Supplementary Material 8: Seed Germination Over Latitude Notebook

Withheld

This is the notebook for all code used in the Seed Germination Over Latitude Project

Submitted to Journal of Global Ecology and Biogeography

Contents

Load packages for this markdown file

- We use 'dplyr', 'ggpubr' and 'metafor' in this document
- 'knitr' and 'rmarkdown' are used to create the pdf and html versions

```
library(dplyr)
library(ggpubr)
library(metafor)
```

Data Cleaning Methods

- We start with data downloaded from the Millennium Seed Bank Project (MSBP)
  - Downloaded 16/11/17 (Seed age data downloaded 26/09/19)
  - Data sourced via Millennium Seed Bank Partnership Data Warehouse http://brahmsonline.kew.org/msbp
- Remove duplicates, negative germination IDs, rows with NumberSown = 0
- Add in temperature information
- Make new ID which is Grid.ID - Species_Genus_Lat_Long (Species*Site)
- More information in script: ./R/MSBPDataCleaning.R
- Number of records after initial cleaning: 35456
- Number of different species after initial cleaning: 3867

```
source("./R/MSBPDataCleaning.R") #More info in script
```

Perform Germination Models on each Species*Site

- Details in script (./R/MSBPQuadraticModels.R) and in Supplement 1
- Post model filtering in script

```
source("./R/MSBPQuadraticModels.R") #More info in script
```

Add Climate Data from WorldClim, and clean data for analysis

- Current: "worldclim", var="bio", res=10
- Future: 'CMIP5', var='bio', res=10, rcp=85, model='AC', year=70
- Altitude from worldclim as well
- Neaten up the dataset to allow for analyses
- Number of species*sites after post model cleaning: 1427
- Number of different species after cleaning: 1312
- Number of records used to make final dataset: 9737

```
source("./R/MSBPClimateData.R") #More info in script
```

Data Analysis and Figures

Figure 1.

- Hypothesised Germination Temperature Breadths,
- Map of Species Locations,
- Forecasted Temperature Increase
- No code (not made in R)

Figure 2.

- Example of species records and model estimates
- Max, Topt upper, Topt lower, Min

Figure 3.a Germination Breadth over Latitude

- With phylogeny (with and without hemisphere)
- Without phylogeny (with and without hemisphere)
- Hemisphere not significant:
  - without phy p = 0.8234198
  - with phy p = 0.8832945
- Plot (goes to combination plot)
- Tables to Supp 2.

```
source("./R/germToleranceBreadth.R") #Analysis script
source("./R/metaforR2.R") #Allows for calculation of pseudo R-squared from metafor objects

TB <- getToleranceBreadth(fromfile = T) #Set to T to not rerun everything, F to rerun

#Abslat with phylogeny
TB$Model.phy

#R-squared marginal(fixed effects) and conditional (fixed and random effects)
R2(TB$Model.phy)

#Abslat without phylogeny
TB$Model

#R-squared marginal(fixed effects) and conditional (fixed and random effects)
R2(TB$Model)
```

Figure 3.b Current Climate Mismatch over Latitude

- With phylogeny (with and without hemisphere)
- Without phylogeny (with and without hemisphere)
- Hemisphere not significant:
  - without phy p = 0.7569279
  - with phy p = 0.7143421
- Plot (goes to combination plot)
- Tables to Supp 2.
- In tropics:
  - Above Topt.upp: 37/142
  - Within Topt breadth: 32/142
- Outside tropics:
  - Above Topt.upp: 34/351
  - Within Topt breadth: 47/351
- Above 40:
  - Above Topt.upp: 2/111
  - Within Topt breadth: 6/111

```
source("./R/germClimateMismatchCurrent.R")
source("./R/metaforR2.R") #Allows for calculation of pseudo R-squared from metafor objects

CM <- getCM(fromfile = T) #Set to T to get results from file, set to F to rerun analyses

#Abslat with phylogeny
CM$Model.phy

#R-squared marginal(fixed effects) and conditional (fixed and random effects)
R2(CM$Model.phy)

#Abslat Without phylogeny
CM$Model

#R-squared marginal(fixed effects) and conditional (fixed and random effects)
R2(CM$Model)

#Species at 60N
predict.rma(CM$Model,
            newmods=c(AbsLat = 60,
                      SeedAge = log10(1479),
                      altitude = 393),
            addx=T)

#Species at 60N - lower
predict.rma(CM$Model.low,
            newmods=c(AbsLat = 60,
                      SeedAge = log10(1479),
                      altitude = 393),
            addx=T)

#Species at 5N
predict.rma(CM$Model,
            newmods=c(AbsLat = 5,
                      SeedAge = log10(1479),
                      altitude = 393),
            addx=T)
```

Figure 3.c- Future Warming Risk over Latitude

- With phylogeny (with hemisphere)
- Without phylogeny (with hemisphere)
- Plot
- In tropics:
  - Above Tmax: 41/190
- Outside tropics:
  - Above Tmax: 50/632
- Above 40:
  - Above Tmax: 2/192

```
source("./R/germWarmingRiskFuture.R")
source("./R/metaforR2.R") #Allows for calculation of pseudo R-squared from metafor model objects
```

```
WR <- getWR(fromfile = T) #Set to T to not rerun everything, set to F to rerun

#Abslat and hemipshere with phylogeny
WR$Model.h.phy

#R-squared marginal(fixed effects) and conditional (fixed and random effects)
R2(WR$Model.h.phy)

#Abslat and hemipshere without phylogeny
WR$Model.h

#R-squared marginal(fixed effects) and conditional (fixed and random effects)
R2(WR$Model.h)

#Species at 40N
predict.rma(WR$Model.h,
            newmods=c(AbsLat = 40,
                      SeedAge = log10(1449),
                      altitude = 142,
                      NorthTF = TRUE),
            addx=T)

#Species at 5N
predict.rma(WR$Model.h,
            newmods=c(AbsLat = 5,
                      SeedAge = log10(1449),
                      altitude = 393,
                      NorthTF = TRUE),
            addx=T)
```

Figure 3. Combined graphs (Figure 3 a, b, c)

Figure 4a,b - Warming Risk longevity, woodiness

```
source("./R/germLifeWoodWR.R")

WR.wood.life$Wood.Model

WR.wood.life$Wood.Model.Phy

WR.wood.life$Long.Model

WR.wood.life$Long.Model.Phy

f4combPlot <- ggarrange((WR.wood.life$Wood.Plot + theme(plot.margin = margin(10,30,10,30))),
                        (WR.wood.life$Long.Plot + theme(plot.margin = margin(10,30,10,30))),
                        labels = c("a", "b"), ncol=2, nrow=1)

f4combPlot
```

```
saveRDS(WR.wood.life, "./Outputs/woodlife")

ggsave(f4combPlot,
       filename = "./Outputs/f4combPlot.tiff",
       units="in",
       width=6, height=4, dpi=1000, compression = 'lzw')
```

Figure 5 - Warming Risk Phylogenetic Tree, and text for phylogenetic signal

- Phylogentic tree show warming risk per species
- Test for phylogenetic signal of future warming risk
  - Moran's I:
  - Blomberg's K:

```
source("./R/germTree.R") #Get tree - more detail in script
source("./R/germPhyloSignal.R") #Get phylogentic signals - more detail in script

pclades
```

Appendices

Maximum, upper optimum, lower optimum, and minimum germination temperatures over latitude

```
source("./R/germMaxMinOpt.R")
```

Future (2070) climate mismatch over latitude

```
source("./R/germClimateMismatchFuture.R")

CMF$Plot

CMF$Model.h.Phy
```

Current warming risk over latitude

```
source("./R/germWarmingRiskCurrent.R")

WRC$Plot

WR.c$Model.Phy
```

Methods: We used 9737 records for 1312 species from the Kew Gardens' global germination database to quantify global patterns in germination temperature. Results: We found no evidence for a latitudinal gradient in the breadth of temperatures at which plant species can germinate. However, tropical plants are predicted to face the greatest risk from climate warming, because they experience temperatures closer to their upper germination limits. By 2070, over half (79/142) of tropical plant species are predicted to experience temperatures exceeding their optimum germination temperatures, with some even exceeding their maximum germination temperature (41/190). Conversely, (95% of species at latitudes above 45° are predicted to benefit from warming, with environmental temperatures shifting closer to the species' optimal germination temperatures.

Other

Intro Analysis

- Warming over latitude script ./R/getWarmingPerLat.R
  - Warming at:
    - 60N: 5.7674689
    - 5N: 3.9132302
    - 33S: 2.7758781
- Using climate sources as above, averaged over 1 degree on latitude

Methods Numbers

- Records after duplicate records: 164 428
- "We excluded 848 out of the 2275 species*location values. . . "
- Tolerance breadth: 476 species*locations, 443 species, 276 locations
- Current climate mismatch: 476 species*locations, 443 species, 276 locations
- Future warming risk: 776 species*locations, 723 species, 392 locations
- Woody: 96
- Herbaceous: 270
- Annual:51
- Perennial:158

Scripts used for analyses

Scripts to markdown using knitr::spin

MSBPDataCleaning.R

```
# This script details the data consolidation and cleaning process for the MSBP
# data sets

library(tidyr)
library(ggplot2)
library(dplyr)
library(readxl)

# We start wih the data sets downloaded from the MSBP website -

# Download 1: AccessionNumber, Family, Taxon, Latitude, Longitude, Cultivated (T/F)
# Download 2: AccessionNumber,  Genus, Sp1 (species), Country, Altitude
# Download 3: Accession Number, BrahmsGermTestId, StepSummary
# Download 4: BrahmsGermTestId, Rate, InspectionInterval, Interval
# Download 5: BrahmsGermTestId, Sown, Germinated, Family
# Download 6: BrahmsGermTestId, Fresh, Empty, Infested, Abnormal
# Download 7: BrahmsGermTestId, Unusable, Mouldy, TestNotes

# 1 and 2 are accession info
#
# 3-7 are germination information
#
# See MSBP Data Standards for more information (other varibles possible, many
# blanks at this stage though)

MSBP_DL1 <-
  read_excel("./Data/171117MSBPdownload/2017-11-16_231507939-Download_1.xlsx")
```

```
MSBP_DL2 <-
  read_excel("./Data/171117MSBPdownload/2017-11-16_231507939-Download_2.xlsx")
MSBP_DL3 <-
  read_excel("./Data/171117MSBPdownload/2017-11-16_233020054-Download_3.xlsx")
MSBP_DL4 <-
  read_excel("./Data/171117MSBPdownload/2017-11-16_233902116-Download_4.xlsx")
MSBP_DL5 <-
  read_excel("./Data/171117MSBPdownload/2017-11-16_234637337-Download_5.xlsx")
MSBP_DL6 <-
  read_excel("./Data/171117MSBPdownload/2017-11-16_234948032-Download_6.xlsx")
MSBP_DL7 <-
  read_excel("./Data/171117MSBPdownload/2017-11-16_235202784-Download_7.xlsx")

# Merge Data sets and create raw data files
# - This is done by row, as this was downloaded from the same large dataset.
#   Checks are done to ensure rows match up.

Accession <- bind_cols(MSBP_DL1, MSBP_DL2)

Germ <- bind_cols(MSBP_DL3, MSBP_DL4, MSBP_DL5, MSBP_DL6, MSBP_DL7)

#Save Raw, combined data - Accession
Accession_Raw <- select(Accession,-AccessionNumber1)
write.csv(Accession_Raw, file = "./Data/Accession_Raw.csv")

#Save Raw, combined data - Germination
Germ_Raw <-
  select(
    Germ,
    -BrahmsGermTestId1,
    -BrahmsGermTestId2,
    -BrahmsGermTestId3,
    -BrahmsGermTestId4
  )
write.csv(Germ_Raw, file = "./Data/Germ_Raw.csv")


#  Start to clean germination data set
#
#
#  Start with the following variables:
#    AccessionNumber, BrahmsGermTestID, StepSummary, NumSown, NumGerm
#
#  Steps:
#    - Start with:
#    AccessionNumber, BrahmsGermTestID, StepSummary, NumSown, NumGerm
#  - Remove Duplicates
#  - Make BrahmsGermTestID, NumSown, NumGerm numbers
#  - Remove Negative Germination IDs
#  - Remove rows with Number Sown = 0
#  - Rationalised duplicate GermIDs

#Starting data set
```

```r
Germ_Cleaning <- select(Germ_Raw, one_of(
  c(
    "AccessionNumber",
    "BrahmsGermTestId",
    "StepSummary",
    "NumSown",
    "NumGerm"
  )
))


#Remove duplicates
Germ_Cleaning <- distinct(Germ_Cleaning)

#Makes columns number, rows with characters as NA
Germ_Cleaning <- transform(
  Germ_Cleaning,
  BrahmsGermTestId = as.numeric(BrahmsGermTestId),
  NumSown = as.numeric(NumSown),
  NumGerm = as.numeric(NumGerm),
  AccessionNumber = as.numeric(AccessionNumber)
)


#Remove negative IDS and NAs
Germ_Cleaning <- filter(Germ_Cleaning, AccessionNumber > 0) %>%
  filter(BrahmsGermTestId > 0)

#Remove NumSown = 0
Germ_Cleaning <- filter(Germ_Cleaning, NumSown > 0)

#Number of each ID (should be 1)
countgerm <- count(Germ_Cleaning, BrahmsGermTestId)

#Added a row showing where there are double IDs
Germ_Cleaning <-
  left_join(Germ_Cleaning, countgerm, by = "BrahmsGermTestId")

#Kept 1 of duplucate IDs where NumSown and NumGerm are the same
Germ_Cleaning <-
  distinct(Germ_Cleaning,
           AccessionNumber,
           BrahmsGermTestId,
           NumSown,
           NumGerm,
           .keep_all = TRUE)

countgerm <- count(Germ_Cleaning, BrahmsGermTestId)

Germ_Cleaning <-
  left_join(Germ_Cleaning, countgerm, by = "BrahmsGermTestId")

#Removed GermIds with different NumGerms, can't be sure which is correct
```

```r
Germ_Cleaning <- filter(Germ_Cleaning, nn == 1)

Germ_Cleaning <- select(Germ_Cleaning,-n,-nn)

# Add in temperature information
#
# - Find "step""
# - Find First temp value
# - Return all temps
#
# Issues:
# - From excel method at this stage, need to work out R method (resolved)
# - This leaves non filtered values with NAs (resolved)
# - These are filtered out at this stage (resolved)
# - steps aren't all included e.g. missing step 1 or 4 (sorts itself out in code)

Germ_Temps <- Germ_Cleaning

#No. of Steps
Germ_Temps <-
  mutate(Germ_Temps, Steps = lengths(regmatches(
    StepSummary, gregexpr("*Step", StepSummary)
  )))

#Remove Rows with 10 or more steps
Germ_Temps <- filter(Germ_Temps, Steps < 10)

#Temp 1, if it doesn't find "1:T" it will return everything,
# so that should be NA
Germ_Temps <- Germ_Temps %>%
  mutate(
    Temp1 = ifelse(
      startsWith(sub(".*?1: T(.*?)(/.*|$)", "\\1", StepSummary), "*"),
      NA,
      sub(".*?1: T(.*?)(/.*|$)", "\\1", StepSummary)
    ),
    Temp2 = ifelse(
      startsWith(sub(".*?2: T(.*?)(/.*|$)", "\\1", StepSummary), "*"),
      NA,
      sub(".*?2: T(.*?)(/.*|$)", "\\1", StepSummary)
    ),
    Temp3 = ifelse(
      startsWith(sub(".*?3: T(.*?)(/.*|$)", "\\1", StepSummary), "*"),
      NA,
      sub(".*?3: T(.*?)(/.*|$)", "\\1", StepSummary)
    ),
    Temp4 = ifelse(
      startsWith(sub(".*?4: T(.*?)(/.*|$)", "\\1", StepSummary), "*"),
      NA,
      sub(".*?4: T(.*?)(/.*|$)", "\\1", StepSummary)
    ),
    Temp5 = ifelse(
      startsWith(sub(".*?5: T(.*?)(/.*|$)", "\\1", StepSummary), "*"),
```

```
      NA,
      sub(".*?5: T(.*?)(/.*|$)", "\\1", StepSummary)
    ),
    Temp6 = ifelse(
      startsWith(sub(".*?6: T(.*?)(/.*|$)", "\\1", StepSummary), "*"),
      NA,
      sub(".*?6: T(.*?)(/.*|$)", "\\1", StepSummary)
    ),
    Temp7 = ifelse(
      startsWith(sub(".*?7: T(.*?)(/.*|$)", "\\1", StepSummary), "*"),
      NA,
      sub(".*?7: T(.*?)(/.*|$)", "\\1", StepSummary)
    ),
    Temp8 = ifelse(
      startsWith(sub(".*?8: T(.*?)(/.*|$)", "\\1", StepSummary), "*"),
      NA,
      sub(".*?8: T(.*?)(/.*|$)", "\\1", StepSummary)
    ),
    Temp9 = ifelse(
      startsWith(sub(".*?9: T(.*?)(/.*|$)", "\\1", StepSummary), "*"),
      NA,
      sub(".*?9: T(.*?)(/.*|$)", "\\1", StepSummary)
    )
  )

#Count Unique temps
GermTempCount <- select(Germ_Temps, contains("Temp"))

GermTempCount$nunique <-
  apply(GermTempCount, 1, function(x)
    length(unique(na.omit(x))))

GermTempCount <- select(GermTempCount, nunique)

Germ_Temps <- bind_cols(Germ_Temps, GermTempCount)

# Now we have the first 9 temperatures and the count of unique temperatures tested
#
# Steps:
#
# - If only 1 temperature tested, that is Test.temp
# - If only 2 temperatures tested
# - THEN If the first temeperture was 60 and above
# - First different temperature after that is Test.temp
# - ELSE THEN If the first temeperture was 0 or less
# - First different temperature after that is Test.temp
# - ELSE "Too many"
# - The rest "Too many"
#
# - Any first temperature 60 and above was considered a heatshock,
# while some below could be it was too hard to determine if they were different to a summer treatment
# - Find the first non heatshock temperature, subtract 1 from nunique
# - If the only temeperature tested is 0, that is correct
```

```
# -If 0s followed by a single temperature, we assume coldshock or null value
# - We should now have a filtered list with a single temperature tested
#
# Issues:
# - heatshock defined as >=60 (in methods)
# - 0 should be ignored or considered coldshock (in methods)
# - How to handle NA  values? (removed)
# - How to check for alternating 0 values (e.g. 0, 5, 0, 5) (Ignore)


Germ_Temps <- transform(Germ_Temps, Temp1 = as.numeric(Temp1))
Germ_Temps <- transform(Germ_Temps, Temp2 = as.numeric(Temp2))
Germ_Temps <- transform(Germ_Temps, Temp3 = as.numeric(Temp3))
Germ_Temps <- transform(Germ_Temps, Temp4 = as.numeric(Temp4))
Germ_Temps <- transform(Germ_Temps, Temp5 = as.numeric(Temp5))
Germ_Temps <- transform(Germ_Temps, Temp6 = as.numeric(Temp6))
Germ_Temps <- transform(Germ_Temps, Temp7 = as.numeric(Temp7))
Germ_Temps <- transform(Germ_Temps, Temp8 = as.numeric(Temp8))
Germ_Temps <- transform(Germ_Temps, Temp9 = as.numeric(Temp9))

Germ_Temps <-
  mutate(Germ_Temps, Heatshock = ifelse(is.na(Temp1), (ifelse(
    Temp2 >= 60, TRUE, FALSE
  )),
  (ifelse(
    Temp1 >= 60, TRUE, FALSE
  ))))


Germ_Temps <- Germ_Temps %>%
  mutate(Test.Temp = ifelse(
    nunique == 1,
    ifelse(is.na(Temp1), Temp2, Temp1),
    # If 1 Temp
    ifelse(
      Heatshock == TRUE,
      #Heatshock
      ifelse(nunique == 2,
             ifelse(
               Temp2 == Temp1,
               ifelse(Temp3 == Temp2,
                      ifelse(
                        Temp4 == Temp3,
                        ifelse(Temp5 == Temp4,
                               ifelse(
                                 Temp6 == Temp5,
                                 ifelse(Temp7 == Temp6,
                                        ifelse(Temp8 == Temp7,
                                               Temp9,
                                               Temp8),
                                        Temp7),
                                 Temp6
                               ),
```

```
                                Temp5),
                        Temp4
                      ),
                      Temp3),
              Temp2
            ),
            "Too many"),
      ifelse(Temp1 <= 0, #Coldshock or error
            ifelse(
              nunique == 2,
              ifelse(Temp2 == Temp1,
                     ifelse(
                       Temp3 == Temp2,
                       ifelse(Temp4 == Temp3,
                              ifelse(
                                Temp5 == Temp4,
                                ifelse(Temp6 == Temp5,
                                       ifelse(
                                         Temp7 == Temp6,
                                         ifelse(Temp8 == Temp7,
                                                Temp9,
                                                Temp8),
                                         Temp7
                                       ),
                                       Temp6),
                                Temp5
                              ),
                              Temp4),
                       Temp3
                     ),
                     Temp2),
              "Too many"
            ),
            "Too many")
    )
  ))

#Remove any rows with "Too many" or NA
Single_Temps <-
  filter(Germ_Temps,!Test.Temp %in% c('Too many', NA))

Single_Temps <-
  transform(Single_Temps, Test.Temp = as.numeric(Test.Temp))

Single_Temps <- filter(Single_Temps, Test.Temp < 100)

Single_Temps <-
  select(Single_Temps,-(
    c(
      Temp1,
      Temp2,
      Temp3,
      Temp4,
```

```
      Temp5,
      Temp6,
      Temp7,
      Temp8,
      Temp9,
      Steps,
      nunique,
      StepSummary
    )
  ))

# Adding in taxon information
#
# Important variables: AccessionNumber, Family, Taxon, Latitude, Longitude,
# Cultivated (not relevant)
#
# Steps:
#
# - Merge taxon information based on Accession number - For blank taxon
# information, *Family_[Accession number]  is used - Filtered out rows with
# cultivated =True (not anymore)

x <-
  distinct(select(Accession_Raw, one_of(
    c(
      "AccessionNumber",
      "Family",
      "Genus",
      "Sp1",
      "Latitude",
      "Longitude",
      "CultivatedFlag"
    )
  )))


# arrange to have lat first
x <- arrange(x, Latitude)

x <-
  distinct(x, AccessionNumber, Family, Genus, Sp1, .keep_all = TRUE)

countx <- count(x, AccessionNumber)

uniquex <- left_join(x, countx, by  = "AccessionNumber")

uniquex <- uniquex %>%
  filter (n == 1) %>%
  transform(AccessionNumber = as.numeric(AccessionNumber))

#This has filtered out all Accession numbers with unclear taxon information
Germ_Taxon <-
  inner_join(Single_Temps, uniquex, by  = "AccessionNumber")
```

```r
Germ_Taxon <- select(Germ_Taxon,-(c(CultivatedFlag, n, Heatshock)))

Germ_Taxon <- unite(Germ_Taxon, Taxon_ID, c("Genus", "Sp1"))

Germ_Taxon <- Germ_Taxon %>%
  mutate(Taxon_ID = ifelse(
    Taxon_ID == "Indet_sp.",
    paste("*", Family, AccessionNumber, sep = ""),
    Taxon_ID
  ))

Germ_Taxon <- select(Germ_Taxon,-(c(Family)))

#Moves columns to the front, use -ColumnID to move to back
Germ_Taxon <- select(Germ_Taxon, Taxon_ID, everything())

# Filtering based on Latitude
#
# Steps:
# - Convert lat/long to numeric
# - Round latitudes using floor(), and add 0.5 to centre
# - Make new ID which is Grid.ID - Species_Genus_Lat_Long

Germ_Lat <- filter(Germ_Taxon,!is.na(Latitude))

Germ_Lat <- filter(Germ_Taxon,!is.na(Longitude))

Germ_Lat <- transform(Germ_Lat, Latitude = as.numeric(Latitude))
Germ_Lat <- transform(Germ_Lat, Longitude = as.numeric(Longitude))

Germ_Lat <- mutate(Germ_Lat, Grid.Lat = floor(Latitude) + 0.5)
Germ_Lat <- mutate(Germ_Lat, Grid.Long = floor(Longitude) + 0.5)

Germ_Lat <-
  mutate(Germ_Lat, Grid.ID = paste(Taxon_ID, Grid.Lat, Grid.Long, sep = "_"))

Germ_Lat <- select(Germ_Lat, Grid.ID, everything())

Germ_Lat <- transform(Germ_Lat, Test.Temp = as.numeric(Test.Temp))

Germ_Filt <- Germ_Lat %>%
  group_by(Grid.ID) %>%
  filter(n_distinct(Test.Temp) > 3)

Germ_Filt <- Germ_Filt %>%
  group_by(Grid.ID) %>%
  filter(max(NumGerm) != 0)

Germ_Test <-
  transform(Germ_Raw, BrahmsGermTestId = as.numeric(BrahmsGermTestId))
Germ_Filt <-
  left_join(Germ_Filt, Germ_Test, by = "BrahmsGermTestId")
```

```r
Germ_Filt <- Germ_Filt %>%
  transform(Full = as.numeric(Full)) %>%
  mutate(
    NumNotGermFull = Full,
    NumSownFull = NumGerm.x + Full,
    NumGerm = NumGerm.x,
    NumSownOrig = NumSown.x,
    AccessionNumber = AccessionNumber.x
  )

MSBP_Collect <-
  read_excel("./Data/2019-09-26_035309887-BRAHMSOnlineData.xlsx") %>%
  transform(
    AccessionNumber = as.integer(AccessionNumber),
    DateCollected = as.Date(DateCollected)
  ) %>%
  select(-Taxon)

MSBP_Test  <-
  read_excel("./Data/2019-09-26_044526851-BRAHMSOnlineData.xlsx") %>%
  separate(DateStarted,
           sep = " ",
           into = c("Date", "Time", "AMPM")) %>%
  select(c(BrahmsGermTestId, Date)) %>%
  filter(!is.na(Date)) %>%
  transform(
    BrahmsGermTestId = as.integer(BrahmsGermTestId),
    GermDate = as.Date(Date, tryFormats = "%m/%d/%Y")
  ) %>%
  select(c(BrahmsGermTestId, GermDate))

Germ_Filt <- Germ_Filt %>%
  left_join(MSBP_Collect, by = "AccessionNumber") %>%
  left_join(MSBP_Test, by = "BrahmsGermTestId") %>%
  mutate(SeedAge = GermDate - DateCollected)

Germ_Filt <-   select(
  Germ_Filt,
  c(
    "Grid.ID"           ,
    "Taxon_ID"   ,
    "AccessionNumber" ,
    "BrahmsGermTestId"  ,
    "Test.Temp"         ,
    "Latitude"   ,
    "Longitude"         ,
    "Grid.Lat"          ,
    "Grid.Long"  ,
    "NumNotGermFull"    ,
    "NumSownFull"       ,
    "NumGerm"    ,
    "NumSownOrig",
    "SeedAge",
```

```
    "GermDate",
    "DateCollected"
  )
)

write.csv(Germ_Filt, file = "./Outputs/Germ_Filtered.csv")

rm(list = ls())
```

MSBPQuadraticModels.R

Takes germination trial data, runs mixed effect logistic regression models for each species*location (Grid.ID), then creates temeprature estimates and their standard errors for: Tmax - Temperature where model predicts 5% overall germination from model (upper), Tmin - Temperature where model predicts 5% overall germination from model (lower), Topt - Temperature where model predicts maximum germination, Topt.upp - Temperature where model predicts 95% of maximum germination from model (upper), Topt.low - Temperature where model predicts 95% of maximum germination from model (lower), Tbreadth - Tmax - Tmin

Standard Errors for each of the above are calculated using the delta method

For Seed Germination over Latitude Project

```
getQuadModels <-  function() {
  #Load packages
  require(tidyr)
  require(ggplot2)
  require(dplyr)
  require(lme4)
  require(broom)
  require(purrr)

  # Steps for pre-model filtering:
  # - Open cleaned data file and make it a table.
  # - Data file must include at least:
  # "Grid.ID", "Test.Temp", "NumSown", "NumGerm", "Grid.Lat"
  # - Add in Proportion germinated
  # - Find maximum germination percentage in group
  # - Filter out any groups with no germination lower than
  #  0.25 of maximum germiantion % (too flat)
  # - Filter out any groups with no germination above 10% (too low)
  # - Filter out any groups with only one value above 0 germination (not enough data)

  MSBP <- read.csv("./Outputs/Germ_Filtered.csv") %>%
    tbl_df() %>%
    rename(Observation = X) %>% #add a term for observation for models below
    mutate(propgerm = NumGerm / NumSownOrig)

  #Find maximum germination percentage in group
  MSBP <- MSBP %>%
    group_by(Grid.ID)  %>%
    mutate(maxgerm = max(propgerm))

  #Filter out any groups with no germination lower than
  # 0.25 of maximum germiantion % (too flat)
  MSBP <- MSBP %>%
    group_by(Grid.ID)  %>%
```

```r
  filter(any(propgerm < 0.25 * maxgerm))

#Filter out any groups with no germination higher than 10% (too low)
MSBP <- MSBP %>%
  group_by(Grid.ID)  %>%
  filter(any(propgerm < 0.1))

#Filter out any groups with only one value above 0 germination (not enough data)
MSBP <- MSBP %>%
  group_by(Grid.ID)  %>%
  add_tally(propgerm > 0) %>%
  filter(n > 1)

# Estimating Tmin, Tmax and Topt (and Topt.upp and Topt.low) using Quadratic Model
#
# Steps:
# - Group by Grid.ID and perform quadratic binomial models on each group
# - This is done 'safely' to avoid errors (no model returned if error occurs)
# - Filter out any concave up fits (can't predict cardinal values)
# - Calulate Topt, Tmin, Tmax
# - Calculate standard error using coefficents and variance covariance matrix

#Method for creating a list of models, or errors if they not not converge

ctrl <-
  glmerControl(optimizer = "bobyqa",
               optCtrl = list(maxfun = 100000)) #control parameters for glmer

#continues if there is an error (returns the error if there is one) package:purrr
safe_glm <- safely(glmer)

MSBP_Models <- MSBP %>%
  group_by(Grid.ID)  %>%
  do(germ.model = safe_glm(
    cbind(NumGerm, NumSownOrig - NumGerm) ~
      I(Test.Temp)  + I((Test.Temp) ^ 2) + (1 | Observation),
    family = binomial,
    control = ctrl,
    nAGQ = 0,
    data = .
  ))


#Extract coefficients a, b, and c from models
MSBP_MaxMinOpt <- MSBP_Models %>%
  rowwise() %>%
  mutate(
    a = ifelse(is.null(germ.model[[1]]), NA, fixef(germ.model[[1]])[3]),
    b = ifelse(is.null(germ.model[[1]]), NA, fixef(germ.model[[1]])[2]),
    c = ifelse(is.null(germ.model[[1]]), NA, fixef(germ.model[[1]])[1])
  )

#Remove concave up fits (a<0) and models without fits (a is NA)
```

```r
MSBP_MaxMinOpt <- filter(MSBP_MaxMinOpt, a < 0)

#Add in Topt, Tmax, Tmin estimates from coefficients
MSBP_MaxMinOpt <- MSBP_MaxMinOpt %>%
  mutate(
    Topt = (-b / (2 * a)),
    Logit.Max = a * Topt * Topt + b * Topt + c,
    #Maximum germination from model at Topt (logit)
    Model.Max = (exp(Logit.Max) / (1 + exp(Logit.Max))),
    #convert to actual maximum germiantion
    Topt.upp = (-b - (sqrt((
      b * b - 4 * a * (c - log(0.95 * Model.Max / (1 - 0.95 * Model.Max)))
    )))) / (2 * a),
    Topt.low = (-b + (sqrt((
      b * b - 4 * a * (c - log(0.95 * Model.Max / (1 - 0.95 * Model.Max)))
    )))) / (2 * a),
    Tmax = (-b - (sqrt((
      b * b - 4 * a * (c - log(0.05 / (1 - 0.05)))
    )))) / (2 * a),
    Tmin = (-b + (sqrt((
      b * b - 4 * a * (c - log(0.05 / (1 - 0.05)))
    )))) / (2 * a),
    Tbreadth = Tmax - Tmin,
    Toptbreadth = Topt.upp - Topt.low
  )

# Delta method  - standard error estimate
se.matrix.fun <-
  function(a,
           b,
           c,
           Model.Max,
           prop,
           upp.low,
           germ.model,
           is.opt) {
    if (is.null(germ.model[[1]])) {NA}
    else {
      x <-
        ifelse(is.opt == T,
               log(prop * Model.Max / (1 - prop * Model.Max)), #If calulating Topt, use model maximum
               log(prop / (1 - prop))) #If calculating Min/Max, use total proportion

      delta <- b * b - 4 * c * a + 4 * x * a

      #Original error calculations
  pa <- ifelse(upp.low == "upp",
    1 / (2 * a ^ 2) * (b + (delta ^ (1 / 2)) + 2 * a * (c - x) * (delta ^ (-1 / 2))), #Topt.upp/Tmax
    1 / (2 * a ^ 2) * (b - (delta ^ (1 / 2)) - 2 * a * (c - x) * (delta ^ (-1 / 2)))) #Topt.low/Tmin

  pb <- ifelse(upp.low == "upp",
               -1 / (2 * a) * (1 + b * (delta ^ (-1 / 2))), #Topt.upp/Tmax
               -1 / (2 * a) * (1 - b * (delta ^ (-1 / 2))))#Topt.low/Tmin
```

```
  pc <- ifelse(upp.low == "upp", (delta ^ (-1 / 2)), #Topt.upp/Tmax
               -(delta ^ (-1 / 2)))  #Topt.low/Tmin

  vector.v <- as.vector(c(pc, pb, pa))
  vector.h <- t(vector.v) #transposed vector
  vcov.matrix <- as.matrix(vcov(germ.model[[1]]))
  variance <- vector.h %*% vcov.matrix %*% vector.v
  variance <- ifelse(variance < 0, NA, sqrt(as.numeric(variance)))
    }
  }

se.opt.fun <- function(a, b, germ.model) {
  if (is.null(germ.model[[1]])) {
    NA
  }
  else {
    pa <- b / (2 * a * a)
    pb <- -1 / (2 * a)
    pc <- 0

    vector.v <- as.vector(c(pc, pb, pa))
    vector.h <- t(vector.v) #transposed vector
    vcov.matrix <- as.matrix(vcov(germ.model[[1]]))
    vcov.matrix[1, ] <- 0
    vcov.matrix[, 1] <- 0
    variance <- vector.h %*% vcov.matrix %*% vector.v
    variance <- ifelse(variance < 0, NA, sqrt(as.numeric(variance)))
  }
}

se.breadth.fun <- function(a, b, c, Model.Max, prop, germ.model, is.opt) {
  if (is.null(germ.model[[1]])) {
    NA
  }
  else {
    x <-
      ifelse(is.opt == T,
             log(prop * Model.Max / (1 - prop * Model.Max)), #If calulating Topt breadth, use model m
             log(prop / (1 - prop))) #If calculating breadth, use total proportion

    delta <- b * b - 4 * c * a + 4 * x * a

    ## Corrected formula
    pa <- (-1 / (a * a)) * (sqrt(delta) + (2 * a * (c - x) / (sqrt(delta))))
    pb <- b / (a * sqrt(delta))
    pc <- -2 / sqrt(delta)

    vector.v <- as.vector(c(pc, pb, pa))
    vector.h <- t(vector.v) #transposed vector
    vcov.matrix <- as.matrix(vcov(germ.model[[1]]))
    variance <- vector.h %*% vcov.matrix %*% vector.v
    variance <- ifelse(variance < 0, NA, sqrt(as.numeric(variance)))
  }
```

```
  }

  MSBP_MaxMinOpt <- MSBP_MaxMinOpt %>%
    rowwise() %>%
    mutate(
      SEmax = se.matrix.fun(a, b, c, Model.Max, 0.05, "upp", germ.model, F),
      SEmin = se.matrix.fun(a, b, c, Model.Max, 0.05, "low", germ.model, F),
      SEopt.upp = se.matrix.fun(a, b, c, Model.Max, 0.95, "upp", germ.model, T),
      SEopt.low = se.matrix.fun(a, b, c, Model.Max, 0.95, "low", germ.model, T),
      SEopt = se.opt.fun(a, b, germ.model),
      SEbreadth = se.breadth.fun(a, b, c, Model.Max, 0.05, germ.model, F),
      SEoptbreadth = se.breadth.fun(a, b, c, Model.Max, 0.95, germ.model, T)
    )

  #Remove models
  MSBP_MaxMinOpt <- dplyr::select(MSBP_MaxMinOpt,-germ.model)

  MSBP_MaxMinOpt <- left_join(MSBP, MSBP_MaxMinOpt, "Grid.ID")

  #If there is no germination proportion above Topt that is less than 25% maxgerm,
  #ignore Tmax, Topt.upp, Topt.low and Topt
  #If there is no germination proportion below Topt that is less than 25% maxgerm,
  #ignore Tmin, Topt.upp, Topt.low and Topt
  MSBP_MaxMinOpt <- MSBP_MaxMinOpt %>%
    group_by(Grid.ID)  %>%
    mutate(
      Tmax = ifelse(any((Test.Temp > Topt & propgerm < 0.25 * maxgerm)), Tmax, NA),
      Tmin = ifelse(any((Test.Temp < Topt & propgerm < 0.25 * maxgerm)), Tmin, NA),
      Topt = ifelse((any((Test.Temp > Topt & propgerm < 0.25 * maxgerm)) &
        any((Test.Temp < Topt & propgerm < 0.25 * maxgerm))), Topt, NA),
      Topt.upp = ifelse((any((Test.Temp > Topt & propgerm < 0.25 * maxgerm)) &
        any((Test.Temp < Topt & propgerm < 0.25 * maxgerm))), Topt.upp, NA),
      Topt.low = ifelse((any((Test.Temp > Topt &  propgerm < 0.25 * maxgerm)) &
        any((Test.Temp < Topt & propgerm < 0.25 * maxgerm))), Topt.low, NA),
      Tbreadth = ifelse((any((Test.Temp > Topt & propgerm < 0.25 * maxgerm)) &
        any((Test.Temp < Topt & propgerm < 0.25 * maxgerm))), Tbreadth, NA),
      Toptbreadth = ifelse((any((Test.Temp > Topt & propgerm < 0.25 * maxgerm)) &
        any((Test.Temp < Topt & propgerm < 0.25 * maxgerm))), Tbreadth, NA)
    )


  write.csv(MSBP_MaxMinOpt, file = "./Outputs/MSBP_MaxMinOpt.csv")
}

getQuadModels()
```

MSBPClimateData.R

```
getClimateData <- function() {
  # Load Packages
  require(dplyr)
  require(tidyr)
  require(raster)
  require(sp)
```

```
require("rgdal")

MSBP <- tbl_df(read.csv(file = "./Outputs/MSBP_MaxMinOpt.csv"))

# Load Current Climate Data
Current <- getData("worldclim",
                   var = "bio",
                   res = 10,
                   path = "./")
Current <- Current[[c(1, 8)]]
names(Current) <- c("Current.Temp", "Current.Temp.Hot.Quart")

# weighted means of each polygon, removes NAs first, takes a long time
coords <-
  dplyr::select(MSBP, one_of(c("Longitude", "Latitude"))) %>%
  rename(x = Longitude, y = Latitude)
points <- SpatialPoints(coords, proj4string = Current@crs)
values <- raster::extract(Current, points)
Currentdf <- cbind.data.frame(coordinates(points), values)

## Future (2070) Average Temperature, rcp =85, model = AC
Future85 <-  getData('CMIP5',
    var = 'bio',
    res = 10,
    rcp = 85,
    model = 'AC',
    year = 70,
    path = "./")
Future85 <- Future85[[c(1, 8)]]
names(Future85) <- c("Future.Temp", "Future.Temp.Hot.Quart")

coords <- dplyr::select(MSBP, one_of(c("Longitude", "Latitude"))) %>%
  rename(x = Longitude, y = Latitude)
points <- SpatialPoints(coords, proj4string = Future85@crs)
values <- raster::extract(Future85, points)
Future85df <- cbind.data.frame(coordinates(points), values)

## Altitude from worldclim dataset
Altitude.dl <- getData('worldclim', var = 'alt', res = 10)
Altitude.dl <- Altitude.dl[[1]]
names(Altitude.dl) <- "Altitude"

coords <- dplyr::select(MSBP, one_of(c("Longitude", "Latitude"))) %>%
  rename(x = Longitude, y = Latitude)
points <- SpatialPoints(coords, proj4string = Altitude.dl@crs)
values <- raster::extract(Altitude.dl, points)
Altitudedf <- cbind.data.frame(coordinates(points), values)

#Making one table

MSBP.Climate <- cbind.data.frame(MSBP,
                                 Currentdf[3:4],
                                 Future85df[3:4],
```

```
                                    Altitudedf[3])

  detach("package:raster", unload = TRUE)

  mean.na <- function (...)  {mean(..., na.rm = T)}

  MSBP.Climate <- MSBP.Climate %>%
    mutate(Altitude = ifelse(values < 0, 0, values)) %>%
    select(-values) %>%
    group_by(Grid.ID) %>%
    mutate(
      altitude = mean.na(Altitude),
      SeedAge = mean.na(SeedAge),
      Current.Temp = mean.na(Current.Temp) / 10,
      Future.Temp = mean.na(Future.Temp) / 10,
      Current.Hot.Quart = mean.na(Current.Temp.Hot.Quart) /  10,
      Future.Hot.Quart = mean.na(Future.Temp.Hot.Quart) / 10
    ) %>%
    mutate(
      CM.Current = Current.Hot.Quart - Topt.upp,
      WR.Future = Future.Hot.Quart - Tmax,
      CM.Future = Future.Hot.Quart - Topt.upp,
      WR.Current = Current.Hot.Quart - Tmax,
      TB = Tmax - Tmin,
      AbsLat = abs(Grid.Lat),
      Hemisphere = ifelse(Grid.Lat >= 0, "N", "S"),
      Warming = Future.Temp - Current.Temp
    ) %>%
    as_tibble() %>%
    distinct(Grid.ID,  .keep_all = T) %>%
  select(-c("X")) %>%
  unite(grid.ll, c(Grid.Lat, Grid.Long), remove =F)%>%
  mutate(NorthTF = ifelse(Hemisphere == "N", T, F)) %>%
  filter(!is.na(Tmax)|!is.na(Tmin))

write.csv(MSBP.Climate, file = "./Data/MSBP_Clean", row.names = FALSE)

}

getClimateData()
```

metaforR2.R

Calculating approximate R2 from metafor objects

```
#We are using marginal  R2 here (fix/fix+random)
R2 <- function(model){

  # fixed effect variance
  fix <- var(as.numeric(as.vector(model$b) %*% t(as.matrix(model$X))))

  # marginal
  R2m <- fix / (fix + sum(model$sigma2))

  # conditional
```

```
  R2c <- (fix + sum(model$sigma2) - model$sigma2[length(model$sigma2)]) /
    (fix + sum(model$sigma2))

  R2s <- c(R2_marginal = R2m, R2_conditional = R2c)

  return(R2s)
}
```

getPhyloCor.R

```
getPhyloCor <- function(data) {

  require(ape)
  require(dplyr)
  require(tidyr)

  MSBP.tree <- data
  phyall <- read.tree(file = "./Data/Phylo/ALLMB.tre") #full phylo tree for plants

  rownames(MSBP.tree) <- MSBP.tree$Taxon_ID

  MSBP.Species <- as.vector(MSBP.tree$Taxon_ID)

  #Prune full tree for matches
  PrunedMSBP <- drop.tip(phyall, phyall$tip.label[na.omit(-match(MSBP.Species, phyall$tip.label))])

  MSBP.tree <- rename(MSBP.tree, "PrunedMSBP$tip.label" = Taxon_ID) %>%
               right_join(as.data.frame(PrunedMSBP$tip.label), by = "PrunedMSBP$tip.label") %>%
               rename(Taxon_ID="PrunedMSBP$tip.label")

  MSBP.Species <- as.vector(MSBP.tree$Taxon_ID)

  nSpp <- nrow(MSBP.Species)

  MSBP.tree[, "phylo"] <- MSBP.Species

  tree <- compute.brlen(PrunedMSBP)

  Vphy <- vcv(tree, cor = T) #correlation matrix

  return(list(MSBP.tree = MSBP.tree, cor = Vphy))

}
```

germToleranceBreadth.R

Germination Temperature Breadth Over Latitude analyses and outputs For Seed Germination over Latitude Project

fromfile = TRUE - Use previous analysis file fromfile = FALSE - Rerun analysis

```
getToleranceBreadth <- function(fromfile = T) {
  if (fromfile == T) {
    #switch to not rerun analysis
    output <- readRDS("./Outputs/TB")
  } else {
```

```r
#Load packages
require(dplyr)
require(tidyr)
require(metafor)
require(ggplot2)
source("./R/getPhyloCor.R")

#Load data
MSBP <- read.csv(file = "./Data/MSBP_Clean")

#Prepare data set
MSBP.meta.TB <- MSBP %>%
  mutate(vi = SEbreadth * SEbreadth) %>% #Variances
  filter(!is.na(Tbreadth) &
           !is.na(SeedAge) &
           !is.na(Altitude)) #Remove NA values

#Remove duplicate species, for phylogeny analysis
MSBP.meta.TB.phy <-
  distinct(MSBP.meta.TB, Taxon_ID, .keep_all = T)

#Get phylogeny: tree - .[[1]] and correlation matrix - .[[2]]
MSBP.tree.cor <- getPhyloCor(MSBP.meta.TB.phy)

### Models below:
# breadth - Tbreadth ~ AbsLat
# .h - with fixed effect for hemipshere
# .phy - with random effect for phylogeny, using variance covariance matrix
#
# All have species and site as random factors
# AND log10(SeedAge) and altitude as fixed effects

breadth.h.phy <-  rma.mv(
  yi = TB,
  V = vi,
  mod = ~ AbsLat + AbsLat:NorthTF + log10(SeedAge) + altitude,
  random = list(~ 1 | phylo,
                ~ 1 | Taxon_ID,
                ~ 1 | grid.ll),
  R = list(phylo = MSBP.tree.cor$cor),
  method = "REML",
  data = MSBP.tree.cor$MSBP.tree,
  verbose = F,
  digits = 5,
  control = list(optimizer = "optim", optmethod = "Nelder-Mead")
)
breadth.h.phy #no hemisphere effect, therefore remove term

breadth.h <- rma.mv(
  yi = TB,
  V = vi,
  mod = ~ AbsLat + AbsLat:NorthTF + log10(SeedAge) + altitude,
  random = list( ~ 1 | Taxon_ID,
```

```
                  ~ 1 | grid.ll),
  method = "REML",
  data = MSBP.meta.TB,
  control = list(optimizer = "optim", optmethod = "Nelder-Mead")
)
breadth.h #no hemisphere effect, therefore remove term


breadth.phy <-  rma.mv(
  yi = TB,
  V = vi,
  mod = ~ AbsLat + log10(SeedAge) + altitude,
  random = list(~ 1 | phylo,
                ~ 1 | Taxon_ID,
                ~ 1 | grid.ll),
  R = list(phylo = MSBP.tree.cor$cor),
  method = "REML",
  data = MSBP.tree.cor$MSBP.tree,
  verbose = F,
  digits = 5,
  control = list(optimizer = "optim", optmethod = "Nelder-Mead")
)
breadth.phy


breadth <- rma.mv(
  yi = TB,
  V = vi,
  mod = ~ AbsLat + log10(SeedAge)  + altitude,
  random = list( ~ 1 | Taxon_ID,
                 ~ 1 | grid.ll),
  method = "REML",
  data = MSBP.meta.TB,
  control = list(optimizer = "optim", optmethod = "Nelder-Mead")
)
breadth


#Plot of ranges against absolute latitude with lat effect
TB_plot <- ggplot(data = MSBP.meta.TB, aes(x = AbsLat)) +
  geom_rect(aes(
    xmin = 0,
    xmax = 23.5,
    ymax = Inf,
    ymin = -Inf
  ), fill = "grey90") +
  geom_point(aes(y = TB, alpha = sqrt(1 / SEbreadth)), colour = "green4") +
  geom_rug(
    aes(y = TB),
    alpha = 1 / 2,
    position = "jitter",
    sides = "b"
  ) +
```

```
      theme_classic() +
      ylim(0, 50) +
      ylab(
        expression(
"Maximum Germination Temperature - \n Minimum Germination Temperature  "(degree * C)
        )
      ) +
      xlab(expression("Absolute Latitude "(degree))) +
      theme(legend.position = "none",
            plot.margin = margin(10, 10, 10, 30)) +
      scale_alpha_continuous(range = c(0.5, 1)) +
      scale_x_continuous(limits = c(0, 60), expand = c(0, 0))

    TB_plot

    #Values to get confidence intervals of model fit
    mods <- cbind(
      AbsLat = c(1:57),
      #abslat - 1 to 57
      SeedAge = rep(log10(1437), 57),
      #median of SeedAge, 57 times
      altitude = rep(435, 57)
    ) #median of altitude, 57 times

    ### calculate predicted values from model
    preds1 <- predict.rma(breadth, newmods = mods, addx = T)

    ci.plot1 <- as.data.frame(cbind(c(seq(1, 57)), #1 to 57
                                    preds1[[1]],  #Model fit
                                    preds1[[3]],  #Confidence interval of model
                                    preds1[[4]])) #Confidence interval of model

    #Plot original figure with confidence intervals from above
    TB_plot <-  TB_plot +
      geom_line(data = filter(ci.plot1, V1 > -54 & V1 > 0),
                aes(x = V1, y = V3),
                linetype = "dashed") +
      geom_line(data = filter(ci.plot1, V1 > -54 & V1 > 0),
                aes(x = V1, y = V4),
                linetype = "dashed") +
      xlab("Latitude (°)")

    TB_plot

    ggsave(
      file = "./Outputs/LinesAbsLatRangeSE.tiff",
      units = "in",
      width = 5,
      height = 4,
      dpi = 1000,
      compression = 'lzw'
    )
```

```
    output <- list(
      Model = breadth,
      Model.phy = breadth.phy,
      Plot = TB_plot,
      Model.h = breadth.h,
      Model.h.phy = breadth.h.phy
    )

    saveRDS(output, "./Outputs/TB")
  }

  return(output)

}
```

germClimateMismatchCurrent.R

This is code to run analayses and create outputs for Current Climate Mismatch

```
getCM <- function(fromfile = T) {

  if (fromfile == T) {
    output <- readRDS("./Outputs/CM_current")
  } else {


require(dplyr)
require(tidyr)
require(metafor)
require(ggplot2)
source("./R/getPhyloCor.R")

MSBP <- read.csv(file = "./Data/MSBP_Clean") #core dataset

#Prepare data set
MSBP.CM <- MSBP %>%
  mutate(vi = SEopt.upp * SEopt.upp) %>% #Standard error for Topt.upp
  filter(!is.na(Topt.upp) &
         !is.na(Current.Hot.Quart) &
         !is.na(SeedAge) &
         !is.na(Altitude)) #remove NAs

MSBP.CM.phy <- distinct(MSBP.CM, Taxon_ID, .keep_all = T) #Needed for phylogeny analysis

#Get phylogeny tree[[1]] and correlation matrix[[2]]
MSBP.CM.phy <- getPhyloCor(MSBP.CM.phy)

## Current Climate Mismatch ~ Latitude and Hemispshere (without phylogeny)
# additional fixed effects of log10(SeedAge) and altitude
# using random effect of site (grid.ll) and species (Taxon_ID)
CM.h <- rma.mv(
  yi = Current.Hot.Quart - Topt.upp,
  V = vi,
  mod = ~ AbsLat + (AbsLat:NorthTF) + log10(SeedAge) + altitude,
```

```
  random = list(~ 1 | Taxon_ID,
                ~ 1 | grid.ll),
  method = "REML",
  data = MSBP.CM
)
CM.h #no effect of hemisphere, so repeat without

## Current Climate Mismatch ~ Latitude and Hemispshere
# additional fixed effects of log10(SeedAge) and altitude
# using random effect of site (grid.ll) and species (Taxon_ID)
# and phylogeny with corresponding correlation matrix
CM.h.phy <- rma.mv(
  yi = Current.Hot.Quart - Topt.upp,
  V = vi,
  mod = ~ AbsLat + (AbsLat:NorthTF) + log10(SeedAge) + altitude,
  random = list( ~ 1 | phylo,
                 ~ 1 | Taxon_ID,
                 ~ 1 | grid.ll),
  R = list(phylo = MSBP.CM.phy$cor),
  method = "REML",
  data = (MSBP.CM.phy[[1]]),
  verbose = FALSE,
  digits = 5,
  control = list(optimizer = "optim", optmethod = "Nelder-Mead")
)
CM.h.phy #no effect of hemisphere, so repeat without

## Current Climate Mismatch ~ Latitude
# additional fixed effects of log10(SeedAge) and altitude
# using random effect of site (grid.ll) and species (Taxon_ID)
# and phylogeny with corresponding correlation matrix
CM.phy <- rma.mv(
  yi = Current.Hot.Quart - Topt.upp,
  V = vi,
  mod = ~ AbsLat + log10(SeedAge) + altitude,
  random = list( ~ 1 | phylo,
                 ~ 1 | Taxon_ID,
                 ~ 1 | grid.ll),
  R = list(phylo = MSBP.CM.phy$cor),
  method = "REML",
  data = MSBP.CM.phy[[1]]
)
CM.phy #no interactions with abslat

## Current Climate Mismatch ~ Latitude and Hemispshere
# additional fixed effects of log10(SeedAge) and altitude
# using random effect of site (grid.ll) and species (Taxon_ID)
CM <- rma.mv(
  yi = Current.Hot.Quart - Topt.upp,
  V = vi,
  mod = ~ AbsLat + log10(SeedAge) + altitude,
  random = list( ~ 1 | Taxon_ID,
                 ~ 1 | grid.ll),
```

```
  method = "REML",
  data = MSBP.CM
)


CM.low <- rma.mv(
  yi = Current.Hot.Quart - Topt.low,
  V = vi,
  mod = ~ AbsLat + log10(SeedAge) + altitude,
  random = list( ~ 1 | Taxon_ID,
                 ~ 1 | grid.ll),
  method = "REML",
  data = MSBP.CM
)

CM #no interactions with abslat

CM.plot <- ggplot(data = MSBP.CM , aes(x = AbsLat)) +
  geom_rect(aes(
    xmin = 0, xmax = 23.5,
    ymax = Inf, ymin = -Inf
  ), fill = "grey90") +
    geom_hline(aes(yintercept = 0), size = 0.5) +
  geom_point(aes(y = Current.Hot.Quart - Topt.upp,
                 alpha = sqrt(1 / vi)), colour = "orange") +
  geom_rug(aes(y = Current.Hot.Quart - Topt.upp),
    alpha = 1 / 2,
    position = "jitter",
    sides = "b") +
  theme_classic() +
  ylab(expression(
    "Current Environmental Temperature - \n Upper Optimal Germination Temperature  "
    (degree * C))) +
  xlab(expression("Absolute Latitude "(degree))) +
  theme(legend.position = "none",
        plot.margin = margin(10, 10, 10, 30)) +
    scale_alpha_continuous(range = c(0.4,1))+
  scale_x_continuous(limits = c(0, 60), expand = c(0, 0))


 CM.plot


#Values to get confidence intervals
mods <- cbind(AbsLat = c(1:57),
              SeedAge= rep(log10(1479), 57),
              altitude= rep(393, 57)) #abslat - 1 to 57

### calculate predicted values from model
preds1 <- predict(CM, newmods=mods, addx=T)

ci.plot1 <- as.data.frame(
  cbind(c(seq(1, 57)),
```

```
        preds1[[1]],
        preds1[[3]],
        preds1[[4]])) %>%
  filter(V1 > 0)

#Plot original figure with confidence intervals from above
CM.plot <-  CM.plot +
  geom_line(data = ci.plot1,
            aes(x = V1, y = (V2)),
            colour = "orange",
            linetype = "solid") +
  geom_line(data = ci.plot1,
            aes(x = V1, y = (V3)),
            linetype = "dashed") +
  geom_line(data = ci.plot1,
            aes(x = V1, y = (V4)),
            linetype = "dashed") +
  xlab("Latitude (°)")

CM.plot

ggsave(
  CM.plot,
  filename = "./Outputs/CVSE.tiff",
  units = "in",
  width = 4,
  height = 5,
  dpi = 1000,
  compression = 'lzw'
)


output <- list(
  Model = CM,
  Model.phy = CM.phy,
  Plot = CM.plot,
  Model.h = CM.h,
  Model.h.phy = CM.h.phy,
  Model.low = CM.low
)

saveRDS(output, "./Outputs/CM_current")
  }

  return(output)

}
```

germWarmingRiskFuture.R

Future Warming Risk analayses and outputs For Seed Germination over Latitude Project

```
getWR <- function(fromfile = T) {

  if (fromfile == T) {
```

```
  output <- readRDS("./Outputs/WR_future")
} else {


require(dplyr)
require(tidyr)
require(metafor)
require(ggplot2)
source("./R/getPhyloCor.R")

MSBP <- read.csv(file = "./Data/MSBP_Clean") #core dataset

### Prepare data set, if SE's are too high, the analyses don't run
# (but get weighted out of the model anyway)
MSBP.WR.data <- MSBP %>%
  mutate(vi = SEmax * SEmax) %>% #Standard error for warming risk
  filter(!is.na(Tmax) &
         !is.na(Future.Hot.Quart) &
         !is.na(SeedAge) &
         !is.na(Altitude)) #remove NAs

MSBP.WR.data.phy <- distinct(MSBP.WR.data, Taxon_ID, .keep_all = T) #Needed for phylogeny analysis

# Get phylogeny tree[[1]] and correlation matrix[[2]]
MSBP.tree.cor <- getPhyloCor(MSBP.WR.data.phy)

## Future Warming Risk ~ Latitude and Hemispshere (with phylogeny)
#WR, using random effect of phylo, site (grid.ll) and species (Taxon_ID), and altitude
WR.h.phy <- rma.mv(
  yi = Future.Hot.Quart - Tmax,
  V = vi,
  mod = ~ AbsLat + (AbsLat:NorthTF) + log10(SeedAge) + altitude,
  random = list( ~ 1 | phylo, ~ 1 | Taxon_ID,
                 ~ 1 | grid.ll),
  R = list(phylo = MSBP.tree.cor$cor),
  method = "REML",
  data = MSBP.tree.cor$MSBP.tree,
  verbose = F,
  digits = 5,
  control = list(optimizer = "optim", optmethod = "Nelder-Mead")
)

summary(WR.h.phy) #Hemisphere effect (p = 0.02), no interaction effects

## Future Warming Risk ~ Latitude and Hemispshere (with phylogeny)
#WR, using random effect of phylo, site (grid.ll) and species (Taxon_ID), and altitude
WR.h <- rma.mv(
  yi = Future.Hot.Quart - Tmax,
  V = vi,
  mod = ~ AbsLat + (AbsLat:NorthTF) + log10(SeedAge) + altitude,
  random = list(~ 1 | Taxon_ID, ~ 1 | grid.ll),
  method = "REML",
  data = MSBP.WR.data
```

```
  )

  summary(WR.h) #Hemisphere effect (p = 0.02)


#Plot of future WR with lat and hemisphere effect, no interaction effects

WR.plot.f <- ggplot(data = MSBP.WR.data , aes(x = Latitude)) +
    geom_rect(aes(
      xmin = -23.5,
      xmax = 23.5,
      ymax = Inf,
      ymin = -Inf ), fill = "grey90") +
    geom_rug(aes(y = Future.Hot.Quart - Tmax), alpha = 1/2, position = "jitter", sides="b") +
    geom_point(aes(y = Future.Hot.Quart - Tmax,
                   alpha = sqrt(1/SEmax)), colour = "Red") +
    geom_hline(aes(yintercept = 0), size = 0.5) +
    theme_classic() +
    ylab(expression(
      "Future Environmental Temperature - \n Maximum Germination Temperature   "(degree * C)
    )) +
    xlab(expression("Absolute Latitude "(degree))) +
  scale_alpha_continuous(range = c(0.4,1))+
    theme(legend.position = "none", plot.margin = margin(10,10,10,30))+
     scale_x_continuous(limits = c(-60,64), expand = c(0,0))

  WR.plot.f

  #Values to get confidence intervals
  mods <- cbind(AbsLat = c(0:70, 0:70),
                SeedAge= rep(log10(1479), 142),
                altitude= rep(393, 142),
                NorthTF= c(rep(T, 71), rep(F, 71)))

  ### calculate predicted values from model
  preds1 <- predict(WR.h, newmods=mods, addx=T)

  ci.plot1 <- as.data.frame(
    cbind(c(0:70, 0:(-70)),
          preds1[[1]],
          preds1[[3]],
          preds1[[4]]))

  #Plot original figure with confidence intervals from above
  WR.plot.f <-  WR.plot.f +
    geom_line(data = filter(ci.plot1, V1 > -53 & V1 < 1),
              aes(x = V1, y = (V2)),
              colour = "red",
              linetype = "solid") + #South
    geom_line(data = filter(ci.plot1, V1 > -53 & V1 < 1),
              aes(x = V1, y = (V3)),
              linetype = "dashed") + #South
    geom_line(data = filter(ci.plot1, V1 > -53 & V1 < 1),
```

```
              aes(x = V1, y = (V4)),
              linetype = "dashed") + #South
    geom_line(data = filter(ci.plot1, V1 > -1 & V1 < 64),
              aes(x = V1, y = (V2)),
              colour = "red",
              linetype = "solid") + #North
    geom_line(data = filter(ci.plot1, V1 > -1 & V1 < 64),
              aes(x = V1, y = (V3)),
              linetype = "dashed") + #North
    geom_line(data = filter(ci.plot1, V1 > -1 & V1 < 64),
              aes(x = V1, y = (V4)),
              linetype = "dashed") + #North
    xlab("Latitude (°)")

  WR.plot.f

ggsave(
  WR.plot.f,
  filename = "./Outputs/WRSE.tiff",
  units = "in",
  width = 4,
  height = 5,
  dpi = 1000,
  compression = 'lzw')


#Return results
output <- list(
    Model.h = WR.h,
    Model.h.phy = WR.h.phy,
    Plot = WR.plot.f
  )

saveRDS(output, "./Outputs/WR_future")
  }

  return(output)

}
```

germLifeWoodWR.R

This is code to run analayses and create outputs for the effect of Longevity and growth form on future warming risk

```
getLifeWoodWR <- function() {

  require(dplyr)
  require(tidyr)
  require(metafor)
  require(ggplot2)
  source("./R/getPhyloCor.R")


longevity <- read.csv(file = "./Data/Longevity/TRY.longevity.csv") #longevity dataset
```

```
MSBP <- read.csv(file = "./Data/MSBP_Clean") #core dataset

#Prepare data set, if SE's are too high, the analyses don't run (but get weighted out anyway)
MSBP <- MSBP %>%
        mutate(vi = SEmax*SEmax) %>% #Standard error for warming risk
        filter(!is.na(Tmax)          &
               !is.na(Future.Hot.Quart) &
               !is.na(SeedAge) &
               !is.na(Altitude)) #Remove NAs

#Take longevity data and seperate into "annual" and "perennial"
longevityfilt <- longevity %>%
  mutate(
    ann_per = case_when(
      OrigValueStr == "annual" |
        OrigValueStr == "Annual" |
        OrigValueStr == "annuals" |
        OrigValueStr == "summer annuals" |
        OrigValueStr == "always annual" |
        OrigValueStr == "winter annuals" |
        OrigValueStr == "annual-winter annual" |
        OrigValueStr == "winter annual" |
        (OriglName == "Life history" &
           OrigValueStr == "1") |
        (OriglName == "Plant phenology: Annual" &
           OrigValueStr == "yes")     ~   "annual", #annuals
      OrigValueStr == "perennial" |
        OrigValueStr == "Perennial" |
        OrigValueStr == "perennials" |
        OrigValueStr == "always pluriennial-pollakanthic" |
        (OriglName == "Plant phenology: Biennial" &
           OrigValueStr == "yes") |
        OrigValueStr == "perennial < 20 years" |
        OrigValueStr == "woody" |
        OrigValueStr == "perennial/woody" |
        OrigValueStr == "perennial > 20 years" |
        OrigValueStr == "poly-annuals > 50 years (long-lived perennials)" |
        OrigValueStr == "always biennial, always pluriennial-hapaxanthic" |
        OrigValueStr == "always biennial, always pluriennial-pollakanthic" |
        OrigValueStr == "tree" |
        OrigValueStr == "shrub" |
        OrigValueStr == "always pluriennial-hapaxanthic, always pluriennial-pollakanthic" |
        OrigValueStr == "always pluriennial-hapaxanthic" |
        OrigValueStr == "biennial" |
        OrigValueStr == "annual/biennial" |
        OrigValueStr == "poly-annuals < 5 years (short-lived perennials)" |
        OrigValueStr == "Biennial" |
        OrigValueStr == "biennial/perennial" |
        OrigValueStr == "always biennial" |
        OrigValueStr == "biennial-perennial" |
        OrigValueStr == "sometimes biennial, always pluriennial-hapaxanthic, sometimes pluriennial-polla
        OrigValueStr == "sometimes biennial, sometimes pluriennial-hapaxanthic, always pluriennial-polla
        OrigValueStr == "biennial/perennial/woody" |
```

```
        OrigValueStr == "sometimes biennial, always pluriennial-pollakanthic" |
        OrigValueStr == "poly-annuals 5-50 years (medium-lived perennials)" |
        (OriglName == "Plant phenology: Perennial" &
           OrigValueStr == "yes") |
        (OriglName == "Plant phenology: Annual" &
           OrigValueStr == "no")    ~   "perennial"  #perennials
    )
  )

longevityfilt <- data.frame(lapply(longevityfilt, function(x) {gsub(" ", "_", x)})) %>%
                 distinct(SpeciesName, .keep_all = T) %>%
                 select(SpeciesName, ann_per)

#Add Longevity to MSBP core dataset
MSBPlong <- left_join(MSBP, longevityfilt, by = c("Taxon_ID" ="SpeciesName"))

##Get woodiness information
woody <- read.csv(file = "./Data/GlobalWoodinessDatabase.csv")

woody <- data.frame(lapply(woody, function(x) {gsub(" ", "_", x)}))%>%
          distinct(gs, .keep_all = T) %>%
         select(gs, woodiness)

MSBPlong <- left_join(MSBPlong, woody, by = c("Taxon_ID" ="gs"))

#Assign any taxa with "w" (woody) to be perennial
MSBPlong <- MSBPlong %>%
            mutate(ann_per=replace(ann_per, woodiness=="W", "perennial"))

#Filter dataset for analysis (without phylogeny)
MSBP.meta.wood <- mutate(MSBPlong, vi = SEmax*SEmax) %>%
  filter(woodiness == "W"|woodiness == "H")

#Filter dataset for analysis (with phylogeny)
MSBP.meta.wood.phy <- mutate(MSBPlong, vi = SEmax*SEmax) %>%
                      filter(woodiness == "W"|woodiness == "H")  %>%
                      distinct(Taxon_ID, .keep_all = T) # needed for phlogentic analysis

#Get Phylogentic Matrix
MSBP.tree.cor.wood <- getPhyloCor(MSBP.meta.wood.phy)


## Future warming risk ~ Woodiness only (with phylogeny)
wood.mv.WT.phy <- rma.mv(yi = Future.Hot.Quart - Tmax,
                         V = vi,
                         mod = ~woodiness + log10(SeedAge) + altitude,
                         random = list(~1 | Taxon_ID, ~1 | grid.ll, ~1 | phylo),
                         R = list(phylo = MSBP.tree.cor.wood$cor),
                         method = "REML",
                         data = MSBP.tree.cor.wood$MSBP.tree, verbose=F, digits=5,
                         control=list(optimizer = "optim", optmethod = "Nelder-Mead", maxit= 10000))

## Future warming risk ~ Woodiness only (without phylogeny)
```

```r
wood.mv.WT <- rma.mv(yi = Future.Hot.Quart - Tmax,
                     V = vi,
                     mod = ~woodiness + log10(SeedAge) + altitude,
                     random = list(~1 | Taxon_ID, ~1 | grid.ll),
                     method = "REML",
                     data = MSBP.meta.wood)

#Boxplot comparing woody and herbaceous species
woodbox <- ggplot(data = MSBP.meta.wood, aes(x = woodiness, y = Future.Hot.Quart - Tmax)) +
            geom_boxplot(fill = "orchid") +
            theme_classic()+
            ylab(expression("Predicted 2070 Environment Temperature - \n Maximum Germination Temperatur
            xlab("") +
            scale_x_discrete(breaks=c("W", "H"),  labels=c("Woody", "Herbaceous"))+
            theme(legend.position="none", plot.margin = margin(10,10,10,30))

#Filter dataset for analysis (with phylogeny)
MSBP.meta.long <- mutate(MSBPlong, vi = SEmax*SEmax) %>%
                  filter(ann_per == "annual"|ann_per == "perennial")

#Filter dataset for analysis (without phylogeny)
MSBP.meta.long.phy <- mutate(MSBPlong, vi = SEmax*SEmax) %>%
                      filter(ann_per == "annual"|ann_per == "perennial")%>%
                      distinct(Taxon_ID, .keep_all = T) # needed for phlogentic analysis

#Get Phylogentic Matrix
MSBP.tree.cor <- getPhyloCor(MSBP.meta.long.phy)


## Future warming risk ~ Longevity only (with phylogeny)
long.mv.WT.phy <- rma.mv(yi = Future.Hot.Quart - Tmax,
                     V = vi,
                     mod = ~ann_per + log10(SeedAge) + altitude,
                     random = list(~1 | Taxon_ID, ~1 | grid.ll, ~1 | phylo),
                     R = list(phylo = MSBP.tree.cor$cor),
                     method = "REML",
                     data = MSBP.tree.cor$MSBP.tree, verbose=F, digits=5,
                     control=list(optimizer = "optim", optmethod = "Nelder-Mead", maxit= 10000))

## Future warming risk ~ Longevity only (without phylogeny)
long.mv.WT <- rma.mv(yi = Future.Hot.Quart - Tmax,
                     V = vi,
                     mod = ~ann_per  + log10(SeedAge) + altitude,
                     random = list(~1 | Taxon_ID, ~1 | grid.ll),
                     method = "REML",
                     data = MSBP.meta.long)


#Boxplot comparing woody and herbaceous species
longbox <- ggplot(data = MSBP.meta.long, aes(x = ann_per, y = Future.Hot.Quart - Tmax)) +
            geom_boxplot(fill = "orchid") +
            theme_classic()+
            ylab(expression("Predicted 2070 Environment Temperature - \n Maximum Germination Temperatur
```

```
            xlab("") +
            scale_x_discrete(breaks=c("annual", "perennial"),  labels=c("Annual", "Perennial"))+
            theme(legend.position="none", plot.margin = margin(10,10,10,30))


output <- list(Wood.Model = wood.mv.WT,
                 Wood.Model.Phy = wood.mv.WT.phy,
                 Wood.Plot = woodbox,
                 Long.Model = long.mv.WT,
                 Long.Model.Phy = long.mv.WT.phy,
                 Long.Plot = longbox)

return(output)

}

WR.wood.life <- getLifeWoodWR()
```

germTree.R

```
require(ggplot2)
require(dplyr)
require(tidyr)
require(ape)
require(ggtree)
require(viridis)

MSBP <- read.csv(file = "./Data/MSBP_Clean") #core dataset

#Prepare data set, if SE's are too high, the analyses don't run (but get weighted out anyway)
MSBP.tree.WT <- MSBP %>%
  mutate(vi = SEmax*SEmax) %>% #Standard error for warming risk
  filter(!is.na(Tmax) &
         !is.na(Future.Hot.Quart)) %>% #Filter NAs
  distinct(Taxon_ID, .keep_all = T) #Remove duplicate species

#Seed plant phylogeny - Smith and Brown 2019
phyall <- read.tree(file = "./Data/Phylo/ALLMB.tre")

#List of taxa
rownames(MSBP.tree.WT) <- MSBP.tree.WT$Taxon_ID

MSBP.Species.WT <- as.vector(MSBP.tree.WT$Taxon_ID)

#Prune full tree for matches
PrunedMSBP <- drop.tip(phyall, phyall$tip.label[na.omit(-match(MSBP.Species.WT, phyall$tip.label))])
MSBP.tree.WT <- rename(MSBP.tree.WT, "PrunedMSBP$tip.label" = Taxon_ID)
MSBP.tree.WT <- right_join(MSBP.tree.WT, as.data.frame(PrunedMSBP$tip.label), by = "PrunedMSBP$tip.label
MSBP.tree.WT <- rename(MSBP.tree.WT, Taxon_ID="PrunedMSBP$tip.label")
MSBP.Species.WT <- as.vector(MSBP.tree.WT$Taxon_ID)
MSBP.tree.WT[, "phylo"] <- MSBP.Species.WT

tree.df <- data.frame(Taxon_ID = PrunedMSBP$tip.label)
```

```
tree.df <- data.frame(left_join(tree.df, MSBP.tree.WT, by = "Taxon_ID"))

pruned.tree <- ggtree(PrunedMSBP)


p.tree <- pruned.tree %<+% filter(tree.df) +
  geom_segment2(aes(subset=isTip,
                    yend = y,
                    xend = x+70,
                    color = (Future.Hot.Quart) - Tmax,
                    alpha=(1/log(SEmax))),
                size = 2.5)+
  scale_colour_viridis(option = "plasma")+
  theme(legend.position=c(0,0.7))+
  guides(alpha=FALSE) #removes alpha key

pnames <- geom_tiplab2(size = 2, offset = 30)

pclades <- p.tree +
  theme(plot.margin = unit(c(2,5,2,2), "cm"))+

coord_cartesian(xlim=c(0,1000))

pclades

ggsave(filename = "./Outputs/MSBPtreelabels.png", plot = pclades, height = 15, width =10)
```

germPhyloSignal.R

```
library(dplyr)
require(tidyr)
require(ggtree)
require(viridis)
require(phylobase)
library("phylosignal")
require(ape)
require(phytools)


#core dataset
MSBP <- read.csv(file = "./Data/MSBP_Clean") #core dataset

#Prepare data set
MSBP.tree <- MSBP %>%
             mutate(vi = SEmax*SEmax) %>% #Standard error for future warming risk
             filter(!is.na(Tmax) & #Filter where Tmax and Future.Hot.Quart is not NA
                    !is.na(Future.Hot.Quart) &
                    SEmax < 200) %>% #and SEmax is lower than 200
             mutate(WR = Future.Hot.Quart - SEmax) %>% #Create FWT variable
             distinct(Taxon_ID, .keep_all = T)



#Vascular plant tree
```

```
phyall <- read.tree(file = "./Data/Phylo/ALLMB.tre")

rownames(MSBP.tree) <- MSBP.tree$Taxon_ID

MSBP.Species <- as.vector(MSBP.tree$Taxon_ID)

#Prune full tree for matches
PrunedMSBP <- drop.tip(phyall, phyall$tip.label[na.omit(-match(MSBP.Species, phyall$tip.label))])
MSBP.tree <- rename(MSBP.tree, "PrunedMSBP$tip.label" = Taxon_ID)
MSBP.tree <- right_join(MSBP.tree, as.data.frame(PrunedMSBP$tip.label), by = "PrunedMSBP$tip.label")
MSBP.tree <- rename(MSBP.tree, Taxon_ID="PrunedMSBP$tip.label")
MSBP.Species <- as.vector(MSBP.tree$Taxon_ID)
MSBP.tree[, "phylo"] <- MSBP.Species
tree <- compute.brlen(PrunedMSBP)
cor <- vcv(tree, cor = T)

#List of taxa

MSBP4d <- phylo4d(tree, tip.data = MSBP.tree$WR)

WRSignal <- phyloSignal(MSBP4d, rep =999)

WRSignal
```

germWarmingRiskCurrent.R

Current Warming Risk analayses and outputs For Seed Germination over Latitude Project

```
getWRC <- function() {

  require(dplyr)
  require(tidyr)
  require(metafor)
  require(ggplot2)
  source("./R/getPhyloCor.R")

  MSBP <- read.csv(file = "./Data/MSBP_Clean") #core dataset

  ### Prepare data set, if SE's are too high, the analyses don't run
  # (but get weighted out of the model anyway)
  MSBP.WR.data <- MSBP %>%
    mutate(vi = SEmax * SEmax) %>% #Standard error for warming risk
    filter(!is.na(Tmax) & #Remove NAs
           !is.na(Current.Hot.Quart))  #Remove NAs

  MSBP.WR.data.phy <-
    MSBP.WR.data %>%
    distinct(Taxon_ID, .keep_all = T) #Needed for phylogeny analysis

  # Get phylogeny tree[[1]] and correlation matrix[[2]]
  MSBP.tree.cor <- getPhyloCor(MSBP.WR.data.phy)

  ## Future Warming Risk ~ Latitude and Hemispshere (with phylogeny)
  #WR, using random effect of phylo, site (grid.ll) and species (Taxon_ID), and altitude
  WR.h.phy <- rma.mv(
```

```
  yi = Current.Hot.Quart - Tmax,
  V = vi,
  mod = ~ AbsLat + (AbsLat:NorthTF) +
    log10(SeedAge) + altitude,
  random = list( ~ 1 | phylo, ~ 1 | Taxon_ID,
                 ~ 1 | grid.ll),
  R = list(phylo = MSBP.tree.cor$cor),
  method = "REML",
  data = MSBP.tree.cor$MSBP.tree,
  verbose = TRUE,
  digits = 5,
  control = list(optimizer = "optim", optmethod = "Nelder-Mead")
)

summary(WR.h.phy)


WR.phy <- rma.mv(
  yi = Current.Hot.Quart - Tmax,
  V = vi,
  mod = ~ AbsLat+ log10(SeedAge) + altitude,
  random = list( ~ 1 | phylo, ~ 1 | Taxon_ID,
                 ~ 1 | grid.ll),
  R = list(phylo = MSBP.tree.cor$cor),
  method = "REML",
  data = MSBP.tree.cor$MSBP.tree,
  verbose = TRUE,
  digits = 5,
  control = list(optimizer = "optim", optmethod = "Nelder-Mead")
)
summary(WR.phy)


## Future Warming Risk ~ Latitude and Hemispshere (with phylogeny)
#WR, using random effect of phylo, site (grid.ll) and species (Taxon_ID), and altitude
WR.h <- rma.mv(
  yi = Current.Hot.Quart - Tmax,
  V = vi,
  mod = ~ AbsLat + (AbsLat:NorthTF) +
    log10(SeedAge) + altitude,
  random = list(~ 1 | Taxon_ID, ~ 1 | grid.ll),
  method = "REML",
  data = MSBP.WR.data
)

summary(WR.h)


WR <- rma.mv(
  yi = Current.Hot.Quart - Tmax,
  V = vi,
  mod = ~ AbsLat + log10(SeedAge) + altitude,
  random = list(~ 1 | Taxon_ID, ~ 1 | grid.ll),
```

```
  method = "REML",
  data = MSBP.WR.data
)

summary(WR)

#Plot of future WR with lat and hemisphere effect

WR.plot.c <- ggplot(data = MSBP.WR.data , aes(x = AbsLat)) +
  geom_rect(aes(
    xmin = 0,
    xmax = 23.5,
    ymax = Inf,
    ymin = -Inf
  ), fill = "grey90") +
  geom_rug(aes(y = Current.Hot.Quart - Tmax), alpha = 1/2, position = "jitter", sides="b") +
  geom_point(aes(y = Current.Hot.Quart - Tmax,
                 alpha = log(1 / (vi))), colour = "Red") +
  geom_hline(aes(yintercept = 0), size = 0.5) +
  theme_classic() +
  ylab(expression(
    "2070 Environmental Temperature - \n Maximum Germination Temperature   "(degree * C)
  )) +
  xlab(expression("Absolute Latitude "(degree))) +
  theme(legend.position = "none",
        plot.margin = margin(10,10,10,30))+
  ylim (-45,15)+
  scale_x_continuous(limits = c(0,60), expand = c(0,0))

WR.plot.c


#Values to get confidence intervals
mods <- cbind(AbsLat = c(1:57),
              SeedAge = rep(log10(median(MSBP.WR.data$SeedAge, na.rm =T)), 57),
              altitude= rep(median(MSBP.WR.data$altitude, na.rm =T), 57))

### calculate predicted values from model
preds1 <- predict(WR, newmods=mods, addx=T)

preds1

ci.plot1 <-as.data.frame(cbind(c(seq(1, 57)),
                               preds1[[1]],
                               preds1[[3]],
                               preds1[[4]]))

#Plot original figure with confidence intervals from above
WR.plot.c.ci <-
  WR.plot.c +
  geom_line(data = filter(ci.plot1, V1 > -55),
            aes(x = V1, y = V2),
            colour = "red",
```

```
              linetype = "solid") +
    geom_line(data = filter(ci.plot1, V1 > -55),
              aes(x = V1, y = V3),
              linetype = "dashed") +
    geom_line(data = filter(ci.plot1, V1 > -55),
              aes(x = V1, y = V4),
              linetype = "dashed") +
    xlab("Latitude (°)")

  WR.plot.c.ci

  ggsave(
    WR.plot.c.ci,
    filename = "./Outputs/WRSE_current.tiff",
    units = "in",
    width = 7,
    height = 5,
    dpi = 1000,
    compression = 'lzw')

  #Return results
  output <- list(
    Model = WR,
    Model.phy = WR.phy,
    Plot = WR.plot.c.ci,
    Model.h = WR.h,
    Model.h.phy = WR.h.phy
  )

  return(output)

}


WRC <- getWRC()
```

germClimateMismatchFuture.R

This is code to run analayses and create outputs for Future Climate Mismatch Results in supplement For Seed Germination over Latitude Project

```
getCMF <- function() {

  require(dplyr)
  require(tidyr)
  require(metafor)
  require(ggplot2)
  source("./R/getPhyloCor.R")

  MSBP <- read.csv(file = "./Data/MSBP_Clean") #core dataset

  #Prepare data set, if SE's are too high, the analyses don't run (but get weighted out of the model an
  MSBP.CM <- MSBP %>%
    mutate(vi = SEopt.upp * SEopt.upp) %>% #Standard error for opt
    filter(!is.na(Topt.upp) &
```

```
         !is.na(Future.Hot.Quart))

MSBP.CM.phy <- distinct(MSBP.CM, Taxon_ID, .keep_all = T) #Needed for phylogeny analysis

#Get phylogeny tree[[1]] and correlation matrix[[2]]
MSBP.CM.phy <- getPhyloCor(MSBP.CM.phy)

## Current Climate Mismatch ~ Latitude and Hemispshere (without phylogeny)
#using random effect of site (grid.ll) and species (Taxon_ID), hemisphere
CM.h <- rma.mv(
  yi = Future.Hot.Quart - Topt.upp,
  V = vi,
  mod = ~ AbsLat + (AbsLat:NorthTF) + log10(SeedAge) + altitude,
  random = list(~ 1 | Taxon_ID,
                ~ 1 | grid.ll),
  method = "REML",
  data = MSBP.CM
)

## Current Climate Mismatch ~ Latitude and Hemispshere (with phylogeny)
#using random effect of site (grid.ll) and species (Taxon_ID), hemisphere
CM.h.phy <- rma.mv(
  yi = Future.Hot.Quart - Topt.upp,
  V = vi,
  mod = ~ AbsLat + (AbsLat:NorthTF)+ log10(SeedAge) + altitude,
  random = list( ~ 1 | phylo,
                 ~ 1 | Taxon_ID,
                 ~ 1 | grid.ll),
  R = list(phylo = MSBP.CM.phy$cor),
  method = "REML",
  data = (MSBP.CM.phy[[1]]),
  verbose = TRUE,
  digits = 5,
  control = list(optimizer = "optim", optmethod = "Nelder-Mead")
)

## Current Climate Mismatch ~ Latitude only (without phylogeny)
#TSM using random effect of site (grid.ll) and species (Taxon_ID), no hemisphere
CM <- rma.mv(
  yi = Future.Hot.Quart - Topt.upp,
  V = vi,
  mod = ~ AbsLat + log10(SeedAge) + altitude,
  random = list( ~ 1 | Taxon_ID,
                 ~ 1 | grid.ll),
  method = "REML",
  data = MSBP.CM
)

CM.low <- rma.mv(
  yi = Future.Hot.Quart - Topt.low,
  V = vi,
  mod = ~ AbsLat+ log10(SeedAge) + altitude,
  random = list( ~ 1 | Taxon_ID,
```

```
                  ~ 1 | grid.ll),
  method = "REML",
  data = MSBP.CM
)

## Current Climate Mismatch ~ Latitude only (with phylogeny)
#TSM using random effect of site (grid.ll) and species (Taxon_ID), no hemisphere
CM.phy <- rma.mv(
  yi = Future.Hot.Quart - Topt.upp,
  V = vi,
  mod = ~ AbsLat + log10(SeedAge) + altitude,
  random = list( ~ 1 | phylo,
                 ~ 1 | Taxon_ID,
                 ~ 1 | grid.ll),
  R = list(phylo = MSBP.CM.phy$cor),
  method = "REML",
  data = MSBP.CM.phy[[1]]
)

CM.plot <- ggplot(data = MSBP.CM , aes(x = AbsLat)) +
  geom_rect(aes(
    xmin = 0, xmax = 23.5,
    ymax = Inf, ymin = -Inf
  ), fill = "grey90") +
  geom_point(aes(y = Future.Hot.Quart - Topt.upp,
                 alpha = log(1 / vi)), colour = "orange") +
  geom_rug(aes(y = Future.Hot.Quart - Topt.upp),
           alpha = 1 / 2,
           position = "jitter",
           sides = "b") +
  geom_hline(aes(yintercept = 0), size = 0.5) +
  theme_classic() +
  ylab(expression(
    "Future Environmental Temperature - \n Upper Optimal Germination Temperature  "
    (degree * C))) +
  xlab(expression("Absolute Latitude "(degree))) +
  theme(legend.position = "none",
        plot.margin = margin(10, 10, 10, 30)) +
  ylim (-35, 15) +
  scale_x_continuous(limits = c(0, 60), expand = c(0, 0))

#Values to get confidence intervals
mods <- cbind(AbsLat = c(1:57),
              SeedAge = rep(log10(median(MSBP.CM$SeedAge, na.rm =T)), 57),
              altitude= rep(median(MSBP.CM$altitude, na.rm =T), 57)) #abslat - 1 to 57

### calculate predicted values from model
preds1 <- predict(CM, newmods=mods, addx=T)

ci.plot1 <- as.data.frame(
  cbind(c(seq(1, 57)),
        preds1[[1]],
        preds1[[3]],
```

```
          preds1[[4]]))

  #Plot original figure with confidence intervals from above
  CM.plot.f.ci <-  CM.plot +
    geom_line(data = filter(ci.plot1, V1 > -55),
              aes(x = V1, y = V2),
              colour = "orange",
              linetype = "solid") +
    geom_line(data = filter(ci.plot1, V1 > -55),
              aes(x = V1, y = V3),
              linetype = "dashed") +
    geom_line(data = filter(ci.plot1, V1 > -55),
              aes(x = V1, y = V4),
              linetype = "dashed") +
    xlab("Latitude (°)")

  ggsave(
  CM.plot.f.ci,
  filename = "./Outputs/CVSE_future.tiff",
  units = "in",
  width = 7,
  height = 5,
  dpi = 1000,
  compression = 'lzw')


  output <- list(
    Model = CM,
    Model.Phy = CM.phy,
    Plot = CM.plot.f.ci,
    Model.h = CM.h,
    Model.h.Phy = CM.h.phy,
    Model.low = CM.low
  )

  return(output)
}

CMF <- getCMF()
```

germMaxMinOpt.R

Run tests against latitude

```
lat.tests <- function(MSBP, Temp, SETemp){
require(dplyr)
require(tidyr)
require(metafor)
require(ggplot2)
source("./R/getPhyloCor.R")

#Prepare data sets
MSBP.temp <- MSBP %>%
  mutate(vi = get(SETemp) * get(SETemp)) %>% #Standard error for opt
  filter(!is.na(get(Temp)) &
```

```
         !is.na(SeedAge) &
         !is.na(altitude))

#Remove duplicate species for phylogeny analysis
MSBP.temp.phy <- distinct(MSBP.temp, Taxon_ID, .keep_all = T)

#Get phylogeny tree[[1]] and correlation matrix[[2]]
MSBP.temp.phy <- getPhyloCor(MSBP.temp.phy)

model.h <- rma.mv(
  yi = get(Temp),
  V = vi,
  mod = ~ AbsLat + (AbsLat:NorthTF) + log10(SeedAge) + altitude,
  random = list(~ 1 | Taxon_ID,
                ~ 1 | grid.ll),
  method = "REML",
  data = MSBP.temp
)


model.h.phy <- rma.mv(
  yi = get(Temp),
  V = vi,
  mod = ~ AbsLat + (AbsLat:NorthTF) + log10(SeedAge) + altitude,
  random = list( ~ 1 | phylo,
                 ~ 1 | Taxon_ID,
                 ~ 1 | grid.ll),
  R = list(phylo = MSBP.temp.phy$cor),
  method = "REML",
  data = (MSBP.temp.phy[[1]]),
  verbose = TRUE,
  digits = 5,
  control = list(optimizer = "optim", optmethod = "Nelder-Mead")
)


model <- rma.mv(
  yi = get(Temp),
  V = vi,
  mod = ~ AbsLat + log10(SeedAge) + altitude,
  random = list( ~ 1 | Taxon_ID,
                 ~ 1 | grid.ll),
  method = "REML",
  data = MSBP.temp
)

model.phy <- rma.mv(
  yi = get(Temp),
  V = vi,
  mod = ~ AbsLat + log10(SeedAge) + altitude,
  random = list( ~ 1 | phylo,
                 ~ 1 | Taxon_ID,
                 ~ 1 | grid.ll),
```

```
  R = list(phylo = MSBP.temp.phy$cor),
  method = "REML",
  data = MSBP.temp.phy[[1]]
)


output <- list(
  Model = model,
  Model.Phy = model.phy,
  Data = MSBP.temp,
  Model.h = model.h,
  Model.h.Phy = model.h.phy
)
  return(output)

}

MSBP <- read.csv(file = "./Data/MSBP_Clean") #core dataset

Tmax.lat <- lat.tests(MSBP = MSBP, Temp = "Tmax", SETemp = "SEmax")
Topt.upp.lat <- lat.tests(MSBP = MSBP, Temp = "Topt.upp", SETemp = "SEopt.upp")
Topt.low.lat <- lat.tests(MSBP = MSBP, Temp = "Topt.low", SETemp = "SEopt.low")
Tmin.lat <- lat.tests(MSBP = MSBP, Temp = "Tmin", SETemp = "SEmin")
Topt.range.lat <- lat.tests(MSBP = MSBP, Temp = "Toptbreadth", SETemp = "SEoptbreadth")



Tmax.lat$Model.h.Phy #hemi not significant
Tmax.lat$Model.h #hemi significant
Tmax.lat$Model.Phy #both abslat and alititide significant
Tmax.lat$Model #both abslat and alititide significant

Topt.upp.lat$Model.h.Phy #hemi not significant
Topt.upp.lat$Model.h #hemi not significant
Topt.upp.lat$Model.Phy #abslat significant
Topt.upp.lat$Model #abslat significant

Topt.low.lat$Model.h.Phy #hemi not significant
Topt.low.lat$Model.h #hemi not significant
Topt.low.lat$Model.Phy #abslat significant
Topt.low.lat$Model #abslat significant

Tmin.lat$Model.h.Phy #hemi not significant
Tmin.lat$Model.h #hemi not significant
Tmin.lat$Model.Phy #both abslat and alititide significant
Tmin.lat$Model #both abslat and alititide significant


Topt.range.lat$Model.h.Phy
Topt.range.lat$Model.h
Topt.range.lat$Model.Phy
Topt.range.lat$Model
```

```r
R2(Tmax.lat$Model.Phy)

R2(Topt.upp.lat$Model.Phy)

R2(Topt.low.lat$Model.Phy)

R2(Tmin.lat$Model.Phy)


R2(Tmax.lat$Model)

R2(Topt.upp.lat$Model)

R2(Topt.low.lat$Model)

R2(Tmin.lat$Model)

maxminlines <- ggplot(data = filter(MSBP) , aes(x = AbsLat)) +
  theme_classic() +
  geom_point(data = Tmax.lat$data,
             aes(y = Tmax, alpha = log(1 / SEmax)),
             colour = "red") +
  geom_point(data = Tmin.lat$data,
             aes(y = Tmin, alpha = log(1 /SEmin)),
             colour = "blue") +
  geom_segment(x = 0.5, y = coef(Tmax.lat$Model)[1]+
                 (coef(Tmax.lat$Model)[2])*0.5+
                 (coef(Tmax.lat$Model)[3])*3.07+
                 (coef(Tmax.lat$Model)[4])*462,
               xend = 65, yend = (coef(Tmax.lat$Model)[1]) +
                 (coef(Tmax.lat$Model)[2])*65+
                 (coef(Tmax.lat$Model)[3])*3.07+
                 (coef(Tmax.lat$Model)[4])*462,
               size = 1.5, colour = "red")+
 geom_segment(x = 0.5, y = coef(Tmin.lat$Model)[1]+
                (coef(Tmin.lat$Model)[2])*0.5+
                (coef(Tmin.lat$Model)[3])*3.07+
                (coef(Tmin.lat$Model)[4])*462,
              xend = 67, yend = (coef(Tmin.lat$Model)[1]) +
                (coef(Tmin.lat$Model)[2])*67+
                (coef(Tmin.lat$Model)[3])*3.06+
                (coef(Tmin.lat$Model)[4])*462.3,
              size = 1.5, colour = "blue")+
  ylab("Temperature (°C)")+
  xlab("Absolute Latitude (°)") +
  theme(legend.position="none")+
  scale_y_continuous(limits = c(-10,50)) +
  scale_x_continuous(limits = c(0,70), expand = c(0,0))

maxminlines
```

```
optlines <- ggplot(data = filter(MSBP) , aes(x = AbsLat)) +
  theme_classic() +
  geom_point(data = Topt.upp.lat$data,
             aes(y = Topt.upp, alpha = log(1 / SEopt.upp)),
             colour = "darkorange") +
  geom_point(data = Topt.low.lat$data,
             aes(y = Topt.low, alpha = log(1 /SEopt.low)),
             colour = "orange") +
  geom_segment(x = 0.5, y = coef(Topt.upp.lat$Model)[1]+
                 (coef(Topt.upp.lat$Model)[2])*0.5+
                 (coef(Topt.upp.lat$Model)[3])*3.15569+
                 (coef(Topt.upp.lat$Model)[4])*427,
               xend = 62, yend = (coef(Topt.upp.lat$Model)[1]) +
                 (coef(Topt.upp.lat$Model)[2])*62+
                 (coef(Topt.upp.lat$Model)[3])*3.15569+
                 (coef(Topt.upp.lat$Model)[4])*427,
               size = 1.5, colour = "darkorange")+
  geom_segment(x = 0.5, y = coef(Topt.low.lat$Model)[1]+
                 (coef(Topt.low.lat$Model)[2])*0.5+
                 (coef(Topt.low.lat$Model)[3])*3.15569+
                 (coef(Topt.low.lat$Model)[4])*427,
               xend = 62, yend = (coef(Topt.low.lat$Model)[1]) +
                 (coef(Topt.low.lat$Model)[2])*62+
                 (coef(Topt.low.lat$Model)[3])*3.15569+
                 (coef(Topt.low.lat$Model)[4])*427,
               size = 1.5, colour = "orange")+
  geom_segment(aes(x = AbsLat, y = Topt.low,
                   xend = AbsLat, yend = Topt.upp),
               size = 0.5, colour = "orange")+
  ylab("Temperature (°C)")+
  xlab("Absolute Latitude (°)") +
  theme(legend.position="none")+
  scale_y_continuous(limits = c(0,50)) +
  scale_x_continuous(limits = c(0,70), expand = c(0,0))

optlines

ggsave(
  maxminlines,
  filename = "./Outputs/maxminlines.tiff",
  units = "in",
  width = 7,
  height = 5,
  dpi = 1000,
  compression = 'lzw')


ggsave(
  optlines,
  filename = "./Outputs/optlines.tiff",
  units = "in",
  width = 7,
  height = 5,
```

```
  dpi = 1000,
  compression = 'lzw')
```