Orange Data Mining

ClusterGrader

User Manual

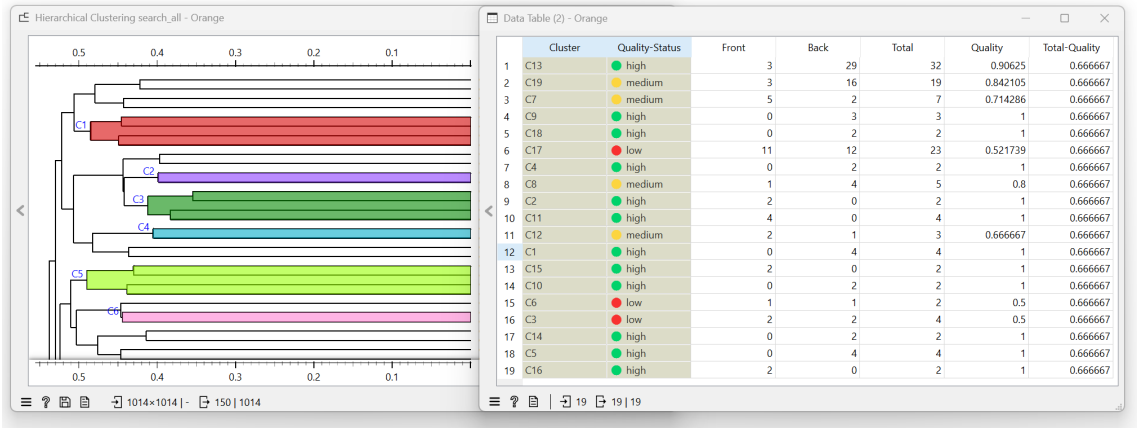
December 14, 2025

Lisa Haußmann, Dominik Schmitt-Klink
Dr. Karsten Tolle
Fachbereich 12, Institut für Informatik
Goethe-Universität Frankfurt am Main

Contents

1 Introduction

This workflow helps researchers and archaeologists to perform and evaluate the clustering of ancient coin images in order to identify groups with similar or related die patterns. The workflow was designed to both perform clustering and to evaluate its quality. It automatically groups images based on visual similarity and then checks how consistent these clusters are. Figure 1 shows the ClusterGrader in action, illustrating how cluster quality is visualized based on the distribution of obverse and reverse images.



	Cluster	Quality-Status	Front	Back	Total	Quality	Total-Quality
1	C13	high	3	29	32	0.90625	0.666667
2	C19	medium	3	16	19	0.842105	0.666667
3	C7	medium	5	2	7	0.714286	0.666667
4	C9	high	0	3	3	1	0.666667
5	C18	high	0	2	2	1	0.666667
6	C17	low	11	12	23	0.521739	0.666667
7	C4	high	0	2	2	1	0.666667
8	C8	medium	1	4	5	0.8	0.666667
9	C2	high	2	0	2	1	0.666667
10	C11	high	4	0	4	1	0.666667
11	C12	medium	2	1	3	0.666667	0.666667
12	C1	high	0	4	4	1	0.666667
13	C15	high	2	0	2	1	0.666667
14	C10	high	0	2	2	1	0.666667
15	C6	low	1	1	2	0.5	0.666667
16	C3	low	2	2	4	0.5	0.666667
17	C14	high	0	2	2	1	0.666667
18	C5	high	0	4	4	1	0.666667
19	C16	high	2	0	2	1	0.666667

Figure 1: The ClusterGrader widget visualizing cluster purity based on obverse and reverse image distributions.

Each coin is typically photographed from two sides, obverse (front) and the reverse (back). During the evaluation, the workflow measures the proportion of front and back images within each cluster. A cluster containing mostly one side type (for example, predominantly obverse images) is considered more homogeneous, while clusters with a balanced mix of both sides tend to indicate lower visual consistency and therefore a lower quality score. The workflow calculates a quality score for each cluster and for the dataset as a whole, allowing researchers to quantify how consistently coins with similar dies have been grouped. Its main benefits include:

- Provides an automated and evaluation of clustering results.
- Quantifies cluster consistency using a quality metric.
- Helps detect mixed or inconsistent clusters that may indicate unsuitable parameter settings.
- Allows direct comparison of different preprocessing steps, distance metrics, and linkage methods.

This project was implemented with Orange version 3.38.1.

2 Overview of the Workflow

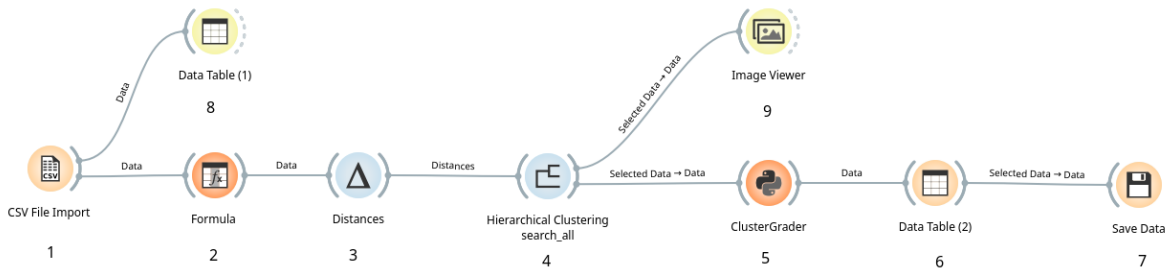


Figure 2: Workflow Overview

The pipeline in Orange Data Mining uses several widgets to import, process, cluster, and analyze image data. The following table lists all widgets used in the correct order and explains their function in the workflow as shown in figure 2.

Step	Widget	Purpose
1	CSV File Import	Load metadata of all coin images (including file names, cluster labels, or additional features).
2	Formula	Adjust the file paths to the image folders as variables.
3	Distances	Compute pairwise distances or similarities between coin images.
4	Hierarchical Clustering (search_all)	Perform clustering on the dataset using all available data. (Modified by Sebastian Gampe, member of the supervising team.)
5	ClusterGrader	Analyze the clusters by counting obverse and reverse images to assess clustering quality.
6	Data Table (2)	Review the cluster assignments and their quality in tabular form.
7	Save Data	Save the clustered dataset and their quality.
8	Data Table (1)	Inspect the imported data in tabular form.
9	Image Viewer	Visually inspect which images belong to each cluster for clustering with search_all.

3 Widget Overview and Usage

This section explains the function and typical use of each widget within the ClusterGrader workflow. The widgets can be combined flexibly depending on the research question and available data. While some steps are essential, others are optional and serve to verify or refine the dataset.

3.1 CSV File Import

The CSV File Import widget is used to load the metadata of all research images into Orange. The file typically contains a matrix with a number, which describes the similarity between every image, for example based on visual similarity measures or extracted features. This file serves as the foundation for further clustering. The CSV file should contain at least the image file names and any associated attributes such as cluster labels or die identifiers. The image file names should either contain an "o" (e.g. coin1o.jpg) or an "r" (e.g. coin2r.jpg). "o" is the obverse and "r" the reverse side of the coins. Additional numeric or categorical data can also be included. After importing, the widget outputs a data table that can be used as input for the next components or to verify the correct import.

3.2 Formula

The Formula widget is used to define the exact file paths to the image folders that should be included in the clustering process. Each variable in the widget represents one processing stage of the coin images, for example, grayscale versions, denoised images, histogram equalization, or cropped variants.
By defining several path variables, researchers can easily switch between or compare different preprocessing steps. This allows flexible experimentation with different preprocessing conditions, enabling comparisons between visual variants of the same coins and their effect on clustering results.
In the Variable Definitions field, a new variable (for example path_1) is created. Its value is the complete path to the folder containing the desired image version, followed by '+name'. The 'name' variable comes from the dataset and represents the individual image file name from the CSV input. Figure 3 illustrates this setup, showing how different image folder paths are defined within the Formula widget.

Examples of correct syntax:

- **Windows:**
  "C:\\Users\\max\\images\\grayscale\\"+name
- **Linux:**
  "/home/max/images/grayscale/"+name
- **macOS:**
  "/Users/max/images/grayscale/"+name

Replace the part of the path (e.g. /home/max/images/grayscale/) with the actual location where your images are stored. The final '+name' part should not be replaced or removed, as it automatically appends each image filename from the dataset to the selected folder path.

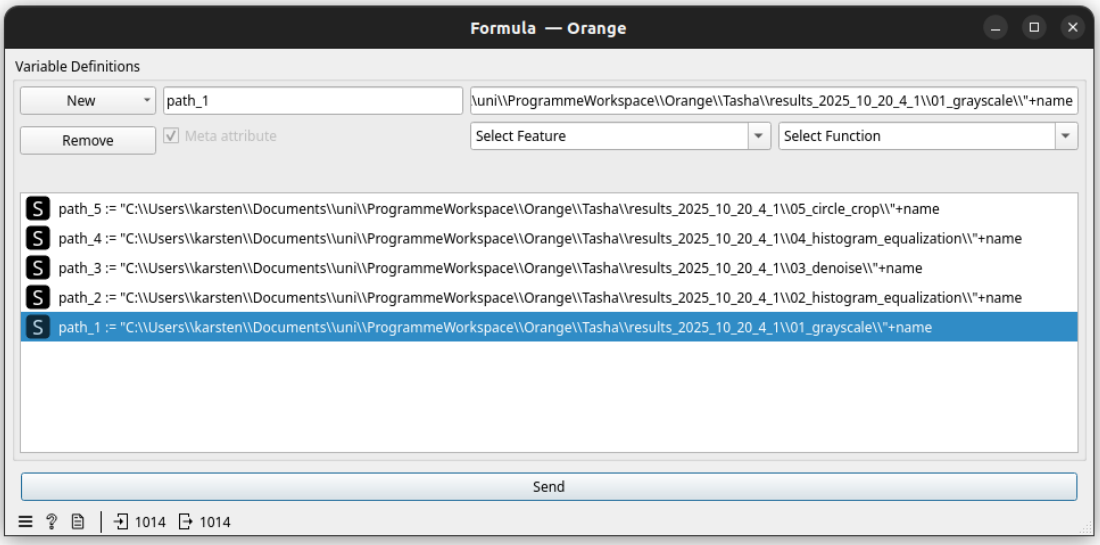


Figure 3: Formula Widget with different paths

Each folder path can represent a specific processing level of the images, such as:

- grayscale: grayscale conversion
- 1_histogram_equalization: enhanced contrast
- denoised: noise reduction
- 2_histogram_equalization: alternative histogram normalization
- circle_crop: cropped and centered images

By adding multiple path variables (e.g. path_1, path_2, path_3), users can experiment with different preprocessing versions of the dataset within the same workflow and analyze how each affects the clustering results.

3.3 Distances

The Distances widget calculates pairwise distances or similarities between the image features. It is one of the core components of the workflow and defines how "similar" two coin images are considered. Different distance metrics can be selected depending on the research focus, for example, Cosine or Pearson (absolute) (see Section Distance Metrics for details). The following figure 4 shows the Distances widget and the available similarity metrics that can be selected for the clustering process.

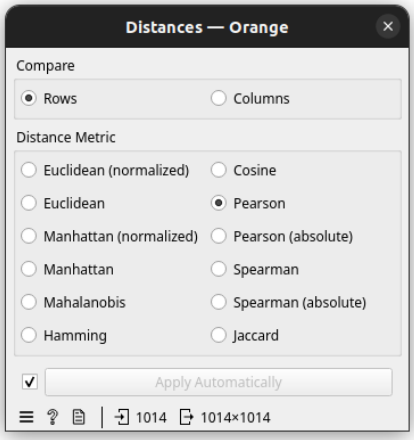


Figure 4: Distance Widget with different distances

3.4 Hierarchical Clustering (search_all)

This custom widget performs the hierarchical clustering on the dataset using all available data. It was modified by a Sebastian Gampe, member of the supervising research group. The widget automatically searches for suitable clustering and outputs the resulting group assignments. It represents the main clustering step in the workflow.
Within the widget interface, users can select which image paths (e.g. path_1, path_2, path_3, etc.) should be used for the clustering. By changing the selected path, one can directly compare how the visual preprocessing affects the cluster formation.
The widget allows the user to choose the desired linkage method (e.g. Average, Complete, or Weighted). This setting determines how the distances between image groups are calculated during the hierarchical merging process and therefore has a major influence on the final cluster structure (see Section Linkage Methods for details). Figure 5 shows the Hierarchical Clustering (search_all) widget with the selectable image paths, the linkage options and the resulting group assignments.

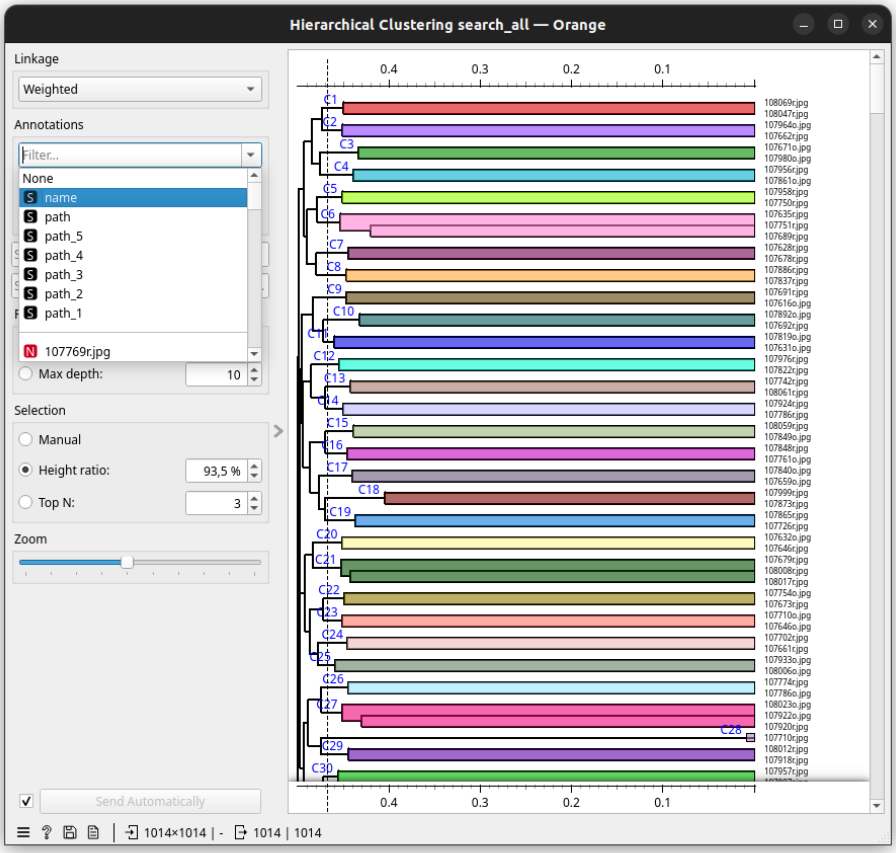


Figure 5: Hierarchical Clustering with search_all widget

3.5 ClusterGrader

The ClusterGrader widget contains the custom script that evaluates the internal consistency of each cluster. It counts how many obverse (o.jpg) and reverse (r.jpg) images are present and calculates a quality score based on their ratio. The resulting values describe how homogeneous each cluster is in terms of the depicted coin side (obverse vs. reverse) and serve as a proxy for clustering reliability.

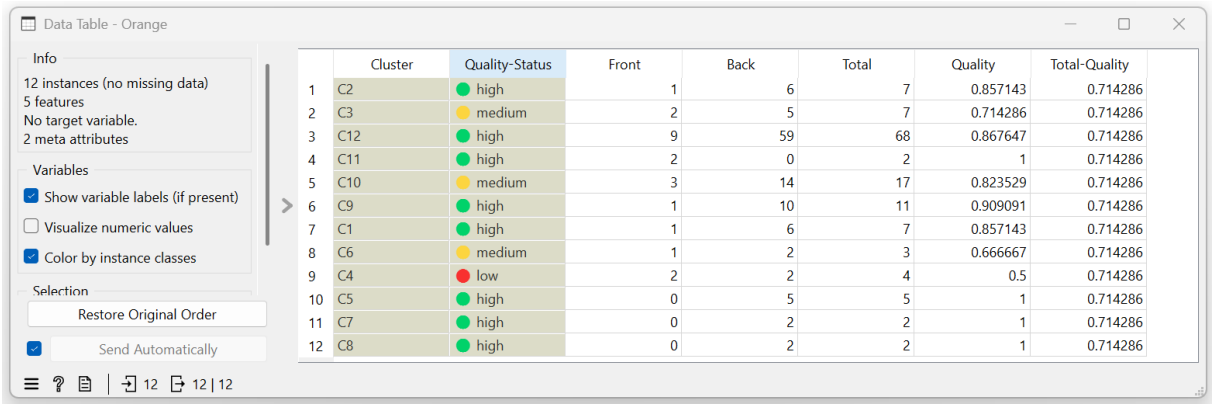
For users interested in reviewing or extending the underlying code of the ClusterGrader, the complete implementation is available on GitHub at: `https://github.com/Garro326/ClusterGrader`.

3.6 Data Table (2)

The final Data Table (2) displays the calculated values, the number of front and back images per cluster, total counts, the local cluster quality and the global average quality. Additionally, the individual cluster quality scores are visualized using a traffic-light system: green for values ≥ 0.85, yellow for values < 0.85 and ≥ 0.65, and red for all scores below 0.65.

These results can be sorted, exported, or directly compared across different clustering

configurations. Figure 6 shows the ClusterGrader output table with the traffic-light visualization of cluster quality.



	Cluster	Quality-Status	Front	Back	Total	Quality	Total-Quality
1	C2	high	1	6	7	0.857143	0.714286
2	C3	medium	2	5	7	0.714286	0.714286
3	C12	high	9	59	68	0.867647	0.714286
4	C11	high	2	0	2	1	0.714286
5	C10	medium	3	14	17	0.823529	0.714286
6	C9	high	1	10	11	0.909091	0.714286
7	C1	high	1	6	7	0.857143	0.714286
8	C6	medium	1	2	3	0.666667	0.714286
9	C4	low	2	2	4	0.5	0.714286
10	C5	high	0	5	5	1	0.714286
11	C7	high	0	2	2	1	0.714286
12	C8	high	0	2	2	1	0.714286

Figure 6: ClusterGrader Table with traffic-light system

3.7 Save Data

The Save Data widget stores the clustered dataset to disk for later reuse or external analysis. This ensures reproducibility and allows further evaluation in other tools. Saving the data is optional but recommended.

Data Table (1)

The Data Table widget allows the user to inspect the data at any point in the workflow. It is mainly used to verify that all attributes have been correctly imported or computed. Opening this widget is optional but recommended before distance calculation or clustering.

3.8 Image Viewer

The Image Viewer widget visualizes the images contained in one or multiple selected clusters. This helps to evaluate whether the clustering corresponds to numismatic expectations. The widget can be placed at different points in the workflow to visually inspect intermediate or final results.

4 Distance Metrics

The workflow supports several distance metrics and linkage methods. In practice, two configurations have proven particularly meaningful for numismatic image clustering, as they reflect different aspects of die similarity and visual variation.

4.1 Cosine Distance

Cosine distance measures the angle between image feature vectors, focusing on their orientation rather than magnitude. It captures how similar the structures or shapes are, regardless of overall brightness or contrast. This makes it highly useful for coin images photographed under different lighting conditions. Using cosine distance helps to group coins that share similar engravings or die characteristics, even if illumination differs.
In summary, cosine distance is well suited for detecting general structural similarities between coin images and remains robust even when illumination or contrast vary.

4.2 Pearson (Absolute) Distance

Pearson correlation distance compares the linear relationship between two feature vectors. When used in its absolute form, it measures how strongly two images vary together, ignoring whether one is brighter or darker. This emphasizes the general similarity of surface patterns and relief structures rather than pixel-level details. In die studies, Pearson (absolute) can reveal coins produced with similar but not identical dies, capturing subtle engraving parallels.
In summary, Pearson (absolute) distance is more sensitive to fine engraving details, capturing subtle variations in die patterns by measuring how strongly image features vary together.

4.3 Overview

The following table provides an overview of the different distance metrics available in Orange.

Distance Metric	Description
Euclidean	The straight-line distance between two points in feature space. Measures geometric similarity.
Manhattan	The sum of absolute differences across all attributes. More robust to outliers than Euclidean.
Cosine	Measures the cosine of the angle between two vectors; Orange computes cosine *distance* as (1 - similarity). Useful for structural similarity independent of brightness.
Jaccard	Ratio of intersection over union of two sets. Suitable for binary or categorical features.
Spearman	Distance based on rank correlation of values, scaled into the ([0,1]) interval. Captures monotonic relationships.
Spearman (absolute)	Like Spearman, but uses absolute ranks. More tolerant to sign changes in the features.

Pearson	Linear correlation between values, mapped to a distance in ([0,1]). Detects linear similarity.
Pearson (absolute)	Pearson correlation computed on absolute values, capturing pattern similarity regardless of sign.
Hamming	Counts how many feature values differ between two items. Used for boolean or categorical features.
Bhattacharyya	Measures similarity between probability distributions; not a true metric (violates triangle inequality). Good for histogram-like features.

For more detailed descriptions and settings of all distance metrics, see the official Orange documentation at the link below.
https://orange3.readthedocs.io/projects/orange-visual-programming/en/latest/widgets/unsupervised/distances.html

5 Linkage Methods

5.1 Average Linkage

Average linkage computes the average distance between all pairs of elements in two clusters. It produces balanced, stable clusters and avoids chaining effects. For die analysis, this method helps identify general families of dies or stylistic groups, providing a broader picture of related coin types.

5.2 Complete Linkage

Complete linkage uses the maximum distance between two elements in different clusters. It forms compact and well-separated groups. In the context of coin analysis, this is suitable when focusing on very close matches, coins almost certainly struck from the same die, while excluding more distant similarities.

Other Metrics and Linkages

For other distance metrics and linkage types (e.g., Euclidean, Manhattan, Single Linkage, Ward's method), please refer to the official Orange Data Mining documentation:
https://orangedatamining.com/widget-catalog/

6 Interpretation

The ClusterGrader workflow provides both a visual and quantitative way to assess how consistently coin images have been grouped according to their die characteristics. The structure of the resulting clusters depends mainly on the selected distance metric and linkage method, which determine how the underlying similarity between images is interpreted.

Cosine distance highlights general structural features and is robust to illumination or contrast differences, often producing broader clusters that group images sharing similar overall shapes or engravings. Pearson (absolute) distance, in contrast, is more sensitive to fine variations in relief and engraving patterns, which can lead to narrower clusters that capture subtle stylistic or die-related similarities. These differences allow researchers to decide whether they want to identify broader die families or focus on more precise engraving parallels.

The chosen linkage method further shapes the character of the clusters. Average linkage produces balanced and stable groups that reflect general stylistic relationships, making it suitable for identifying wider die families or thematic similarities. Complete linkage forms compact, tightly related clusters by prioritizing the greatest pairwise distances; this often isolates visually very close images and may reveal coins that were produced with highly similar or even identical dies.

The ClusterGrader output summarizes the effects of these settings by reporting, for each cluster, the number of obverse and reverse images as well as a quality score based on their ratio. Clusters dominated by a single side type tend to be more visually coherent, whereas clusters containing both sides exhibit a lower degree of internal consistency.