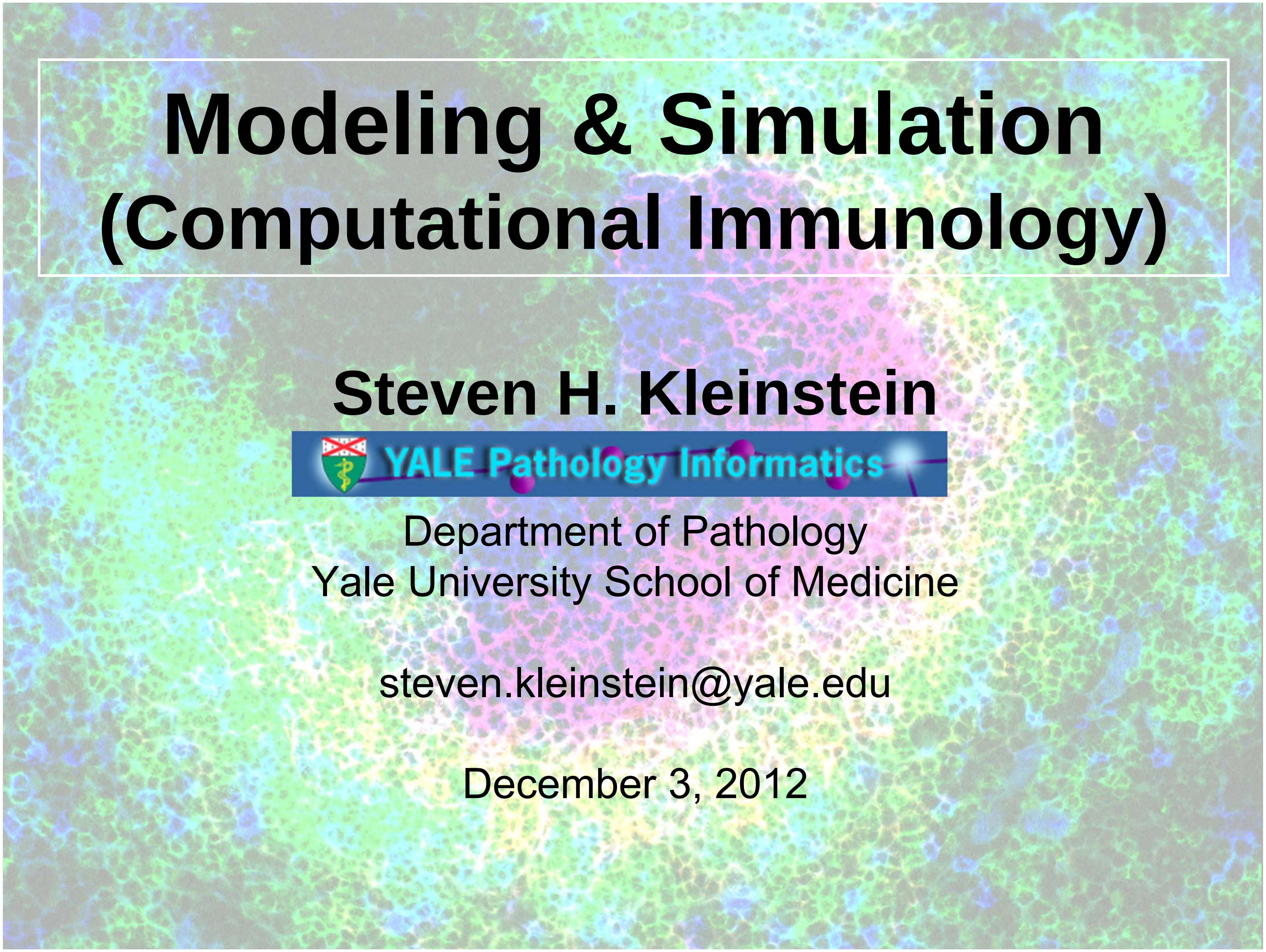


Modeling & Simulation (Computational Immunology)

Steven H. Kleinstein

YALE Pathology Informatics

Department of Pathology
Yale University School of Medicine

steven.kleinstein@yale.edu

December 3, 2012

Inverse Model

- A mathematical model designed to fit experimental data so as to explicitly quantify physical or physiological parameters of interest
- Values of model elements are obtained using parameter estimation techniques aimed at providing a “best fit” to the data
- Generally involves an iterative process to minimize the average difference between the model and the data
- Evaluating the quality of an inverse model involves a combination of established mathematical techniques as well as intuition and creative insight

Six Steps for Inverse-Modeling of Data

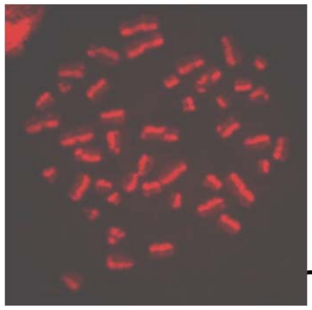
1. Select an appropriate mathematical model
   - Polynomial or other functional form
   - Based on underlying theoretical equations
2. Define a “figure of merit” function
   - Measures agreement between data & model for given parameters
3. Adjust model parameters to get a “best fit”
   - Typically involves minimizing the figure of merit function
4. Examine “goodness of fit” to data
   - Never perfect due to measurement noise
5. Determine whether a much better fit is possible
   - Tricky due to possible local minima vs. global minimum
   - F-test for comparing models of different complexity
6. Evaluate accuracy of best-fit parameter values
   - Provide confidence limits and determine uniqueness
   - Assess physical reasonability of estimated parameter values

(Costa, Kleinstein and Hershberg, Sci Signal. 2011)

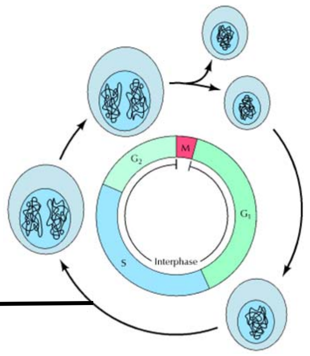
Understanding cell proliferation and death

BrdU (thymidine analog) incorporated into cell DNA during S-phase

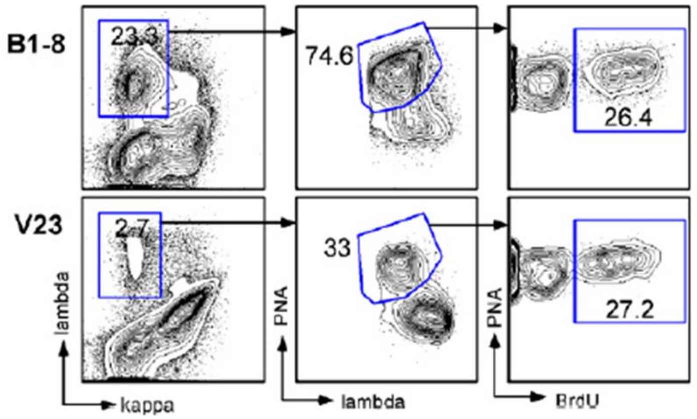
Flow cytometry to quantify antigen-specific germinal center B cells…

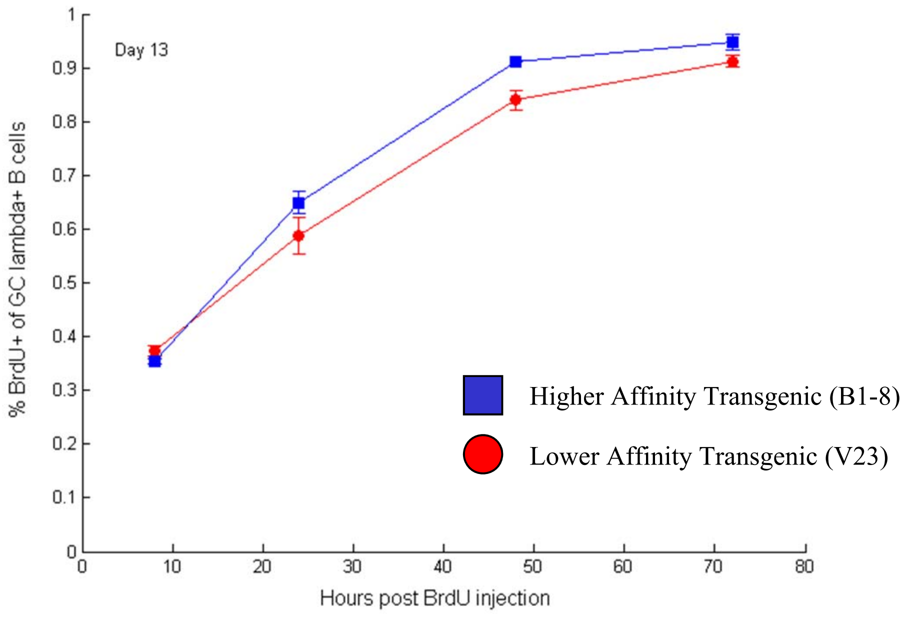


BrdU incorporated during S phase



science.csustan.edu/confocal/Images/SCE/index.SCE.htm

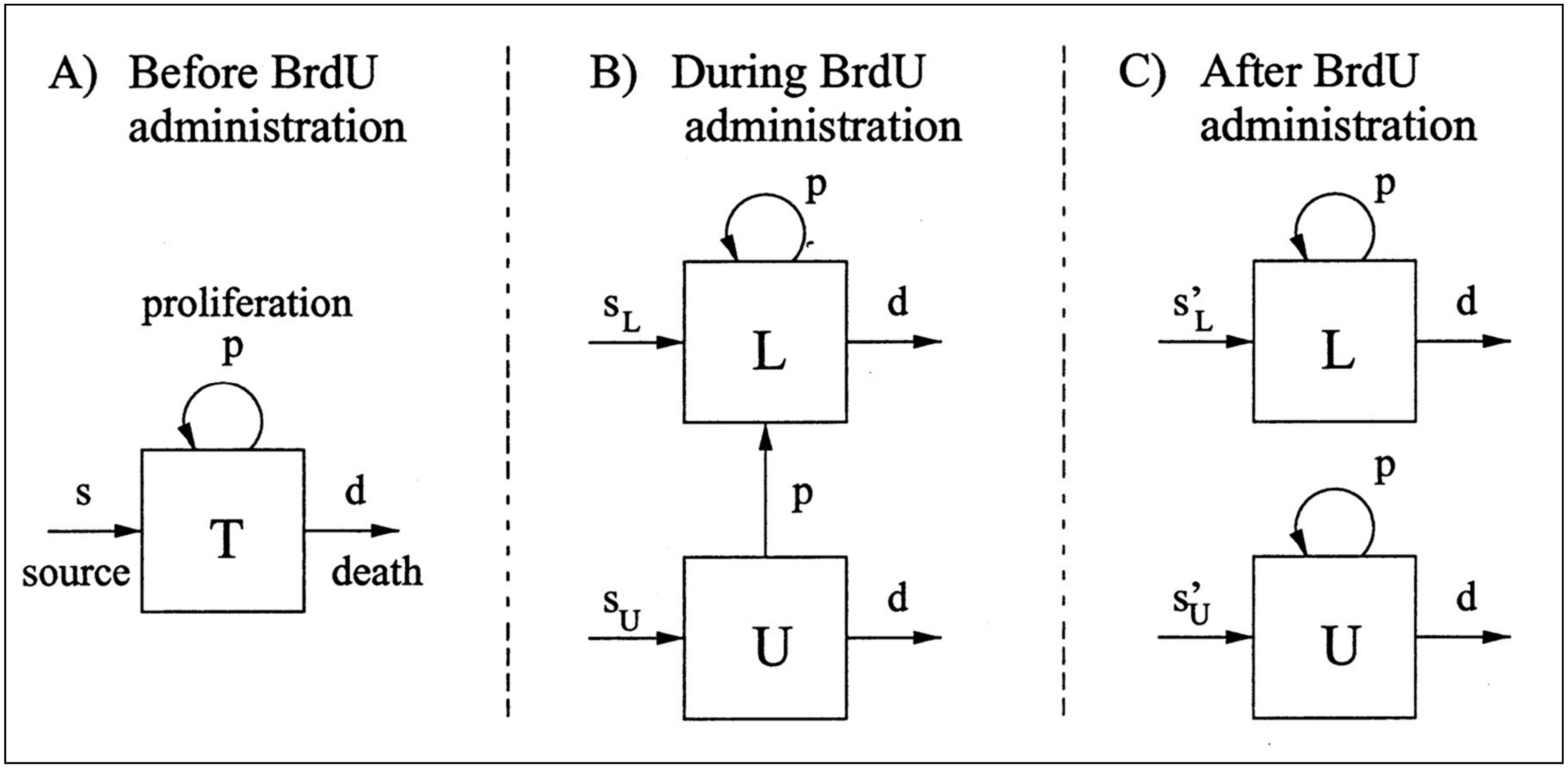




Labeling curves look similar – suggests same proliferation rate?

Model of BrdU Labeling

Model changes with time, expressed as a set of ODEs

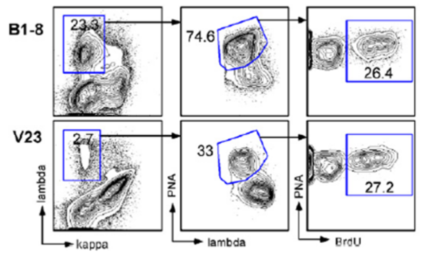

A) Before BrdU administration
proliferation
p
s
source
T
d
death
B) During BrdU administration
p
s_L
L
d
p
s_U
U
d
C) After BrdU administration
p
s'_L
L
d
p
s'_U
U
d

What are the implicit biological assumptions in this model?

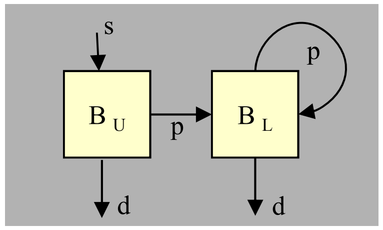
Interaction of Computation & Experiment

Compare simulation and experiment using least-squares objective

Experimental Observations



Computational Model

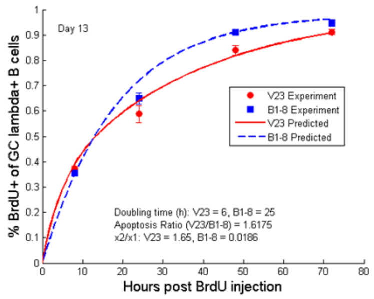


Least-squares objective function

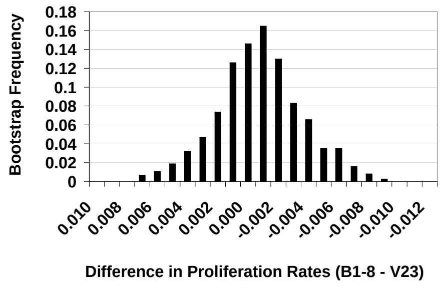
$$E = \sum_i \frac{(y_i - \hat{y}_i)^2}{VAR(y_i)}$$

New Experiments

Fit Model to Data



Model Predictions

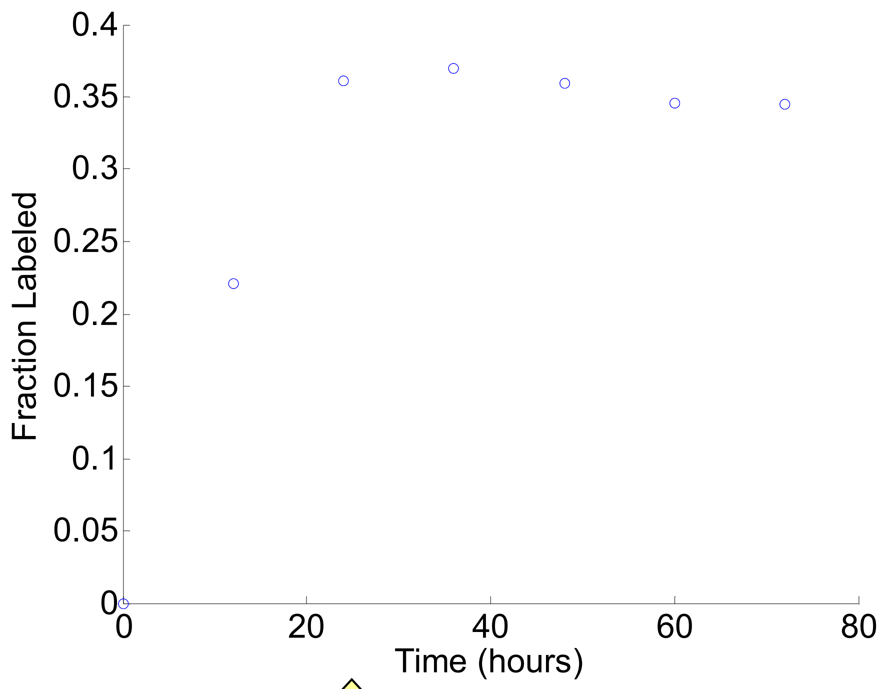


Bootstrapping Confidence Intervals

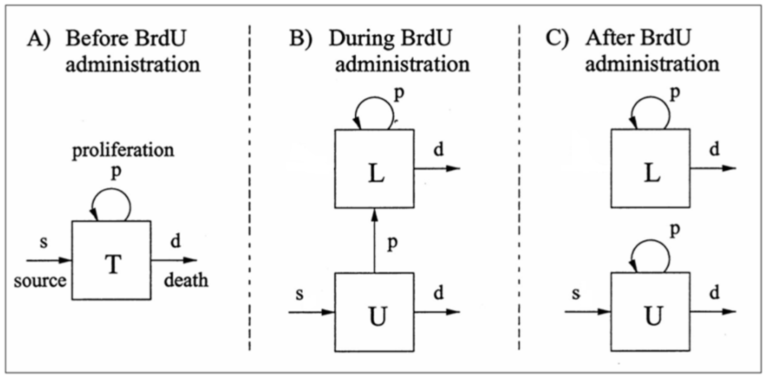
Continuous cycle of modeling and experimentation

Simulated Experiment

Demonstrate full cycle of fitting model to data to estimate parameters



BrdU withdrawn


A) Before BrdU administration
proliferation p
s
source
T
d
death
B) During BrdU administration
p
L
d
p
s
U
d
C) After BrdU administration
p
L
d
p
s
U
d

Parameters used to create synthetic data

s = 0.003 per hour

p = 0.01 per hour

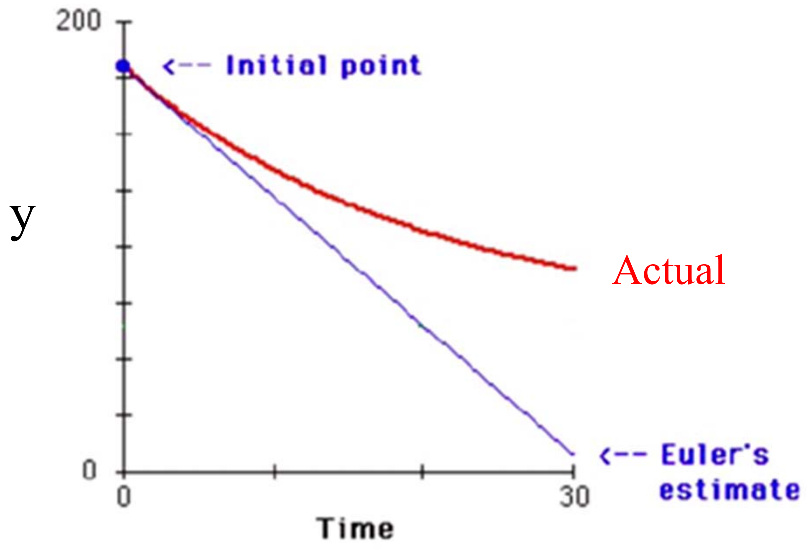
d = p + s (to achieve steady state)

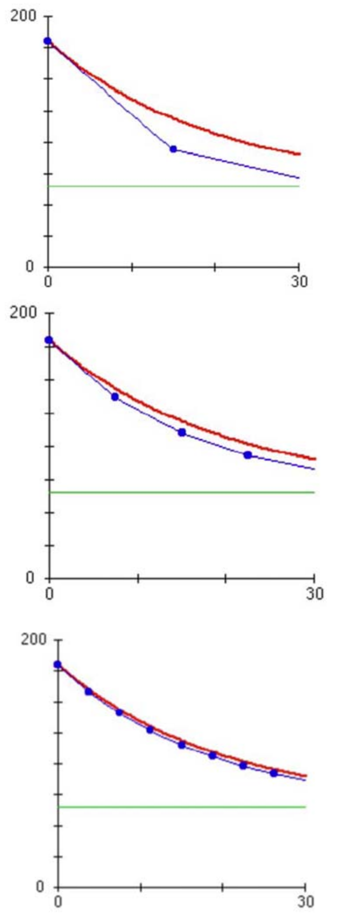
Random noise added to each data point

How can we estimate flow/proliferation/death rates?

Numerical solution to ODEs: Euler Method

From any point on curve, find approximation of nearby point on curve by moving a short distance along a line tangent to the curve





$$y'(t) = f(t, y(t)), \qquad y(t_0) = y_0,$$

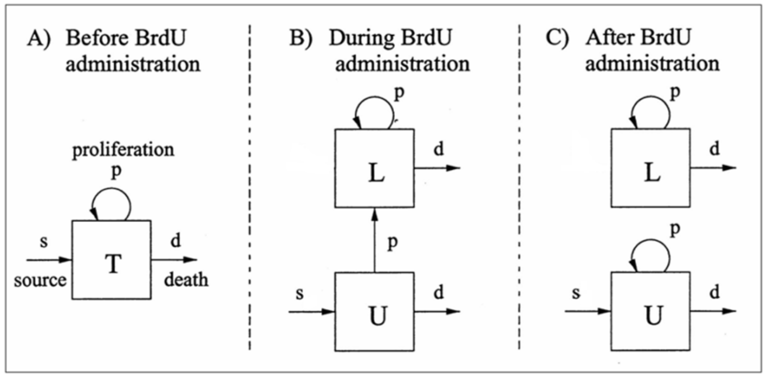
$$y'(t) \approx \frac{y(t+h) - y(t)}{h},$$

$$y(t+h) \approx y(t) + hf(t, y(t)).$$

Much better ways to do this in practice. Eg, Runge-Kutta

Simulating the BrdU Labeling Model

Use integration functions (e.g., ode45 in MATLAB)

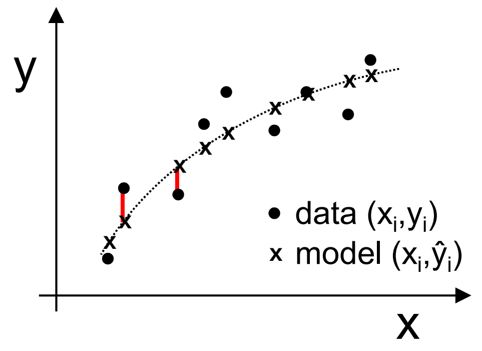

A) Before BrdU administration
proliferation
p
s
source
T
d
death
B) During BrdU administration
p
L
d
p
s
U
d
C) After BrdU administration
p
L
d
p
s
U
d

```
Yin = [1 0];        % Initial Conditions [unlabeled labeled]
pr = [s p d tau];   % Model Parameters
t = [0,12,24,36,48,60,72];   % Times to evaluate
[T,Y] = ode45(@fode,t,Yin,opts,pr);
fl = Y(:,2) ./ sum(Y,2);   % Fraction labeled
```

```
function dy = fode(t, y, pr)
s = pr(1); p = pr(2); d = pr(3); tau = pr(4);
U = y(1); L = y(2);
dy = zeros(2,1); % Vector of derivatives
if (t<tau)  % During BrdU Administration (B)
   dy(1) =  s - p.*U - d.*U;              % dbU/dt
   dy(2) =  2.*p.*U + p.*L - d.*L;        % dbL/dt
else      % After BrdU Administration (C)
   dy(1) =  s + p.*U - d.*U;              %dbU/dt
   dy(2) =       p.*L - d.*L;             %dbL/dt
end
```

Simple models can be solved analytically -- faster

Least-Squares Error Minimization



- Goal is to fit *N* data points (x_i, y_i) i=1..N
- The model is a function with *M* adjustable parameters a_k, k=1..M used to generate *N* model points (x_i, $\hat{y}_i$)

$$\hat{y}_i = \hat{y}(x_i, a_1..a_M)$$

- The <u>residual</u> measures the difference between a data point and the corresponding model estimate

$$y_i - \hat{y}(x_i, a_1..a_M)$$

- Since residuals can be positive or negative, a sum of residuals is <u>not</u> a good measure of overall error in the fit

$$\sum_{i=1}^{N}[y_i - \hat{y}(x_i, a_1..a_M)]$$

- A better measure is the sum of squared residuals, *E*, which is only zero if each and every residual is zero

$$E = \sum_{i=1}^{N}[y_i - \hat{y}(x_i, a_1..a_M)]^2$$

(Costa, Kleinstein and Hershberg, Sci Signal. 2011)

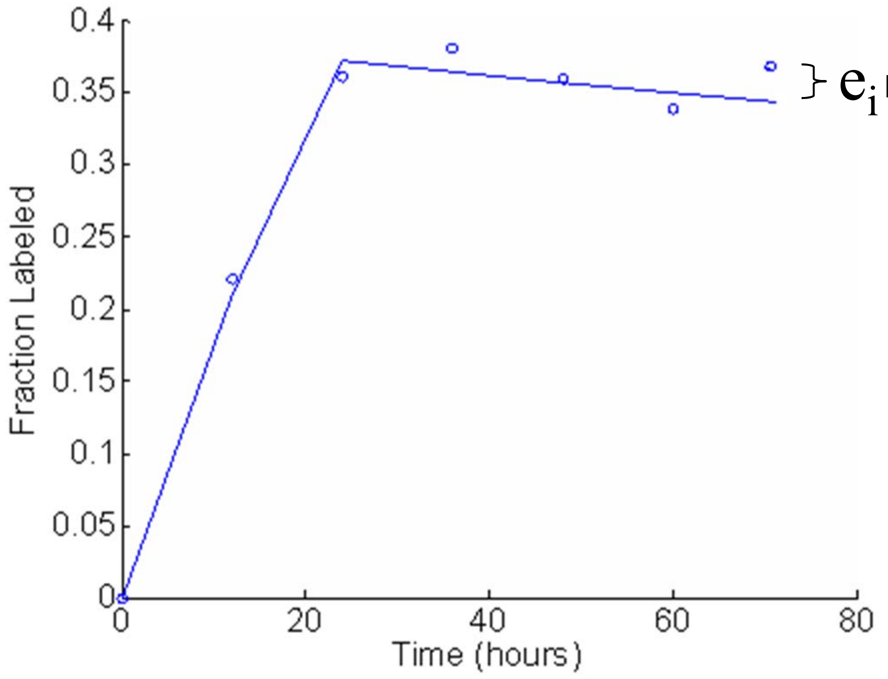
Maximum Likelihood Estimation

- Not meaningful to ask "What is the probability that my set of model parameters is correct?"
  - Only one correct parameter set ➔ Mother Nature!
- Better to ask "Given my set of model parameters, what is the probability that this data set could be obtained?"
  - What is the <u>likelihood</u> of the parameters given the data?
- Inverse modeling is also known as "maximum likelihood estimation".

(Costa, Kleinstein and Hershberg, Sci Signal. 2011)

Fitting the Model to Experimental Data

Compare simulation and experiment using least-squares objective



Least-squares objective function

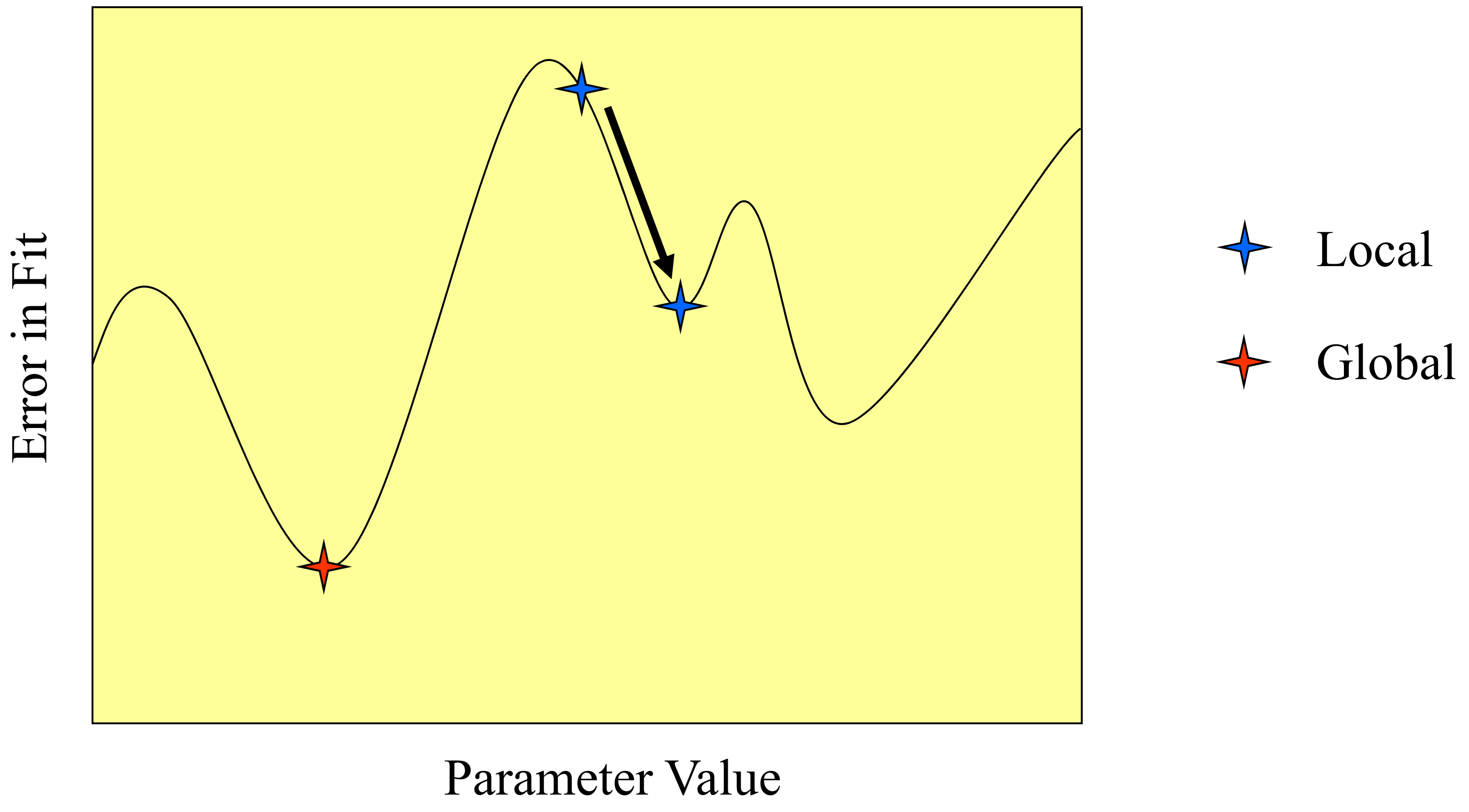
$$E = \sum_i \frac{(y_i - \hat{y}_i)^2}{VAR(y_i)}$$

Find parameters to minimize objective

Many options for how to optimize the fit

Local and Global Optimization

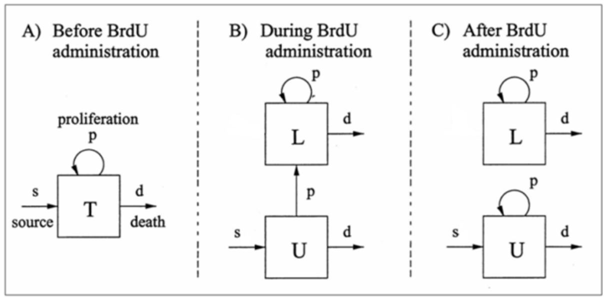
The error function depends on M model parameters, and can be thought of as an M-dimensional “surface” of which we seek the minimum



Local optimization techniques find optimal fit around given starting point
Global optimization attempts to avoid local minima

Fitting Models to Data in MATLAB

Several optimization functions available in many programming languages


A) Before BrdU administration
B) During BrdU administration
C) After BrdU administration
proliferation
p
s
source
T
d
death
L
U

```
pri = [.01 .01];  %Initial guess for parameter values to be fitted [s p]
[pr,fval,exitflag] = lsqnonlin (@efun,pri,[],[],options,fl_observed,t,tau);
s = pr(1); p = pr(2); % Optimal parameter values
```

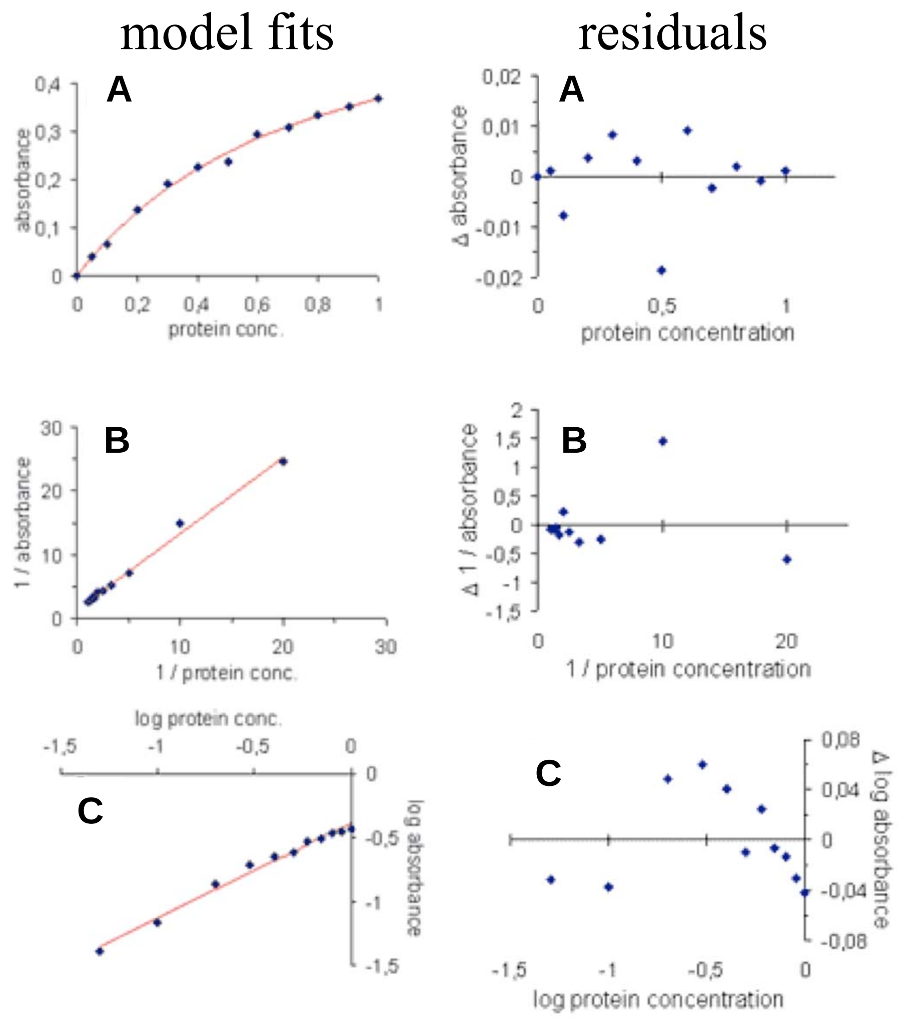
$\underbrace{\texttt{fl\_observed,t,tau}}_{\text{Optional parameters}}$

```
function error = efun (pr,fl_observed,t,tau)
  s = pr(1); p = pr(2); d = s+p;                 % Assume steady-state
  [fl_predicted] = labelBrdU(s,p,d,tau,t);       % Function that simulates model
  error = sum((fl_predicted-fl_observed).^2);    % Least-squares objective
```

lsqnonlin, fminsearch, fmincon, fminbnd

Goodness of Fit and the Residuals Plot

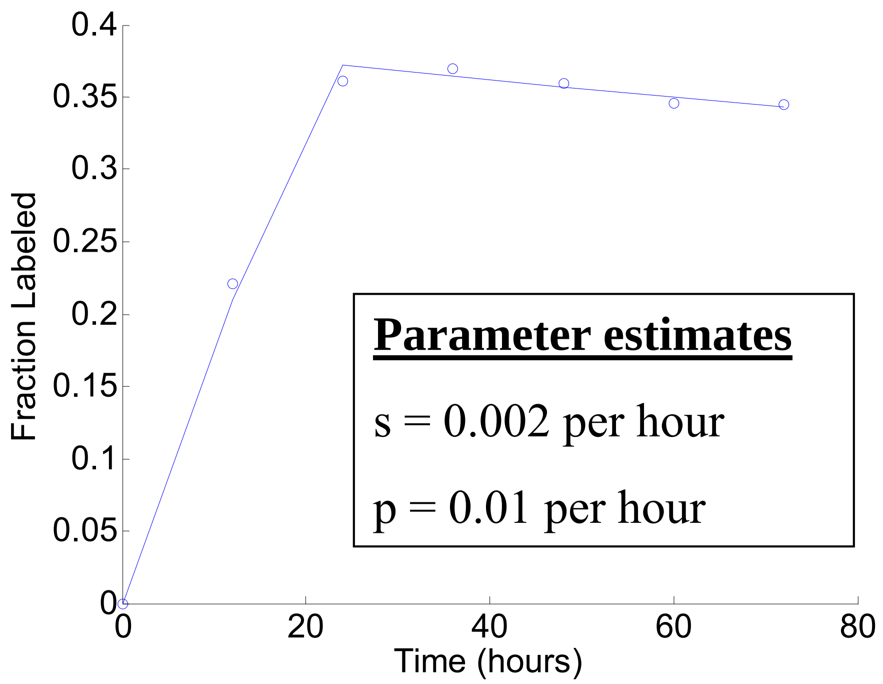
- A high correlation can exist even for a model that systematically differs from the data (all 3 examples have $r^2 > 0.99$)
- One must also examine the distribution of residuals—a good model fit should yield residuals equally distributed along *x* and normally distributed around zero with no systematic trends, as in A rather than B or C



adapted from Lobemeier, 2000

Optimal Parameter Estimates

Least-squares fit using lsqnonlin in MATLAB

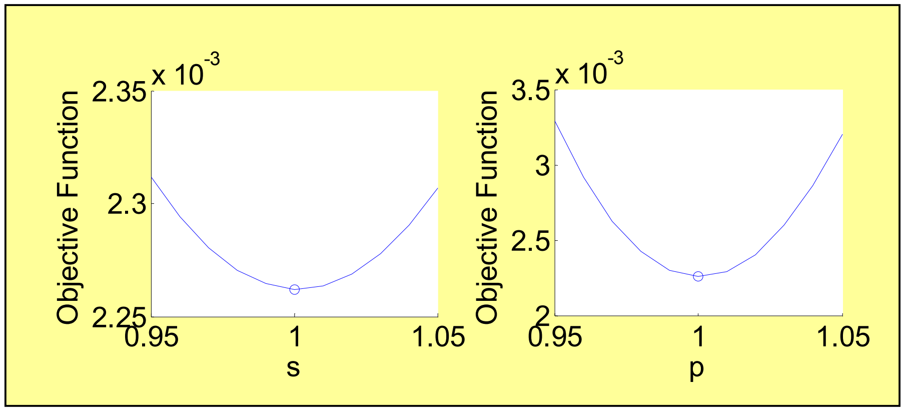


Parameter estimates

s = 0.002 per hour

p = 0.01 per hour

Plot local curvature to check minimization…



Recall, parameters used to create data:

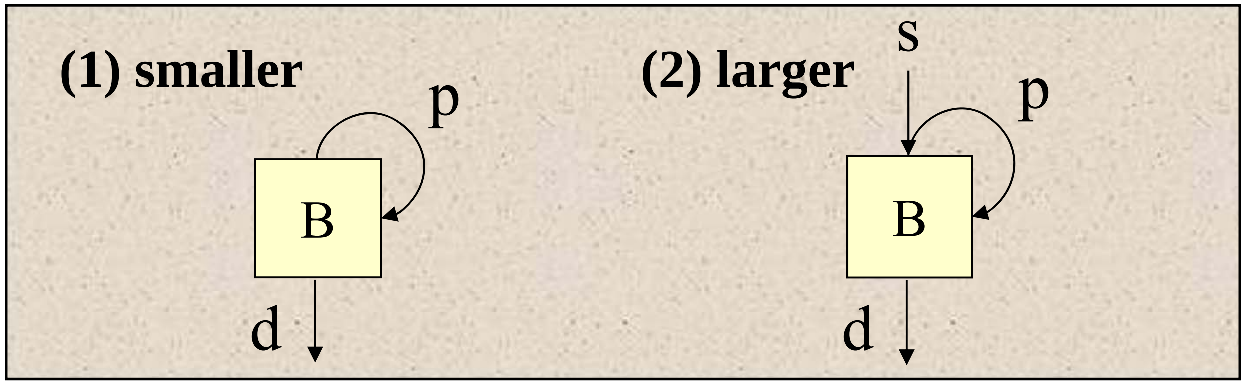
s = 0.003 per hour

p = 0.01 per hour

d = p + s (to achieve steady state)

Is inflow necessary to fit the data? Can we use simpler model?

Is inflow (s) significant?



Residual Sum of Squares $\boxed{RSS = \sum_i (y_i - \hat{y}_i)^2}$

$$F = \frac{\left(\text{RSS}_{\text{smaller}} - \text{RSS}_{\text{larger}}\right) / \left(df_{\text{smaller}} - df_{\text{larger}}\right)}{\text{RSS}_{\text{larger}} / df_{\text{larger}}}$$

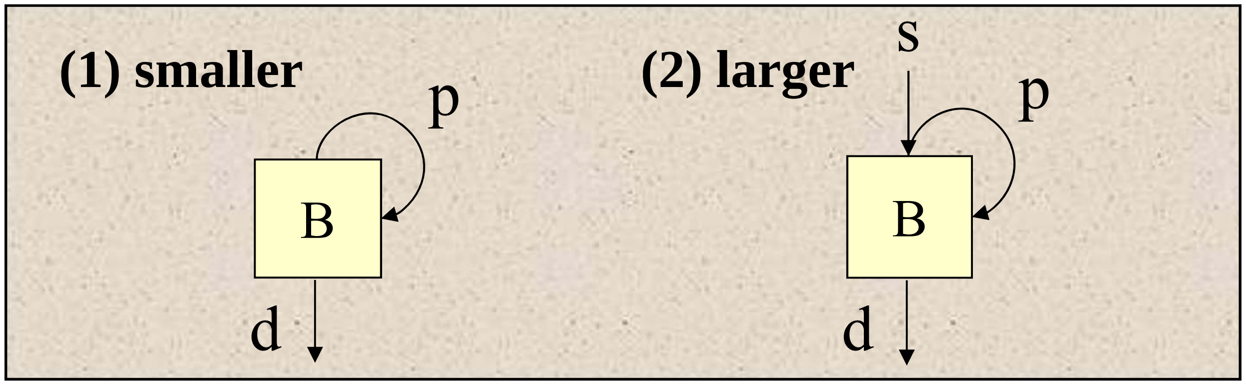
} Reduction in RSS per extra parameter

} Measure of ‘noise’ in model

Degrees of Freedom $\boxed{df = \#observations - \#parameters}$

F distribution with ($df_{smaller}$-df_{larger}, df_{larger}) degrees of freedom

Is inflow (s) significant?



$$F = \frac{\left. \mathrm{RSS}_{\mathrm{smaller}} - \mathrm{RSS}_{\mathrm{larger}} \right/ df_{\mathrm{smaller}} - df_{\mathrm{larger}}}{\left. \mathrm{RSS}_{\mathrm{larger}} \right/ df_{\mathrm{larger}}}$$

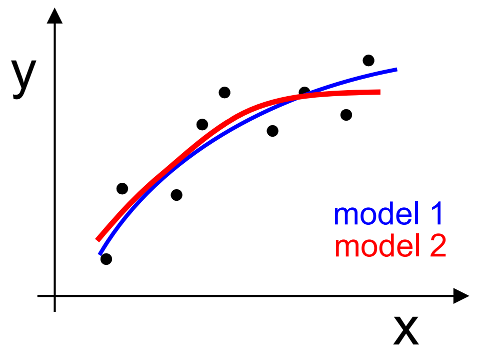
Reduction in RSS per extra parameter

Measure of 'noise' in model

	Observations	**Parameters**	**RSS**	**F test (1-fcdf in MATLAB)**
(1) No flow (s=0)	6	1	9.38e-7	
(2) Including flow	6	2	0.95e-7	**53.1 (p<0.0004)**

Inflow (s) is important to explain observations

Comparing Two Model Fits



- The number of data points, *N*, must exceed the number of model parameters, *M*, yielding the degrees of freedom (*DOF* = *N*-*M*)

$$M \le N - 1$$

- Increasing *M* using a more complex model will generally improve the quality of fit and reduce RSS
- Increasing *MSE* with decreasing RSS can reveal an over-parameterized model

$$MSE = \frac{RSS}{N - M} = \frac{RSS}{DOF}$$

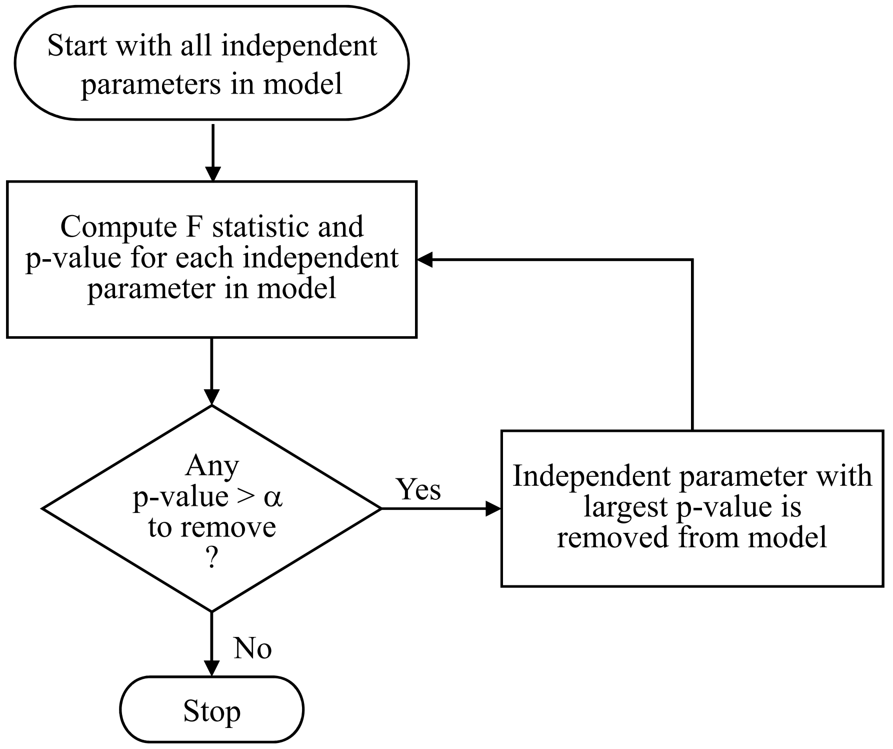
- An F-statistic can be computed to compare the results of two model fits
  - F ~ 1, the simpler model is adequate
  - F > 1, the more complex model is better, or random error led to a better fit with the complex model
  - P-value defines the probability of such a “false positive” result (lookup in F table)

(Costa, Kleinstein and Hershberg, Sci Signal. 2011)

Building models with variable selection

F statistic determines if variable added or deleted from model

Backward Elimination


Start with all independent parameters in model
Compute F statistic and p-value for each independent parameter in model
Any p-value > α to remove ?
Yes
Independent parameter with largest p-value is removed from model
No
Stop

Other Variations:

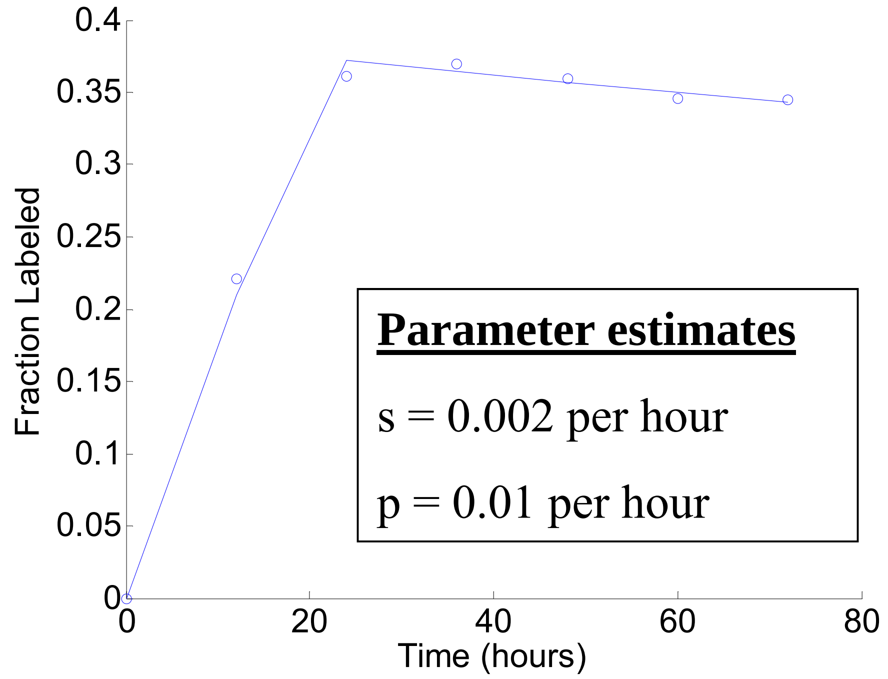
Forward selection: adds variables one at a time as long as significant F test.

Stepwise procedure: allows for removal of a parameter at each step

No guarantee that globally optimal model with be found (need all subsets, but prohibitive for large parameter space)

How much confidence to put in estimate?

Construct confidence intervals for model parameters



Parameter estimates

s = 0.002 per hour

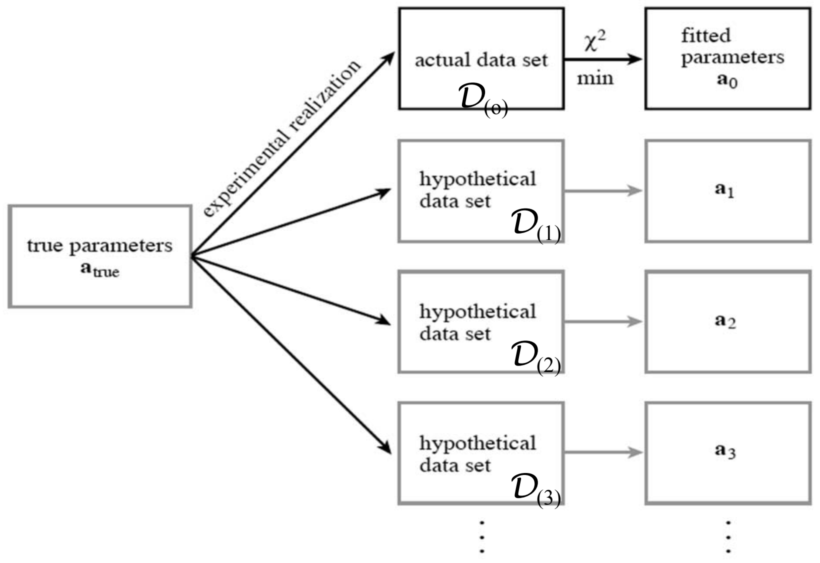
p = 0.01 per hour

Estimate uncertainty given limited number of experimental observations

Accuracy of Estimated Model Parameters

Underlying true set of model parameters ($\mathbf{a}_{\text{true}}$) known to Mother Nature but hidden from the experimenter

- True parameters are statistically realized as measured data set $\mathcal{D}_{(o)}$



true parameters $\mathbf{a}_{\text{true}}$
experimental realization
actual data set $\mathcal{D}_{(o)}$
χ^2 min
fitted parameters $\mathbf{a}_0$
hypothetical data set $\mathcal{D}_{(1)}$
$\mathbf{a}_1$
hypothetical data set $\mathcal{D}_{(2)}$
$\mathbf{a}_2$
hypothetical data set $\mathcal{D}_{(3)}$
$\mathbf{a}_3$

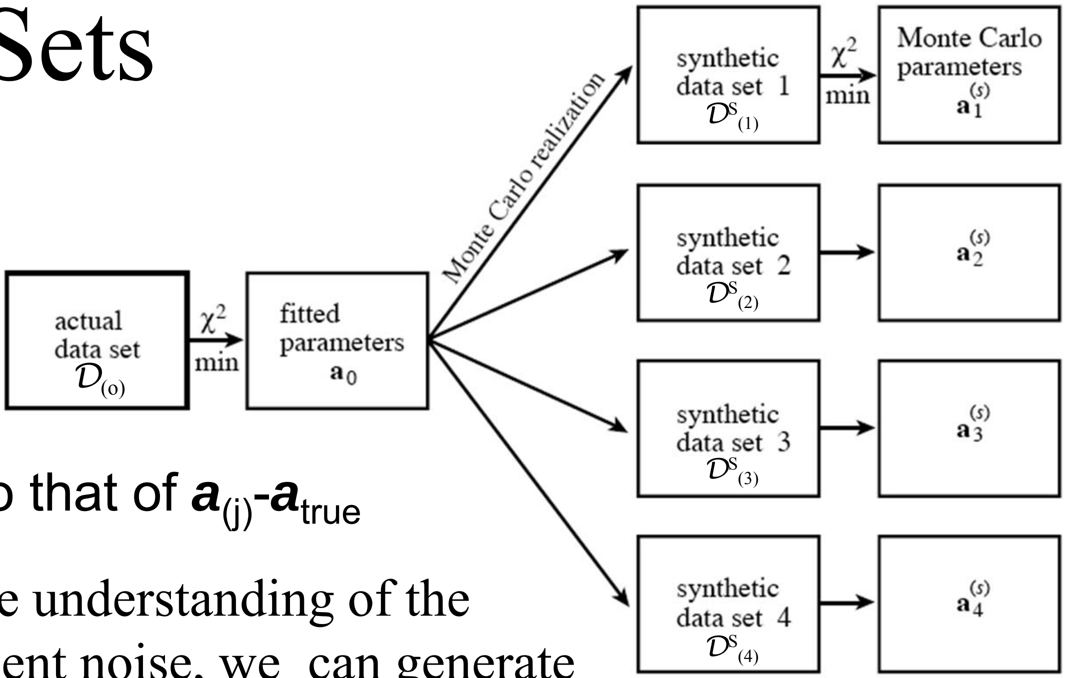
from Numerical Recipes online

- Fitting $\mathcal{D}_{(o)}$ yields estimated model parameters $\mathbf{a}_{(o)}$
- Other experiments could have resulted in data sets $\mathcal{D}_{(1)}$, $\mathcal{D}_{(2)}$, etc. which would have yielded model parameters $\mathbf{a}_{(1)}$, $\mathbf{a}_{(2)}$, etc.

Estimate probability distribution of $\mathbf{a}_{(i)} - \mathbf{a}_{\text{true}}$ without knowing $\mathbf{a}_{\text{true}}$

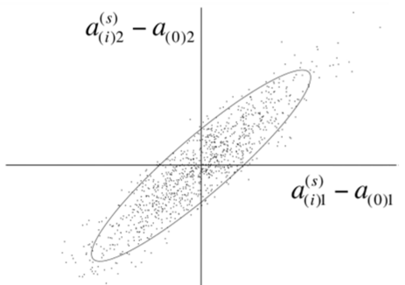
Monte Carlo Simulation of Synthetic Data Sets

- Assume that if $\boldsymbol{a}_{(0)}$ is a reasonable estimate of $\boldsymbol{a}_{\text{true}}$, then the distribution of $\boldsymbol{a}_{(j)}$-$\boldsymbol{a}_{(0)}$ should be similar to that of $\boldsymbol{a}_{(j)}$-$\boldsymbol{a}_{\text{true}}$


actual data set $\mathcal{D}_{(o)}$
χ^2 min
fitted parameters $\mathbf{a}_0$
Monte Carlo realization
synthetic data set 1 $\mathcal{D}^S_{(1)}$
synthetic data set 2 $\mathcal{D}^S_{(2)}$
synthetic data set 3 $\mathcal{D}^S_{(3)}$
synthetic data set 4 $\mathcal{D}^S_{(4)}$
χ^2 min
Monte Carlo parameters $\mathbf{a}_1^{(s)}$
$\mathbf{a}_2^{(s)}$
$\mathbf{a}_3^{(s)}$
$\mathbf{a}_4^{(s)}$

from Numerical Recipes online

- With the assumed $\boldsymbol{a}_{(0)}$, and some understanding of the characteristics of the measurement noise, we can generate "synthetic data sets" $\mathcal{D}^S_{(1)}$, $\mathcal{D}^S_{(2)}$,… at the same x_i values as the actual data set, $\mathcal{D}_{(0)}$, that have the same relationship to $\boldsymbol{a}_{(0)}$ as $\mathcal{D}_{(0)}$ has to $\boldsymbol{a}_{\text{true}}$



2-parameter probability distribution for 1,000 Monte Carlo simulations

- For each $\mathcal{D}^S_{(j)}$, perform a model fit to obtain corresponding $\boldsymbol{a}^S_{(j)}$, yielding one point $\boldsymbol{a}^S_{(j)}$- $\boldsymbol{a}_{(0)}$ for simulating the desired M-dimensional probability distribution. **This is a very powerful technique!!**

(Costa, Kleinstein and Hershberg, Sci Signal. 2011)

The Bootstrap Method

Estimating generalization error based on "resampling":
Randomly draw datasets with replacement from training data

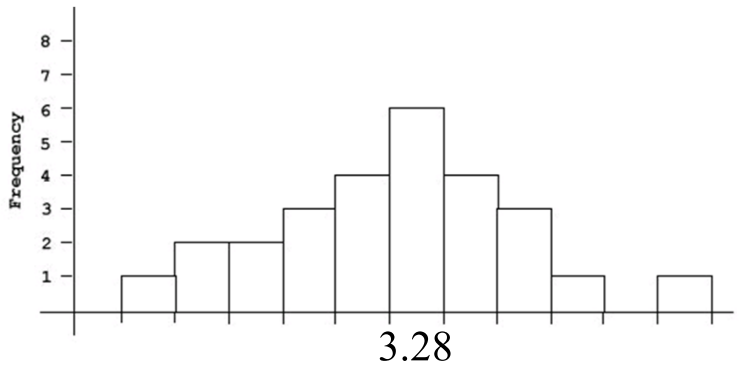
- If don't know enough about the measurement errors (i.e. cannot even say they are normally distributed) so Monte Carlo simulation cannot be used.
- Bootstrap Method uses actual data set $\mathcal{D}_{(o)}$, with its N data points, to generate synthetic data sets $\mathcal{D}^{S}_{(1)}$, $\mathcal{D}^{S}_{(2)}$,… also with N data points.
- Randomly select N data points from $\mathcal{D}_{(o)}$ *with replacement*, which makes $\mathcal{D}^{S}_{(j)}$ differ from $\mathcal{D}_{(o)}$ with a fraction of the original points replaced by *duplicated* original points.
- Fitting the $\mathcal{D}^{S}_{(j)}$ data yields model parameter sets $\mathbf{a}^{S}_{(j)}$ using actual measurement noise.

If sample is good approximation of population, bootstrap method will provide good approximation of sampling distribution of original statistic.

Bootstrap Methods

Randomly draw datasets with replacement from training data

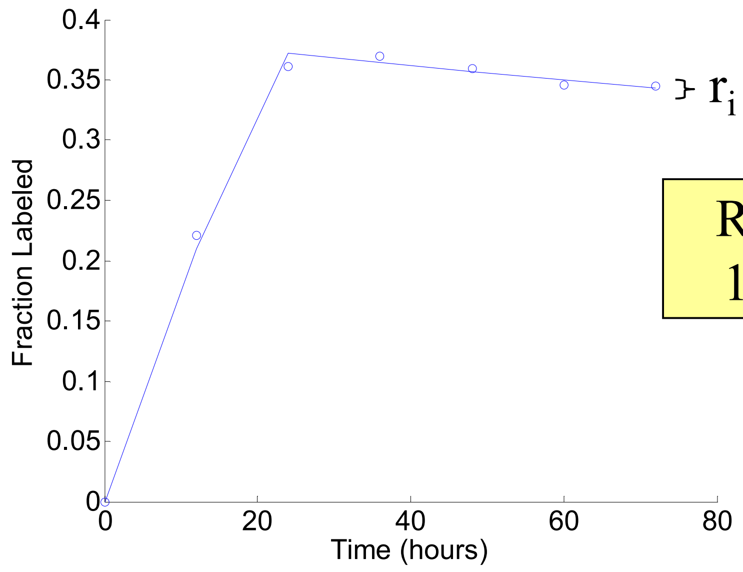
- D = [3.0, 2.8, 3.7, 3.4, 3.5] → average = 3.28
- Bootstrap samples D_N could be:
  - [2.8, 3.4, 3.7, 3.4, 3.5] → 3.36
  - [3.5, 3.0, 3.4, 2.8, 3.7] → 3.28
  - [3.5, 3.5, 3.4, 3.0, 2.8] → 3.24
  - ...



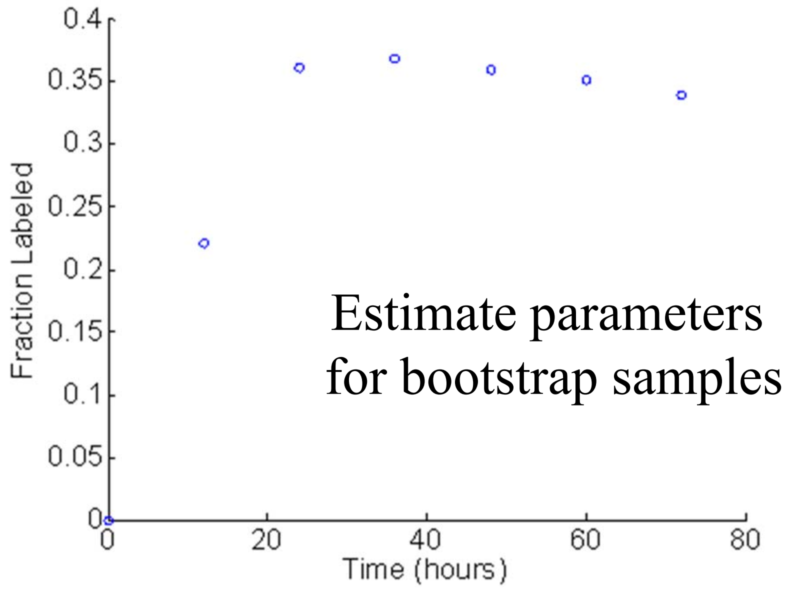
If sample is good approximation of population, bootstrap method will provide good approximation of sampling distribution of original statistic.

Bootstrapping Parameter Confidence Intervals

1) Fit model to data to obtain parameter estimates
2) Draw a bootstrap sample of the residuals (Fixed-X Bootstrapping)
3) Create bootstrap sample of observations by adding randomly sampled residual to predicted value of each observation



Repeat 1000x



Estimate parameters for bootstrap samples

Bootstrapping observations also possible – asymptotically equivalent

Bootstrapping Parameter Confidence Intervals

Three commonly used methods: 1. Normal Theory Intervals, 2. Percentile Intervals, 3. Bias Corrected Percentile Intervals

Percentile Intervals

Calculate the parameter for each bootstrap sample and select α (e.g., 0.05)

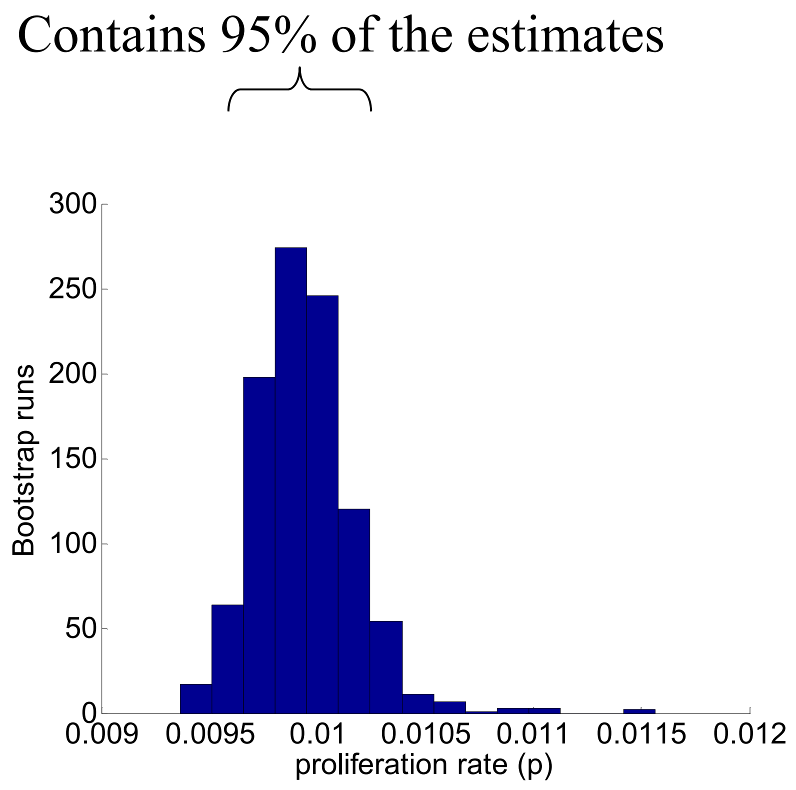
LCL = α /2th percentile.

UCL = $(1-\alpha/2)$th percentile.

Use MATLAB's prctile function:
= prctile(bootstrap estimates, 0.025)

Parameter estimates for synthetic data

Estimate of s = 0.0017 [0.0009,0.0030]
Estimate of p = 0.0099 [0.0095,0.0100]

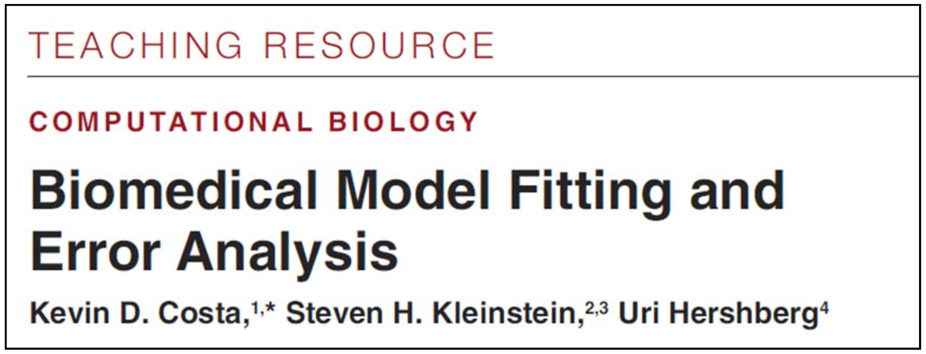


May not have correct coverage when sampling distribution skewed

Practical reference for these kinds of methods

Numerical Recipes:
Includes source code for integration, optimization, etc.

NUMERICAL RECIPES
The Art of Scientific Computing
Third Edition


TEACHING RESOURCE

COMPUTATIONAL BIOLOGY

Biomedical Model Fitting and Error Analysis

Kevin D. Costa,[1,*] Steven H. Kleinstein,[2,3] Uri Hershberg[4]

www.SCIENCESIGNALING.org 27 September 2011 Vol 4 Issue 192

Free NR versions online at http://www.nr.com/oldverswitcher.html