FracPaQ

User Guide

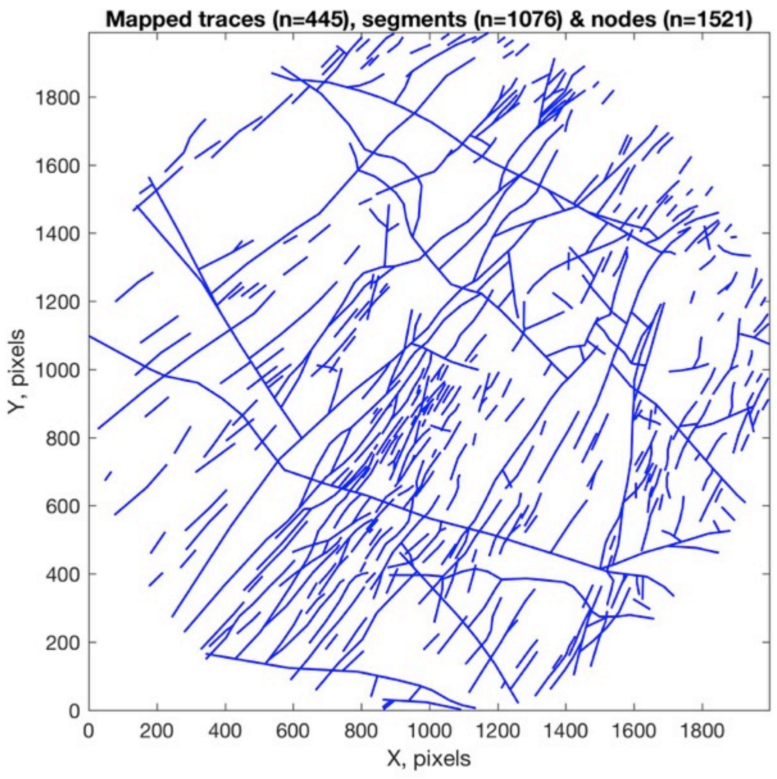
Version 1.6a, November 2016

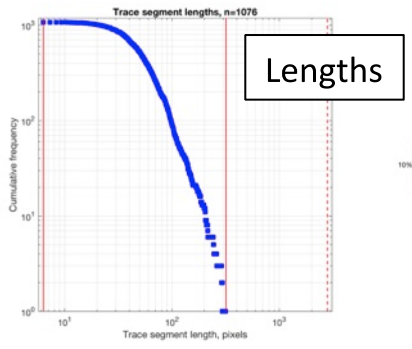
Introduction

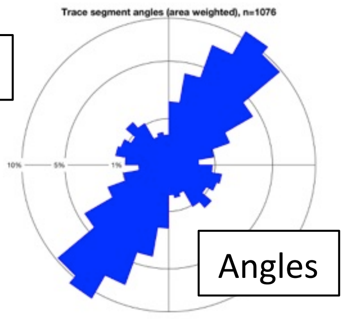
FracPaQ is an open source toolbox written in MATLAB™ and publicly available on GitHub and Mathworks FileExchange. FracPaQ runs on any computer with MATLAB installed – Windows, OS X or Linux. The initial release version is v1.6a.

FracPaQ is designed to quantify fracture patterns in rock. The user supplies either an image file of fractured rock or a text file of traced fractures and their (x,y) coordinates. From either kind of input, the code calculates the fracture lengths, angles and connectivity. These are displayed as maps and graphs, and saved as *.tif files. Estimates are made of fracture intensity (P21), fracture density (P20), and permeability in 2D using a simple parallel-plate model and assuming constant aperture.

We hope that the code will enable researchers to quantify fracture patterns in an open, objective, and consistent way. **We also hope that people will contribute new functions and new tools, and report any bugs ☺**





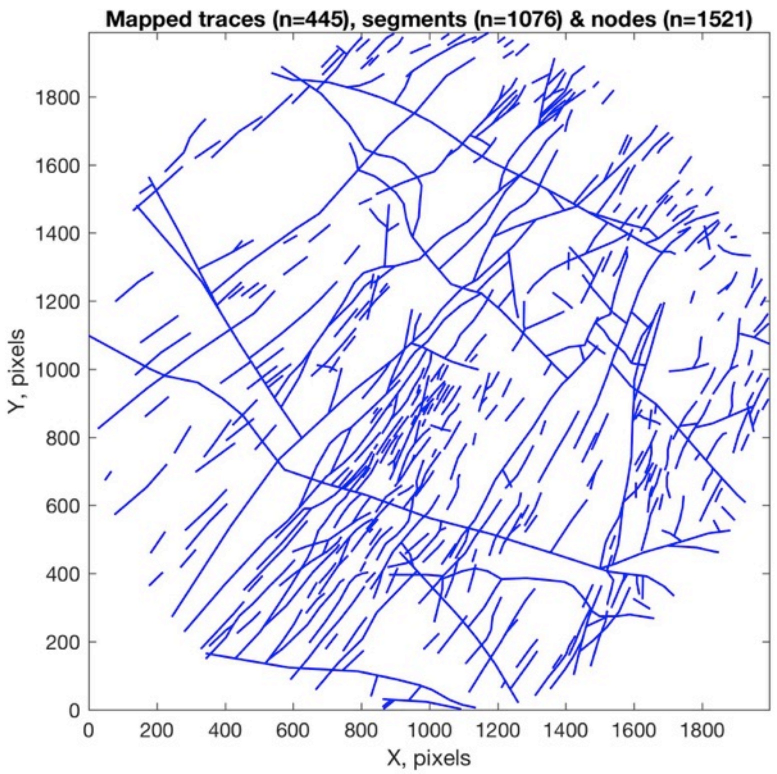


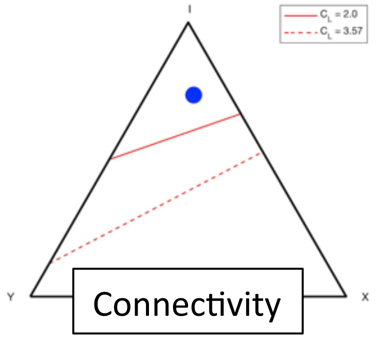
Introduction

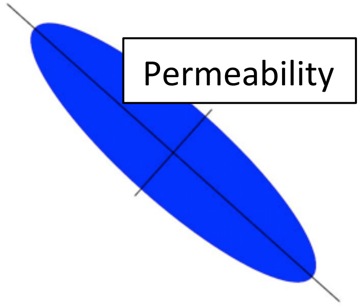
FracPaQ is written in MATLAB™ and totals over 4,000 lines of code. As in any software project of this scale, there will be 'bugs' – i.e. coding errors. If you encounter a bug, please let us know – through GitHub, Mathworks FileExchange or by e-mail (d.healy@abdn.ac.uk); please provide as many details as you can, including (where possible) a screen shot of the error, the input file you were using at the time, and the MATLAB version.

FracPaQ uses code written by others: **`lineSegmentIntersect.m`** by U. Murat Erdem and **`readtext.m`** by Peder Axensten (both available on Mathworks FileExchange).

The handling of image file input in **FracPaQ** also uses functions from the Image Processing Toolbox (version 9.4), a MATLAB™ add-on.







Installing FracPaQ

You can get all the source code and run it directly from the MATLAB command window.

The source code and this User Guide are available on **GitHub**:
http://davehealy-aberdeen.github.io/FracPaQ/

and on the **Mathworks FileExchange**:
https://uk.mathworks.com/matlabcentral/fileexchange/58860-davehealy-aberdeen-fracpaq

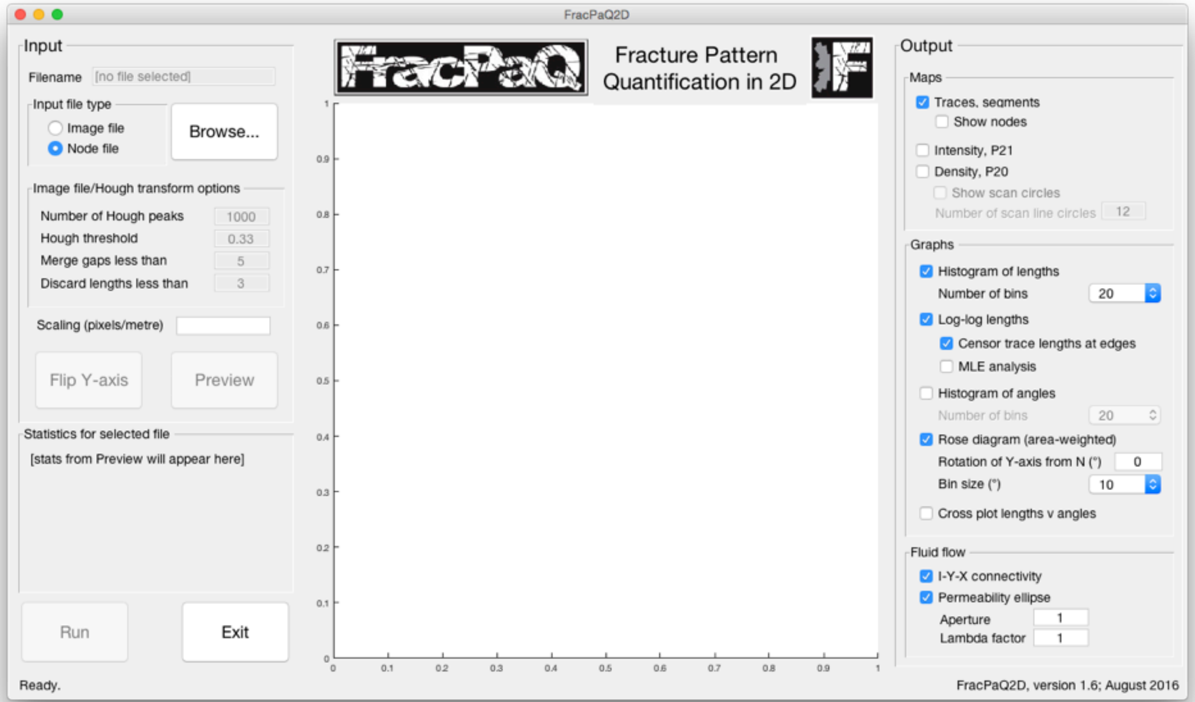
Steps:
1. Download the source code as a .zip or .tar.gz file, and extract all of the files
2. Put all of these files into a single folder
3. Start MATLAB (NB. we developed FracPaQ using MATLAB R2016a)
4. Set the current working folder in MATLAB to the folder you installed the code in
5. At the MATLAB command prompt type 'guiFracPaQ2D' and hit Enter

Output files (graphs, maps and data) will appear as *.tif and *.txt files in the same folder.

Starting FracPaQ

There is only one window in the application (see right). The window can be minimised or closed using the standard GUI controls (top left in the Mac OS X version shown).

Input parameters are on the left side, and output options are on the right. The central area is for previewing the input data as a fracture trace map.


FracPaQ2D

Fracture Pattern Quantification in 2D

Input

Filename [no file selected]

Input file type: Image file [✓ Node file]

Image file/Hough transform options
- Number of Hough peaks: 1000
- Hough threshold: 0.33
- Merge gaps less than: 5
- Discard lengths less than: 3

Scaling (pixels/metre)

Statistics for selected file

[stats from Preview will appear here]

Output

Maps
- ✓ Traces, segments
- ☐ Show nodes
- ☐ Intensity, P21
- ☐ Density, P20
- ☐ Show scan circles
- Number of scan line circles: 12

Graphs
- ✓ Histogram of lengths
- Number of bins: 20
- ✓ Log-log lengths
- ✓ Censor trace lengths at edges
- ☐ MLE analysis
- ☐ Histogram of angles
- Number of bins: 20
- ✓ Rose diagram (area-weighted)
- Rotation of Y-axis from N (°): 0
- Bin size (°): 10
- ☐ Cross plot lengths v angles

Fluid flow
- ✓ I-Y-X connectivity
- ✓ Permeability ellipse
- Aperture: 1
- Lambda factor: 1

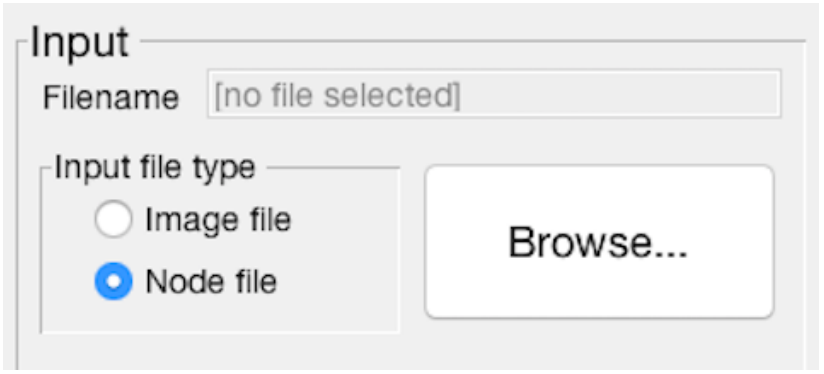
Ready.

FracPaQ2D, version 1.6; August 2016

To get started, click **Browse…** to see a list of possible input files in the current working folder. Then click **Preview** to open the data file and view the traces in the main window; basic statistics on the pattern are also shown (lower left box). Select the Outputs you need (right hand side), and then click **Run** to produce the selected maps and graphs. Each output is shown in a separate figure window, and saved as a separate .jpeg in the current folder.

Click **Exit** to quit **FracPaQ**.

Choosing an input file

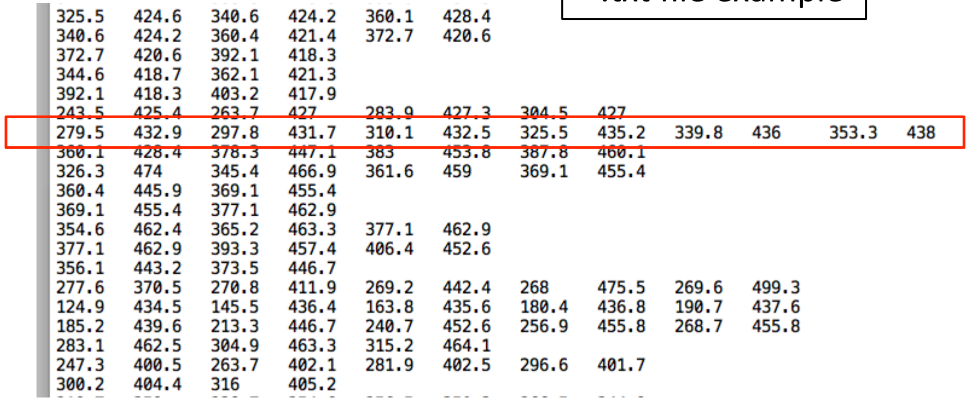

Input

Filename [no file selected]

Input file type

Image file [✓ Node file]

*.txt file example


```
325.5   424.6   340.6   424.2   360.1   428.4
340.6   424.2   360.4   421.4   372.7   420.6
372.7   420.6   392.1   418.3
344.6   418.7   362.1   421.3
392.1   418.3   403.2   417.9
243.5   425.4   263.7   427     283.9   427.3   304.5   427
279.5   432.9   297.8   431.7   310.1   432.5   325.5   435.2   339.8   436     353.3   438
360.1   428.4   378.3   447.1   383     453.8   387.8   460.1
326.3   474     345.4   466.9   361.6   459     369.1   455.4
360.4   445.9   369.1   455.4
369.1   455.4   377.1   462.9
354.6   462.4   365.2   463.3   377.1   462.9
377.1   462.9   393.3   457.4   406.4   452.6
356.1   443.2   373.5   446.7
277.6   370.5   270.8   411.9   269.2   442.4   268     475.5   269.6   499.3
124.9   434.5   145.5   436.4   163.8   435.6   180.4   436.8   190.7   437.6
185.2   439.6   213.3   446.7   240.7   452.6   256.9   455.8   268.7   455.8
283.1   462.5   304.9   463.3   315.2   464.1
247.3   400.5   263.7   402.1   281.9   402.5   296.6   401.7
300.2   404.4   316     405.2
```

Click **Browse…** to select an input file. The file types shown are filtered to those with file extensions *.txt, *.jpeg, *.jpg, *.tiff and *.tif.

If you select a text (*.txt) file of (x,y) fracture trace nodes, the Input file type changes automatically to 'Node file'.

If you choose a graphic file format (*.jpeg, *.jpg, *.tiff, *.tif), the Input file type changes to 'Image file'.

You can manually override these defaults if you wish.

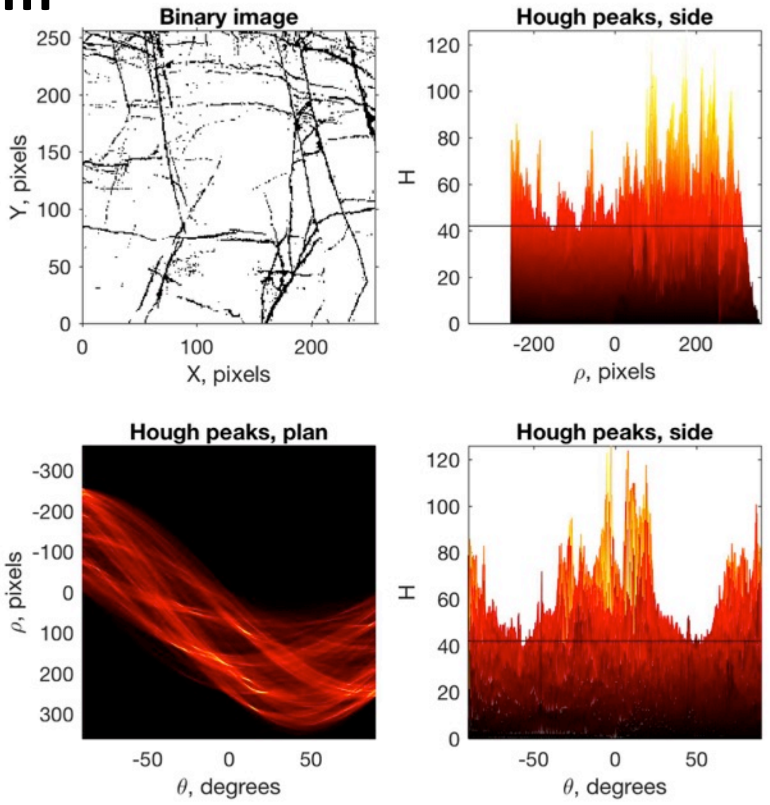
Shown above is an **example of a *.txt file** for input. The file contains (x,y) nodes along each fracture trace. The (x,y) data are **tab-delimited**, and **each fracture trace is on one line**. There is a minimum of 4 columns for each line (i.e. each fracture trace) – (x1, y1) for node 1 and (x2, y2) for node 2.

A fracture trace can be made of many segments; in the example shown above the longest trace (line 7) has 5 segments, bounded by 6 nodes – i.e. 12 columns = 6 x 2 (x,y) pairs.

Image file input – the Hough transform

FracPaQ can also read binary **image files** as input. Two examples are provided, `MacduffBinary.tif` and `OrkneyBinary.tif`. The code uses a Hough transform to find straight lines within the pixels. The user can choose the values at which these lines are merged (if they are close enough together) or discarded (if they are too short). More details on the Hough transform method can be found in the MATLAB Help documentation.

The key parameters that affect the number and length of the fracture traces found are the number of **peaks** and the **threshold**. Using the Hough transform is a 'trial and error' procedure: load an image file and click **Preview**. Then use the graphs (shown on the right, created as Figure 1) and the trace map displayed in the main window to tune the Hough transform parameters. Set the **threshold** to a value between 0.0 and 1.0 that captures only the significant peaks (in orange/yellow on the graphs). 0.33 is a good start.



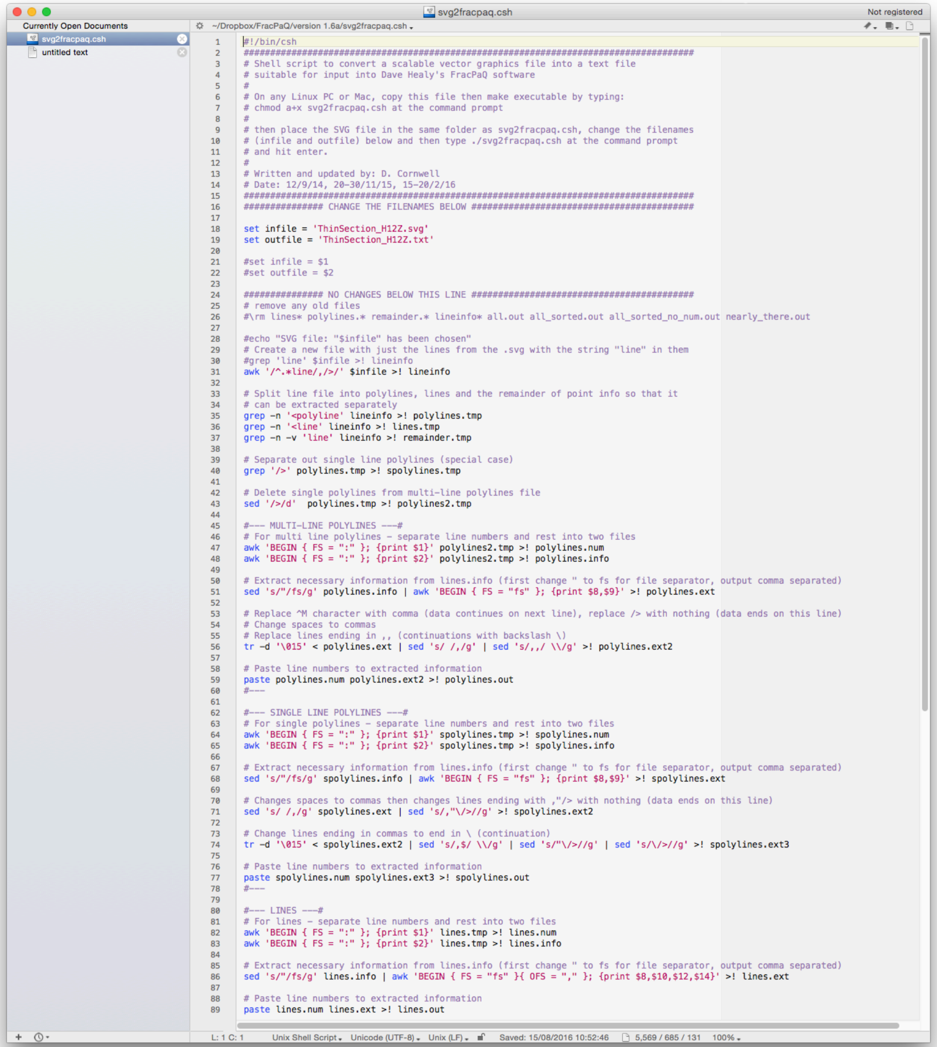
NB: the Hough transform method can **only find straight line fractures** each made of a single trace; multi-segment traces cannot be found. $N_{traces} = N_{segments}$ for this method.

Text file input – conversion from .svg format

Also included with **FracPaQ** is a Unix shell script to convert scaleable vector graphics (*.svg) files into the correct tab-delimited text (*.txt) file format. This file is called **`svg2fracpaq.csh`**.

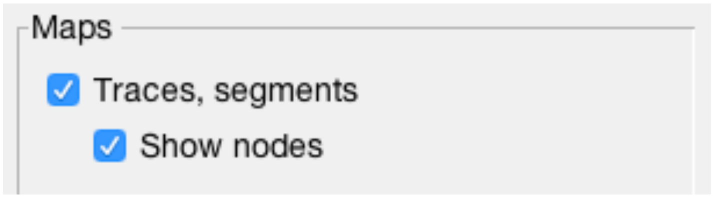
Standard graphics packages, such as Adobe Illustrator, CorelDraw and Inkscape, can all produce *.svg files. Typically, the user imports an image or map of fractured rock, and then traces the fractures onto a new layer in the software. After deleting the original image, the traces can be saved as *.svg format. **We recommend using *.svg format version 1.1.**

The shell script reads in the *.svg file of your choice and produces a new *.txt file with each fracture on a separate line, and all the (x,y) nodes separated by tabs. For Windows users, there may be support for running Unix shell scripts under Windows 10.



```
#!/bin/csh
#####################################################################################
# Shell script to convert a scalable vector graphics file into a text file
# suitable for input into Dave Healy's FracPaQ software
#
# On any Linux PC or Mac, copy this file then make executable by typing:
# chmod a+x svg2fracpaq.csh at the command prompt
#
# then place the SVG file in the same folder as svg2fracpaq.csh, change the filenames
# (infile and outfile) below and then type ./svg2fracpaq.csh at the command prompt
# and hit enter.
#
# Written and updated by: D. Cornwell
# Date: 12/9/14, 20-30/11/15, 15-20/2/16
#####################################################################################
############### CHANGE THE FILENAMES BELOW ##########################################

set infile = 'ThinSection_H12Z.svg'
set outfile = 'ThinSection_H12Z.txt'

#set infile = $1
#set outfile = $2

############### NO CHANGES BELOW THIS LINE ##########################################
# remove any old files
#\rm lines* polylines.* remainder.* lineinfo* all.out all_sorted.out all_sorted_no_num.out nearly_there.out

#echo "SVG file: "$infile" has been chosen"
# Create a new file with just the lines from the .svg with the string "line" in them
#grep 'line' $infile >! lineinfo
awk '/^.*line/,/>/' $infile >! lineinfo

# Split line file into polylines, lines and the remainder of point info so that it
# can be extracted separately
grep -n '<polyline' lineinfo >! polylines.tmp
grep -n '<line' lineinfo >! lines.tmp
grep -n -v 'line' lineinfo >! remainder.tmp

# Separate out single line polylines (special case)
grep '/>' polylines.tmp >! spolylines.tmp

# Delete single polylines from multi-line polylines file
sed '/>/d'  polylines.tmp >! polylines2.tmp

#--- MULTI-LINE POLYLINES ---#
# For multi line polylines - separate line numbers and rest into two files
awk 'BEGIN { FS = ":" }; {print $1}' polylines2.tmp >! polylines.num
awk 'BEGIN { FS = ":" }; {print $2}' polylines2.tmp >! polylines.info

# Extract necessary information from lines.info (first change " to fs for file separator, output comma separated)
sed 's/"/fs/g' polylines.info | awk 'BEGIN { FS = "fs" }; {print $8,$9}' >! polylines.ext

# Replace ^M character with comma (data continues on next line), replace /> with nothing (data ends on this line)
# Change spaces to commas
# Replace lines ending in ,, (continuations with backslash \)
tr -d '\015' < polylines.ext | sed 's/ /,/g' | sed 's/,,/ \\/g' >! polylines.ext2

# Paste line numbers to extracted information
paste polylines.num polylines.ext2 >! polylines.out
#---

#--- SINGLE LINE POLYLINES ---#
# For single polylines - separate line numbers and rest into two files
awk 'BEGIN { FS = ":" }; {print $1}' spolylines.tmp >! spolylines.num
awk 'BEGIN { FS = ":" }; {print $2}' spolylines.tmp >! spolylines.info

# Extract necessary information from lines.info (first change " to fs for file separator, output comma separated)
sed 's/"/fs/g' spolylines.info | awk 'BEGIN { FS = "fs" }; {print $8,$9}' >! spolylines.ext

# Changes spaces to commas then changes lines ending with ,"/> with nothing (data ends on this line)
sed 's/ /,/g' spolylines.ext | sed 's/,"\/>//g' >! spolylines.ext2

# Change lines ending in commas to end in \ (continuation)
tr -d '\015' < spolylines.ext2 | sed 's/,$/ \\/g' | sed 's/"\/>//g' | sed 's/\/>//g' >! spolylines.ext3

# Paste line numbers to extracted information
paste spolylines.num spolylines.ext3 >! spolylines.out
#---

#--- LINES ---#
# For lines - separate line numbers and rest into two files
awk 'BEGIN { FS = ":" }; {print $1}' lines.tmp >! lines.num
awk 'BEGIN { FS = ":" }; {print $2}' lines.tmp >! lines.info

# Extract necessary information from lines.info (first change " to fs for file separator, output comma separated)
sed 's/"/fs/g' lines.info | awk 'BEGIN { FS = "fs" }{ OFS = "," }; {print $8,$10,$12,$14}' >! lines.ext

# Paste line numbers to extracted information
paste lines.num lines.ext >! lines.out
```

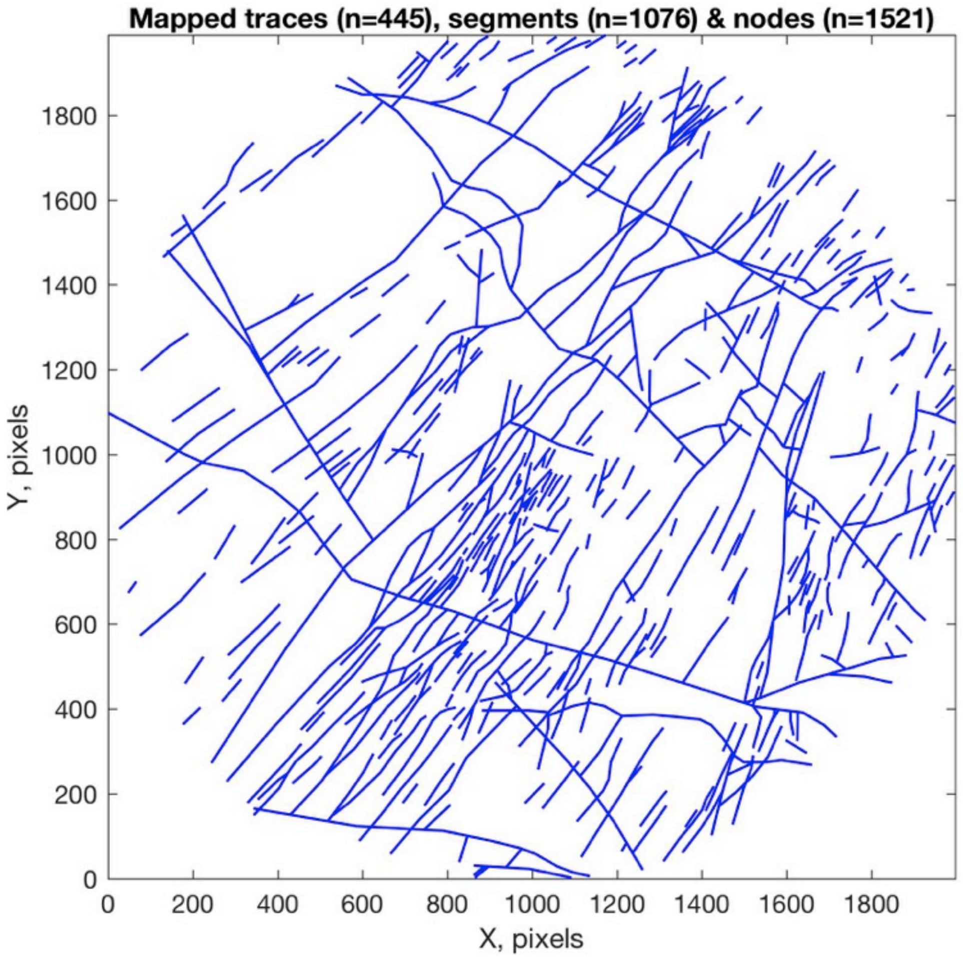
Maps – Traces, segments


Maps
☑ Traces, segments
☑ Show nodes

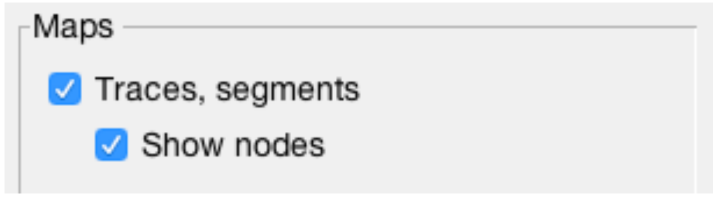
Click on the checkbox "Traces, segments" in the Maps output panel to see a map of all fracture traces and segments. The numbers of traces, segments and nodes (i.e. the end points of traces and segments) are shown in the figure title.

This plot is saved in the current folder as **`FracPaQ2D_tracemap.tif`**, at a default resolution of 300 dpi.

The figure is produced by the script **`guiFracPaQ2Dtracemap.m`**. Edit this script file to change any default settings, such as the plot line colour or the print file resolution.



Maps – Traces, segments; Show nodes

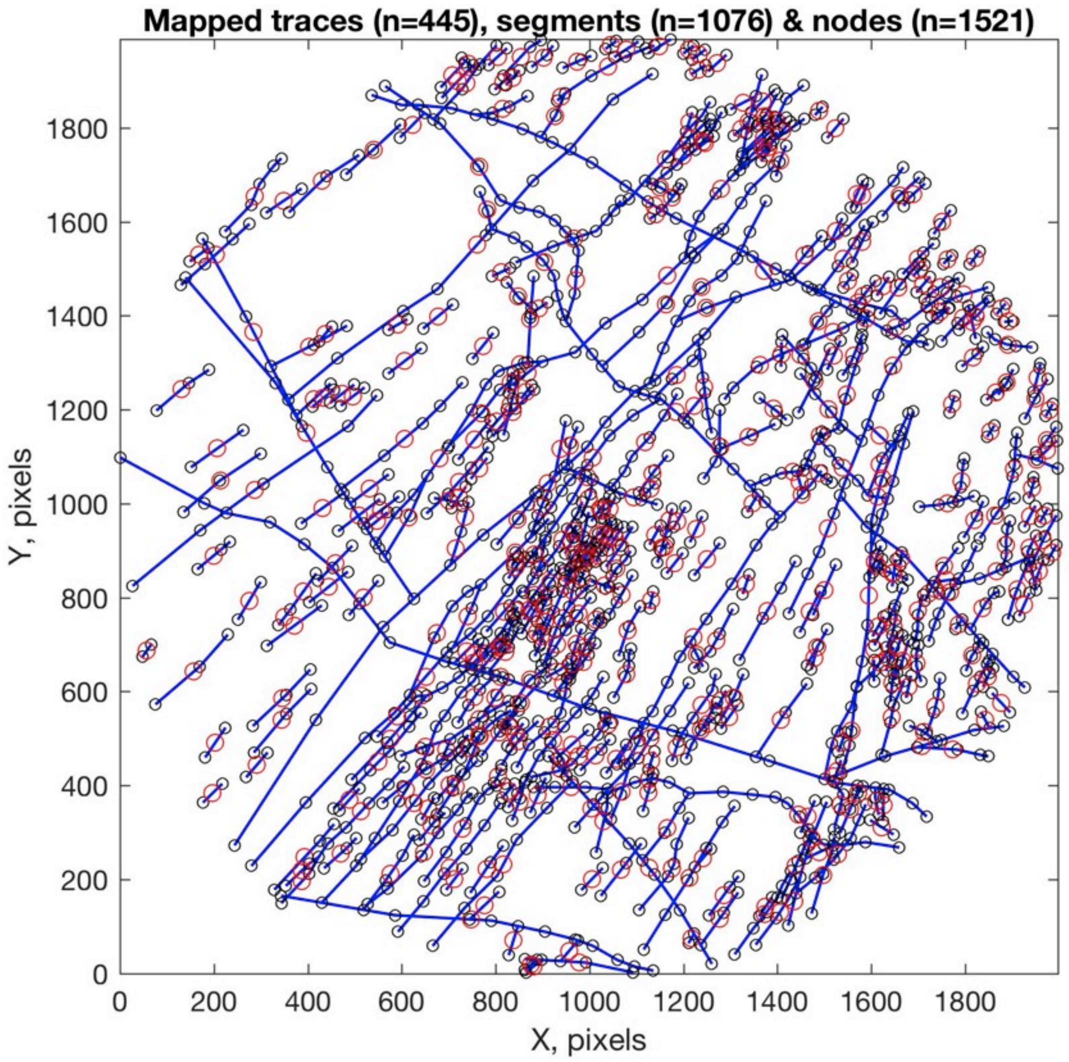

Maps

☑ Traces, segments

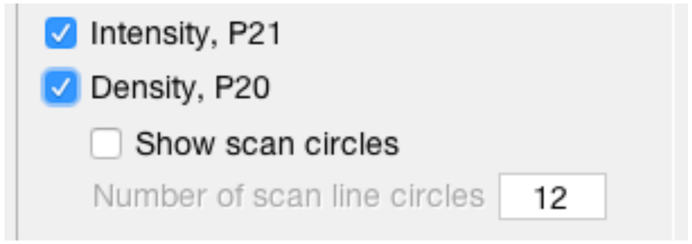
☑ Show nodes

Click on the checkbox "Traces, segments" and also check "Show nodes" to see a map of all fracture traces and segments **with all the nodes displayed**. The numbers of traces, segments and nodes (i.e. the end points of traces and segments) are shown in the figure title. Trace and segment end nodes are shown with black circles, and trace centres are shown with red circles.

This plot is saved in the current folder as **`FracPaQ2D_tracemap.tif`**, at a default resolution of 300 dpi. The figure is produced by the script **`guiFracPaQ2Dtracemap.m`**. Edit this script file to change any default settings, such as the plot line colour or the print file resolution.



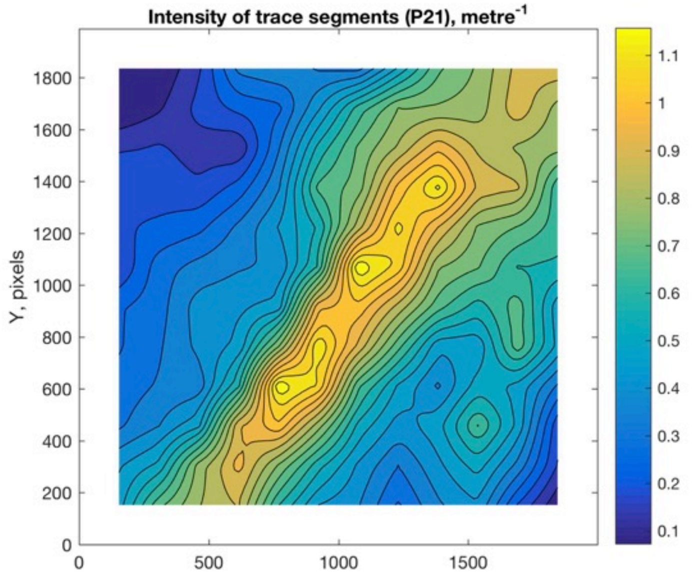
Maps – Intensity (P21), Density (P20)

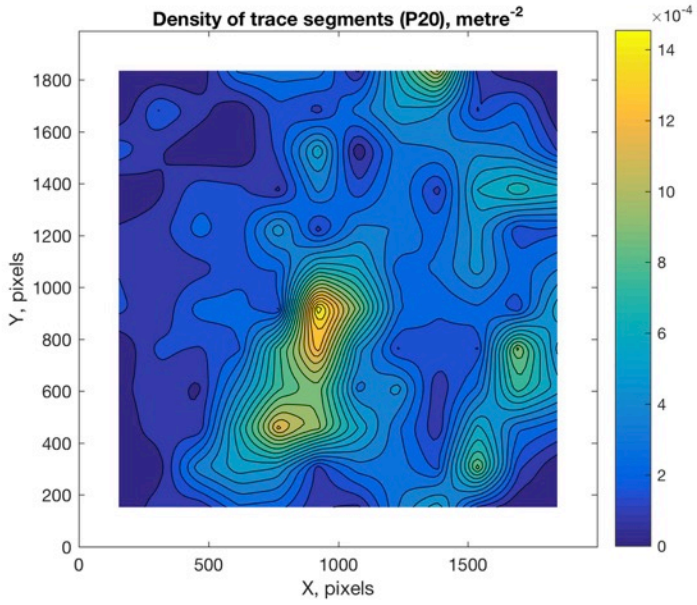

✓ Intensity, P21
✓ Density, P20
☐ Show scan circles
Number of scan line circles 12

Click on the checkbox "Estimated Intensity, P21" and/or "Estimated Density, P20" to see maps of estimated fracture intensity and/or density. These contour maps are produced from the fracture segment data using the circular scan line method of Mauldon et al. (2001).

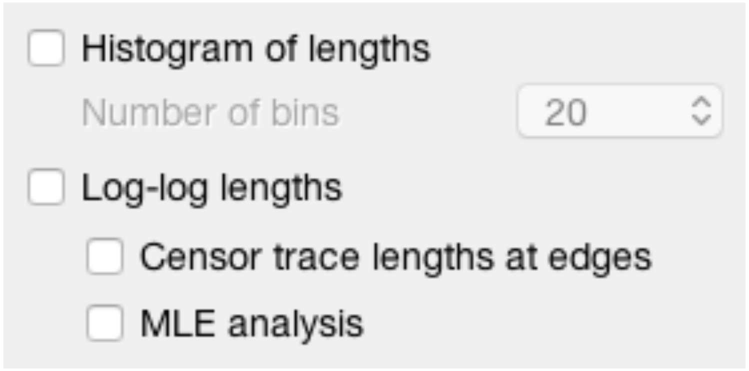
The code places scan circles (using the number in the text box) in the *x*- and *y*-directions, and counts the intersections of fracture segments with the circle perimeter (*n*), and the number of segments that terminate inside the circle (*m*). These measures are used to estimate Intensity (units of L^{-1}) and Density (units of L^{-2}) from the equations in Mauldon et al. (2001). P21 and P20 refer to the measures used by Dershowitz & Herda (1992).

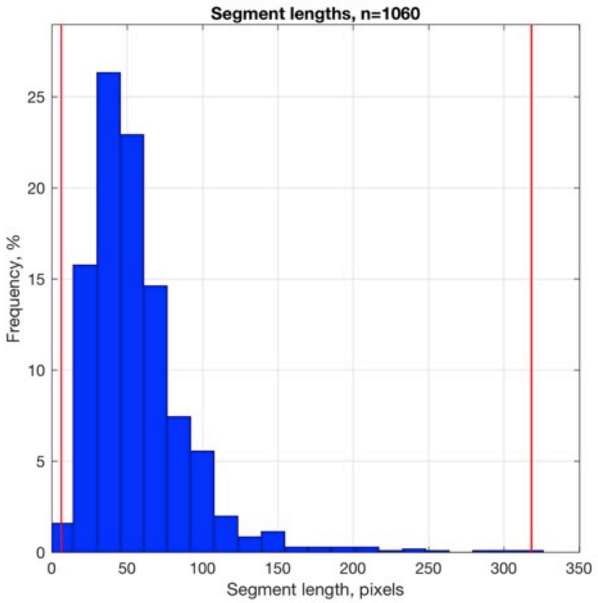
These plots are saved in the current folder as **`FracPaQ2D_intensityP21.tif`** and **`FracPaQ2D_densityP20.tif`**, at a default resolution of 300 dpi. The figure is produced by the script **`guiFracPaQ2Dpattern.m`**. Edit this script file to change any default settings, such as the plot line colour or the print file resolution.

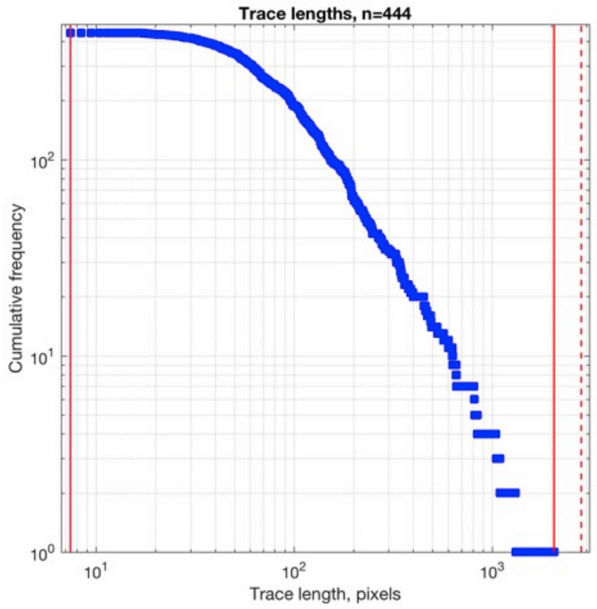




Graphs – Lengths


- [ ] Histogram of lengths
  Number of bins 20
- [ ] Log-log lengths
  - [ ] Censor trace lengths at edges
  - [ ] MLE analysis



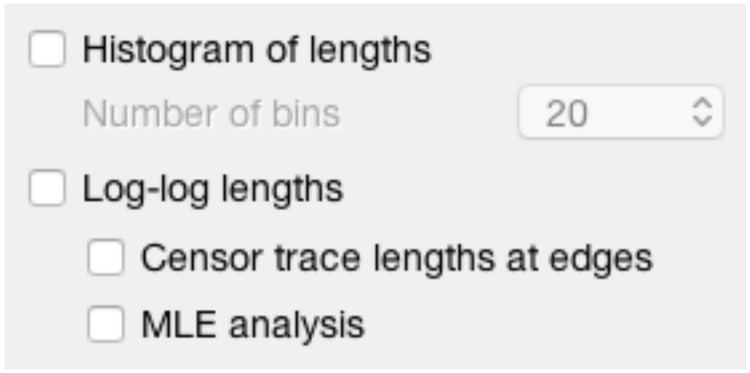


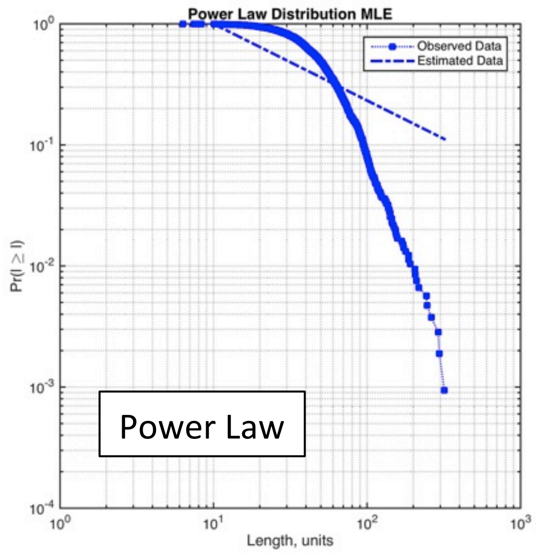
Click on the checkbox “Histogram of lengths” and/or “Log-log lengths” to see a histogram and/or log-log plots of of fracture lengths. **Separate plots are produced for fracture traces** (trace length = sum of segment lengths) **and fracture segments.** Select the number of bins for the histogram from the drop-down list.

Check “Censor trace lengths at edges” to remove from the graphs any fractures with nodes on the edges of the trace map area. The red lines on the graphs show the minimum, maximum and maximum possible (dotted) lengths.

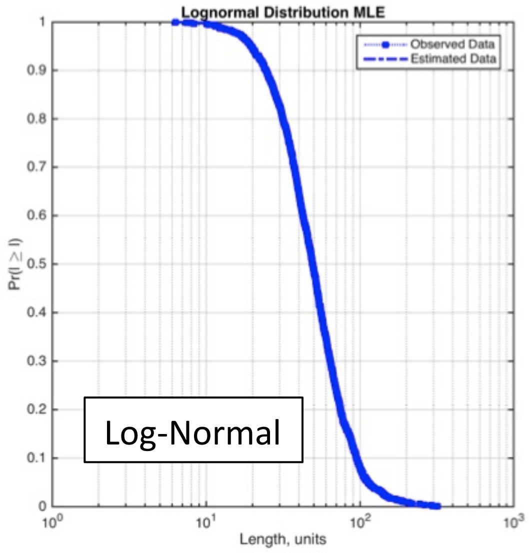
The plots are saved in the current folder as **`FracPaQ2D_*length.tif`**, at a default resolution of 300 dpi. The figures are produced by the script **`guiFracPaQ2Dlength_new.m`**. Edit this script file to change any default settings, such as the plot line colour or the print file resolution.

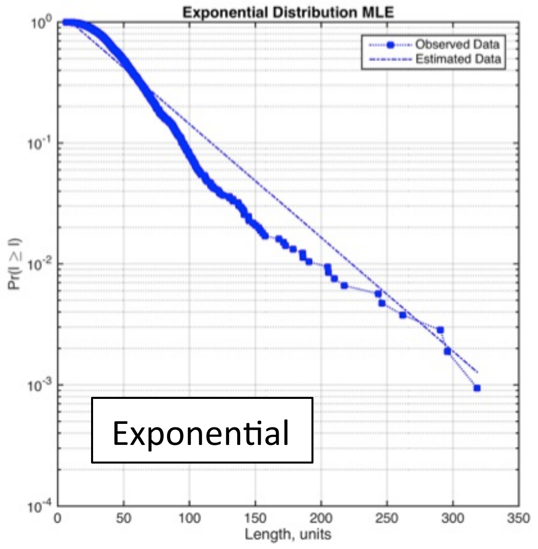
Graphs – MLE Statistical Analysis


☐ Histogram of lengths
Number of bins 20
☐ Log-log lengths
☐ Censor trace lengths at edges
☐ MLE analysis



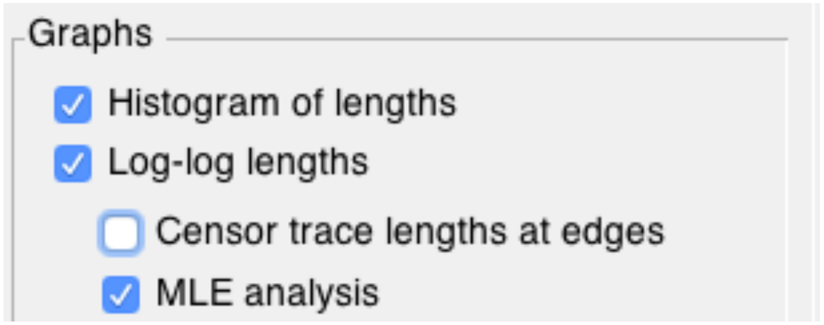
Click on the checkbox “MLE analysis” to perform a Maximum Likelihood Estimation (MLE) for fracture trace and segment lengths. Separate plots are produced for three possible underlying statistical distributions: Power Law, Log-Normal and Exponential. See Clauset et al. (2007) and Rizzo et al. (in review) for further details on MLE.





The plots for fracture segments are saved in the current folder as **`FracPaQ2Dlengths_Fitting*.tif`**, at a default resolution of 300 dpi. The figures are produced by the script **`guiFracPaQ2Dlength_new.m`**. Edit this script file to change any default settings, such as the plot line colour or the print file resolution.

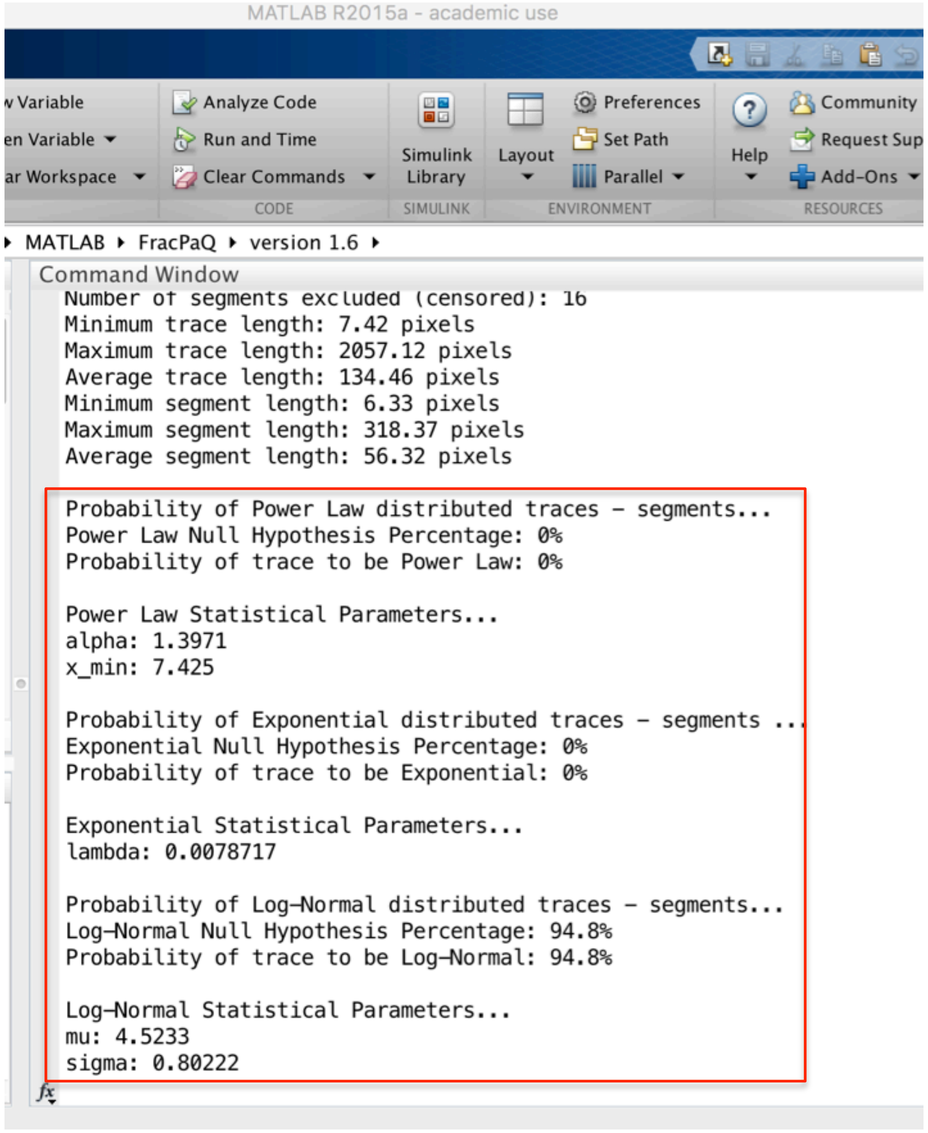
Graphs – MLE Statistical Analysis


Graphs

- [✓] Histogram of lengths
- [✓] Log-log lengths
    - [ ] Censor trace lengths at edges
    - [✓] MLE analysis

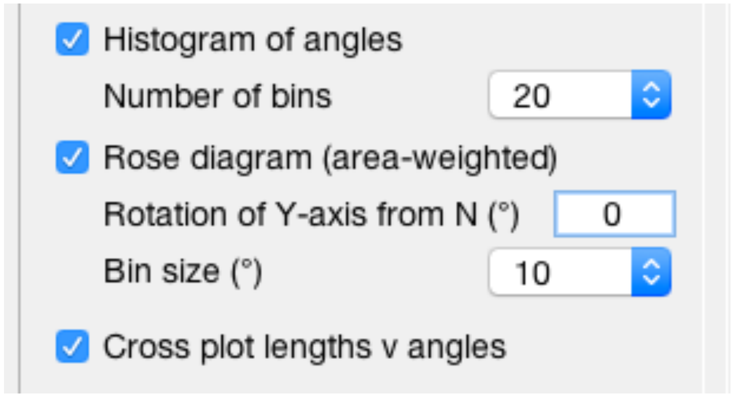
The MLE plots are supplemented with text outputs in the MATLAB™ Command Window. For each of the 3 distributions, the corresponding statistical parameters are given and the probability – expressed in % and calculated based on a Kolmogorov-Smirnoff (KS) test – that the analysed fracture lengths are distributed according to one of the distributions.

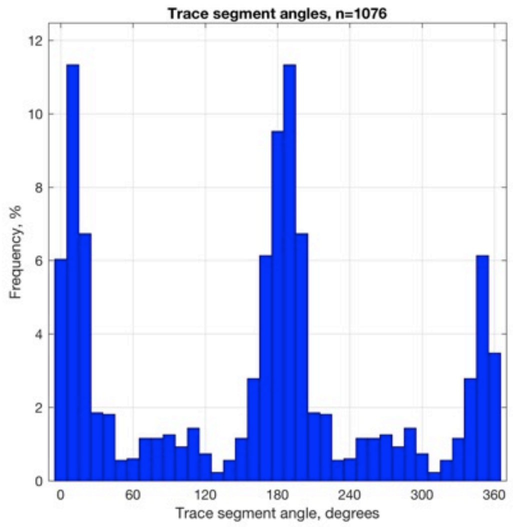
More details and relative references, can be found in the functions **`fitting*.m`**. called by **`guiFracPaQ2Dlength_new.m`**. Edit this script to change the variables 'uc' and 'lc' to cut off a certain percentage from the beginnig ('uc') or the end ('lc') of the data set. Default values are 0% cut-off.

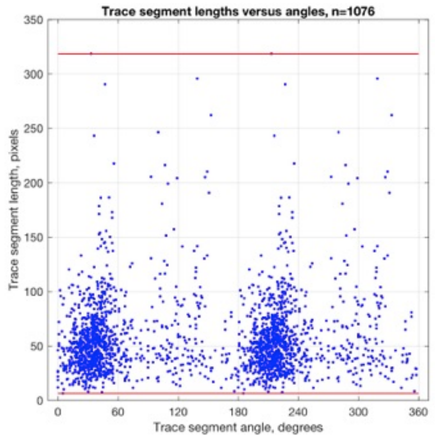


```
Number of segments excluded (censored): 16
Minimum trace length: 7.42 pixels
Maximum trace length: 2057.12 pixels
Average trace length: 134.46 pixels
Minimum segment length: 6.33 pixels
Maximum segment length: 318.37 pixels
Average segment length: 56.32 pixels

Probability of Power Law distributed traces - segments...
Power Law Null Hypothesis Percentage: 0%
Probability of trace to be Power Law: 0%

Power Law Statistical Parameters...
alpha: 1.3971
x_min: 7.425

Probability of Exponential distributed traces - segments ...
Exponential Null Hypothesis Percentage: 0%
Probability of trace to be Exponential: 0%

Exponential Statistical Parameters...
lambda: 0.0078717

Probability of Log-Normal distributed traces - segments...
Log-Normal Null Hypothesis Percentage: 94.8%
Probability of trace to be Log-Normal: 94.8%

Log-Normal Statistical Parameters...
mu: 4.5233
sigma: 0.80222
```

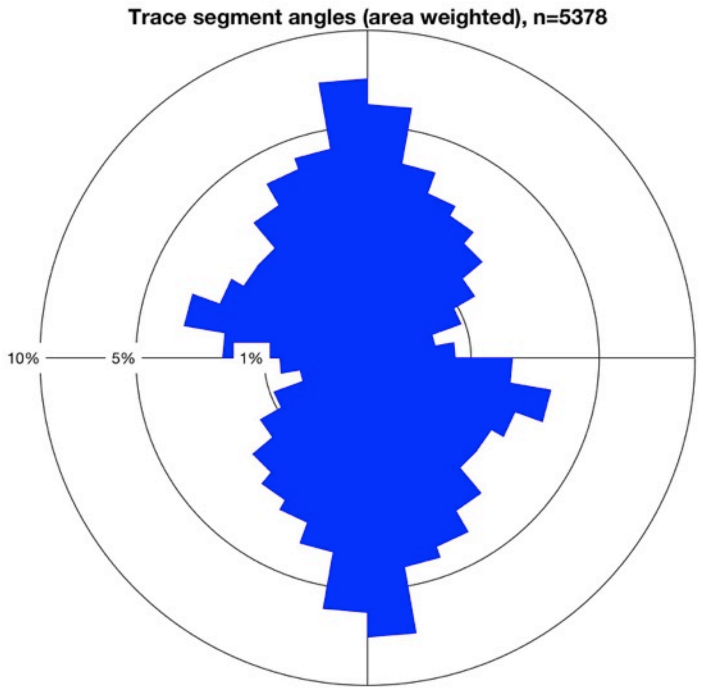
Graphs – Angles (Orientations)


✓ Histogram of angles
Number of bins 20
✓ Rose diagram (area-weighted)
Rotation of Y-axis from N (°) 0
Bin size (°) 10
✓ Cross plot lengths v angles



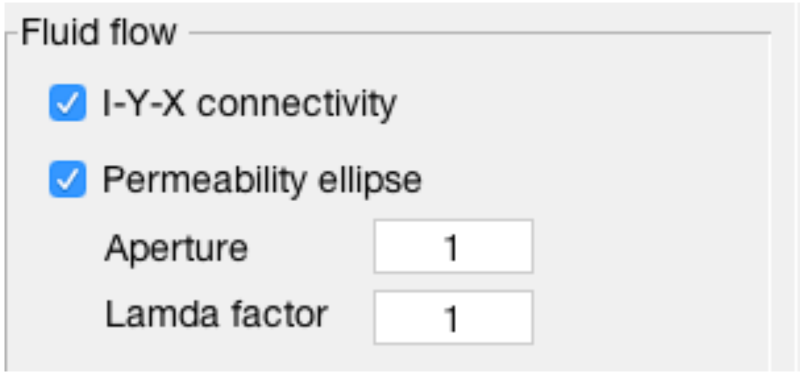


FracPaQ provides 3 plots of orientation data, derived from the angles of the trace segments. Segment angles are calculated with respect to the Y-axis, taking clockwise as positive. To change this in the event that the Y-axis is not aligned due North, enter a number in the text box labelled 'Rotation of Y-axis from N'. This number is then subtracted from each of the segment angles to produce a 'corrected' plot. The number of bins in the histogram and the bin size in the rose diagram can be selected with the dropdown list boxes. The rose diagram is plotted using area-weighting as this provides a more robust measure of orientation distributions in 2D (Nemec, 1988).



The plots are saved in the current folder as **`FracPaQ2D_*angle.tif`**, at a default resolution of 300 dpi. The figures are produced by the script **`guiFracPaQ2Dangle.m`**. Edit this script file to change any default settings, such as the plot line colour or the print file resolution.

Fluid flow – Connectivity


Fluid flow

✓ I-Y-X connectivity

✓ Permeability ellipse

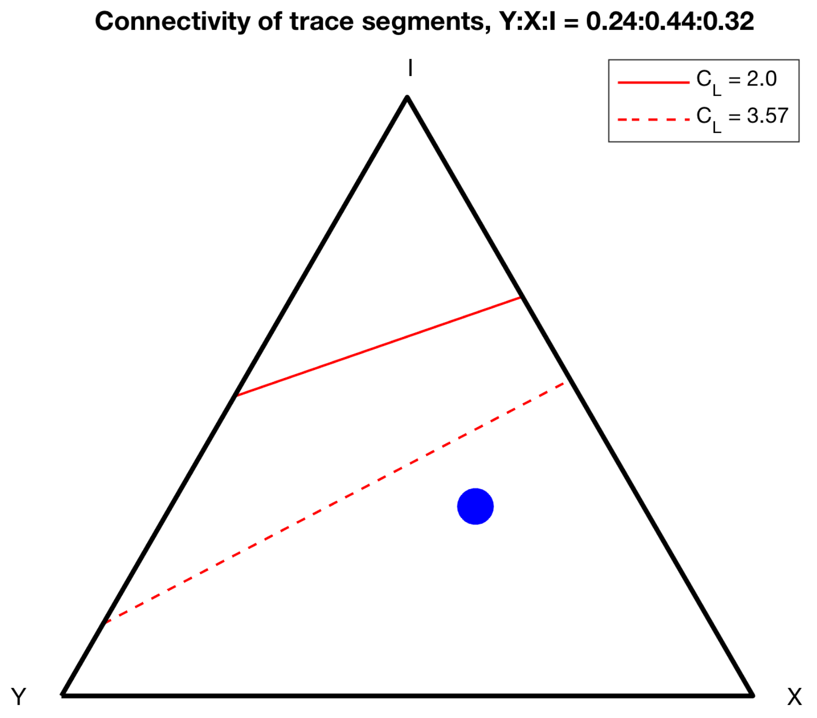
Aperture 1

Lamda factor 1

Click on the checkbox "I-Y-X connectivity" to see a ternary plot of relative proportions of I, Y and X nodes in the fracture network (Manzocchi, 2002). I nodes are the isolated ends of traces, Y nodes are splays or abutments, and X nodes are intersections.

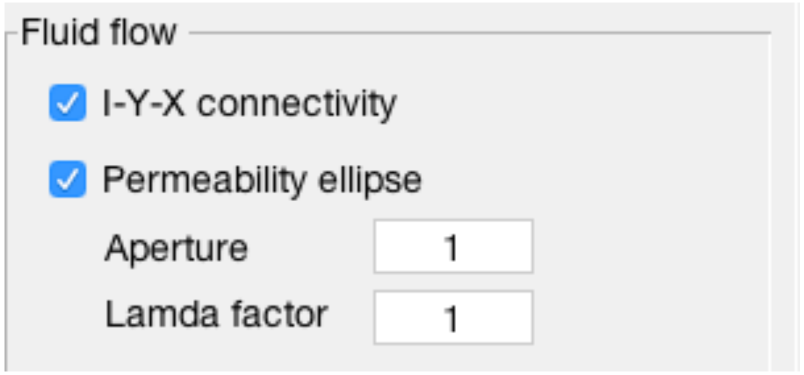
Also shown on this plot are two contour lines for C_L – the number of connections per line (or trace in our terminology) from Sanderson & Nixon (2015). Higher values of C_L are indicative of better connectivity, although the contours depend on other attributes of the fracture network, such as the length distribution and the spatial distribution.

Better connected networks tend to plot towards the bottom of this diagram (higher proportions of Y and X nodes).



The figure is saved in the current folder as **`FracPaQ2D_IYXtriangle.tif`**, at a default resolution of 300 dpi. The figures are produced by the script **`guiFracPaQ2Dpattern.m`**. Edit this script file to change any default settings, such as the colour or the print file resolution.

Fluid flow – Permeability

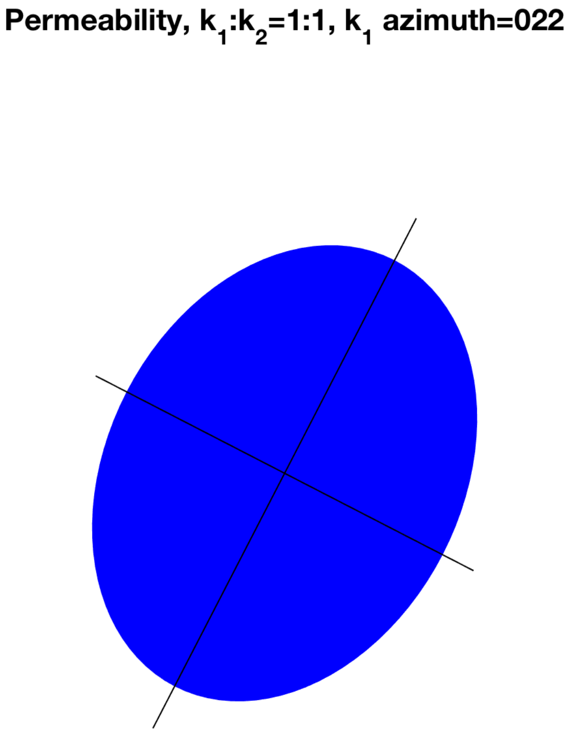


✓ I-Y-X connectivity

✓ Permeability ellipse

Aperture 1

Lamda factor 1

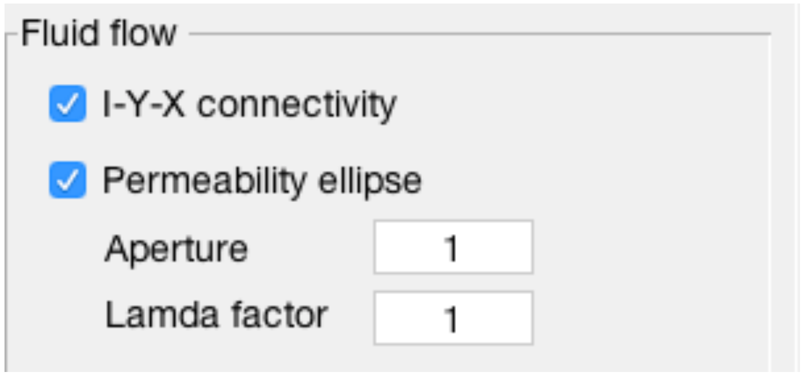


Click on the checkbox "Permeability ellipse" to see the 2D permeability ellipse for the fracture network. This estimate is produced from a combination of lengths, orientations and spatial densities using the tensorial method of Suzuki et al. (1998). The long axis of the ellipse is proportional to the maximum permeability ($\surd \mathbf{k}_1$), and the short axis is proportional to the minimum permeability ($\surd \mathbf{k}_2$) (this is for permeability in the **direction of flow**; see Long et al., 1982 for details).

The user can make changes to the Aperture (default is 1, pixels or metres) and the Lambda factor, which lies between 0 and 1. If all fractures are fully connected, Lambda can be set to 1 (Suzuki et al., 1998). The user can adjust this variable based on the results obtained from the connectivity plot.

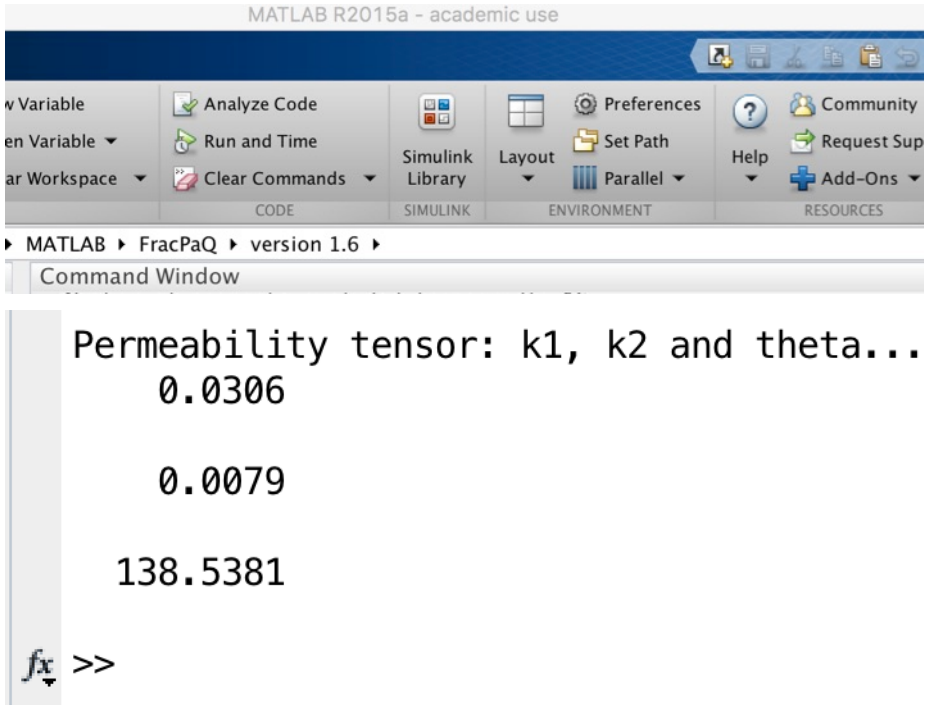
The figure is saved in the current folder as **`FracPaQ2D_permtensor.tif`**, at a default resolution of 300 dpi. The figures are produced by the script **`guiFracPaQ2Dtensor.m`**. Edit this script file to change any default settings, such as the colour or the print file resolution.

Permeability Tensor Ellipse


Fluid flow

- [✓] I-Y-X connectivity
- [✓] Permeability ellipse

Aperture	1
Lamda factor	1



```
Permeability tensor: k1, k2 and theta...
    0.0306

    0.0079

  138.5381

fx >>
```

The permeability ellipse is supplemented with text outputs in the MATLAB™ Command Window.

The numerical values of $\mathbf{k_1}$ (maximum), $\mathbf{k_2}$ (minimum), and the azimuth of $\mathbf{k_1}$ (**theta**, in degrees measured from the Y-axis), are printed in the Command Window.

The units of **k** default to pixels^2, unless a Scaling Factor (pixels/metre) has been entered on the left hand side of the main window, in which case the units are metres^2.

Bibliography

Bonnet, E., Bour, O., Odling, N.E., Davy, P., Main, I., Cowie, P. & Berkowitz, B. (2001). Scaling of fracture systems in geological media. *Reviews of geophysics*, *39*(3), pp.347-383.

Clauset, A., Shalizi, C. R. & Newman, M. E. J. (2009). Power-law distributions in empirical data. *SIAM Review*, *51*(4), 661.

Dershowitz, W.S. & Herda, H.H. (1992). Interpretation of fracture spacing and intensity. In: The 33th US Symposium on Rock Mechanics (USRMS). American Rock Mechanics Association.

Farrell, N.J.C. (2015). Unpublished PhD thesis. University of Aberdeen.

Gillespie, P.A., Howard, C.B., Walsh, J.J. & Watterson, J. (1993). Measurement and characterisation of spatial distributions of fractures. *Tectonophysics*, *226*(1-4), pp.113-141.

Healy, D., Rizzo, R.E., Cornwell, D.C., Farrell, N.J.C., Watkins, H., Timms, N.E., Gomez-Rivas, E. & Smith, M. (in revision). FracPaQ: a MATLAB™ toolbox for the quantification of fracture patterns.

Long, J.C.S., Remer, J.S., Wilson, C.R. & Witherspoon, P.A. (1982). Porous media equivalents for networks of discontinuous fractures. Water Resources Research, 18(3).

Manzocchi, T. (2002). The connectivity of two-dimensional networks of spatially correlated fractures. *Water Resources Research*, *38*(9).

Mauldon, M., Dunne, W.M. & Rohrbaugh, M.B. (2001). Circular scanlines and circular windows: new tools for characterizing the geometry of fracture traces. *Journal of Structural Geology*, *23*(2), pp.247-258.

Nemec, W. (1988). The shape of the rose. *Sedimentary Geology*, *59*(1-2), pp.149-152.

Newman, M. E. J. (2005). Power laws, Pareto distributions and Zipf's law. *Contemporary Physics*, *46*(5), 323–351.

Rizzo, R.E., Healy, D. & De Siena, L. (in revision). The Benefits of a Maximum Likelihood Estimator.

Sanderson, D.J. & Nixon, C.W. (2015). The use of topology in fracture network characterization. *Journal of Structural Geology*, *72*, pp.55-66.

Smith, M. (2016). Unpublished MSc thesis. University of Aberdeen.

Suzuki, K., Oda, M., Yamazaki, M. & Kuwahara, T. (1998). Permeability changes in granite with crack growth during immersion in hot water. *International Journal of Rock Mechanics and Mining Sciences*, *35*(7), pp.907-921.

Watkins, H. (2015). Unpublished PhD thesis. University of Aberdeen.

Acknowledgements

`lineSegmentIntersect.m` written by **U. Murat Erdem**; available on Mathworks FileExchange:
http://uk.mathworks.com/matlabcentral/fileexchange/27205-fast-line-segment-intersection

`readtext.m` written by **Peder Axensten**; available on Mathworks FileExchange:
http://uk.mathworks.com/matlabcentral/fileexchange/10946-readtext

We thank **Paul Gillespie** (Statoil), **Mark Fischer** (Northern Illinois University) and **Bill Dunne** (University of Tennessee; and Editor of the Journal of Structural Geology) for comments on our submitted manuscript, which have led to major improvements in the code, the paper and the User Guide.