Circulation Weather Typing with `climate4R`: GCM evaluation with Reanalysis using Lamb Weather Types

Paper notebook - submitted to Climate Dynamics

J. A. Fernández, A. Casanueva, J. Bedia & J. Fernández

2020-07-04

Abstract

This is an example notebook illustrating the calculations undertaken in the paper. It is not intended to provide full reproducibility of the results, but a sample on how to achieve this using the climate4R framework, used in the paper. To this aim, we provide an example using public datasets only, namely the NCEP-NCAR Reanalisys1 (NNRP) and the CMIP5 simulations of the EC-EARTH model, considering the historical and RCP8.5 experiments and the ensemble member *r12i1pi*.

Contents

1 Used packages

To ensure the reproducibility of the paper results as accurately as possible, it is recommended to install the package versions used to compile this notebook. The appropriate package versions are indicated here through their version tags using the `devtools` package function `install_github` (Wickham *et al.* 2020), or alternatively, their commit has:

```
devtools::install_github(c("SantanderMetGroup/loadeR.java@v1.1.1",
                           "SantanderMetGroup/climate4R.UDG@0.1.1",
                           "SantanderMetGroup/loadeR@1.6.1",
                           "SantanderMetGroup/transformeR@7005f67",
                           "SantanderMetGroup/visualizeR@v1.6.0"))
```

Alternatively, and updated image of the packages can be installed using the conda recipe for climate4R.

1.1 Cloud computing with the climate4R Hub

Furthermore, there is a docker `climate4R` installation available. The docker file also includes the jupyter framework enabling a direct usage of `climate4R` via the **climate4R Hub**, a cloud-based computing facility to run `climate4R` notebooks on the cloud using the IFCA/CSIC Cloud Services).

The `climate4R` packages used in this experiment are next loaded:

```
require(loadeR)
require(transformeR)
```

```r
require(visualizeR)
```

Additional packages will be used for convenience. For instance, the package `magrittr` (Bache and Wickham 2014) allows to conveniently concatenate functions via the pipe operator `%>%`. In addition, the `philentropy` package is used for calculating KL Divergences (Drost, 2018). `sp` (Pebesma and Bivand 2005, Bivand *et al.* 2013) and `lattice` (Sarkar 2008) are used for plotting.

```r
require(magrittr)
require(philentropy)
require(sp)
require(lattice)
```

2 Analysis of Lamb Weather Types from NCEP reanalysis

We next load the required datasets. First of all, the NCEP-NCAR reanalysis1 (NNRP) will be loaded, using to this aim the Santander MetGroup Climate Data Service (User Data Gateway, UDG), and the package loadeR for remote access.

Note that prior to remotely accessing the UDG, login is required. To obtain credentials, pease visit the Thredds Administration Panel (TAP). These components are further described in Cofiño *et al.* 2018 and Iturbide *et al.* 2019.

```r
loginUDG(username, password)
```

The `lonLim` and `latLim` vectors are used in the following to consider the Euro-CORDEX domain as bounding box for data load. We consider the period 1981-2010, following the WMO guidelines on the calculation of climate normals (WMO, 2017).

```
wmo.years <- 1981:2010
lonLim = c(-45, 66)
latLim = c(22, 73)
```

2.1 Loading reanalysis data

```
var <- "slp"
dataset <- "http://meteo.unican.es/tds5/dodsC/ncepReanalysis1/ncepReanalysis1_4xDaily
ncep <- loadGridData(dataset = dataset,
                     var = var,
                     lonLim = lonLim,
                     latLim = latLim,
                     season = c(12, 1:11),
                     years = wmo.years,
                     time = "DD",
                     aggr.d = "mean")
```

The function `clusterGrid` of package `transformeR` is the workhorse for the application of clustering methods to climate datasets. Here, we indicate the Lamb Weater Typing through argument `type = "lamb"`. The default options are fine to compute the LWTs as presented in this paper.

```
clusters.ncep <- clusterGrid(ncep, type = "lamb")
```

```r
## Figure 1: SpatialPlot of annual climatologies from 8-LWTs-subset of NCEP:

wt.names <-  c("A", "ANE", "AE", "ASE", "AS", "ASW", "AW", "ANW", "AN",
               "NE",  "E", "SE",  "S",  "SW",  "W",  "NW",  "N",
               "C", "CNE", "CE", "CSE", "CS", "CSW", "CW", "CNW", "CN")

#grid of points from Lamb "cross":
centerlon = -5
centerlat = 55
lon.array <- rep(centerlon, times = 16) + c(-5, 5, -15, -5, 5, 15, -15, -5, 5, 15,
                                            -15, -5, 5, 15, -5, 5)
lat.array <- rep(centerlat, times = 16) + c(10, 10, 5, 5, 5, 5, 0, 0, 0, 0,
                                            -5, -5, -5, -5, -10, -10)
coords <- cbind(lon.array, lat.array) %>% sp::SpatialPoints()
l.points <- list("sp.points", coords, col = 1)

subsetLWT <- c(1, 18, 15, 14, 16, 13, 7, 17)
names.subset <- c("A", "C", "W", "SW", "NW", "S", "AW", "N")
freqLWT <- getWT(clusters.ncep) %>% names() %>% table() %>% prop.table()

#To set freqLWTs[i] = 0 when LWTs 'i' does not occur:
freqLWT <- freqLWT[match(wt.names, names(freqLWT))]
freqLWT[which(is.na(freqLWT))] <- 0
names(freqLWT) <- wt.names
```

```
freqLWT <- round(freqLWT * 100, 2)

names.attr <- paste0(names.subset, ": ", freqLWT[subsetLWT], "%")

LWTs.list <- lapply(1:8, function(x) {
  suppressMessages(climatology(subsetGrid(clusters.ncep, cluster = subsetLWT[x])))
})

LWTs.mg <- makeMultiGrid(LWTs.list, skip.temporal.check = TRUE)
dev.new()
breaks <- seq(99300, 103300, 200)
colorkey.labels <- c(99300, 99800, 100300, 100800, 101300, 101800, 102300, 102800, 13
```

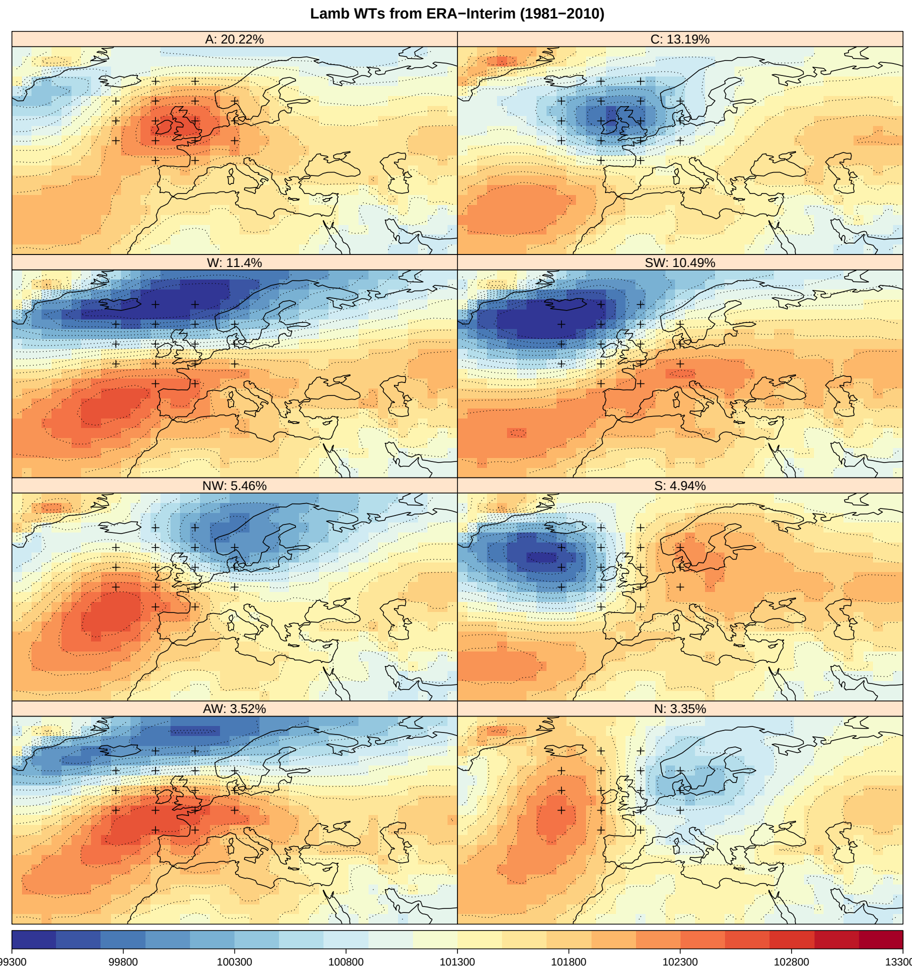
The function `spatialPlot` from package `visualizeR` (Frías *et al.* 2018) is a wrapper for the `spplot` method in package `sp`, thus accepting the many possible arguments of the lattice framework for fine tuning of the plot. Here, we reproduce the paper Fig. 1 as accurately as possible:

```
visualizeR::spatialPlot(LWTs.mg,
                        sp.layout = list(l.points),
                        backdrop.theme = "coastline",
                        rev.colors = TRUE,
                        main = "Lamb WTs from ERA-Interim (1981-2010)",
                        useRaster = TRUE,
                        set.min = min(breaks),
                        set.max = max(breaks),
                        at = breaks,
```

```
colorkey = list(space = 'bottom',
                labels = list(at = seq(99300, 103300, 500),
                              labels = colorkey.labels)),
layout = c(2,4),
as.table = TRUE,
names.attr = names.attr,
contour = TRUE,
lty = 3)
```

\begin{figure}

{



}

\caption{Composite maps of Lamb Weather Types (LWTs) derived from MSLP (Pa) from the NNRP reanalysis for the period 1981-2010. A subset of the 8 (out of 26) most frequent LWTs annually is displayed. Sub-panels are labelled with their LWT abbreviation (frequency in % in parenthesis) and sorted in decreasing frequency order from top to bottom and from left to right. Colorbar is centered on average sea-level atmospheric pressure (reds are highs andblues are lows). Lamb's cross coordinates are also indicated over the British Isles domain.}

\end{figure}

Next, the LWT frequencies as captured by the NNRN reanalysis are also displayed as a barplot, similar to paper Fig. 2:

```
# Definition of seasons:

seasons <- list(
  DJF = c(12,1,2),
  MAM = c(3,4,5),
  JJA = c(6,7,8),
  SON = c(9,10,11)
)

ncep.freqs.LWTs <- lapply(1:length(seasons), function(x) {
  grid <- subsetGrid(clusters.ncep, season = seasons[[x]])
  freqLWT <- getWT(grid) %>% names() %>% table() %>% prop.table()
  # Missing LWTs are set to zero frequency:
  freqLWT <- freqLWT[match(wt.names, names(freqLWT))]
  freqLWT[which(is.na(freqLWT))] <- 0
  names(freqLWT) <- wt.names
  freqLWT <- round(freqLWT * 100, 2)
})

seasonal.freqs <- lapply(1:length(seasons), function(x) {
  sort.int(ncep.freqs.LWTs[[x]], decreasing = TRUE)
```

```r
})

seasonal.freqs.mat <- matrix(as.numeric(unlist(seasonal.freqs)),
                             ncol = length(wt.names),
                             byrow = TRUE,
                             dimnames = list(c("DJF", "MAM", "JJA", "SON"), wt.names)

layout(matrix(c(rep(1, 6), 2), ncol = 1))
par(mai = rep(0.6, 4))
bar.colors <- c("#612C69", "#459ED5", "#FCCF61", rgb(0.3, 0.9, 0.4, 0.6))
bp <- barplot(seasonal.freqs.mat,
              beside = TRUE, col = bar.colors, border = NA, las = 1,
              ylab = "freq. [%]", ylim = c(0, 21),
              main = "NCEP-NCAR Reanalysis (NNRP) LWT frequencies")

par(mai = c(0, 0, 0, 0))
plot.new()
legend(legend = c("DJF", "MAM","JJA", "SON") ,
       fill = c("#612C69", "#459ED5", "#FCCF61", rgb(0.3, 0.9, 0.4, 0.6)),
       "center", horiz = TRUE, border = "transparent", bty = "n")
```

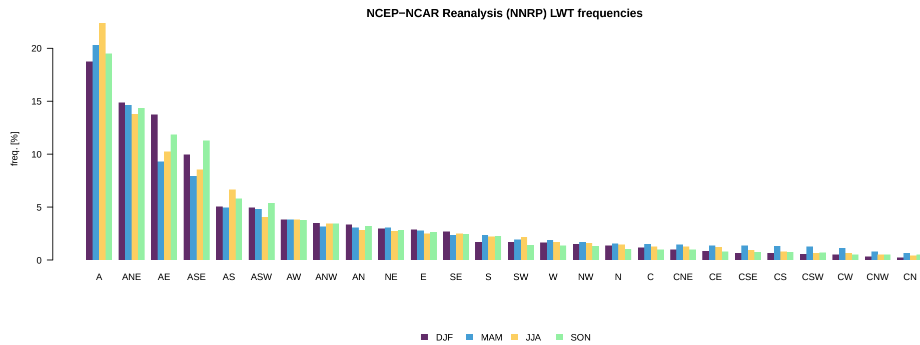


Figure 1: Comparison of the seasonal relative frequencies of Lamb Weather Types (LWTs) obtained from the NNRP reanalysis. The LWTs are sorted in decreasing order of their annual frequencies.

3 Evaluation of EC-EARTH vs. NCEP

3.1 Lamb WTs of EC-Earth:

3.1.1 Loading GCM data

```
historical = "http://meteo.unican.es/tds5/dodsC/cmip5/EC-EARTH/EC-EARTH/historical/da
rcp8.5 = "http://meteo.unican.es/tds5/dodsC/cmip5/EC-EARTH/EC-EARTH/rcp85/day/ec-eart
var <- "psl"

grid1 <- loadGridData(dataset = historical,
                      var = var,
                      lonLim = lonLim,
                      latLim = latLim,
                      years = 1980:2005,
                      time = "DD",
                      aggr.d = "mean")
```

```r
# merge of 5 years from rcp8.5 as specified in section 2.1 of the paper:

grid2 <- loadGridData(dataset = rcp8.5,
                      var = var,
                      lonLim = lonLim,
                      latLim = latLim,
                      years = 2006:2010,
                      time = "DD",
                      aggr.d = "mean")
```

Both datasets (historical and RCP8.5) are next joined with `bindGrid` along their time dimension:

```r
ec.earth <- bindGrid(grid1, grid2, dimension = "time")
ec.earth <- subsetGrid(ec.earth, season = c(12,1:11))
```

The LWTs from the EC-EARTH model are next computed:

```r
wts.ec.earth <- clusterGrid(ec.earth, type = "lamb")
```

3.2 Relative Biases between EC-Earth and NCEP seasonal LWTs

```r
ec.earth.freqs.LWTs <- lapply(1:length(seasons), function(x) {
  grid <- subsetGrid(wts.ec.earth, season = seasons[[x]])
  freqLWT <- getWT(grid) %>% names() %>% table() %>% prop.table()
  # Non existing LWTs are assigned a zero probability
  freqLWT <- freqLWT[match(wt.names, names(freqLWT))]
  freqLWT[which(is.na(freqLWT))] <- 0
```

```
  names(freqLWT) <- wt.names
  freqLWT <- round(freqLWT*100, 2)
})
names(ec.earth.freqs.LWTs) <- c("DJF", "MAM", "JJA", "SON")

rel.bias <- lapply(1:length(seasons), function(x) {
    diff.freqs <- ec.earth.freqs.LWTs[[x]][subsetLWT] - ncep.freqs.LWTs[[x]][subsetLW
    diff.freqs/ncep.freqs.LWTs[[x]][subsetLWT]
})
names(rel.bias) <- c("DJF", "MAM", "JJA", "SON")

rel.bias.mat <- matrix(as.numeric(unlist(rel.bias)),
                       ncol = length(subsetLWT), byrow = TRUE,
                       dimnames = list(c("DJF", "MAM", "JJA", "SON"),
                                       names.subset))

RColorBrewer::brewer.pal(n = 9, "RdBu") %>% rev()

## [1] "#2166AC" "#4393C3" "#92C5DE" "#D1E5F0" "#F7F7F7" "#FDDBC7" "#F4A582"
## [8] "#D6604D" "#B2182B"

pcolors <- RColorBrewer::brewer.pal(n = 9, "RdBu") %>% colorRampPalette()
levelplot(rel.bias.mat,
          ylab = "LWTs", xlab = "Season",
          main = "DJF - LWT Relative Bias",
          col.regions = rev(pcolors(201)),
          at = seq(-0.5, 0.5, 0.05))
```

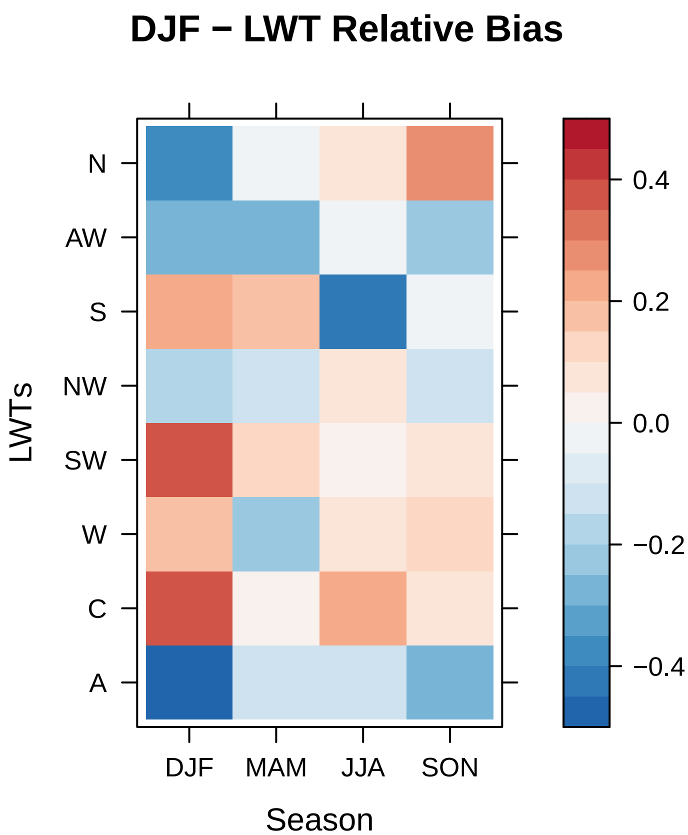


Figure 2: Seasonal relative biases of the eight main LWT frequencies as simulated by EC-EARTH, compared against the NNRP reanalysis.

3.3 Transition probabilities

3.3.1 TPM of the NNRP reanalysis

The following helper functions from this public repository are used to calculate the tansition probability matrices (TPM) and the associated score (TPMS). Note that the plot depends on the `image.plot` function from package `fields` (Nychka *et al.* 2017), that is also loaded.

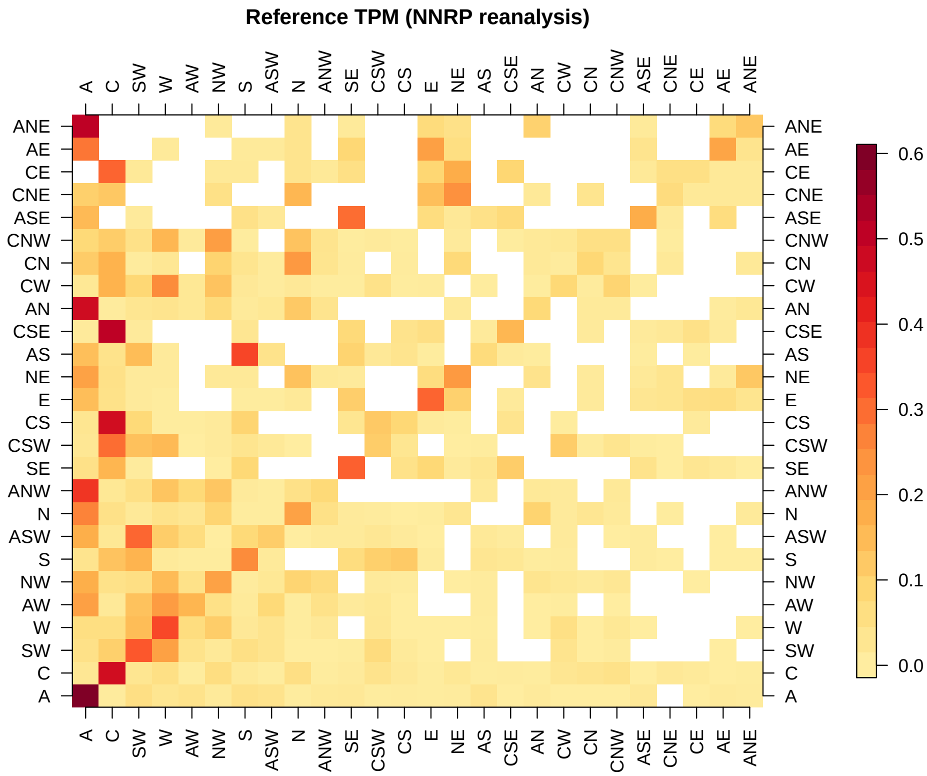


Figure 3: Transition probability matrix of the Lamb Weather Type classification, as produced by the NNRP reanalysis for the period 1981-2010.

```
require(fields)

source("https://raw.githubusercontent.com/juanferngran/TFM/master/R/tprobPlot.R")

source("https://raw.githubusercontent.com/juanferngran/TFM/master/R/transitionProb.R"

source("https://raw.githubusercontent.com/juanferngran/TFM/master/R/transitionProb.pv

source("https://raw.githubusercontent.com/juanferngran/TFM/master/R/transitionProbMat

tprobPlot(tprob.matrix = transitionProb(clusters.ncep), title = "Reference TPM (NNRP
```

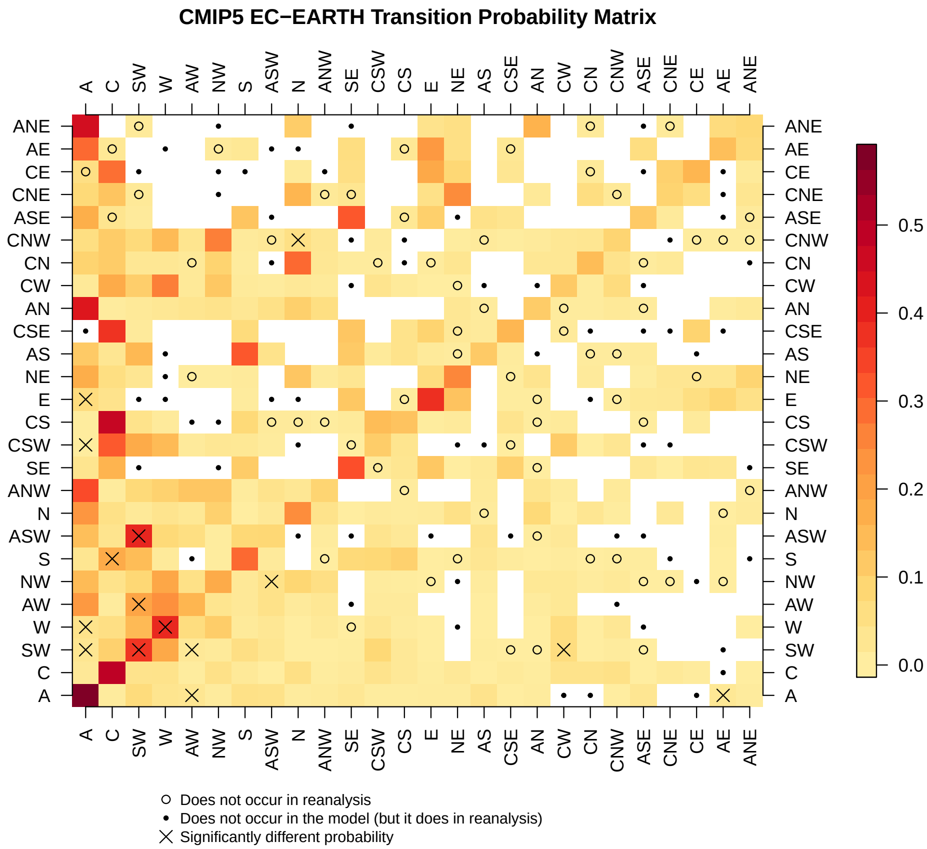


Figure 4: Transition Probability Matrix of CMIP5 EC-EARTH model, considering the Lamb Weather Type classification for the period 1981-2010, using as reference the NNRP reanalysis.

3.3.2 TPM of the EC-EARTH GCM

```
tprobPlot(tprob.matrix = transitionProb(wts.ec.earth),
          pval.matrix = transitionProb.test(obs.grid = clusters.ncep,
                                            gcm.grid = wts.ec.earth),
          tprob.ref = transitionProb(clusters.ncep),
          title = "CMIP5 EC-EARTH Transition Probability Matrix")
```

3.3.3 TPM Score calculation

The TPMS provides an overall score of the similarity between the NNRP and EC-EARTH transition probability measures:

```
TPMS(obs.wt.grid = clusters.ncep, gcm.wt.grid = wts.ec.earth, include.nonexisting = T
```

```
## [1] 1.963675
```

3.4 Kullback-Leibler Divergence between EC-Earth and NCEP annual/seasonal LWTs

Next, the KL divergence is computed, used in this study as a measure of departure between the reanalysis and the GCM representation of the Lamb Weather typing classification. To this aim, we use the from package `phylentropy` (Drost, 2018).

```
seasons <- list(
  DJF = c(12,1,2),
  MAM = c(3,4,5),
  JJA = c(6,7,8),
  SON = c(9,10,11),
  year = c(12,1:11)
)


ncep.abs.freqs <- lapply(1:length(seasons), function(x) {
  grid <- subsetGrid(clusters.ncep, season = seasons[[x]])
  freqLWT <- getWT(grid) %>% names() %>% table()
  #To set freqLWTs[i] = 0 when LWTs 'i' does not occur:
```

```r
  freqLWT <- freqLWT[match(wt.names, names(freqLWT))]
  freqLWT[which(is.na(freqLWT))] <- 0
  names(freqLWT) <- wt.names
  return(freqLWT)
})
names(ncep.abs.freqs) <- c("DJF", "MAM", "JJA", "SON", "Annual")

ec.earth.abs.freqs <- lapply(1:length(seasons), function(x) {
  grid <- subsetGrid(wts.ec.earth, season = seasons[[x]])
  freqLWT <- attr(x = grid, which = "wt.index") %>% table()
  # Non existing LWTs are assigned a zero probability
  freqLWT <- freqLWT[match(1:26, names(freqLWT))]
  freqLWT[which(is.na(freqLWT))] <- 0
  names(freqLWT) <- wt.names
  return(freqLWT)
})
names(ec.earth.abs.freqs) <- c("DJF", "MAM", "JJA", "SON", "Annual")

divergence <- sapply(1:length(seasons), function(x) {
  vector1 <- ec.earth.abs.freqs[[x]]
  vector2 <- ncep.abs.freqs[[x]]
  x <- rbind(vector1, vector2)
  philentropy::KL(x, unit = "log",  est.prob = "empirical")
})
names(divergence) <- c("DJF", "MAM", "JJA", "SON", "Annual")
```

```
print(divergence)
```

```
##        DJF        MAM        JJA        SON     Annual
## 0.06835525 0.01979763 0.02094779 0.02458057 0.01508341
```

4 References

- Cofiño, A.S., Bedia, J., Iturbide, M., Vega, M., Herrera, S., Fernández, J., Frías, M.D., Manzanas, R., Gutiérrez, J.M., 2018. The ECOMS User Data Gateway: Towards seasonal forecast data provision and research reproducibility in the era of Climate Services. Climate Services 9, 33–43. https://doi.org/10.1016/j.cliser.2017.07.001
- Drost HG., 2018. Philentropy: Information Theory and Distance Quantification with R. Journal of Open Source Software. doi:10.21105/joss.00765
- Frías, M.D., Iturbide, M., Manzanas, R., Bedia, J., Fernández, J., Herrera, S., Cofiño, A.S., Gutiérrez, J.M., 2018. An R package to visualize and communicate uncertainty in seasonal climate prediction. Environmental Modelling & Software 99, 101–110. https://doi.org/10.1016/j.envsoft.2017.09.008
- Iturbide, M., Bedia, J., Herrera, S., Baño-Medina, J., Fernández, J., Frías, M.D., Manzanas, R., San-Martín, D., Cimadevilla, E., Cofiño, A.S., Gutiérrez, J.M., 2019. The R-based climate4R open framework for reproducible climate data access and post-processing. Environmental Modelling & Software 111, 42–54. https://doi.org/10.1016/j.envsoft.2018.09.009

- Milton Bache, S. and Wickham, H., 2014. magrittr: A Forward-Pipe Operator for R. R package version 1.5. https://CRAN.R-project.org/package=magrittr
- Nychka, D., Furrer, R., Paige, J. and Sain, S., 2017. "fields: Tools for spatial data." doi: 10.5065/D6W957CT (URL: https://doi.org/10.5065/D6W957CT), R package version 10.3, <URL: https://github.com/NCAR/Fields>.
- Pebesma, E.J., R.S. Bivand, 2005. Classes and methods for spatial data in R. R News 5 (2), https://cran.r-project.org/doc/Rnews/.
- Roger S. Bivand, Edzer Pebesma, Virgilio Gomez-Rubio, 2013. Applied spatial data analysis with R, Second edition. Springer, NY. https://asdar-book.org/
- Sarkar, Deepayan, 2008. Lattice: Multivariate Data Visualization with R. Springer, New York. ISBN 978-0-387-75968-5
- Wickham, H., Hester, J. and Chang, W., 2020. devtools: Tools to Make Developing R Packages Easier. R package version 2.3.0. https://CRAN.R-project.org/package=devtools

5 Session info

```
sessionInfo() %>% print()
```

```
## R version 3.6.3 (2020-02-29)
## Platform: x86_64-pc-linux-gnu (64-bit)
## Running under: Ubuntu 18.04.4 LTS
##
```

```
## Matrix products: default
## BLAS:   /usr/lib/x86_64-linux-gnu/blas/libblas.so.3.7.1
## LAPACK: /usr/lib/x86_64-linux-gnu/lapack/liblapack.so.3.7.1
##
## locale:
##  [1] LC_CTYPE=en_US.UTF-8          LC_NUMERIC=C
##  [3] LC_TIME=es_ES.UTF-8           LC_COLLATE=en_US.UTF-8
##  [5] LC_MONETARY=es_ES.UTF-8       LC_MESSAGES=en_US.UTF-8
##  [7] LC_PAPER=es_ES.UTF-8          LC_NAME=es_ES.UTF-8
##  [9] LC_ADDRESS=es_ES.UTF-8        LC_TELEPHONE=es_ES.UTF-8
## [11] LC_MEASUREMENT=es_ES.UTF-8    LC_IDENTIFICATION=es_ES.UTF-8
##
## attached base packages:
## [1] grid      stats     graphics  grDevices utils     datasets  methods
## [8] base
##
## other attached packages:
##  [1] fields_10.3         maps_3.3.0          spam_2.5-1
##  [4] dotCall64_1.0-0     lattice_0.20-41     sp_1.4-2
##  [7] philentropy_0.4.0   magrittr_1.5        visualizeR_1.6.0
## [10] transformeR_2.0.2   loadeR_1.7.0        climate4R.UDG_0.2.0
## [13] loadeR.java_1.1.1   rJava_0.9-12
##
## loaded via a namespace (and not attached):
##  [1] padr_0.5.0              RcppEigen_0.3.3.7.0
```

```
##  [3] zoo_1.8-6               akima_0.6-2.1
##  [5] xfun_0.10               pbapply_1.4-2
##  [7] tcltk_3.6.3             colorspace_1.4-1
##  [9] mapplots_1.5.1          htmltools_0.4.0
## [11] yaml_2.2.0              rlang_0.4.4
## [13] sm_2.2-5.6              easyVerification_0.4.4
## [15] CircStats_0.2-6         RColorBrewer_1.1-2
## [17] lifecycle_0.1.0         stringr_1.4.0
## [19] verification_1.42       munsell_0.5.0
## [21] codetools_0.2-16        evaluate_0.14
## [23] latticeExtra_0.6-28     knitr_1.25
## [25] parallel_3.6.3          Rcpp_1.0.4.6
## [27] scales_1.1.0            abind_1.4-5
## [29] kohonen_3.0.10          digest_0.6.24
## [31] stringi_1.4.6           dtw_1.21-3
## [33] tools_3.6.3             bitops_1.0-6
## [35] RCurl_1.98-1.2          proxy_0.4-23
## [37] SpecsVerification_0.5-2 MASS_7.3-51.6
## [39] Matrix_1.2-18           data.table_1.12.6
## [41] rmarkdown_1.16          R6_2.4.1
## [43] boot_1.3-25             vioplot_0.3.2
## [45] compiler_3.6.3
```