Paleomagnetic Data Analysis of Late Stage Volcanics from the Midcontinent Rift

This Jupyter notebook is provided as the main Data Repository for a manuscript in review entitled **The end of Midcontinent Rift magmatism and the paleogeography of Laurentia** by Luke M. Fairchild, Nicholas L. Swanson-Hysell, Jahandar Ramenzani, Courtney J. Sprain, and Samuel A. Bowring. This notebook can be downloaded for interactive viewing[1] or viewed statically within a web browser[2]; it is highly recommended that this notebook be viewed in either of these two ways, although a PDF rendering of the notebook is also provided in the Data Repository.

[1] https://github.com/Swanson-Hysell-Group/2016_Late_Rift/blob/master/Code/Late_Rift_Data_Analysis.ipynb

[2] https://nbviewer.jupyter.org/github/swanson-hysell-group/2016_Late_Rift/blob/master/Code/Late_Rift_Data_Analysis.ipynb

Contents

1 Import and develop functions for use within the Jupyter notebook

The code block below imports necessary libraries and defines functions that will be used in the data analysis below. The modules `pmag.py`, `pmagplotlib.py` and `ipmag.py` are from the PmagPy software package (https://github.com/ltauxe/PmagPy). The current version of these modules (PmagPy 3.4) are included in the repository with this notebook. Other necessary function libraries (`matplotlib`, `pandas`, `Basemap`, `IPython`, `numpy`, `scipy`) are included with standard scientific Python distributions.

```
In [1]: import pmagpy_3_4.pmag as pmag
        import pmagpy_3_4.pmagplotlib as pmagplotlib
        import pmagpy_3_4.ipmag as ipmag

        import matplotlib.pyplot as plt
        from mpl_toolkits.basemap import Basemap
        from matplotlib.patches import Polygon
        import matplotlib.patches as mpatch
        from matplotlib.collections import PatchCollection
        import pandas as pd
        from IPython.core.display import HTML
        import numpy as np
        import scipy as sp
        from scipy import special
        from IPython.display import Image
```

This notebook runs with figures inline in the Jupyter notebook (instead of opening up in another window) by executing this command:

```
In [2]: %matplotlib inline
        %config InlineBackend.figure_formats = {'svg',}
```

Throughout this analysis there will be maps that show pole positions. The `pole_figure_appearance` function controls aspects of the appearance of these maps and eliminates the need to make these customizations for every plot.

```
In [3]: def pole_figure_appearance(size = (7,7)):
            m = Basemap(projection='ortho',lat_0=35,lon_0=200,resolution='c',
                        area_thresh=50000)
            plt.figure(figsize=size)
            m.drawcoastlines(linewidth=0.25)
            m.fillcontinents(color='bisque',lake_color='white',zorder=1)
            m.drawmapboundary(fill_color='white')
            m.drawmeridians(np.arange(0,360,30))
            m.drawparallels(np.arange(-90,90,30))
            return m
```

2 Schroeder-Lutsen Basalts

2.1 Prior data

Tauxe and Kodama (2009) published data from sites of the North Shore Volcanic Group and Schroeder-Lutsen Basalts that was the first study of these units to use modern methods. Books (1968, 1972) and Palmer (1970) also developed data that can largely be considered to be superseded by the Tauxe and Kodama results. In the code below, the Tauxe and Kodama (2009) data are imported and then the data that are from the Schroeder-Lutsen Basalts are split off from the rest of the data that were generated by Tauxe and Kodama (2009) from the North Shore Volcanic Group.

```
In [4]: Tauxe_NSVG_Data=pd.read_csv('../Data/Previous_studies/Tauxe2009a_data.csv',sep=',')
        Tauxe_NSVG_Data.head()
```

```
Out[4]:   Click to view data of Tauxe and Kodama (2009)
```

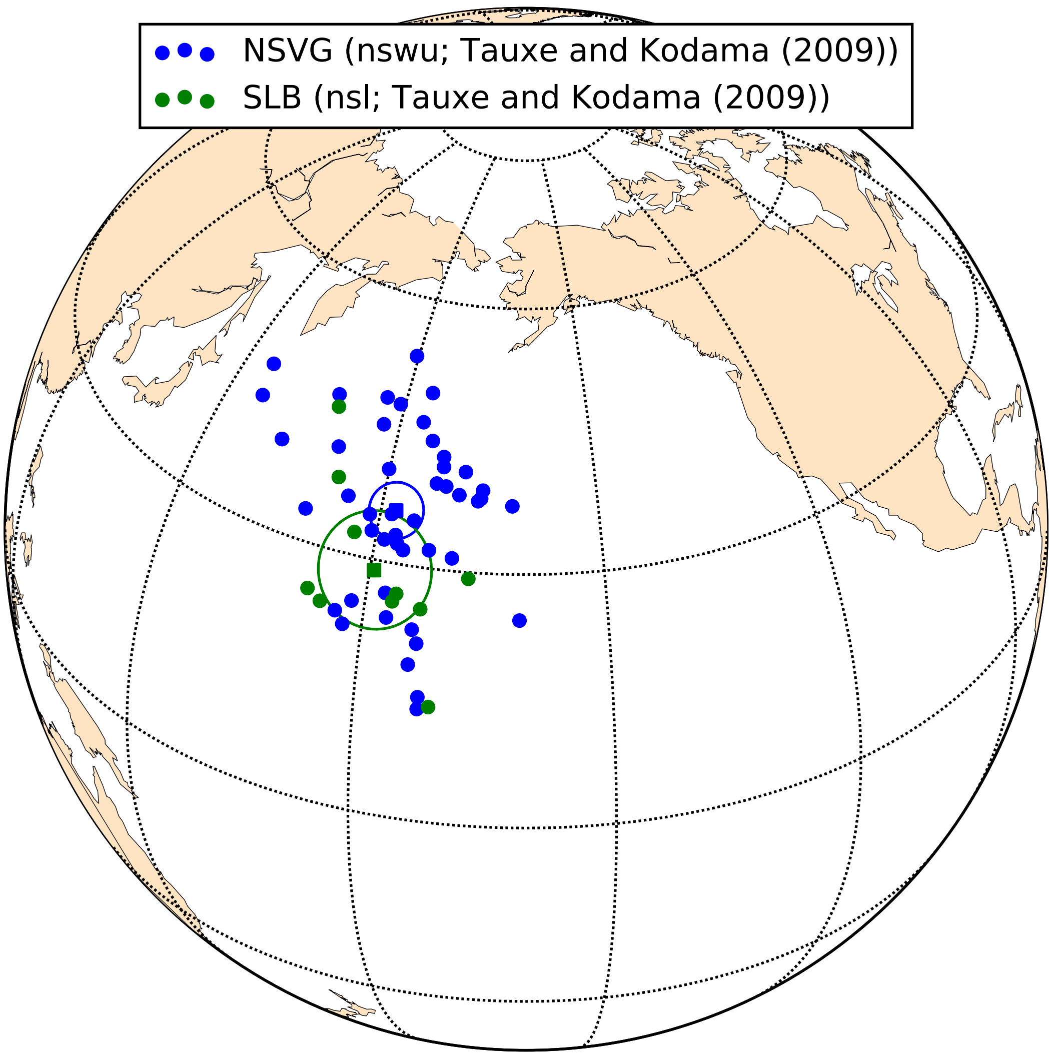
NOTE: The hyperlink above may only be active if you are reading this in Adobe Acrobat. If you are having trouble opening the local file via this link, or if you have downloaded this document separate from the accompanying data files, the file can be found in the Data Repository under

/Data/Previous_studies/Tauxe2009a_data.csv and opened manually.

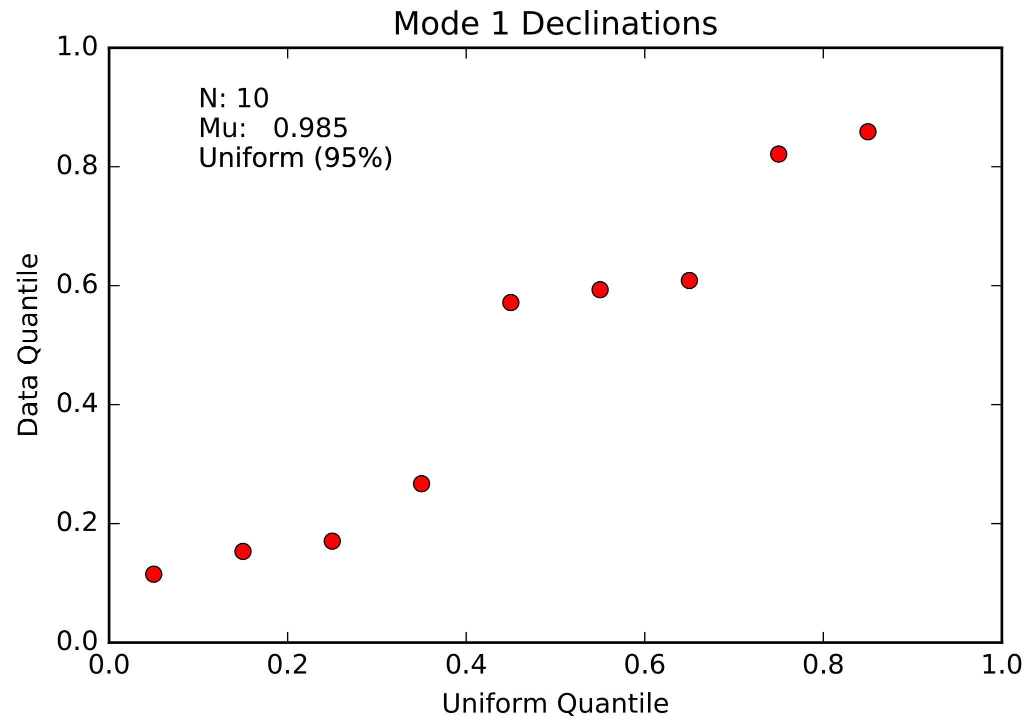
The **sequence** column contains an assignment of the sequence with 'nsl' signifying that we interpret the flow to be within the Schroeder-Lutsen basalts. A new dataframe can be made with only these flows.

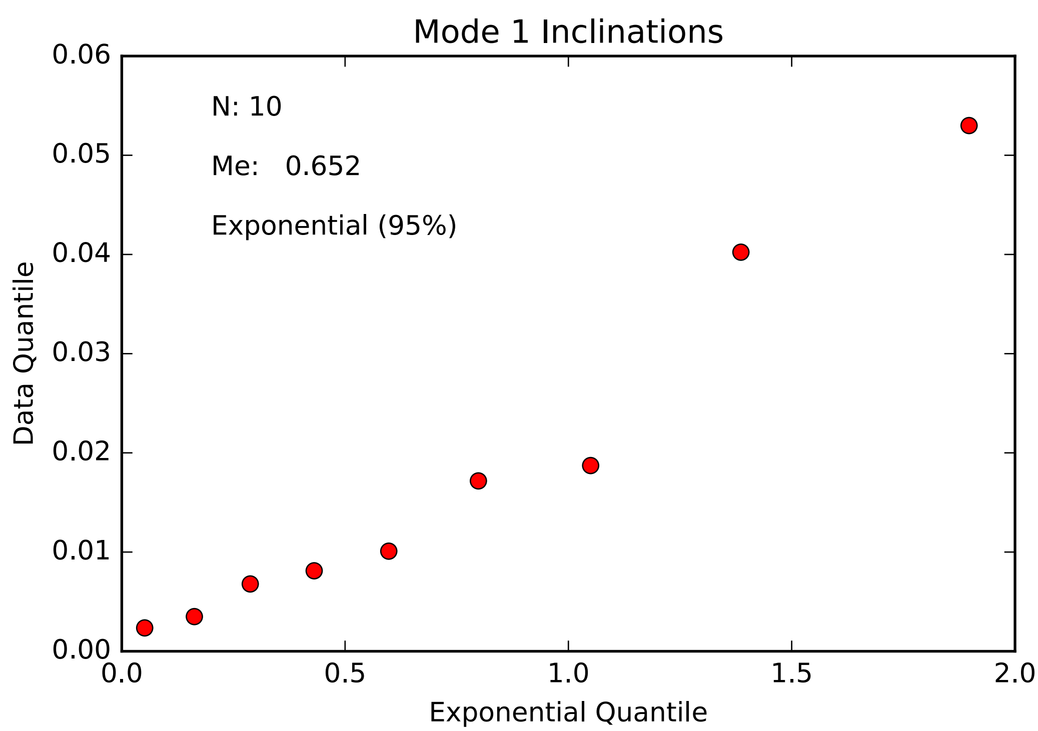
```
In [5]: NSVG_nswu = Tauxe_NSVG_Data.ix[Tauxe_NSVG_Data['sequence'] == 'nswu']
        NSVG_nswu.reset_index(inplace=True)

        NSVG_nswu_VGPs = ipmag.make_di_block(NSVG_nswu['pole_lon'],NSVG_nswu['pole_lat'])
        NSVG_nswu_mean = pmag.fisher_mean(NSVG_nswu_VGPs)
        ipmag.print_pole_mean(NSVG_nswu_mean)

        NSVG_nsl = Tauxe_NSVG_Data.ix[Tauxe_NSVG_Data['sequence'] == 'nsl']
        NSVG_nsl.reset_index(inplace=True)

        NSVG_nsl_VGPs = ipmag.make_di_block(NSVG_nsl['pole_lon'],NSVG_nsl['pole_lat'])
        NSVG_nsl_mean = pmag.fisher_mean(NSVG_nsl_VGPs)
        ipmag.print_pole_mean(NSVG_nsl_mean)

        m = pole_figure_appearance()

        ipmag.plot_vgp(m,NSVG_nswu['pole_lon'].tolist(),NSVG_nswu['pole_lat'].tolist(),
                       color='b',label='NSVG (nswu; Tauxe and Kodama (2009))')
        ipmag.plot_pole(m,NSVG_nswu_mean['dec'],NSVG_nswu_mean['inc'],
                        NSVG_nswu_mean['alpha95'], marker='s',color='b')
        ipmag.plot_vgp(m,NSVG_nsl['pole_lon'].tolist(),NSVG_nsl['pole_lat'].tolist(),
                       color='g',label='SLB (nsl; Tauxe and Kodama (2009))')
        ipmag.plot_pole(m,NSVG_nsl_mean['dec'],NSVG_nsl_mean['inc'],
                        NSVG_nsl_mean['alpha95'], marker='s',color='g')
        plt.legend(loc='upper center')
        plt.show()
```

```
Plong: 182.1  Plat: 35.8
Number of directions in mean (n): 47
Angular radius of 95% confidence (A_95): 3.1
Precision parameter (k) estimate: 45.7
Plong: 180.5  Plat: 28.8
Number of directions in mean (n): 10
Angular radius of 95% confidence (A_95): 6.5
Precision parameter (k) estimate: 55.6
```




```
In [6]: ipmag.fishqq(NSVG_nsl['pole_lon'].tolist(),NSVG_nsl['pole_lat'].tolist())

Out[6]: {'Dec': 180.5436552009686,
         'Inc': 28.803422228130206,
         'Me': 0.65209594384149339,
         'Me_critical': 1.094,
         'Mode': 'Mode 1',
         'Mu': 0.98495553684353243,
         'Mu_critical': 1.207,
         'N': 10,
         'Test_result': 'consistent with Fisherian model'}
```


Mode 1 Declinations
N: 10
Mu: 0.985
Uniform (95%)
Data Quantile
Uniform Quantile
0.0
0.2
0.4
0.6
0.8
1.0


Mode 1 Inclinations
N: 10
Me: 0.652
Exponential (95%)
Data Quantile
Exponential Quantile
0.00
0.01
0.02
0.03
0.04
0.05
0.06
0.0
0.5
1.0
1.5
2.0

2.2 New data from Two Island River

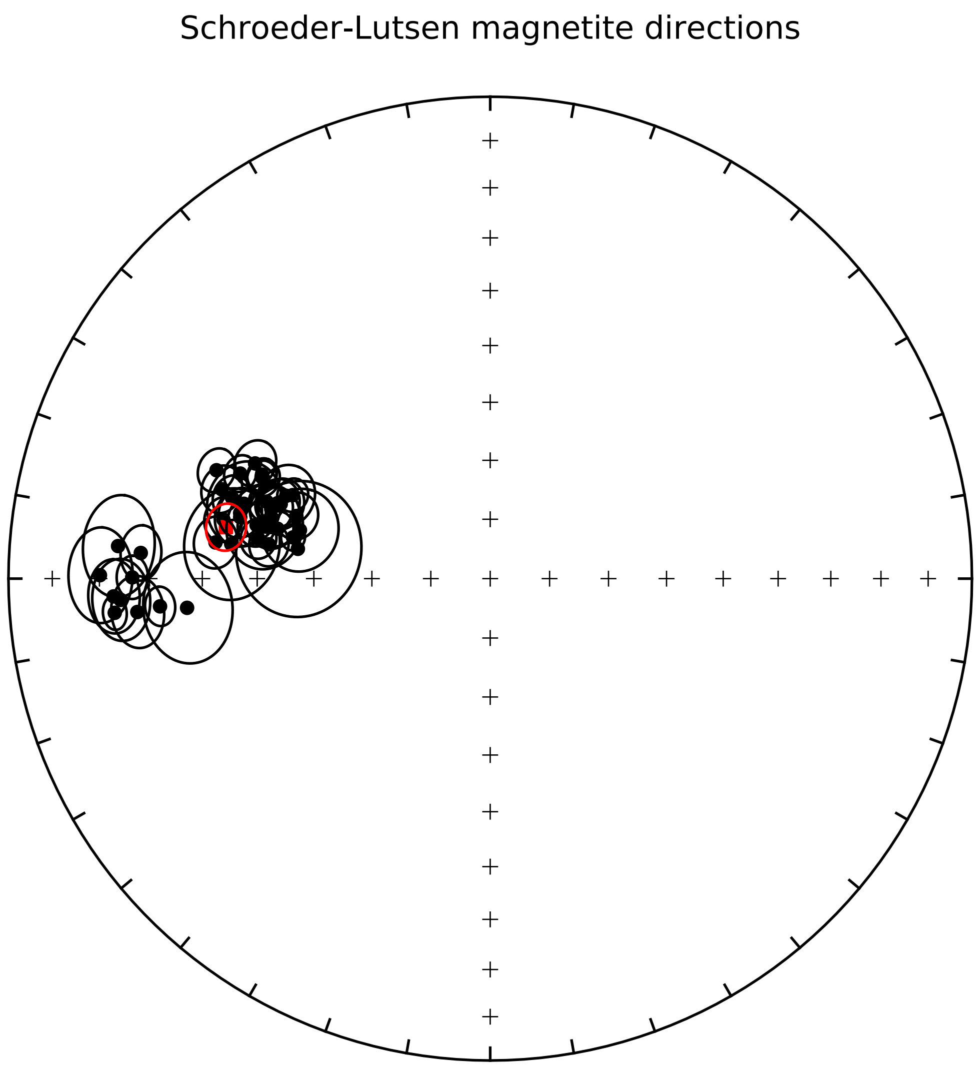
We present new paleomagnetic data from 40 Schroeder-Lutsen basalt flows exposed within Two Island River. Magnetizations held by magnetite and those held by hematite or maghemite are virtually identical (see equal area plots below), which suggests oxidation of these lava flows in association with eruption or shortly thereafter.

```
In [7]: SLB_Data_all = pd.read_csv('../Data/SLB/pmag_results.txt',sep='\t',skiprows=1)
        SLB_Data_all_tc = SLB_Data_all.ix[SLB_Data_all['tilt_correction'] == 100.0]
        SLB_Data_hem = SLB_Data_all.ix[SLB_Data_all['pole_comp_name'] == 'hem']
        SLB_Data_hem = SLB_Data_hem.ix[SLB_Data_hem['tilt_correction'] == 100.0]
        SLB_Data_hem.reset_index
        SLB_Data = SLB_Data_all.ix[SLB_Data_all['pole_comp_name'] == 'mag']
        SLB_Data = SLB_Data.ix[SLB_Data['tilt_correction'] == 100.0]
        SLB_Data.reset_index()
        SLB_Data.head()
```

```
Out[7]: Click to view MagIC file of new Schroeder-Lutsen paleomagnetic data
```

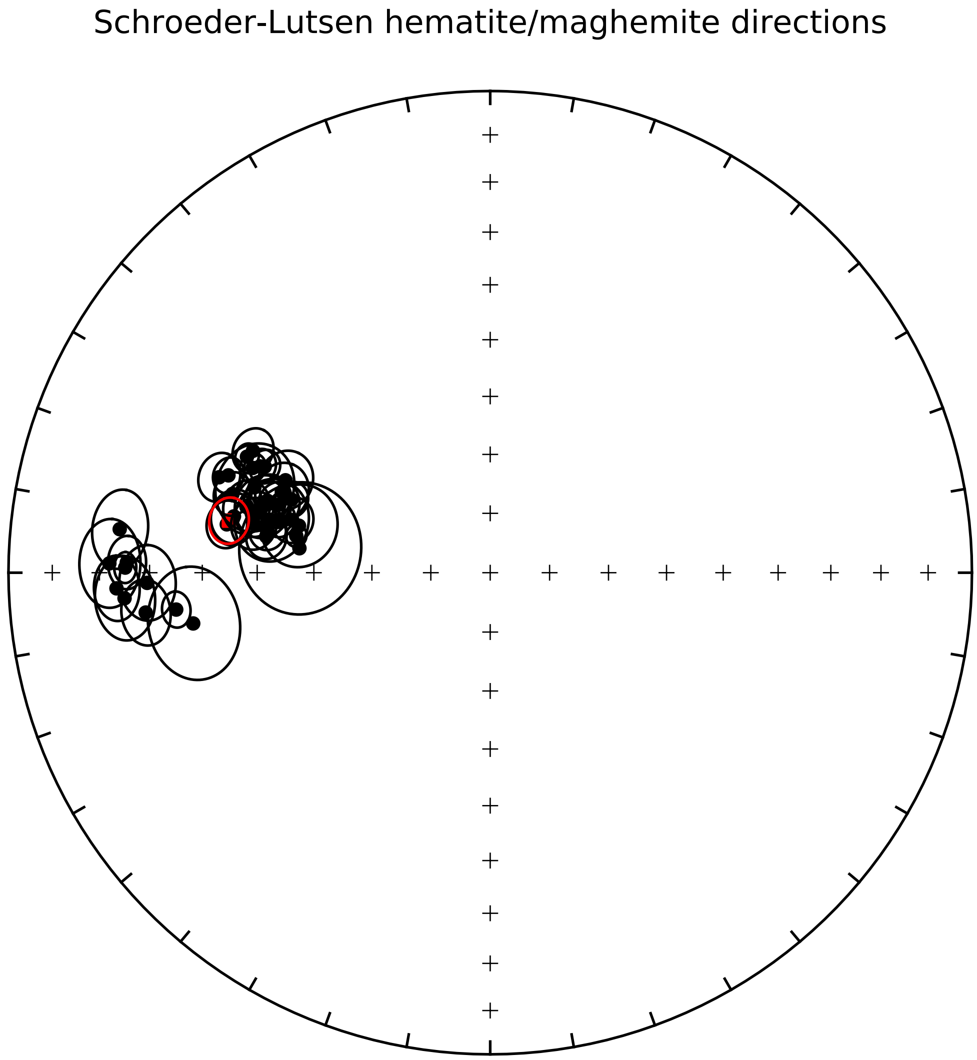
If you are having trouble opening the local file via this link, the file can be found in the Data Repository under `/Data/SLB/pmag_results` and opened manually.

```
In [8]: plt.figure(num=1,figsize=(6,6))
        ipmag.plot_net(1)

        SLB_dec = SLB_Data['average_dec'].tolist()
        SLB_inc = SLB_Data['average_inc'].tolist()
        SLB_a95 = SLB_Data['average_alpha95'].tolist()

        for i in range(len(SLB_dec)):
            ipmag.plot_di_mean(SLB_dec[i], SLB_inc[i], SLB_a95[i])

        SLB_mean_dir = ipmag.fisher_mean(SLB_dec,SLB_inc)

        ipmag.plot_di_mean(SLB_mean_dir['dec'],SLB_mean_dir['inc'],
                           SLB_mean_dir['alpha95'],marker='s', color='r')
        plt.title('Schroeder-Lutsen magnetite directions')

        plt.savefig('Code_output/All_SLB_mag_data.pdf')
        plt.show()
```




```
In [9]: plt.figure(num=1,figsize=(6,6))
        ipmag.plot_net(1)

        SLB_hem_dec = SLB_Data_hem['average_dec'].tolist()
        SLB_hem_inc = SLB_Data_hem['average_inc'].tolist()
        SLB_hem_a95 = SLB_Data_hem['average_alpha95'].tolist()

        for i in range(len(SLB_hem_dec)):
            ipmag.plot_di_mean(SLB_hem_dec[i], SLB_hem_inc[i], SLB_hem_a95[i])

        SLB_hem_mean_dir = ipmag.fisher_mean(SLB_hem_dec,SLB_hem_inc)

        ipmag.plot_di_mean(SLB_hem_mean_dir['dec'],SLB_hem_mean_dir['inc'],
                           SLB_hem_mean_dir['alpha95'],marker='s', color='r')
```

```
plt.title('Schroeder-Lutsen hematite/maghemite directions')

plt.savefig('Code_output/All_SLB_hem_data.pdf')
plt.show()
```



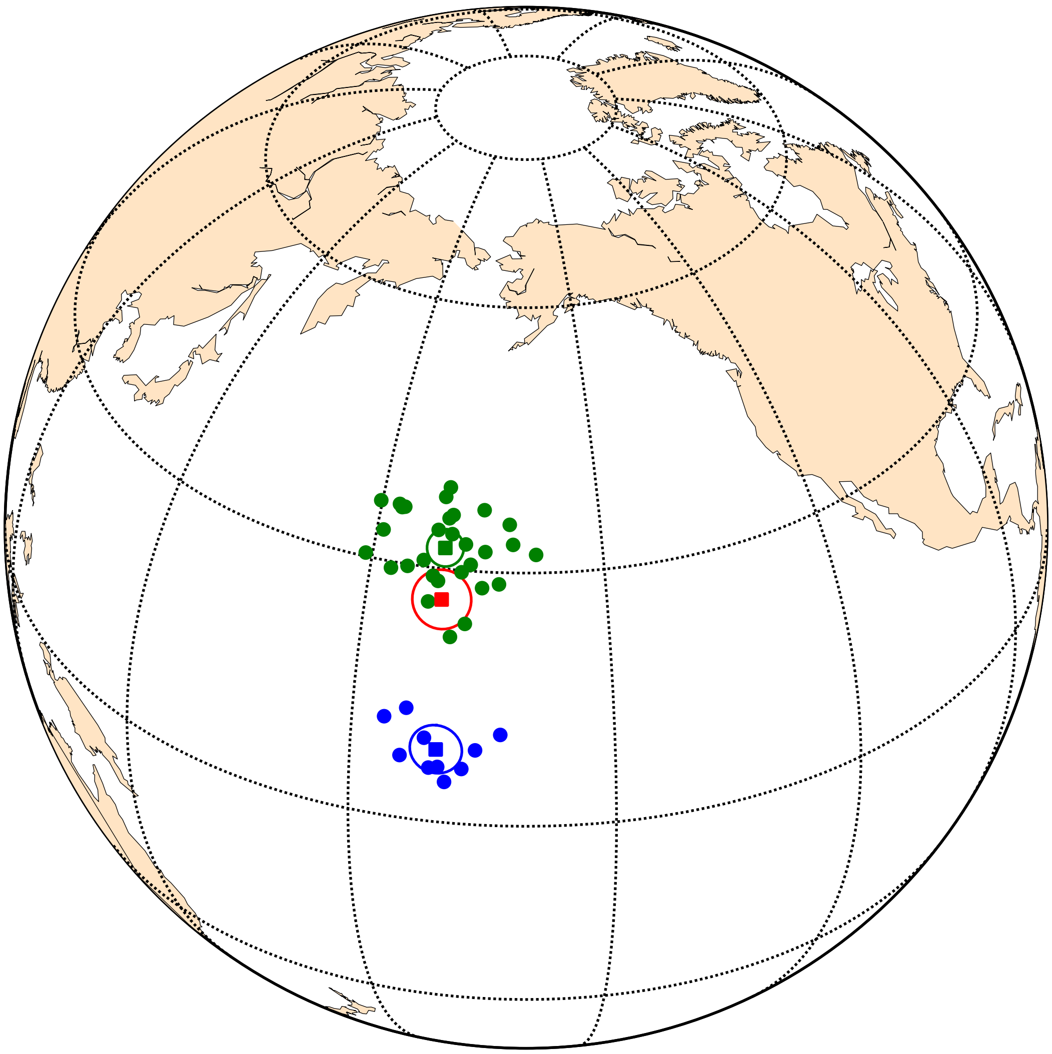
Below, we calculate the average angular difference between the magnetite and hematite paleomagnetic directions of each sampled flow.

```
In [10]: SLB_samples = pd.read_csv('../Data/SLB/pmag_specimens.txt',sep='\t',skiprows=1)
         SLB_samples = SLB_samples.ix[SLB_samples['specimen_tilt_correction'] == 100.0]

In [11]: SLB_mag_hem_diff = []
         for sample in SLB_samples.er_specimen_name.unique().tolist():
             try:
                 mag_dec = float(SLB_samples.loc[SLB_samples['er_specimen_name']==sample]\
```

```
                        .loc[SLB_samples['specimen_comp_name']=='mag'].specimen_dec)
            mag_inc = float(SLB_samples.loc[SLB_samples['er_specimen_name']==sample]\
                        .loc[SLB_samples['specimen_comp_name']=='mag'].specimen_inc)
            hem_dec = float(SLB_samples.loc[SLB_samples['er_specimen_name']==sample]\
                        .loc[SLB_samples['specimen_comp_name']=='hem'].specimen_dec)
            hem_inc = float(SLB_samples.loc[SLB_samples['er_specimen_name']==sample]\
                        .loc[SLB_samples['specimen_comp_name']=='hem'].specimen_inc)
            SLB_mag_hem_diff.append(pmag.angle([mag_dec, mag_inc],[hem_dec, hem_inc]))
        except:
            continue
```

```
Average angle between magnetite and hematite/maghemite fits for all samples:  2.86292160508
```

In [12]:
```
SLB_Data_low_lat = SLB_Data.ix[SLB_Data['vgp_lat'] < 15]
SLB_Data_hi_lat = SLB_Data.ix[SLB_Data['vgp_lat'] >= 15]

new_SLB_mean = ipmag.fisher_mean(SLB_Data['vgp_lon'].tolist(),
                SLB_Data['vgp_lat'].tolist())
new_SLB_low_lat_mean = ipmag.fisher_mean(SLB_Data_low_lat['vgp_lon'].tolist(),
                SLB_Data_low_lat['vgp_lat'].tolist())
new_SLB_hi_lat_mean = ipmag.fisher_mean(SLB_Data_hi_lat['vgp_lon'].tolist(),
                SLB_Data_hi_lat['vgp_lat'].tolist())

m = pole_figure_appearance()

# ipmag.plot_vgp(m,SLB_Data['vgp_lon'].tolist(),SLB_Data['vgp_lat'].tolist(),
#                label='SLB VGPs',color='r')
ipmag.plot_vgp(m,SLB_Data_low_lat['vgp_lon'].tolist(),
                SLB_Data_low_lat['vgp_lat'].tolist(),
                label='SLB low lat VGPs',color='b')
ipmag.plot_pole(m,new_SLB_low_lat_mean['dec'],new_SLB_low_lat_mean['inc'],
                new_SLB_low_lat_mean['alpha95'],
                label='SLB low lat mean',marker='s',color='b')
ipmag.plot_vgp(m,SLB_Data_hi_lat['vgp_lon'].tolist(),
                SLB_Data_hi_lat['vgp_lat'].tolist(),
               label='SLB high lat VGPs',color='g')
ipmag.plot_pole(m,new_SLB_hi_lat_mean['dec'],new_SLB_hi_lat_mean['inc'],
                new_SLB_hi_lat_mean['alpha95'],
                label='SLB high lat mean',marker='s',color='g')
ipmag.plot_pole(m,new_SLB_mean['dec'],new_SLB_mean['inc'],new_SLB_mean['alpha95'],
                label='All SLB mean',marker='s',color='red')
```

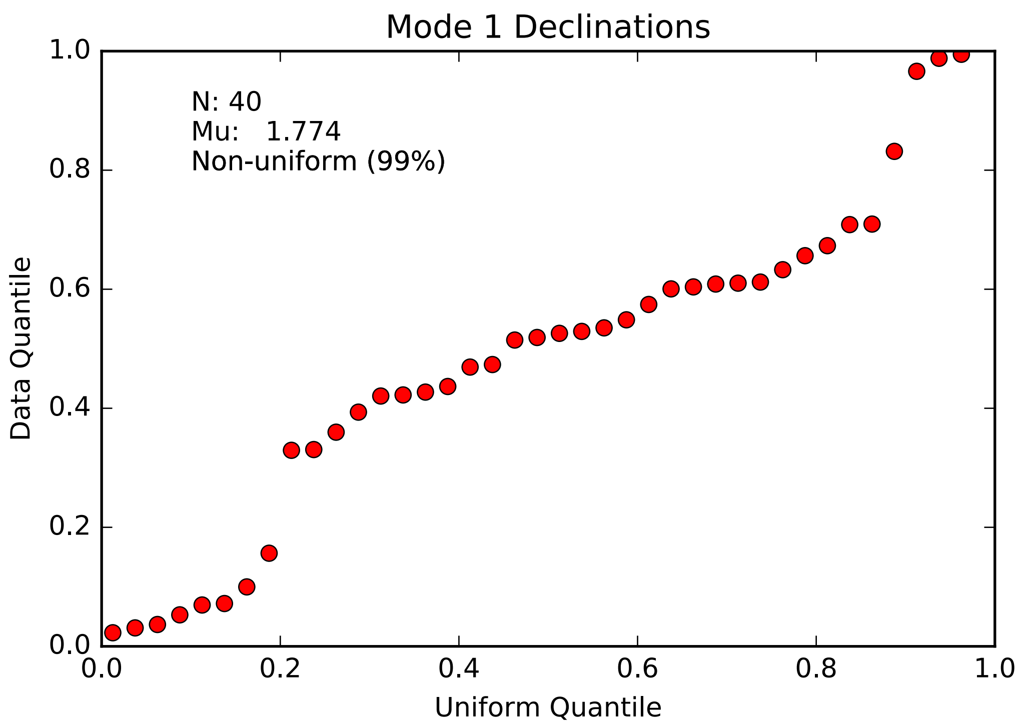


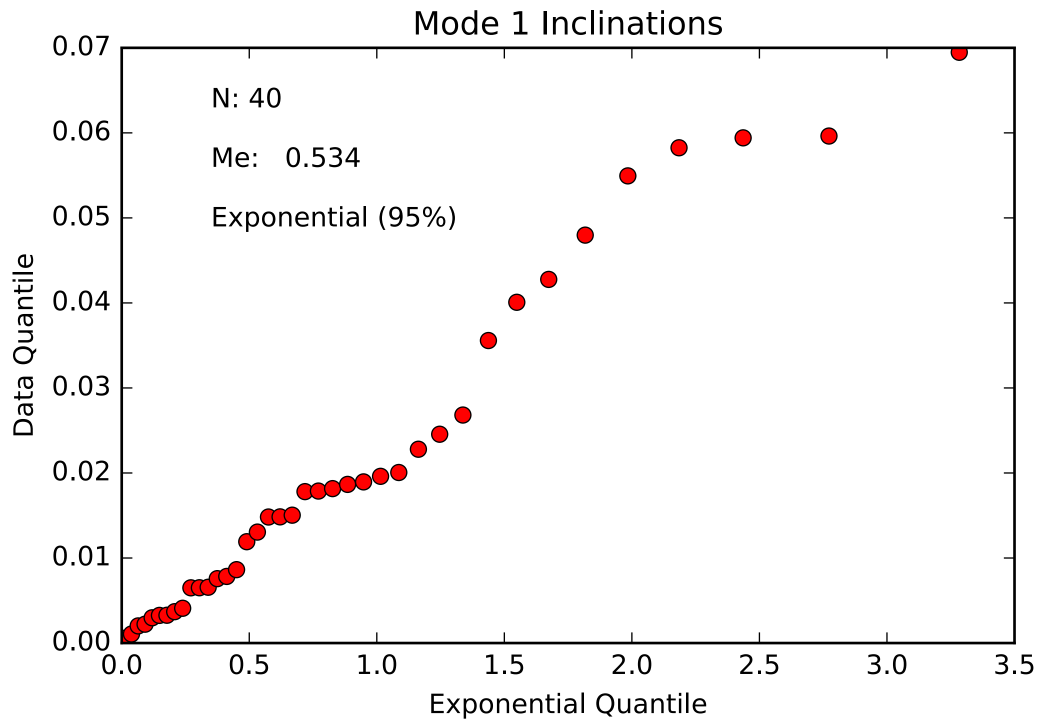
As is apparent in the plots above, SLB paleomagnetic directions fall into two distinct populations. We organize these into lower and higher latitude poles below.

```
In [13]: SLB_Data_low_lat.head()

Out[13]: VGP Site: SLB01
         VGP Site: SLB02
         VGP Site: SLB10
         VGP Site: SLB11
         VGP Site: SLB12

In [14]: SLB_Data_hi_lat.head()

Out[14]: VGP Site: SLB03
         VGP Site: SLB04
         VGP Site: SLB05
         VGP Site: SLB06
         VGP Site: SLB07
```

```
In [15]: # Test whether new SLB data qualifies as a Fisherian distribution
         ipmag.fishqq(SLB_Data['vgp_lon'].tolist(),SLB_Data['vgp_lat'].tolist())
```

```
Out[15]: {'Dec': 189.55557939703078,
          'Inc': 26.745255032788066,
          'Me': 0.53371584170306374,
          'Me_critical': 1.094,
          'Mode': 'Mode 1',
          'Mu': 1.7737870824702469,
          'Mu_critical': 1.207,
          'N': 40,
          'Test_result': 'Fisherian model rejected'}
```

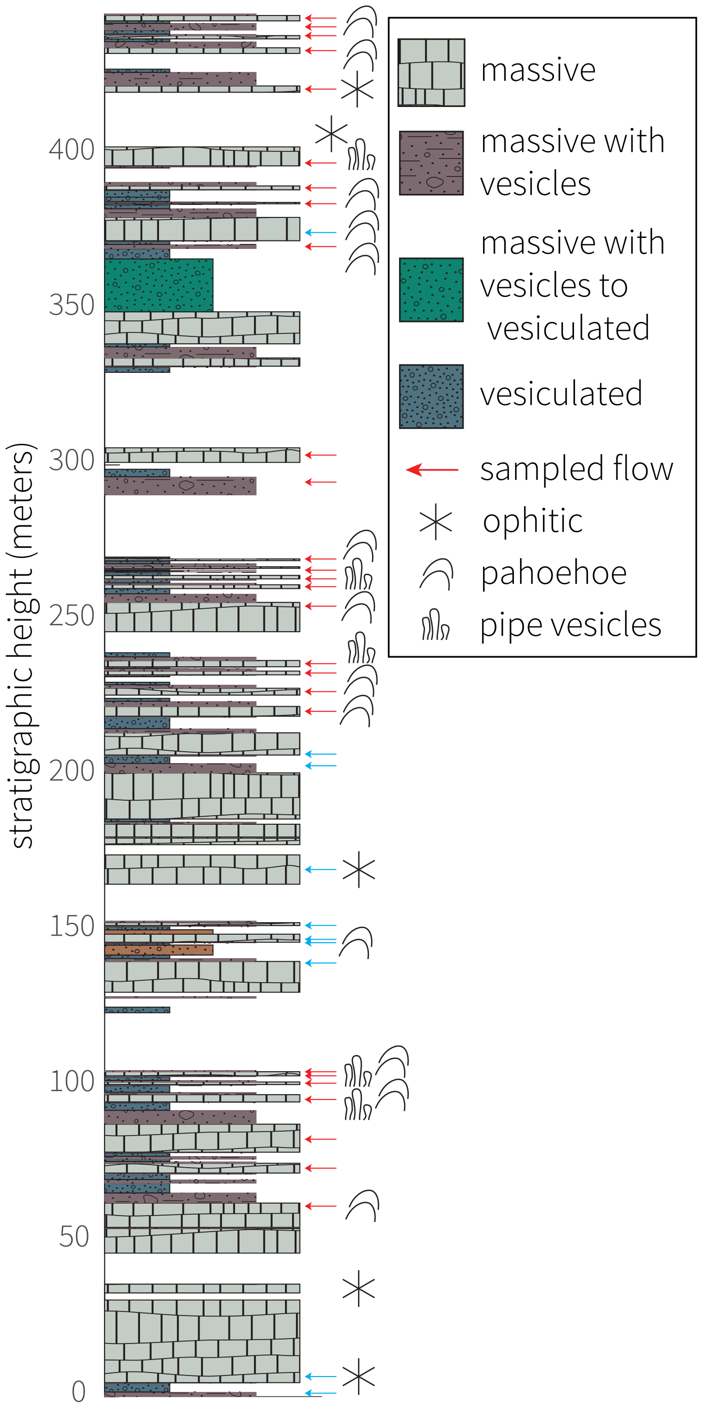




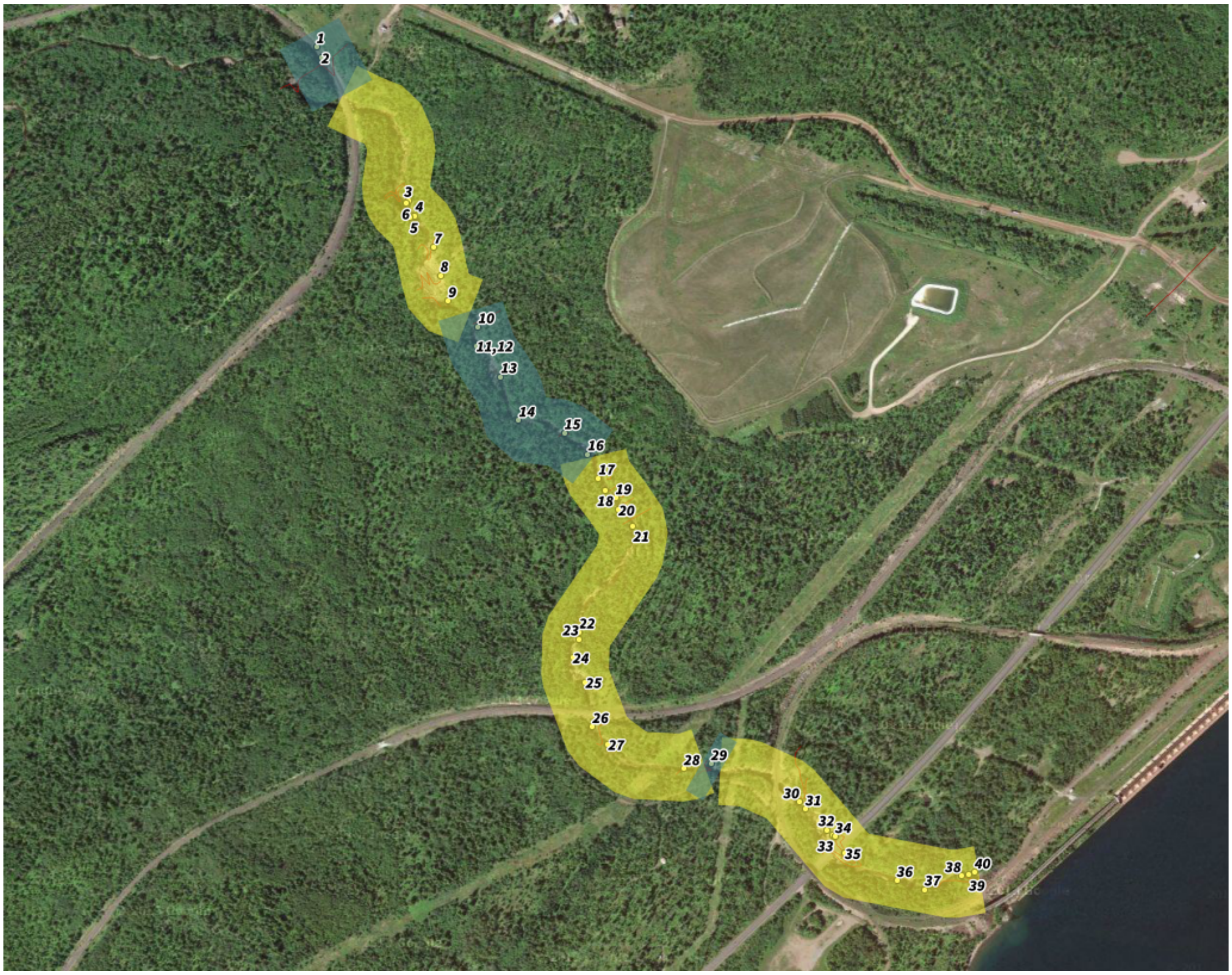
When the two populations are grouped, the combined SLB data fail the Fisher quantile-quantile test above. Possible explanations for this non-Fisherian distribution are explored below.

2.2.1 Divergent VGP populations from new SLB data

Rapid multiple-eruption episodes that are causing excursional behavior to be overrepresented in the dataset This explanation would likely predict that the more southerly VGPs are excursional and therefore stratigraphically grouped. However, approaching the problem from a stratigraphic standpoint reveals that the low latitude VGPs are encountered in multiple parts of the stratigraphic succession. Since these VGPS are coming from flows throughout SLB stratigraphy (some sequential flows, others isolated) if the southerly population is excursional it would have needed to be a direction that was returned to repeatedly by the field. In the stratigraphic column below, the blue arrows mark sampled flows that exhibit low latitude directions, and the red arrows mark sampled flows that exhibit high latitude directions. It is quite clear from this figure that there is no stratigraphic pattern of the anomalous paleomagnetic directions.



Additionally, a broader map view of the stack of lava flows highlights the discontinuity of the two VGP populations.

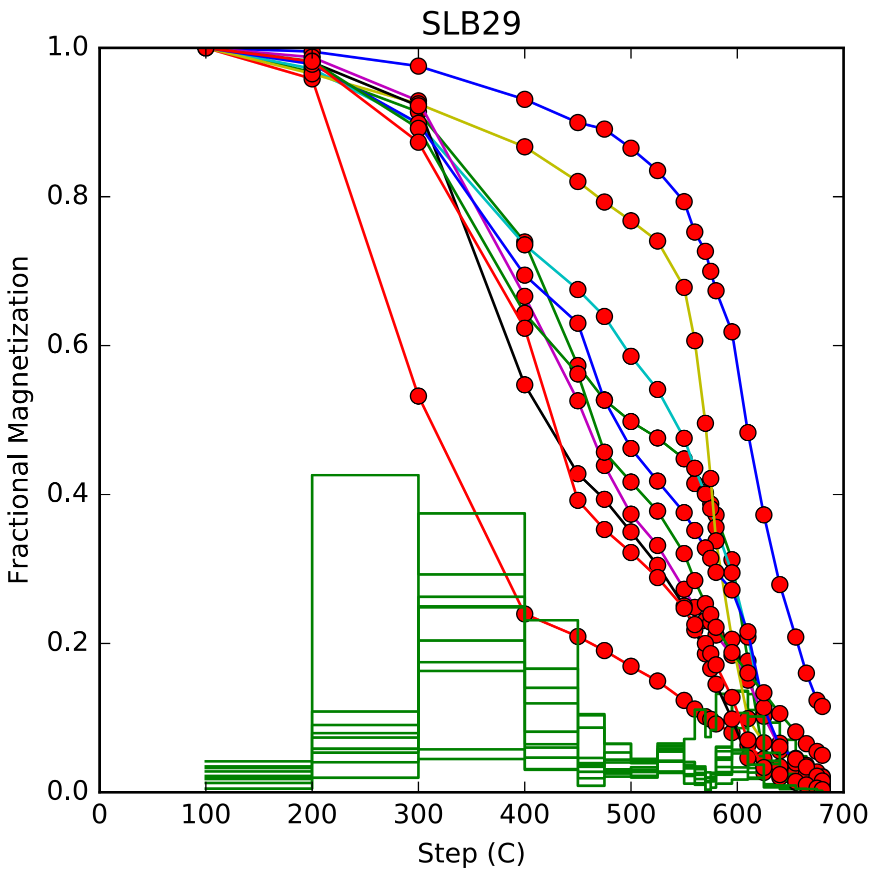


Rock magnetism A difference in the magnetic mineralogy of certain flows could indicate a chemical remagnetization representing the local geomagnetic field at a later date leading to the more southerly VGPs. However, the specimens in the two VGP populations encompass the same range of demagnetization behaviors. Therefore a rock magnetic explanation for the divergent directions is not readily apparent within the data. The demagnetization data (averaged by flow) of both high and low latitude VGPs are presented below.

```
In [16]: high_lat_sites = SLB_Data_hi_lat.er_site_names.tolist()
         low_lat_sites = SLB_Data_low_lat.er_site_names.tolist()
```

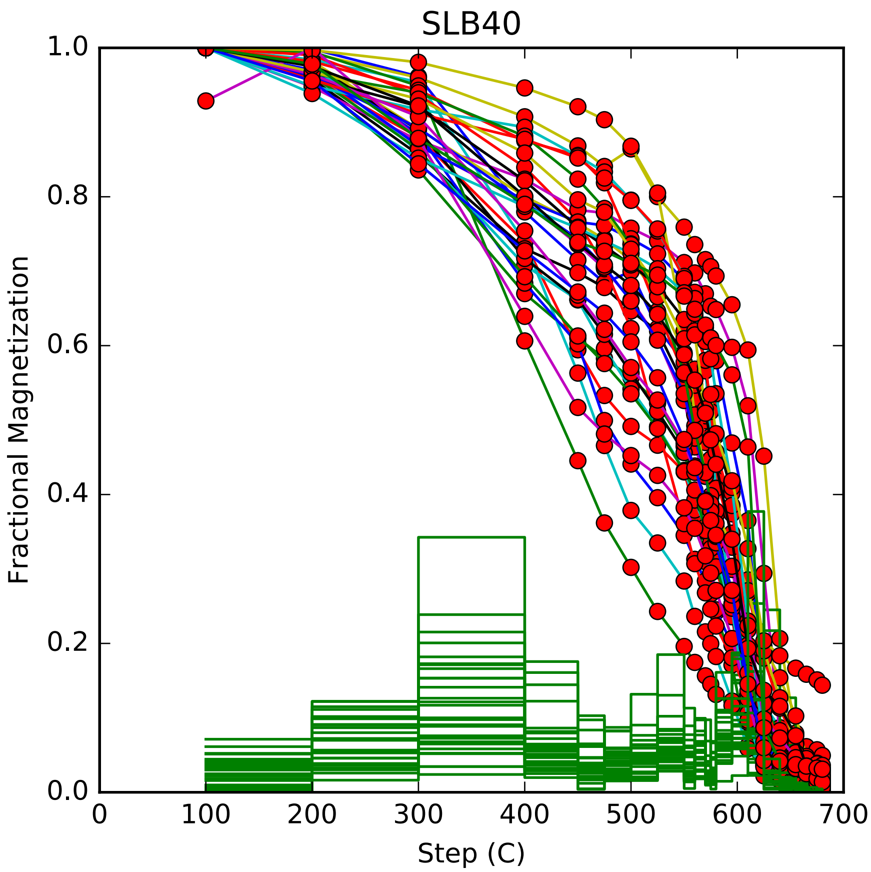
```
In [17]: ipmag.demag_magic('../Data/SLB/', plot_by='site', treat='T',
                           individual=low_lat_sites,average_measurements=True,
                           single_plot=True)
```

```
8204  records read from  ../Data/SLB/magic_measurements.txt
```




```
In [18]: ipmag.demag_magic('../Data/SLB/', plot_by='site', treat='T',
                           individual=high_lat_sites, average_measurements=True,
                           single_plot=True)
```

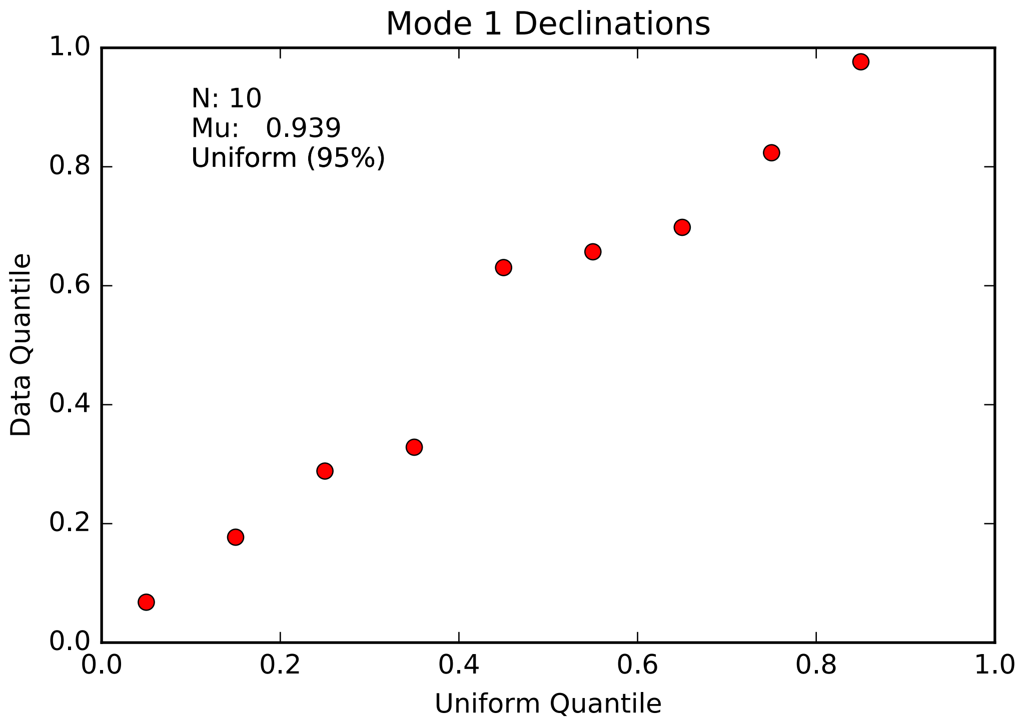
```
8204  records read from  ../Data/SLB/magic_measurements.txt
```

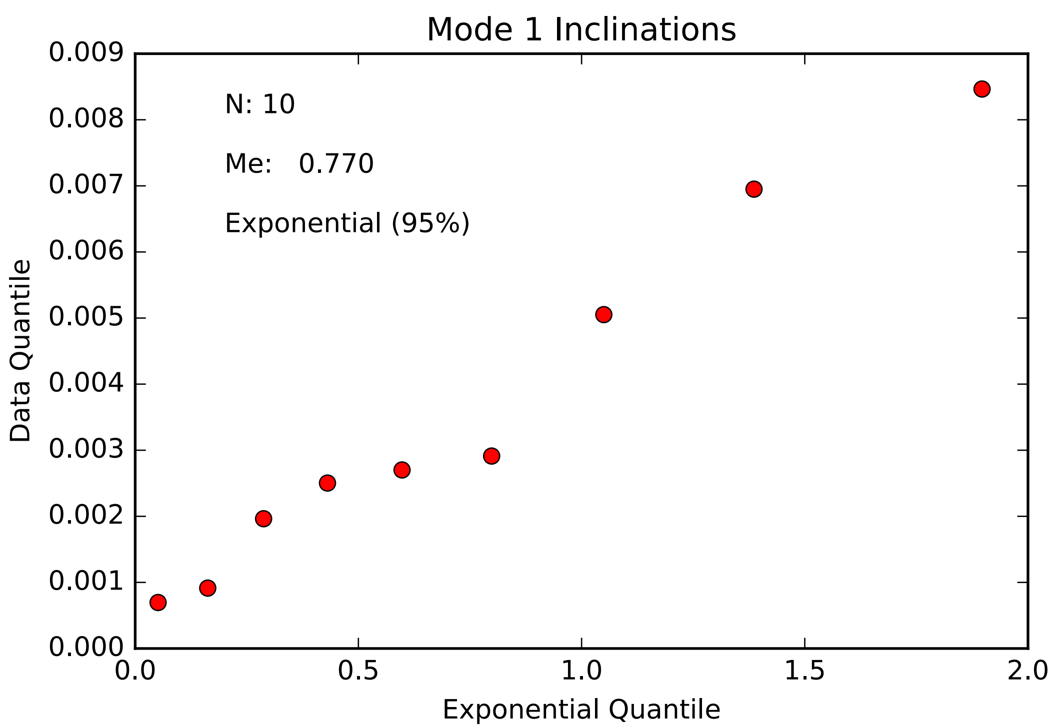


Patterns in the distribution and/or secular variation of poles Considered individually, both VGP populations are Fisher-distributed.

```
In [19]: ipmag.fishqq(SLB_Data_low_lat['vgp_lon'].tolist(),SLB_Data_low_lat['vgp_lat'].tolist())
```

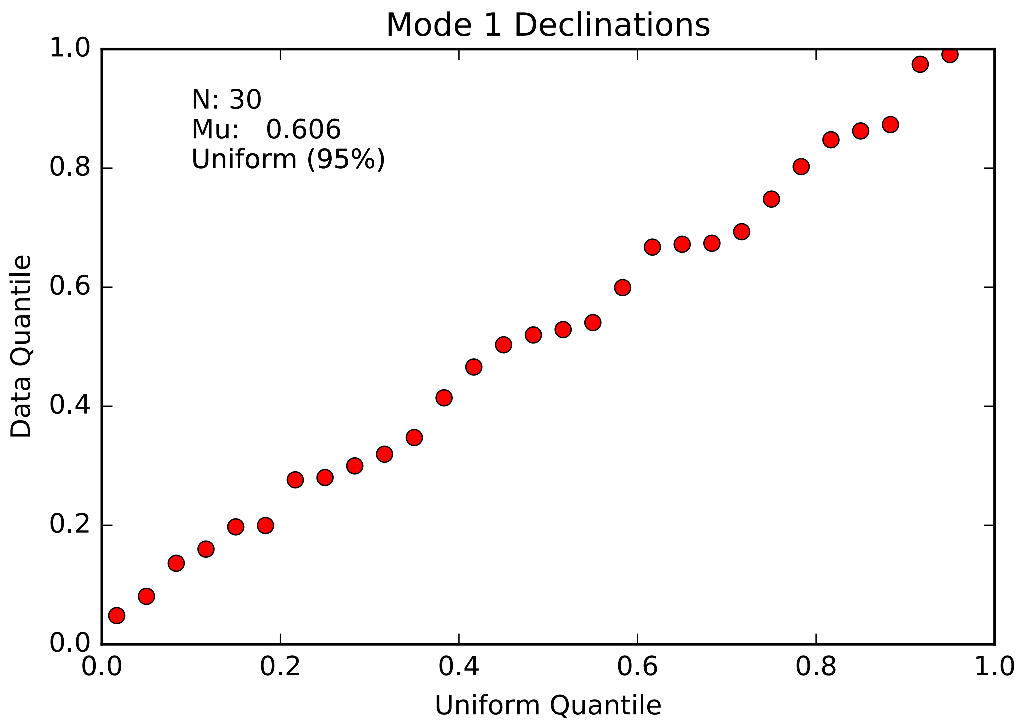
```
Out[19]: {'Dec': 189.87618031371633,
          'Inc': 9.225749891843499,
          'Me': 0.77027561013556023,
          'Me_critical': 1.094,
          'Mode': 'Mode 1',
          'Mu': 0.93893808899861431,
          'Mu_critical': 1.207,
          'N': 10,
          'Test_result': 'consistent with Fisherian model'}
```

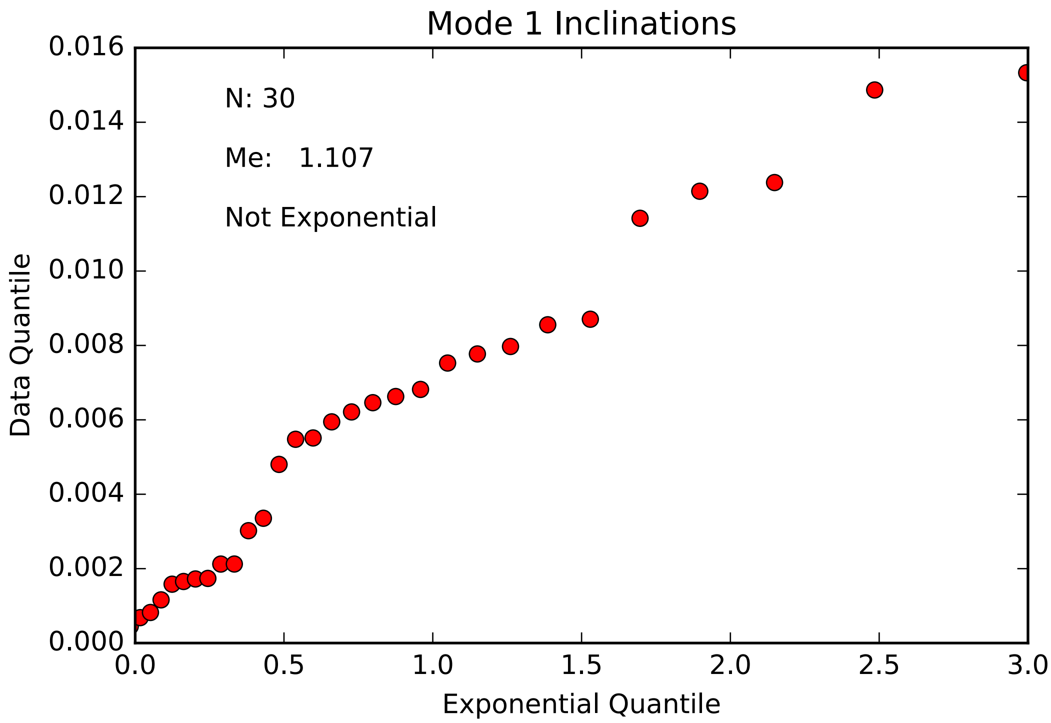





```
In [20]: ipmag.fishqq(SLB_Data_hi_lat['vgp_lon'].tolist(),SLB_Data_hi_lat['vgp_lat'].tolist())
```

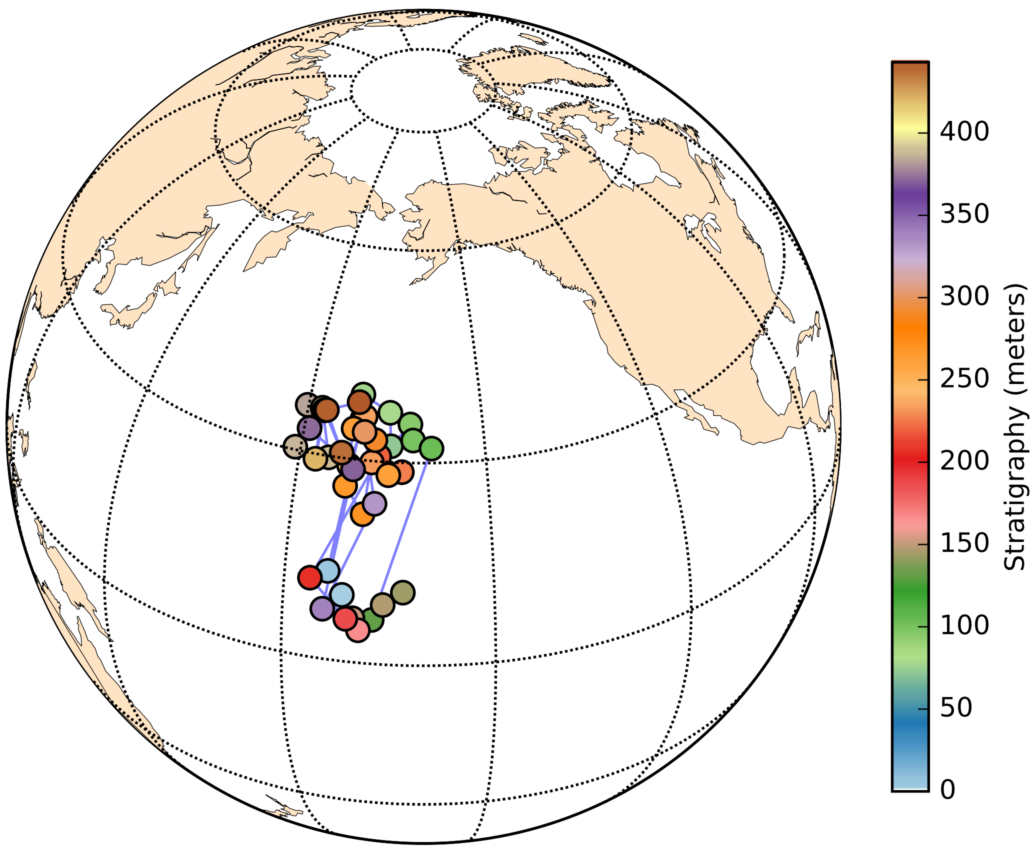
Out[20]: {'Dec': 189.43399983176758,
'Inc': 32.306644938683029,
'Me': 1.1071268307909761,
'Me_critical': 1.094,
'Mode': 'Mode 1',
'Mu': 0.60645097771482315,
'Mu_critical': 1.207,
'N': 30,
'Test_result': 'Fisherian model rejected'}





One last way to approach this problem is to inspect the path of the paleomagnetic poles being traced out by secular variation. We can look for potential patterns in the path of the poles through time by considering each cooling unit's stratigraphic placement.

```
In [21]: # upload strat heights
         SLB_strat_df = pd.read_csv('../Data/SLB/er_sites.txt', sep='\t', skiprows=1,
                                    usecols=['er_site_name', 'site_height'])
         SLB_strat_df = SLB_strat_df.set_index('er_site_name')

         m = pole_figure_appearance()

         centerlon, centerlat = m(SLB_Data['vgp_lon'].tolist(),SLB_Data['vgp_lat'].tolist())
         m.plot(centerlon, centerlat, alpha=0.5, zorder=1)
         m.scatter(centerlon, centerlat, marker='.', s=300.0,
                   c = SLB_strat_df.site_height.tolist(), cmap='Paired', zorder=2)

         plt.colorbar(label='Stratigraphy (meters)', shrink=0.7)
         plt.show()
```



From this view, we notice the relatively tight grouping of poles among certain stratigraphic intervals as they trace out sections of the entire distribution. However, there is still no apparent explanation for the occasional excursion of the geomagnetic pole to lower latitudes. It therefore seems likely that the divergence of these two VGP populations can be attributed to a recurrent geomagnetic excursion rather than a post-emplacement offset of magnetizations. We also note the similarity of this anomalous behavior to the distribution of paleomagnetic poles from the Lake Shore Traps (see below).

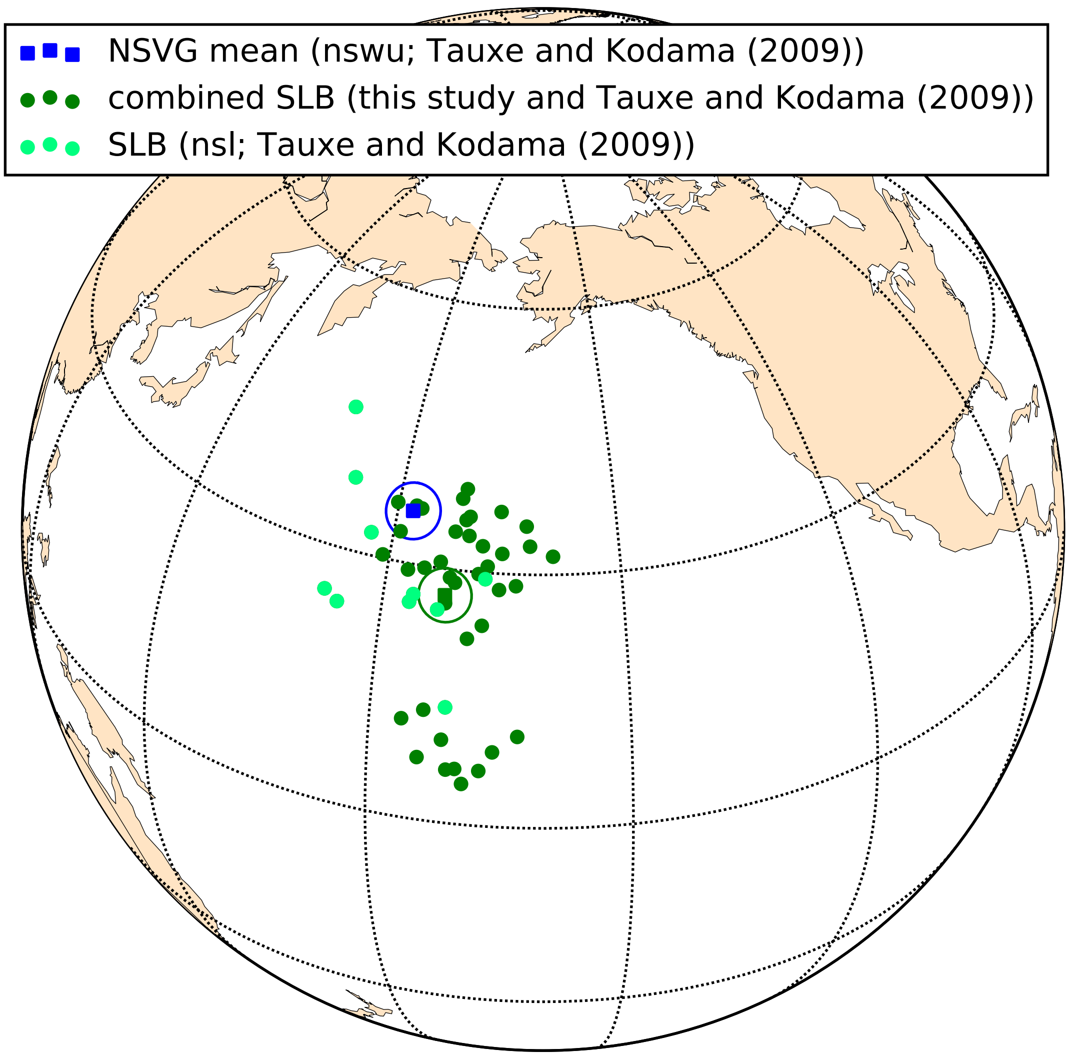
2.3 Combine new SLB data with Tauxe and Kodama (2009)

```
In [22]: combined_SLB_lon = NSVG_nsl['pole_lon'].tolist() + SLB_Data['vgp_lon'].tolist()
         combined_SLB_lat = NSVG_nsl['pole_lat'].tolist() + SLB_Data['vgp_lat'].tolist()
         combined_SLB_mean = ipmag.fisher_mean(combined_SLB_lon,combined_SLB_lat)
```

```
In [23]: m = pole_figure_appearance()
         ipmag.plot_pole(m,NSVG_nswu_mean['dec'],NSVG_nswu_mean['inc'],
                         NSVG_nswu_mean['alpha95'], marker='s',color='b',
                         label='NSVG mean (nswu; Tauxe and Kodama (2009))')
         ipmag.plot_vgp(m,combined_SLB_lon,combined_SLB_lat, color='g',
                        label='combined SLB (this study and Tauxe and Kodama (2009))')
         ipmag.plot_vgp(m,NSVG_nsl['pole_lon'].tolist(),NSVG_nsl['pole_lat'].tolist(),
                        color='springgreen',label='SLB (nsl; Tauxe and Kodama (2009))')
         ipmag.plot_pole(m,combined_SLB_mean['dec'],combined_SLB_mean['inc'],
                         combined_SLB_mean['alpha95'], marker='s',color='g')
```

```
ipmag.print_pole_mean(combined_SLB_mean)

plt.legend()
plt.savefig('./Code_output/NSVG_SLB_poles_combined.pdf')
plt.show()
```

```
Plong: 187.8  Plat: 27.1
Number of directions in mean (n): 50
Angular radius of 95% confidence (A_95): 3.0
Precision parameter (k) estimate: 46.5
```

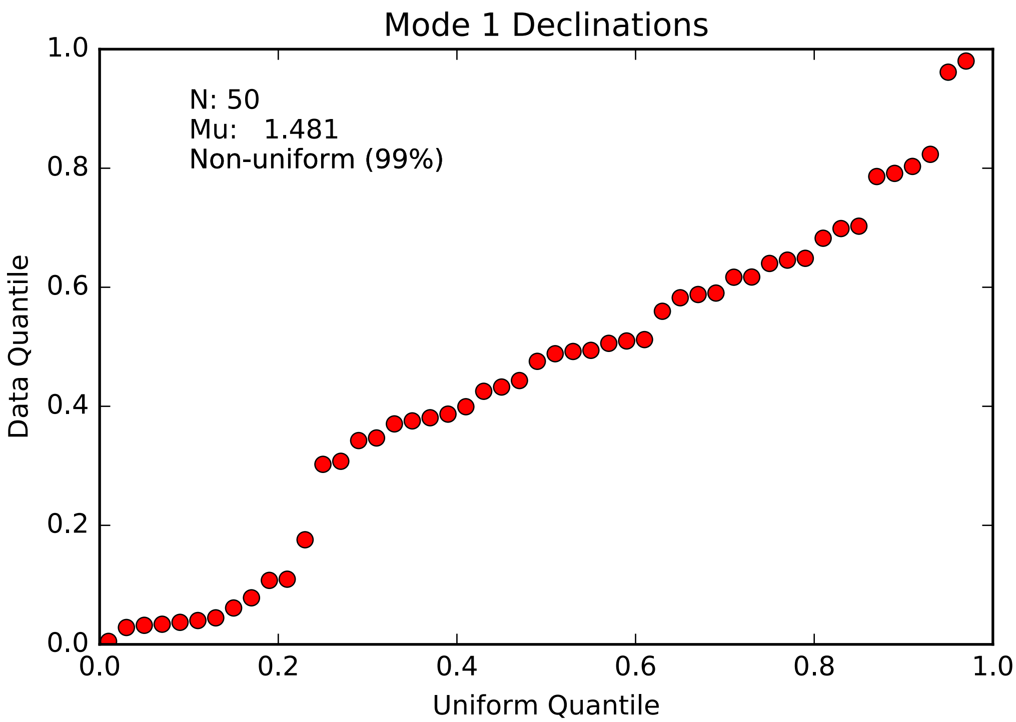


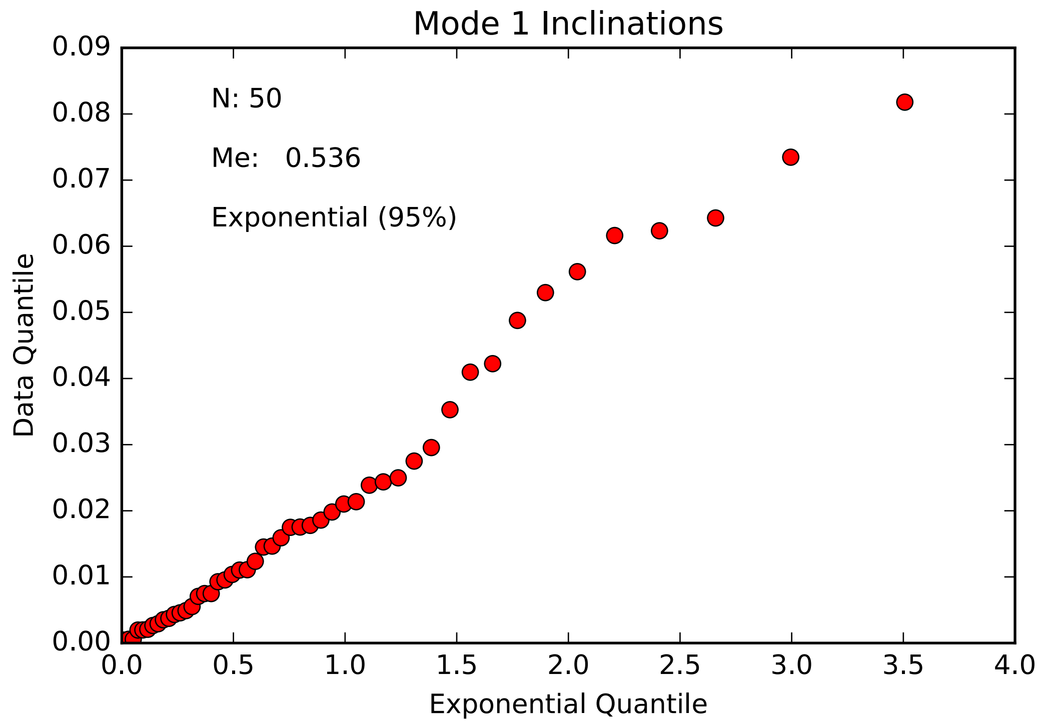
This combined mean still reflects a non-Fisherian distribution, as seen below.

In [24]:
```
ipmag.fishqq(combined_SLB_lon,combined_SLB_lat)
```

Out[24]:
```
{'Dec': 187.79191234206988,
 'Inc': 27.199685356762615,
 'Me': 0.53556909069644332,
```

'Me_critical': 1.094,
'Mode': 'Mode 1',
'Mu': 1.4805683534761531,
'Mu_critical': 1.207,
'N': 50,
'Test_result': 'Fisherian model rejected'}





3 Lake Shore Traps

In [25]:
```
Diehl1994a_LST_Data_all=pd.read_csv('../Data/Previous_studies/Diehl1994a_data.csv',sep=',')
#Kulakov2013 reported data for the flow LST28 that supersedes
#the Diehl direction which should accordingly be dropped
Diehl1994a_LST_Data=Diehl1994a_LST_Data_all.drop(17)
Diehl1994a_LST_Data.reset_index(inplace=True)
Diehl1994a_LST_Data.head()
```

Out[25]: Click to view the data of Diehl and Haig (1994)

If you are having trouble opening the local file via this link, the file can be found in the Data Repository under /Data/Previous_studies/Diehl1994a_data.csv and opened manually.

In [26]:
```
Kulakov2013a_LST_Data=pd.read_csv('../Data/Previous_studies/Kulakov2013a_data.csv',sep=',')
Kulakov2013a_LST_Data.head()
```

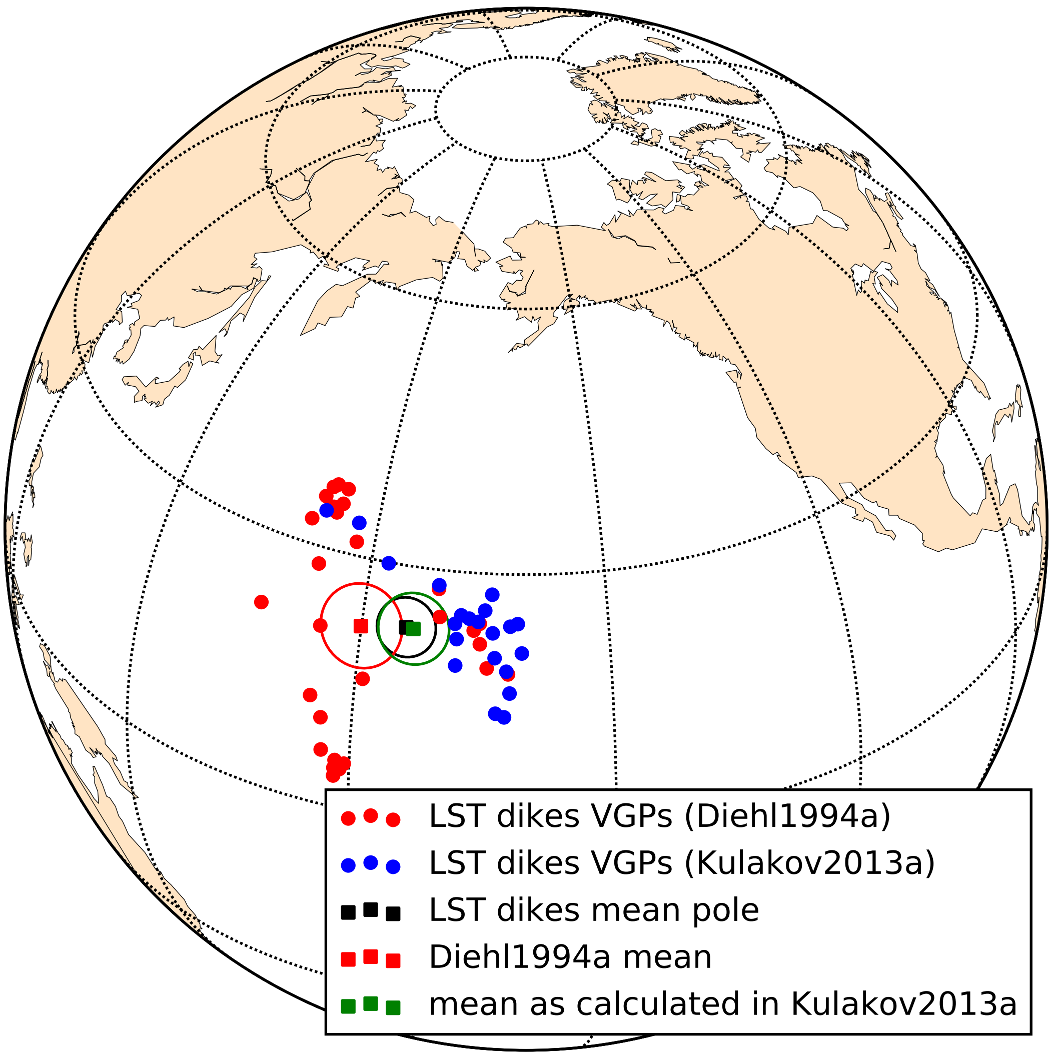
Out[26]: Click to view the data of Kulakov et al. (2013)

If you are having trouble opening the local file via this link, the file can be found in the Data Repository under /Data/Previous_studies/Kulakov2013a_data.csv and opened manually.

In [27]:
```
LST_Diehl_VGPs=ipmag.make_di_block(Diehl1994a_LST_Data['vgp_lon'],
                                   Diehl1994a_LST_Data['vgp_lat'])
```

```
        LST_Kulakov_VGPs=ipmag.make_di_block(Kulakov2013a_LST_Data['vgp_lon'],
                                             Kulakov2013a_LST_Data['vgp_lat'])

        LST_VGPs=np.concatenate((LST_Diehl_VGPs,LST_Kulakov_VGPs))

        LST_Diehl_mean = pmag.fisher_mean(LST_Diehl_VGPs)
        LST_Kulakov_mean=pmag.fisher_mean(LST_Kulakov_VGPs)
        LST_all_mean=pmag.fisher_mean(LST_VGPs)
```

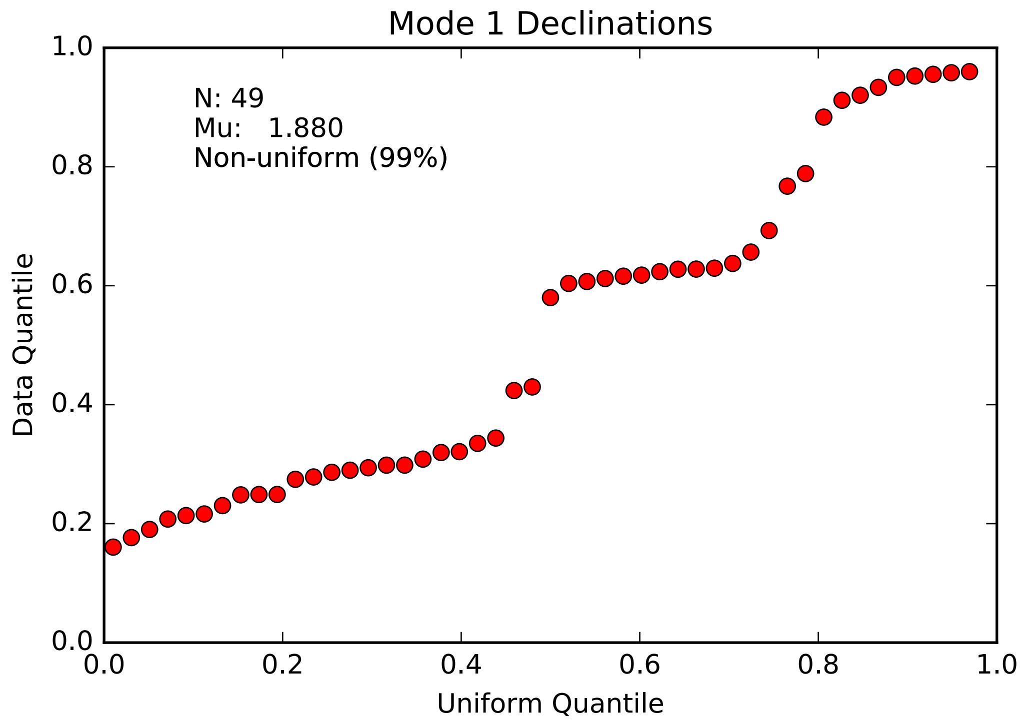
```
In [28]: m = Basemap(projection='ortho',lat_0=35,lon_0=200,resolution='c',
                     area_thresh=50000)
         pole_figure_appearance()

         ipmag.plot_vgp(m,Diehl1994a_LST_Data['vgp_lon'].tolist(),
                         Diehl1994a_LST_Data['vgp_lat'].tolist(),
                        label='LST dikes VGPs (Diehl1994a)',color='r')
         ipmag.plot_vgp(m,Kulakov2013a_LST_Data['vgp_lon'].tolist(),
                         Kulakov2013a_LST_Data['vgp_lat'].tolist(),
                        label='LST dikes VGPs (Kulakov2013a)',color='b')
         ipmag.plot_pole(m,LST_all_mean['dec'],LST_all_mean['inc'],
                          LST_all_mean['alpha95'], label='LST dikes mean pole',marker='s')
         ipmag.plot_pole(m,LST_Diehl_mean['dec'],LST_Diehl_mean['inc'],
                          LST_Diehl_mean['alpha95'], color='r',
                          label='Diehl1994a mean',marker='s')
         #from Kulakov2013a
         ipmag.plot_pole(m,186.4,23.1,4.0,color='g',
                          label='mean as calculated in Kulakov2013a',marker='s')

         plt.legend(loc=4)
         plt.savefig('Code_output/LST_vgps.pdf')
         plt.show()
```

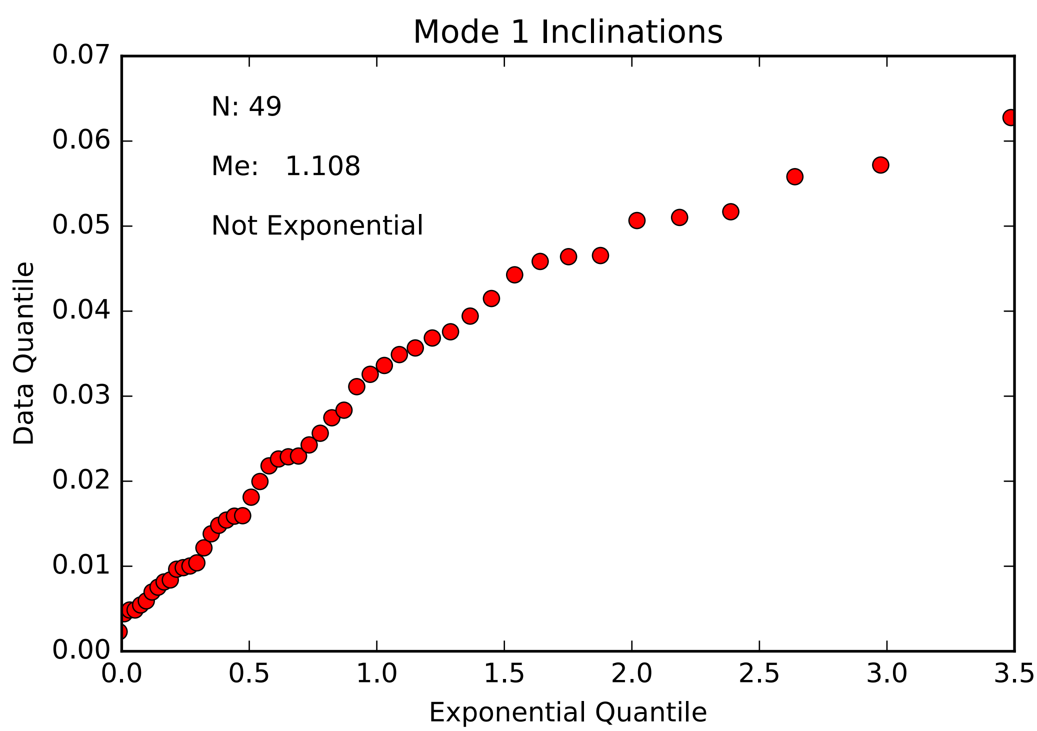


Here we test whether the distribution of Lake Shore Traps VGPs are Fisherian.

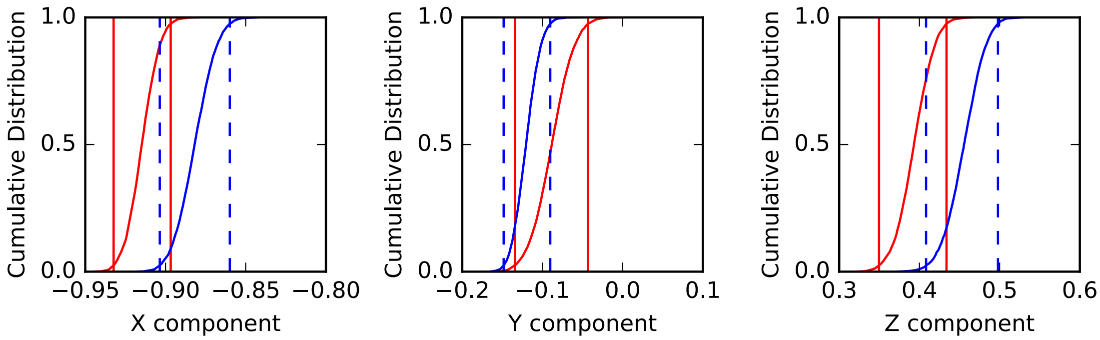
```
In [29]: ipmag.fishqq(di_block=LST_VGPs)
```

```
Out[29]: {'Dec': 185.61848690207148,
          'Inc': 23.207239038463833,
          'Me': 1.1080580586218494,
          'Me_critical': 1.094,
          'Mode': 'Mode 1',
          'Mu': 1.8795789739707827,
          'Mu_critical': 1.207,
          'N': 49,
          'Test_result': 'Fisherian model rejected'}
```


Mode 1 Declinations
N: 49
Mu: 1.880
Non-uniform (99%)
Data Quantile
Uniform Quantile
0.0
0.2
0.4
0.6
0.8
1.0

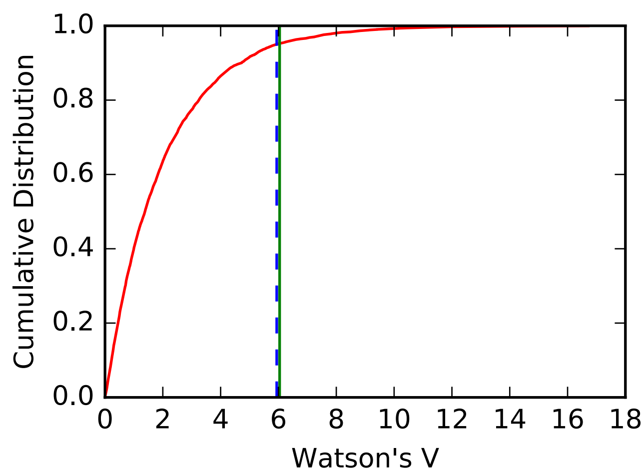

Mode 1 Inclinations
N: 49
Me: 1.108
Not Exponential
Data Quantile
Exponential Quantile
0.00
0.01
0.02
0.03
0.04
0.05
0.06
0.07
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5

```
In [30]: ipmag.common_mean_bootstrap(LST_VGPs,ipmag.make_di_block(combined_SLB_lon,
                                                                   combined_SLB_lat))
```




```
In [31]: ipmag.common_mean_watson(LST_VGPs,
                                  ipmag.make_di_block(combined_SLB_lon,combined_SLB_lat),
                                  plot='yes')
```

```
Results of Watson V test:

Watson's V:           6.0
Critical value of V:  5.9
"Fail": Since V is greater than Vcrit, the two means can
be distinguished at the 95% confidence level.

M&M1990 classification:

Angle between data set means: 4.4
Critical angle for M&M1990:   4.4
```



4 Michipicoten Island

4.1 Palmer and Davis (1987) data

In [32]:
```
Palmer1987_data=pd.read_csv('../Data/Previous_studies/Palmer1987a_data_combined_sites.csv')
ipmag.vgp_calc(Palmer1987_data)
```

Out[32]: Click to view the data of Palmer and Davis (1987)

If you are having trouble opening the local file via this link, the file can be found in the Data Repository under /Data/Previous_studies/Palmer1987a_data_combined_sites.csv and opened manually.

The code below splits this dataframe into sites from older main stage basalts that have been correlated to the Mamainse Point Formation, sites from the instrusive suite, and sites from the Michipicoten Island Formation.

While not explicitly described as such, it appears that each of the sites sampled in Palmer and Davis (1987) are treated as single cooling units except for site 'KM' which is decribed as being collected over a stratigraphic interval containing multiple flows. As the site is not from a single cooling unit, data from this are excluded. Also, site 18 is described as coming from a fault bound block with anomalous steep dip where a structural attitude could not be determined and should be excluded.

Palmer and Davis (1987) sampled the Channel Lake Member, the Cuesta Member, and the Davieaux Island Member at multiple sites from which VGPs were found and used in the final calculation of the Michipicoten Island Formation mean paleomagnetic pole. However, field observations reveal that some of these sites represented the redundant sampling of a single cooling unit, meaning that several VGPs that were originally treated as independent should have been combined with one another in the final calculation. Here we reduce the Palmer and Davis (1987) data set to satisfy these concerns. The reduced number of sites for each Michipicoten Island Formation member is shown below.

In [33]:
```
print 'Number of Cuesta Member flows: ',
len(Palmer1987_data.loc[Palmer1987_data['member']=='Cuesta_Member'])
print 'Number of Channel Lake Member flows: ',
len(Palmer1987_data.loc[Palmer1987_data['member']=='Channel_Lake_Member'])
print 'Number of Quebec Harbour Member flows: ',
len(Palmer1987_data.loc[Palmer1987_data['member']=='Quebec_Harbour_Member'])
print 'Number of South Shore Member flows: ',
len(Palmer1987_data.loc[Palmer1987_data['member']=='South_Shore_Member'])
print 'Number of Davieaux Island Member flows: ',
len(Palmer1987_data.loc[Palmer1987_data['member']=='Davieaux_Island_Member'])
print 'Total number of cooling units from the Michipicoten Island Formation: ',
len(Palmer1987_data.loc[Palmer1987_data['formation']=='Michipicoten_Island'])
```

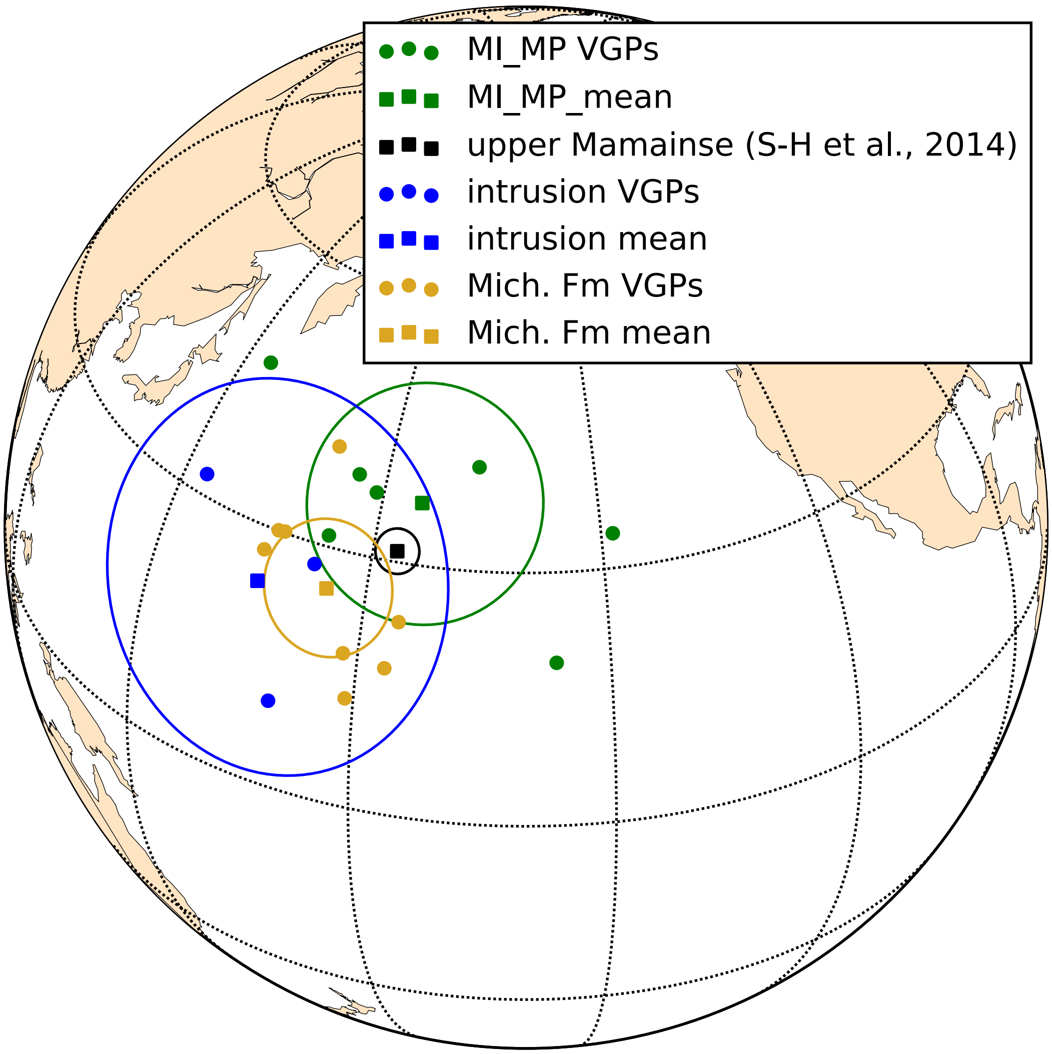
```
Number of Cuesta Member flows:  2
Number of Channel Lake Member flows:  1
Number of Quebec Harbour Member flows:  1
Number of South Shore Member flows:  3
Number of Davieaux Island Member flows:  1
Total number of cooling units from the Michipicoten Island Formation:  8
```

In [34]:
```
MI_Mamainse_Fm=Palmer1987_data.ix[Palmer1987_data['formation'] == 'Quebec_Mine']
MI_Mamainse_Fm.reset_index(inplace=True, drop=True)
#make dataframe that excludes sites 18 and KM and reset the index of the dataframe
#MI_Mamainse_Fm=MI_Mamainse_Fm_all.drop([1,8])
MI_Mamainse_Fm.reset_index(inplace=True)
MI_Mamainse_Fm_VGPs = ipmag.make_di_block(MI_Mamainse_Fm['vgp_lon'],
                MI_Mamainse_Fm['vgp_lat'])
MI_Mamainse_Fm_mean = pmag.fisher_mean(MI_Mamainse_Fm_VGPs)
```

```python
print 'Michipicoten Island Mamainse Formation mean:'
ipmag.print_pole_mean(MI_Mamainse_Fm_mean)

MI_intrusions=Palmer1987_data.ix[Palmer1987_data['formation'] == 'intrusive']
MI_intrusions.reset_index(inplace=True, drop=True)
MI_intrusions_VGPs = ipmag.make_di_block(MI_intrusions['vgp_lon'],
            MI_intrusions['vgp_lat'])
MI_intrusions_mean = pmag.fisher_mean(MI_intrusions_VGPs)
print '\nMichipicoten Island instrusions mean:'
ipmag.print_pole_mean(MI_intrusions_mean)

Michipicoten_Island_Fm=Palmer1987_data.ix[Palmer1987_data['formation'] ==
        'Michipicoten_Island']
Michipicoten_Island_Fm.reset_index(inplace=True, drop=True)
Michipicoten_Island_Fm_VGPs = ipmag.make_di_block(Michipicoten_Island_Fm['vgp_lon'],
        Michipicoten_Island_Fm['vgp_lat'])
Michipicoten_Island_Fm_mean = pmag.fisher_mean(Michipicoten_Island_Fm_VGPs)
print '\nMichipicoten Island Formation mean:'
ipmag.print_pole_mean(Michipicoten_Island_Fm_mean)

m = pole_figure_appearance()

ipmag.plot_vgp(m,MI_Mamainse_Fm['vgp_lon'].tolist(),MI_Mamainse_Fm['vgp_lat'].tolist(),
               label='MI_MP VGPs',color='g')
ipmag.plot_pole(m,MI_Mamainse_Fm_mean['dec'],MI_Mamainse_Fm_mean['inc'],
                MI_Mamainse_Fm_mean['alpha95'],
                label='MI_MP_mean',marker='s',color='g')
ipmag.plot_pole(m,183.2,31.2,2.5,label='upper Mamainse (S-H et al., 2014)',marker='s')

ipmag.plot_vgp(m,MI_intrusions['vgp_lon'].tolist(),MI_intrusions['vgp_lat'].tolist(),
               label='intrusion VGPs',color='blue')
ipmag.plot_pole(m,MI_intrusions_mean['dec'],MI_intrusions_mean['inc'],
                MI_intrusions_mean['alpha95'],
                label='intrusion mean',marker='s',color='blue')

ipmag.plot_vgp(m,Michipicoten_Island_Fm['vgp_lon'].tolist(),
               Michipicoten_Island_Fm['vgp_lat'].tolist(),
               label='Mich. Fm VGPs',color='goldenrod')
ipmag.plot_pole(m,Michipicoten_Island_Fm_mean['dec'],Michipicoten_Island_Fm_mean['inc'],
                Michipicoten_Island_Fm_mean['alpha95'],
                label='Mich. Fm mean',marker='s',color='goldenrod')
plt.legend()
plt.show()
```

```
Michipicoten Island Mamainse Formation mean:
Plong: 185.6  Plat: 36.9
Number of directions in mean (n): 7
Angular radius of 95% confidence (A_95): 13.4
Precision parameter (k) estimate: 21.2

Michipicoten Island instrusions mean:
Plong: 165.7  Plat: 23.9
Number of directions in mean (n): 3
Angular radius of 95% confidence (A_95): 22.5
```

```
Precision parameter (k) estimate: 31.2

Michipicoten Island Formation mean:
Plong: 174.9  Plat: 25.5
Number of directions in mean (n): 8
Angular radius of 95% confidence (A_95): 7.7
Precision parameter (k) estimate: 53.2
```



In [35]: Michipicoten_Island_Fm

4.2 New data from the Michipicoten Island Formation

New paleomagnetic data from the Michipicoten Island Formation are imported below. We first separate data by *in situ* ("geo") and tilt-corrected ("tc") coordinates. We then further separate data by member ("SS" = South Shore Member basalts, "CM" = Cuesta Member andesite).

```
In [36]: All_Michi = pd.read_csv('../Data/Michipicoten/pmag_results.txt',
                                 sep='\t',skiprows=1)
```

```
All_Michi_geo = All_Michi.ix[All_Michi['pole_comp_name'] == 'HT']\
.ix[All_Michi['tilt_correction']==0]
All_Michi_tc = All_Michi.ix[All_Michi['pole_comp_name'] == 'HT']\
.ix[All_Michi['tilt_correction']==100]
All_Michi_tc.reset_index(inplace=True, drop=True)
All_Michi_geo.reset_index(inplace=True, drop=True)
```

Out[36]: Click to view MagIC file of new Michipicoten Island paleomagnetic data

If you are having trouble opening the local file via this link, the file can be found in the Data Repository under /Data/Michipicoten/pmag_results and opened manually.

In [37]:
```
SS_final = All_Michi.ix[All_Michi['er_site_names'].str.startswith('SS')]
SS_final_tc = All_Michi_tc.ix[All_Michi_tc['er_site_names'].str.startswith('SS')]
SS_final_geo = All_Michi_geo.ix[All_Michi_geo['er_site_names'].str.startswith('SS')]
SS_final_tc.head()
```

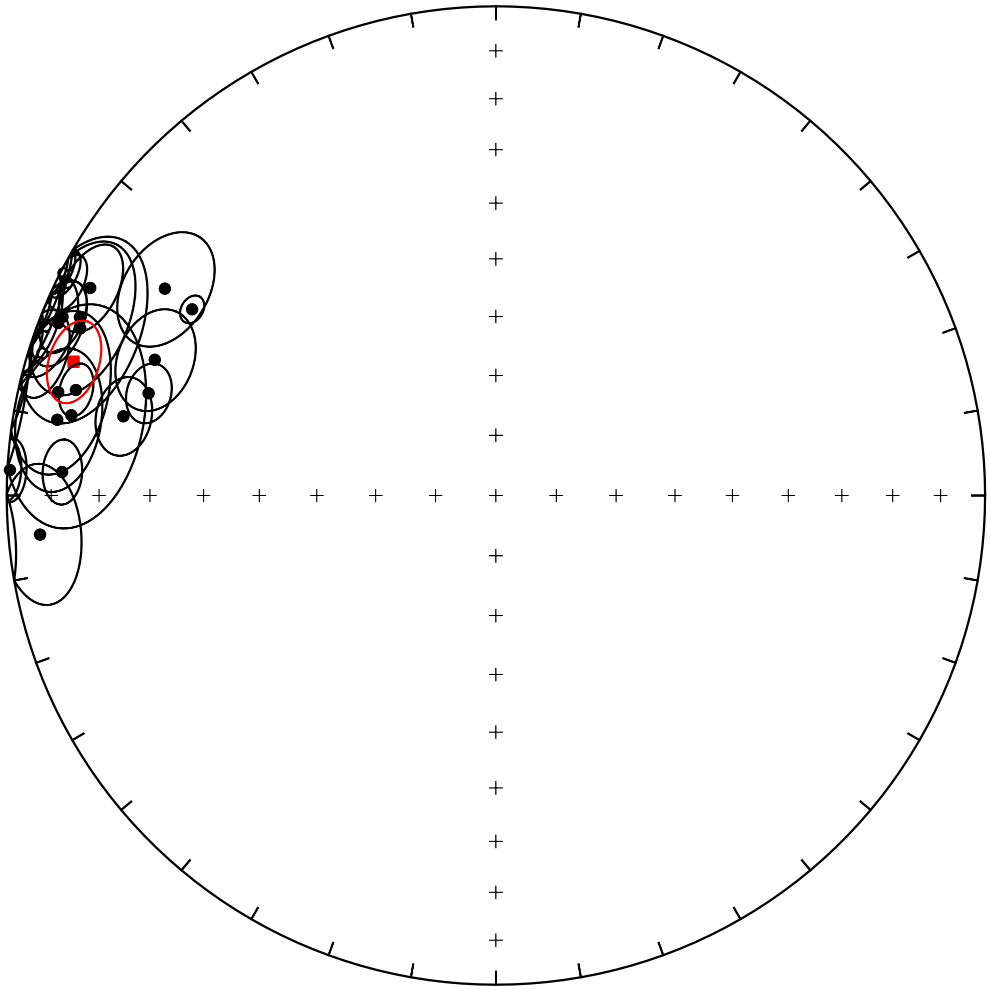
In [38]:
```
CM_final_tc = All_Michi_tc.ix[All_Michi_tc['er_site_names'].str.startswith('CM')]
CM_final_tc
```

Paleomagnetic site means of the South Shore basalts are plotted below with the overall mean.

In [39]:
```
plt.figure(num=1,figsize=(7,7))
ipmag.plot_net(1)

SS_dec = SS_final_tc['average_dec'].tolist()
SS_inc = SS_final_tc['average_inc'].tolist()
SS_a95 = SS_final_tc['average_alpha95'].tolist()

for i in range(len(SS_dec)):
    ipmag.plot_di_mean(SS_dec[i], SS_inc[i], SS_a95[i])

SS_mean_dir = ipmag.fisher_mean(SS_dec,SS_inc)

ipmag.plot_di_mean(SS_mean_dir['dec'],SS_mean_dir['inc'],
                   SS_mean_dir['alpha95'],marker='s', color='r')
ipmag.print_direction_mean(SS_mean_dir)
plt.savefig('Code_output/All_SS_data.pdf')
plt.title('Magnetization directions of South Shore basalts')
plt.clabel
plt.show()
```

```
Dec: 287.6  Inc: 10.4
Number of directions in mean (n): 21
Angular radius of 95% confidence (a_95): 5.4
Precision parameter (k) estimate: 35.4
```

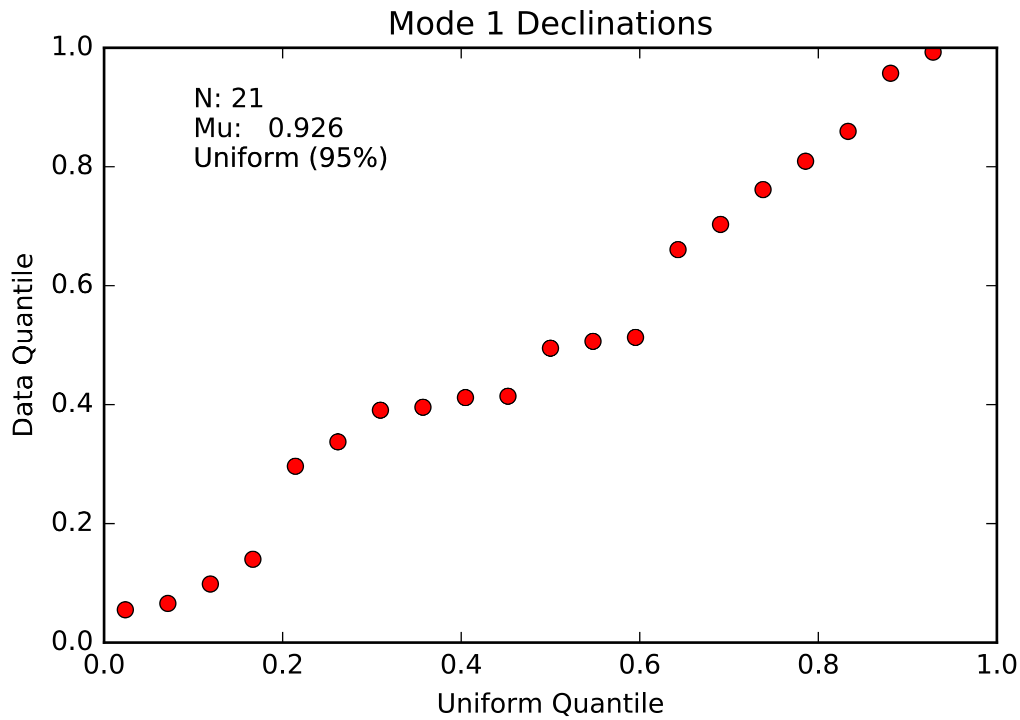
Magnetization directions of South Shore basalts

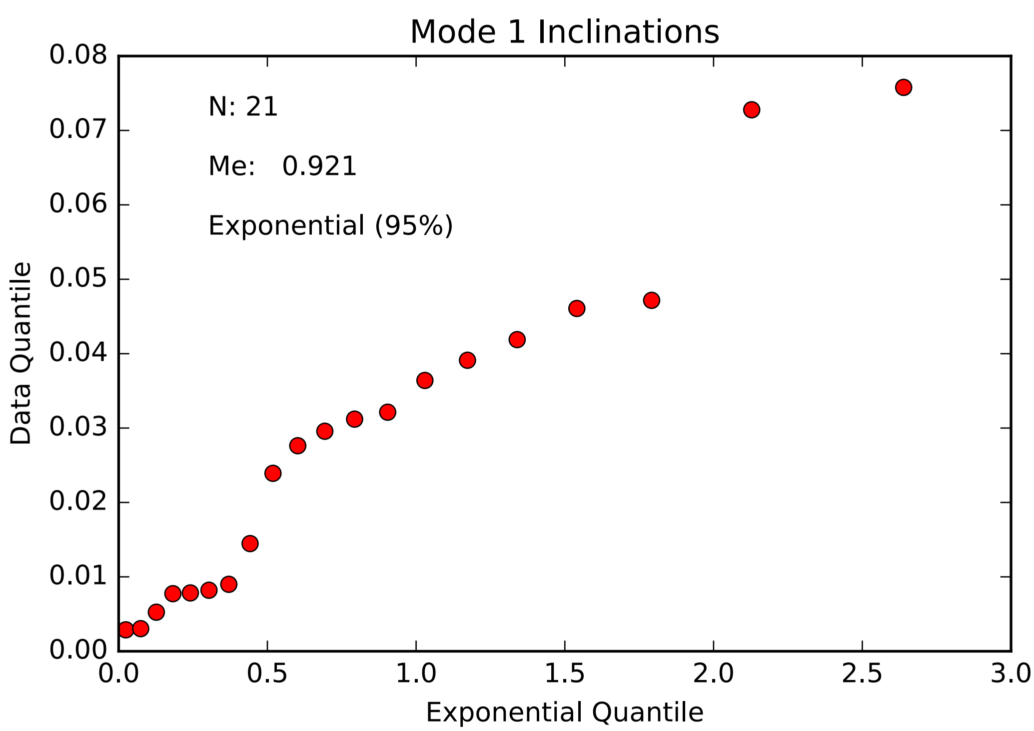


A Fisher Q-Q test reveals these data to be consistent with the Fisherian model.

In [40]: ipmag.fishqq(SS_dec, SS_inc)

Out[40]:
```
{'Dec': 287.60360855657012,
 'Inc': 10.297033320865525,
 'Me': 0.92123018046225358,
 'Me_critical': 1.094,
 'Mode': 'Mode 1',
 'Mu': 0.92607542329666181,
 'Mu_critical': 1.207,
 'N': 21,
 'Test_result': 'consistent with Fisherian model'}
```


Mode 1 Declinations
N: 21
Mu: 0.926
Uniform (95%)
Data Quantile
Uniform Quantile
0.0
0.2
0.4
0.6
0.8
1.0


Mode 1 Inclinations
N: 21
Me: 0.921
Exponential (95%)
Data Quantile
Exponential Quantile
0.00
0.01
0.02
0.03
0.04
0.05
0.06
0.07
0.08
0.0
0.5
1.0
1.5
2.0
2.5
3.0

The bedding is similar through the South Shore Member such that a fold test is inconclusive (see below). A more complete treatment of our structural analysis of the Michipicoten Island lava flows, including a walkthrough of how we determined tilt-correction for different stratigraphic sections of the South Shore Member, is included within the Data Repository as a separate Jupyter notebook.

```
In [41]: # get bedding info
         bedding = pd.read_csv('../Data/Michipicoten/er_samples.txt',
                               sep='\t',skiprows=1, usecols=['er_site_name',
                                                              'sample_bed_dip',
                                                              'sample_bed_dip_direction'])
```

```
In [42]: bedding = bedding.drop_duplicates()
         bedding = bedding.set_index('er_site_name')
```

```
In [43]: bedding.head()
```

Out[43]:

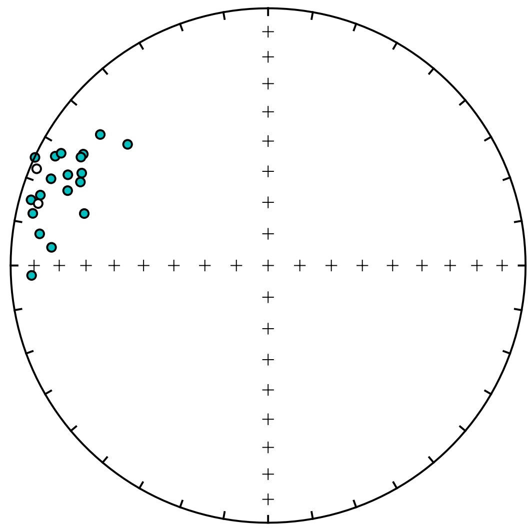
er_site_name	sample_bed_dip	sample_bed_dip_direction
CM1	31.8	171.9
CM2	25.5	177.7
SS1	20.0	237.6
SS10	20.0	237.6
SS11	20.0	237.6

```
In [44]: SS_diddd = []

         for site in bedding.index:
             if str(site) in SS_final_geo.er_site_names.tolist():
                 SS_diddd.append([float(SS_final_geo.loc[SS_final_geo['er_site_names']\
                                                         ==str(site)]['average_dec']),
                                  float(SS_final_geo.loc[SS_final_geo['er_site_names']\
                                                         ==str(site)]['average_inc']),
                                  bedding.loc[str(site), 'sample_bed_dip_direction'],
                                  bedding.loc[str(site), 'sample_bed_dip']])
```

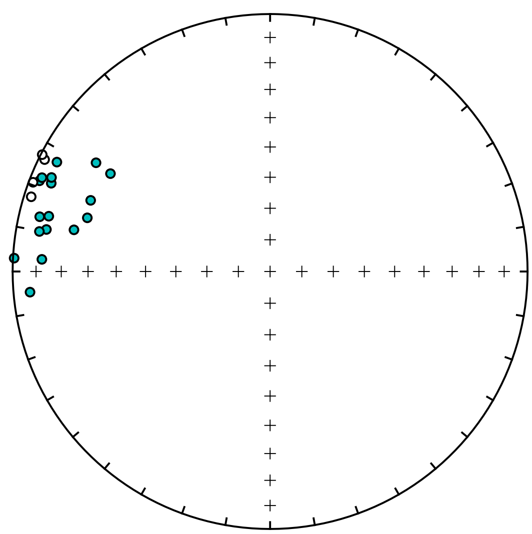
```
In [45]: ipmag.bootstrap_fold_test(np.array(SS_diddd))
```

```
doing  1000  iterations...please be patient...
```

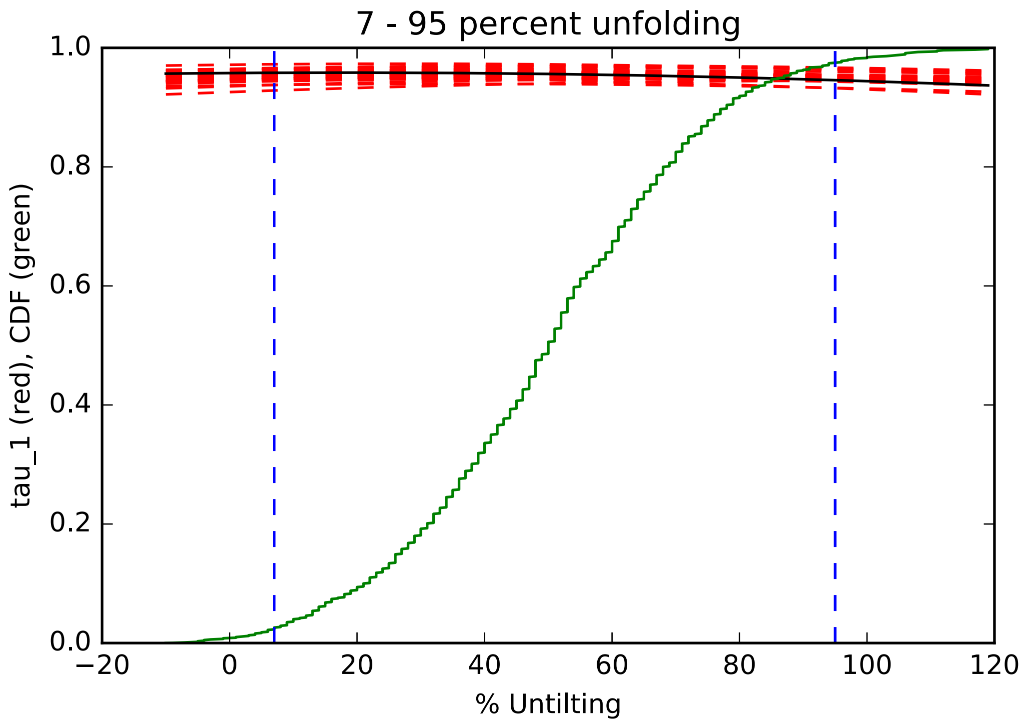
Geographic



Tilt-corrected

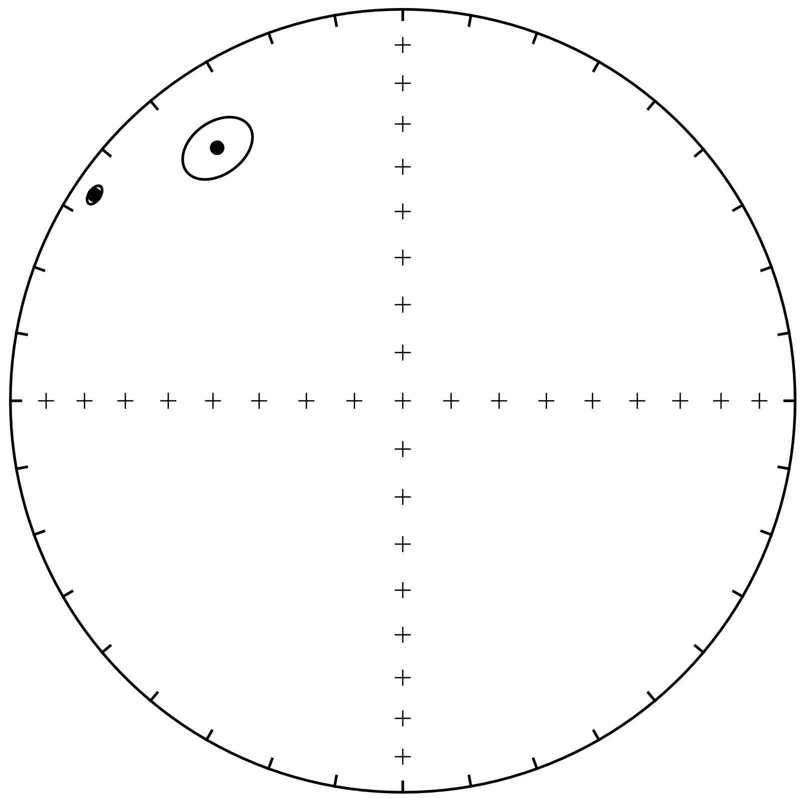


tightest grouping of vectors obtained at (95% confidence bounds):
7 - 95 percent unfolding
range of all bootstrap samples:
-10 - 119 percent unfolding



Paleomagnetic site means of the Cuesta Member andesite are plotted below. Only two Cuesta Member flows were identified in the field, so a Cuesta Member mean would not have much significance.

In [46]:
```
plt.figure(num=1,figsize=(5,5))
ipmag.plot_net(1)

CM_dec = CM_final_tc['average_dec'].tolist()
CM_inc = CM_final_tc['average_inc'].tolist()
CM_a95 = CM_final_tc['average_alpha95'].tolist()

for i in range(len(CM_dec)):
    ipmag.plot_di_mean(CM_dec[i], CM_inc[i], CM_a95[i])

plt.show()
```



4.3 Combine new Michipicoten data with Palmer and Davis (1987)

Given that we have obtained U-Pb dates for the West Sand Bay tuff and the Davieaux Island Member rhyolite, we are interested in generating a pole that is bracketed by these dates. The units between and including these dates are the West Sand Bay Member, the Quebec Harbor Member, the East Sand Bay Member, the South Shore Member and the Davieaux Island Member. Therefore we combine our data from the South Shore Member with the Quebec Harbour Member and Davieaux Island Member data from Palmer and Davis (1987).

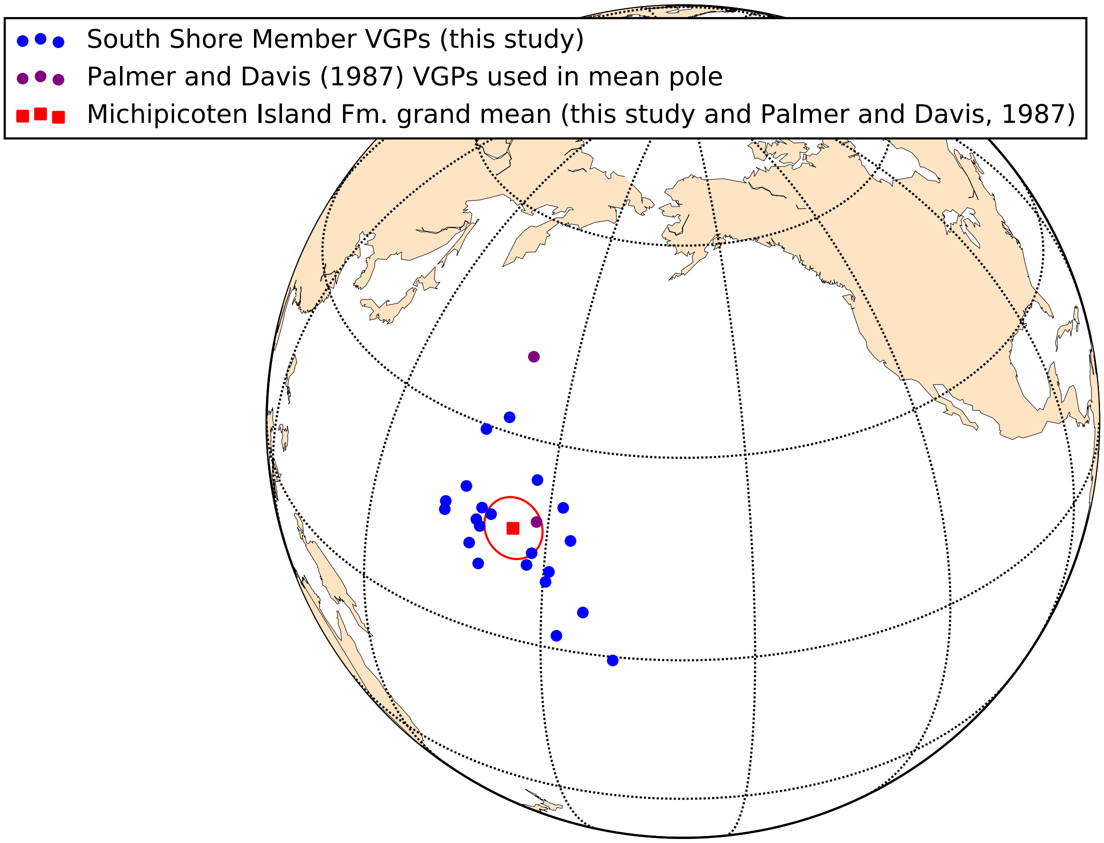
```
In [47]: Michipicoten_Island_Fm.iloc[[3,7], :]
```

```
In [48]: Palmer_data_trimmed = Michipicoten_Island_Fm.iloc[[3,7], :]
         combined_Michi_lon = Palmer_data_trimmed['vgp_lon'].tolist()\
         + SS_final_tc['vgp_lon'].tolist()
         combined_Michi_lat = Palmer_data_trimmed['vgp_lat'].tolist()\
         + SS_final_tc['vgp_lat'].tolist()
         combined_Michi_mean = ipmag.fisher_mean(combined_Michi_lon,combined_Michi_lat)
```

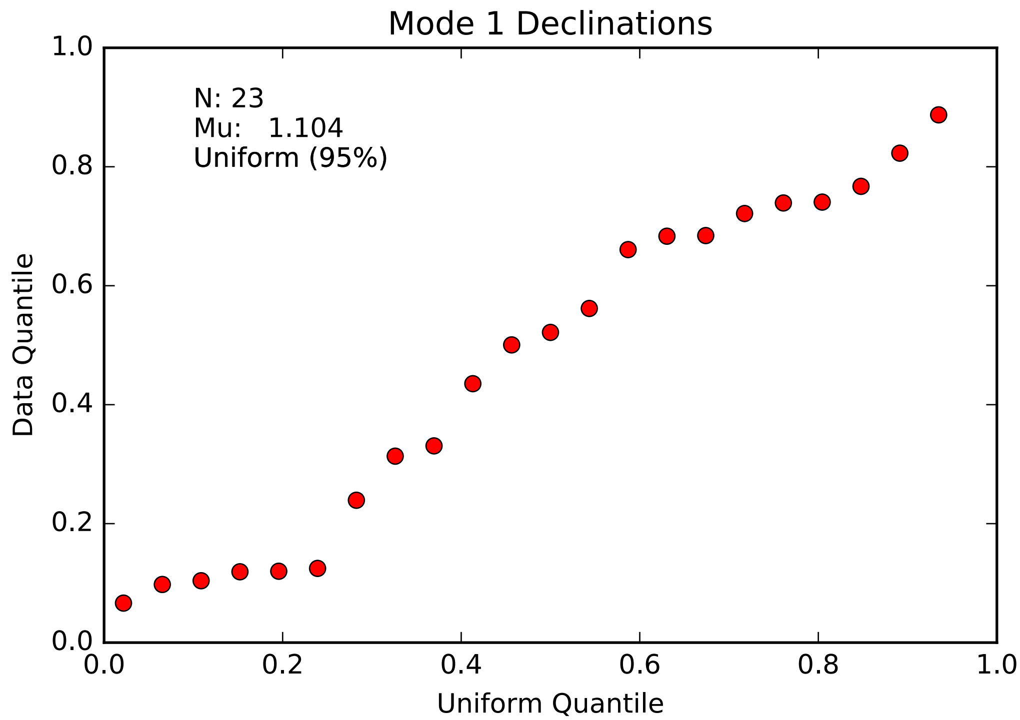
```
In [49]: SS_final_mean = ipmag.fisher_mean(SS_final_tc['vgp_lon'].tolist(),
                                           SS_final_tc['vgp_lat'].tolist())
```

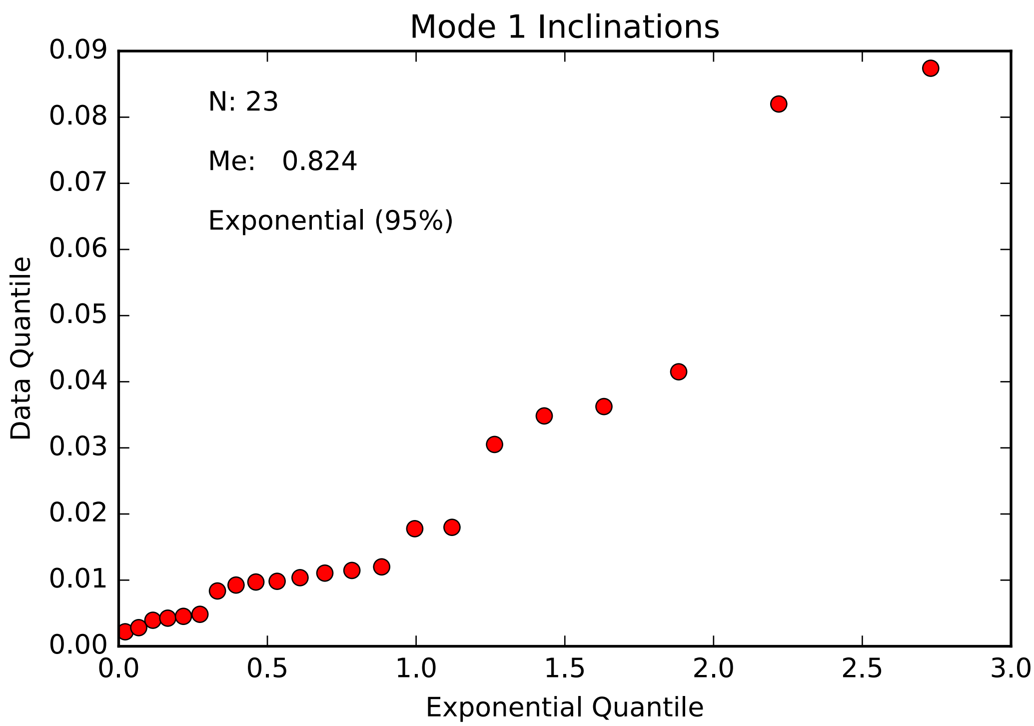
```
m = pole_figure_appearance()

ipmag.plot_vgp(m,SS_final_tc['vgp_lon'].tolist(),
               SS_final_tc['vgp_lat'].tolist(),
               label='South Shore Member VGPs (this study)',color='b')

ipmag.plot_vgp(m,Palmer_data_trimmed['vgp_lon'].tolist(),
               Palmer_data_trimmed['vgp_lat'].tolist(),
               label='Palmer and Davis (1987) VGPs used in mean pole',
               color='purple')

ipmag.plot_pole(m,combined_Michi_mean['dec'],combined_Michi_mean['inc'],
                combined_Michi_mean['alpha95'], label='Michipicoten Island Fm. \
grand mean (this study and Palmer and Davis, 1987)',
                marker='s',markersize=25, color='red')

plt.legend()
plt.savefig('Code_output/Michi_VGPs.pdf')

print "South Shore Member pole: "
ipmag.print_pole_mean(SS_final_mean)

print "\nGrand mean: "
ipmag.print_pole_mean(combined_Michi_mean)

plt.show()
```

```
South Shore Member pole:
Plong: 174.7  Plat: 15.8
Number of directions in mean (n): 21
Angular radius of 95% confidence (A_95): 4.3
Precision parameter (k) estimate: 55.4

Grand mean:
Plong: 174.7  Plat: 17.0
Number of directions in mean (n): 23
Angular radius of 95% confidence (A_95): 4.4
Precision parameter (k) estimate: 48.4
```




```
In [50]: ipmag.fishqq(combined_Michi_lon,combined_Michi_lat)

Out[50]: {'Dec': 174.68056363405637,
          'Inc': 16.974664865055974,
          'Me': 0.8235241344357922,
          'Me_critical': 1.094,
          'Mode': 'Mode 1',
          'Mu': 1.1041506813063025,
          'Mu_critical': 1.207,
          'N': 23,
          'Test_result': 'consistent with Fisherian model'}
```


Mode 1 Declinations
N: 23
Mu: 1.104
Uniform (95%)
Data Quantile
Uniform Quantile
0.0
0.2
0.4
0.6
0.8
1.0


Mode 1 Inclinations
N: 23
Me: 0.824
Exponential (95%)
Data Quantile
Exponential Quantile
0.00
0.01
0.02
0.03
0.04
0.05
0.06
0.07
0.08
0.09
0.0
0.5
1.0
1.5
2.0
2.5
3.0

5 Paleogeography of Laurentia

5.1 Late Rift Poles Compilation

Here we compile selected paleomagnetic poles of the Keweenawan Track and add our newly developed data. This section of the code integrates the functionality of the GPlates software for our paleogeographic reconstruction of Laurentia. The `pygplates` package is a dependecy of the following code blocks and is imported below. If you wish to render this notebook interactively, the `pygplates` package can be downloaded from the GPlates website: http://www.gplates.org/download.html

In [51]:
```
# If this code block gives an error after you have already
# installed PyGPlates, you probably still need to add the package to your path
import pygplates
```

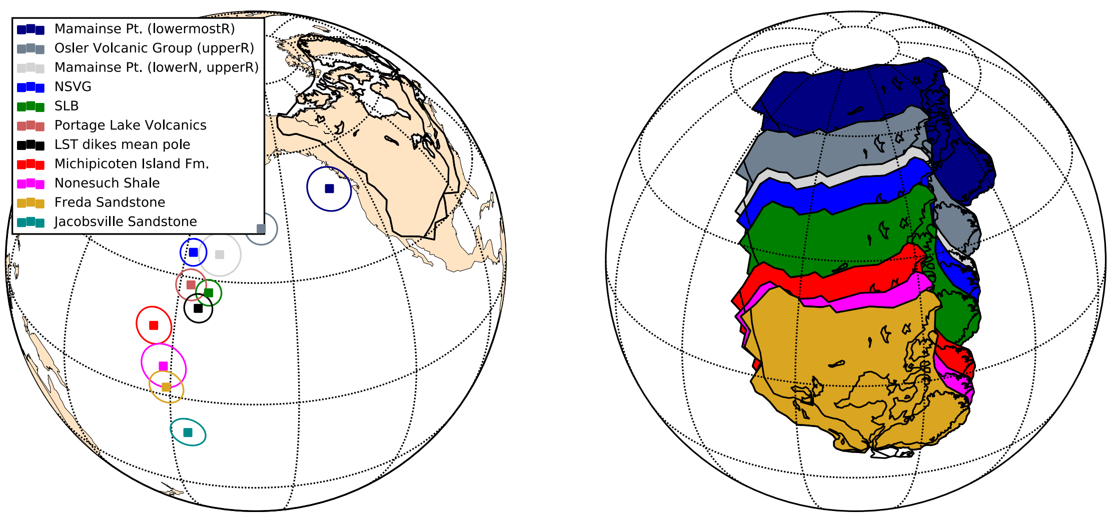
In [52]:
```
# 1080 Ma = Nonesuch pole in recon, 1070 Ma = Freda pole, 1050 = Jacobsville
for reconstruction_time in [1109, 1105, 1100, 1097, 1090, 1083, 1080, 1070, 1050]:
    rotation_model = pygplates.RotationModel(
    '../Data/Reconstruction/Late_Rift_recon.rot')
    plate = pygplates.FeatureCollection(
    '../Data/Reconstruction/1000_Laurentia.gpml')
    plate_outline = pygplates.FeatureCollection(
    '../Data/Reconstruction/1000_Laurentia2.gpml')
    export_filename = './Code_output/Reconstructions/reconstructed_{0}Ma.shp'
            .format(reconstruction_time)
    export_outline_filename =
        './Code_output/Reconstructions/reconstructed_outline_{0}Ma.shp'
        .format(reconstruction_time)
    pygplates.reconstruct(plate, rotation_model, export_filename,
                reconstruction_time)
    pygplates.reconstruct(plate_outline, rotation_model,
                export_outline_filename, reconstruction_time)
```

In [53]:
```
fig = plt.figure(figsize=(12,6))
ax1 = fig.add_subplot(121)
m = Basemap(projection='ortho',lat_0=35,lon_0=200,resolution='c',
            area_thresh=50000)
ax1.set_aspect(30, adjustable='box')
m.drawcoastlines(linewidth=0.25)
m.fillcontinents(color='bisque',lake_color='white')
m.drawmapboundary(fill_color='white')
m.drawmeridians(np.arange(0,360,30))
m.drawparallels(np.arange(-90,90,30))
m.readshapefile('../Data/Reconstruction/Laurentia', 'Laurentia', linewidth=1)
# Swanson-Hysell et al., 2014a -- 1100 Ma
ipmag.plot_pole(m, 227.0, 49.5, 5.3, marker='s', color='navy',
                label='Mamainse Pt. (lowermostR)')
# Swanson-Hysell et al., 2014b -- 1105 Ma
ipmag.plot_pole(m, -158.3, 42.5, 3.7, marker='s', color='slategrey',
                label='Osler Volcanic Group (upperR)')
# Swanson-Hysell et al., 2014a -- 1100 Ma
ipmag.plot_pole(m, -170.3, 36.1, 4.9, marker='s', color='lightgray',
                label='Mamainse Pt. (lowerN, upperR)')
# Tauxe and Kodama (2009) -- 1097 Ma
ipmag.plot_pole(m,NSVG_nswu_mean['dec'],NSVG_nswu_mean['inc'],
                NSVG_nswu_mean['alpha95'], marker='s',color='b',label='NSVG')
```

```
# this study + Tauxe and Kodama (2009) -- 1090 Ma
ipmag.plot_pole(m,combined_SLB_mean['dec'],combined_SLB_mean['inc'],
                combined_SLB_mean['alpha95'], marker='s',color='g',
                label='SLB')
# Kulakov (2014) -- 1093 Ma
ipmag.plot_pole(m, 182.9, 28.3, 3.6, marker='s', color='indianred',
                label='Portage Lake Volcanics')
# Kulakov et al. (2013)
ipmag.plot_pole(m,LST_all_mean['dec'],LST_all_mean['inc'],
                LST_all_mean['alpha95'], label='LST dikes mean pole',
                marker='s')
# this study + Palmer and Davis (1987) -- 1083 Ma
ipmag.plot_pole(m,combined_Michi_mean['dec'],combined_Michi_mean['inc'],
                combined_Michi_mean['alpha95'],
                label='Michipicoten Island Fm.',marker='s', color='red')
# Henry et al. (1977)
ipmag.plot_pole(m, 178.1, 7.6, 5.5, label='Nonesuch Shale',
                marker='s', color='magenta')
# Henry et al. (1977)
ipmag.plot_pole(m, 179, 2.2, 4.2,label='Freda Sandstone',
                marker='s', color='goldenrod')
# Roy et al. (1978)
ipmag.plot_pole(m, 184, -10, 4.2,label='Jacobsville Sandstone',
                marker='s', color='darkcyan')

ax1.legend(fontsize=8, loc='upper left')

ax2 = fig.add_subplot(122)
m = Basemap(projection='ortho', lat_0=30, lon_0=-16)
m.drawmapboundary(fill_color='white')
m.drawmeridians(np.arange(0,360,30))
m.drawparallels(np.arange(-90,90,30))
m.readshapefile('./Code_output/Reconstructions/reconstructed_outline_{0}Ma'
        .format(1109),
                'Mamainse_lowermost', color='navy', linewidth=1)
patches   = []
for info, shape in zip(m.Mamainse_lowermost_info, m.Mamainse_lowermost):
    patches.append( Polygon(np.array(shape), True) )
ax2.add_collection(PatchCollection(patches, facecolor= 'navy', edgecolor='k',
                                   linewidths=1., zorder=2))
m.readshapefile('./Code_output/Reconstructions/reconstructed_{0}Ma'.format(1109),
                'Mamainse_lowermost', color='k', linewidth=1)
m.readshapefile('./Code_output/Reconstructions/reconstructed_outline_{0}Ma'
            .format(1105),
                'Osler_outline', color='slategrey', linewidth=1)
patches   = []
for info, shape in zip(m.Osler_outline_info, m.Osler_outline):
    patches.append( Polygon(np.array(shape), True) )
ax2.add_collection(PatchCollection(patches, facecolor= 'slategrey',
                                   edgecolor='k', linewidths=1., zorder=2))
m.readshapefile('./Code_output/Reconstructions/reconstructed_{0}Ma'.format(1105),
                'Osler', color='k', linewidth=1)
m.readshapefile('./Code_output/Reconstructions/reconstructed_outline_{0}Ma'
            .format(1100),
```

```
                'Mamainse', color='lightgray', linewidth=1)
patches    = []
for info, shape in zip(m.Mamainse_info, m.Mamainse):
    patches.append( Polygon(np.array(shape), True) )
ax2.add_collection(PatchCollection(patches, facecolor= 'lightgray',
                                   edgecolor='k', linewidths=1., zorder=2))
m.readshapefile('./Code_output/Reconstructions/reconstructed_{0}Ma'
            .format(1100),
                'Mamainse', color='k', linewidth=1)
m.readshapefile('./Code_output/Reconstructions/reconstructed_outline_{0}Ma'
            .format(1097),
                'NSVG', color='b', linewidth=1)
patches = []
for info, shape in zip(m.NSVG_info, m.NSVG):
    patches.append( Polygon(np.array(shape), True) )
ax2.add_collection(PatchCollection(patches, facecolor= 'b', edgecolor='k',
                                   linewidths=1., zorder=2))
m.readshapefile('./Code_output/Reconstructions/reconstructed_{0}Ma'.format(1097),
                'NSVG', color='k', linewidth=1)
m.readshapefile('./Code_output/Reconstructions/reconstructed_outline_{0}Ma'
            .format(1090),
                'SLB', color='g', linewidth=1)
patches = []
for info, shape in zip(m.SLB_info, m.SLB):
    patches.append( Polygon(np.array(shape), True) )
ax2.add_collection(PatchCollection(patches, facecolor= 'g', edgecolor='k',
                                   linewidths=1., zorder=2))
m.readshapefile('./Code_output/Reconstructions/reconstructed_{0}Ma'.format(1090),
                'SLB', color='k', linewidth=1)
m.readshapefile('./Code_output/Reconstructions/reconstructed_outline_{0}Ma'
            .format(1083),
                'Michipicoten', color='r', linewidth=1)
patches = []
for info, shape in zip(m.Michipicoten_info, m.Michipicoten):
    patches.append( Polygon(np.array(shape), True) )
ax2.add_collection(PatchCollection(patches, facecolor= 'r', edgecolor='k',
                                   linewidths=1., zorder=2))
m.readshapefile('./Code_output/Reconstructions/reconstructed_{0}Ma'.format(1083),
                'Michipicoten', color='k', linewidth=1)
m.readshapefile('./Code_output/Reconstructions/reconstructed_outline_{0}Ma'
            .format(1080),
                'Nonesuch', color='magenta', linewidth=1)
patches = []
for info, shape in zip(m.Nonesuch_info, m.Nonesuch):
    patches.append( Polygon(np.array(shape), True) )
ax2.add_collection(PatchCollection(patches, facecolor= 'magenta',
                                   edgecolor='k', linewidths=1., zorder=2))
m.readshapefile('./Code_output/Reconstructions/reconstructed_{0}Ma'.format(1080),
                'Nonesuch', color='k', linewidth=1)
m.readshapefile('./Code_output/Reconstructions/reconstructed_outline_{0}Ma'
            .format(1070),
                'Freda', color='goldenrod', linewidth=1)
patches = []
for info, shape in zip(m.Freda_info, m.Freda):
```

```
    patches.append( Polygon(np.array(shape), True) )
ax2.add_collection(PatchCollection(patches, facecolor= 'goldenrod',
                                   edgecolor='k', linewidths=1., zorder=2))
m.readshapefile('./Code_output/Reconstructions/reconstructed_{0}Ma'.format(1070),
                'Freda', color='k', linewidth=1)
#plt.savefig('Code_output/poles.pdf')
plt.show()
```



5.2 Age of Keweenawan sediments

5.2.1 *Nonesuch Shale*

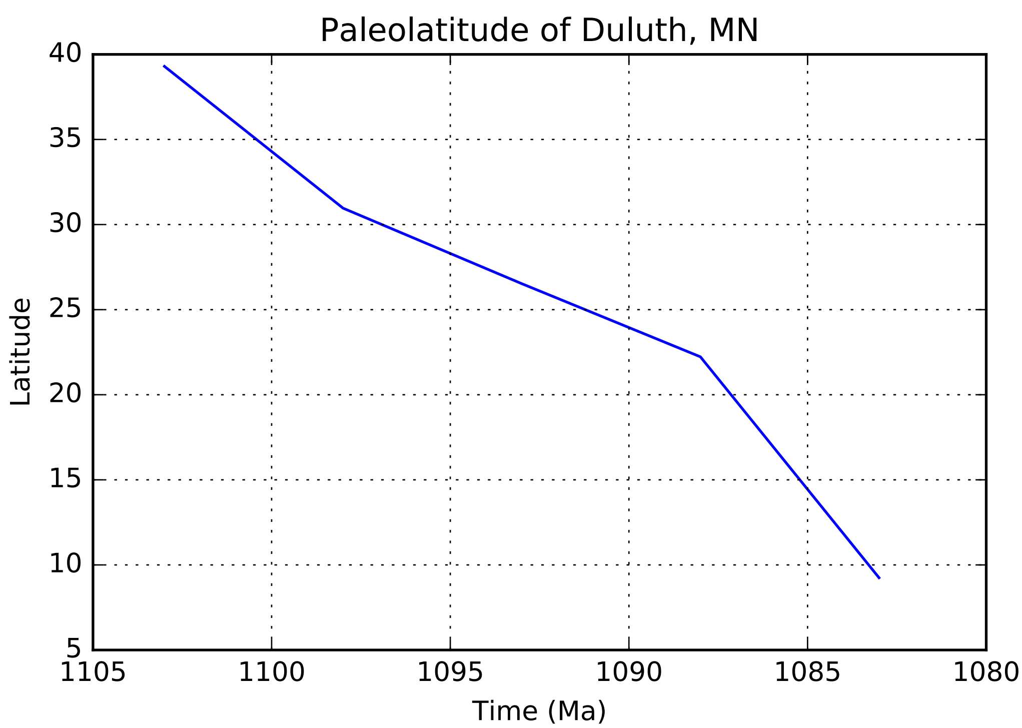
There is much evidence to support the deposition of the Nonesuch shale shortly after or perhaps during late stage volcanism in the Midcontinent Rift (see main text). However, the magnetization of this formation cannot be dated as definitively. Symons et al. (2013) suggested that the magnetization of the Nonesuch shale is secondary, likely a product of oxidation and mineralization in the Nonesuch Formation that postdates its deposition and subsequent burial by the Freda sandstone. They dated this magnetization at 1063 $\pm$ 8 Ma by projecting Laurentia's latitudinal rate of motion to the paleolatitude implied by the Nonesuch paleomagnetic pole. Their analysis assumes a rate of motion consistent with the previously hypothesized slowdown of Laurentia during the late stage of rifting (Davis and Green, 1997). However, our results suggest no significant change in Laurentia's motion over this period. Redoing the analysis of Symons et al. (2013) with our revised rate estimates yields an age of Nonesuch shale magnetization of approximately 1078 Ma (see below).

Create a motion path of Laurentia that captures the paleolatitude of Duluth, MN at 5 myr intervals through active rifting (1108-1083 Ma).

```
In [54]: # Duluth, MN
         SeedPoint = (46.78,-92.1)
         MovingPlate = 1000
         RelativePlate = 0
         times = np.arange(1083,1108,5)

         # Create a motion path feature
         digitisation_time = 0
         seed_points_at_digitisation_time = pygplates.MultiPointOnSphere([SeedPoint])
```

```
motion_path_feature = pygplates.Feature.create_motion_path(
        seed_points_at_digitisation_time,
        times,
        valid_time=(1110, 1000),
        relative_plate=RelativePlate,
        reconstruction_plate_id = MovingPlate)

# Create the shape of the motion path
reconstruction_time = 1083
reconstructed_motion_paths = []
pygplates.reconstruct(motion_path_feature, rotation_model,
                      reconstructed_motion_paths, reconstruction_time,
                      reconstruct_type=pygplates.ReconstructType.motion_path)

# get the reconstructed coordinates into numpy arrays
for reconstructed_motion_path in reconstructed_motion_paths:
    trail = reconstructed_motion_path.get_motion_path().to_lat_lon_array()
```

In [55]:
```
plt.plot(times,np.flipud(trail[:,0]))
plt.title('Paleolatitude of Duluth, MN')
plt.xlabel('Time (Ma)')
plt.ylabel('Latitude')
plt.gca().grid()
plt.gca().invert_xaxis()
plt.show()
```



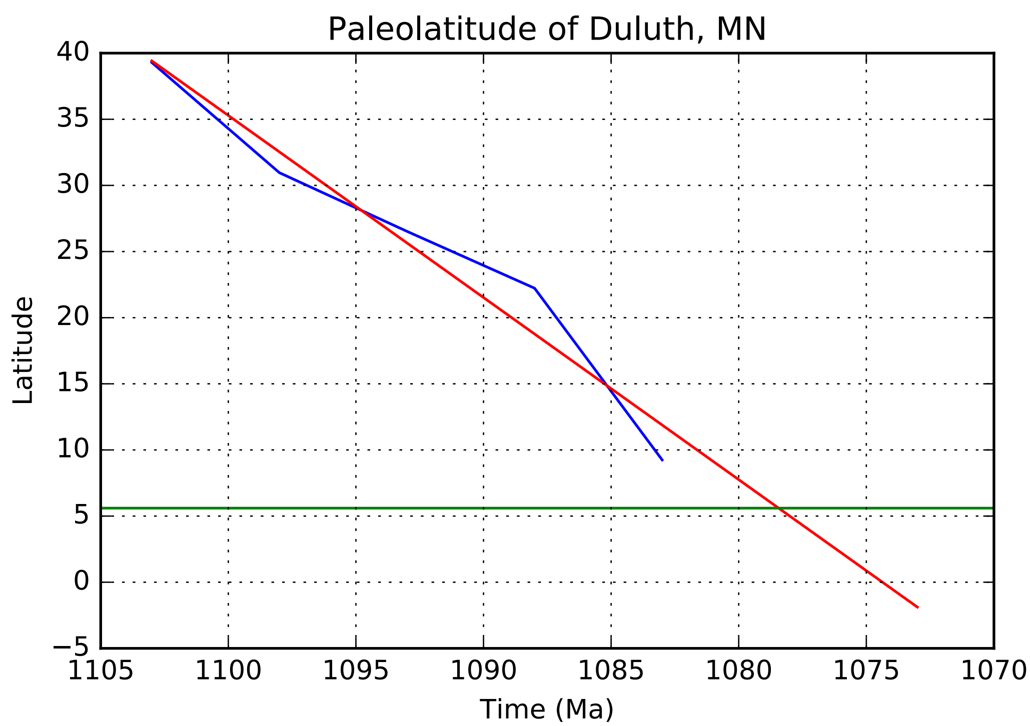
The latitudinal change plotted above appears fairly constant. We therefore make a least squares fit to this data below.

```
In [56]: m, c = np.linalg.lstsq(np.vstack([times, np.ones(len(times))]).T,
                                np.flipud(trail[:,0]))[0]
         print 'Rate of Laurentia\'s latitudinal motion = %0.4f degrees/myr' % (m)
         print 'or %0.4f cm/yr' % (m*11.132)
```

```
Rate of Laurentia's latitudinal motion = 1.3767 degrees/myr
or 15.3254 cm/yr
```

```
In [57]: plt.plot(times,np.flipud(trail[:,0]))
         times_new = np.arange(1073,1108,5)
         plt.plot(times_new, m*times_new + c, 'r')
         # Paleolatitude of Nonesuch is 5.6 degrees
         plt.plot([1070,1105], [5.6,5.6])
         NS_inferred_age = (5.6-c)/m
         plt.title('Paleolatitude of Duluth, MN')
         plt.xlabel('Time (Ma)')
         plt.ylabel('Latitude')
         plt.gca().grid()
         plt.gca().invert_xaxis()
         plt.show()
```

```
Inferred age of Nonesuch Shale if Laurentia
continued moving at the same rate after Midcontinent Rift volcanism ended:
1078.4340 Ma
```



5.2.2 *Jacobsville Sandstone*

New detrital zircon dates from the Jacobsville sandstone, which unconformably overlies the Freda sandstone, reveal a much younger age for this formation than estimated for the Oronto Group sediments ($<$959 ± 19

Ma; Malone et al., 2015). The paleolatitude of Laurentia implied by the Neoproterozoic APWP was then matched with that implied by the Jacobsville paleomagnetic pole, yielding a set of possible ages for Jacobsville deposition: ~780-755, ~700-610 or ~570-555 Ma (Malone et al., 2015). Motivations for redoing this analysis with paleomagnetic poles from both the APWP of Laurentia and the APWP of Baltica (hypothesized to have a longstanding connection with Laurentia's northeastern margin throughout the Neoproterozoic; Pisarevsky et al., 2003; Li et al., 2008; Evans, 2009) are outlined in the main text (*Discussion: Age of Keweenawan sediments*).

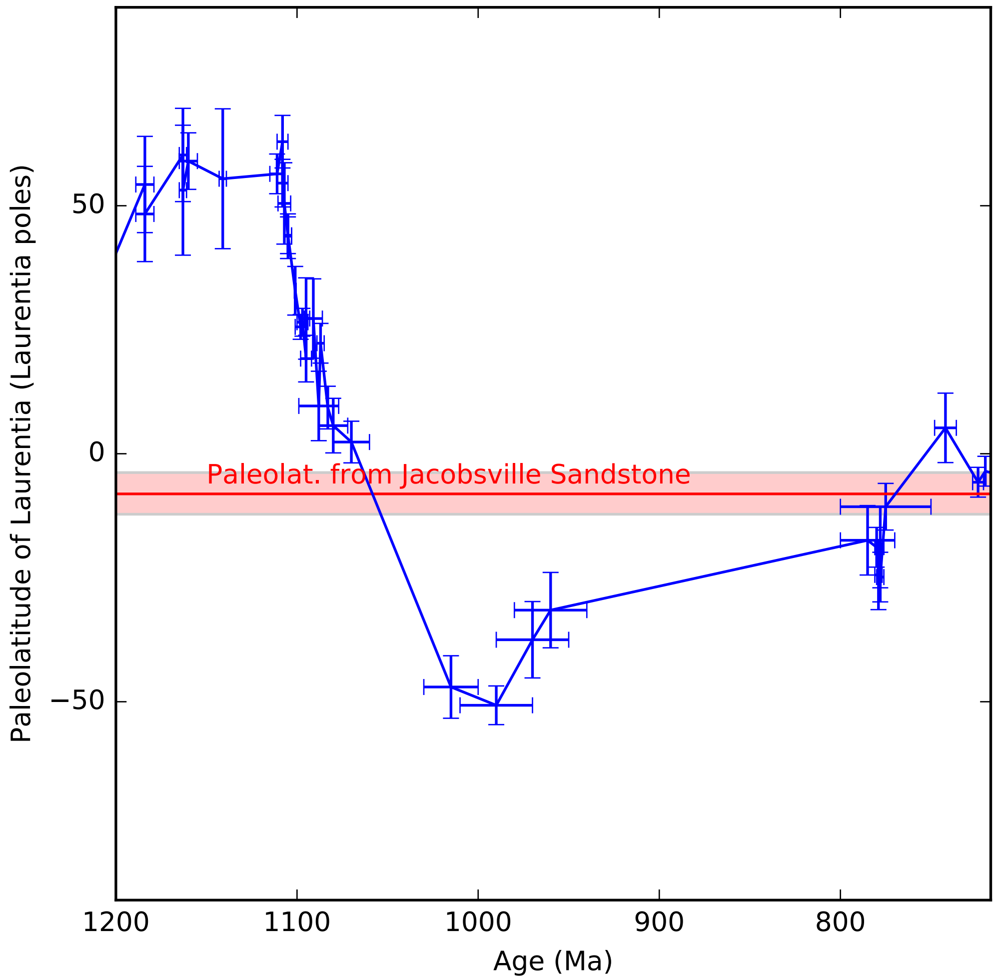
We first upload paleomagnetic poles of Laurentia as compiled by the Nordic Supercontinent Workshop, Haraldvangen, Norway in 2014 (see references in Figure 7 of the manuscript).

```
In [58]: Laurentia_poles = pd.read_csv('../Data/Reconstruction/Laurentia_Poles.csv',
                                       usecols=['Formation', 'Terrane', 'AgeUpper',
                                                'AgeLower', 'AgeNominal', 'A95',
                                                'SLat', 'SLon', 'PLat', 'PLon',
                                                'RefLat', 'RefLon'])
         Laurentia_poles = Laurentia_poles.sort_values(by='AgeNominal',
                                                       ascending=False)
         Laurentia_poles.reset_index(inplace=True, drop=True)
         Laurentia_poles
```

The paleolatitude of Duluth, MN ("RefLat", "RefLon") is then calculated for each of these paleomagnetic poles and plotted below.

```
In [59]: Laurentia_poles['Latitude'] = pd.Series(90 - np.rad2deg(np.arccos\
         (np.sin(np.deg2rad(Laurentia_poles['RefLat']))\
         *np.sin(np.deg2rad(Laurentia_poles['PLat']))\
         +np.cos(np.deg2rad(Laurentia_poles['RefLat']))\
         *np.cos(np.deg2rad(Laurentia_poles['PLat']))\
         *np.cos(np.deg2rad(Laurentia_poles['RefLon']-Laurentia_poles['PLon'])))))

In [60]: # First we must tweak the format of the pole list
         Laurentia_copy = pd.DataFrame(Laurentia_poles, columns=['PLat','PLon'])
         Laurentia_copy = Laurentia_copy.rename_axis({'PLat':'PLat_rot',
                                                      'PLon':'PLon_rot'},
                                                     axis='columns')
         Laurentia_poles['PLat_rot'] = Laurentia_copy.PLat_rot
         Laurentia_poles['PLon_rot'] = Laurentia_copy.PLon_rot
         Laurentia_poles['AgeNominal_neg'] = Laurentia_poles.AgeNominal.apply(np.negative)

         # Plot the paleolatitude of Duluth using Laurentia poles
         fig = plt.figure(figsize=(20,6))

         ax1 = fig.add_subplot(131)
         ax1.add_patch(mpatch.Rectangle((1300, -12.2), -1000, 8.4,
                                        facecolor='r', alpha=0.2))
         ax1.plot([-8.1]*2000, 'r-')
         ax1.errorbar(Laurentia_poles['AgeNominal'].tolist(),
                      Laurentia_poles['Latitude'].tolist(),
                      yerr=Laurentia_poles['A95'].tolist(),
                      xerr=(Laurentia_poles['AgeUpper']-Laurentia_poles['AgeLower'])/2)
         ax1.text(1150, -6,'Paleolat. from Jacobsville Sandstone',
                  color='r', withdash=True)
         # Cut off at Tonian/Cryogenian boundary
         ax1.set_xlim(1200,717)
         ax1.set_ylim(-90,90)
```

```
ax1.set_ylabel('Paleolatitude of Laurentia (Laurentia poles)')
ax1.set_xlabel('Age (Ma)')
plt.show()
```



Next, we upload paleomagnetic poles of Baltica as compiled by the Nordic Supercontinent Workshop, Haraldvangen, Norway in 2014 (see references in Figure 7 of the manuscript).

```
In [61]: Baltica_poles = pd.read_csv('../Data/Reconstruction/Baltica_Poles.csv',
                                     usecols=['Formation', 'Terrane', 'AgeUpper',
                                              'AgeLower', 'AgeNominal', 'A95',
                                              'SLat', 'SLon', 'PLat', 'PLon',
                                              'RefLat', 'RefLon'])
         Baltica_poles = Baltica_poles.sort_values(by='AgeNominal', ascending=False)
         Baltica_poles.reset_index(inplace=True, drop=True)
         Baltica_poles
```

```
Out[61]: Click to view Baltica poles
        (as compiled by the 2014 Nordic Supercontinent Workshop, Haraldvangen, Norway)
```

If you are having trouble opening the local file via this link, or if you have downloaded this document separate from the accompanying data files, the file can be found in the Data Repository under

/Data/Reconstruction/Baltica_Poles.csv and opened manually.

We then rotate these Baltica poles based on the reconstruction of Evans (2009) and do the same calculation for paleolatitude of Duluth, MN.

```
In [62]: Baltica_poles['PLat_rot'], Baltica_poles['PLon_rot'] = pmag.PTrot([81.5,
                                                                               -110.0,
                                                                               -50.0],
                                                         Baltica_poles.PLat.tolist(),
                                                         Baltica_poles.PLon.tolist())
         Baltica_poles['Latitude'] = pd.Series(90 - np.rad2deg(np.arccos\
         (np.sin(np.deg2rad(Baltica_poles['RefLat']))\
         *np.sin(np.deg2rad(Baltica_poles['PLat_rot']))\
         +np.cos(np.deg2rad(Baltica_poles['RefLat']))\
         *np.cos(np.deg2rad(Baltica_poles['PLat_rot']))\
         *np.cos(np.deg2rad(Baltica_poles['RefLon']-Baltica_poles['PLon_rot'])))))
```

We then combine the Laurentia and Baltica poles.

```
In [63]: combined_poles = Baltica_poles.append(Laurentia_poles)
         combined_poles = combined_poles.sort_values(by='AgeNominal', ascending=False)
         combined_poles = combined_poles.loc[combined_poles['AgeNominal']<=1200]
         # Cut off at Tonian/Cryogenian boundary
         combined_poles = combined_poles.loc[combined_poles['AgeNominal']>=717]
         combined_poles.reset_index(inplace=True, drop=True)
         combined_poles['AgeNominal_neg'] = combined_poles.AgeNominal.apply(np.negative)
```

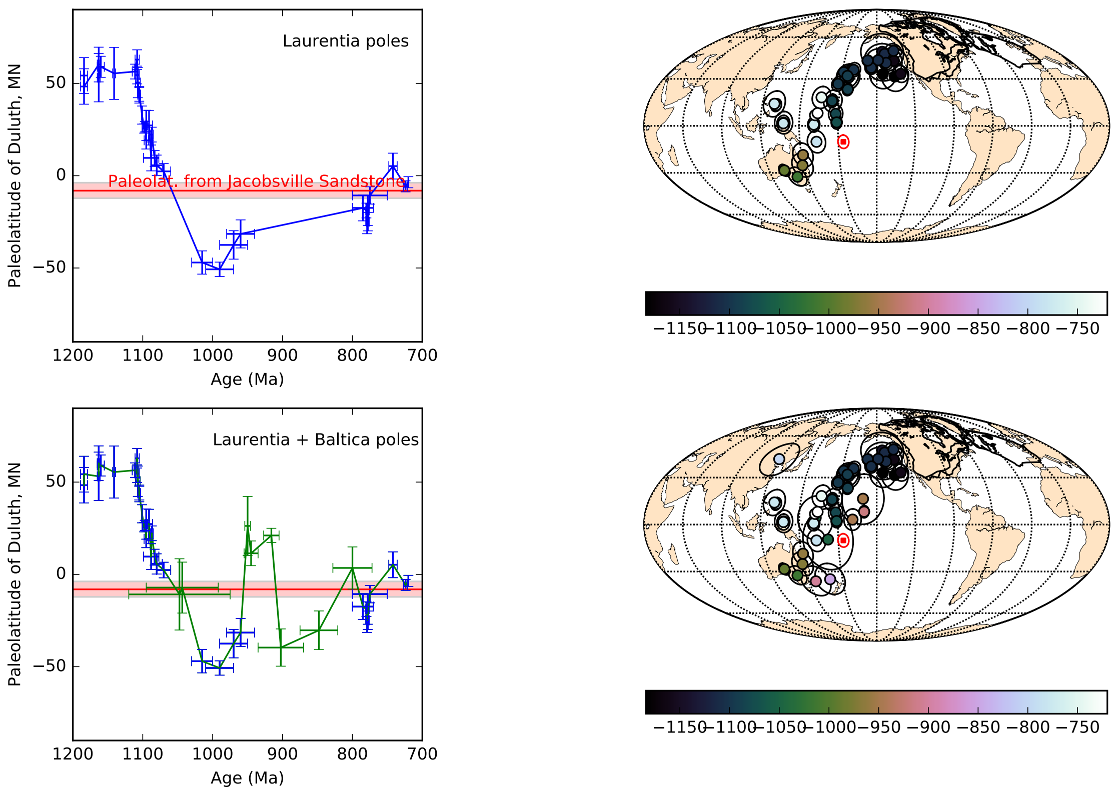
```
Out[63]: Click to view Laurentia poles
        (as compiled by the 2014 Nordic Supercontinent Workshop, Haraldvangen, Norway,
        and with additional data from this study)
```

If you are having trouble opening the local file via this link, or if you have downloaded this document separate from the accompanying data files, the file can be found in the Data Repository under /Data/Reconstruction/Laurentia_Poles.csv and opened manually.

```
In [64]: plt.figure(figsize=(13,8))
        plt.subplot2grid((2,3), (0,1), colspan=2)
        m = Basemap(projection='moll',lat_0=30,lon_0=210,resolution='c',area_thresh=50000)
        m.readshapefile('../Data/Reconstruction/Laurentia_Baltica',
                        'Laurentia_Baltica', drawbounds=True, linewidth=1)
        m.drawcoastlines(linewidth=0.25)
        m.fillcontinents(color='bisque',lake_color='white')
        m.drawmapboundary(fill_color='white')
        m.drawmeridians(np.arange(0,360,30))
        m.drawparallels(np.arange(-90,90,30))
        Laurentia_poles = Laurentia_poles.loc[Laurentia_poles['AgeNominal']<=1200]
        Laurentia_poles = Laurentia_poles.loc[Laurentia_poles['AgeNominal']>=635]
        Laurentia_poles.reset_index(inplace=True, drop=True)
        centerlon, centerlat = m(Laurentia_poles['PLon_rot'].tolist(),
                                 Laurentia_poles['PLat_rot'].tolist())
        for n in range(len(Laurentia_poles)):
            ipmag.plot_pole(m, Laurentia_poles['PLon_rot'][n],
                                 Laurentia_poles['PLat_rot'][n],
                                 Laurentia_poles['A95'][n])
```

```
ipmag.plot_pole(m, 184, -10, 4.2, marker='s', color='r', markersize=6.0)
m.scatter(centerlon, centerlat, c=Laurentia_poles['AgeNominal_neg'].tolist(),
          cmap='cubehelix', s=40, zorder=101)
plt.colorbar(orientation='horizontal', shrink=0.6)

ax1 = plt.subplot2grid((2,3), (0,0), colspan=1)
ax1.add_patch(mpatch.Rectangle((1300, -12.2), -1000, 8.4,
                               facecolor='r', alpha=0.2))
ax1.plot([-8.1]*2000, 'r-')
ax1.errorbar(Laurentia_poles['AgeNominal'].tolist(),
             Laurentia_poles['Latitude'].tolist(),
             yerr=Laurentia_poles['A95'].tolist(),
             xerr=(Laurentia_poles['AgeUpper']-Laurentia_poles['AgeLower'])/2, c='b')
ax1.text(1150, -6,'Paleolat. from Jacobsville Sandstone',
         color='r',withdash=True)
ax1.text(900, 70,'Laurentia poles')
ax1.set_xlim(1200,700)
ax1.set_ylim(-90,90)
ax1.set_ylabel('Paleolatitude of Duluth, MN')
ax1.set_xlabel('Age (Ma)')

m = Basemap(projection='moll',lat_0=30,lon_0=210,resolution='c',
            area_thresh=50000)
plt.subplot2grid((2,3), (1,1), colspan=2)
m.readshapefile('../Data/Reconstruction/Laurentia_Baltica',
                'Laurentia_Baltica', drawbounds=True, linewidth=1)
m.drawcoastlines(linewidth=0.25)
m.fillcontinents(color='bisque',lake_color='white')
m.drawmapboundary(fill_color='white')
m.drawmeridians(np.arange(0,360,30))
m.drawparallels(np.arange(-90,90,30))
centerlon, centerlat = m(combined_poles['PLon_rot'].tolist(),
                         combined_poles['PLat_rot'].tolist())
for n in range(len(combined_poles)):
    ipmag.plot_pole(m, combined_poles['PLon_rot'][n],
                    combined_poles['PLat_rot'][n],
                    combined_poles['A95'][n])
ipmag.plot_pole(m, 184, -10, 4.2, marker='s', color='r', markersize=6.0)
m.scatter(centerlon, centerlat, c=combined_poles['AgeNominal_neg'].tolist(),
          cmap='cubehelix', s=40, zorder=101)
plt.colorbar(orientation='horizontal', shrink=0.6)

ax2 = plt.subplot2grid((2,3), (1,0), colspan=1)
ax2.add_patch(mpatch.Rectangle((1300, -12.2), -1000, 8.4,
                               facecolor='r', alpha=0.2))
ax2.plot([-8.1]*2000, 'r-')
ax2.errorbar(combined_poles['AgeNominal'].tolist(),
             combined_poles['Latitude'].tolist(),
             yerr=combined_poles['A95'].tolist(),
             xerr=(combined_poles['AgeUpper']-
             combined_poles['AgeLower'])/2, c='g')
for n in range(len(Laurentia_poles)):
    ax2.errorbar(Laurentia_poles['AgeNominal'][n],
                 Laurentia_poles['Latitude'][n],
```

```
                                  yerr=Laurentia_poles['A95'][n],
                                  xerr=(Laurentia_poles['AgeUpper'][n]-
                                  Laurentia_poles['AgeLower'][n])/2, c='b')

        ax2.text(1000, 70,'Laurentia + Baltica poles')
        ax2.set_xlim(1200,700)
        ax2.set_ylim(-90,90)
        ax2.set_ylabel('Paleolatitude of Duluth, MN')
        ax2.set_xlabel('Age (Ma)')


        plt.savefig('Code_output/Laur_Bal_poles.svg')
        plt.show()
```

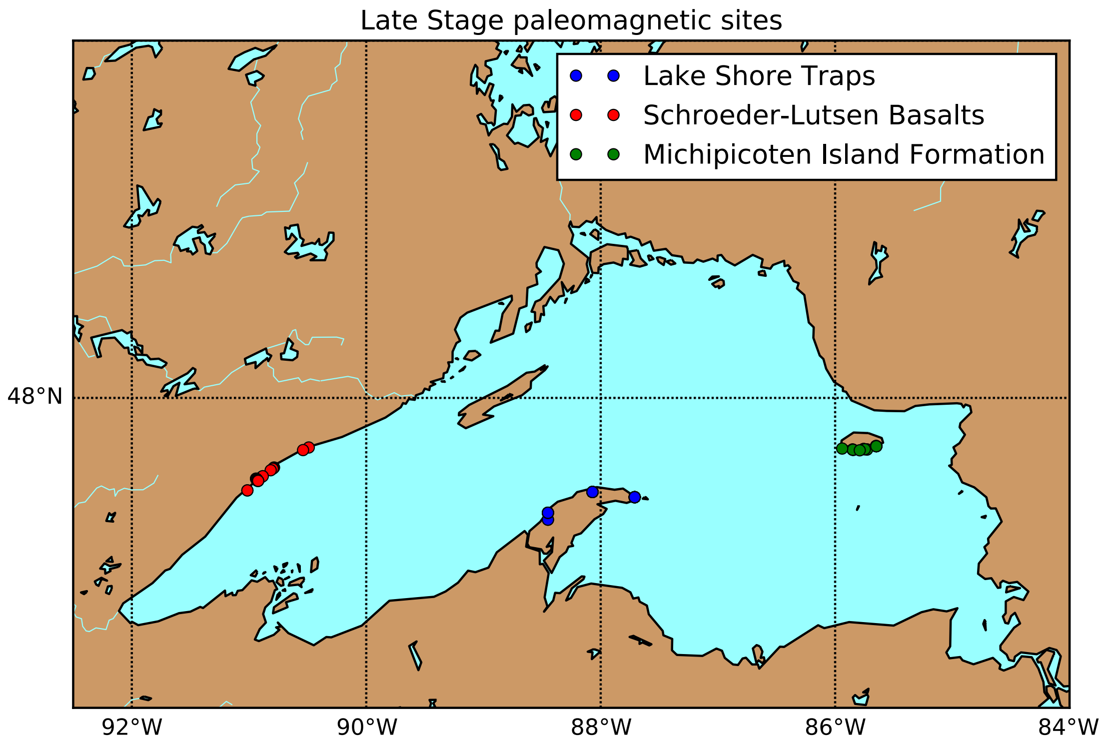


6 Map of late stage volcanics paleomagnetic sites

The following is an overview map of all paleomagnetic sites used in the preceding analysis.

```
In [65]: fig = plt.figure(figsize=(8,8))
        m = Basemap(projection='merc',llcrnrlat=46.2,urcrnrlat=50,llcrnrlon=-92.5,
                    urcrnrlon=-84,resolution='i',area_thresh = 0.1) #lat_ts=-25
        m.drawrivers(color='#99ffff')
        m.drawcoastlines()
        #m.drawcountries(linewidth=1.5)
        m.drawmapboundary(fill_color='#99ffff')
```

```
m.fillcontinents(color='#cc9966',lake_color='#99ffff')
parallels = np.arange(-90,90,2.)
m.drawparallels(parallels,labels=[1,0,0,0],fontsize=10)
meridians = np.arange(0.,360.,2.)
m.drawmeridians(meridians,labels=[0,0,0,1],fontsize=10)
plt.title('Late Stage paleomagnetic sites')

LST_site_lon=[]
LST_site_lat=[]
for n in range(0,len(Kulakov2013a_LST_Data)):
    LST_site_lon.append(Kulakov2013a_LST_Data['site_lon'][n])
    LST_site_lat.append(Kulakov2013a_LST_Data['site_lat'][n])
for n in range(0,len(Diehl1994a_LST_Data_all)):
    LST_site_lon.append(Diehl1994a_LST_Data_all['site_lon'][n])
    LST_site_lat.append(Diehl1994a_LST_Data_all['site_lat'][n])

LST_x,LST_y = m(LST_site_lon,LST_site_lat)
m.plot(LST_x, LST_y, 'bo', markersize=5,
       label='Lake Shore Traps')

SLB_x,SLB_y = m(NSVG_nsl['site_lon'].tolist()+SLB_Data['average_lon'].tolist(),
                NSVG_nsl['site_lat'].tolist()+SLB_Data['average_lat'].tolist())
m.plot(SLB_x, SLB_y, 'ro', markersize=5,
       label='Schroeder-Lutsen Basalts')

MI_x,MI_y = m(SS_final_tc['average_lon'].tolist()+Palmer_data_trimmed['site_lon'].tolist(),
              SS_final_tc['average_lat'].tolist()+Palmer_data_trimmed['site_lat'].tolist())
m.plot(MI_x, MI_y, 'go', markersize=5,
       label='Michipicoten Island Formation')
plt.legend()
plt.show()
```


Late Stage paleomagnetic sites
Lake Shore Traps
Schroeder-Lutsen Basalts
Michipicoten Island Formation
48°N
92°W
90°W
88°W
86°W
84°W