

# Expanding window CCM analyses

## Load necessary libraries

`dplyr` gives the pipe operator. `tstools` loads wrappers for the `rEDM` package. `plotly` for plotting.

In [125]:

```
library(dplyr)
library(tstools) # can be installed from https://github.com/kahaaga/tstools using devtools
library(plotly)
library(rEDM)
```

## Load aragonite sea intensity (ASI) and SCOR data

In [78]:

```
dt = data.table::fread("aragonite_SCOR_intensity.txt", sep = ",") %>% as.data.frame
head(dt)
```

| index | stage       | stage_midp | stage_base | stage_top | SCORara   | ara_sea_intensity |
|-------|-------------|------------|------------|-----------|-----------|-------------------|
| 1     | Tremadocian | 481.55     | 485.4      | 477.7     | 0.2322918 | 0.3631756         |
| 2     | Floian      | 473.85     | 477.7      | 470.0     | 0.4026790 | 0.2765494         |
| 3     | Dapingian   | 468.65     | 470.0      | 467.3     | 0.2614996 | 0.2245077         |
| 4     | Darriwilian | 462.85     | 467.3      | 458.4     | 0.2653675 | 0.1997403         |
| 5     | Sandbian    | 455.70     | 458.4      | 453.0     | 0.1695741 | 0.1212638         |
| 6     | Katian      | 449.10     | 453.0      | 445.2     | 0.2111111 | 0.1045704         |

In [79]:

```
tail(dt)
```

|    | index | stage        | stage_midp | stage_base | stage_top | SCORara   | ara_sea_intensi |
|----|-------|--------------|------------|------------|-----------|-----------|-----------------|
| 80 | 80    | Serravallian | 12.72500   | 13.820     | 11.6300   | 0.6649996 | 0.8447181       |
| 81 | 81    | Tortonian    | 9.43800    | 11.630     | 7.2460    | 0.7981285 | 0.9244199       |
| 82 | 82    | Messinian    | 6.28950    | 7.246      | 5.3330    | 0.7090742 | 0.9902333       |
| 83 | 83    | Zanclean     | 4.46650    | 5.333      | 3.6000    | 0.7691805 | 1.0000000       |
| 84 | 84    | Piacenzian   | 3.09000    | 3.600      | 2.5800    | 0.8567812 | 0.9732759       |
| 85 | 85    | Pleistocene  | 1.29585    | 2.580      | 0.0117    | 0.8158894 | 0.9201933       |

## Expanding window analyses

We want windows all starting at the first stage (Tremadocian, which starts at 485.4 Ma), expanding in width towards present day, until the last stage (Pleistocene).

In [6]:

```
# We'll let the first expanding window start at the 12th stage (Homerian, starting at 427.4 Ma).
# This is likely too few points for CCM to give meaningful results, but that will be taken care
# of by the convergence test (and, if not, the surrogate test).
timeidx_stops <- seq(12, max(dt$index), 1)
timeidx_start <- rep(min(dt$index), n = length(timeidx_stops))
timeintervals <- data.frame(timeidx_start, timeidx_stops)
head(timeintervals)
```

| timeidx_start | timeidx_stops |
|---------------|---------------|
| 1             | 12            |
| 1             | 13            |
| 1             | 14            |
| 1             | 15            |
| 1             | 16            |
| 1             | 17            |

In [8]:

```
tail(timeintervals)
```

|    | timeidx_start | timeidx_stops |
|----|---------------|---------------|
| 69 | 1             | 80            |
| 70 | 1             | 81            |
| 71 | 1             | 82            |
| 72 | 1             | 83            |
| 73 | 1             | 84            |
| 74 | 1             | 85            |

## Perform CCM analysis

We will now perform CCM separately on each time series window. What CCM does it to see how well the assumed response (SCOR) is able to predict the putative forcing (ASI). Due to the limited dataset, we use embedding dimension 2 and embedding lag 1.

We'll perform two types of analyses: one without the exclusion of temporal neighbours in the predictions, and one excluding one temporal neighbour to prevent bias due to autocorrelation. In the paper, we present the CCM analysis without exclusion of temporal neighbours, as the resolution of the data is very coarse and the exclusion of temporal neighbours may lead to a loss of genuine information. We will find that the overall shape of the CCM prediction skill across the expanding windows is roughly the same with and without this correction, changing only the significance level as the number of excluded neighbours increases.

In [80]:

```
# For this notebook, to reduce the time the code takes to run we'll use 100
# samples for both the original and surrogate runs, and use 100 randomly
# shuffled surrogates. For the paper, all values used are 500.
samples = 100
samples_surr = 100
n.surrogates = 100

results <- list()
results_withexclusion <- list()
for (i in 1:nrow(timeintervals)) {
  t1 <- Sys.time()
  start <- dt[timeintervals[i, 1], ]$stage_midp
  stop <- dt[timeintervals[i, 2], ]$stage_midp
  interval <- c(start, stop)
  cat(paste("Interval", i, "/", nrow(timeintervals), "start:", start, "stop: ",
stop, "\n"))

  B <- dt %>% filter(index <= timeintervals[i, 2]) %>% select(index, SCORara, ar
a_sea_intensity)

  # No exclusion of temporaal neighbors
  res <- ccm_lagged(data = B[, 2:3], E = 2, tau = 1,
                    exclusion.radius = 0, lags = 0,
                    samples.original = samples,
                    library.sizes = nrow(B),
                    with.replacement = T,
                    convergence.test = T, n.libsizes.to.check = 30,
                    samples.surrogates = samples_surr, n.surrogates = n.surrogates,
surrogate.methods = "random")
  results[[toString(timeintervals[i, 2])]] <- res

  # Exclude temporal neighbors in radius of 3 (that is, dim + 1) time steps
  res_excl <- ccm_lagged(data = B[, 2:3], E = 2, tau = 1,
                        exclusion.radius = 1, lags = 0,
                        samples.original = samples,
                        library.sizes = nrow(B),
                        with.replacement = T,
                        convergence.test = T, n.libsizes.to.check = 30,
                        samples.surrogates = samples_surr, n.surrogates = n.surrogates,
surrogate.methods = "random")

  results_withexclusion[[toString(timeintervals[i, 2])]] <- res_excl
}
```

Interval 1 / 74 start: 481.55 stop: 428.95  
Interval 2 / 74 start: 481.55 stop: 426.5  
Interval 3 / 74 start: 481.55 stop: 424.3  
Interval 4 / 74 start: 481.55 stop: 421.1  
Interval 5 / 74 start: 481.55 stop: 415  
Interval 6 / 74 start: 481.55 stop: 409.2  
Interval 7 / 74 start: 481.55 stop: 400.45  
Interval 8 / 74 start: 481.55 stop: 390.5  
Interval 9 / 74 start: 481.55 stop: 385.2  
Interval 10 / 74 start: 481.55 stop: 377.45  
Interval 11 / 74 start: 481.55 stop: 365.55  
Interval 12 / 74 start: 481.55 stop: 352.8  
Interval 13 / 74 start: 481.55 stop: 338.8  
Interval 14 / 74 start: 481.55 stop: 327.05  
Interval 15 / 74 start: 481.55 stop: 319.2  
Interval 16 / 74 start: 481.55 stop: 311.1  
Interval 17 / 74 start: 481.55 stop: 305.35  
Interval 18 / 74 start: 481.55 stop: 301.3  
Interval 19 / 74 start: 481.55 stop: 296.95  
Interval 20 / 74 start: 481.55 stop: 292.55  
Interval 21 / 74 start: 481.55 stop: 286.8  
Interval 22 / 74 start: 481.55 stop: 278.225  
Interval 23 / 74 start: 481.55 stop: 270.875  
Interval 24 / 74 start: 481.55 stop: 266.95  
Interval 25 / 74 start: 481.55 stop: 262.1  
Interval 26 / 74 start: 481.55 stop: 256.62  
Interval 27 / 74 start: 481.55 stop: 253.021  
Interval 28 / 74 start: 481.55 stop: 249.551  
Interval 29 / 74 start: 481.55 stop: 244.6  
Interval 30 / 74 start: 481.55 stop: 239.5  
Interval 31 / 74 start: 481.55 stop: 232  
Interval 32 / 74 start: 481.55 stop: 217.9  
Interval 33 / 74 start: 481.55 stop: 205.05  
Interval 34 / 74 start: 481.55 stop: 200.3  
Interval 35 / 74 start: 481.55 stop: 195.05  
Interval 36 / 74 start: 481.55 stop: 186.75  
Interval 37 / 74 start: 481.55 stop: 178.4  
Interval 38 / 74 start: 481.55 stop: 172.2  
Interval 39 / 74 start: 481.55 stop: 169.3  
Interval 40 / 74 start: 481.55 stop: 167.2  
Interval 41 / 74 start: 481.55 stop: 164.8  
Interval 42 / 74 start: 481.55 stop: 160.4  
Interval 43 / 74 start: 481.55 stop: 154.7  
Interval 44 / 74 start: 481.55 stop: 148.55  
Interval 45 / 74 start: 481.55 stop: 142.4  
Interval 46 / 74 start: 481.55 stop: 136.35  
Interval 47 / 74 start: 481.55 stop: 131.15  
Interval 48 / 74 start: 481.55 stop: 127.2  
Interval 49 / 74 start: 481.55 stop: 119  
Interval 50 / 74 start: 481.55 stop: 106.75  
Interval 51 / 74 start: 481.55 stop: 97.2  
Interval 52 / 74 start: 481.55 stop: 91.85  
Interval 53 / 74 start: 481.55 stop: 88.05  
Interval 54 / 74 start: 481.55 stop: 84.95  
Interval 55 / 74 start: 481.55 stop: 77.85  
Interval 56 / 74 start: 481.55 stop: 69.05  
Interval 57 / 74 start: 481.55 stop: 63.8  
Interval 58 / 74 start: 481.55 stop: 60.4  
Interval 59 / 74 start: 481.55 stop: 57.6  
Interval 60 / 74 start: 481.55 stop: 51.9  
Interval 61 / 74 start: 481.55 stop: 44.5

|                  |               |               |
|------------------|---------------|---------------|
| Interval 62 / 74 | start: 481.55 | stop: 39.5    |
| Interval 63 / 74 | start: 481.55 | stop: 35.85   |
| Interval 64 / 74 | start: 481.55 | stop: 31      |
| Interval 65 / 74 | start: 481.55 | stop: 25.565  |
| Interval 66 / 74 | start: 481.55 | stop: 21.735  |
| Interval 67 / 74 | start: 481.55 | stop: 18.205  |
| Interval 68 / 74 | start: 481.55 | stop: 14.895  |
| Interval 69 / 74 | start: 481.55 | stop: 12.725  |
| Interval 70 / 74 | start: 481.55 | stop: 9.438   |
| Interval 71 / 74 | start: 481.55 | stop: 6.2895  |
| Interval 72 / 74 | start: 481.55 | stop: 4.4665  |
| Interval 73 / 74 | start: 481.55 | stop: 3.09    |
| Interval 74 / 74 | start: 481.55 | stop: 1.29585 |

In [50]:

```
head(results[[1]])
```

| E | tau | lag | num.neighbours | library.column | target.column     | library.size | num.pred | rmse    |
|---|-----|-----|----------------|----------------|-------------------|--------------|----------|---------|
| 2 | 1   | 0   | 3              | SCORara        | ara_sea_intensity | 3            | 11       | 0.0000  |
| 2 | 1   | 0   | 3              | SCORara        | ara_sea_intensity | 3            | 11       | 0.0000  |
| 2 | 1   | 0   | 3              | SCORara        | ara_sea_intensity | 3            | 11       | 0.0000  |
| 2 | 1   | 0   | 3              | SCORara        | ara_sea_intensity | 3            | 11       | 0.0000  |
| 2 | 1   | 0   | 3              | SCORara        | ara_sea_intensity | 3            | 11       | -0.0000 |
| 2 | 1   | 0   | 3              | SCORara        | ara_sea_intensity | 3            | 11       | -0.0000 |

The column 'causal.direction' tells us how to interpret the results (note: this is opposite of the prediction direction).

In [81]:

```
head(results[[1]]$causal.direction)[1]
```

```
'ara_sea_intensity -> SCORara'
```

The 'convergent' column indicates whether the analysis is convergent or not.

In [82]:

```
head(results[[1]])[, c('alpha', 'convergent', 'k', 'a', 'b')]
```

[illegible]

## Convergence plots

We'll now plot CCM skill versus library size separately for each window, and determine whether the analysis is convergent in each case. In the plots below, convergent analyses are plotted as solid lines, and non-convergent analyses are plotted as dotted lines.

### Without exclusion of temporal neighbours

In [107]:

```
colfunc <- colorRampPalette(c("black", "red"))
cs <- colfunc(length(results))
```

In [108]:

```
p <- plot_ly(width = 800, height = 600)

for (i in 1:length(results)) {
  n <- names(results)[i]
  age <- dt[dt$index == as.numeric(as.character(n)), ]$stage_midp[1]
  r <- results[[n]]

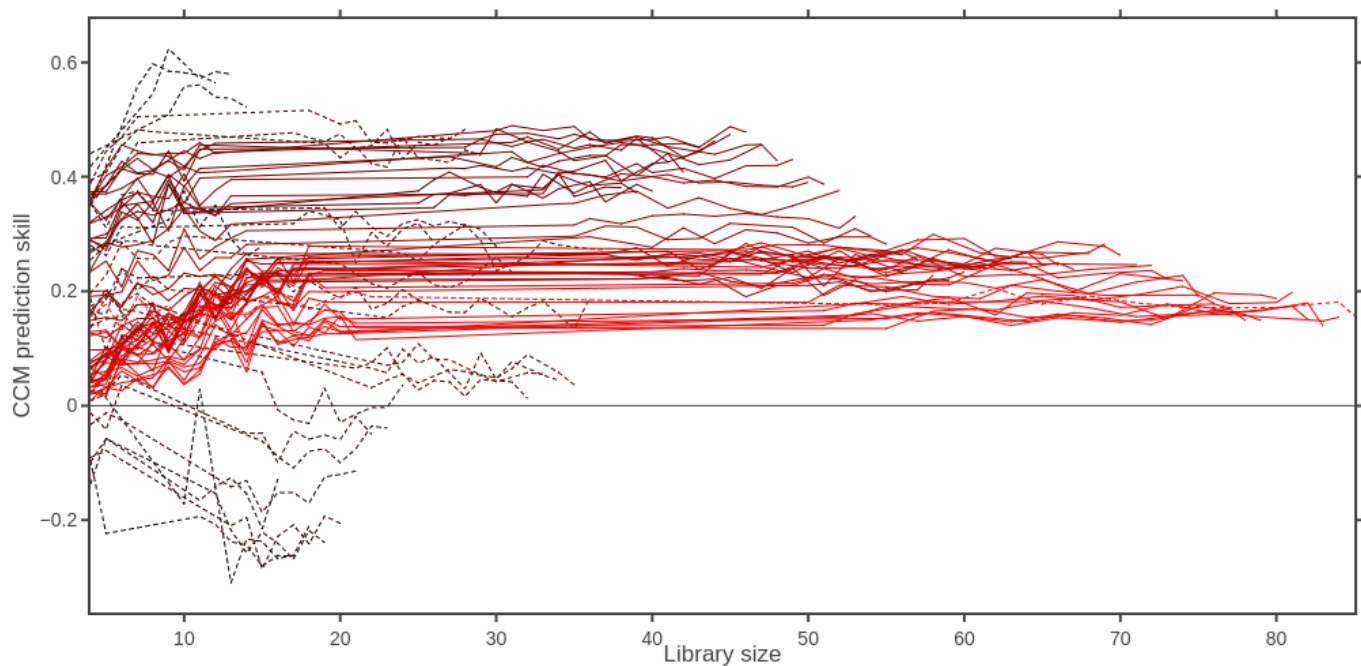
  D <- r %>%
    # We only want to look at the results for the original analysis, not surrogates
    filter(surrogate.index == 0) %>%
    # Group by library size and summarise
    group_by(library.size) %>%
    summarise(rho = median(rho))

  linestyle = ifelse(unique(r$convergent)[1] == 1, "solid", "dot")
  p <- p %>%
    add_lines(data = D, x = ~library.size, y = ~rho, name = age,
              line = list(dash = linestyle, width = 1.0, color = cs[i]),
              showlegend = F)
}
p <- p %>%
  layout(
    xaxis = list(title = "Library size",
                  zeroline = T, showgrid = F,
                  showline = TRUE,
                  mirror = "ticks",
                  linewidth = 2,
                  tickwidth = 2),
    yaxis = list(title = "CCM prediction skill",
                  zeroline = T, showgrid = F,
                  showline = TRUE,
                  mirror = "ticks",
                  linewidth = 2,
                  tickwidth = 2),
    font = list(family = "Helvetica", size = 14)
  )
```

Export the plot:

In [110]:

```
plotly_IMAGE(p, format = "png", out_file = "convergence_plot_noexclusion.png")
```



## With temporal exclusion

In [111]:

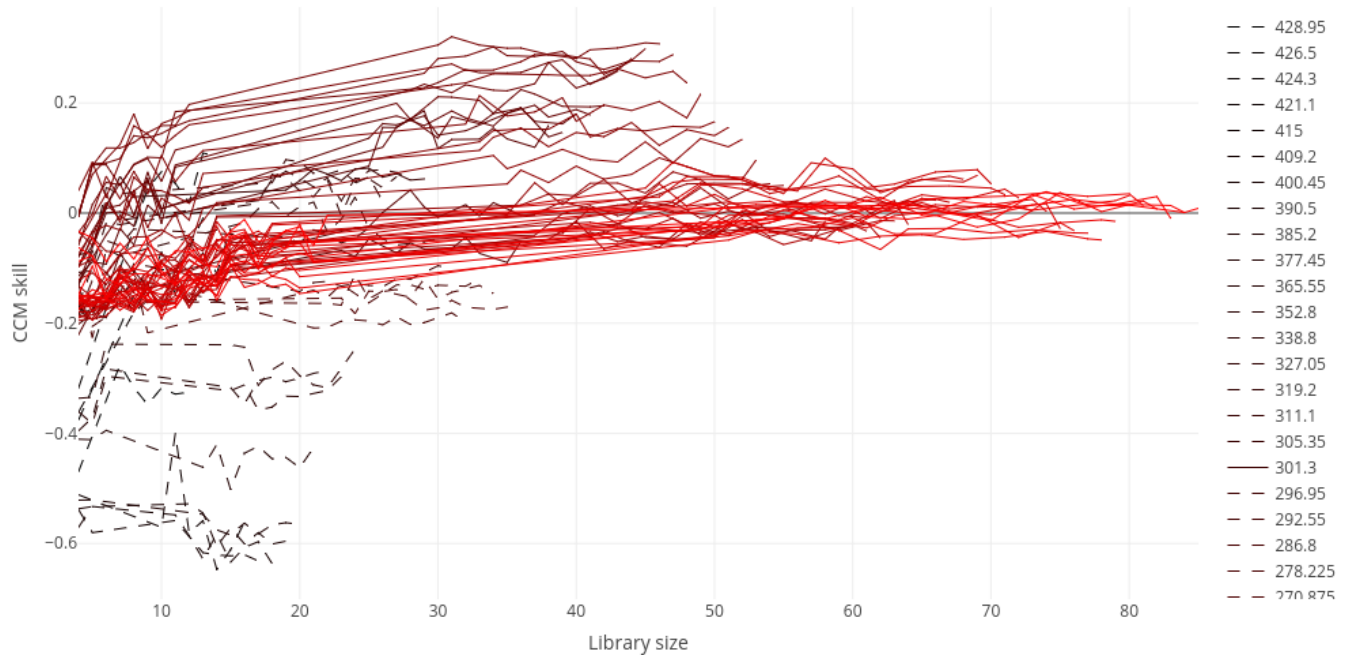
```
p <- plot_ly()
for (i in 1:length(results_withexclusion)) {
  n <- names(results_withexclusion)[i]
  age <- dt[dt$index == as.numeric(as.character(n)), ]$stage_midp[1]
  r <- results_withexclusion[[n]]

  D <- r %>%
    filter(surrogate.index == 0) %>%
    group_by(library.size) %>%
    summarise(rho = median(rho))

  linestyle = ifelse(unique(r$convergent)[1] == 1, "solid", "dash")
  p <- p %>%
    add_lines(data = D, x = ~library.size, y = ~rho, name = age,
              line = list(dash = linestyle, width = 1.0, color = cs[i]))
}
p <- p %>%
  layout(
    xaxis = list(title = "Library size"),
    yaxis = list(title = "CCM skill")
  )
```

In [113]:

```
plotly_IMAGE(p, format = "png", out_file = "convergence_plot_withexclusion1.png")
)
```



## Expanding window summary

Without exclusion of temporal neighbors:

In [114]:

```
age <- vector(length = length(results))
convergent <- vector(length = length(results))
p50 <- vector(length = length(results))
surr_p67 <- vector(length = length(results))
surr_p95 <- vector(length = length(results))
surr_p99 <- vector(length = length(results))

mc <- vector(length = length(results)) # marker color, depends on convergent or
not

summary <- cbind(age, convergent, p50, surr_p67, surr_p95, surr_p99)#, mc)

for (i in 1:length(results)) {
  n <- names(results)[i]
  age <- dt[dt$index == as.numeric(as.character(n)), ]$stage_midp[1]
  maxage <- dt[1, ]$stage_midp[1]
  r <- results[[n]]
  largest_binsize = max(r$library.size)

  # Filter so that we get the results only for the largest binsize
  r_orig = r[r$library.size == largest_binsize & surrogate.index == 0, ]

  # Is the analysis convergent?
  convergent <- unique(r_orig$convergent)[1]
  p50 <- median(r_orig$rho)

  # Filter so that we get the results only for the largest binsize,
  # and only for the surrogates (surrogate.index > 0)
  r_surr = r[r$library.size == largest_binsize & surrogate.index > 0, ]
  surr_p67 = quantile(r_surr$rho, 0.67)
  surr_p95 = quantile(r_surr$rho, 0.95)
  surr_p99 = quantile(r_surr$rho, 0.99)

  summary[i, "age"] <- age
  summary[i, "convergent"] <- convergent
  summary[i, "p50"] <- p50
  summary[i, "surr_p67"] <- surr_p67
  summary[i, "surr_p95"] <- surr_p95
  summary[i, "surr_p99"] <- surr_p99
}
```

In [115]:

```
head(summary)
```

| age    | convergent | p50        | surr_p67     | surr_p95  | surr_p99  |
|--------|------------|------------|--------------|-----------|-----------|
| 428.95 | 0          | 0.5642674  | -0.028232043 | 0.4429538 | 0.6787411 |
| 426.50 | 0          | 0.5793972  | 0.001990166  | 0.3793467 | 0.5289160 |
| 424.30 | 0          | 0.5218753  | 0.008055644  | 0.3585266 | 0.4929859 |
| 421.10 | 0          | 0.2789965  | -0.008389504 | 0.3682583 | 0.5622979 |
| 415.00 | 0          | -0.1284019 | 0.019314020  | 0.3498519 | 0.5271109 |
| 409.20 | 0          | -0.2629493 | 0.033681448  | 0.3845249 | 0.5353877 |

In [116]:

```
summary <- as.data.frame(summary, stringsAsFactors = F)
```

In [117]:

```
head(summary)
```

| age    | convergent | p50        | surr_p67     | surr_p95  | surr_p99  |
|--------|------------|------------|--------------|-----------|-----------|
| 428.95 | 0          | 0.5642674  | -0.028232043 | 0.4429538 | 0.6787411 |
| 426.50 | 0          | 0.5793972  | 0.001990166  | 0.3793467 | 0.5289160 |
| 424.30 | 0          | 0.5218753  | 0.008055644  | 0.3585266 | 0.4929859 |
| 421.10 | 0          | 0.2789965  | -0.008389504 | 0.3682583 | 0.5622979 |
| 415.00 | 0          | -0.1284019 | 0.019314020  | 0.3498519 | 0.5271109 |
| 409.20 | 0          | -0.2629493 | 0.033681448  | 0.3845249 | 0.5353877 |

**With exclusion of temporal neighbors**

In [118]:

```
age <- vector(length = length(results))
convergent <- vector(length = length(results))
p50 <- vector(length = length(results))
surr_p67 <- vector(length = length(results))
surr_p95 <- vector(length = length(results))
surr_p99 <- vector(length = length(results))

summary_withexclusion <- cbind(age, convergent, p50, surr_p67, surr_p95, surr_p99) #, mc)

for (i in 1:length(results_withexclusion)) {
  n <- names(results_withexclusion)[i]
  age <- dt[dt$index == as.numeric(as.character(n)), ]$stage_midp[1]
  maxage <- dt[1, ]$stage_midp[1]
  r <- results_withexclusion[[n]]
  largest_binsize = max(r$library.size)

  # Filter so that we get the results only for the largest binsize
  r_orig = r[r$library.size == largest_binsize & surrogate.index == 0, ]

  # Is the analysis convergent?
  convergent <- unique(r_orig$convergent)[1]
  p50 <- median(r_orig$rho)

  # Filter so that we get the results only for the largest binsize,
  # and only for the surrogates (surrogate.index > 0)
  r_surr = r[r$library.size == largest_binsize & surrogate.index > 0, ]
  surr_p67 = quantile(r_surr$rho, 0.67)
  surr_p95 = quantile(r_surr$rho, 0.95)
  surr_p99 = quantile(r_surr$rho, 0.99)

  summary_withexclusion[i, "age"] <- age
  summary_withexclusion[i, "convergent"] <- convergent
  summary_withexclusion[i, "p50"] <- p50
  summary_withexclusion[i, "surr_p67"] <- surr_p67
  summary_withexclusion[i, "surr_p95"] <- surr_p95
  summary_withexclusion[i, "surr_p99"] <- surr_p99
}

summary_withexclusion <- as.data.frame(summary_withexclusion, stringsAsFactors = F)
```

## Plot the results

We'll go over each analysis and plot the results one by one, coloring each value as transparent grey if the analysis is not convergent, and black if it is convergent (convergence is a necessary requirement for CCM results to be valid).

### Without exclusion of temporal neighbors

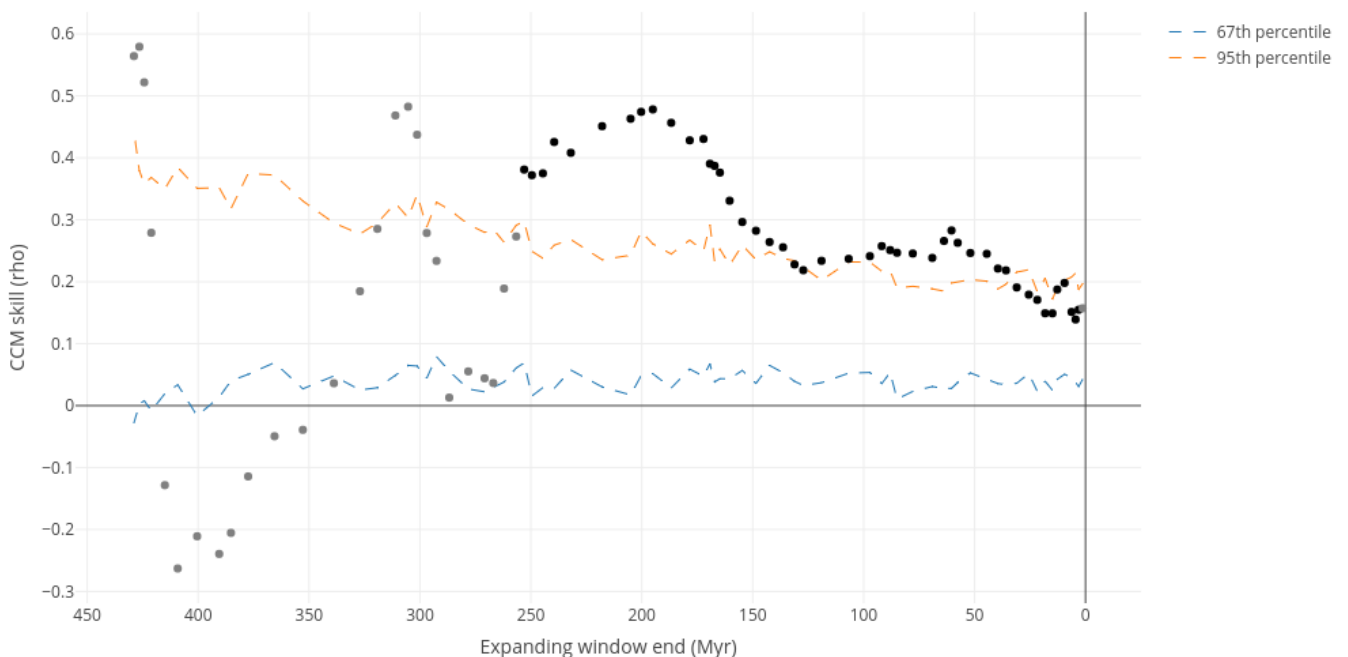
In [119]:

```
p_noexclusion <- plot_ly() %>%
  add_lines(data = summary, x = ~age, y=~surr_p67, name = "67th percentile",
            line = list(width = 1, dash = "dash")) %>%
  add_lines(data = summary, x = ~age, y=~surr_p95, name = "95th percentile",
            line = list(width = 1, dash = "dash")) %>%
  # Dummy point, invisible, to force axes to desired range
  #add_lines(x = ~483, y = 0, lines = list(alpha = 0), showlegend = F) %>%
  layout(
    yaxis = list(title = "CCM skill (rho)",
                  xaxis = list(title = "Expanding window end (Myr)", autorange = "reverse
d", range = c(482, 0))
  )

for (i in 1:nrow(summary)) {
  age <- summary[i, "age"]
  p50 <- summary[i, "p50"]
  conv <- summary[i, "convergent"]
  p_noexclusion <- p_noexclusion %>%
    add_markers(x = age, y=p50, showlegend = F,
               marker = list(
                 color = ifelse(conv == 1, "black", "grey"),
                 alpha = ifelse(conv == 1, 1, 0.5)
               )
    )
}
```

In [120]:

```
plotly_IMAGE(p_noexclusion, format = "png", out_file = "expanding_window_plot_no
exclusion.png")
```



**With exclusion of temporal neighbors**

In [122]:

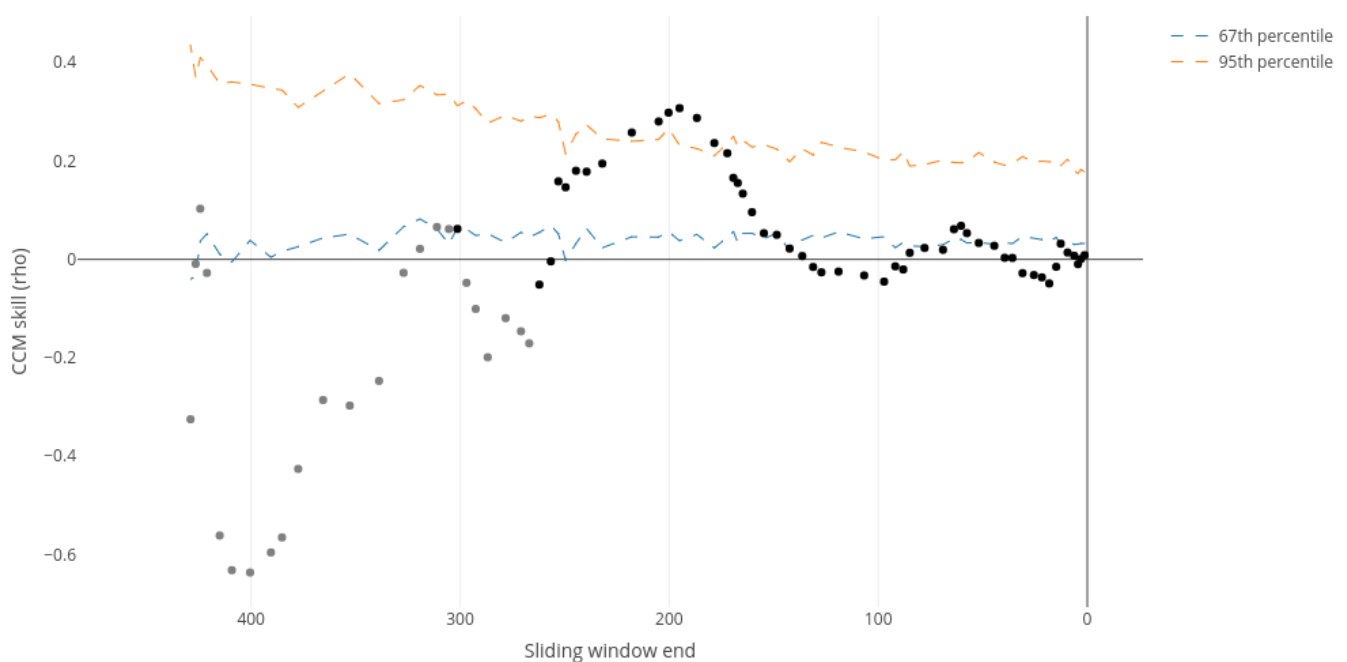
```
p_withexclusion <- plot_ly() %>%
  add_lines(data = summary_withexclusion, x = ~age, y=~surr_p67, name = "67th
percentile",
    line = list(width = 1, dash = "dash")) %>%
  add_lines(data = summary_withexclusion, x = ~age, y=~surr_p95, name = "95th
percentile",
    line = list(width = 1, dash = "dash")) %>%
  # Dummy point, invisible, to force axes to desired range
  add_lines(x = ~483, y = 0, lines = list(alpha = 0), showlegend = F) %>%
  layout(
    xaxis = list(title = "Sliding window end", autorange = "reversed", range
= c(482, 0))
  )

for (i in 1:nrow(summary)) {
  age <- summary_withexclusion[i, "age"]
  p50 <- summary_withexclusion[i, "p50"]
  conv <- summary_withexclusion[i, "convergent"]
  p_withexclusion <- p_withexclusion %>%
    add_markers(x = age, y=p50, showlegend = F,
      marker = list(
        color = ifelse(conv == 1, "black", "grey"),
        alpha = ifelse(conv == 1, 1, 0.5)
      )
    )
}

p_withexclusion <- p_withexclusion %>%
  layout(yaxis = list(title = "CCM skill (rho)", showgrid = F))
```

In [123]:

```
plotly_IMAGE(p_withexclusion, format = "png", out_file = "expanding_window_plot_
withexclusion1.png")
```



## Plot results along with time series

In [103]:

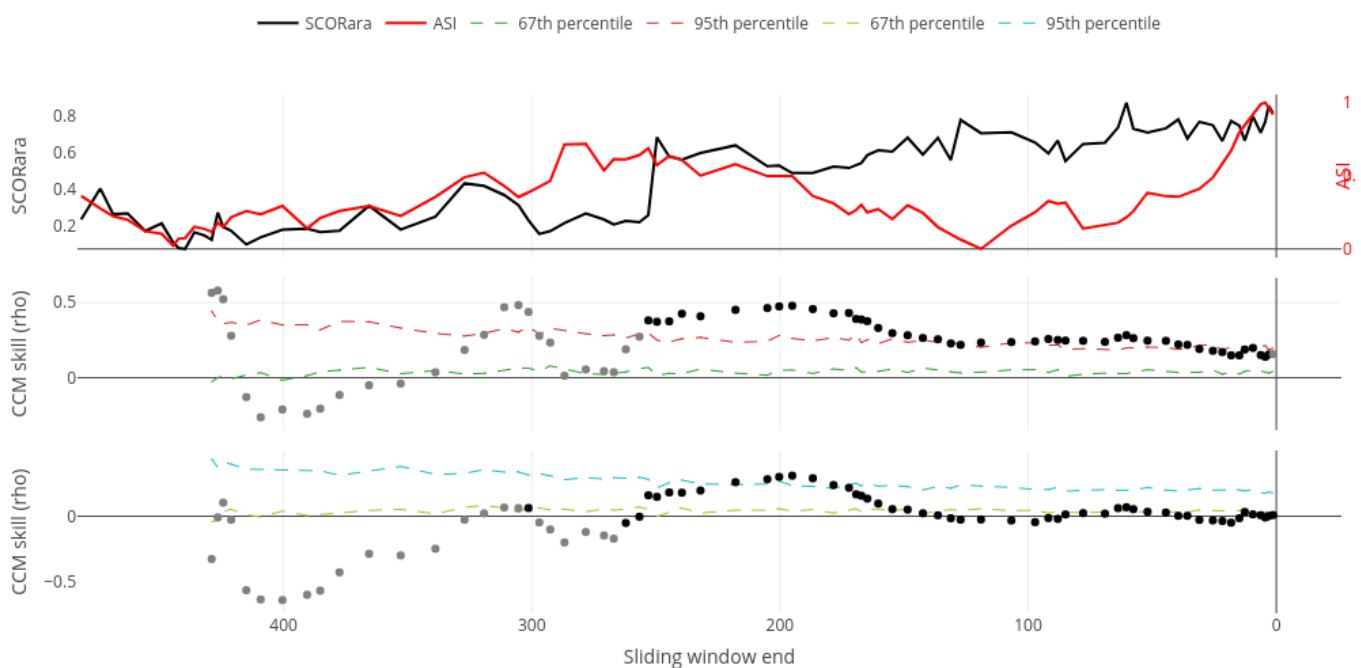
```
ay <- list(
  tickfont = list(color = "red"),
  titlefont = list(color = "red"),
  overlaying = "y",
  side = "right",
  title = "ASI",
  showgrid = F
)

p_ts <- plot_ly() %>%
  add_lines(data = dt, x = ~stage_midp, y = ~SCORara, name = "SCORara",
            line = list(color = "black")) %>%
  add_lines(data = dt, x = ~stage_midp, y = ~ara_sea_intensity, name = "ASI",
            yaxis = "y2",
            line = list(color = "red")) %>%
  layout(
    yaxis = list("Value", showgrid = F), yaxis2 = ay
  )

ts_plot <- subplot(p_ts, p_noexclusion, p_withexclusion,
  nrow = 3, shareX = T, titleY = T) %>%
  layout(
    legend = list(orientation = "h",
                  yanchor = "bottom", y = 1.1,
                  xanchor = "center", x = 0.5),
    height = 700,
    width = 500
  )
```

In [124]:

```
plotly_IMAGE(ts_plot, format = "png", out_file = "timeseries_plot.png")
```



The number of excluded temporal neighbors changes the overall CCM prediction skill and but gives the same pattern. In the main paper, we present the analysis for the analysis with no exclusion of temporal neighbors.

## Export results

In [519]:

```
write.table(x = summary, file = "ccm_summary.txt")
```