ArchaeoGLOBE Analysis

Nick Gauthier and Ben Marwick

Last knit on: 13 February, 2019

Analysis code for the ArchaeoGLOBE project. This notebook was used to produce all the analyses and figures in the publication. All data are downloaded from their associated online repositories prior to the analysis.

Setup

Import packages needed for analysis. We'll use packages from the `tidyverse`, such as `readr`, `dplyr`, and `ggplot2` for data import, processing, and plotting. We'll also use `mgcv` for fitting nonlinear trends to the data. We'll use the `sf` and `raster` packages to handle spatial data and plotting. The `dataverse` and `osfr` packages allows us to pull the raw survey data and precomputed analysis files from their online repositories. Finally, we'll use `patchwork` (installed from GitHub) to combine multiple ggplots in the same image.

```
library(raster)
library(tidyverse)
library(mgcv)
library(sf)
library(ggplot2)
library(dataverse)

#install patchwork  and osfr from github if needed
#devtools::install_github('thomasp85/patchwork')
library(patchwork)
# devtools::install_github('centerforopenscience/osfr')
library(osfr)
```

Data import

Download all data necessary for the analysis and import into R. By default, the code chunks that actually download the data are hidden here, so please refer to the source .rmd document for the relevant code.

Read in the latest version of the ArchaeoGLOBE database and the consensus assessment from the Dataverse repository. Refer to the source .rmd document for the code to download the shapefiles from the Dataverse repository.

```
archaeoglobe <- read_csv('data/raw-data/ARCHAEOGLOBE_PUBLIC_DATA.tab')
consensus <- read_csv('data/raw-data/ARCHAEOGLOBE_CONSENSUS_ASSESSMENT.tab')
```

Repeat for the archaeological regions shapefile. We'll use the "simplified regions" shapefile for plotting purposes, and the original shapefile for the ArchaeoGLOBE – HYDE comparison at the end of this notebook. Refer to the source .rmd document for the code to download the shapefiles from the Dataverse repository.

```
# read into the current environment
regions <- st_read('data/raw-data/ArchaeoGLOBE_Simplified_Regions/ArchaeoGLOBE_Simplified_Regions.shp',
         quiet = TRUE) %>%
  # add labels for just the islands, will make plotting easier in the future
  mutate(region_label = replace(Archaeo_RG, !(Archaeo_RG %in%
                                              c('Hawaii','Polynesia','Micronesia','Melanesia')), NA))

regions_hyde <- st_read('data/raw-data/ArchaeoGLOBE_Regions/ArchaeGLOBE_Regions.shp',
         quiet = TRUE) %>%
  # reproject to match HYDE data
  st_transform('+proj=longlat +datum=WGS84 +no_defs +ellps=WGS84 +towgs84=0,0,0')
```

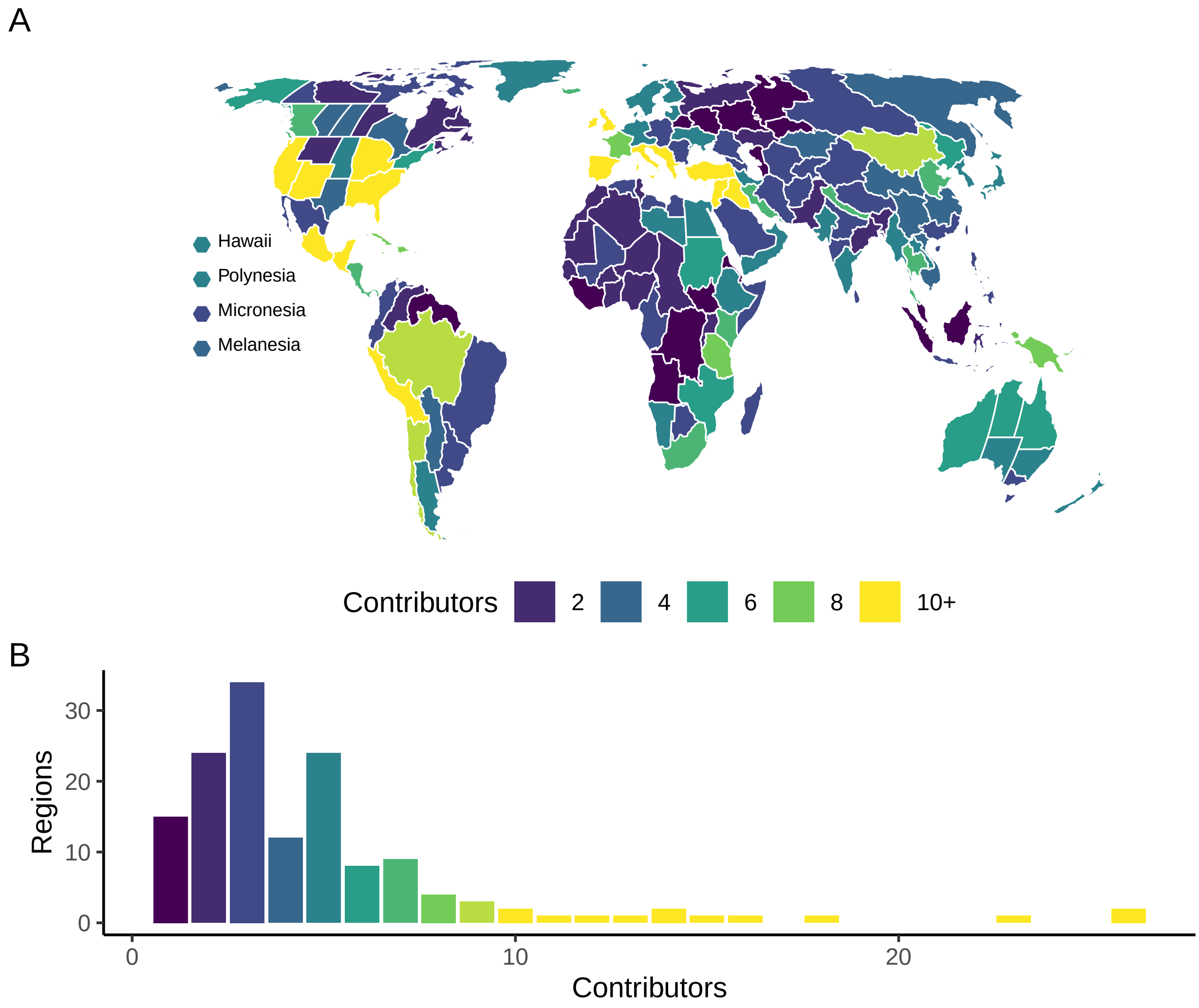
Download and import HYDE raster data, version 3.2. Refer to the source .rmd document for the code to download the .zip files from the ftp server. If running for the first time, this will download about 500mb to your computer.

```
hyde <- list.files('data/raw-data/HYDE', full.names = TRUE) %>%
  .[c(11:8, 3, 1:2, 4:7)] %>% # temporal order
  map(raster) %>%
  brick %>%
  `crs<-`(value = '+proj=longlat +datum=WGS84 +no_defs +ellps=WGS84 +towgs84=0,0,0')
```

Exploratory visualization

Before running any analyses, let's look at the data. How many responses do we have per region?

```
response_counts <- archaeoglobe %>%
  group_by(REGION_ID) %>%
  count %>%
  mutate(n10 = replace(n, n > 10, 10))
```

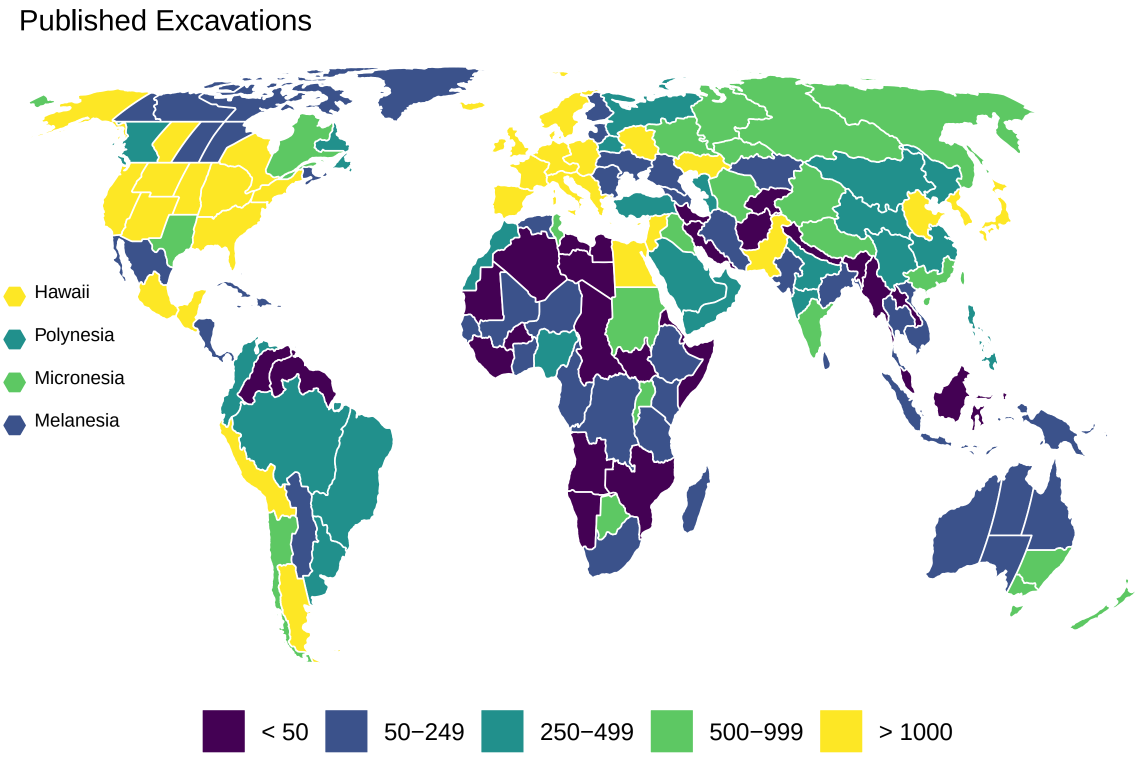


How many published archaeolgical excavations are estimated for each region?

```
# a function for calculating the mode, from https://stackoverflow.com/a/46846474
calculate_mode <- function(x) {
  uniqx <- unique(x)
  uniqx[which.max(tabulate(match(x, uniqx)))]
}

# use this vector to order the RN_SITES variable by increasing number of sites
site_order <- c('< 50', '50-249', '250-499', '500-999', '> 1000')

# find the modal response for the number of published excavations in each region
site_counts <- archaeoglobe %>%
  select(REGION_ID, RN_SITES) %>%
  group_by(REGION_ID) %>%
  summarise(sites = calculate_mode(RN_SITES)) %>%
  mutate(sites = ordered(sites, levels = site_order))
```



GAMM Trends

Here we use Generalized Additive Models (GAMs), a flexible form of nonlinear regression model capable of fitting smooth, time-varying trends to the ordered categorical ArchaeoGLOBE response data. We model ordered categorical data using a latent variable following a logistic distribution. The model identifies a series of cut points, which correspond the the probabilities of the latent variable falling within each of our categories.

We fit two sets of trends. One trend is fitted to all the data simultaneously, representing the global trend across all archaeological regions. Then we fit region-level trends, which represent the deviation of each region from the global trend. By penalizing the "wiggliness" of the trend lines, we allow regional trends that don't significantly deviate from the global trend to be penalized to 0, effectively reducing that particular region to the global trend. This is a form of partial pooling, allowing the model to share information between groups and in so doing make the results less sensitive to regions with exceptionally low response rates.

After fitting the model, we can extract the region-specific trends, use a k-means clustering algorithm to group together regions with similar trends, and map the results. We repeat this analysis for both self-reported expertise and perceived data quality.

Analysis functions

Define some analysis functions that we'll be using repeatedly in the analysis, so that we don't have to keep copying and pasting the same lines of code.

This function subsets the data to highlight a variable of interest, and converts it from a wide to a long "tidy" format to make analysis and plotting easier.

```
preprocess <- function(prefix, categories){
  archaeoglobe %>% # start with the full ArcheoGlobe data
    # drop columns not related to the variable of interest
    select(c(CONTRIBUTR:LAND_AREA, starts_with(prefix))) %>%
    gather(time, value, starts_with(prefix)) %>% # one value per row
    mutate(time = parse_number(time) * -1, # convert time period labels to years
           value = ordered(value, levels = categories),
           cat_num = as.numeric(value)) %>%
    mutate_if(is.character, as.factor) # convert characters to factors
}
```

This function takes a data frame produced by the above function and fits GAM to the global trend and local deviations for each region, accounting for inter-observer variability. This function takes as arguments a preprocessed data frame containing time slices, regions, contributors, and the ordered categorical response variable transformed to a numeric vector.

```
cores <- max(parallel::detectCores() / 2, 1) # physical cores for parallelization
cl <- parallel::makeCluster(cores)

fit_gam <- function(x, n_cats){
  bam(cat_num ~
        # this spline is for the global trend
        s(time, bs = 'cr', m = 2) +
        # region-specific trends. bs = 'ts' and m = 1
        # help penalize deviation from the global model
        s(time, by = REGION_LAB, bs = 'cs', m = 1) +
        # add back in region-specific intercepts
        REGION_LAB  +
        # model contributor as a random effect
        s(CONTRIBUTR, bs = 're'),
      data = x, # data frame to analyize
      family = ocat(R = n_cats), # ordered categorical with n levels
      # final 3 arguments just speed up the model fitting
      method = 'fREML',
      discrete = TRUE,
      cluster = cl)
}
```

This function extracts the estimated trends for each region, incorporating the global and regional splines as well as the region and contributor specific intercepts. Then it clusters these trends into 6 discrete clusters using k-means. The choice of 6 clusters is somewhat arbitrary, and is made simply based on visual comparisons of different cluster solutions with the goal of ensuring visually interpretable results.

```
extract_trends <-function(mod, n_clusters = 6){
  set.seed(1000) # set seed for reproducability of clusters
  archaeoglobe %>% # create dummy data for prediction in the following lines
    select(REGION_LAB) %>%
    group_by(REGION_LAB) %>%
    slice(1) %>%
    slice(rep(1:n(), each = 198)) %>%
    ungroup %>%
    mutate(time = rep_len(seq(-10000, -150, 50), n()),
           CONTRIBUTR = 'CYRBU') %>% # select an arbitrary contributor
    mutate(preds = predict(mod, .)) %>% # estimate trend lines
    mutate(preds = plogis(preds)) %>% # transform responses to [0,1] scale
    spread(time, preds) %>%
    # next is the actual kmeans clustering code
    mutate(cluster = kmeans(.[,-c(1,2)], n_clusters, iter.max = 100, nstart = 100)$cluster)
}
```

Analysis

Now we use the functions defined above to estimate trends in ArchaeoGlobe data. For convenience, first define a data frame that lists the prefixes of the variables we are interested in (e.g. "EXP" for expertise) and the levels of the ordered factors associated with each variable. This will make it easier to quickly focus on a specific variable. The `tribble` command is simply a way to make a data frame by row rather than column, which makes the code easier to read.

```
response_levels <- tribble(
  ~prefix, ~categories,
  'EXP', c('None', 'Low', 'High'),                    # Expertise
  'DQ', c('Unknown', 'Low', 'Moderate', 'Good'),      # Data Quality
  'HUNT', c('none', 'minimal (<1%)', 'common (1-20%)', 'widespread (>20%)'),
  'EXAG', c('none', 'minimal (<1%)', 'common (1-20%)', 'widespread (>20%)'),
  'INAG', c('none', 'minimal (<1%)', 'common (1-20%)', 'widespread (>20%)'),
  'PAST', c('none', 'minimal (<1%)', 'common (1-20%)', 'widespread (>20%)'),
  'URBN', c('Absent', 'Present')
)
```

Now map each of the above functions to each variable. This allows us to run the analysis for all variables of interest in a single step, and save all the outputs in a tibble format for easy plotting. If you're running this for the first time, it should take about 40 minutes to run on a Intel NUC with a 5th-gen Intel Core i7-5557U processor and 16gb of RAM running Linux. By default, we pull pre-computed results from a repository rather than running the time consuming analysis.

```
trend_dat <- response_levels %>%
  mutate(data = map2(prefix, categories, ~preprocess(.x,.y)),
         n_cats = map_dbl(categories, length),
         mod = map2(data, n_cats, fit_gam),
         trends = map(mod, extract_trends))
```

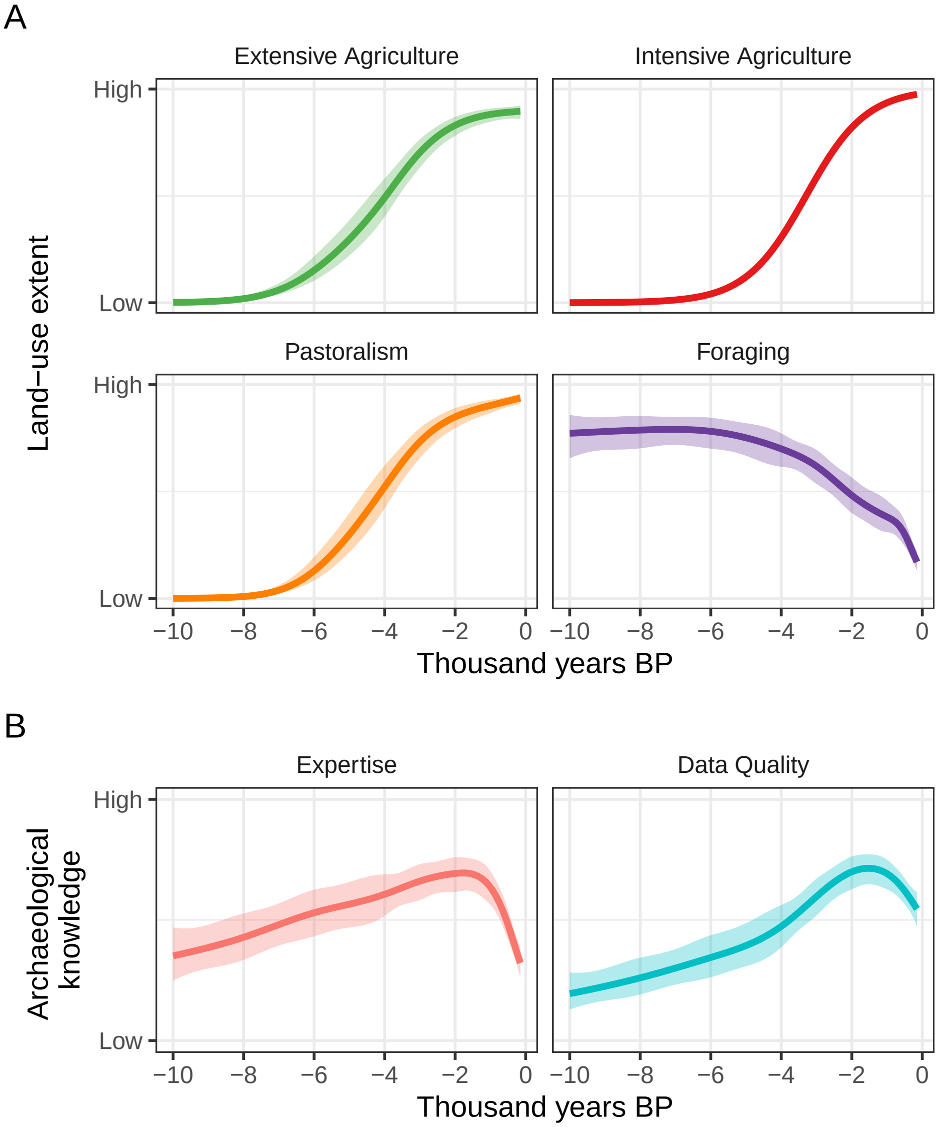
Results

First we plot out the global trends for each land use type, and compare them to the consensus estimates. Then we plot the local (regional trends) for all land use types, and map out their associated clusters. Please refer to the .rmd source file for the code to make the plots.

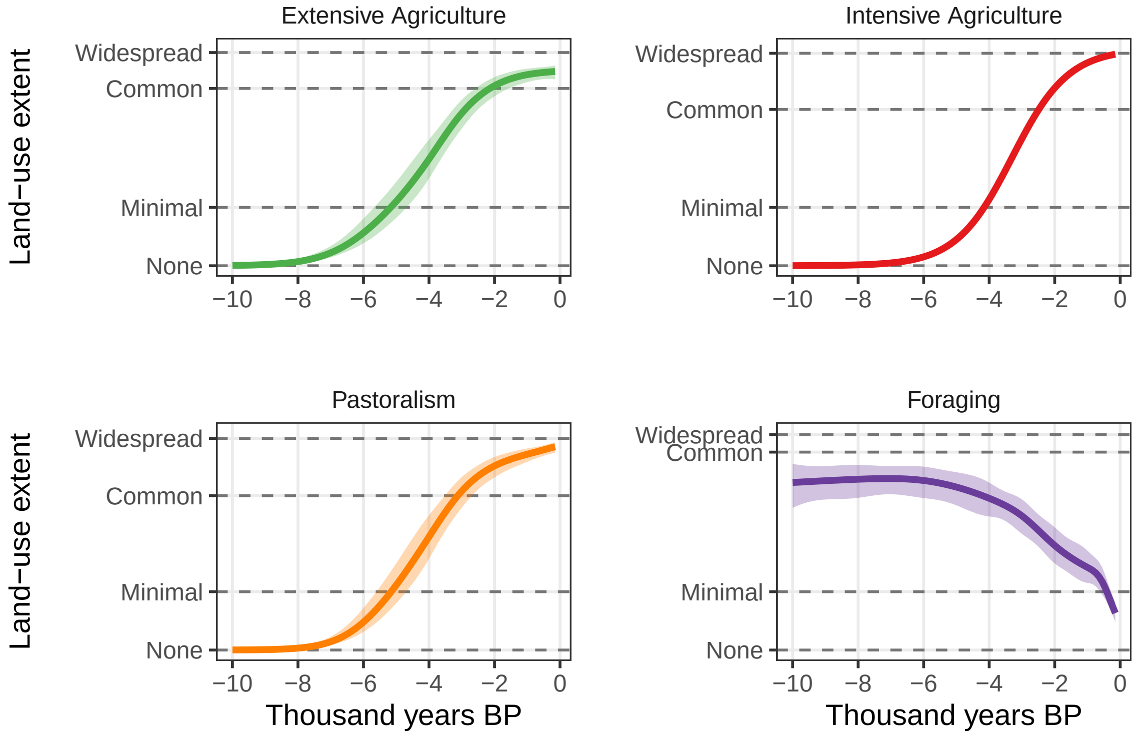
Global Trends

The global trend in foraging shows constant high prevalence until around 6,000 years ago, after which there is a smooth decline until the present day when it is very rare. Mapping out the clusters reveals a clear east-west divide, which regions in Afro-eurasia seeing foraging earlier then the global mean, and regions in the Americas and Oceania seeing later peaks in foraging.

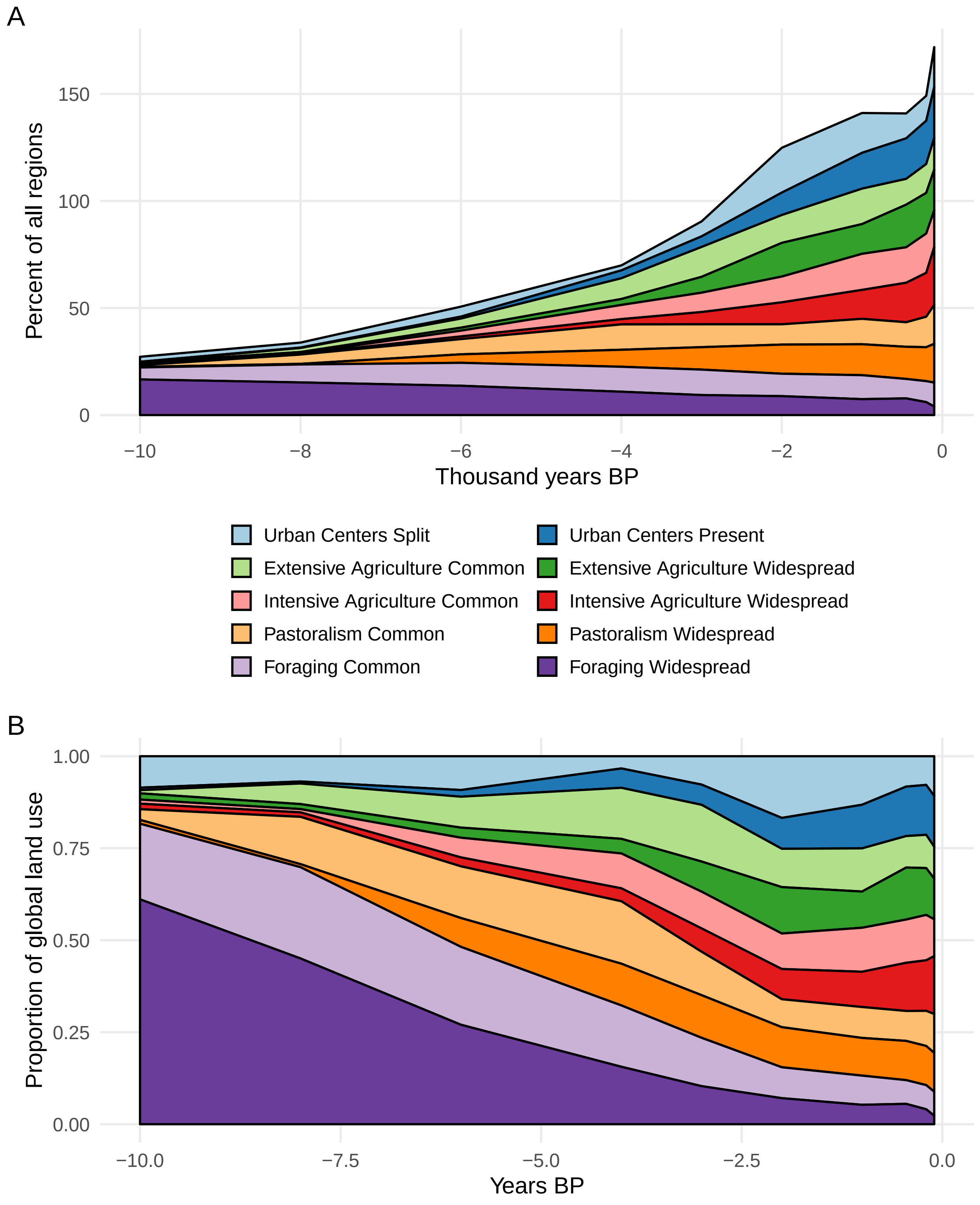
The global trends in the prevalence of pastoralism, extensive and intensive agriculture, and urbanism all follow a sigmoidal curve, which means the trend is linear on the scale of the linear predictor (the ordered categorical GAM uses a logit transform as a latent link function). This means that there is a simple increase in the probability of each land use type being prevalent over time.



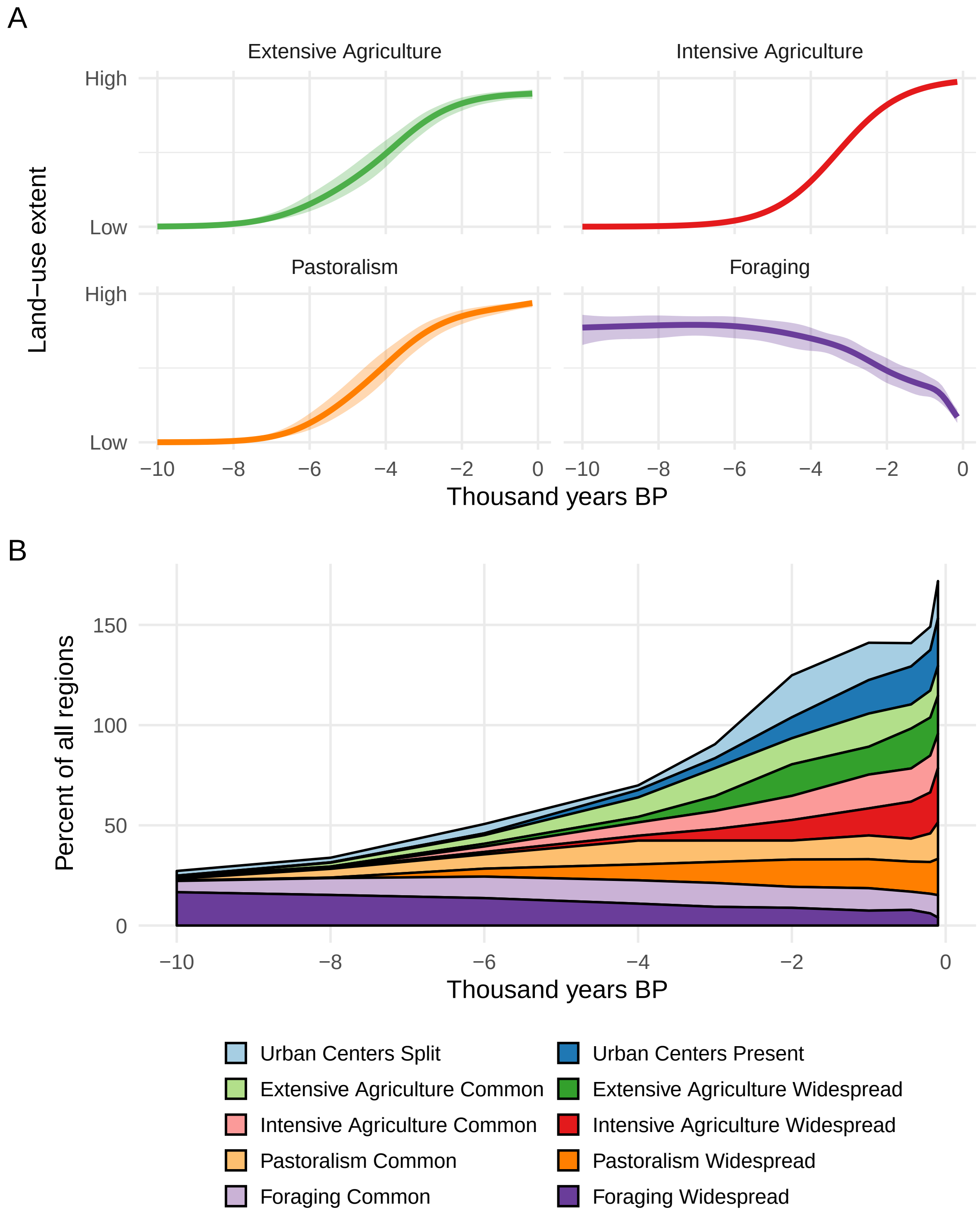
The numerical cutpoints between the ordered categorical response levels estimated by the model vary across land-use type. This is a normal result of the ordered categorical regression, and basically means that different sources of error/uncertainty impact how contributors translate their mental models of areal extent (the latent, “real” value the regression is trying to estimate) into discrete categories across the different land use types.



Compare the global trends to the consensus assessments.

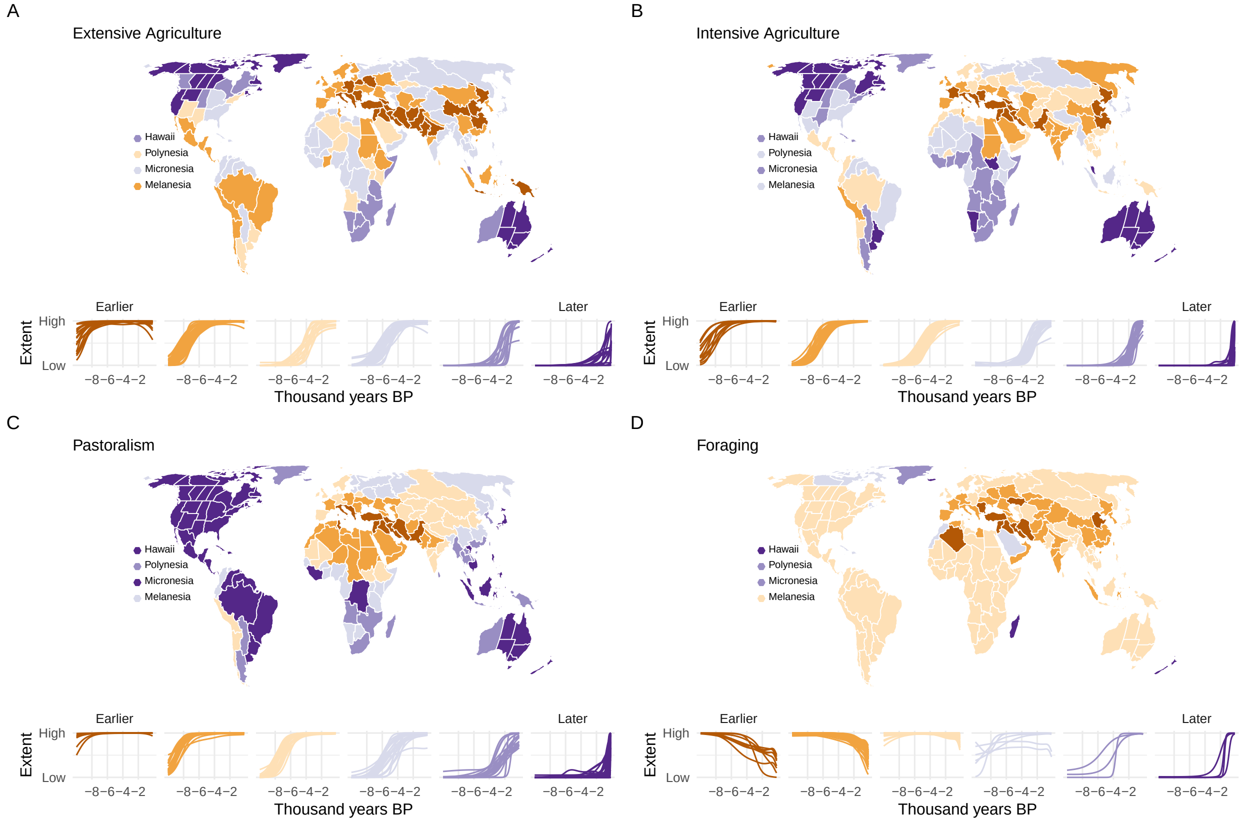


```
(g1 + theme_minimal()) / cs1 + plot_annotation(tag_levels = 'A')
```



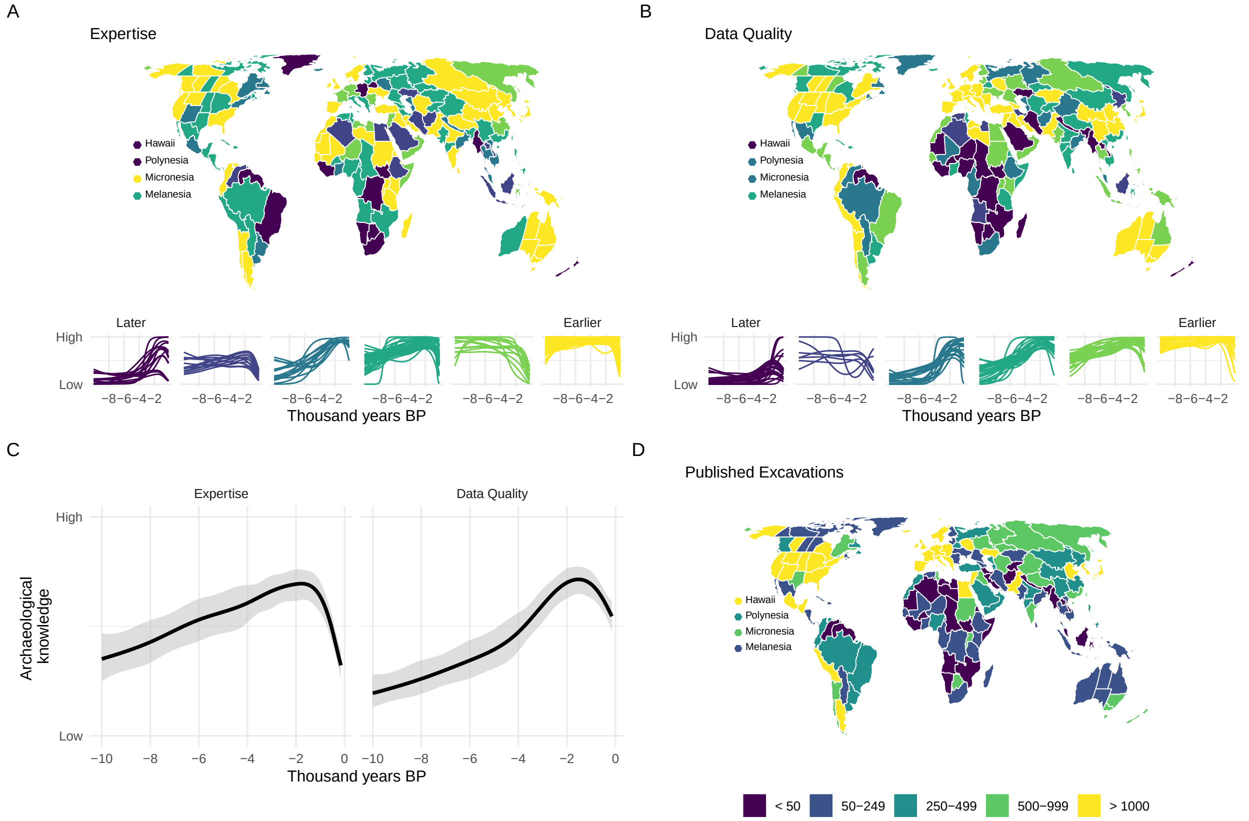
```
ggsave('figures/3_trends_global.png', height = 7.5, width = 6.5)
```

Regional Trends



How does self-professed level of expertise vary in each region over time? The global trend is a roughly linear increase in self-reported expertise from 10ka BP up to 2ka BP, then a falloff continuing to the present day. The present day expertise values are approximately the same as at 10ka BP. This makes sense, as it points to both the increased frequency of preserved archaeological materials with time as well as the reduction in archaeological attention in periods with extensive historical records.

The global trend in data quality is more or less the same as the expertise data, with the peak in data quality occurring more recently than for expertise and with a less dramatic falloff leading to the present day. Unlike expertise, which reaches the same values at 10ky BP and present, data quality in the present day remains high in spite of the falloff in the last 2 millennia. Also note the confidence interval for the global trend is generally wider than for the expertise responses.



Was the abandonment of widespread foraging more correlated closely with the spread of pastoralism than crop agriculture?

We generated a structural equation model to estimate parameters for a path schematic to test hypotheses about causal relationships between latent variables in the consensus land use data. Structural equation modeling is a multivariate statistical method for analyzing structural relationships that include latent variables (Bollen 1989, Beaujean 2014). The consensus data were recoded to ordinal factors and then input to a diagonally weighted least squares procedure to estimate the structural equation model parameters. To investigate weather the abandonment of widespread foraging was more correlated closely with the spread of pastoralism than crop agriculture, we modeled the consensus responses for foraging, pastoralism and crop agriculture for all regions during the middle and late Holocene. We used the lavaan R package (Rosseel 2012) to fit a model with a Model Fit Test Statistic of 75.199, 18 degrees of freedom. The model fit is good, as indicated by a Comparative Fit Index (CFI) 0.997 and a Root Mean Square Error of Approximation (RMSEA) of 0.148. The model output shows a regression estimate of -0.674 for foraging and pastoralism, compared to -0.574 for foraging and crop agriculture. The regression coefficient of foraging predicting pastoralism is more negative than foraging predicting crop agriculture, indicating that as foraging was abandoned it was more often replaced by pastoralism than crop agriculture.

```
consensus_cat <-
  consensus %>%
  # convert consensus variables to ordinal factors
  mutate_at(.vars = vars(FHG_10KBP:URBAN_1850CE),
            .funs = funs(case_when(. == "Widespread" ~ 3,
                                   . == "Common" ~ 2,
                                   . == "Minimal" ~  1,
                                   . == "None" ~ 0))) %>%
  mutate_at(.vars = vars(FHG_10KBP:URBAN_1850CE),
            .funs = funs(factor(., ordered = TRUE)))

mod.sem1 <- '
# latent variable definitions
hunt  =~ 0 * FHG_6KBP  + 1 *  FHG_4KBP  + 2 * FHG_3KBP
past  =~ 0 * PAS_6KBP  + 1 * PAS_4KBP   + 2 * PAS_3KBP
crop  =~ 0 * INAG_6KBP + 1 * INAG_4KBP  + 2 * INAG_3KBP
# regressions
past ~ hunt
crop ~ hunt
# residual correlations
FHG_6KBP ~~ PAS_6KBP + INAG_6KBP
FHG_4KBP ~~ PAS_4KBP + INAG_4KBP
FHG_3KBP ~~ PAS_3KBP + INAG_3KBP
'
```

```
library(lavaan)
sem.fit <- sem(mod.sem1,
               data = consensus_cat[,-c(1:2)])
# inspect the summary
summary(sem.fit,
        fit.measures=TRUE,
        standardized = TRUE)
```

```
## lavaan 0.6-3 ended normally after 26 iterations
##
##   Optimization method                           NLMINB
##   Number of free parameters                         39
##
##   Number of observations                           146
##
##   Estimator                                       DWLS      Robust
##   Model Fit Test Statistic                   10759.405    6669.361
##   Degrees of freedom                                24          24
##   P-value (Chi-square)                           0.000       0.000
##   Scaling correction factor                                  1.616
##   Shift parameter                                           11.028
##     for simple second-order correction (Mplus variant)
##
## Model test baseline model:
##
##   Minimum Function Test Statistic            21562.343    9594.431
##   Degrees of freedom                                36          36
##   P-value                                        0.000       0.000
##
## User model versus baseline model:
##
##   Comparative Fit Index (CFI)                    0.501       0.305
##   Tucker-Lewis Index (TLI)                       0.252      -0.043
##
##   Robust Comparative Fit Index (CFI)                            NA
##   Robust Tucker-Lewis Index (TLI)                               NA
```

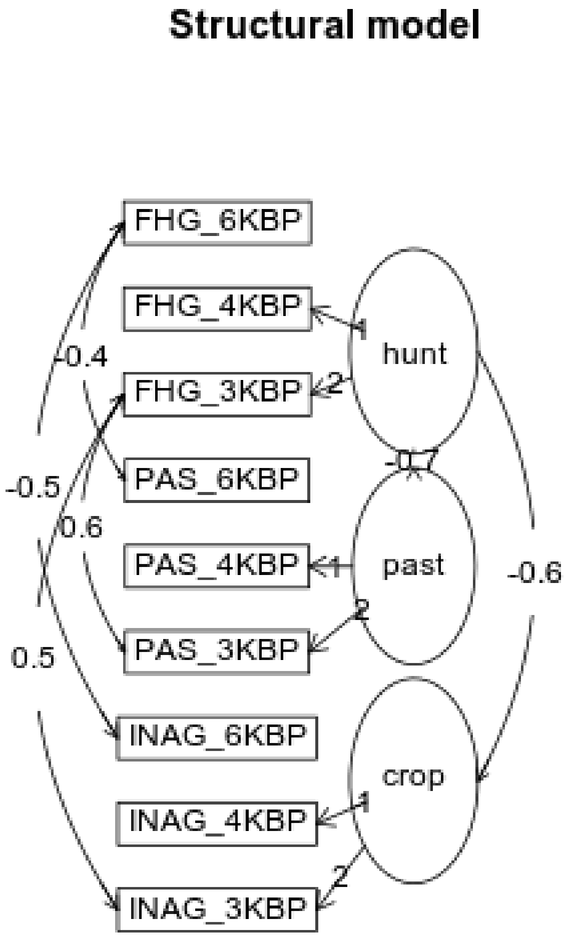
```
##
## Root Mean Square Error of Approximation:
##
##   RMSEA                                          1.756       1.382
##   90 Percent Confidence Interval          1.729  1.784       1.354  1.410
##   P-value RMSEA <= 0.05                          0.000       0.000
##
##   Robust RMSEA                                                  NA
##   90 Percent Confidence Interval                                NA     NA
##
## Standardized Root Mean Square Residual:
##
##   SRMR                                           0.456       0.456
##
## Parameter Estimates:
##
##   Information                                 Expected
##   Information saturated (h1) model        Unstructured
##   Standard Errors                           Robust.sem
##
## Latent Variables:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   hunt =~
##     FHG_6KBP          0.000                               0.000    0.000
##     FHG_4KBP          1.000                               0.668    0.668
##     FHG_3KBP          2.000                               1.336    1.336
##   past =~
##     PAS_6KBP          0.000                               0.000    0.000
##     PAS_4KBP          1.000                               0.687    0.687
##     PAS_3KBP          2.000                               1.374    1.374
##   crop =~
##     INAG_6KBP         0.000                               0.000    0.000
##     INAG_4KBP         1.000                               0.694    0.694
##     INAG_3KBP         2.000                               1.389    1.389
##
## Regressions:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   past ~
##     hunt             -0.688    0.046  -15.063    0.000   -0.669   -0.669
##   crop ~
##     hunt             -0.594    0.055  -10.793    0.000   -0.572   -0.572
##
## Covariances:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##  .FHG_6KBP ~~
##    .PAS_6KBP         -0.353    0.084   -4.225    0.000   -0.353   -0.353
##    .INAG_6KBP        -0.537    0.092   -5.819    0.000   -0.537   -0.537
##  .FHG_4KBP ~~
##    .PAS_4KBP         -0.228    0.037   -6.148    0.000   -0.228   -0.422
##    .INAG_4KBP        -0.242    0.045   -5.354    0.000   -0.242   -0.451
##  .FHG_3KBP ~~
##    .PAS_3KBP          0.628    0.045   13.974    0.000    0.628    0.752
##    .INAG_3KBP         0.528    0.068    7.802    0.000    0.528    0.618
##  .past ~~
##    .crop               0.086    0.027    3.191    0.001    0.296    0.296
##
## Intercepts:
##                     Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##    .FHG_6KBP           0.000                                0.000    0.000
##    .FHG_4KBP           0.000                                0.000    0.000
##    .FHG_3KBP           0.000                                0.000    0.000
##    .PAS_6KBP           0.000                                0.000    0.000
##    .PAS_4KBP           0.000                                0.000    0.000
##    .PAS_3KBP           0.000                                0.000    0.000
##    .INAG_6KBP          0.000                                0.000    0.000
##    .INAG_4KBP          0.000                                0.000    0.000
##    .INAG_3KBP          0.000                                0.000    0.000
##     hunt               0.000                                0.000    0.000
##    .past               0.000                                0.000    0.000
##    .crop               0.000                                0.000    0.000
##
## Thresholds:
##                     Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##     FHG_6KBP|t1       -1.541    0.164   -9.388    0.000   -1.541   -1.541
##     FHG_6KBP|t2       -1.063    0.129   -8.271    0.000   -1.063   -1.063
##     FHG_6KBP|t3       -0.103    0.104   -0.990    0.322   -0.103   -0.103
##     FHG_4KBP|t1       -1.738    0.187   -9.287    0.000   -1.738   -1.738
##     FHG_4KBP|t2       -0.752    0.116   -6.510    0.000   -0.752   -0.752
##     FHG_4KBP|t3        0.173    0.105    1.649    0.099    0.173    0.173
##     FHG_3KBP|t1       -1.822    0.199   -9.155    0.000   -1.822   -1.822
##     FHG_3KBP|t2       -0.580    0.111   -5.243    0.000   -0.580   -0.580
##     FHG_3KBP|t3        0.332    0.106    3.128    0.002    0.332    0.332
##     PAS_6KBP|t1        0.369    0.107    3.456    0.001    0.369    0.369
##     PAS_6KBP|t2        0.775    0.116    6.665    0.000    0.775    0.775
##     PAS_6KBP|t3        1.305    0.144    9.085    0.000    1.305    1.305
##     PAS_4KBP|t1        0.000    0.104    0.000    1.000    0.000    0.000
##     PAS_4KBP|t2        0.260    0.105    2.472    0.013    0.260    0.260
##     PAS_4KBP|t3        0.871    0.120    7.274    0.000    0.871    0.871
##     PAS_3KBP|t1       -0.086    0.104   -0.825    0.410   -0.086   -0.086
##     PAS_3KBP|t2        0.155    0.105    1.484    0.138    0.155    0.155
##     PAS_3KBP|t3        0.664    0.113    5.881    0.000    0.664    0.664
##     INAG_6KBP|t1       1.005    0.126    7.998    0.000    1.005    1.005
##     INAG_6KBP|t2       1.390    0.150    9.250    0.000    1.390    1.390
##     INAG_6KBP|t3       2.043    0.238    8.589    0.000    2.043    2.043
##     INAG_4KBP|t1       0.406    0.107    3.783    0.000    0.406    0.406
##     INAG_4KBP|t2       0.871    0.120    7.274    0.000    0.871    0.871
##     INAG_4KBP|t3       1.738    0.187    9.287    0.000    1.738    1.738
##     INAG_3KBP|t1       0.120    0.104    1.154    0.248    0.120    0.120
##     INAG_3KBP|t2       0.520    0.109    4.759    0.000    0.520    0.520
##     INAG_3KBP|t3       1.305    0.144    9.085    0.000    1.305    1.305
##
## Variances:
##                     Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##    .FHG_6KBP           1.000                                1.000    1.000
##    .FHG_4KBP           0.554                                0.554    0.554
##    .FHG_3KBP          -0.786                               -0.786   -0.786
##    .PAS_6KBP           1.000                                1.000    1.000
##    .PAS_4KBP           0.528                                0.528    0.528
##    .PAS_3KBP         -0.887                              -0.887   -0.887
##    .INAG_6KBP         1.000                               1.000    1.000
##    .INAG_4KBP         0.518                               0.518    0.518
##    .INAG_3KBP        -0.928                              -0.928   -0.928
##     hunt              0.446    0.013   35.535    0.000    1.000    1.000
##    .past              0.260    0.027    9.648    0.000    0.552    0.552
##    .crop              0.325    0.031   10.460    0.000    0.673    0.673
##
## Scales y*:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##     FHG_6KBP          1.000                               1.000    1.000
##     FHG_4KBP          1.000                               1.000    1.000
##     FHG_3KBP          1.000                               1.000    1.000
##     PAS_6KBP          1.000                               1.000    1.000
##     PAS_4KBP          1.000                               1.000    1.000
##     PAS_3KBP          1.000                               1.000    1.000
##     INAG_6KBP         1.000                               1.000    1.000
##     INAG_4KBP         1.000                               1.000    1.000
##     INAG_3KBP         1.000                               1.000    1.000
```

```
# modificationindices(sem.fit, sort = TRUE)

# look at the diagram:
png("figures/lavaan.diagram.png")
psych::lavaan.diagram(sem.fit)
dev.off()
```

```
## pdf
##   2
```

```
knitr::include_graphics("figures/lavaan.diagram.png")
```



Structural equation modelling shows that a causal inference of abandonment of widespread foraging correlating more closely with the spread of pastoralism than crop agriculture is consistent with the data.

We can also investigate this using an odds ratio approach. We can create a table of counts of regions that show a decline in foraging over time, and counts of regions where pastoralism is more widespread than intensive agriculture at an arbitrary time point, such as 2 k BP. We can then compute an odds ratio for this table, and if the result is greater than one, we can conclude that the outcome of pastoralism more widespread than crop agriculture after widespread foraging is abandoned is more likely that the alternative outcome.

```
# odds ratio approach
consensus_cat_df <-
consensus_cat %>%
  # label those regions that show a decline in foraging over time
  mutate(shows_decline_in_foraging =  as.numeric((FHG_10KBP > FHG_2KBP)) ) %>%
  # label those regions that show pastoralism more widespread than crop agriculture
  mutate(shows_more_pastoralism_than_crop =   as.numeric(PAS_2KBP > INAG_2KBP )) %>%
  # check
  select( shows_decline_in_foraging,
          shows_more_pastoralism_than_crop) %>%
  group_by(shows_decline_in_foraging,
           shows_more_pastoralism_than_crop) %>%
  tally() %>%
  spread(shows_more_pastoralism_than_crop, n, fill = 0) %>%
  arrange(desc(shows_decline_in_foraging)) %>%
  select(shows_decline_in_foraging, `1`, `0`)

# show a table
consensus_cat_df_show <- consensus_cat_df
names(consensus_cat_df_show) <- c(" ",
                                  "pastoralism more widespread than crops",
                                  "pastoralism less widespread than crop")
consensus_cat_df_show$` ` <- c('shows a decline in foraging over time',
                              'shows no decline in foraging over time')
knitr::kable(consensus_cat_df_show)
```

	pastoralism more widespread than crops	pastoralism less widespread than crop
shows a decline in foraging over time	28	39
shows no decline in foraging over time	19	60

```
# get odds ratio and p-value
tab <- as.matrix(consensus_cat_df[,2:3])
ft <- fisher.test(tab)

# another way
d <- data.frame(g=factor(1:2),
                s=tab[c(1,3)],
                f=tab[c(2,4)])
g <- glm(s/(s+f) ~ g,
         weights = s + f,
         data = d,
         family="binomial")
# coef(summary(g))["g2",c("Estimate","Pr(>|z|)")]
# To get the likelihood ratio test (slightly more accurate
# than the Wald -value shown above), do
lrt <- anova(g,test="Chisq")
p_value <- round(lrt $`Pr(>Chi)`[2], 3)

# odds ratio, check it by hand
A <- tab[1]
B <- tab[3]
C <- tab[2]
D <- tab[4]
or <- (A/B) / (C/D)
```

The odds ratio for this table is 2.267, with a p-value of 0.022. This indicates that that claim of pastoralism being more widespread than crop agriculture after widespread foraging is abandoned is supported by the data.

Bollen, K. A. (1989). Structural Equations with Latent Variables (Wiley Series in Probability and Statistics, Canada

Beaujean, A. A. (2014). Latent variable modeling using R: A step-by-step guide. Routledge.

Rosseel, Y. (2012). Lavaan: An R package for structural equation modeling and more. Version 0.5–12 (BETA). Journal of statistical software, 48(2), 1-36.

HYDE Comparison

Here we compare the onset times for intensive agriculture derived from the ArchaeoGLOBE consensus estimates with those calculated from HYDE 3.2 data. We assess these differences at both the Common (>= 1%) and Widespread (>= 20%) levels.

First calculate the total cropland area from HYDE in each of the ArchaeoGLOBE regions, and convert to proportion of land area under cultivation in order to compare HYDE and ArchaeoGLOBE. Be warned the `raster::extract()` command will take a while to run.

```r
hyde_crop_prop <- hyde %>%
  raster::extract(regions_hyde, na.rm = TRUE, fun = sum, df = TRUE) %>% # sample at region locations
  `names<-`(c('ID', -10, -8, -6, -4, -3, -2, -1, -0.45, -0.2, -0.1, 0.05)) %>%
  gather(time, value, 2:12) %>%
  mutate(time = as.numeric(time)) %>%
  left_join(regions, by = c('ID' = 'Archaeo_ID')) %>%
  mutate(prop = value / Land_Area)
```

Calculate the earliest onset time for intensive agriculture at common and widespread thresholds for the HYDE data.

```r
hyde_onset <- hyde_crop_prop %>%
  filter(prop >= 0.01) %>%
  mutate(level = if_else(prop >= 0.2, 'Widespread', 'Common')) %>%
  group_by(ID, level) %>%
  summarise(onset = min(time)) %>%
  spread(level, onset) %>%
  mutate(Common = if_else(is.na(Common), Widespread, Common)) %>%
  gather(level, onset, Common:Widespread) %>%
  mutate(source = 'HYDE',
         level = if_else(level == 'Common',
                         'Common (> 1% land area)',
                         'Widespread (> 20% land area)')) %>%
  rename(Region = ID)
```

Repeat for the ArchaeoGLOBE consensus assessment.

```r
archaeoglobe_onset <- consensus %>%
  select(Region, Label, INAG_10KBP:INAG_1850CE) %>%
  gather(time_step, value, INAG_10KBP:INAG_1850CE) %>%
  filter(value %in% c('Common', 'Widespread')) %>%
  mutate(time = parse_number(time_step),
         time = case_when(time == 1500 ~ .45,
                              time == 1750 ~ .2,
                              time == 1850 ~ .1,
                              time <= 10 ~ time),
         time = time * -1) %>%
  group_by(Region, value) %>%
  summarise(onset = min(time)) %>%
  spread(value, onset) %>%
  mutate(Common = if_else(is.na(Common), Widespread, Common)) %>%
  gather(level, onset, Common:Widespread) %>%
  mutate(source = 'ArchaeoGLOBE',
         level = if_else(level == 'Common',
                         'Common (> 1% land area)',
                         'Widespread (> 20% land area)'))
```

Combine into a single data frame.

```
onsets <- bind_rows(hyde_onset, archaeoglobe_onset)
```

What are the differences in onset times between the two datasets?

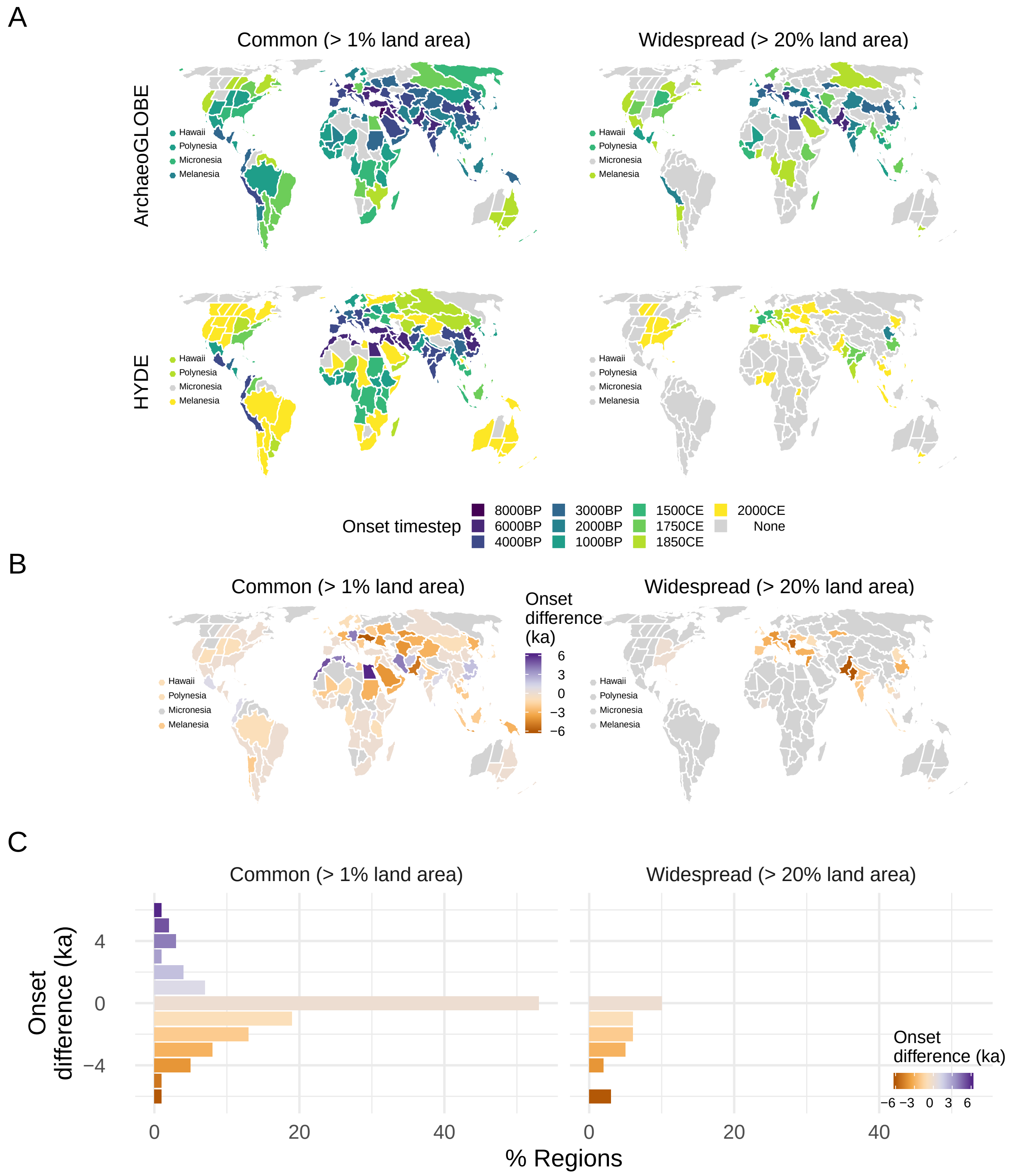
```
# world regions that have crops in HYDE at 2000CE
hyde_ag_regions <- hyde_crop_prop %>%
  filter(time == 0.05 & prop >= 0.01) %>%
  pull(ID)

onset_difference <- onsets %>%
  spread(source, onset) %>%
  filter(Region %in% hyde_ag_regions) %>%
  mutate(diff = ArchaeoGLOBE - HYDE,
         diff = round(diff))
```

Plot the results.

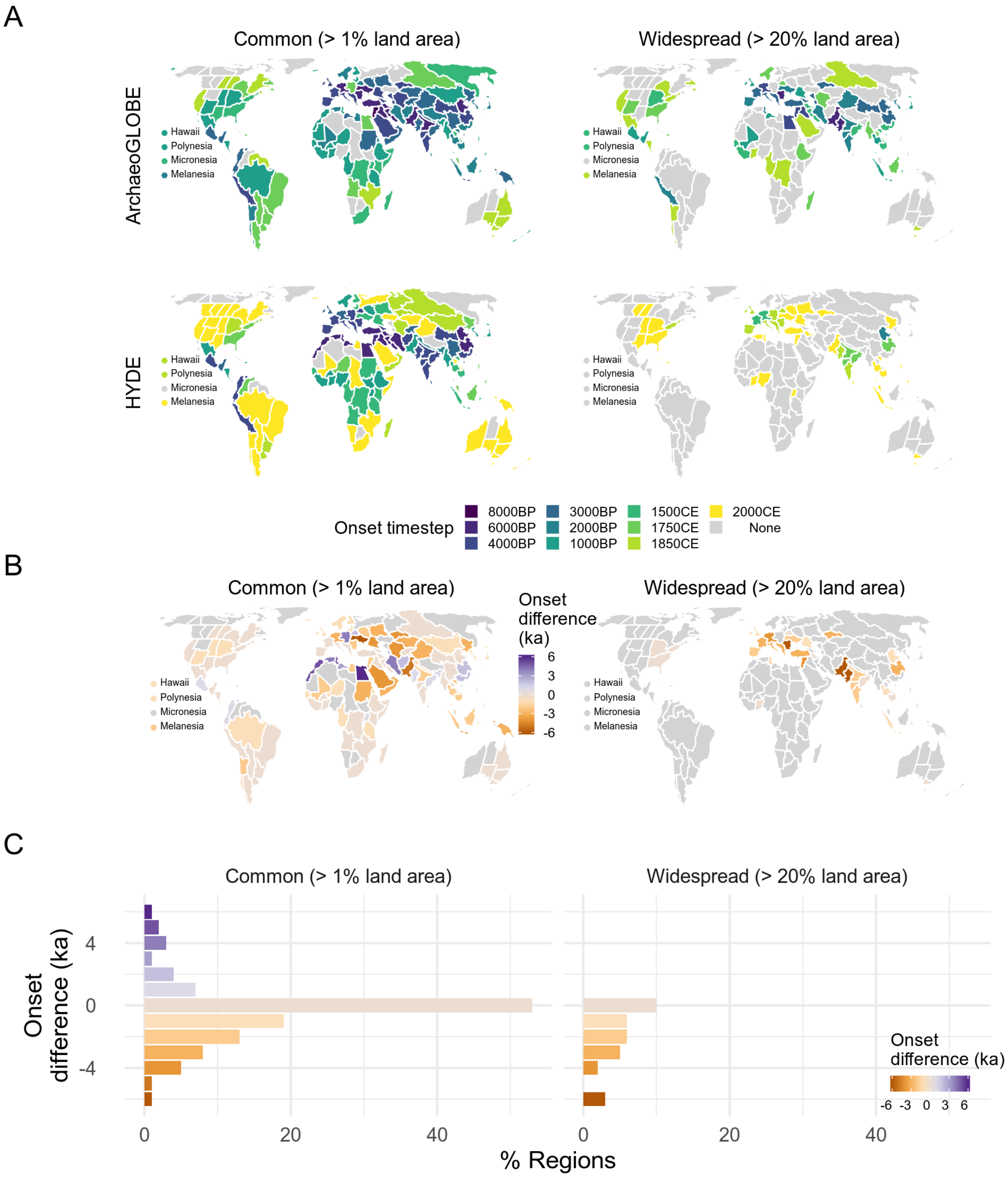
```
# put the figs together for the ArchaeoGLOBE-HYDE comparison
# looks good when run interactively, but not when knit

(onset_maps / onset_diff_maps / onset_diff_barplot) +
  plot_layout(heights = c(2, 1, 1)) +
  plot_annotation(tag_levels = "A")
```



```r
ggsave("figures/5_ArchaeoGLOBE_HYDE_comparison.png", width = 6.5, height = 7.5)

# doesn't look good when knit
knitr::include_graphics("figures/5_ArchaeoGLOBE_HYDE_comparison.png")
```


A
Common (> 1% land area)
Widespread (> 20% land area)
ArchaeoGLOBE
HYDE
Hawaii
Polynesia
Micronesia
Melanesia
Onset timestep
8000BP
6000BP
4000BP
3000BP
2000BP
1000BP
1500CE
1750CE
1850CE
2000CE
None
B
Common (> 1% land area)
Widespread (> 20% land area)
Onset difference (ka)
6
3
0
-3
-6
C
Common (> 1% land area)
Widespread (> 20% land area)
Onset difference (ka)
4
0
-4
0
20
40
% Regions
-6 -3 0 3 6