FracPaQ

Fracture Pattern Quantification

User Guide

Version 2.2, February 2018

David Healy & Roberto Rizzo

d.healy@abdn.ac.uk

With contributions from Nick Timms, Matthew Belshaw & Daniel Knight

Contents

1. Summary of changes in FracPaQ version 2.2

In this new release, we have **added**:

- 2D wavelet analysis of fracture maps to determine scale transitions in fracture orientations;
- a new rose diagram with segment orientations colour-coded by strike, matching the segment strike map;
- a new (semi-)variogram plot of segment lengths using their calculated centroids;
- 2 new command line utilities to help convert image files into 8-bit binary format and to convert trace maps from Move™ into **FracPaQ** format.

We have **modified**:

- the MLE plots of fracture lengths to fix reported bugs;
- this User Guide to provide more detail on preparing image files for input and details of the new features listed above.

2. Introduction

FracPaQ is an open source toolbox written in MATLAB™ and publicly available on GitHub and the Mathworks™ FileExchange. **FracPaQ** is designed to quantify fracture patterns in rock or other materials from 2D images. The user supplies either an image file of fractured material, or a text file of traced fractures and their (x,y) coordinates. From either kind of input, the code calculates fracture lengths, angles and connectivity. These are displayed as maps and graphs, and saved as *.tif files. Estimates are made of fracture intensity (P21) & density (P20), and permeability in 2D using a simple parallel-plate model.

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, **readtext.m** by Peder Axensten, **cmocean.m** by Kristen Thyng, **variogram.m** by Wolfgang Schwangart (all available on Mathworks FileExchange), and various wavelet routines by Roseanna Neupauer and Kaye Powell. The handling of image file input in **FracPaQ** also uses functions from the Image Processing Toolbox (version 9.4), a MATLAB™ add-on.

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 ☺**

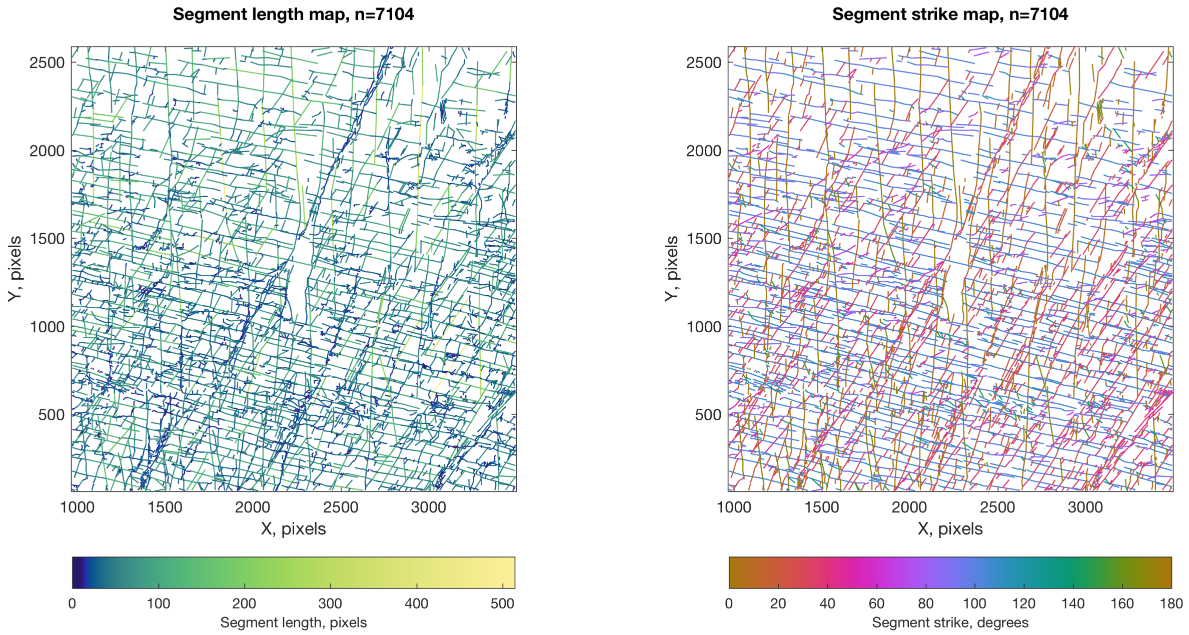


Figure 1. Example of fracture trace maps from **FracPaQ** version 2.2. Left side shows a map with segments coloured by length and the right side shows a map with segments coloured by strike (assuming North is parallel to the y-axis).

3. System requirements

FracPaQ runs on any computer with MATLAB™ installed – Microsoft Windows™, Mac OS X™ or various flavours of Linux. The current version of **FracPaQ** is 2.2 (February 2018). We developed **FracPaQ** version 2.2 using MATLAB™ version 9, R2016a.

4. Obtaining & installing the code

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

Step-by-step guide:

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™.
4. Set the current working folder in MATLAB™ to the same folder you installed the code in.
5. At the MATLAB™ command prompt type 'guiFracPaQ2D' and hit Enter.

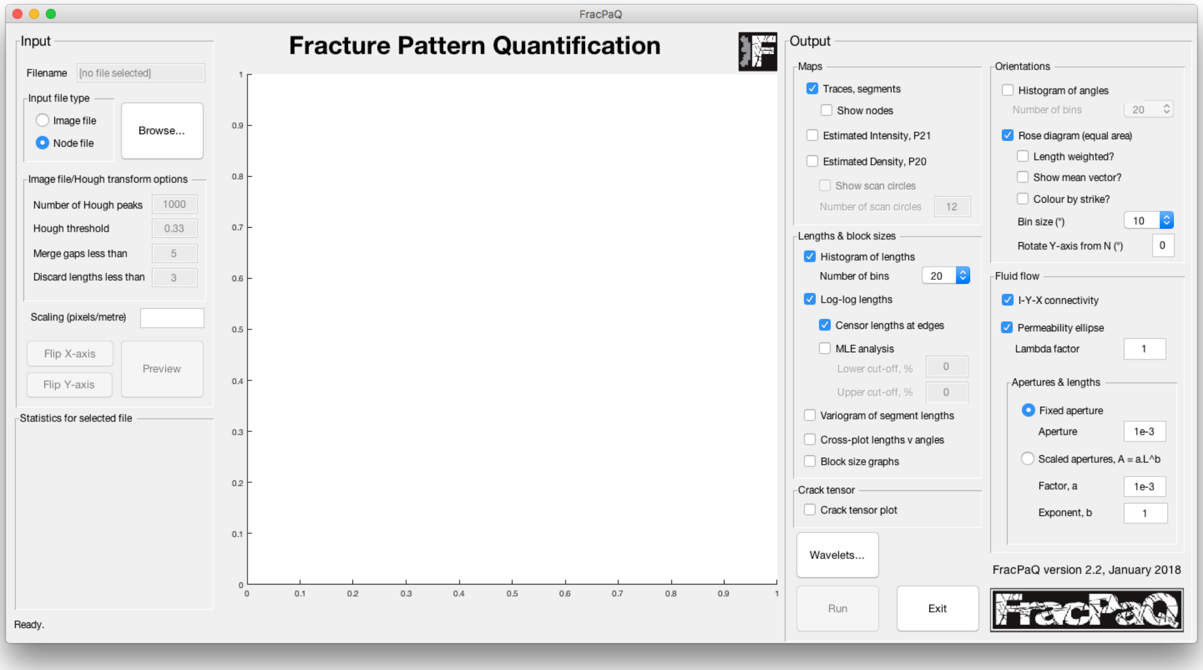
The output files (i.e. the graphs, maps and data) will appear as *.tif and *.txt files in the same folder as the code.

5. Starting FracPaQ

There is one main window in the **FracPaQ** application (see below). The window can be minimized 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.

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 text box).



FracPaQ

Fracture Pattern Quantification

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

Ready.

Output

Maps
☑ Traces, segments
□ Show nodes
□ Estimated Intensity, P21
□ Estimated Density, P20
□ Show scan circles
Number of scan circles 12

Lengths & block sizes
☑ Histogram of lengths
Number of bins 20
☑ Log-log lengths
☑ Censor lengths at edges
□ MLE analysis
Lower cut-off, % 0
Upper cut-off, % 0
□ Variogram of segment lengths
□ Cross-plot lengths v angles
□ Block size graphs

Crack tensor
□ Crack tensor plot

Orientations
□ Histogram of angles
Number of bins 20
☑ Rose diagram (equal area)
□ Length weighted?
□ Show mean vector?
□ Colour by strike?
Bin size (°) 10
Rotate Y-axis from N (°) 0

Fluid flow
☑ I-Y-X connectivity
☑ Permeability ellipse
Lambda factor 1

Apertures & lengths
● Fixed aperture
Aperture 1e-3
○ Scaled apertures, A = a.L^b
Factor, a 1e-3
Exponent, b 1

FracPaQ version 2.2, January 2018

Figure 2. Screenshot of the graphical user interface (GUI) in FracPaQ version 2.2. This is the only input window. Each output (map, graph etc.) is directed to a separate MATLAB™ figure window and saved to a separate graphics file in the current folder.

Select the Outputs you need (right hand side), and then click **Run** to produce the selected maps and graphs. Each output map or plot is shown in a separate figure window, and saved as a separate .tif image file in the current folder.

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

6. Choosing an input file

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

6.1 Image file input

If you choose a graphic file format (*.jpeg, *.jpg, *.tiff, *.tif), the Input file type changes to 'Image file'. This file will then be processed by the Hough transform method to build fracture traces (see below). **NB input image files must be 8-bit and binary** (i.e. grayscale). You can check the format of your any potential image file before loading into FracPaQ using the MATLAB™ function **imfinfo()**. At the MATLAB™ command line, type 'imfinfo('this_is_my_file.tif')' and hit Enter. MATLAB™ will display a range of file format information. Check out the fields for BitDepth and ColorType. BitDepth should be 8 and ColorType needs to be 'grayscale'. You can modify existing image files - e.g. photographs of outcrops taken on digital cameras using your own preferred software (e.g. Adobe Photoshop™ or ImageJ), or you can use our new command line utility, **FracPaQ2D_imagePreprocess.m** (see section 13).

Image files are processed by a Hough transform method which looks for co-linear pixels and builds straight line fracture traces. With this method each fracture has one trace and only one segment per trace. 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 numbers and lengths of fracture traces 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 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.

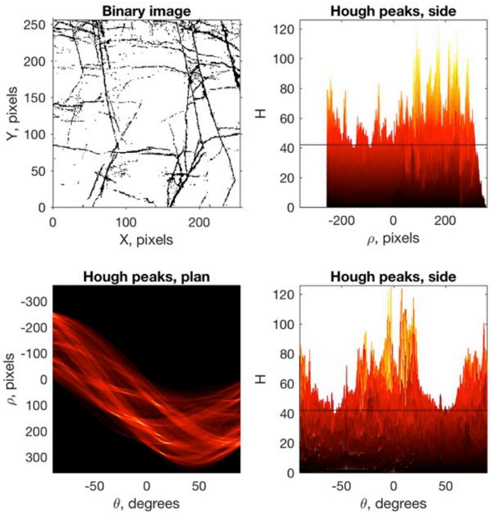
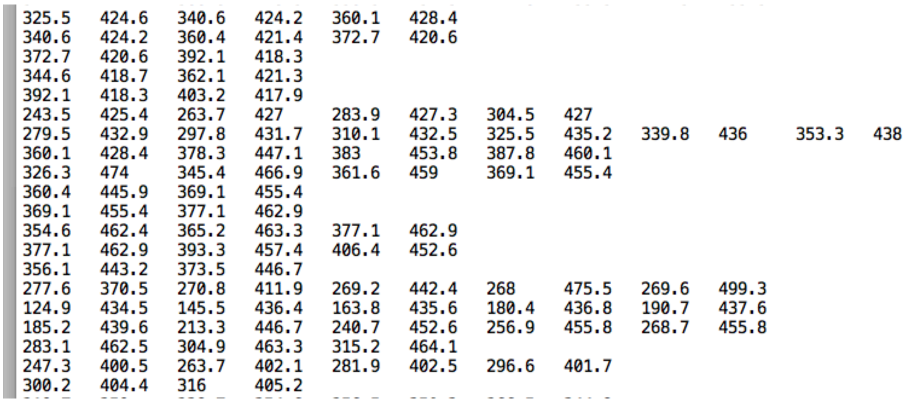


Figure 3. Output from Hough transform applied to the sample image input file MacduffBinary.tif. Set the **Threshold** parameter, between 0.0 and 1.0, to a value that just captures the maxima - i.e. the yellow/orange peaks. Values of 0.3-0.4 often work well.

6.2 Node file input

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



```
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
```

Figure 4. Example of a .txt node file. Tab-delimited (x,y) pairs, one fracture per line.

A node file contains (x,y) coordinates of 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.

7. Building a node input file in .svg format

As noted above, one option for the Node file input is to use a file format of .svg – or Scalable Vector Graphics (an open standard based on XML). Graphics packages like Adobe Illustrator™ and Corel Draw™ (and many others) can produce .svg files from images of fracture traces.

Open your chosen graphics package and import your image of fractures – this could be a photograph from an outcrop or thin section, satellite or aerial imagery from Google™ or Bing™, or a scan of a geological map. Before spending time tracing fractures it makes sense to assess the appropriate level of magnification to use. Choose a sub-area of your fracture image and increase the magnification (zoom in) until the fracture traces pixelate. Typically, we have found that the best compromise between resolution and time spent tracing is around 4-5 levels of zoom below the maximum (i.e. the level at which the fractures pixelate). However, each project is different and you must decide on your own preferences. In selecting a sub-area from an imported image of fractured rock it is best to avoid areas with significant amounts of vegetation cover or scree etc.

1. Import the base image containing the 'raw' fracture traces.
2. Select an area of the image and draw a bounding rectangle (Rectangle tool).

3. Create a new layer (see Figure 5 for Adobe Illustrator™). Lock the base layer.
4. Use the Pen tool (Adobe Illustrator™) or Polyline tool (Corel Draw™) to trace fractures on this new layer. Click along the fracture trace at intervals where the orientation changes. Each click defines a new *node*, and the nodes define the single straight-line *segments* that make up a fracture *trace*. Take care not to drag the cursor when clicking along a trace. When you reach the end of a trace click Enter. Ensure 'Snap to Grid' (in the View menu) is turned off, to ensure optimum accuracy of fracture node positions.

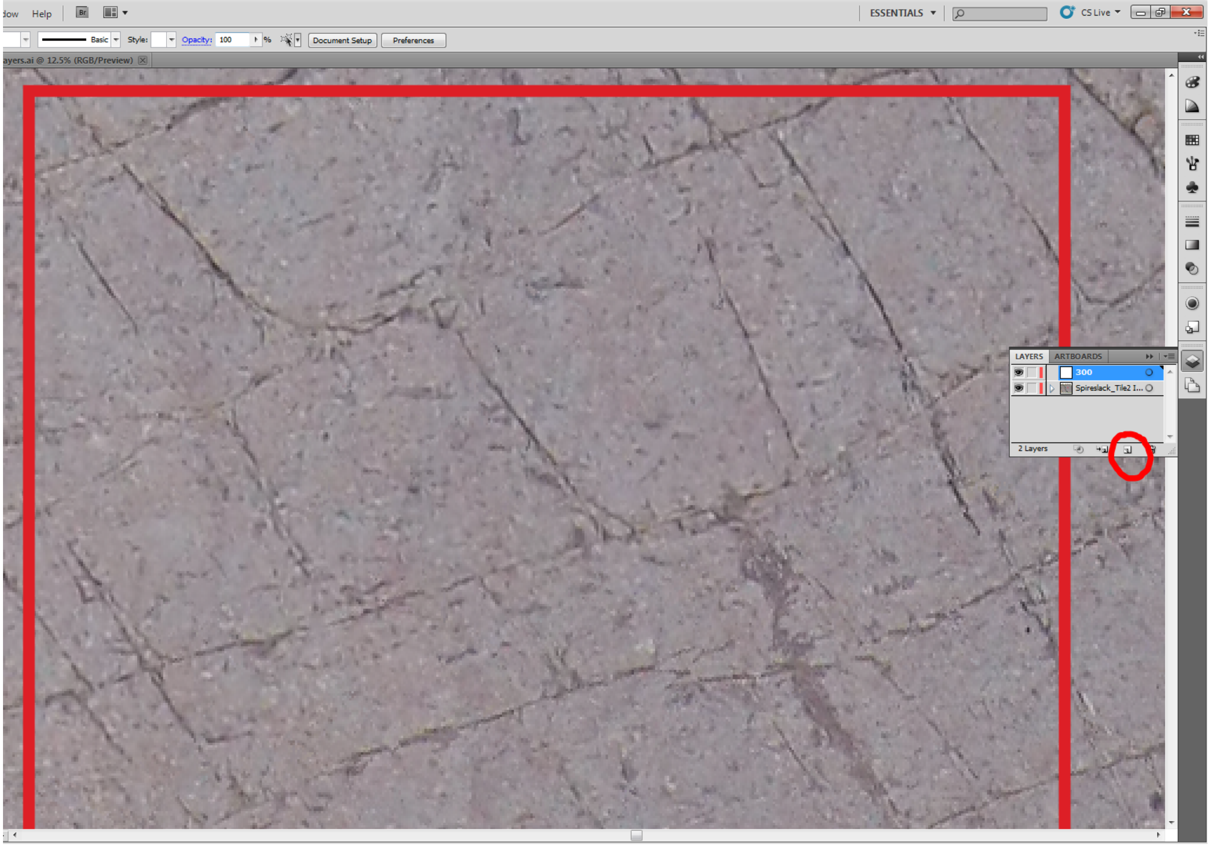


Figure 5. Screenshot of Adobe Illustrator™ showing how to create a new layer on top of an imported base image of fractured rock.

5. Trace all visible fractures at the selected magnification; do not zoom in and out as this can create a sampling bias. Take care not to accidentally join (or merge) two distinct fracture traces, as **FracPaQ** will read this as a single fracture trace.
6. If you want to separate distinct sets of fracture traces (e.g. by mineral fill, inferred age, orientation, or some other criterion) then you can add a new layer and choose a new Line and Fill colour. Then use the Pen or Polyline tool to trace this fracture set on the new layer, with as many layers as there are fracture sets.
7. Save the file in .ai format (Adobe Illustrator™) or .cdr format (Corel Draw™).
8. Delete the layer containing the base level image.
9. [If the area traced is in the top left corner of the document page then skip this step.] Unlock all layers; select all fractures (Ctrl+A then Ctrl+C) using the Selection Tool. Drag all the fractures to the top left corner of the document.

10. Using ‘Save As…’ save the file in .svg format, with the following options: SVG version 1.1 and CSS Properties = ‘Presentation Attributes’ (Adobe Illustrator™) or Styling Options = ‘Presentation Attributes’ (Corel Draw™).

8. Scaling

The default unit of length in **FracPaQ** is the pixel. For any input file, the x- and y-coordinates on the maps and all length data are reported in pixels. However, if you wish to change this to SI units of length (i.e. metres) enter a scaling factor in the ‘Scaling’ text box. For example, if you used a binary image photograph as input with a width of 100 pixels, and you know the true field of view is 1.0 metre, then enter 100.0 in the ‘Scaling’ text box before you click Run to generate output maps or plots.

9. North correction

By default, **FracPaQ** assumes that fracture orientations (i.e. their strikes) are measured clockwise from the positive Y-axis, assumed to be pointing North. If this is not the case, you can enter an angle (0-180°) between the Y-axis and true North for the mapped data in the text box marked ‘North correction for Y-axis, °’. This angle is then added to all orientations as plotted in the selected outputs, including the rose plot, the segment strike map, the crack tensor plot and the permeability ellipses. If you want to remove an angle then enter a negative value (< −180°) in the text box.

10. Outputs

10.1. Maps – Traces, segments

Tick the check box 'Traces, segments' in the Maps section of Outputs to generate four maps of fracture traces and segments. These are: the basic single-colour map of all traces and segments; a map with traces colour-coded by trace length; a map with segments colour-coded by segment length; and a map with segments colour-coded by segment strike (NB if you have entered a non-zero value in the text box 'North correction for Y-axis, °', then this angle is added to these colour coded strike values, see below).

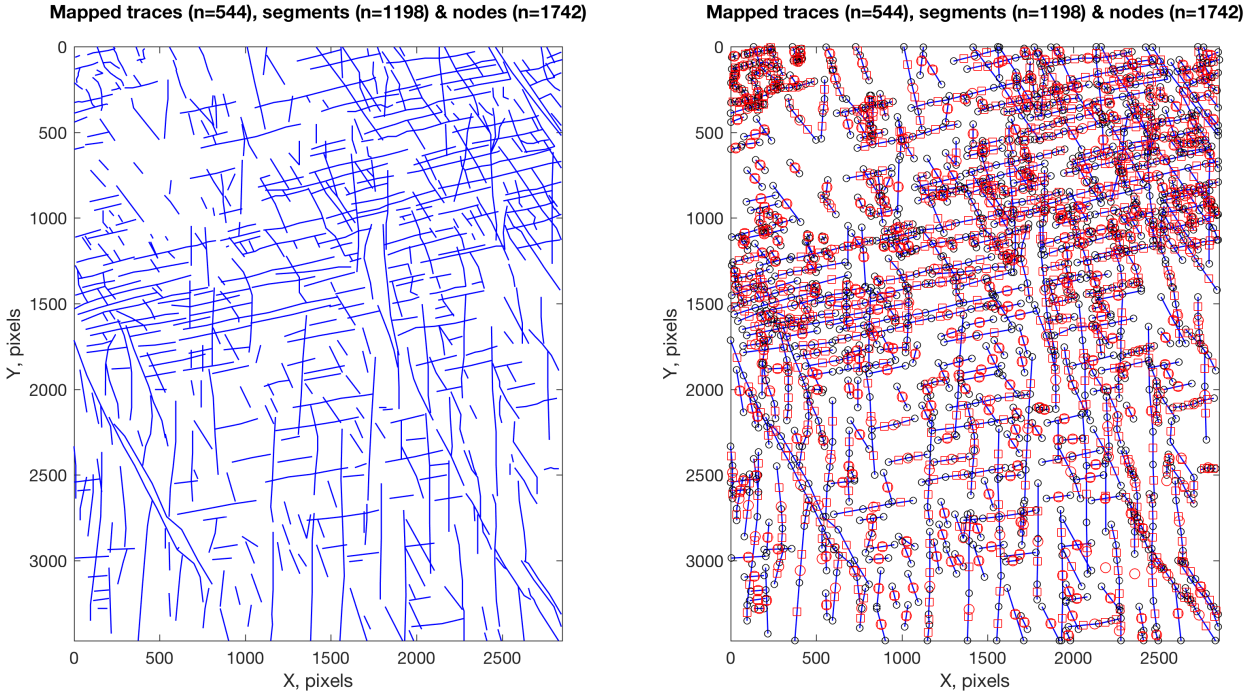


Figure 6. Left side shows the basic trace map, and right side shows the trace map with 'Show nodes' option checked. Nodes are shown with black circles, segment mid-points with red squares and trace mid-points with red circles.

Maps – Traces, segments; Show nodes

FracPaQ is based on coordinate geometry in 2D. Therefore, when the traces and segments are loaded into the application, we also calculate the mid-points (or centroids) of each trace and each segment. To display these data on a trace map, simply check the 'Show nodes' check box and click Run. The mid-points, together with the nodes located at the ends of all traces and segments, are displayed on the map. The node and centroid data are stored in the underlying traces data structure (see guiFracPaQ2D_list.m), and may be useful for new maps and plots exploring the spatial statistics of fracture patterns.

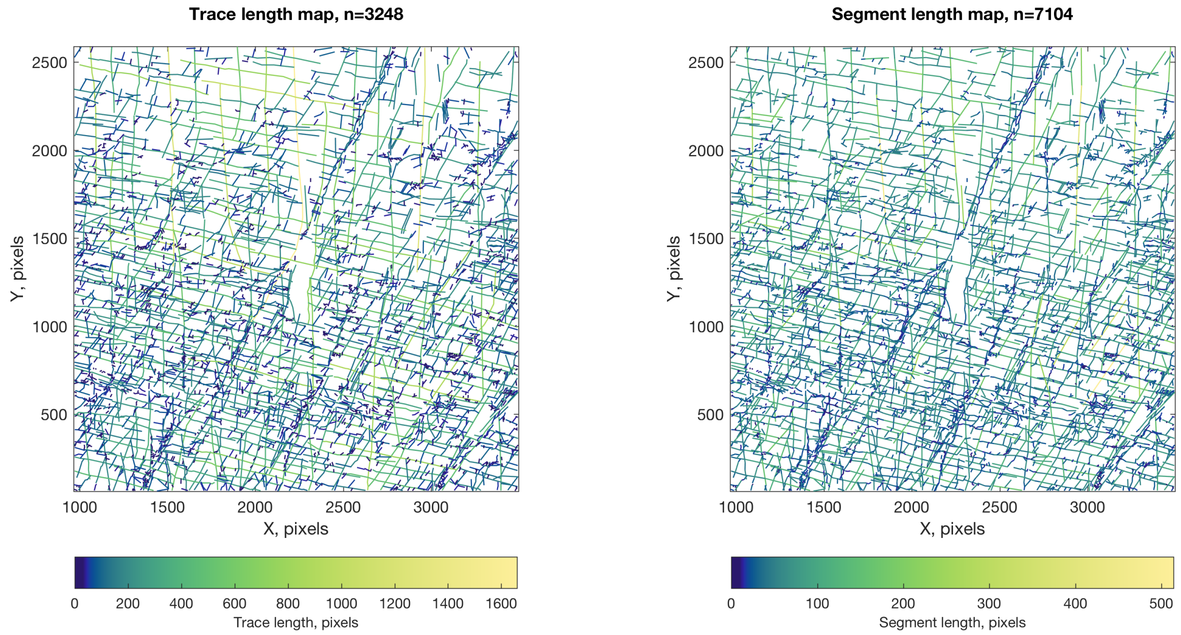


Figure 7. Left side shows the trace map, with *traces* coloured by length. Right side shows the trace map with *segments* coloured by length.

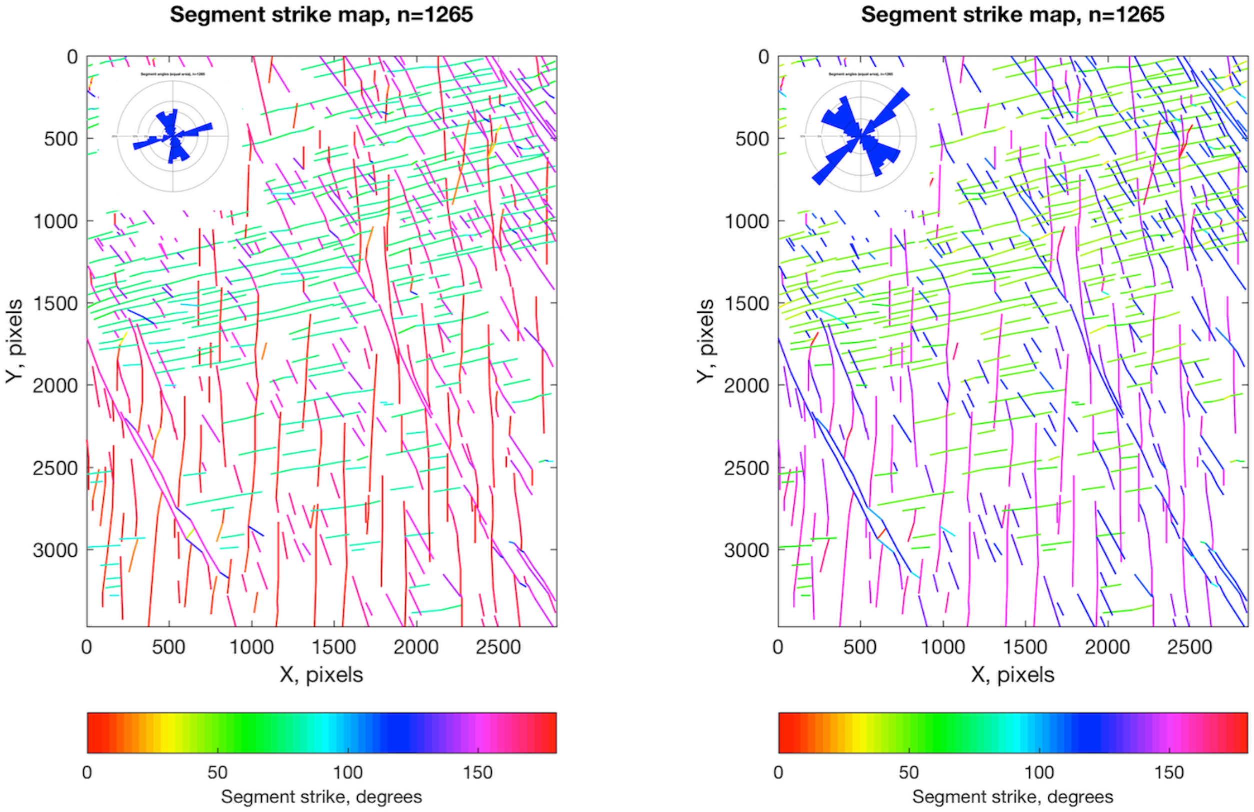


Figure 8. Left side shows a trace map, with *segments* coloured by strike. Right side shows the same map data, but now with a North correction of −30° applied. Insets show the equal area rose plots superimposed.

Maps – Estimated Intensity, P21

FracPaQ uses the circular scan window method of Mauldon et al. (2001) to calculate an estimate of fracture intensity (units of L^{-1}). Click on the check box 'Estimated Intensity, P21' to generate this map. Estimates of intensity are made on a regular grid pattern using the number of scan circles specified in the text box 'Number of scan circles'. If the map area is not square, then this value is the *minimum* number of scan circles used along the shorter side of the area.

Maps – Estimated Density, P20

FracPaQ uses the circular scan window method of Mauldon et al. (2001) to calculate an estimate of fracture intensity (units of L^{-2}). Click on the check box 'Estimated Density, P20' to generate this map. Estimates of density are made on a regular grid pattern using the number of scan circles specified in the text box 'Number of scan circles'. If the map area is not square, then this value is the *minimum* number of scan circles used along the shorter side of the area.

Maps – Estimated Intensity or Density; Show scan circles

Click on the check box 'Show scan circles' to generate a trace map showing the location and size of the scan circles used to estimate Intensity or Density.

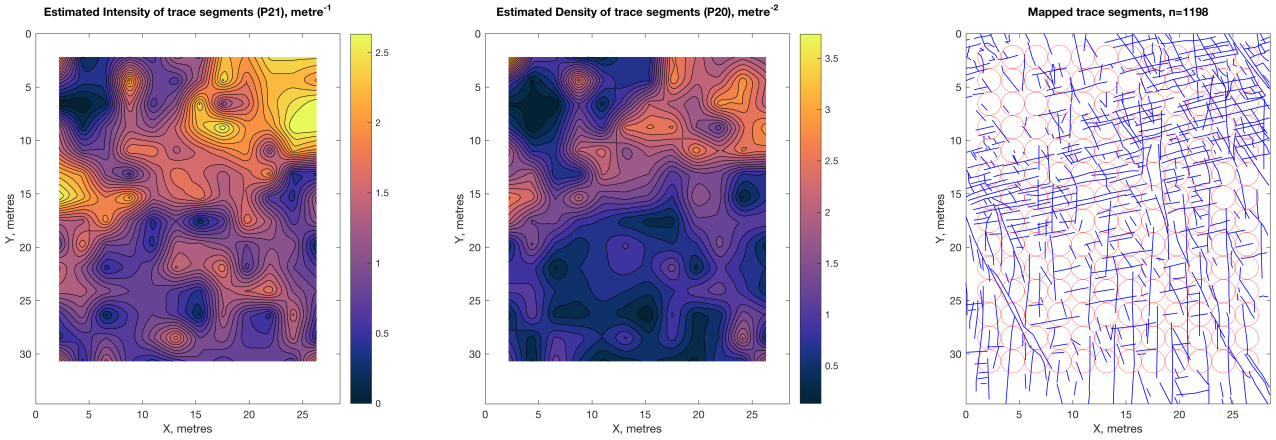


Figure 9. Left side shows a map of Estimated Intensity (P21, m^{-1}). Centre shows a map of Estimated Density (P20, m^{-2}). Right side shows the positions and sizes of the scan circles used in the estimations.

10.2. Lengths & sizes – Histogram of lengths

Tick the check box 'Histogram of lengths' to generate two frequency-size plots for trace lengths and segments lengths. You can change the number of bins used in these histograms by picking a different value in the dropdown list box 'Number of bins' (default is 20). Lengths are reported in pixels, unless a non-zero value is entered in the text box 'Scaling (pixels/metre)'.

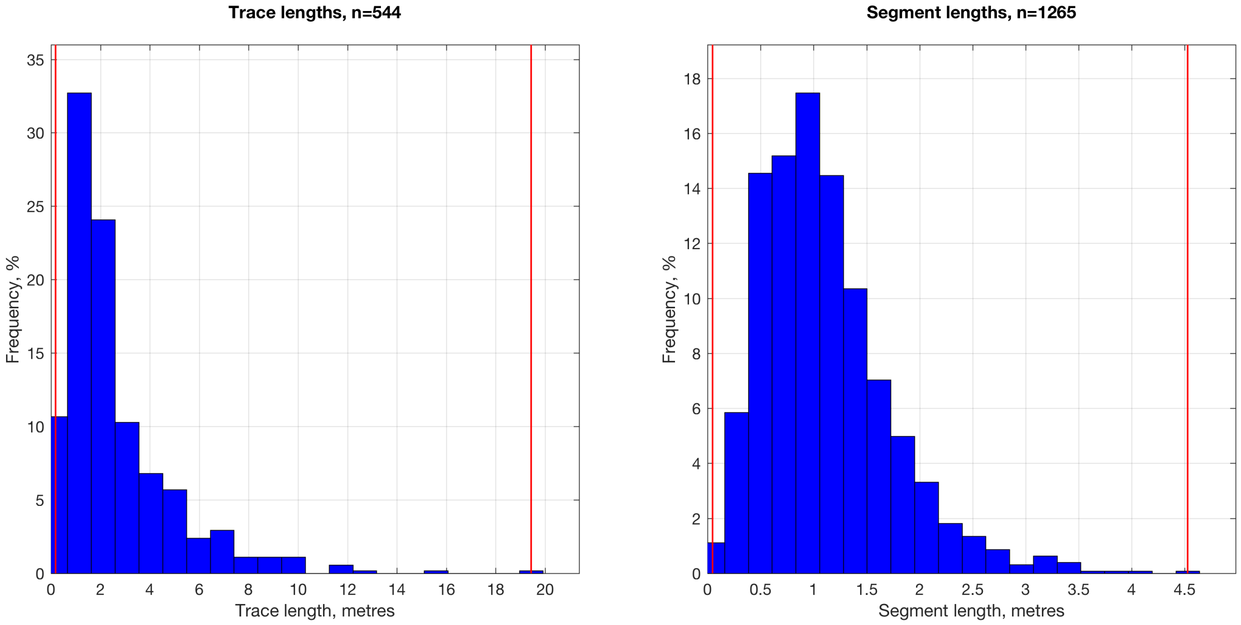


Figure 10. Left side shows frequency-size histogram for trace lengths. Right side shows frequency-size histogram for segment lengths. Red lines show the minimum and maximum lengths.

Lengths & sizes – Log-log lengths

Tick the check box 'Log-log lengths' to generate two cumulative frequency versus length plots on log-log scales for trace lengths and segment lengths. These can be useful for assessing the scaling of fracture lengths. Tick the check box 'Censor lengths at edges' to remove any traces or segments from the plots that intersect the boundaries of the mapped area (i.e. those fractures for which the true length is unknown).

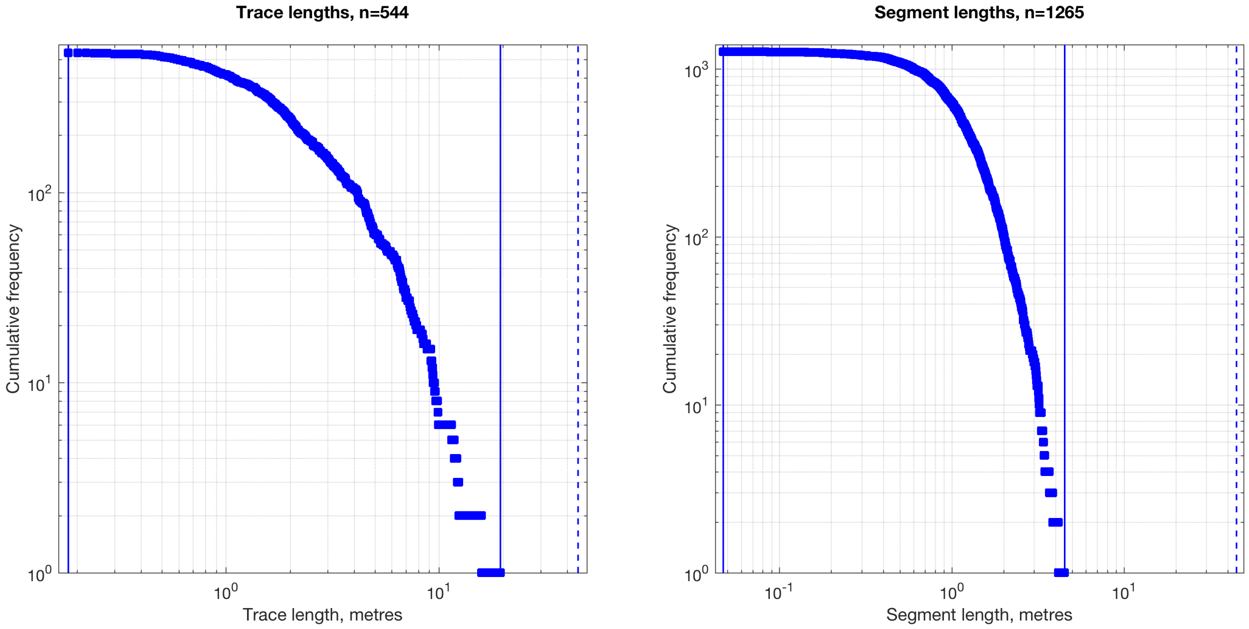


Figure 11. Left side shows a log-log plot of cumulative frequency versus trace lengths. Right side shows log-log plot of cumulative frequency versus segment lengths. The blue solid lines show the minimum and maximum lengths, and the dashed line shows the maximum possible length for this area (the longest diagonal of the rectangular bounding box).

Lengths & sizes – Maximum Likelihood Estimators (MLE analysis)

Tick the check box 'MLE analysis' to calculate the best-fitting power law, exponential and log-normal scaling distributions for the trace and segment lengths. Six plots are produced, three for trace lengths (power law, exponential, log-normal) and three for segment lengths (power law, exponential, log-normal). The key parameters for each of the best-fit distributions are displayed in a text boxes within the plots. The % degree of fit for each distribution for each of the trace and segment length distributions is printed in the MATLAB™ Command window.

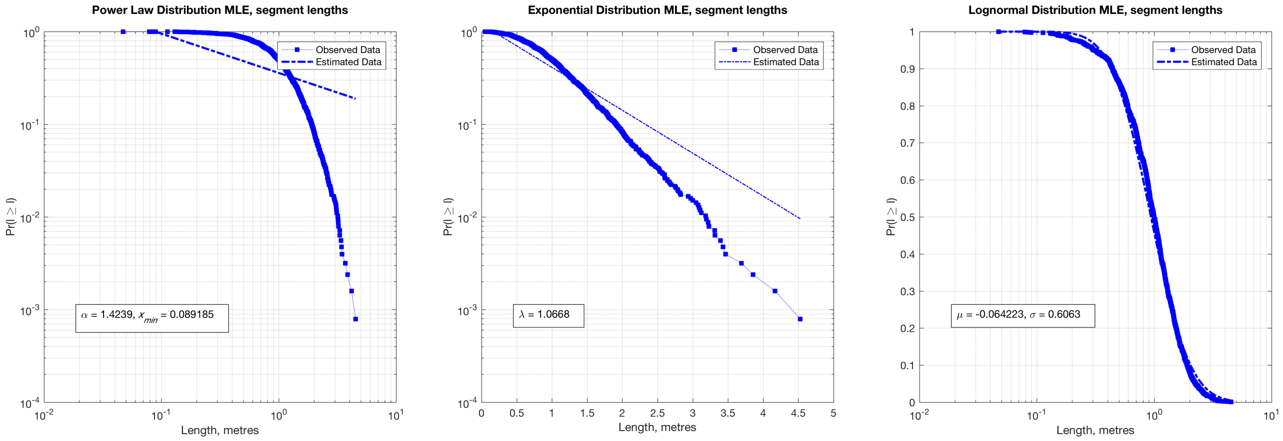


Figure 12. MLE analysis for segment lengths (trace length plots not shown). Left side shows the power-law scaling estimate, centre shows the exponential scaling estimate, right shows the log-normal scaling estimate. In each case, the key parameters of the best-fit distribution are provided in the inset text box. Note: different X- and Y-axis scales as appropriate for these plots (see Rizzo et al., 2017a for more details).

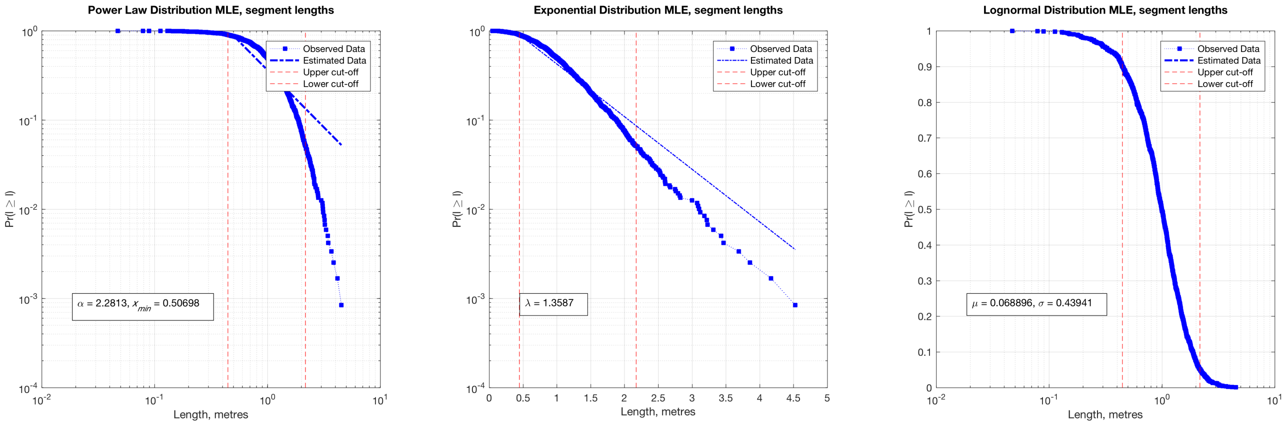


Figure 13. MLE analysis for segment lengths with applied upper and lower cut-offs, respectively of 10% and 5%, of the initial dataset (marked with red dashed lines). Compared with the previous figure (Figure 10), note the changes in the estimated statistical parameters (see inset text boxes).

Lengths & sizes – Variogram of segment lengths

From version 2.2 we include a plot showing the variogram (or semi-variogram) of segment lengths (see Bohling (2005) for an introduction). We calculate gamma as a function of the separation of two segments from the stored coordinate positions of all the segment mid-points (centroids) for every pair of segments. In future versions we plan to add curve fitting to this initial raw data plot, and allow for anisotropy (directional dependence) in the data.

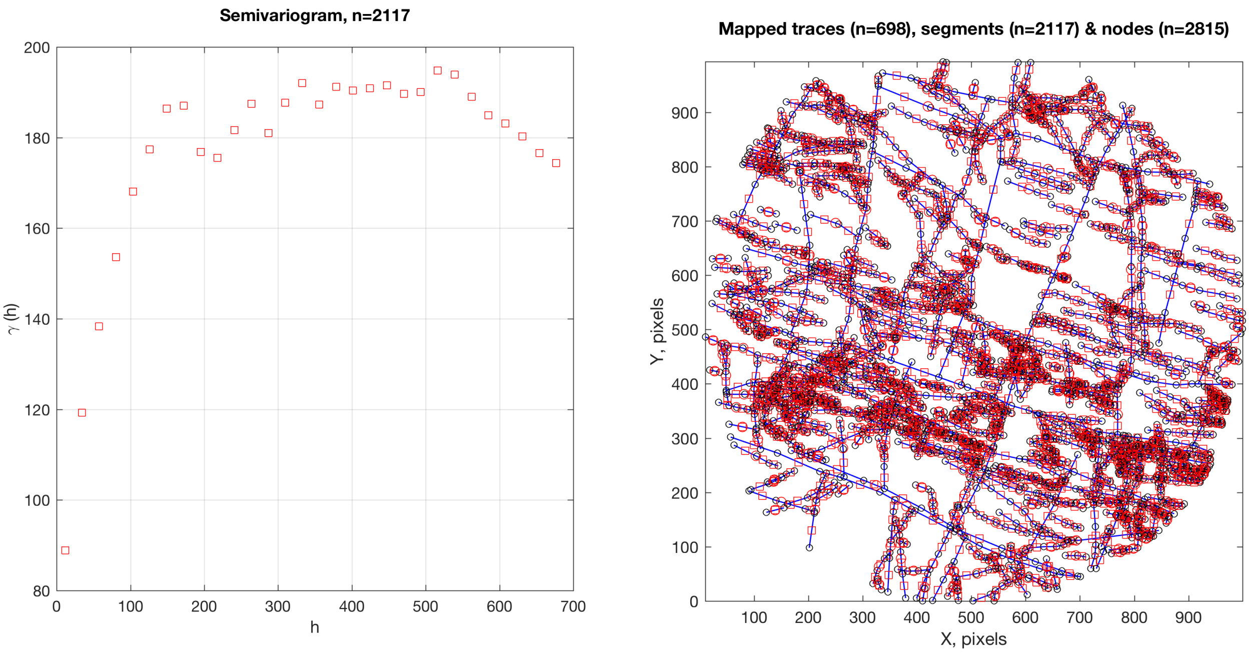


Figure 14. Variogram of segment lengths (left) for the fracture map shown with nodes (right).

Lengths & sizes – Cross plot lengths v angles

Tick the 'Cross-plot lengths v angles' check box to generate a scatter plot of segment lengths versus their angles. This may be useful to assess any correlation between fracture size and orientation.

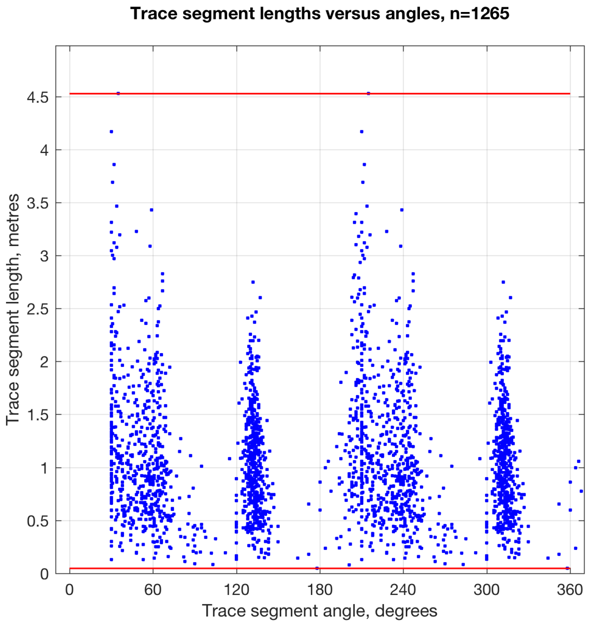


Figure 15. Cross plot of segment lengths and angles. The red lines mark the minimum and maximum lengths.

Lengths & sizes – Block sizes

Click on the check box 'Block size graphs' to generate data on block sizes bound by fractures in the loaded pattern. This produces three output figures: a map showing the locations of each block centroid, a log-log plot of cumulative frequency versus block area to assess the scaling relationship, and a simple histogram of block sizes (areas).

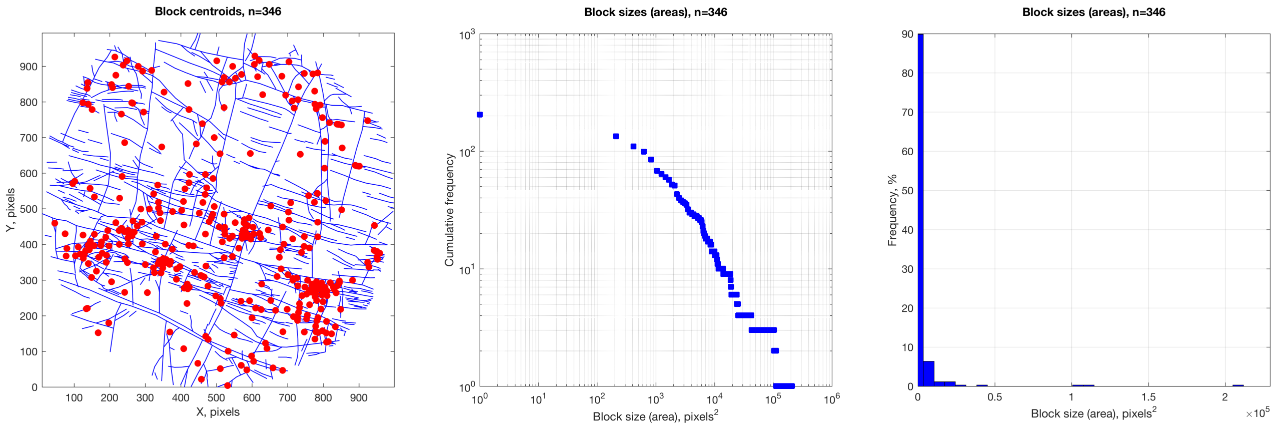


Figure 16. Output for block sizes. Left shows the centroid map, with block centroids shown as red dots overlain on the fracture trace map. Centre shows the log-log frequency-size (area) graph. Right shows a simple histogram of block sizes (areas).

10.3. Crack tensor plot

Click the check box 'Crack tensor plot' to generate 2-dimensional crack tensors from the fracture traces (see Oda et al., 1987 and Suzuki et al., 1997 for more details). The crack tensor combines data from the orientation distribution (angles) with the sizes (lengths) of the fractures and their spatial density to provide a single dimensionless measure of a crack pattern. The crack tensors of 0th, 2nd, 4th and 8th rank can be related to other physical properties, such as bulk permeability (see below), bulk elasticity and the acoustic velocities (e.g. P- and S-wave anisotropy). Note that the orientations used in the crack tensor calculation are the *poles* to the fracture trace segments; and therefore the magnitude of the tensor is largest in directions perpendicular to those shown in the rose diagram for the same data, which uses the angles of the trace segments, and not the poles.

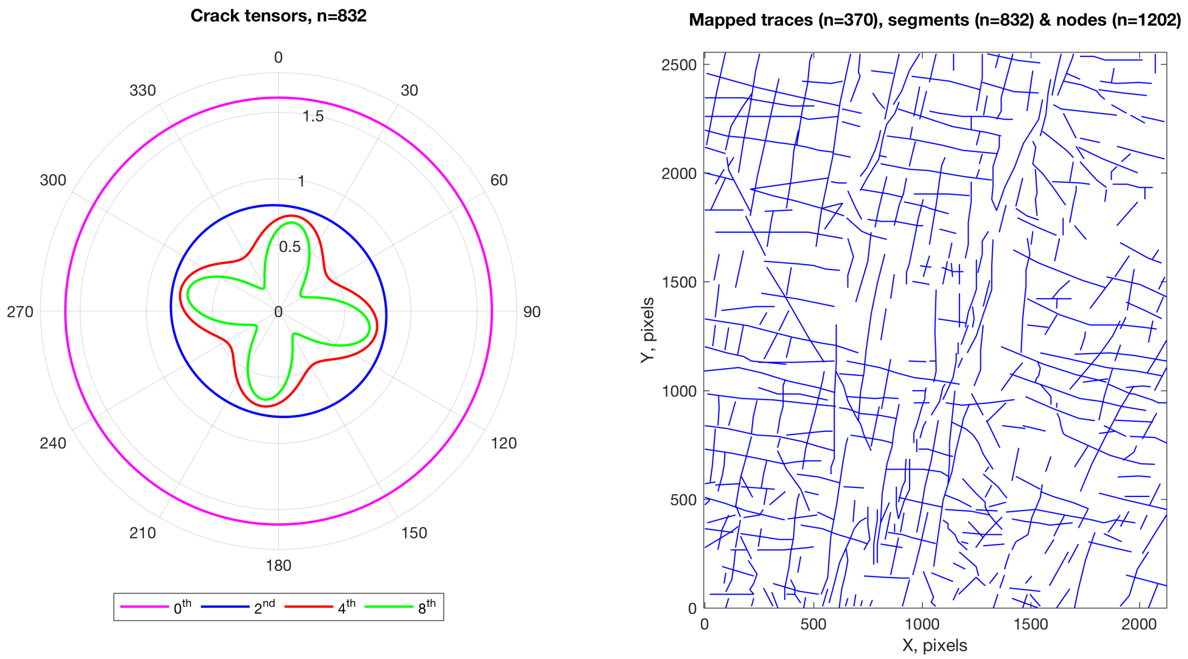


Figure 17. Crack tensors (left) for the fracture pattern shown in the map (right). 0th, 2nd, 4th and 8th rank crack tensors are shown in different colours.

10.4. Orientations – Histogram of angles

Click the check box 'Histogram of angles' to generate a simple graph of segment angles and their frequencies, expressed as a percentage of the total. The user can adjust the number of bins used in the histogram using the dropdown list box 'Number of bins' (choices of 10, 20, 30 or 50).

Orientations – Rose diagram

Click the check box 'Rose diagram (equal area)' to generate a rose plot of segment angles. An equal area format is preferred over linear scaling (Nemec, 1988) as it provides a more consistent orientation distribution. Tick the box labelled 'Length weighted' to weight each bin by the total segment length that lies within that angular arc. Tick the check box 'Show mean vector' to display a red line across the rose plot marking circular mean. Note that this measure is really only valid for *unimodal* distributions. The user can adjust the angular bin size using the dropdown list box ‘Bin size’ (choices of 5, 10, 20 and 30), and rotate the data by a specified angle in cases where the Y-axis of the mapped fracture pattern is not parallel to North by supplying a value in the text box labelled ‘Rotate Y-axis from N’ (see the section ‘**North Correction’**, above).

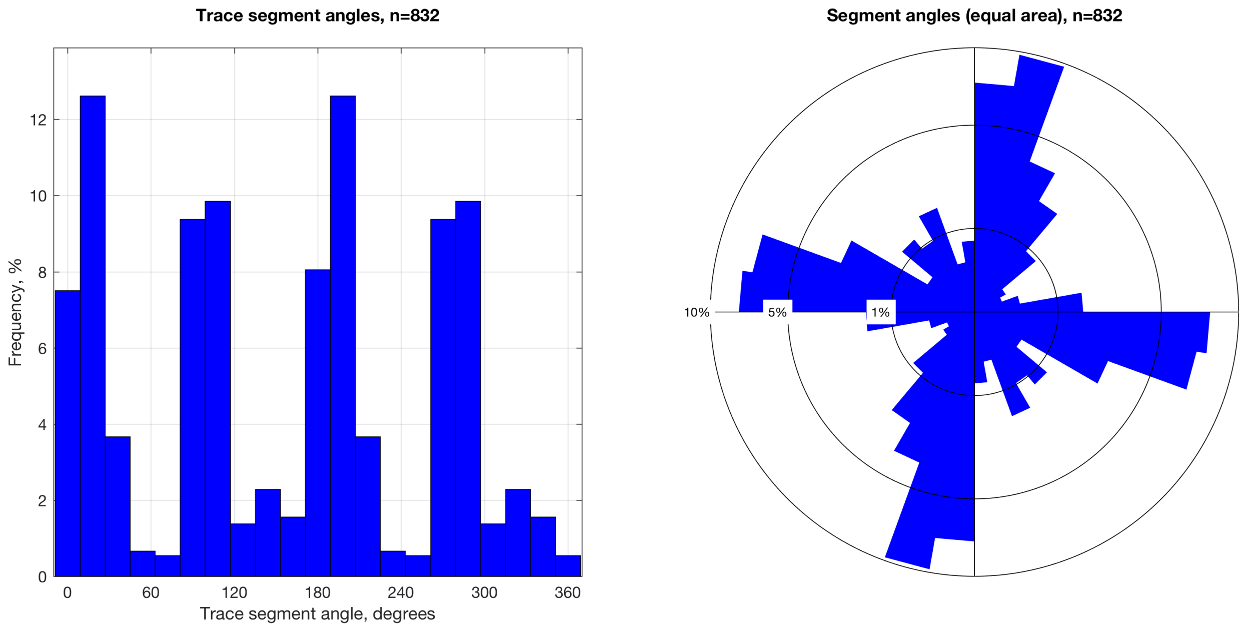


Figure 18. A histogram of segment angles (left) and an equal area rose diagram (right) of the same data.

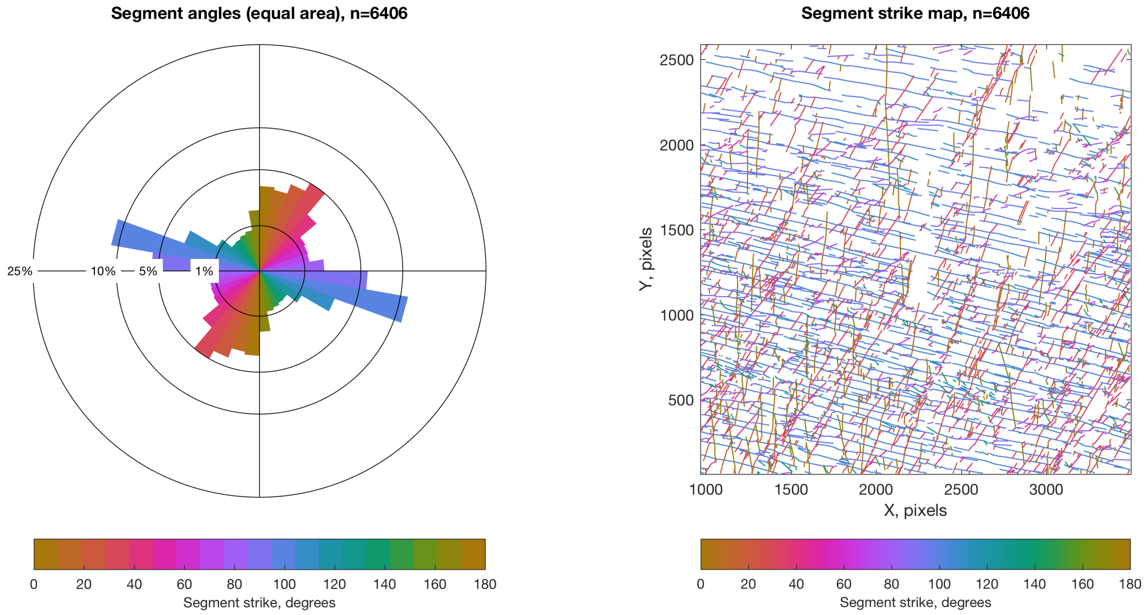


Figure 19. An equal area rose diagram colour-coded by segment strike (left) and the corresponding segment strike map (right).

10.5 Fluid flow – I-Y-X connectivity

Click on the check box 'I-Y-X connectivity' to generate a ternary plot of segment connectivity. The ternary plot is from Manzocchi (2002) and shows the relative proportions of isolated (I), splay or abutment (Y), and intersection (X) nodes in the fracture network. Better connected fracture patterns plot towards the base of the triangle (i.e. a higher proportion of X+Y nodes). If the hydraulic conductivity of the fractures differs significantly from that of the rock matrix (either higher or lower) then the connectivity of the network has implications for fluid flow through the rock mass.

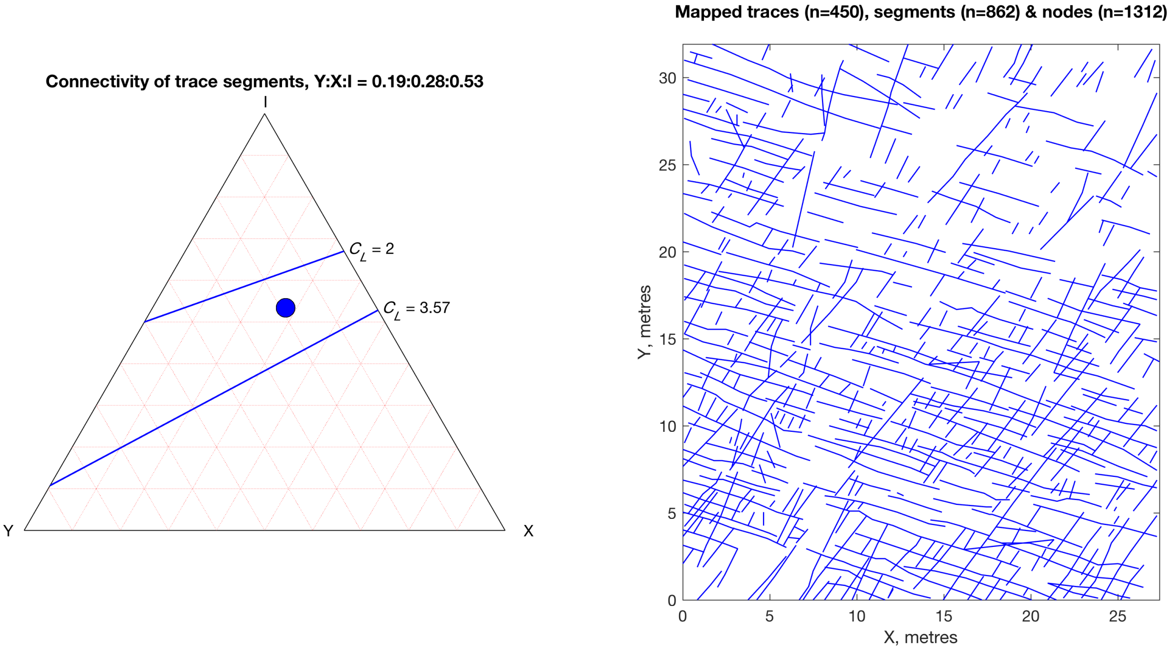


Figure 20. A ternary plot of fracture segment connectivity (left) for the fracture pattern shown in the map (right). Two contour lines for Connections per Line (C_L) are also shown, using indicative values described by Sanderson & Nixon (2015).

Fluid flow – Permeability ellipses

Click on the check box 'Permeability ellipse' to generate two plots of permeability anisotropy, one in the direction of flow and one in the direction of fluid pressure gradient (Long et al., 1982). Permeabilities are calculated using the 2nd rank crack tensor following the method of Oda et al. (1987) and Suzuki et al. (1997). Plotting of the ellipses follows the method of Long et al. (1982), with the semi-axes scaled as $\sqrt{k_1}$ and $\sqrt{k_2}$ for the direction of flow, and scaled as $1/\sqrt{k_1}$ and $1/\sqrt{k_2}$ for the direction of gradient, where k_1 is the maximum permeability and k_2 is the minimum. The permeability calculation is based on a parallel plate model of fractures, and a constant default aperture of 1 x 10^{-3} units is assumed. For the example shown, this equates to 1 x 10^{-3} metres = 1 mm, a reasonable default value. Scaled apertures can also be used, where aperture A is a function of segment length according to the general equation $A = aL^b$, where a is a constant factor and b is a power law exponent (>= 1).

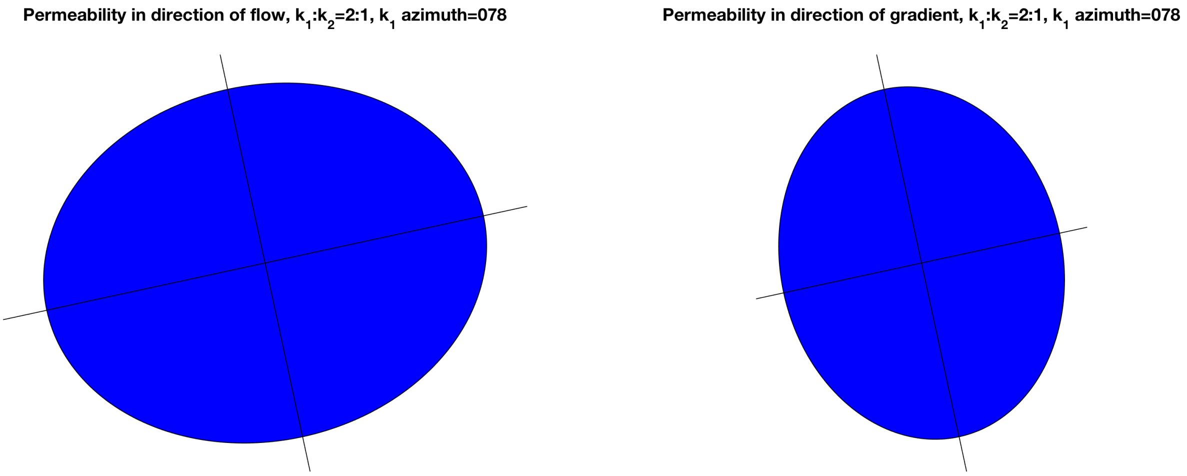
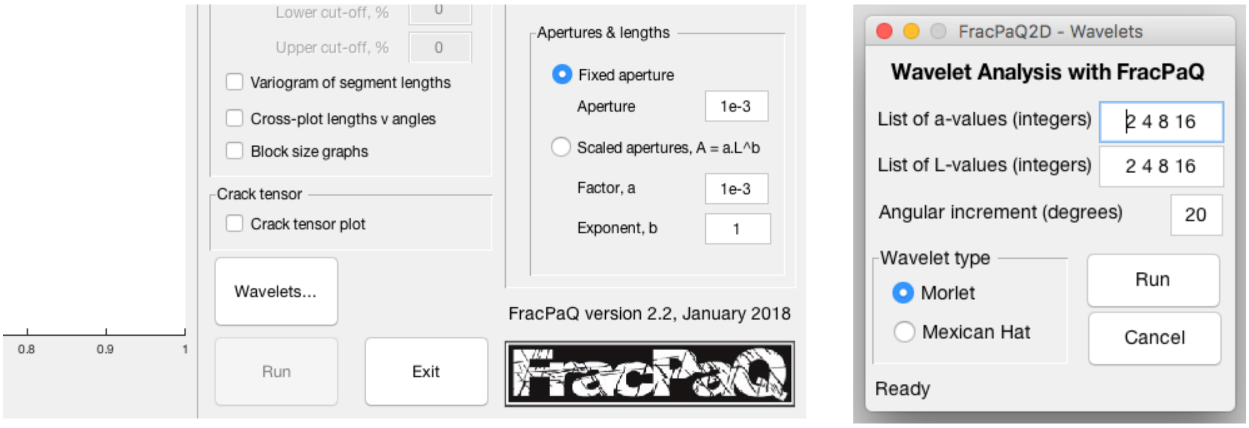


Figure 21. Permeability of a fracture network, plotted as 2-D ellipses to visualise the anisotropy in the direction of flow (left) and the direction of pressure gradient (right).

10.6 Wavelet analysis

FracPaQ version 2.2 implements 2D wavelet analysis of fracture maps (Rizzo et al., 2017b), using either Morlet or Mexican hat wavelet filters. Wavelet analysis (WA) is based on scaling and moving a filter, the selected wavelet, onto a signal which in our case is a binary fracture trace map. Compared to the classical Fourier Transform, WA results are better for two reasons: 1) the size of the wavelet can be adjusted according to the scale of the entity under consideration; 2) critically, the wavelet can be selected to best match the geometry of the entity, i.e. cracks. Building on the work of Ouillon et al. (1995), we have recently demonstrated the superiority of the Morlet wavelet over the Mexican hat wavelet in quantifying scale transitions in 2D fracture maps (Rizzo et al., 2017b).


Lower cut-off, % 0
Upper cut-off, % 0
□ Variogram of segment lengths
□ Cross-plot lengths v angles
□ Block size graphs
Crack tensor
□ Crack tensor plot

Apertures & lengths
● Fixed aperture
Aperture 1e-3
○ Scaled apertures, A = a.L^b
Factor, a 1e-3
Exponent, b 1

FracPaQ version 2.2, January 2018

FracPaQ2D - Wavelets
Wavelet Analysis with FracPaQ
List of a-values (integers) 2 4 8 16
List of L-values (integers) 2 4 8 16
Angular increment (degrees) 20
Wavelet type
● Morlet
○ Mexican Hat
Ready

Figure 22. The new Wavelets... button on the main GUI window (left) and the Wavelet analysis parameter dialog (right).

To run a Wavelet analysis, click on the Wavelets button on the main GUI window. A new smaller dialog window appears to collect the parameters needed for the analysis. The list of a-values controls the sizes of the wavelets used. Note that 1a = 2.2 pixels. The list of L-values controls the shapes (anisotropies) of the wavelets used. In general, it is best to use the same values for a and L, and a common pattern is to use a list such as 2^n, n=1,2,3... etc. The angular increment controls the subdivision of orientations used in the analysis: setting a smaller number here can significantly increase run-times. Wavelet analysis can be slow: processing of a typical fracture map for 3-4 values of a and L, and with a 20° angular increment can take 10-15 minutes on a recent (February 2018) computer. After choosing the parameter values, click Run, or cancel to return back to the main GUI window.

FracPaQ outputs from the Wavelet analysis are shown in Figure 23. For each a-value (selected length scale), 3 plots are produced: the Wavelet Coefficient (WC) map, the rose diagram of optimal orientations and the histogram of WC values.

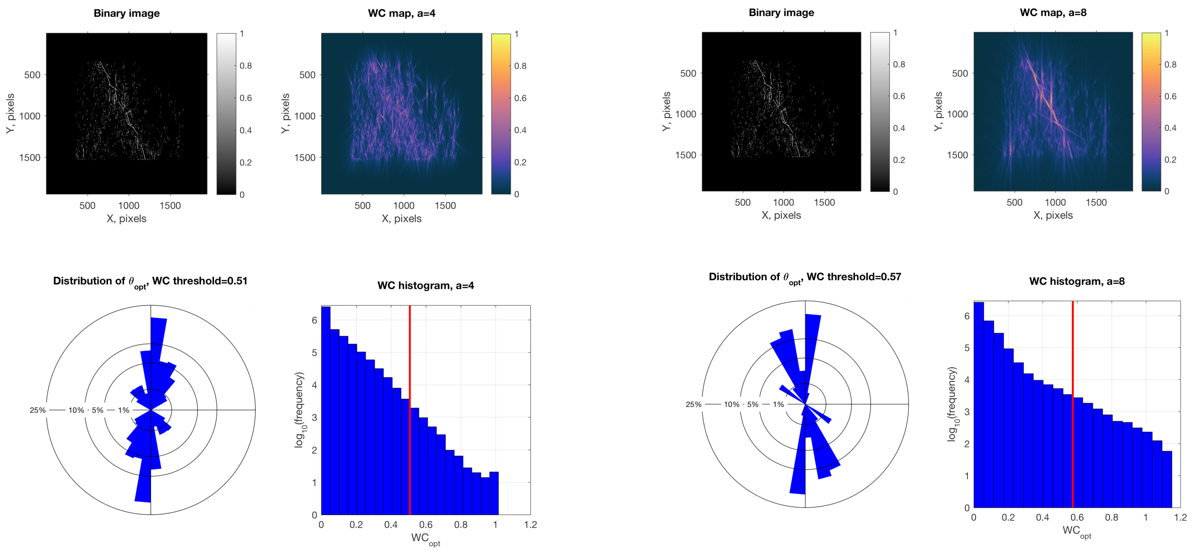


Figure 23. Examples of output from Wavelet analysis in **FracPaQ**. Left side shows results from Morlet wavelet analysis at a length scale of a=4, and right side shows same analysis for a length scale a=8. Note how the orientation distribution (rose diagrams) change at bigger length scales. See Rizzo et al., 2017b for more details.

11. Multi-coloured fracture maps

If you supply a multi-layer .svg node file as input, where each layer has traces of different colours, then **FracPaQ** version 2.2 can produce a separate set of output plots for each fracture set in the corresponding colours.

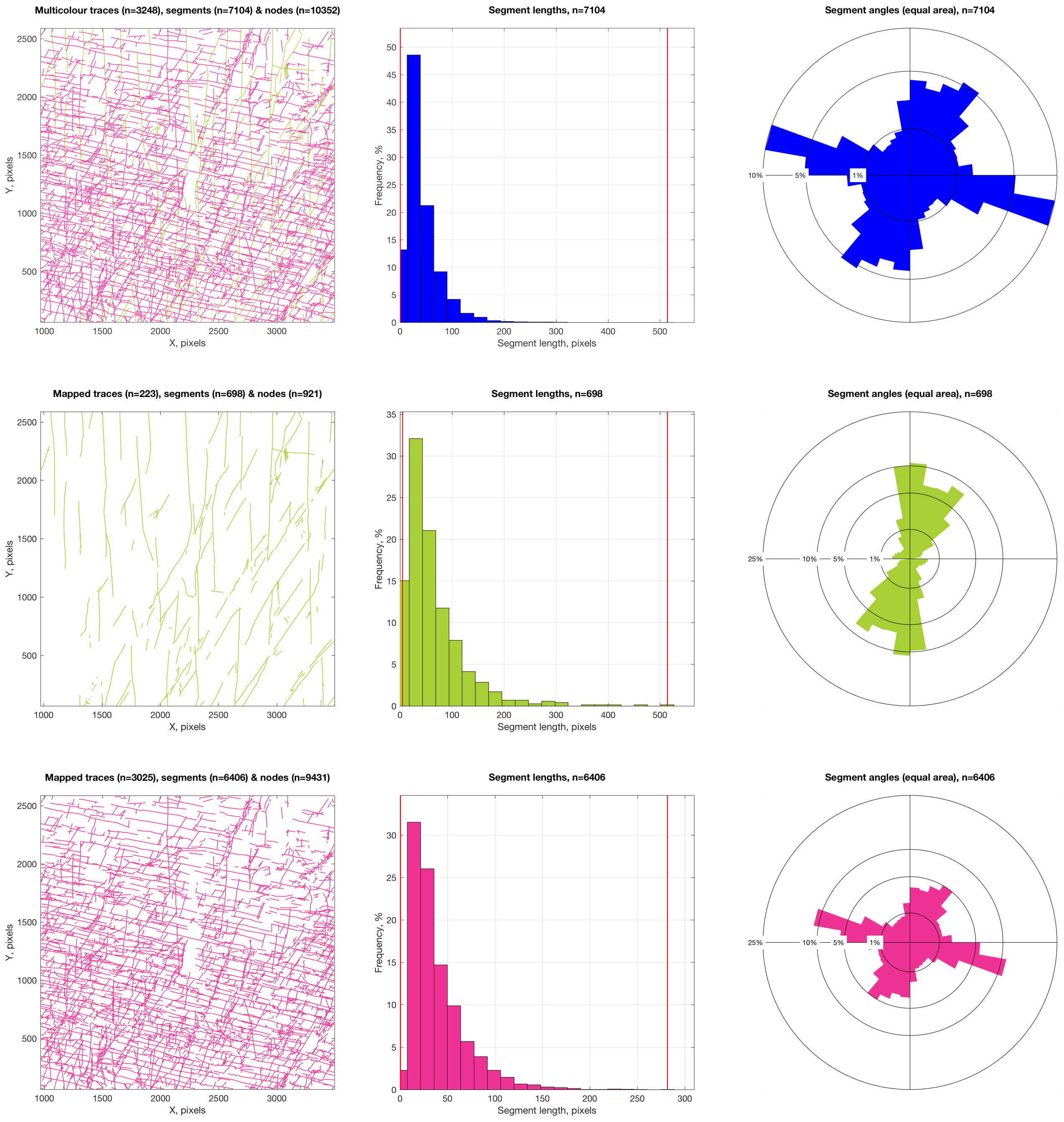


Figure 24. Examples of multicolour fracture trace support in **FracPaQ** version 2.2. Top row shows the outputs for the original input *.svg file containing two coloured layers for 2 different fracture sets (in this cases separated on the basis of vein fill). The middle row shows the trace map, segment lengths and rose plot for the green fractures only; the bottom row shows the same outputs for the pink fractures. These last two sets of output were generated by inputting the '*_colour*_converted.txt' files produced from the first step.

On reading the .svg file, **FracPaQ** reads in the colour code (in RGB format) from each layer and uses this colour to write out a separate .txt node file for each fracture set. These are named in the pattern 'name_colourHHHHHH_converted.txt', where HHHHHH is the hexadecimal colour code for the RGB colour used in the original .svg file for that set of fracture traces. See here for examples. There will be as many '*_converted.txt' files as there are different colours in the original .svg file. If you now input each of these files into **FracPaQ**, all the output plots will be coloured according to the fracture set colour.

12. Statistical estimations using MLE analysis

FracPaQ uses Maximum Likelihood Estimators (MLEs) to fit a given dataset of fracture lengths and then to extract the corresponding best-fit distribution parameters. To illustrate the methodology used in **FracPaQ** for fitting a given dataset and extracting the corresponding distribution parameters, we employ a synthetic power-law dataset. Before attempting any fitting or parameter estimation, data are arranged following a *cumulative distribution function* (CDF), $F(x)$, where the fracture trace length data ($X = x_1, x_2, \dots, x_n$) are plotted against the *probability* that x_i (a random variable) has a value greater than or equal to x, i.e. $Pr(x_i \geq x)$ (see Figure 19).

After this stage, **FracPaQ** can estimate the parameters governing the distribution, which, for a power-law distribution, are the α-value (i.e. the scaling parameter) and x_{min} (i.e. the minimum value until which the power-law function holds). In this case, for estimating α we need to have information about x_{min}, which constitutes the most challenging part of the estimation method. This involves the application of the Kolmogorov-Smirnoff (K-S) statistic, through which x_{min} is found minimising the value of the K-S statistic. The latter simply requires **FracPaQ** to calculate the maximum distance between the CDF of the data and the fitted model (inset in Figure 19):

$$D = \max_{x \geq x_{min}} |S(x) - F(x)| \qquad \text{(Eq. 1)}$$

where $S(x)$ and $F(x)$ are, respectively, the theoretical probability distribution and the empirical distribution (e.g. measured values for fracture trace lengths). The estimate of the lower bound of the observed data, $\hat{x}_{min}$, is the value of x that minimises D. This approach is completely general and is also used for the other statistical distribution considered in **FracPaQ**. Practically, to find the best possible $\hat{x}_{min}$ value, **FracPaQ** first runs through all the observed data and uses each datum as the 'true' $\hat{x}_{min}$; then it truncates the dataset to include every data above the temporary $\hat{x}_{min}$, and finally it uses these data to compute the *empirical* cumulative density function $F(x)$, and the theoretical CDF $S(x)$. Next, **FracPaQ** takes the maximum of the absolute value of the difference between each pair of empirical and theoretical CDF value (Eq. 1): this computation is the actual K-S statistic.

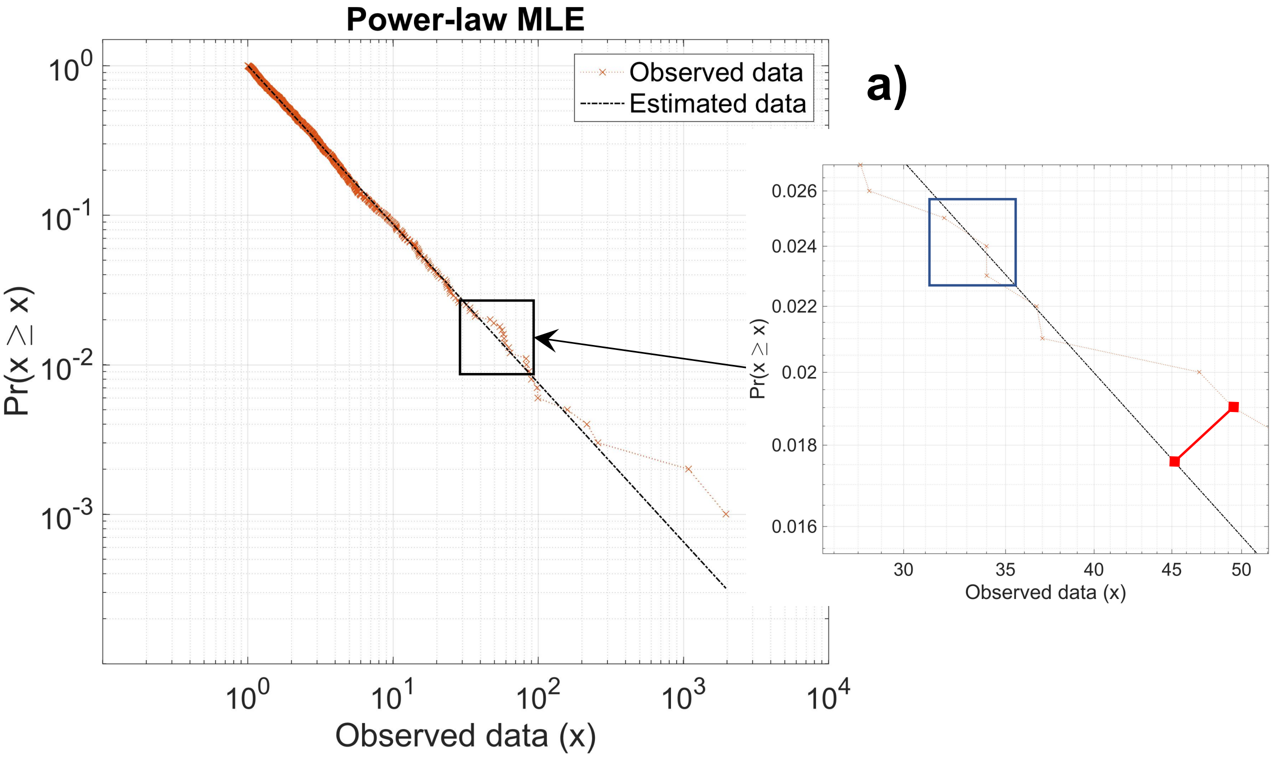


Figure 25. Example of cumulative density function probability $Pr\ (x_i \geq x)$ for MLE plot. Inset plot is a closer caption of the respective graph. Red bar is simply a graphical illustration of the calculation of the K-S statistic.

FracPaQ then calculates the minimum of the K-S statistic, finds the corresponding x_{min} value, and assigns it to be the most likely value of the *theoretical* CDF; in other words, the new 'true' x_{min}. Once the lower bound has been found, FracPaQ can then find the other parameter for the distribution of interest.

Despite its unequivocal robustness for parameter estimation (e.g. Rizzo et al., 2017a), the maximum likelihood method always yields a value of the MLE relative to the used model function, no matter which data one is using, even in the case where the analysed sample does not derive from that theoretical distribution. Therefore, no *quality* of fit is guaranteed. For this reason, a goodness-of-fit is needed and, following Clauset et al. (2009), **FracPaQ** uses the K-S test.

Computationally, in **FracPaQ** this K-S test is performed generating a large number of random datasets, all following the theoretical distribution – $S(x)$ – of interest, based on the corresponding estimated parameters. Hence, for the example case (i.e. a power-law distribution), these random *'new'* distributions would be power-law functions, having as parameter those estimated. Each of these random *'new'* distributions are then compared to the empirical dataset ($F(x)$) to determine if the two populations have the same underlying function. Intuitively, if D is large the fit is poor, whereas if D is small the fit can be considered good. At the same time, the relative scale of D is provided by its own probability distribution, through the calculation of the p-value. Under the hypothesis that the data follow the theoretical distribution, with the parameter obtained from the previous estimation (this is the null-hypothesis), the p-value provides the probability that the K-S statistic takes a value larger than the one obtained empirically. This mean that, if the p-value is greater than a set significance level, the test fails to reject the null-hypothesis and it be concluded that both the datasets follow the same statistical function; i.e. both $S(x)$ and $F(x)$ have the same underlying distribution. In contrast, if the p-value is smaller than the set significance level, the null-hypothesis is rejected. More formally, in **FracPaQ** if

$$p - \text{value} \geq 0.05 \ \Rightarrow \text{accept the null} - \text{hypothesis}$$

otherwise,

$$p - \text{value} \leq 0.05 \ \Rightarrow \text{reject the null} - \text{hypothesis}$$

Of course, the significance level of 0.05 is arbitrary and can be changed to another value. Because the $p - \text{value}$ derived from the test needs to be comparable with the chosen significance level (i.e. 0.05), to have a $p - \text{value}$ correct within two decimal points (Rizzo et al., 2017a; Clauset et al., 2009), 2500 K-S tests are by default performed when running the MLE analysis in **FracPaQ**.

13. Utilities

In FracPaQ v2.2, we provide two utility MATLAB scripts that may help users with common tasks in preparing input for **FracPaQ**. We will look to incorporate these within the main body of code in future versions. Run these scripts from the MATLAB™ command line.

13.1 Move2FracPaQ.m (contributed by X. Ogaya & J. Alcalde)

This script converts fracture traces from Midland Valley's Move™ software into the Node file format of **FracPaQ**. Full instructions for use are included in the header comments. of this script.

13.2 FracPaQ2D_imagePreprocess.m

This script illustrates some of the functions in the MATLAB™ Image Processing toolbox that can help to convert a 'raw' digital image, e.g. an outcrop photograph taken with a digital camera, into the 8-bit binary format needed for the **FracPaQ** Hough transform method. An excellent example of the methods that can be employed to improve 'signal-to-noise' ratios in these kinds of images can be found in Griffiths et al. (2017).

NB we strongly advise that, for the best results with FracPaQ, the user should digitise their images (or maps) using a graphics package and use the Node file input format (see section 7 above).

14. Acknowledgements

We have used public domain MATLAB™ code to enhance **FracPaQ**:

- 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
- cmocean.m written by **Kristen Thyng** and **Chad Greene**; available on Mathworks™ FileExchange: http://uk.mathworks.com/matlabcentral/fileexchange/57773-cmocean-perceptually-uniform-colormaps
- variogram.m written by **Wolfgang Schwangart**; available on Mathworks™ FileExchange: https://uk.mathworks.com/matlabcentral/fileexchange/20355-experimental--semi---variogram

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 original manuscript, which led to major improvements in the code, the published paper in the Journal of Structural Geology and this User Guide.

Matthew Belshaw and **Daniel Knight** (both MSc students at the University of Aberdeen, 2016-2017) contributed to the testing of FracPaQ, and specific details for producing .svg files in Adobe Illustrator™ and CorelDraw™, respectively.

If you use **FracPaQ** in your scientific research, please include a citation to one or more of the following papers, as appropriate:

Healy, D., Rizzo, R.E., Cornwell, D.C., Farrell, N.J.C., Watkins, H., Timms, N.E., Gomez-Rivas, E. & Smith, M. (2017). FracPaQ: a MATLAB™ toolbox for the quantification of fracture patterns. *Journal of Structural Geology, 95*, pp1-16.

Rizzo, R.E., Healy, D. & De Siena, L. (2017a). Benefits of a Maximum Likelihood Estimator for fracture attribute analysis. *Journal of Structural Geology, 95*, pp.17-31.

Rizzo, R. E., Healy, D., Farrell, N. J., & Heap, M. J. (2017b). Riding the right wavelet: Quantifying scale transitions in fractured rocks. *Geophysical Research Letters*.

NB these are all Gold open access.

15. Bibliography

Belshaw, M. (2017). Unpublished MSc thesis. University of Aberdeen.

Bohling, G. (2005). Introduction to geostatistics and variogram analysis. *Kansas Geological Survey*, *1*, 1-20.

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.

Griffiths, L., Heap, M. J., Baud, P. & Schmittbuhl, J. (2017). Quantification of microcrack characteristics and implications for stiffness and strength of granite. *International Journal of Rock Mechanics and Mining Sciences*, *100*, 138-150.

Healy, D., Rizzo, R.E., Cornwell, D.C., Farrell, N.J.C., Watkins, H., Timms, N.E., Gomez-Rivas, E. & Smith, M. (2017). FracPaQ: a MATLAB™ toolbox for the quantification of fracture patterns. *Journal of Structural Geology, 95*, pp1-16.

Knight, D. (2017). Unpublished MSc thesis. University of Aberdeen.

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.

Middleton, G. V. (2000). *Data analysis in the Earth Sciences using MATLAB*. Prentice Hall.

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

Neupauer, R. M. & Powell, K. L. (2005). A fully-anisotropic Morlet wavelet to identify dominant orientations in a porous medium. *Computers & geosciences*, *31*(4), 465-471.

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

Oda, M., Hatsuyama, Y. & Ohnishi, Y. (1987). Numerical experiments on permeability tensor and its application to jointed granite at Stripa mine, Sweden. *Journal of Geophysical Research: Solid Earth*, *92*(B8), pp.8037-8048.

Ouillon, G., Sornette, D., & Castaing, C. (1995). Organisation of joints and faults from 1-cm to 100-km scales revealed by optimized anisotropic wavelet coefficient method and multifractal analysis. *Nonlinear Processes in Geophysics*, *2*(3/4), 158-177.

Rizzo, R.E., Healy, D. & De Siena, L. (2017a). Benefits of a Maximum Likelihood Estimator for Fracture Attribute Analysis. *Journal of Structural Geology, 95*, pp.17-31.

Rizzo, R. E., Healy, D., Farrell, N. J., & Heap, M. J. (2017b). Riding the right wavelet: Quantifying scale transitions in fractured rocks. *Geophysical Research Letters*.

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.

Thyng, K. M., Greene, C. A., Hetland, R. D., Zimmerle, H. M., & DiMarco, S. F. (2016). True Colors of Oceanography. *Oceanography*, *29*(3), 10.

Vermilye, J. M. & Scholz, C. H. (1995). Relation between vein length and aperture. *Journal of structural geology*, *17*(3), 423-434.

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