A crash-course introduction to Bayesian approaches for TPC estimation

Mauricio Cruz Loya

Before we start

There will be a brief coding exercise at the end of the talk.

Code:
https://github.com/mcruzloya/bayesTPC

If you want to follow along with your own computer, please make sure you have the following installed (links in Github):

JAGS (Just Another Gibbs Sampler)

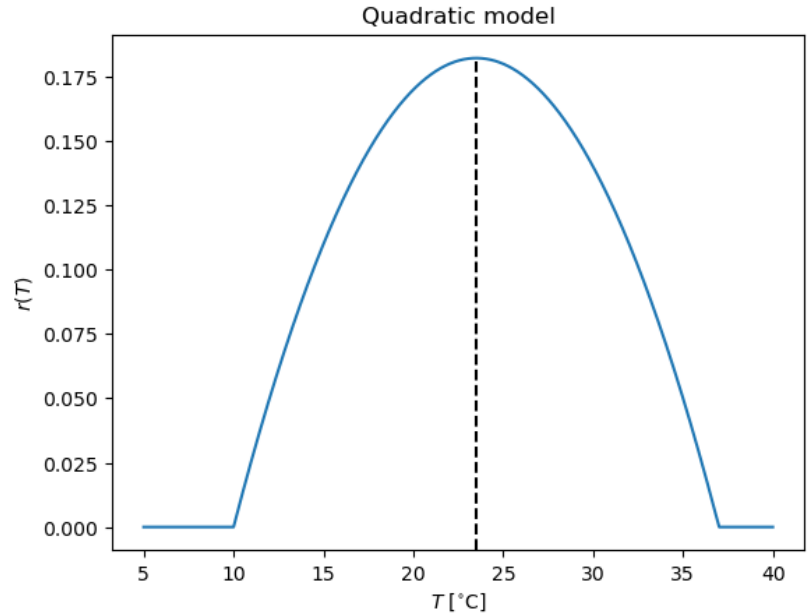
R packages

r2jags
mcmcplots
MCMCvis

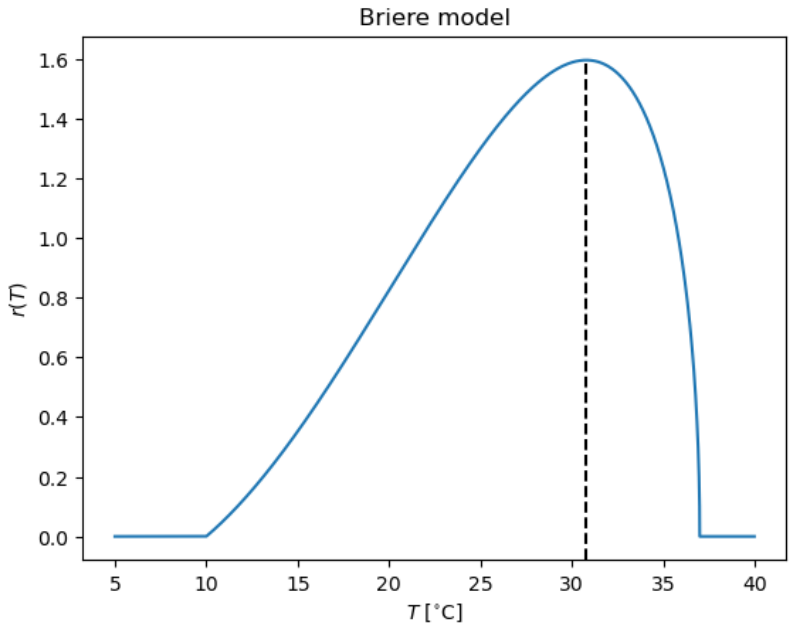
Temperature performance curves

Thermal performance curves (TPCs) model how the performance of a living organism (or some life-history trait) varies with temperature.

Many different mathematical models can be used for a TPC. A couple common examples previously used in the lab are:



$$r(T) = \begin{cases} c(T - T_{\min})(T_{\max} - T) & T_{\min} < T < T_{\max} \\ 0 & \text{otherwise} \end{cases}$$



$$r(T) = \begin{cases} cT(T - T_{\min})\sqrt{T_{\max} - T} & T_{\min} < T < T_{\max} \\ 0 & \text{otherwise} \end{cases}$$

Step 1: Pick a suitable TPC model for your data

- Does the data you want to fit look roughly symmetric or not?

- Has similar data been analyzed before? What kind of model was used then, and does it look like it fits well?

- If multiple candidate models look feasible, we can fit various models and choose one based on statistical model selection criteria (e.g. DIC or cross-validation).

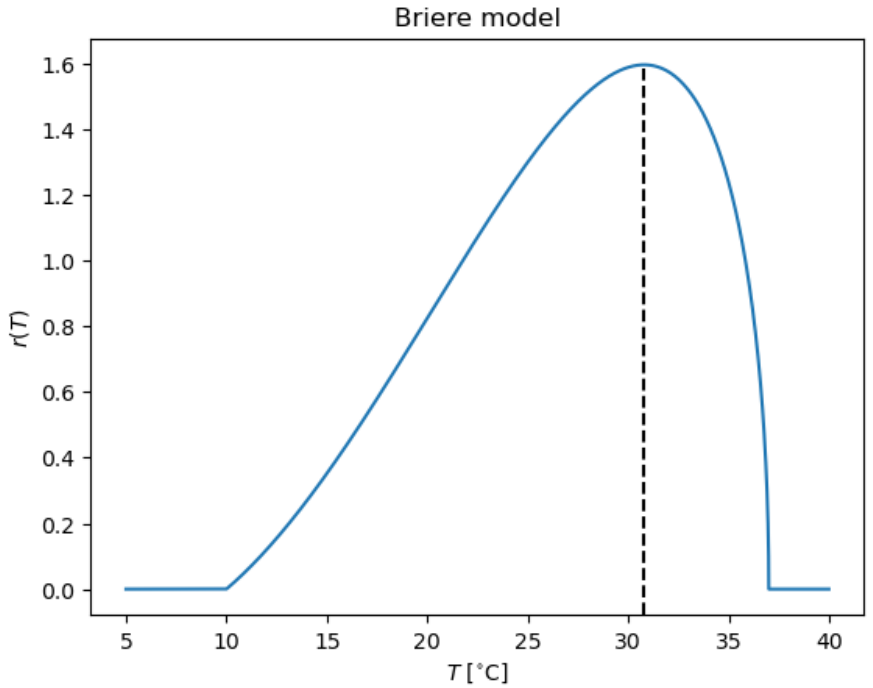
Models and parameters

Briere model

$$r(T) = \begin{cases} cT(T - T_{\min})\sqrt{T_{\max} - T} & T_{\min} < T < T_{\max} \\ 0 & \text{otherwise} \end{cases}$$

Parameters: c, T_{min}, T_{max}

Different parameters will change the shape of the curve.



When fitting a TPC, we want to find the model parameters that best describe our data (and our previous biological knowledge).

Bayesian approaches

Statistical model

By themselves, TPC models are *deterministic*. That is, they predict a single performance/trait value at each temperature.

However, real data will have variability due to both population heterogeneity and experimental error.

Because of this, we need a statistical model to account for variability in the data. We usually do this with a *nonlinear regression*.

Statistical model

$$r(T) = \begin{cases} cT(T - T_{\min})\sqrt{T_{\max} - T} & T_{\min} < T < T_{\max} \\ 0 & \text{otherwise} \end{cases}$$

For example:

Mean is assumed to be given by TPC model.

$$y_i | T_i, P, \sigma^2 \sim \text{Normal}(r(T_i; P), \sigma^2)$$

Performance at i-th datapoint

Temperature at i-th datapoint

Parameters from TPC model.

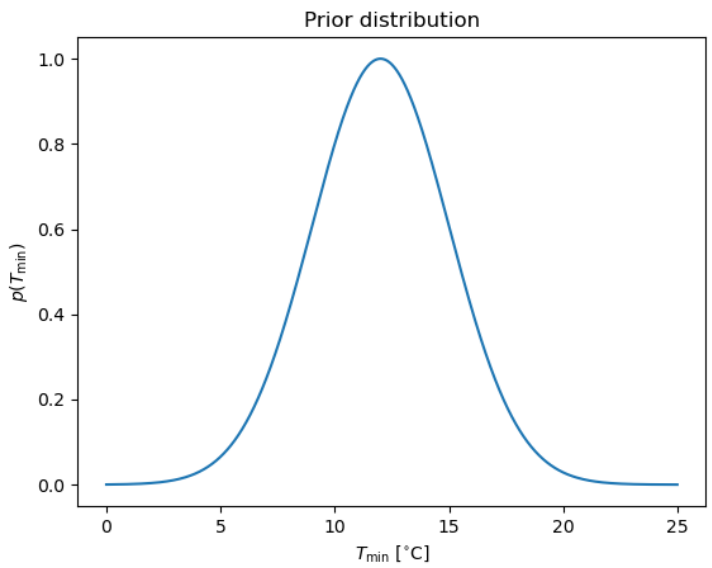
$$P = \{T_{\min}, T_{\max}, c\}$$

Variance parameter

This is only a simple example: different models are possible (e.g. different distributions, nonconstant variance, etc.).

Bayesian statistics in a nutshell

In Bayesian statistics, we describe our uncertainty about model parameters as a probability distribution.



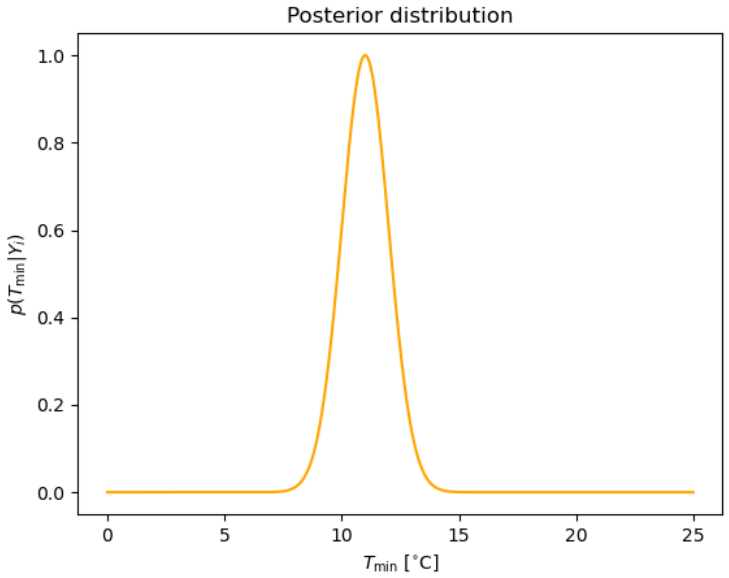
Parameter values assumed to be likely (or not) before we see the data.

Can be based on:

- Biological knowledge.
- Previous experiments.
- Can be diffuse if we know little about parameter.

Data

Bayes's Theorem



Posterior distribution, which describes our updated uncertainty on the parameter values *after* seeing the data.

Bayes's Theorem

Previous biological knowledge → Probability of parameter values **before** seeing the data.

Probability of seeing the data given parameter values. ← Data

prior **likelihood**

$$p(\theta|Y) = \frac{p(\theta)p(Y|\theta)}{p(Y)}$$

posterior

Probability of parameter values **after** seeing the data.

marginal data distribution

Overall probability of seeing the data.

A Bayesian statistical model

In a Bayesian analysis, the **first step** is to specify a full probability model of observations and *all* model parameters.

likelihood

$$y_i | T_i, P, \sigma^2 \sim \text{Normal}(r(T_i; P), \sigma^2)$$

prior

$$P \begin{cases} T_{\text{min}} \sim \ldots \\ T_{\text{max}} \sim \ldots \\ c \sim \ldots \end{cases}$$

$$\sigma^2 \sim \ldots$$

TPC model parameters

Variance parameter

Suitable TPC model for your data.

(we will use Briere in this tutorial).

How to choose prior distributions?

Priors can have various degrees of “informativeness”. The more informative a prior distribution, the more impact it has on the result of the analysis (the posterior distribution).

- “Uninformative” priors: Diffuse priors that place very few constraints in the model parameters. The idea is to let the data have as much influence in the posterior distribution as possible.

e.g. $T_{\max} \sim \text{Uniform}(20, 100)$ $T_{\max} \sim \text{Normal}(\mu = 35, \sigma = 20)$ 95% likely to be in (-5, 75)

Note: These priors are used merely as examples to illustrate the difference between approaches. You should think carefully about what priors to use in your own problem!

How to choose prior distributions?

Priors can have various degrees of “informativeness”. The more informative a prior distribution, the more impact it has on the result of the analysis (the posterior distribution).

- “Uninformative” priors: Diffuse priors that place very few constraints in the model parameters. The idea is to let the data have as much influence in the posterior distribution as possible.

  e.g. $T_{\max} \sim \text{Uniform}(20, 100)$ $T_{\max} \sim \text{Normal}(\mu = 35, \sigma = 20)$ 95% likely to be in (-5, 75)

- Weakly informative priors: Purpose is to keep inferences in a reasonable range (regularization) but let the data do the heavy lifting.

  e.g. $T_{\max} \sim \text{Normal}(\mu = 35, \sigma = 10)$ 95% likely to be in (15, 55)

Note: These priors are used merely as examples to illustrate the difference between approaches. You should think carefully about what priors to use in your own problem!

How to choose prior distributions?

Priors can have various degrees of “informativeness”. The more informative a prior distribution, the more impact it has on the result of the analysis (the posterior distribution).

- “Uninformative” priors: Diffuse priors that place very few constraints in the model parameters. The idea is to let the data have as much influence in the posterior distribution as possible.

  e.g. $T_{\max} \sim \text{Uniform}(20, 100)$ $T_{\max} \sim \text{Normal}(\mu = 35, \sigma = 20)$ 95% likely to be in (-5, 75)

- Weakly informative priors: Purpose is to keep inferences in a reasonable range (regularization) but let the data do the heavy lifting.

  e.g. $T_{\max} \sim \text{Normal}(\mu = 35, \sigma = 10)$ 95% likely to be in (15, 55)

- Strongly informative priors: Place stronger constraints on model parameters. Prior will have greater effect in inference.

  e.g. $T_{\max} \sim \text{Normal}(\mu = 35, \sigma = 2)$ 95% likely to be in (31, 39)

Note: These priors are used merely as examples to illustrate the difference between informative/noninformative priors. You should think carefully about what priors to use in your own problem!

How to choose prior distributions?

Good sources of information to use in prior distributions:

- Previous experiments.
- Knowledge about biology of organism (e.g. does not live below freezing temperatures).

Always avoid:

- Picking prior distribution based on eyeballing the dataset we're fitting.

For Bayesian inference to do its job, the prior should not be informed by the data we're fitting. Otherwise we run the risk of being overconfident in our knowledge of the parameter.

Reporting prior distributions

- When writing up analysis, there should be a section on how prior distributions were chosen that is explicit about the assumptions taken and cite any relevant references (e.g. literature or previous experiments).

- However, if prior information is available, it is OK to use it and incorporate it in the prior distribution. But we need to be explicit about our modeling choices.

Probabilistic programming languages

JAGS model

```
model{

## Priors
c ~ dunif(0, 0.01)
Tmin ~ dunif(0, 20)
Tmax ~ dunif(28, 45)
sigma ~ dunif(0, 0.5)
tau <- 1 / (sigma * sigma)

## Likelihood
for(i in 1:N.obs){
  trait.mu[i] <- c * temp[i] * (temp[i] - Tmin) * sqrt((Tmax - temp[i]) * (Tmax > temp[i])) * (Tmin < temp[i])
  trait[i] ~ dnorm(trait.mu[i], tau)
}

## Derived Quantities
Topt = (4*Tmax + 3*Tmin)/10 + sqrt(4 * Tmax^2 + (9/4) * Tmin^2 - 4*Tmax*Tmin) / 5

} # close model
```

Briere model

Topt formula for Briere model.

Steps in Bayesian analysis

1. Set up a full probability model.
   - Choose TPC model for performance/trait mean.
   - Choose statistical model for data (which distribution, variance assumptions).
   - Choose prior distributions for all model parameters.

2. Find posterior distribution.

3. Evaluate model fit and implications of posterior distribution. Might alter/expand model and go back to step 1.

Steps in Bayesian analysis

1. Set up a full probability model.
   - Choose TPC model for performance/trait mean.
   - Choose statistical model for data (which distribution, variance assumptions).
   - Choose prior distributions for all model parameters.

2. Find posterior distribution.

3. Evaluate model fit and implications of posterior distribution. Might alter/expand model and go back to step 1.

Finding the posterior distribution

Bayes's Theorem

prior

We know this (we set prior distributions)

likelihood

We know this (from statistical model)

posterior

We want to find this.

$$p(\theta|Y) = \frac{p(\theta)p(Y|\theta)}{p(Y)}$$

marginal data distribution

We don't know this.

Problem:

Except for some very specific cases, this is impossible to find analytically.

Finding the posterior distribution

One main problem in finding the posterior distribution is that we need to find the following integral:

marginal data distribution

Overall probability of seeing the data.

$$p(Y) = \int p(Y|\theta)p(\theta)d\theta$$

We need to integrate over all parameters in the model. This is impossible to do analytically for most models!

Because of this, we need a way to approximate the posterior distribution instead of calculating it directly.

How to find posterior distribution

We can use **Markov Chain Monte Carlo (MCMC)** methods to approximate the posterior distribution. There's many different variants of MCMC.

In a nutshell:

MCMC methods explore the parameter space of our model in a clever way so that *in the long run* we will draw samples from the posterior distribution of our model.

An example MCMC chain

	Tmin	Tmax	c	sigma
[1,]	2.81671	39.93408	0.00005	0.01304
[2,]	9.88984	36.05754	0.00008	0.01423
[3,]	7.90026	39.41115	0.00006	0.01390
[4,]	8.41343	33.27025	0.00009	0.01536
[5,]	3.88351	36.13755	0.00006	0.02565
[6,]	9.90926	33.69211	0.00011	0.03257
[7,]	9.05633	37.36933	0.00007	0.01082
[8,]	0.46384	35.80023	0.00004	0.02032
[9,]	3.01310	37.99319	0.00005	0.00892
[10,]	7.30570	38.19713	0.00005	0.00633
[11,]	9.50733	37.24365	0.00007	0.00656
[12,]	6.96093	41.31859	0.00004	0.01751
[13,]	9.28582	35.49384	0.00007	0.00360
[14,]	13.91484	34.04061	0.00013	0.05249
[15,]	12.59842	41.64609	0.00007	0.04088
[16,]	4.59222	43.14088	0.00004	0.00642

We start at some initial parameter values.

Based on the current values, the MCMC algorithm will then pick new values of the parameters for the next step.

If things are working correctly, we should eventually get samples from the posterior distribution. ***But only if we run the chain long enough***.

We can approximate the posterior distribution (and calculate useful quantities like means and credible intervals) from these samples.

Convergence checks

We cannot know for sure if the chains have run long enough (so that we're sampling the posterior distribution).

But we can (and should) do some checks that show when they have not!

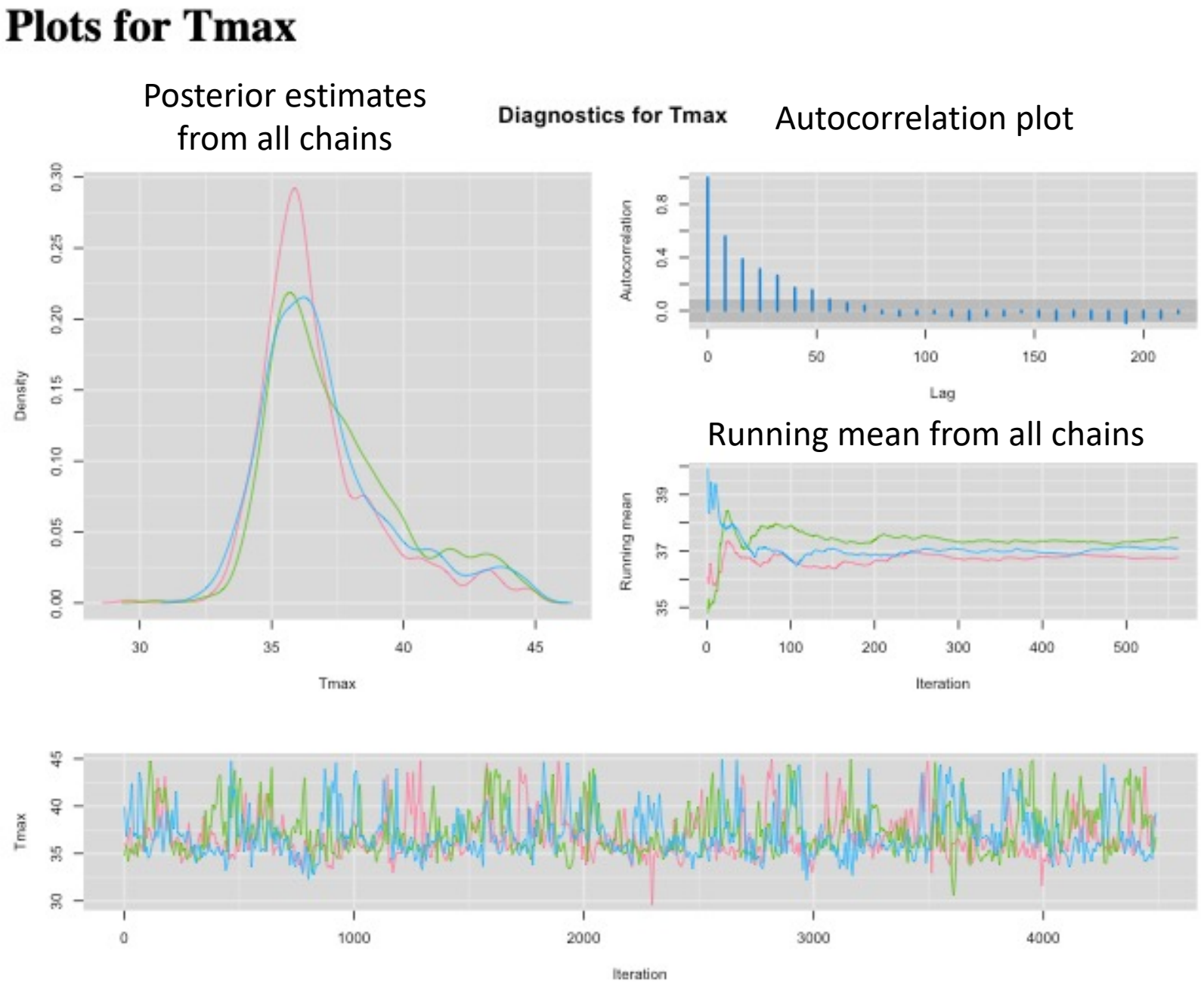
Main idea: Run multiple chains (starting from different initial values) and check if the approximated posterior distribution of our parameters agrees.

Graphical MCMC diagnostics

Looks like the chains have not converged yet.

Can try running the chain for more iterations (or change starting values).

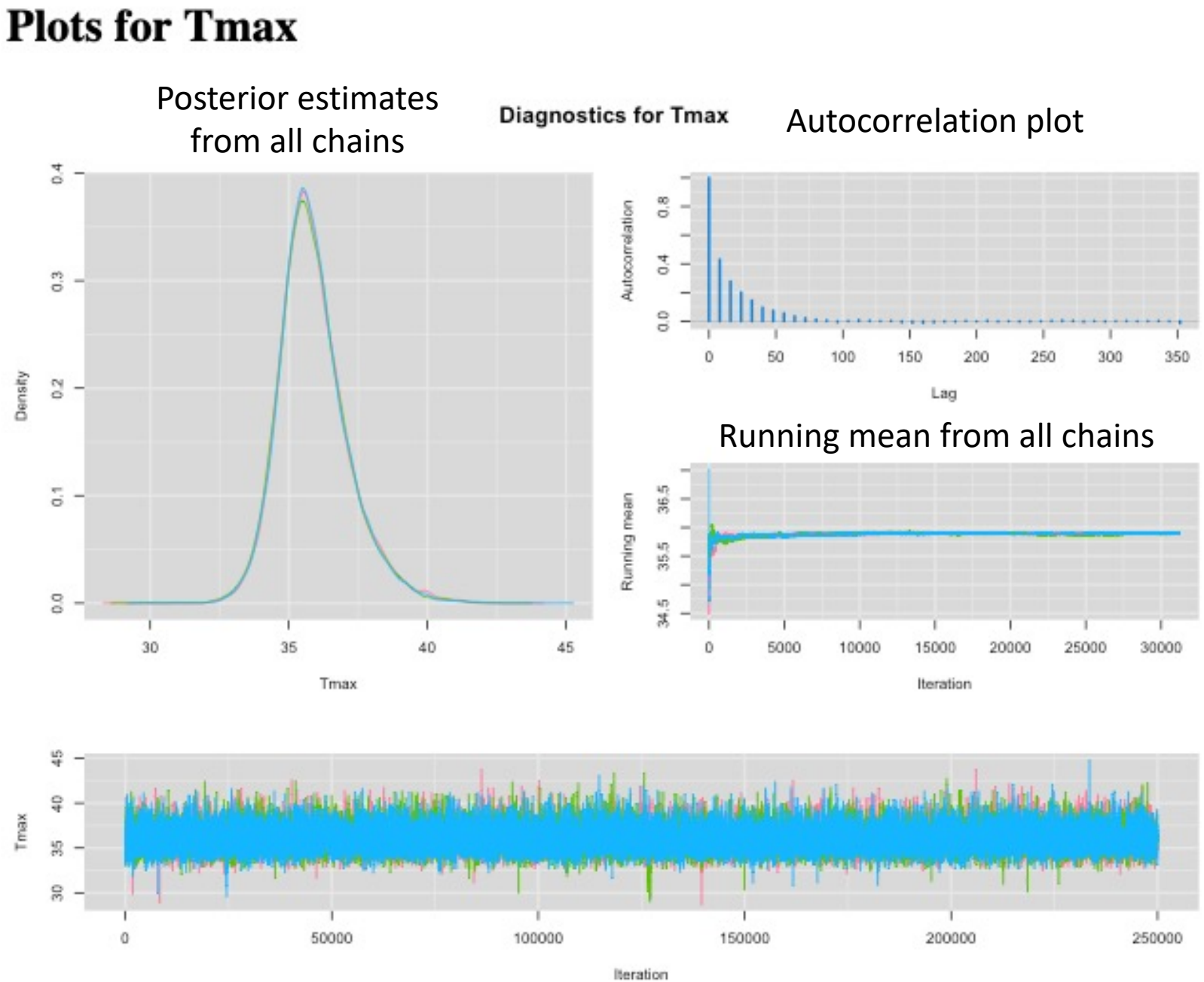
Traceplot (values at each iteration)



Graphical MCMC diagnostics

The different chains agree.

This looks OK!



Summary of MCMC output

MCMC convergence statistics

	mean	sd	2.5%	25%	50%	75%	97.5%	Rhat	n.eff
Tmax	35.8979	1.2487	33.7884	35.0711	35.7530	36.5741	38.8001	1.0010	54000
Tmin	8.9839	2.4098	3.5844	7.6709	9.1633	10.4752	13.3837	1.0011	14000
Topt	29.7926	0.8908	28.2908	29.2266	29.6799	30.2474	31.9065	1.0010	94000
c	0.0001	0.0000	0.0000	0.0001	0.0001	0.0001	0.0001	1.0011	16000
deviance	-34.7939	4.3485	-40.6200	-38.0308	-35.6686	-32.4944	-24.0339	1.0010	94000
sigma	0.0096	0.0069	0.0034	0.0055	0.0076	0.0112	0.0277	1.0010	94000

Rhat: Measure of how much the samples in different chains agree with each other. Should be close to one.
Rule of thumb: We want Rhat < 1.01 for all parameters.

n.eff: Number of effective samples in chain (not the same as number of iterations: MCMC samples will be correlated).
Rule of thumb: For all parameters of interest, if we want to calculate CIs we ideally want n.eff > 10000.

Low n.eff despite running many iterations may indicate a problem with the model (priors, likelihood or both) or the initial values we're picking for the chain.

Steps in Bayesian analysis

1. Set up a full probability model.
   - Choose TPC model for performance/trait mean.
   - Choose statistical model for data (which distribution, variance assumptions).
   - Choose prior distributions for all model parameters.

2. Find posterior distribution.

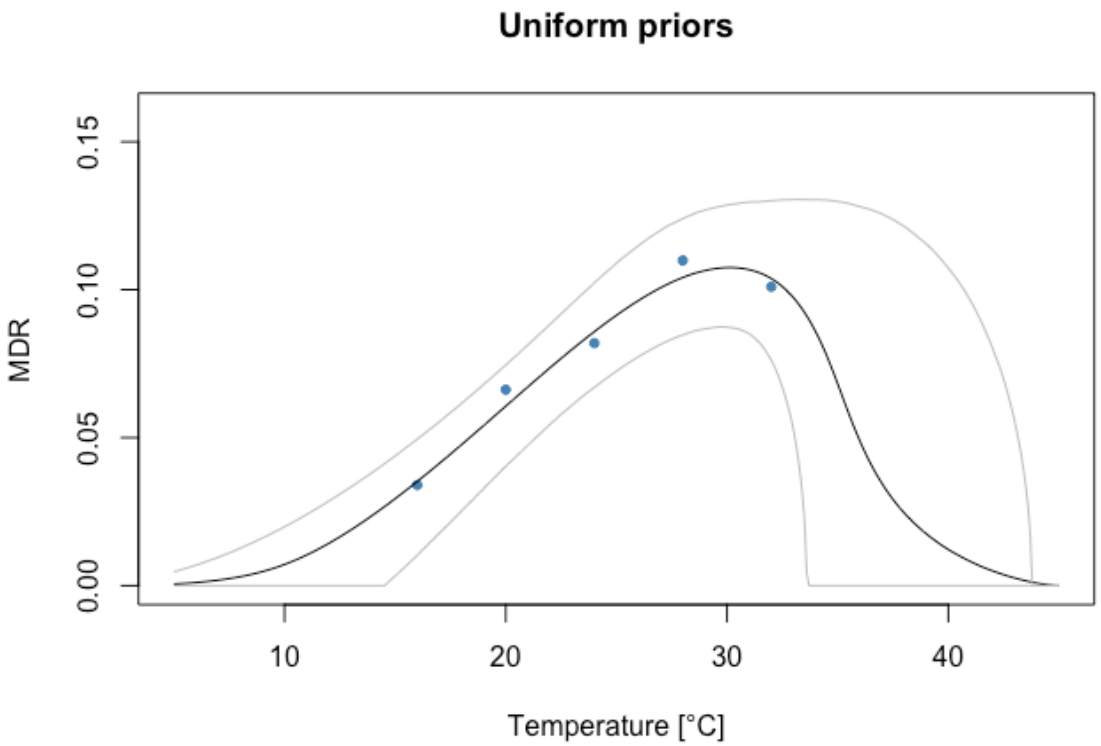
3. Evaluate model fit and implications of posterior distribution. Might alter/expand model and go back to step 1.

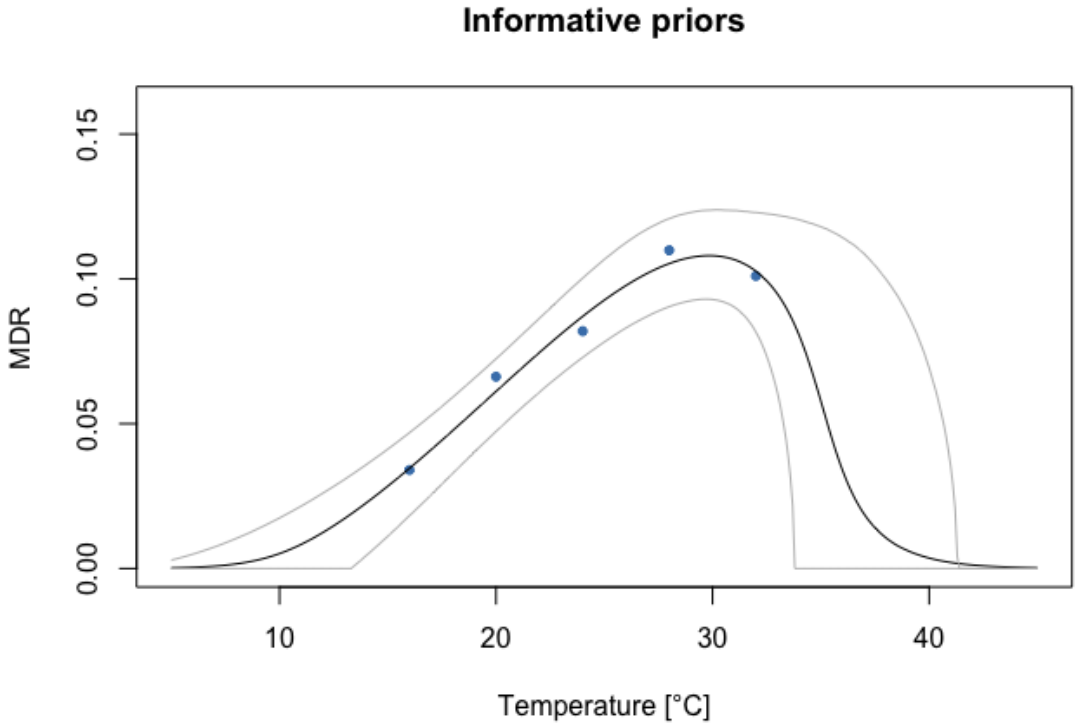
Model checking

- Does the model fit look reasonable?
- Do the estimated curves/parameters (and credible intervals) make sense? Do they look like they include values that should be unfeasible?
- How much do the inferences change when using reasonable alternative modeling choices (e.g. different priors)?

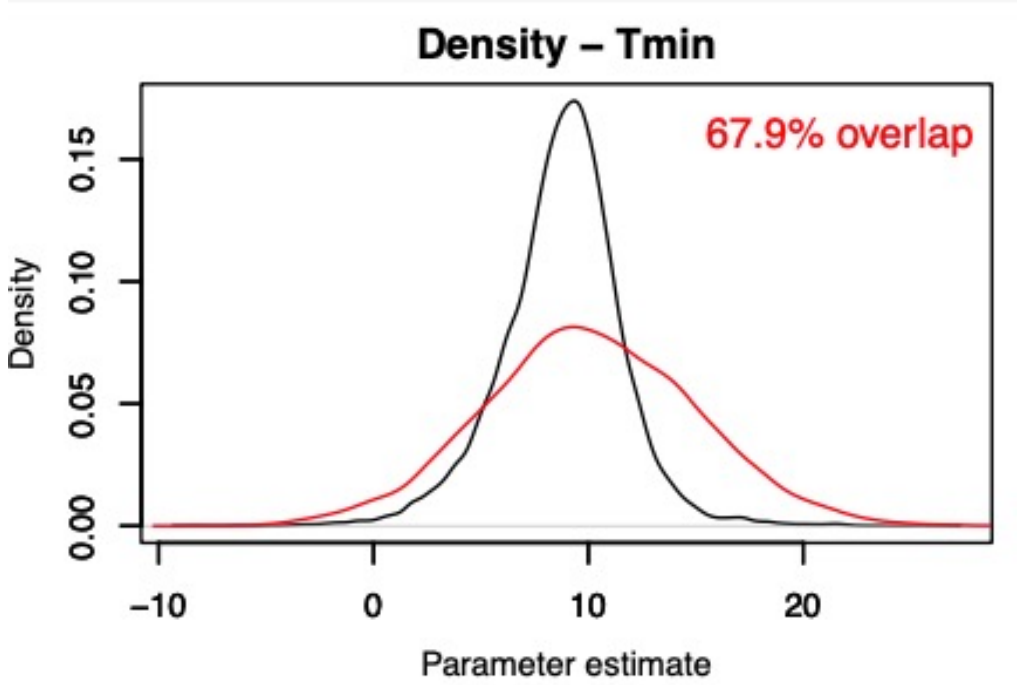
Based on these questions, we may need to revisit model assumptions (TPC model, prior, likelihood).

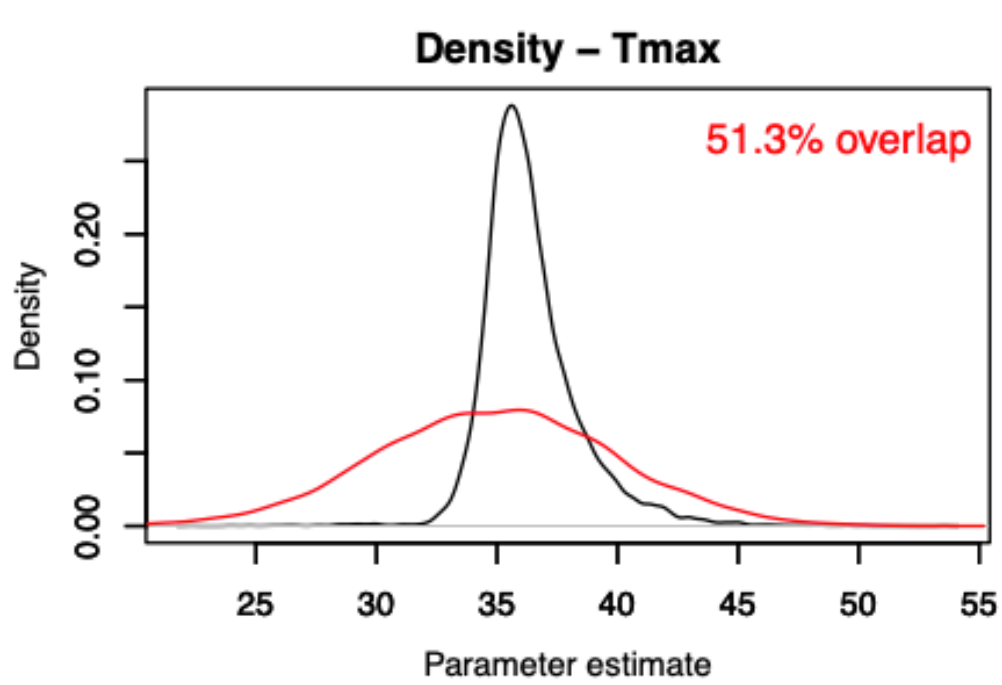
Example with few datapoints

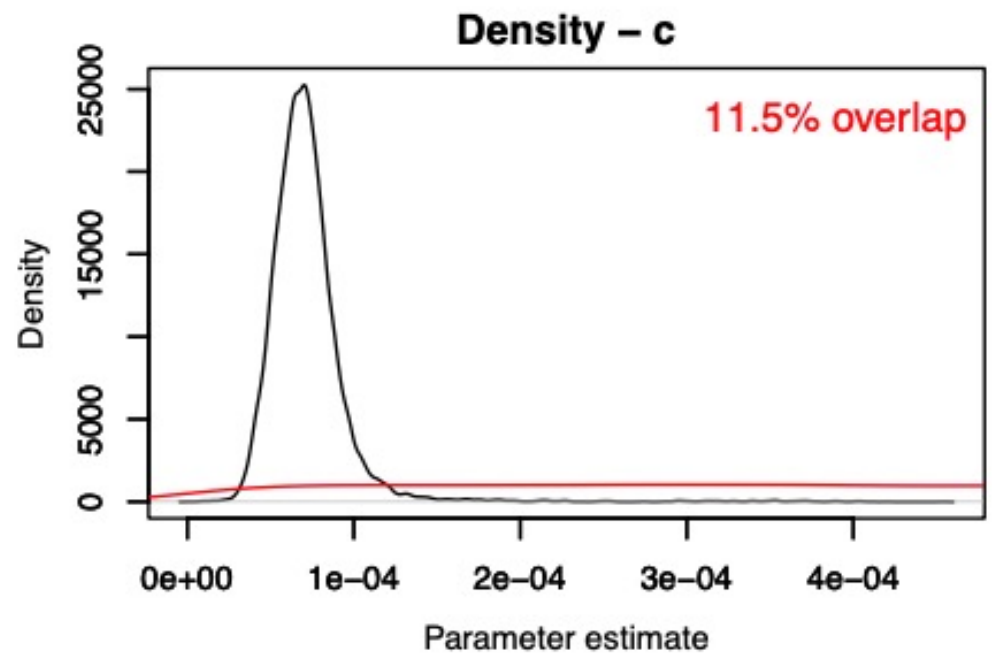


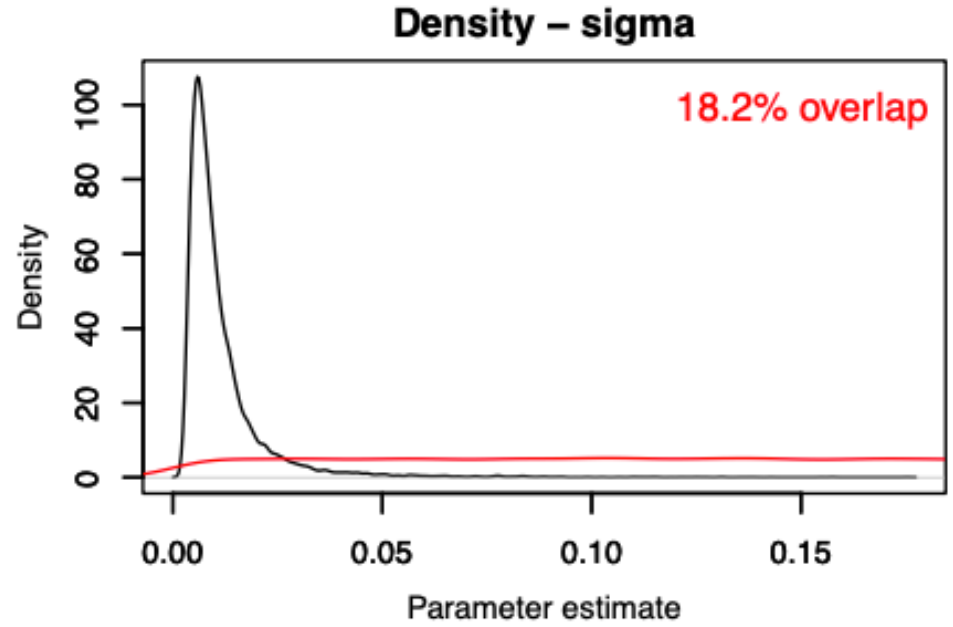


Culex pipiens MDR

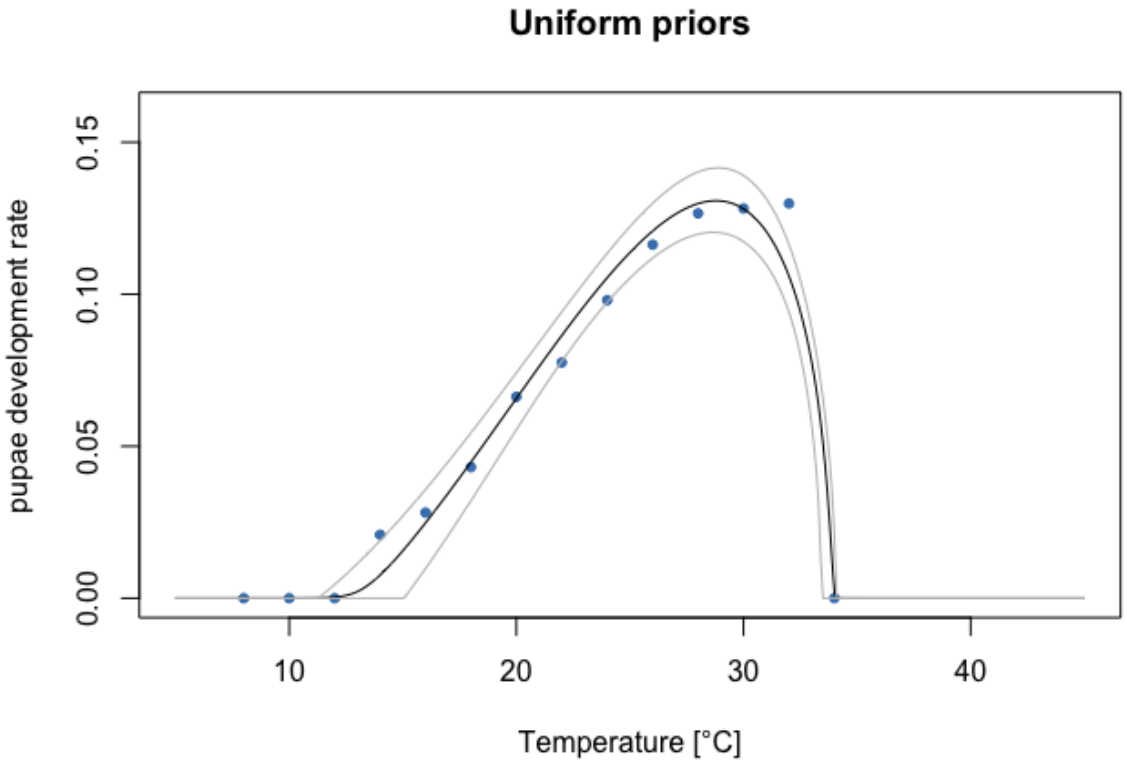

Density – Tmin
67.9% overlap
Density
0.00
0.05
0.10
0.15
−10
0
10
20
Parameter estimate

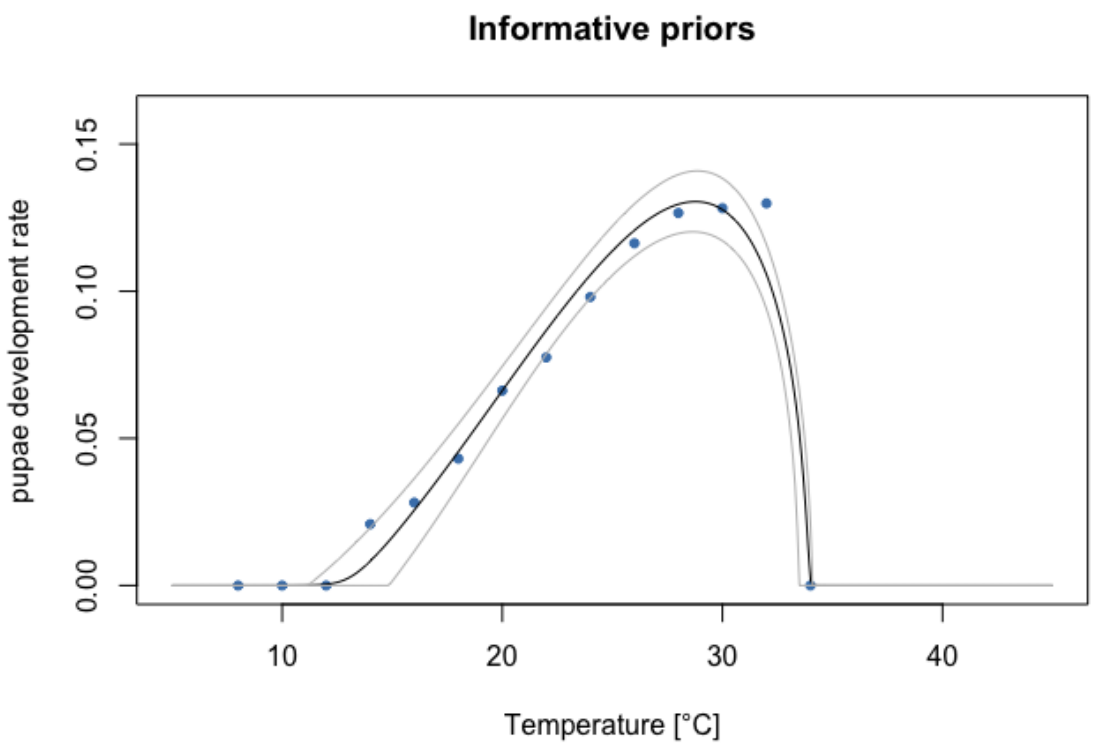

Density – Tmax
51.3% overlap
Density
0.00
0.10
0.20
25
30
35
40
45
50
55
Parameter estimate


Density – c
11.5% overlap
Density
0
5000
15000
25000
0e+00
1e−04
2e−04
3e−04
4e−04
Parameter estimate

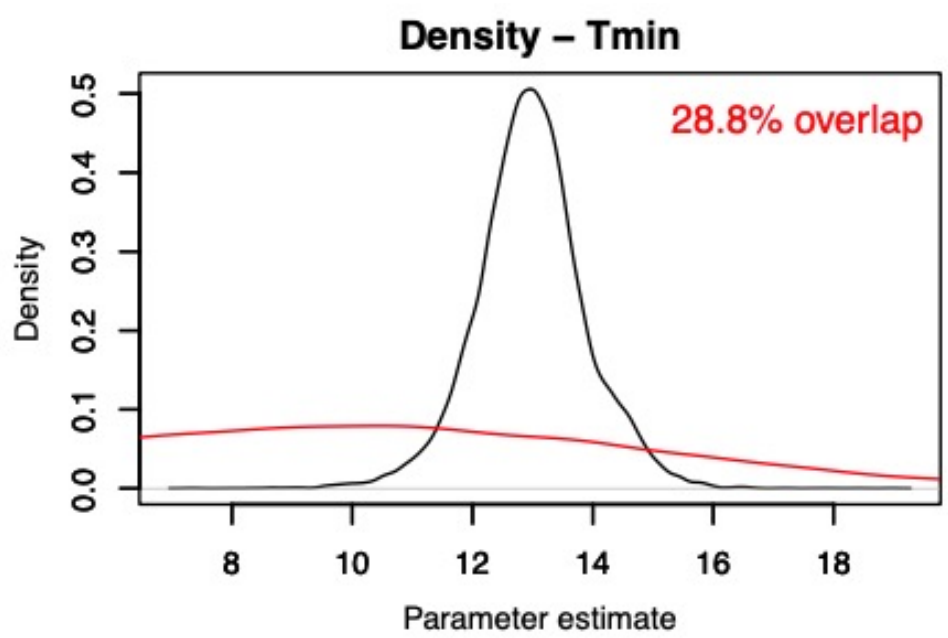

Density – sigma
18.2% overlap
Density
0
20
40
60
80
100
0.00
0.05
0.10
0.15
Parameter estimate

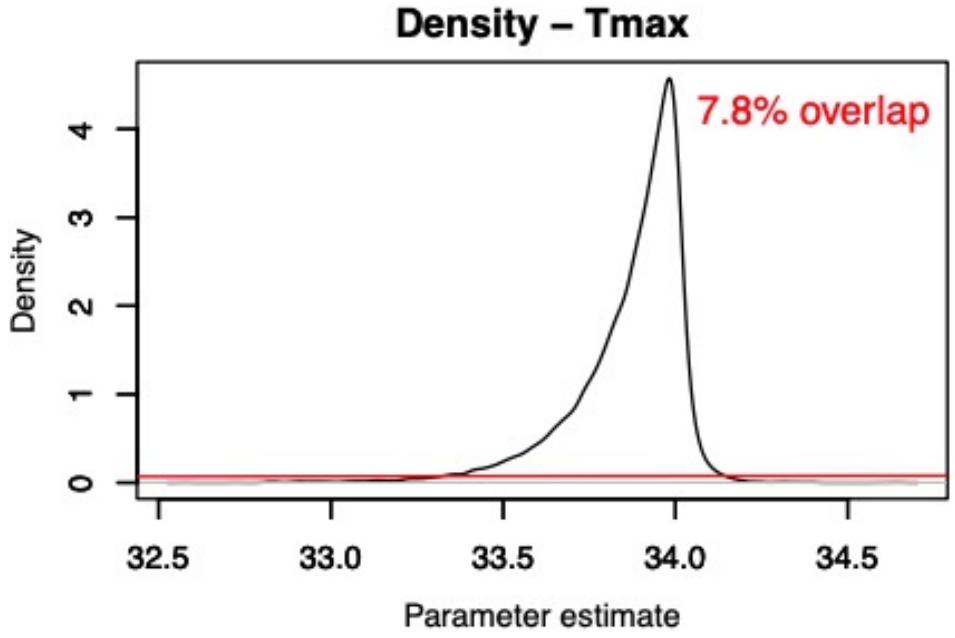
Example with many datapoints

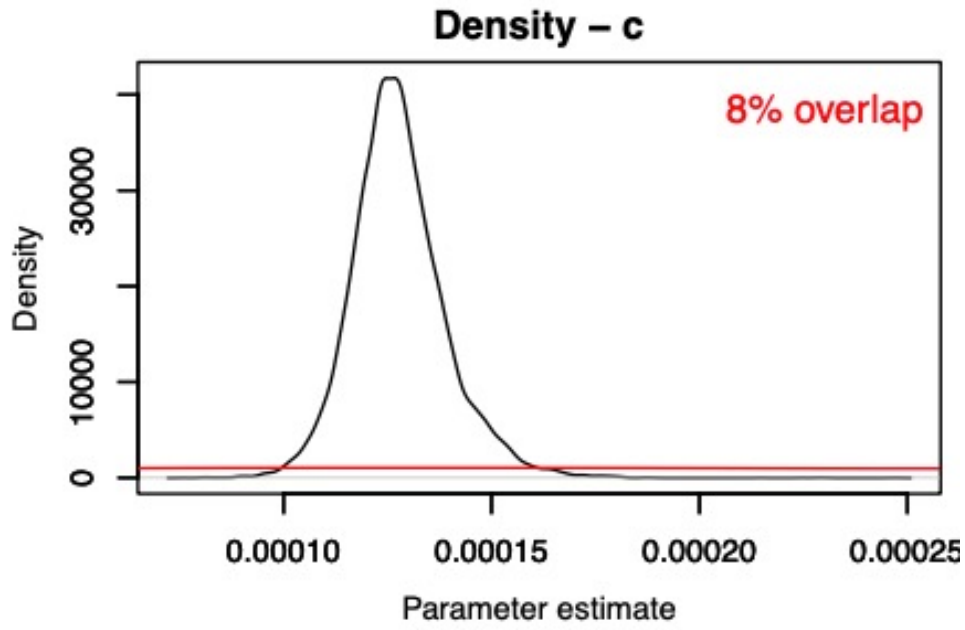


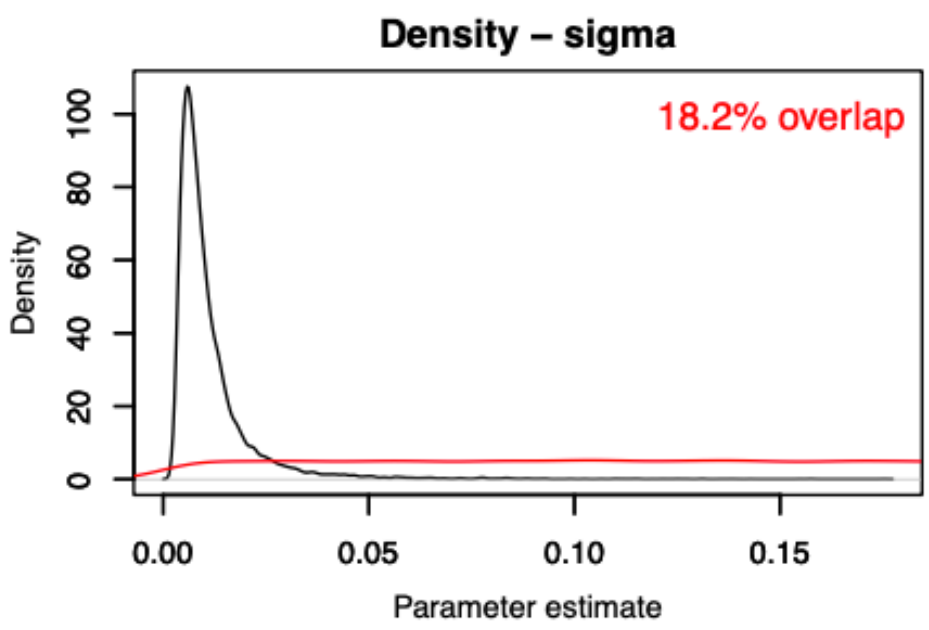


Lobesia botrana pupae development rate


Density – Tmin
28.8% overlap
Density
0.0
0.1
0.2
0.3
0.4
0.5
8
10
12
14
16
18
Parameter estimate


Density – Tmax
7.8% overlap
Density
0
1
2
3
4
32.5
33.0
33.5
34.0
34.5
Parameter estimate


Density – c
8% overlap
Density
0
10000
30000
0.00010
0.00015
0.00020
0.00025
Parameter estimate


Density – sigma
18.2% overlap
Density
0
20
40
60
80
100
0.00
0.05
0.10
0.15
Parameter estimate

Some additional resources

Bayesian statistics

Bayesian Data Analysis. Andrew Gelman, John Carlin, Hal Stern, David Dunson, Aki Vehtari, and Donald Rubin.

TPC specific code:

Marta Shocket's code
https://github.com/mshocket/Six-Viruses-Temp

Lisa Couper's tutorial on fitting TPCs
https://github.com/lcouper/AnalysisTutorials

Thank you for your time!