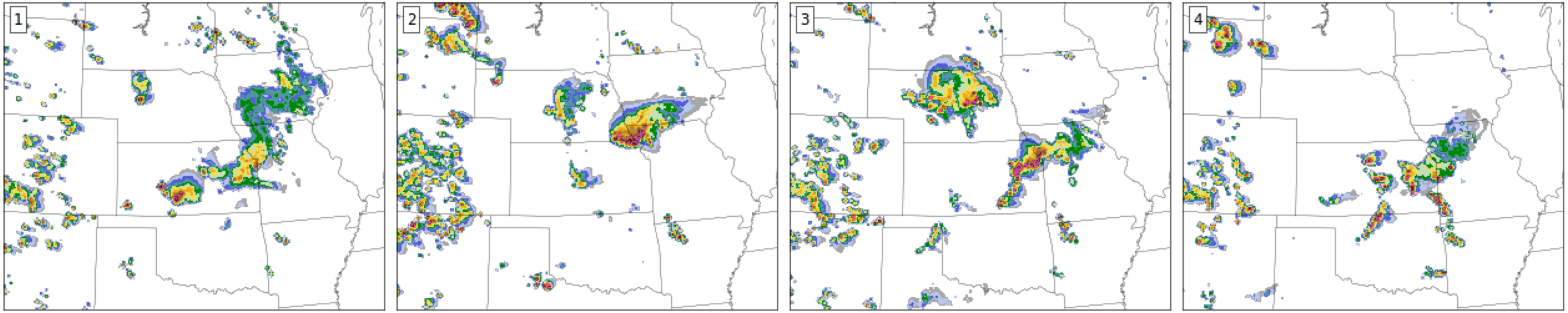
Visualizing WRF and MPAS datasets with Python

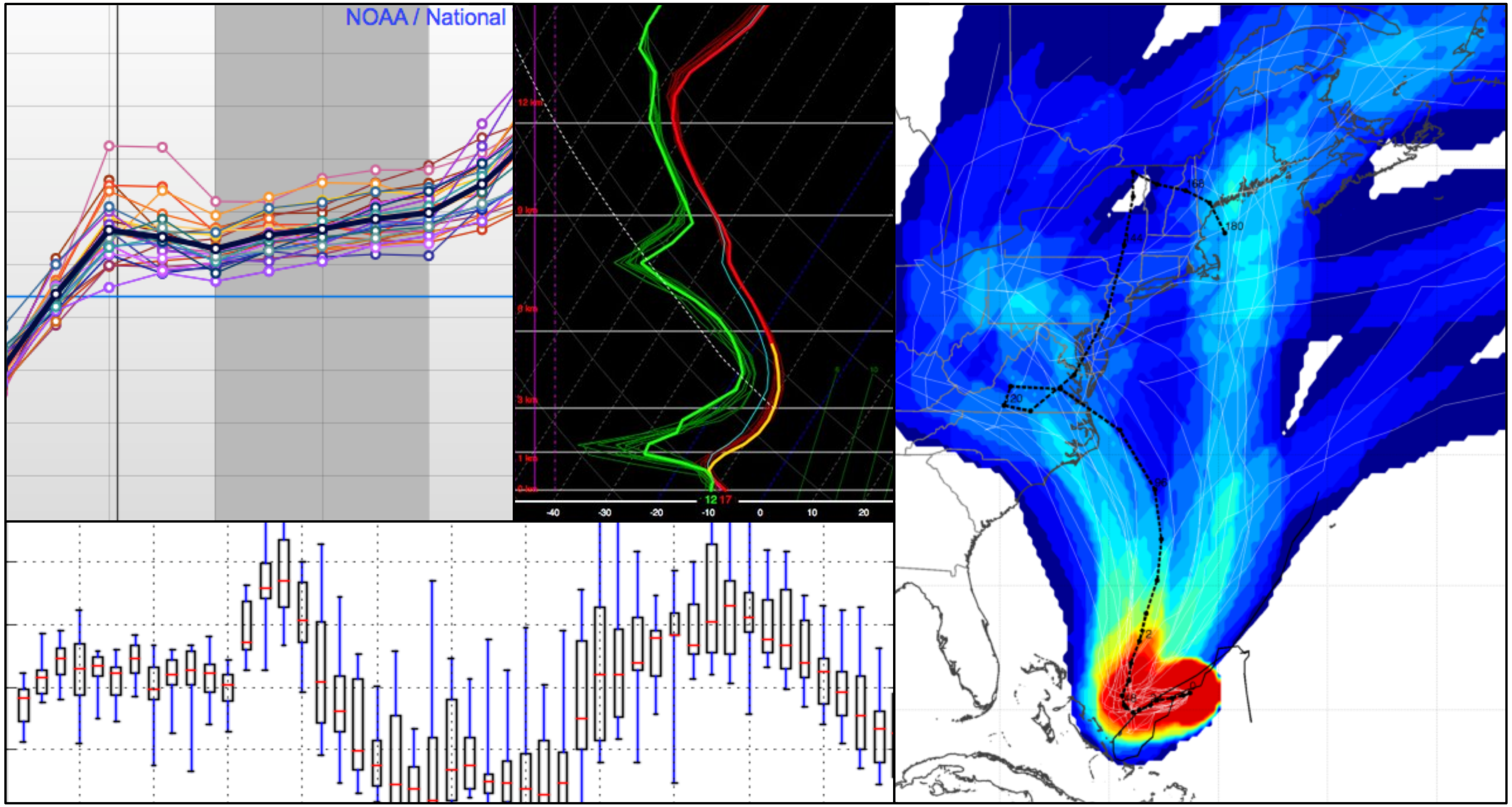
Ryan Sobash, NCAR/Mesoscale and Microscale Meteorology (MMM) Laboratory, sobash@ucar.edu

Thanks to: Dave Ahijevych, Craig Schwartz, Glen Romine, Kate Fossell

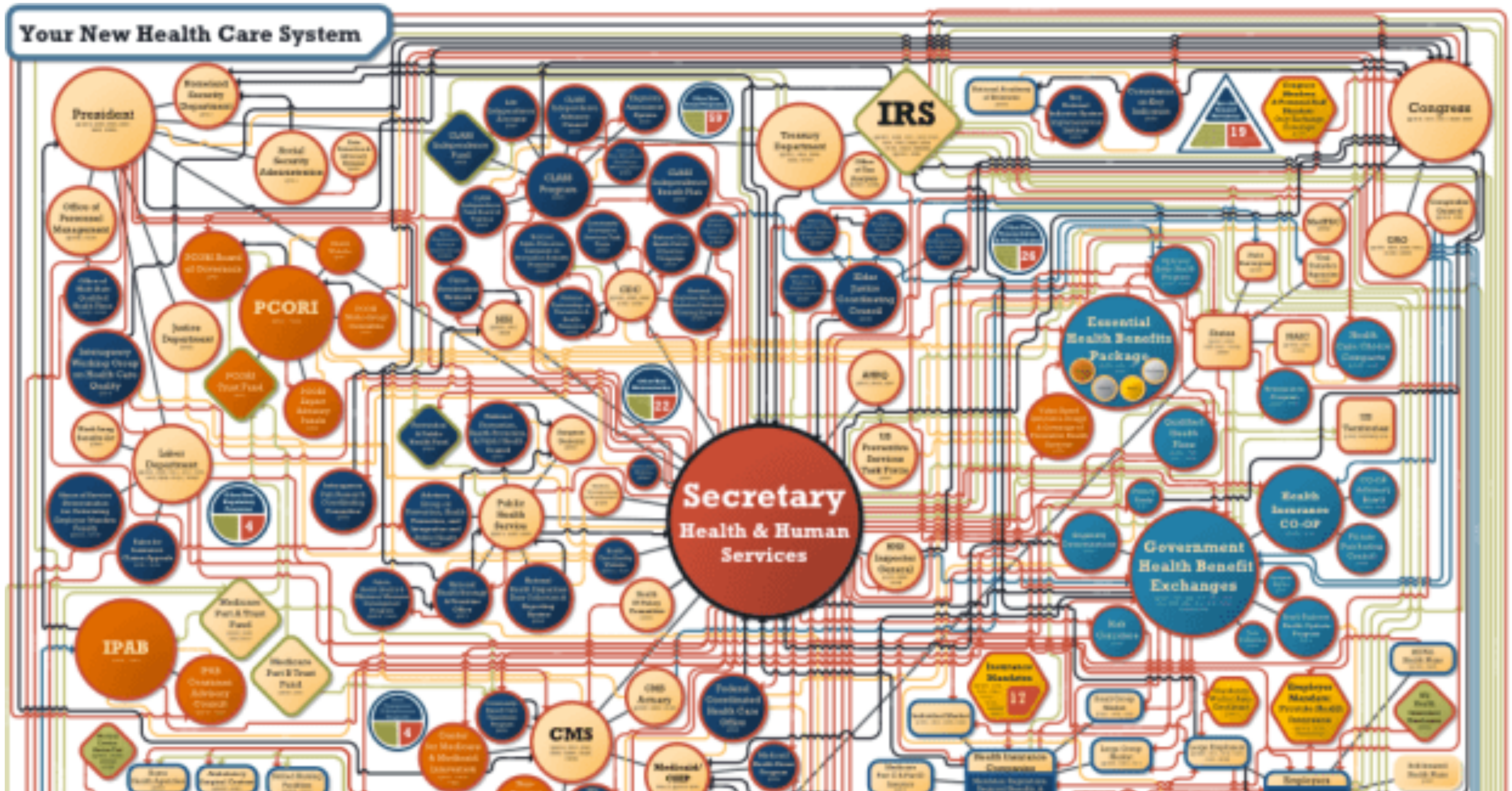


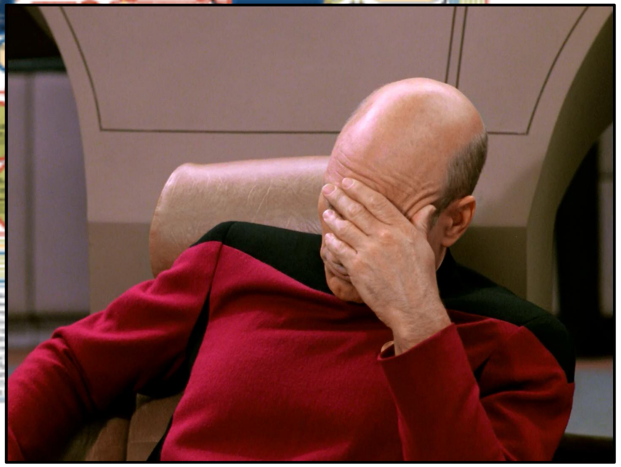
2019 WRF Users' Workshop, Boulder, CO

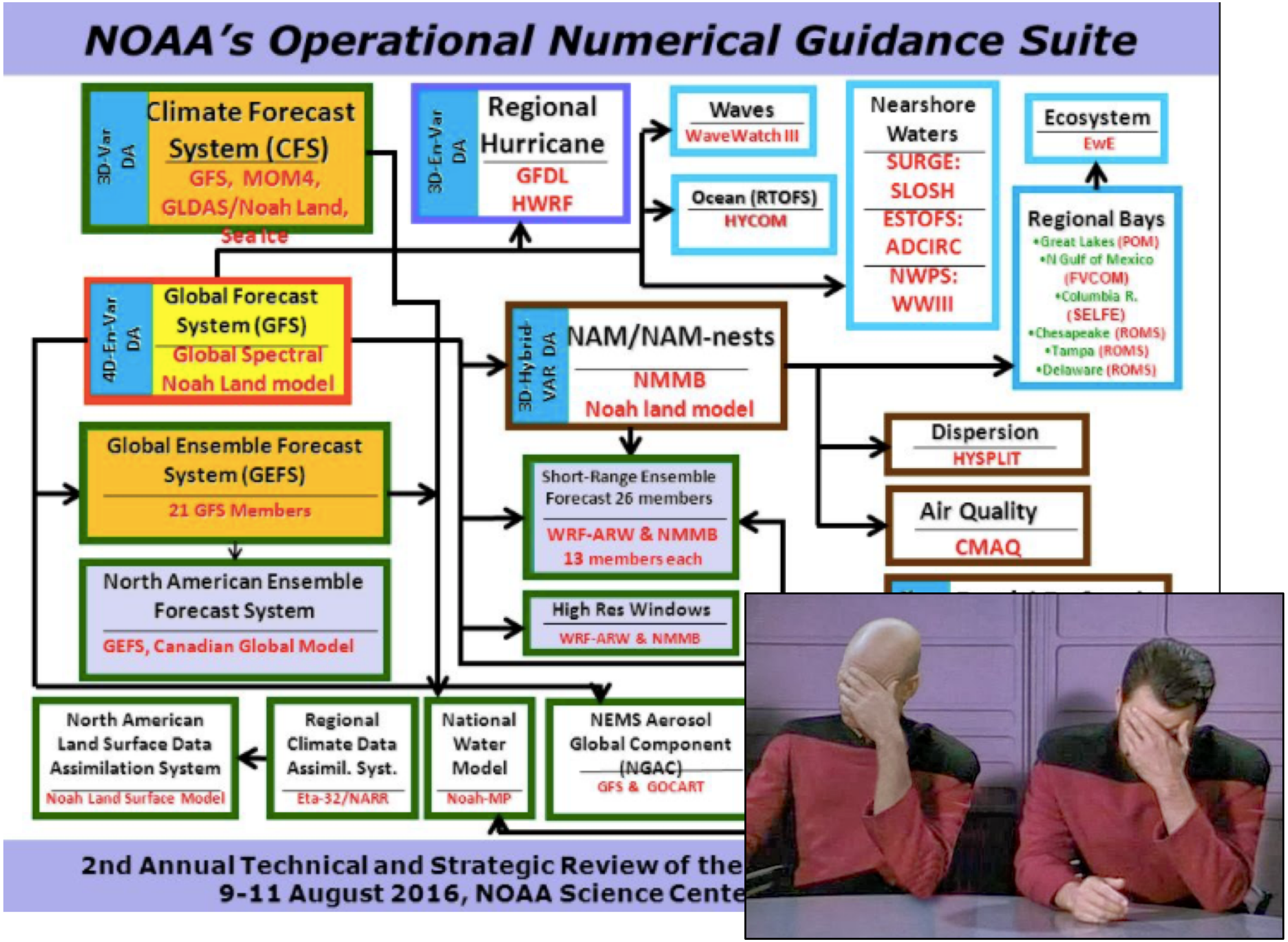
Effective visualization of ensemble forecast data in real-time is a challenge…



Relies partly on **good tools** and partly on **good taste**.
I'll mostly talk about the former.


Your New Health Care System
President
Homeland Security Department
Social Security Administration
Office of Personnel Management
PCORI Board of Governors
PCORI
PCORI Trust Fund
Justice Department
Interagency Working Group on Health Care Quality
Labor Department
IPAB
Medicare Part A Trust Fund
Medicare Part B Trust Fund
CLASS Independence Fund
CLASS Program
CLASS Independence Benefit Plan
CDC
NIH
Public Health Service
Treasury Department
IRS
Congress
CBO
Comptroller General
AHRQ
Elder Justice Coordinating Council
US Preventive Services Task Force
HHS Inspector General
Secretary
Health & Human Services
Essential Health Benefits Package
States
Health Insurance CO-OP
Government Health Benefit Exchanges
CMS
CMS Actuary
Center for Medicare & Medicaid Innovation
Medicaid/CHIP
Medicare
Insurance Mandates
Employer Mandate
Health Insurance Companies
Employers
Health Industry
Physicians
19
26
22
4
4
17
New Government
Rationing Potential
Involvement in Health Insurance Market
Other Expansions
Represents Bundles of Additional Entities
Mandates
Taxes & Monetary Fees/Penalties/Cuts
Trust Fund (Rationing Potential)
Other New Trust Funds/Monetary Benefits
Expanded Government
Government with Expanded Authority/Responsibility
Government Financial Entity with New Inflows/Outflows
State/Territory with Expanded Authority/Responsibility
Private
Private Entity with New Mandates/Regulations/Responsibilities
Unchanged Private Entity
Special Interest Provisions
New Relationships
Regulations/Requirements/Mandates
Reporting Requirements
Oversight
Money Flows
Consultation/Advisory/Info Sharing
Structural Connections (Includes Existing)





NOAA's Operational Numerical Guidance Suite

3D-Var DA

Climate Forecast System (CFS)

GFS, MOM4, GLDAS/Noah Land, Sea Ice

3D-En-Var DA

Regional Hurricane

GFDL
HWRF

Waves

WaveWatch III

Ocean (RTOFS)

HYCOM

Nearshore Waters

SURGE: SLOSH

ESTOFS: ADCIRC

NWPS: WWIII

Ecosystem

EwE

Regional Bays

- Great Lakes (POM)
- N Gulf of Mexico (FVCOM)
- Columbia R. (SELFE)
- Chesapeake (ROMS)
- Tampa (ROMS)
- Delaware (ROMS)

4D-En-Var DA

Global Forecast System (GFS)

Global Spectral
Noah Land model

3D-Hybrid-VAR DA

NAM/NAM-nests

NMMB
Noah land model

Dispersion

HYSPLIT

Air Quality

CMAQ

Global Ensemble Forecast System (GEFS)

21 GFS Members

Short-Range Ensemble Forecast 26 members

WRF-ARW & NMMB
13 members each

North American Ensemble Forecast System

GEFS, Canadian Global Model

High Res Windows

WRF-ARW & NMMB

North American Land Surface Data Assimilation System

Noah Land Surface Model

Regional Climate Data Assimil. Syst.

Eta-32/NARR

National Water Model

Noah-MP

NEMS Aerosol Global Component (NGAC)

GFS & GOCART

2nd Annual Technical and Strategic Review of the
9-11 August 2016, NOAA Science Cente

Development of real-time convection-allowing forecast system

In 2015, a real-time, WRF-based, convection-allowing forecast system was developed at NCAR/MMM.

Previous real-time systems were only run for short periods (e.g., a month) and mainly used NCL or RIP for visualization.

New project provided a testbed for the usage of python-based tools to visualize and post-process WRF ensemble forecasts in real-time.

A new visualization package was developed with Python to run within the real-time forecasting framework.

Some constraints…

Wanted something that would run fast and could scale well, taking advantage of NCAR visualization cluster resources (at the time, geyser/caldera).

Wanted to build a framework that would be easily adaptable for use with other forecast datasets and could easily make new products as ideas emerged.

Wanted to separate the visual aspects of graphics (color table, interval frequencies, line widths, etc) from the actual plotting code, to rapidly iterate on best ways to effectively visualize convection-allowing ensemble data for users.

Wanted ability to plot novel convection-allowing ensemble fields.

Python-based post-processing system

Custom designed python package to visualize convection-allowing ensemble data (simply referred to as “webplot” internally, since it’s prime use-case is plotting for use on the web).

Very small footprint, not structured as official python package (yet)

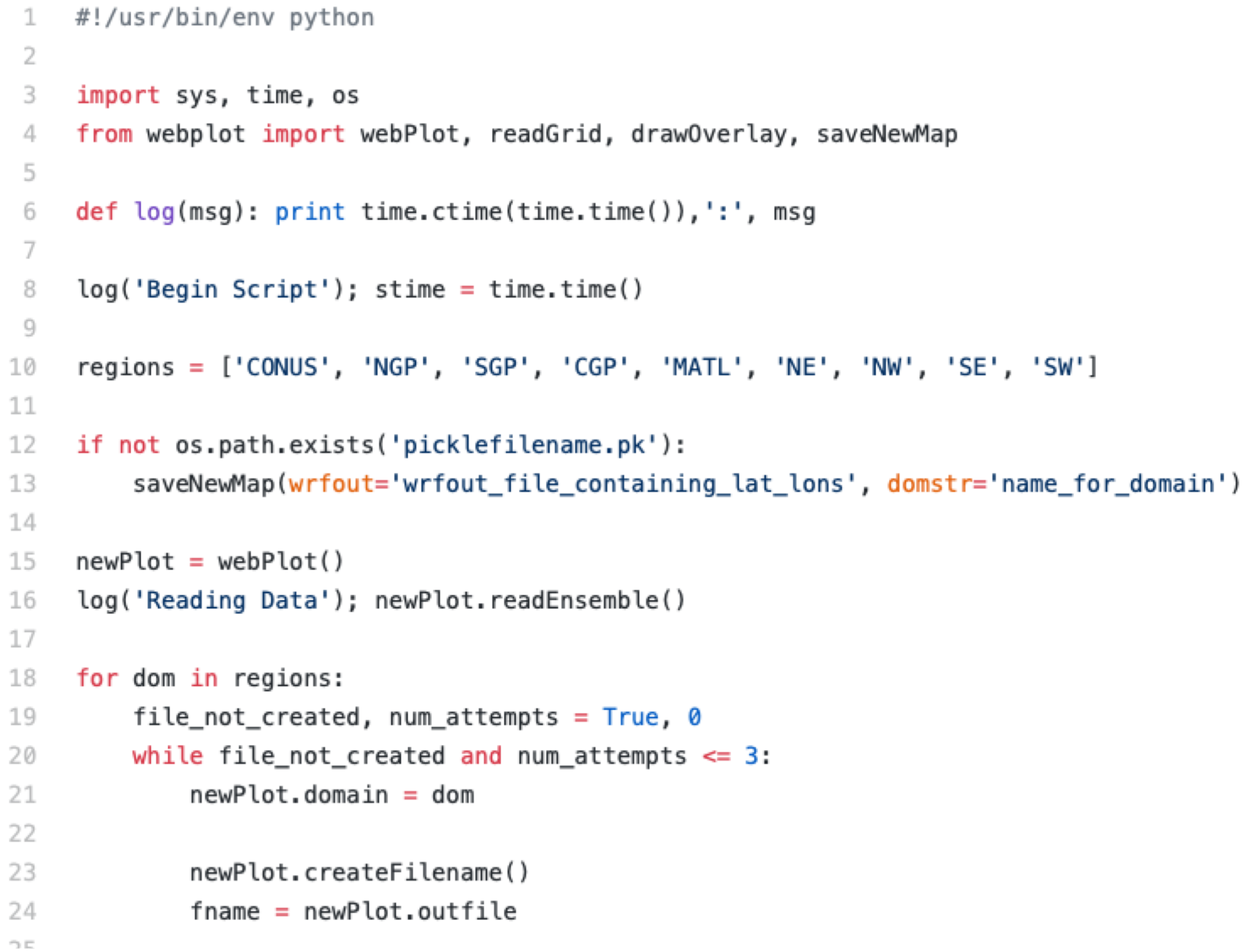
Depends on output fields being available in WRF or MPAS output files (e.g., package will not compute CAPE). Some support for derived fields.

Dependencies: numpy, scipy, matplotlib, netcdf4-python, basemap.

https://github.com/rsobash/webplot

Object-oriented based code: a new object is a plot type and can be run separately.

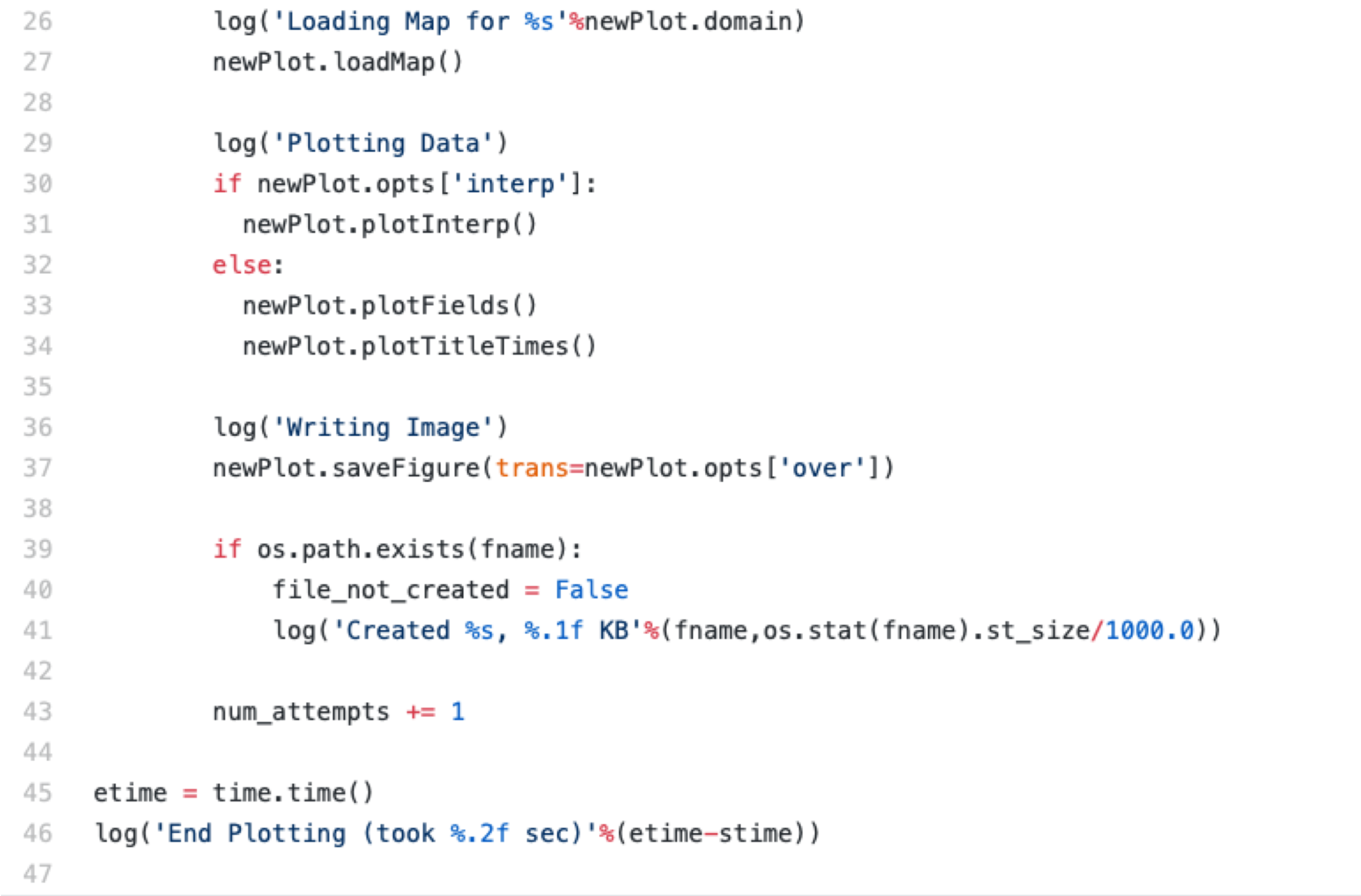
Model data read in once, then plotted on pre-stored basemap projection files for each region.



```python
#!/usr/bin/env python

import sys, time, os
from webplot import webPlot, readGrid, drawOverlay, saveNewMap

def log(msg): print time.ctime(time.time()),':', msg

log('Begin Script'); stime = time.time()

regions = ['CONUS', 'NGP', 'SGP', 'CGP', 'MATL', 'NE', 'NW', 'SE', 'SW']

if not os.path.exists('picklefilename.pk'):
    saveNewMap(wrfout='wrfout_file_containing_lat_lons', domstr='name_for_domain')

newPlot = webPlot()
log('Reading Data'); newPlot.readEnsemble()

for dom in regions:
    file_not_created, num_attempts = True, 0
    while file_not_created and num_attempts <= 3:
        newPlot.domain = dom

        newPlot.createFilename()
        fname = newPlot.outfile
```

Arguments passed in to define what fields to plot.

Load map projection file, plot fields, titles, save png files.



```
    log('Loading Map for %s'%newPlot.domain)
    newPlot.loadMap()

    log('Plotting Data')
    if newPlot.opts['interp']:
     newPlot.plotInterp()
    else:
     newPlot.plotFields()
     newPlot.plotTitleTimes()

    log('Writing Image')
    newPlot.saveFigure(trans=newPlot.opts['over'])

    if os.path.exists(fname):
      file_not_created = False
      log('Created %s, %.1f KB'%(fname,os.stat(fname).st_size/1000.0))

    num_attempts += 1

etime = time.time()
log('End Plotting (took %.2f sec)'%(etime-stime))

```

Python-based post-processing system

`./make_webplot.py -d=2019010800` ***Ensemble forecast date***

Forecast hour/range `-tr=40`

Fill field `-f=td2_mean`

Barb field `-b=wind10m_mean`

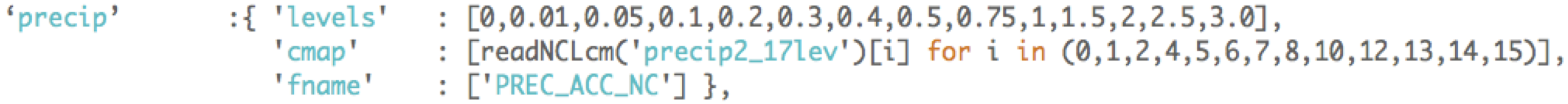
Contour field `-c=mslp_mean`

Title

```
-t=’Ensemble mean 2-m temperature (fill;F),
    MSLP (contour; hPa), and 10-m wind (kts)'
```

Python-based post-processing system

Python dictionary to control plot properties



```
‘precip’         :{ 'levels'    : [0,0.01,0.05,0.1,0.2,0.3,0.4,0.5,0.75,1,1.5,2,2.5,3.0],
                    'cmap'      : [readNCLcm('precip2_17lev')[i] for i in (0,1,2,4,5,6,7,8,10,12,13,14,15)],
                    'fname'     : ['PREC_ACC_NC'] },
```

Python-based post-processing system

```
./make_webplot.py -d=2018060800 -tr=28 -f=stp_mean
                  -t=’Ensemble mean significant tornado parameter'
```

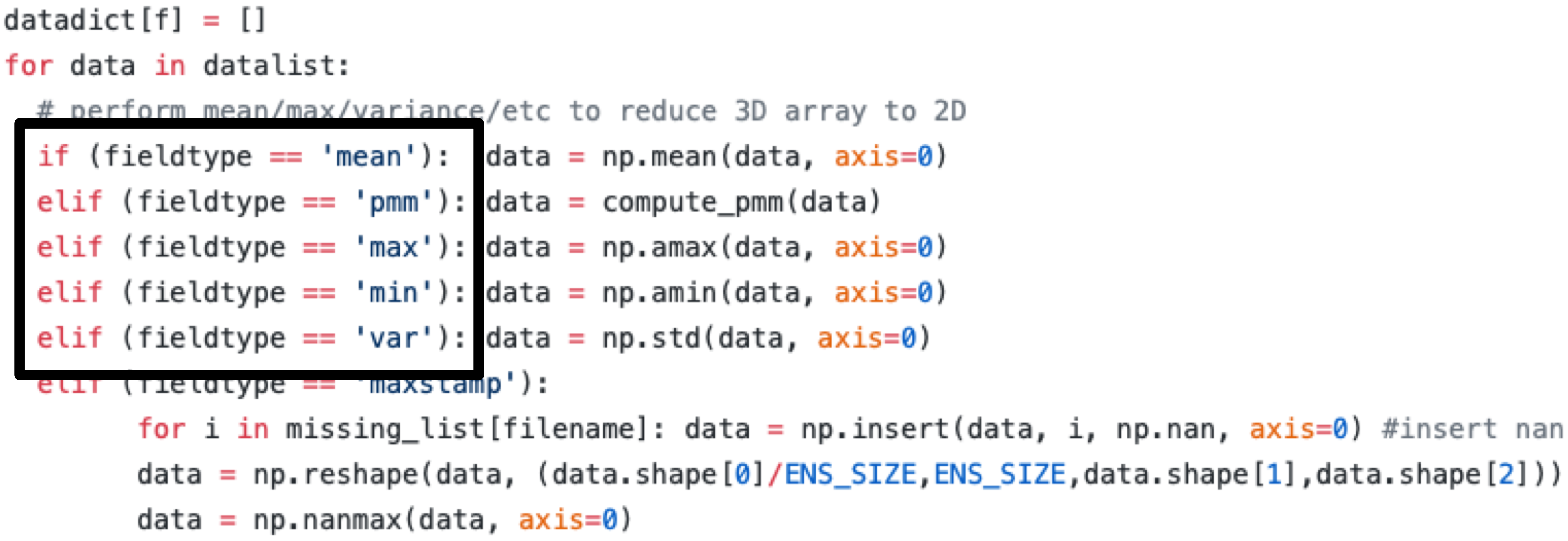
From Python dictionary…

```
'stp’ : {
      'levels’      : [0.5,0.75,1.0,1.5,2.0,3.0,4.0,5.0,6.0,7.0,8.0],
      'cmap’        : readNCLcm('perc2_9lev’),
      'fname’       : ['SBCAPE','LCL_HEIGHT','SR_HELICITY_1KM’,
                       'UBSHR6','VBSHR6’],
      'arraylevel’  : [None,None,None,None,None],
      'filename’    : 'upp’
}
```

Custom compute_stp() function computes STP using fields in ‘fname’

All NCL colormaps available by name

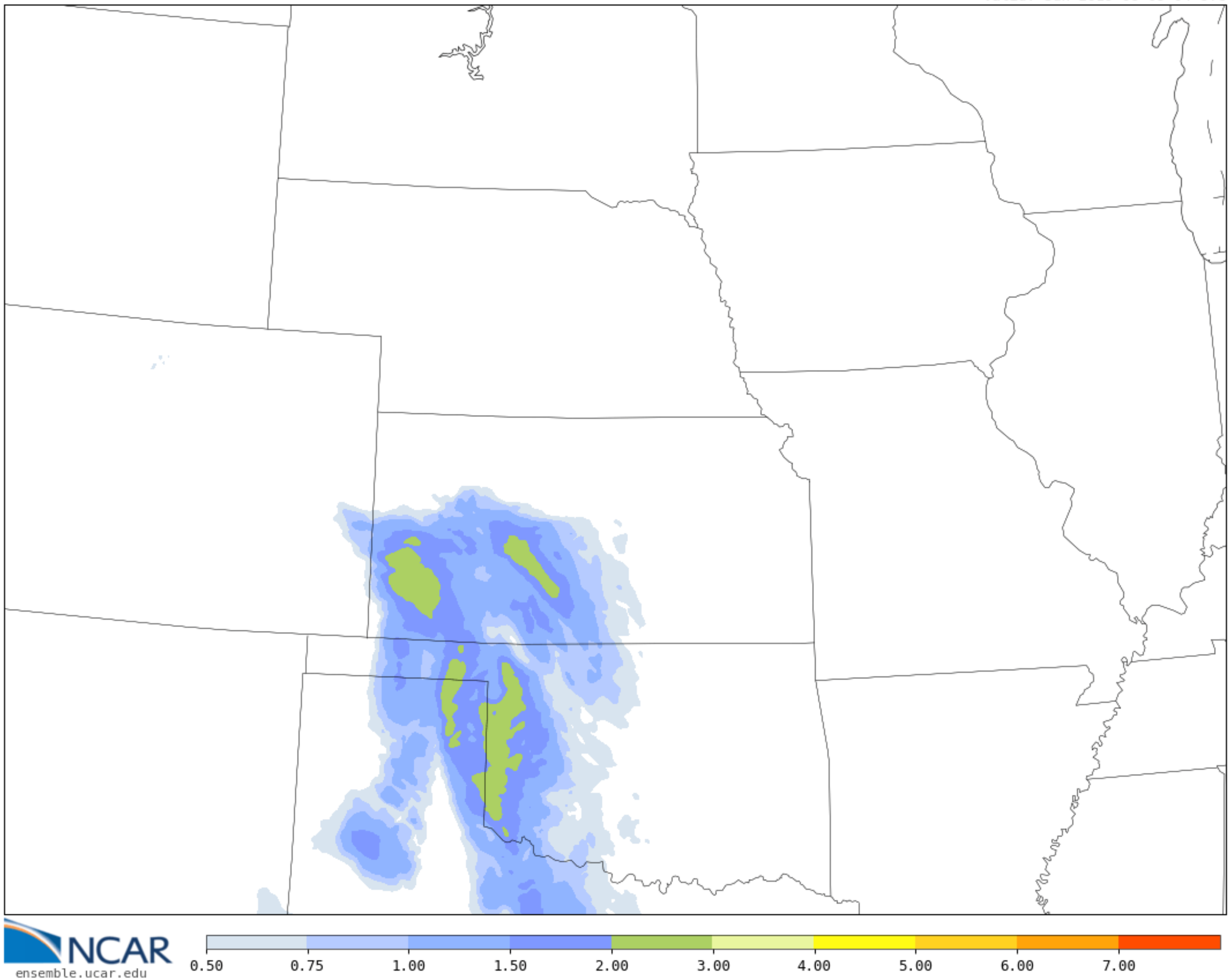
After fields read and derived fields computed, they are collapsed to mean/max/etc…



```
datadict[f] = []
for data in datalist:
  # perform mean/max/variance/etc to reduce 3D array to 2D
  if (fieldtype == 'mean'):  data = np.mean(data, axis=0)
  elif (fieldtype == 'pmm'): data = compute_pmm(data)
  elif (fieldtype == 'max'): data = np.amax(data, axis=0)
  elif (fieldtype == 'min'): data = np.amin(data, axis=0)
  elif (fieldtype == 'var'): data = np.std(data, axis=0)
  elif (fieldtype == 'maxstamp'):
        for i in missing_list[filename]: data = np.insert(data, i, np.nan, axis=0) #insert nan
        data = np.reshape(data, (data.shape[0]/ENS_SIZE,ENS_SIZE,data.shape[1],data.shape[2]))
        data = np.nanmax(data, axis=0)
```

Ensemble mean significant tornado parameter

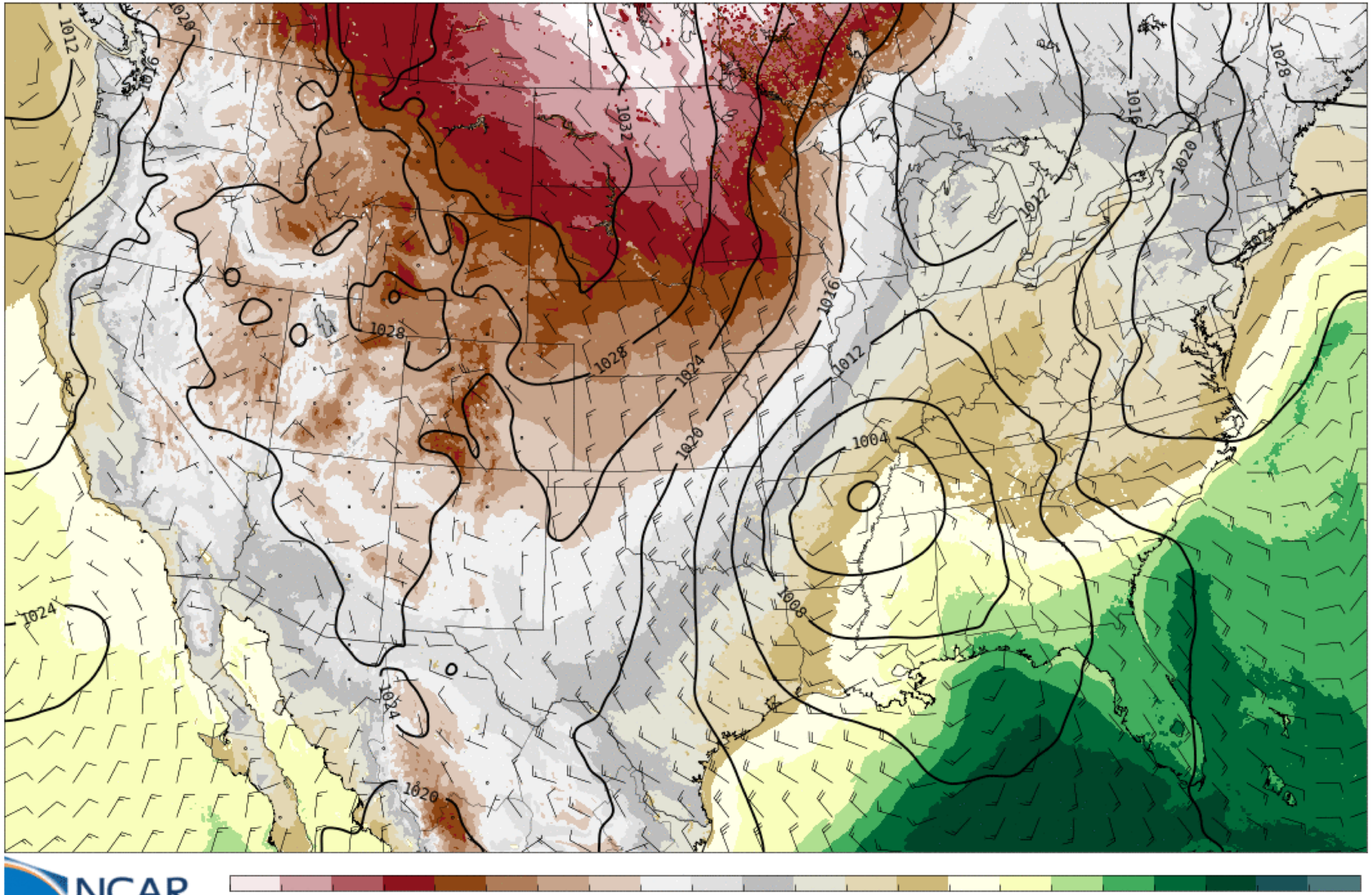
Init: Sat 2019-06-08 00 UTC
Valid: Sun 2019-06-09 04 UTC



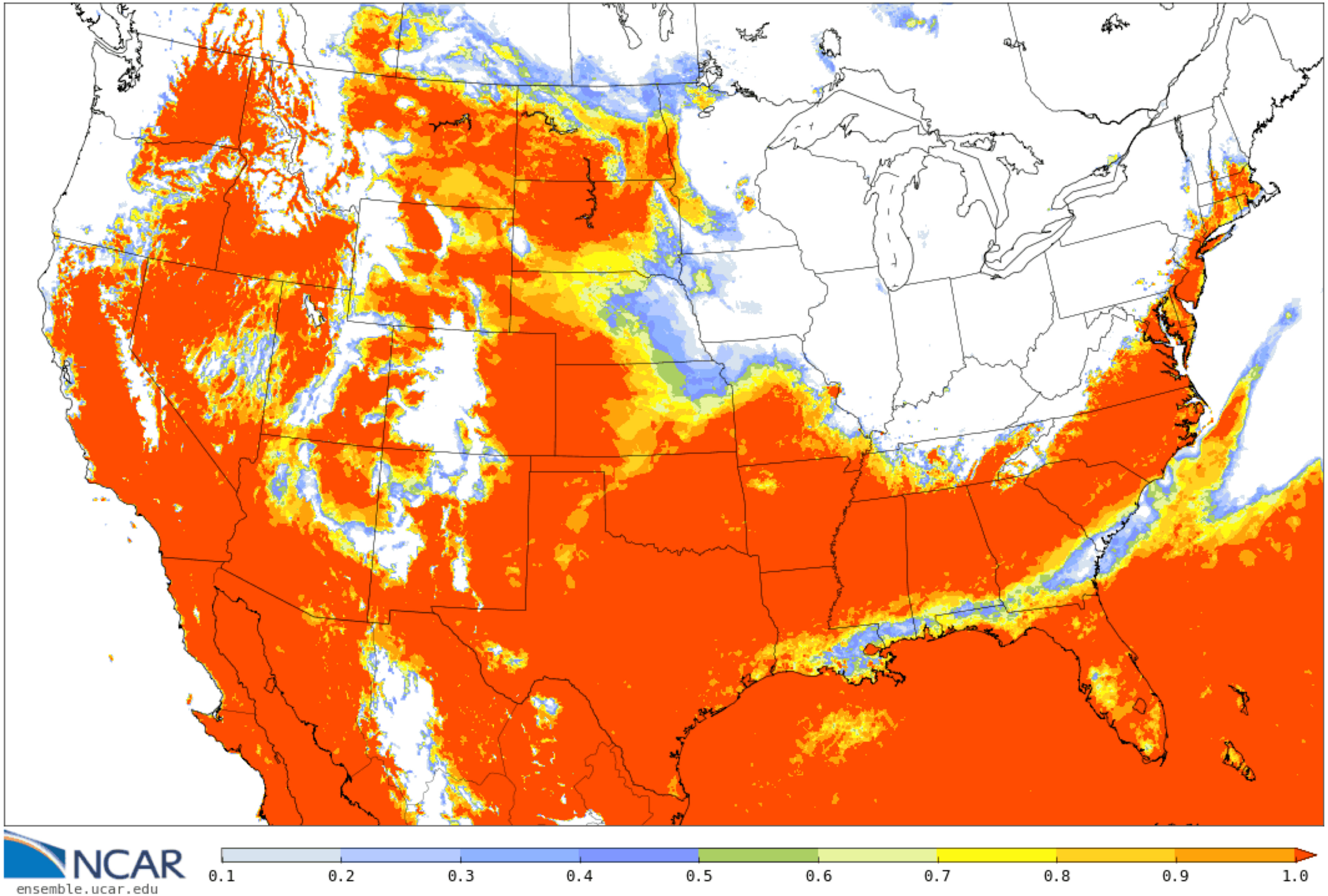
NCAR
ensemble.ucar.edu

Ensemble mean 2-m dewpoint (fill; F), MSLP (contour; hPa), and 10-m wind (kts)

Init: Fri 2016-01-08 00 UTC
Valid: Sat 2016-01-09 16 UTC



NCAR
ensemble.ucar.edu



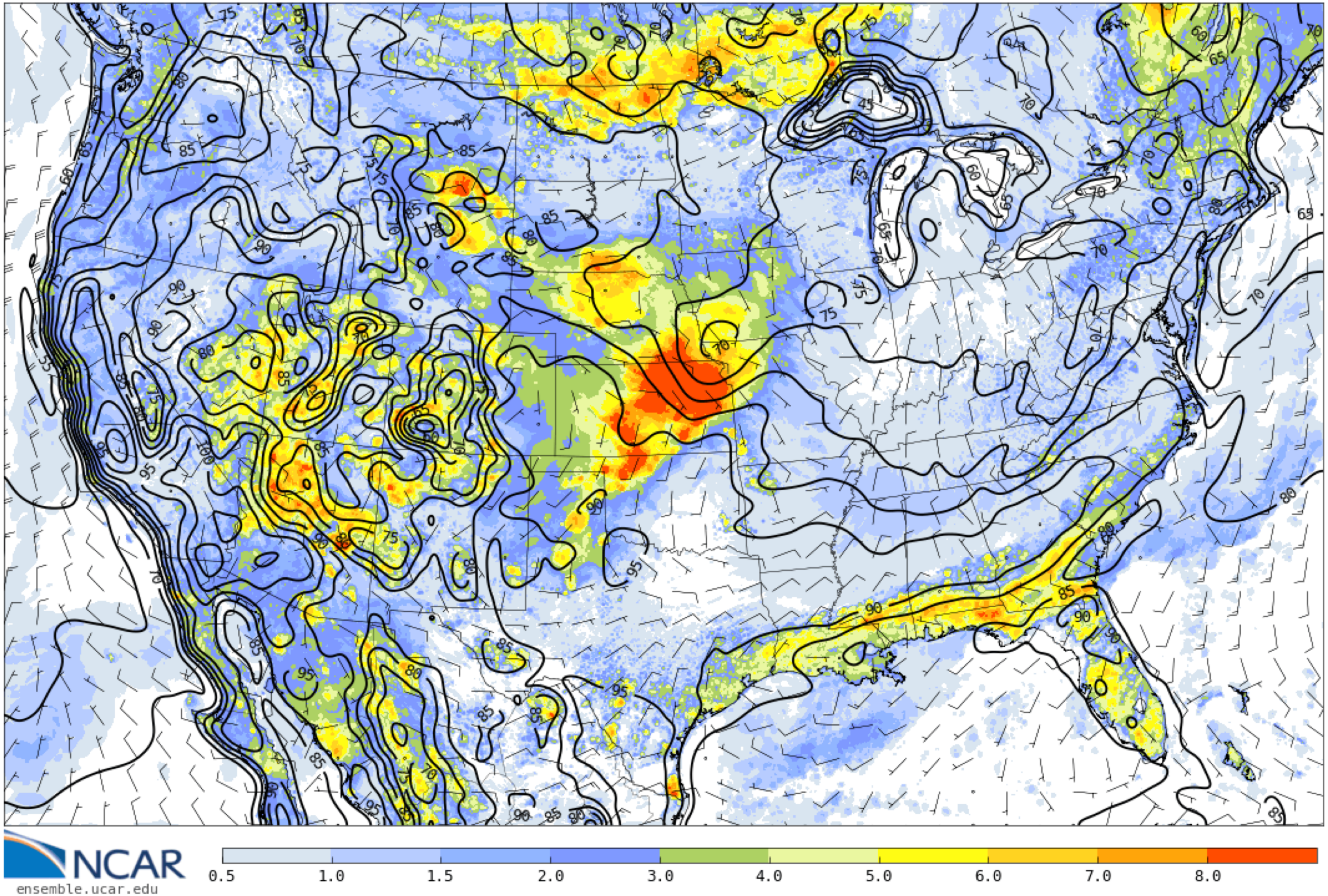
Probability of 2-m temperature > 80F (fill)

Init: Wed 2016-06-29 00 UTC
Valid: Wed 2016-06-29 20 UTC

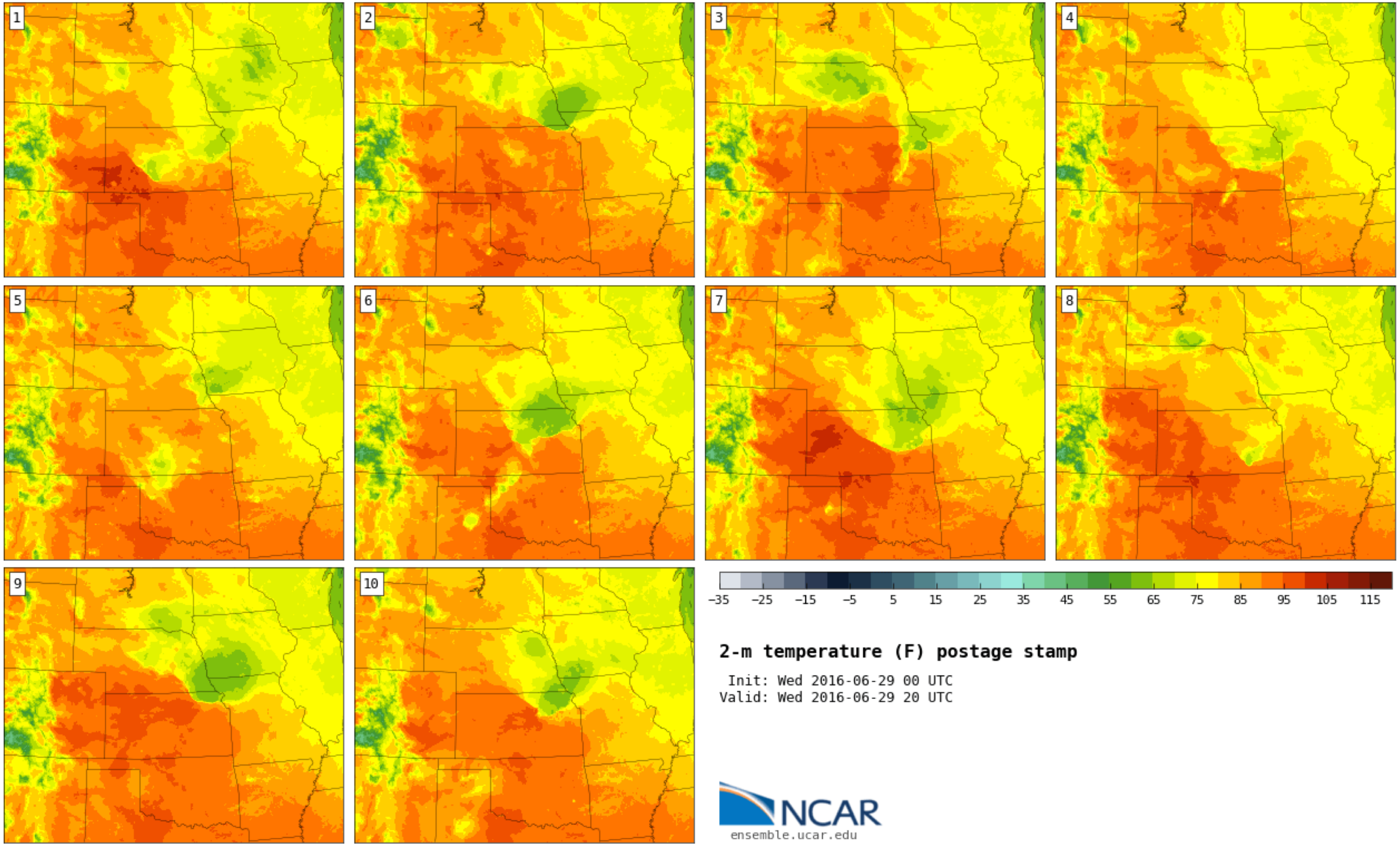
NCAR
ensemble.ucar.edu

Ensemble mean (contour; F) and spread (fill; F) of 2-m temperature, and mean 10-m wind (kts)

Init: Wed 2016-06-29 00 UTC
Valid: Wed 2016-06-29 20 UTC



NCAR
ensemble.ucar.edu



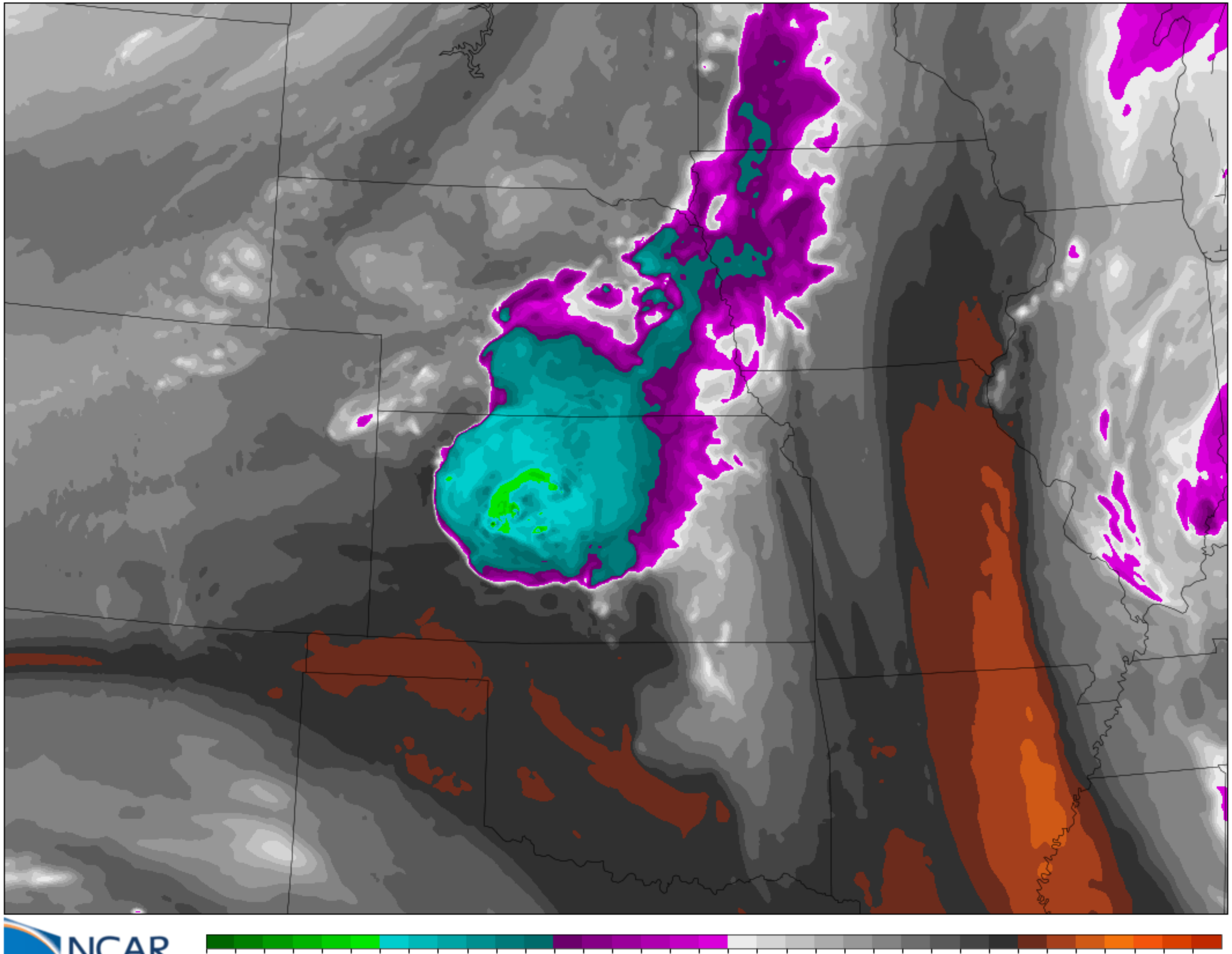
2-m temperature (F) postage stamp

Init: Wed 2016-06-29 00 UTC
Valid: Wed 2016-06-29 20 UTC

NCAR
ensemble.ucar.edu

Member 1 GOES-E Channel 3 (WV) brightness temperature (C)

Init: Sat 2019-06-08 00 UTC
Valid: Sun 2019-06-09 07 UTC



NCAR
ensemble.ucar.edu

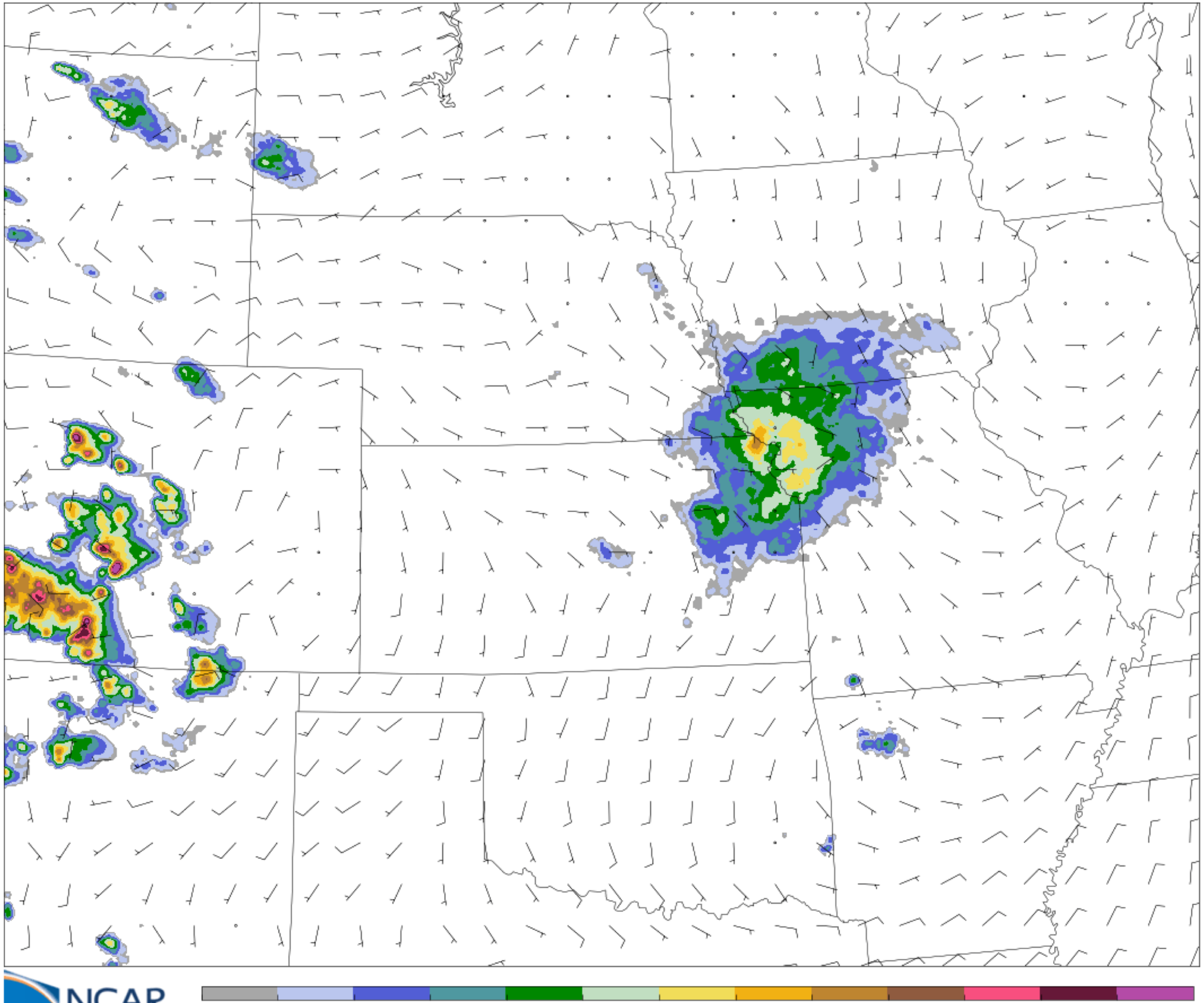
Python-based post-processing system

Custom products for convection-allowing ensembles…

- Probability matched mean products
- Paintball plots (filled contours of convective storms)
- Neighborhood probabilities and filtering/smoothing (Both NEP and NMEP)
- Overlays of MRMS products (e.g., precipitation/reflectivity/rotation tracks)
- Overlays of archived NWS warning shapefiles

Probability matched mean max. column reflectivity and ensemble mean 10-m wind (kts)

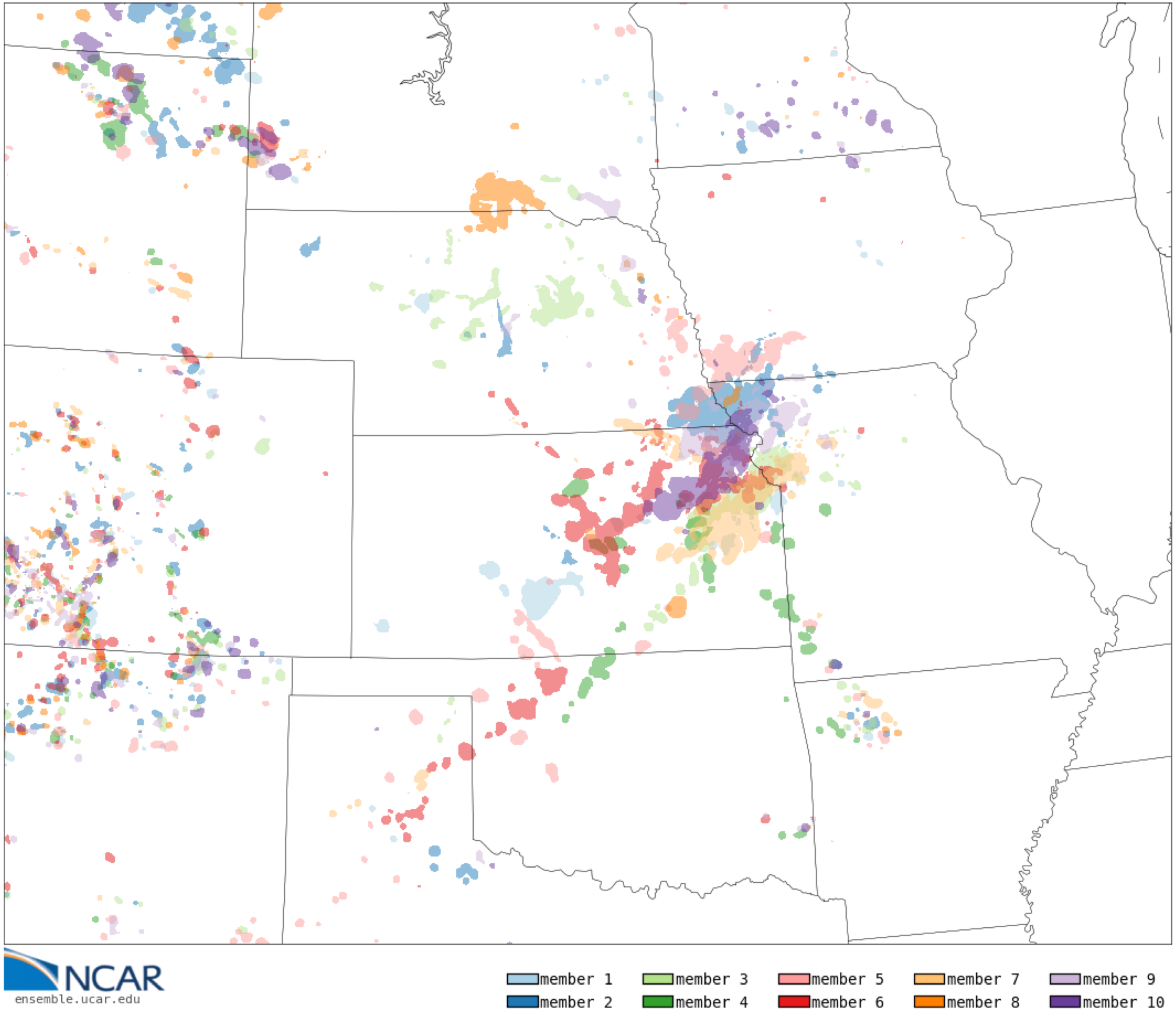
Init: Wed 2016-06-29 00 UTC
Valid: Wed 2016-06-29 20 UTC

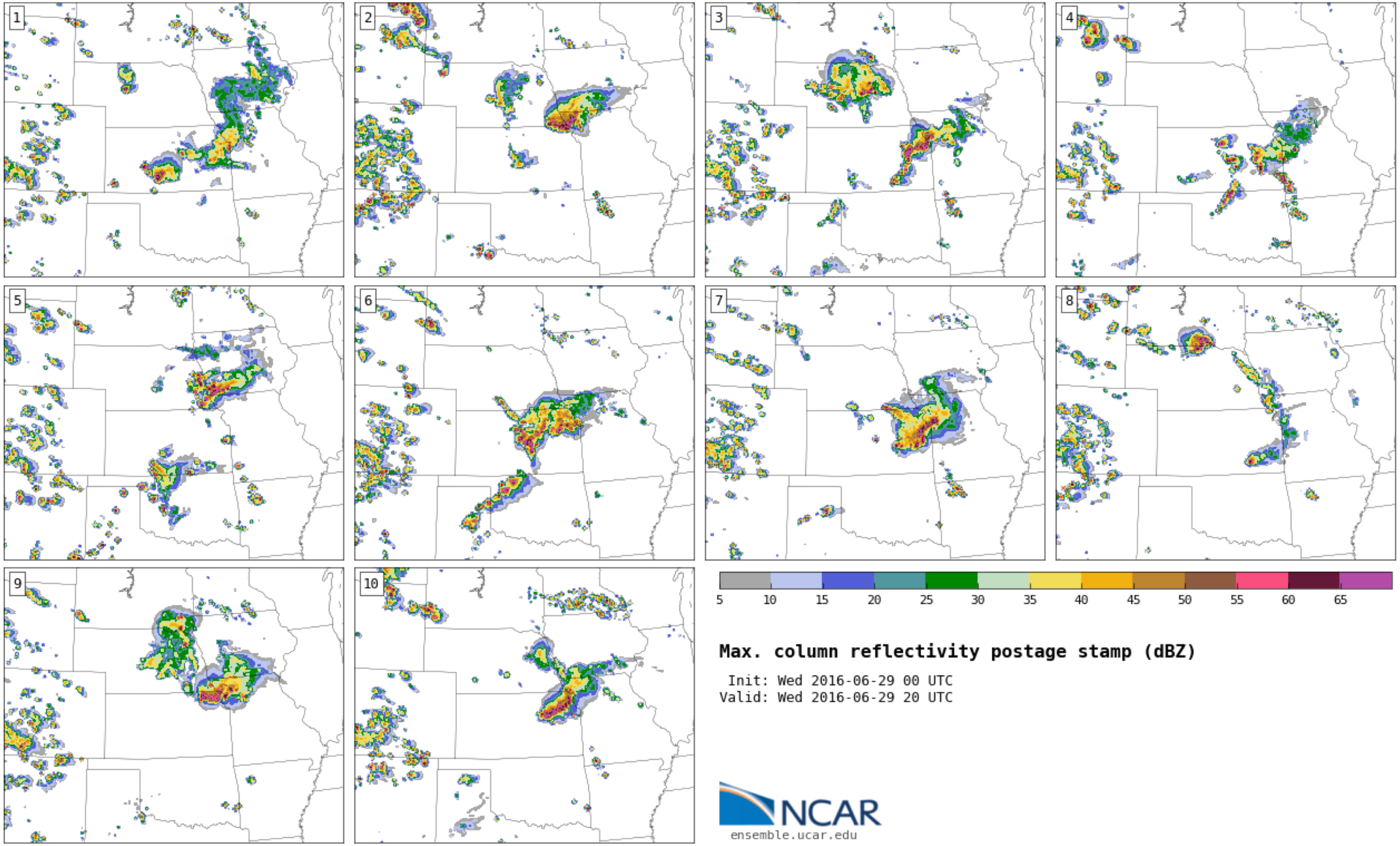


ensemble.ucar.edu

Max. column reflectivity > 40 dBZ from each member

Init: Wed 2016-06-29 00 UTC
Valid: Wed 2016-06-29 20 UTC





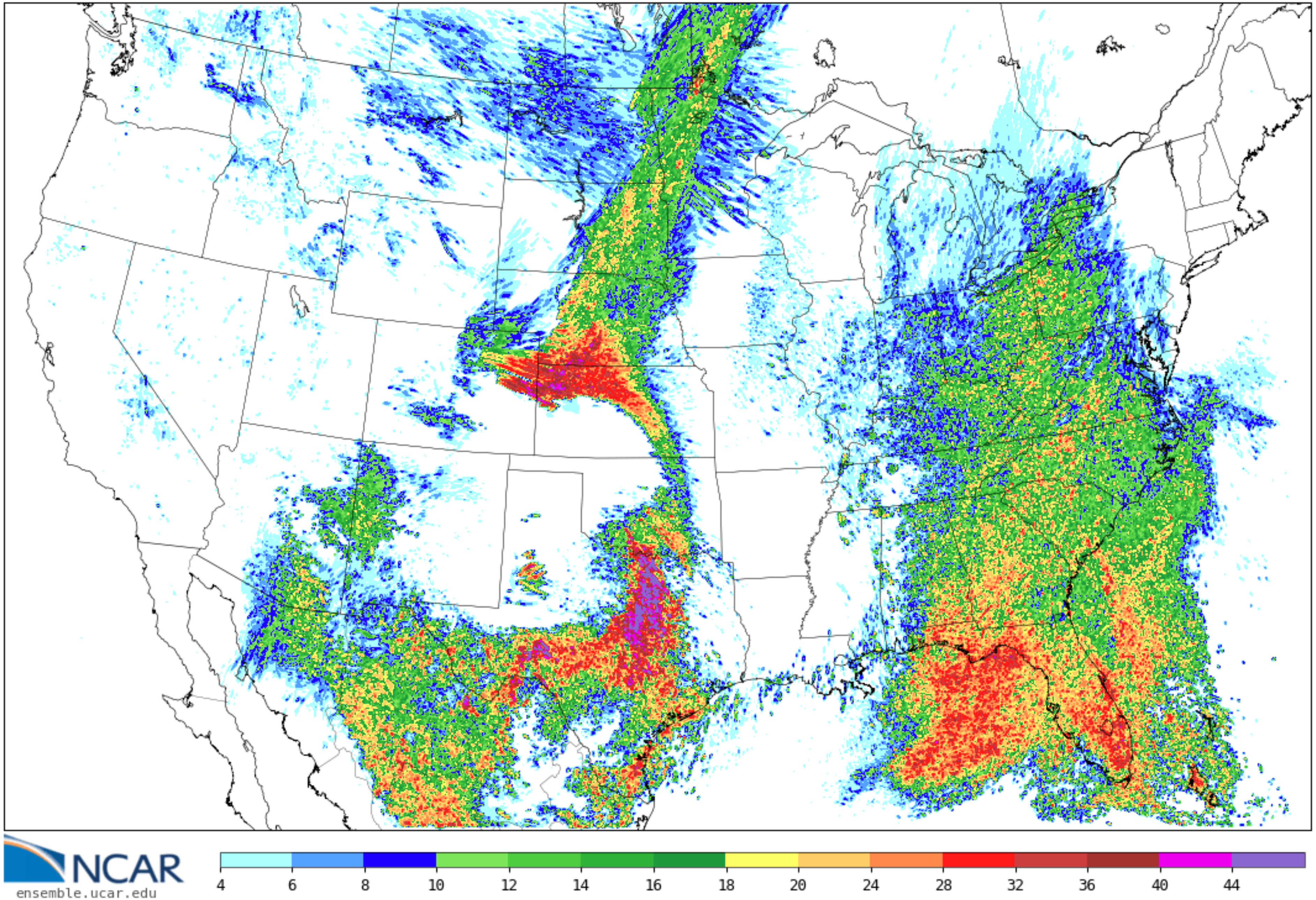
Max. column reflectivity postage stamp (dBZ)

```
 Init: Wed 2016-06-29 00 UTC
Valid: Wed 2016-06-29 20 UTC
```

NCAR
ensemble.ucar.edu

Cumulative ensemble max. updraft speed (m/s)

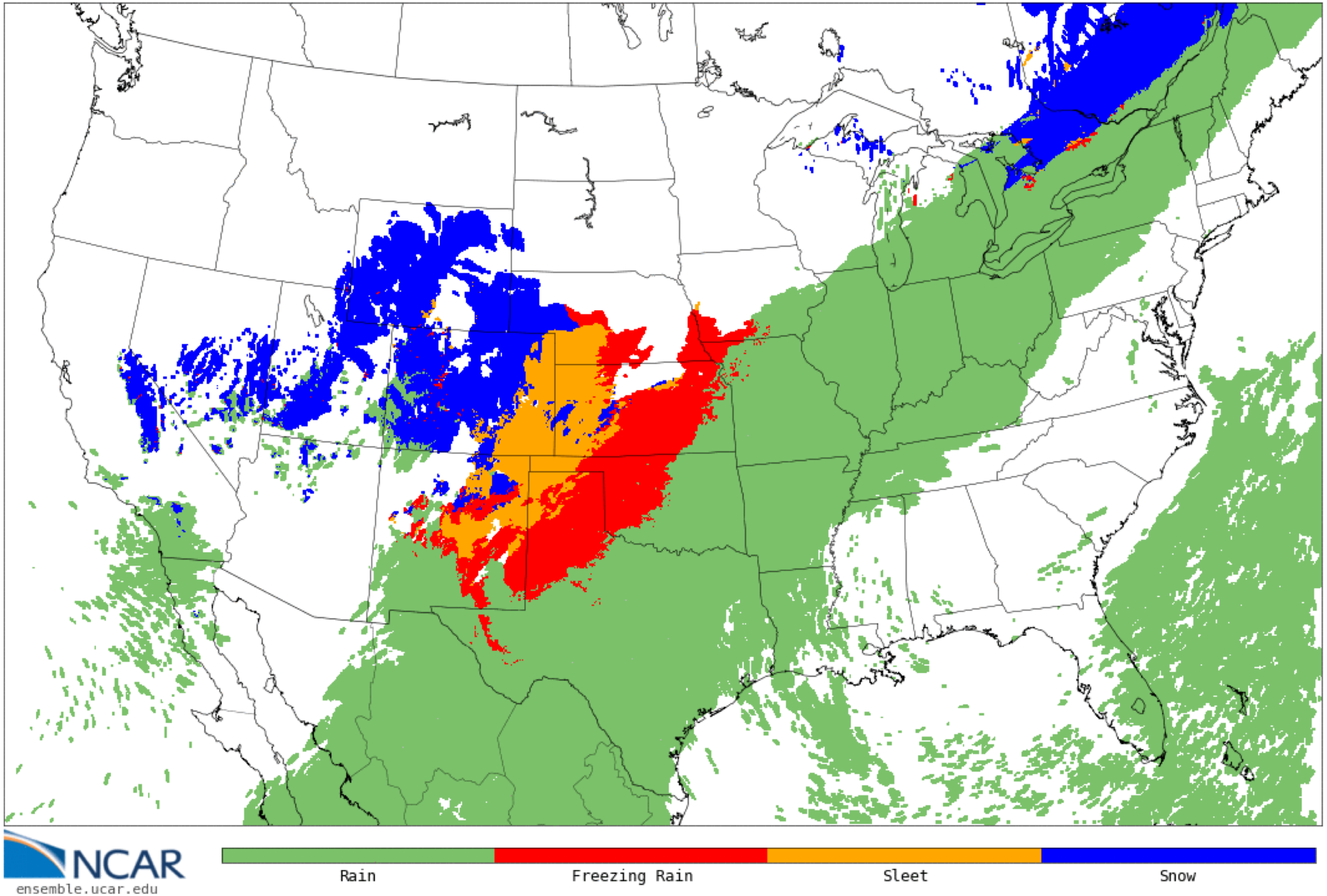
Init: Sun 2019-06-09 00 UTC
Valid: Sun 2019-06-09 00 UTC - Tue 2019-06-11 00 UTC



ensemble.ucar.edu

Dominant 1-hr Precipitation Type (given ens mean precip >= 0.01 in)

Init: Thu 2015-11-26 00 UTC
Valid: Sat 2015-11-28 00 UTC



NCAR
ensemble.ucar.edu

Usage of webplot package with other forecast datasets

- **NCAR WRF Ensemble**

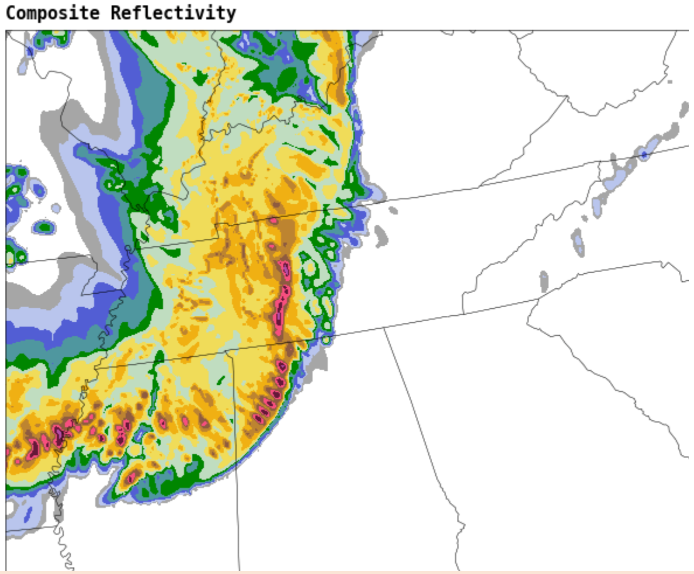
~1000, 10-member, 3-km dx, full-CONUS WRF forecasts. Real-time forecasts.

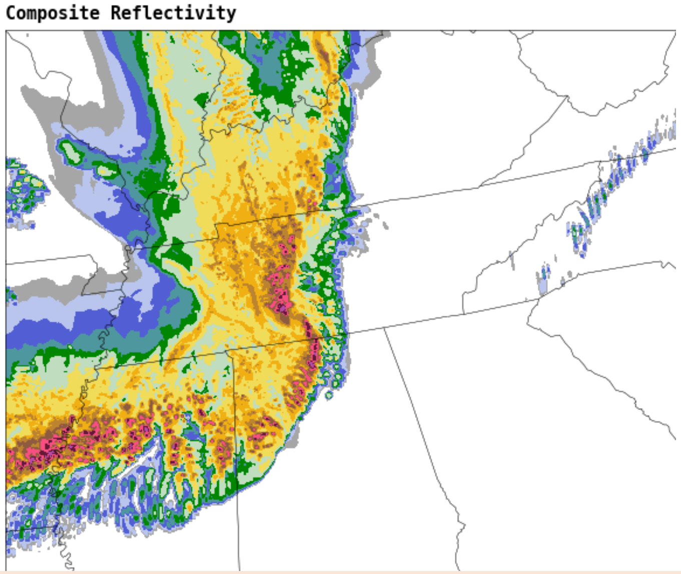
- **WRF grid-spacing sensitivity study**

~500 deterministic, 1-km & 3-km dx, full-CONUS WRF forecasts. Retrospective forecasts.

- **Convection-allowing MPAS ensembles**

35, 10-member, 3-km grid spacing over CONUS (15-km elsewhere). Retrospective forecasts.





Python-based post-processing system

Modifications for MPAS ensemble output

MPAS uses unstructured grid, which presents some challenges when using webplot in its current form.

New capabilities in development:

- MPAS output is often in netCDF4, building in xarray usage (MFDataset doesn’t support netCDF4).
- Modifications to contour function calls for unstructured data (e.g., tri=True).
- Added interpolation functions to place fields on structured grid. Needed for things like two-dimensional smoothing and plotting of thinned wind barbs.

Use of NCAR cheyenne/casper supercomputer

```
qsub –q share –W 00:05 –n 1 –P P123456
        ./make_webplot.py –d=2015080600 -tr=12 -f=t2_var -t=’Ensemble spread 2-m T’
```

Produces a plot for each of 9 regions (CONUS, NE US, SE US, etc.)

Each product (e.g., 2-m temperature mean, spread, max, etc.) run as a separate job, submitted by a master script.

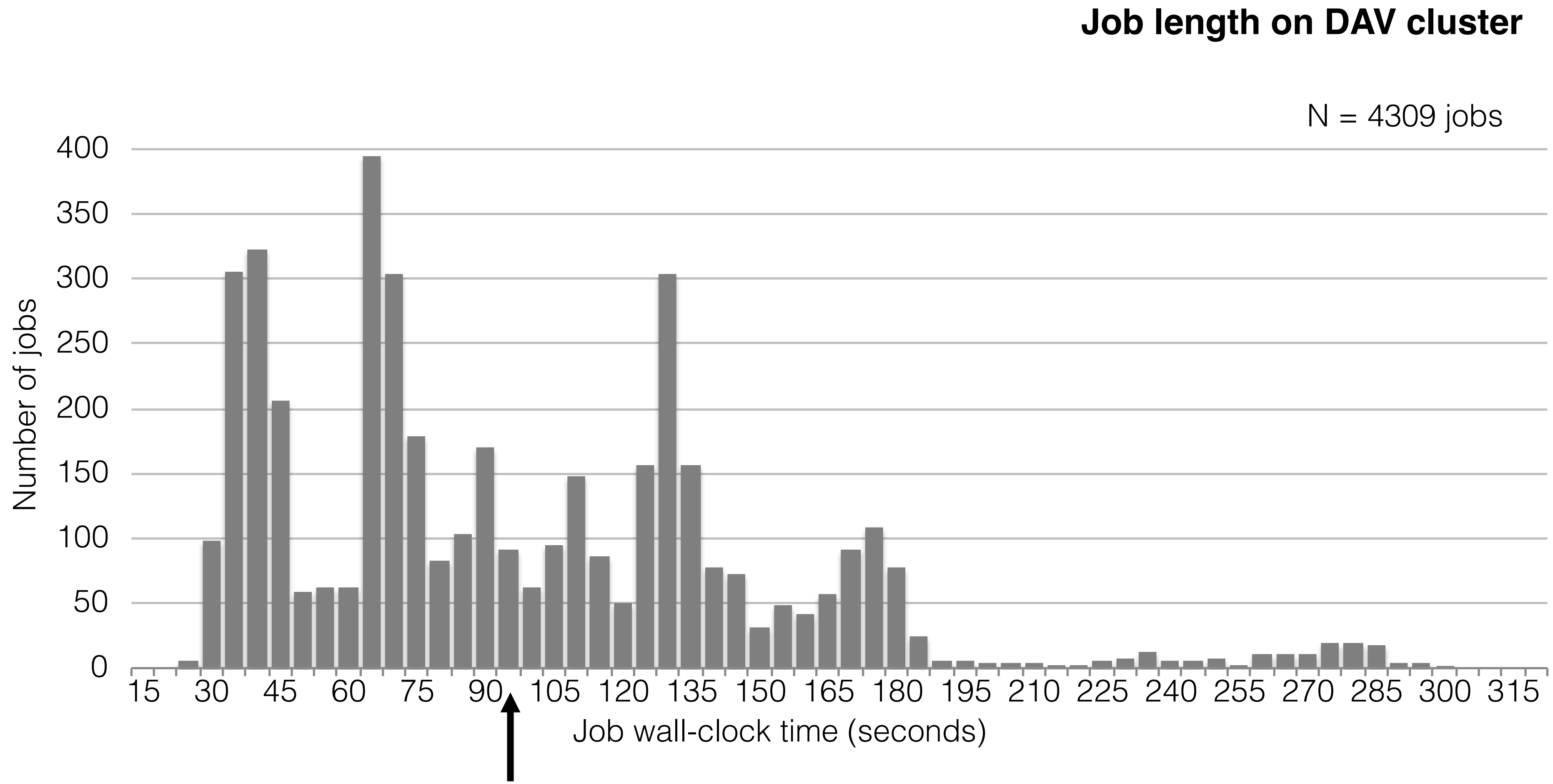
System can handle a lot of products run in real-time…

~200 individual products for each forecast hour, spread out over 4 hr period

~5,000 jobs per day on DAV cluster (many jobs grouped together)

~120,000 graphics per forecast (for 60-hr forecast)

Use of DAV cluster



Mean job length ~ 95 seconds

Computing convective diagnostics inline within WRF/MPAS

Rather than using post-processing tools to compute convective-scale diagnostics, much more efficient to compute in WRF.

Takes advantage of WRF parallelism and in-memory arrays.

Using WRF build with UPP diagnostics computed in module_diag_misc.F

- CAPE / CIN for various parcels
- Shear over variety of layers
- Storm-relative helicity for variety of layers
- Bunkers storm motion estimates

Future developments

- More efficient I/O for similar products (e.g., 2-m temperature mean/max/spread).
- Replace netcdf4-python and basemap with xarray and cartopy.
- Improvements for handling MPAS data (can be slow for fine meshes).
- Support for Python 3.
- Ability to download and plot NCEP convection-allowing guidance (HRRR/HREF) in real-time.

Early version available on github: https://github.com/rsobash/webplot

Examples of graphics: http://ensemble.ucar.edu