Part 3: LASSO-O Workflow Software

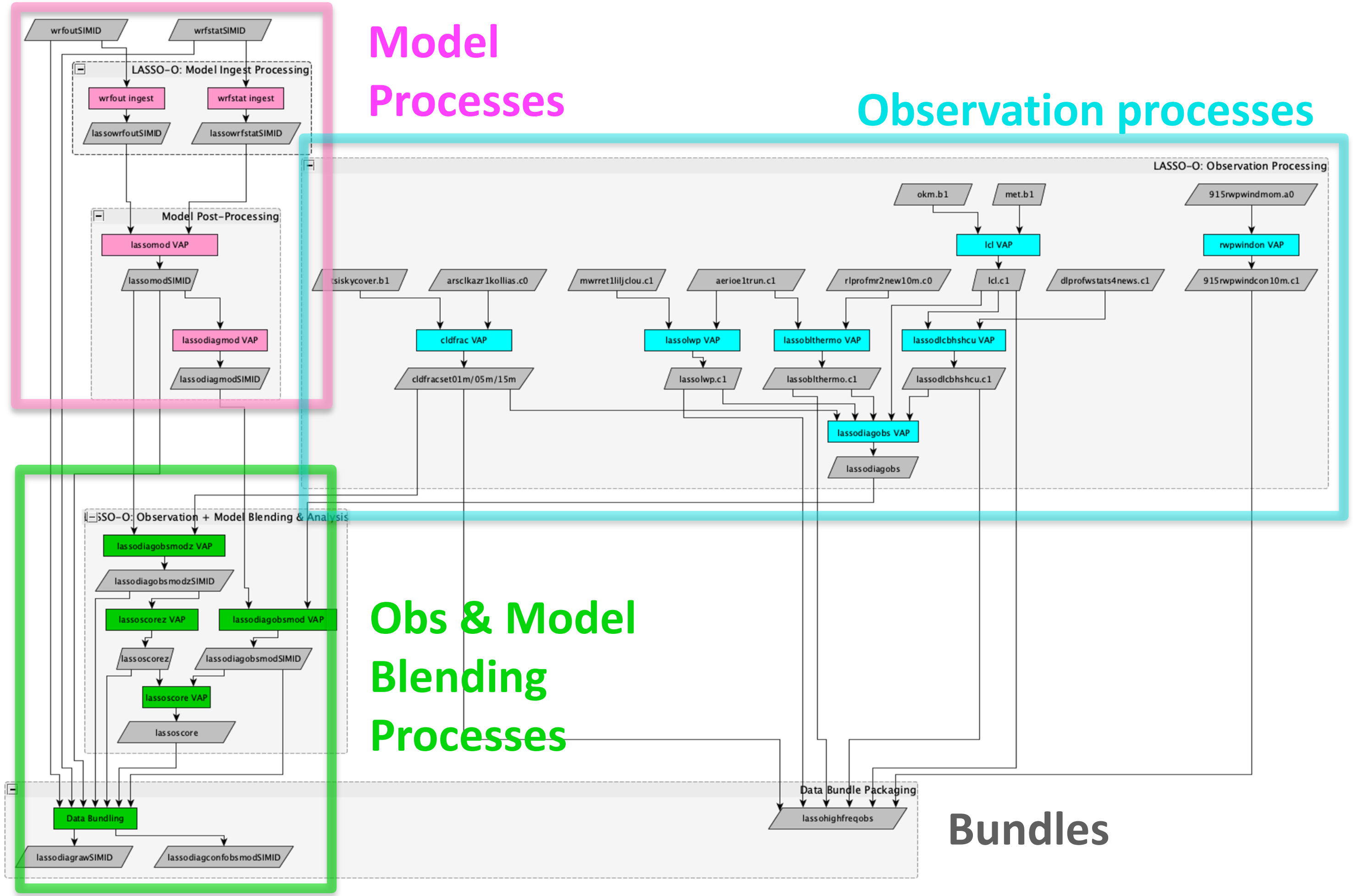
Presenters: Tami Fairless (BNL), Carina Lansing (PNNL)

What you will learn

- Introduction:
  - What is LASSO-O
  - What is a container?
  - The LASSO-O Container
- LASSO-O Container:
  - Workflow
  - Input & Output
- Running LASSO-O in a container:
  - Getting started using LASSO-O in a container
  - Steps to use LASSO-O in a container
- Plotting the Results

What is LASSO-O?

- LASSO-O is the ARM Operationalization of the LASSO workflow
  - from WRF LES output
  - through to the model evaluation components of LASSO bundles
- Three categories of processes in LASSO-O:
  - Model
  - Observation
  - Obs & Model Blending
- Each process in the workflow is packaged as an RPM (i.e., binary)



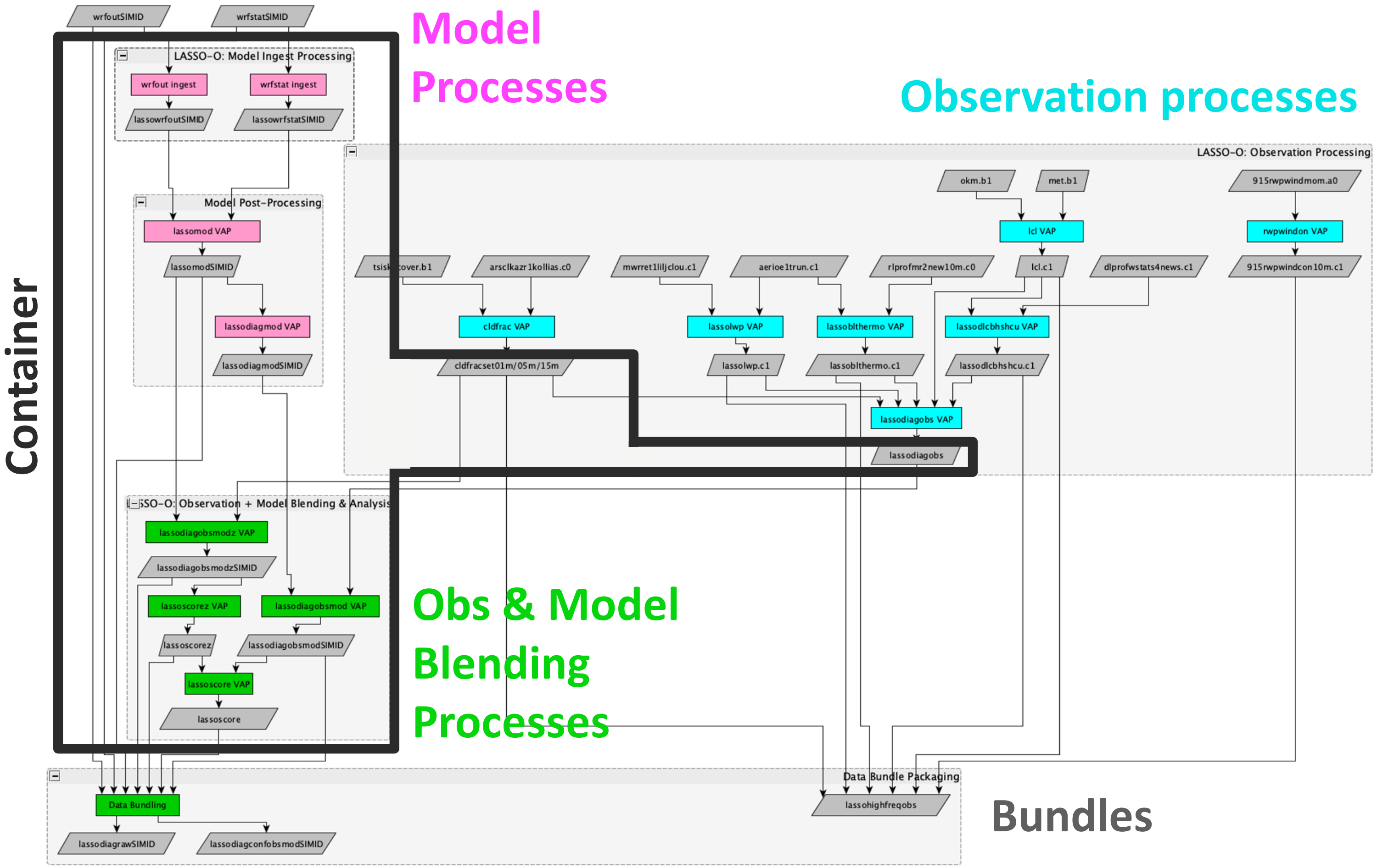
What is a container?

- Software packaged into standardized units which include all dependencies
- Supports reproducible results from one environment to another
- Easy to ship and deploy

- Common container technologies:
  - Docker
  - Singularity
  - Shifter

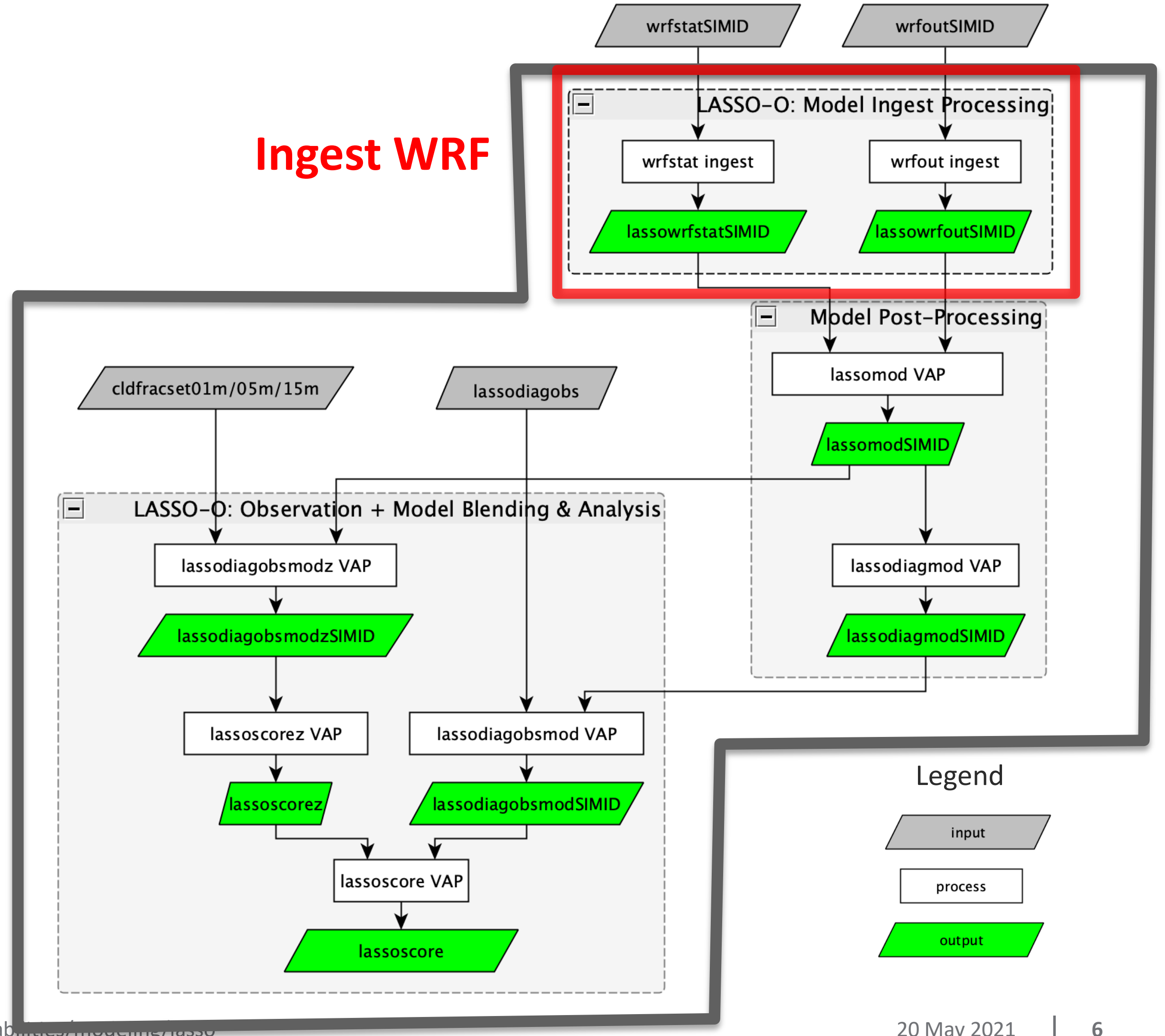
The LASSO-O Container

- We assembled into a container the RPMs and input data necessary to produce LASSO bundle data ***using non-LASSO WRF simulation input***
- This means modelers can run their own simulations through the container to:
  - Evaluate them against the same observations used by LASSO
  - Compare their simulations directly to LASSO simulations
  - Compute the same skill scores as LASSO
  - Develop new skill scores and easily apply them to simulations



LASSO-O Container Workflow

The portion of the LASSO-O workflow within the container is diagrammed at right

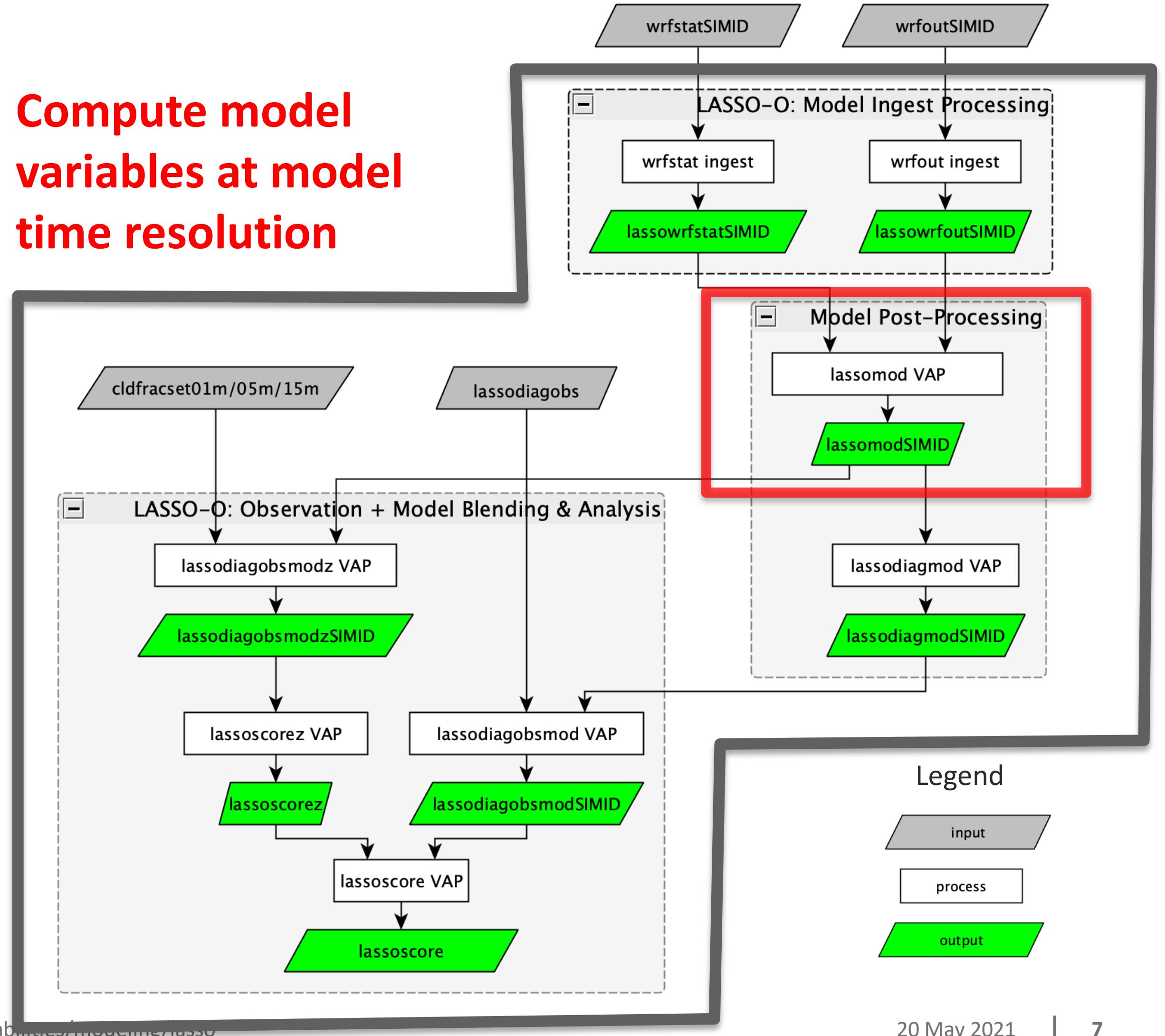


https://www.arm.gov/capabilities/modeling/lasso

LASSO-O Container Workflow

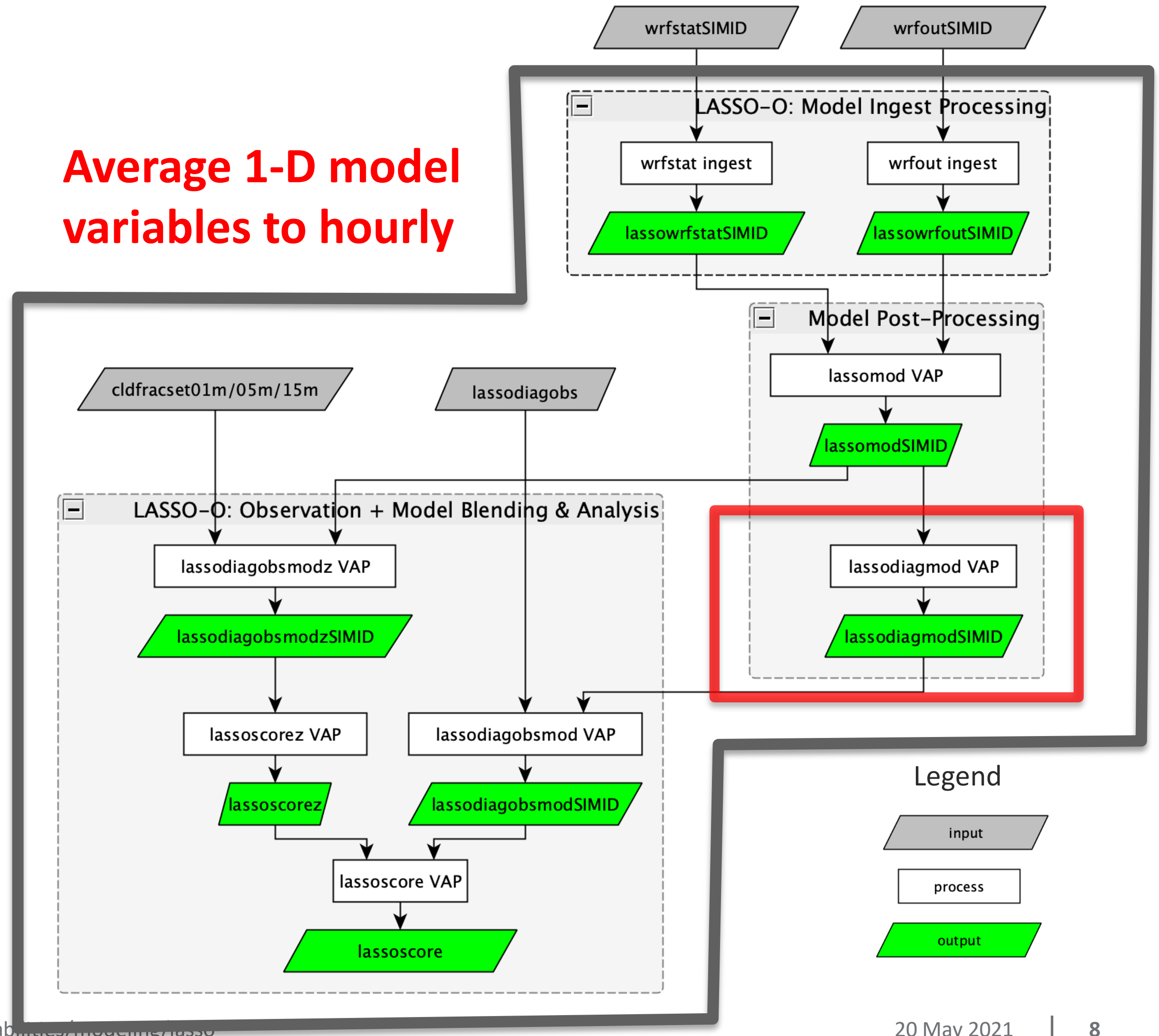
The portion of the LASSO-O workflow within the container is diagrammed at right

Compute model variables at model time resolution



LASSO-O Container Workflow

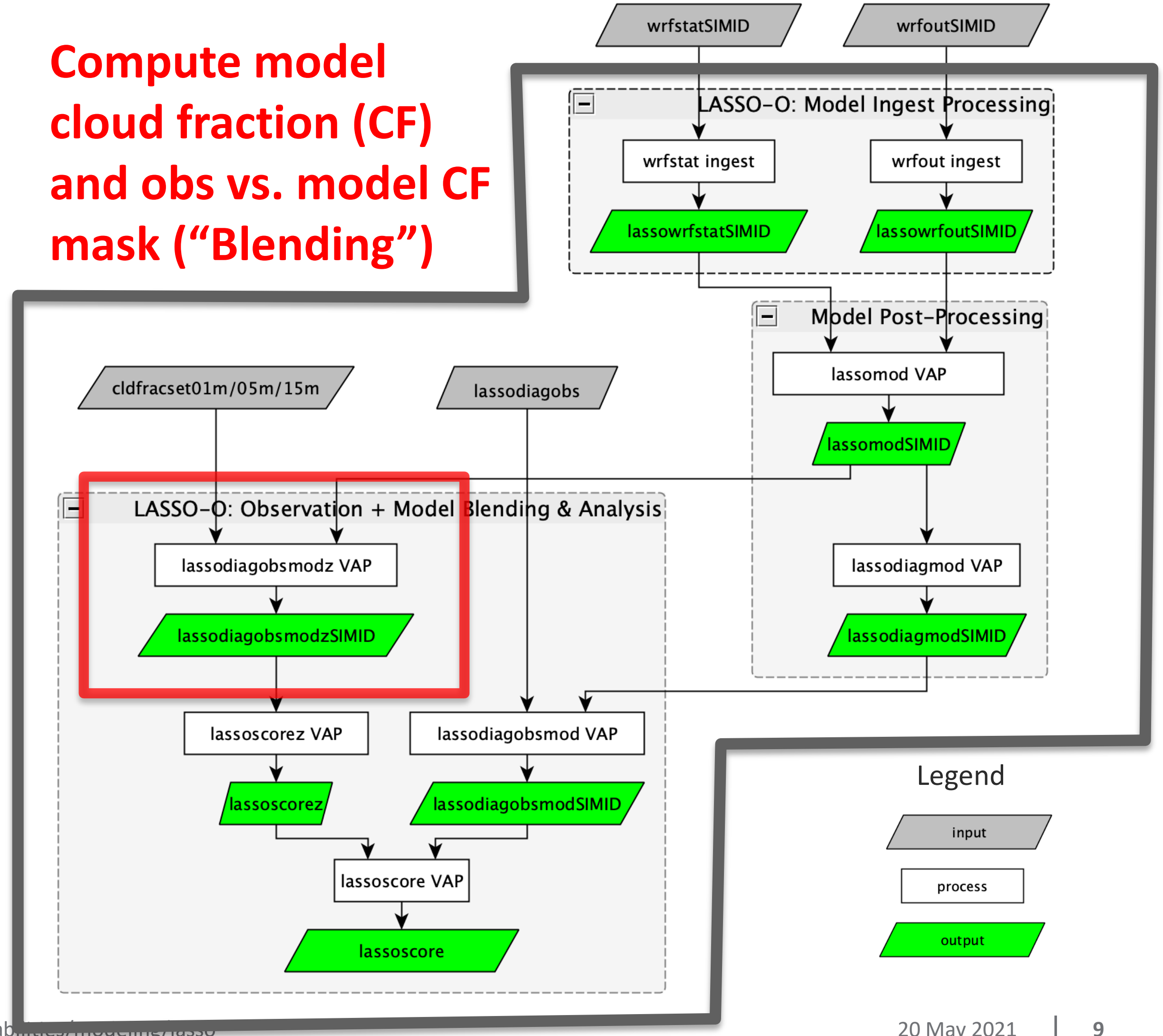
The portion of the LASSO-O workflow within the container is diagrammed at right



LASSO-O Container Workflow

The portion of the LASSO-O workflow within the container is diagrammed at right

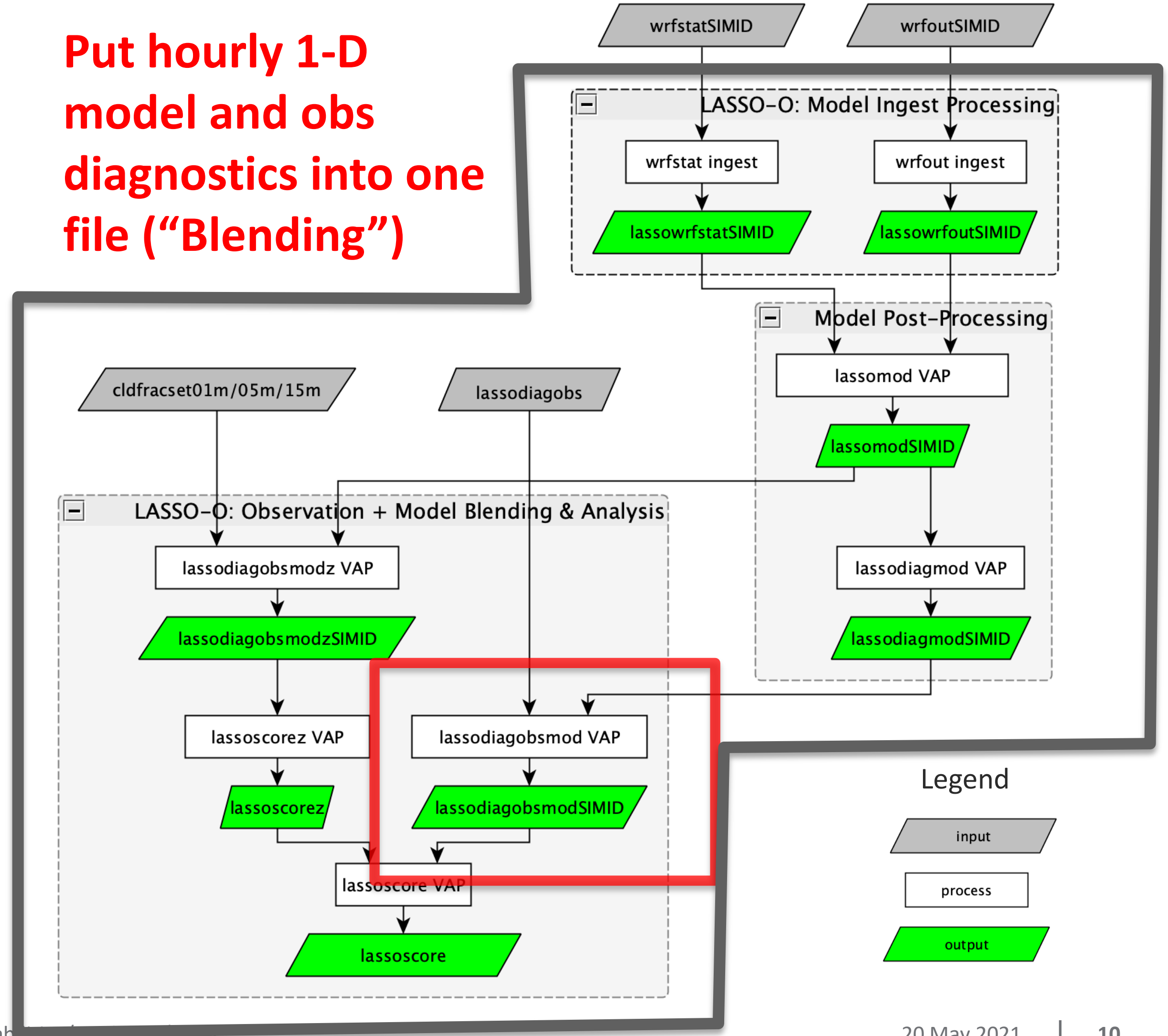
Compute model cloud fraction (CF) and obs vs. model CF mask ("Blending")



LASSO-O Container Workflow

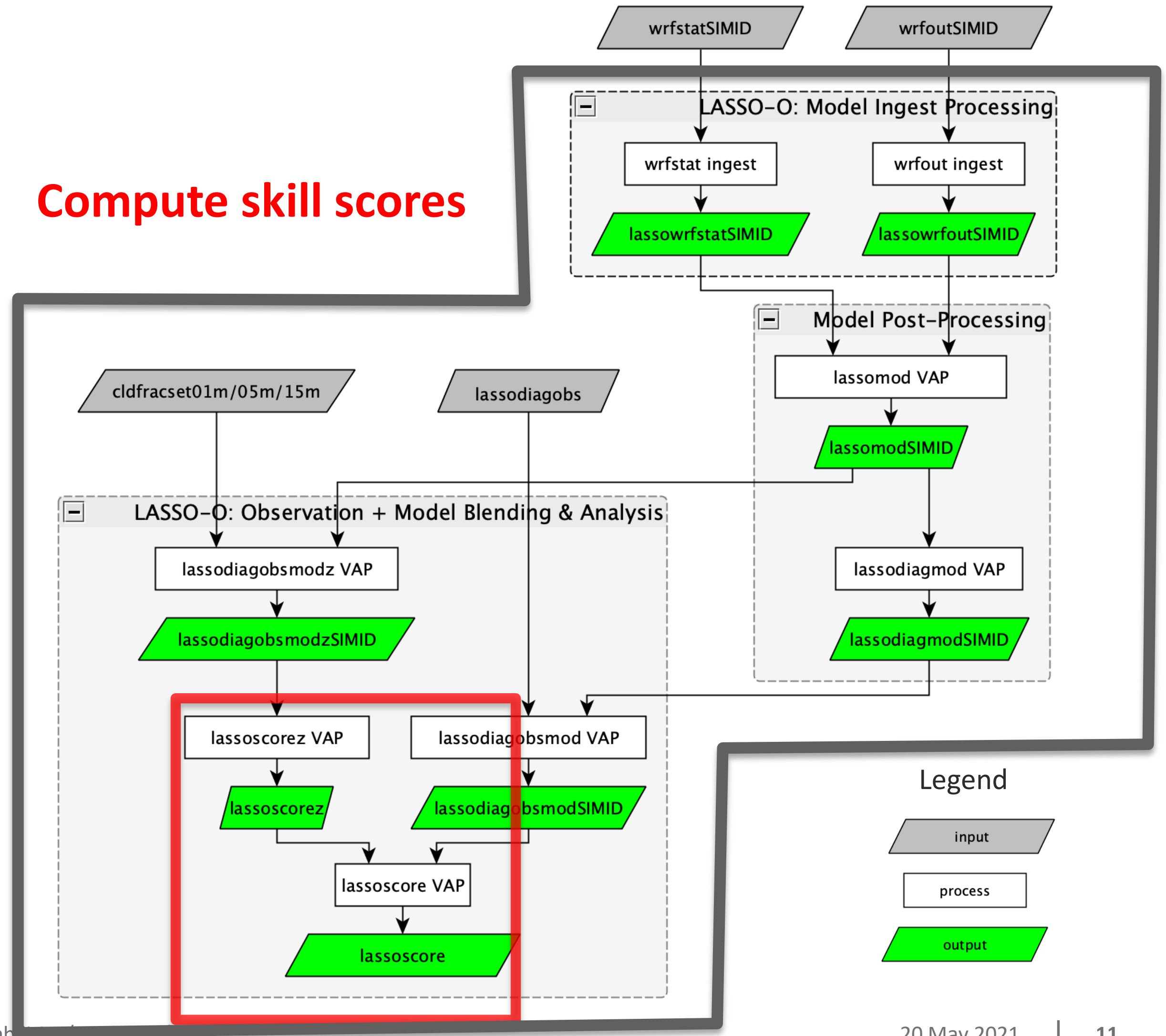
The portion of the LASSO-O workflow within the container is diagrammed at right

Put hourly 1-D model and obs diagnostics into one file (“Blending”)



LASSO-O Container Workflow

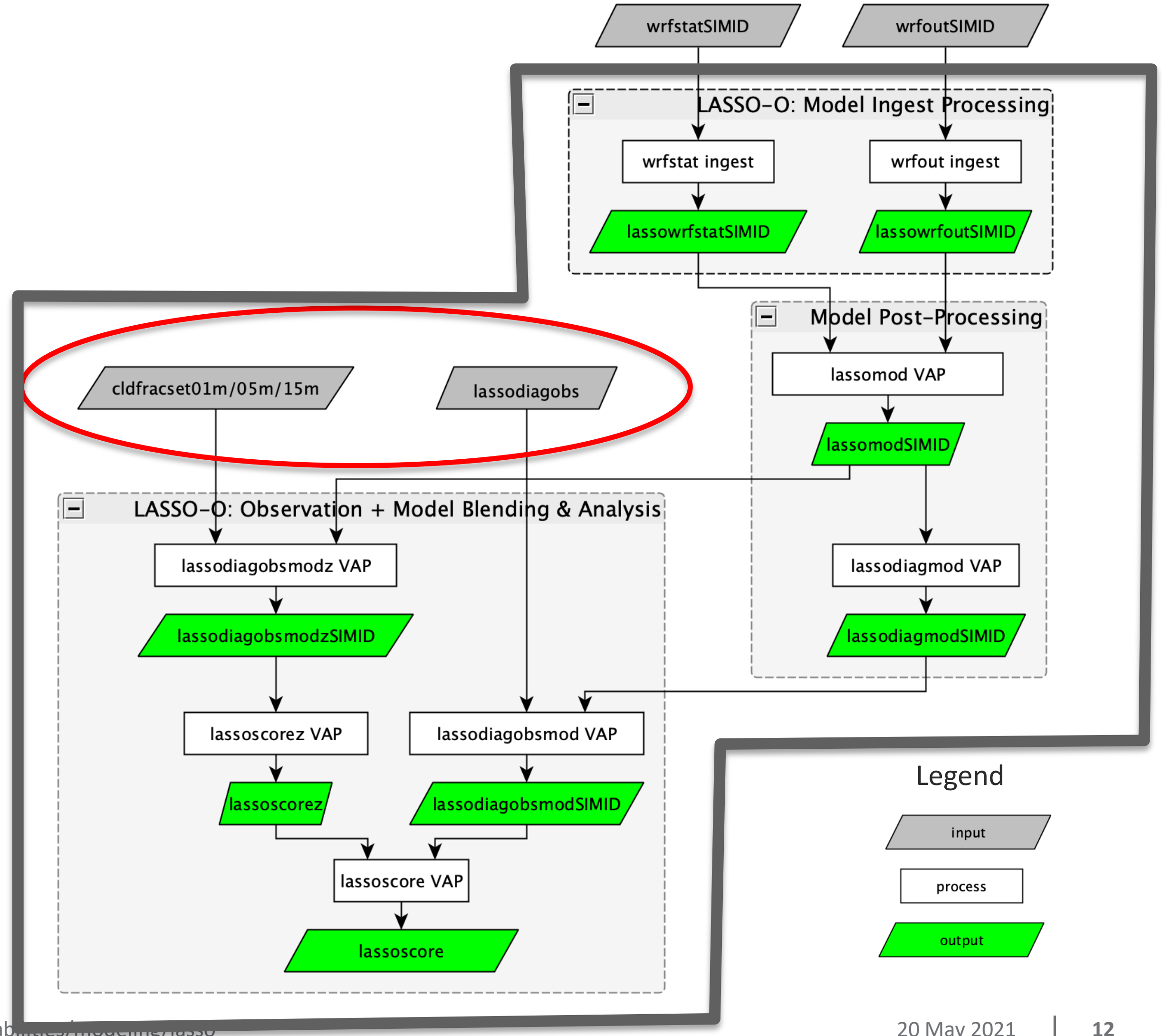
The portion of the LASSO-O workflow within the container is diagrammed at right



LASSO-O Container Input & Output

The portion of the LASSO-O workflow within the container is diagrammed at right

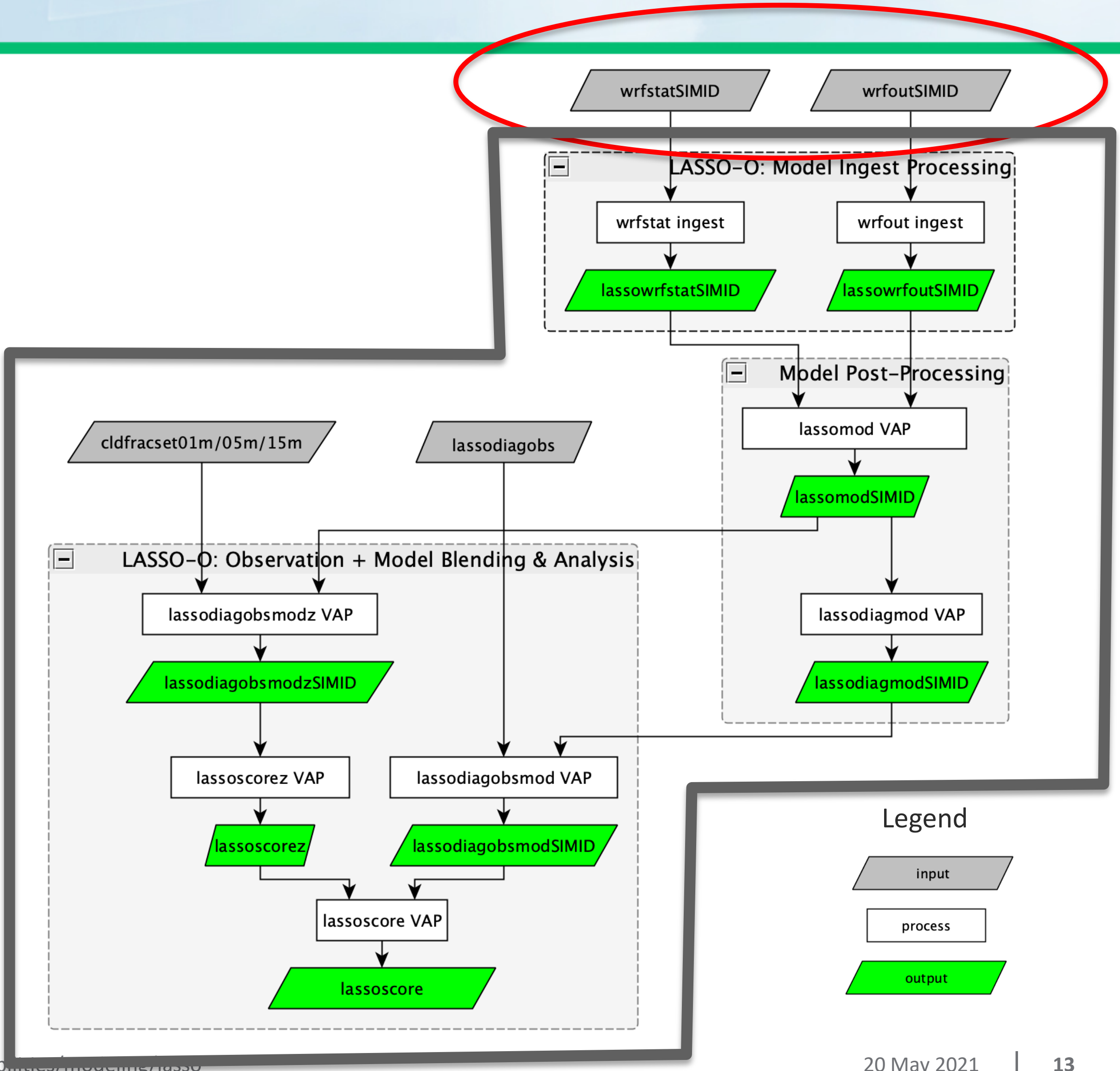
- Input provided in/by container:
  - Cloud fraction profile observations (cldfracset)
  - 1D (in time) observations (lassodiagobs)



LASSO-O Container Input & Output

The portion of the LASSO-O workflow within the container is diagrammed at right

- Input provided in/by container:
  - Cloud fraction profile observations (cldfracset)
  - 1D (in time) observations (lassodiagobs)
- Input provided by user:
  - WRF LES raw output (wrfstat and wrfout files)



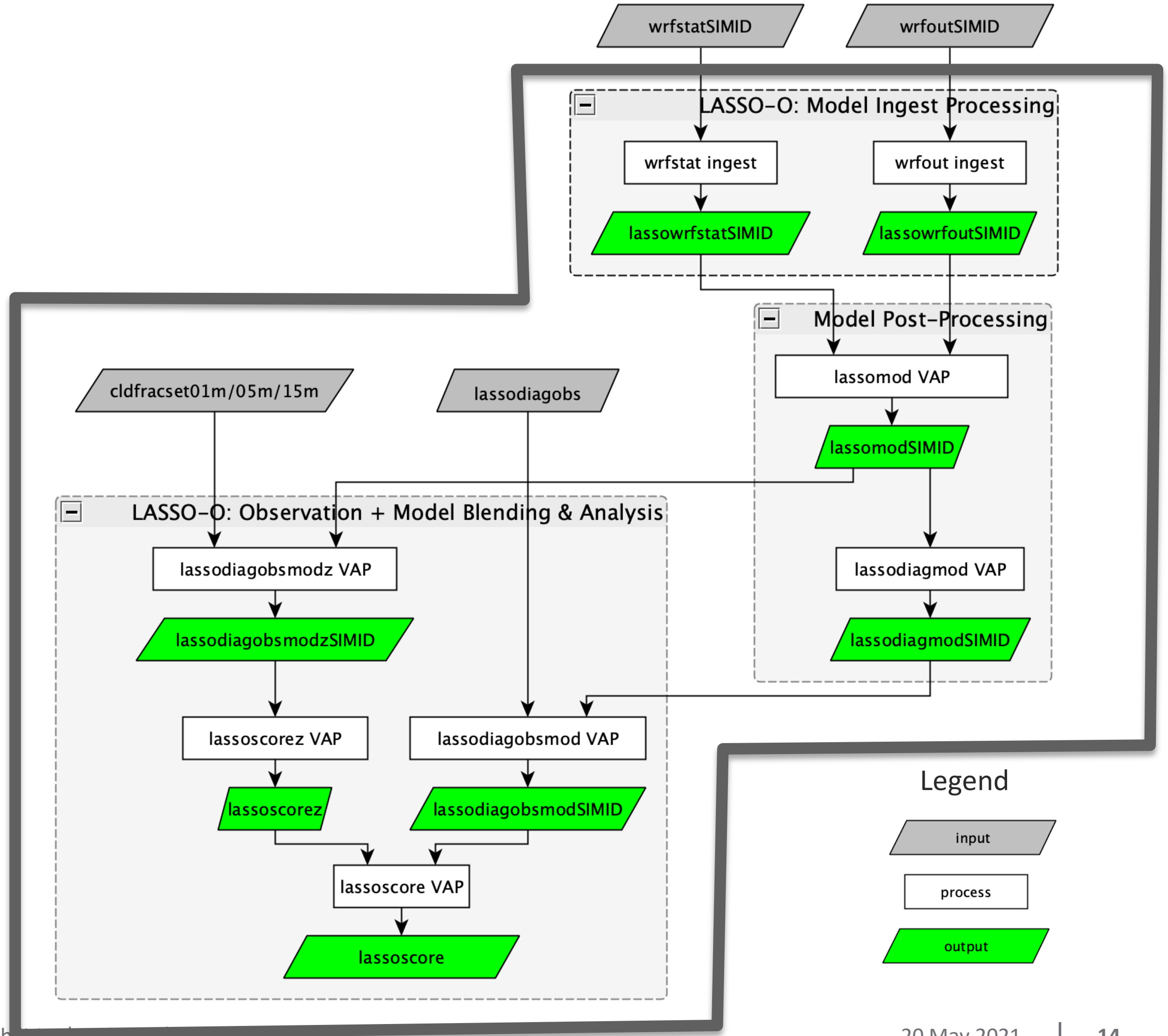
LASSO-O Container Input & Output

The portion of the LASSO-O workflow within the container is diagrammed at right

- Input provided in/by container:
  - Cloud fraction profile observations (cldfracset)
  - 1D (in time) observations (lassodiagobs)

- Input provided by user:
  - WRF LES raw output (wrfstat and wrfout files)

- Output (shown in green at right):
  - Ingested WRF data
  - Contents of lassodiagconfobsmodSIMID bundle



LASSO-O Container User Input

- WRF LES output: wrfstat and wrfout files ***… That's all!***

- Details:
  - User-provided WRF simulations must be for one of the LASSO case dates during 2017, 2018, or 2019.
  - User-provided WRF simulations must be for the same duration and times as the LASSO simulations.
  - LASSO WRF simulation output has extra global netCDF attributes (e.g., simulation_id_number, output_domain_size, output_number_of_levels, output_horizontal_grid_spacing, etc.), but these are NOT required for user-provided WRF simulation input.
  - User-provided WRF simulations may have custom horizontal grid spacing, domain size, number of levels, and time resolution.

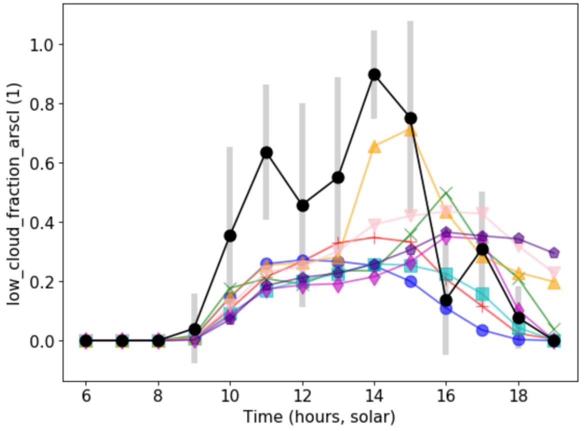
LASSO-O Container Output

- Output of container is the contents of lassodiagconfobsmod bundle:
  - Results of model post-processing to compute variables comparable to observations (lassomod, native model time resolution)

LASSO-O Container Output

- Output of container is the contents of lassodiagconfobsmod bundle:
  - Results of model post-processing to compute variables comparable to observations (lassomod, native model time resolution)
  - Time series of 1D variables for observations and model (lassodiagobsmod, hourly)

Note: All examples use case day 2018-07-10

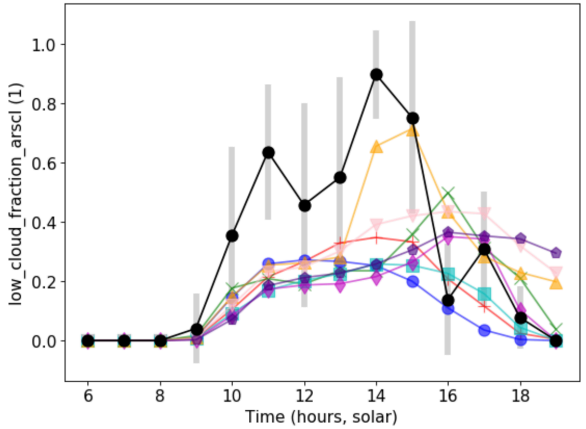


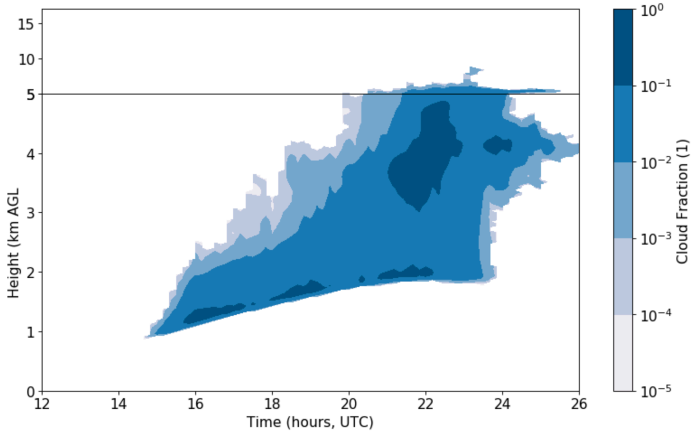
Plots produced from notebooks provided in the container repo:
https://code.arm.gov/lasso/containers/run-lasso-o_shcu/-/tree/master/notebooks

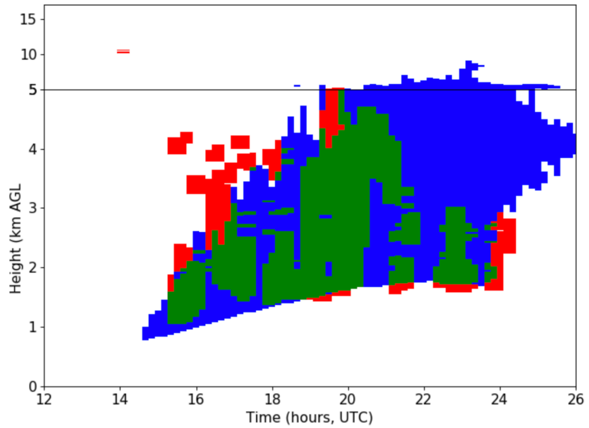
LASSO-O Container Output

- Output of container is the contents of lassodiagconfobsmod bundle:
  - Results of model post-processing to compute variables comparable to observations (lassomod, native model time resolution)
  - Time series of 1D variables for observations and model (lassodiagobsmod, hourly)
  - Model cloud fraction profile and observation vs. model cloud (lassodiagobsmodz, native model time resolution)

Note: All examples use case day 2018-07-10





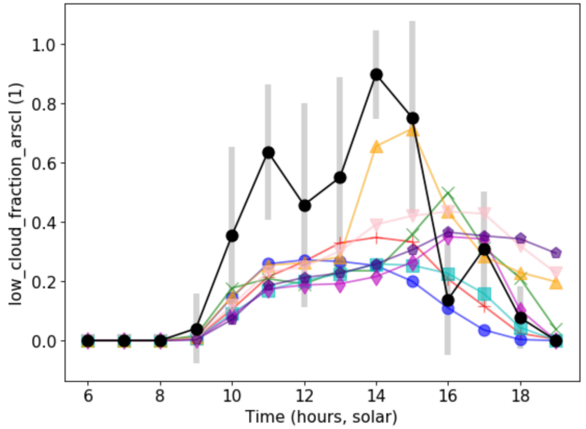


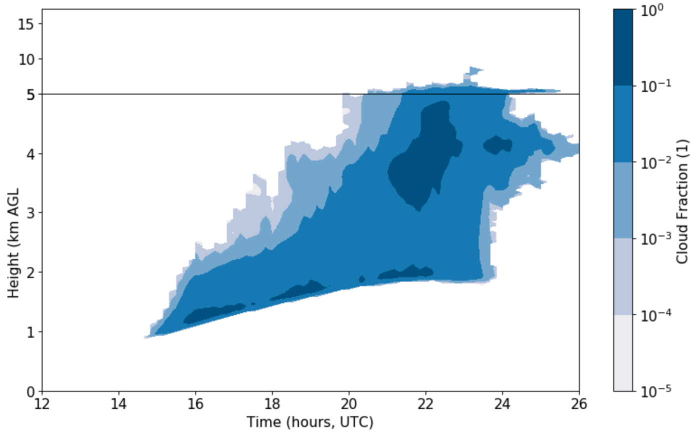
Plots produced from notebooks provided in the container repo:
https://code.arm.gov/lasso/containers/run-lasso-o_shcu/-/tree/master/notebooks

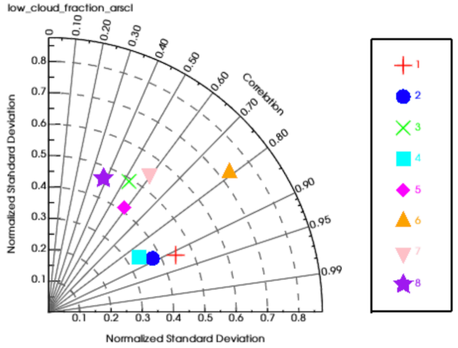
LASSO-O Container Output

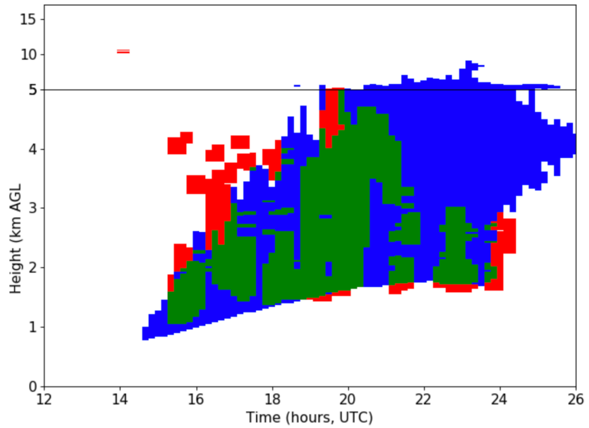
Note: All examples use case day 2018-07-10

- Output of container is the contents of lassodiagconfobsmod bundle:
  - Results of model post-processing to compute variables comparable to observations (lassomod, native model time resolution)
  - Time series of 1D variables for observations and model (lassodiagobsmod, hourly)
  - Model cloud fraction profile and observation vs. model cloud (lassodiagobsmodz, native model time resolution)
  - Skill scores (lassoscore, lassoscorez)









Plots produced from notebooks provided in the container repo:
https://code.arm.gov/lasso/containers/run-lasso-o_shcu/-/tree/master/notebooks

Getting Started Using LASSO-O in a Container

- You'll likely want to run the container from somewhere you can access your WRF output, like an HPC cluster.
- The computer you use must be able to deploy one of the container platforms: Docker, Singularity, or Shifter.
- Start by going to the LASSO-O container gitlab repository:
  - https://code.arm.gov/lasso/containers/run-lasso-o_shcu/-/tree/master/
- Clone the repository and review the instructions in README.md

Steps to use LASSO-O in a Container

1. Set up your container runtime environment:
   - See the README-DOCKER.md, README-SHIFTER.md, or README-SINGULARITY.md for instructions specific to the available platform

Steps to use LASSO-O in a Container

1. Set up your container runtime environment:
   - See the README-DOCKER.md, README-SHIFTER.md, or README-SINGULARITY.md for instructions specific to the available platform
2. Prepare Simulation Data
   - Place or symbolically link your WRF LES simulation wrfstat and wrfout files into the data/inputs directory →
   - Up to 10 simulations may be processed

```
$ ls data/inputs/*
data/inputs/sgpwrfout3C1.00:
wrfout_d01_2017-04-03_12:00:00.nc  wrfout_d01_2017-04-03_20:00:00.nc
wrfout_d01_2017-04-03_13:00:00.nc  wrfout_d01_2017-04-03_21:00:00.nc
wrfout_d01_2017-04-03_14:00:00.nc  wrfout_d01_2017-04-03_22:00:00.nc
wrfout_d01_2017-04-03_15:00:00.nc  wrfout_d01_2017-04-03_23:00:00.nc
wrfout_d01_2017-04-03_16:00:00.nc  wrfout_d01_2017-04-04_00:00:00.nc
wrfout_d01_2017-04-03_17:00:00.nc  wrfout_d01_2017-04-04_01:00:00.nc
wrfout_d01_2017-04-03_18:00:00.nc  wrfout_d01_2017-04-04_02:00:00.nc
wrfout_d01_2017-04-03_19:00:00.nc  wrfout_d01_2017-04-04_03:00:00.nc

data/inputs/sgpwrfstat3C1.00:
wrfstat_d01_2017-04-03_12:00:00.nc
```

Steps to use LASSO-O in a Container

1. Set up your container runtime environment:
   - See the README-DOCKER.md, README-SHIFTER.md, or README-SINGULARITY.md for instructions specific to the available platform
2. Prepare Simulation Data
   - Place or symbolically link your WRF LES simulation wrfstat and wrfout files into the data/inputs directory
   - Up to 10 simulations may be processed
3. Edit date and container_runtime in config.yml file

```yaml
#-----------------------------------------------------------------------
# Begin Datetime
#
# Enter the UTC start date and time for your simulation data
# in the format YYYYMMDD.HHMMSS
#
# The entered date MUST be one listed in the adjacent lasso_dates.txt
# file.
#-----------------------------------------------------------------------
begin_datetime: 20180710.115900
```

Steps to use LASSO-O in a Container

1. Set up your container runtime environment:
   - See the README-DOCKER.md, README-SHIFTER.md, or README-SINGULARITY.md for instructions specific to the available platform
2. Prepare Simulation Data
   - Place or symbolically link your WRF LES simulation wrfstat and wrfout files into the data/inputs directory
   - Up to 10 simulations may be processed
3. Edit date and container_runtime in config.yml file
4. Run the LASSO-O container!
   - Refer back to the README-DOCKER.md, README-SHIFTER.md, or README-SINGULARITY.md for the container-platform-specific run command.

Steps to use LASSO-O in a Container

1. Set up your container runtime environment:
   - See the README-DOCKER.md, README-SHIFTER.md, or README-SINGULARITY.md for instructions specific to the available platform
2. Prepare Simulation Data
   - Place or symbolically link your WRF LES simulation wrfstat and wrfout files into the data/inputs directory
   - Up to 10 simulations may be processed
3. Edit date and container_runtime in config.yml file
4. Run the LASSO-O container!
   - Refer back to the README-DOCKER.md, README-SHIFTER.md, or README-SINGULARITY.md for the container-platform-specific run command.

Note:

Expect a single simulation with 226 levels, a domain size of 14.4 km, and grid spacing of 100 m to take about an hour to complete.

Results

After the container runs, you'll find the following in your data directory:

```
data/inputs:
sgpwrfout4C1.00
sgpwrfstat4C1.00
```

Your WRF LES Input

```
data/outputs:
logs
```

```
sgpcldfracset01mC1.c1
sgpcldfracset15mC1.c1
sgplassodiagobsC1.c1
```

Input that was contained in the container, now accessible to you

```
sgplassodiagmod4C1.m1
sgplassodiagobsmod4C1.m1
sgplassodiagobsmodz4C1.m1
sgplassomod4C1.m1
sgplassoscoreC1.m1
sgplassoscorezC1.m1
sgpwrfout4C1.m0
sgpwrfstat4C1.m0
```

Output from the container

Plotting the Results

The LASSO-O container gitlab repository includes a notebooks folder which contains all the python code and instructions to produce plots from your results.

Prerequisites: Anaconda/Miniconda 3

Follow the instructions in the notebooks/README.md file to create the 'lasso' conda environment, which includes:

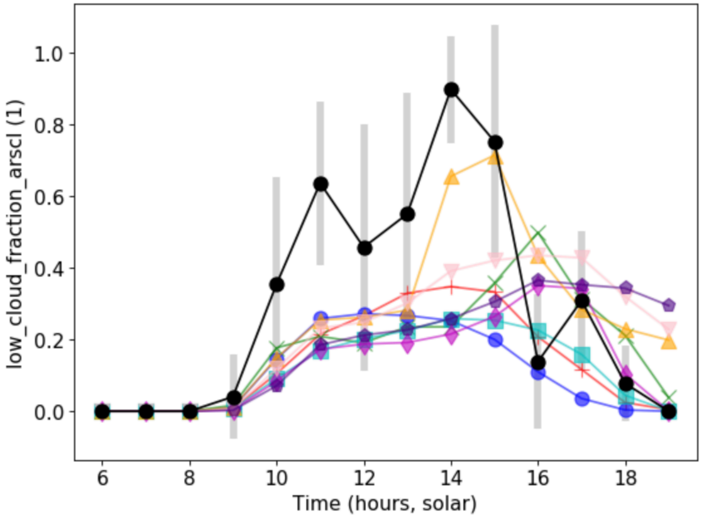
- CDAT (https://github.com/CDAT/cdat/wiki/install) is required for plotting the Taylor diagram in plot_1d.ipynb
- xarray is used for reading in most data
- netCDF4 is used for reading in data files with time-resolved height bins (e.g., data stream lassodiagobsmodz.m1)

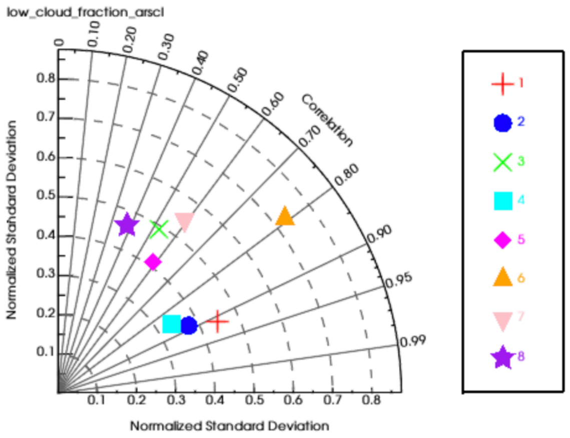
Plotting the Results

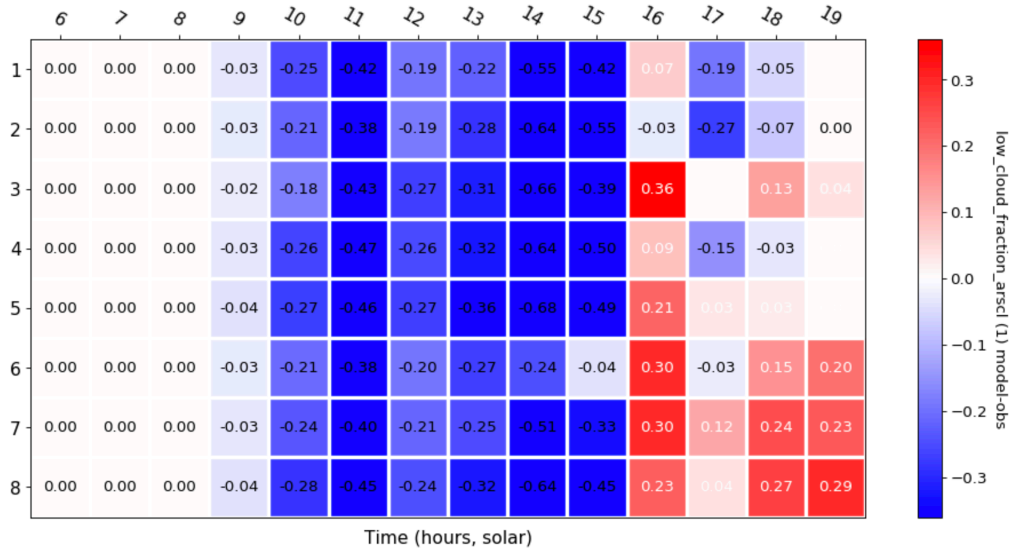
Note: All examples use case day 2018-07-10

The available notebooks include:

- plot_1D.ipynb for plotting time series, Taylor diagrams, and heatmaps







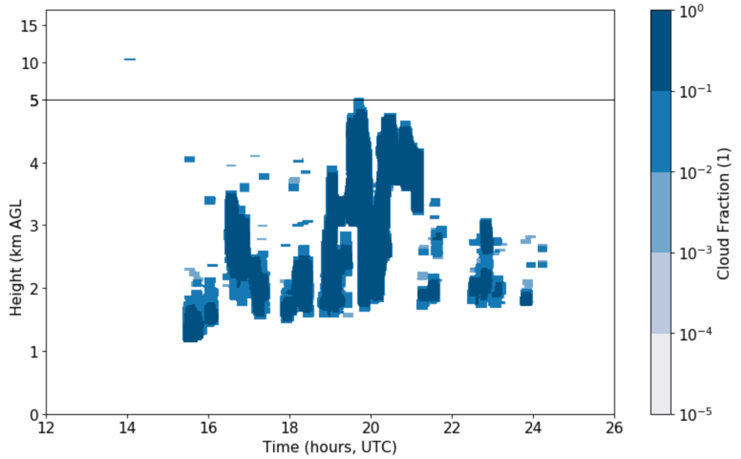
Plotting the Results

Note: All examples use case day 2018-07-10

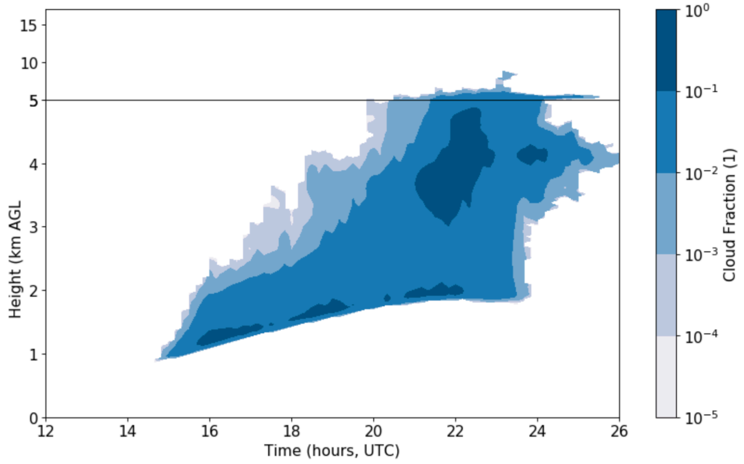
The available notebooks include:

- plot_1D.ipynb for plotting time series, Taylor diagrams, and heatmaps
- plot_cloud_fraction.ipynb for plotting time-height cloud fraction plots

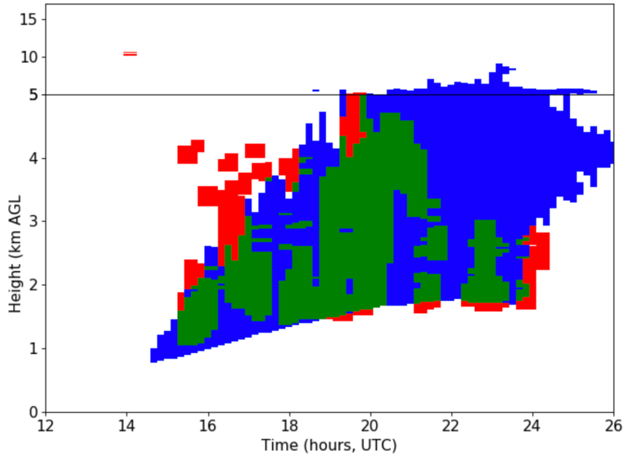
Observation



Model



Mask

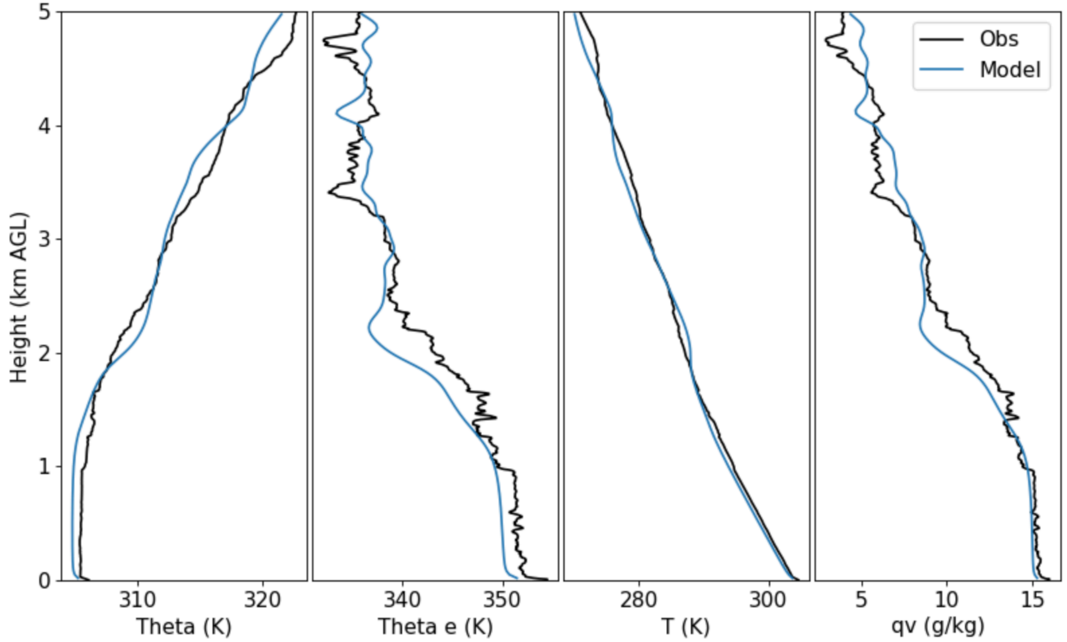


Plotting the Results

Note: All examples use case day 2018-07-10

The available notebooks include:

- plot_1D.ipynb for plotting time series, Taylor diagrams, and heatmaps
- plot_cloud_fraction.ipynb for plotting time-height cloud fraction plots
- plot_profiles.ipynb for plotting sounding profiles

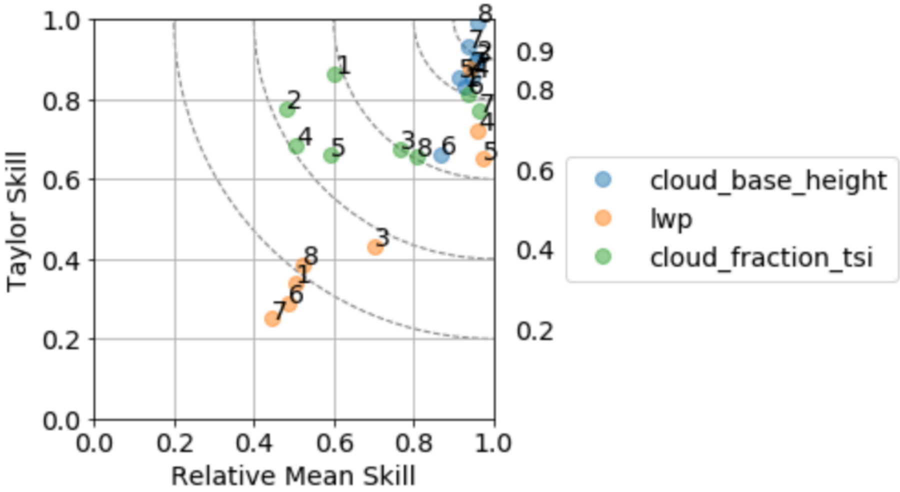


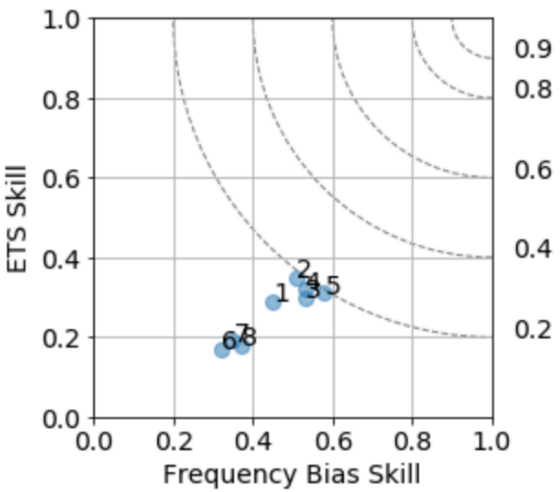
Plotting the Results

Note: All examples use case day 2018-07-10

The available notebooks include:

- plot_1D.ipynb for plotting time series, Taylor diagrams, and heatmaps
- plot_cloud_fraction.ipynb for plotting time-height cloud fraction plots
- plot_profiles.ipynb for plotting sounding profiles
- plot_scores.ipynb for plotting scatter plots of the LASSO skill scores





Summary

- LASSO-O in a container gives modelers the ability to process their own WRF LES models in the same way LASSO sims are processed, which streamlines model evaluation.

- LASSO-O in a container and all the documentation is available at:

https://code.arm.gov/lasso/containers/run-lasso-o_shcu

- Running LASSO-O in a container requires:
  - A computing platform with Docker, Singularity, or Shifter
  - Access to LASSO-WRF raw output