Received: 29 April 2024 | Accepted: 3 October 2024

DOI: 10.1111/2041-210X.70004

Methods in Ecology and Evolution

APPLICATION

bayesTPC: Bayesian inference for thermal performance curves in R

Sean Sorek[1] | John W. Smith Jr.[1,2] | Paul J. Huxley[1] | Leah R. Johnson[1]

[1]Virginia Tech, Statistics, Blacksburg, Virginia, USA

[2]Mathematical Sciences, Montana State University, Bozeman, Montana, USA

Correspondence
Sean Sorek
Email: sssorek1@gmail.com

Leah R. Johnson
Email: lrjohn@vt.edu

Funding information
National Science Foundation, Grant/Award Number: DBI #2016264 and DMS/DEB #1750113

Handling Editor: Nicholas Clark

Abstract

1. Reliable predictions of ectotherm responses to climatic warming are important because many of these organisms perform important roles that can directly impact human society.
2. Thermal performance curves (TPCs) provide useful information on the physiological constraints that limit the capacity of these temperature-sensitive organisms to exist and grow.
3. NLS pipelines for fitting TPCs are widely available, but these approaches rely on assumptions that can yield unreliable parameter estimates.
4. We present bayesTPC, an R package for fitting TPCs to trait responses using the nimble language and machinery as the underlying engine for Markov Chain Monte Carlo. bayesTPC aims to support the adoption of Bayesian approaches in thermal physiology, and promote TPC fitting that adequately quantifies uncertainty.

KEYWORDS
Bayesian statistics, ecology, ectotherms, thermal performance curves (TPCs), traits

1 | INTRODUCTION

Predicting ectotherm responses to climatic warming is important because these organisms perform critical roles in virtually all ecosystems. Often, research effort focuses on determining species' lethal temperatures (e.g. high or low temperature limits; Deutsch et al., 2008; Weaving et al., 2022) or predicting temperatures at which their thermal fitness is optimized (Pawar et al., 2024). Alternatively, thermal performance experiments measure traits (e.g., juvenile development time) across a range of constant temperatures to obtain thermal performance curves (TPCs; Huxley et al., 2022; Schoolfield et al., 1981). Specific mathematical functions are often used as TPCs to quantify the shape of the thermal response (Huey & Stevenson, 1979). Many possible shapes are available (e.g. symmetric, left/right-skewed, Figure 1) through different proposed TPC functions see (Amarasekare & Savage, 2012; Gajewski et al., 2021).

TPCs are typically fitted to trait data using nonlinear least squares NLS, for example rTPC (Padfield et al., 2021). This choice is pragmatic—NLS pipelines are becoming widely available, and, if optimization routines work, fitting occurs quickly. However, this approach has several drawbacks. First is that NLS algorithms require good initial guesses for the algorithms to converge. Multi-start schemes (Padfield & Matheson, 2018) can ameliorate this issue, but they cannot guarantee convergence to a global optima. Second, NLS inherently assumes that observed data are normally distributed with constant variance. This assumption suits many data sets, but it can be poor for data with well-defined bounds (e.g. strictly positive data), high skew or non-constant variance. Such assumption mismatch can yield poorly estimated parameters and biased predictions. Further, NLS fits typically

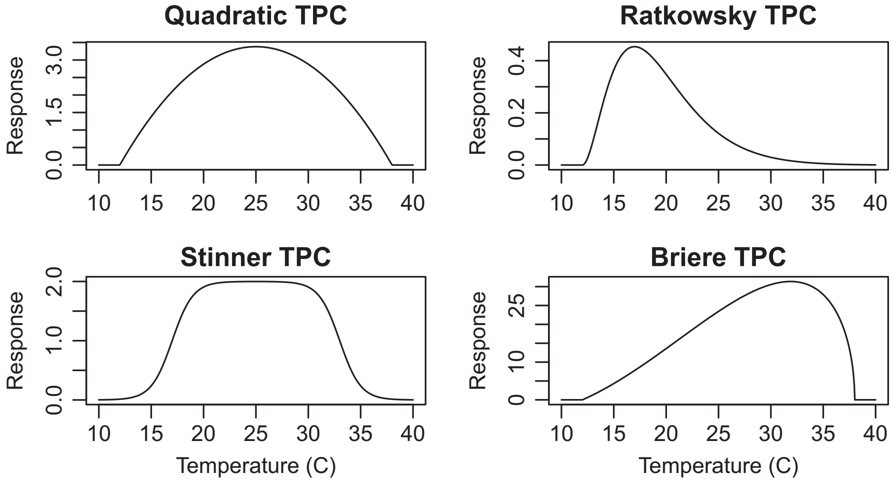

FIGURE 1 Examples of some TPC model shapes implemented in bayesTPC.

use normal approximations for the uncertainty in parameters/confidence intervals. Because normal distributions predict values from $-\infty$ to ∞, this can result in biologically unrealistic parameter bounds. Bootstrapping can partially address this, for example in rTPC (Padfield et al., 2021), but it tends to be unstable for small data sets.

Likelihood-based methods (including Bayesian approaches) can ameliorate many of the issues with NLS. For example, formal definitions of data distributions can properly account for non-normal data and thus properly constrain predictions. They also allow model comparison through information theory (Angilletta Jr., 2006). Further, Bayesian approaches allow relatively straightforward, principled inclusion of biological knowledge into analyses (through prior specification) while providing multiple uncertainty estimates, including predictive uncertainty (not available for NLS), even for small data sets.

Bayesian approaches have previously been used to fit TPCs for vector-borne disease systems (El Moustaid et al., 2021; Johnson et al., 2015). Although training and code for researchers to employ Bayesian approaches for TPCs are available (e.g. VectorBiTE RCN), their adoption remains slow. This is likely because the barrier to entry remains high and because fitting nonlinear models to data is hard.

Accordingly, we developed the bayesTPC package to allow simpler Bayesian TPC fitting in R. We include tools to assist output inspection, provide summaries and plot results. Our package release includes implementations of multiple TPCs and simple Bayesian generalized linear models (GLMs), including sensible default priors for TPCs suitable for small arthropods.

Below, we introduce the general approach to Bayesian TPC fitting and its implementation in bayesTPC. We outline package functionality and provide a simple example of how the package can be used. Finally, we discuss some of our specific choices and subtleties for users to bear in mind. Note that we do not seek to introduce all functionality in depth, nor provide a general purpose introduction to Bayesian methods (we point readers to an approachable introduction in Hobbs & Hooten, 2015). We provide additional documentation and training in Bayesian methods in our online training materials.

2 | CONCEPTUAL APPROACH

bayesTPC attempts to estimate the probability distribution of the parameters, $\boldsymbol{\theta}$, of a specific TPC and observation model given a data set of a performance trait $\mathbf{Y}_{\text{obs}}$ across a range of temperatures $\mathbf{T}$ while accounting for data uncertainty. We assume that the observed data represent underlying traits that can be modelled as a function of the parameters and temperatures, $\boldsymbol{f}$, together with an observation/noise model $\mathscr{M}$:

$$\mathbf{Y} = f(\theta_f, \mathbf{T})$$
$$\mathbf{Y}_{\text{obs}} = \mathscr{M}(\mathbf{Y}, \boldsymbol{\theta}_{\mathscr{M}}),$$

where θ_f are the TPC-specific parameters, and $\theta_{\mathscr{M}}$ are the parameters of the observation model (so that $\boldsymbol{\theta} = (\theta_f, \theta_{\mathscr{M}})$). For example, if there is additive Gaussian noise with mean θ and variance σ^2, then the model is:

$$\mathbf{Y} = f(\theta_f, \mathbf{T})$$
$$\mathbf{Y}_{\text{obs}} \sim \mathcal{N}(\mu = \mathbf{Y}, \sigma^2).$$

Our inference target is the posterior distribution of parameters (or simply the posterior distribution), $\Pr(\theta | \mathcal{Y})$, where $\Pr(\cdot)$ denotes a probability, $\mathcal{Y}$ denotes the data (both traits and temperatures) and θ denotes the set of model parameters (both for the TPC and the observation model). Using Bayes theorem, the posterior distribution is calculated as:

$$\Pr(\theta | \mathcal{Y}) = \frac{\Pr(\mathcal{Y} | \theta) \Pr(\theta)}{\int \Pr(\mathcal{Y} | \theta) \Pr(\theta) d\theta},$$

where $\Pr(\mathcal{Y} | \theta)$ is the likelihood and $\Pr(\theta)$ is the prior, which encodes the prior knowledge about the parameters before data collection. The denominator is the probability of observing the data under any settings of model parameters. Given observed data, the denominator is a constant.

For sufficiently complex models, a closed form solution of the posterior distribution is not obtainable. Instead, the posterior may be approximated using methods such as Markov Chain Monte Carlo

(MCMC, Robert & Casella, 1999). MCMC approximates the posterior by taking pseudo-random samples from the target distribution. The samples can be used to obtain the posterior distribution of any function of the parameters (Efron, 2003) through the 'plug-in principle'. For example, the TPC posterior distribution can be obtained, as can the posterior predictive distribution (which includes both parameter uncertainty and observation variability).

Many flexible, general use tools facilitate Bayesian model building and fitting (e.g. JAGS, Plummer, 2003; nimble, de Valpine et al., 2017; Stan, Stan Development Team, 2024), reducing the necessity to code MCMC samplers from scratch. Using such tools is easier than implementing MCMC directly, but they are inaccessible for inexperienced users.

3 | TECHNICAL OVERVIEW

3.1 | Installation

bayesTPC version 1.1.6 (the version described here) is available on Zenodo (Sorek et al., 2024) and after download can be installed using install.packages("PATH/bayesTPC_1.1.6.tar.gz", repos=NULL, type="source") where the PATH indicates the location where the tarball was saved. Alternatively, users may download updated development version from GitHub using remotes::install_github("johnwilliamsmithjr/bayesTPC"). Note that nimble (de Valpine et al., 2017) must be installed and loaded before bayesTPC can be used. Further tutorial materials are available from associated links on the GitHub repository.

3.2 | Thermal performance curve (TPC) models

bayesTPC uses a bespoke format to store the model formula, likelihood and priors. We include eight common TPC models in the package (listed with get_models(), Table 1). Users can obtain the default specification and priors for these models with get_default_model_specification(). The package also includes linear and quadratic formulations for Bernoulli, binomial and Poisson GLMs, assuming canonical link functions. Users can also define custom TPC models with specify_model() and, for advanced users, likelihoods with specify_likelihoods() in case the default ones are not sufficient.

3.3 | Fitting

TPC models are fitted using the b_TPC() function, which requires two user-specified inputs. The first is data; a list with expected entries named Trait corresponding to the trait being modelled by the TPC and Temp corresponding to the temperature in Celsius that the trait was measured at. The second input is model, which is a string specifying the model name or a btpc_model object. If a string is provided, the corresponding default model specification is used.

The default specification for each model includes a default likelihood (a truncated normal distribution) and a set of default priors for all TPC and observation parameters. Default values for priors, constants and other model information are obtained using built-in helper functions (Table 2).

In addition to changing prior specification, other MCMC control parameters can be adjusted as arguments to b_TPC(), including the number of iterations (niter), burn-in period length (burn), or number of MCMC chains (nchains). Also, any of four sampling methods implemented in nimble can be specified using the samplerType argument.

3.4 | MCMC diagnostics

The output of b_TPC() includes multiple components, including a sample portion in the mcmc.list format from the coda package (Plummer et al., 2006). bayesTPC provides MCMC diagnostic plots to support a variety of posterior checks built around this common format. For example, chain convergence needs to be confirmed for analyses that depend on MCMC samples to be reliable. The

TABLE 1 Thermal performance curves included in bayesTPC.

TPC name	TPC mathematical formula
Briere	$qT(T-T_0)\sqrt{\mid T_{\max}-T\mid}\,\delta(T_{\max}>T)\delta(T>T_{\min})$
Gaussian	$r_{\max}\exp\left(-0.5\left(\frac{\mid T-T_{\mathrm{opt}}\mid}{a}\right)^2\right)$
Kamykowski	$a\delta(T_{\max}>T)\delta(T>T_{\min})(1-\exp(-b(T-T_{\min})))(1-\exp(-c(T_{\max}-T)))$
Pawar_shsch	$\delta(e_h>e)r_{T_{ref}}\frac{\exp\left[\frac{e}{8.62*10^{-5}}\left(\frac{1}{T_{ref}+273.15}-\frac{1}{T+273.15}\right)\right]}{1+\frac{e}{e_h-e}\exp\left[\frac{e}{8.62*10^{-5}}\left(\frac{1}{T_{ref}+273.15}-\frac{1}{T+273.15}\right)\right]}$
Quadratic	$-q(T-T_{\min})(T-T_{\max})\delta(T_{\max}>T)\delta(T>T_{\min})$
Ratkowsky	$\delta(T_{\max}>T)\delta(T>T_{\min})[a(T-T_{\min})(1-\exp(b(T-T_{\max})))]^2$
Stinner	$\frac{C}{1+\exp(k_1+k_2(T_{\mathrm{opt}}-\mid T_{\mathrm{opt}}-T\mid))}$
Weibull	$a\left(\frac{c-1}{c}\right)^{\frac{1-c}{c}}\left[\frac{T-T_{opt}}{b}+\left(\frac{c-1}{c}\right)^{1/c}\right]^{c-1}\exp\left[-\left(\frac{T-T_{opt}}{b}+\left(\frac{c-1}{c}\right)^{1/c}\right)^c+\frac{c-1}{c}\right]$

TABLE 2 bayesTPC functions to access information on implemented model components.

get_formula()	Gets the formula for a model
get_model_params()	Gets all parameters to be fitted
get_default_priors()	Gets all priors for fitted parameters
get_model_constants()	Gets model constants, if they exist
get_default_constants()	Gets default values for model constants, if they exist
get_models()	Lists the implemented TPC models in the package
get_default_model_specification()	Obtain the default specification and priors for implemented TPC models

traceplot() function wraps coda::traceplot() and shows the sampled values by sequential iteration. If burn has been specified, traceplot() only shows the samples after the burn-in period.

Another important task in a Bayesian analysis is to examine the relationship between the prior information and the posterior samples. The ppo_plot() function shows the degree of overlap between the priors specified versus a kernel density estimation of the posterior sample. This enables users to determine whether data are informative for a particular parameter, and provides some indication of how strongly the prior information has influenced the analysis.

We choose not to create further wrappers for additional, particularly multiple chain, diagnostics, since we believe these are sufficiently provided through coda (e.g. gelman.diag() and gelman.plot(), Figure 4). One can simply pass the sample object to these functions or any other diagnostic tool that accepts the coda::mcmc.list object type.

3.5 | Summaries and visualization

The fitted object returned by b_TPC() provides information about the fitted model (including data, model formula, and priors) with the samples from the posterior distribution. Generic methods are provided for print(), summary(), and predict(). print() provides a quick overview of the fitted model, summary() prints a detailed summary of the MCMC results and returns summary statistics of the sample including means and credible intervals for all model parameters, and predict() returns the fitted centre (mean or median) and bounding (95% quantiles or HPD intervals) of the TPC based on the MCMC samples. We additionally provide the sample maximum a posteriori (MAP) estimator—the MCMC sample with the highest posterior probability among the obtained samples, similar to a classical maximum likelihood estimator.

It is also possible to plot the marginal and joint posterior distributions. As noted above, marginal priors/posteriors can be plotted using ppo_plot(). The ipairs() function wraps IDPmisc::ipairs() to visualize the (pairwise) joint posterior distribution of parameters using default colours to represent the relative density of samples.

We also provide visualizations of the fitted TPC's posterior distribution. The defaults for the (generic method) plot() function plots the median and 95% Highest Posterior Density (HPD) interval of the *fitted function* (i.e. plugging the samples into the TPC function, and calculating the median and HPD interval at all evaluated temperatures). Alternatively one may use posterior_predictive() and plot_prediction(). The posterior_predictive() function simulates draws from the posterior predictive distribution, so it includes both the samples describing the TPC function and the observational model. It then uses these samples to calculate the mean/median and the HPD interval of those simulated points. plot_prediction() simulates from the posterior and then plots them, or can be fed input from posterior_predictive() to visualize a previously calculated prediction.

4 | EXAMPLE: LONGEVITY DATA FOR *Aedes aegypti* MOSQUITOES

4.1 | Data

To demonstrate the basic workflow of bayesTPC, we fit TPCs to individual-level data on adult longevity in *Aedes aegypti* (Figure 5, left panel, from, Huxley et al., 2022). We choose individual-level data to highlight that the Bayesian approach allows both quantification of the uncertainty in a TPC fit as well as predictive uncertainty in the range of possible trait values. These data can be loaded into R from the VecTraits database (https://vectorbyte.crc.nd.edu/vectraits-explorer) using get_VB_datasets() a helper function that interacts with VecTraits using data set id numbers for retrieval. Alternatively, one can search for data sets using find_VB_datasets().

As described in Section 3.3, trait data must be stored as a list with names Trait for the response and Temp for the corresponding temperatures (in°C) before fitting is performed:

4.2 | Inference with default settings

Once the data are formatted, we can fit a TPC to the data set using default settings with a single function call. Adult longevity are numeric data where a concave down, unimodal response, such as a Briere function, is likely appropriate.

```
bayesTPC Model Specification of Type:
  briere

Model Formula:
  m[i] <- (q * Temp * (Temp - T_min) * sqrt((T_max > Temp) * abs(T_
max - Temp))
* (T_max > Temp) * (Temp > T_min))

Model Distribution:
  Trait[i] ~ T(dnorm(mean = m[i], tau = 1/sigma.sq), 0,)

Model Parameters and Priors:
  q ~ dexp(1)
  T_max ~ dunif(25, 60)
  T_min ~ dunif(0, 20)
  sigma.sq ~ dexp(1)
```

We fit the Briere model using the b_TPC function with arguments corresponding to the data object and the name of the TPC model that we want to fit.

Once the fitting process completes, we can examine the fitted model object using print. This provides details about the model fit, priors and simple summaries of the MCMC samples.

```
bayesTPC MCMC of Type:
  briere

Formula:
  m[i] <- (q * Temp * (Temp - T_min) * sqrt((T_max > Temp) * abs(T_
max - Temp))
* (T_max > Temp) * (Temp > T_min))

Distribution:
  Trait[i] ~ T(dnorm(mean = m[i], tau = 1/sigma.sq), 0,)

Parameters:

Chain 1:
```

	MAP	Mean	Median	Priors
T_max	4.440	34.391	34.416	dunif(25, 60)
T_min	0.052	3.278	1.616	dunif(0, 20)
q	0.005	0.007	0.005	dexp(1)
sigma.sq	16.705	17.596	17.022	dexp(1)

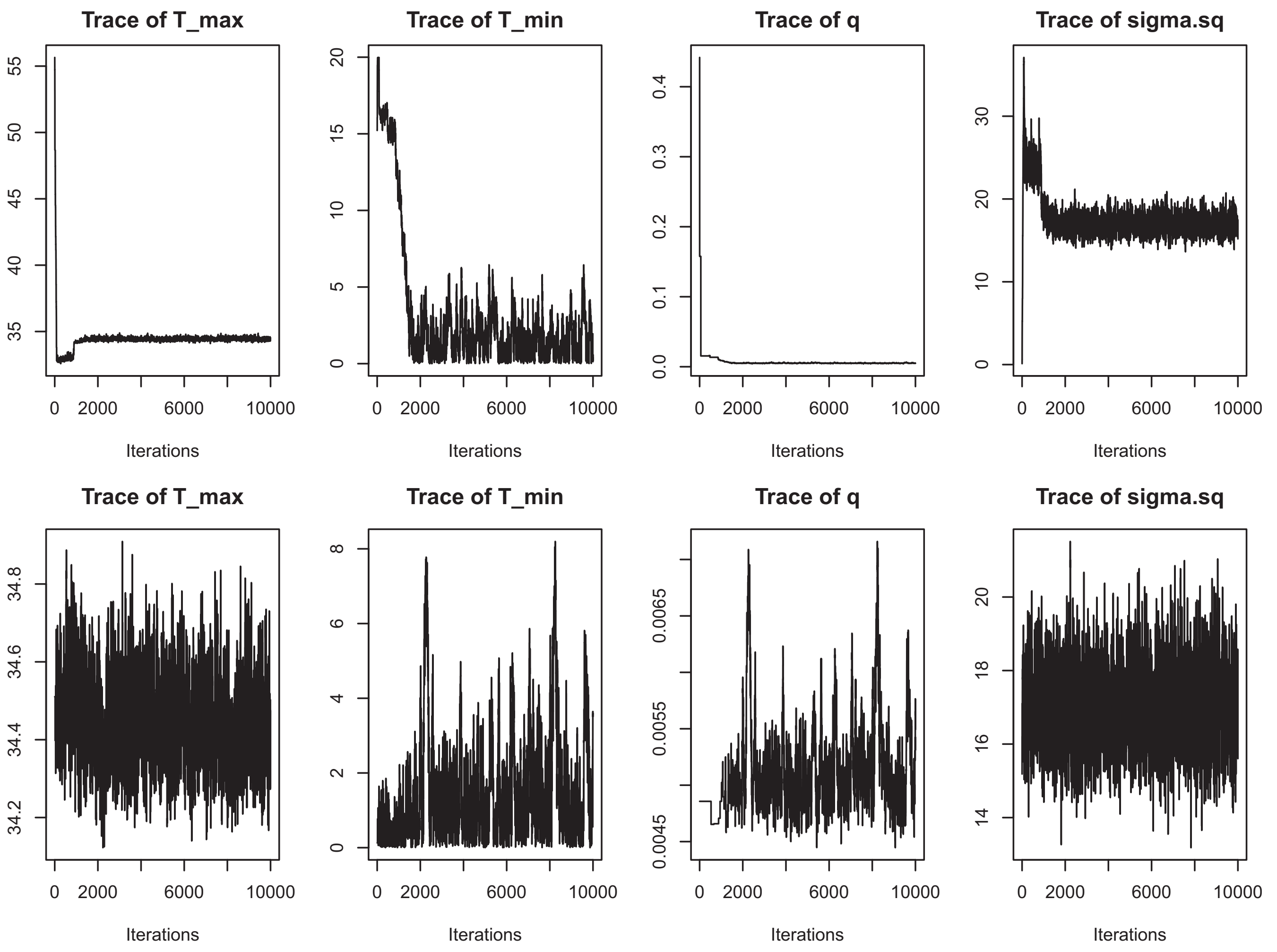

FIGURE 2 Traceplots for the three parameters of the Briere TPC and the observation parameter for model fitted to the adult longevity data. Each row represents one of two chains.

Chain 2:

	MAP	Mean	Median	Priors
T_max	34.440	34.447	34.440	dunif(25, 60)
T_min	0.052	1.462	0.981	dunif(0, 20)
q	0.005	0.005	0.005	dexp(1)
sigma.sq	16.705	16.980	16.938	dexp(1)

4.3 | Plots

b_TPC() returns an object of class btpc_MCMC containing model specification information, data, and the MCMC samples (as an mcmc.list object from the package coda, Plummer et al., 2006). Before using or interpreting a fitted model, users should check the MCMC traceplots to ensure chains have converged. If the model has converged, the marginal traceplot will eventually start varying around a single value, resembling a 'fuzzy caterpillar'.

The chains in Figure 2 are slow to converge, so we specify a burn-in period and only consider samples obtained after burn-in. Here, a value of 5000 should be sufficient. We can re-visualize with this burn-in (by adding burn = 5000 as an argument to traceplot), and we see that all of the chains now have the desired 'fuzzy caterpillar' look (Figure 3).

The Gelman plot (from coda) also shows no obvious issues, confirming that we appear to have reached convergence before the proposed burn-in value of 5000.

At this point, we would usually re-run the original fitting function, specify the burn-in and increase the total sample size to ensure sufficient samples. This approach drops the burn-in samples from the returned btpc_MCMC object. Since the chains appear to have converged to the same values, only results from the first chain are shown in the plots above for brevity.

With appropriate burn-in values selected, we can use plot() and plot_prediction() to examine the fit of the model in two ways (Figure 5). With plot() we show the mean/median and credible interval of the fitted TPC function. In contrast plot_prediction() visualizes the median prediction with the *predictive* intervals.

We can also visualize marginal priors and posteriors of our model parameters with ppo_plot(). This enables us to confirm that our posterior distributions have been informed by the data (Figure 6).

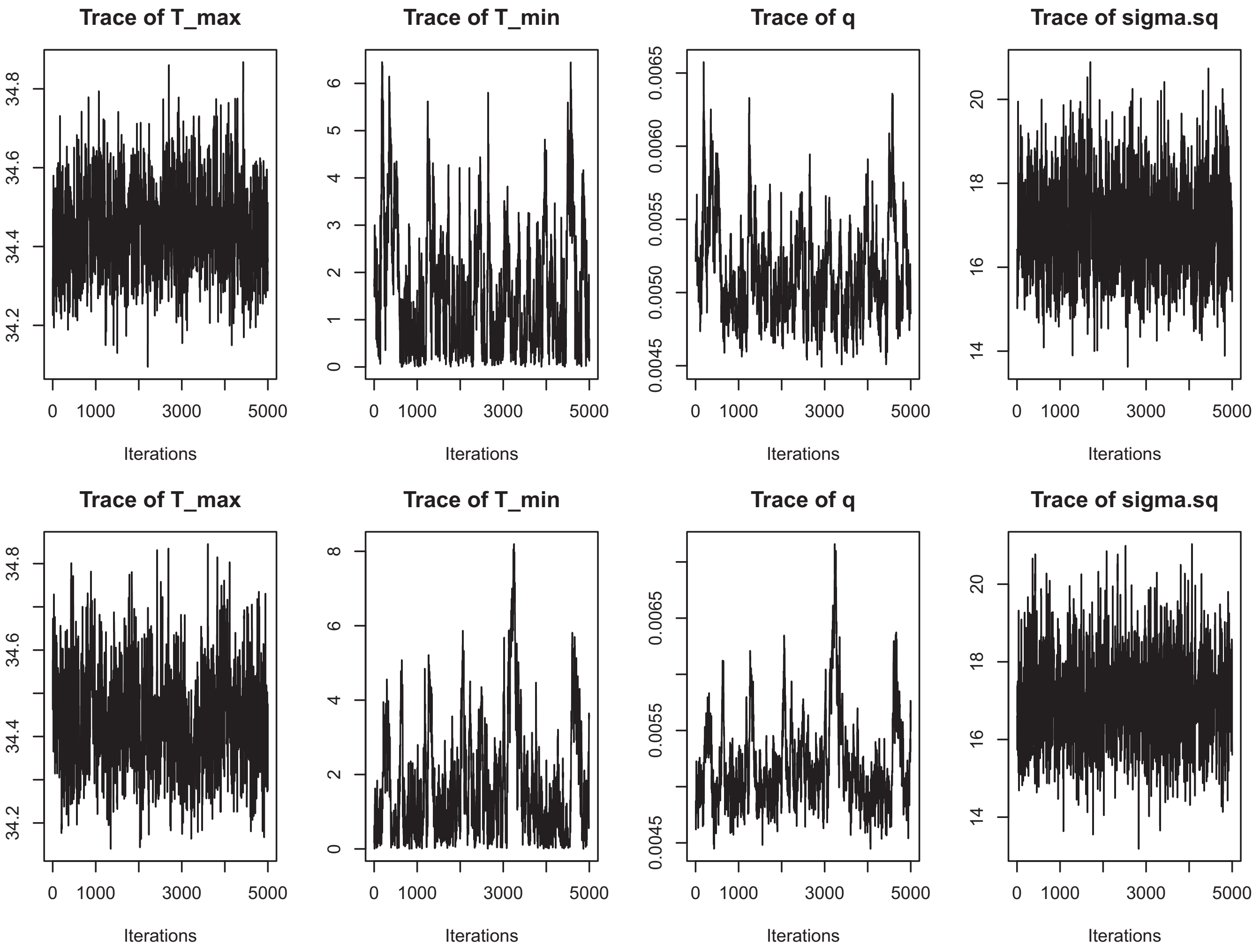

FIGURE 3 Traceplots for the Briere model parameters and the observation parameter for model fitted to the adult longevity data. Each row represents one of two chains. Here, the burn-in portion of the MCMC chains has been dropped.

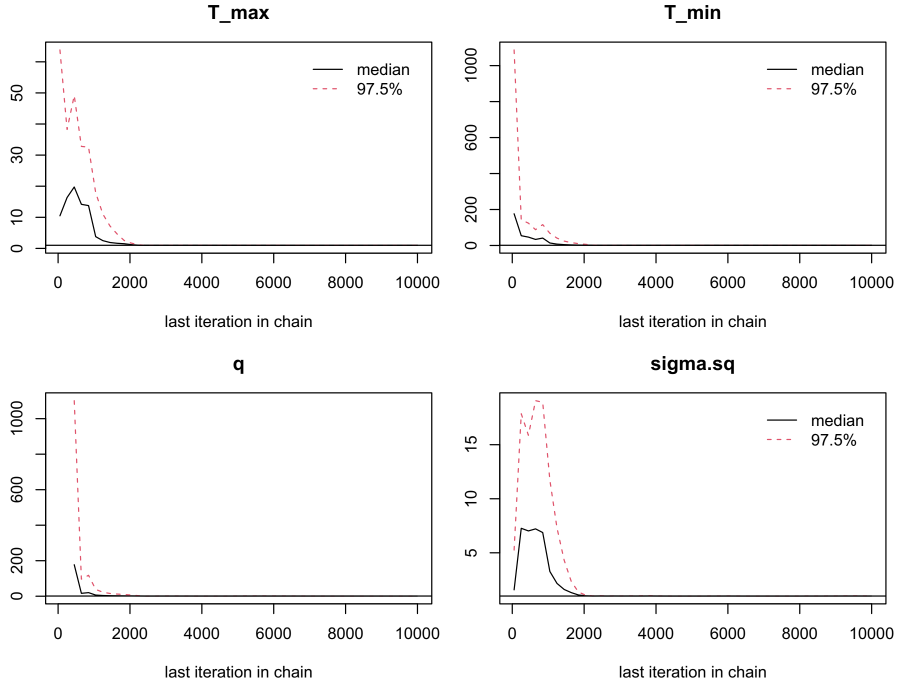

FIGURE 4 Checking chain convergence using coda::gelman.plot().

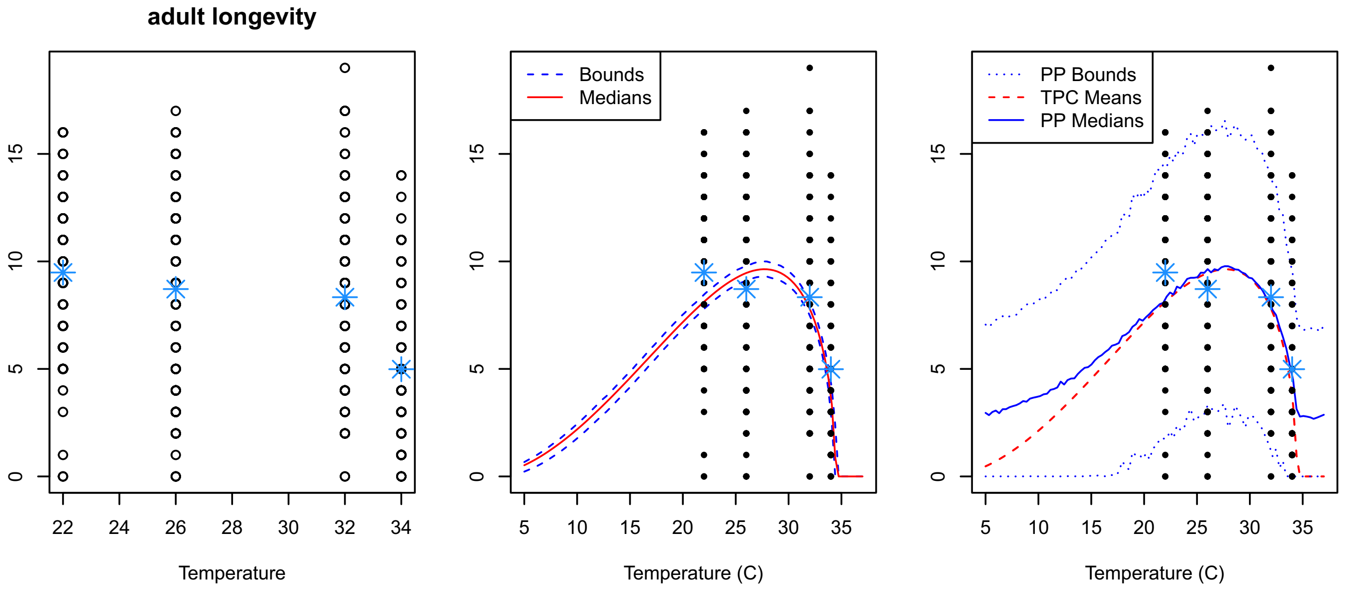

FIGURE 5 Plots of the adult longevity data (LEFT) with a comparison of the plots produced by plot() (CENTER) and the combination of posterior_predictive() and plot_prediction() (RIGHT). By default, these only make plots/predictions within the range of temperatures included in the fitted data. The temp_interval argument can be used to plot predictions across a broader temperature range. Blue stars denote mean trait values at each temperature (not included in the default plotting).

Finally, we can visualize the joint posterior distribution of parameters to understand how estimates are related to each other using ipairs() (Figure 7).

4.4 | Summaries

If we are satisfied with the traceplots, fits and prior/posterior plots, users may want to examine additional summary output (e.g. to make tables) or to save summaries. Numerical summaries are available through the print() function shown above, but we can see more details using summary().

```
bayesTPC MCMC of Type:
  briere

Formula:
  m[i] <– (q * Temp * (Temp - T_min) * sqrt((T_max > Temp) * abs(T_
max - Temp))
```

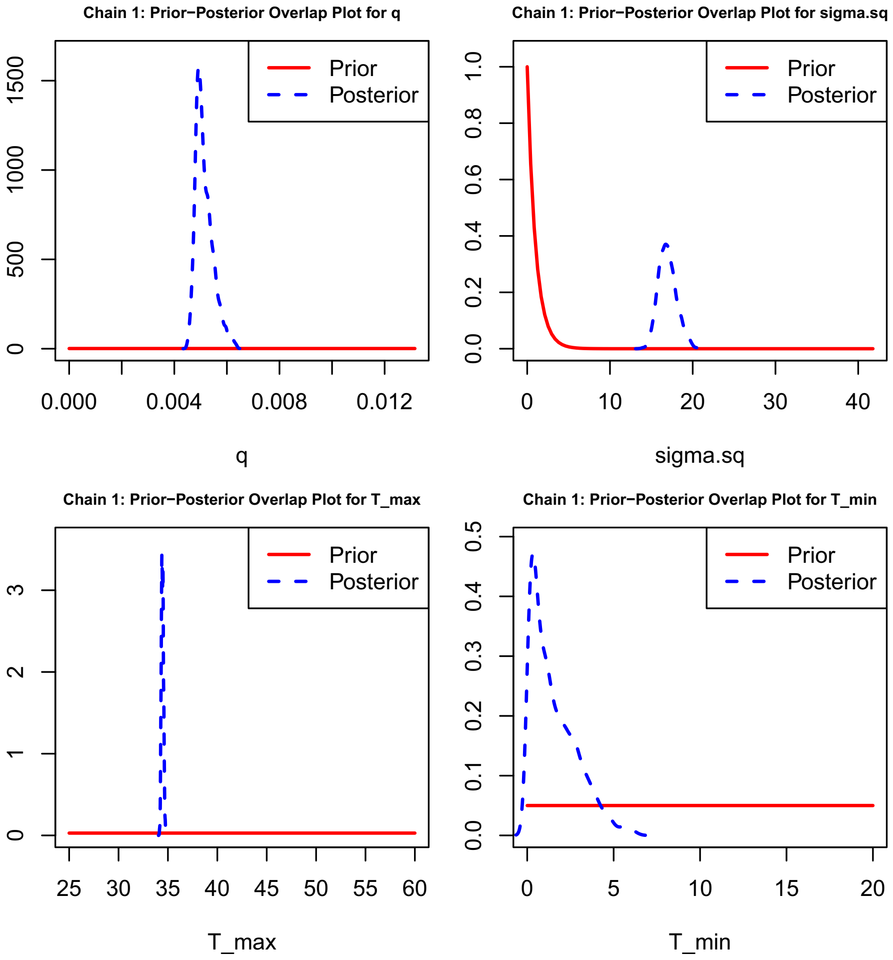

FIGURE 6 Marginal prior/posterior for the Briere TPC parameters and the observation parameter for the model fitted to the adult longevity data. The burn-in was dropped using the burn argument.

```
* (T_max > Temp) * (Temp > T_min))

Distribution:
  Trait[i] ~ T(dnorm(mean = m[i], tau = 1/sigma.sq), 0,)

Priors:
  q ~ dexp(1)
  T_max ~ dunif(25, 60)
  T_min ~ dunif(0, 20)
  sigma.sq. ~ dexp(1)

Max. A Post. Parameters:
```

T_max	T_min	q	sigma.sq	log_prob
34.4397	0.0523	0.0048	16.7049	−1925.6289

```
MCMC Results:
Iterations = 1:10000
Thinning interval = 1
Number of chains = 2
Sample size per chain = 10000

1. Empirical mean and standard deviation for each variable, plus
standard error of the mean:
```

	Mean	SD	Naive SE	Time-series SE
T_max	34.419067	0.750915	5.310e-03	0.0491462
T_min	2.370031	3.484475	2.464e-02	0.7122682
q	0.006017	0.008692	6.146e-05	0.0006164
sigma.sq	17.288402	2.010725	1.422e-02	0.1832582

```
2. Quantiles for each variable:
```

	2.5%	25%	50%	75%	97.5%
T_max	33.029047	34.346731	34.428180	34.51213	34.69102
T_min	0.040078	0.503859	1.232008	2.65858	15.65965
q	0.004656	0.004886	0.005062	0.00539	0.01352
sigma.sq	14.934202	16.224626	16.982632	17.83215	23.77183

We may want to store some of the sample statistics for later use. The sample-based MAP estimator is calculated and saved as part of our fitting process, and can be obtained directly from the fit object.

T_max	T_min	q	sigma.sq	log_prob
34.43974	0.05230	0.00483	16.70488	−1925.62888

4.5 | Comparing and selecting models

Often, we are uncertain of what is the preferred shape for a TPC (e.g. left- or right-skewed, symmetric, etc). In these cases, we may want to fit (and check) multiple model fits and then choose between them. For the example data, we fit two additional functional forms, quadratic and Stinner (Figure 8). For the quadratic, we demonstrate a default fit. The Stinner function is very flexible and can be slow to converge if data are not available at both the upper and lower limits (as in this case). Thus, we show how to modify the priors from the default in this case—to constrain the optimum of the trait to occur over a more feasible range, and ensure that the curve will be hump-shaped.

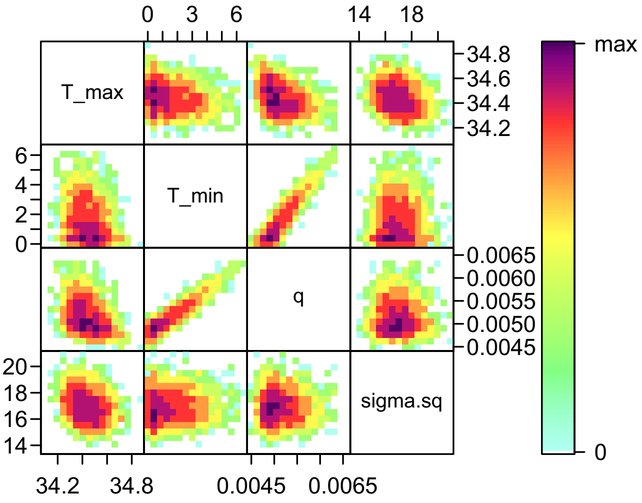

FIGURE 7 Pairwise visualization of the joint posterior distribution of parameters for the Briere TPC parameters and the observation parameter for the model fitted to the adult longevity data. The burn-in is dropped for this plot using the burn argument.

Although, a priori we would expect the quadratic fit to be best (previous work suggests it is a good choice for lifespans), all of the fits seem visually reasonable. bayesTPC allows access to the Widely Applicable Information Criterion (WAIC, Gelman et al., 2013) with get_WAIC(). This can be used to compare our fitted models. The preferred model will be the one with the lowest wAIC value.

briere.WAIC	quad.WAIC	stin.WAIC
3811.866	3825.433	3793.256

The output here indicates that the Stinner model is the best/most parsimonious fit *within sample*, even though it has more parameters. However, more data at lower temperatures are likely needed to more convincingly choose between the three models and enable reasonable extrapolation.

5 | DISCUSSION/CONCLUSION

We developed bayesTPC to support the adoption of Bayesian approaches in thermal biology research, and promote TPC fitting that adequately quantifies uncertainty. To make Bayesian TPCs easier to fit, we implemented sets of default TPCs, likelihoods and priors that are well-suited for non-negative performance data that often consist of averages across individuals/replicates. The methods implemented here can also be used for individual-level data, and for count (Poisson) or success/failure data (Binomial or Bernoulli). For more advanced practitioners, we also provide approaches and guidance for extensibility of models and likelihoods. Thus, although we assume TPCs for ectotherms as a typical use case, the package could also be used in other domains to model data exhibiting a hump-shaped response to other abiotic or biotic variables.

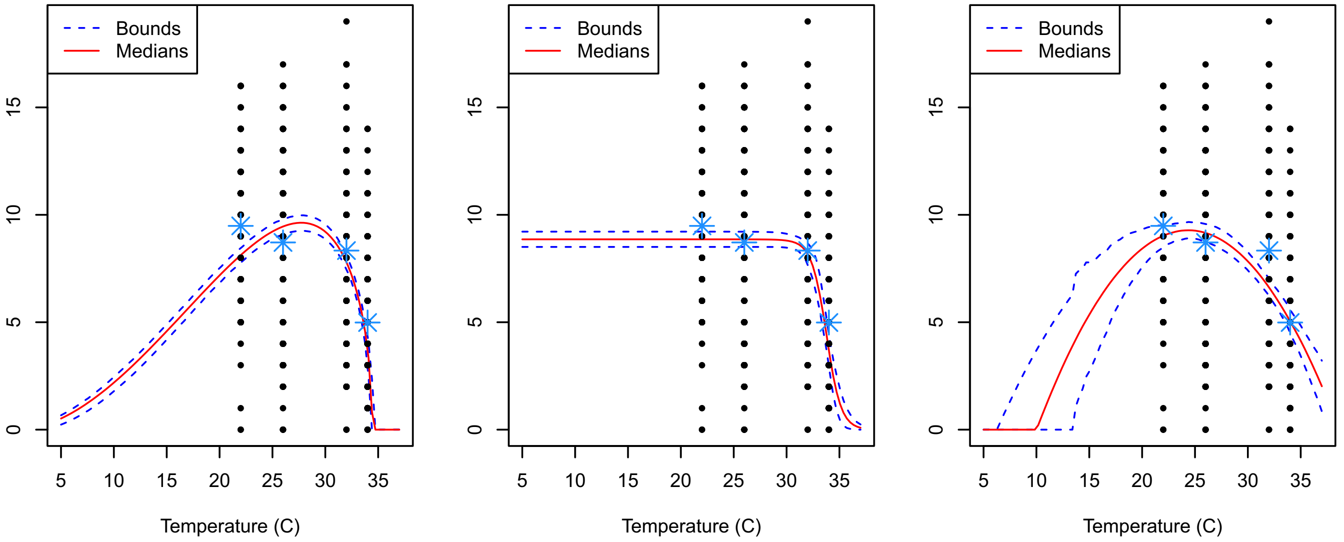

FIGURE 8 Observed adult longevity data with plots produced by the plot() function for the Briere (left), Stinner (centre) and quadratic (right) model fits. Blue stars denote mean trait values at each temperature.

We set reasonable defaults to allow straightforward fitting, but we acknowledge that our choices could be overly restrictive for some users. For example, many TPC models include T_{max} and T_{min} parameters that determine where the TPC reaches zero. These functional forms require that $T_{max} > T_{min}$. We therefore set non-overlapping uniform priors for these parameters to ensure the inequality is preserved for any sampler and is reflected in the resulting posterior distribution. This is a convenient and robust choice for this purpose. We suggest that most users simply change the bounds to modify these priors. However, because support for the uniform is finite, the posterior probability of any settings outside the prior is zero. Thus, the sampler cannot reach a 'true' value of a parameter if it is outside the prior range. Inspection of the prior-posterior plot along with the other diagnostics and fits shown here, coupled with system-specific knowledge, is required to decide whether changes in particular prior bounds are needed.

Uniform priors are also highly restrictive in terms of the prior shape that we assume for T_{max} and T_{min}. Uniform priors assume that any value within the given range is equally likely. If, at the outset, we have little information, this is not a terrible assumption. However, if prior information about these limits is available, we may want to assume a different shape. In this case, users can specify different priors for these parameters but they should be chosen carefully to prevent the sampler violating the condition that $T_{max} > T_{min}$ (as most other commonly used priors have infinite support).

We include two additional features in bayesTPC. First, the MAP estimator can be obtained as a model summary. We maintain that the full posterior is superior for most purposes. However, the MAP estimator is likely to be the best summary for users requiring single point estimates for use in follow-on modelling exercises (e.g. as an input into a differential equation model). This is because posterior distributions may not be symmetric, and in some cases, marginal means of parameters may lie away from the bulk of the posterior distribution, and thus be poor representations of the behaviour of the fitted TPC. Second, bayesTPC users can obtain wAIC values for fitted models. This is in contrast to other pipelines for Bayesian TPC fitting that rely on deviance information criterion (DIC) for model selection. We choose wAIC as recent work suggests that DIC is less theoretically grounded and more likely to have stability issues than wAIC (Gelman et al., 2013).

We have endeavoured to develop a tool that allows those with relatively little experience implementing Bayesian models to fit TPCs. We have purposefully chosen to build on a tool nimble that uses standard MCMC and a probabilistic programming language that is similar to classic BUGS/JAGS for which a large number of examples exist for ecological applications. We have also made output compatible, where possible, with many existing packages for visualization and diagnostics of MCMC chains. Thus, bayesTPC should be able to serve as a tool for simple stand-alone TPC analyses, or as a stepping stone to more complex analyses.

AUTHOR CONTRIBUTIONS

Sean Sorek: Methodology, software, data curation, visualization, validation, writing—original draft, writing—review & editing; **John W. Smith Jr.:** Methodology, software, writing—review & editing; **Paul J. Huxley:** Data curation, validation, visualization, writing—review & editing; **Leah R. Johnson:** Conceptualization, methodology, visualization, validation, writing—review & editing.

ACKNOWLEDGEMENTS

L.R.J., J.W.S. and P.J.H. were funded on NSF DMS/DEB #1750113. L.R.J., P.J.H., and S.S. were funded on NSF DBI #2016264. We thank the participants of the 2023 VectorByte Training Workshop for very helpful feedback on earlier versions of the package.

CONFLICT OF INTEREST STATEMENT

The authors have no conflict of interest to declare.

PEER REVIEW

The peer review history for this article is available at https://www.webofscience.com/api/gateway/wos/peer-review/10.1111/2041-210X.70004.

DATA AVAILABILITY STATEMENT

bayesTPC version 1.1.6 (the version described here) is available as a compressed file on Zenodo (https://zenodo.org/records/14269863; Sorek et al., 2024). Alternatively, users may download updated development versions from GitHub using remotes::install_github("johnwilliamsmithjr/bayesTPC"). Further installation instructions are detailed above. Data products are available from the VecTraits repository, as described in the main manuscript.

STATEMENT ON INCLUSION

Our study describes the development of a software package, and only uses of secondary laboratory data rather than primary data. As such, there was no local data collection.

ORCID

Paul J. Huxley https://orcid.org/0000-0001-9211-9479
Leah R. Johnson https://orcid.org/0000-0002-9922-579X

REFERENCES

Amarasekare, P., & Savage, V. (2012). A framework for elucidating the temperature dependence of fitness. *The American Naturalist*, *179*(2), 178–191.

Angilletta, M. J., Jr. (2006). Estimating and comparing thermal performance curves. *Journal of Thermal Biology*, *31*(7), 541–545.

de Valpine, P., Turek, D., Paciorek, C., Anderson-Bergman, C., Temple Lang, D., & Bodik, R. (2017). Programming with models: writing statistical algorithms for general model structures with NIMBLE. *Journal of Computational and Graphical Statistics*, *26*, 403–413.

Deutsch, C. A., Tewksbury, J. J., Huey, R. B., Sheldon, K. S., Ghalambor, C. K., Haak, D. C., & Martin, P. R. (2008). Impacts of climate warming on terrestrial ectotherms across latitude. *National Academy of Sciences of the United States of America*, *105*(18), 6668–6672.

Efron, B. (2003). Second thoughts on the bootstrap. *Statistical Science*, *18*(2), 135–140. https://doi.org/10.1214/ss/1063994968

El Moustaid, F., Thornton, Z., Slamani, H., Ryan, S. J., & Johnson, L. R. (2021). Predicting temperature-dependent transmission suitability of bluetongue virus in livestock. *Parasites & Vectors*, *14*(1), 382.

Gajewski, Z., Stevenson, L. A., Pike, D. A., Roznik, E. A., Alford, R. A., & Johnson, L. R. (2021). Predicting the growth of the amphibian chytrid fungus in varying temperature environments. *Ecology and Evolution*, *11*(24), 17920–17931.

Gelman, A., Hwang, J., & Vehtari, A. (2013). Understanding predictive information criteria for Bayesian models. *Statistics and Computing*, *24*(6), 997–1016. https://doi.org/10.1007/s11222-013-9416-2

Hobbs, N. T., & Hooten, M. B. (2015). *Bayesian models: A statistical primer for ecologists*. Princeton University Press.

Huey, R. B., & Stevenson, R. (1979). Integrating thermal physiology and ecology of ectotherms: A discussion of approaches. *American Zoologist*, *19*(1), 357–366.

Huxley, P. J., Murray, K. A., Pawar, S., & Cator, L. J. (2022). Competition and resource depletion shape the thermal response of population fitness in Aedes aegypti. *Communications Biology*, *5*(1), 66.

Johnson, L. R., Ben-Horin, T., Lafferty, K. D., McNally, A., Mordecai, E., Paaijmans, K. P., Pawar, S., & Ryan, S. J. (2015). Understanding uncertainty in temperature effects on vector-borne disease: A Bayesian approach. *Ecology*, *96*(1), 203–213.

Padfield, D., & Matheson, G. (2018). nls. multstart: Robust non-linear regression using AIC scores. R package version 1(0): 1–5.

Padfield, D., O'Sullivan, H., & Pawar, S. (2021). rTPC and nls.multstart: A new pipeline to fit thermal performance curves in R. *Methods in Ecology and Evolution*, *12*(6), 1138–1143.

Pawar, S., Huxley, P. J., Smallwood, T. R., Nesbit, M. L., Chan, A. H., Shocket, M. S., Johnson, L. R., Kontopoulos, D., & Cator, L. J. (2024). Variation in temperature of peak trait performance constrains adaptation of arthropod populations to climatic warming. *Nature Ecology & Evolution*, *8*, 1–11.

Plummer, M. (2003). JAGS: A program for analysis of Bayesian graphical models using Gibbs sampling. In: *Proceedings of the 3rd International Workshop on Distributed Statistical Computing*. Vol. 124. Vienna, Austria. (pp. 1–10).

Plummer, M., Best, N., Cowles, K., & Vines, K. (2006). CODA: Convergence diagnosis and output analysis for MCMC. *R News*, *6*(1), 7–11.

Robert, C. P., & Casella, G. (1999). *Monte Carlo statistical methods* (Vol. *2*). Springer.

Schoolfield, R. M., Sharpe, P., & Magnuson, C. E. (1981). Non-linear regression of biological temperature-dependent rate models based on absolute reaction-rate theory. *Journal of Theoretical Biology*, *88*(4), 719–731.

Sorek, S., Smith, J. W., Jr., Huxley, P. J., & Johnson, L. R. (2024). bayesTPC: Bayesian inference for Thermal Performance Curves in R (version 1.1.6). zenodo.org/records/14269863

Stan Development Team. (2024). *Stan modeling language users guide and reference manual*. 2.35.

Weaving, H., Terblanche, J. S., Pottier, P., & English, S. (2022). Meta-analysis reveals weak but pervasive plasticity in insect thermal limits. *Nature Communications*, *13*(1), 5292.

How to cite this article: Sorek, S., Smith, J. W. Jr., Huxley, P. J., & Johnson, L. R. (2025). bayesTPC: Bayesian inference for thermal performance curves in R. *Methods in Ecology and Evolution*, *16*, 687–697. https://doi.org/10.1111/2041-210X.70004