The Journal of Open Source Software

ggoncoplot: an R package for visualising somatic mutation data from cancer patient cohorts

Sam El-Kamand [1], **Julian M. W. Quinn** [1], **and Mark J. Cowley** [1,2¶]

1 Children's Cancer Institute, Australia **2** School of Clinical Medicine, UNSW Medicine & Health, Australia ¶ Corresponding author

DOI: 10.xxxxxx/draft

Software

- Review
- Repository
- Archive

Editor: Open Journals

Reviewers:

- @openjournals

Submitted: 01 January 1970
Published: unpublished

License
Authors of papers retain copyright and release the work under a Creative Commons Attribution 4.0 International License (CC BY 4.0).

Summary

The ggoncoplot R package generates interactive oncoplots (also called oncoprints) to visualize mutational patterns across patient cancer cohorts (Figure 1). Oncoplots reveal patterns of gene co-mutation and include marginal plots that indicate co-occurrence of gene mutations and tumour features. It is useful to relate gene mutation patterns seen in an oncoplot to patterns seen in other plot types, including gene expression t-SNE plots or methylation UMAPs. There are, however, no existing oncoplot-generating R packages that support dynamic data linkage between different plots. To addresses this gap and enable rapid exploration of a variety of data types we constructed the ggoncoplot package for the production of oncoplots that are easily integrated with custom visualisations and that support synchronised data-selections across plots (Figure 2). ggoncoplot is available on GitHub at https://github.com/selkamand/ggoncoplot.

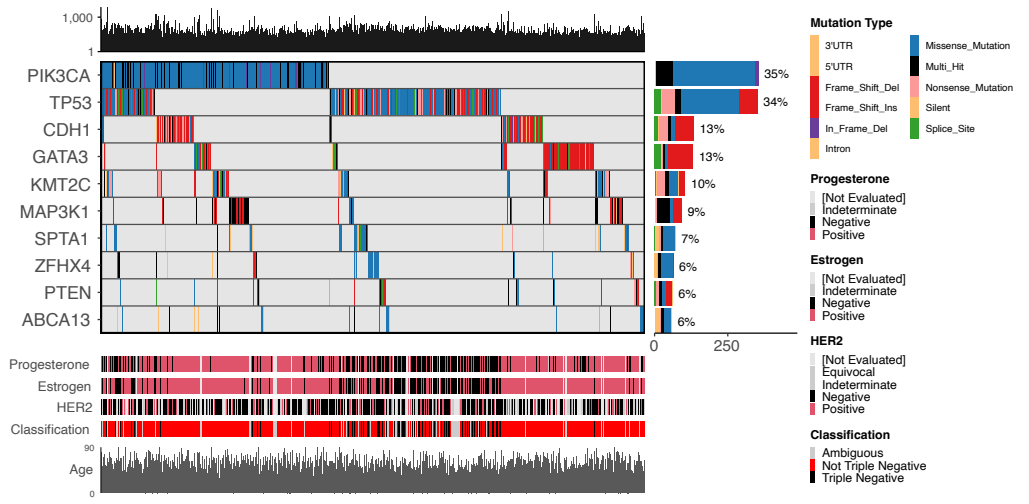

Figure 1: ggoncoplot output visualising mutational trends in the TCGA breast carcinoma cohort. Individual patient samples are plotted on the x-axis, hierarchically sorted so that samples with the most frequent gene mutations appear on the leftmost side. The plot indicates that PIK3CA is the most frequently mutated gene, followed by TP53. Marginal plots indicate the total number of mutations per sample (top), and the number of samples showing mutations in each gene, coloured by mutation type (right). A range of clinical features, including progesterone and estrogen receptor status are shown on the marginal plot at the bottom. A detailed description of the ggoncoplot sorting algorithm is available here

DRAFT

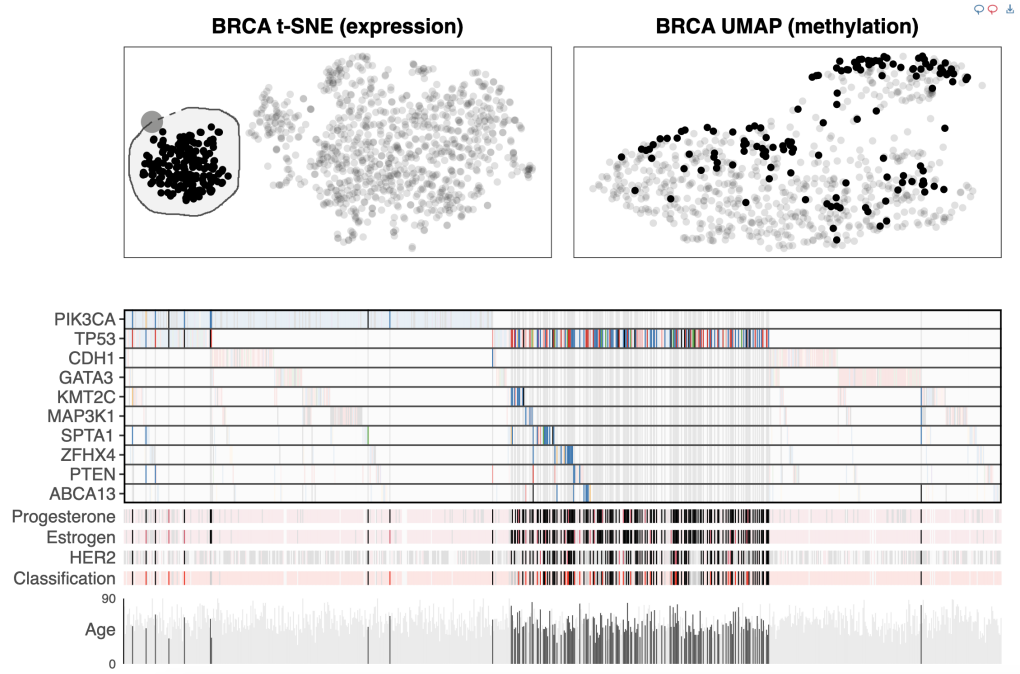

Figure 2: Example of the ggoncoplot shown in Figure 1, where the oncoplot has been dynamically cross-linked to a gene expression t-SNE plot (top left) and a methylation UMAP (top right). Here, the lasso tool (see Figure 3) was used to select a cluster of gene expression data points (i.e., individual samples) in the t-SNE plot. Selected samples were automatically highlighted on the UMAP and oncoplot. This reveals that samples which cluster on the left of the t-SNE plot also cluster in the oncoplot, chiefly containing mutations in TP53 but wild type PIK3CA. The plots of progesterone, estrogen, HER2 status and triple negative classification show that the samples selected in the t-SNE are virtually all triple negative breast cancers. In contrast to the oncoplot, the methylation UMAP shows no strong clustering, in line with knowledge of methylation patterns in triple negative breast cancer.

DRAFT

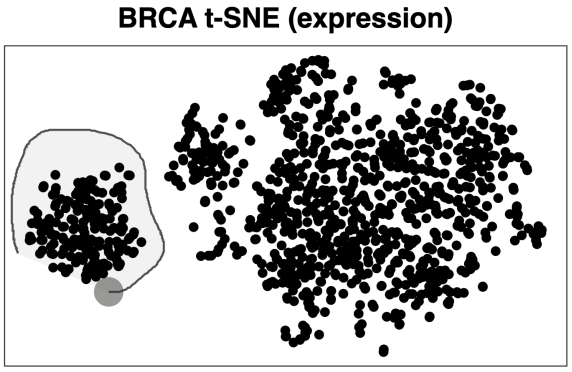

Figure 3: Example image of the t-SNE plot where the lasso tool is being used to manually delineate a data point cluster on the left side.

Statement of Need

Oncoplots are highly effectively for visualising mutation data in cancer cohorts but are challenging to generate with the major R plotting systems (base, lattice, or ggplot2) due to their algorithmic and graphical complexity. Simplifying the process would make oncoplots more accessible to researchers. Packages like ComplexHeatmap (Gu, 2022), maftools (Mayakonda

et al., 2018), and genVisR (Skidmore et al., 2016) all make static oncoplots easier to create, but there is still a significant unmet need for an easy method of creating oncoplots with the following features:

- **Interactive plots**: Customizable tooltips, cross-selection of samples across different plots, and auto-copying of sample identifiers on click.
- **Support for tidy datasets**: Compatibility with tidy, tabular mutation-level formats (MAF files or relational databases), typical of cancer cohort datasets.
- **Auto colouring**: Automatic selection of color palettes for datasets where consequence annotations are aligned with standard variant effect dictionaries (PAVE, SO, or MAF).
- **Versatility**: The ability to visualize entities other than gene mutations, including non-coding features (e.g., enhancers) and non-genomic entities (e.g., microbial presence in microbiome datasets).

We developed ggoncoplot as the first R package that addresses all these challenges simultaneously (Figure 4). Examples of all key features are available in the ggoncoplot manual.

Property	complexheatmap	maftools	GenVisR	ggoncoplot
Sample sorting algorithm	memo sort	heirarchical sort	heirarchical sort	heirarchical sort
Plotting framework	BaseR	BaseR	ggplot2	ggplot2
Automatic rendering of clinical annotations as bar or tile plots based on datatype	No	No	No	**Yes**
Works on tabular, tidy, long-form input data as would be stored in large databases	No	**Yes**	**Yes**	**Yes**
Interactive	Yes[1]	No	No	**Yes**
Customisable tooltips	No	No	No	**Yes**
Allows any mutation dictionary to be used	**Yes**	No	**Yes**	**Yes**
Automatic colour palette selection when mutation impact dictionary conforms to known ontologies	No	Yes (MAF only)	No	**Yes (MAF, SO, or PAVE)**
Approach for resolving genes with multiple mutations	Different Visualisation on Plot[2]	**Flags as Multi-Hit**	Picks more severe consequence or leaves to user[3]	**Flags as Multi-Hit**
Supports a mutation level dataset as input	No[4]	**Yes**	**Yes**	**Yes**
Native support for faceting by pathway	No	**Yes**	No	**Yes**
Supports marginal plots describing TMB, gene mutation recurrence, and clinical annotations	**Yes**	**Yes**	**Yes**	**Yes**

Figure 4: Comparison of R packages for creating oncoplots. [1]Requires the shiny and interactiveComplexHeatmap packages. [2]Requires the user to first summarise mutations at the gene level and format as a sample by gene matrix with mutations separated by semicolons (wide format). [3]For MAF inputs the most severe consequence is chosen, however for non-MAF datasets users must manually define the mutation impact hierarchy. [4]Non-unique mutation types are treated as one observation, however if different mutation types affect one gene, the indiviual mutations can be plotted with different shapes or sizes in a user-configured manner.

Acknowledgements

We thank the developers of the packages integral to ggoncoplot, especially David Gohel for ggiraph (Gohel & Skintzos, 2024), which enables its interactivity, and Thomas Lin Pedersen for patchwork (Pedersen, 2024) and ggplot2 maintenance. We also acknowledge Hadley Wickham and all contributors to ggplot2 (Wickham, 2016). Additionally, we thank Dr. Marion Mateos for her insightful feedback during the development of ggoncoplot.

References

Gohel, D., & Skintzos, P. (2024). *Ggiraph: Make 'ggplot2' graphics interactive*. https://davidgohel.github.io/ggiraph/

Gu, Z. (2022). Complex heatmap visualization. *iMeta*, *1*(3), e43. https://doi.org/10.1002/imt2.43

Mayakonda, A., Lin, D.-C., Assenov, Y., Plass, C., & Koeffler, H. P. (2018). Maftools: Efficient and comprehensive analysis of somatic variants in cancer. *Genome Research*, *28*, 1747–1756. https://doi.org/10.1101/gr.239244.118

Pedersen, T. L. (2024). *Patchwork: The composer of plots*. https://patchwork.data-imaginist.com

Skidmore, Z. L., Wagner, A. H., Lesurf, R., Campbell, K. M., Kunisaki, J., Griffith, O. L., & Griffith, M. (2016). GenVisR: Genomic visualizations in r. *Bioinformatics*, *32*, 3012–3014. https://doi.org/10.1093/bioinformatics/btw325

Wickham, H. (2016). *ggplot2: Elegant graphics for data analysis*. Springer-Verlag New York. ISBN: 978-3-319-24277-4

DRAFT