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
ARE NETWORKS OF BIOCHEMICAL REACTIONS

\[ 2 \text{NAD}^+ + 2 \text{H}_2\text{O} \rightarrow 2 \text{NADH} + 2 \text{H}^+ + \text{O}_2 \]
Bio Networks

... ARE NETWORKS OF BIOCHEMICAL REACTIONS ...

... NATURALLY EXPRESSIBLE AS PETRI NETS ...

\[ 2 \text{NAD}^+ + 2 \text{H}_2\text{O} \rightarrow 2 \text{NADH} + 2 \text{H}^+ + \text{O}_2 \]
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

[Fas-Ligand]

Fas receptor
FADD

Procaspase-8
Bid (C terminal)

Caspase-8

Procaspase-3

Caspase-3

Procaspase-9

Caspase-9

DFF

DFF oligomer

DNA fragmentation

cell membrane

Apoptotic Stimuli

Apopptic Stimuli

Bax, Bad, Bim

Bcl-2, Bcl-xL

Cytochrome c

Mitochondrion

dATP/ATP

Apaf-1

Procaspase-8

Procaspase-3

Procaspase-9

[Apoptotic Stimuli]

FADD

Bax_Bad_Bim

Bcl-2_Bcl-xL

Apaf-1

Bid_C-Terminal

Cytochrome c

Mitochondrion

dATP/ATP

Procaspase-3

Procaspase-9

[Apoptotic Stimuli]

Fas-Ligand

Procaspase-8

Bax_Bad_Bim

Bcl-2_Bcl-xL

Apaf-1

Bid_C-Terminal

Cytochrome c

Mitochondrion

dATP/ATP

Procaspase-3

Procaspase-9

[Apoptotic Stimuli]

Fas-Ligand

Procaspase-8

Bax_Bad_Bim

Bcl-2_Bcl-xL

Apaf-1

Bid_C-Terminal

Cytochrome c

Mitochondrion

dATP/ATP

Procaspase-3

Procaspase-9

[Apoptotic Stimuli]

Fas-Ligand

Procaspase-8

Bax_Bad_Bim

Bcl-2_Bcl-xL

Apaf-1

Bid_C-Terminal

Cytochrome c

Mitochondrion

dATP/ATP

Procaspase-3

Procaspase-9

[Apoptotic Stimuli]

Fas-Ligand

Procaspase-8

Bax_Bad_Bim

Bcl-2_Bcl-xL

Apaf-1

Bid_C-Terminal

Cytochrome c

Mitochondrion

dATP/ATP

Procaspase-3

Procaspase-9

[Apoptotic Stimuli]

Fas-Ligand

Procaspase-8

Bax_Bad_Bim

Bcl-2_Bcl-xL

Apaf-1

Bid_C-Terminal

Cytochrome c

Mitochondrion

dATP/ATP

Procaspase-3

Procaspase-9

[Apoptotic Stimuli]

Fas-Ligand

Procaspase-8

Bax_Bad_Bim

Bcl-2_Bcl-xL

Apaf-1

Bid_C-Terminal

Cytochrome c

Mitochondrion

dATP/ATP

Procaspase-3

Procaspase-9

[Apoptotic Stimuli]

Fas-Ligand

Procaspase-8

Bax_Bad_Bim

Bcl-2_Bcl-xL

Apaf-1

Bid_C-Terminal

Cytochrome c

Mitochondrion

dATP/ATP

Procaspase-3

Procaspase-9

[Apoptotic Stimuli]
STATE-DEPENDENT RATE FUNCTIONS
ADDING TIME

STATE-DEPENDENT RATE FUNCTIONS

STOCHASTIC RATES

LAMBDA OF EXPONENTIAL WAITING TIME

CTMC

CONTINUOUS RATES

STRENGTH OF CONTINUOUS FLOW

ODEs

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

STOCHASTIC

CONTINUOUS
FRAMEWORK 2007

QUALITATIVE

STOCHASTIC

discrete state space

CONTINUOUS

continuous state space

time-free

timed, quantitative

abstraction

extension
QUALITATIVE

STOCHASTIC

CONTINUOUS

LTS / PO
CTL, LTL

CTMC
CSL, PLTL

ODEs
LTLc

discrete state space

continuous state space

approximation

extension

abstraction

time-free

timed, quantitative
**FRAMEWORK 2010**

- **QUALITATIVE**
  - time-free
  - abstraction
  - extension

- **CONTINUOUS**
  - timed, quantitative
  - abstraction
  - extension

- **STOCHASTIC**
  - discrete state space
  - approximation

- **HYBRID**
  - continuous state space
  - approximation

- **HYBRID**
  - LTS / PO
  - CTL, LTL
  - approximation

- **STOCHASTIC**
  - CTMC
  - CSL, PLTLc
  - approximation

- **CONTINUOUS**
  - ODEs
  - LTLc
COLOURED FRAMEWORK 2011

- LTS / PO
- CTL, LTL
- QUALITATIVE
- time-free
- timed, quantitative
- abstraction
- extension
- approximation
- extension
- approximation
- CTMC
- CSL, PLTLc
- STOCHASTIC
- discrete state space
- continuous state space
- HYBRID
- COLOURED
- CONTINUOUS
- ODEs
- LTLc
- COLOURED
- COLOURED
- COLOURED
KEY IDEA

4x2 MODELS SHARING STRUCTURE

QUANTITATIVE MODEL = QUALITATIVE MODEL + RATE FUNCTIONS (KINETICS)
COLOURED FRAMEWORK

Markov model

Hybrid world

ODE world

Petri net

continuous state space

discrete state space

discrete state space

timed, quantitative

time-free

approximation

abstraction

extension

extension

approximation

ODES

CTMC

LTS / PO

CTMC

CTMC

LTL, CSL, PLTL

Petri net

world

world

world

world

world

world

world

world
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, 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

SBML import/export

EXPORT TO MATLAB AND MANY OTHER TOOLS
MODELLING BIO (PETRI) NETS
- THREE APPROACHES
APPROACH 1
signalling cascade as wiring diagram
enzymatic reaction, mass-action kinetics
enzymatic reaction, mass-action kinetics
enzymatic reaction, mass-action kinetics

\[ A \leftrightarrow A|E \rightarrow B \]
enzymatic reaction, mass-action kinetics
enzymatic reaction, mass-action kinetics
enzymatic reaction, mass-action kinetics
NET COMPOSITION FROM BUILDING BLOCKS

SINGLE MASS-ACTION STEP
NET COMPOSITION FROM BUILDING BLOCKS

SINGLE PHOSPHOYLATION / DEPHOSPHORYLATION
**NET COMPOSITION FROM BUILDING BLOCKS**

* monika.heiner@b-tu.de  October 2015

DOUBLE PHOSPHOYLATION / DEPHOSPHORYLATION

[Diagram of double phosphorylation/dephosphorylation process]
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
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]
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

- reproduction_of_prey
- predator_death
- consumption_of_prey

Prey1: 50
Predator1: 100

2
Ex: Prey - Predator

- reproduction_of_prey
- predator_death

Prey1 → Predator1

- consumption_of_prey

2
**Ex: Prey - Predator**

- **Sub-system 1**
  - Reproduction of Prey
  - Predator Death
  - Prey 1: 50
  - Predator 1: 100
  - Consumption of Prey

- **Sub-system 2**
  - Reproduction of Prey
  - Predator Death
  - Prey 2: 50
  - Predator 2: 100
  - Consumption of Prey
**EX: PREY - PREDATOR**

- **definitions**
  
  ```
  colourset CS = 1-2;
  var x : CS;
  ```

- **better:**
  
  ```
  const SIZE = 2;
  colourset CS = 1-SIZE;
  var x : CS;
  ```

![Diagram of Prey-Predator model]
**EX: PREY - PREDATOR**

- **definitions**
  - `colours CS = 1-2;`
  - `var x : CS;`

- **better:**
  - `const SIZE = 2;`
  - `colours CS = 1-SIZE;`
  - `var x : CS;`

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

![Diagram](image_url)
EXAMPLE 1:  
DIFFUSION IN SPACE
Ex1: Diffusion - 1D
Ex1: Diffusion - 1D
Ex1: Diffusion - 1D
definitions

class D1 = 5; // grid size
const MIDDLE = D1/2;
colorset CS = 1-D1; // grid positions
var x, y : CS;
**Ex1: Diffusion - 1D**

- **definitions**
  
  ```c++
  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);
  ```

---

```
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

```plaintext
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

\( (a=x-1 \mid a=x+1) \& (1\leq a) \& (a\leq D1) \);

## Movement = Changing Colour
We obtain a general model pattern for an arbitrary, but static size of the discrete,
**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
Ex1: Diffusion - 2D

Scheme
Ex1: Diffusion - 2D

Scheme

Definitions

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

```
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);

Diagram:

- t1
- (x,y)
- (a,b)
- cAMP
- Grid2D
- 100 ((x=MIDDLE & y=MIDDLE)
- [neighbour2D4(x,y,a,b)]
Ex1: Diffusion - 2D4 Neighbourhood
eight neighbours

\textbf{function} neighbour2D8 (CD1 \(x\), CD2 \(y\), CD1 \(a\), CD2 \(b\)) \textbf{bool}:

// \((a,b)\) is one of the up to eight neighbours of \((x,y)\)
\((a=x-1 \mid a=x \mid a=x+1) \& (b = y-1 \mid b=y \mid b=y+1) \& (! (a=x \& b=y)) \& (1<=a \& a<=D1) \& (1<=b \& b<=D2);
Ex1: Diffusion - 2D8 Neighbourhood
Ex1: Diffusion - 2D4 Neighbourhood, 15x15

'data.dat.00000000' matrix
Ex1: Diffusion - 2D4 Neighbourhood, 15x15

'data.dat.00000006' matrix
Ex1: Diffusion - 2D4 Neighbourhood, 15x15
Ex1: Diffusion - 2D4 Neighbourhood, 15x15

'data.dat.00000015' matrix
Ex1: Diffusion - 2D4 Neighbourhood, 15x15

'data.dat.00000020' matrix
Ex1: Diffusion - 2D4 Neighbourhood, 15x15

'data.dat.00000030' matrix
Ex1: Diffusion - 2D4 Neighbourhood, 15x15

'data.dat.00000040' matrix
**Ex1: Diffusion - 2D4 Neighbourhood, 15x15**

'data.dat.00000050' matrix
Ex1: Diffusion - 2D4 Neighbourhood, 30x30

'data.dat.00000050' matrix
Ex1: Diffusion - 2D4 Neighbourhood, 60x60

'data.dat.00000050' matrix
Ex1: Diffusion - 2D4 Neighbourhood, 120x120

'data.dat.00000050' matrix
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)
Mutation rates: estimating phase variation rates when fitness differences are present and their impact on population structure

Nigel J. Saunders,¹† E. Richard Moxon¹ and Mike B. Gravenor²

¹Molecular Infectious Diseases Group, Institute of Molecular Medicine, University of Oxford, Headington, Oxford OX3 9DS, UK
²Institute 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.
Mutation rates: estimating phase variation rates when fitness differences are present and their impact on population studies

Nigel J. Saunders,1† E. Richard Moxon1 and Mike B. Gravenor2

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

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

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
**Ex2: Cell Colonies, Basics**

- **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**

\[
\begin{align*}
\text{A2A} & \quad \text{da} \times (1-\alpha) \times \text{A} \\
\text{B2A} & \quad \text{da} \times \alpha \times \text{A} \\
\text{B2B} & \quad \text{db} \times (1-\beta) \times \text{B} \\
\text{A2B} & \quad \text{db} \times \beta \times \text{B}
\end{align*}
\]
Ex2: Cell Colonies, Continuous Plot
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;
```

![Diagram](image)

```
(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;

(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)
**Ex2: Cell Colonies, Basic Model**

`colorset Phenotype = enum with A, B;`

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

(\(c=A\)) & (\(\text{div}=\text{replicate}\)) : cell*da*(1-\(\alpha\))
(\(c=A\)) & (\(\text{div}=\text{mutate}\)) : cell*(da*\(\alpha\))
(\(c=B\)) & (\(\text{div}=\text{replicate}\)) : cell*(db*(1-\(\beta\)))
(\(c=B\)) & (\(\text{div}=\text{mutate}\)) : cell*(db*\(\beta\))
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;
**Ex2: Cell Colonies, Controlling Colony Spreading**

\[
\text{colorset } \text{Grid } = \text{product with } \text{Grid2D x Phenotype};
\]
**Ex2: Cell Colonies, Controlling Thickness**

\[
\text{colorset } \text{Grid} = \text{product with } \text{Grid2D} \times \text{Phenotype};
\]
**Ex2: Cell Colonies, Controlling Colony Size**

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

![Diagram of cell colonies and control mechanisms](image)
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)

'data.dat.00000030' matrix
Ex2: 2D - TRACE 1 (HIGH, F=1)
Ex2: 2D - TRACE 1 (HIGH, F=1)

'data.dat.00000050' matrix
Ex2: 2D - TRACE 1 (HIGH, F=1)
Ex2: 2D - TRACE 1 (HIGH, F=1)

'data.dat.00000070' matrix
Ex2: 2D - TRACE 1 (HIGH, F=1)

'data.dat.00000080' matrix
Ex2: 2D - TRACE 1 (HIGH, F=1)
Ex2: 2D - Trace 2 (HIGH, F=1)
Ex2: 2D - TRACE 3 (HIGH, F=1)

'data.dat.00000066' matrix
Ex2: 2D - Varying Fitness, Trace 1 (Medium, F=1)

'data.dat.00000100' matrix

0 20 40 60 80 100
0 20 40 60 80 100
0 20 40 60 80 100
0 20 40 60 80 100
0 20 40 60 80 100
0 20 40 60 80 100
0 20 40 60 80 100
0 20 40 60 80 100
0 20 40 60 80 100
0 20 40 60 80 100

0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9
Ex2: 2D - VARYING FITNESS, TRACE 2 (MEDIUM, F=1)

'data.dat.00000100' matrix
Ex2: 2D - Varying Fitness, Trace 1 (Medium, F=0.99)

'data.dat.00000100' matrix
Ex2: 2D - Varying Fitness, Trace 1 (Medium, F=0.90)

'data.dat.00000093' matrix
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]
Ex3 - Planar Cell Polarity

(a) (b) (c) (d)
EX3: PLANAR CELL POLARITY

[FQIAN GAO, PHD THESIS 2013]
EX3: PLANAR CELL POLARITY, PLAIN MODEL (7 CELLS)
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

[TCBB 2012]
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?
  - how to analyse coloured Petri nets?
  - model checking
  - efficient hybrid simulation
HTTP://WWW-DSSZ.INFORMATIK.TU-COTTBUS.DE
HTTP://MULTISCALEPN.BRUNEL.AC.UK

monika.heiner@b-tu.de

October 2015