Q-Facies
Reference Guide

Version 1.0

This document is a quick guide for Q-Facies software installation and use.

Q-Facies software code is publicly accessible via GitHub (https://github.com/chesstor/Qfacies)

Q-Facies has been developed with Python version 3.7.6, using the libraries NumPy 1.19.2, Pandas 1.1.3, Matplotlib 3.3.2, and Scikit-learn 0.23.2.

All the information about the application and scientific details of this software can be found in the paper:

Q-Facies a tool for the quantitative interpretation of spatial and temporal distribution of hydrochemical facies

Miguel González Jiménez, Héctor Aguilera Alonso, Luis Moreno Merino, Amalia Romero Prados.
Instituto Geológico y Minero de España (IGME), CSIC, Ríos Rosas, 23, 28003 Madrid, Spain

miguigonn@gmail.com (M. González-Jiménez)
h.aguilera@igme.es (H. Aguilera)
l.moreno@igme.es (L. Moreno Merino)
a.romero@igme.es (A. Romero)

It is necessary to complete the following steps to use the software:

- **Set up the program files on the computer**
- **Prepare the input data**
- **Set the analysis, graphic, and output options**
- **Run the program**

Keep in mind:

The program reads all the operating options from the Options.txt file.

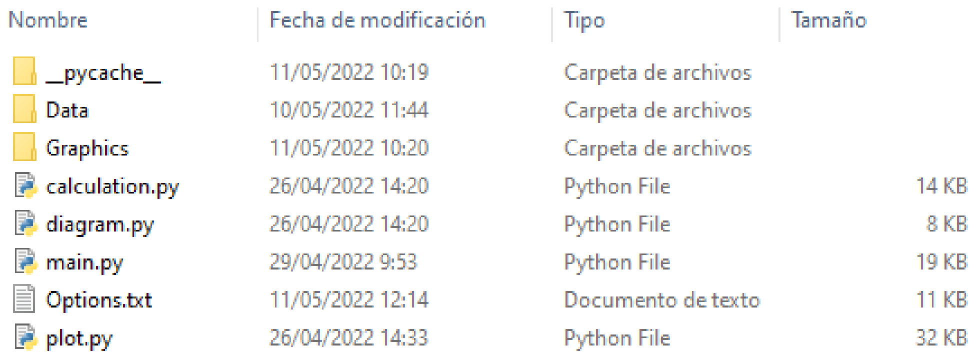
The user must locate the program files and the Options.txt file in the same folder.

It is essential to read this document and follow all the instructions and recommendations to avoid errors.

1st step: Prepare the program files

Unzip the Q_Facies.zip file in a folder with any valid name in your operating system.

The file structure should look like this:



Nombre	Fecha de modificación	Tipo	Tamaño
__pycache__	11/05/2022 10:19	Carpeta de archivos	
Data	10/05/2022 11:44	Carpeta de archivos	
Graphics	11/05/2022 10:20	Carpeta de archivos	
calculation.py	26/04/2022 14:20	Python File	14 KB
diagram.py	26/04/2022 14:20	Python File	8 KB
main.py	29/04/2022 9:53	Python File	19 KB
Options.txt	11/05/2022 12:14	Documento de texto	11 KB
plot.py	26/04/2022 14:33	Python File	32 KB

To avoid errors in the program's operation, the user should not modify the name or extension of any of these files or folders.

The Data folder contains the input data file and the Excel output file with the results.

Generated graphics are stored in the Graphics folder.

The calculation.py, diagram.py, plot.py, and main.py files contain the program. These files are profusely commented on and can be used and modified according to the copyright rules (License GPL-3).

The Options.txt file contains the name of the data file and the options to perform the calculations and graphs.

It is only necessary to run the file main.py.

The file How_to Q_Facies.pdf is the present document.

2nd step: Prepare the input data

The input data must be provided in ASCII format, with a .txt or .csv extension.

The data file must contain nine tab-separated columns.

The first column identifies each analysis by group identifier or date.

You cannot mix identifiers and dates in the same file.

Analytical results expressed in mg/l or ppm are entered in the other columns.

The decimal separator is the decimal point (e.g., 12.45)

You can enter the data directly in percentage milliequivalents (%epm) by setting the Transform variable to False.

The order of the columns from left to right is:

Group identifier – Ca – Mg – Na – K – HCO3 – CO3 – SO4 – Cl
or
Date – Ca – Mg – Na – K – HCO3 – CO3 – SO4 – Cl

Data can be grouped in two different ways:

a) As **groups**, each group can correspond, for example, to a geographical area, a type of water, different water origins such as streams, wells, etc.

For example:

Group	Ca	Mg	Na	K	HCO3	CO3	SO4	Cl
1	44	32	51	3	238	0	37	93
1	37	30	57	4	183	18	38	92
1	45	34	60	4	227	0	46	97
2	46	34	61	3	229	0	38	103
2	47	36	65	3	230	12	41	112

b) They can also be identified by **dates**, for example, when all the analyses belong to the same spring, and you want to make a study of the temporal evolution of the facies.

Date	Ca	Mg	Na	K	HCO3	CO3	SO4	Cl
1972-09-20	116	67	147	6.0	252		210	328
1976-11-04	127	44	147	7.0	246		249	284
1977-05-05	146	45	156	6.0	216		237	319
1977-10-21	168	46	121	9.0	240		219	309
1978-05-09	151	41	130	9.0	293		217	257

It is imperative to put the dates in a compatible format. For more information about the date formats, please visit:

https://www.dataindependent.com/pandas/pandas-to-datetime/

3th step: Set the analysis, graphics, and output options

The following describes the options that allow you to indicate to the program which graphics to perform, if you are working with temporal or spatial data, choose colors, etc.

2.1 General Options

2.1.1 Name of the input file (including extension .txt or .csv). It must be stored in the Data folder.

Variable Name	fname
Value	'Any valid name' (e.g., 'Data_mysite.txt')

2.1.2 Way of analyzing the dataset.
Two types of datasets are allowed: time series of a single point identified by sampling date or group sets of analysis determined by a unique identifier.

Variable Name	way
Value	'by_time' or 'by_group'

2.1.3 Transform concentrations from mg/l or ppm to percentage of meq/l
If the data in the input file is in mg/l or ppm, this variable must be set to True and to False if it is already in %meq/l.

Variable Name	Transform
Value	True or False

2.1.4 Whether to consider outlier points or not (an outlier is an observation that lies at an abnormal distance from other values in a random sample of a population)

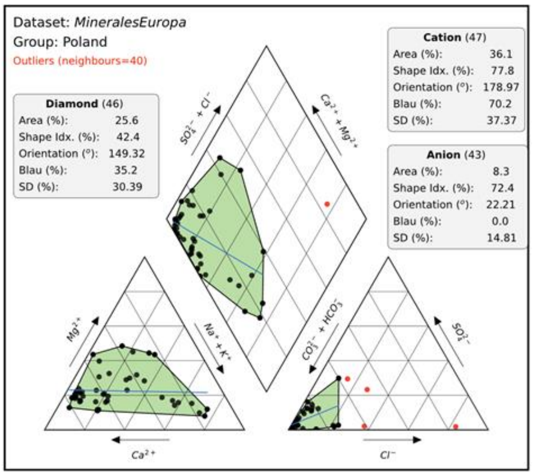
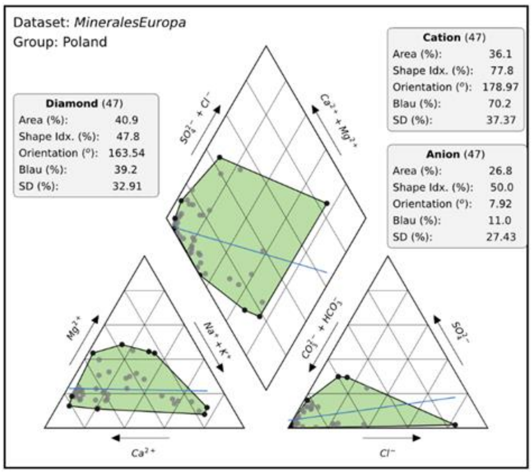
Variable Name	lof
Value	True or False

Outliers detection is computed in two steps:

a) In the first step, a preliminary detection of outliers is done with an unsupervised outlier detection method: Local Outlier Factor (LOF).

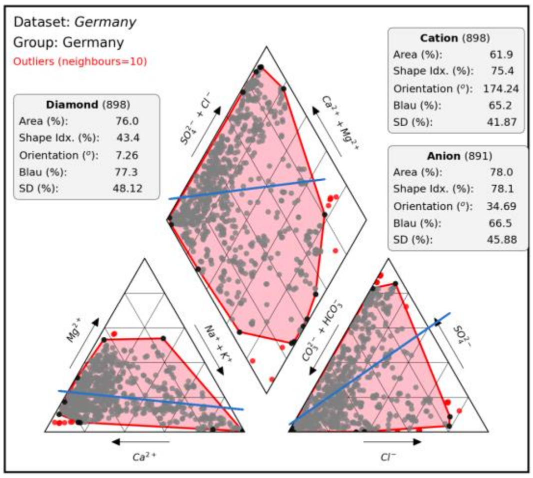
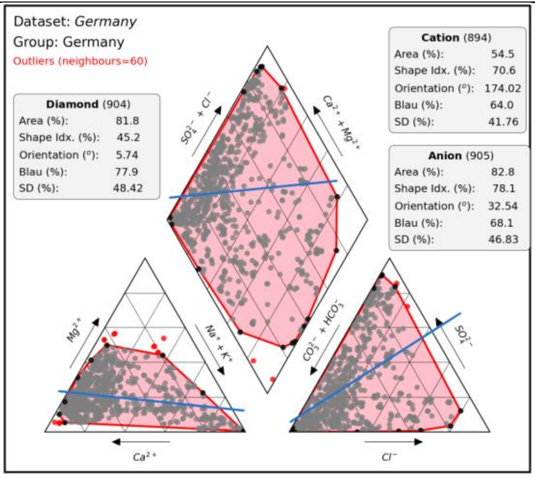
Find more information about outlier detection with Local Outlier Factor in the following link:
https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.LocalOutlierFactor.html

b) In the second step, these outliers are checked and only those outside the convex hull polygon will be discarded.

<table>
<tr><th>lof = True</th><th>lof = False</th></tr>
<tr><td>
</td><td>
</td></tr>
</table>

2.1.5 Define the number of surrounding points to compute the LOF algorithm. Please note that the lower this value, the more points will be considered outliers. If the value is larger than the number of samples provided, all samples will be used.

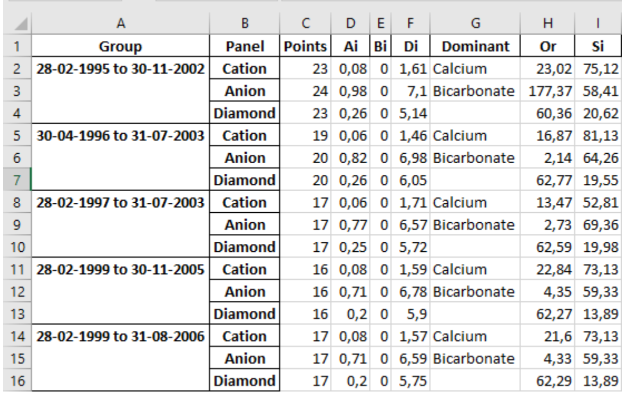
Variable Name	lof_neighbours
Value	Positive integer

<table>
<tr><th>lof_neighbours = 10</th><th>lof_neighbours = 60</th></tr>
<tr><td>
</td><td>
</td></tr>
</table>

2.1.6 Whether to create an Excel file with the output information (saved to the Data folder).

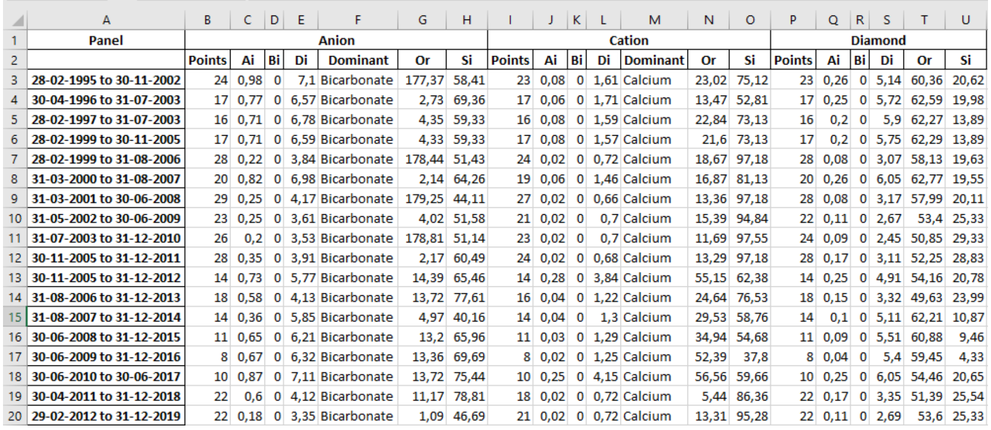
Variable Name	Excel
Value	True or False

The table in the first sheet contains each panel's indices, the facies name, and the number of points used for their calculation in long format:



	A	B	C	D	E	F	G	H	I
1	**Group**	**Panel**	**Points**	**Ai**	**Bi**	**Di**	**Dominant**	**Or**	**Si**
2	**28-02-1995 to 30-11-2002**	**Cation**	23	0,08	0	1,61	Calcium	23,02	75,12
3		**Anion**	24	0,98	0	7,1	Bicarbonate	177,37	58,41
4		**Diamond**	23	0,26	0	5,14		60,36	20,62
5	**30-04-1996 to 31-07-2003**	**Cation**	19	0,06	0	1,46	Calcium	16,87	81,13
6		**Anion**	20	0,82	0	6,98	Bicarbonate	2,14	64,26
7		**Diamond**	20	0,26	0	6,05		62,77	19,55
8	**28-02-1997 to 31-07-2003**	**Cation**	17	0,06	0	1,71	Calcium	13,47	52,81
9		**Anion**	17	0,77	0	6,57	Bicarbonate	2,73	69,36
10		**Diamond**	17	0,25	0	5,72		62,59	19,98
11	**28-02-1999 to 30-11-2005**	**Cation**	16	0,08	0	1,59	Calcium	22,84	73,13
12		**Anion**	16	0,71	0	6,78	Bicarbonate	4,35	59,33
13		**Diamond**	16	0,2	0	5,9		62,27	13,89
14	**28-02-1999 to 31-08-2006**	**Cation**	17	0,08	0	1,57	Calcium	21,6	73,13
15		**Anion**	17	0,71	0	6,59	Bicarbonate	4,33	59,33
16		**Diamond**	17	0,2	0	5,75		62,29	13,89

The table in the second sheet has the same information but in wide format:



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
1	**Panel**	**Anion**							**Cation**							**Diamond**					
2		**Points**	**Ai**	**Bi**	**Di**	**Dominant**	**Or**	**Si**	**Points**	**Ai**	**Bi**	**Di**	**Dominant**	**Or**	**Si**	**Points**	**Ai**	**Bi**	**Di**	**Or**	**Si**
3	**28-02-1995 to 30-11-2002**	24	0,98	0	7,1	Bicarbonate	177,37	58,41	23	0,08	0	1,61	Calcium	23,02	75,12	23	0,26	0	5,14	60,36	20,62
4	**30-04-1996 to 31-07-2003**	17	0,77	0	6,57	Bicarbonate	2,73	69,36	17	0,06	0	1,71	Calcium	13,47	52,81	17	0,25	0	5,72	62,59	19,98
5	**28-02-1997 to 31-07-2003**	16	0,71	0	6,78	Bicarbonate	4,35	59,33	16	0,08	0	1,59	Calcium	22,84	73,13	16	0,2	0	5,9	62,27	13,89
6	**28-02-1999 to 30-11-2005**	17	0,71	0	6,59	Bicarbonate	4,33	59,33	17	0,08	0	1,57	Calcium	21,6	73,13	17	0,2	0	5,75	62,29	13,89
7	**28-02-1999 to 31-08-2006**	28	0,22	0	3,84	Bicarbonate	178,44	51,43	24	0,02	0	0,72	Calcium	18,67	97,18	28	0,08	0	3,07	58,13	19,63
8	**31-03-2000 to 31-08-2007**	20	0,82	0	6,98	Bicarbonate	2,14	64,26	19	0,06	0	1,46	Calcium	16,87	81,13	20	0,26	0	6,05	62,77	19,55
9	**31-03-2001 to 30-06-2008**	29	0,25	0	4,17	Bicarbonate	179,25	44,11	27	0,02	0	0,66	Calcium	13,36	97,18	28	0,08	0	3,17	57,99	20,11
10	**31-05-2002 to 30-06-2009**	23	0,25	0	3,61	Bicarbonate	4,02	51,58	21	0,02	0	0,7	Calcium	15,39	94,84	22	0,11	0	2,67	53,4	25,33
11	**31-07-2003 to 31-12-2010**	26	0,2	0	3,53	Bicarbonate	178,81	51,14	23	0,02	0	0,7	Calcium	11,69	97,55	24	0,09	0	2,45	50,85	29,33
12	**30-11-2005 to 31-12-2011**	28	0,35	0	3,91	Bicarbonate	2,17	60,49	24	0,02	0	0,68	Calcium	13,29	97,18	28	0,17	0	3,11	52,25	28,83
13	**30-11-2005 to 31-12-2012**	14	0,73	0	5,77	Bicarbonate	14,39	65,46	14	0,28	0	3,84	Calcium	55,15	62,38	14	0,25	0	4,91	54,16	20,78
14	**31-08-2006 to 31-12-2013**	18	0,58	0	4,13	Bicarbonate	13,72	77,61	16	0,04	0	1,22	Calcium	24,64	76,53	18	0,15	0	3,32	49,63	23,99
15	**31-08-2007 to 31-12-2014**	14	0,36	0	5,85	Bicarbonate	4,97	40,16	14	0,04	0	1,3	Calcium	29,53	58,76	14	0,1	0	5,11	62,21	10,87
16	**30-06-2008 to 31-12-2015**	11	0,65	0	6,21	Bicarbonate	13,2	65,96	11	0,03	0	1,29	Calcium	34,94	54,68	11	0,09	0	5,51	60,88	9,46
17	**30-06-2009 to 31-12-2016**	8	0,67	0	6,32	Bicarbonate	13,36	69,69	8	0,02	0	1,25	Calcium	52,39	37,8	8	0,04	0	5,4	59,45	4,33
18	**30-06-2010 to 30-06-2017**	10	0,87	0	7,11	Bicarbonate	13,72	75,44	10	0,25	0	4,15	Calcium	56,56	59,66	10	0,25	0	6,05	54,46	20,65
19	**30-04-2011 to 31-12-2018**	22	0,6	0	4,12	Bicarbonate	11,17	78,81	18	0,02	0	0,72	Calcium	5,44	86,36	22	0,17	0	3,35	51,39	25,54
20	**29-02-2012 to 31-12-2019**	22	0,18	0	3,35	Bicarbonate	1,09	46,69	21	0,02	0	0,72	Calcium	13,31	95,28	22	0,11	0	2,69	53,6	25,33

2.2 Time Analysis Options

Time analysis options are only considered when way is set to 'by_time'.

2.2.1 Specify the date format.

Variable Name	datetime_format
Value	Any valid date format (e.g., '%d/%m/%Y')

Find more information about date and time format in the following link: https://www.dataindependent.com/pandas/pandas-to-datetime/

2.2.2 Select the width of the rolling window to create temporal groups. It has to be greater than three, which is the minimum group size.

Variable Name	window_size
Value	Positive integer

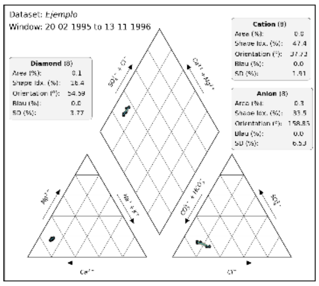
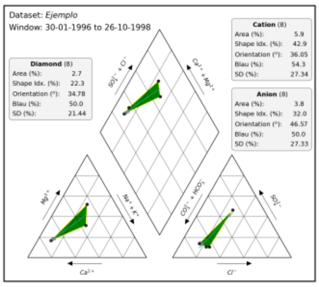
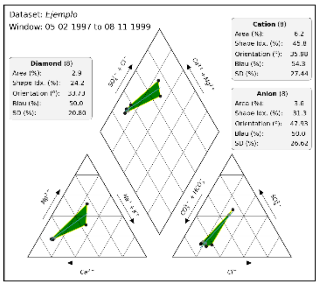
2.2.3 Size of the step for the rolling window.

Variable Name	step_length
Value	Positive integer

2.2.4 Whether to resample the time series to regular intervals. If True, the time series will be downsampled or upsampled to the frequency specified in 'resample_interval'.

Variable Name	resample
Value	True or False

	1st step	2nd step	3rd step

window_size = 8
step_length = 4
resample = False
			

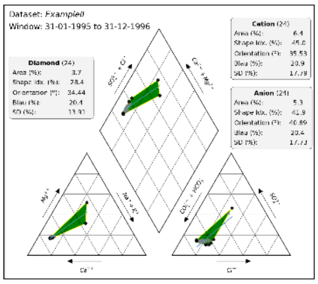
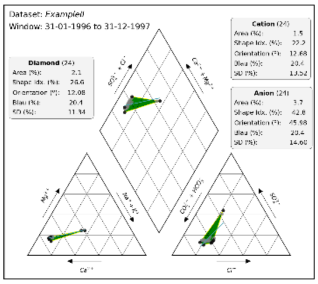
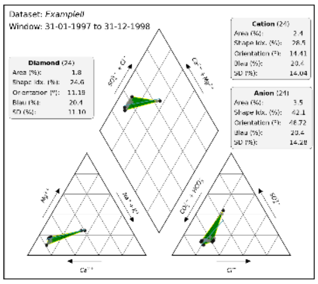
2.2.5 Frequency of time series intervals (only considered when resample=True) Note that NaN values will be assigned to empty date rows.

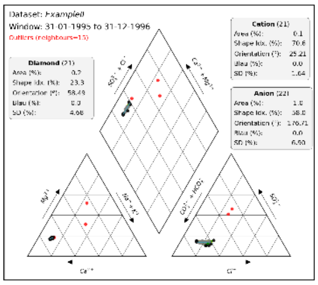
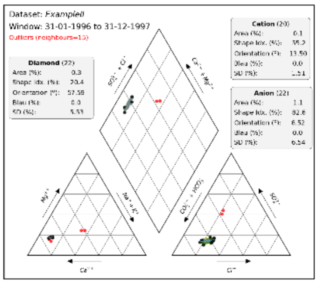
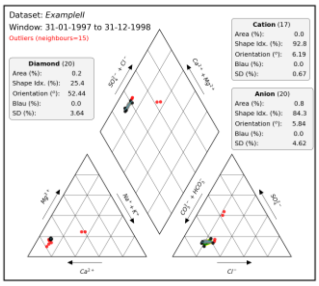
Variable Name	resample_interval
Value	'yearly'/'Y', 'monthly'/'M', 'weekly'/'W', 'daily'/'D'

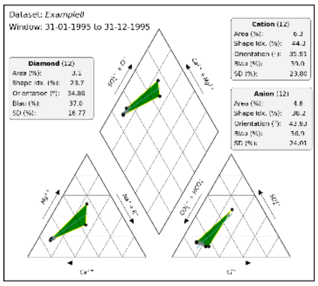
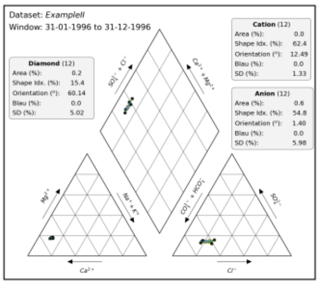
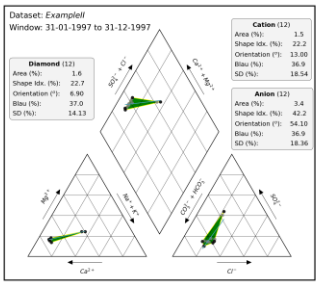
Other options can be found at:
https://pandas.pydata.org/pandas-docs/stable/user_guide/timeseries.html#offset-aliases

A custom frequency can be defined by preceding the acronym by a positive integer (e.g., every-17-days = '17D')

	1st step	2nd step	3rd step

window_size = 24
step_length = 12
resample = True
aggregation='first'
lof=False
			

window_size = 24
step_length = 12
resample = True
aggregation='first'
lof=True
			

window_size = 12
step_length = 6
resample = True
aggregation='first'
lof=False
			

2.2.6 Resampling aggregation method. How to aggregate values belonging to the same time interval (e.g., we want to resample the time series to monthly intervals and we have several samples for a given month or months). The options are mean aggregating or either take the last or first value.

Variable Name	aggregation
Value	'mean','last','first'

2.3 Graphic Options

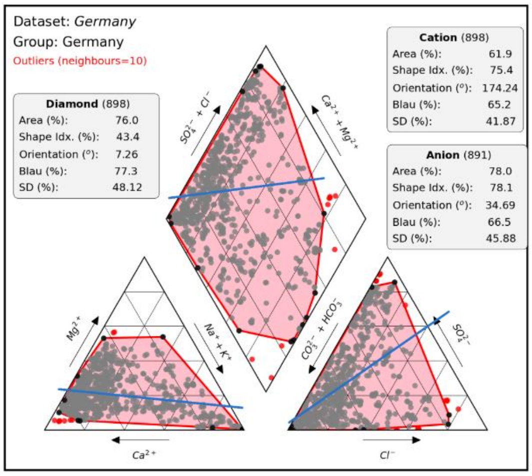
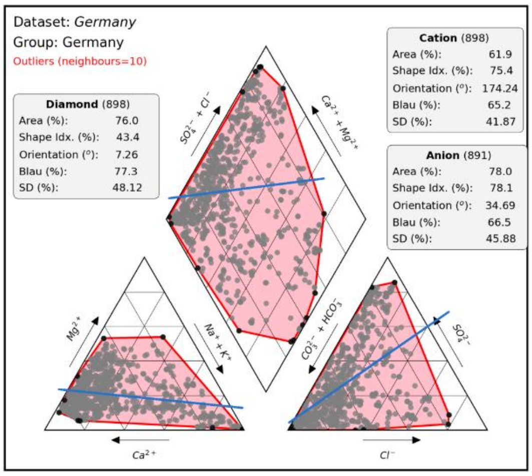
All the output plots will be stored in the 'Graphics' folder. The color scheme to be used in the different elements can be specified in two ways:

a) Following Matplotlib nomenclature (https://matplotlib.org/stable/gallery/color/named_colors.html) E.g., 'blue'

b) Following HEX nomenclature. E.g. '#BB2EB3'

2.3.1 Plot outliers points.
Outliers are not considered for indices calculations.

Variable Name	plot_outliers
Value	True or False

plot_outliers = True	plot_outliers = False
	

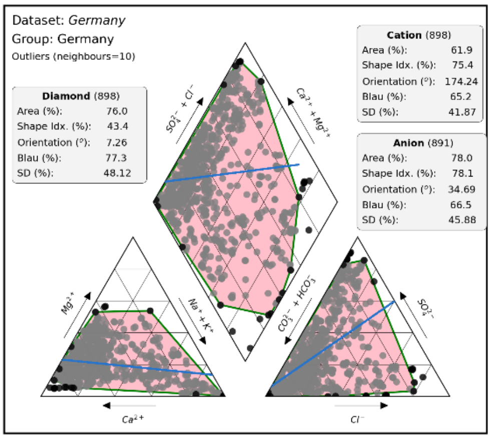
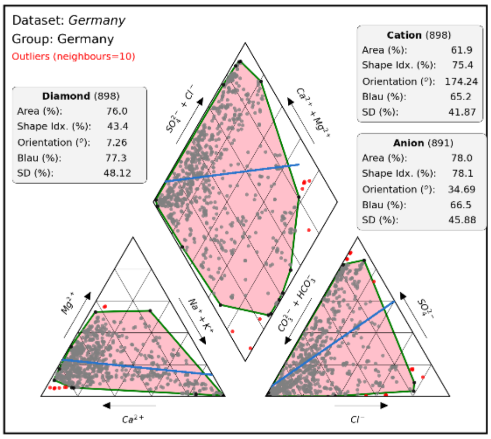
2.3.2 Color of outliers points.

Variable Name	color_outliers
Value	Color name (e.g., 'red') or HEX code (e.g., '#BB2EB3')

2.3.3 Size of all plotted points.

Variable Name	point_size
Value	Positive integer

color_outliers = 'black'
point_size = 15
	color_outliers = 'red'
	point_size = 4

2.3.4 Points transparency.

Variable Name	point_transparency
Value	Real number between 0 (transparent) and 1 (opaque)

2.3.5 Color of the points that conform the convex hull.

Variable Name	color_outter_points
Value	Color name (e.g., 'red') or HEX code (e.g., '#BB2EB3')

2.3.6 Color of the points that lay within the convex hull.

Variable Name	color_inner_points
Value	Color name (e.g., 'red') or HEX code (e.g., '#BB2EB3')

2.3.7 Regression line width.

Variable Name	line_lw
Value	Real number

2.3.8 Regression line color.

Variable Name	line_color
Value	Color name (e.g., 'red') or HEX code (e.g., '#BB2EB3')

2.3.9 Convex hull transparency.

Variable Name	pol_transparency
Value	Real number between 0 (transparent) and 1 (opaque)

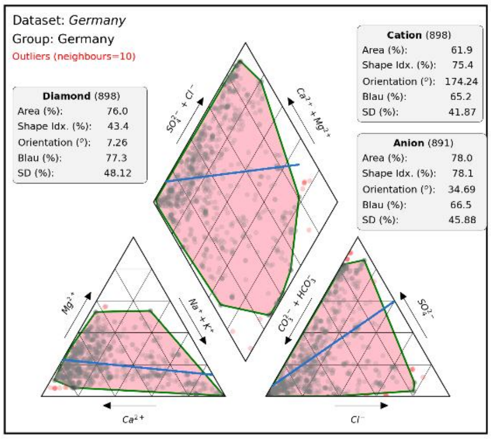
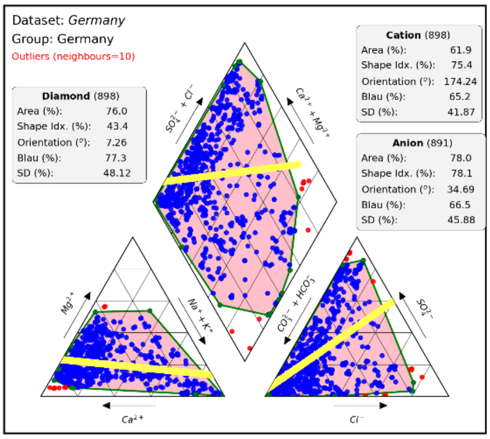
2.3.10 Line width of the convex hull polygon.

Variable Name	pol_lw
Value	Real number

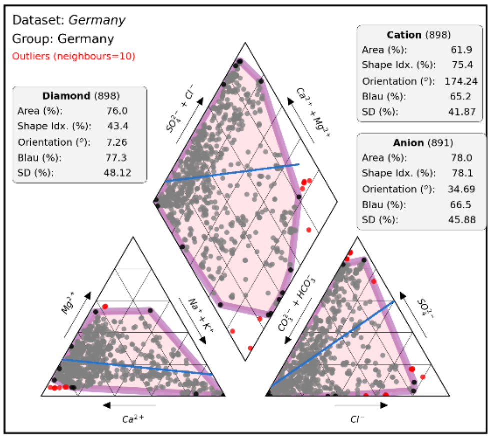
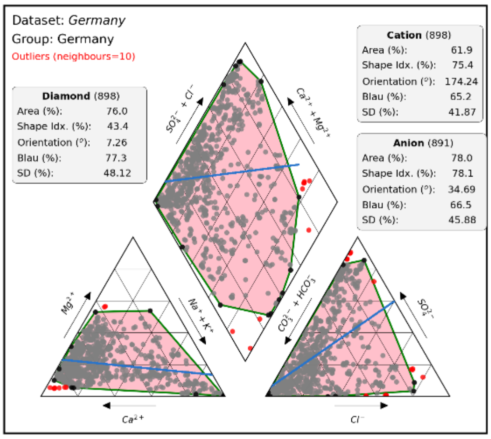
2.3.11 Line color of the convex hull polygon.

Variable Name	pol_ec
Value	Color name (e.g., 'red') or HEX code (e.g., '#BB2EB3')

color_outter_points = 'black'
point_transparency = 0.2
color_inner_points = 'grey'
line_lw = 1
line_color = '#2774CC'
	color_outter_points = 'green'
	point_transparency = 0.9
	color_inner_points = 'blue'
	line_lw = 4
	line_color = '#FFFF54'

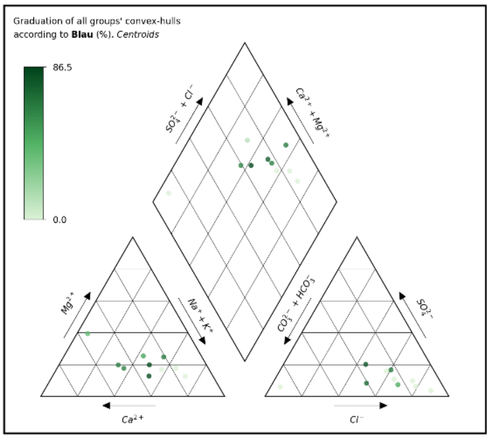
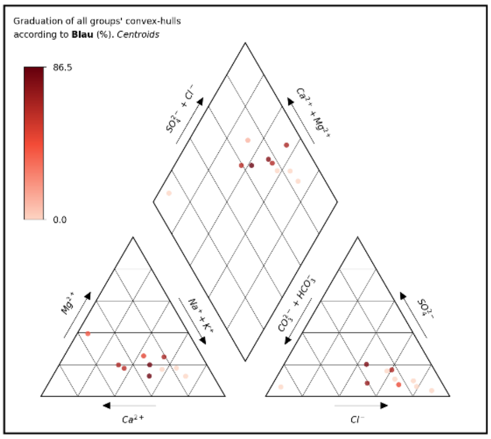
pol_color = 'blue'
pol_transparency = 0.4
pol_lw = 4
pol_ec = 'purple'
	pol_color = 'green'
	pol_transparency = 1
	pol_lw = 1
	pol_ec = 'green'

2.3.12 Color palette to be used in the figures.

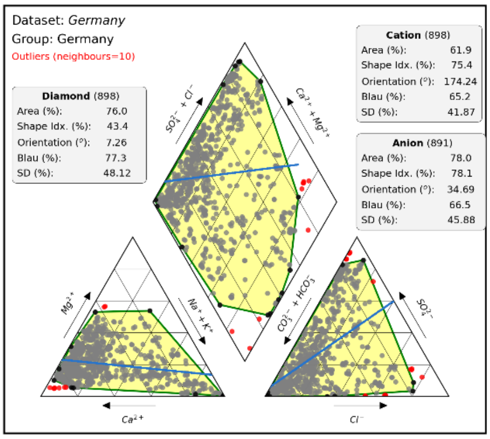
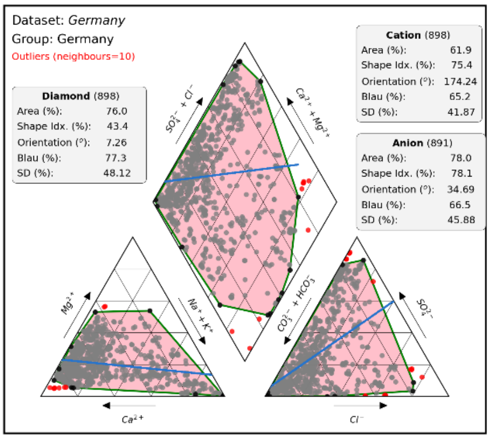
Variable Name	cmap_name
Value	Palette name (e.g., 'Blues')

Examples of color palettes: 'Greens', 'Greys', 'Blues', 'Reds', 'YlOrBr', 'Spectral', etc.
Many more options are available at:
https://matplotlib.org/3.1.1/tutorials/colors/colormaps.html

cmap_name = 'greens'	cmap_name = 'Reds'
	

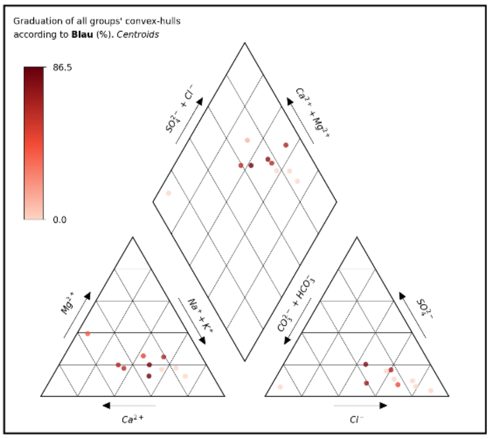
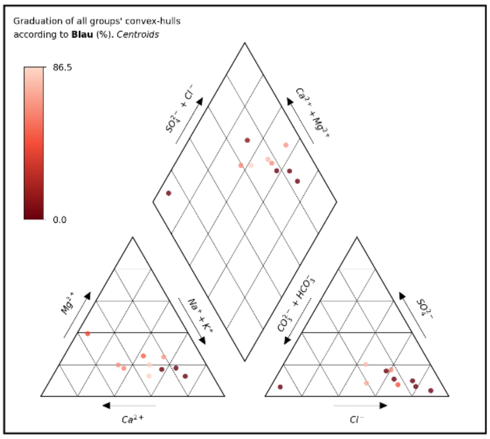
2.3.13 Optional color to plot the figures in the 'Graphics' folder. Either a specific color or a color palette can be used.

Variable Name	folder_color
Value	Color or palette name (e.g., 'green', 'Spectral')

folder_color = '#FFFF98'	folder_color = 'pink'
	

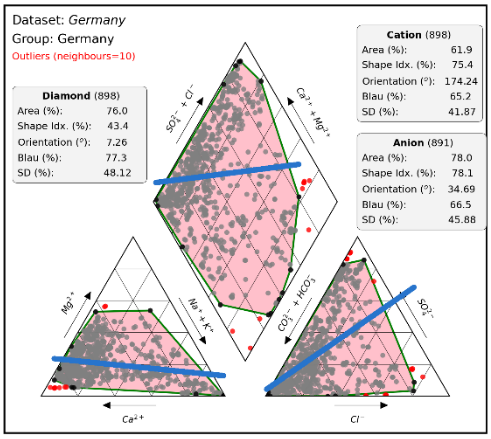
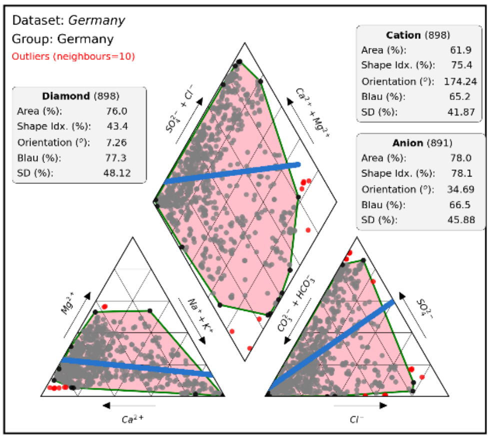
2.3.14 Whether to reverse the color palette or not.

Variable Name	cmap_reversed
Value	True or False

cmap_reversed = False	cmap_reversed = True
	

2.3.15 Clip regression lines at panels' edges.

Variable Name	clip_lines
Value	True or False

clip_lines = False	clip_lines = True
	

2.3.16 Figure resolution (in dots per inch)

Variable Name	dpi
Value	Positive integer

2.3.17 Output graphic format: 'jpg', 'png', 'svg', 'pdf', 'ps'

Variable Name	extension_graph
Value	File extension (e.g., 'svg')

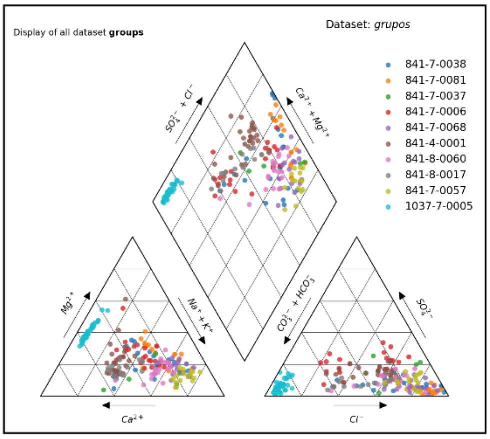
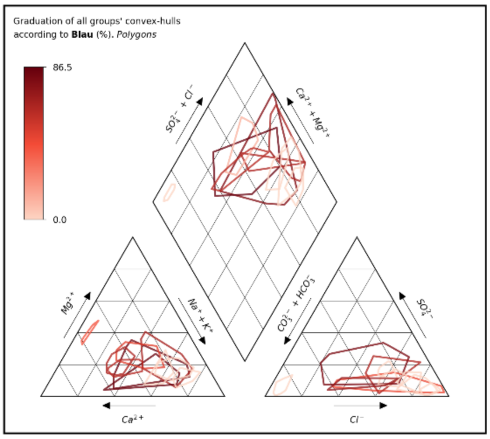
2.4 Graphics

2.4.1 Whether to create a Piper-diagram per group with all the indices information.

Variable Name	Figs
Value	True or False

2.4.2 Whether to create a Piper-diagram including all the convex hulls of the dataset for a given index.

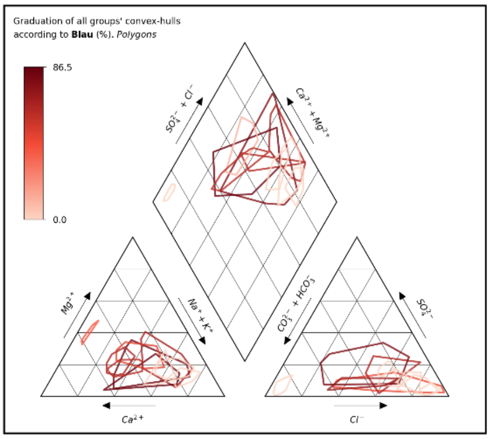
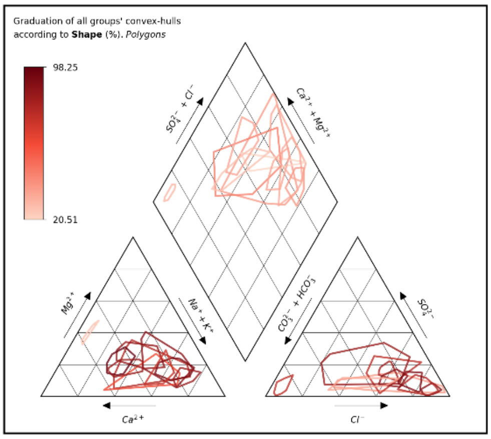
Variable Name	overlap_fig
Value	True or False

Figs = True	overlap_fig = True
	

2.4.3 Index represented by graduated colors of the convex hulls in the 'overlap_fig'.

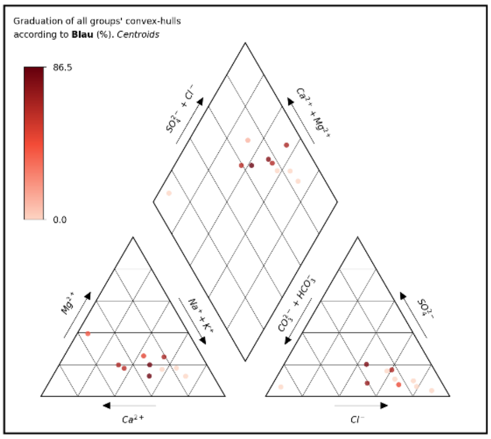
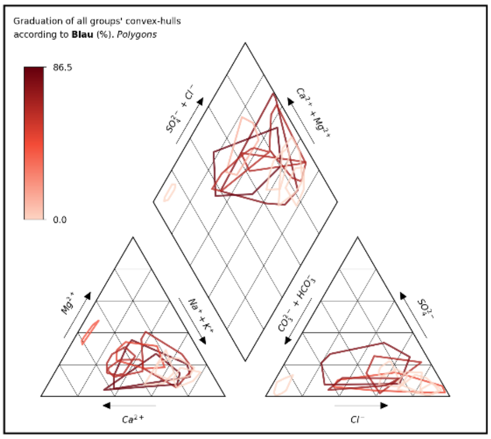
Variable Name	overlap_var
Value	'Area', 'Shape', 'Blau', 'Angle', 'Dispersion' or 'Time'

Note that 'Time' is only available when way = 'by_time'.

overlap_var = 'Blau'	overlap_var = 'Shape'
	

2.4.4 Whether to represent the polygon of the convex hull or its centroid in the 'overlap_fig'.

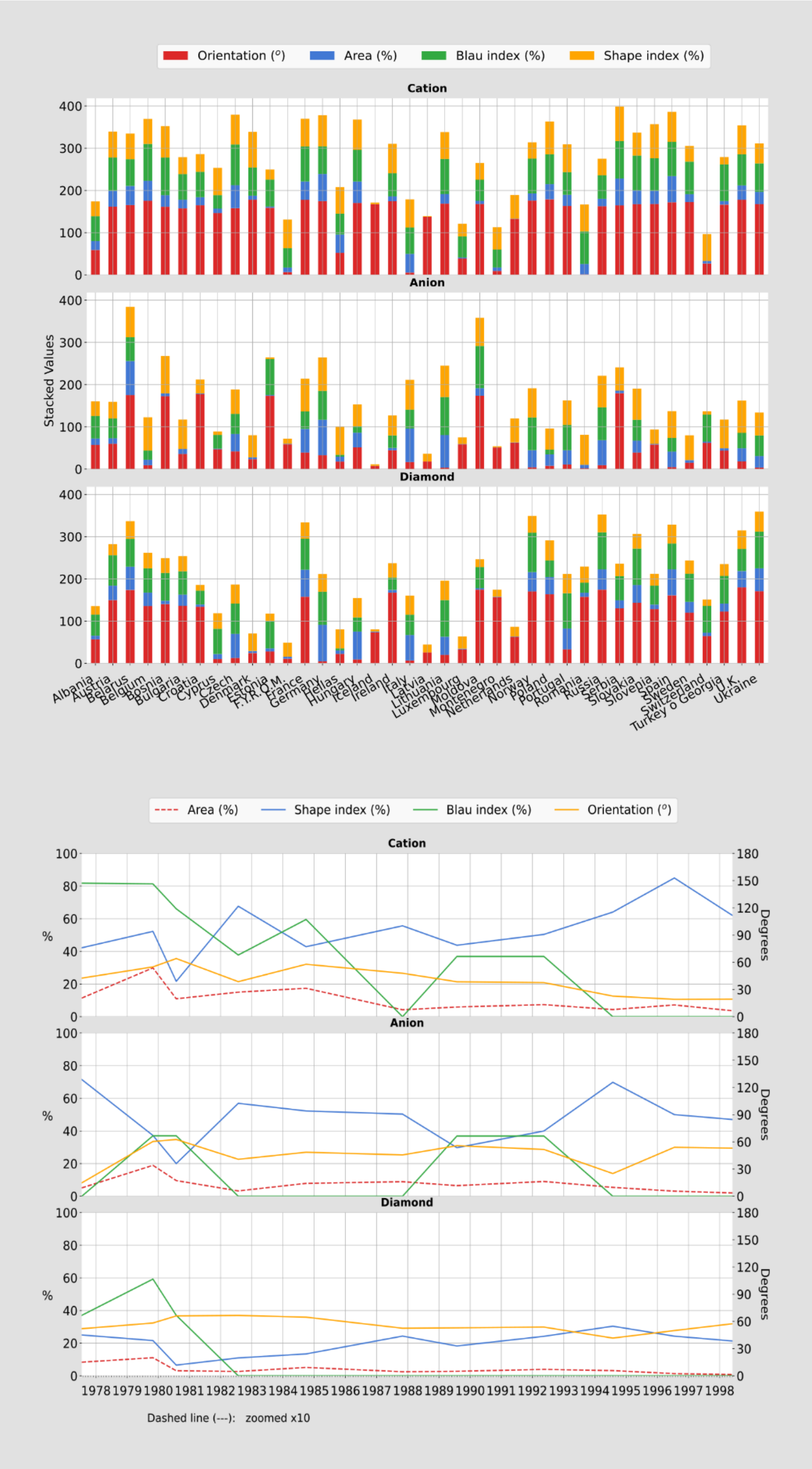
Variable Name	overlap_way
Value	'centroids' or 'polygons'

overlap_way = 'centroids'	overlap_way = 'polygons'
	

2.4.5 Create a summary figure composed by three plots (one per panel) with the information of all the indices.

When way='by_time', each plot shows the indices' temporal evolution.
When way='by_groups', each plot shows a stacked bars histogram.

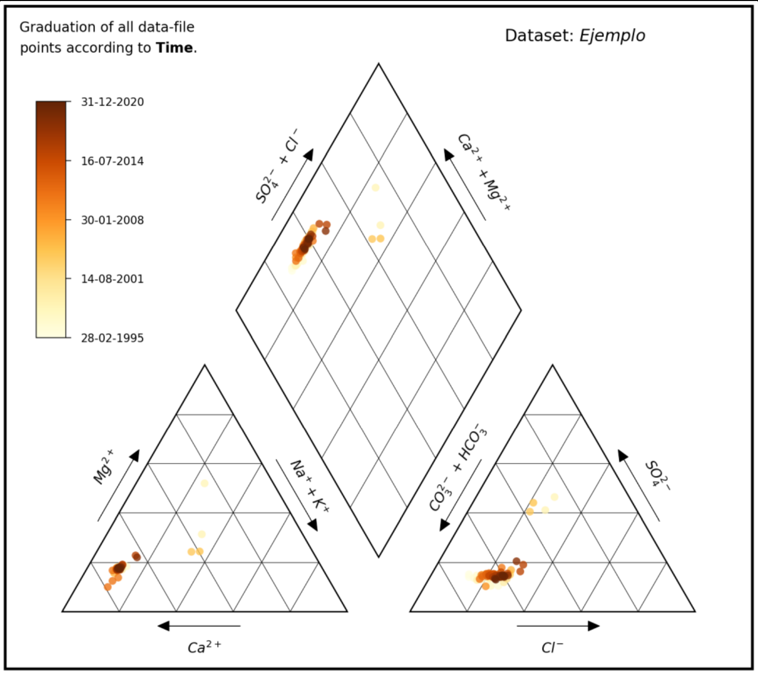
Variable Name	evolution_fig
Value	True or False


Orientation (°)
Area (%)
Blau index (%)
Shape index (%)
Cation
Anion
Diamond
Stacked Values
0
100
200
300
400
Albania
Austria
Belarus
Belgium
Bosnia
Bulgaria
Croatia
Cyprus
Czech
Denmark
Estonia
F.Y.R.O.M.
France
Germany
Hellas
Hungary
Iceland
Ireland
Italy
Latvia
Lithuania
Luxembourg
Moldova
Montenegro
Netherlands
Norway
Poland
Portugal
Romania
Russia
Serbia
Slovakia
Slovenia
Spain
Sweden
Switzerland
Turkey
Georgia
U.K.
Ukraine
Area (%)
Shape index (%)
Blau index (%)
Orientation (°)
Cation
Anion
Diamond
%
Degrees
0
20
40
60
80
100
0
30
60
90
120
150
180
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
Dashed line (---): zoomed x10

2.4.6 Create a figure with all the points graduated by date.
Only available when way='by_time'.
cmap_name will be used as color palette.

Variable Name	plot_all
Value	True or False

plot_all = True



2.5 Other Options

2.5.1 Ignore all execution warnings.

Variable Name	ignore_warnings
Value	True or False

4th step: run the program

Once the data input file and the Options.txt file are prepared, just load the main.py script and run it. If everything has been done correctly, you will find an Excel spreadsheet with the results in the Data folder and the selected plots in the Graphics folder.

It is convenient to save the Options.txt file with another name as a backup to use or modify it if necessary. Q-Facies always reads the Options.txt file for the ongoing iteration.

And enjoy!