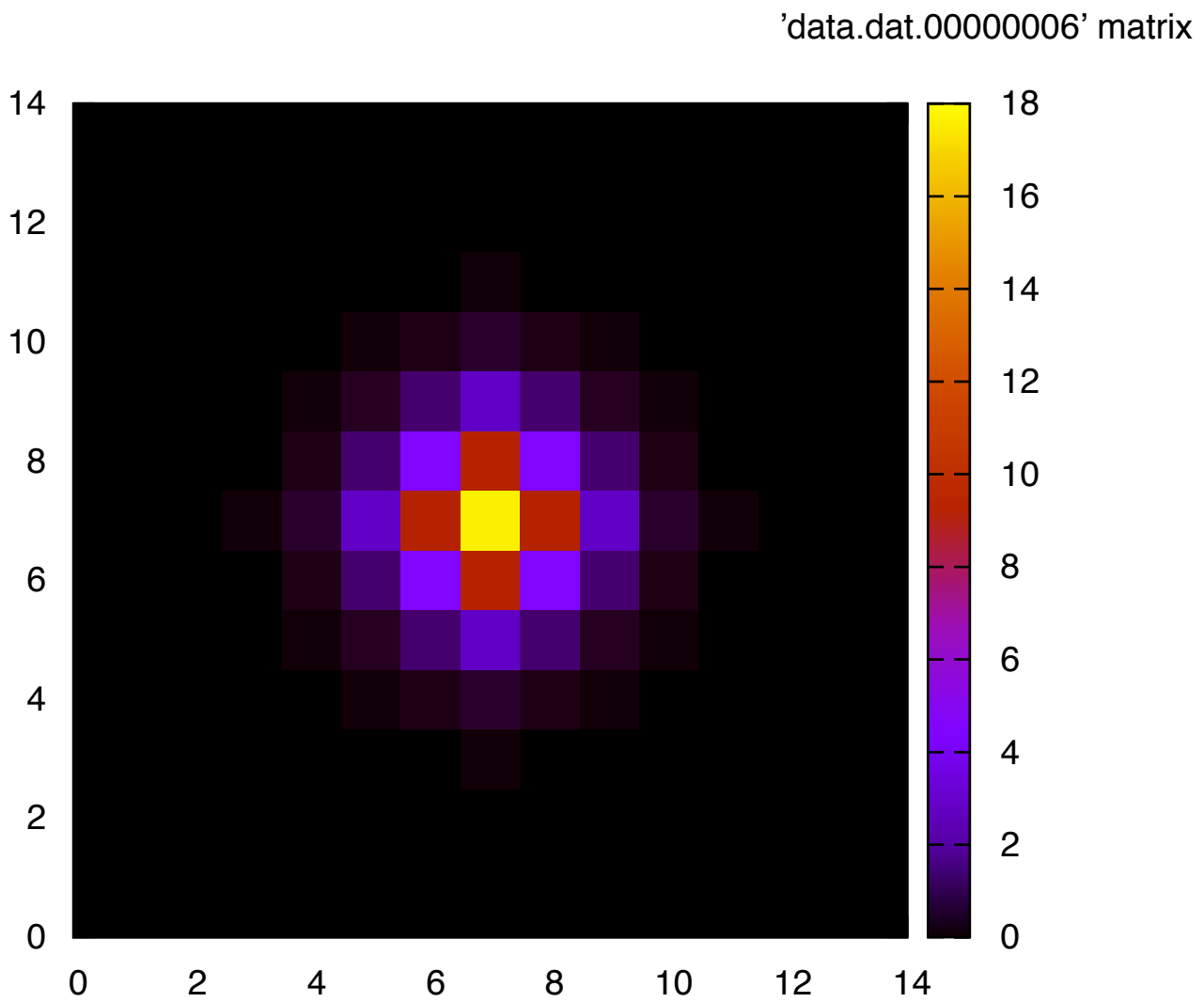
EX1: DIFFUSION - 2D8 NEIGHBOURHOOD



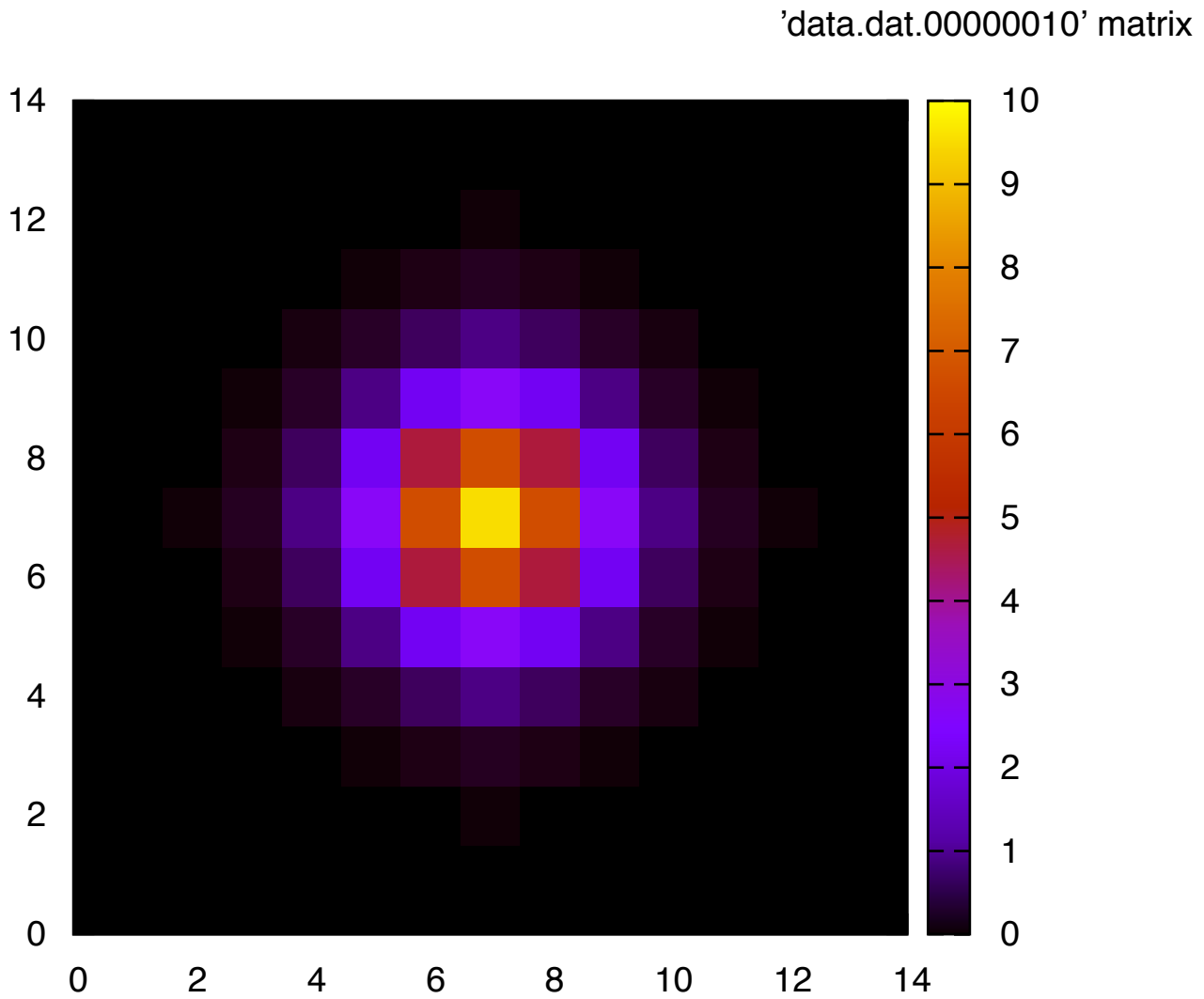
EX1: DIFFUSION - 2D4 NEIGHBOURHOOD, 15X15



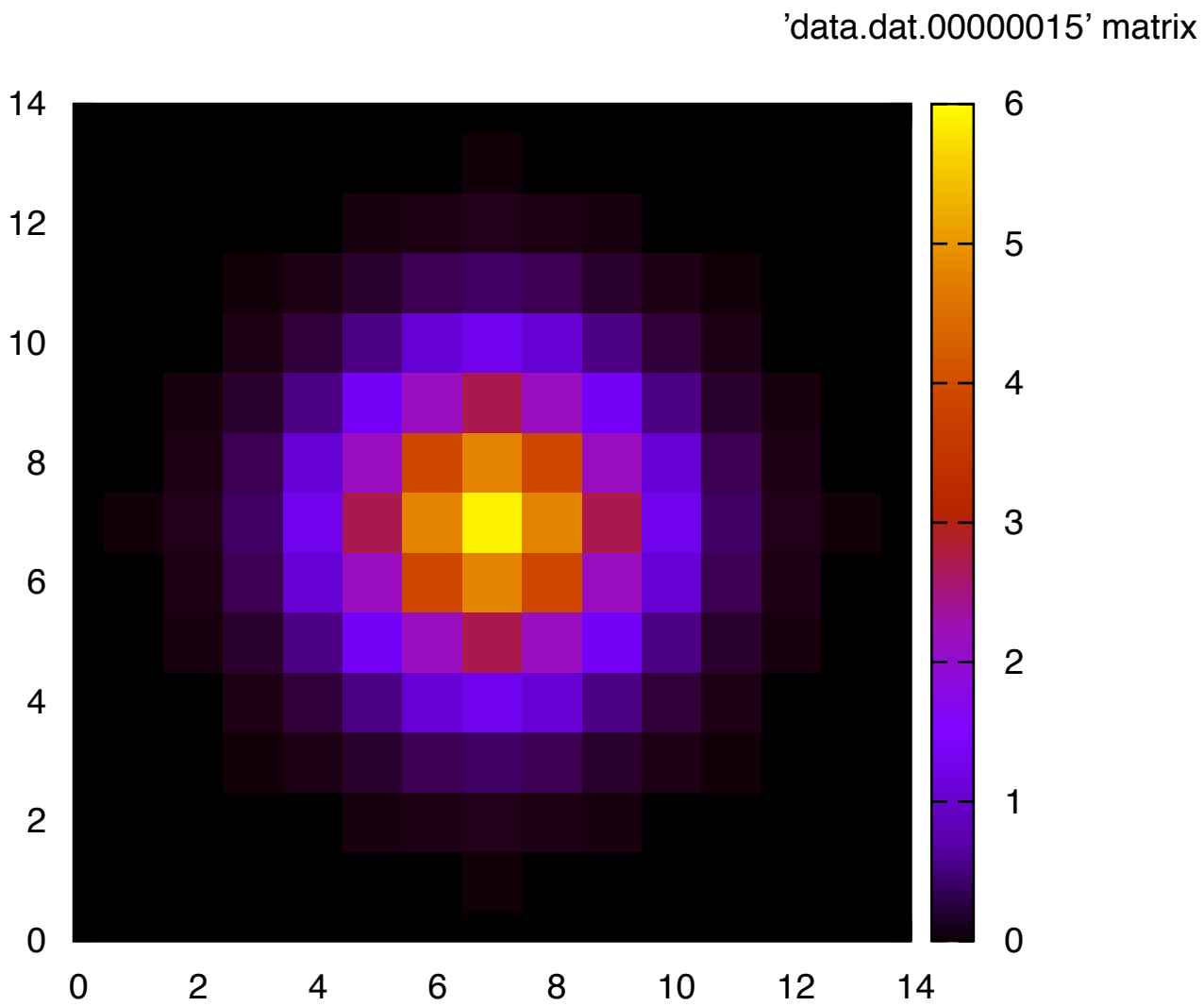
EX1: DIFFUSION - 2D4 NEIGHBOURHOOD, 15X15



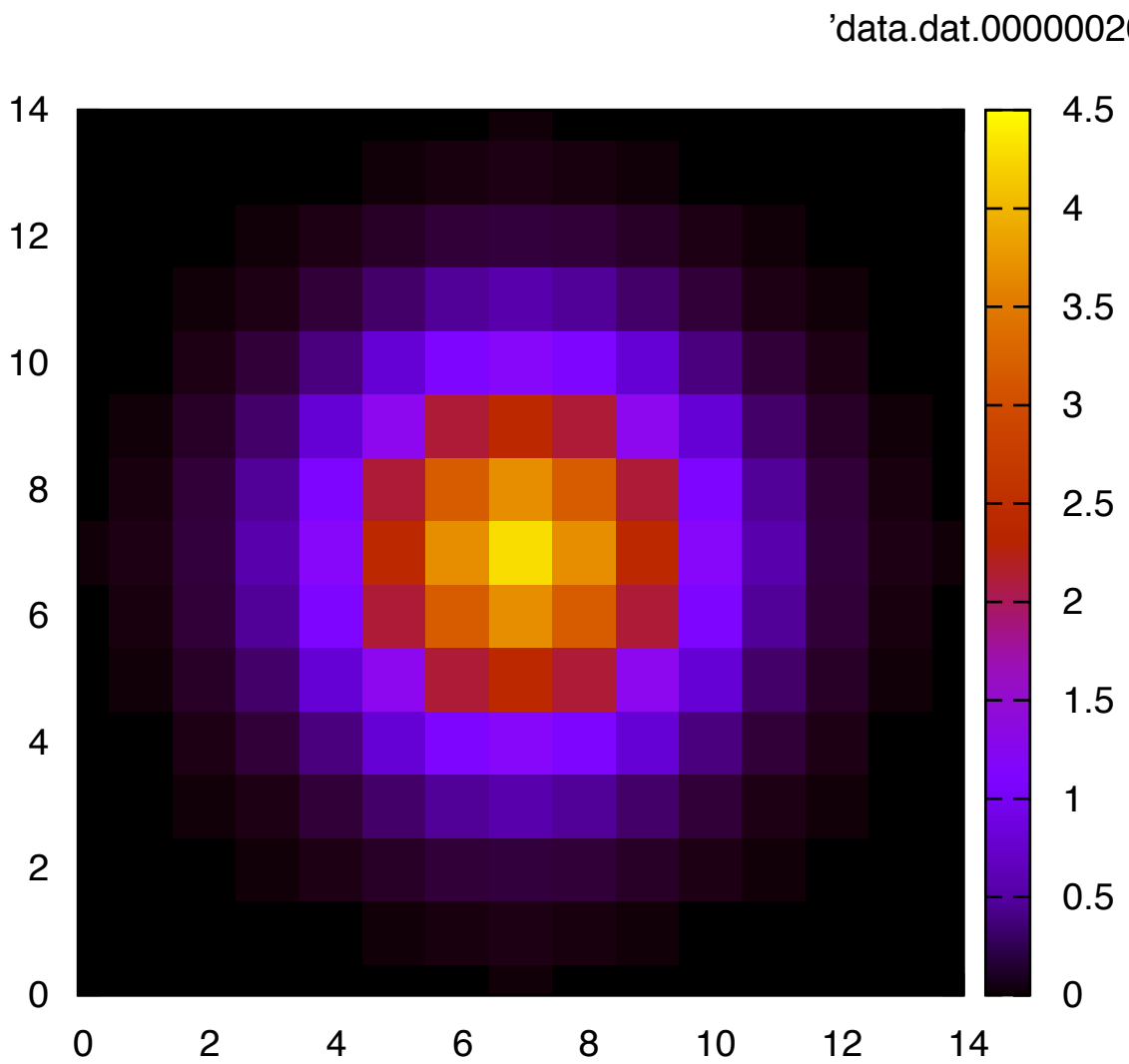
EX1: DIFFUSION - 2D4 NEIGHBOURHOOD, 15X15



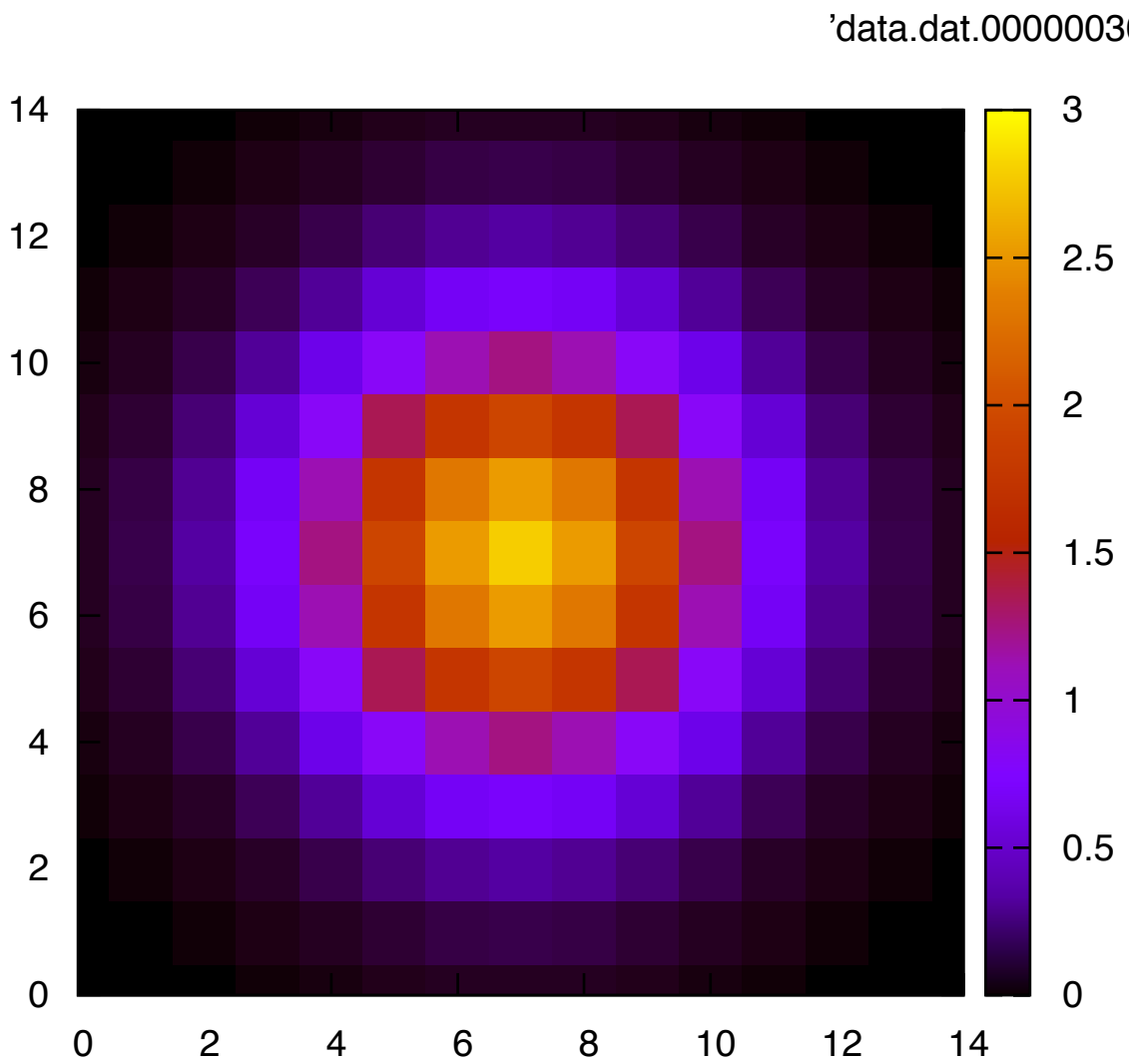
EX1: DIFFUSION - 2D4 NEIGHBOURHOOD, 15X15



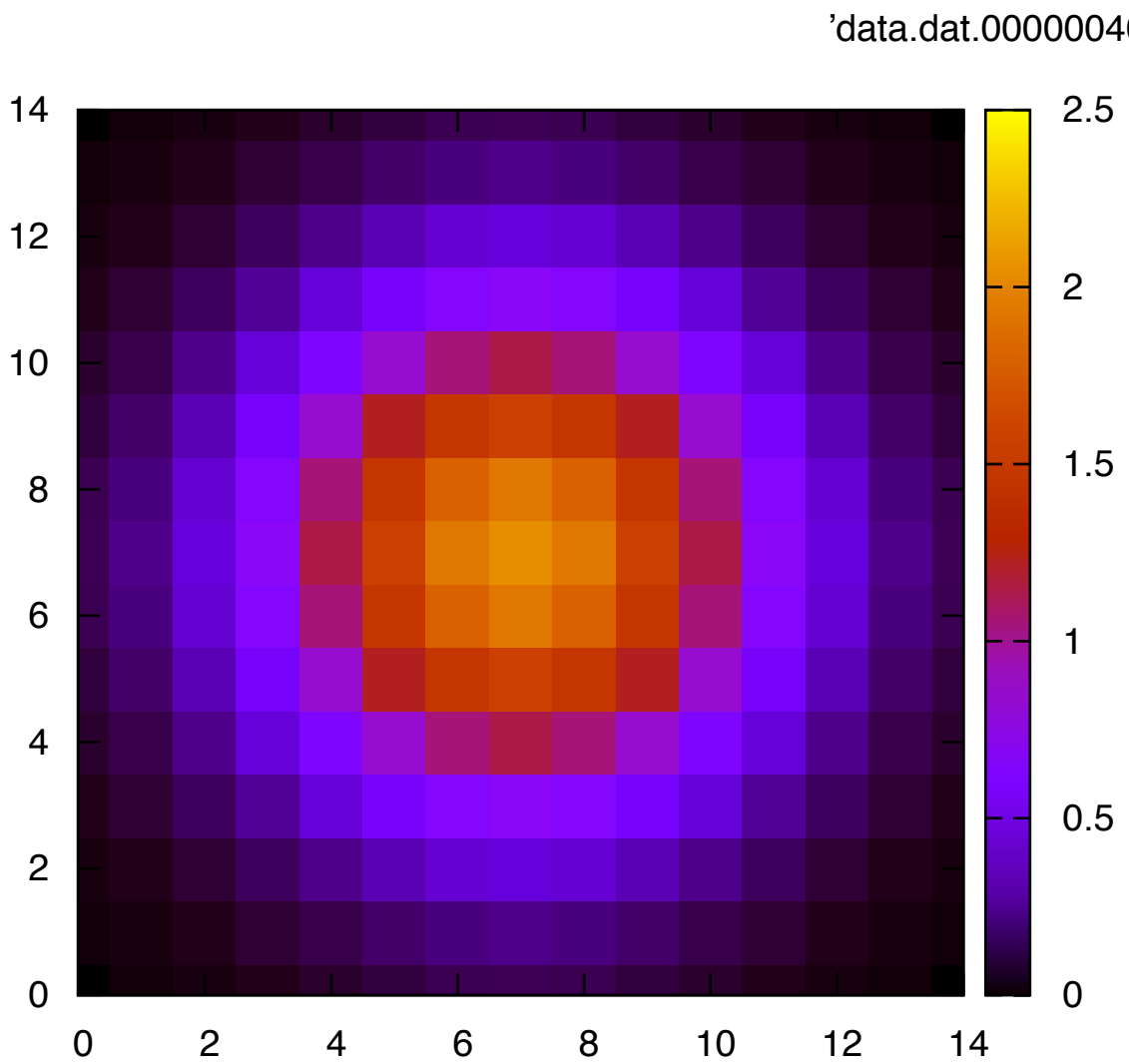
EX1: DIFFUSION - 2D4 NEIGHBOURHOOD, 15X15



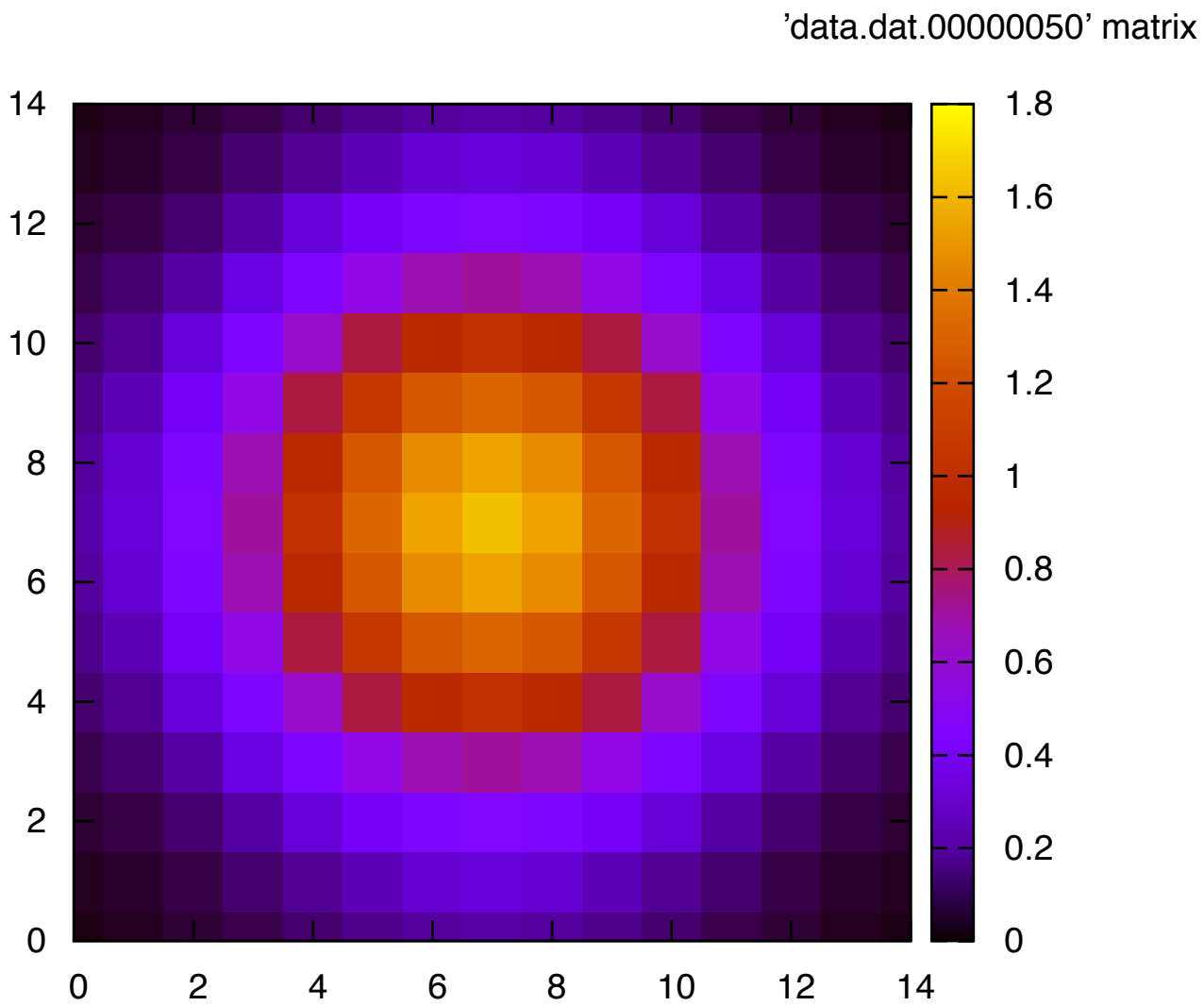
EX1: DIFFUSION - 2D4 NEIGHBOURHOOD, 15X15



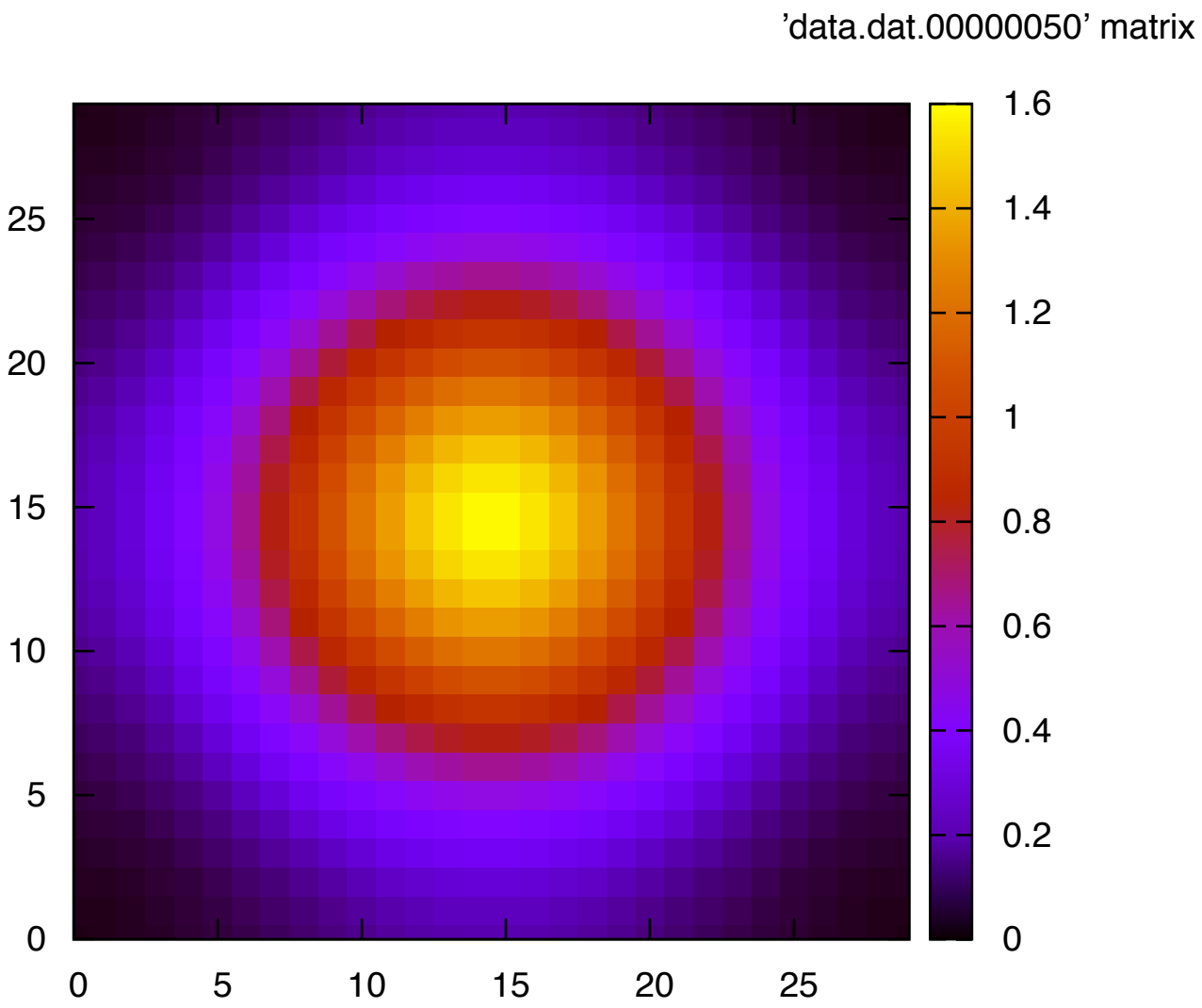
EX1: DIFFUSION - 2D4 NEIGHBOURHOOD, 15X15



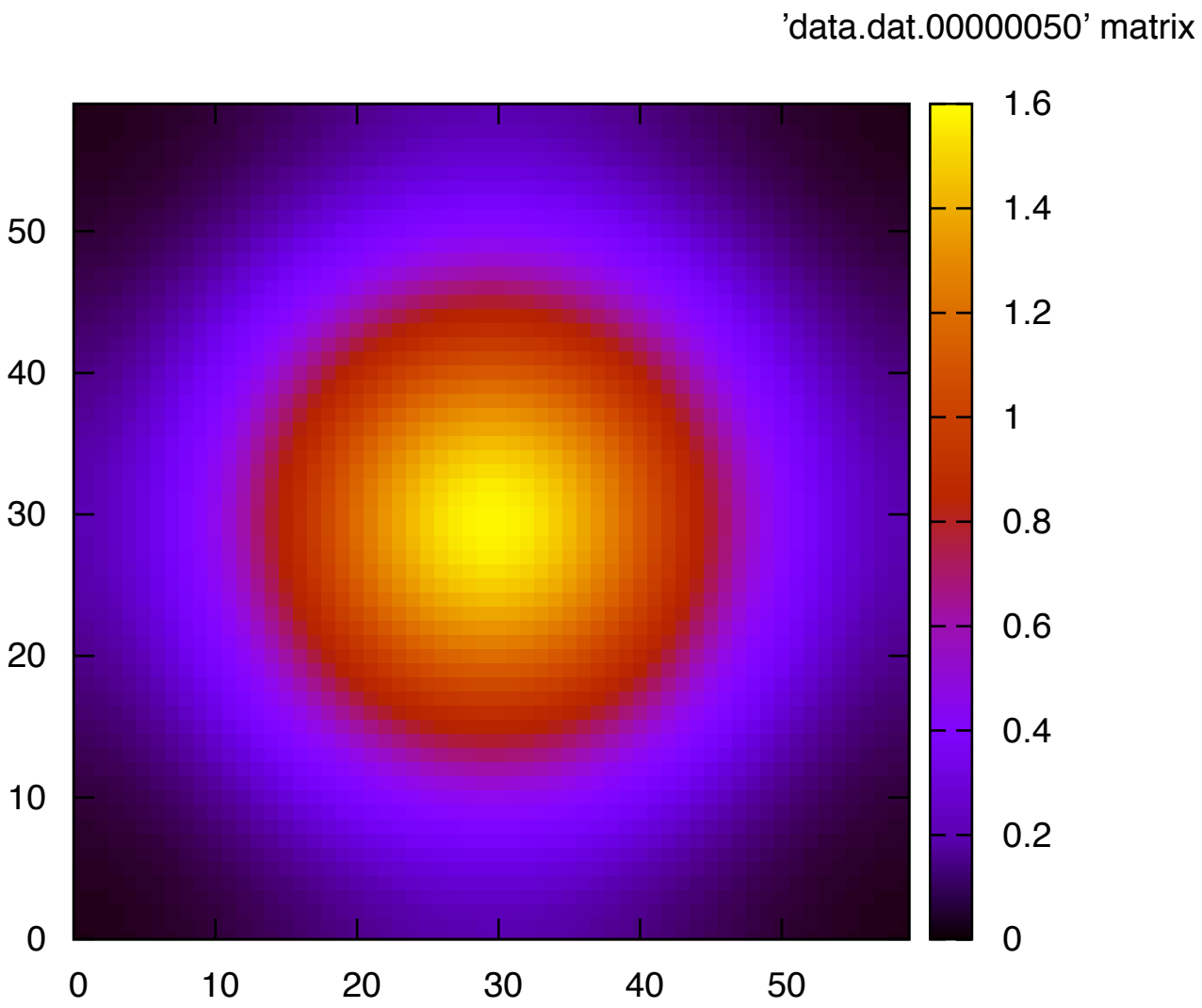
EX1: DIFFUSION - 2D4 NEIGHBOURHOOD, 15X15



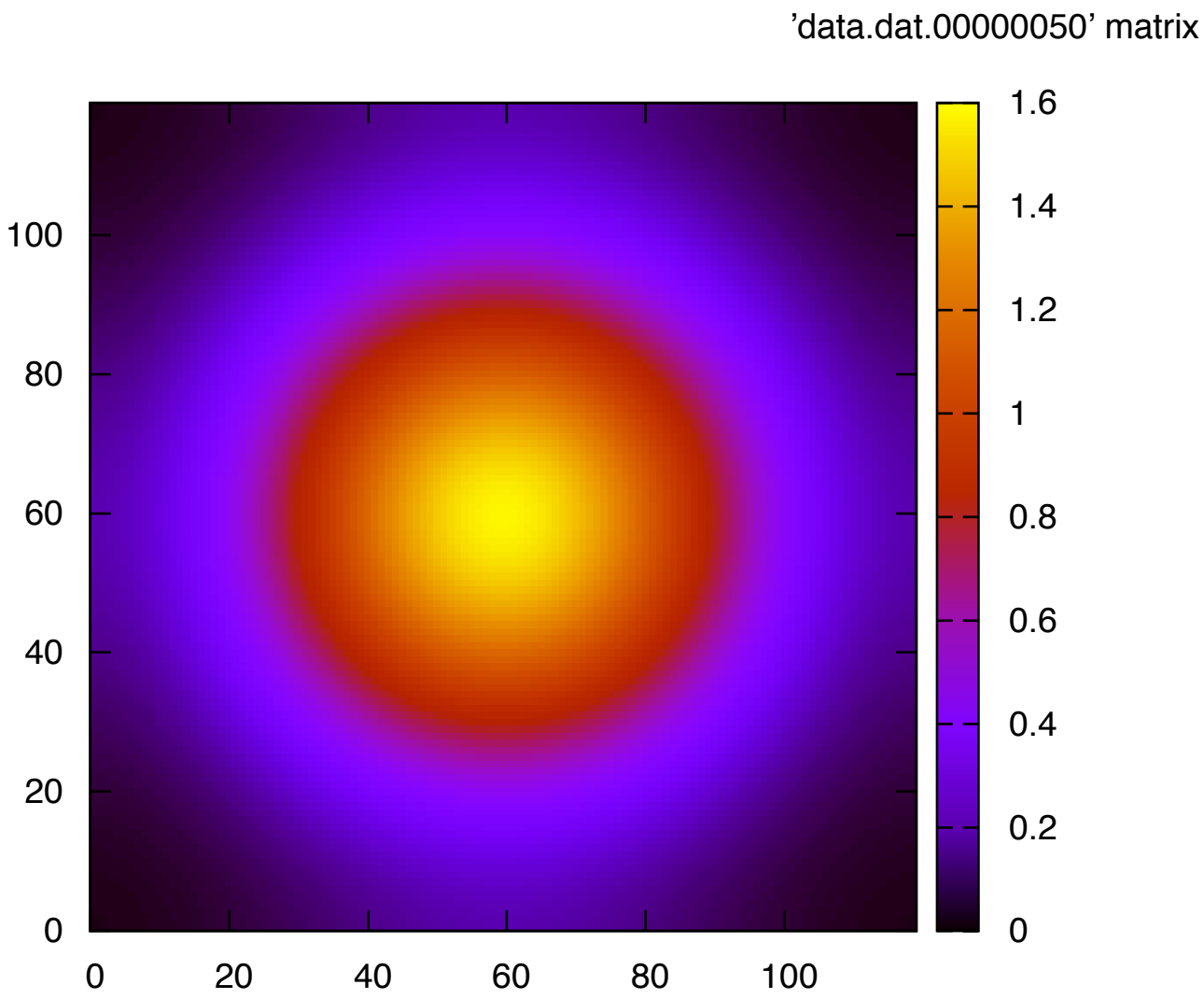
EX1: DIFFUSION - 2D4 NEIGHBOURHOOD, 30X30



EX1: DIFFUSION - 2D4 NEIGHBOURHOOD, 60X60



EX1: DIFFUSION - 2D4 NEIGHBOURHOOD, 120X120

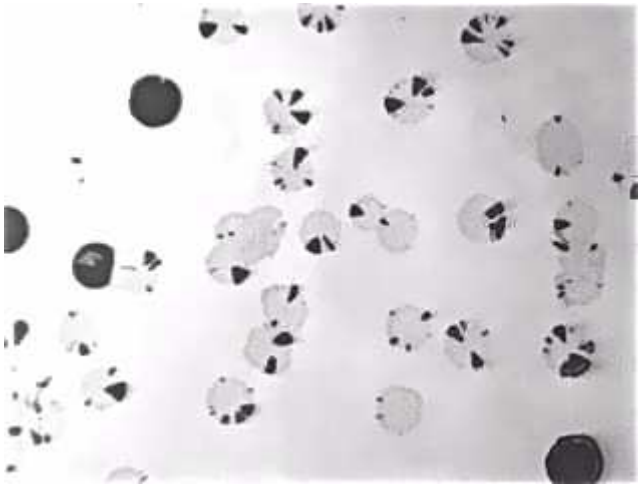


EXAMPLE 2:

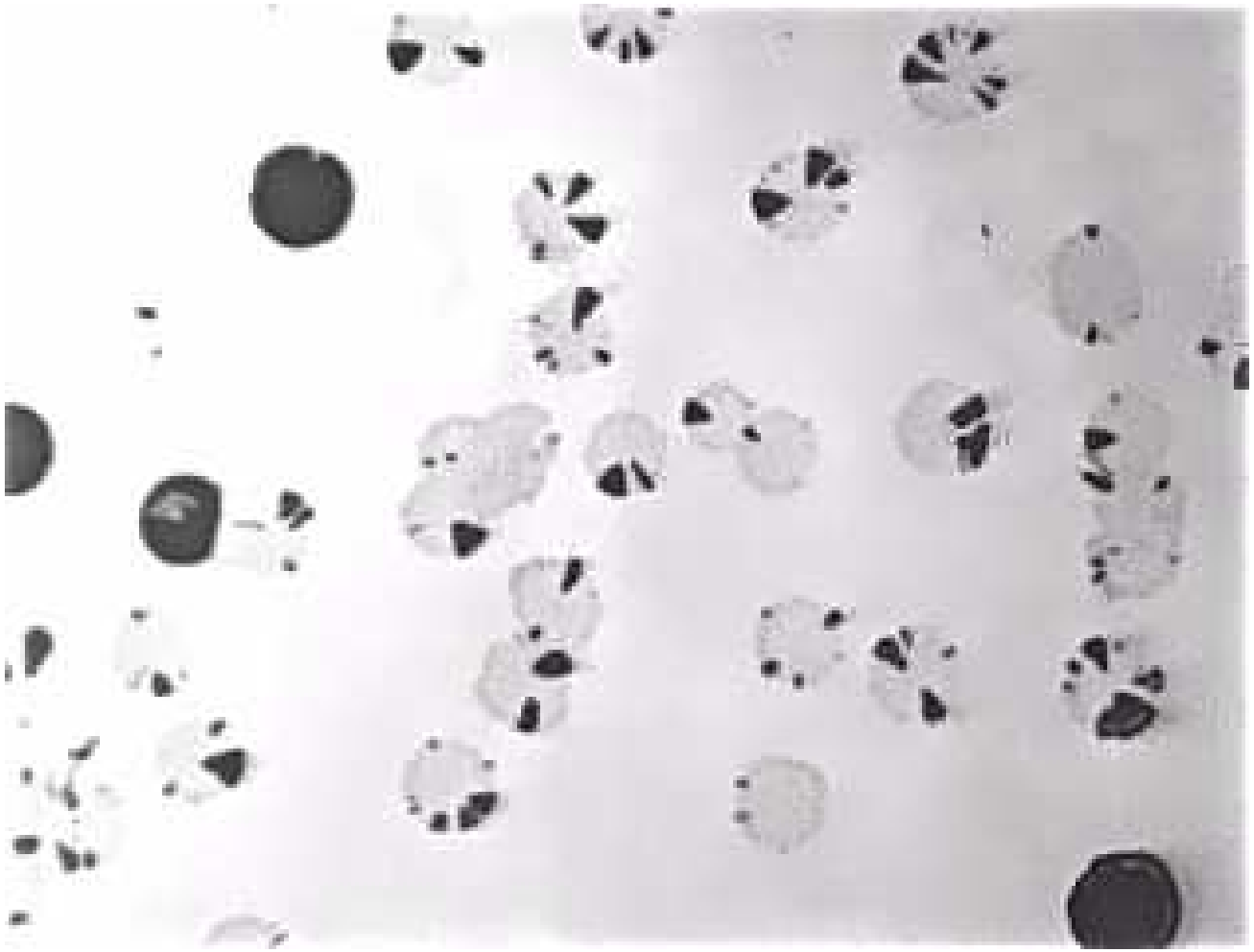
PHASE VARIATION IN MULTISTRAIN CELL COLONIES

EX2: PHASE VARIATION IN CELL COLONIES

- **phase variation**
  - -> *method for dealing with rapidly varying environments without requiring random mutations*
- **contingency genes**
  - -> *populations include variants adapted to "foreseeable" frequently encountered environmental or selective conditions*
- **stochastic gene switching process**
  - -> *controlled by reversible gene mutations, inversions, or epigenetic modification*
  - -> *e.g. switch between two phenotypes A, B*
- **colonial sectoring**
  - -> *observable effect in cultures grown in vitro*



EX2: CELL COLONIES, WETLAB OBSERVATIONS



(courtesy of N Saunders)

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]

[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

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

Microbiology (2003), **149**, 485–495

DOI 10.[illegible]/mic.0.25807-0

Mutation rates: estimati[illegible] phas[illegible] variation [illegible]s when fitness diff[illegible]nce[illegible] a[illegible] pre[illegible]nt a[illegible]d their impact on po[illegible]latio[illegible] st[illegible]

Nigel J.[illegible][1]† E[illegible]ichard M[illegible]xon[1] and Mike B. Gravenor[2]

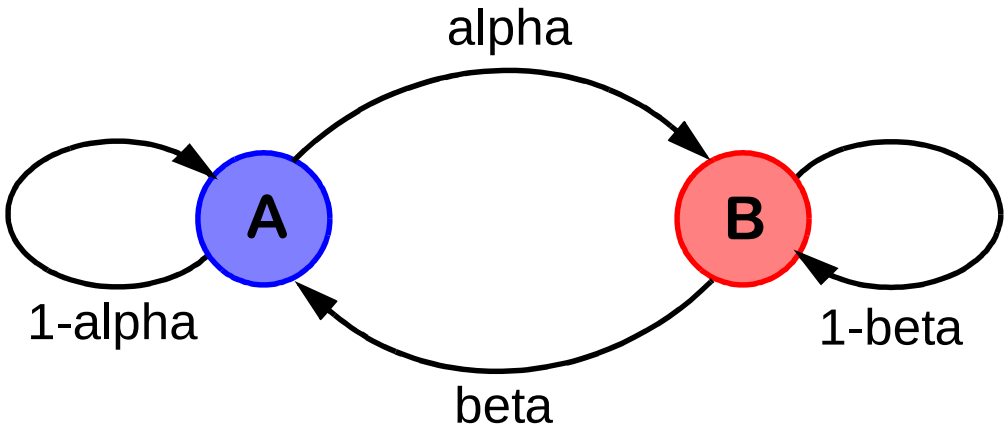
Correspondence
Nigel J. S[illegible]ders
[illegible]nders@[illegible]biol.o[illegible]uk

[1]Molecular Inf[illegible]ious D[illegible]ases Group, Institute of Molecular Medicine, University of Oxford, [illegible]eadington, O[illegible]X3 9DS, UK

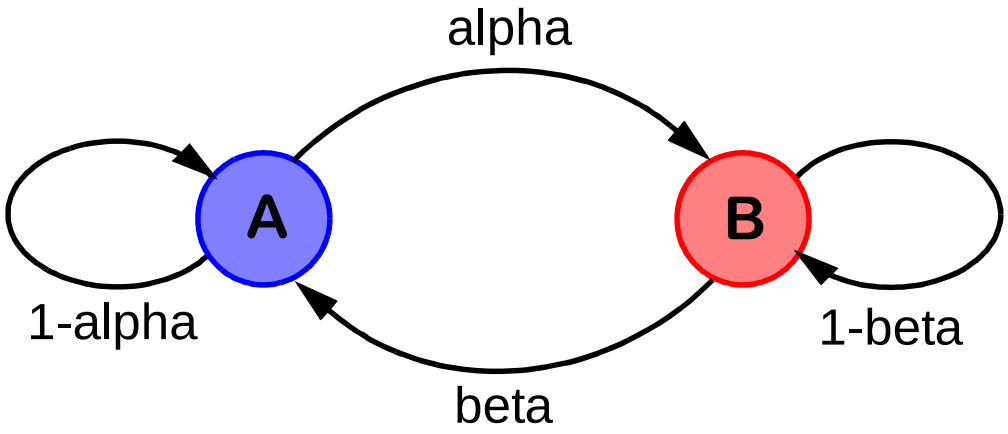
[2]Institute for Animal Health, Compton, Berkshire RG20 7NN, UK

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.

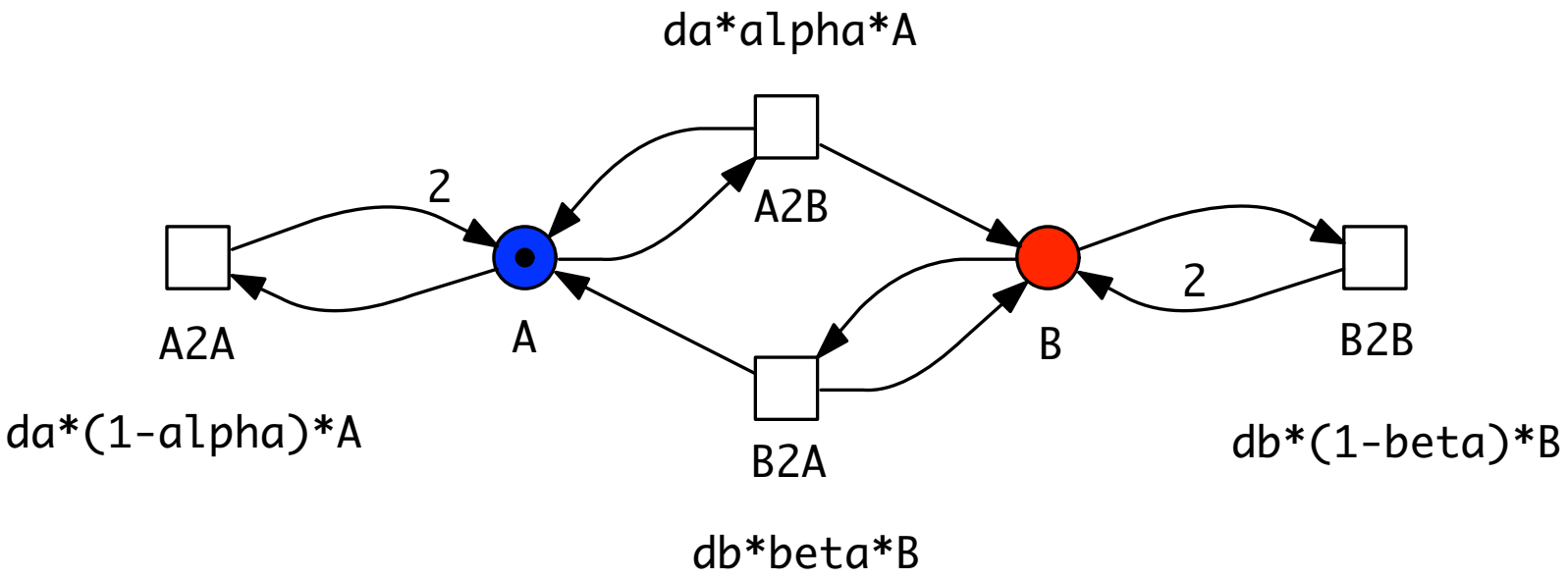
- **two cell types: phenotype A and B**
- **cell divide**
  - -> *cell division may involve mutation of the offspring*
  - -> *parent cell keeps its phenotype*



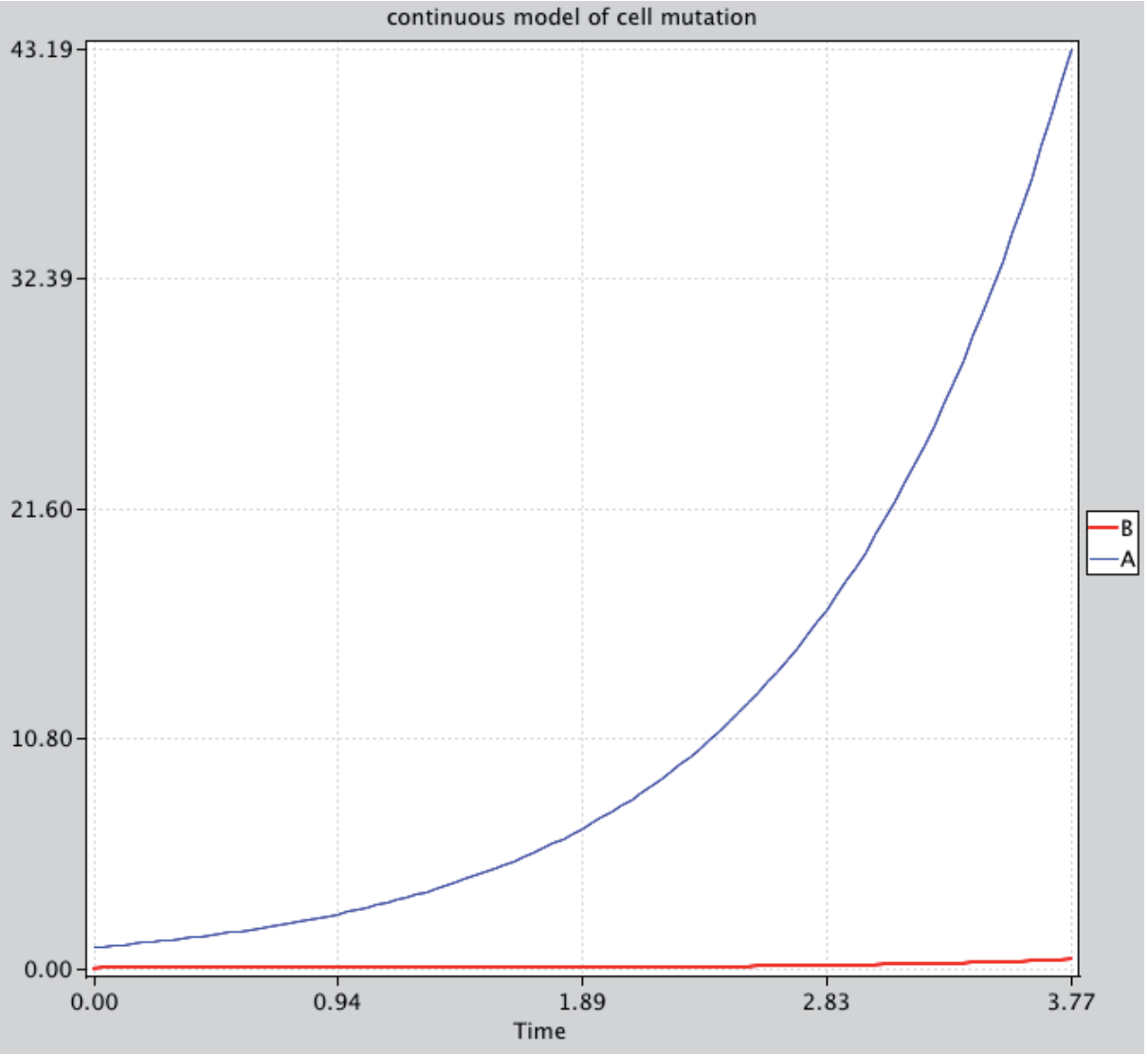
- **two cell types: phenotype A and B**
- **cell divide**
  - -> *cell division may involve mutation of the offspring*
  - -> *parent cell keeps its phenotype*
- **model parameters**
  - -> *alpha = beta - mutation rates*
  - -> *da, db - fitness of A, B*
  - -> *da/db - relative fitness*
- **output**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



EX2: CELL COLONIES, PETRI NET

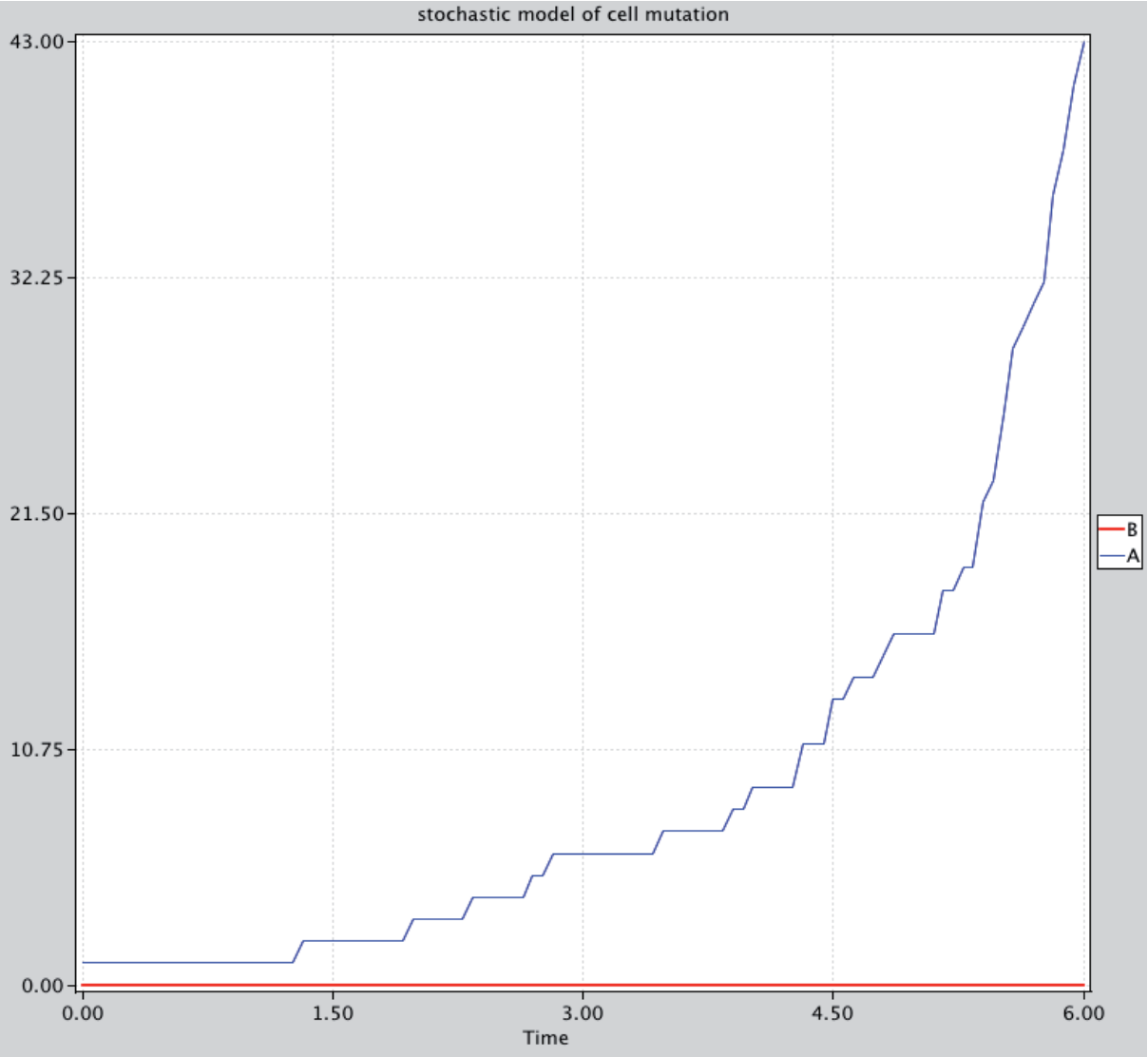


EX2: CELL COLONIES, CONTINUOUS PLOT



monika.heiner@b-tu.de

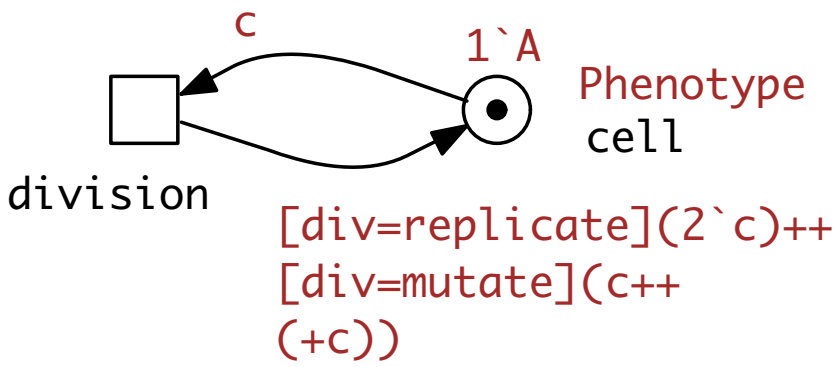
EX2: CELL COLONIES, STOCHASTIC PLOT



colorset Phenotype = **enum with** A, B;
colorset DivisionType = **enum with** replicate , mutate ;

EX2: CELL COLONIES, BASIC MODEL

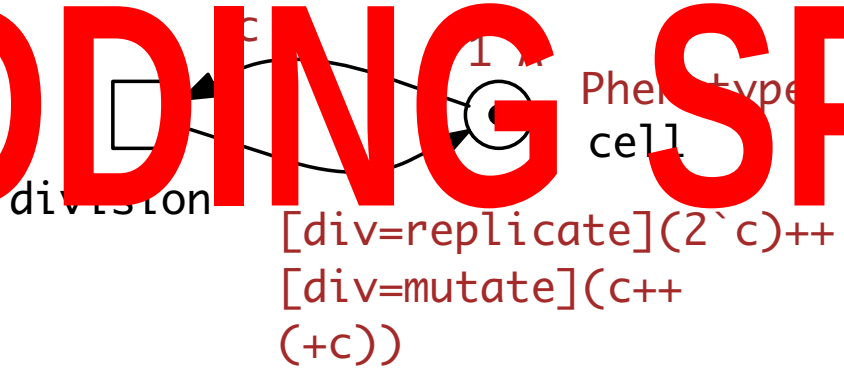
colorset Phenotype = **enum with** A, B;
colorset DivisionType = **enum with** replicate , mutate ;




```
(c=A) & (div=replicate) : cell*da*(1-alpha)
(c=A) & (div=mutate) : cell*(da*alpha)
(c=B) & (div=replicate) : cell*(db*(1-beta))
(c=B) & (div=mutate) : cell*(db*beta)
```

colorset Phenotype = **enum with** A, B;
colorset DivisionType = **enum with** replicate , mutate ;

ADDING SPACE




```
(c=A) & (div=replicate) : cell*da*(1-alpha)
(c=A) & (div=mutate) : cell*(da*alpha)
(c=B) & (div=replicate) : cell*(db*(1-beta))
(c=B) & (div=mutate) : cell*(db*beta)
```

colorset Phenotype = **enum with** A, B;
colorset DivisionType = **enum with** replicate , mutate ;

ADDING SPACE

CONTROLLING COLONY SPREADING

```
(c=A) & (div=replicate) : cell*da*(1-alpha)
(c=A) & (div=mutate) : cell*(da*alpha)
(c=B) & (div=replicate) : cell*(db*(1-beta))
(c=B) & (div=mutate) : cell*(db*beta)
```

colorset Phenotype = **enum with** A, B;
colorset DivisionType = **enum with** replicate , mutate ;

[div=replicate](2`c)++
[div=mutate](c+1
(+c))

(c=A) & (div=replicate) : cell*da*(1-alpha)
(c=A) & (div=mutate) : cell*(da*alpha)
(c=B) & (div=replicate) : cell*(db*(1-beta))
(c=B) & (div=mutate) : cell*(db*beta)

ADDING SPACE

CONTROLLING COLONY SPREADING

CONTROLLING THICKNESS

colorset Phenotype = **enum with** A, B;
colorset DivisionType = **enum with** replicate , mutate ;

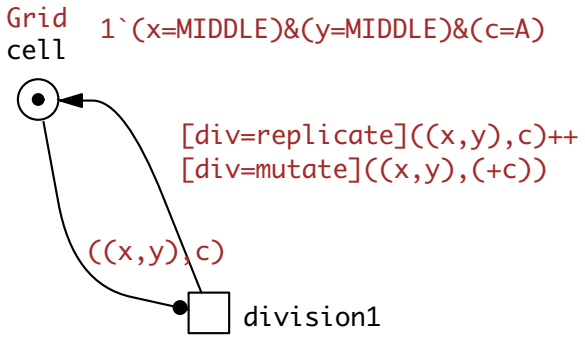
ADDING SPACE

CONTROLLING COLONY SPREADING

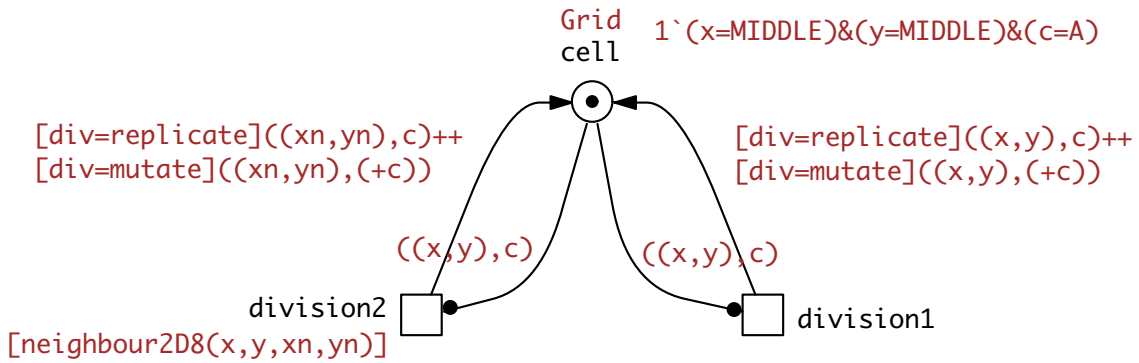
CONTROLLING THICKNESS

CONTROLLING COLONY SIZE

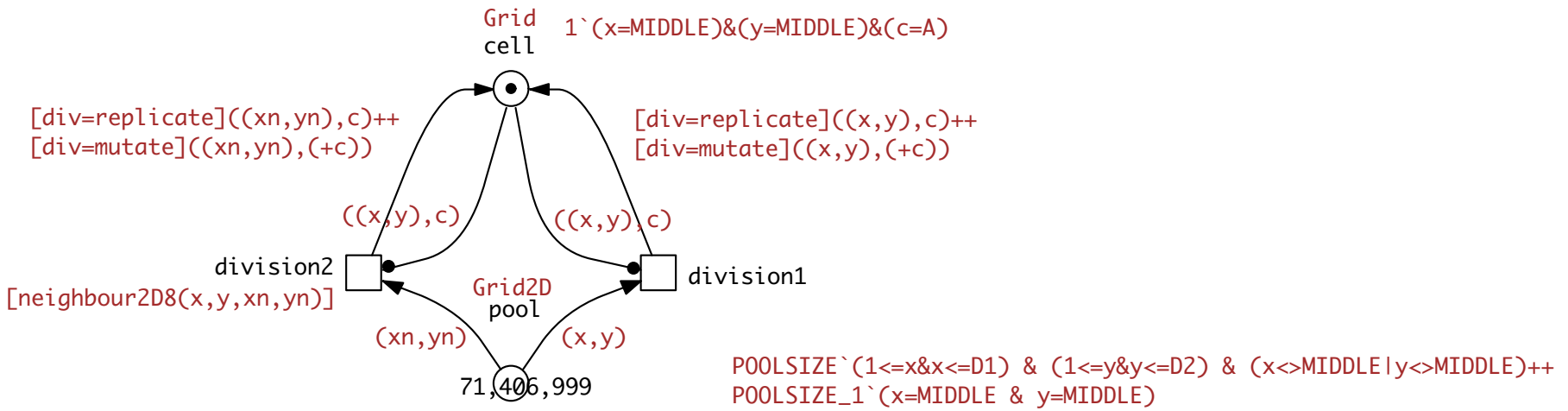
colorset Grid = **product with** Grid2D x Phenotype;



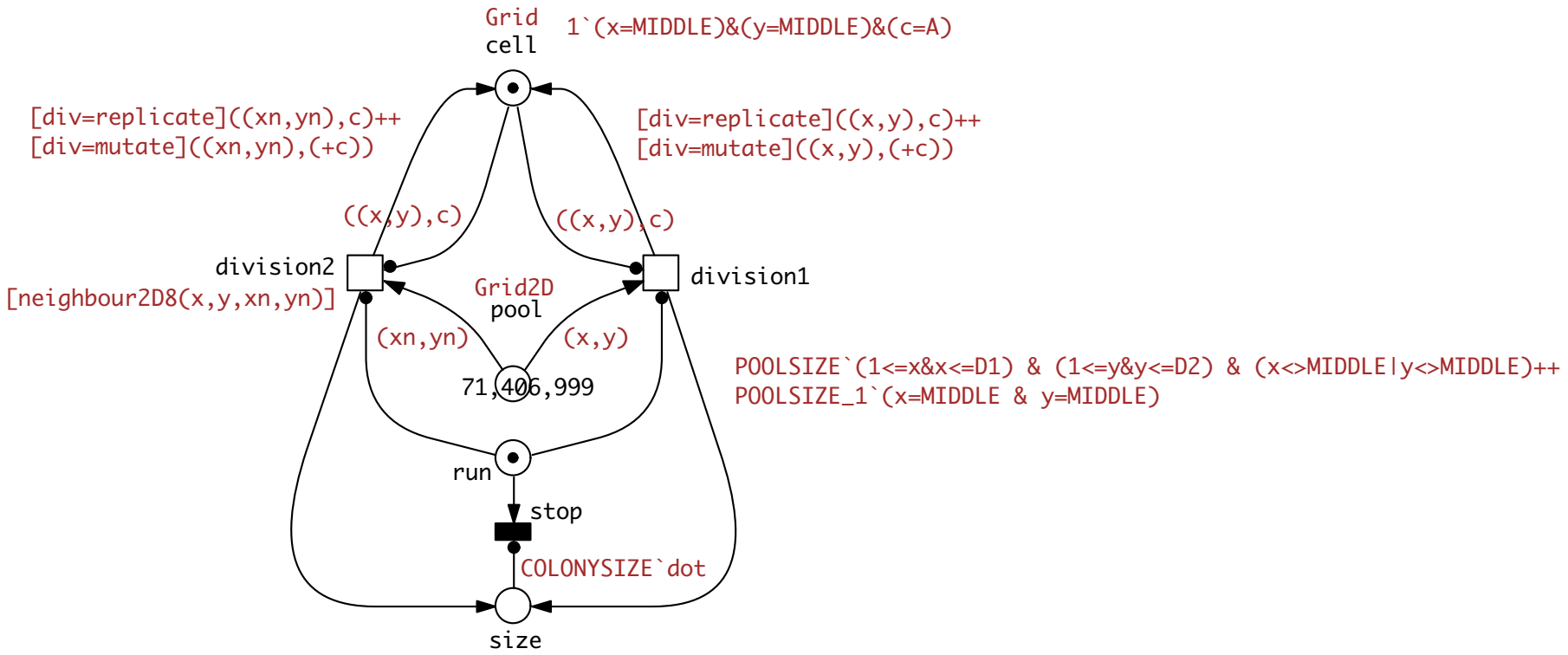
colorset Grid = **product with** Grid2D x Phenotype;



colorset Grid = **product with** Grid2D x Phenotype;



colorset Grid = **product with** Grid2D x Phenotype;



EX2: 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 - 24 h

-> *25 generations: 33.5 E+06*

-> *26 generations: 67 E+06*

-> *COLONYSIZE = 70,000,000*

grid size

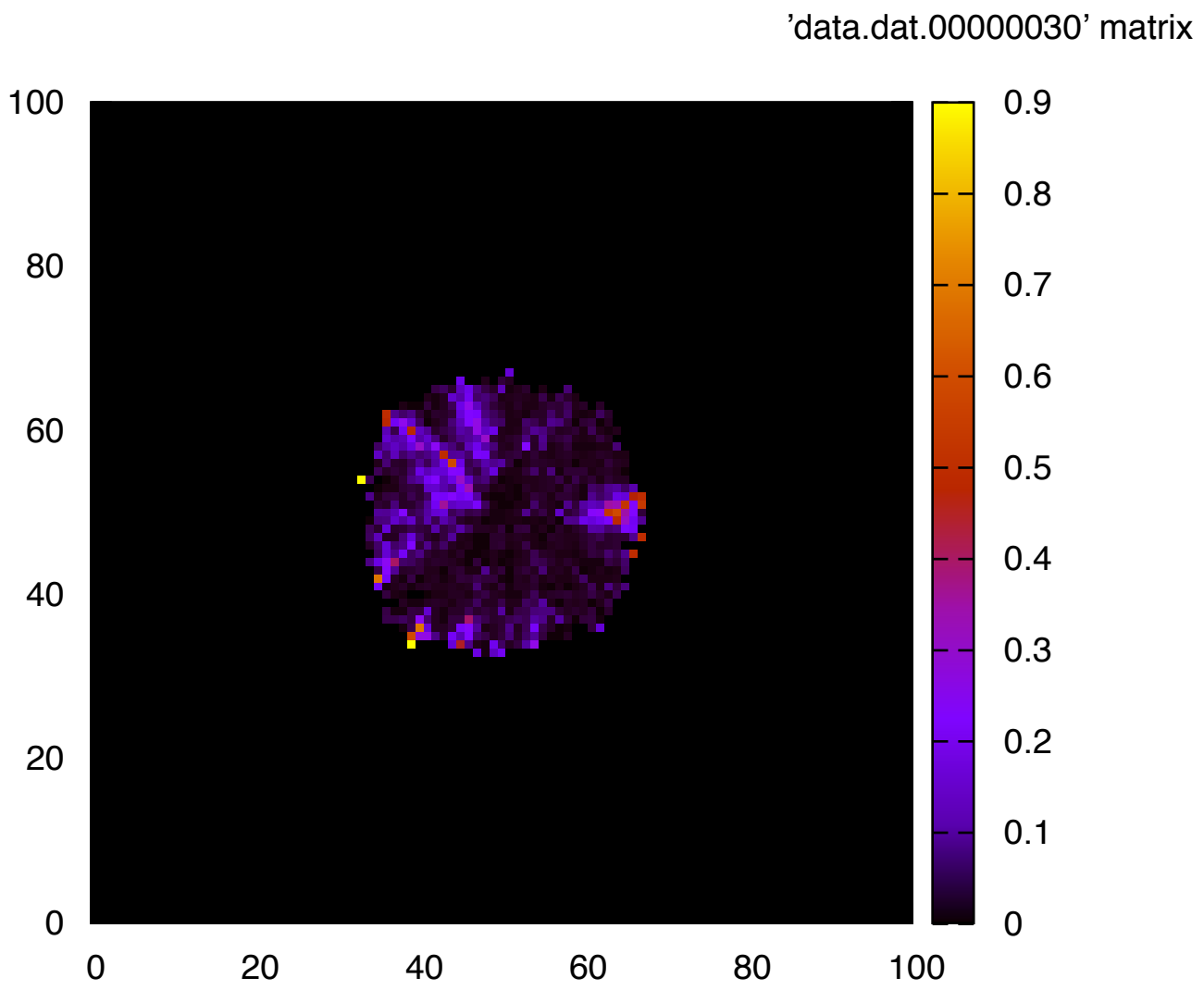
-> *61 x 61 grid:* *11,163 P / 131,044 T;* *unfolding: 152 sec;*

-> *101 x 101 grid:* *30,603 P / 362,404 T;* *unfolding: 9 min;*

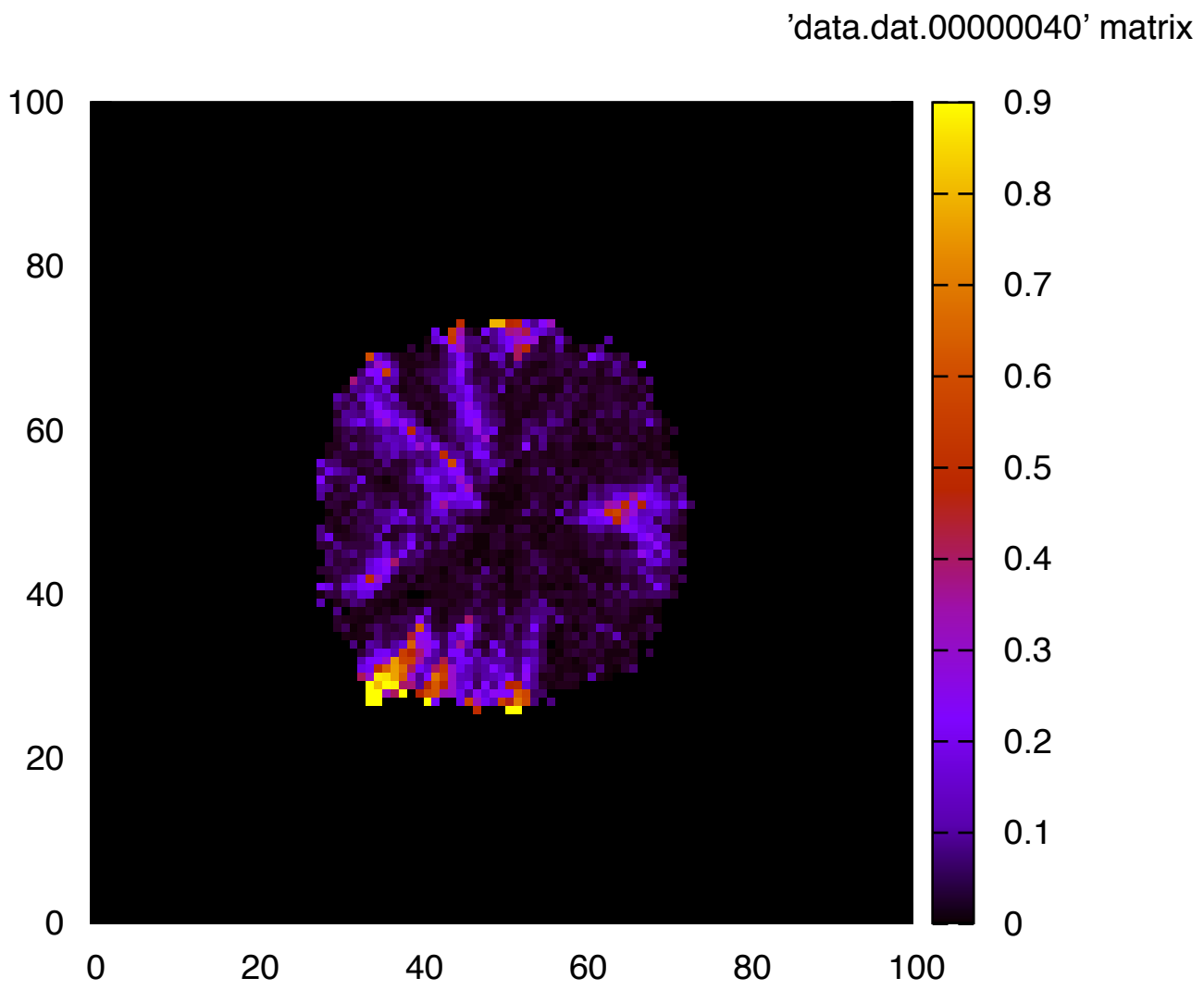
-> *runtime 1 stoch. simulation: 35-40 minutes*

. . . SOME EXPERIMENTS

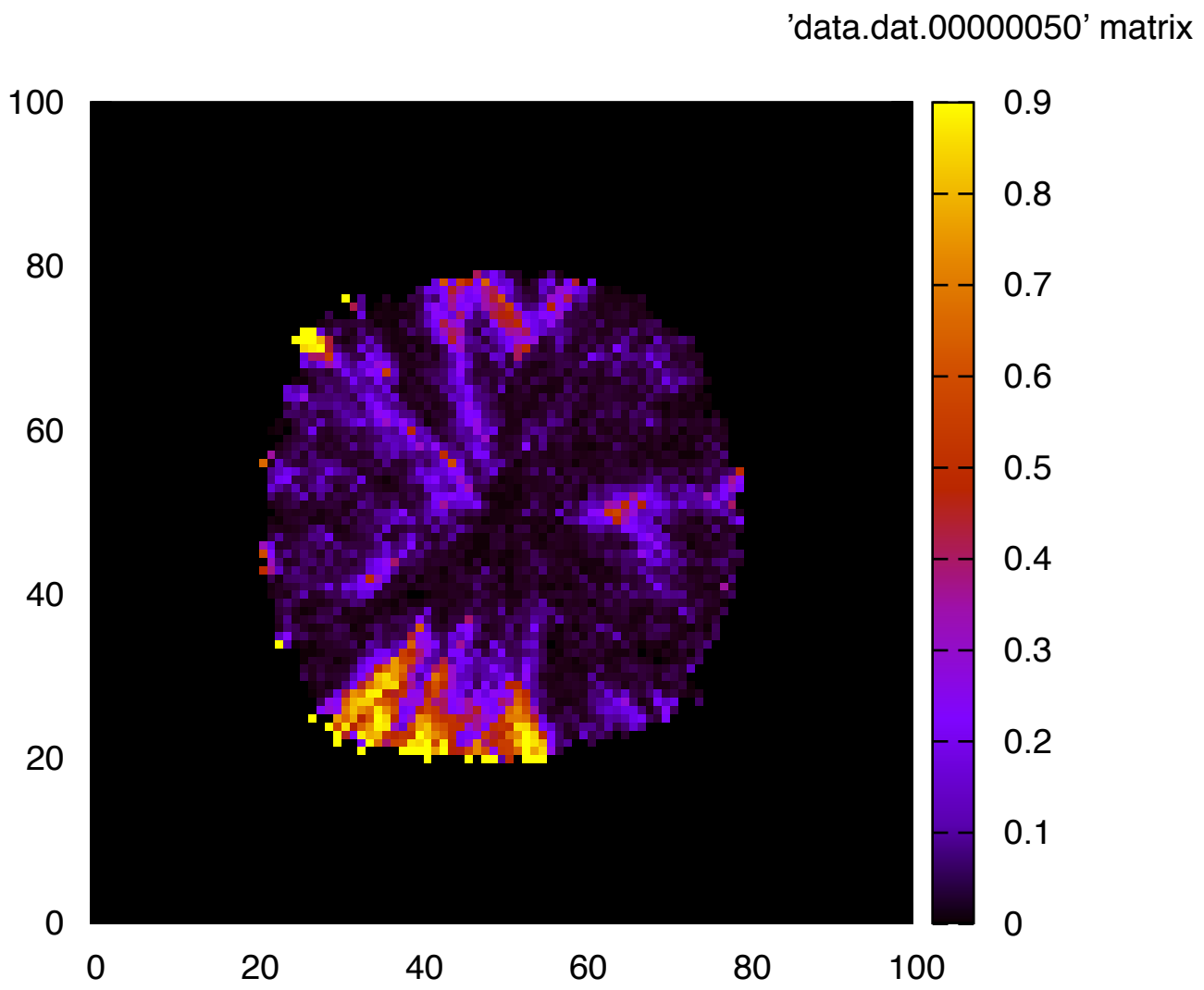
EX2: 2D - TRACE 1 (HIGH, F=1)



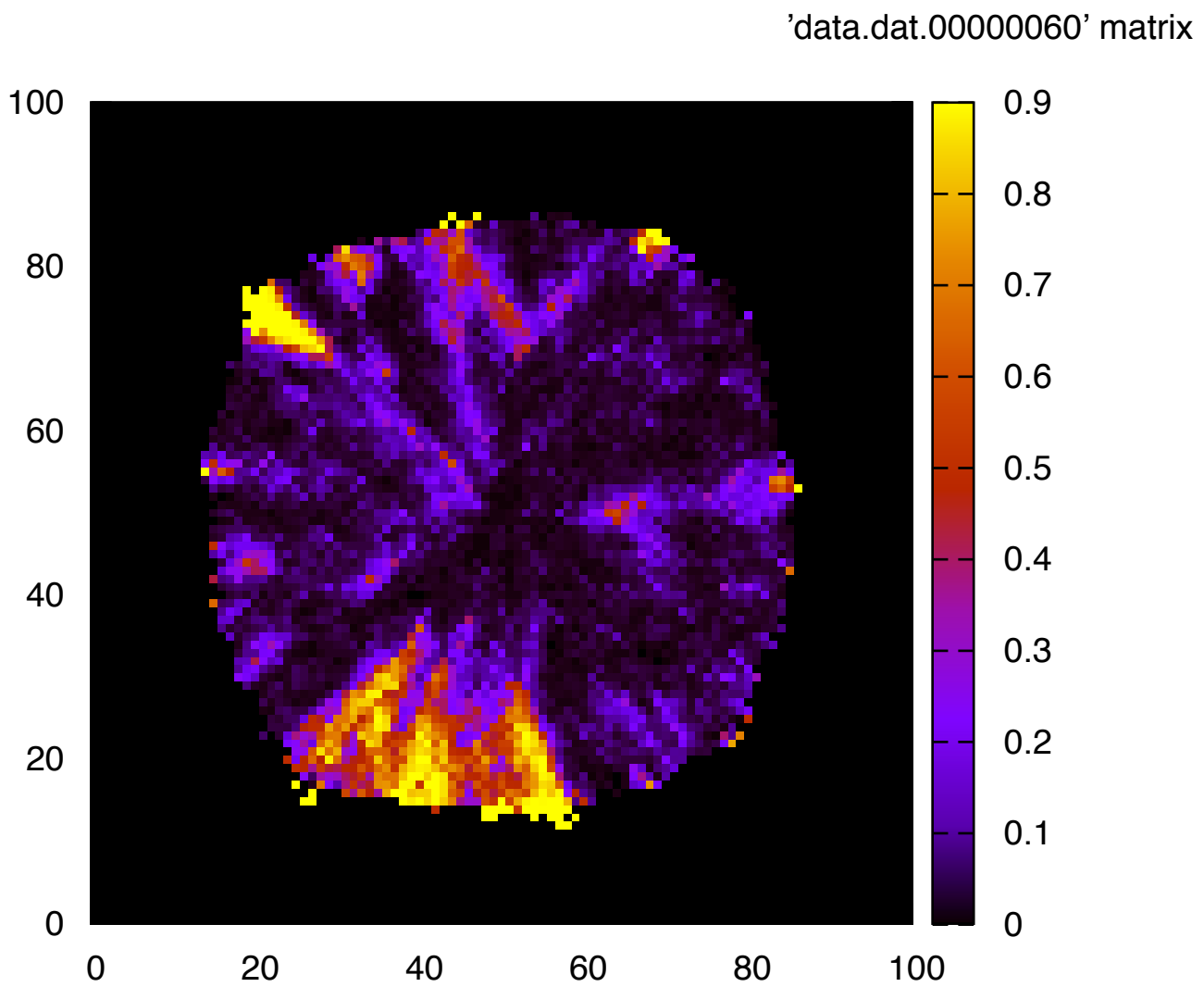
EX2: 2D - TRACE 1 (HIGH, F=1)



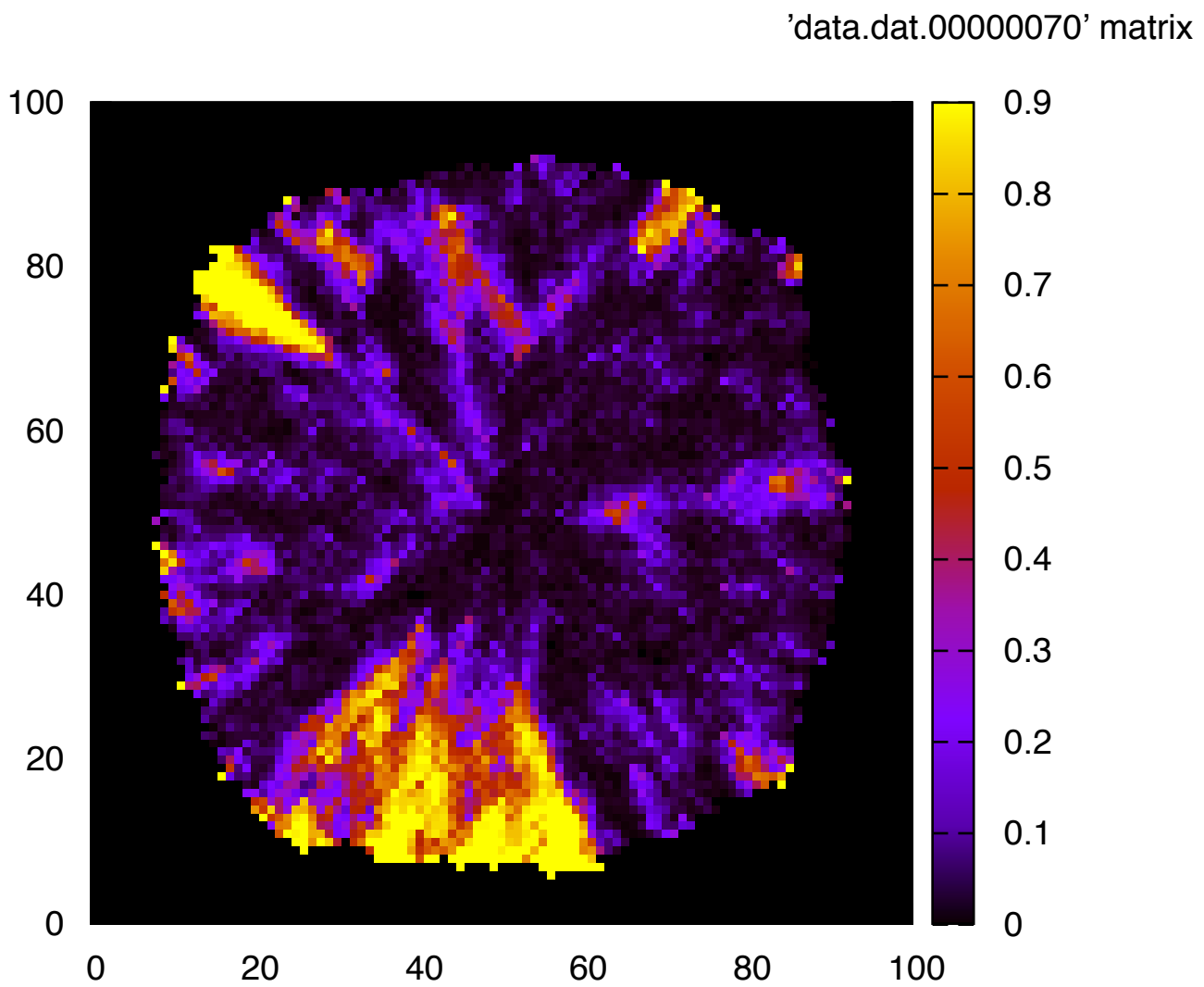
EX2: 2D - TRACE 1 (HIGH, F=1)



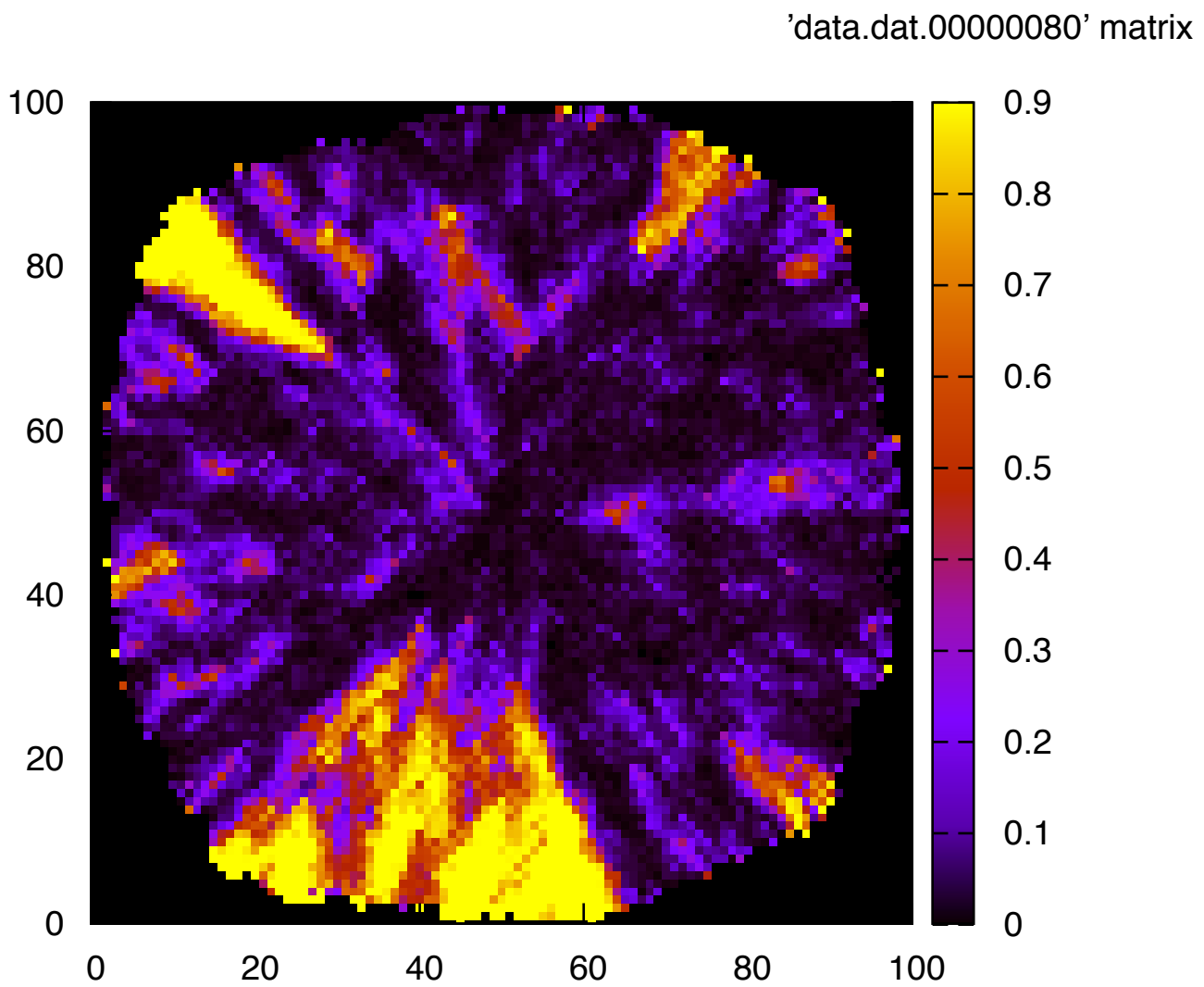
EX2: 2D - TRACE 1 (HIGH, F=1)



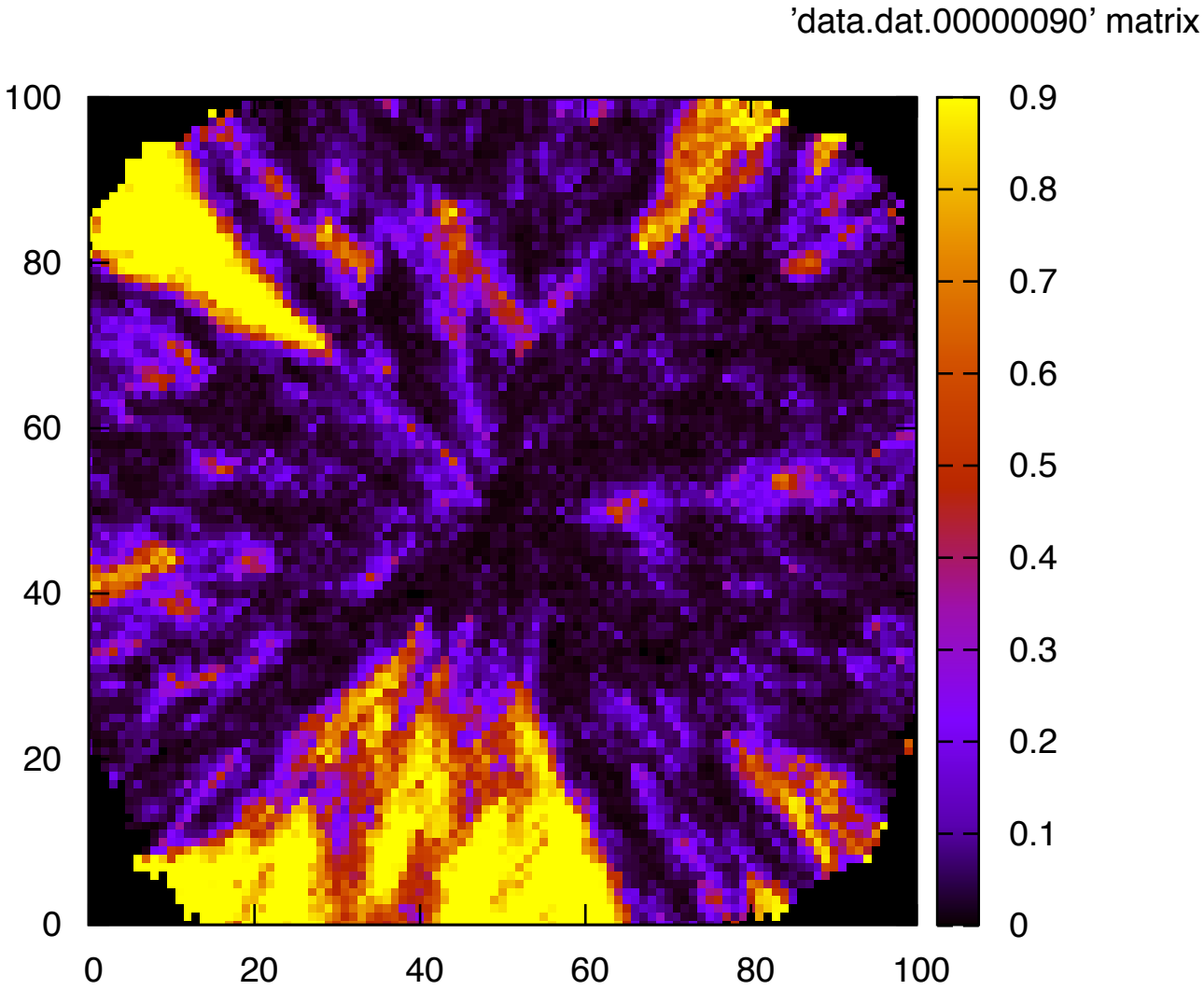
EX2: 2D - TRACE 1 (HIGH, F=1)



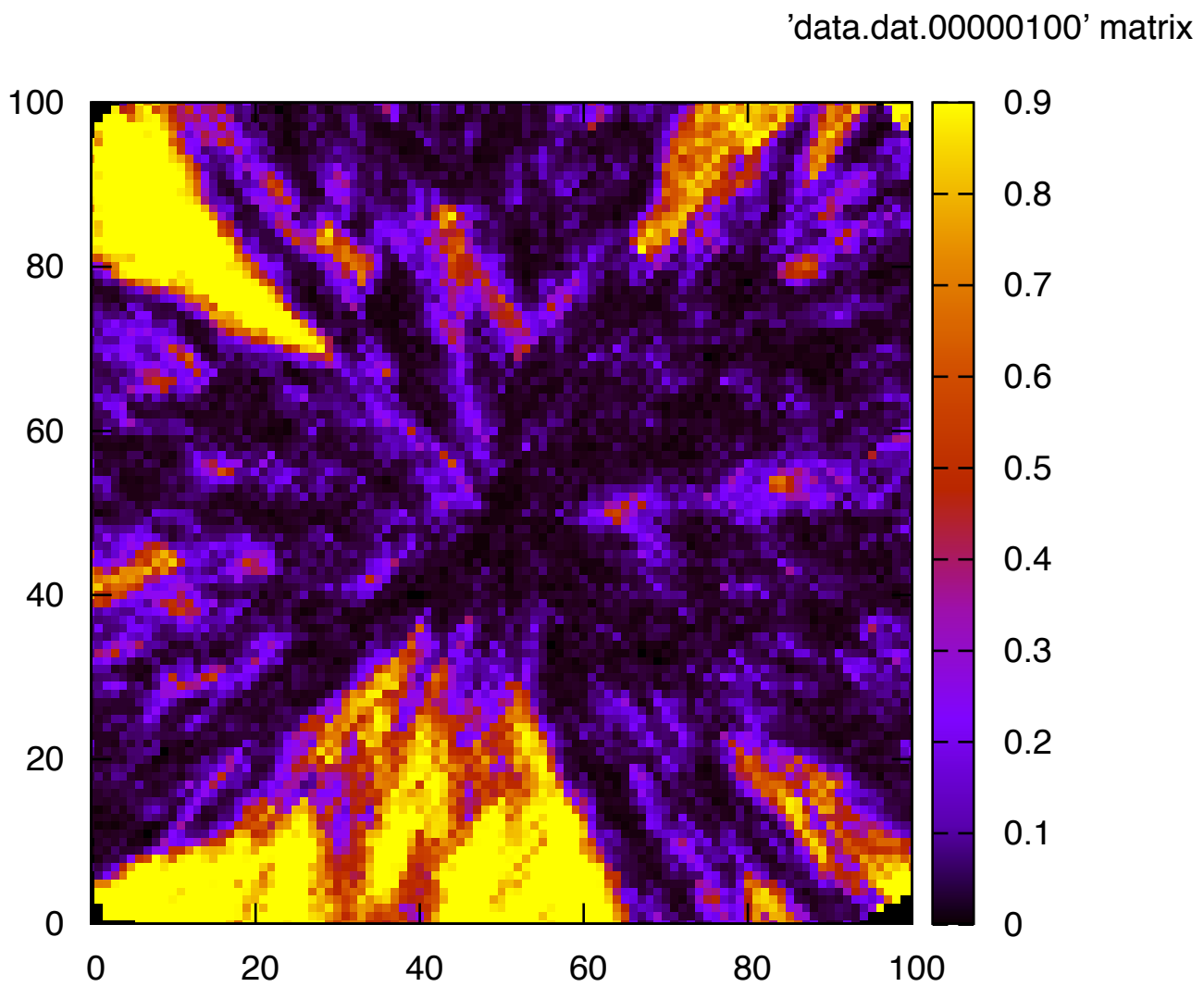
EX2: 2D - TRACE 1 (HIGH, F=1)



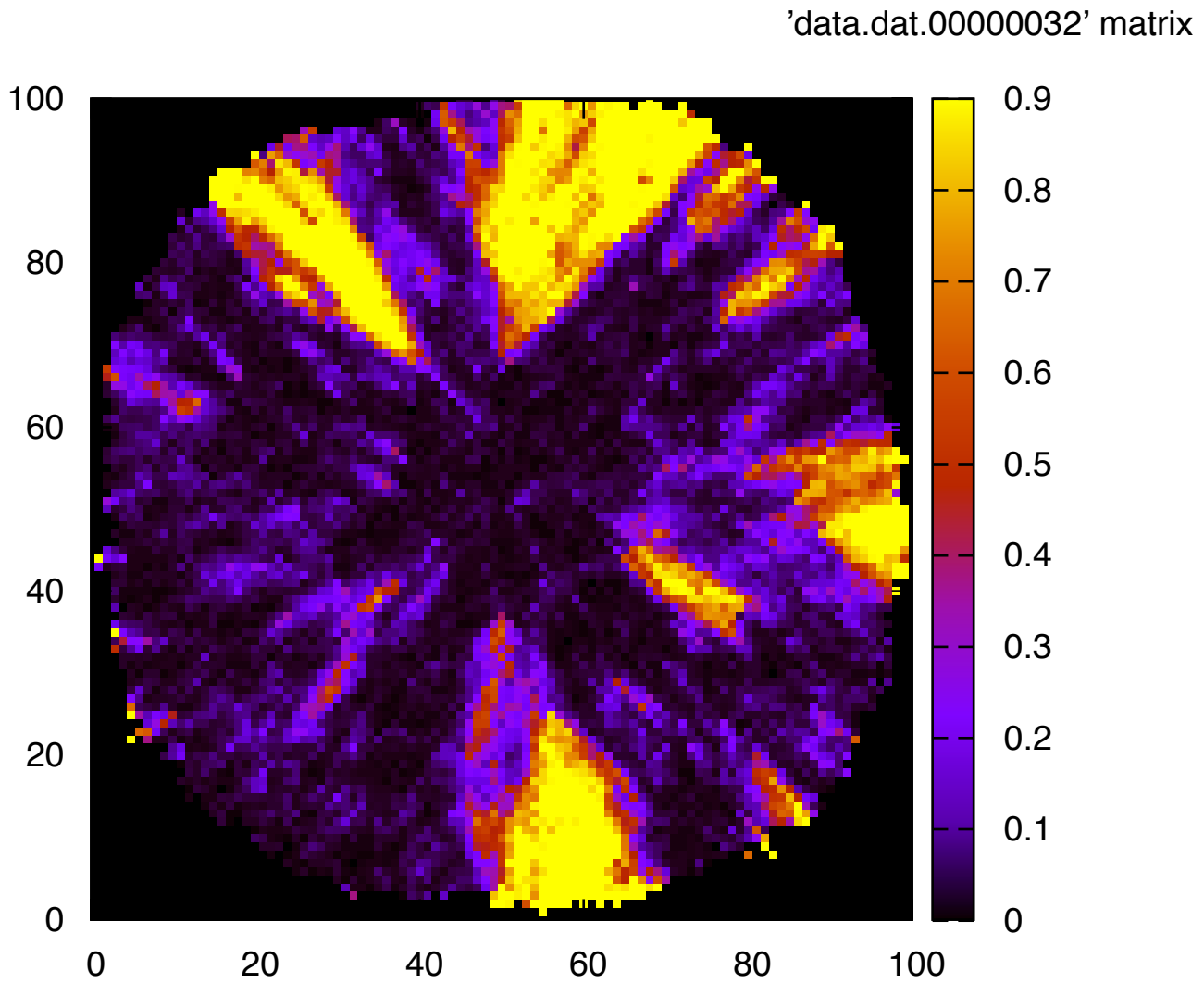
EX2: 2D - TRACE 1 (HIGH, F=1)



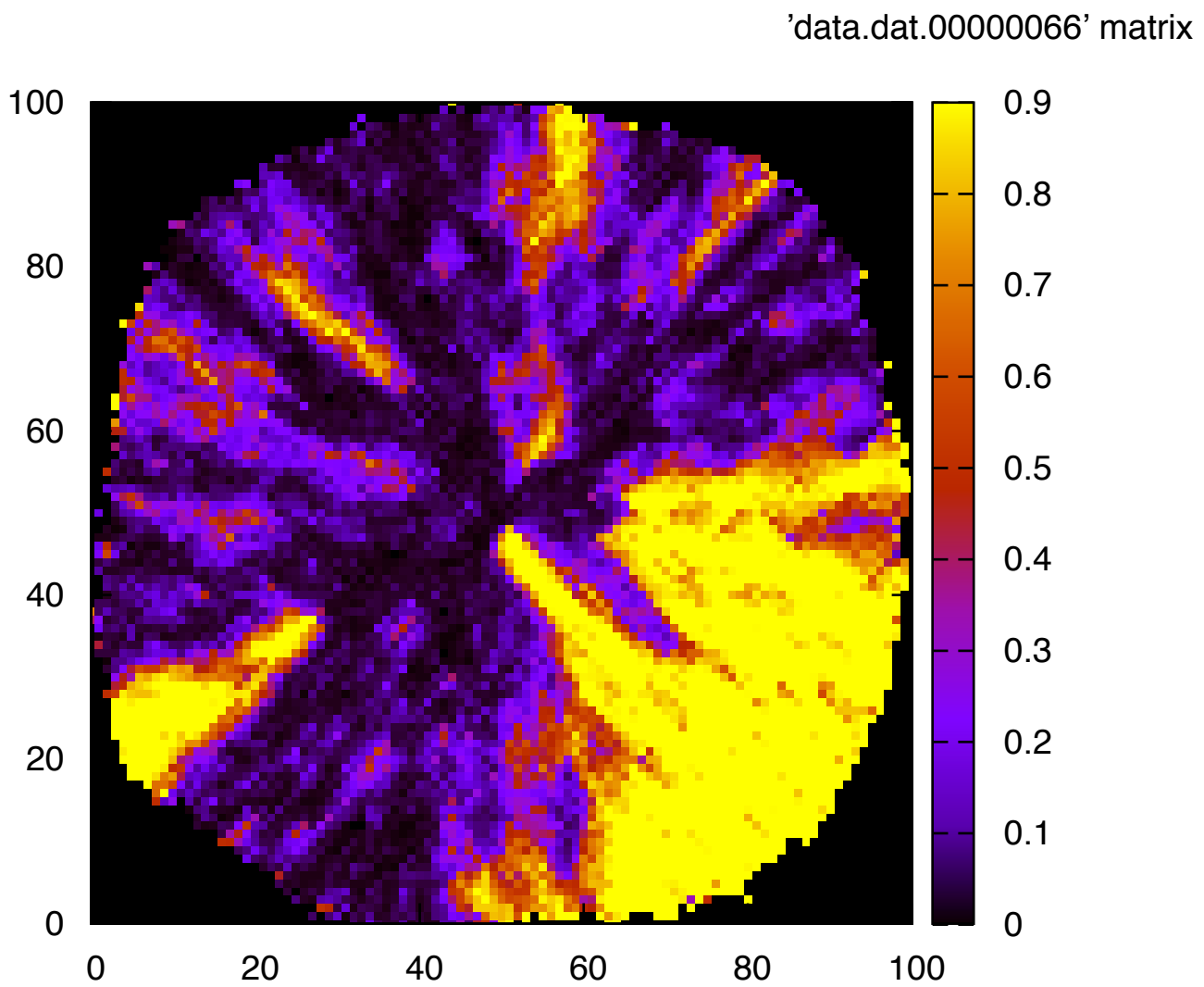
EX2: 2D - TRACE 1 (HIGH, F=1)



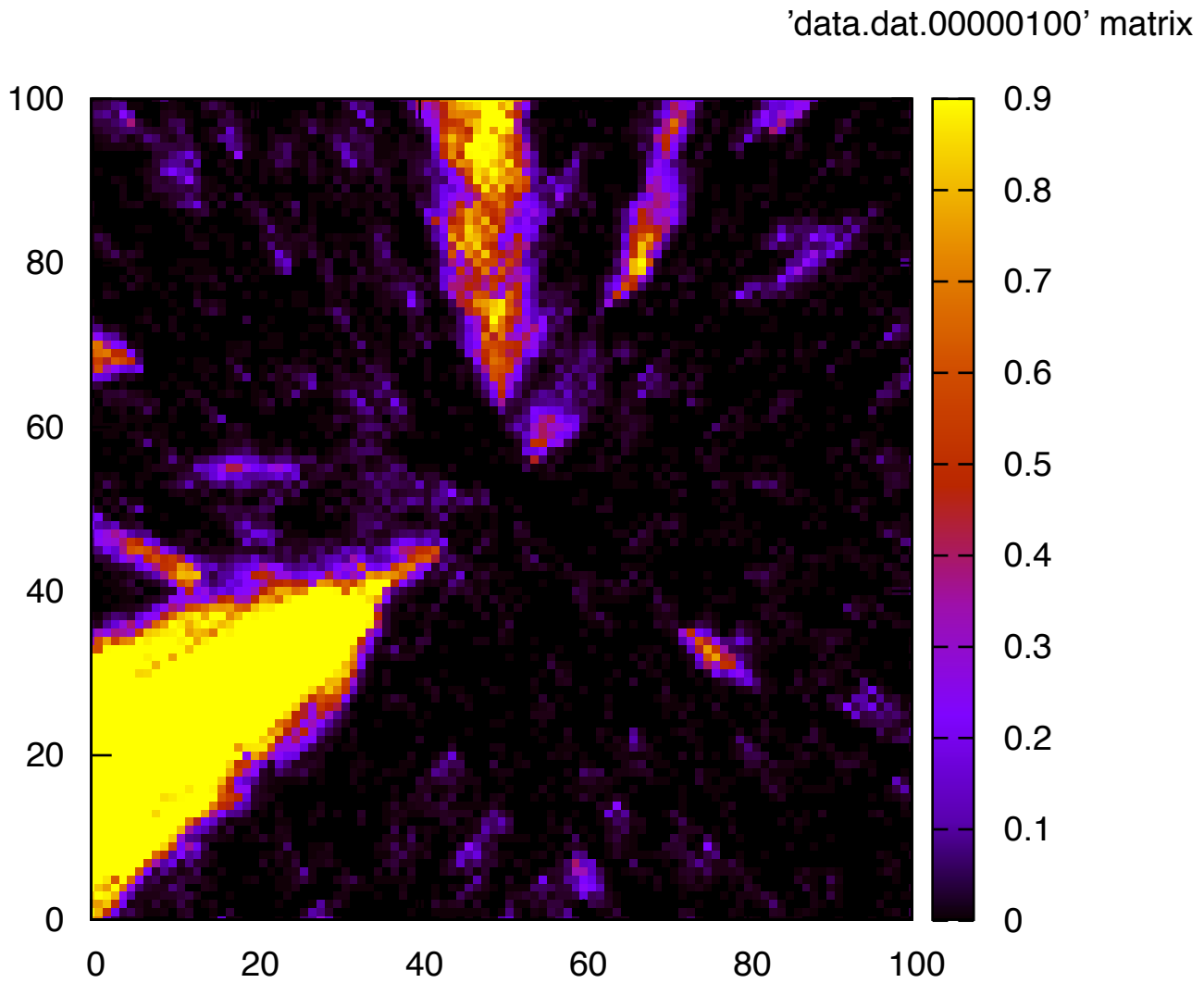
EX2: 2D - TRACE 2 (HIGH, F=1)



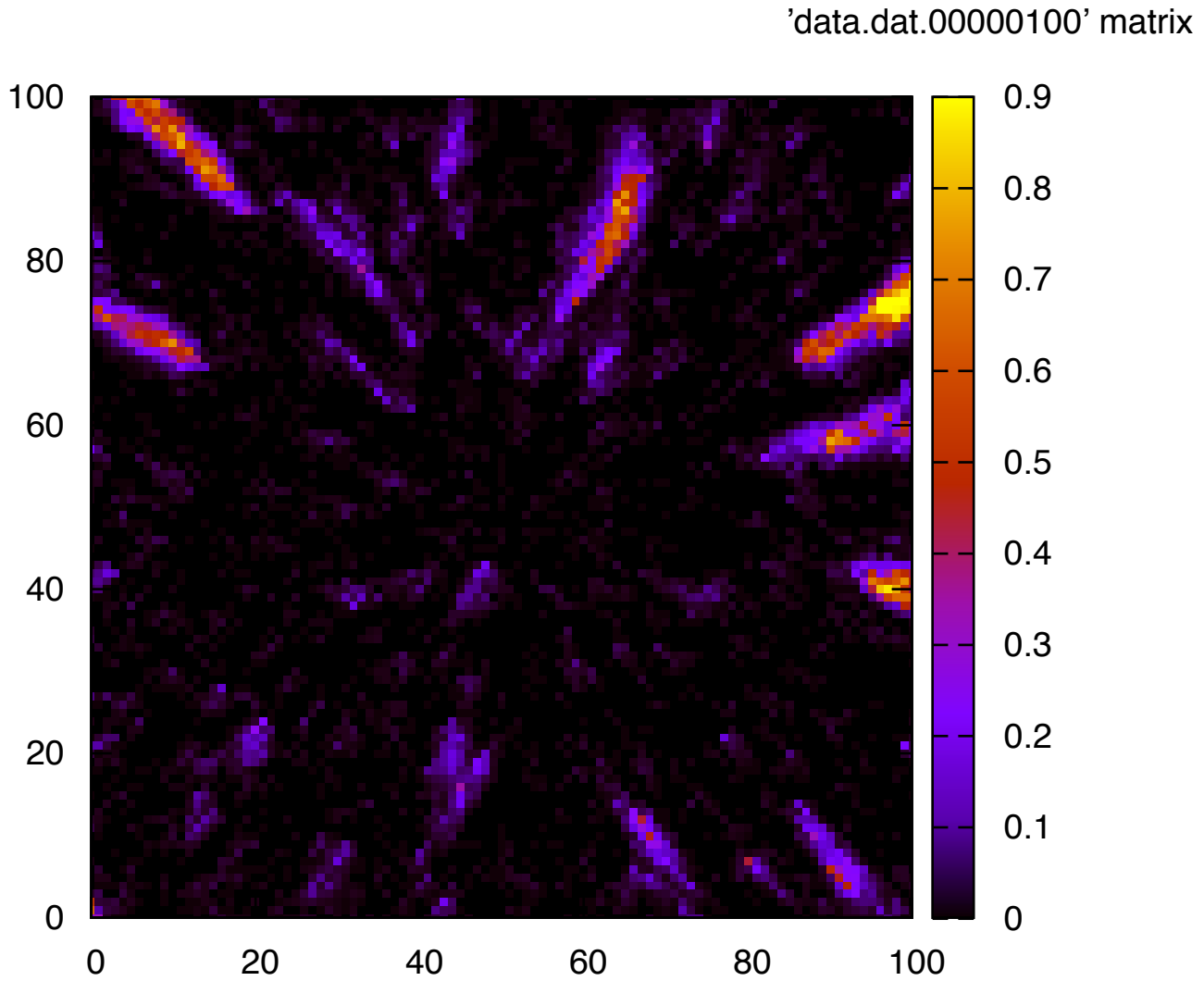
EX2: 2D - TRACE 3 (HIGH, F=1)



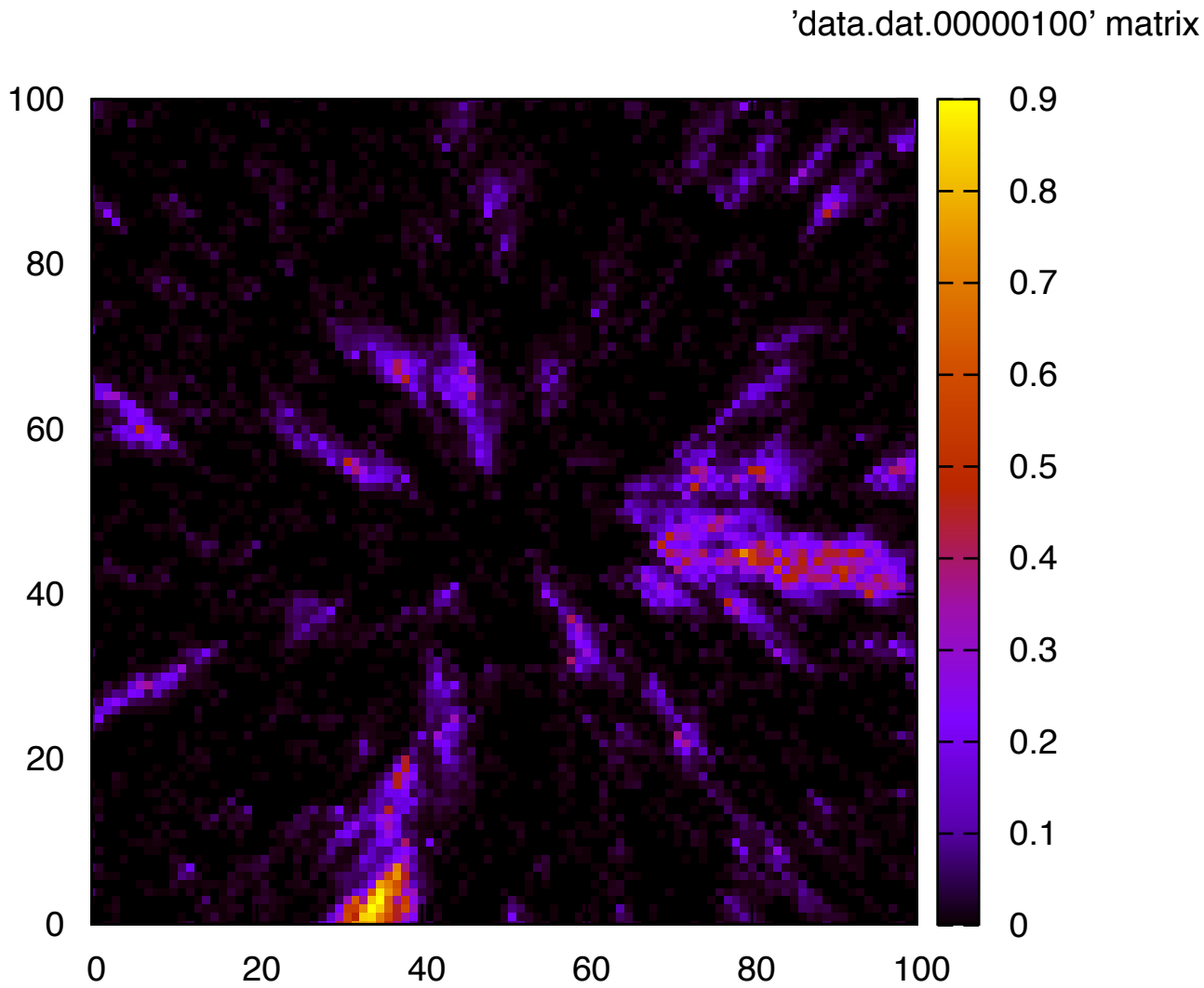
EX2: 2D - VARYING FITNESS, TRACE 1 (MEDIUM, F=1)



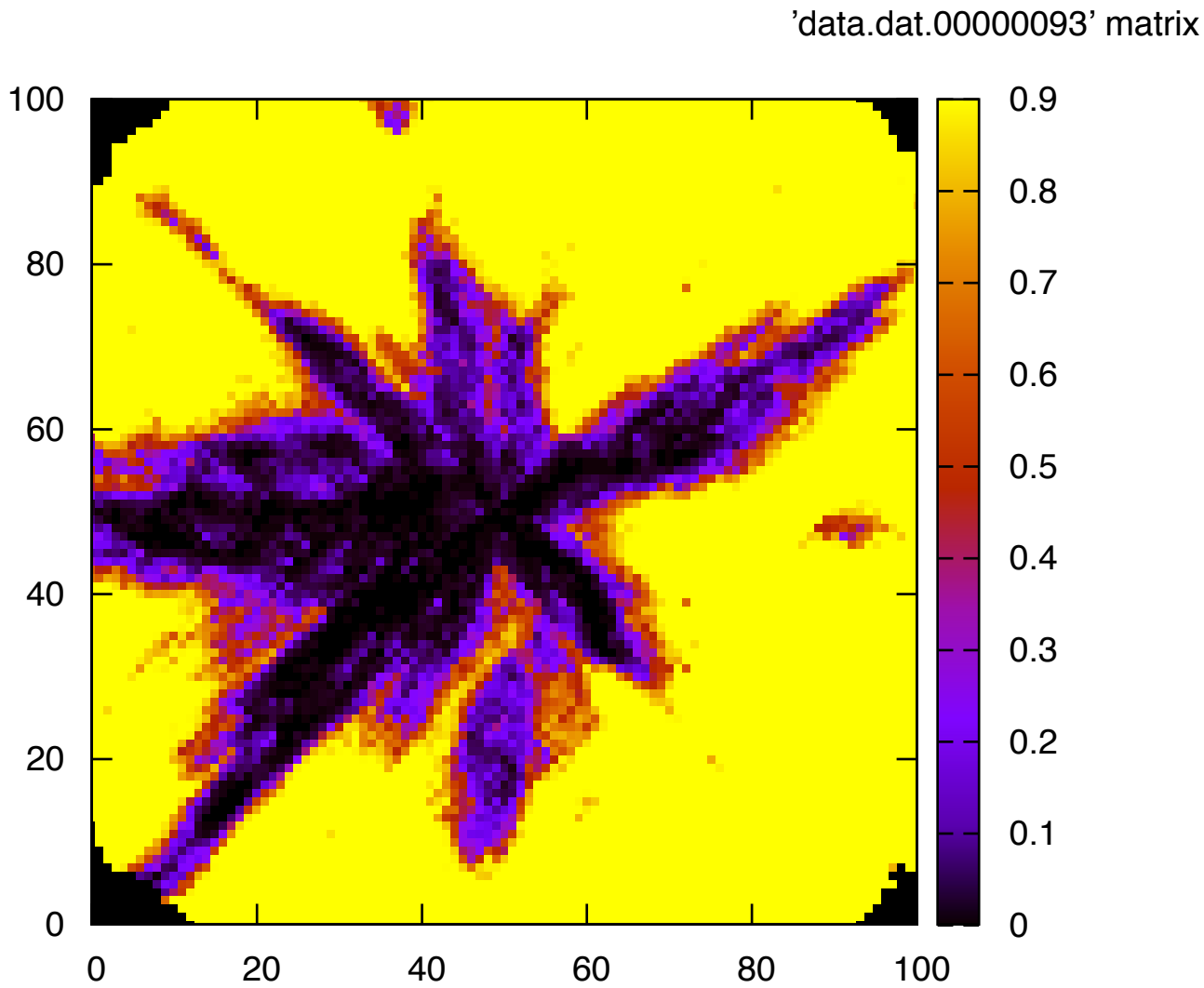
Ex2: 2D - Varying Fitness, Trace 2 (Medium, F=1)



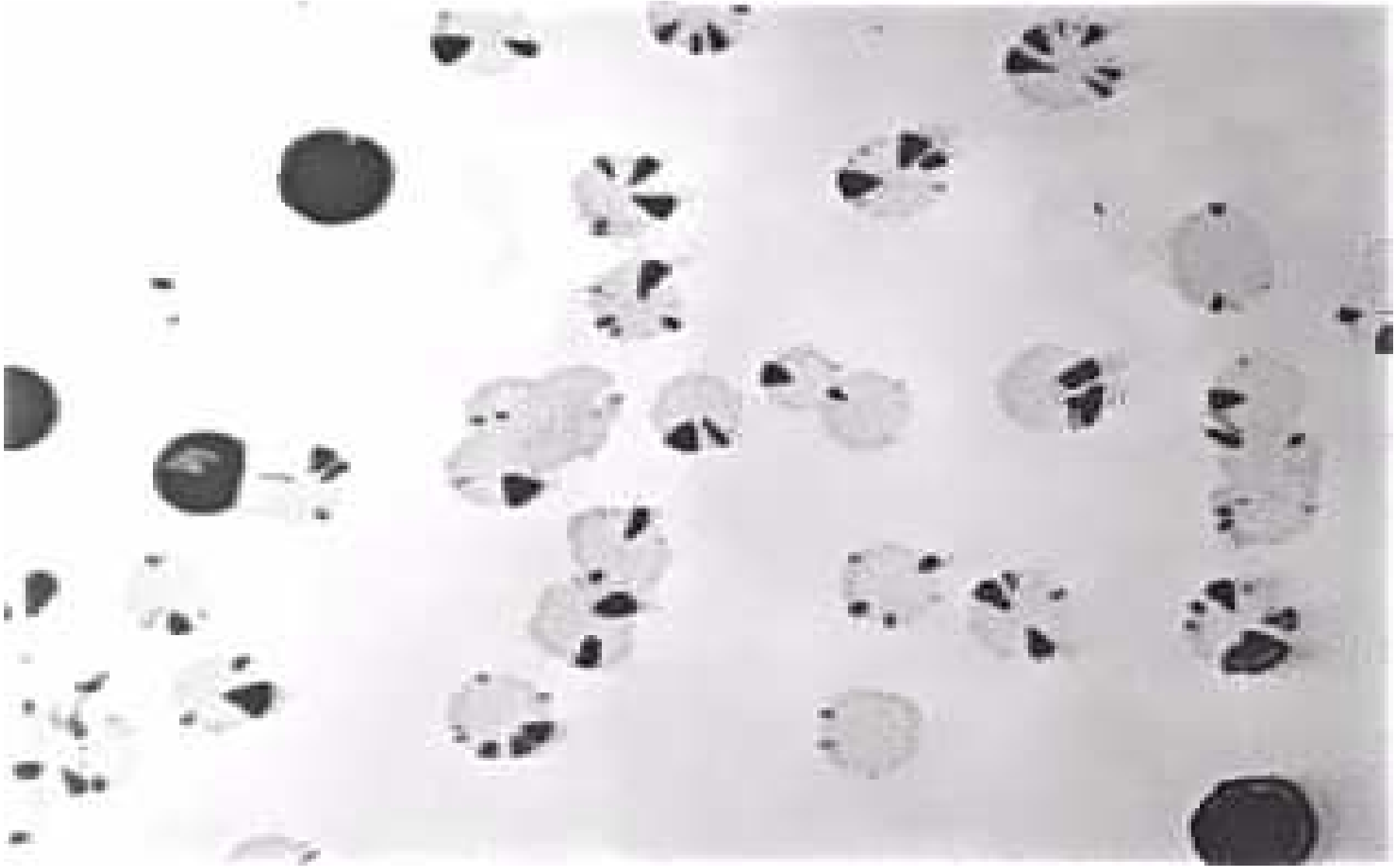
EX2: 2D - VARYING FITNESS, TRACE 1 (MEDIUM, F=0.99)



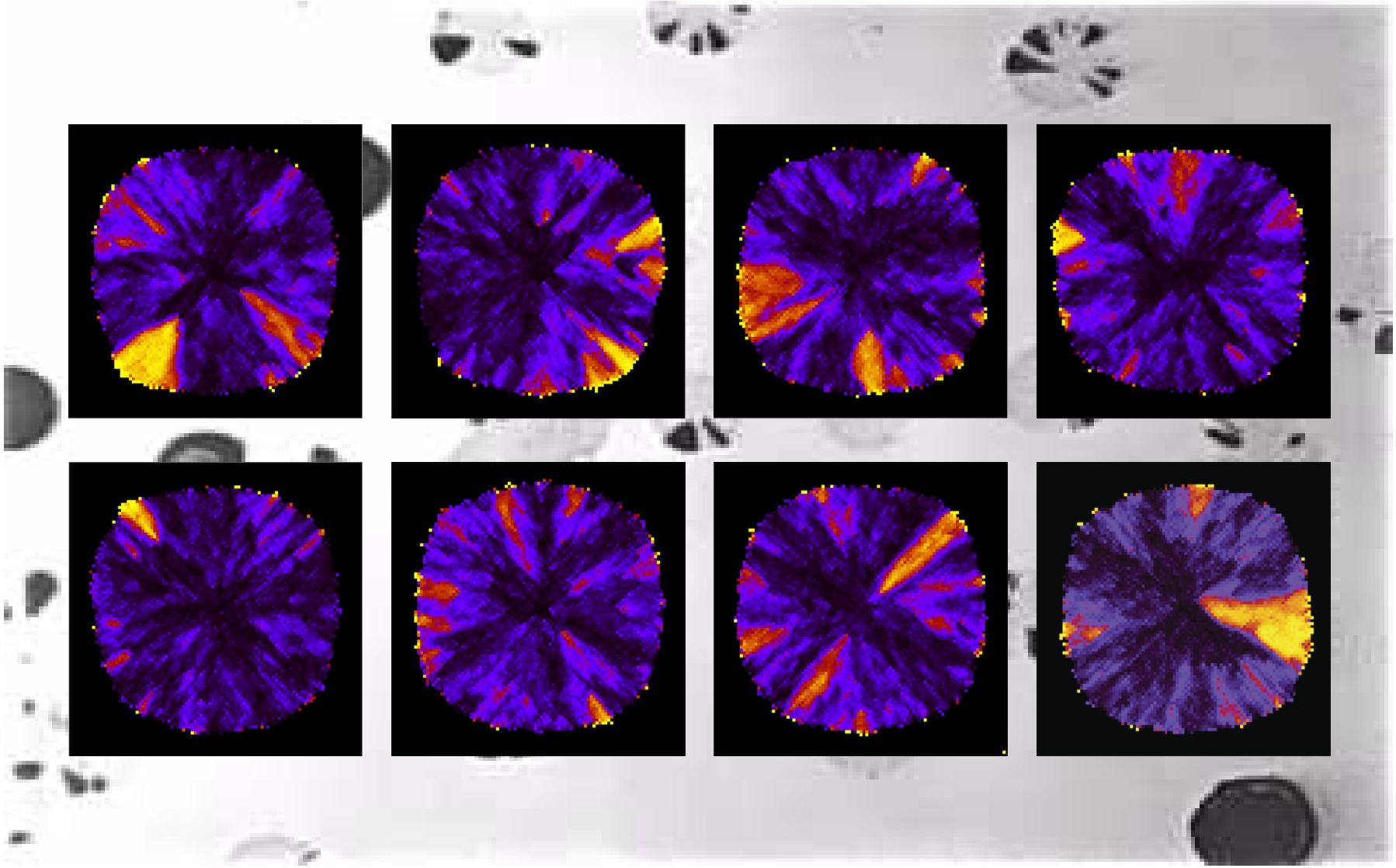
EX2: 2D - VARYING FITNESS, TRACE 1 (MEDIUM, F=0.90)



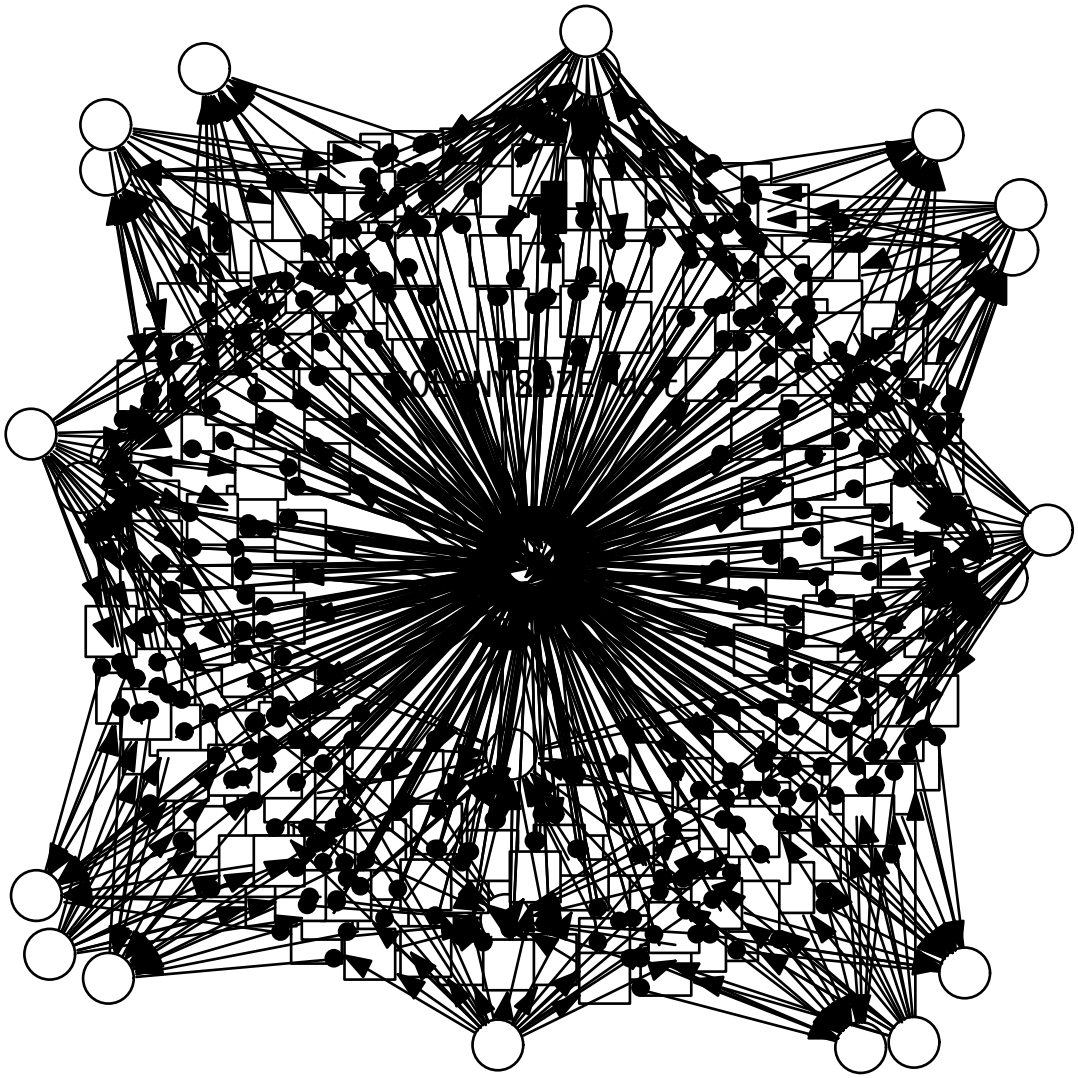
EX2: SOME FINAL STATES (HIGH, F=1)



EX2: SOME FINAL STATES (HIGH, F=1)



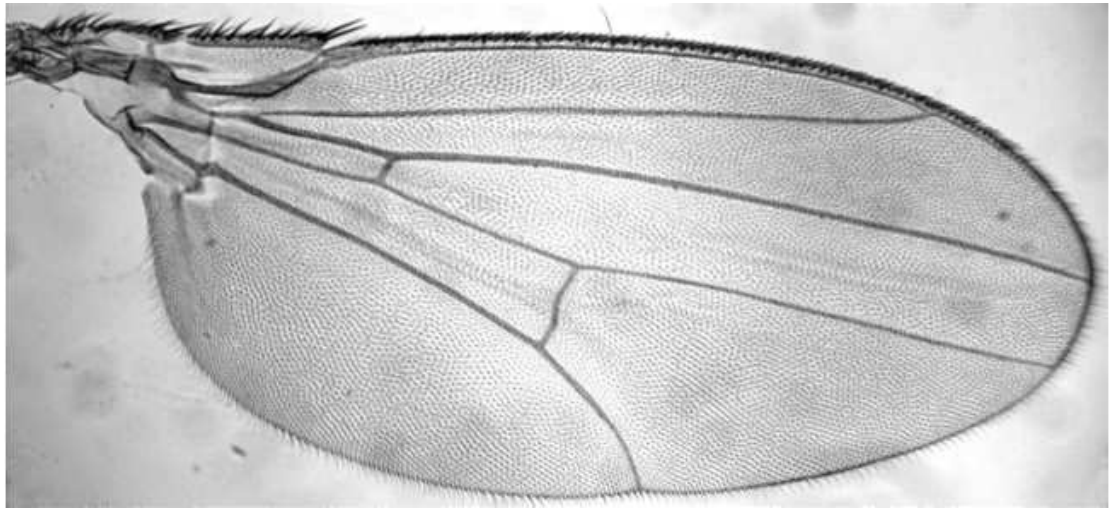
PHASE VARIATION, PLAIN MODEL (3X3)

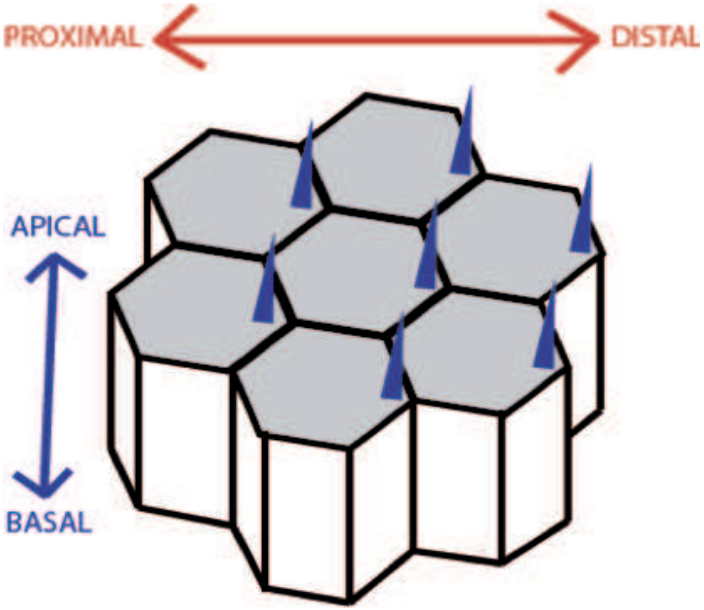


EXAMPLE 3:

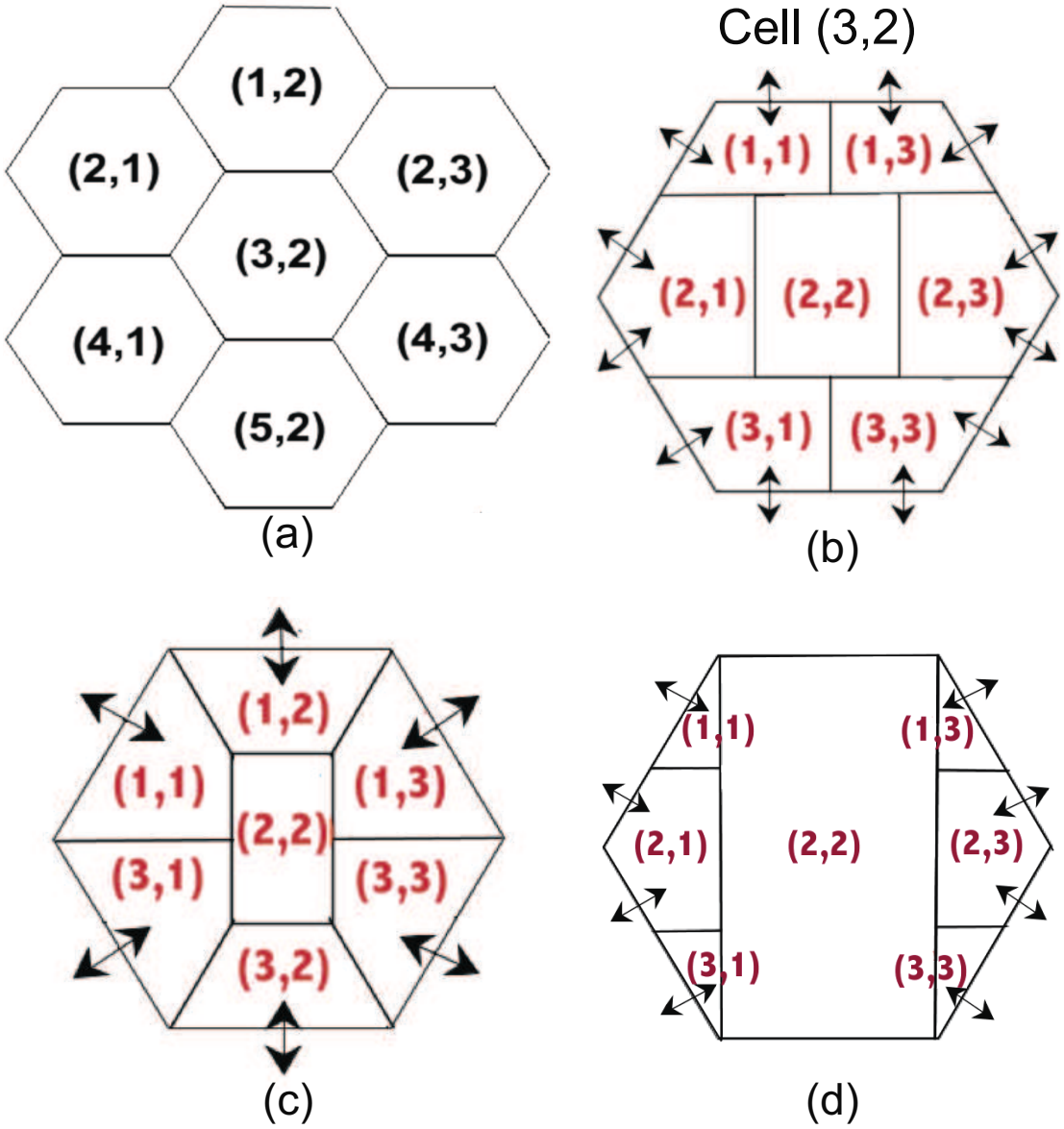
PLANAR CELL POLARITY IN FLY WING

EX3 - PLANAR CELL POLARITY

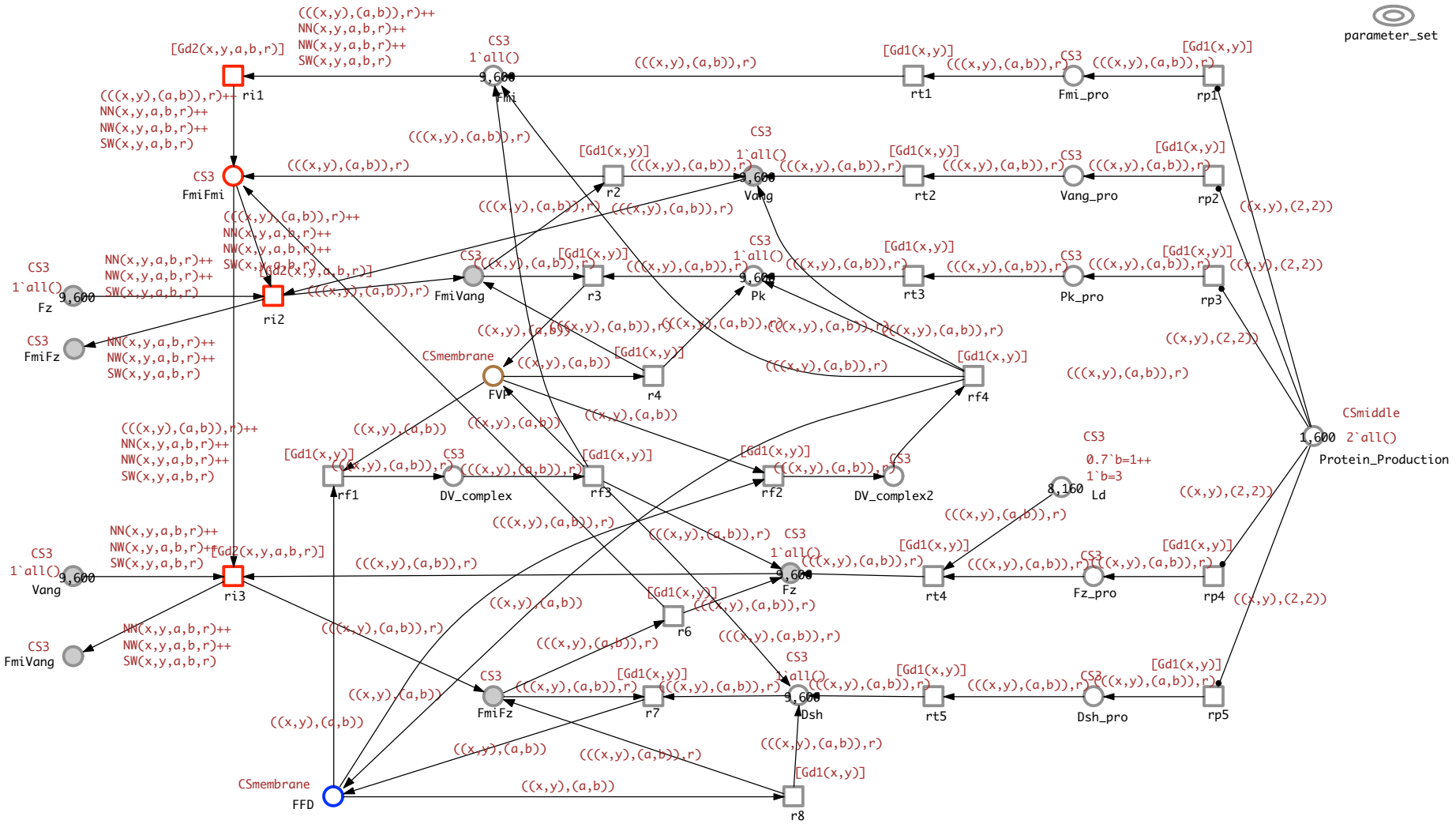




[BIOPPN 2011]
[CMSB 2011]

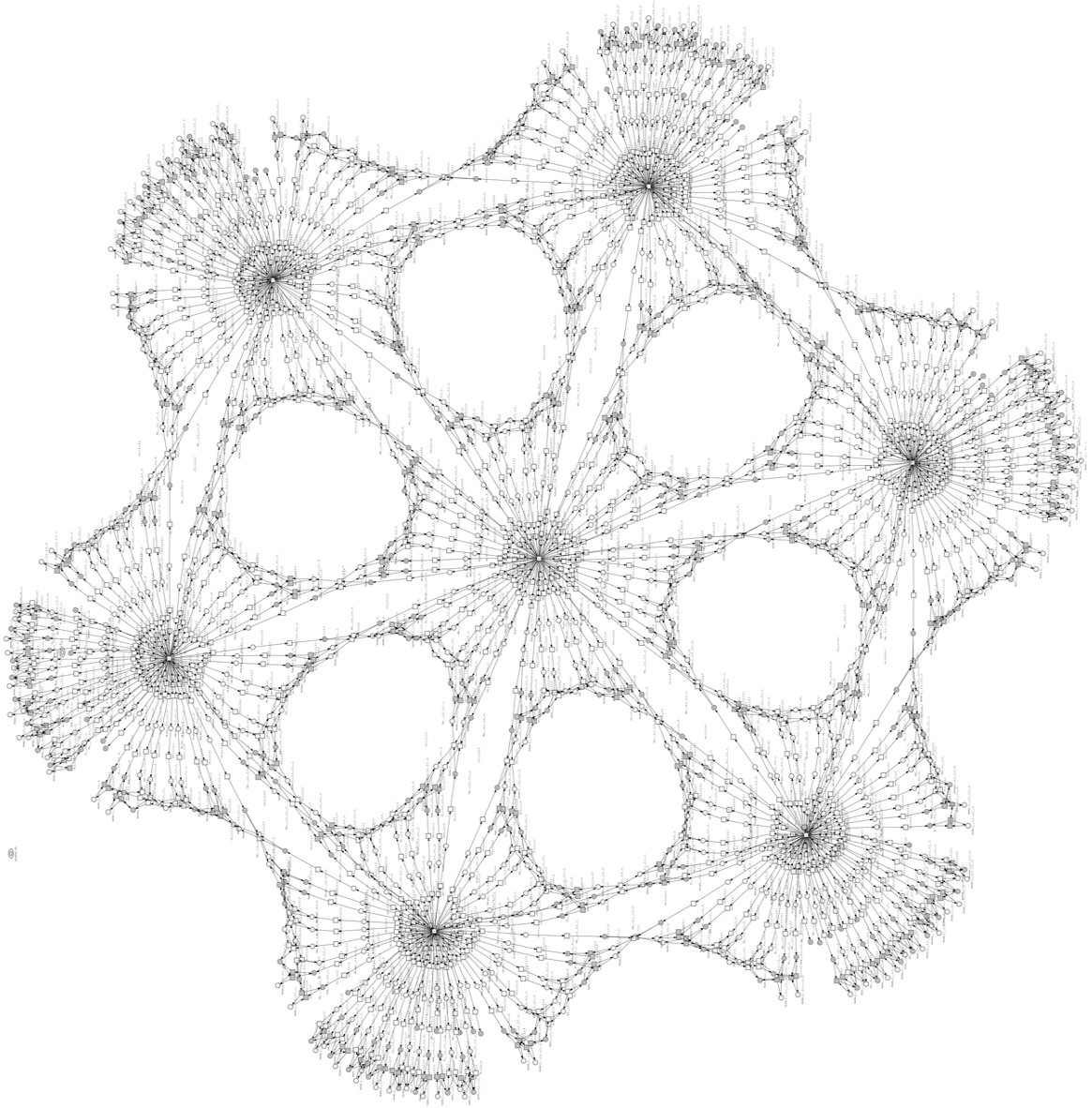

(1,2)
(2,1)
(2,3)
(3,2)
(4,1)
(4,3)
(5,2)
(a)
Cell (3,2)
(1,1)
(1,3)
(2,1)
(2,2)
(2,3)
(3,1)
(3,3)
(b)
(1,2)
(1,1)
(1,3)
(2,2)
(3,1)
(3,3)
(3,2)
(c)
(1,1)
(1,3)
(2,1)
(2,2)
(2,3)
(3,1)
(3,3)
(d)

EX3: PLANAR CELL POLARITY



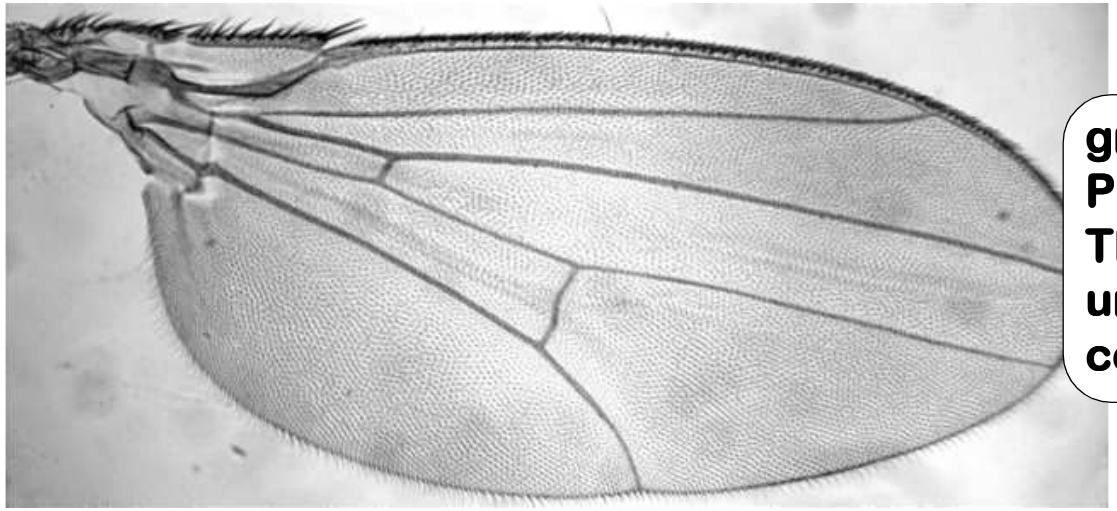
[QIAN GAO, PHD THESIS 2013]

EX3: PLANAR CELL POLARITY, PLAIN MODEL (7 CELLS)

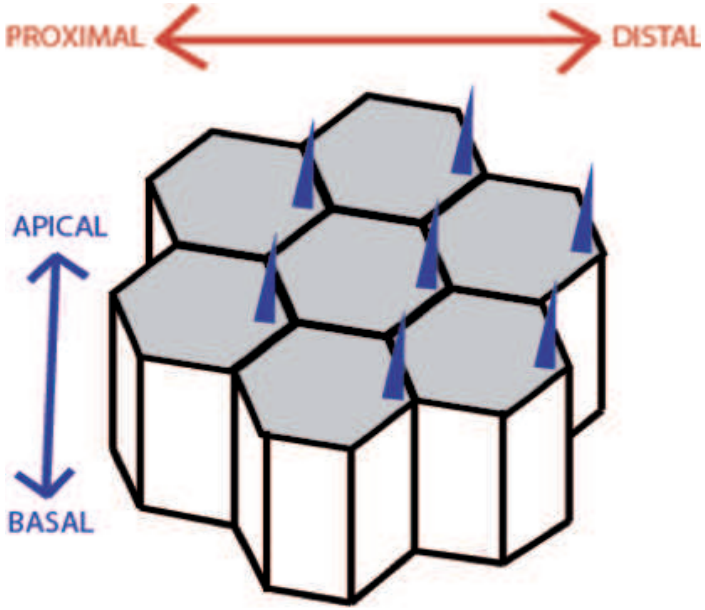


monika.heiner@b-tu.de

EX3 - PLANAR CELL POLARITY



grid size:	40 x 40
PLACES:	164,000
TRANSITIONS:	229,686
unfolding:	4 min
cont. simulation:	2 h

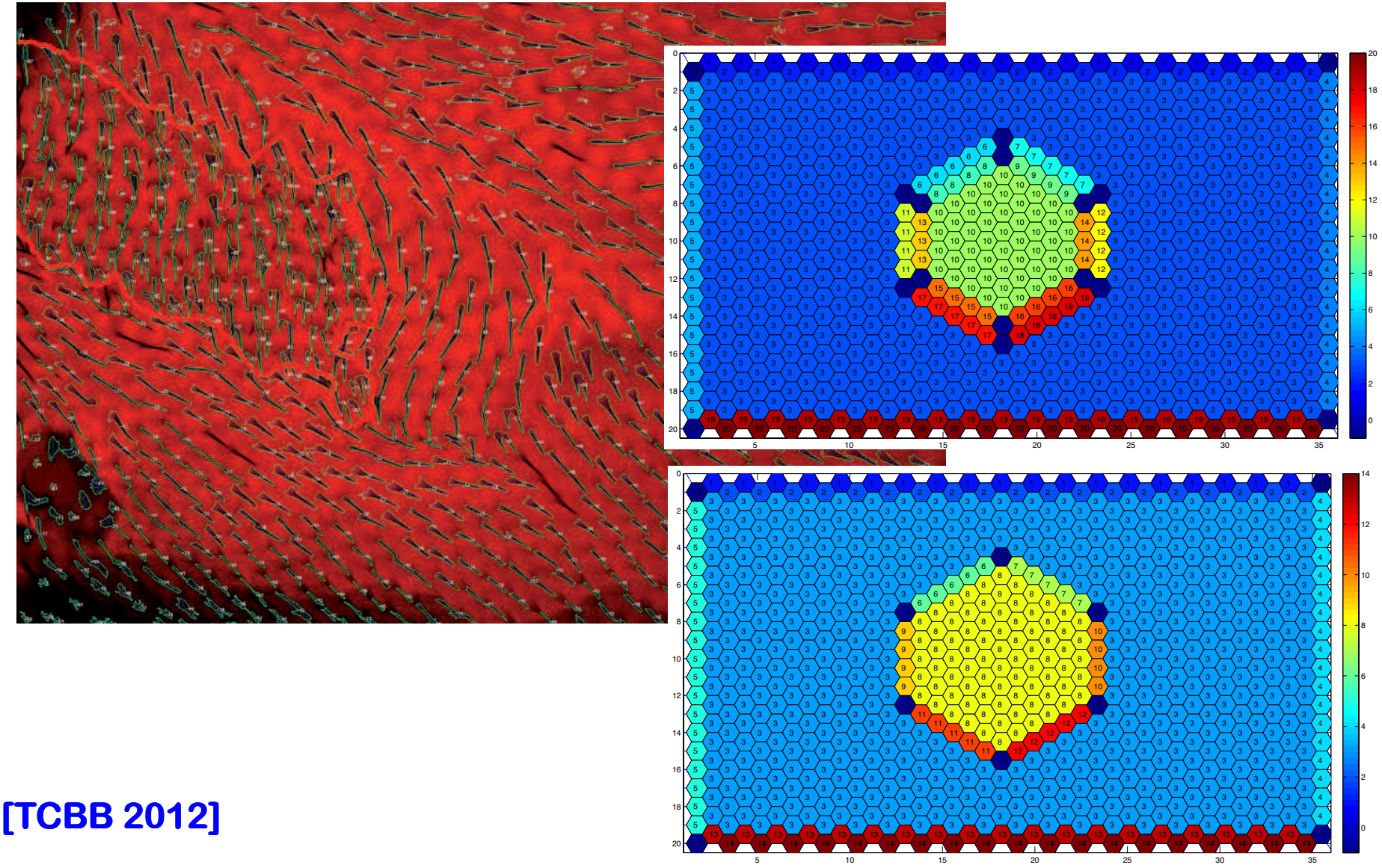


[BIOPPN 2011]
[CMSB 2011]

EX3 - PLANAR CELL POLARITY

monika.heiner@b-tu.de

EX3 - PLANAR CELL POLARITY

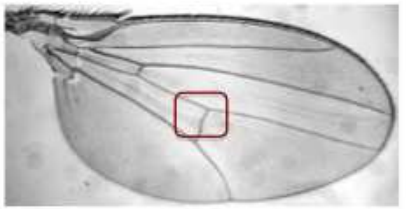


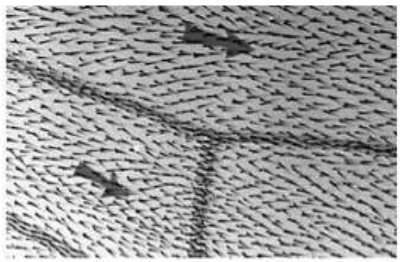
[TCBB 2012]

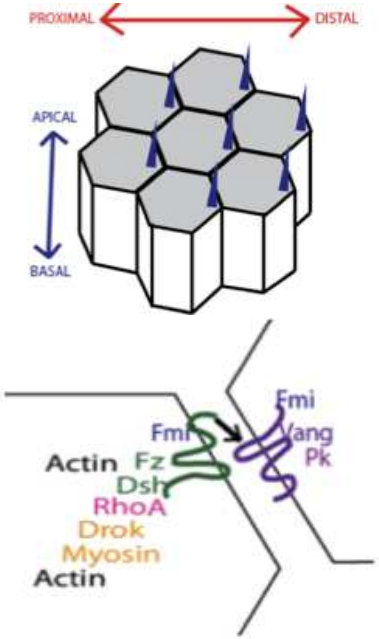
monika.heiner@b-tu.de

SUMMARY

- **FRAMEWORK**
  - -> *QPN - SPN - CPN - HPN*
  - -> *uncoloured / coloured*
- **MODELLING BIO PETRI NETS**
- **COLOUR**
  - -> *gradients*
  - -> *phase variation*
  - -> *fly wing*
- **WHAT NEXT ?**

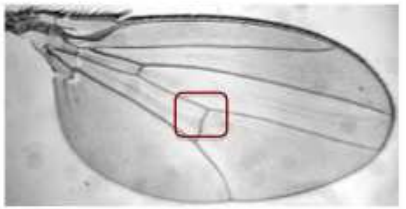


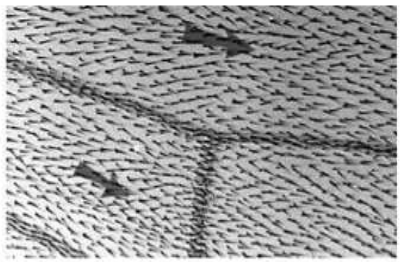


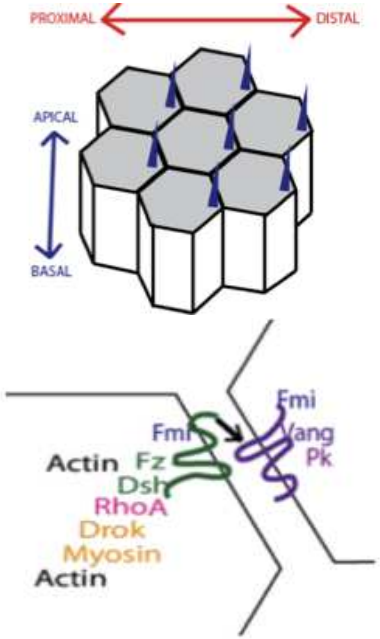


SUMMARY

- **FRAMEWORK**
  - -> *QPN - SPN - CPN - HPN*
  - -> *uncoloured / coloured*
- **MODELLING BIO PETRI NETS**
- **COLOUR**
  - -> *gradients*
  - -> *phase variation*
  - -> *fly wing*
- **WHAT NEXT ?** -> **MODELLING 4 ANALYSING**

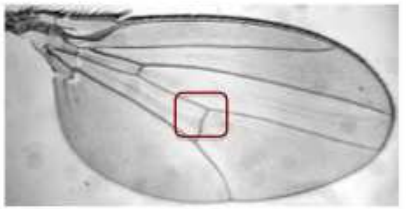


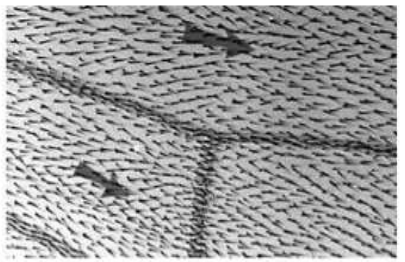


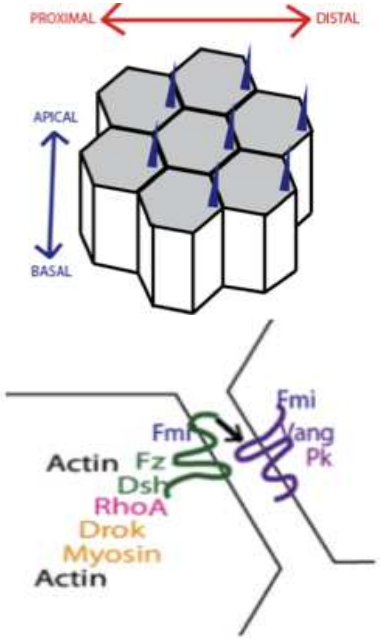


SUMMARY

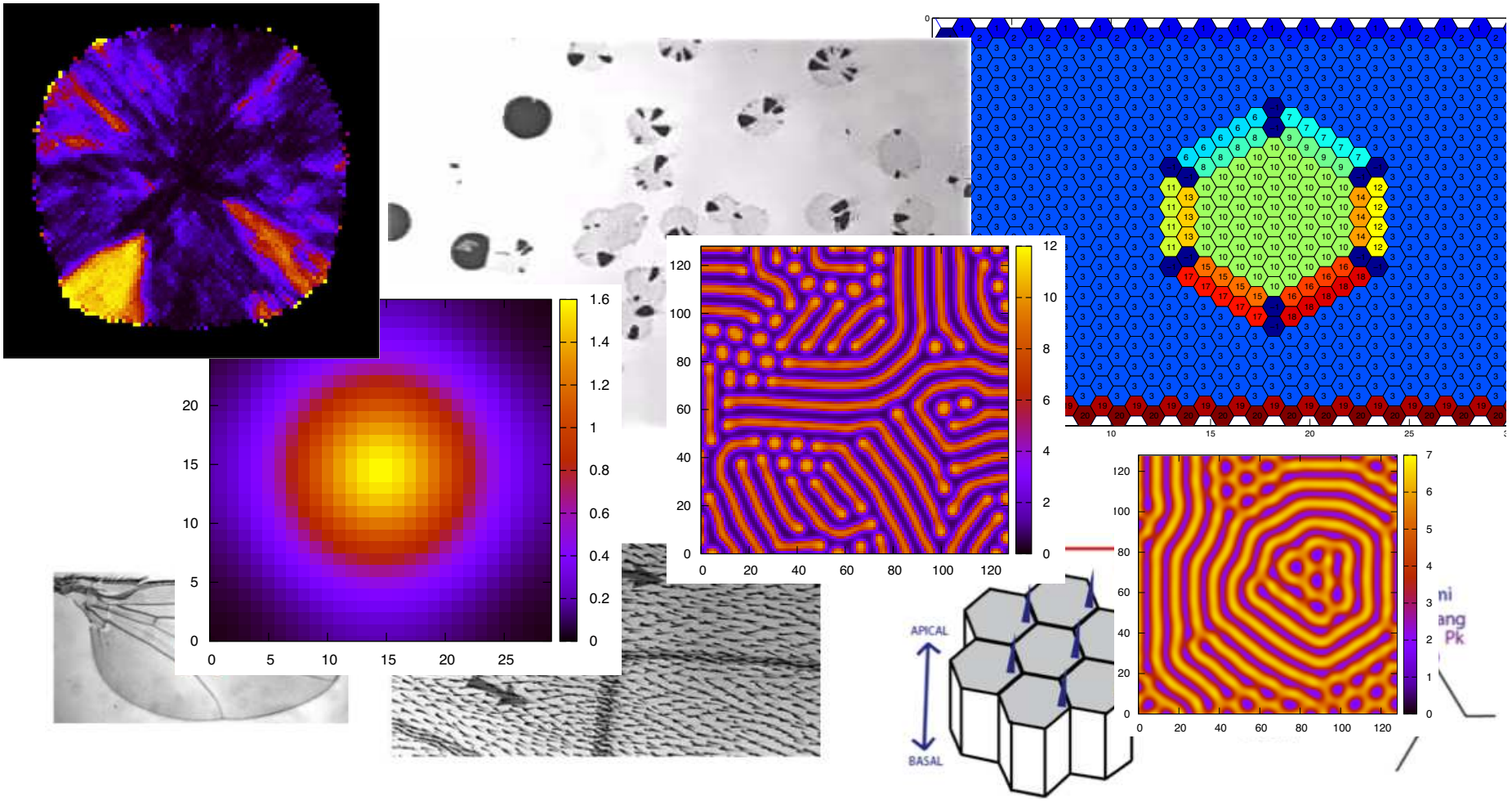
- **FRAMEWORK**
  - -> *QPN - SPN - CPN - HPN*
  - -> *uncoloured / coloured*
- **MODELLING BIO PETRI NETS**
- **COLOUR**
  - -> *gradients*
  - -> *phase variation*
  - -> *fly wing*
- **WHAT NEXT ?** -> **MODELLING 4 ANALYSING**
  - -> *how to analyse coloured Petri nets ?*
  - -> *model checking*
  - -> *efficient hybrid simulation*





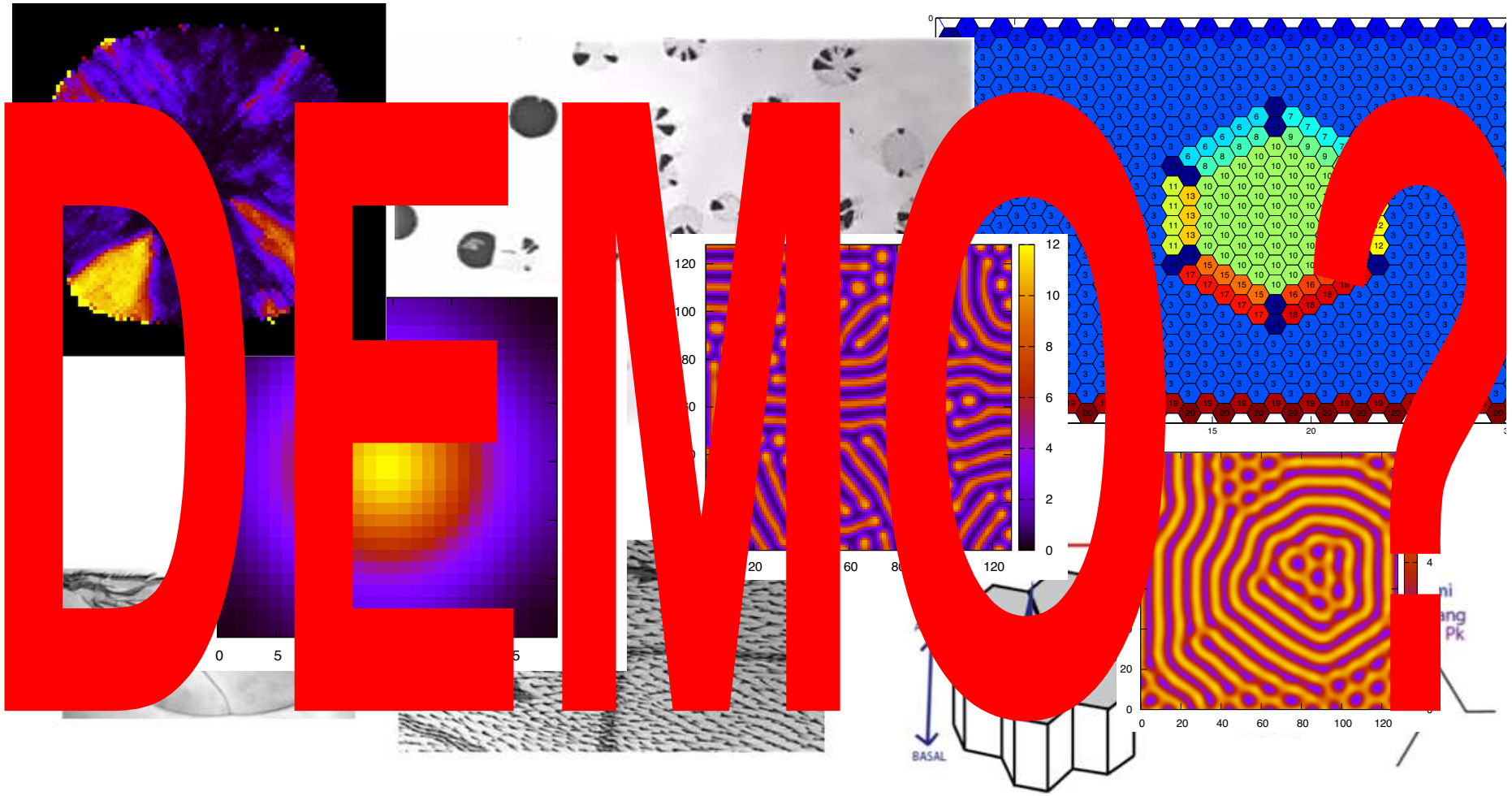


CASE STUDIES



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

HTTP://MULTISCALEPN.BRUNEL.AC.UK



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

HTTP://MULTISCALEPN.BRUNEL.AC.UK