

# Modeling tutorial

INRA workshop

Tech talks: Cell representation and RBA file formats

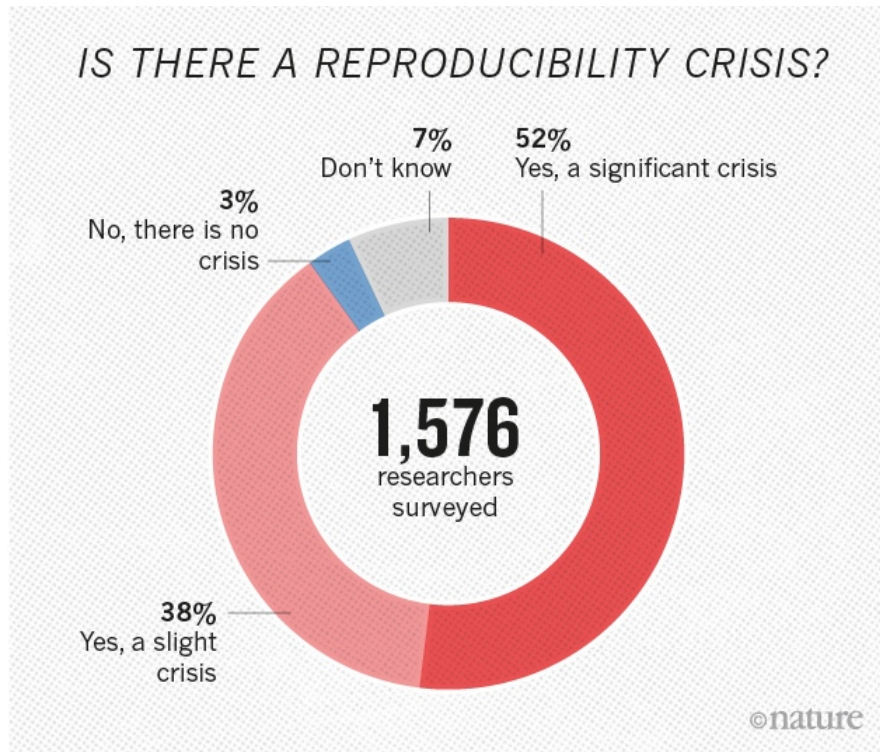
Dr. Matthias König

LiSyM Systems Medicine of the Liver

Humboldt University Berlin, Institute for Theoretical Biology

[livermetabolism.com](http://livermetabolism.com), [konigmatt@googlemail.com](mailto:konigmatt@googlemail.com)

# Reproducibility Crisis



## **“Really Reproducible Research” (1992)**

inspired by Stanford Professor Jon Claerbout:

“The idea is: An article about computational science in a scientific publication is not the scholarship itself, it is merely advertising of the scholarship. The actual scholarship is the complete ... set of instructions [and data] which generated the figures.”

*David Donoho, 1998*

Note the difference between: reproducing the computational steps and, replicating the experiments independently including data collection and software implementation.

(Both required)

<https://www.nature.com/news/1-500-scientists-lift-the-lid-on-reproducibility-1.19970>

<https://web.stanford.edu/~vcs/talks/MSKCC-Sept15-2015-STODDEN.pdf>

# Code repositories

**GitHub** is a code hosting platform for version control and collaboration. It lets you and other work together on projects from anywhere.

- **Version control**
  - Diffs & Branches
- **Collaborative editing**
  - Pull requests
- **Continuous integration**
  - unit tests
  - Commit hooks
- **Releases & snapshots** (citable code)
- **Issue tracker**
- **Work anywhere & offline**

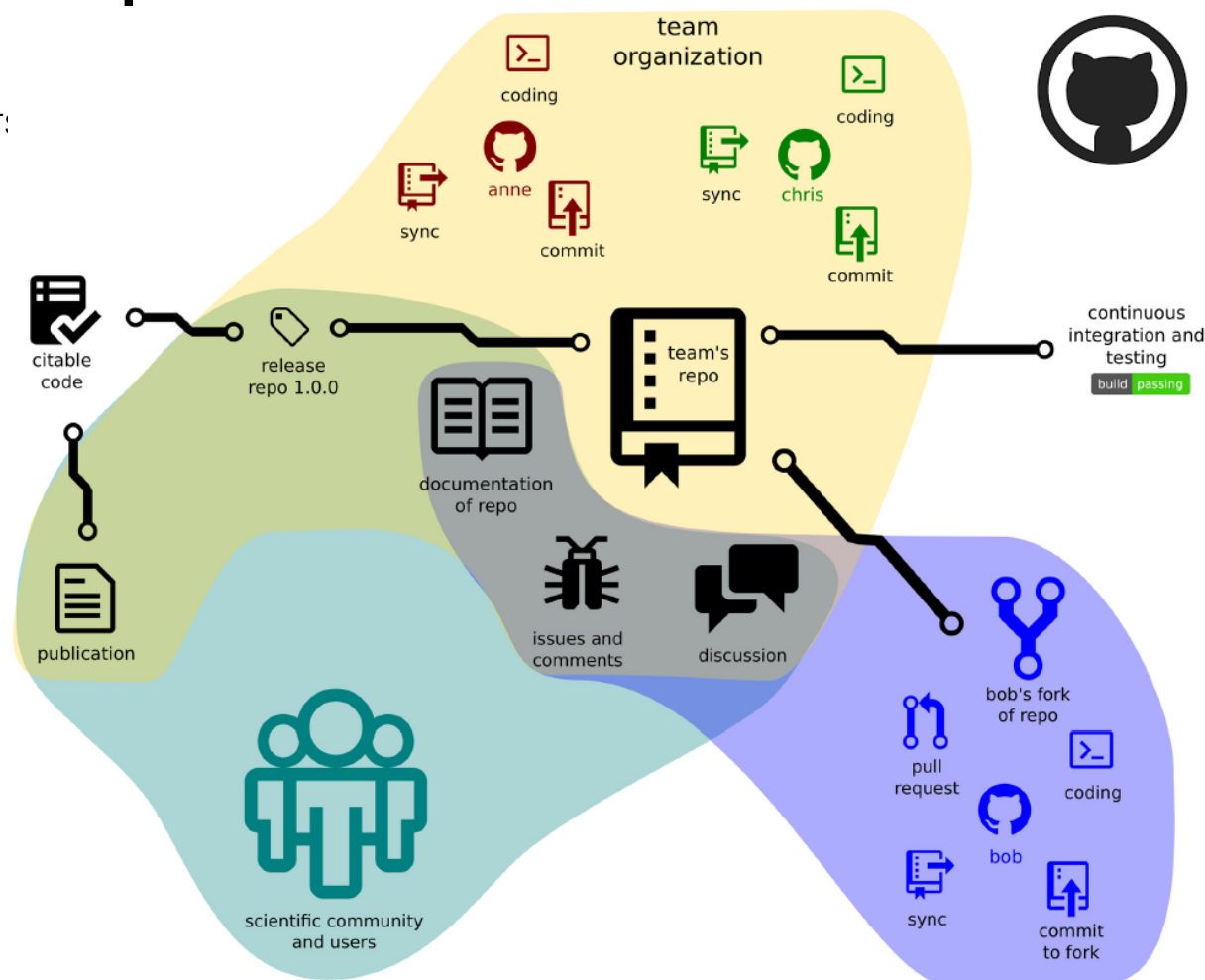


**CODE**

&



**DATA**



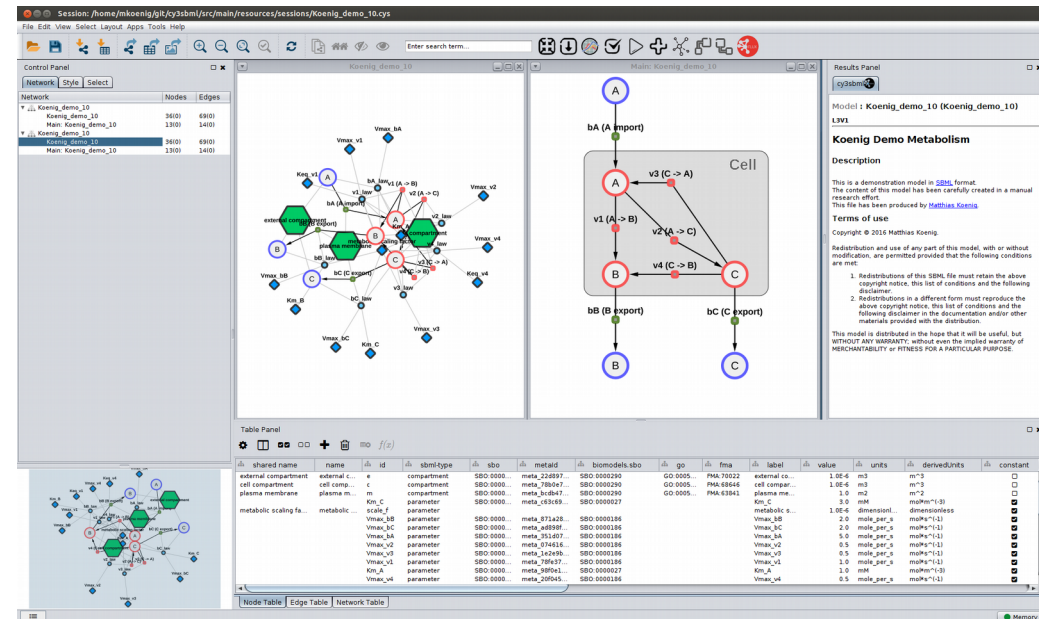
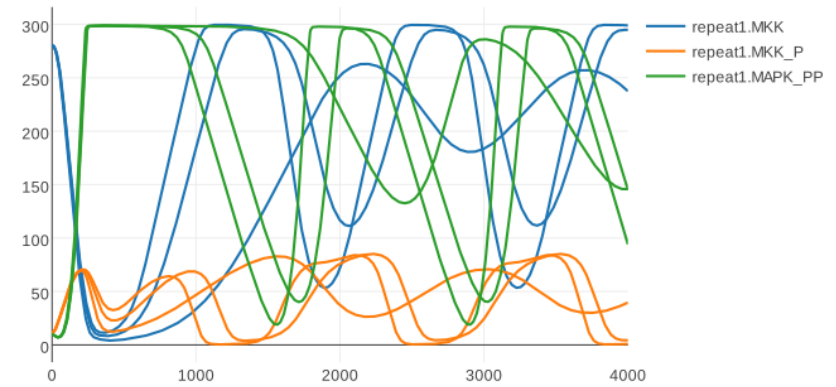
**Fig 1. The structure of a GitHub-based project illustrating project structure and interactions with the community.**

doi:10.1371/journal.pcbi.1004947.g001

Perez.Riverol2016

# Standard Formats

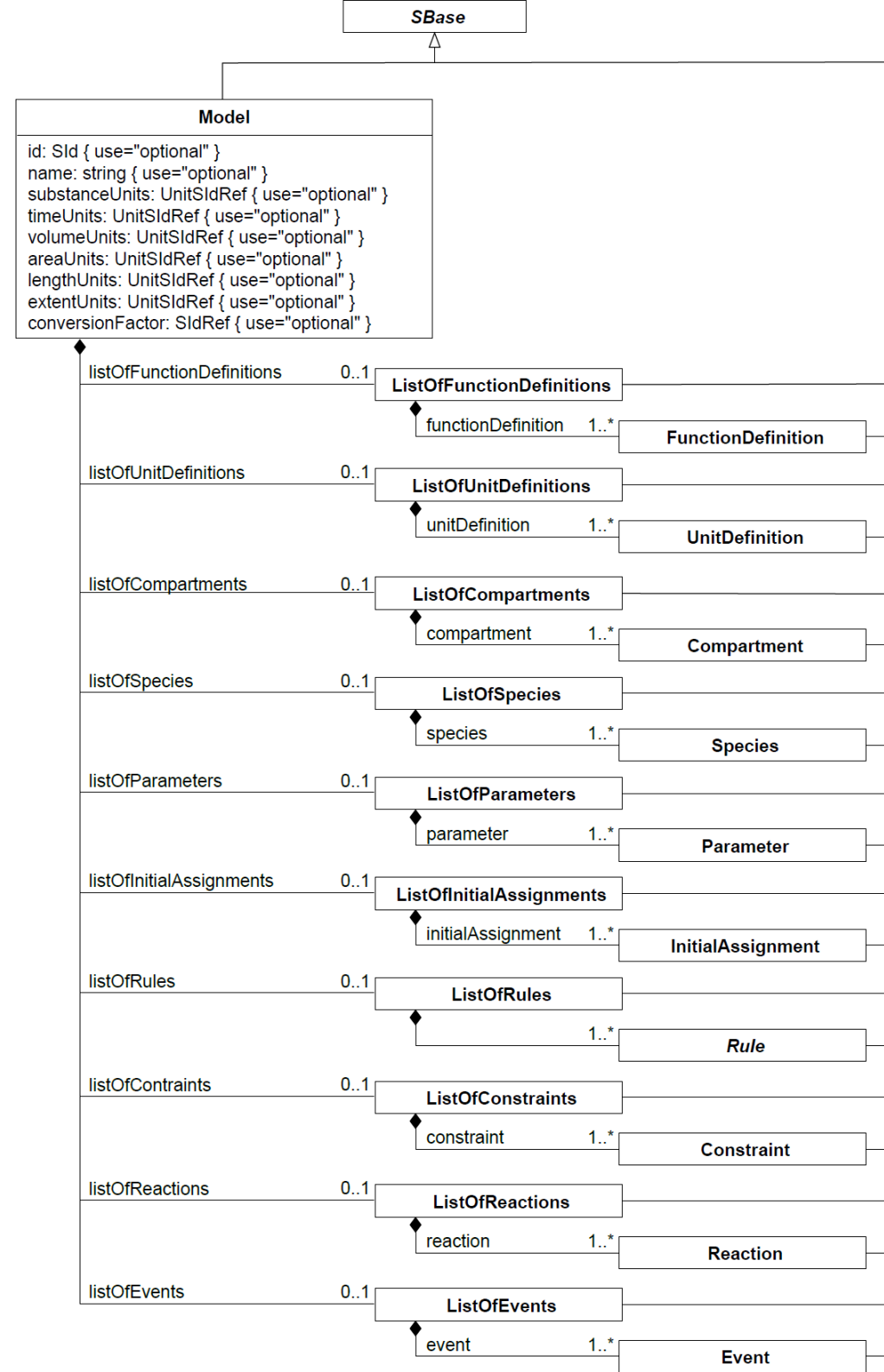
- Encoding of information in computer readable format
  - Minimal Information for models and simulation (MIRIAM, MIASE)
- Exchangeability
- Reproducibility
  - Reproducibility of results (roadrunner, COPASI, JWS)
- Annotations (to ontologies)
  - Knowledge integration (biological, computational)
  - Documentation (what is my model component)
- Quality
  - Automatic validation, e.g., model quality (unit checking, model consistency)
- Software & Tools
  - Simulation, Visualization, Analysis, Repositories, Parameter fitting



# SBML



- Systems Biology Markup Language
- De facto standard for encoding computational models
- Encoding the Math of a model in a computer-readable and exchangeable format
- Libraries: libsbml (C++, python, R, JavaScript, ... & JSBML (Java)

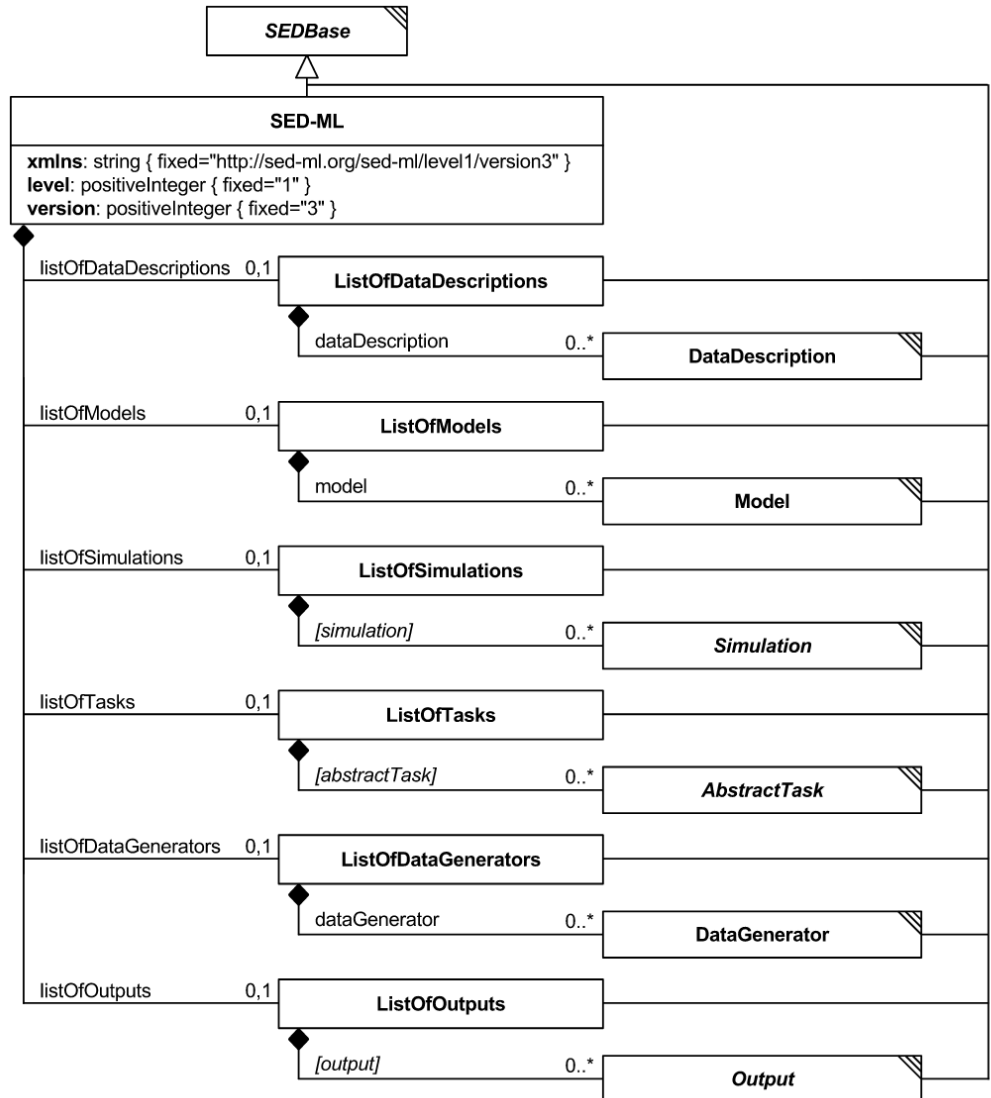
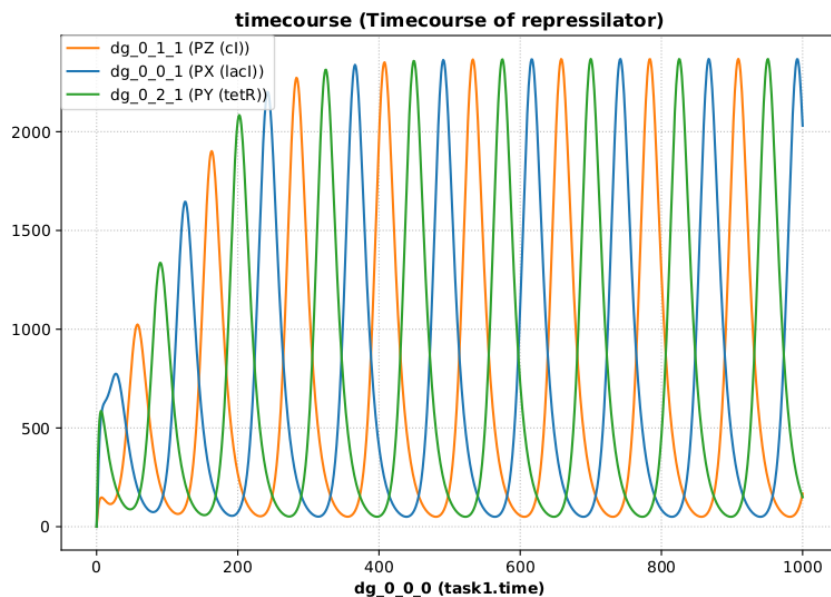


# SED-ML

- Simulation Experiment Description Markup Language (SED-ML)

<https://sed-ml.github.io>

- SED-ML is an XML-based format for encoding simulation setups, to ensure exchangeability and reproducibility of simulation experiments.





# Reproducible Analyses

Dynamic report generation !

Jupyter notebooks



- Web application that allows to create and share documents that contain live code, equations, visualizations and explanatory text.
- Open source, interactive data science and scientific computing across over 40 programming languages.
- Complete analysis, examples, test cases  
<http://tellurium.readthedocs.io/en/latest/notebooks.html#feedback-oscillations>  
[http://localhost:8888/notebooks/feedback\\_oscillations.ipynb](http://localhost:8888/notebooks/feedback_oscillations.ipynb)

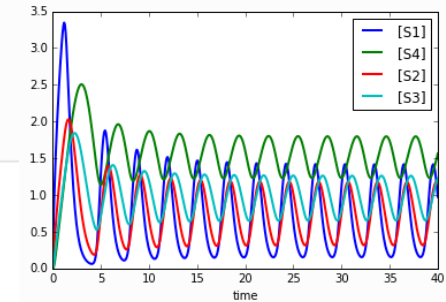
R: Knitr



- Elegant, flexible and fast dynamic report generation with R
- BDL example  
<https://github.com/matthiaskoenig/bdl-analysis>

## Feedback oscillations

Model oscillations via feedback



```
import tellurium as te

r = te.loada('''
model feedback()
// Reactions:
J0: $X0 -> S1; (VM1 * (X0 - S1/Keq1))/(1 + X0 + S1 + S4^h);
J1: S1 -> S2; (10 * S1 - 2 * S2) / (1 + S1 + S2);
J2: S2 -> S3; (10 * S2 - 2 * S3) / (1 + S2 + S3);
J3: S3 -> S4; (10 * S3 - 2 * S4) / (1 + S3 + S4);
J4: S4 -> $X1; (V4 * S4) / (KS4 + S4);

// Species initializations:
S1 = 0; S2 = 0; S3 = 0;
S4 = 0; X0 = 10; X1 = 0;

// Variable initialization:
VM1 = 10; Keq1 = 10; h = 10; V4 = 2.5; KS4 = 0.5;
end''')

res = r.simulate(0, 40, 500)
r.plot()

import matplotlib.pyplot as plt
plt.plot(res["[S1]"], res["[S2]"], 'o-', color="black")
plt.xlabel("[S1]")
plt.ylabel("[S2]");
```

```

[3] // -- Begin Antimony block converted from MAPKcascade.xml
// Created by libAntimony v2.9.3
model *MAPKcascade()
...
// Reactions:
J0: MKKK => MKKK_P; J0_V1*MKKK/((1 + (MAPK_PP/J0_Ki)^J0_n)*(J0_K1 + MKKK));
J1: MKKK_P => MKKK; J1_V2*MKKK_P/(J1_KK2 + MKKK_P);
J2: MKK => MKK_P; J2_k3*MKKK_P*MKK/(J2_KK3 + MKK);
J3: MKK_P => MKK_PP; J3_k4*MKKK_P*MKK_P/(J3_KK4 + MKK_P);
J4: MKK_PP => MKK_P; J4_V5*MKK_PP/(J4_KK5 + MKK_PP);
J5: MKK_P => MKK; J5_V6*MKK_P/(J5_KK6 + MKK_P);
J6: MAPK => MAPK_P; J6_k7*MKK_PP*MAPK/(J6_KK7 + MAPK);
J7: MAPK_P => MAPK_PP; J7_k8*MKK_PP*MAPK_P/(J7_KK8 + MAPK_P);
J8: MAPK_PP => MAPK_P; J8_V9*MAPK_PP/(J8_KK9 + MAPK_PP);
J9: MAPK_P => MAPK; J9_V10*MAPK_P/(J9_KK10 + MAPK_P);
...
end
// -- End Antimony block

// -- Begin PhraSEDML block converted from main.xml
// Created by libphrasedml v1.0.7
// Models
modell = model "MAPKcascade"

// Simulations
sim1 = simulate uniform(0, 4000, 1000)

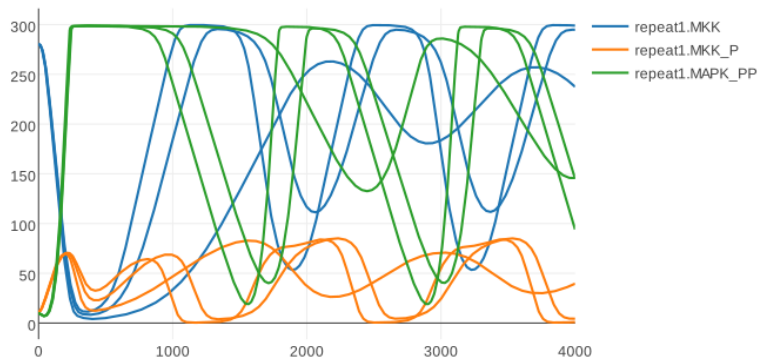
// Tasks
task1 = run sim1 on modell

// Repeated Tasks
repeat1 = repeat task1 for modell.J1_KK2 in [1, 10, 40], reset=true

// Outputs
plot "Sampled Simulation" repeat1.time vs repeat1.MKK, repeat1.MKK_P, repeat1.MAPK_PP
// -- End PhraSEDML block

```

Sampled Simulation



# Modeling Tools

- **libRoadRunner**: High performance SBML simulator
- **tellurium**: Python based modeling environment (library & notebook)  
<http://tellurium.analogmachine.org/>  
<https://tellurium.readthedocs.io/en/latest/>
- Medley K, Choi K, König M., Smith L, Gu S, Joseph Hellerstein, Sealfon S., Sauro HM.  
*Tellurium Notebooks - An Environment for Dynamical Model Development, Reproducibility, and Reuse*  
**[2017-11-14, submitted]**  
 Somogyi, Bouteiller, Glazier, König, Medley, Swat, Sauro.  
*libRoadRunner: a high performance SBML simulation and analysis library.*  
 Bioinformatics. **2015**  
 König M., Janek G., Medley K, Kiri Choi, Sauro H  
*Tellurium-web*  
**[2017, in preparation]**



# What kind of models?

- Ordinary differential equations (ODE)
  - Metabolism
  - Transcription
  - Translation
  - Signaling
- Well stirred container

# Showcase 1-Repressilator

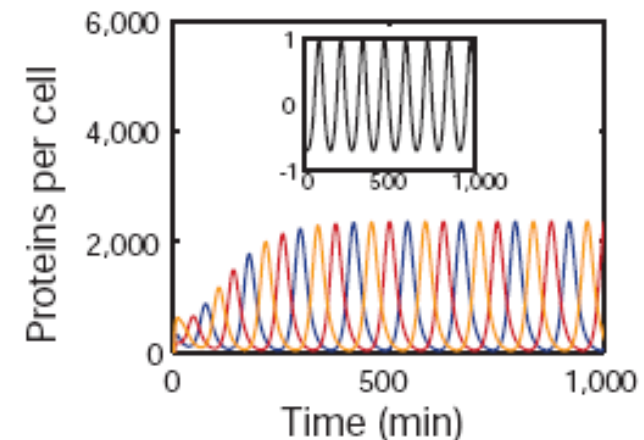
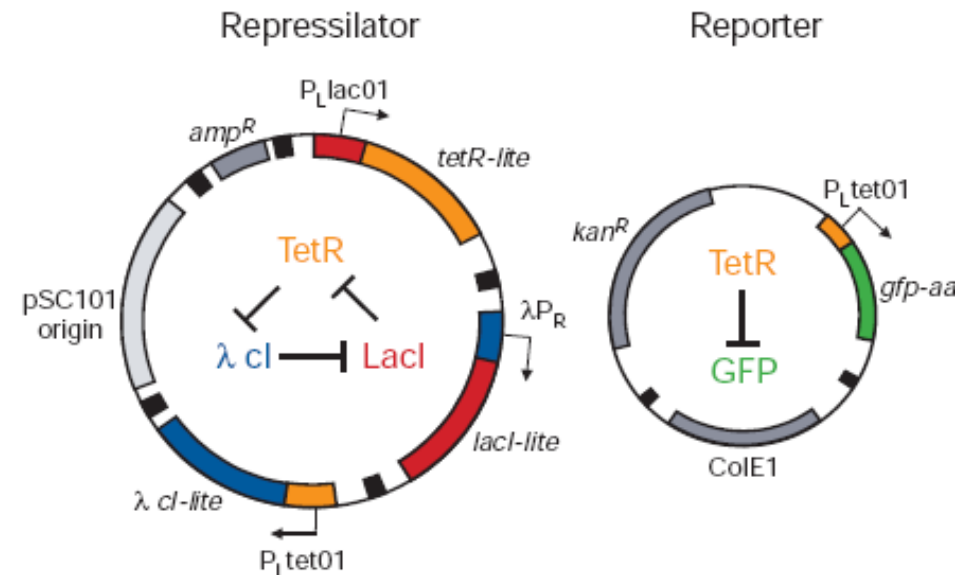
- Understanding a model from the literature

<https://wwwdev.ebi.ac.uk/biomodels/>

- 

- Repressilator

- Download model  
<https://www.ebi.ac.uk/biomodels-main/static-page-006-07>
- Visualize model
- Model report
- Simulate model with tellurium (loading)
- Change the parameters
- Stochastic simulations



# Model building

- Simple linear chain ( $S1 \xrightarrow{v1} S2$ )
  - **Mass-Action kinetics**
  - Stochastic Matrix
  - Steady State
  - Jacobian Matrix
  - Elasticities
- Michaelis-Menten Kinetics (irreversible & reversible)
- Hill equations (cooperativity)
- Competitive Inhibition
- Running models with SED-ML

# Map kinase network

- Stochastic simulation



- initiative to coordinate the development of the various community standards and formats for computational models
- COMBINE meeting & HARMONY hackathon
- Core standards

[http://co.mbine.org/Standards#Core\\_COMBINE\\_standards](http://co.mbine.org/Standards#Core_COMBINE_standards)

- **SBML** Systems Biology Markup Language
- **SED-ML** Simulation Experiment Description Language (SED-ML)
- **SBGN** Systems Biology Graphical Notation
- **CellML**
- **SBOL** Synthetic Biology Open Language Data
- **NeuroML**





# COMBINE archive

- A COMBINE archive is a single file bundling the various documents necessary for a modeling and simulation project, and all relevant information. The archive is encoded using the **Open Modeling EXchange format (OMEX)**.

