```python
import numpy as np
import polykriging as pk
import matplotlib.pyplot as plt
import matplotlib

matplotlib.use('Qt5Agg')

path = pk.io.choose_directory(titl="geo files")
cwd = pk.io.cwd_chdir(path)  # ./local_coordinate
filenames = pk.io.filenames(path, ".geo")

filenames = filenames[6:9] + filenames[11:14] + filenames[17:20]

# Plot the area change
font = {'family': 'times new roman',
        'weight': 'normal',
        'size': 8}
plt.rc('font', **font)

margins = {'left': 0.13, 'right': 0.95, 'top': 0.96, 'bottom': 0.14}
ax_dict = {}
for i in np.arange(0, 3):
    fig = plt.figure(figsize=(8 / 2.54, 6 / 2.54), dpi=400)
    ax_dict["ax" + str(i)] = fig.add_subplot(111)
    ax_dict["ax" + str(i)].tick_params(which='both', direction='in', )
    ax_dict["ax" + str(i)].set_xlabel("$x$ (mm)")
    ax_dict["ax" + str(i)].set_xlim([0, 12.5])
    ax_dict["ax" + str(i)].xaxis.set_minor_locator(plt.MultipleLocator(0.5))
    plt.subplots_adjust(**margins)

geo_warp = {}
labels = {"warp_32.geo": "warp 1", "warp_33.geo": "
```

```
32 warp 2", "warp_34.geo": "warp 3",
33           "warp_38.geo": "warp 4", "warp_39.geo": "
   warp 5", "warp_40.geo": "warp 6",
34           "warp_44.geo": "warp 7", "warp_45.geo": "
   warp 8", "warp_46.geo": "warp 9"}
35 cir_lst, area_lst, ar_list = [], [], []
36 for i, filename in enumerate(filenames):
37     geo = pk.pk_load(filename)
38     print(geo.iloc[:, 0].shape)
39     # update circularity
40     geo.iloc[:, 5] = 4 * np.pi * geo.iloc[:, 0] / (
   geo.iloc[:, 1] ** 2)
41     # Add aspect ratio
42     ar = 0.25 * np.pi * geo["Width"] ** 2 / geo["Area
   "]  # Definition 1
43     # ar = geo["Width"] / geo["Height"]  # Definition
    2
44     geo["Aspect ratio"] = ar  # Add aspect ratio to
   dataframe geo
45     geo_warp[filename] = geo
46
47     ax_dict["ax0"].plot(geo["centroidX"], geo["
   Circularity"], linewidth=0.5, label=labels[filename],
48                         alpha=0.7)  # circularity
49     cir_lst.append(geo_warp[filename].iloc[:, [-4, -5
   ]].values)
50
51     ax_dict["ax1"].plot(geo["centroidX"][3:-3], geo["
   Area"][3:-3], linewidth=0.5, label=labels[filename],
52                         alpha=0.7)  # area
53     area_lst.append(geo_warp[filename].iloc[:, [-4, 0
   ]].values)
54
55     ax_dict["ax2"].plot(geo["centroidX"][3:-3], ar[3
   :-3], linewidth=0.5, label=labels[filename],
56                         alpha=0.7)  # aspect ratio
57     ar_list.append(geo_warp[filename].iloc[:, [-4, -1
```

```
57 ]].values)
58
59 """ Average the circularity """
60 cir_avg, cir_interp = pk.plot.xy_interp(*cir_lst, num
   =100, raw=True)
61 ax_dict["ax0"].plot(cir_avg[:, 0], cir_avg[:, 1],
   "--", linewidth=1, label="Average", color="black")
62 ax_dict["ax0"].set_ylabel("Circularity")
63 ax_dict["ax0"].set_ylim([0.2, 0.3])
64 ax_dict["ax0"].yaxis.set_minor_locator(plt.
   MultipleLocator(0.1))
65 ax_dict["ax0"].legend(ncol=3, frameon=False, fontsize
   =7)
66
67 """ Average the area """
68 area_avg, area_interp = pk.plot.xy_interp(*area_lst,
   num=100, raw=True)
69 ax_dict["ax1"].plot(area_avg[3:-3, 0], area_avg[3:-3
   , 1], "--", linewidth=1, label="Average", color="
   black")
70 ax_dict["ax1"].set_ylabel("Area (mm$ ^2$)")
71 ax_dict["ax1"].set_ylim([1, 1.5])
72 ax_dict["ax1"].yaxis.set_minor_locator(plt.
   MultipleLocator(0.05))
73 ax_dict["ax1"].legend(ncol=3, frameon=False, fontsize
   =7)
74
75 """ Average the aspect ratio """
76 ar_avg, ar_interp = pk.plot.xy_interp(*ar_list, num=
   100, raw=True)
77 ax_dict["ax2"].plot(ar_avg[3:-3, 0], ar_avg[3:-3, 1
   ], "--", linewidth=1, label="Average", color="black")
78 ax_dict["ax2"].set_ylabel("Aspect ratio")
79 ax_dict["ax2"].set_ylim([6.5, 9.5])
80 ax_dict["ax2"].yaxis.set_minor_locator(plt.
   MultipleLocator(0.25))
81 ax_dict["ax2"].legend(ncol=3, frameon=False, fontsize
```

```
 81 =7)
 82 plt.show()
 83
 84 # """ deviation of geometry features from average
     """
 85 # # Circularity
 86 # # clear all plots
 87 # plt.clf()
 88 # cir_delta = np.zeros_like(cir_interp.transpose())
 89 # cir_delta[:, 0] = cir_avg[:, 0]
 90 # cir_delta[:, 1:] = cir_avg[:, 1].reshape(-1, 1)
 91 # cir_delta[:, 1:] = cir_interp.transpose()[:, 1
    :] - cir_delta[:, 1:]
 92 #
 93 # plt.plot(cir_delta[:, 0], cir_delta[:, 1:])
 94 # plt.xlabel("$\delta Cir$")
 95 # plt.hist(cir_delta[:, 1:].flatten(), bins=50)
 96 # # abs() and test if it is normal distribution
 97 #
 98 # # Aspect ratio
 99 # ar_delta = np.zeros_like(ar_interp.transpose())
100 # ar_delta[:, 0] = ar_avg[:, 0]
101 # ar_delta[:, 1:] = ar_avg[:, 1].reshape(-1, 1)
102 # ar_delta[:, 1:] = ar_interp.transpose()[:, 1:] -
    ar_delta[:, 1:]
103 #
104 # plt.plot(ar_delta[:, 0], ar_delta[:, 1:])
105 # plt.xlabel("$\delta Cir$")
106 # plt.hist(ar_delta[:, 1:].flatten(), bins=50)
107 # # abs() and test if it is normal distribution
108 #
109 # # Area
110 # area_delta = np.zeros_like(area_interp.transpose
    ())
111 # area_delta[:, 0] = area_avg[:, 0]
112 # area_delta[:, 1:] = area_avg[:, 1].reshape(-1, 1)
113 # area_delta[:, 1:] = area_interp.transpose()[:, 1
```

```
:] - area_delta[:, 1:]
#
# plt.plot(area_delta[:, 0], area_delta[:, 1:])
# plt.xlabel("$\delta Cir$")
# plt.hist(area_delta[:, 1:].flatten(), bins=50)
# plt.hist(area_interp[1:, :].flatten(), bins=50)
# # abs() and test if it is normal distribution

```