Question 1: Integrated -omics data analysis of the Ras/Raf/MAPK pathway

Purpose: this is to test data collecting and cleaning strategy.

Descriptions: using public database and SRA if necessary (in case SRA data too large, you don't have to include them), please make cross populations (Asian, Caucasians, and African etc.) comparison of genes mutation rates, transcription and epigenetics variations.

Requirements:

1. List the gene name/ID of your analysis, and briefly describe how you identify these genes. We require the genes in human only.
2. Describe where you download and how you organize and clean the related -omics data.
3. Briefly describe your analyzing flow, and the cross comparison results.

Question 2: Develop an *ad hoc* KDE clustering algorithm to identify the cluster falling into a specified area.

Purpose: this is to test algorithmic data processing.

Descriptions: 200 sets data are provided. Each contains two columns, "sim" and "len_diff". The below figure presents three examples: upper panel shows their X-Y plotting (1a, 2a, 3a), and lower panel is the corresponding 3D surface plotting (1b, 2b, 3b) of the kernel density estimations. In the X-Y plots, the data within the black dash-line circle is the cluster to be identified, which includes most of the data points. This circled cluster corresponds to the major peak (high density area) as seen in 3D surface KDE plots. **NB: there are data points in upper area on the X-Y plots as well, which are NOT included in the circled area.** Your task is to identify **a circular border** enclosing data in this big cluster lying along the x-axis (len_diff) with low y (sim) values. There may (like 1b and 2b) or may not (like 3b) be another small cluster (low peak) identified in the upper X-Y plotting area.

Requirements:

1. Algorithm must be 2D KDE based.
2. Must include a step to optimize smoothing parameters (bin size, grid size etc).
3. Please apply your algorithm to all the 200 data sets.
4. Final results must be **the circular border lines.**

Hints:

1. Essentially, you only need to recognize the big cluster lying along the x-axis.

2. Size (number of data points) of the cluster to be identified is always a large cluster, and it at least includes more than 50% of the total data points. Most of big clusters include over 90% of the points.
3. The big cluster always has highest density.
4. Most likely, the big cluster lies in the area with len_diff (x-axis) 0-500, and sim (y-axis) in between 30 to ~45.
5. In your parameter optimizing, you can use available package/algorithm. You can refer to surface smoothing algorithm.

Figure See Q2_figure.