# CODEX core use cases

This document contains the “key path” scenarios. The intent is to describe the primary user-flow through the application so that completeness can be tested before launch.

**Second priorities: expand utility to larger data sets and make more robust. Critical features for key scenario/use-case**

#### Explain-this UI

- Input: 2+ Features and 1 Selection

- Output: interactive decision tree, accuracy of predictions based on the Feature(s) used

- Visualize *why* the Selection is unique

- Explain *how* the answer was derived

- Save a Model that can be used to categorize another data set

#### Session/state-saving

- Save state and let me continue where I left off (state in URL? Save data on server? Cookies + server? Accounts?)

#### Heat maps

- Make heatmaps an overlay/mode of scatterplots (could initially at least highlight the Heatmaps where the Scatterplot selections are, so you can compare Scatterplots to Heatmaps)

- Fix Heatmap helper text to show full coordinates as well as the count/density in that heat-bin

- Make Heatmaps more like Density plot with more granular size of the heat-bins

#### Chart management and Interactions

- Make it easy to copy/paste the settings used for Clustering so that they can be used when in code later (this is not Code-exporter)

- the clusters produced by a clustering algorithm need to be grouped and have some indication of how they were created (e.g. settings for clustering). Need to separate each “set” of clusters in the UI so you can tell which sets logically go together (e.g. you shouldn’t mix a cluster from algo1 with a cluster from algo2)

- select-color on charts (red) is similar to some of the cluster chart colors (e.g. purply-red) so it’s hard to tell what’s selected across charts when the clustering color is active

- update hide/show on chart vs. selected (as input to algorithms) for Features and Selections

- Indicate which graphs and algorithms are possible when a Feature set is selected

- Box-zoom / Chart interaction improvements

**MVP High-level features**

1. Pose questions of the graphs so that they can answer things for you

- heat vs. scatter -> histogram

1. “Explain this”

- “Applied classification”

- Take a selection and :

- make decision-tree classifier on this selection vs. everything else

- visualize why it’s different or unique

- “which of the 1000 values matter?”

- LUKAS HAS AN EXAMPLE HE CODED

- also explain *how* the answer was derived

1.1. Filter out garbage

- what’s **good/bad**

- can’t do *anything* without filtering your data set first

- relationships between Features to make sure that it makes sense

- min/max/etc. for a Feature

- derived Features

- mask things out that are garbage

- anything empty/null etc.

- normalization

- manual filtration

- interactive sortable heatmap of values

*1.5. Directed* classification/regression

- don’t optimize hyper-parameters for me

- but GUIDE me

1.5. Train a model to predict X (Classification/Regression)

What did I learn? How well did it work? Cross-validate. Generalized test error…

even on the same features you can use the Classification as a way to LEARN ABOUT THE DATA

1.5. Reliability:

- Error handling so it’s not busted

- Saving sessions

- Selecting Features & Selections and interacting with algorithms and charts

2. “**More like this**”

2. Dimensionality reduction:

- “not minimal”

- high user need because most people won’t do it

- back-end needs this in order to do clustering, but the user doesn’t need it critically right now

3. Real-time algorithms that show each round so you can decide when to stop

## CODEX “seed utility” (2) key path scenario

(1) The user goes to codex.jpl.nasa.gov and sees the CODEX user-interface. User Imports a file and decides to load the Chemcam demo data to explore— (2) during the load there’s a loading indicator where the Feature list will go (or where the file will be listed, if Feature list is moved).

(3) User selects all four Features at once (Ti vs. Al, Fe, Mg)— (4) there is a message that a Scatterplot takes two Features as inputs and that they may plot the various combinations of the selected Features if they continue— (5) and they plot out in all permutations when the Scatter plot action is taken.

User will explore a few outlier data points to compare across the various Scatterplots by circling them and seeing where the groups lie across other graphs. (6) User box-zooms into the region of interest and pans around to make sure that their selection is as intended.

(7) User flips one of the Scatterplots into Density Plot (Heat Map?) mode to make sure that the outlier is really an outlier group and not a dense cluster.

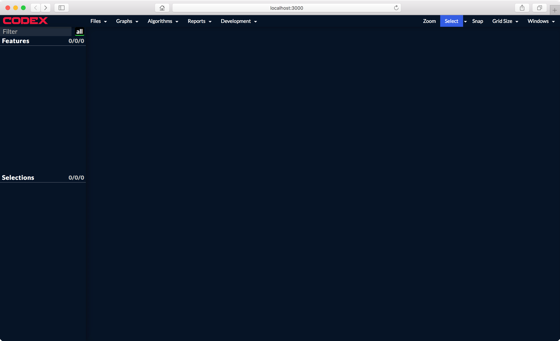
(8) Then user will activate all four (or more) chemical species and perform Clustering. (9) On the Cluster preview page, (10) user will choose help for one or more of the algorithms. (11) User will choose one of the clustering algorithms and select (12,13) execution parameters and outputs. The selection of number of parameters is done visually instead of by flipping an incrementor switch.

(14) User lets the cluster run and exits the web site. (15) The next day, the user comes back to the web site to pick up the analysis.

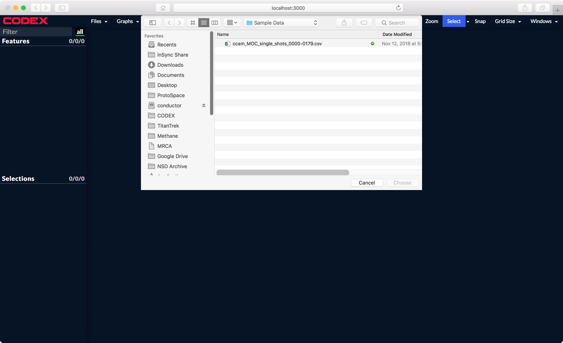
(16) The user will re-color some of the Scatterplots using the new clusters that were created. In the Selections side-bar, the list of clusters created by the Clustering is grouped under a header. (17) Clicking an info button in the header also opens up the settings that were used to generate those clusters/Selections.

### Key path scenario

1. Open the URL



2. Open ChemCam example data and view loading indicator



2b. Features are loaded into left hand panel

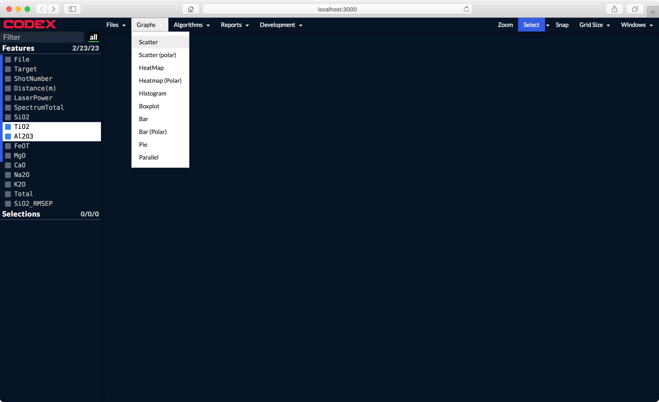
3. Make some scatter plots

V1: Feature -> Chart (can quickly make multiple scatterplots like this)

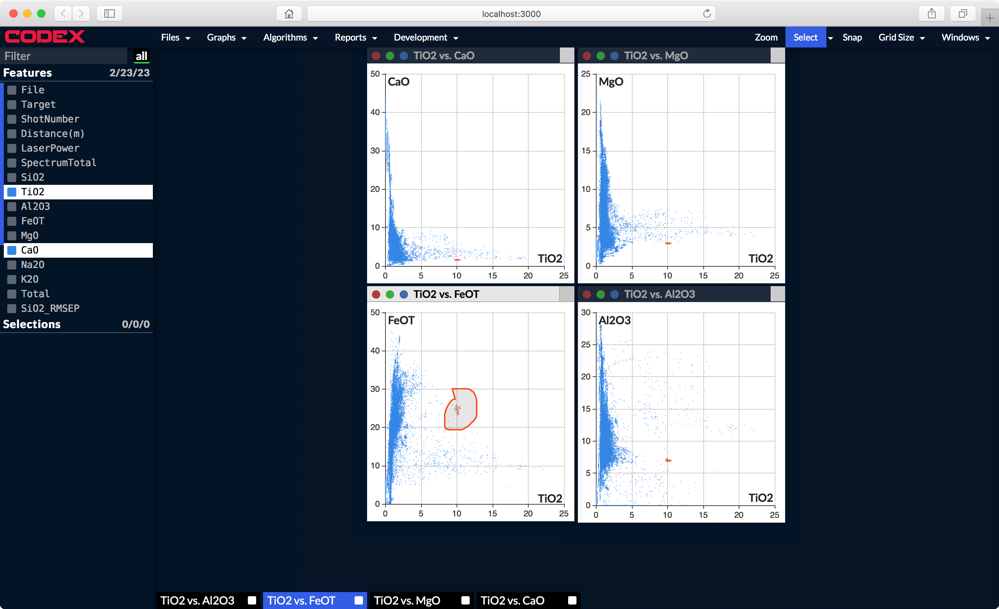
V2: New chart -> Add Features (might need a Correlogram to do n x m scatterplots)

4. Number-of-features helper

5. Plot all permutations on 4 Features



6. Zoom in on a chart (box-zoom and pan)



7. Flip one of the scatterplots into a heat map/density map

8. Run clustering on all four (or more) Features

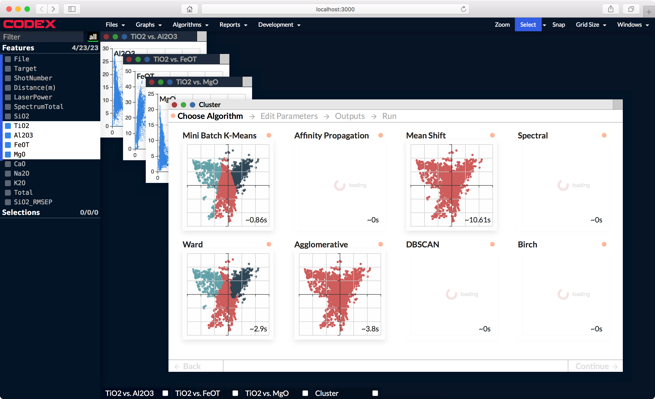
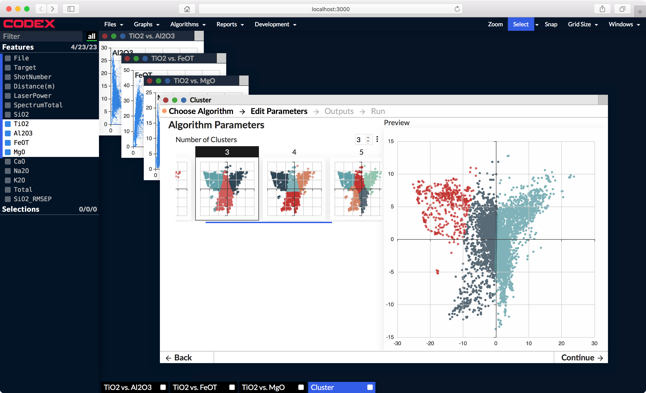
9. Cluster preview

10 choose help for one algorithm

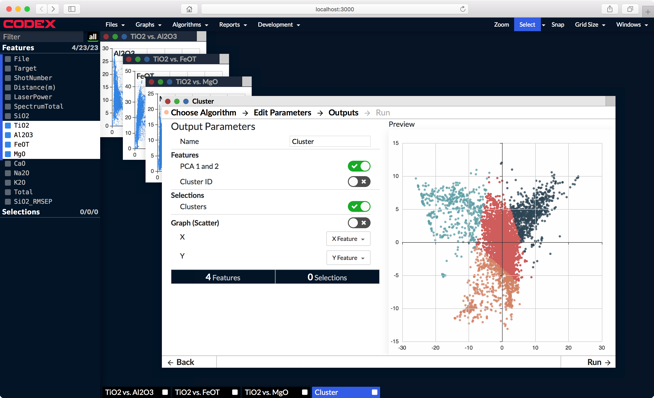
11. choose clustering algo

12. choose number of clusters

13. choose parameters

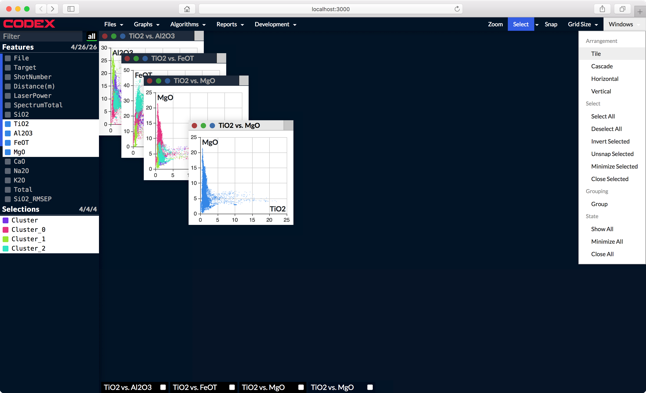
 

14. start clustering and close the app (Save the user’s place) (Do we need to also allow intentional named-session saves on top of an auto-save of the current state?)



15. User returns to completed clustering run

16. Color original scatter plots by the new clusters. Tile windows.



17. User checks the settings used to generate the clusters by clicking widget icon in the title of the Cluster-group header in the Selection panel.

Next the user wants to compare SL classification to the Clustering.

18. Supervised learning (SL): classification. User selects Features and Selections/rows. Options of various SL algorithms with previews (like Clustering, but no visualization). Once processed, an output is a Model (need to figure out what stats/summary visualization to give here).