Science_Flight_20151210

December 10, 2015

In this report, we'll review the science flight of the ER-2 starting on 12/10/2015. This was a ~6-h flight that sampled post-frontal convective showers First, let's import all the needed modules and ingest and process the raw data.

```
In [6]: from __future__ import print_function
        import numpy as np
        import matplotlib.pyplot as plt
        import datetime as dt
        import os
        import glob
        import pyart
        import rawpyampr
        import pyampr
        import dualpol
        from copy import deepcopy
        from IPython.display import Image
        from awot.graph.common import create_basemap
        from awot.graph.flight_level import FlightLevel
        from pyart_tools import (
            plot_list_of_fields, list_nexrad_files, read_nexrad_aws)
        %matplotlib inline
```

```
In [7]: import warnings
        warnings.filterwarnings('ignore')
        def delete_file(fname):
            try:
                os.remove(fname)
            except:
                pass
```

```
In [8]: datadir = './'
        files = glob.glob(datadir + '*.dat')
        print(files)
        fname = os.path.basename(files[0])[:-4]
```

```
['./AMPR-20151210-132821.dat', './AMPR-20151210-155224.dat', './AMPR-20151210-202159.dat']
```

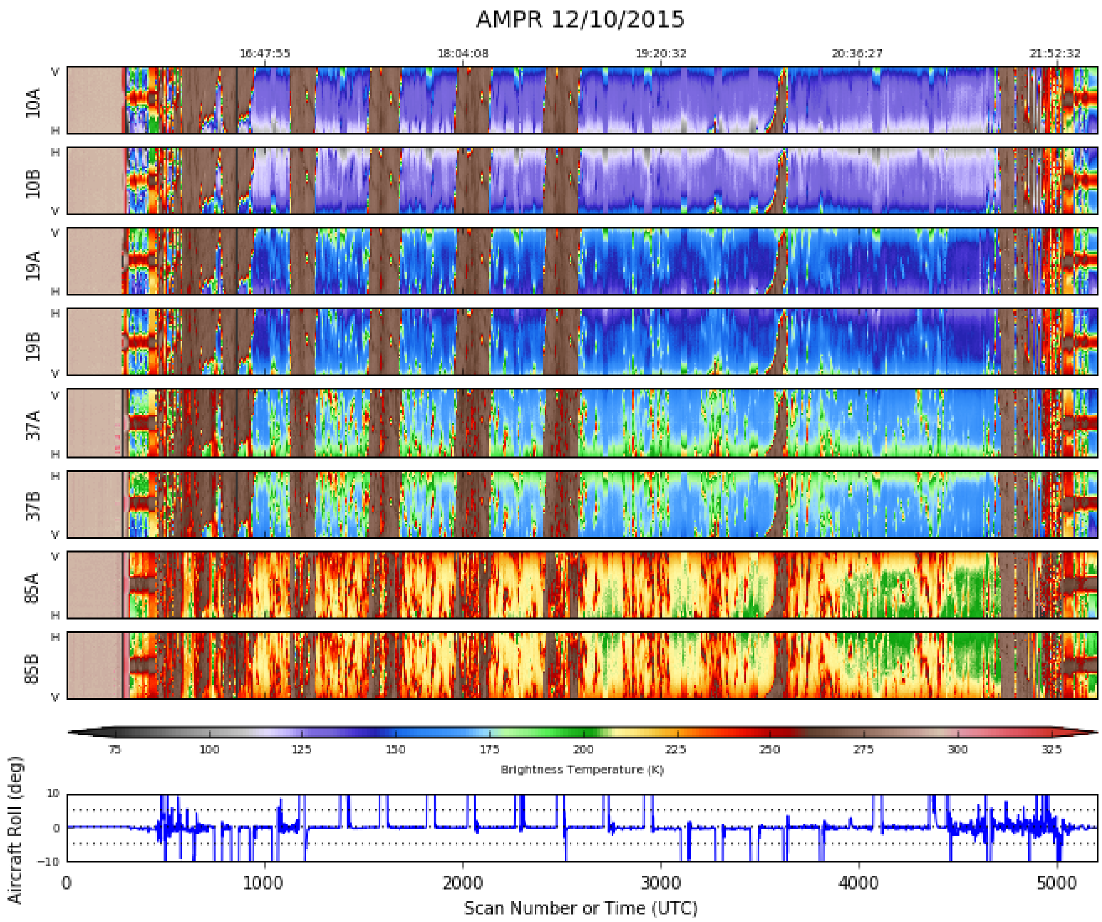
```
In [9]: payload = rawpyampr.ampr_payload.AMPR_Payload(files)
        l1file = fname + '_L1.nc'
        l2file = fname + '_L2.nc'
        delete_file(l1file)
        payload.writeLevel1B(l1file)
        L1B = rawpyampr.ampr_level1b.AMPR_QC(l1file)
        delete_file(l2file)
        L1B.writeLevel2B(l2file)
```

```
All of file: ./AMPR-20151210-132821.dat Read Successfully
End of data stream reached
All of file: ./AMPR-20151210-155224.dat Read Successfully
End of data stream reached
All of file: ./AMPR-20151210-202159.dat Read Successfully
End of data stream reached
Interpreting Navigation Records as: IWG1
 No navigation file found
Navigating pixels using internal recording of nav data.
Number points to converge: 4
Writing to output file:  AMPR-20151210-132821_L1.nc
Found Navigation Data!
Writing to output file: AMPR-20151210-132821_L2.nc
File containing water fraction not on path
```

Now we are ready to read in and display the L2 geolocated brightness temperatures.

```
In [10]: data = pyampr.AmprTb(l2file)
         data.plot_ampr_channels()
```

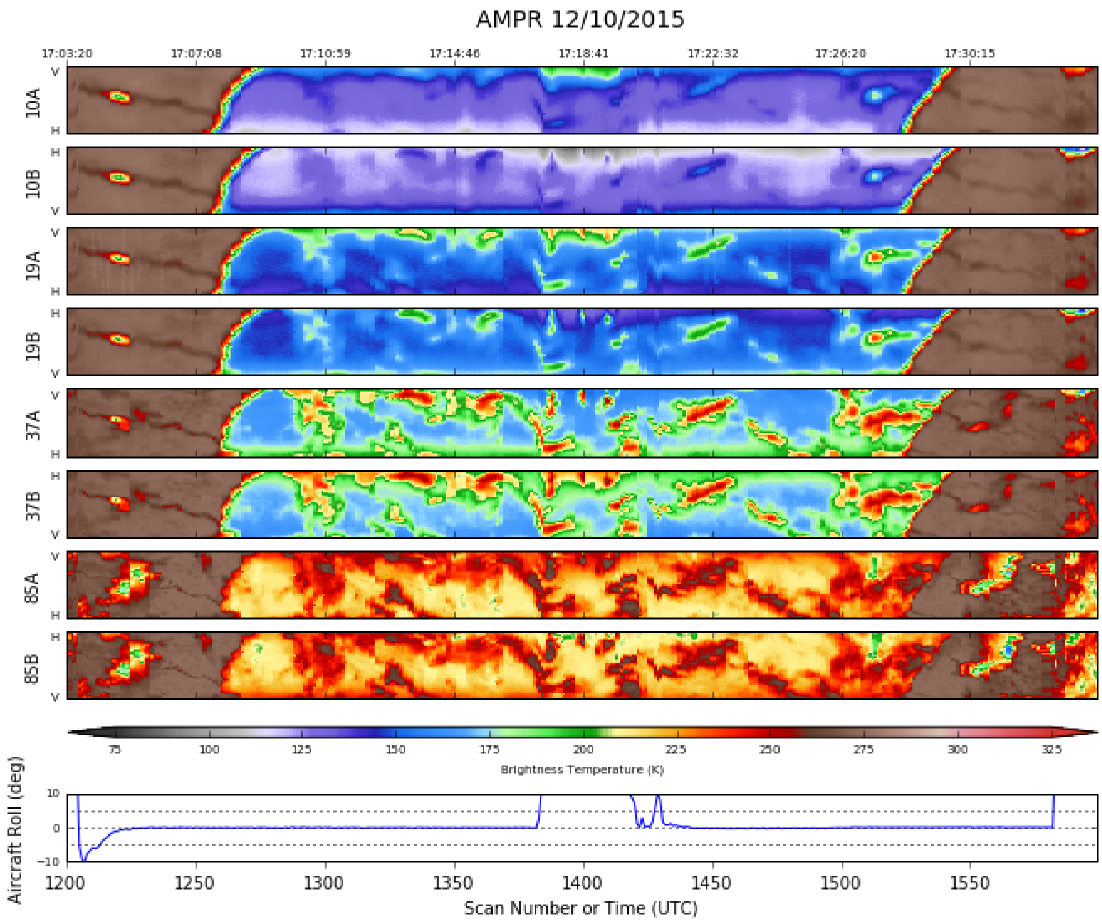
```
********************
read_ampr_tb_level2b(): Reading AMPR-20151210-132821_L2.nc
Assuming IPHEX data structure.
Change to proper project if incorrect, otherwise errors will occur.
Currently available field projects: IPHEX, MC3E, TC4, TCSP, JAX90, COARE,
CAMEX1, CAMEX2, CAMEX3, CAMEX4, TRMMLBA, KWAJEX, TEFLUNA, FIRE3ACE, CAPE
Default: project = 'IPHEX'
Found Navigation Data!
(5204,)
********************


********************
plot_ampr_channels():
Available scans = 1 to 5204
Available times = 13:28:33 - 22:08:14
********************
```



We finally got the 19 GHz channels back for this flight! And they worked well! Kudos to AMPR engineer Dave Simmons for doing the necessary surgery to get parts replaced. The other channels behaved well too. The early part of the chart consists of engineering test data, which is why the channels look so uniform before the flight. Let's focus on the science. The ER-2, after gaining altitude, performed mutliple overflights of post-frontal convective showers. Let's examine one of the orbits.

```
In [12]: data.plot_ampr_channels(scanrange=[1200, 1600])
```

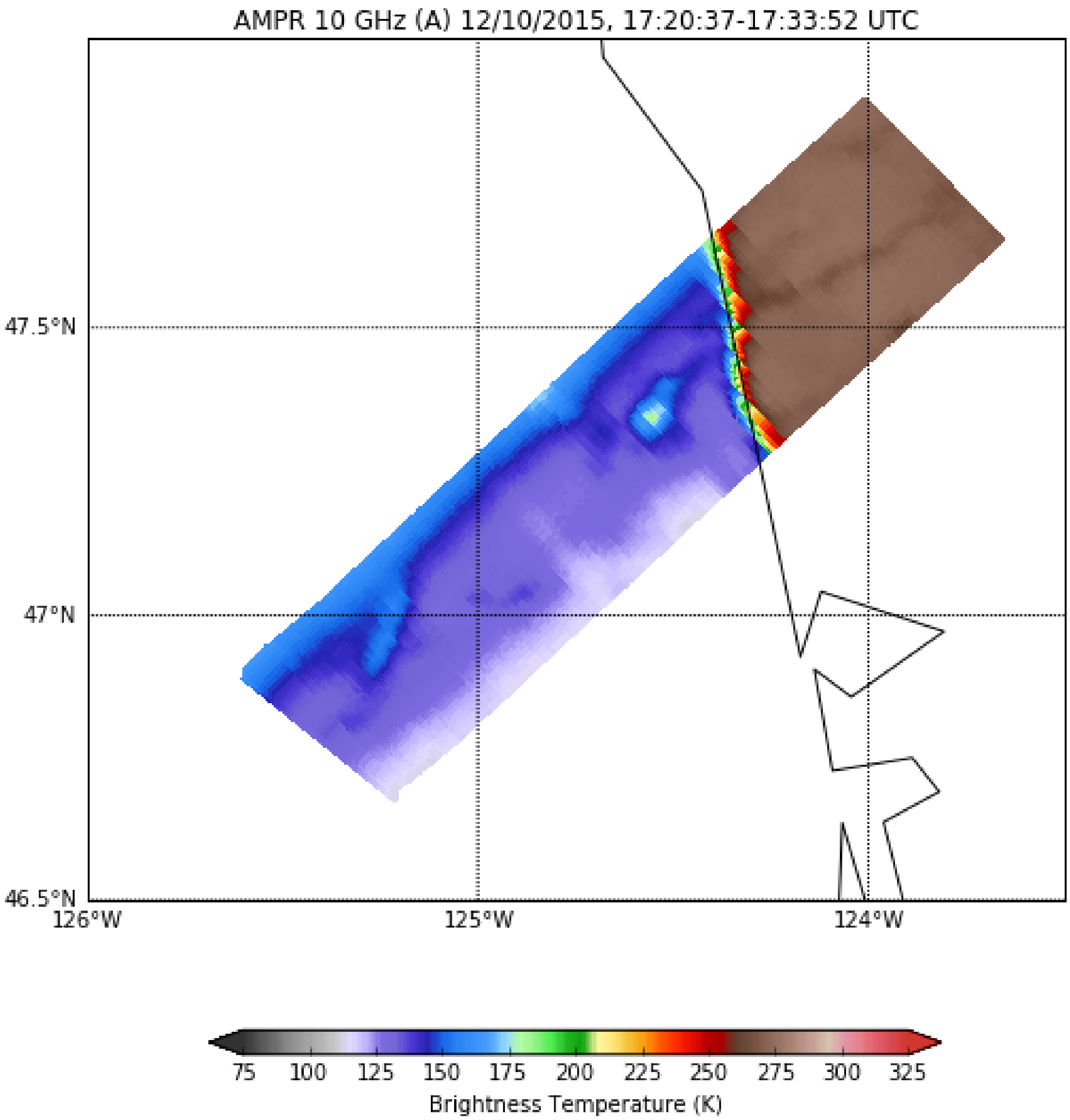
```
********************
plot_ampr_channels():
Available scans = 1 to 5204
Available times = 13:28:33 - 22:08:14
********************
```



It's nice to see 19 GHz back! There was a lot of convection this day, especially early in the flight. Some of it featured strong emission even down to 10 GHz. Let's look at the water-to-onshore portion of this orbit, and check out that cell with really strong 10-GHz emission just after scan 1500. I'm working with an older version of PyAMPR on this computer, so the plots won't be as fancy.

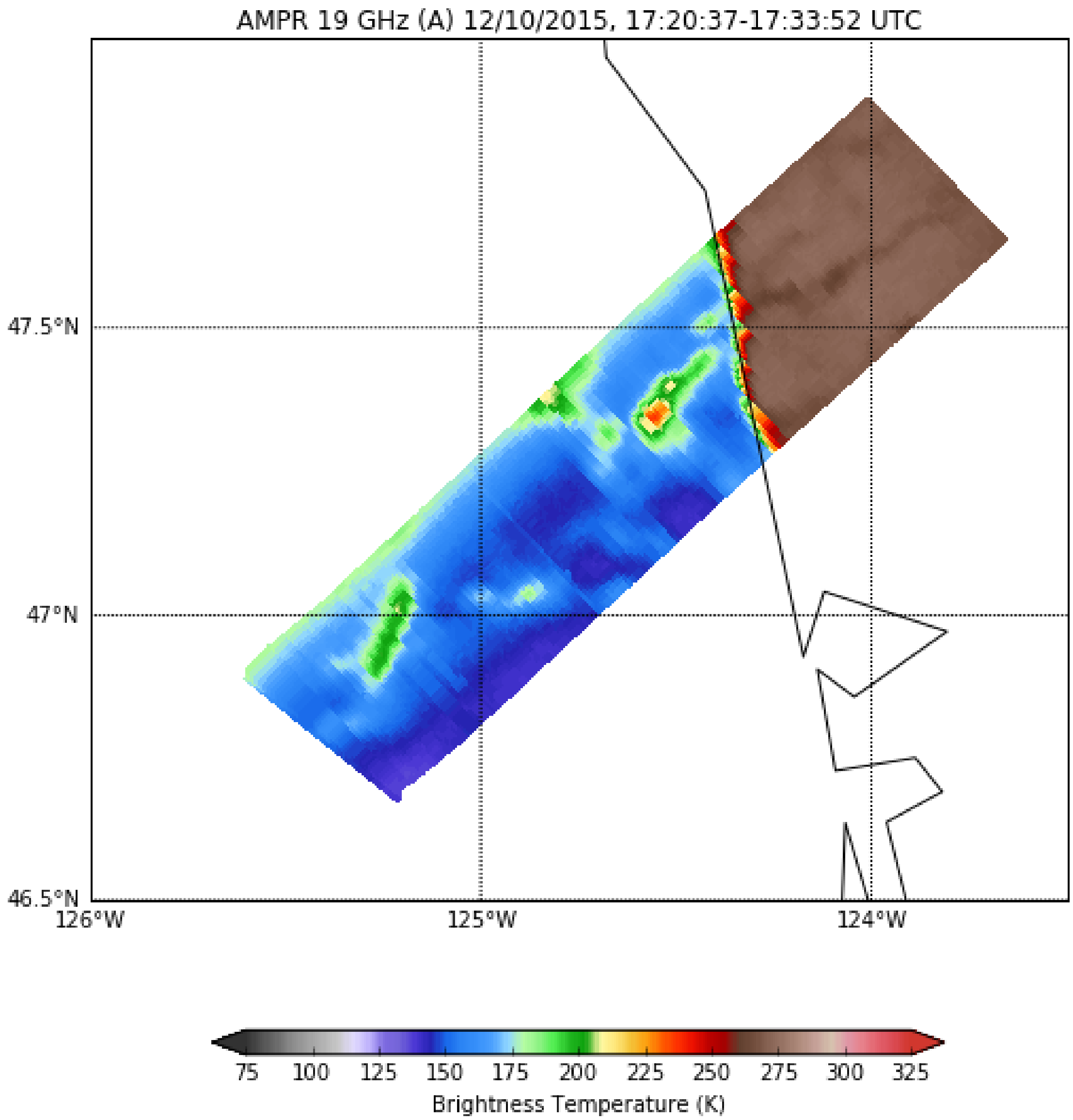
```
In [28]: display = data.plot_ampr_track(
             '10a', scanrange=[1425, 1600], maneuver=False, return_flag=True,
             meridians=1, parallels=0.5, show_grid=True, latrange=[46.5, 48],
             lonrange=[-123.5, -126])
```

```
********************
plot_ampr_track():
Available scans = 1 to 5204
Available times = 13:28:33 - 22:08:14
Filtering out significant aircraft maneuvers
********************
```

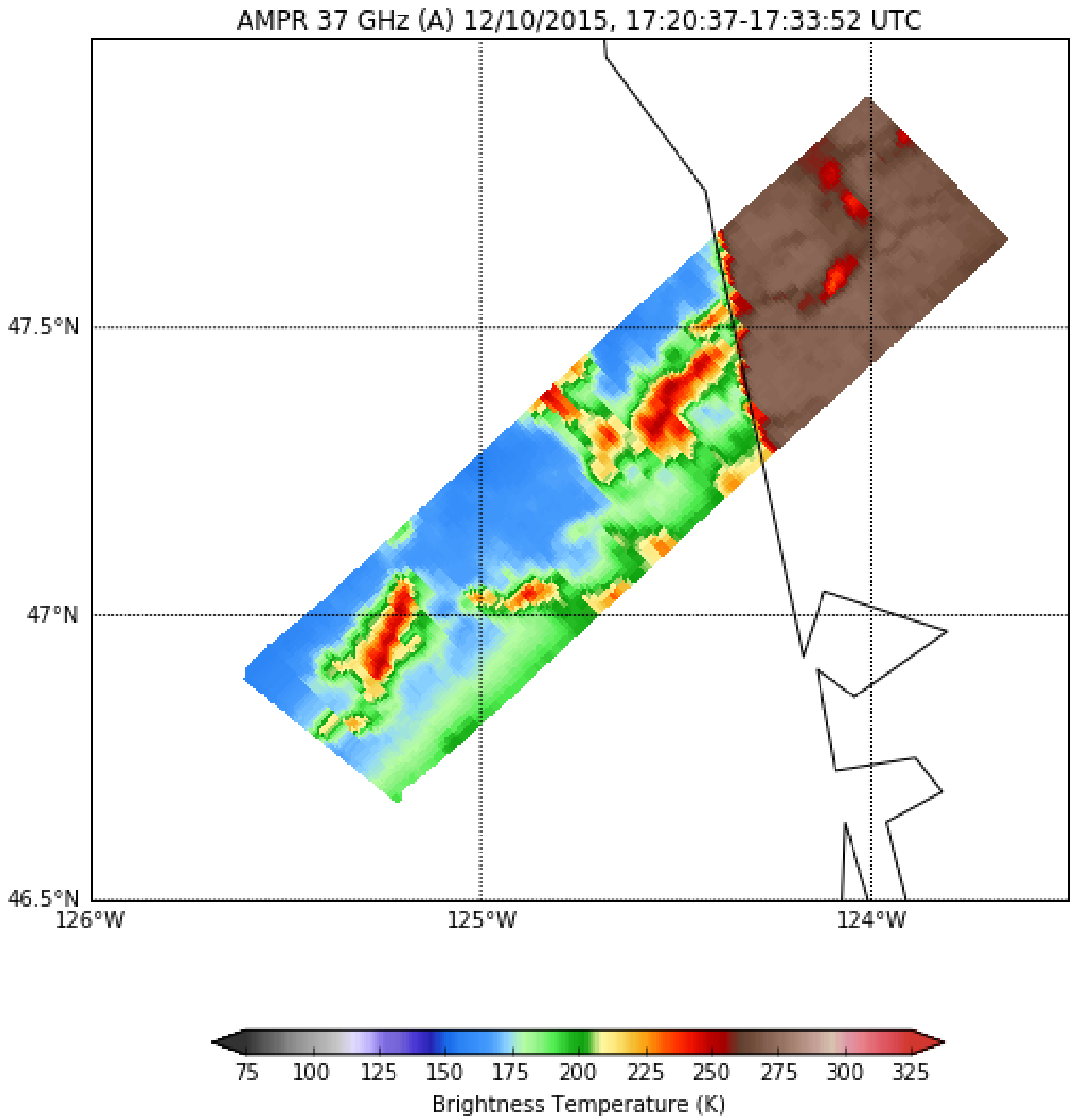



```
In [27]: display = data.plot_ampr_track(
             '19a', scanrange=[1425, 1600], maneuver=False, return_flag=True,
             meridians=1, parallels=0.5, show_grid=True, latrange=[46.5, 48],
             lonrange=[-123.5, -126])
```

```
********************
plot_ampr_track():
Available scans = 1 to 5204
Available times = 13:28:33 - 22:08:14
Filtering out significant aircraft maneuvers
********************
```

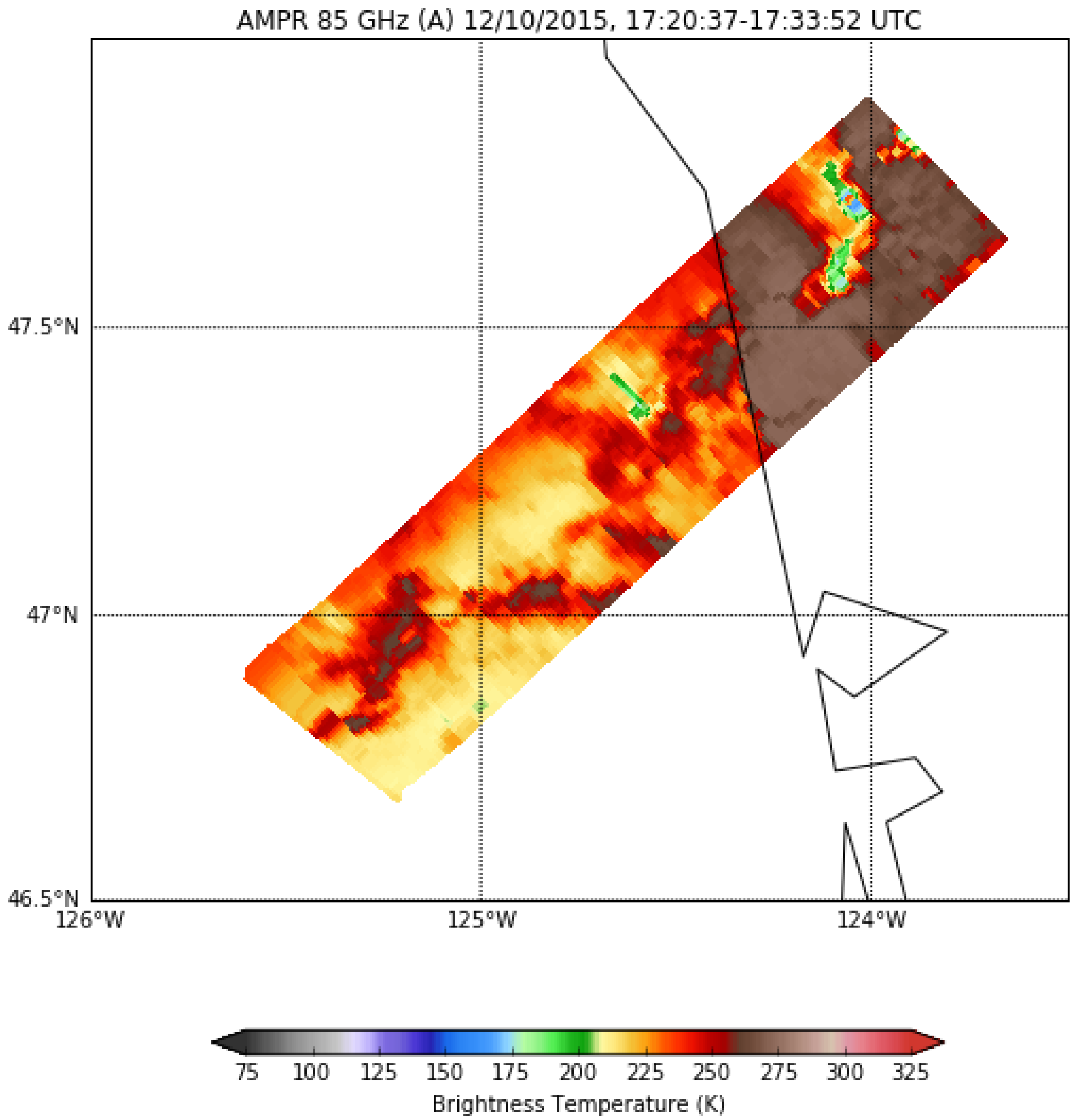



```
In [26]: display = data.plot_ampr_track(
             '37a', scanrange=[1425, 1600], maneuver=False, return_flag=True,
             meridians=1, parallels=0.5, show_grid=True, latrange=[46.5, 48],
             lonrange=[-123.5, -126])

********************
plot_ampr_track():
Available scans = 1 to 5204
Available times = 13:28:33 - 22:08:14
Filtering out significant aircraft maneuvers
********************
```




```
In [25]: display = data.plot_ampr_track(
             '85a', scanrange=[1425, 1600], maneuver=False, return_flag=True,
             meridians=1, parallels=0.5, show_grid=True, latrange=[46.5, 48],
             lonrange=[-123.5, -126])
```

```
********************
plot_ampr_track():
Available scans = 1 to 5204
Available times = 13:28:33 - 22:08:14
Filtering out significant aircraft maneuvers
********************
```




```
In [14]: # Let's grab NEXRAD data for this case
         radlist = list_nexrad_files(2015, month='12', day='10', station='KLGX')
         # print(radlist)
         radar = read_nexrad_aws('KLGX20151210_172841_V06.gz')
```

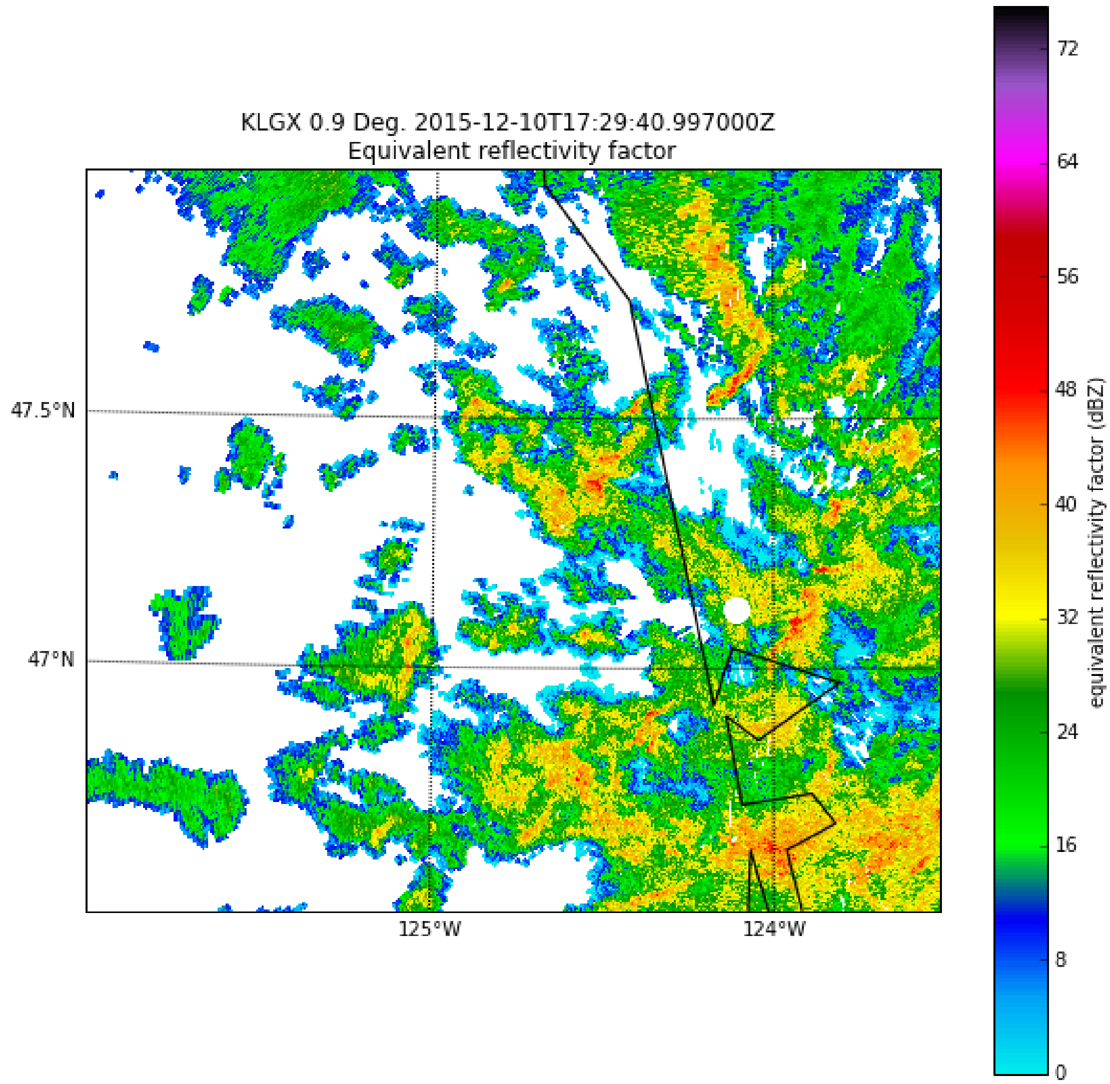
https://noaa-nexrad-level2.s3.amazonaws.com/2015/12/10/KLGX/KLGX20151210_172841_V06.gz

```
In [15]: # Do some quick QC on the NEXRAD data
         zc = deepcopy(radar.fields['REF']['data'])
         radar.add_field_like('REF', 'ZC', zc, replace_existing=True)
         retrieve = dualpol.DualPolRetrieval(
             radar, dz='ZC', dr='ZDR', dp='PHI', rh='RHO', use_temp=False,
             dsd_flag=False, fhc_flag=False, precip_flag=False,
             liquid_ice_flag=False, kdp_window=5, verbose=False,
```

```
              ice_flag=False, qc_flag=True, gs=250.0,
              thresh_sdp=20.0, speckle=3,
              thresh_dr=np.array(dualpol.DEFAULT_DR_THRESH)+1.0)
```

```
1.07802987099 seconds to run csu_kdp
No sounding provided
```

```
In [38]: # And now let's plot the KLGX radar data
         fig = plt.figure(figsize=(10, 10))
         display = pyart.graph.RadarMapDisplay(radar)
         display.plot_ppi_map('ZC', 1, vmin=0, vmax=75, cmap='pyart_NWSRef', min_lon=-126,
                              max_lon=-123.5, min_lat=46.5, max_lat=48,
                              resolution='l', lat_lines=[47, 47.5], lon_lines=[-125, -124])
```

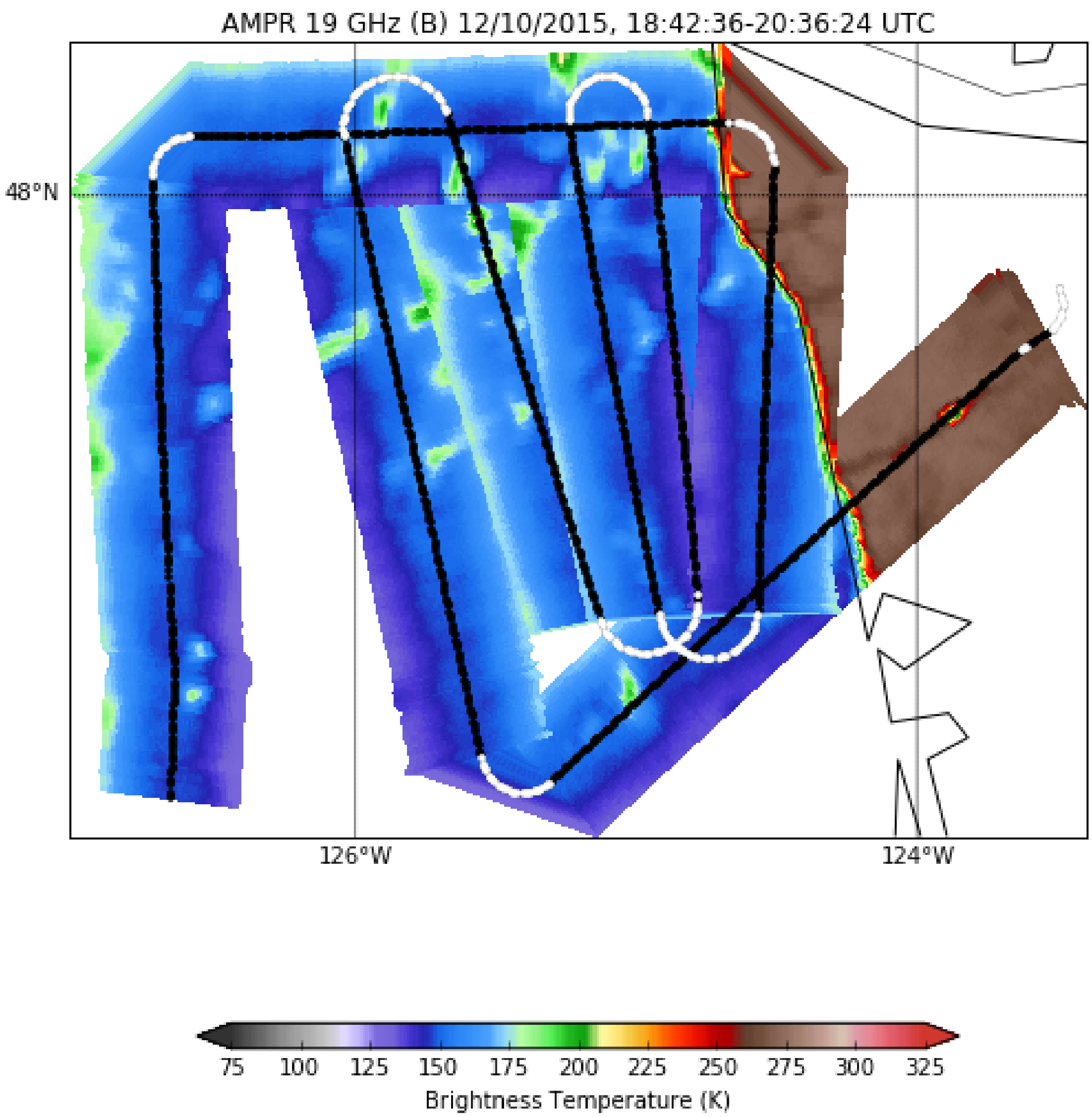


Note the ice scattering at 85 GHz in the angled precipitation system just onshore. Lightning was occasionally observed in the cells on this day, so the ice scattering is not unexpected. There may be ice scattering in the cell offshore as well.

Later during this flight, the ER-2 flew spatially distributed orbits, leading to greater areal coverage. Let's finish up this report by checking out that time period, this time taking advantage of the return of 19 GHz to compare it to 85 GHz.

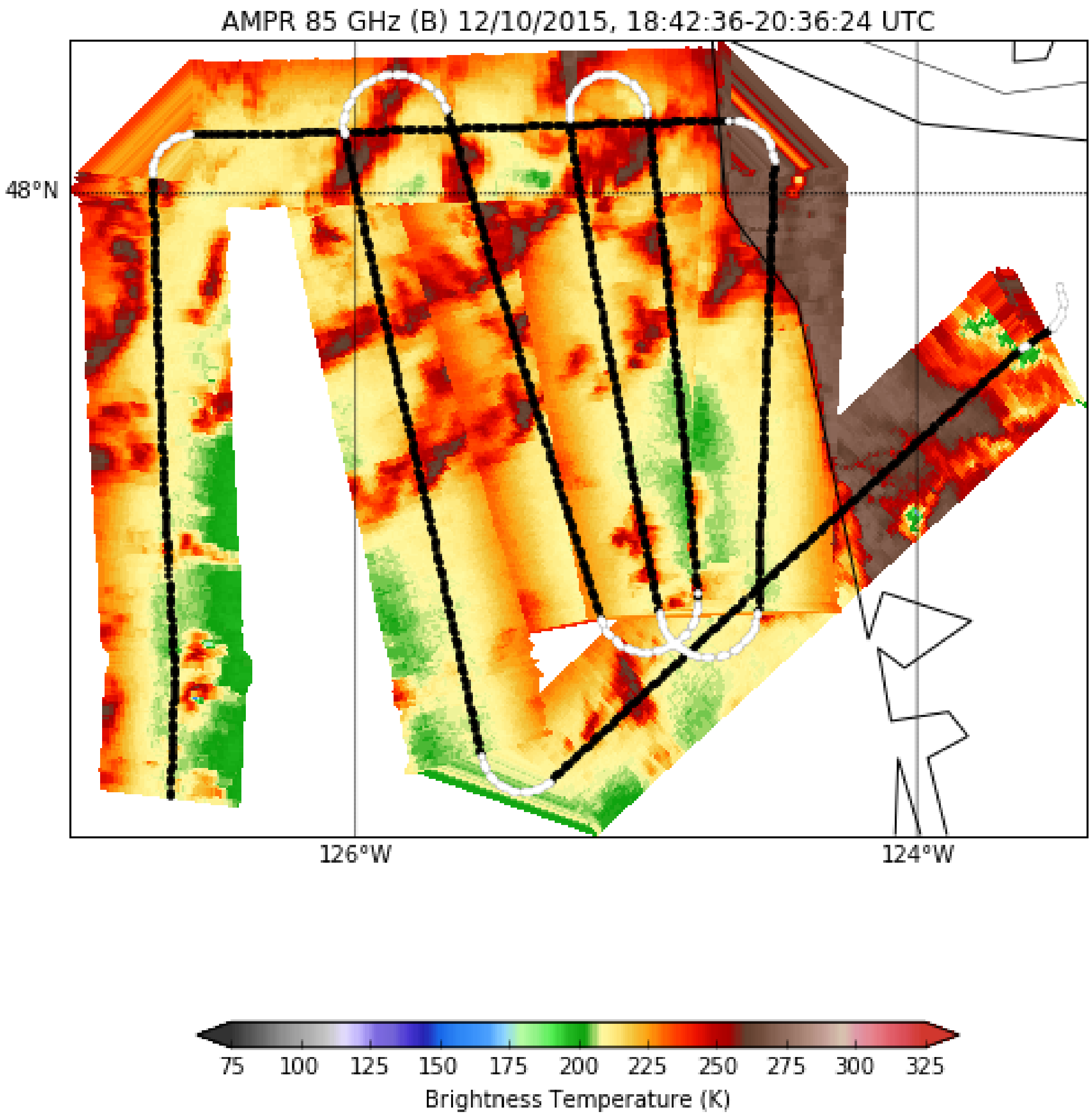
```
In [40]: data.plot_ampr_track('19b', scanrange=(2500, 4000), maneuver=False, show_track=True)
```

```
********************
plot_ampr_track():
Available scans = 1 to 5204
Available times = 13:28:33 - 22:08:14
Filtering out significant aircraft maneuvers
********************
```




```
In [39]: data.plot_ampr_track('85b', scanrange=(2500, 4000), maneuver=False, show_track=True)
```

```
********************
plot_ampr_track():
Available scans = 1 to 5204
Available times = 13:28:33 - 22:08:14
Filtering out significant aircraft maneuvers
********************


********************
plot_ampr_track():
Available scans = 1 to 5204
Available times = 13:28:33 - 22:08:14
Filtering out significant aircraft maneuvers
********************
```



This looks like multiple linear features, but what's really happening is AMPR is mapping the movement of smaller cells onshore. We can show this by examining NEXRAD about halfway through this two-hour period.

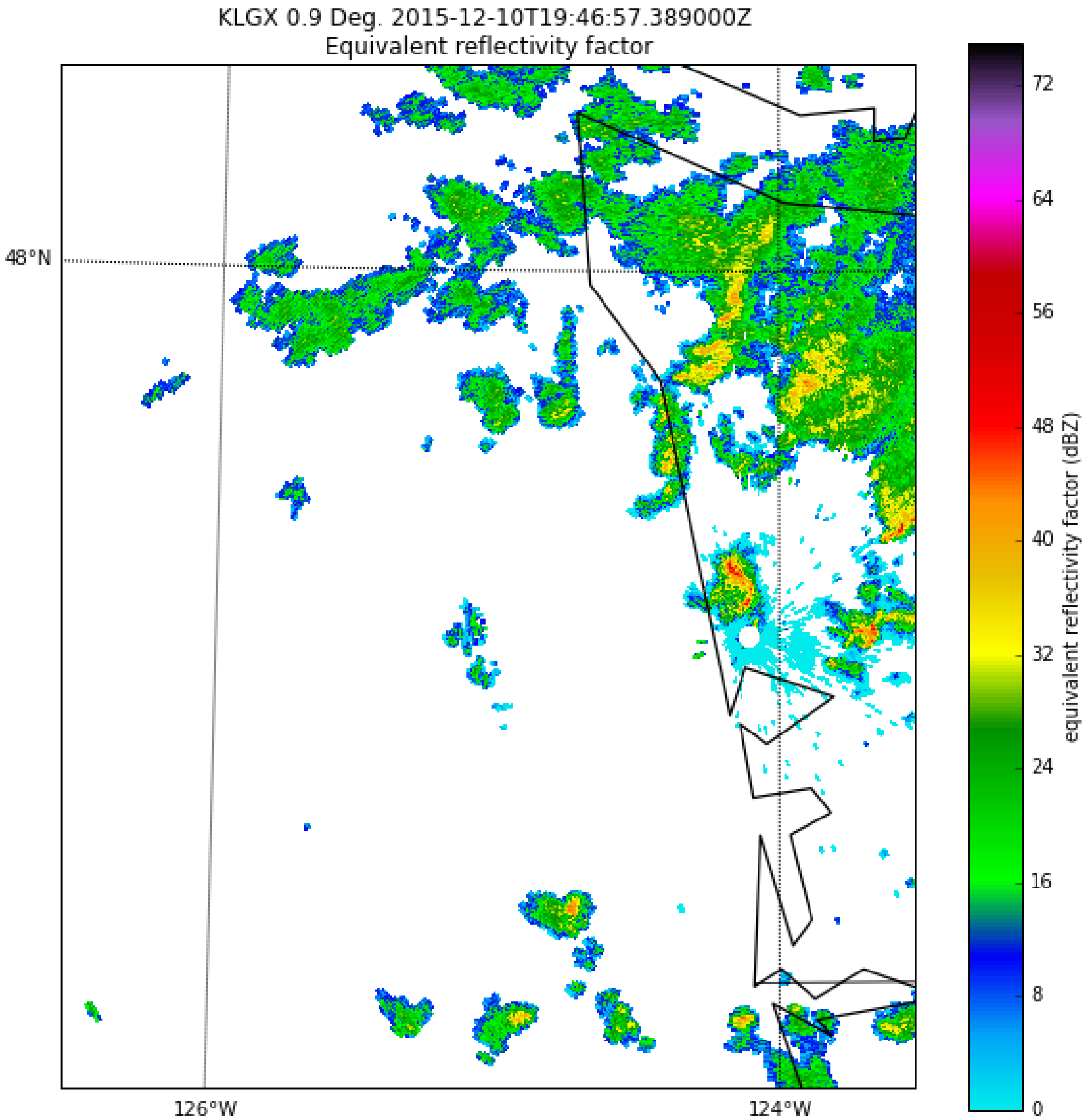
```
In [42]: radar = read_nexrad_aws('KLGX20151210_194558_V06.gz')
```

```
https://noaa-nexrad-level2.s3.amazonaws.com/2015/12/10/KLGX/KLGX20151210_194558_V06.gz
```

```
In [43]: # Do some quick QC on the NEXRAD data
         zc = deepcopy(radar.fields['REF']['data'])
         radar.add_field_like('REF', 'ZC', zc, replace_existing=True)
         retrieve = dualpol.DualPolRetrieval(
             radar, dz='ZC', dr='ZDR', dp='PHI', rh='RHO', use_temp=False,
             dsd_flag=False, fhc_flag=False, precip_flag=False,
             liquid_ice_flag=False, kdp_window=5, verbose=False,
             ice_flag=False, qc_flag=True, gs=250.0,
             thresh_sdp=20.0, speckle=3,
             thresh_dr=np.array(dualpol.DEFAULT_DR_THRESH)+1.0)
```

```
0.967968940735 seconds to run csu_kdp
No sounding provided
```

```
In [45]: display = pyart.graph.RadarMapDisplay(radar)
         fig = plt.figure(figsize=(10, 10))
         display.plot_ppi_map('ZC', 1, vmin=0, vmax=75, cmap='pyart_NWSRef', min_lon=-126.5,
                              max_lon=-123.5, min_lat=46, max_lat=48.5,
                              resolution='l', lat_lines=[48], lon_lines=[-126, -124])
```



As we can see, there is less organization to these cells than the mapping above would imply. However, clearly we captured a nice time history of this convection moving onshore with the ER-2.

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```