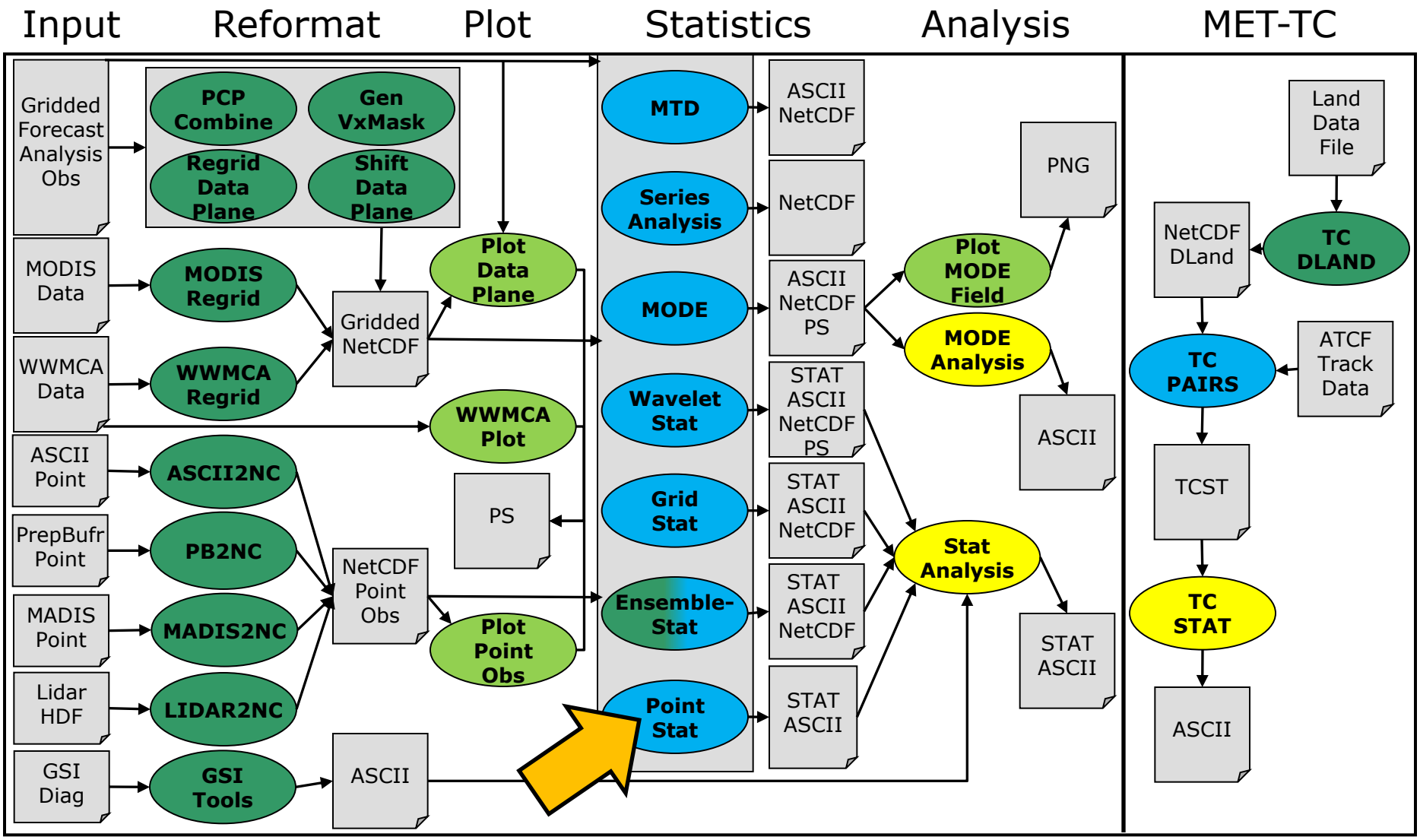
MET Point-Stat Tool

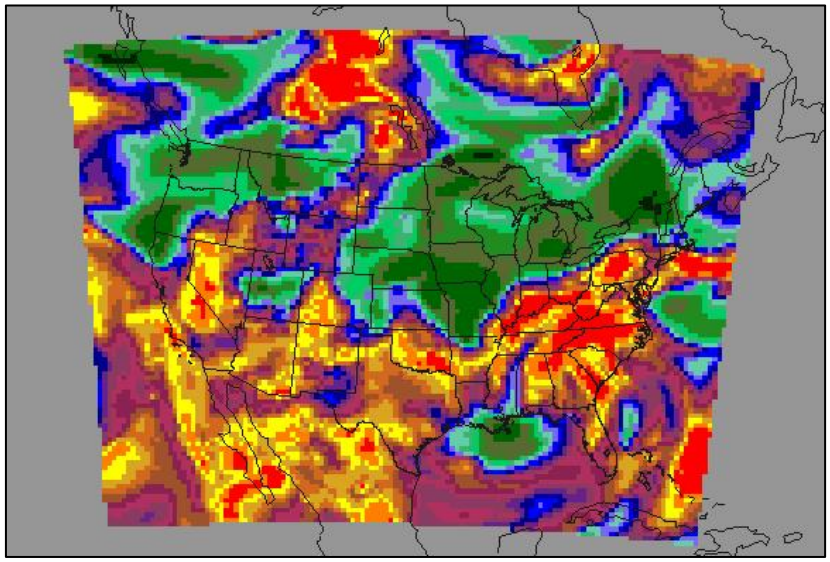
John Halley Gotway

METplus Tutorial
July 31 – August 2, 2019
NRL – Monterey, CA

Point-Stat Tool

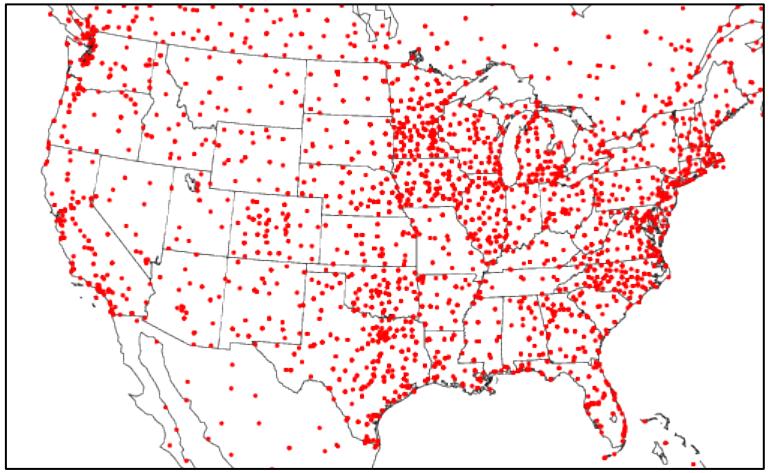


Point-Stat: Overview



- Compare **gridded forecasts** to **point observations**.
- Accumulate matched pairs over a defined area at a single point in time.
- Verify one or more variables/levels.
- Analysis tool provided to aggregate through time.

- Verification methods:
  - Continuous statistics for raw fields.
  - Single and Multi-Category counts and statistics for thresholded fields.
  - Parametric and non-parametric confidence intervals for statistics.
  - Compute partial sums for raw fields and/or the raw matched pair values.
  - Methods for probabilistic forecasts.
  - HiRA spatial verification method.



Point-Stat: Usage

Usage: point_stat

fcst_file

obs_file

config_file

[-point_obs netcdf_file]

[-obs_valid_beg time]

[-obs_valid_end time]

[-outdir path]

[-log file]

[-v level]

fcst_file	Gridded forecast file
obs_file	NC point observation file
config_file	ASCII configuration file
-point_obs	Additional NC point observation files
-obs_valid_beg	Beginning of valid time window for matching
-obs_valid_end	End of valid time window for matching
-outdir	Output directory to be used
-log	Optional log file
-v	Level of logging

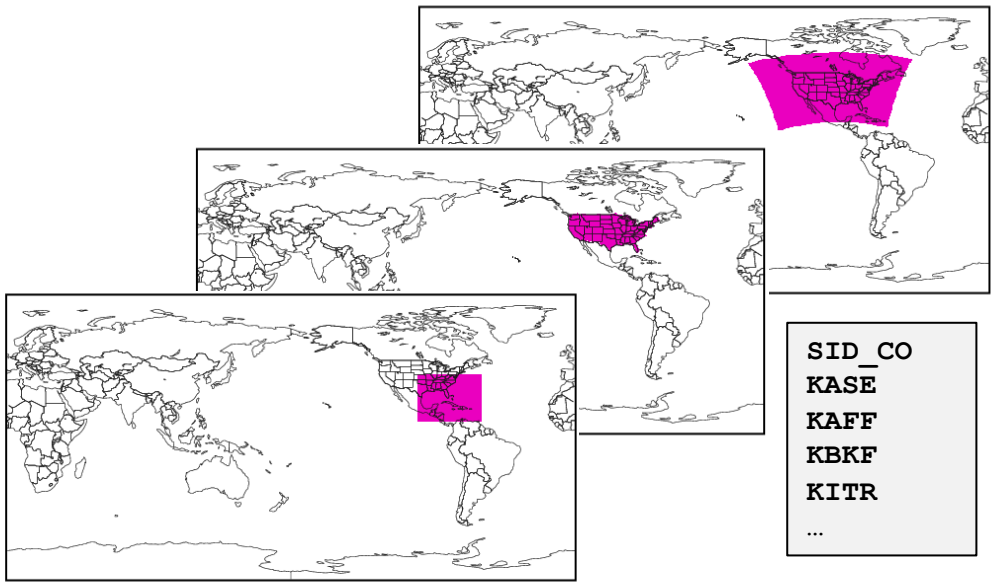
Point-Stat: Input/Output

- Input Files
  - Gridded forecast file
    - GRIB1 output of Unified Post-Processor (or other)
    - GRIB2 from NCEP (or other)
    - NetCDF from PCP-Combine, wrf_interp, or CF-compliant
    - Python Interface
  - Point observation file
    - NetCDF output of PB2NC, ASCII2NC, MADIS2NC, or LIDAR2NC
  - ASCII configuration file
- Output Files
  - ASCII statistics file with all output lines (end with ".stat")
  - Optional ASCII files sorted by line type with a header row (ends with "_TYPE.txt")

Point-Stat: Mask Options

- The **mask** dictionary defines regions over which to compute statistics.
- Define *grid points* to be used:
  - Use **grid** for pre-defined, named grids (**FULL**).
  - Use **poly** for lat/lon polylines, Gen-Vx-Mask output, data masks.
- Define *observation points* to be used:
  - Use **sid** for individual stations or define groups in an ASCII file.
  - Use **llpnt** to threshold obs lat/lon directly.

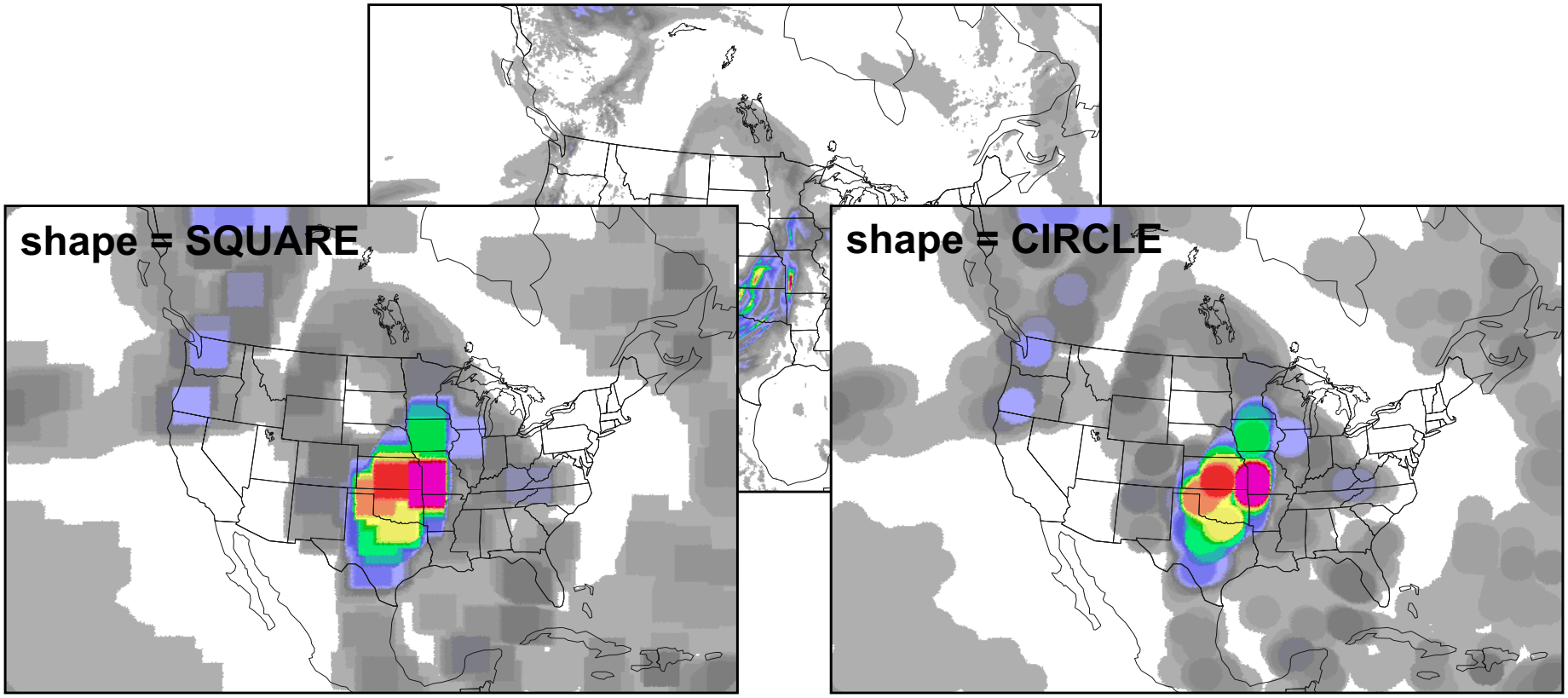
```
mask = {
 grid  = [ "G212" ];
 poly  = [ "CONUS.poly" ];
 sid   = [ "KNXF",
           "SID_CO.txt"];
  llpnt = [ {
   name = "LATLONBOX";
   lat_thresh = >=10&&<=40;
   lon_thresh = >=-100&&<=-60;
  } ];
};
```



```
SID_CO
KASE
KAFF
KBKF
KITR
…
```

Interpolation Shape

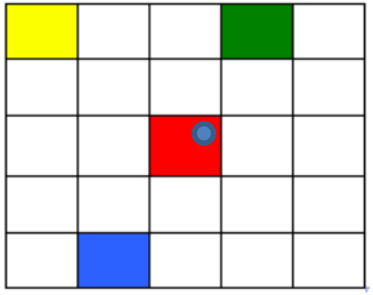
- Support for square or circular interpolation, regridding, and data smoothing areas.
- Smooth data by computing the **MAX** of **WIDTH** of 30.



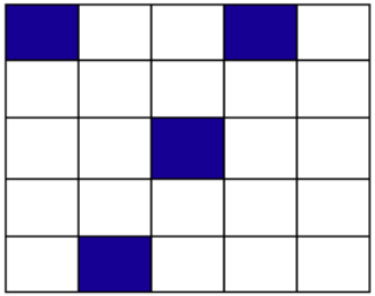
Point-Stat: HiRA Framework

- High Resolution Assessment (HiRA) verification logic is applied to deterministic forecasts matched to point observations (Mittermaier, 2014)
- Process neighborhood values as an ensemble forecast (**ECNT** line type).
- Threshold, compute fractional coverage, and verify as a probability forecast (**PCT**, **PSTD**, **PRC**, and **PJC** line types).
- Allows for some spatial / temporal uncertainty by giving credit for being 'close'.
- Allows for comparison of deterministic and ensemble forecasts via the same set of probabilistic statistics.
- Also allows for comparison of models with different grid resolutions via adjustment of neighborhood size.

Model Forecast
White boxes = 0
Colored boxes > 0



Threshold Forecast
Blue boxes = event
cat_thresh = [>0];



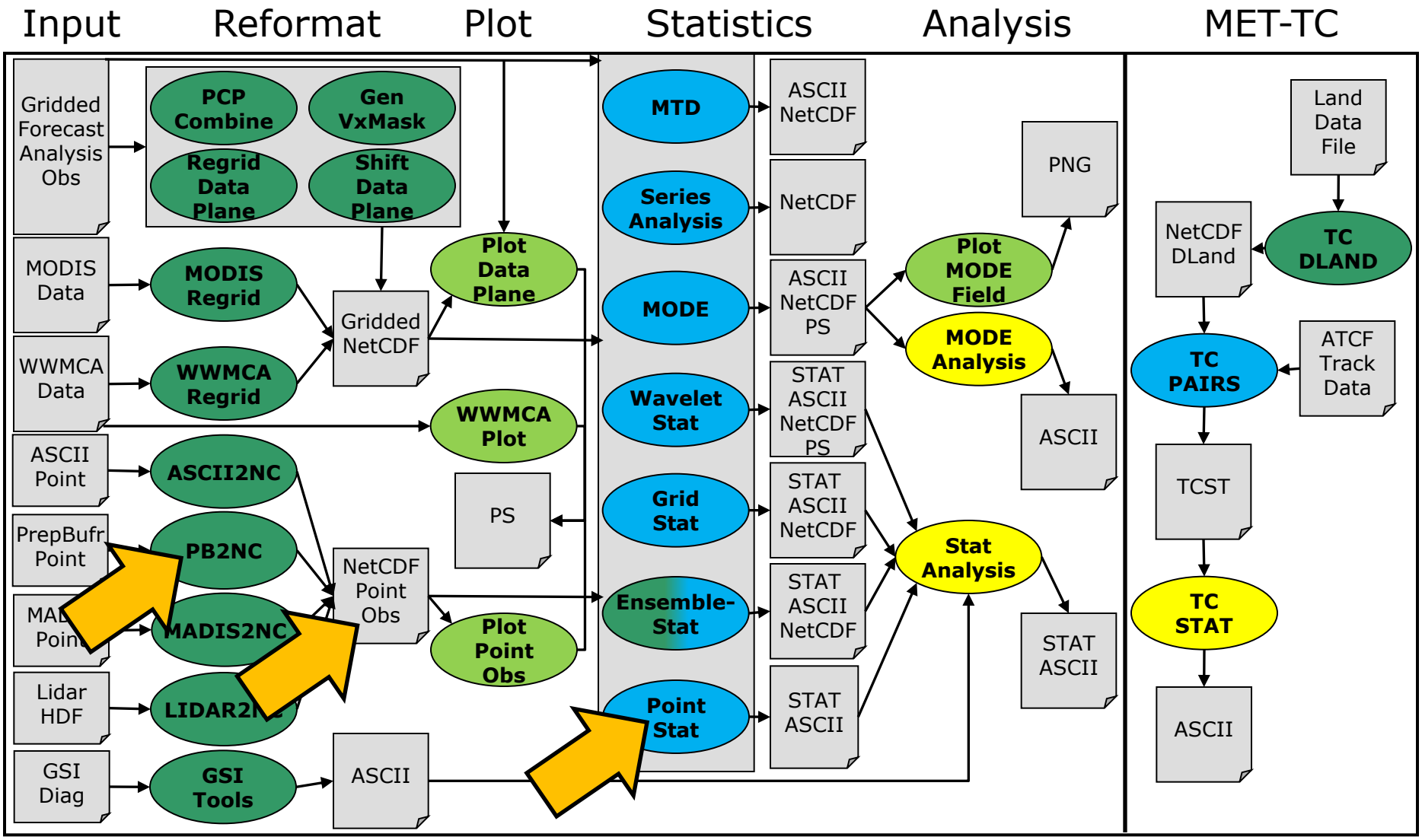
HiRA Ensemble:
- Write ECNT using neighborhood ensemble.

HiRA Probabilities:
- 1x1 NBRHD = 1/1
- 3x3 NBRHD = 1/9
- 5x5 NBRHD = 4/25

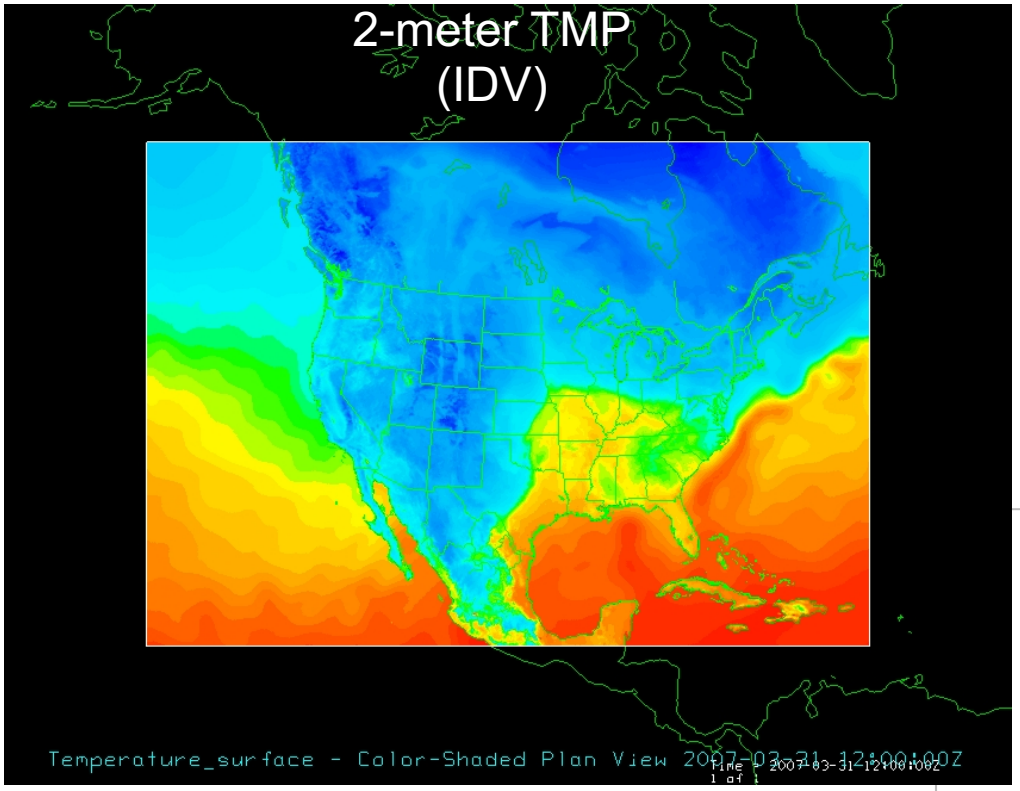
```
hira = {
  // Enable or disable
  flag        = TRUE;
  // Neighborhood sizes
  width       = [ 2, 3, 4, 5 ];
  // Probability thresholds
  cov_thresh  = [ ==0.25 ];
  vld_thresh  = 1.0;
  // Neighborhood shape
  shape       = SQUARE;
};
```

```
output_flag = {
  …
  pct     = STAT;
  pstd    = STAT;
  pjc     = STAT;
  prc     = STAT;
  ecnt    = STAT;
  …
};
```

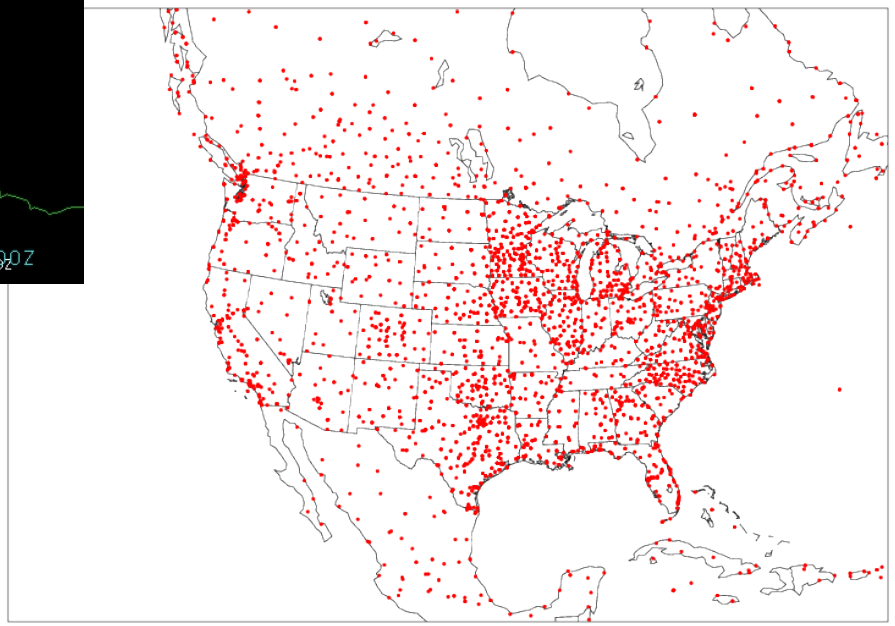
Example Data Flow



Point-Stat: Example



4003 TMP ADPSFC Obs
(plot_point_obs)



Point-Stat: Configuration

- Many configurable parameters – only set a few:
  - 2-meter temperature.
  - Threshold temperatures near freezing.
  - Match to obs at the surface.
  - Accumulate stats over all the points in the domain.
  - Match observation to the nearest forecast value.
  - Generate all output line types other than vector and probabilistic.

```
fcst = {
  message_type = [ "ADPSFC"];
  field = [
   {
    name       = "TMP";
    level      = [ "Z2" ];
    cat_thresh = [ >273.0, >283.0, >293.0 ];
   }
 ];
};
obs = fcst;
```

```
mask = {
 grid  = [ "FULL" ];
 poly  = [];
 sid   = [];
 llpnt = [];
};
```

```
interp = {
 vld_thresh = 1.0;
 type = [
  {
   method = NEAREST;
   width  = 1;
  }
 ];
};
```

```
output_flag = {
 fho    = BOTH;
 ctc    = BOTH;
 cts    = BOTH;
 mctc   = BOTH;
 mcts   = BOTH;
 cnt    = BOTH;
 sl1l2  = BOTH;
 sal1l2 = BOTH;
 vl1l2  = NONE;
 val1l2 = NONE;
 vcnt   = NONE;
 pct    = NONE;
 pstd   = NONE;
 pjc    = NONE;
 prc    = NONE;
 ecnt   = NONE;
 eclv   = NONE;
 mpr    = BOTH;
};
```

Point-Stat: Run

point_stat \

sample_fcst.grb sample_pb.nc \

PointStatConfig_TMPZ2 -outdir out -v 2

```
DEBUG 1: Default Config File: met-X.Y/share/met/data/config/PointStatConfig_default
DEBUG 1: User Config File: PointStatConfig_TMPZ2
DEBUG 1: Forecast File: sample_fcst.grb
DEBUG 1: Climatology File: none
DEBUG 1: Observation File: sample_pb.nc
DEBUG 2: ------------------------------------------------------------------------------
DEBUG 2: Reading data for TMP/Z2.
DEBUG 2: For TMP/Z2 found 1 forecast levels and 0 climatology levels.
DEBUG 2: ------------------------------------------------------------------------------
DEBUG 2: Searching 87752 observations from 9396 messages.
DEBUG 2: ------------------------------------------------------------------------------
DEBUG 2: Processing TMP/Z2 versus TMP/Z2, for observation type ADPSFC, over region FULL, for interpolation method
NEAREST(1), using 4003 pairs.
DEBUG 2: Computing Categorical Statistics.
DEBUG 2: Computing Multi-Category Statistics.
DEBUG 2: Computing Continuous Statistics.
DEBUG 2: Computing Scalar Partial Sums.
DEBUG 2: ------------------------------------------------------------------------------
DEBUG 1: Output file: out/point_stat_360000L_20070331_120000V.stat
DEBUG 1: Output file: out/point_stat_360000L_20070331_120000V_fho.txt
DEBUG 1: Output file: out/point_stat_360000L_20070331_120000V_ctc.txt
DEBUG 1: Output file: out/point_stat_360000L_20070331_120000V_cts.txt
DEBUG 1: Output file: out/point_stat_360000L_20070331_120000V_mctc.txt
DEBUG 1: Output file: out/point_stat_360000L_20070331_120000V_mcts.txt
DEBUG 1: Output file: out/point_stat_360000L_20070331_120000V_cnt.txt
DEBUG 1: Output file: out/point_stat_360000L_20070331_120000V_sl1l2.txt
DEBUG 1: Output file: out/point_stat_360000L_20070331_120000V_sal1l2.txt
DEBUG 1: Output file: out/point_stat_360000L_20070331_120000V_mpr.txt
```

Point-Stat: ASCII Output

- Categorical – Single Threshold
  - Contingency table counts and stats (FHO, CTC, CTS, ECLV)
- Categorical – Multiple Thresholds
  - NxN Contingency table counts and stats (MCTC, MCTS)
- Scalars - raw fields
  - Continuous statistics (CNT) and partial sums (SL1L2, SAL1L2)
- Wind Vectors
  - Vector statistics (VCNT) and partial sums (VL1L2, VAL1L2)
- Probabilistic and HiRA
  - Nx2 Contingency table counts and stats (PCT, PSTD)
  - Continuous statistics and ROC curve (PJC, PRC)
  - Economic Cost/Loss value (ECLV)
  - Continuous ensemble statistics for HiRA (ECNT)
- Matched pairs
  - Raw matched pairs – a lot of data! (MPR)

24 common header columns + Line type specific columns

Point-Stat: Sample Output

1. STAT file output for sample run:
   - 1 line each for CNT, SL1L2 , MCTC, MCTS
   - 3 lines each for FHO, CTC, CTS
   - 4,003 lines for MPR!
2. Additional TXT files for each line type

```
Output file: out/point_stat_360000L_20070331_120000V.stat
Output file: out/point_stat_360000L_20070331_120000V_fho.txt
Output file: out/point_stat_360000L_20070331_120000V_ctc.txt
Output file: out/point_stat_360000L_20070331_120000V_cts.txt
Output file: out/point_stat_360000L_20070331_120000V_mctc.txt
Output file: out/point_stat_360000L_20070331_120000V_mcts.txt
Output file: out/point_stat_360000L_20070331_120000V_cnt.txt
Output file: out/point_stat_360000L_20070331_120000V_sl1l2.txt
Output file: out/point_stat_360000L_20070331_120000V_sal1l2.txt
Output file: out/point_stat_360000L_20070331_120000V_mpr.txt
```

Point-Stat: CTC Output Line

VERSION	VX.Y
MODEL	WRF
DESC	NA
FCST_LEAD	360000
FCST_VALID_BEG	20070331_120000
FCST_VALID_END	20070331_120000
OBS_LEAD	000000
OBS_VALID_BEG	20070331_103000
OBS_VALID_END	20070331_133000
FCST_VAR	TMP
FCST_UNITS	K
FCST_LEV	Z2
OBS_VAR	TMP
OBS_UNITS	K
OBS_LEV	Z2

OBTYPE	ADPSFC
VX_MASK	FULL
INTERP_MTHD	UW_MEAN
INTERP_PNTS	1
FCST_THRESH	>273.000
OBS_THRESH	>273.000
COV_THRESH	NA
ALPHA	NA
LINE_TYPE	CTC
TOTAL	4003
FY_OY (hits)	3111
FY_ON (f.a.)	78
FN_OY (miss)	215
FN_ON (c.n.)	599

Point-Stat: Matched Pairs

- Matched Pair (MPR) line type contains 1 line for each matched pair.
  - Data overload!

TOTAL	INDEX	OBS_SID	OBS_LAT	OBS_LON	OBS_LVL	OBS_ELV	FCST	OBS	OBS_QC	...
4003	1	71600	43.93000	-60.01000	1010.79999	4.01053	272.00000	271.95001	NA	
4003	2	71616	46.43000	-71.93000	1016.09998	102.04903	268.00000	269.64999	NA	
4003	3	71629	44.23000	-78.36000	1004.50000	191.44466	273.00000	272.64999	NA	
4003	4	71028	51.67000	-124.40000	916.50000	872.82202	264.00000	265.25000	NA	
4003	5	71066	58.61000	-117.16000	973.90002	337.50449	272.00000	271.45001	NA	
4003	6	71104	52.18000	-122.04000	906.50000	938.08594	271.00000	264.64999	NA	
4003	7	71109	50.68000	-127.36000	1020.20001	22.03931	275.00000	275.64999	NA	
4003	8	71150	50.45000	-100.59000	949.09998	562.38477	272.00000	271.75000	NA	
4003	9	71177	57.13000	-61.47000	899.70001	834.87476	259.00000	254.64999	NA	
4003	10	71197	47.56000	-59.16000	1000.90002	40.06803	272.00000	269.95001	NA	
4003	11	71378	47.41000	-72.79000	1006.90002	169.37592	267.00000	266.14999	NA	
4003	12	71415	45.76000	-62.68000	1014.00000	1.99518	269.00000	269.04999	NA	
4003	13	71425	49.24000	-65.33000	1014.90002	28.96468	264.00000	267.25000	NA	
4003	14	71437	43.29000	-79.79000	1017.79999	77.03765	274.00000	276.35001	NA	
4003	15	71473	48.78000	-123.04000	1015.70001	23.93772	278.00000	280.95001	NA	
4003	16	71486	52.93000	-118.31000	896.20001	1021.55963	265.00000	270.35001	NA	
4003	17	71573	42.87000	-80.55000	996.09998	231.62808	273.00000	276.54999	NA	
4003	18	71579	49.89000	-97.13000	989.70001	230.44473	276.00000	274.35001	NA	
4003	19	71598	47.78000	-64.83000	1016.70001	5.00338	266.00000	267.54999	NA	
4003	20	71860	50.02000	-100.32000	874.20001	474.10379	273.00000	273.54999	NA	

Point-Stat: No Matched Pairs?

- Several met-help questions ask…
  "*Why did I get 0 matched pairs?*"

```
DEBUG 2: Processing UGRD/P900-750 versus UGRD/P900-750, for observation
         type ADPUPA, over region CONUS, for interpolation method
         NEAREST(1), using 0 pairs
```

- Ask Point-Stat for more details by rerunning at a higher verbosity level (**-v 3**).

```
DEBUG 3: Number of matched pairs  = 0
DEBUG 3: Observations processed   = 89893
DEBUG 3: Rejected: SID exclusion  = 0
DEBUG 3: Rejected: obs type       = 77294
DEBUG 3: Rejected: valid time     = 0
DEBUG 3: Rejected: bad obs value  = 0
DEBUG 3: Rejected: off the grid   = 6
DEBUG 3: Rejected: level mismatch = 8077
DEBUG 3: Rejected: quality marker = 0
DEBUG 3: Rejected: message type   = 4516
DEBUG 3: Rejected: masking region = 0
DEBUG 3: Rejected: bad fcst value = 0
DEBUG 3: Rejected: duplicates     = 0
```

Point-Stat: Debug 0 Pairs

1. SID exclusion
   - List of station ID's to be excluded.
2. obs type
   - Does not match verifying obs variable.
3. valid time
   - Fall outside forecast valid time window.
4. bad obs value
   - Skip bad observations (**-9999**).
5. off the grid
   - Obs falls outside the domain (**plot_point_obs**).
6. level mismatch
   - Obs falls outside requested levels.
7. quality marker
   - Not in specified list.
8. message type
   - Not requested message type.
9. masking region
   - Obs falls outside masking region (**plot_point_obs**).
10. bad fcst value
    - Skip bad forecast values (**plot_data_plane**).
11. duplicates
    - Skip duplicate observations.

```
sid_exc = [ "KDEN" ];
```

```
obs = {
 field = [
  { name = "UGRD";
…
```

```
obs_window = {
   beg = -5400;
   end =  5400;
}
```

```
obs = {
 field = [
  { …
    level = "P900-750";
```

```
obs_quality = [ "0", "1", "2" ];
```

```
message_type = [ "ADPUPA" ];
```

```
duplicate_flag = UNIQUE;
```

Point-Stat: Duplicate Obs

- Checking for duplicates is *disabled by default*.
  - Duplicate means same **lat**, **lon**, **level** (height or pressure), **elevation**, and **time** (often have different station ID's).
  - Recommend choosing **UNIQUE**.

```
// Point-Stat and
// Ensemble-Stat
// Config file defaults
duplicate_flag = NONE;
obs_summary    = NONE;
```

```
DEBUG 4: Skipping duplicate observation for [lat:lon:level:elevation] =
         [37.450:-105.870:767.50:2304.34] valid at 20070331_115201 with
         value = 267.15
```

- Observation summaries are disabled by default.
  - When multiple observations occur at a station in the verifying time window (**obs_window**).
  - Recommend choosing **NEAREST** (in time) or **DW_MEAN** (time interpolation).
  - Depending on the application, might also choose **MIN**, **MAX**, **UW_MEAN**, **MEADIAN**, or **PERC##** (where ## defines the percentile).

```
DEBUG 4: Computed NEAREST of 3 observations = 0 for
         [lat:lon:level:elevation] = [37.950:-79.830:890.30:1158.68]
DEBUG 4:   [sid: KHSP vld: 20070331_114059 obs: 286.15]
DEBUG 4:   [sid: KHSP vld: 20070331_120101 obs: 286.15]
DEBUG 4:   [sid: KHSP vld: 20070331_121959 obs: 286.15]
```

May Be Set Separately

- Options are highly configurable.
- Defaults defined at highest level context but can be refined for each field read or verification task.

```
fcst = { field = [ { … } ]; }

obs  = { field = [ { … } ]; }
```

```
// May be set separately in each
// "field" entry
regrid          = { … }
censor_thresh   = [];
censor_val      = [];
cat_thresh      = [ NA ];
cnt_thresh      = [ NA ];
cnt_logic       = UNION;
wind_thresh     = [ NA ];
wind_logic      = UNION;
```

```
// May be set separately in each
// "obs.field" entry
desc            = "NA";
eclv_points     = 0.05;
rank_corr_flag  = FALSE;
message_type    = [ "ADPUPA" ];
sid_exc         = [];
obs_quality     = [];
duplicate_flag  = NONE;
obs_summary     = NONE;
obs_perc_value  = 50;
obs_window      = { … }
mask            = { … }
ci_alpha        = [ 0.05 ];
interp          = { … }
hira            = { … }
output_flag     = { … }
```

Plot-Point-Obs: Usage

Usage: plot_point_obs

nc_file

ps_file

[-gc code] or

[-obs_var name]

[-msg_typ name]

[-data_file name]

[-dotsize val]

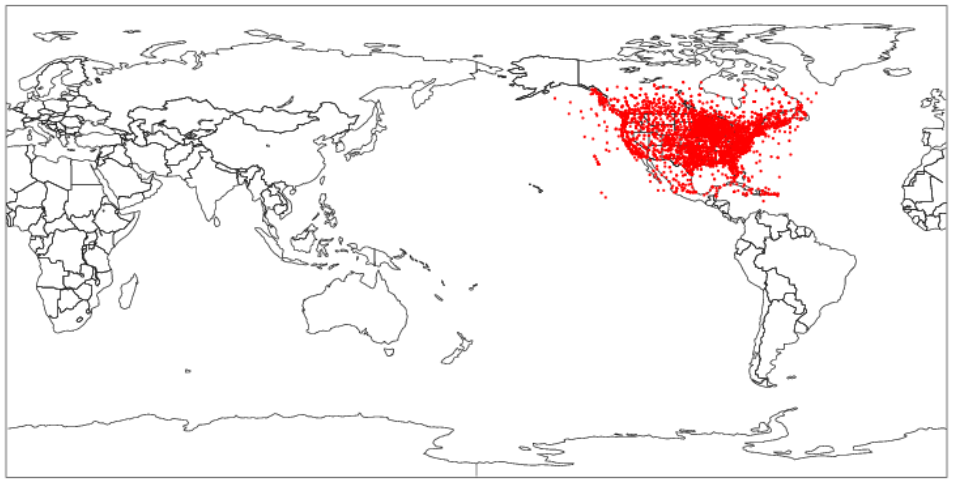
[-log file]

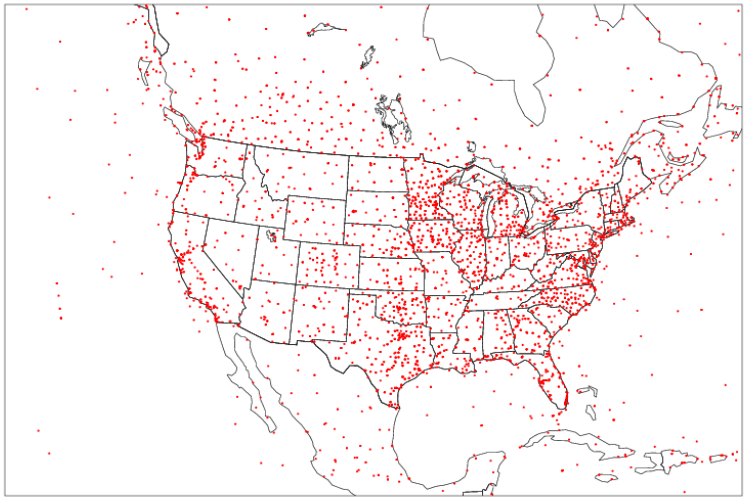
[-v level]

nc_file	Input NetCDF point obs file
ps_file	Output PostScript file
-gc -obs_var	GRIB code(s) (legacy point observation format) or variable name(s) to be plotted
-msg_typ	Beginning of valid time window for matching
-data_file	End of valid time window for matching
-dotsize	Output directory to be used

Plot-Point-Obs: Example

```
plot_point_obs sample_pb.nc UGRD.ps -obs_var UGRD
```





```
plot_point_obs sample_pb.nc UGRD_VXGRID.ps -obs_var UGRD -data_file
               nam.t00z.awip1236.tm00.20070330.grb
DEBUG 2: Retrieving grid from file: nam.t00z.awip1236.tm00.20070330.grb
DEBUG 1: Opening netCDF file: out/pb2nc/sample_pb.nc
DEBUG 2: Processing 87753 observations at 9396 locations.
DEBUG 2: Observation var names: UGRD
DEBUG 2: Observation message types: ALL
DEBUG 1: Creating postscript file: UGRD_VXGRID.ps
DEBUG 2: Finished plotting 4850 locations.
DEBUG 2: Skipped 6 locations off the grid.
```

Rscript: pnt2ascii.R

- Difficult to inspect NetCDF output of PB2NC, ASCII2NC, MADIS2NC, and LIDAR2NC.
- Use Rscript to dump contents to the 11-column ASCII format of ASCII2NC.
- Use grep and other UNIX utilities.

```
Rscript share/met/Rscripts/pntnc2ascii.R out/pb2nc/sample_pb.nc >
sample_pb.txt
```

```
ADPUPA 72233 20070331_120000 30.33 -89.82 8 SPFH 1015 7.988074   2   0.012907000
ADPUPA 72233 20070331_120000 30.33 -89.82 8 TMP  1015 7.988074   2 294.150000000
ADPUPA 72233 20070331_120000 30.33 -89.82 8 HGT  1015 7.988074   2   8.000000000
ADPUPA 72233 20070331_120000 30.33 -89.82 8 DPT  1015 7.988074   2 291.210000000
ADPUPA 72233 20070331_120000 30.33 -89.82 8 RH   1015 7.988074   2  84.748000000
ADPUPA 72233 20070331_120000 30.33 -89.82 8 MIXR 1015 7.988074   2   0.013076000
ADPUPA 72233 20070331_120000 30.33 -89.82 8 SPFH 1000 139.791290 2   0.013608000
... 87754 lines ...
```

Rscript: plot_mpr.R

- Quick way to visualize **MPR** output from Point-Stat.
- Reads one or more input **.stat** or **_mpr.txt** files.
- Identifies unique cases by model, variable name, level, message type, masking region, and interpolation method.
- Creates PDF with summary plots of MPR data.

```
Rscript share/met/Rscripts/plot_mpr.R point_stat_360000L_20070331_120000V.stat
```