

# MDS

Multidimensional scaling (MDS) projects items onto a plane fitted to given distances between points.

## Inputs

- Data: input dataset
- Distances: distance matrix
- Data Subset: subset of instances

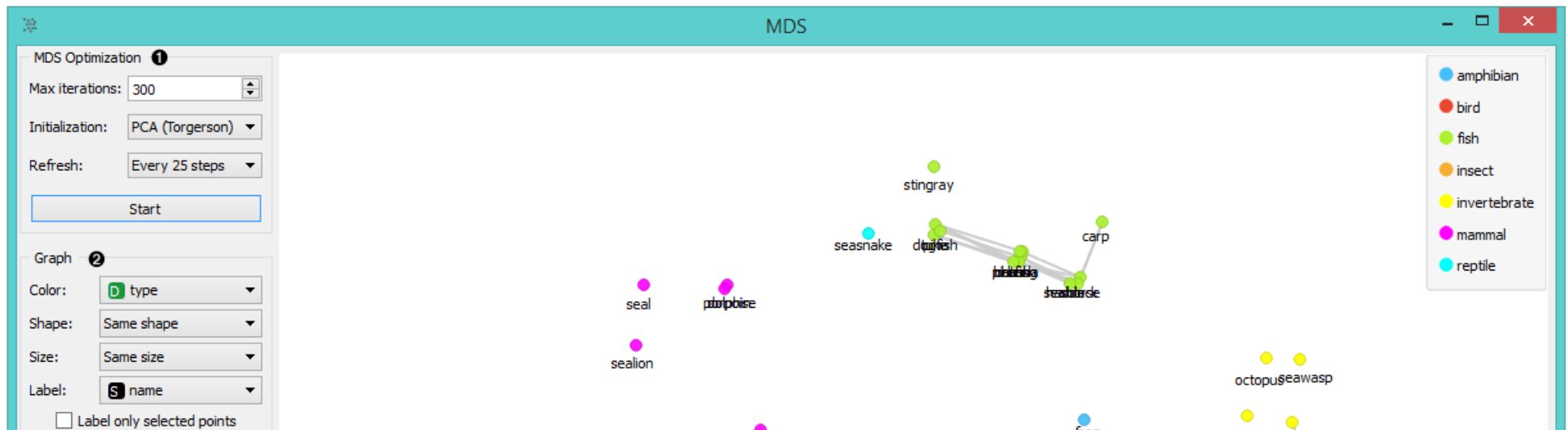
## Outputs

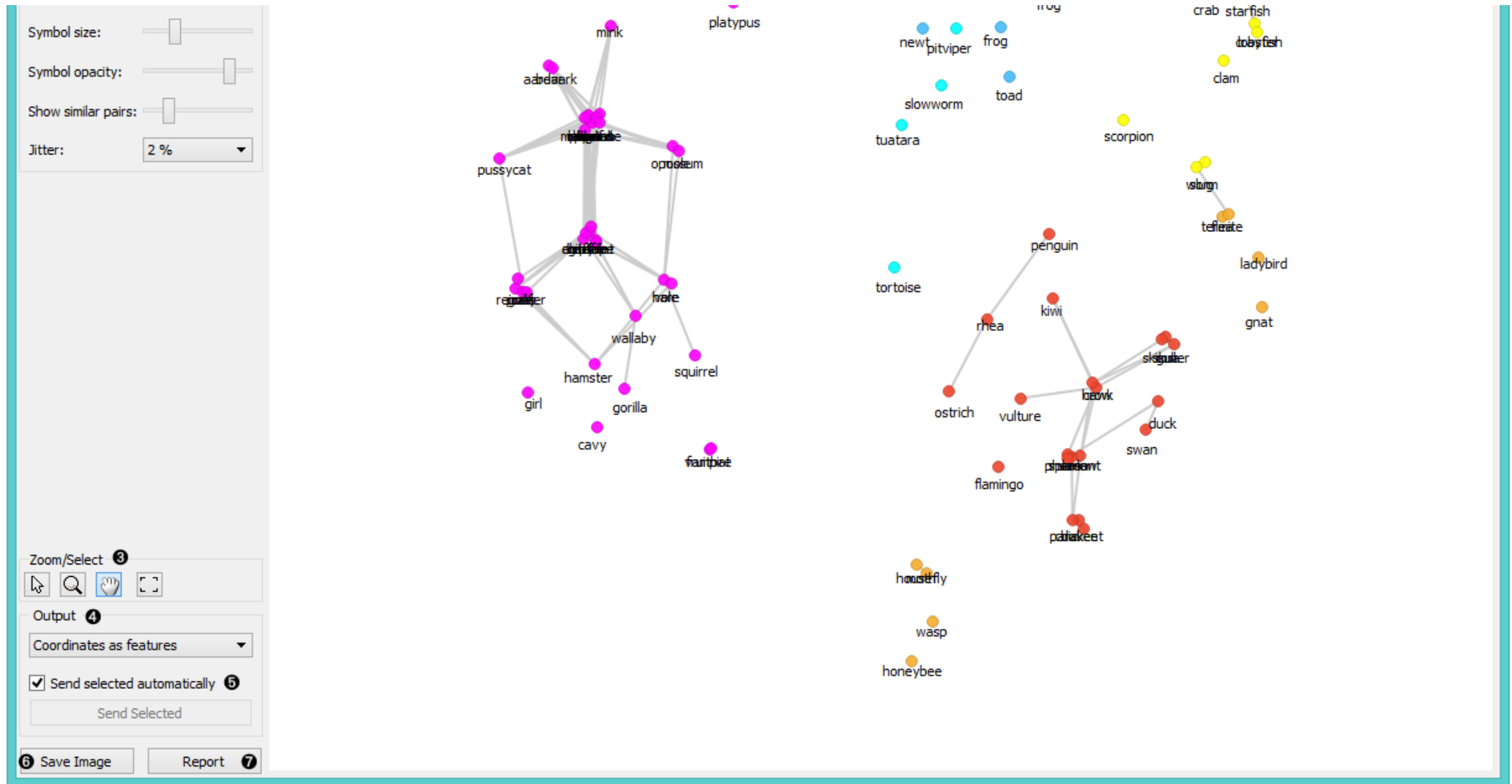
- Selected Data: instances selected from the plot
- Data: dataset with MDS coordinates

**Multidimensional scaling** is a technique which finds a low-dimensional (in our case a two-dimensional) projection of points, where it tries to fit distances between points as well as possible. The perfect fit is typically impossible to obtain since the data is high-dimensional or the distances are not **Euclidean**.

In the input, the widget needs either a dataset or a matrix of distances. When visualizing distances between rows, you can also adjust the color of the points, change their shape, mark them, and output them upon selection.

The algorithm iteratively moves the points around in a kind of a simulation of a physical model: if two points are too close to each other (or too far away), there is a force pushing them apart (or together). The change of the point's position at each time interval corresponds to the sum of forces acting on it.





1. The widget redraws the projection during optimization. Optimization is run automatically in the beginning and later by pushing *Start*.
  - **Max iterations:** The optimization stops either when the projection changes only minimally at the last iteration or when a maximum number of iterations has been reached.
  - **Initialization:** PCA (Torgerson) positions the initial points along principal coordinate axes. *Random* sets the initial points to a random position and then readjusts them.
  - **Refresh:** Set how often you want to refresh the visualization. It can be at *Every iteration*, *Every 5/10/25/50 steps* or never (*None*). Setting a lower refresh interval makes the animation more visually appealing, but can be slow if the number of points is high.
2. Defines how the points are visualized. These options are available only when visualizing distances between rows (selected in the **Distances** widget).

- **Color:** Color of points by attribute (gray for continuous, colored for discrete).
  - **Shape:** Shape of points by attribute (only for discrete).
  - **Size:** Set the size of points (*Same size* or select an attribute) or let the size depend on the value of the continuous attribute the point represents (Stress).
  - **Label:** Discrete attributes can serve as a label.
  - **Symbol size:** Adjust the size of the dots.
  - **Symbol opacity:** Adjust the transparency level of the dots.
  - **Show similar pairs:** Adjust the strength of network lines.
  - **Jitter:** Set **jittering** to prevent the dots from overlapping.
3. Adjust the graph with *Zoom/Select*. The arrow enables you to select data instances. The magnifying glass enables zooming, which can be also done by scrolling in and out. The hand allows you to move the graph around. The rectangle readjusts the graph proportionally.
  4. Select the desired output:
    - **Original features only** (input dataset)
    - **Coordinates only** (MDS coordinates)
    - **Coordinates as features** (input dataset + MDS coordinates as regular attributes)
    - **Coordinates as meta attributes** (input dataset + MDS coordinates as meta attributes)
  5. Sending the instances can be automatic if *Send selected automatically* is ticked. Alternatively, click *Send selected*.
  6. **Save Image** allows you to save the created image either as .svg or .png file to your device.
  7. Produce a report.

The MDS graph performs many of the functions of the Visualizations widget. It is in many respects similar to the `Scatter Plot <../visualize/scatterplot>` widget, so we recommend reading that widget's description as well.

## Example

The above graphs were drawn using the following simple schema. We used the *iris.tab* dataset. Using the `Distances <../unsupervised/distances>` widget we input the distance matrix into the **MDS** widget, where we see the *Iris* data displayed in a 2-dimensional plane. We can see the appended coordinates in the `Data Table <../data/datatable>` widget.

The screenshot shows the Orange Data Mining software interface. On the left, a workflow is visible with three widgets: 'File', 'MDS', and 'Data Table', connected in sequence. The 'MDS' widget is selected, and its 'Info' panel is open, showing details about the dataset: 76 instances (no missing values), 6 features (no missing values), Discrete class with 3 values (no missing values), and No meta attributes. The 'Variables' panel for the MDS widget shows options for 'Show variable labels (if present)', 'Visualize continuous values', and 'Color by instance classes', all of which are checked.

On the right, the 'Data Table' widget is open, displaying a table with 7 columns: 'iris', 'sepal length', 'sepal width', 'petal length', 'petal width', 'X', and 'Y'. The table contains 7 rows of data, showing the MDS coordinates (X and Y) for each instance of the Iris dataset.

|    | iris            | sepal length | sepal width | petal length | petal width | X     | Y      |
|----|-----------------|--------------|-------------|--------------|-------------|-------|--------|
| 40 | Iris-versicolor | 5.800        | 2.600       | 4.000        | 1.200       | 0.202 | -0.398 |
| 41 | Iris-versicolor | 5.600        | 2.700       | 4.200        | 1.300       | 0.342 | -0.478 |
| 42 | Iris-versicolor | 5.700        | 3.000       | 4.200        | 1.200       | 0.318 | -0.177 |
| 43 | Iris-versicolor | 5.700        | 2.900       | 4.200        | 1.300       | 0.367 | -0.268 |
| 44 | Iris-versicolor | 6.200        | 2.900       | 4.300        | 1.300       | 0.621 | 0.073  |
| 45 | Iris-versicolor | 5.700        | 2.800       | 4.100        | 1.300       | 0.288 | -0.320 |
| 46 | Iris-virginica  | 5.800        | 2.700       | 5.100        | 1.900       | 1.432 | -0.630 |



## References

Wickelmaier, F. (2003). An Introduction to MDS. Sound Quality Research Unit, Aalborg University. Available [here](#).