Netherlands Consortium for
Systems Biology

A Bayesian Approach to Targeted Experimental Design

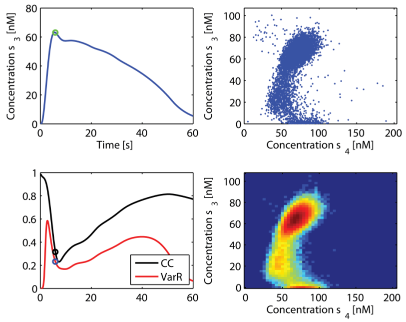
Joep Vanlier

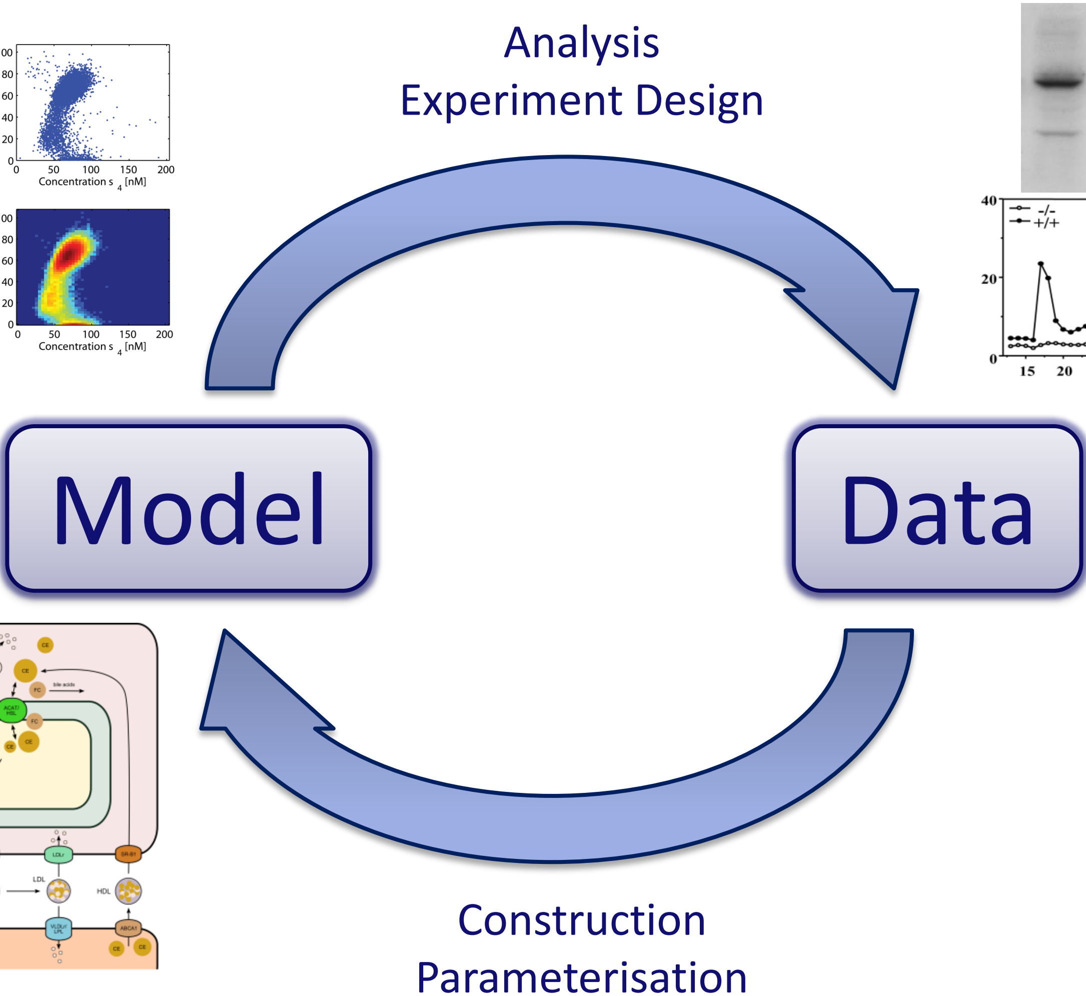
Technische Universiteit
Eindhoven
University of Technology

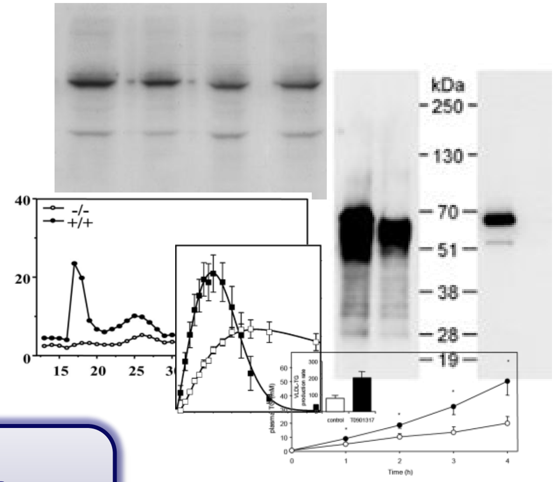
Where innovation starts

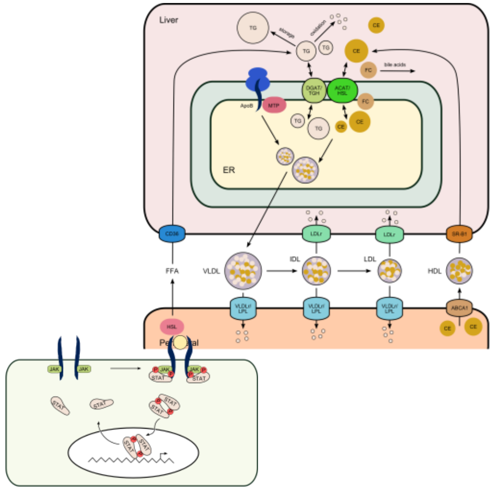
The approach

Vanlier et al, 2012

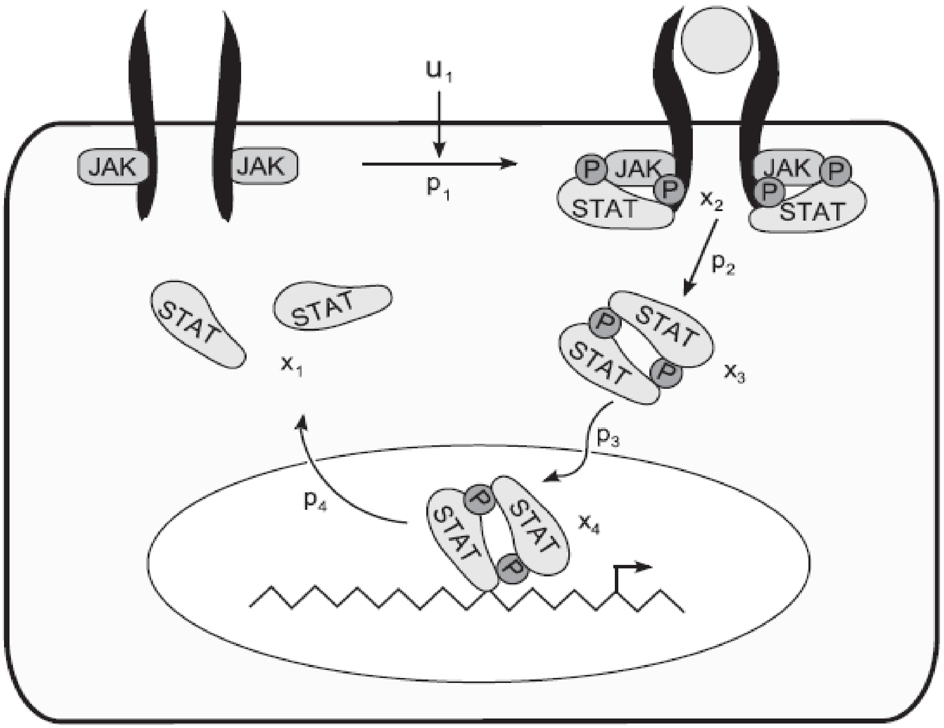








Model: JAK-STAT signaling pathway



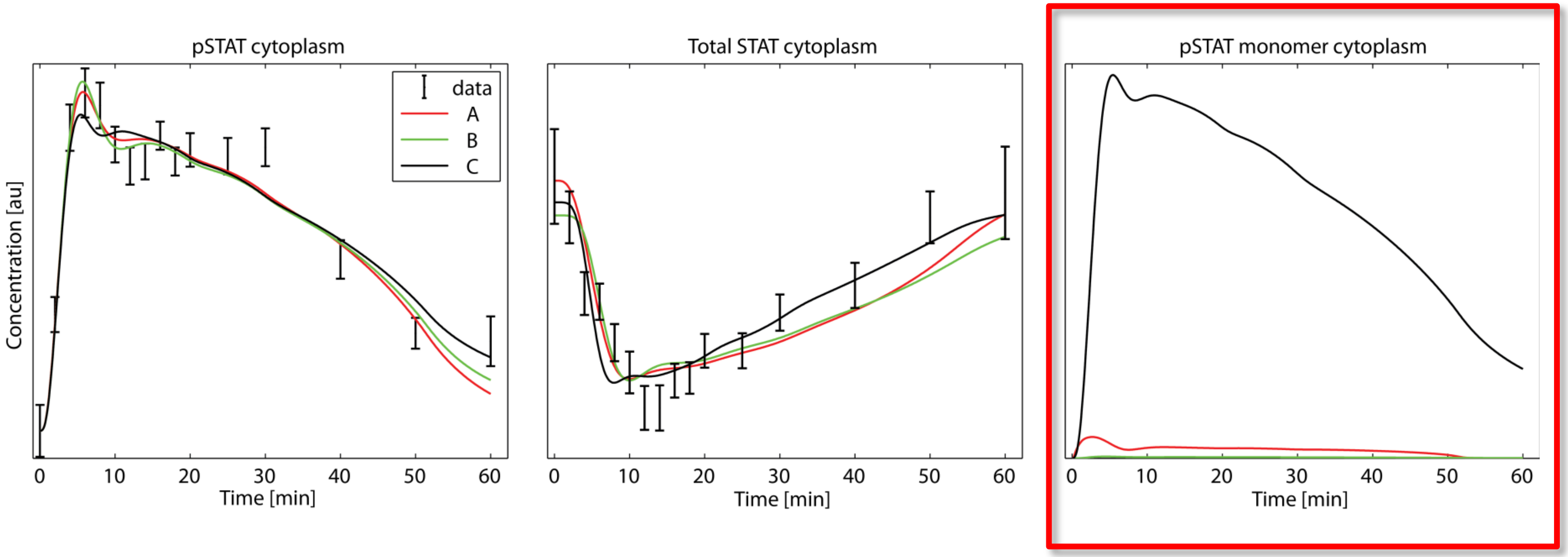
Data is semi-quantitative
→Introduces scaling and offsets

Only sums of states measured

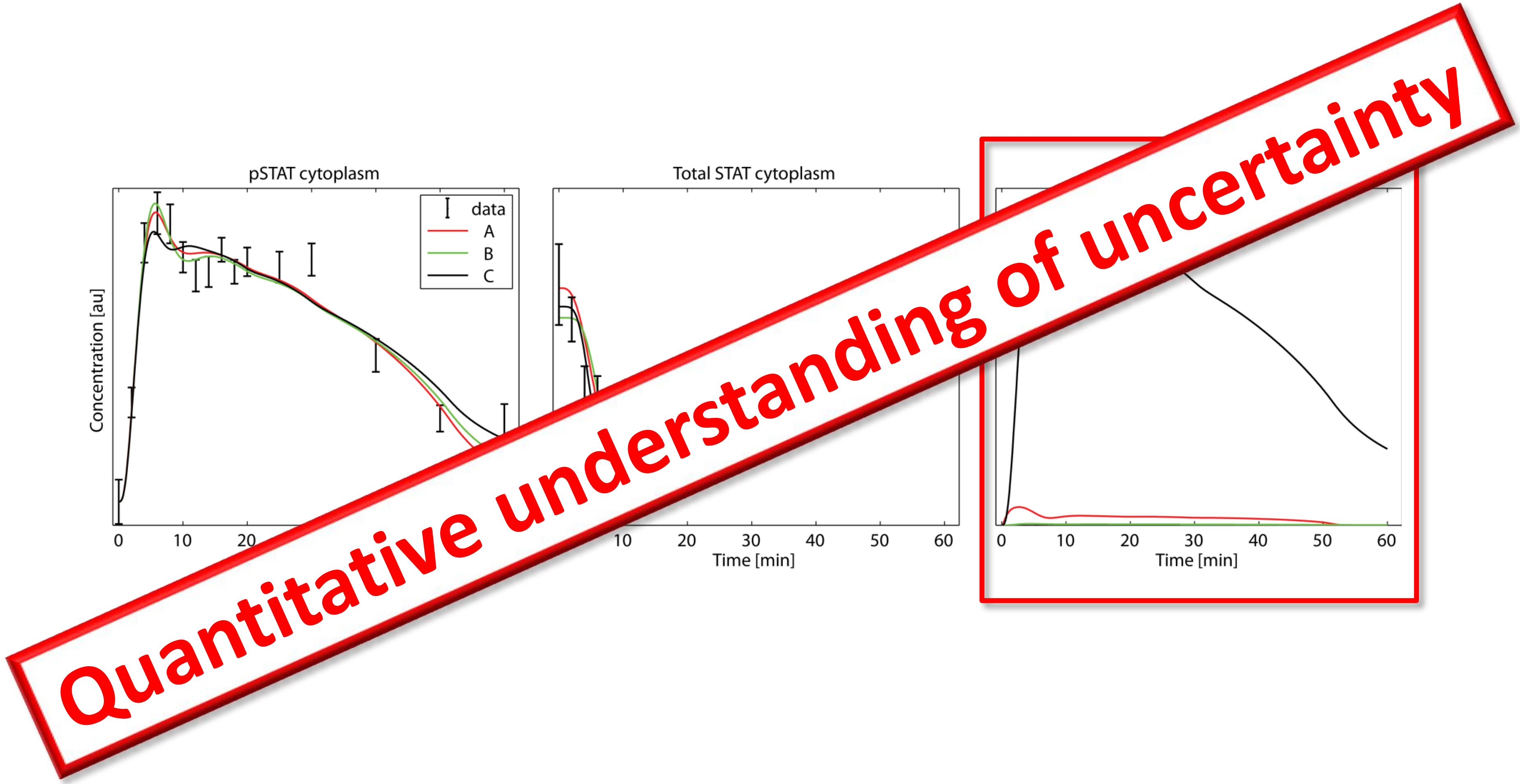
Small enough to perform
in-silico validation experiments

Raue et al, *Bioinformatics* (2009)

JAK-STAT


pSTAT cytoplasm
Total STAT cytoplasm
pSTAT monomer cytoplasm
data
A
B
C
Concentration [au]
0 10 20 30 40 50 60
Time [min]
0 10 20 30 40 50 60
Time [min]
0 10 20 30 40 50 60
Time [min]

JAK-STAT



pSTAT cytoplasm
Total STAT cytoplasm
data
A
B
C
Concentration [au]
0
10
20
30
40
50
60
Time [min]
Quantitative understanding of uncertainty

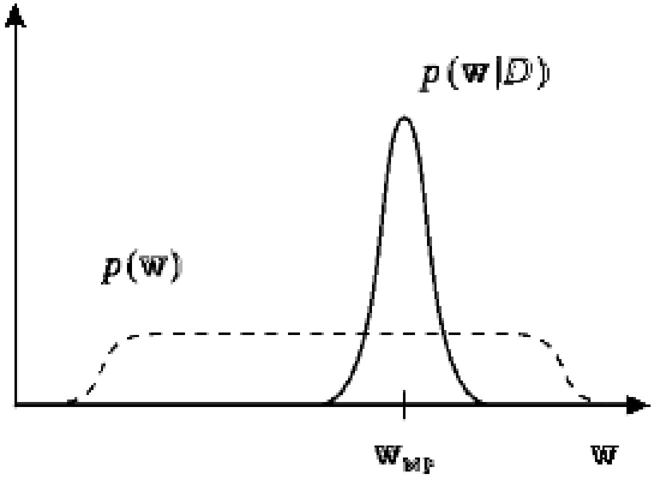
Propagating uncertainty to parameters

Stochastic methods:
Monte Carlo

Prior distribution

Data

Posterior Distribution

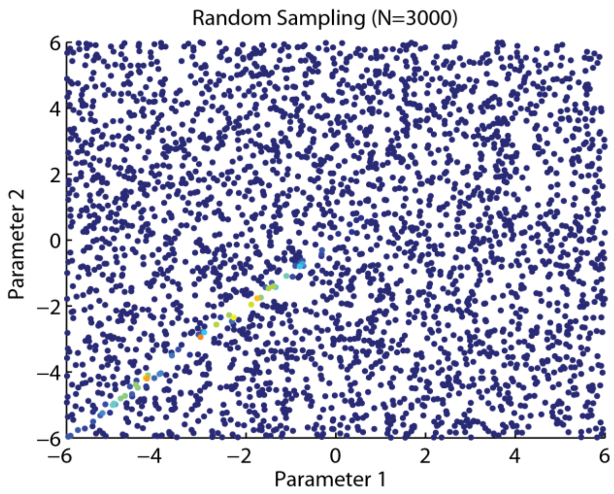


Stochastic methods:
Monte Carlo

Random sampling

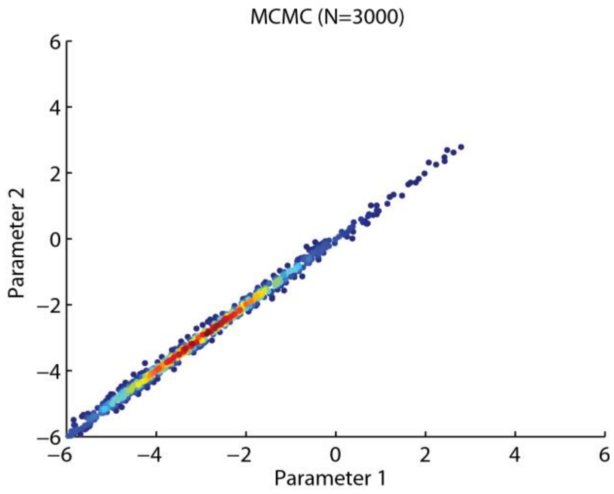
Most samples fit the data poorly

Slow gain in information



Markov Chain Monte Carlo:

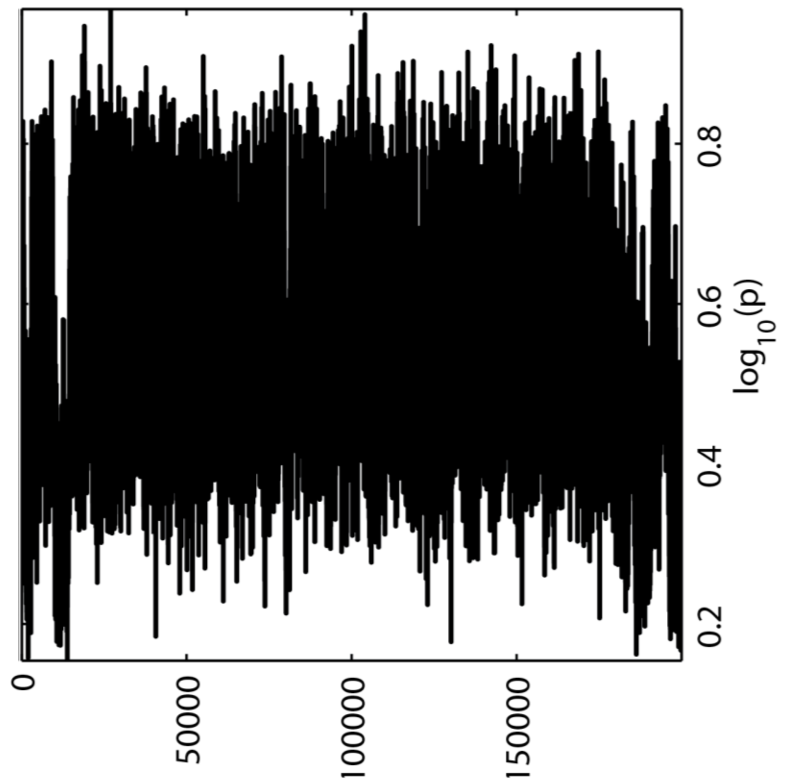
Sample in proportion to probability (proportional to likelihood and prior)



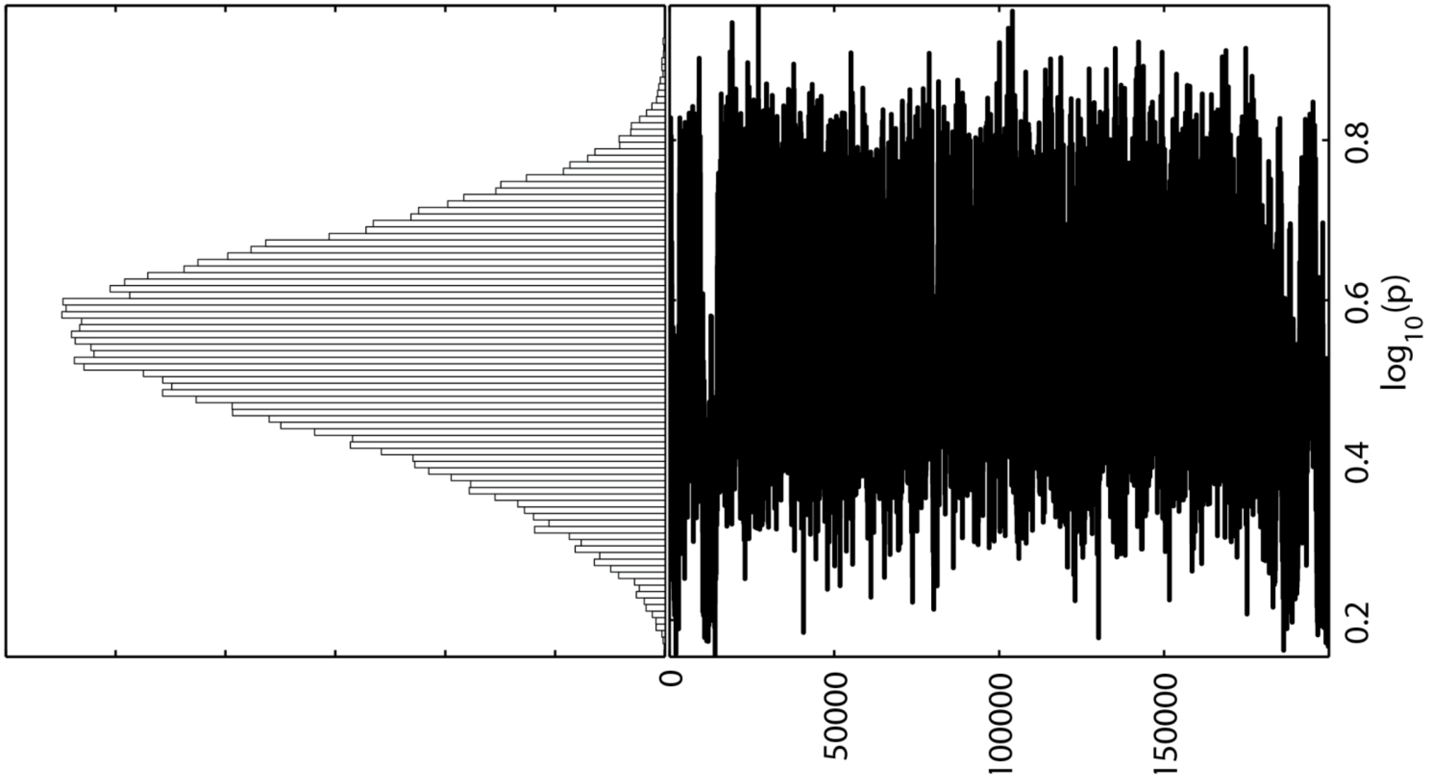
Stochastic methods:
Markov Chain Monte Carlo

- **Models often non-identifiable**
- **We check this by performing a Profile Likelihood prior to the Bayesian sampling**
- ***Non-identifiable parameters require an informative prior***
- ***Identifiable parameters inferred from data***

Posterior distribution

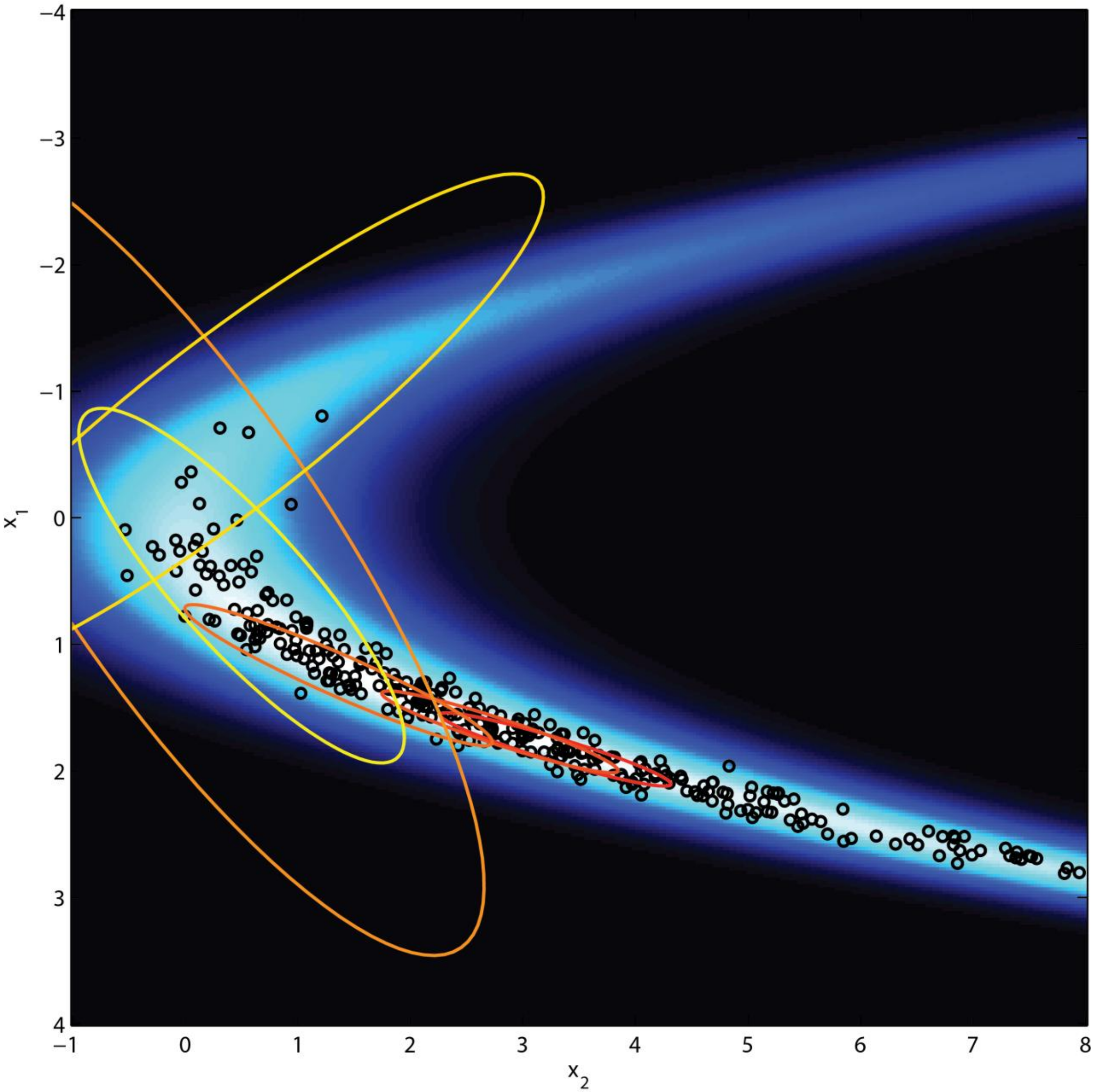


Posterior distribution



Technische Universiteit **Eindhoven** University of Technology

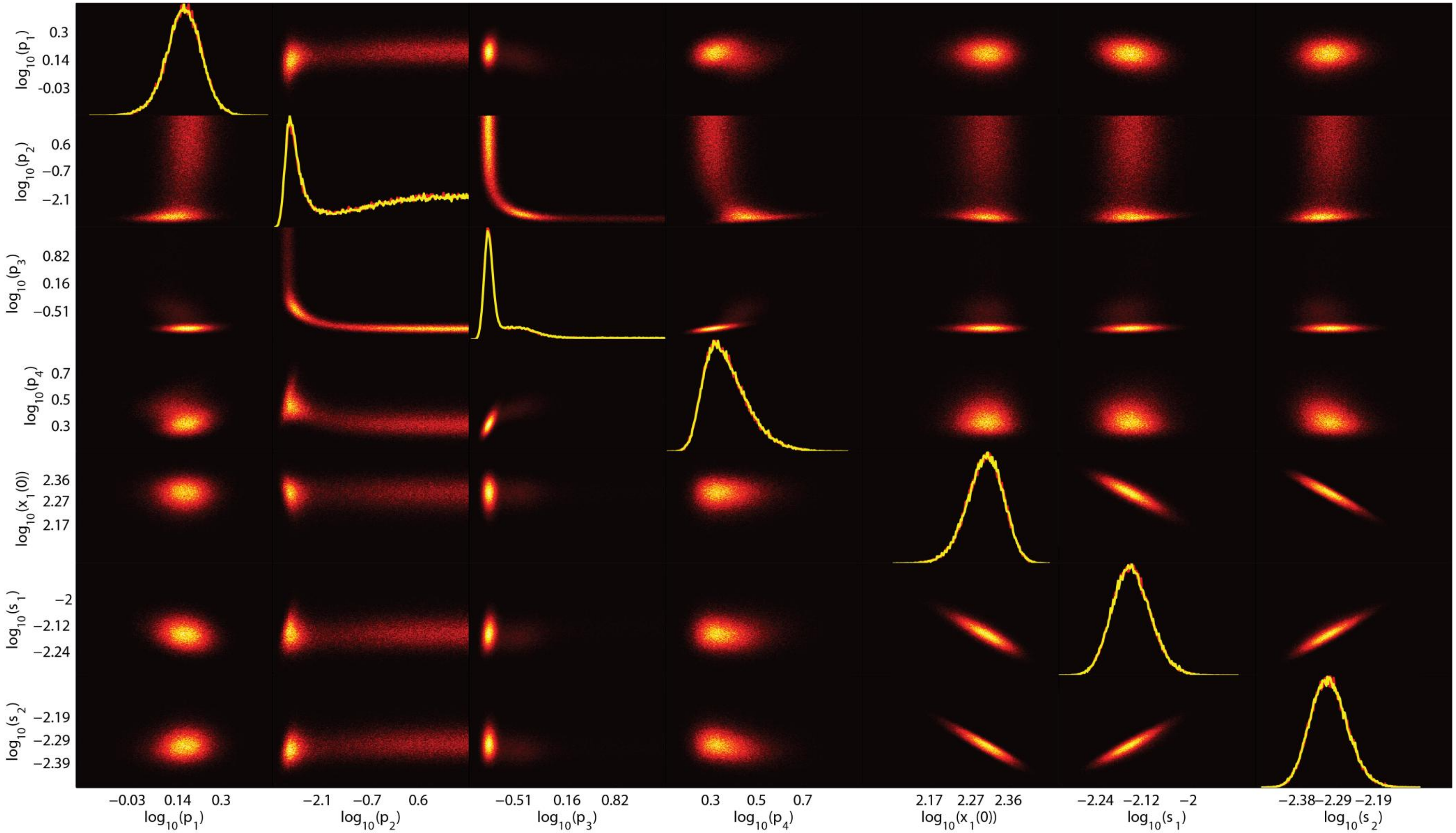
Stochastic methods:
Markov Chain Monte Carlo



Proposal Distribution

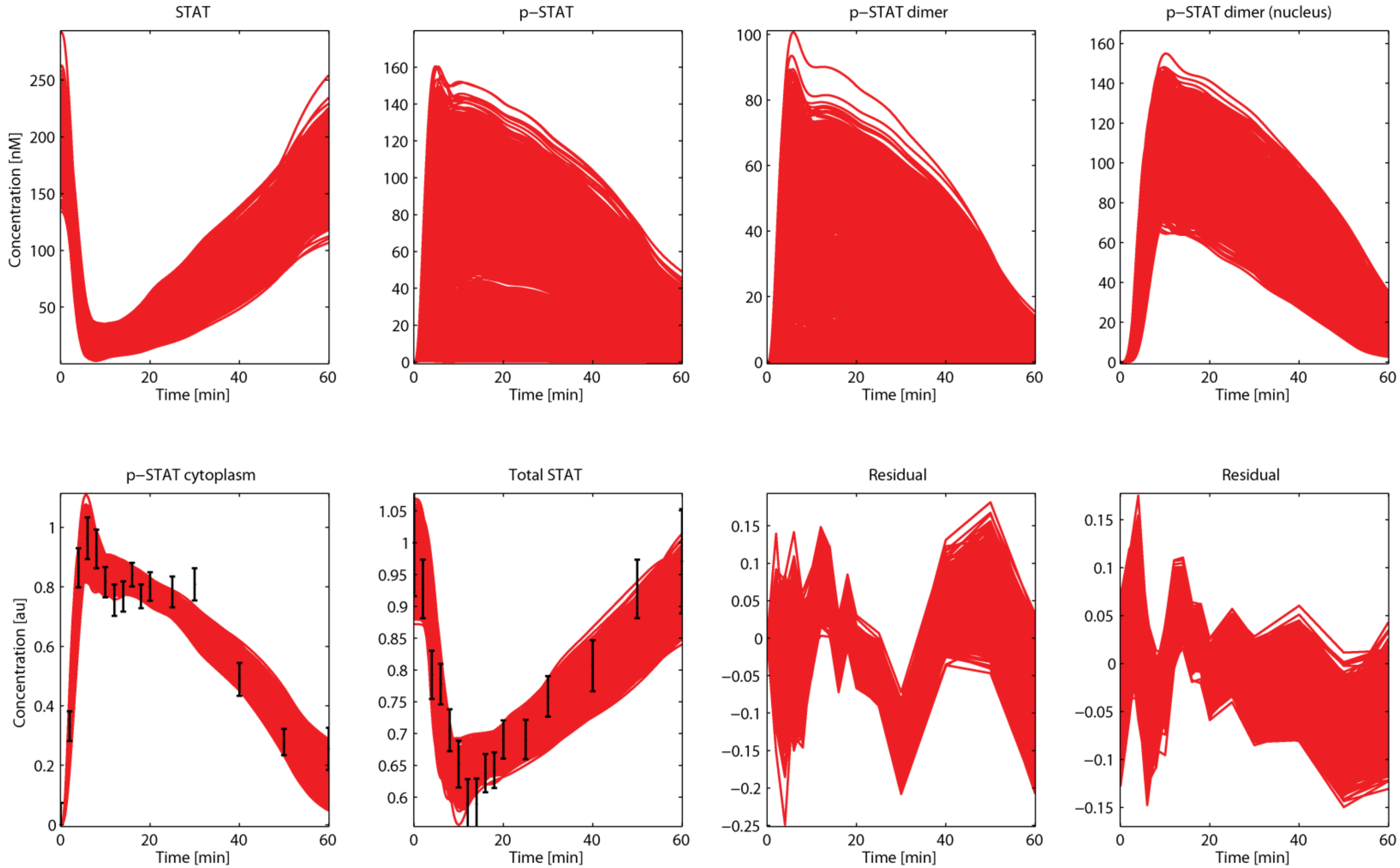
- **Adapt to local geometry**
- **Approximated Fisher Information Matrix**
- **First order sensitivities**
- **Trust Region**

Posterior distribution

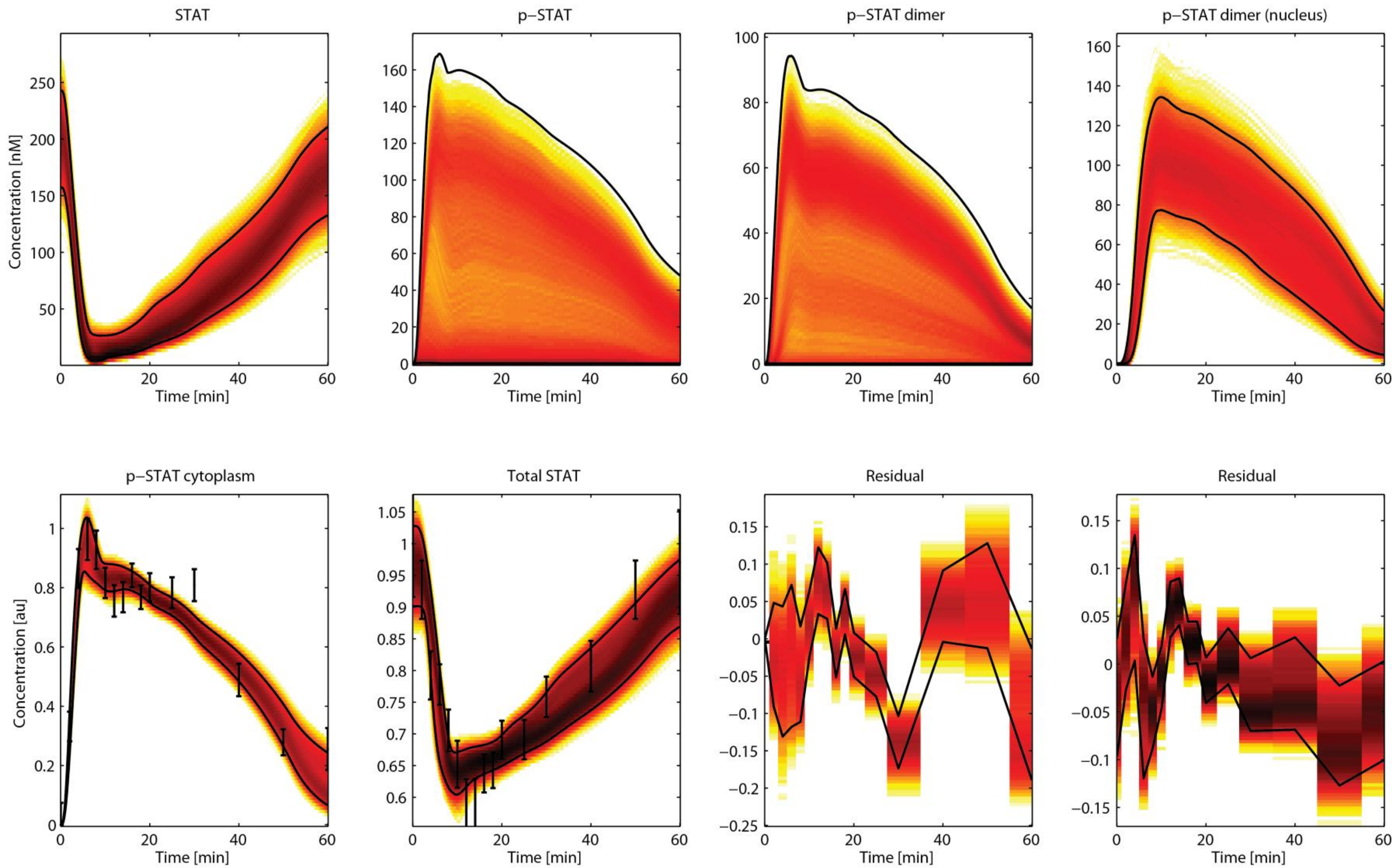


How does this relate to our predictions?

Posterior Predictive Distribution (PPD)

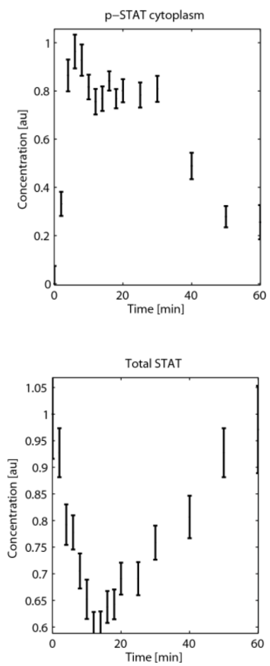


Posterior Predictive Distribution (PPD)

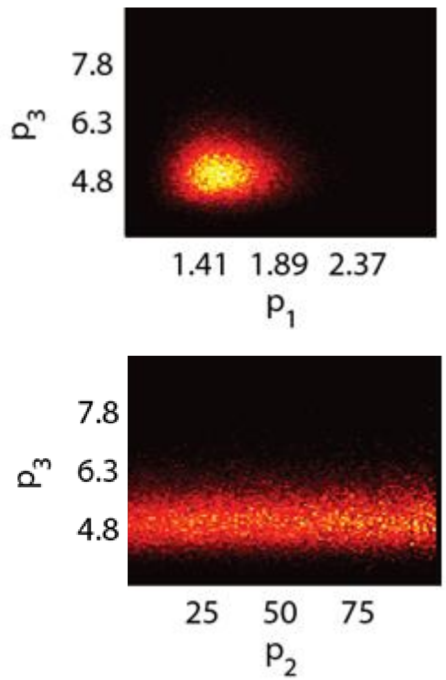


Posterior Predictive Distribution (PPD)

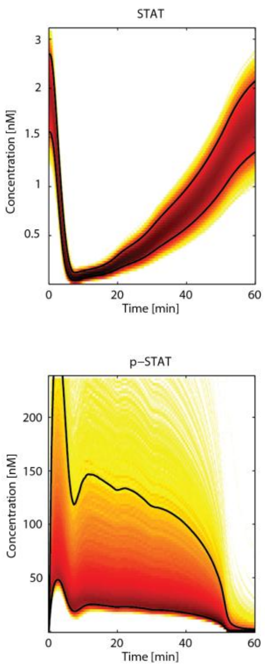
Data



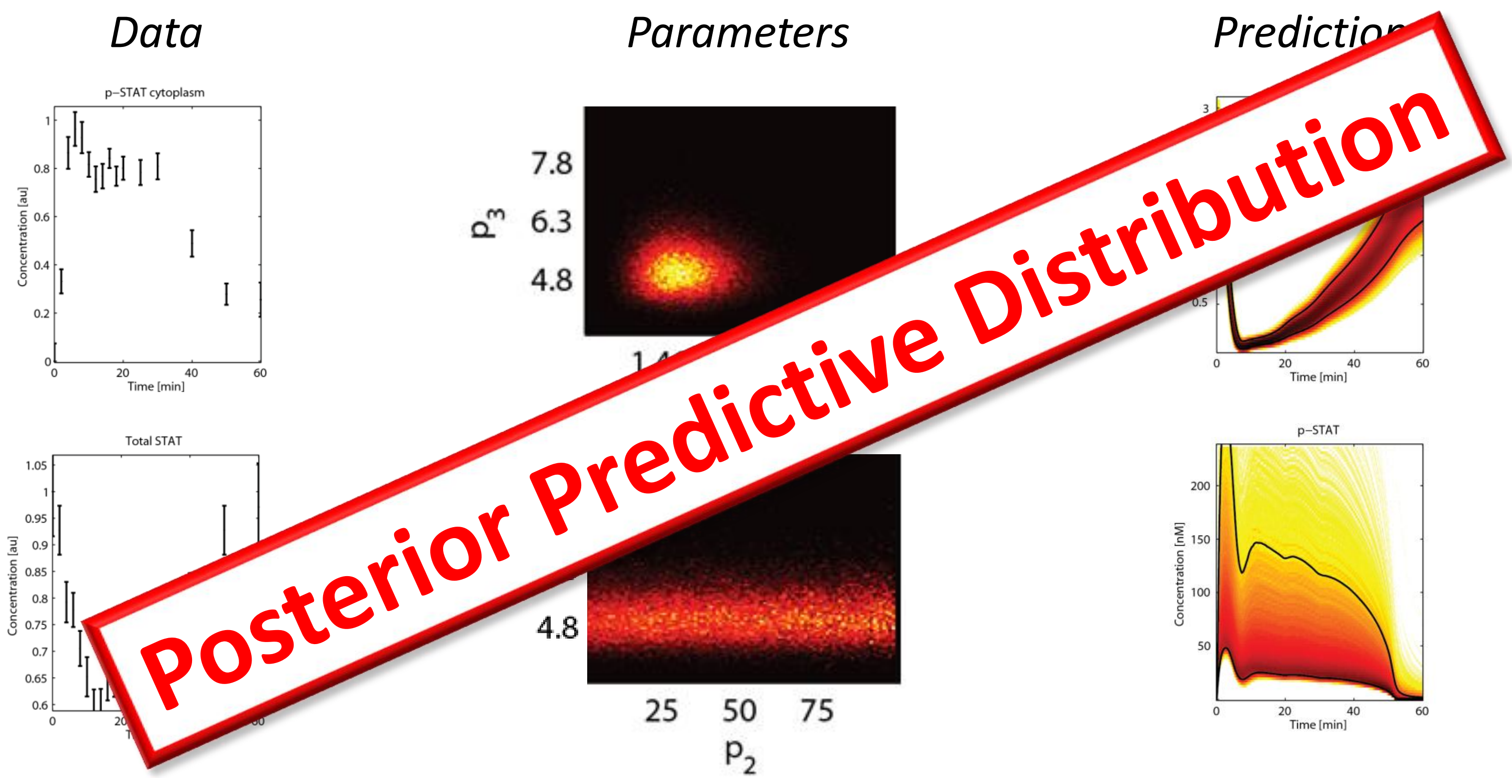
Parameters



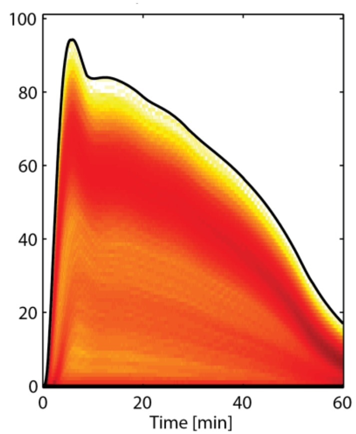
Predictions

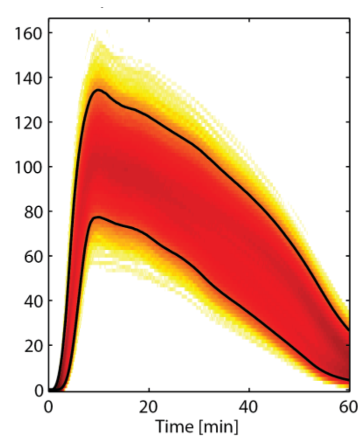


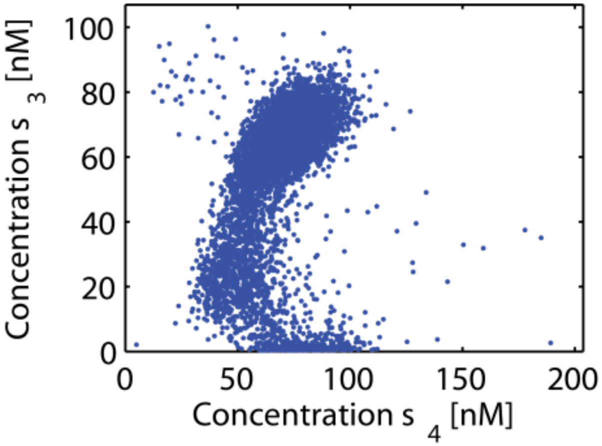
Posterior Predictive Distribution (PPD)



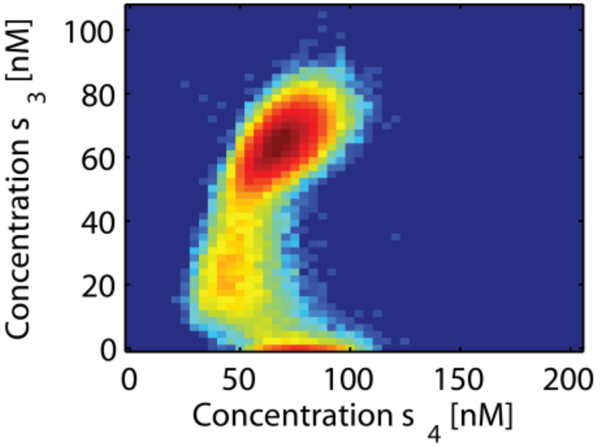
Posterior Predictive Distribution (PPD)







$$\frac{d\vec{x}}{dt} = f(\vec{\theta})$$
$$\vec{y} = g(\vec{x}, \vec{\theta})$$



Posterior Predictive Distribution (PPD)

- **Note that Posterior Predictive Distributions can be obtained for any quantity computable from model simulations**
  - **Response time of the system**
  - **Time to peak**
  - **Area under curves**
  - **Ratios**

  - **Anything that can be mathematically obtained from model simulations**

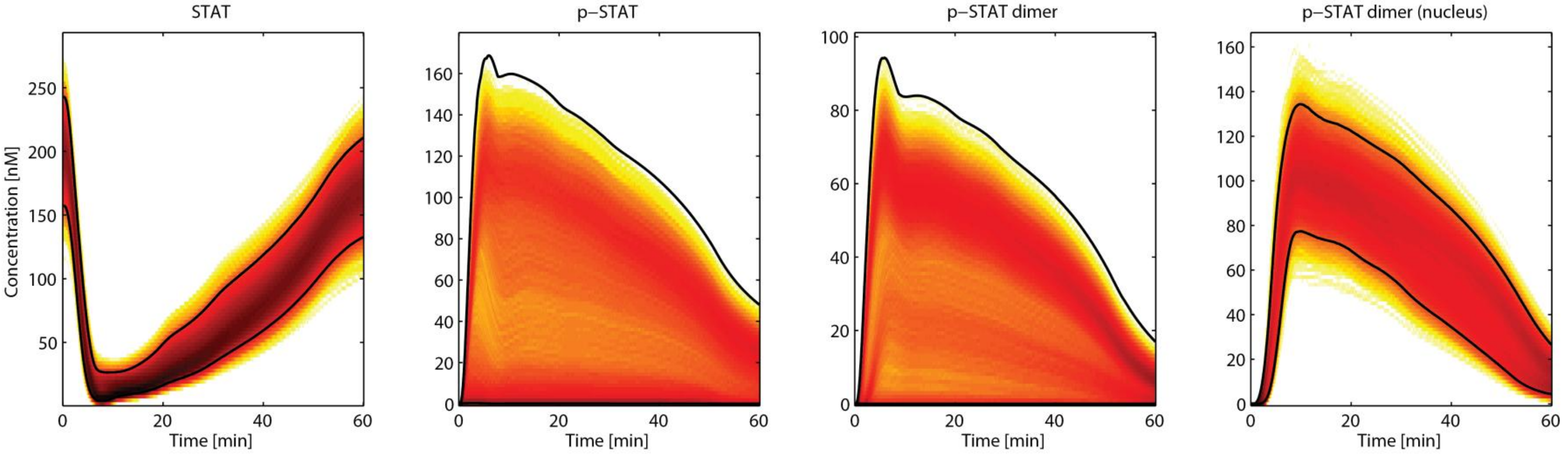
Experiment design

Exploiting the PPD

Vanlier et al, *Bioinformatics* (2012)

Efficacy of a new measurement

Posterior Predictive Distribution (PPD)



Strive for variance reduction

$$1 - \frac{V_{new}}{V_{post}}$$

0 - No variance reduction

1 - High variance reduction

Experimental Design

Our hypothesis is based on unmeasurable prediction B

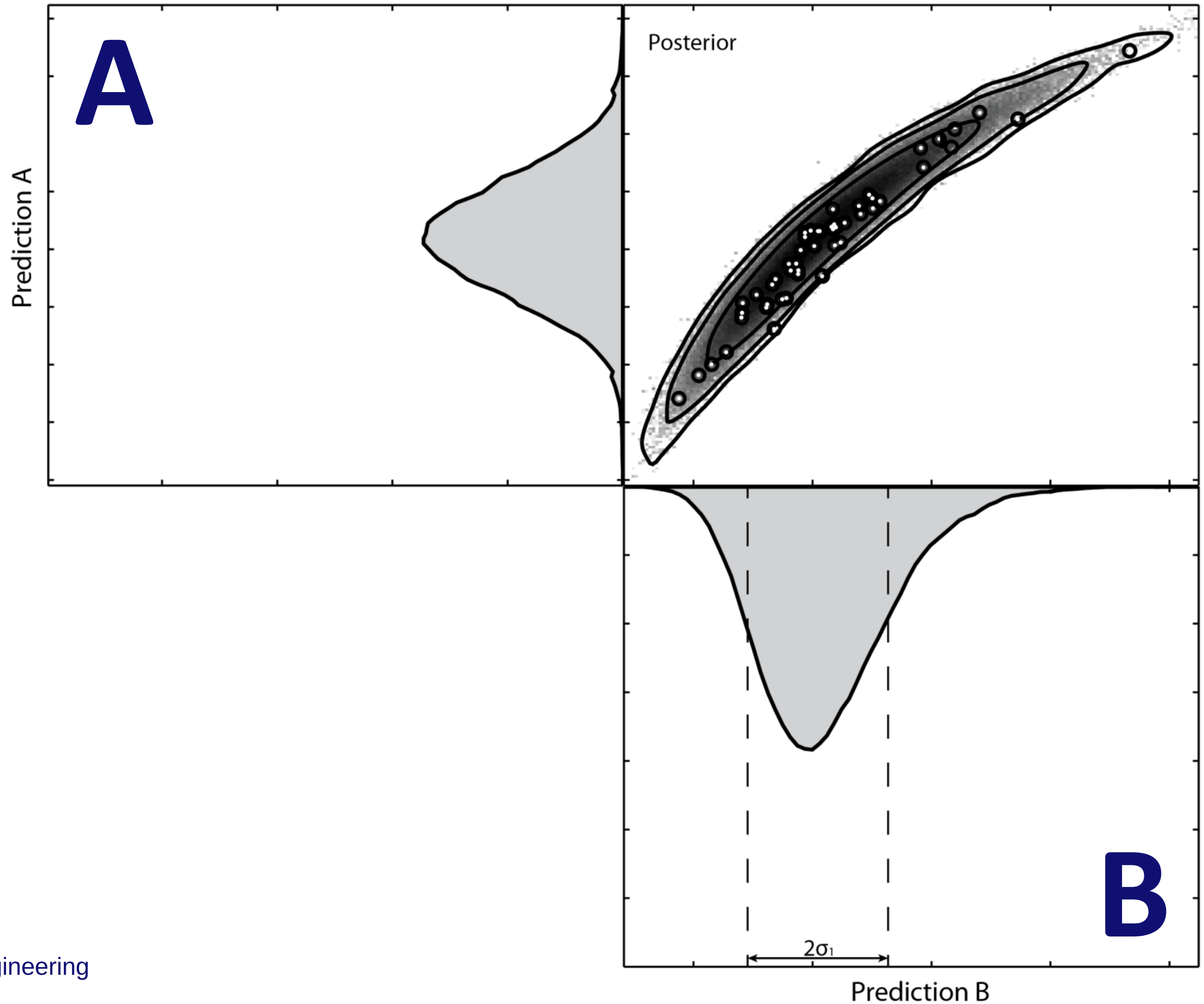
Which measurement gives us the most useful information to constrain B?

Constraints!

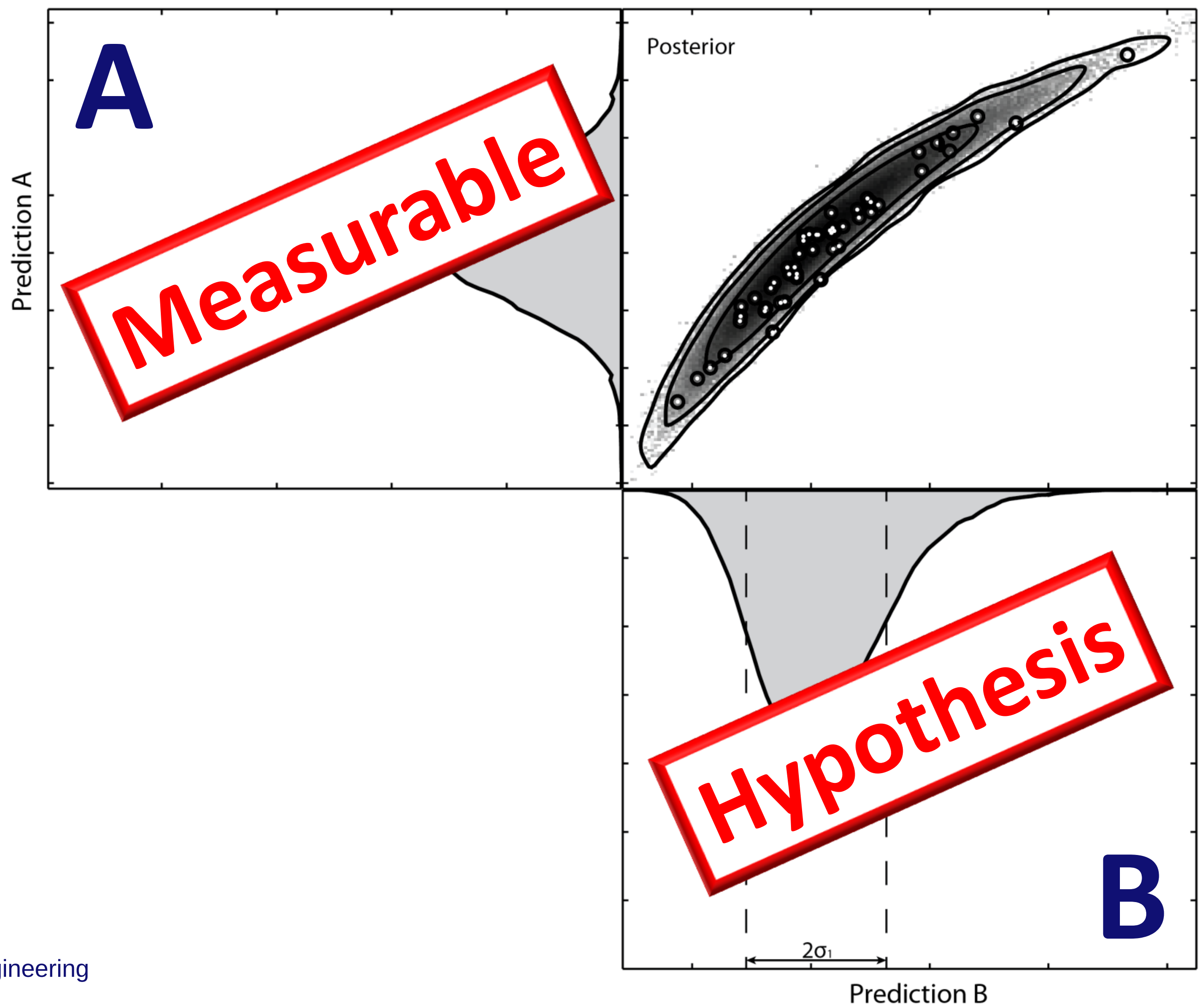
- **Only specific quantities are measurable**
- **Measurement hampered by noise**
- **System *partially* observed**

Technische Universiteit Eindhoven University of Technology

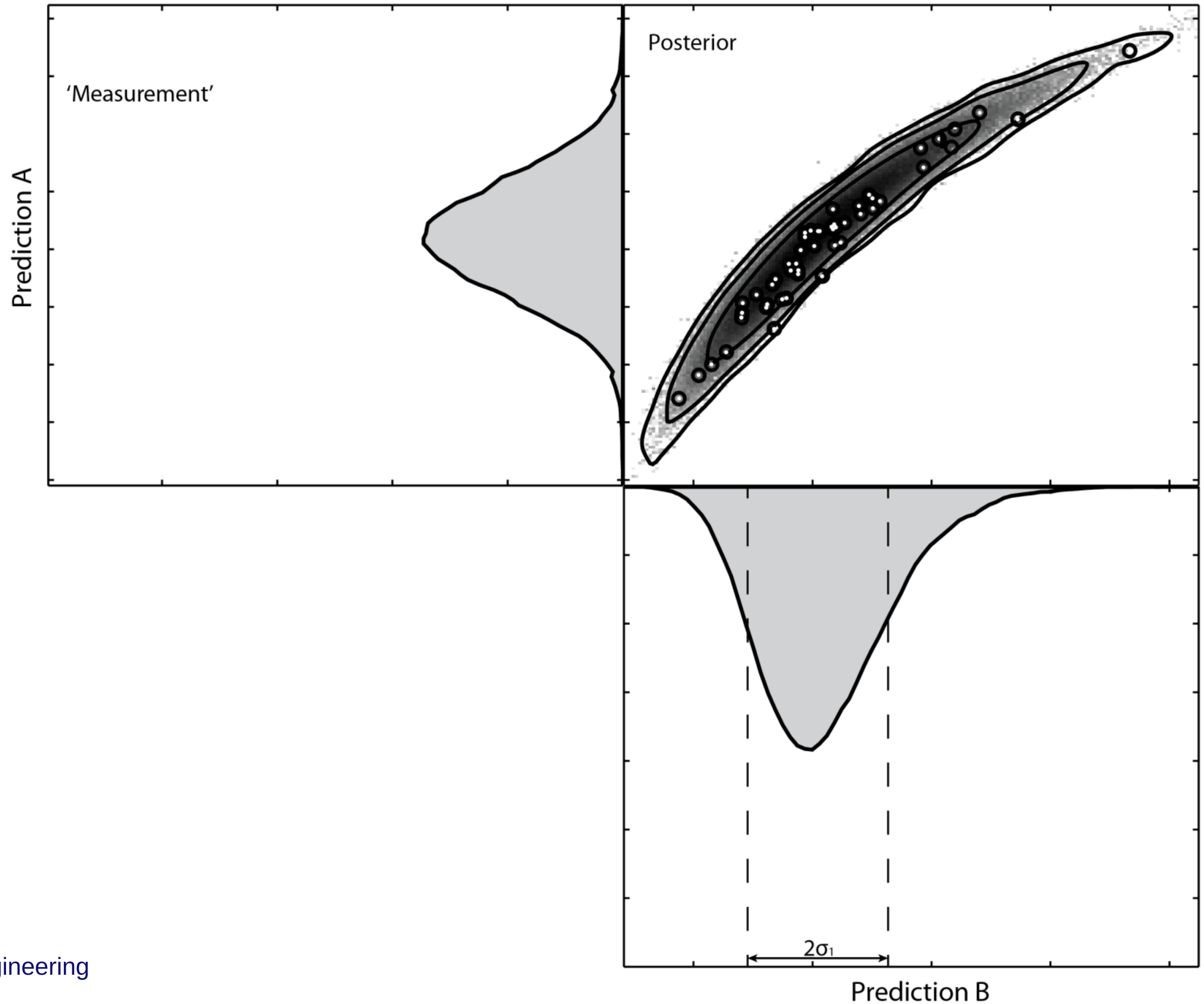
Posterior Predictive Distribution



Posterior Predictive Distribution

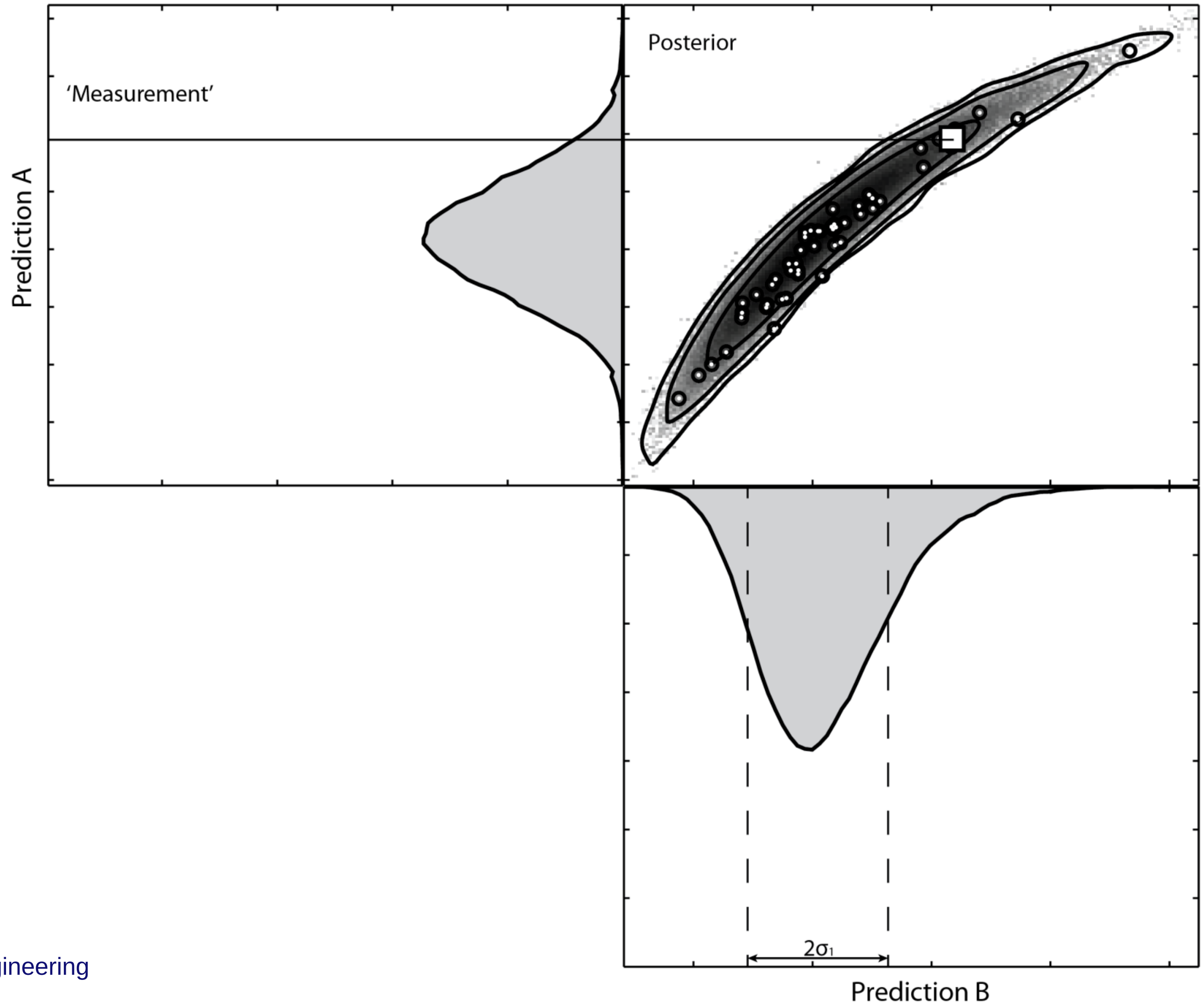


Posterior Predictive Distribution

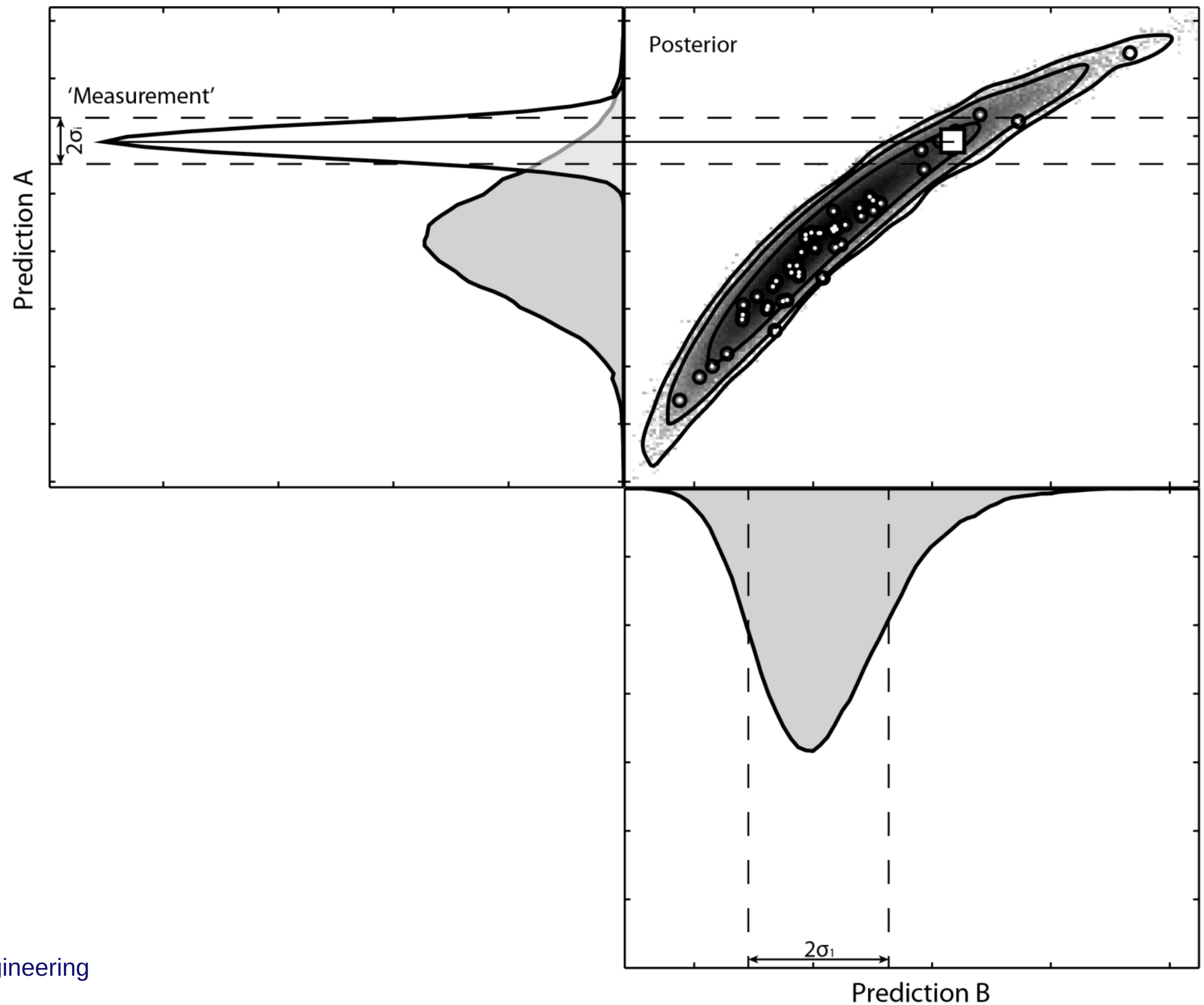


A new measurement

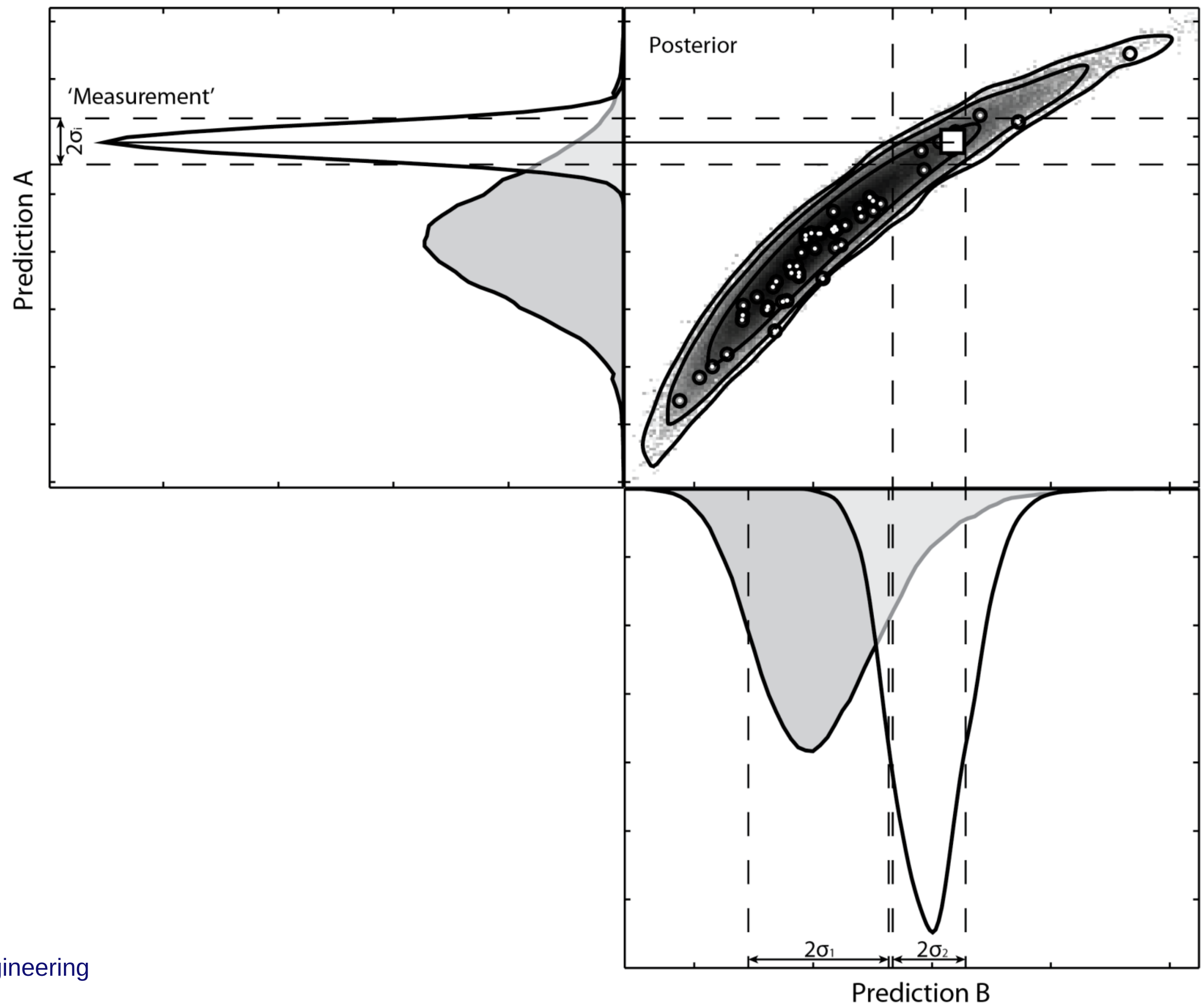
Posterior Predictive Distribution



Posterior Predictive Distribution



Posterior Predictive Distribution



Posterior Predictive Distribution

Problem 1:

New MCMC for every potential experiment is too much computational effort!

Posterior Predictive Distribution

Sampling the updated posterior is not required.

Variance can be expressed in terms of expected values.

$$Var[y] = E[y^2] - E[y]^2$$

$$E[y] = \int_{\Omega} p(\vec{\theta}) y d\theta_1 ... d\theta_n$$

Posterior Predictive Distribution

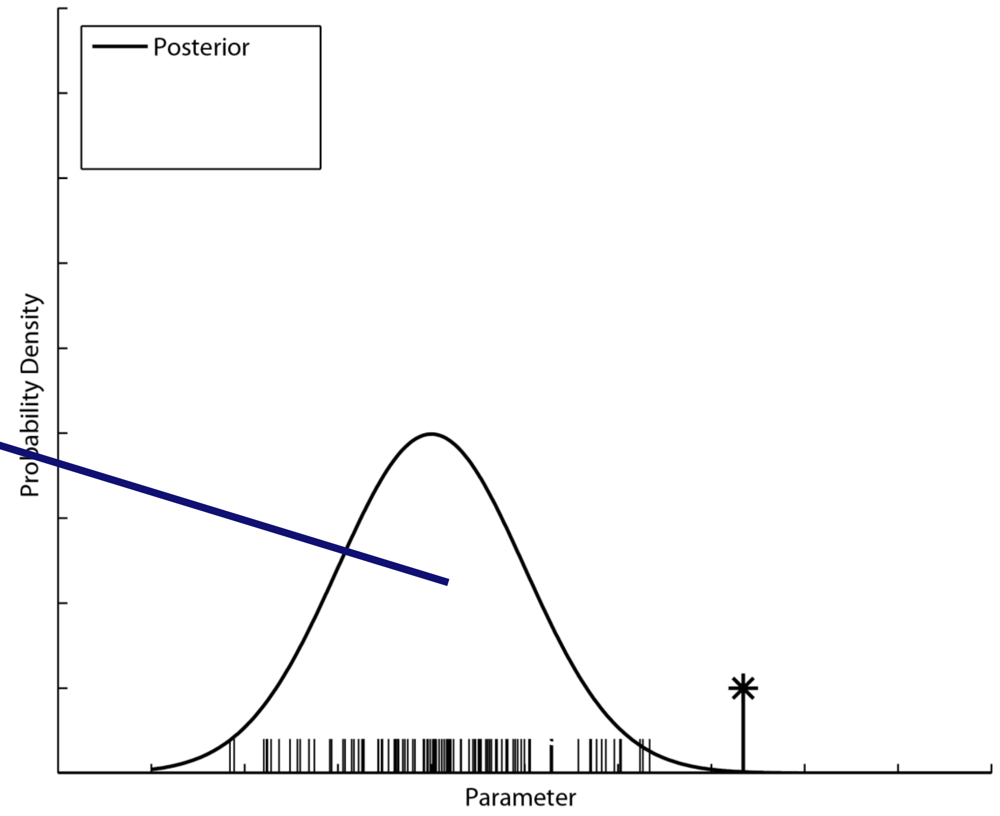
Solution:
Use importance sampling to compute the variance reduction

Posterior Predictive Distribution

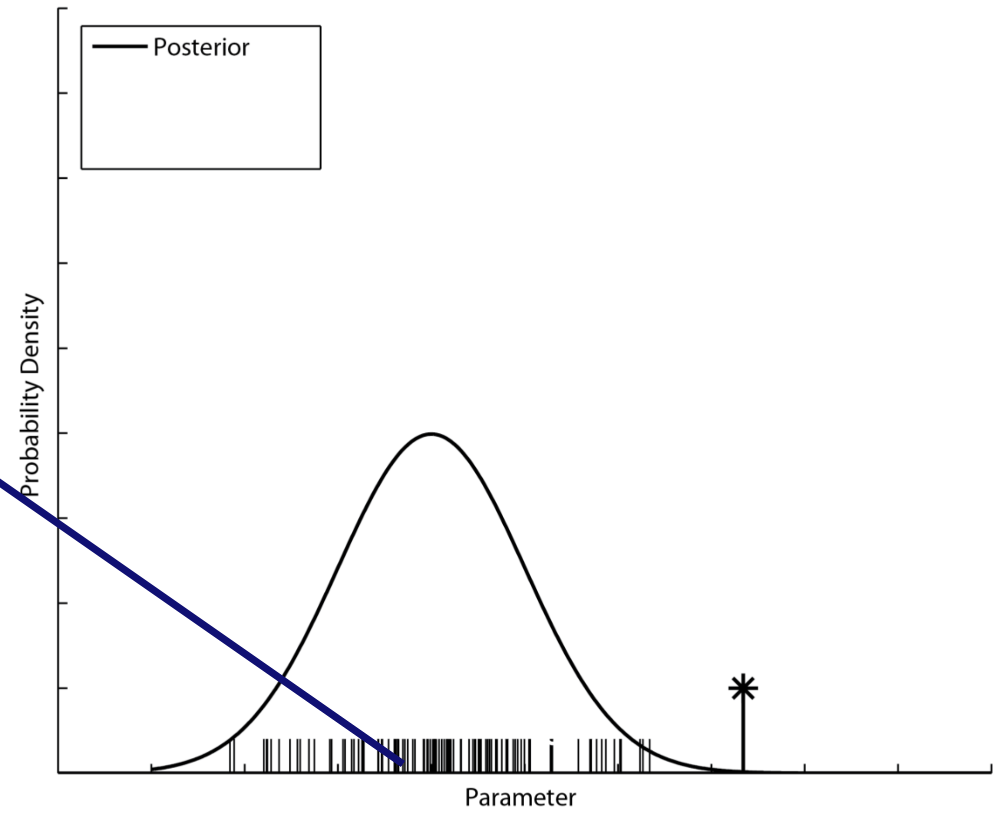
Solution:
Use importance sampling to compute the variance reduction

$$E[z \mid y, y_n] = \int p_{old}(\vec{\theta}) \frac{p_{updated}(\vec{\theta})}{p_{old}(\vec{\theta})} z(\vec{\theta}) d\theta_1 \cdots d\theta$$

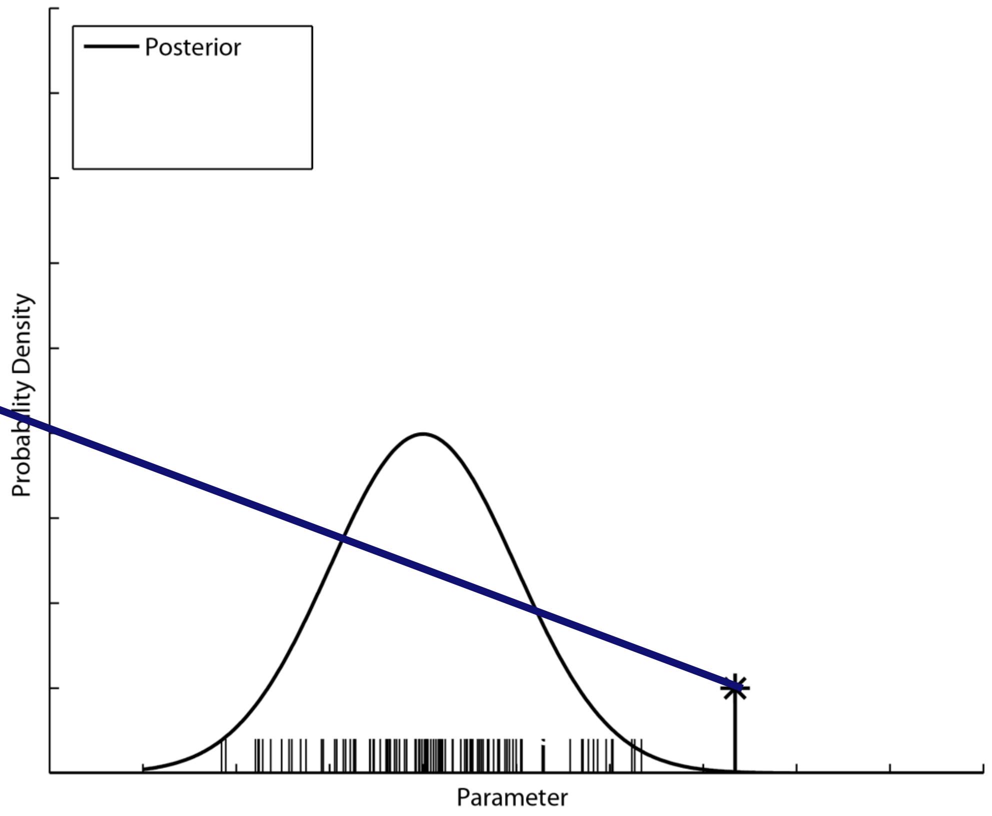
Posterior



Samples from Predictive Distribution

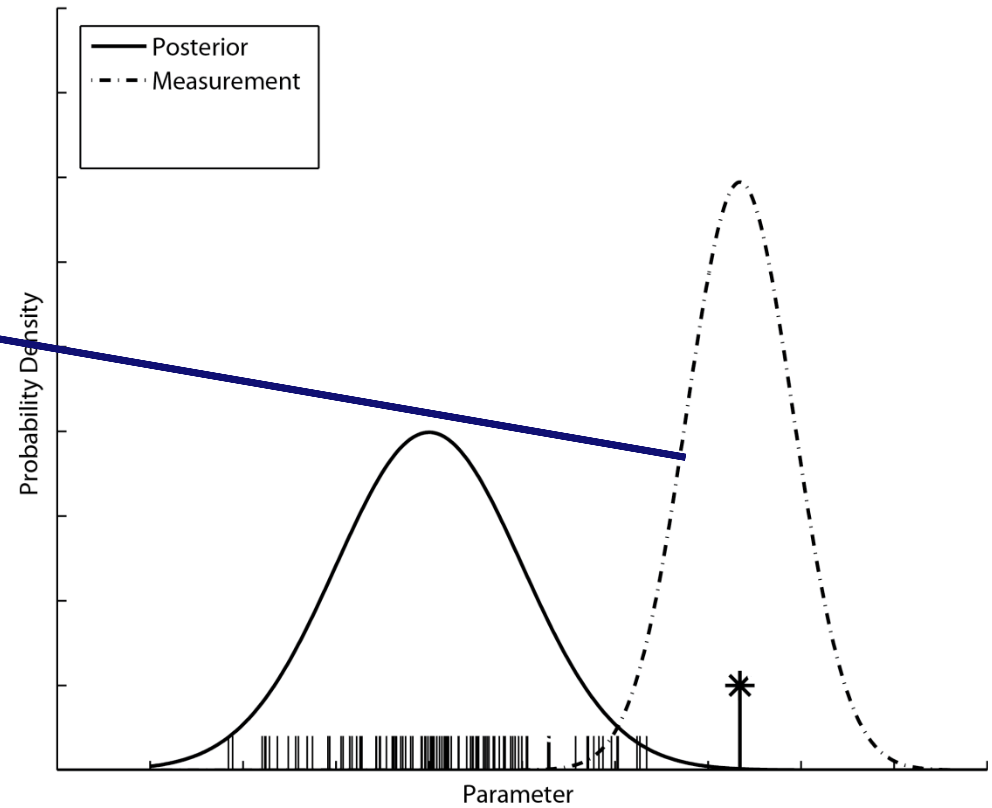


New Measurement



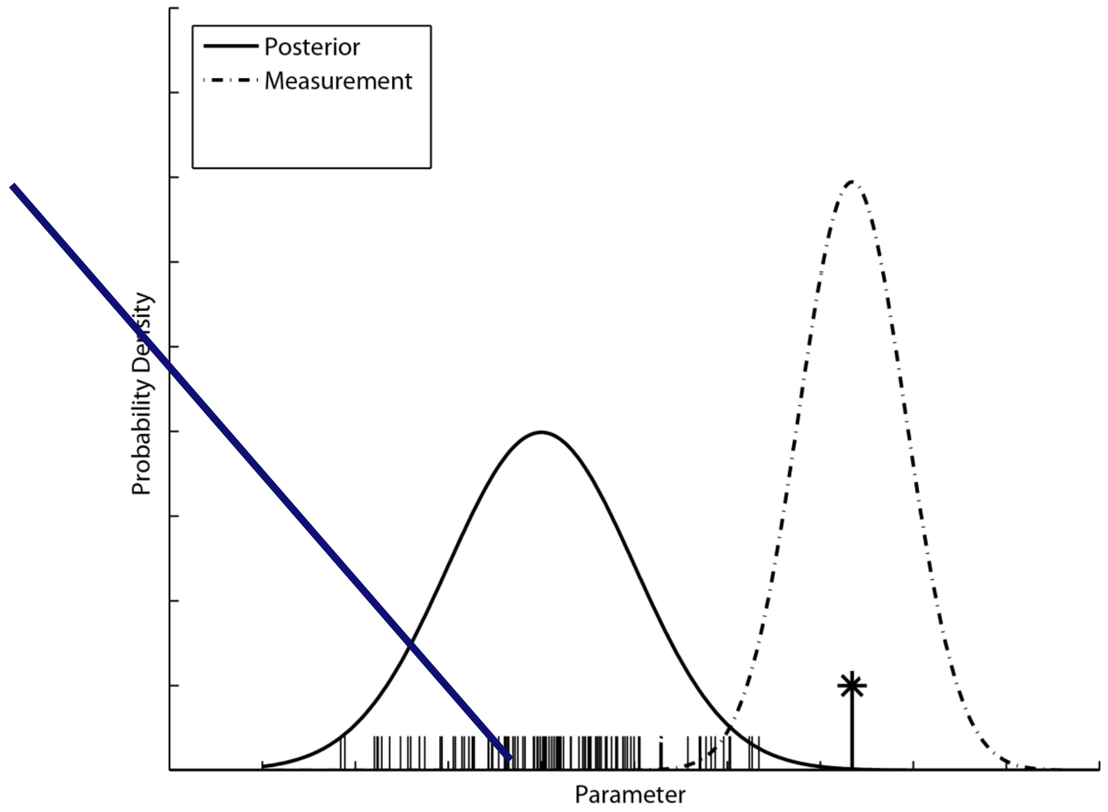
Measurement Error Model

$$p(y^N \mid \vec{\theta})$$



Samples are weighted according to the ratio of the old and new posterior

$$\frac{p_{updated}(\vec{\theta})}{p_{old}(\vec{\theta})}$$



$$p_{old}(\vec{\theta}) = \frac{p(y^D \mid \vec{\theta})\, p(\vec{\theta})}{Z_{old}}$$

$$p_{updated}(\vec{\theta}) = \frac{p(y^D \mid \vec{\theta})\, p(y^N \mid \vec{\theta})\, p(\vec{\theta})}{Z_{updated}}$$

$$\frac{p_{updated}(\vec{\theta})}{p_{old}(\vec{\theta})} = \frac{Z_{old}}{Z_{updated}}\, p(y^N \mid \vec{\theta})$$

$$E[z \mid y, y_n] = \int p_{old}(\vec{\theta}) \frac{Z_{old}}{Z_{updated}} p(y^N \mid \vec{\theta}) z(\vec{\theta}) d\theta_1 \cdots d\theta_n$$

Technische Universiteit Eindhoven University of Technology

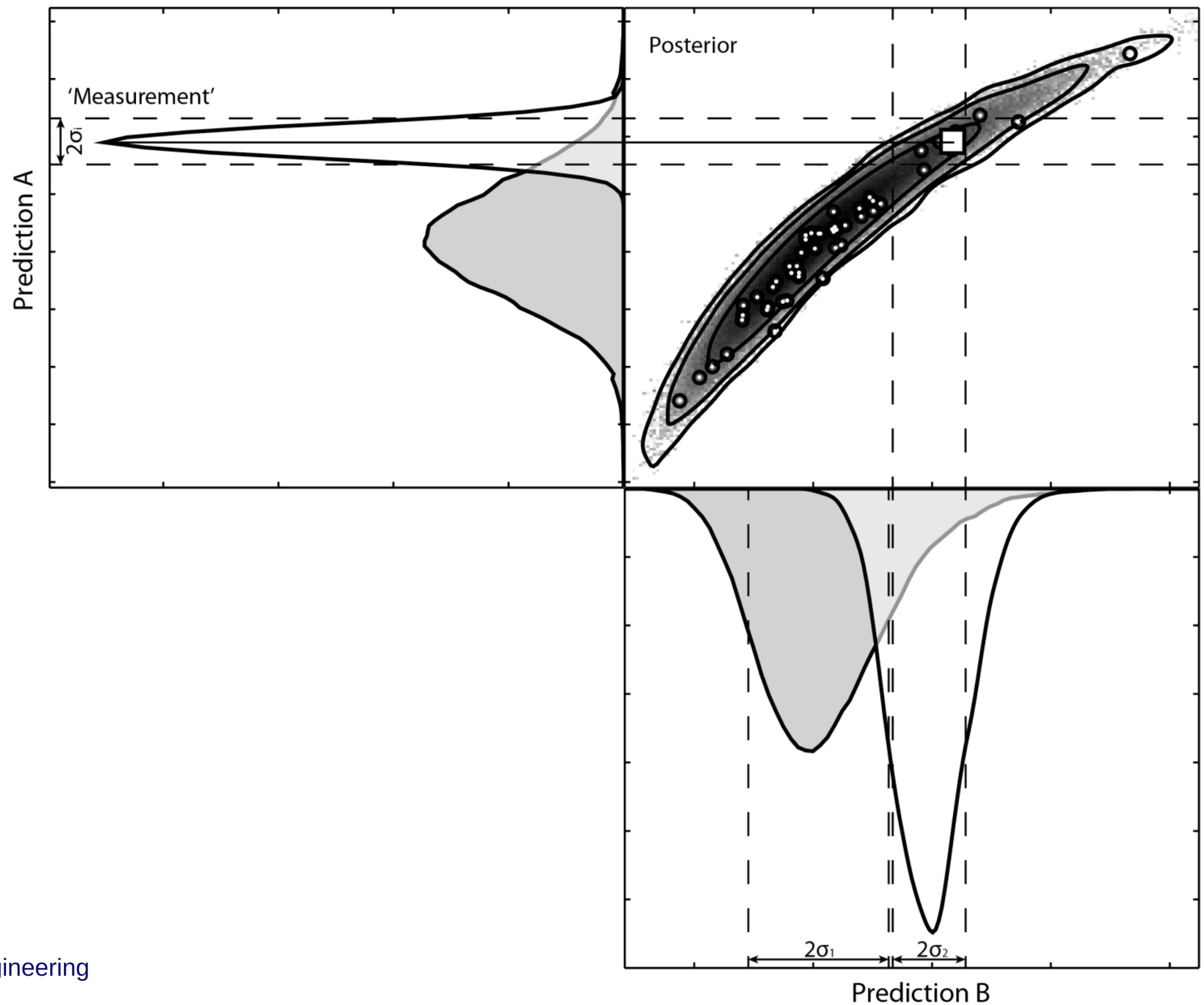
$$E[z \mid y, y_n] \approx \sum_{k=1}^{N} \frac{Z_{old}}{Z_{updated}} p(y^N \mid \vec{\theta}_k) z(\vec{\theta}_k) \qquad \vec{\theta}_k \sim p_{old}(\vec{\theta})$$

$$E[z \mid y, y_n] \approx \sum_{k=1}^{N} \frac{p(y^N \mid \vec{\theta}_k)}{\sum_{t=1}^{N} p(y^N \mid \vec{\theta}_t)} z(\vec{\theta}_k) \qquad \vec{\theta}_k \sim p_{old}(\vec{\theta})$$

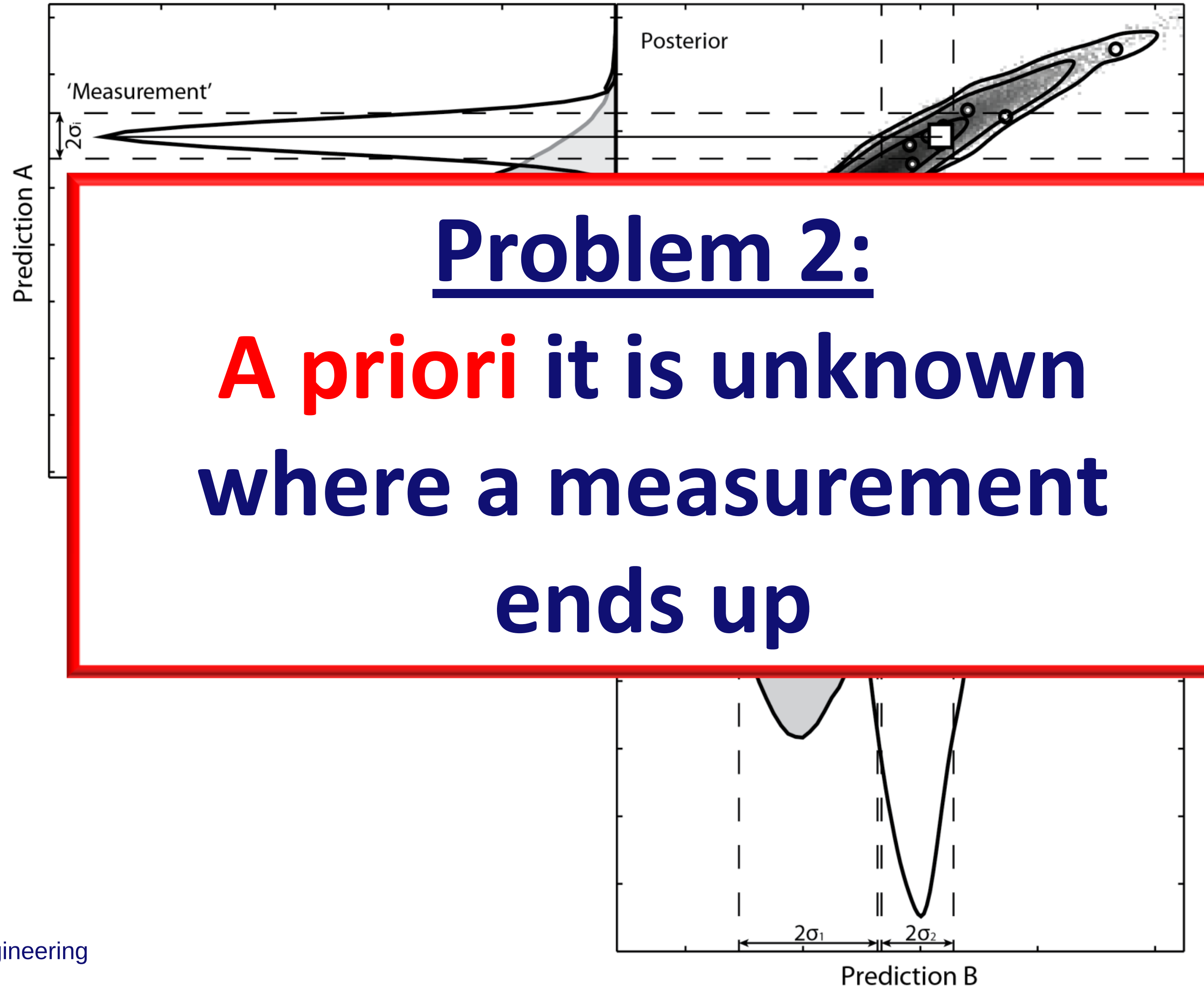
Replace the ratio by its estimate

Technische Universiteit
Eindhoven
University of Technology

Posterior Predictive Distribution



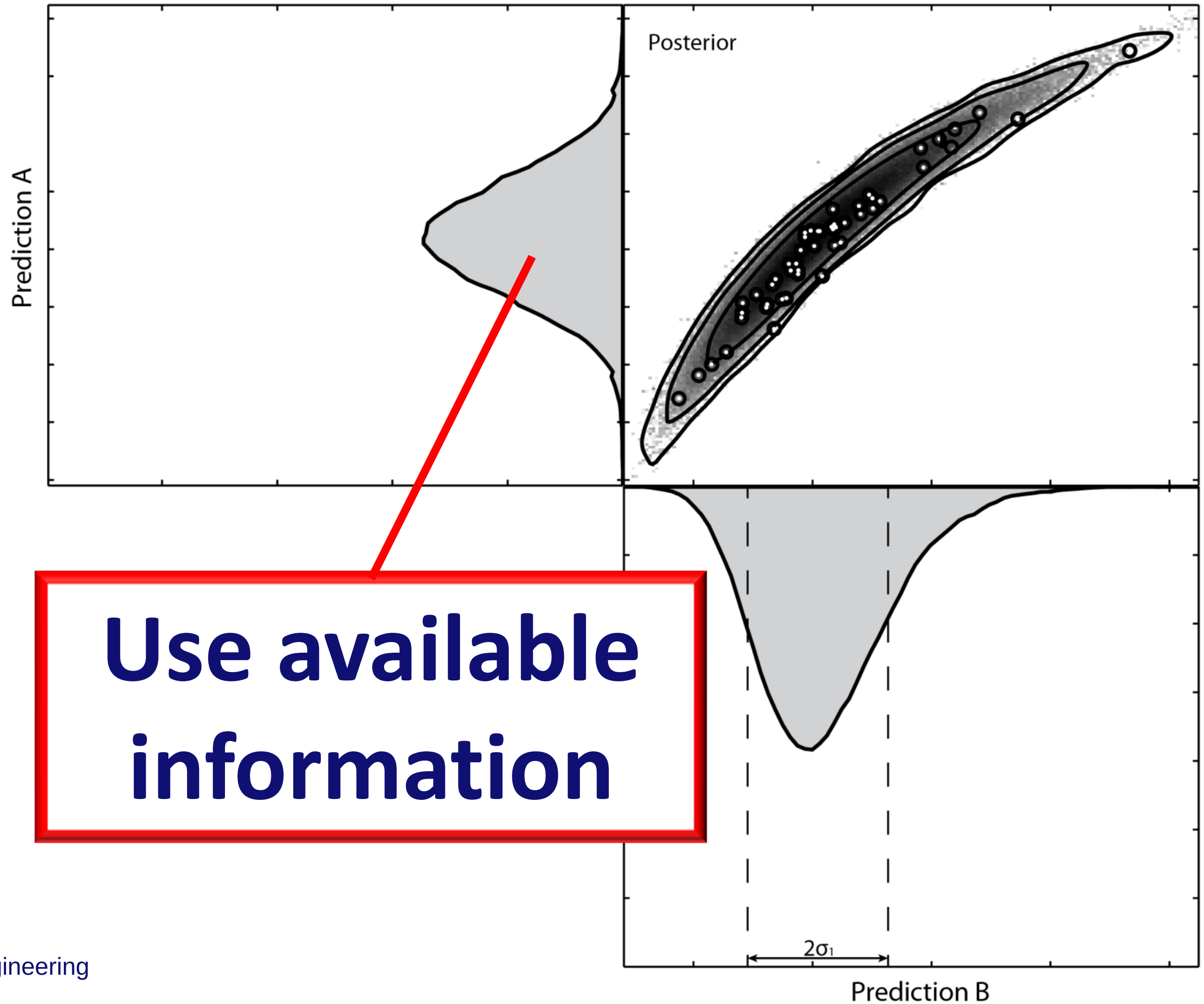
Posterior Predictive Distribution



Problem 2:

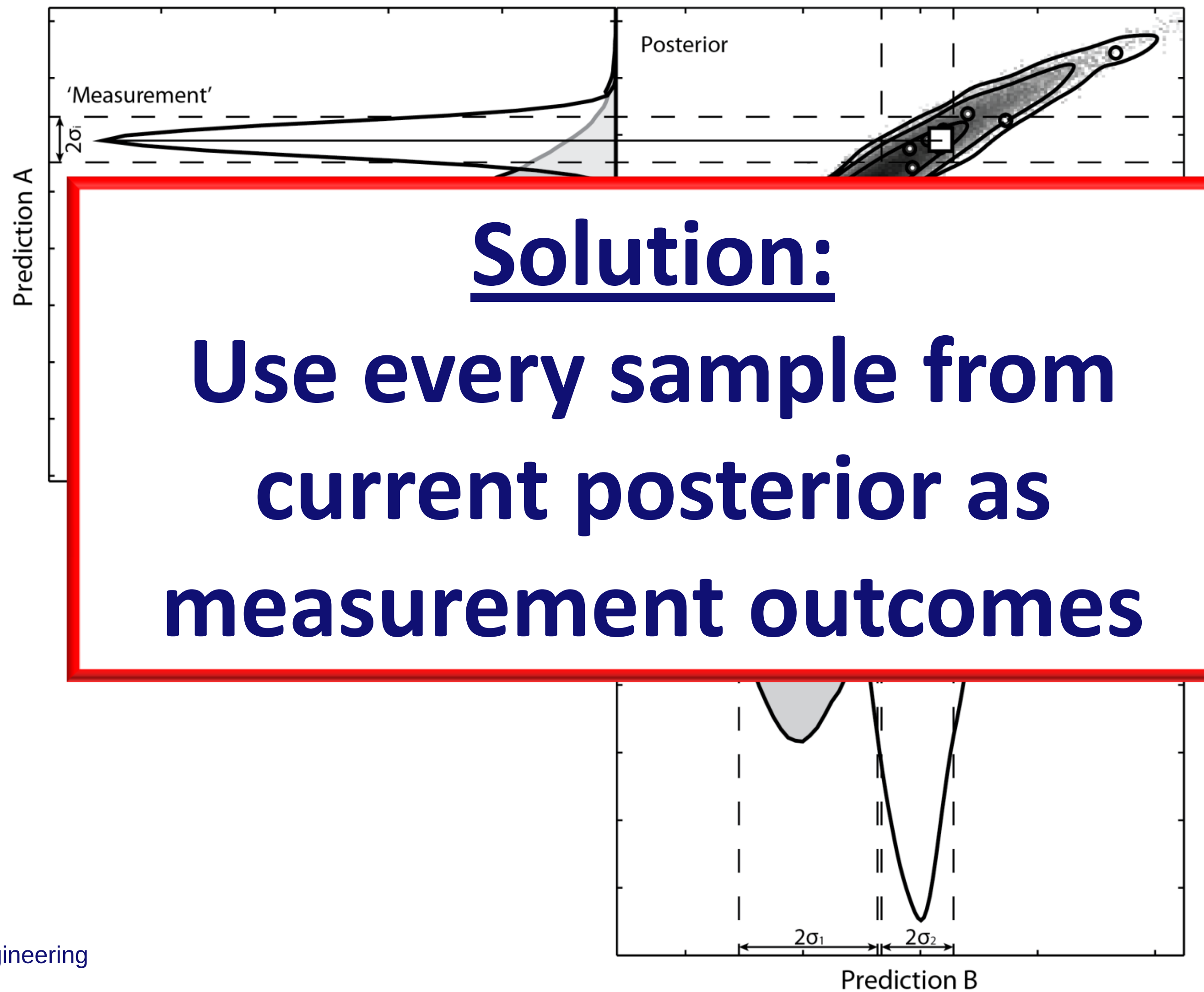
A priori it is unknown where a measurement ends up

Posterior Predictive Distribution



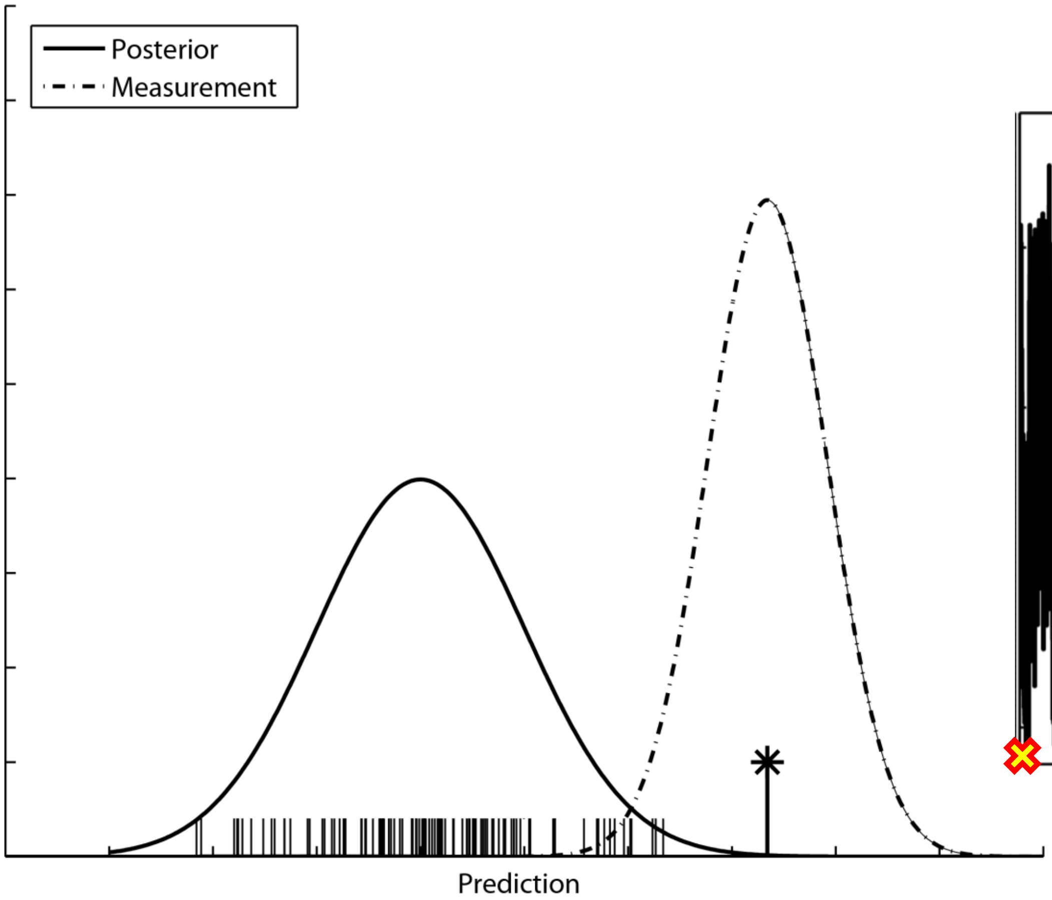
Use available information

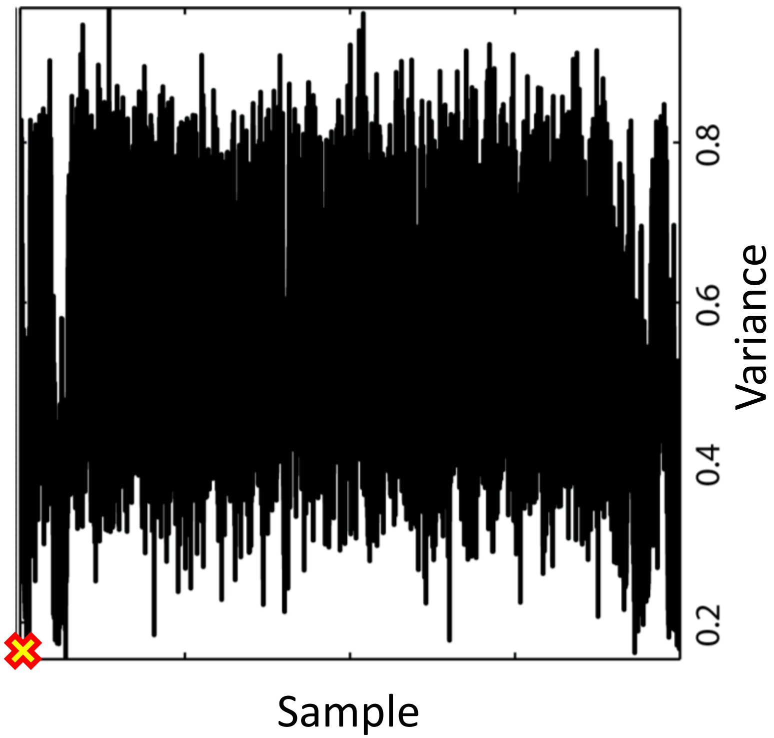
Posterior Predictive Distribution

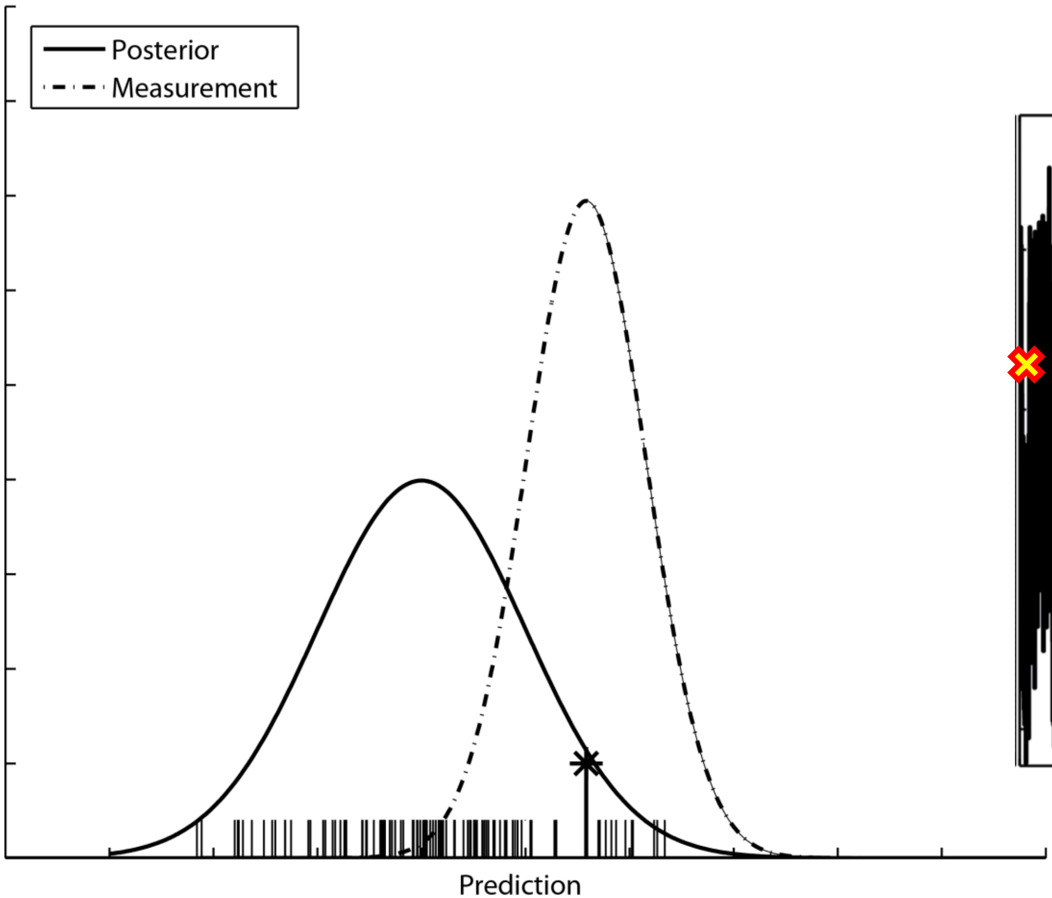


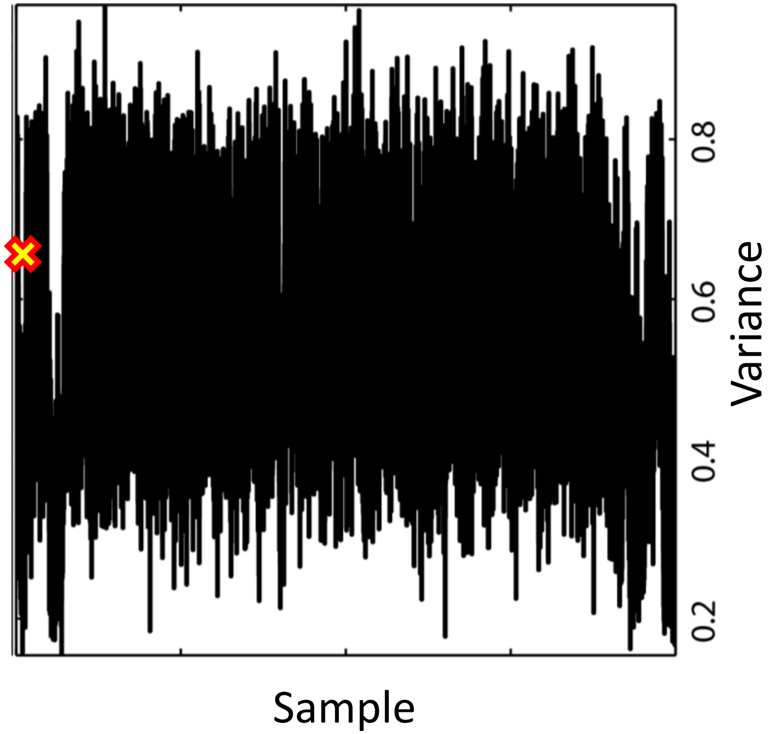
<u>Solution:</u>

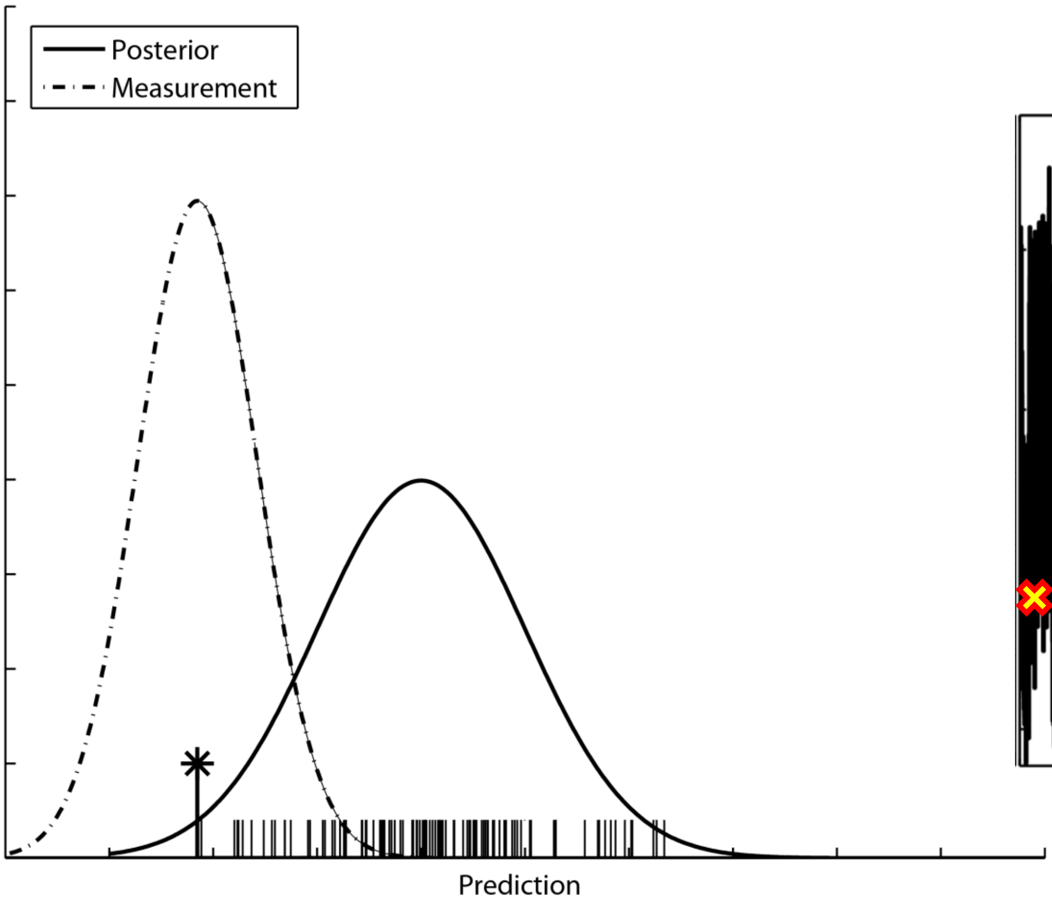
Use every sample from current posterior as measurement outcomes

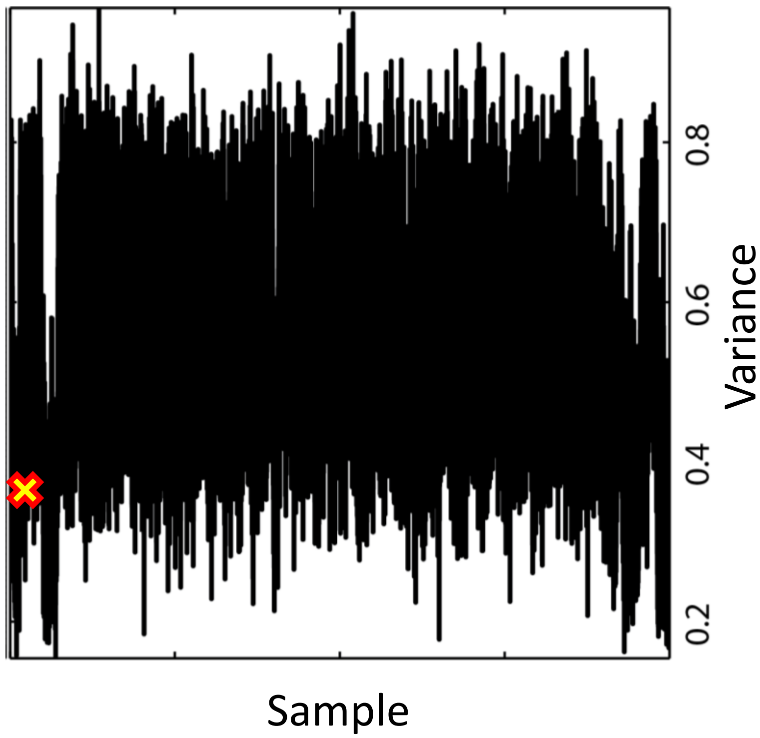

Posterior
Measurement
Prediction

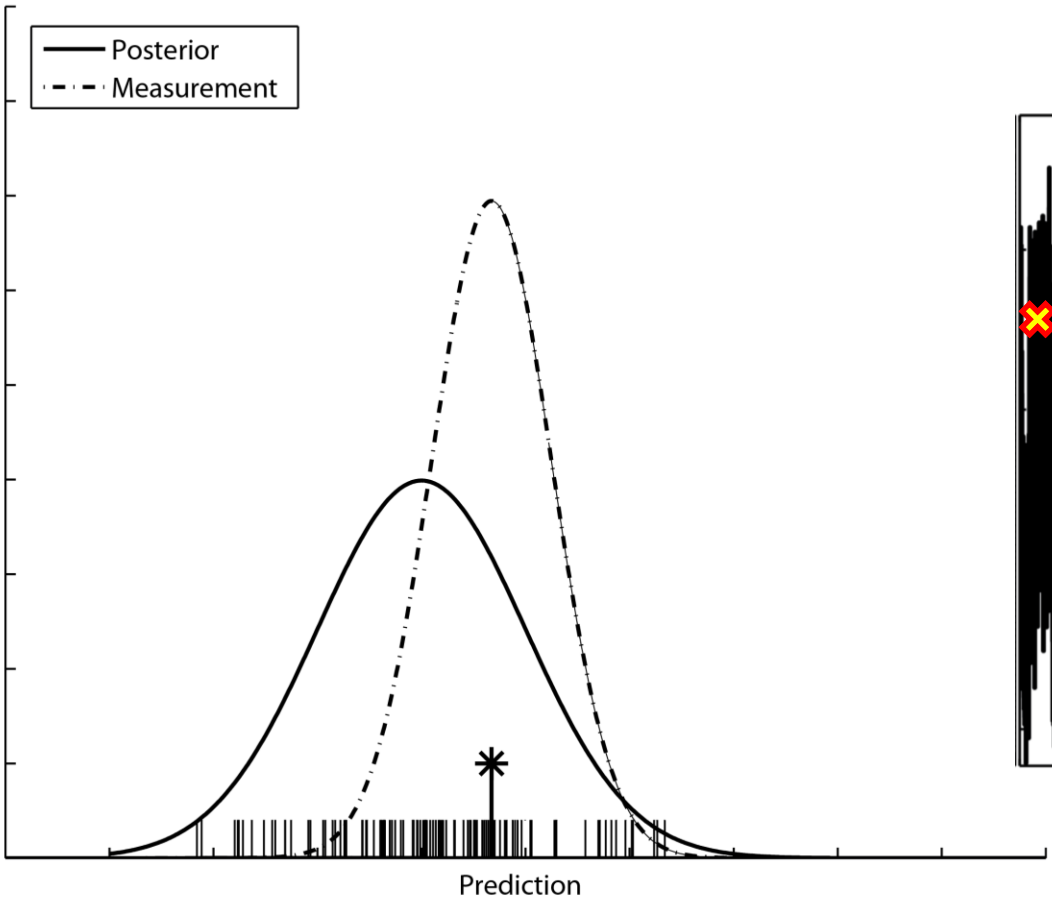

0.2
0.4
0.6
0.8
Variance
Sample

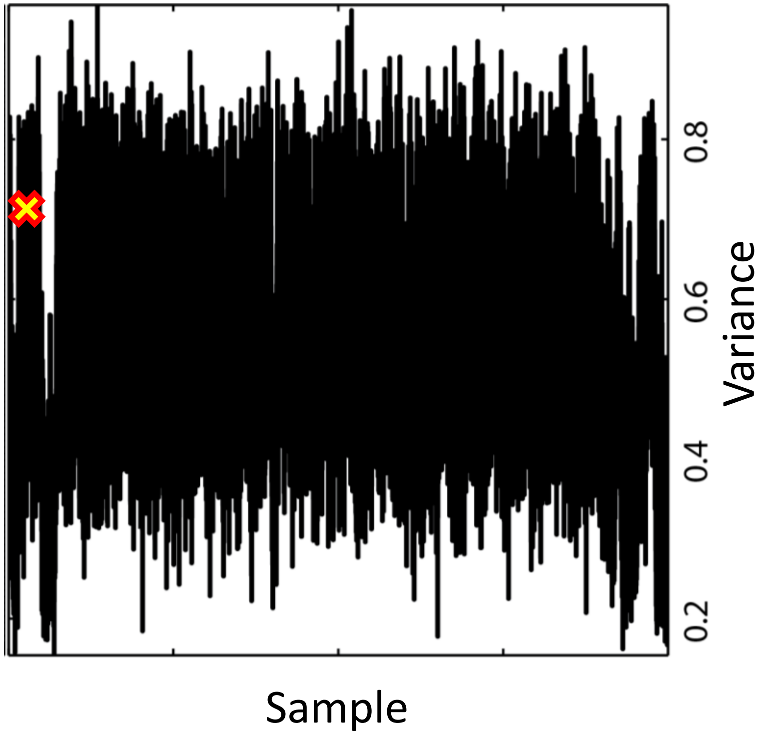

Posterior
Measurement
Prediction

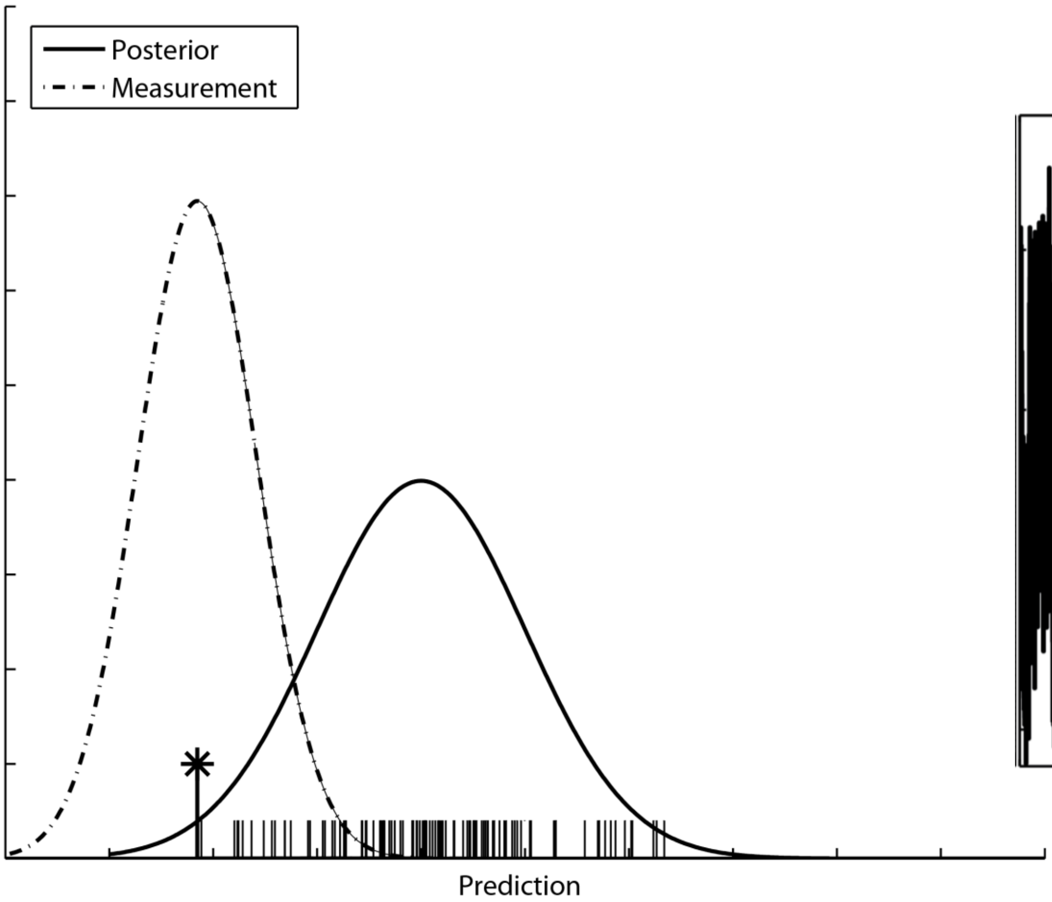

0.2
0.4
0.6
0.8
Variance
Sample

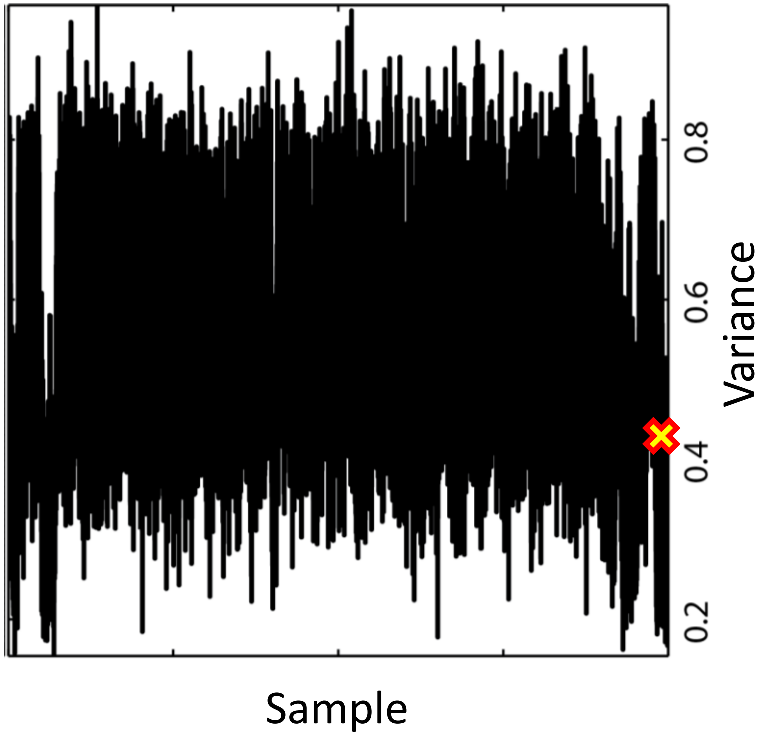

Posterior
Measurement
Prediction

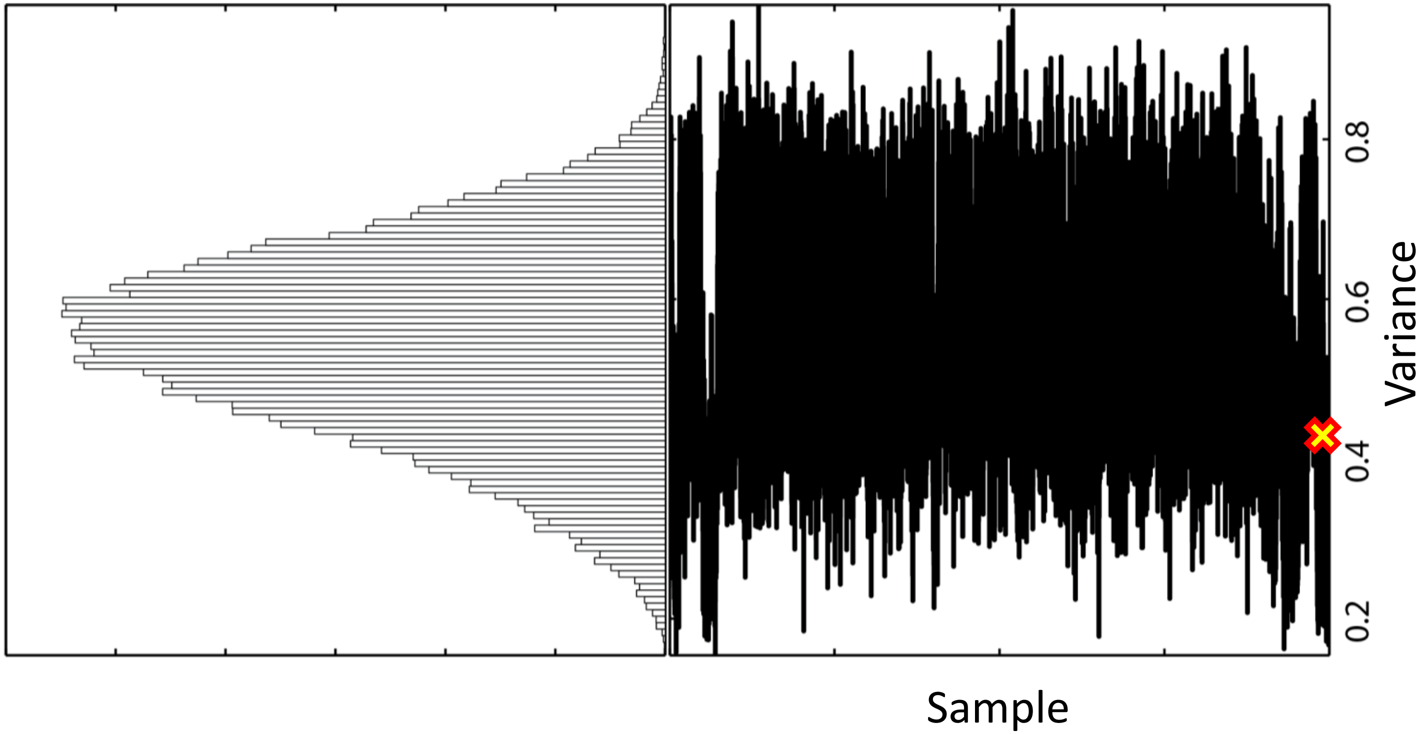

0.2
0.4
0.6
0.8
Variance
Sample

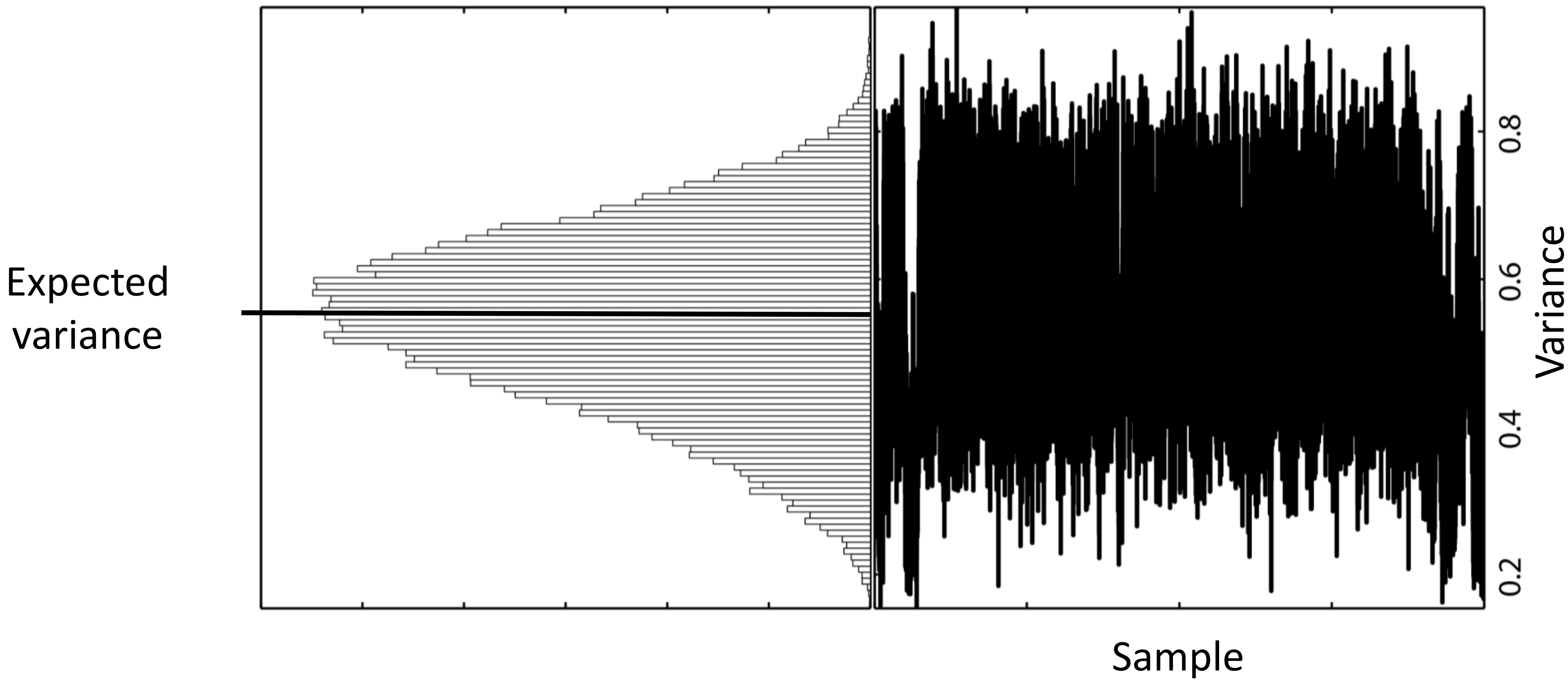

Posterior
Measurement
Prediction


0.2
0.4
0.6
0.8
Variance
Sample


Posterior
Measurement
Prediction


0.2
0.4
0.6
0.8
Variance
Sample


0.2
0.4
0.6
0.8
Variance
Sample


Expected variance
0.8
0.6
0.4
0.2
Variance
Sample

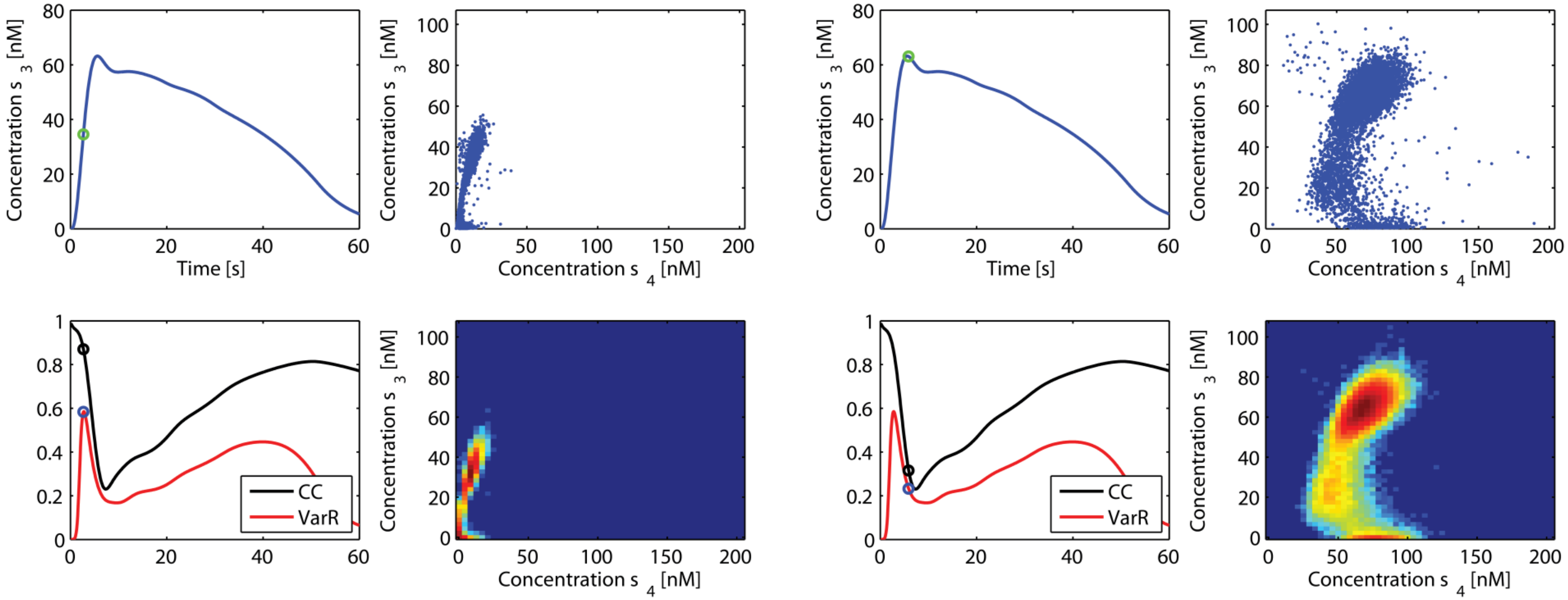
$$E[z \mid y, y_n] \approx \sum_{k=1}^{N} \frac{p(y^N \mid \vec{\theta}_k)}{\sum_{t=1}^{N} p(y^N \mid \vec{\theta}_t)} z(\vec{\theta}_k) \qquad \vec{\theta}_k \sim p_{old}(\vec{\theta})$$

$$\frac{1}{N}\sum_{j=1}^{N}\sum_{k=1}^{N}\frac{p(y(\vec{\theta}_j)\,|\,\vec{\theta}_k)}{\sum_{t=1}^{N}p(y^N\,|\,\vec{\theta}_t)}\,z(\vec{\theta}_k) \qquad \vec{\theta}_k,\vec{\theta}_j \sim p_{old}(\vec{\theta})$$

$$\frac{1}{N}\sum_{j=1}^{N}\sum_{k=1}^{N}\frac{p(y(\vec{\theta}_j)\,|\,\vec{\theta}_k)}{\sum_{t=1}^{N} p(y^N\,|\,\vec{\theta}_t)}\,z(\vec{\theta}_k) \qquad \vec{\theta}_k,\vec{\theta}_j \sim p_{old}(\vec{\theta})$$

Only requires samples from current posterior!
Easily implemented on GPU

Example



Sample from 'experiment space'

Now that we are able to predict effectiveness of experiment.

Trivial to include multiple measurements in the importance sampler

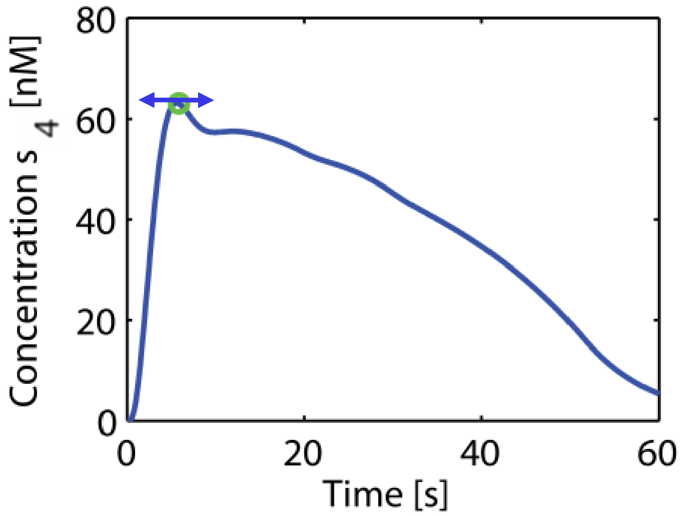
Sample from the space of all possible combinations of experiments and determine the best one.

Ability to reject infeasible experiments.

Technische Universiteit Eindhoven University of Technology

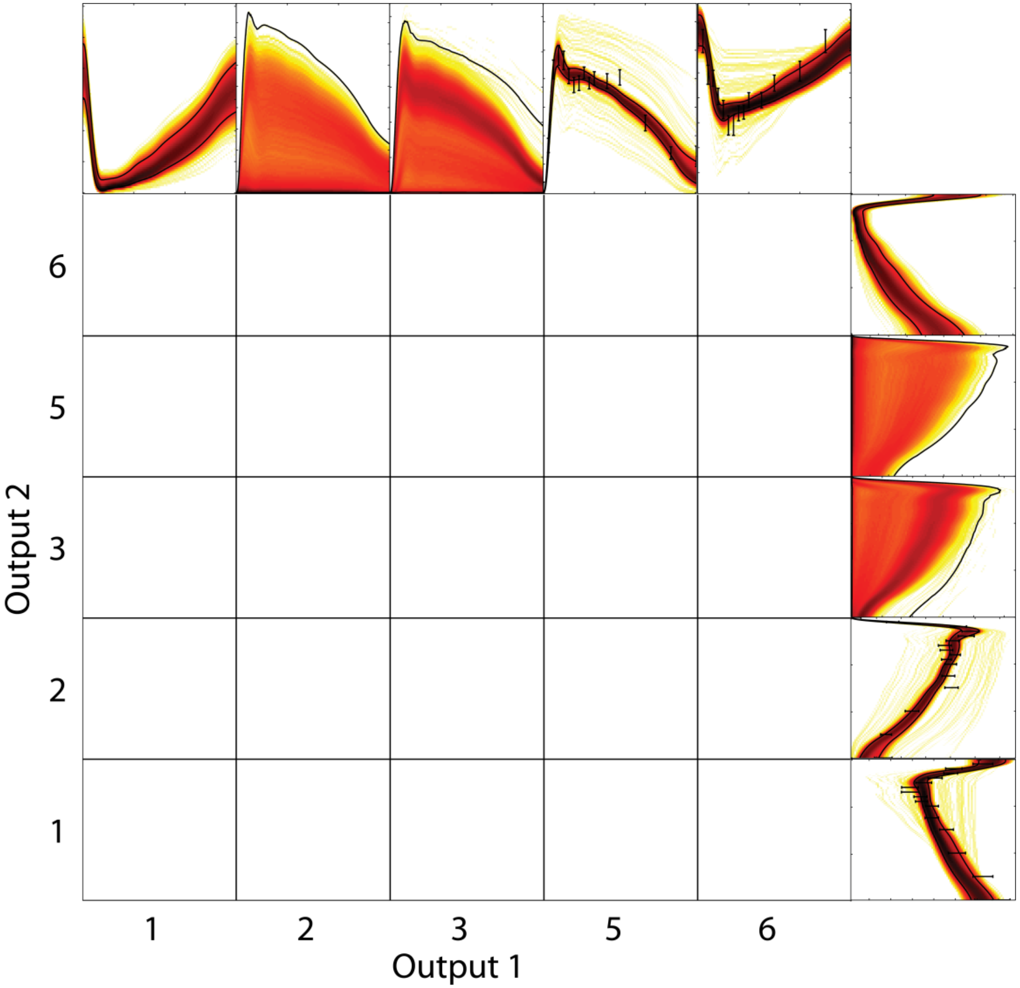
Objective:

- Reduce uncertainty of peak time nucleus

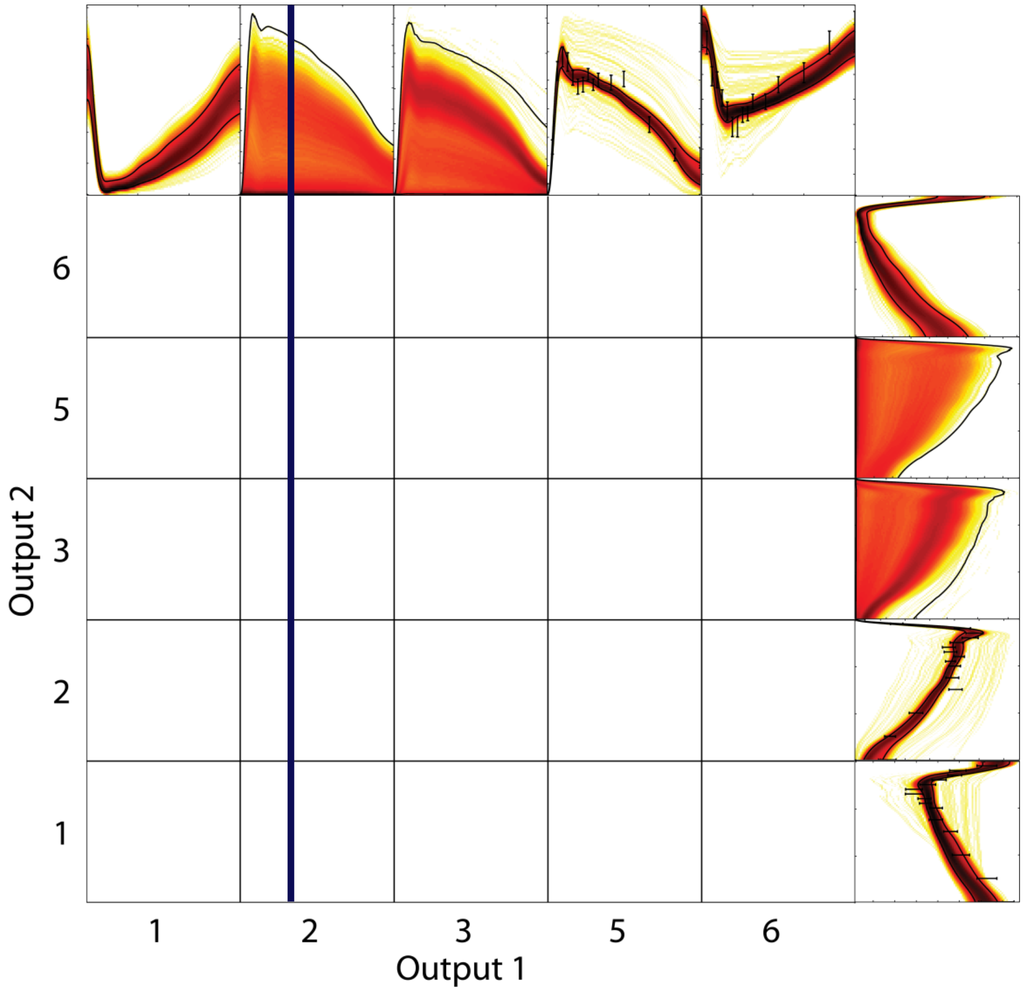


Technische Universiteit
Eindhoven
University of Technology

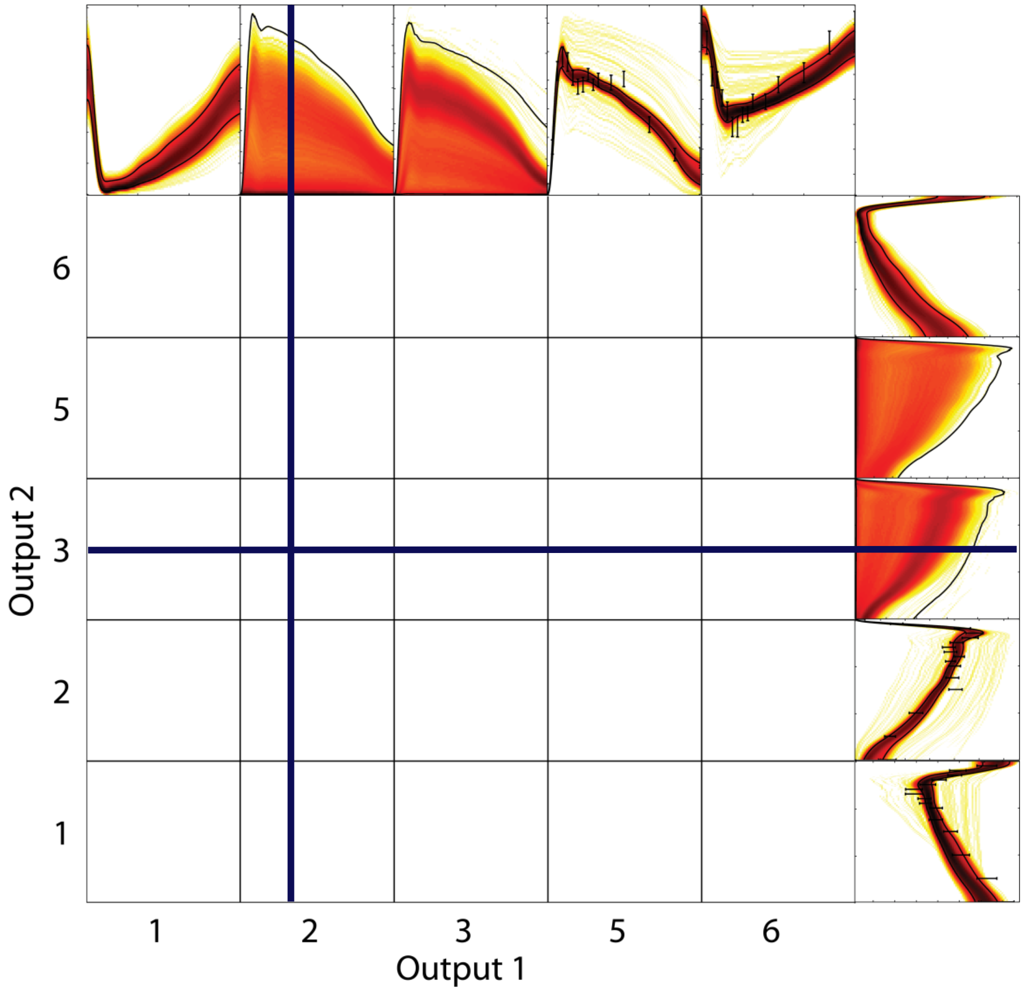
Example: Two experiments



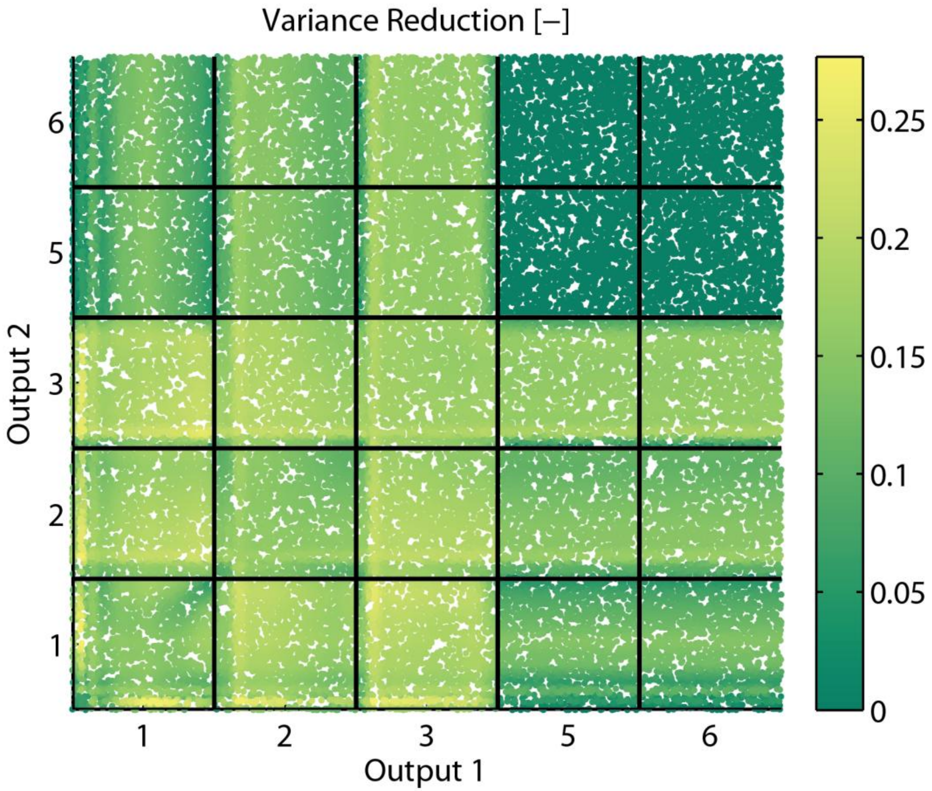
Example: Two experiments



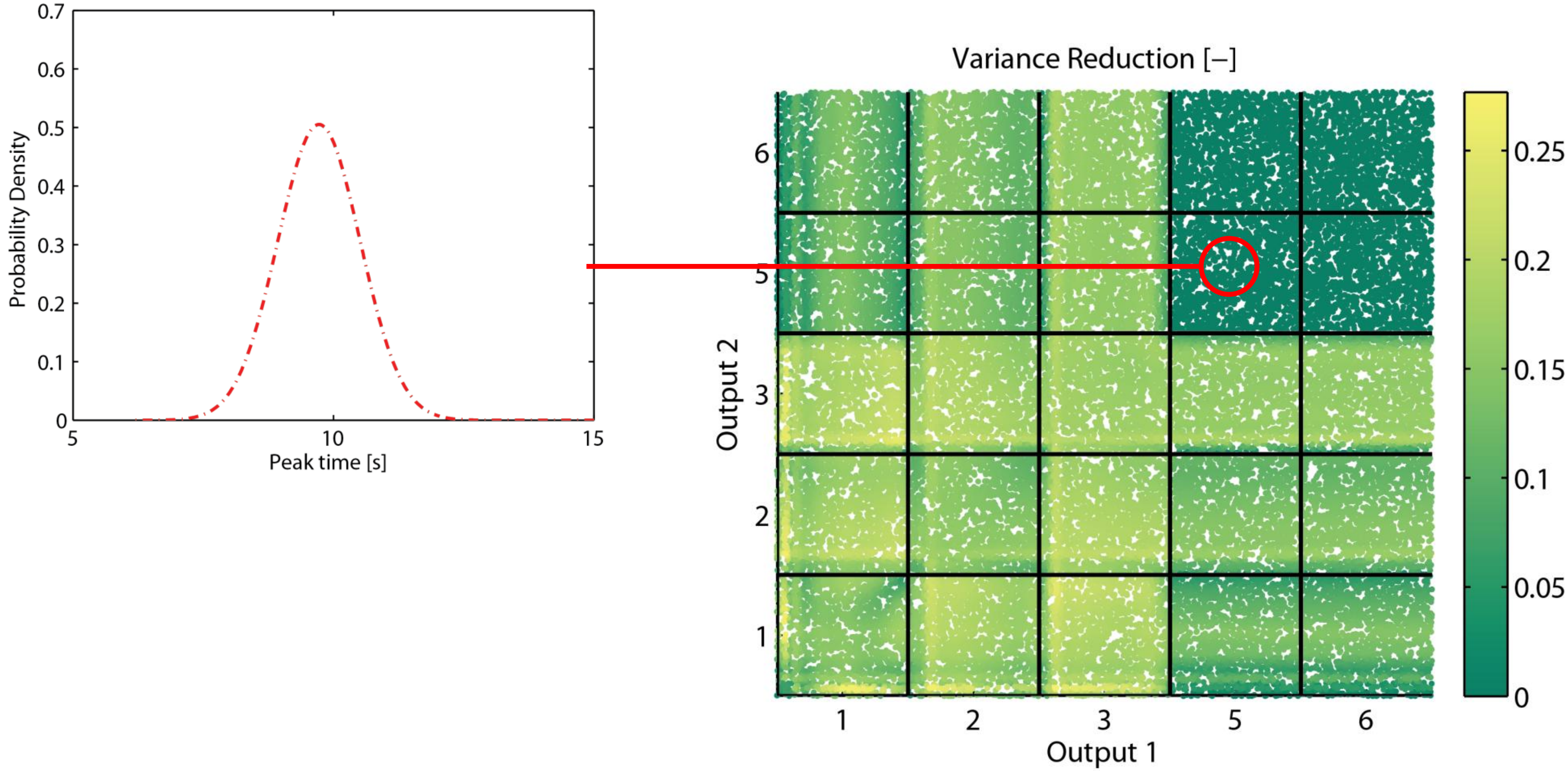
Example: Two experiments



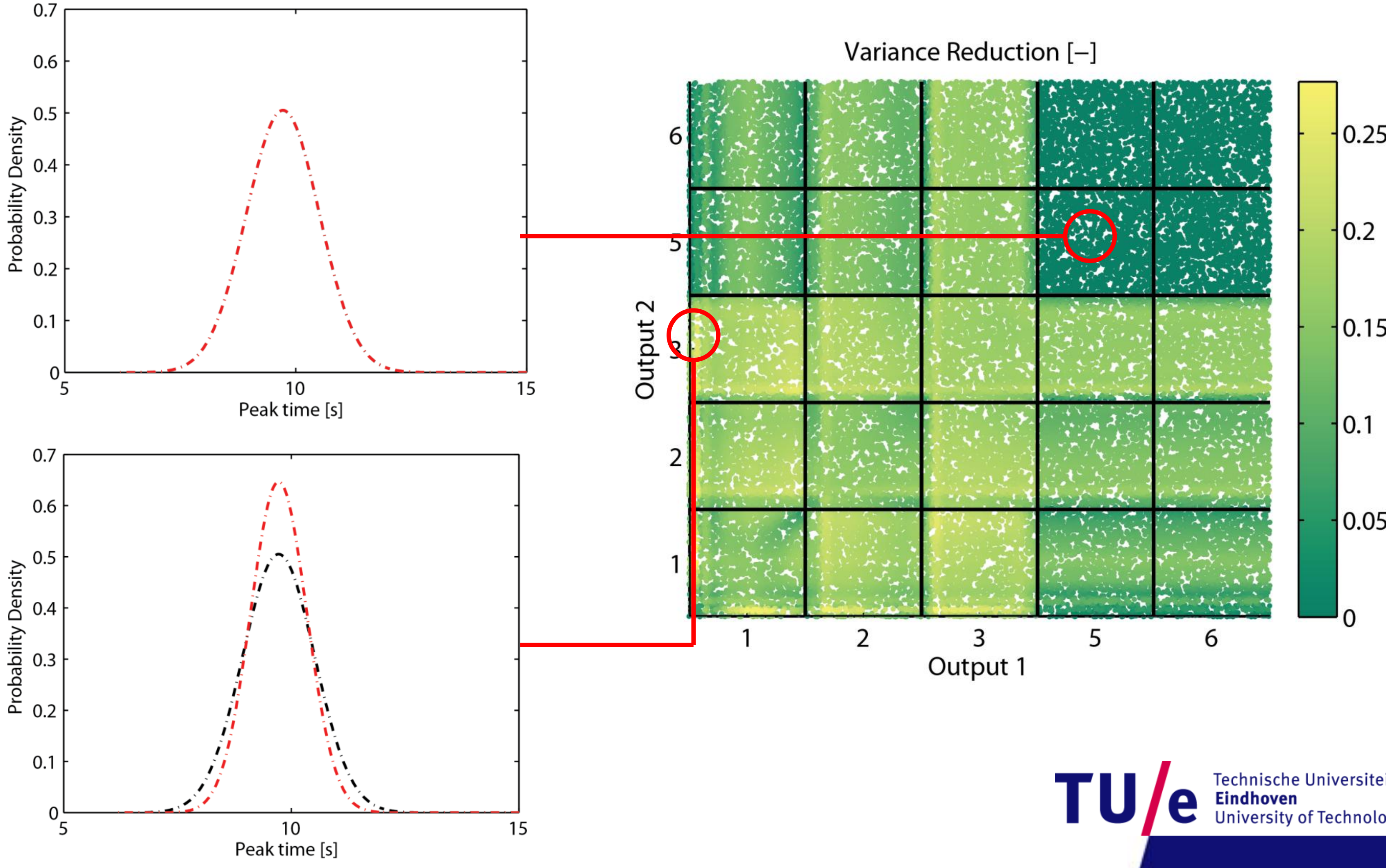
Average variance reduction



Average variance reduction



Average variance reduction



Conclusions

- **Optimal Experiment Design at little extra cost**
  - **Fits well with existing MCMC toolchain**
  - **Faster linearised version also available**
  - **GPU (OpenCL) implementation available**
- **Multiple experiments at once (Combinatorial)**
- **Flexibility of observer and target choice**
- **Flexibility of error models**

Vanlier et al, *Bioinformatics (2012)*

Systems Biology group TU/e

Prediction Uncertainty Analysis (PUA)

by Joep Vanlier and Natal van Riel

- CVode Wrapper: ODE MEX v6 GNU (zip file)
- Prediction Uncertainty Analysis: PUA v8 GNU (zip file)
- Optimal Experimental Design: OED v2 GNU (zip file)

CVode Wrapper

The CVode package can generate compiled MEX files for the simulation of Ordinary Differential Equation (ODE) models, composed of systems of coupled 1st order ODEs.

The CVode Wrapper package for Matlab includes:

- a parser to convert a Matlab m-file containing the ODE's to a C-file
- compile the C source file and the numerical integrators from the SUNDIALS CVode package into a MEX file that can be run in Matlab
- installation and usage instructions

See also 'Speeding up simulations of ODE models in Matlab using CVode and MEX files'.

Prediction Uncertainty Analysis

The Prediction Uncertainty Analysis package for Matlab includes:

- our Markov Chain Monte Carlo (MCMC) sampler for Bayesian parameter estimation
- Profile Likelihood method for indentifiability analysis
- an example: the JAK-STAT pathway model

To reproduce the results from the paper '*An Integrated Strategy for Prediction Uncertainty Analysis*' (Bioinformatics, in press) the software is also available as win32 executables:

- PUA Compiled (zip file, 183 MB)

To run these, one first needs to install the MATLAB Runtime Environment (also provided in the zip file). Please note that a reboot is required after installation. The different steps from the paper can be reproduced by consecutively running the following files:

- JAKLSS
- JAKLSS_2
- JAKPL
- JAKMCMC_log

http://bmi.bmt.tue.nl/sysbio/software/pua.html

Systems Biology group TU/e

home

Prediction Uncertainty Analysis (PUA)

by Joep Vanlier and Natal van Riel

- CVode Wrapper: ODE MEX v6 GNU (zip file)
- Prediction Uncertainty Analysis: PUA v8 GNU (zip file)
- Optimal Experimental Design: OED v2 GNU (zip file)

Software can be used under GNU General Public License. See http://www.gnu.org/copyleft/gpl for terms and conditions.

CVode Wrapper

The CVode package can generate compiled MEX files for the simulation of Ordinary Differential Equation (ODE) models, composed of systems of coupled 1st order ODEs.

The CVode Wrapper package for Matlab includes:

- a parser to convert a Matlab m-file containing the ODE's to a C-file
- compile [illegible] package into a MEX file that can be run in Matlab
- installation and usage instructions

See also 'Speeding up simulations of ODE models in Matlab using CVode and MEX files'.

Prediction Uncertainty Analysis

The Prediction Uncertainty Analysis package for Matlab includes:

- our Markov Chain Monte Carlo (MCMC) sampler for Bayesian parameter estimation
- Profile Likelihood method for indentifiability analysis
- an example: the JAK-STAT pathway model

To reproduce the results from the paper '*An Integrated Strategy for Prediction Uncertainty Analysis*' (Bioinformatics, in press) the software is also available as win32 executables:

- PUA Compiled (zip file, 183 MB)

To run these, one first needs to install the MATLAB Runtime Environment (also provided in the zip file). Please note that a reboot is required after installation. The different steps from the paper can be reproduced by consecutively running the following files:

- JAKLSS
- JAKLSS_2
- JAKPL
- JAKMCMC_log

Future/Ongoing work

- Include temporal uncertainties
- Apply methods to Model Selection (thermodynamic integration)
- Improve sampling of experimental design space
  - Initial sweep using linear approximant
  - Sequential Monte Carlo Methods
  - MCMC/Optimisation in design space

Thank you

Technische Universiteit
Eindhoven
University of Technology

Acknowledgements

Christian Tiemann

Ceylan Colmecki

Haili Liu

Joep Schmitz

Robert Oosterhof

Huan Yang

Rik van Roekel

Natal van Riel

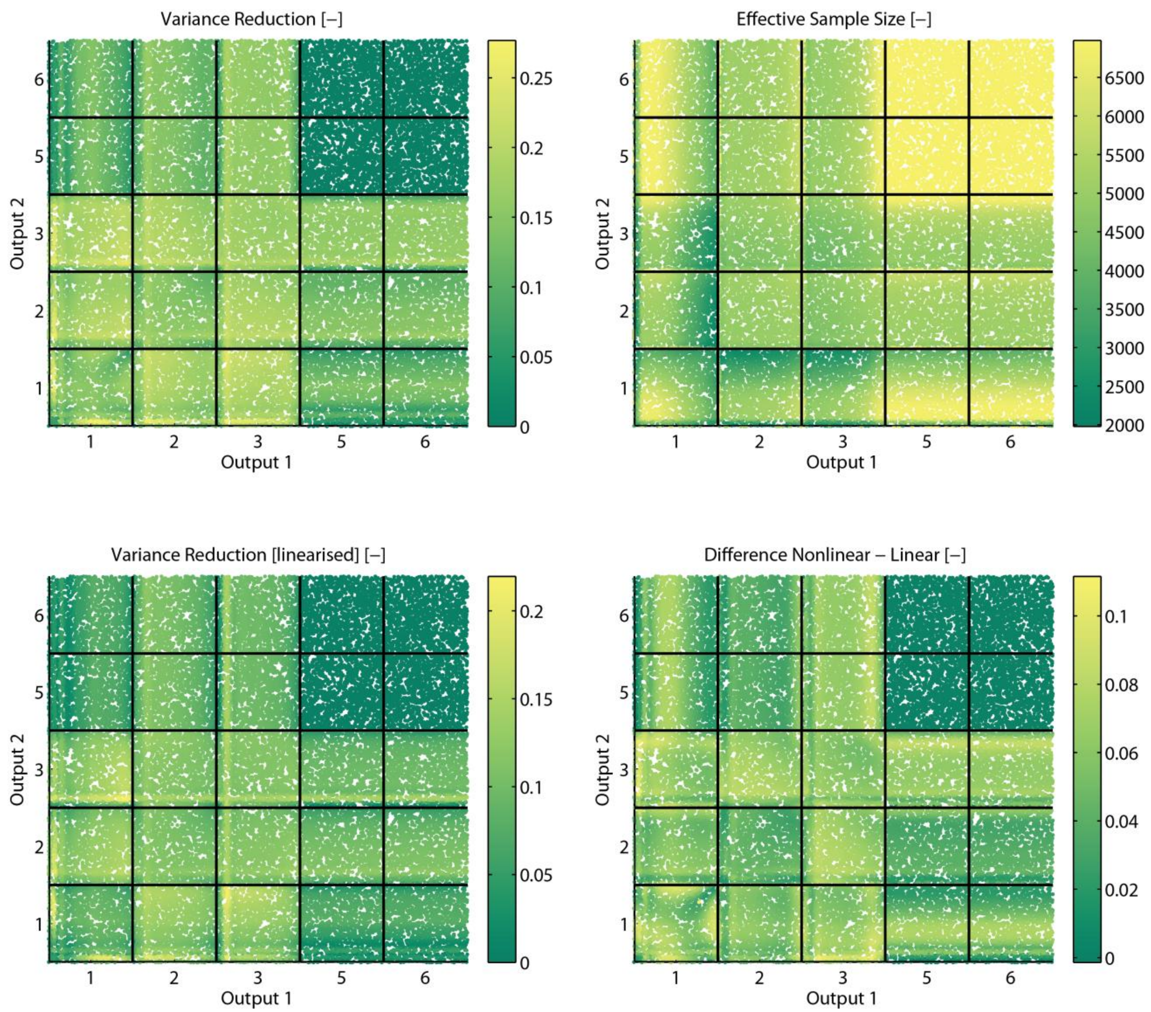
Peter Hilbers

Computational Issues

Option 1:
Linearisation of the PPD

Technische Universiteit Eindhoven University of Technology

Linearisation



Linearisation

- **MC for experimental design takes a long time. Even when using Importance Sampling.**

- **Sampling sparsity might lead to missing important experimental combinations.**

➔ For the computation of variances/variance reductions a speedup is possible!!

Linearisation

- **If the new models of the new measurements are Gaussians, then we can avoid resampling entirely.**
- **We can work directly with covariance matrices**
- **Use multidimensional Gaussian PDF's**

Linearisation

- **Step 1: Construct covariance matrix of output with potential measurements**

Compute covariance of output of interest and potential measurement (think ellipsoid)

$$\Sigma_{posterior} = \mathrm{cov}\left[\begin{pmatrix} o_1 & x_1^1 & \cdots & x_1^n \\ o_2 & x_2^1 & \cdots & x_2^n \\ \vdots & \vdots & \ddots & \vdots \\ o_m & x_m^1 & \cdots & x_m^n \end{pmatrix}\right]$$

Technische Universiteit Eindhoven University of Technology

Linearisation

- **Step 1: Construct covariance matrix of output with potential measurements**

- **Step 2: Compute new covariance after additional measurements (multiplication of two Gaussians)**

$$\Sigma_{new} = \left(\Sigma_{posterior}^{-1} + \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & \frac{1}{\sigma_1^2} & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & \frac{1}{\sigma_n^2} \end{pmatrix} \right)^{-1}$$

Linearisation

- **Step 1: Construct covariance matrix of output with potential measurements**

Compute covariance of output of interest and potential measurement (think ellipsoid)

$$\Sigma_{posterior} = \mathrm{cov}\left[\begin{pmatrix} o_1 & x_1^1 & \cdots & x_1^n \\ o_2 & x_2^1 & \cdots & x_2^n \\ \vdots & \vdots & \ddots & \vdots \\ o_m & x_m^1 & \cdots & x_m^n \end{pmatrix}\right]$$

Linearisation

- **Step 1: Construct covariance matrix of output with potential measurements**
- **Step 2: Compute new covariance after additional measurements (multiplication of two Gaussians)**

$$\Sigma_{new} = \left(\Sigma_{posterior}^{-1} + \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & \dfrac{1}{\sigma_1^{2}} & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & \dfrac{1}{\sigma_n^{2}} \end{pmatrix} \right)^{-1}$$

Technische Universiteit Eindhoven University of Technology

Linearisation

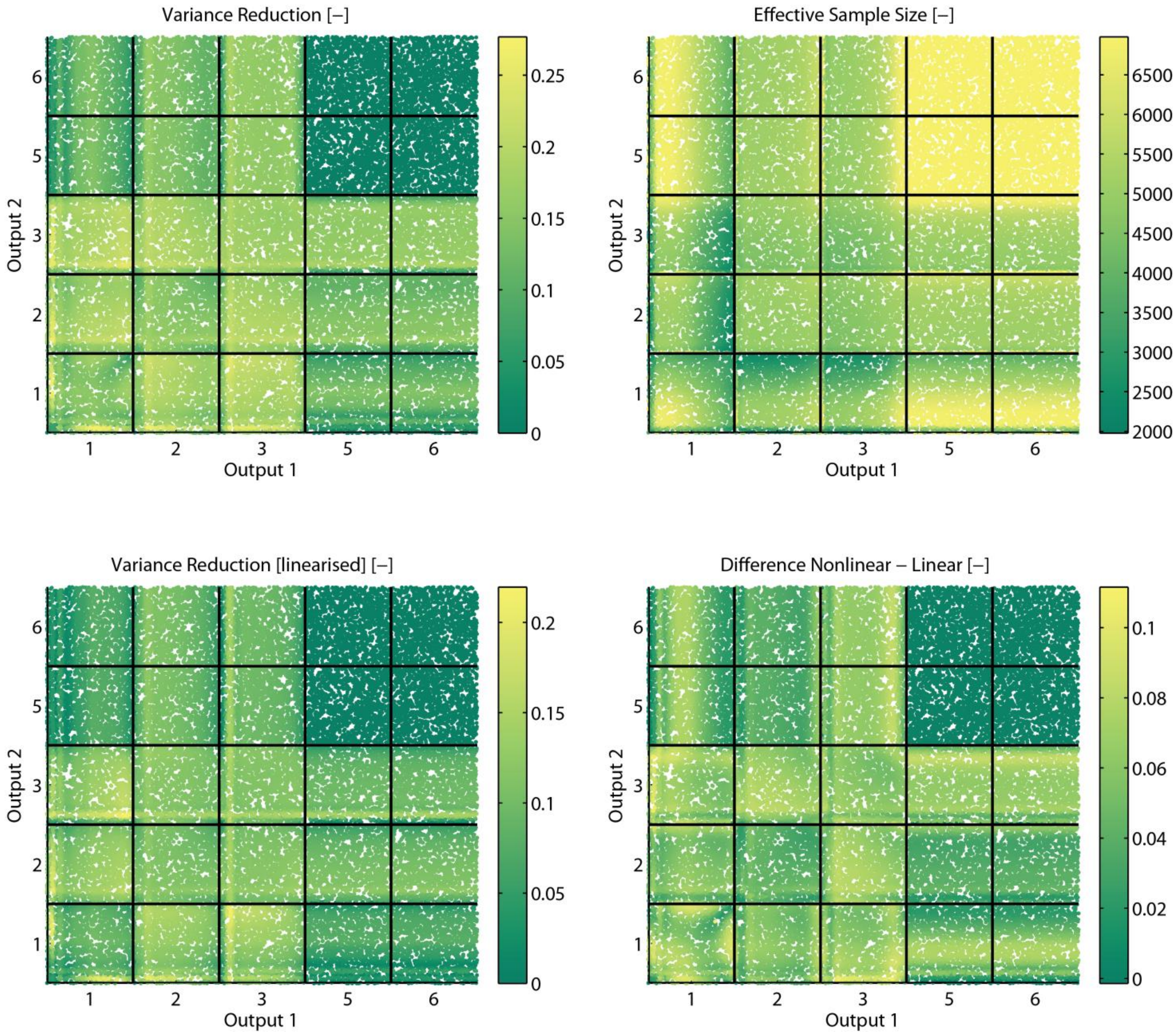
- **Step 1: Construct covariance matrix of output with potential measurements**

- **Step 2: Compute new covariance after additional measurements (multiplication of two Gaussians)**

- **Step 3: Compute resulting variance at the output**

$$\Sigma_{new}(1,1)$$

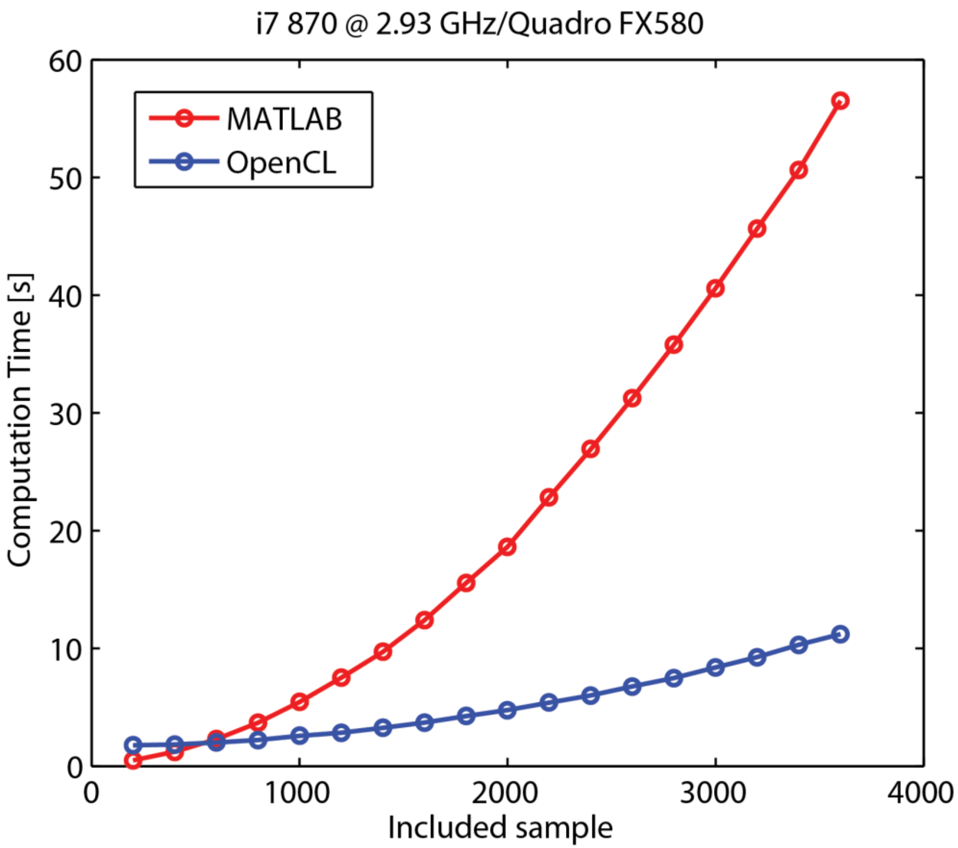
Comparison



Variance Reduction [–]
Effective Sample Size [–]
Variance Reduction [linearised] [–]
Difference Nonlinear – Linear [–]
Output 1
Output 2
1
2
3
5
6
0.25
0.2
0.15
0.1
0.05
0
6500
6000
5500
5000
4500
4000
3500
3000
2500
2000
0.08
0.06
0.04
0.02

Option 2:
GPU Programming

OpenCL



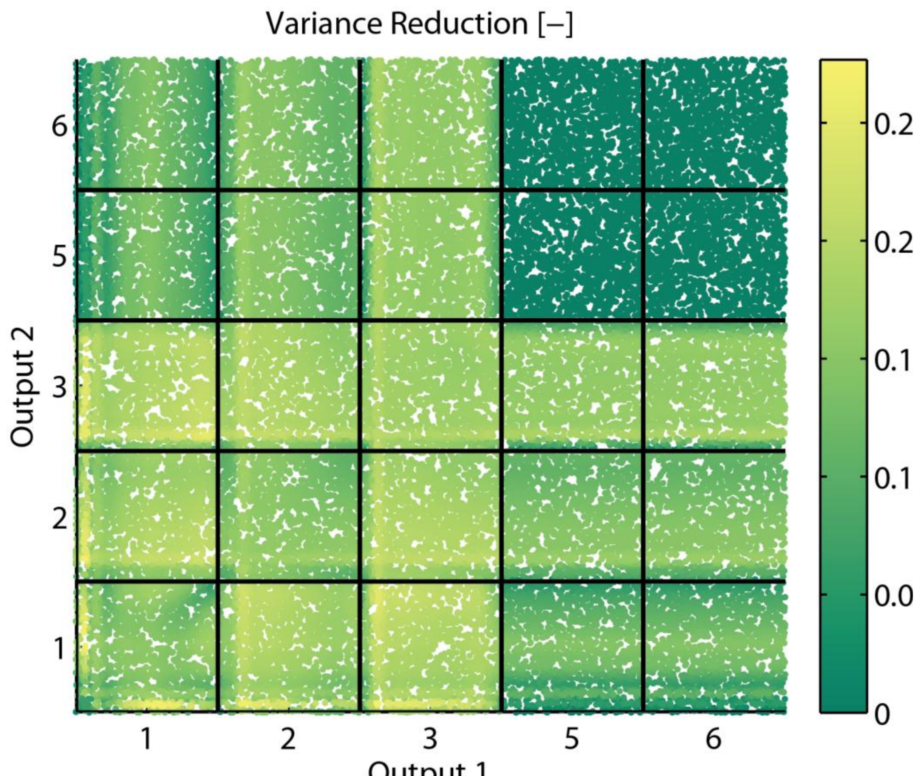
i7 870 @ 2.93 GHz/Quadro FX580
MATLAB
OpenCL
Computation Time [s]
Included sample
0
10
20
30
40
50
60
0
1000
2000
3000
4000

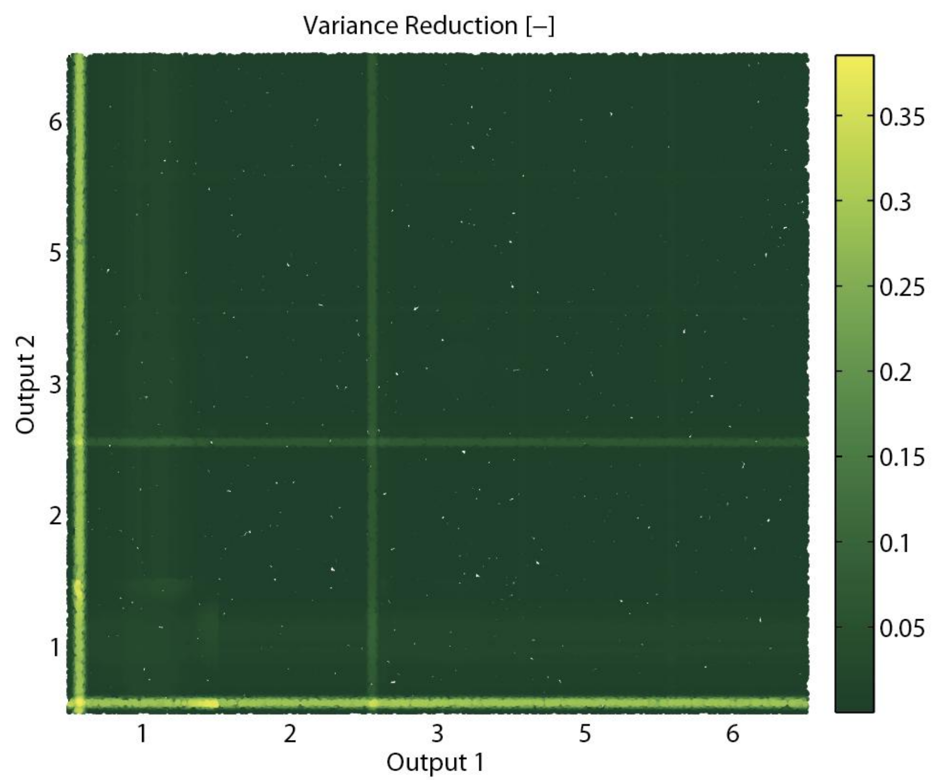
Comparison with Fisher based V-optimal experiment design

- **Fisher based experimental design**

$$F \approx \sum_{a=0}^{M} \sum_{i=1}^{N_a} \frac{1}{\sigma_{ai}} \left(\frac{\delta y_a(t_{ai},\theta)}{\delta\theta}\right)^T \frac{1}{\sigma_{ai}} \left(\frac{\delta y_a(t_{ai},\theta)}{\delta\theta}\right) = J^T J$$

$$Var_n(y_b(t)) \approx \left(\frac{\delta y_b(t,\theta)}{\delta\theta}\right)^T (F + F_n)^{-1} \left(\frac{\delta y_b(t,\theta)}{\delta\theta}\right)$$





Optional Sheets

Bayes Factors

- **Commonly used for testing competing hypotheses**

$$B_{12}$$

$$\frac{p(M_1 \mid D)}{p(M_2 \mid D)} = \boxed{\frac{\int p(D \mid M_1, \theta_1)\, p(\theta_1 \mid M_1)\, d\theta_1}{\int p(D \mid M_2, \theta_2)\, p(\theta_2 \mid M_2)\, d\theta_2}} \frac{p(M_1)}{p(M_2)}$$

Evidence for Model 2 | Likelihood | Parameter prior | Prior probabilities models

Kullback Leibler

Kullback-Leibler

- **The KL divergence effectively measures the average likelihood of observing (infinite) data with the distribution p if the particular model q actually generated the data.**

$$KL(P \mid Q) = E_p[\log \frac{p(i)/Z_P}{q(i)/Z_Q}]$$

Technische Universiteit
Eindhoven
University of Technology

Normalizing constant

- **Kullback Leiber**

$$KL(P|Q) = E_p[\log \frac{p(X_i)/Z_P}{q(X_i)/Z_Q}]$$

p,q - unnormalised density

Z - normalisation constants (integral)

$$\frac{Z_P}{Z_Q} = \frac{1}{N}\sum_{i=1}^{N}\frac{p_P(X_i)}{p_Q(X_i)}$$

With Xi sampled from Q (the old posterior)

Normalizing constant

- **Kullback Leiber**

$$KL(P|Q) = E_p[\log \frac{p(X_i)/Z_P}{q(X_i)/Z_Q}]$$

$$\frac{Z_P}{Z_Q} = \frac{1}{N}\sum_{i=1}^{N}\frac{p_P(X_i)}{p_Q(X_i)} = \frac{1}{N}\sum_{i=1}^{N} p_{ERR}(X_i)$$

With Xi sampled from Q (the old posterior)

Technische Universiteit
Eindhoven
University of Technology

Normalizing constant

- **Kullback Leiber**

$$KL(P|Q) = E_p[\log \frac{p(X_i)/Z_P}{q(X_i)/Z_Q}] \qquad \frac{Z_P}{Z_Q} = \frac{1}{N}\sum_{i=1}^{N} p_{ERR}(X_i)$$

Again we exploit Importance Sampling

$$KL(P|Q) = \frac{1}{N}\sum_{i}^{N} \frac{p(X_i)/Z_P}{q(X_i)/Z_Q} \log \frac{p(X_i)/Z_P}{q(X_i)/Z_Q}$$

With Xi sampled from Q (the old posterior)

Normalizing constant

- **Kullback Leiber**

$$KL(P|Q) = E_p[\log \frac{p(X_i)/Z_P}{q(X_i)/Z_Q}] \qquad \frac{Z_P}{Z_Q} = \frac{1}{N}\sum_{i=1}^{N} p_{ERR}(X_i)$$

Again we exploit Importance Sampling

$$KL(P|Q) = \frac{1}{N}\sum_{i}^{N} \frac{p(X_i)/Z_P}{q(X_i)/Z_Q} \log \frac{p(X_i)/Z_P}{q(X_i)/Z_Q} = \frac{1}{N}\sum_{i}^{N} p_{ERR}(X_i)\frac{Z_Q}{Z_P} \log\left(p_{ERR}(X_i)\frac{Z_Q}{Z_P}\right)$$

With Xi sampled from Q (the old posterior)

Normalizing constant

- **Kullback Leiber**

$$KL(P|Q) = E_p[\log \frac{p(X_i)/Z_P}{q(X_i)/Z_Q}] \qquad \frac{Z_P}{Z_Q} = \frac{1}{N}\sum_{i=1}^{N} p_{ERR}(X_i)$$

Again we exploit Importance Sampling

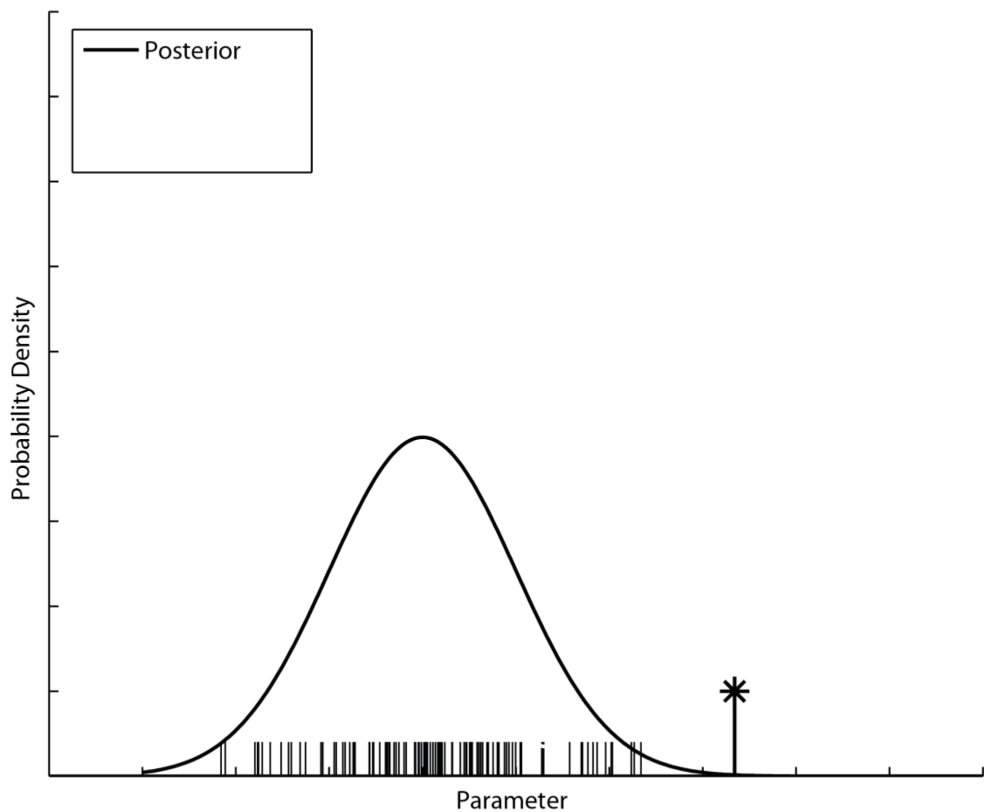
$$KL(P|Q) = \frac{1}{N}\sum_{i}^{N} p_{ERR}(X_i)\frac{Z_Q}{Z_P}\log\left(p_{ERR}(X_i)\frac{Z_Q}{Z_P}\right) = \sum_{i}^{N}\frac{p_{ERR}(X_i)}{\sum_{j=1}^{N} p_{ERR}(X_j)}\log\left(\frac{p_{ERR}(X_i)}{\frac{1}{N}\sum_{j=1}^{N} p_{ERR}(X_j)}\right)$$

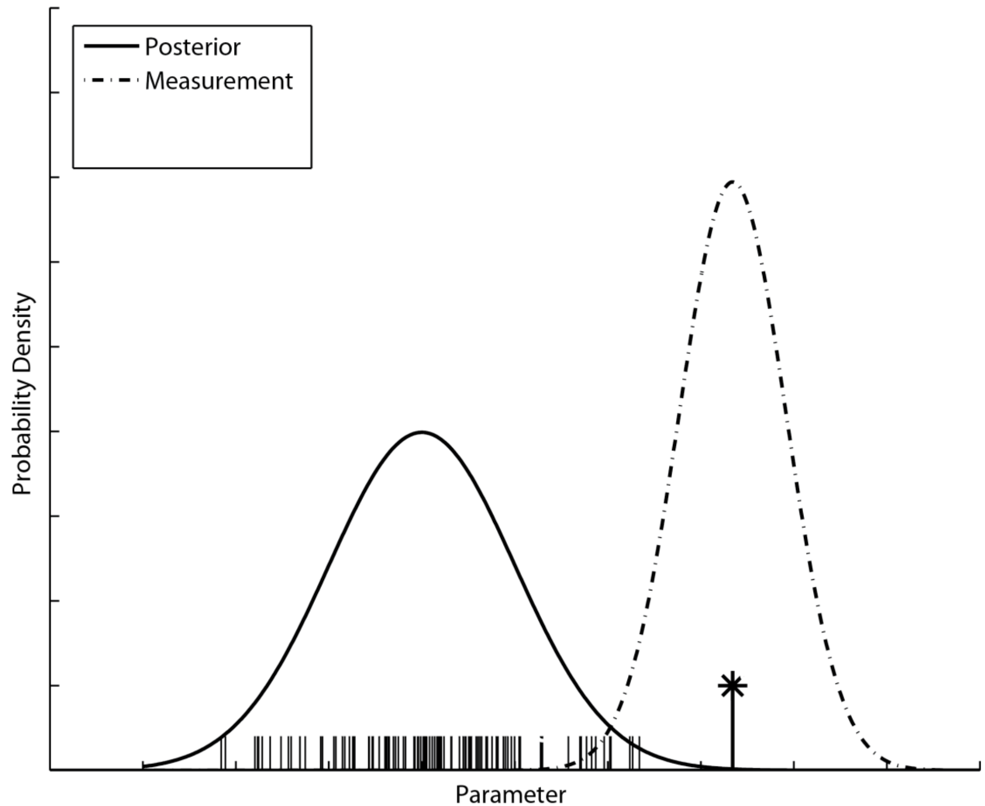
With Xi sampled from Q (the old posterior)

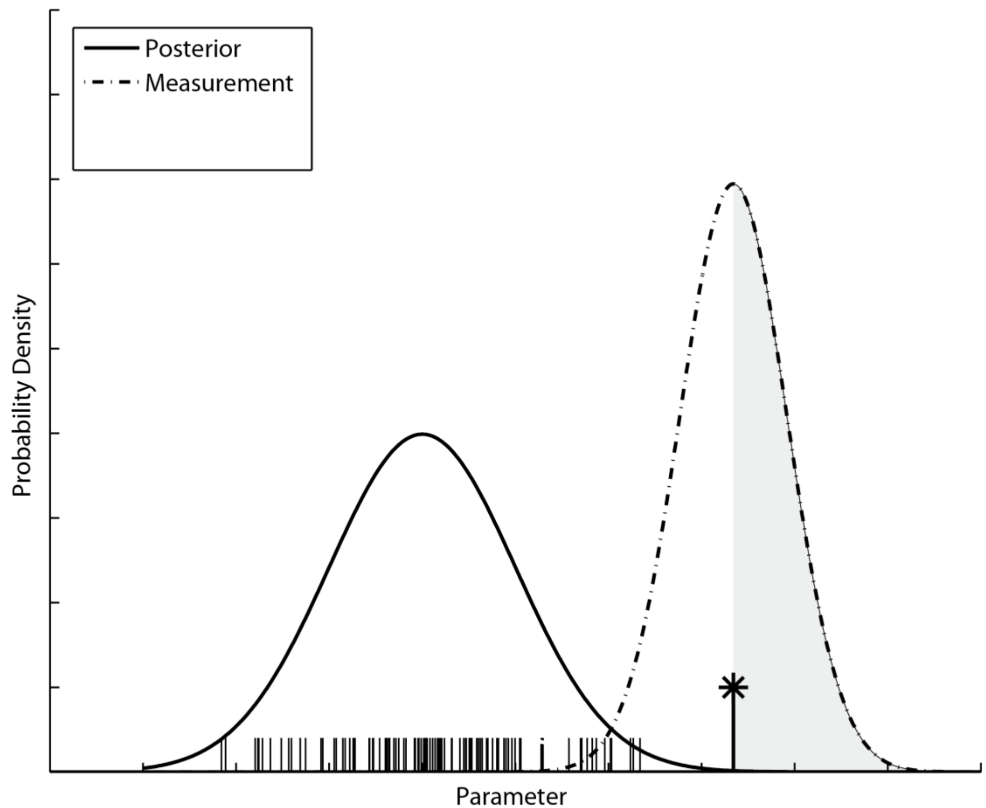
Degeneracy

$$ESS_r = \frac{\left(\sum_{t=1}^{N} p(y^N \mid \vec{\theta}_t)\right)^2}{\sum_{t=1}^{N} p(y^N \mid \vec{\theta}_t)^2}$$

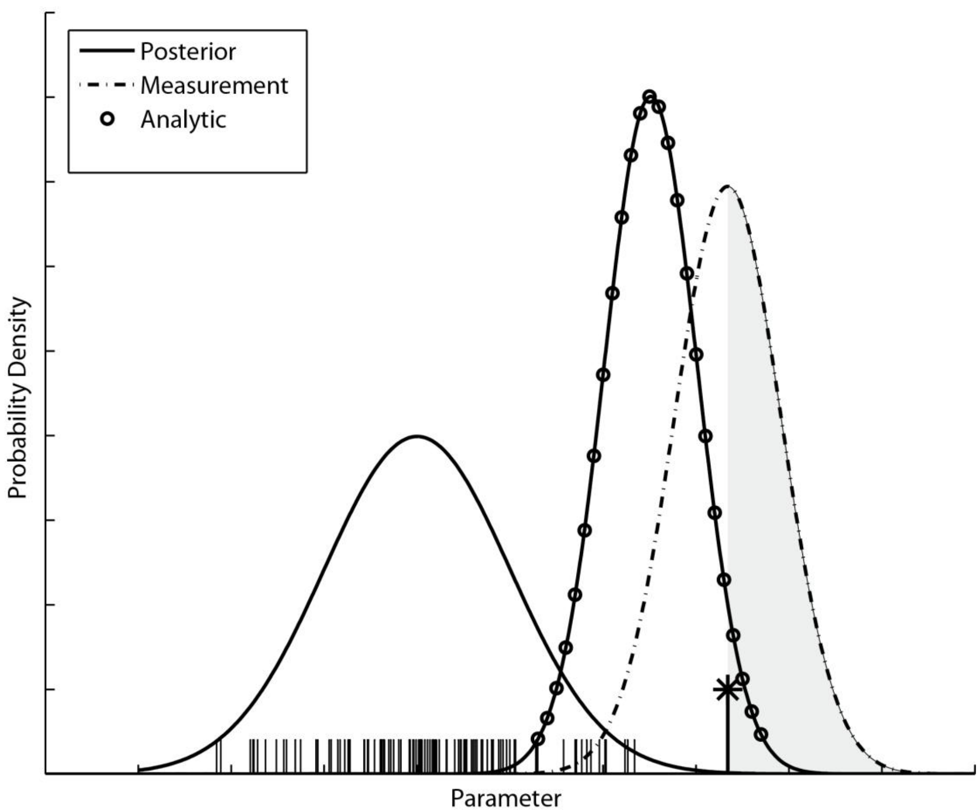
Tails


Posterior
Probability Density
Parameter


Posterior
Measurement
Probability Density
Parameter

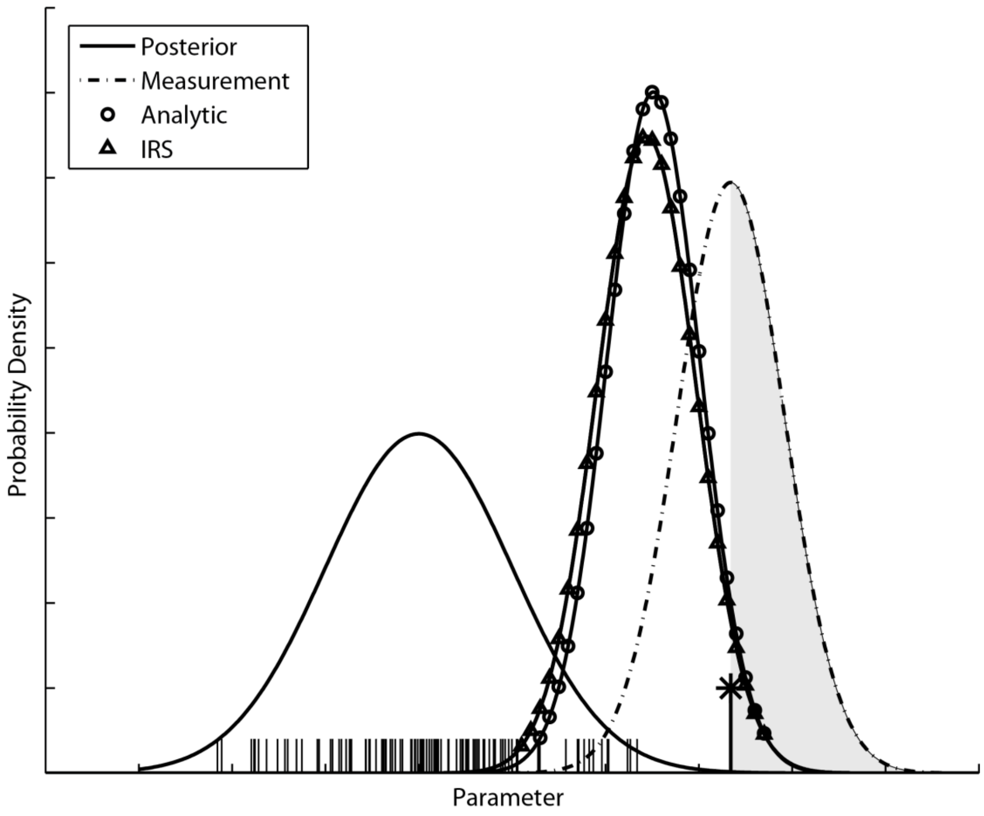


Posterior
Measurement
Probability Density
Parameter

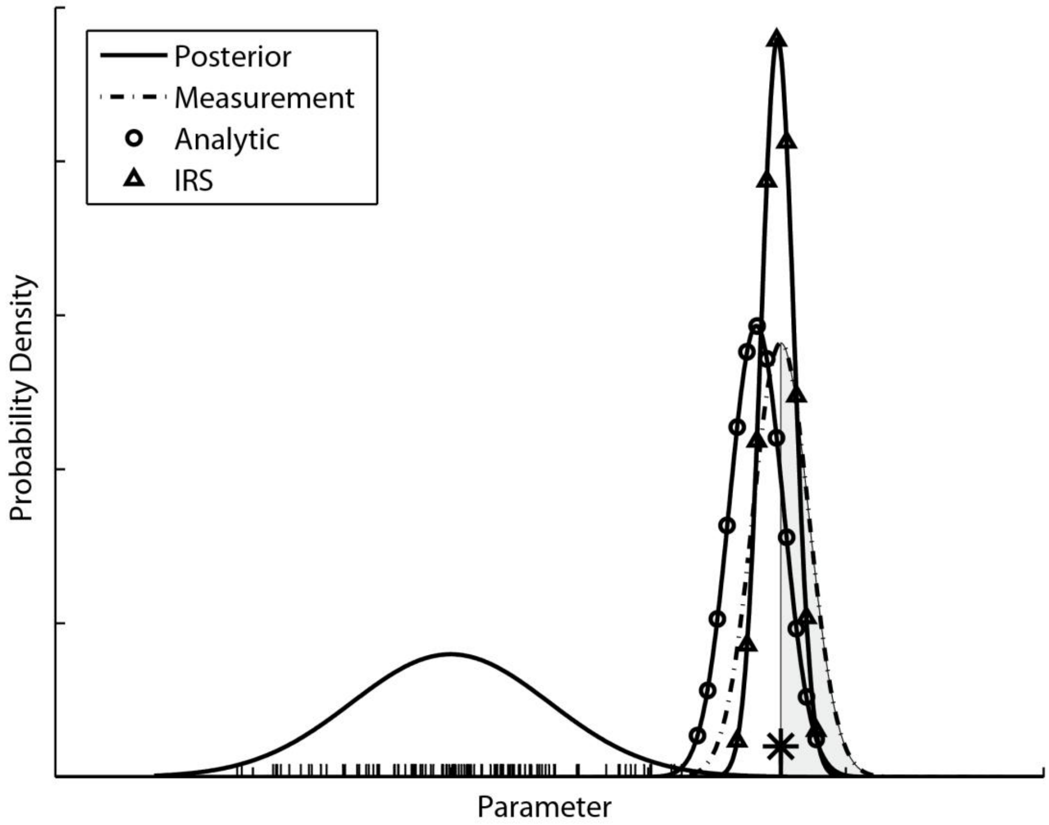


Posterior
Measurement
Analytic
Probability Density
Parameter

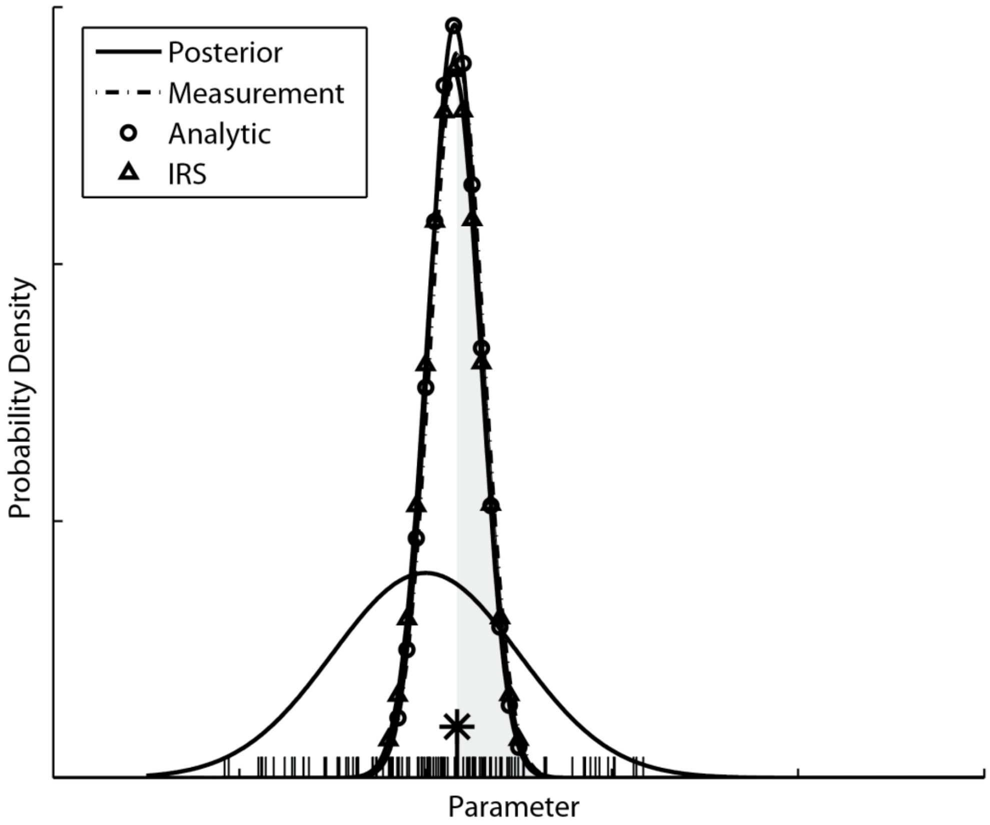
Worst case scenario for this measurement variance



Smaller measurement error



Not all is lost



Quick Recap: Importance Sampling

Assume independence:

$$E_p[f(\vec{\theta})] = \int_\Omega \underbrace{p(\vec{\theta})}_{\text{Old posterior}} \underbrace{p_{new}(\vec{\theta})}_{\text{Measurement Error Model}} f(\vec{\theta}) d\vec{\theta} \approx \frac{1}{N} \sum_{i=1}^{N} p_{new}(\vec{\theta}_i) f(\underbrace{\vec{\theta}_i}_{\text{Samples from p (MCMC)}})$$

Quick Recap: Importance Sampling

Assume independence:

$$\frac{1}{N}\sum_{i=1}^{N} p_{new}(\vec{\theta}_i)\, f(\vec{\theta}_i)$$

$p_{new}(\vec{\theta}_i)$ **only known up to normalising constant!**

Quick Recap: Importance Sampling

Assume independence:

$$\frac{1}{N}\sum_{i=1}^{N} p_{new}(\vec{\theta}_i)\, f(\vec{\theta}_i)$$

$p_{new}(\vec{\theta}_i)$ **only known up to normalising constant!**

Self-normalise

$$\frac{1}{\sum_{i=1}^{N} p_{new}(\vec{\theta}_i)}\sum_{i=1}^{N} p_{new}(\vec{\theta}_i)\, f(\vec{\theta}_i)$$

Posterior Predictive Distribution

Quantities based on expectations:

Variance $\mathrm{Var}\,[y] = E[y^2] - E[y]^2$

Kullback-Leibler Divergence $KL(P|Q) = E[\log(\frac{p(x)}{q(x)})]$

'Distance' between distributions
(entropy or 'information gain')

Importance (Re)Sampling Maths

Single new datapoint k. X_i comes from the posterior
No (1/N) in the first expression since we use self normalized weights in the IS step.

$$E_\theta[y\,|\,k] = \sum_i \frac{G(x(X_k), x(X_i))}{\sum_r G(x(X_k), x(X_r))} y(X_i)$$

Single
IS step

$$G(a,b) = e^{\frac{(a-b)^2}{2\sigma^2}}$$

Error model

$$E_k[E_\theta[y\,|\,k]] = \frac{1}{N} \sum_k \sum_i \frac{G(x(X_k), y(X_i))}{\sum_r G(x(X_k), y(X_r))} y(X_i)$$

IS for every sample

Posterior Predictive Distribution

Quantities based on expectations:

Variance $\mathrm{Var}[y] = E[y^2] - E[y]^2$

Kullback-Leibler Divergence $KL(P|Q) = E\left[\log\left(\frac{p(x)}{q(x)}\right)\right]$

‘Distance’ between distributions
(entropy or ‘information gain’)

Require Normalisation Constant?

Incredibly difficult to compute
(thermodynamic integration)
Requires priors on *all* states

Importance (Re)Sampling Maths

Single new datapoint k. X_i comes from the posterior
No (1/N) in the first expression since we use self normalized weights in the IS step.

$$E_\theta[y \mid k] = \sum_i \frac{G(x(X_k), x(X_i))}{\sum_r G(x(X_k), x(X_r))} y(X_i)$$

Single
IS step

$$G(a,b) = e^{\frac{(a-b)^2}{2\sigma^2}}$$

Error model

$$E_k[E_\theta[y \mid k]] = \frac{1}{N} \sum_k \sum_i \frac{G(x(X_k), y(X_i))}{\sum_r G(x(X_k), y(X_r))} y(X_i)$$

IS for every
sample

Importance (Re)Sampling Maths

Single new datapoint k. X_i comes from the posterior
No (1/N) in the first expression since we use self normalized weights in the IS step.

$$E_\theta[y\,|\,k] = \sum_i \frac{G(x(X_k), x(X_i))}{\sum_r G(x(X_k), x(X_r))}\, y(X_i)$$

Single
IS step

$$G(a,b) = e^{\frac{(a-b)^2}{2\sigma^2}}$$

Error model

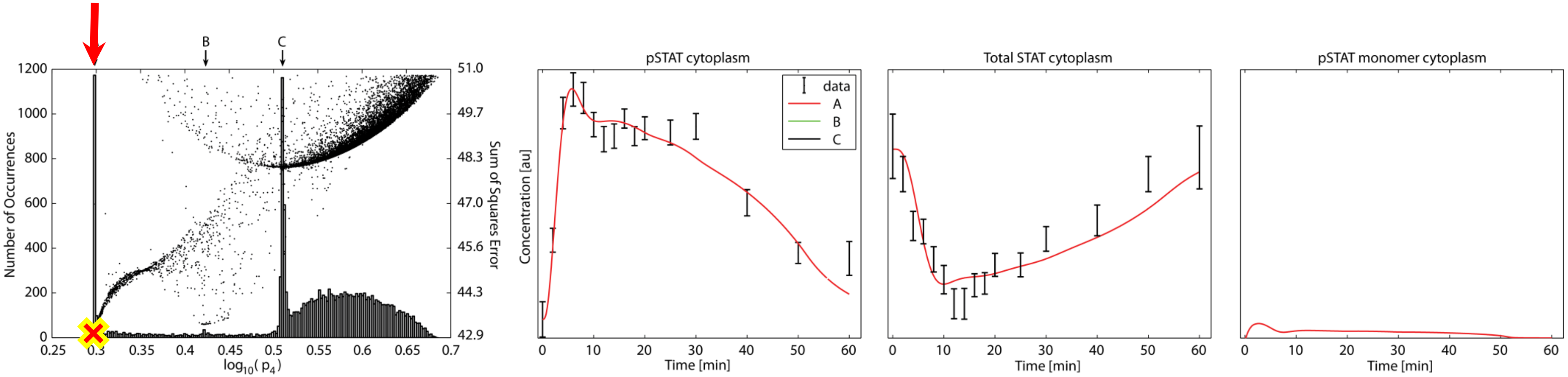
$$E_k[E_\theta[y\,|\,k]] = \frac{1}{N}\sum_k \sum_i \frac{G(x(X_k), y(X_i))}{\boxed{\sum_r G(x(X_k), y(X_r))}}\, y(X_i)$$

IS for every sample

Large scale search

Technische Universiteit
Eindhoven
University of Technology

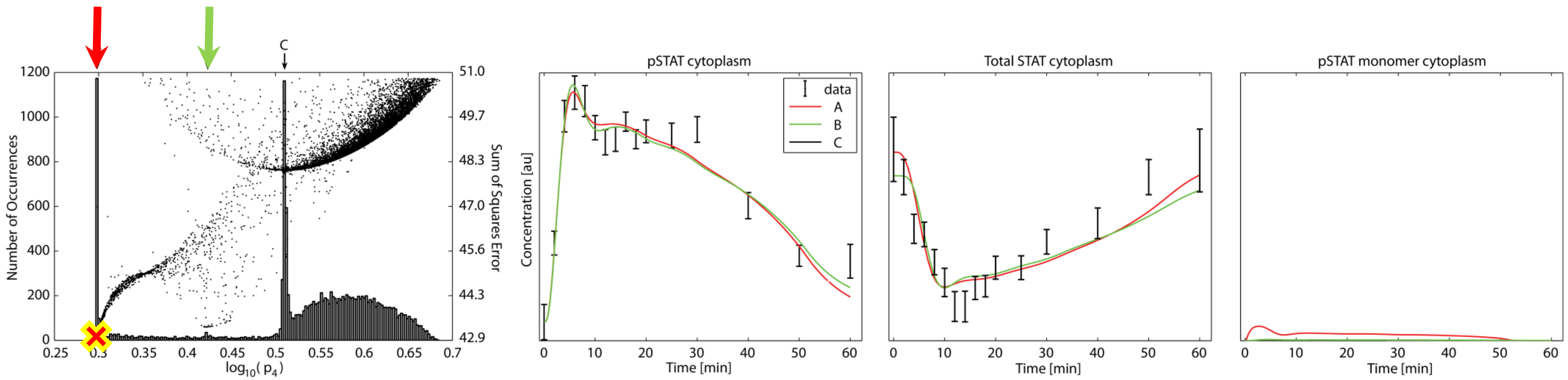
JAK-STAT (Large Scale Search)



Fit error

Minimum value

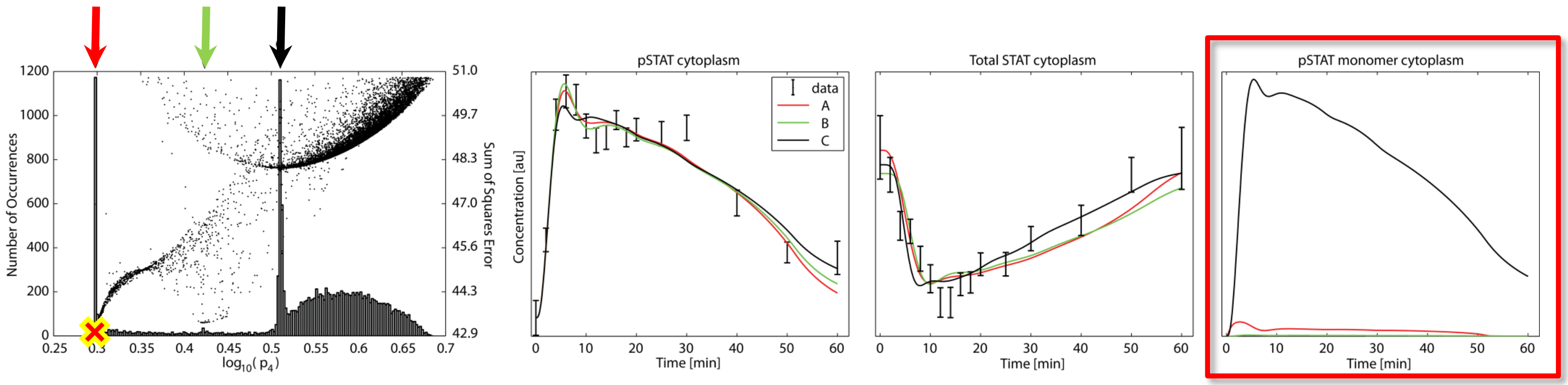
JAK-STAT (Large Scale Search)



Fit error

Minimum value ✖

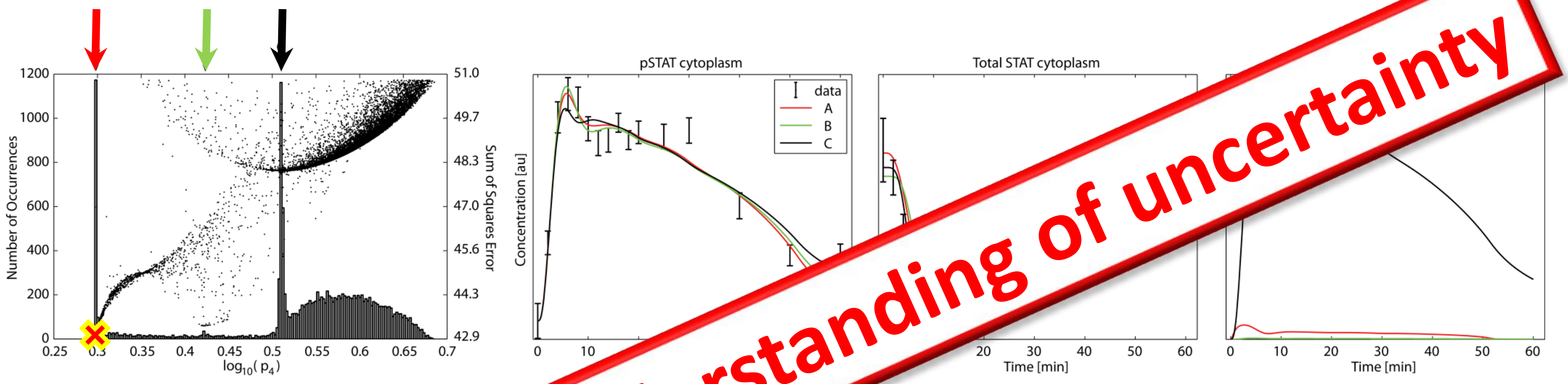
JAK-STAT (Large Scale Search)



Fit error

Minimum value ✖

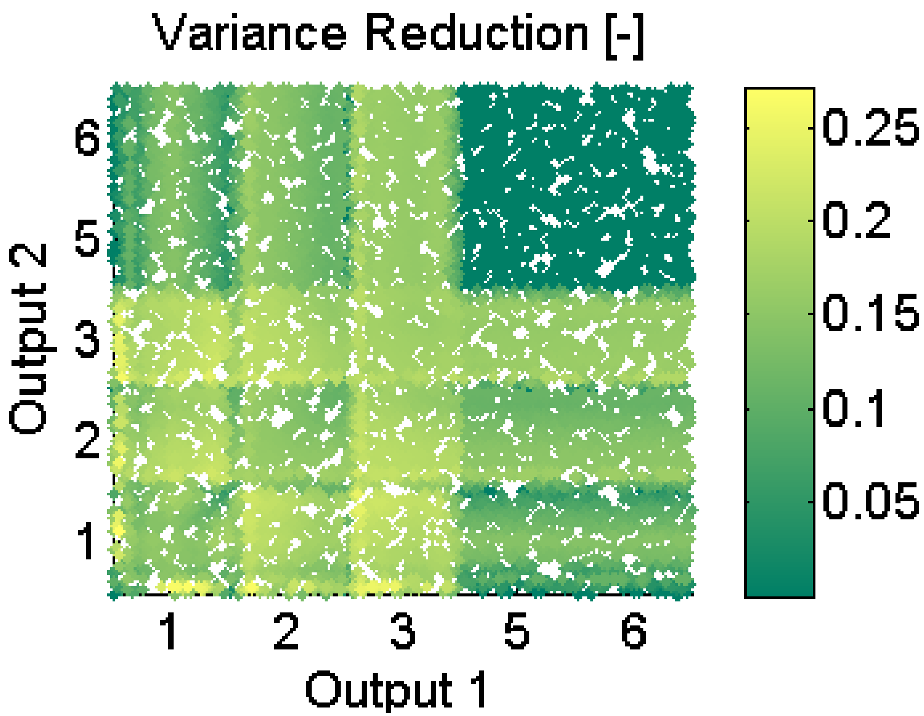
JAK-STAT (Large Scale Search)



Quantitative understanding of uncertainty

Minimum value

Linearisation



Sampling obtained when planning 2 experiments over 6 outputs whilst MC sampling experiment space for 8 hours.

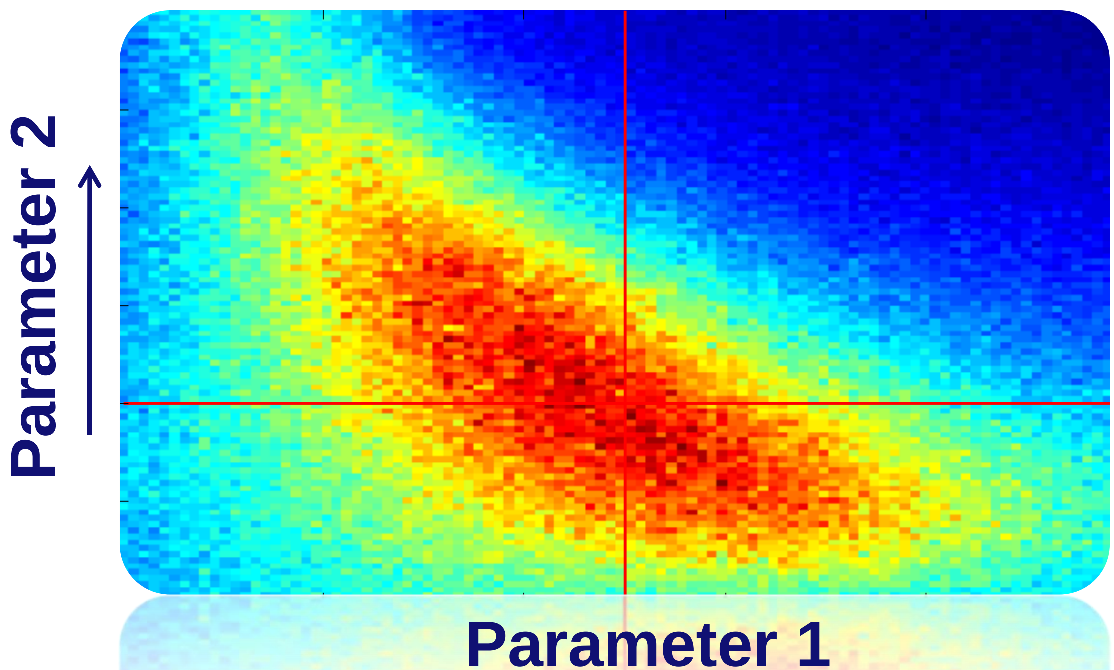
Still rather coarse!

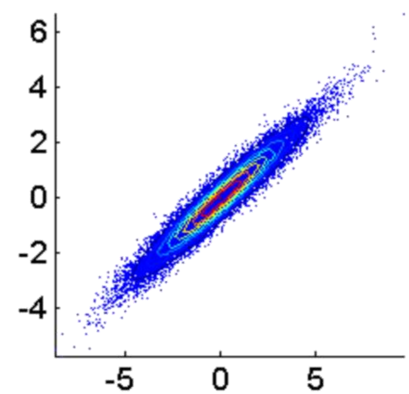
Proposal Distribution

Stochastic methods:
Markov Chain Monte Carlo

- **Proposal distribution**
  - **Quadratic approximation based on model sensitivities J**
  - **Include non-uniform priors in approximation**

$$H = J^TJ + P$$





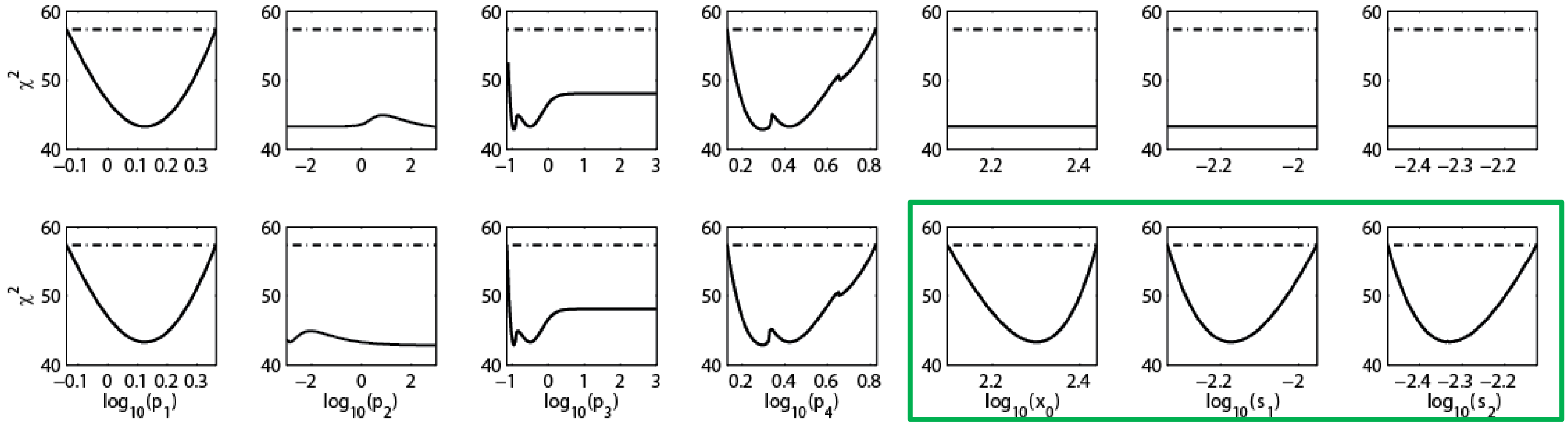
Technische Universiteit Eindhoven University of Technology

Stochastic methods:
Markov Chain Monte Carlo

- **Chain gets ‘stuck’ when quadratic approximation of the landscape is poor (or nearly singular)!**
- **Trust region approach!** $\mathbf{H}_{trust} = \mathbf{H} + \lambda \mathbf{I}$
- **Predict error we’re going to get based on quadratic error landscape ‘model’**
- **If true error is much higher → Smaller proposal**

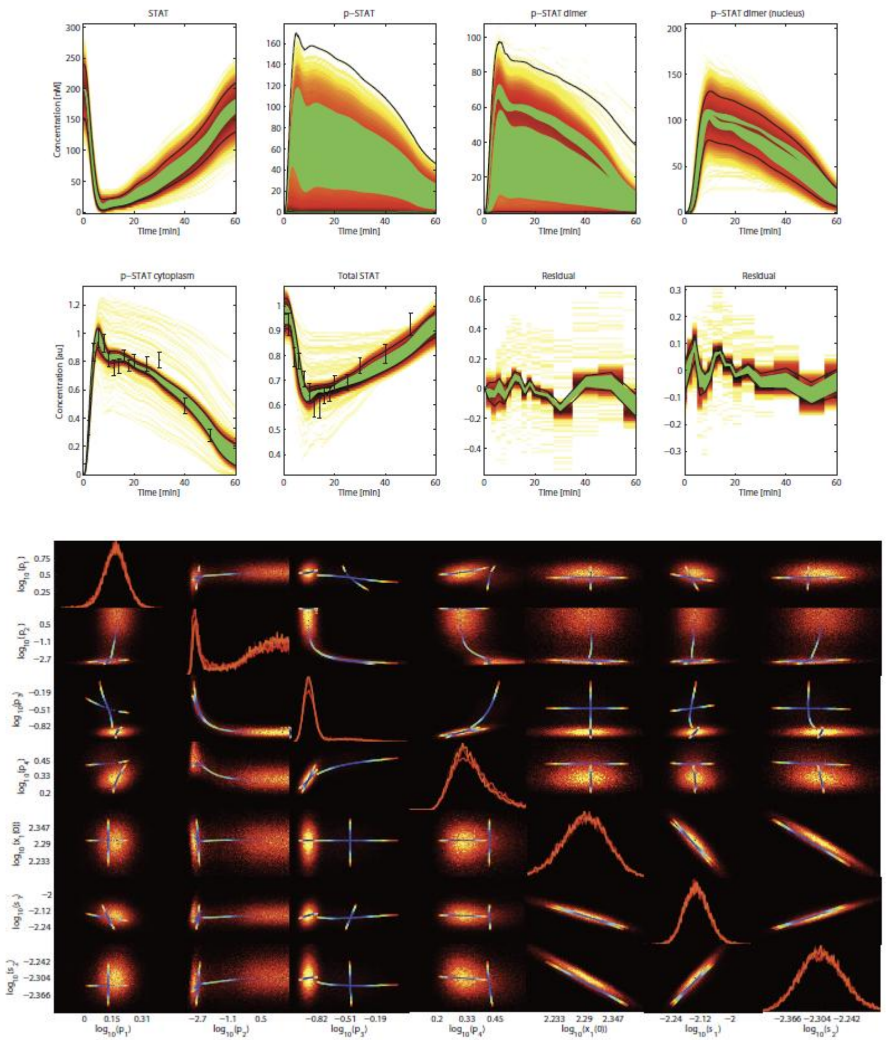
Effect of Priors

JAK-STAT analysis

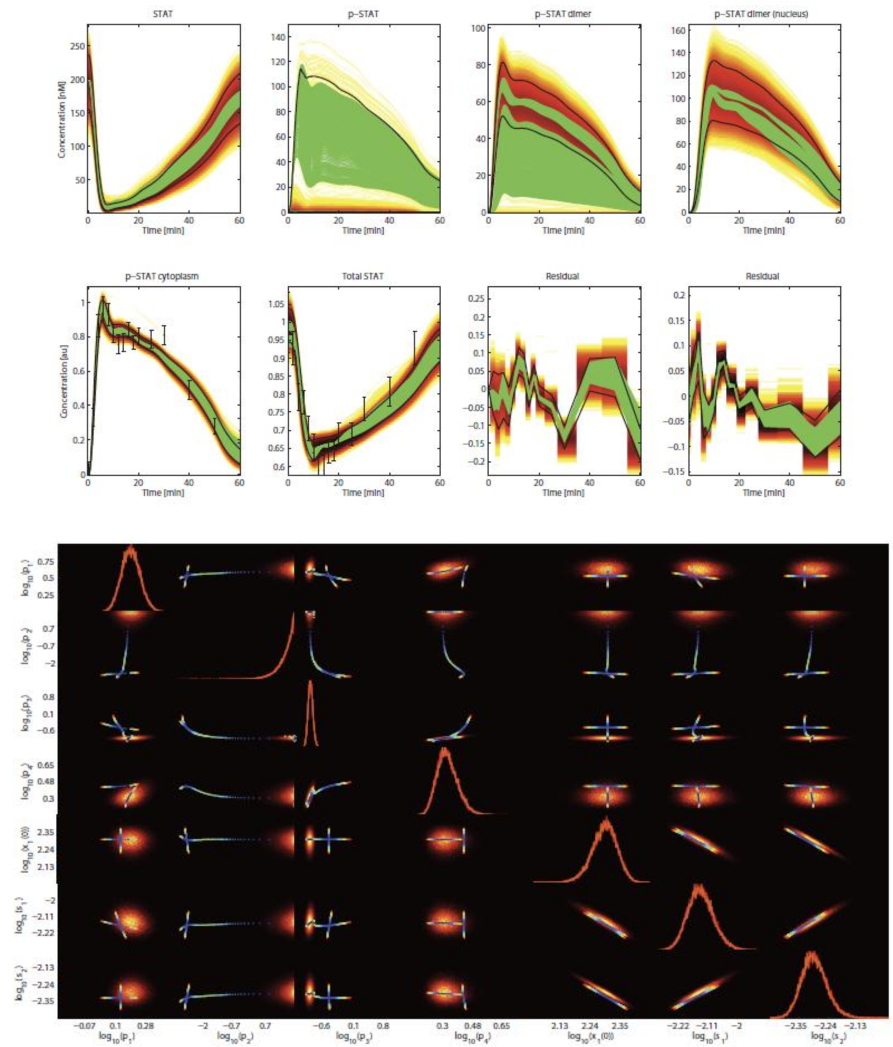


Initial concentration 200 ± 20 nM

Profile Likelihood vs Bayesian



Logarithmic Prior



Uniform Prior

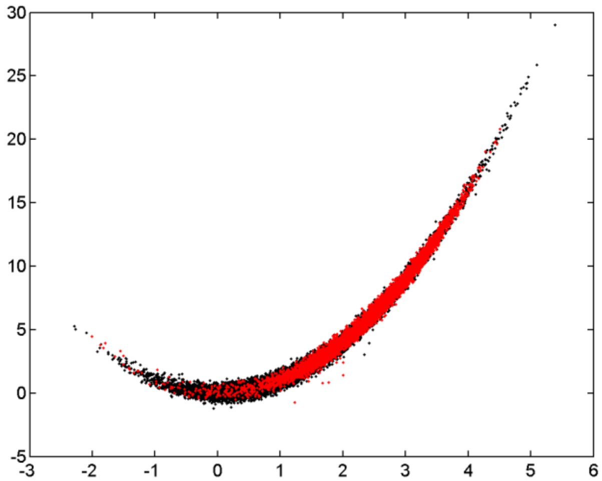
Numerical experiments

Explore the problem:
Numerical Tests!

Analytical 2D Posterior distribution

MC[4]

1. Obtain sample X_i from posterior
2. For all X_i, run MCMC augmenting 'data' with X_i



$$\vec{r}(\vec{x}) = \left[\sqrt{10}\left(x_2 - x_1^{\,2}\right), \sqrt{2} - x_1, \frac{X_i - x_1}{\sigma}\right]$$

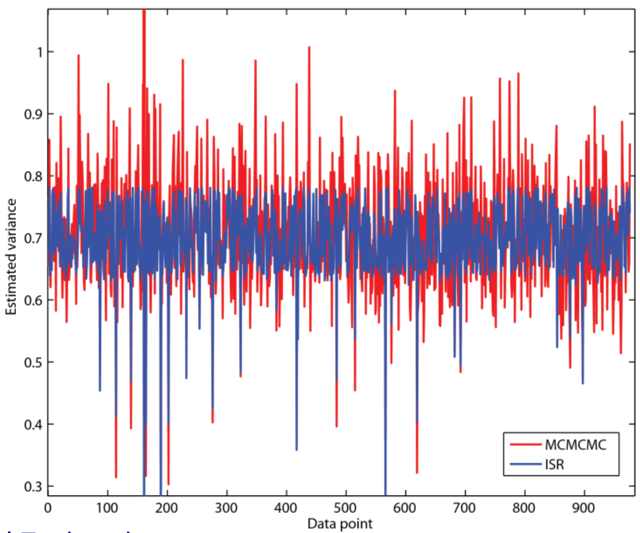
Explore the problem:
Numerical Tests!

Analytical 2D Posterior distribution

$$1 - E_{post}\left[\frac{V_{new}}{V_{post}}\right]$$

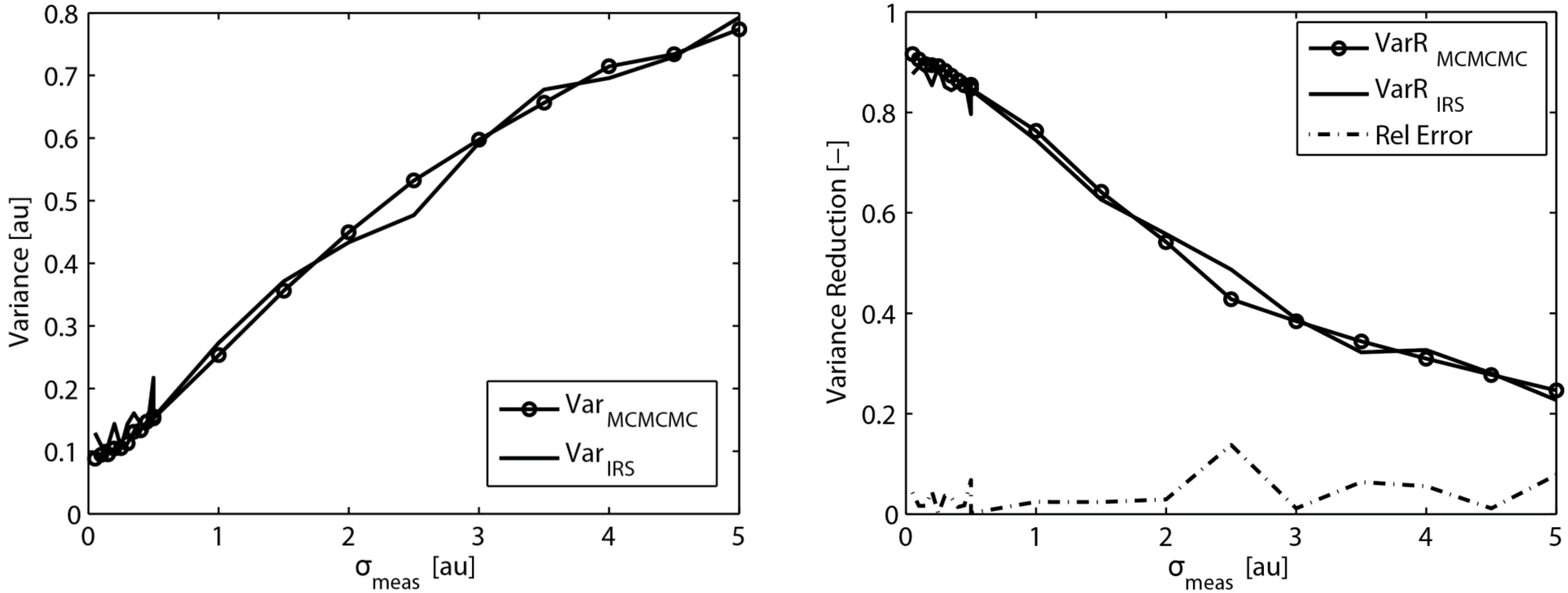
IRS

1. Obtain sample from posterior
2. For each sample, employ importance sampling



➔ Mean looks OK.

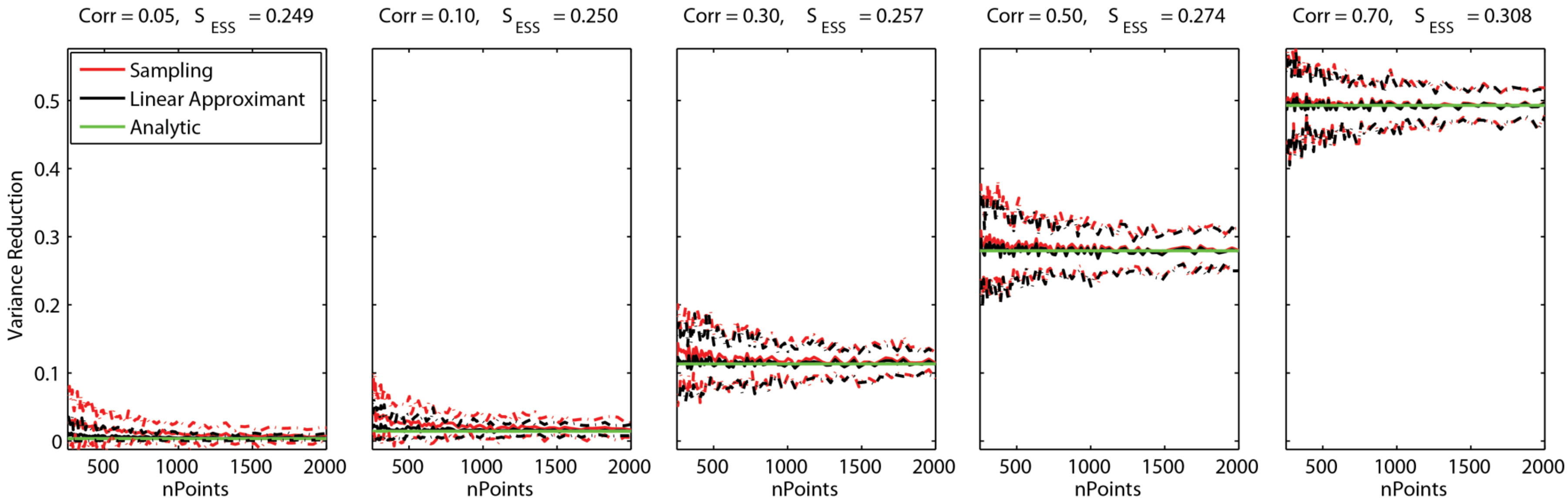
Banana function results:



$$VarR = 1 - E_{post}\left[\frac{V_{new}}{V_{post}}\right]$$

TU/e Technische Universiteit Eindhoven University of Technology

Bias



Corr = 0.05, S ESS = 0.249
Corr = 0.10, S ESS = 0.250
Corr = 0.30, S ESS = 0.257
Corr = 0.50, S ESS = 0.274
Corr = 0.70, S ESS = 0.308
Sampling
Linear Approximant
Analytic
Variance Reduction
0
0.1
0.2
0.3
0.4
0.5
500
1000
1500
2000
nPoints

Quick Recap: Importance Sampling

Computing an expected value:

$$E_p[f(\vec{\theta})] = \int_\Omega \underbrace{p(\vec{\theta})}_{\text{Probability}} \underbrace{f(\vec{\theta})}_{\text{Quantity}} d\vec{\theta} \approx \frac{1}{N}\sum_{i=1}^{N} f(\underbrace{\vec{\theta}_i}_{\text{Samples from p}})$$

Quick Recap: Importance Sampling

Computing an expected value:

$$E_p[f(\vec{\theta})] = \int_\Omega \underbrace{p(\vec{\theta})}_{\text{Probability}} \underbrace{f(\vec{\theta})}_{\text{Quantity}} d\vec{\theta} \approx \frac{1}{N}\sum_{i=1}^{N} f(\underbrace{\vec{\theta}_i}_{\text{Samples from p}})$$

Importance sampling:

$$E_p[f(\vec{\theta})] = \int_\Omega \frac{p(\vec{\theta})}{g(\vec{\theta})} \underbrace{g(\vec{\theta})}_{\text{Sampling distribution}} f(\vec{\theta}) d\vec{\theta}$$

Quick Recap: Importance Sampling

Computing an expected value:

$$E_p[f(\vec{\theta})] = \int_\Omega \underbrace{p(\vec{\theta})}_{\text{Probability}}\, \underbrace{f(\vec{\theta})}_{\text{Quantity}}\, d\vec{\theta} \approx \frac{1}{N}\sum_{i=1}^{N} f(\underbrace{\vec{\theta}_i}_{\text{Samples from p}})$$

Importance sampling:

$$E_p[f(\vec{\theta})] = \int_\Omega \frac{p(\vec{\theta})}{g(\vec{\theta})}\, \underbrace{g(\vec{\theta})}_{\substack{\text{Sampling}\\ \text{distribution}}}\, f(\vec{\theta})\, d\vec{\theta} \approx \frac{1}{N}\sum_{j=1}^{N} \underbrace{\frac{p(\vec{\theta}_j)}{g(\vec{\theta}_j)}}_{\substack{\text{Probability}\\ \text{ratio}}}\, f(\underbrace{\vec{\theta}_j}_{\substack{\text{Samples}\\ \text{from } g}})$$

Quick Recap: Importance Sampling

Assume **independence**:

$$\frac{1}{N}\sum_{i=1}^{N} p_{new}(\vec{\theta}_i)\, f(\vec{\theta}_i)$$

$p_{new}(\vec{\theta}_i)$ **only known up to normalising constant!**

Importance Sampling

Computing an expected value using Importance Sampling:

$$E_p[f(\vec{\theta})] = \int_\Omega \frac{p(\vec{\theta})}{g(\vec{\theta})} \underbrace{g(\vec{\theta})}_{\text{Sampling distribution}} \underbrace{f(\vec{\theta})}_{\text{Quantity}} d\vec{\theta} \approx \frac{1}{N}\sum_{j=1}^{N} \underbrace{\frac{p(\vec{\theta}_j)}{g(\vec{\theta}_j)}}_{\text{Probability ratio}} f(\underbrace{\vec{\theta}_j}_{\text{Samples from } g})$$

From: Old posterior

New: Old posterior + additional experiment

$$p_{updated}(\vec{\theta}) \propto p_{old}(\vec{\theta})\, p_{new}(\vec{\theta})$$

Quick Recap: Importance Sampling

Assume noise independence:

Measurement
Error Model

$$E_p[f(\vec{\theta})] = \int_\Omega p_{new}(\vec{\theta}) f(\vec{\theta}) d\vec{\theta} = \int_\Omega p(\vec{\theta}) \frac{p_{new}(\vec{\theta})}{p(\vec{\theta})} f(\vec{\theta}) d\vec{\theta}$$

Old posterior

$$\approx \frac{1}{N} \sum_{i=1}^{N} \frac{p_{new}(\vec{\theta}_i)}{\sum_{j=1}^{N} p_{new}(\vec{\theta}_j)} f(\vec{\theta}_i)$$

Samples from
old posterior (MCMC)

Model Selection

Can we perform Experiment Design for Model Selection?

Bayes Factors

Bayes Factors

- **Commonly used for testing competing hypotheses**
- **Evidence for each model given the available data**

$$B_{12} = \frac{p(M_1 \mid D)}{p(M_2 \mid D)}$$

Bayes Factors

- **Commonly used for testing competing hypotheses**
- **Evidence for each model given the available data**

$$B_{12} = \frac{p(M_1 \mid D)}{p(M_2 \mid D)} \qquad p(M_1 \mid D) = \frac{p(D \mid M_1)P(M_1)}{P(D)}$$

Bayes Factors

- **Commonly used for testing competing hypotheses**
- **Evidence for each model given the available data**

$$B_{12} = \frac{p(M_1 \mid D)}{p(M_2 \mid D)}$$

$\log_{10}(B)$	B	Evidence support
0 to 1/2	1 to 3.2	Not worth more than a bare mention
1/2 to 1	3.2 to 10	Substantial
1 to 2	10 to 100	Strong
>2	>100	Decisive

Technische Universiteit
Eindhoven
University of Technology

Bayes Factors

- **Commonly used for testing competing hypotheses**

$$\underbrace{\frac{p(M_1 \mid D)}{p(M_2 \mid D)}}_{\text{Evidence for Model 2}} = \frac{\int p(D \mid M_1, \theta_1)\, p(\theta_1 \mid M_1)\, d\theta_1}{\int \underbrace{p(D \mid M_2, \theta_2)}_{\text{Likelihood}}\, \underbrace{p(\theta_2 \mid M_2)}_{\text{Parameter prior}}\, d\theta_2} \underbrace{\frac{p(M_1)}{p(M_2)}}_{\text{Prior probabilities models}}$$

Technische Universiteit **Eindhoven** University of Technology

Calculating the horrible integral

$$\underbrace{p(M_1 \mid D)}_{\text{Evidence for model 1}} = \int \underbrace{p(D \mid M_1, \theta_1)}_{\text{Likelihood}} \underbrace{p(\theta_1 \mid M_1)}_{\text{Parameter prior}} d\theta_1$$

Technische Universiteit Eindhoven University of Technology

Calculating the horrible integral

Option 1: Prior Mean Estimator

$$\int p(\theta_1 \mid M_1)\, p(D \mid M_1, \theta_1)\, d\theta_1 \approx \frac{1}{N} \sum_i^N p(D \mid M_1, \theta_i)$$

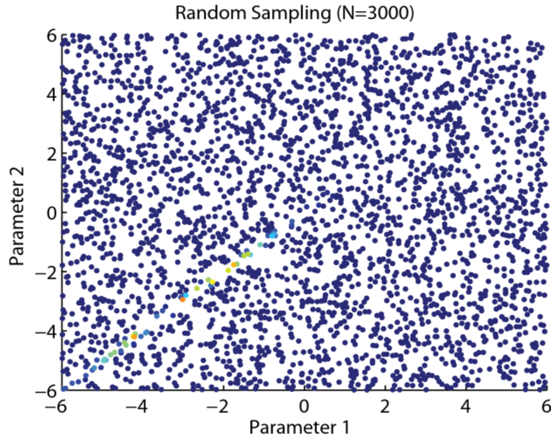
Use samples from prior

Calculating the horrible integral

Option 1: Prior Mean Estimator

$$\int p(\theta_1 \mid M_1) p(D \mid M_1, \theta_1) d\theta_1 \approx \frac{1}{N} \sum_{i}^{N} p(D \mid M_1, \theta_i)$$

Use samples from prior

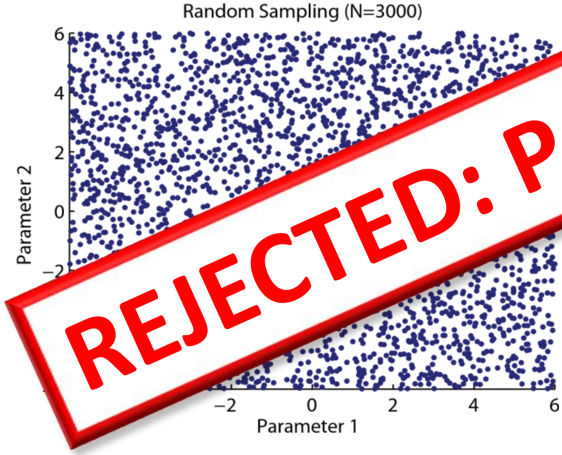


Calculating the horrible integral

Option 1: Prior Mean Estimator

$$\int p(\theta_1 \mid M_1) p(D \mid M_1, \theta_1) d\theta_1 \approx \frac{1}{N} \sum^{N} p(\ldots$$

Use samples from prior

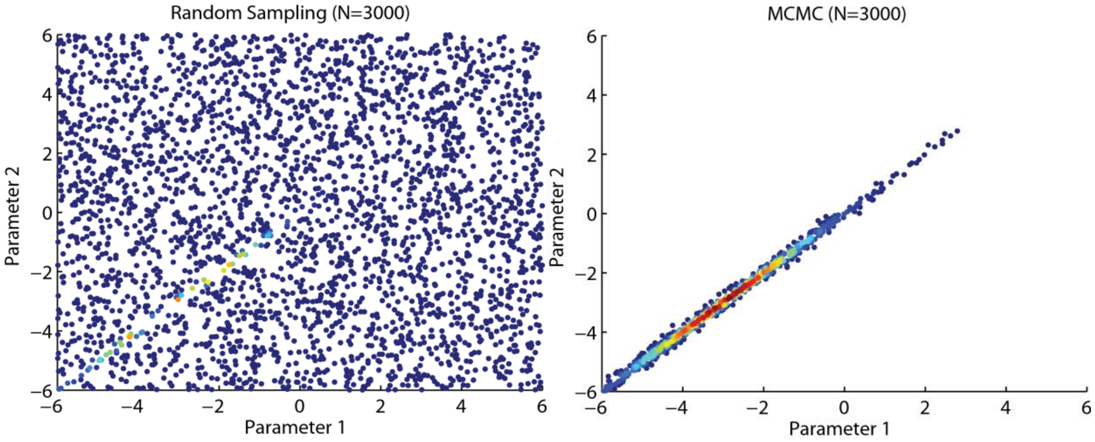


REJECTED: Poor sampling performance

Calculating the horrible integral

Option 2: Harmonic Mean Estimator

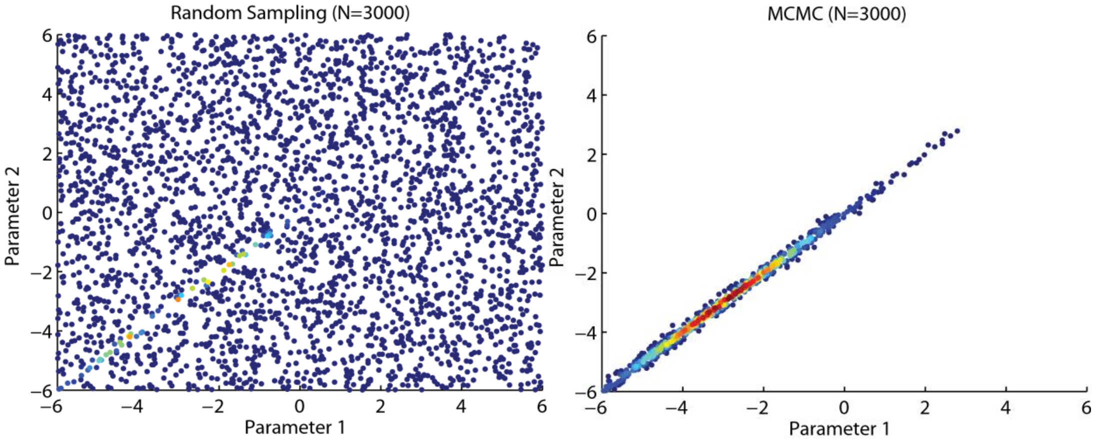
$$\int p(\theta_1 \mid M_1)\, p(D \mid M_1, \theta_1)\, d\theta_1 \approx \frac{1}{N}\left(\sum_{i}^{N} \frac{1}{p(D \mid M_1, \theta_i)}\right)^{-1}$$



Calculating the horrible integral

Option 2: Harmonic Mean Estimator

$$\int p(\theta_1 \mid M_1)\, p(D \mid M_1, \theta_1)\, d\theta_1 \approx \frac{1}{N}\left(\sum_i^N \frac{1}{p(D \mid M_1, \theta_i)} \right)^{-1}$$



Use samples from posterior

Calculating the horrible integral

Option 2: Harmonic Mean Estimator

$$\int p(\theta_1 \mid M_1)\, p(D \mid M_1, \theta_1)\, d\theta_1 \approx \frac{1}{N}\left(\sum_i^N \frac{1}{p(D \mid M_1, \theta_i)}\right)^{-1}$$

Use samples from posterior

Odd property:
Least probable sample contributes the most!

Calculating the horrible integral

Option 2: Harmonic Mean Estimator

$$\int p(\theta_1 \mid M_1)\, p(D \mid M_1, \theta_1)\, d\theta_1 \approx \frac{1}{N}\left(\sum_{i}^{N} \frac{1}{p(D \mid M_1, \theta_i)}\right)^{-1}$$

Use samples from posterior

The Harmonic Mean of the Likelihood: Worst Monte Carlo Method Ever

2008-08-17 at 12:09

The bad news is that the number of points required for this estimator to get close to the right answer will often be greater than the number of atoms in the observable universe. The even worse news is that it's easy for people to not realize this, and to naively accept estimates that are nowhere close to the correct value of the marginal likelihood.

Calculating the horrible integral

Option 2: Harmonic Mean Estimator

$$\int p(\theta_1 \mid M_1)\, p(D \mid M_1, \theta_1)\, d\theta_1 \approx \frac{1}{N}\left(\sum_{i=1}^{N} \frac{1}{p(D \mid M_1, \theta_i)}\right)^{-1}$$

REJECTED: Highly unstable

Use samples from posterior

Calculating the horrible integral

Option 3: Thermodynamic integration

Define a set of intermediate distributions and integrate over these

Calculating the horrible integral

Option 3: Thermodynamic integration

Define a set of intermediate distributions and integrate over these

$$\int \frac{d}{dt} \ln\{z(y \mid t)\} dt = \ln\{z(y \mid t = 1)\} - \ln\{z(y \mid t = 0)\}$$

Calculating the horrible integral

Option 3: Thermodynamic integration

Define a set of intermediate distributions and integrate over these

$$\int_0^1 \frac{d}{dt}\ln\{z(y\mid t)\}dt = \ln\{z(y\mid t=1)\} - \ln\{z(y\mid t=0)\}$$

$$\frac{d}{dt}\ln\{z(y\mid t)\} = \frac{d}{dt}\int e^{\ln(L(y|\theta))t}\, p(\theta)d\theta$$

$$\frac{1}{z(y_n\mid t)}\int e^{\ln(L(y|\theta))t}\ln\big(L(y\mid\theta)\big)p(\theta)d\theta$$

$$\frac{1}{z(y_n\mid t)}\int L(y\mid\theta)^t\ln\big(L(y\mid\theta)\big)p(\theta)d\theta$$

Calculating the horrible integral

Option 3: Thermodynamic integration

Define a set of intermediate distributions and integrate over these

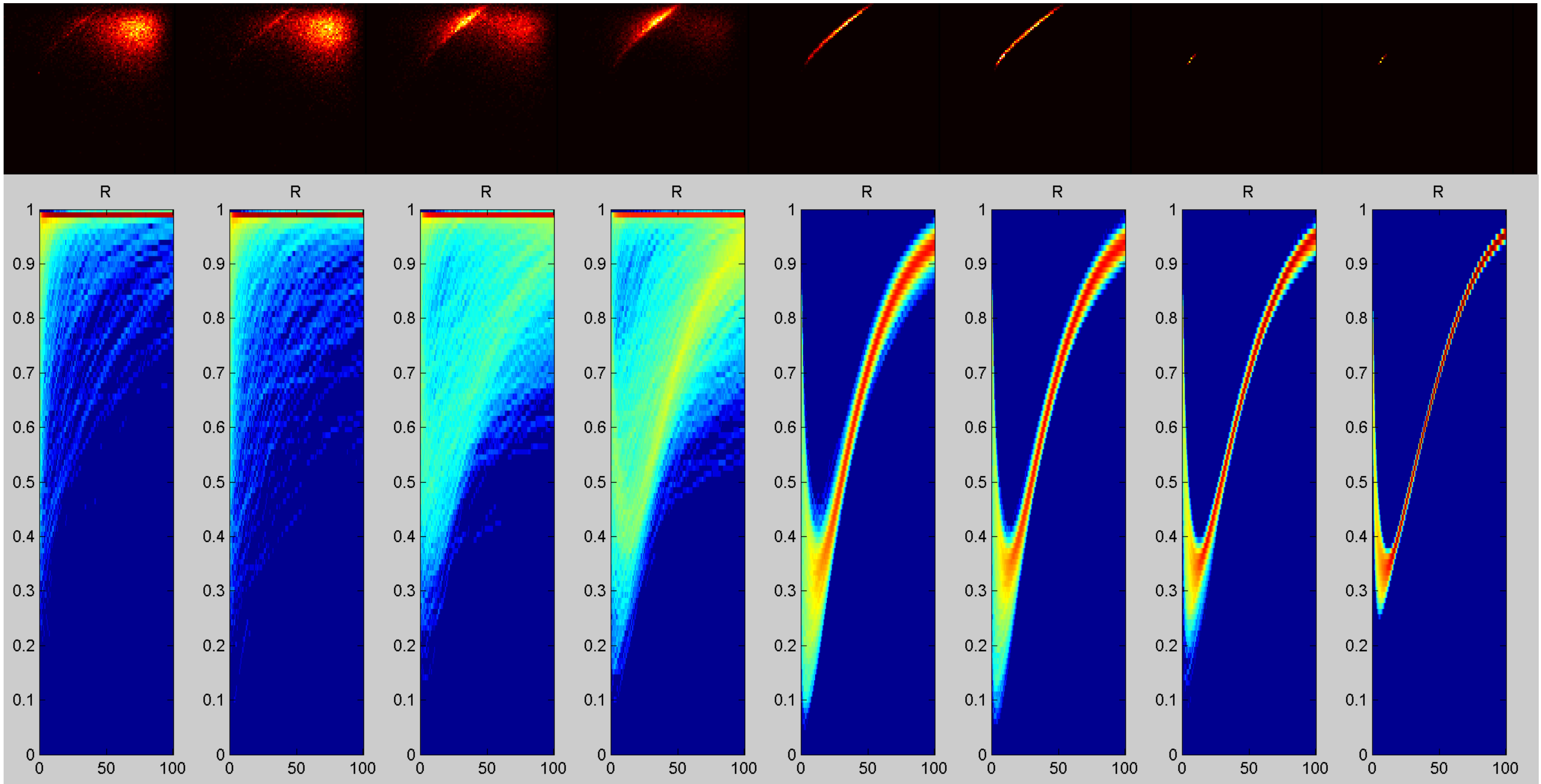
Run a number of MCMC chains at different temperatures (or standard deviations).

Integrate over the result

$$e^{\ln(L(y|\theta))t} = e^{-\frac{1}{2}\chi^2 t} = e^{-\frac{1}{2}\frac{(x-\mu)^2}{\sigma^2}t}$$

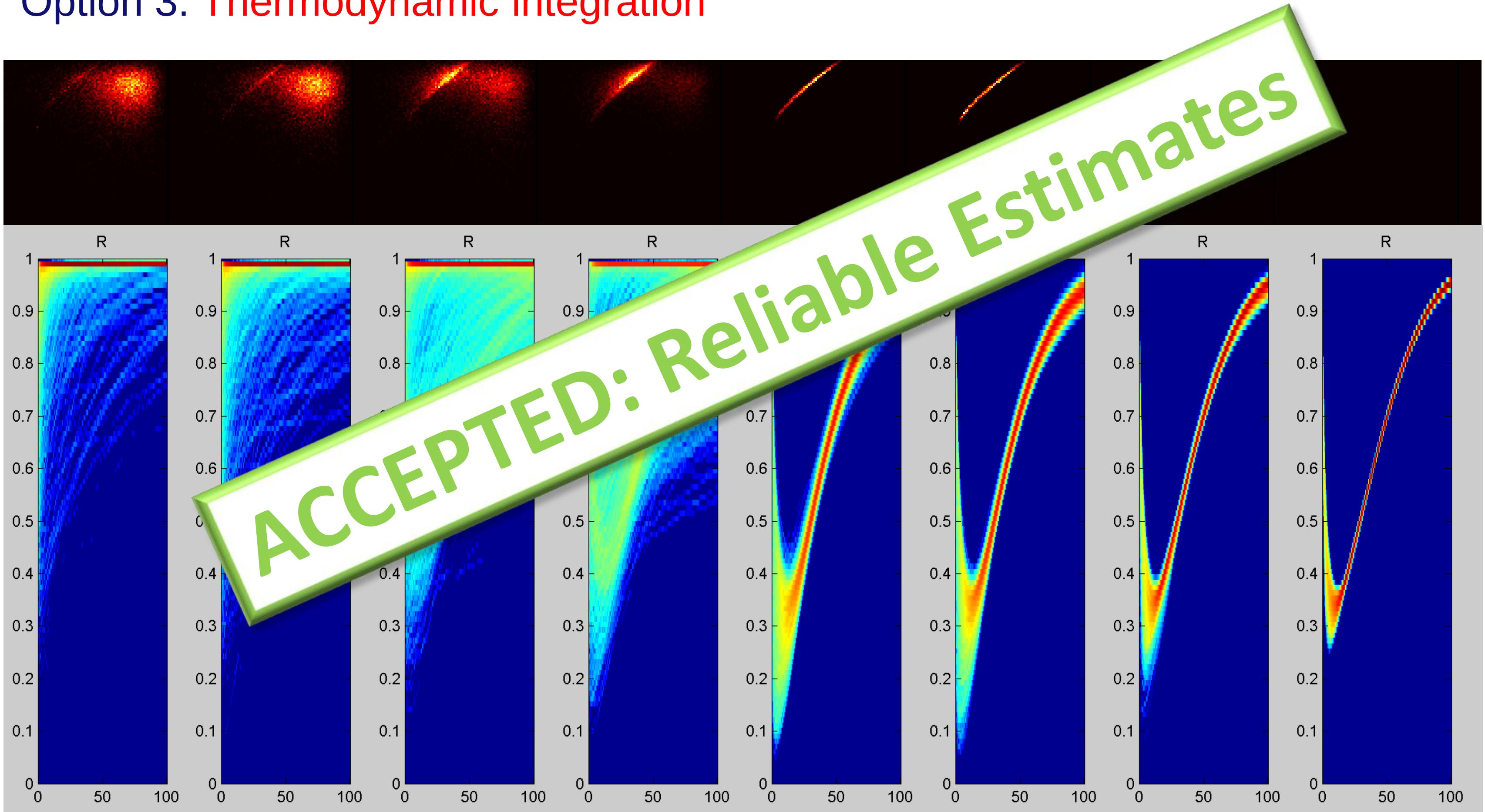
Calculating the horrible integral

Option 3: Thermodynamic integration



Calculating the horrible integral

Option 3: Thermodynamic integration



- **Now we can calculate Bayes Factors**

- **But can we predict what happens upon new measurements?**

- Again, the idea was to use the importance sampling trick

➔ For each temperature we perform importance sampling using the new measurement error model

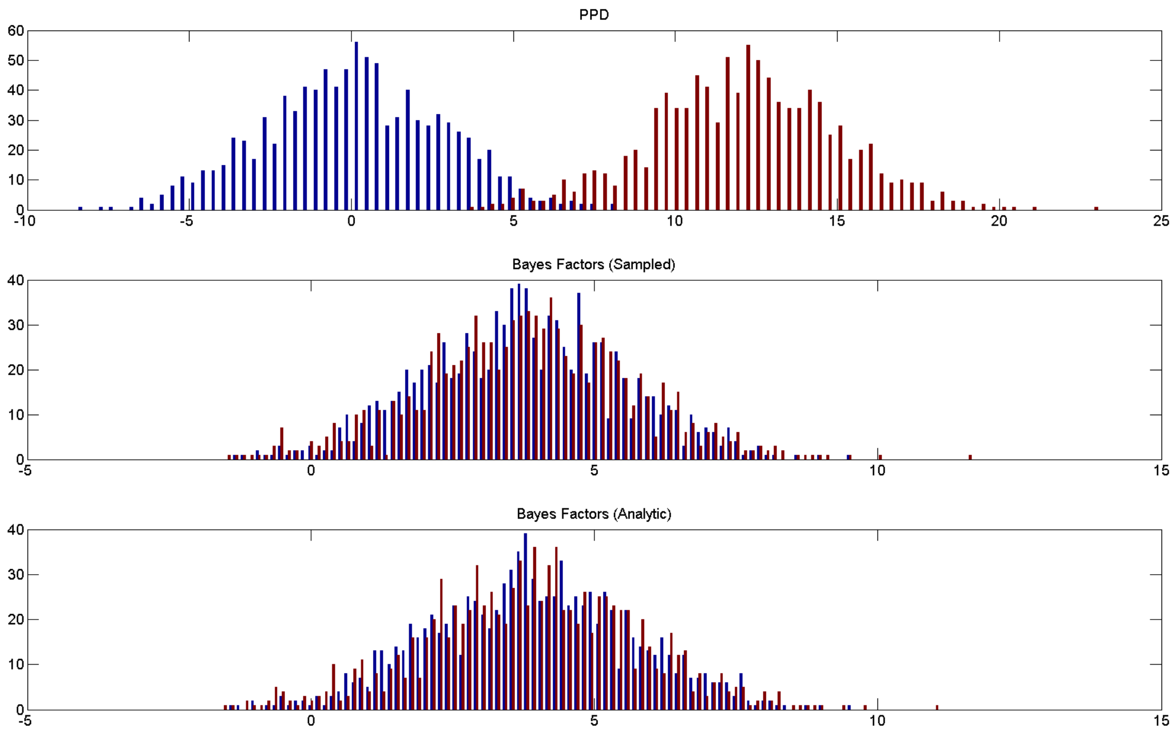
- Again, the idea was to use the importance sampling trick

➔ For each temperature we perform importance sampling using the new measurement error model

➔ Obtain a distribution of Bayes Factors

Distribution of Bayes Factors

- **Assume model 2 is true.**
- **For every possible experimental outcome, we compute a Bayes Factor**

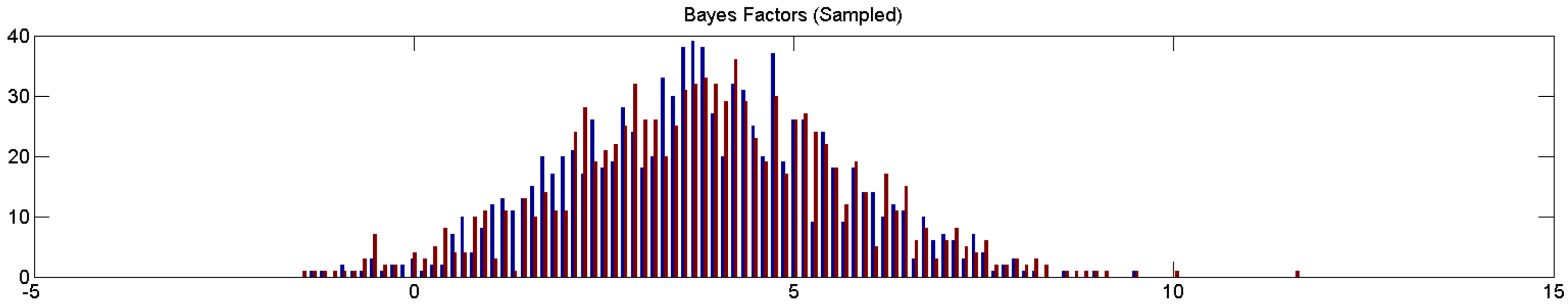


Remember this?

$\log_{10}(B)$	B	Evidence support
0 to 1/2	1 to 3.2	Not worth more than a bare mention
1/2 to 1	3.2 to 10	Substantial
1 to 2	10 to 100	Strong
>2	>100	Decisive

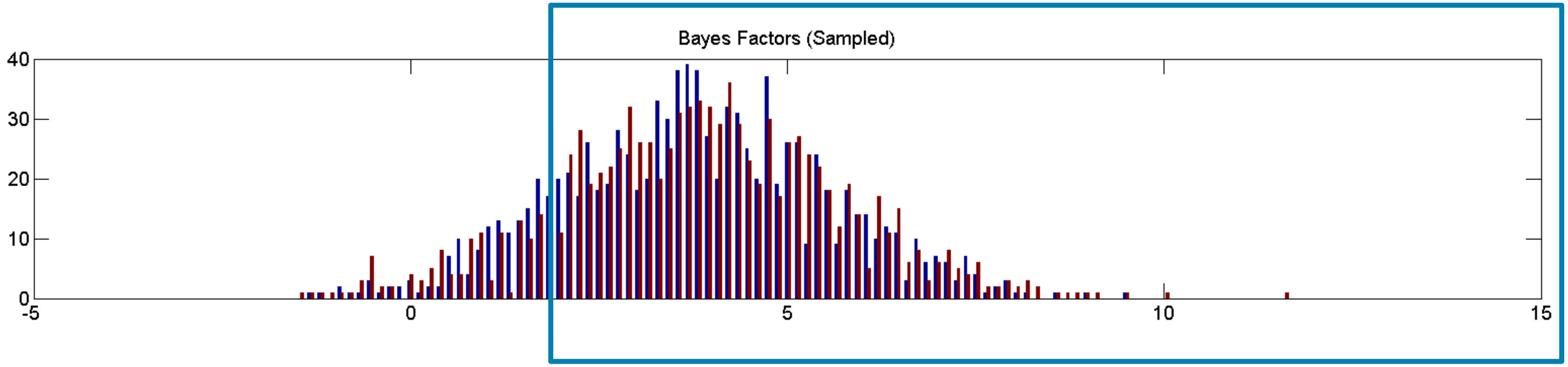
Distribution of Bayes Factors

$\log_{10}(B)$	B	Evidence support
0 to 1/2	1 to 3.2	Not worth more than a bare mention
1/2 to 1	3.2 to 10	Substantial
1 to 2	10 to 100	Strong
>2	>100	Decisive



Distribution of Bayes Factors

$\log_{10}(B)$	B	Evidence support
0 to 1/2	1 to 3.2	Not worth more than a bare mention
1/2 to 1	3.2 to 10	Substantial
1 to 2	10 to 100	Strong
>2	>100	Decisive



Area indicates probability of Model Rejection

Performing the design

- **The next step is to sample the space of possible experiments**

Two major issues

- **Undersampling during self normalisation**

- **Undersampling measurements that are too precise**

Technische Universiteit Eindhoven University of Technology

Why doesn't it work?

- **What happens when PPDs have little overlap**

$$\frac{1}{N}\sum_{i=1}^{N}\frac{p_{new}(\vec{\theta}_i)}{\sum_{j=1}^{N}p_{new}(\vec{\theta}_j)}f(\vec{\theta}_i)$$

Technische Universiteit Eindhoven University of Technology

Why doesn't it work?

- **What happens when PPDs have little overlap**

$$\frac{1}{N}\sum_{i=1}^{N}\frac{p_{new}(\vec{\theta}_i)}{\sum_{j=1}^{N}p_{new}(\vec{\theta}_j)}f(\vec{\theta}_i)$$

As $B_{12} \to \infty$

$$p_{new}(\vec{\theta}_i) \to 0$$

Technische Universiteit **Eindhoven** University of Technology

Why doesn't it work?

- **What happens when PPDs have little overlap**

$$\frac{1}{N}\sum_{i=1}^{N}\frac{p_{new}(\vec{\theta}_i)}{\sum_{j=1}^{N}p_{new}(\vec{\theta}_j)}f(\vec{\theta}_i)$$

As $B_{12}\rightarrow\infty$

$$p_{new}(\vec{\theta}_i)\rightarrow 0$$

$$\frac{1}{\sum_{i=1}^{N}p_{new}(\vec{\theta}_i)}\rightarrow$$

Why doesn't it work?

- **What happens when PPDs have little overlap**

$$\frac{1}{N}\sum_{i=1}^{N}\frac{p_{new}(\vec{\theta}_i)}{\sum_{j=1}^{N}p_{new}(\vec{\theta}_j)}f(\vec{\theta}_i)$$

As $B_{12} \rightarrow \infty$

$$p_{new}(\vec{\theta}_i) \rightarrow 0$$

$$\frac{1}{\sum_{i=1}^{N}p_{new}(\vec{\theta}_i)} \rightarrow$$

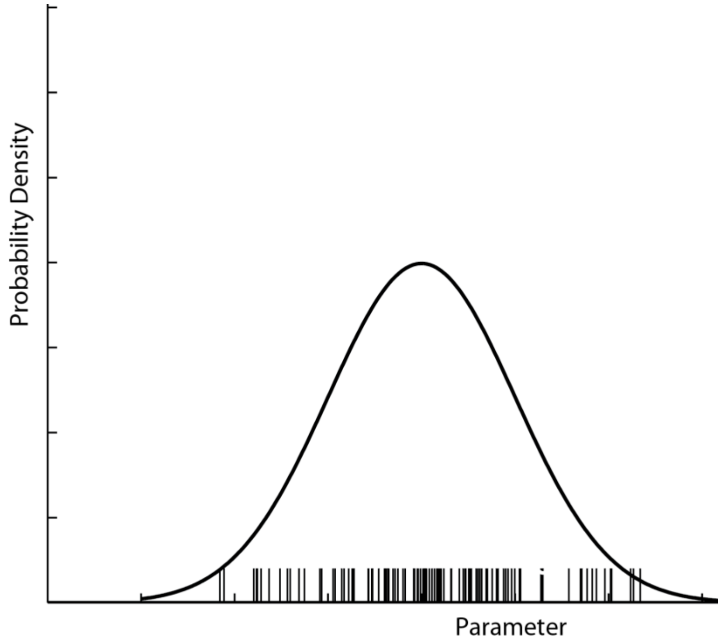
Technische Universiteit Eindhoven University of Technology

Why doesn't it work?

- **Instability of self normalisation**

Why doesn’t it work?

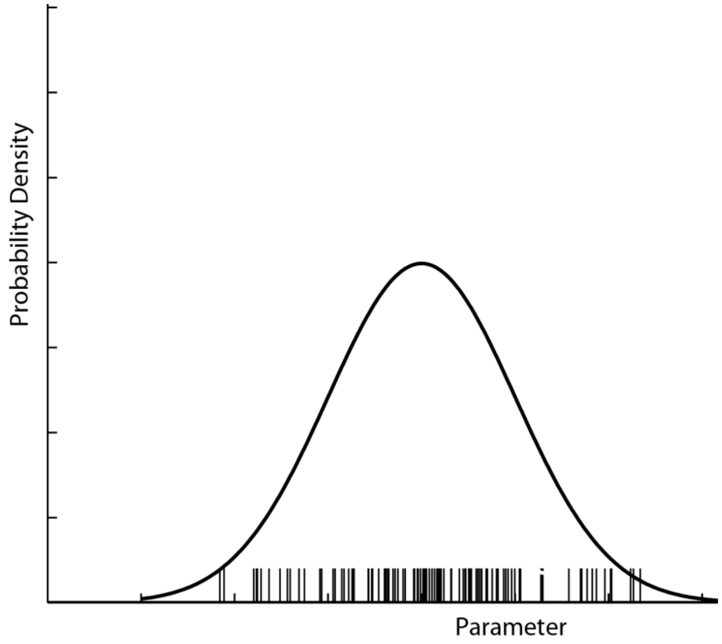
- **Undersampling when experiments are too precise**



Why doesn't it work?

- **Undersampling when experiments are too precise**

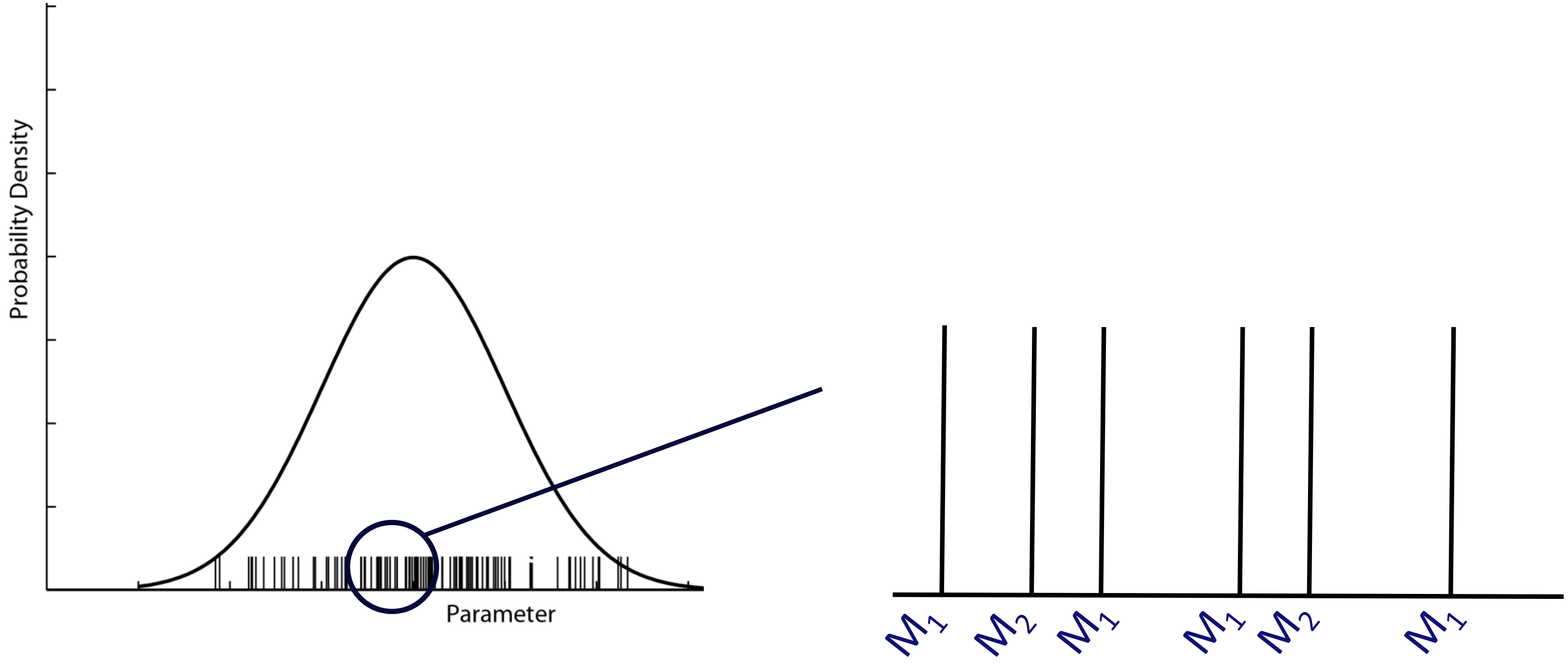
Two models, with identical distributions



Why doesn't it work?

- **Undersampling when experiments are too precise**

Two models, with identical distributions

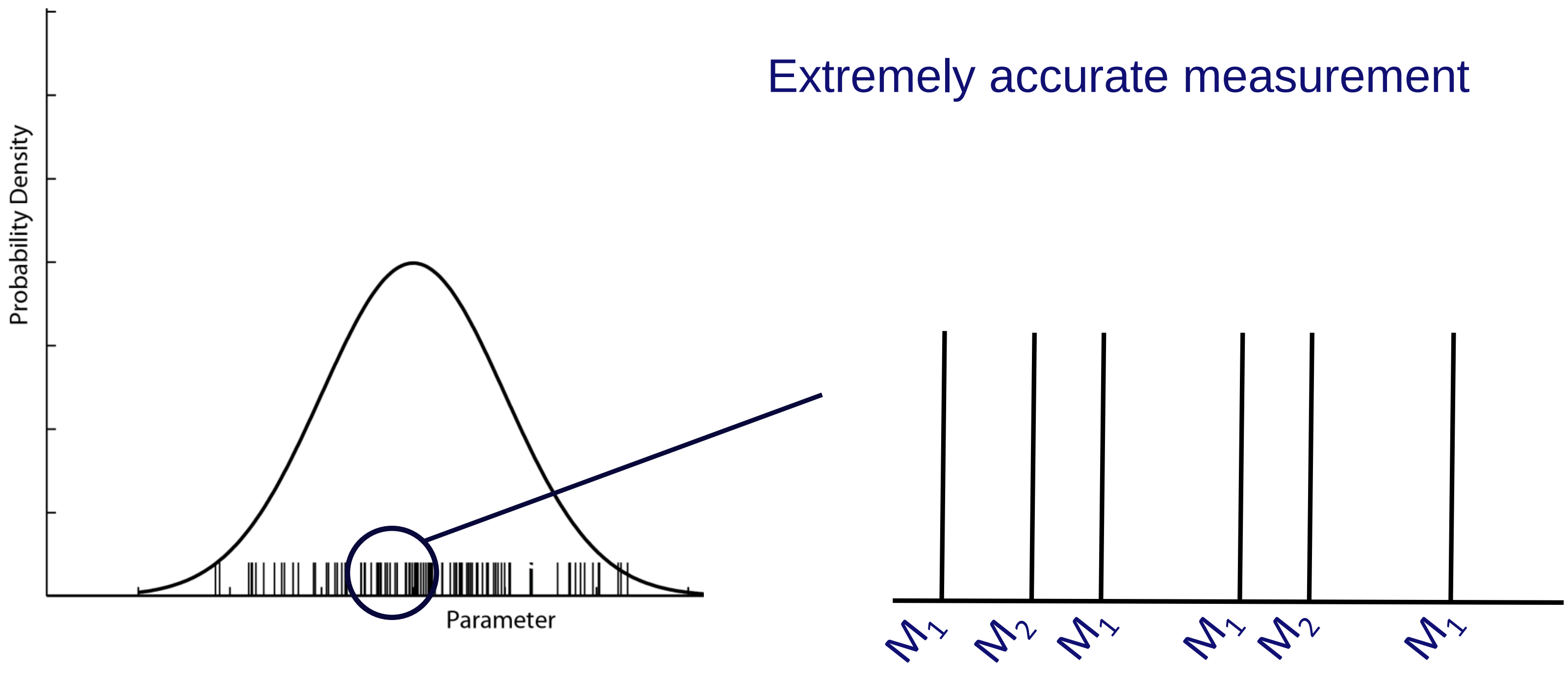


Why doesn't it work?

- **Undersampling when experiments are too precise**

Two models, with identical distributions

Extremely accurate measurement

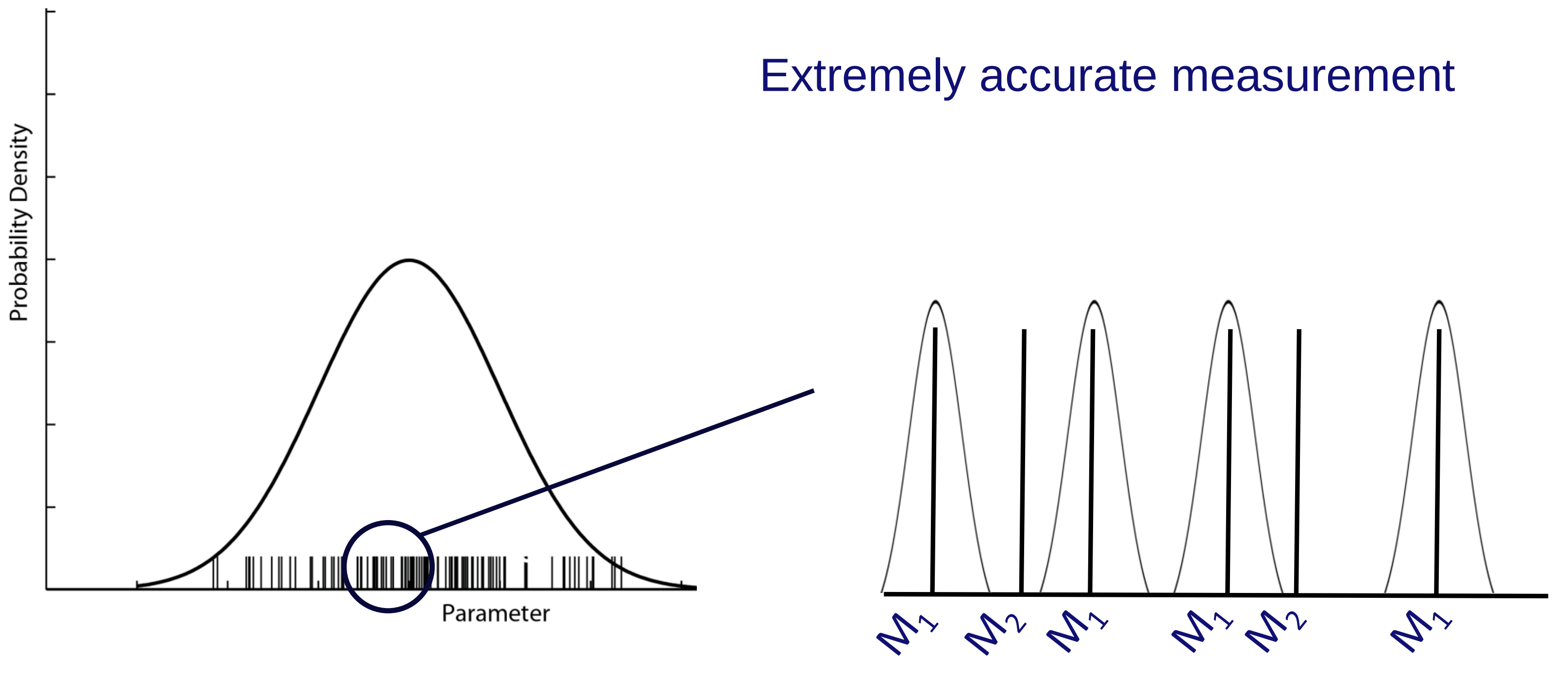


Why doesn’t it work?

- **Undersampling when experiments are too precise**

Two models, with identical distributions

Extremely accurate measurement



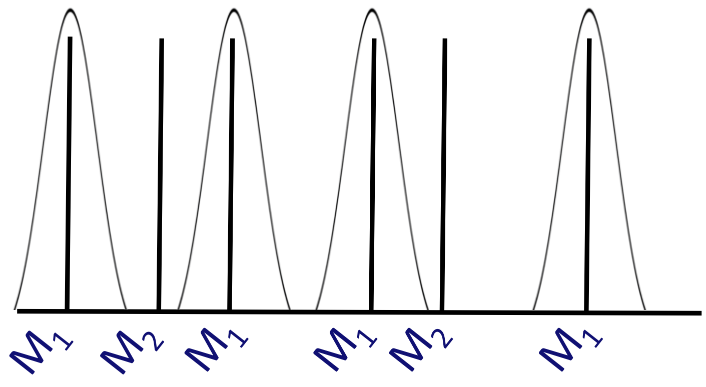
Why doesn't it work?

- **Undersampling when experiments are too precise**

Two models, with identical distributions

Extremely accurate measurement

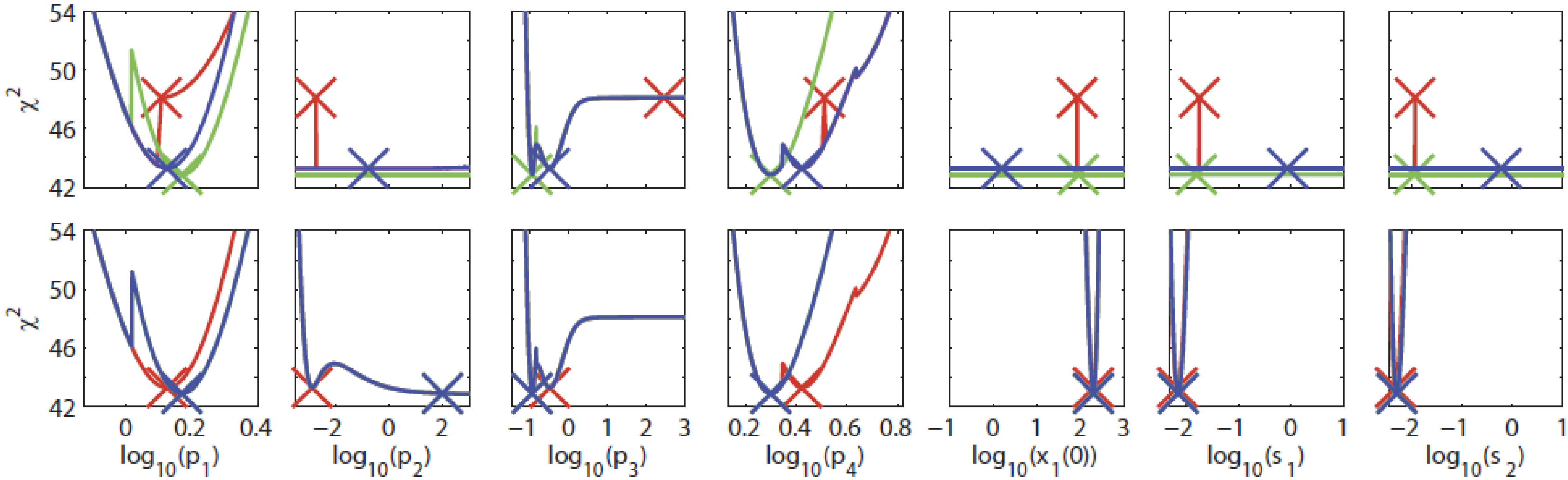
Little **apparent** overlap ➔ **Appears** to be a useful measurement!

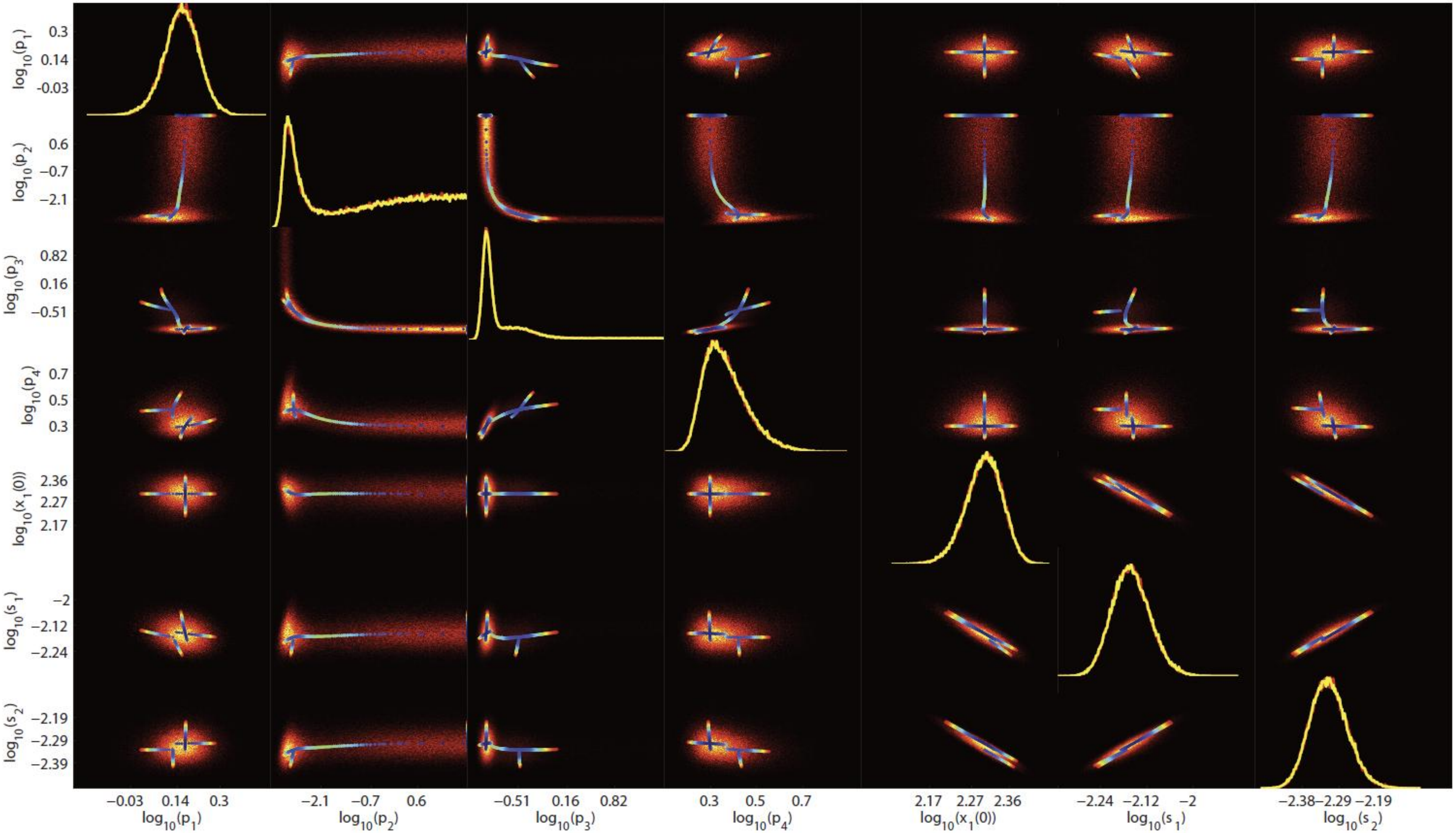


Technische Universiteit
Eindhoven
University of Technology

Possible solutions (not yet explored)

- **Restrict new experiment to Gaussians and approximate PPD by multivariate Gaussian**
- **Thermodynamic integration path from one model to the next rather than from prior to posterior**
- **Approximate distributions using analytic functions**


χ^2
54
50
46
42
0 0.2 0.4
$\log_{10}(p_1)$
−2 0 2
$\log_{10}(p_2)$
−1 0 1 2 3
$\log_{10}(p_3)$
0.2 0.4 0.6 0.8
$\log_{10}(p_4)$
−1 0 1 2 3
$\log_{10}(x_1(0))$
−2 −1 0 1
$\log_{10}(s_1)$
−2 −1 0 1
$\log_{10}(s_2)$



log10(p1)
log10(p2)
log10(p3)
log10(p4)
log10(x1(0))
log10(s1)
log10(s2)
0.3
0.14
-0.03
0.6
−0.7
−2.1
0.82
0.16
−0.51
0.7
0.5
0.3
2.36
2.27
2.17
−2
−2.12
−2.24
−2.19
−2.29
−2.39
−0.03 0.14 0.3
−2.1 −0.7 0.6
−0.51 0.16 0.82
0.3 0.5 0.7
2.17 2.27 2.36
−2.24 −2.12 −2
−2.38−2.29 −2.19