

USER MANUAL

FRACTURE ANALYZER: A Python toolbox for the 2D analysis of fracture patterns

SUMMARY

- 1: INTRODUCTION
- 2: INSTALLING PYTHON ON WINDOWS
- 3: HOW TO PREPARE THE INPUT FILE
 - 3.1: FRACTURE LIST
 - 3.2: SCANLINE
 - 3.3: SCANAREA
- 4: HOW TO USE FA AND OUTPUT DATA
 - 4.1: FRACTURE LIST ANALYSIS OUTPUT FILE
 - 4.2: FIND SCANLINE INTERSECTION ANALYSIS OUTPUT FILE
 - 4.3: COMPUTE SCANAREA DENSITY ANALYSIS OUTPUT FILE
- 5: CONSIDERATIONS

1. INTRODUCTION

This Python script was created with the aim of generating fracture pattern attributes from digitized fracture dataset. The software is currently limited to 2D fracture patterns, assuming that the area has no topographic variation and that the fractures are orthogonal to the outcrop surface.

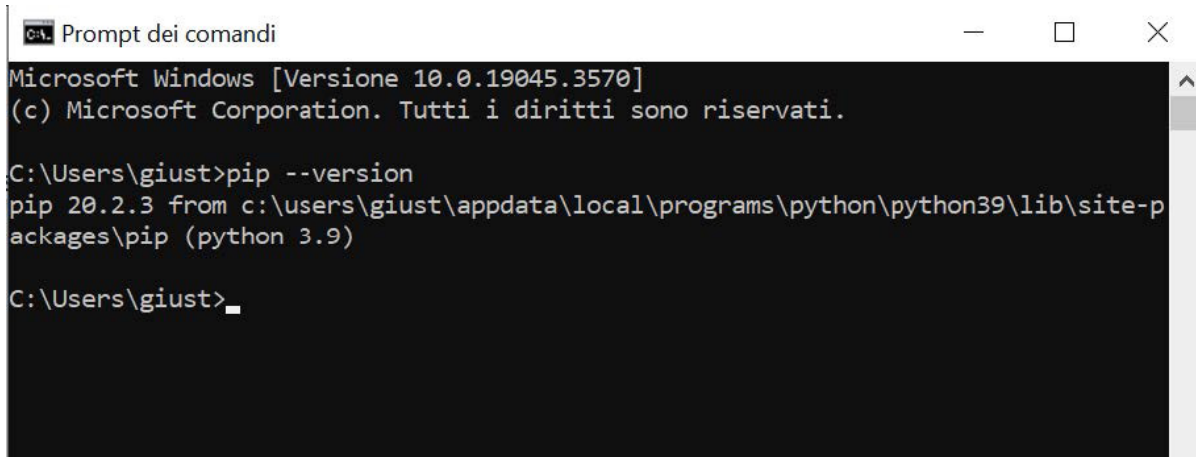
Fracture Analyzer (FA) acquires fracture attributes in the same way they would be acquired in the field, but in a digital way. This allows for the efficient management of large amounts of data and makes the software highly intuitive. Numerous analyses (e.g., scan-line and scan-area) can be performed using the same digital dataset of fractures to obtain new fracture attributes. The most crucial 2D attributes of fractures, such as their length, orientation, and position in space, can all be obtained using FA. Therefore, FA serves as a tool to facilitate the digital analysis of fracture patterns and to return their attributes. The software has no scale or data management limits. The only limitation is the computational capacity of the hardware used for analysis.

The software, source code and test files are available from the GitHub (<https://github.com/LorenzoBorghini/FA>). The script has been developed using the Python programming language.

Before running the Python version of the software, you must have the latest version of Python installed on your computer (3.12.0 or later versions), with the addition of some specific graphics library (i.e., Pillow library) as explained in the following pages.

2. INSTALLING PYTHON ON WINDOWS

- 1) Download and Launch the “windows installer” of Python from the website <https://www.python.org/downloads/windows/>; choose the last version (3.9.2 or later version), leave the default installation path and during the installation process put the check mark on “add Python to path”.
- 2) Control if PIP (which is the management system of the Python’s packages) is correctly installed. Launch on the command prompt (cmd.exe) the command: “pip --version” (Fig. 1). If the command answers with the PIP version 2.22 or later, go on to the next step; if the command was not recognized, Python has not been correctly installed. Uninstall it and repeat the procedure again.



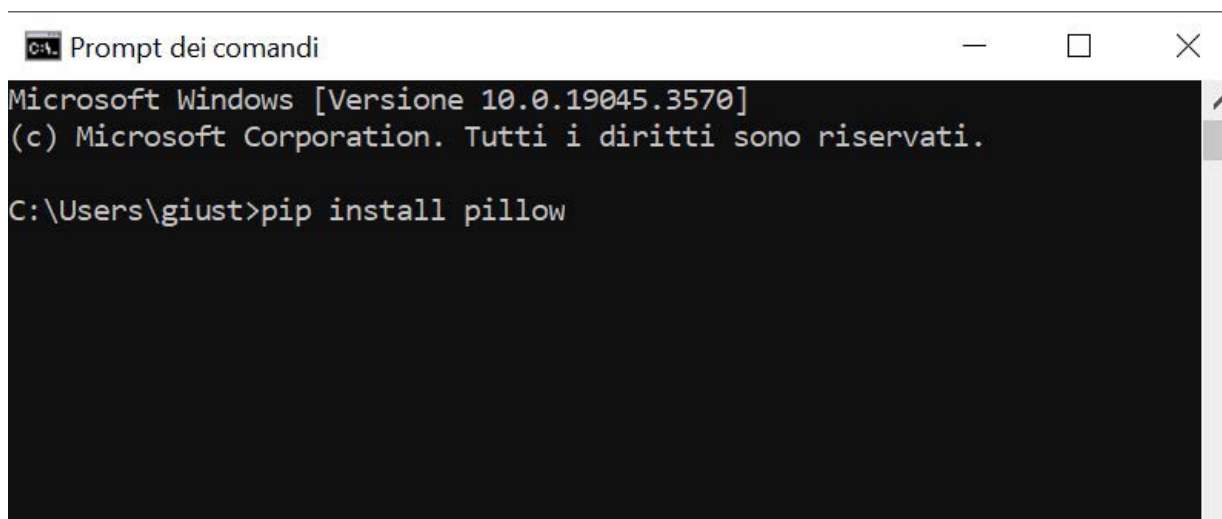
```

C:\Users\giust>pip --version
pip 20.2.3 from c:\users\giust\appdata\local\programs\python\python39\lib\site-packages\pip (python 3.9)
C:\Users\giust>

```

Fig. 1 - Command prompt to use to install Python.

- 3) Install “pillow” (graphic library) using PIP (whose installation was verified in step 2). To install it, open the command prompt (cmd.exe) and write the command: “pip install pillow” (Fig. 2). If the procedure is performed correctly the FA Python script is-ready to be used.



```

C:\Users\giust>pip install pillow

```

Fig. 2 - Command to install Pillow library in Python.

- 4) Finally, to use the program, launch FA.v5.35 (or later) double-clicking on the .py file (after having specified Python as default program for this extension). Alternatively, it is possible to open FA by using the command prompt with the following command: “python FA_v5.35.py”.

3. HOW TO PREPARE THE INPUT FILE

FA currently accepts only appropriately constructed and named “Scalable Vector Graphics” (.SVG) files. The dataset examples in this paper were created using Adobe Illustrator, the only software currently able to generate a fully FA-compatible SVG. The graphic file (.SVG) containing the digitized fractures must contain within it:

- 1) a layer for the scale (to be chimed compulsorily “scale”),
- 2) a layer for the fractures (to be chimed compulsorily “fractures”),
- 3) and a layer for the Scanline or Scanarea (to be chimed compulsorily respectively “scanline” or “scanarea”) (Fig. 3).

It should be noted that the “scale” layer is essential for all analyses and in it are placed a segment and a number indicating its real length in a unit of measurement at the user’s preference. The value assigned to the scale segment is absolute, requiring users to be aware of the specific scale they are working with. The layer containing the fractures must contain only vectors of the digitized fractures, and the layer of the analysis geometry (e.g., Scanline or Scanarea) must contain only a vector geometry (e.g., the “scanline” layer must contain only a line and the “scanarea” layer must contain only a closed polygon).

It’s important that all traces must be a lines or polylines. Occasionally, happens that “paths” (curved lines) are created instead of lines or polylines, which are still not supported by FA (Fig. 4). This happens when during fracture digitization instead of creating a broken line, curved lines are generated. To address this problem, if there are some path objects, it’s necessary to transform “path” structures into “lines” or “polylines” (i.e. rectify this curved lines). In Adobe Illustrator there is a tool called “simplify” (object -> traced -> simplify) which automatically accomplishes this command.

Once the file is complete, it can now be exported in SVG format, and opened with FA.

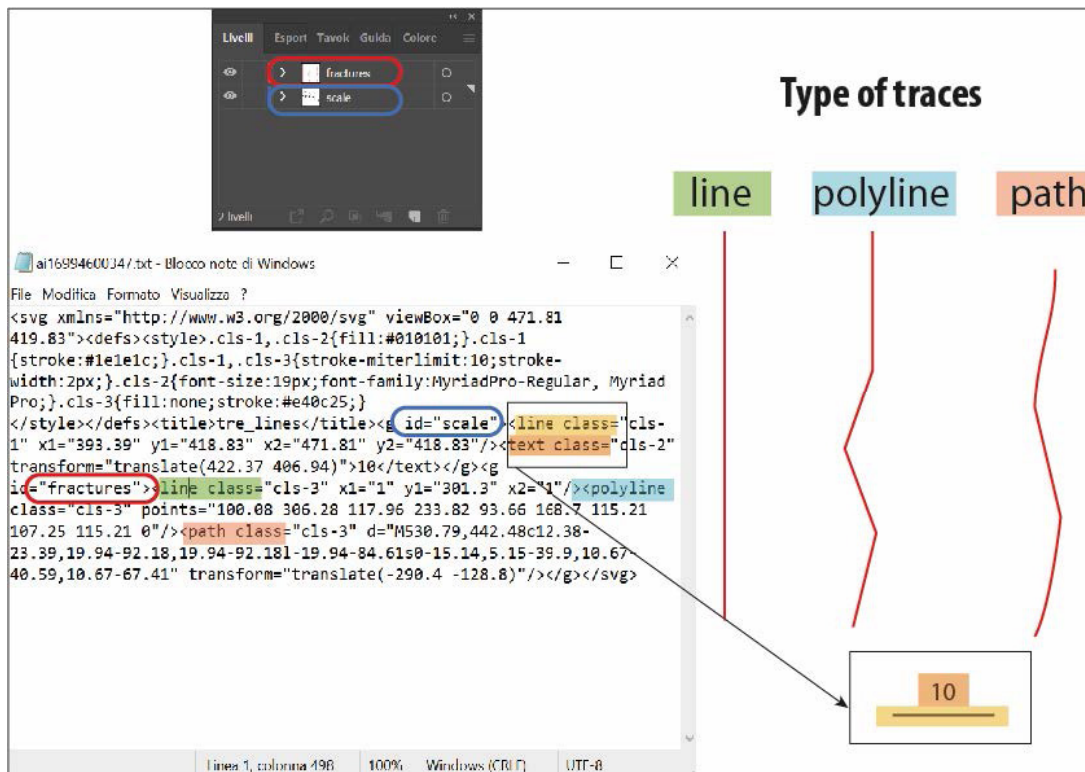


Fig. 3 - Example of creating process of an input file with respective layers and geometries.

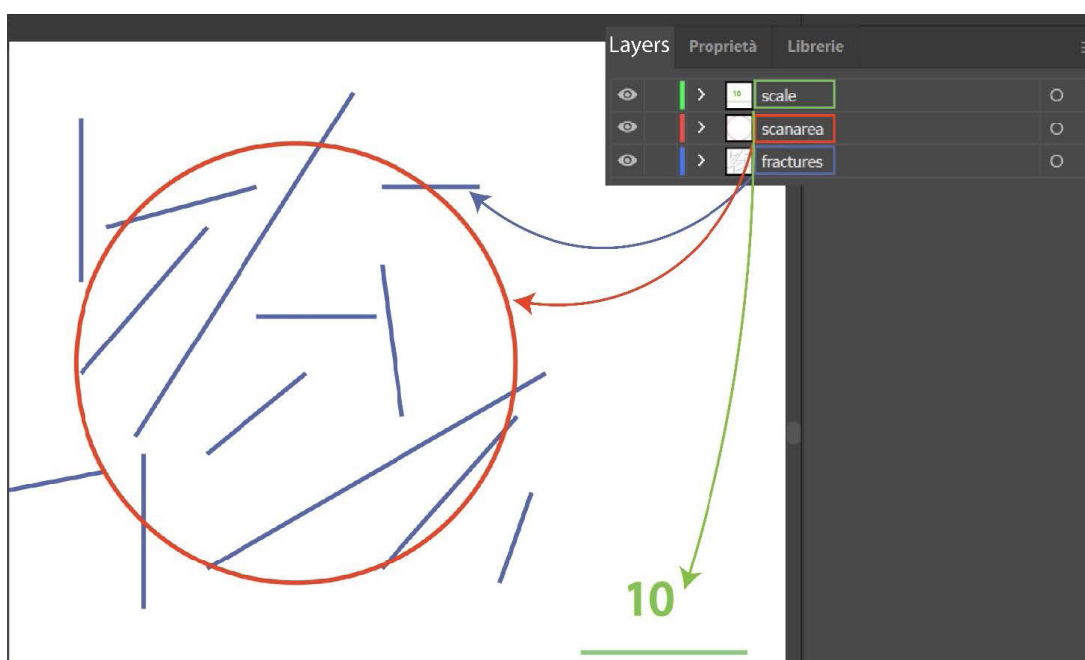


Fig. 4 - Main elements represented in a .SVG file opened with Block Notes.

3.1. Fracture list

The first step for preparing the input file for the fracture list analysis is drawing the fracture traces inside a layer name called “fractures”. The second layer that has to be created is called “scale” which is formed by two elements: a text box with a number (without unit of measure) and a line (Fig. 5). Be careful, the trace of the scale it can be only a line, not a polyline, otherwise the software doesn’t recognize the real length of the scale. After the digitization of the fractures is completed and the file is properly compiled, it can be exported in .svg format.

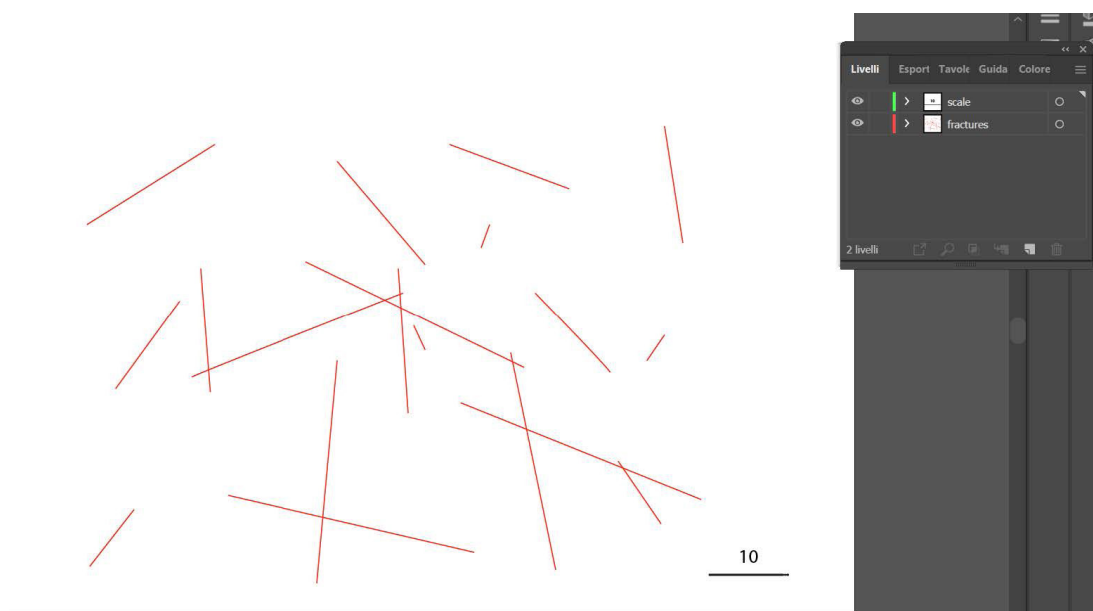


Fig. 5 - Example of an input file preparation in Adobe Illustrator, on the right are represented the layers name.

3.2. Scanline

To perform a linear scan line analysis of the digitized fracture dataset is necessary to have 3 layers: two of them, “fractures” and “scale”, are the same that are needed to analyze a fracture list as explained in chapter 3.4, while the third one must be nominated “scanline”. This layer contains only one line, i.e. the line cannot be made up more than two points (start and end point) (Fig. 6). The scanline trace can be drawn in all directions. It is important to remember that the north is always located at the top of the screen. After the digitization of the fractures is completed and the file is properly compiled, it can be exported in .svg format.

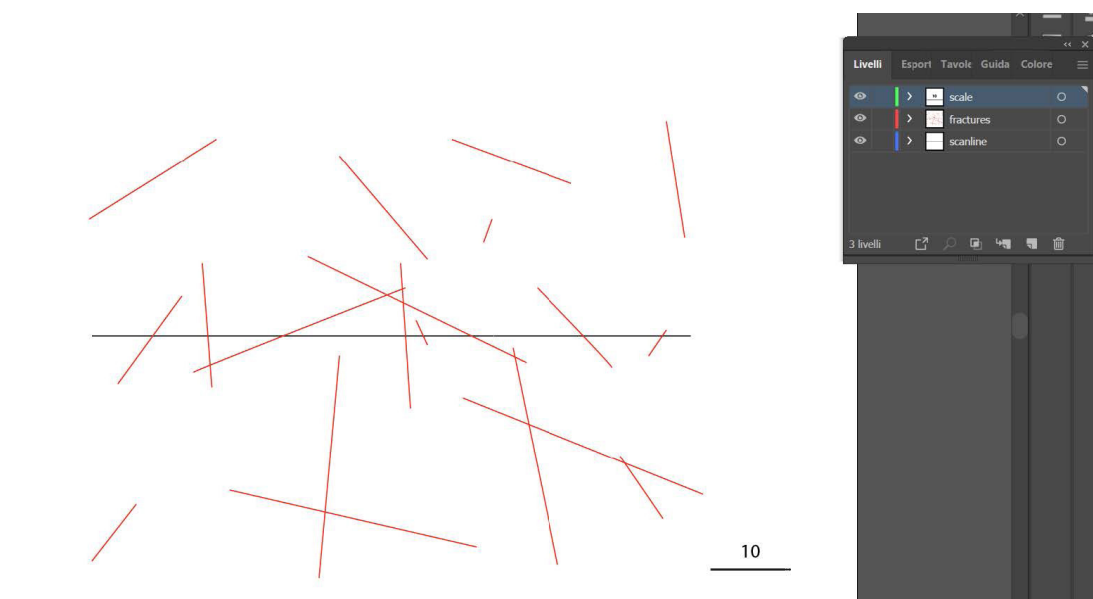


Fig. 6 - Example of a scanline file preparation, on the right are represented the layers name.

3.3. Scanarea

As for the preparation of the file for linear scanlines, to analyze scanareas it is necessary to have three layers, two of which are always the same i.e. “scales” and “fractures”. The third layer must be named “scanarea” in which there must be any closed polygonal shape (Fig. 7). It is recommended to use a command that creates closed polygon, rather than creating them with the “polyline” command, to avoid them remaining open. FA allows to use circular shapes, but also triangular, squares and polygonal shapes for the scanarea analysis. After the digitization of the fractures is completed and the file is properly compiled, it can be exported in .svg format.

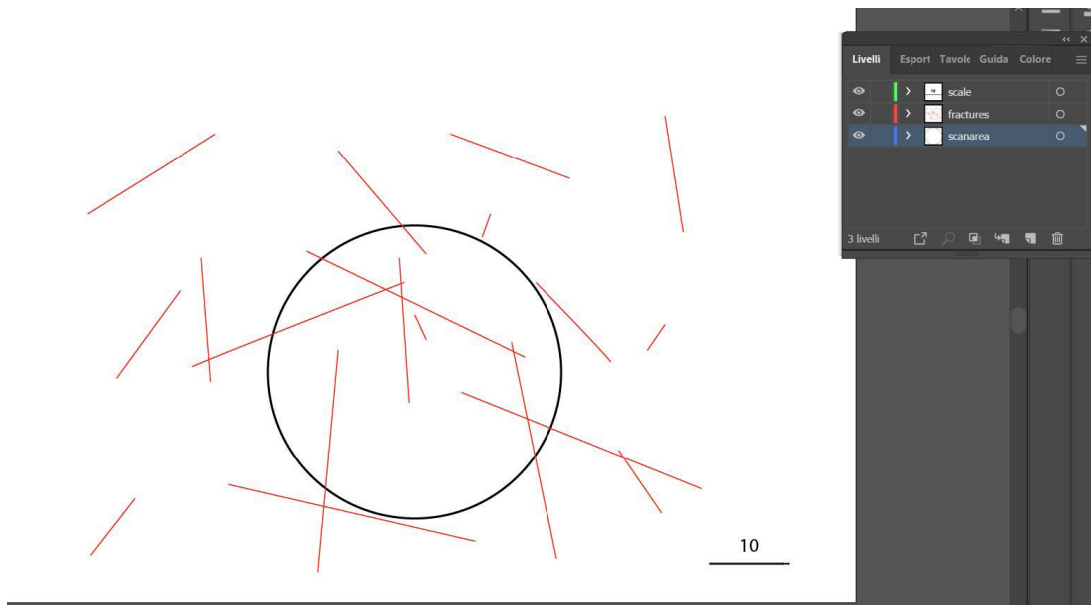


Fig. 7 - Example of a scanarea file preparation, on the right are represented the layers name.

4. HOW TO USE FA AND OUTPUT DATA

After creating the input files and exporting them to .svg it is possible to analyze them using FA. Then open FA and select the desired analysis function (Fig. 8). Upload the input file and wait for the output file to be created.

The output data is saved in the same folder for the input data, and it is made of a txt file containing the result of the analysis; the name of the file will be the same as the one of the input file with the addition of the analysis name (e.g. the scanline analysis having as input file outcrop_1.svg will produce the outcrop_1_scanline_analysis.txt file as output).

In the following subchapters, the possible analysis that FA can perform are described.

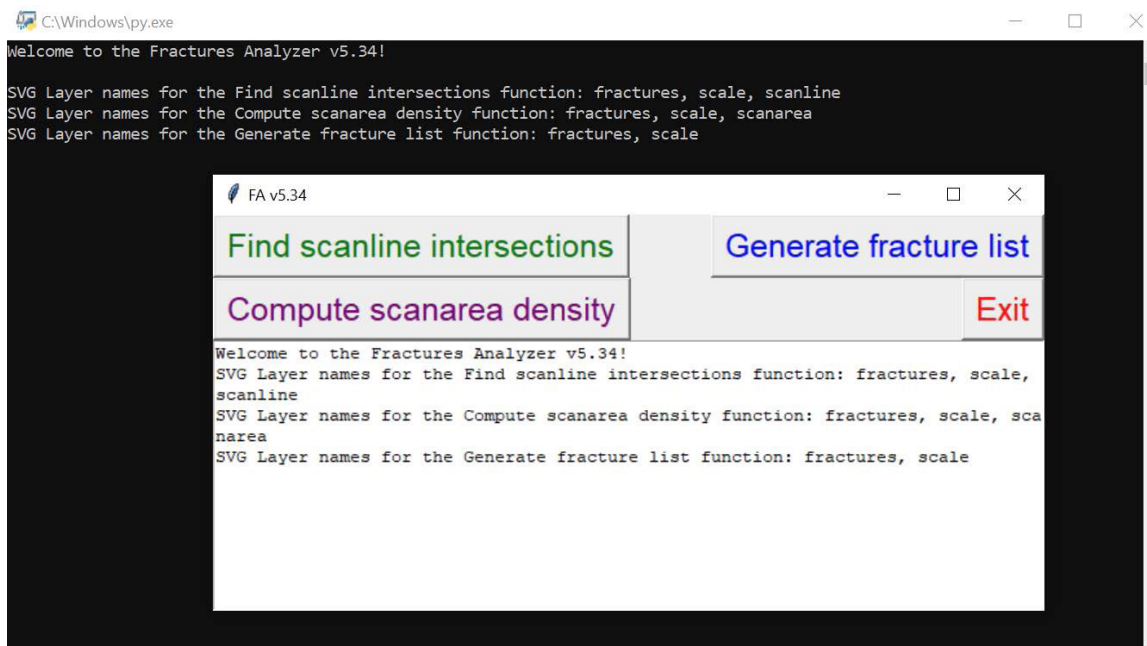


Fig. 8 - Initial screen of the FA software that provides a brief description of the layers name required for each analysis.

4.1. Fracture list analysis output file

The output file is a .txt file composed by a list of all the fractures inside the input file, with the length and strike of each fracture (Fig. 9c). In the window software are represented information about scale factor, length of the scale in pixel and the corresponding value in the normalized scale (Fig. 9a). The software also returns a temporary image representing the analyzed fractures (Fig. 9b).

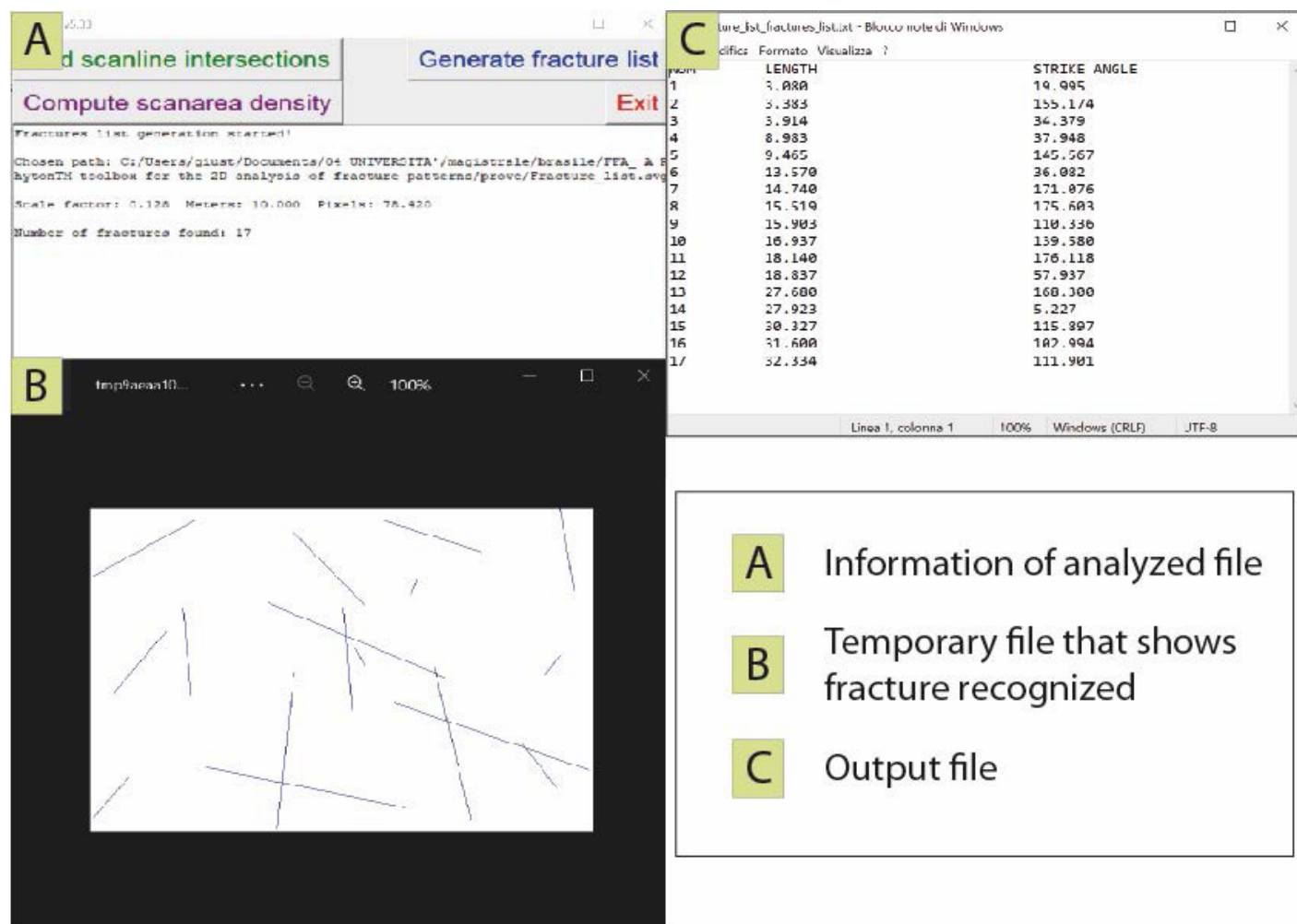


Fig. 9 - Output files from fracture list analysis. The image A shows information about processing. In the image B are shown the fractures that are present in the input file, output file and extracted attributes are showed in the image C.

4.2. Find scanline intersections analysis output file

The output file is a .txt file composed by the list of the fractures that cross the scanline, ordinated from the beginning of the scanline to the end. On the top of the file there are some information about the scanline e.g. length and strike. Each fractures attribute is represented in column and are length, strike angle, angle with the scanline, distance from the origin and apparent spacing (Fig. 10).

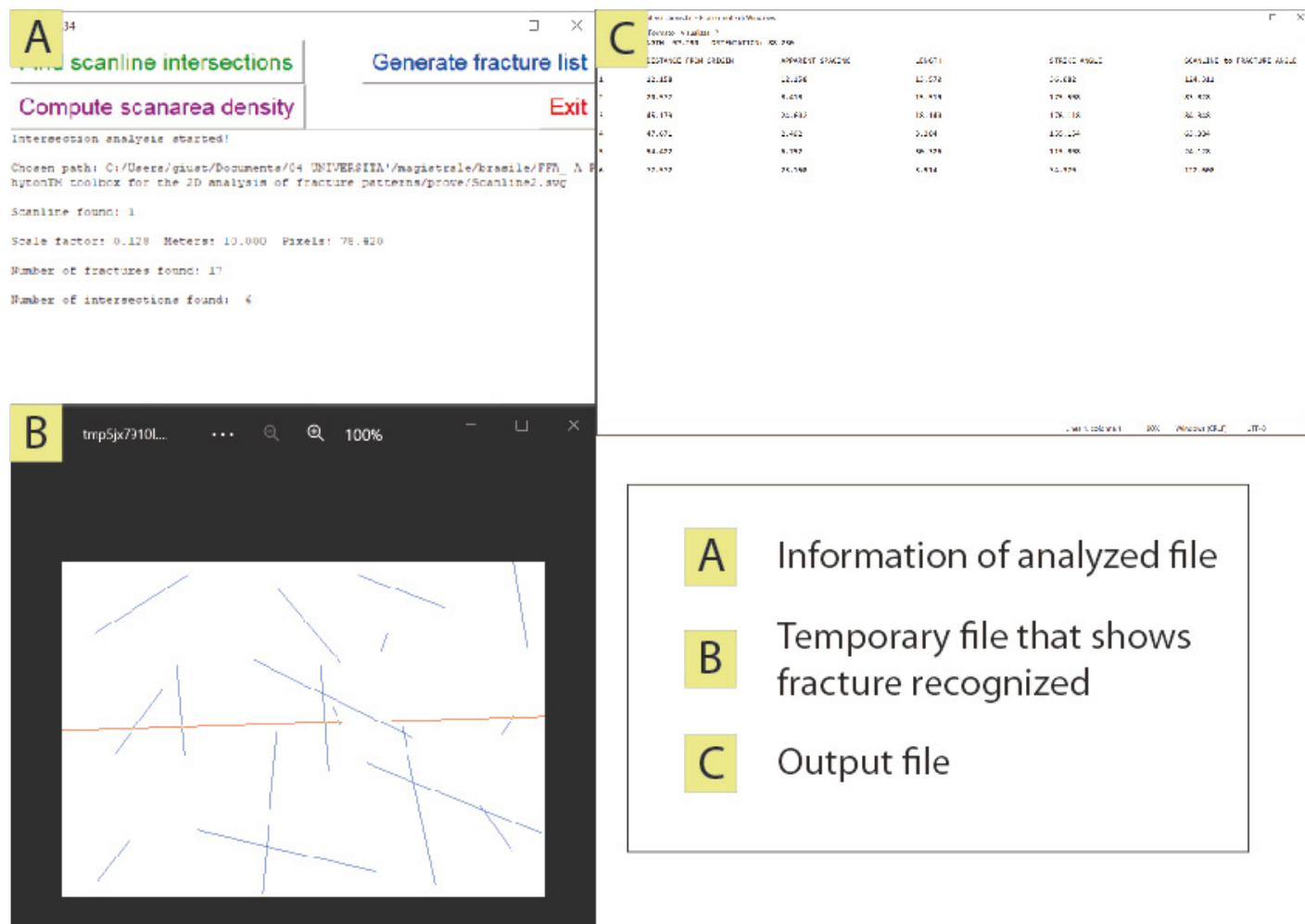


Fig. 10 - Output files from scanline intersections analysis. The image A shows information about processing. In the image B are shown the fractures that cross the scanline, output file and extracted attributes are showed in the image C.

4.3. Compute scanarea density analysis output file

The output file is a .txt file composed by the list of the fracture attributes inside the scanarea. At the top of the text file there are information concerning the area of the scanarea, number of fractures inside the scanarea, cumulative fractures length and P20 (fracture density), P21(fracture intensity) values. Each fractures attribute is represented in column and are number, length and strike angle (Fig. 11).

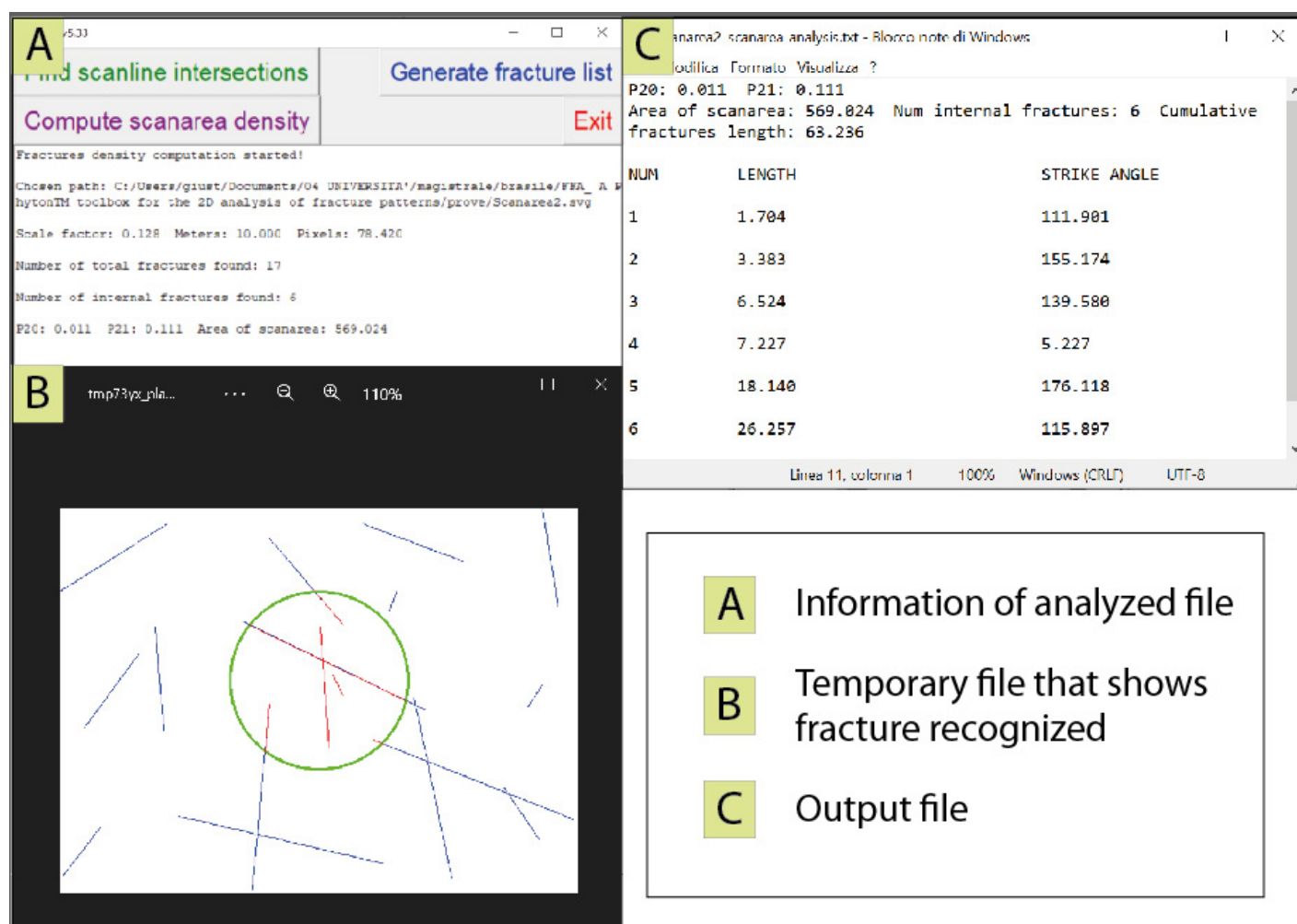


Fig. 11 - The image A shows information about processing. In the image B show the fractures inside the scanarea and the output file with fracture attributes are showed in the image C.

5. CONSIDERATIONS

It may happen that FA can encounter difficulties in processing the input file. The main reason for the failure of the FA output file typically lies in the construction of the input file. The most common errors include:

- Spelling errors in layer names or missing layer (Fig. 12).
- Presence of paths, especially in the scale segment and scanline segment, which should be exclusively lines (Fig. 14).
- Lack of one of the two elements (line and number) within the scale layer.
- Type of export, influenced by factors such as the graphics software type, version, and saving method (e.g., “save as” or “export”) (Fig. 13).
- Incompatible version of Windows.

In the following figures are showed possible problems of FA analysis.

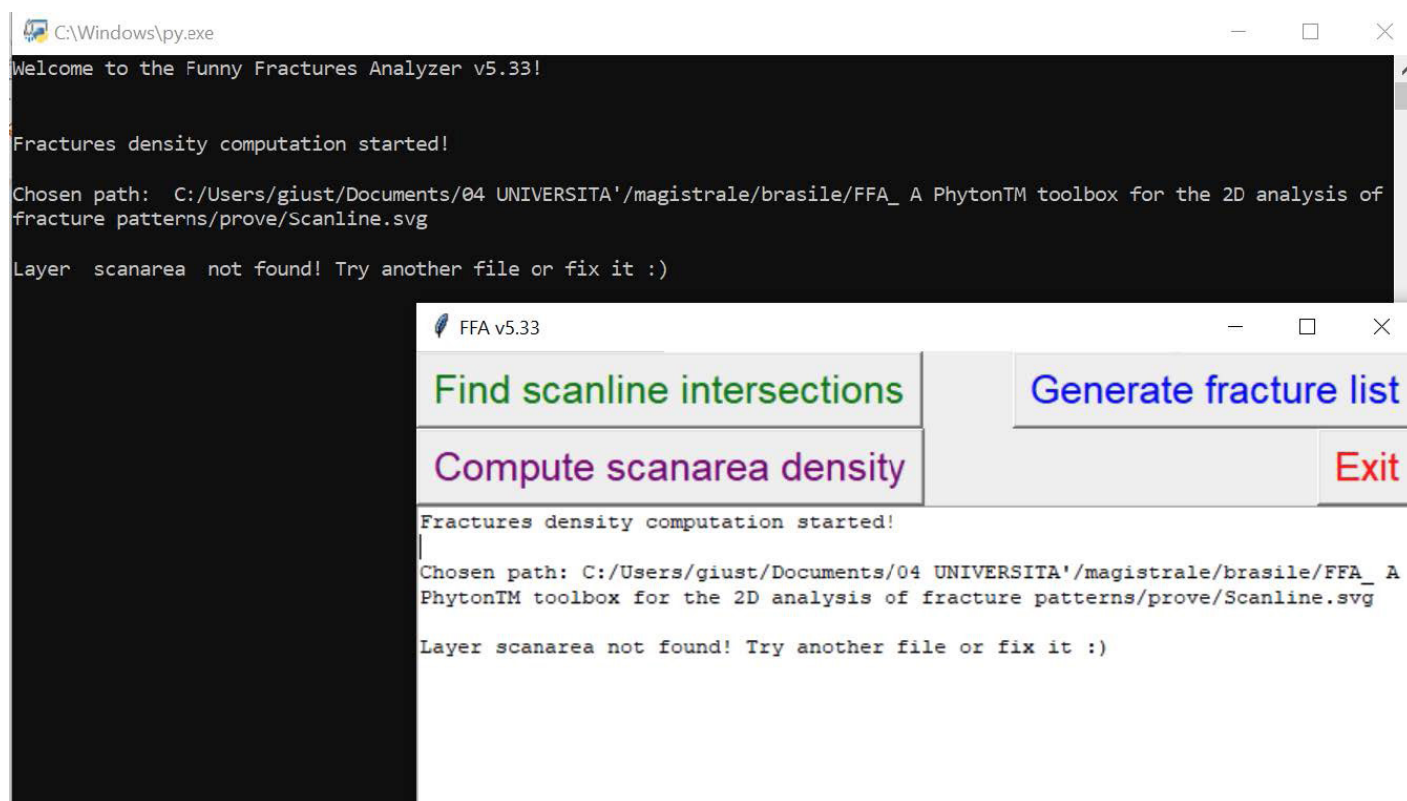
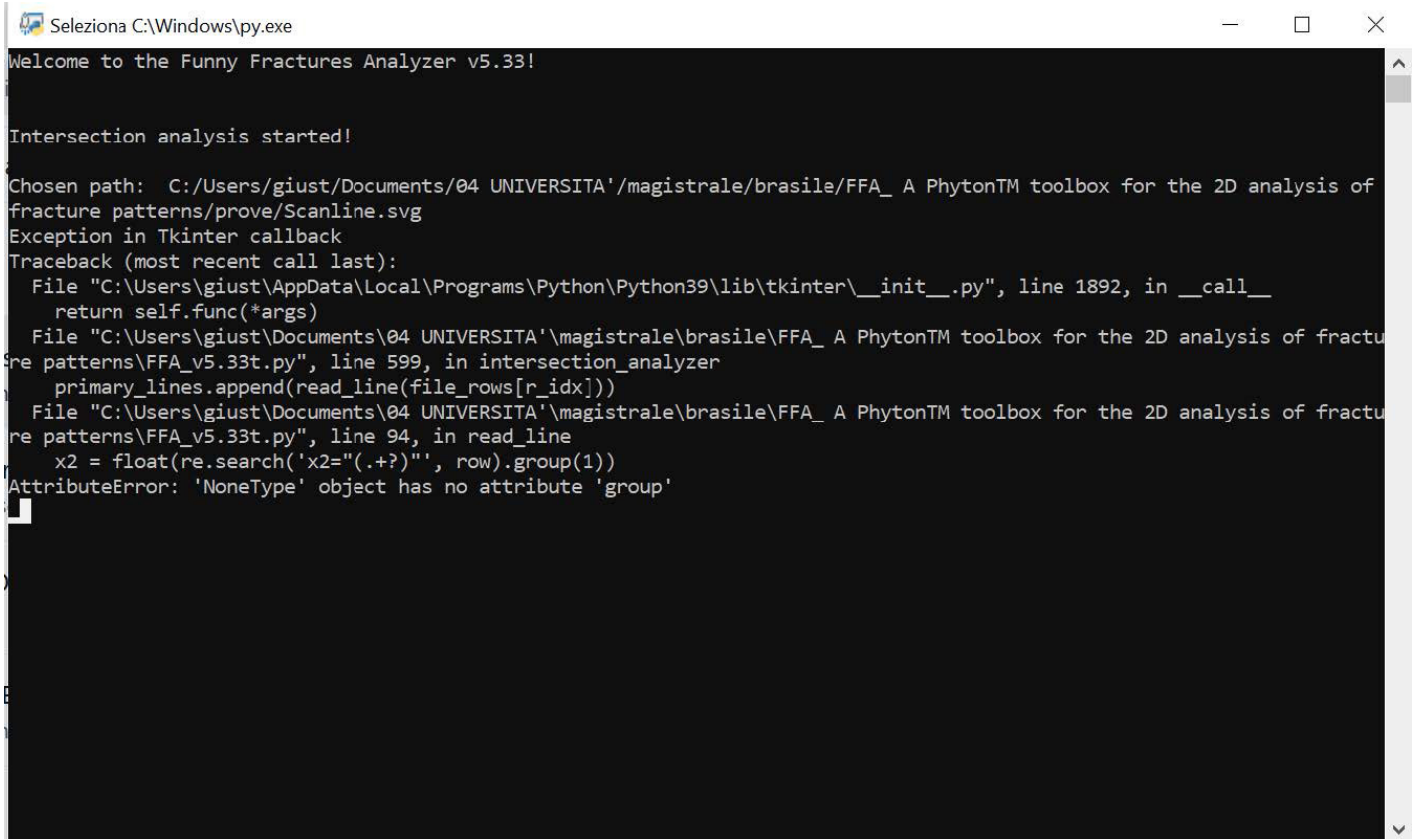


Fig. 12 - Problem observed when a layer missing (in this case layer scanarea).



```

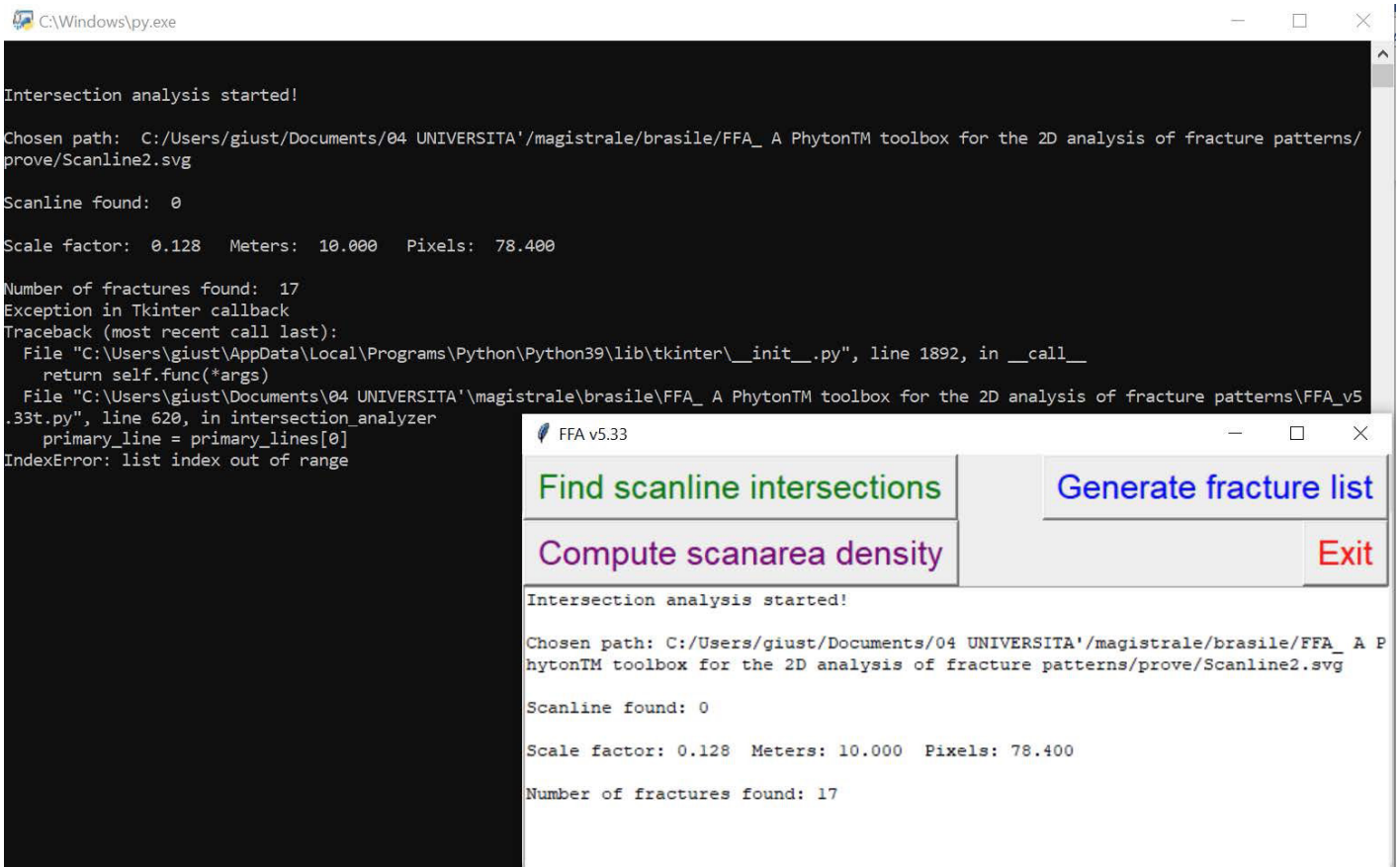
Seleziona C:\Windows\py.exe
Welcome to the Funny Fractures Analyzer v5.33!

Intersection analysis started!

Chosen path: C:/Users/giust/Documents/04 UNIVERSITA'/magistrale/brasile/FFA_ A PhytionTM toolbox for the 2D analysis of fracture patterns/prove/Scanline.svg
Exception in Tkinter callback
Traceback (most recent call last):
  File "C:\Users\giust\AppData\Local\Programs\Python\Python39\lib\tkinter\__init__.py", line 1892, in __call__
    return self.func(*args)
  File "C:\Users\giust\Documents\04 UNIVERSITA'\magistrale\brasile\FFA_ A PhytionTM toolbox for the 2D analysis of fracture patterns\FFA_v5.33t.py", line 599, in intersection_analyzer
    primary_lines.append(read_line(file_rows[r_idx]))
  File "C:\Users\giust\Documents\04 UNIVERSITA'\magistrale\brasile\FFA_ A PhytionTM toolbox for the 2D analysis of fracture patterns\FFA_v5.33t.py", line 94, in read_line
    x2 = float(re.search('x2="(.+?)"', row).group(1))
AttributeError: 'NoneType' object has no attribute 'group'

```

Fig. 13 - Problem with file due to different export path (doesn't recognize group layers), how to solve, change path to save SVG file e.g. not use "export as", try to use "save as".



```

C:\Windows\py.exe
Intersection analysis started!

Chosen path: C:/Users/giust/Documents/04 UNIVERSITA'/magistrale/brasile/FFA_ A PhytionTM toolbox for the 2D analysis of fracture patterns/prove/Scanline2.svg
Scanline found: 0

Scale factor: 0.128 Meters: 10.000 Pixels: 78.400

Number of fractures found: 17
Exception in Tkinter callback
Traceback (most recent call last):
  File "C:\Users\giust\AppData\Local\Programs\Python\Python39\lib\tkinter\__init__.py", line 1892, in __call__
    return self.func(*args)
  File "C:\Users\giust\Documents\04 UNIVERSITA'\magistrale\brasile\FFA_ A PhytionTM toolbox for the 2D analysis of fracture patterns\FFA_v5.33t.py", line 620, in intersection_analyzer
    primary_line = primary_lines[0]
IndexError: list index out of range

```

FFA v5.33

Find scanline intersections Generate fracture list

Compute scanarea density Exit

```

Intersection analysis started!

Chosen path: C:/Users/giust/Documents/04 UNIVERSITA'/magistrale/brasile/FFA_ A PhytionTM toolbox for the 2D analysis of fracture patterns/prove/Scanline2.svg

Scanline found: 0

Scale factor: 0.128 Meters: 10.000 Pixels: 78.400

Number of fractures found: 17

```

Fig. 14 - Problem observed in an input file with 3 layers correctly added and written, but the trace of the scanline is a "path".