SOFTWARE TOOL ARTICLE

Visualize omics data on networks with Omics Visualizer, a Cytoscape App [version 1; peer review: awaiting peer review]

Marc Legeay, Nadezhda T. Doncheva, John H. Morris, Lars Juhl Jensen

1Novo Nordisk Foundation Center for Protein Research, University of Copenhagen, Copenhagen, Denmark
2Center for Non-Coding RNA in Technology and Health, Department of Veterinary and Animal Sciences, University of Copenhagen, Copenhagen, Denmark
3Resource for Biocomputing, Visualization and Informatics, University of California, San Francisco, USA

Abstract

Cytoscape is an open-source software used to analyze and visualize biological networks. In addition to being able to import networks from a variety of sources, Cytoscape allows users to import tabular node data and visualize it onto networks. Unfortunately, such data tables can only contain one row of data per node, whereas omics data often have multiple rows for the same gene or protein, representing different post-translational modification sites, peptides, splice isoforms, or conditions. Here, we present a new app, Omics Visualizer, that allows users to import data tables with several rows referring to the same node, connect them to one or more networks, and visualize the connected data onto networks. Omics Visualizer uses the Cytoscape enhancedGraphics app to show the data either in the nodes (pie visualization) or around the nodes (donut visualization), where the colors of the slices represent the imported values. If the user does not provide a network, the app can retrieve one from the STRING database using the Cytoscape stringApp. The Omics Visualizer app is freely available at https://apps.cytoscape.org/apps/omicsvisualizer.

Keywords

Cytoscape, app, network visualization, omics data, network biology

This article is included in the Cytoscape Apps gateway.
Corresponding authors: Marc Legeay (marc.legeay@cpr.ku.dk), Lars Juhl Jensen (lars.juhl.jensen@cpr.ku.dk)

Author roles: Legeay M: Software, Writing – Original Draft Preparation, Writing – Review & Editing; Doncheva NT: Software, Writing – Review & Editing; Morris JH: Conceptualization, Writing – Review & Editing; Jensen LJ: Conceptualization, Writing – Review & Editing

Competing interests: No competing interests were disclosed.

Grant information: This work was funded by the Novo Nordisk Foundation (NNF14CC0001). NTD was also funded by InnovationFund Denmark "Brainstem" (4108-00008B).

Copyright: © 2020 Legeay M et al. This is an open access article distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

How to cite this article: Legeay M, Doncheva NT, Morris JH and Jensen LJ. Visualize omics data on networks with Omics Visualizer, a Cytoscape App [version 1; peer review: awaiting peer review] F1000Research 2020, 9:157 (https://doi.org/10.12688/f1000research.22280.1)

**Introduction**

Cellular functions are mediated by complex networks of interactions between genes, proteins, and other molecular entities. Omics technologies are commonly used to measure the detailed regulation of these networks by quantifying changes of individual post-translational modification (PTM) sites, peptides, or splice isoforms across different experimental conditions. However, it is not easy to visualize such data sets, which have multiple values per gene or protein, onto the networks using existing network visualization tools such as Cytoscape\(^3\) or Gephi\(^3\).

To address this, we present the new Omics Visualizer app for Cytoscape\(^3\). The app allows users to import data tables with several rows referring to the same node and to visualize such data on networks; while designed with omics data in mind, the app is data agnostic. These values can be shown directly on the nodes of the networks as pie or donut visualizations, in which the color of each slice represents a different value for the same node. The Omics Visualizer app was implemented using the API that Cytoscape makes available to developers, and the data visualization builds upon the enhancedGraphics app\(^5\). The app furthermore integrates with the stringApp\(^3\) to facilitate easy visualization of data onto networks from the STRING database\(^6\) and supports Cytoscape Automation to facilitate integration with other tools\(^1\).

**Methods**

**Implementation**

The typical Omics Visualizer\(^3\) workflow consists of four steps: importing data as a table, optionally filtering the data, connecting it to one or more networks, and finally visualizing the connected data onto the networks.

The Omics Visualizer table import mimics the Cytoscape default import process. The app can handle text files (e.g. comma- or tab-delimited values) as well as spreadsheet files from e.g. Microsoft Excel. The app auto-detects the data type of each column, and the user can subsequently select which columns to import and modify the auto-detected data type for each column if needed. The import creates two private unassigned tables in Cytoscape: one that contains the data imported from the file, and another that stores all associated Omics Visualizer properties. These tables are private, which means that the user cannot interact with them directly in the Cytoscape UI. Instead, the data can be viewed (but not edited) through the Omics Visualizer panel located next to the node, edge, and network table panels. The tables are also accessible via the command interface and the Cytoscape API.

If the data was not already filtered before importing it into Cytoscape, Omics Visualizer enables users to do so afterwards. Omics Visualizer can filter the rows based on selected columns with numeric, string, and Boolean values and offers several operators depending on the data type: \texttt{EQUALS}, \texttt{NOT_EQUALS}, \texttt{NULL} and \texttt{NOT_NULL} for all types; \texttt{CONTAINS}, \texttt{NOT_CONTAINS} and \texttt{MATCHES} for string values; \texttt{LOWER}, \texttt{LOWER_EQUALS}, \texttt{GREATER} and \texttt{GREATER_EQUALS} for numeric values. Similar to the Cytoscape selection filters, Omics Visualizer interface allows the user to create Boolean formulas with nested criteria. Once the filter is applied, rows that do not satisfy the filter are hidden in the panel, and the active filter is stored in the properties table to allow it to be changed later.

To visualize the data table onto a network, the Omics Visualizer table must first be connected to the network by specifying matching key columns in the node and Omics Visualizer tables. This connection information is stored in the network table, which is used to recreate the link when a session file is loaded. An Omics Visualizer table can be connected to several networks, but a network can only be connected to one Omics Visualizer table. A node table key can match the key in several Omics Visualizer table rows, in which case the node is connected to all of them. The number of connected rows for each node is stored in the node table; note that this number does not reflect any filtering.

Alternatively, Omics Visualizer can retrieve and automatically connect a network from the STRING database\(^6\) using the Cytoscape stringApp\(^3\). The user can select any column with gene/protein identifiers, which will be used to first query STRING to retrieve a network and subsequently as the key for connecting the network to the Omics Visualizer table.

Once connected, the user can show the data from the Omics Visualizer table on networks using either a pie or a donut visualization. The data values are mapped to colors using either a discrete mapping, where the user chooses the colors for each value, or a continuous mapping, where the user defines a color gradient. For the continuous mapping, the user specifies three values and corresponding colors, namely the minimum, middle, and maximum. Every value lower than the minimum or greater than the maximum value will be shown using the minimum and maximum color, respectively. Predefined color mappings can be used with the help of Cytoscape palettes such as ColorBrewer\(^4\) and Viridis (originally from Matplotlib\(^4\)). The charts are drawn thanks to the enhancedGraphics app\(^5\). This app draws charts based on a description string, which is specified as a Cytoscape Custom Chart node style.

When the user creates a visualization, Omics Visualizer creates a column in the network table and several columns in the node table, and then it activates a Custom Chart node style: Omics Visualizer uses Custom Chart 7 to visualize pies and Custom Chart 8 to visualize donuts. The network table column is used to store the visualization properties. One node table column is created to store the enhancedGraphics string. Several node table columns are created to store the values of the visualizations: one column for pie visualization, and one column per ring for donut visualization. The values from the Omics Visualizer table rows are formatted into lists to fit the enhancedGraphics requirements. The enhancedGraphics continuous mapping always ranges from minimum to zero, and zero to maximum. Omics Visualizer thus first centers the values around the middle value and adjusts the minimum and maximum accordingly. The values are only modified in the node table columns, not in the actual Omics Visualizer table.
Omics Visualizer can automatically generate legends from the visualizations in the form of Cytoscape annotations, which can be exported as part of the images if the user exports the network. To allow users to easily modify the legend, each element of the legend is a separate annotation. These are grouped so that the user can move or delete the legend in one click. The name of the annotation group is used to differentiate the legend annotations from other annotations and should thus not be changed. When the legend is created, Omics Visualizer creates a network table column to store it.

All the columns created by Omics Visualizer in the node or network tables are in the namespace “Omics Visualizer”, enabling the user to easily identify and hide them if desired.

Omics Visualizer is built from Java 1.8 using Cytoscape API 3.7, meaning the minimum required version of Cytoscape is 3.7.0.

**Operation**

Cytoscape can be used with R or Python through to Cytoscape Automation. Omics Visualizer implements commands in the specific ‘ov’ namespace, allowing Omics Visualizer to be used with the REST API. A full documentation of the commands is available at [https://github.com/marclegeay/omics-visualizer/blob/master/automation_documentation.md](https://github.com/marclegeay/omics-visualizer/blob/master/automation_documentation.md). With the commands, the user can import a table, filter it, connect it with an existing network, retrieve a STRING network, create visualizations, and generate legends.

**Use cases**

We will illustrate how to use Omics Visualizer by visualizing site-specific proteomics data from a phosphoproteomics study of ovarian cancer by Francavilla et al.\(^1\). This study compares the phosphoproteome of primary cells derived from epithelial ovarian cancer (EOC) and two healthy tissues, namely ovarian surface epithelium (OSE) and distal fallopian tube epithelium (FTE). The goal of the study was to uncover cancer-specific changes in expression, phosphorylation state, and kinase signatures by comparing cancer and two healthy tissues (EOC vs. OSE, and EOC vs. FTE) for each site. The sites were then clustered into three clusters: A) sites abundant in healthy (OSE and FTE) but not cancerous tissues (EOC); B) sites abundant in FTE and EOC, but not OSE; and C) sites abundant only in cancerous tissue (EOC).

In this example, we want to visualize this site-specific data set on a network in which each node is a protein. We start from Table S3 from Francavilla et al., where each line represents a phosphorylation site in a protein, as opposed to a protein. We will first load the table file into Cytoscape thanks to Omics Visualizer and then create the network from the data. We will finally apply some style to the network to visualize the data.

**Importing the data**

Table S3 only contains the expression in the different samples, but not the individual comparisons between EOC and healthy tissues, so we modified it to add the comparison columns 'EOC vs OSE' and 'EOC vs FTE'. The modified file is available as Underlying data\(^1\).

To load the data, we will use the specific import feature from Omics Visualizer, that can be accessed from the 'File → Import → Omics Visualizer table from File...' menu, or the 'Apps → Omics Visualizer → Import table from file' menu. This opens a custom dialog, which is very similar to the standard Cytoscape import table dialog. The user can name the table (or a default name will be given) and select the columns and their type to import. By default, all columns are imported and their type is inferred according to the first hundred lines of the file.

Here, we specifically want to import the 'UniProt' column to retrieve a STRING network afterwards, the 'AA Position' column to label the sites, the comparison columns 'EOC vs OSE' and 'EOC vs FTE' with log-ratios to be visualized, the 'Adj p-value' column to filter the table by significant sites, and the cluster assignment column 'Cluster' to visualize it. The file can be loaded with the following command:

```r
ox load file="/path/to/Francavilla2017CellRep.tsv"
```

Once the table is imported, the Omics Visualizer panel appears (Figure 1, left table). The top part of the panel has a row of icons giving access to the different features, displays the number of rows of the table, and enables access to the different Omics Visualizer imported tables. The second part of the panel is the table itself.

**Filtering the table**

It is possible to filter the table with the GUI or with the automation command. The filter GUI can be accessed via the filter icon from the ‘Omics Visualizer Tables’ tab, or with the ‘Apps → Omics Visualizer → Filter table’ menu. We filter the table so that we select only sites that have an adjusted p-value lower or equal to 0.01. This filter can be applied to the current table with the following automation command:

```r
ov filter filter="(Adj p-value,LOWER_EQUALS,0.01)"
```

After the filter has been applied, the filter icon changes colors and the number of rows before and after filtering is displayed (Figure 1, left table).

**Connecting a network**

In this use case, we will use a STRING network corresponding to our data. With the help of the stringApp, we perform a 'protein query' with the list of the UniProt identifiers of proteins from our table that have at least one phosphorylation site with an adjusted p-value lower or equal to 0.01.

The current version of the stringApp retrieves a STRING v11 network with default confidence of 0.4 that consists of 237 nodes and 1020 edges. To reduce the size of the network, we
cluster it using the Markov clustering from the clusterMaker2 app. We used the inflation value of 2.5 and the stringdb score as array sources. For illustrations purposes, we here show only the second biggest cluster, which contains 40 nodes and 107 edges.

Once the STRING network is imported in Cytoscape by the stringApp, we connect the network with the table. The link icon or the 'Apps → Omics Visualizer → Manage table connections' menu give access to the connect dialog. The dialog shows already connected networks and gives the possibility to modify or delete them. You can also create a new connection by selecting the network to connect to, then the two key columns from the network and from the table. In our case, the network was retrieved from the UniProt identifiers stored in the 'UniProt' column of our table. The stringApp stores the query into the column 'query term', so we use it as 'key column from Network' and we use 'UniProt' as the 'key column from Table'.

Figure 1 illustrates the mapping between the two tables. The "P20700" (blue box) from the node table is linked to two rows of the Omics Visualizer table which have the same value in the UniProt column.

The connection between the current network and the current table can be performed with the following automation command:

```bash
ov connect mappingColNet="query term" mappingColTable="UniProt"
```

### Retrieving a STRING network

It is possible to do the previous step more quickly by automatically retrieving a STRING network from the Omics Visualizer table. We can retrieve a STRING network with the icon or the 'Apps → Omics Visualizer → Retrieve STRING network' menu. We have to select the species, the column that contains the identifiers to query, the confidence cutoff and, if filtering was applied as in our case, whether to retrieve the identifiers only from filtered rows. By default the species is human, the cutoff is 0.40, and the query column is identified by a case-insensitive search for "uniprot" among the column names.

Once the STRING network is retrieved thanks to stringApp automation, the network is automatically connected to the Omics Visualizer table with the 'query term' column from the node table with the query column selected by the user. The network can also be retrieved from the current Omics Visualizer table with the following automation command:

```bash
ov retrieve taxonID=9606 filteredOnly=true queryColumn="UniProt"
```

### Outer visualization

The visualization is configured thanks to a specific dialog, that can be reached with the donut icon or the 'Apps → Omics Visualizer → Create donut visualization' menu.

Here, we want to visualize the two disease vs. healthy comparisons for each site. We select as Values the two comparison columns (EOC vs FTE and EOC vs OSE) that contain the numerical values we want to visualize. Because the values are log-ratios, we apply a continuous mapping and use a color scale. To be able to identify the individual phosphorylation sites in the network, we label the charts with the position of the amino acid (AA position column) of the site in the sequence of the protein. We customize the chart so that the first slice starts at a 3 o’clock angle and the label font size is changed to 15. The next dialog enables the user to configure the color scale. By default, a diverging palette is selected, and the color scales from the minimum to the maximum value and is centered on zero. The minimum and maximum bound values are directly computed from the values of the table: the maximum bound is the maximum of the absolute value of the minimum and maximum data values; the minimum bound is the opposite of the maximum bound. The user can change both the values and the colors associated. In our case, we will change the range and set it from -8 to 8. If a table value is lower than the minimum value of the scale or larger than the maximum, the value will be associated with the color associated with the minimum or maximum value, respectively. The visualization can be modified or deleted by accessing the dialog again. The resulting network can be seen in Figure 2. We have two donuts around each node representing the two comparisons. One slice of a donut is one value from the Omics Visualizer table associated with the node: in our case, it is the log-ratio of a specific site. It is possible to flip the visualization, and have as many donuts as sites, with two slices for each comparison.
The visualization of the current table can be obtained with the following automation command:

```
ov viz apply outer continuous attributes="EOC vs FTE, EOC vs OSE" labels="AA position" filteredOnly=true rangeMin=-8 rangeMax=8 chartSettings="arcstart:0,labelsize:15"
```

**Inner visualization**

If we are not interested in the individual log-ratios from the comparisons, we can instead summarize the results as a pie visualization of the expression clusters.

The inner visualization can be configured thanks to the pie icon or the ‘Apps → Omics Visualizer → Create pie visualization’ menu. The inner visualization dialog looks like the outer...
visualization dialog, and the user must first select the column that contains the values to draw. The difference here is that the user can only select one column, because the different slices of the pie come from the different values associated with the same node. In our case, we want to visualize the “Cluster” column and apply a discrete mapping. As for the outer visualization, we will label the chart slices with the amino acid position (AA position). We also customize the chart to have the first slice starting at a 3 o’clock angle and change the label font size to 15. The next dialog enables us to configure the color mapping. By default, Omics Visualizer selects a qualitative palette. Here the user may change the color associated with each value detected in the table. The default colors are too similar to each other and we changed them by clicking on the color square. The inner visualization can also be modified or deleted by accessing the dialog again.

The resulting network is shown in Figure 3 and can be obtained with the following command:

```
ov viz apply inner discrete attributes="Cluster" labels="AA position" colorMapping="A:#1F78B4,B:#FF7F00, C:#E31A1C" filteredOnly=true chartSettings="arcstart:0,labelsize:15"
```

![Figure 3](image-url)  
**Figure 3.** Same network as in Figure 2 with the clusters defined in the original study mapped to the nodes. Each slice of a pie corresponds to a significantly regulated phosphorylation site of the protein and the color of the slice represents the cluster to which the site belongs. Sites abundant in healthy (OSE and FTE), but not cancerous (EOC) tissues are in cluster A; sites abundant in FTE and EOC, but not OSE are in cluster B; and sites abundant only in EOC are in cluster C.
Legend
The user can generate a legend by clicking the map icon and choosing which legends to generate (inner and/or outer visualizations), the title, the font, and the position of the legend. Omics Visualizer will then create Cytoscape annotations corresponding to the current visualizations, displaying names of the columns above the color legends. In the case of the outer visualization, a list of column names shows the order of the columns. Because legends are annotations, they can be moved, edited, and exported with the network as an image.

The legends in Figure 2 and Figure 3 were automatically generated with the following command:

```r
ov legend draw position="EAST_TOP" title=""
```

Conclusions
Omics Visualizer is a Cytoscape app that improves how omics data can be visualized on molecular networks. A key feature of Omics Visualizer is that, unlike the standard Cytoscape node table, it enables the user to import files with several rows of data related to the same node, with each row representing a different site, isofrom, or experimental condition. This data can subsequently be visualized in or around the nodes as charts with a separate slice for each row that is connected to the node. These slices can be colored to show both continuous and discrete values, and Omics Visualizer can automatically produce a color legend to help in preparation of publication-ready figures. All the actions of Omics Visualizer can be done using the Cytoscape GUI or the commands, making it easy to use Omics Visualizer also in R or Python via the Cytoscape REST API.

Data availability
Underlying data

Data is available under the terms of the Creative Commons Attribution 4.0 International.

Software availability
The software, source code, and tutorial are available at the Cytoscape App Store: http://apps.cytoscape.org/apps/omicsvisualizer.

Archived source code at the time of publication: https://doi.org/10.5281/zenodo.3631584.

License: BSD 2-Clause “Simplified” License.

Author contributions
JHM and LJJ designed the app. ML is the main developer of the app, helped by NTD. NTD, JHM and LJJ contributed to the manuscript. ML wrote the manuscript.

Acknowledgements
The authors thank Katerina Nastou and Giulia Corsi for beta-testing the app.

References

The benefits of publishing with F1000Research:

- Your article is published within days, with no editorial bias
- You can publish traditional articles, null/negative results, case reports, data notes and more
- The peer review process is transparent and collaborative
- Your article is indexed in PubMed after passing peer review
- Dedicated customer support at every stage

For pre-submission enquiries, contact research@f1000.com