Installing and Configuring METplus

George McCabe

(Originally created by Julie Prestopnik)

Developmental Testbed Center

Overview

- MET
  - Release History and Users
  - Downloading and Installing
  - Flowchart
  - Configuration
  - Graphics
- METplus wrappers
  - Downloading and Installing
  - Configuring

Developmental Testbed Center

Developmental Testbed Center

Release History

- MET was first released in 2007
- METv8.1 : Released May 2019
  - Major Enhancements
    - Fortify compliance
    - percentile thresholds
    - land/sea mask and topography in Point-Stat
    - the derive option in PCP-Combine
    - Gaussian interpolation
  - Bugfix release v8.1.1 - July 2019

Release History

- METv8.0: Released September 27, 2018
  - Major Enhancements
    - Python embedding – Read gridded data by calling a Python
    - Observation error logic in Ensemble-Stat
    - Shapefiles for masking regions
    - Point observation format overhaul – time summaries
    - Rotated lat/lon projections
  - New Tools
    - Added three new utilities to dump shapefile information (gis_dump_dbf, gis_dump_shp, and gis_dump_shx)

Release History (cont'd)

- METv8.0: Released September 27, 2018 (cont'd)
  - Other Enhancements (full list can be found in the Release Notes)
    - Regrid-Data-Plane can now read read GOES-16 AOD data using the new "-coord" and "-qc" command line options.
    - ASCII2NC now has support for the AERONET version 3 point observation format and support for longer variable names.
    - In MODE, the "area_thresh" and "inten_perc_thresh" config file options were replaced with the more generic "filter_attr_name" and "filter_attr_thresh" options.
    - Stat-Analysis now has the capability to aggregate the ECNT line type and derive it from input ORANK lines.
    - Series-Analysis has added debug messages to inform the user about missing data.
    - Multiple Tools:
      - Added support for area-weighted (AW_MEAN) regridding logic.
      - For climatologies, added the "match_month" flag to the "climo_mean" and "climo_stdev" config file dictionaries.
      - Enhanced the logic for finding matching UGRD/VGRD verification tasks in Point-Stat and Grid-Stat to make it more robust.

https://dtcenter.org/community-code/model-evaluation-tools-met/download

tara.jensen | Home - Workday | Elevations | personal | Forecast Verification - ... | NOAA/NWS Storm Pr... | Weather Prediction Ce... | NOAA's National Wea... | RAL | Tropical Cyclone... | National Hurricane Ce...

CONTACT US LOGIN

Developmental Testbed Center

ABOUT | TESTING + EVALUATION | COMMUNITY CODE | VISITOR PROGRAM | NEWS | EVENTS

MODEL EVALUATION TOOLS (MET) | DOWNLOAD

RECOMMENDED

VERSION	DOWNLOAD	DATE
8.1.1	met-8.1.1.tar.gz | METplus-2.1.1 | METviewer 2.10 | MET User's Guide 8.1.1 | Existing Builds | Online Tutorial | Release Notes	2019-07-08

OTHER

VERSION	DOWNLOAD	DATE
8.1	met-8.1.tar.gz | METplus-2.1 | METviewer 2.10 | MET User's Guide 8.1 | Docker Container | Existing Builds | Online Tutorial | Release Notes | Known Issues	2019-05-03
8.0	met-8.0.tar.gz | met-8.0_bugfix.tar.gz | METplus-2.0.4 | MET User's Guide 8.0 | Docker Container | Existing Builds | Online Tutorial | Release Notes | Known Issues	2018-09-27

MODEL EVALUATION TOOLS (MET)

- Home
- System Architecture
- Download
  - Terms Of Use
  - Sign Up For Updates
  - Sample Analysis Scripts
  - Input Data
- Documentation +
- User Support +

NCAR / MET

Watch 7 | Star 3 | Fork 1

Code | Issues 215 | Pull requests 0 | Projects 9 | Wiki | Security | Insights | Settings

Label issues and pull requests for new contributors

Dismiss

Now, GitHub will help potential first-time contributors discover issues labeled with help wanted or good first issue

Go to Labels

Filters | is:issue is:open | Labels 39 | Milestones 3 | New issue

215 Open | 955 Closed | Author | Labels | Projects | Milestones | Assignee | Sort

- **MET configure/build error with rpl_malloc in config.h** priority: high, type: bug
  #1171 opened 1 hour ago by davidfillmore
- **Simplify logic when processing point observations named APCP.** component: application code, component: library code, priority: high, requestor: Air Force, type: enhancement
  #1169 opened 4 days ago by JohnHalleyGotway | MET 9.0
- **The computed CAPE value is not correct** component: application code, priority: high, type: bug
  #1167 opened 5 days ago by hsoh-u | MET 8.1.2 (bugfix)
- **Update Fortify SCA version to 18.20.** component: fortify, priority: medium, requestor: Air Force, type: task
  #1165 opened 5 days ago by JohnHalleyGotway | MET 9.0
- **Support EOL sounding composite format for acscii2nc** component: application code, priority: medium, requestor: community, type: enhancement
  #1164 opened 6 days ago by hsoh-u | MET Future Versi...

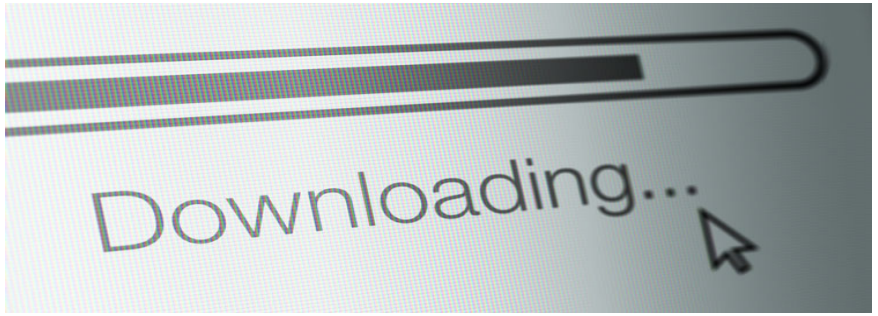
MET Users

- 3800+ registered MET users from 134 countries
  - 48/27/14/11%: University/Gov't/Nonprofit/Private
  - 30/17/6%: USA/China/India
- On-line and hands-on tutorial

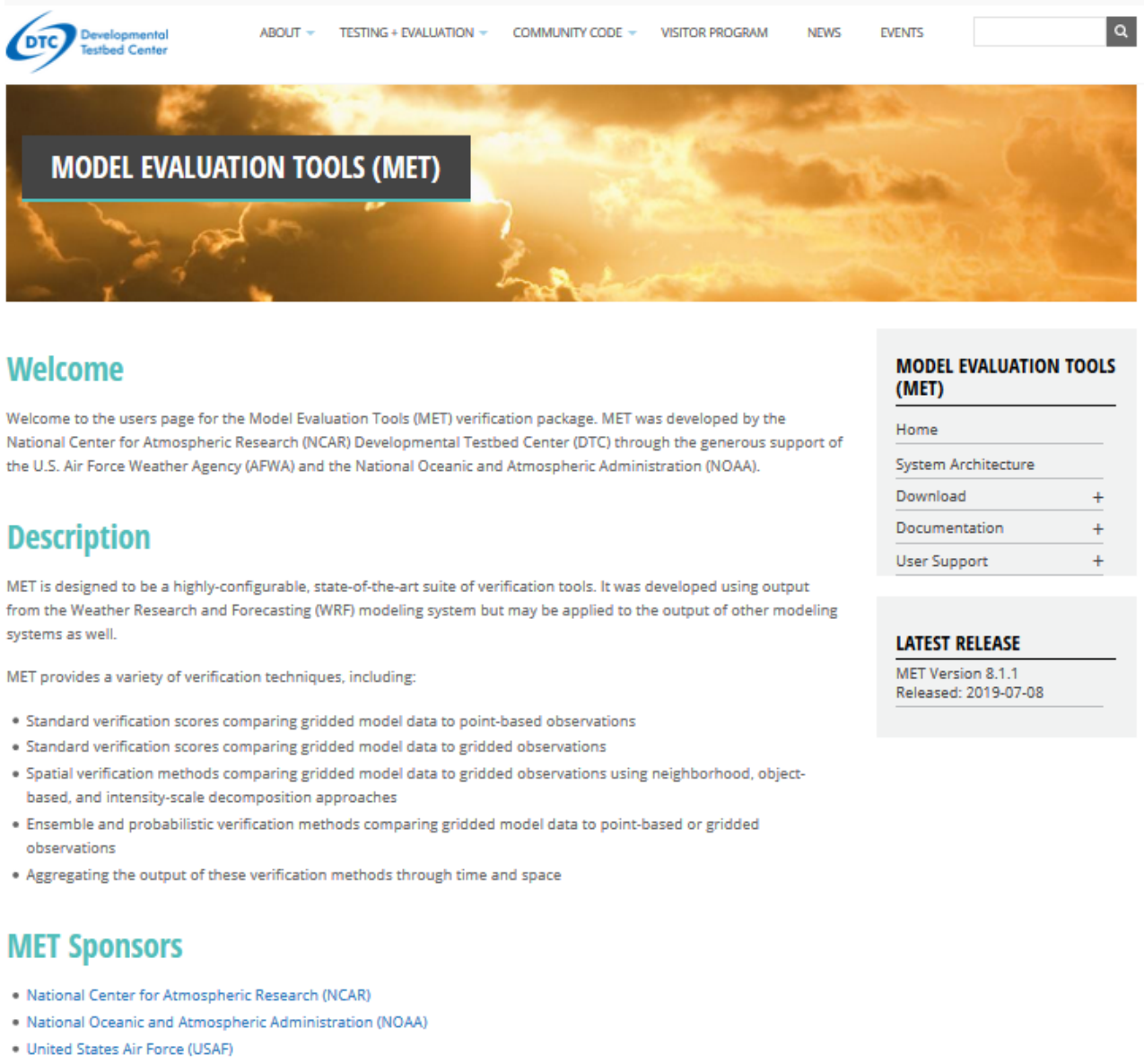
Developmental Testbed Center

Downloading MET

- Download MET release and compile locally
  - https://dtcenter.org/community-code/model-evaluation-tools-met/download
- Language:
  - Primarily C++ with calls to a Fortran Library
  - Python embedding
- Supported Platforms and Compilers
  1. Linux with Intel compilers
  2. Linux with GNU compilers
  3. Linux with Portland Group (PGI) compilers

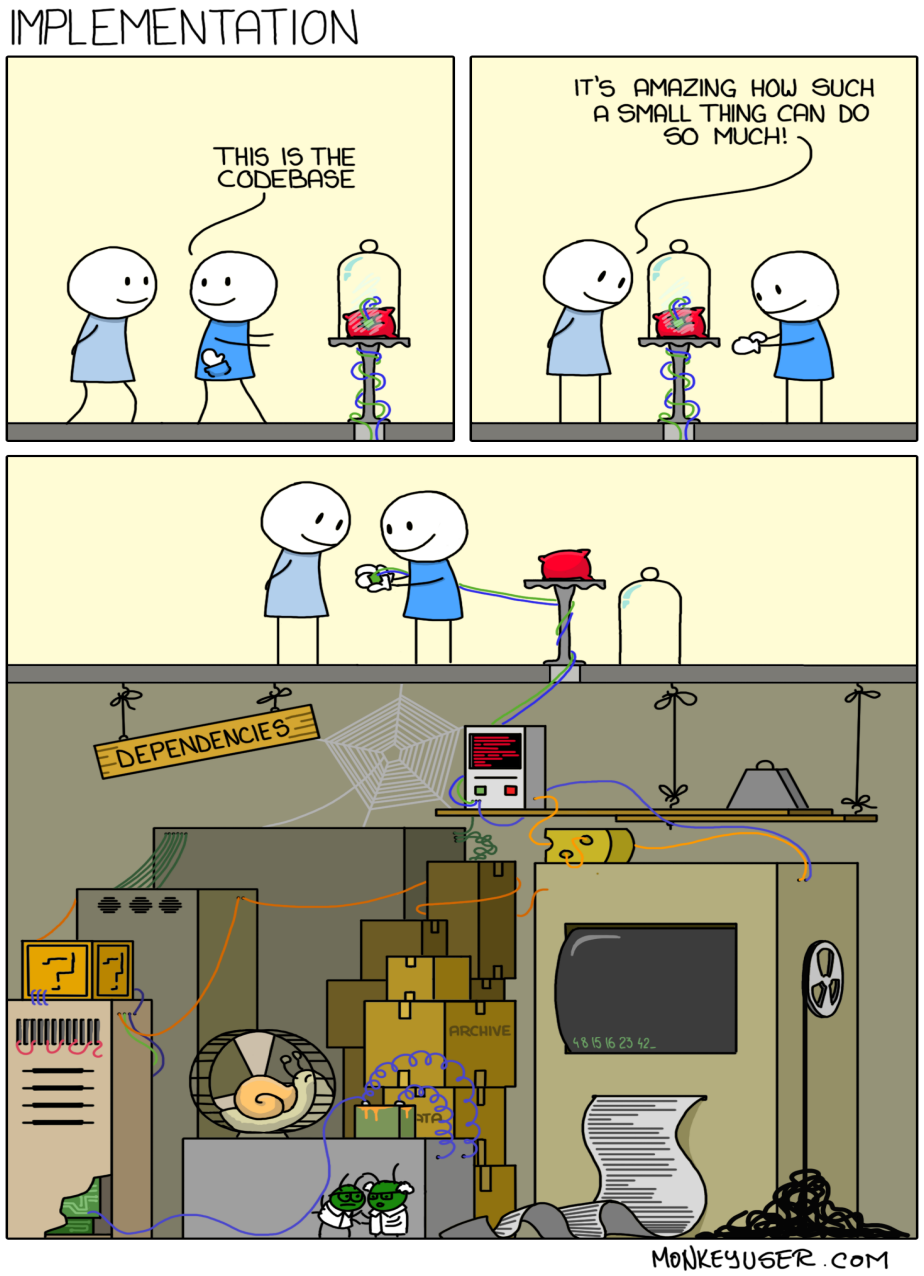


dtcenter.org/community-code/model-evaluation-tools-met



MET Dependencies

- Required
  - C++/Fortran Compilers (GNU, PGI, Intel)
  - GNU Make Utility
  - Unidata's NetCDF4 library (both NetCDF-C and NetCDF-CXX)
  - HDF5 library (required to support NetCDF4)
  - NCEP's BUFRLIB Library v10.2.3
  - GNU Scientific Library (GSL)
  - Z Library (zlib)
- Optional
  - GRIB2 C-Library with JASPER and PNG libraries
  - HDF4 and HDF-EOS2 libraries for MODIS-Regrid tool
  - Cairo and FreeType libraries for MODE-Graphics tool
  - Python 2.7 for Python embedding (numpy or xarray and netCDF4 useful)
- Recommended
  - Unified Post-Processor
  - COPYGB (included with Unified Post-Processor)
  - wgrib and wgrib2
  - R statistics and graphics package

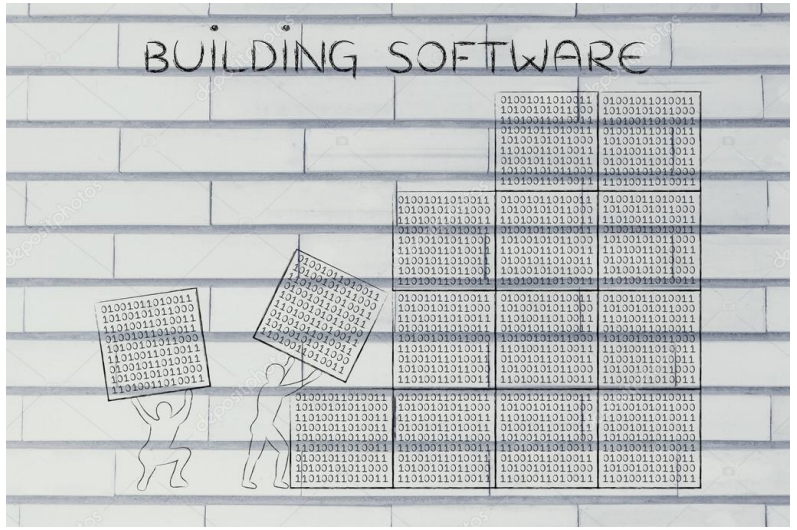


Developmental Testbed Center

Directory Structure

File or Directory	Contents
README	Installation instructions and release notes
configure (plus supporting files and subdir)	Used by the autoconf build process; Configures the MET package for installation on a system
bin/	Built MET executables
data/	Contains map data, colortables, sample input data, GRIB and GRIB2 table files, and default configuration files
doc/	MET User's Guide
out/	Output generated by the test scripts
scripts/	Test scripts to be run after building MET
src/	MET Source Code

Building MET

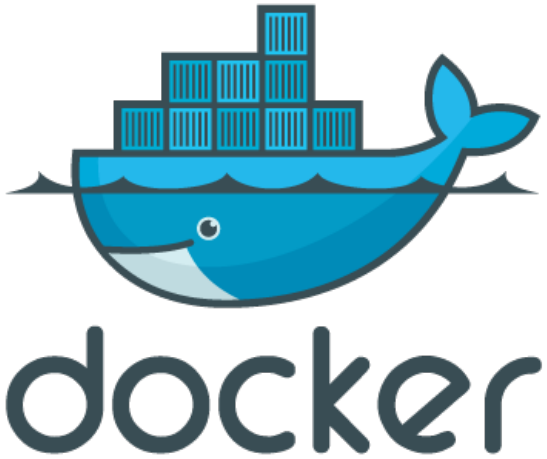


- Steps for building MET:
  1. Build required/optional libraries.
     - Same family of compilers for MET
  2. Download and unpack latest MET patches.
  3. autoconf determines available compilers, but can be explicitly set by the user
  4. Set environment variables in .cshrc or equivalent file
     - Paths for HDF5, NetCDF, BUFRLIB, and GSL libraries plus optional libraries if desired
  5. Configure the installation for your system and run configure
     - The compilation of various tools can be turned on/off
  6. Run make install and make test and check for runtime errors.
     - Test scripts run each of the MET tools at least once.
     - Uses sample data distributed with the tarball.

Existing MET Builds

- MET User's Page
  - Download
- Cheyenne
  - module use /glade/p/ral/jntp/MET/MET_releases/modulefiles
  - module load met/8.1
- NCAR RAL Machines
- NOAA machines

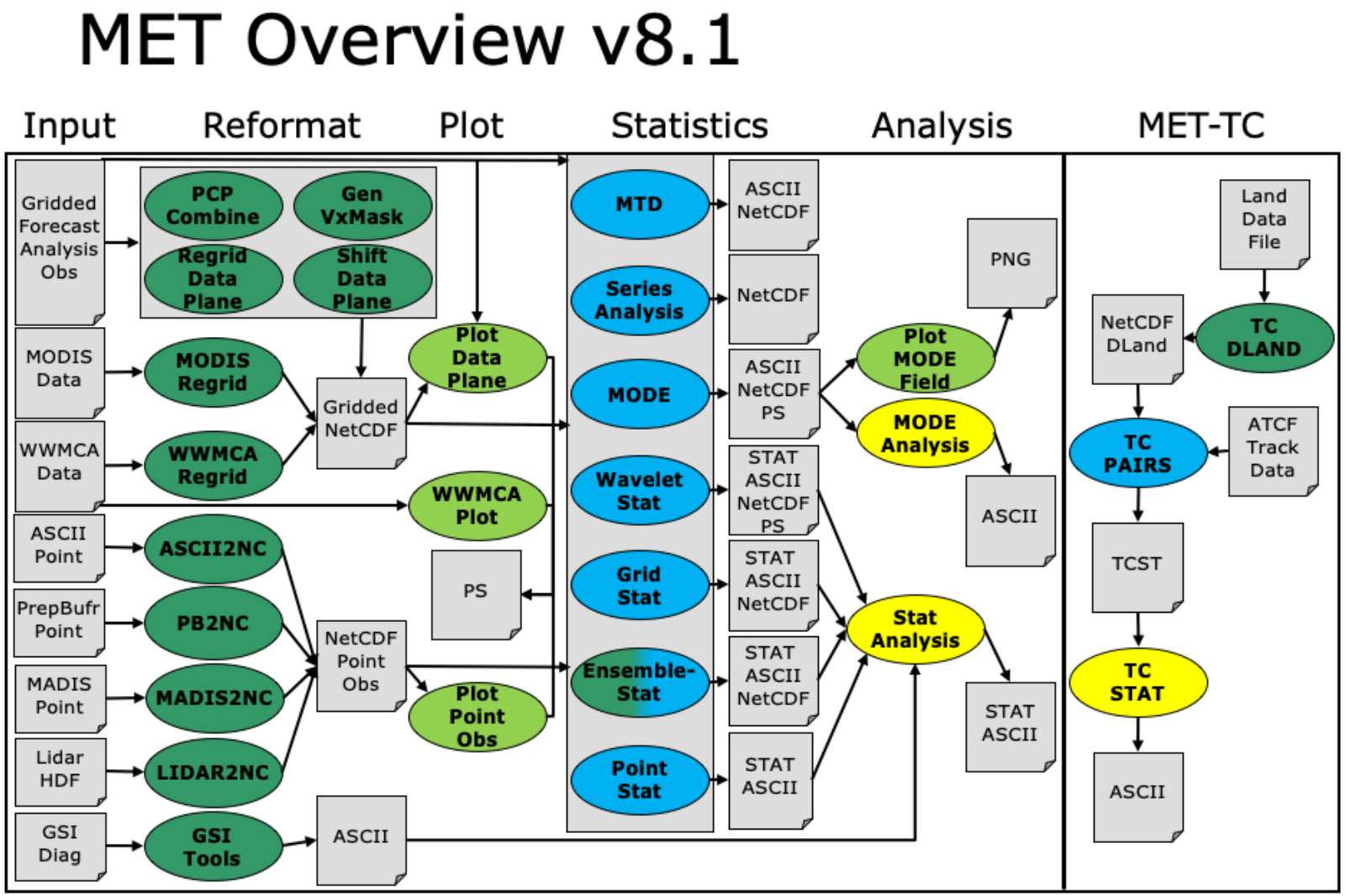
Docker Container



- MET User's Page
  - Download -> Docker Container
    - Select desired MET Version
  - Direct Link
    - https://dtcenter.org/community-code/model-evaluation-tools-met/docker-container-met-8-1

- GitHub Repo
  - Instructions for building and using MET
    - https://github.com/NCAR/container-dtc-met

MET Flowchart

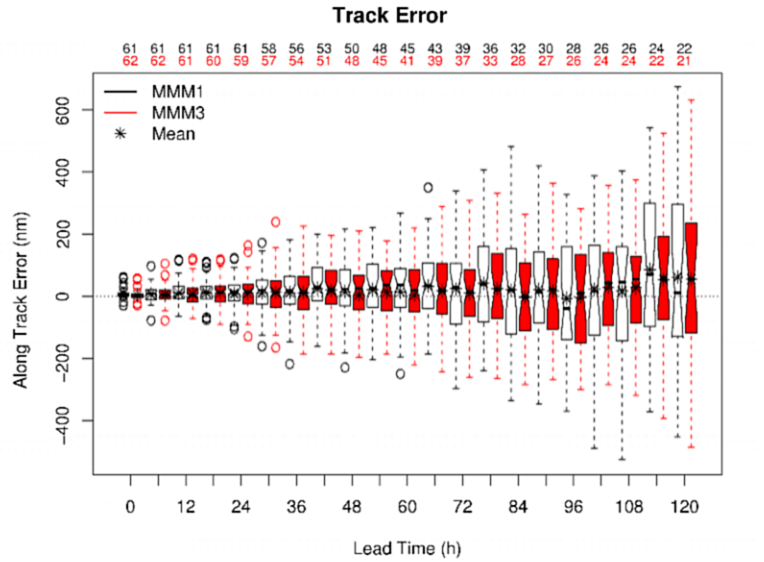


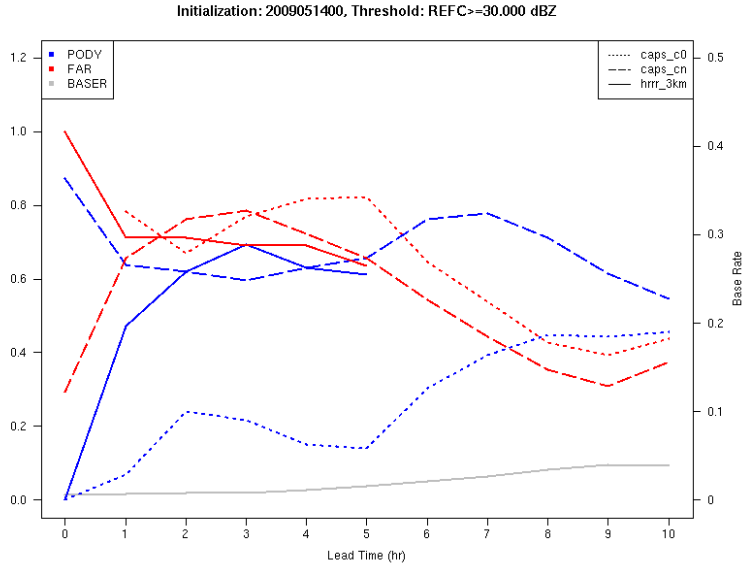
Configuration Files

- MET tools controlled using command line options and ASCII configuration files
  - Well commented and documented in MET User's Guide
  - Easy to modify
  - Distributed with the tarball
- Configuration files control things such as:
  - Fields/levels to be verified
  - Thresholds to be applied
  - Interpolation methods to be used
  - Verification methods to be applied
  - Regions over which to accumulate statistics
  - README file in data/config area describes various settings

Graphics

- Limited graphics incorporated into MET
- Options for plotting MET statistical output
  - R, NCL, IDL, GNUPlot, Python (and more to come!)
- Sample plotting scripts on MET website
- METviewer database/display system





R Statistics and Graphics

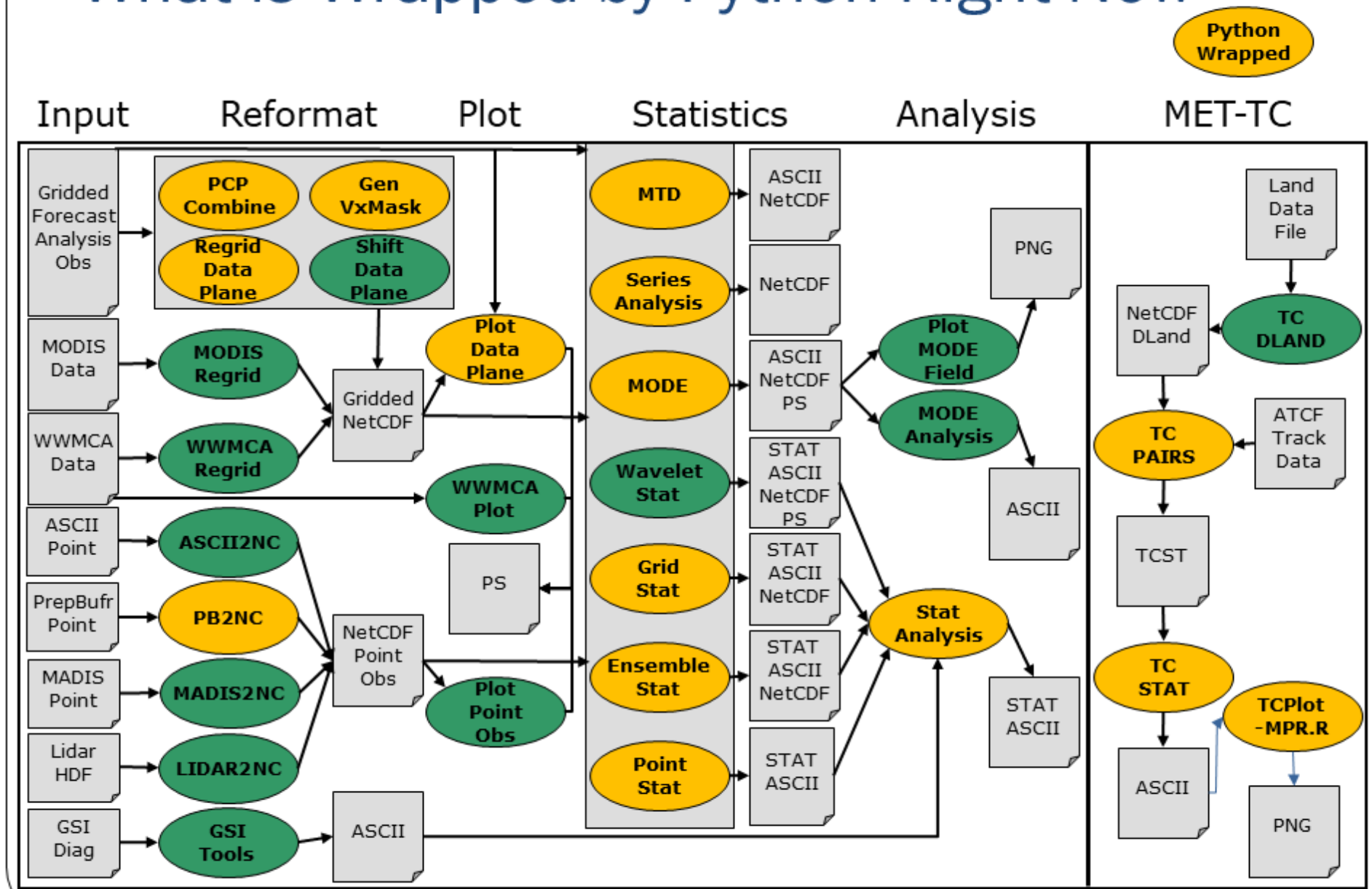
- The R Project for Statistical Computing (www.r-project.org)
  - Powerful statistical analysis and plotting tools
  - Large and growing user community
  - Freely available and well supported for Linux/Windows/Mac
- Sample R plotting and analysis scripts posted on the MET website
- Use R to plot data in the practical sessions

DTC Developmental Testbed Center

METplus Wrappers

Developmental Testbed Center

What is Wrapped by Python Right Now

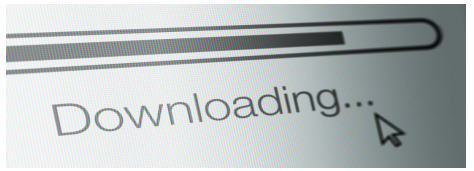


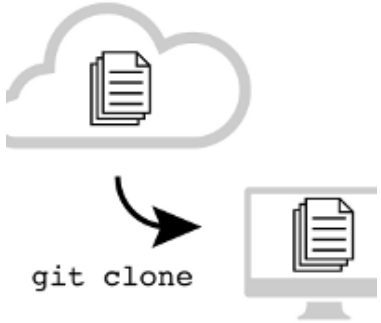
METplus Wrappers Dependencies

- MET and its dependencies
- Python 2.7 (moving to Python 3 very soon) including packages listed below
  - Numpy
  - For plotting: matplotlib, cartopy
  - For development: pytest, pytest-regtest
- External Binaries (not all are required – depends on desired wrappers)
  - generally widely available on most unix machines
    - cut, tr, rm, convert (ImageMagik), egrep, ncap2 (NCO utility), wgrib2 (for use with GRIB2 files)
  - R (for plotting)

Getting the METplus Wrappers package

- Create a directory to store the package
- Obtain the source code
  - Two options:
    1. Getting the latest release
       - https://github.com/NCAR/METplus/releases
       - Source, configuration, and docs
       - Sample data (separate packages)
    2. Clone the repository
       - Best if contributing code
       - git clone https://github.com/NCAR/METplus



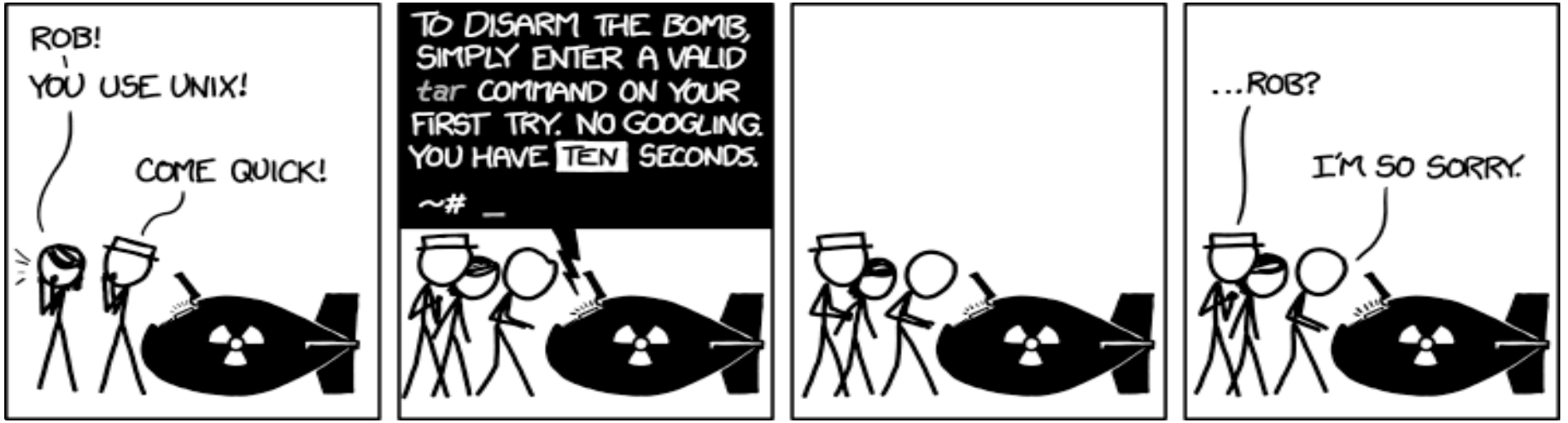


  - Note: No GitHub account is necessary. The repository is public.

Developmental Testbed Center

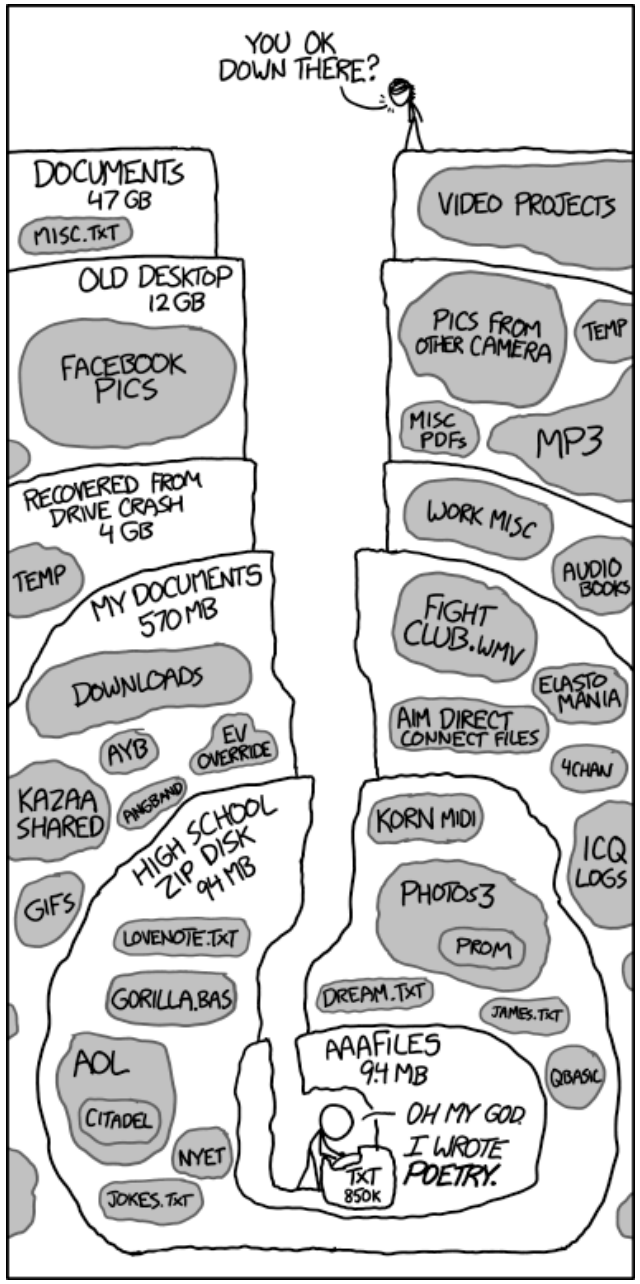
Downloading and Unpacking

- In a given directory:
  - wget https://github.com/NCAR/METplus/archive/vX.Y.tar.gz
  - (Can also download via link on Releases page)
  - tar zxf vX.Y.tar.gz
- This will create a METplus-X.Y directory.



Directory Structure

File or Directory	Contents
doc	METplus User's Guide
internal_tests	For developers
parm	Parameter (configuration) files
sorc	For developers (doxygen)
ush	Python scripts



Developmental Testbed Center

Getting Sample Data

- In a given directory: (where X.Y is the version of METplus, i.e. 2.1)
  - mkdir METplus-X.Y_Data
  - cd METplus-X.Y_Data
- wget https://github.com/NCAR/METplus/releases/download/vX.Y/sample_data-<use_case>-X.Y.tar.gz
- tar zvf sample_data-qpf-2.1.tar.gz (creates qpf directory)
- rm sample_data-qpf-2.1.tar.gz
- Set INPUT_BASE to METplus-X.Y_Data to run use cases

  (more on this later)

Developmental Testbed Center

Environment Setup

- Example Directory Structure
  - METplus Wrappers
    - Created directory: /d1/<user>
    - Unpacked METplus wrappers package in that location, so now a METplus-X.Y directory exists at /d1/<user>/METplus-X.Y
  - MET
    - MET installation: /d1/met-X.Y
    - The bin directory with the MET executables exists at /d1/met-X.Y/bin
  - METplus Wrappers X version number is generally MET version X - 6
    - MET 8.1 corresponds to METplus Wrappers 2.1

Environment Setup (cont'd)

- Set the following environment variables (use setenv for csh, export for bash)
  - MET_PATH: Set to the location where met-8.1 is installed (for R scripts)
    - e.g. setenv MET_PATH /d1/met-8.1
  - PATH: Add the METplus wrappers ush directory (to run code from anywhere)
    - e.g. setenv PATH /d1/<user>/METplus-2.1/ush: ${PATH}

Configuration Files

- Anatomy of a configuration file (sections/"families")
  - [config] – General options (timing, field info, logging)
  - [dir] – Input/Output data (and other) directories
  - [exe] – Paths to non-MET executables
  - [filename_templates] – Format template for filenames
  - [regex_pattern] – Regular expression patterns
- *Note: The section/"family" name (item in []) MUST precede configuration options of that type in the configuration file.*

Configuration Best Practices

1) Copy parm directory to user location
   - Users may work with a shared installation of METplus wrappers
     - Changes to the parm (configuration) directory would affect all users
     - May not have permission to edit parm directory in shared location
   - May want some settings to be static over all runs
2) Create user specific configuration files
   - Used to override configuration settings for a given run
   - Contains only the settings overridden from default in one place
   - Preserves user settings when moving to new version of scripts

User Specific Configuration File

[dir]

INPUT_BASE = /d1/METplus/METplus-2.1_Data

OUTPUT_BASE = /d1/julie/METplus/data/output

MET_INSTALL_DIR = /d1/met/8.1

TMP_DIR = {OUTPUT_BASE}/tmp

[config]

LOG_DIR = {OUTPUT_BASE}/logs

LOG_MET_VERBOSITY = 5

LOG LEVEL = DEBUG

User Specific Configuration File (cont'd)

```
[exe]
WGRIB2 = wgrib2
CUT = cut
TR = tr
RM = rm
NCAP2 = /usr/local/nco/bin/ncap2
CONVERT = convert
NCDUMP = {MET_INSTALL_DIR}/external_libs/bin/ncdump
EGREP = egrep
```

Developmental Testbed Center

Running a Sample Test Case

- Main script – master_metplus.py
- Pass in configuration files using -c option
  - Order matters! Later configuration files override earlier files
- Running precip ruc-vs-s2grib test case:
  - master_metplus.py -c /d1/julie/METplus/METplus-2.1/parm/use_cases/qpf/examples/ruc-vs-s2grib.conf -c /d1/julie/METplus/METplus-2.1/parm/julie.system.conf.cheyenne

Questions?

Developmental Testbed Center