Google Drive with needed input files:

https://drive.google.com/drive/u/1/folders/1mB1umEvFzYN4-NWyQZRF-QI8yr9iBoNv

GitHub with the code:

https://github.com/dancomputer/HydroMet-Clustering/tree/main

This document provides a step-by-step description of the different sections of the Python code allowing to:

1. Compute yearly growing season GDDs, Cumulative Precipitation, and yield in each pixel.
2. Compute agro-climatic regions by grouping pixels on the basis of their long-term averages of the above three "agro-climatic" variables.
3. Compute clusters of pixels, within each region, which have similar yield time-series.
4. Create a result file ('Clustering_results_v1b.nc4').

Compute yearly growing season GDDs, Cumulative Precipitation, and yield in each pixel.

- Code name: MakeBioClimVars.py
- Line 14 to 27: Load in daily gridded data on precipitation (precip), max air surface temperature (tasmax), minimum air surface temperature (tasmin) + yearly planting day (plant_day) and maturity day (mat_day). All files are located in ./Bioclim_vars .
- Line 29 to 36: Create two output files, of the same structure as the yearly gridded input data, to hold the yearly GDDs and cumulative precipitation we will calculate.
- Line 38 to 78:
    - Define a function DaysBefore(*year*) which gives the number of days that have elapsed from the start of the data in Jan. 1, 1861 up to Jan. 1, *year*. This is needed to locate each year's growing season in the daily data based on the data on planting and maturity days which is provided relative to each year.
    - Calculate growing season GDD's and cumulative precipitation using list comprehension over each growing season of each pixel.
- Line 79 to 86: Fill and save the output files as .nC4 to ./Bioclim_vars

Compute agro-climatic regions by grouping pixels on the basis of their long-term averages of the above three "agro-climatic" variables.

Compute clusters of pixels, within each region, which have similar yield time-series.

- Code name: MakeBioClimVars.py
- Line 18 to 40: Load in the data from first step + the calculated yearly growing season GDD's and cumulative precipitation.
- Line 41 to 93: Combine all the data into a dataframe which holds, for each pixel, the average over 1861-2005 of growing season GDD's, cumulative precipitation, & yield + yearly time-series of yield.
- Line 115 to 149: Convert this dataframe into a geopandas object.
    - Lines 147-149: Make an object which says if two pixels are contiguous.

- Line 150 to 232: Regionalize the data grid based on mean growing season GDD's, cumulative precipitation, and yield.
    - Try dividing the data grid into successively more regions, going from n = 1--30 regions.
    - Produce a graph showing the score associated with each n.
    - Choose the n before the highest drop off in score as the final regionalization (Lines 199-202)
    - Add the labels of these regions to the dataframe (Lines 203-232).
- Line 150 to 232: Within each region, make clusters of pixels grouped by closeness of their normalized yearly yield-timeseries.
    - For each region, attempt making from n = 1 to #pixels/5 clusters.
    - Automatically choose the n with the highest sillouhette score.
    - Label the clusters by axxx where the a in [1,6] is the region and xxx in [001, 999] is the cluster number within the region.
- Visualizations
- Line 94 to 114: Plot mean GDDs, cumulative precipitation, and yield on a map of Europe.
- Line 220 to 241: Plot the regions.
- Line 244 to 254: 3-D Scatter plot of the above quantities.
- Line 339 to 377: Plot a sampling of time-series clusters.
- Line 379 to 410: Plot the pdf of mean GDDs, cumulative precipitation, and yield for each region.

Create a result file, structured as follows:

Time dimension: same as original precip file (365/6 days x ~180 years)

Year dimension: same as original planting day file

- Lon/lat: same
- In each pixel:

pixel	Cluster	Variable 1	Variable 2	Variable …
P1	C1	Avrge(P1,P2,P3)	Avrge(P1,P2,P3)	Avrge(P1,P2,P3)
P2	C1	Avrge(P1,P2,P3)	Avrge(P1,P2,P3)	Avrge(P1,P2,P3)
P3	C1	Avrge(P1,P2,P3)	Avrge(P1,P2,P3)	Avrge(P1,P2,P3)

The variables defined on each cluster of pixels are averages of maximum air surface temperature (daily), precipitation rate (daily), planting day (yearly), maturation day (yearly).

- Code name: MakeClusteredResultNC4.py
- Line 18 to 35: Load in the data from first step.
- Line 37 to 62: Make the .nc4 result file.

- Line 63 to 68: Aggregate the data by cluster.
- Line 70 to 84: Fill the result file with the aggregated data.
-

Note on ./Bioclim_vars

1	gepic_hadgem2-es_ewembi_historical_2005soc_co2_matyday-soy-noirr_lat36.25to70.25lon-11.25to40.25_annual_1861_2005.nc4	NetCDF file of the annual planting date in each pixel for the period specified in the name. No irrigation, constant 2005 c02 scenario.
2	gepic_hadgem2-es_ewembi_historical_2005soc_co2_matyday-soy-noirr_lat36.25to70.25lon-11.25to40.25_annual_1861_2005.nc4	Same as above, but of maturity date.
3	gepic_hadgem2-es_ewembi_historical_2005soc_co2_yield-soy-noirr_europe_annual_1861_2004.nc4	Same as above, but of soy yield.
4	pr_day_HadGEM2-ES_historical_r1i1p1_EWEMBI_europe_18610101-20051231.nc4	Same clim. scenario, daily precipitation rate.
6	tasmax_day_HadGEM2-ES_historical_r1i1p1_EWEMBI_europe_18610101-20051231.nc4	Same clim. scenario, daily max. air surface temperature.
7	tasmin_day_HadGEM2-ES_historical_r1i1p1_EWEMBI_europe_18610101-20051231.nc4	Same clim. scenario, daily min. air surface temperature.

Note on the Python Environment

The above code was run using the environment *HydroMet-Clustering_Python-Environment.yml* .
To reproduce this environment, you can run the following command in your Anaconda terminal:

>> conda env create -f *HydroMet-Clustering_Python-Environment.yml*