

Exploratory simulation of cell ageing using hierarchical models

GIW 2008

▶ Hayssam Soueidan

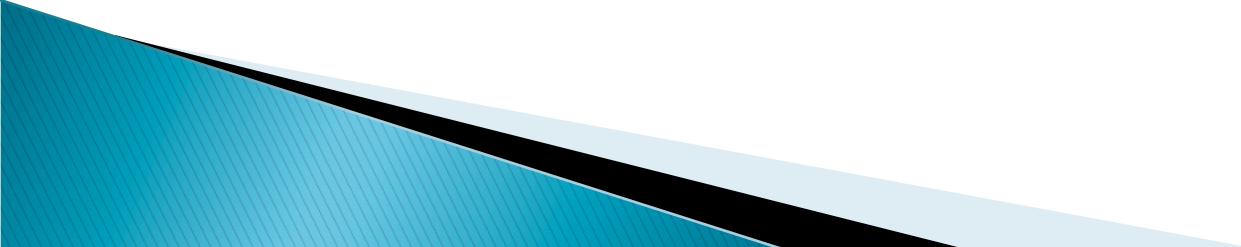
U.Bordeaux, France

▶ LaBRI & INRIA project “MAGNOME”

Modeling challenges in Systems Biology

- ▶ Why do we care about modeling?
- ▶ → Understanding how systems give rise to emerging behaviour
- ▶ Types of models:
 - Stoichiometric flux models
 - Metabolic kinetic models
- ▶ Analysis:
 - Metabolic analysis *global qualitative*
 - Kinetic analysis *local quantitative*
- ▶ ★ Klipp *et al.*, 2007:
 - Modeling: PowerPoint and Excel
 - Analysis: MatLab

Outline

- ▶ Why hierarchical modular modeling ?
 - ▶ How we do it in BioRica modeling and simulation tool
 - ▶ Yeast rejuvenation study
- 

Modeling challenges in Systems Biology

- ▶ ESF Grand Challenges, NSF Funding Objectives
 - Hierarchical modeling
 - Integrating genome-scale and fine-grained phenomena
- ▶ 1. Multiple time-scales
 - Transforming metabolites: fast
 - Synthesizing an enzyme: slow
- **ODE stiff systems** (Brenan *et al.*, 1996) fast-scaled reactions are stable, but slow reactions determine the system's trajectory
- 2. Deterministic vs stochastic
 - Excess of metabolites
 - Lack of metabolites

Modeling in BioRica

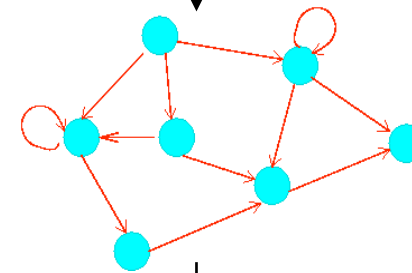
- ▶ Webster's dictionary: “a *system* is a complex unity formed of many often diverse parts subject to a common plan, serving a common purpose”
 - an electronic circuit
 - nuclear power plant
 - an *S. cerevisiae* cell
- ▶ Systems are comprised of specific components (sub-systems) that interact
 - some components' activities occur concurrently (independently)
 - some components' activities are linked in some way
 - system's behaviour mainly defined by interactions of components
- ▶ → **two main specificities: hierarchy and composition**

Where we come from

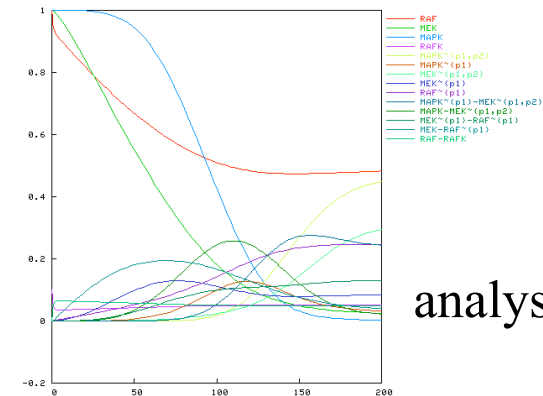
- ▶ “System” approaches in industrial engineering (since late 1980’s)
 - AltaRica modeling language (1999)
 - Modeling platform
 - textual and graphical modeling interface
 - simulation and verification tools
 - trace analysis
 - Models are compositional and **hierarchical**
 - behavioral hierarchy
 - architectural hierarchy
- ▶ → Well-suited to modeling challenges



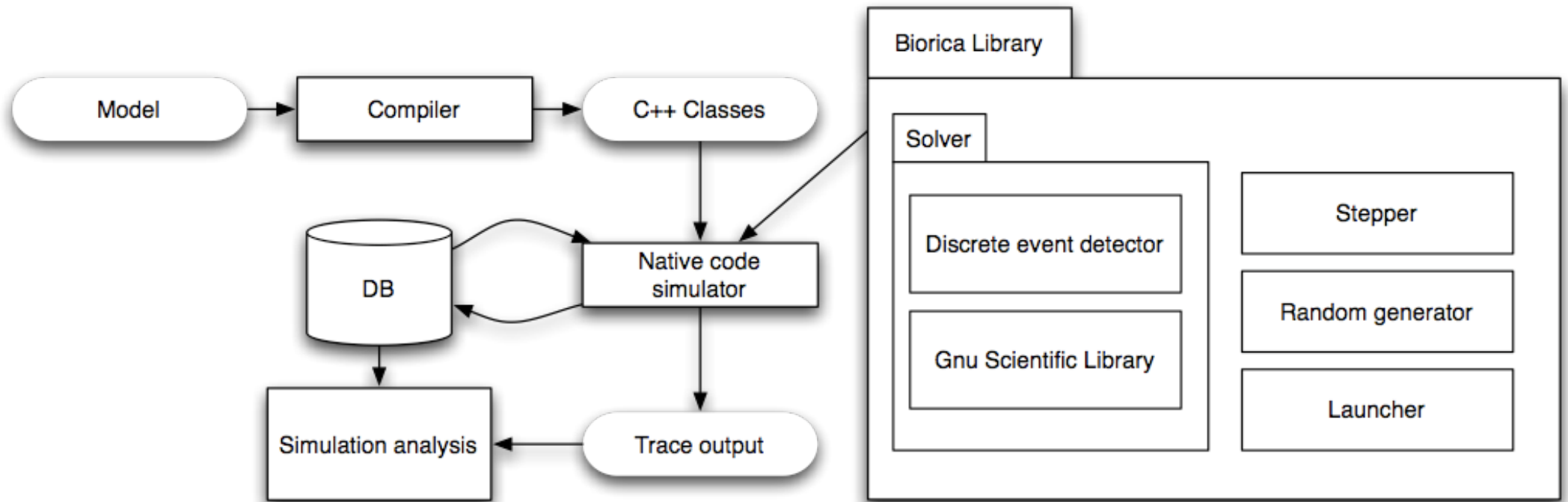
specification

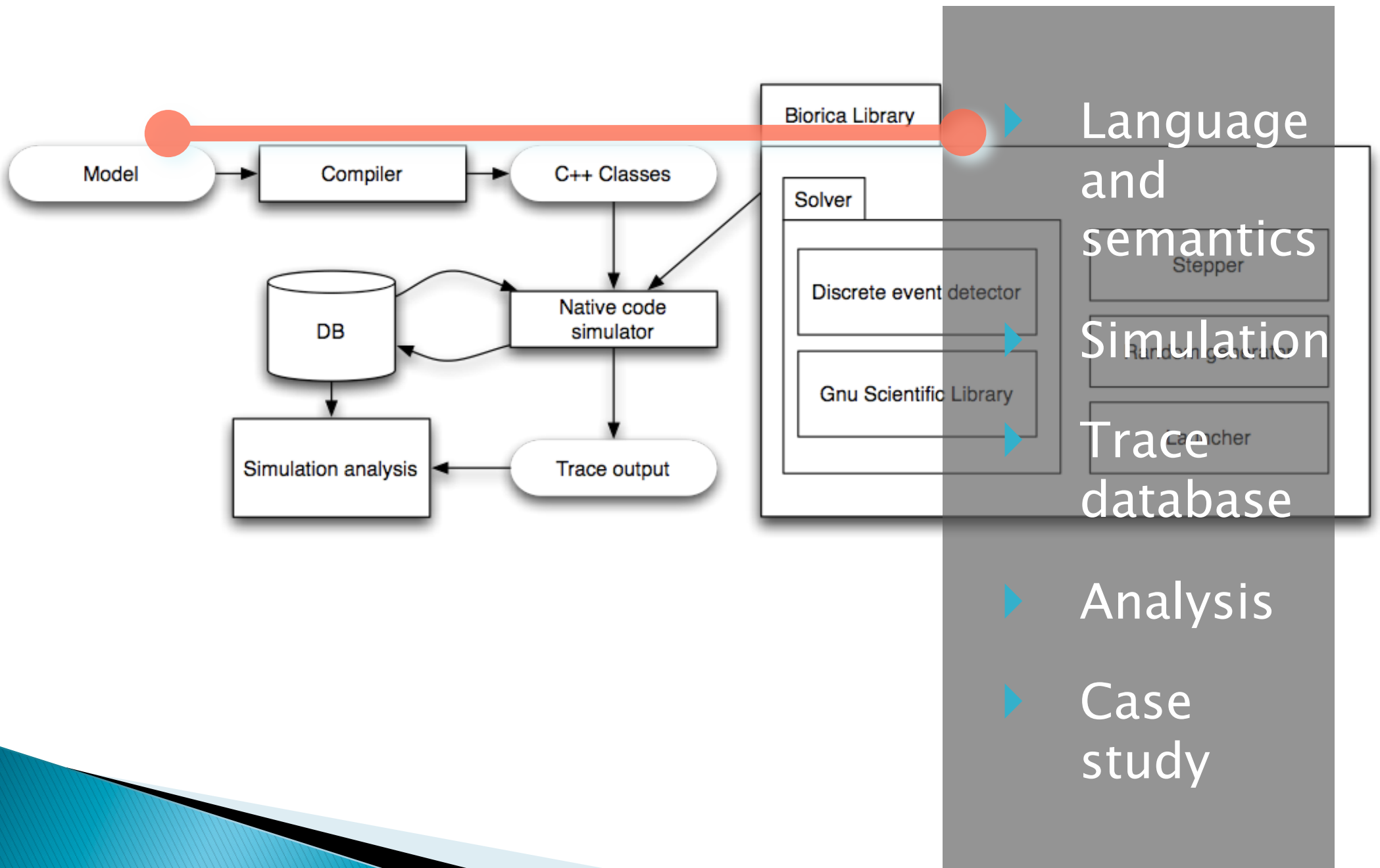


model



Compiling models with BioRica





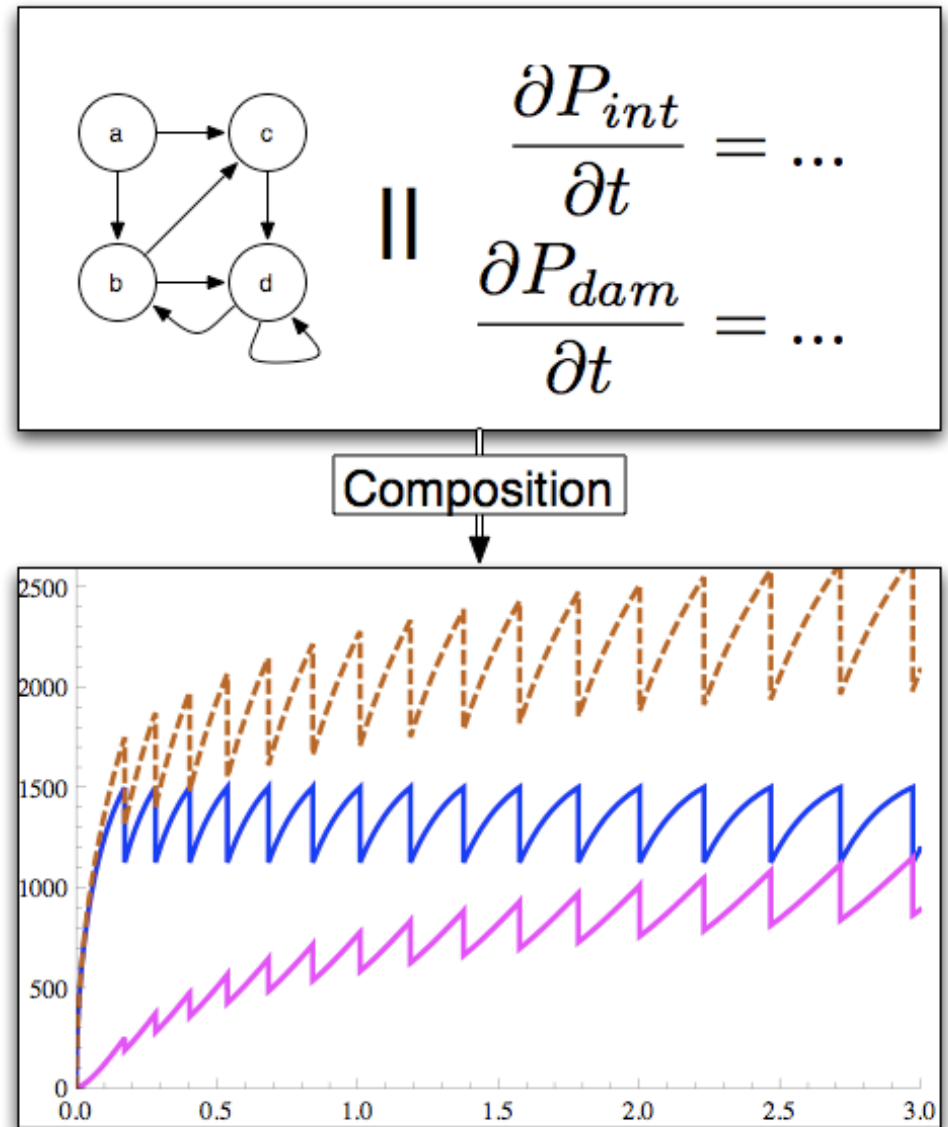
Language

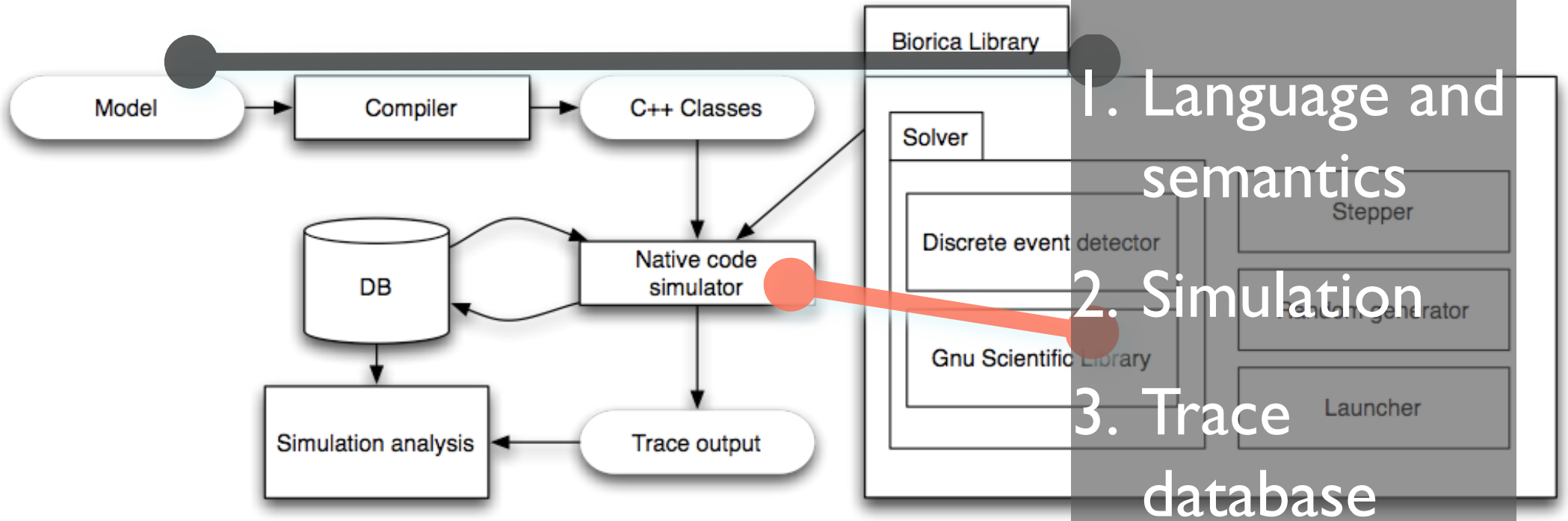
- ▶ Based on AltaRica
 - Industrial-strength
 - Designed for engineers
- ▶ Simple declarative and hierarchical language
 - Formal semantics
- ▶ Description of both models and properties of models
- ▶ (e.g. Teddy onthology)

```
node Porte-  
  > state-  
  > > ouverture:BOOL;-  
  > > transferee:[0,30];-  
  > flow-  
  > > apicale:[0,30]:i;-  
  > > basale:[0,30]:o;-  
  > event-  
  > > fermeture,transfert;-  
  > trans-  
  > > true |- fermeture -> ouverture := ~ouverture;-  
  > > ouverture |- transfert -> transferee := transferee+1;-  
  > assert-  
  > > basale = transferee;-  
  > extern-  
  > > law<transfert>:Exponential{0.1};-  
  > > law<fermeture>:Exponential{0.01};-  
edon-  
-  
node Endothelium-  
  > state-  
  > > edhf:[0,30]-
```

Semantics

- ▶ Compositional (Connections, synchronization)
- ▶ Generality: High expressivity, captures most existing models
- ▶ Mathematically sound and operational
- ▶ **Why do we care ?** Coherent composition with almost any calculation (ex: MatLab)





1. Language and semantics

2. Simulation

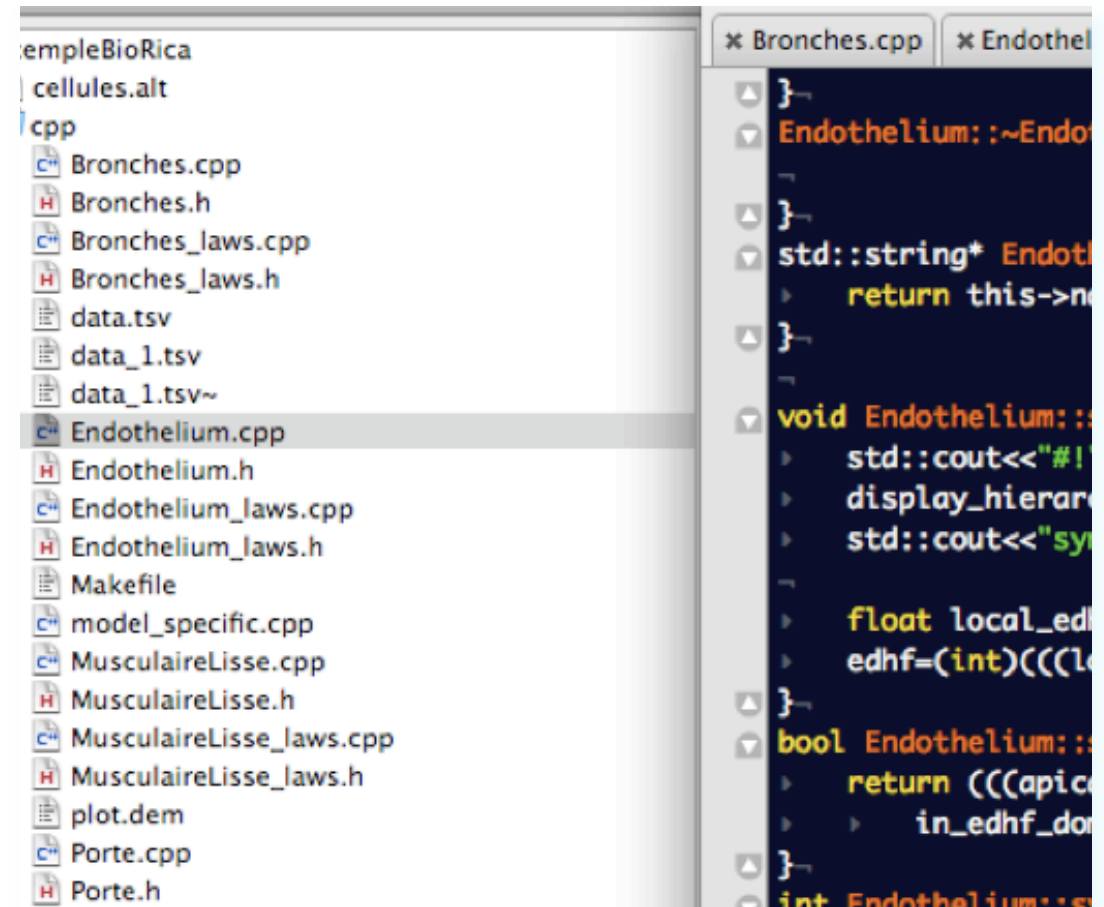
3. Trace database

4. Analysis

5. Case study

Simulation

- ▶ Automatic translation of models into C++ code
- ▶ Simulator architecture open for customization and extension
- ▶ Efficient
- ▶ Parameterized
- ▶ Subject to profiling and debugging tools (GDB, Shark etc.)



The screenshot shows a C++ IDE with a project named 'exempleBioRica'. The file explorer on the left lists the following files:

- cellules.alt
- cpp
 - Bronches.cpp
 - Bronches.h
 - Bronches_laws.cpp
 - Bronches_laws.h
- data.tsv
- data_1.tsv
- data_1.tsv~
- Endothelium.cpp (highlighted)
- Endothelium.h
- Endothelium_laws.cpp
- Endothelium_laws.h
- Makefile
- model_specific.cpp
- MusculaireLisse.cpp
- MusculaireLisse.h
- MusculaireLisse_laws.cpp
- MusculaireLisse_laws.h
- plot.dem
- Porte.cpp
- Porte.h

The right pane shows the code for 'Endothelium.cpp' with the following visible snippets:

```
}  
Endothelium::~Endothelium()  
{  
}  
std::string* Endothelium::get_edhf()  
{  
    return this->edhf;  
}  
void Endothelium::display_hierarchy()  
{  
    std::cout<<"#!"  
    display_hierarchy()  
    std::cout<<"sy"  
}  
float local_edhf(const int i)  
{  
    float local_edhf=(int)((local_edhf[i]  
}  
bool Endothelium::is_in_edhf(const int i)  
{  
    return ((capic[i]  
        in_edhf_dor  
}  
int Endothelium::s
```

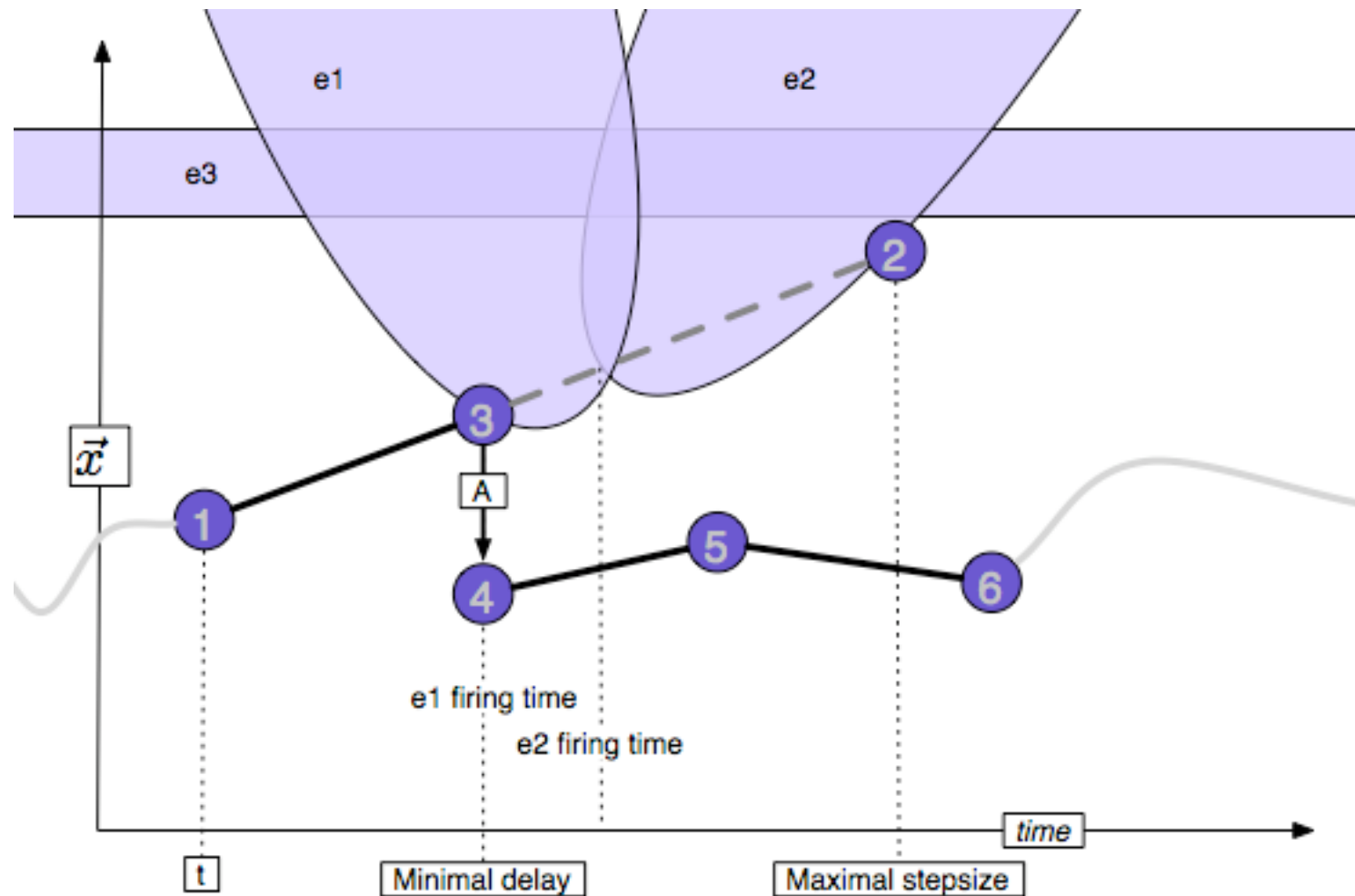

Simulation in BioRica

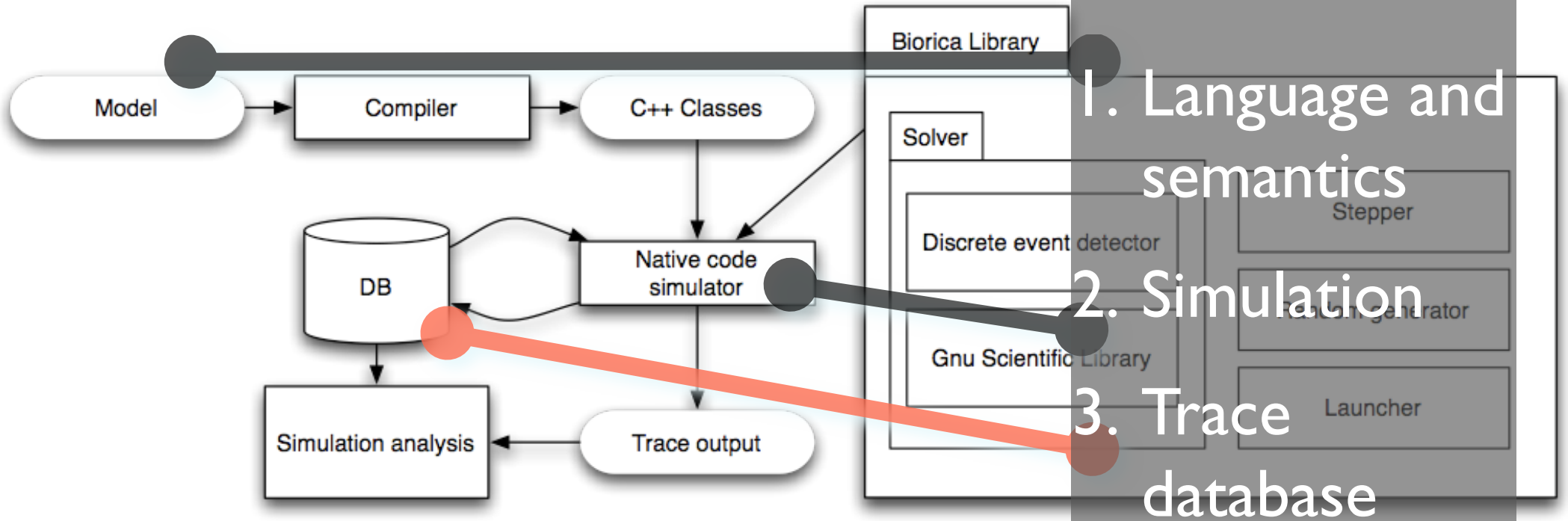
Algorithm 1 General simulation schema

Require: current state S , current simulation time t , maximal simulation time t_{\max}

```
1:  $S' = S$ 
2: while alive( $S, S'$ ) = 1 and  $t < t_{\max}$  do
3:    $S' = S$ 
4:    $t, S =$  advance_numerical_integration()
5:   if  $e =$  discrete_events() then
6:      $t =$  get_discrete_event_time()
7:     store_event( $e$ )
8:      $S =$  update( $S, e$ )
9:     reset_numerical_integrator()
10:  end if
11:  store_state( $S$ )
12: end while
```

Simulation in BioRica





1. Language and semantics

2. Simulation

3. Trace database

4. Analysis

5. Case study

Trace database

- ▶ Saves simulation results for varying parameters
- ▶ Comparative studies
- ▶ Simulation time control (replay, pause, reverse, fast forward)
- ▶ Randomized decisions can be replayed for different outcomes

Database Browser

Databases

- [information_schema](#)
- [cellageing](#)
 - [MaturingTime](#)
 - [batch](#)
 - [batchTree](#)
 - [datapoint](#)
- [mysql](#)
- [semtest](#)
- [test](#)

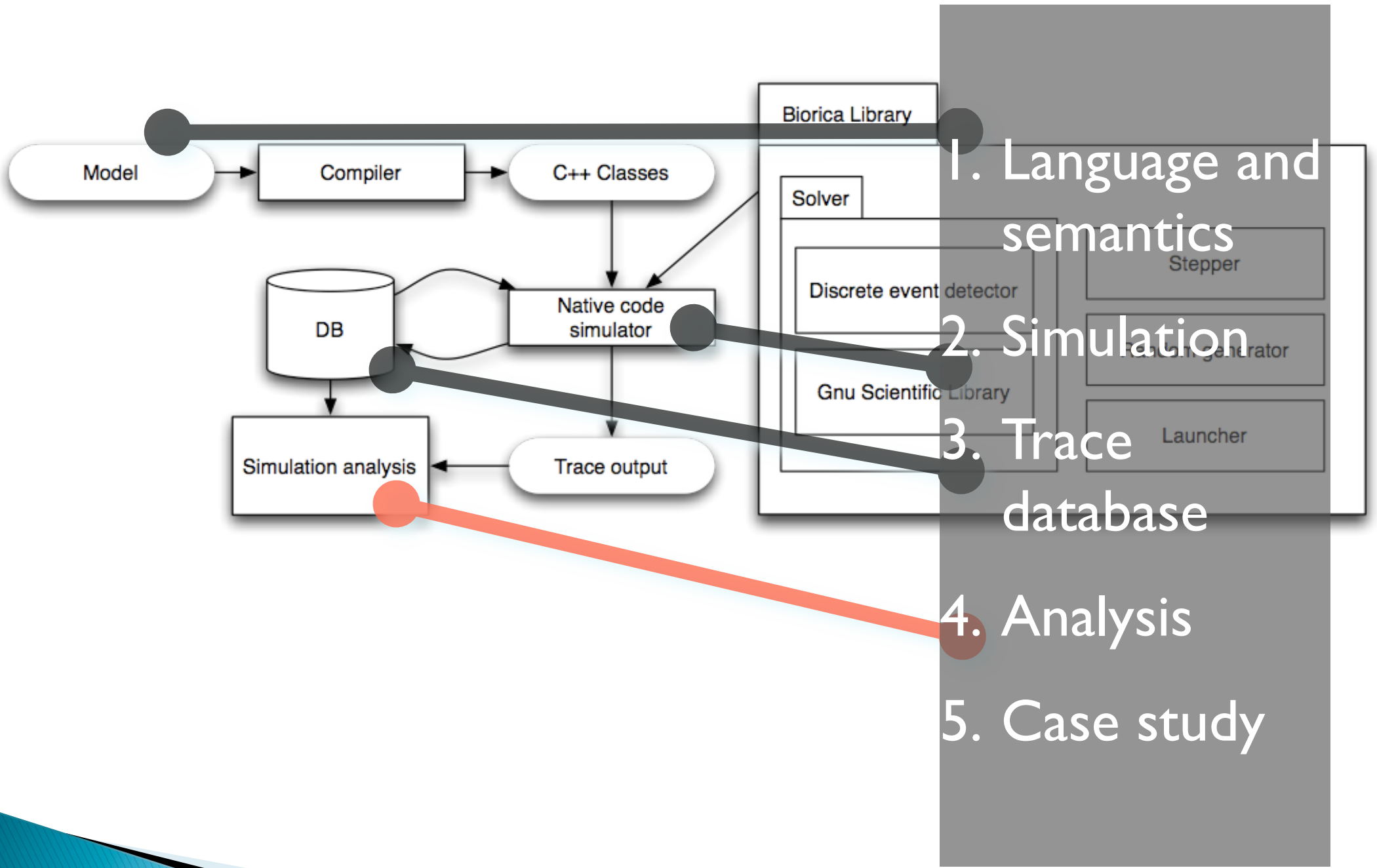
Columns in table: batch

| Name | Type | Nullable | Default | Search |
|-----------|------------------|----------|---------|--------|
| k1 | double | YES | NULL | |
| k2 | double | YES | NULL | |
| k3 | double | YES | NULL | |
| k4 | double | YES | NULL | |
| retention | double | YES | NULL | |
| Pdiv | double | YES | NULL | |
| size | double | YES | NULL | |
| daughter | bit(1) | YES | NULL | |
| imRoot | bit(1) | YES | NULL | |
| fitness | double | YES | NULL | |
| id | int(10) unsigned | NO | NULL | |

SELECT * FROM batch

Records 1 to 5 [Next >>](#)

| k1 | k2 | k3 | k4 | retentio |
|----------|-------------------|-----|----------|----------|
| 10000000 | 0.693147180559945 | 1.2 | 0.693147 | |
| 10000000 | 0.693147180559945 | 1.2 | 0.693147 | |
| 10000000 | 0.693147180559945 | 1.2 | 0.693147 | |
| 10000000 | 0.693147180559945 | 1.2 | 0.693147 | |

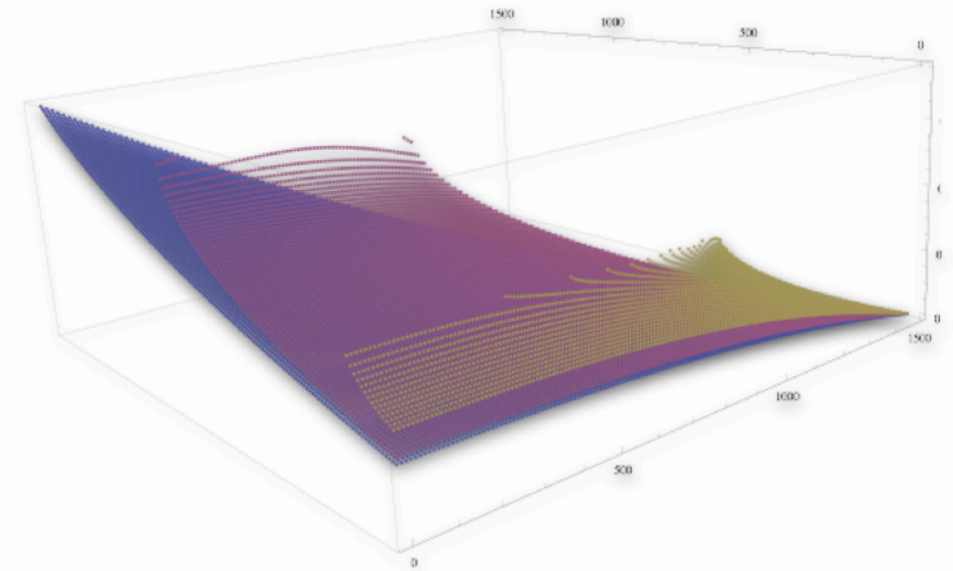


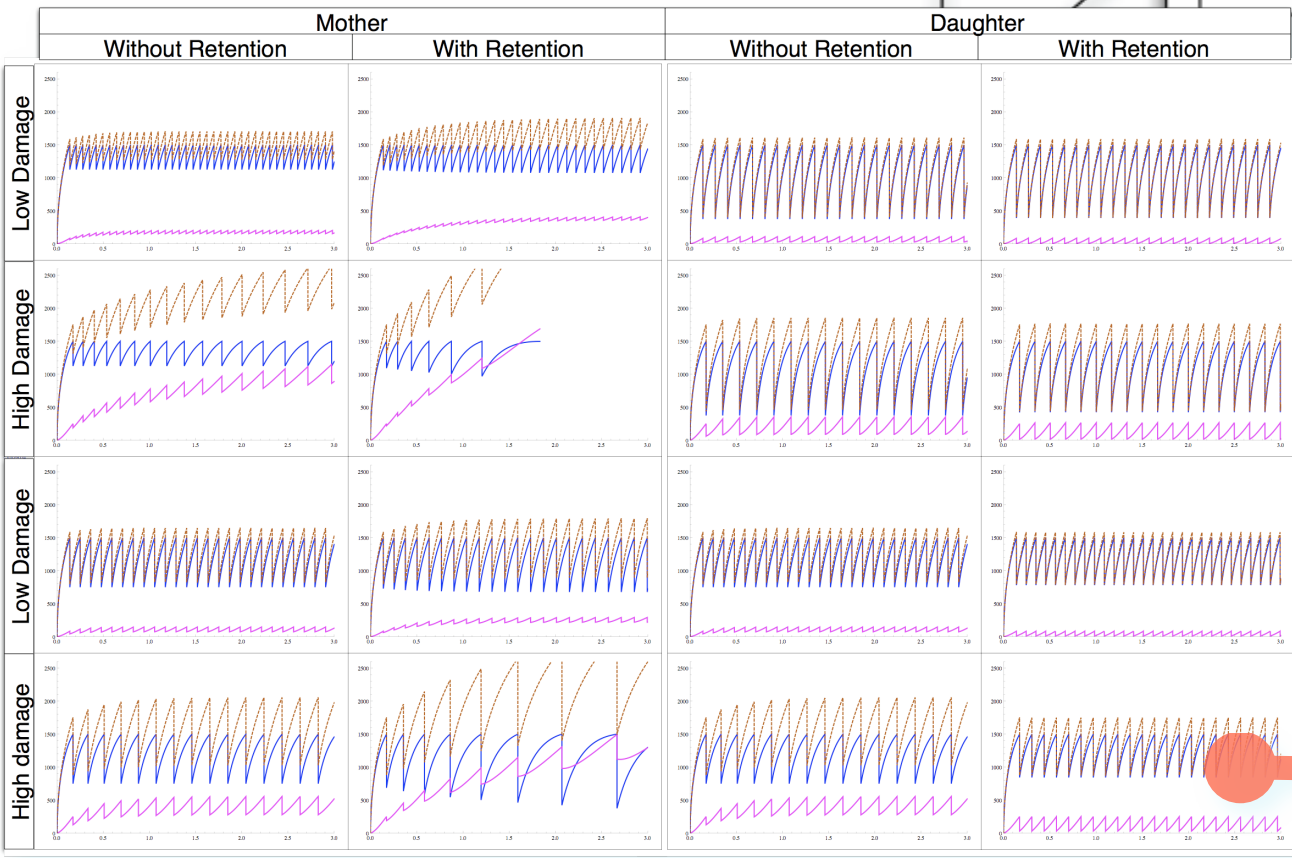
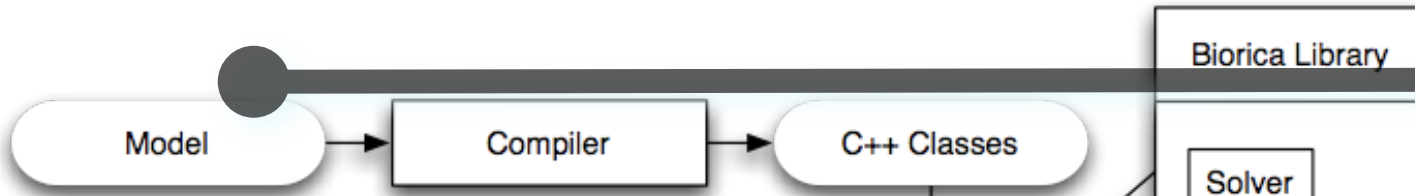
Exploration, Visualization, Analysis tool

- ▶ Extraction of high level qualitative properties
- ▶ Statistical synthesis
- ▶ Features extraction by user defined scripts connecting to the database
 - Python
 - Mathematica
 - Tulip
- ▶ MySQL

Studying Lifetime, w/ varying k_3 from 0.1 -> 2.2

```
In[4]:= ValForK3[k3_] := SQLExecute[
  SQLSelect["Cell Ageing", {"MaturingTime"}],
  {SQLColumn["MaturingTime.Pint_init"],
   SQLColumn["MaturingTime.Pdam_init"],
   SQLColumn["MaturingTime.t"]},
  SQLColumn["MaturingTime.k3"] = k3,
  SortingColumns -> None, Distinct -> False, Timeout -> 30, ShowColumnHeadings -> True, GetAsStrings -> False];
ListPointPlot3D[Table[{ValForK3[k3][[2 ;;]]}, {k3, 0, 2.2, 0.2}]]
```





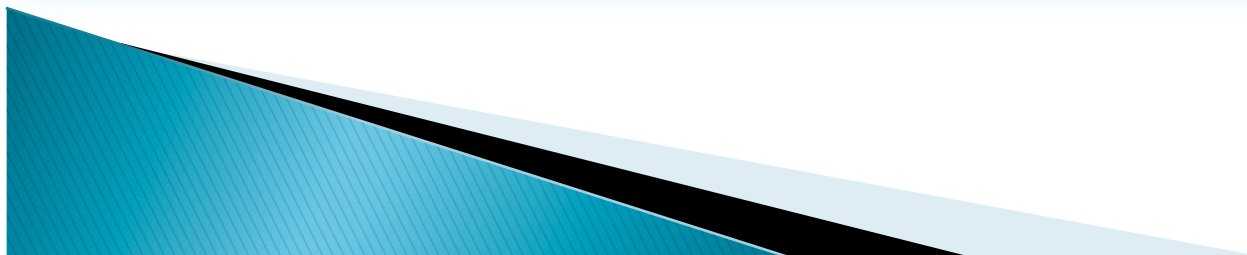
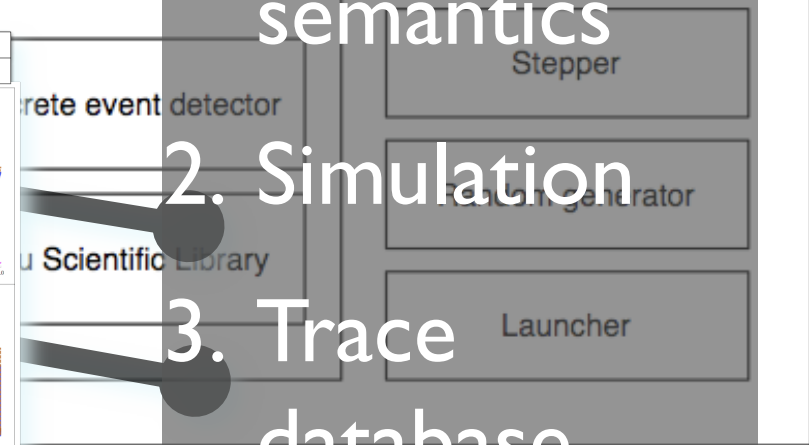
1. Language and semantics

2. Simulation

3. Trace database

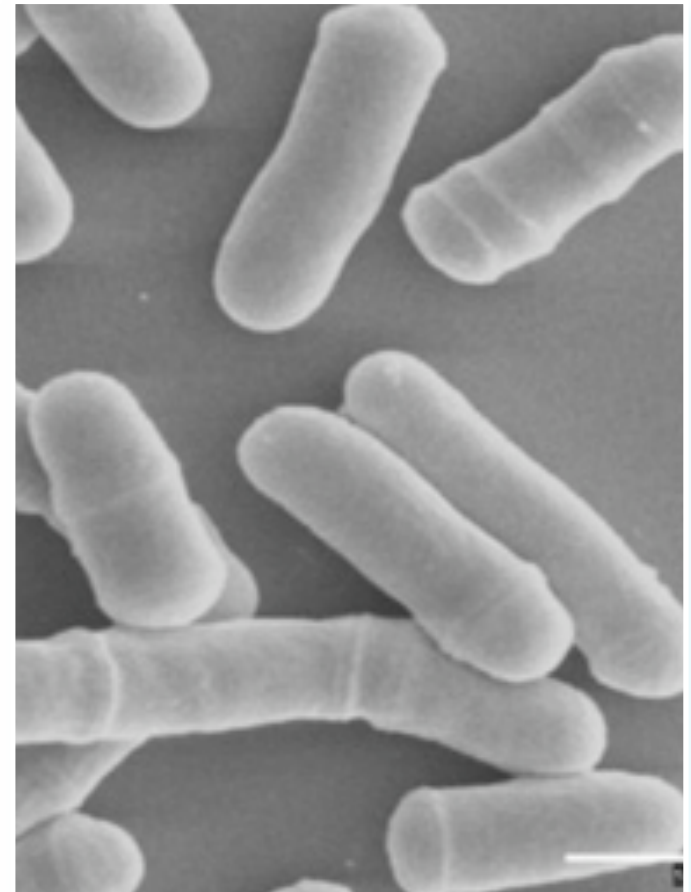
4. Analysis

5. Case study

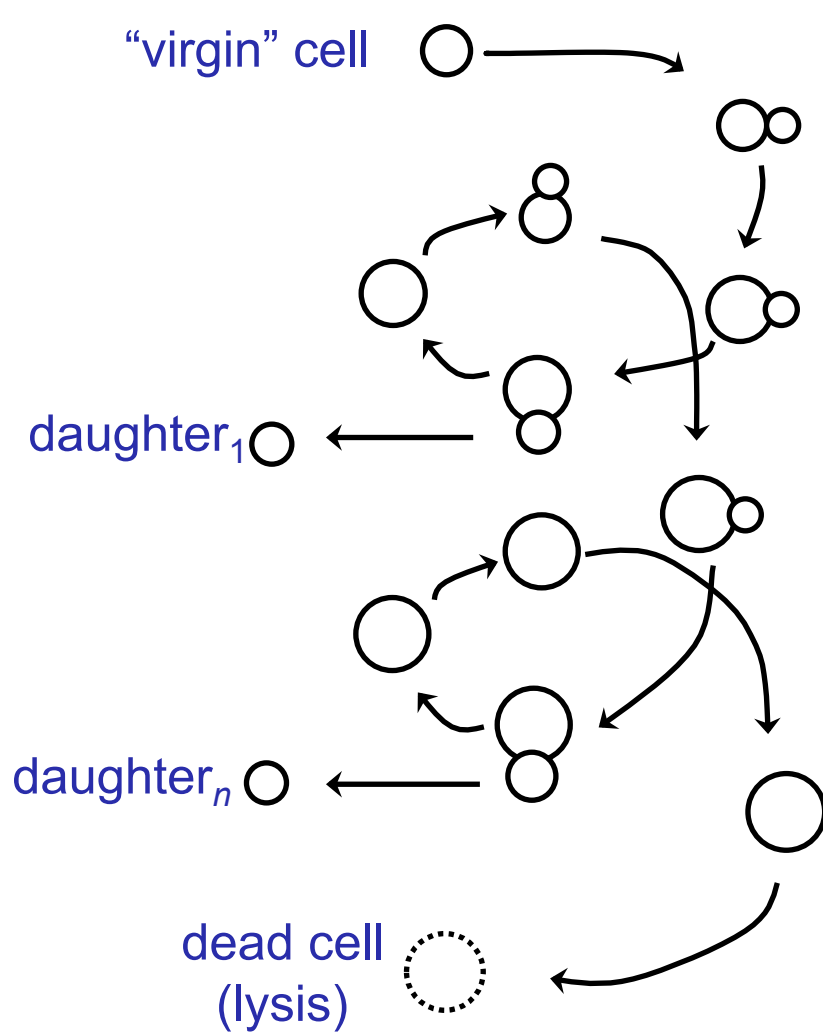


Rejuvenation in a cell ageing model

- ▶ Joint work with M. Cvijovic and E. Klipp (MPI)
- ▶ Extension of a single-cell model to a structured population
- ▶ Hierarchical simulation with dynamic creation
- ▶ Automated large scale exploration of parameter sets (>100 000 simulations)



The Cell Spiral Model of Yeast Aging

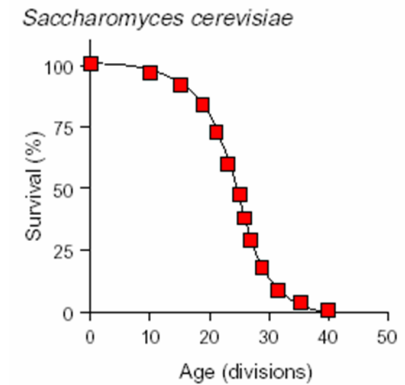
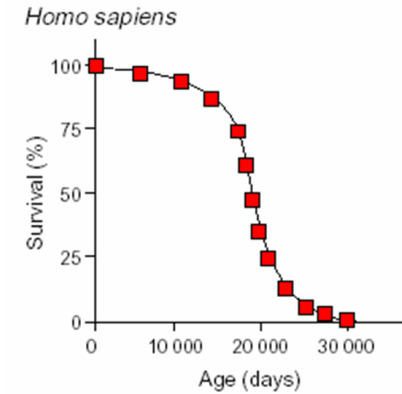


Generation
(cell cycle)

1st

*n*th

Ageing



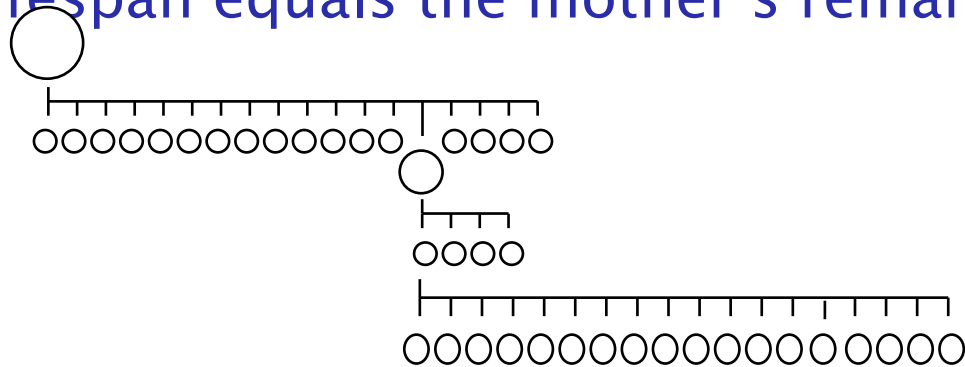
Lifespan = *n* (20-40)

How does population remain "immortal"?

- In every daughter cell, the lifespan “clock” is reset to zero
- Each division produces a cell that can divide many more times
- “Old” cells are very rare in a large exponentially growing population ($1 / 2^{a+1}$)

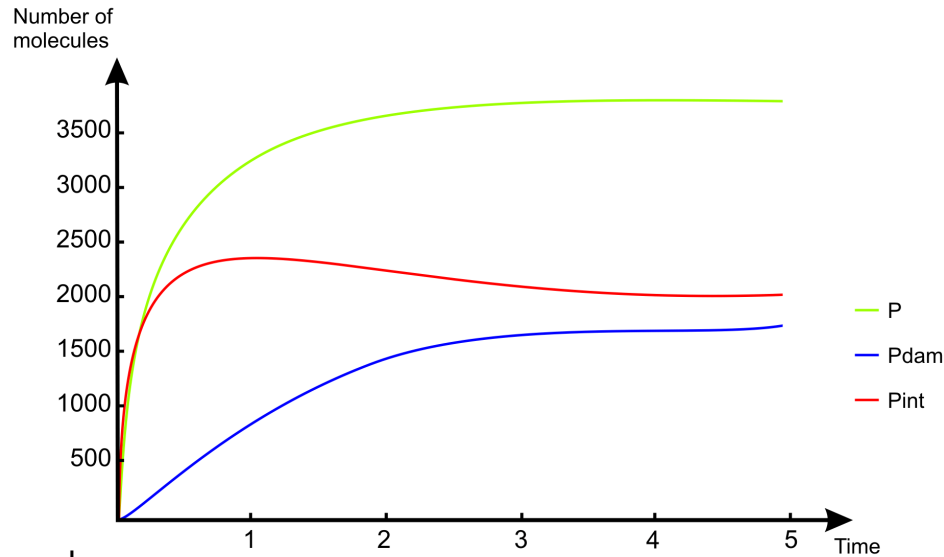
What limits yeast's lifespan?

- Occasionally, daughters of old mothers are born prematurely aged!
- ▶ Their lifespan equals the mother's remaining lifespan



- The asymmetry has broken down - accompanied by loss of size asymmetry (“symmetric buds”)
 - The daughters of symmetric buds have normal lifespan
- Symmetric buds have inherited a “senescence factor”...

ODE model description



Legend:

P - total protein

Pint - intact proteins

Pdam - damaged proteins

Pdiv - treshold for division

k_1 - limits the system in terms not to have continues expansion in volume

k_2 - represents degradation rate of intact proteins

k_3 - represents damage rate

k_4 - represents degradation rate of damaged proteins

re - retention coefficient

daughters - daughter size (25 or 50)

mothers - mother size (50 or 75)

Equations:

$$\frac{dP}{dt} = \frac{k_1}{k_s + P_{int} + P_{dam}} - k_2 P_{int} - k_4 P_{dam}$$

$$\frac{dP_{int}}{dt} = \frac{k_1}{k_s + P_{int} + P_{dam}} - k_2 P_{int} - k_3 P_{int}$$

$$\frac{dP_{dam}}{dt} = k_3 P_{int} - k_4 P_{dam}$$

Trigger:

$P_{int} > P_{div}$

Events:

for daughters:

$$P \rightarrow P_{int} \cdot daughter \cdot (1 + re \cdot \frac{P_{dam}}{P_{int}}) + P_{dam} \cdot daughter \cdot (1 - re)$$

$$P_{int} \rightarrow P_{int} \cdot daughter \cdot (1 + re \cdot \frac{P_{dam}}{P_{int}})$$

$$P_{dam} \rightarrow P_{dam} \cdot daughter \cdot (1 - re)$$

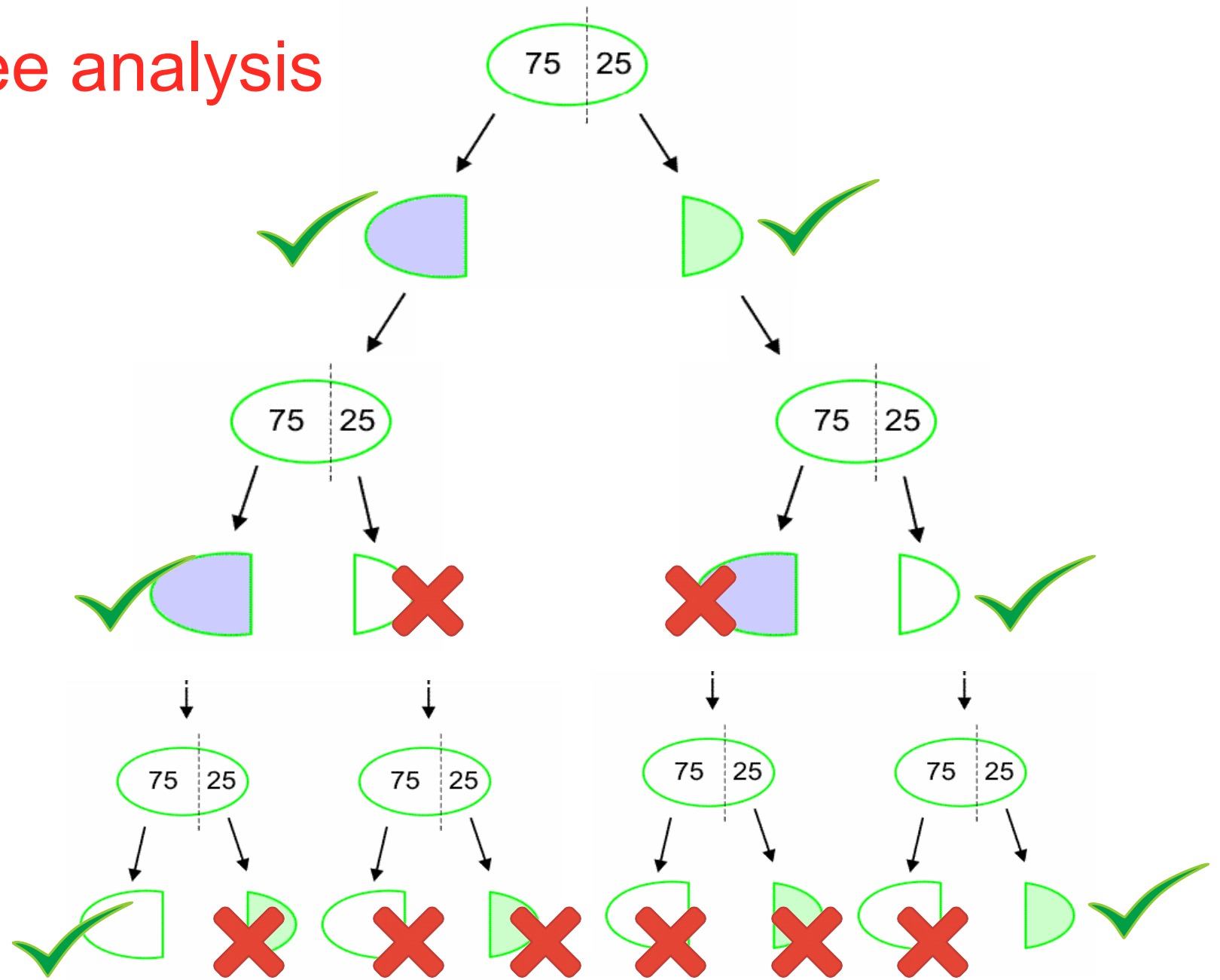
for mothers:

$$P \rightarrow P_{int} \cdot mother \cdot (1 - re \cdot \frac{P_{dam}}{P_{int}} \cdot \frac{1 - mother}{mother}) + P_{dam} \cdot (mother + (1 - mother) \cdot re)$$

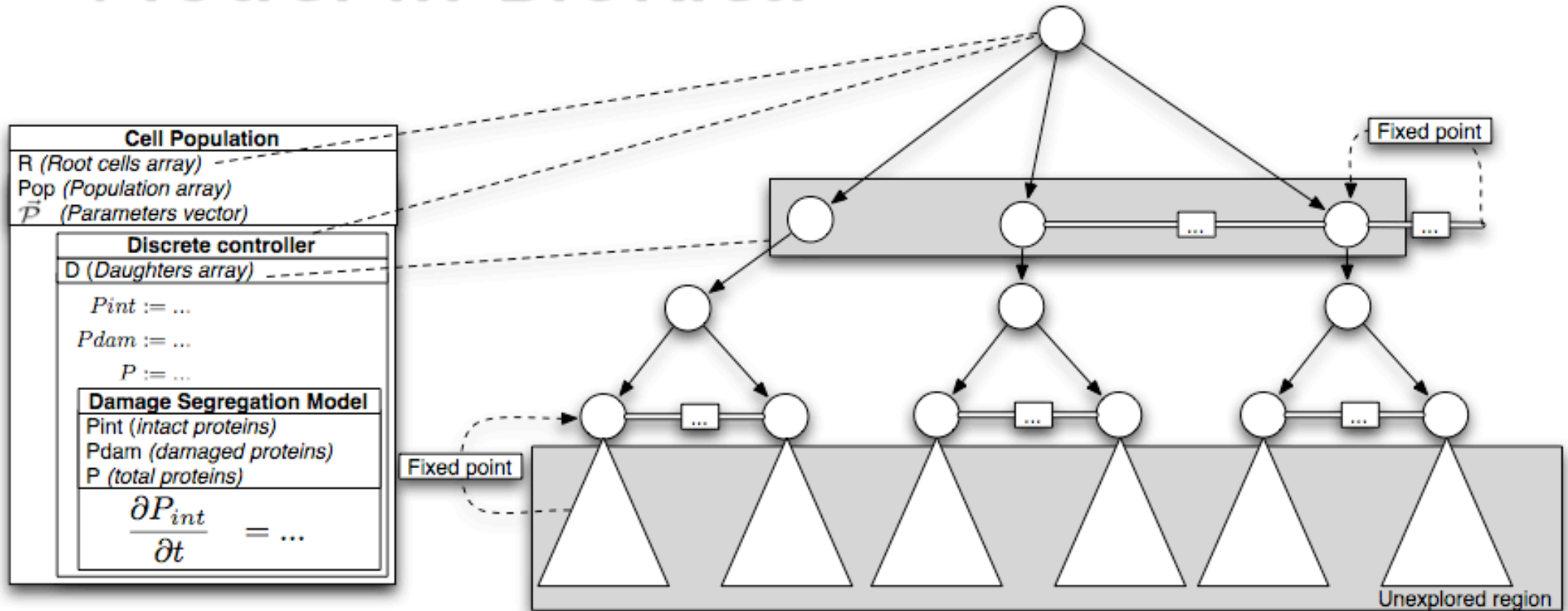
$$P_{int} \rightarrow P_{int} \cdot mother \cdot (1 - re \cdot \frac{P_{dam}}{P_{int}} \cdot \frac{1 - mother}{mother})$$

$$P_{dam} \rightarrow P_{dam} \cdot (mother + (1 - mother) \cdot re)$$

Pedigree analysis



Model in BioRica



Discrete events: division ($P_{int} > 1500$) and clonal senescence ($\partial P_{int} < 0$)

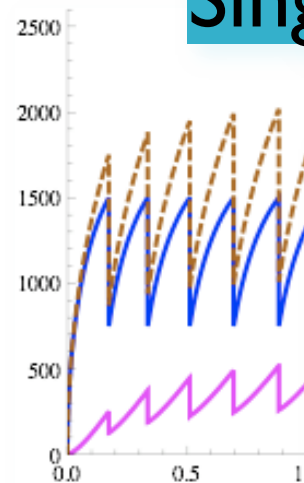
alive: immortality, senescence

update: state vars of the mother, statistics; create daughter, add to D and P

Rejuvenation in a cell ageing model

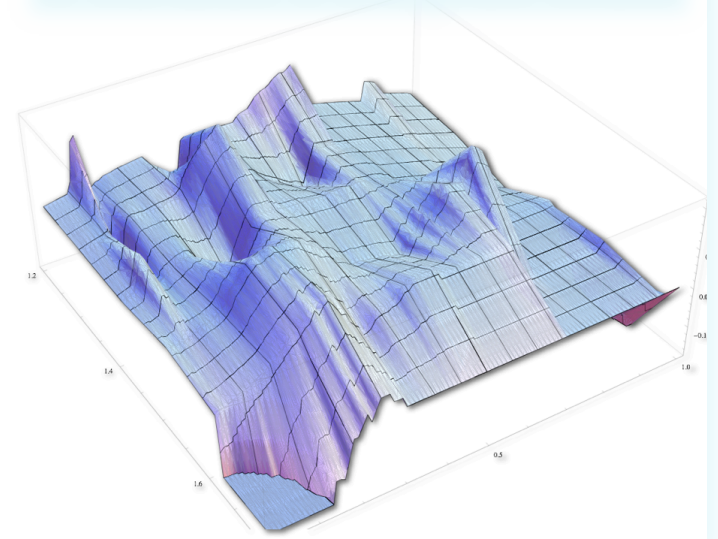
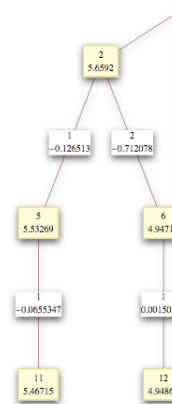
- ▶ Mathematica and BioRica
- ▶ Maximization of rejuvenation for the whole population
- ▶ Three layer approach

Single Cell



Population

Model Properties



Study outline

Calibrate and validate the hierarchical model: 625 simulations

- Complete simulations to depth 4
- Symmetric and asymmetric, k_1 , k_2 and k_4 fixed, k_3 step $0.1 \in [0.1, 2.3]$

Simulation to depth 30

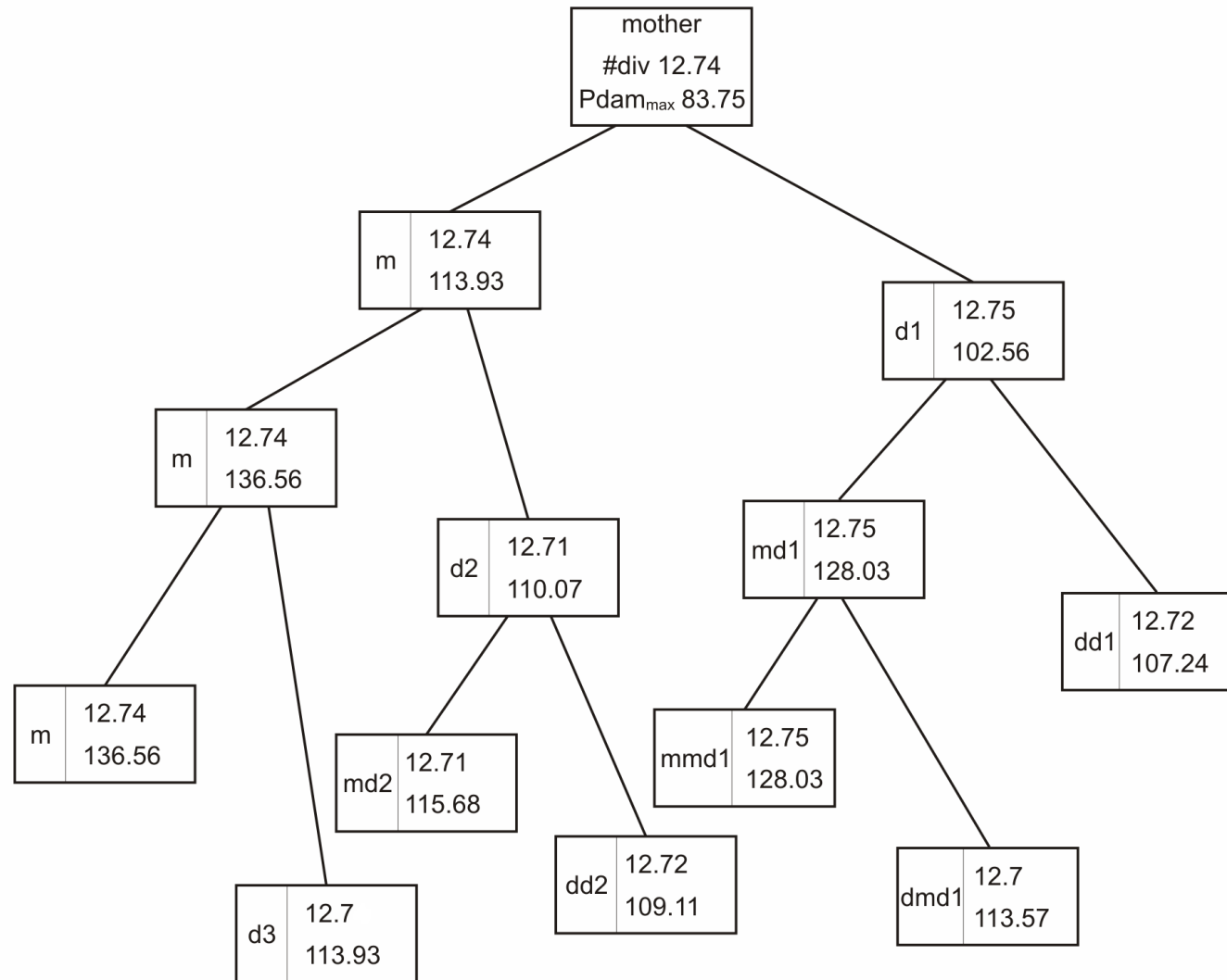
- Select simulations: #daughters 20-24
- For each cell: in a cell cycle P_{dam} *initial* and *final*, *generation time*, *date of birth* (arb. time units), *fitness* (#div in 1st time unit)

→ 2^{30} cells in parallel, 2^{32} diff. equations

Parameter exploration

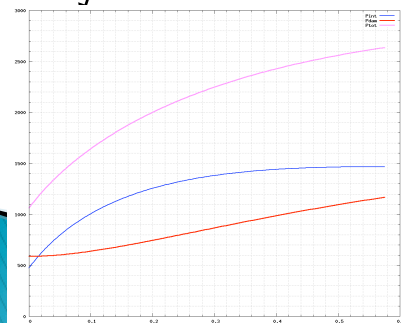
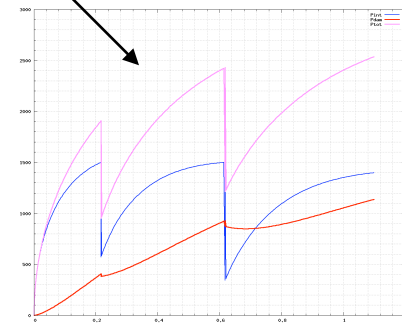
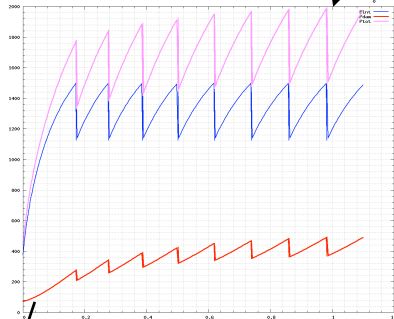
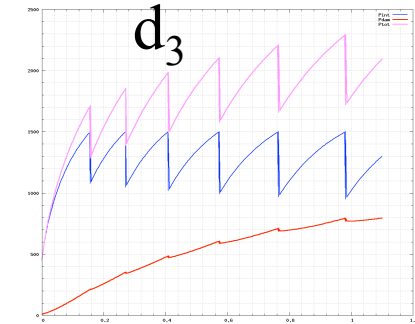
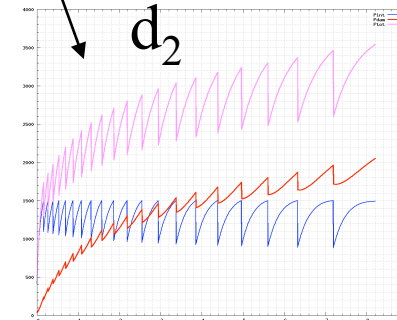
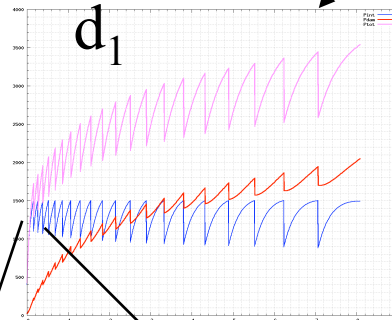
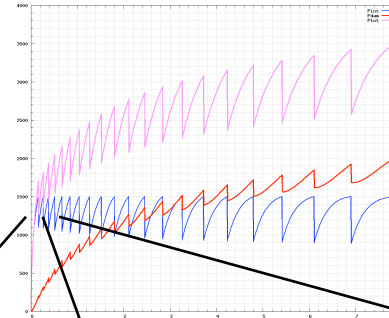
- Identify sets of parameters that exhibit emerging behaviour
- Ex: #daughters, high rejuvenation value

Pedigree analysis



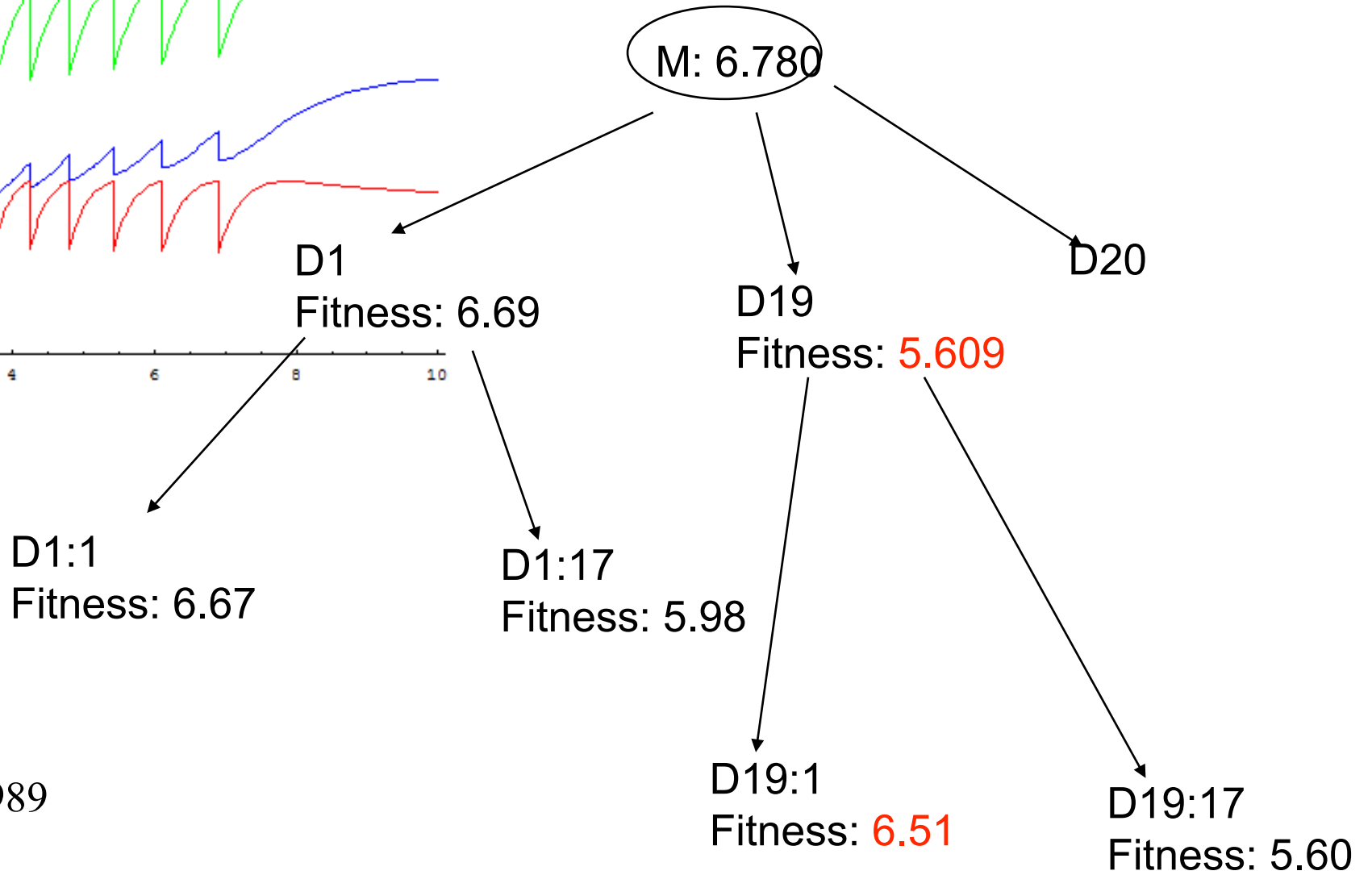
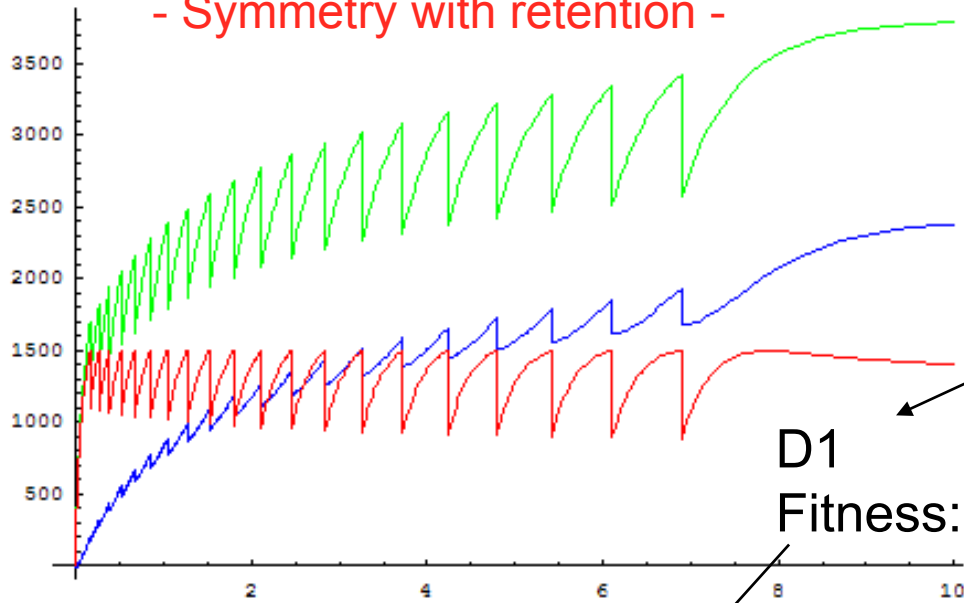
Pedigree analysis

mother



Rejuvenation

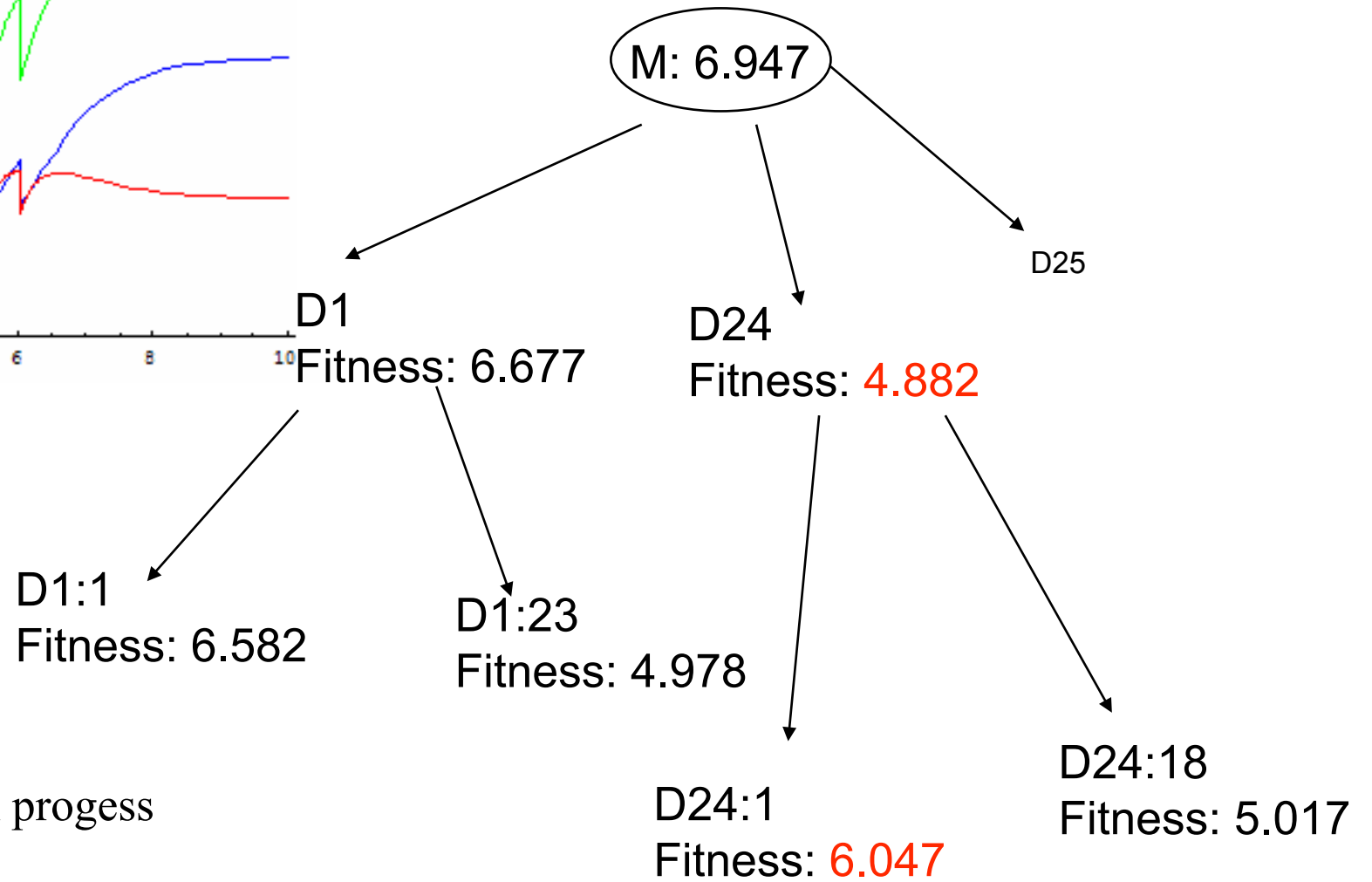
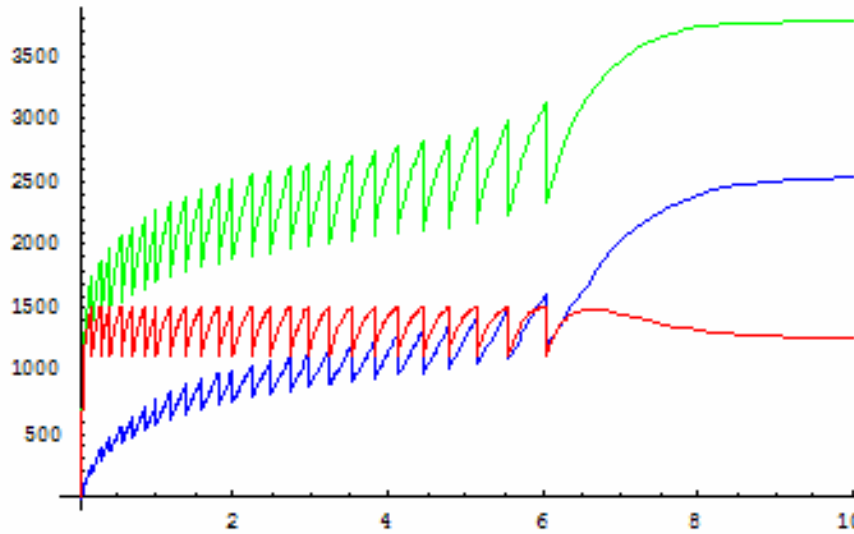
- Symmetry with retention -



Eglimez *et al*, 1989

Rejuvenation

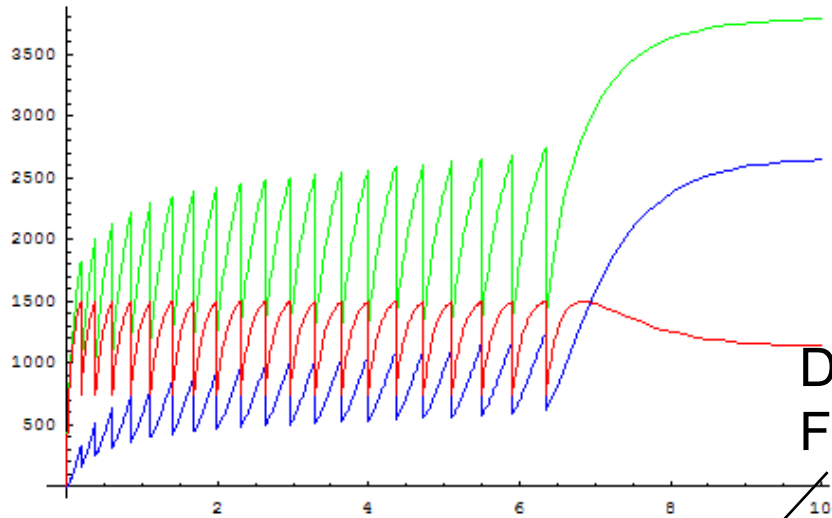
- Asymmetry without retention -



Nyström *et al*, work in progress

Rejuvenation

- Symmetry without retention -



D1:1
Fitness: 3.962

D1:18
Fitness: 0

M: 4.563

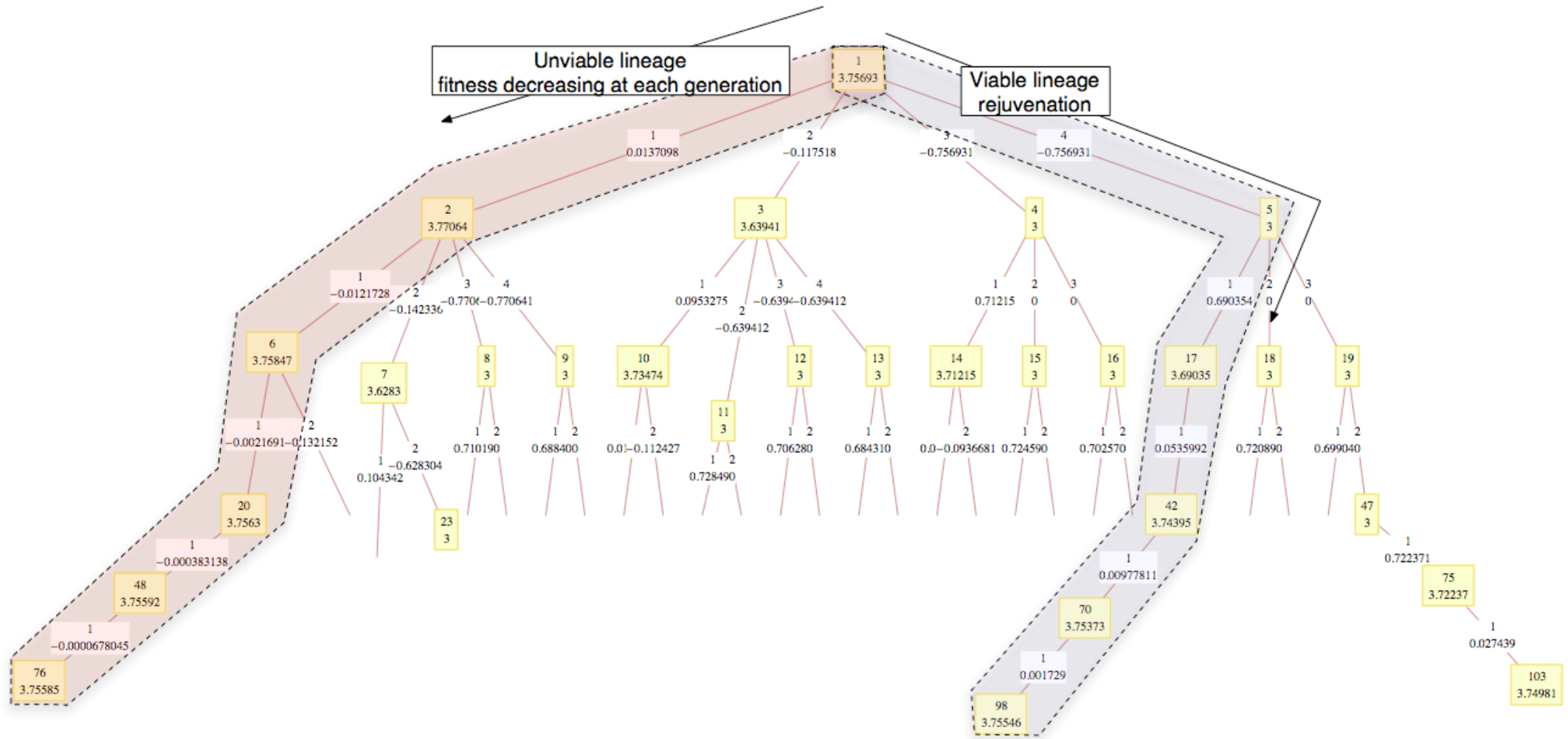
D11
Fitness: 2.827

D11:1
Fitness: 2.765

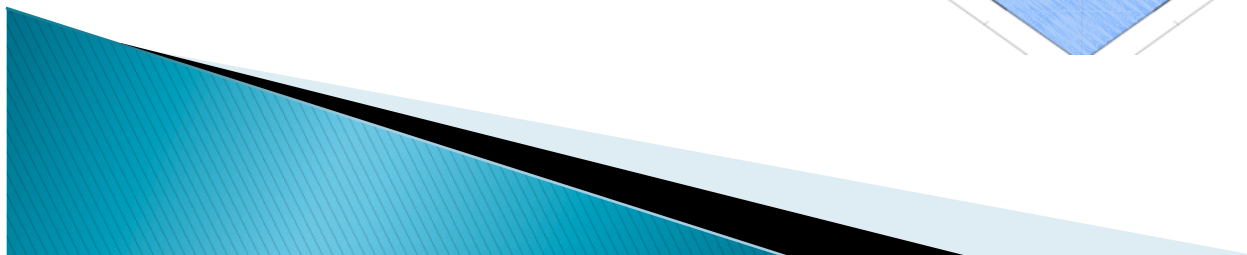
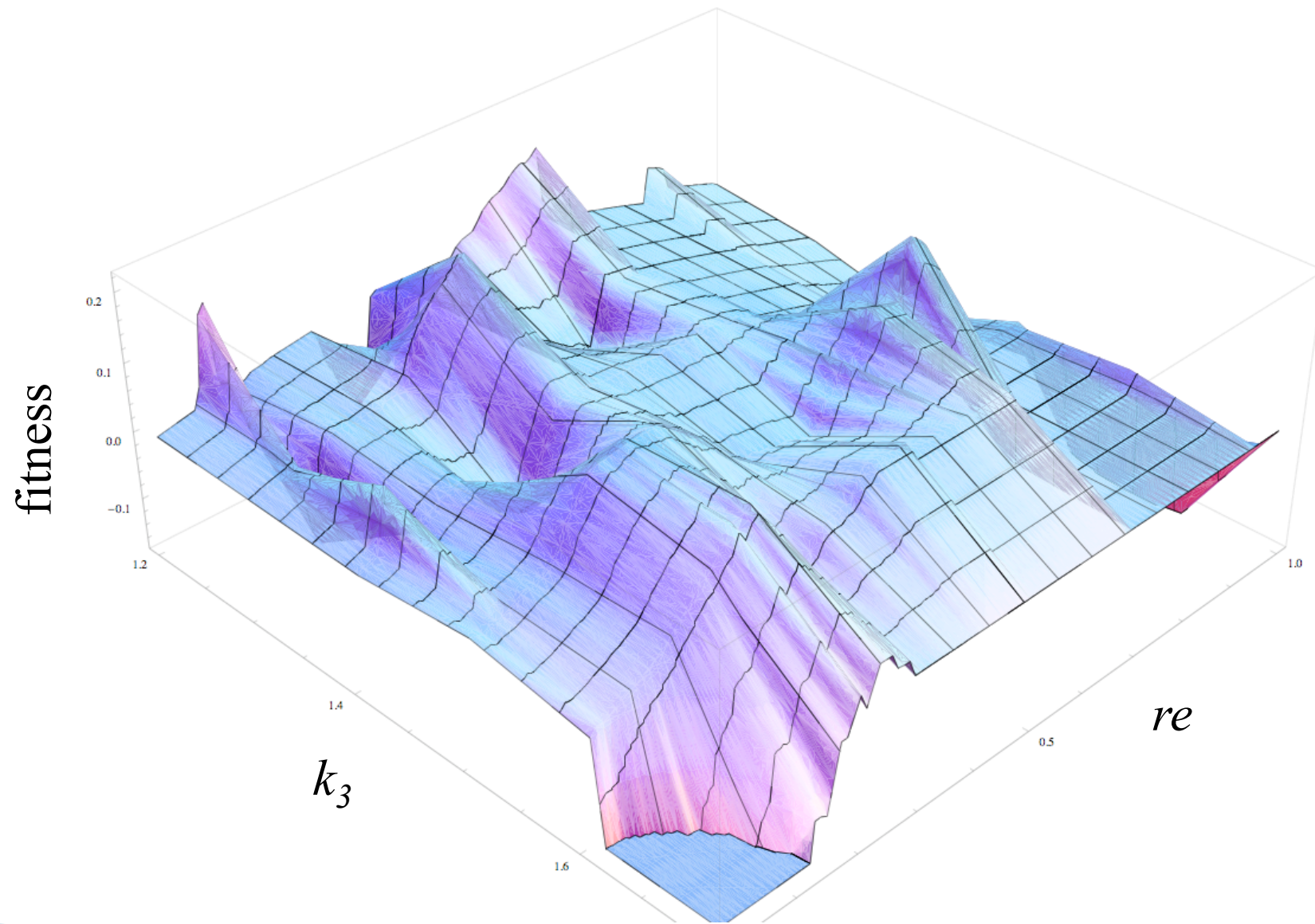
D11:7
Fitness: 0

D20

Exploring lineage viability



Compare whole populations



People

Original model MPI, Berlin

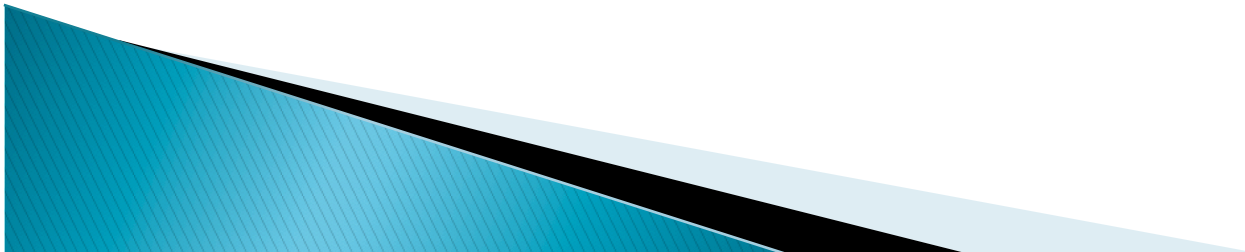
Marija Cvijovic, Edda Klipp, Thomas Nyström

Experimental work CMB, Goteborg

Marija Cvijovic, Thomas Nyström

Hierarchical model Bordeaux

Macha Nikolski, David Sherman, Grégoire Sutre



Summary

- Younger siblings born "prematurely old" (Kennedy *et. al.*, 1994)
 - Daughters born early have low damage, and their daughters have normal fitness
 - Both asymmetry and retention increase fitness in second generation
 - Control: symmetry, no retention
 - Exponential increase in generation time for symmetry and retention (Egilmez *et al.*, 1989)
 - Linear increase in generation time for asymmetry and no retention (T. Nystrom, work in progress on Δ Sir2 mutant)
 - Fitness and viability sensitive to k_3
- 