Welcome to Bayesian CBIOMES

January 8-10, 2020
Massachusetts Institute of Technology

Instructors:
Michael Dowd (mdowd@mathstat.dal.caa)
Gregory Britten (gbritten@mit.edu)
Paul Mattern (jmattern@ucsc.edu)

INTRODUCTIONS

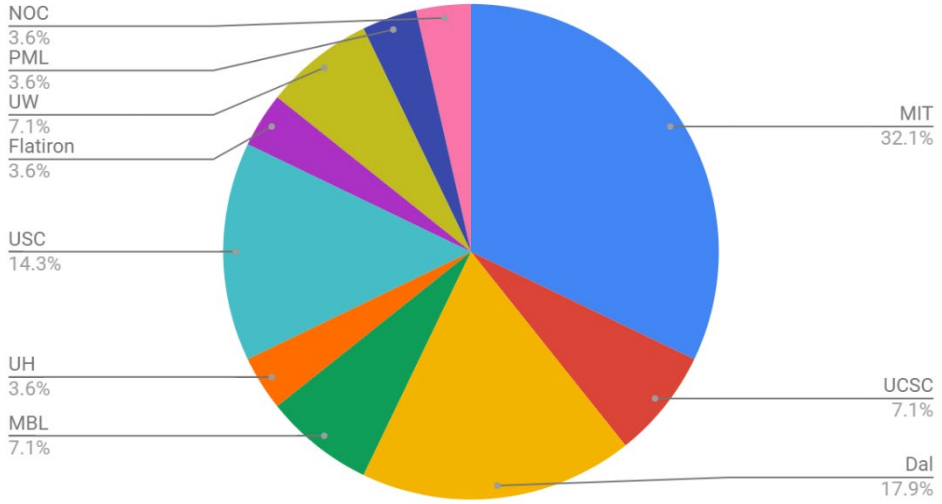
Who are you, where do you work, and what do you work on?

What’s your stats and non-stats background?

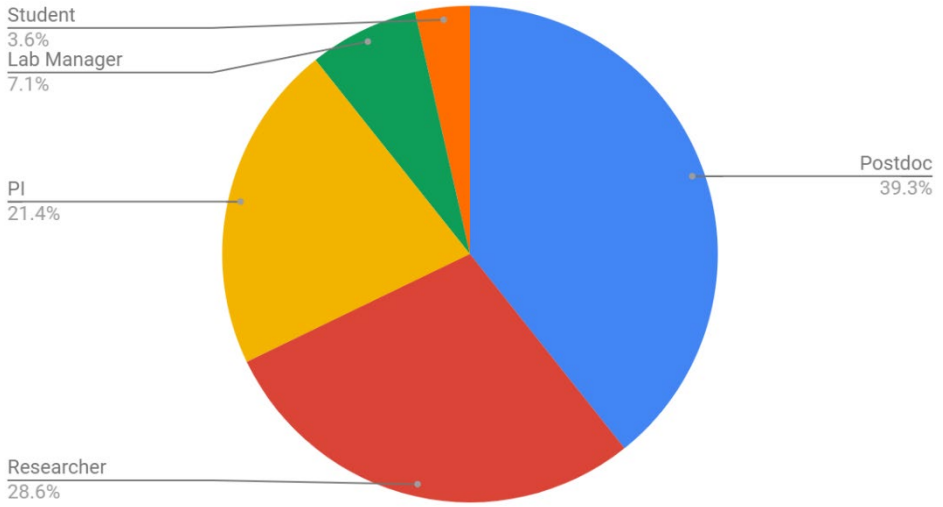
What do you want to get out of the workshop?

Our inter-disciplinary breakdown

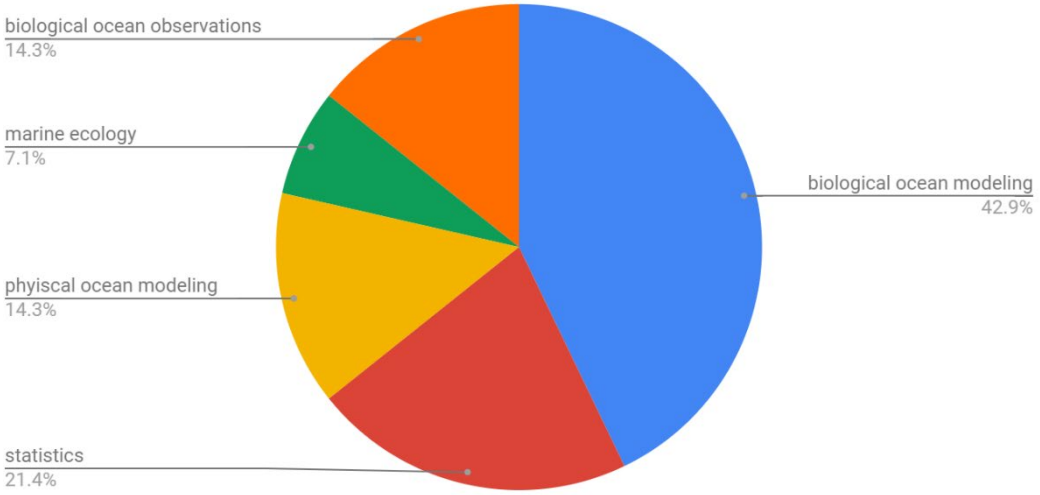
Institution



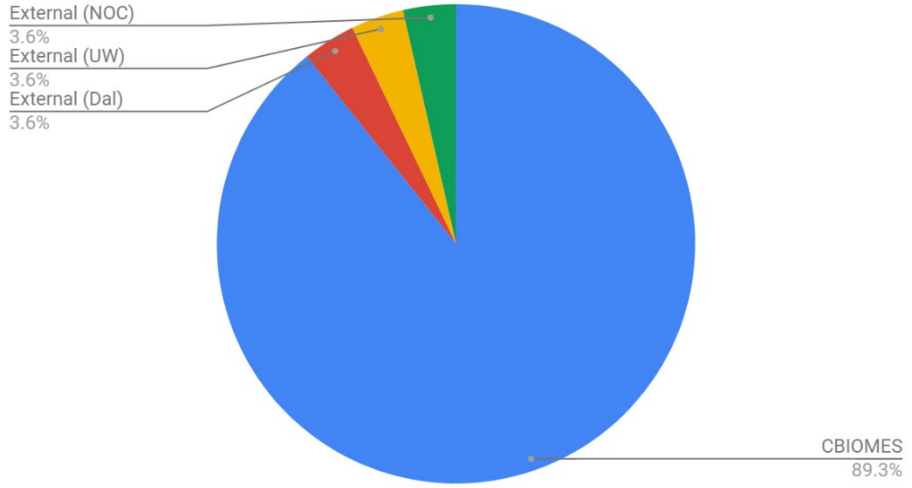
Academic Level



Primary Research Category



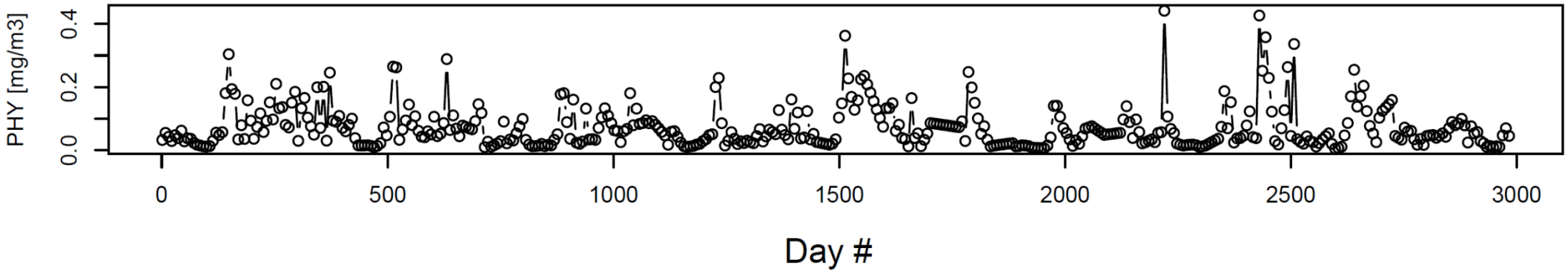
Project Affiliation



Goal: Bayesian analysis of dynamic biological marine systems

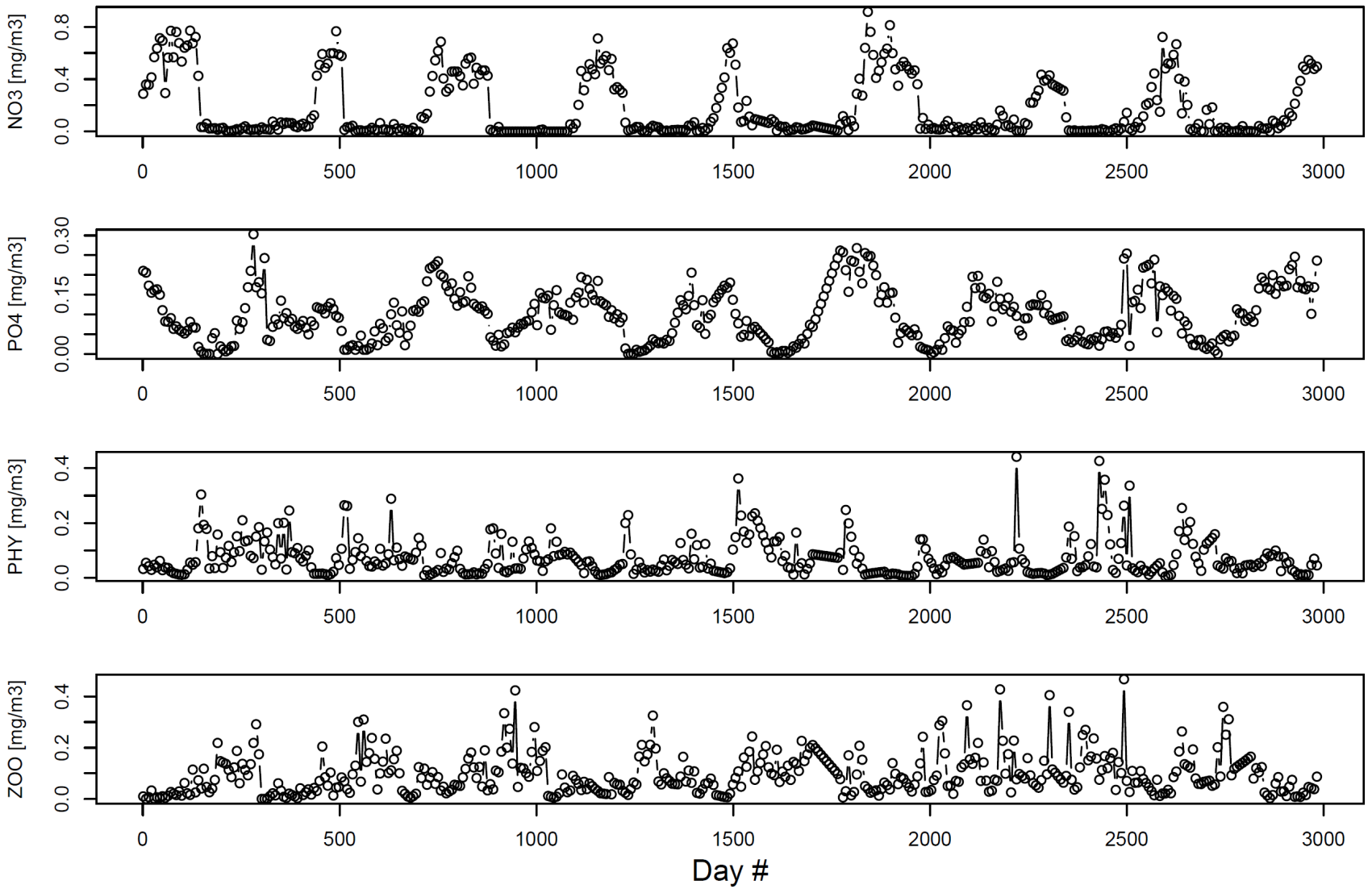
- Dynamic systems that vary with respect to time
- Biological information in rate processes and system interactions
- Use Bayesian analysis to learn information from noisy time series observations

- Will work through four CBIOMES-relevant case studies as examples and/or projects that we can contribute to

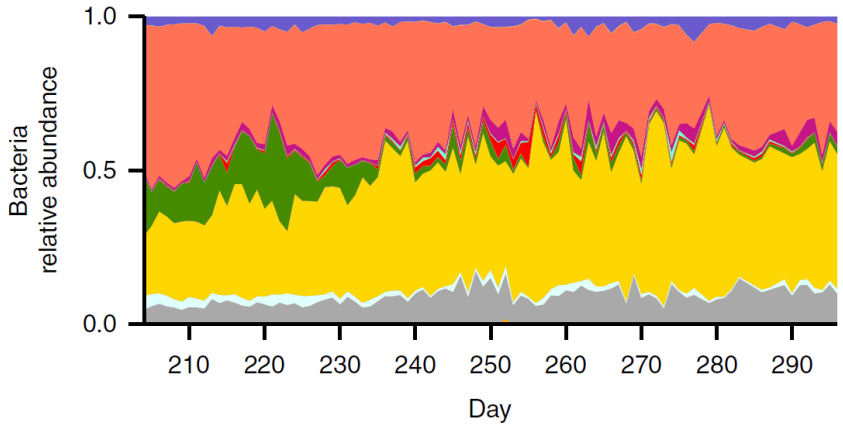


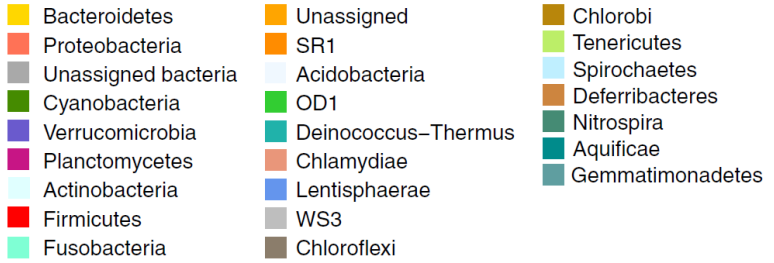
Dynamic systems: *Case Study: Narragansett Nutrients-Phytoplankton-Zooplankton*

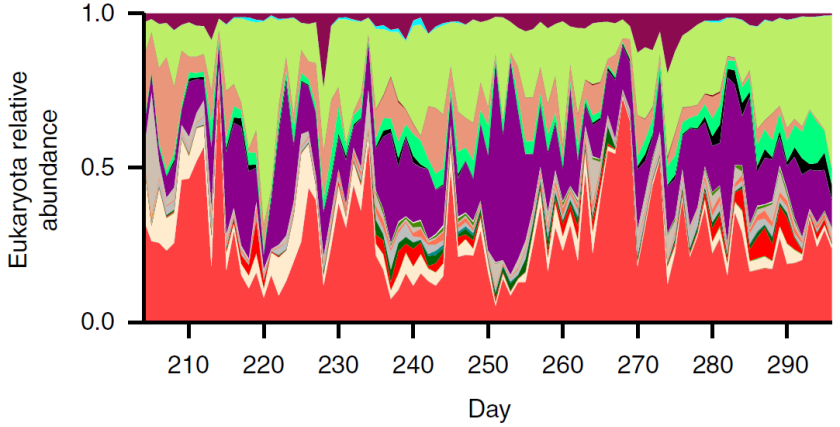
A multivariate time series from Narragansett Bay, Rhode Island

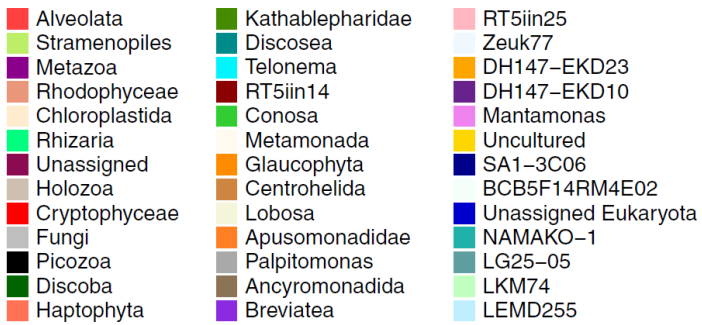


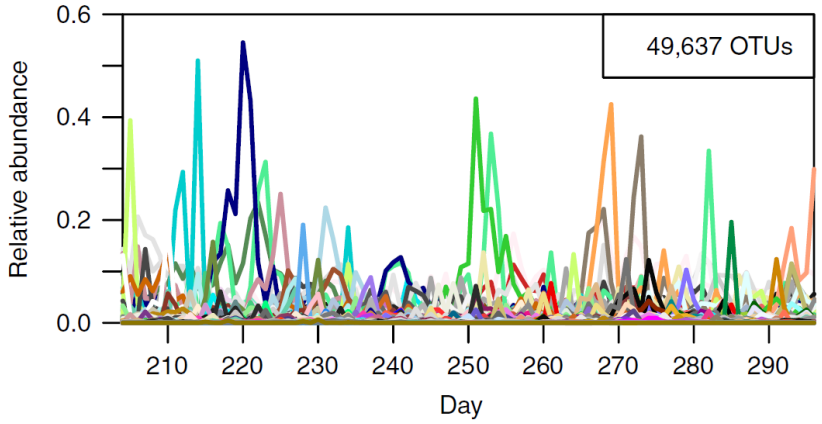
Dynamic systems: *Case study: Multivariate bacterial OTUs*

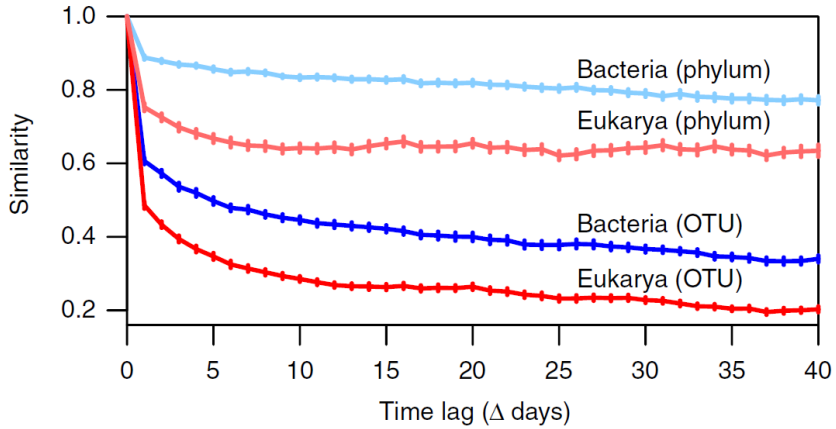




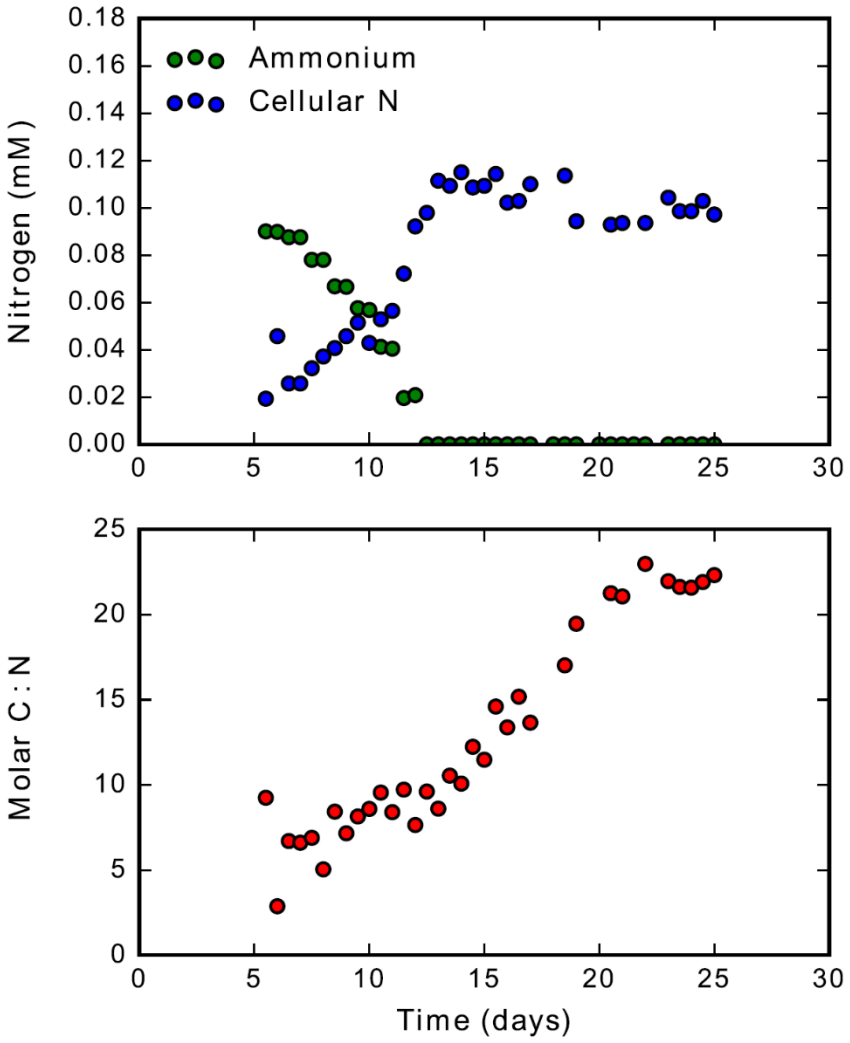


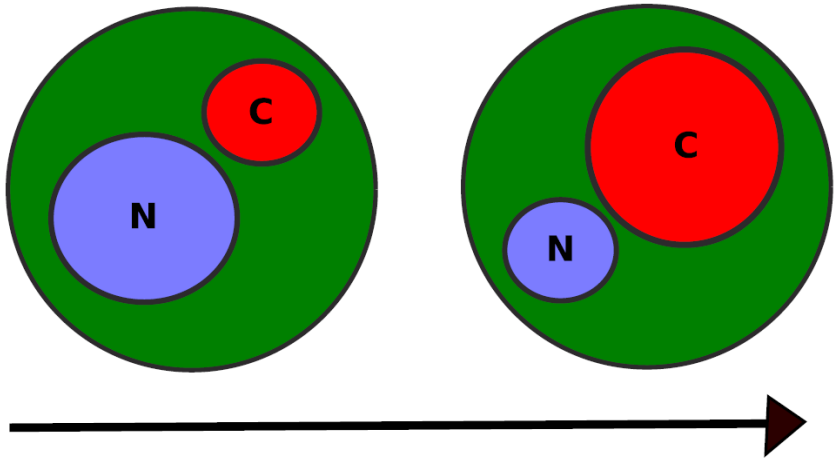




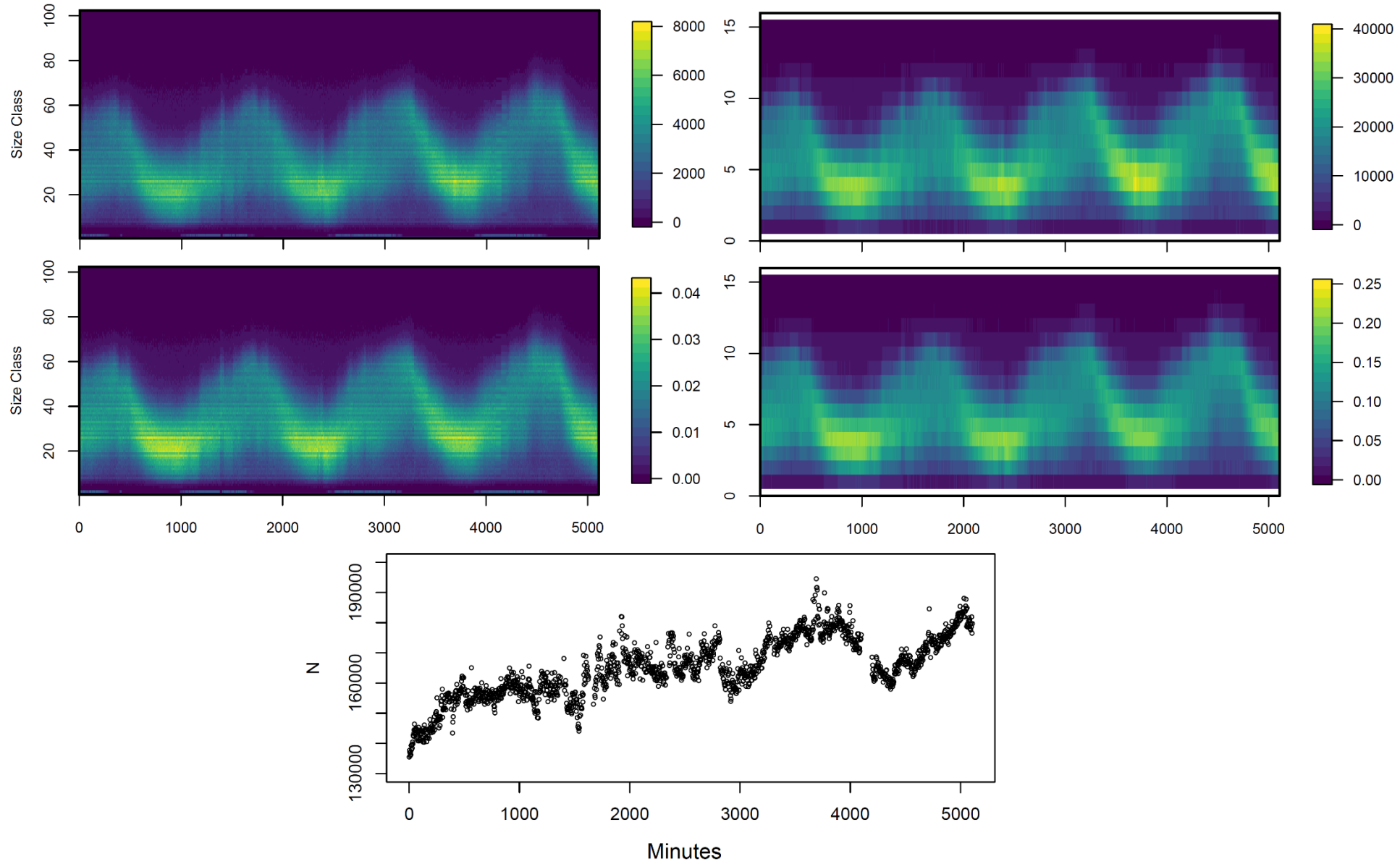


Dynamic systems: *Intra-cellular macromolecules*





Dynamic systems: *Cellular size distributions over the cell cycle*



Outline of workshop

Day #1

1. Introductions
2. GitHub orientation
3. Brief intro to Bayes' theorem
4. Bayesian analysis of dynamical systems
5. MCMC with and without Stan using linear regression and a simple phytoplankton growth model
6. Case study #1: Fitting an NPZ model

Day #2

1. Case study #2: Linear autoregressive model fit to multivariate OTU time series
2. Case study #3: Intracellular macromolecular pool dynamics
3. Case study #4: Fitting cell cycle models to diel size distribution time series
4. Form case study groups and brainstorm analysis ideas
5. Discuss analysis ideas as a group

Day #3

1. Analysis hackathon!
2. Presentations of analyses

GitHub/notebook orientation

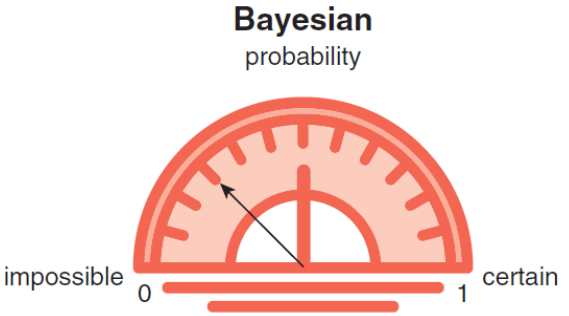
- Can use materials with/without GitHub and with/without Jupyter

- Forking, cloning, pulling, and pushing
- Download as zip

- Lecture materials
- Case study folders
- Notebooks and scripts

Why Bayesian inference?

A more principled way to learn from data in the presence of uncertainty



#0 Concepts of probability

- Two concepts of probability: long-run frequencies vs. state of knowledge
- Bayesians describe all unknowns with probability
- Frequentists only use probability to describe data uncertainty
- Frequentists find the parameters that maximize the probability of the data, assuming the model is true and that all error comes from sampling noisy data; limited ability to describe parameter or model uncertainty
- Bayesians assign prior probabilities to their belief about certain parameters, update those priors using Bayes' rule according to how consistent they are with the data; can characterize uncertainty in a much broader and general way
- BIOLOGY IS FULL OF UNCERTAINTY AND REQUIRES PROPER ACCOUNTING TO MAKE ROBUST INFERENCES

#1 Interpretability of probabilities

- Results in terms of uncertainties/intervals straightforward to interpret
- Results are directly carried forward to predictions
- A lot of information about parameter co-dependencies in the joint posterior distribution

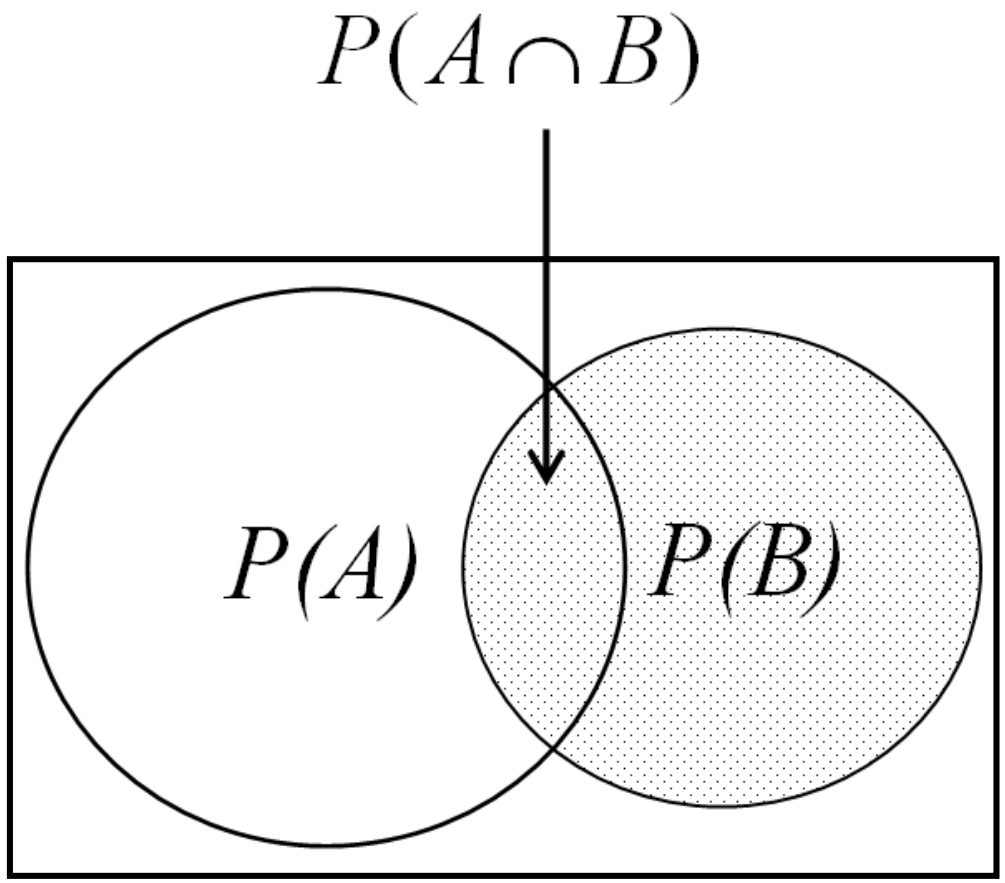
#2 The usefulness of prior distributions

- Allows us to rigorously quantify our prior scientific knowledge about a process and unknown quantities in the calculations
- Prior and observational knowledge naturally blend propagate forward through subsequent calculations

#3 Mathematical consistency, simplicity, and transparency

- ONE PROCEDURE FOR ALL INFERENCE PROBLEMS
- Modifications are transparent – usually manifest as choices of prior distribution or limited search parameters
- More computationally challenging, less so with software and hardware advancements

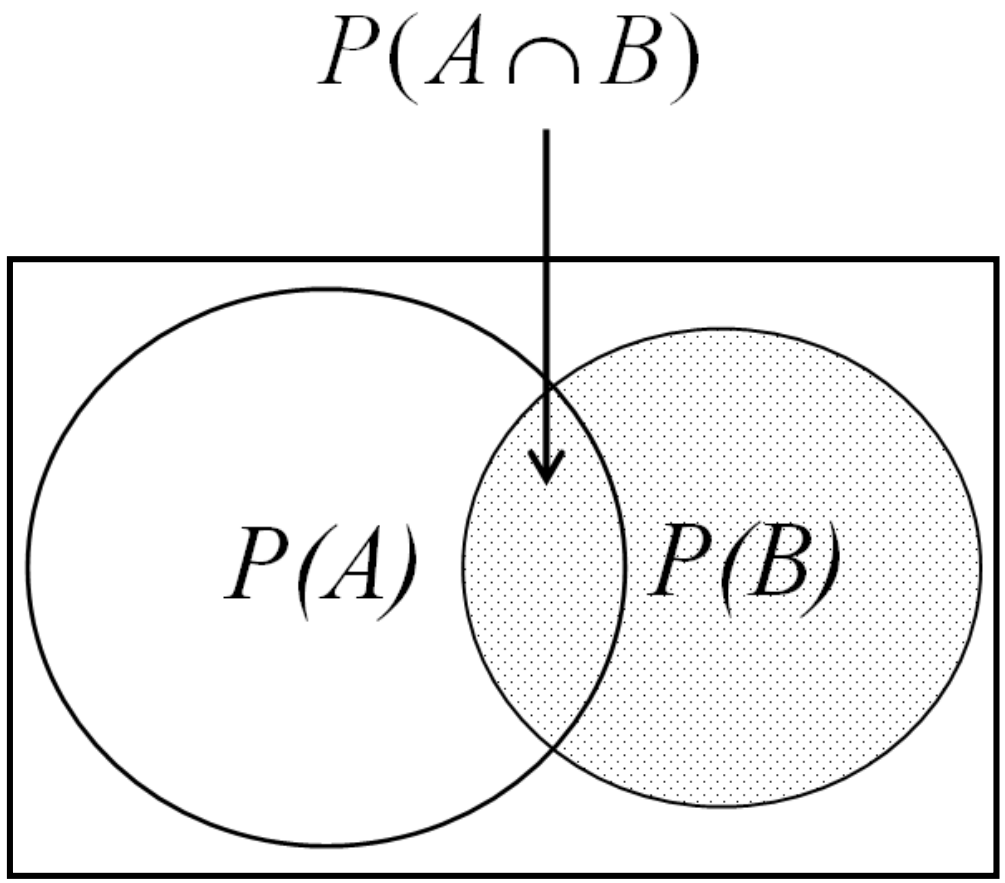
Bayes theorem is extremely simple to derive



$$p(A|B) = \frac{p(A \cap B)}{p(B)}$$

$$p(B|A) = \frac{p(A \cap B)}{p(A)}$$

Bayes theorem is extremely simple to derive



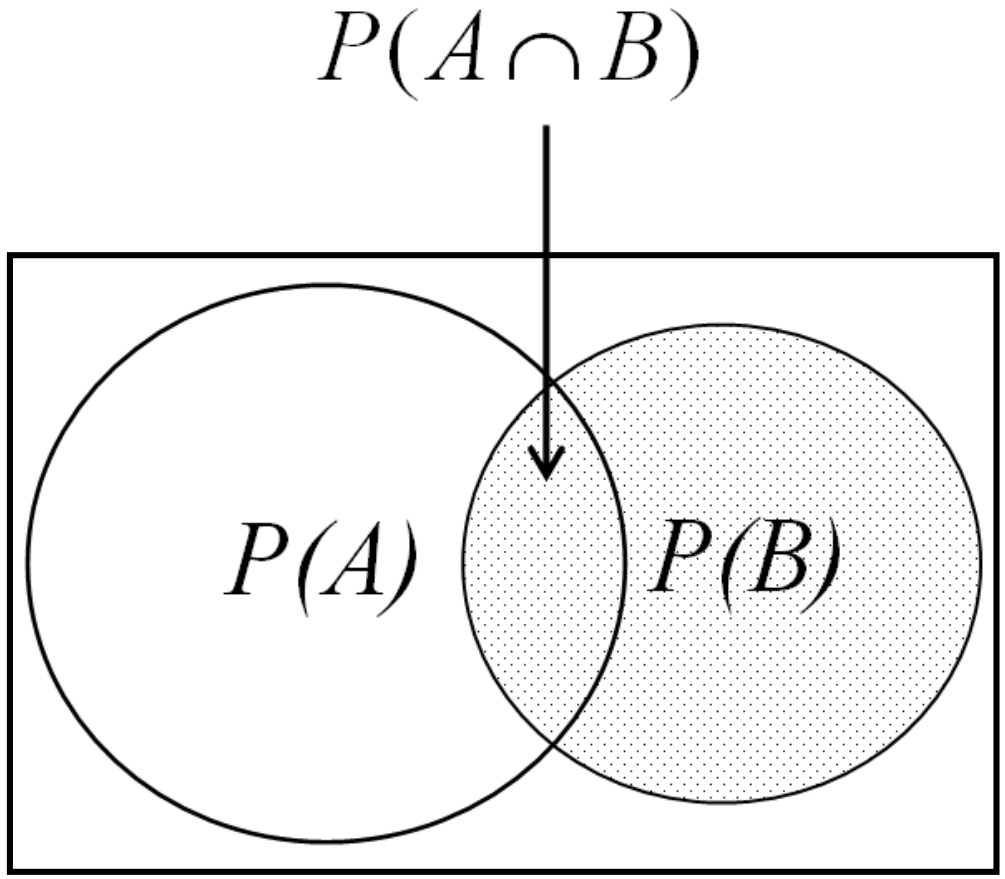
$$p(A|B) = \frac{p(A \cap B)}{p(B)}$$

$$p(B|A) = \frac{p(A \cap B)}{p(A)}$$

$$p(A|B)p(B) = p(A \cap B)$$

$$p(B|A)p(A) = p(B \cap A)$$

Bayes theorem is extremely simple to derive



$$p(A|B) = \frac{p(A \cap B)}{p(B)}$$

$$p(B|A) = \frac{p(A \cap B)}{p(A)}$$

$$p(A|B)p(B) = p(A \cap B)$$

$$p(B|A)p(A) = p(B \cap A)$$

$$\Longrightarrow \; p(A|B) = \frac{p(B|A)p(A)}{p(B)}, \quad p(B|A) = \frac{p(A|B)p(B)}{p(A)}$$

Understanding Bayes' theorem can be more subtle

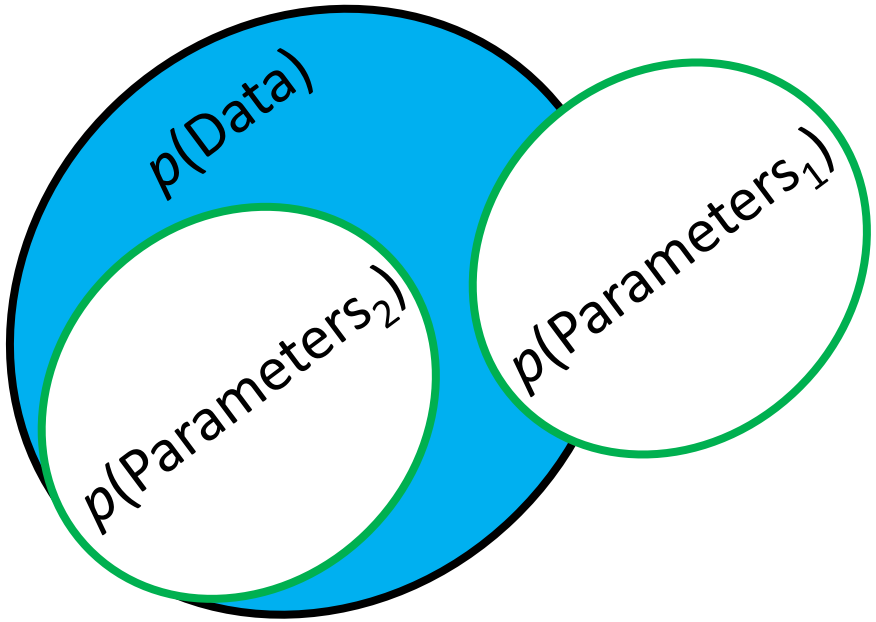
$$p(\text{parameters}|\text{data}) = \frac{p(\text{data}|\text{parameters})p(\text{parameters})}{p(\text{data})}$$

$$p(\text{parameters}|\text{data}) = \underbrace{\frac{p(\text{data}|\text{parameters})}{p(\text{data})}}p(\text{parameters})$$

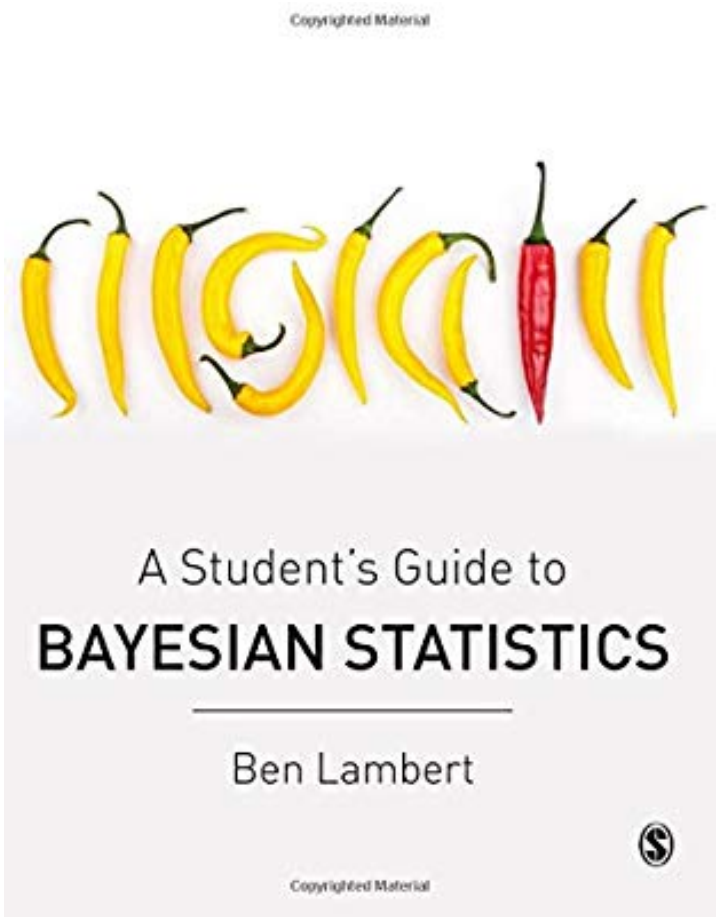
This quotient quantifies the explanatory power of parameter values relative to others

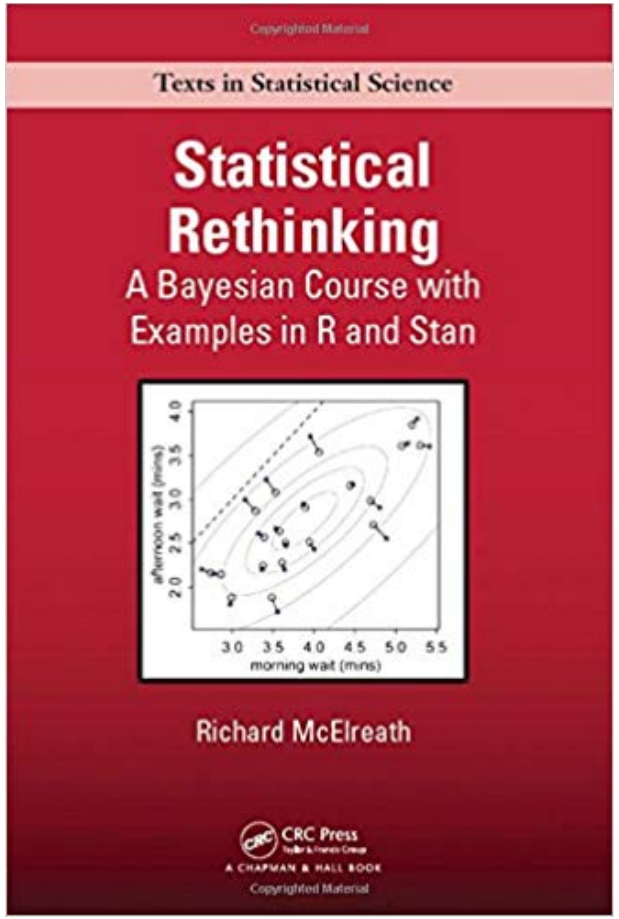
It is a normalized and therefore 'weights' the parameters according to how well they fit the data

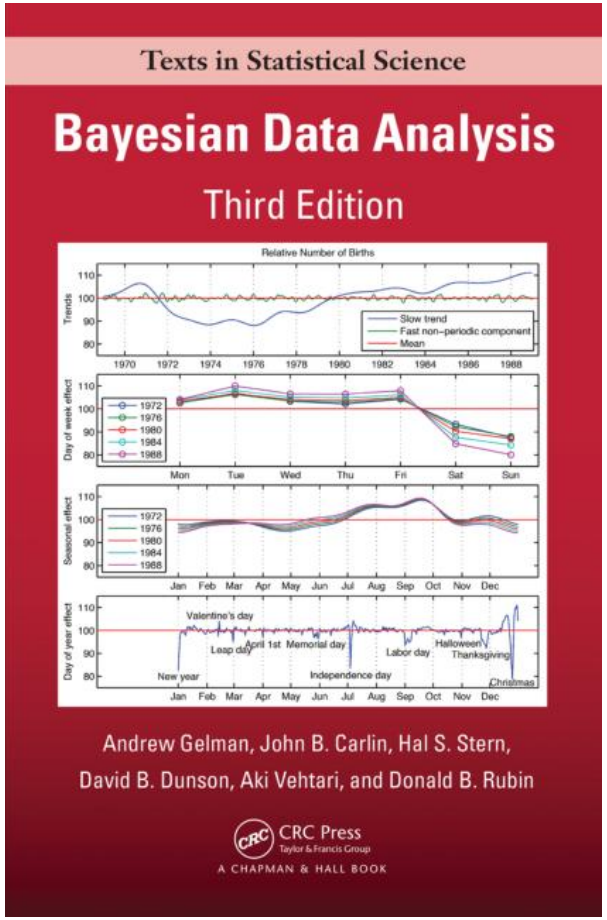
The product of the prior and relative goodness of fit weights gives us the posterior



Bayesian resources







MIKE SLIDES

Introducing Stan

- A sophisticated C++ software package for performing efficient MCMC that can be interfaced with R, Python, Julia, Stata, Mathematica, Matlab, or called from command line
  - R and Python best supported, followed by Command Line, increasing support for Julia, less well supported for others (version cross-compatibility, etc.)

- Write your model and specify parameters in the C++ template (i.e. the 'Stan language) and Stan will do the rest
  - Stan code identical when called from various languages, so easy to share

- Implements a version of 'Hamiltonian Monte Carlo' which has many practical advantages and is much faster than random walk MH

Stan Modeling Language

User's Guide and Reference Manual

Stan Development Team

Stan Version 2.8.0

Tuesday 8th September, 2015

mc-stan.org

About Stan

Stan is a state-of-the-art platform for statistical modeling and high-performance statistical computation. Thousands of users rely on Stan for statistical modeling, data analysis, and prediction in the social, biological, and physical sciences, engineering, and business.

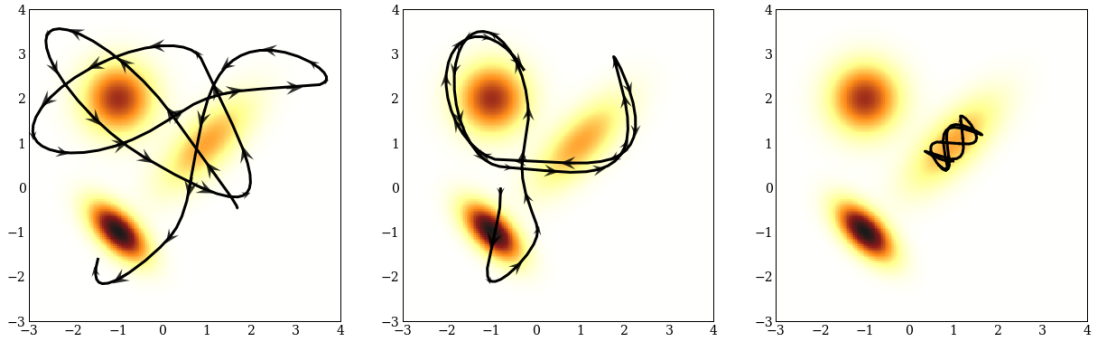
Users specify log density functions in Stan's probabilistic programming language and get:

- full Bayesian statistical inference with MCMC sampling (NUTS, HMC)
- approximate Bayesian inference with variational inference (ADVI)
- penalized maximum likelihood estimation with optimization (L-BFGS)

Stan's math library provides differentiable probability functions & linear algebra (C++ autodiff). Additional R packages provide expression-based linear modeling, posterior visualization, and leave-one-out cross-validation.

A brief and relatively uninformed description of Hamiltonian Monte Carlo

$$\frac{\mathrm{d}\mathbf{p}}{\mathrm{d}t} = -\frac{\partial \mathcal{H}}{\partial \mathbf{q}} \qquad \frac{\mathrm{d}\mathbf{q}}{\mathrm{d}t} = \frac{\partial \mathcal{H}}{\partial \mathbf{q}}$$



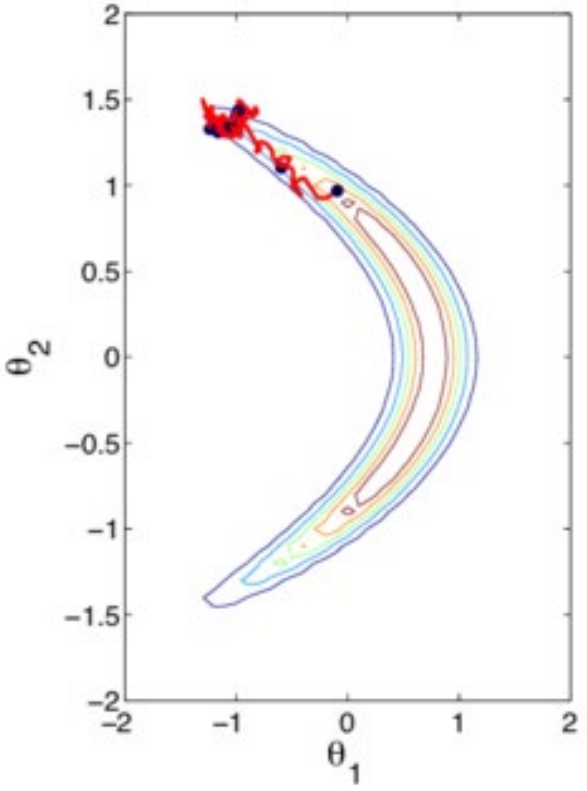
- Hamilton's equations are a reformulation of Newton's laws into a phase space of generalized coordinates
- The Hamiltonian describes the sum of potential and kinetic energies
- The Hamiltonian dynamics in phase space describe the time evolution of a classical mechanical system

- In HMC, we describe the potential energy as the negative log(posterior) and choose a Gaussian function for the kinetic energy
- Integrate the Hamiltonian forward in time to traverse the posterior
- Integration time and variance of the kinetic energy determine the efficiency of the exploration

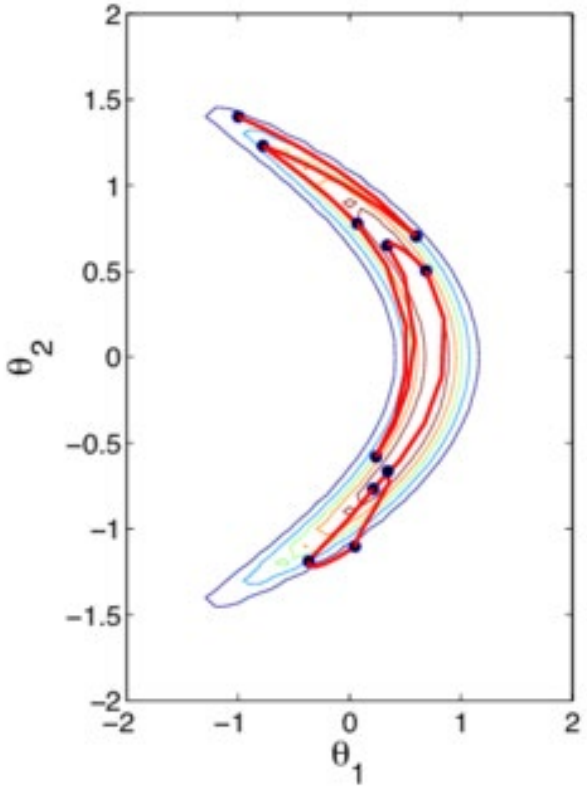
- Due to compiling time and tuning operations, Stan can (sometimes) be slower than RWMH for simple, very low dimensional models
- Speed and efficiency increase rapidly with the dimension of parameter space to explore

Sampling efficiency of HMC

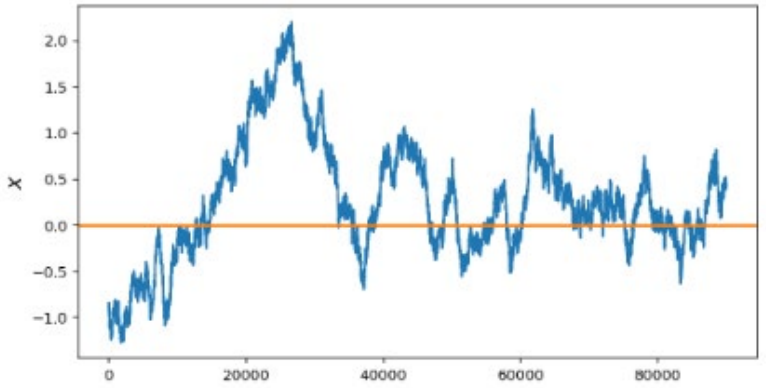
Random walk MCMC



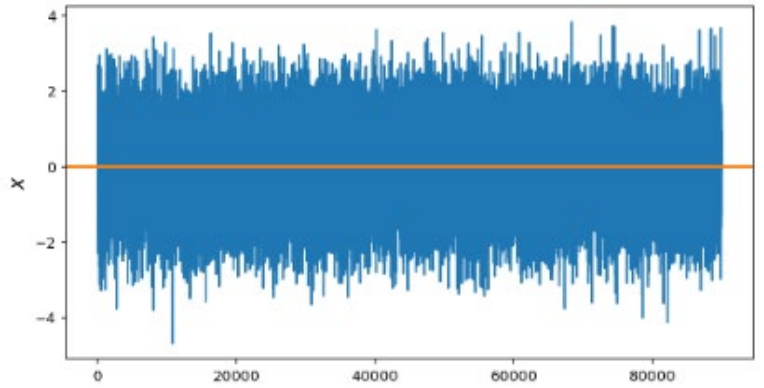
Hamiltonian MC



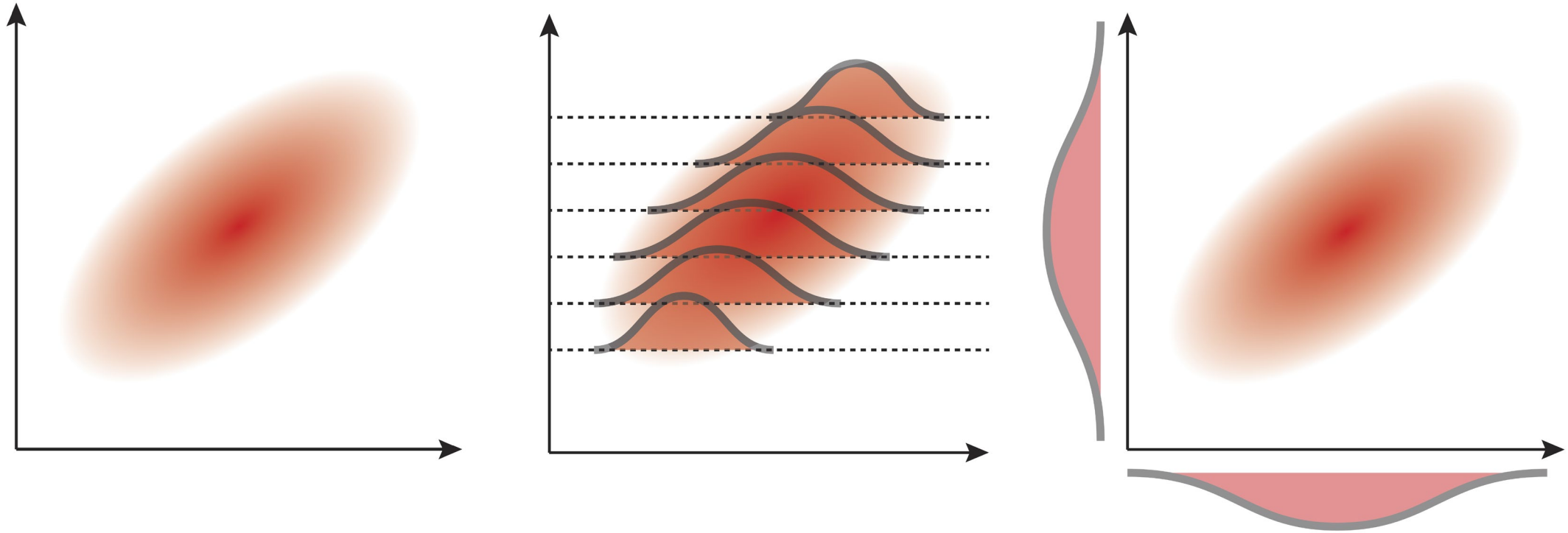
Random walk MCMC chain



Hamiltonian MC chain



Joint, conditional, and marginal probability distributions



In general we deal with multidimensional joint posterior probability distributions, characterized by a large sample-based approximation (i.e. histograms)

MCMC builds up the joint distribution from repeatedly sampling from conditionals

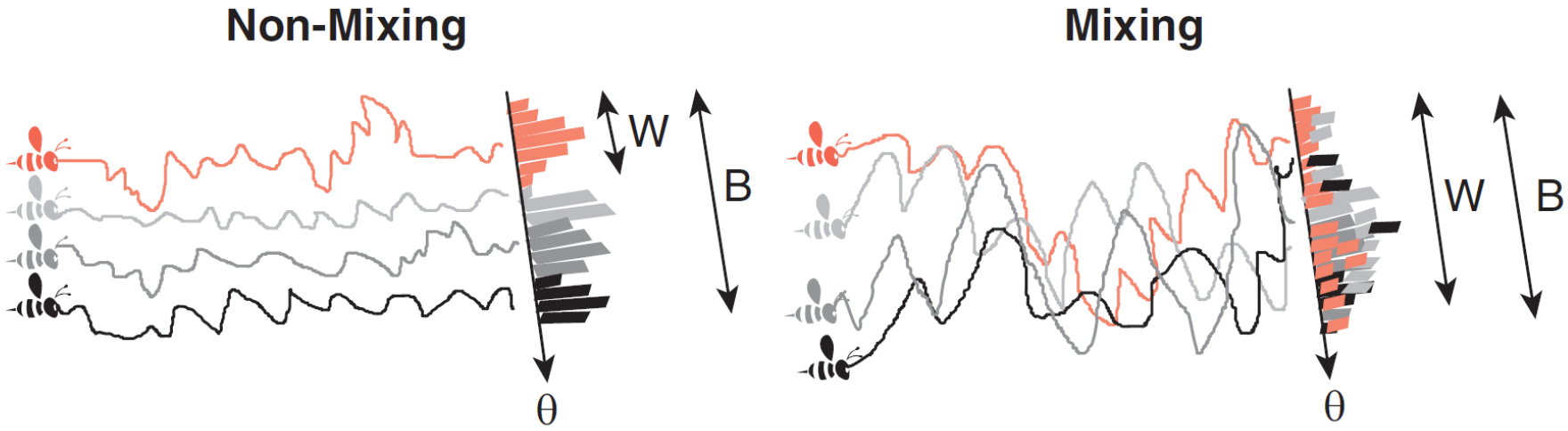
We can characterize distributions or do posterior calculations anyway we want once we have the full joint distribution

LINEAR REGRESSION

Explaining *Rhat*

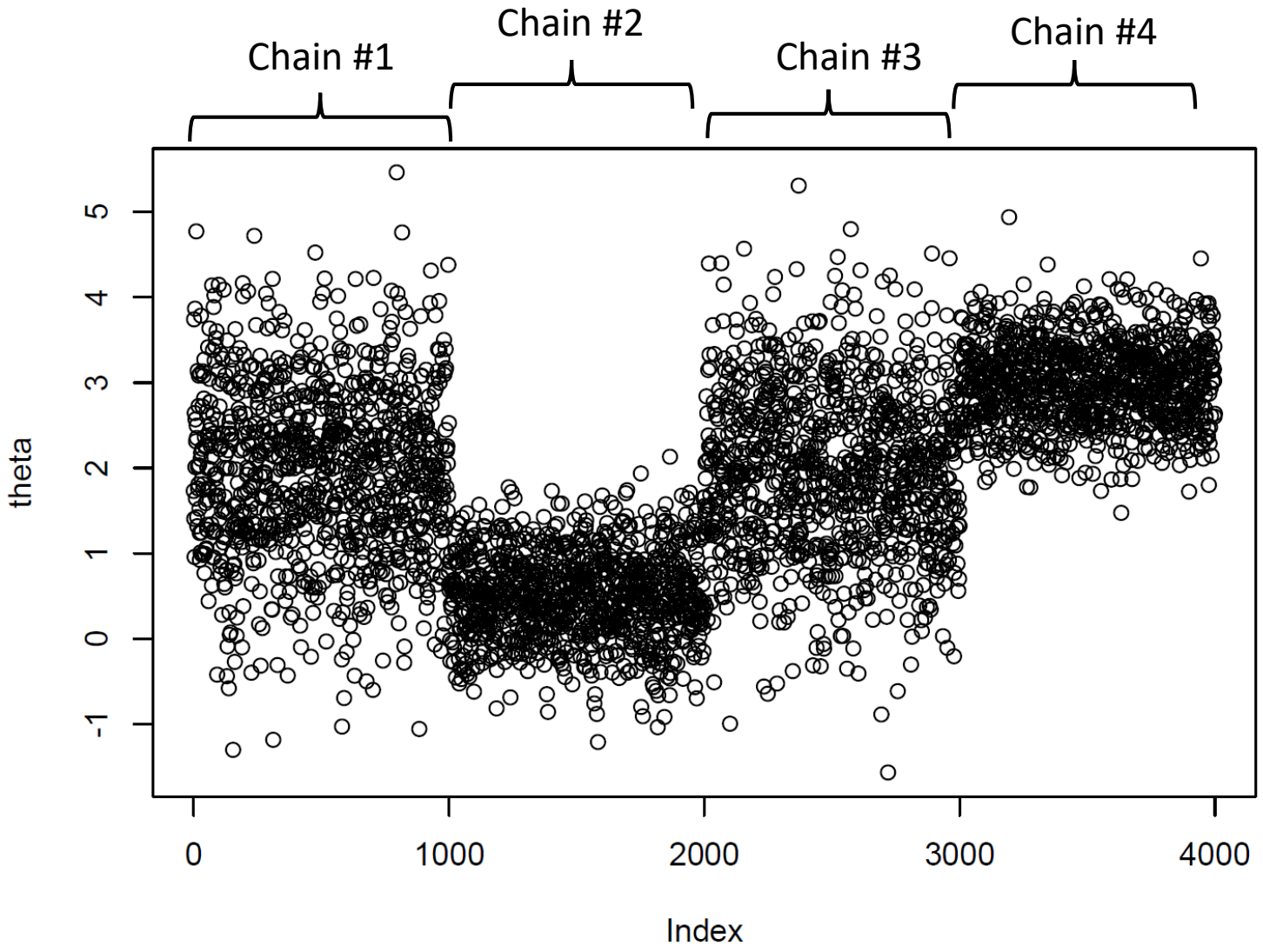
$$s_j^2 = \frac{1}{n-1}\sum_{i=1}^{n}(\theta_{ij} - \bar{\theta}_j)^2 \qquad B = \frac{n}{m-1}\sum_{j=1}^{m}(\bar{\theta}_j - \bar{\theta})^2$$

$$W = \frac{1}{m}\sum_{j=1}^{m} s_j^2 \qquad \hat{R} = \sqrt{\frac{W + \frac{1}{n}(B - W)}{W}}$$



Rhat is equivalent to an ANOVA on within vs. between chain variability -> a chain has converged (with high probability) if the within chain variability accounts for all the between chain variability (i.e. Rhat -> 1)

What unmixed chains look like



You should *always* run multiple chains with different initial conditions when doing MCMC

Poor mixing is usually due to too small MCMC step size

Explaining *n_eff*

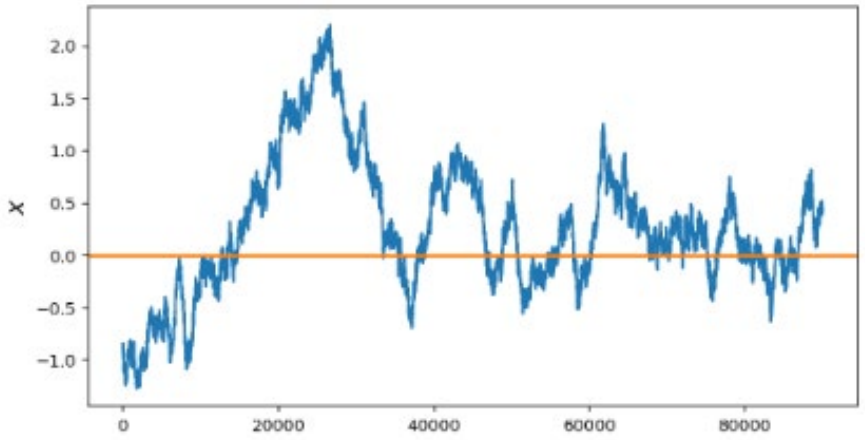
samples in chain

$$n_{eff} = \sum_{m} \frac{n}{1 + 2\sum_{\tau=1}^{\infty} \rho_m(\tau)}$$

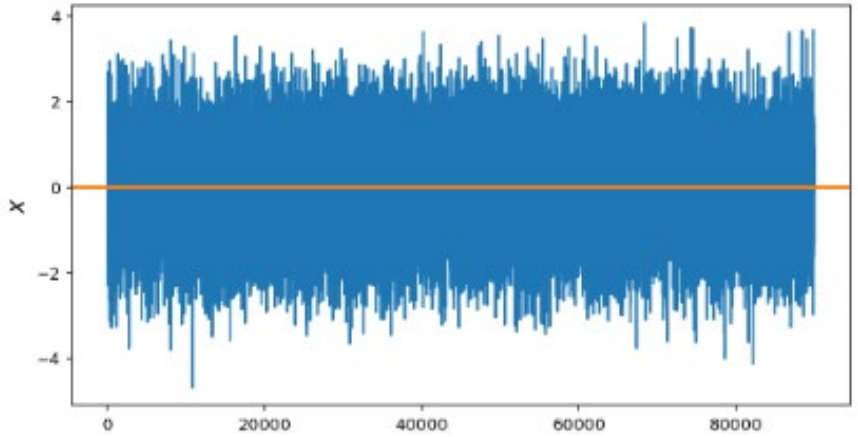
chains

Autocorrelation of chains

Random walk MCMC chain



Hamiltonian MC chain



http://mc-stan.org/misc/warnings.html

Brief Guide to Stan's Warnings

Stan Development Team

2019-07-11


```
1: There were 15 divergent transitions after warmup. Increasing adapt_delta above 0.8 may help.
2: Examine the pairs() plot to diagnose sampling problems
```

```
Exception thrown at line 24: normal_log: Scale parameter is 0, but must be positive!
```

```
## Warning: There were 1 chains where the estimated Bayesian Fraction of Missing Information was low. See
## http://mc-stan.org/misc/warnings.html#bfmi-low
```

```
## Warning: There were 2396 transitions after warmup that exceeded the maximum treedepth. Increase max_treedepth above 10. S
ee
## http://mc-stan.org/misc/warnings.html#maximum-treedepth-exceeded
```

```
## Warning: The model has not converged (some Rhats are > 1.1). Do not analyse the results!
## We recommend running more iterations and/or setting stronger priors.
```

Group exercise

- Read in the dataset ‘x3y.csv’ from the data folder within the linear regression case study folder. This dataset contains 3 variables: a simulated ‘x’, its square ‘x2’, and its cube ‘x3’
- Modify the Stan code to read these three variables
- Modify the Stan code to include a quadratic and cubed term in the regression
- Compile and fit the model and look at the results

```
stancode <- "
data {
    int       N;
    vector[N] x;
    vector[N] y;
}
parameters {
    real              beta0;
    real              beta1;
    real<lower=1E-15> sigma;
}
model{
    // Priors
    beta0 ~ normal(0,100);
    beta1 ~ normal(0,100);

    // Likelihood
    y ~ normal(beta0 + beta1*x, sigma);
}"
```

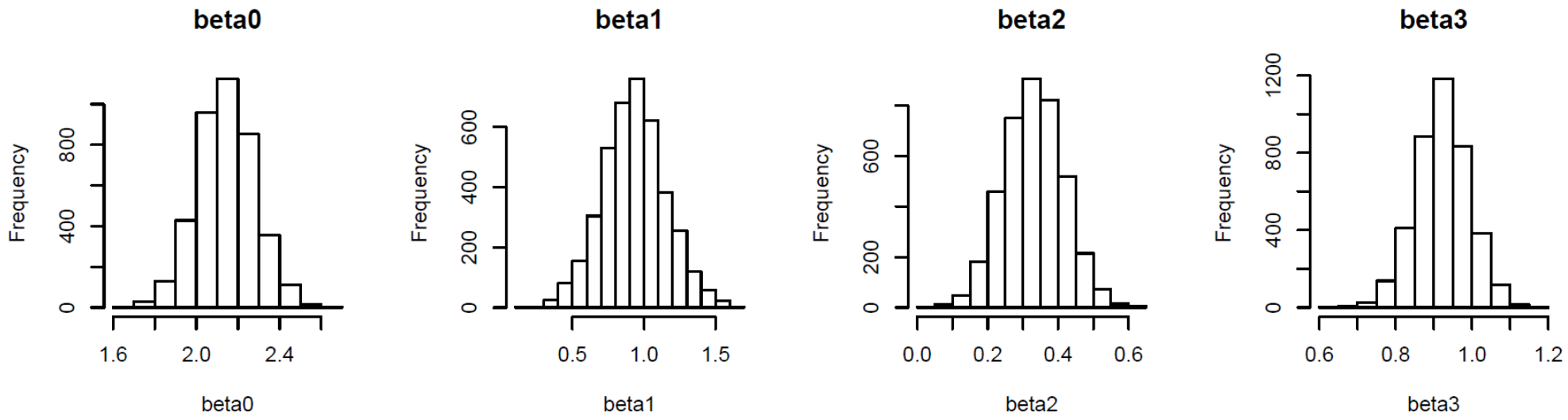
Need additional variables

Need additional parameters

Need additional regression terms

Results

```
Inference for Stan model: bf2873f840500d14abb94828b84dcf77.
4 chains, each with iter=2000; warmup=1000; thin=1;
post-warmup draws per chain=1000, total post-warmup draws=4000.

        mean se_mean   sd   2.5%    25%    50%    75%  97.5% n_eff Rhat
beta0   2.14    0.00 0.14   1.87   2.05   2.14   2.23   2.41  2873    1
beta1   0.93    0.00 0.22   0.49   0.78   0.93   1.07   1.39  2261    1
beta2   0.33    0.00 0.08   0.17   0.27   0.33   0.39   0.50  2685    1
beta3   0.92    0.00 0.07   0.78   0.88   0.92   0.97   1.06  2332    1
sigma   1.09    0.00 0.08   0.95   1.04   1.09   1.14   1.25  3374    1
lp__  -58.09    0.04 1.59 -61.96 -58.95 -57.77 -56.89 -55.99  1599    1

Samples were drawn using NUTS(diag_e) at Mon Jan 06 14:39:20 2020.
For each parameter, n_eff is a crude measure of effective sample size,
and Rhat is the potential scale reduction factor on split chains (at
convergence, Rhat=1).
```



PHYTOPLANKTON GROWTH MODEL

Group exercise

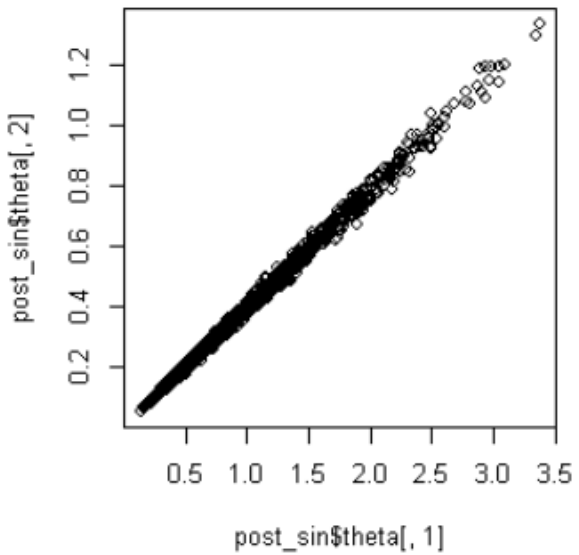
- Plot the joint posterior distribution of gamma, lambda, and x0
- Calculate the posterior correlation among these variables

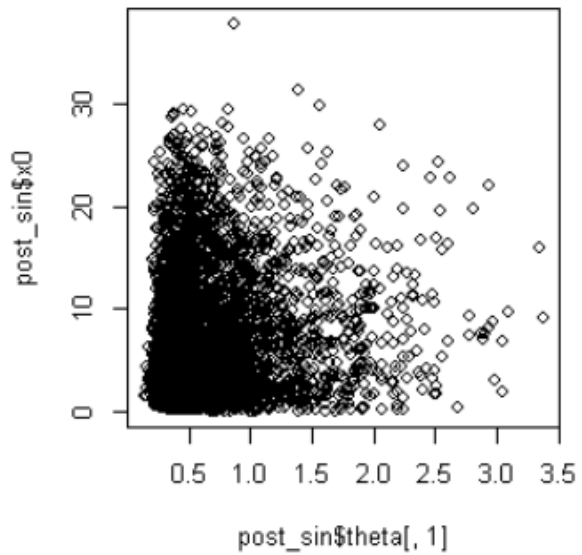
Results

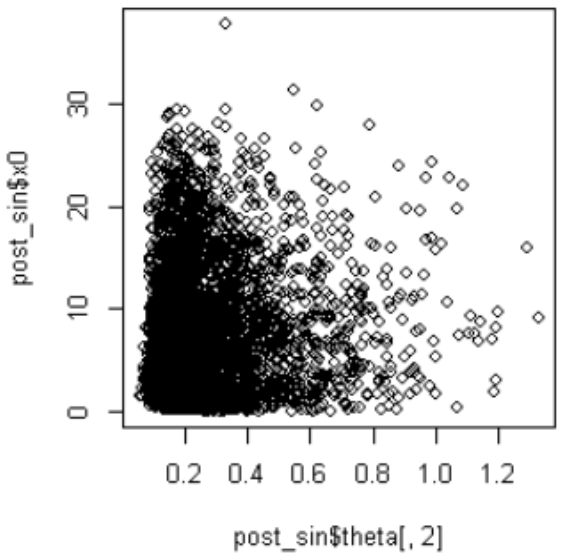
```
options(repr.plot.width=8, repr.plot.height=3)
par(mfrow=c(1,3))
plot(post_sin$theta[,1],post_sin$theta[,2])
plot(post_sin$theta[,1],post_sin$x0)
plot(post_sin$theta[,2],post_sin$x0)

data.frame(theta12= cor(post_sin$theta[,1],post_sin$theta[,2]),
           theta1x0=cor(post_sin$theta[,1],post_sin$x0),
           theta2x0=cor(post_sin$theta[,2],post_sin$x0))
```

A data.frame: 1 × 3

theta12	theta1x0	theta2x0
<dbl>	<dbl>	<dbl>
0.9986621	0.07251768	0.07361037







NUTRIENTS-PHYTOPLANKTON-ZOOPLANKTON

PAUL SLIDES

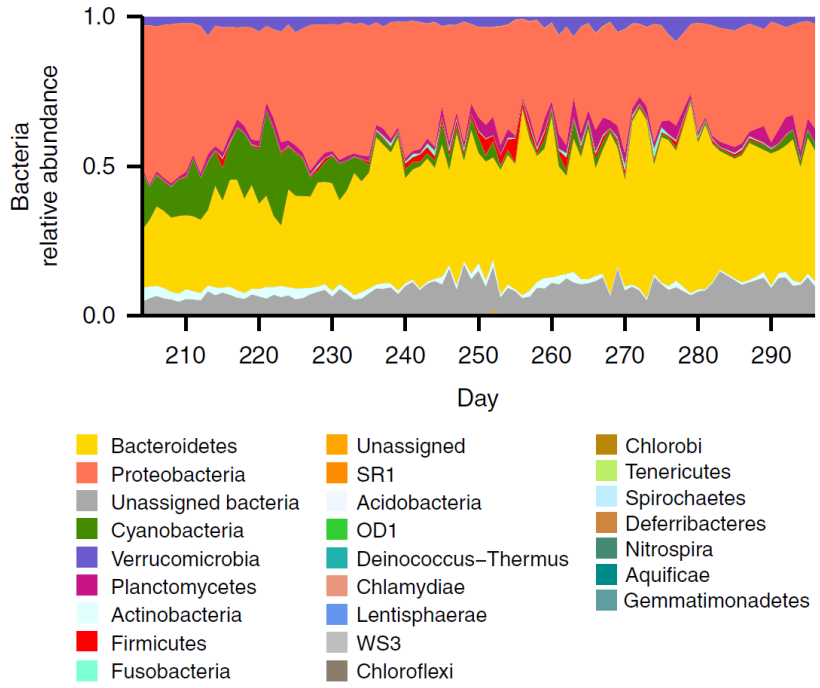
MULTIVARIATE FIRST ORDER AUTOREGRESSIVE MODEL MV AR(1)

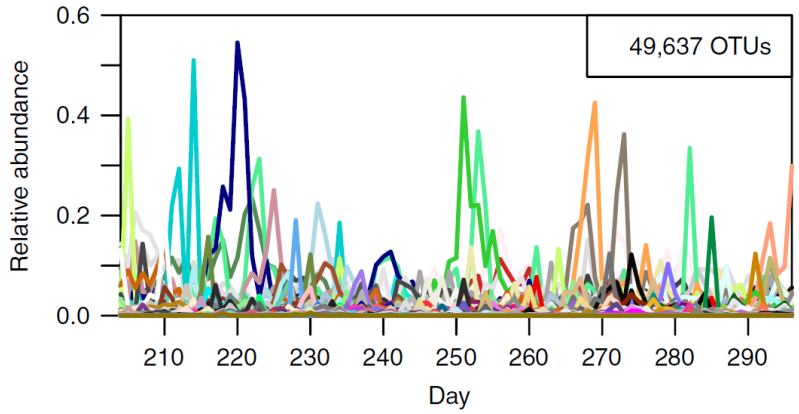
ARTICLE

DOI: 10.1038/s41467-017-02571-4 OPEN

High resolution time series reveals cohesive but short-lived communities in coastal plankton

Antonio M. Martin-Platero[1,6], Brian Cleary[2,3], Kathryn Kauffman[1], Sarah P. Preheim[1,7], Dennis J. McGillicuddy, Jr[4], Eric J. Alm[1,2,5] & Martin F. Polz[1]





Group exercise

```
mod_code_D_struc_2 <- "data {
    int T;          //length of time series
    int p;          //number of variables
    matrix[p,T] Y; //matrix of observations; variables are rows; time is columns
}
parameters{
    real phi_12;
    real phi_23;
    vector[p] phi_diag;
    vector<lower=1E-15>[p] sigma;  //variances of stochastic forcing
    vector[p] init;                //mean of initial conditions as parameter vector
}
transformed parameters{
    matrix[p,p] PHI;
    PHI[1,3] = 0;
    PHI[3,1] = 0;
    PHI[1,2] = phi_12;
    PHI[2,1] = phi_12;
    PHI[2,3] = phi_23;
    PHI[3,2] = phi_23;
    for(i in 1:p){
        PHI[i,i] = phi_diag[i];
    }
}
model{
    Y[,1] ~ normal(init, sigma);          //distribution of the initial conditions
    for(i in 2:T){
        Y[,i] ~ normal(PHI*Y[,i-1],sigma); //conditional predictive distribution
    }
}"
```

Use the transformed parameters block to build matrix operators from parameters (will be useful for SeaFlow model)

Can fit particular interaction structures, etc.

Try setting other parameters to zero, or specified values. See how the other estimates change

Results

```
Inference for Stan model: d78b8154621b0b092fcd3133a6b1526d.
4 chains, each with iter=2000; warmup=1000; thin=1;
post-warmup draws per chain=1000, total post-warmup draws=4000.
```

	mean	se_mean	sd	2.5%	25%	50%	75%	97.5%	n_eff	Rhat
phi12	-0.03	0.00	0.05	-0.12	-0.06	-0.03	0.00	0.06	7603	1
phi23	0.17	0.00	0.04	0.08	0.14	0.17	0.20	0.25	6735	1
phi_diag[1]	0.39	0.00	0.07	0.26	0.34	0.39	0.44	0.52	9152	1
phi_diag[2]	0.40	0.00	0.07	0.27	0.35	0.40	0.44	0.53	8074	1
phi_diag[3]	0.35	0.00	0.06	0.22	0.30	0.35	0.39	0.47	7162	1
sigma[1]	0.94	0.00	0.05	0.86	0.91	0.94	0.97	1.04	7255	1
sigma[2]	1.09	0.00	0.06	0.99	1.05	1.09	1.13	1.21	7555	1
sigma[3]	0.88	0.00	0.04	0.80	0.85	0.88	0.91	0.97	8906	1
init[1]	-0.95	0.01	0.95	-2.80	-1.61	-0.95	-0.29	0.85	8348	1
init[2]	-0.29	0.01	1.08	-2.39	-1.02	-0.29	0.45	1.86	8544	1
init[3]	1.34	0.01	0.90	-0.39	0.72	1.35	1.96	3.07	9291	1
PHI[1,1]	0.39	0.00	0.07	0.26	0.34	0.39	0.44	0.52	9152	1
PHI[1,2]	-0.03	0.00	0.05	-0.12	-0.06	-0.03	0.00	0.06	7603	1
PHI[1,3]	0.00	NaN	0.00	0.00	0.00	0.00	0.00	0.00	NaN	NaN
PHI[2,1]	-0.03	0.00	0.05	-0.12	-0.06	-0.03	0.00	0.06	7603	1
PHI[2,2]	0.40	0.00	0.07	0.27	0.35	0.40	0.44	0.53	8074	1
PHI[2,3]	0.17	0.00	0.04	0.08	0.14	0.17	0.20	0.25	6735	1
PHI[3,1]	0.00	NaN	0.00	0.00	0.00	0.00	0.00	0.00	NaN	NaN
PHI[3,2]	0.17	0.00	0.04	0.08	0.14	0.17	0.20	0.25	6735	1
PHI[3,3]	0.35	0.00	0.06	0.22	0.30	0.35	0.39	0.47	7162	1
lp__	-279.02	0.06	2.34	-284.54	-280.33	-278.72	-277.36	-275.39	1665	1

Group exercise (NPZ_narragansett.ipynb)

- Try fitting the dynamics of only the two most abundance phyla
  - Do you recover the same interactions for those phyla when other phyla are included?

OR

- Try grouping by a different taxonomic level and estimating the interactions
  - Note that the fitting will slow down as *p* increases

	A	B	C	D	E	F	G
1	OTU	ConsensusLineage	phylum	class	order	family	genus
2	MERGE33	k__Bacteria	p__Bacteroidetes	c__Flavobacteria	o__Flavobacteriales	f__Flavobacteriaceae	g__
3	MERGE441	k__Bacteria	p__Proteobacteria	c__Alphaproteobacteria	o__Rhodobacterales	f__Rhodobacteraceae	g__
4	MERGE70	k__Bacteria	p__Cyanobacteria	c__Cyanobacteria	o__	f__Family II	g__GpIIa
5	MERGE59	k__Bacteria	p__Bacteroidetes	c__Flavobacteria	o__Flavobacteriales	f__Flavobacteriaceae	g__Tenacibaculum
6	MERGE29	k__Bacteria	p__Proteobacteria	c__Gammaproteobacteria	o__	f__	g__
7	MERGE342	k__Bacteria	p__Bacteroidetes	c__Flavobacteria	o__Flavobacteriales	f__Flavobacteriaceae	g__
8	MERGE49	k__Bacteria	p__Bacteroidetes	c__Flavobacteria	o__Flavobacteriales	f__Flavobacteriaceae	g__Tenacibaculum
9	MERGE71	k__Bacteria	p__Cyanobacteria	c__Cyanobacteria	o__	f__Family II	g__GpIIa
10	MERGE231	k__Bacteria	p__	c__	o__	f__	g__
11	MERGE337	k__Bacteria	p__Bacteroidetes	c__Flavobacteria	o__Flavobacteriales	f__Flavobacteriaceae	g__
12	MERGE107	k__Bacteria	p__	c__	o__	f__	g__
13	MERGE154	k__Bacteria	p__Proteobacteria	c__Alphaproteobacteria	o__Rickettsiales	f__SAR11	g__Pelagibacter
14	MERGE153	k__Bacteria	p__Proteobacteria	c__Alphaproteobacteria	o__Rickettsiales	f__SAR11	g__Pelagibacter
15	MERGE71	k__Bacteria	p__Cyanobacteria	c__Cyanobacteria	o__	f__Family II	g__GpIIa
16	MERGE666	k__Bacteria	p__Proteobacteria	c__Gammaproteobacteria	o__	f__	g__
17	MERGE537	k__Bacteria	p__Proteobacteria	c__Gammaproteobacteria	o__	f__	g__
18	MERGE53	k__Bacteria	p__Proteobacteria	c__Gammaproteobacteria	o__Thiotrichales	f__Piscirickettsiaceae	g__Methylophaga

Results

MACROMOLECULAR MODEL

INSERT AW SLIDES

Group exercise

- Try fixing some parameters and analyzing the impact on the fit

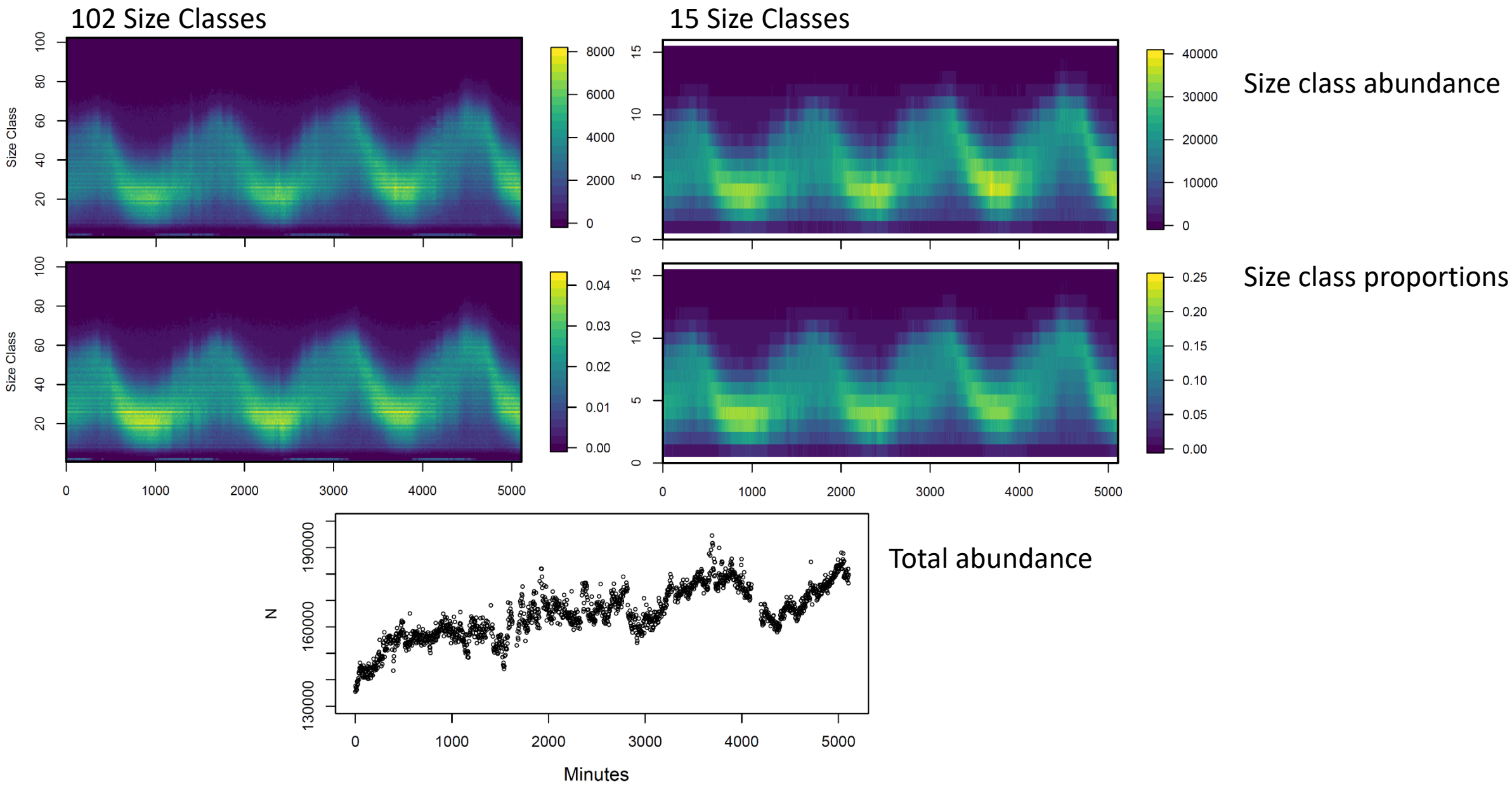
```
mod_code2 <- "functions {
  real[] macro(real   t,           // time
               real[] x,           // state x[1]:CH  x[2]:PR, x[3]:Chl , x[4]:N
               real[] theta,
               real[] x_r,
               int[]  x_i) {       // parameters

    // real KN    = theta[1];
    real mu    = theta[1];
    real CHsyn = theta[2];
    real m_ex  = theta[3];
    real R_ex  = theta[4];
    real tau   = theta[5];
    real b     = theta[6];

    real PRsynth = theta[1]*x[4]/(x_r[1]+x[4]);
    real r0      = theta[6]*(x[2]/x[1]);
    real Chl     = x[3]*x[2];
    real Rcell   = x[1]/x[2];
    real excr    = (1/2)*theta[3]*(1+tanh(Rcell - theta[4]));

    real dCH    = x[2]*(theta[2] - excr);
    real dr     = (1/theta[5])*(r0-x[3]);
    real dPR    = x[2]*PRsynth;
    real dN     = -dPR/(1+exp(-10000*x[4]));

    return {dCH,dPR,dr,dN};
  }
}
```

```
transformed parameters {
  real x[n,4] = integrate_ode_rk45(macro, x0, -1, t_obs, theta, {0.02}, rep_array(0,0),
    1e-6, 1e-5, 1e3) ;
  for(i in 1:n){
    x[i,3] = x[i,3]*x[i,2]*1E6;
  }
}
```

Results

SEAFLOW MATRIX POPULATION MODEL

SeaFlow Data



INSERT FRANCOIS SLIDES

INSERT PAUL SLIDES

GROUP HACKATHON BEGINS!

1. Join the case study you'd like to work on
    1. Group Leaders:
       NPZ + Pgrowth (Mike)
       MV AR(1) (Greg)
       SeaFlow (Francois/Paul)
       Macromolecular (AW)
    2. Brainstorm analysis ideas
       What is your scientific question?
       How will you use or modify a model?
       How will you assign prior distributions to model parameters?
       How will you assess the fit of the model?
       What analysis does your question require? Parameter estimation? Prediction? Model selection?

2. Come back together to present and discuss ideas as a group

Some hackathon ideas to get you started

One compartment P growth model

NPZ model

- Test different forms for the seasonal nutrient supply
- Try fitting the univariate growth model to the P time series

Multivariate AR(1)

- Evaluate the strength of interactions between and within phyla
- Compute the distribution of matrix eigenvalues

Macromolecular model

- Functional form for nutrient uptake
- Model the diel cycle

Matrix population model

- Fit the model to Zinser data (FCS)
- Incorporate population size into
- Examine the importance of model resolution
- Different error distributions (Dirichlet, normal-approx. to bernouli, effective sample size)