Chao Phraya Monthly Streamflow Reconstruction

Hung Nguyen

2022-08-23

Introduction and Preparations

This document details the process of producing the results presented in *Droughts, Pluvials, and Wet Season Timing across the Chao Phraya River Basin: a 254-Year Monthly Reconstruction from Tree Rings and \(\delta^{18}O\)* by Nguyen et al. (2022).

To reproduce the results, please do the following:

- This code requires R 4.1.0 and above.
- Download the code repository from the GitHub repo and extract the downloaded `.zip` file to your working folder.
- Open `chao-phraya-monthly.Rproj` in RStudio (**It's important to open this first so that the file path is loaded properly**).
- Install and load the following packages if you don't already have them:

```
library(mbr)        # Mass balance regression - this is the key package
library(dplR)       # Tree ring data processing
library(ldsr)       # Tree ring data processing
library(ggplot2)    # Plotting
library(cowplot)    # Plotting
library(patchwork)  # Plotting
library(ggtext)     # Plotting
library(ggprism)    # Plotting
library(data.table) # Data handling
library(doFuture)   # Parallel computing
library(glue)       # String handling
library(GA)         # Genetic algorithm
library(memoise)    # To cache intermediate results in GA runs
library(SPEI)       # Calculating drought index
```

- Open `paper-code.Rmd`, which is the source code for this document.
- Follow the written details below and run the code chunks one by one.
- The main reconstruction step with `mbr` should be run on a computing clusters. It still works on a desktop or laptop, but will take much longer, so you may need to change the genetic algorithm settings (more explanations later).

For quick access to the final results please see the `.csv` files in `results/`. This folder is organized by the tributaries.

The code utilities to support the main code are stored in the folder `R/`. We need to load them first before running the main code.

```
source('R/init.R')
source('R/correlation_functions.R')
source('R/input_selection_functions.R')
source('R/drought_analysis_functions.R')
```

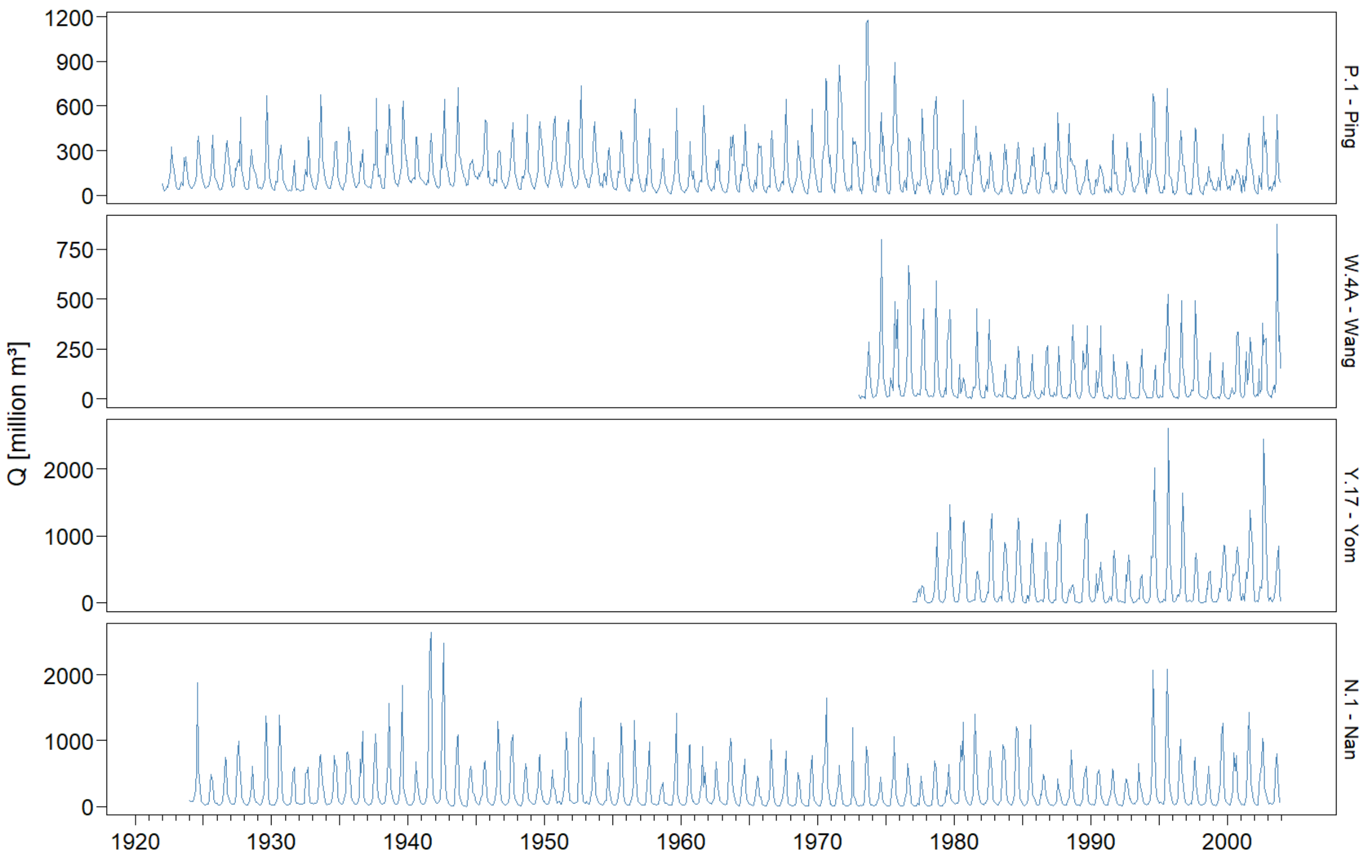
Data

Streamflow Data

```
# Time span of each station
p1years  <- 1922:2003
w4years  <- 1973:2003
y17years <- 1977:2003
n1years  <- 1924:2003
# For P.1, we need to combine the post-1986 naturalized flow with the pre-1986 flow.
# 1986 is the year of dam construction.
p1raw  <- fread('data/streamflow/P1-monthly-raw.csv')
p1nat  <- fread('data/streamflow/P1-monthly-naturalized.csv')
p1m     <- rbind(p1raw[year < 1986], p1nat[year >= 1986])[year %in% p1years]
# For the other stations we can read the data directly
w4m  <- fread('data/streamflow/w4a-monthly.csv')[year %in% w4years]
y17m <- fread('data/streamflow/y17-monthly.csv')[year %in% y17years]
n1m  <- fread('data/streamflow/n1-monthly.csv')[year %in% n1years]
# Now we combine them into a single data.table
stations <- c('p1', 'w4', 'y17', 'n1')
inst <- rbindlist(
  list(p1 = p1m, w4 = w4m, y17 = y17m, n1 = n1m),
  id = 'station')
inst[, station := factor(station, stations)]
inst[, month2 := factor(month.abb[month], month.abb)]
```

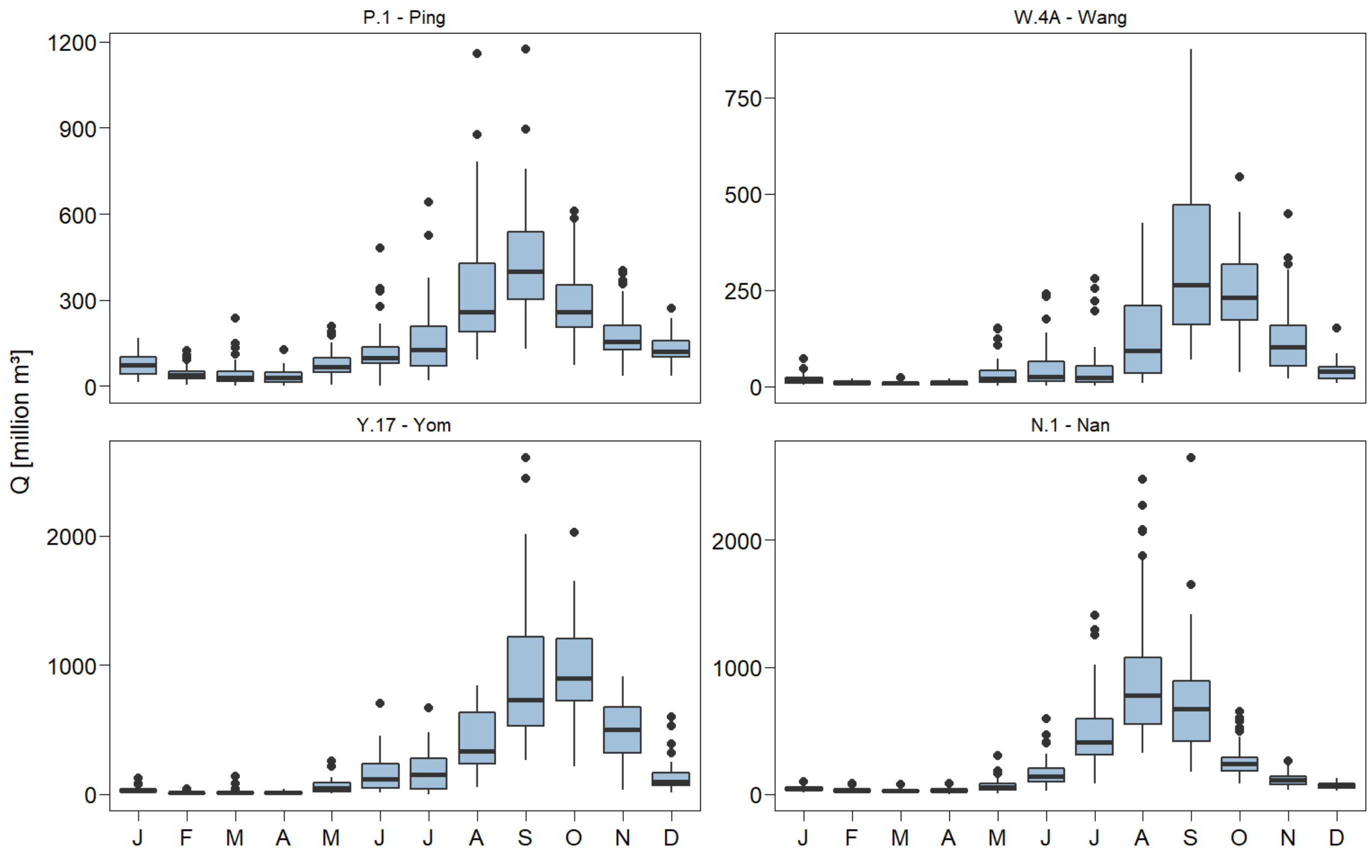
Time series plot of instrumental data

```
ggplot(inst) +
  geom_line(aes(year + (month - 1) / 12, Q), color = 'steelblue') +
  facet_wrap(
    vars(station), ncol = 1,
    strip.position = 'right',
    labeller = as_labeller(stnLab), scales = 'free_y') +
  scale_x_continuous(
    breaks = seq(1920, 2000, 5),
    minor_breaks = 1920:2004,
    labels = skip_label(2),
    guide = guide_prism_minor()) +
  labs(x = NULL, y = 'Q [million m\u00b3]') +
  panel_border('black', 0.2)
```



Monthly streamflow distribution

```
ggplot(inst) +
  geom_boxplot(aes(month2, Q), fill = 'steelblue', alpha = 0.5, outlier.alpha = 1) +
  facet_wrap(vars(station), scales = 'free_y', labeller = as_labeller(stnLab)) +
  scale_x_discrete(labels = monthLabShort) +
  labs(x = NULL, y = 'Q [million m\u00b3]') +
  panel_border('black', 0.2)
```



Now we prepare the reconstruction targets, which are the 12 months plus the annual flow time series.

```
seasons <- c(month.abb, 'Ann') # This is useful for making plots and factor levels
names(ssnLab) <- ssnLab <- seasons
p1tar  <- make_target(p1m)
w4tar  <- make_target(w4m)
y17tar <- make_target(y17m)
n1tar  <- make_target(n1m)
```

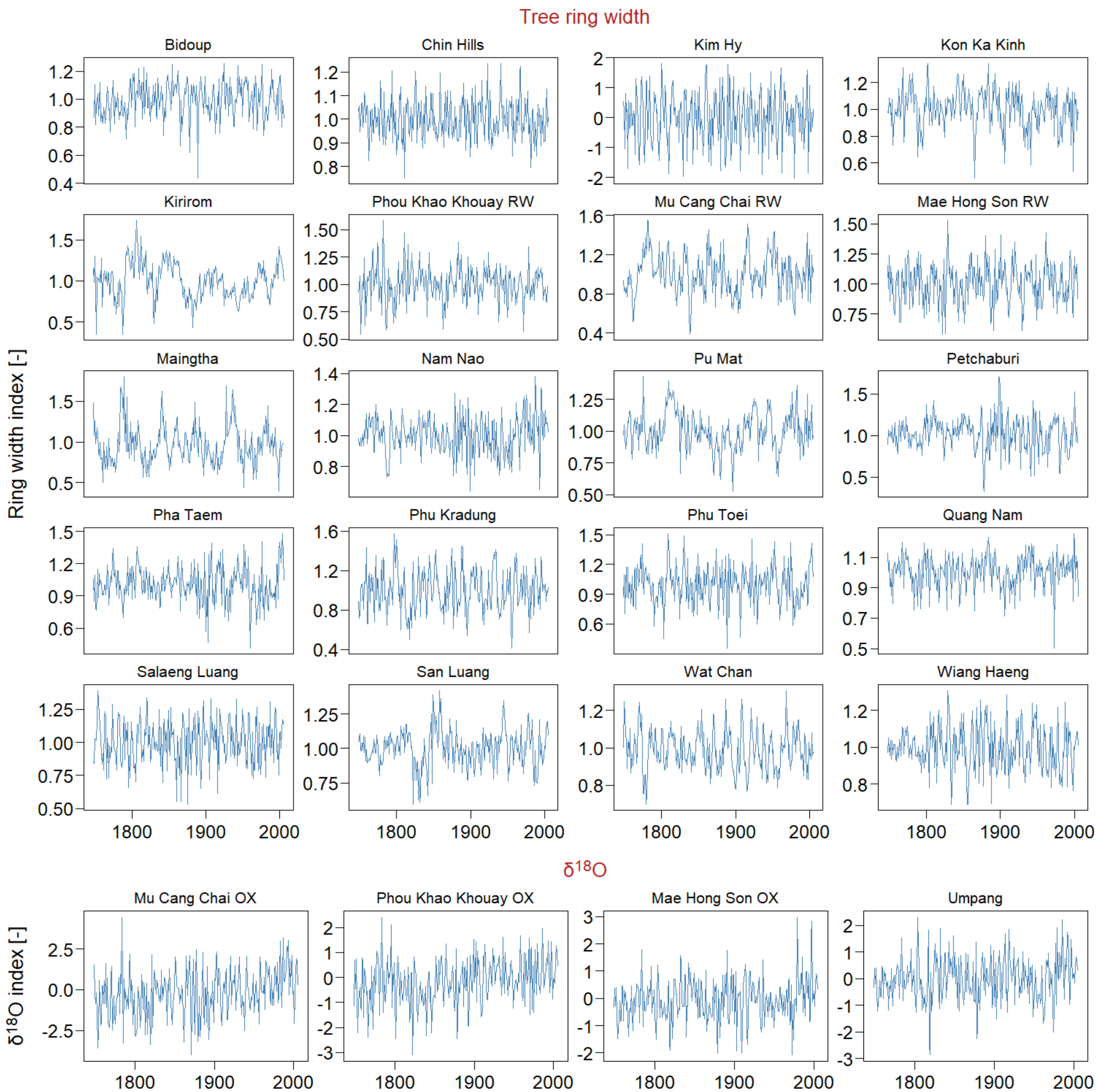
Tree Ring Data

We use the same chronologies that we used in our prior work (Nguyen et al. 2021). As the chronologies have different starting and stopping year, in that earlier paper we imputed the chronologies using the R package `missMDA` (Josse and Husson 2016). Details of the imputation procedure and results have been described in that paper. Here we simply reuse the same imputed chronologies.

```
crn <- fread('data/proxies/crn-filled.csv')
oxi <- fread('data/proxies/oxi-filled.csv')
# Melt data to long format for plotting
crnLong <- melt(crn, id.vars = 'year', variable.name = 'site', value.name = 'rwi')
oxiLong <- melt(oxi, id.vars = 'year', variable.name = 'site', value.name = 'oxi')
# Create matrices for correlation analyses
crnMat <- as.matrix(crn[, -'year'])
oxiMat <- as.matrix(oxi[, -'year'])
```

Time series plots

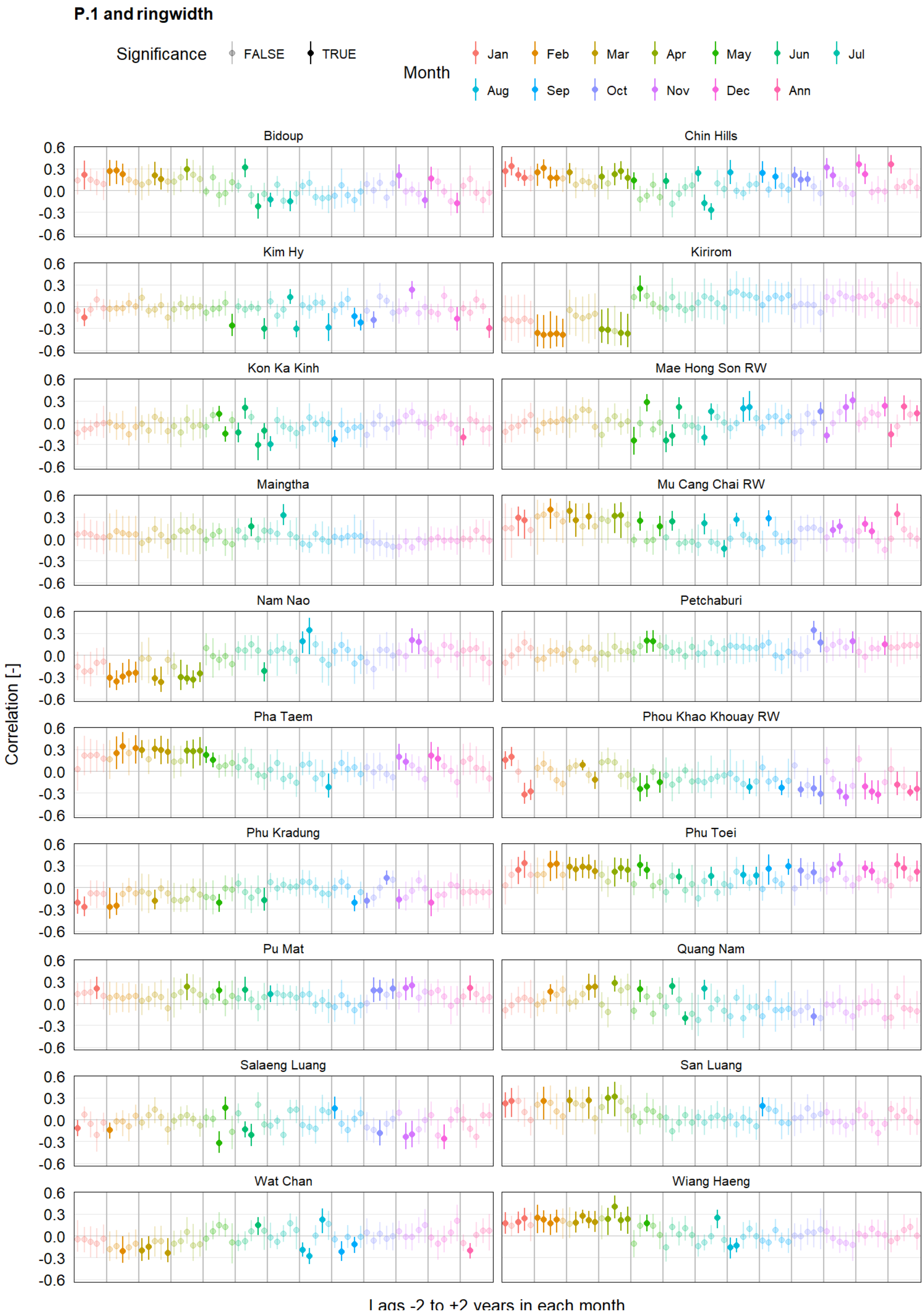
```
p1 <- ggplot(crnLong) +
  geom_line(aes(year, rwi), color = 'steelblue') +
  facet_wrap(vars(site), scales = 'free_y', ncol = 4) +
  panel_border('black', 0.2) +
  labs(x = NULL, y = 'Ring width index [-]',
       subtitle = 'Tree ring width') +
  theme(plot.subtitle = element_text(color = 'firebrick'))

p2 <- ggplot(oxiLong) +
  geom_line(aes(year, oxi), color = 'steelblue') +
  facet_wrap(vars(site), scales = 'free_y', ncol = 4) +
  panel_border('black', 0.2) +
  labs(x = NULL, y = '&delta;<sup>18</sup>O index [-]',
       subtitle = '&delta;<sup>18</sup>O') +
  theme(
    axis.title.y.left = element_markdown(),
    plot.subtitle = element_markdown(color = 'firebrick'))

p1 / p2 +
  plot_layout(heights = c(5, 1))
```



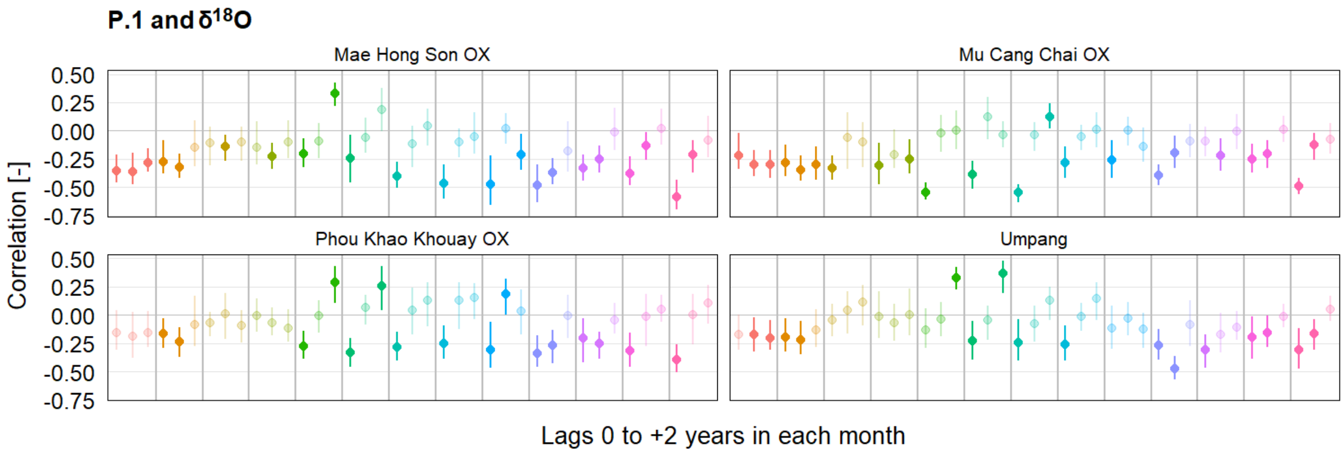
Tree Ring – Streamflow Correlations

P.1

```
rhoP1 <- cor_Q_proxy(p1tar)
plot_cor_station(rhoP1, 'P.1')
```

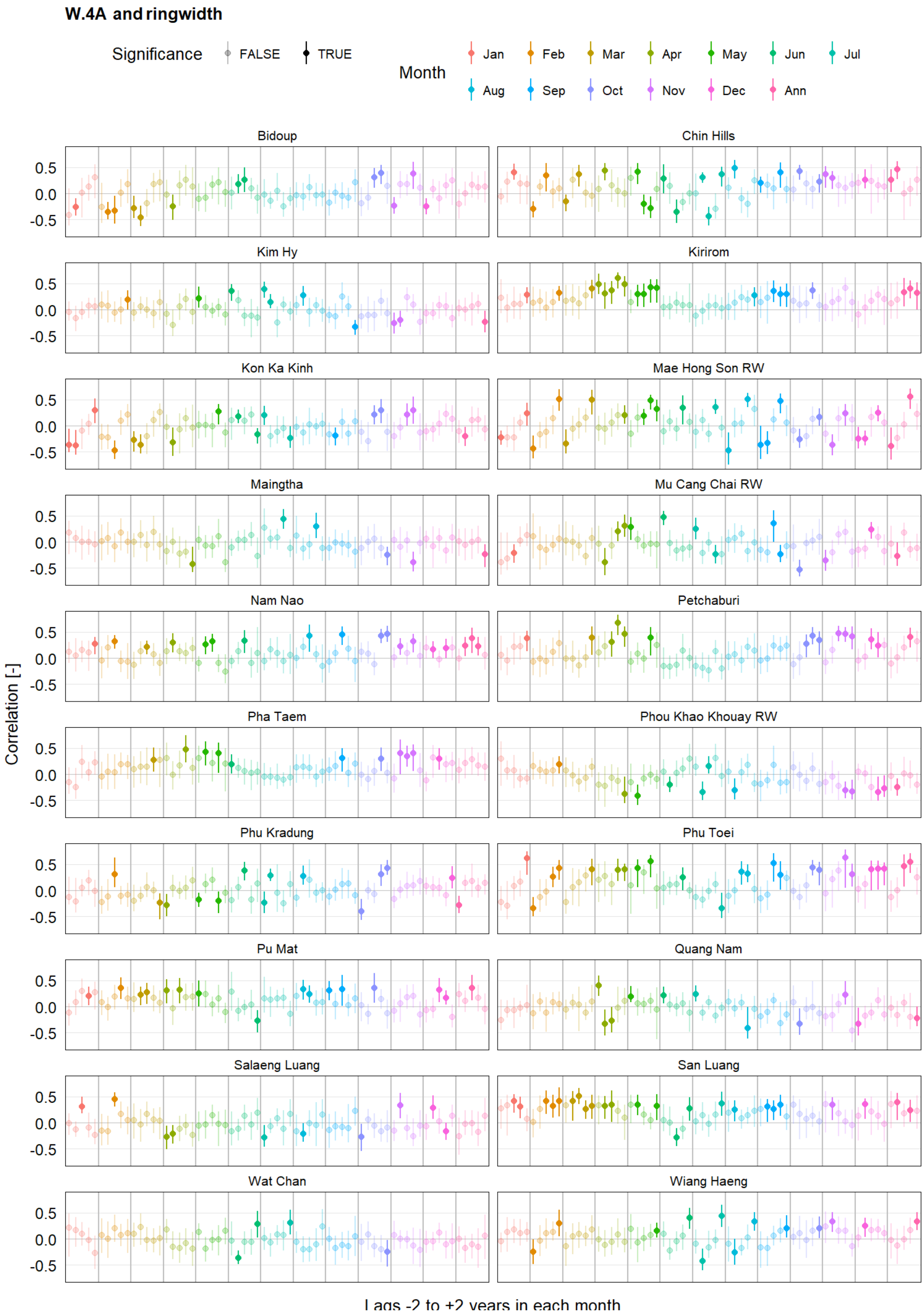


P.1 and ringwidth
Significance
FALSE
TRUE
Month
Jan
Feb
Mar
Apr
May
Jun
Jul
Aug
Sep
Oct
Nov
Dec
Ann
Bidoup
Chin Hills
Kim Hy
Kirirom
Kon Ka Kinh
Mae Hong Son RW
Maingtha
Mu Cang Chai RW
Nam Nao
Petchaburi
Pha Taem
Phou Khao Khouay RW
Phu Kradung
Phu Toei
Pu Mat
Quang Nam
Salaeng Luang
San Luang
Wat Chan
Wiang Haeng
Correlation [-]
0.6
0.3
0.0
-0.3
-0.6
Lags -2 to +2 years in each month

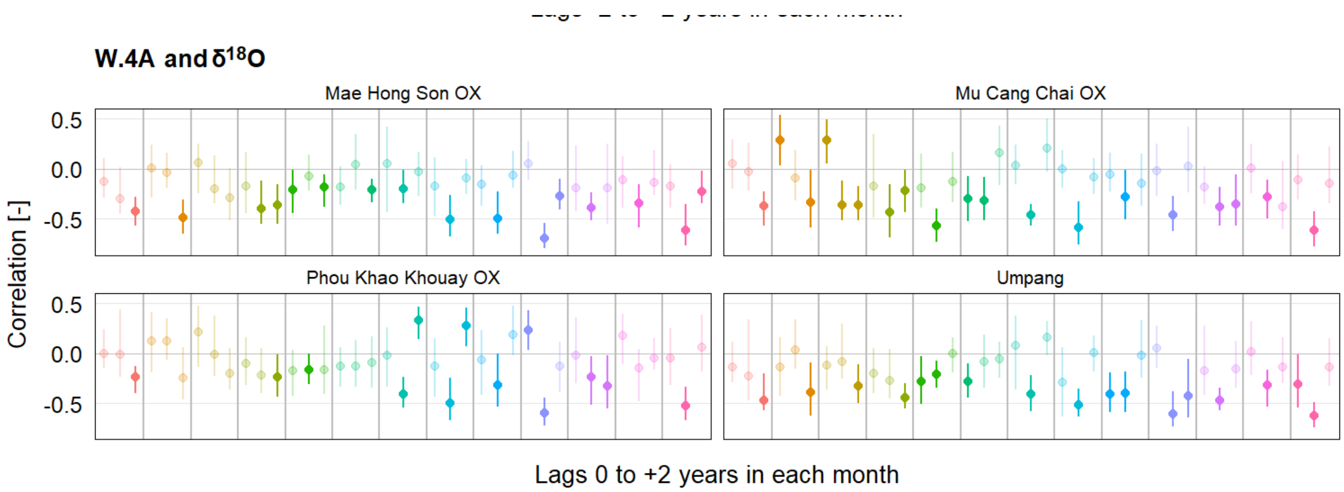


W.4A

```
rhoW4 <- cor_Q_proxy(w4tar)
plot_cor_station(rhoW4, 'W.4A')
```

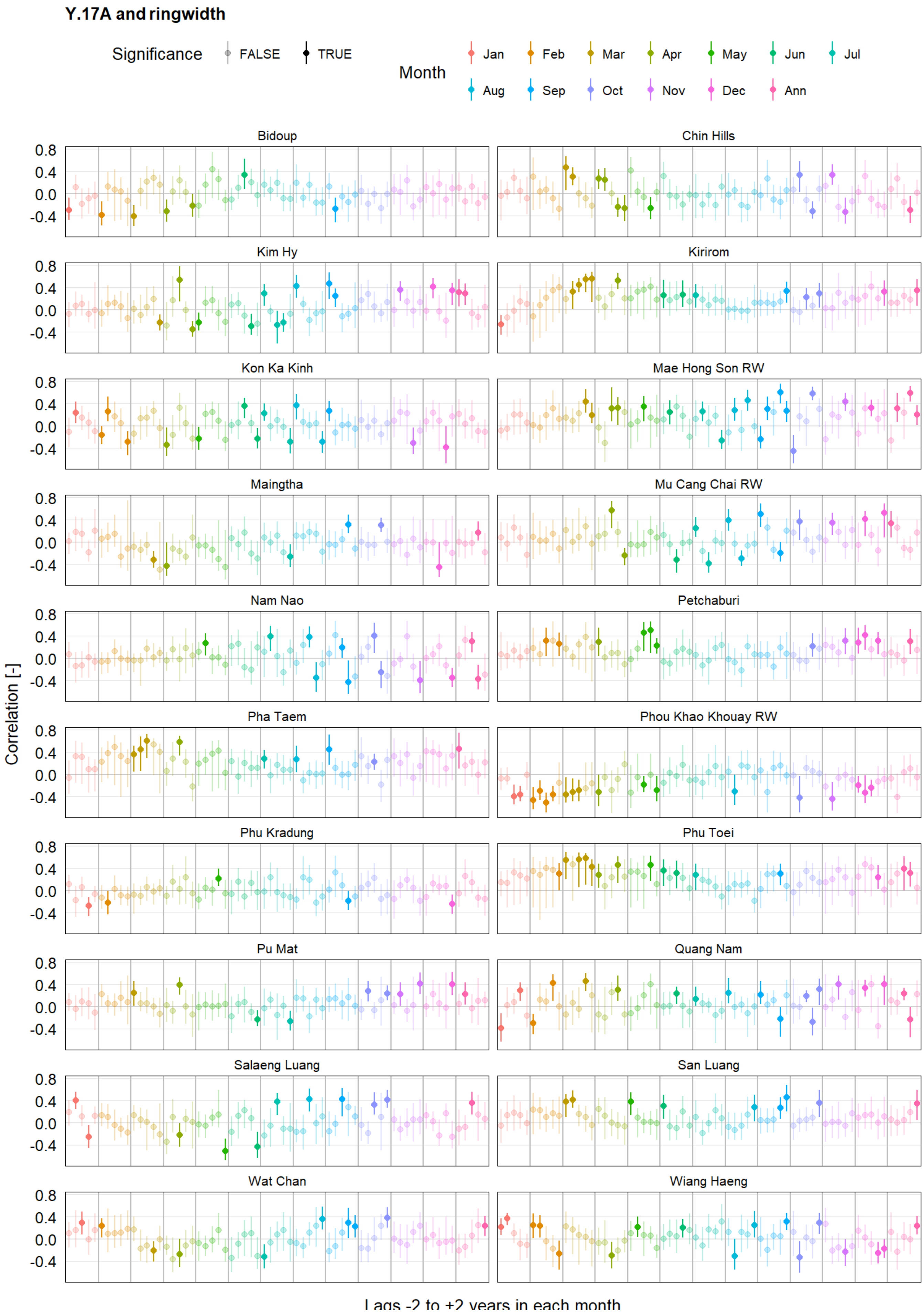


W.4A and ringwidth
Significance
FALSE
TRUE
Month
Jan
Feb
Mar
Apr
May
Jun
Jul
Aug
Sep
Oct
Nov
Dec
Ann
Bidoup
Chin Hills
Kim Hy
Kirirom
Kon Ka Kinh
Mae Hong Son RW
Maingtha
Mu Cang Chai RW
Nam Nao
Petchaburi
Pha Taem
Phou Khao Khouay RW
Phu Kradung
Phu Toei
Pu Mat
Quang Nam
Salaeng Luang
San Luang
Wat Chan
Wiang Haeng
0.5
0.0
-0.5
Correlation [-]
Lags -2 to +2 years in each month

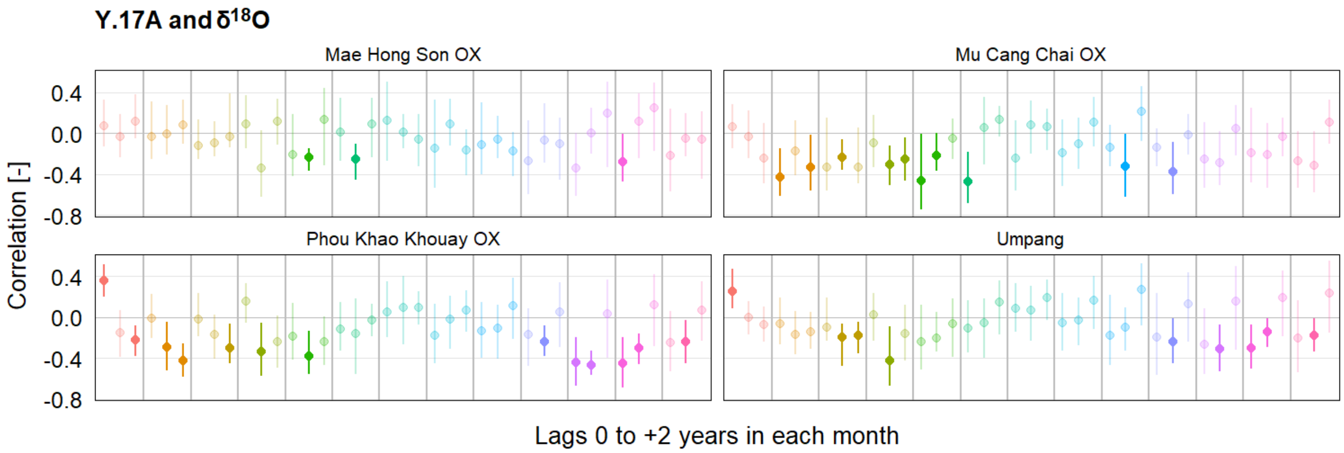


Y.17A

```
rhoY17 <- cor_Q_proxy(y17tar)
plot_cor_station(rhoY17, 'Y.17A')
```

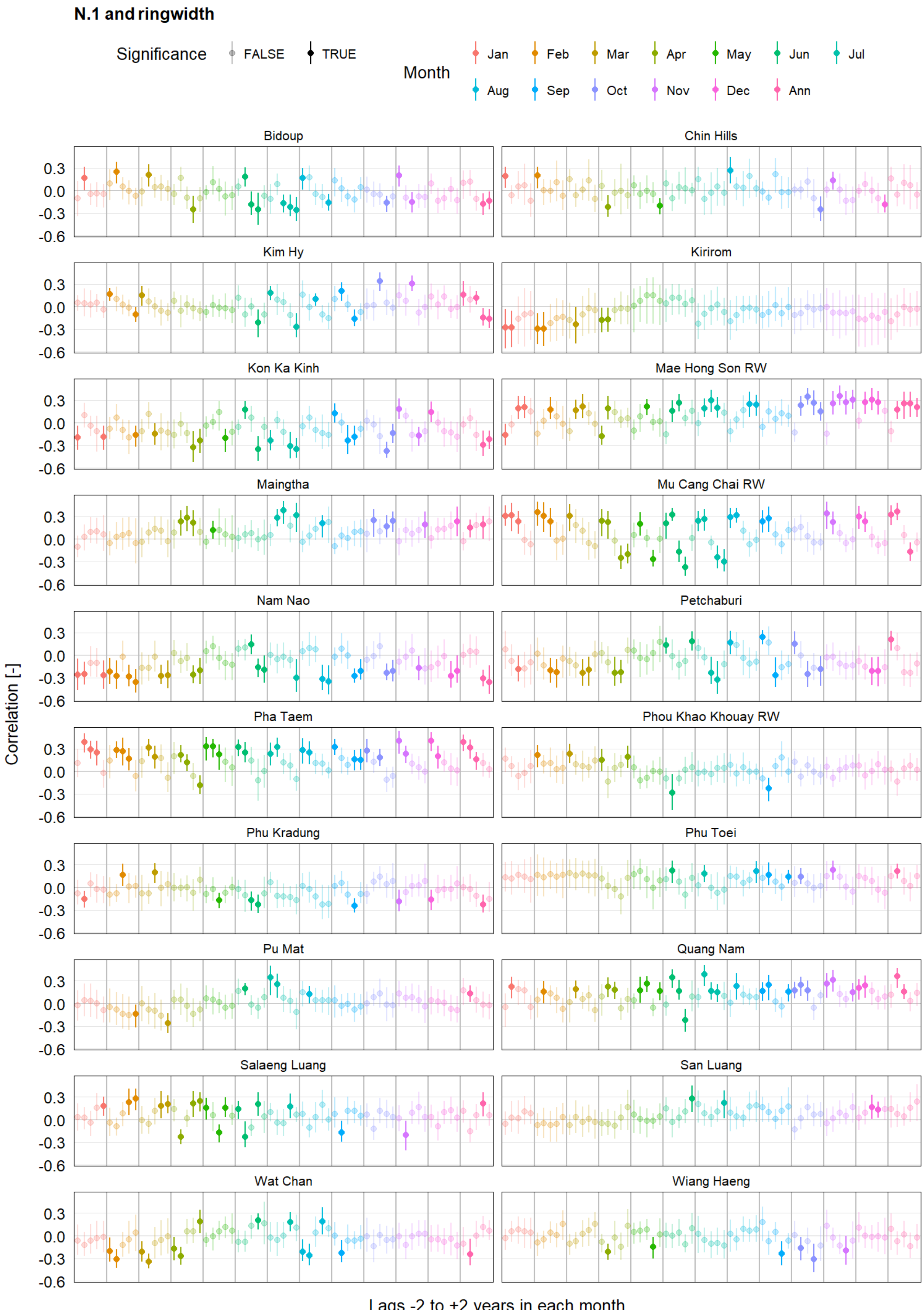


Y.17A and ringwidth
Significance
FALSE
TRUE
Month
Jan
Feb
Mar
Apr
May
Jun
Jul
Aug
Sep
Oct
Nov
Dec
Ann
Bidoup
Chin Hills
Kim Hy
Kirirom
Kon Ka Kinh
Mae Hong Son RW
Maingtha
Mu Cang Chai RW
Nam Nao
Petchaburi
Pha Taem
Phou Khao Khouay RW
Phu Kradung
Phu Toei
Pu Mat
Quang Nam
Salaeng Luang
San Luang
Wat Chan
Wiang Haeng
0.8
0.4
0.0
-0.4
Correlation [-]
Lags -2 to +2 years in each month

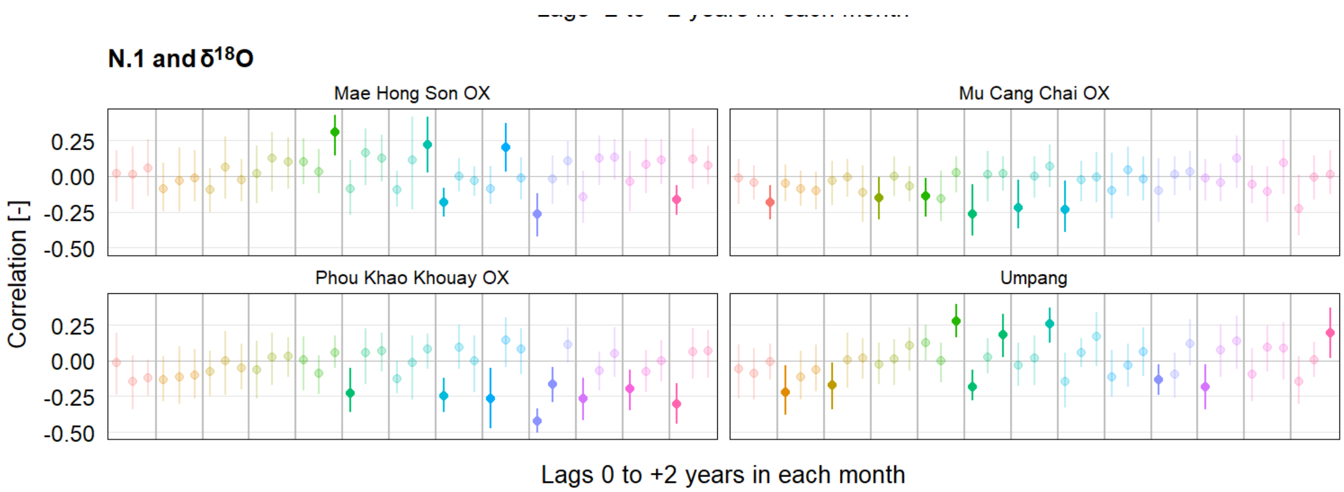


N.1

```
rhoN1 <- cor_Q_proxy(n1tar)
plot_cor_station(rhoN1, 'N.1')
```



N.1 and ringwidth
Significance
FALSE
TRUE
Month
Jan
Feb
Mar
Apr
May
Jun
Jul
Aug
Sep
Oct
Nov
Dec
Ann
Bidoup
Chin Hills
Kim Hy
Kirirom
Kon Ka Kinh
Mae Hong Son RW
Maingtha
Mu Cang Chai RW
Nam Nao
Petchaburi
Pha Taem
Phou Khao Khouay RW
Phu Kradung
Phu Toei
Pu Mat
Quang Nam
Salaeng Luang
San Luang
Wat Chan
Wiang Haeng
0.3
0.0
-0.3
-0.6
Correlation [-]
Lags -2 to +2 years in each month



Reconstruction

Pre-screening predictors

As there are many potential predictors, the genetic algorithm search may take a very long time, even on a computing cluster. To reduce the search time, we conducted a manual pre-screening. For each station, we retained only predictor-predictant pairs that have correlation magnitudes higher than a bespoke threshold \(r_0\), which was chosen such that between 5–20 predictors were retained for each predictant. \(r_0\) ranged between 0.20 and 0.28. The code to obtain the final predictor pools are as follows.

```
p1poolDT  <- prescreening(rhoP1, threshold = 0.21)
w4poolDT  <- prescreening(rhoW4, threshold = 0.28)
y17poolDT <- prescreening(rhoY17, threshold = 0.26)
n1poolDT  <- prescreening(rhoN1, threshold = 0.20)
```

Once we identify the potential predictors to retain, we can extract their time series from the respective columns of the predictor matrices.

```r
# Predictor matrix for ring width, from lags -2 to 2
Xrw <- do.call(cbind, lapply(-2:2, function(l) {
  x <- crnMat[3:256 - l, ]
  colnames(x) <- paste0(colnames(x), l)
  x
}))
# Predictor matrix for d18O, from lags 0 to 2
Xdo <- do.call(cbind, lapply(0:2, function(l) {
  x <- oxiMat[3:256 - l, ]
  colnames(x) <- paste0(colnames(x), l)
  x
}))
# Combine them into a single matrix
Xrwdo <- cbind(Xrw, Xdo)

# The predictor pool contains the column names.
# Now we extract the respective columns as indicatd by the predictor pool
p1Xused  <- Xrwdo[,  p1poolDT[, unique(site2)]]
w4Xused  <- Xrwdo[,  w4poolDT[, unique(site2)]]
y17Xused <- Xrwdo[, y17poolDT[, unique(site2)]]
n1Xused  <- Xrwdo[,  n1poolDT[, unique(site2)]]
```

Optimal input selection with incremental λ values

Demo code

The reconstruction should be run on a cluster as it is computationally heavy (instructions to run on clusters are provided next). The important settings are the lambda value for MBR, and the population and generation for GA. We varied lambda from 0 to 30 in 5-increment, fixed a population of 600, and used 600 generations. If you don't have access to a cluster, you may still run the reconstruction script on a normal laptop or desktop, but you may need to reduce the search size.

Results for the full runs are saved in the folder `results/` . Here, for demonstration, we will use only `pop = 50` and `gen = 50` .

```r
# Important settings
lambda <- 0 # To be changed from 0 to 30 in increments of 5
pop <- 50   # Fixed at 600 in actual runs
gen <- 50   # Fixed at 600 in actual runs

# Set up the cross validation folds
cvFolds <- make_Z(unique(n1m$year), nRuns = 50, frac = 0.25, contiguous = TRUE)

# Set up parallel backend using the doFuture package
registerDoFuture()
plan(multisession)

# GA search
siteOptim <- ga(
  type = 'binary',
  fitness = memoise::memoise(cv_site_selection),
  pool = n1poolDT,
  Xpool = n1Xused,
  instQ = n1tar,
  cv.folds = cvFolds,
  start.year = 1750,
  lambda = lambda,
  log.trans = NULL,
  force.standardize = TRUE,
  popSize = pop,
  maxiter = gen,
  run = min(c(gen, 100)),
  parallel = TRUE,
  monitor = FALSE,
  nBits = nrow(n1poolDT))
```

How to run on a cluster

To reproduce the full set of results with multiple λ values for all stations, there are two options:

- Manually copy the scripts and change the value of `lambda` and the reconstruction target in each copy.
- Modify the script to read `lambda` from the command line, and create a job array that runs with an array of `lambda` values and reconstruction targets.

The specific details of how to run these scripts depend on the job manager used on your cluster (e.g., SLURM or PBS). If you need help with setting up the script on your cluster, please contact Dr. Hung Nguyen (hnguyen@ldeo.columbia.edu (mailto:hnguyen@ldeo.columbia.edu)).

Build reconstructions and select λ

```r
lambdas <- seq(0, 30, 5)
names(lambdas) <- paste0('l', lambdas)
lambdaChars <- sprintf('%02d', lambdas)
```

P.1

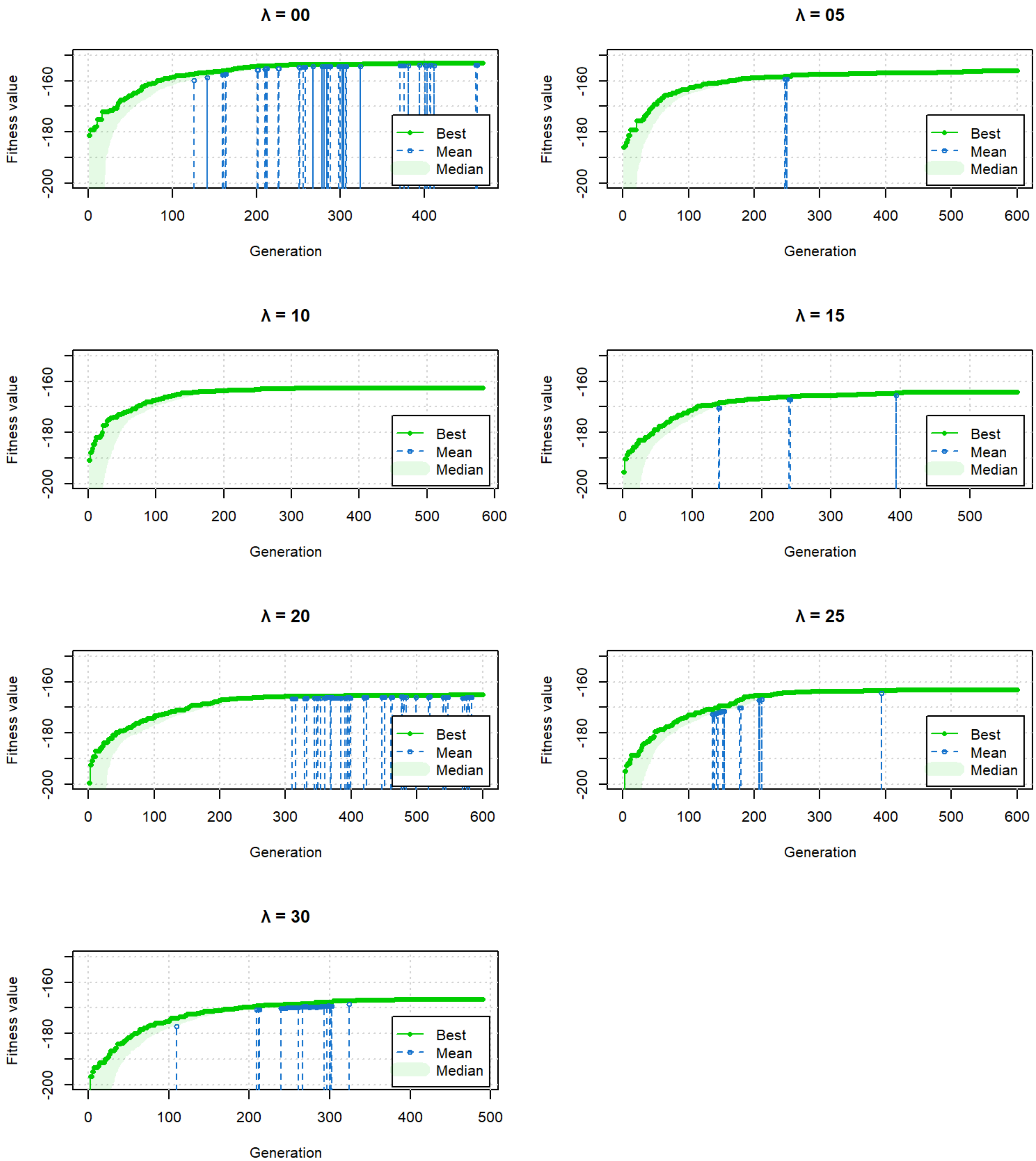
Check GA convergence

Read saved results

```
p1sols <- lapply(lambdaChars, \(s) readRDS(glue('results/P1/P1_pop600_gen600_lambda{s}_no_log
         _std_s100.RDS')))
names(p1sols) <- lambdaChars
```

Plot GA outputs

```
par(mfrow = c(4, 2))
invisible(mapply(\(s, ch) plot(s, main = paste0('\u03bb = ', ch), ylim = c(-200, -150)),
       p1sols, lambdaChars))
```

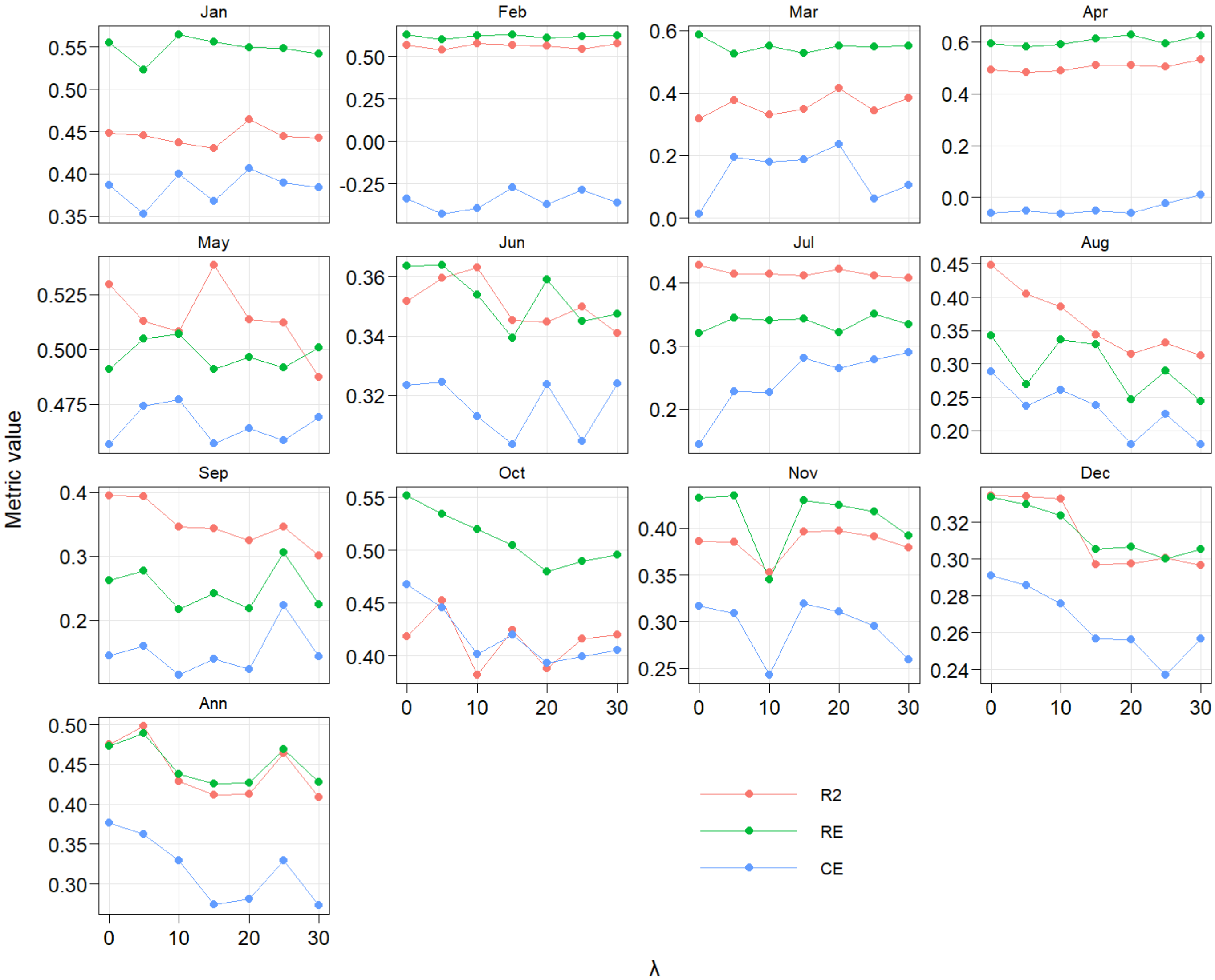


Convergence is good.

Skills

```r
p1mbPool <- rbindlist(lapply(p1sols, \(s) p1poolDT[c(s@solution) == 1]), idcol = 'lambda')
p1mbPool[, lambda := as.integer(lambda)]

set.seed(24)
p1cvFolds <- make_Z(p1years, nRuns = 50, frac = 0.25, contiguous = TRUE)

p1pcAllseasons <-
  lapply(lambdas, \(l)
    lapply(seasons, \(s) {
      Qa <- p1tar[s]
      X <- p1Xused[, p1mbPool[season == s & lambda == l, site2]]
      PC <- wPCA(X, use.eigen = FALSE, return.matrix = TRUE)
      sv <- input_selection(PC[which(1750:2005 %in% p1years), ], Qa$Qa, 'leaps backward', nvm
         ax = 8)
      PC[, sv, drop = FALSE]
  }))
p1cv <- lapply(names(lambdas), \(l)
  cv_mb(p1tar, p1pcAllseasons[[l]], p1cvFolds, 1750,
        lambda = as.numeric(substr(l, 2, 3)),
        log.trans = NULL, force.standardize = TRUE,
        return.type = 'metrics')) |>
  rbindlist(idcol = 'lambda')
p1cv[, lambda := lambdas[lambda]]
p1cv[, season := factor(season, seasons)]
p1cvMean <- p1cv[, lapply(.SD, tbrm), .SDcols = c('R2', 'RE', 'CE'), by = .(season, lambda)]
p1cvMeanLong <- melt(p1cvMean, id.vars = c('season', 'lambda'), variable.name = 'metric')
```

```r
ggplot(p1cvMeanLong) +
  geom_line(aes(lambda, value, colour = metric)) +
  geom_point(aes(lambda, value, colour = metric)) +
  facet_wrap(vars(season), ncol = 4, scales = 'free_y') +
  labs(x = '\u03bb', y = 'Metric value') +
  theme(
    panel.border = element_rect(NA, 'black', 0.2),
    panel.grid.major.x = element_line('gray90'),
    panel.grid.major.y = element_line('gray90'),
    legend.title = element_blank(),
    legend.key.width = unit(2, 'cm'),
    legend.position = c(0.6, 0.1))
```



Reconstructions

```
p1rec <- lapply(names(lambdas), \(l)
  mb_reconstruction(p1tar, p1pcAllseasons[[l]], 1750,
                    lambda = as.numeric(substr(l, 2, 3)),
                    log.trans = NULL, force.standardize = TRUE)) |>
  rbindlist()
p1rec[, season := factor(season, seasons)]
setkey(p1rec, season)
```

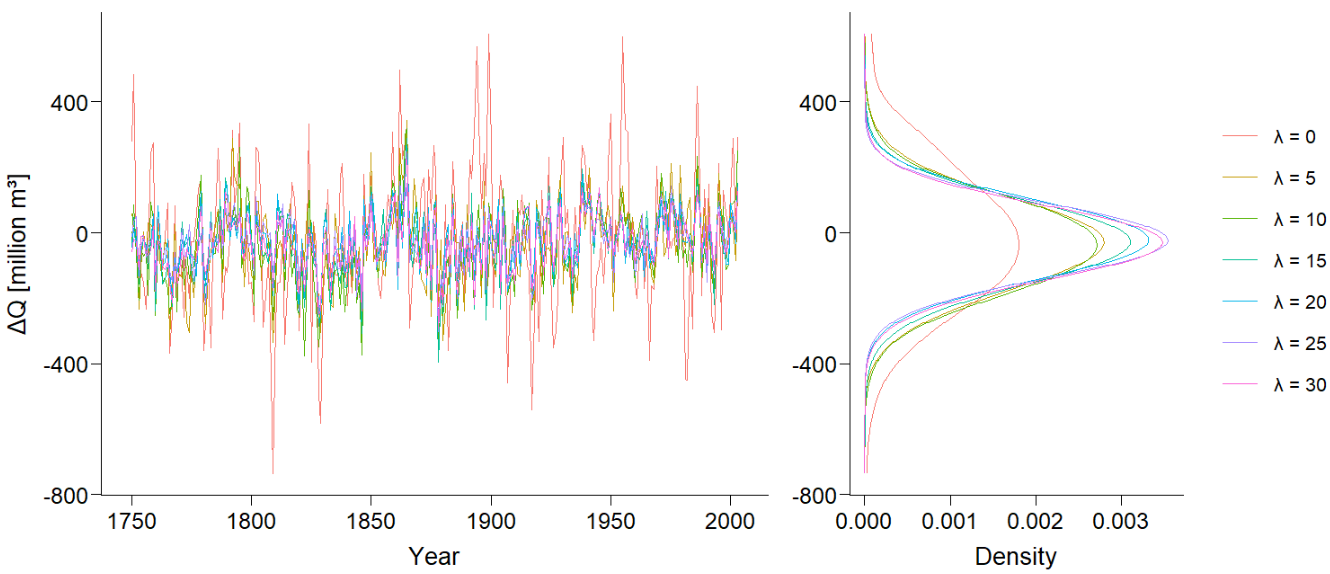
Plot for instrumental period

```
ggplot(p1rec[year %in% p1years]) +
  geom_line(aes(year, Q, colour = factor(lambda))) +
  geom_line(aes(year, Qa, colour = 'Inst'), p1tar) +
  scale_x_continuous(
    breaks = seq(1920, 2000, 10),
    labels = skip_label(2)) +
  scale_colour_discrete(name = NULL) +
  labs(x = NULL,
       y = 'Q [million m\u00b3]') +
  facet_wrap(
    vars(season),
    ncol = 2,
    scales = 'free_y',
    labeller = as_labeller(ssnLab)) +
  panel_border('black', 0.2) +
  theme(
    strip.background = element_rect('gray95', NA),
    legend.direction = 'horizontal',
    legend.position = c(0.75, 0.05),
    legend.key.width = unit(1, 'cm'))
```

Mass difference

```
p1deltaQ <- p1rec[!'Ann', .(tQ = sum(Q)), by = .(year, lambda)
                ][p1rec['Ann', .(lambda, year, Q)], on = c('year', 'lambda')
                ][, dQ := tQ - Q
                ][, lambda := factor(lambda, labels = names(lambdas))]
```

```
p1 <- ggplot(p1deltaQ) +
  geom_line(
    aes(year, dQ, colour = lambda)) +
  labs(x = 'Year', y = '\u0394Q [million m\u00b3]') +
  scale_colour_discrete(name = NULL, labels = lambdaLab)

p2 <- ggplot(p1deltaQ) +
  stat_density(
    aes(y = dQ, group = lambda, colour = lambda),
    geom = 'line',
    position = 'identity',
    bw = 80) +
  labs(x = 'Density', y = NULL) +
  scale_colour_discrete(name = NULL, labels = lambdaLab)

dqPlot <- p1 + p2 +
  plot_layout(ncol = 2, widths = c(2, 1), guides = 'collect')

dqPlot
```



Negative flow

```
p1rec[Q < 0][order(lambda)][, .N, by = lambda]
```

```
##    lambda  N
## 1:      0  8
## 2:      5 13
## 3:     10  8
## 4:     15 13
## 5:     20 11
## 6:     25  6
## 7:     30  9
```

Comparing skill scores, mass balance, and the count of months with negative flow, we chose \(\lambda = 25\).

W.4A

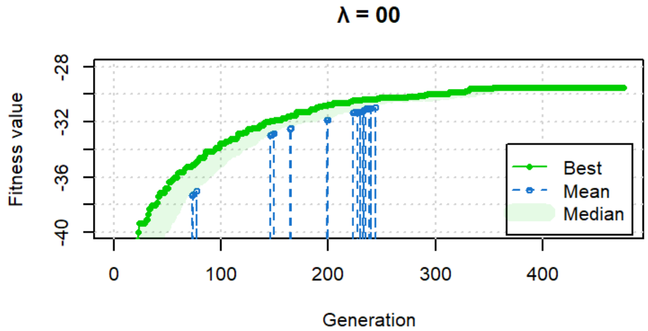
Check GA convergence

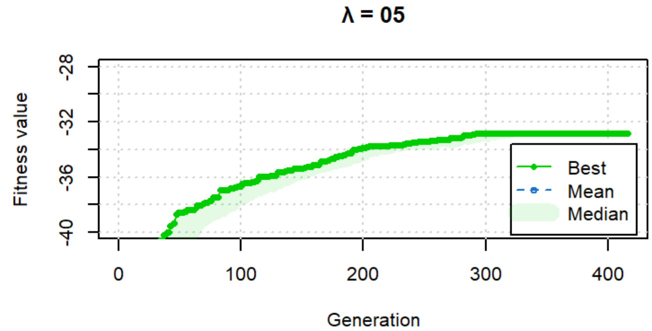
Read saved results

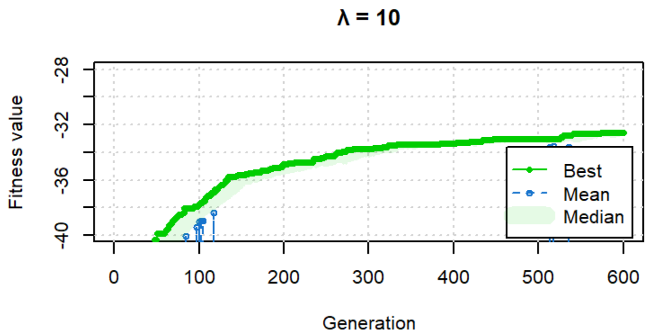
```
w4sols <- lapply(lambdaChars, \(s) readRDS(glue('results/W4A/W4_pop600_gen600_lambda{s}_no_lo
         g_std_s100.RDS')))
names(w4sols) <- lambdaChars
```

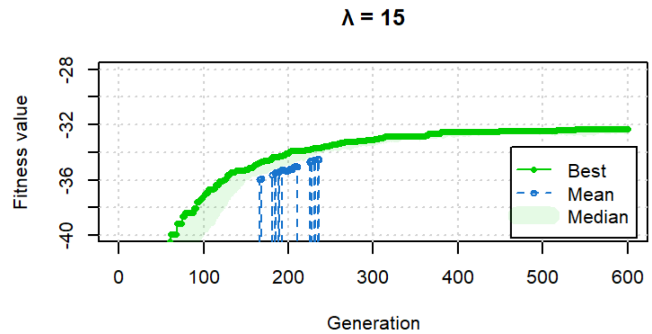
Plot GA outputs

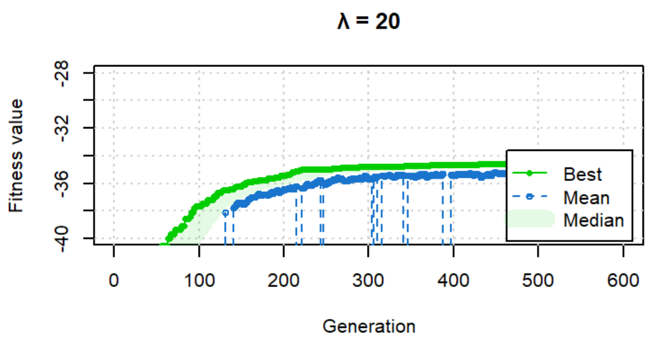
```
par(mfrow = c(4, 2))
invisible(mapply(\(s, ch) plot(s, main = paste0('\u03bb = ', ch), ylim = c(-40, -28)),
          w4sols, lambdaChars))
```

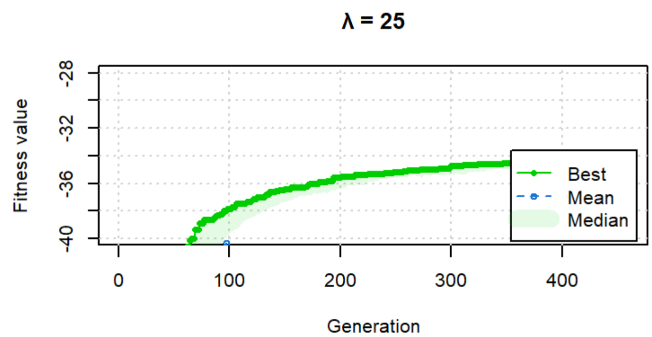


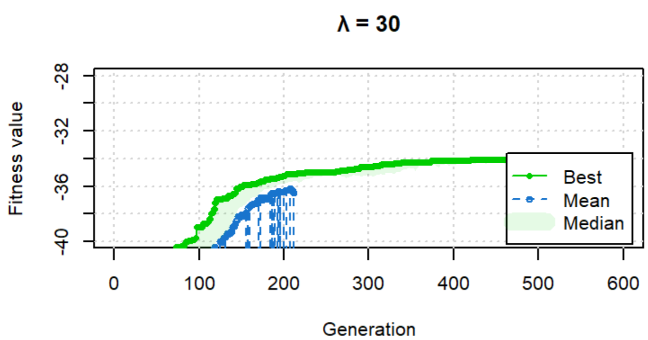










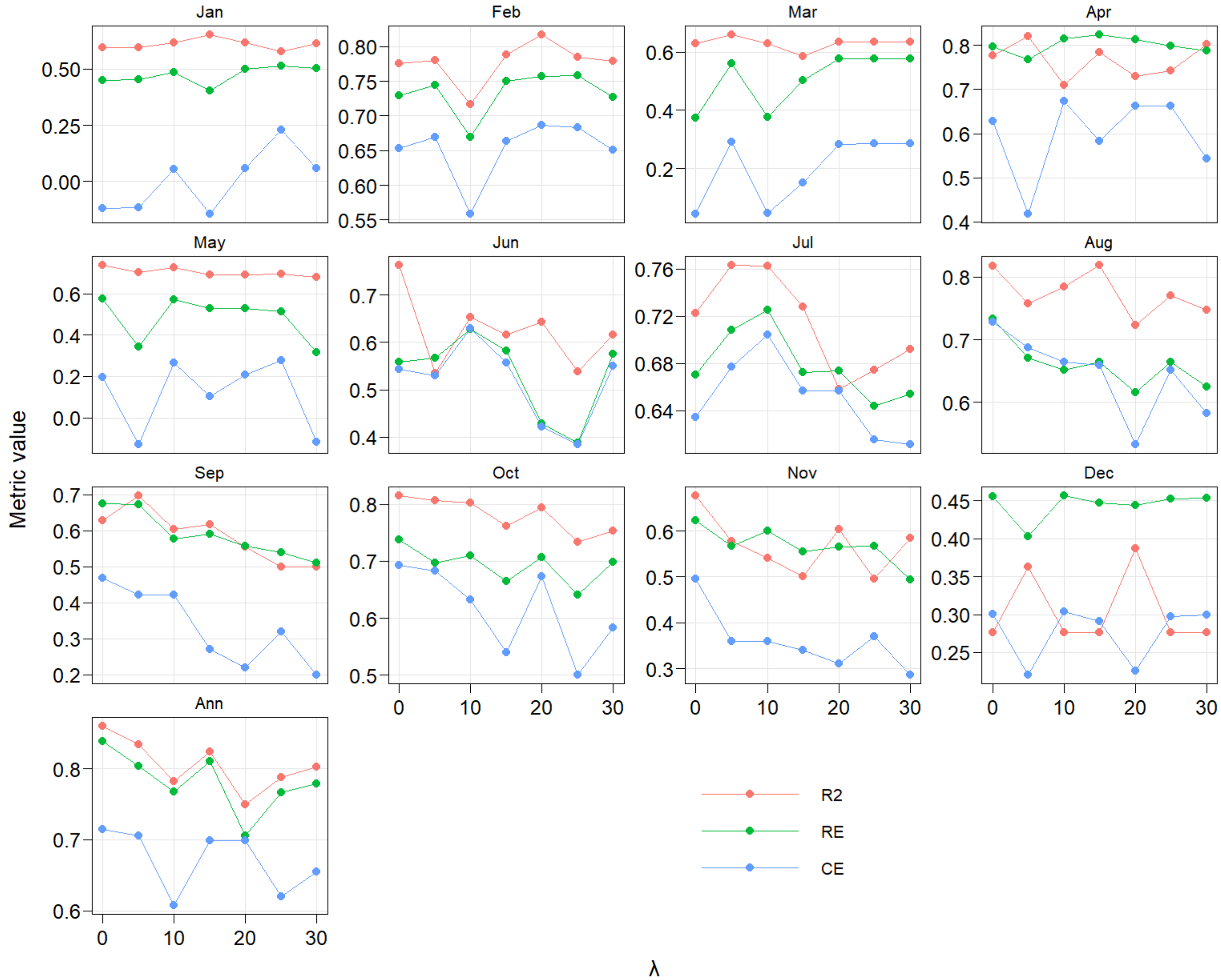


Convergence is good.

Skills

```r
w4mbPool <- rbindlist(lapply(w4sols, \(s) w4poolDT[c(s@solution) == 1]), idcol = 'lambda')
w4mbPool[, lambda := as.integer(lambda)]

set.seed(24)
w4cvFolds <- make_Z(w4years, nRuns = 20, frac = 0.25, contiguous = TRUE)

w4pcAllseasons <-
  lapply(lambdas, \(l)
    lapply(seasons, \(s) {
      Qa <- w4tar[s]
      X <- w4Xused[, w4mbPool[season == s & lambda == l, site2]]
      PC <- wPCA(X, use.eigen = FALSE, return.matrix = TRUE)
      sv <- input_selection(PC[which(1750:2005 %in% w4years), ], Qa$Qa, 'leaps backward', nvm
         ax = 8)
      PC[, sv, drop = FALSE]
  }))
w4cv <- lapply(names(lambdas), \(l)
  cv_mb(w4tar, w4pcAllseasons[[l]], w4cvFolds, 1750,
        lambda = as.numeric(substr(l, 2, 3)),
        log.trans = NULL, force.standardize = TRUE,
        return.type = 'metrics')) |>
  rbindlist(idcol = 'lambda')
w4cv[, lambda := lambdas[lambda]]
w4cv[, season := factor(season, seasons)]
w4cvMean <- w4cv[, lapply(.SD, tbrm), .SDcols = c('R2', 'RE', 'CE'), by = .(season, lambda)]
w4cvMeanLong <- melt(w4cvMean, id.vars = c('season', 'lambda'), variable.name = 'metric')
```

```r
ggplot(w4cvMeanLong) +
  geom_line(aes(lambda, value, colour = metric)) +
  geom_point(aes(lambda, value, colour = metric)) +
  facet_wrap(vars(season), ncol = 4, scales = 'free_y') +
  labs(x = '\u03bb', y = 'Metric value') +
  theme(
    panel.border = element_rect(NA, 'black', 0.2),
    panel.grid.major.x = element_line('gray90'),
    panel.grid.major.y = element_line('gray90'),
    legend.title = element_blank(),
    legend.key.width = unit(2, 'cm'),
    legend.position = c(0.6, 0.1))
```



Reconstructions

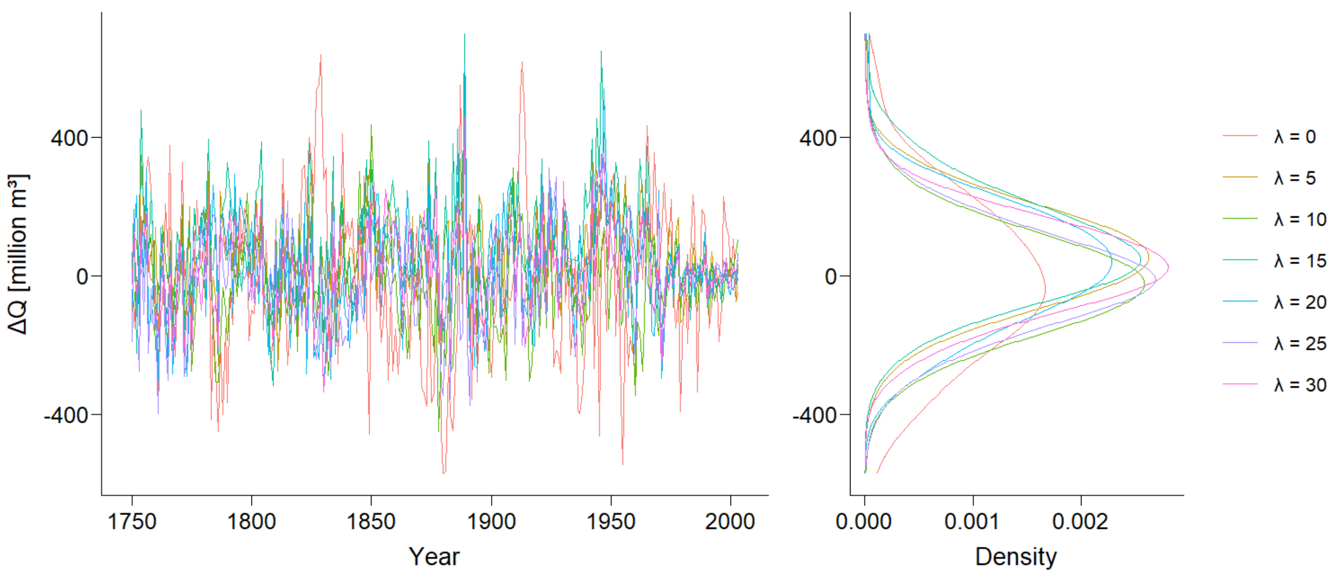
```
w4rec <- lapply(names(lambdas), \(l)
  mb_reconstruction(w4tar, w4pcAllseasons[[l]], 1750,
                    lambda = as.numeric(substr(l, 2, 3)),
                    log.trans = NULL, force.standardize = TRUE)) |>
  rbindlist()
w4rec[, season := factor(season, seasons)]
setkey(w4rec, season)
```

```
ggplot(w4rec[year %in% w4years]) +
  geom_line(aes(year, Q, colour = factor(lambda))) +
  geom_line(aes(year, Qa, colour = 'Inst'), w4tar) +
  scale_colour_discrete(name = NULL) +
  labs(x = NULL,
       y = 'Q [million m\u00b3]') +
  facet_wrap(
    vars(season),
    ncol = 2,
    scales = 'free_y',
    labeller = as_labeller(ssnLab)) +
  panel_border('black', 0.2) +
  theme(
    strip.background = element_rect('gray95', NA),
    legend.direction = 'horizontal',
    legend.position = c(0.75, 0.05),
    legend.key.width = unit(1, 'cm'))
```

Mass difference

```
w4deltaQ <- w4rec[!'Ann', .(tQ = sum(Q)), by = .(year, lambda)
                ][w4rec['Ann', .(lambda, year, Q)], on = c('year', 'lambda')
                ][, dQ := tQ - Q
                ][, lambda := factor(lambda, labels = names(lambdas))]
```

```
p1 <- ggplot(w4deltaQ) +
  geom_line(
    aes(year, dQ, colour = lambda)) +
  labs(x = 'Year', y = '\u0394Q [million m\u00b3]') +
  scale_colour_discrete(name = NULL, labels = lambdaLab)

p2 <- ggplot(w4deltaQ) +
  stat_density(
    aes(y = dQ, group = lambda, colour = lambda),
    geom = 'line',
    position = 'identity',
    bw = 80) +
  labs(x = 'Density', y = NULL) +
  scale_colour_discrete(name = NULL, labels = lambdaLab)

dqPlot <- p1 + p2 +
  plot_layout(ncol = 2, widths = c(2, 1), guides = 'collect')

dqPlot
```



Negative flow

```
w4rec[Q < 0][order(lambda)][, .N, by = lambda]
```

```
##    lambda  N
## 1:      0 76
## 2:      5 62
## 3:     10 55
## 4:     15 62
## 5:     20 74
## 6:     25 65
## 7:     30 73
```

Comparing skill scores, mass balance, and the count of months with negative flow, we chose \(\lambda = 10\).

Y.17A

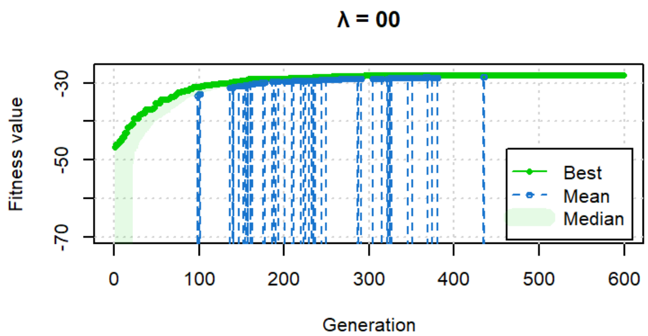
Check GA convergence

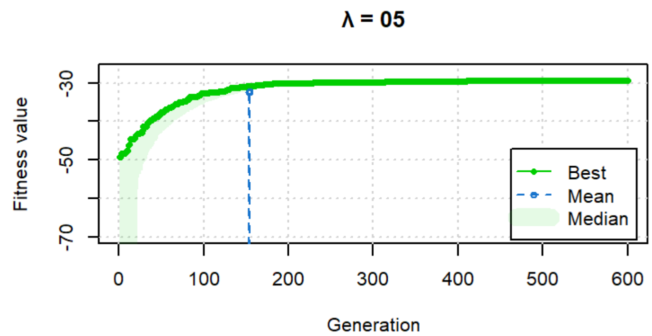
Read saved results

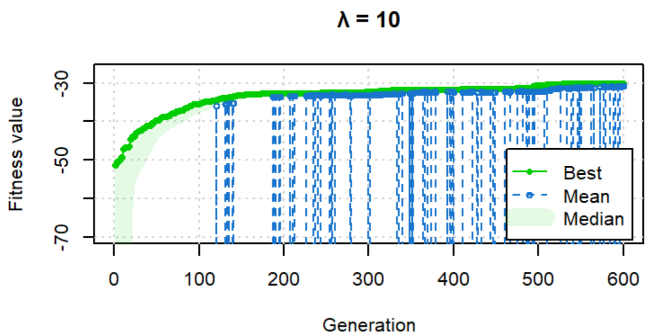
```
y17sols <- lapply(lambdaChars, \(s) readRDS(glue('results/Y17A/Y17_pop600_gen600_lambda{s}_no
         _log_std_s100.RDS')))
names(y17sols) <- lambdaChars
```

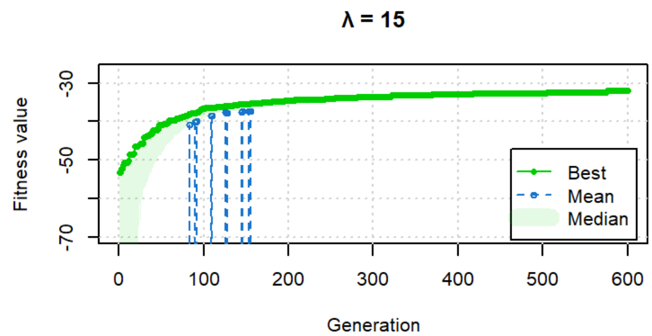
Plot GA outputs

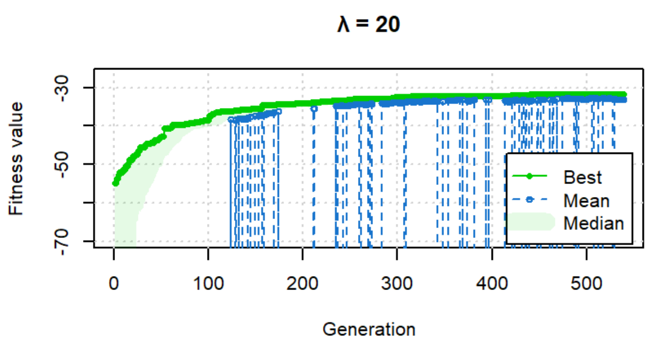
```
par(mfrow = c(4, 2))
invisible(mapply(\(s, ch) plot(s, main = paste0('\u03bb = ', ch), ylim = c(-70, -27)),
          y17sols, lambdaChars))
```

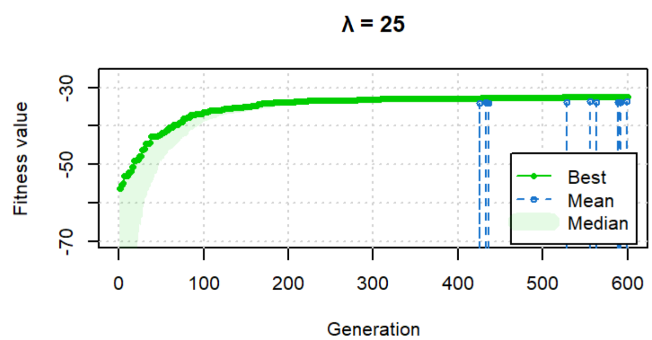


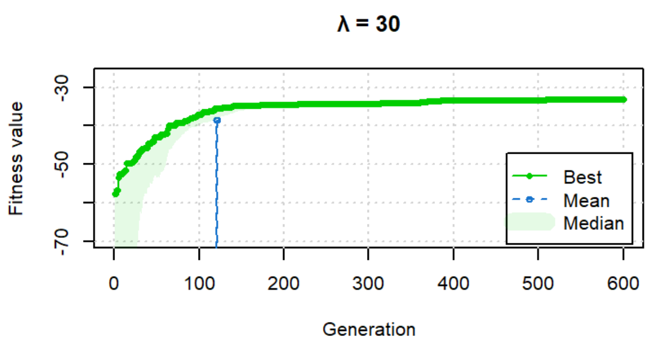










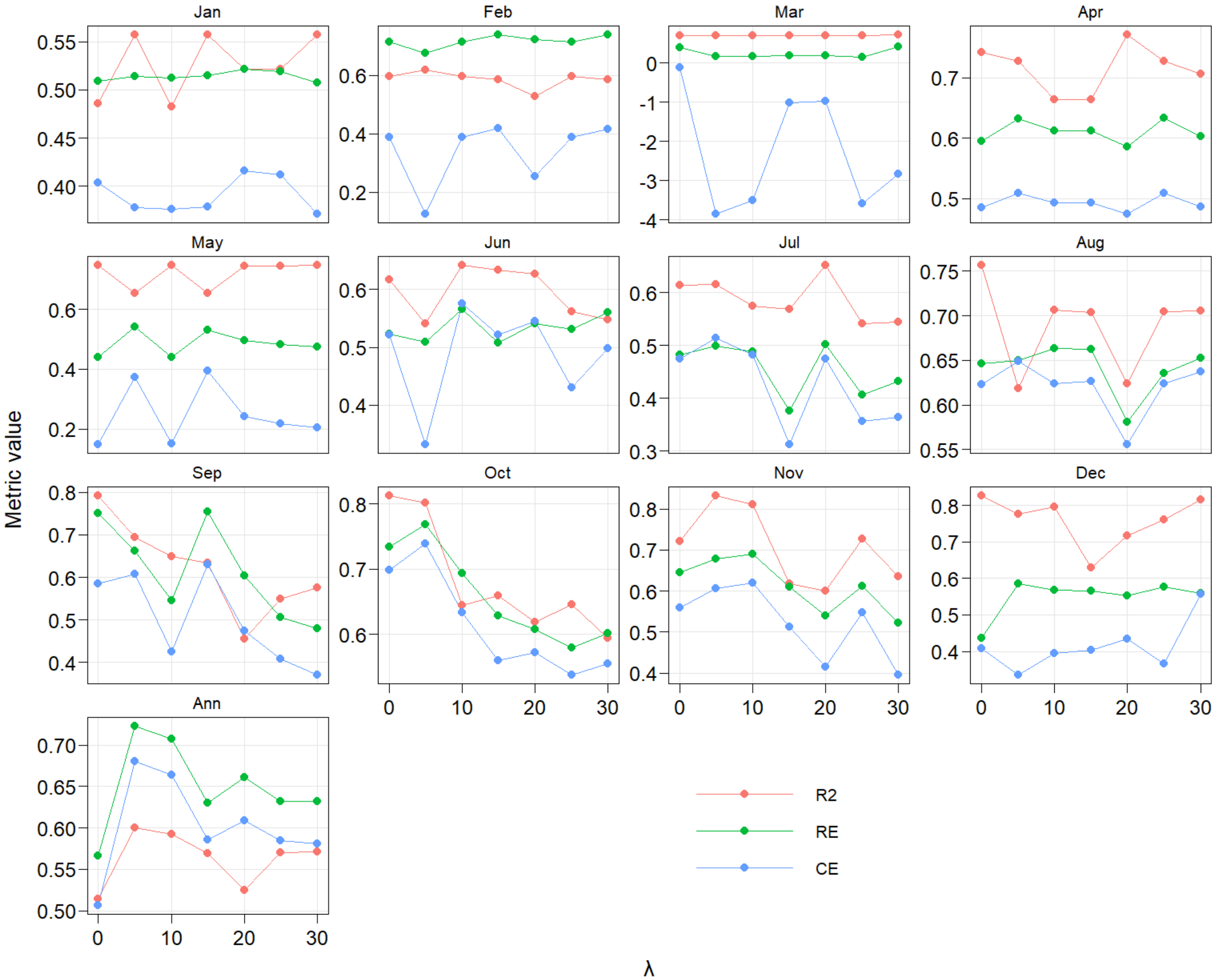


Convergence is good.

Skills

```
y17mbPool <- rbindlist(lapply(y17sols, \(s) y17poolDT[c(s@solution) == 1]), idcol = 'lambda')
y17mbPool[, lambda := as.integer(lambda)]

set.seed(24)
y17cvFolds <- make_Z(y17years, nRuns = 20, frac = 0.25, contiguous = TRUE)

y17pcAllseasons <-
  lapply(lambdas, \(l)
    lapply(seasons, \(s) {
      Qa <- y17tar[s]
      X <- y17Xused[, y17mbPool[season == s & lambda == l, site2]]
      PC <- wPCA(X, use.eigen = FALSE, return.matrix = TRUE)
      sv <- input_selection(PC[which(1750:2005 %in% y17years), ], Qa$Qa, 'leaps backward', nvmax = 8)
      PC[, sv, drop = FALSE]
  }))
y17cv <- lapply(names(lambdas), \(l)
  cv_mb(y17tar, y17pcAllseasons[[l]], y17cvFolds, 1750,
        lambda = as.numeric(substr(l, 2, 3)),
        log.trans = NULL, force.standardize = TRUE,
        return.type = 'metrics')) |>
  rbindlist(idcol = 'lambda')
y17cv[, lambda := lambdas[lambda]]
y17cv[, season := factor(season, seasons)]
y17cvMean <- y17cv[, lapply(.SD, tbrm), .SDcols = c('R2', 'RE', 'CE'), by = .(season, lambda)]
y17cvMeanLong <- melt(y17cvMean, id.vars = c('season', 'lambda'), variable.name = 'metric')
```

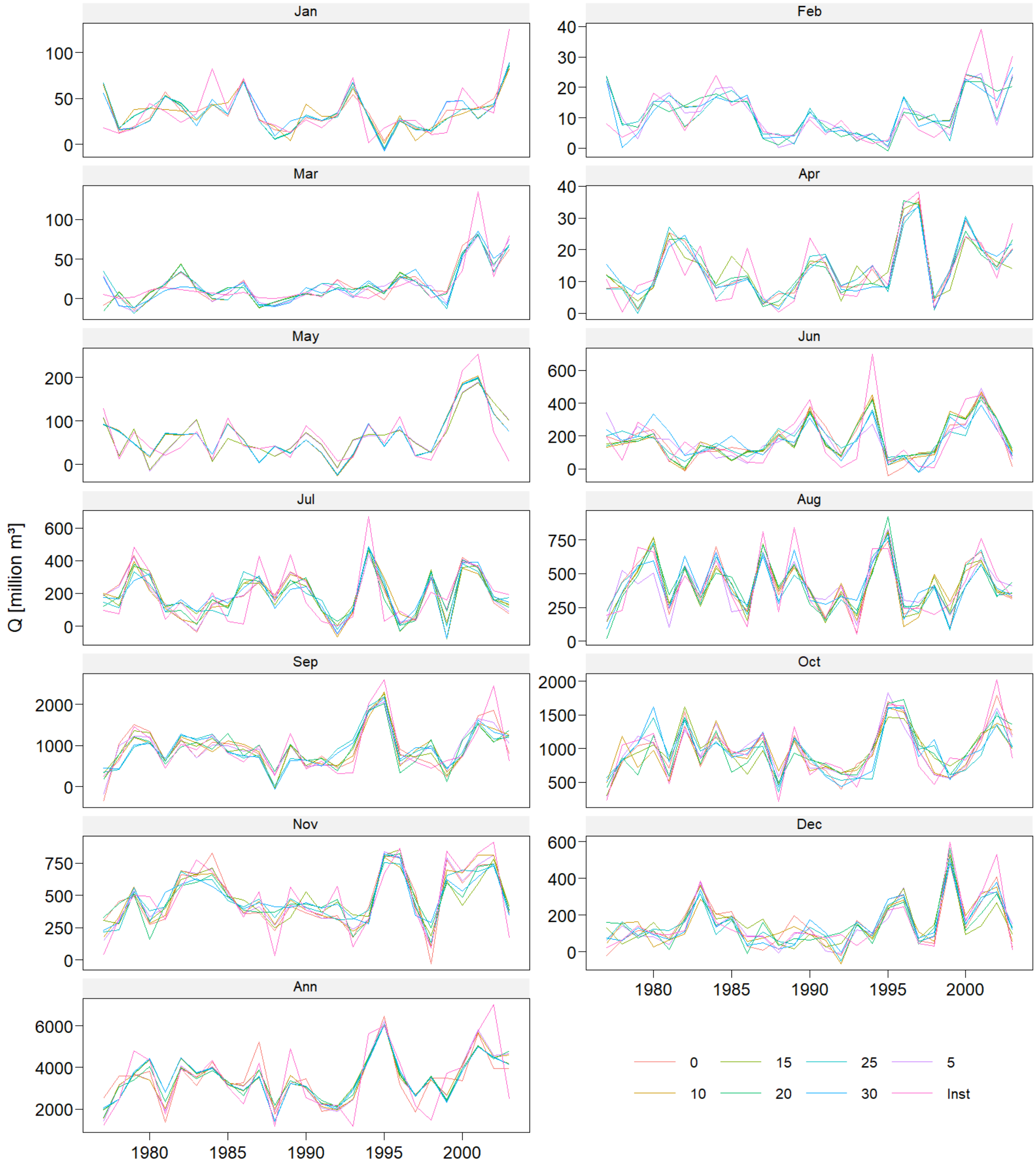
```
ggplot(y17cvMeanLong) +
  geom_line(aes(lambda, value, colour = metric)) +
  geom_point(aes(lambda, value, colour = metric)) +
  facet_wrap(vars(season), ncol = 4, scales = 'free_y') +
  labs(x = '\u03bb', y = 'Metric value') +
  theme(
    panel.border = element_rect(NA, 'black', 0.2),
    panel.grid.major.x = element_line('gray90'),
    panel.grid.major.y = element_line('gray90'),
    legend.title = element_blank(),
    legend.key.width = unit(2, 'cm'),
    legend.position = c(0.6, 0.1))
```



Reconstructions

```
y17rec <- lapply(names(lambdas), function(l)
  mb_reconstruction(y17tar, y17pcAllseasons[[l]], 1750,
                    lambda = as.numeric(substr(l, 2, 3)),
                    log.trans = NULL, force.standardize = TRUE)) |>
  rbindlist()
y17rec[, season := factor(season, seasons)]
setkey(y17rec, season)
```

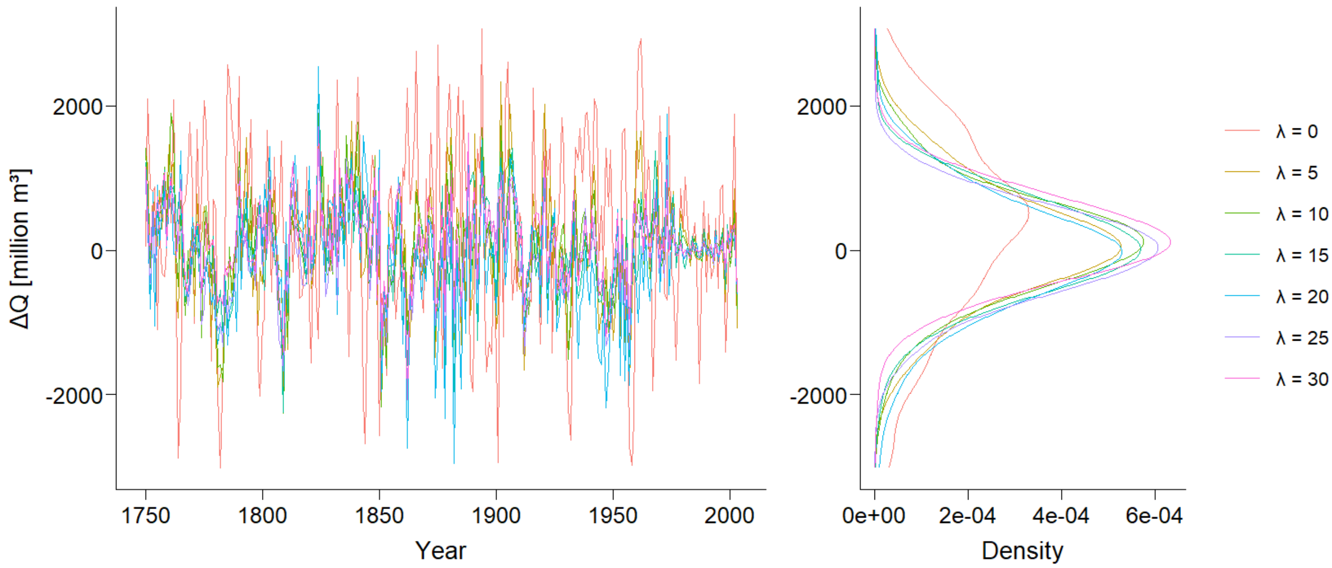
```
ggplot(y17rec[year %in% y17years]) +
  geom_line(aes(year, Q, colour = factor(lambda))) +
  geom_line(aes(year, Qa, colour = 'Inst'), y17tar) +
  scale_colour_discrete(name = NULL) +
  labs(x = NULL,
       y = 'Q [million m\u00b3]') +
  facet_wrap(
    vars(season),
    ncol = 2,
    scales = 'free_y',
    labeller = as_labeller(ssnLab)) +
  panel_border('black', 0.2) +
  theme(
    strip.background = element_rect('gray95', NA),
    legend.direction = 'horizontal',
    legend.position = c(0.75, 0.05),
    legend.key.width = unit(1, 'cm'))
```



Mass difference

```
y17deltaQ <- y17rec[!'Ann', .(tQ = sum(Q)), by = .(year, lambda)
                ][y17rec['Ann', .(lambda, year, Q)], on = c('year', 'lambda')
                ][, dQ := tQ - Q
                ][, lambda := factor(lambda, labels = names(lambdas))]
```

```
p1 <- ggplot(y17deltaQ) +
  geom_line(
    aes(year, dQ, colour = lambda)) +
  labs(x = 'Year', y = '\u0394Q [million m\u00b3]') +
  scale_colour_discrete(name = NULL, labels = lambdaLab)

p2 <- ggplot(y17deltaQ) +
  stat_density(
    aes(y = dQ, group = lambda, colour = lambda),
    geom = 'line',
    position = 'identity',
    bw = 350) +
  labs(x = 'Density', y = NULL) +
  scale_colour_discrete(name = NULL, labels = lambdaLab)

dqPlot <- p1 + p2 +
  plot_layout(ncol = 2, widths = c(2, 1), guides = 'collect')

dqPlot
```



Negative flow

```
y17rec[Q < 0][order(lambda)][, .N, by = lambda]
```

```
##    lambda   N
## 1:      0 177
## 2:      5 185
## 3:     10 145
## 4:     15 188
## 5:     20 181
## 6:     25 162
## 7:     30 155
```

Comparing skill scores, mass balance, and the count of months with negative flow, we chose \(\lambda = 0\).

N.1

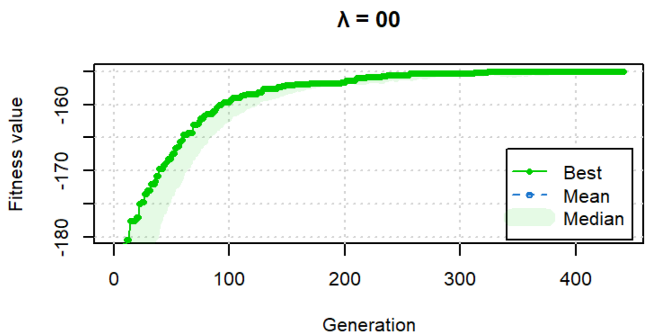
Check GA convergence

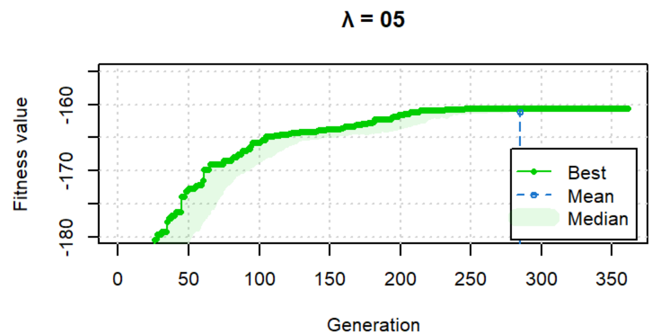
Read saved results

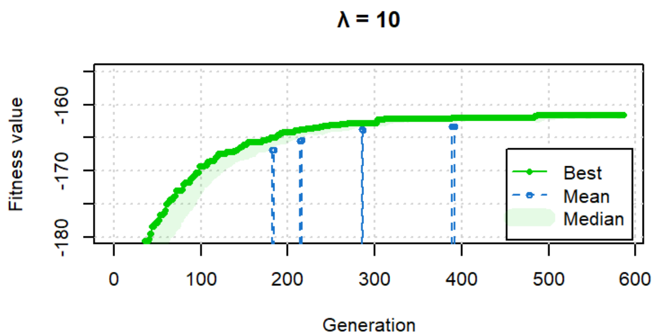
```
n1sols <- lapply(lambdaChars, \(s) readRDS(glue('results/N1/N1_pop600_gen600_lambda{s}_no_log
         _std_s100.RDS')))
names(n1sols) <- lambdaChars
```

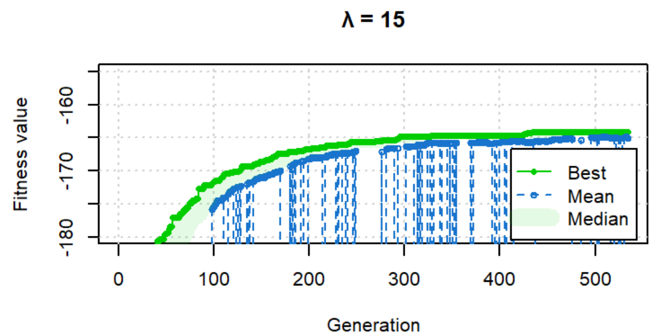
Plot GA outputs

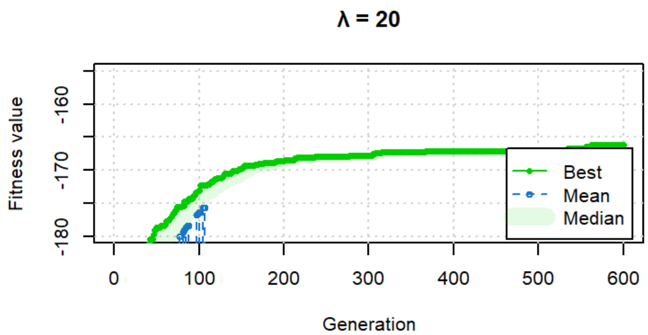
```
par(mfrow = c(4, 2))
invisible(mapply(\(s, ch) plot(s, main = paste0('\u03bb = ', ch), ylim = c(-180, -155)),
          n1sols, lambdaChars))
```

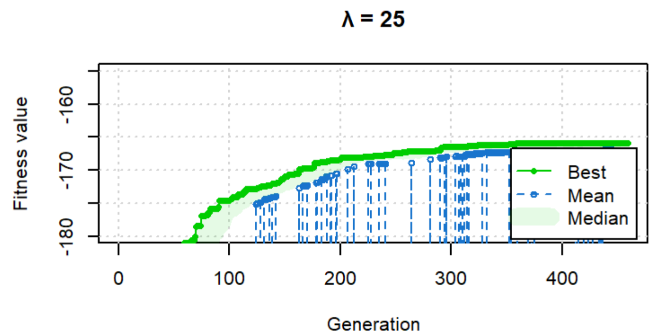


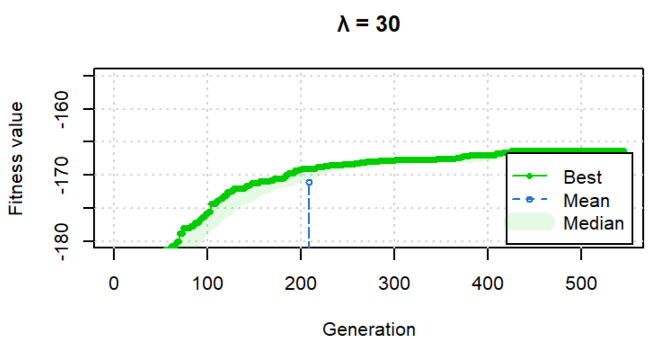










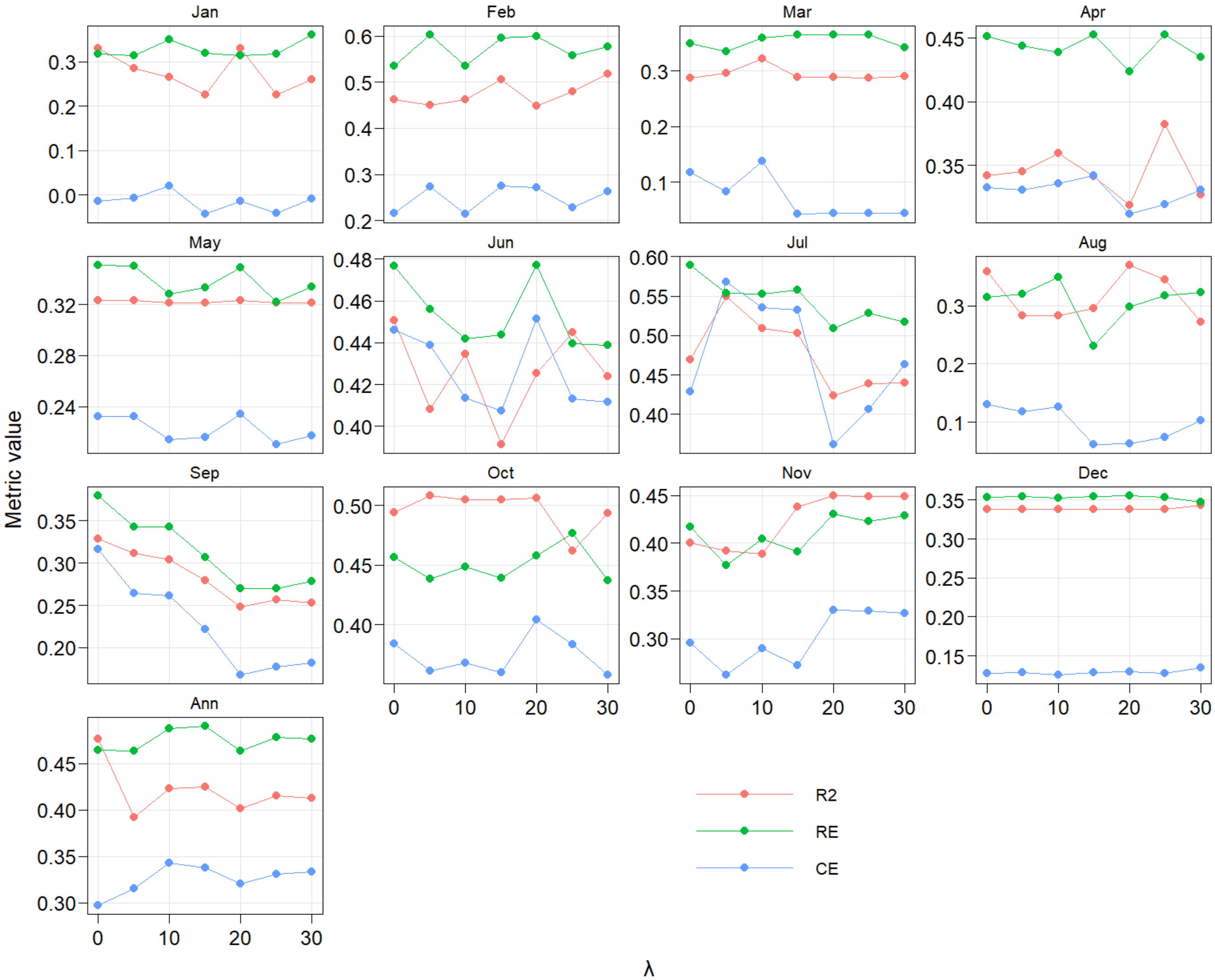


Convergence is good.

Skills

```r
n1mbPool <- rbindlist(lapply(n1sols, \(s) n1poolDT[c(s@solution) == 1]), idcol = 'lambda')
n1mbPool[, lambda := as.integer(lambda)]

set.seed(24)
n1cvFolds <- make_Z(n1years, nRuns = 50, frac = 0.25, contiguous = TRUE)

n1pcAllseasons <-
  lapply(lambdas, \(l)
    lapply(seasons, \(s) {
      Qa <- n1tar[s]
      X <- n1Xused[, n1mbPool[season == s & lambda == l, site2]]
      PC <- wPCA(X, use.eigen = FALSE, return.matrix = TRUE)
      sv <- input_selection(PC[which(1750:2005 %in% n1years), ], Qa$Qa, 'leaps backward', nvmax = 8)
      PC[, sv, drop = FALSE]
  }))
n1cv <- lapply(names(lambdas), \(l)
  cv_mb(n1tar, n1pcAllseasons[[l]], n1cvFolds, 1750,
        lambda = as.numeric(substr(l, 2, 3)),
        log.trans = NULL, force.standardize = TRUE,
        return.type = 'metrics')) |>
  rbindlist(idcol = 'lambda')
n1cv[, lambda := lambdas[lambda]]
n1cv[, season := factor(season, seasons)]
n1cvMean <- n1cv[, lapply(.SD, tbrm), .SDcols = c('R2', 'RE', 'CE'), by = .(season, lambda)]
n1cvMeanLong <- melt(n1cvMean, id.vars = c('season', 'lambda'), variable.name = 'metric')
```

```r
ggplot(n1cvMeanLong) +
  geom_line(aes(lambda, value, colour = metric)) +
  geom_point(aes(lambda, value, colour = metric)) +
  facet_wrap(vars(season), ncol = 4, scales = 'free_y') +
  labs(x = '\u03bb', y = 'Metric value') +
  theme(
    panel.border = element_rect(NA, 'black', 0.2),
    panel.grid.major.x = element_line('gray90'),
    panel.grid.major.y = element_line('gray90'),
    legend.title = element_blank(),
    legend.key.width = unit(2, 'cm'),
    legend.position = c(0.6, 0.1))
```



Reconstructions

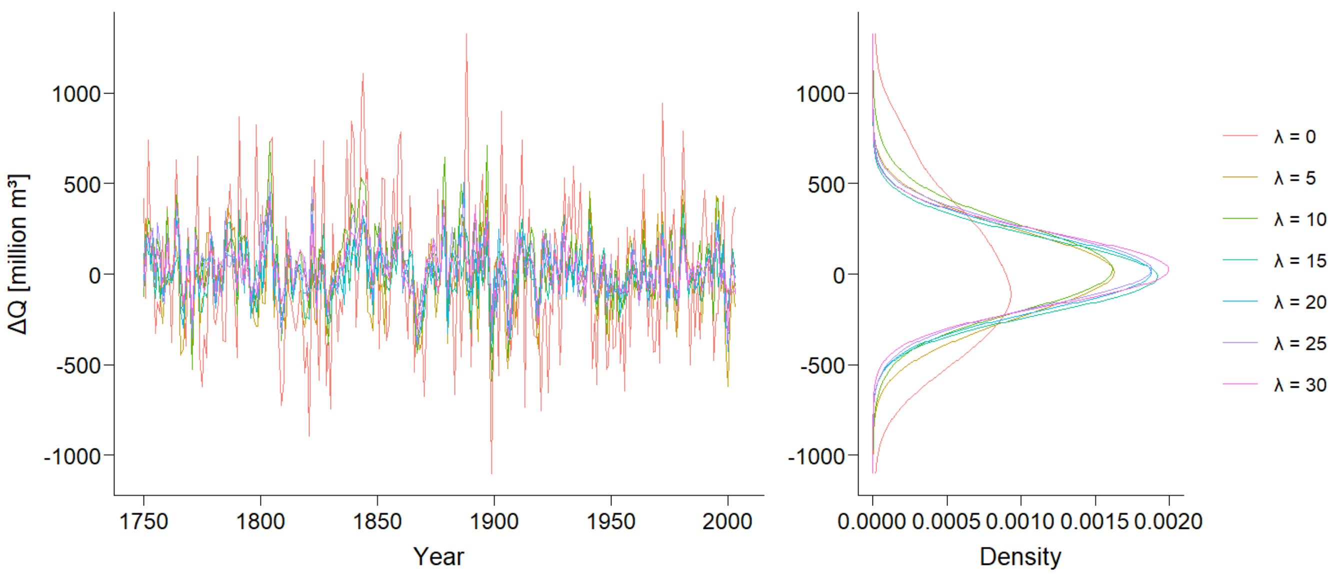
```
n1rec <- lapply(names(lambdas), \(l)
  mb_reconstruction(n1tar, n1pcAllseasons[[l]], 1750,
                    lambda = as.numeric(substr(l, 2, 3)),
                    log.trans = NULL, force.standardize = TRUE)) |>
  rbindlist()
n1rec[, season := factor(season, seasons)]
setkey(n1rec, season)
```

```
ggplot(n1rec[year %in% n1years]) +
  geom_line(aes(year, Q, colour = factor(lambda))) +
  geom_line(aes(year, Qa, colour = 'Inst'), n1tar) +
  scale_colour_discrete(name = NULL) +
  labs(x = NULL,
       y = 'Q [million m\u00b3]') +
  facet_wrap(
    vars(season),
    ncol = 2,
    scales = 'free_y',
    labeller = as_labeller(ssnLab)) +
  panel_border('black', 0.2) +
  theme(
    strip.background = element_rect('gray95', NA),
    legend.direction = 'horizontal',
    legend.position = c(0.75, 0.05),
    legend.key.width = unit(1, 'cm'))
```

Mass difference

```
n1deltaQ <- n1rec[!'Ann', .(tQ = sum(Q)), by = .(year, lambda)
                ][n1rec['Ann', .(lambda, year, Q)], on = c('year', 'lambda')
                ][, dQ := tQ - Q
                ][, lambda := factor(lambda, labels = names(lambdas))]
```

```
p1 <- ggplot(n1deltaQ) +
  geom_line(
    aes(year, dQ, colour = lambda)) +
  labs(x = 'Year', y = '\u0394Q [million m\u00b3]') +
  scale_colour_discrete(name = NULL, labels = lambdaLab)

p2 <- ggplot(n1deltaQ) +
  stat_density(
    aes(y = dQ, group = lambda, colour = lambda),
    geom = 'line',
    position = 'identity',
    bw = 150) +
  labs(x = 'Density', y = NULL) +
  scale_colour_discrete(name = NULL, labels = lambdaLab)

dqPlot <- p1 + p2 +
  plot_layout(ncol = 2, widths = c(2, 1), guides = 'collect')

dqPlot
```



Negative flow

```
n1rec[Q < 0][order(lambda)][, .N, by = lambda]
```

```
##    lambda N
## 1:      0 4
## 2:      5 5
## 3:     10 3
## 4:     15 3
## 5:     20 3
## 6:     25 4
## 7:     30 2
```

Comparing skill scores, mass balance, and the count of months with negative flow, we chose $\lambda = 10$.

Final reconstructions and skills

Skills for full time series

```
p1cvQ <- cv_mb(p1tar, p1pcAllseasons$l25, p1cvFolds, 1750,
        lambda = 25,
        log.trans = NULL, force.standardize = TRUE,
        return.type = 'Q')[season != 'Ann'][order(rep, year, season)]
numYears <- length(p1years)
ZMat <- matrix(seq_len(numYears * 12), nrow = numYears)
p1cvQScores <- p1cvQ[, {
  r <- .BY$rep
  z <- p1cvFolds[[r]]
  Z <- c(ZMat[z, ])
  as.data.table(t(calculate_metrics(.SD$Q, .SD$Qa, Z)))
}, by = rep
  ][, lapply(.SD, tbrm), .SDcols = c('R2', 'RE', 'CE')]

w4cvQ <- cv_mb(w4tar, w4pcAllseasons$l10, w4cvFolds, 1750,
        lambda = 10,
        log.trans = NULL, force.standardize = TRUE,
        return.type = 'Q')[season != 'Ann'][order(rep, year, season)]
numYears <- length(w4years)
ZMat <- matrix(seq_len(numYears * 12), nrow = numYears)
w4cvQScores <- w4cvQ[, {
  r <- .BY$rep
  z <- w4cvFolds[[r]]
  Z <- c(ZMat[z, ])
  as.data.table(t(calculate_metrics(.SD$Q, .SD$Qa, Z)))
}, by = rep
  ][, lapply(.SD, tbrm), .SDcols = c('R2', 'RE', 'CE')]

y17cvQ <- cv_mb(y17tar, y17pcAllseasons$l0, y17cvFolds, 1750,
        lambda = 0,
        log.trans = NULL, force.standardize = TRUE,
        return.type = 'Q')[season != 'Ann'][order(rep, year, season)]
numYears <- length(y17years)
ZMat <- matrix(seq_len(numYears * 12), nrow = numYears)
y17cvQScores <- y17cvQ[, {
  r <- .BY$rep
  z <- y17cvFolds[[r]]
  Z <- c(ZMat[z, ])
  as.data.table(t(calculate_metrics(.SD$Q, .SD$Qa, Z)))
}, by = rep
  ][, lapply(.SD, tbrm), .SDcols = c('R2', 'RE', 'CE')]

n1cvQ <- cv_mb(n1tar, n1pcAllseasons$l10, n1cvFolds, 1750,
        lambda = 10,
        log.trans = NULL, force.standardize = TRUE,
        return.type = 'Q')[season != 'Ann'][order(rep, year, season)]
numYears <- length(n1years)
ZMat <- matrix(seq_len(numYears * 12), nrow = numYears)
n1cvQScores <- n1cvQ[, {
  r <- .BY$rep
  z <- n1cvFolds[[r]]
```

```
  Z <- c(ZMat[z, ])
  as.data.table(t(calculate_metrics(.SD$Q, .SD$Qa, Z)))
}, by = rep
  ][, lapply(.SD, tbrm), .SDcols = c('R2', 'RE', 'CE')]
```

Combining all results

```
# Monthly reconstructions
rec <- rbindlist(
  list(
    p1  =  p1rec[lambda == 25],
    w4  =  w4rec[lambda == 10],
    y17 = y17rec[lambda ==  0],
    n1  =  n1rec[lambda == 10]),
  idcol = 'station')[season != 'Ann']
rec[Q < 0, Q := 0.01]
rec[, month := as.numeric(season)
  ][, c('season', 'lambda') := NULL
  ][, month2 := factor(month.abb[month], month.abb)
  ][, station := factor(station, stations)]
setkey(rec, station)
setorder(rec, station, year, month)

# Skills for individual targets
scores <- rbindlist(
  list(
    p1  =  p1cvMeanLong[lambda == 25],
    w4  =  w4cvMeanLong[lambda == 10],
    y17 = y17cvMeanLong[lambda ==  0],
    n1  =  n1cvMeanLong[lambda == 10]),
  idcol = 'station')
scores[, ':='(station = factor(station, stations),
              lambda = NULL)]

# Skills for full time series
scoreQ <- rbindlist(
  list(p1 = p1cvQScores, w4 = w4cvQScores, y17 = y17cvQScores, n1 = n1cvQScores),
  idcol = 'station') |>
  roundDT()
scoreQ[, station := factor(station, stations)]

# Merged data of instrumental period for plotting
instDT <- merge(rec, inst, by = c('station', 'year', 'month', 'month2'), suffixes = c('rec',
         'obs'))
```

Results

Final skill scores

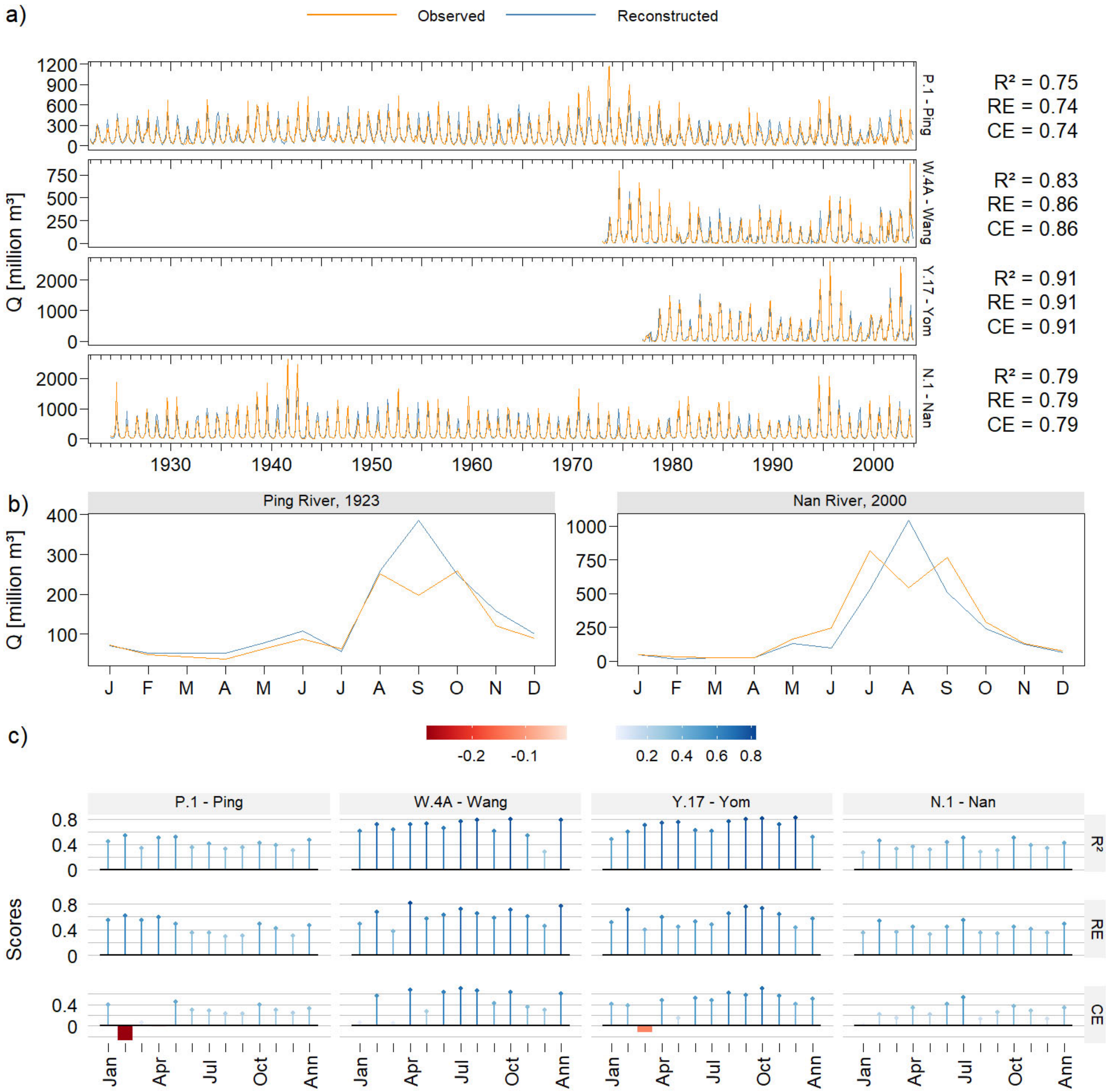
First we create the examples of years with two peaks in panel b.

```
y2p <- data.table(
  station = c('p1', 'n1'),
  year = c(1923, 2000))
rec2pMonth <- merge(instDT, y2p, by = c('year', 'station'))
rec2pMonth[, station := factor(fifelse(station == 'p1', 'Ping River', 'Nan River'),
                               levels = c('Ping River', 'Nan River'))]
```

Figure 2

```
p1 <- ggplot(instDT) +
  geom_line(aes(year + (month - 1) / 12, Qrec, colour = 'Reconstructed')) +
  geom_line(aes(year + (month - 1) / 12, Qobs, colour = 'Observed')) +
  scale_colour_manual(name = NULL, values = c('darkorange', 'steelblue')) +
  scale_x_continuous(
    guide = guide_prism_minor(),
    expand = c(0, 0.25),
    breaks = seq(1930, 2005, 5),
    labels = skip_label(2),
    minor_breaks = 1920:2005,
    limits = c(1922, 2004)) +
  facet_wrap(~station, ncol = 1, scales = 'free_y',
             labeller = as_labeller(stnLab), strip.position = 'right') +
  guides(color = guide_legend(override.aes = list(size = 0.4))) +
  annotation_ticks(sides = 't', type = 'both', size = 0.2) +
  theme(
    strip.text = element_text(size = 7),
    plot.tag.position = c(0.012, 0.95),
    legend.key.width = unit(2, 'cm'),
    legend.position = 'top') +
  labs(x = NULL, y = 'Q [million m\u00b3]', tag = 'a)') +
  panel_border('black', 0.2)

p2 <- ggplot(scoreQ) +
  geom_text(aes(x = 1, y = 3, label = paste0('R\u00b2 = ', R2)), hjust = 1, size = 3.5) +
  geom_text(aes(x = 1, y = 2, label = paste0('RE = ', RE)), hjust = 1, size = 3.5) +
  geom_text(aes(x = 1, y = 1, label = paste0('CE = ', CE)), hjust = 1, size = 3.5) +
  facet_wrap(vars(station), ncol = 1, strip.position = 'right') +
  scale_x_continuous(limits = c(0.75, 1)) +
  scale_y_continuous(expand = c(0.4, 0)) +
  theme_void() +
  theme(strip.text = element_blank())
g1 <- p1 + p2 +
  plot_layout(widths = c(6, 1))

p3 <- ggplot(rec2pMonth) +
  geom_line(aes(month, Qrec, colour = 'Reconstructed')) +
  geom_line(aes(month, Qobs, colour = 'Observed')) +
  scale_colour_manual(name = NULL, values = c('darkorange', 'steelblue')) +
  scale_x_continuous(breaks = 1:12, labels = monthLabNum) +
  facet_wrap(vars(station, year), nrow = 1, labeller = label_wrap_gen(multi_line = FALSE), sc
        ales = 'free_y') +
  labs(x = NULL, y = 'Q [million m\u00b3]', tag = 'b)') +
  theme(
    plot.tag.position = c(0.012, 0.95),
    strip.background = element_rect('gray90', NA),
    legend.position = 'none',
    legend.key.width = unit(2, 'cm')) +
  panel_border('black', 0.2)

p4 <- ggplot() +
```

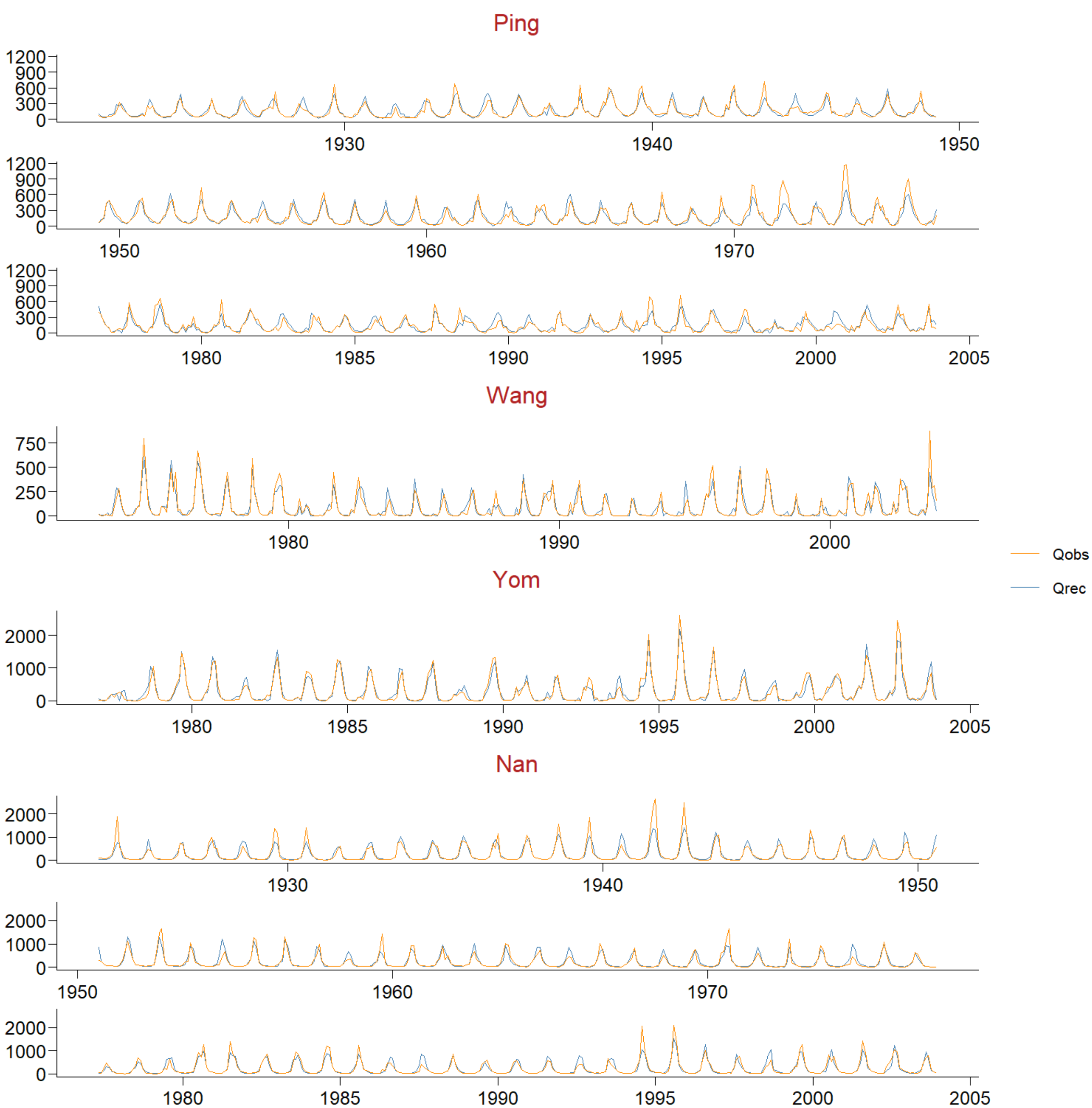
```
  facet_grid(vars(metric), vars(station), scales = 'free', labeller = labeller(.rows = metLa
         b, .cols = stnLab)) +
  geom_linerange(aes(ymin = 0, ymax = value, x = season, colour = value), scores[value > 0],
         size = 0.5) +
  geom_point(aes(y = value, x = season, colour = value), scores[value > 0], size = 0.8) +
  geom_col(aes(y = value, x = season, fill = value), scores[value < 0]) +
  annotate(geom = 'linerange', y = 0, xmin = 0.5, xmax = 13.5, size = 0.4) +
  scale_colour_distiller(
    palette = 'Blues', name = NULL, direction = 1, guide = guide_colorbar(order = 2)) +
  scale_fill_distiller(
    palette = 'Reds', name = 'Scores', guide = guide_colorbar(order = 1)) +
  scale_y_continuous(
    labels = skip_label(2),
    breaks = seq(-0.4, 0.8, 0.2)) +
  scale_x_discrete(
    labels = skip_label(3),
    # breaks = c('Jan', 'Apr', 'Jul', 'Oct', 'Ann'),
    expand = c(0.1, 0)) +
  labs(y = 'Scores', x = NULL, tag = 'c)') +
  theme(
    plot.tag.position = c(0.012, 0.95),
    strip.background = element_rect('gray95', NA),
    axis.line = element_blank(),
    axis.ticks.y = element_blank(),
    axis.text.x = element_text(angle = 90),
    panel.grid.major.y = element_line('gray'),
    panel.spacing.y = unit(0.5, 'cm'),
    legend.position = 'top',
    legend.key.height = unit(0.25, 'cm'),
    legend.key.width = unit(0.5, 'cm'))

wrap_plots(g1) + p3 + p4 + plot_layout(ncol = 1, heights = c(1, 0.4, 0.6))
```




```
# ggsave('inst_period.pdf', width = 7, height = 8.5)
```

Figure S1 - Close-up time series comparison

```
p1 <- multi_line_plot(instDT['p1'], c('Qrec', 'Qobs'),
                      3, title = 'Ping', colour = c('darkorange', 'steelblue'))
p2 <- multi_line_plot(instDT['w4'], c('Qrec', 'Qobs'),
                      1, title = 'Wang', colour = c('darkorange', 'steelblue'))
p3 <- multi_line_plot(instDT['y17'], c('Qrec', 'Qobs'),
                      1, title = 'Yom', colour = c('darkorange', 'steelblue'))
p4 <- multi_line_plot(instDT['n1'], c('Qrec', 'Qobs'),
                      3, title = 'Nan', colour = c('darkorange', 'steelblue'))

pl <- p1 + p2 + p3 + p4 &
  theme(plot.title = element_text(color = 'firebrick'))
pl + plot_layout(ncol = 1, heights = c(3, 1, 1, 3), guides = 'collect')
```

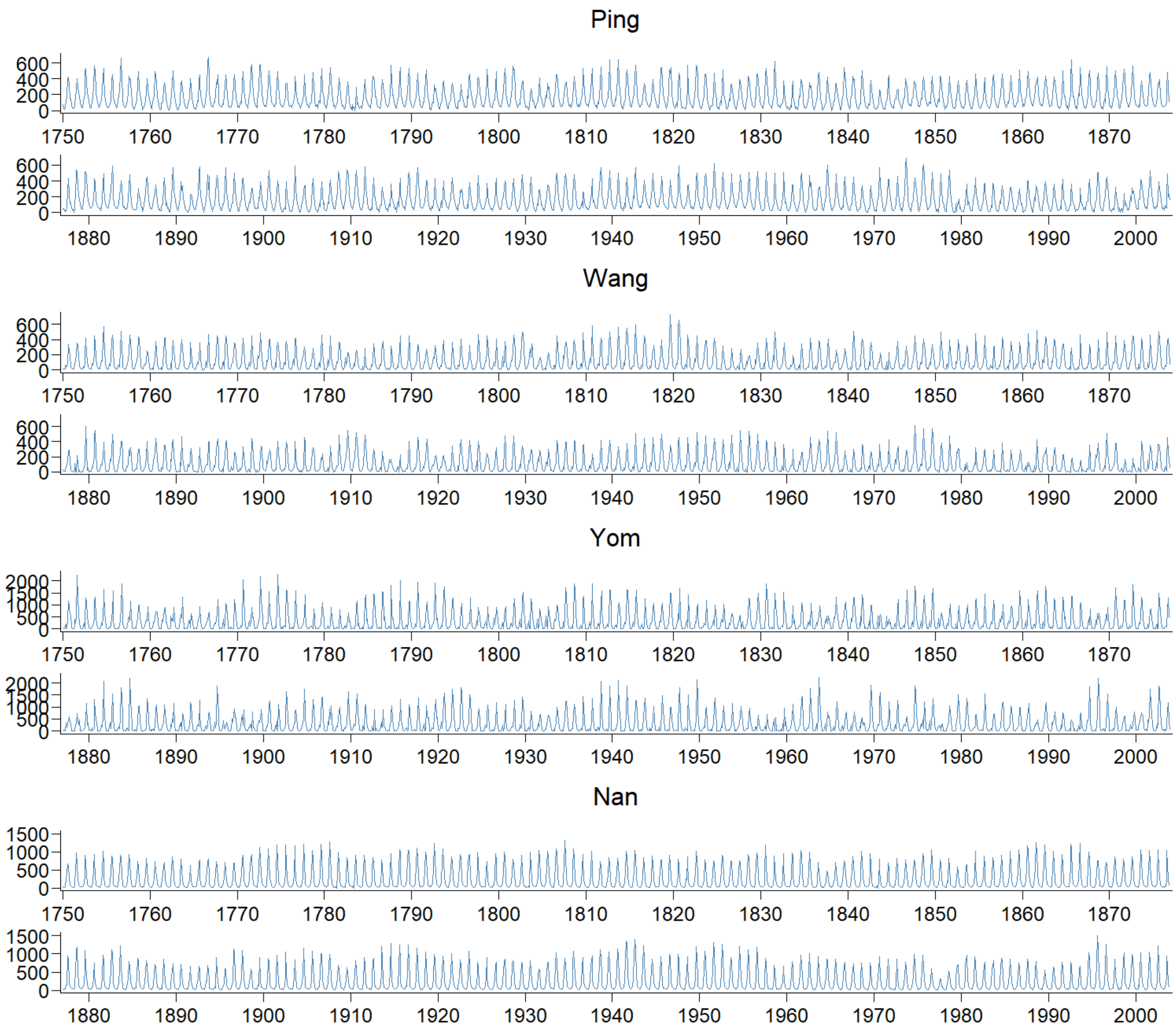


```
# ggsave('inst_period_zoom.pdf', width = 7, height = 8)
```

Full monthly time series

Figure S2

```
nr <- 2
p1 <- multi_line_plot(rec['p1'],  'Q', nr, title = 'Ping', colour = 'steelblue') +
  scale_x_continuous(expand = c(0, 0.25), breaks = seq(1750, 2000, 10))
p2 <- multi_line_plot(rec['w4'],  'Q', nr, title = 'Wang', colour = 'steelblue') +
  scale_x_continuous(expand = c(0, 0.25), breaks = seq(1750, 2000, 10))
p3 <- multi_line_plot(rec['y17'], 'Q', nr, title = 'Yom',  colour = 'steelblue') +
  scale_x_continuous(expand = c(0, 0.25), breaks = seq(1750, 2000, 10))
p4 <- multi_line_plot(rec['n1'],  'Q', nr, title = 'Nan',  colour = 'steelblue') +
  scale_x_continuous(expand = c(0, 0.25), breaks = seq(1750, 2000, 10))

p1 + p2 + p3 + p4 + plot_layout(ncol = 1)
```

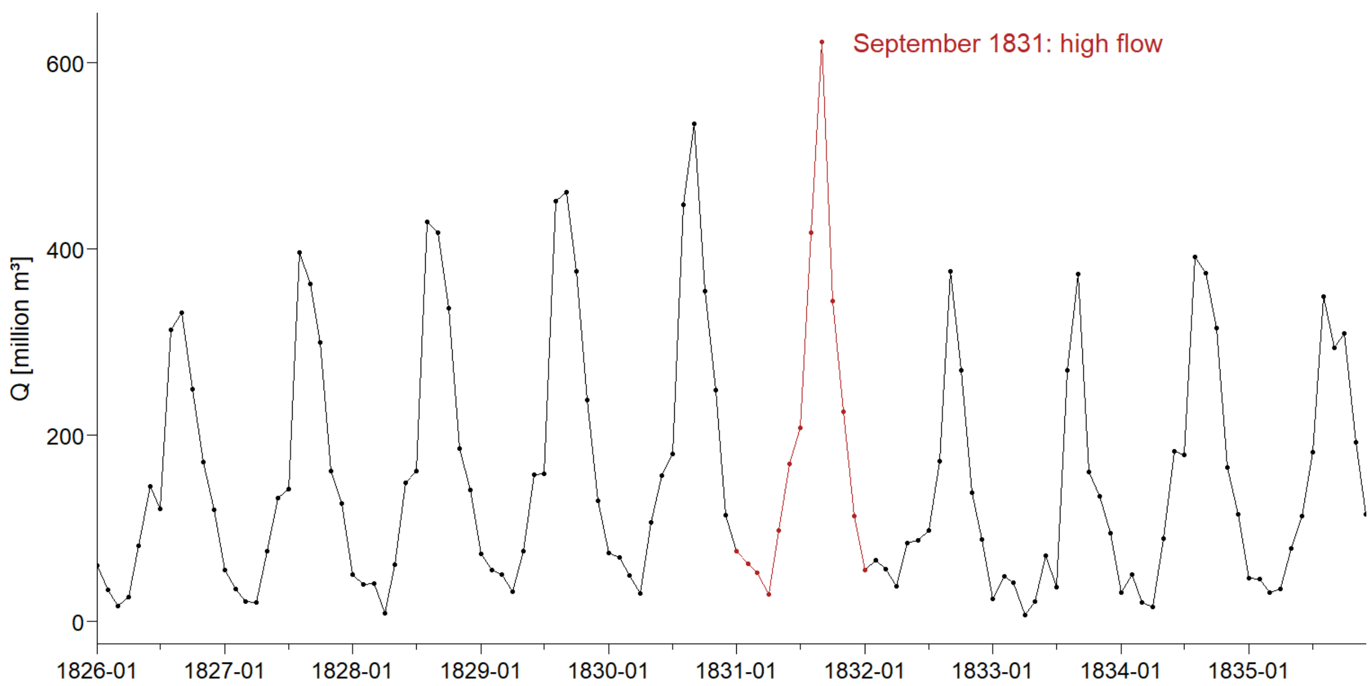


The 1831 flood in the Ping

Wasson et al. (2021) reported a flood in 1831 that caused floodplain stripping along the Ping River, consistent with historical record. Our reconstruction shows extreme high flow in September 1831; this was the seventh highest monthly flow among the 3,048 months in the reconstruction.

Figure S3

```r
p1sub <- rec['p1'
           ][year %in% 1826:1835
           ][, date := as.IDate(paste0(year, '-', sprintf('%02d', month), '-01'))]
ggplot(p1sub) +
  geom_line(aes(date, Q)) +
  geom_point(aes(date, Q), size = 0.5) +
  geom_line(aes(date, Q),
            p1sub[year == 1831 | (year == 1832 & month == 1)],
            color = 'firebrick') +
  geom_point(aes(date, Q),
            p1sub[year == 1831 | (year == 1832 & month == 1)],
            size = 0.5,
            color = 'firebrick') +
  geom_text(aes(date, Q, label = 'September 1831: high flow'),
            p1sub[year == 1831 & month == 9],
            hjust = -0.1,
            color = 'firebrick') +
  scale_x_date(
    date_breaks = '1 year',
    date_minor_breaks = '6 months',
    date_labels = '%Y-%m',
    guide = guide_prism_minor(),
    expand = c(0, 0)) +
  labs(x = NULL, y = 'Q [million m\u00b3]')
```




```r
# ggsave('Ping_1831.pdf', width = 7, height = 4)
```

Ranking of September 1831 flood

```
rec['p1'][order(-Q)][1:10]
```

```
##     station year        Q month month2
##  1:      p1 1973 688.5703     9    Sep
##  2:      p1 1766 665.5233     9    Sep
##  3:      p1 1756 660.8500     9    Sep
##  4:      p1 1813 643.4400     9    Sep
##  5:      p1 1812 641.8619     9    Sep
##  6:      p1 1865 637.7527     9    Sep
##  7:      p1 1831 622.7123     9    Sep
##  8:      p1 1951 618.5068     9    Sep
##  9:      p1 1766 610.2944     8    Aug
## 10:      p1 1975 603.7446     9    Sep
```

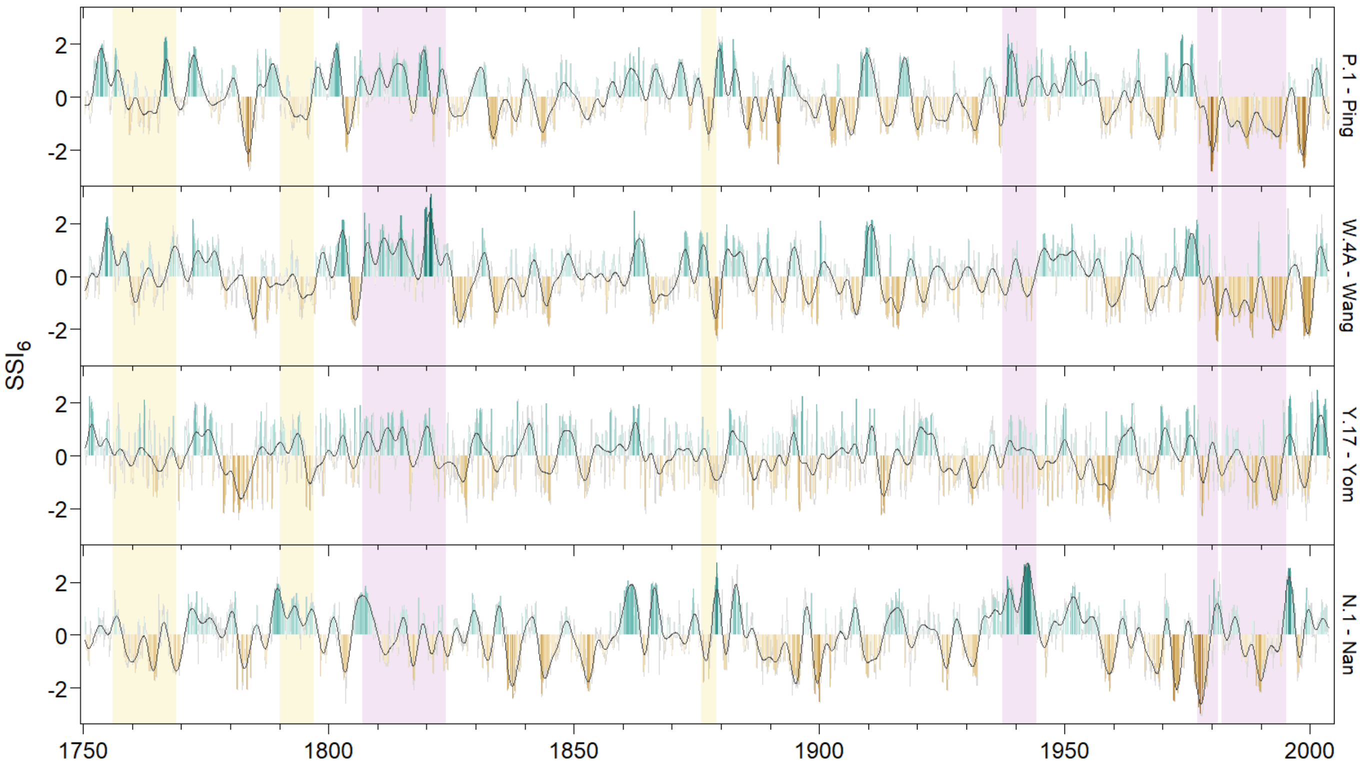
Standardized streamflow index

Calculating SSI

```r
rec[, ':='(ssi1  = c(spei(Q, 1)$fitted),
           ssi6  = c(spei(Q, 6)$fitted),
           ssi12 = c(spei(Q, 12)$fitted)),
    by = station]

ssi6 <- rec[!is.na(ssi6), .(station, year, month, ssi = ssi6)]
ssi6[, lp3 := pass.filt(ssi, 36)]
ssi6[, lp20 := pass.filt(ssi, 240)]
ssi6[, type := classify_events(ssi), by = station]

# Reusable function to plot the indices
plot_ssi <- function(ssiDT) {
  shaded <- data.table(
  start = c(1937 + 3/12, 1982       , 1977       , 1807),
  final = c(1944 + 3/12, 1995 + 3/12, 1981 + 3/12, 1824))
  ggplot(ssiDT) +
    geom_rect(
      aes(xmin = start, xmax = final, ymin = -Inf, ymax = Inf),
      shaded, fill = 'magenta4', alpha = 0.1) +
    geom_rect(
      aes(xmin = start, xmax = final + 1, ymin = -Inf, ymax = Inf),
      mgd, fill = 'khaki', alpha = 0.3) +
    geom_line(aes(year + (month - 1) / 12, ssi), color = 'gray85', size = 0.1) +
    geom_col(aes(year + (month - 1) / 12, ssi, fill = ssi), show.legend = FALSE) +
    geom_line(aes(year + (month - 1) / 12, lp3), color = 'gray30') +
    facet_wrap(vars(station), ncol = 1, strip.position = 'right',
               labeller = as_labeller(stnLab)) +
    scale_x_continuous(
      expand = c(0, 1),
      breaks = seq(1750, 2000, 50),
      minor_breaks = seq(1750, 2000, 10),
      guide = guide_prism_minor()) +
    scale_fill_distiller(
      palette = 'BrBG',
      direction = 1,
      limits = abs_range(ssiDT$ssi)) +
    labs(x = NULL, y = 'SSI<sub>6</sub>') +
    panel_border('black', 0.2) +
    annotation_ticks(sides = 't', type = 'both', size = 0.2) +
    theme(
      panel.spacing.y = unit(0, 'cm'),
      axis.title.y.left = element_markdown(),
      legend.box.margin = margin(),
      legend.position = 'none')
}
```

Figure 3

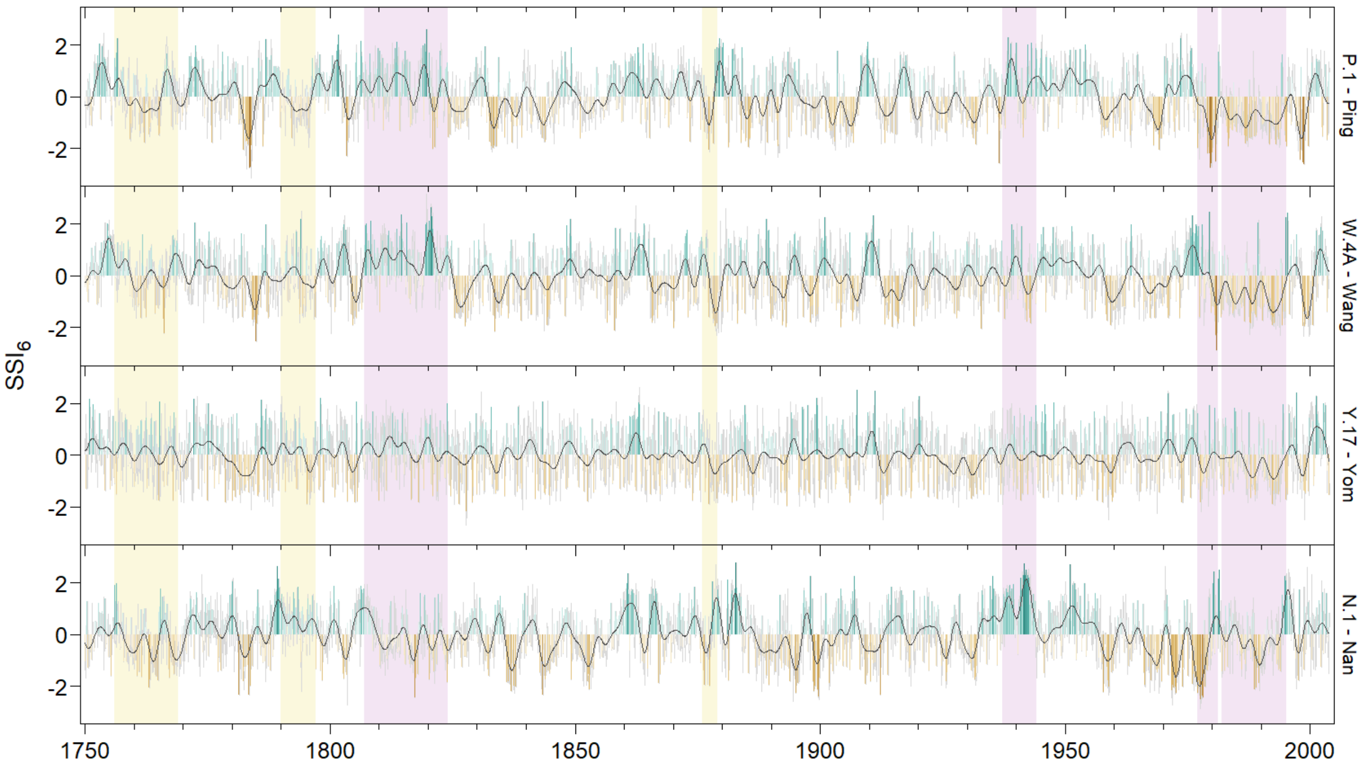
```r
plot_ssi(ssi6)
```




```
# ggsave('ssi.pdf', width = 7, height = 4.5)
```

Figure S4 for SSI_1

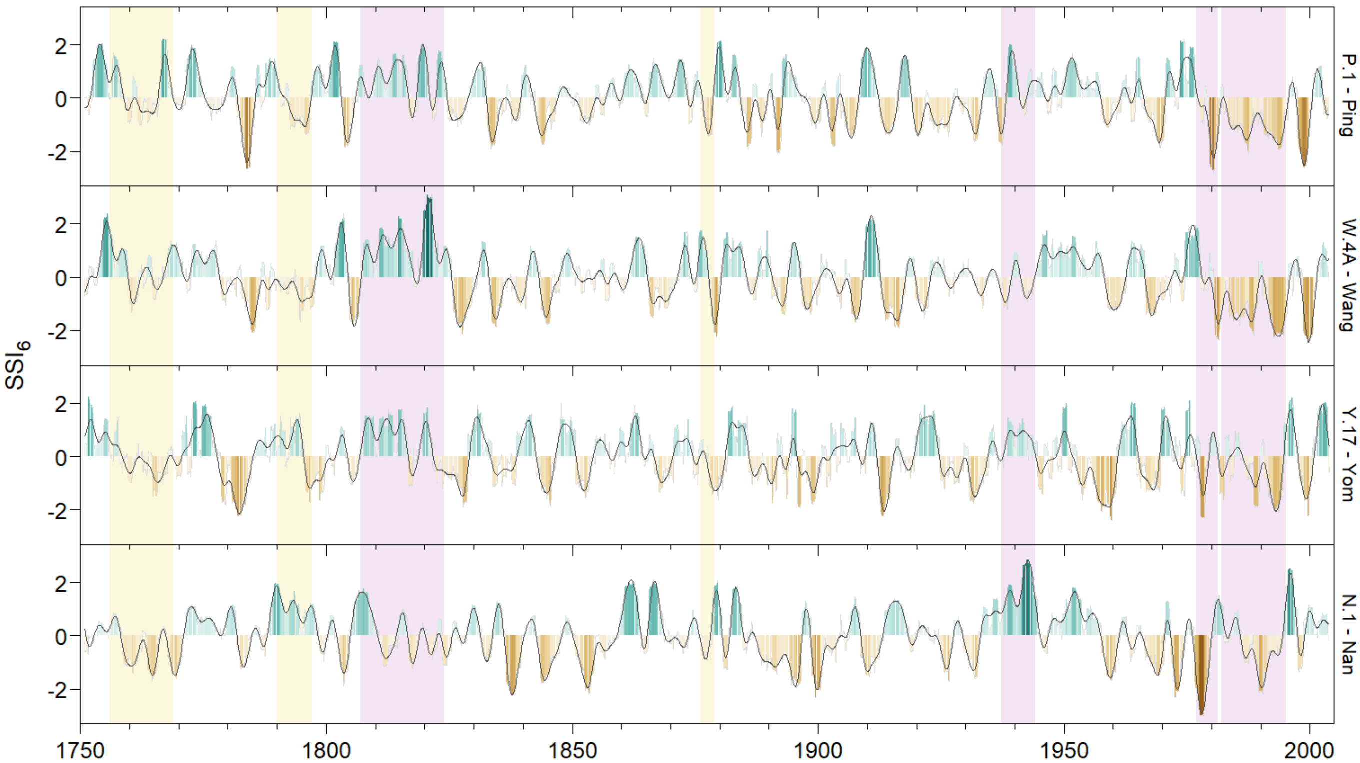
```
ssi1 <- rec[!is.na(ssi1), .(station, year, month, ssi = ssi1)]
ssi1[, lp3 := pass.filt(ssi, 36)]
ssi1[, lp20 := pass.filt(ssi, 240)]
ssi1[, type := classify_events(ssi), by = station]

plot_ssi(ssi1)
```




```
# ggsave('ssi1.pdf', width = 7, height = 4.5)
```

Figure S5 for SSI_{12}

```
ssi12 <- rec[!is.na(ssi12), .(station, year, month, ssi = ssi12)]
ssi12[, lp3 := pass.filt(ssi, 36)]
ssi12[, lp20 := pass.filt(ssi, 240)]
ssi12[, type := classify_events(ssi), by = station]

plot_ssi(ssi12)
```




```
# ggsave('ssi12.pdf', width = 7, height = 4.5)
```

Wet season timing

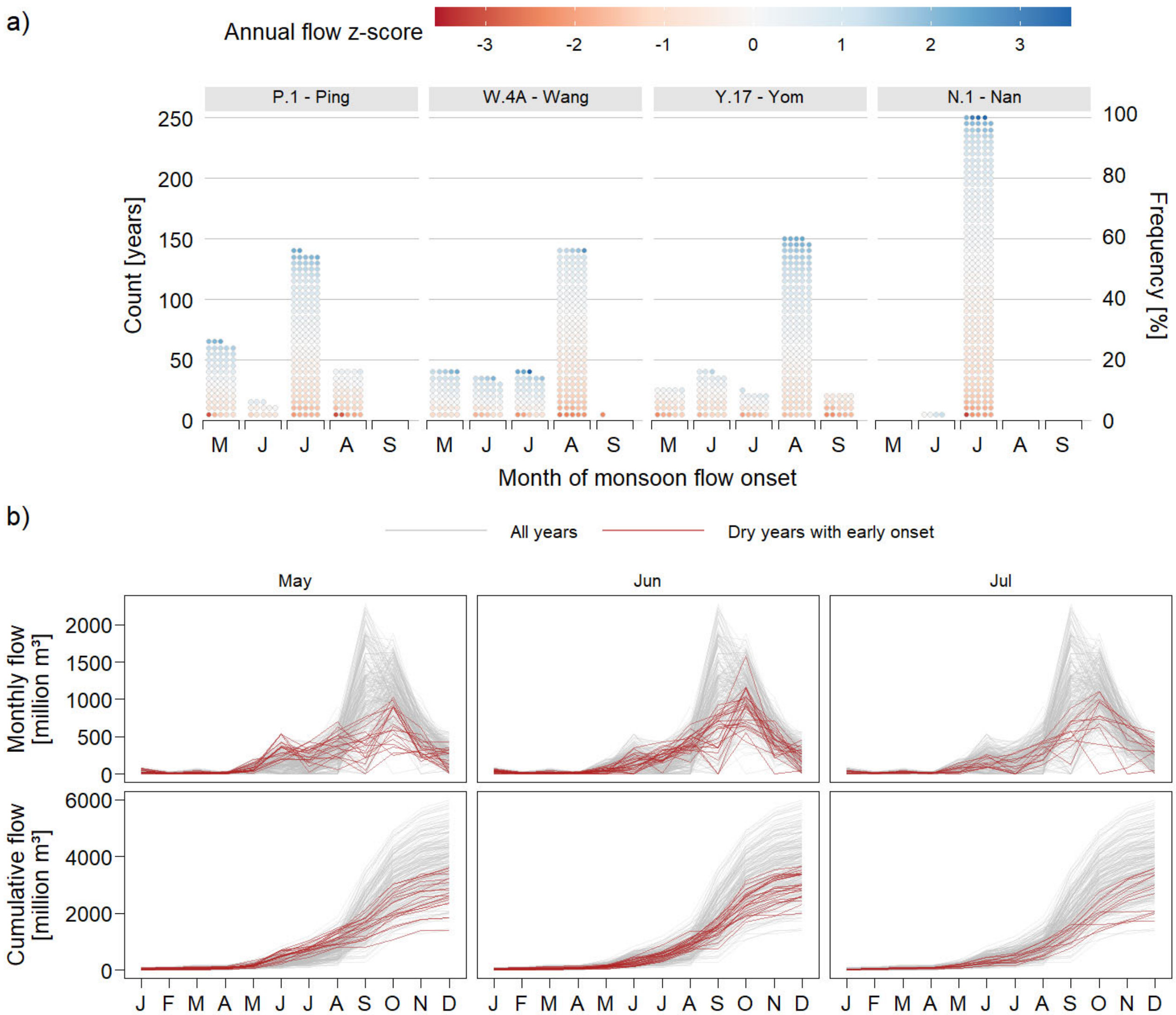
```
recAnn <- rec[, .(Qa = sum(Q)), by = .(station, year)]
recAnn[, za := standardize(Qa), by = station]
rec[, Qcumu := cumsum(Q), by = .(station, year)]
peakStart <- rec[, .(
  start = {
    ssq <- sapply(5:10, function(m) {
      x1 <- matrix(c(rep(1, m), 1:m), ncol = 2)
      x2 <- matrix(c(rep(1, 12 - m), (m+1):12), ncol = 2)
      y1 <- .SD[1:m, Qcumu]
      y2 <- .SD[(m+1):12, Qcumu]
      fit1 <- .lm.fit(x1, y1)$residuals
      fit2 <- .lm.fit(x2, y2)$residuals
      sum(fit1^2) + sum(fit2^2)
    })
    which.min(ssq) + 4
  }), by = .(station, year)]

recShift <- copy(rec)
recShift[month %in% 1:2, year := year - 1]
recShift[month %in% 1:2, month := month + 12]

peakEnd <- recShift[year %in% 1750:2002, .(
  end = {
    s <- .BY$station
    y <- .BY$year
    stt <- peakStart[station == s & year == y, start]
    ssq <- sapply((stt+1):12, function(m) {
      t1 <- stt:m
      t2 <- (m+1):13
      x1 <- matrix(c(rep(1, length(t1)), t1), ncol = 2)
      x2 <- matrix(c(rep(1, length(t2)), t2), ncol = 2)
      y1 <- .SD[month %in% t1, Qcumu]
      y2 <- .SD[month %in% t2, Qcumu]
      fit1 <- .lm.fit(x1, y1)$residuals
      fit2 <- .lm.fit(x2, y2)$residuals
      sum(fit1^2) + sum(fit2^2)
    })
    which.min(ssq) + stt - 1
  }), by = .(station, year)]

peakSeason <- merge(peakStart, peakEnd, by = c('station', 'year'))

peakSeason <- merge(recAnn, peakSeason, by = c('station', 'year')
          )[, .SD[order(za)], by = .(station, start)
          ][, plotOrder := 1:.N, by = .(station, start)
          ][, startChar := factor(month.abb[start], month.abb[4:9])]

nFiles <- 5
peakSeason[, c('file', 'rank') := {
  nRanks <- ceiling(.N / nFiles)
  list(rep(1:nFiles, nRanks)[1:.N],
```

```
      rep(1:nRanks, each = nFiles)[1:.N])
}, by = .(station, start)]

span <- range(peakSeason$start)
spanChar <- month.abb[span[1]:span[2]]
nSkips <- 2
xLevels <- c(sapply(spanChar, function(x) paste0(x, 1:(nFiles + nSkips))))
peakSeason[, startChar2 := factor(paste0(month.abb[start], file), xLevels)]

Qcumu <- rec[, .(station, year, month, month2, Q, Qcumu)
           ][peakStart, on = c('station', 'year')
           ][recAnn, on = c('station', 'year')]
Qcumu2 <- copy(Qcumu[station == 'y17' & start < 8 & za < 0]
             )[, start2 := factor(month.abb[start], month.abb[5:7])]
```

Figure 4

```
p1 <- ggplot(peakSeason) +
  geom_point(
    aes(startChar2, rank, fill = za),
    shape = 21, size = 1.2, colour = 'gray85',
    stroke = 0.1) +
  scale_fill_distiller(
    name = 'Annual flow z-score',
    palette = 'RdBu',
    breaks = -3:3,
    limits = abs_range(peakSeason$za),
    direction = 1) +
  facet_wrap(
    vars(station),
    nrow = 1,
    labeller = as_labeller(stnLab)) +
  scale_x_discrete(
    guide = guide_prism_bracket(),
    breaks = paste0(spanChar, '3'),
    labels = function(x) substr(x, 1, 1),
    drop = FALSE) +
  scale_y_continuous(
    name = 'Count [years]',
    expand = expansion(0, c(0, 1)),
    limits = c(0, 250 / nFiles),
    labels = function(x) x * nFiles,
    sec.axis = sec_axis(
      trans = ~ . * nFiles / 2.54,
      breaks = seq(0, 100, 20),
      name = 'Frequency [%]')) +
  labs(x = 'Month of monsoon flow onset', tag = 'a)') +
  theme(
    legend.position = 'top',
    legend.title = element_text(),
    legend.key.width = unit(2, 'cm'),
    legend.key.height = unit(0.3, 'cm'),
    panel.grid.major.y = element_line('gray', 0.2),
    plot.tag.position = c(0.01, 0.95),
    strip.background = element_rect('gray90', NA),
    axis.line.y = element_blank(),
    axis.ticks.y = element_blank()) +
  coord_equal()
p2 <- ggplot(Qcumu[station == 'y17']) +
  geom_line(
    aes(month2, Q, group = year, color = 'All years'), alpha = 0.25) +
  geom_line(
    aes(month2, Q, group = year, color = 'Dry years with early onset'),
    Qcumu2, alpha = 0.5) +
  facet_wrap(vars(start2)) +
  scale_x_discrete(labels = monthLabShort) +
  scale_color_manual(values = c('gray', 'firebrick')) +
  labs(x = NULL, y = 'Monthly flow\n[million m\u00b3]', tag = 'b)') +
```

```r
  guides(color = guide_legend(override.aes = list(size = 0.4))) +
  panel_border('black', 0.2) +
  theme(
    legend.position = 'top',
    legend.key.width = unit(2, 'cm'),
    legend.box.margin = margin(b = -0.2, unit = 'cm'),
    plot.margin = margin(b = 0),
    plot.tag.position = c(0.01, 0.95),
    axis.ticks.x = element_blank(),
    axis.text.x = element_blank())
p3 <- ggplot(Qcumu[station == 'y17']) +
  geom_line(
    aes(month2, Qcumu, group = year, color = 'All years'), alpha = 0.25) +
  geom_line(
    aes(month2, Qcumu, group = year, color = 'Dry years with early onset'),
    Qcumu2, alpha = 0.5) +
  facet_wrap(vars(start2)) +
  scale_x_discrete(labels = monthLabShort) +
  scale_color_manual(values = c('gray', 'firebrick')) +
  labs(x = NULL, y = 'Cumulative flow\n[million m\u00b3]') +
  panel_border('black', 0.2) +
  theme(
    legend.position = 'none',
    strip.text = element_blank(),
    plot.margin = margin(t = 0))
p1 / p2 / p3 +
  plot_layout(heights = c(1.65, 1, 1))
```




```
# ggsave('peak_and_yom.pdf', width = 7, height = 6.5)
```

References

Josse, Julie, and Francois Husson. 2016. “missMDA: A Package for Handling Missing Values in Multivariate Data Analysis” 70. https://doi.org/10.18637/jss.v070.i01 (https://doi.org/10.18637/jss.v070.i01).

Nguyen, Hung T. T., Stefano Galelli, Chenxi Xu, and Brendan M. Buckley. 2021. “Multi-Proxy, Multi-Season Streamflow Reconstruction with Mass Balance Adjustment.” *Water Resources Research* 57 (8): e2020WR029394. https://doi.org/10.1029/2020WR029394 (https://doi.org/10.1029/2020WR029394).

Wasson, Robert J., Alan Ziegler, Han She Lim, Elisha Teo, Daryl Lam, David Higgitt, Tammy Rittenour, Khairun Nisha Bte Mohamed Ramdzan, Chuah Chong Joon, and Ashok K. Singhvi. 2021. “Episodically Volatile High Energy Non-Cohesive River-Floodplain Systems: New Information from the Ping River, Thailand, and a Global Review.” *Geomorphology*, February, 107658. https://doi.org/10.1016/j.geomorph.2021.107658 (https://doi.org/10.1016/j.geomorph.2021.107658).