Appendix S1 - Statistical explanation and specification

Linear regression

The basic linear regression model can be written as follows:

$$\begin{aligned} \mathbf{y} &\sim \mathbf{N}(\mu, \Sigma) \\ \mu &= \alpha + \beta \mathbf{x} \\ \Sigma &= \mathbf{I}\sigma^2 \end{aligned} \tag{S1.1}$$

where $\mathbf{y}$ is a vector of observed responses, which is assumed to be drawn from a multivariate normal distribution with mean vector μ and covariance matrix Σ. In order to describe how $\mathbf{y}$ responds to changes in the covariates $\mathbf{x}$, we model μ as a linear combination of $\mathbf{x}$ and vector of regression coefficients β with intercept α. We also assume that that each element of $\mathbf{y}$ is independent conditional on μ, so we can decompose Σ into an identity matrix $\mathbf{I}$ (having diagonal elements 1 and all other elements 0) and a single variance parameter σ^2.

Gaussian processes

Gaussian process models also assume that the observations $\mathbf{y}$ are drawn from a multivariate normal distribution, but rather than describing the relationship between $\mathbf{y}$ and $\mathbf{x}$ via the mean, μ is specified in advance and the relationship is modelled in terms of the *error* from the mean. Unlike in the linear regression model, these errors are assumed to be correlated, with the expected correlation between any two errors defined by a *covariance function*.

We can therefore specify a Gaussian process model as:

$$\begin{aligned} \mathbf{y} &\sim \mathbf{N}(\mu, \Sigma) \\ \mu &= f(\mathbf{x}) \\ \Sigma &= g(\mathbf{x}, l) \\ ln(l_k) = \theta_k &\sim N(\mu_\theta, \sigma_\theta^2) \end{aligned} \tag{S1.2}$$

where $f(\mathbf{x})$ is the mean function, evaluated at $\mathbf{x}$ and $g(\mathbf{x}, l)$ is a covariance function evaluated at $\mathbf{x}$, with lengthscale hyperparameter l. We place a

normally-distributed hyperprior over each element of the natural log of l, which we denote θ, with mean μ_θ and standard deviation σ_θ. This hyperprior defines what values l can take, and therefore how smooth the fitted terms are, and is specified in advance, along with the mean function, to express the modeller's prior belief in the shape and flexibility of the GP.

`GRaF` - specification

`GRaF` fits a *latent* GP model, allowing us to consider binary (presence/absence) observations. This model is specified as follows:

$$\begin{aligned} \mathbf{y} &\sim Bernoulli(\mathbf{p}) \\ \mathbf{p} &= \Phi(\mathbf{z}) \\ \mathbf{z} &\sim \mathbf{N}(\mu, \Sigma) \\ \mu &= f(\mathbf{x}) \\ \Sigma &= g(\mathbf{x}, l) = e^{\sum_{k=1}^{m} -\frac{r_k^2}{2l_k}} \\ ln(l_k) = \theta_k &\sim N(\mu_\theta, \sigma_\theta^2) \end{aligned} \tag{S1.3}$$

where $\Phi(.)$ denotes the cumulative density function of a standard normal distribution (the probit link function) which maps the latent GP $\mathbf{z}$ to the vector of probabilities of presence $\mathbf{p}$ and $g(.)$ is the squared exponential covariance function for m covariates and r_k is the symmetric matrix of environmental distances between records for covariate k. When an error measurement is supplied with the covariates, the squared exponential covariance function of Dallaire *et al.* (2011) is used to incorporate this error.

`GRaF` users may specify a mean function $f(.)$ and parameters μ_θ and σ_θ of the hyperprior in order to reflect their prior knowledge about the species being studied. By default we set $f(\mathbf{x})$ to return the species' prevalence as calculated from the input data (a prior which assumes no effect of the covariates) and place an informative prior on θ with parameters $\mu_\theta = ln(10)$ and $\sigma_\theta = 1$ to represent our belief in relatively smooth effects of covariates on species' niches. This prior is illustrated in Fig. S1.1.

`GRaF` - fitting

Whilst there is a closed-form solution to the conditional posterior of a GP with normally-distributed response data, finding a solution to the latent-Gaussian model is somewhat more difficult. We use an iterative procedure to

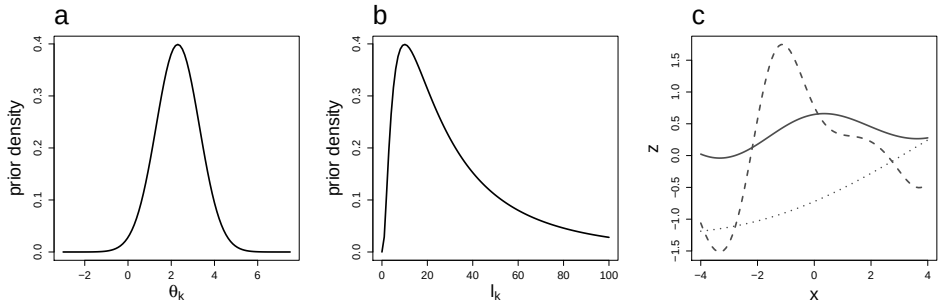


Figure S1.1: Default hyperprior distribution ($\mu_\theta = ln(10)$, $\sigma_\theta = 1$) over each element of the hyperparameter θ. (**a**) Hyperprior distribution shown on the scale of θ_k. (**b**) Hyperprior distribution shown on the scale of l_k. (**c**) Examples of GPs fitted using hyperparameters at the 0.025 ($l_k = 1.4$, dashed line), 0.5 ($l_k = 10$, solid line) and 0.975 ($l_k = 71.0$, dotted line) quantiles of the hyperprior.

compute a Laplace approximation (though the `GRaF` R package also enables the use of an expectation-propagation algorithm, see Rasmussen & Willians (2006) for details of both approaches) to the posterior distribution of $\mathbf{z}$, conditional on some value of the hyperparameters θ.

The posterior density of θ is given by:

$$p(\theta|y, x) \propto p(y|\theta, x)p(\theta) \tag{S1.4}$$

where $p(\theta)$ is the prior probability of θ (defined by the hyperprior, described above) and $p(y|\theta, x)$ is the marginal likelihood of the model given θ and is calculated from the Laplace approximation. We use a standard numerical optimisation routine (BFGS, implemented using R's `optim` function) to find the peak of this density, giving us the maximum *a posteriori* estimate for θ which we use to approximate the conditional posterior distribution of the GP and make predictions from the model.

Appendix S2 - Assessment of approximation error

As with many Bayesian statistical models, inference of the `GRaF` model is analytically intractable, so approximate inference methods are required to fit models and make predictions. Markov-chain Monte-Carlo (MCMC) methods are commonly used for inference in intractible Bayesian models. One of the reasons for the popularity of MCMC is that it can produce estimates of the posterior distributions of parameters (and predictions) with arbitrarily low approximation error, provided the model is left to run for a long enough period of time. Other approximation techniques, such as the Laplace approximation used in `GRaF`, are much more efficient but are subject to a fixed approximation error for a given model and dataset.

Here we examine the error in `GRaF`'s Laplace approximation and an alternative method: the expectation propagation algorithm (EP; (Minka, 2001; Rasmussen & Willians, 2006)) which is also implemented in the `GRaF` R package. Using a synthetic example we compare the approximate posteriors from models fitted using these approximations with those of a model fitted using a highly-efficient MCMC algorithm: Hamiltonian Monte Carlo using the No U-Turn Sampler (NUTS; Hoffman & Gelman (2011)) implemented using the STAN inference language (Stan Development Team, 2013) Note that we compare the approximate posteriors to the more accurate MCMC estimates to assess error in the approximation to the posterior distribution, rather than comparing it to the function used to generate the data, since the posterior is also influenced by the prior distribution and the dimension of the data.

Methods

We generate synthetic presence-absence data using a simple function (a sine curve and a probit link) applied to along a sequence of values of an arbitrary environmental covariate. We fit univariate latent Gaussian process models using a probit link (the simplest form of the `GRaF` model) with fixed lengthscale parameters on datasets of 100, 200, 500 and 5000 data points using the three inference methods and compare the approximate posteriors both visually and by comparing the root mean squared error (RMSE) of the Laplace and EP posteriors (evaluated at the input data points) from that of MCMC. Computation times for each model run are also provided.

For each comparison, a single MCMC chain was run for 1000 iterations, discarding the first 500 iterations as a burn-in period and using the remainder as samples from the posterior. Note that NUTS converges very rapidly

by comparison with other MCMC algorithms and this number of samples was sufficient to ensure convergence (confirmed by potential scale reduction factor) and adequate sampling from the posterior in a reasonable time. All experiments were run sequentially on a single core of a laptop computer (3GHz intel® Core™ i7). R code to reproduce this experiment and the resulting figures is provided in the supplementary information.

Results

Table S2.1 gives the RMSE of the Laplace and EP approximations from the more accurate MCMC approximation. The posterior predictive means and credible intervals estimated using the Laplace and MCMC approximations are plotted in Fig. S2.1.

For small datasets the error in both the Laplace and EP approximations is obvious - both methods under-predict at the extremes of the data range (probabilities close to 0 or 1). As more data is provided to the model this error decreases, with the Laplace and EP approximate posterior distributions being visibly very similar to those of MCMC for n=5000.

	$n = 100$	$n = 200$	$n = 500$	$n = 5000$
Laplace approximation	0.092	0.066	0.040	0.007
Expectation-propagation	0.090	0.064	0.039	0.007

Table S2.1: Root-mean-squared-error (RMSE) of the approximated posterior mean estimated using the Laplace approximation and expectation-propagation. RMSE was evaluated at the input data points for each model comparison.

Table S2.2 gives the approximate computation time for each of the models. With relatively few data points the MCMC approximation was computed in a reasonable amount of time (in less than 2 minutes for 500 datapoints) but with 5,000 datapoints the MCMC sampler took over 26 hours to run.

Discussion

For relatively small datasets and in applications where the accuracy of posterior estimates outweighs computational restrictions, using an efficent MCMC

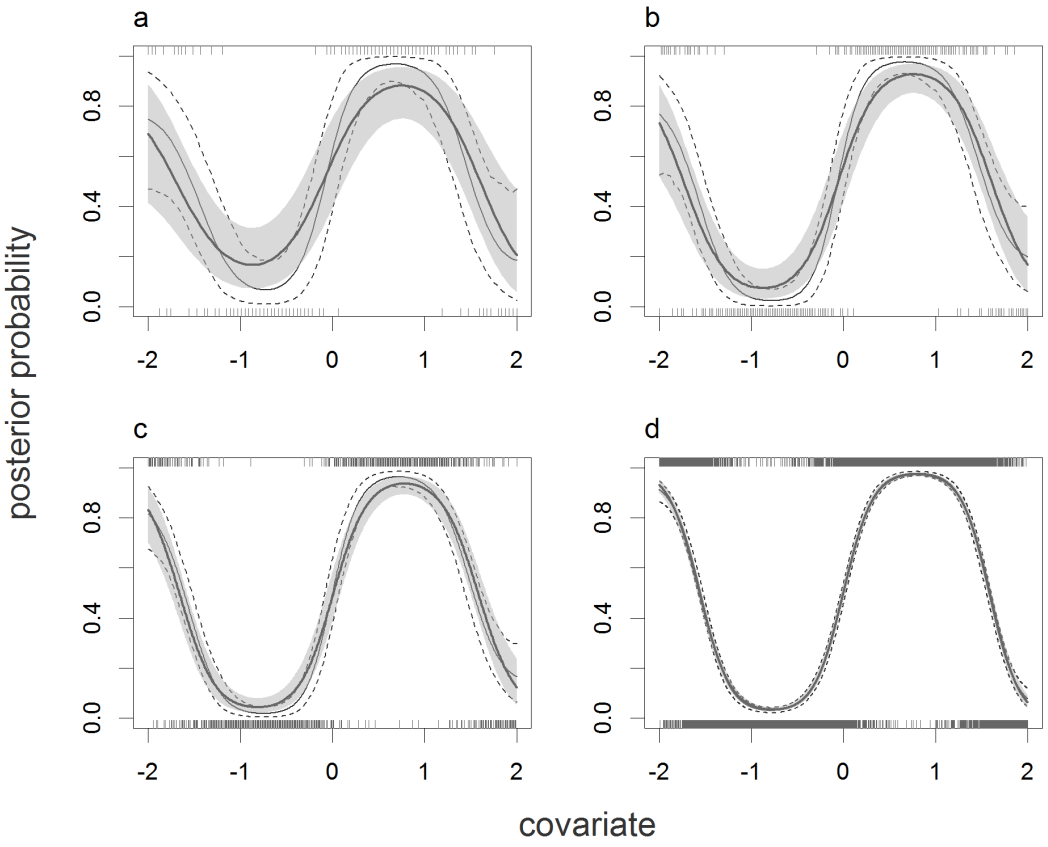


Figure S2.1: Approximate posterior distributions for `GRaF` models fitted using the Laplace approximation and MCMC to a) 100, b) 200, c) 500 and d) 5000 simulated data points. The grey line and shaded area give the mean and 95% credible intervals of the Laplace approximation to the posterior distribution. The black solid and dashed lines give the mean and 95% credible intervals of the MCMC approximation. Rug plots give the values of the data points used to train and evaluate the models. The approximate posterior of the expectation-propagation algorithm was indistinguishable from that of the Laplace approximation in each model run and so is not shown.

	$n = 100$	$n = 200$	$n = 500$	$n = 5000$
Laplace approximation	0.02	0.09	0.39	80.9
Expectation-propagation	0.17	1.01	15.29	9541.36
MCMC	2.5	13.42	151.24	94973.14

Table S2.2: Computation times (in seconds) for each of the approximation methods for each model comparison. Each MCMC model run took an additional period of between 25 and 60 seconds to compile the code for the sampler.

sampler may be preferable to the Laplace approximation used in `GRaF`. Unfortunately implementing efficient MCMC algorithms for latent GPs generally requires writing bespoke code in domain-specific programming languages such as STAN which is likely to unfeasible for the vast majority of SDM users (though see Vanhatalo *et al.* (2012))

With even moderately-sized datasets however, the computational advantage of the Laplace approximation over MCMC is clear. In our comparison the EP algorithm had little advantage over the Laplace approximation, being only very slightly more accurate, but with a markedly longer running time - though see Rasmussen & Willians (2006) for a demonstration (using a more 'extreme' artificial function) of where EP outperforms the Laplace approximation.

The results we provide here are meant only as an illustration of the error inherent in the Laplace approximation. In a practical SDM application the degree of error in the Laplace approximation error will depend on a number of factors including the number of covariates used in the model and the shape of the species' response to these covariates.

Appendix S3 - Explanation of data for SDM comparison

Distribution data

Of the 1335 vascular plant distributions available in (Preston *et al.*, 2002), 1098 were selectively removed, leaving 227 distributions with which to compare SDMs. The rejection procedure was as follows:

1. Remove any distribution with a name containing the strings: 's.l.', 's.str', 'agg.', or 'sensu' since modelling of individual species, rather than species complexes, is more representative of standard applications of SDM. *95 removed*

2. Remove any species with more than 5% of records classed as 'non-native' or which were mapped as 'native' (where non-native occurrences were not considered and which may therefore contain non-native records; see Preston *et al.* (2002)), since modelling of non-equilibrium species distributions generally requires specific, case-by-case consideration (see e.g. Elith *et al.* (2010)). *251 removed*

3. Remove species for which distributions may be biased, according to expert opinion (Dr Chris Preston, *pers. comm.*). Reasons given were: incomplete recording, failure to distinguish between native and alien spp., species status debated or discredited. *7 removed*

4. Remove species classed as hydrophytes since their distributions are likely to be dependent on very local water conditions poorly described by remotely sensed imagery. *102 removed*

5. Retain one species at random from each genus to reduce the chance of phylogenetic correlation (and associated correlation of preferred habitat types) between species from reducing the independence of the data points. *522 removed*

6. Remove species with prevalence <0.075 or >0.925 to ensure that each distribution contained at least 200 presence and 200 absence records, enough to enable both training and testing of models. *131 removed*

The distributions of the remaining 227 vascular plant species showed no clear habitat bias, though species richness displayed a slight north-south gradient (Fig. S3.1).

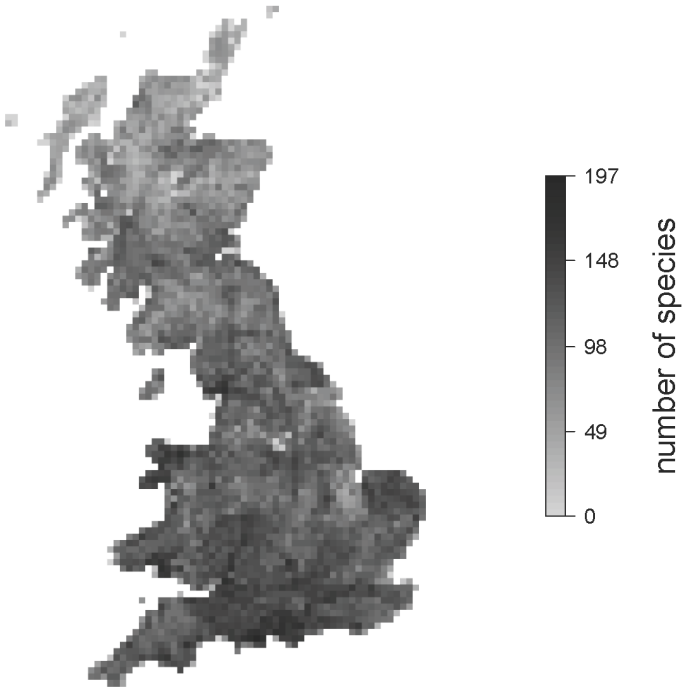


Figure S3.1: Density of presence records for the 227 vascular plant species used to compare SDMs

Environmental covariates

In order to build distribution models for the plant data, we require gridded maps of environmental data. We produced maps of the major axes of enviromental variation in Great Britain, derived from a 7-year time series of high-resolution satellite images.

Unlike maps of land cover or meteorological conditions from weather stations which are commonly used for building SDMs, satellite imagery has the advantages that it is directly recorded at high resolution (rather than being interpolated from a small number of sites) and represents a continuous measure of environmental conditions (rather than subjective land cover classes). Whilst a single satellite image provides useful information about environmental conditions, by using a time series of images, we are able to capture additional detailed information of seasonal change.

High resolution (approx. 200m^2) satellite images of Great Britain, recorded

by the moderate-resolution infrared spectrometry (MODIS) sensor on NASA's Terra and Aqua satellites every 8 days between 2001 and 2007 were obtained. For each pass, we obtained images of the following three bands: mid infrared (MIR; raw output of wavelengths 3-5 μm); enhanced vegetation index (EVI; a metric of vegetation cover) and normalized vegetation index (NDVI; a metric of vegetation cover complementary to EVI). In order to render this huge dataset (containing more than 800 images, each with around 15 million pixels) usable for SDM, it was necessary to compress this information into a smaller set of gridded maps. The imagery were first decomposed using a temporal Fourier analysis (TFA) procedure, designed specifically for MODIS data and described in detail in (Scharlemann *et al.*, 2008). This procedure returns, for each pixel and each band, 10 variables: the mean, variance, maximum and minimum of the band and 6 Fourier components: the phase and amplitude of annual, bi-annual and tri-annual cycles. These TFA variables together describe the temporal pattern of each band within each pixel, and have been successfully used in SDMs (Rogers *et al.*, 2002; Gilbert *et al.*, 2005; Johnson *et al.*, 1998). Whilst the Fourier components within each band are orthogonal to one another, correlations remain between bands and with the other TFA variables.

We use principle components analysis (PCA) on these 30 TFA variables, plus an elevation map at the same resolution from the Shuttle Radar Topography Mission (Farr *et al.*, 2007), to derive a set of 31 orthogonal principle components which describe the *major axes* of environmental variation in the TFA variable dataset. The first 10 principle components represent around 90% of the total variation in the TFA dataset (see Table S3.1) and we retain these for use in the SDM comparison. In order to model the plant distribution data at 10km resolution, we resampled the maps of these 10 components from 200m resolution to 10km resolution (the spatial distribution of these principal components is shown in Fig. S3.3).

Whilst it would have been possible to carry out the PCA on the original MODIS time series, this would have required the highly computationally-intensive decomposition of a matrix of around 12 billion elements. Carrying out the PCA on the TFA variables (which had already been produced prior to this study) required decomposition of a much smaller matrix.

Despite the comparatively high information content of these principle components, they have the disadvantage that they are not as directly ecologically interpretable as other commonly used environmental variables, such as land cover classes. In order to interpret the environmental gradients that these indices represent, we compare each of the 10 principal components with broad land cover classes at the same resolution obtained from the CEH land

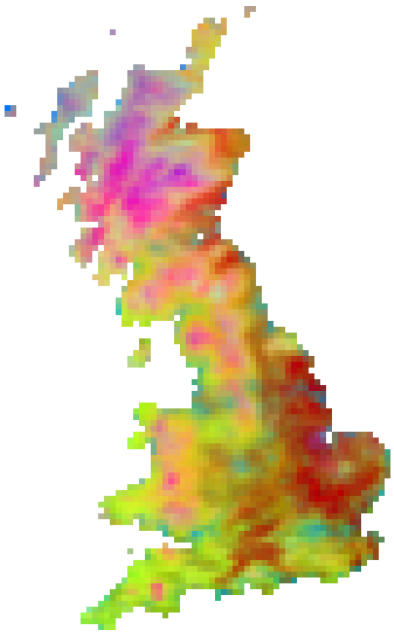


Figure S3.2: Composite image showing the spatial distribution of principal components 1 (green), 2 (blue) & 3 (red) in Great Britain. These three components account for 60% of the total variance in the TFA dataset.

cover map 2000 (Fuller *et al.*, 2002) (Table S3.1). The first three components appear to represent gradients from Arable land to pasture, from lowlands to uplands and from urban areas to coniferous woodlands respectively.

The ability of these components to describe environmental variation becomes more clear when they are combined. Fig. S3.2 combines the first three principal components. It is clear in this figure that arable land (red), pasture (yellow-green), uplands (purple) and urban areas (turquoise) are well described by the first three principal components. The additional components will combine to describe finer details of the environment which may be equally as important for describing species' niches.

	PC1	PC2	PC3	PC4	PC5	PC6	PC7	PC8	PC9	PC10
Broadleaved/mixed woodland	0.10	-0.52	-0.03	0.13	0.21	-0.12	0.10	0.04	0.21	0.37
Coniferous woodland	0.08	0.19	**0.59**	0.13	-0.09	-0.02	0.10	-0.22	**0.26**	0.21
Arable & horticulture	**-0.29**	**-0.86**	-0.41	-0.11	0.10	**-0.29**	-0.05	0.04	-0.03	0.07
Improved grassland	**0.42**	-0.56	0.11	0.22	0.22	-0.17	**0.21**	0.11	0.25	0.24
Semi-natural grassland	-0.06	0.14	0.49	**0.36**	0.05	**0.20**	0.18	-0.09	0.11	**0.40**
Mountain, heath & bog	-0.04	**0.64**	0.58	0.22	**-0.24**	0.12	0.10	**-0.24**	0.07	0.12
Built-up areas & gardens	0.04	-0.62	**-0.50**	-0.02	**0.34**	-0.11	0.08	0.15	-0.06	0.22
Standing open water	-0.15	0.26	0.10	0.06	-0.10	0.20	-0.06	-0.06	-0.10	0.11
Coastal	0.19	0.22	-0.24	-0.25	0.12	0.08	-0.14	0.17	**-0.29**	-0.49
Oceanic seas	0.19	0.36	-0.18	**-0.31**	0.05	0.12	**-0.18**	**0.18**	-0.27	**-0.52**
Proportion of variance explained	0.27	0.22	0.11	0.08	0.05	0.04	0.03	0.03	0.02	0.02
Cumulative proportion explained	0.27	0.49	0.60	0.68	0.73	0.77	0.80	0.83	0.85	0.87

Table S3.1: Spearman's rank correlation coefficients between principal components and broad land cover classes from LCM 2000, and proportion of variance of TFA components explained by each component. The strongest positive and negative correlation coefficients between each principal component and land cover classes are given in boldface.

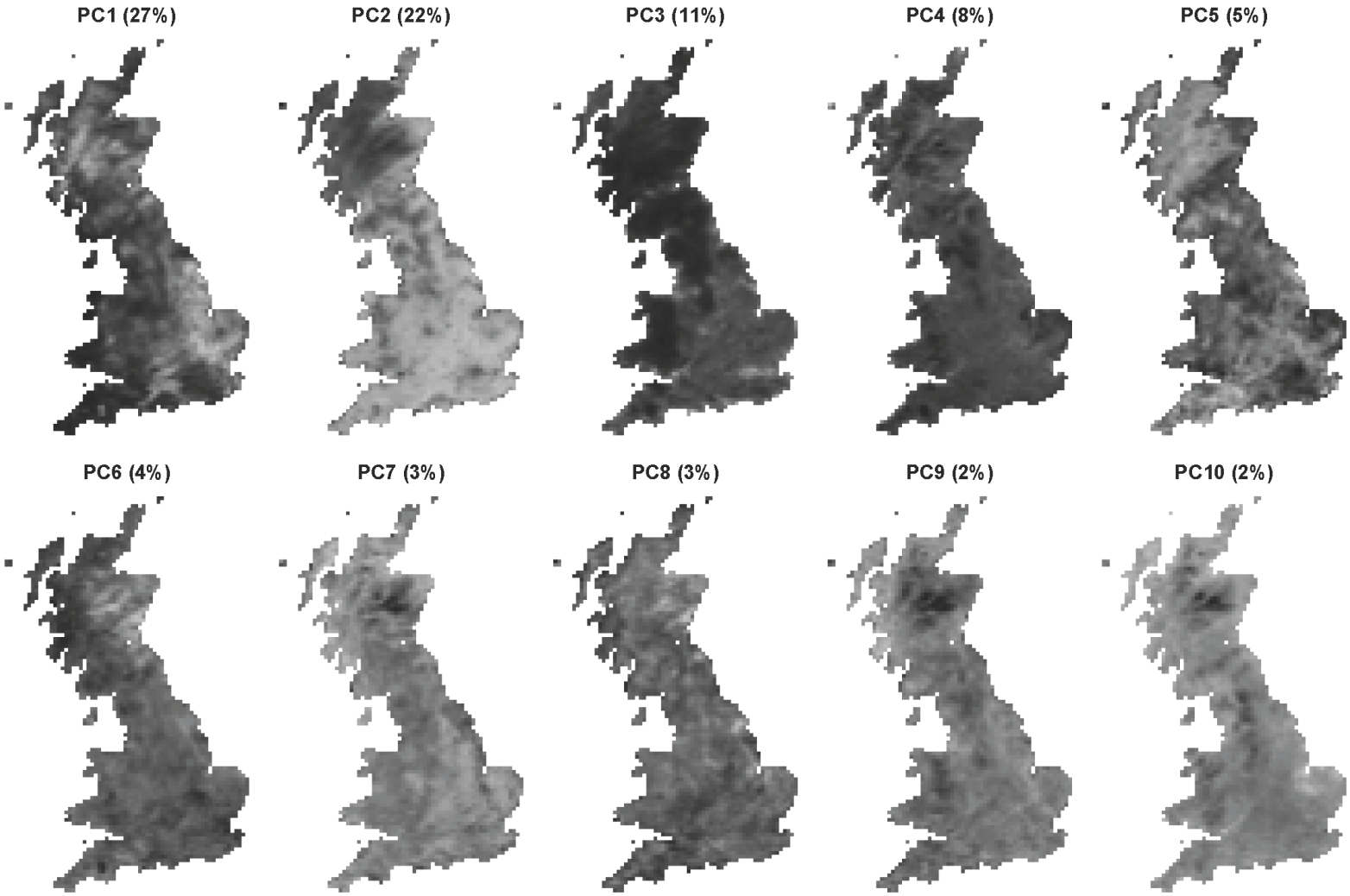


Figure S3.3: Spatial distribution of principal components in Great Britain. Darker shades represent higher values of the component. Proportion of variance in TFA dataset explained by each component is given in brackets.

Appendix S4. Demonstration of GRaF R package

Introduction

We demonstrate here some features of the GRaF R package for fitting species distribution models using Gaussian random fields. As an example, we model the distribution of Bog Myrtle (*Myrica gale*) in Great Britain using data detailed in the methods section and Appendix B. We assume that both the GRaF package and the raster package are installed and that the file `BogMyrtleData.RData` is the working directory.

Loading and extracting data

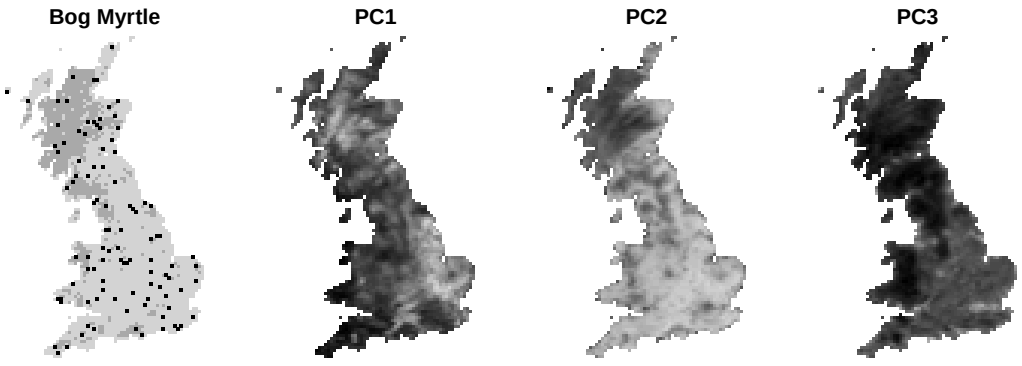
We load these packages and data:

```
> library(GRaF)
> library(raster)
> load('BogMyrtleData.RData')
```

`BogMyrtleData.RData` contains three objects: `PCs` - a `RasterBrick` object containing 10 Principal Components environmental layers; `BM` - a `RasterLayer` object with the known distribution of Bog Myrtle at 10km resolution, and `points` - a `SpatialPoints` object (from the `sp` package, loaded by `raster`) with the 300 randomly selected grid squares used in the model comparison.

We only use the first 100 of these random datapoints and the first 3 environmental layers. We plot the random points over the known distribution along with the environmental covariates:

```
> par(mfrow = c(1, 4), mar = c(1, 1, 2, 1), oma = c(2, 0, 2, 0))
> greys <- colorRampPalette(c('light grey', 'black'))
> plot(BM, col = c('light grey', 'dark grey'), axes = F, box = F,
        legend = F, main = 'Bog Myrtle')
> plot(points[1:100], add = T, pch = '.', cex = 3)
> plot(PCs, 1, col = greys(50), axes = F, box = F, legend = F)
> plot(PCs, 2, col = greys(50), axes = F, box = F, legend = F)
> plot(PCs, 3, col = greys(50), axes = F, box = F, legend = F)
```



To fit the models we extract the environmental data and occurrences for the 100 random points:

```
> y <- extract(BM, points[1:100])
> x <- extract(PCs[[1:3]], points[1:100])
> x <- data.frame(x)
```

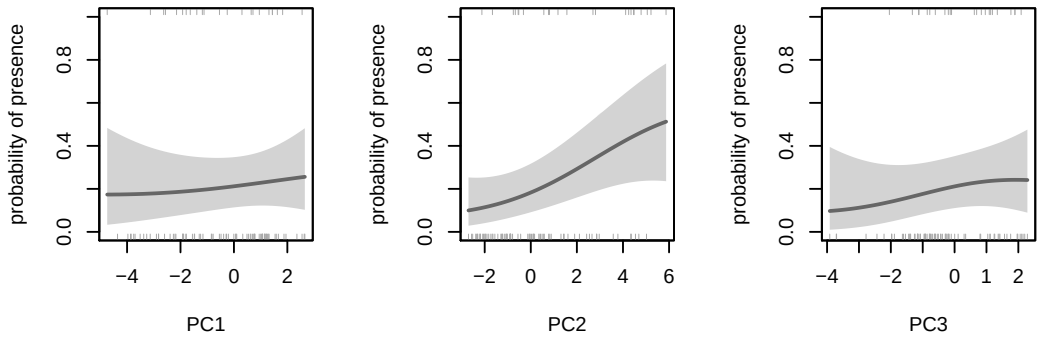
Fitting and visualising GRaF models

To begin with we fit a `graf` model to this data without tuning the hyperparameters by setting `opt.l = FALSE`. When we do this `graf` makes a guess at appropriate hyperparameters from the data. Details of how it does this are given in the helpfile, which can be accessed using `?graf`.

```
> m1 <- graf(y, x, opt.l = FALSE)
```

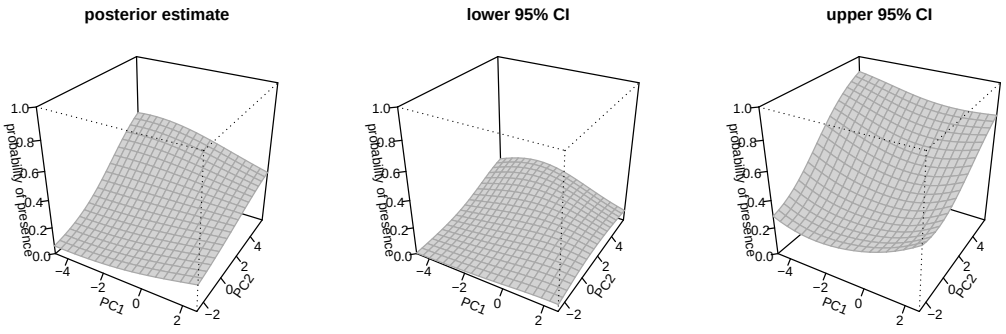
We set up the plotting panel and use `graf`'s `plot` function to visualise the fitted random field along each of the covariates, along with 95% credible intervals and the presence-absence data shown on the axes. See `?plot.graf` for more plotting options.

```
> par(mfrow = c(1, 3))
> plot(m1)
```



We visualise the interaction between the first two covariates with a 3-dimensional perspective plot by using the `plot3d` function:

```
> par(mfrow = c(1, 3))
> plot3d(m1, c(1, 2), prior = FALSE)
```



Note that we suppress plotting of the mean function which in this case defaults to a flat surface.

Model complexity

We can calculate the deviance information criterion (DIC) of the model using the `DIC` function.

```
> DIC(m1)
      DIC        pD
99.387577  4.605478
```

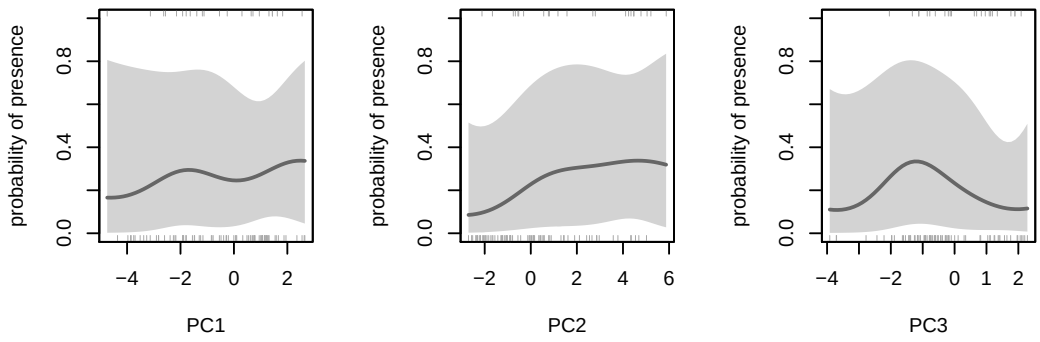
`DIC` also returns the number of *effective parameters* of the model (pD). This is a measure of the complexity of the model and is analagous to the number of parameters in standard statistical models such as logistic regression.

We can adjust the complexity of the model by setting the lengthscale hyperparameters using the `l` argument. We access the lengthscales from our first model and create a model with much shorter lengthscales (a more complex model) and a model with much longer lengthscales (a less complex model):

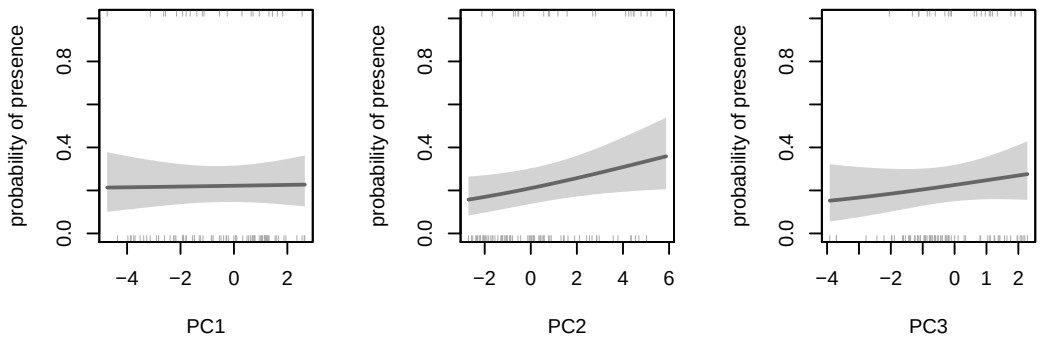
```
> m1$l
[1] 8.069940 8.759522 6.820797
> m2 <- graf(y, x, opt.l = FALSE, l = m1$l * 0.1)
> m3 <- graf(y, x, opt.l = FALSE, l = m1$l * 10)
> DIC(m2)
     DIC       pD
99.96790 13.66754
> DIC(m3)
       DIC         pD
102.476963   2.107745
```

We visualise the surfaces fitted by these models:

```
> par(mfrow = c(1, 3))
> plot(m2)
```



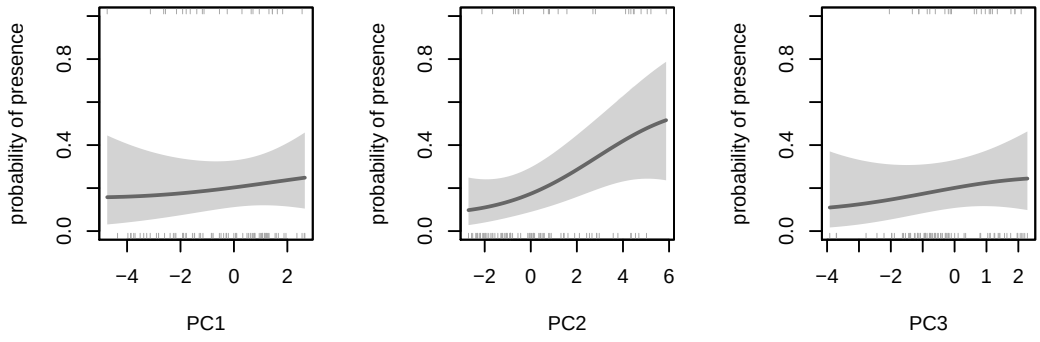

```
> par(mfrow = c(1, 3))
> plot(m3)
```



Optimising the lengthscales

In practice we would prefer to calculate an optimal value for the lengthscales and we can do this by setting `opt.l = TRUE`. `graf` now runs multiple models in an optimisation routine and therefore takes longer to run. Because this optimisation trades complexity off against fit to the data, it guards against model overfitting in a similar way to regularisation methods such as the lasso.

```
> par(mfrow = c(1, 3))
> m4 <- graf(y, x, opt.l = TRUE)
> m4$l
[1] 10.821711  6.769081 11.956881
> DIC(m4)
      DIC        pD
99.029125  4.204193
> plot(m4)
```



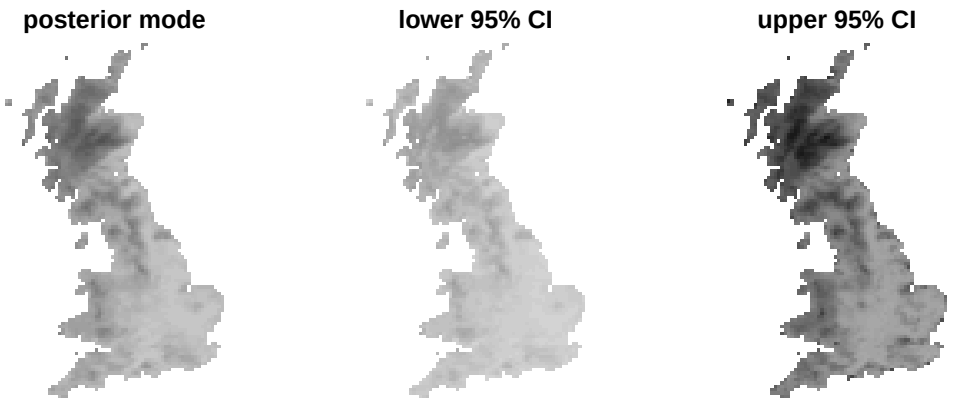
Making predictions

To generate predictions from this model for the rest of Great Britain we extract the relevant environmental covariates and use `graf`'s `predict` function (see `?predict.graf` for details).

```
> unmask <- which(!is.na(PCs[[1]][]))
> xall <- extract(PCs[[1:3]], 1:ncell(PCs))[unmask, ]
> p <- predict(m1, data.frame(xall))
> head(p)
     posterior mode lower 95% CI upper 95% CI
[1,]      0.3590668    0.1934960    0.5569247
[2,]      0.3560689    0.1936734    0.5503185
[3,]      0.3206399    0.1838494    0.4876237
[4,]      0.3027491    0.1671409    0.4730940
[5,]      0.2971340    0.1443506    0.4982656
[6,]      0.2776231    0.1517646    0.4400118
```

As well as a posterior mode - the 'best guess' prediction - lower and upper 95% credible intervals around these predicitons are also provided. We can plot these predictions by putting them back into a raster file:

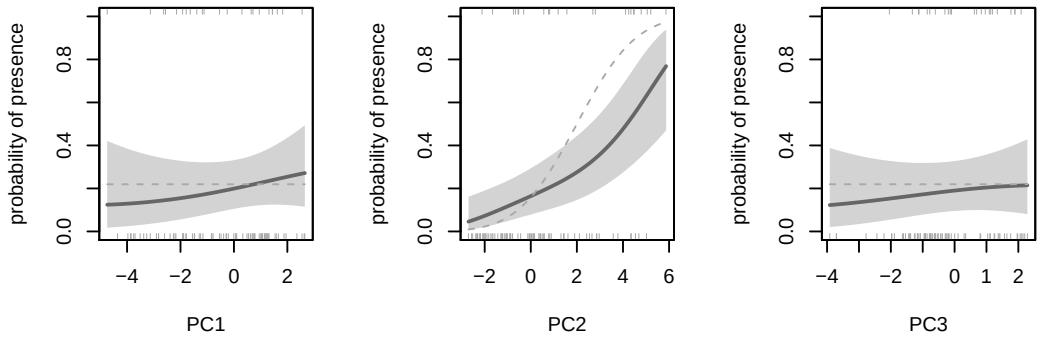
```
> pred <- PCs[[1:3]]
> pred[][unmask,] <- p
> plot(pred, col = greys(50), zlim = c(0, 1), nc = 3,
         box = F, axes = F, legend = F, main = colnames(p))
```



Adding prior ecological knowledge

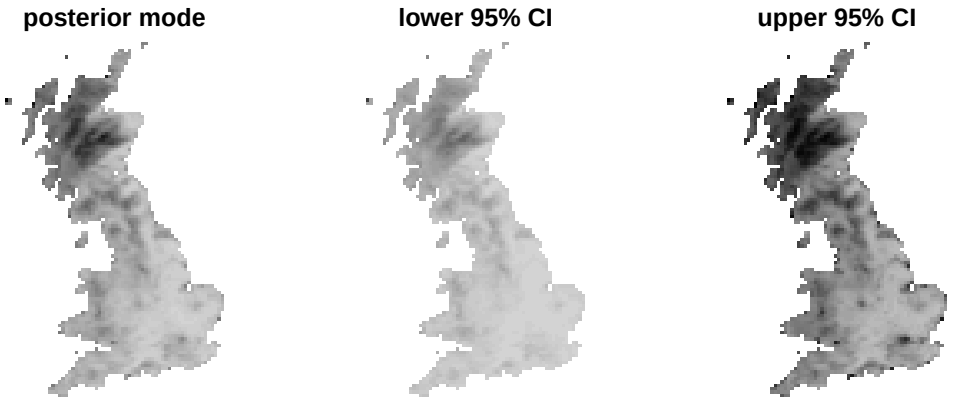
We can augment the distribution data with knowledge of the ecolgoy of Bog Myrtle. As its name would suggest, Bog Myrtle displays a preference for upland bogs, a habitat with which PC2 is positively correlated (see Appendix B). We specify a simple linear probit model which assigns high probability of presence to sites with a high value of PC2, and low probability of presence elsewhere. We plot the fitted surfaces, with the prior overlaid as a dashed line:

```
> pri <- function(x) pnorm(-1 + 0.5 * x$PC2)
> m5 <- graf(y, x, opt.l = TRUE, prior = pri)
> par(mfrow = c(1, 3))
> plot(m5, prior = T)
```



the predicted surface is a compromise between our prior knowledge of the species' ecology and what can be determined from the data. We plot the predictions from this model:

```
> pred[][unmask,] <- predict(m5, data.frame(xall))
> plot(pred, col = greys(50), zlim = c(0, 1), nc = 3,
         box = F, axes = F, legend = F, main = colnames(p))
```



References

Dallaire, P., Besse, C. & Chaib-draa, B. (2011) An approximate inference with Gaussian process to latent functions from uncertain data. *Neurocomputing*, **74**, 1945–1955.

Elith, J., Kearney, M. & Phillips, S.J. (2010) The art of modelling range-shifting species. *Methods in Ecology and Evolution*, **1**, 330–342.

Farr, T.G., Rosen, P.A., Caro, E., Crippen, R., Duren, R., Hensley, S., Kobrick, M., Paller, M., Rodriguez, E., Roth, L., Seal, D., Shaffer, S., Shimada, J., Umland, J., Werner, M., Oskin, M., Burbank, D. & Alsdorf, D. (2007) The shuttle radar topography mission. *Review of Geophysics*, **45**.

Fuller, R., Smith, G. & Sanderson, J. (2002) Countryside Survey 2000 Module 7. Land Cover Map 2000. Final Report. Technical Report March 2002.

Gilbert, M., Mitchell, A., Bourn, D., Mawdsley, J., Clifton-Hadley, R. & Wint, G.R.W. (2005) Cattle movements and bovine tuberculosis in Great Britain. *Nature*, **435**, 491–6.

Hoffman, M.D. & Gelman, A. (2011) The No-U-Turn Sampler : Adaptively Setting Path Lengths in Hamiltonian Monte Carlo. *arXiv*, **1111**.

Johnson, D., Hay, S.I. & Rogers, D.J. (1998) Contemporary environmental correlates of endemic bird areas derived from meteorological satellite sensors. *Philosophical Transactions of the Royal Society of London Series B, Biological sciences*, **265**, 951–959.

Minka, T. (2001) *A family of algorithms for approximate Bayesian inference*. Phd thesis, Massachusetts Institute of Technology.

Preston, C., Pearman, D. & Dines, T. (2002) *New atlas of the British and Irish flora: an atlas of the vascular plants of Britain, Ireland, the Isle of Man and the Channel Islands*. Oxford University Press, Oxford.

Rasmussen, C. & Willians, C. (2006) *Gaussian processes for machine learning*. MIT Press.

Rogers, D.J., Randolph, S.E., Snow, R.W. & Hay, S.I. (2002) Satellite imagery in the study and forecast of malaria. *Nature*, **415**, 710–715.

Scharlemann, J.P.W., Benz, D., Hay, S.I., Purse, B.V., Tatem, A.J., Wint, G.R.W. & Rogers, D.J. (2008) Global data for ecology and epidemiology: a novel algorithm for temporal Fourier processing MODIS data. *PloS ONE*, **3**, e1408.

Stan Development Team (2013) Stan: A c++ library for probability and sampling, version 2.1.

Vanhatalo, J., Riihimäki, J., Hartikainen, J., Pasi, J., Tolvanen, V. & Vehtari, A. (2012) Bayesian modeling with Gaussian processes using the GPstuff toolbox. *arXiv*, **1206**.