**Enhancing Cell Cycle Analysis: A Combined Approach Integrating Rolling Average Smoothing and Autophagy Tracking**

# Import necessary libraries

import pandas as pd  # For data manipulation using DataFrames

import numpy as np  # For numerical operations

import matplotlib.pyplot as plt  # For creating plots

import scipy.ndimage as ndimage  # For applying the rolling average smoothing

# --- Data Loading and Preprocessing ---

# Define the path to the CSV file containing the data

file\_path = "Day2\_PC\_006010-9\_Analysis2\_byTrackID.csv"

# Load the data from the CSV file into a pandas DataFrame

df = pd.read\_csv(file\_path)

# Sort the data by TrackID and then by Time within each TrackID

# This ensures that the data is in the correct order for analysis and plotting

df = df.sort\_values(by=["TrackID", "Time"])

# --- Smoothing Function ---

# Define a function to apply rolling average smoothing to a data series

def rolling\_average\_smoothing(data, window\_size=5):

    """

    This function smooths a data series using a rolling average.

    It helps to reduce noise and highlight trends in the data.

    Parameters:

    data (numpy array): The input data to be smoothed.

    window\_size (int): The size of the rolling window (number of data points to average). Default is 5.

    Returns:

    numpy array: The smoothed data.

    """

    # Use the uniform\_filter1d function from scipy.ndimage to apply the rolling average

    return ndimage.uniform\_filter1d(data, size=window\_size, mode="nearest")

# --- Plotting ---

# Get a list of unique Track IDs from the DataFrame

unique\_tracks = df["TrackID"].unique()

# Loop through each unique Track ID

for track in unique\_tracks:

    # Handle potential NaN (Not a Number) values in TrackID

    # If the current track is NaN, skip it to avoid errors

    if np.isnan(track):

        continue

    # Extract data for the current track

    track\_data = df[df["TrackID"] == track]

    # Apply rolling average smoothing to the 'ClassPrediction' column for the current track

    smoothed\_predictions = rolling\_average\_smoothing(track\_data["ClassPrediction"].values)

    # Create a new figure for the plot

    plt.figure(figsize=(10, 6))  # Set the figure size (width, height) in inches

    # Plot the smoothed Class Prediction vs. Time for the current track

    plt.plot(track\_data["Time"], smoothed\_predictions,

             marker='o', linestyle='-',  # Set marker and line style

             label=f"Track ID {track} (Smoothed)")  # Add a label for the legend

    # Set plot labels and title

    plt.xlabel("Time")

    plt.ylabel("Class Prediction")

    plt.title(f"Track ID: {track} - Cell Cycle Progression (Smoothed)")

    # Customize plot appearance

    plt.xticks(rotation=45)  # Rotate x-axis labels for better readability

    plt.grid()  # Add a grid to the plot

    plt.legend()  # Display the legend

    plt.tight\_layout()  # Adjust layout to prevent overlapping elements

    # Show the plot

    plt.show()