**Running Met Downscaling Workflow with CRC**

**I. Setting up**

**Requesting a CRC account**

Prior to submitting any jobs through the CRC, one must have an activated CRC account. To request an account, follow the directions in this link: [New User Account](https://crc.nd.edu/index.php/services/forms/user-account-request-form). You will not be able to activate the account for 1 hour after you receive your confirmation e-mail for your account. After an hour, follow the directions in the e-mail to begin using your account.

The CRC also requires that you register for and attend a 1.5-hour “New-user” training session with the center. You can register for a training session at this link: [CRC Events Calendar](https://crc.nd.edu/index.php/news-events/calendar). There are roughly two sessions held each week, but you do not need to attend a session prior to doing anything with your account.

**Logging on to the CRC machines**

In order to employ the CRC machines, you must remotely connect to one of the center’s three front-end machines: crcfe01, crcfe02, and crcfeIB01. You can do this through the command line.

*ssh -X <username>@crcfe01.crc.nd.edu*

This command allows you to remotely access the CRC front-end machine of your choice. In this case, you have accessed the crcfe01 machine. You will be prompted to enter your CRC password before receiving access, which you set up when you activated your account.

**Setting up CRC working directory**

Once have successfully logged onto your user account on the CRC machine, you need to set up your working directory for running the met data workflow by creating the correct directories and adding the necessary scripts and data files to your account. The organization of the main folder *met-crc-workflow* should have the same structure found on the following page or it will not correctly run. It looks kind of complicated, but the workflow actually creates all of the non-bolded directories on its own. You just need to follow these steps to set the rest up.

1. Clone the *met-crc-workflow* (which can be found on GitHub) into the main directory on your CRC account with the following command. This repository contains all of the functions, documentation, scripts, and some of the base data that you will need to run the workflow.

*git clone https://github.com/marissakivi/met-crc-workflow.git*

1. Create the Rlibs folder where you will store any R packages needed for the workflow that are not currently installed on the machine. You are not able to install packages on the machine, so you’ll have to install and call the packages from a folder on your account. These processes are written into the script already, but you do have to make the directory in your main directory as follows.

mkdir Rlibs

1. Create the *ensembles* folder with the following commands.

*cd met-crc-workflow*

*mkdir ensembles*

Once you’ve completed all of these steps, you are ready to start submitting jobs for the met workflow.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | jobs | jobs scripts for each step | | | | | | | |
| doc | documentation, examples, etc. | | | | | | | |
| scripts | edited & condensed scripts for the different steps of the workflow | | | | | | | |
| functions | all PEcAn and Christy scripts that are necessary for the workflow | | | | | | | |
| data | paleon\_sites | HARVARD | bcc-csm1-1 | historical | annual files | | | |
| p1000 | annual files | | | |
| CCSM4 | historical | annual files | | | |
| p1000 | annual files | | | |
| CRUNCEP | annual files | | | | |
| MIROC-ESM | historical | annual files | | | |
| p1000 | annual files | | | |
| MPI-ESM-P | historical | annual files | | | |
| p1000 | annual files | | | |
| NLDAS | annual files | | | | |
| NLDAS\_day (Step 1) | annual files | | | | |
| other sites | same as above | | | | | |
| ensembles | HARVARD.v1 | day | QAQC figure folders | | | | | |
| saved figures + csv files | | | | | |
| ensembles (Step 2) | bcc-csm1-1 | listed ensembles | | | nc files |
| CCSM4 | listed ensembles | | | nc files |
| MIROC-ESM | listed ensembles | | | nc files |
| MPI-ESM-P | listed ensembles | | | nc files |
| rejected | rejected ensembles | nc files | | | |
| 1hr | mods.tdm | model folder for each variable | | | | |
| ensembles | bcc-csm1-1 | hourly ensembles | | annual files | |
| CCSM4 | hourly ensembles | | annual files | |
| MIROC-ESM | hourly ensembles | | annual files | |
| MPI-ESM-P | hourly ensembles | | annual files | |
| aggregated | month | PDSI, temp, and precip CSV files  (Step 5) | | | | |
| linkages | | linkages ensemble folders with data  (Step 6) | | | | |
| weights  (Step 8) | | Rdata file, diagnostic plots | | |

**II. Submitting a job**

Job submissions on the CRC’s “back-end” machines for the met workflow are pretty straightforward. The process is given by the following steps:

1. Make sure your “working directory” is set up correctly, so the machine can find the correct data and save output to the correct locations.
2. Change variable values in the step’s R script to reflect information for your directory and the new site. All of the variables that will need adjusting should be indicated in the first section of code for the step. You can edit the job script in the *met* directory from the command line with the following command.

*vi ~/met-crc-workflow/scripts/<script.name>*

This command will open up the file in a makeshift word processor in the same window. To edit the text, press “i” and use the arrow keys to move around the file. Once you have made your changes, press the esc key, type “:wq,” and press the enter key.

1. Create a job script using the following command when you are in your home user directory. This script will tell the machine what you want it to do. Base job scripts for each workflow step have been written and included in the folder met-crc-workflow/jobs, but this information is helpful in the case that you would like to change the script or write a new one altogether.

*cd met-crc-workflow/jobs*

*vi step-1.sh*

This *vi* command will open a word processor where you can type into the newly-created file now in your directory when you use a file name that does not exist already. Press “i” in order to be able to add text. Once done, close the file as you did in step two and make your job script executable by typing the following command:

*chmod +x <file.name>*

For more information on writing a proper CRC job script, see the given job script template (“template\_job.sh”) in the *doc* folder or visit the CRC’s Wiki page.

1. Once your job script and the met script are ready to go, you just need to submit the job on the command line as follows. RUN THIS COMMAND WHEN YOU ARE IN THE HOME DIRECTORY (i.e. type ‘cd’ first). The output file will contain all of the R output from running the script and is useful for finding errors if the job was not successful; it will be saved in your user home directory.

*qsub met-crc-workflow/jobs/<job script file name>*

Once your job has been submitted, you can check the status of your job and all other jobs you have running with the following command.

*qstat -u <username>*

1. Once you receive the email saying your job has completed, you can proceed to the next step of the workflow. The completed job creates two fils in your home directory: the output file and the job file (<job name>.<job number>). You can open these files using the vicommand. The output file will contain the script output from the R module, and the job file will contain the script output from the job script. There is more information about troubleshooting steps using these files in the next part of this document.

Note: Some of these scripts take from several hours to days to run on the CRC machines. If they are completed very quickly, it is likely there was an error that didn’t allow the script to fully run. Check the outputs!

1. If you realize that a job was incorrectly submitted or you would like to cancel a job for any reason, you can delete a job using the following command on your CRC user account. The job name can be found using the ‘qstat’ command above.

qdel -j <job.name>

**Storage Space**

Each CRC user account is automatically given 100 GB of storage space on the server. Since each completed site has between 10-150 MB of data, we will not be able to store the data for all of the sites on a single account and also use the account for processing more met data. Thus, completed met data should be moved onto a different machine (local or remote). One can also request more space on the CRC servers if desired.

A site folder can be moved to a local machine with the following command from the local machine’s command line. You will be prompted to enter your CRC password before the copy can be made.

scp -r *<username>@crcfe01.crc.nd.edu*:~/met-crc-workflow/ensembles/site.name/linkages path/on/local/machine

Once a site’s data has been safely copied to another location, you can free up the storage space with the following command when on your CRC account’s home directory. The ‘-r’ tag allows you to delete a folder. WARNING: this action cannot be undone so be careful!

rm -r *<username>@crcfe01.crc.nd.edu*:~/met-crc-workflow/ensembles/site.name

**III. Workflow Details**

**Workflow**

The current version of the met workflow is composed of 8 steps, which requires only that raw grid files are saved on your CRC account and that the site coordinates are entered as inputs. The entire workflow results in LINKAGES-specific met ensemble folders and ensemble weights saved in your CRC directory. With any new site, all of these steps will need to be run, some more than once. The entire process of extracting, debiasing, and downscaling met data for a new site requires a lot of time, memory usage, and memory storage, even when using the CRC machines. Therefore, it is important, for the sake of your time, to follow a certain process when generating met ensembles.

Most of the scripts in this workflow contain modified versions of Christy’s met workflow, which is available on [GitHub](https://github.com/PalEON-Project/modeling_met_ensemble). The original workflow is composed of 11 steps, but to simplify the number of job submissions on the CRC, some of the steps have been combined and tested within the CRC environment. For more information on the original workflow and its scripts, see the *met\_documentation* Word Document in the *doc* folder or read Christy’s documentation on GitHub. Otherwise, basic details for the modified workflow are below.

The modified workflow is depicted in the following figure. The deletion of intermediate files is shown in red italics; deletion steps are important to ensure that there is enough space to complete each step.

PRISM

Repeat 2-3 times

Weighting

Extraction

Debias

Repeat 2-3 times

Aggregation to Monthly

PDSI

Downscale

LINKAGES

**Steps**

*Step 1: Extraction of point data from raw grid files*

This script extracts the site-specific met data from all of the large MET grid files for the different necessary met data sources (i.e. NLDAS, four GCMs, and CRUNCEP) required for the bias correction. It also aggregates the NLDAS data to be at a daily resolution so that it can be used in the bias-correction script.

*Step 2: Bias correction for site data*

-- Runtime: 17 hours for long run

This script takes the site-specific met data extracted in Step 1 and combines the data from all of the different sources to generate a smooth, daily met product for the given time period. Since each of the data sources is at a different temporal scale as well, different data products contribute to different years of the final product, as shown in the following table:

|  |  |  |
| --- | --- | --- |
| **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 |

This step also removes any ensemble that contains highly unlikely values and produces a few different figures that one must check to ensure the data products are good.

Each time the debias script is run, by default, it generates 10 daily ensembles for each of the four GCMs, thus, creating 40 ensemble members in total. Ideally, we would like to have about 200 ensembles at the end of the workflow. Therefore, before moving on to the downscaling step (which will generate 2 sub-daily ensembles for each daily ensemble), we should generate about 100 ensemble members by running the bias correction step three times.

**IMPORTANT:** When running the script for the second and third time, you must change the variable *ens* in the first part of the Step 2 script to be 20-30 and, then, 30-40, so that the script doesn’t write over the other ensembles. You should NOT run Step 2 and Step 3 at the same time for a site, as Step 3 could start processing an incomplete ensemble and end in an error.

*Step 3: Downscaling daily product to sub-daily product*

-- Runtime:

This step is compose of three different steps. The steps are necessary so that we can run each of the GCMs in parallel and, thereby, speed up the prediction process. The first step generates statistical models to predict sub-daily climate variables from daily values. The second step (which has four different scripts, one per GCM) uses these models to predict sub-daily values for all the years present in Step 2’s daily product. Each of the daily ensembles from Step 2 are used to generate 2 sub-daily ensembles. The last part of the step removes any and all ensembles that contain highly unlikely values and produces a few different figures that one must check to ensure the data products are good.

Like the second step, this second part of this step will also have to be run two or three times, depending on how many ensembles you have created. The script will process 10 ensembles from each GCM each time it is run by default. You do not need to adjust the ensemble members that it processes as you did in Step 2.

\*\* This step has been altered compared to Christy’s workflow in that it only predicts sub-daily met variables which are needed for conversion of LINKAGES met, as well as site PDSI calculations. This change required that all of the functions used in this step be altered as well. However, any change that I made is marked and commented out in the script. \*\*

*Step 4: Aggregation to monthly data products*

-- Runtime: 5 hours, 55 minutes for long run

1 hour, 10 minutes for short run

This script takes the sub-daily met products created in Step 3 and determines the monthly values for each of the climate variables. This step also calculates the *daylength* variable, which is used to calculate PDSI for the site in the next step. If you are running this script with data products downloaded from Cyverse, be sure to unzip the folders before running the script.

*Step 5: Calculation of PDSI*

-- Runtime: 1 hour, 10 minutes for long run

3 minutes for short run

This script takes the monthly precipitation flux and air temperature, as well as the daylength variable, and calculates the PDSI of the site based on an empirical formula created by . . . For the calculation, it requires information on the site’s soil conditions, which are found using environmental drivers from the ED model saved on the CRC account. At the end of the script, three important CSV files are saved that will be needed later. These files contain the monthly precipitation (inches), mean air temperature (F), and PDSI for the site.

*Step 6: Conversion to LINKAGES met formatting*

-- Runtime: < 5 minutes

This script takes the precipitation and temperature CSV files created in Step 5 and converts and reformats the data to be appropriate for use with the LINKAGES model. Precipitation values are in units of centimeters/month and temperature is given in degrees Celsius. The values are saved as two matrix variables (precip.mat and temp.mat) in a *climate.Rdata* file in separate folders titled as the different sub-daily ensemble members from which they came.

*Step 7: Extraction of PRISM data for weighting*

-- Runtime:

This script does not require any preliminary steps. It only requires the name and coordinates for your site. It, then, extracts the mean temperature values from 1895 to 2017 for the 9 PRISM grid cells that fall within the NLDAS grid cell that contains the inputted site. The matrix containing all of these values is saved in the appropriate folder to be used in the weighting script. This code could be used to also extract other variables, such as minimum temperature, maximum temperature, and precipitation, but it currently only extracts mean temperature as that is all that is necessary for the weighting step.

*Step 8: Weighting of ensemble members using filtering*

-- Runtime: 1 hour 10 minutes for long run

30 minutes for short run

This script is a modified version of John Tipton’s filtering script, which generates filtered weight distributions for the generated met ensembles at the annual scale. It compares the values to reconstructed climate data from a variety of sources for both air temperature and PDSI. The weights can be computed for any year (depending on the site) between 850 and 2006. Both the filtered weight distributions and the mean weights are saved with this script; John recommends considering the entire distribution of weights for each ensemble, as he believes the variability of the weights is important.

**IV. Troubleshooting**

Hopefully, with all of the test runs I did with this workflow, you should not have any errors when running this workflow step-by-step. However, if issues do arise and your met products look a little wonky or are not created at all, here is how you can take a look at what is going on inside the CRC machines.

**Finding errors**

The easiest way to check for and troubleshoot errors in the workflow scripts is by reading the CRC’s R output, which is created for each and every job submission. You can read the output file on the command line using the ‘vi’ command that is described above. However, I prefer to use the ‘scp’ command to copy the file onto my local machine so I can use a more user-friendly text editor. This way I am able to use CTRL-F to find the word “error” and more quickly figure out what went wrong. The output file is placed in whichever directory you were in when you submitted the job.

If the output file is blank or does not contain an error, check the job output file which is in the same directory as the output file. It contains the job name and the job number in its name. This reads the output from running the job script. The most common errors in the job output files are missing files and incorrect script path given.

**Fixing errors**

If you find an error in the step scripts, there are two different ways you can fix it. Like the output file, you can (a) copy in onto your local machine, fix it in RStudio, remove the old script on the CRC, and copy it back onto the remote server or (b) edit the issue on the remote server directly using the “vi” text editor. Either one of these methods will work.

**Moving files between local machine and CRC server**

You can copy files or directories to your CRC account by using the following command with the command line. You will be prompted for your CRC password before the copy will begin.

*scp /path/to/file/on/local/machine <username>@crcfe01.crc.nd.edu*:/destination/on/crc

*If you have any questions at all, do not hesitate to contact me at* [*marissakivi3@gmail.com*](mailto:marissakivi3@gmail.com)*.*

**TO DO:**

* **what to check in figures? talk to Christy maybe**
* **determine how much space is required for all foundational data files (grids, scripts, etc.)**