Grid to Obs Use Case

Purpose:

Performs grid-to-obs verification using the MET tools. First, pb2nc is applied to prepBUFR files (observation files) to stratify and convert those files into NetCDF files that MET can recognize. Then, using the pb2nc output as input, along with gridded forecast files (model data) to the MET point_stat tool to calculate statistics at observation points.

Overview:

The configuration files provided in the METplus/parm/use_cases_grid_to_obs directory allow you to perform a grid-to-obs verification using MET tools for two specific examples: upper air and conus surface. The configurations can be run after the user replaces */path/to* with the full path to the directories and files in her/his running environment. The user can customize/experiment by modifying the dates, message types, variables, etc. which are highlighted as red text (illustrated below in the configuration file description).

The grid_to_obs.conf file is the 'default' configuration file for performing a grid_to_obs verification on for conus surface data. For an example of performing a grid-to-obs verification for upper air data, refer to the upper_air.conf configuration file in the METplus/parm/use_cases/grid_to_obs/examples directory. The conus_sfc.conf file in the METplus/parm/use_cases/grid_to_obs/examples directory is identical to the grid_to_obs.conf file.

To run the conus surface grid_to_obs verification, run the following:
master_metplus.py -c
<your/path/to/METplus/code>/METplus/parm/use_cases/grid_to_obs/grid_to_obs.conf

To run the upper air grid_to_obs verification, run the following:
master_metplus.py -c
<your/path/to/METplus/code>/METplus/parm/use_cases/grid_to_obs/examples/upper_air.conf

You may also override any of these setting by creating your own configuration file, and call this last. Following the example for running the upper air grid_to_obs verification, one can simply leave the original settings in the upper_air.conf file and define custom begin and end times in a custom configuration file*:
master_metplus.py -c
<your/path/to/METplus/code>/METplus/parm/use_cases/grid_to_obs/examples/upper_air.conf
<your/full/path/to/your/custom/config>/my_custom_overriding_upper_air.conf

**Make sure your key-value settings are under the same section header that correspond to the section where you are overriding values:*
[config]
START_HOUR - 0
END_HOUR - 18
BEG_TIME – 20170605
END_TIME = 20170615
INTERVAL_TIME=12

Configuration file description:

For the directory section, indicated by [dir], replace any '/path/to' text with the full path for that associated value:

[dir]
PROJ_DIR - where you want your project files to reside
TMP_DIR - where you want your temporary files to reside
OUTPUT_BASE - the base directory for all output files
METPLUS_BASE - the base directory where MET+ is installed
MET_INSTALL_DIR - the location of the MET installation
PARM_BASE - the base directory where all parameter files are located (MET and MET+ param files)

For pb2nc:
PREPBUFR_DATA_DIR – directory where prepBUFR files reside

For point_stat:
FCST_INPUT_DIR - the directory where your model data resides
OBS_INPUT_DIR - the directory where your pb2nc output files (in NetCDF) are located
POINT_STAT_OUTPUT_DIR – location for all point_stat output files to reside, based on the OUTPUT_BASE value:
{OUTPUT_BASE}/path/to

For the configuration section, indicated by [config], set the following values to meet your specifications:

[config]
LOOP_METHOD – 'processes' is currently supported, 'time' to be implemented later.

LOG_LEVEL -
DEBUG is the most verbose and is valuable for debugging (provides all error, warning, and info messages in addition to debug messages)

INFO is the next least verbose (provides all error, warning and info messages)

WARN is the next least verbose (provides only error and warning messages)

ERROR is the least verbose (provides only error messages)

OVERWRITE_NC_OUTPUT – yes or no; overwrite the pb2nc output if it already exists in the specified output directory

TIME_METHOD – BY_VALID or BY_INIT selects times based on either valid times or initialization times. For this use case, BY_VALID is used.

START_HOUR - The beginning hour defining your time window of interest
END_HOUR - The ending hour defining hour time window of interest
BEG_TIME – The year, month, day (YYYYMMDD) defining the start of your time window of interest

END_TIME – The year, month, day (YYYYMMDD) defining the end of your time window of interest.
INTERVAL_TIME – The interval time in hours/time step, used by the pb2nc wrapper.

VERTICAL_LOCATION – upper_air for the upper air example, or conus_sfc for the conus surface example

For pb2nc:
PB2NC_CONFIG_FILE - the location of your MET pb2nc config file, this is relative to the PARM_BASE you defined in the directory section of the MET+ grid_to_obs.conf or one of the example configuration files (conus_sfc.conf, upper_air.conf).
PB2NC_GRID - corresponds to *grid* in the verification masking regions in the MET pb2nc config file.
PB2NC_POLY – corresponds to *poly* in the verification masking regions in the MET pb2nc config file. If you are defining a long list of poly masking files, you may need to define these in the MET pb2nc config file, as there is a limit to the string length of environment variables.
PB2NC_STATION_ID – corresponds to the prepBufr station id, *station_id* in the MET pb2nc config file.
PB2NC_MESSAGE_TYPE – corresponds to the prepBufr message type, *message_type* in the MET pb2nc config file.

TIME_SUMMARY_FLAG – set to False to skip performing time summary. There is a bug in met-6.1 that will produce undesirable results if set to True.

For point_stat:
POINT_STAT_CONFIG_FILE – the location of your MET point_stat config file, this is relative to the PARM_BASE you defined in the directory section of the MET+ grid_to_obs.conf or one of the example configuration files (conus_sfc.conf, upper_air.conf).

FCST_HR_START – the starting hour of the forecast hour of interest.
FCST_HR_END - the ending hour of the forecast hour of interest.
FCST_HR_INTERVAL – the step size/interval in hours used to define all the forecast hours of interest.

MODEL_NAME - the name of the model/fcst data (e.g. GFS)
OBS_NAME - the name of the observation data (e.g. NAM, GDAS)

REGRID_TO_GRID – the verification grid. Corresponds to *to_grid* in the regrid dictionary of the MET point_stat config file.

POINT_STAT_GRID – the verification masking region, corresponds to *grid* in the mask dictionary of the MET point_stat config file.
POINT_STAT_POLY – the verification masking region, corresponds to *poly* in the mask dictionary of the MET point_stat config file.
POINT_STAT_STATION_ID – corresponds to *sid* in the mask dictionary used for the verification

masking regions.

POINT_STAT_MESSAGE_TYPE - corresponds to *message_type* in the point observation filtering options in the MET point_stat config file.

The following represent the field names, levels, and options in the fcst dictionary where forecast and observation fields to be verified are located in the MET point_stat config file. Options can be cat_thresholds or GRIB_lvl_val1 = ###, where ### represents a numerical grib level value.
FCST_VAR1_NAME – variable name for the first variable
FCST_VAR1_LEVELS – a list of levels for the first variable
FCST_VAR1_OPTIONS – optional, options such as cat_thresh or GRIB_lvl_val1, or unit conversion definitions
…
FCST_VARn_NAME – variable name for the nth variable
FCST_VARn_LEVELS – variable name for the nth variable
FCST_VARn_OPTIONS – optional. Options for the nth variable

[filename_templates]
NC_FILE_TMPL – the output NetCDF filename template from the output generated from running pb2nc.

For the configuration section, indicated by [regex_pattern], set the following values to meet your specifications:
[regex_pattern]
PREPBUFR_DIR_REGEX – the regular expression describing the directory where prepBUFR input files reside.
PREPBUFR_FILE_REGEX – the regular expression describing the format of the prepBUFR input files.