Supplementary Material (Document Title: S3)

Article Title:

Systematizing Data Preparation in Smart Manufacturing via Axiomatic Design: A Toolkit Integrating GUI and Agentic AI

Authors:

Angkush Kumar Ghosh, Saman Fattahi, Yu Kogawara, Bahman Azarhoushang, Takuya Okamoto, and Sharifu Ura

Description:

This document details the application scenario of the Agentic AI-based toolkit by providing step-by-step conversational logs, tool invocation traces, and corresponding visual screenprints.

Table S3-1. Conversational logs and tool invocation traces for data ingestion and exploration.

<table>
<tr><td>User</td><td>Show me the content of this file: “D:\1-1B.txt”.</td></tr>
<tr><td>Agent</td><td>Here are the first 15 lines of the file “D:\1-1B.txt”:
Date: 11.02.2025
Time: 12:04
Comment:
time [s] Fx Fy Fz Fres [N]
0,000000 -0,005206 0,639062 -0,005206 0,639105
0,000100 0,188074 0,639062 0,639062 0,923132
0,000200 0,252501 0,574636 0,316928 0,703140
0,000300 0,188074 0,703489 0,574636 0,927618
0,000400 0,188074 1,090050 0,445782 1,192603
0,000500 0,252501 0,703489 0,639062 0,983389
0,000600 0,252501 0,832343 0,445782 0,977381
0,000700 0,059221 0,832343 0,381355 0,917460
0,000800 -0,005206 0,832343 0,252501 0,869815
0,000900 0,188074 0,574636 0,639062 0,879761
0,001000 0,123648 0,639062 0,381355 0,754401

The file contains metadata at the top, followed by tab-separated data columns. The data uses a comma as the decimal separator. Let me know if you need further assistance with this file!</td></tr>
<tr><td colspan="2">Backend (Tool Invocation Trace):
[Agent] Calling tool: show_raw_content
[Agent] With arguments: {‘file_path’: ‘D:\\1-1B.txt’, ‘num_lines’: 15}
Executing tool: show_raw_content
Arguments: {‘file_path’: ‘D:\\1-1B.txt’, ‘num_lines’: 15}</td></tr>
</table>

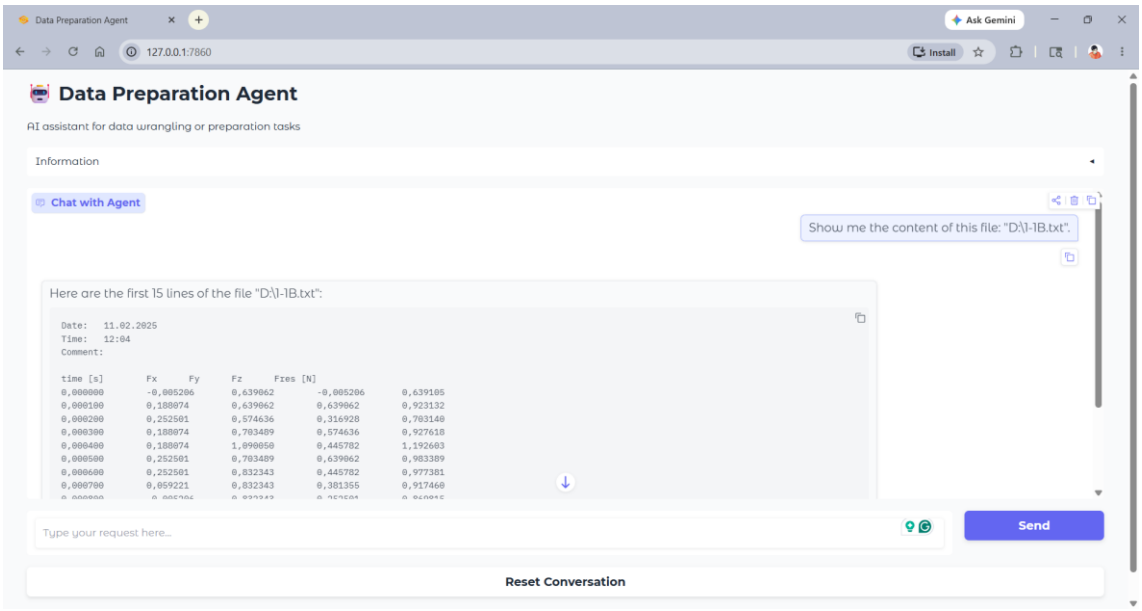


Figure S3-1. Screenprint related to Table S3-1.

Table S3-2. Conversational logs and tool invocation traces for data parsing.

User	As you identified and I also see, I do not need the first three rows (metadata). Also, only need the first column and the fourth column. Comma decimal separator should also be corrected. Can you do this and show me the modified file?
Agent	The modified file has been created with the specified changes. The first three rows (metadata) have been removed, and only the first and fourth columns have been retained. The comma decimal separator has also been corrected. You can find the modified file here: D:\modified_1-1B_parsed_20260516-165516-113152.txt Let me know if you need any further assistance!
Backend (Tool Invocation Trace): [Agent] Calling tool: save_parsed_data [Agent] With arguments: {'file_path': 'D:\\1-1B.txt', 'skip_rows': 3, 'header_present': True, 'separator': 'tab', 'comma_as_decimal': True, 'columns': ['time [s]', 'Fz'], 'output_path': 'D:\\modified_1-1B.txt'} Executing tool: save_parsed_data Arguments: {'file_path': 'D:\\1-1B.txt', 'skip_rows': 3, 'header_present': True, 'separator': 'tab', 'comma_as_decimal': True, 'columns': ['time [s]', 'Fz'], 'output_path': 'D:\\modified_1-1B.txt'}	

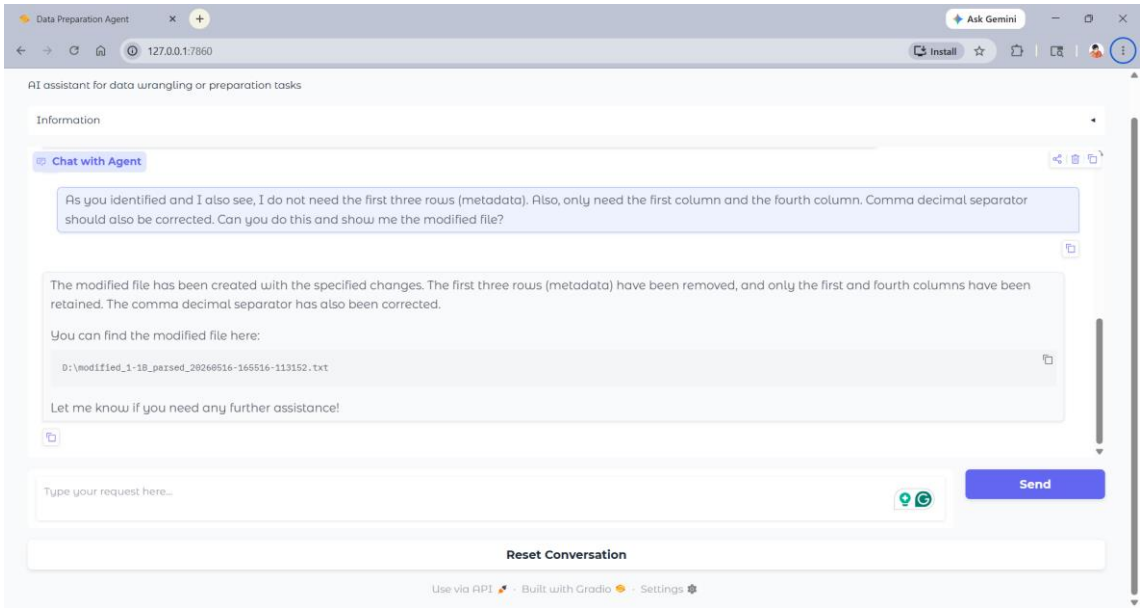


Figure S3-2. Screenprint related to Table S3-2.

Table S3-3. Conversational logs and tool invocation traces for batch-processing ingestion and parsing.

User	That's neat! Can you please extend this same thing to batch process other files? All the files are here: "D:\parse_in". The processed files I need to save in the location is here: "D:\parse_out". Also I want to rename the first column header as "time [s]" and the fourth column header as "Fz [N]" while doing this.
Agent	The batch processing is complete! All files from "D:\parse_in" have been processed, and the specified columns have been extracted with the headers renamed. The processed files are saved in "D:\parse_out". Here are some of the saved files: D:\parse_out\1-1B_parsed_20260516-170750-444019.txt D:\parse_out\10-1B_parsed_20260516-170833-738493.txt D:\parse_out\11-1B_parsed_20260516-170917-416154.txt D:\parse_out\12-1B_parsed_20260516-171000-958023.txt D:\parse_out\13-1B_parsed_20260516-171043-052970.txt A total of 33 files were successfully processed, and there were no failures. Let me know if there's anything else you need!
Backend (Tool Invocation Trace): [Agent] Calling tool: batch_parse_data [Agent] With arguments: {'input_folder': 'D:\\parse_in', 'output_folder': 'D:\\parse_out', 'skip_rows': 3, 'header_present': True, 'separator': 'tab', 'comma_as_decimal': True, 'columns': ['time [s]', 'Fz'], 'rename_headers': ['time [s]', 'Fz [N]']} Executing tool: batch_parse_data Arguments: {'input_folder': 'D:\\parse_in', 'output_folder': 'D:\\parse_out', 'skip_rows': 3, 'header_present': True, 'separator': 'tab', 'comma_as_decimal': True, 'columns': ['time [s]', 'Fz'], 'rename_headers': ['time [s]', 'Fz [N]']}	

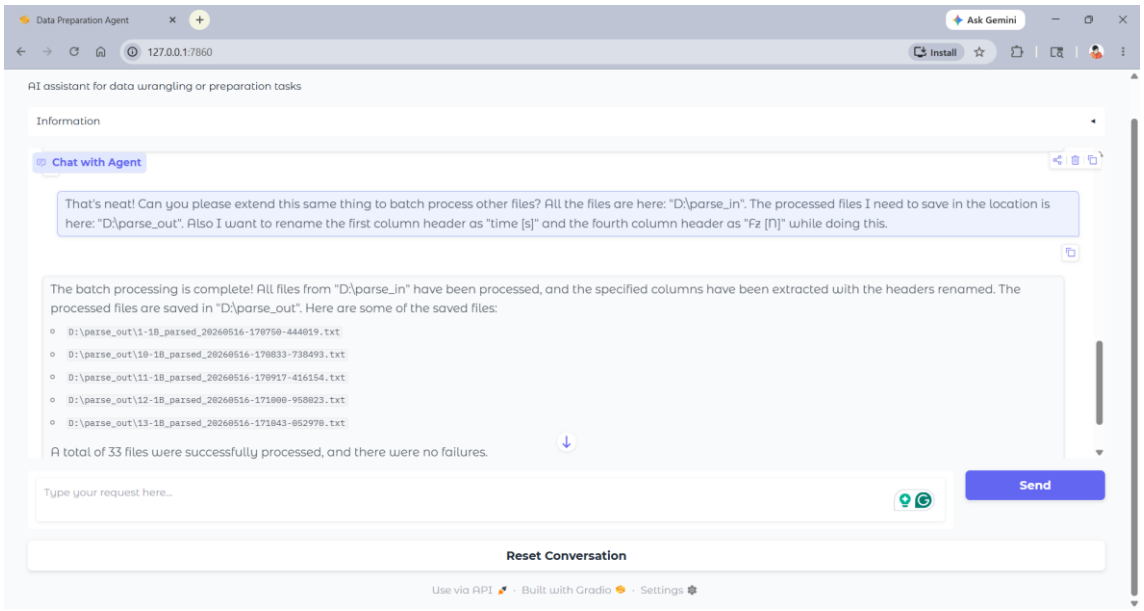
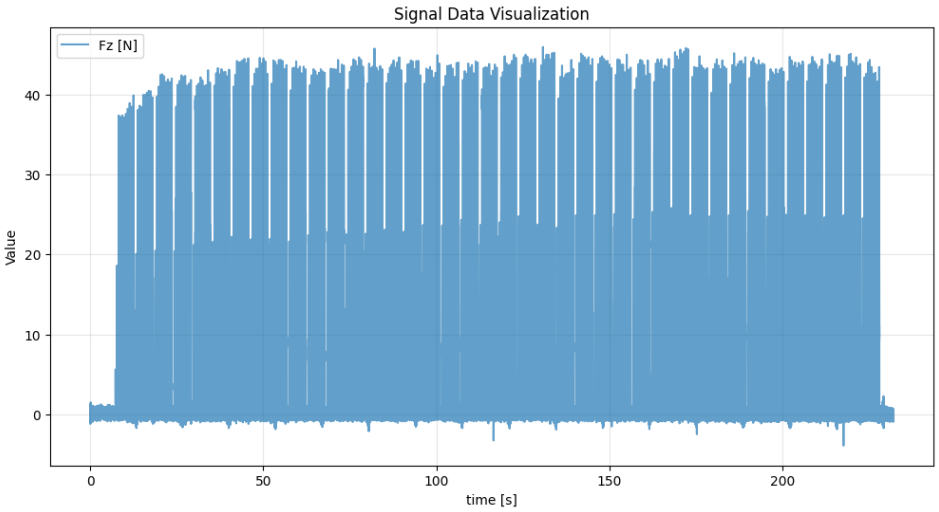


Figure S3-3. Screenprint related to Table S3-3.

Table S3-4. Conversational logs and tool invocation traces for data visualization.

User	Great! Please create a plot for this file: "D:\parse_out\1-1B_parsed_20260516-170750-444019.txt".
Agent	The plot for the file "D:\parse_out\1-1B_parsed_20260516-170750-444019.txt" has been created. You can view the visualization here:  Let me know if there's anything else you'd like to do! Plot saved at: D:\Codes\Data Processing Systems\Data-Preparation-Toolkit-Agentic AI\temp_processing\1-1B_parsed_20260516-170750-444019_visualization_20260516-174326-007305.png
Backend (Tool Invocation Trace): [Agent] Calling tool: visualize_data [Agent] With arguments: {'file_path': 'D:\\parse_out\\1-1B_parsed_20260516-170750-444019.txt', 'x_column': 'time [s]', 'y_columns': ['Fz [N]']} Executing tool: visualize_data Arguments: {'file_path': 'D:\\parse_out\\1-1B_parsed_20260516-170750-444019.txt', 'x_column': 'time [s]', 'y_columns': ['Fz [N]']}	

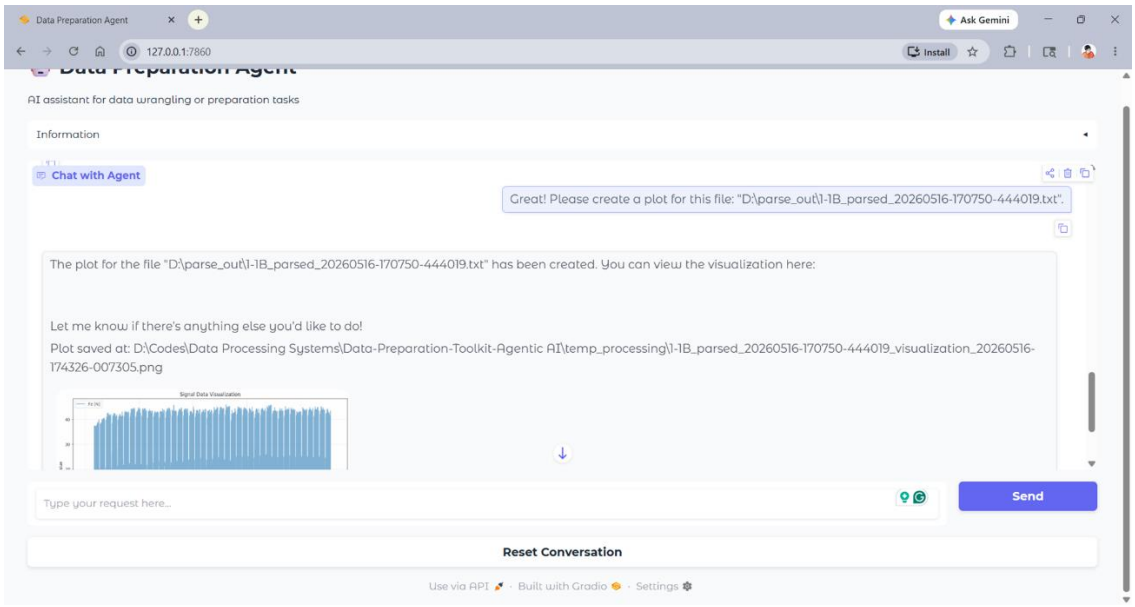


Figure S3-4. Screenprint related to Table S3-4.

Table S3-5. Conversational logs and tool invocation traces for exploring and initiating segmentation.

User	Thanks ... I see ... I need to segment this file into multiple parts. What are the ways?
Agent	There are two primary ways to segment a data file into multiple parts: 1. Segment by X-Axis Range: This method involves specifying a range of values on the x-axis (e.g., time) to extract a specific portion of the data. You define the minimum and maximum x-values, and the data within this range is extracted. 2. Segment by Threshold Rule: This method involves identifying regions in the data where a specific column exceeds a certain threshold value. You can specify additional parameters like a search window to determine how far to look for the continuation of an event, and padding to include data before and after the detected event. Which Method to Use? Segment by X-Axis Range: Use this if you know the exact time or index range you want to extract. Segment by Threshold Rule: Use this if you want to extract events based on specific conditions in your data, such as peaks or spikes. If you have a specific method in mind or need further guidance on setting parameters, let me know!
Backend (Tool Invocation Trace): [No tool invoked; Conversational mode for guidance and explanation]	

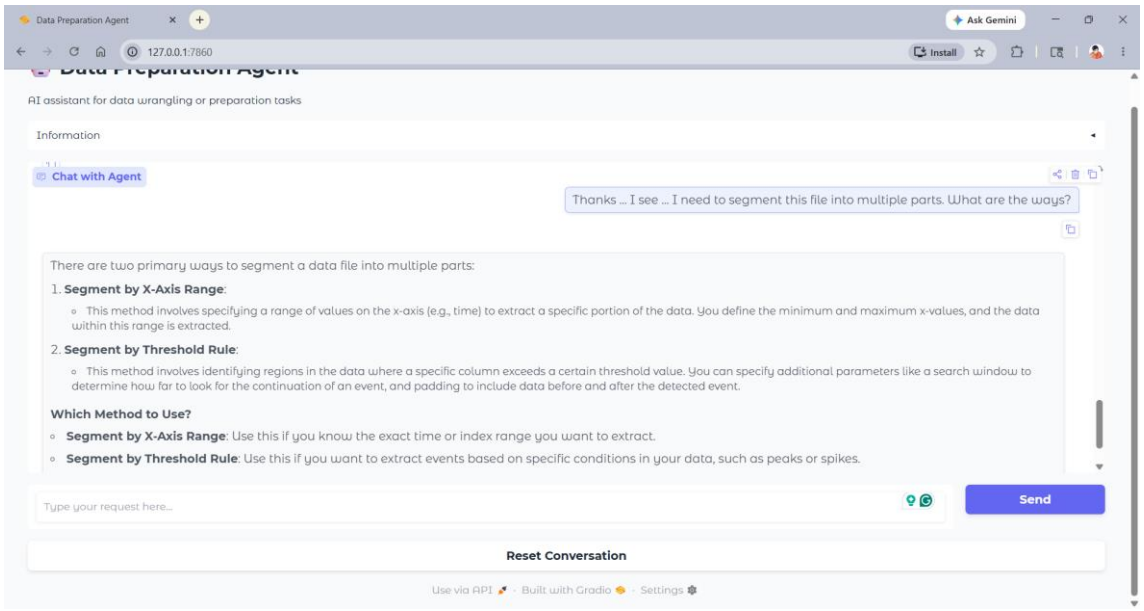


Figure S3-5. Screenprint related to Table S3-5.

Table S3-6. Conversational logs and tool invocation traces for exploring and initiating segmentation.

User	I think “by threshold” will help. What information do I need to provide?
Agent	To segment the file by threshold, you’ll need to provide the following information: 1. X-Axis Column Name (required): The column that represents the x-axis (e.g., “time [s]”). 2. Data Column Name (required): The column on which you want to apply the threshold (e.g., “Fz [N]”). 3. Threshold Value (required): The value above which the data is considered part of an event. 4. Search Window (optional, default: 6000 samples): The number of samples to look ahead to determine if an event continues. 5. Start Padding (optional, default: 5000 samples): The number of samples to include before the start of an event. 6. End Padding (optional, default: 6000 samples): The number of samples to include after the end of an event. 7. Output Folder (optional): The folder where the segmented files will be saved. If not specified, a timestamped temporary folder will be used. Example: X-Axis Column: “time [s]” Data Column: “Fz [N]” Threshold: 0.5 Search Window: 6000 Start Padding: 5000 End Padding: 6000 Please provide the necessary details, and I’ll help you set up the segmentation!
Backend (Tool Invocation Trace): [No tool invoked; Conversational mode for guidance and explanation]	

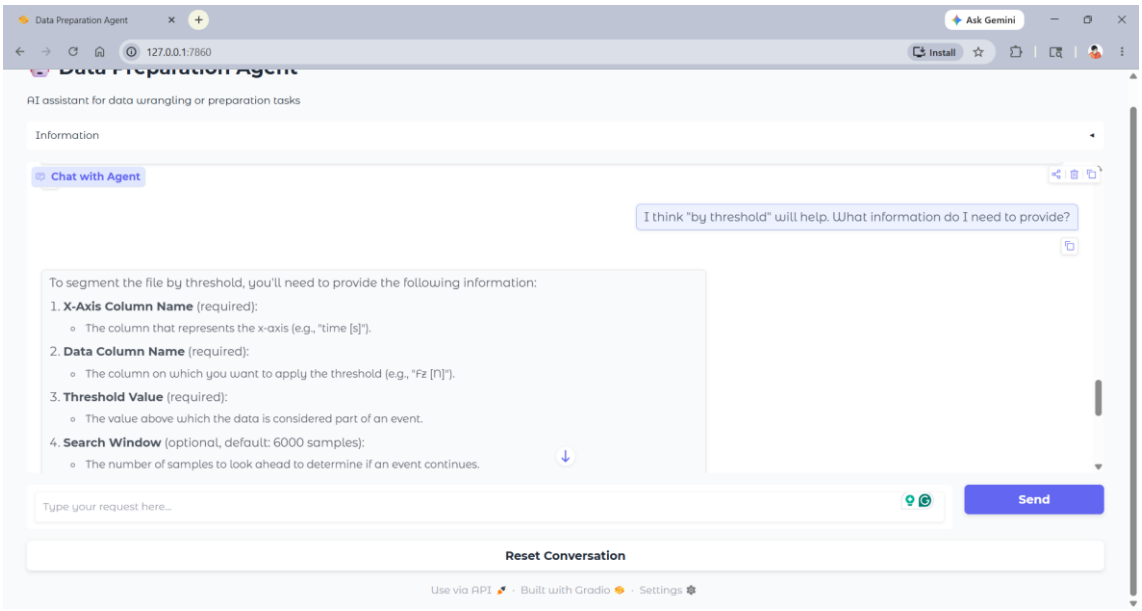


Figure S3-6. Screenprint related to Table S3-6.

Table S3-7. Conversational logs and tool invocation traces for segmenting a single file.

User	Threshold = 10, Search Window = 6000, Start and End Padding both are zero (0). X-axis column is the first one (time [s]). Data column is the second one (Fz [N]). And the segmented files should be saved in this folder: "D:\segment".
Agent	The file has been successfully segmented based on the threshold value of 10. A total of 40 segments were identified and saved. The segmented files are located in the following folder: D:\segment\1-1B_parsed_20260516-170750-444019_segments Each segment is saved as a separate file. Let me know if there's anything else you need!
Backend (Tool Invocation Trace): [Agent] Calling tool: segment_by_threshold [Agent] With arguments: {'file_path': 'D:\\parse_out\\1-1B_parsed_20260516-170750-444019.txt', 'x_column': 'time [s]', 'data_column': 'Fz [N]', 'threshold': 10, 'search_window': 6000, 'start_padding': 0, 'end_padding': 0, 'output_folder': 'D:\\segment'} Executing tool: segment_by_threshold Arguments: {'file_path': 'D:\\parse_out\\1-1B_parsed_20260516-170750-444019.txt', 'x_column': 'time [s]', 'data_column': 'Fz [N]', 'threshold': 10, 'search_window': 6000, 'start_padding': 0, 'end_padding': 0, 'output_folder': 'D:\\segment'}	

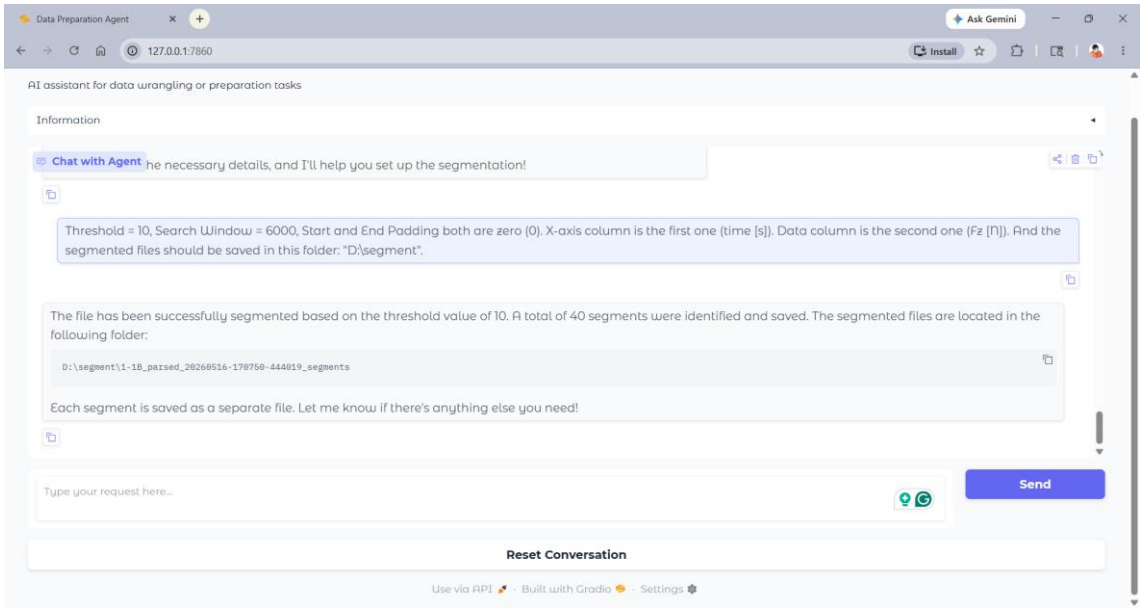


Figure S3-7. Screenprint related to Table S3-7.

Table S3-8. Conversational logs and tool invocation traces for batch-processing segmentation.

User	Great! Can we extend this segmentation process for batch-processing the other files present in the folder: “D:\segment_in”? The output location would be: “D:\segment_out”.
Agent	The batch segmentation process is complete! Here’s a summary of the results: Total Files Processed: 33 Total Segments Found: 1317 Failed Files: 0 Each file from “D:\segment_in” has been segmented based on the threshold value of 10, and the segments have been saved in “D:\segment_out”. If you need further assistance or have any other requests, feel free to ask!
Backend (Tool Invocation Trace): [Agent] Calling tool: batch_segment_threshold [Agent] With arguments: {‘input_folder’: ‘D:\\segment_in’, ‘output_folder’: ‘D:\\segment_out’, ‘x_column’: ‘time [s]’, ‘data_column’: ‘Fz [N]’, ‘threshold’: 10, ‘search_window’: 6000, ‘start_padding’: 0, ‘end_padding’: 0} Executing tool: batch_segment_threshold Arguments: {‘input_folder’: ‘D:\\segment_in’, ‘output_folder’: ‘D:\\segment_out’, ‘x_column’: ‘time [s]’, ‘data_column’: ‘Fz [N]’, ‘threshold’: 10, ‘search_window’: 6000, ‘start_padding’: 0, ‘end_padding’: 0}	

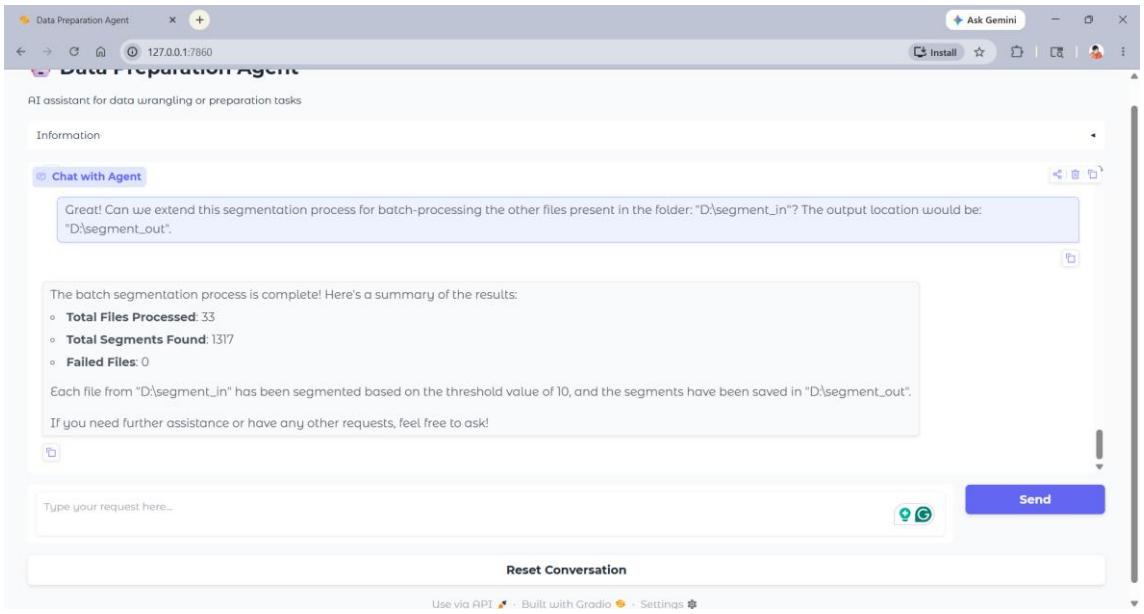


Figure S3-8. Screenprint related to Table S3-8.

Table S3-9. Conversational logs and tool invocation traces for data ingestion and exploration.

User	Show me the contents of these two files: "D:\1-1B_parsed_20260516-170750-444019_segment_1.txt" and "D:\1-1B_parsed_20260516-170750-444019_segment_40.txt".
Agent	Here are the first 15 lines of each file: File: D:\1-1B_parsed_20260516-170750-444019_segment_1.txt time [s],Fz [N] 7.758,10.560797 7.7581,10.882932 7.7582,11.076212 7.7583,10.882932 7.7584,10.947358 7.7585,10.754078 7.7586,10.882932 7.7587,11.720481 7.7588,12.364749 7.7589,12.55803 7.759,12.686883 7.7591,13.137871 7.7592,13.202298 7.7593,13.331152 File: D:\1-1B_parsed_20260516-170750-444019_segment_40.txt time [s],Fz [N] 223.1851,10.238663 223.1852,10.625224 223.1853,11.269493 223.1854,11.205066 223.1855,11.269493 223.1856,11.784908 223.1857,11.913761 223.1858,12.235895 223.1859,12.944591 223.186,13.653286 223.1861,13.588859

	223.1862,13.653286 223.1863,13.524433 223.1864,13.653286 If you need further analysis or operations on these files, let me know!
Backend (Tool Invocation Trace): [Agent] Calling tool: show_raw_content [Agent] With arguments: {'file_path': 'D:\\1-1B_parsed_20260516-170750-444019_segment_1.txt', 'num_lines': 15} Executing tool: show_raw_content Arguments: {'file_path': 'D:\\1-1B_parsed_20260516-170750-444019_segment_1.txt', 'num_lines': 15} [Agent] Calling tool: show_raw_content [Agent] With arguments: {'file_path': 'D:\\1-1B_parsed_20260516-170750-444019_segment_40.txt', 'num_lines': 15} Executing tool: show_raw_content Arguments: {'file_path': 'D:\\1-1B_parsed_20260516-170750-444019_segment_40.txt', 'num_lines': 15}	

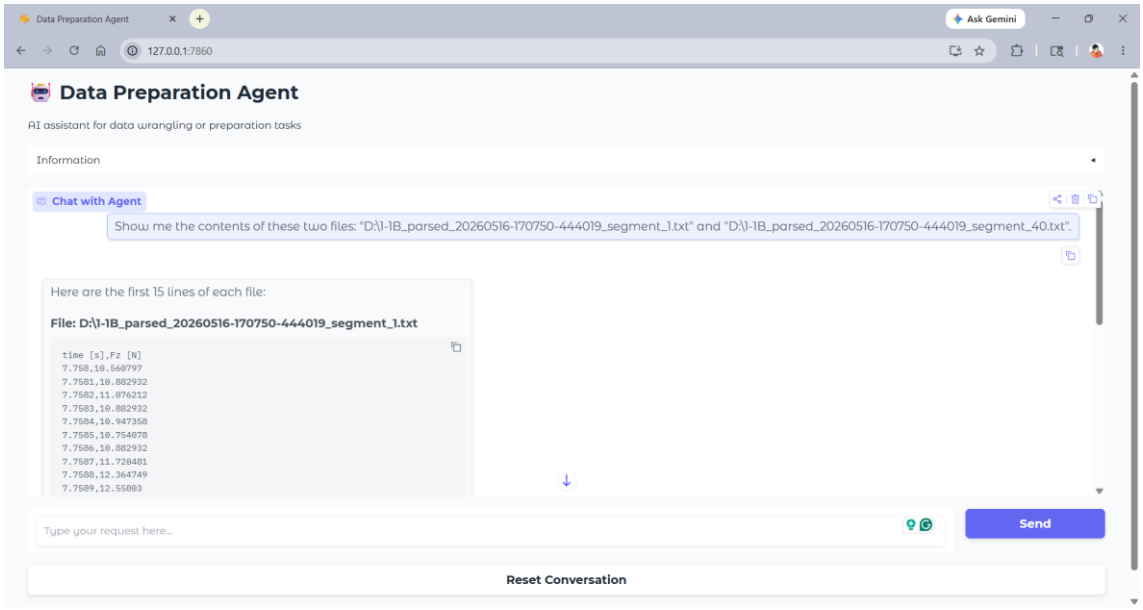
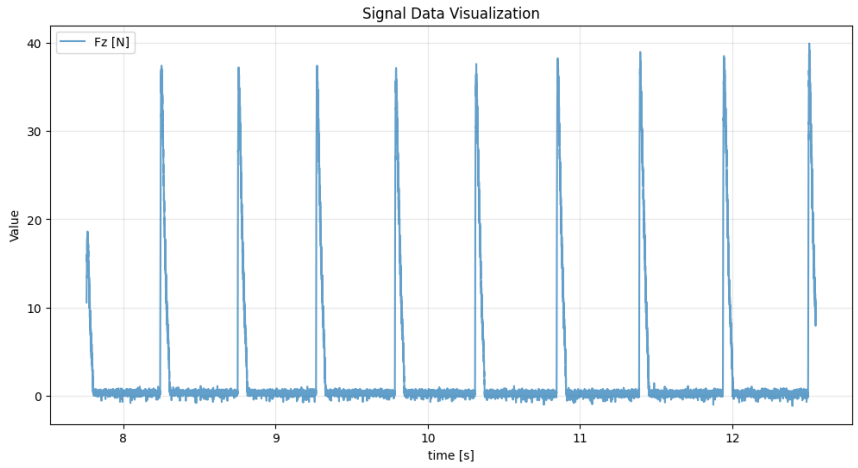
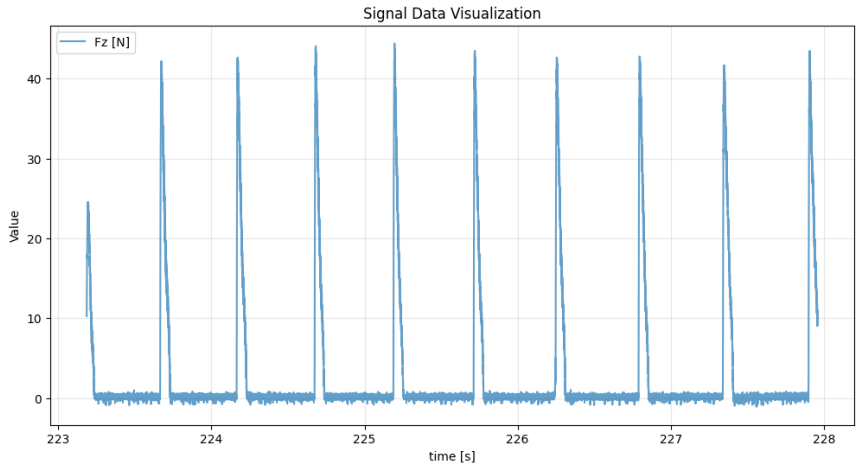


Figure S3-9. Screenprint related to Table S3-9.

Table S3-10. Conversational logs and tool invocation traces for data visualization.

User	I need to see the plots of these files as well. The first column (time [S]) as X-axis and the second one (Fz [N]) as Y-axis.
Agent	Here are the plots for the two files: Plot for D:\1-1B_parsed_20260516-170750-444019_segment_1.txt  Plot for D:\1-1B_parsed_20260516-170750-444019_segment_40.txt  If you need further analysis or any other operations, feel free to ask! Plot saved at: D:\Codes\Data Processing Systems\Data-Preparation-Toolkit-Agentic AI\temp_processing\1-1B_parsed_20260516-170750-444019_segment_1_visualization_20260516-190409-680040.png Plot saved at: D:\Codes\Data Processing Systems\Data-Preparation-Toolkit-Agentic AI\temp_processing\1-1B_parsed_20260516-170750-444019_segment_40_visualization_20260516-190416-008751.png

Backend (Tool Invocation Trace):

[Agent] Calling tool: visualize_data

[Agent] With arguments: {‘file_path’: ‘D:\\1-1B_parsed_20260516-170750-444019_segment_1.txt’, ‘x_column’: ‘time [s]’, ‘y_columns’: [‘Fz [N]’]}

Executing tool: visualize_data

Arguments: {‘file_path’: ‘D:\\1-1B_parsed_20260516-170750-444019_segment_1.txt’, ‘x_column’: ‘time [s]’, ‘y_columns’: [‘Fz [N]’]}

[Agent] Calling tool: visualize_data

[Agent] With arguments: {‘file_path’: ‘D:\\1-1B_parsed_20260516-170750-444019_segment_40.txt’, ‘x_column’: ‘time [s]’, ‘y_columns’: [‘Fz [N]’]}

Executing tool: visualize_data

Arguments: {‘file_path’: ‘D:\\1-1B_parsed_20260516-170750-444019_segment_40.txt’, ‘x_column’: ‘time [s]’, ‘y_columns’: [‘Fz [N]’]}

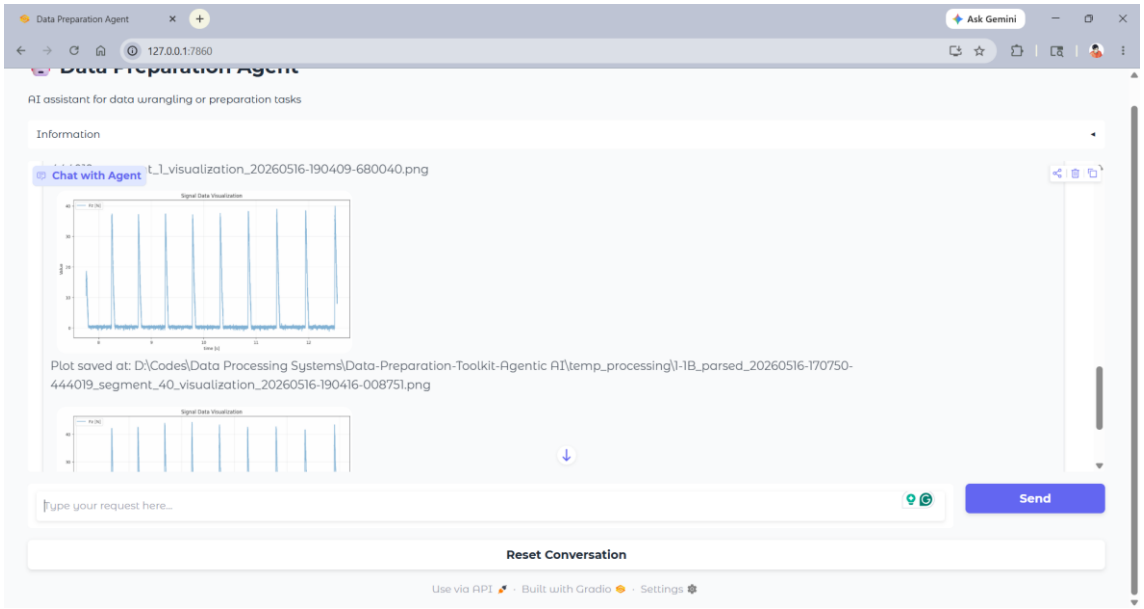


Figure S3-10. Screenprint related to Table S3-10.

Table S3-11. Conversational logs and tool invocation traces for exploring and initiating feature-based sampling (peak detection and extraction).

User	From the plots, I see there are certain peaks in each. I need to perform such peak extraction. Any way?
Agent	To perform peak extraction from the plots, we can use the peak detection tool. Here's what we need to proceed: Required Parameters: 1. File Path: The path to the file containing the data. 2. X Column: The column to use for the x-axis (in this case, "time [s]"). 3. Y Column: The column to analyze for peaks (in this case, "Fz [N]"). Optional Parameters: Peak Type: Choose between "maxima", "minima", or "both". Default is "maxima". Prominence: Minimum prominence of peaks to detect. Height: Minimum height of peaks. Distance: Minimum distance between peaks. Width: Minimum width of peaks. Execution Plan: 1. Step 1: Detect peaks in "D:\1-1B_parsed_20260516-170750-444019_segment_1.txt" X Column: "time [s]" Y Column: "Fz [N]" Peak Type: (default or specify) Prominence: (default or specify) Height: (default or specify) Distance: (default or specify) Width: (default or specify) 2. Step 2: Detect peaks in "D:\1-1B_parsed_20260516-170750-444019_segment_40.txt" Same parameters as Step 1 Please specify any optional parameters you would like to set, or confirm if the default settings are acceptable.
Backend (Tool Invocation Trace): [No tool invoked; Conversational mode for guidance, explanation, and planning]	

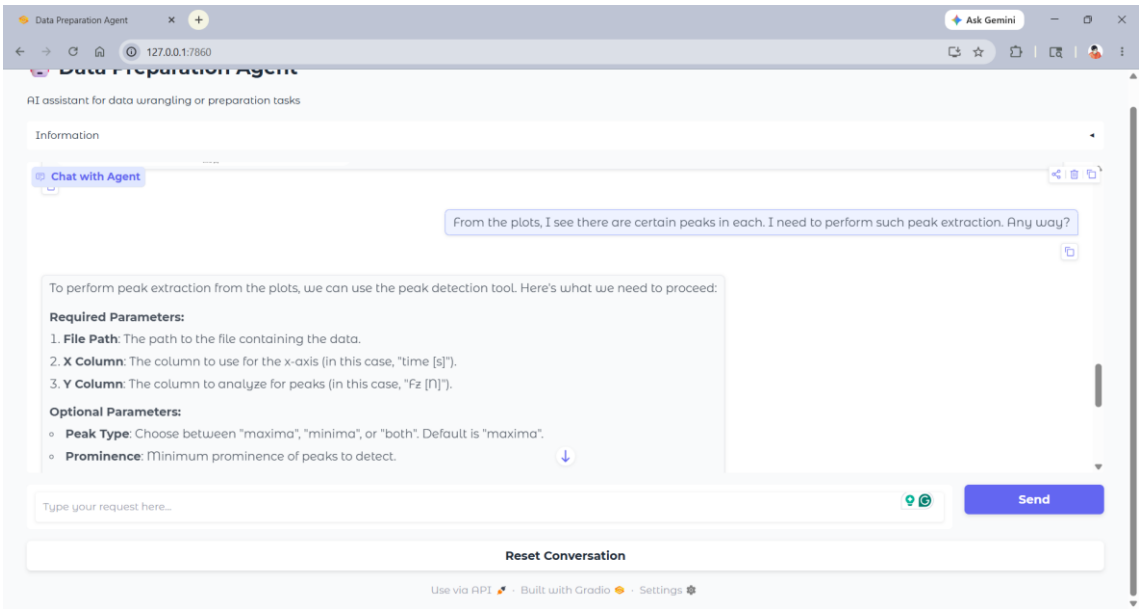
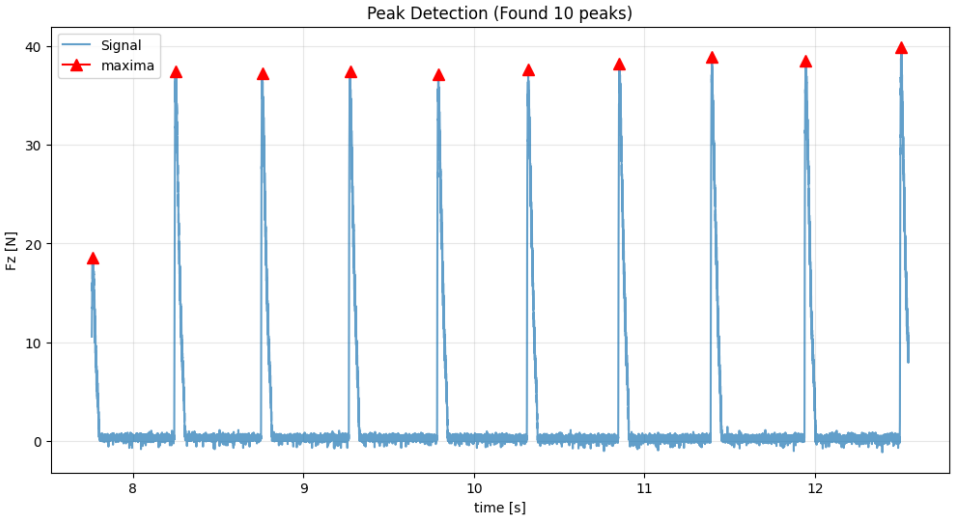


Figure S3-11. Screenprint related to Table S3-11.

Table S3-12. Conversational logs and tool invocation traces for extracting peaks from a single file.

User	Thanks! For trial, let's perform on this file: "D:\1-1B_parsed_20260516-170750-444019_segment_1.txt". Settings are: maxima, X Column: "time [s]", Y Column: "Fz [N]", prominence: 5, height: 10, and distance: 50. And also show me the plot.
Agent	The peak detection process is complete. Here are the results: Number of Peaks Found: 10 Peak Type: Maxima Peaks Plot:  Would you like to perform any further analysis or operations? Plot saved at: D:\Codes\Data Processing Systems\Data-Preparation-Toolkit-Agentic AI\temp_processing\1-1B_parsed_20260516-170750-444019_segment_1_peaks_plot_20260516-192130-366096.png
Backend (Tool Invocation Trace): [Agent] Calling tool: find_peaks [Agent] With arguments: {'file_path': 'D:\\1-1B_parsed_20260516-170750-444019_segment_1.txt', 'x_column': 'time [s]', 'y_column': 'Fz [N]', 'peak_type': 'maxima', 'prominence': 5, 'height': 10, 'distance': 50} Executing tool: find_peaks Arguments: {'file_path': 'D:\\1-1B_parsed_20260516-170750-444019_segment_1.txt', 'x_column': 'time [s]', 'y_column': 'Fz [N]', 'peak_type': 'maxima', 'prominence': 5, 'height': 10, 'distance': 50}	

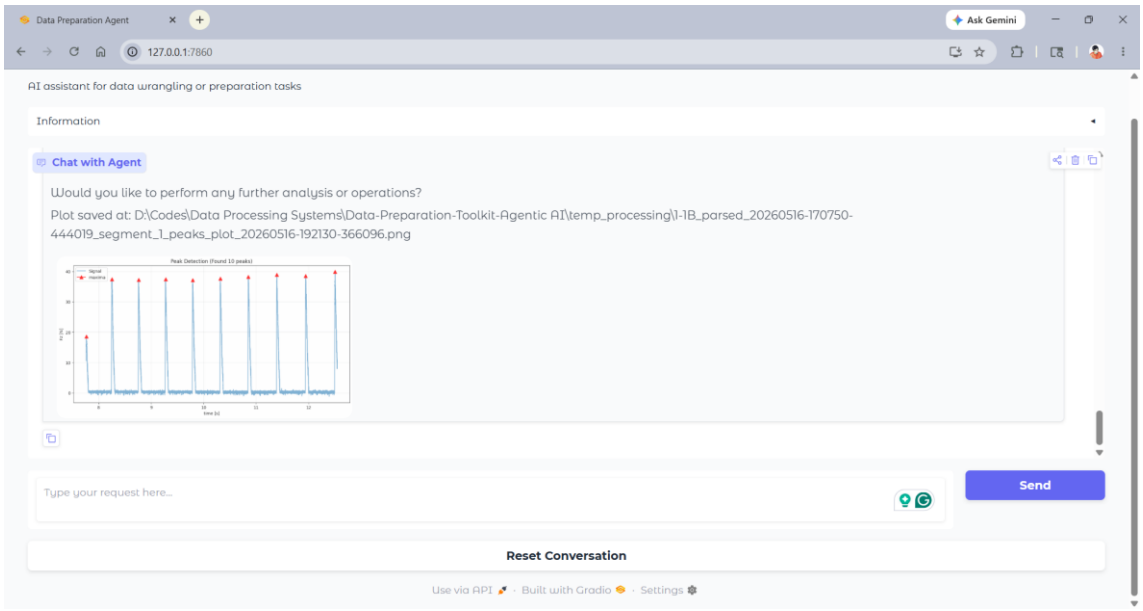


Figure S3-12. Screenprint related to Table S3-12.

Table S3-13. Conversational logs and tool invocation traces for initiating and planning batch-processing peak extraction.

User	Great! Can we extend this peak extraction for batch-processing the files I have here: "D:\peak_in"? I want the output files saved in this folder: "D:\peak_out".
Agent	To perform batch peak extraction on all files in the folder "D:\peak_in" and save the results in "D:\peak_out", we'll use the batch peak detection tool. Here's the plan: Required Parameters: 1. Input Folder: "D:\peak_in" 2. Output Folder: "D:\peak_out" Optional Parameters (same for all files): X Column: "time [s]" Y Column: "Fz [N]" Peak Type: "maxima" Prominence: 5 Height: 10 Distance: 50 Execution Plan: Step: Batch process all files in "D:\peak_in" with the specified parameters and save the results in "D:\peak_out". Does this plan look good to you? If so, I'll proceed with the batch processing.
Backend (Tool Invocation Trace): [No tool invoked; Conversational mode for guidance, explanation, and planning]	

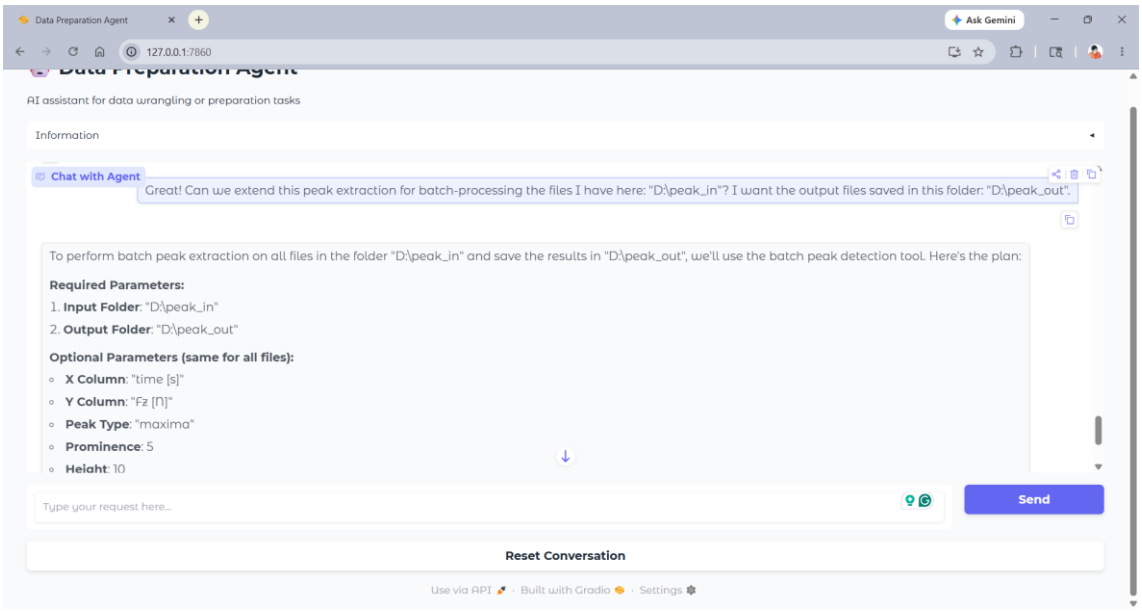


Figure S3-13. Screenprint related to Table S3-13.

Table S3-14. Conversational logs and tool invocation traces for batch-processing peak extraction.

User	Yes. Please proceed.
Agent	The batch peak extraction process is complete. Here are the results: Total Files Processed: 40 Files Successfully Saved: 40 Files Failed: 0 Saved Files: The peak data for each file has been saved in the "D:\peak_out" directory. Here are a few examples of the saved files: D:\peak_out\1-1B_parsed_20260516-170750-444019_segment_1_peaks_20260516-193809-593123.txt D:\peak_out\1-1B_parsed_20260516-170750-444019_segment_10_peaks_20260516-193810-060482.txt D:\peak_out\1-1B_parsed_20260516-170750-444019_segment_20_peaks_20260516-193815-273211.txt D:\peak_out\1-1B_parsed_20260516-170750-444019_segment_30_peaks_20260516-193820-255255.txt D:\peak_out\1-1B_parsed_20260516-170750-444019_segment_40_peaks_20260516-193824-717765.txt You can find all the peak data files in the specified output directory. If you need further assistance or analysis, feel free to ask!
Backend (Tool Invocation Trace): [Agent] Calling tool: batch_peaks [Agent] With arguments: {'input_folder': 'D:\\peak_in', 'output_folder': 'D:\\peak_out', 'x_column': 'time [s]', 'y_column': 'Fz [N]', 'peak_type': 'maxima', 'prominence': 5, 'height': 10, 'distance': 50} Executing tool: batch_peaks Arguments: {'input_folder': 'D:\\peak_in', 'output_folder': 'D:\\peak_out', 'x_column': 'time [s]', 'y_column': 'Fz [N]', 'peak_type': 'maxima', 'prominence': 5, 'height': 10, 'distance': 50}	

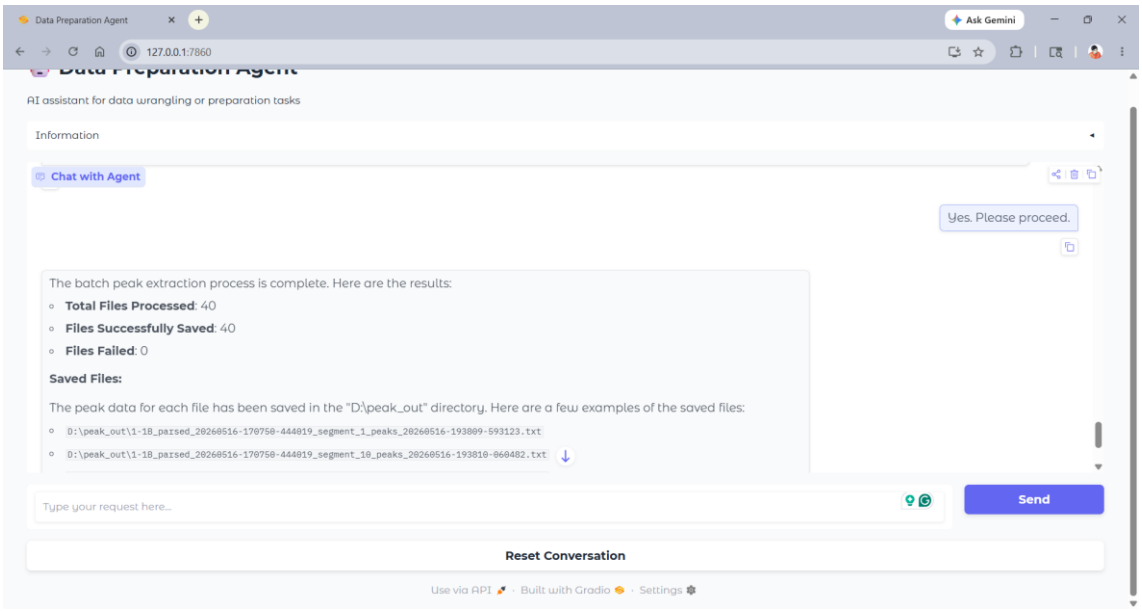


Figure S3-14. Screenprint related to Table S3-14.

Table S3-15. Conversational logs and tool invocation traces for data ingestion and exploration.

User	Show me the contents of this file: "D:\1-1B_parsed_20260516-170750-444019_segment_1_peaks_20260516-193809-593123.txt".
Agent	Here are the first 11 lines of the file "D:\1-1B_parsed_20260516-170750-444019_segment_1_peaks_20260516-193809-593123.txt": original_index,time [s],Fz [N],peak_type 69,7.7649,18.614154,maxima 4926,8.2506,37.426795,maxima 9994,8.7574,37.233515,maxima 15150,9.273,37.426795,maxima 20339,9.7919,37.169088,maxima 25604,10.3184,37.620076,maxima 30956,10.8536,38.264344,maxima 36383,11.3963,38.97304,maxima 41871,11.9451,38.522052,maxima 47484,12.5064,39.939443,maxima If you need further analysis or operations on this data, let me know!
Backend (Tool Invocation Trace): [Agent] Calling tool: show_raw_content [Agent] With arguments: {'file_path': 'D:\\1-1B_parsed_20260516-170750-444019_segment_1_peaks_20260516-193809-593123.txt', 'num_lines': 15} Executing tool: show_raw_content Arguments: {'file_path': 'D:\\1-1B_parsed_20260516-170750-444019_segment_1_peaks_20260516-193809-593123.txt', 'num_lines': 15}	

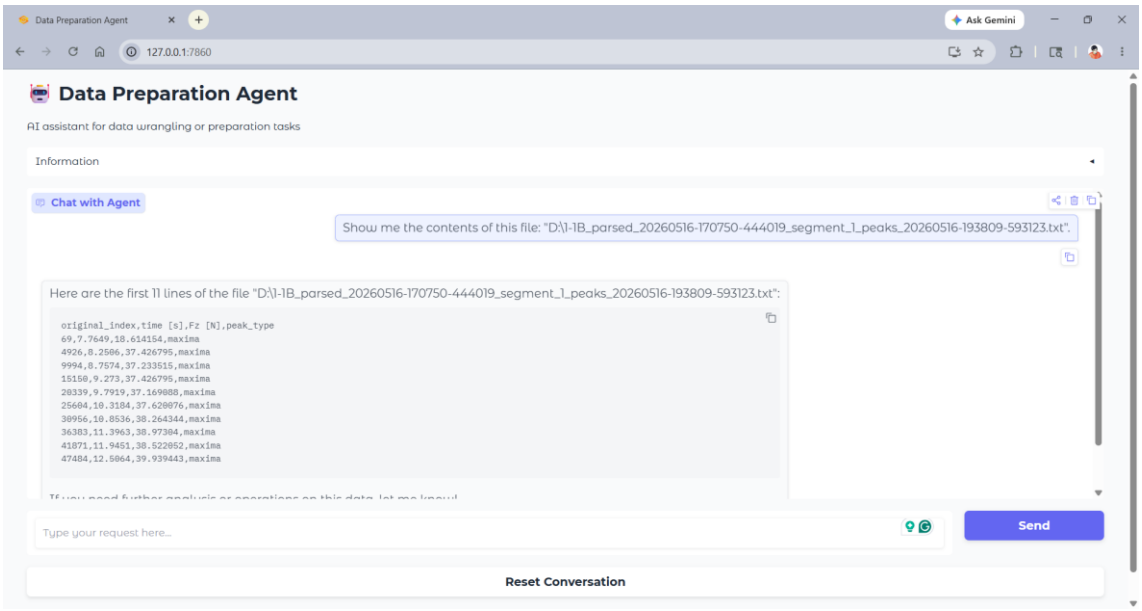


Figure S3-15. Screenprint related to Table S3-15.

Table S3-16. Conversational logs and tool invocation traces for exploring and initiating augmentation.

User	I see I need to delete some columns, add a new column, and add some metadata on top. Can we do that?
Agent	Certainly! To modify the file by deleting columns, adding a new column, and inserting metadata, we'll follow these steps: PHASE 1: PLANNING & VALIDATION 1. Delete Columns: Required: List of column names to delete. Please specify which columns you want to remove. 2. Add New Column: Required: Name for the new column. Required: Starting value for the sequence. Required: Increment (step) between values. Optional: Position to insert the new column (start or end). Optional: Insert after a specific column (overrides position). 3. Add Metadata: Required: Metadata text to prepend before data. Execution Plan 1. Step 1: Delete specified columns. 2. Step 2: Insert a new column with a sequence. 3. Step 3: Add metadata at the top of the file. Please provide the following details: - Which columns should be deleted? - What should be the name, starting value, and step for the new column? - Where should the new column be inserted (start, end, or after a specific column)? - What metadata text should be added at the top? Once you provide these details, I'll present the complete execution plan for your confirmation.
Backend (Tool Invocation Trace): [No tool invoked; Conversational mode for guidance, explanation, and planning]	

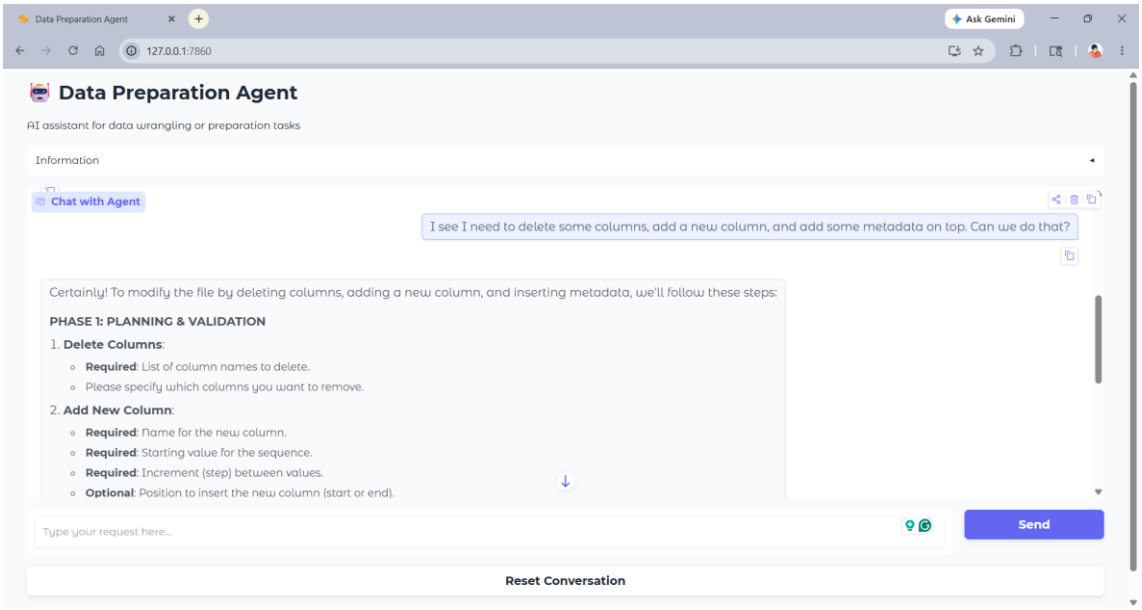


Figure S3-16. Screenprint related to Table S3-16.

Table S3-17. Conversational logs and tool invocation traces for initiating and planning augmentation.

User	Columns to delete: 1st, 2nd, and 4th. Add a new column with header ‘i’. Start with 1, with Step increase ‘1’. This ‘i’ column should be the first column in the modified file. The metadata on top is: (1st line) Process: Micro Drilling (2nd line) Tool: Solid Carbide micro-INOX 8×D drill (3rd line) Workpiece: Titanium Block (Ti6Al4V) (4th line) Cycle: 1 .... Finally, the modified file should be saved in the same location as this file.
Agent	Great! Let’s outline the complete execution plan based on your requirements: EXECUTION PLAN 1. Step 1: Delete Columns File: “D:\1-1B_parsed_20260516-170750-444019_segment_1_peaks_20260516-193809-593123.txt” Columns to Delete: 1st (original_index), 2nd (time [s]), and 4th (peak_type) 2. Step 2: Insert New Column File: Output from Step 1 New Column Header: ‘i’ Starting Value: 1 Step: 1 Position: Start (first column) 3. Step 3: Add Metadata File: Output from Step 2 Metadata: Process: Micro Drilling Tool: Solid Carbide micro-INOX 8×D drill Workpiece: Titanium Block (Ti6Al4V) Cycle: 1 4. Output File: Save the modified file in the same location as the original file. Confirmation Does this plan look correct to you? If so, I’ll proceed with the execution. If there are any changes needed, please let me know!
Backend (Tool Invocation Trace): [No tool invoked; Conversational mode for guidance, explanation, and planning]	

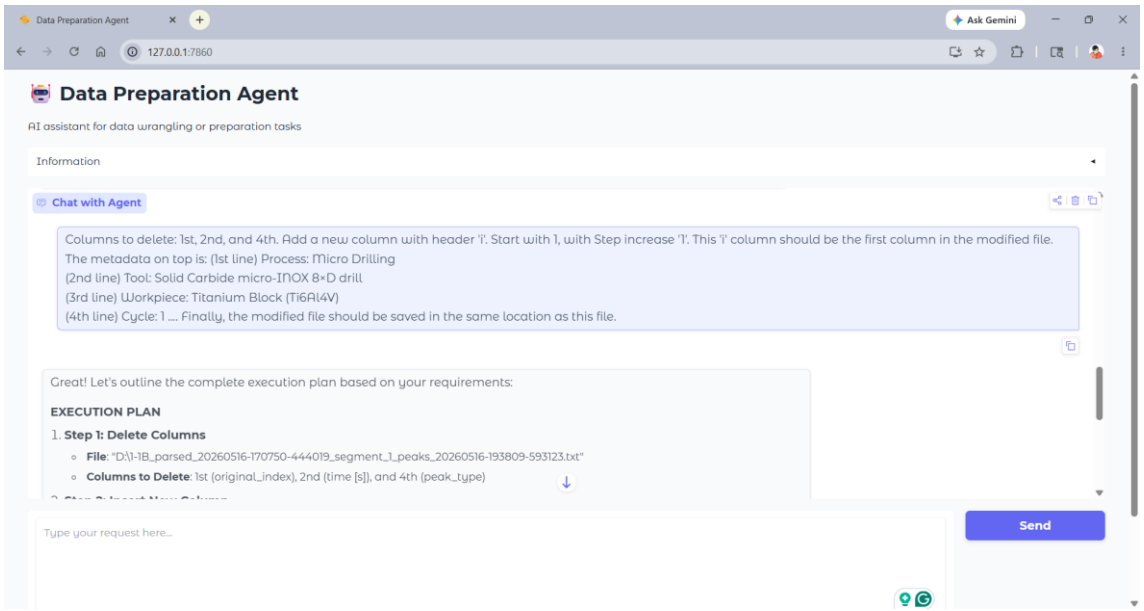


Figure S3-17. Screenprint related to Table S3-17.

Table S3-18. Conversational logs and tool invocation traces for augmenting a single file.

User	Plan looks good. Please go ahead.
Agent	Step 1 Completed: The specified columns have been deleted. The remaining column is “Fz [N]”. The intermediate file is saved at: temp_processing\1-1B_parsed_20260516-170750-444019_segment_1_peaks_20260516-193809-593123_modified_20260516-211623-654515.txt Proceeding to Step 2: Insert the new column ‘i’.
Backend (Tool Invocation Trace): [Agent] Calling tool: delete_data [Agent] With arguments: {‘file_path’: ‘D:\\1-1B_parsed_20260516-170750-444019_segment_1_peaks_20260516-193809-593123.txt’, ‘column_names’: [‘original_index’, ‘time [s]’, ‘peak_type’]} Executing tool: delete_data Arguments: {‘file_path’: ‘D:\\1-1B_parsed_20260516-170750-444019_segment_1_peaks_20260516-193809-593123.txt’, ‘column_names’: [‘original_index’, ‘time [s]’, ‘peak_type’]}	
User	Okay ...
Agent	Step 2 Completed: The new column ‘i’ has been inserted as the first column with values starting from 1 and incrementing by 1. The updated file is saved at: temp_processing\1-1B_parsed_20260516-170750-444019_segment_1_peaks_20260516-193809-593123_modified_20260516-211623-654515_modified_20260516-211648-027110.txt Proceeding to Step 3: Add the specified metadata at the top of the file.
Backend (Tool Invocation Trace): [Agent] Calling tool: insert_data [Agent] With arguments: {‘file_path’: ‘temp_processing\\1-1B_parsed_20260516-170750-444019_segment_1_peaks_20260516-193809-593123_modified_20260516-211623-654515.txt’, ‘column_header’: ‘i’, ‘start’: 1, ‘step’: 1, ‘position’: ‘start’} Executing tool: insert_data Arguments: {‘file_path’: ‘temp_processing\\1-1B_parsed_20260516-170750-444019_segment_1_peaks_20260516-193809-593123_modified_20260516-211623-654515.txt’, ‘column_header’: ‘i’, ‘start’: 1, ‘step’: 1, ‘position’: ‘start’}	
User	Okay ...
Agent	Step 3 Completed: The specified metadata has been added to the top of the file. The final modified file is saved at: D:\1-1B_parsed_20260516-170750-444019_segment_1_peaks_modified_modified_20260516-211707-831374.txt

	The file now includes the new column ‘i’, the remaining column ‘Fz [N]’, and the metadata at the top. If you need any further modifications or have other tasks, feel free to let me know!
Backend (Tool Invocation Trace): [Agent] Calling tool: insert_data [Agent] With arguments: {‘file_path’: ‘temp_processing\\1-1B_parsed_20260516-170750-444019_segment_1_peaks_20260516-193809-593123_modified_20260516-211623-654515_modified_20260516-211648-027110.txt’, ‘metadata’: ‘Process: Micro Drilling\nTool: Solid Carbide micro-INOX 8×D drill\nWorkpiece: Titanium Block (Ti6Al4V)\nCycle: 1’, ‘output_path’: ‘D:\\1-1B_parsed_20260516-170750-444019_segment_1_peaks_modified.txt’} Executing tool: insert_data Arguments: {‘file_path’: ‘temp_processing\\1-1B_parsed_20260516-170750-444019_segment_1_peaks_20260516-193809-593123_modified_20260516-211623-654515_modified_20260516-211648-027110.txt’, ‘metadata’: ‘Process: Micro Drilling\nTool: Solid Carbide micro-INOX 8×D drill\nWorkpiece: Titanium Block (Ti6Al4V)\nCycle: 1’, ‘output_path’: ‘D:\\1-1B_parsed_20260516-170750-444019_segment_1_peaks_modified.txt’}	

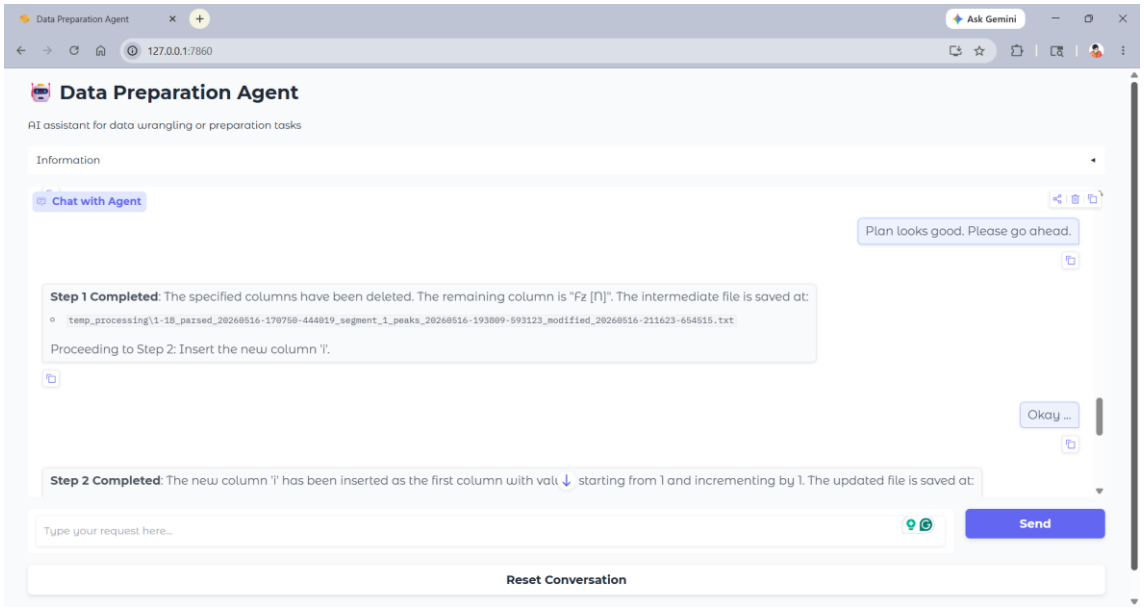


Figure S3-18. Screenprint related to Table S3-18.

Table S3-19. Conversational logs and tool invocation traces for initiating and planning batch-processing augmentation.

User	Great! Can we extend this to batch-process all the files in this folder: “D:\augment_in”? The final modified files should be saved in this folder: “D:\augment_out”.
Agent	To batch-process all files in the specified folder, we’ll follow a similar sequence of operations for each file: Batch Processing Plan 1. Delete Columns: Columns to Delete: 1st (original_index), 2nd (time [s]), and 4th (peak_type) 2. Insert New Column: New Column Header: ‘i’ Starting Value: 1 Step: 1 Position: Start (first column) 3. Add Metadata: Metadata: Process: Micro Drilling Tool: Solid Carbide micro-INOX 8×D drill Workpiece: Titanium Block (Ti6Al4V) Cycle: 1 4. Output Folder: Save the modified files in “D:\augment_out”. Execution Plan We’ll use the batch processing tool to apply these operations to all files in “D:\augment_in” and save the results in “D:\augment_out”. Does this plan look correct to you? If so, I’ll proceed with the execution. If there are any changes needed, please let me know!
Backend (Tool Invocation Trace): [No tool invoked; Conversational mode for guidance, explanation, and planning]	

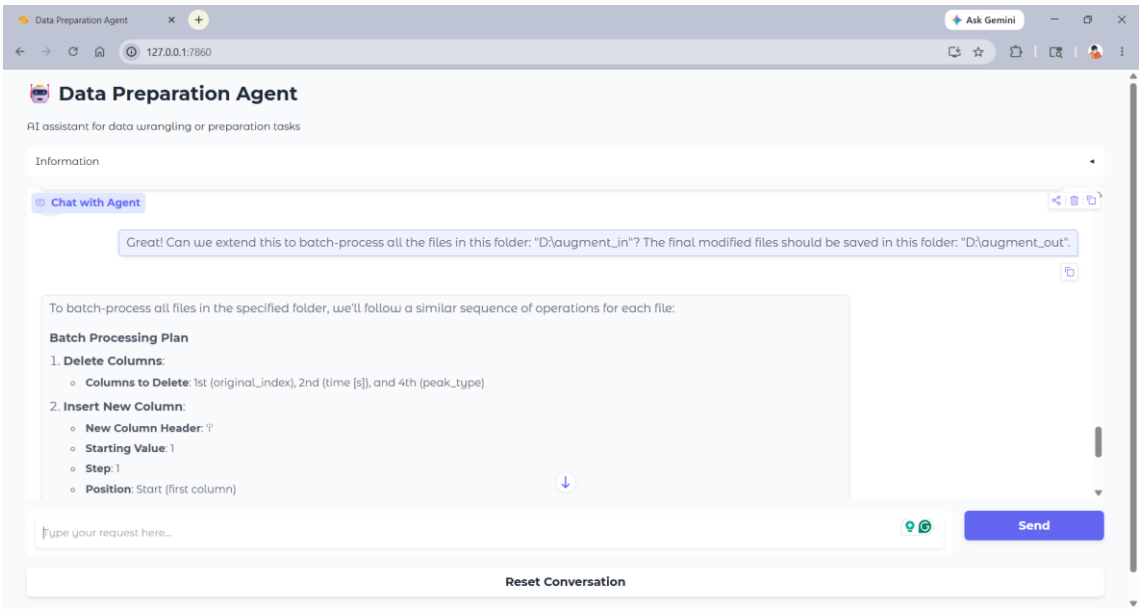


Figure S3-19. Screenprint related to Table S3-19.

Table S3-20. Conversational logs and tool invocation traces for batch-processing augmentation.

User	Plan looks good. Let's do it.
Agent	Batch processing is complete! All files in "D:\augment_in" have been processed, and the modified files have been saved in "D:\augment_out". Here are some of the saved files: D:\augment_out\1-1B_parsed_20260516-170750-444019_segment_10_peaks_20260516-193810-060482_modified_20260516-213002-036077.txt D:\augment_out\1-1B_parsed_20260516-170750-444019_segment_11_peaks_20260516-193810-459441_modified_20260516-213002-220854.txt D:\augment_out\1-1B_parsed_20260516-170750-444019_segment_12_peaks_20260516-193810-992386_modified_20260516-213002-352609.txt D:\augment_out\1-1B_parsed_20260516-170750-444019_segment_13_peaks_20260516-193811-542708_modified_20260516-213002-521933.txt D:\augment_out\1-1B_parsed_20260516-170750-444019_segment_14_peaks_20260516-193811-979074_modified_20260516-213002-637412.txt A total of 40 files were successfully processed with no failures. If you need further assistance or have additional tasks, feel free to let me know!
Backend (Tool Invocation Trace): [Agent] Calling tool: batch_insert [Agent] With arguments: {'input_folder': 'D:\\augment_in', 'output_folder': 'D:\\augment_out', 'column_header': 'i', 'start': 1, 'step': 1, 'position': 'start', 'delete_columns': ['original_index', 'time [s]', 'peak_type'], 'metadata': 'Process: Micro Drilling\nTool: Solid Carbide micro-INOX 8×D drill\nWorkpiece: Titanium Block (Ti6Al4V)\nCycle: 1'} Executing tool: batch_insert Arguments: {'input_folder': 'D:\\augment_in', 'output_folder': 'D:\\augment_out', 'column_header': 'i', 'start': 1, 'step': 1, 'position': 'start', 'delete_columns': ['original_index', 'time [s]', 'peak_type'], 'metadata': 'Process: Micro Drilling\nTool: Solid Carbide micro-INOX 8×D drill\nWorkpiece: Titanium Block (Ti6Al4V)\nCycle: 1'}	

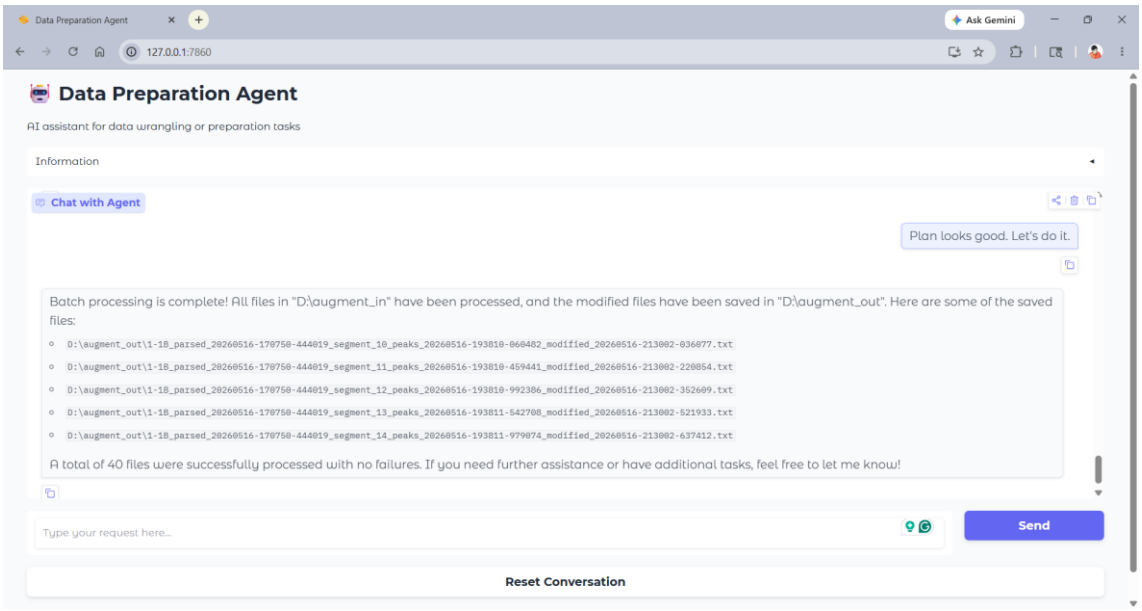


Figure S3-20. Screenprint related to Table S3-20.

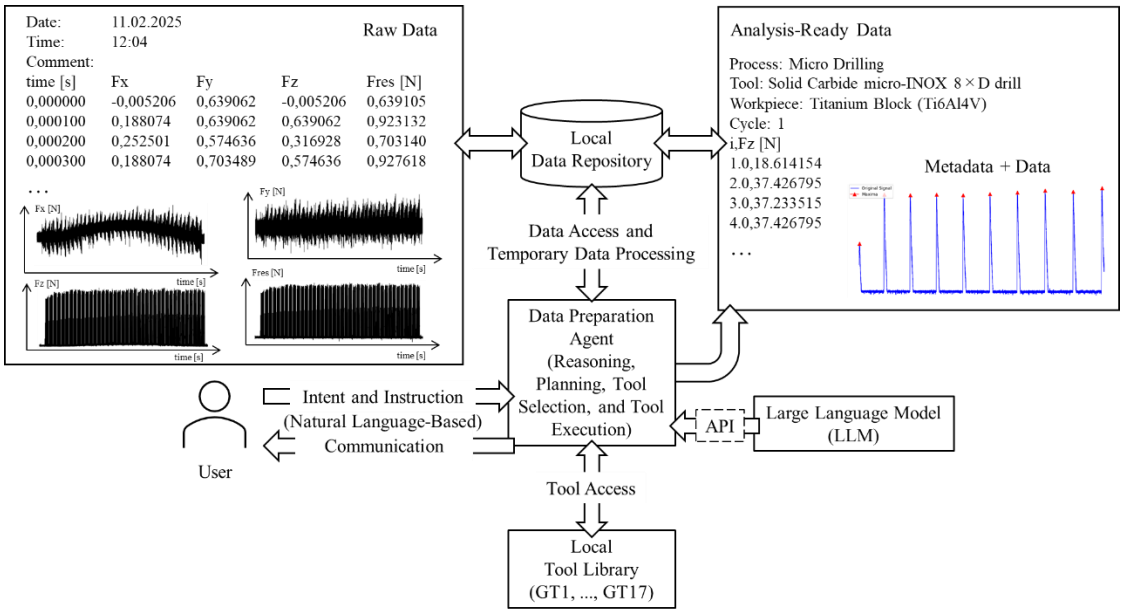


Figure S3-21. Summarizing the above workflow using the Agentic AI-based toolkit.