Analysing Scales of Precipitation

User Guide

Ana Ordoñez

Gill Martin

Robert Lee

Nick Klingaman

November 2021

LLNL-SM-829359

Acknowledgement

Work at LLNL was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344.

Thank you to Chia-Wei Hsu for testing the first draft of this manual. Thanks also to Charlotte Demott, Paul Ullrich, and Peter Gleckler for contributions to this project.

How to use this guide

This guide is divided into two sections: **Set Up** and **Running ASoP with CMEC driver**. The first section contains instructions for first-time set-up of cmec-driver and ASoP. The second section contains instructions for configuring ASoP, running CMEC driver, and finding results.

These instructions do not guide the user in interpreting the ASoP outputs. More information about the results can be found in the README documents within the ASoP code and in Klingaman et al. (2017).

Set Up

Installing conda

A conda installation is required for running the ASoP module in cmec-driver. You do not need to take any action if you already have a working conda installation. If you are unsure, consult the documentation for the system you are working on.

If conda is not installed on your system, there are several options for obtaining it. Miniconda and Anaconda are two popular choices. Miniconda can be downloaded here: https://docs.conda.io/en/latest/miniconda.html.

Users will be asked to create three new conda environments in this documentation. Example commands will be provided; however, your computing platform may have different instructions for creating new environments.

Install cmec-driver

If you already have a recent version of cmec-driver, no further action is needed. Otherwise, choose one of the two options presented below for installing cmec-driver.

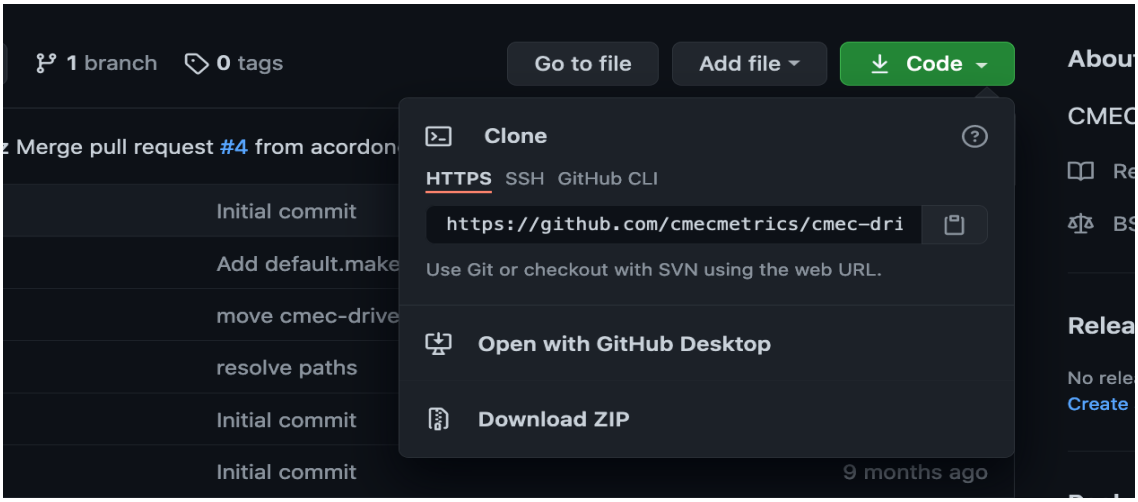
Install via conda-forge

Create a new conda environment for the cmec-driver. Use the conda install command to obtain "cmec_driver" (note this uses an underscore rather than a dash) from the conda-forge channel. The new environment does not need to have a specific name, but it should be one that is memorable and unique. This command creates an environment called "cmec":

```
conda create -n cmec -c conda-forge cmec_driver
```

Install from source code

Go to https://github.com/cmecmetrics/cmec-driver. Click the green "Code" button and copy the URL shown to clone via https (or use your preferred method for cloning).



Navigate to the directory that you want your cmec-driver repository to live in. Use the copied url in the following command to clone the repo:

```
git clone https://copy_url_here
```

Enter the cloned directory. Create a new conda environment and use pip to install cmec-driver. For example:

```
conda create -n cmec
conda activate cmec
pip install .
```

Clone ASoP

The ASoP repository is located at https://github.com/nick-klingaman/ASoP. Click the green "Code" button and copy the URL shown to clone via https (or use your preferred method for cloning).

Navigate to the directory that you want your ASoP repository to live in. Use the copied url in the following command to clone the repo:

```
git clone https://copy_url_here
```

Install ASoP environments

To run the ASop with cmec-driver, a conda environment is needed for each configuration. The environment for Spectral must be named "_CMEC_asop_spectral" and the environment for Coherence must be named "_CMEC_asop_coherence". Two yml files are provided to help the user create these environments, using the instructions in the next paragraph. However, these environments can also be created from scratch using the packages outlined in the README files within the ASoP source code. It is not guaranteed that the package versions provided in the environment yml files will work on every system.

Enter the ASoP repository. Use the following conda commands from within the ASoP directory to install the conda environments called _CMEC_asop_coherence and _CMEC_asop_spectral:

```
conda env create --file ASoP-Coherence/asop_coherence_env.yml
conda env create --file ASoP-Spectral/ASoP1_Spectral/asop_spectral_env.yml
```

Use the following command to list your environments and verify that the ASoP environments were created:

```
conda env list
```

Register ASoP with cmec-driver

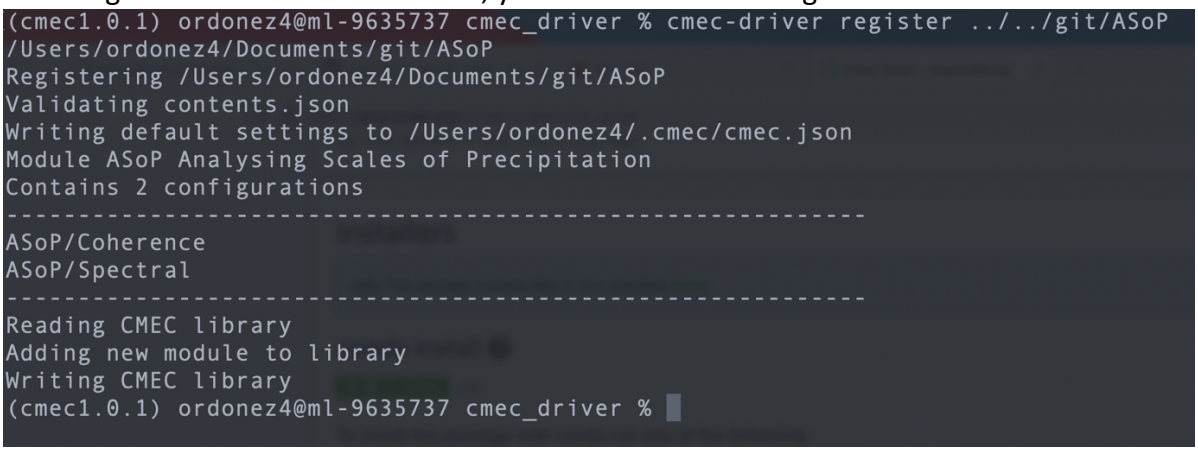
Activate the conda environment in which cmec-driver is installed:

```
conda activate your_cmec_environment
```

If installing ASoP for the first time, use the following command to register ASoP, using the path to the ASoP directory:

```
cmec-driver register path_to_asop_repo_here
```

If the register command is successful, you should see a message like this:



Verify that the cmec configuration file "cmec.json" exists. This file will be found in a hidden folder called ".cmec" in your user home directory. Try "ls ~/.cmec/" in the terminal to see if this file is present.

If the .cmec folder does not exist after registering ASoP, do the following. Go to your home directory and create .cmec ("mkdir .cmec"). Unregister ASoP (see section "Updates to ASoP"). The register ASoP again.

Register conda environment information

This section only needs to be completed if you have newly installed cmec-driver and have not previously registered the conda environment information.

Activate the conda environment in which cmec-driver is installed:

```
conda activate your_cmec_environment
```

Setup

Add the paths for your conda executable and environment directory to your cmec library file. This is a **one-time command** (unless your conda installation changes in the future). At runtime, cmec-driver makes these paths accessible to modules as environment variables.

The command is:

```
cmec-driver setup \
--conda_source path_to_conda_executable \
--env_root path_to_conda_env_directory
```

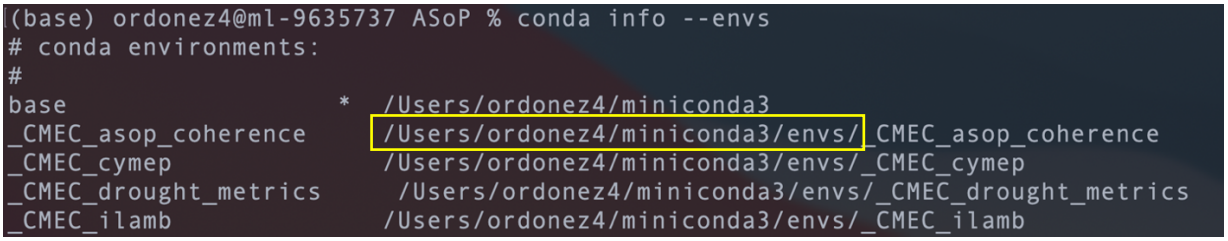
Example:

```
cmec-driver setup \
--conda_source ~/miniconda3/etc/profile.d/conda.sh \
--env_root ~/miniconda3/envs
```

How to find your conda environment information

The conda executable is the file that is sourced to activate conda. In a typical miniconda installation, the executable path looks like "miniconda3/etc/profile.d/conda.sh". If you allowed conda to edit your .bashrc or .bash_profile files during installation, you can likely find this path in the "conda initialize" block of the edited file.

The environment folder is the folder that your cmec environments are located in. By default, in miniconda this is "miniconda3/envs/". The "conda info --envs" command will display the paths to your environments, which can help locate which folder to use. The screenshot below shows an example. Consult the documentation for your conda installation if you are unsure.



Updating and removing conda information

Running the setup command with new paths will overwrite the previously saved paths in your cmec library file. You can also run the next command to remove your saved conda information from your cmec library file:

```
cmec-driver setup --clear_conda
```

Updates to ASoP

To pull updates from the ASoP repository, enter the ASoP repository and type:

```
git pull origin master
```

If your local ASoP repository is moved after being registered with cmec-driver, it must be registered again.

```
cmec-driver unregister ASoP
cmec-driver register path_to_new_asop_repo
```

Running ASoP with CMEC driver

Set up data

Users must create two input and one output folder for cmec-driver. If you have not run cmec-driver before, create these folders now. Instructions are provided here using the default cmec-driver folder names that will be referenced in later instructions. However, users may choose any names for these folders.

Do not create these folders in the ASoP code directory. It is recommended to create a separate "cmec-driver" directory as a location for running cmec-driver.

In the cmec-driver directory, create an "obs" directory, a "model" directory, and an "output" directory, i.e.:

```
mkdir obs
mkdir model
mkdir output
```

ASoP Coherence

Move netCDF time series data to the "obs" and "model" folders. "Obs" data is optional and is present because the user may also be running other metrics for which this folder is required. ASoP Coherence will loop through all the datasets in both folders, and it produces the same types of metric and figure outputs for models or observations.

Do not use the histogram cube netCDF files created by ASoP-Spectral as inputs for ASoP Coherence. ASoP Coherence will fail if any of the ASoP Spectral cube files are in the input folders.

ASoP Spectral

ASoP Spectral expects 1 control dataset in "obs" and 1 or more "model" datasets in netCDF format. Move a copy of 1 or more model datasets into the "model" folder. Add one observational or model control dataset to "obs". Either netCDF time series or preprocessed histogram files can be used. Histogram file names must end with "_hist.nc".

If running the histogram metric, do not put the mask file in the "model" or "obs" folder. Enter the absolute path to the mask file in cmec.json (described in CMEC Configuration section).

CMEC configuration file

Once you register a package with cmec-driver for the first time, a user configuration file will be created. This JSON file is a one-stop-shop for user settings that need to be modified at runtime. Changes can be made in a text editor, and they are immediately available to cmec-driver once saved.

The location of this file changed with cmec-driver version 1.0.1. For version 0.9, this file is located at cmec-driver/config/cmec.json. For verson 1.0.1 and higher, it is located in a hidden folder in the user's home directory at .cmec/cmec.json.

If you unregister a module, that module's settings are removed from cmec.json. A warning will print if cmec.json is malformed and cmec-driver is unable to open it.

All the settings for a given module and configuration are contained in a JSON array. The key for the array is either the name of the module or module/configuration (e.g. ASoP/Coherence or ASoP/Spectral).

Note that the keys, their default settings and applications are independent, and may differ, between the ASoP Spectral and ASoP Coherence modules. See "CMEC configuration options".

Run cmec-driver

Activate your cmec-driver conda environment.

The general command to run cmec-driver is:

```
cmec-driver run --obs obs_dir model_dir output_dir
Module/Configuration
```

Example:

```
cmec-driver run --obs obs model output ASoP/Spectral
```

The obs folder is optional and uses a flag. The other directory inputs are required positional arguments.

When you run the command, cmec-driver will ask if you want to overwrite the output directory. Answer "y" to proceed. If you answer "n", cmec-driver will immediately exit without overwriting the directory.

```
(cmec1.0.1) ordonez4@ml-9635737 cmec_driver % cmec-driver run --obs obs model output ASoP/Spectral
Reading CMEC library
Identifying drivers
The following 1 modules will be executed:
------------------------------------------------------------
MODULE_NAME: ASoP/Spectral
MODULE_PATH: ASoP/Spectral
  /Users/ordonez4/Documents/git/ASoP/ASoP-Spectral/ASoP1_Spectral/asop_spectral.sh
------------------------------------------------------------
The following environment variables will be set:
------------------------------------------------------------
CMEC_OBS_DATA=/Users/ordonez4/Documents/Demo/cmec_driver/obs
CMEC_MODEL_DATA=/Users/ordonez4/Documents/Demo/cmec_driver/model
CMEC_WK_DIR=/Users/ordonez4/Documents/Demo/cmec_driver/output/$MODULE_NAME
CMEC_CODE_DIR=$MODULE_PATH
CMEC_CONFIG_DIR=/Users/ordonez4/.cmec
------------------------------------------------------------
Creating output directories
Path /Users/ordonez4/Documents/Demo/cmec_driver/output/ASoP/Spectral already exists. Overwrite? [y/n]
```

CMEC configuration options

A note about JSON format and line spacing

These examples show arrays and objects being written across multiple lines. Each object and array could also be written on a single line. CMEC driver writes JSONs with a 4-space indent and a single object or array entry on each line.

The following screenshot shows a snippet from cmec.json with the default settings from ASoP Coherence and ASoP Spectral.

```
"ASoP/Coherence": {
    "regions": {
        "Indian Ocean": [
            -10,
            10,
            60,
            90
        ]
    },
    "grid": "native"
},
"ASoP/Spectral": {
    "regions": {
        "Indian Ocean": [
            -10,
            10,
            60,
            90
        ]
    },
    "mask": null
}
```

Coherence

There are no required settings for ASoP-Coherence. The module either assigns default values or estimates settings from the input data. See Appendix 1 for details about these default settings.

Sometimes the default value may be incompatible with a dataset or science question. Options are provided to specify settings for all datasets ("global keys") or for a single dataset. Settings for individual datasets take precedence over those for all datasets.

Two examples are provided below to demonstrate how to change the settings in cmec.json.

Examples of global keys

figure_type: (optional, string) A valid figure extension. Do not include a leading period. Default is "png"

regions: (optional, object) A dictionary of region names and coordinates. Coordinates are [S N W E]. Default coordinates are [-10,10,60,90]. Region definitions should not cross grid boundaries and should use grid units.

grid: (optional, string) Recommended values are "native" or "interpolated". Default is "native".

box_size: Length of sub-regions (square boxes) to consider for correlation analysis as a function of physical distance (in km, recommended value is > 6.5*dx). By default, this is 7*dx.

lag_length: Maximum lag to consider for correlation analysis as a function of model gridpoints, for constructing distance vs. lag correlation diagrams. Correlations for lags from 0 (coincidence) until lag_length will be computed. By default, the lag length is 6.

Example: The following object resets the lag_length and box_size for all datasets, along with the grid and region. Figures will be created as PostScript files:

```
"ASoP/Coherence": {
     "box_size": 1200,
     "lag_length": 10,
     "grid": "native",
     "regions": {"Australia": [-41 -5 104 160]},
     "figure_type": "ps"
}
```

Examples of dataset-specific keys

To set data dictionary keys for individual datasets, use the file base name as the key and create a new object with the desired settings. A key set for a single dataset will take precedence over a key set for all datasets.

The "region" key cannot be set for individual datasets. All datasets use the same region.

name: (optional, string) Name for model in plot files (no spaces).

legend_name: (optional, string) Name for model in legends and titles on plots (spaces allowed)

Example: This example applies the "name", "legend_name" and "box_size" settings to a single dataset called pr_file_name.nc, but applies "grid" to all datasets:

```
"ASoP/Coherence": {
     "grid": "interpolated",
     "pr_file_name.nc": {
          "name": "precip_1",
          "legend_name": "My model",
          "box_size": 800
     }
}
```

Spectral

Data dictionary keys can be set here either as global values for all datasets or for individual datasets. See Appendix 2 for all the data dictionary key definitions that can be set in the JSON file. A key set for a single dataset will take precedence over a key set for all datasets.

Note for histogram metrics: To create the histogram metrics and metrics maps, **all input datasets** must use the same timescale and mask, and must be on the same grid. ASoP-Spectral will run these metrics only if the "mask" and "timescale-all" keys are present in the settings. While not enforced, datasets should also belong to the same season and dates.

Examples of global keys

figure_type: (optional, string) A valid figure extension. Do not include a leading period. Default is "png".

regions: (optional, object) A dictionary of region names and coordinates. Coordinates are [S N W E]. Default coordinates are [-10.0,10.0,60.0,90.0]. Region definitions should not cross grid boundaries and should use grid units.

timescale-all: (string) Timescale for all datasets. Required for histogram metrics. Optional for other results. Will be used in legends, titles, and filenames if provided. This variable is

only used in a descriptive capacity by AsoP-Spectral. Do not set if input datasets have different timescales.

season-all: (optional, string) Season for file names. Must be set to have season appear in histogram metric plot title. Will be used in other legends, titles, and filenames if provided. This variable is only used in a descriptive capacity by AsoP-Spectral. Do not set if input datasets use different seasons.

mask: (string) Full path to mask file, empty string, or null, for histogram metric plot. Setting this key to a valid file name tells ASoP-Spectral to run the histogram metrics. This key is required for the histogram metrics, but do not set this key if you do **not** want to produce the histogram metrics. Do not place the mask file in the model or obs input folder.

Example: The following is a general example:

```
"ASoP/Spectral": {
     "regions": {"Australia": [-41 -5 104 160]},
     "timescale-all": "3hr",
     "mask": "/Users/user1/data/mask_file.nc",
     "season-all": "JJA"
}
```

Examples of Dataset-specific keys

To set keys for individual datasets, use the file base name as a key and create a new object with the desired settings within the ASoP/Spectral object (example at end of section). These keys will be used in the dataset name for figures and filenames.

The "regions" key cannot be set for an individual dataset.

The global "-all" version of the timescale, dates, or season setting, if present, will take precedence over a dataset-level setting in legends, titles, and file names.

name: (optional, string) A short, descriptive, and unique name for this dataset. Will be used in legends, titles, and filenames if provided. By default, ASoP spectral will search the file metadata for the keys "source_id", "short_name", "name", "source", and "model" to use as a name. It will also add the "variant_label" value if present. If no name is found in the settings or metadata, the file will be called "no_name".

timescale: (optional, string) Timescale for single dataset. Will be used in legends, titles, and filenames if provided. If running histogram metrics, the global setting "timescale-all" must be set. The dataset-specific timescale will not be used for that figure title. This variable is only used in a descriptive capacity by AsoP-Spectral.

Example: The following settings will result in a dataset name of "precip_1_3hr" for file "pr_file_name_1.nc". This name will appear in figure titles and legends. Any other input datasets are not specified in these settings, and they will use the default name.

```
"ASoP/Spectral":{
     "regions": {"Australia": [-41 -5 104 160]},
     "pr_file_name_1.nc": {
          "name": "precip_1",
          "timescale": "3hr"
     }
}
```

Results

If the run succeeds, results will be located at "output/ASoP/Spectral" or "output/ASoP/Coherence" ("output" being the output folder provided to cmec-driver). From the results folder, open "index.html" in your web browser to view the plots. Depending on your choice of figure extension and browser, you might see download links rather than images if your figure format is not supported.

A file called "output.json" in the output directory lists all of the module outputs and provides a short description of each.

If you provide the same output directory name the next time you run the same metric with cmec-driver, it will overwrite the directory. To keep your previous results, move them to a new location or provide a different output folder name in the future.

ASoP-Coherence generates data dictionaries on the fly when settings are not provided by the user. The dictionaries are printed to the asop_coherence log (asop_coherence.log.txt) so that users can evaluate the module-generated settings.

The ASoP-Spectral settings are also printed to a log called asop_spectral.log.txt.

ASoP-Coherence also provides a CSV file containing metrics and dataset information for each region. This is located in the asop_metrics subdirectory of the results folder. The file name format is <region>_summary.csv. The intermittency metrics can also be found in a CMEC-compliant JSON to asop_metrics/intermittency_metrics.json.

ASoP-Spectral creates a histogram metrics JSON file if those metrics are run. That file is found in the results directory under asop_metrics/histogram_metrics.json.

Appendix 1

ASoP Coherence data dictionary keys and definitions

There are no required run-time settings for ASoP-Coherence. Some settings have default values, and others are estimated on-the-fly from the input datasets. However, there can be cases where the default value is not satisfactory. Options are provided in the CMEC driver settings file to set custom key-value pairs for all datasets or for each individual dataset. Settings for individual datasets take precedence over those for all datasets. Any keys in the configuration file that do not match one of the below keys will be ignored. See section "CMEC configuration options" for examples.

Global-only keys

regions: Regions of data to read. Formatted as a JSON object containing `region_name: [coordinates]` pairs. The coordinate order is [minlat, maxlat, minlon, maxlon]. The default region is {"Indian Ocean": [-10,10,60,90]}. Region definitions should not cross grid boundaries and should use grid units.

Global or dataset-specific keys

These can be set either globally or for individual datasets. Keys set for individual datasets will take precedence over keys set for all datasets.

name: Name for model in plot files (no spaces). By default, ASoP will use (in order of precedence) the source_id, source_label, short_name, name, or long_name attribute if present. If the variant_label attribute exists, that will be appended to the name. If none of these attributes are found, the first 15 character of the file name will be used. This setting is purely for labeling purposes.

legend_name: The model name to display in legends and titles on plots (spaces allowed). By default, this is equal to the "name" key, with underscores replaced by spaces. This setting is purely for labeling purposes.

dt: Timestep of input data (in seconds). By default, this is calculated by differencing the first two timesteps and converting the difference into seconds.

dx: Longitudinal grid spacing at equator (in km). By default, this is the average difference between longitude points multiplied by a factor of 110 to convert to km. For each region this is then multiplied by the cosine of the mean latitude of the region in radians.

dy: Latitudinal grid spacing (in km). By default, this is the average difference between latitude points multiplied by a factor of 110 to convert to km.

constraint: Standard_name of data to read from netCDF file (e.g., precipitation flux). By default this is obtained from the "standard_name" attribute. If datasets contain multiple variables, this key should be set in the configuration file to specify the precipitation variable.

scale_factor: Multiplier necessary to convert precipitation to units of mm day-1 (kg m-2 day-1). By default, this is determined using the precipitation units and dt.

box_size: Length of sub-regions (square boxes) to consider for correlation analysis as a function of physical distance (in km, recommended value is > 6.5*dx). By default, this is 7*dx.

color: Name of color to use in line graphs (must be recognised by matplotlib). By default, the colors are taken from the matplotlib 'Paired' colormap.

region_size: Length of sub-regions (square boxes) to consider for correlation analysis as a function of model gridpoints (including lag correlations, see below) Use units of native gridpoints. Odd integers are strongly recommended. By default, this is 7.

lag_length: Maximum lag to consider for correlation analysis as a function of model gridpoints, for constructing distance vs. lag correlation diagrams. Correlations for lags from 0 (coincidence) until lag_length will be computed. By default, the lag length is 6.

autocorr_length: The maximum autocorrelation to analyse (in seconds). By default, this is 8*dt.

grid_type: A string describing the grid, used in output filenames (recommend no spaces). By default, this is "native". If data is interpolated, it is suggested to use "interpolated" in the settings. This setting is purely for labeling purposes.

time_type: A string describing the temporal sampling, used in output filenames (recommend no spaces). By default, this is determined by the size of dt. This setting is purely for labeling purposes.

grid_desc: A string describing the grid, used in plot titles (can contain spaces) but NOT in legends (use legend_name key). By default, this is "native". If data is interpolated, it is suggested to use "interpolated" in the settings. This setting is purely for labeling purposes.

time_desc: A string describing the temporal sampling, used in plot titles (can contain spaces) but NOT in legends (use legend_name key). By default, this is determined by the size of dt. This setting is purely for labeling purposes.

Appendix 2

ASoP Spectral data dictionary keys and definitions

There are no required run-time settings for ASoP-Spectral. Options are provided in the CMEC driver settings file to set custom key-value pairs for all datasets or for each individual dataset. Settings for all datasets take precedence over those for individual datasets. Any keys in the configuration file that do not match one of the below keys will be ignored.

Global keys

figure_type: (optional, string) A valid figure extension. Do not include a leading period. Default is "png".

regions: (optional, object) A JSON object of region names and coordinates. Coordinates are [minlat, maxlat, minlon, maxlon]. The default region is {"Indian Ocean": [-10,10,60,90]}. Region definitions should not cross grid boundaries and should use grid units.

timescale-all: (string) Timescale for all datasets. Required for histogram metrics. Optional for other results. Will be used in legends, titles, and filenames if provided. This variable is only used in a descriptive capacity by ASoP-Spectral. Do not set if input datasets have different timescales.

dates-all: (optional, string) Dates for file names. Must be set to have dates appear in histogram metric plot title. Will be used in other legends, titles, and filenames if provided. Do not set if input datasets have different date ranges. This setting is purely for labeling purposes.

season-all: (optional, string) Season for file names. Must be set to have season appear in histogram metric plot title. Will be used in other legends, titles, and filenames if provided. Do not set if input datasets use different seasons. This setting is purely for labeling purposes.

mask: (string) Full path to mask file, empty string, or null, for histogram metric plot. Setting this key to a valid file name tells ASoP-Spectral to run the histogram metrics. This key is required for the histogram metrics, but do not set this key if you do not want to produce the histogram metrics. Do not place the mask file in the model or obs input folder.

Dataset-specific keys

name: (optional, string): (optional, string) A short, descriptive, and unique name for this dataset. Will be used in legends, titles, and filenames if provided. By default, ASoP spectral will search the file metadata for the keys "source_id", "short_name", "name", "source", and "model" to use as a name. It will also add the "variant_label" value if present. If no name is found in the settings or metadata, the file will be called "no_name".

timescale: (optional, string) Timescale for single dataset. Will be used in legends, titles, and filenames if provided. If running histogram metrics, the global setting "timescale-all" must be set instead. The dataset-specific timescale will not be used for that figure title. This setting is purely for labeling purposes.

dates: (optional, string) Dates for file names. Will be used in legends, titles, and filenames if provided. If running histogram metrics, the global setting "dates-all" must be set instead. The dataset-specific dates will not be used for that figure title. This setting is purely for labeling purposes.

season: (optional, string) Season for file names. Will be used in legends, titles, and filenames if provided. If running histogram metrics, the global setting "season-all" must be set. The dataset-specific seasons will not be used for that figure title. This setting is purely for labeling purposes.