SWARM NIMO

In [1]:

```
import pandas as pd
from datetime import datetime, timedelta
# Self Created Functions ---------------------------
# Swarm downlaod and load functions
from download_swarm import download_and_unzip

# Plotting NIMO and Swarm together
from NIMO_Swarm_Map_Plotting import find_all_gaps, NIMO_SWARM_mapplot
from NIMO_SWARM_single import nimo_swarm_single_plot
```

Example of Downloading Swarm File

Function : download_swarm.download_and_unzip
Required Parameters

> ymd : datetime
>
> satellite : string 'A', B', or 'C'
>
> > Any other satellite letter will return file does not exist
>
> out_dir : string directory for output

Key Word Arguments

> bse_url : URL where data can be found
>
> > Defualt 'https://swarm-diss.eo.esa.int/?do=download&file=swarm%2FLevel'
> > The base URL to use can be found by going to
> > https://swarm-diss.eo.esa.int/#
> > and navigating to desired file, right clicking and
> > choosing "Copy Link Address"
> > Use string before the level is specified
>
> level : product level
>
> > Default '1b' can also use '2daily'
>
> baseline : product baseline
>
> > 'Latest_baselines' is recommended, has not been tested for 'Entire Mission Data'

> instrument: Instrument type
>
> > Default 'EFI' (Electric Field Instruments
>
> instrument2: Speficic instrument
>
> > Default 'LP' (Langmuir Probe)
>
> f_end : file ending
>
> > Default '0602_MDR_EFI_LP'
> > 0602 represents the file version
> > MDR_EFI_LP represents the Record Type
>
> T1 string starting time
>
> > Default '000000' (midnight)
> > MOST swarm files will follow this format but NOT ALL
>
> T2 string ending time
>
> > Default '235959' (1 minute before midnight)
> > MOST swarm files will follow this format but NOT ALL
>
> num_days : number of days that will be downloaded after initial file
>
> > Default is 0

File will be downloaded if it does not already exist in out_dir
File will not be downlaoded if that filename dows not exist on Swarm website
NOTE: often if a file does not exist, it is because T1 is not '000000' and
T2 is not '235959'
If that is the case, check the Swarm Data Website to find the proper times

In [3]:
```python
out_dir = '~/swarm_data/'
fdate = datetime(2020, 12, 29)
download_and_unzip(fdate, 'A', out_dir)
```

```
Downloading: SW_OPER_EFIA_LP_1B_20201229T000000_20201229T235959_0602.CD
F.ZIP
Extracted to: /Users/aotoole/Documents/Python_Code/EFI/Sat_A/2020/20201
229
```

Single NIMO/Swarm Plot Example

Function NIMO_SWARM_single.nimo_swarm_single_plot

This makes a single plot for Swarm and NIMO as close to the provided time as possible
If Swarm file is not found then it will attempt to download

Required Parameters

> stime : datetime to plot
> satelltie : string 'A', B', or 'C' for Swarm
> swarm_file_dir : file directory for Swarm data
> nimo_file_dir : file directory for NIMO data

Key Word Arguments

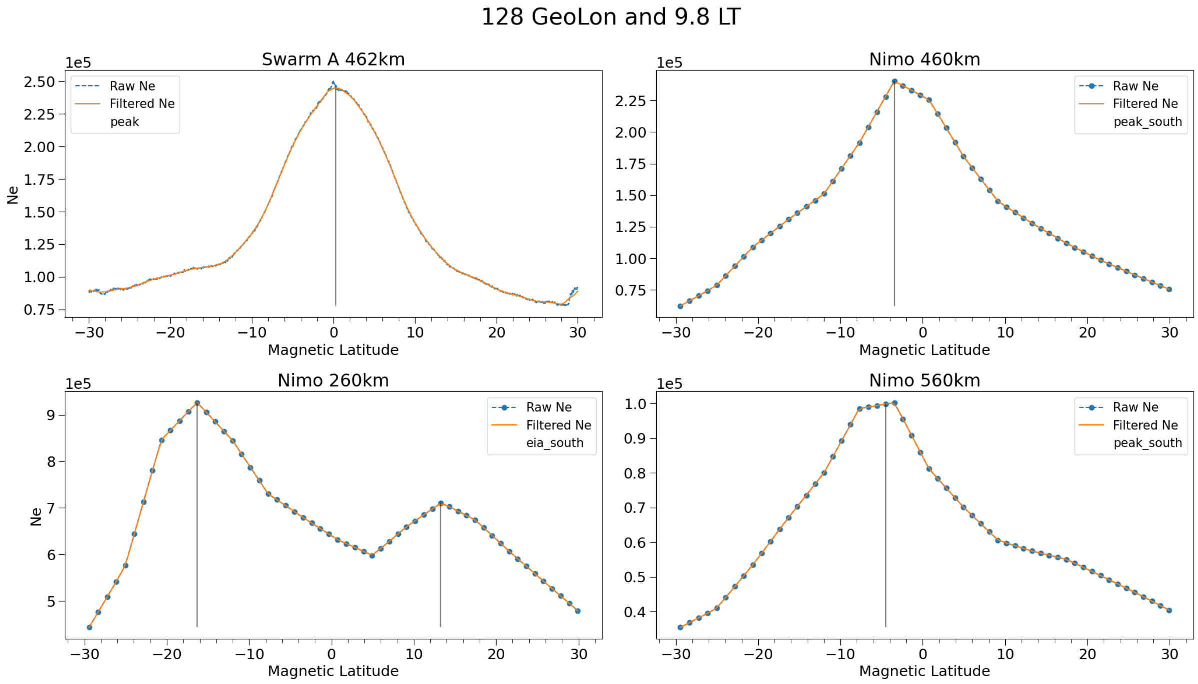
> MLat : Magnetic Latitude cutoff
>
>> 30° Default
>
> swarm_filt : filter for swarm data
>
>> Default is 'barrel_average'
>
> swarm_interpolate : linear interpolation parameter
>
>> the number of data points will incrase by swarm_interpolate
>> Default is 1 (no interpolation)
>
> swarm_envelope : boolean
>
>> determines if an envelope is used if barrel is in filter
>> Default is True
>
> swarm_barrel : double deteriming magnetic latitude radius of barrel
>
>> Default is 3°
>
> swarm_window : double determing magnetic latitude moving average window size
>
>> Default is 2°
>
> nimo_filt : filter for nimo data
>
>> Default '' (no filter)
>
> nimo_interpolate : linear interpolation parameter

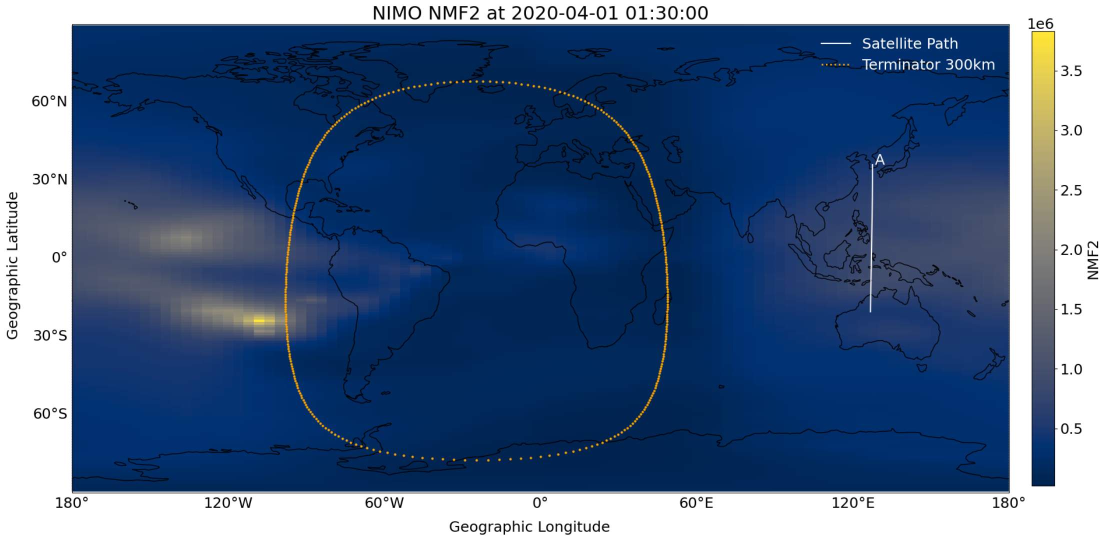
> > the number of data points will incrase by
> > swarm_interpolate
> > Default is 2 (doubles number of points)
>
> nimo_envelope : boolean
>
> > determines if an envelope is used if barrel is in filter
> > Default is False (no envelope)
>
> nimo_barrel : double deteriming magnetic latitude radius of barrel
>
> > Default is 3°
>
> nimo_window : double determing magnetic latitude moving average window size
>
> > Default is 3°
>
> fosi : int for plot font size
>
> > Default 18
> > Exceptions:
> >
> > > Super Title (fosi + 10)
> > > legends (fosi - 3)
>
> out_dir : string of output directory
>
> > if it is left empty ('' default), then cwd will be used
>
> nimo_name_format : string specifying nimo filename before '.nc'
>
> > Default is 'NIMO_AQ_%Y%j'
>
> *_var : str of variable names for NIMO
>
> > variable names to be opened in the NIMO file
> > * ne, lon, lat, alt, hr, min, tec, hmf2, nmf2
> > Defaults
> >
> > > electron density - 'dene'
> > > geo longitude - 'lon'
> > > geo latitude - 'lat'
> > > altitude - 'alt'
> > > hour - 'hour'
> > > minute - 'minute'

> > > TEC - 'tec'
> > >
> > > hmf2 - 'hmf2'
> > >
> > > nmf2 - 'nmf2'
>
> nimo_cadence: int
>
> > time cadence of NIMO data in minutes
> >
> > default is 15 minutes
>
> max_tdif : double
>
> > maximum time distance (in minutes) between a NIMO and Swarm
> >
> > conjunction allowed (default 15)

In [2]:

```
swarm_fdir = '~/swarm_data'
nimo_fdir = '~/NIMO/*'
stime = datetime(2020, 4, 1,1,17)
satellite = 'A'
nimo_swarm_single_plot(stime, satellite, swarm_fdir, nimo_fdir);
```





Creating Daily Figs and Daily Files

Function NIMO_Swarm_Map_Plotting.NIMO_SWARM_mapplot
This function creates daily files and figures for SWARM (all_satellites) and NIMO conjunctions
Returns adataframe of the information that goes into the daily file

Required Parameters

> start_day : datetime (day to create file for)
> swarm_file_dir : file directory for Swarm data
> nimo_file_dir : file directory for NIMO data

Key Word Arguments

> MLat : Magnetic Latitude cutoff
>
> > 30° Default
>
> file_dir: string of output directory for file
>
> > if it is left empty ('' default), then cwd will be used
>
> fig_dir: string of output directory for figures
>
> > if it is left empty ('' default), then cwd will be used
>
> fig_on : boolean specifying whether or not to make the file Default True
> swarm_filt : filter for swarm data
>
> > Default is 'barrel_average'
>
> swarm_interpolate : linear interpolation parameter
>
> > the number of data points will incrase by swarm_interpolate
> > Default is 1 (no interpolation)
>
> swarm_envelope : boolean
>
> > determines if an envelope is used if barrel is in filter
> > Default is True
>
> swarm_barrel : double deteriming magnetic latitude radius of barrel
>
> > Default is 3°
>
> swarm_window : double determing magnetic latitude moving average window size
>
> > Default is 2°
>
> nimo_filt : filter for nimo data
>
> > Default '' (no filter)
>
> nimo_interpolate : linear interpolation parameter
>
> > the number of data points will incrase by swarm_interpolate

> > Default is 2 (doubles number of points)
>
> nimo_envelope : boolean
>
> > determines if an envelope is used if barrel is in filter
> > Default is False (no envelope)
>
> nimo_barrel : double deteriming magnetic latitude radius of barrel
>
> > Default is 3°
>
> nimo_window : double determing magnetic latitude moving average window size
>
> > Default is 3°
>
> fosi : int for plot font size
>
> > Default 18
> > Exceptions:
> >
> > > Super Title (fosi + 10)
> > > legends (fosi - 3)
>
> nimo_name_format : string specifying nimo filename before '.nc'
>
> > Default is 'NIMO_AQ_%Y%j'
>
> *_var : str of variable names for NIMO
>
> > variable names to be opened in the NIMO file
> > * ne, lon, lat, alt, hr, min, tec, hmf2, nmf2
> > Defaults
> >
> > > electron density - 'dene'
> > > geo longitude - 'lon'
> > > geo latitude - 'lat'
> > > altitude - 'alt'
> > > hour - 'hour'
> > > minute - 'minute'
> > > TEC - 'tec'
> > > hmf2 - 'hmf2'
> > > nmf2 - 'nmf2'
>
> nimo_cadence: int

> > time cadence of NIMO data in minutes
> > default is 15 minutes
>
> max_tdif : double
>
> > maximum time distance (in minutes) between a NIMO
> > and Swarm
> > conjunction allowed (default 15)

In [3]:
```
fig_dir='~/Plots/NIMO_SWARM'
file_dir='~/Type_Files/Daily'
swarm_fdir = '~/swarm_data'
nimo_fdir='~/Python_Code/data/NIMO/'
mlat_val = 30
stime1 = datetime(2020, 4,1,0,0)  # Starting Date
for i in range(30):  # How many days you want to make files for
    stime = stime1 + timedelta(days=i)
    print(stime)
    df_big = NIMO_SWARM_mapplot(stime,swarm_fdir, nimo_fdir, MLat=mlat
                                file_dir=file_dir, fig_on=True,
                                fig_dir=fig_dir)
```

```
2020-04-01 00:00:00
2020-04-02 00:00:00
2020-04-03 00:00:00
2020-04-04 00:00:00
2020-04-05 00:00:00
Odd Orbit longitude span > 5 degrees: Skipping Pass
2020-04-06 00:00:00
Odd Orbit longitude span > 5 degrees: Skipping Pass
2020-04-07 00:00:00
2020-04-08 00:00:00
2020-04-09 00:00:00
Odd Orbit longitude span > 5 degrees: Skipping Pass
2020-04-10 00:00:00
2020-04-11 00:00:00
2020-04-12 00:00:00
2020-04-13 00:00:00
2020-04-14 00:00:00
Odd Orbit longitude span > 5 degrees: Skipping Pass
2020-04-15 00:00:00
2020-04-16 00:00:00
2020-04-17 00:00:00
Odd Orbit longitude span > 5 degrees: Skipping Pass
2020-04-18 00:00:00
2020-04-19 00:00:00
2020-04-20 00:00:00
2020-04-21 00:00:00
2020-04-22 00:00:00
Odd Orbit longitude span > 5 degrees: Skipping Pass
2020-04-23 00:00:00
2020-04-24 00:00:00
2020-04-25 00:00:00
Odd Orbit longitude span > 5 degrees: Skipping Pass
Odd Orbit longitude span > 5 degrees: Skipping Pass
2020-04-26 00:00:00
2020-04-27 00:00:00
2020-04-28 00:00:00
2020-04-29 00:00:00
2020-04-30 00:00:00
```

Swarm vs PyIRI at NIMO Conjunctions

In [5]:

```python
from datetime import datetime, timedelta
# Self Created Function
# PyIRI daily files and plotting function
from SwarmPyIRI import PyIRI_NIMO_SWARM_plot
```

PyIRI daily Files at Nimo Conjunctions

Function SwarmPyIRI.PyIRI_NIMO_SWARM_plot
This function creates daily plots and a daily file

based on the NIMO-Swarm conjunctions found in the
NIMO Swarm daily files

Required parameters

> sday: datetime
>
> > (day starting at 0,0)
>
> daily_dir : str
>
> > directory of daily files made by
> > NIMO_Swarm_Map_Plotting.NIMO_SWARM_mapplot
>
> swarm_dir : str
>
> > Swarm data directory to which data will be
> > downloaded into an
> > appropriate date/satellite directory structure

Key Word Arguments

> file_save_dir : str kwarg
>
> > directory where file should be saved, default cwd
>
> fig_on : kwarg bool
>
> > set to true, plot will be made, if false, plot will not be
> > made
>
> fig_save_dir : str kwarg
>
> > directory where figure should be saved, default cwd
>
> pyiri_filt : str kwarg
>
> > Desired Filter for nimo data (no filter default)
>
> pyiri_interpolate : int kwarg
>
> > int that determines the number of data points in
> > interpolation
> > new length will be len(density)xinterpolate
> > default is 2

> pyiri_envelope : bool kwarg
>
> > if True, barrel roll will include points inside an envelope, if false (default), no envelope will be used
>
> pyiri_barrel : double
>
> > latitudinal radius of barrel for swarm (default: 3 degrees maglat)
>
> pyiri_window : double kwarg
>
> > latitudinal width of moving window (default: 3 degrees maglat)
>
> fosi : int kwarg
>
> > fontsize for plot (default is 18)
> > Exceptions:
> >
> > > Super Title (fosi + 10)
> > > legends (fosi - 3)

The returns include daily files, figures (if fig_on),
and a dataframe with what is contained in the daily files

In [8]:

```
daily_files = '~/Type_Files/Daily'
pyiri_fig_dir = '~/Plots/NIMO_SWARM/'
pyiri_file_dir = '~/Type_Files/Daily/'

stime1 = datetime(2020, 4,1,0,0) # Starting Date
for i in range(30):  # How many days you want to make files for
    stime = stime1 + timedelta(days=i)
    print(stime)
    iridf,dailydf = PyIRI_NIMO_SWARM_plot(stime, daily_files,swarm_fdi
                                          file_save_dir=pyiri_file_dir)
```

```
2020-04-01 00:00:00
2020-04-02 00:00:00
2020-04-03 00:00:00
2020-04-04 00:00:00
2020-04-05 00:00:00
2020-04-06 00:00:00
2020-04-07 00:00:00
2020-04-08 00:00:00
2020-04-09 00:00:00
2020-04-10 00:00:00
2020-04-11 00:00:00
2020-04-12 00:00:00
2020-04-13 00:00:00
2020-04-14 00:00:00
2020-04-15 00:00:00
2020-04-16 00:00:00
2020-04-17 00:00:00
2020-04-18 00:00:00
2020-04-19 00:00:00
2020-04-20 00:00:00
2020-04-21 00:00:00
2020-04-22 00:00:00
2020-04-23 00:00:00
2020-04-24 00:00:00
2020-04-25 00:00:00
2020-04-26 00:00:00
2020-04-27 00:00:00
2020-04-28 00:00:00
2020-04-29 00:00:00
2020-04-30 00:00:00
```

What still needs to be done?

Create separate PyIRI plots with maps

```
In [ ]:
```