Supplement of Solid Earth, 8, 1193–1209, 2017
https://doi.org/10.5194/se-8-1193-2017-supplement

Solid Earth
Open Access

Supplement of

Hierarchical creep cavity formation in an ultramylonite and implications for phase mixing

James Gilgannon et al.

Correspondence to: James Gilgannon (james.gilgannon@geo.unibe.ch)

1 Image analysis

The pre-processing workflow was as follows:

1. Make image 8 bit
2. Despeckle
3. Non-local Means Denoise (sigma = 5, smoothing = 1)
4. Non-local Means Denoise (sigma = 15, smoothing = 1)
5. Threshold (<10)
6. Make binary
7. Manually erase cross cutting pores

The workflow for the mask used used in figure 2 was as follows:

1. Make image 8 bit
2. Non-local Means Denoise (sigma = 15, smoothing = 1)
3. Remove outliers: Dark (radius = 4, threshold = 50)
4. Threshold (<50)
5. Make binary
6. Remove outliers: Bright (radius = 12, threshold = 50)
7. Threshold (<50)
8. Make binary

Parameters for kernel density analysis using SciPy and NumPy libraries:

```
'''
Part of script for calculating Kernel density analysis (KDE) using SciPy and NumPy libraries.

NOTE: KDE must be rotated 90 degrees when visualising
'''

# Import Python libraries
from scipy import stats
import numpy as np

# Define objects to preform kernel density analysis on
area_obj = np.asarray(less_1_circ['area'])
angle_obj = np.asarray(less_1_circ['angle_90'])

### Kernel density analysis (KDE) ###
# Make analysis grid
X, Y = np.mgrid[xbnds[0]:xbnds[1]:100j, ybnds[0]:ybnds[1]:100j]
positions = np.vstack([X.ravel(), Y.ravel()])
# Use objects to make stack for KDE
values = np.vstack([area_obj, angle_obj])
# Bandwith for manipulating the kernels shape
bandwidth = 4
# KDE analysis
kernel = stats.gaussian_kde(values, bw_method = bandwidth / values.std(ddof=1))
Z = np.reshape(kernel(positions).T, X.shape)
```

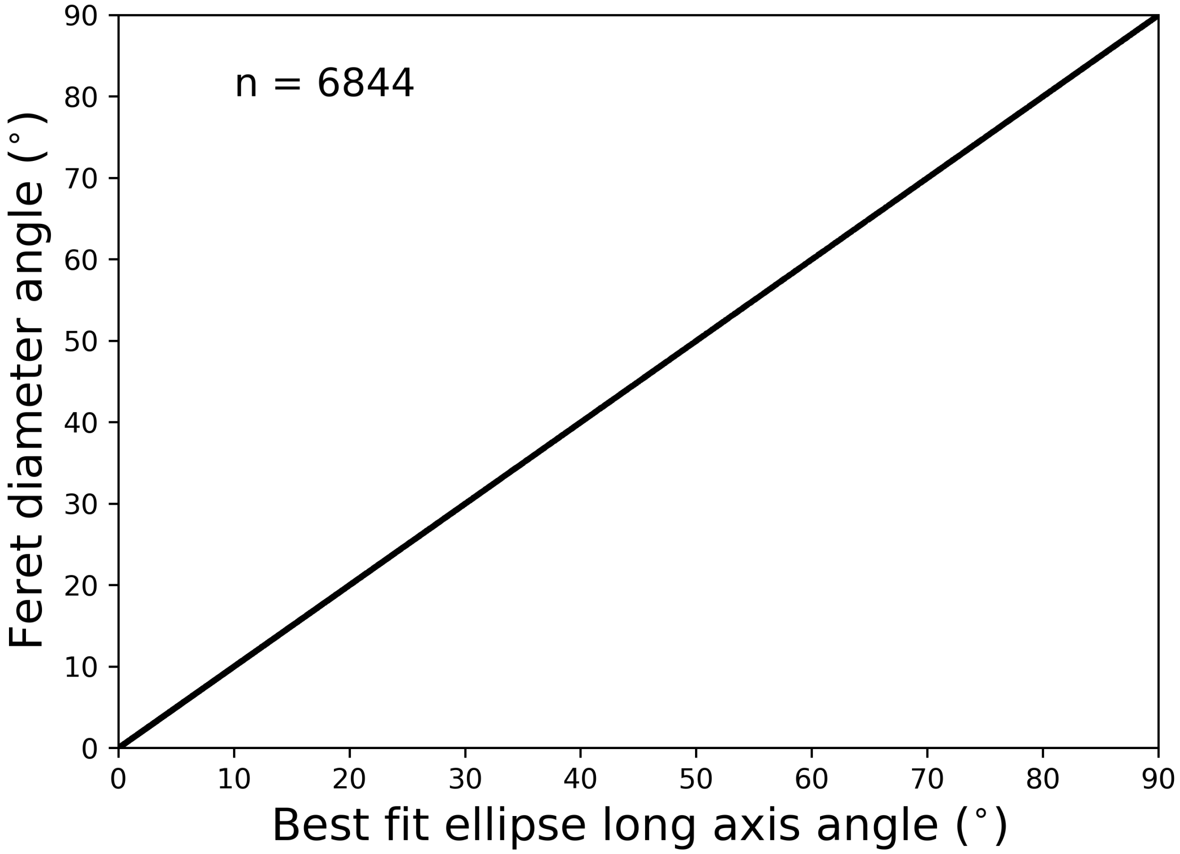

Figure S 1: Comparison of the Feret's diameter and the long axis of the best fit ellipse, for pores with a circularity < 1 .