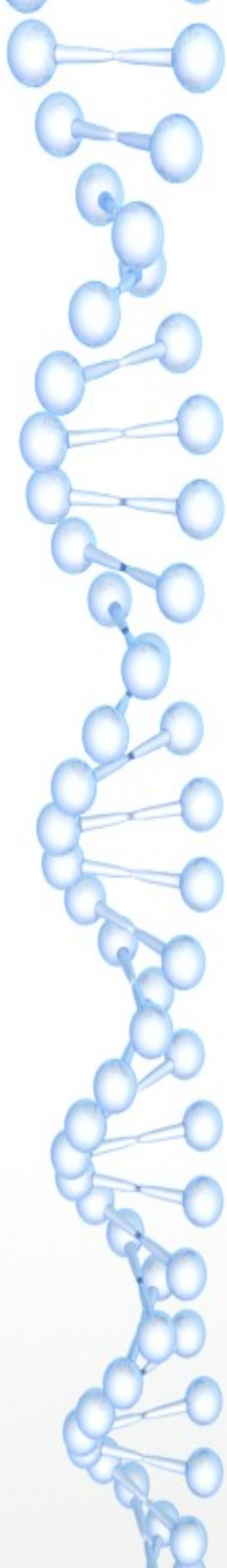


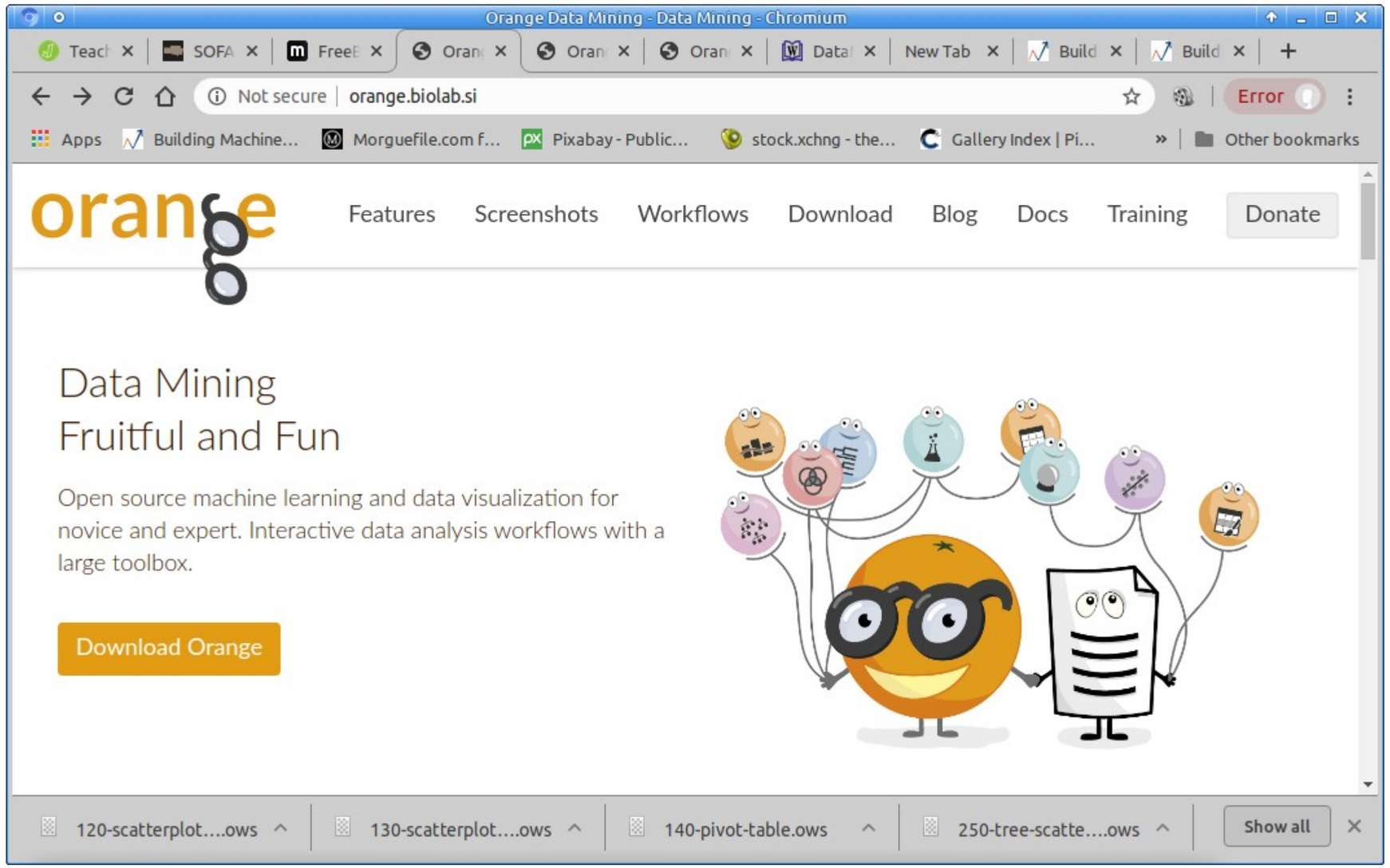
Data Science with Orange3

Practical Session

José R. Valverde, PhD, MD
CNB-CSIC
2024

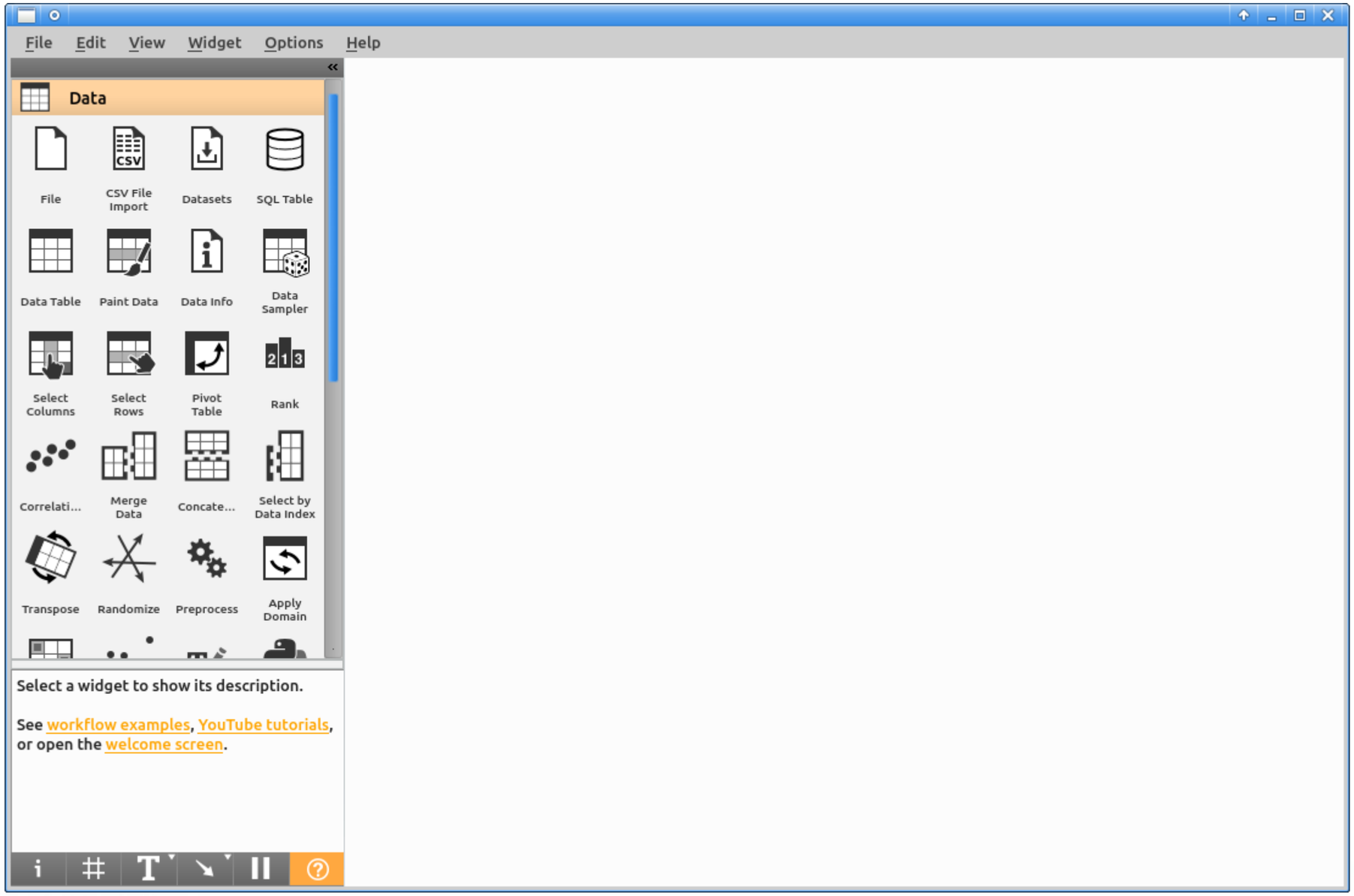
Orange 3

- Statistics tool
- Specially oriented to Data Mining
- Visualization
- Graphical User Interface
- Interactive
- **Workflow**-oriented (requires a new mindset)
- Versatile and **extensible**
- http://orange.biolab.si/


Orange Data Mining - Data Mining - Chromium
Not secure
orange.biolab.si
Error
Apps
Building Machine...
Morguefile.com f...
Pixabay - Public...
stock.xchng - the...
Gallery Index | Pi...
Other bookmarks
orange
Features
Screenshots
Workflows
Download
Blog
Docs
Training
Donate
Data Mining
Fruitful and Fun
Open source machine learning and data visualization for novice and expert. Interactive data analysis workflows with a large toolbox.
Download Orange
120-scatterplot....ows
130-scatterplot....ows
140-pivot-table.ows
250-tree-scatte....ows
Show all

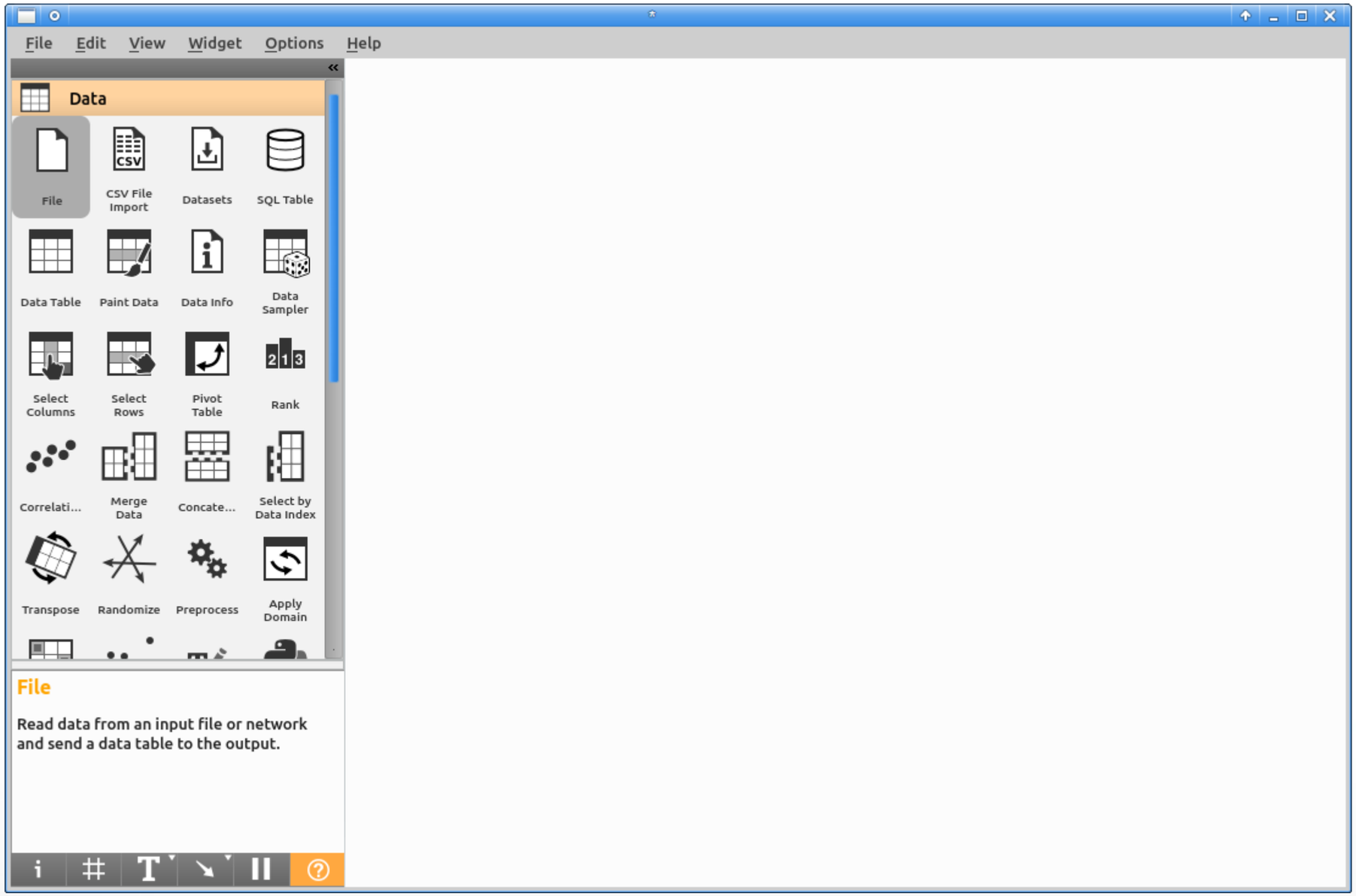
Getting started

Open Orange3 and select **New**

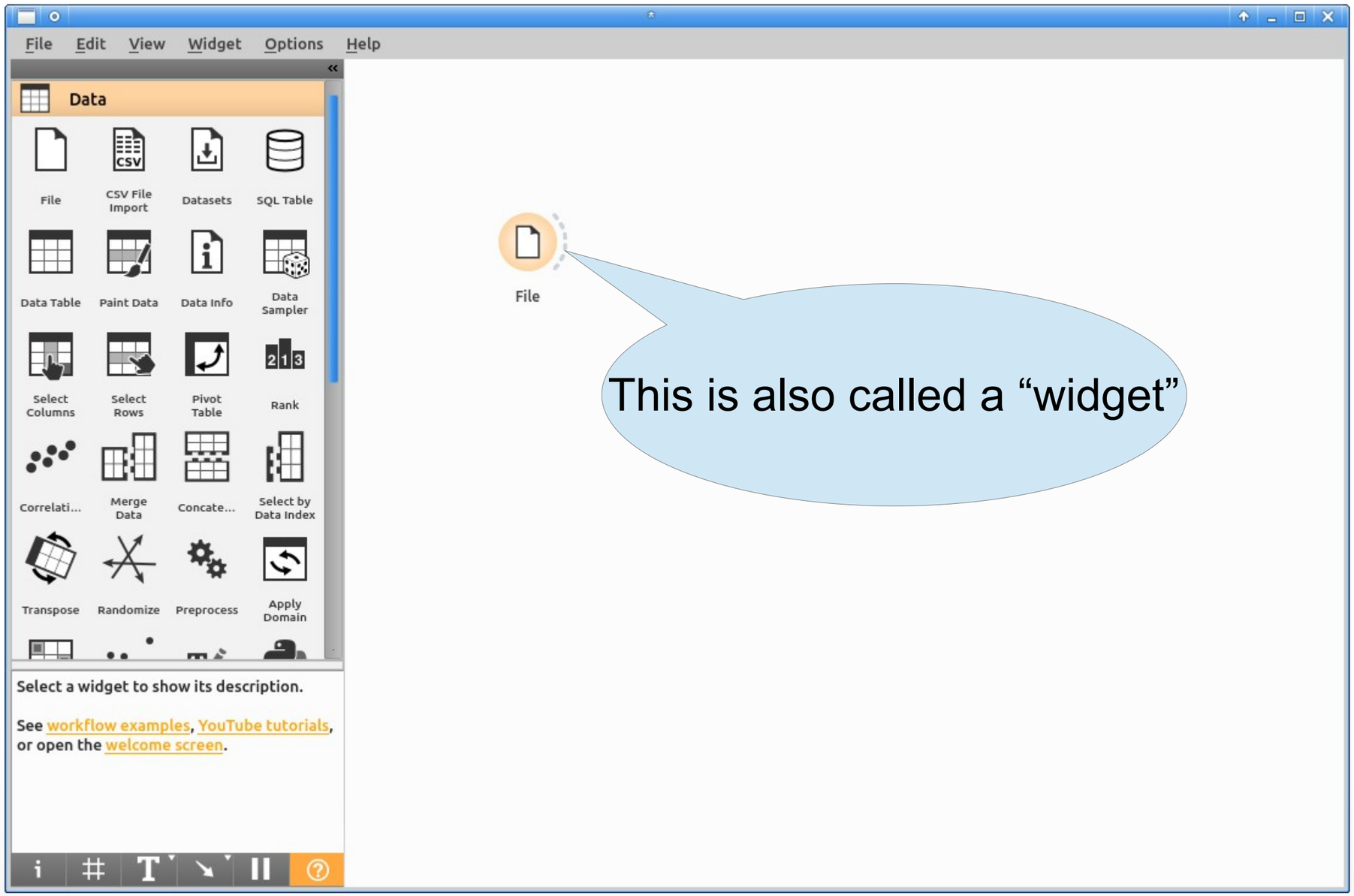


You should get a window like this one.

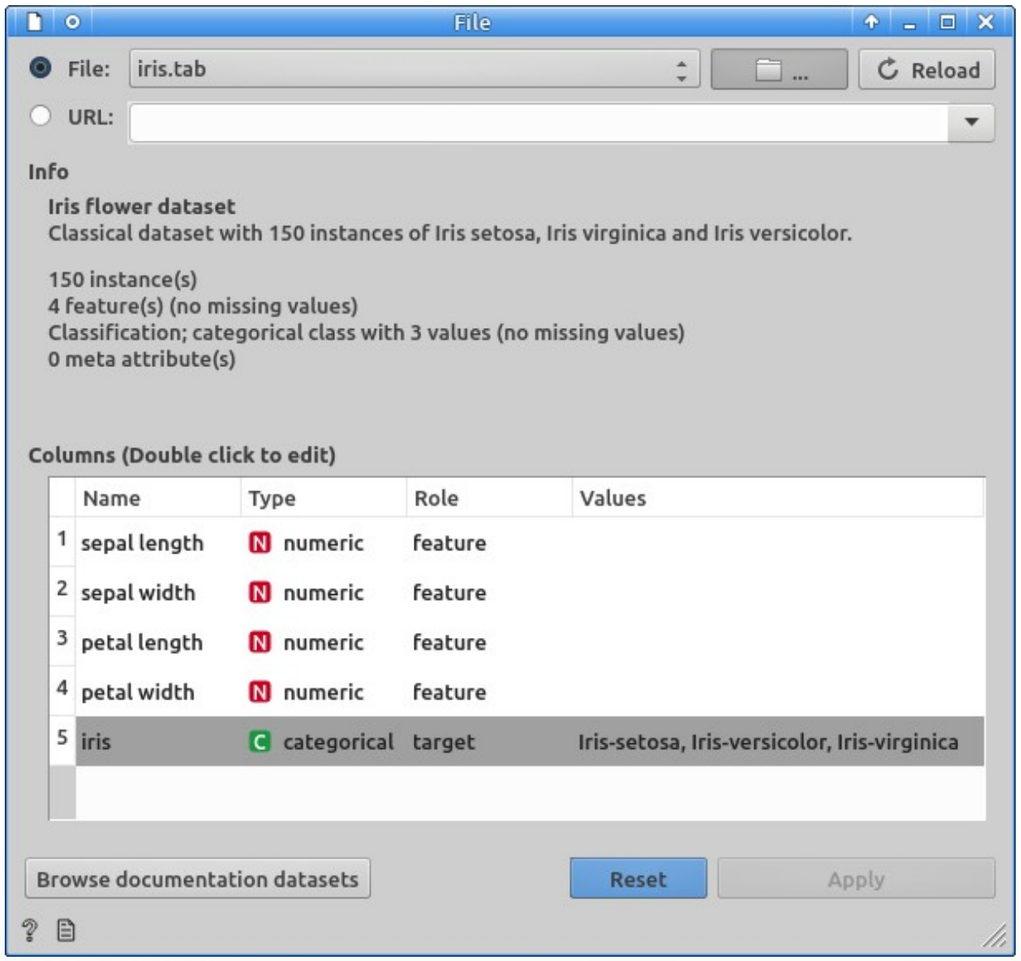
Click on “File” and drag to the canvas



Double click on the new “File” icon

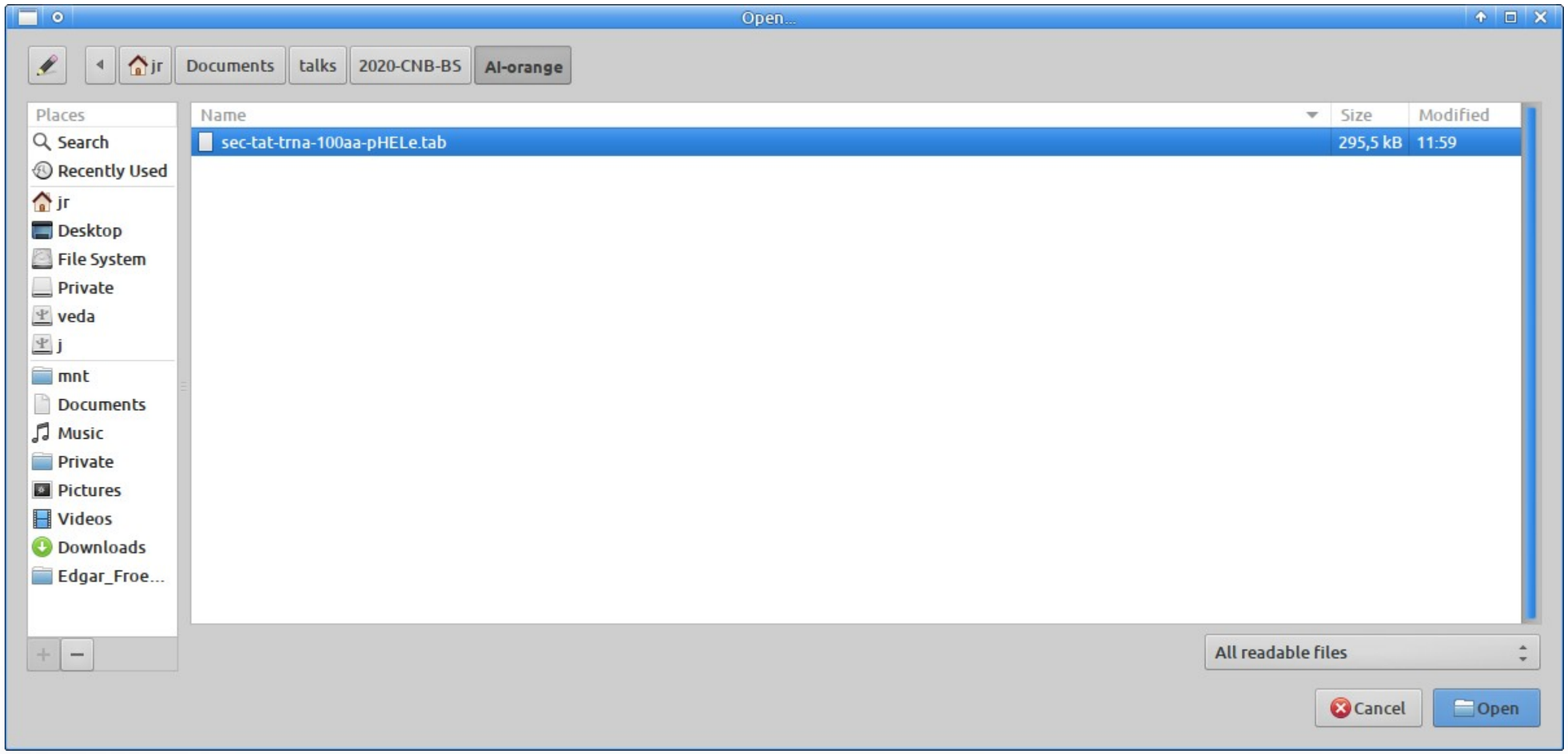


Select file to open



- By default it selects the “iris” data set.
- We want to use our own one
- Click on the folder icon close to the file name/drop down menu

Choose the data file and click “Open”



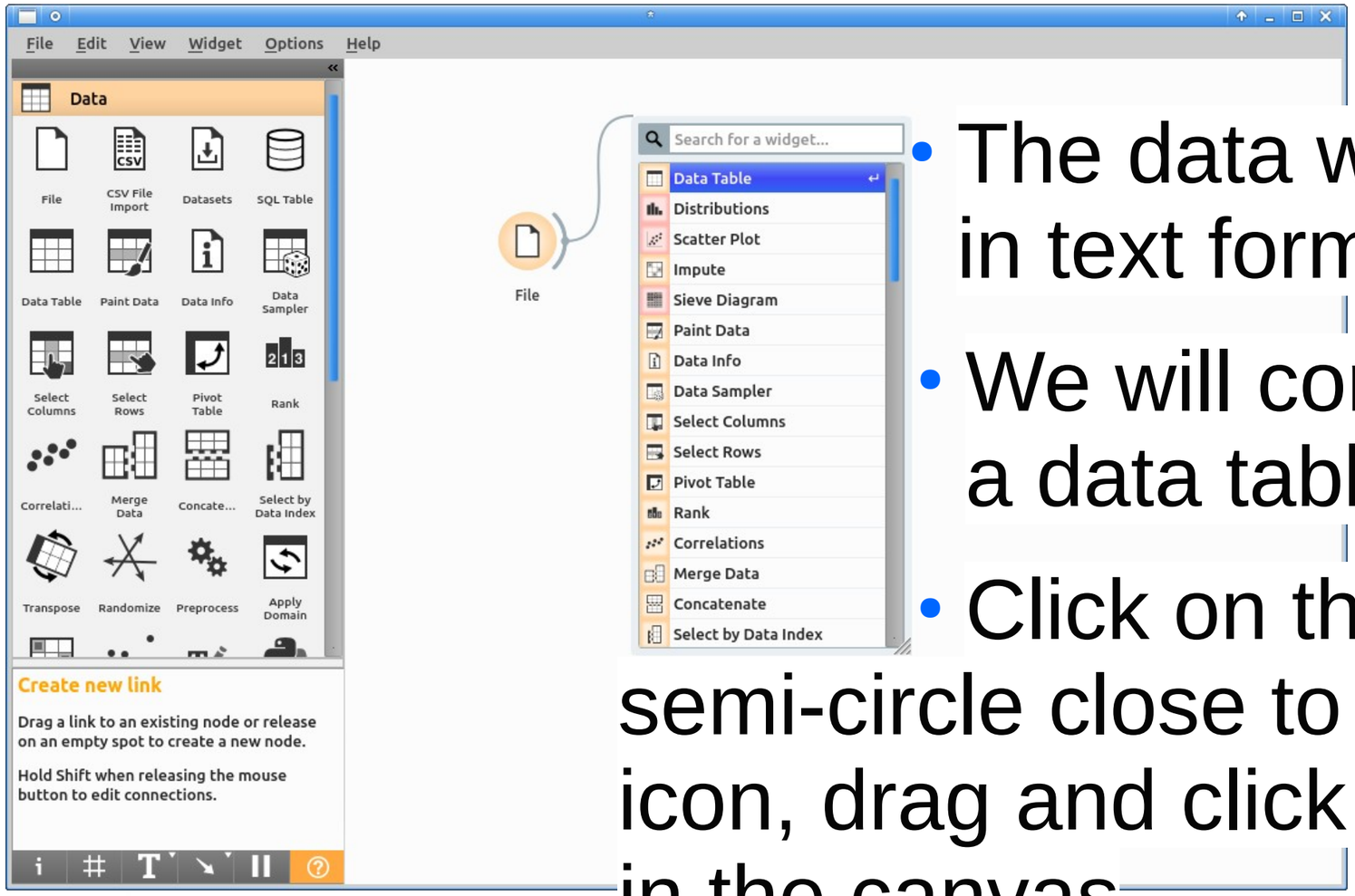
The sec-tat-trna data files

- Our goal is to predict the secretion route for a given *Streptomyces lividans* TK24 protein.
  - *S. lividans* secretes proteins using mainly one of two routes, Sec or Tat.
- We have chosen 30 proteins secreted through Sec, 30 secreted through Tat and 68 proteins involved in tRNA metabolism (which we assume will not be secreted).
- We have arbitrarily selected the first 100 amino acids as those determinant for secretion

Choosing the data to analyze

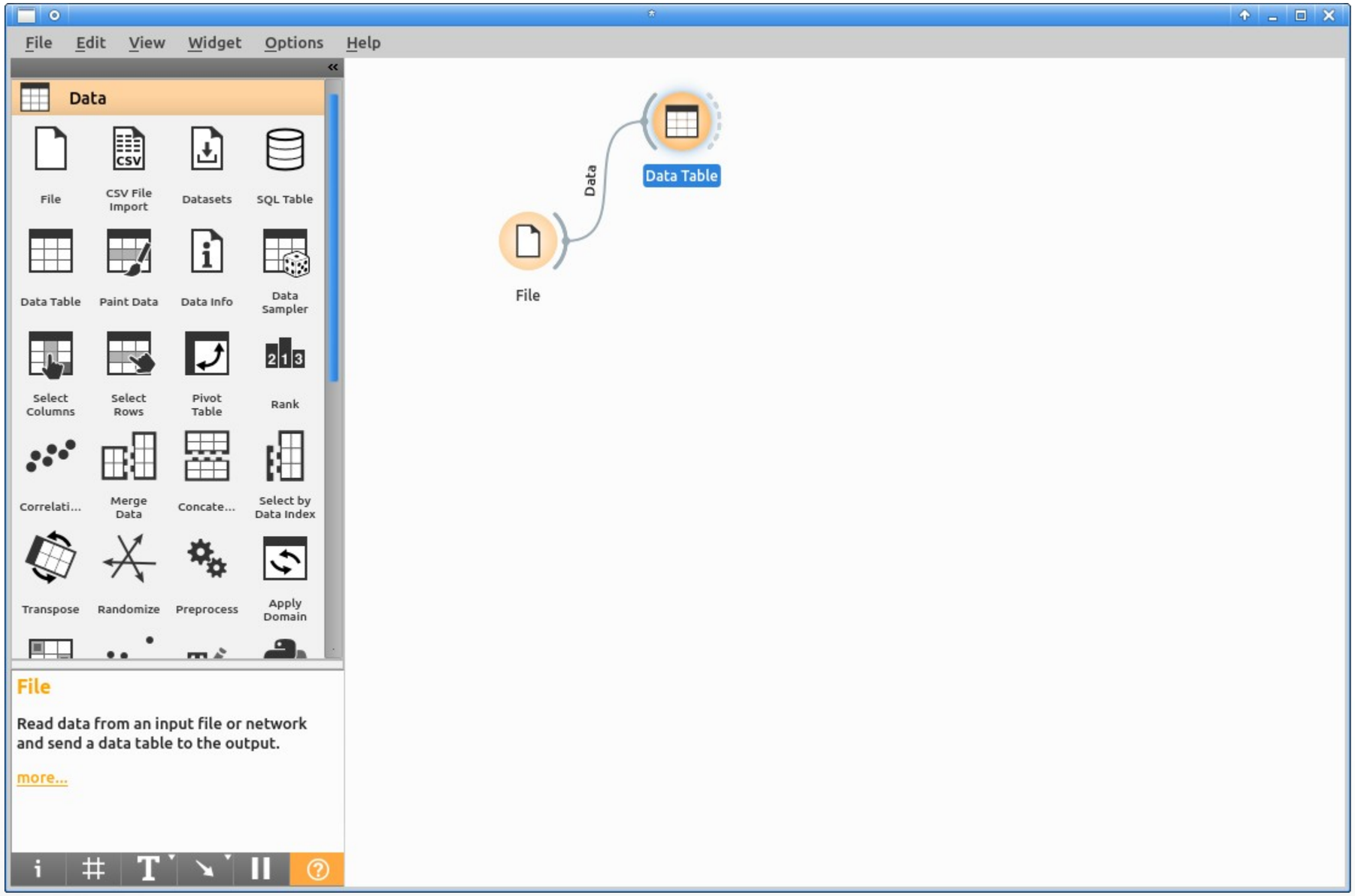
- Previous work based only on the sequence failed to be accurate enough
  - There are sequence-based secretory signals
  - But they are tremendously unspecific in *S. lividans*
    - Multiple sequence alignments show a total lack of differentiating/specific regions
  - We have computed properties for each amino acid in the sequence, e.g.
    - Secondary structure: Helix, Extended, Loop probabilities
    - Solvent accessibility: buried, intermediate, exposed
    - Hidrophobicity: raw, α-helix moment, β-sheet moment...
  - We have labeled SEC proteins as such.

See the data

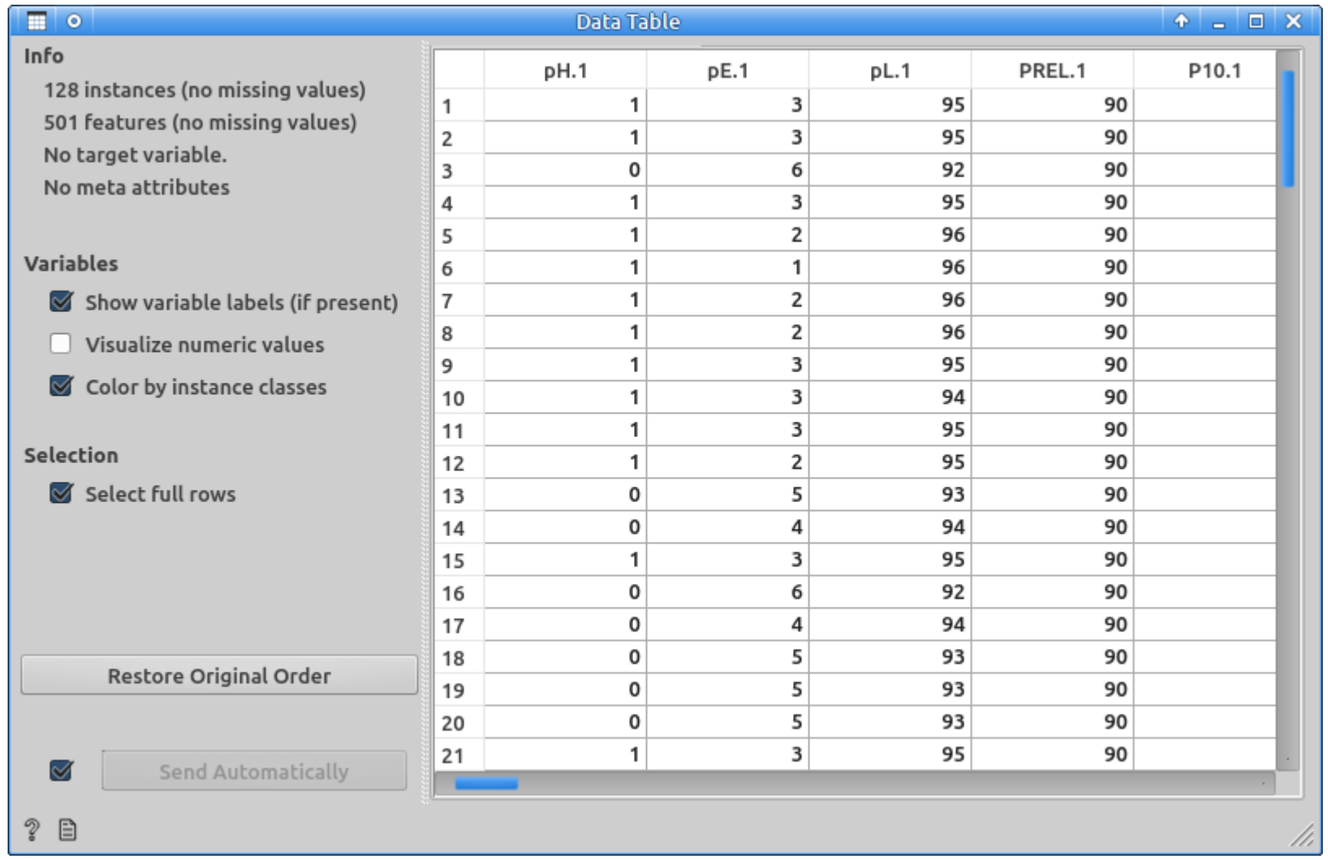


- The data we have is in text format.
- We will convert it to a data table to see it.
- Click on the dotted semi-circle close to the “File” icon, drag and click anywhere in the canvas
- Select “Data Table” from the drop-down menu.

Double click on the *Data Table* widget

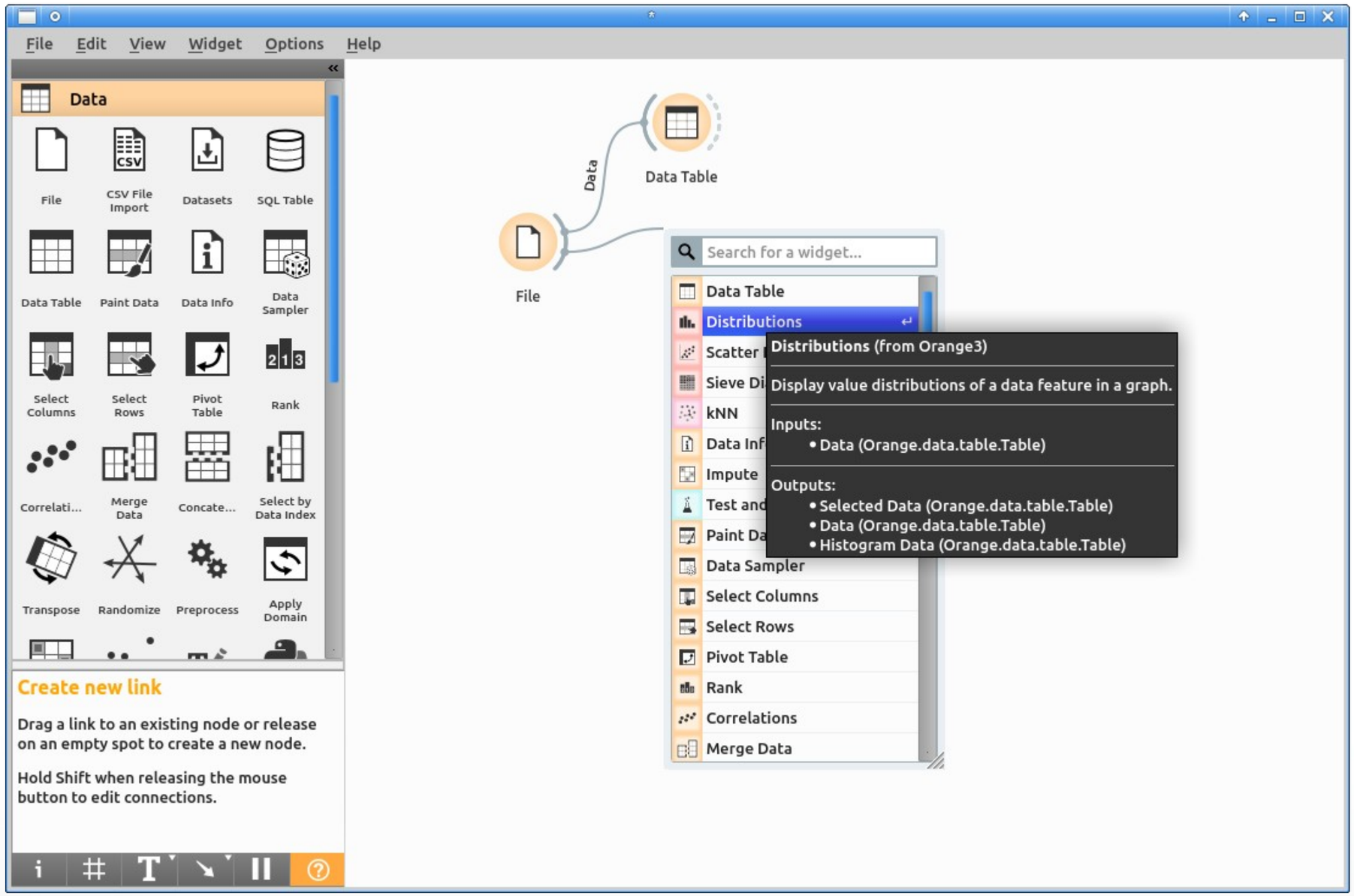


Explore the data

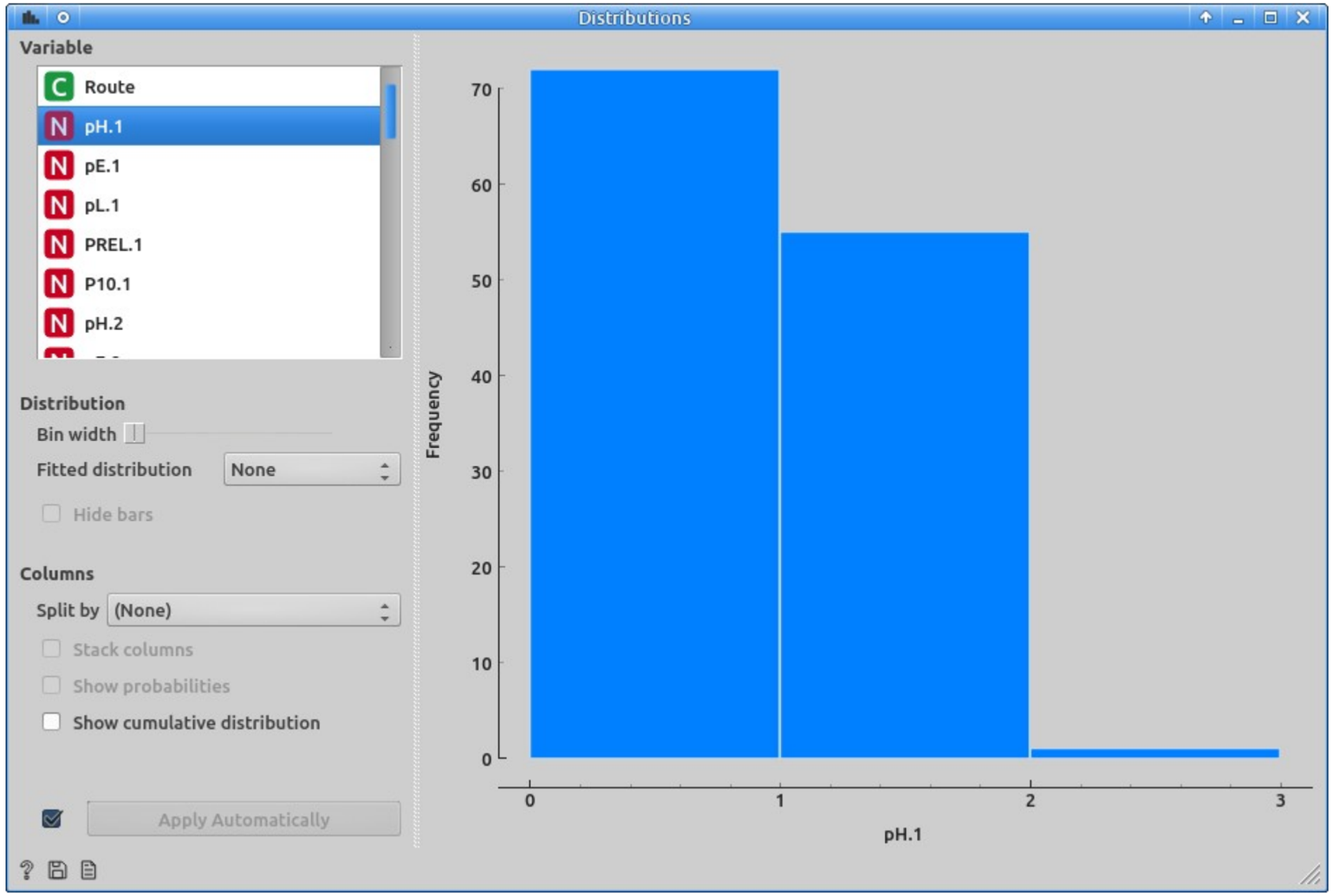


Move around with the scrollbars. On the far right you should see the protein classification. When you are done, close the dialog window.

Visualization

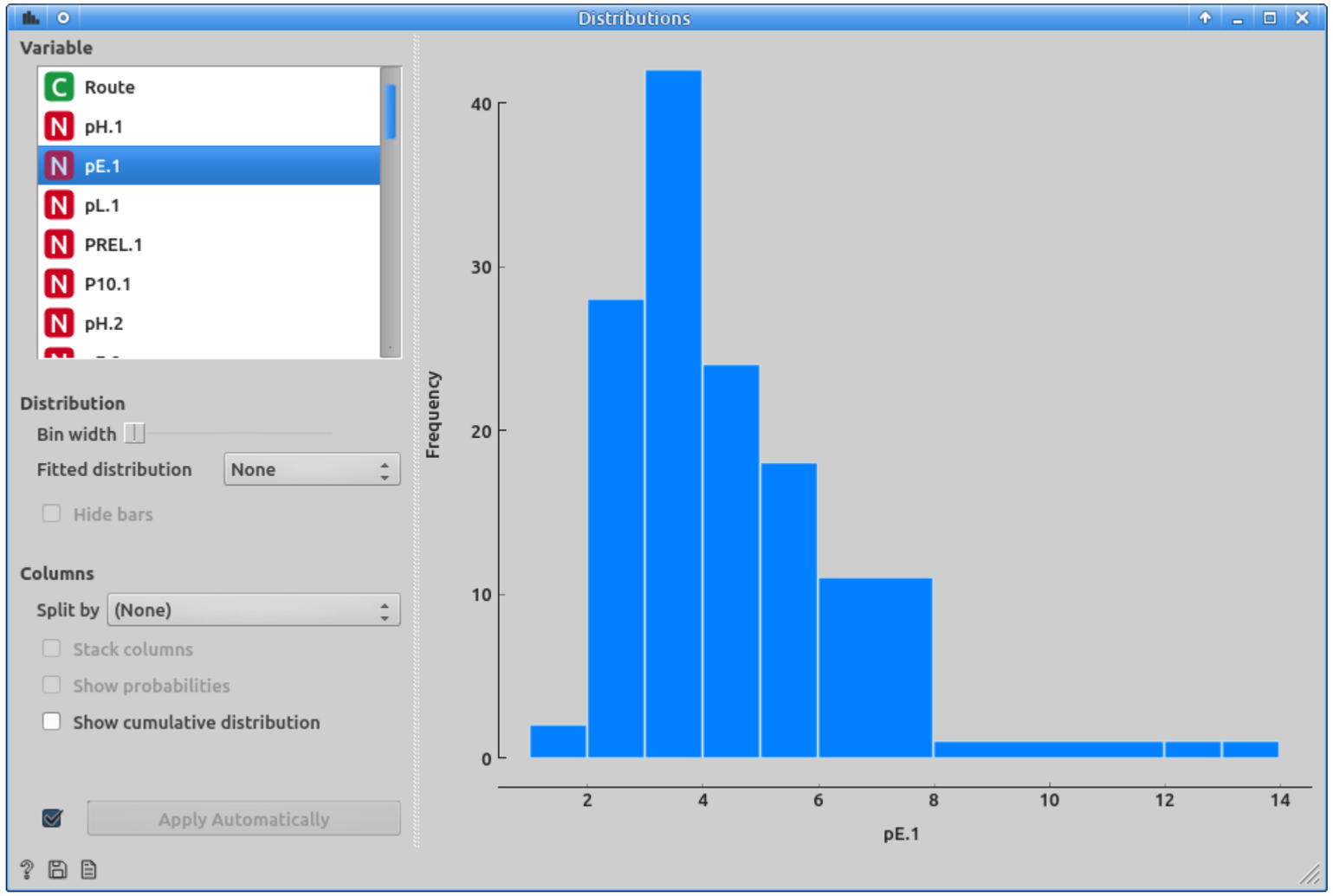


We will now add a “distribution” widget. This will allow us to explore how the data in our variables is distributed

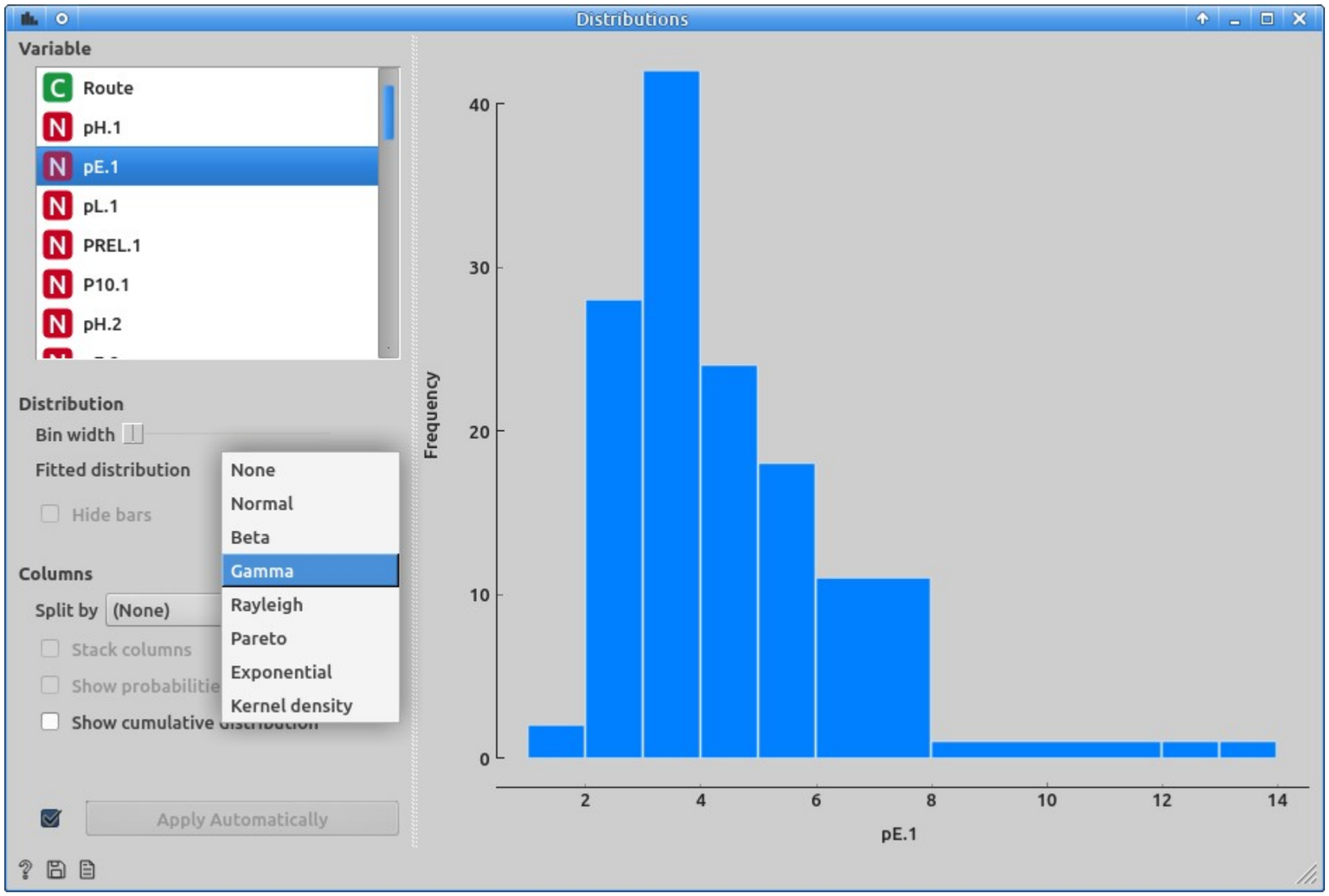


Probabilities were encoded as values from 0 to 10. There doesn't seem to be much variability in the probability that the first amino acid is in an helical state.

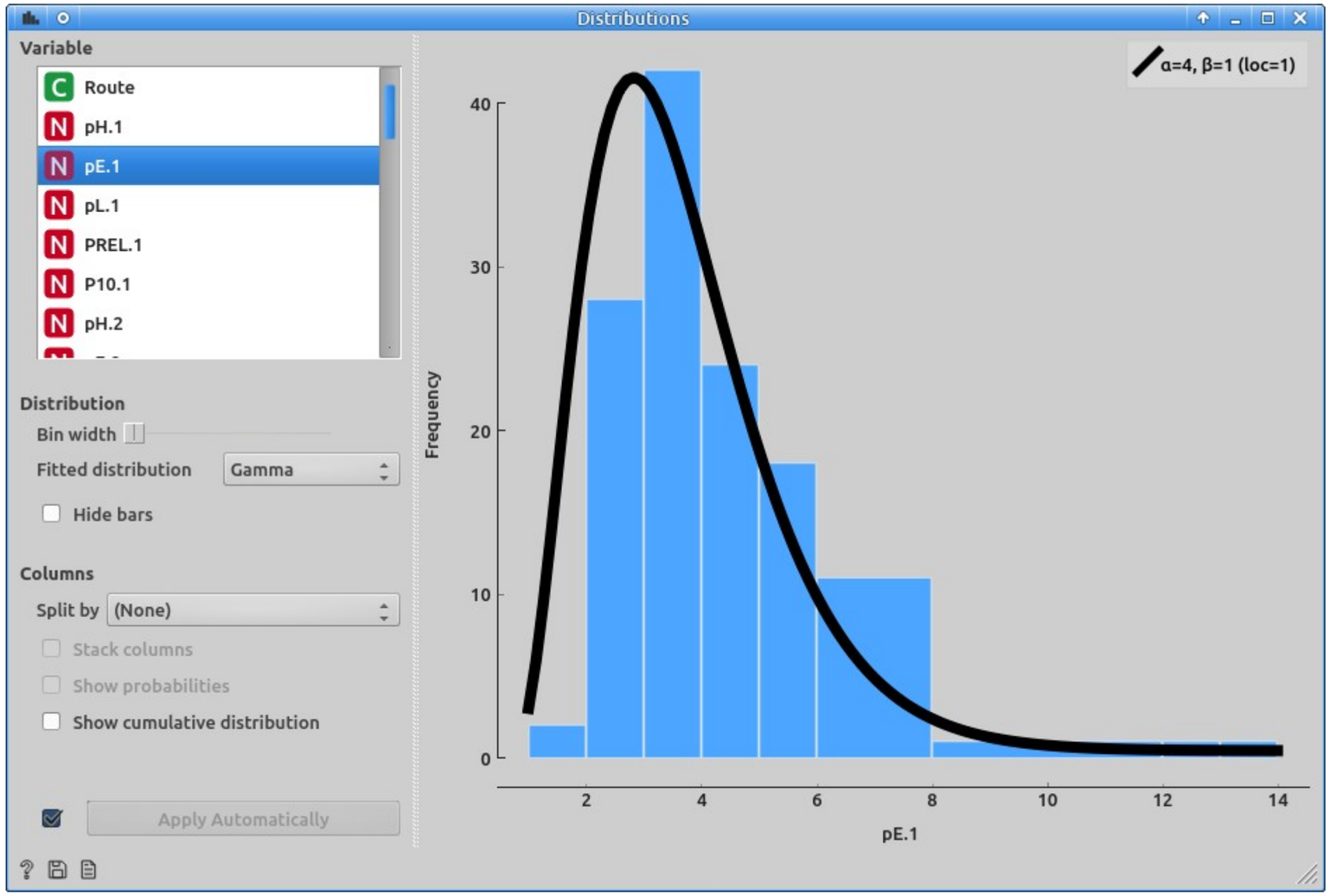
To look at the probability that it is in an extended sheet state, click on pE.1



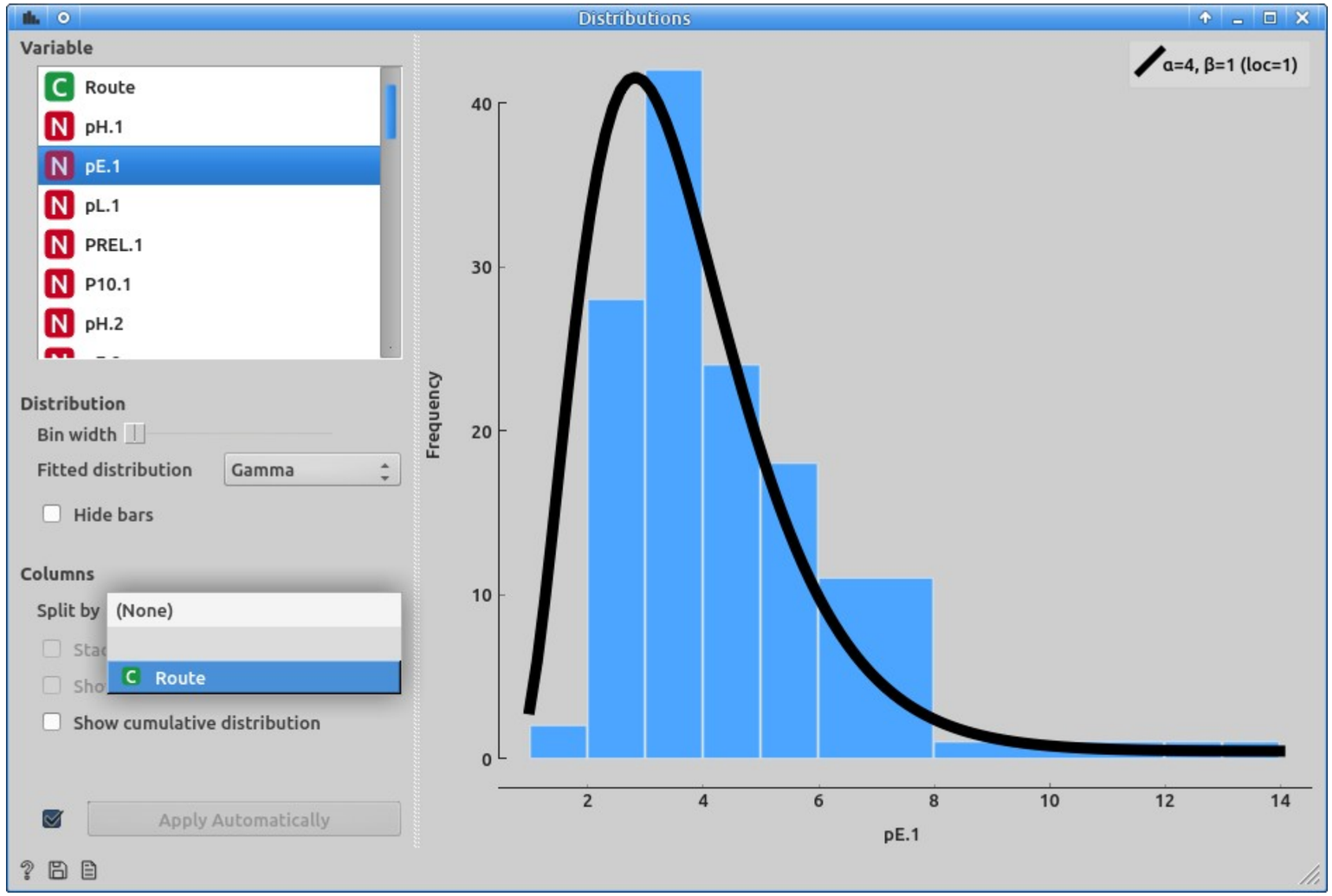
We will not look at the probabilities for each state of each of the 100 chosen amino acids.
But we can use this one to explore the data visualization facilities of Orange 3



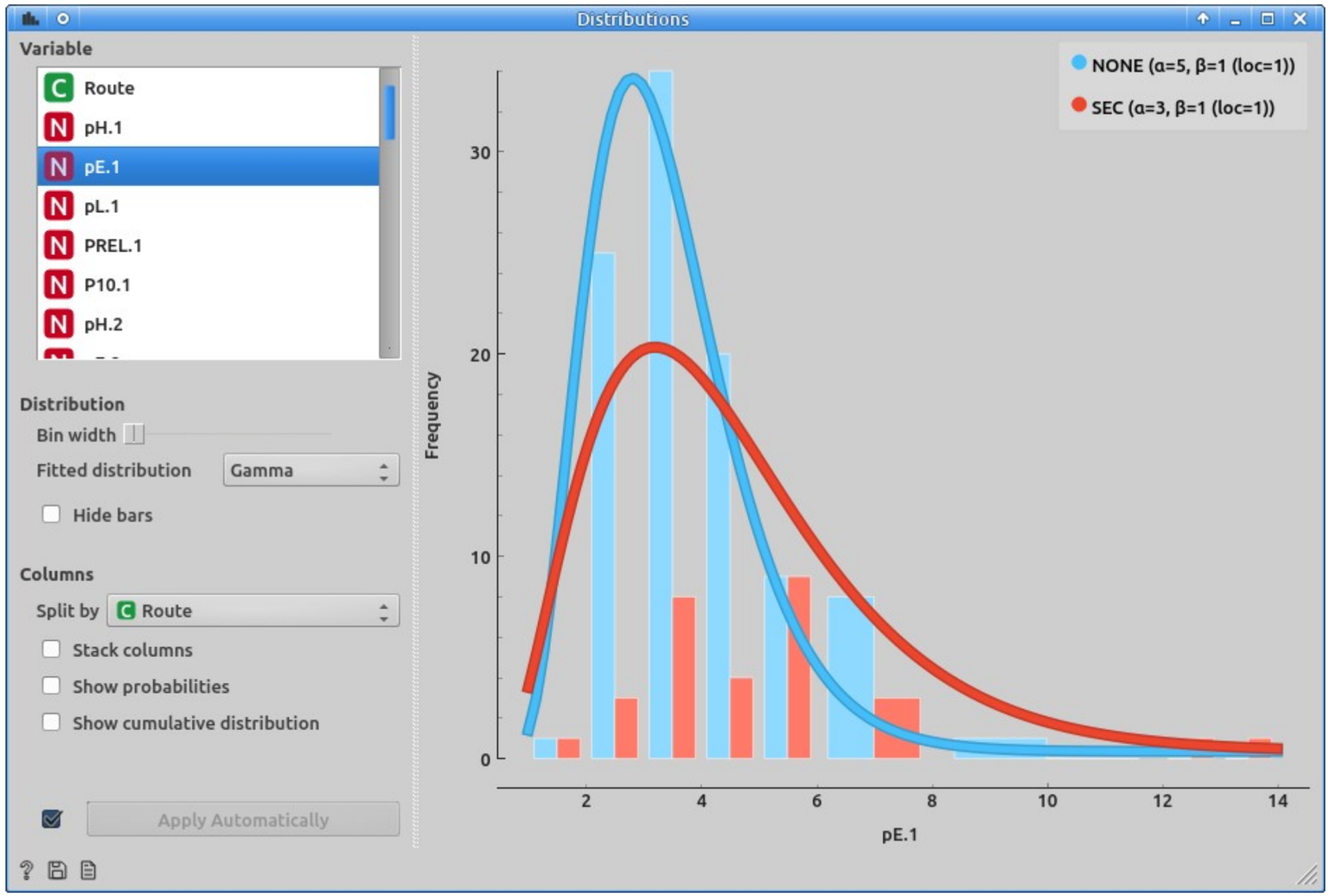
Click on “Distribution”. This will unfold a drop-down menu with a list of various distributions. You can try all of them and see how well our data adjusts to each corresponding distribution.
For instance, let us select “Gamma”.



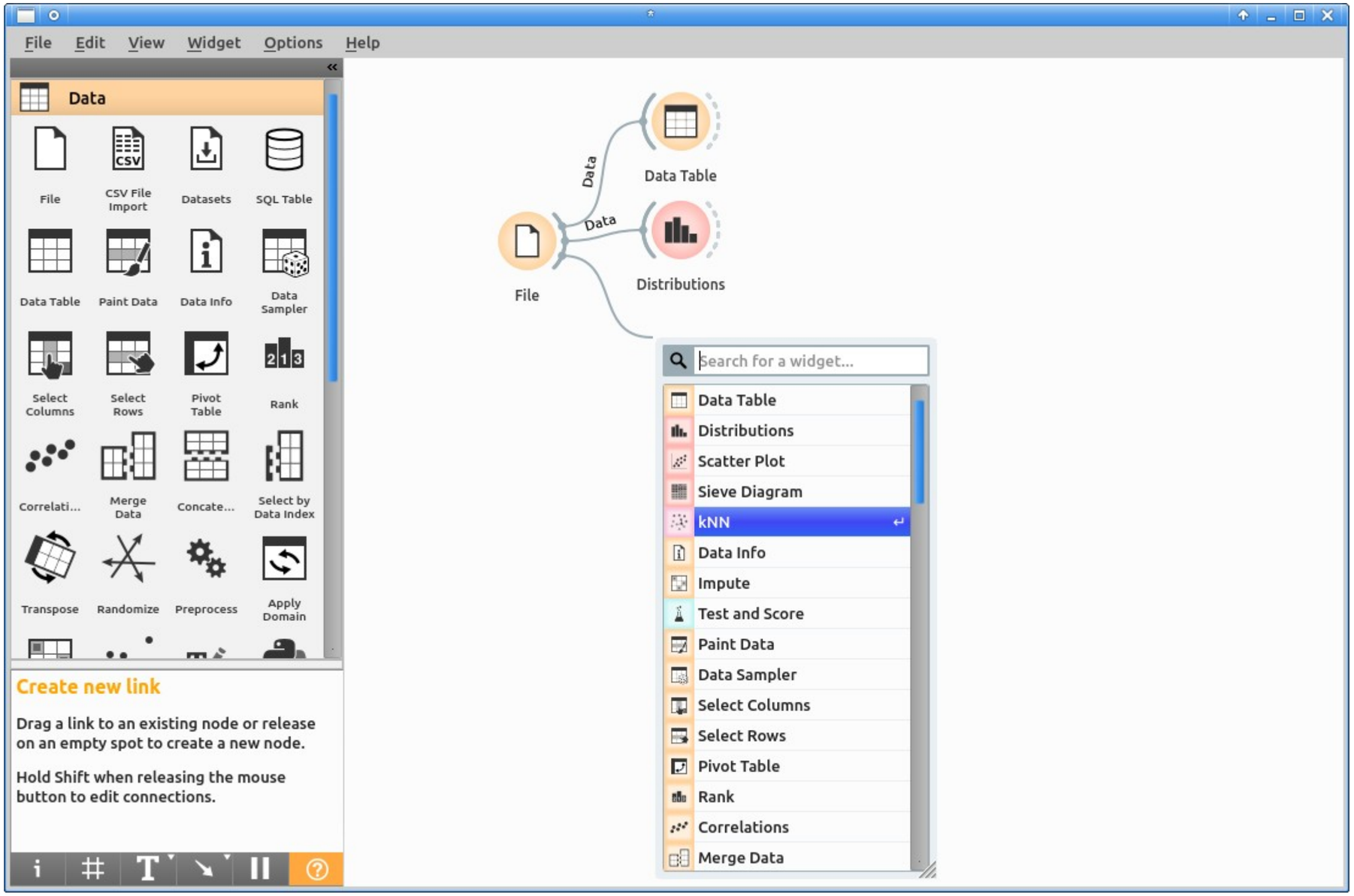
Doesn’t look bad!
But, will this information be helpful to distinguish Sec secreted proteins from the rest?



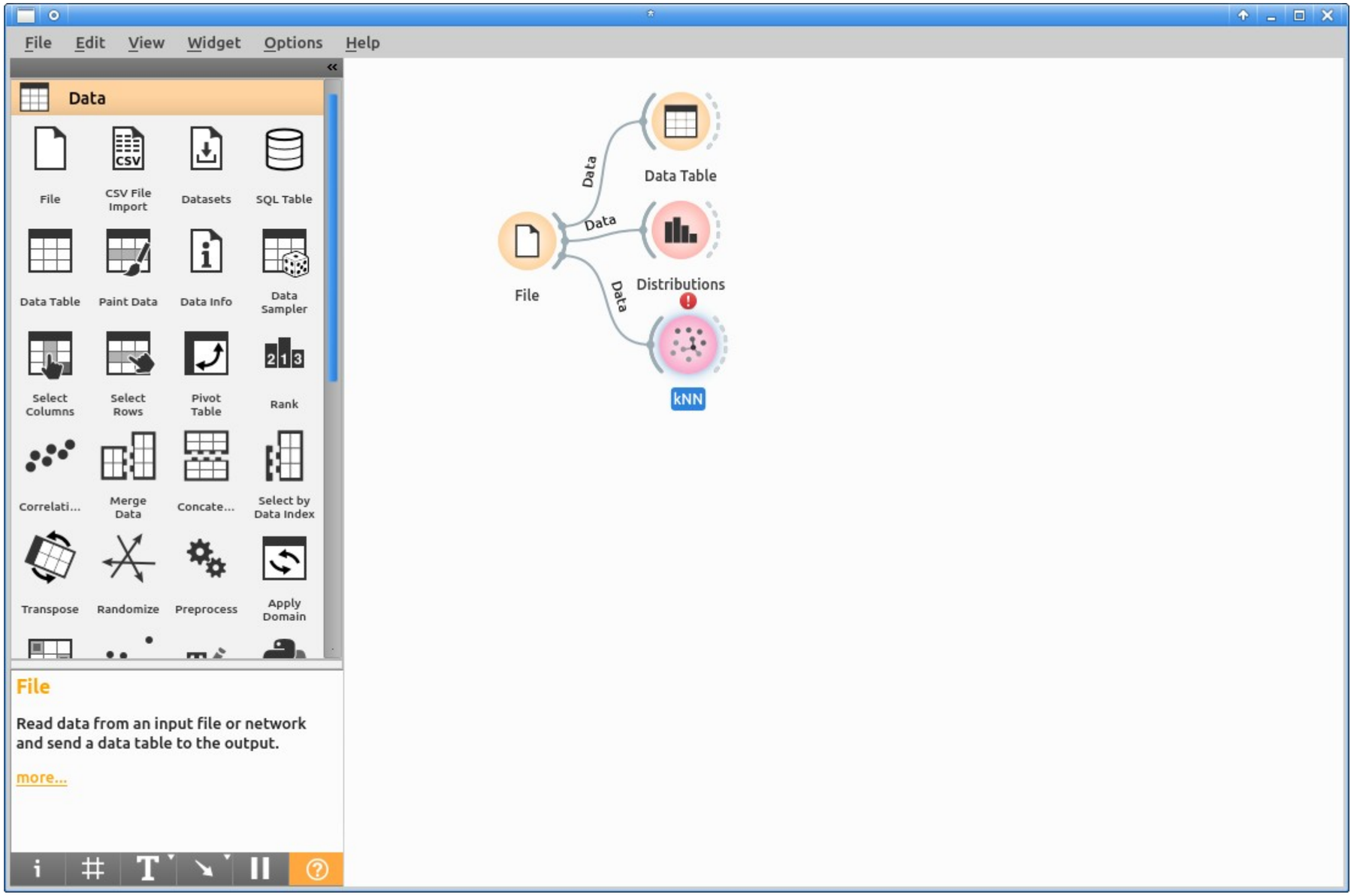
We can check if the distribution is different depending on the secretion route.
Click on the drop down menu that is close to “Split” and select “Route”.



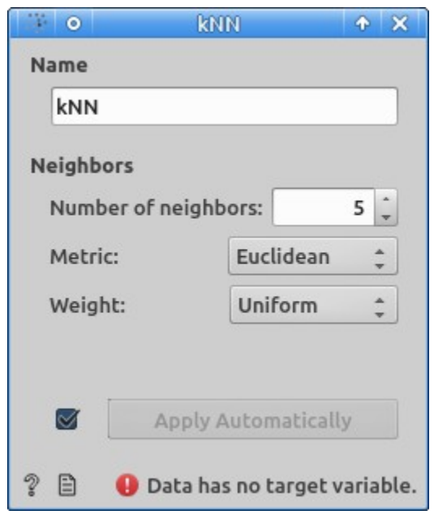
This splits the data according to variable “*Route*”. *Route* is a categorical variable where we have indicated the secretion route for each protein. We get two separate (and superposed) distributions, one for SEC secreted proteins and the other for the rest (NONE). It looks like there may be a difference in the distribution, so maybe this variable will be able to provide some information for classifying proteins by secretion route.



Once we are confident we have variables that may carry meaningful information, we can proceed to train the computer.
k-Nearest Neighbours (*k*NN) is a relatively simple, yet very popular ML method.
Click on the arc by “File”, drag and then click anywhere else. Search for KNN and choose it.



When the widget for kNN appears, you will notice a warning sign (an exclamation mark) above it. This is telling you that there is something missing,

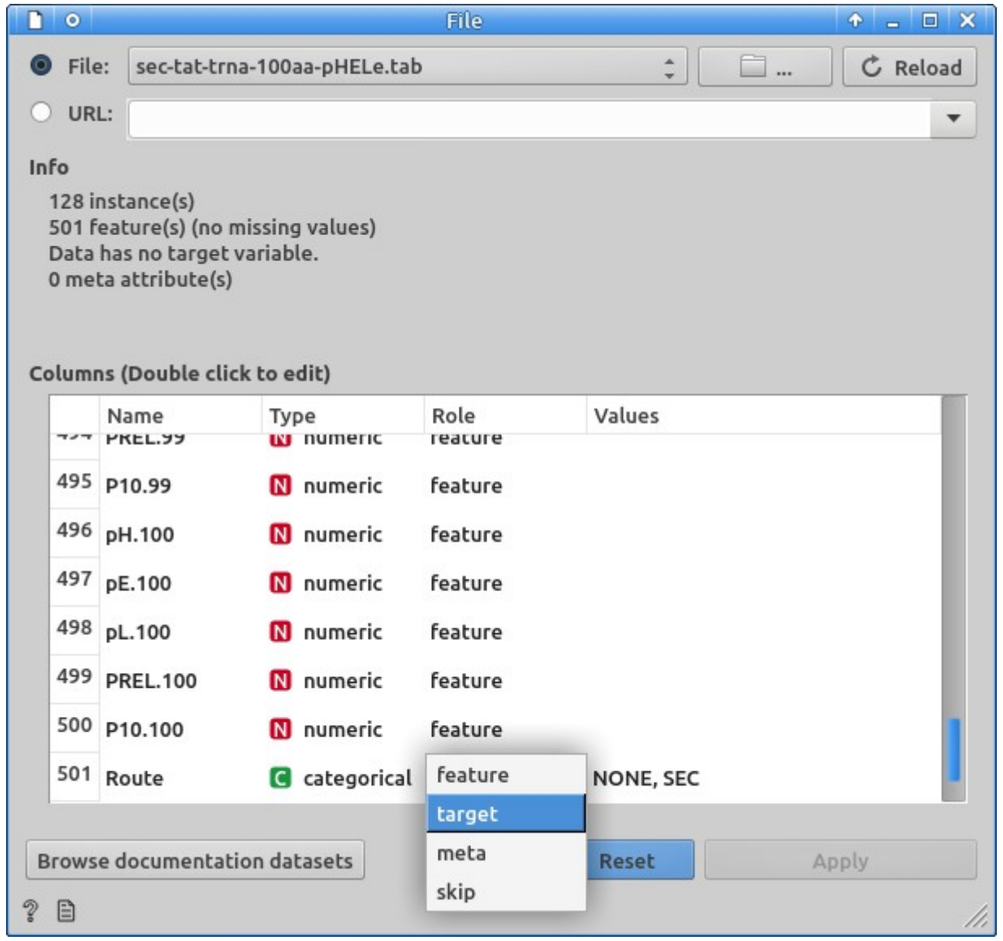


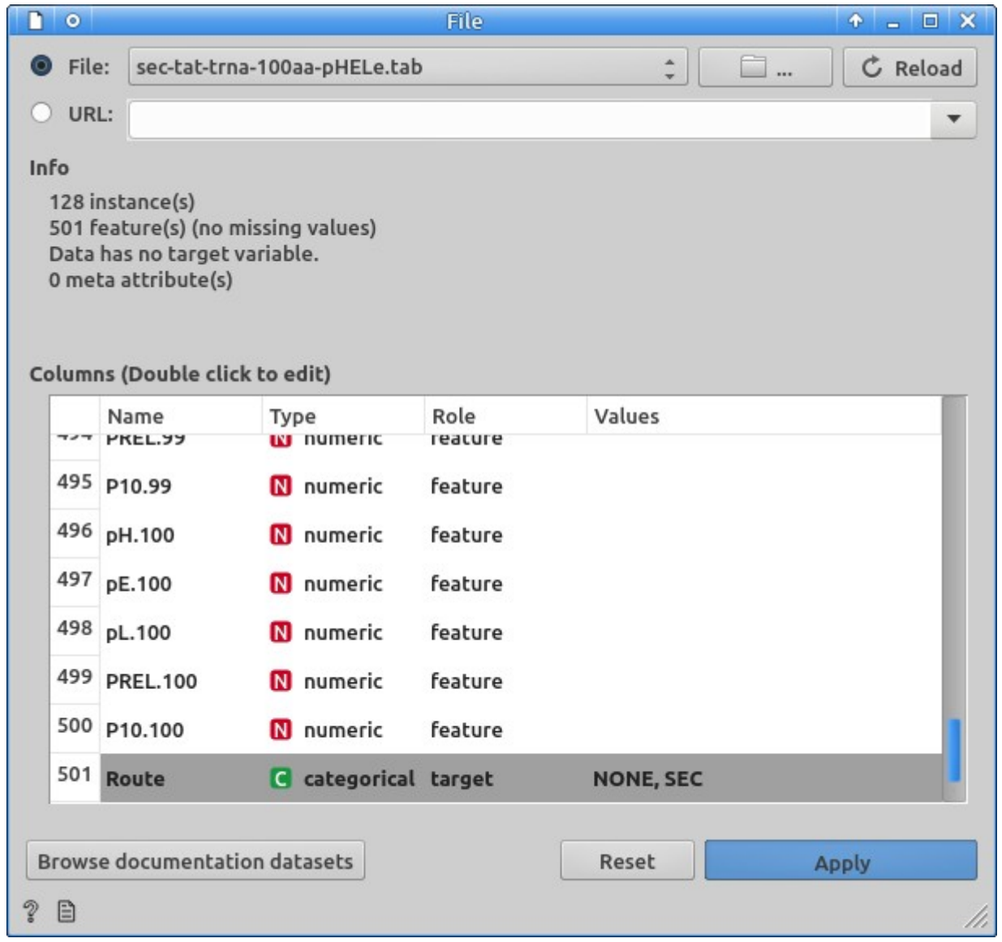
If you double click on the *k*NN widget, you will get to see a dialog with its properties. Let us ignore them for now.
At the bottom of the dialog you will notice a warning message:

“Data has no target variable”

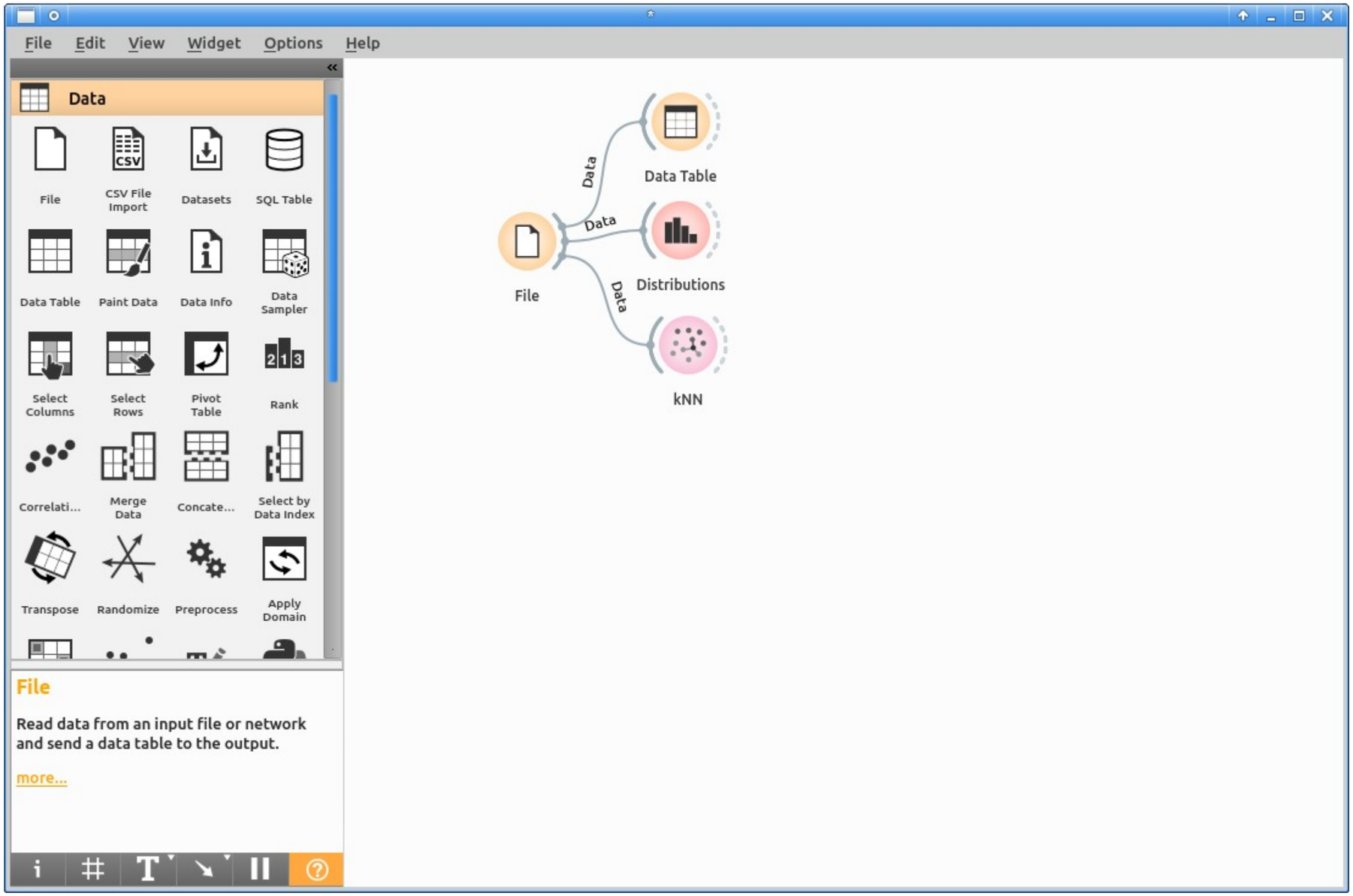
This means that we have not yet told the computer what is it that it must learn or predict.
We want to predict the secretion route (SEC or NONE), so we need to tell it.
Close the window.

Double click again in the “File” widget.
This will open the file selection dialog, and will show you the list of variables.
Scroll down using the scrollbar to the bottom (remember we have several hundred variables).
The last variable should be “Route”.
This is what we want the computer to learn and predict.
On the “Route” row, at the “Role” column, click to unfold the drop down menu and select “**Target**”.
This tells the computer that its target is to predict the “Route”.

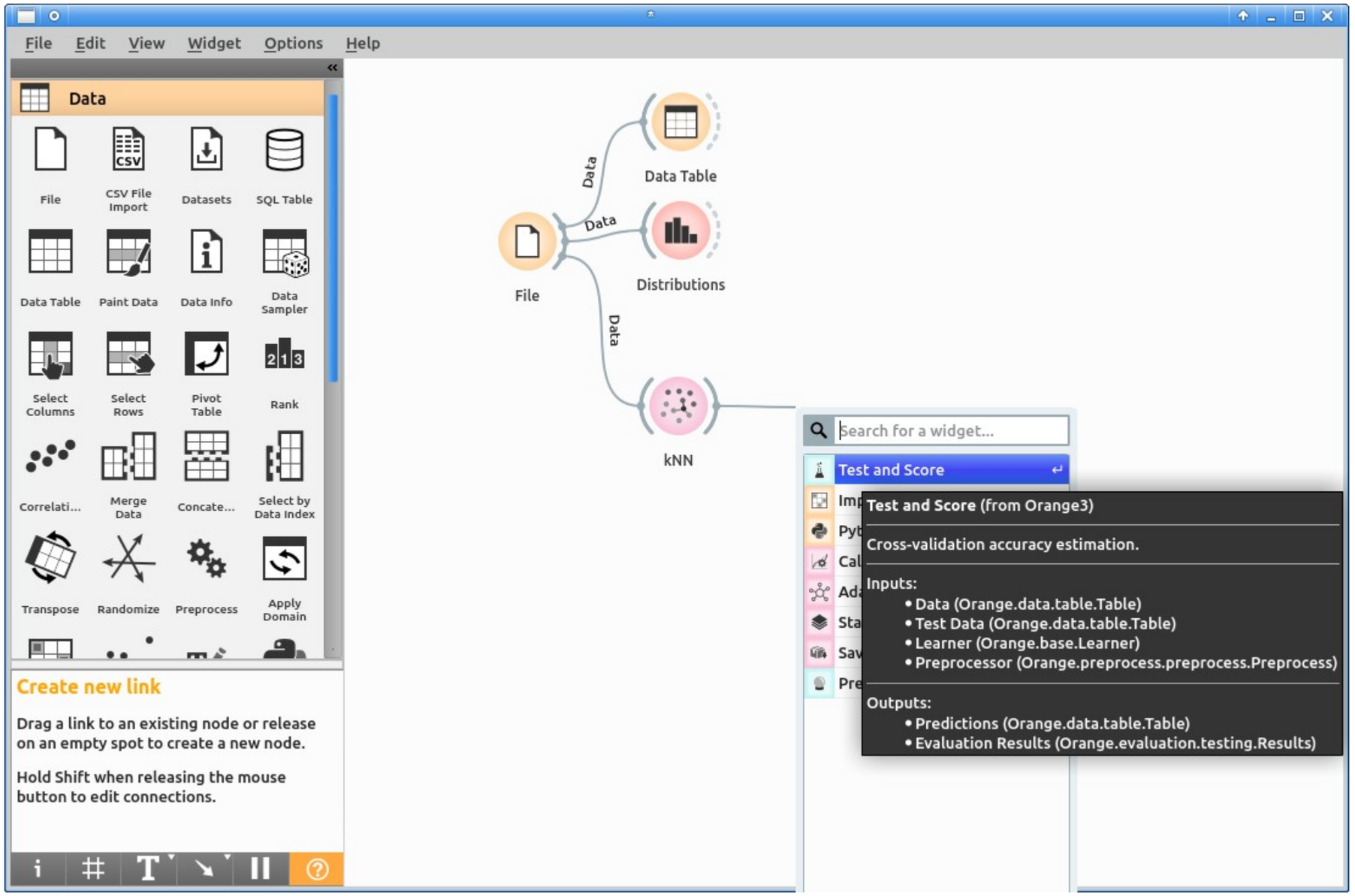




Make sure that the “Route” variable is now defined as the “target” variable (your dialog should look similar to the one in the figure), click on **Apply** and close the dialog window.

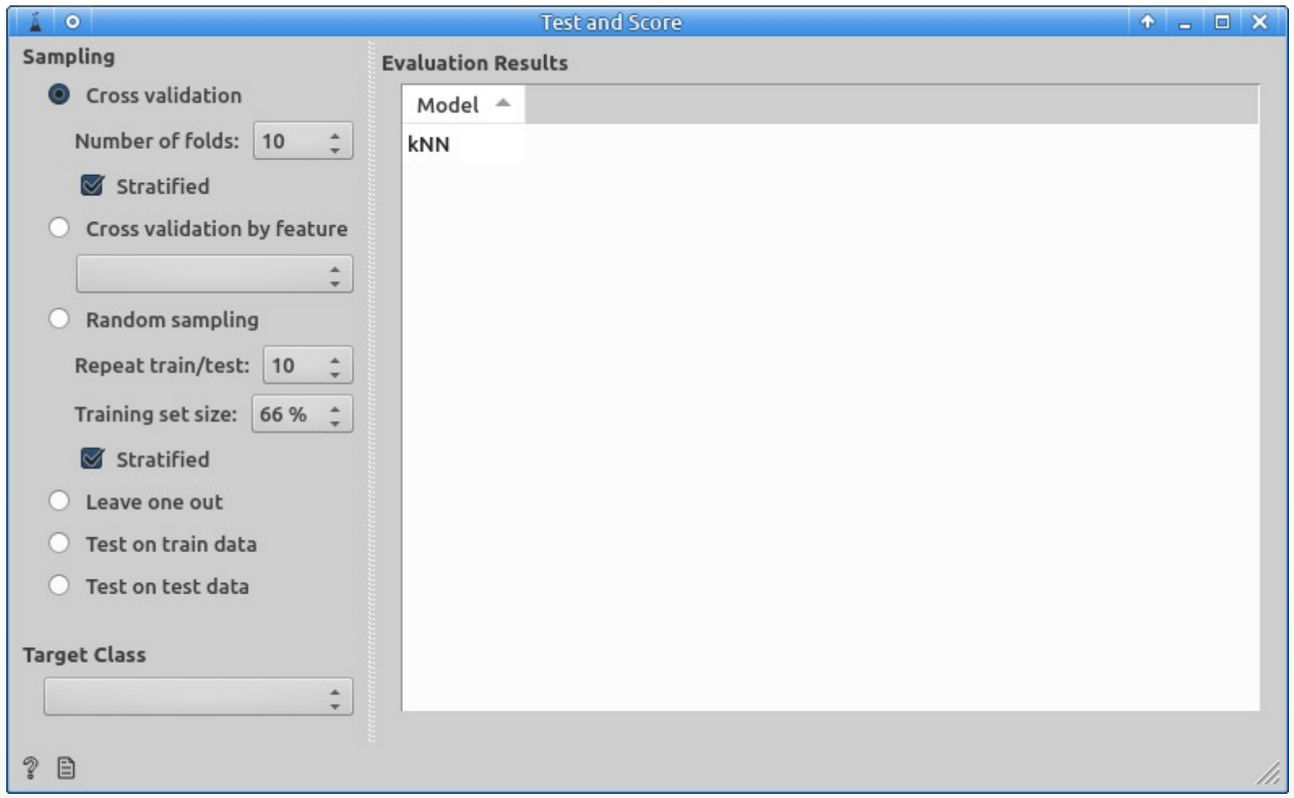


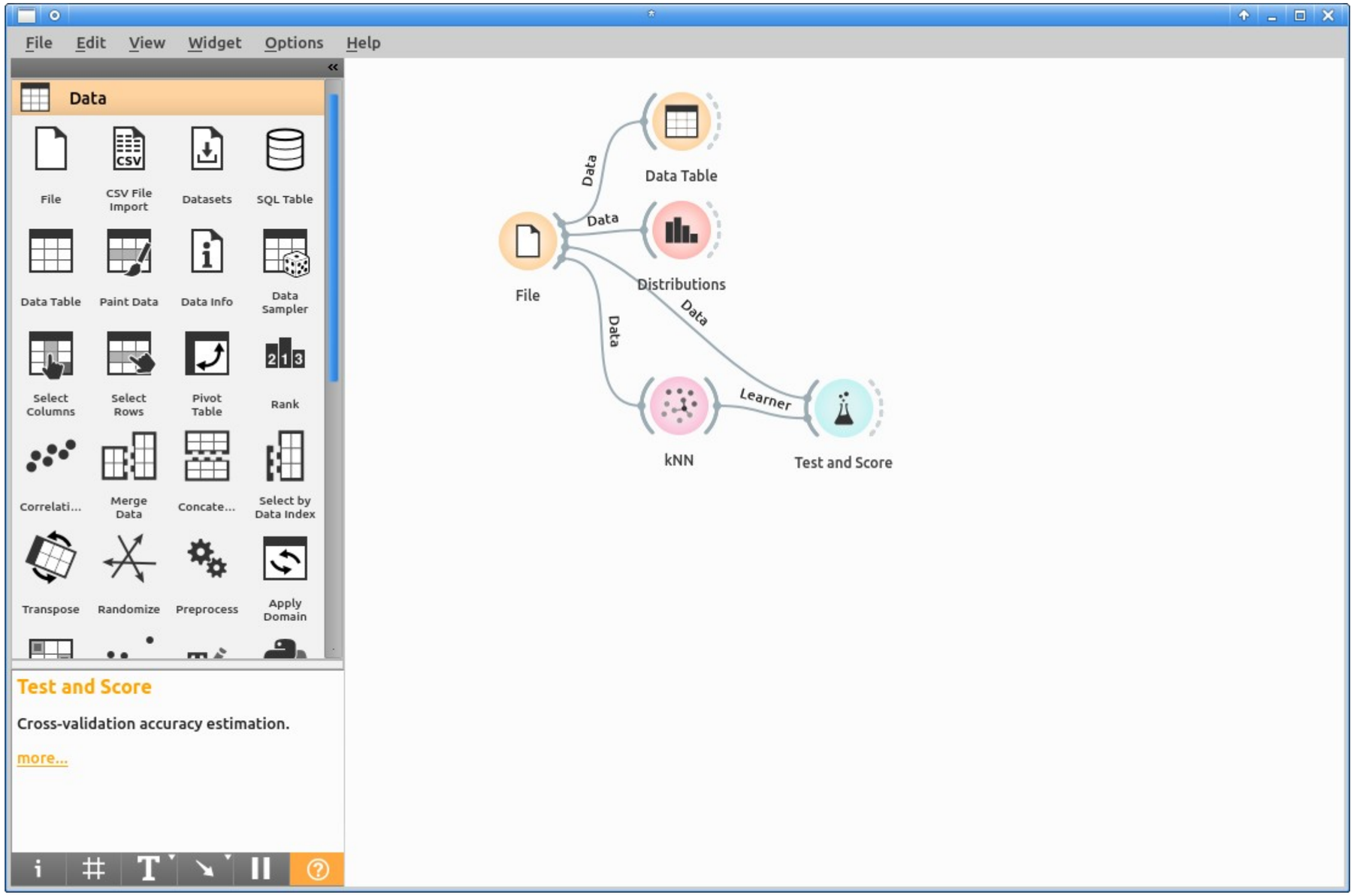
The exclamation mark above the *k*NN widget should have disappeared now. It means we are ready to train the computer. You can double click the icon to modify *k*NN parameters at any time if you want.



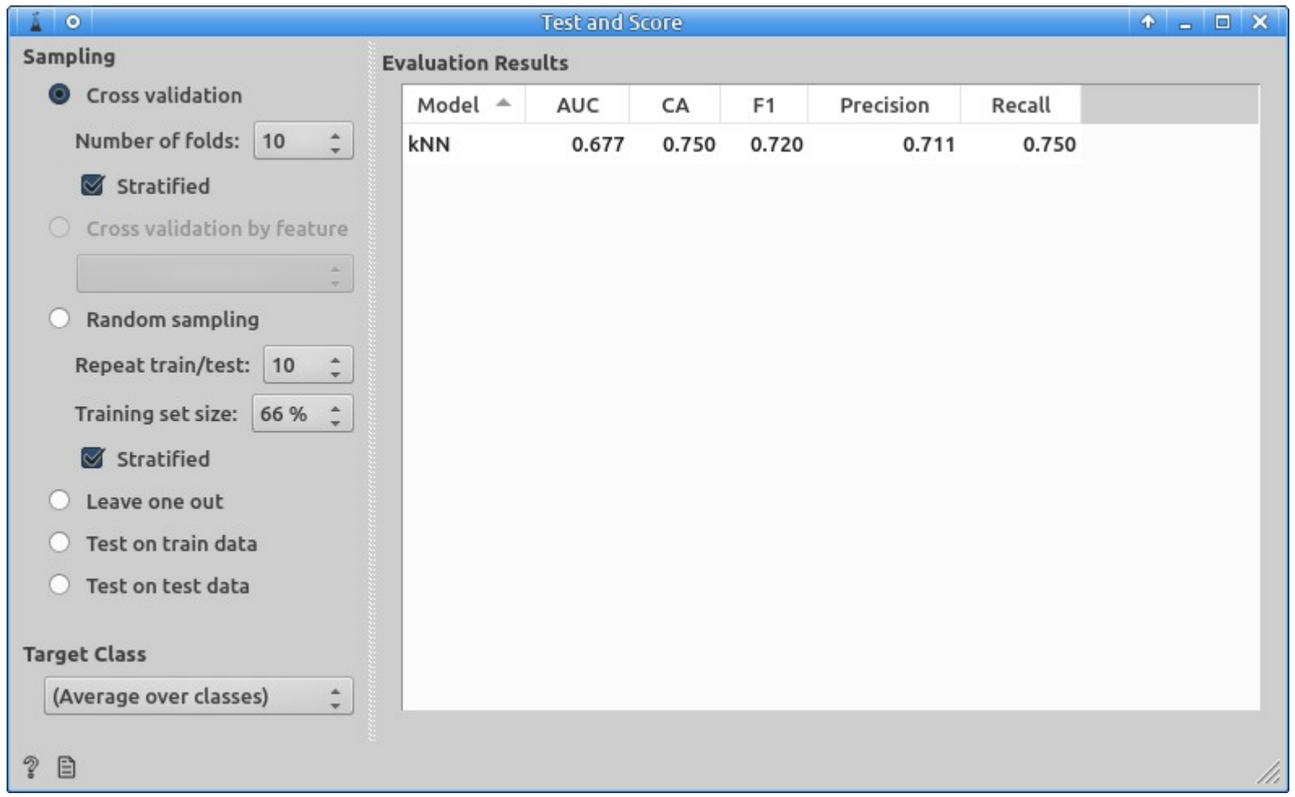
Next, we want to verify how well does this model work for our problem. For this, we need to apply the model to the data and check the results.
Add a new “Test and Score” widget that is connected directly to the *k*NN widget dotted semicircle and double click on it to see the result.

As you can see, besides the *Test and Score* options, there is not much to be seen in the white results window. Remember, we have built a model to learn to discern the secretion route. It is that model that we are testing (that's what the line connecting both widgets means).
But we have no data to check if it has worked. We also need data to test and score our model. Without data, there are no results.

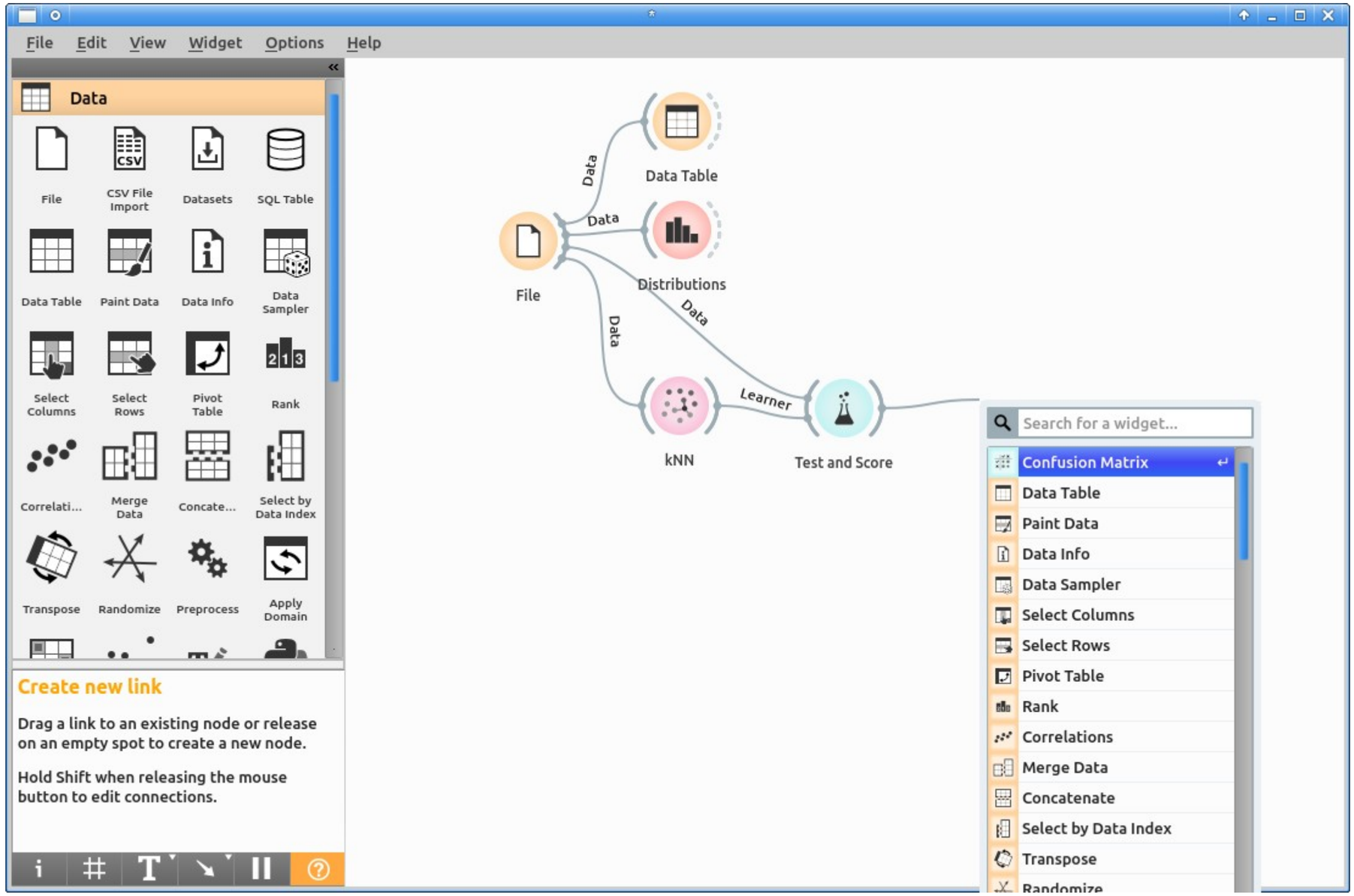




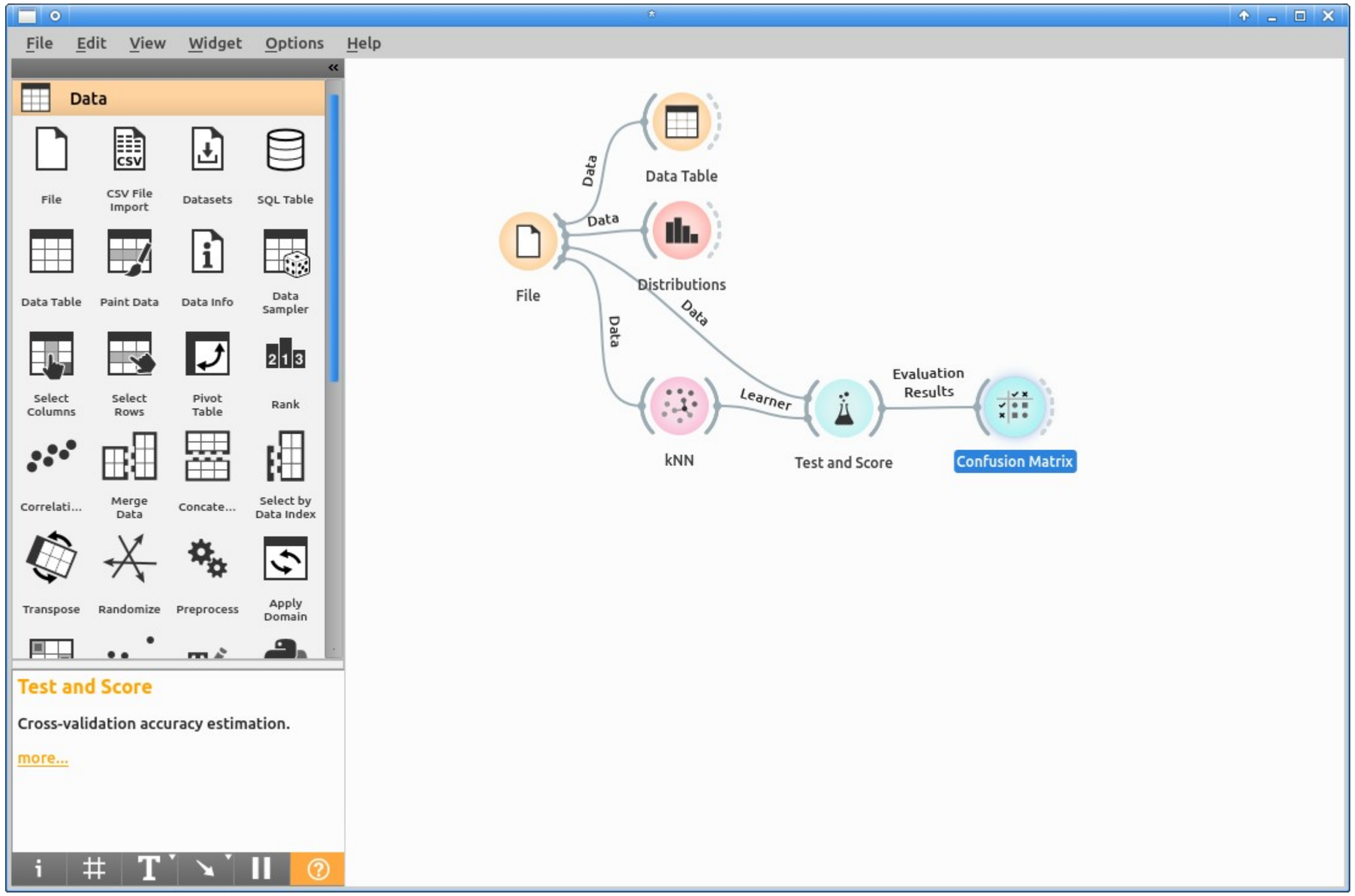
Connect our data source, the “File” widget, with the “Test and Score” widget (you know, click on the “File” semicircle, drag to the “Test and Score” semicircle” and release to connect them).
“File” produces data. We are effectively connecting the data produced by “File” with “Test and Score”.
Now, “Test and Score” is receiving (working with) two objects, the data produced from “File”, and the “kNN” model trained on the data produced by “File”



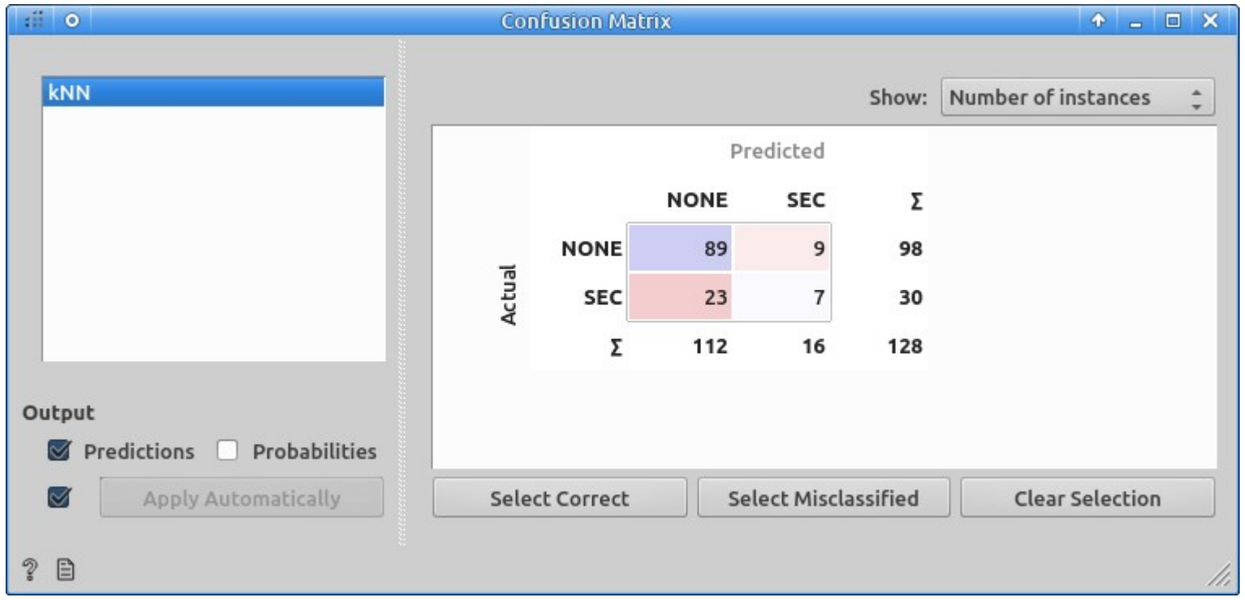
Now, when you double click on “Test and Score”, you will finally see the results.
You can try different “validation” methods if you like.
Well, these results are not very informative for us, humans. We would like to see something that is easier to interpret.
Close the window.



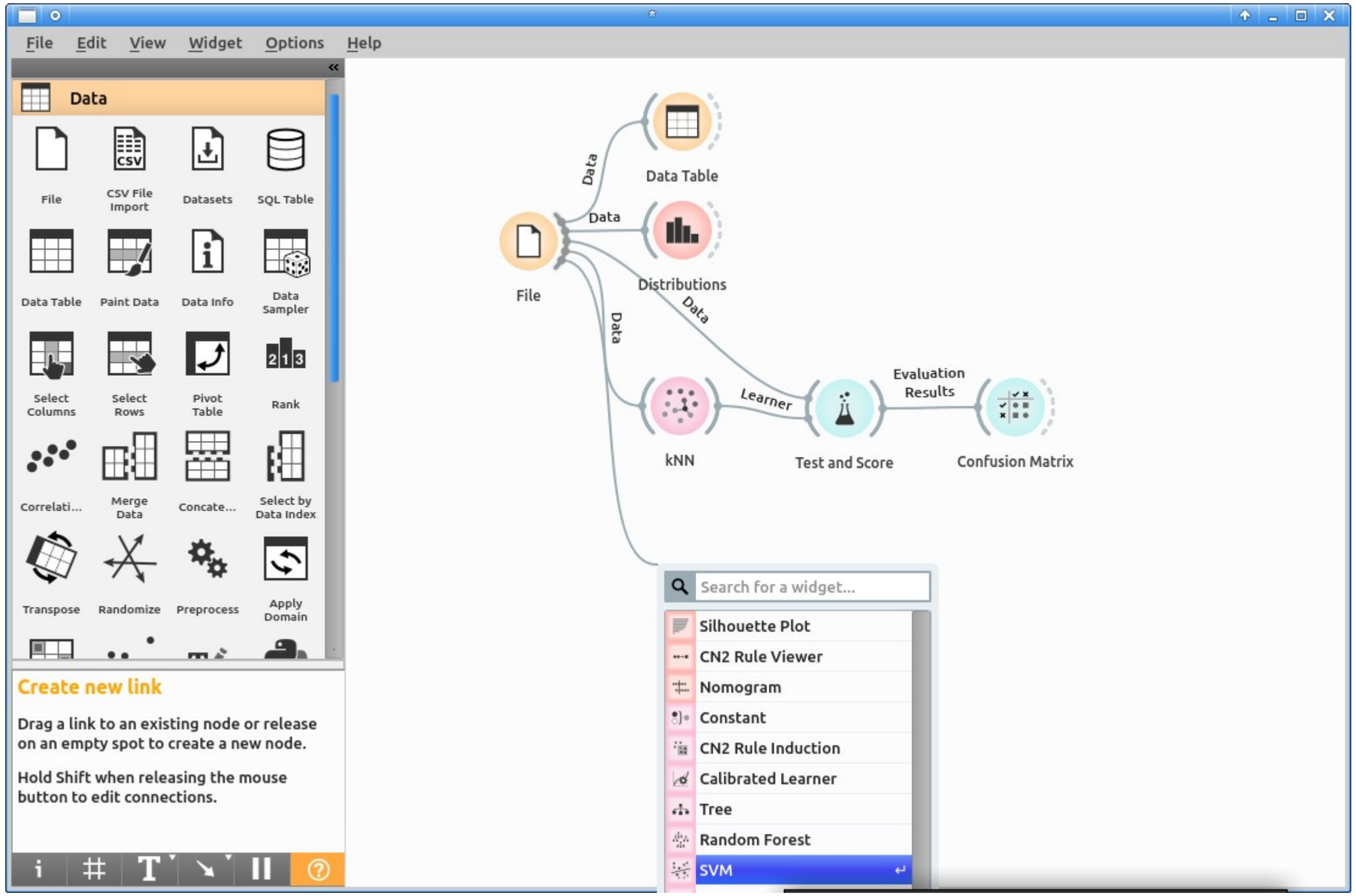
We will use a “Confusion Matrix” to gain some additional insight on the performance of our learner.
Add a new widget that connects to *Test and Score*, and select “Confusion Matrix”.
This tells the computer “take the results of *Test and Score*” and pass them on to “Confusion Matrix”.



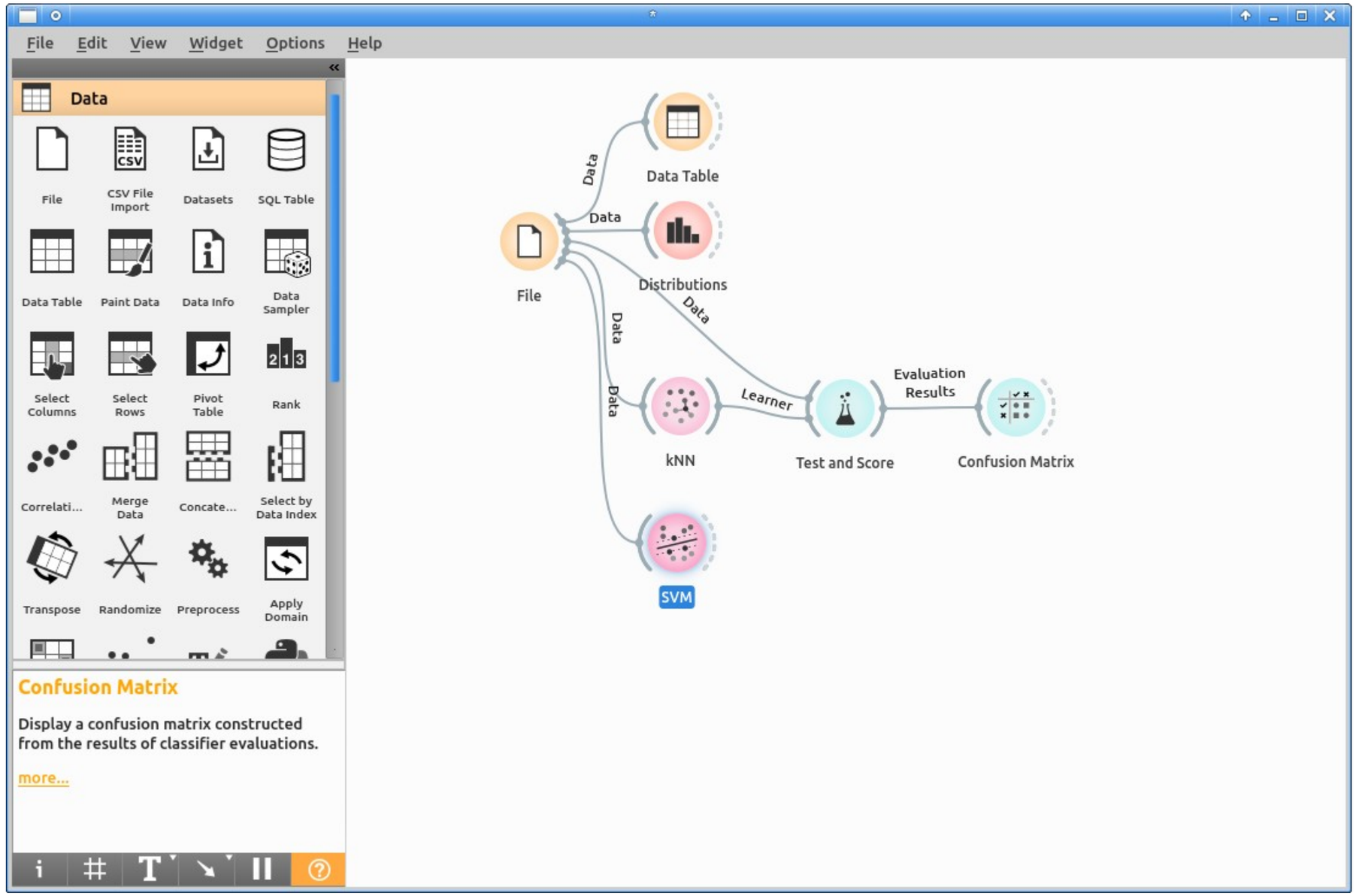
If you look at the canvas, you will notice that lines are labeled with the kind of information that flows through them. “File” is always providing “Data”. “*k*NN” produces a “Learner”, and “Test and Score” produces “Evaluation Results”.
To see what “Confusion Matrix” has done with these evaluation results, double click on it.



A confusion matrix ranks the results according to their validity. It tells you how many sequences that were actually SEC secreted (or not) were predicted correctly or not.
You may choose alternate figures in the drop down menu (sometimes, like here, it is more useful to look at percentages -of right and wrong- than at frequencies).
If you have trouble figuring out the matrix, don't worry. Click on the buttons below the matrix to highlight relevant data (wrong and correct predictions).
When you are done, close the window.



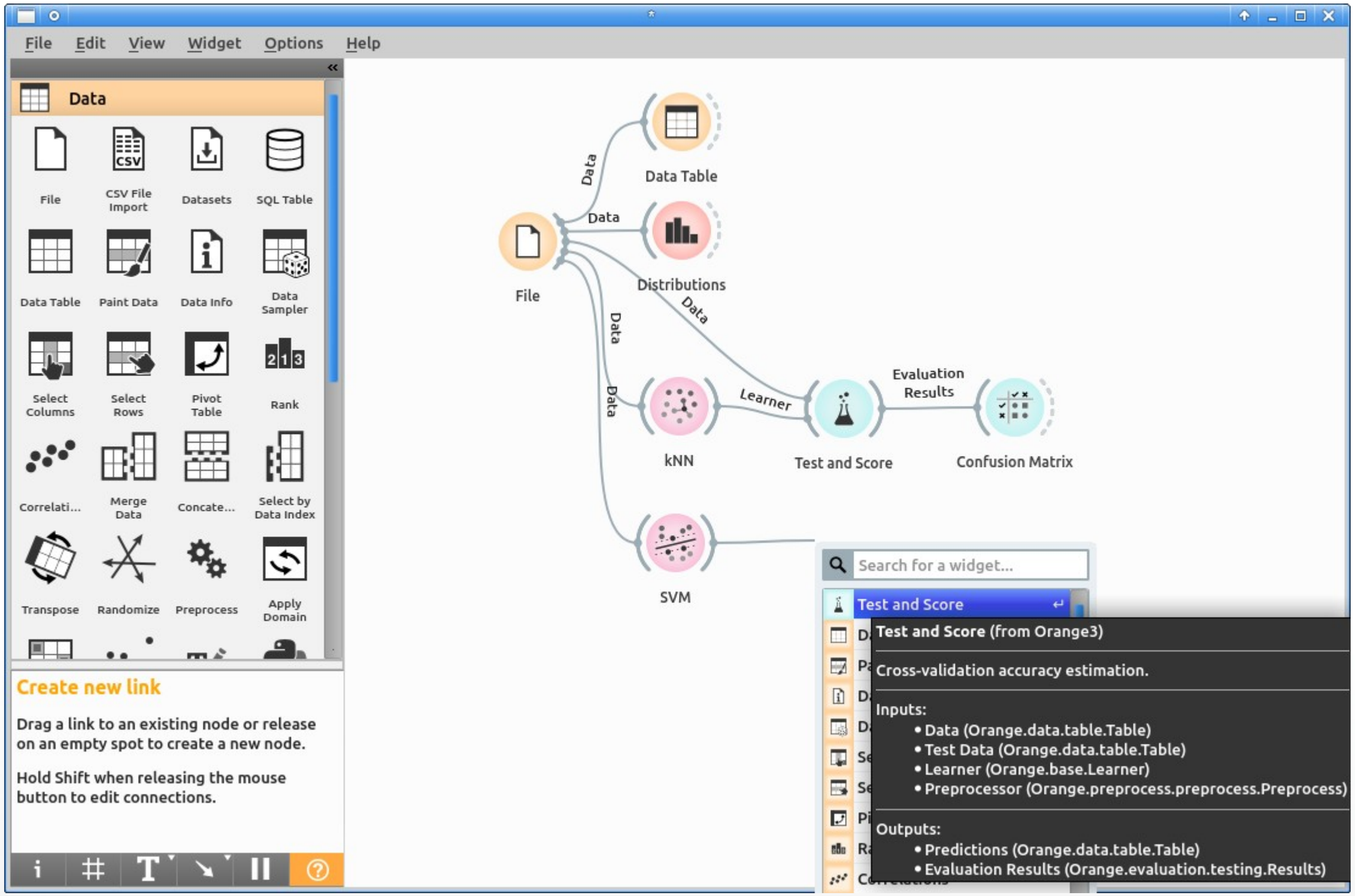
Was this fun? Was it any good?
Let us try a different statistical modeling method. Depending on the problem, some methods may work better than others, and it is rarely obvious which will be the best.
Lets connect our data source (File) to a new widget: a Support Vector Machine (SVM).

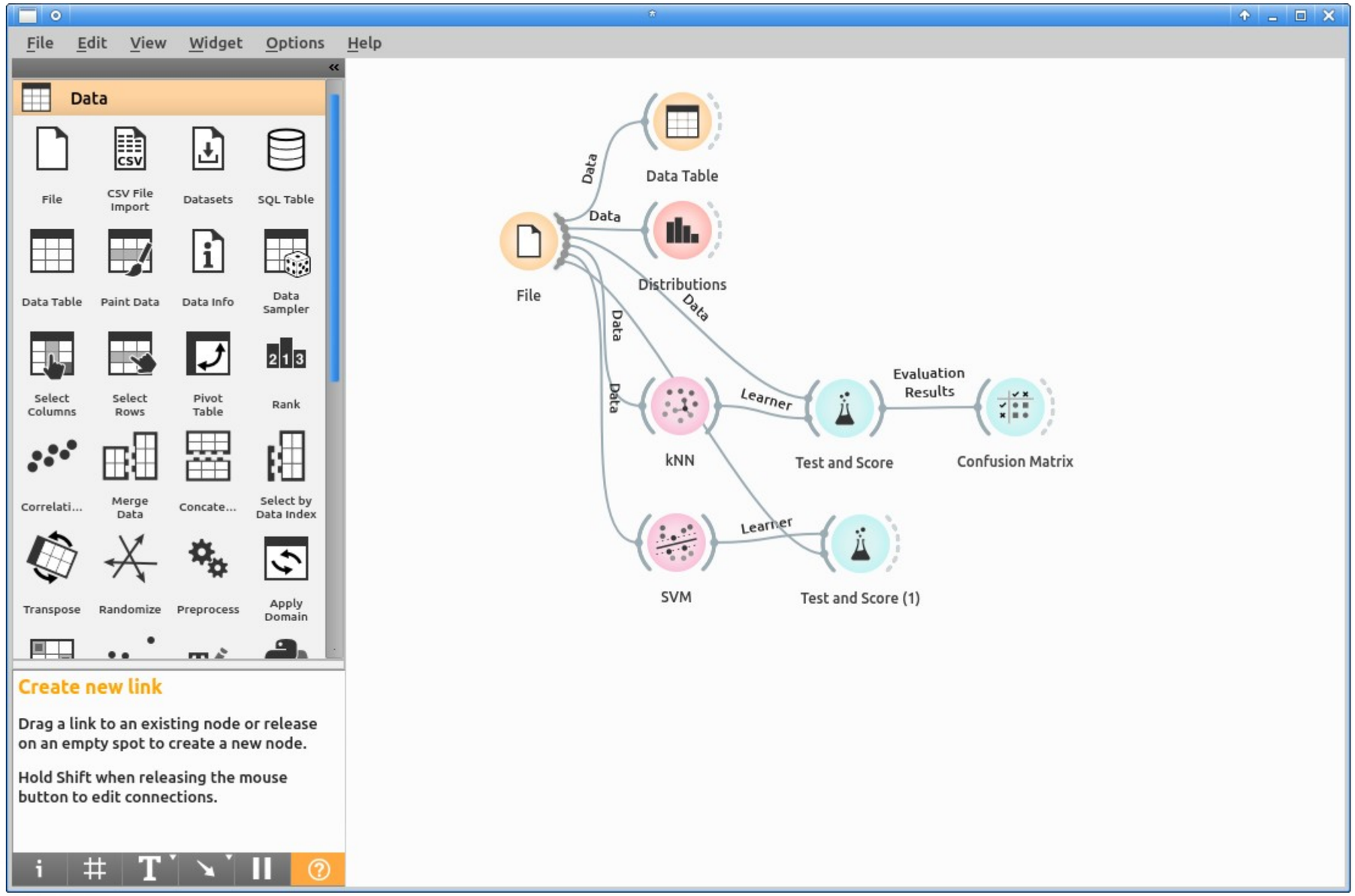


You will notice that this time there is no exclamation mark hinting a warning. The reason is that we have already chosen our target variable, so this time there is no ambiguity.
You can double click on SVM to see the training options available.
For now, let us just proceed to test and score our learner.
Can you guess what comes next?

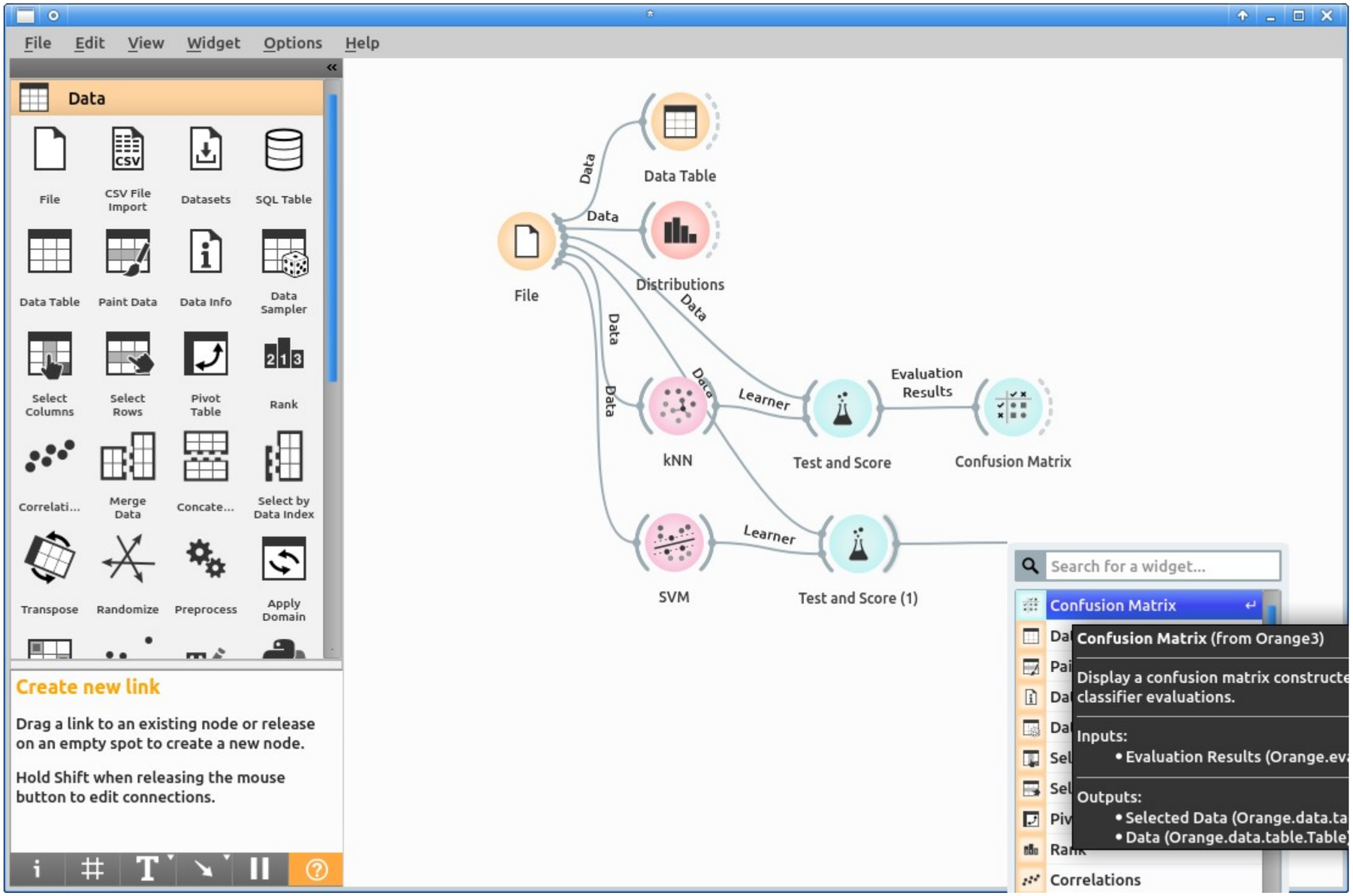
Connect the learner to a new “Test and Score” widget.

Do not worry if you didn’t see this coming. We all make mistakes. That is why we want to teach computers to learn.

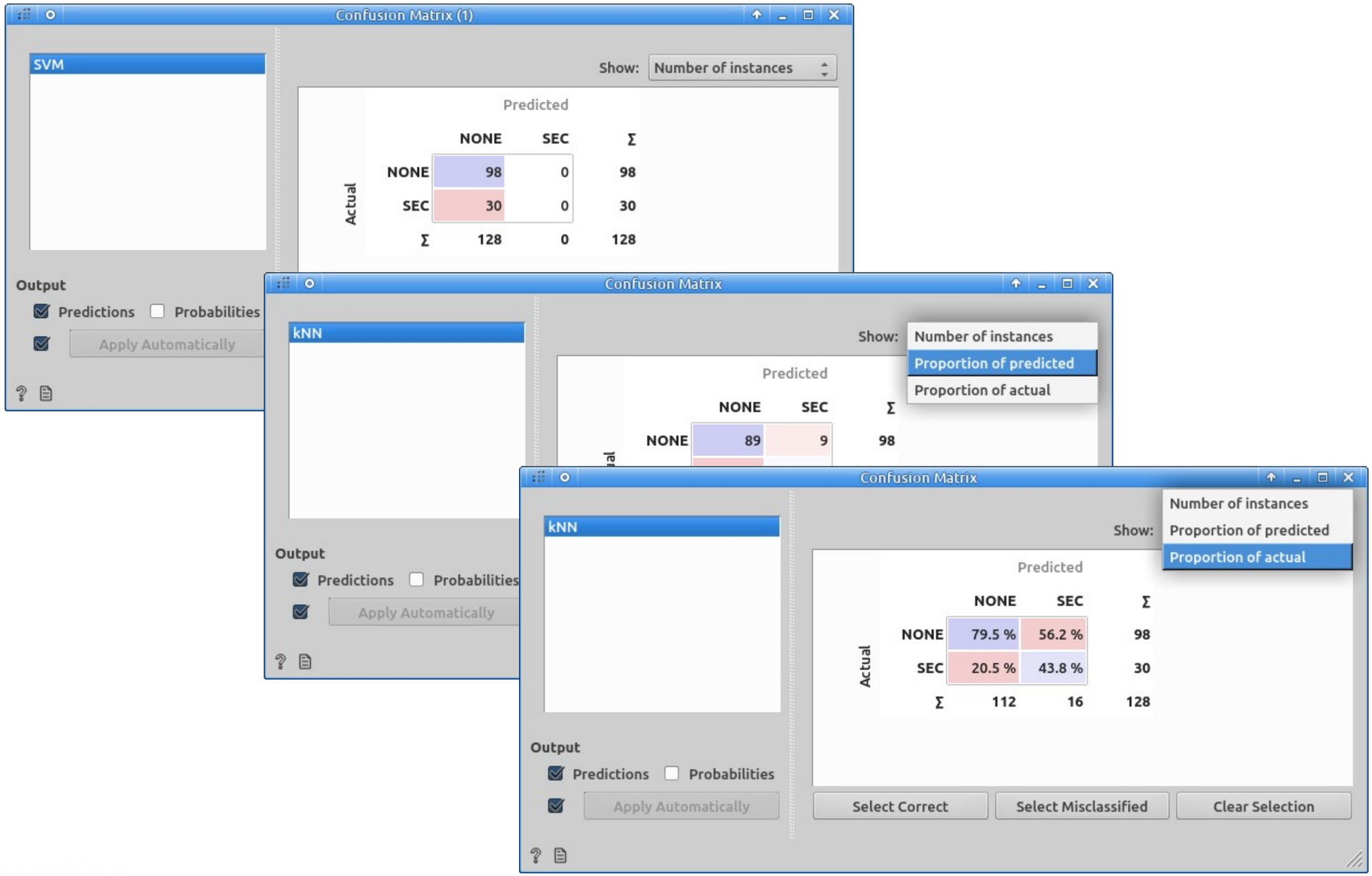




Remember to add a connection between “File” and the newly created “Test and Score” widget as well: we need both, a learner to test and a data to test it on.



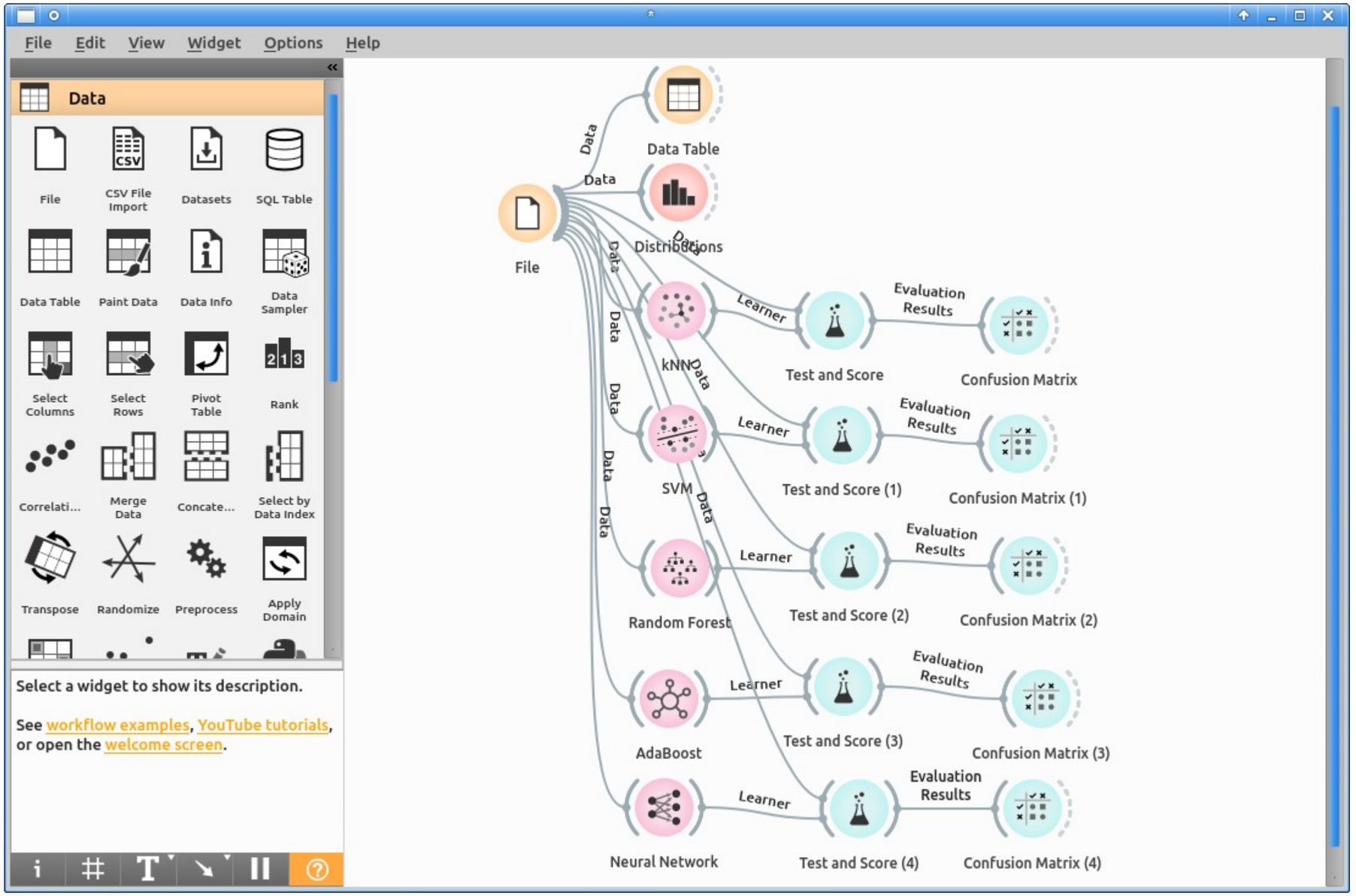
And now, add also a Confusion Matrix to explore the evaluation results for our SVM.



Double click on the new “Confusion Matrix” to see how it behaved.
Did it work better than *k*NN?
You can double click on the *k*NN confusion matrix to recover its results.

Time for fun

- Now that you are becoming confident with Machine Learning, you can start playing like a pro.
- *k*NN and SVM are two very popular (and efficient) methods highly used in Biology.
- Other such methods are PCA, CCA, Random Forest, AdaBoost and Neural Networks.
- As an exercise, repeat the training with Random Forest, AdaBoost and Neural Network.
- When you have finished your workflow should look like the next slide.


File
Edit
View
Widget
Options
Help
Data
File
CSV File Import
Datasets
SQL Table
Data Table
Paint Data
Data Info
Data Sampler
Select Columns
Select Rows
Pivot Table
Rank
Correlati...
Merge Data
Concate...
Select by Data Index
Transpose
Randomize
Preprocess
Apply Domain
Select a widget to show its description.
See workflow examples, YouTube tutorials, or open the welcome screen.
Data Table
Distributions
File
kNN
SVM
Random Forest
AdaBoost
Neural Network
Learner
Evaluation Results
Test and Score
Confusion Matrix
Test and Score (1)
Confusion Matrix (1)
Test and Score (2)
Confusion Matrix (2)
Test and Score (3)
Confusion Matrix (3)
Test and Score (4)
Confusion Matrix (4)

What Next?

- **Good news**! From here on, it is a downward slope. You just need to
  - Play with learners, widgets and their settings
  - Learn more about them
- At some point, you will want to use more advanced settings or will need to create many analyses
  - You can still do that graphically
  - You may prefer to learn programming with R program to get maximal power and flexibility.

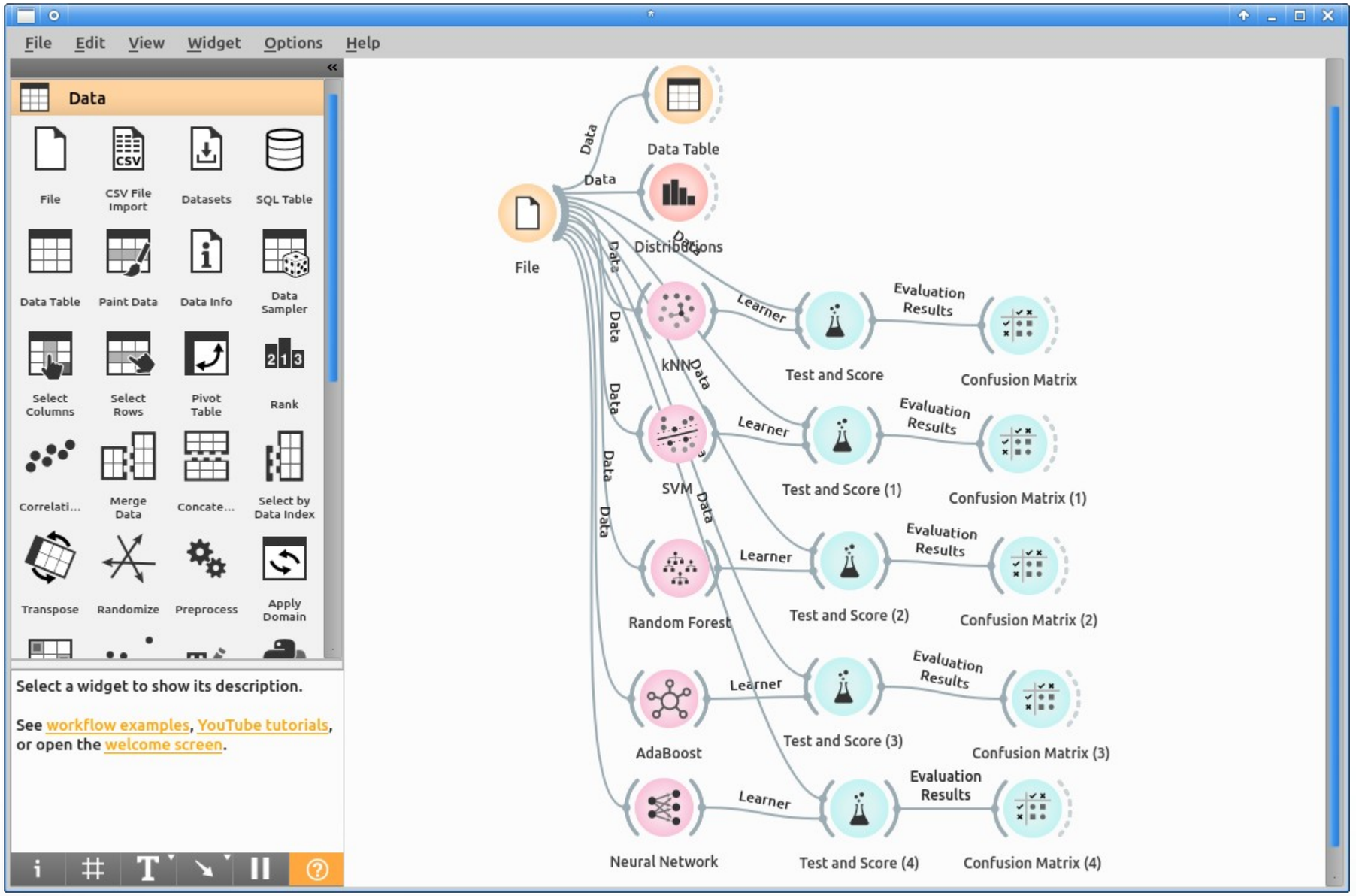
What before?

- These are the **bad news**.
- Here, we started with a highly processed data set. Getting the data, however, required a complex and large research process previous to training the computer.
- Typically, the most difficult task will be choosing the variables to include, ensuring you get quality data, verifying the data provides useful information, discarding unneeded data, etc...
- Which is the core of research.

Another exercise

- In this case, our results may be compromised because we only considered one secretion route.
- Data set “**sec-tat-trna-pHELe-100aa-3.tab**” contains exactly the same data with only one difference: proteins are tagged “SEC”, “TAT” or “NONE” instead of only “SEC” and “NONE”.
- Repeat the whole process with this data set.
- Extra score points will be awarded for doing it faster. Can you figure out how?

Hint:



When you are done, your workflow should look like this.

Closing

- You may want to play, explore and experiment. You can double click on a learner, to build a new model with different settings: it will happen automatically when you close the configuration window.
- You can save your workflow for later, to share it with colleagues, or for publication.
- When you are done, simply close the program from the “File” menu as usual.