Supplementary information for NAU-23 XRF analyses.

Paul Lincoln

Contents

List of Figures

This document details the R code used to run standard analyses for multivariate XRF data (e.g. central log-ratios, PCAs, clustering) in RStudio for the ITRAX XRF from Nautajarvi in Finland. The document is split into three sections. First, the code used format and analyse the XRF data are presented in Section 1. Code used to plot results are shown in Section 2 and supplementary data and figures are presented in Section 3.

1 Statistical analyses

First, clear the work environment to remove residual objects from any previous analyses.

```
rm(list = ls(all.names=T))
```

1.1 Load required libraries and data

Load in the libraries required to run the analyses

```
library(pacman)

p_load(dplyr,
       tidyverse,
       compositions,
       readxl,
       zoo,
       corrplot,
       robCompositions,
       car,
       FactoMineR,
       factoextra,
       janitor,
       ggrepel,
       dendextend,
       jpeg,
       gttable,
       NbClust,
       gt,
       ComplexHeatmap,
       grid)
```

The following code reads in the replicate averaged 0.2mm resolution XRF data into an R object called 'XRF', replicate variances at 0.2 and 0.4mm into an object called 'variance', the Nautajarvi varve thicknesses from Ojala and Alenius (2005) into an object called 'VT', and the maximum varve depth (mm) in the composite stratigraphy into an object called 'max_depth'.

```r
# Set the file directory for the RData files
RDATdir <- '/Users/paullincoln/Dropbox/2024/Papers/RData_files/'
#Set a figure directory to save output plots
figdir <- '/Users/paullincoln/Dropbox/2024/Papers/R_Figures'

#Load in XRF, VT, variance and max_depth objects
load(file = paste0(RDATdir,'XRF_Varve_data.RData', sep = ''))
```

1.2 Transform and log the data for further analyses

Raw XRF measurements are affected by variability in the physical properties of the host sediment & compositional constant-sum constraints. This means that element intensities do not linearly represent concentration. To account for this, values are log ratioed and centre-log ratioed (CLR).

First, we calculate the normal log ratio between elements of interest. Here, the log function is used to calculate log ratios. The values are then written to the XRF data frame.

```r
#remove replicates
XRF <- XRF[is.na(XRF$Replicate),]
XRF <- XRF %>% dplyr::select(-Replicate)
#resample data to 0.4mm resolution
coarsen_resolution <- function(data, depth_col, resolution = 0.4) {
  data %>%
    mutate(group = floor(!!sym(depth_col) / resolution)) %>%
    group_by(group) %>%
    summarise(
      across(1:4, mean, na.rm = TRUE),
      across(5:15, sum, na.rm = TRUE),
      across(16:18, mean, na.rm = TRUE)
    ) %>%  # Calculate mean for columns 1:4 and 16:18, sum for columns 5:15
    select(-group) %>%
    mutate(!!sym(depth_col) := seq(0, (n() - 1) * resolution, by = resolution))
}

XRF <- coarsen_resolution(XRF, 'DepthID [mm]')

#add in log ratios, calculated prior to CLR transformation
XRF$`ln(inc/coh)` <- log(XRF$`Rh inc`/XRF$`Rh coh`)
XRF$`ln(Fe/Mn)` <- log(XRF$Fe/XRF$Mn)
XRF$`ln(Fe/Ti)` <- log(XRF$Fe/XRF$Ti)
XRF$`ln(Mn/Ti)` <- log(XRF$Mn/XRF$Ti)
```

We can now central log ratio the data. Note that the column numbers here are specific to the Nautajarvi data. These will change for other datasets so make sure that they are altered correctly to cover just the raw element data.

```
#re-arrange data frame columns, the number reflects the column number from the left.
XRF <-
  XRF[c(
    'DepthID [mm]',
    'Age (calBP)',
    'Age (yrCE)',
    '1% MCE',
    'Ca',
    'Fe',
    'K',
    'Mn',
    'S',
    'Si',
    'Ti',
    'Rh coh',
    'Rh inc',
    'ln(Fe/Ti)',
    'ln(Mn/Ti)',
    'ln(inc/coh)'
  )]

#clr XRF data.
CLR <- cenLR(XRF[c(5:13)])
CLR <- data.frame(CLR$x.clr)
#write the CLR transformed data back into the XRF data frame
XRF[c(5:13)] <- CLR[c(1:9)]

#Preserve total dataset for plotting in Figure 2
XRF_total <- XRF
#filter data to cover only varved section of the NAU-23 sequence
XRF <- XRF %>% dplyr::filter(`DepthID [mm]` <= max_depth)
```

1.3 Scale for variance

Prior to multivariate statistical analyses, the data are scaled according to the replicate variance. This is done to statistically downweight elements with low analytical precision. This follows the methodology adopted in the Xelerate software for XRF analyses (Weltje et al. (2015)).

```r
# Scale data for replicate variance to account for variable precision.
#re-order variance columns to match XRF
colorder<-colnames(XRF)[c(5:11)]

#Select 0.4mm replicate variances from variance df.
variance <- variance[2,]
variance <- subset(variance, select = -Resolution_mm)
sum(variance)
```

```
[1] 0.1574456
```

```r
variance<-variance[c(1:7)]
variance <- variance[, c(colorder, setdiff(names(variance), colorder))]

#Normalize the weights so that sum equals one
normalized_variance <- 1-(variance / sum(variance))
# Multiply the variance values to scale each element relative to analytical precision.
scaled_data <- as.data.frame(mapply(`*`, XRF[c(5:11)], normalized_variance))
#Create a matrix with selected elements to run clustering algorithm
clust_XRF<- as.matrix(scaled_data[c(1:7)])
```

1.4 Ward's Hierarchical clustering.

The number of clusters is pre-selected here, but is guided by assessment of the dendrogram, sedimentological core descriptions and the results of NBClust (Section 3.4).

```
#ward's hierarchical clustering
k <- 4 #choose number of clusters.
clust_col <- c('firebrick1','yellow2', 'steelblue','forestgreen')

clust_col_num <- c('1','2','3', '4')


set.seed(123)
dend <- clust_XRF %>%
  dist('euclidean') %>%
  hclust(method= 'ward.D') %>%
  as.dendrogram %>%
  dendextend::set("branches_k_color", k = k, value = clust_col)

# Extract labels/sample and corresponding branches
labels <- labels(dend)
branch_colors <- dend %>% get_leaves_branches_col


# Create a data frame containing labels assigned to row numbers
c <- data.frame(
  label = as.numeric(labels),
  branch_color = branch_colors
)
# Order by label value/ row
c <- c[order(c$label), ]
#mutate cluster colours back to numbers for plotting
c <- c %>% mutate(branch_color = dplyr::case_when(branch_color ==  clust_col[1] ~ '1',
                                                  branch_color ==  clust_col[2] ~ '2',
                                                  branch_color ==  clust_col[3] ~ '3',
                                                  branch_color ==  clust_col[4] ~ '4'))
#write back into XRF and scaled_data files
scaled_data$cluster <- c$branch_color
XRF$cluster <- c$branch_color

# Count occurrences of 4 in a 50-row rolling window
XRF <- XRF %>%
  dplyr::mutate(cluster_4 = zoo::rollapply(cluster, width = 50,
```

```
                                                      FUN = function(x) sum(x == 4),
                                                      align = "right", fill = NA),
                       cluster_3 = zoo::rollapply(cluster, width = 50,
                                                      FUN = function(x) sum(x == 3),
                                                      align = "right", fill = NA),
                       cluster_2 = zoo::rollapply(cluster, width = 50,
                                                      FUN = function(x) sum(x == 2),
                                                      align = "right", fill = NA),
                       cluster_1 = zoo::rollapply(cluster, width = 50,
                                                      FUN = function(x) sum(x == 1),
                                                      align = "right", fill = NA))
```

1.5 Principal components analysis (PCA)

```
#Perform the PCA on the scaled data and write output to a new object.
#Note that  clr-transformed data scaled for replicate variance are used here,
#secondary scaling is therefore not applied
data.pca <- PCA(scaled_data[c('Si','S','K','Ca','Ti','Mn','Fe')],
                scale.unit = F,
                graph = F)

#Assign depositional stages to subset data for PCAs
pcaval<-as.data.frame(data.pca$ind$coord[,1:2])
XRF$PC1 <- pcaval$Dim.1
XRF$PC2 <- pcaval$Dim.2

rm(pcaval)
```

2 Figures

This section presents the code used to plot the figures in the manuscript and supplementary information.

2.1 Format data for plotting

```
#set the figure output directory. Code will export pdfs of all plots into this folder.
setwd(figdir)

#load in meterological data from Halli weather station for Figure 1.
load(paste0(RDATdir,'Fig1_Halli_Data.RData'))

#load in core photo images for the plots in Figures 3-4. This object contains three files.
#G1_chronology is the chronology for the surface gravity core taken from the NAU-23.
#G1_image is the corresponding core photo for the gravity core.
#Core_image is a JPEG of the core section used in Figure 3.
load(paste0(RDATdir, 'Image_plots.RData'))

#Load in external data for discussion figures.
load(paste0(RDATdir,'Discussion_Figs_External_Data.RData'))

#set uniform figure colours
PCA_col <- c('orange2', 'purple3') #PCA colours

#add new column to list phases
XRF <- XRF %>%
  mutate(
    Phase = case_when(
      `Age (calBP)` > 7000 & `Age (calBP)` <= 9900 ~ "Phase 1",
      `Age (calBP)` >= 5000 & `Age (calBP)` <= 7000 ~ "Phase 2",
      `Age (calBP)` >= -80 & `Age (calBP)` <= 5000 ~ "Phase 3",
      TRUE ~ "Other" # In case the age does not fall into any of the defined phases
    )
  )

#transform XRF data to mean annual resolution via linear interpolation for discussion plots
XRFannual <- XRF[c(2,5:11,22,23)]

XRFannual$`Varve (yr BP)` <- trunc(XRFannual$`Age (calBP)`)
nam<-colnames(XRFannual[c(1,2:10)])
XRFannual <- XRFannual[c(11,2:10)] %>% group_by(`Varve (yr BP)`) %>%
```

```
  summarise(across(everything(), list(mean)))
colnames(XRFannual) <- nam
```

2.2 Figure 1: Halli station data

```
WIND_Met <- WIND_Met %>% mutate(`Mean wind speed anomaly (m/s)` =
                                `Monthly mean wind speed (m/s)`-
                                mean(`Monthly mean wind speed (m/s)`),
                                Color = ifelse(`Mean wind speed anomaly (m/s)` >0 ,
                                               'red', 'blue'))



Te<-ggplot(TEMP_Met, aes(x=Month+0.5)) +
  geom_rect(aes(xmin = 0.5, xmax = 5.1, ymin = -Inf, ymax = Inf),
            fill =  'grey80',
            color = 'black',
            linetype = "dashed",
            alpha = 0.25) +
  geom_rect(aes(xmin = 12, xmax = 12.9, ymin = -Inf, ymax = Inf),
            fill = 'grey80',
            color = 'black',
            linetype = "dashed",
            alpha = 0.25)+
  geom_ribbon(aes(ymin = `T average min monthly`,
                  ymax = `T average max monthly`),
              fill = 'red', alpha = 0.25)+
  geom_path(aes(y= `T average (°C)`)) +
  geom_hline(yintercept = mean(TEMP_Met$`T average (°C)`[2:13]),
             color = 'red',
             linetype = 'dashed')+
  ggpubr::theme_pubr() +
  scale_x_continuous(limits = c(0.5,12.9),
                     breaks = seq(1,12,1),
                     expand = c(0,0)) +
  scale_y_continuous(limits = c(-12,23),
                     breaks = seq(-20,30, 2),
                     expand = c(0,0)) +
  labs(x = 'Month' ,
       y= expression(Temperature~(degree*C)))
```

```
P<-ggplot(PREC_Met, aes(x=Month+0.5)) +
  geom_rect(aes(xmin = 0.5, xmax = 5.1, ymin = -Inf, ymax = Inf),
            fill =  'grey80',
            color = 'black',
            linetype = "dashed",
            alpha = 0.25) +
  geom_rect(aes(xmin = 12, xmax = 12.9, ymin = -Inf, ymax = Inf),
            fill = 'grey80',
            color = 'black',
            linetype = "dashed",
            alpha = 0.25)+
  geom_ribbon(aes(ymin = `Min (mm)`, ymax = `Max (mm)`),
              fill = 'blue3',
              alpha = 0.5)+
  geom_path(aes(y= `Prec average (mm)`)) +
  geom_hline(yintercept = PREC_Met_annual$`Prec average (mm)`/12,
             color = 'navy',
             linetype = 'dashed')+
  ggpubr::theme_pubr() +
  scale_x_continuous(limits = c(0.5,12.9),
                     breaks = seq(1,12,1),
                     expand = c(0,0)) +
  scale_y_continuous(limits = c(0,180),
                     breaks = seq(0,500, 20),
                     expand = c(0,0)) +
  labs(x = 'Month' , y= 'Precipitation (mm)')

W <- ggplot(WIND_Met, aes(x = Month+0.5)) +
  geom_rect(aes(xmin = 0.5, xmax = 5.1, ymin = -Inf, ymax = Inf),
            fill =  'grey80',
            color = 'black',
            linetype = "dashed",
            alpha = 0.25) +
  geom_rect(aes(xmin = 12, xmax = 12.9, ymin = -Inf, ymax = Inf),
            fill = 'grey80',
            color = 'black',
            linetype = "dashed",
            alpha = 0.25)+
  geom_crossbar(aes(ymin = 0,
                    ymax =`Mean wind speed anomaly (m/s)`,
                    y = `Mean wind speed anomaly (m/s)`,
                    fill = Color),
                width = 0.5) +
```

```
  geom_path(aes(y = `Mean wind speed anomaly (m/s)`),
            color = 'black')+
  geom_hline(yintercept =0,
             color = 'black') +
  ggpubr::theme_pubr()+
  scale_fill_identity()+
  scale_x_continuous(limits = c(0.5,12.9), breaks = seq(1,12,1),
                     expand = c(0,0)) +
  labs(x = 'Month' , y= 'Wind speed anomaly (m/s)')


Figure_1 <- cowplot::plot_grid(Te,P,W, nrow = 1, align = 'hv')


ggsave(filename = file.path(figdir, "Figure_1.pdf"),
       plot = Figure_1,
       width = 305.30292,
       height = 57.37082,
       units = "mm")

#print(Figure_1)
```

2.3 Figure 2: XRF elements against depth

```
#write a plot function
plot_function <- function(element, rollmean, ax_lab){

  ggplot(XRF_total) +
    geom_path(aes(y= as.numeric(`DepthID [mm]`/10),
                  x= rollmean(element, 1, na.pad = T)),
              color = 'grey88', linewidth = 0.2)+
    geom_path(aes(y= as.numeric(`DepthID [mm]`/10),
                  x= rollmean(element, rollmean, na.pad = T)),
              linewidth = 0.5) +
    ggpubr::theme_pubr() +
    scale_y_reverse(limits = c(724.96,0),
                    breaks = seq(0,1000,50),
                    expand = c(0,0))+
    geom_hline(yintercept = c(674,437.8,308,0),
               color = 'black') +
    ylab("Depth (cm)") +
    xlab(bquote(.(ax_lab)[clr])) +
```

```
    theme(text = element_text(),
          axis.text.x = element_text(angle = 90, hjust = 1))
}

#this smooth value acts as a moving average for the time series
smooth <- 25

Varvesannual <- ggplot(VT, aes(y=Depth_Interpolation/10)) +
  geom_path(aes(x = `Varve thickness (mm)`),
            size = 0.2,
            alpha = 0.5) +
  geom_path(aes(x = rollmean(`Varve thickness (mm)`,
                             smooth, na.pad = T))) +
  scale_x_log10(breaks = c(0.1,0.5,1,2.5,5,10)) +
  ggpubr::theme_pubr()+
  geom_hline(yintercept = c(674,437.8,308,0),
             color = 'black') +
  scale_y_reverse(limits = c(720, 0),
                  breaks = seq(0,10000, 500),
                  expand = c(0,0))+
  theme(legend.position = 'none')


Varvesseasonal <- ggplot(VT, aes(y=Depth_Interpolation/10)) +
  geom_path(aes(x = `winter thickness`),
            color = 'blue',
            size = 0.2,
            alpha = 0.5) +
  geom_path(aes(x = rollmean(`winter thickness`, smooth, na.pad = T)),
            color = 'blue') +
  geom_path(aes(x = `summer thickness`),
            color = 'red',
            size = 0.2,
            alpha = 0.5) +
  geom_path(aes(x = rollmean(`summer thickness`, smooth, na.pad = T)),
            color = 'red') +
  scale_x_log10(breaks = c(0.1,0.5,1,2.5,5,10)) +
  ggpubr::theme_pubr()+
  geom_hline(yintercept = c(674,437.8,308,0),
             color = 'black') +
  scale_y_reverse(limits = c(720, 0),
                  breaks = seq(0,10000, 500),
                  expand = c(0,0)) +
```

```r
  theme(legend.position = 'none')

xrf_runav <- 25

Ti_plot<-plot_function(XRF_total$Ti, xrf_runav, 'Ti')
S_plot <- plot_function(XRF_total$ `S`, xrf_runav, 'S')
Mn_plot <- plot_function(XRF_total$`Mn`, xrf_runav, 'Mn')
Fe_plot <- plot_function(XRF_total$`Fe`, xrf_runav, 'Fe')
Ca_plot <- plot_function(XRF_total$`Ca`, xrf_runav, 'Ca')
Si_plot <- plot_function(XRF_total$`Si`, xrf_runav, 'Si')
K_plot <- plot_function(XRF_total$`K`, xrf_runav, 'K')
S_plot<- plot_function(XRF_total$`S`, xrf_runav, 'S')

Figure_2 <- cowplot::plot_grid(Ti_plot,
                               K_plot + theme(axis.text.y = element_blank(),
                                              axis.ticks.y = element_blank(),
                                              axis.title.y = element_blank(),
                                              axis.line.y = element_blank()),
                               Si_plot + theme(axis.text.y = element_blank(),
                                               axis.ticks.y = element_blank(),
                                               axis.title.y = element_blank(),
                                               axis.line.y = element_blank()),
                               S_plot + theme(axis.text.y = element_blank(),
                                              axis.ticks.y = element_blank(),
                                              axis.title.y = element_blank(),
                                              axis.line.y = element_blank()),
                               Fe_plot + theme(axis.text.y = element_blank(),
                                               axis.ticks.y = element_blank(),
                                               axis.title.y = element_blank(),
                                               axis.line.y = element_blank()),
                               Mn_plot+ theme(axis.text.y = element_blank(),
                                              axis.ticks.y = element_blank(),
                                              axis.line.y = element_blank(),
                                              axis.title.y = element_blank()),
                               Varvesseasonal+ theme(axis.text.y = element_blank(),
                                                     axis.ticks.y = element_blank(),
                                                     axis.line.y = element_blank(),
                                                     axis.title.y = element_blank()),
                               Varvesannual+ theme(axis.text.y = element_blank(),
                                                   axis.ticks.y = element_blank(),
                                                   axis.line.y = element_blank(),
                                                   axis.title.y = element_blank()),

```

```
                              align = 'hv',
                              nrow = 1)
ggsave(filename = file.path(figdir, "Figure_2.pdf"),
       plot = Figure_2,
       width = 319.172,
       height = 211.361,
       units = "mm")

#print(Figure_2)
```

2.4 Figure 3: PCA, clustering & core plot

PCA results

```
#Assign depositional stages to subset data for PCAs
# Plot with custom colors
PCA_cluster<-factoextra::fviz_pca_biplot(data.pca,
                                        legend = 'none',
                                        title = element_blank(),
                                        axes = c(1,2),
                                        label = 'var',
                                        pointshape = 21,
                                        pointsize = 0.5,
                                        addEllipses = F,
                                        # Customizations
                                        alpha.ind = 0.5,
                                        labelsize = 5,
                                        col.ind = XRF$cluster,
                                        fill.ind = NA,
                                        col.var = 'black',
                                        theme = ggpubr::theme_pubr() +
                                          theme(text = element_text(size = 12),
                                                axis.title = element_text(size = 14))) +
  scale_color_manual(values = c('1'= clust_col[1],
                                '2' = clust_col[2],
                                '3' = clust_col[3],
                                '4' = clust_col[4]))

phase_colors <- c( "cyan2", "orangered2",'seagreen2')

PCA_phase<-factoextra::fviz_pca_ind(
  data.pca,
```

```
  geom = "point",
  invisible = 'point',
  col.ind = XRF$Phase, # Color by the Phase column
  alpha.ind = 0,
  addEllipses = TRUE, # Add concentration ellipses
  ellipse.level = 0.95,
  ellipse.alpha = 0
) + ggpubr::theme_pubr() + theme(legend.title = element_blank())


ellipses_layer <- PCA_phase$layers[[2]]
PCA<-PCA_cluster + ellipses_layer + scale_color_manual(values = c(phase_colors,clust_col))

#calculate PC variable loadings
pc_loading<-as.data.frame(sweep(data.pca$var$coord,
                                2,
                                sqrt(data.pca$eig[1:ncol(data.pca$var$coord),
                                                  1]),
                                FUN="/"))

pc_loading <- pc_loading[c(1:2)]
colnames(pc_loading) <- c('PC1', 'PC2')
pc_loading$element <- rownames(pc_loading)

  pc_loading <- reshape2::melt(pc_loading,
                               id.vars = "element",
                               variable.name = "PC",
                               value.name = "Value")
  # Change the order of elements
  pc_loading$element <- factor(pc_loading$element,
                               levels = c('Fe','Mn','S','Ca','Si','Ti','K'))

 pc_loading<- ggplot(pc_loading, aes(x = element, y = Value, fill = PC)) +
    geom_bar(stat = "identity", position = "dodge",  color = 'black') +
    geom_hline(yintercept = 0)+
    scale_fill_manual(values = PCA_col)+
    labs(title = "Element loadings to PC1 and PC2", y = element_blank()) +
    ggpubr::theme_pubr() +
   facet_wrap(~PC, nrow =2) +
   scale_y_continuous(limits = c(-0.5,0.75),
                      breaks = seq(-0.75,0.75, 0.25)) +
    theme(legend.position = 'none',
          axis.title.x = element_blank(),
```

```
          axis.title.y = element_blank(),
          strip.text = element_blank())
```

Core plots to investigate seasonal signals of PC axes 1 and 2

```
######This code is for core G1#######
XRF_imageplot<-XRF %>% dplyr::filter(`DepthID [mm]` < 60)
#rotate core image 90 degrees
G1_image_vert <- OpenImageR::rotateFixed(G1_image,90)

###plot assigned ages to geom_vlines
hline_labels <- c(as.character(G1_chronology$`Age YrAD`))
hline_positions <- c(as.numeric(G1_chronology$`Core depth (mm)`))


p_PC1<-ggplot(XRF_imageplot)+
  ggpubr::background_image(G1_image_vert)+
  geom_rug(aes(y = `DepthID [mm]`,
               color =as.factor(cluster)),
           size = 1,
           length = unit(0.1, "npc")) +
  scale_color_manual(values = c('1'= clust_col[1],
                                '2' = clust_col[2],
                                '3' = clust_col[3],
                                '4' = clust_col[4])) +
  geom_path(aes(y=`DepthID [mm]`, x = `PC1`-mean(PC1)),
            color = PCA_col[1],
            linewidth = 0.5)+
  geom_hline(yintercept = hline_positions,
             color = 'grey80',
             linetype = 'dashed',
             linewidth = 0.5) +
  ggpubr::theme_pubr() +
  scale_y_reverse(limits = c(60,0),
                  breaks = seq(0,60,2),
                  expand = c(0,0)) +
  scale_x_continuous(limits = c(-2.5,2.5) )+
  theme(legend.position = 'none',
        text = element_text(size = 10)) +
  xlab('PC1')

p_PC2<-ggplot(XRF_imageplot)+
```

```
  ggpubr::background_image(G1_image_vert)+
  geom_path(aes(y=`DepthID [mm]`, x = (`PC2`-mean(PC2))*2),
            color = PCA_col[2], linewidth = 0.5)+
  geom_hline(yintercept = hline_positions,
             color = 'grey80',
             linetype = 'dashed',
             linewidth = 0.5) +
  ggpubr::theme_pubr() +
  scale_y_reverse(limits = c(60,0),
                  breaks = seq(0,60,2),
                  expand = c(0,0)) +
  scale_x_continuous(limits = c(-1.5,1.5)) +
  theme(legend.position = 'none',
        text = element_text(size = 10))+
  xlab('PC2')

for (i in seq_along(hline_positions)) {
  p_PC2 <- p_PC2 +
    annotate("text", y = hline_positions[i], x= 1.1 ,
             label = hline_labels[i],
             vjust = -0.5,
             hjust = 0,
             angle = 0)
}


core_pca_plot <-cowplot::plot_grid(p_PC1,p_PC2 + theme(axis.line.y = element_blank(),
                                                       axis.text.y = element_blank(),
                                                       axis.title.y = element_blank(),
                                                       axis.ticks.y= element_blank()),
                                   nrow = 1, align = 'hv')

rm(G1_chronology,XRF_imageplot,G1_image)
```

Hierarchical clustering results (dendrogram, heatmap and area plot)

```
#Extract PC values from XRF and transpose to plot as rows
PCs_t <- as.data.frame(c(XRF[c(22:23)])) %>% as.matrix() %>% t()
#transpose dendrogram to plot as columns
dend_t<-t(dend)
#transpose dendrogram to plot as columns

```

```r
#plot heatmap and dendrogram
heatdendrplot<-grid::grid.grabExpr(ComplexHeatmap::draw(
  ComplexHeatmap::Heatmap(PCs_t,name = 'PC_value',
                          cluster_columns = dend_t,
                          column_dend_height = unit (3, 'cm'),
                          use_raster = TRUE,
                          raster_quality = 5,
                          heatmap_height = unit (5, 'cm'),
                          column_split = k,
                          heatmap_legend_param = list(legend_direction = "horizontal",
                                                      legend_width = unit(5, "cm"))),

Clust_plot <- XRF %>%
  dplyr::select(`DepthID [mm]`,
                `Age (calBP)`,
                `Age (yrCE)`,
                cluster_1,
                cluster_2,
                cluster_3,
                cluster_4) %>%
  tidyr::pivot_longer(cols = starts_with("cluster_"),
                      names_to = "cluster",
                      values_to = "count")
```

Combine plots

```r
#Combine plots and save
Figure_3 <-cowplot::plot_grid(heatdendrplot,
                              PCA,pc_loading,
                              core_pca_plot,
                              rel_heights = c(0.8,1),
                              nrow = 2,
                              align = 'hv',
                              labels = c('A.', 'B.', 'C.', 'D.'))

ggsave(filename = file.path(figdir, "Figure_3.pdf"), plot = Figure_3,
       width = 181,
       height = 190,
       units = "mm")

#print(Figure_3)
```

2.5 Figure 4: Geochemical signals from the mid-Holocene ferrogenic sections

```
#Create subsetted dataframe for plotting
XRF_313_333cm <- XRF %>% dplyr::filter(`DepthID [mm]` > 3129 & `DepthID [mm]` < 3331)

core_plot <- ggplot(XRF_313_333cm, aes(y= `DepthID [mm]`/10))+
  ggpubr::background_image(Core_image) +
  geom_rug(aes(color= as.factor(cluster)),
           sides = 'r',
           size = 0.25,
           length = unit(0.1, "npc")) +
  scale_color_manual(values = c('1'= clust_col[1],
                                '2' = clust_col[2],
                                '3' = clust_col[3],
                                '4' = clust_col[4])) +
  scale_y_reverse(limits = c(333, 313),
                  breaks = seq(300,350,1),
                  expand = c(0,0)) +
  ggpubr::theme_pubr() +
  theme(legend.position = 'none') +
  labs(y='Depth (cm)')

Mn_Ti <-ggplot(XRF_313_333cm, aes(y= `DepthID [mm]`/10)) +
  geom_path(aes(x=rollmean(`ln(Mn/Ti)`, 1, na.pad = T)),
            color = 'green4') +
  scale_y_reverse(limits = c(333, 313),
                  breaks = seq(300,350,1),
                  expand = c(0,0))+
  ggpubr::theme_pubr() +
  theme(axis.text.y=element_blank(),
        axis.ticks.y=element_blank(),
        axis.title.y = element_blank(),
        axis.line.y = element_blank()) +
  labs(x= expression(ln(Mn/Ti)))

Fe_Ti <-ggplot(XRF_313_333cm, aes(y= `DepthID [mm]`/10)) +
  geom_path(aes(x=rollmean(`ln(Fe/Ti)`, 1, na.pad = T)),
            color = 'brown') +
  scale_y_reverse(limits = c(333, 313),
                  breaks = seq(300,350,1),
                  expand = c(0,0))+
  ggpubr::theme_pubr() +
```

```
  theme(axis.text.y=element_blank(),
        axis.ticks.y=element_blank(),
        axis.title.y = element_blank(),
        axis.line.y = element_blank()) +
  labs(x= expression(ln(Fe/Ti)))


PC2 <-ggplot(XRF_313_333cm, aes(y= `DepthID [mm]`/10)) +
  geom_path(aes(x=rollmean(PC2, 1, na.pad = T)),
            color = PCA_col[2]) +
  scale_y_reverse(limits = c(333, 313),
                  breaks = seq(300,350,1),
                  expand = c(0,0))+
  ggpubr::theme_pubr() +
  theme(axis.text.y=element_blank(),
        axis.ticks.y=element_blank(),
        axis.title.y = element_blank(),
        axis.line.y = element_blank())  +
  labs(x='PC2')
PC1 <-ggplot(XRF_313_333cm, aes(y= `DepthID [mm]`/10)) +
  geom_path(aes(x=zoo::rollmean(PC1, 1, na.pad = T)),
            color = PCA_col[1]) +
  scale_y_reverse(limits = c(333, 313),
                  breaks = seq(300,350,1),
                  expand = c(0,0))+
  ggpubr::theme_pubr() +
  theme(axis.text.y=element_blank(),
        axis.ticks.y=element_blank(),
        axis.title.y = element_blank(),
        axis.line.y = element_blank()) + labs(x='PC1')

C_plot <- XRF_313_333cm %>%
  dplyr::select(`DepthID [mm]`,
                `Age (calBP)`,
                `Age (yrCE)`,
                cluster_1, cluster_2, cluster_3, cluster_4) %>%
  tidyr::pivot_longer(cols = starts_with("cluster_"),
                      names_to = "cluster",
                      values_to = "count")

Figure_4 <-cowplot::plot_grid(core_plot,Fe_Ti,Mn_Ti,PC2,PC1, nrow=1, align = 'hv')


```

```r
ggsave(filename = file.path(figdir, "Figure_4.pdf"),
       plot = Figure_4,
       width = 272.509,
       height = 195.737,
       units = "mm")

#print(Figure_4)
```

2.6 Discussion figures:

The discussion figures include multiple time series which are loaded into an RData file in Section 2.1. These include the varve thickness data and GDD data from Nautajarvi (Ojala and Alenius (2005)) pollen-derived temperature reconstructions from northern (Salonen et al. (2024)) and southern Scandinavia (Sejrup et al. (2016)), dendrological isotopic reconstructions (Helama et al. (2021)), alkenone sea surface temperature records from the North Atlantic (Sicre et al. (2021)) and model-derived Boreal temperature reconstructions (Van Dijk et al. (2024)). The following code plots these time series against the NAU-23 principal components.

Figure 5

```r
max_time <- 9900
min_time <- -73

HTM_l <- c(7000, 5000)

cp<-ggplot(Clust_plot, aes(x = `Age (calBP)`, y = count*2,
                           fill = cluster)) +
  geom_area(color = 'black', size = 0.1) +
  scale_fill_manual(values = c('cluster_1' = clust_col[1],
                               'cluster_2'=clust_col[2],
                               'cluster_3' = clust_col[3],
                               'cluster_4' = clust_col[4]))+
  geom_vline(xintercept = HTM_l, color = 'black')+
  scale_y_continuous(limits =c(0,100),
                     breaks =seq(0,10000, 10),
                     expand = c(0,0))+
  scale_x_reverse(limits =c(max_time,min_time),
                  breaks =seq(-10000,10000, 500),
                  expand = c(0,0))+
  labs(
    x = "Varve yr BP",
```

```
    y = "(%)") +
  ggpubr::theme_pubr() +
  theme(legend.position = 'none')

TJul <- ggplot(Salonen, aes(x= Cal_a_BP)) +
  geom_ribbon(aes(ymin = Tjul_min95, ymax = Tjul_max95),
              fill = 'orange',
              alpha = 0.2)+
  geom_path(aes(y=Tjul_Median)) +
    geom_vline(xintercept = HTM_l,
               color = 'black')+
  scale_x_reverse(limits = c(max_time,min_time),
                  breaks = seq(0,10000, 500),
                  expand = c(0,0))+
  scale_y_continuous(limits = c(9.8, 16.8),
                     breaks = seq(0,50,1))+
  ggpubr::theme_pubr()

VTplot <- tidyr::pivot_longer(VT[c(3,4,8,9)],
                              cols = c(`Varve thickness (mm)`,
                                       `winter thickness`,
                                       `summer thickness`),
                              names_to = "Type",
                              values_to = "Thickness (mm)")

VTplot <- VTplot %>%
  group_by(Type) %>%
  arrange(BP) %>%
  mutate(`50yr_running_mean` = rollmean(`Thickness (mm)`, 50,
                                        fill = NA,
                                        align = "right"))

Varves <- ggplot(VTplot, aes(x=BP)) + geom_path(aes(y = `Thickness (mm)`,
                                                    color = Type),
                                                size = 0.1,
                                                alpha = 0.5) +
  geom_path(aes(y = `50yr_running_mean`,
                color = Type),
            linetype = 'solid',
            size = 0.25) +
  geom_vline(xintercept = HTM_l,
             color = 'black')+
  scale_color_manual(values = c('red','black','blue'))+
```

```
  scale_y_log10(breaks = c(0.1,0.5,1,2.5,5,10)) +
  ggpubr::theme_pubr()+
  scale_x_reverse(limits = c(max_time,min_time),
                  breaks = seq(0,10000, 500),
                  expand = c(0,0)) +
  theme(legend.position = 'none')

PCs <- tidyr::pivot_longer(XRFannual[c('Age (calBP)','PC1','PC2')],
                           cols = c(`PC1`, `PC2`),
                           names_to = "PC",
                           values_to = "Value")

PCs <- PCs %>%
  group_by(PC) %>%
  arrange(`Age (calBP)`) %>%
  mutate(`50yr_running_mean` = rollmean(`Value`, 50, fill = NA, align = "right"))

PC <- ggplot(PCs, aes(x=`Age (calBP)`)) + geom_path(aes(y = `Value`, color = PC),
                                                    size = 0.1,
                                                    alpha = 0.5) +
  geom_path(aes(y = `50yr_running_mean`, color = PC),
            linetype = 'solid',
            size = 0.25) +
  geom_vline(xintercept = HTM_l,
             color = 'black')+
  scale_color_manual(values = c(PCA_col))+
  ggpubr::theme_pubr()+
  scale_x_reverse(limits = c(max_time,min_time),
                  breaks = seq(0,10000, 500),
                  expand = c(0,0))+
  theme(legend.position = 'none') +
  scale_y_continuous(limits = c(-3,2.5),
                     breaks = seq(-10,10,1))

GDD_plot <- ggplot(GDD, aes(x= `Age (BP)`, y= GDD)) +
    geom_rect(aes(ymax = 1400, ymin = 1280, xmin = -Inf, xmax = Inf),
              fill = 'grey80',
              alpha = 0.1)+
  geom_path() +
  geom_path(aes(y= zoo::rollmean(GDD, 10, na.pad = T)),
            colour = 'red') +
  geom_vline(xintercept = HTM_l,
             color = 'black')+
```

```
  geom_hline(yintercept = 1280) +
  geom_hline(yintercept = 1400) +
  geom_hline(yintercept = 1340)+
  ggpubr::theme_pubr()+
  scale_x_reverse(limits = c(max_time,min_time),
                  breaks = seq(0,10000, 500),
                  expand = c(0,0)) +
  theme(legend.position = 'none')

Sejrup_PC <- ggplot(Sejrup_stack, aes(x=`Age (yr BP)`, y = `PC`)) +
  geom_ribbon(aes(ymin = `PC (+1 sigma)`, ymax = `PC (-1 sigma)`),
              fill = 'forestgreen',
              alpha = 0.2)+
  geom_ribbon(aes(ymin = `PC (+2 sigma)`, ymax = `PC (-2 sigma)`),
              fill = 'forestgreen',
              alpha = 0.2)+
  geom_path()+
  geom_ribbon(data =Sejrup_stack_PC2, aes(y = `PC (median)`,
                                          ymin = `PC (+1 sigma)`,
                                          ymax = `PC (-1 sigma)`),
              fill = 'navy', alpha = 0.2)+
  geom_ribbon(data =Sejrup_stack_PC2,aes(y = `PC (median)`,
                                         ymin = `PC (+2 sigma)`,
                                         ymax = `PC (-2 sigma)`),
              fill = 'navy',
              alpha = 0.2)+
  geom_path(data =Sejrup_stack_PC2,aes(x=`Age (yr BP)`, y = `PC (median)`))+
  geom_hline(yintercept = 0)+ggpubr::theme_pubr()+
  geom_vline(xintercept = HTM_1,
             color = 'black')+
  ggpubr::theme_pubr()+
  scale_x_reverse(limits = c(max_time,min_time),
                  breaks = seq(0,10000, 500),
                  expand = c(0,0)) +
  theme(legend.position = 'none')+
  scale_y_continuous(limits = c(-4.6,2),
                     breaks = seq(-10,10,2))

Figure_5<-cowplot::plot_grid(cp + theme(axis.title.x = element_blank(),
                              axis.line.x = element_blank(),
                              axis.text.x = element_blank(),
                              axis.ticks.x = element_blank()),
                   Varves+ theme(axis.title.x = element_blank(),
```

```
                                 axis.line.x = element_blank(),
                                 axis.text.x = element_blank(),
                                 axis.ticks.x = element_blank()),
                   PC+ theme(axis.title.x = element_blank(),
                             axis.line.x = element_blank(),
                             axis.text.x = element_blank(),
                             axis.ticks.x = element_blank()),
                   GDD_plot+ theme(axis.title.x = element_blank(),
                                   axis.line.x = element_blank(),
                                   axis.text.x = element_blank(),
                                   axis.ticks.x = element_blank()),
                   TJul+ theme(axis.title.x = element_blank(),
                               axis.line.x = element_blank(),
                               axis.text.x = element_blank(),
                               axis.ticks.x = element_blank()),
                   Sejrup_PC, ncol=1, align = 'hv',
                   rel_heights = c(0.5,1,1,1,1,1),
                   labels = c('A.','B.','C.','D.','E.','F.'))

ggsave(filename = file.path(figdir, "Figure_5.pdf"), plot = Figure_5,
       width = 190,
       height = 280,
       units = "mm")

#print(Figure_5)
```

Figure 6

```
XRF_plot <- XRFannual %>% dplyr::filter(`Age (calBP)` >3999 & `Age (calBP)`<8001)
smot <- 50

min_time2 <- 4500
max_time2  <- 7500


#add boxes around observed events and phases discussed in the manuscript
add_events <- function(box_col, box_col1) {
  list(
    geom_rect(aes(xmax = 5000, xmin = 7000,ymin = -Inf, ymax = Inf),
              fill = box_col1,
              alpha = 0.2,
              inherit.aes = FALSE),
```

```
    geom_rect(aes(xmax = 6910,xmin = 6790, ymin = -Inf, ymax = Inf),
              fill = box_col,
              alpha = 0.2,
              inherit.aes = FALSE),
    geom_rect(aes(xmax = 6660, xmin = 6560, ymin = -Inf, ymax = Inf),
              fill = box_col,
              alpha = 0.2,
              inherit.aes = FALSE),
    geom_rect(aes(xmax = 6405,xmin = 6290, ymin = -Inf,ymax = Inf),
              fill = box_col,
              alpha = 0.2,
              inherit.aes = FALSE),
    geom_rect(aes(xmax = 6250, xmin = 6080, ymin = -Inf, ymax = Inf),
              fill = box_col,
              alpha = 0.2,
              inherit.aes = FALSE),
    geom_rect(aes(xmax = 5980, xmin = 5950, ymin = -Inf, ymax = Inf),
              fill = box_col,
              alpha = 0.2,
              inherit.aes = FALSE),
    geom_rect(aes(xmax = 5850, xmin = 5810, ymin = -Inf, ymax = Inf),
              fill = box_col,
              alpha = 0.2,
              inherit.aes = FALSE),
    geom_rect(aes(xmax = 5430, xmin = 5380, ymin = -Inf, ymax = Inf),
              fill = box_col,
              alpha = 0.2,
              inherit.aes = FALSE),
    geom_rect(aes(xmax = 5350, xmin = 5280, ymin = -Inf, ymax = Inf),
              fill = box_col,
              alpha = 0.2,
              inherit.aes = FALSE),
    geom_rect(aes(xmax = 5230, xmin = 5100, ymin = -Inf, ymax = Inf),
              fill = box_col,
              alpha = 0.2,
              inherit.aes = FALSE)
  )
}

#define the box colour for these events/ phases
box_col1 <- 'grey90'
box_col <- 'grey60'

```

```
#NAU-23 XRF PC-2
p <-ggplot(XRF_plot, aes(x= `Age (calBP)`, y = PC2)) +
  add_events(box_col, box_col1)+
  geom_rug(data = XRF, aes(x= `Age (calBP)`, color = cluster),
           linewidth = 5,
           sides = 't',
           show.legend = FALSE)+
  scale_color_manual(values = clust_col)+
  geom_path(aes(y = rollmean(PC2,1, na.pad = TRUE)),
            color = PCA_col[2],
            alpha = 0.5,
            linetype = 'solid',
            size = 0.2) +
  geom_path(aes(y = rollmean(PC2, smot, na.pad = TRUE)),
            color = 'purple4',
            linetype = 'solid',
            size = 0.5) +
  geom_hline(yintercept = mean(XRF_plot$PC2))+
  scale_x_reverse(limits = c(max_time2,min_time2),
                  breaks = seq(max_time2,min_time2,-100),
                  expand = c(0,0)) +
  theme(legend.position = 'none') +
  ggpubr::theme_pubr()

# Calculate the mean GDD between 5000 and 7000 BP
GDD_MH <- GDD %>%
  dplyr::filter(`Age (BP)` >= 5000 & `Age (BP)` <= 7000) %>%
  summarize(mean_GDD = mean(GDD, na.rm = TRUE)) %>%
  pull(mean_GDD)

# Adjust the GDD values to represent anomalies wrt the mid Holocene mean
GDD <- GDD %>%
  mutate(GDD_anomaly = GDD - GDD_MH)

#Plot the GDD anomaly
GDD_anom<-ggplot(GDD, aes(x = `Age (BP)`, y = GDD_anomaly)) +
  add_events(box_col, box_col1)+
  geom_path() +
  geom_hline(yintercept = 0,
             linetype = "dashed") +
  geom_area(aes(y = ifelse(GDD_anomaly > 0, GDD_anomaly, 0)),
            fill = "red",
```

```
              alpha = 0.3) +
  geom_area(aes(y = ifelse(GDD_anomaly < 0, GDD_anomaly, 0)),
              fill = "blue",
              alpha = 0.3) +
  scale_x_reverse(limits = c(7500, 4500),
                  breaks = seq(0, 10000, 500),
                  expand = c(0, 0)) +
  ggpubr::theme_pubr() +
  theme(legend.position = 'none') +
  labs(x = "Age (BP)", y = "Adjusted GDD")

#Plot model-derived Boreal temps from van Dijk et al. (2024)
Tas <- ggplot(Van_Dijk_Boreal, aes(x=`Age_BP`, y=Annual)) +
  add_events(box_col, box_col1)+
  geom_hline(yintercept = 0)+
  ggpubr::theme_pubr()+
  geom_path(color = 'red',)+
  geom_path(aes(y = rollmean(Annual, 50, na.pad = TRUE)),
              color = 'red4',
              linetype = 'solid',
              size = 0.5) +
  scale_x_reverse(limits = c(7500,4500),
                  breaks = seq(7500,4500,-100),
                  expand = c(0,0))+
  labs(x = "Age (calBP)",
       y = expression("T (\u00B0C)"))

#Plot TSI
TSI <- ggplot(Steinhilber, aes(x=`Age`, y=dTSI)) +
  add_events(box_col, box_col1)+
  geom_hline(yintercept = 0)+
  ggpubr::theme_pubr()+
  geom_path(color = 'red')+
  ggpubr::theme_pubr()+
  scale_x_reverse(limits = c(7500,4500),
                  breaks = seq(7500,4500,-100),
                  expand = c(0,0))+
  labs(
    x = "Age (calBP)",
    y = expression(Delta * "TSI")
  )
```

```
#Plot dendro-derived isotopic reconstructions from Helama et al. (2021)
H <-ggplot(Helama, aes(x=Age, y = oktas_r)) +
  add_events(box_col, box_col1)+
  geom_path()+
  scale_x_reverse(limits = c(7500,4500),
                  breaks = seq(7500,4000,-100),
                  expand = c(0,0)) +
  xlab('Age (varve yr BP)')+
  ggpubr::theme_pubr()


#Plot N. Atlantic SSTs from MD99-2275 (Sicre et al., 2021)
M99 <-ggplot(Sicre_MD99_2275, aes(x=yearBP, y = SST)) +
  add_events(box_col, box_col1)+
  geom_path(color = 'green2')+
  geom_path(aes(y = rollmean(SST, 5, na.pad = TRUE)),
            color= 'green4',
            linetype = 'solid',
            size = 0.5) +
  scale_x_reverse(limits = c(7500,4500),
                  breaks = seq(7500,4000,-100),
                  expand = c(0,0)) +
  xlab('Age (varve yr BP)') +
  scale_y_continuous(limits = c(6.5, 12.5))+
  ggpubr::theme_pubr()+
  labs(
    x = "Age (calBP)",
    y = expression("SST (\u00B0C)")
  )

#Plot N. Atlantic IRD stack from Bond et al. (2001).
IRD <- ggplot(Bond_IRD, aes(x=age, y = stacked)) +
  add_events(box_col, box_col1)+
  geom_path()+
  ggpubr::theme_pubr()+
  scale_x_reverse(limits = c(7500,4500),
                  breaks = seq(7500,4000,-100),
                  expand = c(0,0)) + xlab('Age (varve yr BP)')

Figure_6 <-cowplot::plot_grid(p + theme(axis.title.x = element_blank(),
                                        axis.line.x = element_blank(),
                                        axis.text.x = element_blank(),
                                        axis.ticks.x = element_blank()),
```

```
                    GDD_anom+ theme(axis.title.x = element_blank(),
                                    axis.line.x = element_blank(),
                                    axis.text.x = element_blank(),
                                    axis.ticks.x = element_blank()),
                    TSI+ theme(axis.title.x = element_blank(),
                               axis.line.x = element_blank(),
                               axis.text.x = element_blank(),
                               axis.ticks.x = element_blank()),
                    Tas +theme(axis.title.x = element_blank(),
                               axis.line.x = element_blank(),
                               axis.text.x = element_blank(),
                               axis.ticks.x = element_blank()),
                    H+ theme(axis.title.x = element_blank(),
                             axis.line.x = element_blank(),
                             axis.text.x = element_blank(),
                             axis.ticks.x = element_blank()),
                    M99+ theme(axis.title.x = element_blank(),
                               axis.line.x = element_blank(),
                               axis.text.x = element_blank(),
                               axis.ticks.x = element_blank()),
                    IRD,align = 'hv', ncol = 1)




ggsave(filename = file.path(figdir, "Figure_6.pdf"),
       plot = Figure_6,
       width = 231.353,
       height = 323.545,
       units = "mm")

#print(Figure_6)
```

3 Supplementary Data

The following section presents supplementary information and data from the analyses of the NAU-23 sequence. Section 3.1 presents the NAU-23 composite stratigraphy and chronology. Section 3.2 shows more detailed SEM-EDS analysis of the mid-Holocene Nautajärvi varve structure, supporting continuous a continuous clastic-biogenic varve forming process throughout the HTM. Section 3.3 shows code used to plot co-variance matrices of elemental compositions through Phase 1-3 of the NAU-23 stratigraphy. Section 3.4 shows results of the NBcluster of the NAU-23 XRF data, used to identify the optimum number of clusters. Section 3.5 presents individual PCA biplots of each cluster in the NAU-23 geochemical data, alongside tables containing supplementary statistics of the composite PCA analyses in the main paper.

3.1 NAU-23 chronology

The NAU-23 chronology was constructed by visually matching 210 marker layers from previous records to the NAU-23 stratigraphy (Ojala and Alenius (2005)). For the section post -46 varve yr BP, used the updated Nautajärvi chronology (Kosonen et al. (2023)). The distribution of the marker layers was used to transfer the Nautajarvi varve chronology to the NAU-23 sequence. Figure 1 presents the distribution of the marker layers and the interpolated age-depth model for the NAU-23 sequence.

3.2 NAU-23 SEM-EDS

Figure 2 supplements the SEM-EDS data presented in Figure 4B of the main manuscript. Here, results obtained from additional analyses on the clastic laminae of the varve structure in the HTM are presented. These analyses support continual clastic biogenic varve formation throughout the HTM.

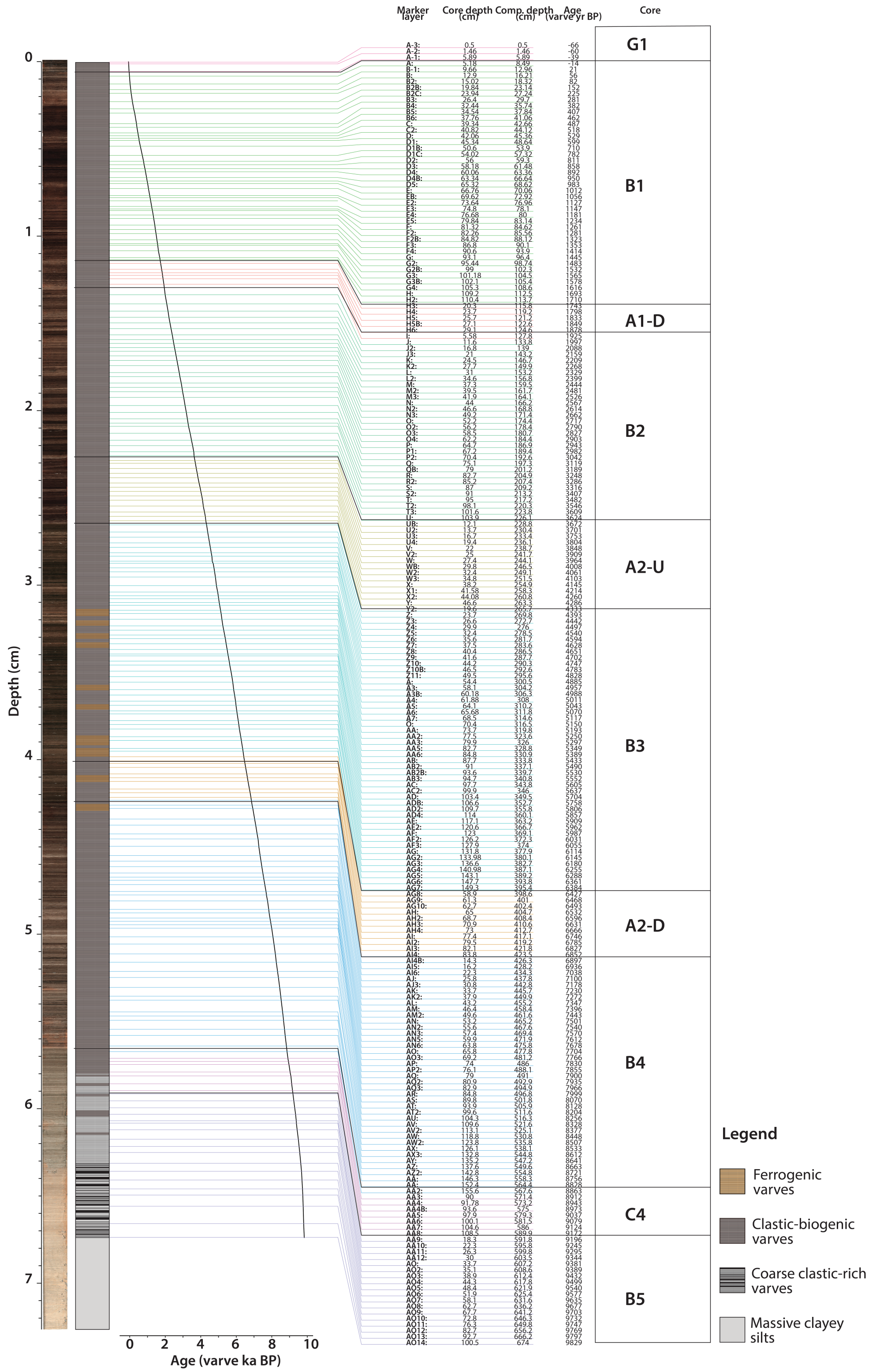
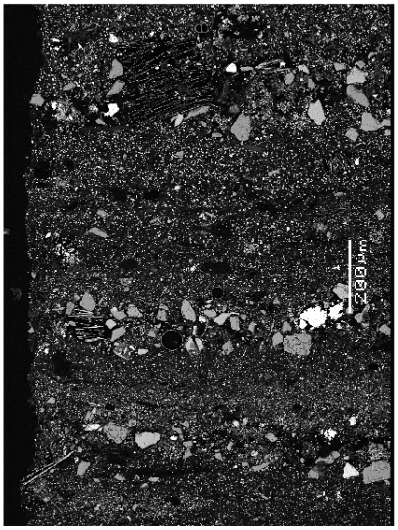
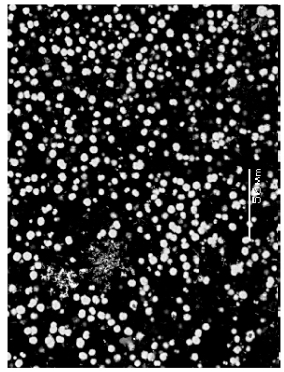
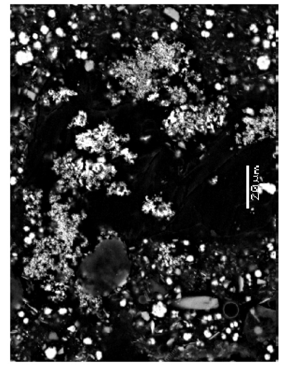


Figure 1: The NAU-23 composite stratigraphy, constructed from 10 individual drives (G1, B1, A1-D, B2, A2-U, B3, A2-D, B4, C4 and B5). The location of 210 marker horizons used to transfer the varve-counted chronology to the NAU-23 sequence. Linear interpolation between these marker horizons is used to construct the NAU-23 age-depth model (shown). Core lithofacies follows those in Figure 2 of the main manuscript.

“5.2 ka” varves in Lake Nautajärvi



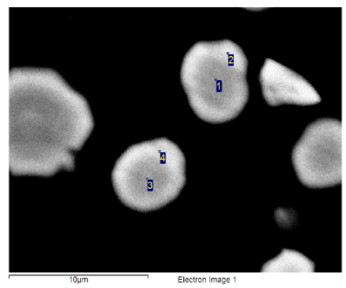




Processing option : Oxygen by stoichiometry (Normalised)

Spectrum	In stats.	P	Ca	Mn	Fe	Total
1	Yes	3.31	2.20	8.60	85.90	100.00
2	Yes	4.33	2.27	7.97	85.42	100.00
3	Yes			7.80	92.20	100.00
4	Yes	3.98		7.94	88.08	100.00
Max.		4.33	2.27	8.60	92.20	
Min.		3.31	2.20	7.80	85.42	

All results in compound%



“Normal” clastic-biogenic varves in Lake Nautajärvi

Processing option : Oxygen by stoichiometry (Normalised)

Spectrum	In stats.	Na	Mg	Al	Si	K	Ca	Ti	Fe	Total	
1	Yes	6.94		23.72	62.16		7.19			100.00	Plg
2	Yes	0.92	2.35	7.53	29.55	1.97	16.73		40.95	100.00	
3	Yes	0.90		17.61	64.24	15.26			1.99	100.00	K-Fsp
4	Yes	9.43		19.20	69.14		2.23			100.00	Alb
5	Yes	8.22		23.16	63.88		4.74			100.00	Alb
6	Yes				100.00					100.00	Qtz
7	Yes		8.21	18.75	39.52	1.13	0.84	2.43	29.13	100.00	Amph
Max.		9.43	8.21	23.72	100.00	15.26	16.73	2.43	40.95		
Min.		0.90	2.35	7.53	29.55	1.13	0.84	2.43	1.99		

All results in compound%

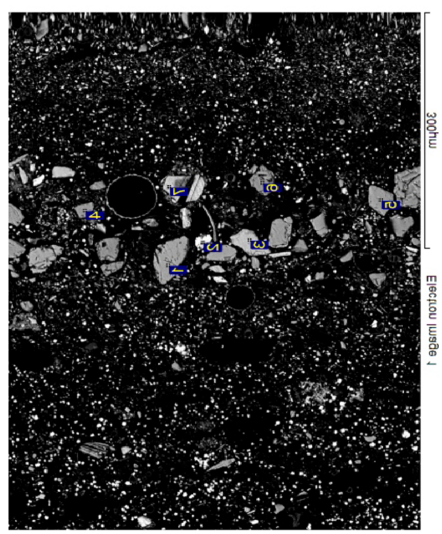


Figure 2: SEM images of the colloidal material deposited in the HTM. Analysis on the clastic laminae demonstrates that they represent plagioclase and potassium feldspars, consistent with the catchment geology. 34

3.3 Co-variance matrices

This section presents

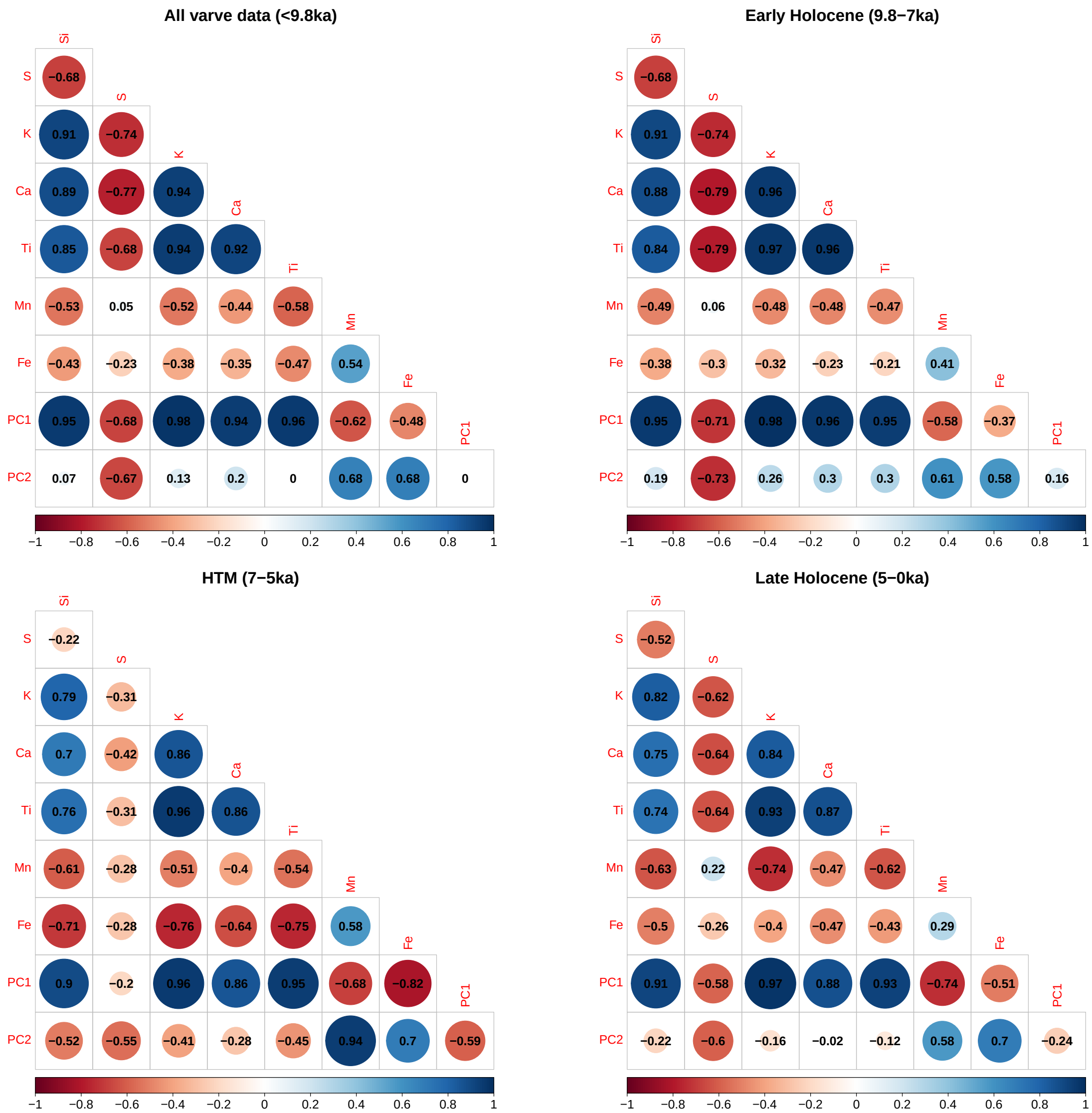


Figure 3: Co-variance matrices showing the Pearson correlation coefficient between each pairwise elemental combination and PC1-2 for A. all varved data; B. The Early Holocene (Phase 1); C. The HTM (Phase 2); D. the Late Holocene (Phase 3)

3.4 NBClust results

This section lists the code used to run the NBClust of Charrad et al. (2014) in R to objectively determine the best number of clusters to use for the NAU-23 dataset. The summary produced lists the number of clusters selected by each methodology and assigns the optimum number of clusters to use.

```
           Number_clusters   Value_Index
KL                       6  3.397000e+00
CH                       4  1.798020e+04
Hartigan                 6  2.580191e+03
CCC                      4 -8.747960e+01
Scott                    6  1.223032e+04
Marriot                  6  1.641167e+26
TrCovW                   6  1.136795e+08
TraceW                   6  5.814667e+03
Friedman                 6  5.277100e+00
Rubin                    6 -1.883000e-01
Cindex                  10  1.223000e-01
DB                       4  1.257600e+00
Silhouette               4  2.661000e-01
Beale                    4  1.706500e+00
Ratkowsky                4  3.905000e-01
Ball                     5  6.294946e+03
PtBiserial               4  5.238000e-01
Frey                     4  3.025500e+00
McClain                  4  1.054900e+00
Dunn                     4  1.270000e-02
SDindex                  4  2.203800e+00
SDbw                    10  5.393000e-01
Among all indices:
===================
* 11 proposed  4 as the best number of clusters
* 1 proposed  5 as the best number of clusters
* 8 proposed  6 as the best number of clusters
* 0 proposed  7 as the best number of clusters
* 0 proposed  8 as the best number of clusters
* 0 proposed  9 as the best number of clusters
* 2 proposed  10 as the best number of clusters

Conclusion
=========================
```

```
* According to the majority rule, the best number of clusters is  4 .
```

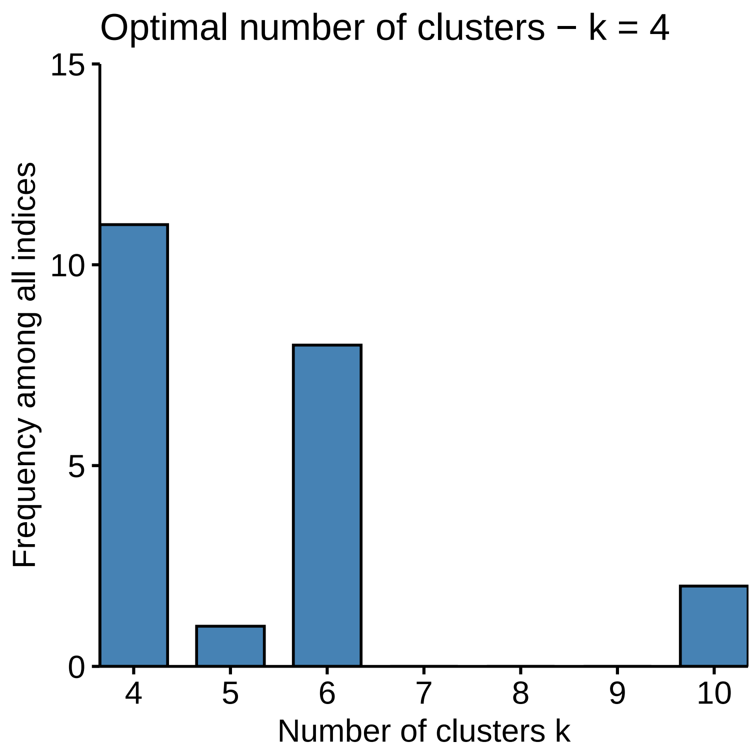


Figure 4: Barplot showing the modal number of clusters selected by NBClust for the XRF NAU-23 varved dataset. The majority rule shows 4 to be the preferred cluster number.

3.5 PCA statistics

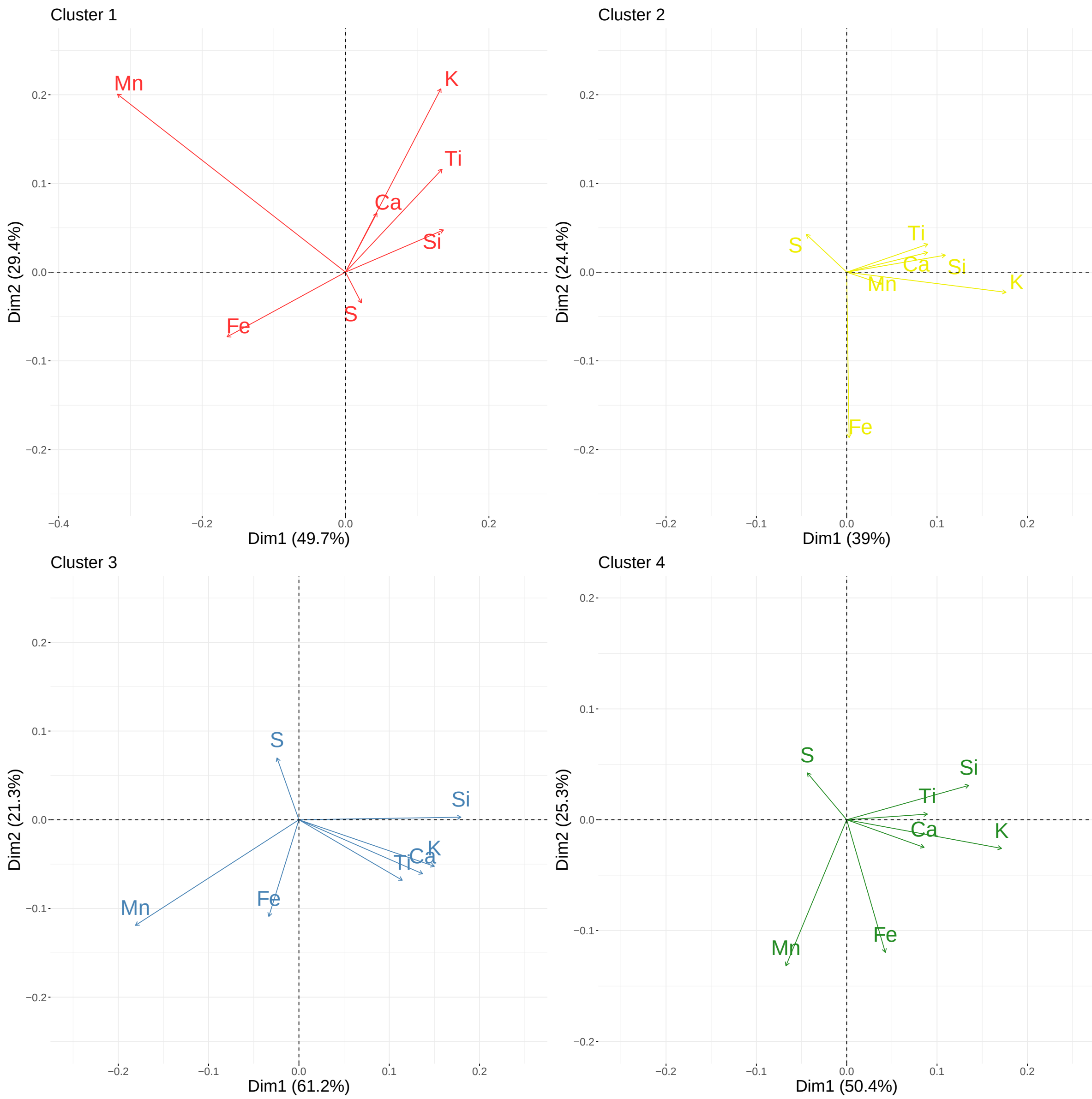


Figure 5: Individual PCA biplots for the four NAU-23 hierarchical clusters.

Table S1: Eigenvalues and Eigenvectors from the PCA analyses run on the NAU-23 sequence

	PC1	PC2	PC3	PC4	PC5	PC6	PC7
Eigenvalues and Variance %							
Eigen Value	0.48	0.07	0.03	0.02	0.01	0.00	0.00
Variance %	78.77	11.74	4.78	3.04	1.05	0.58	0.06
Cumulative Variance %	78.77	90.50	95.28	98.32	99.36	99.94	100.00
Eigenvectors							
Ca	0.44	-0.13	-0.15	0.15	0.81	-0.29	0.05
Fe	-0.21	-0.65	0.57	-0.02	0.00	-0.25	0.37
K	0.45	-0.07	-0.03	0.06	-0.49	-0.68	-0.29
Mn	-0.28	-0.48	-0.80	0.00	-0.11	-0.09	0.17
S	-0.32	0.56	-0.06	-0.03	0.04	-0.53	0.54
Si	0.43	-0.02	-0.07	-0.82	-0.08	0.12	0.35
Ti	0.44	0.02	-0.04	0.55	-0.27	0.29	0.58

Table S2: Contributions(%) of the Variables to the Principal Components

	PC1	PC2	PC3	PC4	PC5
Ca	12.75	3.87	0.43	3.11	45.13
Fe	2.62	35.11	54.69	0.73	0.09
K	35.45	4.37	0.33	8.21	44.72
Mn	6.21	49.34	41.96	0.19	0.64
S	0.98	6.36	1.30	0.02	2.55
Si	27.51	0.95	1.28	64.49	0.33
Ti	14.48	0.00	0.01	23.24	6.53

Table S3: Correlation between PCs and Variables

	PC1	PC2	PC3	PC4	PC5
Ca	0.94	0.20	0.04	0.09	0.20
Fe	-0.48	0.68	-0.54	-0.05	0.01
K	0.98	0.13	-0.02	0.09	-0.13
Mn	-0.62	0.68	0.40	0.02	-0.02
S	-0.68	-0.67	0.19	-0.02	-0.13
Si	0.95	0.07	0.05	-0.29	0.01
Ti	0.96	0.00	0.01	0.24	0.07

Charrad, Malika, Nadia Ghazzali, Véronique Boiteau, and Azam Niknafs. 2014. "NbClust: An R Package for Determining the Relevant Number of Clusters in a Data Set." *Journal of Statistical Software* 61 (November): 1–36. https://doi.org/10.18637/jss.v061.i06.

Helama, Samuli, Markus Stoffel, Richard J. Hall, Phil D. Jones, Laura Arppe, Vladimir V. Matskovsky, Mauri Timonen, Pekka Nöjd, Kari Mielikäinen, and Markku Oinonen. 2021. "Recurrent Transitions to Little Ice Age-Like Climatic Regimes over the Holocene." *Climate Dynamics* 56 (11): 3817–33. https://doi.org/10.1007/s00382-021-05669-0.

Kosonen, Emilia, Antti E. K Ojala, E Silvester, P. Francis, and J Kuva. 2023. "Extracting Seasonal Climate Signals from Land-Use Events with Updated Lake Nautajärvi (Finland) Clastic-Biogenic Varve Section." In *XXI INQUA Congress – Time for Change, 2023 Jul 13-20, Rome, Italy. Abstract Nr. 2191*.

Ojala, Antti E. K., and Teija Alenius. 2005. "10000 Years of Interannual Sedimentation Recorded in the Lake Nautajärvi (Finland) Clastic–organic Varves." *Palaeogeography, Palaeoclimatology, Palaeoecology* 219 (3): 285–302. https://doi.org/10.1016/j.palaeo.2005.01.002.

Salonen, J. Sakari, Niina Kuosmanen, Inger G. Alsos, Peter D. Heintzman, Dilli P. Rijal, Frederik Schenk, Freja Bogren, et al. 2024. "Uncovering Holocene Climate Fluctuations and Ancient Conifer Populations: Insights from a High-Resolution Multi-Proxy Record from Northern Finland." *Global and Planetary Change* 237 (June): 104462. https://doi.org/10.1016/j.gloplacha.2024.104462.

Sejrup, Hans Petter, Heikki Seppä, Nicholas P. McKay, Darrell S. Kaufman, Áslaug Geirsdóttir, Anne de Vernal, Hans Renssen, Katrine Husum, Anne Jennings, and John T. Andrews. 2016. "North Atlantic-Fennoscandian Holocene Climate Trends and Mechanisms." *Quaternary Science Reviews*, Special issue: PAST gateways (palaeo-arctic spatial and temporal gateways), 147 (September): 365–78. https://doi.org/10.1016/j.quascirev.2016.06.005.

Sicre, Marie-Alexandrine, Bassem Jalali, Jón Eiríksson, Karen-Luise Knudsen, Vincent Klein, and Violaine Pellichero. 2021. "Trends and Centennial-Scale Variability of Surface Water Temperatures in the North Atlantic During the Holocene." *Quaternary Science Reviews* 265 (August): 107033. https://doi.org/10.1016/j.quascirev.2021.107033.

Van Dijk, Evelien J. C., Johann Jungclaus, Michael Sigl, Claudia Timmreck, and Kirstin Krüger. 2024. "High-Frequency Climate Forcing Causes Prolonged Cold Periods in the Holocene." *Communications Earth & Environment* 5 (1): 242. https://doi.org/10.1038/s43247-024-01380-0.

Weltje, Gert, Menno Bloemsma, Rik Tjallingii, D Heslop, Ursula Röhl, Ian Croudace, and I Croudace. 2015. "Prediction of Geochemical Composition from XRF Core Scanner Data: A New Multivariate Approach Including Automatic Selection of Calibration Samples and Quantification of Uncertainties." In, 507–34. https://doi.org/10.1007/978-94-017-9849-5_21.