omicplotR: A tool for visualization of omic datasets as compositions

Daniel Giguere

2017-11-21

Contents

1 What is omicplotR?

`omicplotR` is an R package containing a `Shiny` app used to visually explore omic datasets, where the input is a table of read counts from high-throughput sequencing runs. It integrates the `ALDEx2`[1] package for compositional analysis of differential abundance. `omicplotR` is intended to speed up the process of visually exploring high-throughput sequencing datasets by providing an easy to use graphical user interface for users with and without experience in R.

2 Introduction

High-throughput sequencing (HTS) instruments generate an amount of reads that is constrained by limitations of the sequencing instrument itself, and do not represent the absolute number of DNA molecules in a sample. For example, an Illumina NextSeq can deliver up to 400 million single-end reads, whereas an Illumina MiSeq2 can only deliver up to 15 million single-end reads[2]. This type of data, which is constrained by an arbitrary or constant sum, is referred to as compositional data, and high-throughput sequencing data must be treated

[1] Fernandes et al (2013) PLOS ONE https://doi.org/10.1371/journal.pone.0067019
[2] https://www.illumina.com/systems/sequencing-platforms/nextseq/specifications.html

as such[3]. `omicplotR` incorporates `ALDEx2` for compositional differential abundance analysis (see `ALDEx2` for more information).

Although several R packages exist for exploring high-throughput sequencing data, they are typically command line based, which presents a barrier for users without any significant command line or scripting experience. `omicplotR` was created to facilitate the exploratory phase of high-throughput sequencing data analysis by allowing several basic exploratory plots to be generated automatically with adjustable features.

This guide provides an overview of the R package `omicplotR`, and the input requirements. A tutorial for each component of the `Shiny` app is available on my GitHub: https://github.com/dgiguer/omicplotR/wiki. omicplotR was developed for several types of HTS datasets including RNASeq, meta RNASeq, and 16s rRNA gene sequencing, and in principle, can be used for nearly any type of data generated by HTS which generates a tables of counts per feature for each sample.

3 Features

omicplotR provides a graphical user interface using the `Shiny` package for the following visualizations for HTS data:

- Principal Component Analysis (PCA) biplots
- Colouring PCA biplots by metadata
- Association plots using `igraph`
- Dendrograms
- Stacked barplots of taxonomic relative abundance
- Compositional differential abundance analysis

Additionional features include:

- Filtering count tables per sample or feature (PCA biplots)
- Filtering data into groups for PCA biplots using metadata
- Visualizing which features and samples are removed from analysis
- Colour PCA biplots using metadata (continuously, by quartile, categorical)
- Generate effect plots between conditions of associated metadata
- Interact with effect plots to visualize difference between groups
- Plot pre-calculated `ALDEx2` tables to colour points by matching search terms

4 Installation and example

Install the latest version of `omicplotR`. Make sure you have the newest version of R, `ALDEx2`, and other dependancies. `omicplotR` requires you to have at least R version 3.4.

First, load the omicplotR package. All other dependencies will be loaded automatically. This will launch the `Shiny` app in your default browser.

```
library(omicplotR)
omicplotr.run()
```

5 Input data

The tabs on the sidebar panel allow you to choose your own datafile by clicking 'Browse', or to select a provided dataset. To follow along with this vignette, please click the 'Example data' tab on the sidebar panel,

[3]Gloor et al (2017) Front. Microbiol. https://doi.org/10.3389/fmicb.2017.02224

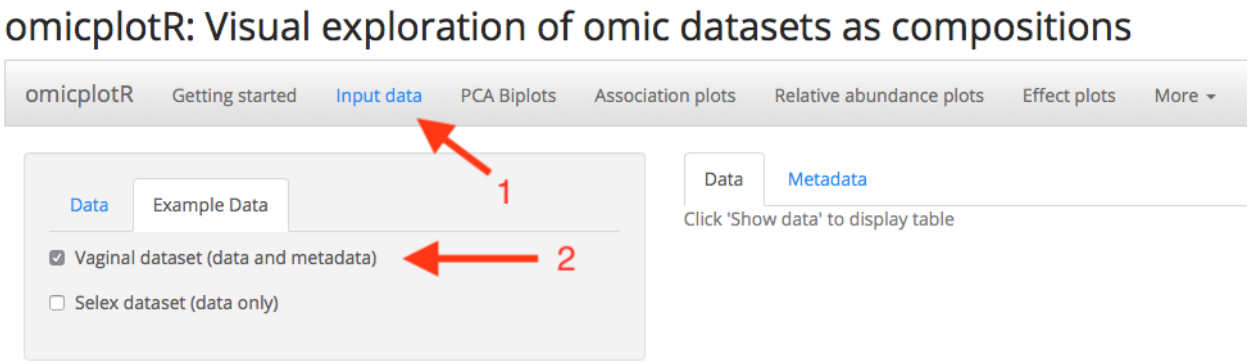

Figure 1: Screenshot of input data page. Click on 'Example data' to access the provided datasets within the Shiny app

	A	B	C	D	E	F	G
1	#OTU ID	226bbv	35bcont	261bbv	21bcont	138bvvc	taxonomy
2	0	2003	1814	654	19424	4419	Bacteria;Firmicutes;Bacilli;Lactobacillales;Lactobacillaceae;Lactobacillus;
3	1	4392	1081	269	2428	31495	Bacteria;Firmicutes;Bacilli;Lactobacillales;Lactobacillaceae;Lactobacillus;
4	2	886	6379	5286	1926	1840	Bacteria;Actinobacteria;Actinobacteria;Bifidobacteriales;Bifidobacteriace
5	3	41	48	54	402	266	Bacteria;Firmicutes;Bacilli;Lactobacillales;Lactobacillaceae;Lactobacillus;
6	4	193	130	109	83	312	Bacteria;Firmicutes;Negativicutes;Selenomonadales;Veillonellaceae;Mega
7	5	179	338	3093	221	264	Bacteria;Bacteroidetes;Bacteroidia;Bacteroidales;Prevotellaceae;Prevote
8	6	35	379	6	131	110	Bacteria;Firmicutes;Bacilli;Lactobacillales;Lactobacillaceae;Lactobacillus;
9	7	158	681	167	247	426	Bacteria;Actinobacteria;Actinobacteria;Coriobacteriales;Coriobacteriacea
10	8	221	144	211	120	357	Bacteria;Bacteroidetes;Bacteroidia;Bacteroidales;Prevotellaceae;Prevote

Figure 2: Example data. If taxonomy column introduced, it must use the column name 'taxonomy'.

and click the checkbox for the 'Vaginal dataset'. Return to the 'Data' tab on the sidebar panel to view the data and metadata by clicking 'Show data' and 'Show metadata'. The tabs on the main panel allow you to switch between displaying your data and metadata tables.

5.1 Data

For both metadata and data, each sample name and operational taxonomic unit (OTU) name **must be unique**.

1. The data file must be a **tab delimited .txt** (this is an option when you click 'Save as' from Excel).
2. The first column must contain gene/OTU identifier.
3. The first row must contain sample identifiers.
4. The last column may contain taxonomic level information, but is not required. If present, it must be labelled 'taxonomy'. **Taxonomy column must have at least four levels, separated by a semi colon.**
5. Data table must have all blank rows removed (this may require you to check in a text editor like Notepad ++ or Atom before using the app)

An example of an appropriate data file is shown in Figure 1.

Your metadata file must follow this format.

1. Must be a **tab delimited .txt** (this is an option when you click 'Save as' from Excel).
2. The first column must contain sample identifiers. The sample identifiers must be identical to the data file (not required to be in the same order).
3. The first row must contain phenotypic information.

	A	B	C	D	E	F	G
1	#SampleID	person_id	time	study	probio	age	b_contra
2	2bcont	2	0	b_cont	NA	35	y
3	3bcont	3	0	b_cont	NA	44	n
4	4bcont	4	0	b_cont	NA	33	y
5	5bcont	5	0	b_cont	NA	33	y
6	6bcont	6	0	b_cont	NA	21	y
7	7bcont	7	0	b_cont	NA	36	y
8	8bcont	8	0	b_cont	NA	18	y
9	9bcont	9	0	b_cont	NA	29	y
10	10bcont	10	0	b_cont	NA	38	y

Figure 3: Example metadata file. Metadata maybe be numerical or categorical. Any blank spaces will be replaced as NA when importing the file.

An example of an appropriate metadata file is shown in Figure 2.

5.2 Example Data

This tab on the sidebar panel provides access to two example datasets. For this vignette, follow along by clicking the checkbox next to 'Vaginal dataset'[4]. Return to the 'Data' tab to and click 'Show data' to view the data. You can view the metadata by clicking 'Show metadata' and switching the tab in the main panel to 'Metadata'.

6 PCA Biplots

The tabs within the sidebar panel allow you to choose filtering options under the 'Filtering' tab. Colouring options for a coloured PCA biplot and the ability to filter the data used for a plot by using metadata is available under the 'Colouring options' tab. The tabs within the main panel allow you to switch between displaying a non-coloured biplot under 'Biplot', a coloured biplot under 'Coloured Biplot', and a visualization of the removed samples from filtering under 'Removed data'.

6.1 Filtering

Reasonable filtering of sparse features (ie, row) from your dataset **should not** affect any conclusions made from the PCA biplots. Conclusions based on seperations or clustering driven by sparse OTUs are not reliable. Filtering is exploratory, and you should experiment to see how removing sparse data affects the structure of your dataset within the PCA biplot. The default filtering removes any feature that has a maximum value less than 10. This is the filtering we will use for the purposes of this vignette.

Under the 'Filtering' tab within the sidebar panel, the following options are available to filter your data.

1. **Minimum count per OTU**
   This filter removes any rows with a maximum sample count less than the input. Default is 10 to remove sparse OTUs.
2. **Minimum count sum per sample**
   This filter removes any columns that have a lower sum of counts than the filter. You can visualize

[4]Macklaim et al (2015) Microb. Ecol. Heal. Dis. https://doi.org/10.3402/mehd.v26.27799.

which samples are being removed when applying this filter using the 'Filtering by counts per sample' graph under the 'Removed data' tab.

3. **Minimum proportional abundance**
   This filter calculates frequency for each sample (by column) after filters 1. and 2. are applied, and removes any rows that have no frequencies above the threshold.
4. **Maximum proportional abundance**
   This filter calculates frequency for each sample, and removes any rows that have no frequencies below the threshold.
5. **Minimum count sum per OTU**
   This filter removes any rows that have a lower sum of counts than the filter. You can visualize which rows are removed with the 'Filtering by counts per row' graph under the 'Removed data' tab.

Variance cutoff

This filter calculates the variance for each OTU (row), and removes any rows that have lower variance than the filter. Variance is calculated after all other filters, and after the reads have been transformed by the centre-log ratio.

Adjust scale

This changes the scale of the biplot. When set to zero, it shows the relationship between samples (columns), while being set to 1 shows the relationship between OTUs (rows). Data is not filtered from this operation.

6.2 Colouring options

To view the coloured PCA biplot, click the 'Coloured Biplot' tab within the main panel. The 'Colouring options' panel within the side bar panel allows you to choose which metadata to colour your plot by. Choose the column 'study' to view the data coloured by study. You will need to click the 'nonnumeric metadata' radio button since the values of the column are categorical. The sample names (ie, 128bvvc) that are black on the PCA biplot represent samples without metadata. We can easily filter the dataset to replot only the samples from the 'b_bv' study by clicking on 'Filter by metadata values' under the 'Colouring options' tab of the side bar panel. In the pop up, select the 'study' column, and input 'b_bv' into the text field for Value 1. Click 'Update Biplot' to update the biplot, or 'Reset Filter' to reset the filter.

6.3 Filter data table by metadata

Click the 'Filter by metadata values' to replot samples according the groups in the metadata. For example, we can filter the dataset to only plot the samples that are either positive or negative for bacterial vaginosis according to nugest status by inputting 'bv' or 'n' for the column 'n_status' from the data table. This allows the user to explore virtually all variables in their dataset.

We will continue using the example dataset to generate effect plots using `ALDEx2`

7 Generating ALDEx2 effect plots

`ALDEx2` is an R package for analyze differential abundance of HTS data as compositions between two groups.

To generate effect plots, click on the 'Effect plots' tab. Here, you can choose to calculate the `ALDEx2` manually (ie, choosing which columns to compare) or by using metadata. For this example, we will compare the nugent status of the samples.

Select the 'n_status' column from metadata. For Group 1 choose 'n', and for Group 2 choose 'bv'. After these have been selected, click the 'Generate effect plot' button. For large datasets, this will take a long

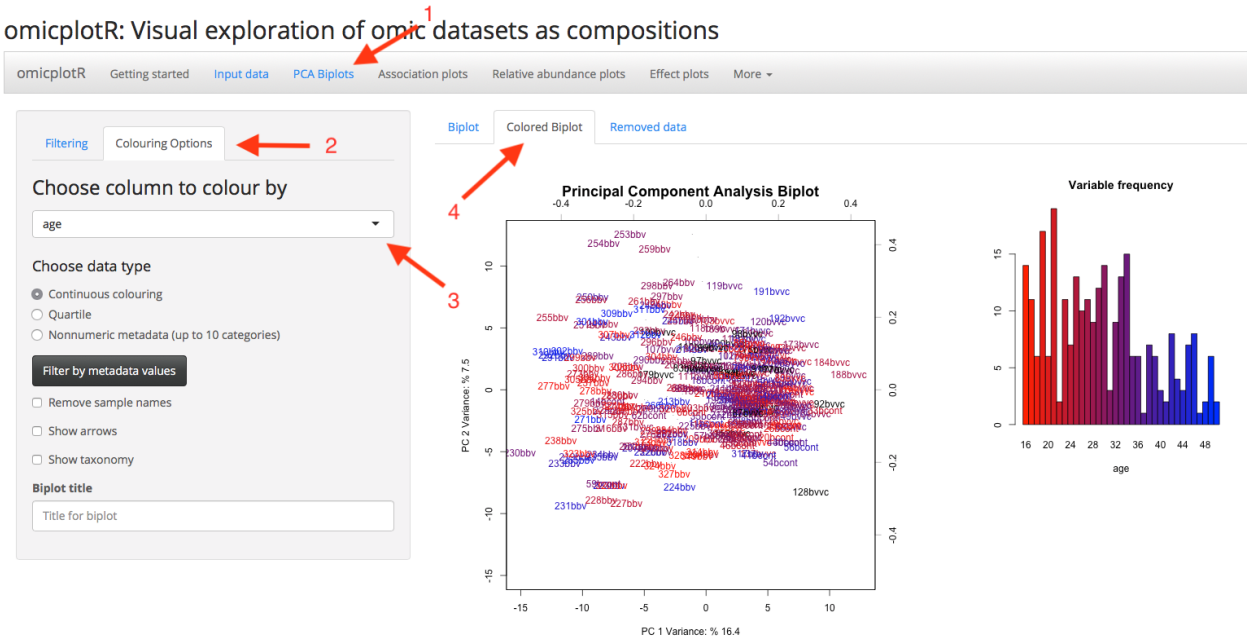

Figure 4: Screen of coloured principal component analysis biplot. Steps to recreate this plot are lablled 1-4.

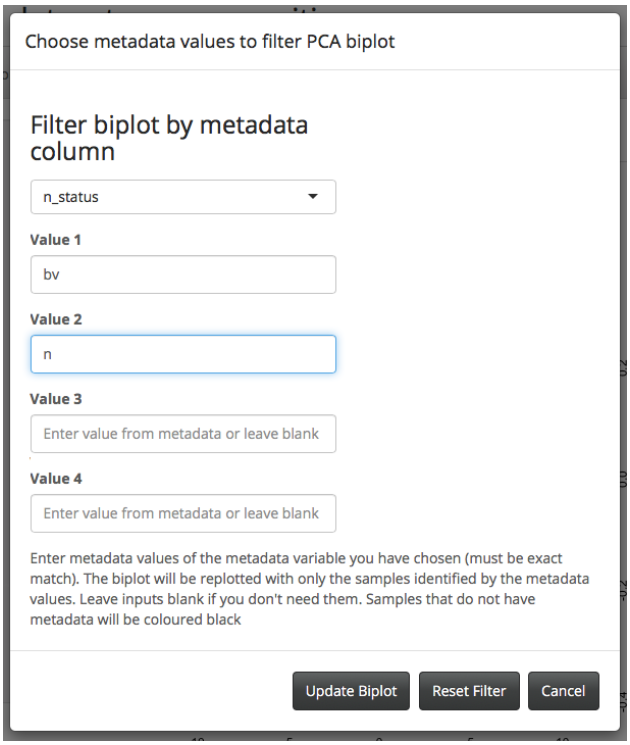

Figure 5: Pop-up generated from click 'Filter by metadata values'. Select a column an enter values from the column to re-plot only those values. They must match exactly. Filter can be reset or updated.

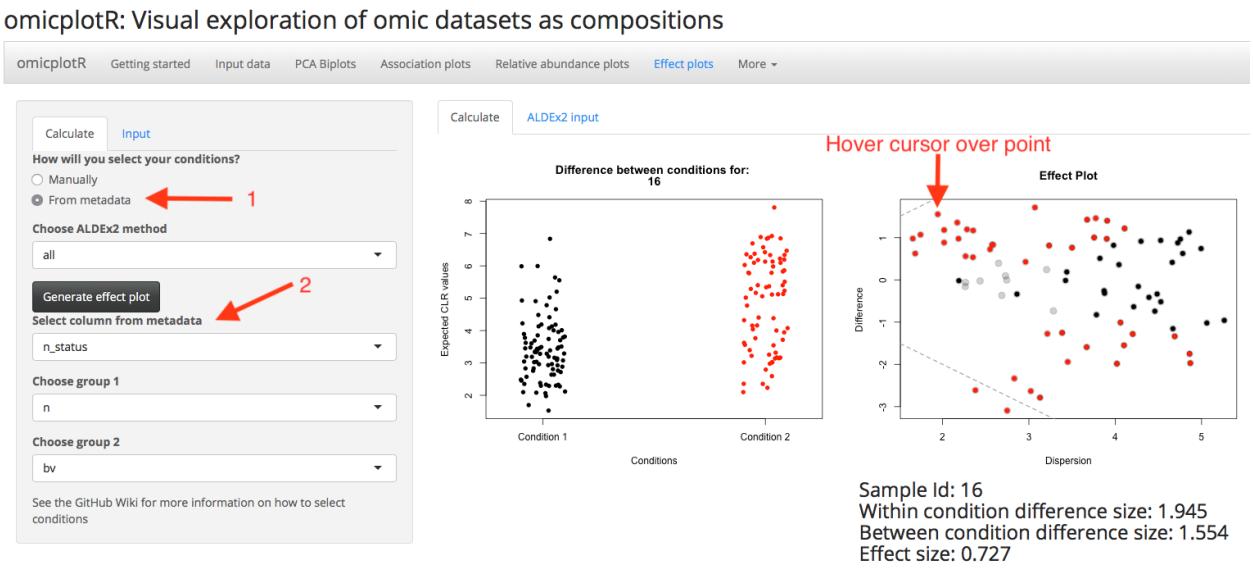

Figure 6: Screenshot of effect plots. Hovering over a feautre's point in the effect plot generates a stripchart to compare the relative abundances calculated by ALDEx2 for each sample.

time. It takes about 10 seconds to calculate for the example data, which has 77 features for 297 samples. By hovering over a point on the graph, a strip chart of the expected CLR abundance (see `ALDEx2`) for each sample will be shown, allowing you to compare the differences between your samples for a given OTU. It also displays information from the effect table generated by `ALDEx2`

If you are using your own data and want to select groups by columns, you will need to reorder your file to have the first n columns as group 1, and the last n columns as group 2.

8 Contributors

Daniel J Giguere wrote the original omicplotR code and designed the `Shiny` app, with help from Brandon Lieng, Jean Macklaim, and Greg Gloor. Brandon Lieng wrote the function `clr.strip()` needed for the strip charts. Jean Macklaim conceptualized this project, contributed numerous ideas for the design, and wrote the original code for the taxonomic distribution and dendrograms. Both Greg and Jean played roles in designing and implementing omicplotR.

9 Version information

Currently version 0.99.0.

For more information about how to use `omicplotR`, please visit the wiki on my Github page:

https://github.com/dgiguer/omicplotR/wiki