# Rcost Introduction

Reto Stauffer
University of Innsbruck

October 3, 2017

**Abstract**

This package contains a user-friendly and handy interface to the COST Action 733 WLK (Wetterlagenklassifikation) algorithm. The motivation of the COST Action 733 was to "achieve a general numerical method for assessing, comparing and classifying weather situations in Europe, scalable to any European (sub)region with time scales between 12 hours and 3 days and spatial scales of ca. 200 to 2 000 km."

Details can be found on `http://cost733.met.no/`. This R package interfaces one routine which was developed during the COST 733 Action with some minor modifications. Not all tuning parameters are implemented yet. Implementation of these parameters should be feasible if needed.

## 1 COST Action 733: WLK

The WLK (Wetterlagenklassifikation) algorithm is a relatively simple grid-based method to classify large-scale weather patterns. The `Rcost` package interfaces the fortran code `cost733_wlk_1.07.f90` which can be found here: http://cost733class.geo.uni-augsburg.de/moin/cost733wiki/data/pages_bak/WLKC733/

The analysis uses the two wind speed components (`u` zonal, `v` meridional) in 700 hecto Pascal, the (geopotential) height of the 500 hecto Pascal and 925 hecto Pascal surface plus (`z` in our case) the total column water content (`tcw`) of a pre-specified (sub)region.

Thus, the classification can easily be performed on analysis or re-analysis data (as in the package example), but also on numerical weather prediction (NWP) forecasts. Please note that only regular latlong grids are supported!

## 2 Input Data

The main method to retrieve the WLK classification is `getClassification(...)` which expects the input data as two NetCDF files. Two files are required as surface level data and pressure level data cannot be mixed (at least not when using `grib_to_netcdf` from the ECMWF GribAPI library I used).

`Rcost` contains two example files which can be loaded via `demofiles()`:

```
> ## Loading demo data (as NetCDF)
> library("Rcost")
> nc <- demofiles()
```

```
Surface file:        /usr/local/lib/R/site-library/Rcost/extdata/ITERIM_sfc.nc
Pressure level file: /usr/local/lib/R/site-library/Rcost/extdata/ITERIM_pl.nc

> print( names(nc) )

[1] "sfc" "pl"
```

The function `demofiles()` returns a `list` containing the file names (`character` strings) of the two demo files.

# 3 Parameters and Settings

COST 733 WLK provides a set of "tuning parameters", options, or settings. Not all of them are implemented yet in the `Rcost` package, but—if required—adding them should not be a big deal.

Implemented settings or options:

- `steps`: allows to subset in time (only take some steps out of all time steps provided by the input data files).
- `subset`: an `extent` object to specify an areal subset on which the classification should be performed. If not given, the whole are as provided by the input files will be used.
- `pwclim`: precipitable water climatology (warning: that's the original COST 733 name, while I am currently using total column water content (`tcw`); be aware of this). If given a binary dry/wet class will be returned.
- `weights`: weights for the 'core', 'mid', and 'margin' area (see below).
- `nsector`: number of wind sectors for the classification.
- `maindirthreshold`: threshold of (weighted) grid cells to have the same wind direction for a wind sector to be choosen. If none of the wind sectors exceed this threshold no main wind direction will be classified (variable).
- `zamg`: special mode as used by the ZAMG[1]. Uses pre-specified `subset`, `weights`, and tailored wind sectors for the European Alps.

## 3.1 Subset and Weights

The classification uses a (squared) centered weighting mask for the area of interest. This is done by specifying a weight mask, typically done internally (in `getClassification(...)` given the inputs.

For demonstration purposes these masks can also be created manually using the `weightmask(...)` function of the package. The extent is here defined by `lon` and `lat`, the weights for the three different segments by vector `weights`. `stripsize` can be used to change the weighting scheme. By default `stripsize=5`: the area is split by 5 along longitude and latitude. The outer $1/5th$ of the area then belongs to 'margin', the next $1/5th$ to 'mid' and $1/5 \times 1/5$ defines the core.

Note that, if you set `getClassification(...,zamg=TRUE)` `weightmaskZAMG(...)` instead of `weightmask(...)` is used.

[1] Zentralanstalt für Meteorologie und Geodynamik, Vienna, Austria

```
> ## Helper function: Converts weight mask into raster
> wm2raster <- function( x, grid, name ) {
+    attach(grid); delta <- median(diff(lon))
+    res <- raster( x[nrow(x):1,], xmn=min(lon)-delta/2, xmx=max(lon)+delta/2,
+                                  ymn=min(lat)-delta/2, ymx=max(lat)+delta/2 )
+    names(res) <- name; return( res )
+ }
> ## Sepcify weight mask for a grid 10E40N to 20E50N with
> ## a grid spacing of 0.25 degrees (regular ll)
> grid <- list( lon = seq(20, 40, by = 0.25),
+               lat = seq(30, 50, by = 0.25) )
> ## Specify the weights for:
> ## - core segment
> ## - mid segment
> ## - margin segment
> weights <- c(15,2,1)
> ## Get weight mask
> w1 <- weightmask( grid$lon, grid$lat, weights )
> ## We can, of course, also use different weights
> w2 <- weightmask( grid$lon, grid$lat, c(10,5,3) )
> ## Or different stripsize
> w3 <- weightmask( grid$lon, grid$lat, c(10,5,3), stripsize = 7 )
> w4 <- weightmask( grid$lon, grid$lat, c(10,5,3), stripsize = 11 )
> ## Create a RasterStack object for plotting purposes.
> library("raster")
> res <- stack( wm2raster( w1, grid, "weightmask_w1" ),
+               wm2raster( w2, grid, "weightmask_w1" ),
+               wm2raster( w3, grid, "weightmask_w1" ),
+               wm2raster( w4, grid, "weightmask_w1" ) )
> ## Plot
> plot( res, nc = 4, col=rev(heat.colors(20))[6:20], asp = NA )
```

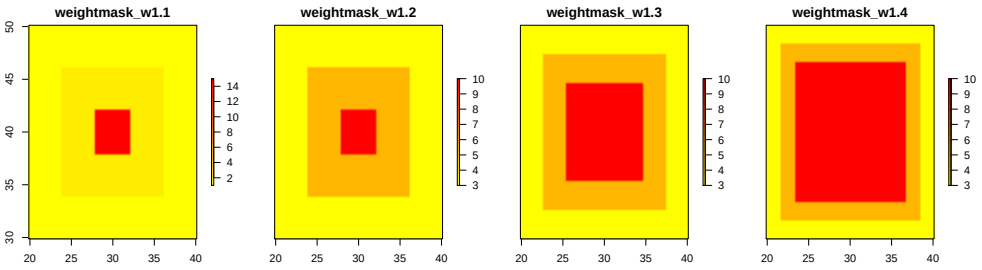


Figure 1: Example of four different weighting mask specifications. Typically done inside the getClassification method.

Warning: please note that the `zamg = TRUE` mode works with a hardcoded sub-region! This mode is designed for the Alps and can only be applied there. Let's try what happens if we try to use a sub-region different than 1.5E40.5N to 25.5E54.5N with `weightmaskZAMG(...)`:

```
> ## Create RasterStack again, once with the default weight mask,
> ## once with the weight mask for the ZAMG mode.
> ## WARNING: zamg = TRUE uses 1.5E40.5N to 25.5E54.5N (hardcoded)
> grid <- list( lon = seq( 1.5, 25.5, by = 0.5),
+               lat = seq(40.5, 54.5, by = 0.5) )
> w1 <- weightmask( grid$lon, grid$lat )
> wZ <- weightmaskZAMG( grid$lon, grid$lat )
> ## Create a RasterStack object for plotting purposes.
> res <- stack( wm2raster( w1, grid, "weight_cost" ),
+               wm2raster( wZ, grid, "weight_ZAMG" ) )
> names(res) <- c("weightmask","weightmask_ZAMG")
> ## Plot
> plot( res, nc = 2, col=rev(heat.colors(20))[6:20], asp = NA )
```

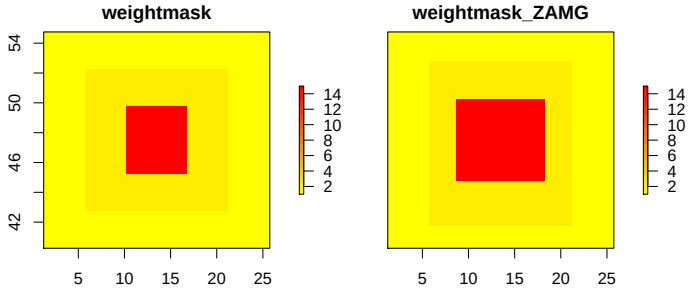


Figure 2: foo

```
> ## Create RasterStack again, once with the default weight mask,
> ## once with the weight mask for the ZAMG mode.
> grid <- list( lon = seq( 10, 30, by = 0.5),
+               lat = seq( 45, 60, by = 0.5) )
> w1 <- weightmask( grid$lon, grid$lat )
> wZ <- weightmaskZAMG( grid$lon, grid$lat )
> ## Create a RasterStack object for plotting purposes.
> res <- stack( wm2raster( w1, grid, "weightmask_cost" ),
+               wm2raster( wZ, grid, "weightmask_ZAMG" ) )
> ## Plot
> plot( res, nc = 2, col=rev(heat.colors(20))[6:20], asp = NA )
```

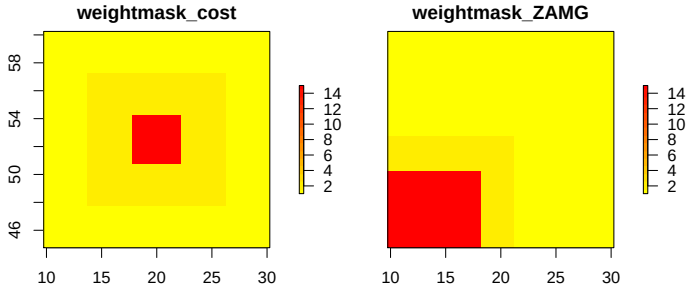


Figure 3: Weight mask using the cost weightmask function (left) and ZAMG weightmask function (right). Note that the ZAMG weight mask uses hardcoded longitude/latitude limits and is not centered anymore!

# 4 Run Classification

The only function needed (beside `nc_open` and `nc_close` from the `ncdf4` package) is `getClassification`. `getClassification` takes care of identifying the correct fields in the input files, the spatial extent and the subsetting (if required), and the weights.

```
> ## Getting reuqired packages
> library("Rcost"); library("ncdf4")
> ## Get demo file location
> files <- demofiles()

Surface file:        /usr/local/lib/R/site-library/Rcost/extdata/ITERIM_sfc.nc
Pressure level file: /usr/local/lib/R/site-library/Rcost/extdata/ITERIM_pl.nc

> ## Open NetCDF file connection
> nc <- list( nc_open(files$sfc), nc_open(files$pl) )
> ## Extract correct initialization date/time (used to properly
> ## specify the output object)
> init <- as.POSIXct( ncvar_get(nc[[1]],"time")[1]*3600, origin="1900-01-01" )
> x  <- getClassification( nc, init )
> ## Close NetCDF files
> for ( n in nc ) nc_close( n )
```

The output variable `x` contains the COST 733 classification. Namely the columns:

```
> print( x )

                    yday sector cyclonic925 cyclonic500 wet       c925
2000-07-01 12:00:00  182      6           0           0 -99 -0.9964708
2000-07-02 12:00:00  183      6           1           0 -99  0.2332306
2000-07-03 12:00:00  184      6           1           0 -99  0.7174525
2000-07-04 12:00:00  185      6           1           1 -99  5.3083287
2000-07-05 12:00:00  186      6           0           1 -99 -1.8150534
                          c500      pwi
2000-07-01 12:00:00 -0.3431060 23.56324
2000-07-02 12:00:00 -0.5033953 25.19957
2000-07-03 12:00:00 -0.7103127 27.66346
2000-07-04 12:00:00  0.2180474 26.65847
2000-07-05 12:00:00  0.2089095 21.13933
attr(,"init")
[1] "2000-07-01"
attr(,"created")
[1] "2017-10-03 14:17:12 UTC"
```

The returned object `x` is of type `zoo` (see `zoo` package manual) containing the following information:

- `yday`: day of the year, $[0 - 364]$ (analog to the R `POSIXlt` specification) where `yday = 0` is the first of January.
- `sector`: main wind sector, or `0` if no wind sector was found.
- `cyclonic925`/`cyclonic500`: binary class whether the current circulation is anticyclonic (`cyclonicXXX = 0`) or cyclonic (`cyclonicXXX = 1`) on the 925 hPa and 500 hPa surface, respectively.

- `cyclonic500`: same as `cyclonic925` but for the 500 hPa level.
- `wet`: binary response whether this time step is dry (`wet = 0`) or wet (`wet = 1`) compared to a climatology. As we haven't specified `pwclim` when calling `getClassification(...)` all values will be `-99` (classification not possible).
- `c925`/`c500`: numeric value on which `cyclonic925` and `cyclonic500` are based.
- `pwi`: numeric value on which `wet` is based (if `pwclim` is given). Note: still based on total column water (`tcw`).

As we have not specified a subset the classification `x` is based on the whole domain as provided by the demo files (-15.0E15.0N to 35.2E65.2N). However, the function provides a variety of input arguments to adjust the classification and customize it. As an example:

```
> files <- demofiles(TRUE)
> nc <- list(nc_open(files$sfc),nc_open(files$pl))
> init <- as.POSIXct( ncvar_get(nc[[1]],"time")[1]*3600, origin="1900-01-01" )
```

### Step-Subset

```
> ## Only the first two steps: +12h and +36h
> ## Not sure why I count from 00 UTC from the init??
> getClassification( nc, init, steps = c(12,36) )

                    yday sector cyclonic925 cyclonic500 wet       c925
2000-07-01 12:00:00  182      6           0           0 -99 -0.9964708
2000-07-02 12:00:00  183      6           1           0 -99  0.2332306
                         c500      pwi
2000-07-01 12:00:00 -0.3431060 23.56324
2000-07-02 12:00:00 -0.5033953 25.19957
attr(,"init")
[1] "2000-07-01"
attr(,"created")
[1] "2017-10-03 14:17:12 UTC"
```

### Provide pw climatology (creates binary dry/wet)

```
> ## Specify a subset (ZAMG subset) and use ZAMG mode
> pwclim <- rep(24.5,365) # demo: constant clim for the whole year
> getClassification( nc, init, pwclim = pwclim )

                    yday sector cyclonic925 cyclonic500 wet       c925
2000-07-01 12:00:00  182      6           0           0   0 -0.9964708
2000-07-02 12:00:00  183      6           1           0   1  0.2332306
2000-07-03 12:00:00  184      6           1           0   1  0.7174525
2000-07-04 12:00:00  185      6           1           1   1  5.3083287
2000-07-05 12:00:00  186      6           0           1   0 -1.8150534
                         c500      pwi
2000-07-01 12:00:00 -0.3431060 23.56324
2000-07-02 12:00:00 -0.5033953 25.19957
2000-07-03 12:00:00 -0.7103127 27.66346
2000-07-04 12:00:00  0.2180474 26.65847
2000-07-05 12:00:00  0.2089095 21.13933
```

```
attr(,"init")
[1] "2000-07-01"
attr(,"created")
[1] "2017-10-03 14:17:12 UTC"
```

### Different number of wind sectors

```
> ## Only four wind sectors
> getClassification( nc, init, nsector = 4 )

                    yday sector cyclonic925 cyclonic500 wet       c925
2000-07-01 12:00:00  182      3           0           0 -99 -0.9964708
2000-07-02 12:00:00  183      3           1           0 -99  0.2332306
2000-07-03 12:00:00  184      3           1           0 -99  0.7174525
2000-07-04 12:00:00  185      3           1           1 -99  5.3083287
2000-07-05 12:00:00  186      3           0           1 -99 -1.8150534
                          c500      pwi
2000-07-01 12:00:00 -0.3431060 23.56324
2000-07-02 12:00:00 -0.5033953 25.19957
2000-07-03 12:00:00 -0.7103127 27.66346
2000-07-04 12:00:00  0.2180474 26.65847
2000-07-05 12:00:00  0.2089095 21.13933
attr(,"init")
[1] "2000-07-01"
attr(,"created")
[1] "2017-10-03 14:17:12 UTC"
```

### Spatial subset

```
> ## Specify a subset (using extent)
> getClassification( nc, init, subset = extent(c(1.5,25.5,40.5,54.5)) )

                    yday sector cyclonic925 cyclonic500 wet       c925
2000-07-01 12:00:00  182      6           0           0 -99 -0.9964708
2000-07-02 12:00:00  183      6           1           0 -99  0.2332306
2000-07-03 12:00:00  184      6           1           0 -99  0.7174525
2000-07-04 12:00:00  185      6           1           1 -99  5.3083287
2000-07-05 12:00:00  186      6           0           1 -99 -1.8150534
                          c500      pwi
2000-07-01 12:00:00 -0.3431060 23.56324
2000-07-02 12:00:00 -0.5033953 25.19957
2000-07-03 12:00:00 -0.7103127 27.66346
2000-07-04 12:00:00  0.2180474 26.65847
2000-07-05 12:00:00  0.2089095 21.13933
attr(,"init")
[1] "2000-07-01"
attr(,"created")
[1] "2017-10-03 14:17:13 UTC"
```

### Spatial subset in combination with ZAMG mode

```
> ## Specify a subset (ZAMG subset) and use ZAMG mode
> getClassification( nc, init, subset = extent(c(1.5,25.5,40.5,54.5)), zamg = TRUE )
```

```
                    yday sector cyclonic925 cyclonic500 wet       c925
2000-07-01 12:00:00  182      7           0           0 -99 -1.6281022
2000-07-02 12:00:00  183      7           0           0 -99 -0.6642625
2000-07-03 12:00:00  184      6           1           0 -99  0.2760319
2000-07-04 12:00:00  185      6           1           1 -99  4.9578962
2000-07-05 12:00:00  186      7           0           1 -99 -3.2255428
                          c500      pwi
2000-07-01 12:00:00 -0.3914810 23.58414
2000-07-02 12:00:00 -0.5575750 25.43368
2000-07-03 12:00:00 -0.7588967 27.96158
2000-07-04 12:00:00  0.1638929 26.75438
2000-07-05 12:00:00  0.1999336 21.10858
attr(,"init")
[1] "2000-07-01"
attr(,"created")
[1] "2017-10-03 14:17:13 UTC"
```

### Use custom weights

```
> ## Specify a subset (ZAMG subset) and use ZAMG mode
> getClassification( nc, init, weights = c(10,5,4) )

                    yday sector cyclonic925 cyclonic500 wet        c925
2000-07-01 12:00:00  182      6           0           0 -99 -1.99145341
2000-07-02 12:00:00  183      6           0           0 -99 -0.63573195
2000-07-03 12:00:00  184      6           1           0 -99  0.74285285
2000-07-04 12:00:00  185      6           1           1 -99  2.66918790
2000-07-05 12:00:00  186      6           0           1 -99 -0.02188352
                          c500      pwi
2000-07-01 12:00:00 -0.33589588 23.36183
2000-07-02 12:00:00 -0.48405983 24.82067
2000-07-03 12:00:00 -0.42354198 26.35777
2000-07-04 12:00:00  0.02290187 25.89018
2000-07-05 12:00:00  0.20040048 21.81381
attr(,"init")
[1] "2000-07-01"
attr(,"created")
[1] "2017-10-03 14:17:13 UTC"
```

### Higher ‘main wind sector‘ frequency threshold

```
> ## Specify a subset (ZAMG subset) and use ZAMG mode
> getClassification( nc, init, maindirthreshold = 0.8 )

                    yday sector cyclonic925 cyclonic500 wet       c925
2000-07-01 12:00:00  182      0           0           0 -99 -0.9964708
2000-07-02 12:00:00  183      0           1           0 -99  0.2332306
2000-07-03 12:00:00  184      0           1           0 -99  0.7174525
2000-07-04 12:00:00  185      0           1           1 -99  5.3083287
2000-07-05 12:00:00  186      6           0           1 -99 -1.8150534
                          c500      pwi
2000-07-01 12:00:00 -0.3431060 23.56324
2000-07-02 12:00:00 -0.5033953 25.19957
2000-07-03 12:00:00 -0.7103127 27.66346
2000-07-04 12:00:00  0.2180474 26.65847
```

```
2000-07-05 12:00:00  0.2089095 21.13933
attr(,"init")
[1] "2000-07-01"
attr(,"created")
[1] "2017-10-03 14:17:13 UTC"
```