forests

MDPI

Article

Effects of Climate Change and Fire on the Middle and Late Holocene Forest History in Yenisei Siberia

Elena Novenko [1,2], Olga Rudenko [3], Natalia Mazei [2], Dmitriy Kupriyanov [1,4], Rodion Andreev [2], Anton Shatunov [1], Maria Kusilman [5], Anatoly Prokushkin [6,7] and Alexander Olchev [8,*]

1 Department of Quaternary Paleogeography, Institute of Geography of the Russian Academy of Sciences, Staromonetny lane, 29, Moscow 119017, Russia; eynovenko@igras.ru or lenanov@mail.ru (E.N.); dmitriykupriyanov1994@yandex.ru (D.K.); toxavilli@yandex.ru (A.S.)
2 Department of Physical Geography and Landscape Science, Faculty of Geography, Lomonosov Moscow State University, GSP−1, Leninskie Gory, 1, Moscow 119991, Russia; natashamazei@mail.ru (N.M.); dorionio40@gmail.com (R.A.)
3 Department of Geography, Ecology and General Biology, Orel State University Named After I.S. Turgenev, Komsomol'skaya Str., 95, Orel 302026, Russia; olrudenko2011@yandex.ru
4 Context Anthropology Laboratory, Institute of Archaeology of the Russian Academy of Sciences, Dmitriya Ulyanova St., 19, Moscow 117292, Russia
5 Department of Cartography and Geoinformatics, Faculty of Geography, Lomonosov Moscow State University, GSP−1, Leninskie Gory, 1, Moscow 119991, Russia; kusilman@rambler.ru
6 Laboratory of Biogeochemical Cycles in Forest Ecosystems, V.N. Sukachev Institute of Forest SB RAS, Federal Research Center "Krasnoyarsk Science Center SB RAS", Akademgorodok 50/28, Krasnoyarsk 660036, Russia; prokushkin@ksc.krasn.ru
7 School of Ecology and Geography, Siberian Federal University, Svobodny av., 79, Krasnoyarsk 660041, Russia
8 Department of Meteorology and Climatology, Faculty of Geography, Lomonosov Moscow State University, GSP−1, Leninskie Gory, 1, Moscow 119991, Russia
* Correspondence: aoltche@yandex.ru or aoltche@gmail.com

Supplementary Materials

S1. R Code for the modern analogue technique

```
# Package loading
library('analogue')
library('readxl')
library('tidyverse')

## This script does reconstruction using 'analogue' package by G.L. Simpson.
## method is Modern Analogue Technique (MAT) or best modern analogues (BMA).
## Training set is in variable "modern", which has pollen spectra, prediction
## set is in variable "fossil", which contains fossil spectra.
## Environmental parameters are in "parameters_modern". For column naming and
```

```
## to provide multiple parameters (working by cycle operator) vector
## "parameters_types" is loaded.
## Variable "mat" contains reconstructed parameters.

# Prediction set loading (example for data, used for Igarka reconstruction)
fossil = read_excel('%FilePath%') |>
  # Following transformations are specific for used data (needs to transpose)
  gather(variable, value, -depth) |>
  spread(depth, value) |>
  transform(variable = as.numeric(variable)) |>
  arrange(variable)

# Modern data (testing set)
modern = read_excel('%FilePath%') |>
  as.data.frame()|>
  arrange(points)

# Parameters
# Woody cover
cover = read_excel('%FilePath%') |>
  as.data.frame() |>
arrange(points)

# Climate
climate = read_excel('%FilePath%') |>
arrange(points)

# Column names switching (it's better if they will be identical)
row.names(modern) = modern$points
row.names(fossil) = fossil$variable
colnames(fossil) = colnames(modern)
```

```
# Creating data frame with modern parameters
parameters_modern = semi_join(climate, modern, by = 'points') |>
inner_join(cover)

# Ages loading (they are separated from initial spectra data )
ages = read_excel('%FilePath%', col_names = F) |>
pull()

# Data frames join
dat = join(modern, fossil, verbose = T)

# For MAT data needs to be proportional
modern = dat$modern / 100
fossil = dat$fossil / 100
set.seed(1234)

# MAT transfer function
# Transfer function & leave-one-out (LOO) or bootstrap cross-validation (CV)
i = 0 # Increment
mat = data.frame(ages) # Data frame creation

# Cycle operator provides
for (i in 1:length(parameters_types)){
# Parameter loading into vector
parameter = pull(parameters_modern, var = parameters_types[i])

func_mat = mat(modern, parameter,
method = "SQchord") # Transfer function
mat_boot = bootstrap(func_mat, n.boot = 100) # Bootstrap CV (not used) recon_mat = predict(func_mat, fossil,
k = getK(func_mat)) # Prediction, k gets from LOO CV
```

```
# Fitted values extraction in vector variable
parameter1 = recon_mat[["predictions"]][["model"]][["predicted"]][func_mat[["standard"]][["k"]],]
mat = cbind(mat, parameter1)

# Parameter-specific column names
colnames(mat)[length(mat)] = paste0(parameters_types[i], '.mat')
}
```