NorESM model output and post-processing

Yanchun He
Nansen Environment and Remote Sensing Center
14-16, January, 2026

The Research Council of Norway

INES

Where to find and store the model output?

1. **On the Sigma2 HPC (Betzy):**
   NorESM will constantly write time-slice of model state during the runtime and archive the data after the simulation is completed.

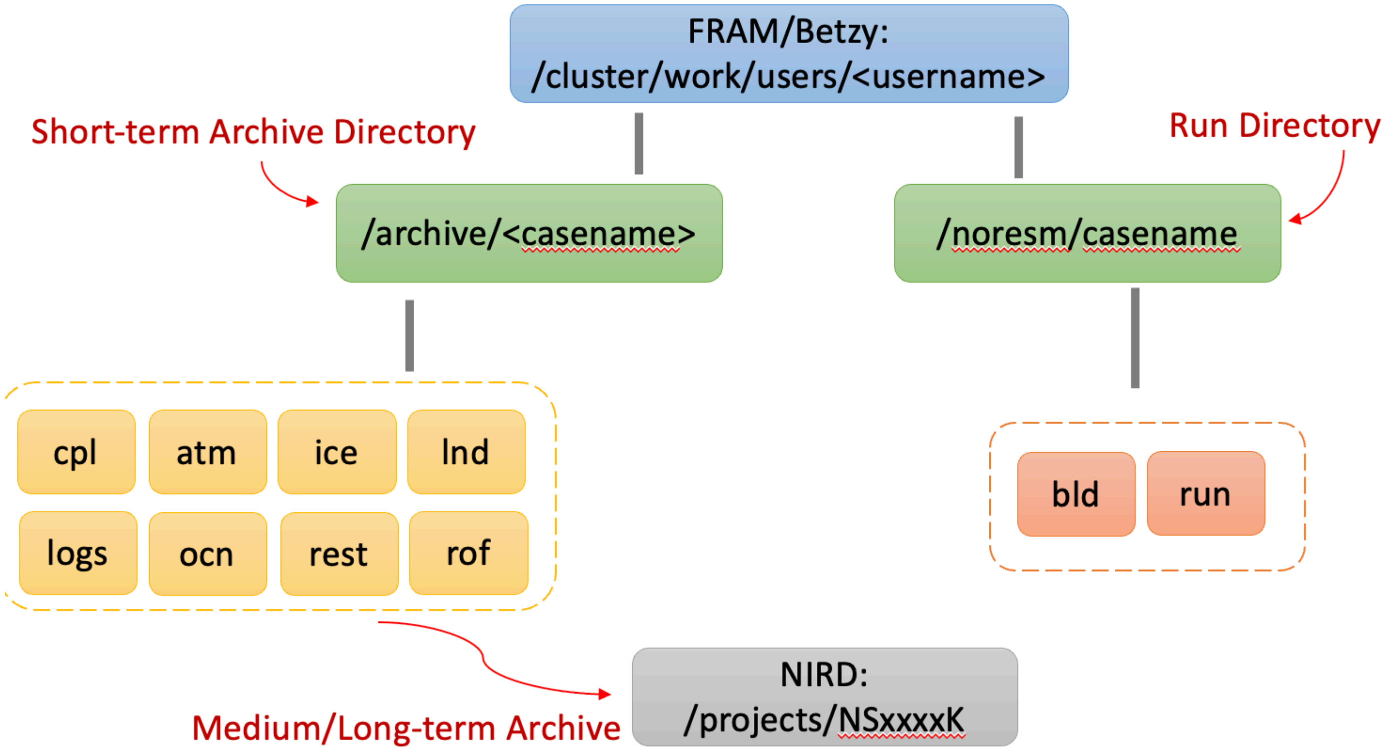
- Model output during runtime (the `run` folder):
  `/cluster/work/users/<username>/noresm/cases/$CASE_NAME`
  (including model build and executables, input and configurations, model output and log files, etc)

- Short-term archive after model finishes successfully (the `archive` folder):
  `/cluster/work/users/<username>/archive/cases/$CASE_NAME`
  (simulated physical/biogeochemical fileds from all active model output)

2. **On Sigma2 storage platform (NIRD):**
   The finished simulation should be transferred to Sigma2 storage facility, NIRD for long-term data archive and analysis.

- Medium/Long-term archive.
  (NIRD project areas, e.g., `/projects/NS9560K` for INES project.)

Structure of model output archive



Model output are usually transferred from the HPC (Betzy) to the storage (NIRD) with `rsync` or `scp` (refer to Sigma2 documentation File Transfer).

```
$ cd /cluster/work/users/$USER/archive
$ rsync -vazu <your_case_name> $USER@login.nird.sigma2.no:/projects/NSxxxxK/<folder_to_case>/<your_case_name>/
```

File naming convention of model output

```
$ ls
atm/  case/  cpl/  esp/  ice/  lnd/  logs/  ocn/  rest/  rof/
```

- **History files:** `<component>/hist`, e.g, `atm/hist`,`ocn/hist`.
- **Restart files:** `rest/hist`.
- **Log files**: `<component>.log.$pid.ddmmyy-hhmmss.gz`

```
$ tree -L 2
.
├── archive.log.200921-151923
├── atm
│   └── hist
├── cpl
│   └── hist
├── esp
│   └── hist
├── ice
│   └── hist
├── lnd
│   └── hist
├── logs
│   ├── atm.log.781577.200921-144102.gz
│   ├── cesm.log.781577.200921-144102.gz
│   ├── cpl.log.781577.200921-144102.gz
│   ├── ice.log.781577.200921-144102.gz
│   ├── lnd.log.781577.200921-144102.gz
│   ├── ocn.log.781577.200921-144102.gz
│   └── rof.log.781577.200921-144102.gz
├── ocn
│   └── hist
├── rest
│   └── 0001-02-01-00000
└── rof
    └── hist
```

Example history file names:

```
<compset name>_<resolution sname>_<opt_desc_string>_<component>.<frequency>_<date>.nc
```

- N1850frc2_f19_tn14_test.blom.hm.0001-01.nc
- N1850frc2_f19_tn14_test.cam.h0.0001-01.nc

By default, `h0,hm` denotes that the time sampling frequency is monthly.
Other frequencies are saved under the h1, h2, etc.
Different time sampling frequencies have distinct tags in the file names.

A full list of the tags:

```
    - blom.hy    = blom yearly
    - blom.hbgcy = blom/bgc yearly
    - blom.hm    = blom monthly
    - blom.hbgcm = blom/bgc monthly
    - blom.hd    = blom daily
    - blom.hbgcd = blom/bgc daily
    - cice.h     = ice monthly
    - cice.h1    = ice daily
    - cam.h0     = cam monthly
    - cam.h1     = cam daily
    - cam.h2     = cam 6-hourly average
    - cam.h3     = cam 6-hourly instant
    - cam.h4     = cam 3-hourly average
    - cam.h5     = cam 3-hourly instant
    - clm2.h4    = clm yearly
    - clm2.h0    = clm monthly
    - clm2.h1    = clm daily
    - clm2.h2    = clm 3-hourly average
    - clm2.h3    = clm 3-hourly instant
```

Metadata of model output

Taking a peek on the metadata of the atmosphere and ocean components.

`grid:f19_tn14, CAM 2°x2°`

```
netcdf NHISTfrc2_f19_tn14_1950.cam.h0.2014-12 {
dimensions:
        lat = 96 ;
        lon = 144 ;
        zlon = 1 ;
        nbnd = 2 ;
        lev = 32 ;
        ilev = 33 ;
        time = UNLIMITED ; // (1 currently)
        chars = 8 ;
```

```
variables:
        double lat(lat) ;
                lat:_FillValue = -900. ;
                lat:long_name = "latitude" ;
                lat:units = "degrees_north" ;
        double lon(lon) ;
                lon:_FillValue = -900. ;
                lon:long_name = "longitude" ;
                lon:units = "degrees_east" ;
        double gw(lat) ;
                gw:_FillValue = -900. ;
                gw:long_name = "latitude weights" ;
...
        double lev(lev) ;
                lev:long_name = "hybrid level at midpoints (1000*(A+B))" ;
                lev:units = "hPa" ;
                lev:positive = "down" ;
                lev:standard_name = "atmosphere_hybrid_sigma_pressure_coordinate" ;
                lev:formula_terms = "a: hyam b: hybm p0: P0 ps: PS" ;
        double hyam(lev) ;
                hyam:long_name = "hybrid A coefficient at layer midpoints" ;
        double hybm(lev) ;
                hybm:long_name = "hybrid B coefficient at layer midpoints" ;
        double P0 ;
                P0:long_name = "reference pressure" ;
                P0:units = "Pa" ;
        double ilev(ilev) ;
                ilev:long_name = "hybrid level at interfaces (1000*(A+B))" ;
                ilev:units = "hPa" ;
                ilev:positive = "down" ;
                ilev:standard_name = "atmosphere_hybrid_sigma_pressure_coordinate" ;
                ilev:formula_terms = "a: hyai b: hybi p0: P0 ps: PS" ;
        double hyai(ilev) ;
                hyai:long_name = "hybrid A coefficient at layer interfaces" ;
        double hybi(ilev) ;
                hybi:long_name = "hybrid B coefficient at layer interfaces" ;
```

```
...
        TREFHT:cell_methods = "time: mean" ;
```

`grid:f09_tn14, CAM 1°x1°`

```
netcdf NHISTfrc2_f09_tn14_20191025.cam.h0.2014-12 {
dimensions:
        lat = 192 ;
        lon = 288 ;
        zlon = 1 ;
        nbnd = 2 ;
        time = UNLIMITED ; // (1 currently)
        chars = 8 ;
        lev = 32 ;
        ilev = 33 ;
```

```
variables:
        double lat(lat) ;
                lat:_FillValue = -900. ;
                lat:long_name = "latitude" ;
                lat:units = "degrees_north" ;
        double lon(lon) ;
                lon:_FillValue = -900. ;
                lon:long_name = "longitude" ;
                lon:units = "degrees_east" ;
        double gw(lat) ;
                gw:_FillValue = -900. ;
                gw:long_name = "latitude weights" ;
...
        double lev(lev) ;
                lev:long_name = "hybrid level at midpoints (1000*(A+B))" ;
                lev:units = "hPa" ;
                lev:positive = "down" ;
                lev:standard_name = "atmosphere_hybrid_sigma_pressure_coordinate" ;
                lev:formula_terms = "a: hyam b: hybm p0: P0 ps: PS" ;
        double hyam(lev) ;
                hyam:long_name = "hybrid A coefficient at layer midpoints" ;
        double hybm(lev) ;
                hybm:long_name = "hybrid B coefficient at layer midpoints" ;
        double P0 ;
                P0:long_name = "reference pressure" ;
                P0:units = "Pa" ;
        double ilev(ilev) ;
                ilev:long_name = "hybrid level at interfaces (1000*(A+B))" ;
                ilev:units = "hPa" ;
                ilev:positive = "down" ;
                ilev:standard_name = "atmosphere_hybrid_sigma_pressure_coordinate" ;
                ilev:formula_terms = "a: hyai b: hybi p0: P0 ps: PS" ;
        double hyai(ilev) ;
                hyai:long_name = "hybrid A coefficient at layer interfaces" ;
        double hybi(ilev) ;
                hybi:long_name = "hybrid B coefficient at layer interfaces" ;
        double time(time) ;
                time:long_name = "time" ;
                time:units = "days since 1850-01-01 00:00:00" ;
                time:calendar = "noleap" ;
                time:bounds = "time_bnds" ;
```

`grid:f19_tn14, BLOM nominal 1°x1°`

```
netcdf NHISTfrc2_f19_tn14_1950.blom.hm.2014-12 {
dimensions:
        time = UNLIMITED ; // (1 currently)
        sigma = 53 ;
        depth = 70 ;
        bounds = 2 ;
        lat = 171 ;
        region = 4 ;
        slenmax = 50 ;
        section = 17 ;
        y = 385 ;
        x = 360 ;
```

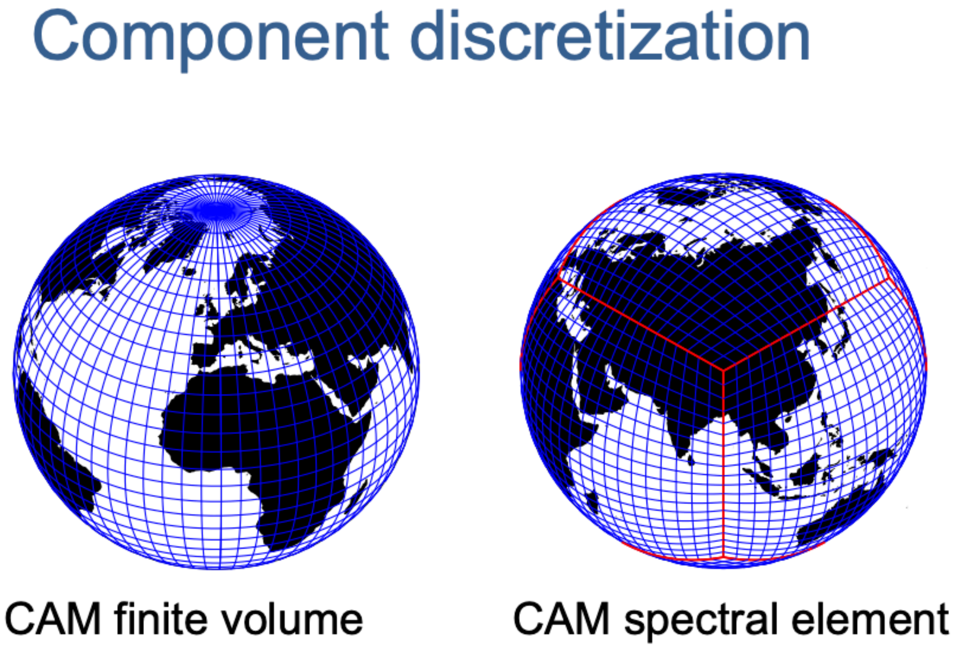
```
variables:
        double time(time) ;
                time:long_name = "time" ;
                time:units = "days since 1800-01-01 00:00" ;
                time:calendar = "noleap" ;
        double sigma(sigma) ;
                sigma:_FillValue = 9.96920996838687e+36 ;
                sigma:long_name = "Potential density" ;
                sigma:standard_name = "sea_water_sigma_theta" ;
                sigma:units = "kg m-3" ;
                sigma:positive = "down" ;
        double depth(depth) ;
                depth:_FillValue = 9.96920996838687e+36 ;
                depth:long_name = "z level" ;
                depth:units = "m" ;
                depth:positive = "down" ;
                depth:bounds = "depth_bnds" ;
        double depth_bnds(depth, bounds) ;
                depth_bnds:_FillValue = 9.96920996838687e+36 ;
        double lat(lat) ;
                lat:_FillValue = 9.96920996838687e+36 ;
                lat:long_name = "Latitude" ;
                lat:standard_name = "latitude" ;
                lat:units = "degree_north" ;
        char region(region, slenmax) ;
                region:long_name = "Region name" ;
        char section(section, slenmax) ;
                section:long_name = "Section name" ;
```

NorESM horizontal grid: CAM and CLM

The atmphere (CAM) and land (CLM) components share the same grid.

- NorESM2-LM (CAM) for CMIP6: 2x2 degree
- NorESM2-MM (CAM) for CMIP6: 2x2 degree
- NorESM2.5/3 (CAM-SE) for CMIP7: SE grid

```
CAM: 2x2 deg
netcdf NHISTfrc2_f19_tn14_1950.cam.h0.2014-12 {
dimensions:
        lat = 96 ;
        lon = 144 ;
        zlon = 1 ;
        nbnd = 2 ;
        lev = 32 ;
        ilev = 33 ;
        time = UNLIMITED ; // (1 currently)
        chars = 8 ;
```



CMIP6 (left) and CMIP7 (right) versions of NorESM.

NorESM horizontal grid: BLOM and CICE

The ocean (BLOM) and sea ice (CICE) components share the same grid.

Bipolar (gx1v6) for CMIP5; Tripolar (tnx1v4) for CMIP6.

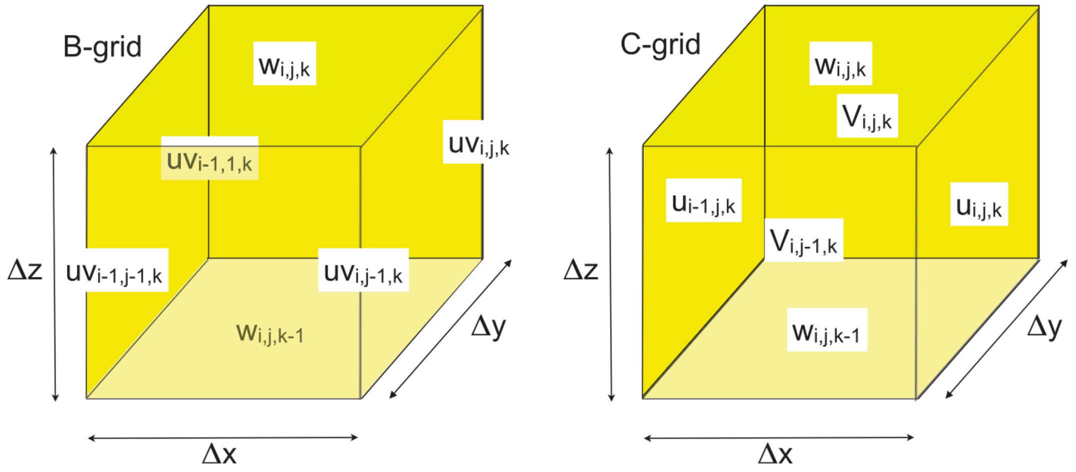
In case folder, Buildconf/blom.input_data_list:

```
grid_file = /cluster/shared/noresm/inputdata/ocn/blom/grid/grid_tnx1v4_20170622.nc
```

or in run folder, noresm/run/ocn_in:

```
GRFILE = '/cluster/shared/noresm/inputdata/ocn/blom/grid/grid_tnx1v4_20170622.nc
```

```
netcdf grid_tnx1v4_20170622 {
dimensions:
        x = 360 ;
        y = 385 ;
        nv = 4 ;
variables:
...
        double plon(y, x) ;
                plon:units = "degrees_east" ;
                plon:long_name = "Longitude at p-points" ;
                plon:corners = "pclon" ;
        double plat(y, x) ;
                plat:units = "degrees_north" ;
                plat:long_name = "Latitude at p-points" ;
                plat:corners = "pclat" ;
        double ulon(y, x) ;
                ulon:units = "degrees_east" ;
...
```



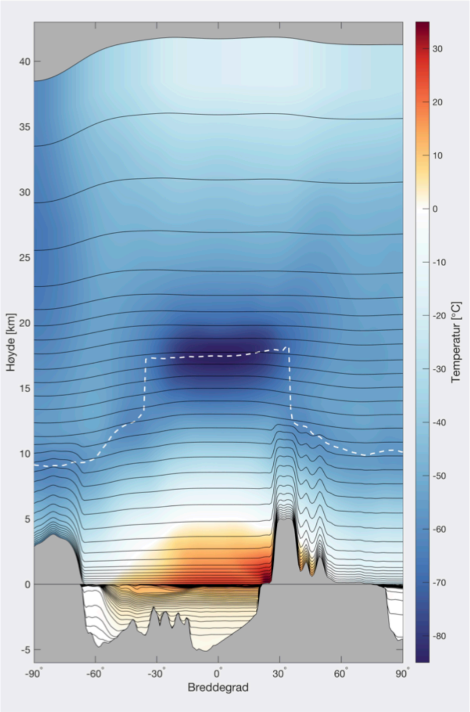
Staggering of the vectors and scalars at different positions of a model cell for B-grid and C-grid.

NorESM vertical coordinates

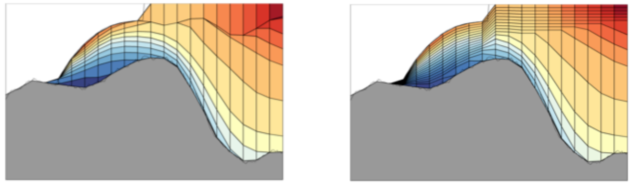
- CAM: terrian-following sigma coordinate
- BLOM: isopycnic (potential density σ_2) coordinated vertical coordinate

```
float temp(time, sigma, y, x) ;
        temp:_FillValue = 9.96921e+36f ;
        temp:units = "degC" ;
        temp:long_name = "Temperature" ;
        temp:standard_name = "Ocean temperature" ;
        temp:coordinates = "plon plat" ;
        temp:cell_measures = "area: parea" ;

float templvl(time, depth, y, x) ;
        templvl:_FillValue = 9.96921e+36f ;
        templvl:units = "degC" ;
        templvl:long_name = "Temperature" ;
        templvl:standard_name = "Ocean temperature" ;
        templvl:coordinates = "plon plat" ;
        templvl:cell_measures = "area: parea" ;
```



April temperature of NorESM2 along 87.5°E



Isopycnic with bulk mixed layer

Hybrid

NorESM time-dimension (time slices)

BLOM

The time coordinate variable in ocean model BLOM history represents the middle of the averaging period for variables that are averages. No `time_bounds` for the `time` axis.

BLOM output

```
$ ncdump -t -v time N1850frc2_f19_tn14_Workshop2020.blom.hm.0001-01.nc |tail -4
data:

 time = "0001-01-17" ;
}
```

NorESM time-dimension (time slices)

CAM

The time coordinate variable in atmospheric model CAM history and timeseries files represents the end of the averaging period for variables that are averages (inherited from CESM). Its `time_bnds` attribute of `time` axis gives over which period the field is averaged.

Example File: `N1850frc2_f19_tn14_Workshop2020.cam.h0.0001-01.nc`
When the time coordinate variable is translated, the time is 00Z Februray 1st 0001, even though the file holds averaged variables for January 0001.

CAM output

```
$ ncdump -v time,time_bnds -t NHISTfrc2_f19_tn14_1950.cam.h0.2014-12.nc
data:

 time = "2015-01-01" ;

 time_bnds =
  "2014-12-01", "2015-01-01" ;
}
}
```

Summary:

- Different model resoltions
- Different components have different model grids
- Different vertical coordinates
- Huge number of output files and variables (> 1000)
- ...
- A way to quickly diagnose my simulation and benchmark with observations?

The NorESM Diagnostic Tool Package

NorESM Diagnostic Package:

> is a NorESM model evaluation tool written with a set of scripts and utilities (bash, NCL, NCO, CDO etc) to provide a general evaluation and quick preview of the model performance with only one command line. The results (tables, figures) are presented with generated web pages.

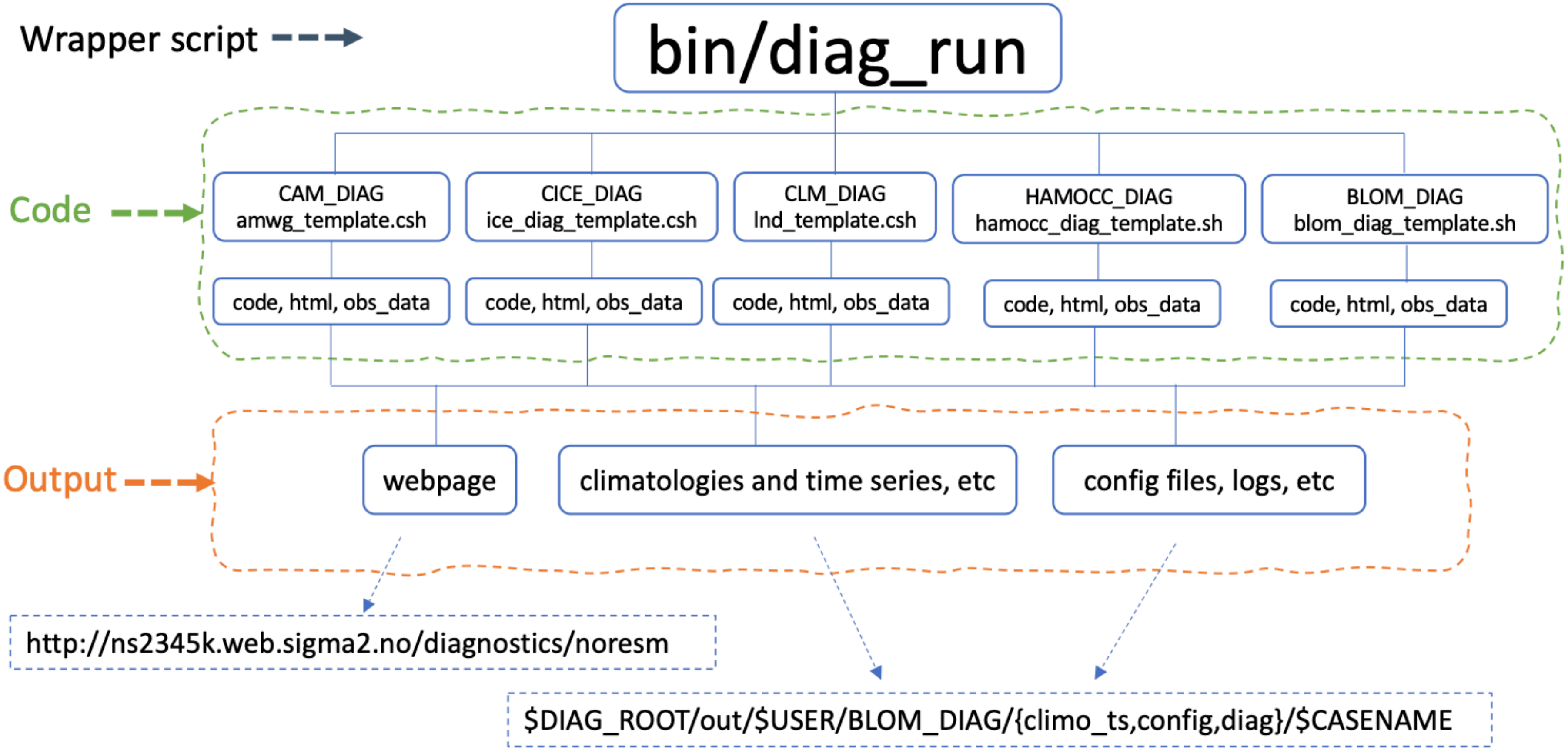
Components of the package:

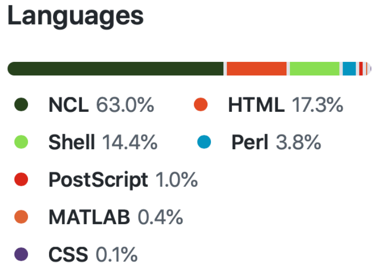
The diagnostic tool package consists atmospheric/land components based on the NCAR package.

- **CAM_DIAG:** (NCAR's AMWG Diagnostics Package)
- **CLM_DIAG:** (CESM Land Model Diagnostics Package)
- **CICE_DIAG:** snow/sea ice volume/area
- **HAMOCC_DIAG:** time series, climaotology, zonal mean, regional mean
- **BLOM_DIAG:** time series, climatologies, zonal mean, fluxes, etc
- **CISM_DIAG:** time series, climatologies, etc

NorESM Diagnostic Package (cont.)

Code structure





NorESM Diagnostic Package (cont.)

It has a one-line command interface, and is simple-to-use.

```
# run this wraper script without parameters shows basic usage
$ diag_run
--------------------------------------------------
Program:
/diagnostics/noresm-diagnostics-src/bin/diag_run
Version: 2.x
--------------------------------------------------
Short description:
A wrapper script for NorESM diagnostic packages.

Basic usage:
# model-obs comparison
$ diag_run -m [model] -c  [test case name] -s  [test case start yr] -e  [test case end yr]

# model-model comparison
$ diag_run -m [model] -c  [test case name] -s  [test case start yr] -e  [test case end yr] \
                      -c2 [cntl case name] -s2 [cntl case start yr] -e2 [cntl case end yr]
...
```

Two types of analysis

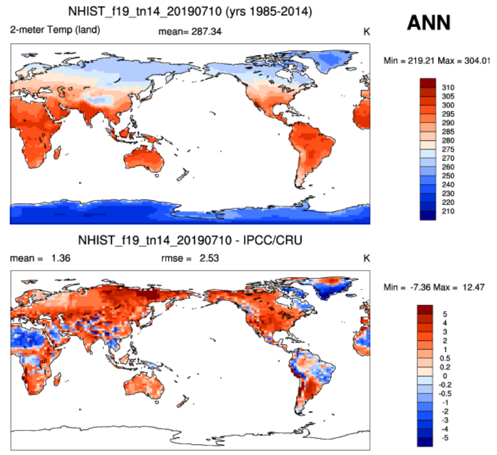
1. Compare model with observations

- sample plots: Historical simulation of ocean compared to observations

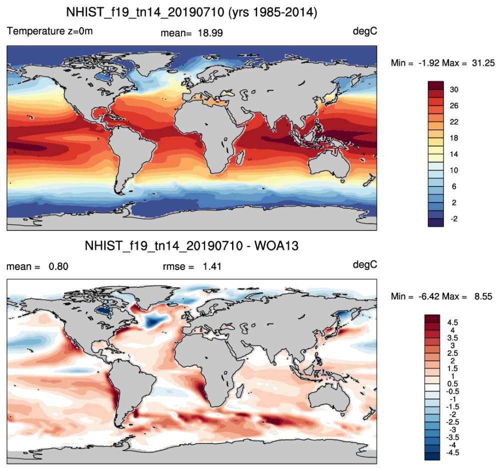
```
$ diag_run --model=cam,cice,blom \
--case=CASENAME \
--start_year=51 \
--end_year=100 \
--input-dir=/PATH/TO/MODEL/FOLDER \
--output-dir=/PATH/TO/OUTPUT/DATA \
--web-dir=/PATH/TO/GENERATED/WEBPAGES \

# or its short version
$ diag_run -m cam,cice,blom \
  -c CASENAME \
  -s 51 -e 100 \
  -i /PATH/TO/MODEL/FOLDER \
  -o /PATH/TO/OUTPUT/DATA \
  -w /PATH/TO/GENERATED/WEBPAGES
```

2m Air temperature



Sea surface temperature



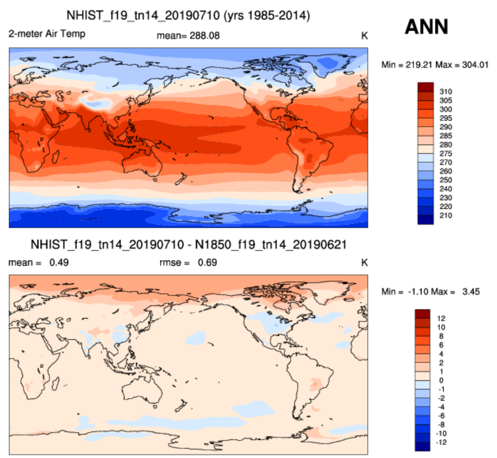
2. Compare model with control (another simulation)

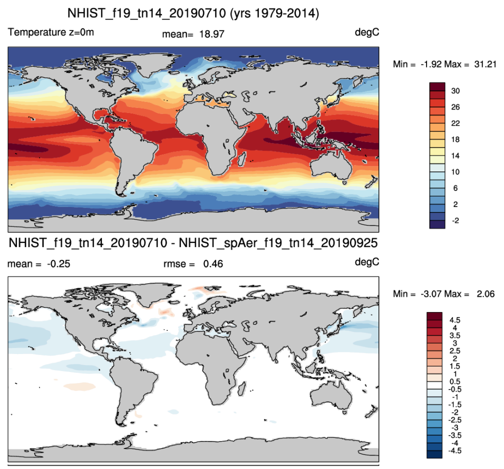
Sample plots: Historical simulation of atmosphere compared to PI control

```
$ diag_run --model=cam,cice,blom \
--case1=CASENAME1 \
--start_year1=51 \
--end_year1=100 \
--input-dir1=/PATH/TO/MODEL/FOLDER1 \
--case2=CASENAME2 \
--start_year2=2 \
--end_year2=50 \
--input-dir2=/PATH/TO/MODEL/FOLDER2 \
--output-dir=/PATH/TO/OUTPUT/DATA \
--web-dir=/PATH/TO/GENERATED/WEBPAGES \

# or its short version
$ diag_run -m cam,cice,blom \
 -c1 CASENAME1 -s1 51 -e1 100 -i1 /PATH/TO/MODEL/FOLDER1 \
 -c2 CASENAME2 -s2 1  -e2 50  -i2 /PATH/TO/MODEL/FOLDER2 \
 -o /PATH/TO/OUTPUT/DATA \
 -w /PATH/TO/GENERATED/WEBPAGES
```

2m Air temperature	Sea surface temperature

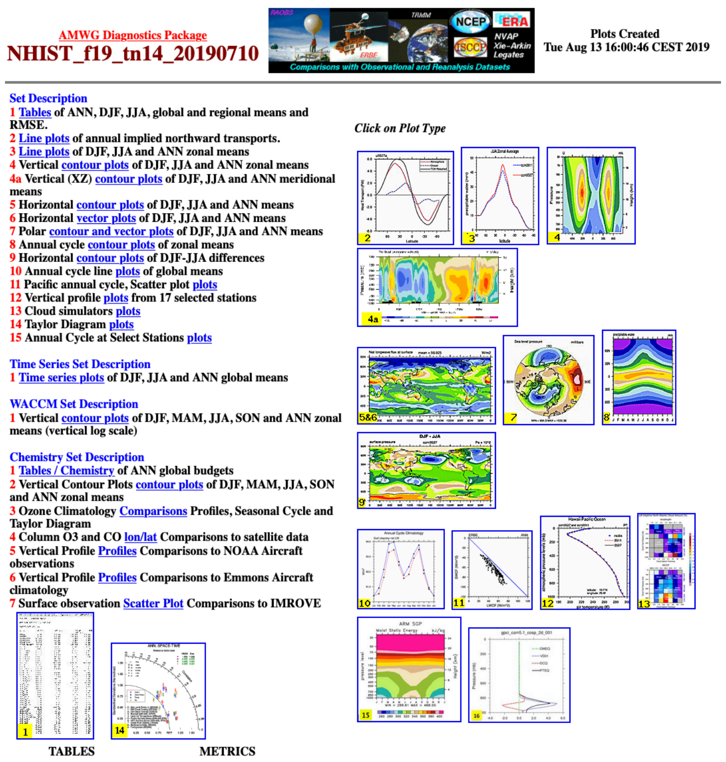




Sets of diagnostics

Atmospheric diagnostics

(example plots)

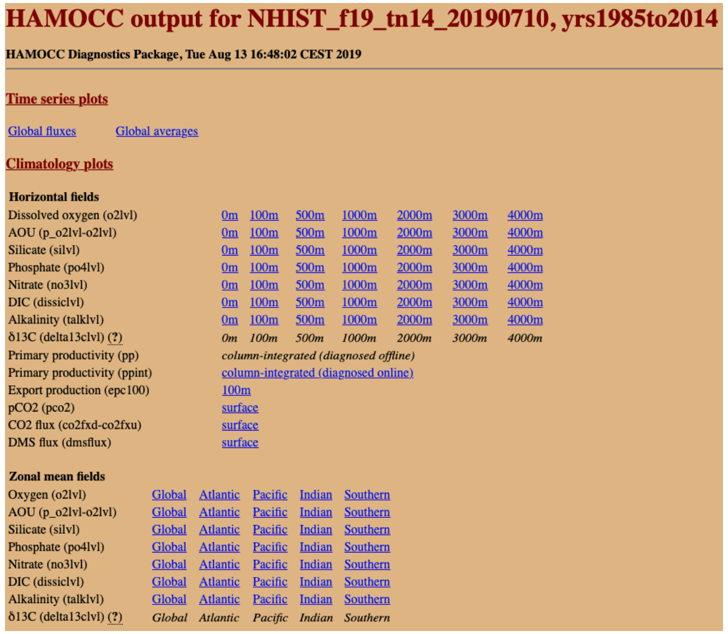


Ocean diagnostics

(Example plots)

Biogeochemistry diagnostics

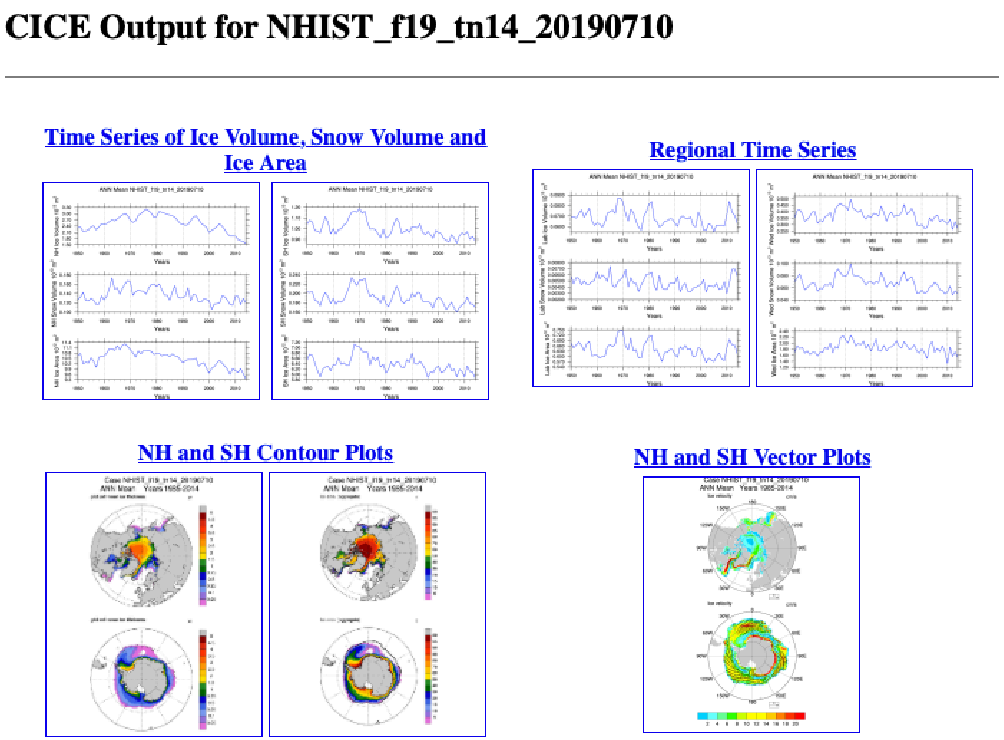
(Example plots)



Sets of diagnostics (cont.)

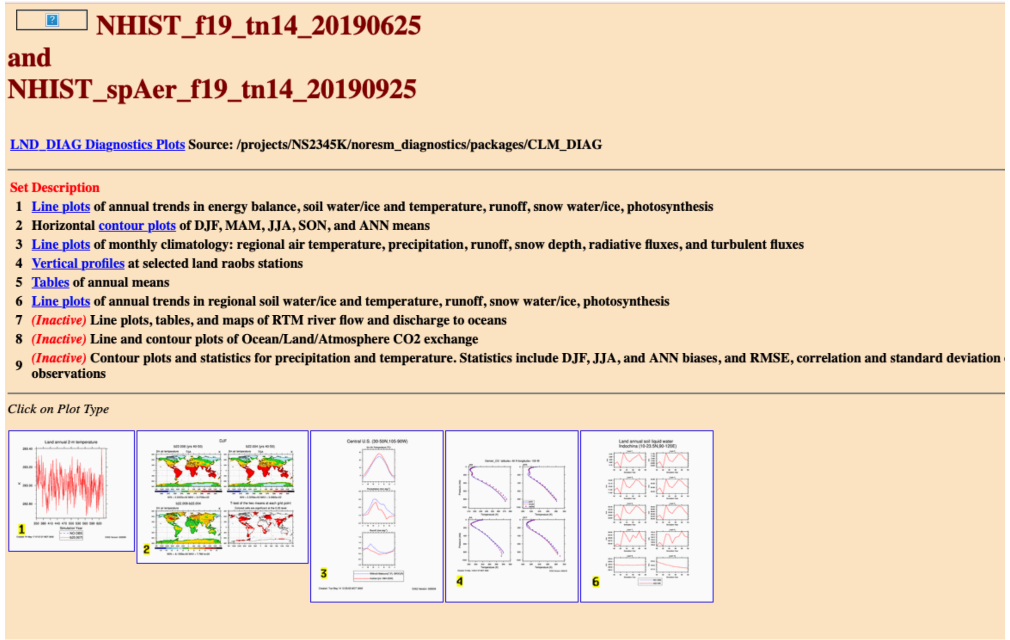
Land diagnostics

(example plots)



Sea ice diagnostics

(Example plots)



Browse plots online, while you enjoy the peace ☕ 🍵

1. http://ns2345k.web.sigma2.no/datapeak/diagnostics/noresm
    - most of CMIP6 diagnostics are stored under `commom/`
    - personal diagnostics are store under `$username/`
2. http://ns9560k.web.sigma2.no/datalake/diagnostics/noresm

Note

The dianostic tool will write the webpage output to NS9560K by default.
If you don't have access to it, specify `-w` option to direct your webpage to a NIRD project area where you have write permission.
You can then make a tarball (`tar -cvzf casenme.tar.gz /path/to/the/weboutput`)
And download to your local computer to view with your browser.

Resources

Where is it?

- Github: https://github.com/NorESMhub/NorESM-Diagnostics
- Normal NIRD login nodes (login.nird.sigma2.no): `/projects/NS16000B/diagnostics/noresm-diagnostics-src`
- "IPCC" node of NIRD (ipcc.nird.sigma2.no): `/diagnostics/noresm-diagnostics-src`
- Betzy (betzy.sigma2.no): `/cluster/shared/noresm/diagnostics/noresm`

Do **NOT** directly modify these installations

Find the full doocumentation:

- https://noresm-docs.readthedocs.io/en/noresm2/diagnostics/diag_run.html
- https://noresmhub.github.io/Noresm-Diagnostics/ (NOT up-to-date)

Exercises

- Task 1: set up the tool
- Task 2: model-obs comparison
- Task 3: model-model comparison
- Task 4: run the tool under `passive mode`

Task 1. Set up the tool

option 1: on NIRD

Use pre-installed tool under NIRD (Recommended)

```
# First, logon NIRD
$ ssh -l <your_username> ipcc.nird.sigma2.no

# Next, add alias in '~/.bashrc' for diag_run
if [ -f /diagnostics/noresm/bin/diag_run ];then
    alias diag_run='/diagnostics/noresm/bin/diag_run'
fi

# Then, source the bash configuration to take effect
$ source ~/.bashrc

# check if diag_run is in your searching path
$ which diag_run
```

(See https://noresm-docs.readthedocs.io/en/latest/output/archive_output.html on how to transfer the NorESM output from Betzy to NIRD.)

Task 1. Set up the tool

Option 2: on Betzy

```
# Fist, logon betzy
$ ssh -l <your_username> betzy.sigma2.no

# Next, add alias in '~/.bashrc' for diag_run
alias diag_run='/cluster/shared/noresm/diagnostics/noresm/bin/diag_run'
alias diag_srun='/cluster/shared/noresm/diagnostics/noresm/bin/diag_srun'

# Then, source the bash configuration to take effect
$ source ~/.bashrc

# check if diag_run is in your searching path
$ which diag_srun
```

`diag_run` and `diag_srun`

- job with `diag_run` will run directly on the login nodes (both on NIRD and Betzy)
- job with `diag_srun` will submit a job to the compute node (on Betzy).

It is important to use `diag_srun` on Betzy to avoid clogging the login nodes!

Get familiar with the `diag_run` (`diag_srun`)

`diag_run`

```
$ diag_run

Short description:
  A wrapper script for NorESM diagnostic packages.

Basic usage:
  # model-obs comparison
  diag_run -m [model] -c [test case name] -s [test case start yr] -e [test case end yr]
  # model-model comparison
  diag_run -m [model] -c [test case name] -s [test case start yr] -e [test case end yr] -c2 [cntl case name] -s2 [cntl case start yr] -e2 [cntl case end yr]
...
```

`diag_srun`

```
$ diag_srun

Short Description:
  A wrapper script for submitting a preprocessing job on Betzy for the NorESM diagnostic packages.

Command-Line Options (in addition to options as shown with diag_run -h):
  --account=<project_account> (OPTIONAL)
      Specify the project account for CPU hours.
      Default: ns9560k

  --time=<HH:MM:SS> (OPTIONAL)
      Specify CPU walltime.
      Default: Automatically calculated based on the length of years and active components.

  --remove-source-files-flag=true|false (OPTIONAL)
      Set whether source files should be removed after transferring the webpage from Betzy (/cluster) to NIRD (/nird).
      Default: false
```

Task 2. Model-obs comparison

Compare model to observation

```
# syntax:
$ diag_run  -m MODEL -c CASENAEME -s START_YEAR -e  END_YEAR -i INPUT -o OUTPUT -w WEBPAGE
```

Example 1: compare NorESM2-LM historical run (years 1985 - 2014) with observations

```
$ diag_run -m blom -c NHIST_f19_tn14_20190710 -s 1985 -e 2014 \
        -i /projects/NS9560K/noresm/cases  \
        -o /projects/NS2345K/diagnostics/noresm/out/$USER \
        -w /projects/NS2345K/www/diagnostics/noresm/$USER
```

If have no access to NS9560K, try:

```
$ diag_run -m blom -c NHISTfrc2_workshop2021 -s 1850 -e 1859 \
        -i /cluster/shared/noresm/NorESMWS2023/archive \
        -o /cluster/work/users/yanchun/diagnostics/out \
        -w /cluster/work/users/yanchun/diagnostics/www
```

Task 2. Model-obs comparison (cont.)

Example 2: compare NorESM2-LM piControl run (years 1735 - 1764, equivalent to 1985 - 2014), only diagnose the ocean component, and omit the `-o` and `-w` options (default to the above settings).

```
$ diag_run -m blom -c N1850_f19_tn14_20190621 -s 1735 -e 1764 \
                        -i /projects/NS9560K/noresm/cases
```

If have no access to NS9560K, try

```
$ diag_run -m blom -c N1850frc2_workshop2021 -s 1 -e 10 \
        -i /cluster/shared/noresm/NorESMWS2023/archive \
        -o /cluster/work/users/$USER/diagnostics/out \
        -w /cluster/work/users/$USER/diagnostics/www
```

Example 3: run on Betzy with `sbatch` job to backend.

```
$ diag_srun -m blom -c <case_name> -s 2000 -e 2010 \
        -i /cluster/work/users/$USER/archive \
        -o /cluster/work/users/$USER/diagnostics/out \
        -w /cluster/work/users/$USER/diagnostics/www \
        --account=nn9560k --time=00:59:00
```

Task 2. Model-obs comparison (cont.)

Browse the output of the diagnostics.

If one store the output of diagnostics (with `-w`) to certain NIRD project area, e.g., `/nird/datalake/NS2345K/www/diagnostics/noresm`, the results are available directly online at: https://ns2345k.web.sigma2.no/datalake/diagnostics/noresm.

If the ouput is under `Betzy`, one can download the generate figures and webpages, or start a temporary python server to browse result as the following:

```
# step 1: connect to Betzy with port forwarding
$ ssh -L 8080:localhost:8080 yanchun@betzy.sigma2.no
# step 2: change to the directory where you wish to be the start of the web address
$ cd /cluster/work/users/$USER/diagnostics/www/
# step 3: start a temporary web server with python
$ python -m http.server 8080 --bind 127.0.0.1
# step 4: open http://127.0.0.1:8080 on your local browser
...
```

Task 3. Model-model comparison

Compare model to model

Syntax:

```
$ diag_run  -m cam -c1 CASENAME1 -s1 START_YEAR1 -e1 END_YEAR1 \
                   -c2 CASENAME2 -s2 START_YEAR2 -e2 END_YEAR2 \
                   -i1 /PATH/TO/INPUT1 -i2 /PATH/TO/INPUT2 \
                   -o /PATH/TO/OUTPUT \
                   -w /PATH/TO/WEBPAGE
```

Example: compare NorESM2-LM historical run, years, 1985 to 2014 to piControl (years, 1735 - 1764)

```
$ diag_run  -m cam -c1 NHIST_f19_tn14_20190710 -s1 1985 -e1 2014 -i1 /projects/NS9560K/noresm/cases \
                   -c2 N1850_f19_tn14_20190621 -s2 1735 -e2 1764 -i2 /projects/NS9560K/noresm/cases
```

To run on Betzy with `sbatch` job to backend.

```
$ diag_srun ....
```

Task 4. Run `diag_run` with the `passive mode`

The wrapper script `diag_run` can be invoked but not really executed, so one can futher tweak the diagnostic scripts. This is called the `passive mode` (with `-p` or `--passive-mode`)

```
# Example
$ diag_run -m cam -c NHIST_f19_tn14_20190710 -s 1985 -e 2014 -p
```

In the standard output, you can find lines like:

```
...
CAM DIAGNOSTICS SUCCESSFULLY CONFIGURED in /cluster/work/users/$USER/diagnostics/out/CAM_DIAG
...
```

Go the that directory and check the shell script there, `amwg_template.csh`

For example, to switch on/off some options:

`/cluster/work/users/$USER/diagnostics/out/CAM_DIAG/amwg_template.csh`

```
set significance = 0   # (0=ON,1=OFF)  Turn on significance test
...
set all_waccm_sets = 0 # (0=ON,1=OFF)  Do all the WACCM sets
set all_chem_sets  = 0 # (0=ON,1=OFF)  Do all the CHEM sets
```

then start the scripit `./amwg_template.csh`.