A Guide to Emulation, Implausibility and History Matching, and an Introduction to Designing Future Experiments.

Tuesday Session 2

UNICAMP Workshop on Bayesian Uncertainty Analysis
of Complex Models

Ian Vernon
Department of Mathematical Sciences
Durham University

Work done in collaboration with Michael Goldstein (Dept. Mathematical Sciences), Junli Liu and Keith Lindsey (Biological Sciences), Durham University, UK.

Quick Overview of Tuesday

Bayesian uncertainty analysis of complex models

Quick Overview of Tuesday

Bayesian uncertainty analysis of complex models

- Session 1: a case study in Galaxy formation including
  - Emulation (proxy modelling)

Quick Overview of Tuesday

Bayesian uncertainty analysis of complex models

- Session 1: a case study in Galaxy formation including
  - Emulation (proxy modelling)
  - Implausibility Measures

Quick Overview of Tuesday

Bayesian uncertainty analysis of complex models

- Session 1: a case study in Galaxy formation including
  - Emulation (proxy modelling)
  - Implausibility Measures
  - Iterative History Matching via implausibility

Quick Overview of Tuesday

Bayesian uncertainty analysis of complex models

- Session 1: a case study in Galaxy formation including
  - Emulation (proxy modelling)
  - Implausibility Measures
  - Iterative History Matching via implausibility
  - Visualisation

Quick Overview of Tuesday

Bayesian uncertainty analysis of complex models

- Session 1: a case study in Galaxy formation including
  - Emulation (proxy modelling)
  - Implausibility Measures
  - Iterative History Matching via implausibility
  - Visualisation
- Session 2: Step-by-step guide to
  - Emulation

Quick Overview of Tuesday

Bayesian uncertainty analysis of complex models

- Session 1: a case study in Galaxy formation including
  - Emulation (proxy modelling)
  - Implausibility Measures
  - Iterative History Matching via implausibility
  - Visualisation

- Session 2: Step-by-step guide to
  - Emulation
  - Implausibility Measures

Quick Overview of Tuesday

Bayesian uncertainty analysis of complex models

- Session 1: a case study in Galaxy formation including
  - Emulation (proxy modelling)
  - Implausibility Measures
  - Iterative History Matching via implausibility
  - Visualisation

- Session 2: Step-by-step guide to
  - Emulation
  - Implausibility Measures
  - Iterative History Matching via implausibility

Quick Overview of Tuesday

Bayesian uncertainty analysis of complex models

- Session 1: a case study in Galaxy formation including
  - Emulation (proxy modelling)
  - Implausibility Measures
  - Iterative History Matching via implausibility
  - Visualisation

- Session 2: Step-by-step guide to
  - Emulation
  - Implausibility Measures
  - Iterative History Matching via implausibility
  - Introduction to Designing future experiments: what data should I pay for?

Session 2: Step-by-step guide

- Emulation
  - Simple 1D emulator construction

Session 2: Step-by-step guide

- Emulation
  - Simple 1D emulator construction
  - Applying the Bayes Linear Update

Session 2: Step-by-step guide

- Emulation
  - Simple 1D emulator construction
  - Applying the Bayes Linear Update
  - More complex emulator construction

Session 2: Step-by-step guide

- Emulation
  - Simple 1D emulator construction
  - Applying the Bayes Linear Update
  - More complex emulator construction
- Implausibility Measures
  - Simple 1D construction

Session 2: Step-by-step guide

- Emulation
  - Simple 1D emulator construction
  - Applying the Bayes Linear Update
  - More complex emulator construction
- Implausibility Measures
  - Simple 1D construction
  - Combining implausibilities from several outputs

Session 2: Step-by-step guide

- Emulation
  - Simple 1D emulator construction
  - Applying the Bayes Linear Update
  - More complex emulator construction
- Implausibility Measures
  - Simple 1D construction
  - Combining implausibilities from several outputs
- Iterative History Matching via implausibility
  - Imposing cutoffs

Session 2: Step-by-step guide

- Emulation
  - Simple 1D emulator construction
  - Applying the Bayes Linear Update
  - More complex emulator construction
- Implausibility Measures
  - Simple 1D construction
  - Combining implausibilities from several outputs
- Iterative History Matching via implausibility
  - Imposing cutoffs
  - Performing more waves

Session 2: Step-by-step guide

- Emulation
  - Simple 1D emulator construction
  - Applying the Bayes Linear Update
  - More complex emulator construction
- Implausibility Measures
  - Simple 1D construction
  - Combining implausibilities from several outputs
- Iterative History Matching via implausibility
  - Imposing cutoffs
  - Performing more waves
- Introduction to Designing future experiments: what data should I pay for?

Guide to Emulation

- As we have discussed, complex computer models are in general very slow.

Guide to Emulation

- As we have discussed, complex computer models are in general very slow.
- Many calculations that take account of uncertainty need huge numbers of model evaluations.

Guide to Emulation

- As we have discussed, complex computer models are in general very slow.
- Many calculations that take account of uncertainty need huge numbers of model evaluations.
- We have seen History Matching, but forecasting, making decisions to optimise certain outputs, designing new experiments etc. require even more model evaluations.

Guide to Emulation

- As we have discussed, complex computer models are in general very slow.
- Many calculations that take account of uncertainty need huge numbers of model evaluations.
- We have seen History Matching, but forecasting, making decisions to optimise certain outputs, designing new experiments etc. require even more model evaluations.
- Emulators, which represent our beliefs about the computer model, are very fast.

Guide to Emulation

- As we have discussed, complex computer models are in general very slow.
- Many calculations that take account of uncertainty need huge numbers of model evaluations.
- We have seen History Matching, but forecasting, making decisions to optimise certain outputs, designing new experiments etc. require even more model evaluations.
- Emulators, which represent our beliefs about the computer model, are very fast.
- Hence Emulators are essential for any serious analysis.

Guide to Emulation

- As we have discussed, complex computer models are in general very slow.
- Many calculations that take account of uncertainty need huge numbers of model evaluations.
- We have seen History Matching, but forecasting, making decisions to optimise certain outputs, designing new experiments etc. require even more model evaluations.
- Emulators, which represent our beliefs about the computer model, are very fast.
- Hence Emulators are essential for any serious analysis.
- We will now build a simple emulator.

Full Emulator Equation

- For each output $f_i(x)$ we pick active variables x^A then emulate using:

$$f_i(x) = \sum_j \beta_{ij}\, g_{ij}(x^A) + u_i(x^A) + \delta_i(x)$$

Full Emulator Equation

- For each output $f_i(x)$ we pick active variables x^A then emulate using:

$$f_i(x) = \sum_j \beta_{ij}\, g_{ij}(x^A) + u_i(x^A) + \delta_i(x)$$

- The $\sum_j \beta_{ij}\, g_{ij}(x^A)$ is a 3rd order polynomial (say) in the active inputs.

Full Emulator Equation

- For each output $f_i(x)$ we pick active variables x^A then emulate using:

$$f_i(x) = \sum_j \beta_{ij}\, g_{ij}(x^A) + u_i(x^A) + \delta_i(x)$$

- The $\sum_j \beta_{ij}\, g_{ij}(x^A)$ is a 3rd order polynomial (say) in the active inputs.
- $u_i(x^A)$ is a Gaussian process.

Full Emulator Equation

- For each output $f_i(x)$ we pick active variables x^A then emulate using:

$$f_i(x) = \sum_j \beta_{ij}\, g_{ij}(x^A) + u_i(x^A) + \delta_i(x)$$

- The $\sum_j \beta_{ij}\, g_{ij}(x^A)$ is a 3rd order polynomial (say) in the active inputs.
- $u_i(x^A)$ is a Gaussian process.
- The nugget $\delta_i(x)$ models the effects of inactive variables as random noise.

Full Emulator Equation

- For each output $f_i(x)$ we pick active variables x^A then emulate using:

$$f_i(x) = \sum_j \beta_{ij}\, g_{ij}(x^A) + u_i(x^A) + \delta_i(x)$$

- The $\sum_j \beta_{ij}\, g_{ij}(x^A)$ is a 3rd order polynomial (say) in the active inputs.
- $u_i(x^A)$ is a Gaussian process.
- The nugget $\delta_i(x)$ models the effects of inactive variables as random noise.
- The $u_i(x^A)$ have covariance structure given by:

$$\mathrm{Cov}(u_i(x_1^A), u_i(x_2^A)) \;=\; \sigma_i^2 \exp[-|x_1^A - x_2^A|^2/\theta_i^2]$$

Full Emulator Equation

- For each output $f_i(x)$ we pick active variables x^A then emulate using:

$$f_i(x) = \sum_j \beta_{ij}\, g_{ij}(x^A) + u_i(x^A) + \delta_i(x)$$

- The $\sum_j \beta_{ij}\, g_{ij}(x^A)$ is a 3rd order polynomial (say) in the active inputs.
- $u_i(x^A)$ is a Gaussian process.
- The nugget $\delta_i(x)$ models the effects of inactive variables as random noise.
- The $u_i(x^A)$ have covariance structure given by:

$$\mathrm{Cov}(u_i(x_1^A), u_i(x_2^A)) \;=\; \sigma_i^2 \exp[-|x_1^A - x_2^A|^2/\theta_i^2]$$

- The Emulators give the expectation $\mathsf{E}[f_i(x)]$ and variance $\mathsf{Var}(f_i(x))$ at point x for each output given by $i = 1, .., 11$, and are fast to evaluate.

Simple Emulator Equation

- Lets consider a 1-dimensional example for simplicity, so $f(x) = f_1(x)$ and x are both 1-dimensional.

Simple Emulator Equation

- Lets consider a 1-dimensional example for simplicity, so $f(x) = f_1(x)$ and x are both 1-dimensional.
- Therefore we don't consider active and inactive inputs (as there is just one), so we can drop the nugget term $\delta_i(x)$ and forget about x^A and the index i.

Simple Emulator Equation

- Lets consider a 1-dimensional example for simplicity, so $f(x) = f_1(x)$ and x are both 1-dimensional.
- Therefore we don't consider active and inactive inputs (as there is just one), so we can drop the nugget term $\delta_i(x)$ and forget about x^A and the index i.
- So the emulator equation reduces to

$$f(x) = \sum_j \beta_j \, g_j(x) + u(x)$$

Simple Emulator Equation

- Lets consider a 1-dimensional example for simplicity, so $f(x) = f_1(x)$ and x are both 1-dimensional.
- Therefore we don't consider active and inactive inputs (as there is just one), so we can drop the nugget term $\delta_i(x)$ and forget about x^A and the index i.
- So the emulator equation reduces to

$$f(x) = \sum_j \beta_j \, g_j(x) + u(x)$$

- The first term is a regression term, often composed of low order polynomials in x (or other appropriate deterministic functions), so

$$\sum_j \beta_j \, g_j(x) \;=\; \beta_0 + \beta_1 x + \beta_2 x^2 + \beta_3 x^3 + \ldots$$

Simple Emulator Equation

- Lets consider a 1-dimensional example for simplicity, so $f(x) = f_1(x)$ and x are both 1-dimensional.
- Therefore we don't consider active and inactive inputs (as there is just one), so we can drop the nugget term $\delta_i(x)$ and forget about x^A and the index i.
- So the emulator equation reduces to

$$f(x) = \sum_j \beta_j \, g_j(x) + u(x)$$

- The first term is a regression term, often composed of low order polynomials in x (or other appropriate deterministic functions), so

$$\sum_j \beta_j \, g_j(x) \;=\; \beta_0 + \beta_1 x + \beta_2 x^2 + \beta_3 x^3 + \ldots$$

- To really simplify things, lets set this polynomial to a constant β_0 (we will return to the polynomial later). Therefore we have:

$$f(x) = \beta_0 + u(x)$$

Simple Emulator Equation

- So we have that: $f(x) = \beta_0 + u(x)$
- As we are going to use Bayes Linear methods to build our emulator, we need to specify a priori the $\mathrm{E}(\beta_0), \mathrm{Var}(\beta_0), \mathrm{E}(u(x))$ and the covariance structure of $u(x)$ which simply means the $\mathrm{Cov}(u(x), u(x'))$.

Simple Emulator Equation

- So we have that: $f(x) = \beta_0 + u(x)$
- As we are going to use Bayes Linear methods to build our emulator, we need to specify a priori the $\mathrm{E}(\beta_0), \mathrm{Var}(\beta_0), \mathrm{E}(u(x))$ and the covariance structure of $u(x)$ which simply means the $\mathrm{Cov}(u(x), u(x'))$.
- We are free to set $\mathrm{E}(u(x)) = 0$, as we can always redefine β_0.

Simple Emulator Equation

- So we have that: $f(x) = \beta_0 + u(x)$
- As we are going to use Bayes Linear methods to build our emulator, we need to specify a priori the $\mathrm{E}(\beta_0), \mathrm{Var}(\beta_0), \mathrm{E}(u(x))$ and the covariance structure of $u(x)$ which simply means the $\mathrm{Cov}(u(x), u(x'))$.
- We are free to set $\mathrm{E}(u(x)) = 0$, as we can always redefine β_0.
- We then choose an appropriate covariance structure depending what we suspect the behaviour of the computer model $f(x)$ to be: if it is pretty smooth as a function of x we may choose the Gaussian form:

$$\mathrm{Cov}(u(x), u(x')) \quad = \quad \sigma^2 \exp[-|x - x'|^2/\theta^2]$$

where σ^2 and θ are parameters we should specify.

Simple Emulator Equation

- So we have that: $f(x) = \beta_0 + u(x)$
- As we are going to use Bayes Linear methods to build our emulator, we need to specify a priori the $\mathrm{E}(\beta_0), \mathrm{Var}(\beta_0), \mathrm{E}(u(x))$ and the covariance structure of $u(x)$ which simply means the $\mathrm{Cov}(u(x), u(x'))$.
- We are free to set $\mathrm{E}(u(x)) = 0$, as we can always redefine β_0.
- We then choose an appropriate covariance structure depending what we suspect the behaviour of the computer model $f(x)$ to be: if it is pretty smooth as a function of x we may choose the Gaussian form:

$$\mathrm{Cov}(u(x), u(x')) \quad = \quad \sigma^2 \exp[-|x - x'|^2/\theta^2]$$

  where σ^2 and θ are parameters we should specify.
- In this case we specify $\mathrm{E}(\beta_0)$ corresponding to our beliefs about the mean of $f(x)$, with $\mathrm{Var}(\beta_0)$ representing how unsure we are.

Simple 1-dimensional Sine Example

- Remember that $f(x)$ could represent any output of a complex model, so for a reservoir model, it could be the BHP or oil production at 1000 days, while x would be a vector of inputs e.g. porosities, permeabilities, fault multipliers.

Simple 1-dimensional Sine Example

- Remember that $f(x)$ could represent any output of a complex model, so for a reservoir model, it could be the BHP or oil production at 1000 days, while x would be a vector of inputs e.g. porosities, permeabilities, fault multipliers.

- However, we will stick to a simple 1-dimensional example where

$$f(x) = \sin\left(\frac{2\pi x}{50}\right)$$

Simple 1-dimensional Sine Example

- Remember that $f(x)$ could represent any output of a complex model, so for a reservoir model, it could be the BHP or oil production at 1000 days, while x would be a vector of inputs e.g. porosities, permeabilities, fault multipliers.

- However, we will stick to a simple 1-dimensional example where

$$f(x) \;=\; \sin\left(\frac{2\pi x}{50}\right)$$

- We imagine that due to computer time constraints we can only evaluate $f(x)$ at 6 points and choose the (not ideal) locations:

$$x^{(j)} \;=\; 0, 10, 20, 30, 43, 50$$

Simple 1-dimensional Sine Example

- Remember that $f(x)$ could represent any output of a complex model, so for a reservoir model, it could be the BHP or oil production at 1000 days, while x would be a vector of inputs e.g. porosities, permeabilities, fault multipliers.

- However, we will stick to a simple 1-dimensional example where

$$f(x) \;=\; \sin\left(\frac{2\pi x}{50}\right)$$

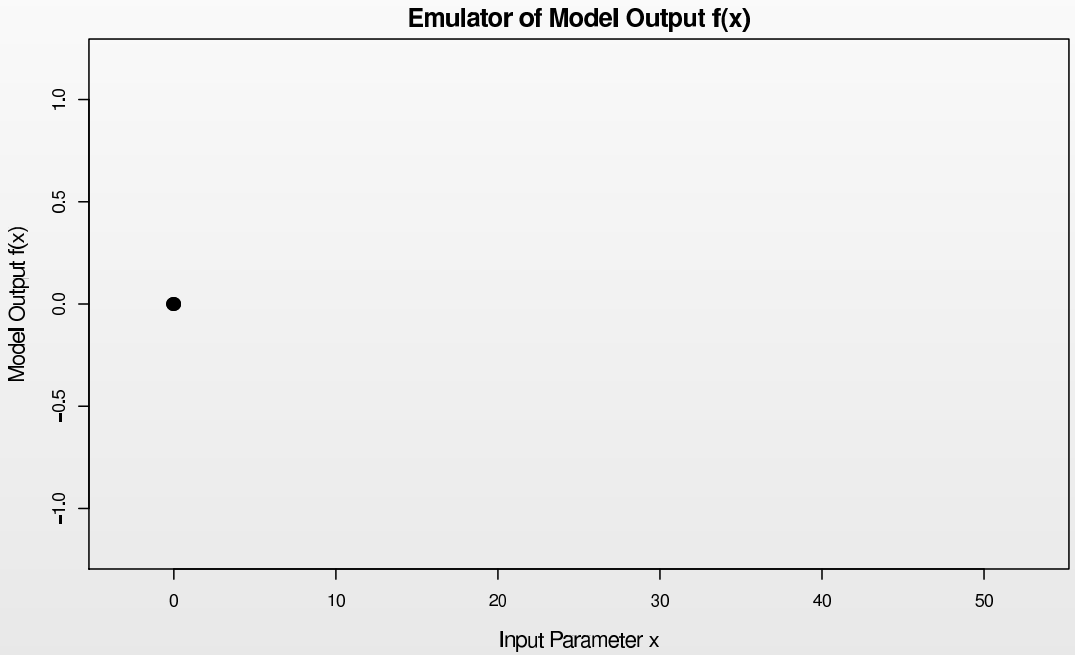
- We imagine that due to computer time constraints we can only evaluate $f(x)$ at 6 points and choose the (not ideal) locations:

$$x^{(j)} \;=\; 0, 10, 20, 30, 43, 50$$

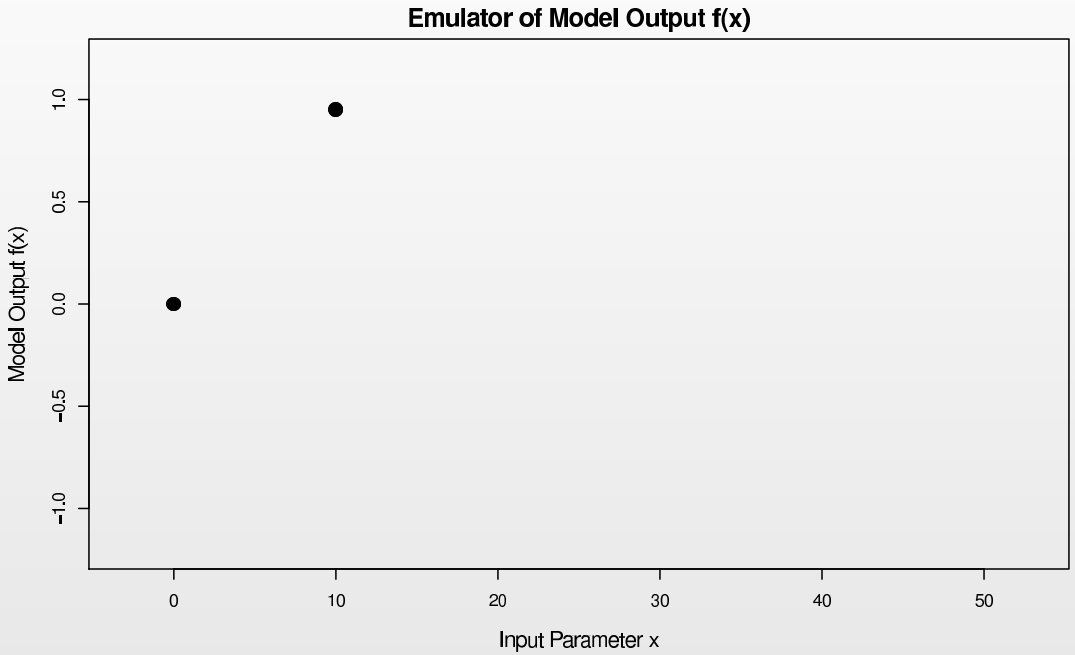
- We evaluate $f(x^{(j)})$ for $j = 1$ to 6, and gather the results into vector D:

$$\begin{aligned} D \;&= (f(x^{(1)}), f(x^{(2)}), \ldots, f(x^{(6)}))^T \\ &= \left(\sin\left(\frac{2\pi \times 0}{50}\right), \sin\left(\frac{2\pi \times 10}{50}\right), \ldots, \sin\left(\frac{2\pi \times 50}{50}\right)\right)^T \end{aligned}$$

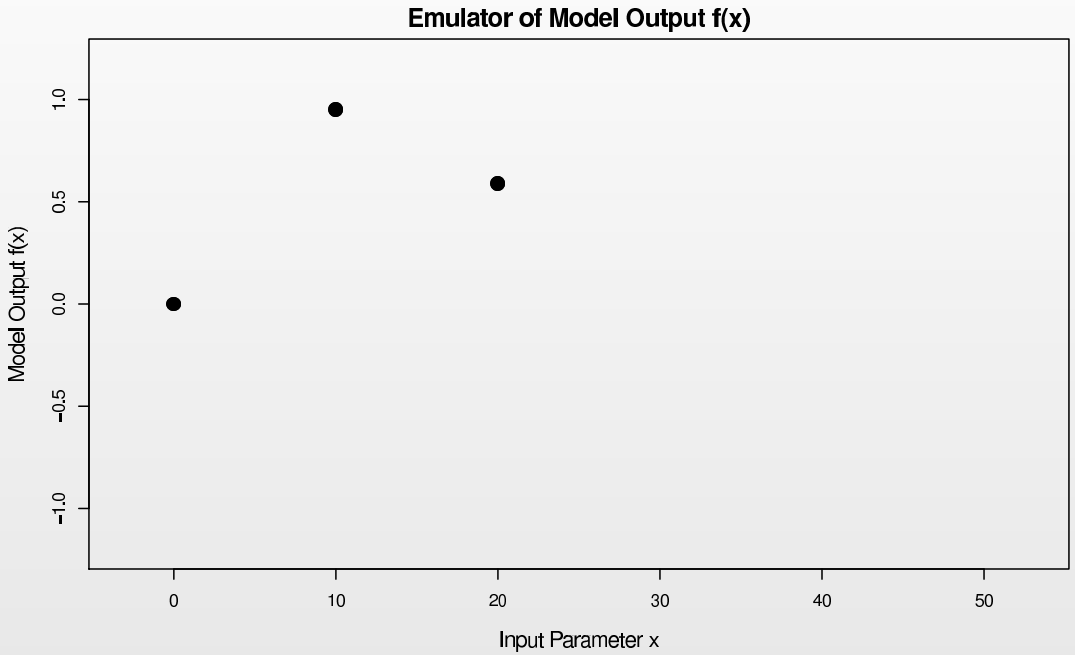
Emulation: a 1D Example



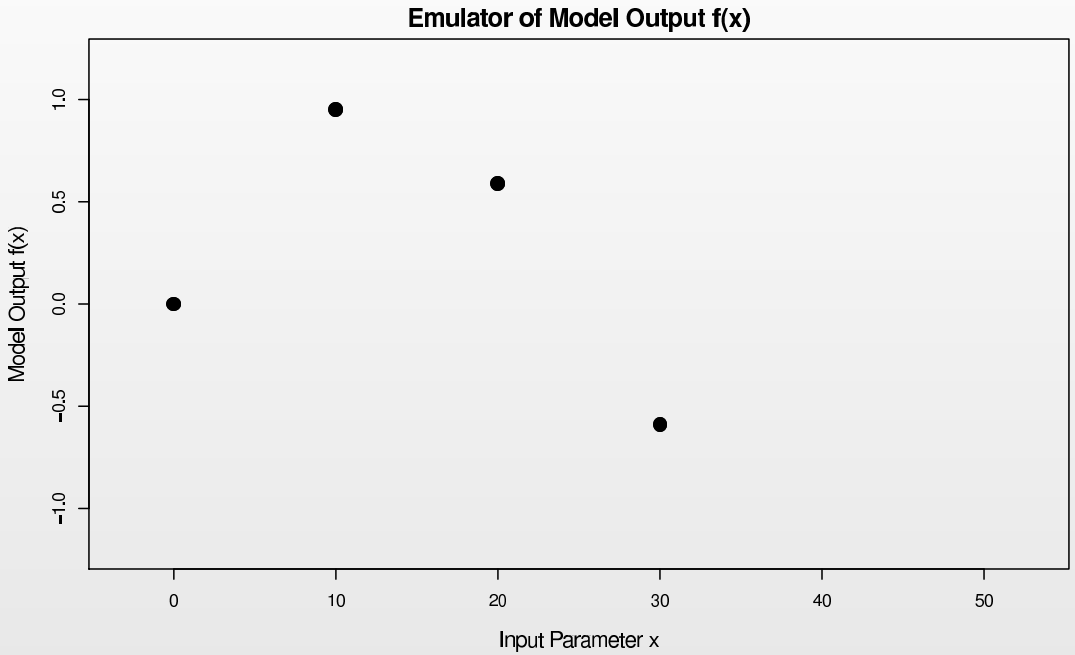
Emulation: a 1D Example



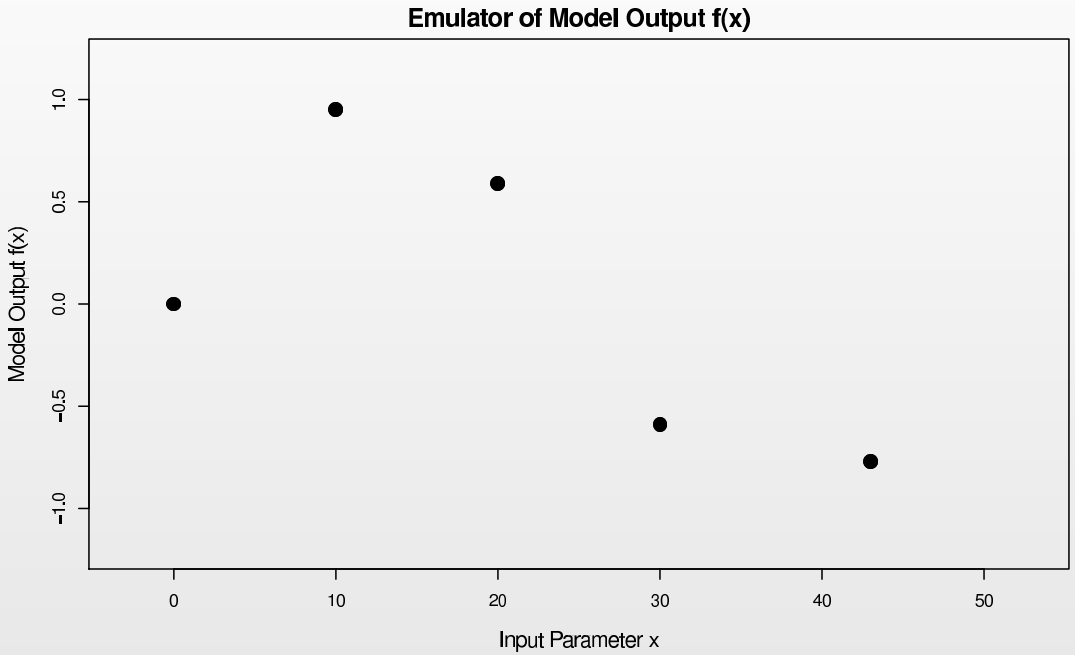
Emulation: a 1D Example



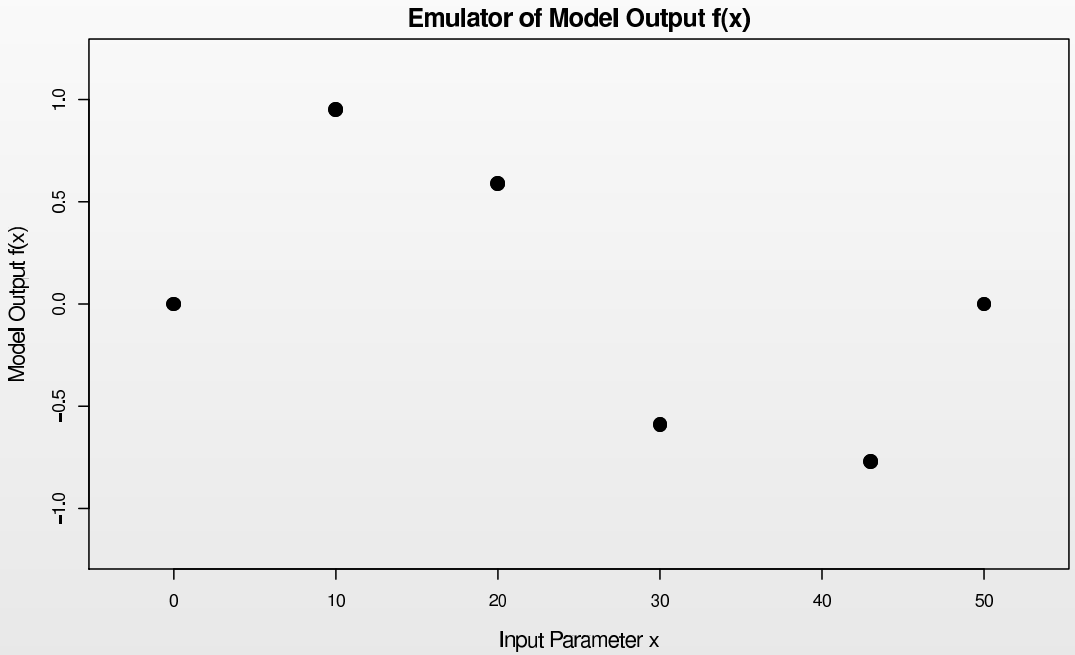
Emulation: a 1D Example



Emulation: a 1D Example



Emulation: a 1D Example



Simple 1-dimensional Sine Example

- For simplicity we treat the constant term β_0 as known and hence set $\mathrm{Var}(\beta_0) = 0$, and choose the prior expectation $\mathrm{E}(\beta_0) = 0$ (a sensible choice for a sine function).

Simple 1-dimensional Sine Example

- For simplicity we treat the constant term β_0 as known and hence set $\mathrm{Var}(\beta_0) = 0$, and choose the prior expectation $\mathrm{E}(\beta_0) = 0$ (a sensible choice for a sine function).

- We choose the parameters in the covariance function for $u(x)$ to be $\sigma = 1$ and $\theta = 15$ representing curves of moderate smoothness (over the range of x values $[0, 50]$).

Simple 1-dimensional Sine Example

- For simplicity we treat the constant term β_0 as known and hence set $\mathrm{Var}(\beta_0) = 0$, and choose the prior expectation $\mathrm{E}(\beta_0) = 0$ (a sensible choice for a sine function).

- We choose the parameters in the covariance function for $u(x)$ to be $\sigma = 1$ and $\theta = 15$ representing curves of moderate smoothness (over the range of x values $[0, 50]$).

- We now have all we need to use the Bayes Linear equations to update our prior beliefs $\mathrm{E}(f(x))$ and $\mathrm{Var}(f(x))$ about the behaviour of $f(x)$ given new run data D, to obtain adjusted beliefs $\mathrm{E}_D(f(x))$ and $\mathrm{Var}_D(f(x))$:

$$\mathrm{E}_D(f(x)) = \mathrm{E}(f(x)) + \mathrm{Cov}(f(x), D)\mathrm{Var}(D)^{-1}(D - \mathrm{E}(D))$$
$$\mathrm{Var}_D(f(x)) = \mathrm{Var}(f(x)) - \mathrm{Cov}(f(x), D)\mathrm{Var}(D)^{-1}\mathrm{Cov}(D, f(x))$$

Simple 1-dimensional Sine Example

- Everything on the right hand side of the BL update equations has been specified: as $f(x) = \beta_0 + u(x)$, $\beta_0 = 0$, $\sigma = 1$ and $\theta = 15$, we have:

$$\mathrm{E}(f(x)) = \beta_0, \qquad \mathrm{Var}(f(x)) = \sigma^2$$
$$\mathrm{E}(D) = (\beta_0, \ldots, \beta_0)^T$$

Simple 1-dimensional Sine Example

- Everything on the right hand side of the BL update equations has been specified: as $f(x) = \beta_0 + u(x)$, $\beta_0 = 0$, $\sigma = 1$ and $\theta = 15$, we have:

$$\mathrm{E}(f(x)) = \beta_0, \qquad \mathrm{Var}(f(x)) = \sigma^2$$
$$\mathrm{E}(D) = (\beta_0, \ldots, \beta_0)^T$$

- $\mathrm{Cov}(f(x), D)$ is a row vector of length $n = 6$ with jth component

$$\begin{aligned} \mathrm{Cov}(f(x), D)_j &= \mathrm{Cov}(f(x), f(x^{(j)})) \\ &= \sigma^2 \exp\left\{ -\frac{\|x - x^{(j)}\|^2}{\theta^2} \right\} \end{aligned}$$

Simple 1-dimensional Sine Example

- Everything on the right hand side of the BL update equations has been specified: as $f(x) = \beta_0 + u(x)$, $\beta_0 = 0$, $\sigma = 1$ and $\theta = 15$, we have:

$$\begin{aligned} \mathrm{E}(f(x)) &= \beta_0, \qquad \mathrm{Var}(f(x)) = \sigma^2 \\ \mathrm{E}(D) &= (\beta_0, \ldots, \beta_0)^T \end{aligned}$$

- $\mathrm{Cov}(f(x), D)$ is a row vector of length $n = 6$ with jth component

$$\begin{aligned} \mathrm{Cov}(f(x), D)_j &= \mathrm{Cov}(f(x), f(x^{(j)})) \\ &= \sigma^2 \exp\left\{ -\frac{\|x - x^{(j)}\|^2}{\theta^2} \right\} \end{aligned}$$

- Similarly $\mathrm{Var}(D)$ is an $n \times n$ matrix with (j, k) element

$$\begin{aligned} \mathrm{Var}(D)_{jk} &= \mathrm{Cov}(f(x^{(j)}), f(x^{(k)})) \\ &= \sigma^2 \exp\left\{ -\frac{\|x^{(j)} - x^{(k)}\|^2}{\theta^2} \right\} \end{aligned}$$

Simple 1-dimensional Sine Example

- Note that we can do the BL update for $f(x)$ for a single new untried input x, or we can simultaneously update for a set of m new inputs, where x would now become a vector of length m.

Simple 1-dimensional Sine Example

- Note that we can do the BL update for $f(x)$ for a single new untried input x, or we can simultaneously update for a set of m new inputs, where x would now become a vector of length m.

- Now $f(x)$ is a column vector of length m.

Simple 1-dimensional Sine Example

- Note that we can do the BL update for $f(x)$ for a single new untried input x, or we can simultaneously update for a set of m new inputs, where x would now become a vector of length m.
- Now $f(x)$ is a column vector of length m.
- In this case we use the same BL equations, but use $\mathrm{Cov}(f(x), D)$ which is now an $m \times n$ matrix, $\mathrm{Var}(f(x))$ is an $m \times m$ matrix, and $\mathrm{E}(f(x))$ is a column vector of length m.

Simple 1-dimensional Sine Example

- Note that we can do the BL update for $f(x)$ for a single new untried input x, or we can simultaneously update for a set of m new inputs, where x would now become a vector of length m.
- Now $f(x)$ is a column vector of length m.
- In this case we use the same BL equations, but use $\mathrm{Cov}(f(x), D)$ which is now an $m \times n$ matrix, $\mathrm{Var}(f(x))$ is an $m \times m$ matrix, and $\mathrm{E}(f(x))$ is a column vector of length m.
- You will try R code this afternoon that does this example with $n = 6$ and $m = 100$.

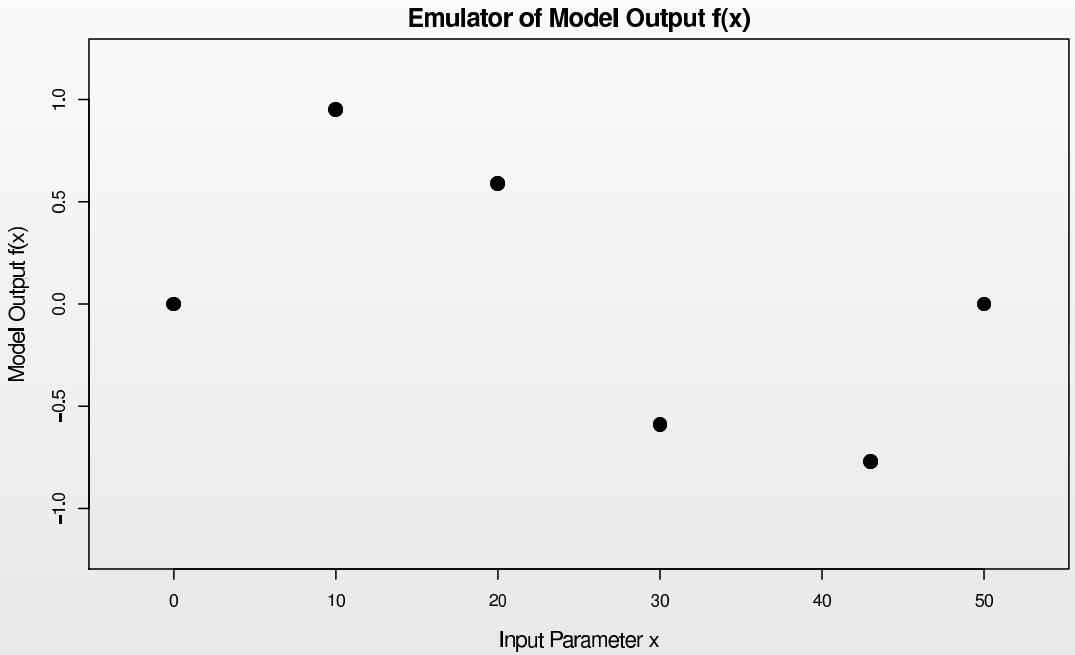
Simple 1-dimensional Sine Example

- Note that we can do the BL update for $f(x)$ for a single new untried input x, or we can simultaneously update for a set of m new inputs, where x would now become a vector of length m.
- Now $f(x)$ is a column vector of length m.
- In this case we use the same BL equations, but use $\mathrm{Cov}(f(x), D)$ which is now an $m \times n$ matrix, $\mathrm{Var}(f(x))$ is an $m \times m$ matrix, and $\mathrm{E}(f(x))$ is a column vector of length m.
- You will try R code this afternoon that does this example with $n = 6$ and $m = 100$.
- Our emulator is complete and is represented as the adjusted expectation and variance of $f(x)$: $\mathrm{E}_D(f(x))$ and $\mathrm{Var}_D(f(x))$.

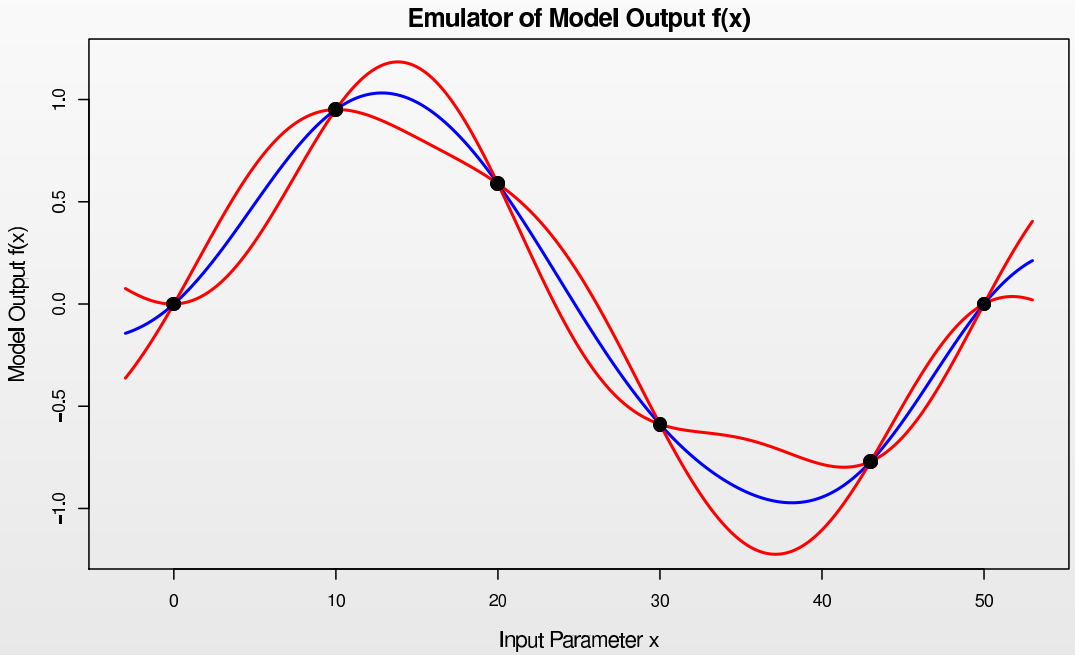
Simple 1-dimensional Sine Example

- Note that we can do the BL update for $f(x)$ for a single new untried input x, or we can simultaneously update for a set of m new inputs, where x would now become a vector of length m.
- Now $f(x)$ is a column vector of length m.
- In this case we use the same BL equations, but use $\mathrm{Cov}(f(x), D)$ which is now an $m \times n$ matrix, $\mathrm{Var}(f(x))$ is an $m \times m$ matrix, and $\mathrm{E}(f(x))$ is a column vector of length m.
- You will try R code this afternoon that does this example with $n = 6$ and $m = 100$.
- Our emulator is complete and is represented as the adjusted expectation and variance of $f(x)$: $\mathrm{E}_D(f(x))$ and $\mathrm{Var}_D(f(x))$.
- We can now plot $\mathrm{E}_D(f(x))$ as a function of x (in blue), and plot credible intervals (in red) as $\mathrm{E}_D(f(x)) \pm 3\sqrt{\mathrm{Var}_D(f(x))}$.

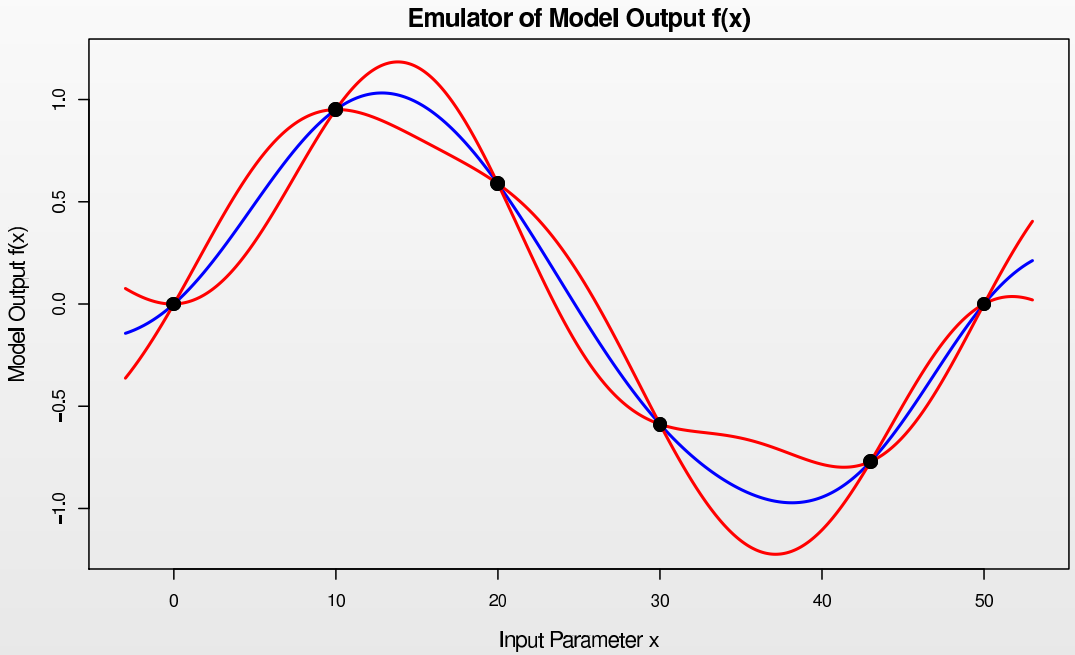
Emulation: a 1D Example



Emulation: a 1D Example

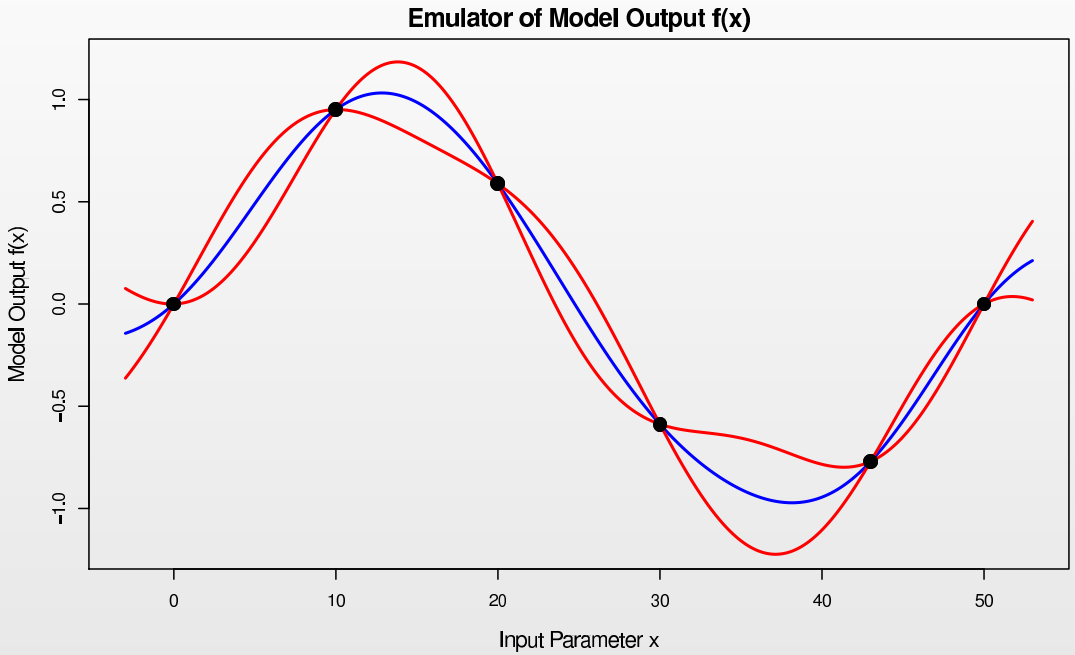


Emulation: a 1D Example



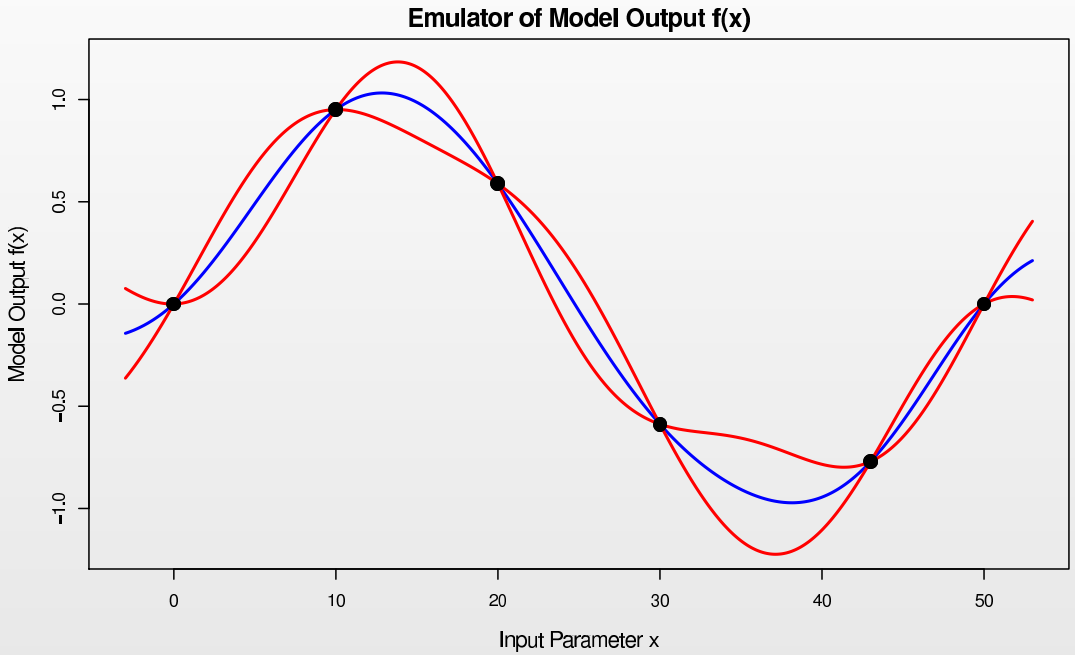
- Note the way that $\mathrm{E}_D(f(x))$ (the blue line) agrees precisely with the 6 known runs (the black dots) given by D.

Emulation: a 1D Example



- Note the way that $\mathrm{E}_D(f(x))$ (the blue line) agrees precisely with the 6 known runs (the black dots) given by D.
- Note also that the credible interval (the red lines) shrinks to zero at the 6 known runs: this is due to $\mathrm{Var}_D(f(x))$ going to zero at the points in D.

Emulation: a 1D Example



- Note the way that $\mathrm{E}_D(f(x))$ (the blue line) agrees precisely with the 6 known runs (the black dots) given by D.
- Note also that the credible interval (the red lines) shrinks to zero at the 6 known runs: this is due to $\mathrm{Var}_D(f(x))$ going to zero at the points in D.
- Today you will work with R code that does the above, and you will be able to play with β_0, σ, θ, D, $x^{(j)}$ and the function $f(x)$.

Simple 1-dimensional Sine Example: Implausibility

- Often we would want to emulate an expensive function because we are interested in history matching: finding the inputs x that give acceptable matches between the outputs $f(x)$ and some history data z.

Simple 1-dimensional Sine Example: Implausibility

- Often we would want to emulate an expensive function because we are interested in history matching: finding the inputs x that give acceptable matches between the outputs $f(x)$ and some history data z.
- z is measured with observation error e, with variance σ_e^2, so were the model perfect, we would use an implausibility measure $I(x)$:

$$I^2(x) = \frac{|\mathrm{E}_D(f(x)) - z|^2}{(\mathrm{Var}_D(f(x)) + \mathrm{Var}(e))}$$

Simple 1-dimensional Sine Example: Implausibility

- Often we would want to emulate an expensive function because we are interested in history matching: finding the inputs x that give acceptable matches between the outputs $f(x)$ and some history data z.
- z is measured with observation error e, with variance σ_e^2, so were the model perfect, we would use an implausibility measure $I(x)$:

$$I^2(x) = \frac{|\mathrm{E}_D(f(x)) - z|^2}{(\mathrm{Var}_D(f(x)) + \mathrm{Var}(e))}$$

- Remember that high values of $I(x)$ imply that x is unlikely to give a good match (x is implausible), but low values just mean we are unsure about x.

Simple 1-dimensional Sine Example: Implausibility

- Often we would want to emulate an expensive function because we are interested in history matching: finding the inputs x that give acceptable matches between the outputs $f(x)$ and some history data z.
- z is measured with observation error e, with variance σ_e^2, so were the model perfect, we would use an implausibility measure $I(x)$:

$$I^2(x) = \frac{|\mathrm{E}_D(f(x)) - z|^2}{(\mathrm{Var}_D(f(x)) + \mathrm{Var}(e))}$$

- Remember that high values of $I(x)$ imply that x is unlikely to give a good match (x is implausible), but low values just mean we are unsure about x.
- Usually we would judge that our model is inaccurate and hence has a structural model discrepancy ϵ, with variance σ_ϵ^2. The implausibility becomes:

$$I^2(x) = \frac{|\mathrm{E}_D(f(x)) - z|^2}{(\mathrm{Var}_D(f(x)) + \mathrm{Var}(\epsilon) + \mathrm{Var}(e))}$$

Simple 1-dimensional Sine Example: Implausibility

- Often we would want to emulate an expensive function because we are interested in history matching: finding the inputs x that give acceptable matches between the outputs $f(x)$ and some history data z.
- z is measured with observation error e, with variance σ_e^2, so were the model perfect, we would use an implausibility measure $I(x)$:

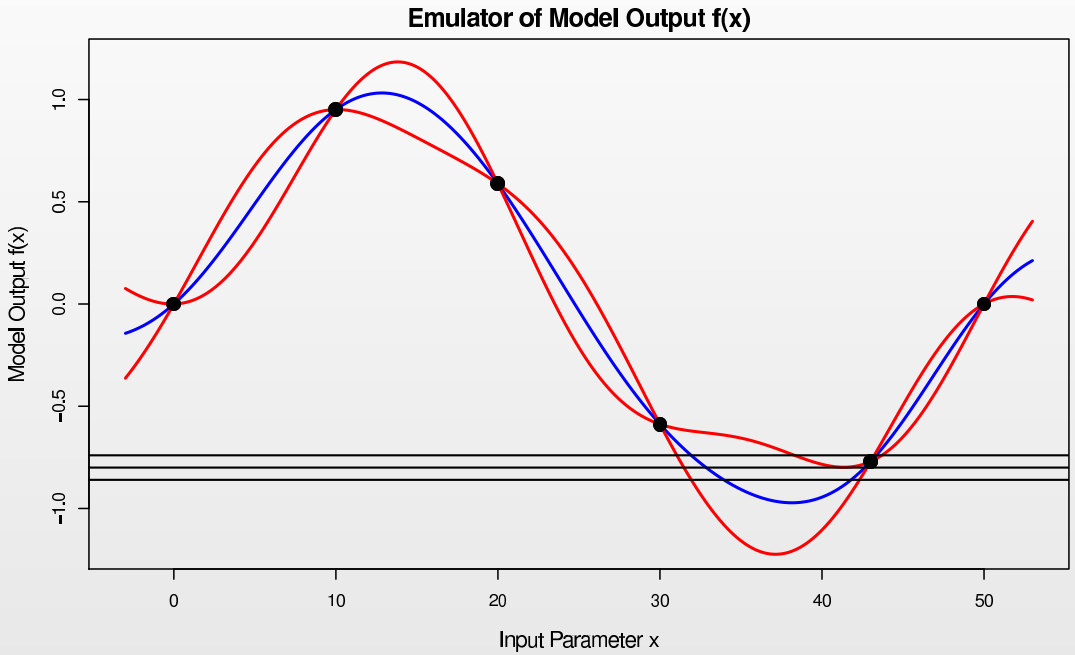
$$I^2(x) = \frac{|\mathrm{E}_D(f(x)) - z|^2}{(\mathrm{Var}_D(f(x)) + \mathrm{Var}(e))}$$

- Remember that high values of $I(x)$ imply that x is unlikely to give a good match (x is implausible), but low values just mean we are unsure about x.
- Usually we would judge that our model is inaccurate and hence has a structural model discrepancy ϵ, with variance σ_ϵ^2. The implausibility becomes:

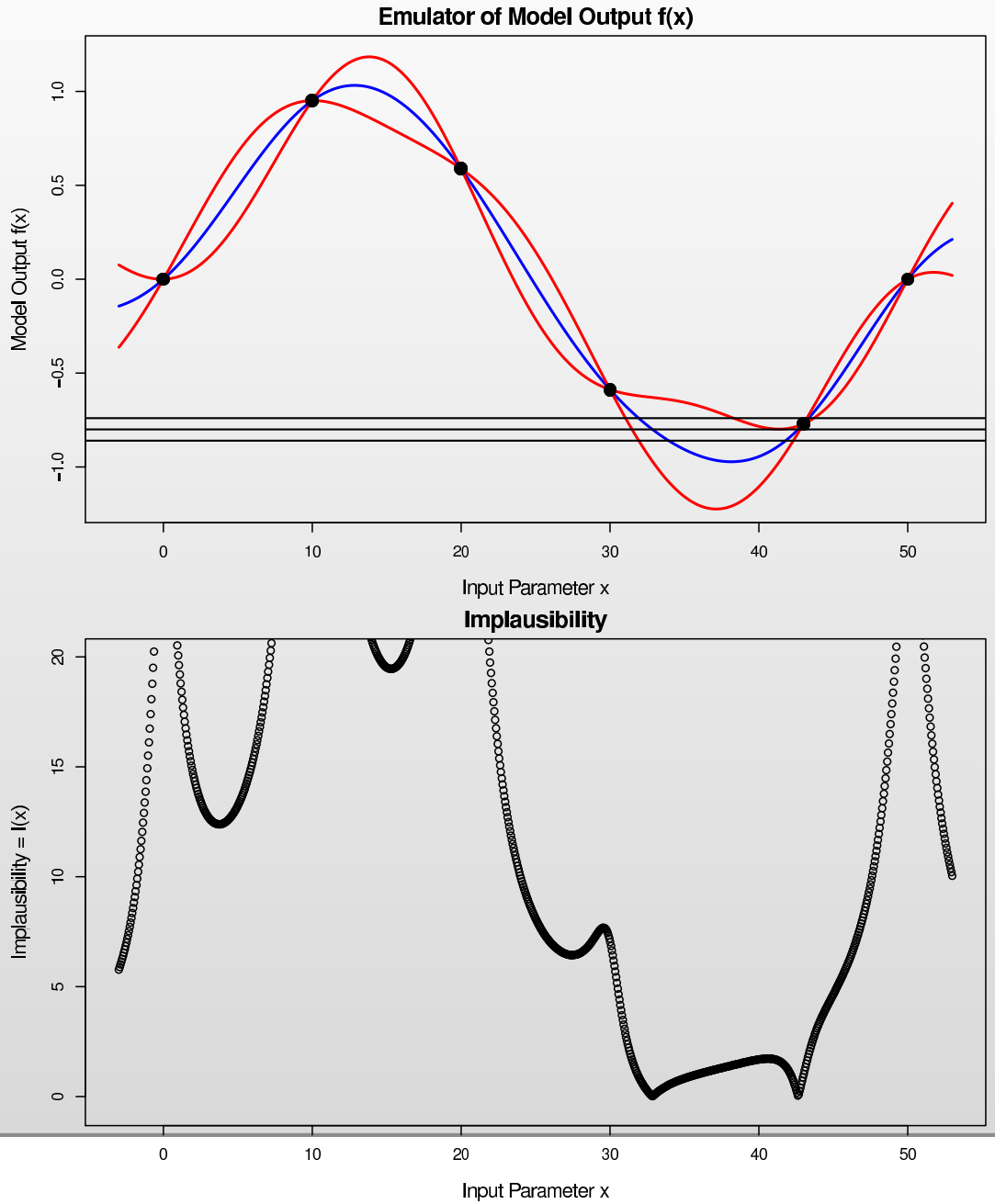
$$I^2(x) = \frac{|\mathrm{E}_D(f(x)) - z|^2}{(\mathrm{Var}_D(f(x)) + \mathrm{Var}(\epsilon) + \mathrm{Var}(e))}$$

- For our example we take $z = -0.8$ and $\sigma_e^2 + \sigma_\epsilon^2 = 0.02^2$ and plot $I(x)$.

Simple 1-dimensional Sine Example: Implausibility



Simple 1-dimensional Sine Example: Implausibility



Simple Example: 2nd Wave History Match

- We now impose a cutoff on the implausibility:

$$I(x) < c$$

Simple Example: 2nd Wave History Match

- We now impose a cutoff on the implausibility:

$$I(x) < c$$

- Where c is often chosen to be 3 due to Pukelsheim's 3-sigma rule.

Simple Example: 2nd Wave History Match

- We now impose a cutoff on the implausibility:

$$I(x) < c$$

- Where c is often chosen to be 3 due to Pukelsheim's 3-sigma rule.
- x values that violate this cutoff are deemed implausible (plotted as red)

Simple Example: 2nd Wave History Match

- We now impose a cutoff on the implausibility:

$$I(x) < c$$

- Where c is often chosen to be 3 due to Pukelsheim's 3-sigma rule.
- x values that violate this cutoff are deemed implausible (plotted as red)
- x values that satisfy it are deemed non-implausible, which are OK for now, but could be ruled out later (plotted as green).

Simple Example: 2nd Wave History Match

- We now impose a cutoff on the implausibility:

$$I(x) < c$$

- Where c is often chosen to be 3 due to Pukelsheim's 3-sigma rule.
- x values that violate this cutoff are deemed implausible (plotted as red)
- x values that satisfy it are deemed non-implausible, which are OK for now, but could be ruled out later (plotted as green).
- We perform a second wave of the iterative history match by running the model at three more x points, chosen only from the non-implausible space.

Simple Example: 2nd Wave History Match

- We now impose a cutoff on the implausibility:

$$I(x) < c$$

- Where c is often chosen to be 3 due to Pukelsheim's 3-sigma rule.
- x values that violate this cutoff are deemed implausible (plotted as red)
- x values that satisfy it are deemed non-implausible, which are OK for now, but could be ruled out later (plotted as green).
- We perform a second wave of the iterative history match by running the model at three more x points, chosen only from the non-implausible space.
- So we add the following runs to the vector D:

$$x^{(j)}_{wave2} = 40, 37, 33$$

Simple Example: 2nd Wave History Match

- We now impose a cutoff on the implausibility:

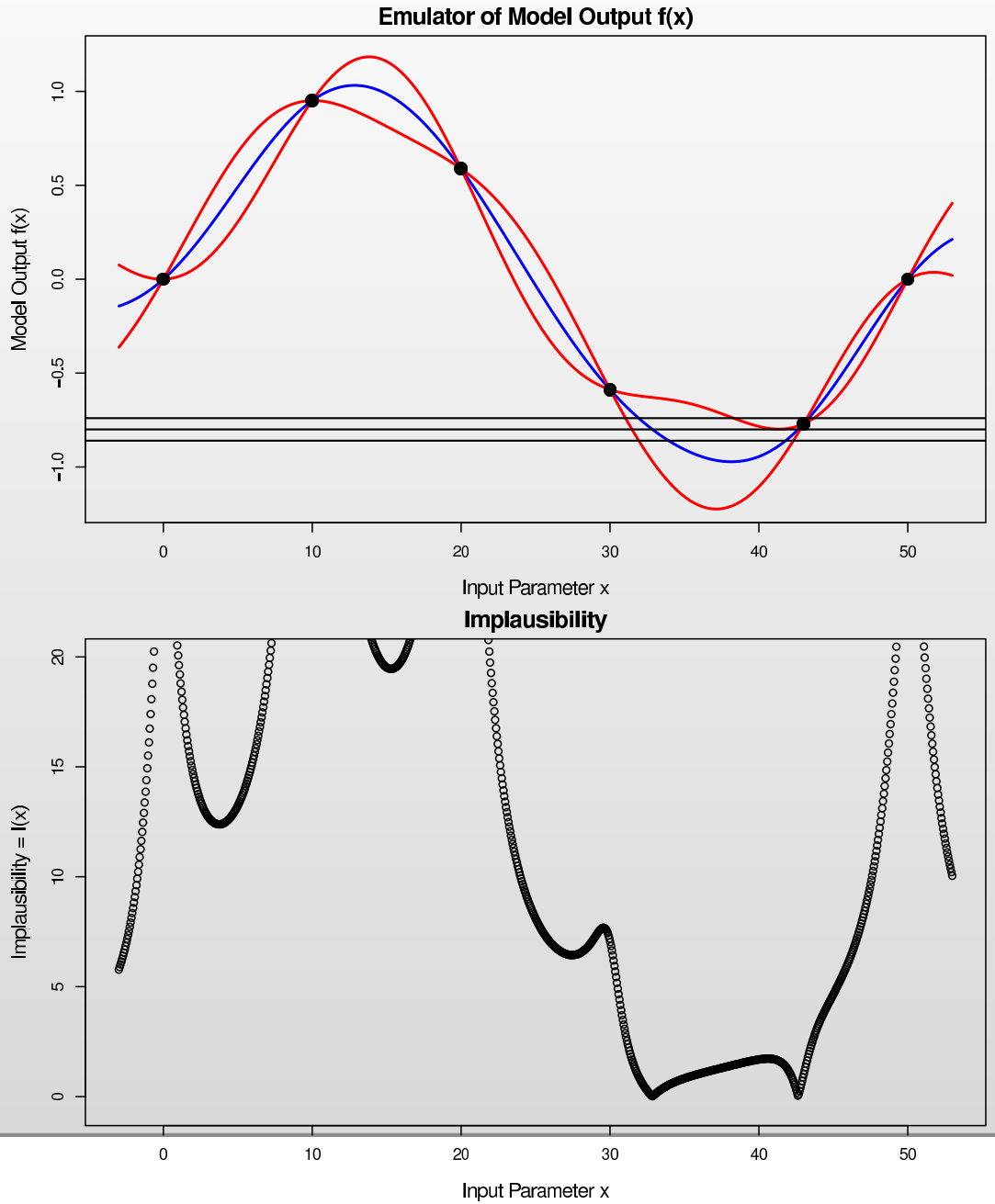
$$I(x) < c$$

- Where c is often chosen to be 3 due to Pukelsheim's 3-sigma rule.
- x values that violate this cutoff are deemed implausible (plotted as red)
- x values that satisfy it are deemed non-implausible, which are OK for now, but could be ruled out later (plotted as green).
- We perform a second wave of the iterative history match by running the model at three more x points, chosen only from the non-implausible space.
- So we add the following runs to the vector D:

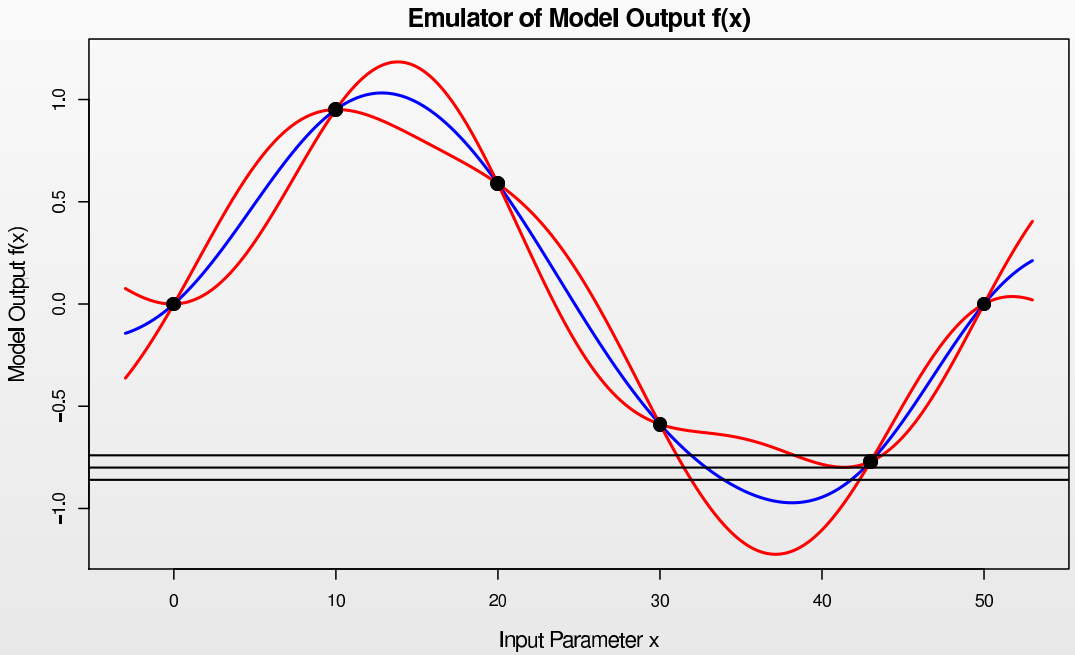
$$x^{(j)}_{wave2} = 40, 37, 33$$

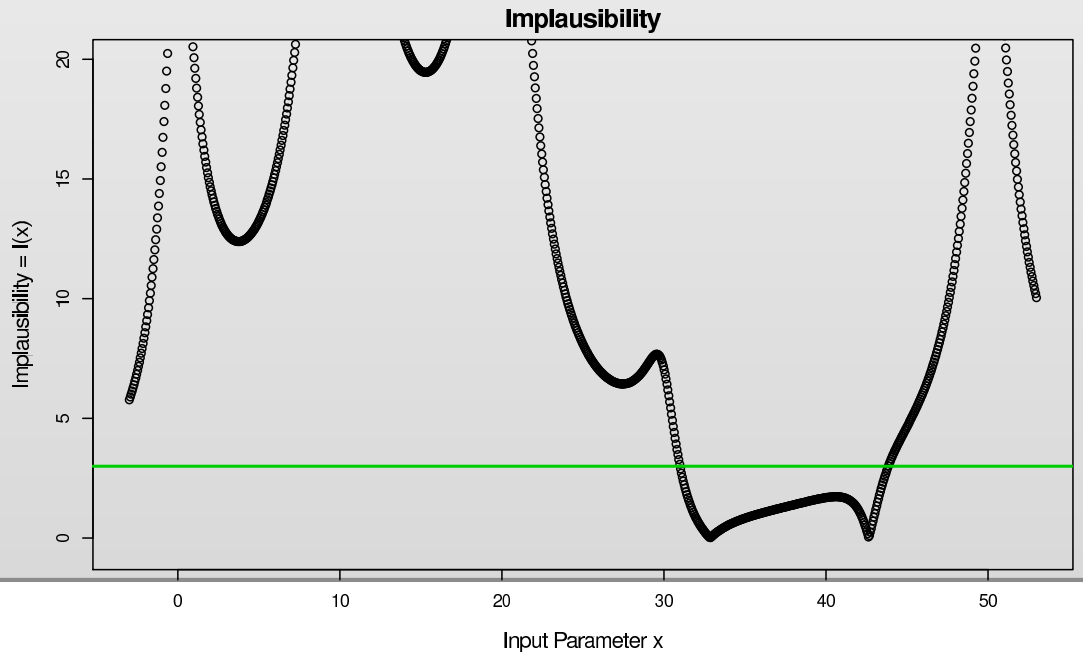
- (This is a slight simplification, often we may create a new emulator from the $x^{(j)}_{wave2}$ points alone, perhaps using the wave 1 emulator as prior).

Simple Example: 2nd Wave History Match

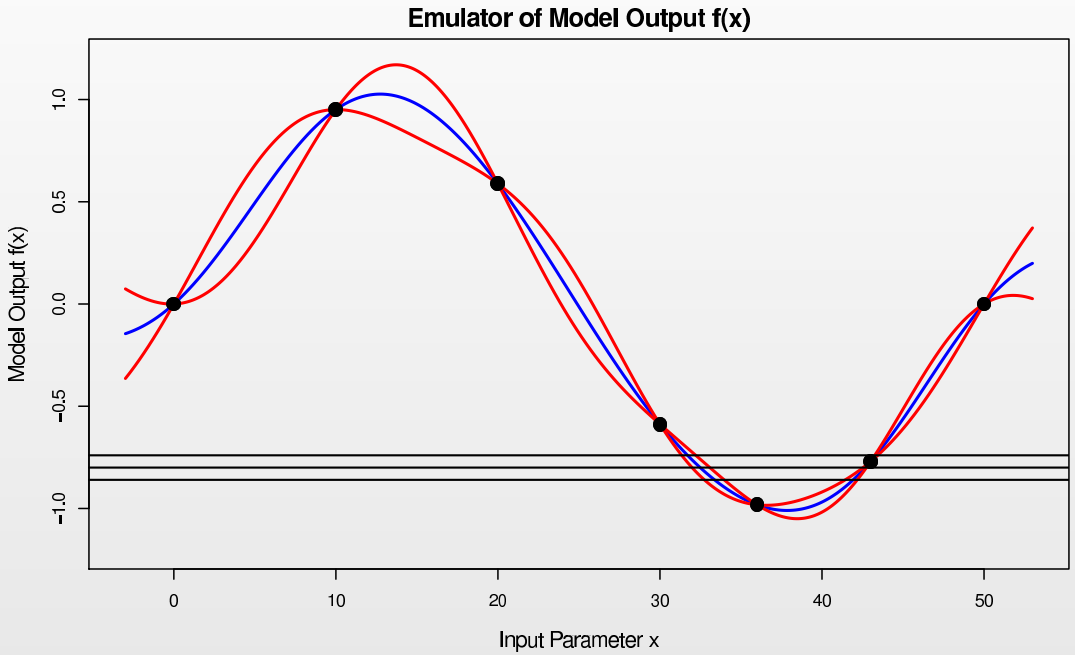


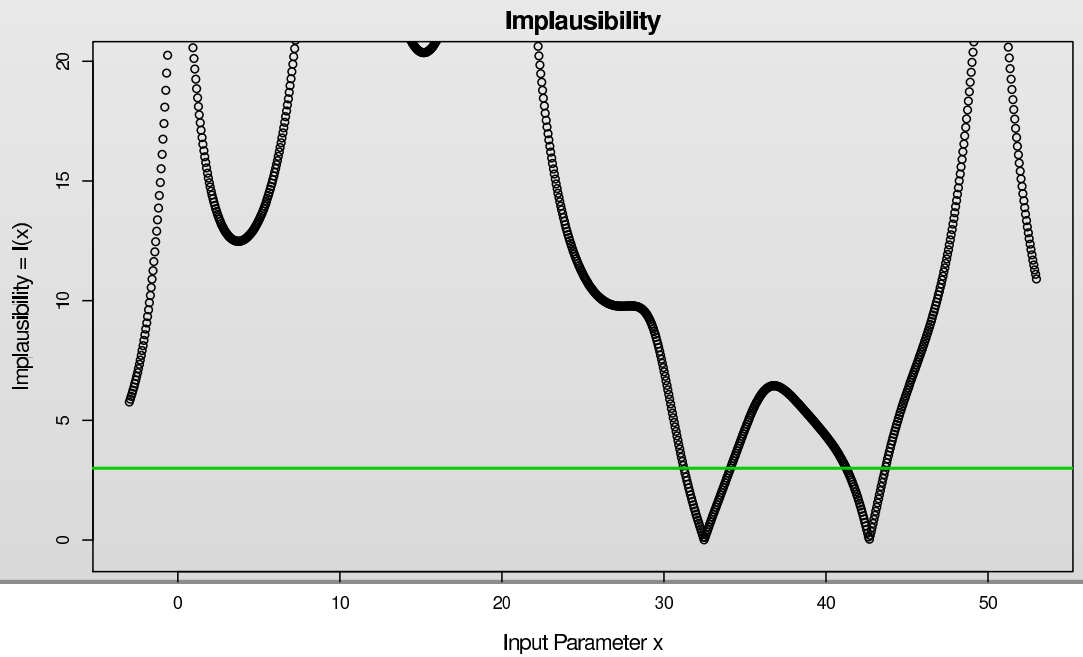
Simple Example: 2nd Wave History Match



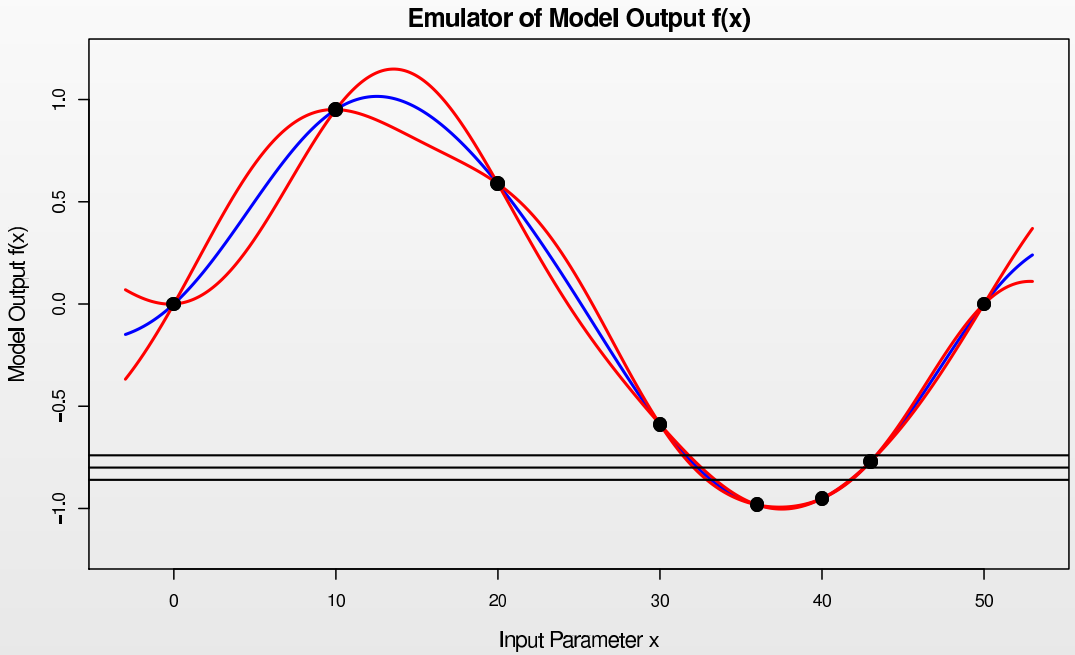


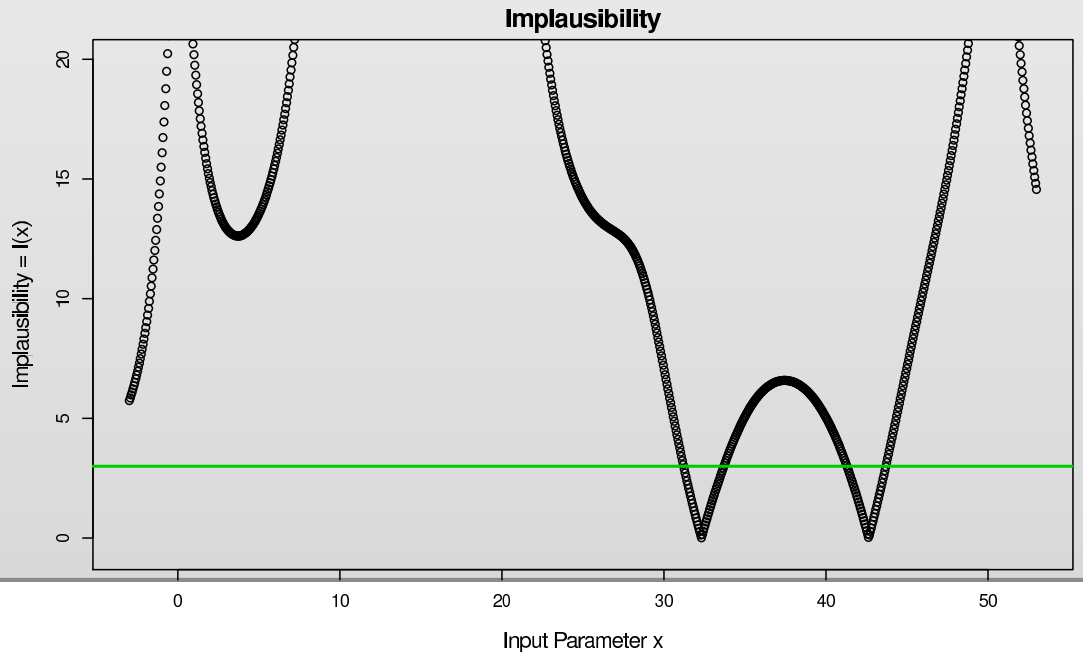
Simple Example: 2nd Wave History Match



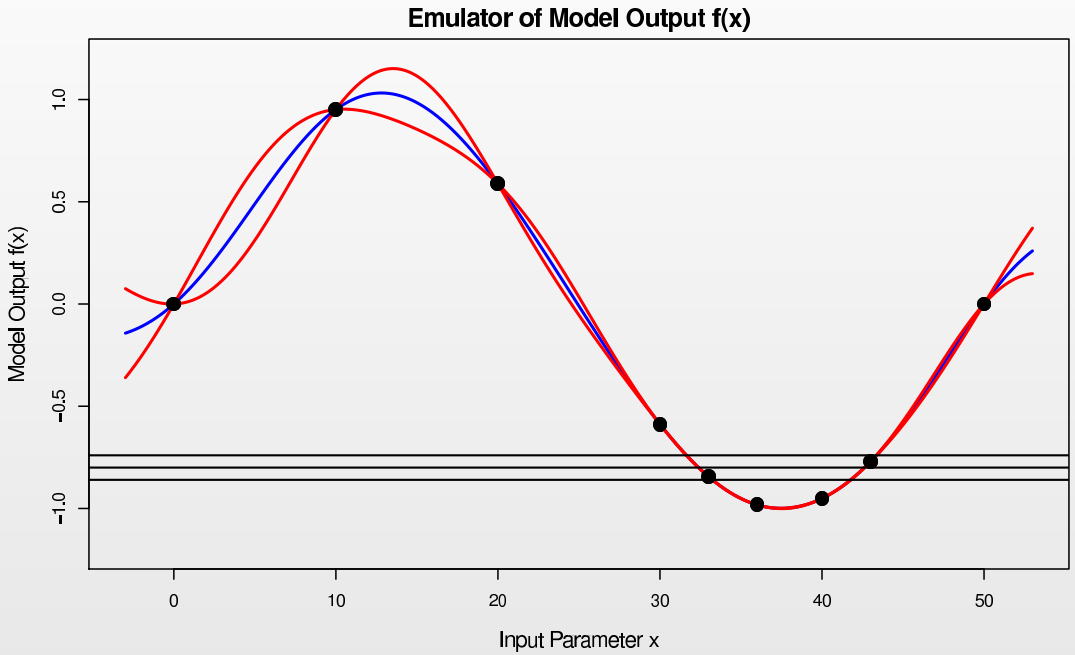


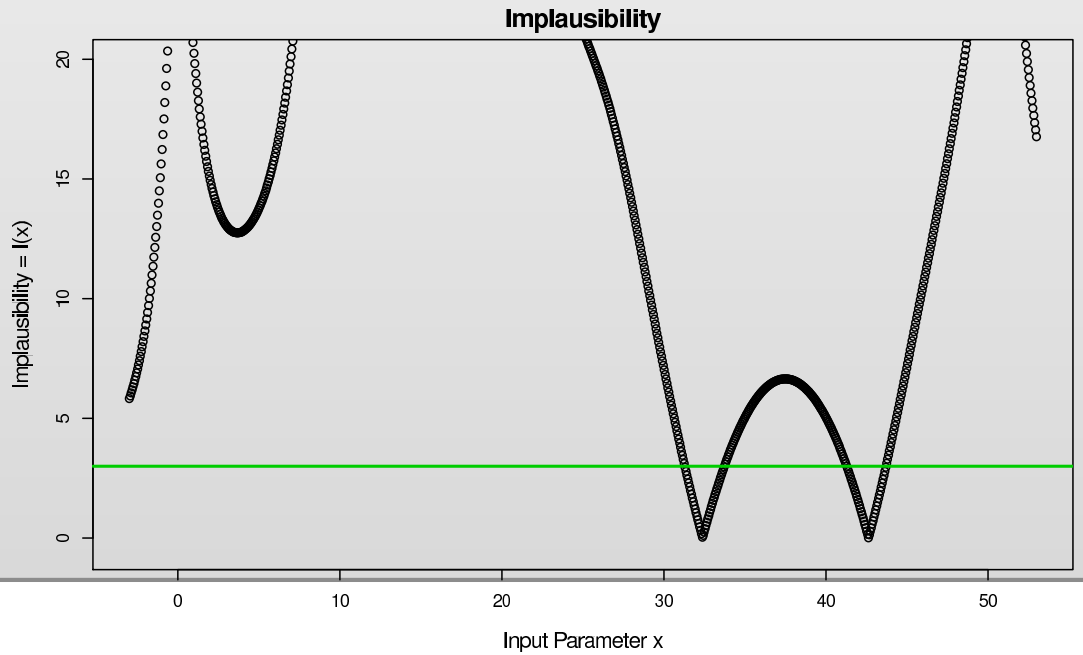
Simple Example: 2nd Wave History Match



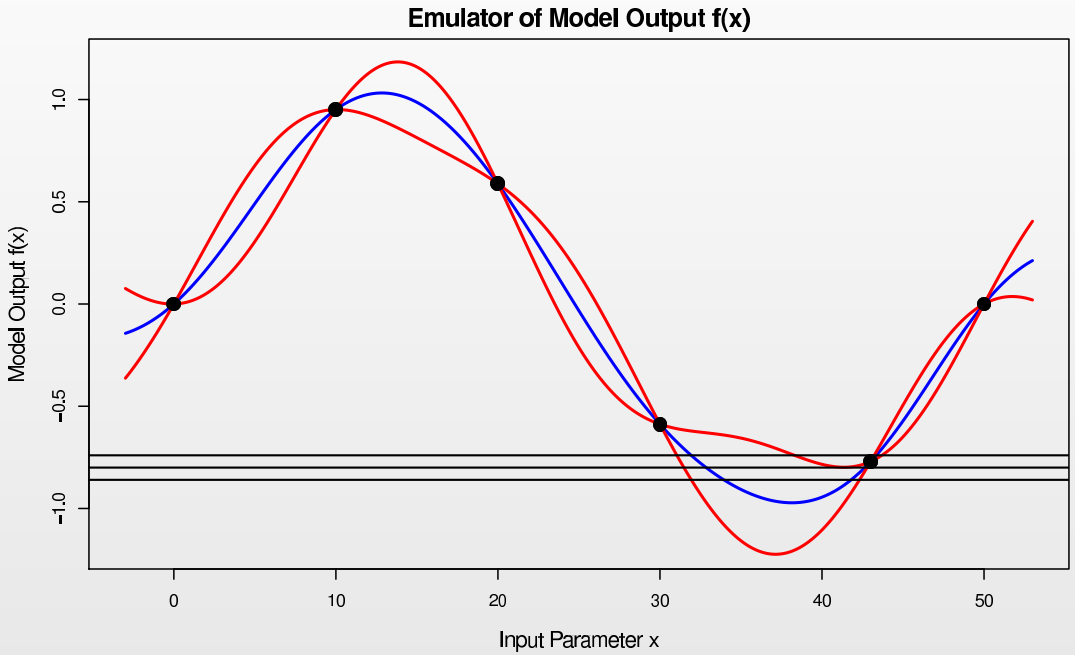


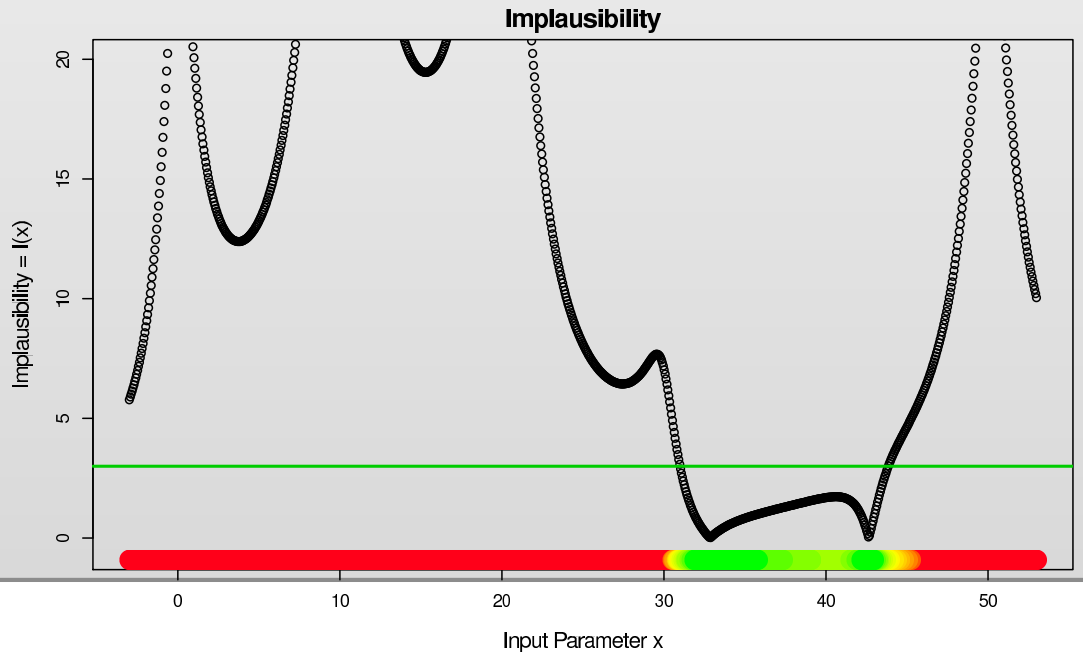
Simple Example: 2nd Wave History Match



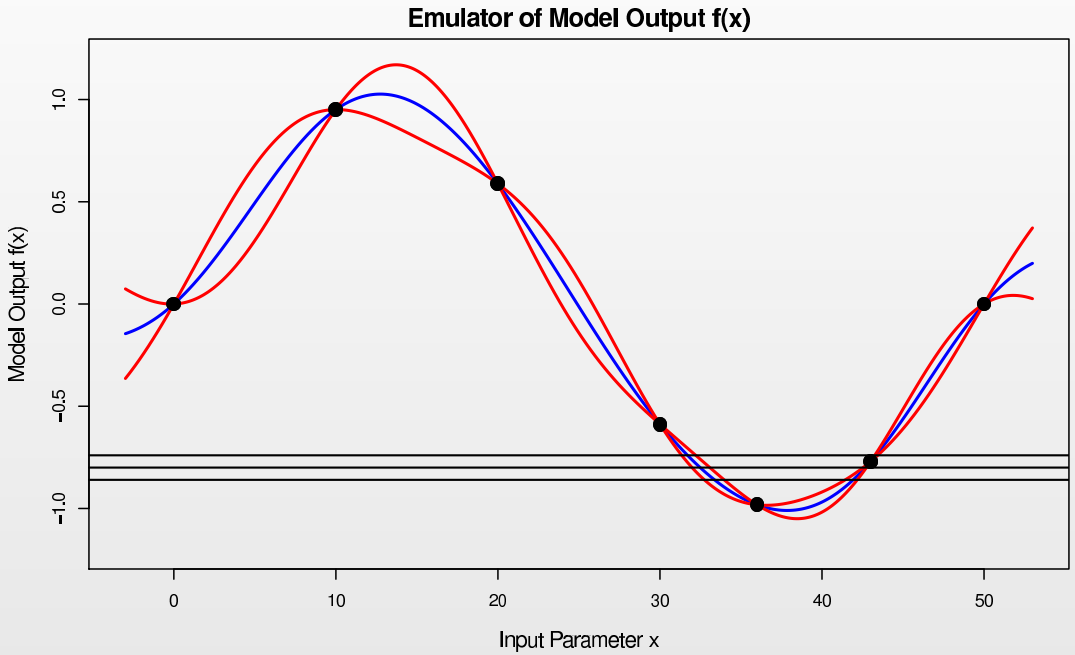


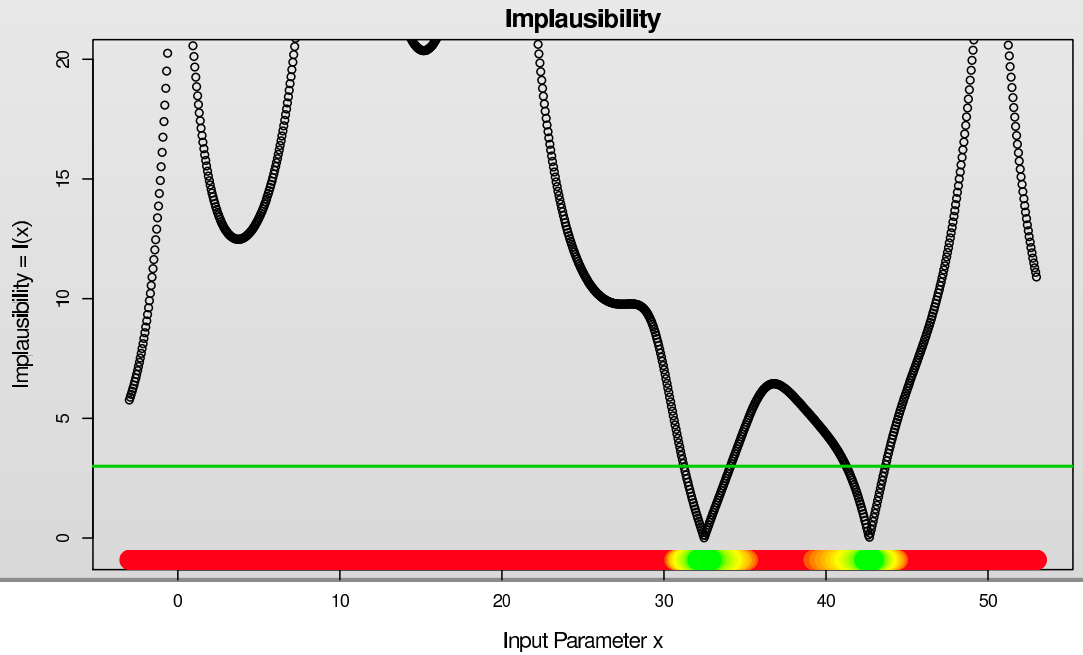
Simple Example: 2nd Wave History Match



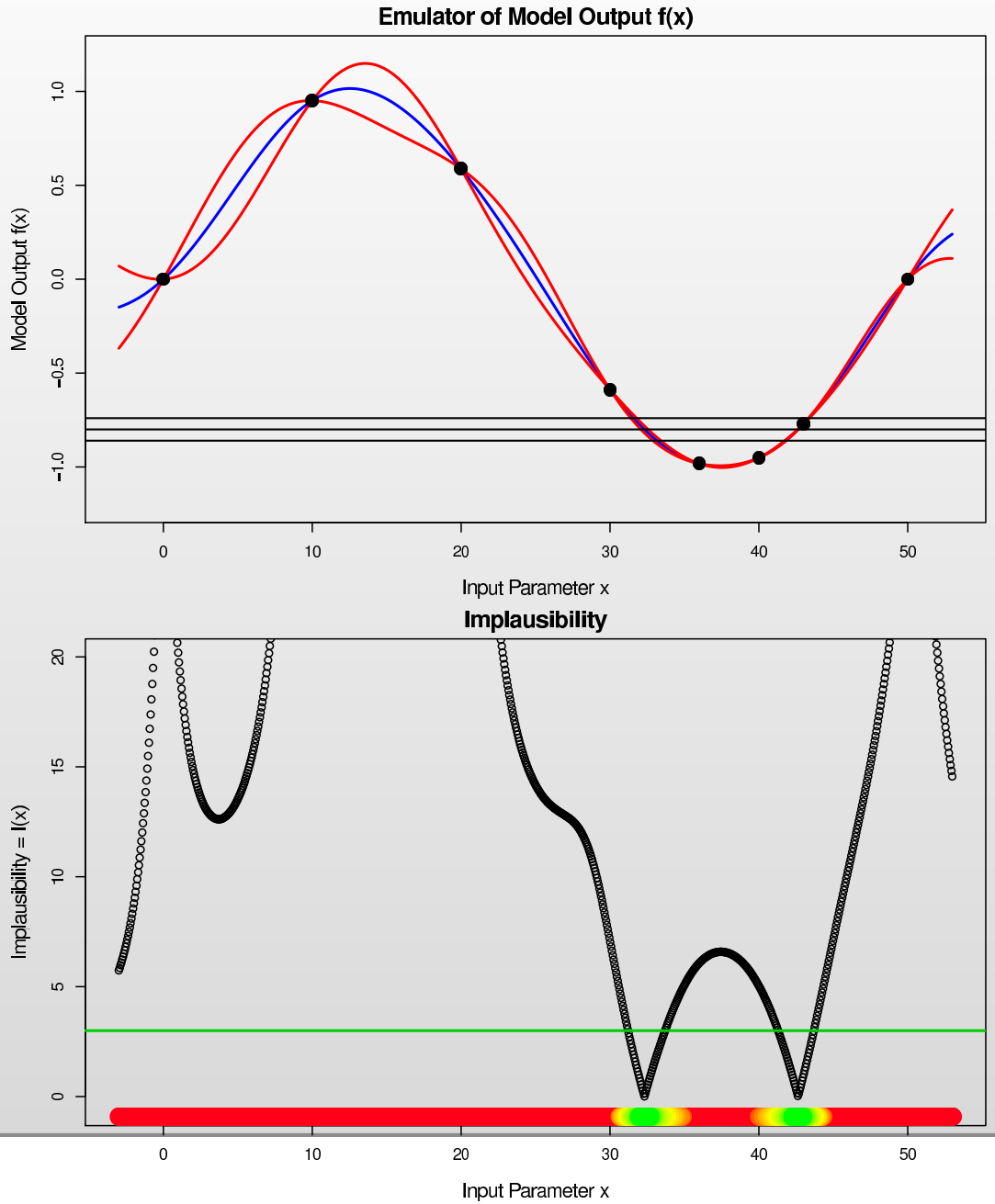


Simple Example: 2nd Wave History Match

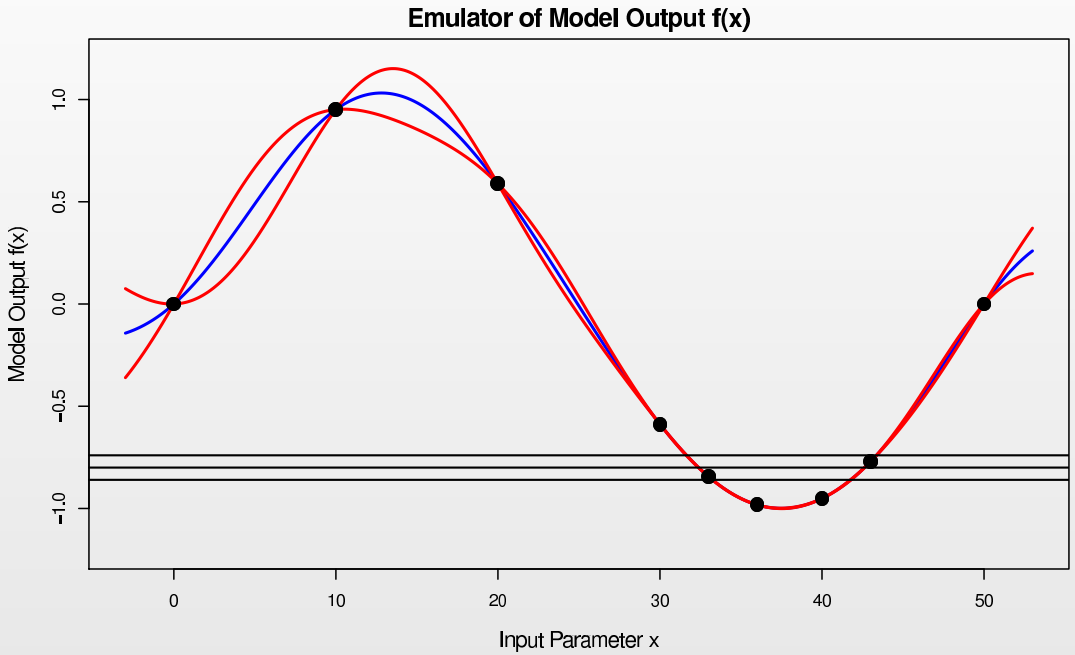


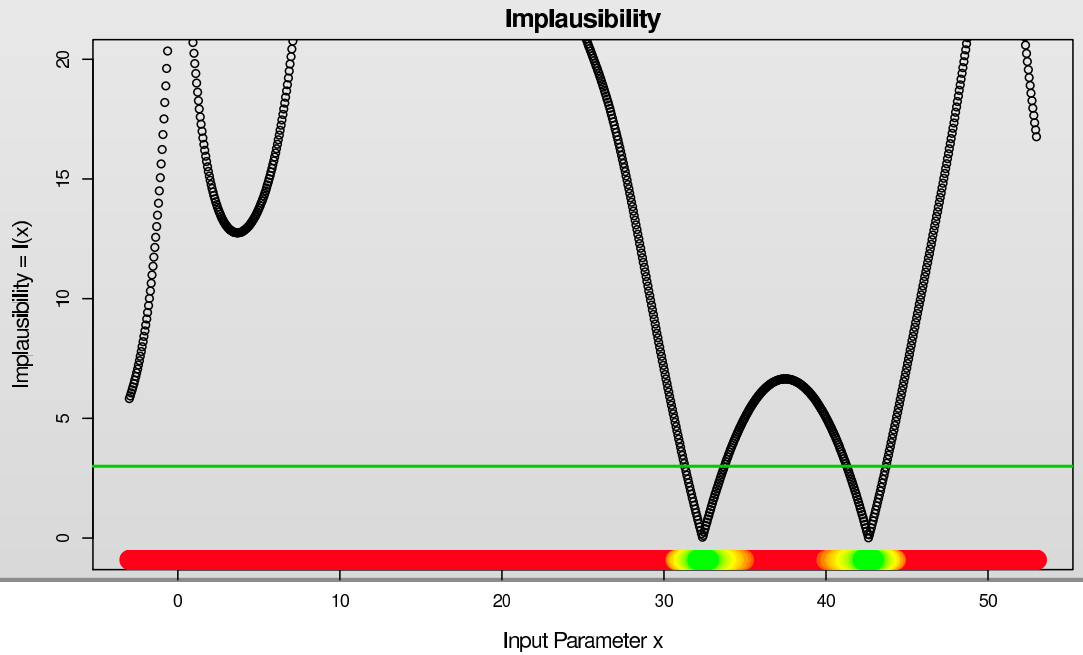


Simple Example: 2nd Wave History Match



Simple Example: 2nd Wave History Match





Iterative History Matching for Reducing Input Space.

We use an iterative strategy to reduce the input parameter space. Denoting the current non-implausible volume by $\mathcal{X}_j$, at each stage or wave we:

Iterative History Matching for Reducing Input Space.

We use an iterative strategy to reduce the input parameter space. Denoting the current non-implausible volume by $\mathcal{X}_j$, at each stage or wave we:

1. Design and perform a set of runs over the non-implausible input region $\mathcal{X}_j$

Iterative History Matching for Reducing Input Space.

We use an iterative strategy to reduce the input parameter space. Denoting the current non-implausible volume by $\mathcal{X}_j$, at each stage or wave we:

1. Design and perform a set of runs over the non-implausible input region $\mathcal{X}_j$
2. Identify the set Q_{j+1} of informative outputs that we can emulate easily

Iterative History Matching for Reducing Input Space.

We use an iterative strategy to reduce the input parameter space. Denoting the current non-implausible volume by $\mathcal{X}_j$, at each stage or wave we:

1. Design and perform a set of runs over the non-implausible input region $\mathcal{X}_j$
2. Identify the set Q_{j+1} of informative outputs that we can emulate easily
3. Construct new emulators for $f_i(x)$, where $i \in Q_{j+1}$ defined only over $\mathcal{X}_j$

Iterative History Matching for Reducing Input Space.

We use an iterative strategy to reduce the input parameter space. Denoting the current non-implausible volume by $\mathcal{X}_j$, at each stage or wave we:

1. Design and perform a set of runs over the non-implausible input region $\mathcal{X}_j$
2. Identify the set Q_{j+1} of informative outputs that we can emulate easily
3. Construct new emulators for $f_i(x)$, where $i \in Q_{j+1}$ defined only over $\mathcal{X}_j$
4. Evaluate the new implausibility functions $I_i(x), i \in Q_{j+1}$ only over $\mathcal{X}_j$

Iterative History Matching for Reducing Input Space.

We use an iterative strategy to reduce the input parameter space. Denoting the current non-implausible volume by $\mathcal{X}_j$, at each stage or wave we:

1. Design and perform a set of runs over the non-implausible input region $\mathcal{X}_j$
2. Identify the set Q_{j+1} of informative outputs that we can emulate easily
3. Construct new emulators for $f_i(x)$, where $i \in Q_{j+1}$ defined only over $\mathcal{X}_j$
4. Evaluate the new implausibility functions $I_i(x), i \in Q_{j+1}$ only over $\mathcal{X}_j$
5. Define a new (reduced) non-implausible region $\mathcal{X}_{j+1}$, by $I_M(x) < c_M$, which should satisfy $\mathcal{X} \subset \mathcal{X}_{j+1} \subset \mathcal{X}_j$

Iterative History Matching for Reducing Input Space.

We use an iterative strategy to reduce the input parameter space. Denoting the current non-implausible volume by $\mathcal{X}_j$, at each stage or wave we:

1. Design and perform a set of runs over the non-implausible input region $\mathcal{X}_j$
2. Identify the set Q_{j+1} of informative outputs that we can emulate easily
3. Construct new emulators for $f_i(x)$, where $i \in Q_{j+1}$ defined only over $\mathcal{X}_j$
4. Evaluate the new implausibility functions $I_i(x), i \in Q_{j+1}$ only over $\mathcal{X}_j$
5. Define a new (reduced) non-implausible region $\mathcal{X}_{j+1}$, by $I_M(x) < c_M$, which should satisfy $\mathcal{X} \subset \mathcal{X}_{j+1} \subset \mathcal{X}_j$
6. Unless (a) the emulator variances are now small in comparison to the other sources of uncertainty (model discrepancy and observation errors) or (b) computational resources are exhausted or (c) all the input space is deemed implausible, return to step 1

Iterative History Matching for Reducing Input Space.

We use an iterative strategy to reduce the input parameter space. Denoting the current non-implausible volume by $\mathcal{X}_j$, at each stage or wave we:

1. Design and perform a set of runs over the non-implausible input region $\mathcal{X}_j$
2. Identify the set Q_{j+1} of informative outputs that we can emulate easily
3. Construct new emulators for $f_i(x)$, where $i \in Q_{j+1}$ defined only over $\mathcal{X}_j$
4. Evaluate the new implausibility functions $I_i(x), i \in Q_{j+1}$ only over $\mathcal{X}_j$
5. Define a new (reduced) non-implausible region $\mathcal{X}_{j+1}$, by $I_M(x) < c_M$, which should satisfy $\mathcal{X} \subset \mathcal{X}_{j+1} \subset \mathcal{X}_j$
6. Unless (a) the emulator variances are now small in comparison to the other sources of uncertainty (model discrepancy and observation errors) or (b) computational resources are exhausted or (c) all the input space is deemed implausible, return to step 1
7. If 6(a) true, generate a large number of acceptable runs from the final non-implausible volume $\mathcal{X}$

More Advanced Emulators

- The simple emulator above had $f(x) = \beta_0 + u(x)$: this is a useful emulator but will fail for many functions that have high input dimension.

More Advanced Emulators

- The simple emulator above had $f(x) = \beta_0 + u(x)$: this is a useful emulator but will fail for many functions that have high input dimension.
- We can improve our emulators by bringing back the polynomial terms so:

$$f(x) = \sum_j \beta_j \, g_j(x) + u(x)$$

More Advanced Emulators

- The simple emulator above had $f(x) = \beta_0 + u(x)$: this is a useful emulator but will fail for many functions that have high input dimension.
- We can improve our emulators by bringing back the polynomial terms so:

$$f(x) = \sum_j \beta_j \, g_j(x) + u(x)$$

- Now lets say we have 2 inputs so x is now a vector $x = (x_1, x_2)$. The polynomial term of order 2 would look like:

$$\sum_j \beta_j \, g_j(x) \;=\; \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_1^2 + \beta_4 x_2^2 + \beta_5 x_1 x_2$$

More Advanced Emulators

- The simple emulator above had $f(x) = \beta_0 + u(x)$: this is a useful emulator but will fail for many functions that have high input dimension.
- We can improve our emulators by bringing back the polynomial terms so:

$$f(x) = \sum_j \beta_j \, g_j(x) + u(x)$$

- Now lets say we have 2 inputs so x is now a vector $x = (x_1, x_2)$. The polynomial term of order 2 would look like:

$$\sum_j \beta_j \, g_j(x) \;=\; \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_1^2 + \beta_4 x_2^2 + \beta_5 x_1 x_2$$

- These polynomial terms are very good a describing the global behaviour of $f(x)$ while the stationary process $u(x)$ can mimic the more local behaviour.

More Advanced Emulators

- The simple emulator above had $f(x) = \beta_0 + u(x)$: this is a useful emulator but will fail for many functions that have high input dimension.
- We can improve our emulators by bringing back the polynomial terms so:

$$f(x) = \sum_j \beta_j \, g_j(x) + u(x)$$

- Now lets say we have 2 inputs so x is now a vector $x = (x_1, x_2)$. The polynomial term of order 2 would look like:

$$\sum_j \beta_j \, g_j(x) \;=\; \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_1^2 + \beta_4 x_2^2 + \beta_5 x_1 x_2$$

- These polynomial terms are very good a describing the global behaviour of $f(x)$ while the stationary process $u(x)$ can mimic the more local behaviour.
- We can in principle specify prior expectations, variances and covariances for all the β_j terms: $\mathrm{E}(\beta_j)$ and $\mathrm{Cov}(\beta_j, \beta_k)$, if we have informed beliefs about them.

More Advanced Emulators

- For example, if we have access to a fast version of the model (e.g. a coarse or lower resolution version of a reservoir model) that we can run many times, we can use this to obtain highly informed priors for β_j.

More Advanced Emulators

- For example, if we have access to a fast version of the model (e.g. a coarse or lower resolution version of a reservoir model) that we can run many times, we can use this to obtain highly informed priors for β_j.
- This is a powerful technique that is noticably under used in the oil industry.

More Advanced Emulators

- For example, if we have access to a fast version of the model (e.g. a coarse or lower resolution version of a reservoir model) that we can run many times, we can use this to obtain highly informed priors for β_j.
- This is a powerful technique that is noticably under used in the oil industry.
- Alternatively, if we have performed a reasonable number of runs that are well spaced in the input space, the correlation between the $u(x^{(j)})$ terms will be small and our emulator will tend toward a standard regression model (or linear model).

More Advanced Emulators

- For example, if we have access to a fast version of the model (e.g. a coarse or lower resolution version of a reservoir model) that we can run many times, we can use this to obtain highly informed priors for β_j.
- This is a powerful technique that is noticably under used in the oil industry.
- Alternatively, if we have performed a reasonable number of runs that are well spaced in the input space, the correlation between the $u(x^{(j)})$ terms will be small and our emulator will tend toward a standard regression model (or linear model).
- Hence we can approximate the β_j terms by their Ordinary Least Squares (OLS) estimate, (and use a corresponding estimate for σ^2 too).

More Advanced Emulators

- For example, if we have access to a fast version of the model (e.g. a coarse or lower resolution version of a reservoir model) that we can run many times, we can use this to obtain highly informed priors for β_j.
- This is a powerful technique that is noticably under used in the oil industry.
- Alternatively, if we have performed a reasonable number of runs that are well spaced in the input space, the correlation between the $u(x^{(j)})$ terms will be small and our emulator will tend toward a standard regression model (or linear model).
- Hence we can approximate the β_j terms by their Ordinary Least Squares (OLS) estimate, (and use a corresponding estimate for σ^2 too).
- As this is actually corresponds to using a Maximum Likelihood Estimate (MLE), it can be used to approximate the full Bayesian posterior for the β's, in the large run number limit, as described yesterday.

More Advanced Emulators

- For example, if we have access to a fast version of the model (e.g. a coarse or lower resolution version of a reservoir model) that we can run many times, we can use this to obtain highly informed priors for β_j.
- This is a powerful technique that is noticably under used in the oil industry.
- Alternatively, if we have performed a reasonable number of runs that are well spaced in the input space, the correlation between the $u(x^{(j)})$ terms will be small and our emulator will tend toward a standard regression model (or linear model).
- Hence we can approximate the β_j terms by their Ordinary Least Squares (OLS) estimate, (and use a corresponding estimate for σ^2 too).
- As this is actually corresponds to using a Maximum Likelihood Estimate (MLE), it can be used to approximate the full Bayesian posterior for the β's, in the large run number limit, as described yesterday.
- We can obtain all the OLS estimates we need from the R function "lm()" (see later this afternoon).

More Advanced Emulators: choosing the polynomial

- Only some of the polynomial terms will be useful, and there may be a long list of them if the input dimension is high.

More Advanced Emulators: choosing the polynomial

- Only some of the polynomial terms will be useful, and there may be a long list of them if the input dimension is high.
- We reduce this problem by identifying the active inputs x^A for the output $f(x)$: those inputs that have a clear effect on $f(x)$.

More Advanced Emulators: choosing the polynomial

- Only some of the polynomial terms will be useful, and there may be a long list of them if the input dimension is high.
- We reduce this problem by identifying the active inputs x^A for the output $f(x)$: those inputs that have a clear effect on $f(x)$.
- Now we build the polynomial just out of terms involving the active inputs: $\beta_j g_j(x^A)$. (The other inputs are referred to as inactive).

More Advanced Emulators: choosing the polynomial

- Only some of the polynomial terms will be useful, and there may be a long list of them if the input dimension is high.
- We reduce this problem by identifying the active inputs x^A for the output $f(x)$: those inputs that have a clear effect on $f(x)$.
- Now we build the polynomial just out of terms involving the active inputs: $\beta_j g_j(x^A)$. (The other inputs are referred to as inactive).
- To reduce the size of the polynomial further we can use model selection techniques, e.g. based on AIC or BIC criteria, to select only the polynomial terms that seem to mimic the function $f(x)$ well.

More Advanced Emulators: choosing the polynomial

- Only some of the polynomial terms will be useful, and there may be a long list of them if the input dimension is high.
- We reduce this problem by identifying the active inputs x^A for the output $f(x)$: those inputs that have a clear effect on $f(x)$.
- Now we build the polynomial just out of terms involving the active inputs: $\beta_j g_j(x^A)$. (The other inputs are referred to as inactive).
- To reduce the size of the polynomial further we can use model selection techniques, e.g. based on AIC or BIC criteria, to select only the polynomial terms that seem to mimic the function $f(x)$ well.
- This can be done with the R function "step()" (see this afternoon's course).

More Advanced Emulators: choosing the polynomial

- Only some of the polynomial terms will be useful, and there may be a long list of them if the input dimension is high.
- We reduce this problem by identifying the active inputs x^A for the output $f(x)$: those inputs that have a clear effect on $f(x)$.
- Now we build the polynomial just out of terms involving the active inputs: $\beta_j g_j(x^A)$. (The other inputs are referred to as inactive).
- To reduce the size of the polynomial further we can use model selection techniques, e.g. based on AIC or BIC criteria, to select only the polynomial terms that seem to mimic the function $f(x)$ well.
- This can be done with the R function "step()" (see this afternoon's course).
- Now the emulator takes its full form (for a single output $f(x)$):

$$f(x) = \sum_j \beta_j \, g_j(x^A) + u(x^A) + \delta(x)$$

More Advanced Emulators: choosing the polynomial

- Only some of the polynomial terms will be useful, and there may be a long list of them if the input dimension is high.
- We reduce this problem by identifying the active inputs x^A for the output $f(x)$: those inputs that have a clear effect on $f(x)$.
- Now we build the polynomial just out of terms involving the active inputs: $\beta_j g_j(x^A)$. (The other inputs are referred to as inactive).
- To reduce the size of the polynomial further we can use model selection techniques, e.g. based on AIC or BIC criteria, to select only the polynomial terms that seem to mimic the function $f(x)$ well.
- This can be done with the R function "step()" (see this afternoon's course).
- Now the emulator takes its full form (for a single output $f(x)$):

$$f(x) = \sum_j \beta_j \, g_j(x^A) + u(x^A) + \delta(x)$$

- $\delta(x)$ an uncorrelated nugget that has $\mathrm{E}(\delta(x)) = 0$ and $\mathrm{Var}(\delta(x)) = \sigma_\delta^2$.

More Advanced Emulators: Final thoughts

- We prefer to put a lot of effort into finding a good polynomial that mimics most of the behaviour of the function $f(x)$.

More Advanced Emulators: Final thoughts

- We prefer to put a lot of effort into finding a good polynomial that mimics most of the behaviour of the function $f(x)$.
- This then frees the weakly stationary process (or Gaussian process) $u(x)$ to deal with the often more complex local behaviour.

More Advanced Emulators: Final thoughts

- We prefer to put a lot of effort into finding a good polynomial that mimics most of the behaviour of the function $f(x)$.
- This then frees the weakly stationary process (or Gaussian process) $u(x)$ to deal with the often more complex local behaviour.
- It also means that specifying attributes of the weakly stationary process $u(x)$, such as the correlation length θ and the form of the correlation structure itself, become less critical, as $u(x)$ represents a less influential part of the emulator.

More Advanced Emulators: Final thoughts

- We prefer to put a lot of effort into finding a good polynomial that mimics most of the behaviour of the function $f(x)$.
- This then frees the weakly stationary process (or Gaussian process) $u(x)$ to deal with the often more complex local behaviour.
- It also means that specifying attributes of the weakly stationary process $u(x)$, such as the correlation length θ and the form of the correlation structure itself, become less critical, as $u(x)$ represents a less influential part of the emulator.
- The identification of active inputs is also vital to the whole emulation process: it can greatly simplify a very complex and high dimensional function.

More Advanced Emulators: Conclusion

- If the polynomial or regression terms are good enough, we can build fast and useful emulators that are just linear models, with uncorrelated $u(x)$.

More Advanced Emulators: Conclusion

- If the polynomial or regression terms are good enough, we can build fast and useful emulators that are just linear models, with uncorrelated $u(x)$.
- Will see (a bit) more of this in this afternoon's session.

More Advanced Emulators: Conclusion

- If the polynomial or regression terms are good enough, we can build fast and useful emulators that are just linear models, with uncorrelated $u(x)$.
- Will see (a bit) more of this in this afternoon's session.
- Remember that at wave 1 it is unlikely that the reservoir model can be accurately approximated by a low order polynomial, but in later waves when we are in a small part of the input space, the approximation is often much more accurate.

More Advanced Emulators: Conclusion

- If the polynomial or regression terms are good enough, we can build fast and useful emulators that are just linear models, with uncorrelated $u(x)$.
- Will see (a bit) more of this in this afternoon's session.
- Remember that at wave 1 it is unlikely that the reservoir model can be accurately approximated by a low order polynomial, but in later waves when we are in a small part of the input space, the approximation is often much more accurate.
- This is due to $f(x)$ having smoother behaviour over smaller input spaces.

More Advanced Emulators: Conclusion

- If the polynomial or regression terms are good enough, we can build fast and useful emulators that are just linear models, with uncorrelated $u(x)$.
- Will see (a bit) more of this in this afternoon's session.
- Remember that at wave 1 it is unlikely that the reservoir model can be accurately approximated by a low order polynomial, but in later waves when we are in a small part of the input space, the approximation is often much more accurate.
- This is due to $f(x)$ having smoother behaviour over smaller input spaces.
- For an explicit demonstration of this, see our Galaxy papers.

Introduction to Designing future experiments

- We will now have a brief introduction to Designing future experiments or: what data should I pay for?
- Often we want to use a complex model to predict the future, with appropriate uncertainty.
- We would then want to use such predictions to inform us about decision we have to make.
- For example, we may wish to choose the best experiment from a list of possible experiments, to achieve some scientific goal.
- Similarly we may have the choice of paying for extra data, e.g. 4D seismic data, and wondered whether it is worth it.
- We do this by History Matching first, then employing Decision Theory.
- In the examples I show, we are going to judge future experiments based on how good we think they will be at reducing the size of the input space in a future history match: therefore our Utility will be linked to space reduction.

Simple 1D Exponential Growth Example

- Say we are interested in the concentration of a chemical which evolves in time. We will model this concentration as $f(x,t)$ where x is a rate parameter and t is time.

Simple 1D Exponential Growth Example

- Say we are interested in the concentration of a chemical which evolves in time. We will model this concentration as $f(x,t)$ where x is a rate parameter and t is time.
- We think $f(x,t)$ satisfies the differential equation or model:

$$\frac{df(x,t)}{dt} = xf(x,t) \qquad \Longrightarrow \qquad f(x,t) = f_0 \exp(xt)$$

Simple 1D Exponential Growth Example

- Say we are interested in the concentration of a chemical which evolves in time. We will model this concentration as $f(x,t)$ where x is a rate parameter and t is time.
- We think $f(x,t)$ satisfies the differential equation or model:

$$\frac{df(x,t)}{dt} = xf(x,t) \qquad \Longrightarrow \qquad f(x,t) = f_0 \exp(xt)$$

- We will temporarily assume the initial conditions are $f_0 = f(x, t=0) = 1$.

Simple 1D Exponential Growth Example

- Say we are interested in the concentration of a chemical which evolves in time. We will model this concentration as $f(x,t)$ where x is a rate parameter and t is time.
- We think $f(x,t)$ satisfies the differential equation or model:

$$\frac{df(x,t)}{dt} = xf(x,t) \qquad \Longrightarrow \qquad f(x,t) = f_0 \exp(xt)$$

- We will temporarily assume the initial conditions are $f_0 = f(x, t=0) = 1$.
- The system runs from $t=0$ to $t=5$ and we will measure $f(x,t)$ with error at $t=3.5$.

Simple 1D Exponential Growth Example

- Say we are interested in the concentration of a chemical which evolves in time. We will model this concentration as $f(x,t)$ where x is a rate parameter and t is time.
- We think $f(x,t)$ satisfies the differential equation or model:

$$\frac{df(x,t)}{dt} = xf(x,t) \qquad \Longrightarrow \qquad f(x,t) = f_0 \exp(xt)$$

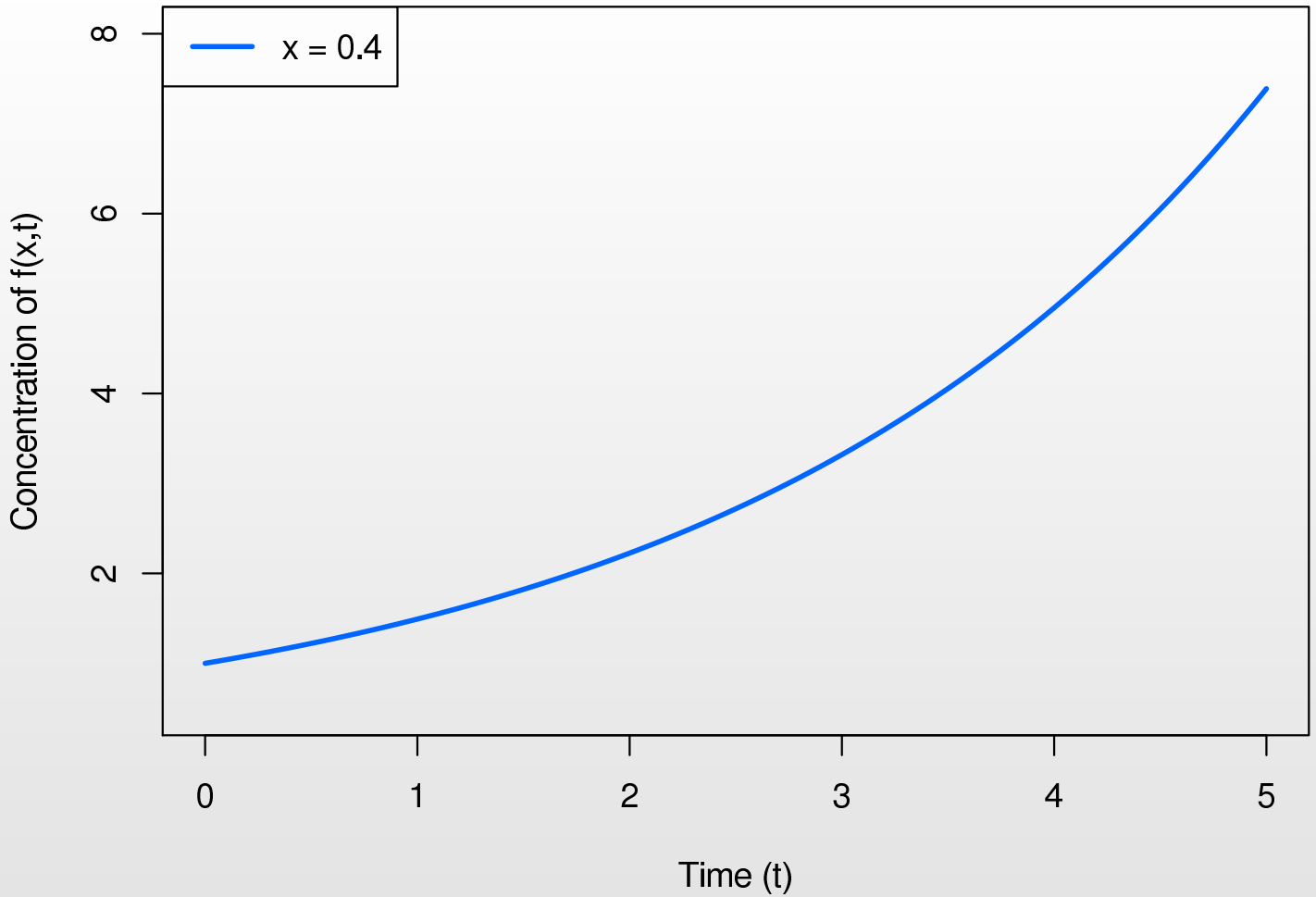
- We will temporarily assume the initial conditions are $f_0 = f(x, t=0) = 1$.
- The system runs from $t=0$ to $t=5$ and we will measure $f(x,t)$ with error at $t=3.5$.
- Model features an input parameter x which we want to learn about.

Simple 1D Exponential Growth Example

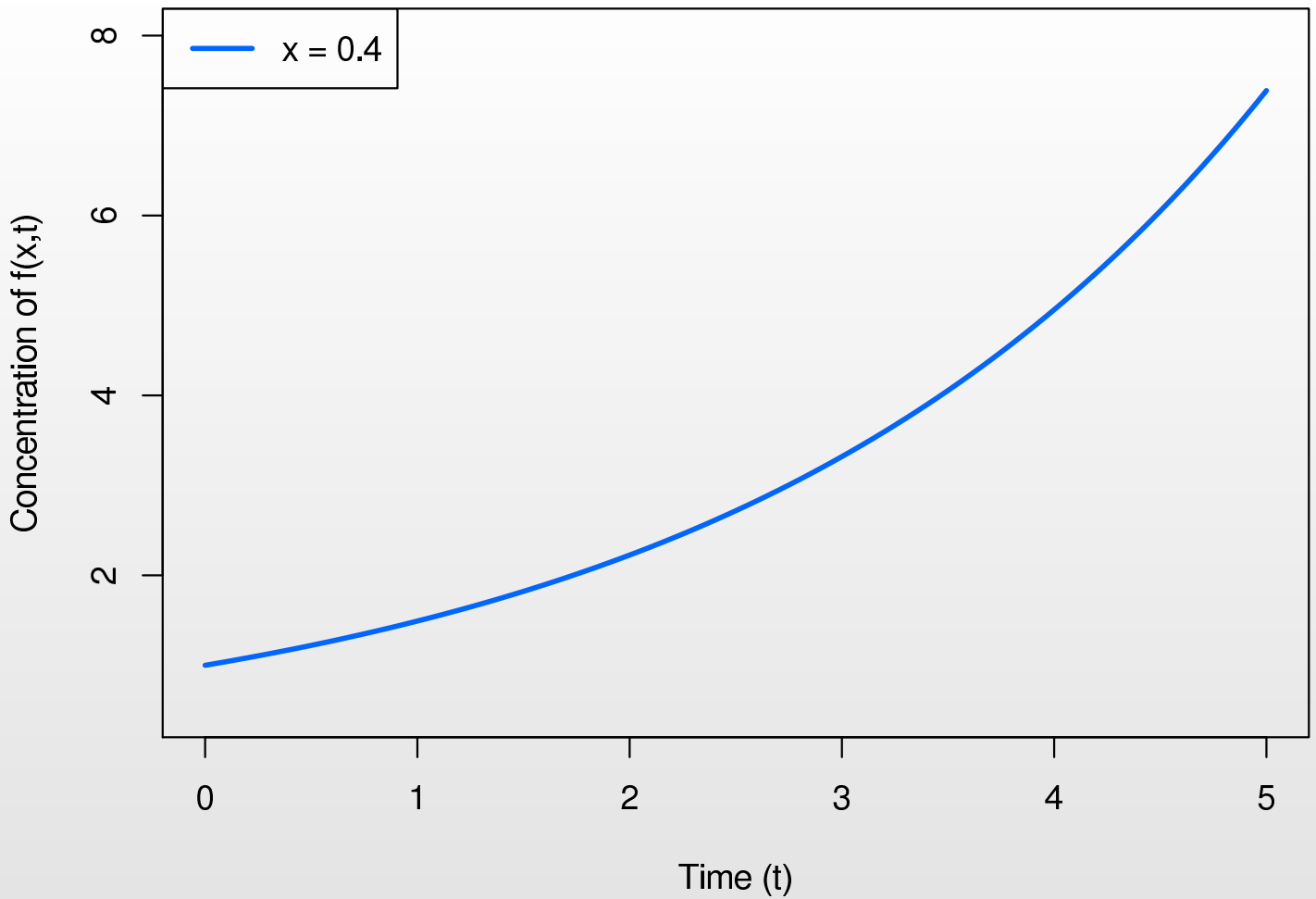
- Say we are interested in the concentration of a chemical which evolves in time. We will model this concentration as $f(x,t)$ where x is a rate parameter and t is time.
- We think $f(x,t)$ satisfies the differential equation or model:

$$\frac{df(x,t)}{dt} = xf(x,t) \qquad \Longrightarrow \qquad f(x,t) = f_0 \exp(xt)$$

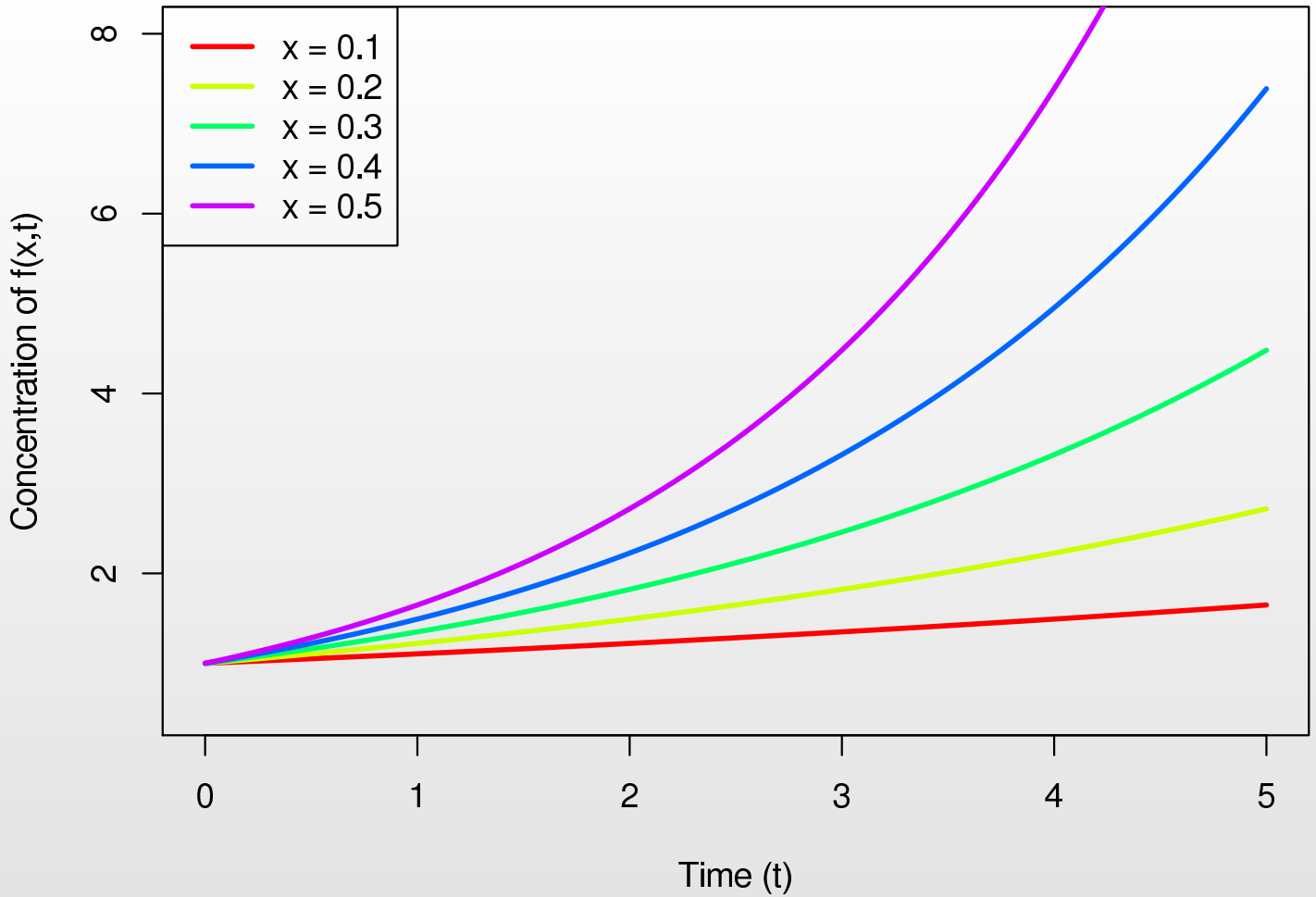
- We will temporarily assume the initial conditions are $f_0 = f(x, t=0) = 1$.
- The system runs from $t=0$ to $t=5$ and we will measure $f(x,t)$ with error at $t=3.5$.
- Model features an input parameter x which we want to learn about.
- Note that normally we would not have the analytic solution for $f(x,t)$.



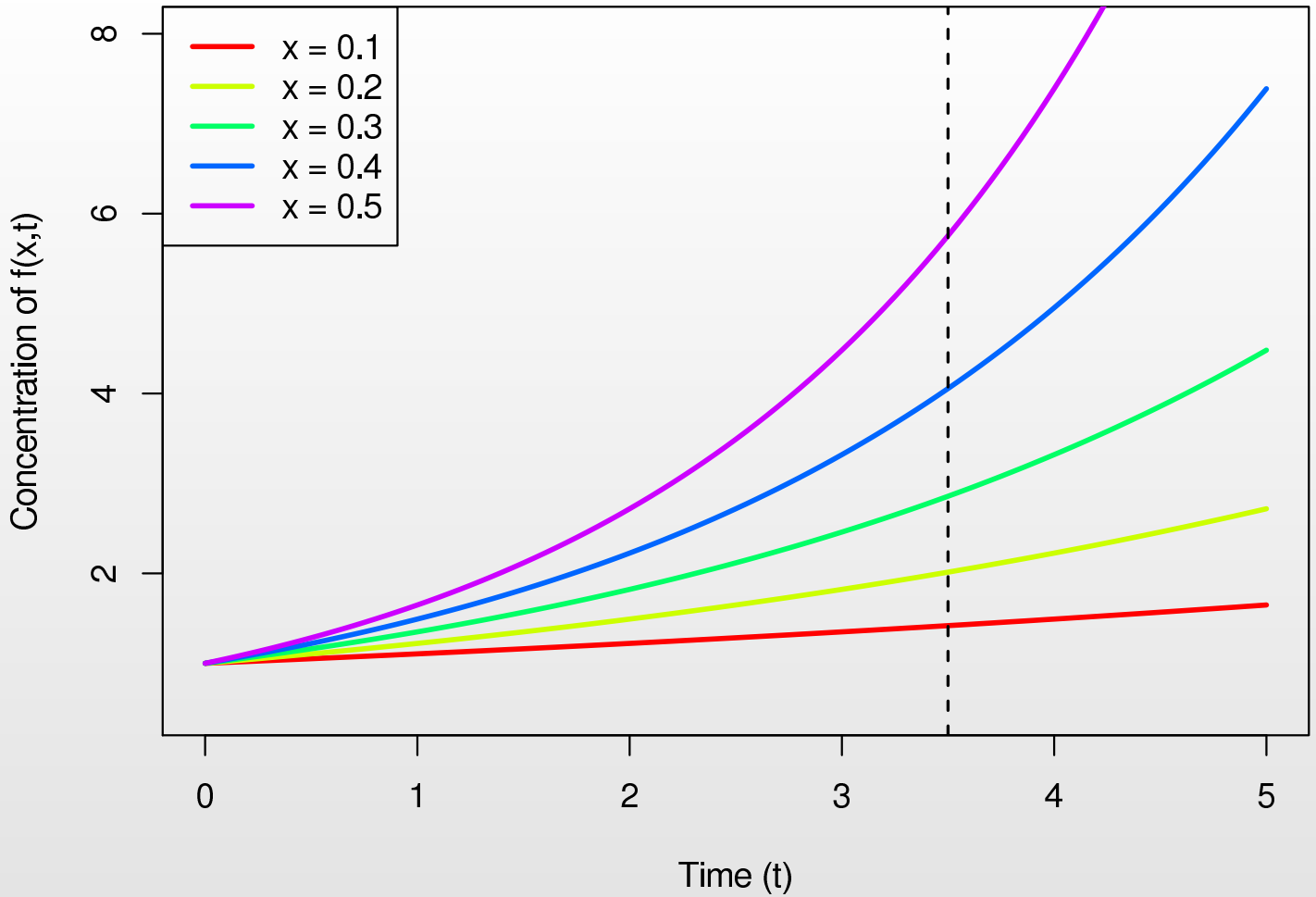
- One “model run” with the input parameter $x = 0.4$



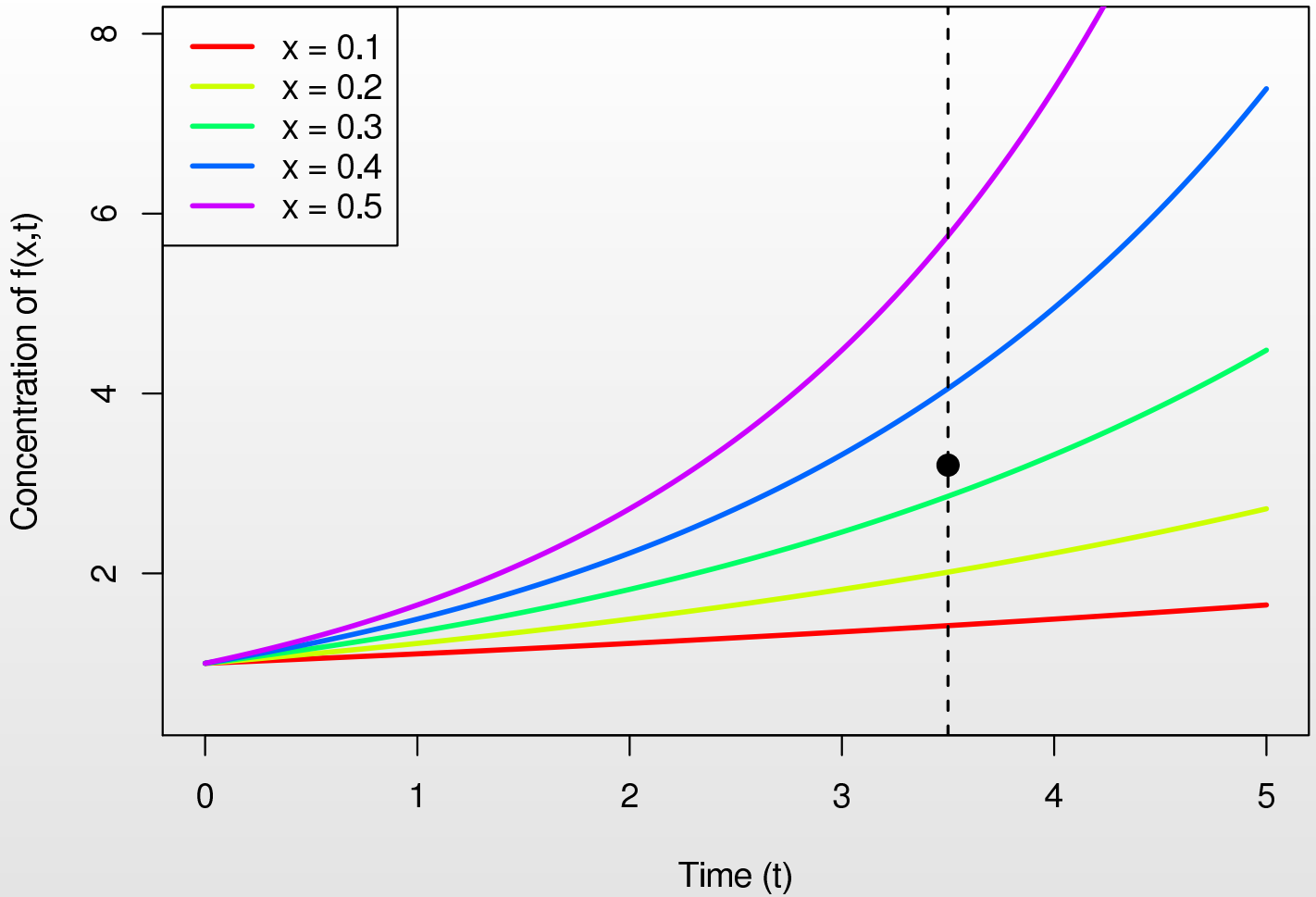
- One “model run” with the input parameter $x = 0.4$
- If we did not know the analytic solution for $f(x, t)$ this would be generated by numerically solving the differential equation.



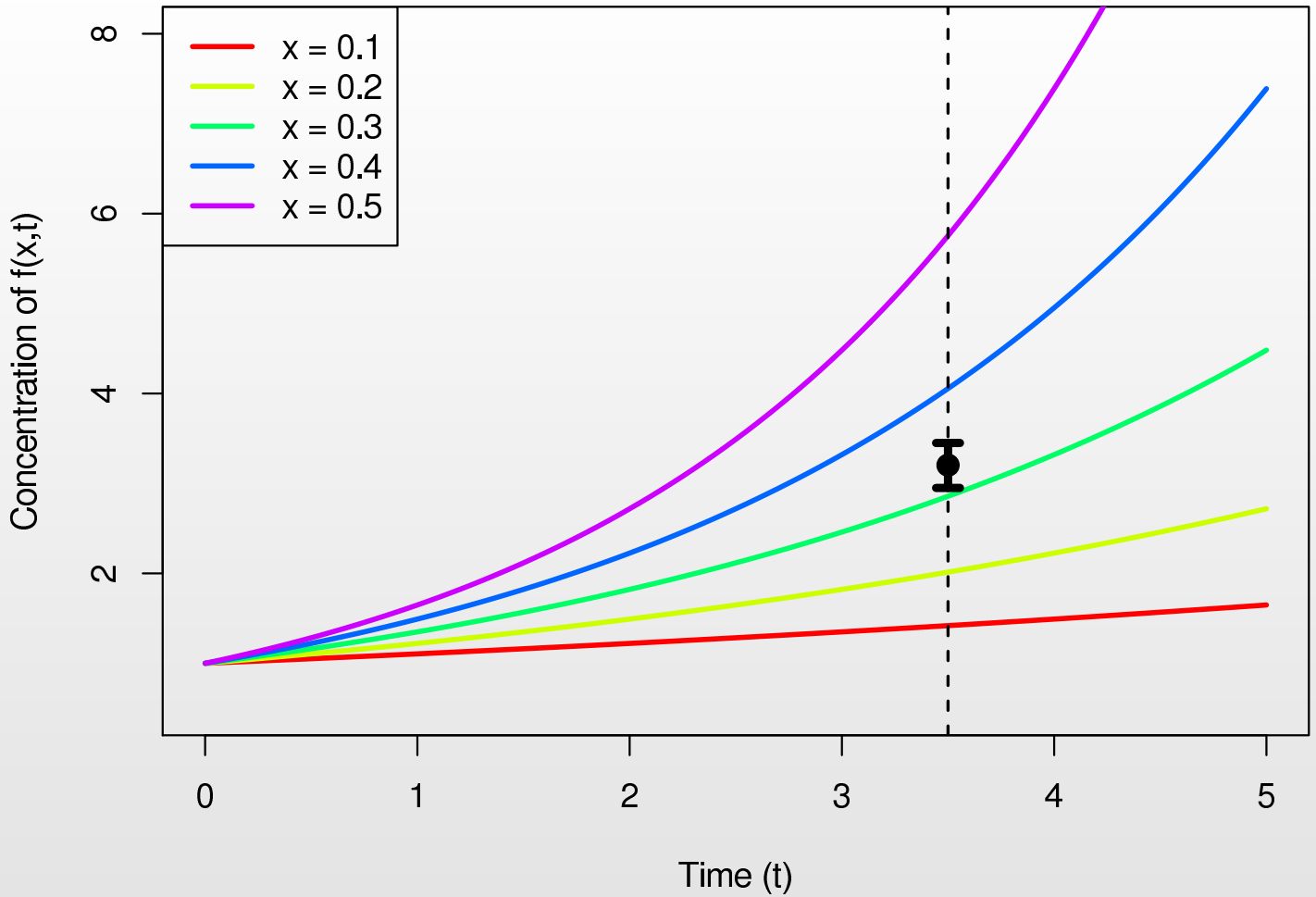
- Five model runs with the input parameter varying from $x = 0.1$ to $x = 0.5$



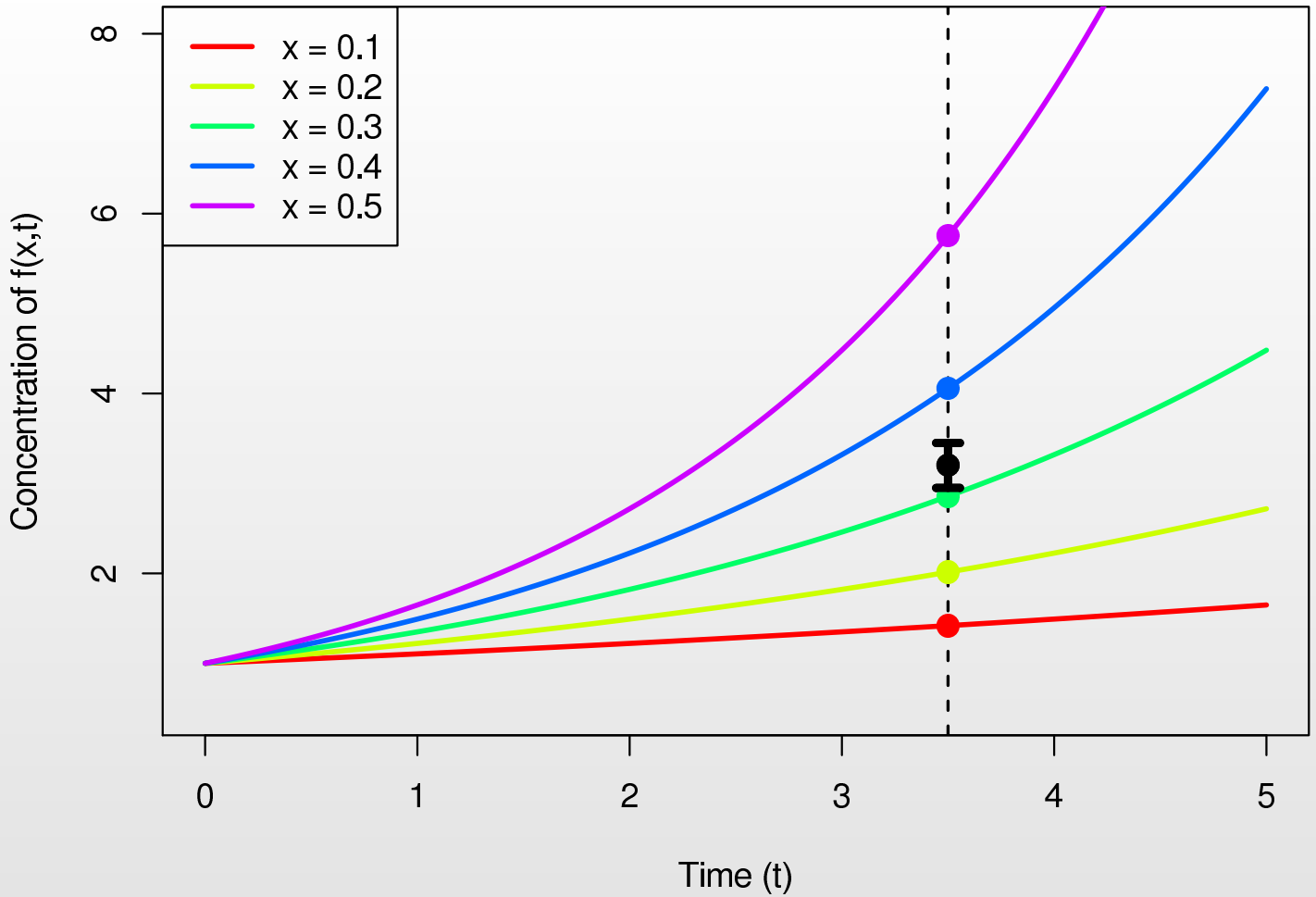
- Five model runs with the input parameter varying from $x = 0.1$ to $x = 0.5$
- We are going to measure $f(x, t)$ at $t = 3.5$



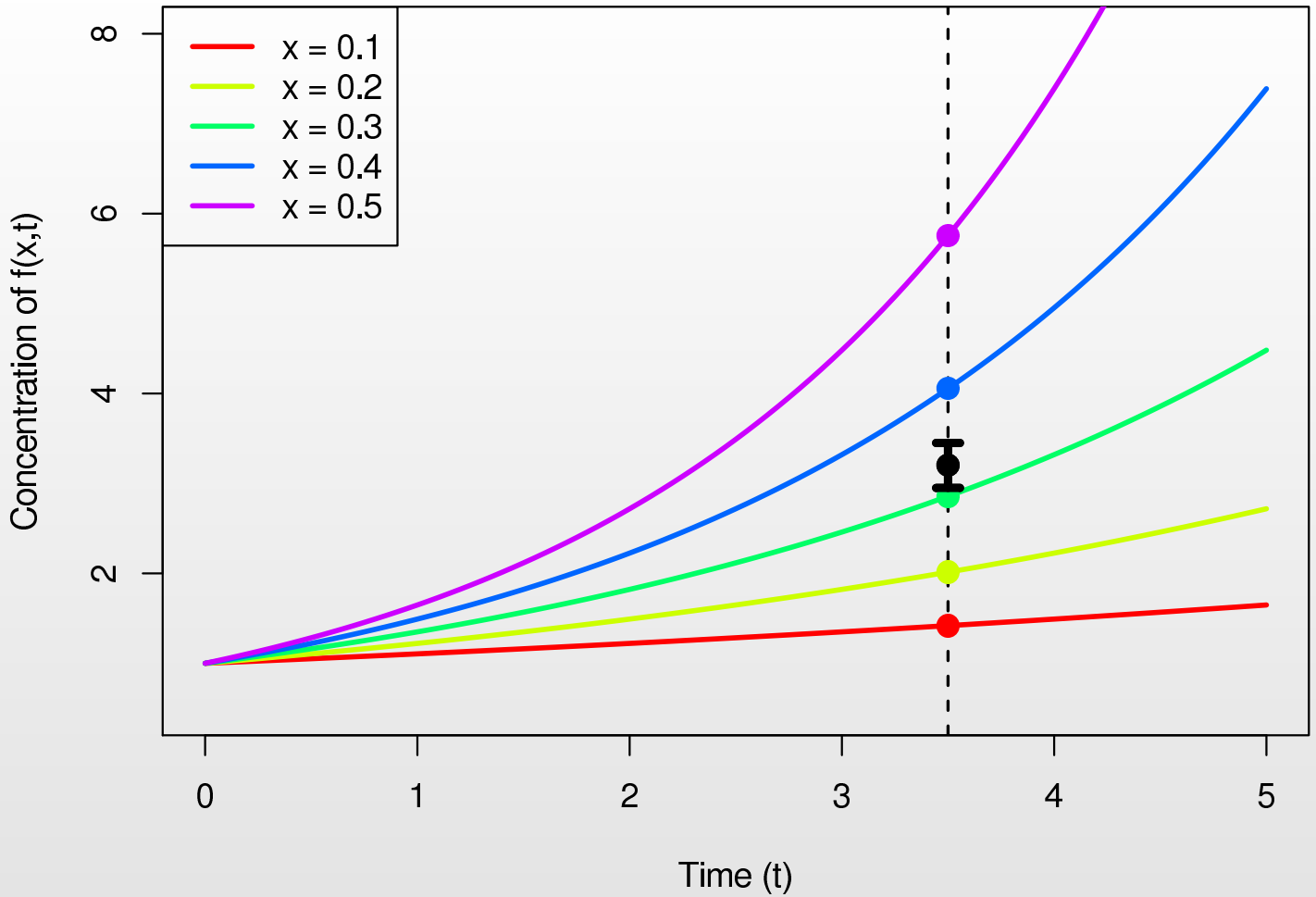
- Five model runs with the input parameter varying from $x = 0.1$ to $x = 0.5$
- We are going to measure $f(x, t)$ at $t = 3.5$



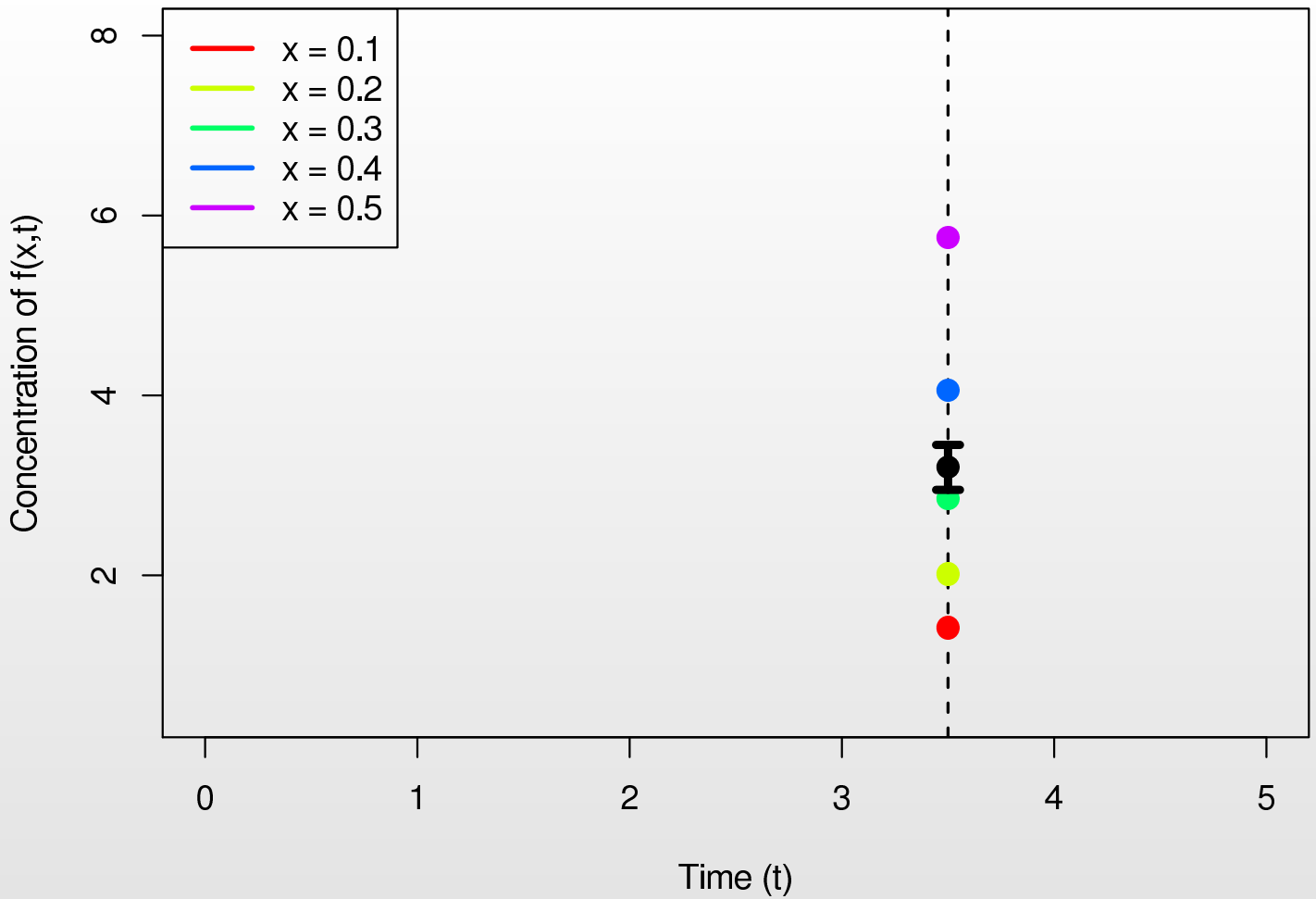
- Five model runs with the input parameter varying from $x = 0.1$ to $x = 0.5$
- We are going to measure $f(x, t)$ at $t = 3.5$
- The measurement is not a point but comes with measurement error.



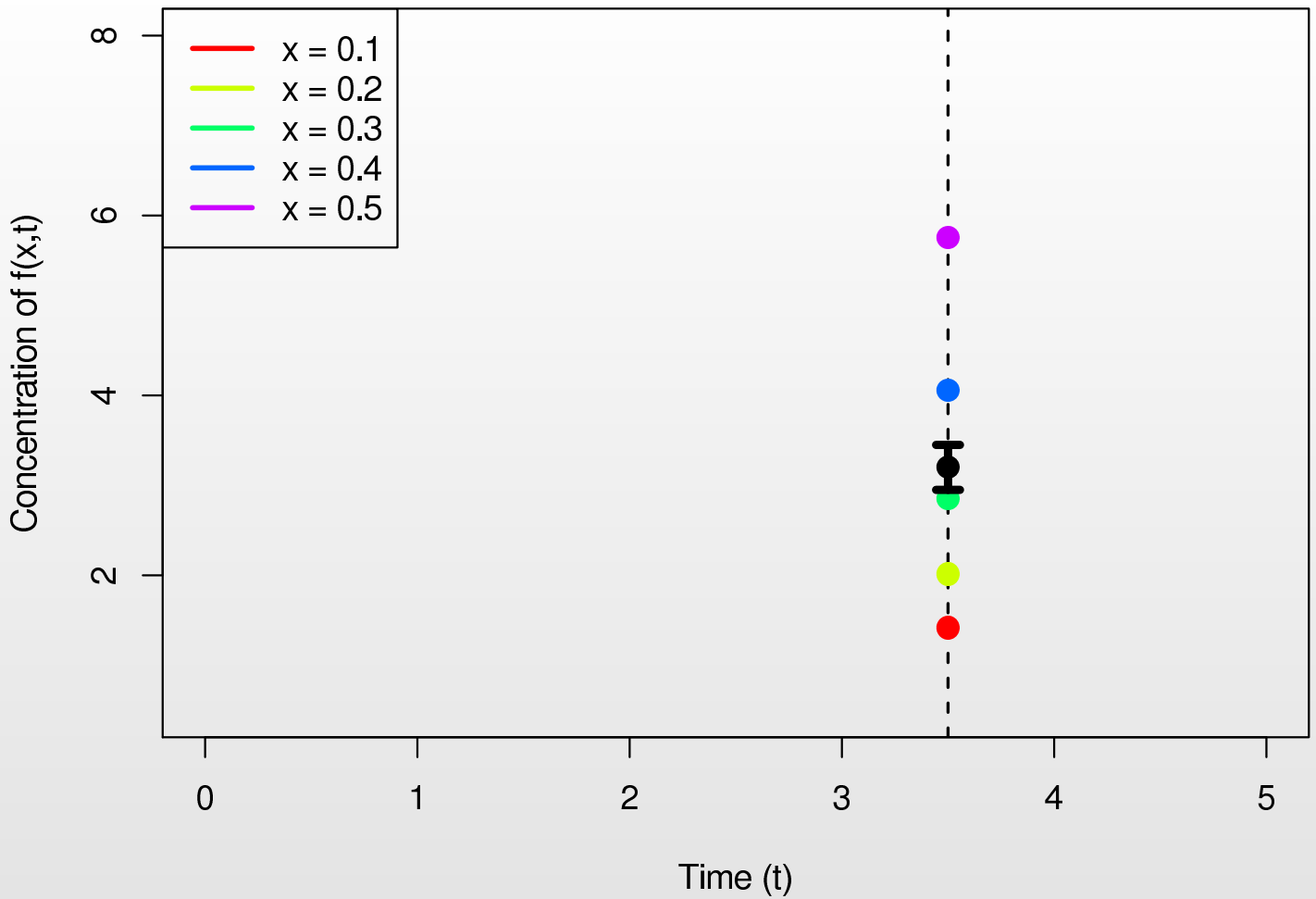
- Major question: which values of x ensure the output $f(x, t = 3.5)$ is consistent with the observations?



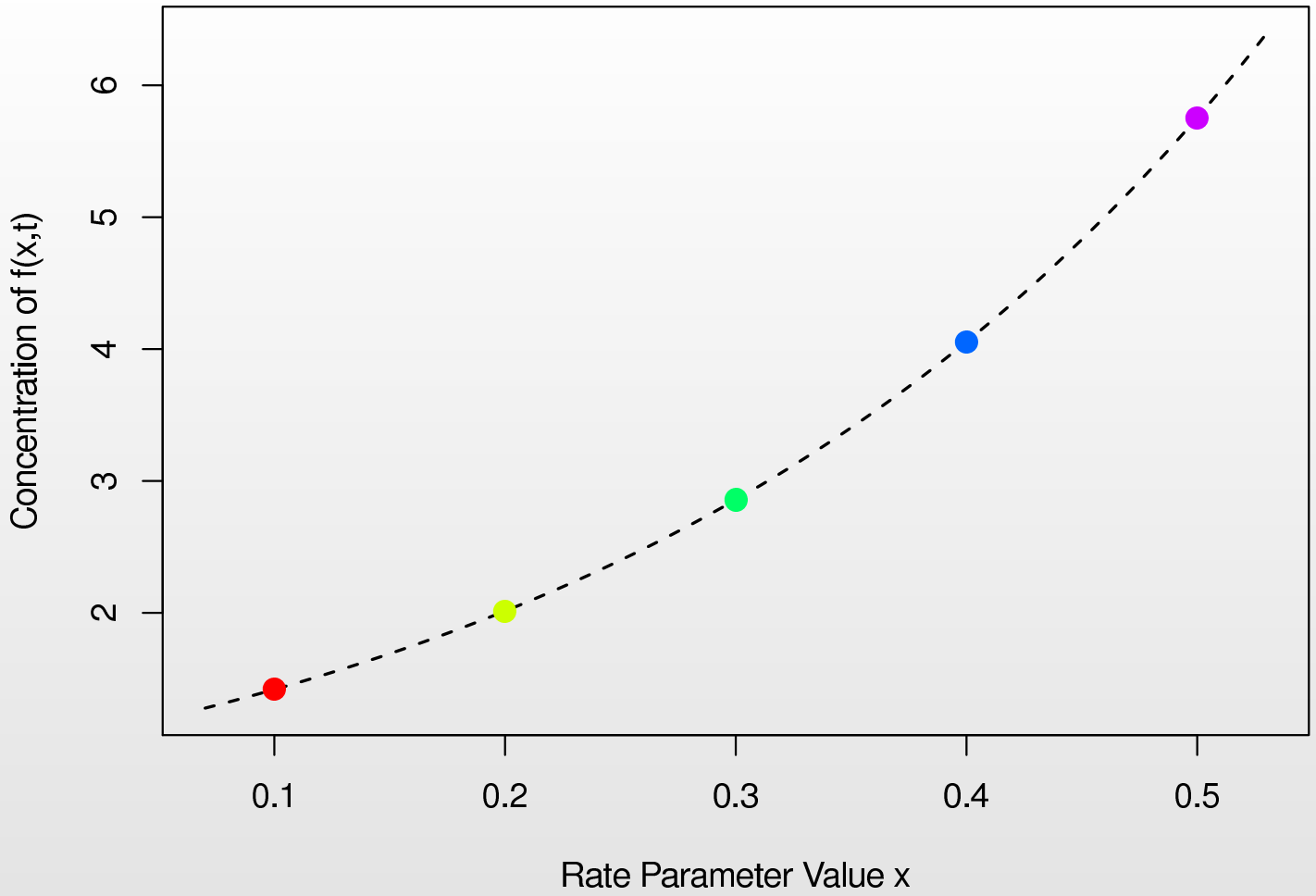
- Major question: which values of x ensure the output $f(x, t = 3.5)$ is consistent with the observations?
- It would seem that x has to be at least between 0.3 and 0.4.



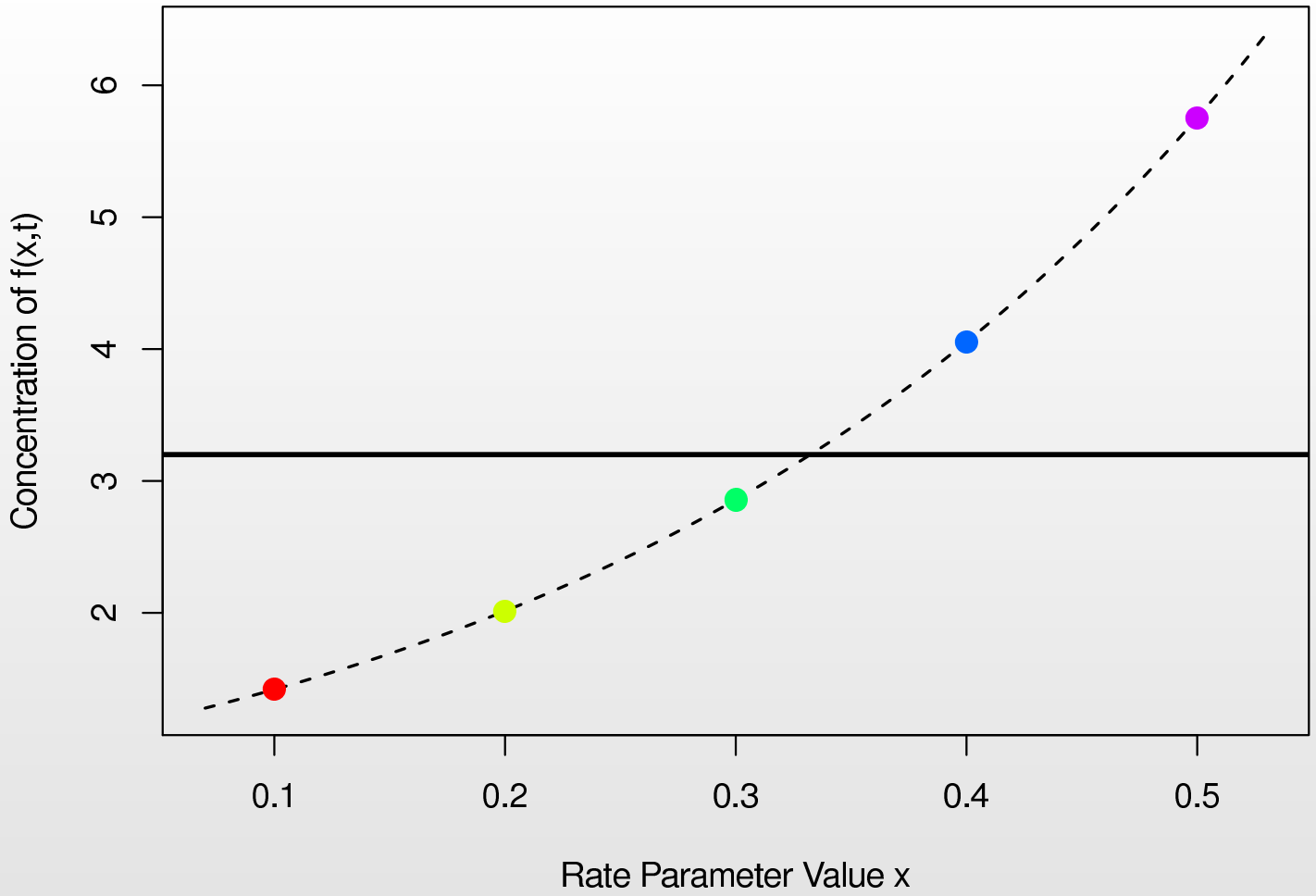
- To answer this, we can now discard other values of $f(x,t)$ and think of $f(x, t = 3.5)$ as a function of x only.



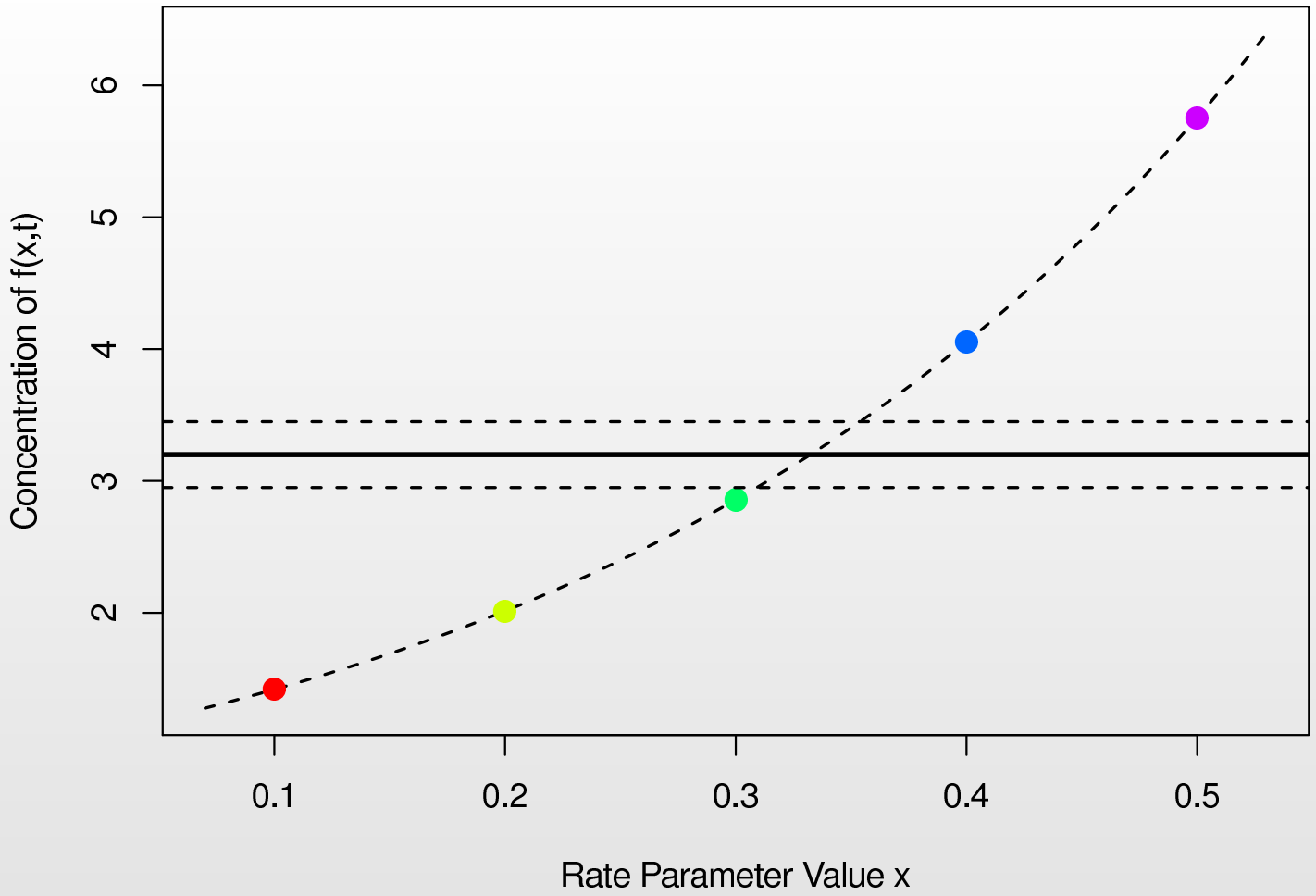
- To answer this, we can now discard other values of $f(x,t)$ and think of $f(x, t = 3.5)$ as a function of x only.
- That is take $f(x) \equiv f(x, t = 3.5)$



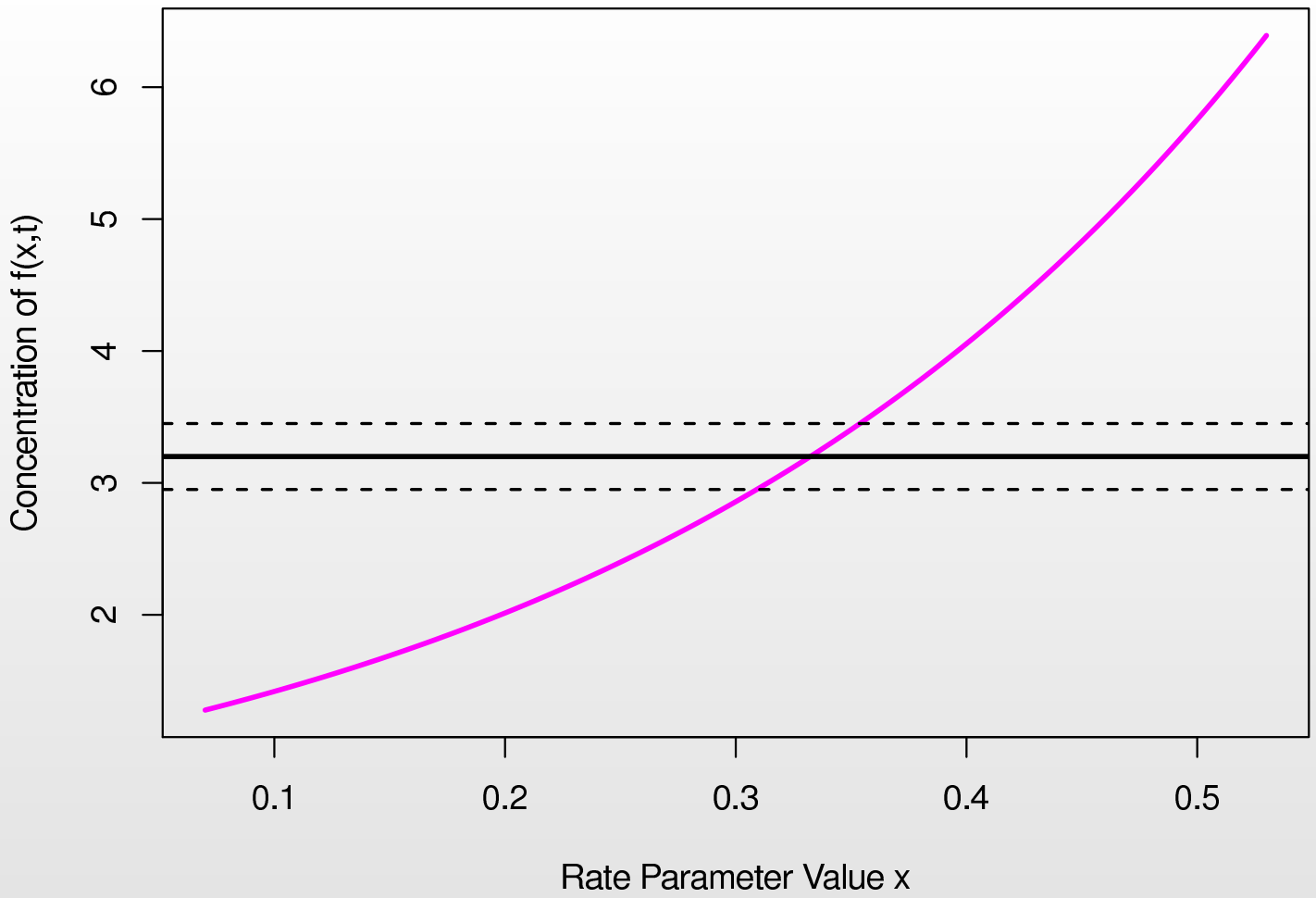
- We can now plot the concentration $f(x)$ as a function of the input parameter x.



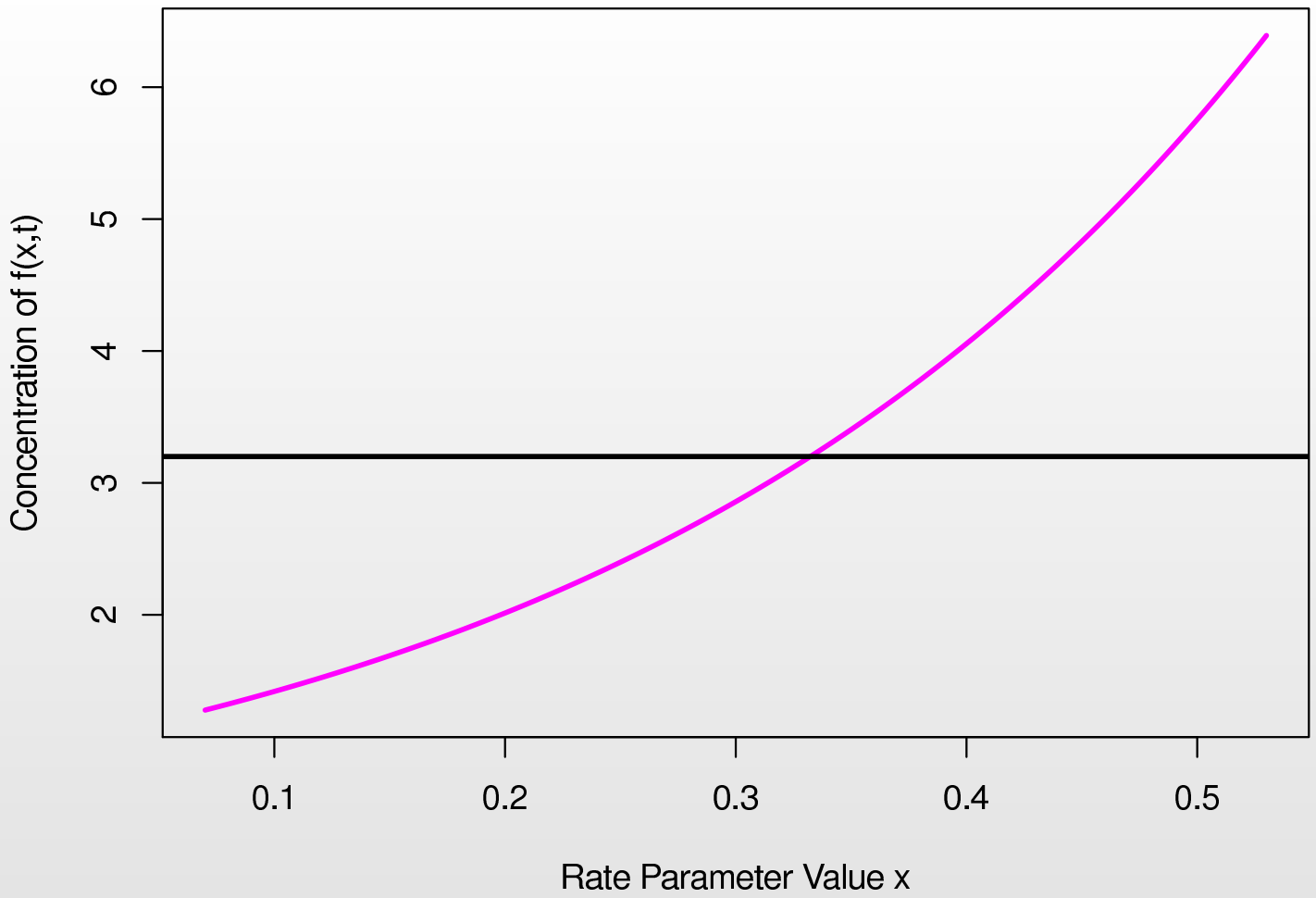
- We can now plot the concentration $f(x)$ as a function of the input parameter x.
- Black horizontal line: the observed measurement of f



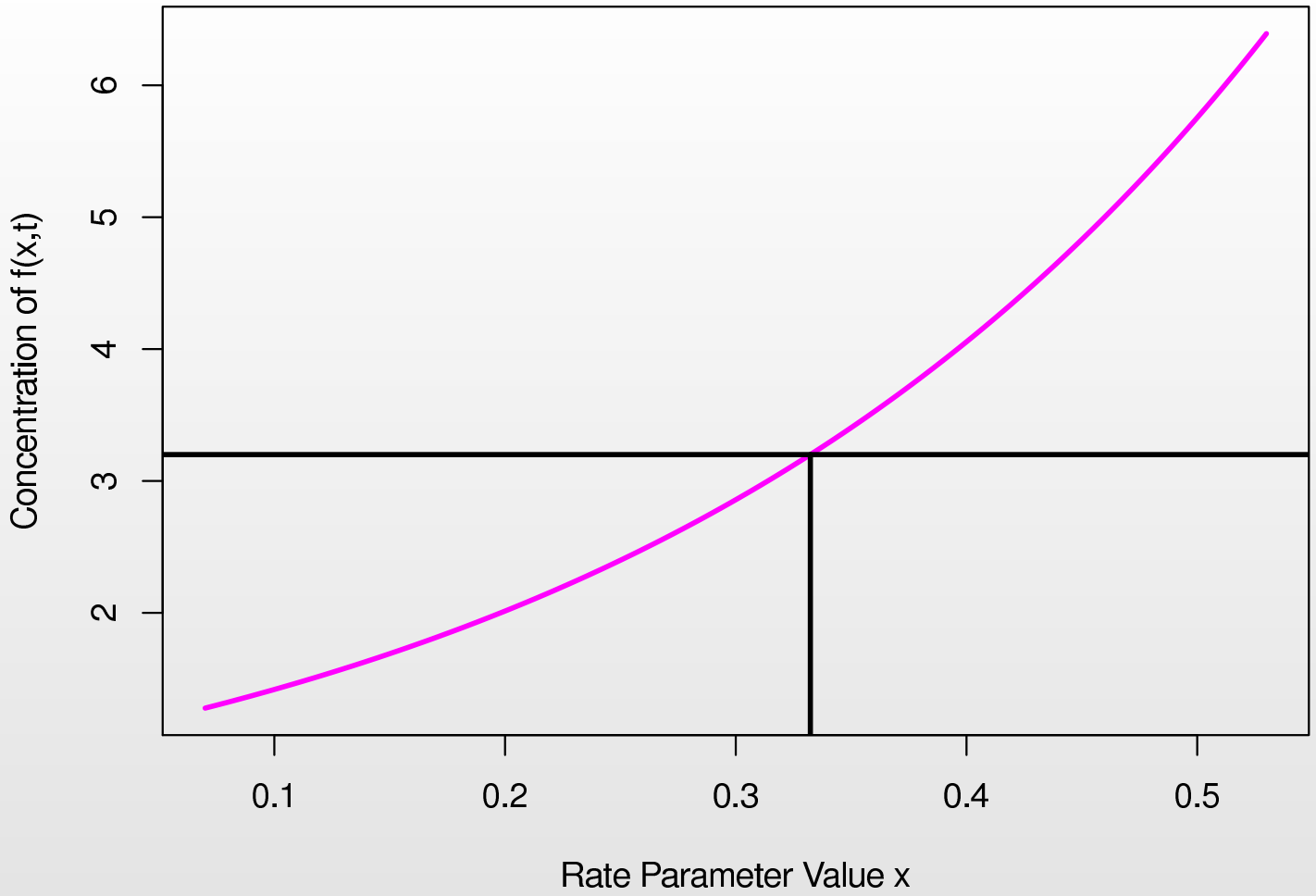
- We can now plot the concentration $f(x)$ as a function of the input parameter x.
- Black horizontal line: the observed measurement of f
- Dashed horizontal lines: the measurement errors



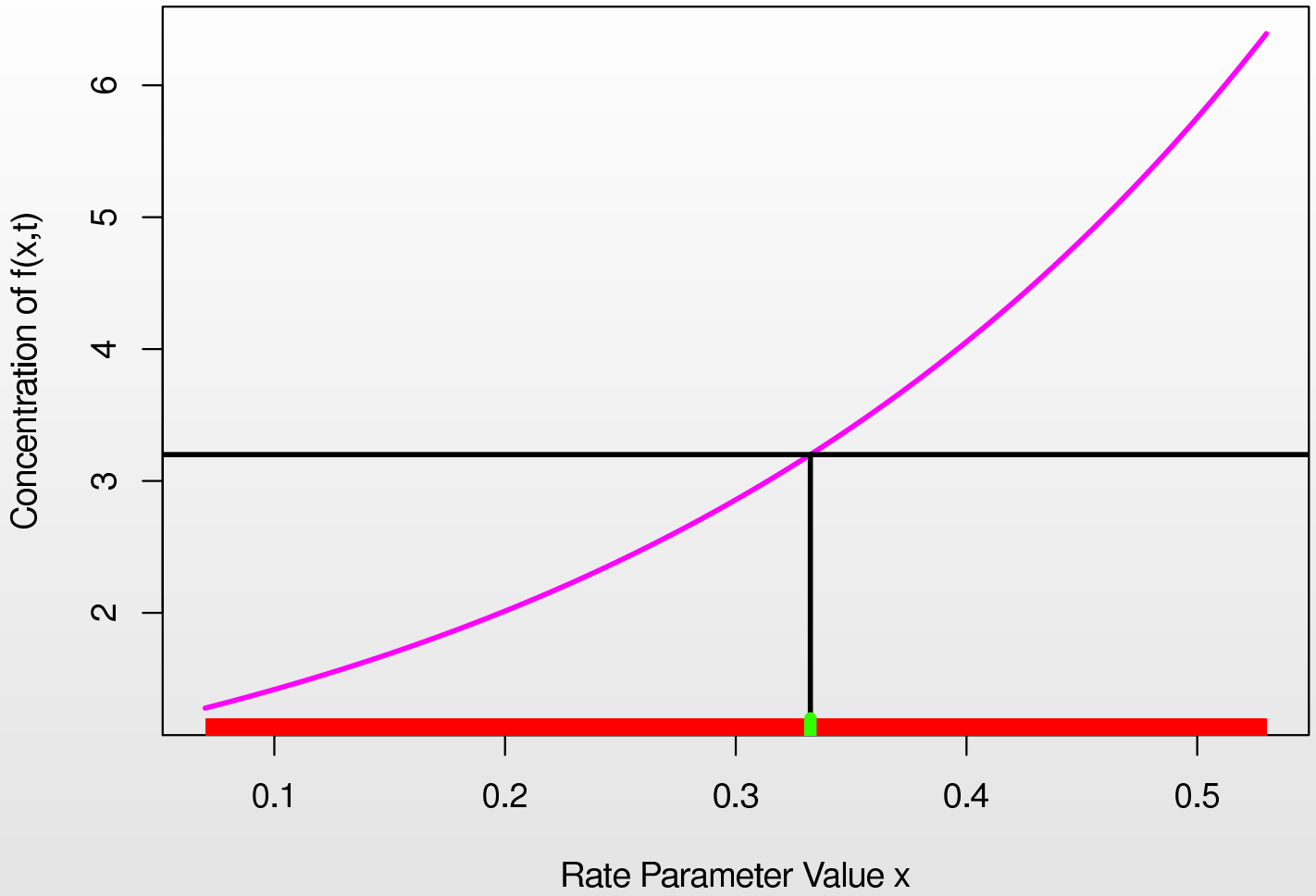
- If we know the analytical expression for $f(x) = \exp(3.5x)$, then we can identify the values of x of interest.



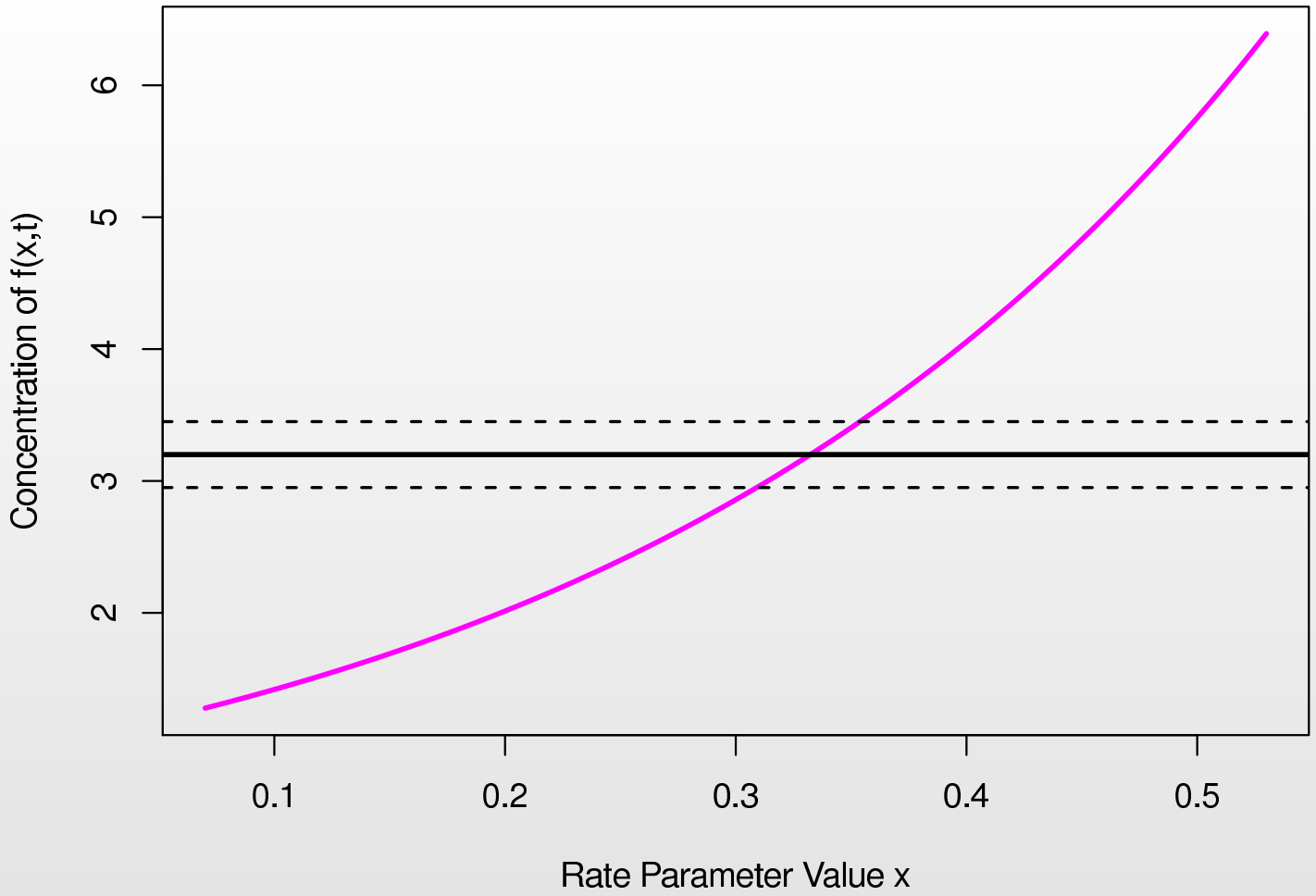
- If we know the analytical expression for $f(x) = \exp(3.5x)$, then we can identify the values of x of interest.
- Ignoring the measurement error would lead to a single value for x but this is incorrect: we have to include the errors.



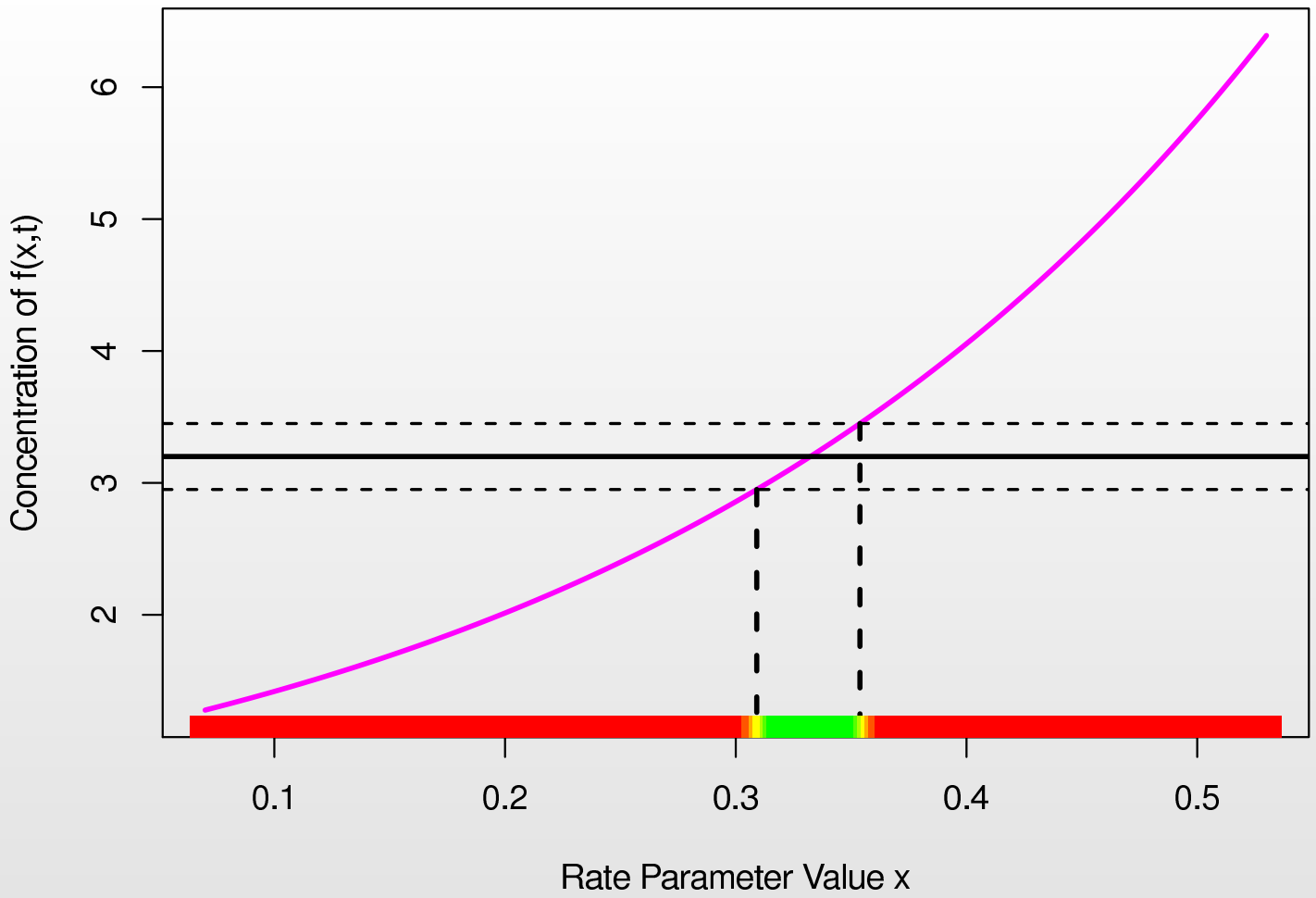
- If we know the analytical expression for $f(x) = \exp(3.5x)$, then we can identify the values of x of interest.
- Ignoring the measurement error would lead to a single value for x but this is incorrect: we have to include the errors.



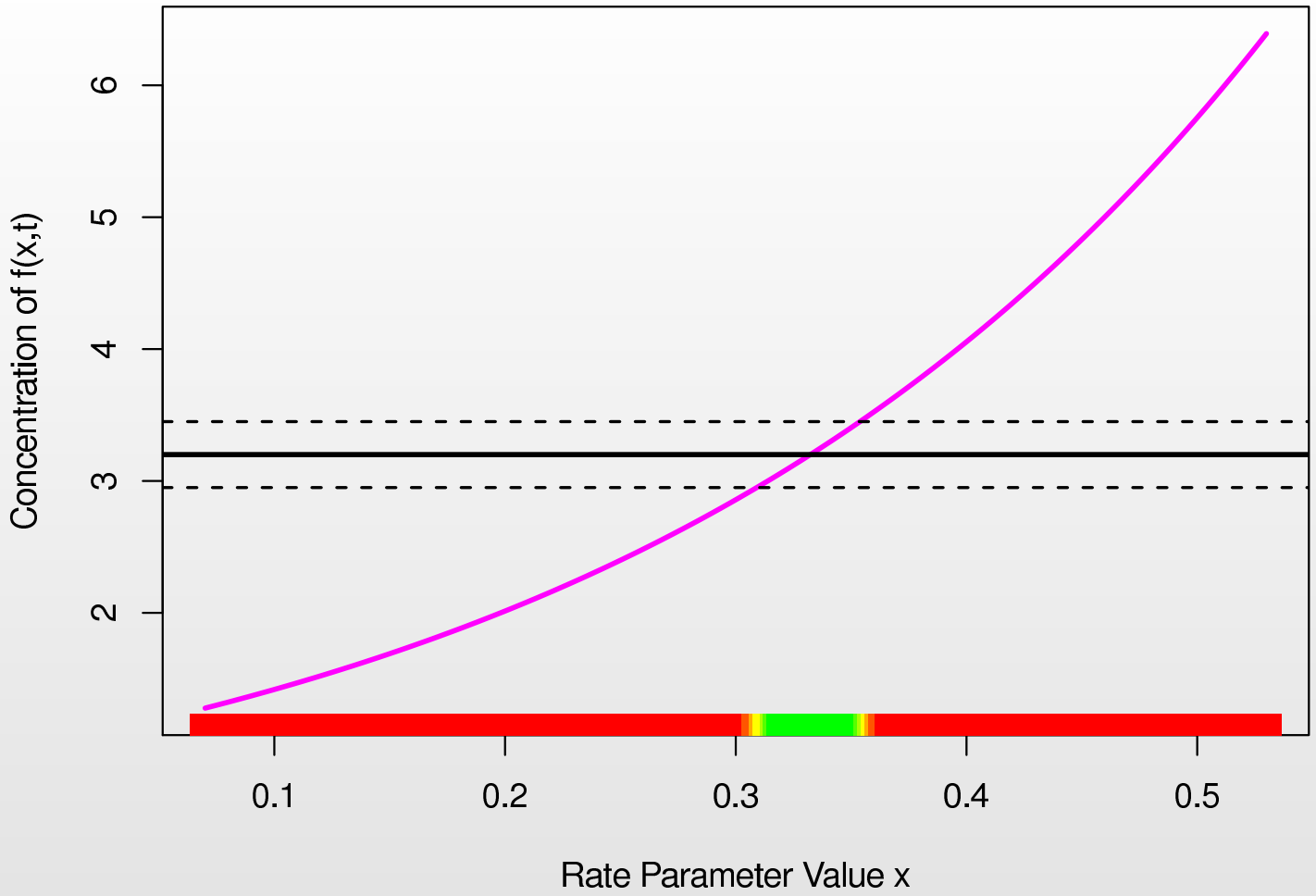
- If we know the analytical expression for $f(x) = \exp(3.5x)$, then we can identify the values of x of interest.
- Ignoring the measurement error would lead to a single value for x but this is incorrect: we have to include the errors.



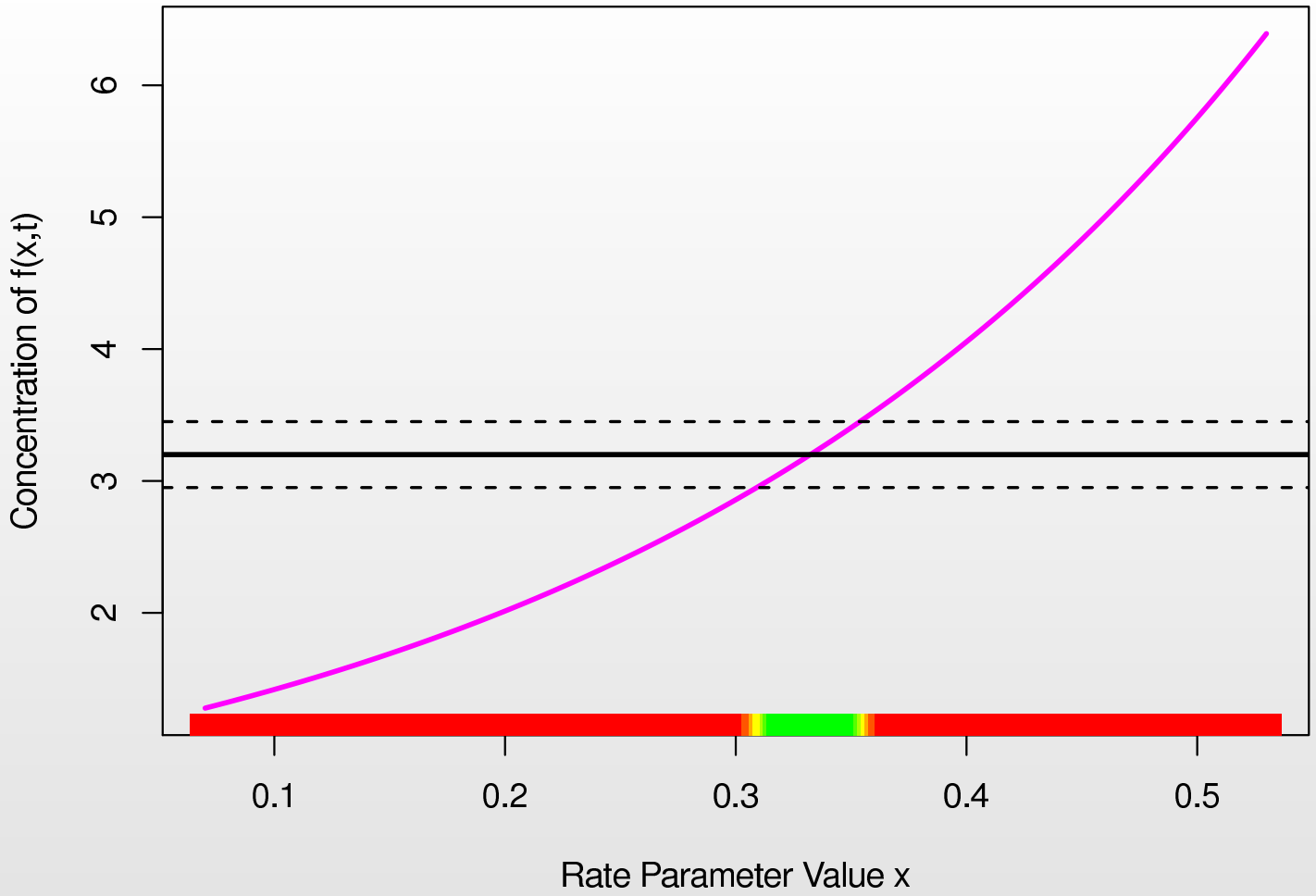
- Uncertainty in the measurement of $f(x, t = 3.5)$ leads to uncertainty in the inferred values of x.



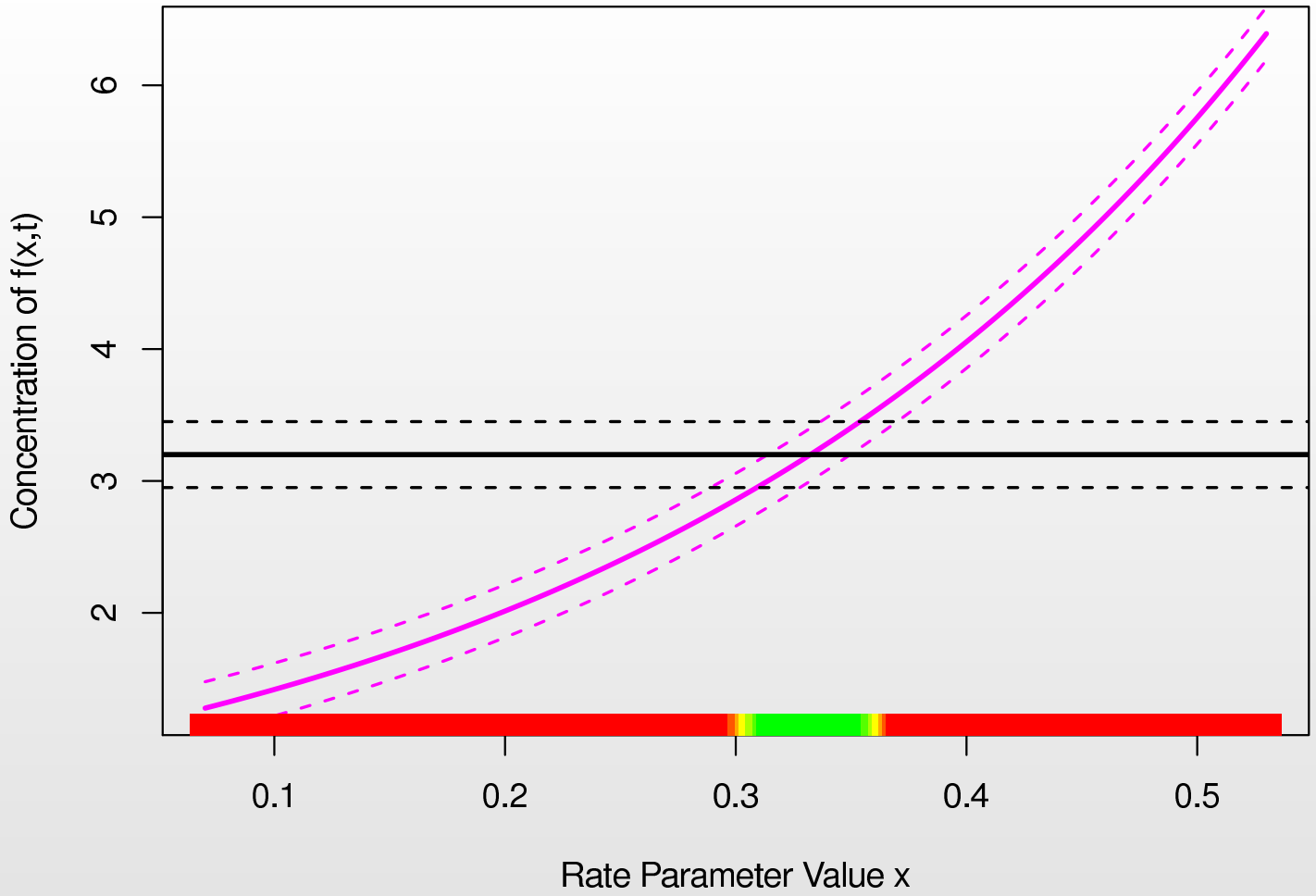
- Uncertainty in the measurement of $f(x, t = 3.5)$ leads to uncertainty in the inferred values of x.
- Hence we see a range (green/yellow) of possible values of x consistent with the measurements, with all the implausible values of x in red.



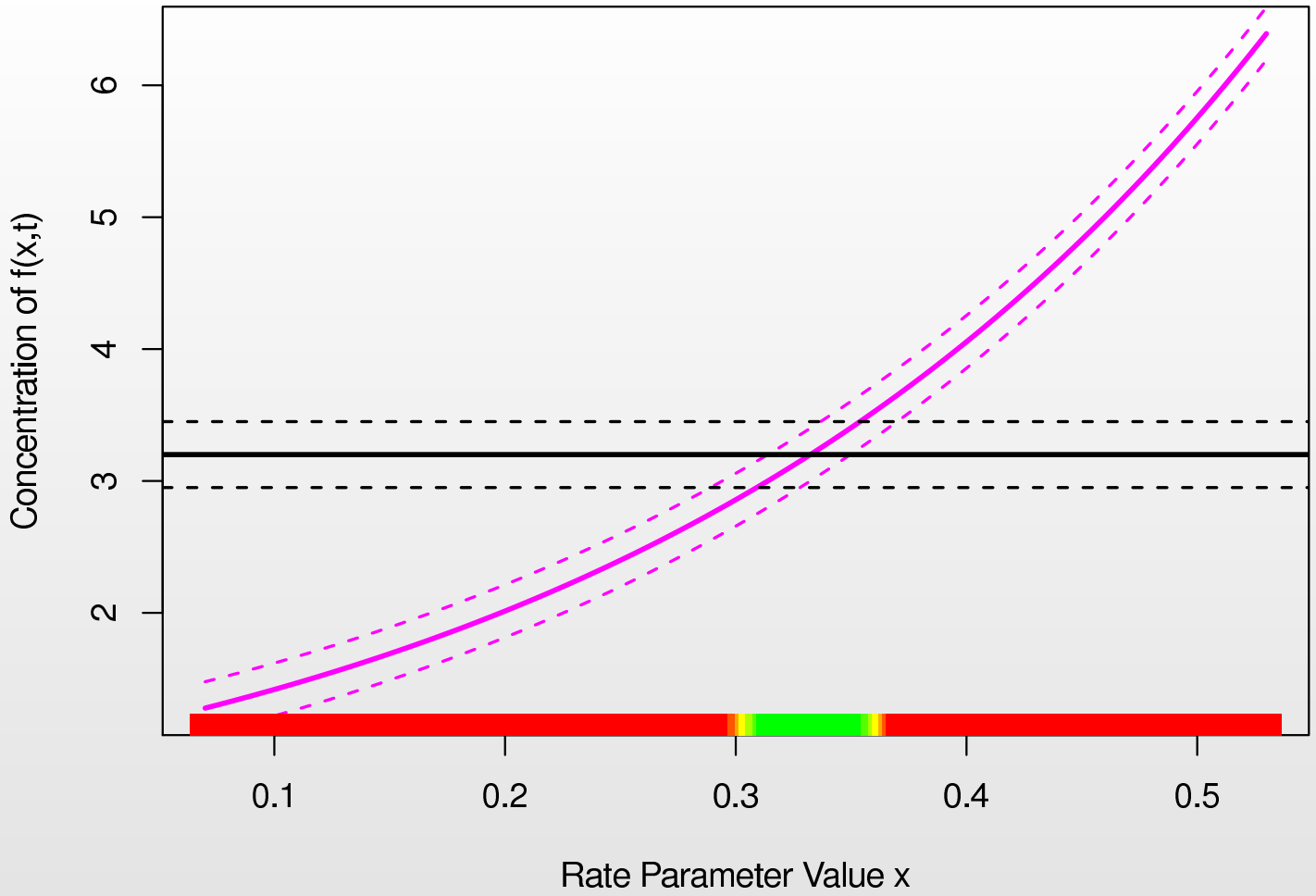
- Another important form of uncertainty is that of model discrepancy related to how accurate we believe the model to be.



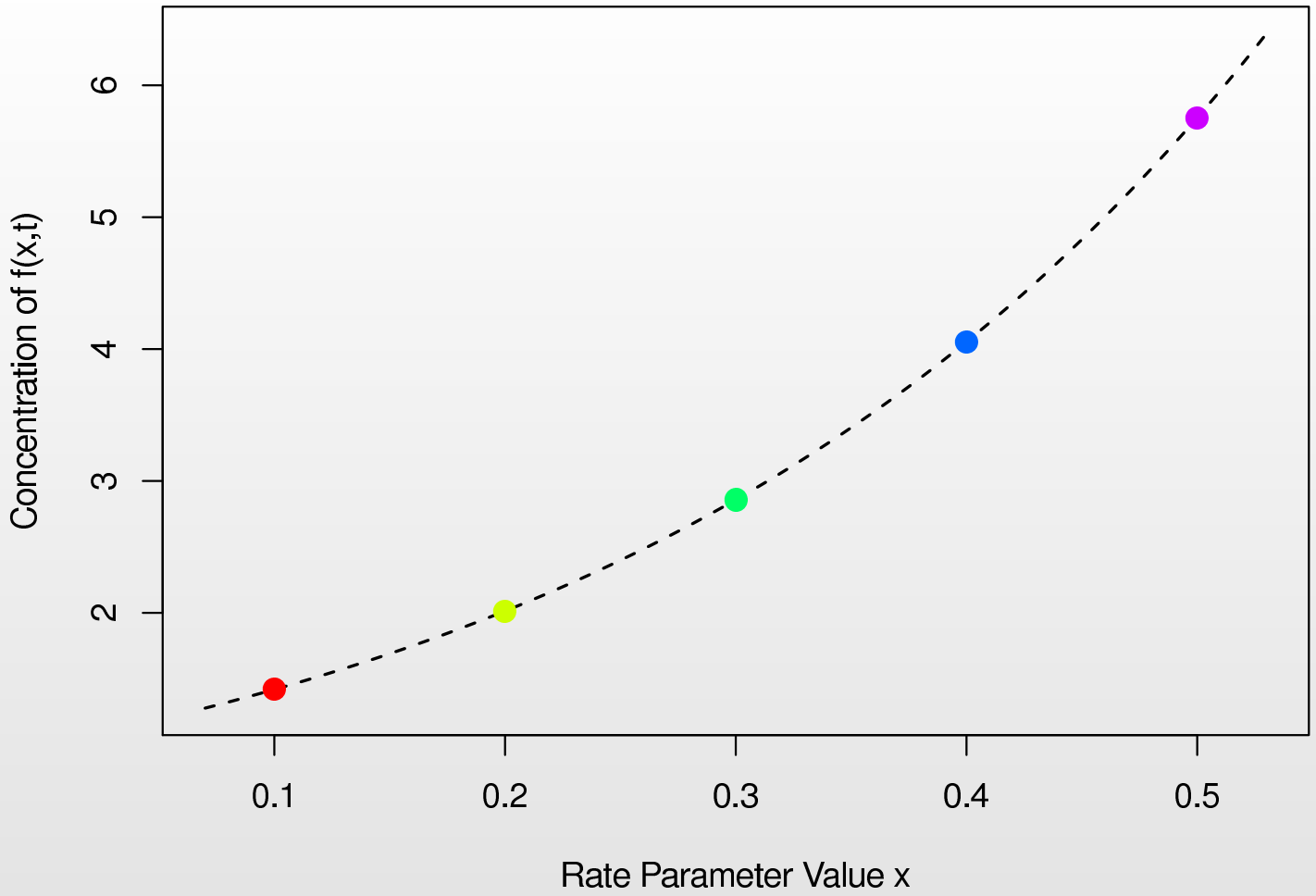
- Another important form of uncertainty is that of model discrepancy related to how accurate we believe the model to be.
- This uncertainty arises from many issues e.g. is the form of the model (the differential equation) appropriate, is the model a simplified description of a more complex system, is there uncertainty in the initial conditions etc?



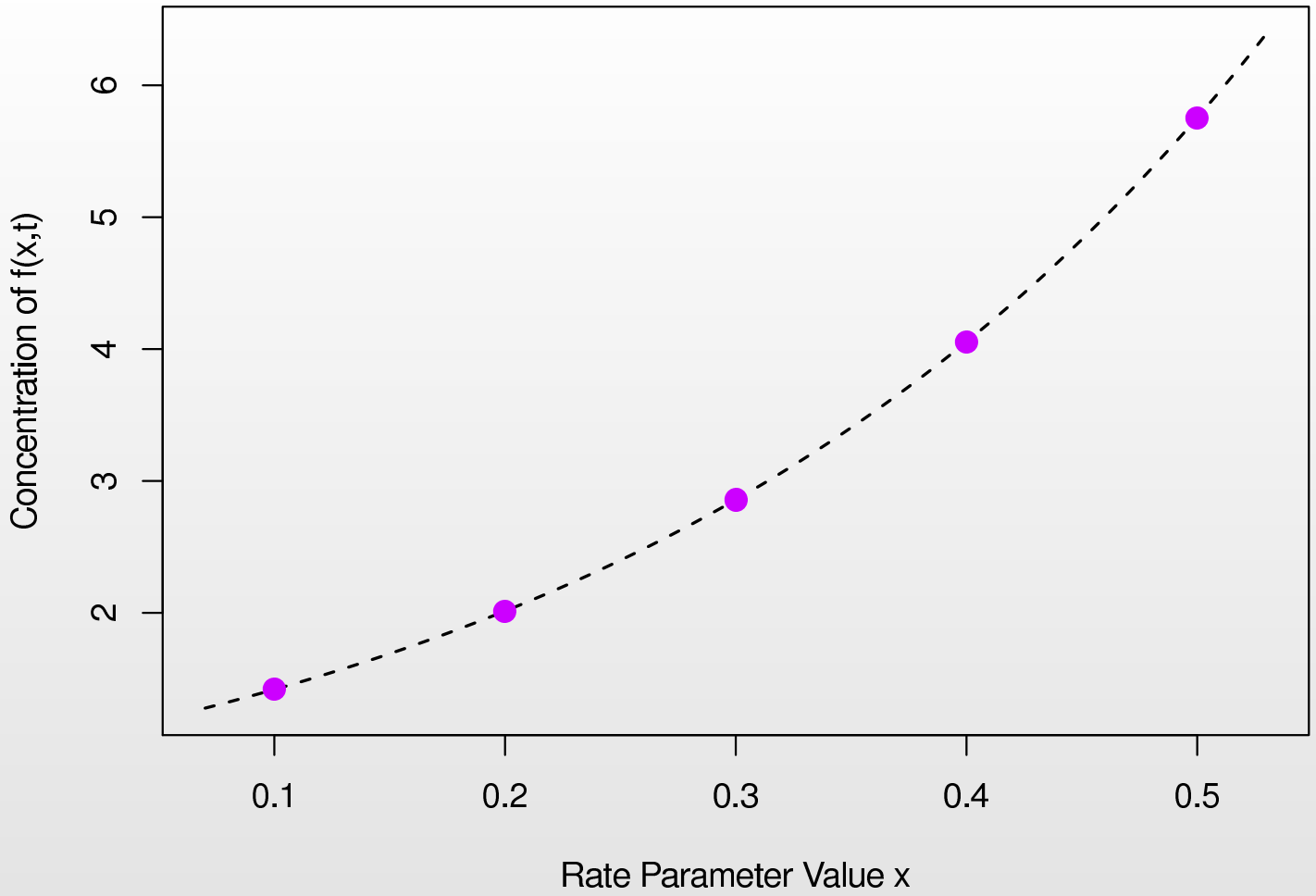
- Model discrepancy is represented as uncertainty around the model output $f(x)$ itself: here the purple dashed lines.



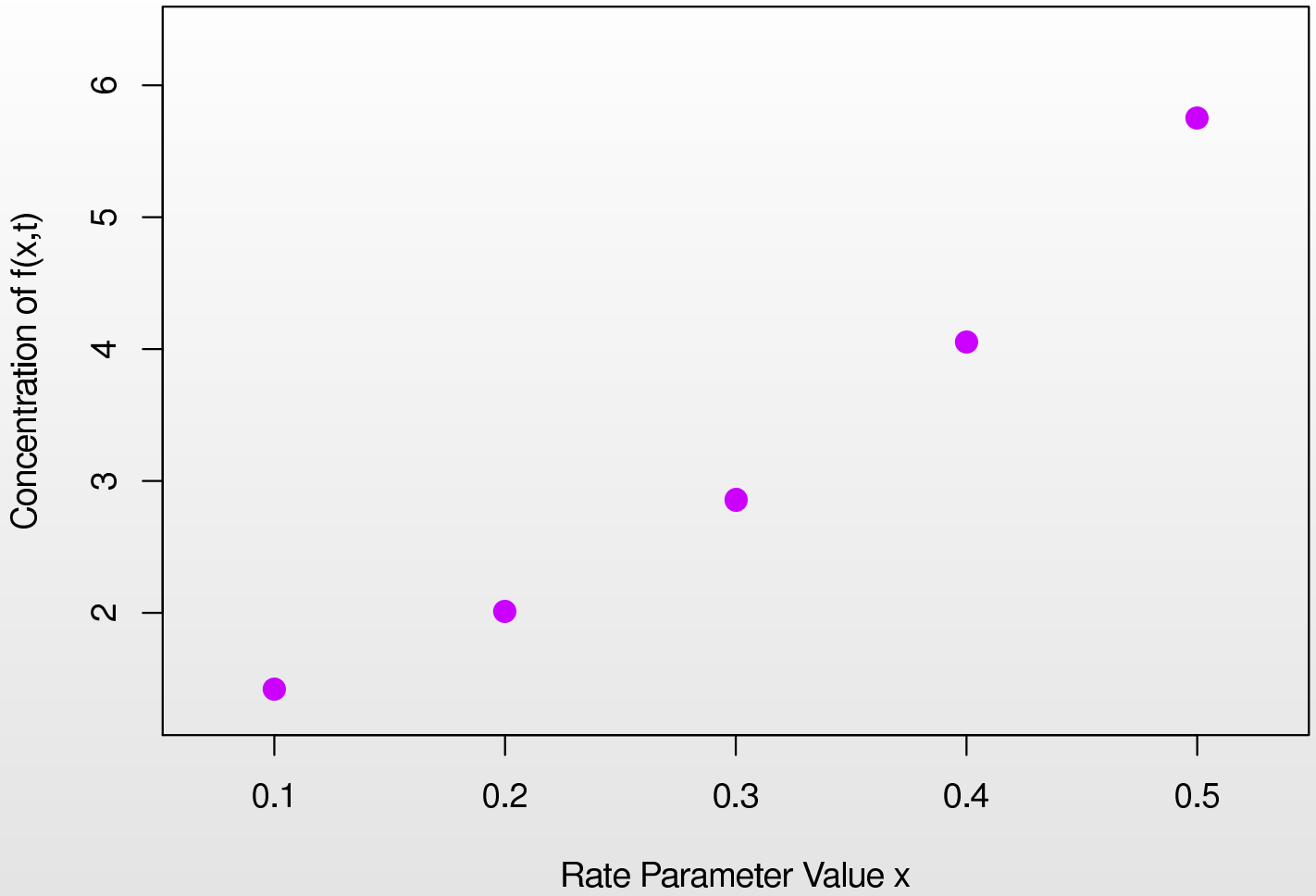
- Model discrepancy is represented as uncertainty around the model output $f(x)$ itself: here the purple dashed lines.
- This results in more uncertainty in x, and hence a larger range of x values.



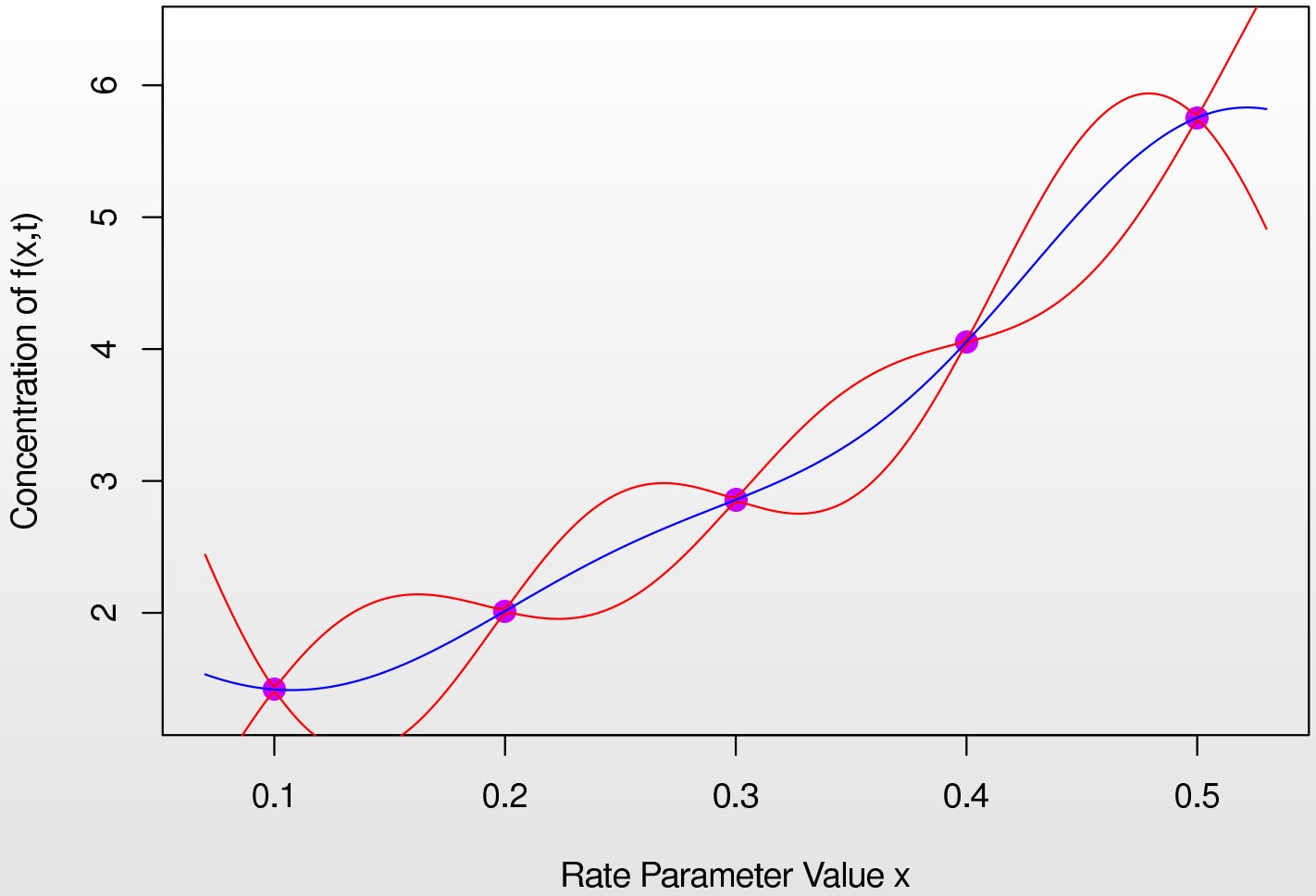
- Consider the graph of $f(x)$: in general we do not have the analytic solution of $f(x)$, here given by the dashed line.



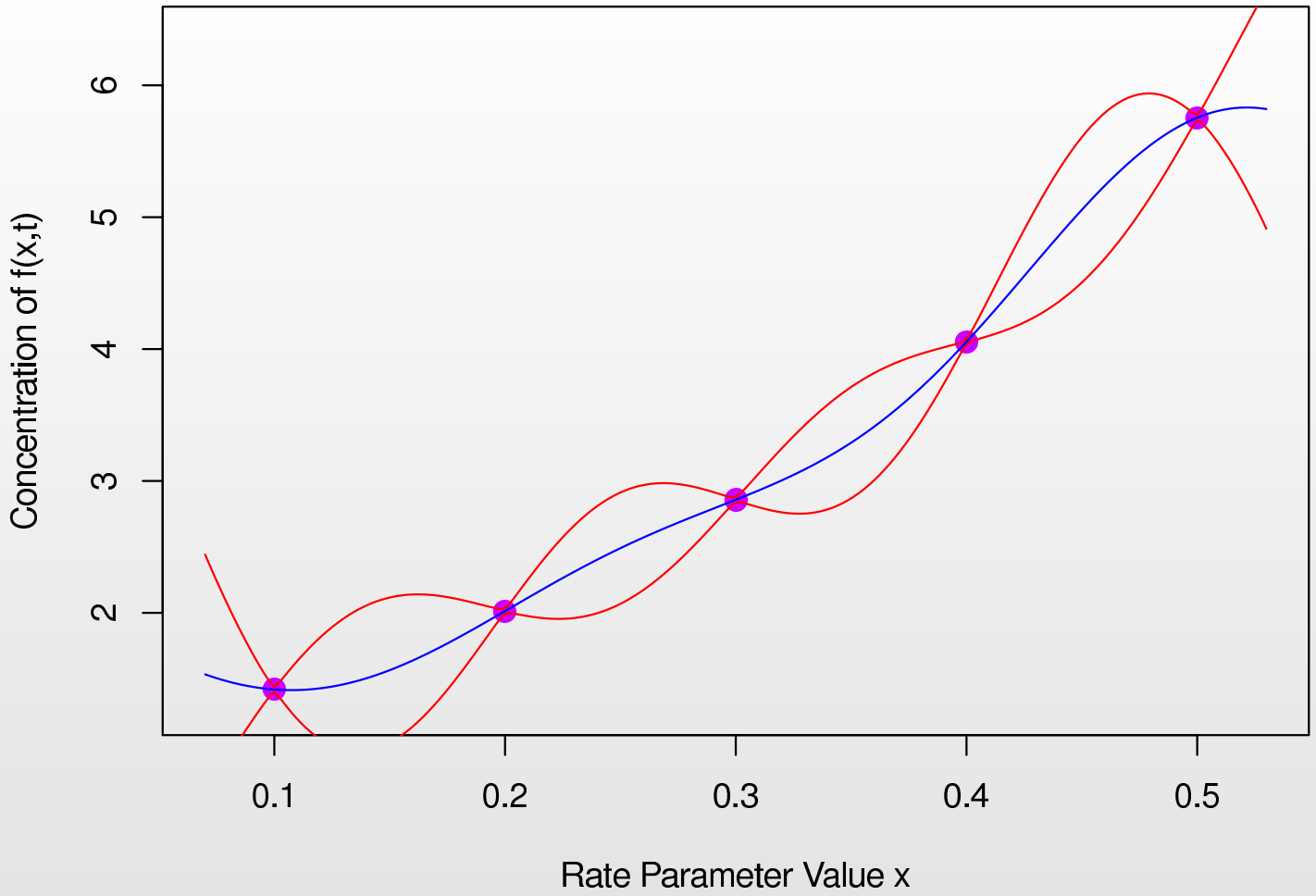
- Consider the graph of $f(x)$: in general we do not have the analytic solution of $f(x)$, here given by the dashed line.



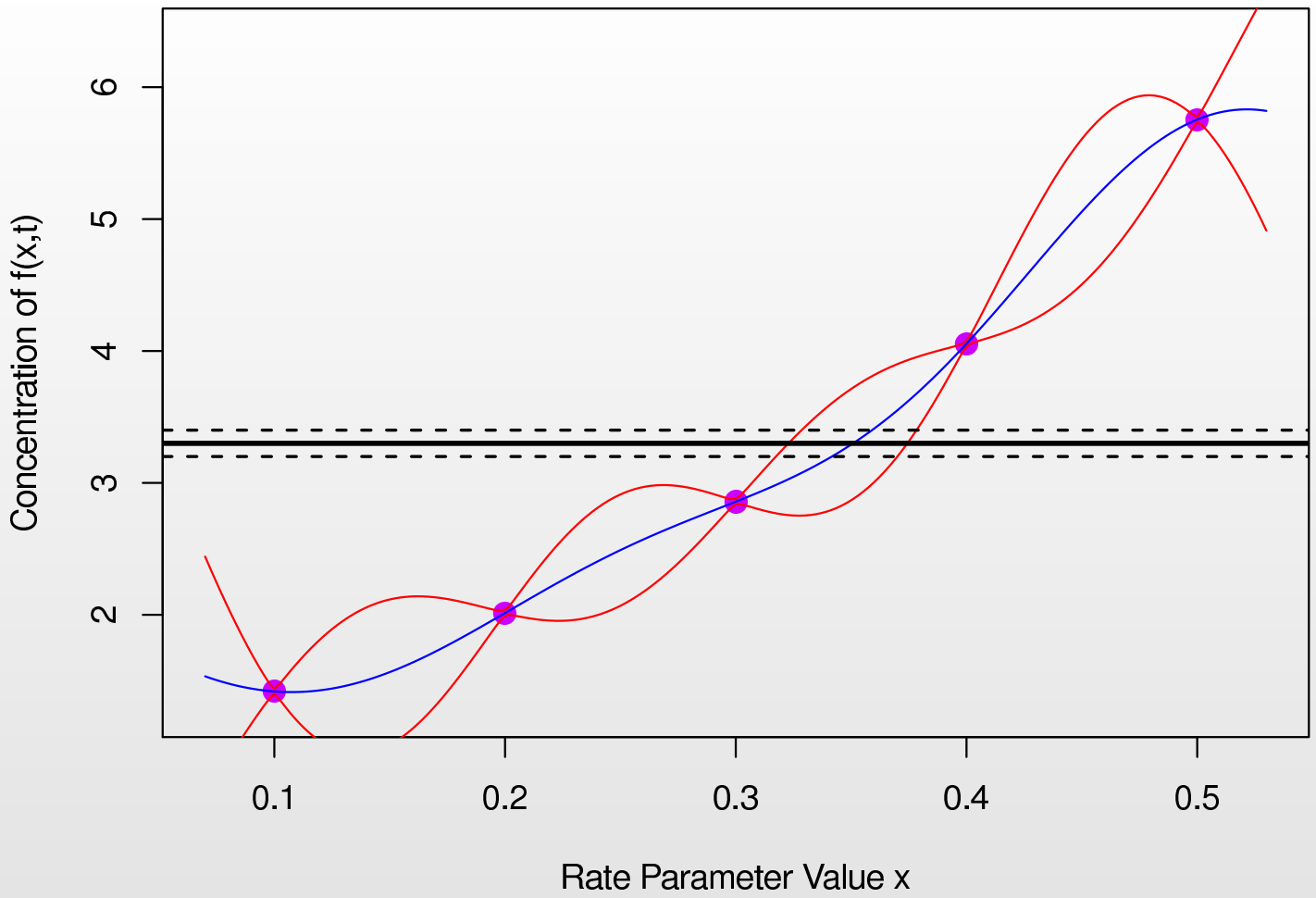
- Consider the graph of $f(x)$: in general we do not have the analytic solution of $f(x)$, here given by the dashed line.
- Instead we only have a finite number of runs of the model, in this case five.



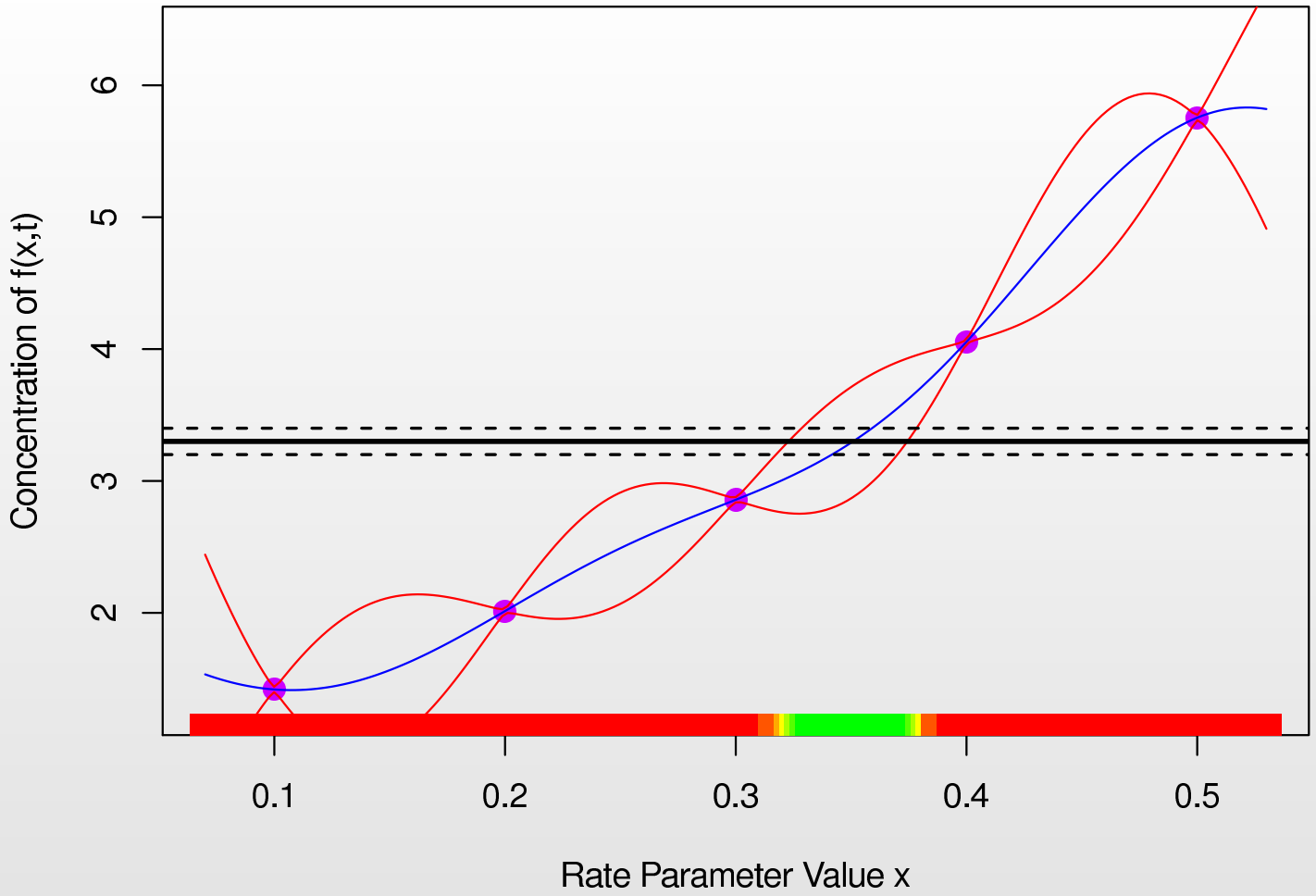
- The emulator can be used to represent our beliefs about the behaviour of the model at untested values of x, and is fast to evaluate.



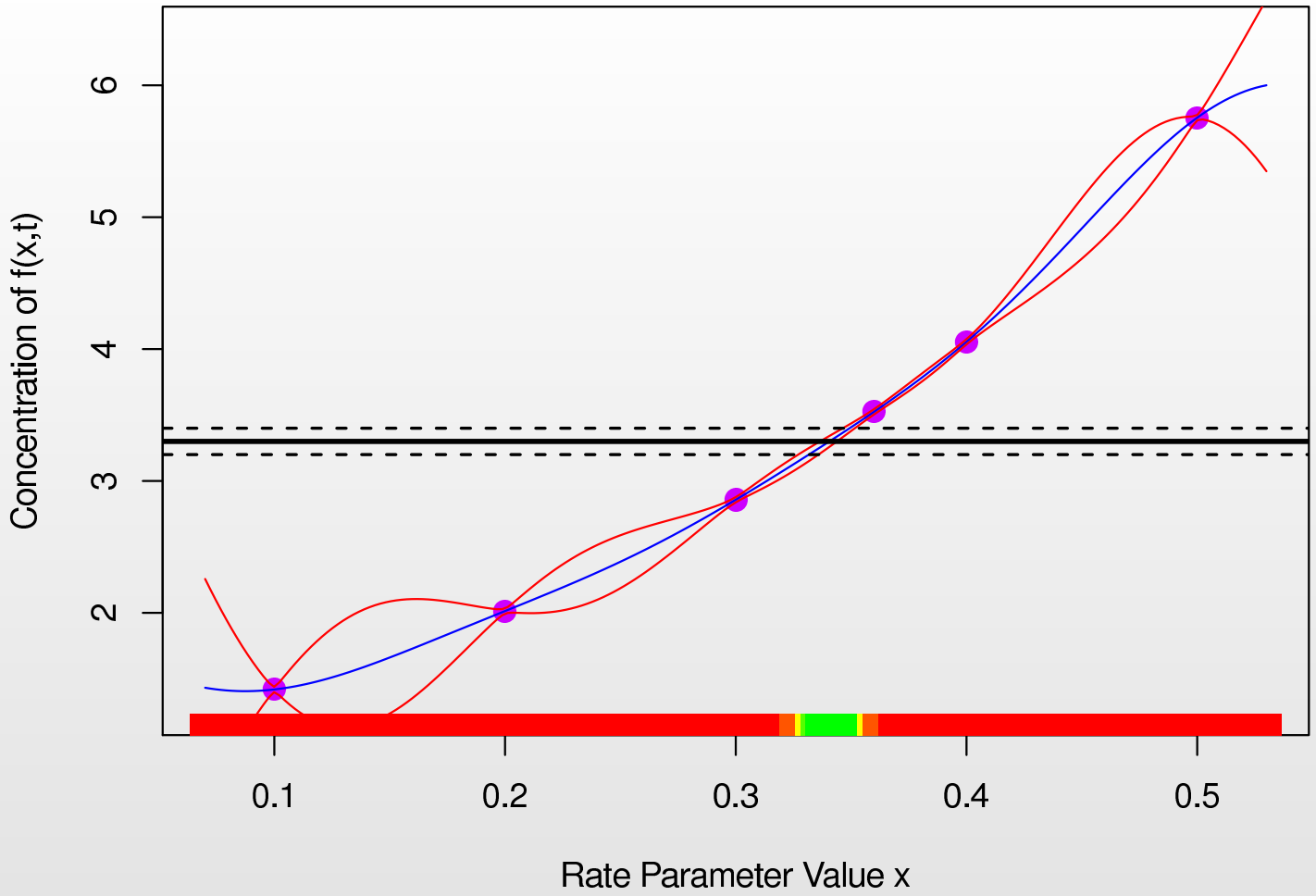
- The emulator can be used to represent our beliefs about the behaviour of the model at untested values of x, and is fast to evaluate.
- It gives both the expected value of $f(x)$ (the blue line) along with a credible interval for $f(x)$ (the red lines) representing the uncertainty about the model's behaviour.



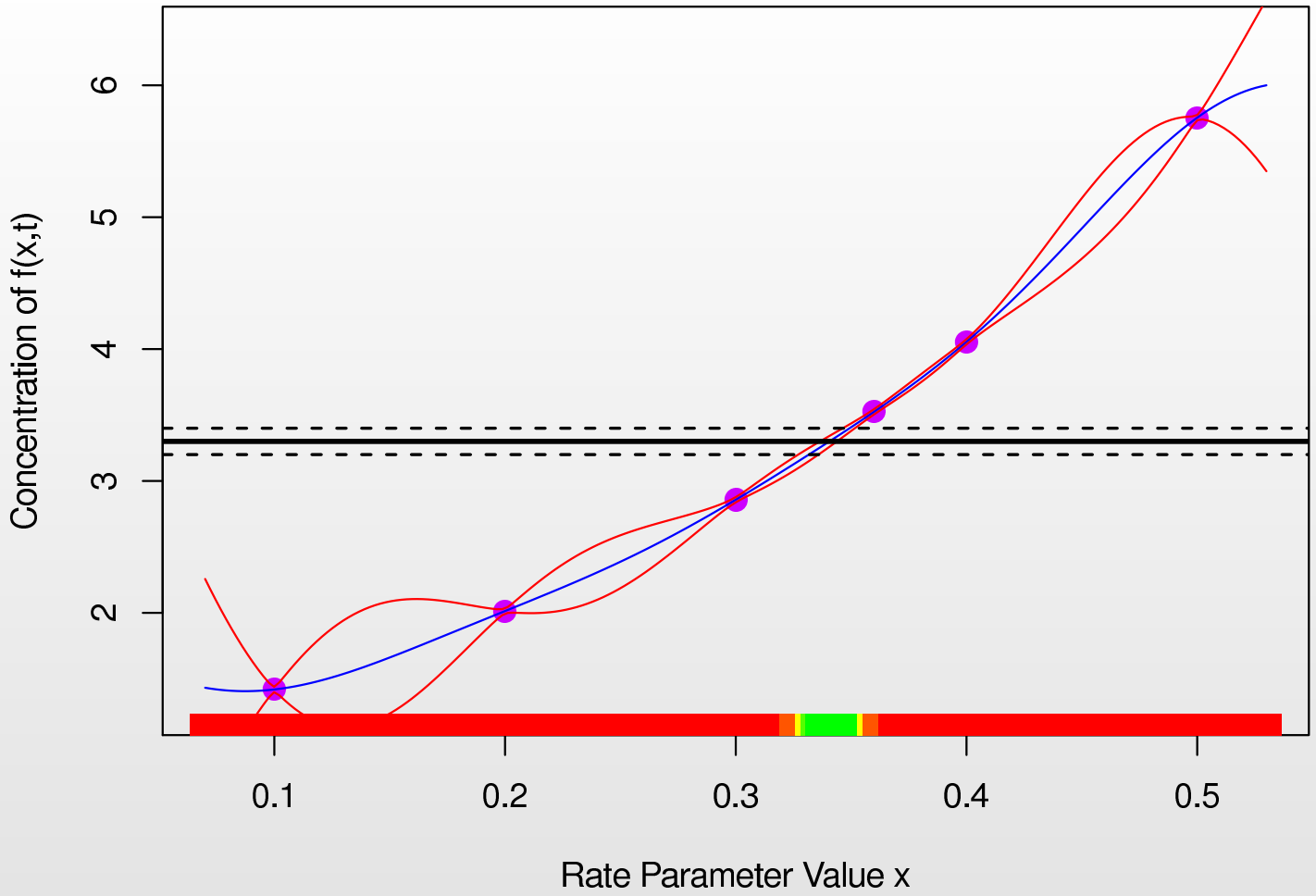
- Comparing the emulator to the observed measurement we again identify the set of x values currently consistent with this data (the observed errors here have been reduced for clarity).



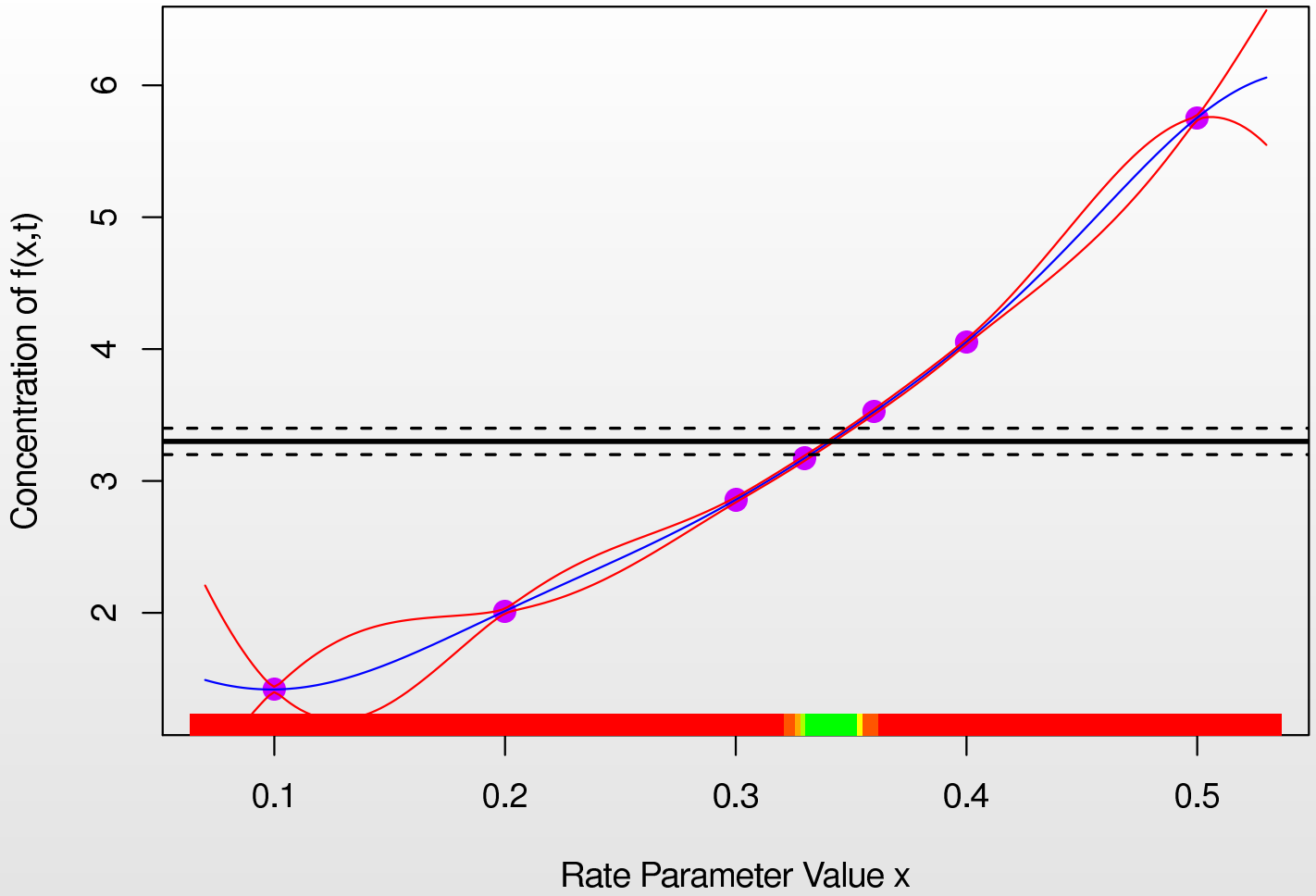
- Comparing the emulator to the observed measurement we again identify the set of x values currently consistent with this data (the observed errors here have been reduced for clarity).
- Note the uncertainty on x now includes uncertainty coming from the emulator.



- We perform a 2nd iteration or wave of runs to improve emulator accuracy.



- We perform a 2nd iteration or wave of runs to improve emulator accuracy.
- The runs are located only at non-implausible (green/yellow) points.



- We perform a 2nd iteration or wave of runs to improve emulator accuracy.
- The runs are located only at non-implausible (green/yellow) points.
- Now the emulator is more accurate than the observation, and we can identify the set of all x values of interest.

Designing a Future Experiment.

- We have seen how we can use an emulator to learn about the input parameter x even for a slow model.

Designing a Future Experiment.

- We have seen how we can use an emulator to learn about the input parameter x even for a slow model.

- We had measured the system at $t = 3.5$ and subsequently learnt about x. Now imagine we have to choose between two future experiments:

Designing a Future Experiment.

- We have seen how we can use an emulator to learn about the input parameter x even for a slow model.

- We had measured the system at $t = 3.5$ and subsequently learnt about x. Now imagine we have to choose between two future experiments:

Experiment A: Measure $f(x, t)$ at $t = 2$ with same observed error as before.

Experiment B: Measure $f(x, t)$ at $t = 5$ with same observed error as before.

Designing a Future Experiment.

- We have seen how we can use an emulator to learn about the input parameter x even for a slow model.

- We had measured the system at $t = 3.5$ and subsequently learnt about x. Now imagine we have to choose between two future experiments:

  Experiment A: Measure $f(x, t)$ at $t = 2$ with same observed error as before.

  Experiment B: Measure $f(x, t)$ at $t = 5$ with same observed error as before.

- We only have the money/resources to do one of these experiments, so which is best?

Designing a Future Experiment.

- We have seen how we can use an emulator to learn about the input parameter x even for a slow model.

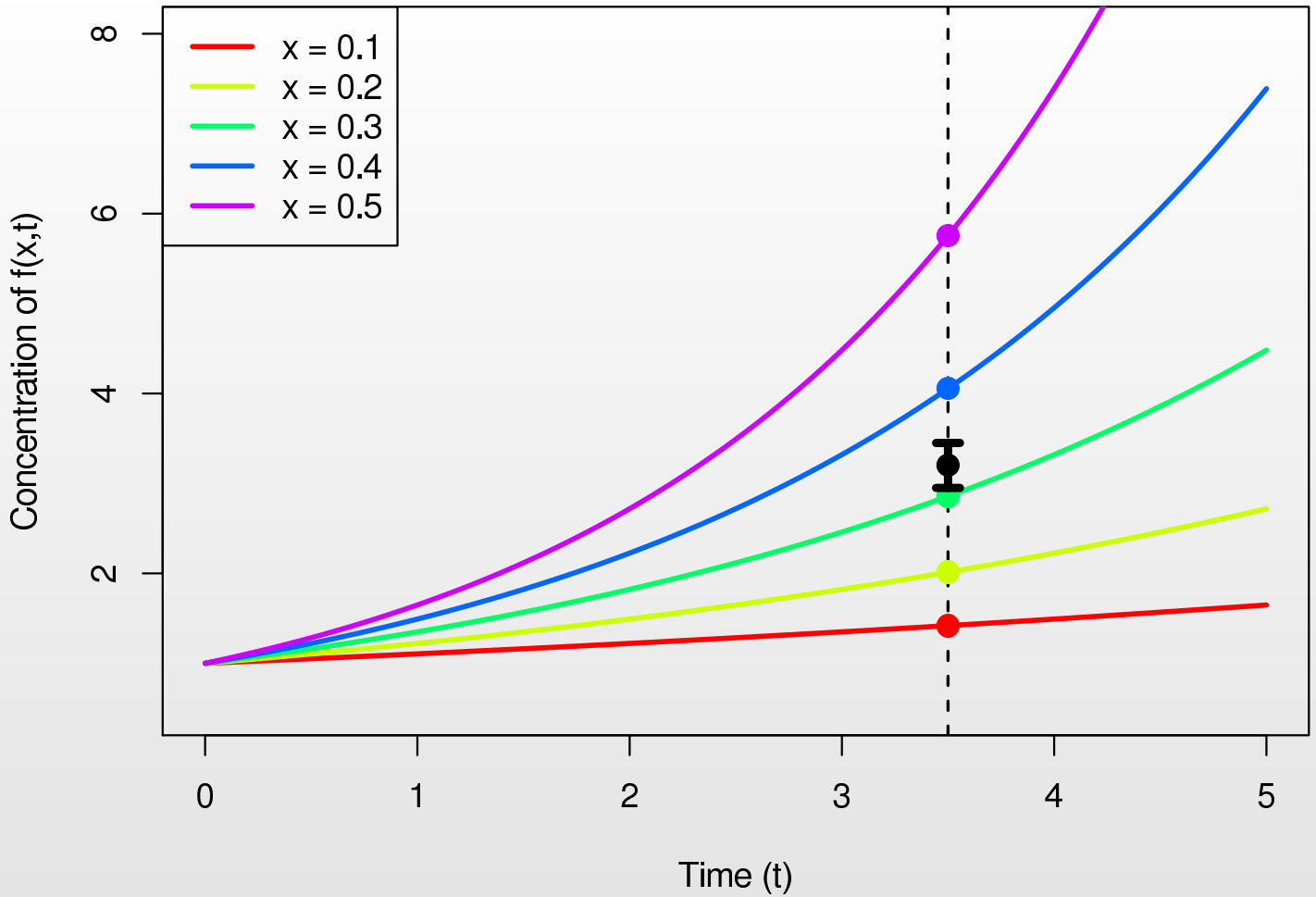
- We had measured the system at $t = 3.5$ and subsequently learnt about x. Now imagine we have to choose between two future experiments:

  Experiment A: Measure $f(x, t)$ at $t = 2$ with same observed error as before.

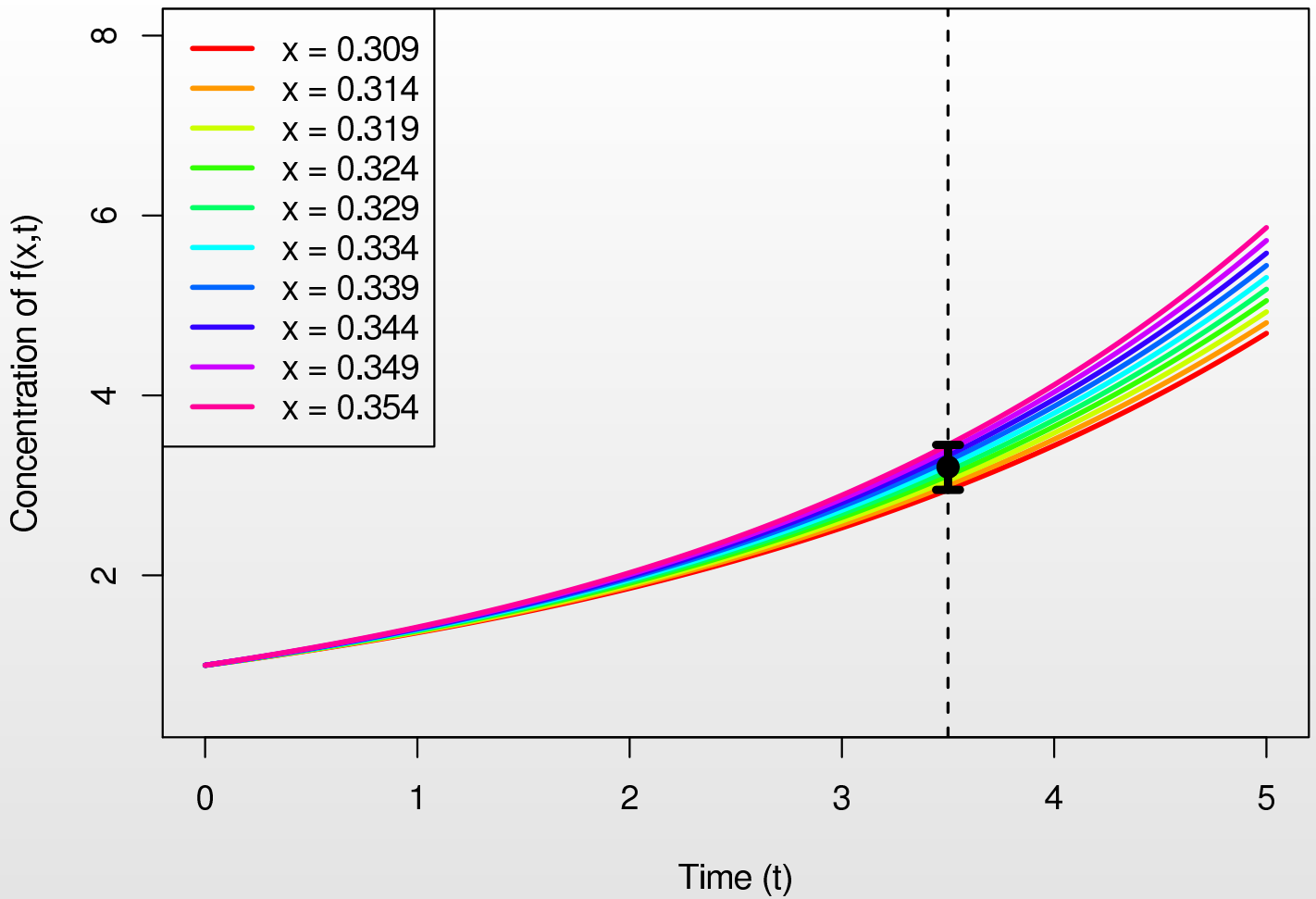
  Experiment B: Measure $f(x, t)$ at $t = 5$ with same observed error as before.

- We only have the money/resources to do one of these experiments, so which is best?

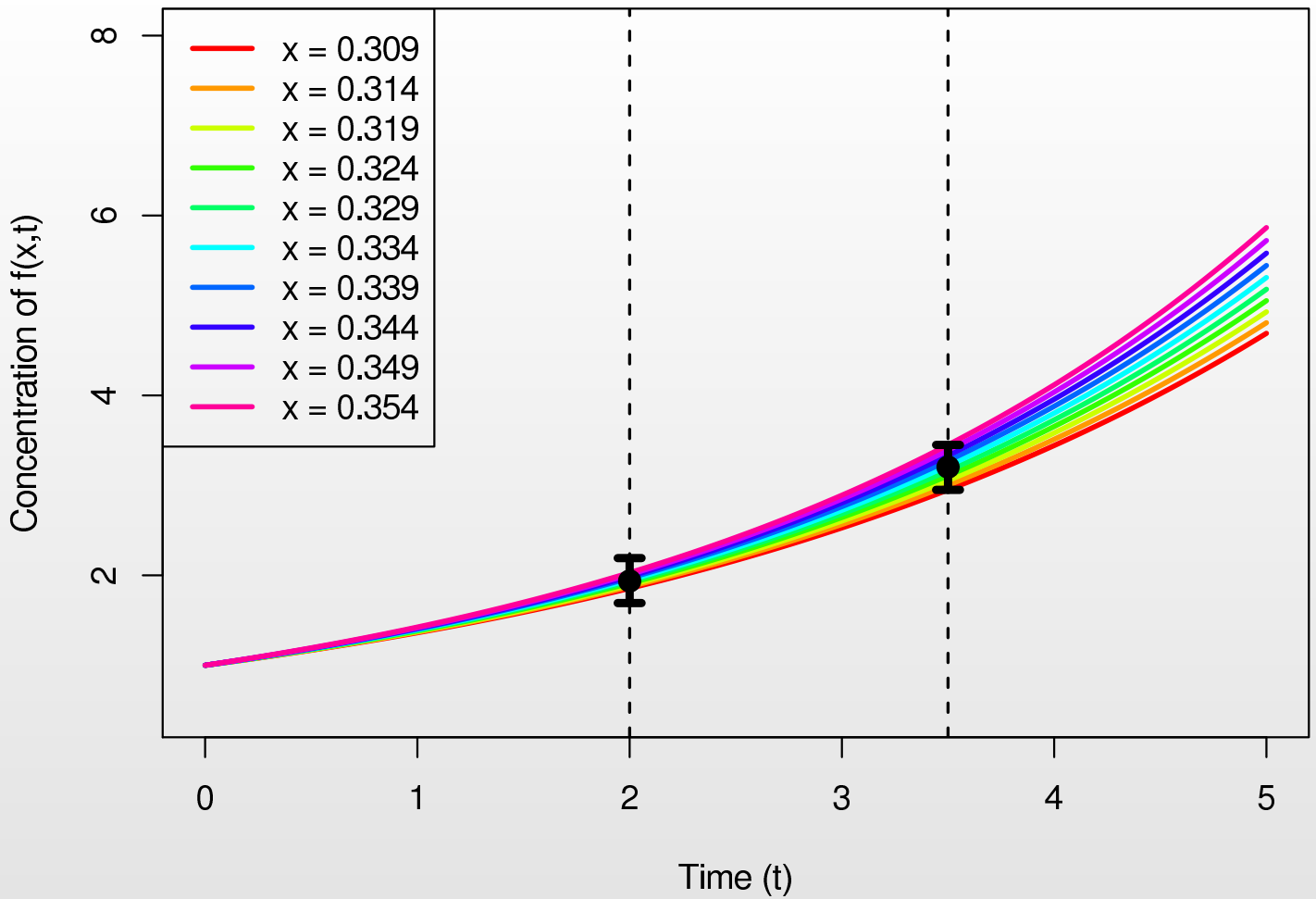
- We can use the model's predictions at $t = 2$ and $t = 5$ to determine which experiment A or B is expected to be most informative about the input parameter x, given our knowledge about $f(x, t)$ at $t = 3.5$.



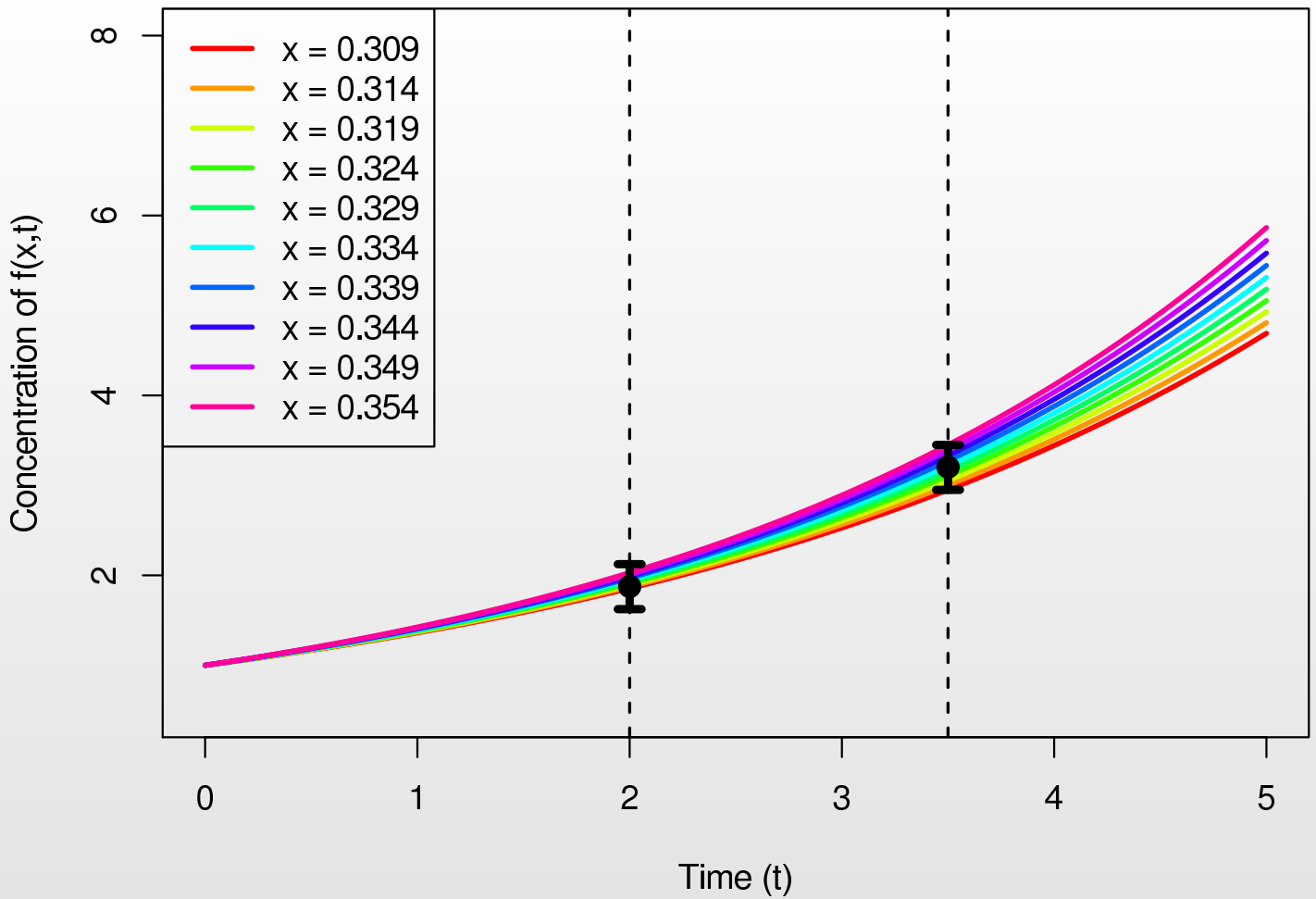
- Using the emulator we can choose several values of x consistent with the measurement of $f(x,t)$ at $t = 3.5$, and perform corresponding runs of the model.



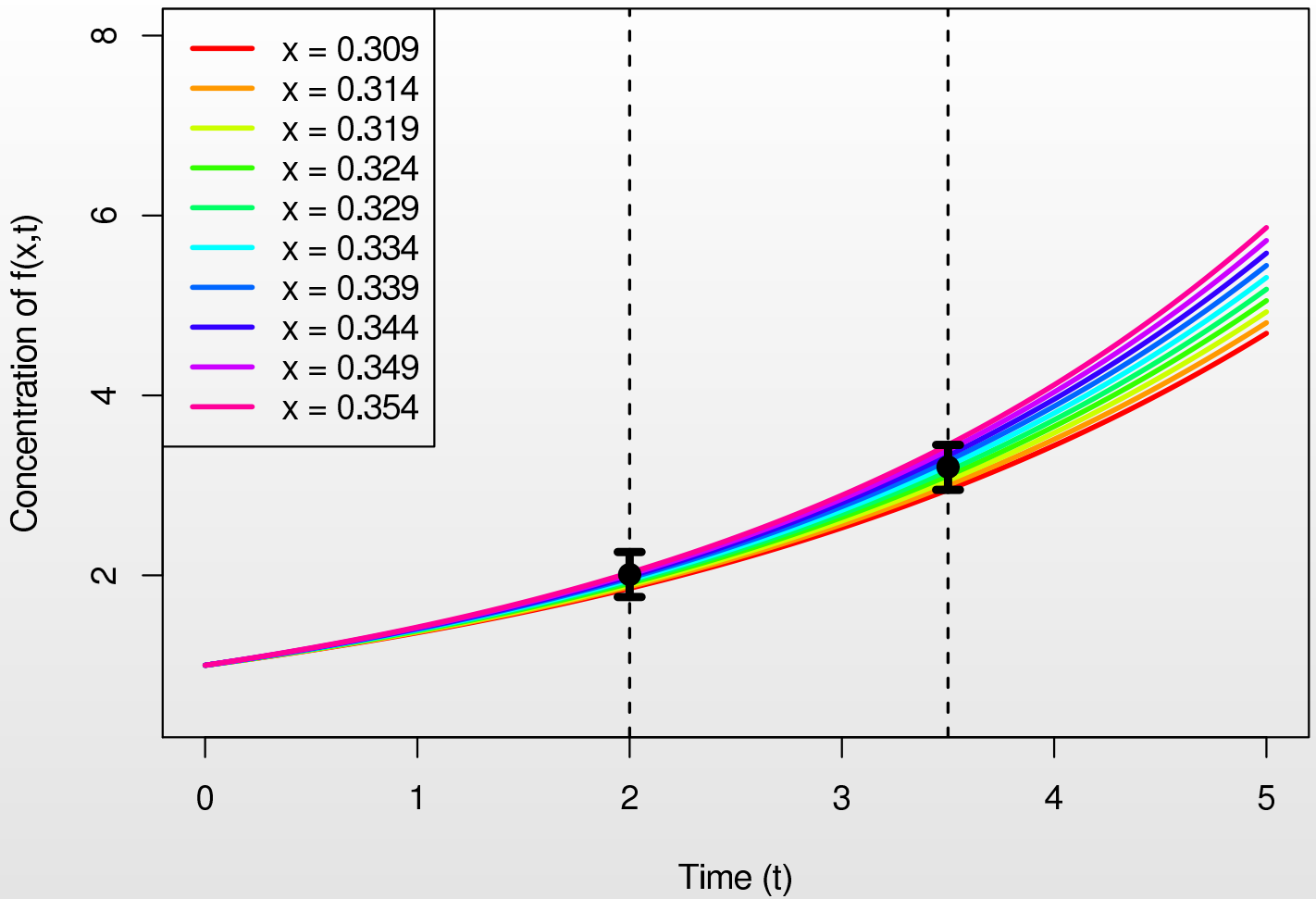
- Using the emulator we can choose several values of x consistent with the measurement of $f(x,t)$ at $t = 3.5$, and perform corresponding runs of the model.



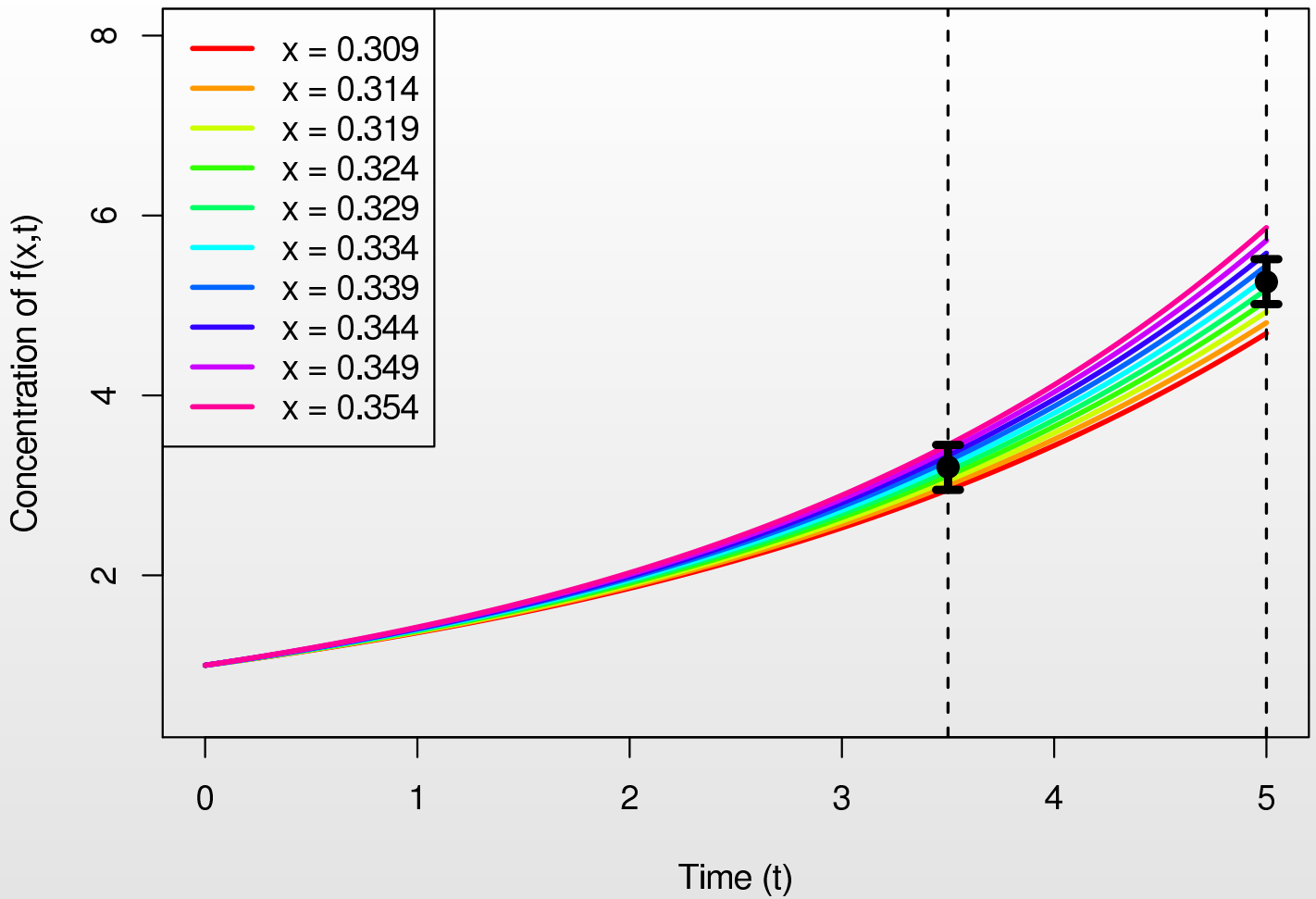
- Using the emulator we can choose several values of x consistent with the measurement of $f(x,t)$ at $t = 3.5$, and perform corresponding runs of the model.
- We can check the predictions made by these runs for $f(x, t = 2)$.



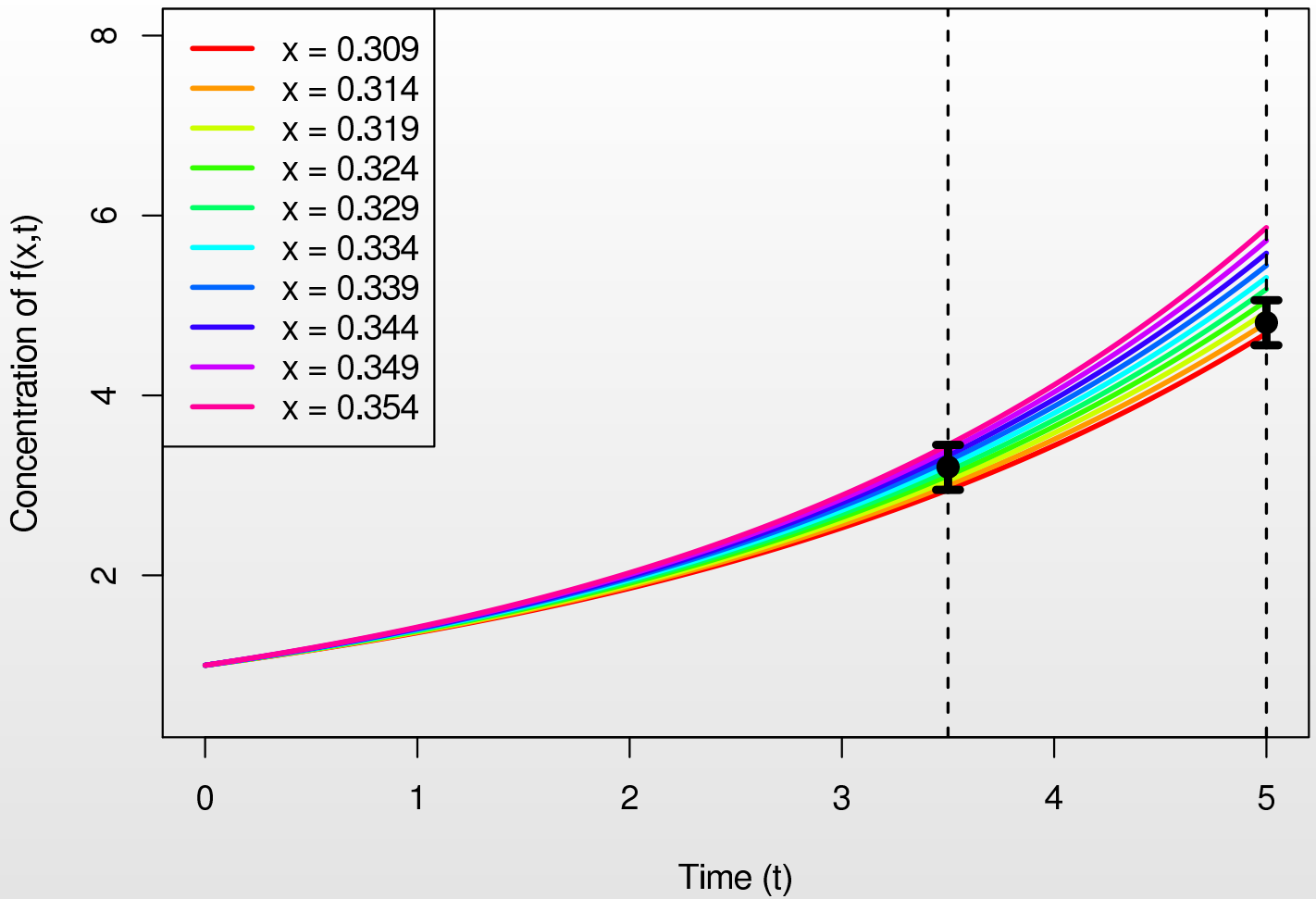
- Using the emulator we can choose several values of x consistent with the measurement of $f(x,t)$ at $t = 3.5$, and perform corresponding runs of the model.
- We can check the predictions made by these runs for $f(x, t = 2)$.



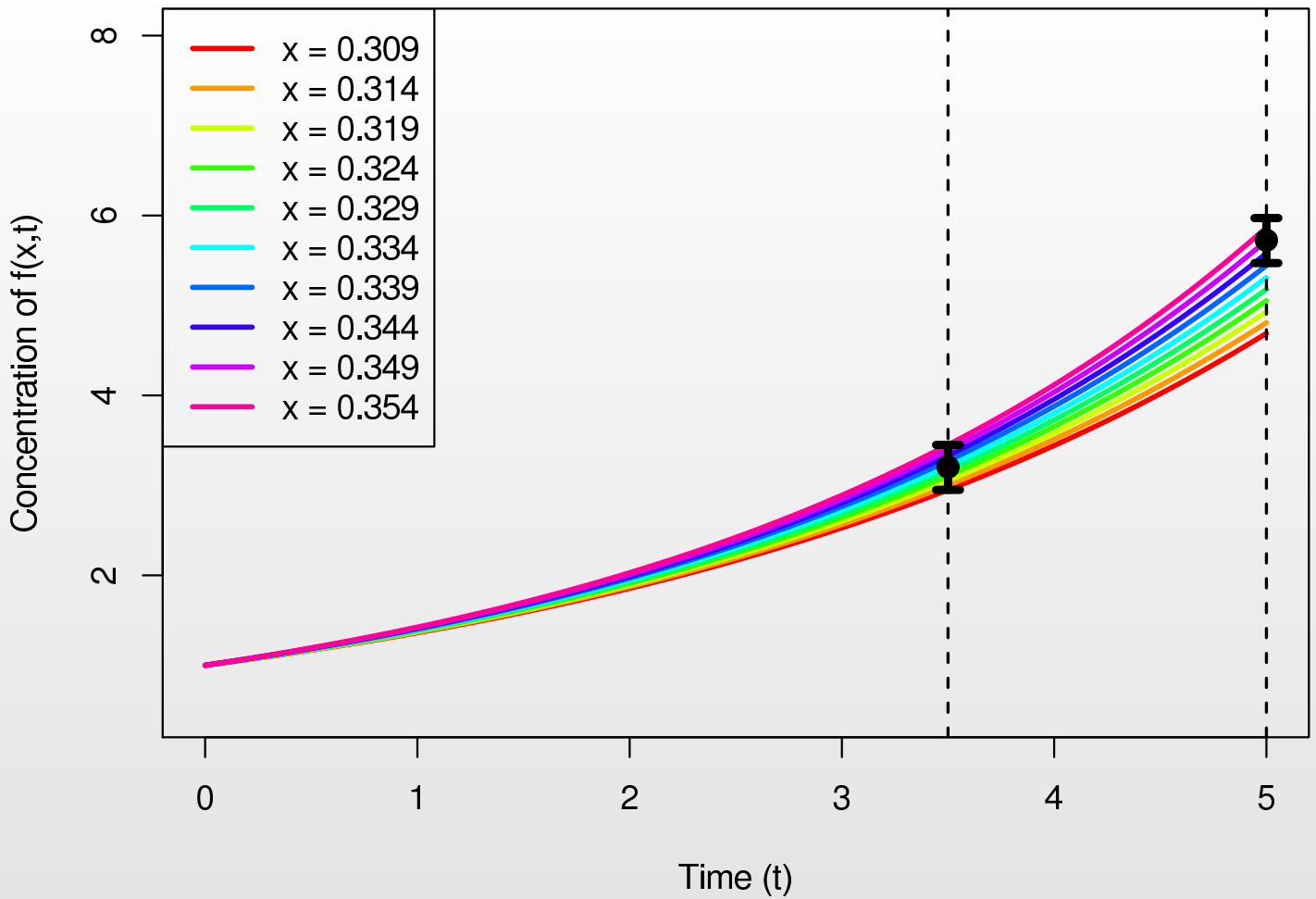
- The predictions imply that any measurement of $f(x, t = 2)$ is highly unlikely to be informative for x.
- This is due to the measurement errors swamping the signal from the model output $f(x, t = 2)$.



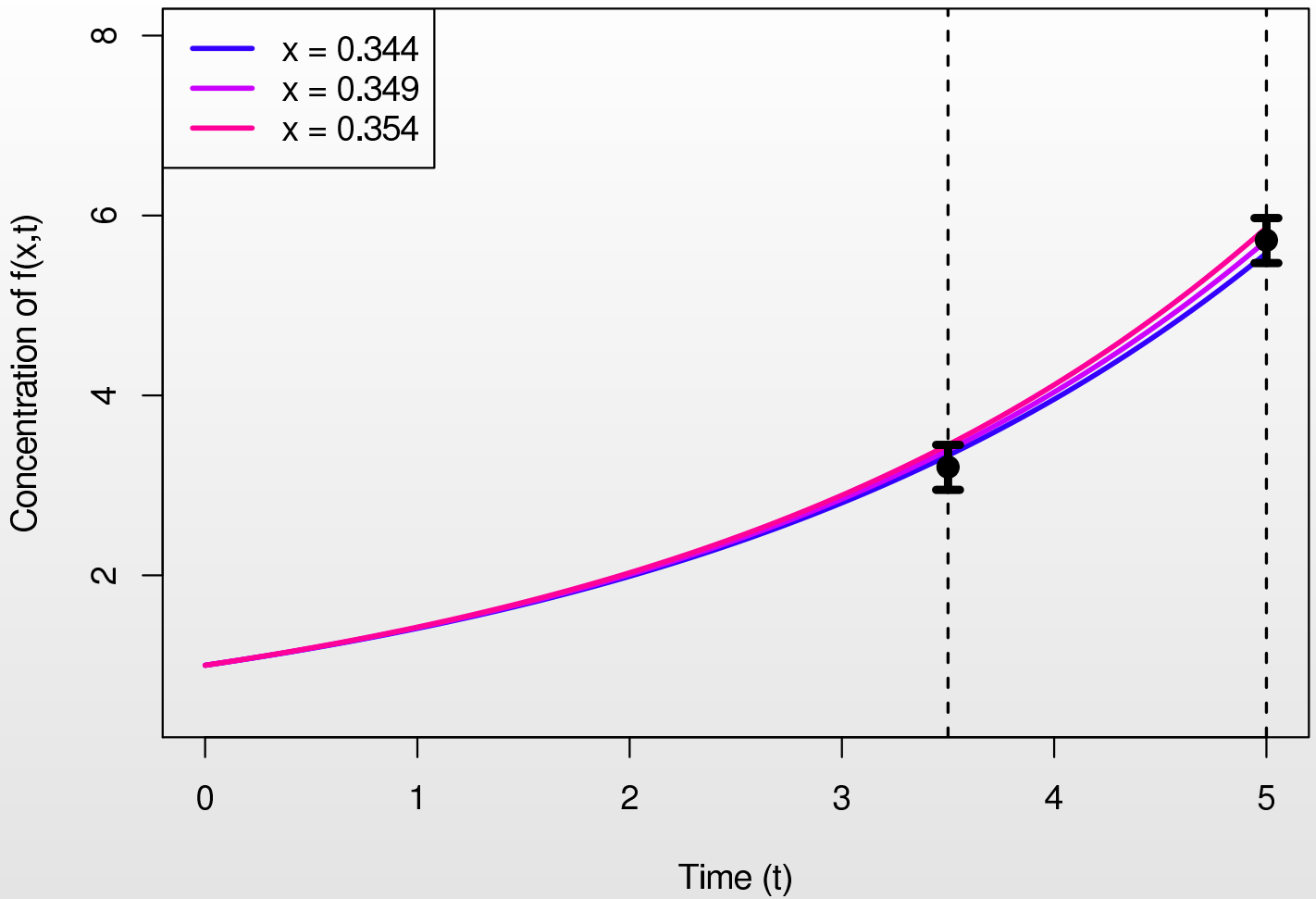
- The predictions for $f(x, t = 5)$ show a different conclusion.



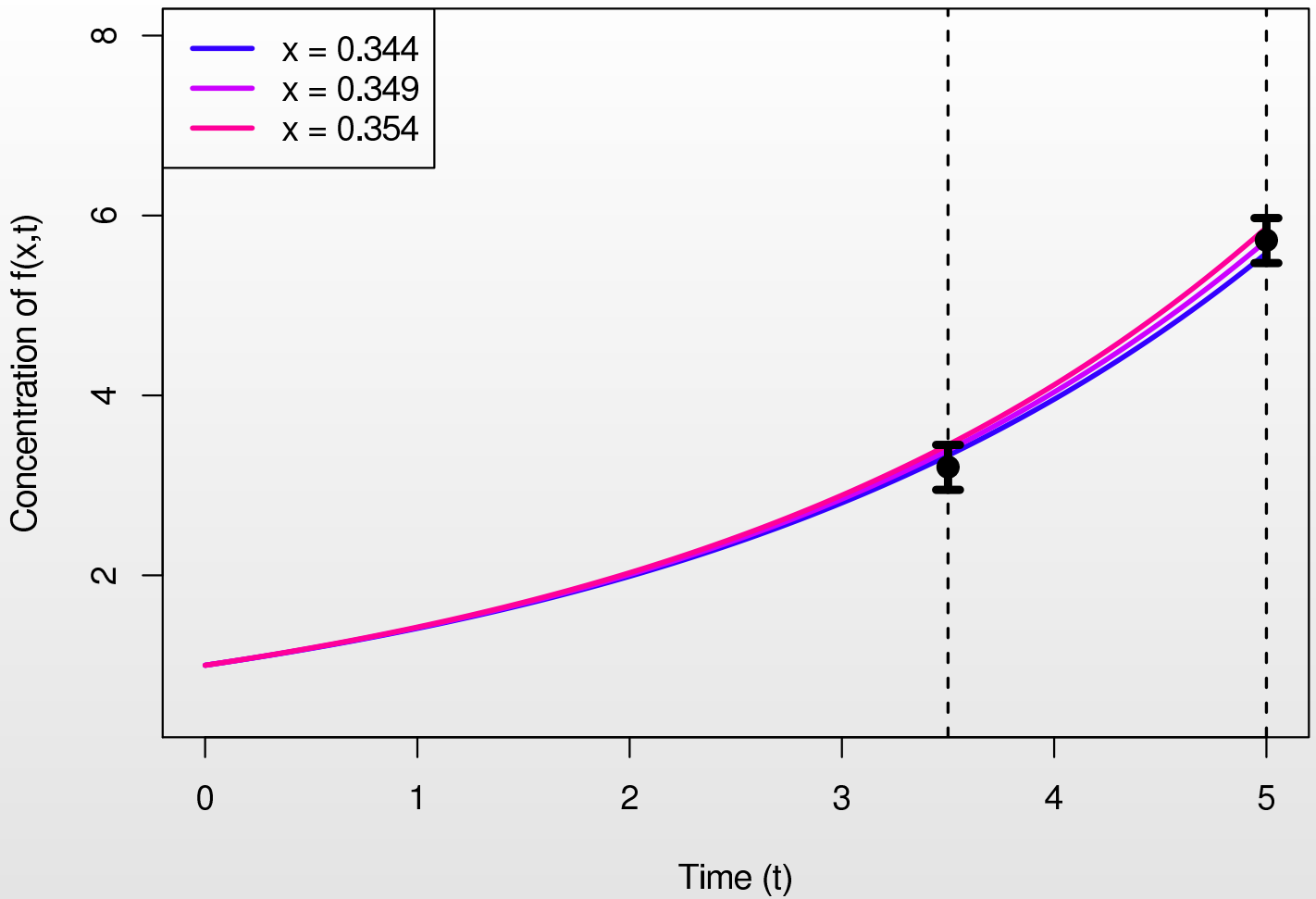
- The predictions for $f(x, t = 5)$ show a different conclusion.



- The predictions for $f(x, t = 5)$ show a different conclusion.
- For each possible measurement of $f(x, t = 5)$ it is highly likely that we will be able to rule out several more values of x as implausible.



- For one possible measurement, we see that non-implausible values of x would lie approximately between 0.344 and 0.354, ruling out approximately 70% of the possible values of x.



- For one possible measurement, we see that non-implausible values of x would lie approximately between 0.344 and 0.354, ruling out approximately 70% of the possible values of x.
- This high expected space reduction in x implies that Experiment B, measuring $f(x,t)$ at $t = 5$, is clearly the best choice.

Simple Example: Conclusions

- We have hence used the results of the first experiment, measurement of $f(x, t = 3.5)$ to make predictions of the two possible experiments A and B.

Simple Example: Conclusions

- We have hence used the results of the first experiment, measurement of $f(x, t = 3.5)$ to make predictions of the two possible experiments A and B.

- We then choose Experiment B, measurement of $f(x, t = 5)$, as it is the most efficient experiment for learning about x, as it has the highest expected space reduction (reduction in the range) of x, for the given measurement errors.

Simple Example: Conclusions

- We have hence used the results of the first experiment, measurement of $f(x, t = 3.5)$ to make predictions of the two possible experiments A and B.

- We then choose Experiment B, measurement of $f(x, t = 5)$, as it is the most efficient experiment for learning about x, as it has the highest expected space reduction (reduction in the range) of x, for the given measurement errors.

- Note that although this method requires trusting the model predictions, in general we would incorporate a model discrepancy term to allow for a certain level of inaccuracy of the model.

Systems Biology: Arabidopsis

- Small flowering plant related to cabbage and mustard.

Systems Biology: Arabidopsis

- Small flowering plant related to cabbage and mustard.
- One of the model organisms used for studying plant biology and the first plant to have its entire genome sequenced.

Systems Biology: Arabidopsis

- Small flowering plant related to cabbage and mustard.
- One of the model organisms used for studying plant biology and the first plant to have its entire genome sequenced.
- Changes in it are easily observed, making it very useful.

Hormonal Crosstalk in Arabidopsis

- Liu et. al. developed a kinetic model of hormonal crosstalk in Arabidopsis,

Hormonal Crosstalk in Arabidopsis

- Liu et. al. developed a kinetic model of hormonal crosstalk in Arabidopsis,
- Model describes the function of POLARIS (PLS) peptide in auxin biosysthesis.

Hormonal Crosstalk in Arabidopsis

- Liu et. al. developed a kinetic model of hormonal crosstalk in Arabidopsis,
- Model describes the function of POLARIS (PLS) peptide in auxin biosysthesis.
- Also describes the complex interactions of three measurable hormones: auxin, ethylene and cytokinin in various cases.

Hormonal Crosstalk in Arabidopsis

- Liu et. al. developed a kinetic model of hormonal crosstalk in Arabidopsis,
- Model describes the function of POLARIS (PLS) peptide in auxin biosysthesis.
- Also describes the complex interactions of three measurable hormones: auxin, ethylene and cytokinin in various cases.
- Model effectively has 96 outputs, 16 of which have been measured (z).

Hormonal Crosstalk in Arabidopsis

- Liu et. al. developed a kinetic model of hormonal crosstalk in Arabidopsis,
- Model describes the function of POLARIS (PLS) peptide in auxin biosysthesis.
- Also describes the complex interactions of three measurable hormones: auxin, ethylene and cytokinin in various cases.
- Model effectively has 96 outputs, 16 of which have been measured (z).
- We have money to measure 4 more outputs in the future.

Hormonal Crosstalk in Arabidopsis

- Liu et. al. developed a kinetic model of hormonal crosstalk in Arabidopsis,
- Model describes the function of POLARIS (PLS) peptide in auxin biosysthesis.
- Also describes the complex interactions of three measurable hormones: auxin, ethylene and cytokinin in various cases.
- Model effectively has 96 outputs, 16 of which have been measured (z).
- We have money to measure 4 more outputs in the future.
- Model has 32 unknown input rate parameters which have ranges of 5 orders of magnitude,

Hormonal Crosstalk in Arabidopsis

- Liu et. al. developed a kinetic model of hormonal crosstalk in Arabidopsis,
- Model describes the function of POLARIS (PLS) peptide in auxin biosysthesis.
- Also describes the complex interactions of three measurable hormones: auxin, ethylene and cytokinin in various cases.
- Model effectively has 96 outputs, 16 of which have been measured (z).
- We have money to measure 4 more outputs in the future.
- Model has 32 unknown input rate parameters which have ranges of 5 orders of magnitude,
- Here the input x is a vector of length 32, but the output $f(x)$ is of length 96.

Arabidopsis Design of Experiment

- First major question: Is the model $f(x)$ currently consistent with the observed measurements z? If so can we identify the set $\mathcal{X}(z)$ of all acceptable inputs?

Arabidopsis Design of Experiment

- First major question: Is the model $f(x)$ currently consistent with the observed measurements z? If so can we identify the set $\mathcal{X}(z)$ of all acceptable inputs?
- To answer this we perform a History Match.

Arabidopsis Design of Experiment

- First major question: Is the model $f(x)$ currently consistent with the observed measurements z? If so can we identify the set $\mathcal{X}(z)$ of all acceptable inputs?
- To answer this we perform a History Match.
- Second major question: What is the most informative future experiment we can perform to learn about the Arabidopsis system?

Arabidopsis Design of Experiment

- First major question: Is the model $f(x)$ currently consistent with the observed measurements z? If so can we identify the set $\mathcal{X}(z)$ of all acceptable inputs?
- To answer this we perform a History Match.
- Second major question: What is the most informative future experiment we can perform to learn about the Arabidopsis system?
- To answer this we need to:
  - Specify the class of possible experiments considered.

Arabidopsis Design of Experiment

- First major question: Is the model $f(x)$ currently consistent with the observed measurements z? If so can we identify the set $\mathcal{X}(z)$ of all acceptable inputs?
- To answer this we perform a History Match.
- Second major question: What is the most informative future experiment we can perform to learn about the Arabidopsis system?
- To answer this we need to:
  - Specify the class of possible experiments considered.
  - Use the results of the History Match to obtain model predictions for all future experiments that are consistent with current observations.

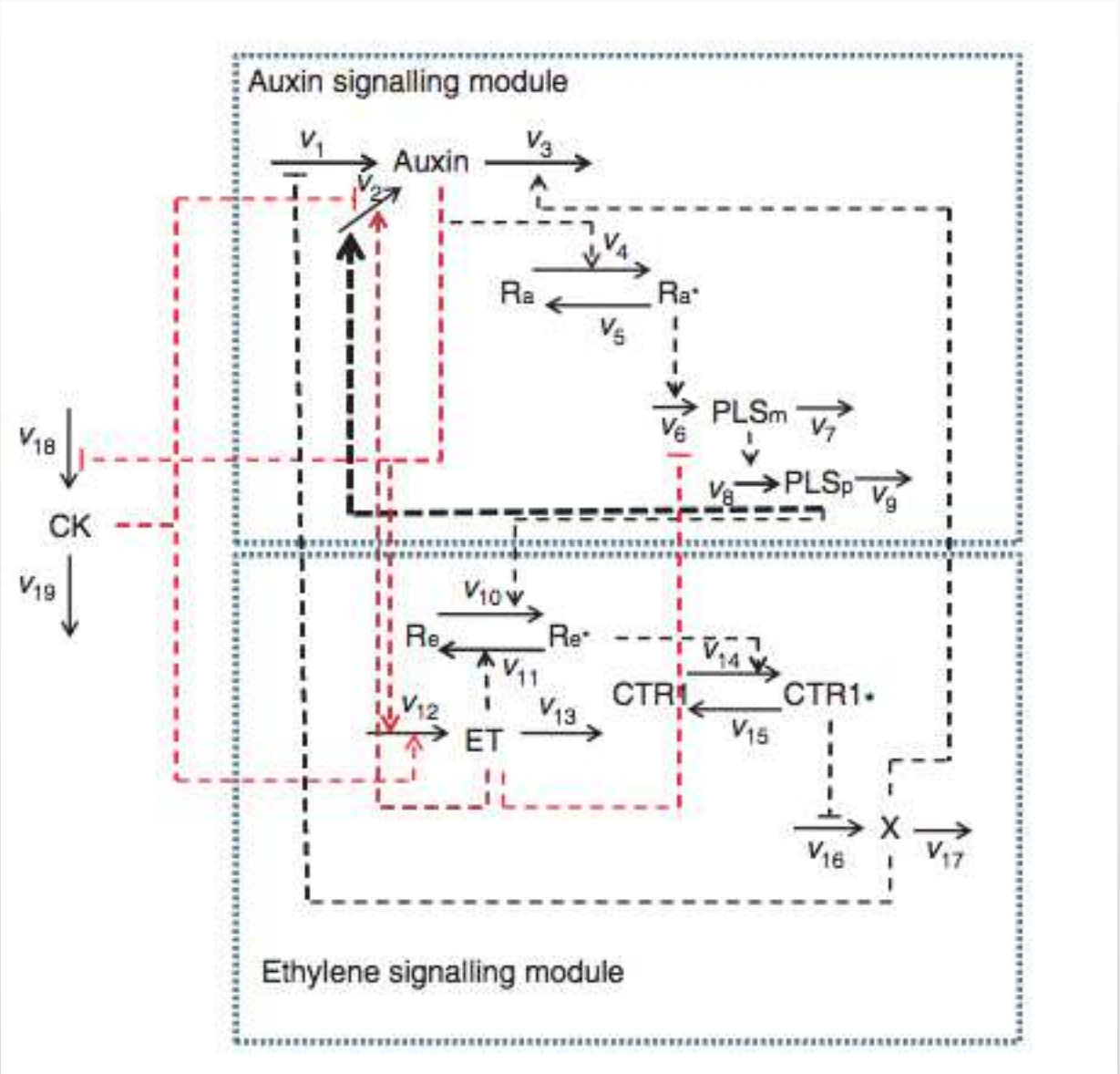
Arabidopsis Design of Experiment

- First major question: Is the model $f(x)$ currently consistent with the observed measurements z? If so can we identify the set $\mathcal{X}(z)$ of all acceptable inputs?
- To answer this we perform a History Match.
- Second major question: What is the most informative future experiment we can perform to learn about the Arabidopsis system?
- To answer this we need to:
  - Specify the class of possible experiments considered.
  - Use the results of the History Match to obtain model predictions for all future experiments that are consistent with current observations.
  - Choose the most efficient experiment based on an Expected Space Reduction criteria and complementary robustness considerations.
- This will result in a design for a new experiment that is expected to be highly informative about the input or rate parameters of the Arabidopsis system.

Slides describing model inputs and outputs.

Chemical Output	Initial concentration	Measurable
Auxin	0.1	Yes
X	0.1	
PLSp	0.1	Yes
Ra	0	
Ra_star	1	
CK	0.1	Yes
ET	0.1	Yes
PLSm	0.1	
Re	0	
Re_star	0.3	
CTR1	0	
CTR1_star	0.3	
IAA	0	
cytokinin	0	
ACC	0	

Reaction Network Model



32 Reaction Rates: 32 Input parameters

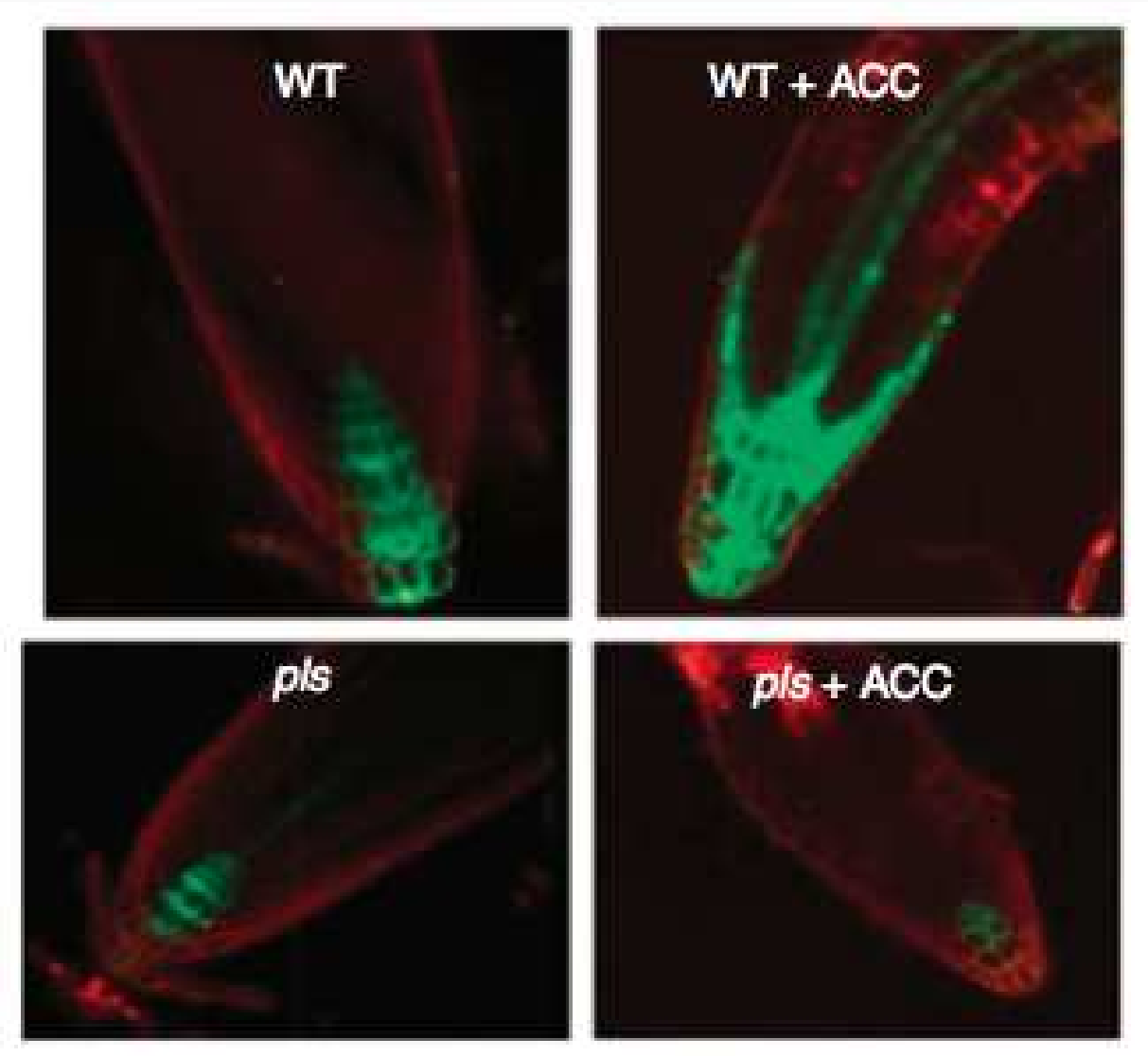
Input	min	max	Input	min	max
k1	0.001	100	k1a	0.001	100
k2	0.0002	20	k2a	0.0028	280
k2b	0.001	100	k2c	1×10^{-5}	1
k3	0.002	200	k3a	0.00045	45
k4	0.001	100	k5	0.001	100
k6	0.3	0.3	k6a	0.0002	20
k7	0.001	100	k8	0.001	100
k9	0.001	100	k10	3×10^{-7}	0.03
k10a	0.0005	50	k11	0.005	500
k12	0.0001	10	k12a	0.0001	10
k13	0.001	100	k14	0.003	300
k15	8.5×10^{-5}	8.5	k16	0.0003	30
k16a	0.001	100	k17	0.0001	10
k18	0.0001	10	k18a	0.001	100
k19	0.001	100	k1vauxin	0.001	100
k1vCK	0.001	100	k1veth	0.001	100

- So now the input $x =$ (k1, k1a, k2, k2a, . . . ,k19, k1vauxin, k1vCK, k1veth)

Arabidopsis Model

$$
\begin{aligned}
\frac{dAuxin}{dt} &= \frac{k_{1a}}{1+\frac{X}{k_1}} + k_2 + \frac{k_{2a}ET}{1+\frac{CK}{k_{2b}}}\frac{PLSp}{k_{2c}+PLSp} - \\
&\quad (k_3 + k_{3a}X)Auxin + k_{1vauxin}IAA \\
\frac{dX}{dt} &= k_{16} - k_{16a}CTR1_{star} - k_{17}X \\
\frac{dPLSp}{dt} &= k_8 PLSm - k_9 PLSp \\
\frac{dRa}{dt} &= -k_4 AuxinRa + k_5 Ra_{star} \\
\frac{dRa_{star}}{dt} &= k_4 AuxinRa - k_5 Ra_{star} \\
\frac{dCK}{dt} &= \frac{k_{18a}}{1+\frac{Auxin}{k_{18}}} - k_{19}CK + k_{1vCK}cytokinin \\
\frac{dET}{dt} &= k_{12} + k_{12a}AuxinCK - k_{13}ET + k_{1veth}ACC \\
\frac{dPLSm}{dt} &= \frac{k_6 Ra_{star}}{1+\frac{ET}{k_{6a}}} - k_7 PLSm \\
\frac{dRe}{dt} &= k_{11}Re_{star}ET - (k_{10} + k_{10a}PLSp)Re \\
\frac{dRe_{star}}{dt} &= -k_{11}Re_{star}ET + (k_{10} + k_{10a}PLSp)Re \\
\frac{dCTR1}{dt} &= -k_{14}Re_{star}CTR1 + k_{15}CTR1_{star} \\
\frac{dCTR1_{star}}{dt} &= k_{14}Re_{star}CTR1 - k_{15}CTR1_{star} \\
\frac{dIAA}{dt} &= 0, \quad \frac{dcytokinin}{dt} = 0, \quad \frac{dACC}{dt} = 0
\end{aligned}
$$

Measurements of root hormone level.



Fundamental Scientific Questions

Fundamental scientific questions:

Fundamental Scientific Questions

Fundamental scientific questions:

1 (a) Are there any choices of rate parameters consistent with the 16 observed trends z?

Fundamental Scientific Questions

Fundamental scientific questions:

1 (a) Are there any choices of rate parameters consistent with the 16 observed trends z?

(b) Can we identify the set $\mathcal{X}(z)$ of all such input or rate parameters?

Fundamental Scientific Questions

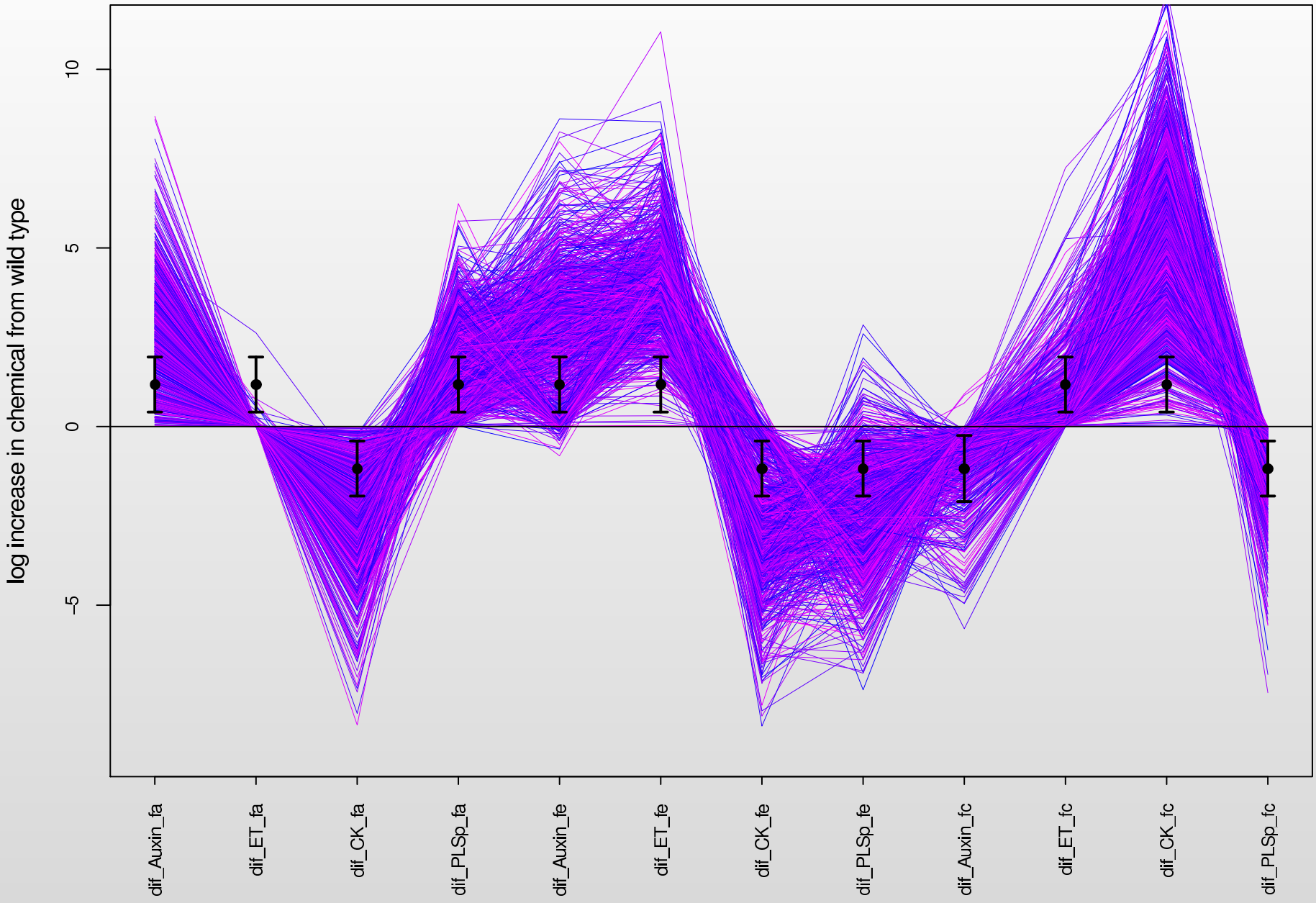
Fundamental scientific questions:

1 (a) Are there any choices of rate parameters consistent with the 16 observed trends z?

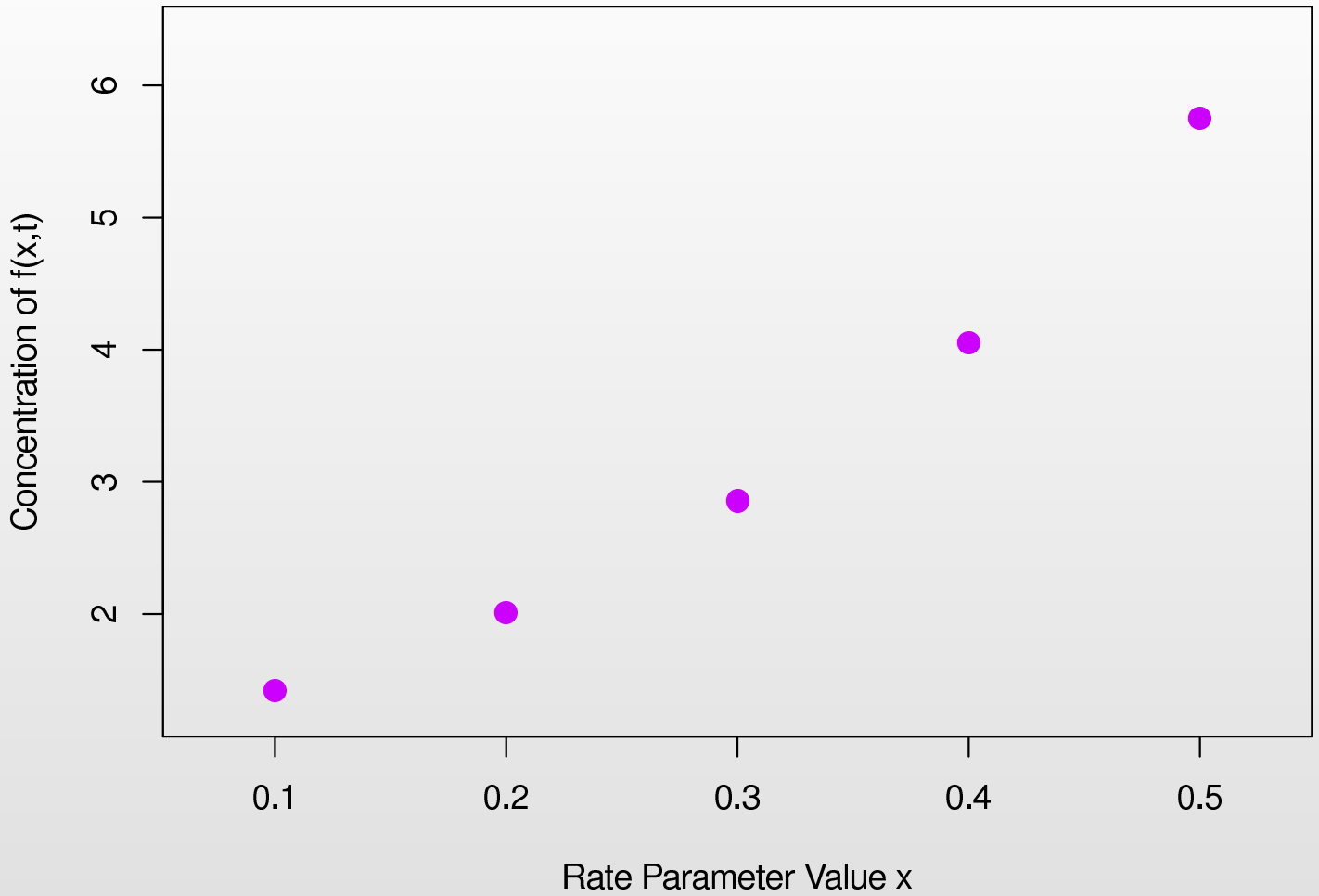
(b) Can we identify the set $\mathcal{X}(z)$ of all such input or rate parameters?

3 What design of future experiment will reduce this set $\mathcal{X}(z)$, and hence resolve uncertainty about the rate parameters?

Observed Trends plus 2000 runs of the model

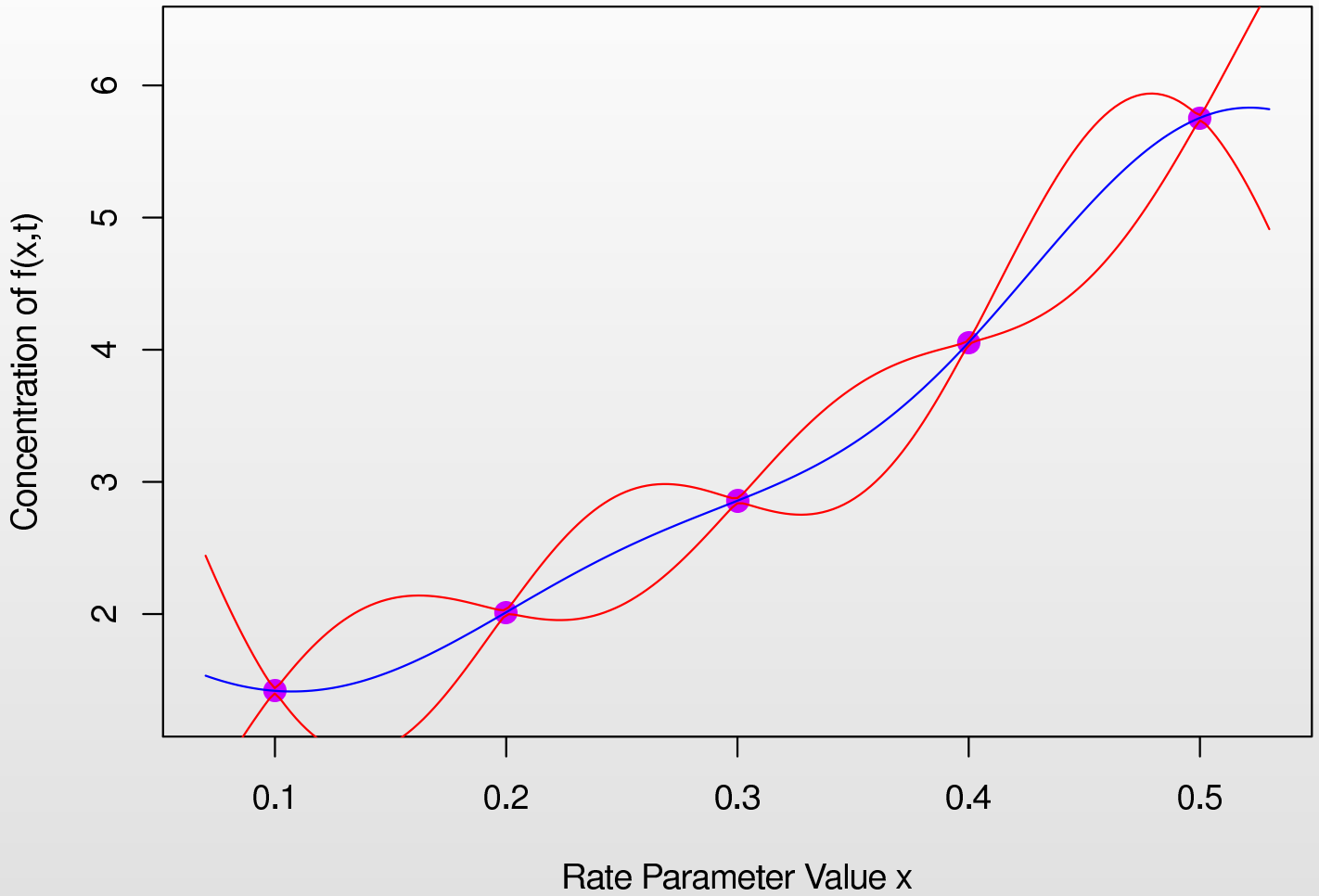


Emulation: 1D example



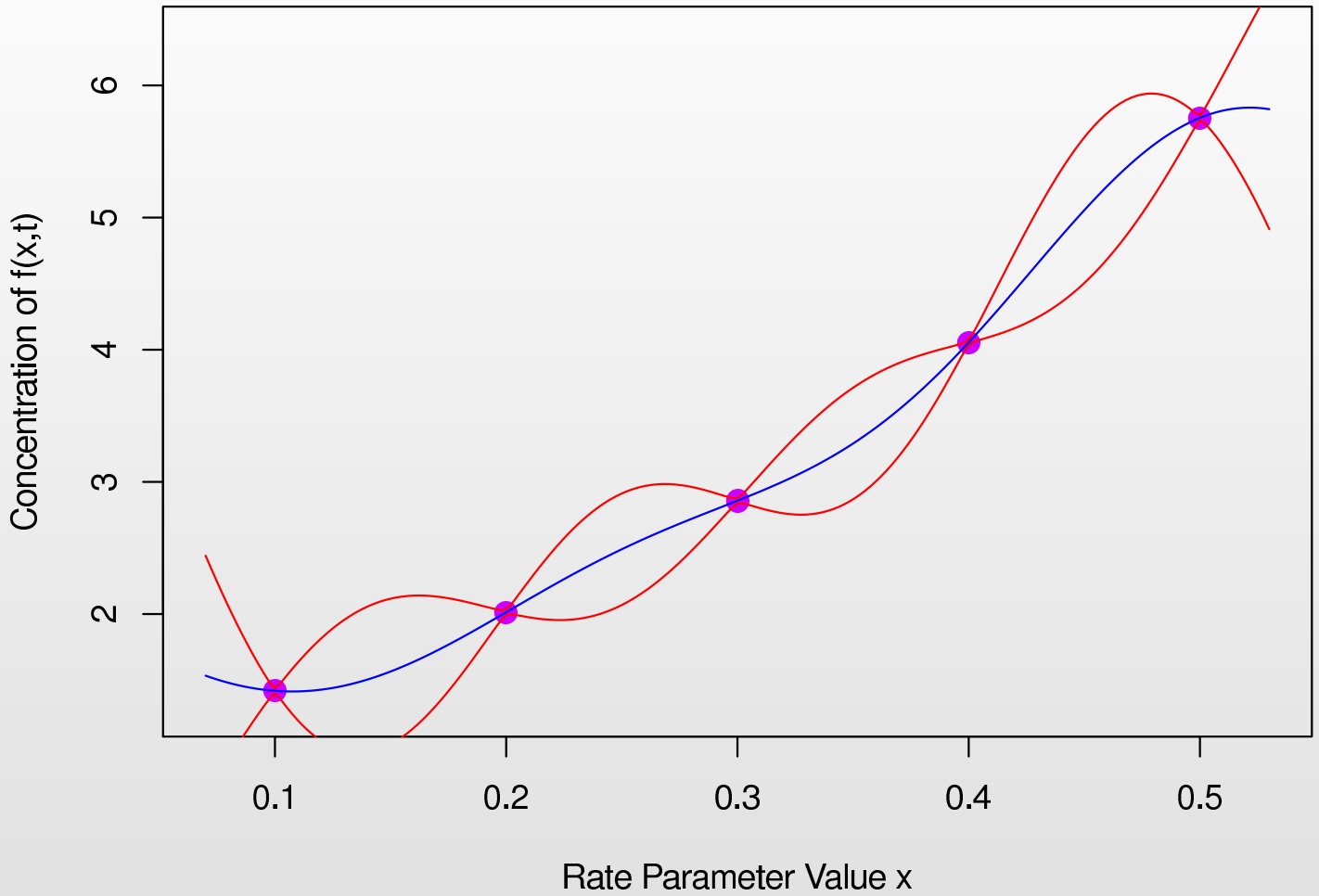
- Consider the graph of $f(x)$: in general we do not have the analytic solution of $f(x)$, here given by the dashed line.
- Instead we only have a finite number of runs of the model, in this case five.

Emulation: 1D example



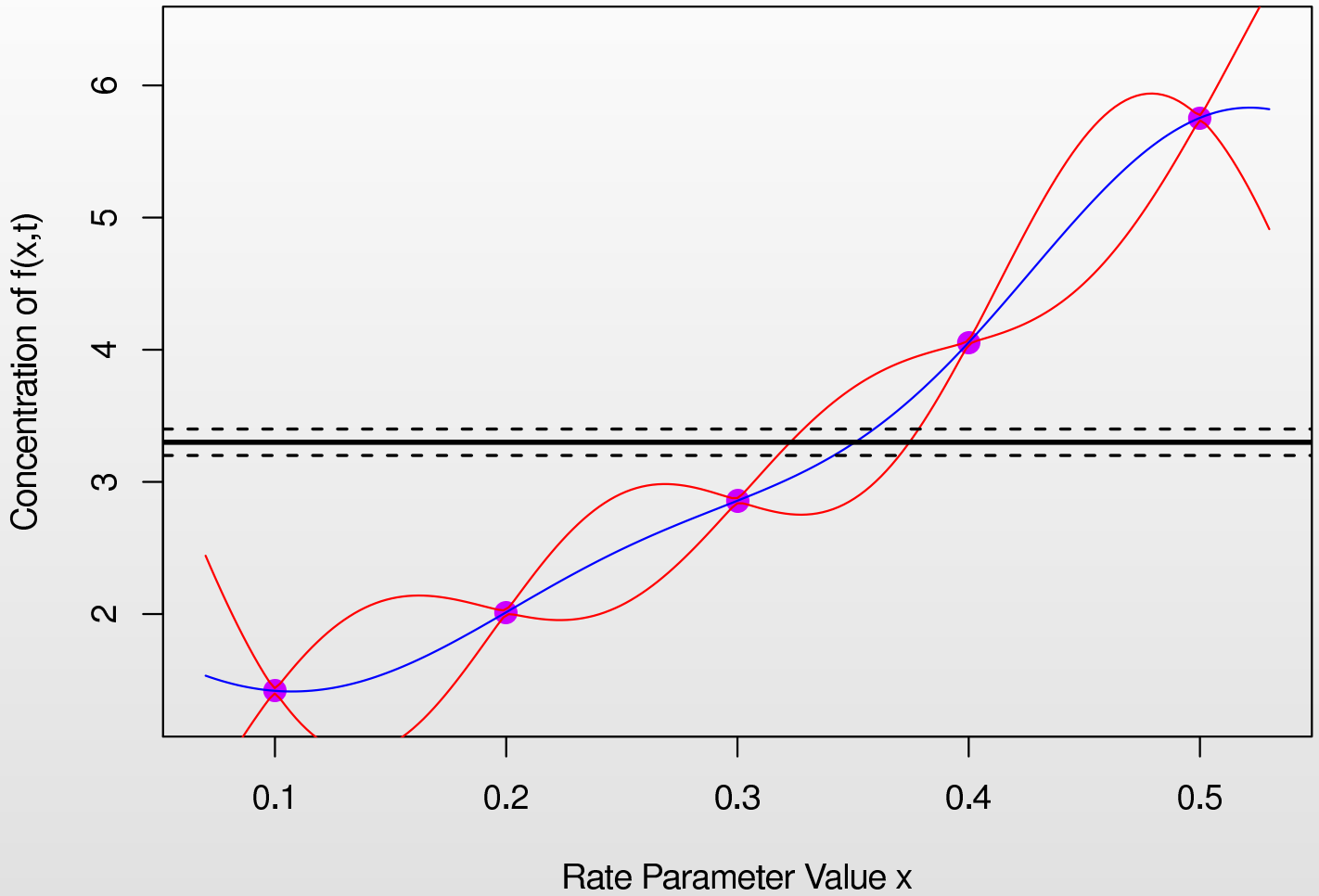
- The emulator can be used to represent our beliefs about the behaviour of the model at untested values of x, and is fast to evaluate.

Emulation: 1D example



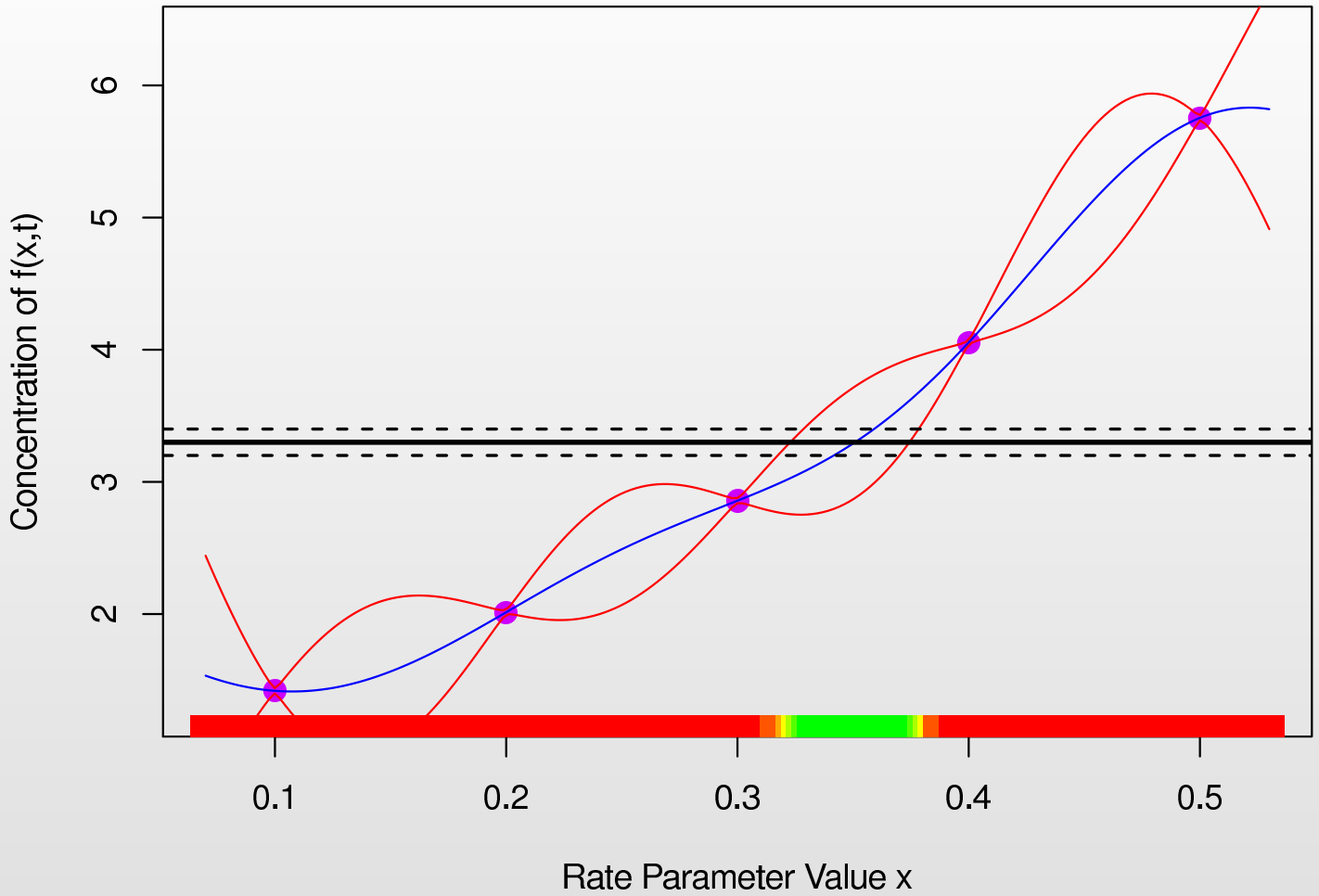
- The emulator can be used to represent our beliefs about the behaviour of the model at untested values of x, and is fast to evaluate.
- Gives the expected value of $f(x)$ (blue line) along with a credible interval for $f(x)$ (red lines) representing the uncertainty about the model's behaviour.

Implausibility Measures: 1D example



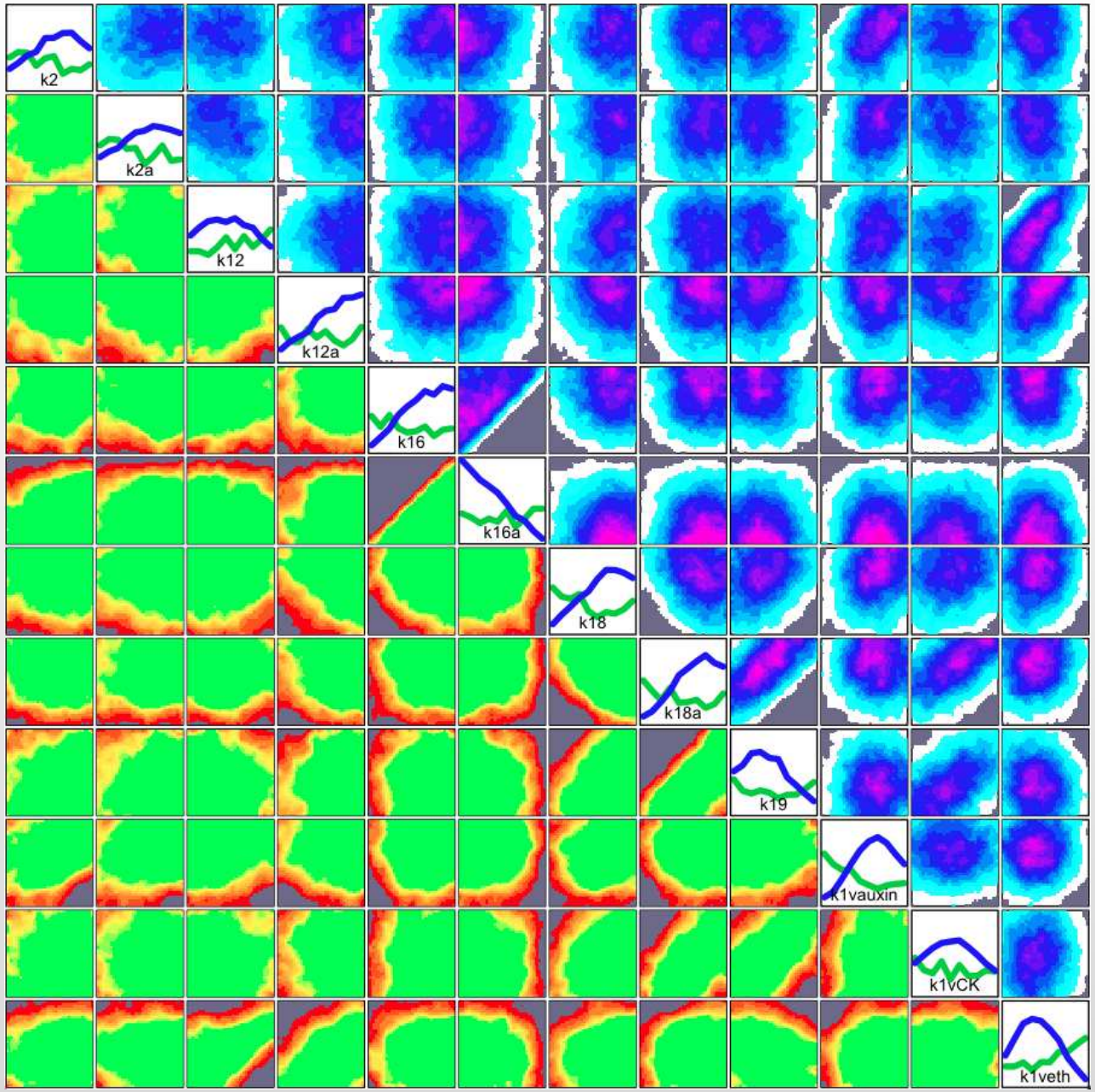
- Comparing the emulator to the observed measurement we again identify the set of x values currently consistent with this data (the observed errors here have been reduced for clarity).

Implausibility Measures: 1D example

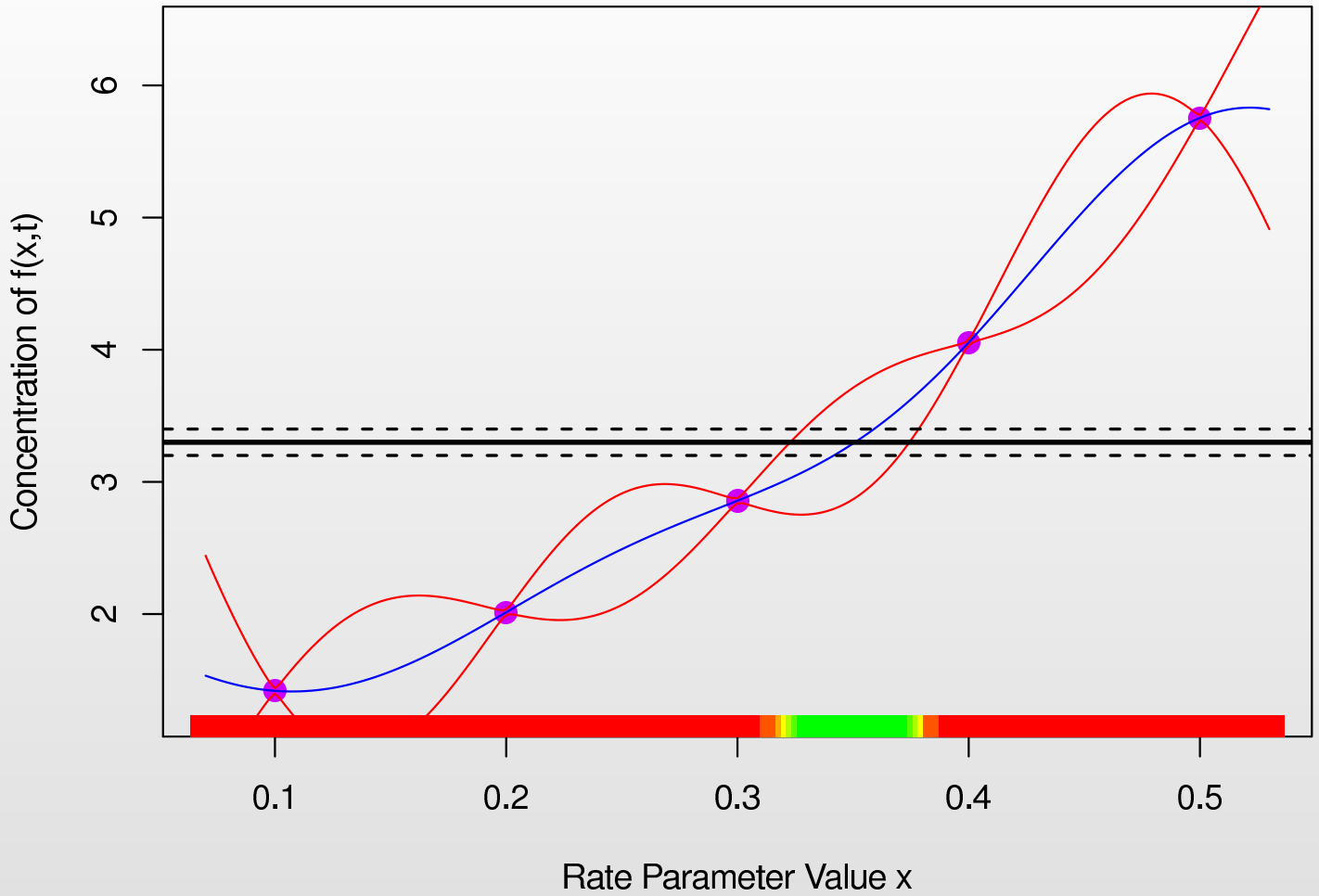


- Comparing the emulator to the observed measurement we again identify the set of x values currently consistent with this data (the observed errors here have been reduced for clarity).
- Note: uncertainty on x now includes uncertainty coming from the emulator.

Implausibility Measures: Arabidopsis Model

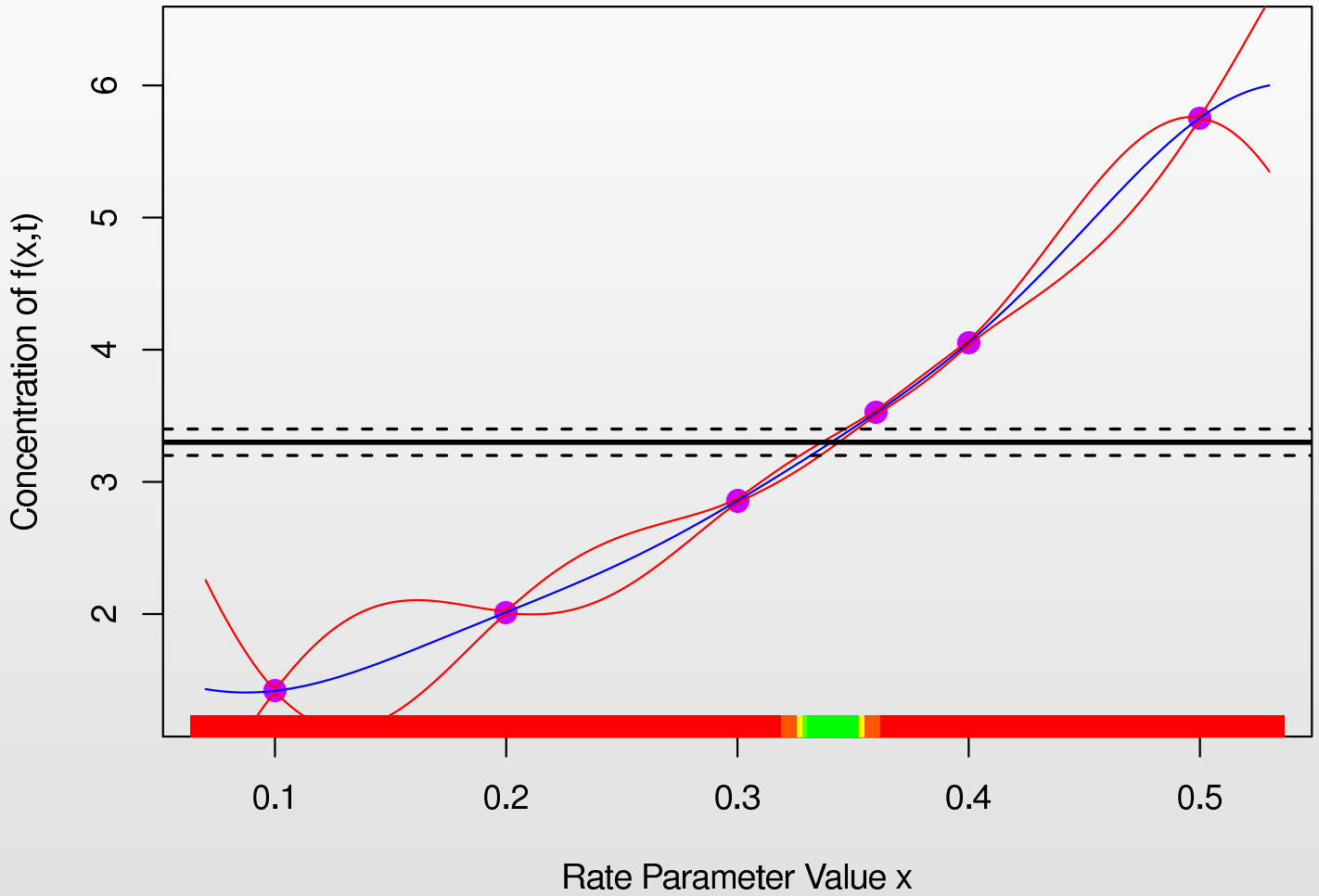


Iterative Input Space Reduction: 1D example



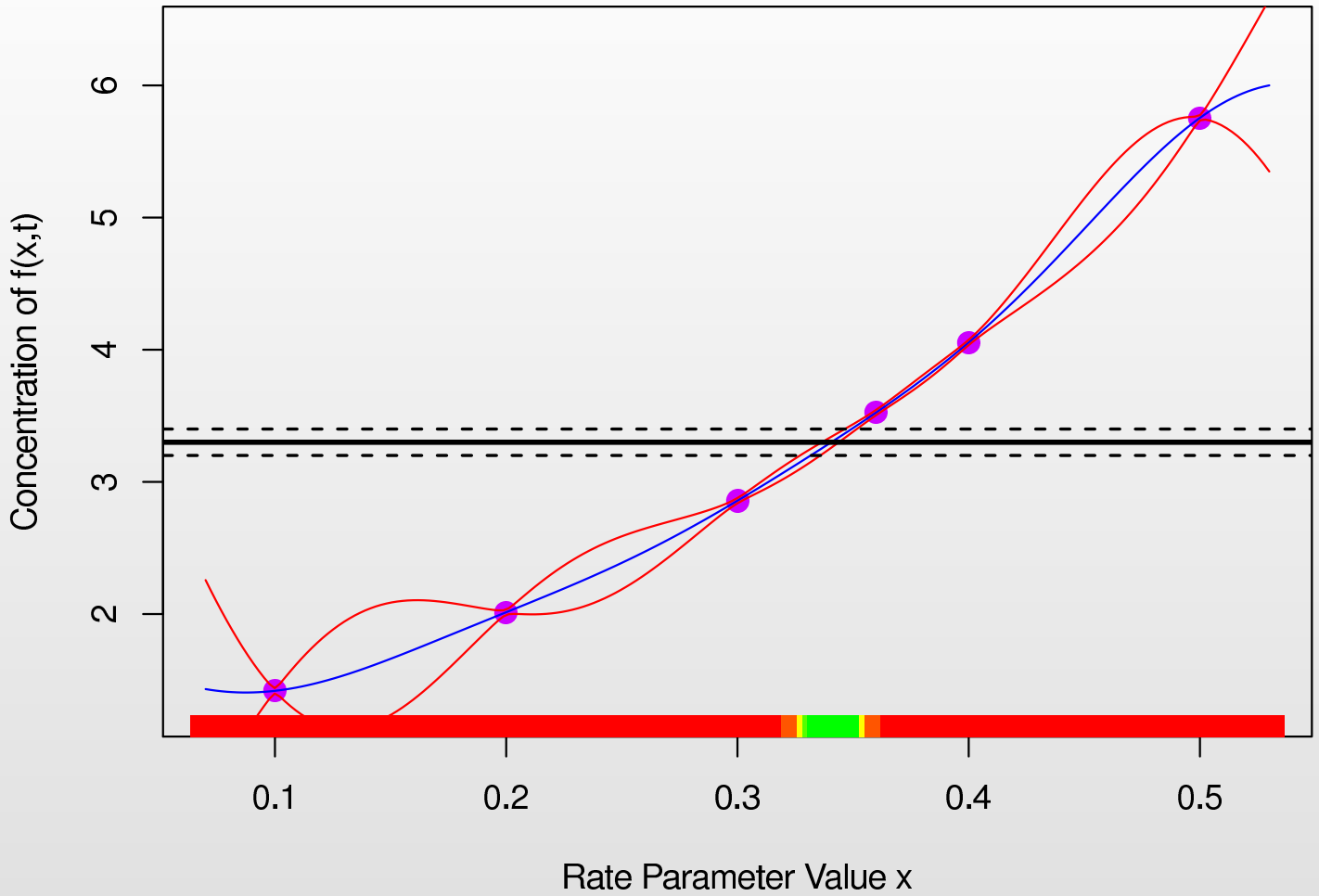
- Comparing the emulator to the observed measurement we again identify the set of x values currently consistent with this data (the observed errors here have been reduced for clarity).
- Note: uncertainty on x now includes uncertainty coming from the emulator.

Iterative Input Space Reduction: 1D example



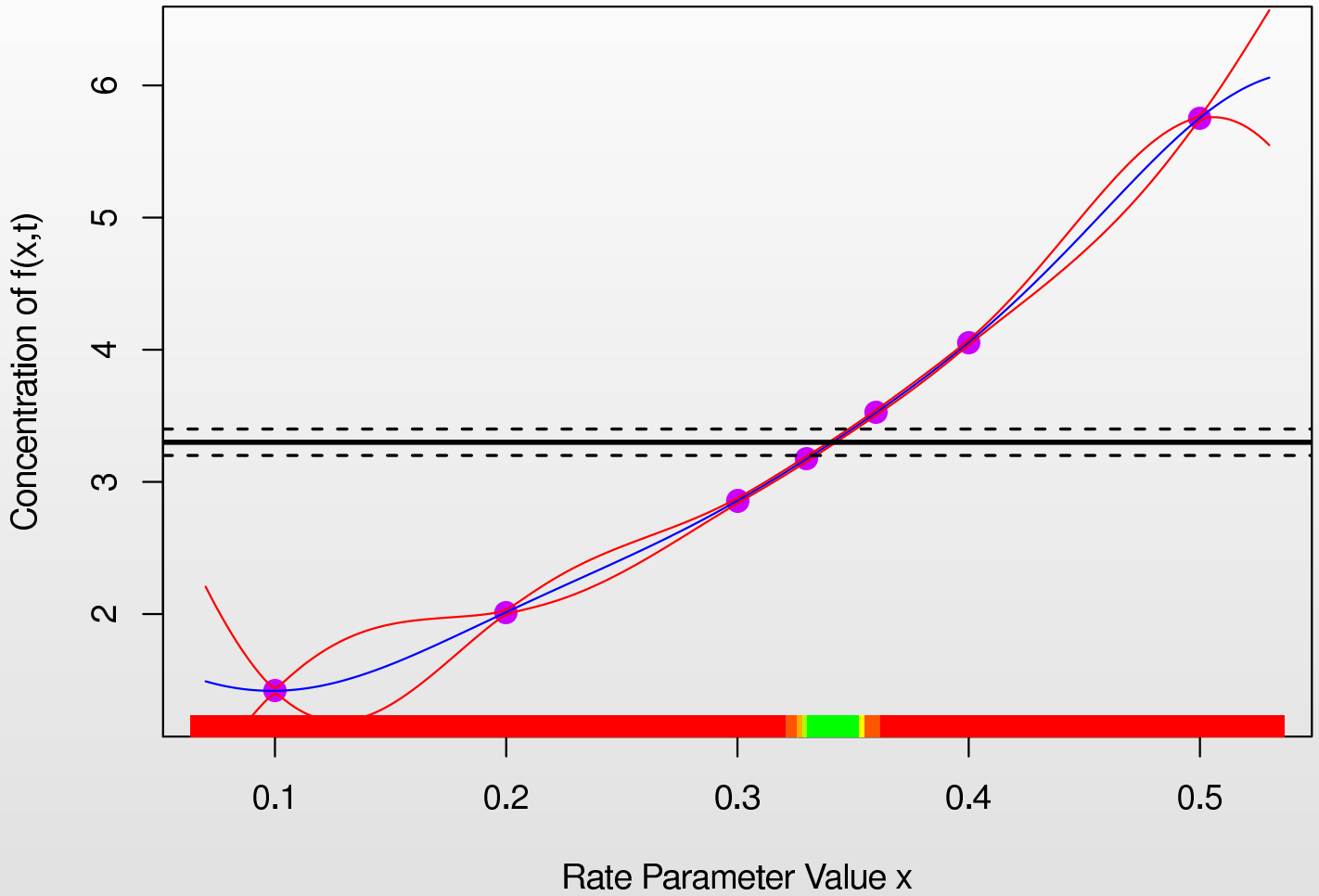
- We perform a 2nd iteration or wave of runs to improve emulator accuracy.

Iterative Input Space Reduction: 1D example



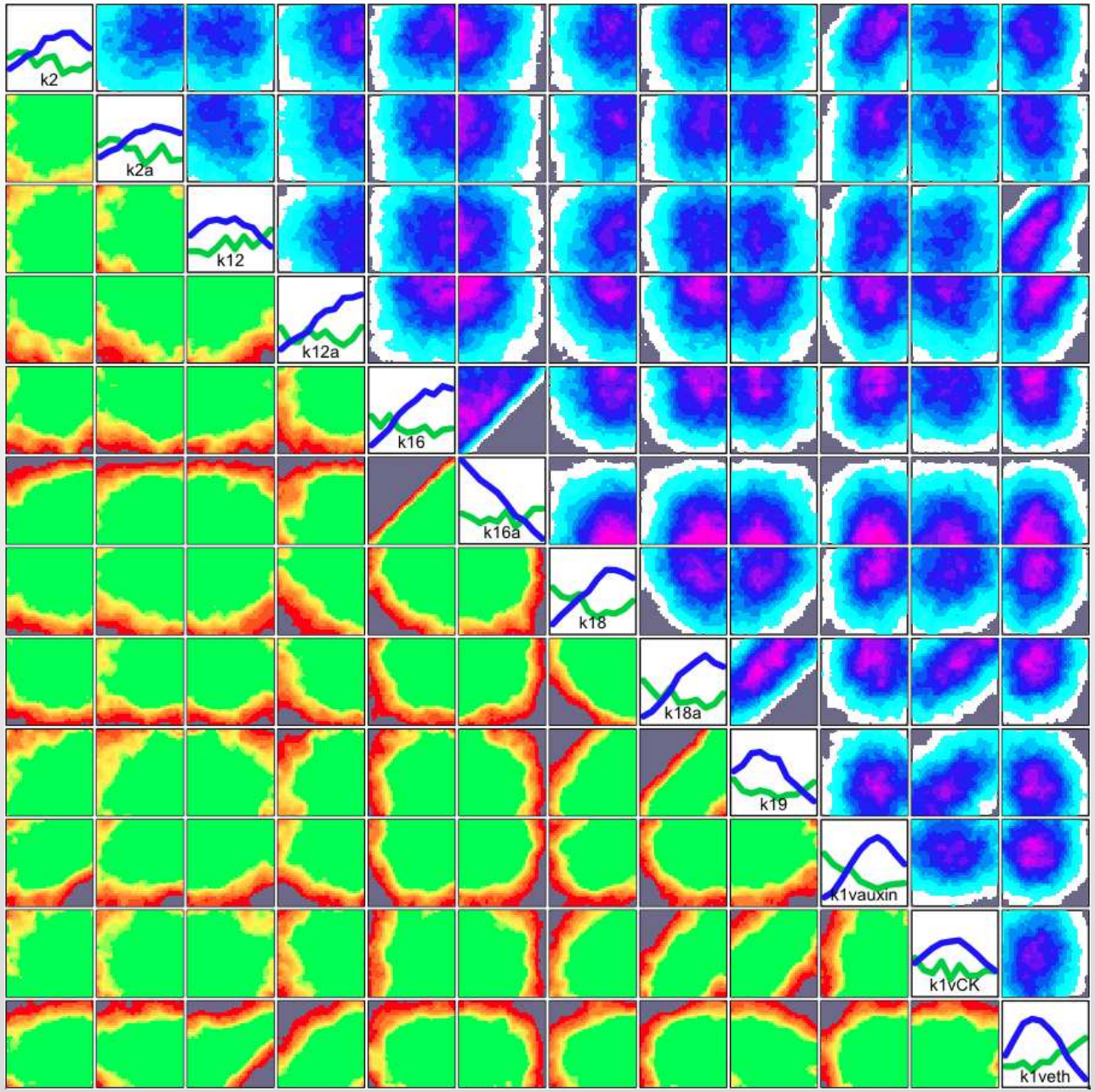
- We perform a 2nd iteration or wave of runs to improve emulator accuracy.
- The runs are located only at non-implausible (green/yellow) points.

Iterative Input Space Reduction: 1D example

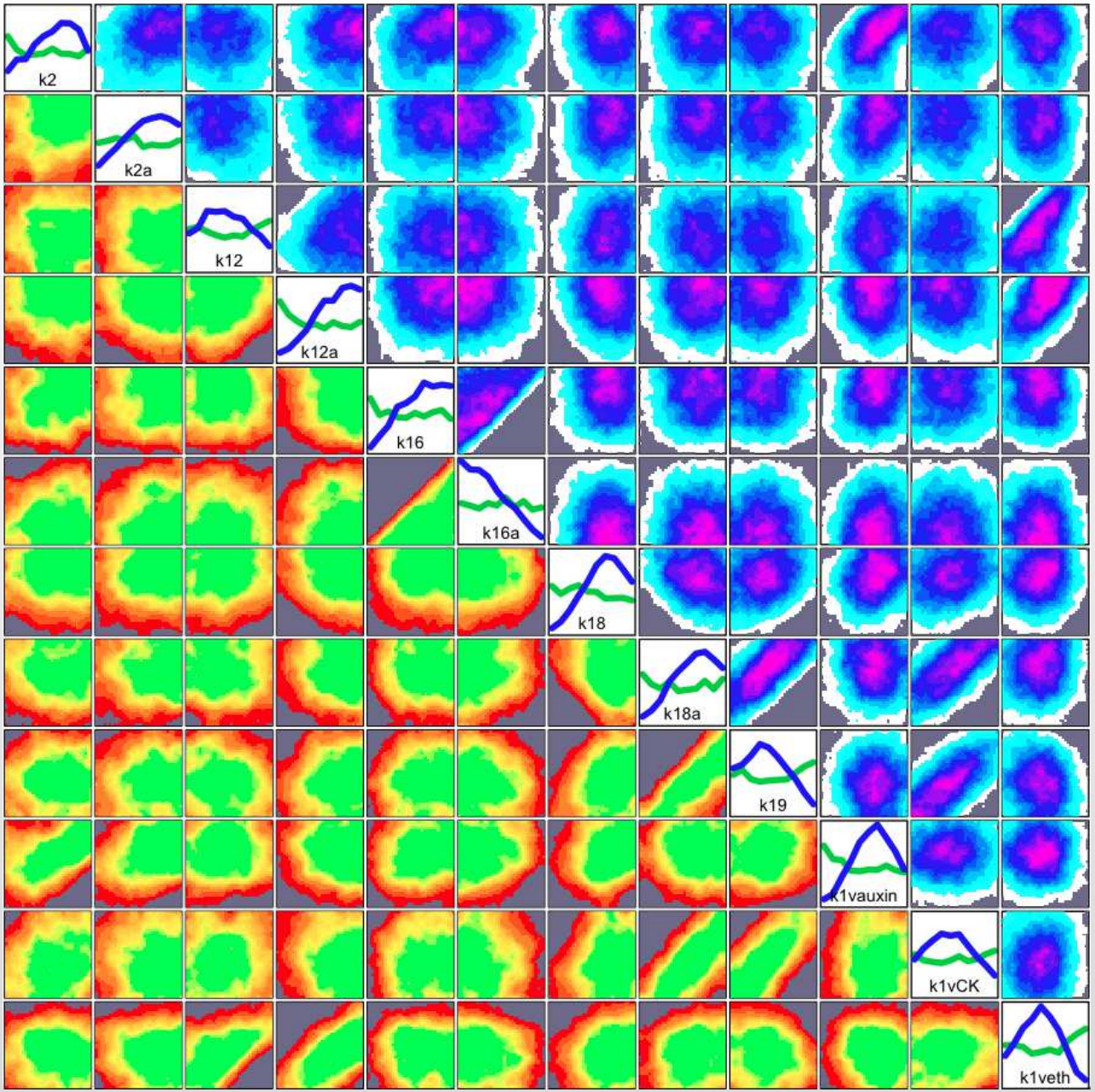


- We perform a 2nd iteration or wave of runs to improve emulator accuracy.
- The runs are located only at non-implausible (green/yellow) points.
- Now the emulator is more accurate than the observations, and we can identify the set of all x values of interest.

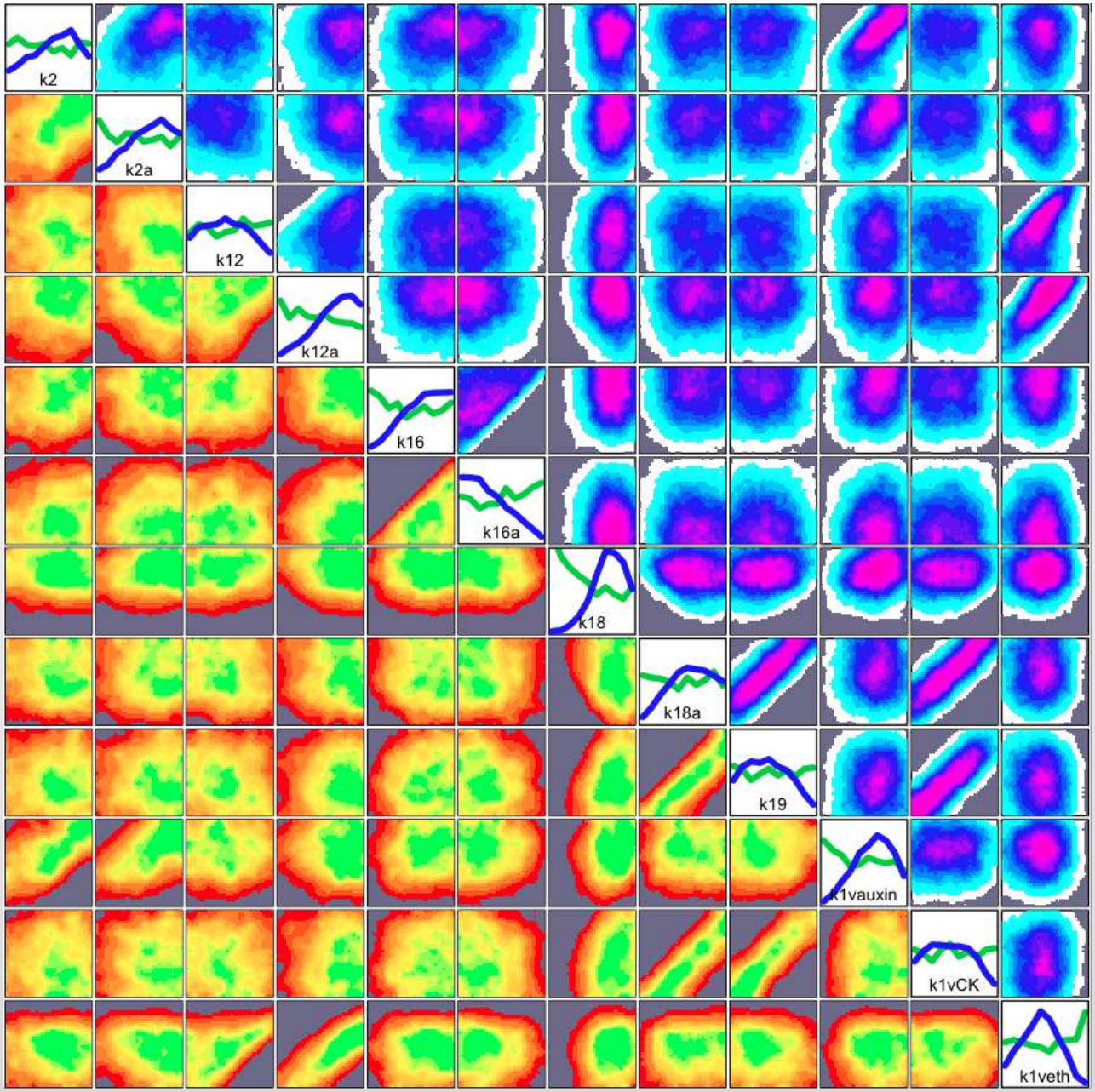
Iterative Input Space Reduction: Arabidopsis Model Wave 1



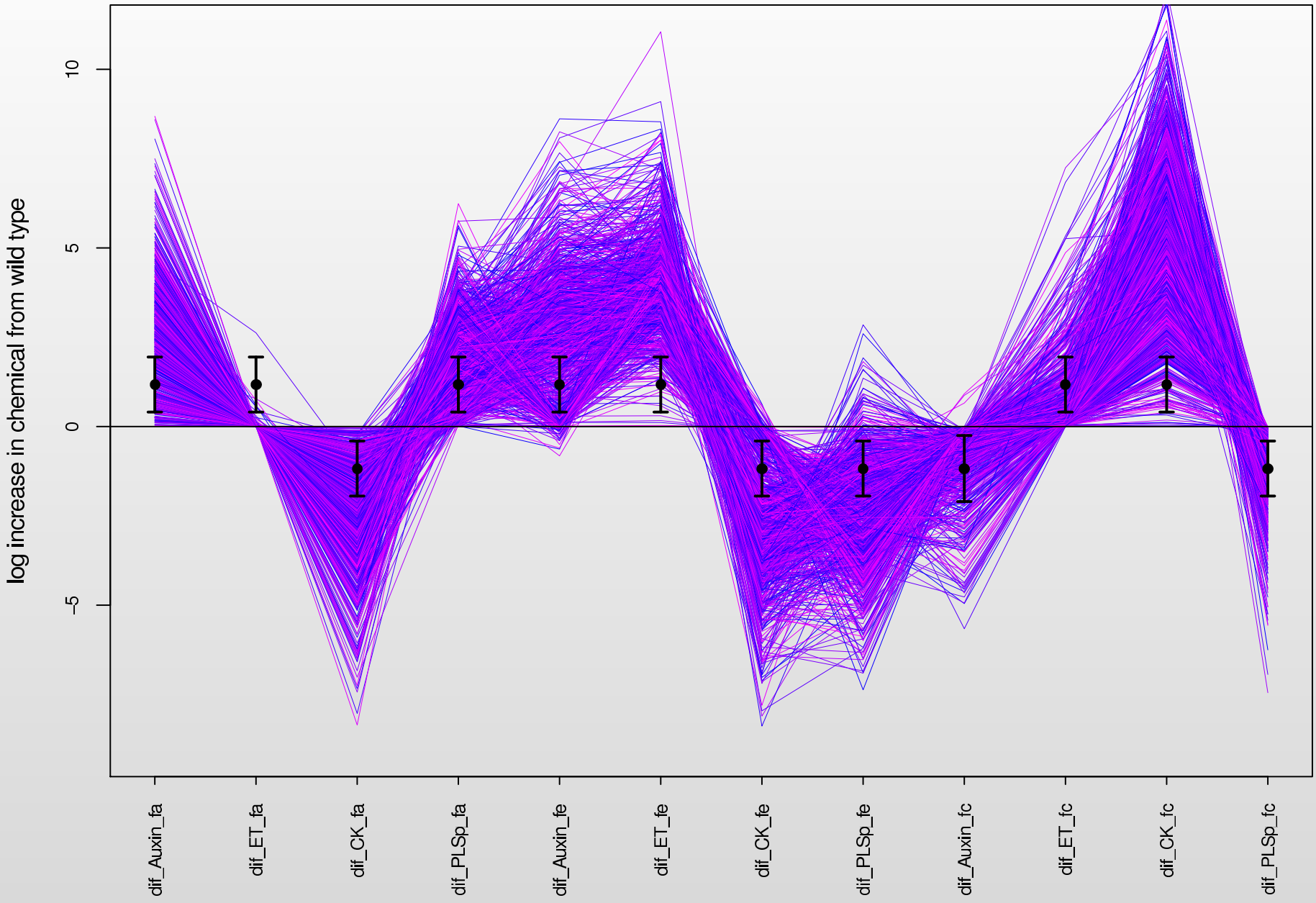
Iterative Input Space Reduction: Arabidopsis Model Wave 2



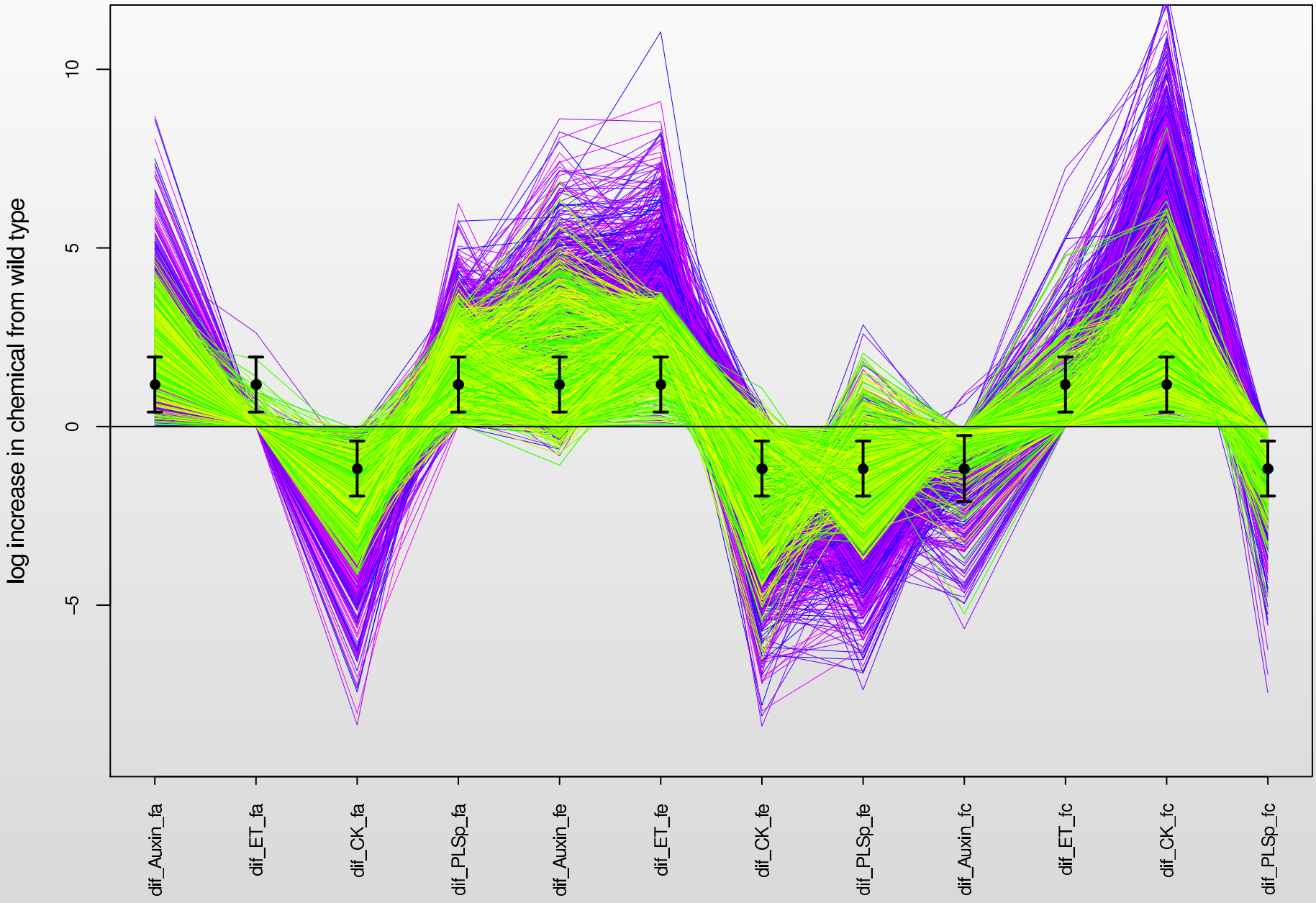
Iterative Input Space Reduction: Arabidopsis Model Wave 3



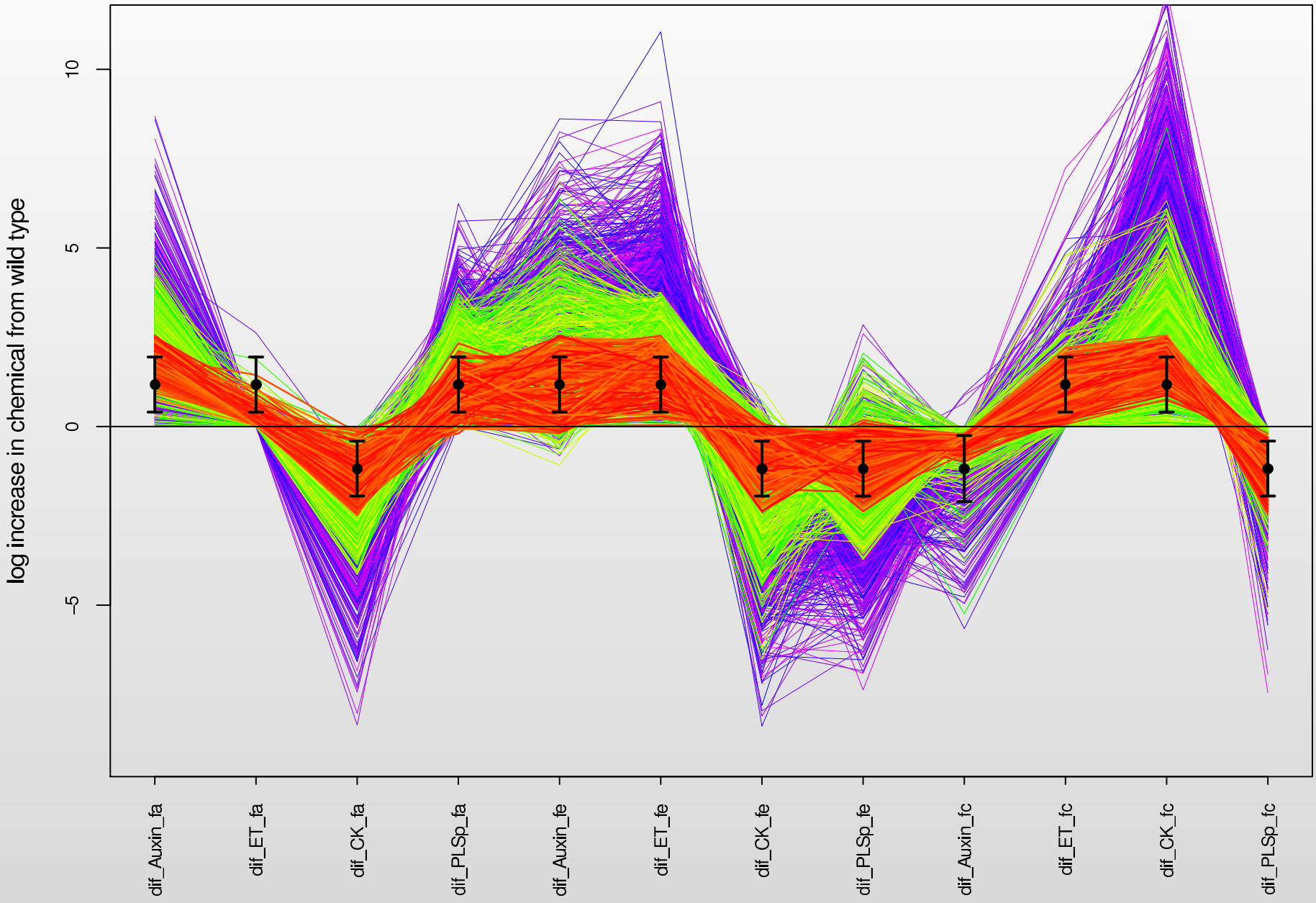
Iterative Strategy for Arabidopsis Model: Wave 1



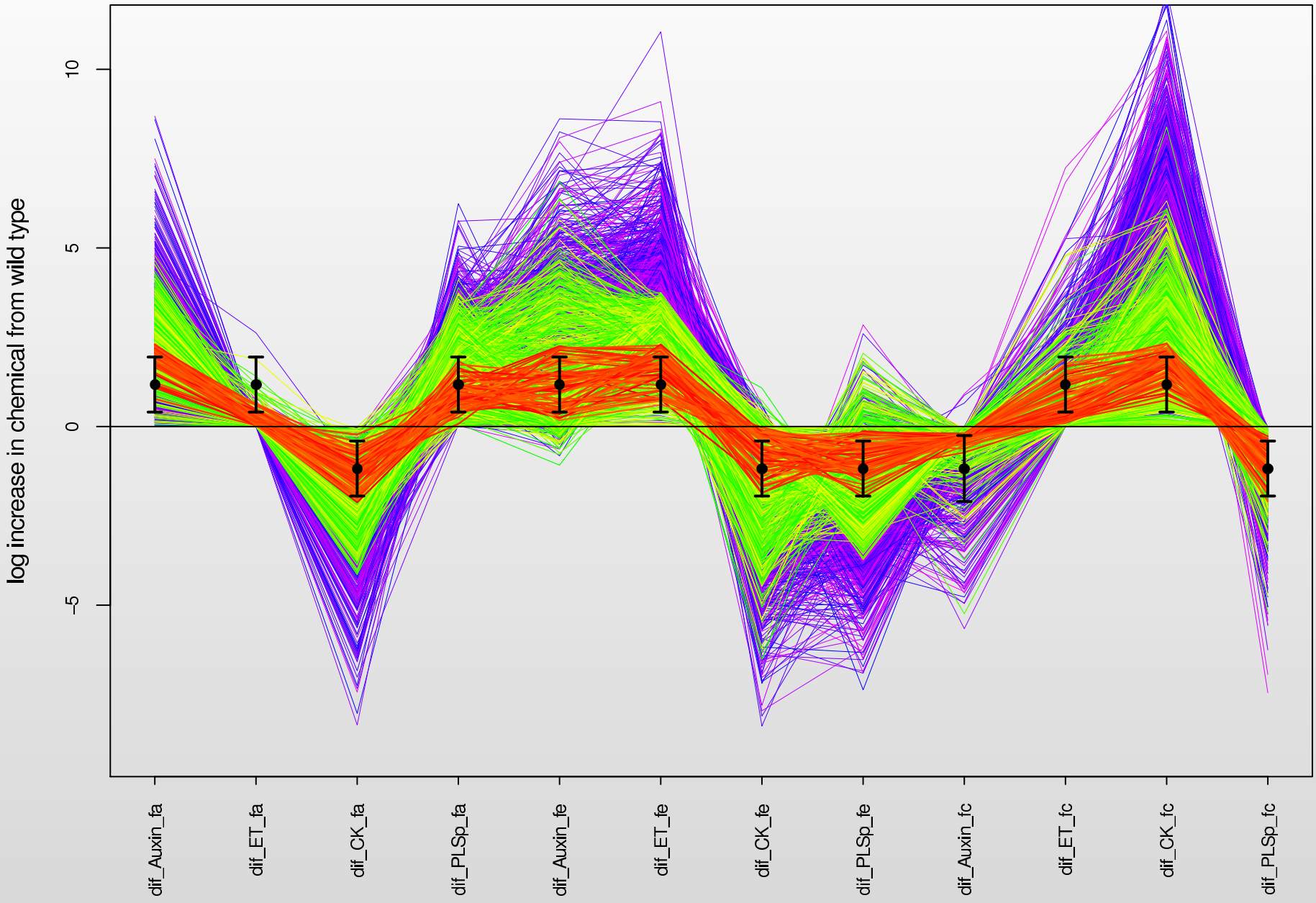
Iterative Strategy for Arabidopsis Model: Waves 1 and 2



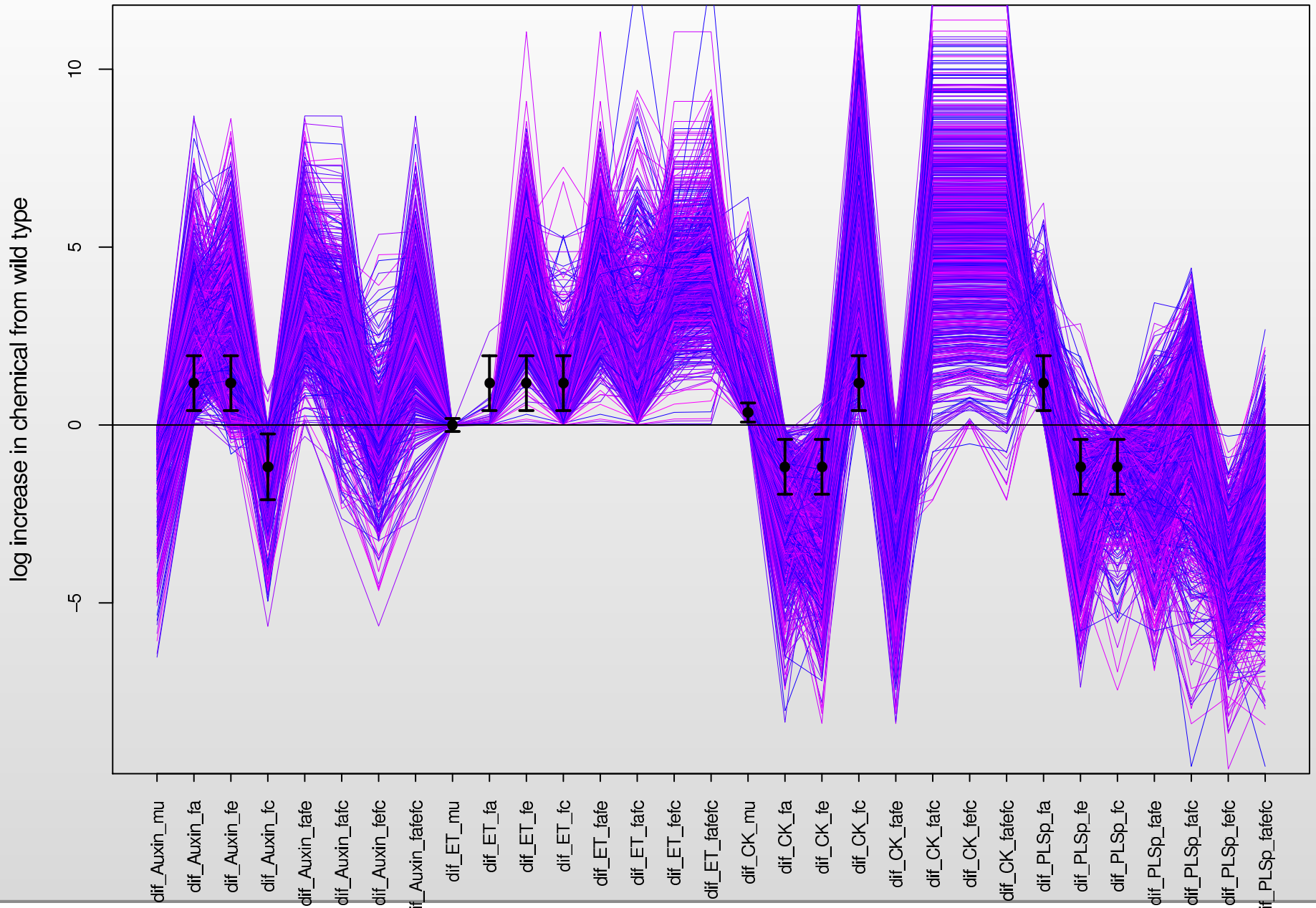
Iterative Strategy for Arabidopsis Model: Wave 1, 2 and 3



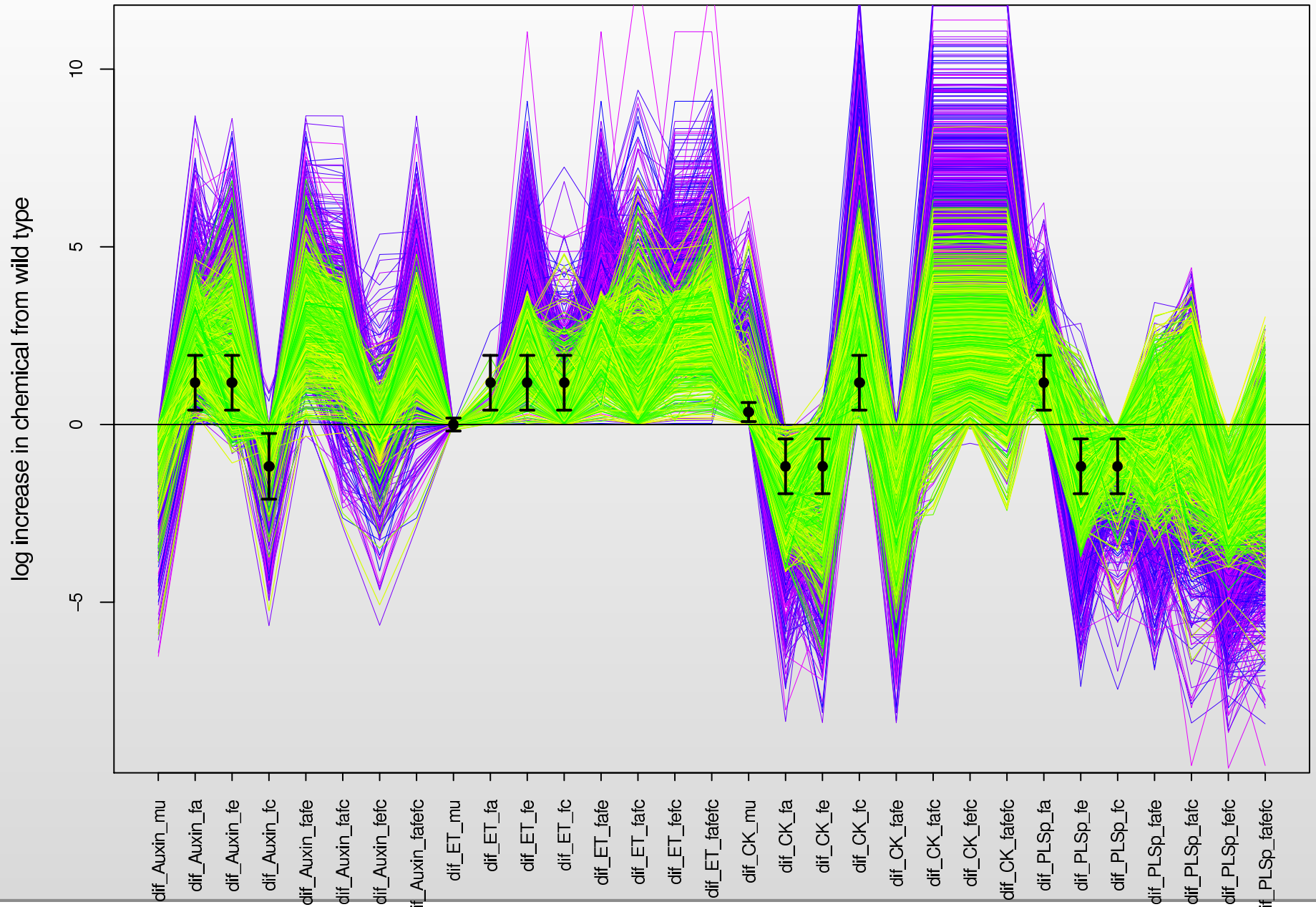
Iterative Strategy for Arabidopsis Model: Wave 1, 2 and 3



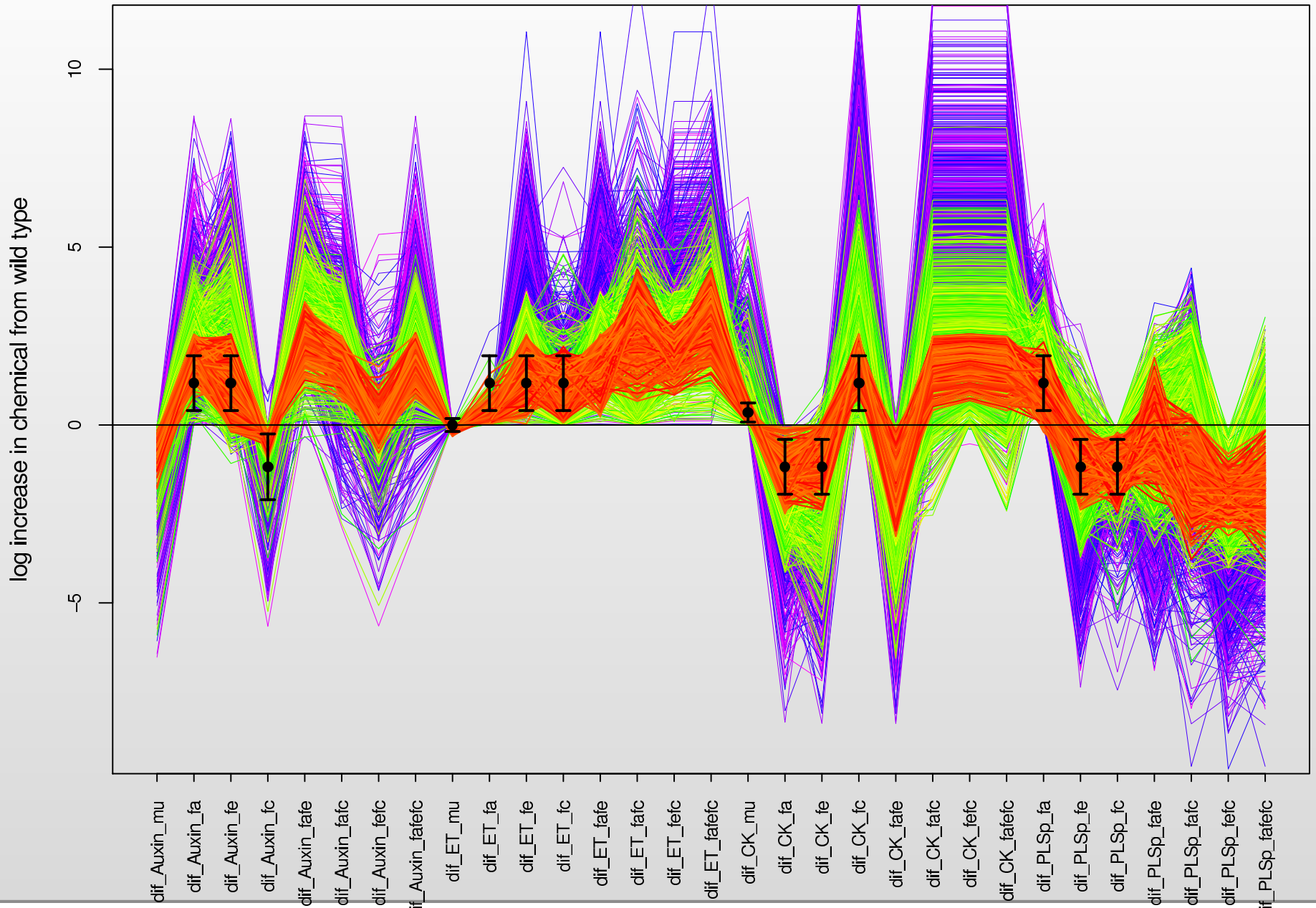
Iterative Strategy for Arabidopsis Model: Wave 1



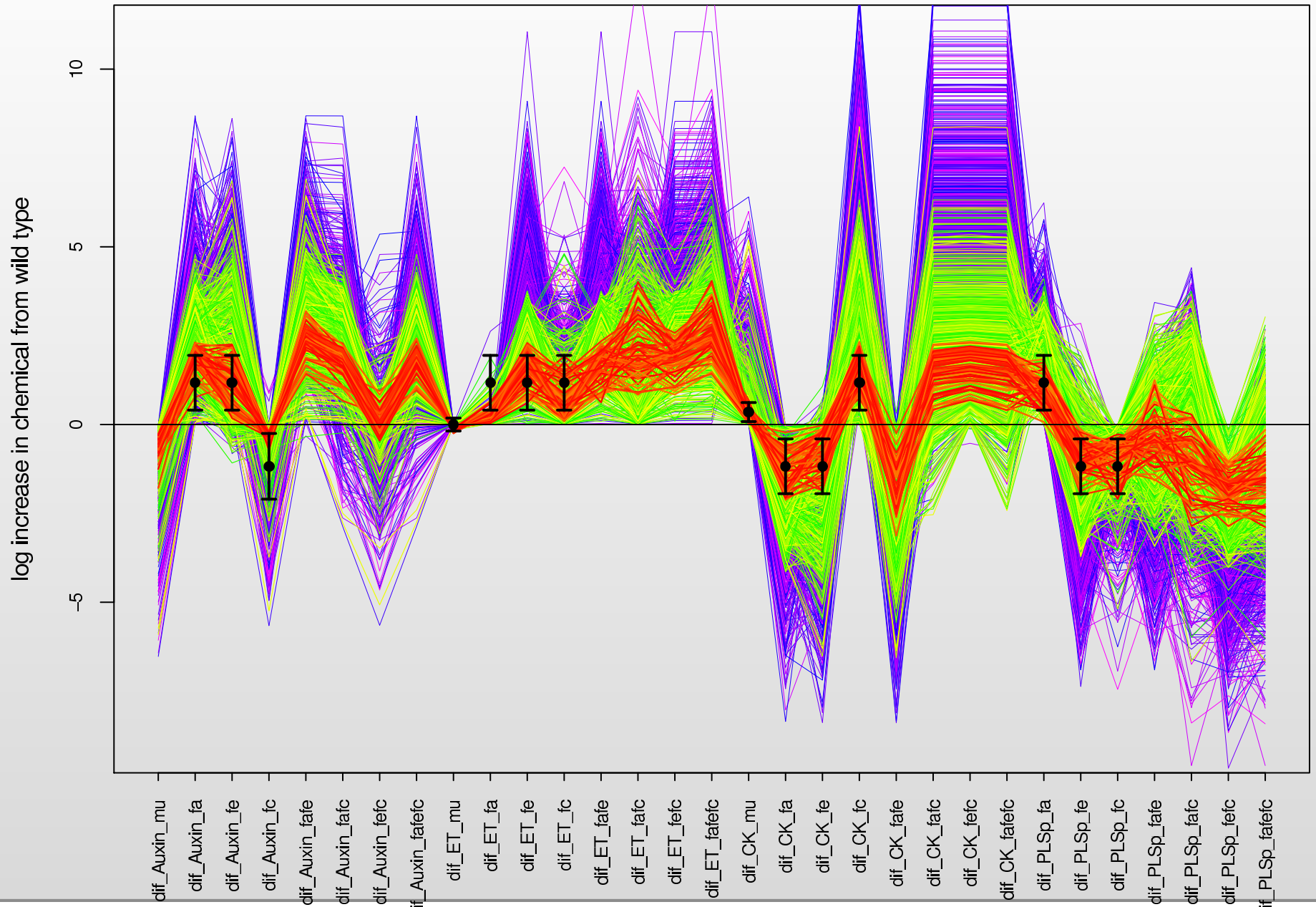
Iterative Strategy for Arabidopsis Model: Waves 1 and 2



Iterative Strategy for Arabidopsis Model: Waves 1, 2 and 3



Iterative Strategy for Arabidopsis Model: Waves 1, 2 and 3



Designing New Experiments

- We now have found several runs that belong to the set $\mathcal{X}$, consistent with 16 observed trends.

Designing New Experiments

- We now have found several runs that belong to the set $\mathcal{X}$, consistent with 16 observed trends.
- We have funding for 4 additional experiments: want to choose these to maximise space reduction (to reduce the size of $\mathcal{X}$), to learn about the inputs x.

Designing New Experiments

- We now have found several runs that belong to the set $\mathcal{X}$, consistent with 16 observed trends.
- We have funding for 4 additional experiments: want to choose these to maximise space reduction (to reduce the size of $\mathcal{X}$), to learn about the inputs x.
- We are considering a class of 96 experiments formed from a combination of plant type, chemical measured and feeding regime, which leaves 96 - 16 = 80 possible future experiments to choose from.

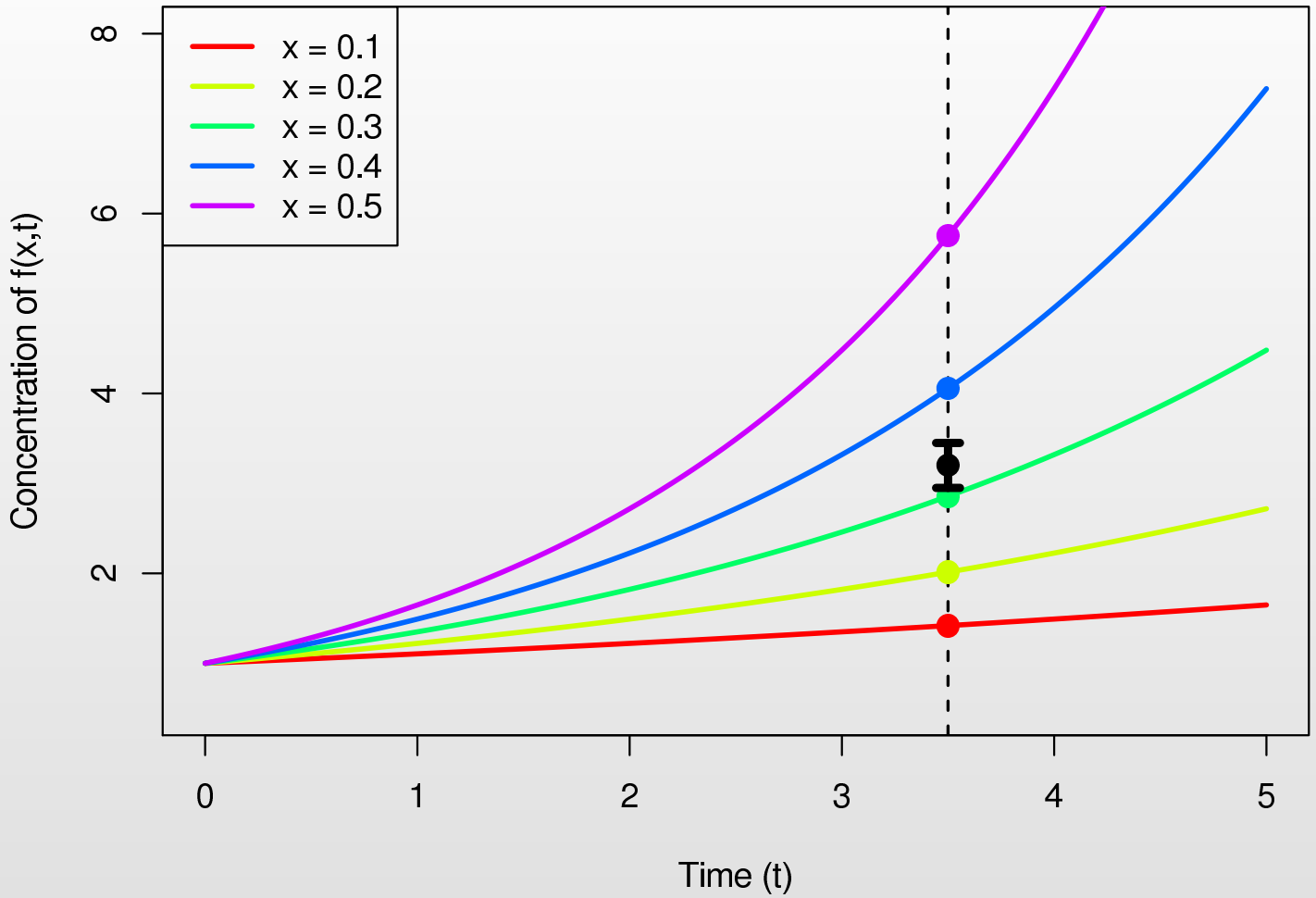
Designing New Experiments

- We now have found several runs that belong to the set $\mathcal{X}$, consistent with 16 observed trends.
- We have funding for 4 additional experiments: want to choose these to maximise space reduction (to reduce the size of $\mathcal{X}$), to learn about the inputs x.
- We are considering a class of 96 experiments formed from a combination of plant type, chemical measured and feeding regime, which leaves 96 - 16 = 80 possible future experiments to choose from.
- We will select 4 experiments from 80 based on an expected space reduction criteria, using implausibility measures.

Designing New Experiments

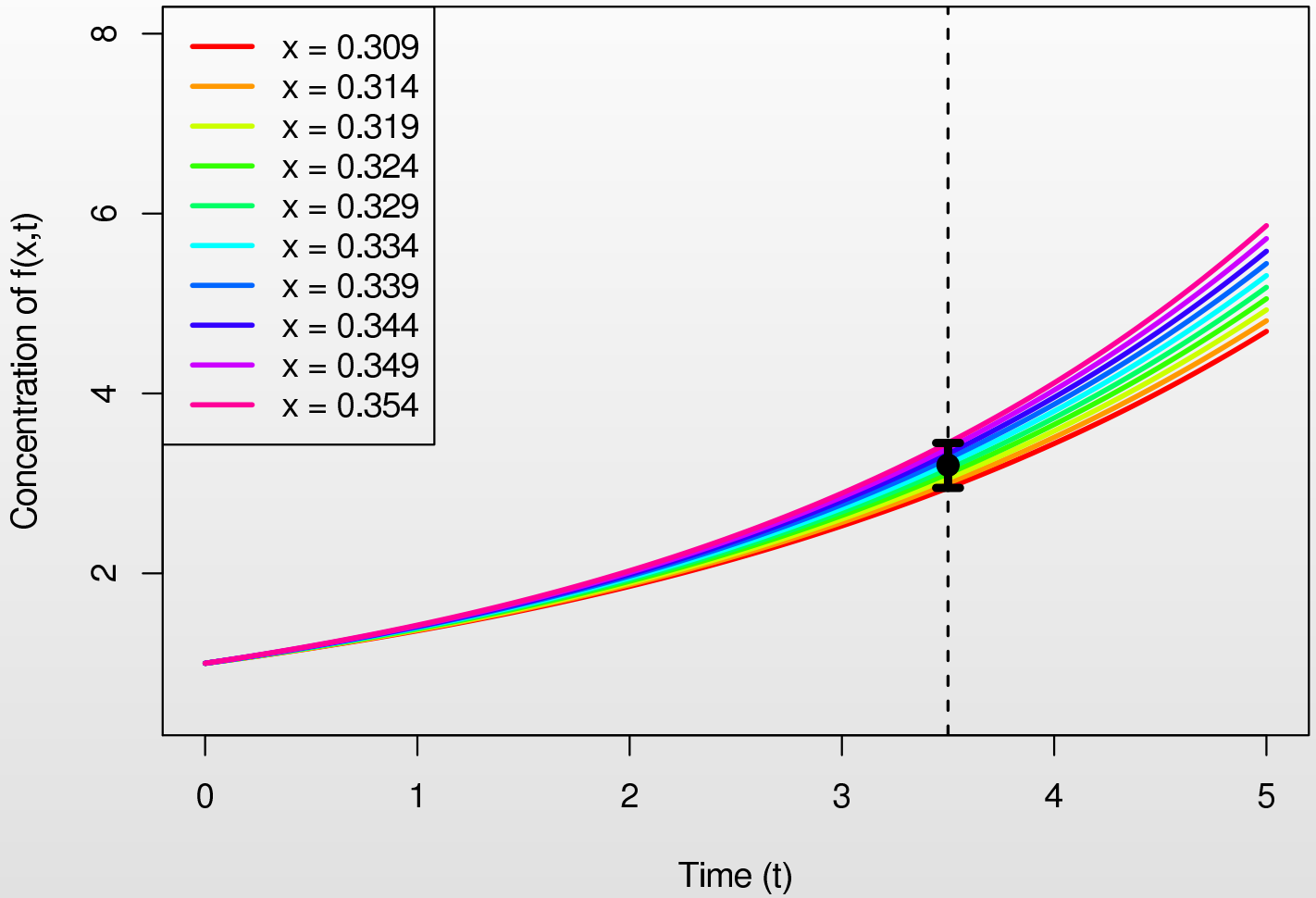
- We now have found several runs that belong to the set $\mathcal{X}$, consistent with 16 observed trends.
- We have funding for 4 additional experiments: want to choose these to maximise space reduction (to reduce the size of $\mathcal{X}$), to learn about the inputs x.
- We are considering a class of 96 experiments formed from a combination of plant type, chemical measured and feeding regime, which leaves 96 - 16 = 80 possible future experiments to choose from.
- We will select 4 experiments from 80 based on an expected space reduction criteria, using implausibility measures.
- We hence expect to learn most efficiently about the rate parameters x from this design of 4 experiments.

Designing new experiment: 1D example



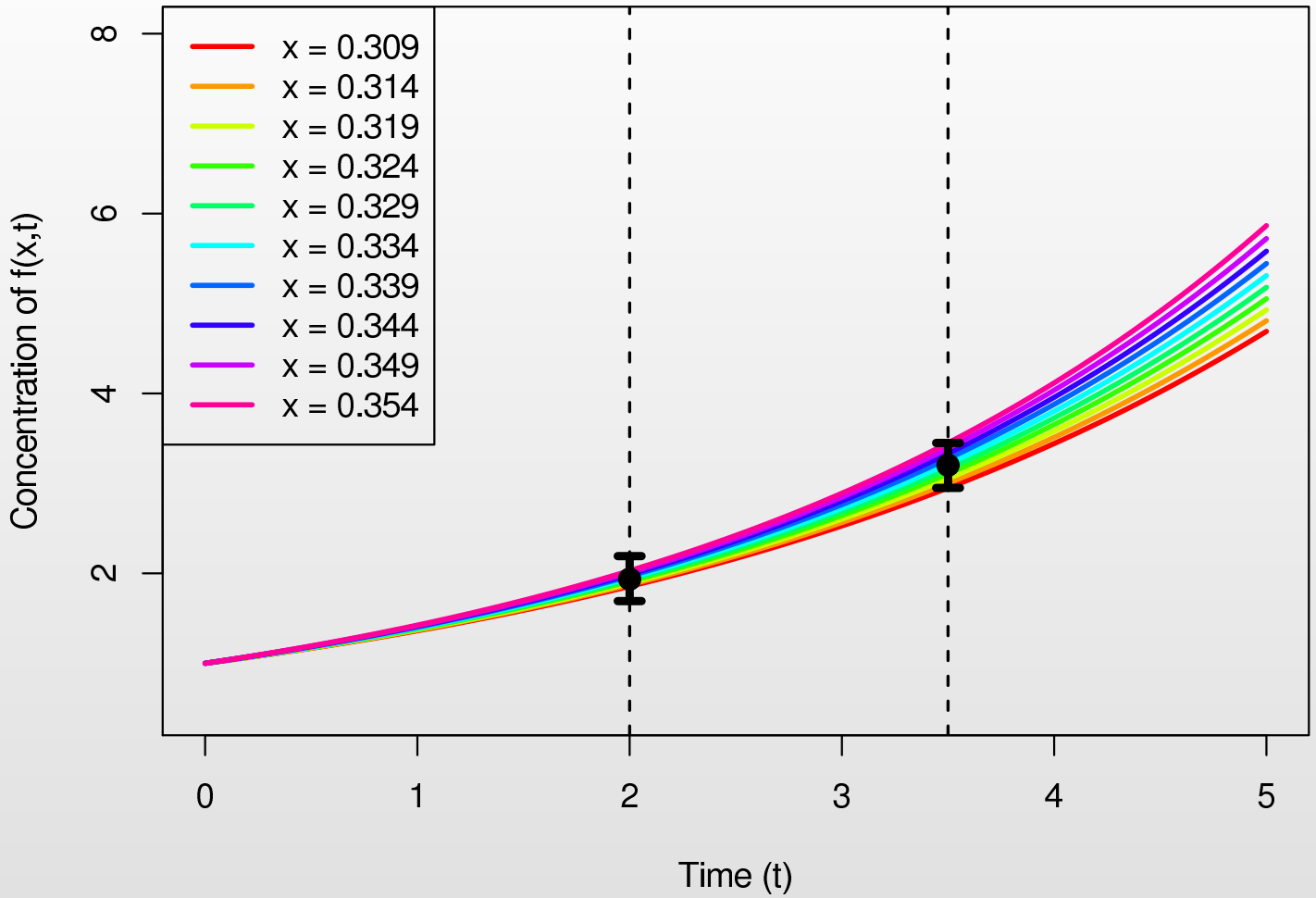
- Using the emulator we can choose several values of x consistent with the measurement of $f(x,t)$ at $t = 3.5$, and perform corresponding runs of the model.

Designing new experiment: 1D example



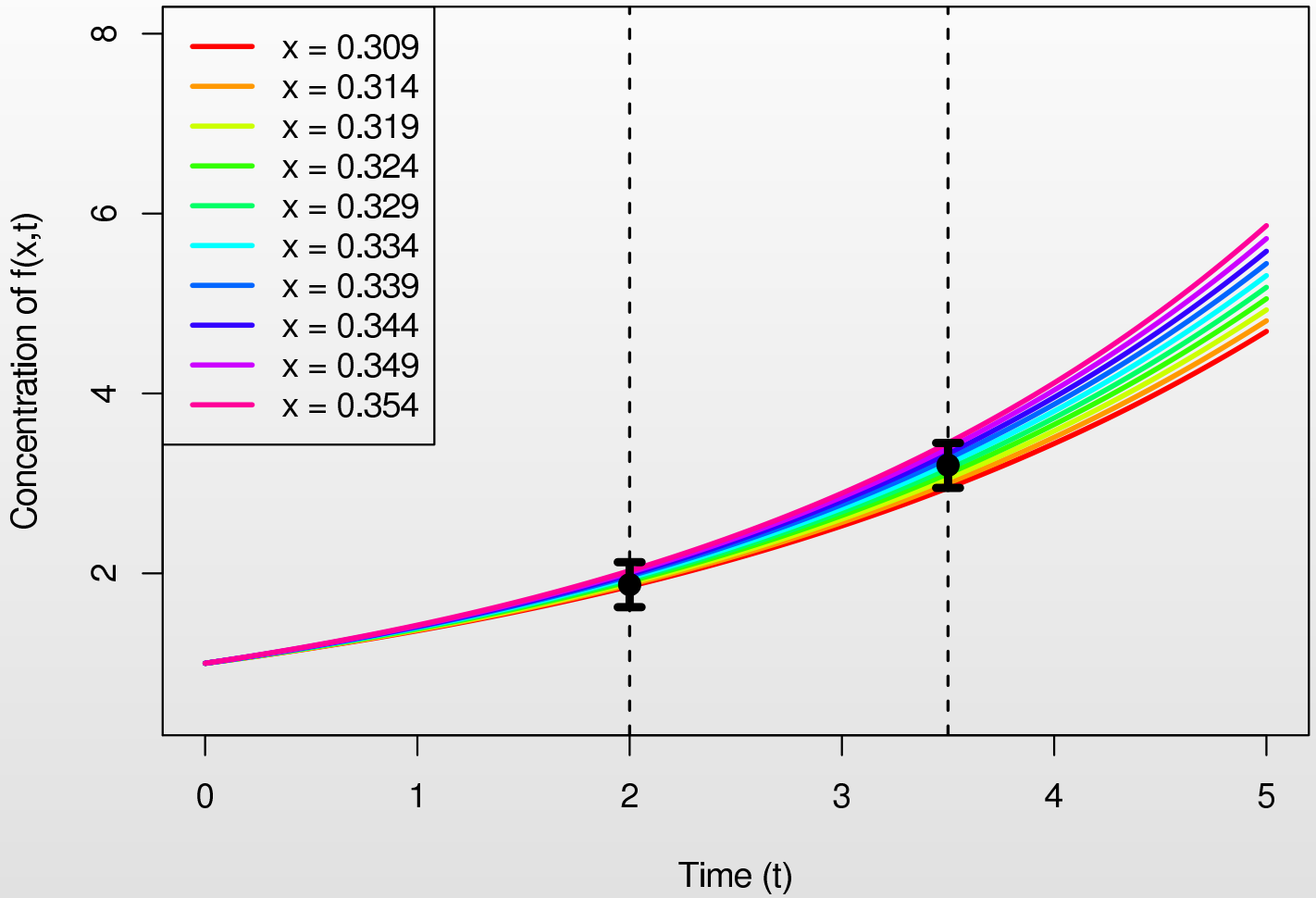
- Using the emulator we can choose several values of x consistent with the measurement of $f(x,t)$ at $t = 3.5$, and perform corresponding runs of the model.

Designing new experiment: 1D example



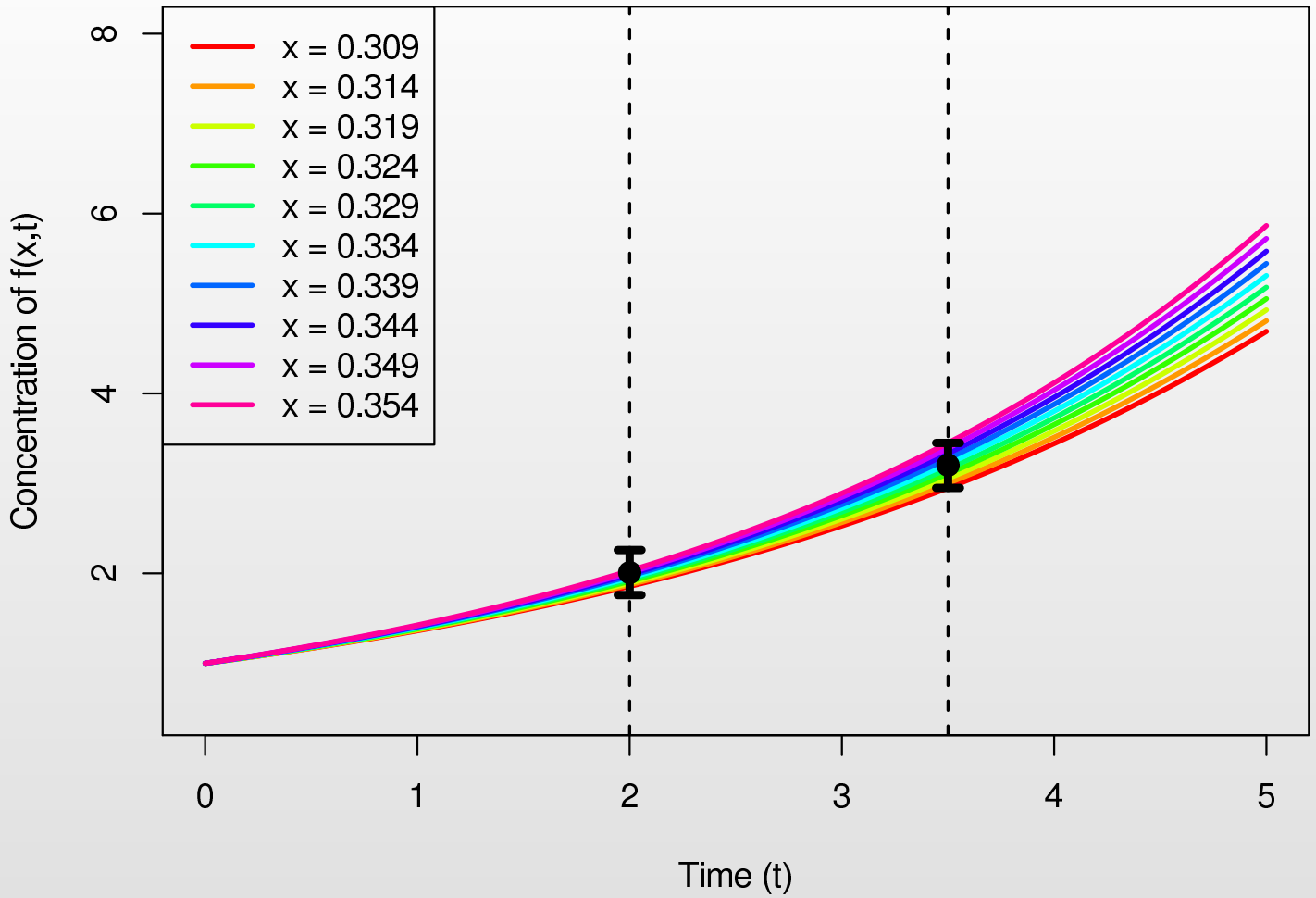
- Using the emulator we can choose several values of x consistent with the measurement of $f(x,t)$ at $t = 3.5$, and perform corresponding runs of the model.
- We can check the predictions made by these runs for $Y(t = 2)$.

Designing new experiment: 1D example



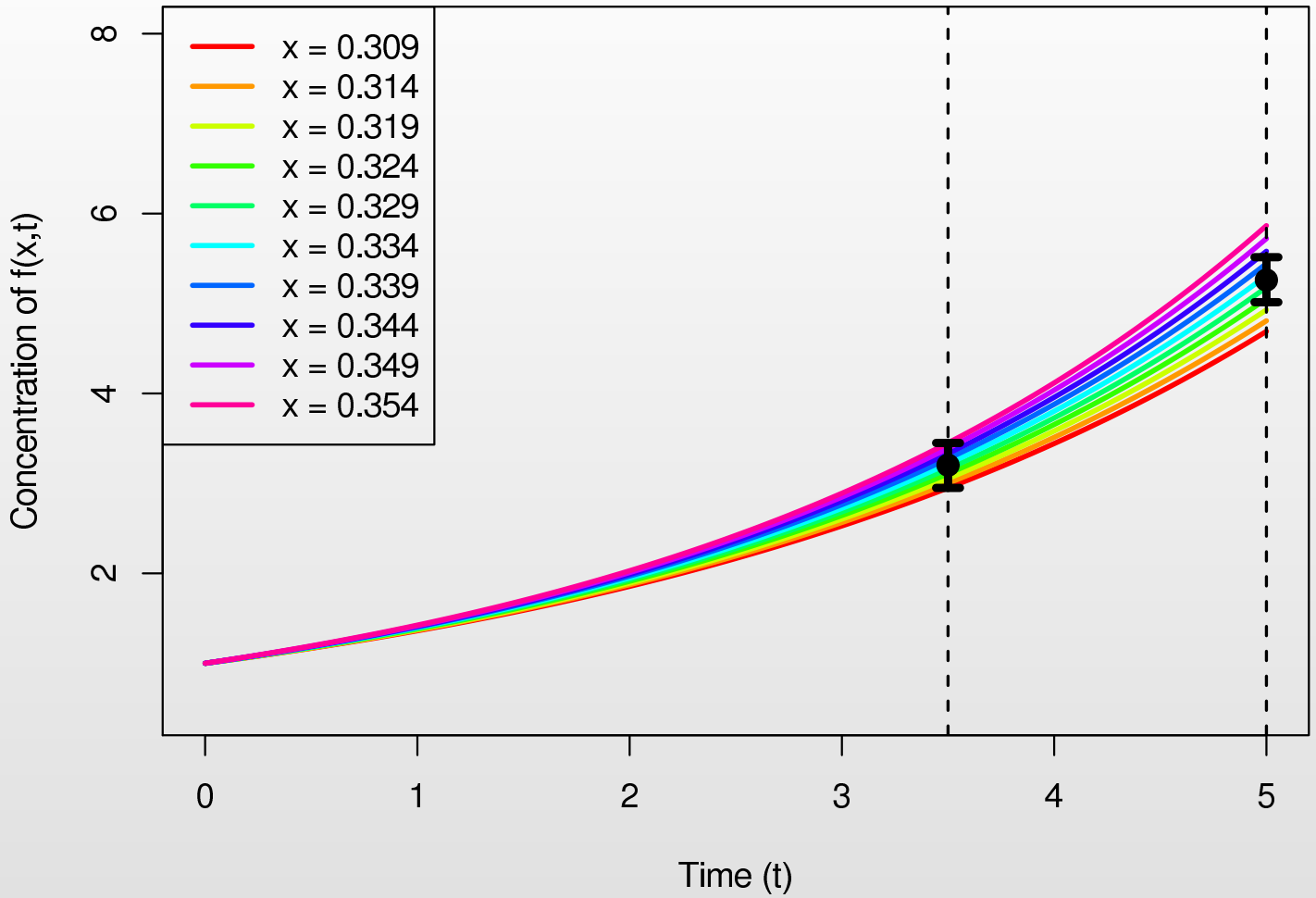
- Using the emulator we can choose several values of x consistent with the measurement of $f(x,t)$ at $t = 3.5$, and perform corresponding runs of the model.
- We can check the predictions made by these runs for $Y(t = 2)$.

Designing new experiment: 1D example



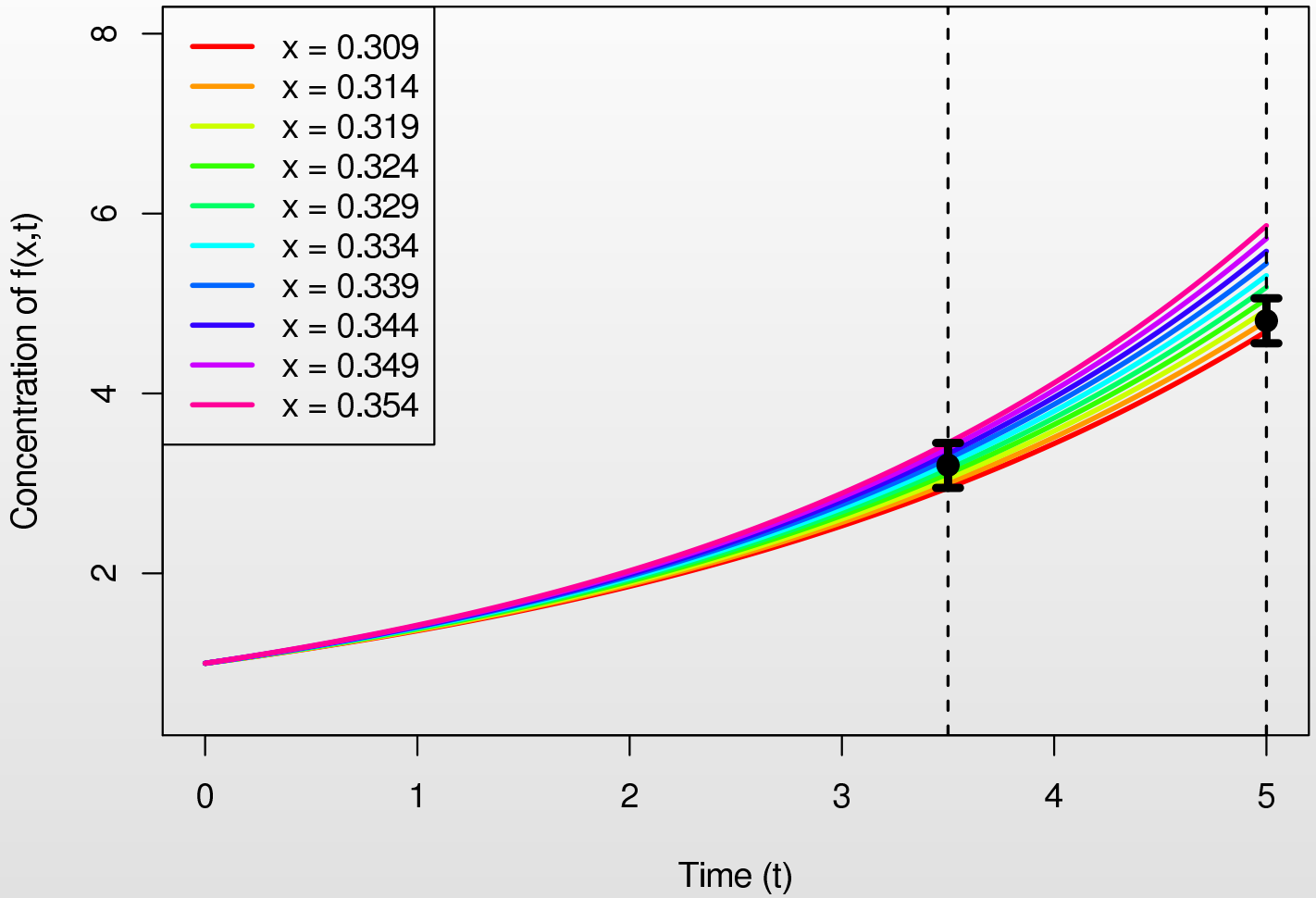
- The predictions imply that any measurement of $Y(t = 2)$ is highly unlikely to be informative for x.
- This is due to the measurement errors swamping the signal from the model output $Y(t = 2)$.

Designing new experiment: 1D example



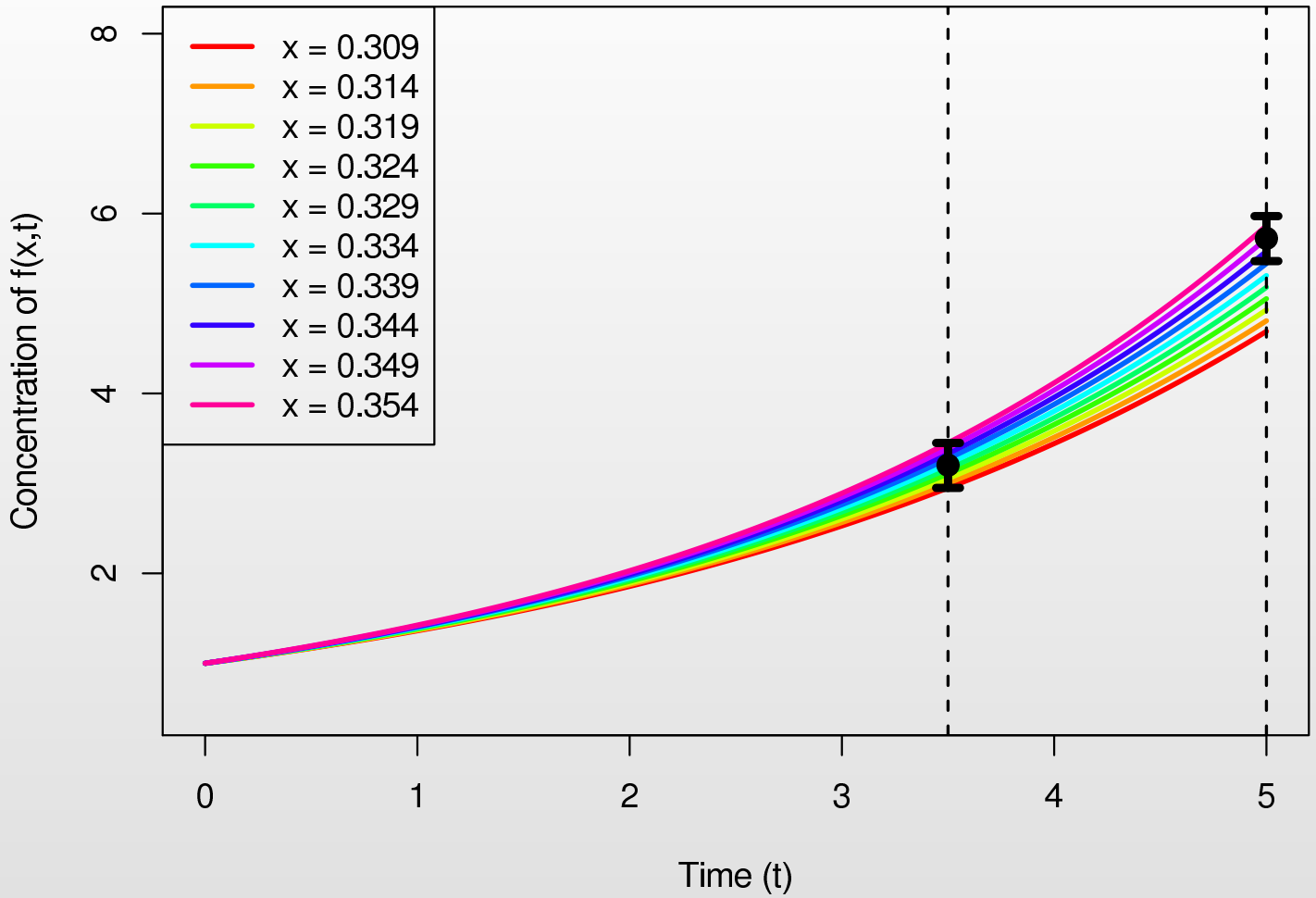
- The predictions for $Y(t = 5)$ show a different conclusion.

Designing new experiment: 1D example



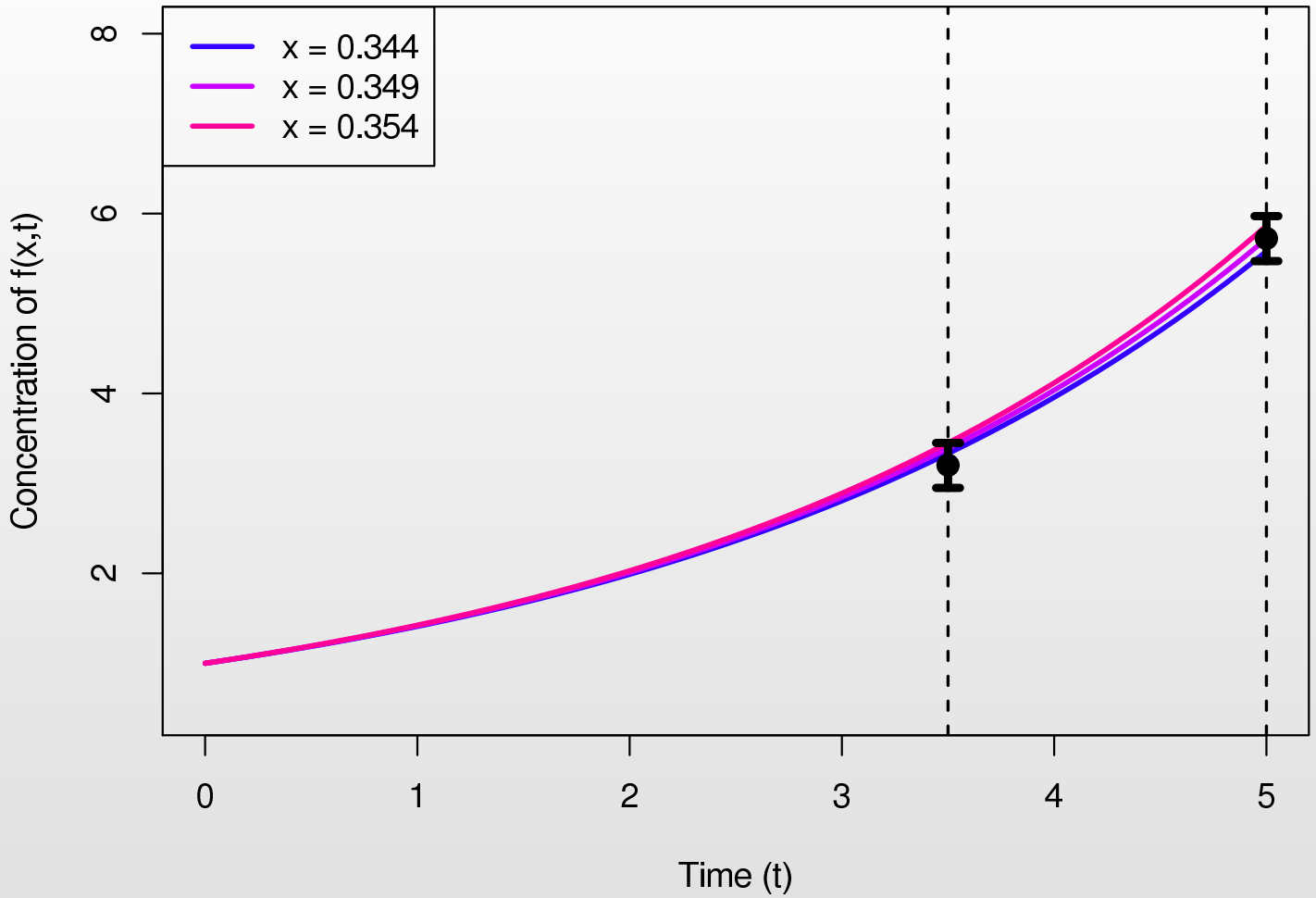
- The predictions for $Y(t = 5)$ show a different conclusion.

Designing new experiment: 1D example



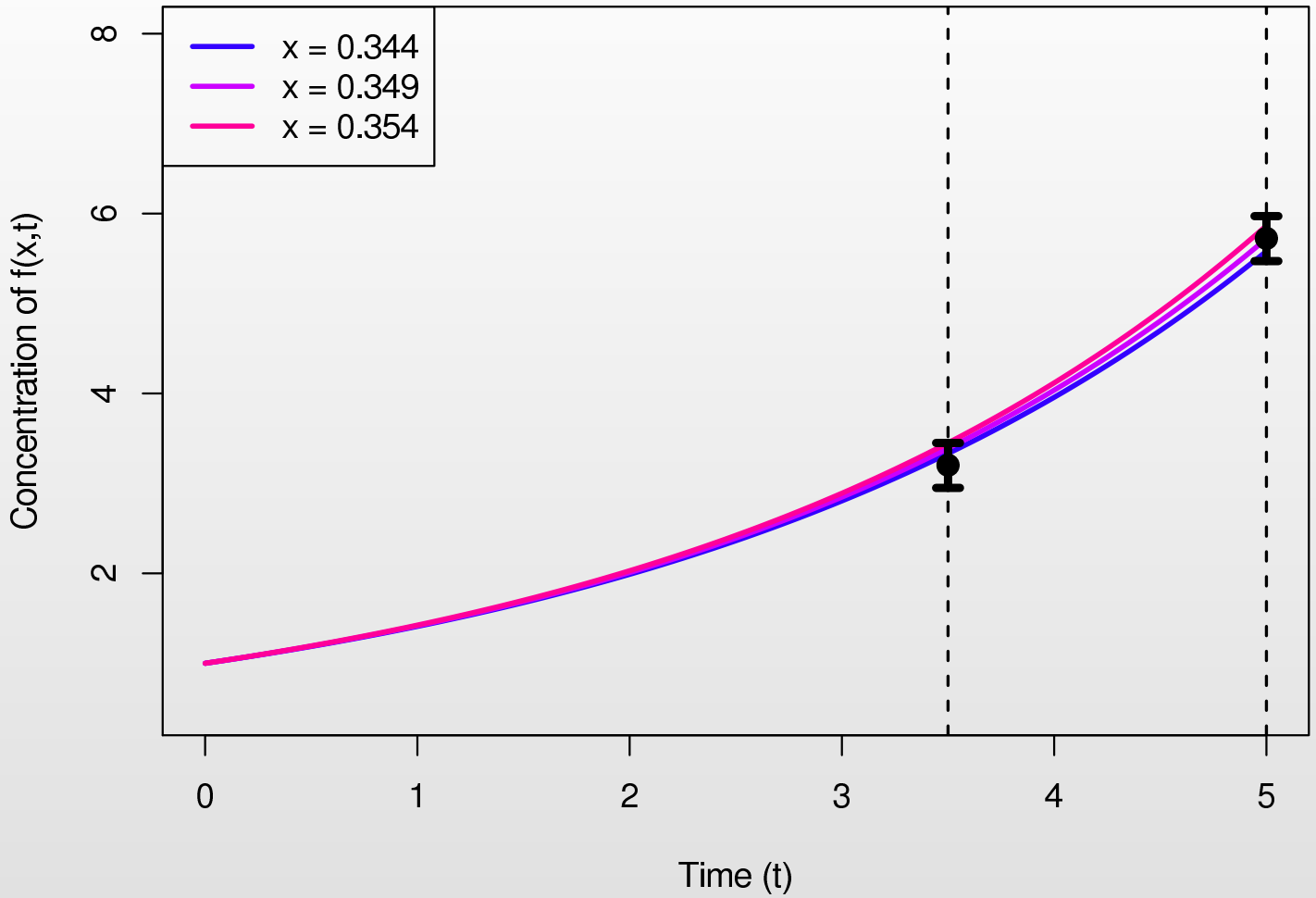
- The predictions for $Y(t = 5)$ show a different conclusion.
- For each possible measurement of $Y(t = 5)$ it is highly likely that we will be able to rule out several more values of x as implausible.

Designing new experiment: 1D example



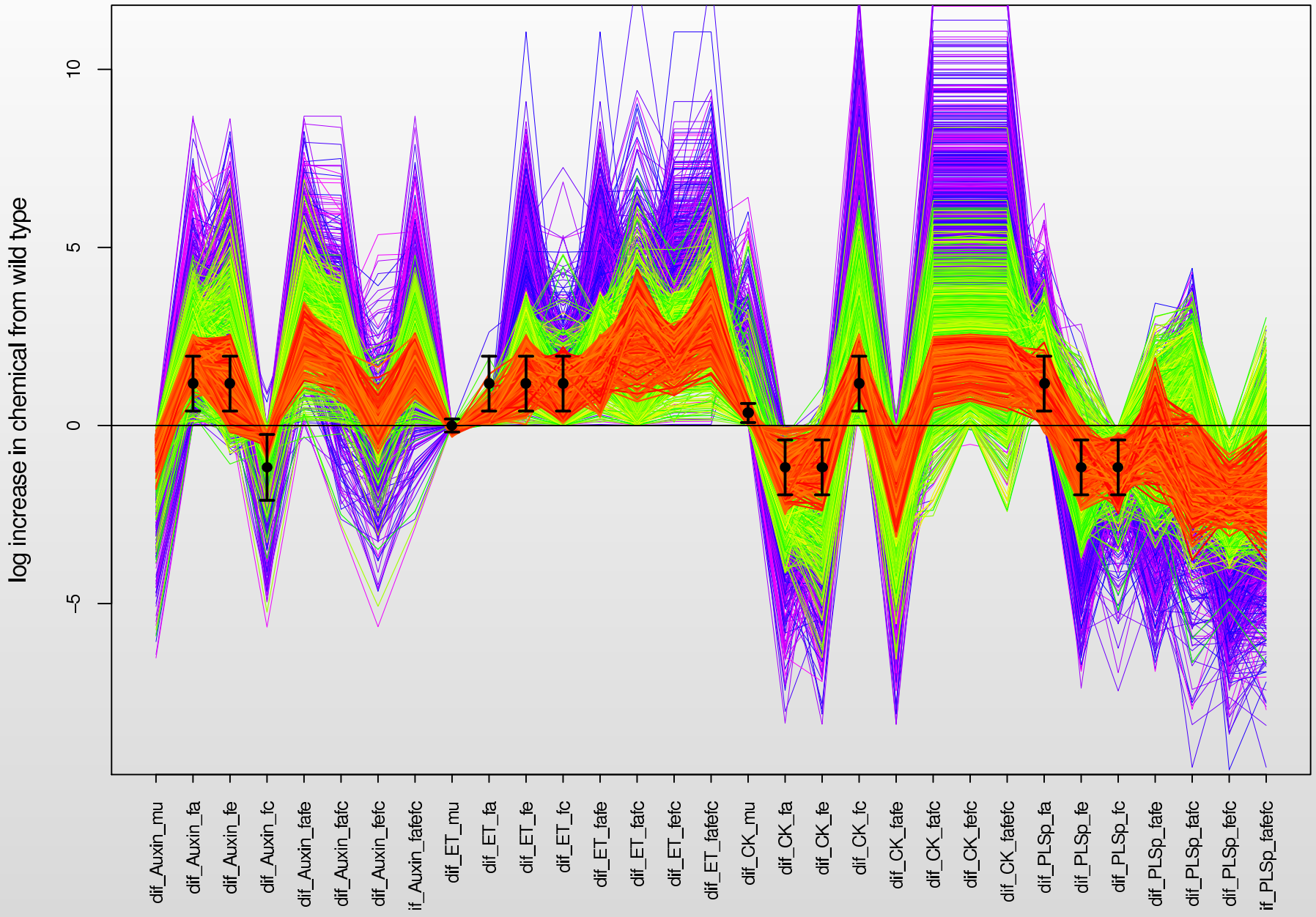
- For one possible measurement, see that non-implausible values of x would lie between 0.344 and 0.354, ruling out 70% of the possible values of x.

Designing new experiment: 1D example

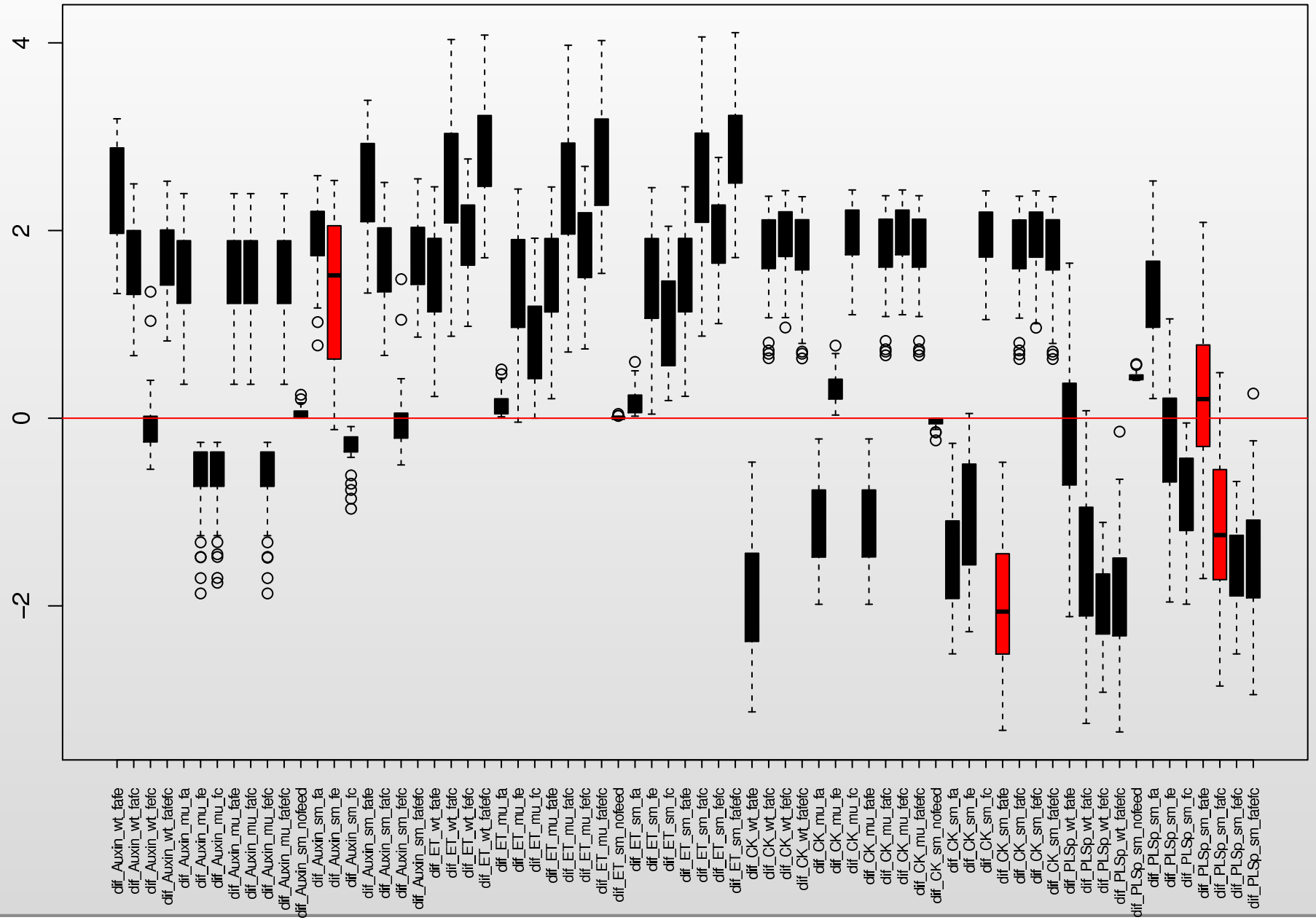


- For one possible measurement, see that non-implausible values of x would lie between 0.344 and 0.354, ruling out 70% of the possible values of x.
- This high expected space reduction in x implies that Experiment B, measuring $f(x,t)$ at $t = 5$, is clearly the best choice.

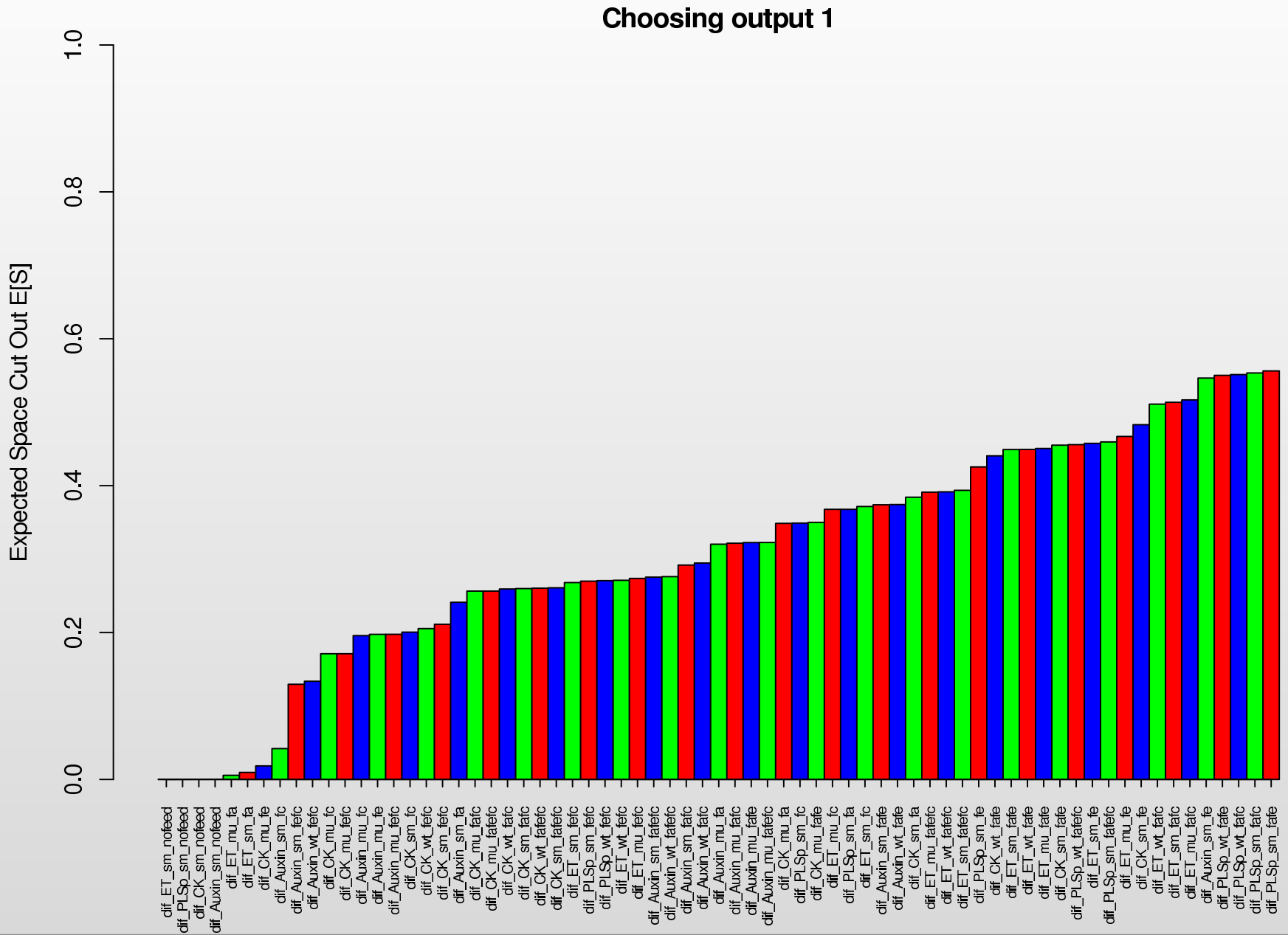
History Matching Plots Plus New Outputs



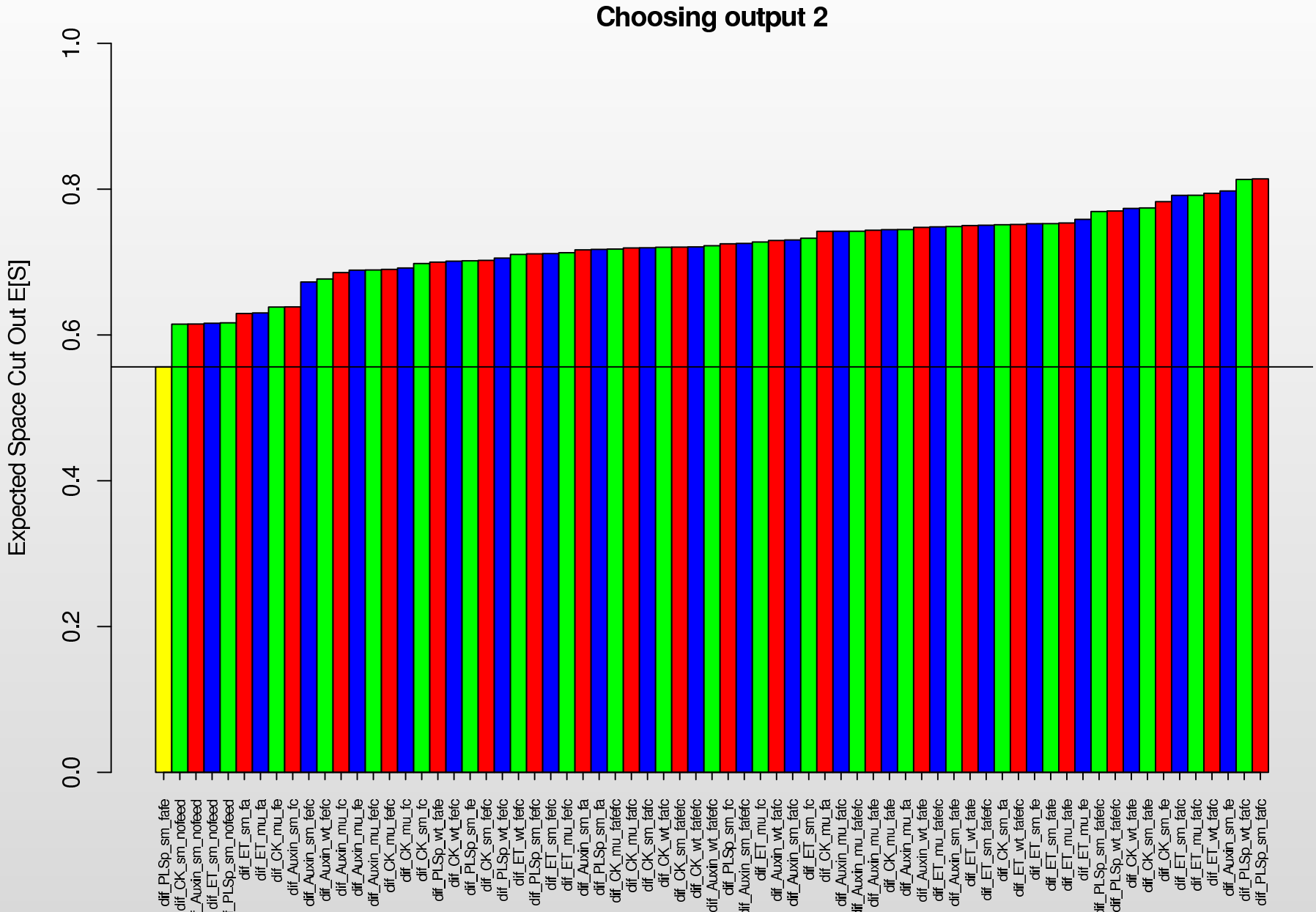
Predictions for New Outputs



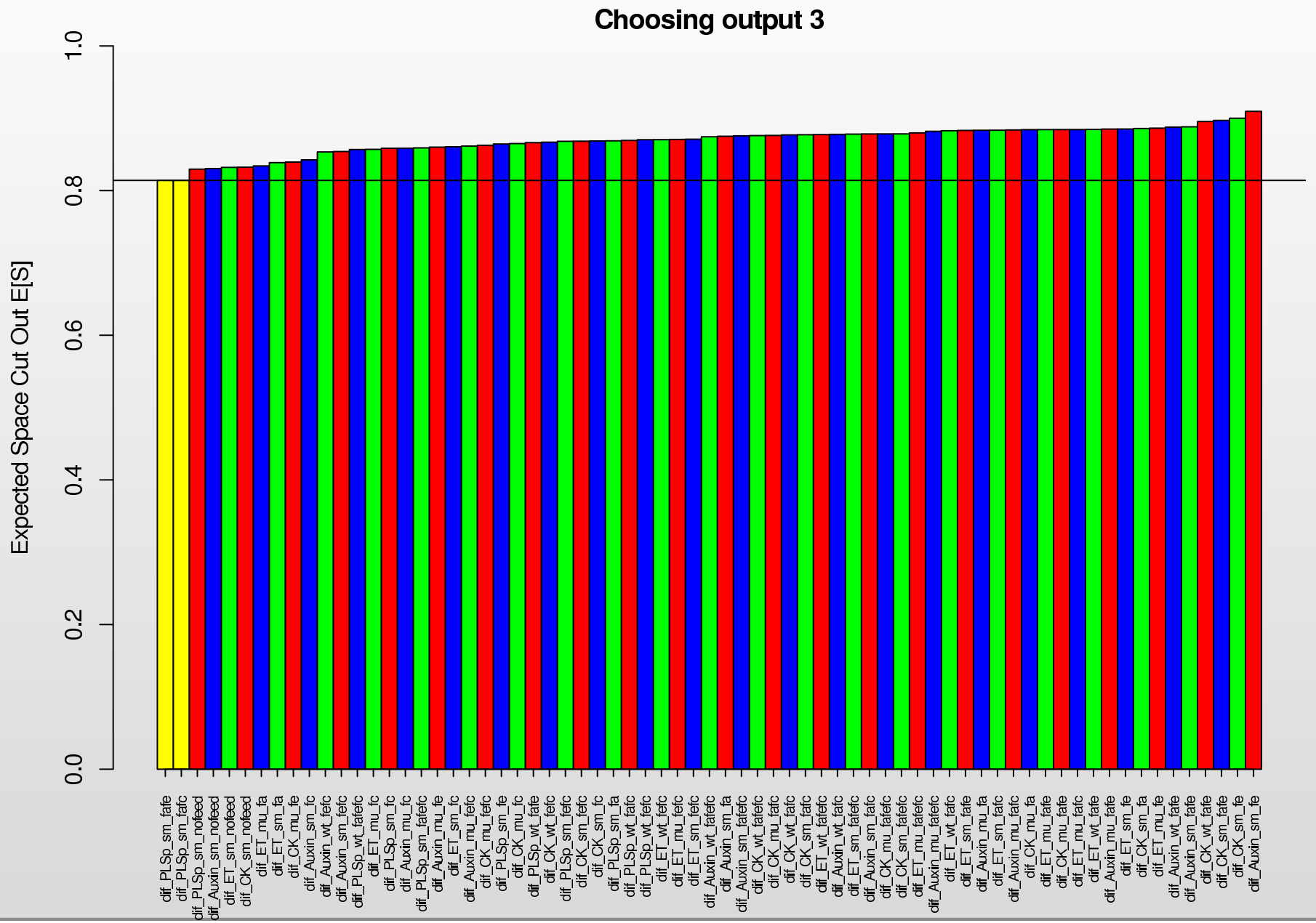
Space Cut Out Criteria for New Outputs



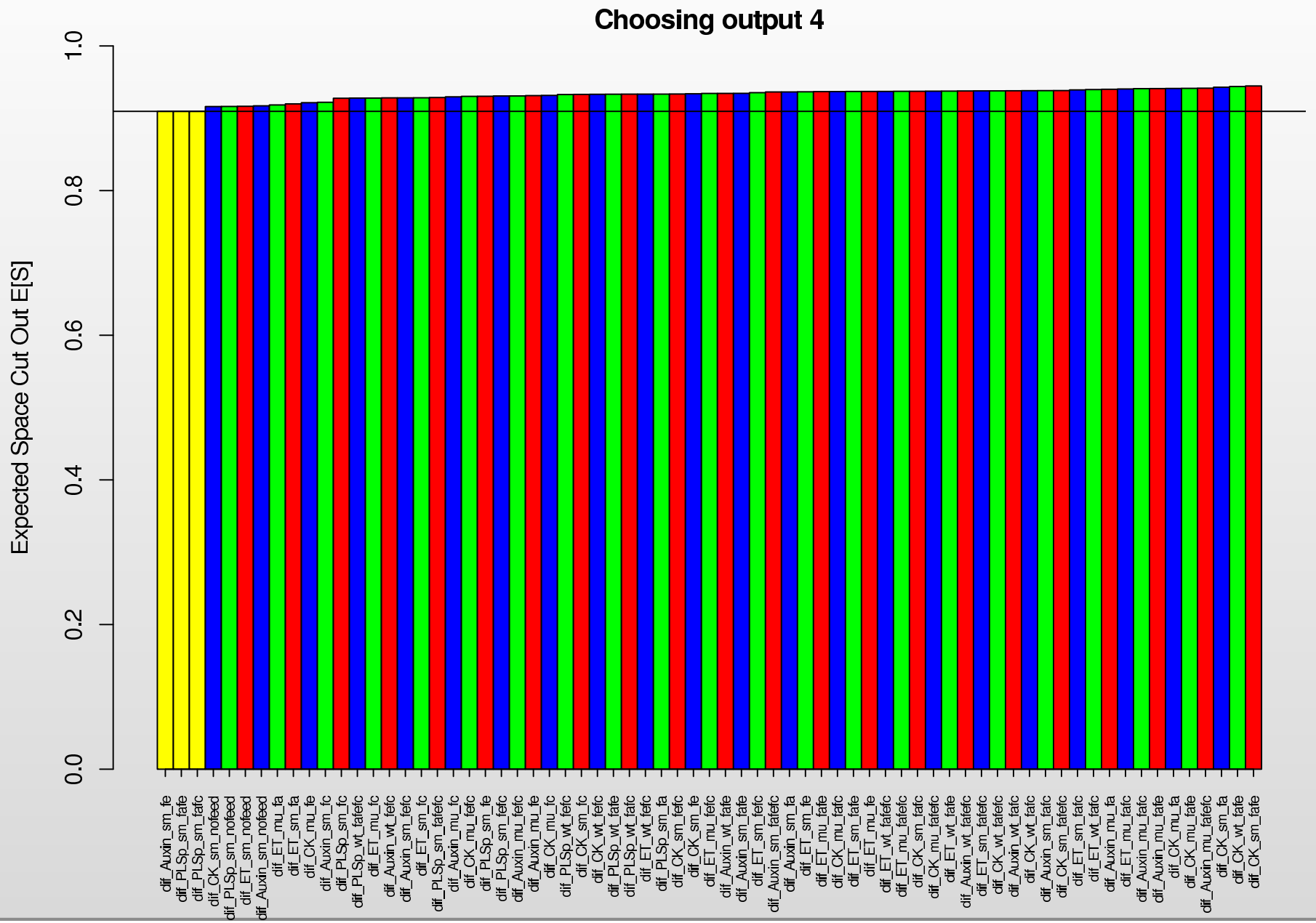
Space Cut Out Criteria for New Outputs



Space Cut Out Criteria for New Outputs



Space Cut Out Criteria for New Outputs

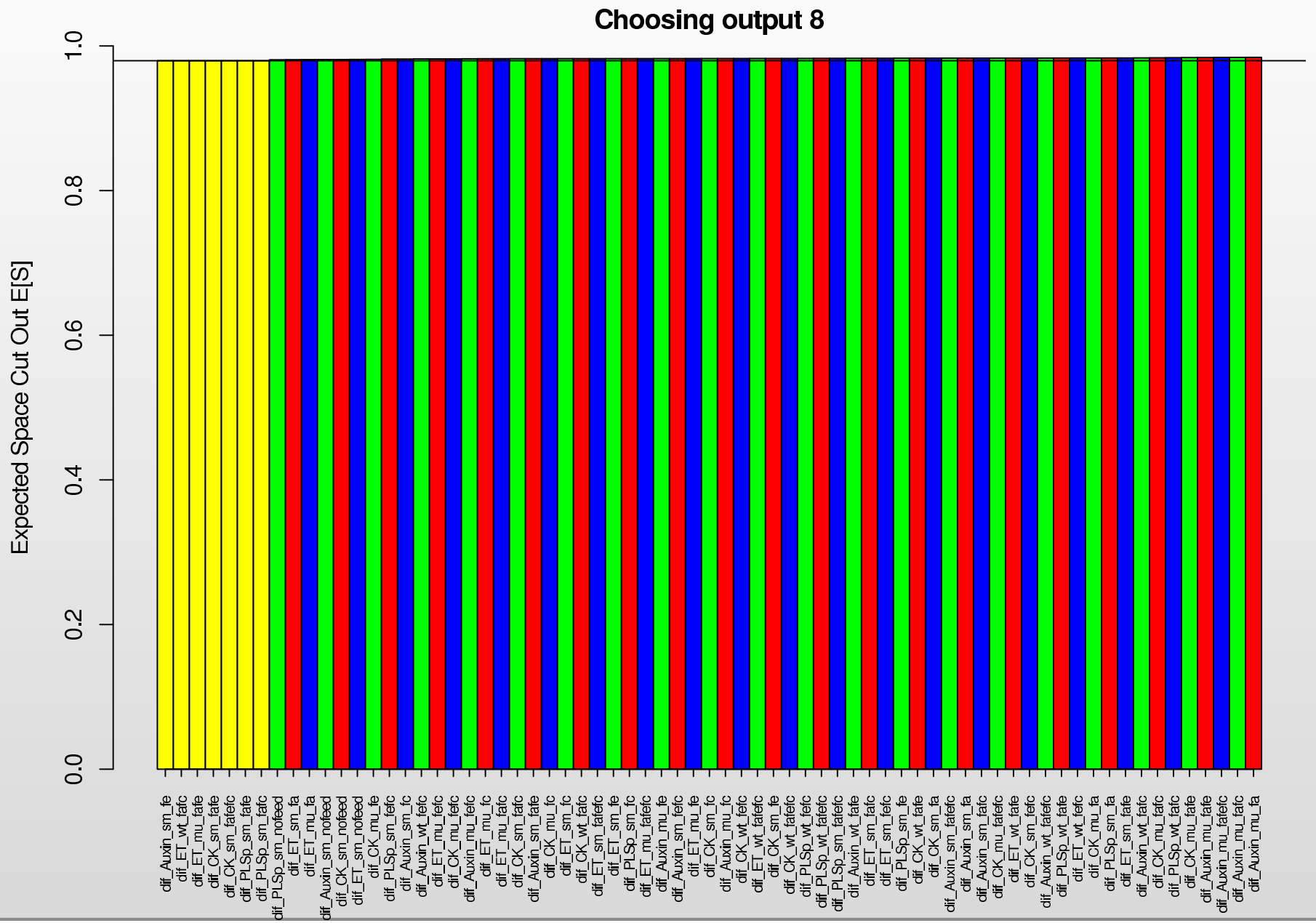


Space Cut Out Criteria for New Outputs

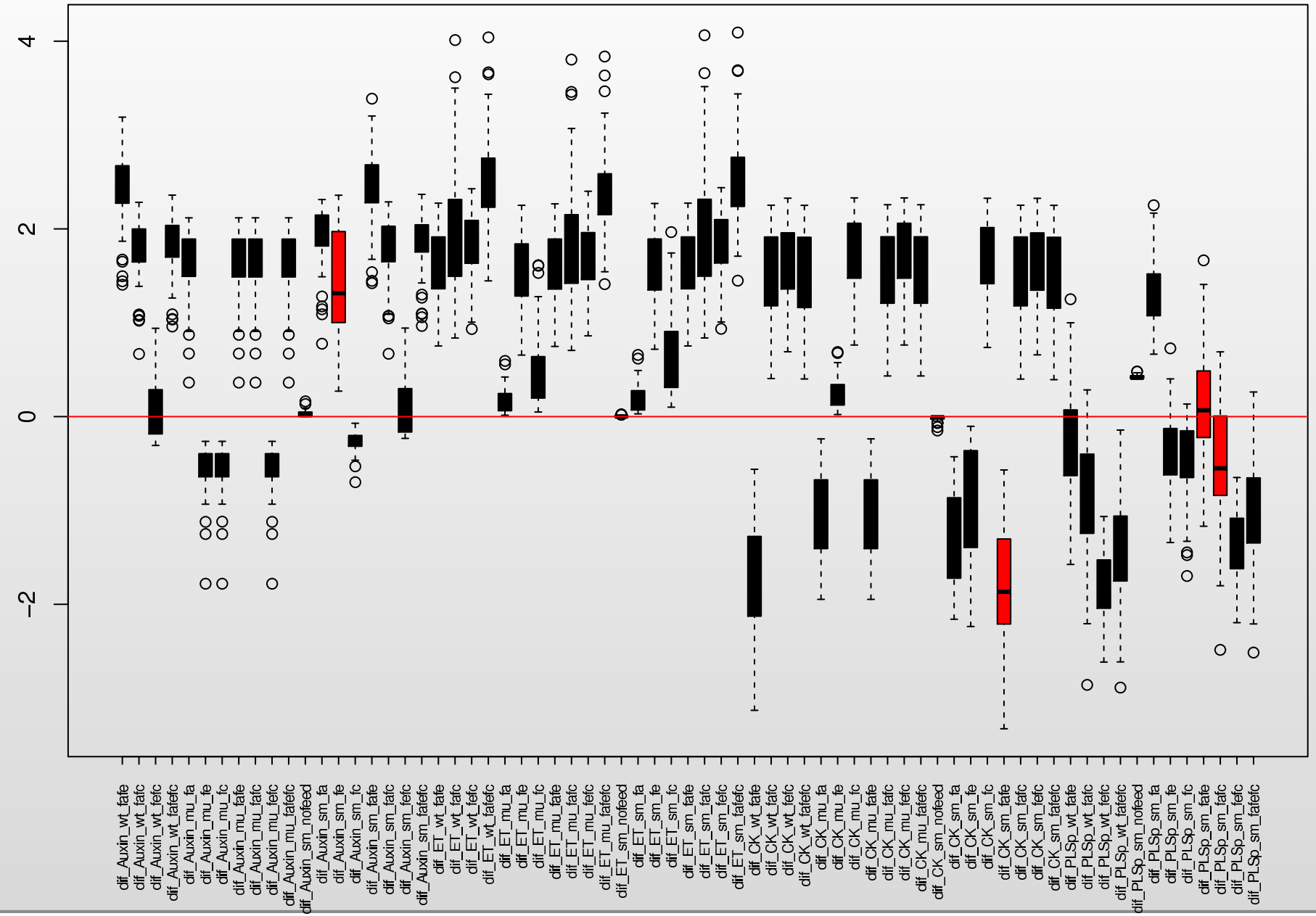
Space Cut Out Criteria for New Outputs

Space Cut Out Criteria for New Outputs

Space Cut Out Criteria for New Outputs



Predictions for New Outputs



Space Cut Out Criteria

- Consider the implausibility measure for a future measurement z_i :

$$I^2_{(i)}(x) = \frac{|\mathrm{E}(f_i(x)) - z_i|^2}{(\mathrm{Var}(f_i(x)) + \mathrm{Var}(\epsilon_i) + \mathrm{Var}(e_i))}$$

Space Cut Out Criteria

- Consider the implausibility measure for a future measurement z_i :

$$I^2_{(i)}(x) = \frac{|\mathrm{E}(f_i(x)) - z_i|^2}{(\mathrm{Var}(f_i(x)) + \mathrm{Var}(\epsilon_i) + \mathrm{Var}(e_i))}$$

- We will cut out x from further analysis if $I_{(i)}(x) > c_M$ as before, but now z_i is a random quantity.

Space Cut Out Criteria

- Consider the implausibility measure for a future measurement z_i :

$$I^2_{(i)}(x) = \frac{|\mathrm{E}(f_i(x)) - z_i|^2}{(\mathrm{Var}(f_i(x)) + \mathrm{Var}(\epsilon_i) + \mathrm{Var}(e_i))}$$

- We will cut out x from further analysis if $I_{(i)}(x) > c_M$ as before, but now z_i is a random quantity.
- Given z_i, define the indicator function $\mathrm{I}_i(x, z_i)$ s.t.

$$\mathrm{I}_i(x, z_i) = \begin{cases} 1 & \text{if} \quad I_{(i)}(x) > c_M, \quad x \text{ cut out} \\ 0 & \text{if} \quad I_{(i)}(x) < c_M, \quad x \text{ not cut out} \end{cases} \tag{1}$$

Space Cut Out Criteria

- Consider the implausibility measure for a future measurement z_i :

$$I^2_{(i)}(x) = \frac{|\mathrm{E}(f_i(x)) - z_i|^2}{(\mathrm{Var}(f_i(x)) + \mathrm{Var}(\epsilon_i) + \mathrm{Var}(e_i))}$$

- We will cut out x from further analysis if $I_{(i)}(x) > c_M$ as before, but now z_i is a random quantity.
- Given z_i, define the indicator function $\mathrm{I}_i(x, z_i)$ s.t.

$$\mathrm{I}_i(x, z_i) = \begin{cases} 1 & \text{if } I_{(i)}(x) > c_M, \quad x \text{ cut out} \\ 0 & \text{if } I_{(i)}(x) < c_M, \quad x \text{ not cut out} \end{cases} \tag{1}$$

- For given z_i, the fraction of space cutout S_i due to output i is:

$$S_i(z_i) = \frac{1}{V_{\mathcal{X}}} \int_{x \in \mathcal{X}} \mathrm{I}_i(x, z_i) dx$$

Space Cut Out Criteria

- Given the best input x^*, and distributional assumptions for z_i we have that:

$$z_i|x^* \quad \sim \quad N(\mu_i(x^*), \sigma_i^2(x^*) + \mathrm{Var}(\epsilon_i) + \mathrm{Var}(e_i))$$

with $\mu_i(x^*) = \mathrm{E}_D(f_i(x{=}x^*))$ and $\sigma_i(x^*) = \mathrm{Var}_D(f_i(x{=}x^*))$.

Space Cut Out Criteria

- Given the best input x^*, and distributional assumptions for z_i we have that:

$$z_i|x^* \quad \sim \quad N(\mu_i(x^*), \sigma_i^2(x^*) + \mathrm{Var}(\epsilon_i) + \mathrm{Var}(e_i))$$

with $\mu_i(x^*) = \mathrm{E}_D(f_i(x{=}x^*))$ and $\sigma_i(x^*) = \mathrm{Var}_D(f_i(x{=}x^*))$.

- Therefore the expected space cut out S_i given x^* is then

$$\mathrm{E}(S_i|x^*) = \frac{1}{V_{\mathcal{X}}} \int_{z_i} \int_{x \in \mathcal{X}} \mathrm{I}_i(x, z_i) \pi(z_i|x^*) dx dz_i$$

Space Cut Out Criteria

- Given the best input x^*, and distributional assumptions for z_i we have that:

$$z_i|x^* \quad \sim \quad N(\mu_i(x^*), \sigma_i^2(x^*) + \mathrm{Var}(\epsilon_i) + \mathrm{Var}(e_i))$$

with $\mu_i(x^*) = \mathrm{E}_D(f_i(x{=}x^*))$ and $\sigma_i(x^*) = \mathrm{Var}_D(f_i(x{=}x^*))$.

- Therefore the expected space cut out S_i given x^* is then

$$\mathrm{E}(S_i|x^*) = \frac{1}{V_{\mathcal{X}}} \int_{z_i} \int_{x \in \mathcal{X}} \mathrm{I}_i(x, z_i) \pi(z_i|x^*) dx dz_i$$

- and the expected space cut out S_i for new output i is

$$\mathrm{E}(S_i) = \frac{1}{V_{\mathcal{X}}^2} \int_{x^* \in \mathcal{X}} \int_{z_i} \int_{x \in \mathcal{X}} \mathrm{I}_i(x, z_i) \pi(z_i|x^*) dx dz_i dx^*$$

Space Cut Out Criteria

- Given the best input x^*, and distributional assumptions for z_i we have that:

$$z_i|x^* \sim N(\mu_i(x^*), \sigma_i^2(x^*) + \mathrm{Var}(\epsilon_i) + \mathrm{Var}(e_i))$$

with $\mu_i(x^*) = \mathrm{E}_D(f_i(x{=}x^*))$ and $\sigma_i(x^*) = \mathrm{Var}_D(f_i(x{=}x^*))$.

- Therefore the expected space cut out S_i given x^* is then

$$\mathrm{E}(S_i|x^*) = \frac{1}{V_{\mathcal{X}}} \int_{z_i} \int_{x \in \mathcal{X}} \mathrm{I}_i(x, z_i) \pi(z_i|x^*) dx dz_i$$

- and the expected space cut out S_i for new output i is

$$\mathrm{E}(S_i) = \frac{1}{V_{\mathcal{X}}^2} \int_{x^* \in \mathcal{X}} \int_{z_i} \int_{x \in \mathcal{X}} \mathrm{I}_i(x, z_i) \pi(z_i|x^*) dx dz_i dx^*$$

- If we want to identify our Utility U_i with the space cutout S_i, we will then choose the experiment to measure output i to maximise $\mathrm{E}(S_i)$.

Space Cut Out Criteria

- Given the best input x^*, and distributional assumptions for z_i we have that:

$$z_i|x^* \quad \sim \quad N(\mu_i(x^*), \sigma_i^2(x^*) + \mathrm{Var}(\epsilon_i) + \mathrm{Var}(e_i))$$

with $\mu_i(x^*) = \mathrm{E}_D(f_i(x{=}x^*))$ and $\sigma_i(x^*) = \mathrm{Var}_D(f_i(x{=}x^*))$.

- Therefore the expected space cut out S_i given x^* is then

$$\mathrm{E}(S_i|x^*) = \frac{1}{V_{\mathcal{X}}} \int_{z_i} \int_{x \in \mathcal{X}} \mathrm{I}_i(x, z_i) \pi(z_i|x^*) dx dz_i$$

- and the expected space cut out S_i for new output i is

$$\mathrm{E}(S_i) = \frac{1}{V_{\mathcal{X}}^2} \int_{x^* \in \mathcal{X}} \int_{z_i} \int_{x \in \mathcal{X}} \mathrm{I}_i(x, z_i) \pi(z_i|x^*) dx dz_i dx^*$$

- If we want to identify our Utility U_i with the space cutout S_i, we will then choose the experiment to measure output i to maximise $\mathrm{E}(S_i)$.
- In fact we want to choose 4 outputs i, j, k, l such that the analogous expected space cut out $\mathrm{E}(S_{i,j,k,l})$ is maximised.

Approximate Space Cut Out Criteria

- Integrals are expensive so we use the set of n_a acceptable runs x_j, $j = 1, .., n_a$ where $x_j \in \mathcal{X}$ to approximate the integrals.

Approximate Space Cut Out Criteria

- Integrals are expensive so we use the set of n_a acceptable runs x_j, $j = 1, .., n_a$ where $x_j \in \mathcal{X}$ to approximate the integrals.
- In which case $\mathrm{E}(S_i)$ becomes

$$\mathrm{E}(S_i) \approx \frac{1}{n_a^2 n_{sim}} \sum_{k=1}^{n_a} \sum_{a=1}^{n_{sim}} \sum_{j=1}^{n_a} \mathrm{I}_i(x_j, z_i^a)$$

- where we approximate the z_i integral by simulating n_{sim} draws of z_i from $\pi(z_i|x_k^*)$ for each x_k^*. Can do analytically in some cases.

Approximate Space Cut Out Criteria

- Integrals are expensive so we use the set of n_a acceptable runs x_j, $j = 1, .., n_a$ where $x_j \in \mathcal{X}$ to approximate the integrals.
- In which case $\mathrm{E}(S_i)$ becomes

$$\mathrm{E}(S_i) \approx \frac{1}{n_a^2 n_{sim}} \sum_{k=1}^{n_a} \sum_{a=1}^{n_{sim}} \sum_{j=1}^{n_a} \mathrm{I}_i(x_j, z_i^a)$$

- where we approximate the z_i integral by simulating n_{sim} draws of z_i from $\pi(z_i|x_k^*)$ for each x_k^*. Can do analytically in some cases.
- Should really do this using emulators, but for this calculation the runs may be sufficient.

Approximate Space Cut Out Criteria

- Integrals are expensive so we use the set of n_a acceptable runs x_j, $j = 1, .., n_a$ where $x_j \in \mathcal{X}$ to approximate the integrals.
- In which case $\mathrm{E}(S_i)$ becomes

$$\mathrm{E}(S_i) \approx \frac{1}{n_a^2 n_{sim}} \sum_{k=1}^{n_a} \sum_{a=1}^{n_{sim}} \sum_{j=1}^{n_a} \mathrm{I}_i(x_j, z_i^a)$$

- where we approximate the z_i integral by simulating n_{sim} draws of z_i from $\pi(z_i|x_k^*)$ for each x_k^*. Can do analytically in some cases.
- Should really do this using emulators, but for this calculation the runs may be sufficient.
- This is because the runs would inform the most important parts of the integrals.

Approximate Space Cut Out Criteria

- Integrals are expensive so we use the set of n_a acceptable runs x_j, $j = 1, .., n_a$ where $x_j \in \mathcal{X}$ to approximate the integrals.
- In which case $\mathrm{E}(S_i)$ becomes

$$\mathrm{E}(S_i) \approx \frac{1}{n_a^2 n_{sim}} \sum_{k=1}^{n_a} \sum_{a=1}^{n_{sim}} \sum_{j=1}^{n_a} \mathrm{I}_i(x_j, z_i^a)$$

- where we approximate the z_i integral by simulating n_{sim} draws of z_i from $\pi(z_i|x_k^*)$ for each x_k^*. Can do analytically in some cases.
- Should really do this using emulators, but for this calculation the runs may be sufficient.
- This is because the runs would inform the most important parts of the integrals.
- Again, we are interested in the analogous multivariate quantity $\mathrm{E}(S_{i,j,k,l})$

Experimental Design Results: 4 new experiments chosen

- Selected outputs by stepping up to 8 outputs, then back down to 4: robust.

Experimental Design Results: 4 new experiments chosen

- Selected outputs by stepping up to 8 outputs, then back down to 4: robust.
- Sensitivity analysis: performed two calculations with high/low model discrepancy and observed errors: same choice of outputs in both cases.

Experimental Design Results: 4 new experiments chosen

- Selected outputs by stepping up to 8 outputs, then back down to 4: robust.
- Sensitivity analysis: performed two calculations with high/low model discrepancy and observed errors: same choice of outputs in both cases.
- The four most informative experiments chosen to maximise $\mathrm{E}(S_{i,j,k,l})$:

plant	chemical measured	feeding regime	expected space cut
PSLox	PLSp	auxin + ethylene	56%
PSLox	PLSp	auxin + cytokinin	82%
PSLox	Auxin	ethylene	91%
PSLox	Cytokinin	auxin + ethylene	94%

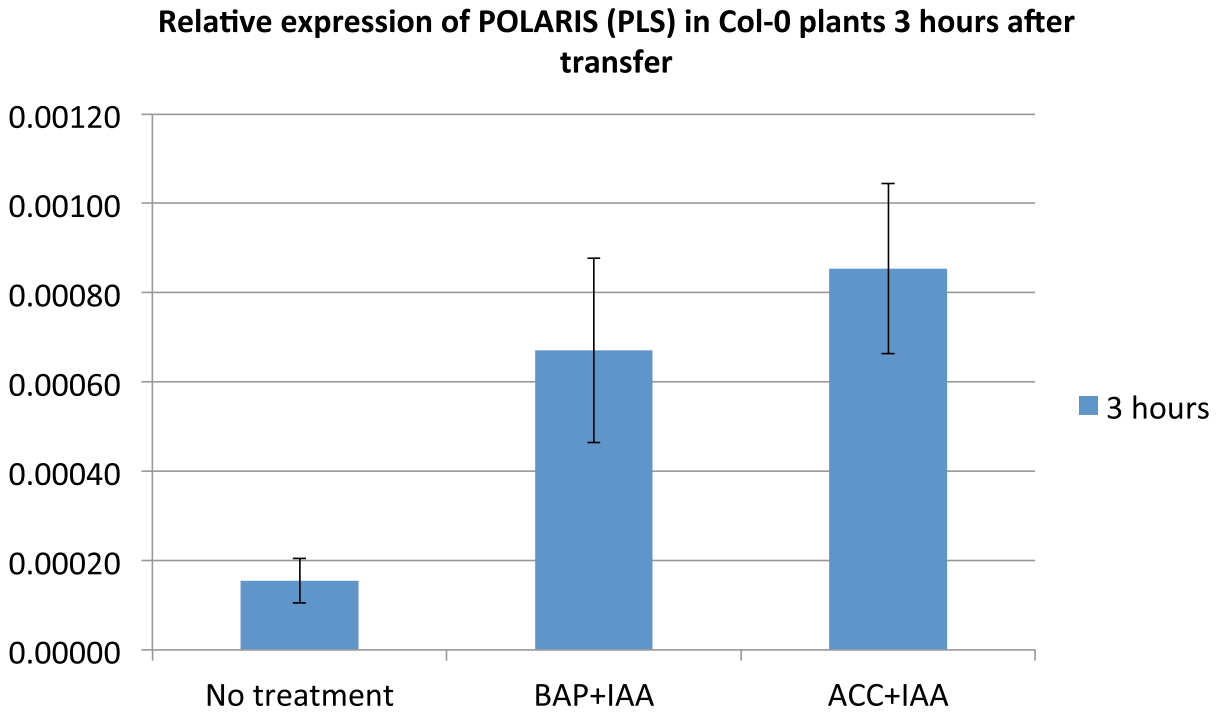
Experimental Design Results: 4 new experiments chosen

- Selected outputs by stepping up to 8 outputs, then back down to 4: robust.
- Sensitivity analysis: performed two calculations with high/low model discrepancy and observed errors: same choice of outputs in both cases.
- The four most informative experiments chosen to maximise $\mathrm{E}(S_{i,j,k,l})$:

plant	chemical measured	feeding regime	expected space cut
PSLox	PLSp	auxin + ethylene	56%
PSLox	PLSp	auxin + cytokinin	82%
PSLox	Auxin	ethylene	91%
PSLox	Cytokinin	auxin + ethylene	94%

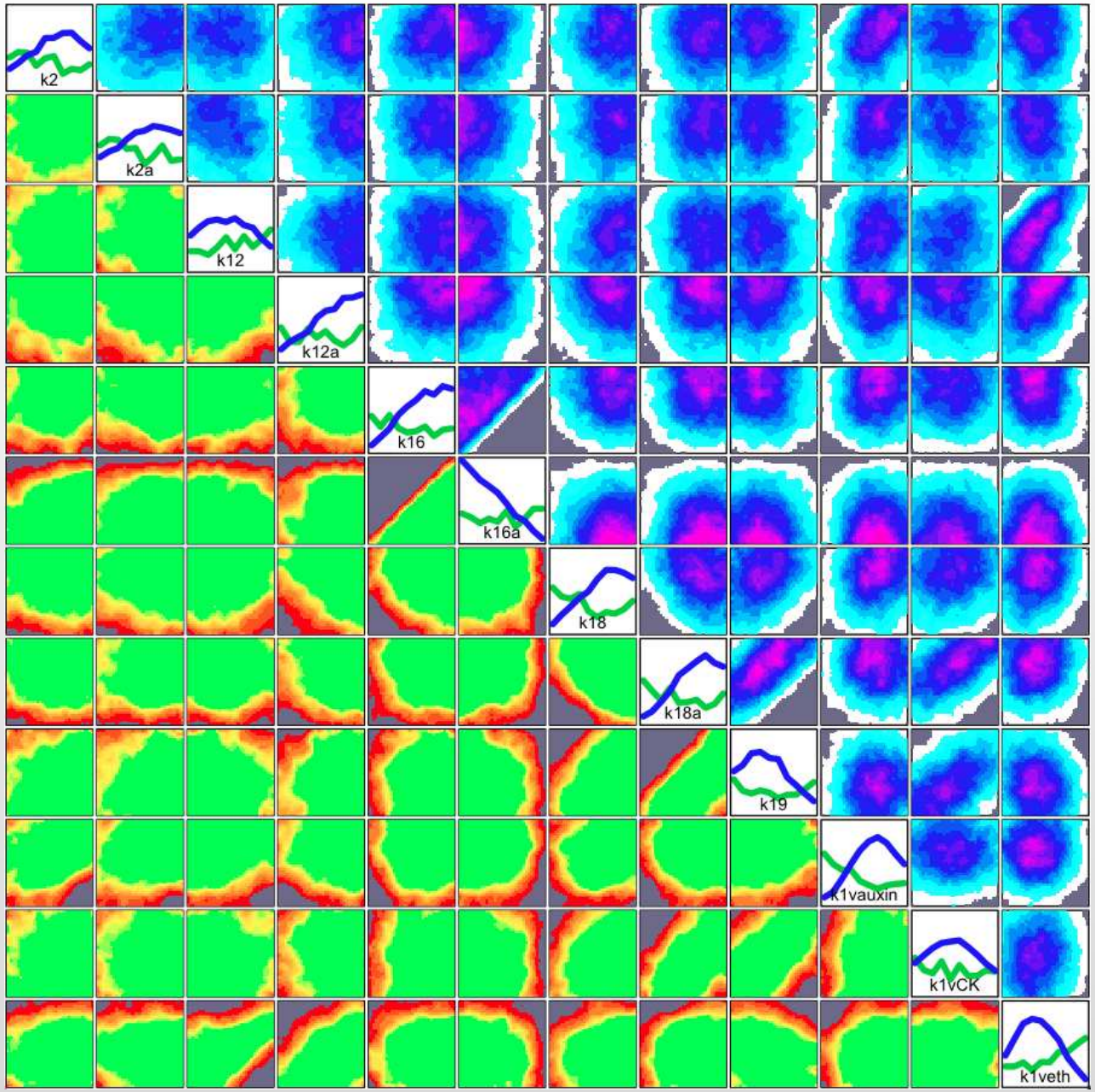
- First two experiments completed.

Results for First Two New Experiments

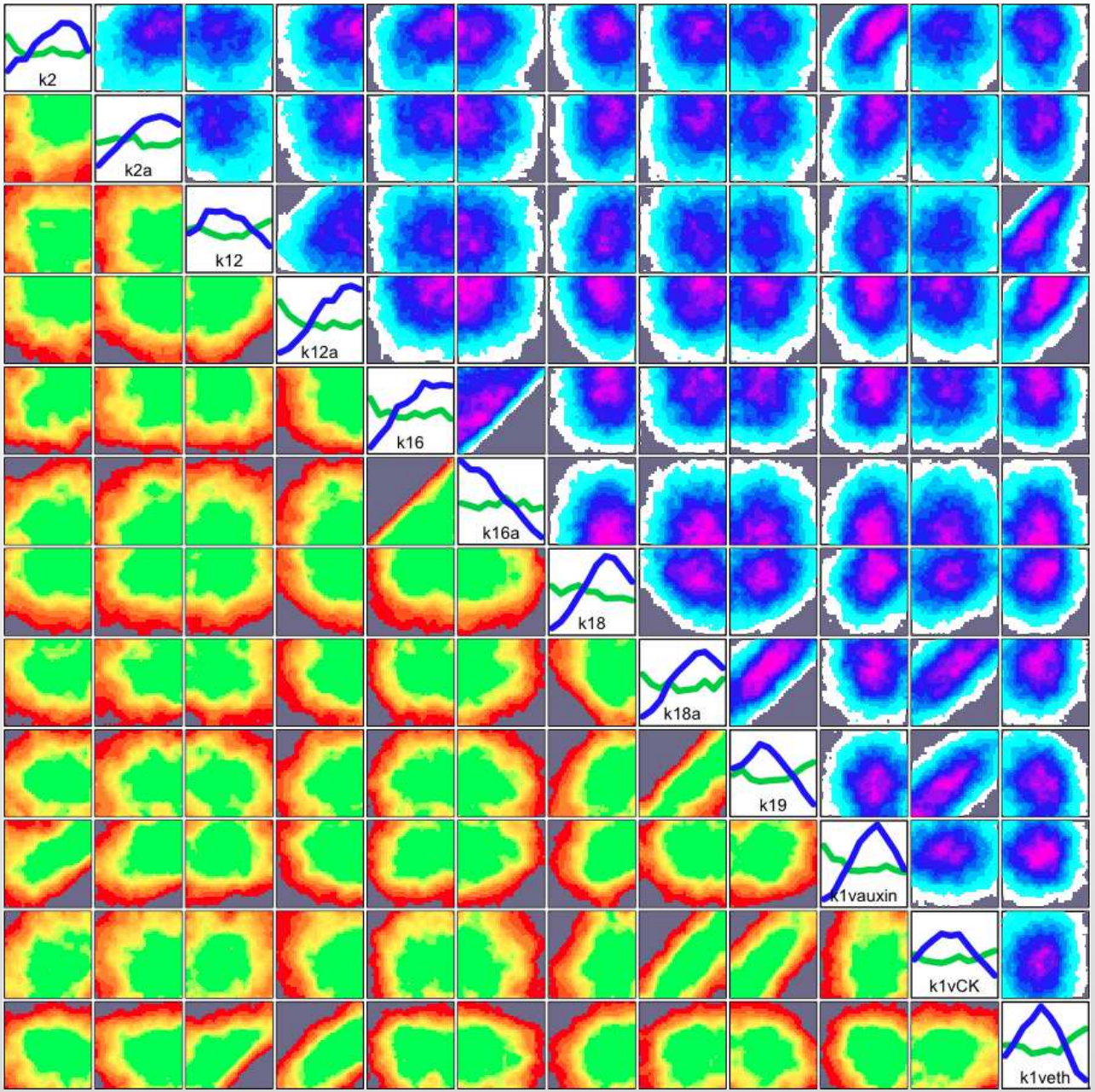


Seven day old Columbia wildtype plants were transferred to media containing either cytokinin and auxin (BAP + IAA), an ethylene precursor and auxin (ACC + IAA) or no hormone treatment. After three hours, the relative abundance (expression) of the POLARIS mRNA was measured with qPCR. Three separate biological replicates were used and error bars represent the standard error of the mean.

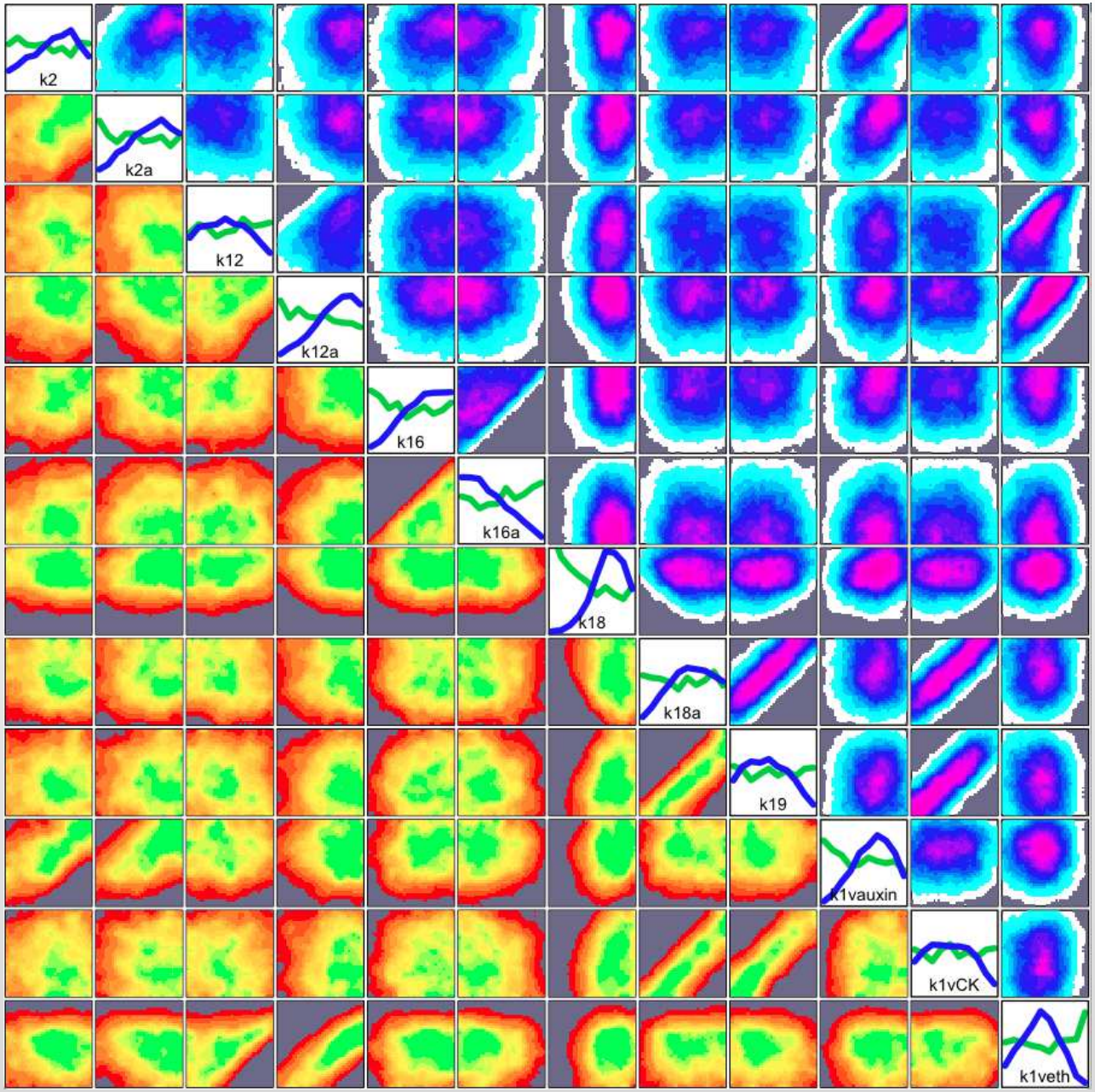
Iterative Input Space Reduction: Arabidopsis Model Wave 1



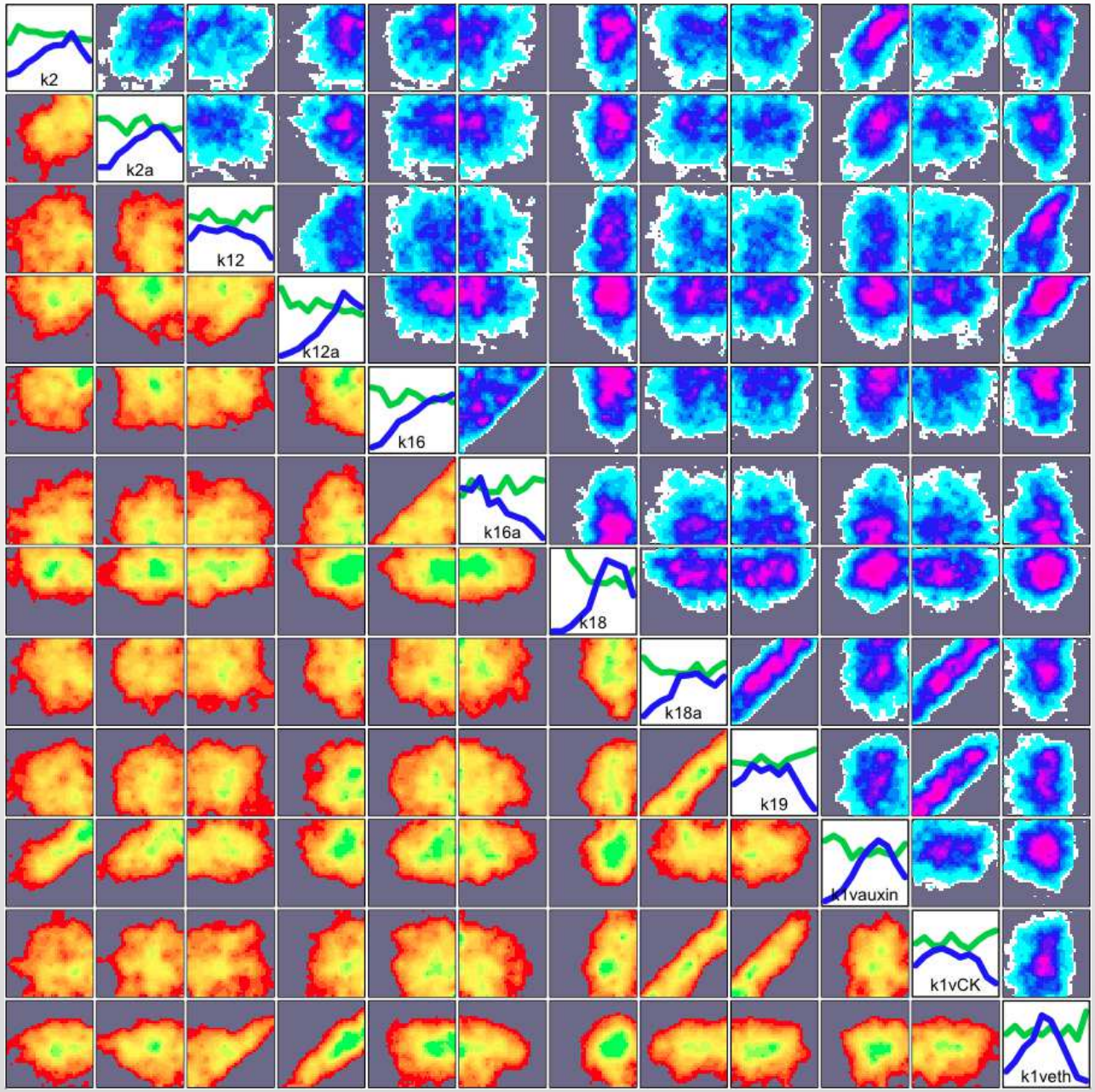
Iterative Input Space Reduction: Arabidopsis Model Wave 2



Iterative Input Space Reduction: Arabidopsis Model Wave 3



Arabidopsis Model with 2 New Results



Concluding Comments

- Large reduction in input space due to just 2 new experiments.

Concluding Comments

- Large reduction in input space due to just 2 new experiments.
- All these calculations are designed to be efficient: approximations used are very beneficial.

Concluding Comments

- Large reduction in input space due to just 2 new experiments.
- All these calculations are designed to be efficient: approximations used are very beneficial.
- We chose a good set of new experiments, not necessarily the theoretical best (which we wouldn't believe anyway).

Concluding Comments

- Large reduction in input space due to just 2 new experiments.
- All these calculations are designed to be efficient: approximations used are very beneficial.
- We chose a good set of new experiments, not necessarily the theoretical best (which we wouldn't believe anyway).
- We have chosen experiments to learn about all input or rate parameters using the expected space reduction criteria: we could have chosen to learn about specific rate parameters of interest.

Concluding Comments

- Large reduction in input space due to just 2 new experiments.
- All these calculations are designed to be efficient: approximations used are very beneficial.
- We chose a good set of new experiments, not necessarily the theoretical best (which we wouldn't believe anyway).
- We have chosen experiments to learn about all input or rate parameters using the expected space reduction criteria: we could have chosen to learn about specific rate parameters of interest.
- We can also design experiments to challenge the model, i.e. to validate it if necessary.

Final Concluding Comments

- We have a broad methodology for performing full uncertainty analyses on such complex models of biological systems.

Final Concluding Comments

- We have a broad methodology for performing full uncertainty analyses on such complex models of biological systems.

- The correct treatment of uncertainty is vital: without this, any analysis will be problematic and untrustworthy.

Final Concluding Comments

- We have a broad methodology for performing full uncertainty analyses on such complex models of biological systems.
- The correct treatment of uncertainty is vital: without this, any analysis will be problematic and untrustworthy.
- The emulation methods we describe can be used to exhaustively explore model features (helpful when developing models).

Final Concluding Comments

- We have a broad methodology for performing full uncertainty analyses on such complex models of biological systems.
- The correct treatment of uncertainty is vital: without this, any analysis will be problematic and untrustworthy.
- The emulation methods we describe can be used to exhaustively explore model features (helpful when developing models).
- Due to the need to synthesis many sources of uncertainty within one coherent calculation, a Bayesian approach is ideal.

References

History Matching on Galaxy simulation papers:

Vernon, I., Goldstein, M., Bower, R. G., Galaxy Formation: "Bayesian History Matching for the Observable Universe". *Statistical Science* 29 (2014), no. 1, 81–90.

Vernon, I., Goldstein, M., and Bower, R. (2010), "*Galaxy Formation: a Bayesian Uncertainty Analysis*", *Bayesian Analysis*, 5(4): 619–670. Invited discussion paper. MUCM Technical Report 10/03. Awarded 2010 Mitchell Prize.

Bower, R., Vernon, I., Goldstein, M., et al. (2010), "*The Parameter Space of Galaxy Formation*", *Mon.Not.Roy.Astron.Soc.*, 407: 2017–2045. MUCM Technical Report 10/02.

The Bayes Linear Book:

Goldstein, M., and Wooff, D. A. (2007) *"Bayes Linear Statistics: Theory and Methods"*, Wiley.

Arabidopsis Model:

Liu, J., Mehdi, S., Topping, J., Tarkowski, P., and Lindsey, K. (2010), Modelling and experimental analysis of hormonal crosstalk in Arab., *Mol Syst Biol*, 6.

Developments in Emulation

- Much work in the emulation of deterministic models (much to be done too!). Some decent papers are (see Managing Uncertainty in Complex Models website http://www.mucm.ac.uk):

Developments in Emulation

- Much work in the emulation of deterministic models (much to be done too!). Some decent papers are (see Managing Uncertainty in Complex Models website http://www.mucm.ac.uk):

History Matching and use of fast approximate models:

P.S. Craig, M. Goldstein, A.H. Seheult, J.A. Smith (1997). Pressure matching for hydocarbon reservoirs: a case study in the use of Bayes linear strategies for large computer experiments (with discussion), in Case Studies in Bayesian Statistics, vol. III, eds. C. Gastonis et al. 37-93. Springer-Verlag.

Probabilistic Calibration:

Kennedy, M.C. and O'Hagan, A. (2001). Bayesian calibration of computer models (with discussion). Journal of the Royal Statistical Society B 63, 425-464

Classic book on design:

Santner, T., Williams, B. and Notz, W. (2003). The Design and Analysis of Computer Experiments. Springer Verlag: New York.

Developments in Emulation

A good introductory tutorial:

OHagan, A. (2006). Bayesian analysis of computer code outputs: A tutorial. Reliability Engineering and System Safety 91 1290-1300

More advanced model discrepancy:

M. Goldstein and J.C.Rougier (2008). Reified Bayesian modelling and inference for physical systems (with discussion), JSPI.

Sensitivity Analysis:

Oakley, J. E. and O'Hagan, A, Probabilistic sensitivity analysis of complex models: a Bayesian approach, J. R. Statist. Soc. B (2004) 66, Part 3, pp.751-769

Fast Multivariate Emulators:

Rougier, J. 2008. Efficient Emulators for Multivariate Deterministic Functions. Journal of Computational and Graphical Statistics 17:4, 827-843.

Developments in Emulation

Example of dimension reduction on the outputs:

Higdon, D., Gattiker, J., Williams, B. and Rightley, M., Computer Model Calibration Using High-Dimensional Output, JASA (2008), Vol. 103, No. 482,

Dimension reduction on the outputs using Principal Variables:

Cumming, J., A. and Goldstein, M., Bayes linear uncertainty analysis for oil reservoirs based on multiscale computer experiments, (2009), Handbook of Bayesian Analysis, eds A O'Hagan and M West, Oxford University Press.

Dynamic Emulation:

Contia, S. and O'Hagan, A., Bayesian emulation of complex multi-output and dynamic computer models, Journal of Statistical Planning and Inference 140 (2010) 640 – 651

Assessing internal model discrepancy by adding noise to model:

Goldstein, M., Seheult, A. and Vernon, I. (2012). Assessing Model Adequacy. In Environmental Modelling: Finding Simplicity in Complexity (to appear) Wiley-Blackwell.

Developments in the Emulation of Stochastic Models

For Bayes Linear Stochastic Emulation:

Vernon, I. & Goldstein, M. (2010) *A Bayes Linear Approach to Systems Biology*, MUCM Technical Report 10/10.

Vernon, I., & Goldstein, M. *Bayes Linear Emulation and History Matching of Stochastic Systems Biology Models*, in preparation.

For Bayes Linear Variance Learning see:

Goldstein, M. and Wooff, D. A. (2007). *Bayes Linear Statistics: Theory and Methods*, Chichester: Wiley. Chapter 8.

For the Arabidopsis model used in this talk see (further papers in preparation:

Liu, J., Mehdi, S., Topping, J., Tarkowski, P., and Lindsey, K. (2010), Modelling and experimental analysis of hormonal crosstalk in Arabidopsis, *Mol Syst Biol*, 6.

Emulation in Epidemiology

I. Andrianakis, I. Vernon, N. McCreesh, T.J. McKinley, J.E. Oakley, R. Nsubuga, M. Goldstein and R.G. White (2014), Bayesian history matching and calibration of complex infectious disease models using emulation: a tutorial and a case study on HIV in Uganda, submitted to PLOS Computational Biology

H.C. Johnson, S. Conti, K. Kalageropoulos, R.G. White, K.M. Elfstrom and W.J.Edmunds (2013), A novel emulation-based algorithm for likelihood-free model calibration, in preparation.

M. Farah, P. Birrell, S. Conti, and D. De Angelis (2013), ”Bayesian Emulation and Calibration of a Dynamic Epidemic Model for H1N1 Influenza”. Submitted, under review.

Developments in the Emulation of Stochastic Models

More emulation of stochastic systems biology models:

Henderson, D. A., Boys, R. J., Krishnan, K. J., Lawless, C., and Wilkinson, D. J. (2009). Bayesian emulation and calibration of a stochastic computer model of mitochondrial DNA deletions in substantia nigra neurons. Journal of the American Statistical Association 104, 7687.

Henderson, D. A., Boys, R. J. and Wilkinson, D. J. (2010), Bayesian Calibration of a Stochastic Kinetic Computer Model Using Multiple Data Sources. Biometrics 66, 249-256.

Design for stochastic models (needs simple heteroscedasticity):

Boukouvalas, A., Cornford, D., Stehlk, M. (2009), Approximately Optimal Experimental Design for Heteroscedastic Gaussian Process Models. MUCM Technical report 09/06.