**Downscaling and debiasing MET data**

**Script Author:** Christy Rollinson

**Online Location:** https://github.com/PalEON-Project/modeling\_met\_ensemble

**Required Repositories:** modeling\_met\_ensemble, PEcAn

* The MET downscaling workflow contains 11 different steps contained in 11+ R scripts. However, depending on what you are doing, you do not need to complete all of the steps.
* The raw MET grids should be on Cyverse for all PalEON sites. However, if you need to download the raw grids onto your personal machine, the *GetMet\_Scripts* folder on GitHub contains scripts to download the raw data from CRUNCEP, GLDAS, NLDAS, and all of the GCMs. The extraction of the raw grid files is actually faster than the site-specific extraction from the grid files.

**Information on available data variables and data sources can be found on GitHub.**

---------------------------------------------------------------------

**Step 1: Extract raw site data from the grid files**

---------------------------------------------------------------------

Scripts:

* get\_raw\_site.R

For this step, you must:

* Have access to raw LDAS (GLDAS or NLDAS) grid files stored locally
  + GLDAS must be used if state is outside of the lower 48 states.
  + NLDAS is usually extracted
* Have access to PEcAn functions

If you are working with a PalEON site, you can skip this step and use the extracted raw site data which can be found on Cyverse.

This script extracts/processes three types of MET data: LDAS, CRUNCEP, and GCM. The extracted data is stored in your specified <path.out>/<site>/<data type>. The second half of the script generates visualizations of the extracted data so that you are able to visually check to make sure nothing is weird with the data.

You must specify the type of LDAS data that you want to extract for site, as well as the latitude and longitude coordinates of your site. GLDAS stands for Global Land Data Assimilation System and NLDAS stands for North American Land Data Assimilation. NLDAS is more commonly used in this workflow.

---------------------------------------------------------------------

**Step 2: Aggregate LDAS data**

---------------------------------------------------------------------

Scripts:

* aggregate\_LDAS.R

There are no required data files or PEcAn functions required for this step.

This script aggregates the LDAS data, in this case the training data, to daily values. The data has to be at a daily resolution or the debiasing step will not work. This is why the workflow aggregates the values and then, later, re-generates the sub-daily values.

---------------------------------------------------------------------

**Step 3: Bias correction of MET data**

---------------------------------------------------------------------

Scripts:

* bias\_correct.R

\*\* Included in the repository are four other scripts (Scripts 3a-d), which perform bias correction for each of the GCM datasets. These scripts are meant to increase the parallelization potential of the step, but it is not necessary to use these scripts.

For this step, you must:

* Have access to PEcAn functions
* Have local access to CRUNCEP and GCM data (both recent and historical)
* Adjust the <ens> numbers (i.e. 1-10 to 11-20 after running one batch). Otherwise you will just overwrite your ensemble members 1-10 again and again.
* Set seeds to create reproducible results.

This script works to create a smooth daily MET product from 850 – 2010+ based on multiple data sources of varying temporal frequencies and extents. It also takes monthly variables to daily time steps. The resulting dataset is compiled in temporal sections by the different MET datasets.

|  |  |  |
| --- | --- | --- |
| **Training Data** | **Source Data** | **Years Saved** |
|  | LDAS | 1979+ |
| LDAS (1 series) | CRUNCEP (1 series) | 1901-1979 |
| CRUNCEP (n.ens series) | GCM Historical (1 series) | 1850-1900 |
| GCM Historical (n.ens series) | GCM P.1000 (1 series) | 850-1849 |

Important Notes:

* The function *align.met* lines up the training and source data sets of varying temporal resolutions and adds missing data.
* The function *debias.met.regression* has a few important arguments to mention.
  + The script crashes when argument *parallel* is set to TRUE.
  + As default, the argument *uncert.prop* is set to “mean,” which is the only option that has been used in practice.
  + The argument *pair.anoms* should be set to FALSE when working GCMs. It should be set to TRUE if working with empirical values.
* CRUNCEP has some constant values for a few variables, so do not worry if you see some horizontal lines in the data visualizations for CRUNCEP.
* Christy generally creates 50 daily ensembles per GCM, as the rejection rate (Step 4) is around 10%.

---------------------------------------------------------------------

**Step 4: Assessment of corrected daily ensembles**

---------------------------------------------------------------------

Scripts:

* check\_bias.R
* reject\_ens\_mems\_day.R

For this step, you must:

* Have access to PEcAn functions

The first script generates figures of the corrected data so that you can visually check the mean and variances of the generated daily ensembles and verify that they are more or less okay.

The second script rejects ensemble members which contain highly unlikely or impossible values by “exiling” them to a special folder where they will no longer be considered in other steps.

---------------------------------------------------------------------

**Step 5: Generation of statistical models**

---------------------------------------------------------------------

Scripts:

* generate\_tdm\_models.R

For this step, you must:

* Have access to PEcAn functions

This script generates and saves the statistical models to represent the relationship between daily and sub-daily values. It generates a model for each met variable for each day of the year. These models will be used in Step 6 to predict sub-daily values (e.g. hourly or 3-hourly) from the daily values created in Step 3 using the training dataset (e.g. NLDAS or GLDAS).

Important Notes:

* The function *gen.sub-daily.models* has a few arguments to mention.
  + The argument *day.window* denotes the number of days to be considered when predicting sub-daily values. It is set to be 7 days, but it can be adjusted depending on the dataset.
  + The argument *direction.filter* indicates which direction the model should be predicting. For PalEON runs, the direction should always be “backward,” but if you would like to forecast into the future, the direction should be set to “forward.” It is important to keep this direction consistent in the next step of the workflow.
  + The argument *n.beta* indicates the number of values saved for the distribution of each beta coefficient. These distributions are saved for each beta coefficient in the model.
* This step needs to be done only once per location as the models are saved.

---------------------------------------------------------------------

**Step 6: Prediction of sub-daily values**

---------------------------------------------------------------------

Scripts:

* predict\_sub-daily\_met.R

\*\* Included in the repository are four other scripts (Scripts 3a-d), which perform bias correction for each of the GCM datasets. These scripts are meant to increase the parallelization potential of the step, but it is not necessary to use these scripts.

For this step, you must:

* Have access to PEcAn functions

Using the statistical models generated in Step 5, this script predicts sub-daily met values from the bias-corrected daily met values found in Step 3. According to Christy, coefficient values are pulled from the saved beta coefficient distributions based on time of the year (i.e. the model for predicting sub-daily values in January is different than that for June). Prediction uncertainty is propagated as values are predicted through time.

Important Notes:

* Be sure to set the argument *direction.filter* to be the same direction in which the models were fit in Step 5.
* The variable *ens.hr* denotes the number of sub-daily ensembles that are created for each daily ensemble processed. Currently, Christy has set the variable to be 2 sub-daily created per daily.
* The variable *yrs.plot* is a vector containing the years in which the script will pause predictions and plot the predictions up until that point. These plots can be used to verify that the predictions are going well.

---------------------------------------------------------------------

**Step 7: Assessment of sub-daily ensembles**

---------------------------------------------------------------------

Scripts:

* check\_sub-daily\_met.R
* Reject\_Ens\_mems\_sub-daily.R

For this step, you must:

* Have access to PEcAn functions

The first script generates figures of the corrected data so that you can visually check the mean and variances of the generated sub-daily ensembles (temporally downscaled values) and verify that they are more or less okay. It is important to verify that the diurnal patterns and transitions between days are present in the predicted values. Additionally, the shortwave flux should go to zero at night.

The second script rejects ensemble members which contain highly unlikely or impossible values by “exiling” them to a special folder where they will no longer be considered in other steps.

---------------------------------------------------------------------

**Step 8: Reaggregate to daily MET values**

---------------------------------------------------------------------

Scripts:

* aggregate\_met.R
* aggregate\_met\_day.R

\*\* As of June 4, 2019, these two R scripts are the same.

For this step, you must:

* Have access to the aggregate\_met.R script and its included function

This script takes the accepted predicted sub-daily ensembles from Step 6 and aggregates the values back to daily and monthly values. These values will not necessarily be the same as the original daily ensembles due to the randomness in the predictions.

**\*\* Once this step is complete, you have sub-daily, daily, and monthly MET ensembles for your site. The next steps are not necessary for you to complete. If your goal was to get bias-corrected, downscaled MET data for your site.**

---------------------------------------------------------------------

**Step 9: Calculation of PDSI from ensembles**

---------------------------------------------------------------------

Scripts:

* calculate\_PDSI.R

For this step, you must:

* Have access to 6 scripts (and functions), which include:
  + calc\_pdsi.R
  + calc.awc.R
  + pdsi1.R
  + pdsix.R
  + PE.thornthwaite.R
  + soilmoi1.R

This script calculates Palmer Drought Severity Index (PDSI) for the site in question using the aggregated monthly MET ensembles from Step 8. This calculation allows for comparison between sites and across time.

Important Notes:

* This script relies on “bad things” for calculating PDSI. It uses ED’s calculation of PDSI based on the MET data. If empirical values for available water capacity is available for the site, the prediction can be better. Otherwise, the calculation is based only on PalEON drivers.

---------------------------------------------------------------------

**Step 10: Compression of sub-daily MET ensembles**

---------------------------------------------------------------------

Scripts:

* compress\_members.R

This script compresses sub-daily ensembles into zip tar files to make more manageable for exporting, downloading, moving, etc.

---------------------------------------------------------------------

**Step 11: Randomization of ensemble order**

---------------------------------------------------------------------

Scripts:

* Randomized\_MetOrder.R

This script randomizes the order of the ensemble members.