Identification of cell signaling pathways based on biochemical reaction kinetics repositories

Student: Gustavo Estrela

Advisor: Marcelo da Silva Reis (Butantan Institute)

May 2019

Instituto de Matemática e Estatística
Centro de Toxinas, Resposta-imune e Sinalização Celular (CeTICS)
Laboratório Especial de Ciclo Celular, Instituto Butantan

This project receives funding from FAPESP

Introduction

Cell Signaling

Cell signaling allows cells to respond to signals that come from its environment changing its behaviour accordingly.

Cell Signaling

Cell signaling allows cells to respond to signals that come from its environment changing its behaviour accordingly.

This mechanism is essential for many cell functions, including reproduction, growth and death.

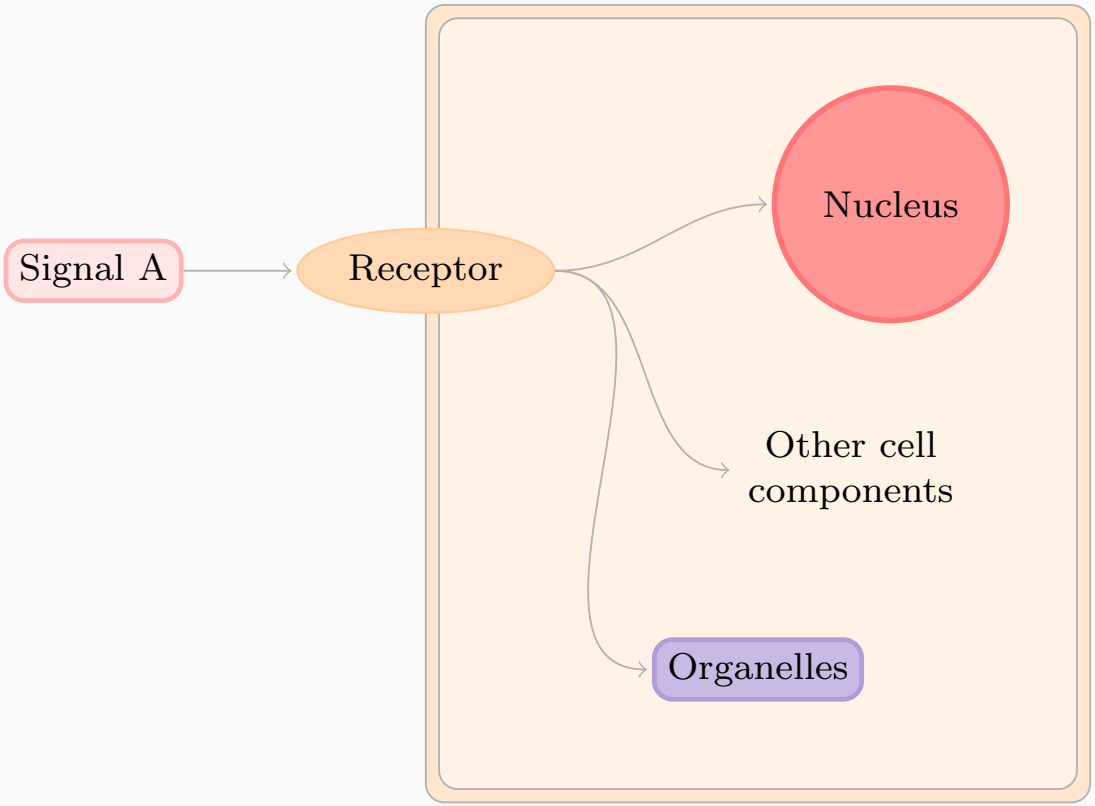
Cell Signaling

Cell signaling allows cells to respond to signals that come from its environment changing its behaviour accordingly.

This mechanism is essential for many cell functions, including reproduction, growth and death.

Understanding the functioning of cell signaling is important in many biological areas.

Cell Signaling



Cell Signaling

A signal propagates in an organism though chemical reactions that are caused by the change of concentration of chemical species.

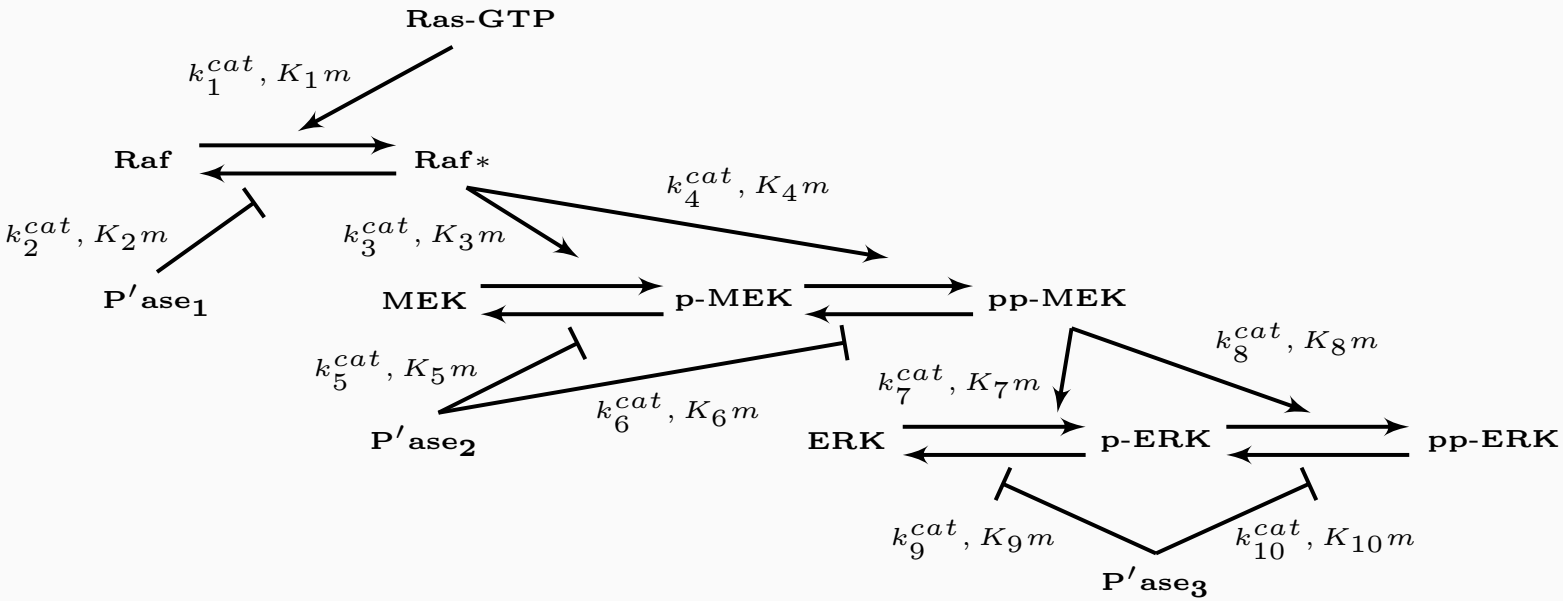
Cell Signaling

A signal propagates in an organism though chemical reactions that are caused by the change of concentration of chemical species.

We call the path of a signal a cell signaling pathway.

Cell Signaling Pathways

A cell signaling network can be characterized by a sequence of chemical reactions



Mathematical Models of Signaling Networks

We can summarize the state of the cell with measurements based on the concentration of some chemical species.

Mathematical Models of Signaling Networks

We can summarize the state of the cell with measurements based on the concentration of some chemical species.

Using biochemical and enzymatic kinetics, we can model the concentration change of chemical species over time of a pathway.

Identification of Cell Signaling Pathways

The problem of identification of cell signaling pathways is the problem of finding the components of a signaling pathway and how they interact given a set of experimental measurement.

Identification of Cell Signaling Pathways

The problem of identification of cell signaling pathways is the problem of finding the components of a signaling pathway and how they interact given a set of experimental measurement.

As the input, a description of a biological experiment and a set of experimental measurements are given.

Identification of Cell Signaling Pathways

The problem of identification of cell signaling pathways is the problem of finding the components of a signaling pathway and how they interact given a set of experimental measurement.

As the input, a description of a biological experiment and a set of experimental measurements are given. A possible output to the problem is composed by:

Identification of Cell Signaling Pathways

The problem of identification of cell signaling pathways is the problem of finding the components of a signaling pathway and how they interact given a set of experimental measurement.

As the input, a description of a biological experiment and a set of experimental measurements are given. A possible output to the problem is composed by:

- a model composed by a set of chemical reactions that are relevant for the biological experiment;

Identification of Cell Signaling Pathways

The problem of identification of cell signaling pathways is the problem of finding the components of a signaling pathway and how they interact given a set of experimental measurement.

As the input, a description of a biological experiment and a set of experimental measurements are given. A possible output to the problem is composed by:

- a model composed by a set of chemical reactions that are relevant for the biological experiment;
- information about the reaction rate constants of the model.

Identification of Cell Signaling Pathways

One can search for the set of chemical reactions relevant for a biological experiment in repositories like the Kyoto Encyclopedia of Genes and Genomes (KEGG).

Identification of Cell Signaling Pathways

One can search for the set of chemical reactions relevant for a biological experiment in repositories like the Kyoto Encyclopedia of Genes and Genomes (KEGG). However, the pathway maps from KEGG may be incomplete or have impertinent reactions for the biological experiment of interest.

Identification of Cell Signaling Pathways

One can search for the set of chemical reactions relevant for a biological experiment in repositories like the Kyoto Encyclopedia of Genes and Genomes (KEGG). However, the pathway maps from KEGG may be incomplete or have impertinent reactions for the biological experiment of interest.

Hence, it is desirable to construct a method that can systematically modify these models and choose the one that better represents the experiment.

Identification of Cell Signaling Pathways

Lulu Wu (2015) presented in her master dissertation a methodology that proposes to systematically modify models of signaling network in order to better represent experiments.

Identification of Cell Signaling Pathways

Lulu Wu (2015) presented in her master dissertation a methodology that proposes to systematically modify models of signaling network in order to better represent experiments.

On her work, the problem of identification of cell signaling pathways is treated as a feature selection problem.

Feature Selection for Identification of Signaling Pathways

The methodology proposed by Wu defines the set of features as a set of chemical reactions that can be added to a starting model.

Feature Selection for Identification of Signaling Pathways

The methodology proposed by Wu defines the set of features as a set of chemical reactions that can be added to a starting model.This set of chemical reactions is fetched from KEGG and stored in a database of interactions.

Wu's Search Algorithm for Feature Selection

The search algorithm used by Wu is the Sequential Forward Selection (SFS).

Wu's Cost Function for Feature Selection

Wu defines the cost function as the minimum distance between experimental and simulated data.

Wu's Cost Function for Feature Selection

Wu defines the cost function as the minimum distance between experimental and simulated data. The minimum distance is found using a Simmulated Annealing that traverses the parameter space.

Results of Wu's Methodology

Lulu Wu tested her methodology by trying to recreate models given a cut of the original model.

Results of Wu's Methodology

Lulu Wu tested her methodology by trying to recreate models given a cut of the original model. However, the methodology worked satisfactorily only when the cut was similar to the original model.

Difficulties of Wu's Methodology

We can point three aspects of Wu's work that could explain its limitations.

Difficulties of Wu's Methodology

We can point three aspects of Wu's work that could explain its limitations.

- the database of interactions used could be more nearly complete;

Difficulties of Wu's Methodology

We can point three aspects of Wu's work that could explain its limitations.

- the database of interactions used could be more nearly complete;
- the search algorithm could also consider removing interactions;

Difficulties of Wu's Methodology

We can point three aspects of Wu's work that could explain its limitations.

- the database of interactions used could be more nearly complete;
- the search algorithm could also consider removing interactions;
- the cost function could implement a proper penalization of models;

What we Propose on this Project

We propose to create a methodology that uses a feature selection approach for identification of signaling pathways, tackling the difficulty of penalizing complex models.

What we Propose on this Project

We intend to use Bayesian approaches of model selection that allow us to create estimates of $p(M|D)$ or $p(D|M)$.

Objectives of this Project

Objectives of this Project

- Study state of the art Bayesian algorithms for signaling network model selection.

Objectives of this Project

- Study state of the art Bayesian algorithms for signaling network model selection.
- Implementation and comparison of cost functions for model selection.

Objectives of this Project

- Study state of the art Bayesian algorithms for signaling network model selection.
- Implementation and comparison of cost functions for model selection.
- Formulate systematic modifications to a model as the search space of a feature selection model.

Objectives of this Project

- Study state of the art Bayesian algorithms for signaling network model selection.
- Implementation and comparison of cost functions for model selection.
- Formulate systematic modifications to a model as the search space of a feature selection model.
- Observe the surface induced by the cost function over the search space.

Fundamental Concepts

Kinetics Modeling of Chemical Reactions

Mathematical Modeling of Reactions

In this project we use three possible models of kinetics of an interaction:

Mathematical Modeling of Reactions

In this project we use three possible models of kinetics of an interaction:

- first order interaction kinetics;

Mathematical Modeling of Reactions

In this project we use three possible models of kinetics of an interaction:

- first order interaction kinetics;
- second order interaction kinetics;

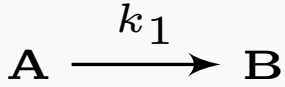
Mathematical Modeling of Reactions

In this project we use three possible models of kinetics of an interaction:

- first order interaction kinetics;
- second order interaction kinetics;
- Michaelis-Menten enzymatic kinetics.

Kinetic Modeling of First Order Iteration

A first order reaction:



Kinetic Modeling of First Order Iteration

A first order reaction:

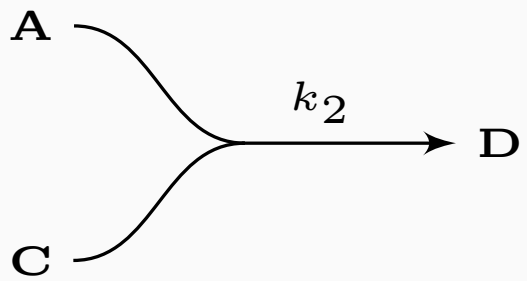
$$\mathbf{A} \xrightarrow{k_1} \mathbf{B}$$

has rate of:

$$k_1[\mathrm{A}].$$

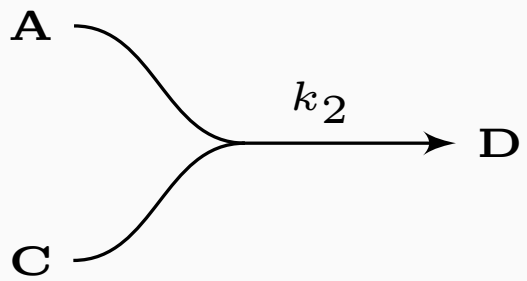
Kinetic Modeling of Second Order Iteration

A second order reaction:



Kinetic Modeling of Second Order Iteration

A second order reaction:

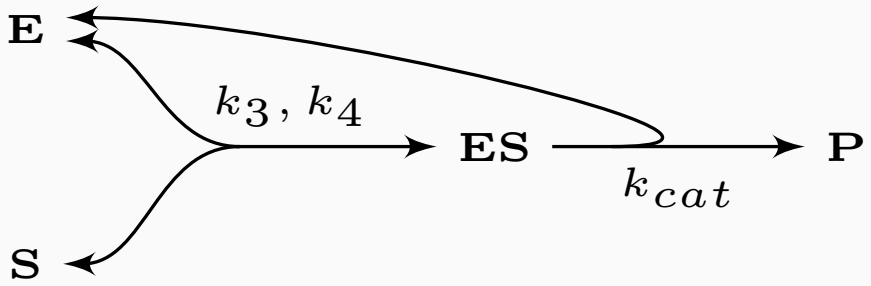


has rate of:

$$k_2[\mathrm{A}][\mathrm{C}].$$

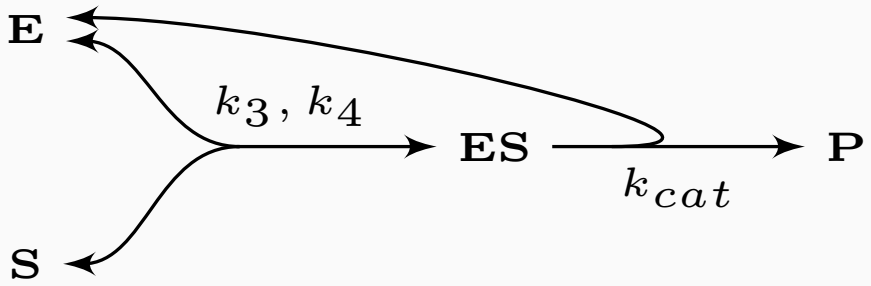
Kinetic Modeling of Enzymatic Reactions

An enzymatic reaction:



Kinetic Modeling of Enzymatic Reactions

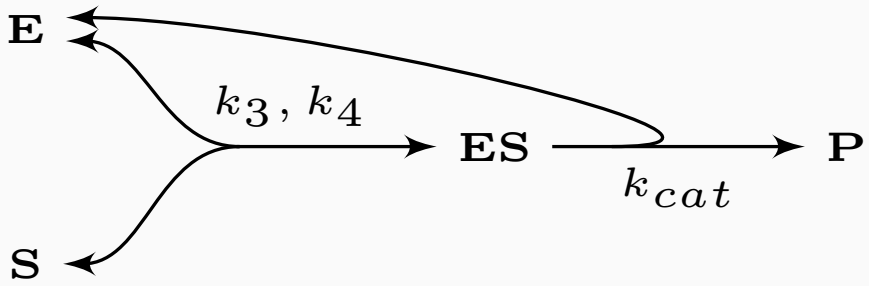
An enzymatic reaction:



Can be divided in two first order reactions plus a second order reaction.

Kinetic Modeling of Enzymatic Reactions

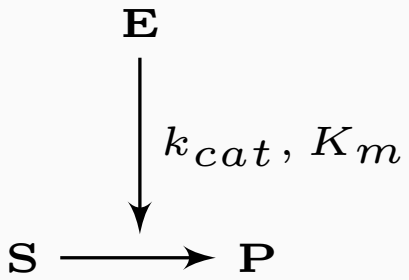
An enzymatic reaction:



Can be divided in two first order reactions plus a second order reaction. However, with the appropriate assumptions, it is possible to use a Michaelis-Menten simplification of this reaction.

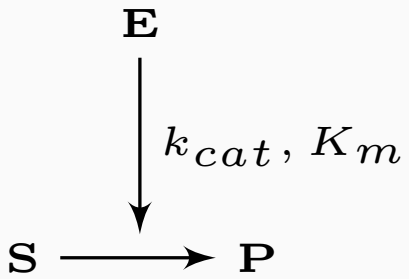
Michaelis-Menten Kinetics

We denote Michaelis-Menten simplification of the last enzymatic reaction as



Michaelis-Menten Kinetics

We denote Michaelis-Menten simplification of the last enzymatic reaction as

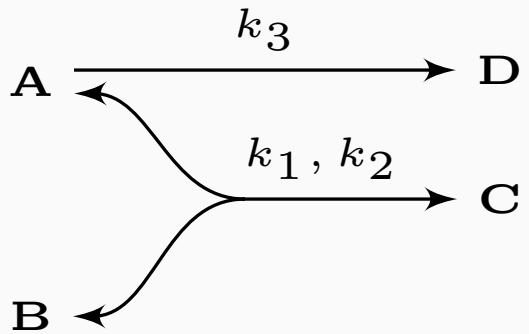


and it has rate of:

$$k_{cat} \frac{[\mathrm{E}][\mathrm{S}]}{K_M + [\mathrm{S}]}.$$

Kinetics of a System of Reactions

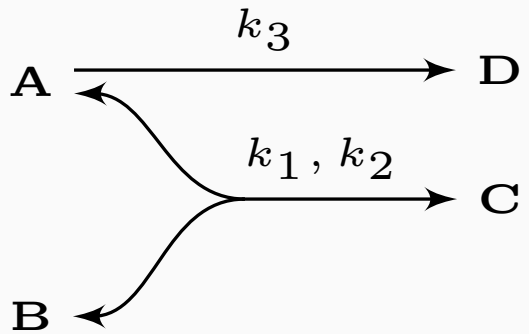
Suppose we want to model the kinetics of A on these reactions:



This system can be divided in three reactions:

Kinetics of a System of Reactions

Suppose we want to model the kinetics of A on these reactions:

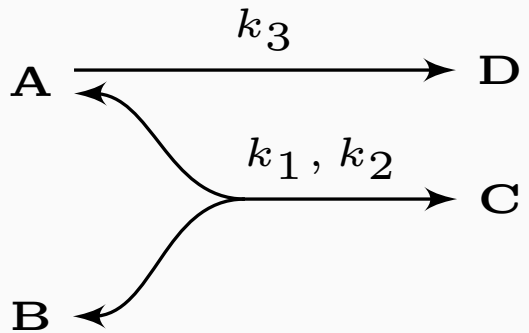


This system can be divided in three reactions:

- $A + B \longrightarrow C$, with rate $k_1[A][B]$,

Kinetics of a System of Reactions

Suppose we want to model the kinetics of A on these reactions:

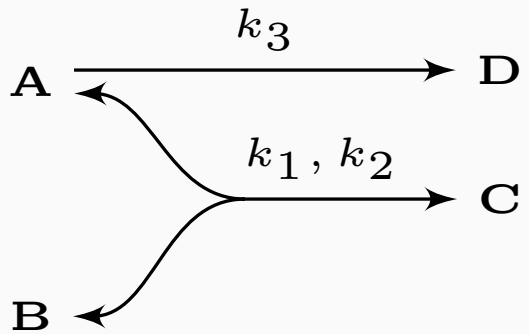


This system can be divided in three reactions:

- $A + B \longrightarrow C$, with rate $k_1[A][B]$,
- $C \longrightarrow A + B$, with rate $k_2[C]$,

Kinetics of a System of Reactions

Suppose we want to model the kinetics of A on these reactions:



This system can be divided in three reactions:

- $A + B \longrightarrow C$, with rate $k_1[A][B]$,
- $C \longrightarrow A + B$, with rate $k_2[C]$,
- $A \longrightarrow D$, with rate $k_3[A]$.

Kinetics of a System of Reactions

In $A + B \longrightarrow C$, with rate $k_1[A][B]$, A is a reactant.

Kinetics of a System of Reactions

In $A + B \longrightarrow C$, with rate $k_1[A][B]$, A is a reactant.

In $C \longrightarrow A + B$, with rate $k_2[C]$, A is a product.

Kinetics of a System of Reactions

In $A + B \longrightarrow C$, with rate $k_1[A][B]$, A is a reactant.

In $C \longrightarrow A + B$, with rate $k_2[C]$, A is a product.

In $A \longrightarrow D$, with rate $k_3[A]$, A is a reactant.

Kinetics of a System of Reactions

In $A + B \longrightarrow C$, with rate $k_1[A][B]$, A is a reactant.

In $C \longrightarrow A + B$, with rate $k_2[C]$, A is a product.

In $A \longrightarrow D$, with rate $k_3[A]$, A is a reactant.

Then, the differential equation that models the concentration change of A is:

Kinetics of a System of Reactions

In $A + B \longrightarrow C$, with rate $k_1[A][B]$, A is a reactant.

In $C \longrightarrow A + B$, with rate $k_2[C]$, A is a product.

In $A \longrightarrow D$, with rate $k_3[A]$, A is a reactant.

Then, the differential equation that models the concentration change of A is:

$$\frac{d[A]}{dt} = -k_1[A][B] + k_2[C] - k_3[D].$$

Bayesian Methods for Biochemical Model Selection

State of the Art Methods for Model Selection

There are two main Bayesian methods available for biochemical model selection:

State of the Art Methods for Model Selection

There are two main Bayesian methods available for biochemical model selection:

- Approximate Bayesian Computation;

State of the Art Methods for Model Selection

There are two main Bayesian methods available for biochemical model selection:

- Approximate Bayesian Computation;
- Marginal likelihood estimation through Thermodynamic Integration.

State of the Art Methods for Model Selection

There are two main Bayesian methods available for biochemical model selection:

- Approximate Bayesian Computation;
- Marginal likelihood estimation through Thermodynamic Integration.

For both methods, we resort to Metropolis-Hastings algorithm to generate samples of distributions.

Metropolis-Hastings algorithm

With Metropolis-Hastings, we can generate a sample of a distribution $p(\lambda)$ doing the following:

Metropolis-Hastings algorithm

With Metropolis-Hastings, we can generate a sample of a distribution $p(\lambda)$ doing the following:

Metropolis-Hastings algorithm

With Metropolis-Hastings, we can generate a sample of a distribution $p(\lambda)$ doing the following:

1. Choose some λ_0 for which $p(\lambda_0) > 0$, and set $t = 1$;

Metropolis-Hastings algorithm

With Metropolis-Hastings, we can generate a sample of a distribution $p(\lambda)$ doing the following:

1. Choose some λ_0 for which $p(\lambda_0) > 0$, and set $t = 1$;
2. Sample a candidate point λ^* from a jump distribution, $J(\lambda|\lambda_{t-1})$;

Metropolis-Hastings algorithm

With Metropolis-Hastings, we can generate a sample of a distribution $p(\lambda)$ doing the following:

1. Choose some λ_0 for which $p(\lambda_0) > 0$, and set $t = 1$;
2. Sample a candidate point λ^* from a jump distribution, $J(\lambda|\lambda_{t-1})$;
3. Calculate the ratio $r = \frac{p(\lambda^*)J_t(\lambda^{t-1}|\lambda^*)}{p(\lambda^{t-1})J_t(\lambda^*|\lambda^{t-1})}$;

Metropolis-Hastings algorithm

With Metropolis-Hastings, we can generate a sample of a distribution $p(\lambda)$ doing the following:

1. Choose some λ_0 for which $p(\lambda_0) > 0$, and set $t = 1$;
2. Sample a candidate point λ^* from a jump distribution, $J(\lambda|\lambda_{t-1})$;
3. Calculate the ratio $r = \frac{p(\lambda^*)J_t(\lambda^{t-1}|\lambda^*)}{p(\lambda^{t-1})J_t(\lambda^*|\lambda^{t-1})}$;
4. Set $\lambda_t = \lambda^*$ with probability $\min(1, r)$ and $\lambda_t = \lambda_{t-1}$ otherwise;

Metropolis-Hastings algorithm

With Metropolis-Hastings, we can generate a sample of a distribution $p(\lambda)$ doing the following:

1. Choose some λ_0 for which $p(\lambda_0) > 0$, and set $t = 1$;
2. Sample a candidate point λ^* from a jump distribution, $J(\lambda|\lambda_{t-1})$;
3. Calculate the ratio $r = \frac{p(\lambda^*)J_t(\lambda^{t-1}|\lambda^*)}{p(\lambda^{t-1})J_t(\lambda^*|\lambda^{t-1})}$;
4. Set $\lambda_t = \lambda^*$ with probability $\min(1, r)$ and $\lambda_t = \lambda_{t-1}$ otherwise;
5. Increase t by one and repeat from Step 2 if not reached iteration limit.

Model Selection

Ranking with Marginal Likelihood Estimation

Likelihood of Data Given Model and Parameters

If we consider that a model M with parameters θ correctly represent the signaling pathway

Likelihood of Data Given Model and Parameters

If we consider that a model M with parameters θ correctly represent the signaling pathway and that there is a Gaussian observation error on D.

Likelihood of Data Given Model and Parameters

If we consider that a model M with parameters θ correctly represent the signaling pathway and that there is a Gaussian observation error on D. Then, the likelihood of observing experimental data D is:

$$p(D|M, \theta) =$$

Likelihood of Data Given Model and Parameters

If we consider that a model M with parameters θ correctly represent the signaling pathway and that there is a Gaussian observation error on D. Then, the likelihood of observing experimental data D is:

$$p(D|M,\theta) = p_{\mathcal{N}_{(\vec{0},\Sigma)}}(\phi(M,\theta) - D).$$

Where $\phi(M,\theta)$ is the simulated observation.

Marginal Likelihood of Data

We can marginalize the likelihood to obtain:

$$p(D|M) = \int_{\Theta} p(D|M, \theta) p(\theta|M) d\theta$$

Marginal Likelihood of Data

We can marginalize the likelihood to obtain:

$$p(D|M) = \int_{\Theta} p(D|M,\theta)p(\theta|M)d\theta$$

Calculating this integral is hard, therefore we resort to estimating another integral.

Power-posterior distributions

We define a power-posterior distribution as:

$$p_\beta(\theta) = \frac{p(D|\theta, M)^\beta p(\theta|M)}{\int_\Theta p(D|\theta, M)^\beta p(\theta|M) d\theta},$$

Power-posterior distributions

We define a power-posterior distribution as:

$$p_\beta(\theta) = \frac{p(D|\theta, M)^\beta p(\theta|M)}{\int_\Theta p(D|\theta, M)^\beta p(\theta|M) d\theta},$$

Note that:

$$p_{\beta=0}(\theta) = p(\theta|M),$$

Power-posterior distributions

We define a power-posterior distribution as:

$$p_\beta(\theta) = \frac{p(D|\theta, M)^\beta p(\theta|M)}{\int_\Theta p(D|\theta, M)^\beta p(\theta|M) d\theta},$$

Note that:

$$p_{\beta=0}(\theta) = p(\theta|M),$$

and that

$$p_{\beta=1}(\theta) = \frac{p(D, \theta|M)}{\int_\Theta p(D, \theta|M) d\theta} = \frac{p(\theta|D, M)p(D|M)}{p(D|M)} = p(\theta|D, M).$$

The Thermodynamic Integral

Using power-posteriors distributions, it is possible to show that

The Thermodynamic Integral

Using power-posteriors distributions, it is possible to show that

$$\ln p(D|M) = \int_0^1 \mathbb{E}_{p_\beta(\theta)}[\ln p(D|\theta, M)] d\beta.$$

Estimating the Thermodynamic Integral

It is possible to estimate the Thermodynamic Integral using the trapezoidal rule.

Estimating the Thermodynamic Integral

It is possible to estimate the Thermodynamic Integral using the trapezoidal rule. Setting $0 = \beta_0 < \beta_1 < \ldots < \beta_T = 1$, the marginal likelihood is approximately equal to:

Estimating the Thermodynamic Integral

It is possible to estimate the Thermodynamic Integral using the trapezoidal rule. Setting $0 = \beta_0 < \beta_1 < \ldots < \beta_T = 1$, the marginal likelihood is approximately equal to:

$$\sum_{t=0}^{T-1} (\beta_{t+1} - \beta_t) \frac{\mathbb{E}_{p_{\beta_{t+1}}(\theta)}[\log p(D|M,\theta)] + \mathbb{E}_{p_{\beta_t}(\theta)}[\log p(D|M,\theta)]}{2}$$

Estimating the Thermodynamic Integral

To produce the estimates of

$$\mathbb{E}_{p_{\beta_t}(\theta)}[\log p(D|M,\theta)] \text{ for } t \in \{0,\ldots,T\}$$

we need to produce samples of the power-posteriors $p_{\beta_t}(\theta)$.

Sampling from Power-posteriors

The sampling of the power-posteriors are generated using Metropolis-Hastings algorithms in three steps.

Sampling from Power-posteriors

The sampling of the power-posteriors are generated using Metropolis-Hastings algorithms in three steps. In all of the steps, the proposal distribution used is multivariate lognormal.

Sampling from Power-posteriors

On the first step, called naive burn-in the jump distribution has a diagonal covariance matrix.

Sampling from Power-posteriors

On the first step, called naive burn-in the jump distribution has a diagonal covariance matrix. This matrix is updated according to the rate of acceptance of parameters.

Sampling from Power-posteriors

On the first step, called naive burn-in the jump distribution has a diagonal covariance matrix. This matrix is updated according to the rate of acceptance of parameters.

- if the acceptance rate is high, then increase the variance of the jump;

Sampling from Power-posteriors

On the first step, called naive burn-in the jump distribution has a diagonal covariance matrix. This matrix is updated according to the rate of acceptance of parameters.

- if the acceptance rate is high, then increase the variance of the jump;
- if the acceptance rate is low, then decrease the variance of the jump.

Sampling from Power-posteriors

On the first step, called naive burn-in the jump distribution has a diagonal covariance matrix. This matrix is updated according to the rate of acceptance of parameters.

- if the acceptance rate is high, then increase the variance of the jump;
- if the acceptance rate is low, then decrease the variance of the jump.

Sampling from the Power-posteriors

On the second sampling step, called posterior shaped burn-in, we use the covariance of the current sample times some constant as the covariance of the jump distribution.

Sampling from the Power-posteriors

On the third step, we perform the Populational Monte Carlo Markov Chain sampling. This algorithm allows us to mix samples from different power posteriors.

Ranking with Approximate Bayesian Computation

Approximate Bayesian Computation

Approximate Bayesian Computation (ABC) is a method that allows one to obtain samples of a distribution close to $p(\theta, M|D)$.

Approximate Bayesian Computation

Approximate Bayesian Computation (ABC) is a method that allows one to obtain samples of a distribution close to $p(\theta, M|D)$. A general ABC implementation works as follow:

Approximate Bayesian Computation

Approximate Bayesian Computation (ABC) is a method that allows one to obtain samples of a distribution close to $p(\theta, M|D)$. A general ABC implementation works as follow:

1. Sample a parameter candidate (θ^*, M^*) from some proposal distribution.

Approximate Bayesian Computation

Approximate Bayesian Computation (ABC) is a method that allows one to obtain samples of a distribution close to $p(\theta, M|D)$. A general ABC implementation works as follow:

1. Sample a parameter candidate (θ^*, M^*) from some proposal distribution.
2. Generate simulations $\phi(M^*, \theta^*) = D^*$.

Approximate Bayesian Computation

Approximate Bayesian Computation (ABC) is a method that allows one to obtain samples of a distribution close to $p(\theta, M|D)$. A general ABC implementation works as follow:

1. Sample a parameter candidate (θ^*, M^*) from some proposal distribution.
2. Generate simulations $\phi(M^*, \theta^*) = D^*$.
3. Calculate $d(D^*, D)$. If $d(D^*, D) < \epsilon$ for some previously specified ϵ, then add (θ^*, M^*) to the sample.

Approximate Bayesian Computation

Approximate Bayesian Computation (ABC) is a method that allows one to obtain samples of a distribution close to $p(\theta, M|D)$. A general ABC implementation works as follow:

1. Sample a parameter candidate (θ^*, M^*) from some proposal distribution.
2. Generate simulations $\phi(M^*, \theta^*) = D^*$.
3. Calculate $d(D^*, D)$. If $d(D^*, D) < \epsilon$ for some previously specified ϵ, then add (θ^*, M^*) to the sample.
4. Repeat until some iteration limit.

Approximate Bayesian Computation

The result of the algorithm is a sample of the distribution

$$p(\theta, M | d(\phi(M, \theta), D) < \epsilon).$$

ABC Sequential Monte Carlo

ABC Sequential Monte Carlo (`ABC-SMC`) improves a simple ABC algorithm by using a sequence $\epsilon_0 > \ldots > \epsilon_T$ acceptance tolerances.

ABC Sequential Monte Carlo

ABC Sequential Monte Carlo (`ABC-SMC`) improves a simple ABC algorithm by using a sequence $\epsilon_0 > \ldots > \epsilon_T$ acceptance tolerances. The sample for a tolerance ϵ_i is used to generate candidates for sample of tolerance ϵ_{i+1}.

ABC Sequential Monte Carlo

ABC Sequential Monte Carlo (`ABC-SMC`) improves a simple ABC algorithm by using a sequence $\epsilon_0 > \ldots > \epsilon_T$ acceptance tolerances. The sample for a tolerance ϵ_i is used to generate candidates for sample of tolerance ϵ_{i+1}.

We can use the accepted parameters of tolerance ϵ and model M to estimate

$$p(M|d(\phi(M,\theta)) < \epsilon).$$

Development of SigNetMS

The SigNetMS Software

A software that estimates the marginal likelihood

To choose a cost function, we needed to compare the ABC-SMC and Marginal Likelihood approaches for model selection.

A software that estimates the marginal likelihood

To choose a cost function, we needed to compare the ABC-SMC and Marginal Likelihood approaches for model selection.

- `ABC-SysBio` is a software that implements `ABC-SMC`
- `BioBayes` is a software that implements the estimation of the marginal likelihood.

A software that estimates the marginal likelihood

However, the usage of `BioBayes` in our context was cumbersome.

A software that estimates the marginal likelihood

However, the usage of `BioBayes` in our context was cumbersome.
Therefore, we decided to implement SigNetMS.

The SigNetMS software

SigNetMS is a Python package and command line software that estimates the marginal likelihood of a model given experimental data.

The input expected by SigNetMS

The input to SigNetMS includes:

- An SBML file model;
- An XML file with prior distributions of parameters;
- An XML file with experimental data;

The output produced by SigNetMS

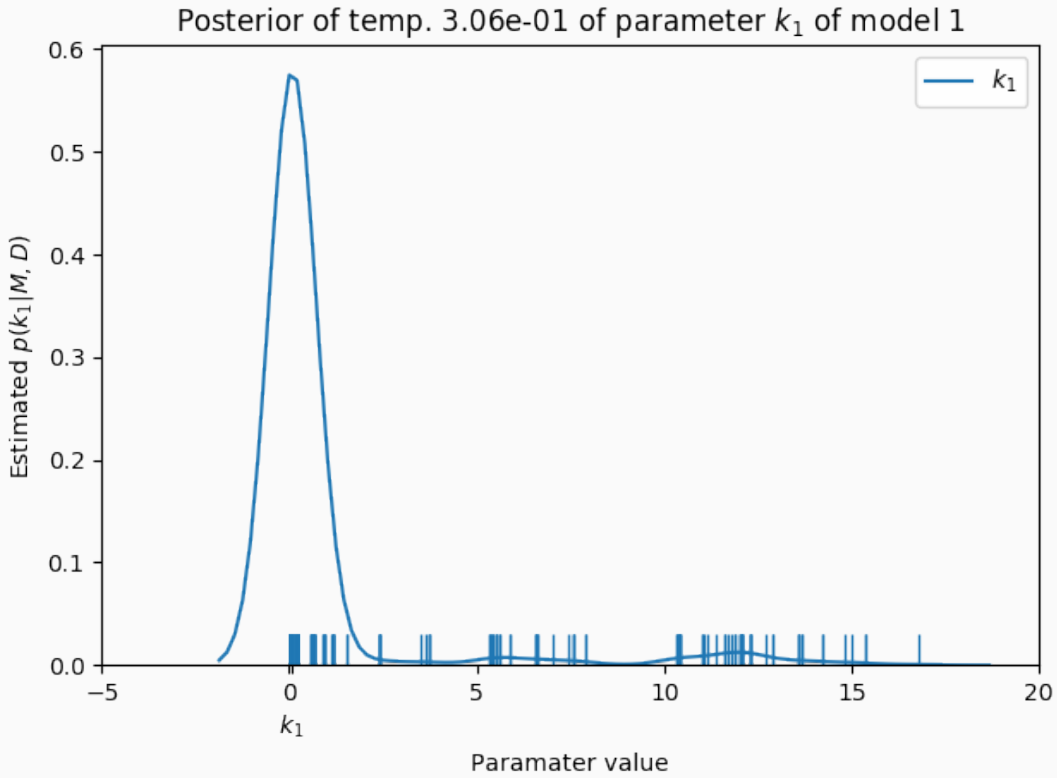
SigNetMS produces an estimate of the marginal likelihood, as you may have suspected.

The output produced by SigNetMS

SigNetMS produces an estimate of the marginal likelihood, as you may have suspected. Moreover, the software also produces samples of each power posterior of parameters, which is in turn a sample of the posterior distribution of parameters.

The output produced by SigNetMS

SigNetMS produces an estimate of the marginal likelihood, as you may have suspected. Moreover, the software also produces samples of each power posterior of parameters, which is in turn a sample of the posterior distribution of parameters.



Fast integration and parameter sampling

Problems with efficiency

To generate a sample to each power posterior, we need to iterate the Monte Carlo Markov Chain algorithm tens of thousands of times.

Problems with efficiency

To generate a sample to each power posterior, we need to iterate the Monte Carlo Markov Chain algorithm tens of thousands of times.

For each step we need to evaluate the likelihood function, and numerically integrate the system.

Problems with efficiency

To generate a sample to each power posterior, we need to iterate the Monte Carlo Markov Chain algorithm tens of thousands of times.

For each step we need to evaluate the likelihood function, and numerically integrate the system. That makes sampling the most time consuming procedure of SigNetMS.

Our first implementation was not very efficient

The first implementation of SigNetMS did not cope with larger instances of model selection.

Our first implementation was not very efficient

The first implementation of SigNetMS did not cope with larger instances of model selection. We tackled this problem in two ways:

- change the representation of the system of ordinary differential equations;
- implement parallelization.

Changing the representation of the system of ordinary differential equations

In the first implementation of SigNetMS, systems of EDOs were represented as an array of strings.

Changing the representation of the system of ordinary differential equations

In the first implementation of SigNetMS, systems of EDOs were represented as an array of strings.

We used SymPy to provide automatically generated code that allowed us to create a C function to represent the system of ODEs.

Comparing the representation of the system of ordinary differential equations

Number of Integrations	Average time (seconds) to perform a sequence of integrations	
	String Evaluation	`sympy.autowrap`
10	2.98	0.9
100	35.3	6.6
200	72.1	13.1
400	139.1	26.9

Parallelizing the sampling of multiple power posteriors

The first two phases of the sampling procedure occurs independently between different power posteriors.

Parallelizing the sampling of multiple power posteriors

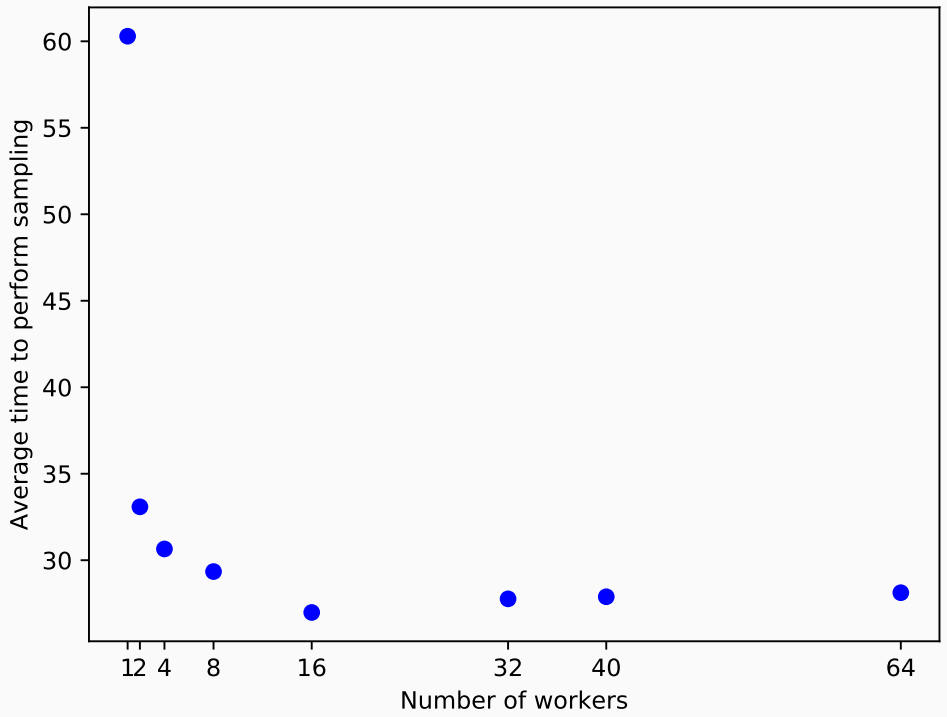
The first two phases of the sampling procedure occurs independently between different power posteriors.

We used the map pattern to parallelize the sampling of different power posterior distributions.

Parallelizing the sampling of multiple power posteriors

The first two phases of the sampling procedure occurs independently between different power posteriors.

We used the map pattern to parallelize the sampling of different power posterior distributions.



Experiments and Results

Experiments and Results

We prepared two experiments in this work:

Experiments and Results

We prepared two experiments in this work:

- Comparison between ABC-SysBio and SigNetMS
- Solving model selection as a feature selection instance

Comparison between ABC-SySBio and SigNetMS

Chossing a software for model selection

ABC-SysBio and SigNetMS use different Bayesian approaches for model selection.

Chossing a software for model selection

ABC-SysBio and SigNetMS use different Bayesian approaches for model selection. The first creates an estimate of $p(M|D)$,

Chossing a software for model selection

ABC-SysBio and SigNetMS use different Bayesian approaches for model selection. The first creates an estimate of $p(M|D)$, and the second creates an estimate of $p(D|M)$ (the marginal likelihood).

Experiment description

To compare both software we ran two experiments based on the same procedure:

Experiment description

To compare both software we ran two experiments based on the same procedure:

- Create 4 candidate models.

Experiment description

To compare both software we ran two experiments based on the same procedure:

- Create 4 candidate models.
- For one of the models, choose a set of parameter values and time steps and simulate data.

Experiment description

To compare both software we ran two experiments based on the same procedure:

- Create 4 candidate models.
- For one of the models, choose a set of parameter values and time steps and simulate data.
- Add Gaussian noise to the simulations. Repeat two more times to generate three observations of the system.

Experiment description

To compare both software we ran two experiments based on the same procedure:

- Create 4 candidate models.
- For one of the models, choose a set of parameter values and time steps and simulate data.
- Add Gaussian noise to the simulations. Repeat two more times to generate three observations of the system.
- Neglect chosen parameter values and define prior distributions for every parameter.

Experiment description

To compare both software we ran two experiments based on the same procedure:

- Create 4 candidate models.
- For one of the models, choose a set of parameter values and time steps and simulate data.
- Add Gaussian noise to the simulations. Repeat two more times to generate three observations of the system.
- Neglect chosen parameter values and define prior distributions for every parameter.
- Rank the four models.

Experiment description

To compare both software we ran two experiments based on the same procedure:

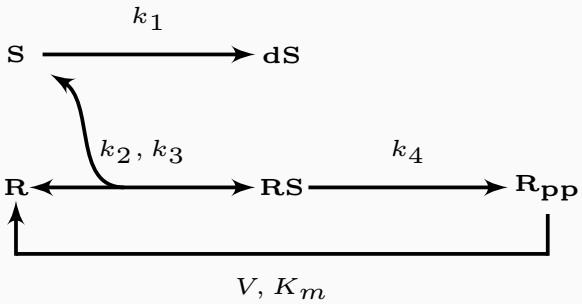
- Create 4 candidate models.
- For one of the models, choose a set of parameter values and time steps and simulate data.
- Add Gaussian noise to the simulations. Repeat two more times to generate three observations of the system.
- Neglect chosen parameter values and define prior distributions for every parameter.
- Rank the four models.

In this presentation we show results of the first experiment only.

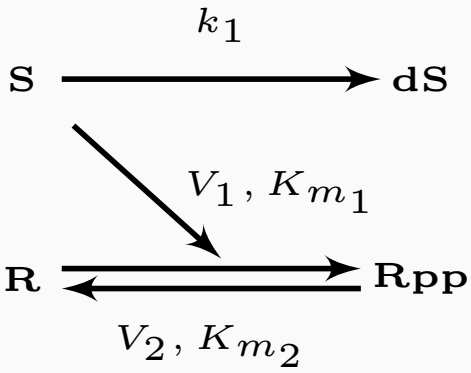
The instance

This instance is originally from Vyshemirsky and Girolami (2007), in which they present results of Annealing Melting Integration.

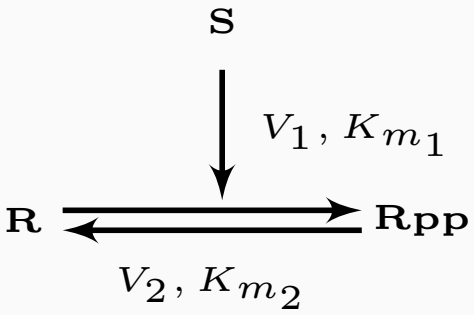
The instance



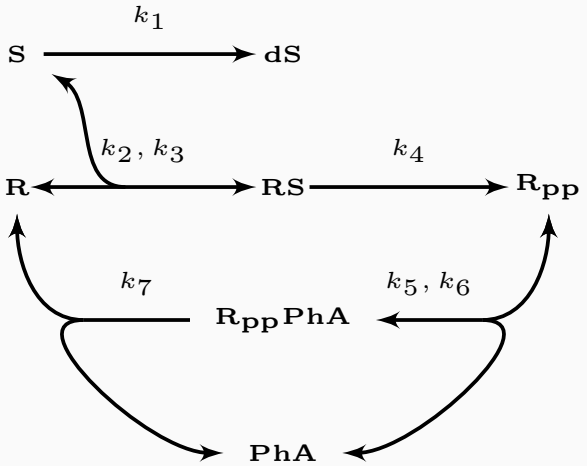
The "correct" model



The simplification model

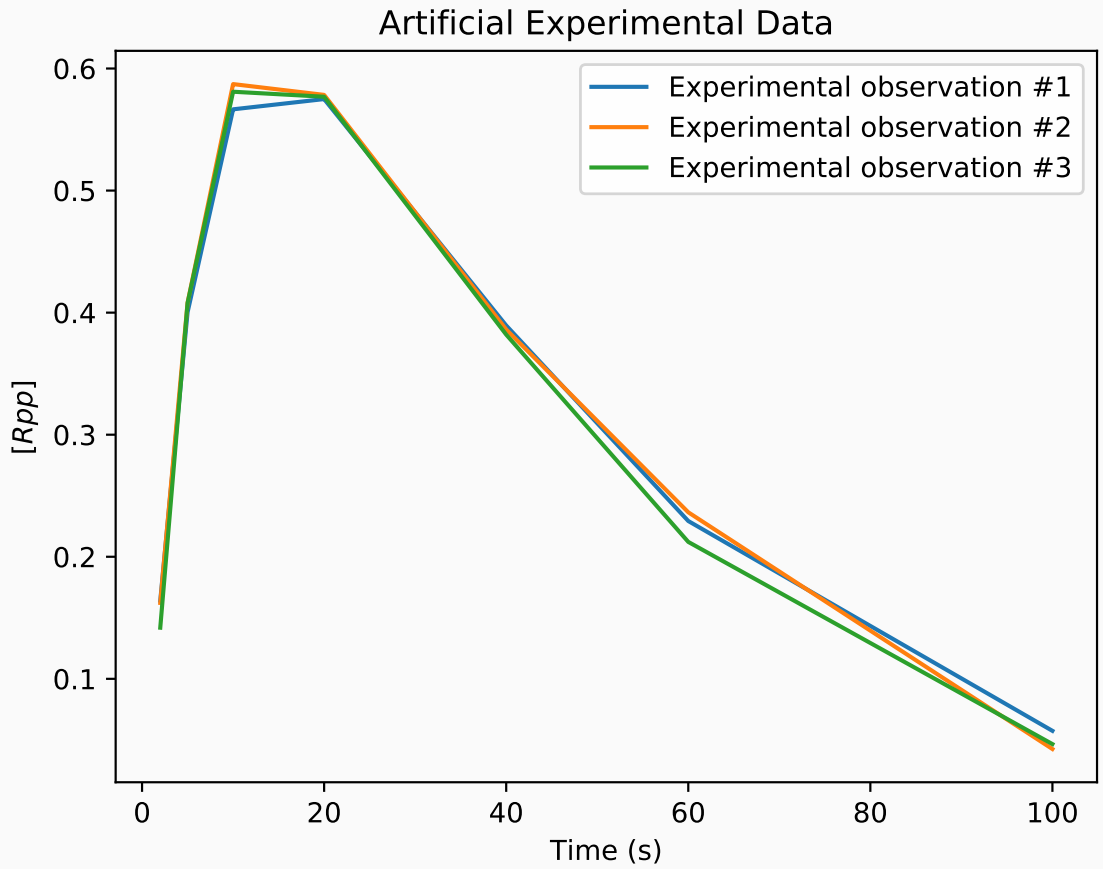


The incorrect model



The generalization model

The instance



Results on ABC-SysBio

The ABC-SysBio software returned the following ranking of models:

Results on ABC-SysBio

The ABC-SysBio software returned the following ranking of models:

1. incorrect model

Results on ABC-SysBio

The ABC-SysBio software returned the following ranking of models:

1. incorrect model
2. simplification model

Results on ABC-SysBio

The ABC-SysBio software returned the following ranking of models:

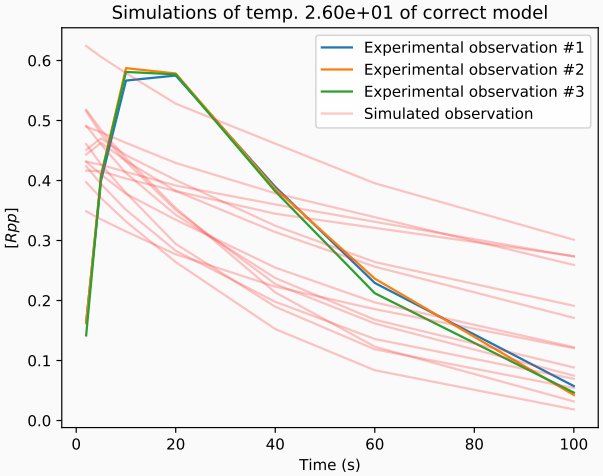
1. incorrect model
2. simplification model
3. generalization model

Results on ABC-SysBio

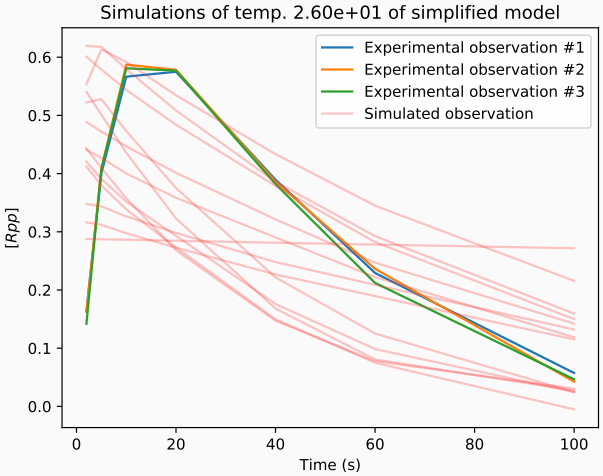
The ABC-SysBio software returned the following ranking of models:

1. incorrect model
2. simplification model
3. generalization model
4. correct model

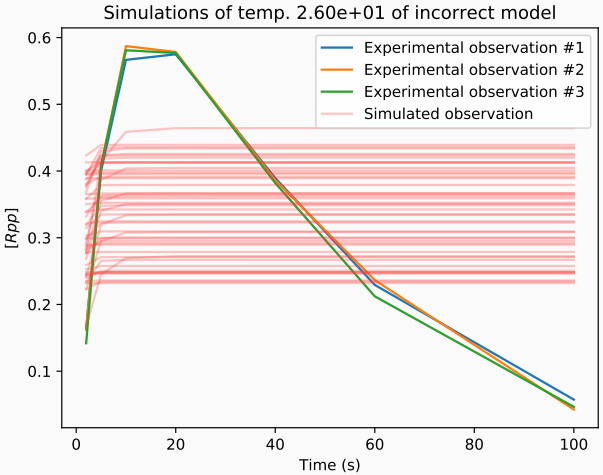
Results on ABC-SysBio



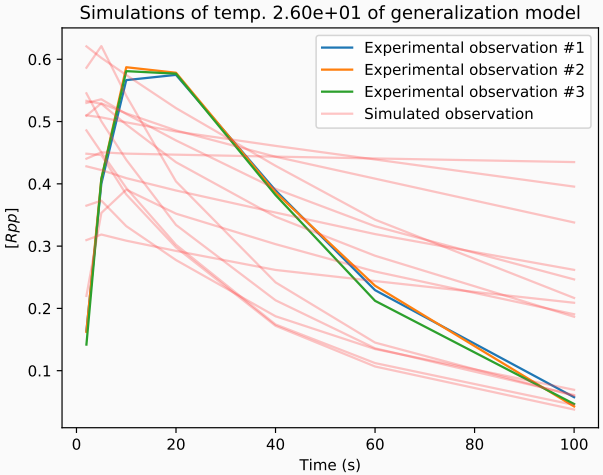
correct model



simplified model



incorrect model



generalization model

Results on SigNetMS

The ranking returned by SigNetMS on the first experiment is:

1. correct model

Results on SigNetMS

The ranking returned by SigNetMS on the first experiment is:

1. correct model
2. simplification model

Results on SigNetMS

The ranking returned by SigNetMS on the first experiment is:

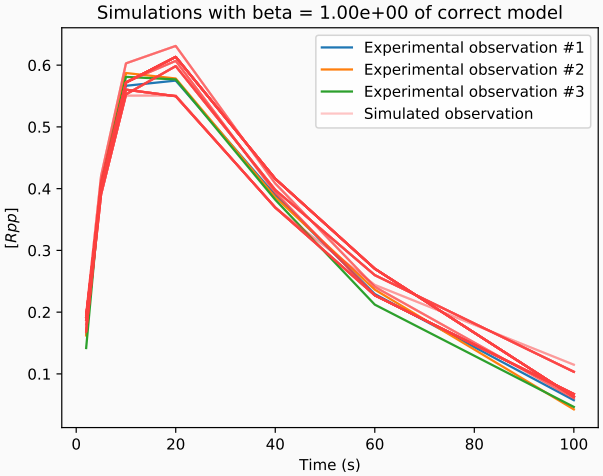
1. correct model
2. simplification model
3. generalization model

Results on SigNetMS

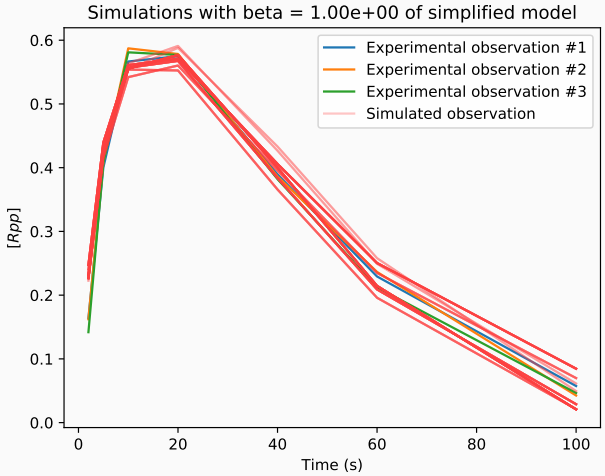
The ranking returned by SigNetMS on the first experiment is:

1. correct model
2. simplification model
3. generalization model
4. incorrect model

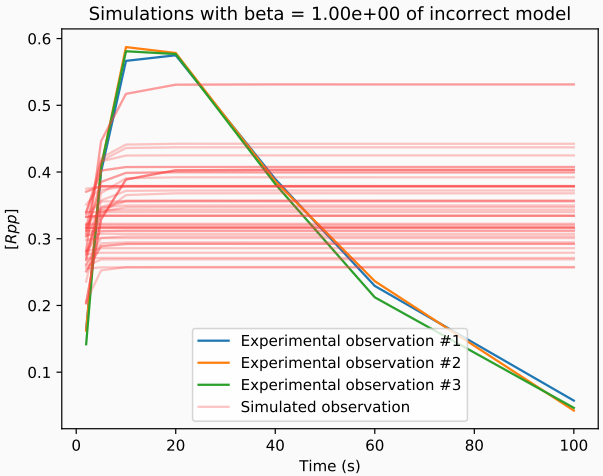
Results on SigNetMS



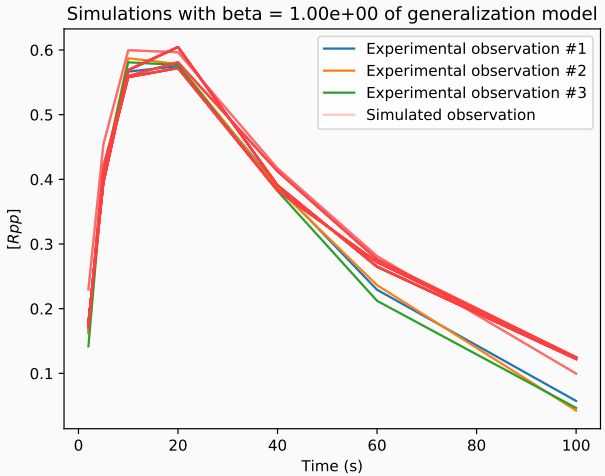
correct model



simplified model



incorrect model



generalization model

Simulations generated by the correct model

Model selection as a feature selection problem

Model selection as a feature selection problem

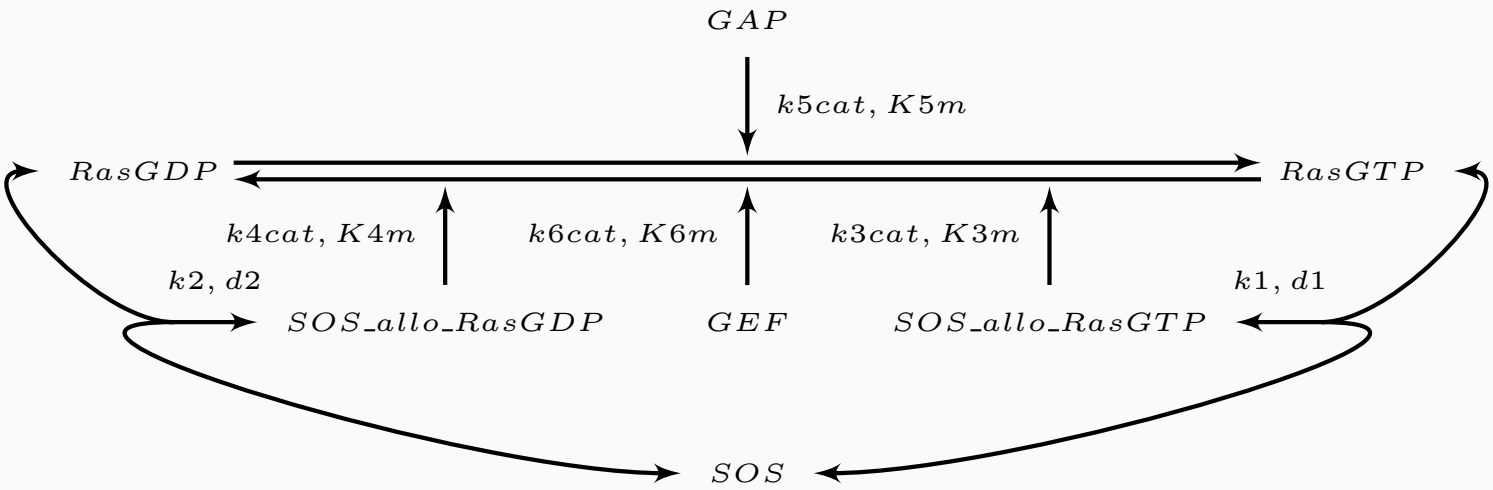
After defining that SigNetMS is our software choice for a cost function, we are able to experiment the approach of solving a model selection instance as a feature selection problem.

The model selection instance

We proposed a Ras switch pathway to experiment on.

The model selection instance

We proposed a Ras switch pathway to experiment on.

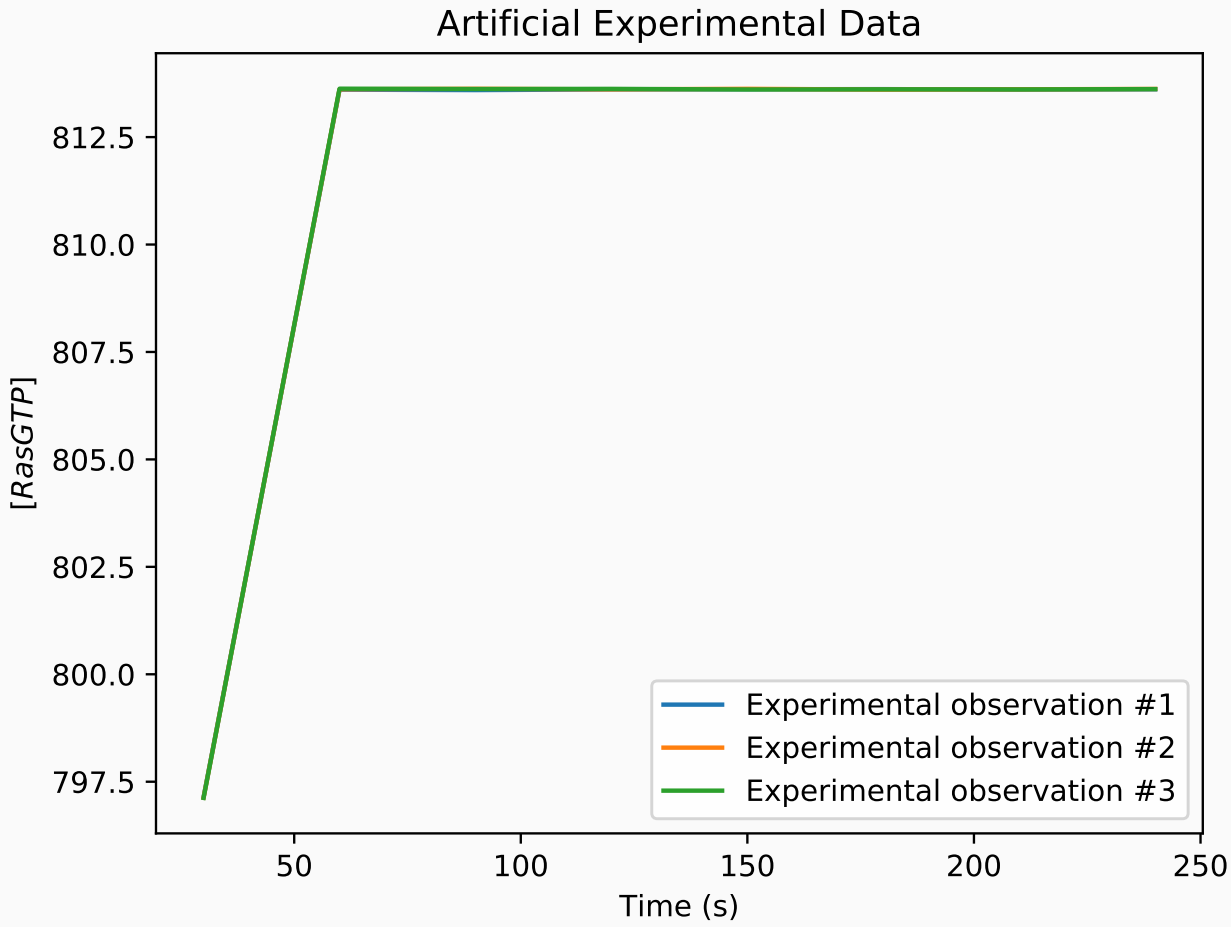


The model selection instance

The concentration of activated Ras was measured at the time steps of 30, 60, 90, 120, 150, 180, 210, and 240 seconds.

The model selection instance

The concentration of activated Ras was measured at the time steps of 30, 60, 90, 120, 150, 180, 210, and 240 seconds.



Model selection as a feature selection problem

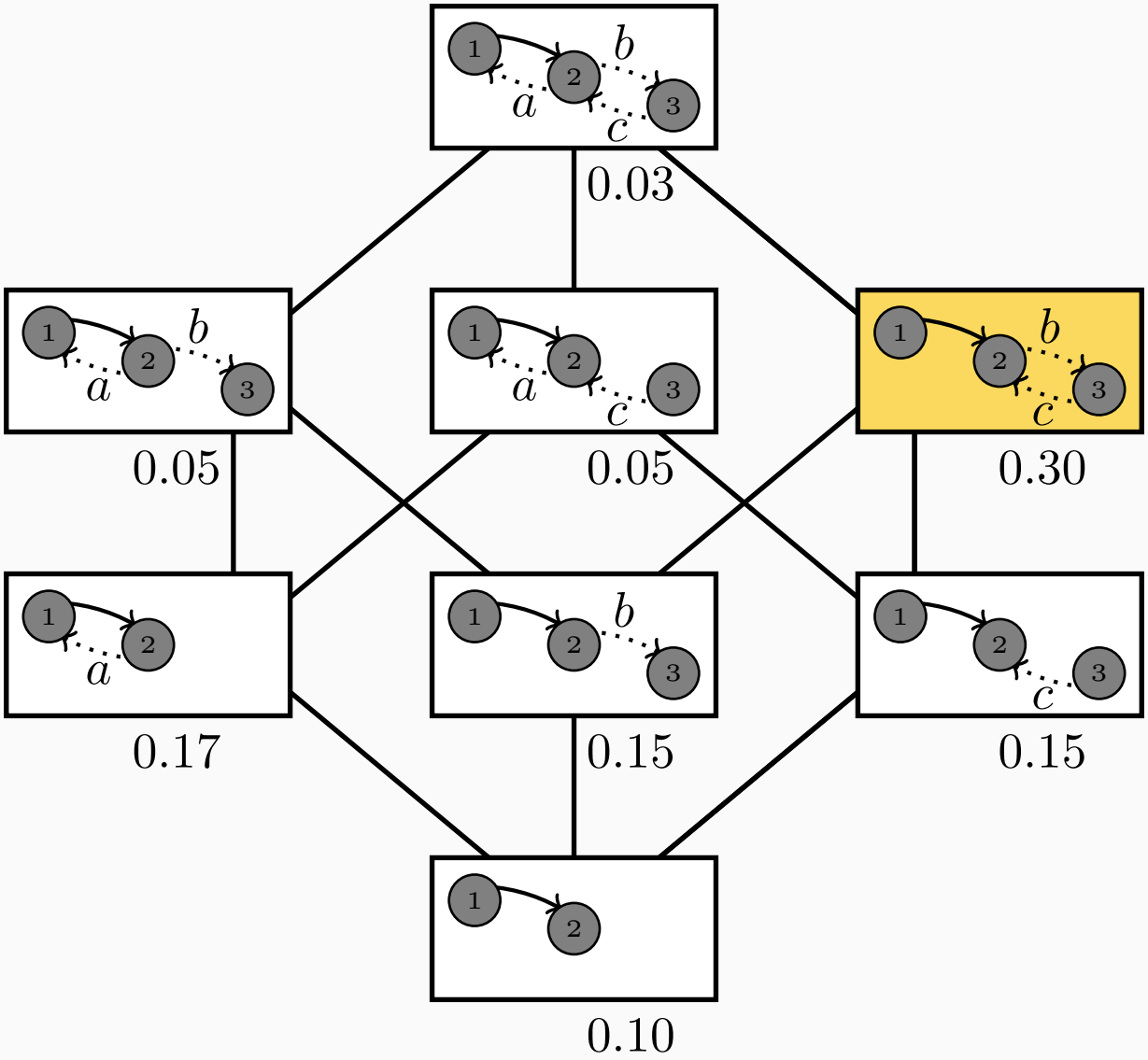
The feature selection problem consists in finding the best subset of a set of features, S, given a cost function c.

Model selection as a feature selection problem

The feature selection problem consists in finding the best subset of a set of features, S, given a cost function c.

If we define the set of feature as a set of reactions, we can create a feature selection instance that represents a model selection instance.

Model selection as a feature selection problem



The set of features of our experiment

In the instance we prepared, the base model has zero reactions,

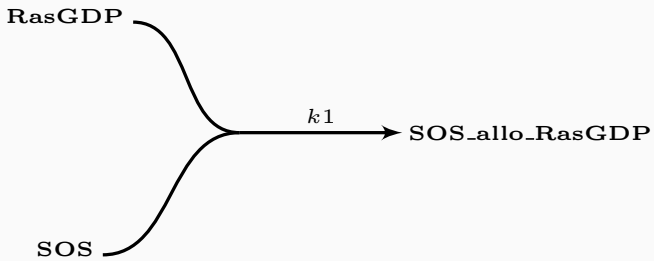
The set of features of our experiment

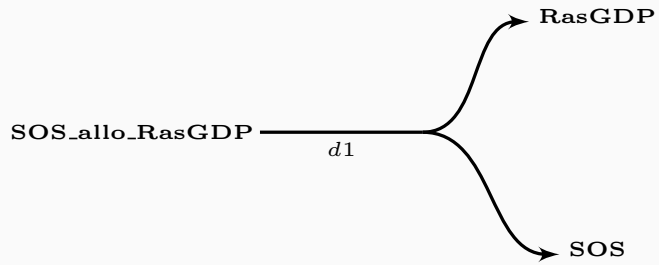
In the instance we prepared, the base model has zero reactions, and the set of features S is composed by 10 reactions,

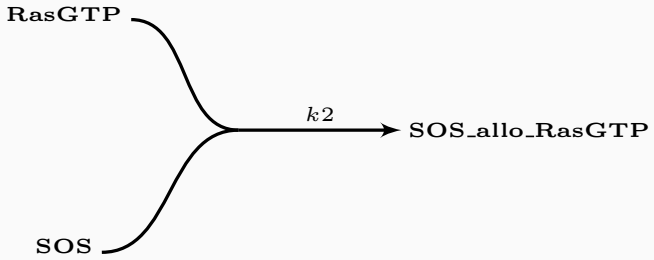
The set of features of our experiment

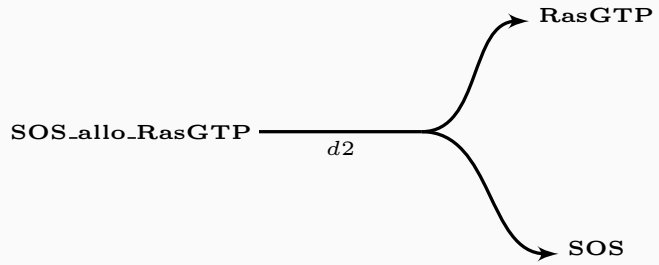
In the instance we prepared, the base model has zero reactions, and the set of features S is composed by 10 reactions, 8 of them present on the correct model.

The set of features of our experiment

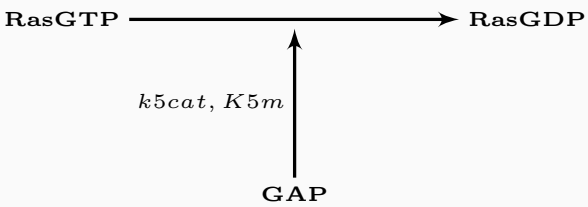


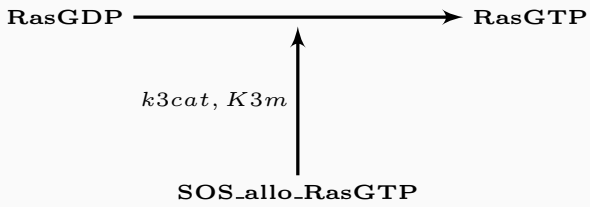


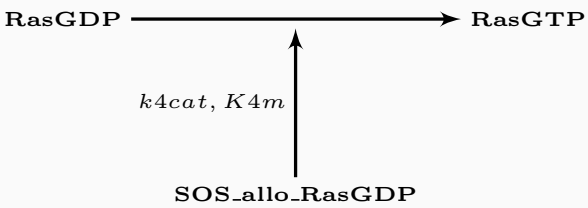


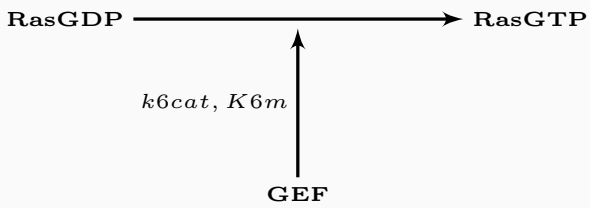


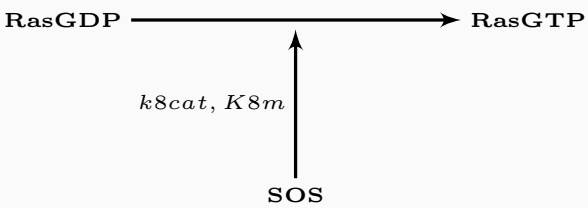
The set of features of our experiment

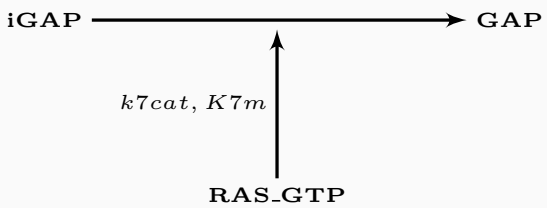












Finding a solution

The search space $\mathcal{P}(S)$, has 2^{10}. Therefore, a heuristic is necessary to traverse the space.

Finding a solution

The search space $\mathcal{P}(S)$, has 2^{10}. Therefore, a heuristic is necessary to traverse the space. We used the Sequential Forward Selection (SFS) algorithm.

Finding a solution

In the SFS procedure, we start from the bottom of the search space.

Finding a solution

In the SFS procedure, we start from the bottom of the search space. And for every iteration, we select the best adjacent model that has one more reaction.

Results of the search

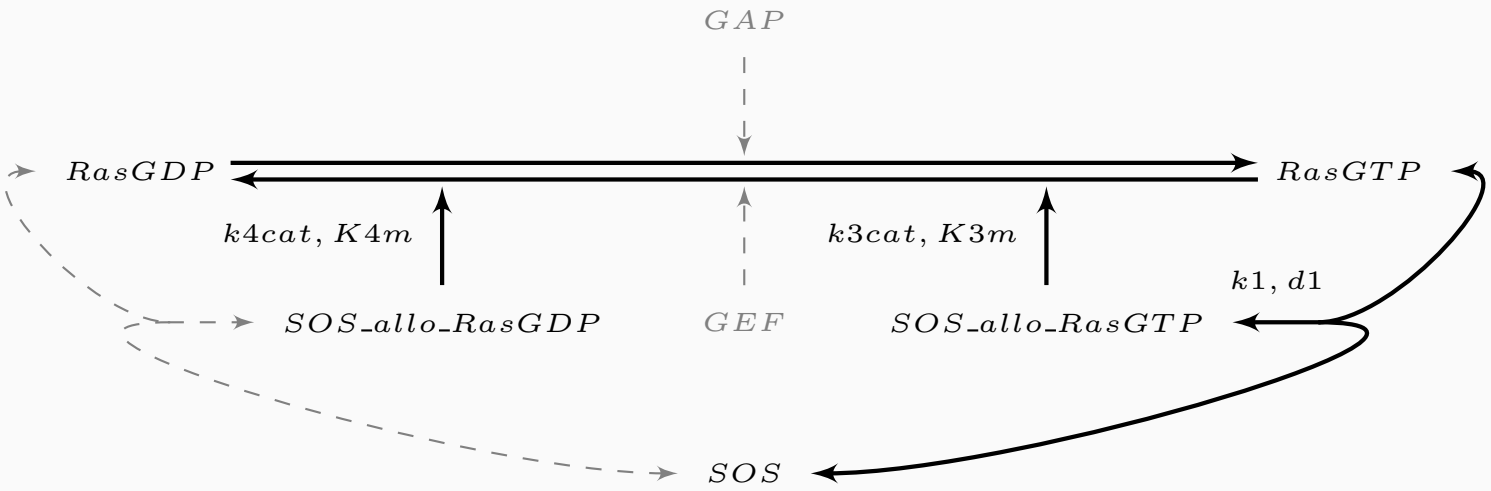
Characteristic Vector	Score	Cost function time (seconds)
0000000000	330721.05	851.3
0010000000	245681.93	1083.4
0010010000	211.62	4257.4
0011010000	-1.32	5007.71
0011011000	-4.27	4458.7
0111011000	-7.90	5035.7

Results of the search

The found model is contained in the correct model:

Results of the search

The found model is contained in the correct model:



Results of the search

Simulations generated by the found model

Results of the search

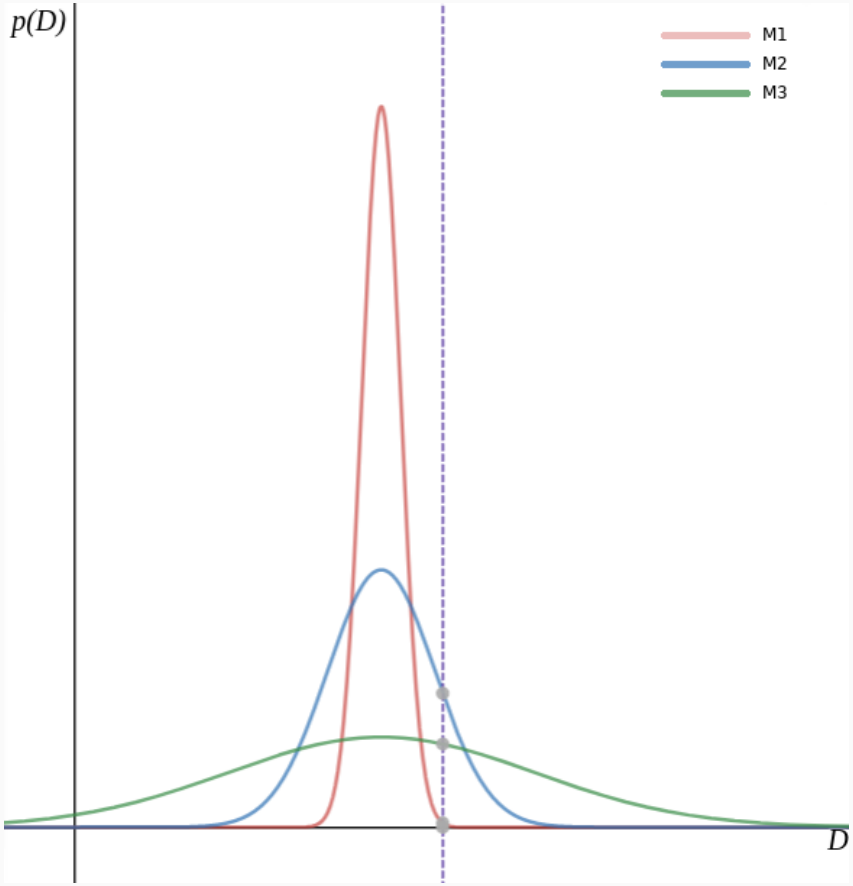
Simulations generated by the correct model

Results of the search

In this experiment, we experienced a known feature of marginal likelihood approaches:

Results of the search

In this experiment, we experienced a known feature of marginal likelihood approaches: intermediate complexity models are preferred.



Conclusions

Contributions of this work

The main contributions of this work are:

Contributions of this work

The main contributions of this work are:

- the implementation of the SigNetMS software;

Contributions of this work

The main contributions of this work are:

- the implementation of the SigNetMS software;
- the comparison between SigNetMS and ABC-SysBio;

Contributions of this work

The main contributions of this work are:

- the implementation of the SigNetMS software;
- the comparison between SigNetMS and ABC-SysBio;
- the experimentation of feature selection on model selection using a marginal likelihood approach to define the cost function.

Suggestions for future work

We also suggest a few topics for future related work:

- efficiency improvements on SigNetMS;

Suggestions for future work

We also suggest a few topics for future related work:

- efficiency improvements on SigNetMS;
- treatment of numerical instabilities on numerical integrations of SigNetMS;

Suggestions for future work

We also suggest a few topics for future related work:

- efficiency improvements on SigNetMS;
- treatment of numerical instabilities on numerical integrations of SigNetMS;
- solving the model selection problem as a U-Curve problem;

Suggestions for future work

We also suggest a few topics for future related work:

- efficiency improvements on SigNetMS;
- treatment of numerical instabilities on numerical integrations of SigNetMS;
- solving the model selection problem as a U-Curve problem;
- experimentation on heterogeneous conditions of experimental measurements;

Suggestions for future work

We also suggest a few topics for future related work:

- efficiency improvements on SigNetMS;
- treatment of numerical instabilities on numerical integrations of SigNetMS;
- solving the model selection problem as a U-Curve problem;
- experimentation on heterogeneous conditions of experimental measurements;
- application of the methodology on real instances.

Thank you!