



The University of Edinburgh

School of Geosciences

**COMPARISON OF TOP-DOWN AND BOTTOM-UP
APPROACHES ON SPECIFIC LEAF AREA PATTERNS,
AT GLOBAL, LATITUDINAL AND BIOME SCALES**

By

ANNA CHIRUMBOLO

in partial fulfilment of the requirement
for the degree of BSc with Honours
in Ecological and Environmental Sciences

May 2020

Abstract

(the spacing is set to 1.5)

no more than 250 words for the abstract

- a description of the research question/knowledge gap – what we know and what we don't know
- how your research has attempted to fill this gap
- a brief description of the methods
- brief results
- key conclusions that put the research into a larger context

Contents

1	Introduction	1
1.1	Research Questions	1
2	Methods	3
2.1	Datasets	3
2.1.1	<i>Plant Functional Traits datasets and publications</i>	3
2.1.2	<i>Other datasets and databases</i>	3
2.2	Data manipulation	4
2.3	Visual analysis	5
2.4	Statistical analysis	6
2.5	Programming	6
3	Results	8
4	Discussion	9
5	Conclusion	10
6	Bibliography	11
7	Appendices	12
7.1	Appendix A	12
7.2	Appendix B	13

Acknowledgements

bla bla

1 Introduction

- introduce the reader to the subject area and clarify the knowledge gap that the dissertation research will fill.
- set the context for the dissertation by reviewing the relevant literature.
- include relevant references to general (theoretical papers and reviews) and specific (specific to the particular question addressed) literature, to justify the research that has been undertaken and define the questions being addressed.
- state the primary research questions and hypotheses in the final paragraph.
- follow an ‘inverted triangle’ format, progressing from general scientific ideas and why they matter to the specific research questions addressed in the dissertation project.

The introduction should not be just a ‘Literature Review’:

1.1 Research Questions

1. Is there a strong degree of overlap between bottom-up and top-down mean and uncertainty SLA estimates, and is it constant at differing spatial extents?

H0. There is no strong degree of overlap, nor at differing spatial extents;

H1. The degree of overlap is strong at varying spatial extents;

H2. The degree of overlap varies with spatial extents.

2. Is there a difference in degree of overlap for SLA mean and uncertainty estimates, among biomes with differing leaf seasonalities (i.e. deciduous versus evergreen)? / Do continent-specific characteristics affect the degree of overlap among biomes?

H0. There is no difference in degree of overlap among biomes with differing leaf seasonalities;

H1. There is a high difference in degree of overlap among biomes with differing leaf seasonalities;

H2. The difference in overlap varies with specific biomes.

3. Do the two SLA mean and uncertainty estimates have similar sensitivities to climate data, i.e. mean global precipitation and mean temperature?

H0. The SLA estimates have no similar sensitivities to climate data;

H1. The SLA estimates have highly similar sensitivities to climate data;

H2. The SLA estimates have low similarity in sensitivity to climate data.

2 Methods

2.1 Datasets

2.1.1 *Plant Functional Traits datasets and publications*

I retrieved the estimates of interest from Bloom *et al* (2015) and Butler *et al.* (2017) publications (Appendix A).

The mean and uncertainty estimates from Bloom *et al.* (2015) relate to Leaf Carbon mass per Mean Area (gC/m^2). This variable was calculated from a fusion (DALEC2) of MODIS satellite imagery of leaf area and biomass, and a Markov Chain Monte Carlo MDF algorithm, to cover for absence of empirical baseline data, over the period ranging from 2001 to 2010. The mean and related uncertainties are derived from 4,000 samples within each pixel. The variables are mapped at a $1^\circ \times 1^\circ$ (360×180) spatial resolution. This data can be downloaded from datashare.is.ed.ac.uk/handle/10283/875.

Mean and standard deviation estimates from Butler *et al.* (2017) relate to Specific Leaf Area (m^2/kg). This data was calculated from the fusion of the largest global Categorical Traits dataset (TRY) - a collection of world-wide field-based data sampling as of 2016 - and a Bayesian framework. The mean and standard deviation variables are presented as a summary of the full log-normal distribution within each pixel. Most measurements are derived from a subset of global regions, i.e. North America, Europe, Australia, China, Japan and Brazil. The variables are mapped at a $0.5^\circ \times 0.5^\circ$ (720×180) spatial resolution.

Both datasets present only one time dimension. I used the dataset from Bloom *et al.* (2015) as the “benchmark” to which I compared the “model” dataset from Butler *et al.* (2017).

2.1.2 *Other datasets and databases*

I used the Ecoregions17 database to retrieve the spatial extents of the 14 major biomes across the world (fig. 2). The dataset is available to download from ecoregions2017.appspot.com/. I used *rworldmap* package from R, to obtain the spatial extents of the 6 continents. I also used historical climate data (i.e. mean annual temperature and total annual precipitation), averaged for the years 1970-2000 at 10 minutes spatial resolution ($\sim 340 \text{ km}^2$), from WorldClim database (Appendix A). These datasets can be downloaded from worldclim.org/data/worldclim21.html (BIO01 and BIO12).

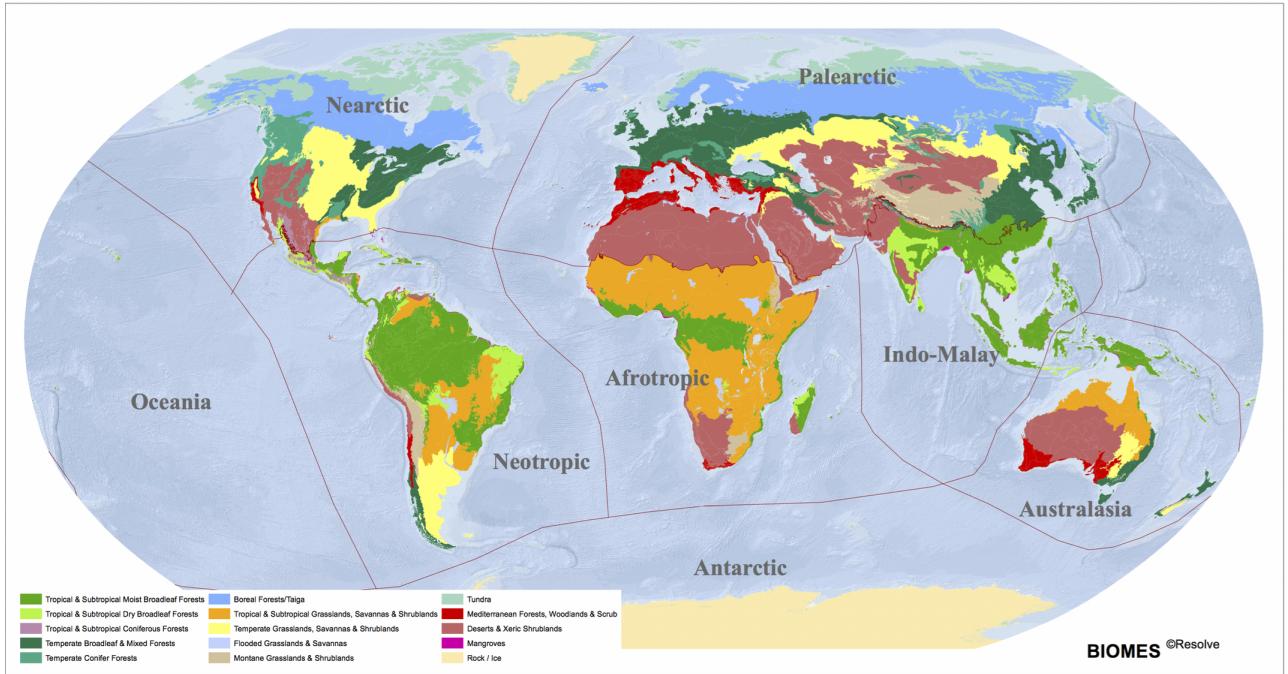


Figure 1: 14 major biomes as retrieved from Supplementary Materials by Dinerstein et al. (2017)

2.2 Data manipulation

The state of the estimates at retrieval required standardisation in order to be compared. I aggregated the spatial resolution of SLA from Butler *et al.* (2017) so that it matched with the $1^\circ \times 1^\circ$ from Bloom *et al.* (2015). I converted LCMA (gC/m^2), from Bloom *et al.* (2015), to SLA (m^2/kg) [1]. I did so knowing that LCMA measures the carbon content within the leaf, thus representing 50% of the its dry-biomass equivalent (LMA) *REF*, and that SLA is the inverse of LMA. Unit conversion from gC/m^2 to m^2/kg was also necessary to ensure standardisation of the two SLA estimates. I executed all conversions within the Linux environment, though this can be done with any programming language at disposition. I carried out this calculation for both mean and standard deviation:

$$[1] \text{SLA} = \frac{1}{2 \times \text{LCMA}}$$

I created subsets of the two SLA datasets at three different spatial extents: by latitudinal gradients, by biomes across the world and by biomes split further by continent. The splitting by latitude mainly takes into account the climatic differences observed from tropics to polar regions. I determined the range of the different climatic zones following convention, thus extracting SLA data at spatial extents of tropical latitudes between 23.5°S and 23.5°N , sub-tropics between 23.5°S/N and 35°S/N , temperate

between 35°S/N and 66.5°S/N and North pole between 66.5°N and 90°N. There were no data available for the South pole from Bloom *et al.* (2015) and Butler *et al.* (2017). The splitting by biome takes into account the vegetation dominance and also climatic variability across the world, thus increasing the subdivision of data points. Furthermore, the splitting of biomes by continents considers the geographical variability between the various land masses, in conjunction with the vegetation dominance and the climatic differences I extracted the spatial polygons representing the 14 major biomes, and those representing the 6 continents, from Ecoregions17 and *rworldmap*, respectively. I masked the two SLA datasets over these spatial polygons, so as to exclude all data points not confined within the polygon coordinates.

Lastly, I prepared the climate datasets from WorldClim by changing their original spatial resolution ($0,1667^\circ \times 0,1667^\circ$) to match that of the SLA datasets ($1^\circ \times 1^\circ$).

2.3 Visual analysis

In order to visualise the spatial distribution of the SLA mean and standard deviation estimates, I first set them at the same scale. Following, I created heatscatters to show the relationship between mean and standard deviation estimates. Heatscatters are a useful visualisation tool to observe the correlation between two continuous variables, and assess the concentration of the data points in the space with a color encoded Kernel density estimation. I also produced density plots grouped by dataset, and I calculated the percentage of the area of overlap between the two density curves. I could relate this percentage to the count of stiples within the stippling maps I later produced. The stippling process involved calculating which mean and standard deviation estimates of the model fell within the 25th and 95th confidence interval of mean SLA of the benchmark. I represented this by plotting spatially the model mean and stdev data and overlaying the results from the stippling. This map is useful for showing the spatial distribution of the model estimates which are more closely related to those of the benchmark. SHOULD MENTION THE SCATTERPLOT OF THE DATA POINTS GROUPED BY LAT/BIOMES/BIOMESBYCONT!!! STILL TO DO. I repeated the analysis at the four different spatial extents, assuming that differences in overlap between benchmark and model would be more evident when accounting for environmental and geographical variability across the world. When testing the sensitivity of the SLA datasets with climate data, I first plotted the spatial correlation between SLA estimates and climate data. I then created a 2d density graph, where I could visualise the spatial distributions of the SLA estimates in relation to climate and also the extent to which these overlapped. I then calculated the percentage overlap between the two density distributions.

2.4 Statistical analysis

I tested the degree of overlap between mean and uncertainty estimates of the SLA estimates, by executing the following statistical analyses - R² [2] or coefficient of determination, Root Mean Square Error (RMSE) [3] and bias [4]. R² is a useful parameter for testing what proportion of variance from the dependent variable (the model) is predictable by the independent variable (the benchmark). I extracted the R² from simple linear regression models I ran. I considered only R² values from model results with a p-value smaller than 0.001.

$$[2] R^2 = \frac{\sum_{i=1} (y_{z,i} - \bar{y}_i)^2}{\sum_{i=1} (y_i - \bar{y}_i)^2} *$$

I calculated RMSE to assess the average magnitude of the error when correlating Bloom *et al.*, again as benchmark, and Butler *et al.* (2017). RMSE was a more appropriate statistical parameter than Mean Absolute Error, as it gives higher weight to undesired large errors.

$$[3] \text{RMSE} = \sqrt{\frac{\sum_{i=1}^n (y_i - x_i)^2}{n}} *$$

Assuming that these SLA estimates are biased, being retrieved from different approaches, I retrieved the bias to estimate the average difference between SLA benchmark and the underlying SLA model. The result from the bias is an indicator of the quality of the methods for collecting and calculating the data.

$$[4] \text{Bias} = \frac{y - x}{2} *$$

Where x is the benchmark, y is the model, y_z is the predicted y from the linear regression equation, and n is the count

2.5 Programming

I executed the greater portion of data visualisation and analysis with R on Rstudio. The most important packages used for observing the relationships between SLA datasets are the following:

- *ncdf4*, *sf*, *tiff* and *raster* for visualisation and manipulation of spatial data;
- *LSD* for creating heatscatters of the relationship between the two data distributions;

- *overlapping* for calculating the percentage overlap of the data density distributions with one another.
- *metrics* for calculating RMSE and Bias.

I created or modified a variety of functions in order to automatise the coding process. These can be found in Appendix B.

3 Results

this section should summarise the findings of the research referring to all figures, tables and statistical results (some of which may be placed in appendices). - include the primary results, ordered logically - it is often useful to follow the same order as presented in the methods. - alternatively, you may find that ordering the results from the most important to the least important works better for your project. - data should only be presented in the main text once, either in tables or figures; if presented in figures, data can be tabulated in appendices and referred to at the appropriate point in the main text.

Often, it is recommended that you write the results section first, so that you can write the methods that are appropriate to describe the results presented. Then you can write the discussion next, then the introduction which includes the relevant literature for the scientific story that you are telling and finally the conclusions and abstract – this approach is called writing backwards.

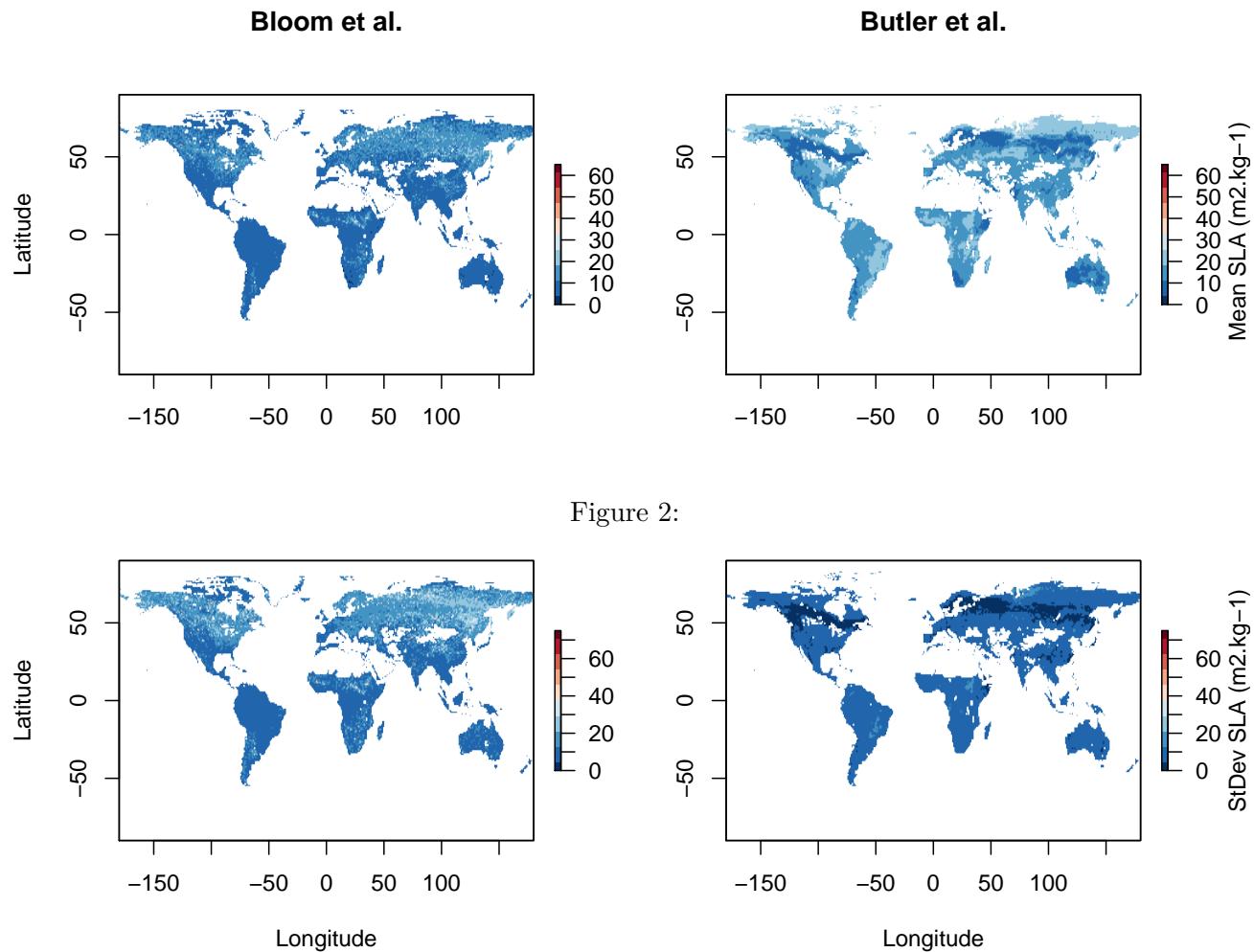


Figure 2:

4 Discussion

the purpose of the discussion is to summarise your major findings and place them in the context of the current state of knowledge in the literature. When you discuss your own work and that of others, back up your statements with evidence and citations. - The first part of the discussion should contain a summary of your major findings (usually 2 – 4 points) and a brief summary of the implications of your findings. Ideally, it should make reference to whether you found support for your hypotheses or answered your questions that were placed at the end of the introduction. - The following paragraphs will then usually describe each of these findings in greater detail, making reference to previous studies. - Often the discussion will include one or a few paragraphs describing the limitations of your study and the potential for future research. - Subheadings within the discussion can be useful for orienting the reader to the major themes that are addressed.

5 Conclusion

the conclusions section should specify the key findings of your study, explain their wider significance in the context of the research field and explain how you have filled the knowledge gap that you have identified in the introduction. This is your chance to present to your reader the major take-home messages of your dissertation research. It should be similar in content to the last sentence of your summary abstract. It should not be a repetition of the first paragraph of the discussion. They can be distinguished in their connection to broader issues. The first paragraph of the discussion will tend to focus on the direct scientific implications of your work (i.e. basic science, fundamental knowledge) while the conclusion will tend to focus more on the implications of the results for society, conservation, etc.

6 Bibliography

7 Appendices

7.1 Appendix A

This appendix contains the spatial representation of LCMA and SLA mean and standard deviation values (fig. 1), as retrieved from Bloom *et al.* (2015) and Butler *et al.* (2017), respectively. It also contains the climate data, i.e. total annual precipitation (mm) and mean annual temperature ($^{\circ}\text{C}$), retrieved from WorldClim v2.1 database (fig. 2).

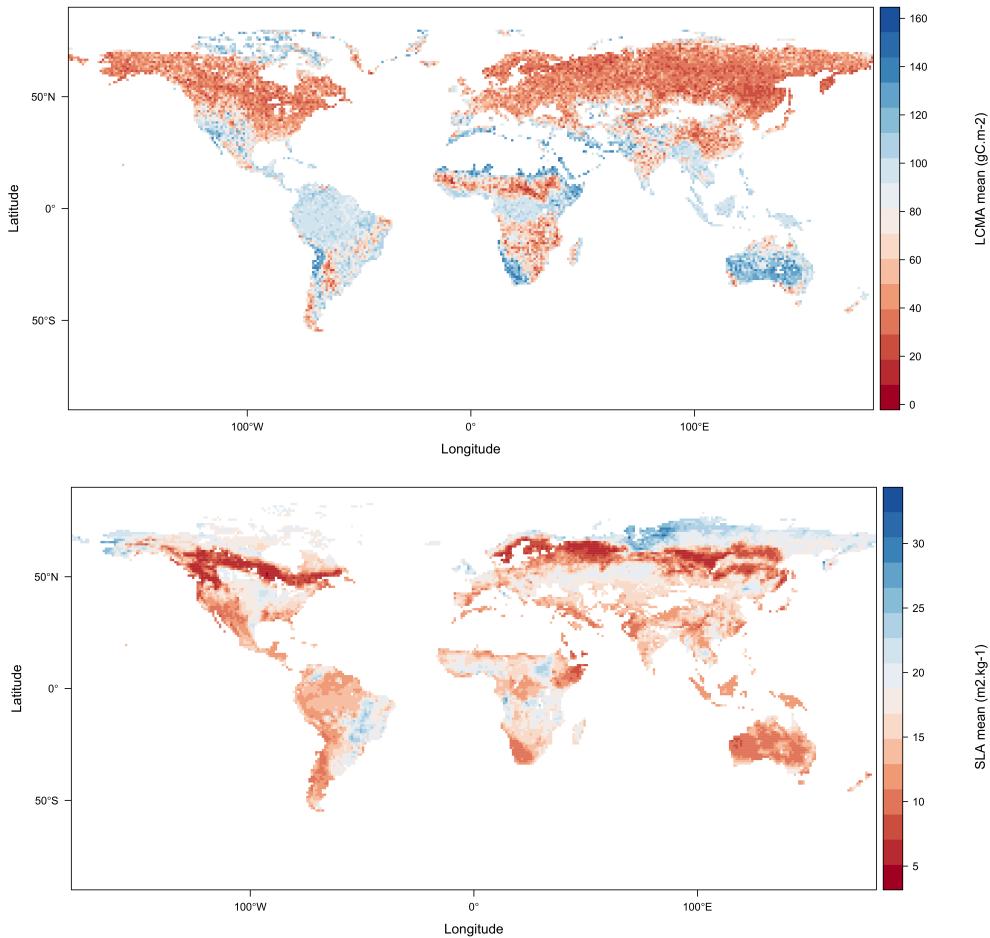


Figure 3: (Left) global distribution of LCMA mean estimates, from Bloom et al. (2015)'s publication. (Right) global distribution of SLA mean estimates from Butler et al. (2017)'s publication.

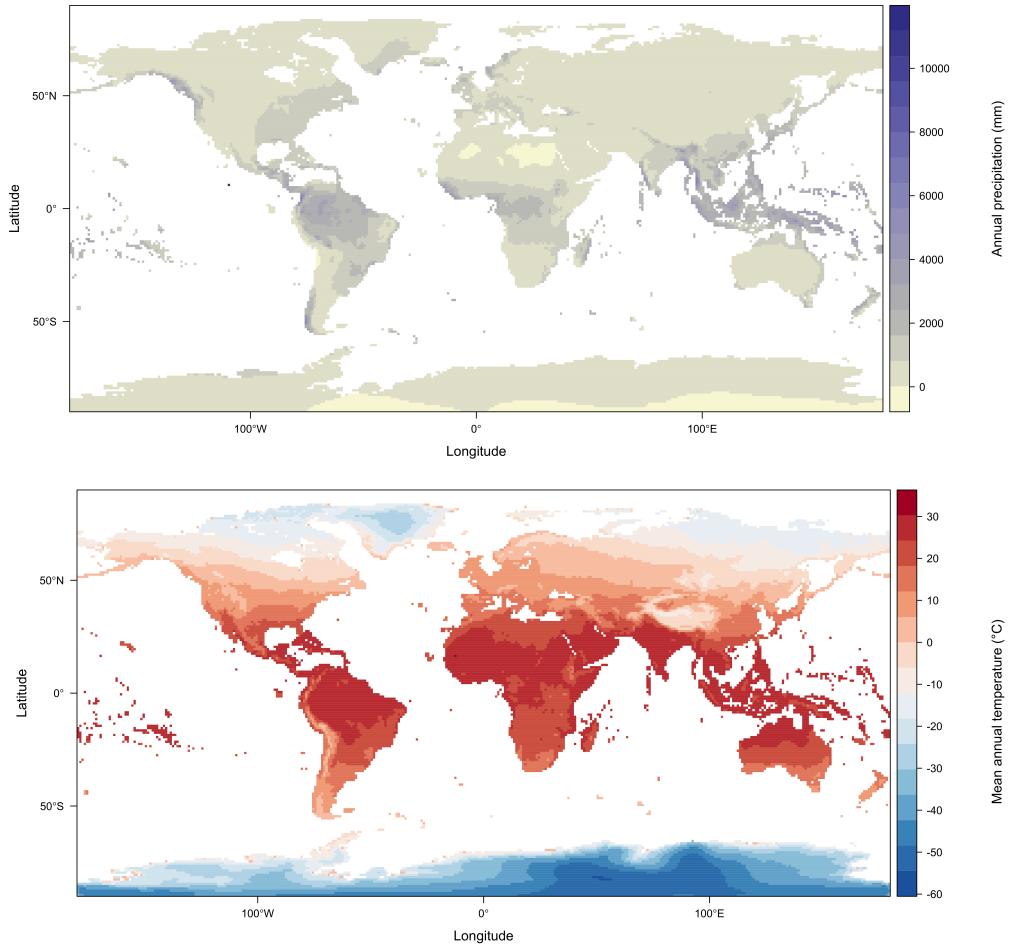


Figure 4: (Top) Total annual precipitation (mm). (Bottom) Mean annual temperature (°C). Both datasets are averaged over 1970-2000 at 10 minutes spatial resolution. From WorldClim v2.1.

7.2 Appendix B

This appendix contains the code for the custom functions used during this project, in order to facilitate automation of data manipulation and visualisation.

```
## custom functions ####

# convert raster to data frame
mask.to.df <- function(x){

  new.list <- list()

  for (i in 1:length(x)){

    df <- raster::as.data.frame(x[[i]], xy =TRUE)
    name.df <- paste("df",names(x)[i],sep = ".")
```

```

    new.list[[name.df]] <- df
  }

  new.list
}

# joining dataframes

join.f <- function(x,y){

  new.list <- list()

  join <- mapply(left_join, x, y,SIMPLIFY = FALSE)
  join <- lapply(join, na.omit)

  name.df <- names(x)[i]

  new.list[[name.df]] <- join
}

# masking the biomes by continent

mask.biome.f <- function(x,y){

  new.list <- list()

  for (i in 1:length(x)){
    for (j in 1:length(y)){
      mask.biome <- raster::mask(x[[i]],y[[j]])

      if (!is.infinite(mask.biome@data@min)&!is.infinite(mask.biome@data@max)){
        name <- paste(names(x)[i],names(y)[j],sep = " ")
        new.list[[name]] <- mask.biome
      }
      else
        NULL
    }
  }
  new.list
}

# density plots with percentage overlap (modif. from overlapping package)

# for SLA mean

```

```

my.final.plot.sla <- function (x, OV = NULL){

  AREA <- NULL

  for (i1 in 1:(length(x) - 1)) {
    for (i2 in (i1 + 1):(length(x))) {
      A <- data.frame(x = x[[i1]], group = names(x)[i1],
                        k = paste(names(x)[i1], names(x)[i2], sep = "-",
                        collapse = ""))
      B <- data.frame(x = x[[i2]], group = names(x)[i2],
                        k = paste(names(x)[i1], names(x)[i2], sep = "-",
                        collapse = ""))
      AREA <- rbind(AREA, rbind(A, B))
    }
  }

  if (!is.null(OV)) {
    OV <- data.frame(OV = OV, k = names(OV))

    AREA <- merge(AREA, OV, by = "k")

    AREA$k <- paste0(AREA$k, " (ov. perc. ", round(AREA$OV *
                                              100), ")")
  }

  ggplot(AREA, aes(x = x)) + facet_wrap(~k) +
    geom_density(aes(fill = AREA$group), alpha = 0.35) +
    xlab("\nSLA mean (m2.kg-1)") +
    ylab("") +
    theme_classic() +
    theme(legend.title = element_blank()) +
    scale_color_brewer(palette = "Set1") +
    scale_x_continuous(expand = c(0,0)) +
    scale_y_continuous(expand = c(0,0))
}

my.overlap.sla <- function (x, nbins = 1024, plot = FALSE, partial.plot = FALSE,
                           boundaries = NULL, ...){

  if (is.null(names(x)))

```

```

names(x) <- paste("Y", 1:length(x), sep = "")
dd <- OV <- FUNC <- DD <- xpoints <- COMPTITLE <- NULL

for (j in 1:length(x)) {
  if (!is.null(boundaries)) {
    Lbound <- lapply(boundaries, FUN = length)
    if ((Lbound$from == 1) & (Lbound$to == 1)) {
      warning("Boundaries were set all equals")
      boundaries$from <- rep(boundaries$from, length(x))
      boundaries$to <- rep(boundaries$to, length(x))
    }
    else {
      if ((Lbound$from != length(x)) | (Lbound$to != length(x))) {
        stop("Boundaries not correctly defined")
      }
    }
    from = boundaries$from[j]
    to = boundaries$to[j]
    dj <- density(x[[j]], n = nbins, from = from, to = to,
                  ...)
  }
  else {
    dj <- density(x[[j]], n = nbins, ...)
  }
  ddd <- data.frame(x = dj$x, y = dj$y, j = names(x)[j])
  FUNC <- c(FUNC, list(with(ddd, approxfun(x, y))))
  dd <- rbind(dd, ddd)
}
for (i1 in 1:(length(x) - 1)) {
  for (i2 in (i1 + 1):(length(x))) {
    comptitle <- paste0(names(x)[i1], "-", names(x)[i2])
    dd2 <- data.frame(x = dd$x, y1 = FUNC[[i1]](dd$x),
                      y2 = FUNC[[i2]](dd$x))
  }
}

```

```

dd2[is.na(dd2)] <- 0

dd2$ovy <- apply(dd2[, c("y1", "y2")], 1, min)
dd2$ally <- apply(dd2[, c("y1", "y2")], 1, max, na.rm = TRUE)
dd2$dominance <- ifelse(dd2$y1 > dd2$y2, 1, 2)
dd2$k <- comptitle

OV <- c(OV, sum(dd2$ovy, na.rm = TRUE)/sum(dd2$ally,
                                              na.rm = TRUE))

dd2 <- dd2[order(dd2$x), ]

CHANGE <- dd2$x[which(dd2$dominance[2:nrow(dd2)] !=

                           dd2$dominance[1:(nrow(dd2) - 1)])]

xpoints <- c(xpoints, list(CHANGE))

if (partial.plot) {

  gg <- ggplot(dd2, aes(x, dd2$y1)) + theme_bw() +
    geom_vline(xintercept = CHANGE, lty = 2, color = "#cccccc") +
    geom_line() + geom_line(aes(x, dd2$y2)) +
    geom_line(aes(x, dd2$ovy), color = "red") +
    geom_line(aes(x, dd2$ally), color = "blue") +
    ggtitle(comptitle) +
    xlab("") +
    ylab("") +
    theme(plot.title = element_text(hjust = 0.5),
          legend.title = element_blank())

  print(gg)

}

DD <- rbind(DD, dd2)

COMPTITLE <- c(COMPTITLE, comptitle)

}

names(xpoints) <- names(OV) <- COMPTITLE

if (plot)

  print(my.final.plot.sla(x, OV))

return(list(DD = DD, OV = OV, xpoints = xpoints))
}

```

```

# for SLA standard deviation

my.final.plot.std <- function (x, OV = NULL){

  AREA <- NULL

  for (i1 in 1:(length(x) - 1)) {

    for (i2 in (i1 + 1):(length(x))) {

      A <- data.frame(x = x[[i1]], group = names(x)[i1],
                        k = paste(names(x)[i1], names(x)[i2], sep = "-",
                        collapse = ""))
      B <- data.frame(x = x[[i2]], group = names(x)[i2],
                        k = paste(names(x)[i1], names(x)[i2], sep = "-",
                        collapse = ""))
      AREA <- rbind(AREA, rbind(A, B))
    }
  }

  if (!is.null(OV)) {

    OV <- data.frame(OV = OV, k = names(OV))

    AREA <- merge(AREA, OV, by = "k")

    AREA$k <- paste0(AREA$k, " (ov. perc. ", round(AREA$OV *
                                              100), ")")

  }

  ggplot(AREA, aes(x = x)) + facet_wrap(~k) +
    geom_density(aes(fill = AREA$group), alpha = 0.35) +
    xlab("\nSLA StDev (m2.kg-1)") +
    ylab("") +
    theme_classic() +
    theme(legend.title = element_blank()) +
    scale_color_brewer(palette = "Set1") +
    scale_x_continuous(expand = c(0,0)) +
    scale_y_continuous(expand = c(0,0))

}

my.overlap.std <- function (x, nbins = 1024, plot = FALSE, partial.plot = FALSE,
                           boundaries = NULL, ...){

```

```

if (is.null(names(x)))
  names(x) <- paste("Y", 1:length(x), sep = "")
dd <- OV <- FUNC <- DD <- xpoints <- COMPTITLE <- NULL
for (j in 1:length(x)) {
  if (!is.null(boundaries)) {
    Lbound <- lapply(boundaries, FUN = length)
    if ((Lbound$from == 1) & (Lbound$to == 1)) {
      warning("Boundaries were set all equals")
      boundaries$from <- rep(boundaries$from, length(x))
      boundaries$to <- rep(boundaries$to, length(x))
    }
    else {
      if ((Lbound$from != length(x)) | (Lbound$to != length(x))) {
        stop("Boundaries not correctly defined")
      }
    }
    from = boundaries$from[j]
    to = boundaries$to[j]
    dj <- density(x[[j]], n = nbins, from = from, to = to,
                  ...)
  }
  else {
    dj <- density(x[[j]], n = nbins, ...)
  }
  ddd <- data.frame(x = dj$x, y = dj$y, j = names(x)[j])
  FUNC <- c(FUNC, list(with(ddd, approxfun(x, y))))
  dd <- rbind(dd, ddd)
}
for (i1 in 1:(length(x) - 1)) {
  for (i2 in (i1 + 1):(length(x))) {
    comptitle <- paste0(names(x)[i1], "-", names(x)[i2])
    dd2 <- data.frame(x = dd$x, y1 = FUNC[[i1]](dd$x),

```

```

y2 = FUNC[[i2]](dd$x))

dd2[is.na(dd2)] <- 0

dd2$ovy <- apply(dd2[, c("y1", "y2")], 1, min)
dd2$ally <- apply(dd2[, c("y1", "y2")], 1, max, na.rm = TRUE)
dd2$dominance <- ifelse(dd2$y1 > dd2$y2, 1, 2)

dd2$k <- comptitle

OV <- c(OV, sum(dd2$ovy, na.rm = TRUE)/sum(dd2$ally,
                                              na.rm = TRUE))

dd2 <- dd2[order(dd2$x), ]

CHANGE <- dd2$x[which(dd2$dominance[2:nrow(dd2)] !=

                      dd2$dominance[1:(nrow(dd2) - 1)])]

xpoints <- c(xpoints, list(CHANGE))

if (partial.plot) {

  gg <- ggplot(dd2, aes(x, dd2$y1)) + theme_bw() +
    geom_vline(xintercept = CHANGE, lty = 2, color = "#cccccc") +
    geom_line() + geom_line(aes(x, dd2$y2)) +
    geom_line(aes(x, dd2$ovy), color = "red") +
    geom_line(aes(x, dd2$ally), color = "blue") +
    ggtitle(comptitle) +
    xlab("") +
    ylab("") +
    theme(plot.title = element_text(hjust = 0.5),
          legend.title = element_blank())

  print(gg)

}

DD <- rbind(DD, dd2)

COMPTITLE <- c(COMPTITLE, comptitle)

}

names(xpoints) <- names(OV) <- COMPTITLE

if (plot)

  print(my.final.plot.std(x, OV))

return(list(DD = DD, OV = OV, xpoints = xpoints))

```

```

}

## extracting p-values from lm summary (credits to Stephen Turner):
#gettinggeneticsdone.blogspot.com/2011/01/
#rstats-function-for-extracting-f-test-p.html

lmp <- function (modelobject) {

  if (class(modelobject) != "lm") stop("Not an object of class 'lm' ")

  if (modelobject[["df.residual"]]!=0){ # if the residual is not = 0
    f <- summary(modelobject)$fstatistic
    p <- pf(f[1] ,f[2] ,f[3] ,lower.tail=F)
    attributes(p) <- NULL
    return(p)
  }
  else
    modelobject <- NULL
}

# perform calculations of statistics iteratively
# linear regression

lm.f <- function(x){

  lm <- lapply(x, function(dat) lm(dat[,4] ~ dat[,3] ,data=dat))
}

# step-by-step calculation of r-squared, and of RMSE and bias

stats.f <- function(x){

  lapply(x,function(df){

    df$c_mean <- mean(df$cardamom)

    df$b_mean <- mean(df$butler)

    df$diff_butler <- df$butler-df$b_mean

    df$diff_butler2<- df$diff_butler^2

    df$sum_diff_butler2 <-sum(df$diff_butler2)

    df$slope_bf <-sum((df$cardamom-df$c_mean)*(df$butler-df$b_mean))/
      sum((df$cardamom-df$c_mean)^2)
  })
}

```

```

df$b_intercept <- df$b_mean - (df$slope_bf*df$c_mean)

df$new_b_val <- df$b_intercept + (df$slope_bf*df$cardamom)

df$dist_mean_new_b <- df$new_b_val - df$b_mean

df$sqrddist_b <- df$dist_mean_new_b^2

df$sum_sqrddist_b <- sum(df$sqrddist_b)

df$sla_r2 <- df$sum_sqrddist_b / df$sum_diff_butler2

df$bias_av <- bias(df$butler, df$cardamom)

df$bias_row <- df$butler - df$cardamom

df$rmse_av <- rmse(df$butler, df$cardamom)

df$rmse_row <- sqrt((df$butler - df$cardamom)^2)

df

})

}

# function for kableExtra, from Michael Harper (not modified):
#stackoverflow.com/questions/28166168/
#how-to-change-fontface-bold-italics-for-a-cell-in-a-kable-table-in-rmarkdown

format_cells <- function(df, rows ,cols, value = c("italics", "bold", "strikethrough")){

  # select the correct markup

  map <- setNames(c("*", "*", "~`~"), c("italics", "bold", "strikethrough"))

  markup <- map[value]

  for (r in rows){

    for(c in cols){

      # Make sure values are not factors

      df[[c]] <- as.character( df[[c]])

      # Update formatting

      df[r, c] <- paste0(markup, df[r, c], markup)

    }

  }

}

```

```
    return(df)
}
```