

METplus Grid to Point Overview

Minna Win

Background

The Grid to Point (also referred to Grid to Obs) use case embodies the steps needed to obtain statistics via the MET point-stat tool for forecasts at observation points (vs a gridded analysis). Point observation data can be in prepBUFR format or netCDF. If the data is in prepBUFR format, it will first need to be converted to netCDF for the MET point_stat tool to read.

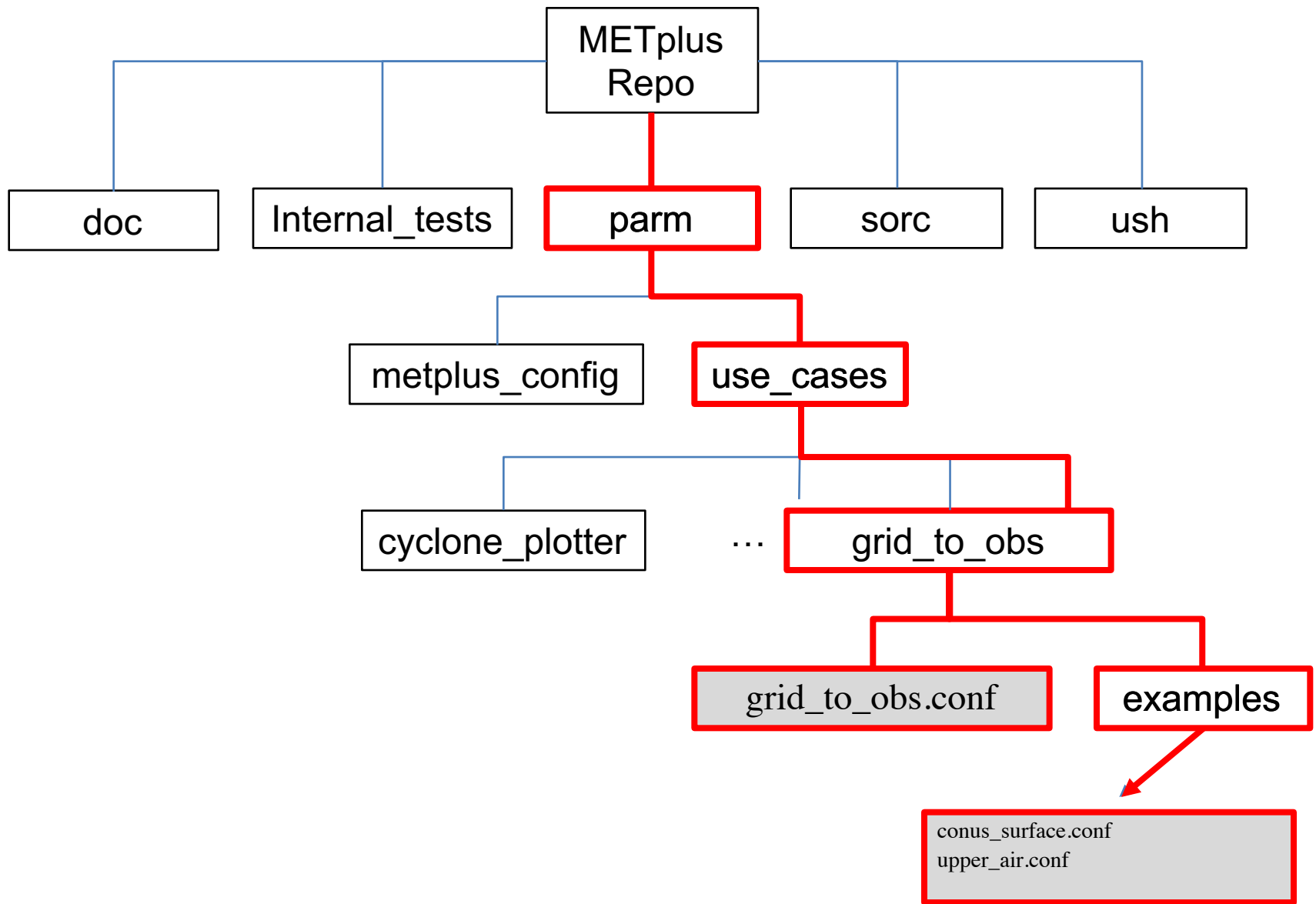
Tools

MET pb2nc

- Wrapped by pb2nc_wrapper.py, under METplus/ush
- Converts prepBufr files into netCDF

MET point-stat

- Wrapped by point_stat_wrapper.py, under METplus/ush
- Provides verification statistics for forecasts at observation points



Configuring METplus for Use Case

Sanity Check-

Verify that you have set up your PYTHONPATH and PATH !

PYTHONPATH:

csh:

```
setenv PYTHONPATH full-path-to-METplus/ush:$PYTHONPATH  
setenv PATH ~/METplus/ush:$PATH
```

Bash:

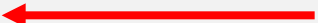

```
export PYTHONPATH=full-path-to-METplus/ush/$PYTHONPATH  
export PATH="~/METplus/ush:$PATH"
```

Configuring METplus for Use Case

Create your own conf file(s)* and set necessary variables:

- a. This can be located anywhere you have read/write access
- b. View the `grid_to_obs.conf` file (in GitHub web interface or an editor) and look for all variables set to `/path/to`.
- c. Replicate the section header where variables set to `/path/to`, replacing `/path/to` with your directory paths

*Keep `pb2nc` and `point-stat` variables in one config file, or separate them into their own

```
[dir]  Make note of the section header/family  
## Input data directories  
## This is the location of your input files for METplus  
  
PROJ_DIR = /path/to  
TMP_DIR = /path/to  
OUTPUT_BASE = /path/to  replace all /path/to's with actual paths  
METPLUS_BASE = /path/to  
MET_INSTALL_DIR = /path/to  
PARAM_BASE = /path/to
```

Configuring METplus for Use Case

In your last "custom" configuration file, under the [config] section, set the PROCESS_LIST:

Scenario 1: Point observation data is in prepBUFR format, require conversion to netCDF

```
PROCESS_LIST = Pb2NC, PointStat
```

Scenario 2: Point observation data is in netCDF format:

```
PROCESS_LIST = PointStat
```

Configuring METplus for Use Case

NOTE: Not sure about the spelling of the wrappers, or which wrappers are available?

1. in your custom config file
`PROCESS_LIST = Usage`

2. Run METplus*, on command line:

```
master_metplus.parm -c path_to_your/custom_conf_file.conf
```

*Don't forget to replace all */path/to's*

USAGE:

This is a default process, please indicate more specific processes in the PROCESS_LIST variable in one or more of the following configuration files:

- parm/metplus_config/metplus_runtime.conf
- parm/metplus_use_cases/<usecase_name>/<usecase_name>.conf
- parm/metplus_use_cases/<usecase_name>/examples/<example_name>.conf

Currently available processes are:

- TcPairs
- ExtractTiles
- SeriesByInit
- SeriesByLead
- PcpCombine
- RegridDataPlane
- GridStat
- Mode
- RegridDataPlane
- CyclonePlotter
- TCMPRPlotter
- PB2NC
- PointStat

Configuring METplus for Use Case

Question: What do I need to do if I want to use my own data?

Specify the appropriate filename variable under [filename templates] or regular expression [regex_pattern] for the filename or directory:

[filename_templates]

To define the format of the input filename

[regex_pattern]

To define the format of output filenames, input directory names, etc.

```
#  
## FILENAME TEMPLATES  
#  
[filename_templates]  
## Output file template  
#  
# PB2NC  
#  
# For NAM (conus surface)  
#NC_FILE_TMPL = prepbufr.nam.{init?fmt=%Y%m%d}.t{cycle?fmt=%HH}z.tm{offset?fmt=%HH}.nc  
  
# For GDAS (upper air)  
NC_FILE_TMPL = prepbufr.gdas.{valid?fmt=%Y%m%d%H}.nc
```

Configuring METplus for Use Case

Question: What do I need to do if I want to use my own data, cont'd

Update the time window of interest, to reflect your data:

```
[config] ← Notice the family/section header
# set looping method to processes-each 'task' in the process list runs to
# completion (for all init times) before the next 'task' is run
LOOP_METHOD = processes

# Configuration files
TC_PAIRS_CONFIG_FILE =
{PARM_BASE}/use_cases/track_and_intensity/met_config/TCPairsETCConfig

# 'Tasks' to be run
PROCESS_LIST = TcPairs, TCMRPlotter

# The init time begin and end times, increment, and last init hour.
INIT_BEG = 20141201
INIT_END = 20141231 } time window of interest
```

Configuring METplus for Use Case

Question: What do I need to do if I want to use my own data?

Specify the appropriate filename variable under [filename templates] or regular expression [regex_pattern] for the filename or directory:

[filename_templates]

To define the format of the input filename

[regex_pattern]

To define the format of output filenames, input directory names, etc.

```
#  
## FILENAME TEMPLATES  
#  
[filename_templates]  
## Output file template  
#  
# PB2NC  
#  
# For NAM (conus surface)  
#NC_FILE_TMPL = prepbufr.nam.{init?fmt=%Y%m%d}.t{cycle?fmt=%HH}z.tm{offset?fmt=%HH}.nc  
  
# For GDAS (upper air)  
NC_FILE_TMPL = prepbufr.gdas.{valid?fmt=%Y%m%d%H}.nc
```

Configuring METplus for Use Case

Example 1: How to set up a variable in [file_templates]:

prepbufr.gdas.2017061618.nc is described by:



```
NC_FILE_TMPL = prepbufr.gdas.{valid?fmt=%Y%m%d%H}.nc
```

The parts of the file that will vary from file to file is contained within {}'s

Example 2: A little more than one piece of time information:

pgbf219.gfs.201706110 is described by:



```
FCST_INPUT_FILE_TMPL = pgbf{lead?fmt=%H}.gfs.{valid?fmt=%Y%m%d%H}
```

Configuring METplus for Use Case

Example 3: Tropical cyclone data with init time, cycle time, and offset time:

prepbufnr.nam.20170707.t00z.tm03.nc is described by:

NC_FILE_TMPL =

prepbufnr.nam.{init?fmt=%Y%m%d}.t{cycle?fmt=%HH}z.tm{offset?fmt=%HH}.nc



Configuring METplus for Use Case

- Variables that are mapped to the MET config files, look at the following files in <https://github.com/NCAR/METplus>:

parm/use_cases/grid_to_obs/grid_to_obs.conf

parm/use_cases/grid_to_obs/examples/conus_surface.conf

parm/use_cases/grid_to_obs/examples/upper_air.conf

- copy variables that are set to /path/to into your own config file, keep headers in your conf file* consistent with the METplus conf file:

PROJ_DIR

TMP_DIR

OUTPUT_BASE

METPLUS_BASE

MET_INSTALL_DIR

PARAM_BASE

PREPBUFR_DATA_DIR - pb2nc input data

FCST_INPUT_DIR – point stat input data for forecast

OBS_INPUT_DIR – point stat input data for obs

** Remember to use the Glossary in the User Documentation to assist you*

Running the use case with METplus

Run `master_metplus.py`, from command line

```
master_metplus.py \  
parm/use_cases/grid_to_obs.grid_to_obs.conf \  
-c parm/use_cases/grid_to_obs/examples/conus_surface.conf \  
-c path_to_your/custom_config_file.conf
```

The `metplus_final.conf` file, log files, intermediate files, and final output files are created in the output directory that were specified in your last custom config file

Post-run overview: logging

Logfile located in your output directory under the 'logs' directory
Format is based on the timestamp template you indicated in your
metplus_logging.conf file:

```
Building MET pb2nc command... 2018-03-21 23:13:13,943 :  
INFO|:extract_prepbufr_file_info|/home/minnawin/latest/METplus/ush/pb2nc_wrapper.py| Creating  
prepbufr file information 2018-03-21 23:13:13,943 :  
DEBUG|:generate_output_nc_filename|/home/minnawin/latest/METplus/ush/pb2nc_wrapper.py  
Generating output NetCDF file name... 2018-03-21 23:13:13,943 :  
DEBUG|:build_pb2nc_command|/home/minnawin/latest/METplus/ush/pb2nc_wrapper.py|pb2nc  
called with: /usr/local/met-6.1/bin/pb2nc /d1/METplus_Mallory/data/prepbufr/gdas/prepbufr  
.gdas.2017060200 /d1/minnawin/pb2nc_crow_test/gdas/upper_air/prepbufr.gdas.2017060200.nc -v  
5 -log /tmp/pb2nc_test.log /home/minnawin/wip/METplus/parm/met_config/PB2NCConfig_upper_air  
2018-03-21 23:13:13,943 : RUNNING: /usr/local/met-6.1/bin/pb2nc  
/d1/METplus_Mallory/data/prepbufr/gdas/prepbufr.gdas.2017060200  
/d1/minnawin/pb2nc_crow_test/gdas/upper_air/prepbufr.gdas.2017060200.n c -v 5 -log  
/tmp/pb2nc_test.log /home/minnawin/wip/METplus/parm/met_config/PB2NCConfig_upper_air
```


Post-run overview: output

Scenario 1: point obs data is in prepBUFR

pb2nc_wrapper is run, followed by point_stat_wrapper:

pb2nc directory with netCDF data created:

in directory <output_base><model_name>/<vertical_location>

output_base, model_name and vertical_location are defined in the use case config file

point_stat text output (.stat files):

In directory <output_base>/<model_name>

Output_base and model_name are defined in the point stat section of the use case file

Post-run overview: output

Scenario 2: point obs data is netCDF

1. point_stat_wrapper is run
2. point_stat text output (.stat files) are created:
In directory <output_base>/<model_name>

Recall:

You set your input data in your last custom config file

The Output_BASE and MODEL_NAME are defined in the point stat section of the use case conf file

Questions