

Project Report

Inference of signal transduction pathways

Silvio Waschina

October 7, 2009

Contents

1	Introduction	2
2	Background	3
3	Deterministic inference	3
3.1	The model	3
3.2	Parameter optimization	4
3.3	Experiment and numerical results	4
3.4	Discussion	5
4	Statistical inference	6
4.1	Methods and algorithms	6
4.1.1	Bayes' theorem	6
4.1.2	Monte-Carlo-Markov-chain methods	7
4.2	Experiments and numerical results	8
4.3	Discussion	12
5	Conclusion	12

A Example: Metropolis-Hastings algorithm	13
B Example: Gibbs sampling	13
C Model ODEs for Experiments in 4.2	14

1 Introduction

Many cellular processes like metabolisms, cell growth and cell division can be regulated from external signals which are transduced into the cell. Like compounds in metabolic systems the components (e.g. enzymes, second messenger) of signal transduction are building occasionally complex networks. It is a main interest of biochemists and medical scientists to clarify these networks and their role for the cell. One prominent example is the enzyme cascade of the MAP-Kinase, especially since it has been asserted to be linked with human cancer diseases e.g. breast cancer[1]. Data from signal transduction networks are stored in databases for instance KEGG (www.genome.jp/kegg/) or REACTOME (www.reactome.org/). Besides the structure of these networks the system biology is interested in what kind of kinetic underlies in each interaction (e.g. mass kinetics, Michaelis-Menten-kinetic).

In this report I present a method how to predict kinetic parameters on the basis of known structure of signal transduction pathways given in ordinary differential equations(ODEs) and time series data from enzymes-,proteins- or other messengers concentrations or activity.

2 Background

Many signaling pathways are known and new information is published consistently, in human as well as in model organisms. This information mainly consists of which cellular protein (or other signal transducers) interacts with which other compound (e.g. determined with Western plots) and what is the molecular mechanism of each interaction. This knowledge is usually represented in networks. These often very complex networks doesn't describe the time-dependent dynamics of the entities in the pathway. To describe the dynamics two steps have to be done first: data acquisition and mathematical modelling.

Collecting data is quite difficult due signal transductions are fast which makes observation very hard and the range of possible protein-protein interactions are broad, e.g. phosphorylation, dephosphorylation, acetylation, degradation and localization, which requires different methods of data acquisition. This kind of data is in comparison to e.g. metabolomics very rare and should be extracted often directly from the literature. However, the data may increase because of improvements of mass spectrometry coupled with liquid chromatography technology[8].

Mathematical models are aimed to be in silico replicas of cellular reaction systems. They are able to describe biological phenomena like multistationarity and oscillations. They can also be used to determine under which conditions can occur chaos or can predict protein interactions which has not been detected yet. There are two main groups of models: stochastic and deterministic models. Since there are high numbers of each molecule in cells, deterministic models are used to describe cellular systems.

Once we have data and mathematical described models we can do parameter fitting which are normally kinetic parameters for mass action-, Michaelis-Menten- or other kinetics. For this inference different problems has to be challenged. Errors and noise in data is very likely and the data may have to be pretreated. Also the mathematical models can contain errors and can lead to wrong conclusions. Since these models are often systems of several differential equations we maybe also have computational issues.

Most researches in signal transduction are motivated by finding new targets for pharmaceuticals or predicting (adverse-) effects of drugs.

3 Deterministic inference

3.1 The model

Heinrich et al. presented a simplified model of protein kinase signal transduction to introduce mathematical definitions for signaling time, signal duration and signal amplitude of transduction cascades[6]. In it they assumed that and receptor R starts in an activated state and loses it's activity with the time. More concretely:

$$R(t) = \exp(-\lambda t)$$

λ determines how fast the receptor loses it's activity. The receptor phosphorylates a cellular protein kinase which again can phosphorylate another protein kinase etc. . Only phosphorylated protein-kinases may phosphorylate the next, while each activated protein kinase can be inactivated by a specific phosphatase. This model in ODEs is given by

$$\frac{dX_1}{dt} = \alpha_1 R(t) \left(1 - \frac{X_1}{C_1}\right) - \beta_1 X_1 \text{ and } \frac{dX_i}{dt} = \alpha_i X_{i-1} \left(1 - \frac{X_i}{C_i}\right) - \beta_i X_i .$$

α_i determines the kinetic parameter for the phosphorylation/activation, β_i for dephosphorylation/inactivation. X_i are the concentrations only from the activated kinases and C_i are the total concentrations of each protein kinases, which are assumed as time-constant. A visualization of this model is shown in figure 1.

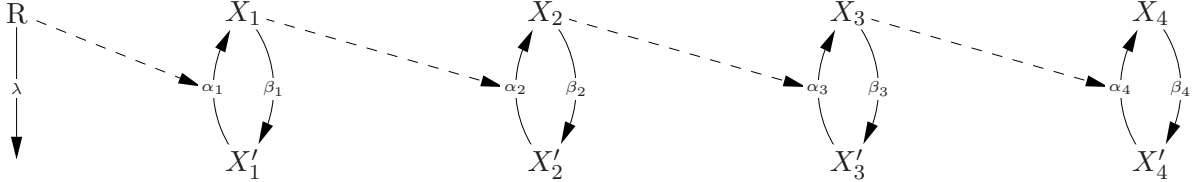


Figure 1: Simple signal transduction cascade with 4 steps.

3.2 Parameter optimization

Assuming that we have a sample of concentrations of signal transducers at different times, and the knowledge of the structure of the signal transduction pathway including a model consisting of ODEs, but no clues about the kinetic parameters values. The objective is now, to find parameters which numerically calculated time series with the ODEs fit the observed data as good as possible. To measure how good the parameters fitting the data the mean-square-error is usually used:

$$MSE = \frac{\sum_{i=1}^{|D|} (D_i - D'_i)^2}{|D|} .$$

D is the observed data and D' is the data calculated from the ODEs with suggested parameters. If we now consider the MSE as an function of the model parameters θ we can now use optimization algorithms to optimize the model parameters to minimize the $MSE(\theta)$. The algorithm I used for the following experiment is the Broyden–Fletcher–Goldfarb–Shanno algorithm. For the experiment I used the programming language R where a version of BFGS is already implemented.

3.3 Experiment and numerical results

To test the deterministic parameter optimization I simulated time-series with the parameters: $\alpha_i = 1$, $\beta_i = 0.6$, $C_i = 20$, $\lambda = 1$ and 5 steps ($i = 1..5$). From this time-series a sample of 13 observations was drawn. Pretending to have no knowledge about the parameters α_i , β_i and C_i we want to estimate these parameters. But since we have 15 parameters to estimate, the quality of the results and the runtime of the BFGS algorithm may be very bad. But in this model, we can divide our search space because α_1 , β_1 and C_1 don't depend on the other parameters, so we can

run BFGS first on these parameters, and use these results for estimation of the next 3 parameters, etc. . Table 1 shows the results of estimation with this procedure.

i	α_i	β_i	C_i
1	1.0067	0.6000	3.9998
2	0.9974	0.5982	15.000
3	1.0021	0.6007	15.000
4	1.0015	0.6000	14.999
5	1.0018	0.5999	15.000

Table 1: Estimated parameters with BFGS. Real values: $\alpha_i = 1$, $\beta_i = 0.6$, $C_i = 20$. Initial parameter for BFGS: $\alpha_i = \beta_i = 0.2$, $C_1 = 4$, $C_i = 15$ for $2 \leq i \leq 5$

All activation parameters α_i and inactivation parameters β_i are estimated exactly or closely to the true value. All C_i hardly moved from their initial values. This is probably of the small influence of these parameters on the systems dynamics: All amplitudes of X_i are much smaller than the $C_i = 20$ and this means that the fractions X_i/C_i in the ODEs are for all t 's almost 0.

To test how noise effects this method of parameter estimation I added to the sample different Gaussian error ($\sigma^2 = 0.001, 0.006, 0.011, 0.016, 0.021$), and measured the error of the estimated parameters α_i and β_i from the true parameter values. Estimation of C_i is no longer considered. To see also if more noise can be compensate by a higher number of observations, I did this test once with 15, once with 27 and once with 43 observations. The results are shown in figure 2.

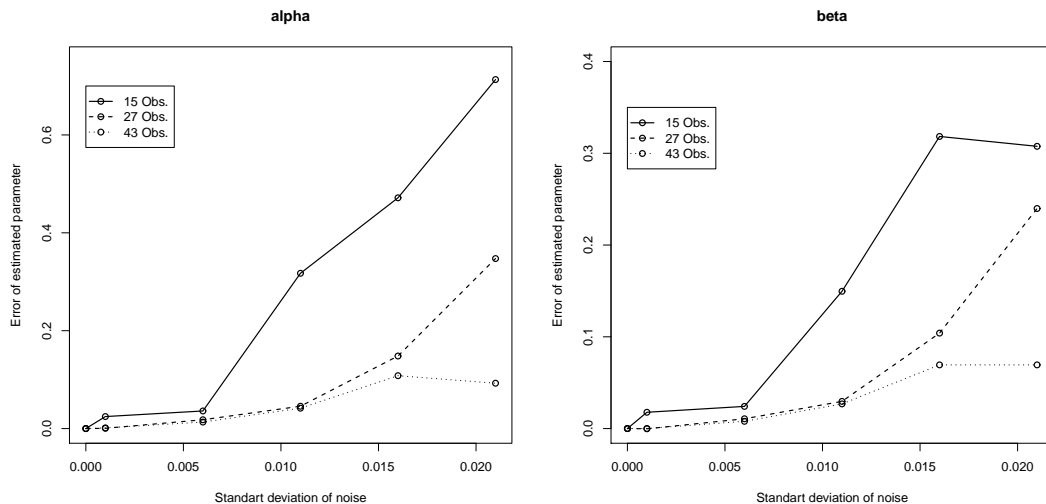


Figure 2: Error of estimated parameters calculated with data with different noise added.

This results are showing that the estimation error increases with more noise, but can greatly compensated by increasing the number of observations.

3.4 Discussion

This deterministic method of parameter estimation seems to work well if a few requirements are fulfilled:

- *only a few parameters has to be optimized or the parameter search-space can be divided in smaller optimization problems* — In reality this might be a rare situation, because signal transduction pathways often contain several feedback loops.
- *initial values are good guesses of true values* — Since this method is deterministic, the algorithm stops as soon as a local optimum is reached. Better parameter estimations with a smaller mean-square-error may be missed. Statistical methods may help to escape from local optima (see chapter 4).
- *parameters have a strong influence of system dynamics* — The BFGS optimization algorithm is a hill-climbing algorithm, this means it is dependent on gradients. If the mean-square-error does not change significantly with changing a parameter, this gradient will be almost 0 and this parameter can not be estimated acceptable.

4 Statistical inference

4.1 Methods and algorithms

In contrast to deterministic inference, the statistical inference methods consider the data (the observations) as well as the model parameters (kinetic parameters) as random quantities. These methods are based on Bayes' theorem.

4.1.1 Bayes' theorem

Let D be the observed data and θ the model parameters. The posterior distribution of θ is defined as

$$p(\theta|D) = \frac{p(D|\theta)p(\theta)}{p(D)} .$$

The marginal probability $p(D)$ can also be written as the total probability of D under all possible model parameters. This leads to

$$p(\theta|D) = \frac{p(D|\theta)p(\theta)}{\int p(D|\theta)p(\theta)d\theta} .$$

The likelihood distribution $p(D|\theta)$ is for all following calculations defined as

$$p(D|\theta) = \prod_{i=1}^{|D|} \frac{1}{\sigma_i^d \sqrt{2\pi}} \exp\left(-\frac{1}{2} \left(\frac{D'_i - D_i}{\sigma_i^d}\right)^2\right) .$$

D' is the matrix of concentrations/activities of all components at the same points of time as the data D were observed.

The choice for a priori distribution $p(\theta)$ can't usually be easy well founded, since we don't know, which model parameters are more likely than others. However, there are methods which don't use a priori distribution.

Since our model parameter space is often high dimensional it is mostly impossible to calculate the a posteriori distribution explicitly. One alternative method to approximate this distribution is the use of Monte-Carlo-Markov-Chain simulations.

4.1.2 Monte-Carlo-Markov-chain methods

MCMC methods are taking draw samples from a probability distribution. Therefor a Markov-chain is constructed, where the chain starts with an initial θ^1 , for each further element in the Markov-chain θ^i a candidate θ' is generated and $\theta^{i+1} = f(\theta', \theta^i)$ is calculated. The objective is to generate a Markov-chain which density converge to the disired distribution. However, we usually don't now this desired distribution, but if we can simplify our model we may calculate the a posteriori distribution explicitly, run our method to construct the Markov-chain and compare the MC density with the true distribution. The MCMC method I used is the Metropolis-Hastings algorithm.

Metropolis-Hastings algorithm

Suppose that we want to draw θ from the likelyhood density $p(D|\theta)$. We will need a candidate-generating density $q(\theta, \theta')$ and intital model parameters θ^1 . Each iteration i runs following steps:

1. generate a draw $\theta' \sim q(\theta', \theta^i)$
2. calculate $\alpha = \begin{cases} \min \left\{ \frac{p(D|\theta')q(\theta^i, \theta')}{p(D|\theta^i)q(\theta', \theta^i)}, 1 \right\} & \text{if } p(D|\theta^i)q(\theta^i, \theta') > 0 \\ 1 & \text{if } p(D|\theta^i)q(\theta^i, \theta') = 0 \end{cases}$
3. accept candidate θ' with probability α . Set $\theta^{i+1} = \theta'$ if accepted, if not set $\theta^{i+1} = \theta^i$.

There are different suggestions for the candidate-generating density q . The density I used for the experiments in chapter 4.2 is the normal distribution

$$q(\theta^i, \theta') = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{1}{2\sigma^2}(\theta^i - \theta')^2\right) \implies \theta' \sim N(\theta^i, \sigma^2) .$$

In appendix A an example illustrates the progress and convergence of the Metropolis-Hastings algorithm.

In the suggested function for q , the choice of σ^2 has no effect on the asymptotic density calculated by Metropolis-Hastings algorithm (if $\sigma^2 > 0$). Anyway, choosing σ^2 is critical: If it is too high, the algorithm will reject the candidates very often, because steps from θ^i to θ' are big and may miss local optima. On the other hand, if σ^2 is too low, the algorithm indeed will accept most candidates, but the 'walking speed' is very slow and the algorithm may need many iterations

til enough places have been visited in the parameter space to get a good approximation of the a posteriori density. It should be tried to choose a σ^2 to get an acceptance rate α between 0.23 and 0.5 [2].

Gibbs sampling

In inference tasks we usually have more than one model parameters and this leads to multivariate likelihood distributions $p(D|\theta)$ where θ is the vector of the parameters. This makes it crucial to think about a good candidate generating function. Single-component Metropolis-Hastings algorithms choosing one element θ_i at the time and calculate it's acceptance rate. Gibbs sampling is one case of these algorithms, where every generated candidate is accepted ($\alpha = 1$).

In appendix B is an example of Gibbs sampling of a two-dimensional correlated normal distribution.

4.2 Experiments and numerical results

To test statistical inference on signal transduction pathways I considered two models. One simple with one feedback-loop and one more complex model with various feedbacks. Since both models containing 9 and 11 kinetic parameters I used a single-component Metropolis-Hastings algorithm but not Gibbs sampling, because the acceptance rate α can vary between 0 and 1. The algorithm I used is analog to the algorithm used by Higham for inference of the feedback loop of the transcription factor HesI [4].

1. initialize θ_j^1 , $j = 1..k$
2. For $i = 1$ to N:
 - choose one j at random
 - set $\theta'_q = \theta_q^i$ for $q \neq j$
 - generate candidate $\theta'_j \sim N(\theta_j^i, \sigma^2) =: q(\theta_j^i, \theta'_j)$
 - generate $\alpha = \begin{cases} \min \left\{ \frac{p(D|\theta')q(\theta^i, \theta^j)}{p(D|\theta^i)q(\theta^j, \theta^i)}, 1 \right\} & \text{if } p(D|\theta^i)q(\theta^i, \theta^j) > 0 \\ 1 & \text{if } p(D|\theta^i)q(\theta^i, \theta^j) = 0 \end{cases}$
 - accept candidate θ' with probability α . Set $\theta^{i+1} = \theta'$ if accepted, if not set $\theta^{i+1} = \theta^i$

Experiment 1

The model is modified from model presented in 3.1. Therefor a negative feedback loop is added by assuming the kinase x_4 enhances the dephosphorylation of kinase x_2 . The kinetic parameter for this feedback will be γ . Together with all activating $\alpha_1 - \alpha_4$ and inactivating parameters $\beta_1 - \beta_4$ there are nine parameters to estimate and calculate their a posteriori densities, respectively. Figure 3 shows the visualization of this model.

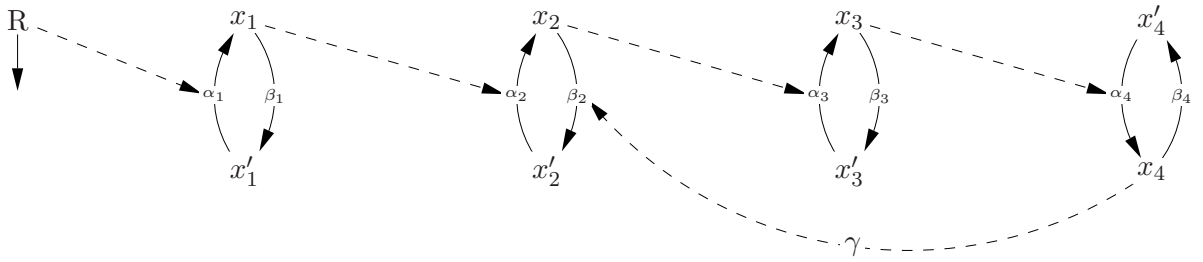


Figure 3: Simple signal transduction cascade with a single feedback loop.

The time series of all 4 activated kinase activities has been calculated numerically with Runge-Kutta-fourth-order method til $t = 20$. All α_i has been set 1, all β_i 0.6, and γ 10. To run Metropolis-Hastings a sample of 25 points of time with Gaussian noise added ($\sigma^2 = 0.01$) was drawn. Pretending to have no knowledge of α_i , β_i and γ I used the Metropolis-Hastings algorithm above to approximate the a posteriori densities of these 9 parameters. The algorithm was initialized with $\alpha_i = \beta_i = 0.5$ and $\gamma = 2$. σ^2 for generating the candidates was set 0.1 to obtain a mean acceptance rate around 0.5. Figure 4 shows the densities after 150000 iterations.

All densities should have their maximum near the red line if approximation is good. While α_1 and β_1 does so, are the density peaks of α_2 , β_2 and γ relatively far from the true parameter value. A possible explanation for this could be that the low start value of γ leads to a higher inactivating parameter β_2 to fit the time series curve of kinase x_2 better. The higher value of β_2 thereby leads also to a higher α_2 to achieve nearly the same kinase activity amplitude as in the sample. The under-estimation of γ has probably a methodical reason: The starting value for γ ($=2$) is quite far from the true γ ($=10$). Since the σ^2 for generating new candidates is 0.1 the steps are relatively short and the algorithm may need many iterations from 2 to 10. The slow movement get worse because of the small influence of this feedback loop parameter to systems dynamics. This leads to a lower difference between the candidate acceptance rates between candidates further from 10 and candidates closer to 10.

Experiment 2

For this experiment I used the MAP-kinase signalling model presented by Kwang-Hyun Cho et al.[5]. They considered experimental observed interactions between Raf-1, RKIP, RP, MEK and the MAP-Kinase ERK including the dissociation and association of RKIP and Raf-1. The mathematical model consist of 11 ODEs with 11 parameters ($k_1 - k_{11}$). In contrast the ODE-system of experiment 1, this system of ODEs leads to an equilibrium with concentrations unequal to 0. For simulation I calculated the 11 time series numerically til $t = 0.1$ with the parameters $k = (0.524, 0.0075, 0.61, 0.00251, 0.032, 0.81, 0.007, 0.069, 0.961, 0.00126, 0.872)$. Like in experiment 1, a sample with 18 points of time and with added Gaussian noise ($\sigma^2 = 0.01$) was drawn. Assuming this synthetic data as experimental observed data the Metropolis-Hastings algorithm was used to approximate all 11 parameters. The results are shown in figure 5.

Also in this experiment, the maxima of the parameter densities are around the real parameter values (red line). Admittedly, the density of k_9 has an unexpected shape with two optima, one at ≈ 1 and the other one at ≈ 3 . The higher value of k_9 means, that the association of phosphorylated RKIP with it's phosphatase RP is stronger than with a lower value. I don't have an explanation for this second peak at ≈ 3 yet, but one reason which has to be considered is, that 100000 iteration

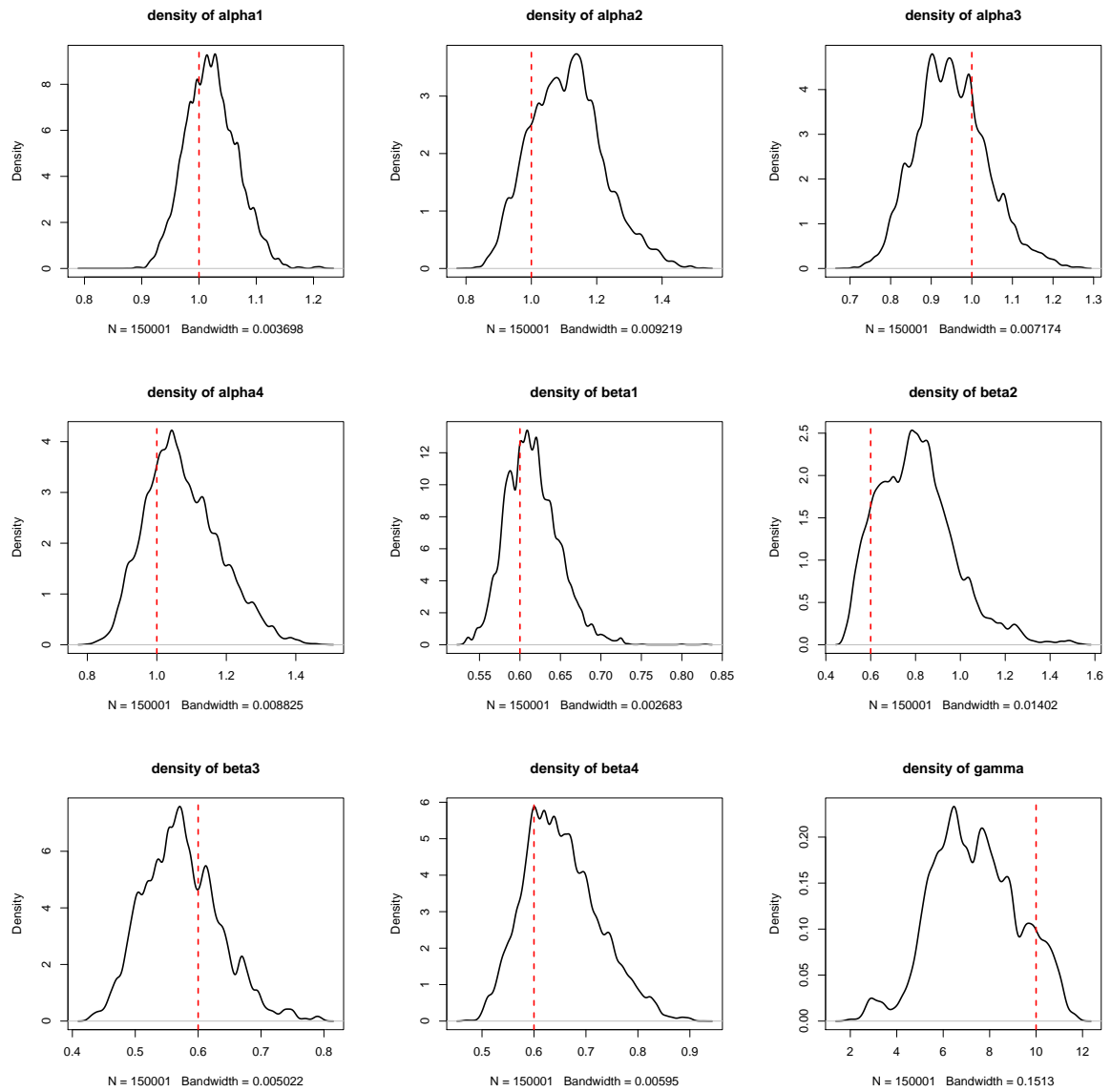


Figure 4: Densities (black) of kinetic parameters calculated with Metropolis-Hastings and 150000 iterations. The red line marks the value of the parameter which was used to generate the sample.

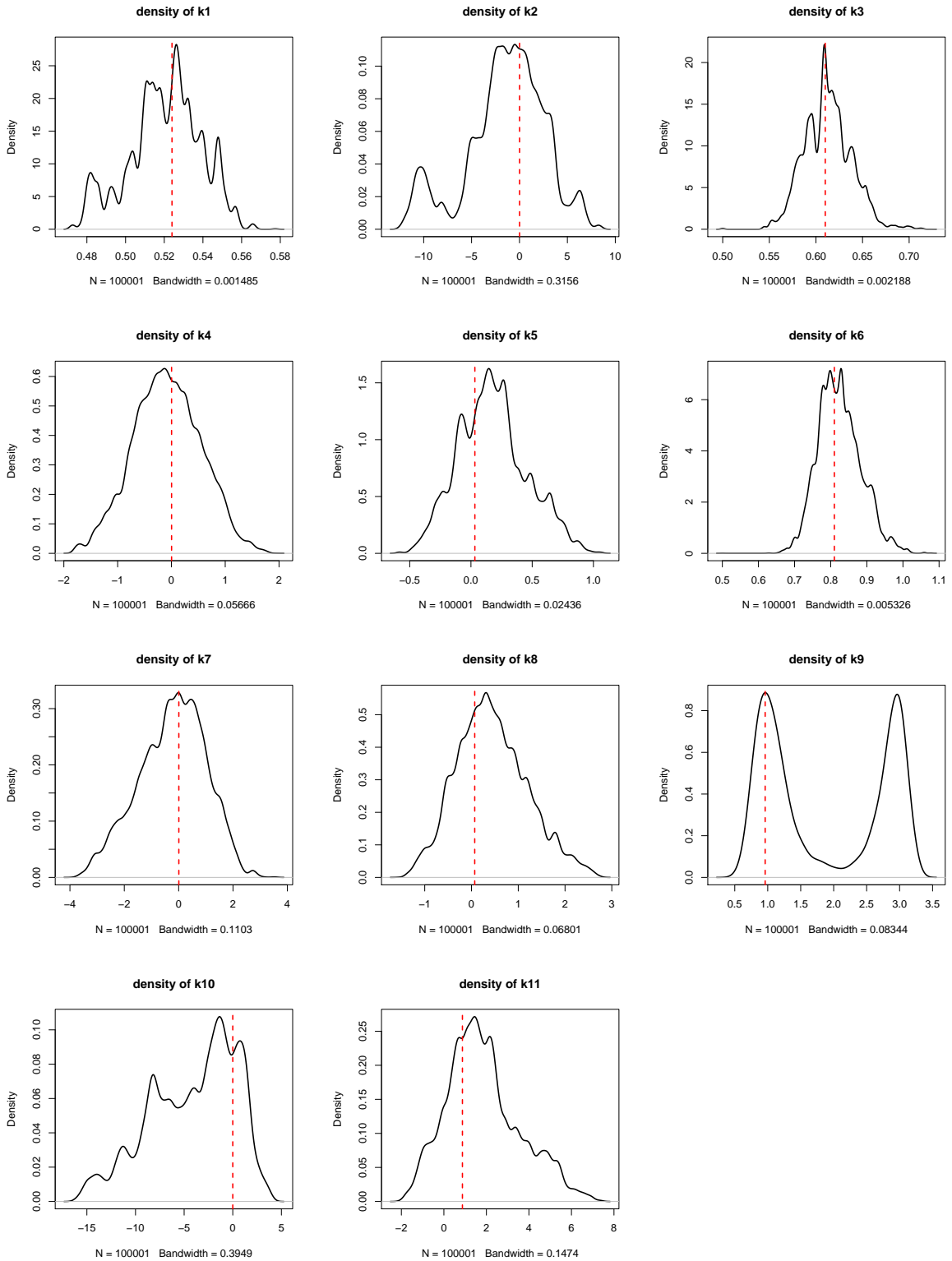


Figure 5: Densities (black) of kinetic parameters of experiment 2 calculated with Metropolis-Hastings. The red line marks the value of the parameter which was used to generate the sample. Algorithm parameters used: iterations = 100000, $\sigma^2 = 0.2$ for generating candidates and initial parameter values $k_i = 0.5$.

were maybe not enough to achieve a good approximation of the real a posteriori density of k_9 .

4.3 Discussion

This Metropolis-Hastings algorithm may be good in some cases to calculate the a posteriori densities of kinetic parameters. Once we have the densities, their maxima can be used as an estimation of the parameters real value. However, there are clues why these results has to be considered carefully. First of all, the algorithm parameter σ^2 for generating candidates does not influence the convergence density, but since we can only realize a finite number of iterations the quality of the resulting densities may differ extremely with different σ^2 (e.g. γ in experiment 1). The same effect can occur with the initial model parameter values. A possible method to reduce this effect is to use deterministic optimization[see chapter 3] to calculate better initial parameters. Additionally, model parameters with relative small influence on systems dynamics may need more iterations for a good density approximation.

Another assignment which has to be done is the prove, that my used algorithm really converge to the real a-posteriori distribution of the parameters. For this, a simple signal transduction pathway should be considered, where the a posteriori distribution can be calculated explicitly.

5 Conclusion

Both inference methods, deterministic and statistical, worked quite well on the example models or show at least convergence to the true model parameters. For the deterministic inference I assumed conditions which made this method more efficient but these conditions are probably not often met in biology. The main disadvantage of deterministic inference is that it only finds a local optimum. However, this parameter optimization can be used to generate better initial parameters for statistical inference.

Statistical inference has the ability to escape from local optima (e.g. parameter k_9 in figure 5) and returns more information about a model parameter when simulating it's a posteriori density. This information can be used for further analysis of the system. The results of the statistical parameter fitting are also showing, that this method is also strongly dependent on initial values for the parameters, but it's difficult to make good guesses if we have no knowledge about parameters value margins.

I assumed complete knowledge of the network and ODE structures, but these information may lack in reality. So other methods should be added to deal with this situations. The suggested statistical inference method should also be tested on more and different signal transductions pathways and data, to determine what can be modified in this method to make better parameter estimations.

References

- [1] S. Zimmermann, K. Moelling. *Phosphorylation and Regulation of Raf by Akt (Protein Kinase B)*. 1999, Science 286:1741–1744
- [2] G.O. Roberts, A. Gelman. *Weak convergence and optimal scaling of random walk Metropolis algorithms*. 1997, Ann. Appl. Probab. 7 (1): 110–120

- [3] D. Wilkinson “<http://www.mas.ncl.ac.uk/~ndjw1/teaching/sim/gibbs/gibbs.html>”. called 10/04/2009
- [4] C.F. Higham *Bifurcation analysis informs Bayesian inference in the Hes1 feedback loop*. 2009, BMC Systems Biology 3:12
- [5] Kwang-Hyun Cho, Sung-Young Shin, Hyun-Woo Kim, O. Wolkenhauser, B. McFerran, W. Kolch *Mathematical Modeling of the Influence of RKIP on the ERK Signaling Pathway*. 2003, Lecture Notes In Computer Science 2602:127-141
- [6] R. Heinrich, B.G. Neel, T.A. Rapoport *Mathematical Models of Protein Kinase Signal Transduction*. 2002, Molecular Cell, Vol. 9:957-970
- [7] W.R. Gilks, S. Richardson, D.J. Spiegelhalter *Markov Chain Monte Carlo in Practice*. 1995, Chapman & Hall, 512 pages
- [8] M. Nagasaki, S. Tasaki, H. Hata, M. Oyama, K. Ueno, R. Yoshida, T. Higuchi, S. Sugano, S. Miyano *Modeling and Estimation of Dynamic EGFR Pathway by Data Assimilation Approach Using Time Series Proteomic Data*. 2006, Genome Inform. 17(2):226-38

A Example: Metropolis-Hastings algorithm

Let $\pi(\theta)$ be the density of the standard normal distribution and the distribution where we draw a sample of θ . Calculated densities after different numbers of iterations are shown in figure 6.

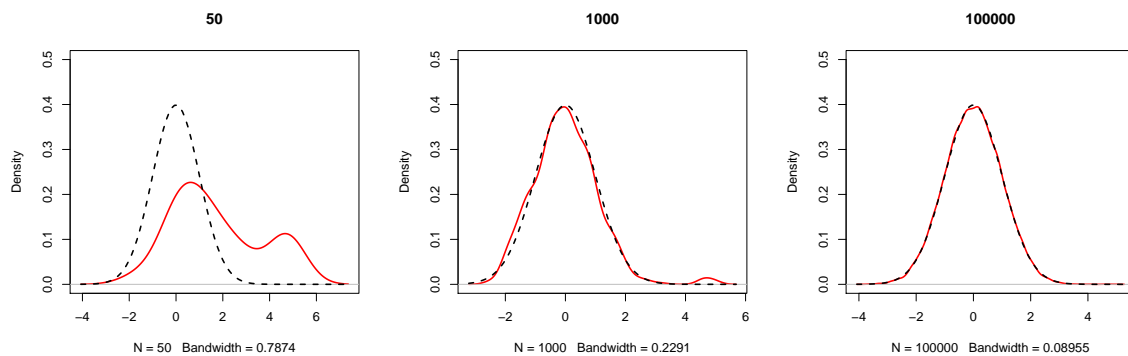


Figure 6: Densities of θ after 50/1000/100000 iterations of Metropolis-Hastings algorithm (red) with $\theta^1 = 5$ and $\sigma^2 = 1$. The standard normal distribution density is shown as the dashed black function.

B Example: Gibbs sampling

Assume we have a bivariate normal distribution with a correlation φ between both components. A Gibbs sampling algorithm may look like this (according to Wilkinson [3]):

1. initialize θ_1 and θ_2

2. repeat N times:

$$\theta_1 \sim N(\theta_2 * \varphi, \sqrt{1 - \varphi^2})$$

$$\theta_2 \sim N(\theta_1 * \varphi, \sqrt{1 - \varphi^2})$$

Figure 7 shows all visited points (θ_1, θ_2) running this algorithm with $\varphi = 0.6$.

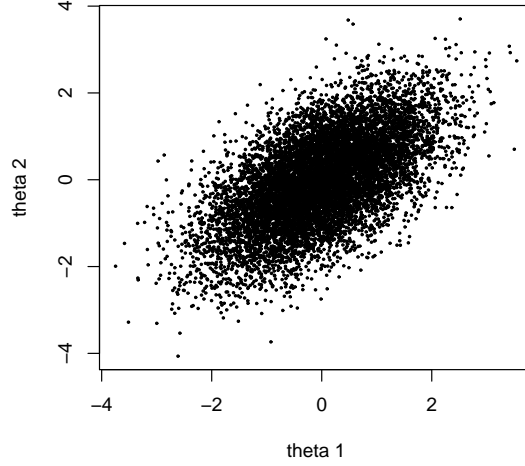


Figure 7: Visited points of Gibbs sampling in bivariate normal distribution with $\varphi = 0.6$, 10000 iterations and the initial $\theta_1 = 5$, $\theta_2 = -3$.

C Model ODEs for Experiments in 4.2

Experiment 1

$$\begin{aligned} \frac{dx_1}{dt} &= \alpha_1 R(t) \left(1 - \frac{x_1}{C_1}\right) - \beta_1 x_1 \\ \frac{dx_2}{dt} &= \alpha_2 x_1 \left(1 - \frac{x_2}{C_2}\right) - \beta_2 x_2 (1 + \gamma x_4) \\ \frac{dx_3}{dt} &= \alpha_2 x_2 \left(1 - \frac{x_3}{C_3}\right) - \beta_3 x_3 \\ \frac{dx_4}{dt} &= \alpha_2 x_3 \left(1 - \frac{x_4}{C_4}\right) - \beta_4 x_4 \\ R(t) &= \exp(-\lambda t) \quad , \quad \lambda = 1 \quad , \quad C_i = 20 \end{aligned}$$

Experiment 2

$$\begin{aligned} \frac{dx_1}{dt} &= -k_1 x_1 x_2 + k_2 x_3 + k_5 x_4 \\ \frac{dx_2}{dt} &= -k_1 x_2 x_2 + k_2 x_3 + k_{11} x_{11} \\ \frac{dx_3}{dt} &= k_1 x_1 x_2 - k_2 x_3 - k_3 x_3 x_9 + k_4 x_4 \\ \frac{dx_4}{dt} &= k_3 x_3 x_9 - k_4 x_4 - k_5 x_4 \\ \frac{dx_5}{dt} &= k_5 x_4 - k_6 x_5 x_7 + k_7 x_8 \\ \frac{dx_6}{dt} &= k_5 x_4 - k_9 x_6 x_{10} + k_{10} x_{11} \\ \frac{dx_7}{dt} &= -k_6 x_5 x_7 + k_7 x_8 + k_8 x_8 \\ \frac{dx_8}{dt} &= k_6 x_5 x_7 - k_7 x_8 - k_8 x_8 \\ \frac{dx_9}{dt} &= -k_3 x_3 x_9 + k_4 x_4 + k_8 x_8 \\ \frac{dx_{10}}{dt} &= -k_9 x_6 x_{10} + k_{10} x_{11} + k_{11} x_{11} \\ \frac{dx_{11}}{dt} &= k_9 x_6 x_{10} - k_{10} x_{11} - k_{11} x_{11} \end{aligned}$$