

Introduction to Data Mining

Working notes for the hands-on course for PhD students at University of Ljubljana

These notes include Orange workflows and visualizations we will construct during the course.

The working notes were prepared by Blaž Zupan and Janez Demšar with help from the members of the Bioinformatics Lab in Ljubljana that develop and maintain Orange.

Welcome to the course on Introduction to Data Mining! This course is designed for students and researchers of life sciences, engineering, and statistics. You will see how common data mining tasks can be accomplished without programming. We will use Orange to construct visual data mining workflows. Many similar data mining environments exist, but the lecturers prefer Orange for one simple reason—they are its authors.

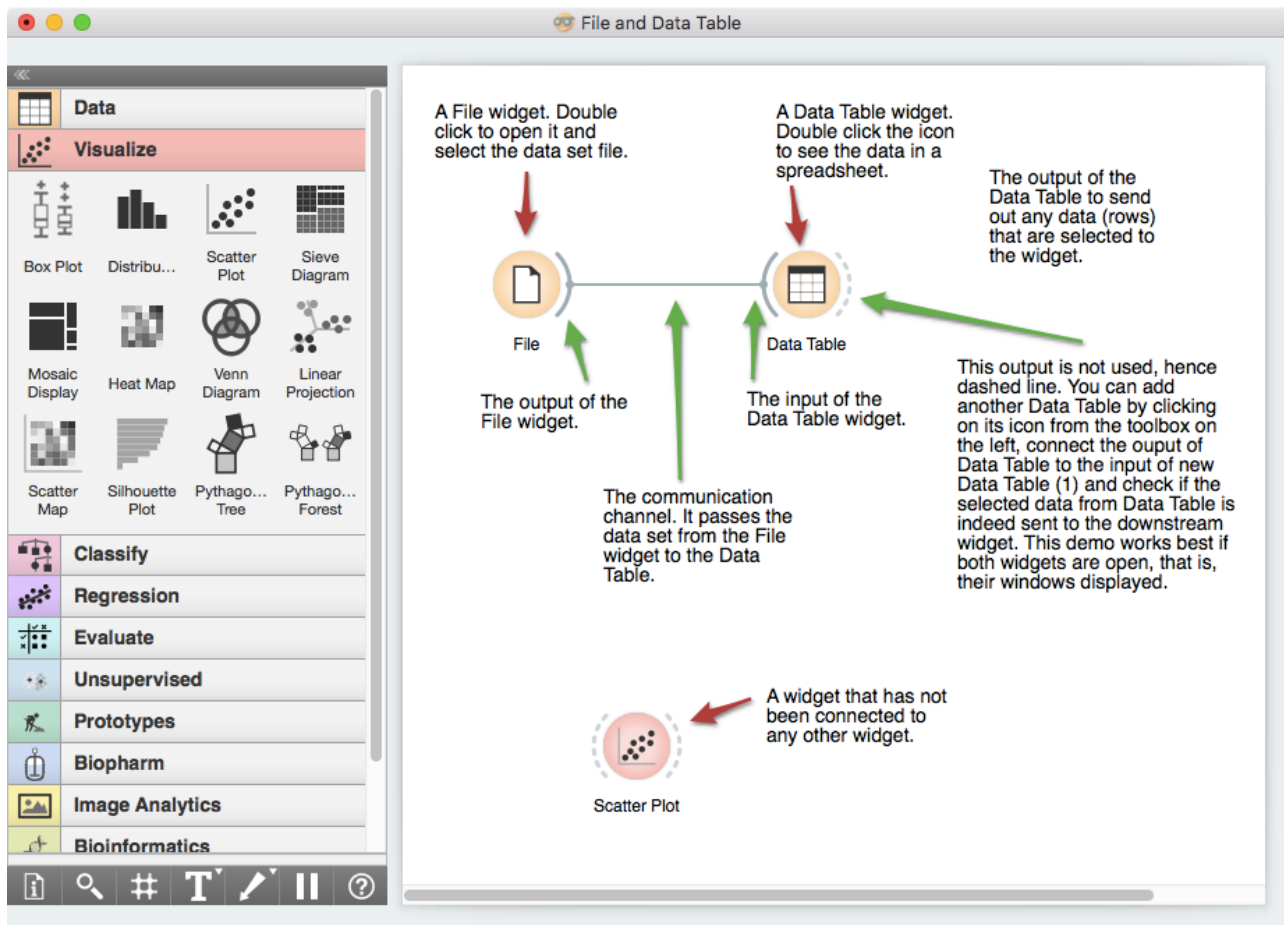
If you haven't already installed Orange, please download the installation package from <http://orange.biolab.si>.



Attribution-NonCommercial-NoDerivs
CC BY-NC-ND

Lesson 1: Workflows in Orange

Orange workflows consist of components that read, process and visualize data. We call them “widgets.” We place the widgets on a drawing board (the “canvas”). Widgets communicate by sending information along with a communication channel. An output from one widget is used as input to another.

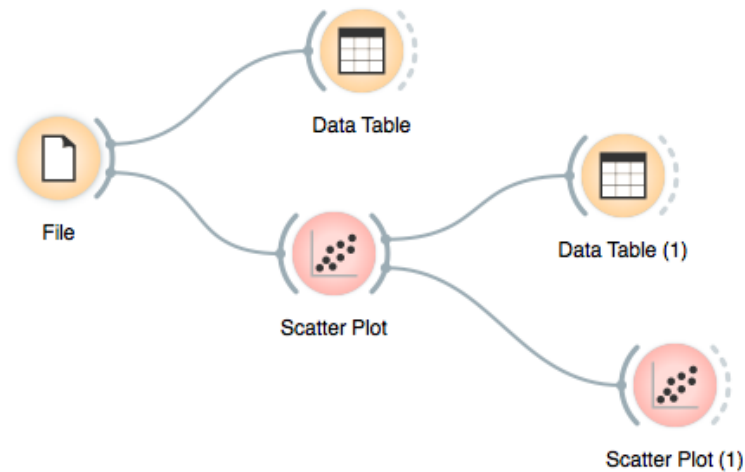


A screenshot above shows a simple workflow with two connected widgets and one widget without connections. The outputs of a widget appear on the right, while the inputs appear on the left.

We construct workflows by dragging widgets onto the canvas and connecting them by drawing a line from the transmitting widget to the receiving widget. The widget’s outputs are on the right and the inputs on the left. In the workflow above, the File widget sends data to the Data Table widget.

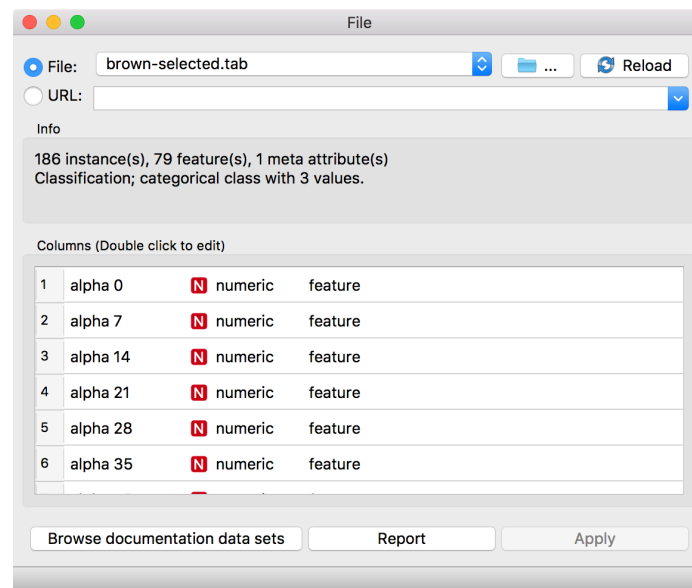
Start by constructing a workflow that consists of a File widget, two Scatter Plot widgets, and two Data Table widgets:

Workflow with a File widget that reads data from disk and sends it to the Scatter Plot and Data Table widget. The Data Table renders the data in a spreadsheet, while the Scatter Plot visualizes it. Selected data points from the Scatterplot are sent to two other widgets: Data Table (1) and Scatter Plot (1).



The File widget reads data from your local disk. Open the File Widget by double clicking its icon. Orange comes with several preloaded data sets. From these (“Browse documentation data sets...”), choose brown-selected.tab, a yeast gene expression data set.

Orange workflows often start with a File widget. The brown-selected data set comprises 186 rows (genes) and 81 columns. Out of the 81 columns, 79 contain gene expressions of baker’s yeast under various conditions, one column (marked as a “meta attribute”) provides gene names, and one column contains the “class” value or gene function.



After you load the data, open the other widgets. In the Scatter Plot widget, select a few data points and watch as they appear in widget Data Table (1). Use a combination of two Scatter Plot widgets,

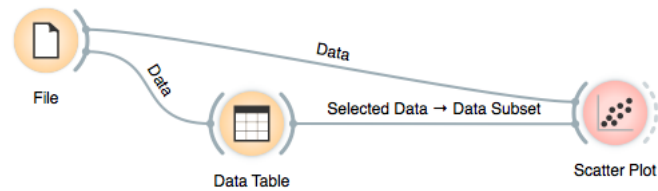
where the second scatter plot shows a detail from a smaller region selected in the first scatterplot.

Following is more of a side note, but it won't hurt. Namely, the scatter plot for a pair of random features does not provide much information on gene function. Does this change with a different choice of feature pairs in the visualization? Rank projections (the button on the top left of the Scatter Plot widget) can help you find a good feature pair. How do you think this works? Could the suggested pairs of features be useful to a biologist?

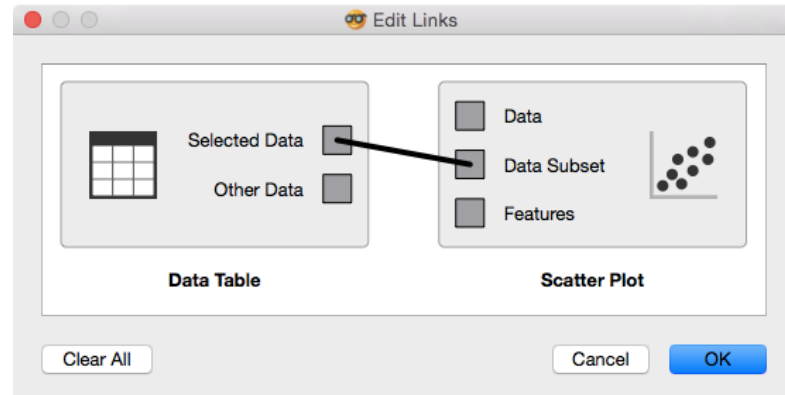


We can connect the output of the Data Table widget to the Scatter Plot widget to highlight the chosen data instances (rows) in the scatter plot.

In this workflow, we have turned on the option “Show channel names between widgets” in File → Preferences.



How does Orange distinguish between the primary data source and the data selection? It uses the first connected signal as the entire data set and the second one as its subset. To make changes or to check what is happening under the hood, double click on the line connecting the two widgets.



Orange comes with a basic set of widgets for data input, preprocessing, visualization and modeling. For other tasks, like text mining, network analysis, and bioinformatics, there are add-ons. Check them out by selecting “Add-ons...” from the options menu.

The rows in the data set we are exploring in this lesson are gene profiles. We can use the Gene Info widget from the Bioinformatics add-on to get more information on the genes we selected in any of the Orange widgets.

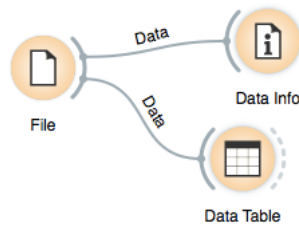


Lesson 2: Basic Data Exploration

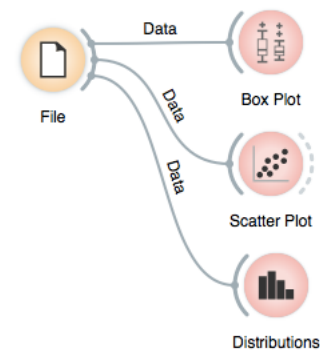
Let us consider another problem, this time from clinical medicine. We will dig for something interesting in the data and explore it a bit with various widgets. You will get to know Orange better and also learn about several interesting visualizations.

We will start with an empty canvas; to clean it from our previous lesson, use either File→New or select all the widgets and remove them (use the backspace/delete key, or Cmd-backspace if you are on Mac).

Now again, add the File widget and open another documentation data set: heart_disease. How does the data look?

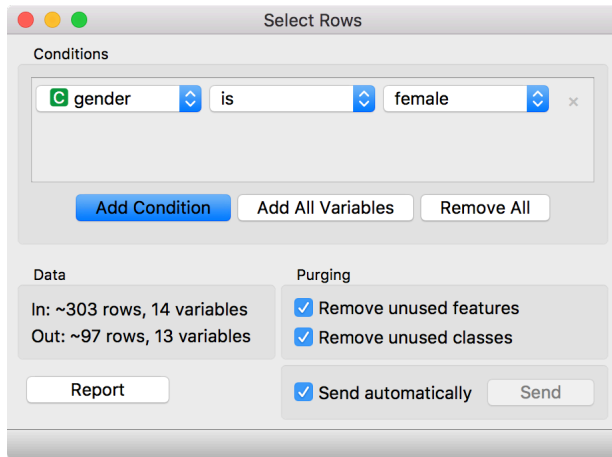
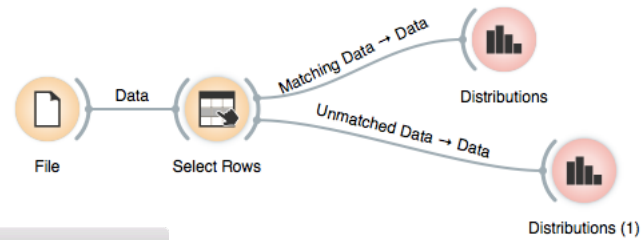


Let us check whether standard visualizations tell us anything interesting. (Hint: look for gender differences. These are always interesting and occasionally even real.)

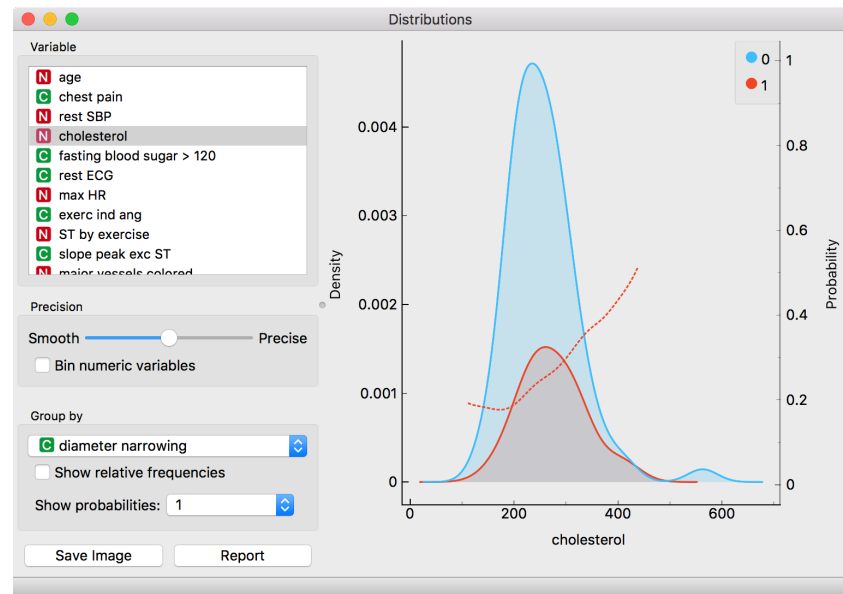


The two Distributions widgets get different data: the upper gets the selected rows, and the lower gets the others. Double-click the connection between the widgets to access setup dialog, as you've learned in the previous lesson.

Data can also be split by the value of features — in this case, gender — and analyze it separately.

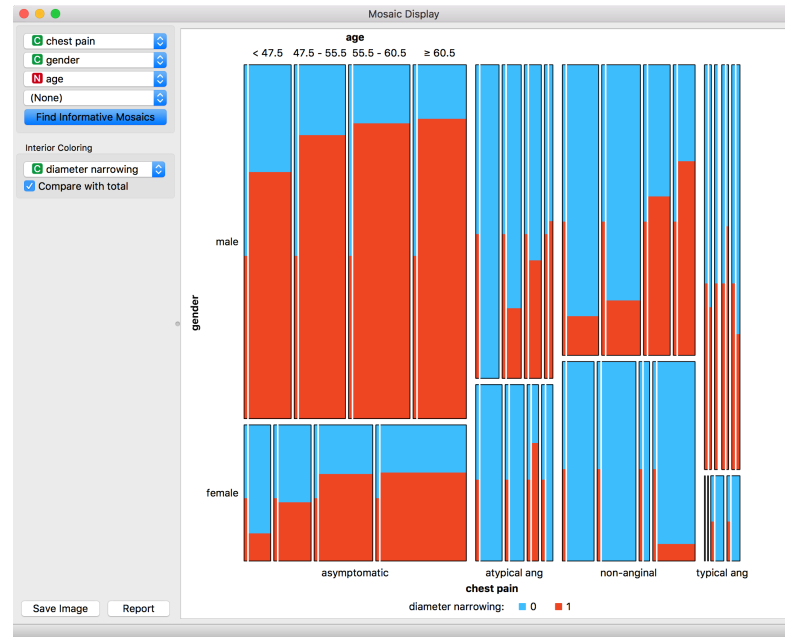


In the Select Rows widget, we choose the female patients. You can also add other conditions. Selection of data instances works well with visualization of data distribution. Try having at least two widgets open at the same time and explore the data.



There are two less known — but great — visualizations for observing interactions between features.

You can play with the widget by trying different combinations of 1-4 features.

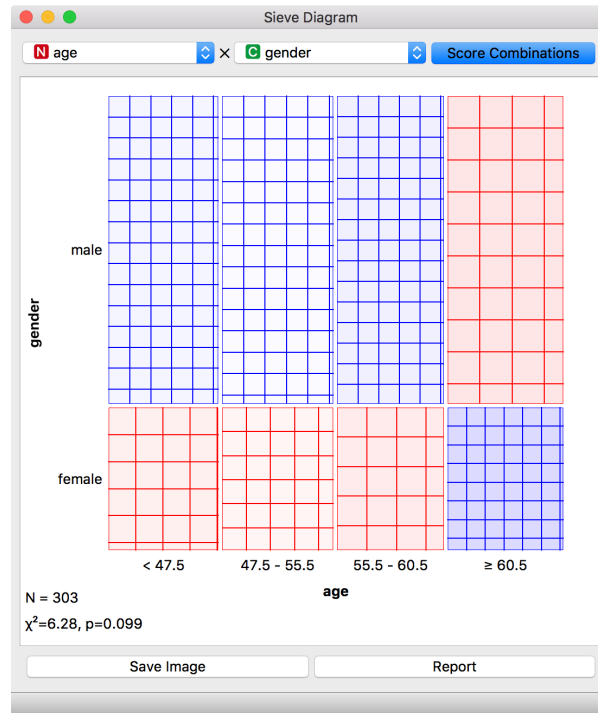


The mosaic display shows a rectangle split into columns with widths reflecting the prevalence of different types of chest pain. Each column is then further split vertically according to gender distributions within the column. The resulting rectangles are divided again horizontally according to age group sizes. Within the resulting bars, the red and blue areas represent the outcome distribution for each group and the tiny strip to the left of each shows the overall distribution.

What can we conclude from this diagram?

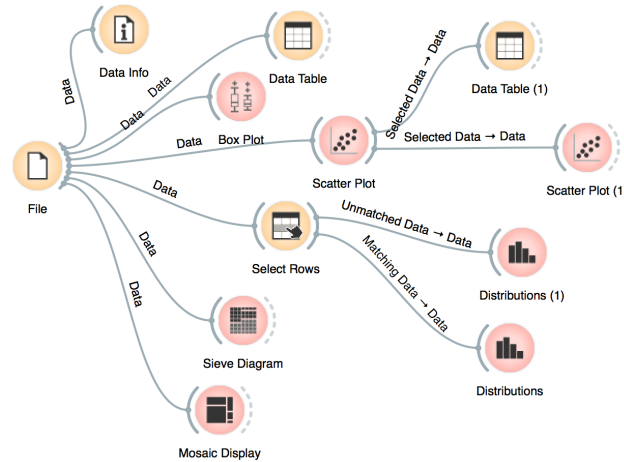
Another visualization, Sieve diagram, also splits a rectangle horizontally and vertically, but with independent cuts, so the areas correspond to the expected number of data instances assuming the observed variables are independent. For example, $1/4$ of patients are older than 60, and $1/3$ of patients are female, so the area of the bottom right rectangle is $1/12$ of the total area. With roughly 300 patients, we would expect $1/12 \times 300 = 25$ older women in our data. There are 34. Sieve diagram shows the difference between the expected and the observed frequencies by the grid density and the color of the field.

See the Score Combinations button? Guess what it does? And how it scores the combinations? (Hint: there are some Greek letters at the bottom of the widget.)

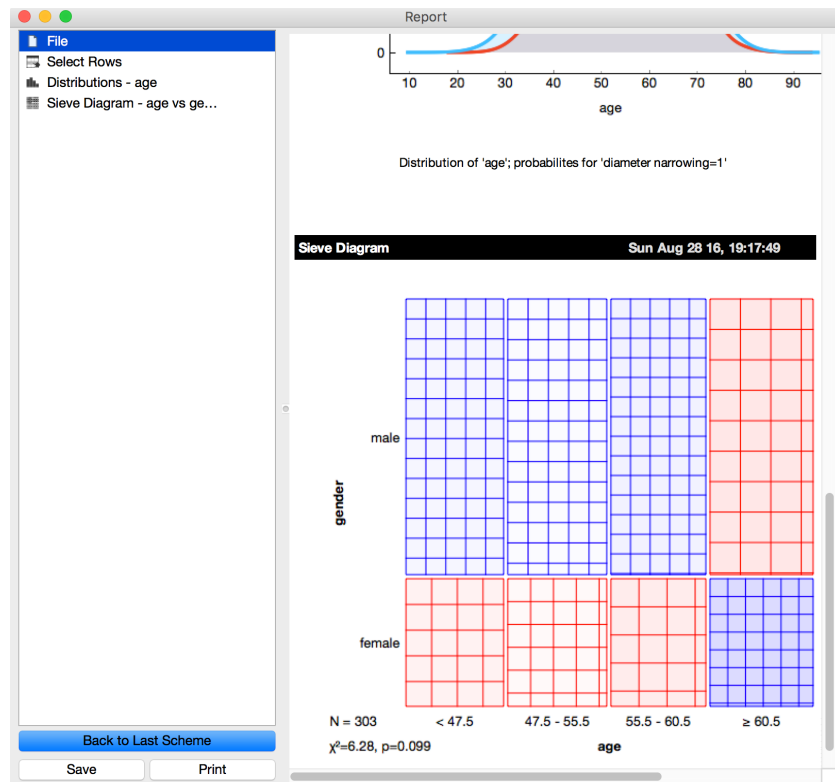


Lesson 3: Saving Your Work

If you followed the instructions so far — except for those about removing widgets — your workflow might look like this.



You can save it (File→Save) and share it with your colleagues. Just don't forget to put the data files in the same directory as the file with the workflow.

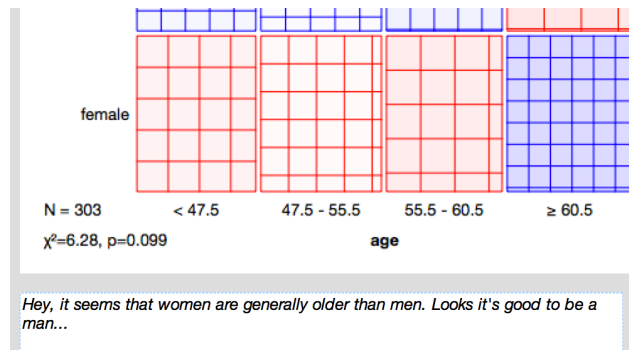


One more trick: Pressing Ctrl-C (or ⌘-C, on Mac) copies a visualization to the clipboard, so you can paste it to another application.

Widgets also have a Report button, which you can use to keep a log of your analysis. When you find something interesting, like an unexpected Sieve Diagram, just click Report to add the graph to your log. You can also add reports from the widgets on the path to this one, to make sure you don't forget anything relevant.

Clicking on the part of the report also allows you to add a comment.

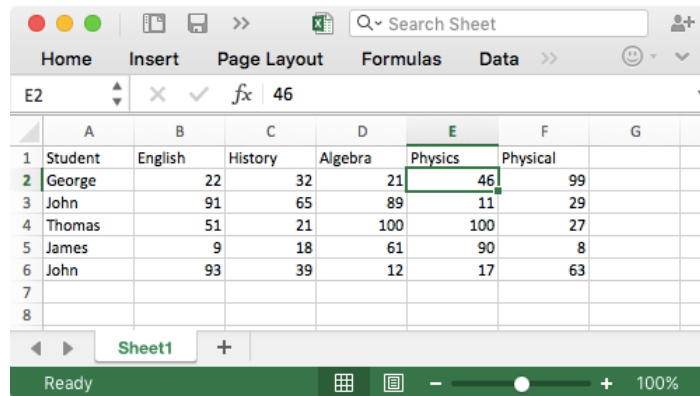
Clicking on a part of the report also allows you to add a comment.



You can save the report as HTML or PDF, or to a file that includes all workflows that are related to the report items and which you can later open in Orange. In this way, you and your colleagues can reproduce your analysis results.

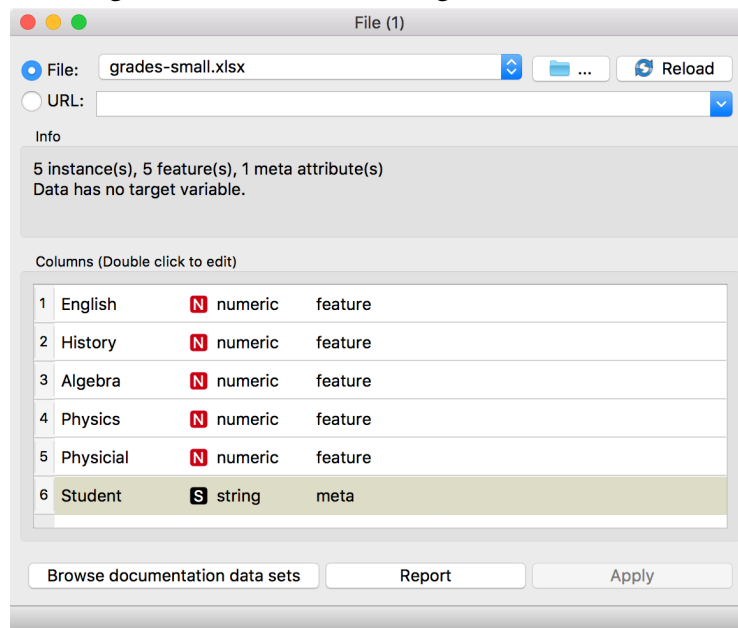
Lesson 4: Loading Your Own Data Set

The data sets we have worked with in previous lessons come with Orange installation. Orange can read data from spreadsheet file formats which include tab and comma separated and Excel files. Let us prepare a data set (with school subjects and grades) in Excel and save it on a local disk.



	A	B	C	D	E	F	G
1	Student	English	History	Algebra	Physics	Physical	
2	George	22	32	21	46	99	
3	John	91	65	89	11	29	
4	Thomas	51	21	100	100	27	
5	James	9	18	61	90	8	
6	John	93	39	12	17	63	
7							
8							

In Orange, we can use the File widget to load this data.



File (1)

File: grades-small.xlsx

URL:

Info

5 instance(s), 5 feature(s), 1 meta attribute(s)
Data has no target variable.

Columns (Double click to edit)

1	English	N	numeric	feature
2	History	N	numeric	feature
3	Algebra	N	numeric	feature
4	Physics	N	numeric	feature
5	Physical	N	numeric	feature
6	Student	S	string	meta

Browse documentation data sets Report Apply

Looks ok. Orange has correctly guessed that student names are character strings and that this column in the data set is special, meant to provide additional information and not to be used for modeling (more about this in the coming lectures). All other columns are numeric features.

It is always good to check if Orange read the data correctly. We can connect our File widget with the Data Table widget,



and double click on the Data Table to see the data in the spreadsheet format.

The screenshot shows the Orange Data Table widget interface. On the left, there is an 'Info' panel with the following details: 5 instances (no missing values), 5 features (no missing values), No target variable, and 1 meta attribute (no missing values). Below this is a 'Variables' section with three checked options: 'Show variable labels (if present)', 'Visualize numeric values', and 'Color by instance classes'. The 'Selection' section has 'Select full rows' checked. At the bottom of the left panel are buttons for 'Restore Original Order', 'Report', and 'Send Automatically' (checked). The main area of the widget displays a spreadsheet with the following data:

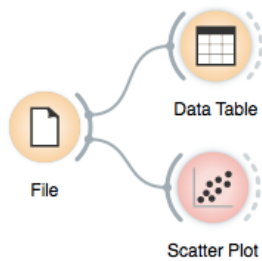
	Student	English	History	Algebra	Physics	Physical
1	George	22.000	32.000	21.000	46.000	99.000
2	John	91.000	65.000	89.000	11.000	29.000
3	Thomas	51.000	21.000	100.000	100.000	27.000
4	James	9.000	18.000	61.000	90.000	8.000
5	John	93.000	39.000	12.000	17.000	63.000

Nice, everything is here.

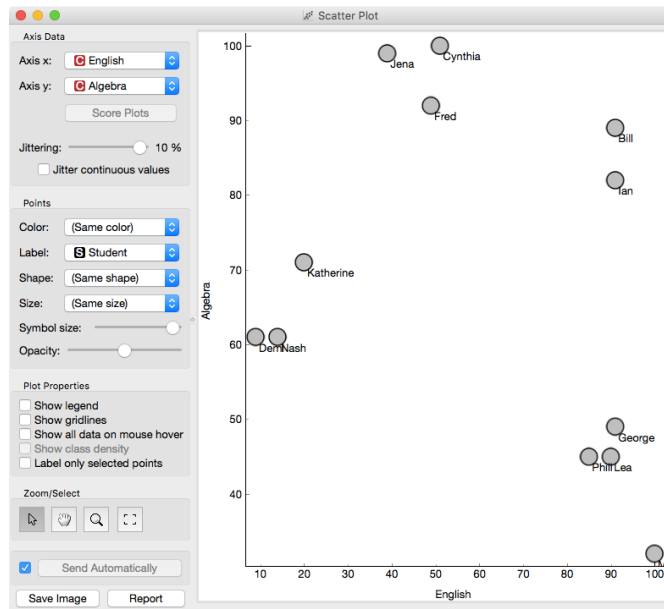
We can also use Google Sheets, a free online spreadsheet alternative. Then, instead of finding the file on the local disk, we would enter its URL address to the File widget's URL entry box.

There is more to input data formatting and loading. We can define the type and kind of the data column, specify that the column is a web address of an image, and more. But enough for the first day. If you would like to dive deeper, check out the [documentation page on Loading your Data](#), or [a video](#) on this subject.

In the class, we will introduce clustering using a simple data set on students and their grades in English and Algebra. Load the data set from <http://file.biolab.si/files/grades2.tab>.



Student	English	Algebra
1 Bill	91.000	89.000
2 Cynthia	51.000	100.000
3 Demi	9.000	61.000
4 Fred	49.000	92.000
5 George	91.000	49.000
6 Ian	91.000	82.000
7 Jena	39.000	99.000
8 Katherine	20.000	71.000
9 Lea	90.000	45.000
10 Maya	100.000	32.000
11 Nash	14.000	61.000
12 Phill	85.000	45.000



How do we measure the similarity between clusters if we only know the similarities between points? By default, Orange computes the average distance between all their pairs of data points; this is called average linkage. We could instead take the distance between the two closest points in each cluster (single linkage), or the two points that are furthest away (complete linkage).

Lesson 5: Hierarchical Clustering

Say that we are interested in finding clusters in the data. That is, we would like to identify groups of data instances that are close together, similar to each other. Consider a simple, two-featured data set (see the side note) and plot it in the Scatter Plot. How many clusters do we have? What defines a cluster? Which data instances belong to the same cluster? What would a procedure for discovering clusters look like?

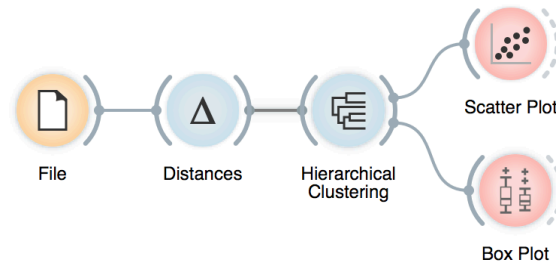
We need to start with a definition of “similar”. One simple measure of similarity for such data is the Euclidean distance: square the differences across every dimension, sum them and take the square root, just like in Pythagorean theorem. So, we would like to group data instances with small Euclidean distances.

Now we need to define a clustering algorithm. We will start with each data instance being in its own cluster. Next, we merge the clusters that are closest together - like the closest two points - into one cluster. Repeat. And repeat. And repeat. And repeat until you end up with a single cluster containing all points.

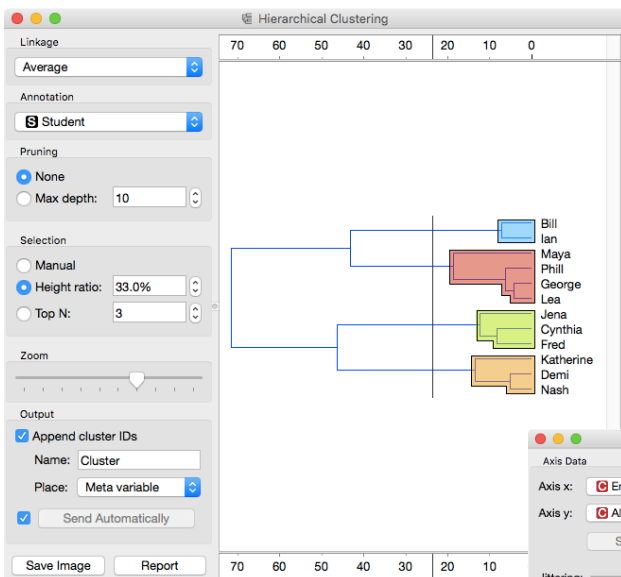
This procedure constructs a hierarchy of clusters, which explains why we call it hierarchical clustering. After it is done, we can

observe the entire hierarchy and decide which would be a good point to stop. With this we decide the actual number of clusters.

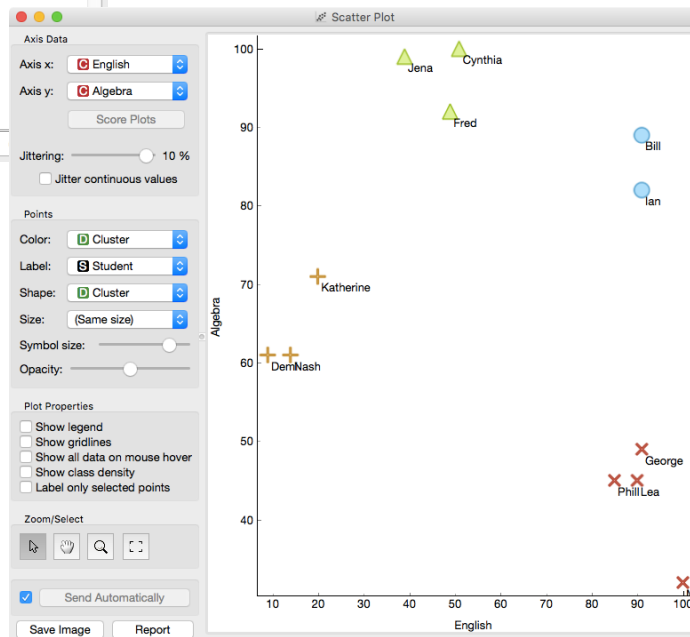
One possible way to observe the results of clustering on our small data set with grades is through the following workflow:



Let us see how this works. Load the data, compute the distances and cluster the data. In the Hierarchical clustering widget, cut hierarchy at a certain distance score and observe the corresponding clusters in the Scatter plot.



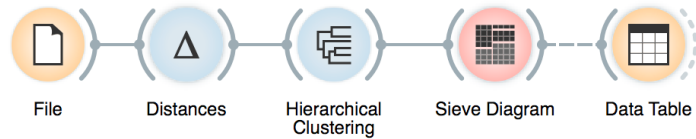
You can also observe the properties of the clusters - that is, the average grades in Algebra and English - in the box plot.



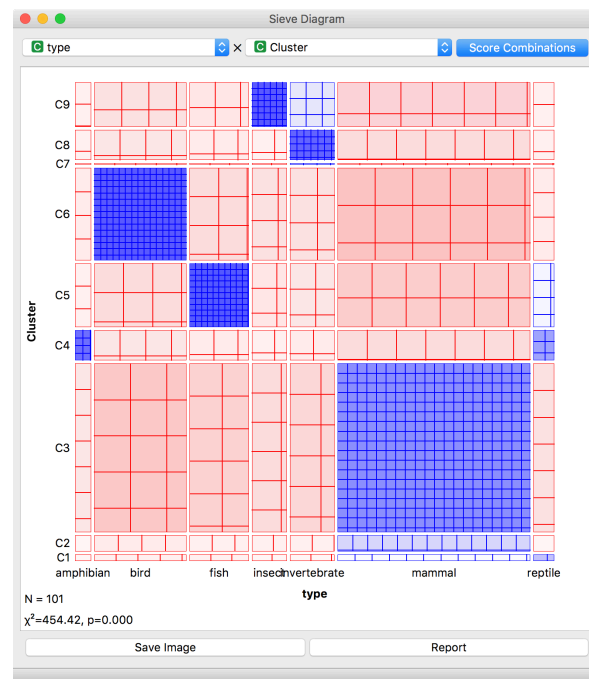
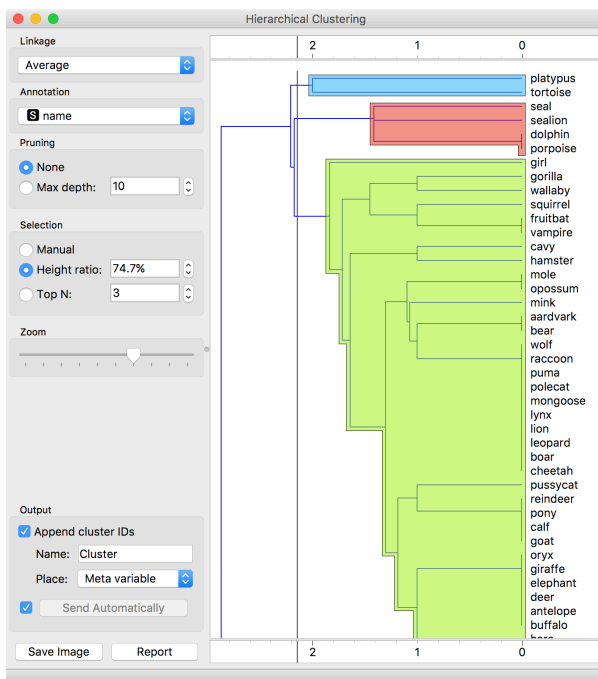
Lesson 6: Animal Kingdom



Your lecturers spent substantial part of their youth admiring a particular Croatian chocolate called Animal Kingdom. Each chocolate bar came with a card — a drawing of some (random) animal, and the associated album made us eat a lot of chocolate. Then our kids came, and the story repeated. Some things stay forever. Funny stuff was we never understood the order in which the cards were laid out in the album. We later learned about taxonomy, but being more inclined to engineering we never mastered learning it in our biology classes. Luckily, there's data mining and the idea that taxonomy simply stems from measuring the distance between species.

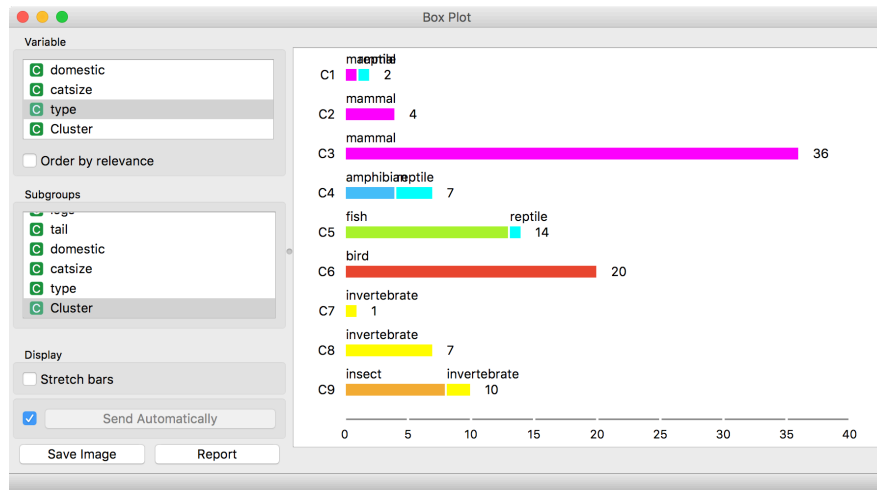


Here we use zoo data (from documentation data sets) with attributes that report on various features of animals (has hair, has feathers, lays eggs). We measure the distance and compute the clustering. Animals in this data set are annotated with type (mammal, insect, bird, and so on). It would be cool to know if the clustering re-discovered these groups of animals. We can do this through marking the clusters in Hierarchical Clustering widget, and then observing the results in the Sieve Diagram.

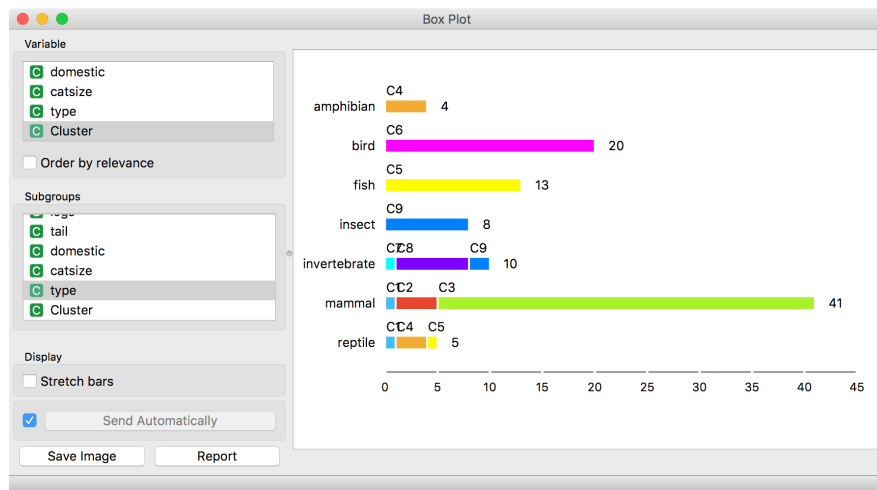


Looks great. Birds, say, are in cluster C6. Cluster C4 consists of amphibians and some reptiles. And so forth.

Checking this in the Box plot is even cooler. We can get a distribution of animal types in each cluster:



Or we can turn it around and see how different types of animals are spread across clusters.



What is wrong with those mammals? Why can't they be in one single cluster? Two reasons. First, they represent 40 % of the data instances. Second, they include some weirdos. Click on the clusters in the box plot and discover who they are.

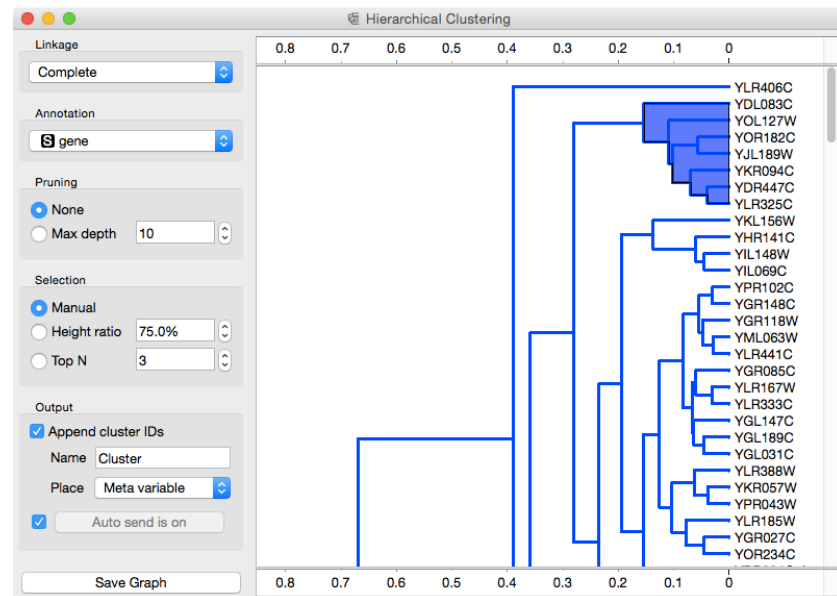
Lesson 7: Discovering clusters

Can we replicate this on some real data? Can clustering indeed be useful for defining meaningful subgroups?

Take brown-selected (from documentation data sets) connect the hierarchical clustering so the you can see a cluster as a subset in the scatterplot.



So far, we used the dendrogram to set a cut-off point. Now we will click on a branch in a dendrogram to select a subset of the data instances. By combining it with the Scatter Plot widget, we get a great tool for exploring the clusters. Try it with an appropriate pair of features to visualize (use Rank projections).



By using a scatter plot or other widgets, an expert can determine whether the clusters are meaningful.

For this data set, though, we can do something even better. The data already contains some predefined groups. Let us check how

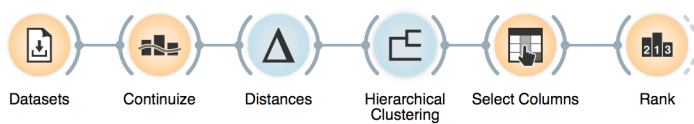
well the clusters match the classes - which we know, but clustering did not.

We will use the dendrogram to set a suitable threshold that splits the data into some three to five clusters. We can plot this data in a new scatter plot; we find a reasonable pair of attributes and then set the color of the points to represent the cluster they belong to. Do the clusters match the actual classes? The result is rather impressive if you keep two things in mind. First, the clustering algorithm did not actually know about the classes, it discovered them by itself. Second, it did not operate on the picture you see in the scatter plot and in which the clusters are quite pronounced, but in a 79-dimensional data space with possibly plenty of redundant features. Yet it identified the three groups of genes almost without mistakes.

This lesson is not a recipe for what you should be doing in practice. If your data already contains group labels, say gene group annotations, there is no need to discover them (again) by using clustering. In this case you should be interested in predictive models from previous lessons. If you do not have such a grouping but you suspect that the data contains distinct subgroups, run clustering. The sole purpose of this lesson was to demonstrate that clustering can indeed find meaningful subgroups in the data; we pretend we did not know the groups, use the clustering to discover them, and checked how well they correspond to the actual groups.

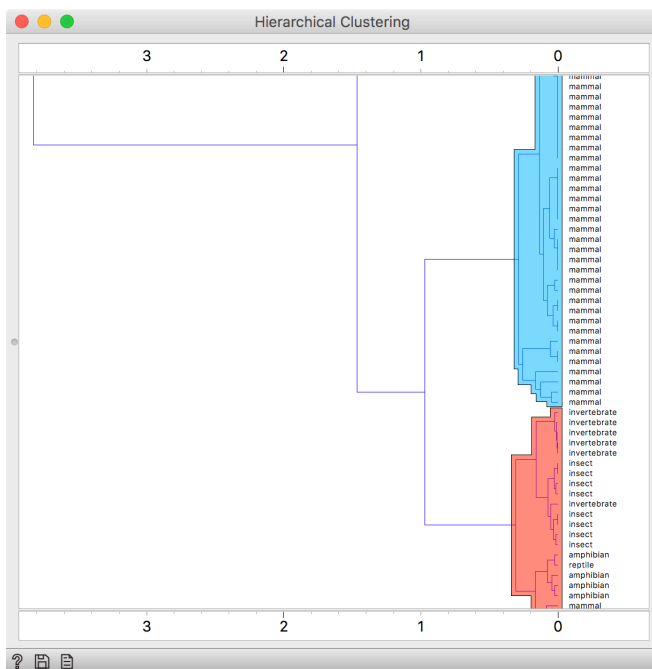
Lesson 8: Cluster interpretation

Once we have inferred the clusters, we would like to know what are the distinguishing features. For the zoo data set, we could, for instance, mark two clusters, and then ask for the features that distinguish among these. Having data marked with cluster identifiers takes us back to classification, and we can use any of visualization, model inference, or feature ranking techniques we have introduced there. Here, we will show how to use ranking to infer what features characterize the group of mammals when compared to a close cluster of other species.

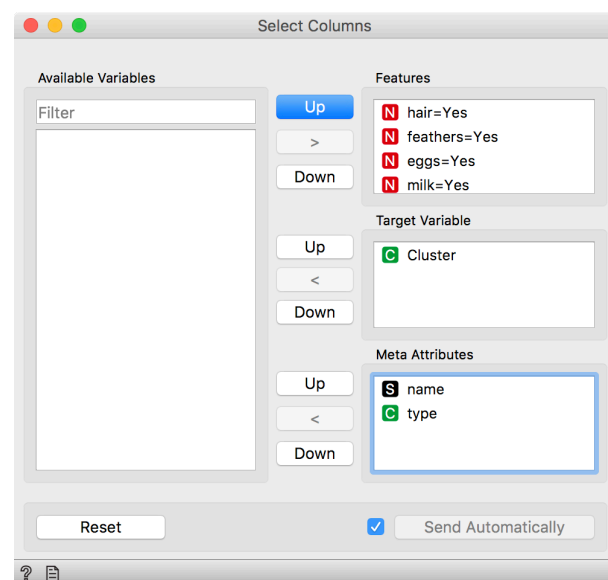


We load zoo data from Datasets, continuize the categorical features (this will be changed in Orange soon, Distances should automatically perform continuization), estimate the distances, and feed everything in Hierarchical Clustering. So far, nothing new.

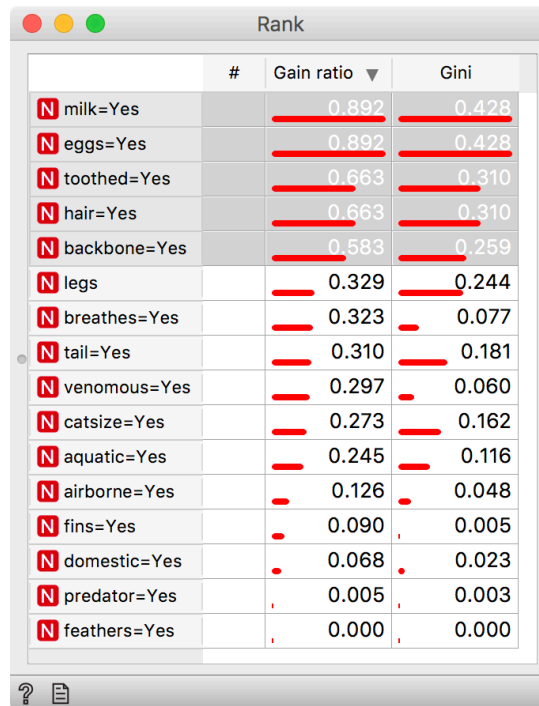
In the hierarchical clustering, we choose two clusters. Note that Hierarchical Clustering adds cluster identifier as a meta feature; to make the data ready for classification-specific tasks, we need to promote cluster identifier into target variable (a class) by reassigning the feature types in Select Columns.



Use modifier keys (command) to select different branches of the dendrogram and mark them as separate clusters. Done correctly, Hierarchical Clustering will mark the branches with different colors.



The data is now ready for classification-based analysis. Here, we used a rank widget.



The screenshot shows a window titled "Rank" with a table of features. The table has four columns: an unlabeled column for feature names, a column for the number of instances (#), a column for Gain ratio (with a downward arrow), and a column for Gini index. The features are sorted in descending order of Gain ratio. Each feature name is preceded by a red square containing a white letter 'N'. Red horizontal bars are drawn under the Gain ratio and Gini index values for each row. At the bottom left of the window, there are icons for help (a question mark) and a document.

	#	Gain ratio ▼	Gini
N milk=Yes		0.892	0.428
N eggs=Yes		0.892	0.428
N toothed=Yes		0.663	0.310
N hair=Yes		0.663	0.310
N backbone=Yes		0.583	0.259
N legs		0.329	0.244
N breathes=Yes		0.323	0.077
N tail=Yes		0.310	0.181
N venomous=Yes		0.297	0.060
N catsize=Yes		0.273	0.162
N aquatic=Yes		0.245	0.116
N airborne=Yes		0.126	0.048
N fins=Yes		0.090	0.005
N domestic=Yes		0.068	0.023
N predator=Yes		0.005	0.003
N feathers=Yes		0.000	0.000