

Open and reproducible paleomagnetic data analysis using IPython

Nicholas L. Swanson-Hysell

Earth and Planetary Science Department // University of California, Berkeley

Calculate and graphically display the confidence interval on the minimum rate estimate implied by the Monte Carlo sampled pairs of sampled ages and paleolatitudes from the two poles

The pairs of sampled ages and poles can be used to calculate rates (with the total number of rates being set by samplesize in the input box above). The rate calculated above of 24.0 cm/year can now be given confidence bounds by taking 2.5 percentile and 97.5 percentile of the Monte Carlo simulated rates.

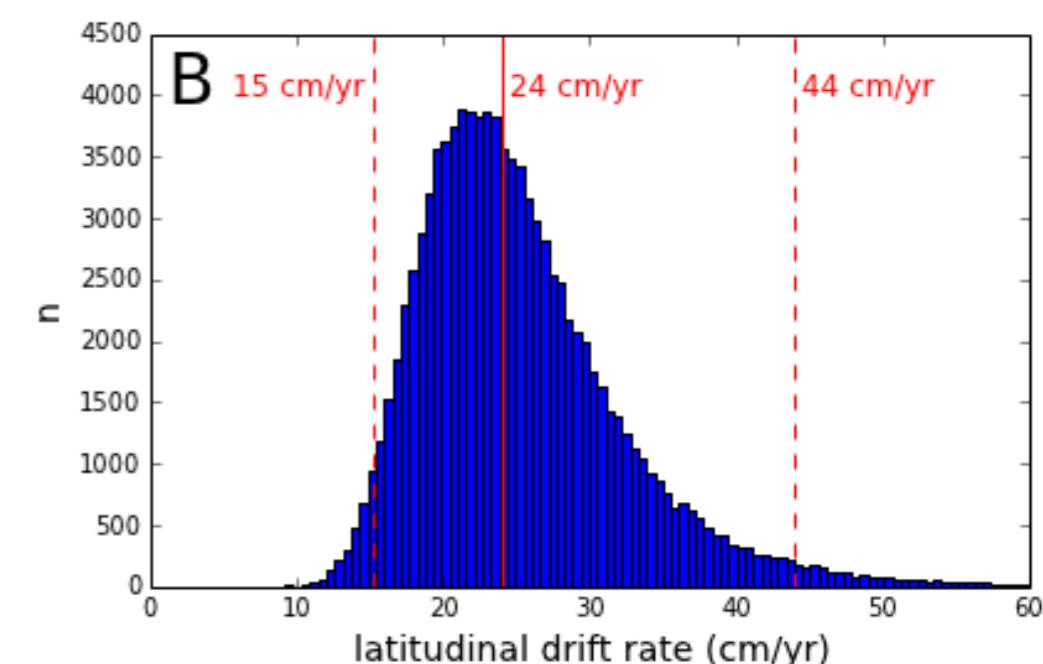
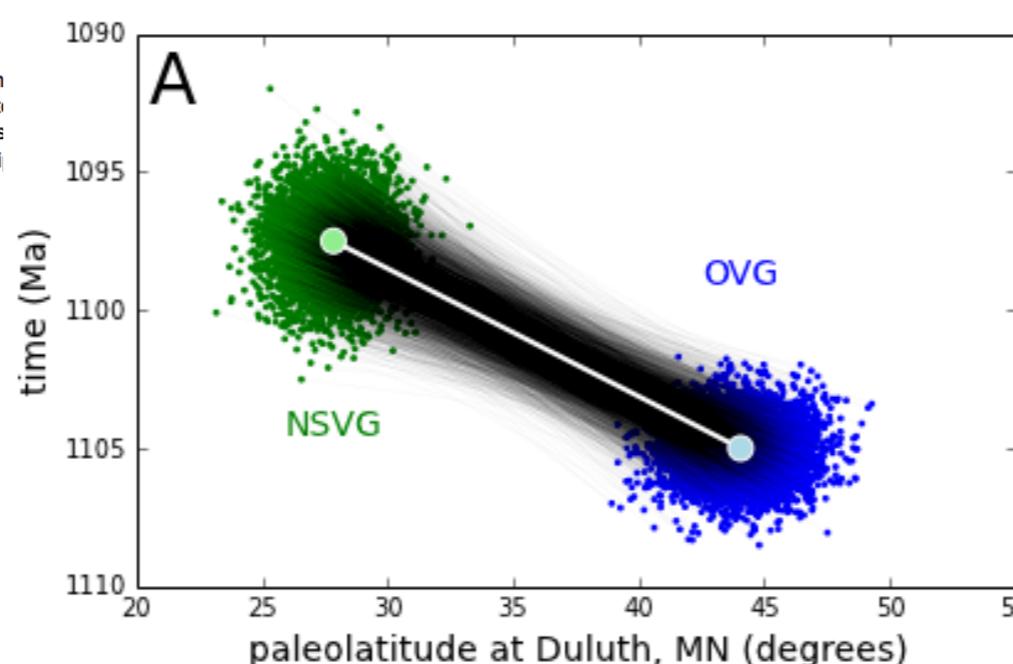
```
In [32]: #calculating the change in paleolatitude between the Monte Carlo pairs
pole1_pole2_Delta_degrees=[]
pole1_pole2_kilometers=[]
pole1_pole2_Delta_myrs=[]
pole1_pole2_degrees_per_myrs=[]
pole1_pole2_cm_per_yrs=[]

for n in range(samplesize):
    Delta_degrees=pole1_MCPaleolat[n]-pole2_MCPaleolat[n]
    Delta_Myr=pole1_MCages[n]-pole2_MCages[n]
    pole1_pole2_Delta_degrees.append(Delta_degrees)
    degrees_per_myrs=Delta_degrees/Delta_Myr
    cm_per_yr=(Delta_degrees*111)*1000000/(Delta_Myr*1000000)
    pole1_pole2_degrees_per_myrs.append(degrees_per_myrs)
    pole1_pole2_cm_per_yrs.append(cm_per_yr)
```

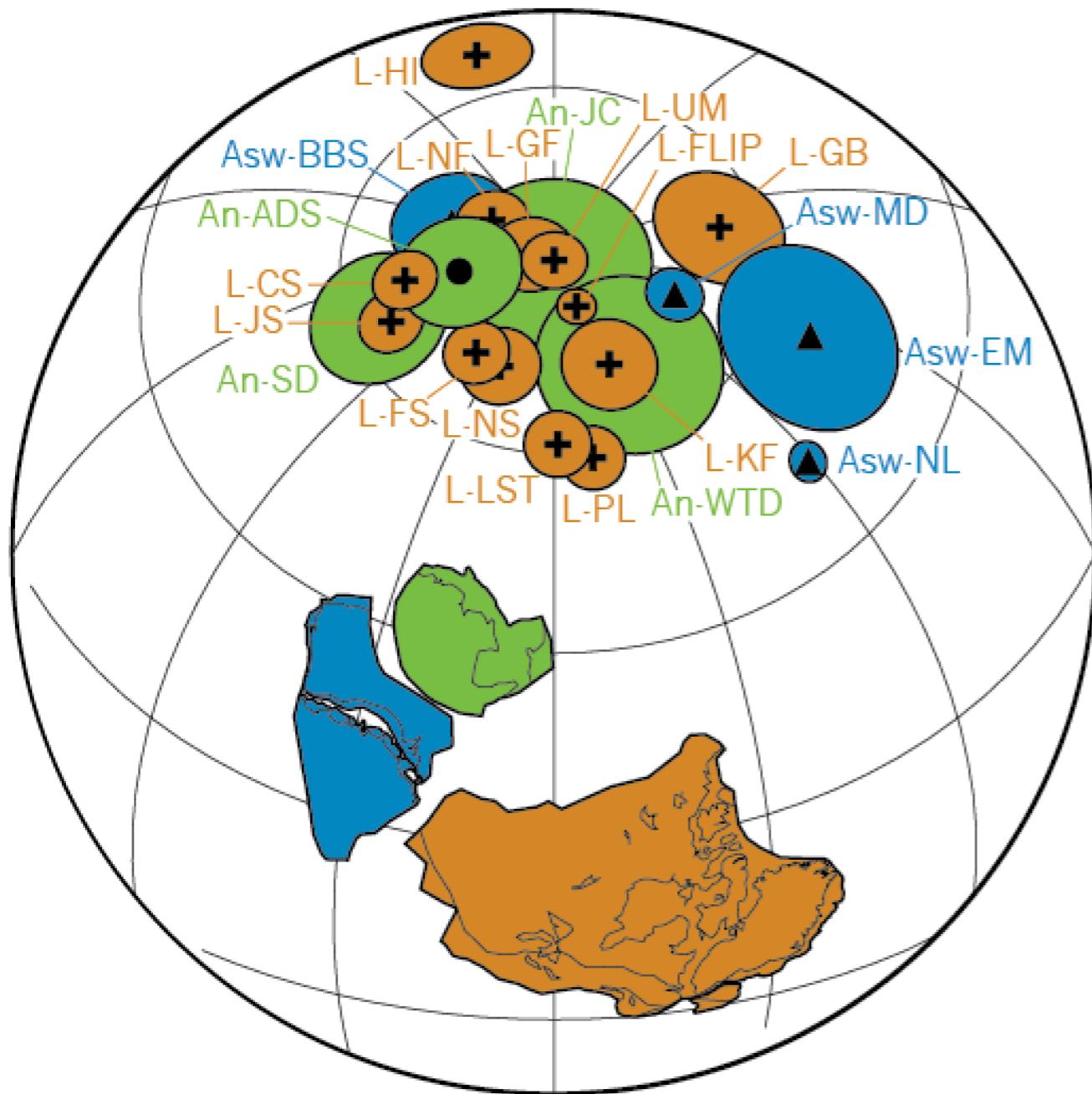
```
In [33]: twopointfive_percentile=stats.scoreatpercentile(pole1_pole2_cm_per_yr,2.5)
fifty_percentile=stats.scoreatpercentile(pole1_pole2_cm_per_yr,50)
ninetysevenpointfive_percentile=stats.scoreatpercentile(pole1_pole2_cm_per_yr,97.5)
print "2.5th percentile is: " + str(twopointfive_percentile)
print "50th percentile is: " + str(fifty_percentile)
print "97.5th percentile is: " + str(ninetysevenpointfive_percentile)
```

2.5th percentile is: 15.1860289306
50th percentile is: 23.9833375899
97.5th percentile is: 44.0185724854

We can see here that the 50th percentile from the percentile give a 95% confidence range of 15.2 to 44.0 cm/year. This is the samplesize when code was executed) pole pairs being marked. This plot is Figure 4 of the manuscript.

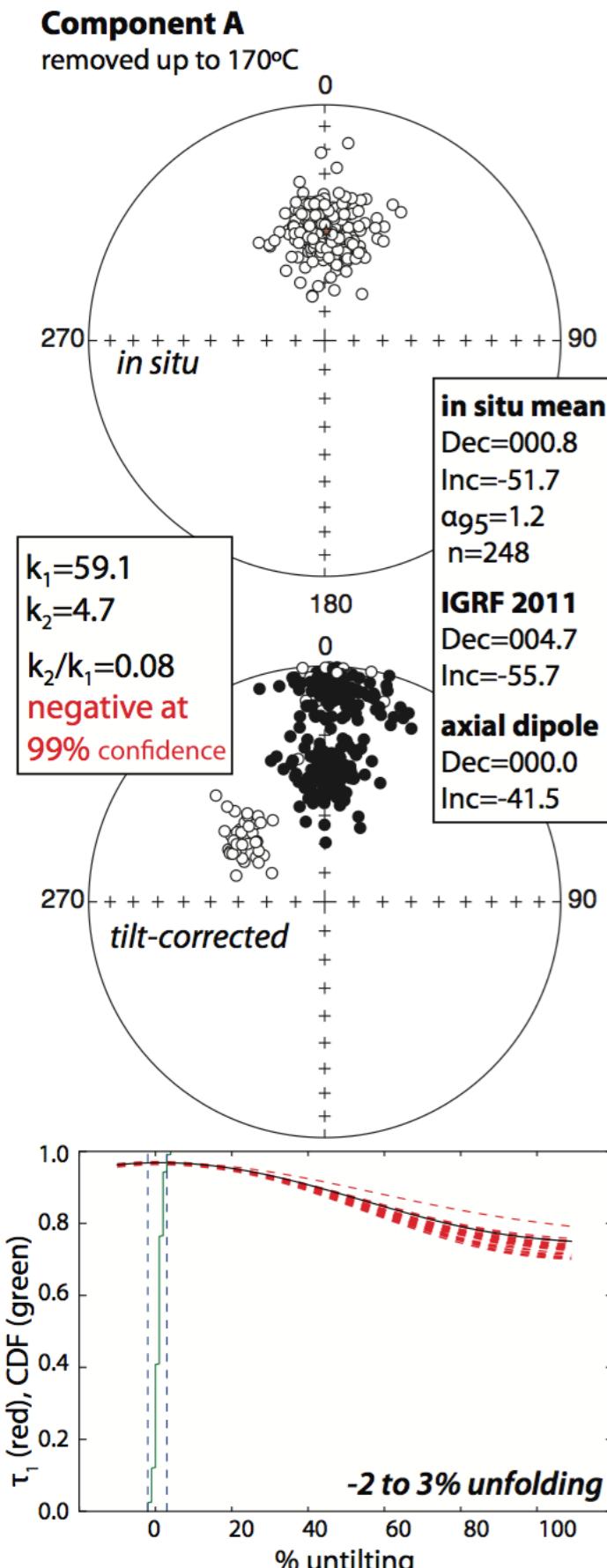


We develop magnetic data to understand the Earth. Our resulting data-driven insights need to be told with:



- the data itself (which need to be robustly archived MagIC!)
- data analysis (statistical summaries, comparisons, tests, simulations)
- models
- plots, images and other visualizations
- descriptive text

An example workflow



A complex disconnected set of tools to make this figure

Paleomag least-square fits

Sheepshaver OS9 emulator

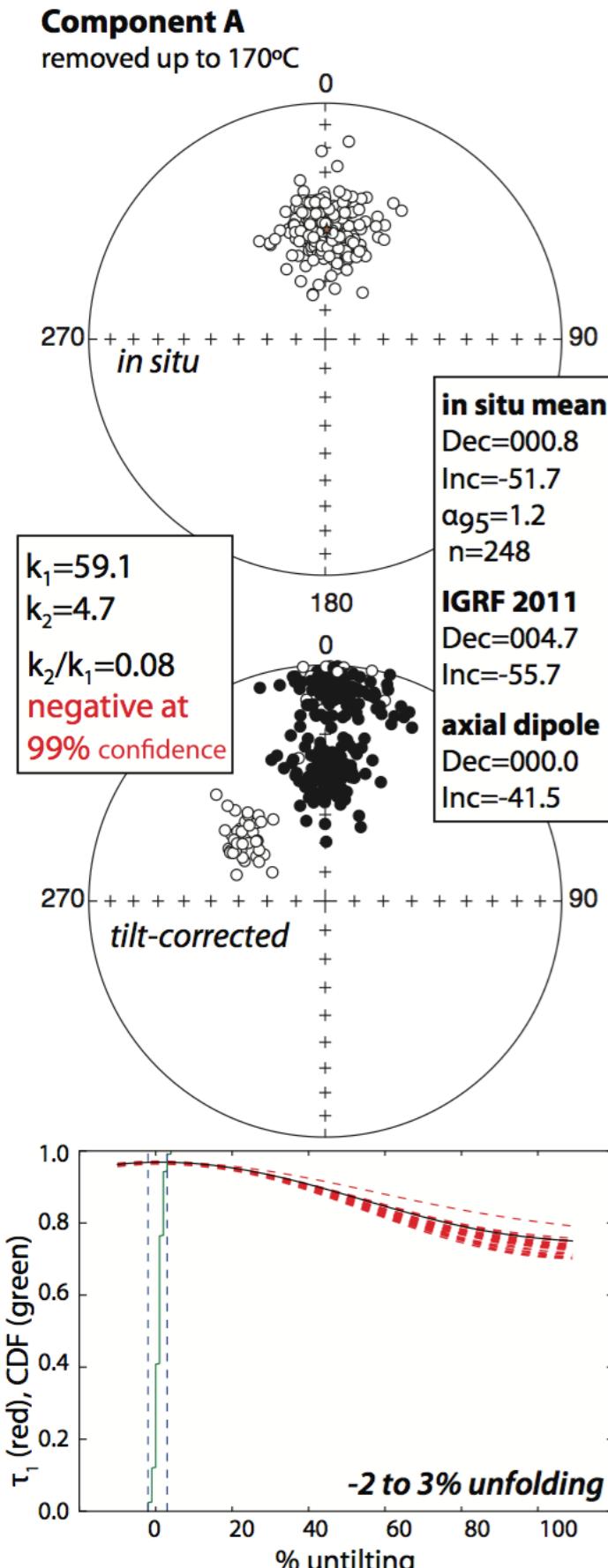
Paleomac equal area plots and Fisher statistics

PmagPy statistical tests

Adobe Illustrator figure compilation

LaTEX document compilation

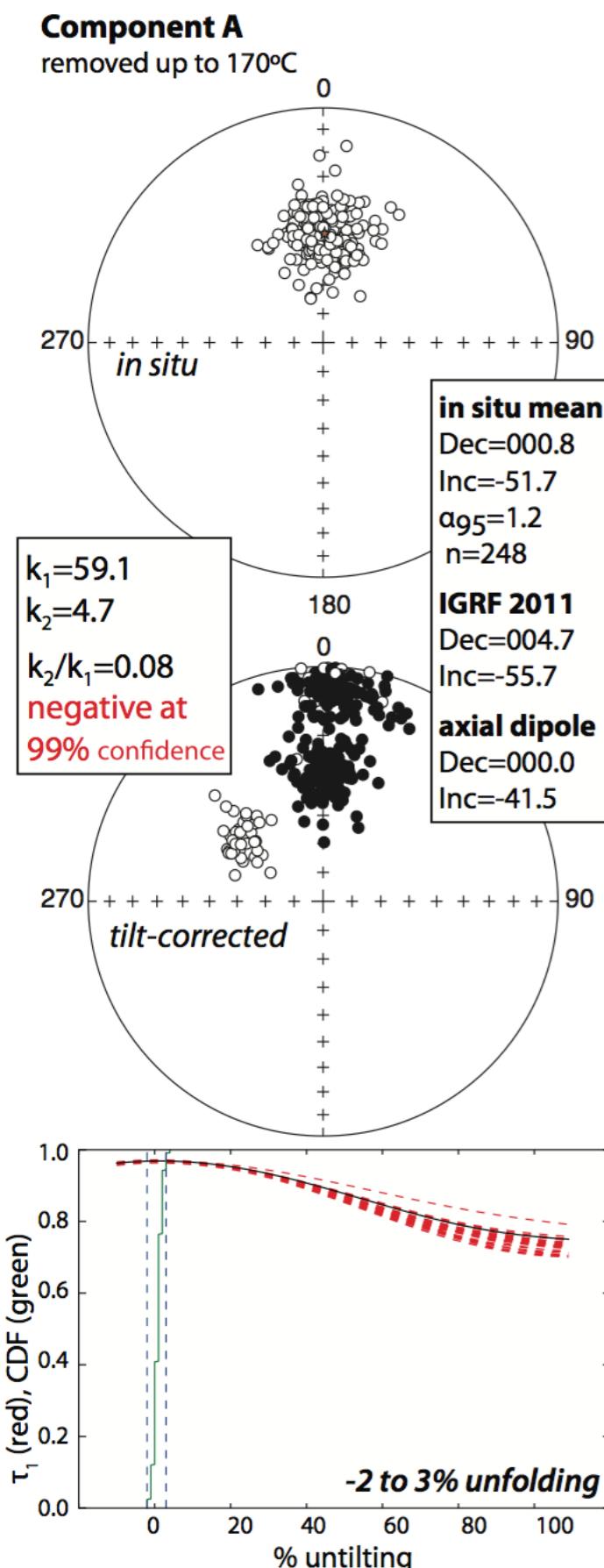
An example workflow



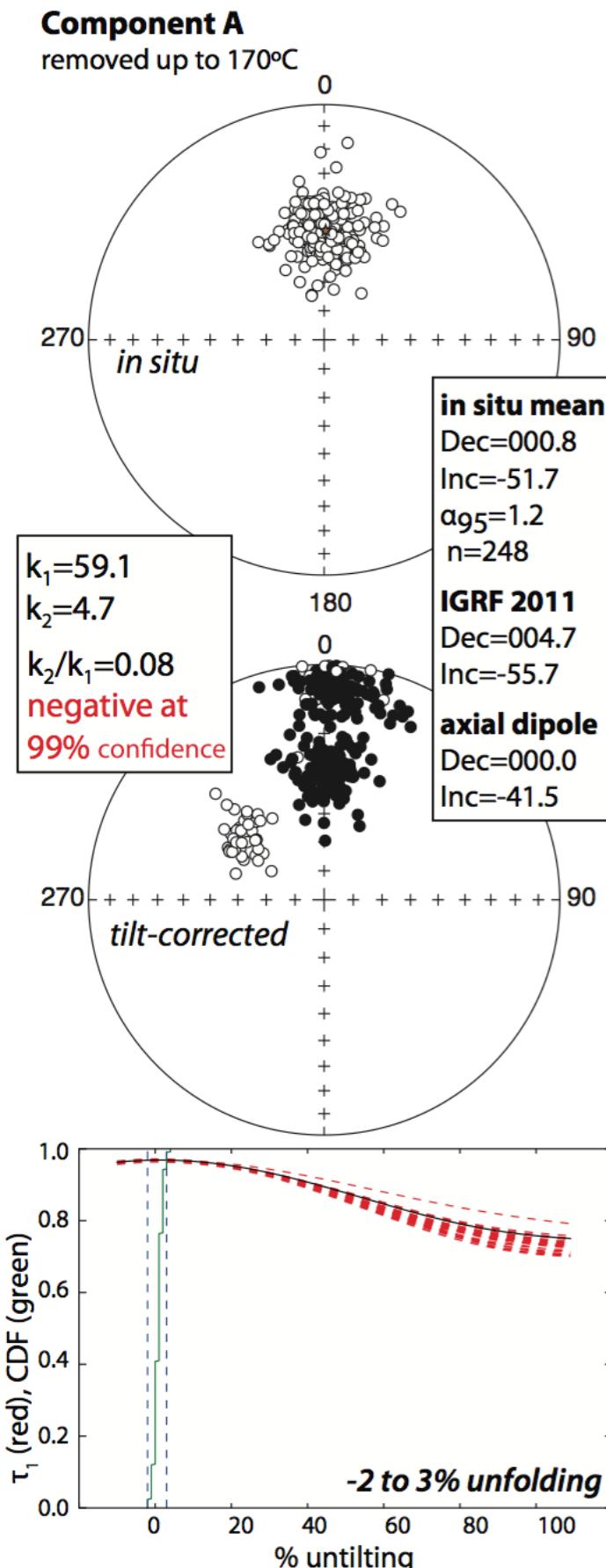
A complex disconnected set of tools to make this figure

- difficult to fully document work flow
- laborious to reproduce
- hinders communication (particularly mid-stream)

**What if all of these data,
statistical results, plots
and underlying code could
be presented in a single
easily-shared document?**



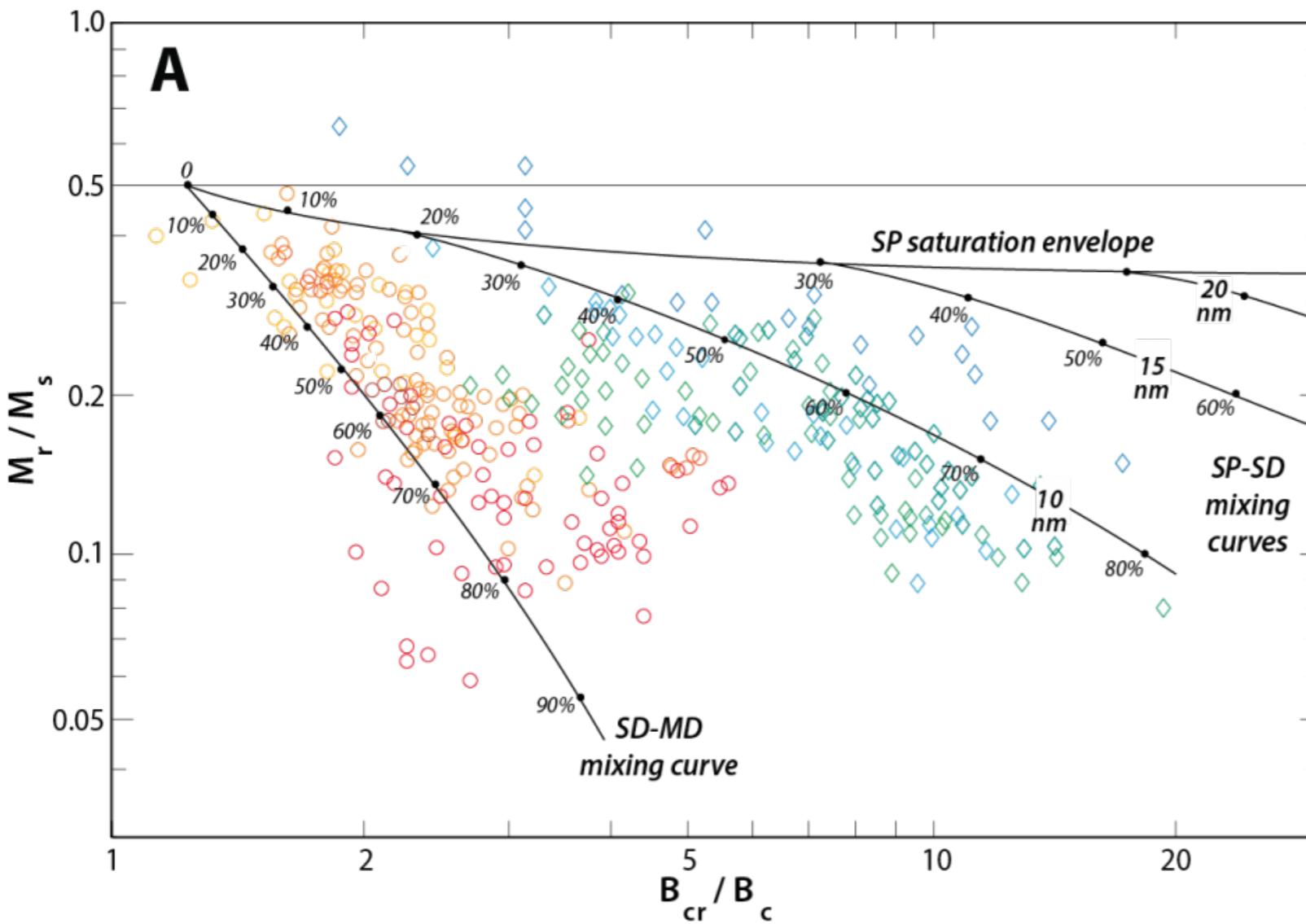
**What if all of these data,
statistical results, plots
and underlying code could
be presented in a single
easily-shared document?**



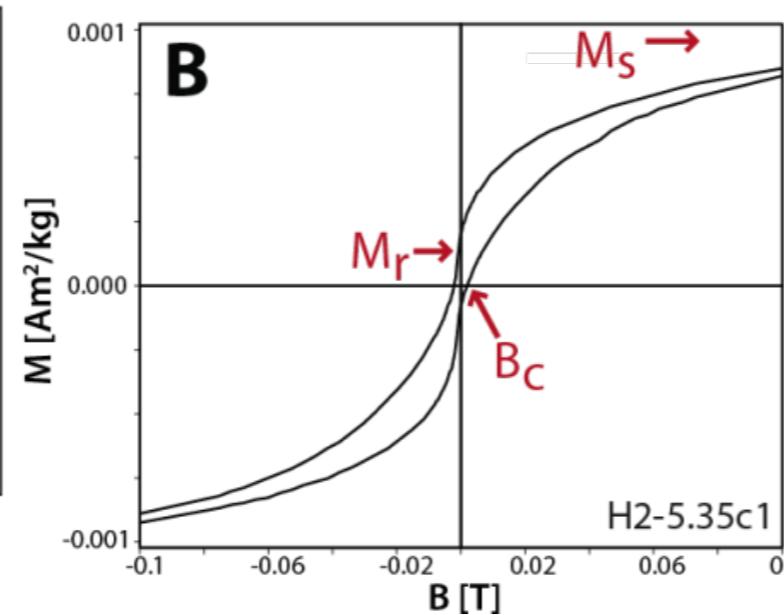
the IPython notebook

**OPEN and
REPRODUCIBLE**

Sharing compilations and plots

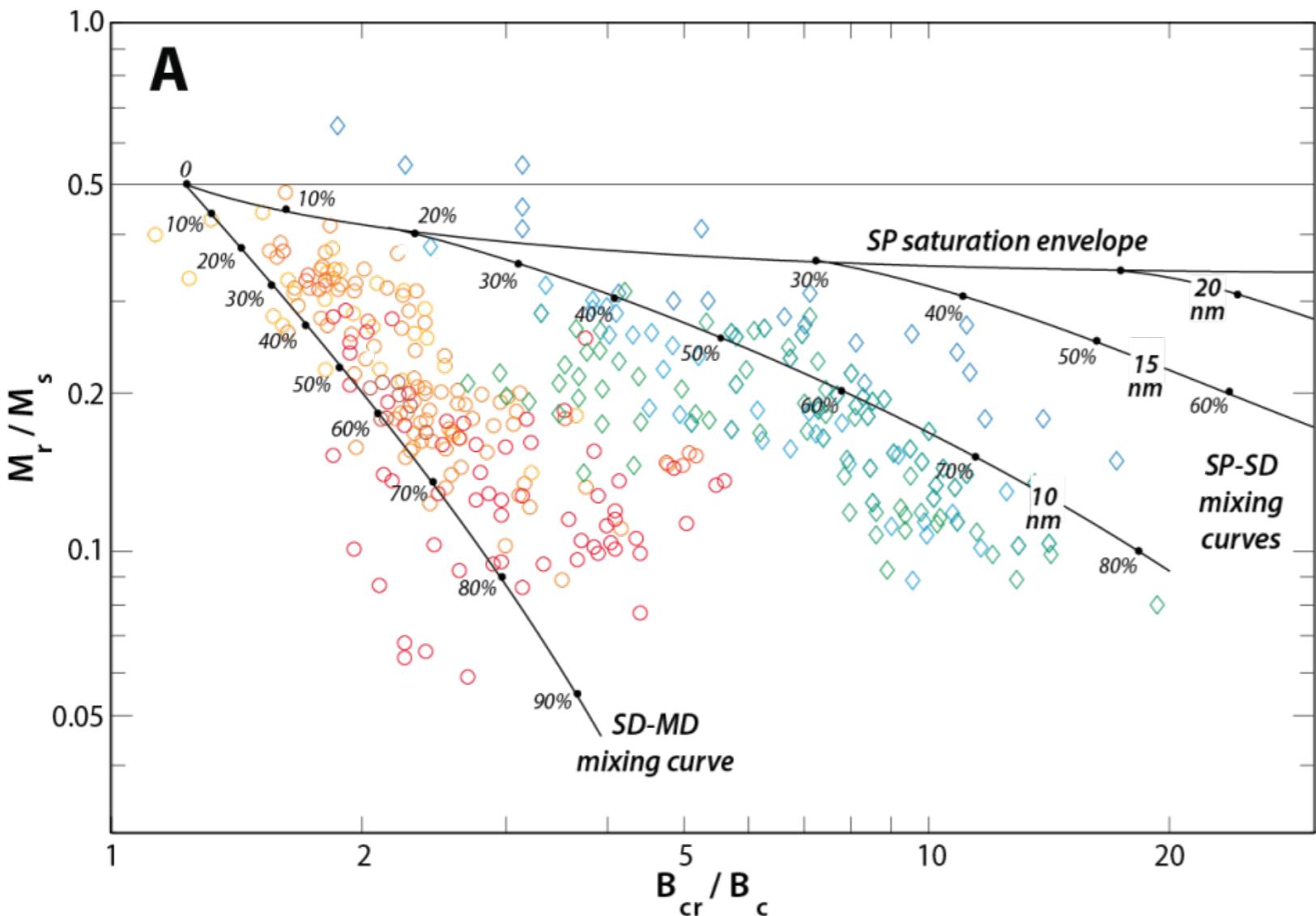


carbonates with primary magnetization	
○	Laytonville Limestone (Tarduno and Myers, 1994)
○	Maiolica Limestone (Channell and McCabe, 1994)
○	Paleocene Chalk (Abrajevitch and Kodama, 2009)
○	Paris Basin Limestones (Belkaaloul and Aissaoui, 1997)
○	Tahiti Pleistocene Reef (Menabreaz et al., 2010)
remagnetized carbonates	
△	NE USA Paleozoic Carbonates (Jackson, 1990)
△	Craven Basin Limestone (McCabe and Channell, 1994)
△	Onondaga Limestone (McCabe and Channell, 1994)
△	Great Basin/Alaska carbonates (McCabe and Channell, 1994)
△	Ordovician/Devonian WV limestone (Elmore et al., 2006)
△	Leadville carbonates (Xu et al., 1998)



Swanson-Hysell et al., 2012
Jackson and Swanson-Hysell, 2012

Sharing compilations and plots

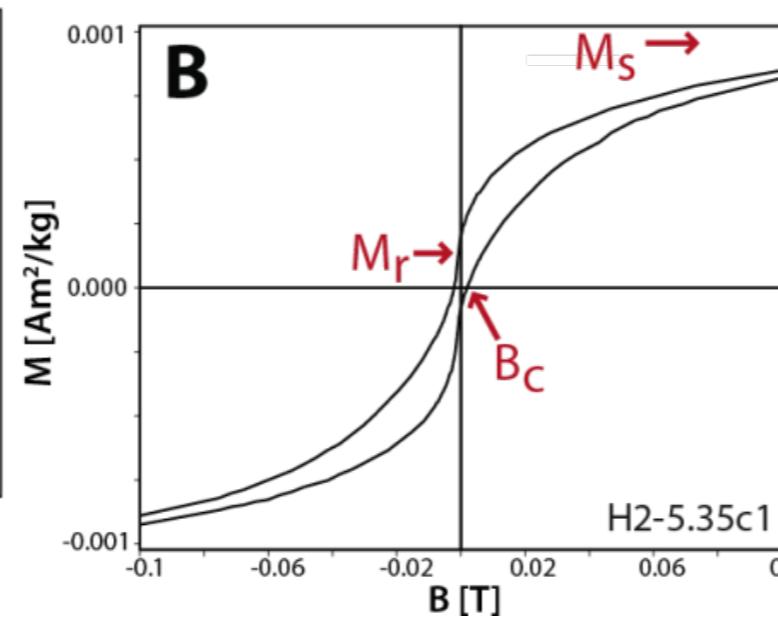


carbonates with primary magnetization

- Laytonville Limestone (Tarduno and Myers, 1994)
- Maiolica Limestone (Channell and McCabe, 1994)
- Paleocene Chalk (Abrajevitch and Kodama, 2009)
- Paris Basin Limestones (Belkaaloul and Aissaoui, 1997)
- Tahiti Pleistocene Reef (Menabreaz et al., 2010)

remagnetized carbonates

- △ NE USA Paleozoic Carbonates (Jackson, 1990)
- △ Craven Basin Limestone (McCabe and Channell, 1994)
- △ Onondaga Limestone (McCabe and Channell, 1994)
- △ Great Basin/Alaska carbonates (McCabe and Channell, 1994)
- △ Ordovician/Devonian WV limestone (Elmore et al., 2006)
- △ Leadville carbonates (Xu et al., 1998)



Andrew Roberts <andrew.roberts@anu.edu.au>
to irm, me ▾

4/22/13



"I wonder if you would be happy to send me the data that you presented in Figure 1a of Jackson& Swanson-Hysell (2012)? I would like to re-plot this figure to avoid copyright issues but would do so with full citation.

Best wishes, Andrew"

Swanson-Hysell et al., 2012
Jackson and Swanson-Hysell, 2012



Data sets from carbonate formations with interpreted primary magnetization

Tarduno1994b Laytonville Limestone (primary magnetization)



[http://www.swanson-hysell.org/
publications/carbonate-
hysteresis-compilation/](http://www.swanson-hysell.org/publications/carbonate-hysteresis-compilation/)

Tarduno, J. A., and Myers, M., 1994, A primary magnetization fingerprint from the Cretaceous Laytonville Limestone: Further evidence for rapid oceanic plate velocities: Journal of Geophysical Research, v. 99, n. B11, p. 21,691–21,703, <http://dx.doi.org/10.1029/94JB01939>

These data are from the Laytonville Limestone—a Cretaceous pelagic Limestone from northern California in an accreted terrain. Some samples contain hematite (Red Laytonville Limestone) and are not included in the compilation. White limestones with negligible hematite are compared in the paper to reference pelagic limestone data and are indistinguishable. A primary magnetization for the Laytonville Limestone is supported by the fact that the trend in inclination matches the expected sense of motion predicated by the North American APW path. [Download .csv file of data for the White Laytonville Limestone](#)

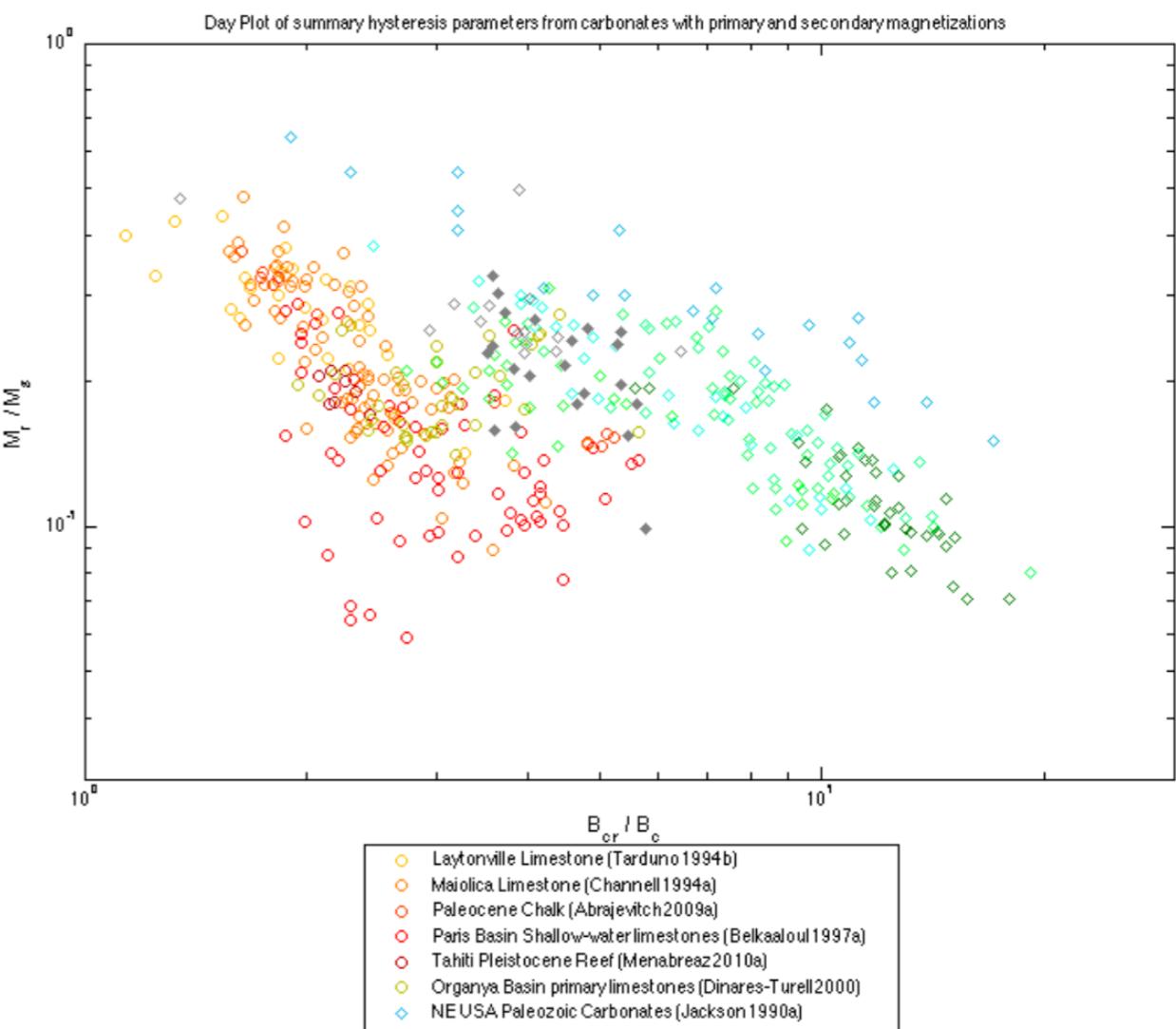
```
[Tarduno1994b_Bcr_Bc Tarduno1994b_Mr_Ms]=textread('Tarduno1994b_white.csv','','delimiter','
```

DAY PLOT LOG AXES

```
DayPlotLog = figure;
%loglog non-remag as circles
loglog(Tarduno1994b_Bcr_Bc,Tarduno1994b_Mr_Ms,'o','MarkerEdgeColor', [1 .75 0])
hold on
loglog(Channell1994a_Bcr_Bc,Channell1994a_Mr_Ms,'o','MarkerEdgeColor', [1 0.5 0])
loglog(Abrajevitch2009a_Bcr_Bc,Abrajevitch2009a_Mr_Ms,'o','MarkerEdgeColor', [1 0.25 0])
loglog(Belkaaloul1997a_Bcr_Bc,Belkaaloul1997a_Mr_Ms,'o','MarkerEdgeColor', [1 0 0])
loglog(Menabreaz2010a_Bcr_Bc,Menabreaz2010a_Mr_Ms,'o','MarkerEdgeColor', [.75 0 0])
loglog(DT2000primary_Bcr_Bc,DT2000primary_Mr_Ms,'o','MarkerEdgeColor', [.75 .75 0])

%loglog remag as diamonds
loglog(Jackson1990a_Bcr_Bc,Jackson1990a_Mr_Ms,'d','MarkerEdgeColor', [0 .7 .9])
loglog(McCabe1994a_Craven_Bcr_Bc,McCabe1994a_Craven_Mr_Ms,'d','MarkerEdgeColor', [0 1 .9])
loglog(McCabe1994a_Onondaga_Bcr_Bc,McCabe1994a_Onondaga_Mr_Ms,'d','MarkerEdgeColor', [0 1 .7])
loglog(McCabe1994a_GreatBasin_Alaska_Bcr_Bc,McCabe1994a_GreatBasin_Alaska_Mr_Ms,'d','MarkerEdgeColor', [0 1 .7])
loglog(Elmore2006a_Bcr_Bc,Elmore2006a_Mr_Ms,'d','MarkerEdgeColor', [0 1 .3])
loglog(Xu1998a_Bcr_Bc,Xu1998a_Mr_Ms,'d','MarkerEdgeColor', [0 1 .1])
loglog(DT2000remag_Bcr_Bc,DT2000remag_Mr_Ms,'d','MarkerEdgeColor', [0 .5 0])
loglog(S_H2012a_core_Bcr_Bc,S_H2012a_core_Mr_Ms,'d','MarkerEdgeColor',[0.5 0.5 0.5],'MarkerFaceColor','k')
loglog(S_H2012a_outcrop_Bcr_Bc,S_H2012a_outcrop_Mr_Ms,'d','MarkerEdgeColor',[0.5 0.5 0.5])
axis([1 30 .03 1])

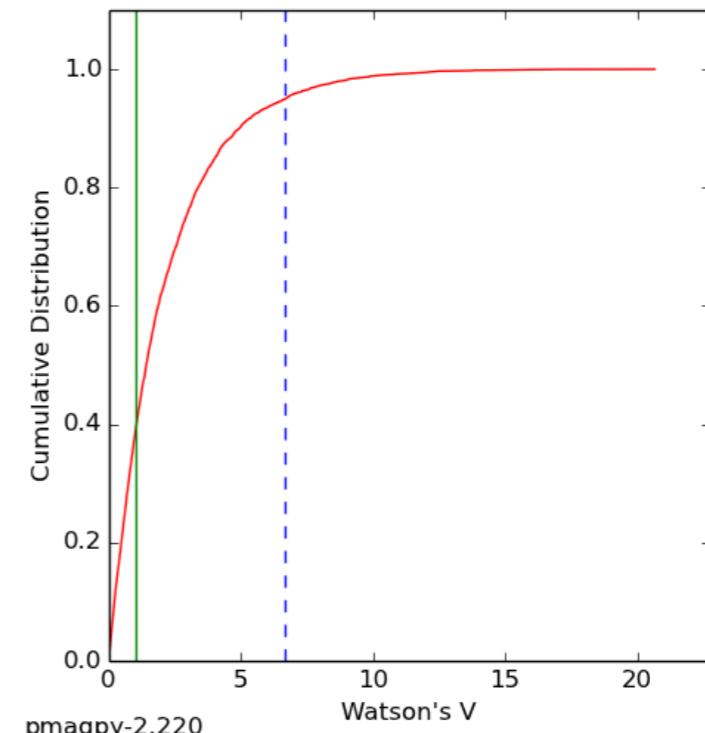
legend('Laytonville Limestone (Tarduno1994b)', 'Maiolica Limestone (Channell1994a)', 'Paleocene  
Paris Basin Shallow-water limestones (Belkaaloul1997a)', 'Tahiti Pleistocene Reef (Menabreaz2010a)',  
'NE USA Paleozoic Carbonates (Jackson1990a)', 'Craven Basin Limestone (McCabe1994a)', 'Onondaga  
Great Basin/Alaska remag carbonates (MacCabe1994a)', 'Ordovician/Devonian WV limestone (Elmore2006a)',  
'Organya Basin remag limestones (Dinares-Turell2000)', 'Bitter Springs Fm core (Swanson-Hysell2012a)',  
'Bitter Springs Fm outcrop (Swanson-Hysell2012a)', 'location', 'southoutside')
xlabel('B_c_r / B_c', 'FontSize',12)
ylabel('M_r / M_s', 'FontSize',12)
title('Day Plot of summary hysteresis parameters from carbonates with primary and secondary magnetizations')
```



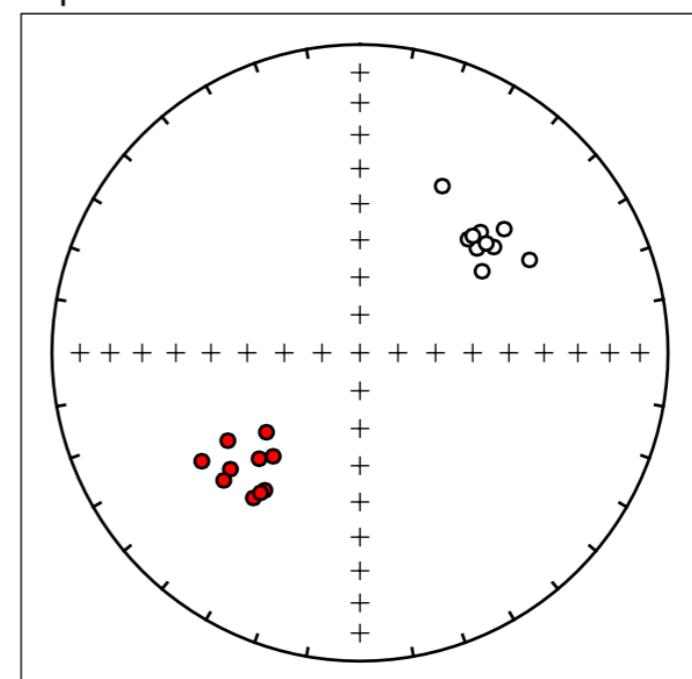


Python-based tools for Paleomagnetic research <http://scripps.ucsd.edu>

```
Nicks-MacBook-Pro:Pmag_Workspace polarwander$ revtest_MM1990.py -f Fisher_data1.txt -f2 Fisher_data2.txt  
be patient, your computer is doing 5000 simulations...  
  
Results of Watson V test:  
  
Watson's V: 1.0  
Critical value of V: 6.7  
"Pass": Since V is less than Vcrit, the null hypothesis that the two populations  
are drawn from distributions that share a common mean direction (antipodal to one another) cannot be rejected.  
  
M&M1990 classification:  
  
Angle between data set means: 2.7  
Critical angle of M&M1990: 6.8  
The McFadden and McElhinny (1990) classification for this test is: 'B'  
S[a]ve to save plot, [q]uit without saving: [ ]
```



Equal Area Plot





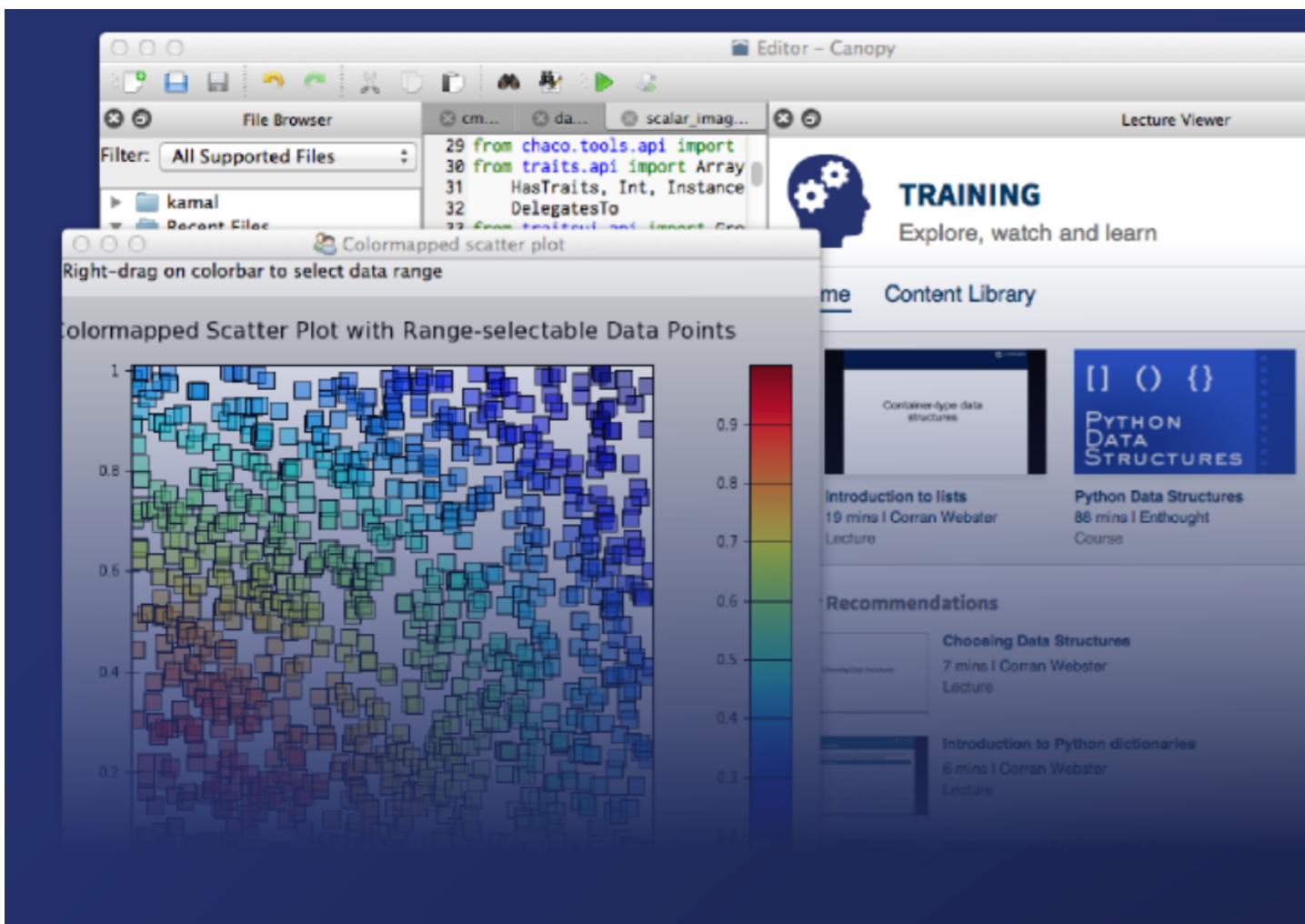
Python-based tools for Paleomagnetic research <http://scripps.ucsd.edu>

Source Code for Module fisher

```
1 #!/usr/bin/env python
2 import pmag,sys
3 -def spitout(kappa):
4     dec,inc= pmag.fshdev(kappa) # send kappa to fshdev
5     print '%7.1f %7.1f ' % (dec,inc)
6     return
7 -def main():
8     """
9     NAME
10        fisher.py
11
12    DESCRIPTION
13        generates set of Fisher distributed data from specified distribution
..
```



Python-based tools for Paleomagnetic research <http://scripps.ucsd.edu>



Data analysis and visualization. Python distribution with GUI. Integrated code editor and IPython console. Graphical package management. Integrated interactive training platform. Easy installation and deployment.

[Learn More ➔](#)

The IPython notebook

A browser-based interactive computational environment where you can combine code execution, text, mathematics, plots and rich media into a single document

The screenshot shows the IPython Notebook interface. The title bar says "IPy Python Dashboard" and "IPy spectrogram". The URL is "127.0.0.1:8888/a5222740-848b-4ac1-b212-d732c9f8f78b". The main area is titled "IP[y]: Notebook" and "spectrogram" with "Last saved: Mar 07 11:14 PM". The menu bar includes File, Edit, View, Insert, Cell, Kernel, Help. Below the menu is a toolbar with various icons. The content area starts with "Simple spectral analysis" and "An illustration of the Discrete Fourier Transform". It shows the formula $X_k = \sum_{n=0}^{N-1} x_n e^{-\frac{2\pi i}{N} kn}$ for $k = 0, \dots, N - 1$. It then discusses windowing to reveal frequency content of a sound signal. A code cell "In [1]" shows the command `from scipy.io import wavfile
rate, x = wavfile.read('test_mono.wav')`. Another code cell "In [2]" shows `fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(12, 4))
ax1.plot(x); ax1.set_title('Raw audio signal')
ax2.specgram(x); ax2.set_title('Spectrogram');`. Below these are two plots: "Raw audio signal" showing a blue waveform from 0 to 50000 samples, and "Spectrogram" showing a heatmap of frequency over time from 0 to 25000 samples.

- Interactive
- Exploratory
- Collaborative
- Open
- Reproducible

The IPython notebook

A browser-based interactive computational environment where you can combine code execution, text, mathematics, plots and rich media into a single document

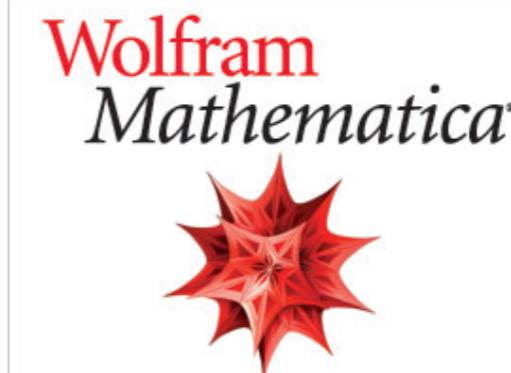
The screenshot shows the IPython Notebook interface. The title bar says "IPy Python Dashboard" and "IPy spectrogram". The main area is titled "IP[y]: Notebook" and "spectrogram" with "Last saved: Mar 07 11:14 PM". The menu bar includes File, Edit, View, Insert, Cell, Kernel, Help. Below the menu is a toolbar with various icons. The main content area has a section titled "Simple spectral analysis" with the subtext "An illustration of the Discrete Fourier Transform". It shows the formula for the DFT:
$$X_k = \sum_{n=0}^{N-1} x_n e^{-\frac{2\pi i}{N} kn} \quad k = 0, \dots, N - 1$$
. Below the formula, it says "using windowing, to reveal the frequency content of a sound signal." A code cell "In [1]" contains:

```
from scipy.io import wavfile  
rate, x = wavfile.read('test_mono.wav')
```

. Another code cell "In [2]" contains:

```
fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(12, 4))  
ax1.plot(x); ax1.set_title('Raw audio signal')  
ax2.specgram(x); ax2.set_title('Spectrogram');
```

. Below the code cells are two plots: "Raw audio signal" showing a blue waveform from 0 to 50000 samples, and "Spectrogram" showing a heatmap of frequency over time from 0 to 25000 samples.





Python-based tools for Paleomagnetic research <http://scripps.ucsd.edu>

- pmag.py

```
3300 def fshdev(k):
3301     """
3302         returns a direction from distribution with mean declination of 0, inclination of 90 and kappa of k
3303     """
3304     R1=random.random()
3305     R2=random.random()
3306     L=numpy.exp(-2*k)
3307     a=R1*(1-L)+L
3308     fac=numpy.sqrt((-numpy.log(a))/(2*k))
3309     inc=90.-2*numpy.arcsin(fac)*180./numpy.pi
3310     dec=2*numpy.pi*R2*180./numpy.pi
3311     return dec,inc
3312
```

- pmagplotlib.py

```
346 def plotNET(fignum):
347     """
348         draws circle and tick marks for equal area projection
349     """
350 #
351 # make the perimeter
352 #
353     pylab.figure(num=fignum)
354     pylab.clf()
355     if not isServer: pylab.figtext(.02,.01,version_num)
356     pylab.axis("off")
357     Dcirc=numpy.arange(0,361.)
```

PUBLIC



Itaixe / PmagPy

Python-based tools for Paleomagnetic research <http://scripps.ucsd.edu>

- pmag.py
- pmagplotlib.py

PUBLIC



Swanson-Hysell / PmagPy_IPython

Pmag function libraries for use within IPython notebooks

- IPmag.py

Import and display data

Import Simpson Island paleomagnetic data

The data file FlowDataAll.csv has data in this format:

SITE, STRAT_HEIGHT, DEC_GEO, INC_GEO, Dec_TC, INC_TC, A95, n, Plat, ABS_Platt, VGP_lat, VGP_long

where:

SITE is the name of the site and corresponds to an individual lava flow where in the first part of the site name (e.g. SI1) corresponds to the name of a stratigraphic section while the second part (e.g. (11.8 to 26.4)) corresponds to the stratigraphic ranges within the measured section over which the flows is exposed.

STRAT_HEIGHT is the cumulative stratigraphic height of the Simpson Island stratigraphy starting with the base of the SI1 section (see map in the main manuscript).

Table of Simpson Island paleomagnetic data

```
In [3]: data = pandas.read_csv('../2014_Osler_Data/SimpsonIsland_OslerData.csv')
#write these data to a latex file for the supplemental PDF
with open('OslerData.txt','w') as f:
    f.write(data.to_latex(index=False,columns=['Site','Strat_Height','Site_Lat','Site_Long','Dec_Geo','Inc_Geo','Dec_TC','Inc_TC','A95','n','Plat','ABS_Platt','VGP_lat','VGP_long']))
    display(HTML(data.to_html()))
```

	Site	Strat_Height	Site_Lat	Site_Long	Dec_Geo	Inc_Geo	Dec_TC	Inc_TC	A95	n	Plat	ABS_Platt	VGP_lat	VGP_long
0	SI1(11.8 to 26.4)	11.8	48.8122	-87.6620	120.3	-77.1	79.7	-70.5	2.7	7	-54.7	54.7	33.1	229.6
1	SI1(28.3 to 29.2)	28.3	48.8107	-87.6623	141.0	-67.4	109.8	-66.8	3.9	7	-49.4	49.4	45.8	210.9
2	SI1(29.2 to 29.7)	29.2	48.8104	-87.6622	131.2	-69.1	99.7	-66.2	5.7	3	-48.6	48.6	39.6	214.5
3	SI1(33.3 to 37.1)	33.3	48.8100	-87.6623	127.7	-72.3	92.6	-68.1	7.7	5	-51.2	51.2	37.2	220.5
4	SI1(40.6 to 42.5)	40.6	48.8095	-87.6626	134.9	-77.7	85.4	-73.3	3.5	6	-59.0	59.0	38.2	231.6
5	SI1(42.5 to 44.4)	42.5	48.8095	-87.6627	103.0	-71.6	77.8	-63.5	4.5	6	-45.1	45.1	25.8	222.3
6	SI1(58.1 to 64.1)	58.1	48.8086	-87.6622	114.4	-70.8	86.1	-64.4	8.7	5	-46.2	46.2	30.8	218.8
7	SI1(87.6 to 88.5)	87.6	48.8073	-87.6620	134.3	-75.8	89.9	-71.9	1.9	6	-56.9	56.9	39.0	227.7
8	SI1(88.5 to 89.2)	88.5	48.8071	-87.6619	128.2	-74.6	88.9	-70.0	5.0	5	-53.9	53.9	36.9	224.9
9	SI1(89.2 to 91.5)	89.2	48.8070	-87.6618	120.1	-72.8	87.0	-67.2	7.6	5	-49.9	49.9	33.6	221.8
10	SI1(91.5 to 92.6)	91.5	48.8069	-87.6618	118.4	-74.8	82.8	-68.5	3.0	6	-51.8	51.8	32.7	225.5

Inline statistical results and plots

Calculating and plotting mean pole positions

The code below calculates the Fisher mean parameters for poles of the stratigraphically grouped virtual geomagnetic poles (VGPs) and then plots them on a "view from space" globe. Note that while where the textual output says "dec" it actually refers to pole longitude and where it says "inc" it actually refers to pole latitude since the pmag.fisher_mean function is being conducted on VGPs.

```
In [15]: LowerThird_MeanPole=pmag.fisher_mean(SI_LowerThird_Poles)
MiddleThird_MeanPole=pmag.fisher_mean(SI_MiddleThird_Poles)
UpperThird_MeanPole=pmag.fisher_mean(SI_UpperThird_Poles)

IPmag.poleplot(m,UpperThird_MeanPole[ 'dec' ],UpperThird_MeanPole[ 'inc' ],
               UpperThird_MeanPole[ 'alpha95' ],color='b',label='Osler Group Upper Reversed Pole')

#show the plot
legend(bbox_to_anchor=(1.05, 1), loc=2, borderaxespad=0.)
plt.show()
```

The Fisher mean parameters for the LowerThird mean pole are:

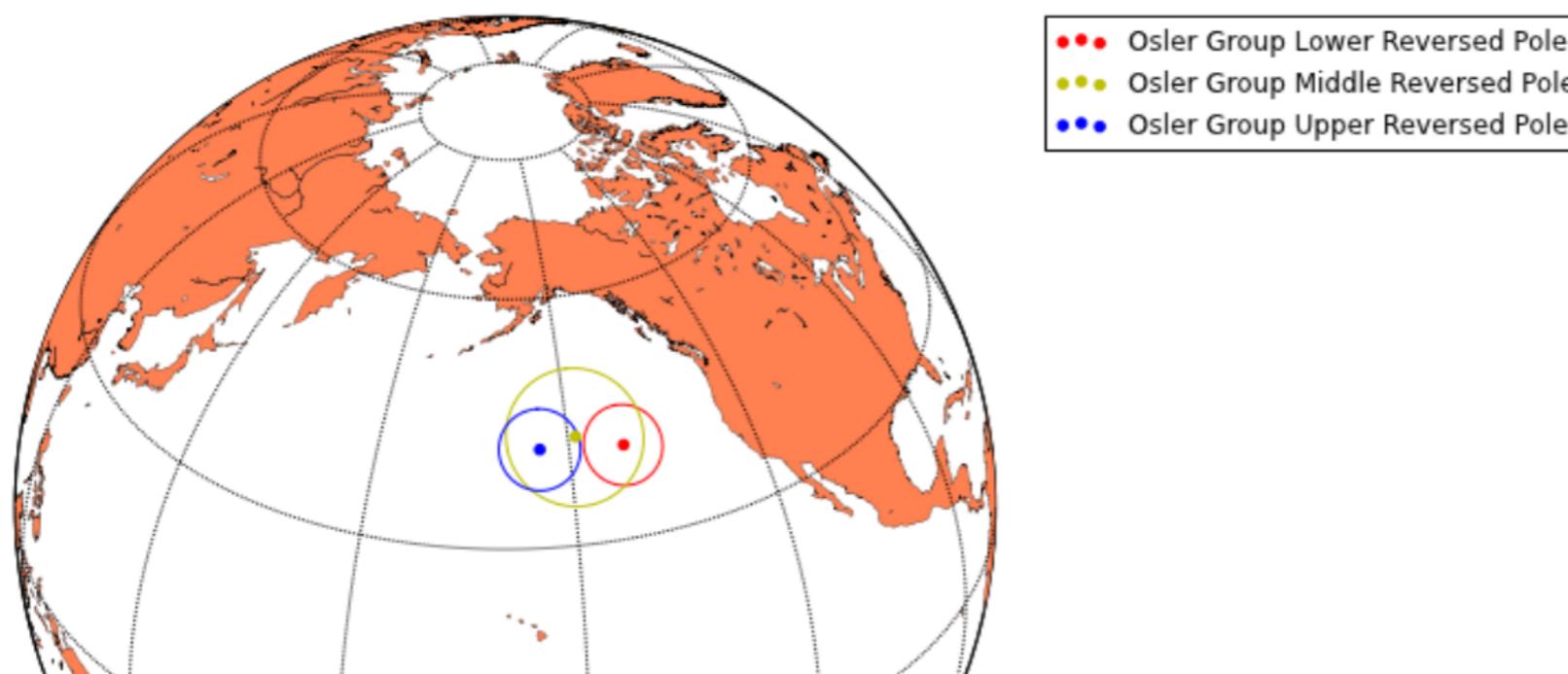
Plong = 218.637832609 Plat = 40.937318541
A95 = 4.76006232039 k= 31.4671112908

The Fisher mean parameters for the MiddleThird mean pole are:

Plong = 211.261736125 Plat = 42.7378078852
A95 = 8.1783359746 k= 16.9030328287

The Fisher mean parameters for the UpperThird mean pole are:

Plong = 205.410227513 Plat = 41.5645761829
A95 = 4.80394668574 k= 27.2260167637



Inline statistical results and plots

Common mean tests between SI_LowerThird_Directions and SI_MiddleThird_Directions:

```
In [6]: IPmag.iWatsonV(SI_LowerThird_Directions,SI_MiddleThird_Directions)  
IPmag.iBootstrap(SI_LowerThird_Directions,SI_MiddleThird_Directions)
```

Results of Watson V test:

Watson's V: 2.6

Critical value of V: 6.2

"Pass": Since V is less than Vcrit, the null hypothesis that the two populations are drawn from distributions that share a common mean direction can not be rejected.

M&M1990 classification:

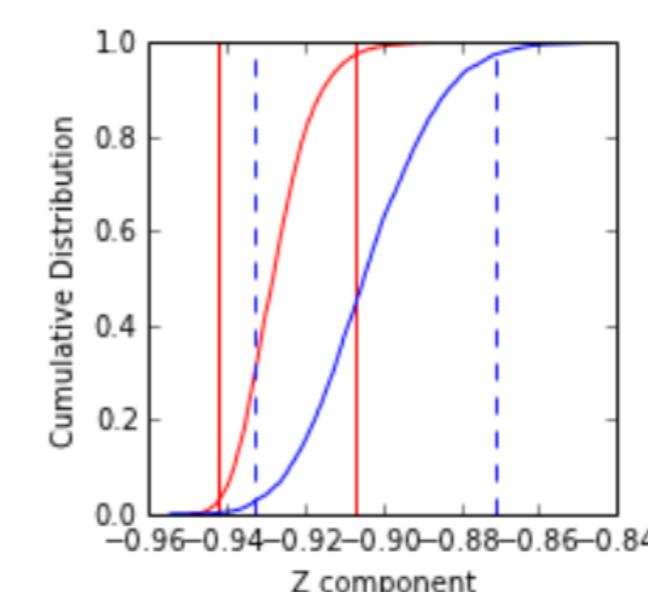
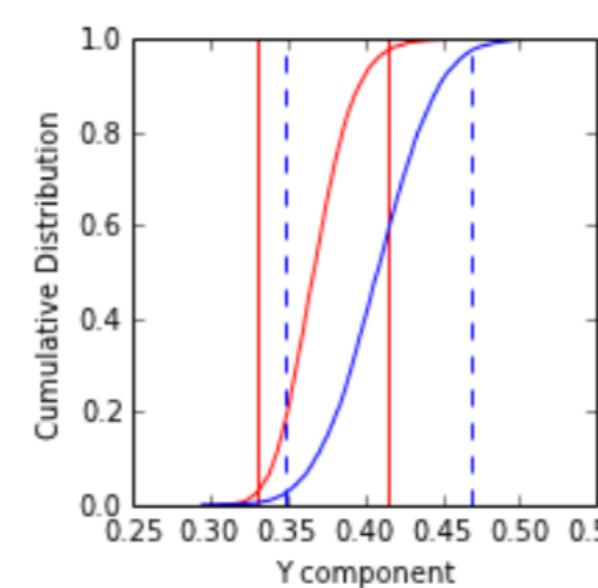
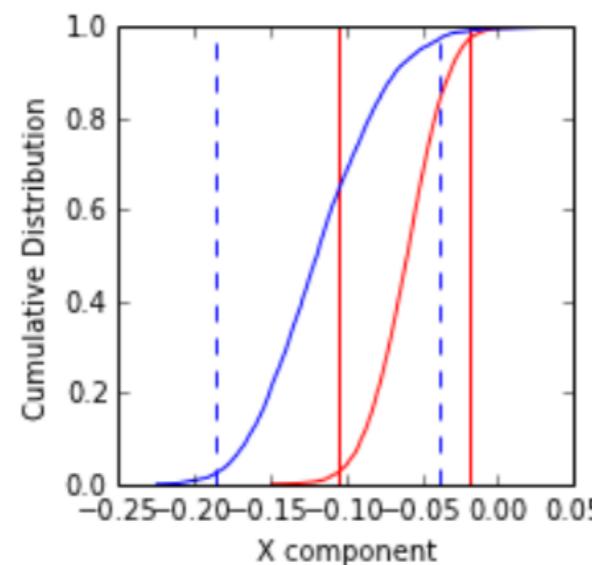
Angle between data set means: 4.1

Critical angle for M&M1990: 6.3

The McFadden and McElhinny (1990) classification for this test is: 'B'

=====

Here are the results of the bootstrap test for a common mean



Inline text and graphics

Paleogeographic work outside of this IPython notebook using the software package GPlates.

For details on working with paleomagnetic data in GPlates check out [this tutorial](#).

The steps necessary to develop the reconstruction using these Osler Volcanic Group poles are detailed below.

(1) Develop .rot file that rotates Laurentia such that the Osler poles are at geographic north. In the text for the .rot file below the Osler_lowerthird pole is assigned to 1110 Ma, the Osler_middlethird pole is assigned to 1107.5 and the Osler_upperthird pole is assigned to 1105 Ma.

The .rot file has the format:

moving plate ID number	age in millions of years	latitude of Euler pole	longitude of Euler pole	angle	fixed plate	comment
------------------------	--------------------------	------------------------	-------------------------	-------	-------------	---------

In this .rot file, the fixed reference frame is 000 and 001 (which could be split into a relative and absolute reference frame that separates out TPW), while the plate ID for Laurentia is 199.

```
001  0.0  0.0  0.0  0.0  000 ! \\
001 3900.0  0.0  0.0  0.0  000 ! \\
199  0.0  0.0  0.0  0.0  001 ! \\
199 1105.0  0.0  115.4  48.4  001 !Put Osler-upperthird at N for Laurentia \\
199 1107.5  0.0  121.3  47.3  001 !Put Osler-middlethird at N for Laurentia \\
199 1110.0  0.0  128.7  49.0  001 !Put Osler-lowerthird at N for Laurentia \\
```

(2) Open the following feature collections in GPlates: 199-Laurentia nogrid.dat, OslerPoles.gpml and Osler.rot where 199-Laurentia nogrid.dat is the outline of Laurentia in the late Proterozoic and OslerPoles.gpml contains the three calculated Osler paleomagnetic poles. An animation from 1110 to 1105 Ma yields the following reconstruction shown here with a time slice for Osler_lowerthird (red), Osler_middlethird (yellow) and Osler_upperthird (blue).

```
In [23]: from IPython.display import SVG
SVG("Paleogeog.svg")
```

Out[23]:



Easily versioned controlled and made available in an archive like Github

The screenshot shows a GitHub repository page. At the top, there's a navigation bar with links for 'Explore', 'Gist', 'Blog', and 'Help'. On the right, it shows the user 'Swanson-Hysell' with a profile picture, and icons for 'Unwatch', 'Star', 'Fork', and a pull request counter (2). Below the header, the repository name 'Swanson-Hysell / 2014_Swanson-Hysell-et-al_Osler' is displayed, along with a 'PUBLIC' label and a 'Website' link.

Description: Short description of this repository

Website: Website for this repository (optional)

Code Metrics: 48 commits, 1 branch, 0 releases, 1 contributor

Branch Selection: branch: master

Recent Activity:

Author	Commit Message	Date
Swanson-Hysell	authored 5 days ago	latest commit d5e95dbb9f
2014_Osler_Code	Add AGU readme text to post nbconvert notes file	5 days ago
2014_Osler_Data	Optimize IPython notebook for Latex output, generate revised latex/PDF	9 days ago
2014_Osler_Manuscript_Files	Final revisions before submitting revised ms to AGU	9 days ago
README.md	update README.md to describe revisions/additions	8 days ago

README.md:

Overview

This repository contains a revised manuscript for publication at the American Geophysical Union journal entitled "Geochemistry Geophys Geosystems" along with the raw data and code used for the manuscript's data analysis. The manuscript is entitled "Confirmation of progressive plate motion during the Midcontinent Rift's early magmatic stage from the Osler Volcanic Group, Ontario, Canada" with the authors being: Nicholas. L. Swanson-Hysell, Angus A. Vaughan, Monica R. Mustain, and Kristofer Asp.

Code Options:

- Issues (0)
- Pull Requests (0)
- Wiki
- Pulse
- Graphs
- Network
- Settings

Clone Options:

- HTTPS clone URL: <https://github.com/>
- You can clone with [HTTPS](#), [SSH](#), or [Subversion](#).
- Clone in Desktop
- Download ZIP

Easily viewed/shared as static html

<http://nbviewer.ipython.org>

nbviewer FAQ IPython

nbviewer

A simple way to share IPython Notebooks

URL | GitHub username | GitHub username/repo | Gist ID

Programming Languages

IPython



In [9]: `display(i)`
IP[y]: IPython Interactive Computing
In [3]: `from IPython.display import SVG
SVG(filename='python-logo.svg')`
Out[3]:  python™

IRuby



File.open('lib/iruby/static/base/images/ipynblogo.png')

IRuby: Notebook

IJulia

An IJulia Preview

This notebook is a preview demo of IJulia: a [Julia-language](#) backend combined with the [IPython](#) interactive environment. This combination allows you to interact with the Julia language using IPython's powerful [graphical notebook](#), which combines code, formatted text, math, and multimedia in a single document.

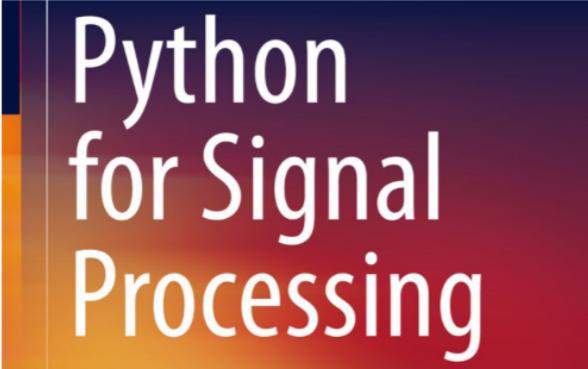


Note: this is a preview, because it relies on pre-release bleeding-edge versions of Julia, IPython, and several Julia packages, as explained on the [IJulia github page](#), and functionality is evolving rapidly. We hope to have a more polished release soon.

Basic Julia interaction

Books

Python for Signal Processing



Python for Signal Processing

O'Reilly Book



O'REILLY®
2nd Edition
Mining the Social Web
DATA MINING FACEBOOK, TWITTER, LINKEDIN, GOOGLE+, GITHUB, AND MORE

Probabilistic Programming



PROBABILISTIC PROGRAMMING & BAYESIAN METHODS FOR HACKERS

Easily exported to PDF

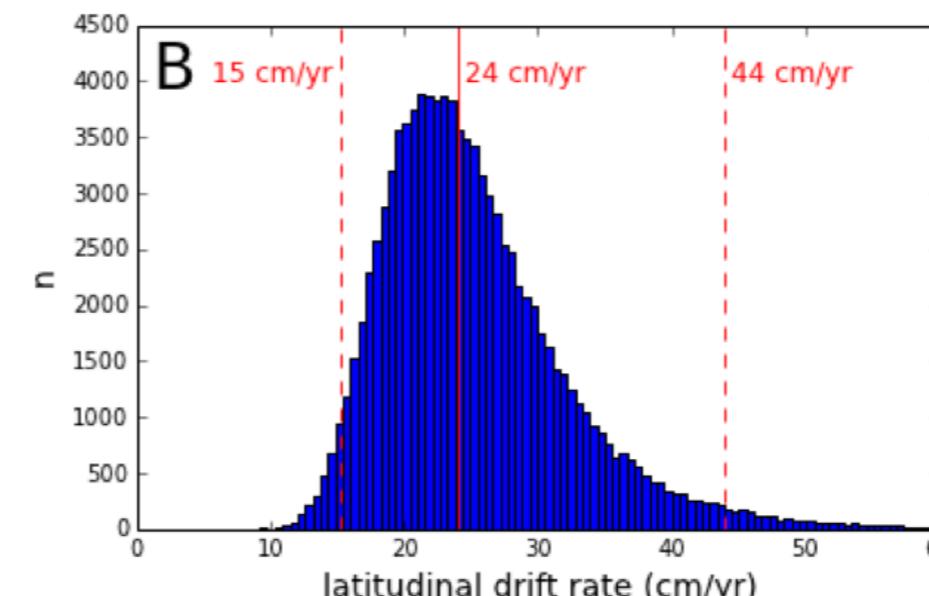
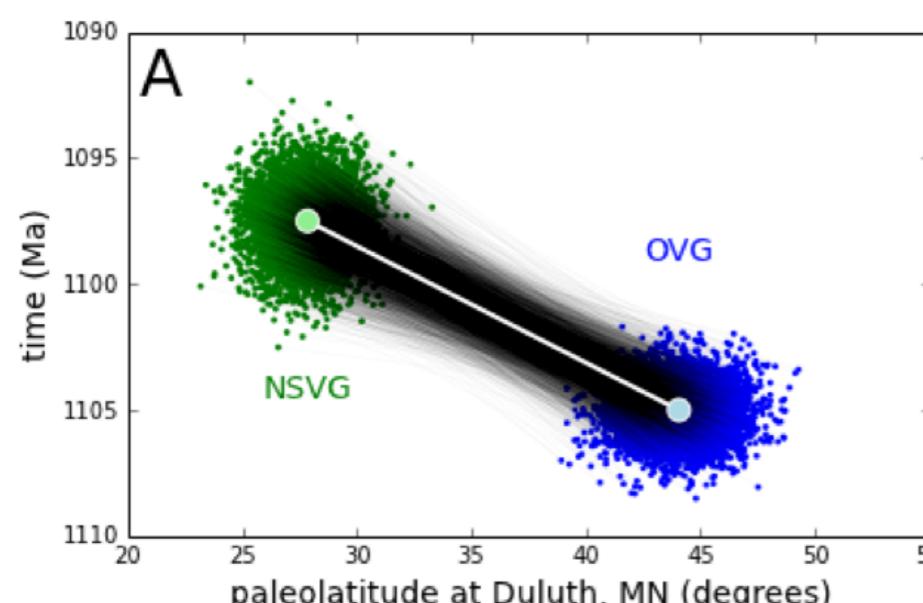
9.5 Calculate and graphically display the confidence interval on the minimum rate estimate implied by the Monte Carlo sampled pairs of sampled ages and paleolatitudes from the two poles

The pairs of sampled ages and poles can be used to calculate rates (with the total number of rates being set by `samplesize` in the input box above). The rate calculated above of 24.0 cm/year can now be given confidence bounds by taking 2.5 percentile and 97.5 percentile of the Monte Carlo simulated rates.

In [32]: *#calculating the change in paleolatitude between the Monte Carlo pairs*

```
pole1_pole2_Delta_degrees=[]
pole1_pole2_Delta_kilometers=[]
pole1_pole2_Delta_myr=[]
pole1_pole2_degrees_per_myr=[]
pole1_pole2_cm_per_yr=[]

plt.xlabel('latitudinal drift rate (cm/yr)',size=14)
plt.xlim([0,60])
plt.ylim([0,4500])
plt.savefig('../2014_Osler_Manuscript_Files/2014_Osler_Figures/MonteCarlo.pdf',
            bbox_inches='tight')
plt.show()
```



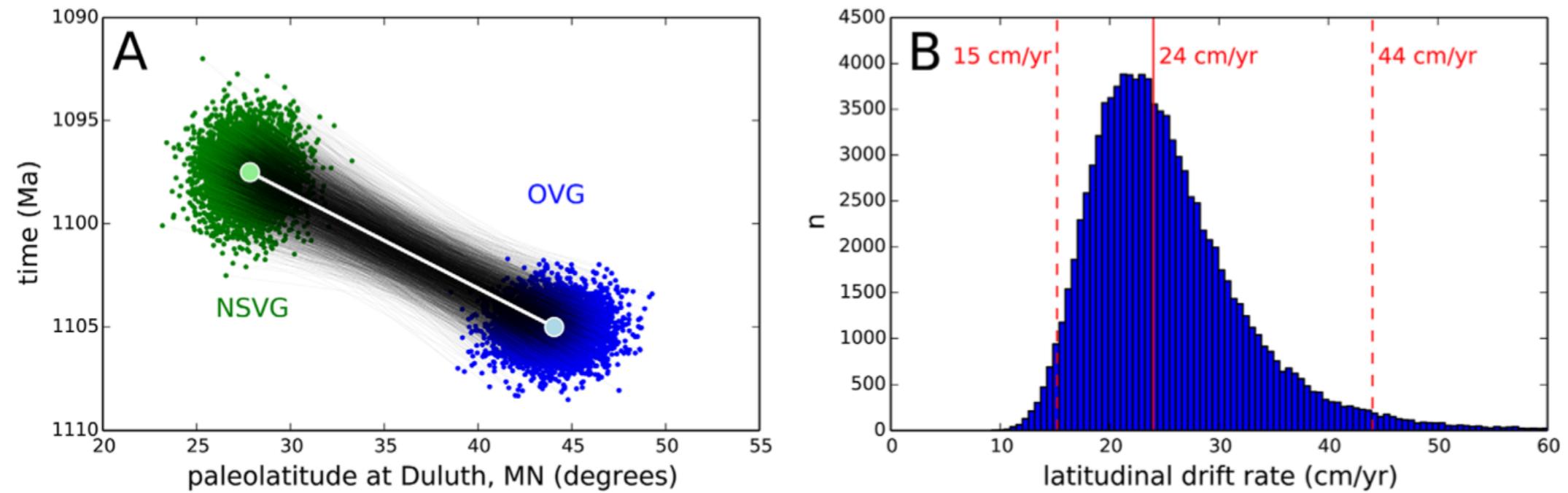


Figure 4. Monte Carlo simulations of rates implied by paleomagnetic and geochronologic data between the Osler Volcanic Group (OVG) and southwest limb of the upper North Shore Volcanic Group (NSVG). In (a), the white outlined points connected by a white line represent the mean paleolatitude (calculated for Duluth, MN) and mean age of the poles. The green and blue points are 5000 of the 100,000 simulated pole pairs (OVG and NSVG, respectively) used in the analysis and are connected by gray lines which represent 5000 of the simulated rates. The histogram in (b) shows all 100,000 of the simulated rates that yield the labeled 2.5 percentile value of 15 cm/yr, median value of 24 cm/yr (also the value as calculated from the means) and 97.5 percentile value of 44 cm/yr.

Live Demo

Questions?

Open and reproducible paleomagnetic data analysis using IPython

Nicholas L. Swanson-Hysell

Earth and Planetary Science Department // University of California, Berkeley

Abstract

For research to be truly reproducible all of the code used to produce results needs to be thoroughly documented and openly presented in addition to the data. Given that many scientific workflows include a variety of software tools to conduct data analysis and visualization, the level of documentation necessary to fully reproduce plots and statistical analyses is rarely achieved. In this talk, I will describe how the language-agnostic IPython notebook environment (<http://ipython.org>) can remove impediments for documenting the work flow of paleomagnetic data analysis in conjunction with functions developed through the PmagPy open source software project.