```
# Supplementary Data S1 - R code for the variance components model
# Accompanies xxx_full_citation_xxx to come here
# File prepared by AJ Tanentzap (ajt65@cam.ac.uk), 26 Nov 2017
#
#
#
#
#
#
chlorophyll_driver_model <- "data{
// HERE WE DEFINE THE DATA THAT WE INPUT INTO STAN
  int<lower=0> N;
// number of observations for chlorophyll fluorescence
  int<lower=0> Nlakes;
// total number of lakes
  int<lower=0> Nlakecars;
// total number of lake-specific characteristics inputted to the model
  vector[N] y;
// observed chlorophyll fluorescence
  int lake[N];
// lake ID associated with each chlorophyll fluorescence observation
  int month[N];
// month ID associated with each chlorophyll fluorescence observation
  int day[N];
// day ID associated with each chlorophyll fluorescence observation
  int<lower=0> surf_wtemps_N;
// number of observations for surface water temperatures
  vector[surf_wtemps_N] surf_wtemps_y;
// observed surface water temperatures
  int surf_wtemps_lake[surf_wtemps_N];
// lake ID associated with each surface water temperature observation
  int surf_wtemps_month[surf_wtemps_N];
// month ID associated with each surface water temperature observation
  int surf_wtemps_day[surf_wtemps_N];
// day ID associated with each surface water temperature observation
  int<lower=0> wind_spds_N;
// number of wind speed observations
  vector<lower=0>[wind_spds_N] wind_spds_y;
// observed wind speeds
  int wind_spds_lake[wind_spds_N];
// lake ID associated with each wind speed observation
  int wind_spds_month[wind_spds_N];
// month ID associated with each wind speed observation
  int wind_spds_day[wind_spds_N];
// day ID associated with each wind speed observation
  int<lower=0> par_N;
// number of PAR observations
  vector[par_N] par_y;
// observed PAR
  int par_lake[par_N];
// lake ID associated with each PAR observation
  int par_month[par_N];
// month ID associated with each PAR observation
  int par_day[par_N];
// day ID associated with each PAR observation
  matrix[Nlakes,Nlakecars] lakechars;
// matrix of lake-specific characteristics, such as depth, area, and trophic
state
  matrix[Nlakecars+12,Nlakecars+12] Imat;
// identity matrix used to setup prior for correlation matrix
```

```
  vector[Nlakecars] mvn_known_mu;
// means calculated across lakes for each lake-specific characteristic
  vector[Nlakecars] mvn_known_sd;
// standard deviations calculated across lakes for each lake-specific
characteristic
  int thetamax;
// index in vector of wind speeds with the largest value
  int thetamin;
// index in vector of wind speeds with the smallest value
  int theta_id[par_N-2];
// indicies in the vector of wind speeds that do not include thetamax or
thetamin
}
parameters{
// HERE WE DEFINE THE PARAMETERS THAT WE ESTIMATE IN STAN
  corr_matrix[Nlakecars+12] Omega;
// correlation matrix
  real betaINT[Nlakes];
// lake-specific intercepts for chlorophyll fluorescence
  matrix[Nlakes,12] betaLAKEMONTH;
// variation in chlorophyll fluorescence associated with each month in each
lake
  matrix[Nlakes,366] betaLAKEDAY;
// variation in chlorophyll fluorescence associated with each day in each lake
  real surf_wtemps_betaINT[Nlakes];
// lake-specific intercepts for surface water temperatures
  matrix[Nlakes,12] surf_wtemps_betaLAKEMONTH;
// variation in surface water temperatures associated with each month in each
lake
  matrix[Nlakes,366] surf_wtemps_betaLAKEDAY;
// variation in surface water temperatures associated with each day in each
lake
  real wind_spds_betaINT[Nlakes];
// lake-specific intercepts for wind speed
  matrix[Nlakes,12] wind_spds_betaLAKEMONTH;
// variation in wind speed associated with each month in each lake
  matrix[Nlakes,366] wind_spds_betaLAKEDAY;
// variation in wind speed associated with each day in each lake
  matrix[Nlakes,2] par_betaINT;
// lake-specific intercepts for PAR
  matrix[Nlakes,12] par_betaLAKEMONTH;
// variation in PAR associated with each month in each lake
  matrix[Nlakes,366] par_betaLAKEDAY;
// variation in PAR associated with each day in each lake
  real<lower=0,upper=1> theta[par_N-2];
// probability of membership into one of two groups for PAR mixing model for
all but two observations for which we define groups to improve fitting
  vector[12] mvn_est_mu;
// means calculated across lakes for each variance component on log-scale
  vector<lower=0>[12] mvn_est_sd;
// standard deviations calculated across lakes for each variance component on
log-scale
  matrix[Nlakes,12] log_varcomp;
// estimated variance component at each of 3 time scales for each of 4
high-resolution variables in each lake
  real<lower=0> int_sd;
// hyper-standard deviation for lake-specific intercepts for chlorophyll
fluorescence
  real int_mu;
// hyper-mean for lake-specific intercepts for chlorophyll fluorescence
```

```
  real<lower=0> int_sd_surfwtemps;
// hyper-standard deviation for lake-specific intercepts for surface water
temperature
  real int_mu_surfwtemps;
// hyper-mean for lake-specific intercepts for surface water temperature
  real<lower=0> int_sd_wind_spds;
// hyper-standard deviation for lake-specific intercepts for wind speed
  real int_mu_wind_spds;
// hyper-mean for lake-specific intercepts for wind speed
  real<lower=0> int_sd_par[2];
// hyper-standard deviation for lake-specific intercepts for PAR
  real int_mu_par[2];
// hyper-mean for lake-specific intercepts for PAR
}
transformed parameters{
// HERE WE DEFINE THE PARAMETERS THAT WE CALCULATE IN STAN
   matrix[Nlakecars+12,Nlakecars+12] LL;
// Cholesky deocmposed covariance matrix
   vector[N] mu;
// mean chlorophyll fluorescence
   vector[N] si;
// standard deviation in chlorophyll fluorescence
   vector[surf_wtemps_N] surf_wtemps_mu;
// mean surface water temperature
   vector[surf_wtemps_N] surf_wtemps_si;
// standard deviation in surface water temperature
   vector[wind_spds_N] wind_spds_mu;
// mean wind speed
   ordered[2] par_mu[par_N];
// mean PAR in each of two groups
   row_vector[Nlakecars+12] lakeparams[Nlakes];
// transformed matrix of lake-specific characteristics and variance components
in each lake for each of estimation
   matrix[Nlakes,Nlakecars+12] lakeparams_tmp;
// concatenated matrix of lake-specific characteristics and variance
components in each lake
   matrix<lower=0>[Nlakes,12] varcomp;
// exponentiated log_varcomp, i.e. estimated variance component at each of 3
time scales for each of 4 high-resolution variables in each lake
   vector[Nlakecars+12] mvn_append_mu;
// concatenated means of lake-specific characteristics and variance components
averaged across all lakes
   vector[Nlakecars+12] mvn_append_sd;
// concatenated standard deviations of lake-specific characteristics and
variance components averaged across all lakes
   real<lower=0,upper=1> theta_N[par_N];
// concatenated probability of group membership
   lakeparams_tmp = append_col(log_varcomp,lakechars);
// concatenate logged variance components with lake-specific characteristics
   for (i in 1:18)
// this transforms for ease of estimation
    lakeparams[i] = lakeparams_tmp[i];
   varcomp = exp(log_varcomp);
// back-transform logged variance components onto normal scale
   mvn_append_mu = append_row(mvn_est_mu, mvn_known_mu);
// concatenate means of lake-specific characteristics and variance components
averaged across all lakes
   mvn_append_sd = append_row(mvn_est_sd, mvn_known_sd);
// concatenate standard deviations of lake-specific characteristics and
variance components averaged across all lakes
```

```
   LL = cholesky_decompose(quad_form_diag(Omega,mvn_append_sd));
// Cholesky decompose of correlation matrix Omega and estimated hyper-standard
deviations for each variance component
   for (i in 1:N){
// calculate mean chlorophyll fluorescence
    mu[i] = int_mu + betaINT[lake[i]]*int_sd +
betaLAKEDAY[lake[i],day[i]]*varcomp[lake[i],2] +
betaLAKEMONTH[lake[i],month[i]]*varcomp[lake[i],1];
    si[i] = varcomp[lake[i],3];
   }
   for (i in 1:surf_wtemps_N){
// calculate mean surface water temperature
    surf_wtemps_mu[i] = int_mu_surfwtemps +
surf_wtemps_betaINT[surf_wtemps_lake[i]]*int_sd_surfwtemps +
surf_wtemps_betaLAKEDAY[surf_wtemps_lake[i],surf_wtemps_day[i]]*varcomp[surf_
wtemps_lake[i],5] +
surf_wtemps_betaLAKEMONTH[surf_wtemps_lake[i],surf_wtemps_month[i]]*varcomp[s
urf_wtemps_lake[i],4];
    surf_wtemps_si[i] = varcomp[surf_wtemps_lake[i],6];
   }
   for (i in 1:wind_spds_N)
// calculate mean wind speed
    wind_spds_mu[i] = int_mu_wind_spds +
int_sd_wind_spds*wind_spds_betaINT[wind_spds_lake[i]] +
wind_spds_betaLAKEDAY[wind_spds_lake[i],wind_spds_day[i]]*varcomp[wind_spds_l
ake[i],8] +
wind_spds_betaLAKEMONTH[wind_spds_lake[i],wind_spds_month[i]]*varcomp[wind_sp
ds_lake[i],7];
   theta_N[thetamax] <- 0;
// assign group membership to largest and smallest wind speed values to help
fitting and avoid chains converging on opposing membership
   theta_N[thetamin] <- 1;
   for (i in 1:(par_N-2))
    theta_N[theta_id[i]] <- theta[i];
// fill vector of theta values with estimated values
   for (i in 1:par_N){
// calculate mean PAR in each group of the mixture
    par_mu[i,1] = int_mu_par[1]+int_sd_par[1]*par_betaINT[par_lake[i],1] +
par_betaLAKEDAY[par_lake[i], par_day[i]]*varcomp[par_lake[i],11] +
par_betaLAKEMONTH[par_lake[i], par_month[i]]*varcomp[par_lake[i],10];
    par_mu[i,2] = int_mu_par[2]+int_sd_par[2]*par_betaINT[par_lake[i],2] +
par_betaLAKEDAY[par_lake[i], par_day[i]]*varcomp[par_lake[i],11] +
par_betaLAKEMONTH[par_lake[i], par_month[i]]*varcomp[par_lake[i],10];
   }
 }
model{
// HERE WE DEFINE THE PRIOR LIKELIHOODS THAT WE ESTIMATE IN STAN
  mvn_est_mu ~ normal(0,1);
  mvn_est_sd ~ normal(0,5);
  int_mu ~ normal(0,5);
  int_sd ~ normal(0,1);
  int_mu_surfwtemps ~ normal(0,10);
  int_sd_surfwtemps ~ normal(0,5);
  int_mu_wind_spds ~ normal(0,5);
  int_sd_wind_spds ~ normal(0,1);
  int_mu_par ~ normal(0,5);
  int_sd_par ~ normal(0,1);
  Omega ~ inv_wishart(Nlakecars+12+1, Imat);
  lakeparams ~ multi_normal_cholesky(mvn_append_mu, LL);
  betaINT ~ normal(0,1);
```

```
  to_vector(betaLAKEMONTH) ~ normal(0,1);
  to_vector(betaLAKEDAY) ~ normal(0,1);
  surf_wtemps_betaINT ~ normal(0,1);
  to_vector(surf_wtemps_betaLAKEMONTH) ~ normal(0,1);
  to_vector(surf_wtemps_betaLAKEDAY) ~ normal(0,1);
  wind_spds_betaINT ~ normal(0,1);
  to_vector(wind_spds_betaLAKEMONTH) ~ normal(0,1);
  to_vector(wind_spds_betaLAKEDAY) ~ normal(0,1);
  to_vector(par_betaINT) ~ normal(0,100);
  to_vector(par_betaLAKEMONTH) ~ normal(0,1);
  to_vector(par_betaLAKEDAY) ~ normal(0,1);
  y ~ normal(mu, si);
  surf_wtemps_y ~ normal(surf_wtemps_mu, surf_wtemps_si);
  for (i in 1:wind_spds_N)
   wind_spds_y[i] ~ normal(wind_spds_mu[i], varcomp[wind_spds_lake[i],9])
T[0,];
  theta ~ beta(5,5);
  for (n in 1:par_N)
   target += log_mix(theta_N[n], normal_lpdf(par_y[n] | par_mu[n,1],
varcomp[par_lake[n],12]), normal_lpdf(par_y[n] | par_mu[n,2],
varcomp[par_lake[n],12]));
}
"
```