BioVenn – an R and Python package for the comparison and visualization of biological lists using area-proportional Venn diagrams

Tim HULSEN[a,1]

[a]Department of Professional Health Solutions & Services, Philips Research, Eindhoven, The Netherlands

[1]Corresponding author. E-mail: tim.hulsen@philips.com. ORCID: 0000-0002-0208-8443

Abstract. One of the most popular methods to visualize the overlap and differences between data sets is the Venn diagram. Venn diagrams are especially useful when they are 'area-proportional' i.e. the sizes of the circles and the overlaps correspond to the sizes of the data sets. In 2007, the BioVenn web interface was launched, which is being used by many researchers. However, this web implementation requires users to copy and paste (or upload) lists of IDs into the web browser, which is not always convenient and makes it difficult for researchers to create Venn diagrams 'in batch', or to automatically update the diagram when the source data changes. This is only possible by using software such as R or Python. This paper describes the BioVenn R and Python packages, which are very easy-to-use packages that can generate accurate area-proportional Venn diagrams of two or three circles directly from lists of (biological) IDs. The only required input is two or three lists of IDs. Optional parameters include the main title, the subtitle, the printing of absolute numbers or percentages within the diagram, colors and fonts. The function can show the diagram on the screen, or it can write output to one of the supported file formats. The function also returns all thirteen lists. The BioVenn R package and Python package were created for biological IDs, but it can be used for other IDs as well. Finally, BioVenn can map Affymetrix and EntrezGene to Ensembl IDs. The BioVenn R package is available in the CRAN repository, and can be installed by running 'install.packages("BioVenn")'. The BioVenn Python package is available in the PyPI repository, and can be installed by running 'pip install BioVenn'. The BioVenn web interface remains available at https://www.biovenn.nl.

Keywords: Bioinformatics, Visualization, Venn diagram, Combinatorics, Set theory, Genomics, Data science, R, Python

1. Introduction

In many 'big data' projects, it can be very useful to see the overlap between different data sets, in terms of patient IDs, gene names, etc. One of the most popular methods to visualize the overlap between data sets is the Venn diagram: a diagram consisting of two or more circles in which each circle corresponds to a data set, and the overlap between the circles corresponds to the overlap between these data sets. Venn diagrams are especially useful when they are 'area-proportional' i.e. the sizes of the circles and the overlaps correspond to the sizes of the data sets. Some web-based tools were created that can create area-proportional Venn diagrams, such as the (deprecated) tools VennMaster [1] and DrawEuler [2]. In 2003, the website Venndiagram.tk [3] was launched, followed in 2007 by the BioVenn web interface [4], which has been used to create publication figures by many researchers [5], and is still available at this moment. However, the BioVenn web application requires users to copy and paste lists of IDs (or upload files with lists of IDs) into the web browser, which is not always convenient and makes it difficult for researchers to create Venn diagrams 'in batch'. Moreover, when the source data changes, it needs to be copy-and-pasted again into the web interface. Using programming languages, it is possible to do batch processing and to quickly rerun a script when the source data has changed. Two of the most popular programming languages used within many scientific fields are R and Python. There are some R and Python packages available that can create Venn diagrams, which are listed in the following paragraphs.

2. Existing R packages

2.1. colorfulVennPlot

The first package is 'colorfulVennPlot' [6]. This package can create 2-circle and 3-circle Venn diagrams, and use ellipses for diagrams of 4 sets. Only the 2-circle diagrams can be made area-proportional, but the user needs to calculate the circles' sizes and overlap by using the separate 'resizeCircles' function.

2.2. eulerr

A second package is 'eulerr' [7], which can generate area-proportional Euler diagrams. A Euler diagram is a generalization of a Venn diagram, relaxing the criterion that all interactions need to be represented. In practice, both terms are used interchangeably. This package uses both ellipses and circles.

2.3. nVennR

A third package is 'nVennR' [8]. This package can create "quasi-proportional Venn and Euler diagrams" for an unlimited number of sets. For a large number of sets, the algorithm might be very slow, because it needs to run many simulation cycles. Because of the resulting complicated shapes, the diagrams might not be easy to read.

2.4. venn

A fourth package is 'venn' [9], which can generate Venn diagrams up to 7 sets, but not in an area-proportional manner. For more than three sets, it uses pre-set polygon shapes.

2.5. VennDiagram

The most popular package at this moment is 'VennDiagram' [10]. This package can generate Venn and Euler diagrams of up to five sets, but these are not area-proportional, unless the user calculates the radii and distances between the circles by himself, and passes these numbers through to one of the draw.*.venn functions.

2.6. venneuler

A sixth package is 'venneuler' [11]. This package can create area-proportional Venn diagrams as well, if the sizes of the overlaps are passed to its venneuler function. The returned object also gives some mathematical information such as the residuals (percentage difference between input intersection area and fitted inter-section area) and stress values.

2.7. vennplot

The seventh and final package is 'vennplot' [12]. It can create area-proportional Venn diagrams in 2D or 3D, with two or three circles or balls. The 3D functionality is interesting (the diagram can be rotated), but the mathematics behind it is actually the same as for the 2D plot.

3. Existing Python packages

3.1. matplotlib-venn

The most popular package at this moment is 'matplotlib-venn' [13]. Its 'venn2' and 'venn3' functions can create area-proportional Venn diagrams of two and three circles, respectively. However, they don't offer the ID mapping functionality of BioVenn, and the 'drag-and-drop' functionality of text and numbers in the SVG mode of BioVenn is missing as well.

3.2. PyVenn

A second package is 'PyVenn' [14]. This package offers plotting of Venn diagrams of two to six circles, but these are not area-proportional like in BioVenn or Matplotlib-Venn: the shapes are always the same.

4. Methods

The PHP script that forms the basis for the BioVenn web interface, was rewritten in the R and Python languages. The only function in the package is "draw.venn" (R) or "draw_venn" (Python), and it follows these steps:

1. Remove duplicate IDs (note: BioVenn is case-sensitive)
2. Map EntrezGene and Affymetrix IDs to Ensembl IDs (using Biomart [15])
3. Generate lists of the thirteen possible sets, and count them
4. Calculate the radii of the circles so that the areas of the circles correspond to the size of the datasets they represent
5. Calculate the distances between the centers of the circles, so that the areas of the two-circle overlaps correspond to the size of the datasets they represent (see figure 1 of [4])

6. Calculate the angles of the XYZ triangle
7. Calculate the centers of the circles
8. Calculate the intersection points of the circles
9. Calculate the points where the numbers will be printed
10. Set output type to file or screen (depending on the output parameter)
11. Print the title and subtitle
12. Print the circles with the calculated centers and radii
13. Print the absolute numbers/percentages
14. Print the texts for the three circles
15. Write to the selected output type and filename
16. If SVG is selected as output type, do some post-processing in order to create the drag-and-drop functionality of the texts
17. Return the contents of the thirteen lists: X, Y, Z, X only, Y only, Z only, XY, XZ, YZ, XY only, XZ only, YZ only and XYZ.

Whereas the BioVenn web interface only supports PNG and SVG as output formats, the Python package also supports JPEG, PDF and TIFF. The R package even supports all of these file formats and BMP.

5. Results

The BioVenn R/Python package can generate area-proportional Venn diagrams of two or three circles from lists of (biological) identifiers. It is a lightweight package, depending on only a small number of other packages, making it more likely that the package will still work in the future. The only function in the first version is the 'draw.venn' function (in R; 'draw_venn' in Python), for which the only required input is two or three lists of identifiers. Optional parameters include the main title, the subtitle, the printing of absolute numbers or percentages within the diagram, colors and fonts. The function can show the diagram on the screen, or it can write output to one of the supported file formats. The SVG mode also supports drag-and-drop of texts and numbers. The function also returns the contents of all possible lists. The BioVenn R/Python package was created for biological identifiers, but it can be used for other identifiers as well. Finally, BioVenn can map Affymetrix and EntrezGene IDs to Ensembl IDs.

5.1. R example

The following very simple R code creates the example plot of figure 1, and returns the data of table 1:

```
list_x <- c("1007_s_at", "1053_at", "117_at", "121_at", "1255_g_at", "1294_at")
list_y <- c("1255_g_at", "1294_at", "1316_at", "1320_at", "1405_i_at")
list_z <- c("1007_s_at", "1405_i_at", "1255_g_at", "1431_at", "1438_at", "1487_at", "1494_f_at")
biovenn <- draw.venn(list_x, list_y, list_z, subtitle="Example diagram", nrtype="abs")
```

5.2. Python example

The Python code works in a very similar manner:

```
list_x = ("1007_s_at", "1053_at", "117_at", "121_at", "1255_g_at", "1294_at")
list_y = ("1255_g_at", "1294_at", "1316_at", "1320_at", "1405_i_at")
```

```
list_z = ("1007_s_at", "1405_i_at", "1255_g_at", "1431_at", "1438_at", "1487_at", "1494_f_at")
biovenn = draw_venn(list_x, list_y, list_z, subtitle="Example diagram", nrtype="abs")
```

Note that the code in both R and Python could be even compressed into one line, by adding the lists directly to the draw_venn command. For improved readability we use a four-line code.

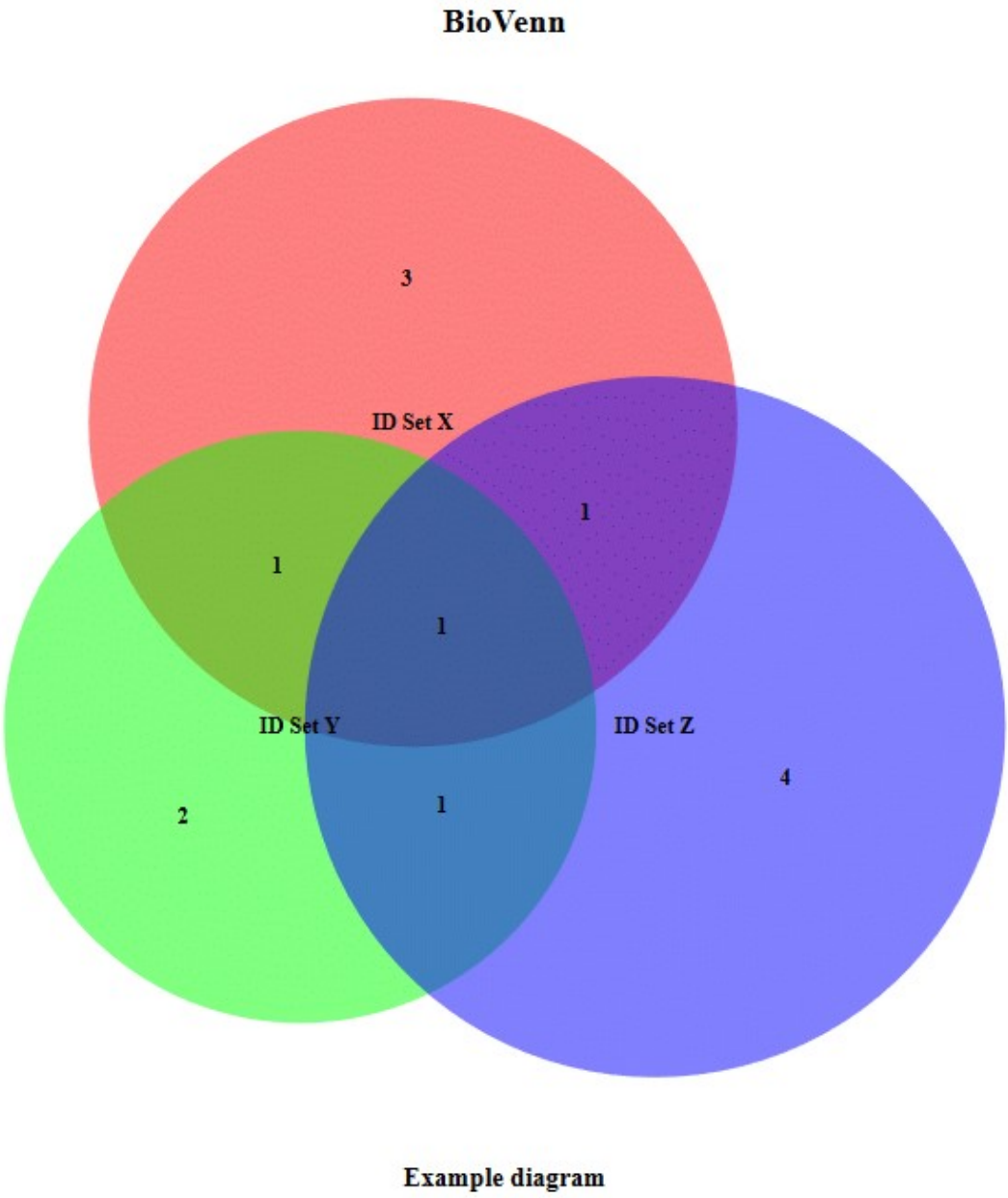


Figure 1. Example BioVenn diagram. This example was created by just entering three lists of IDs and setting three other parameters (title, subtitle and nrtype).

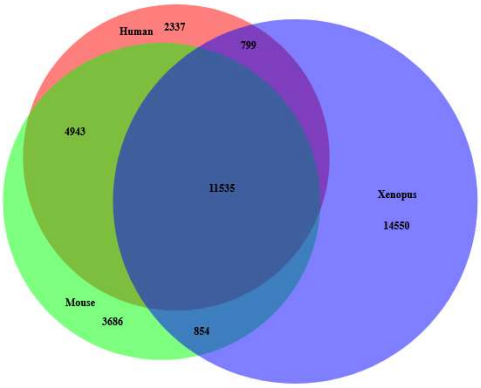
Variable	Data type	Contents
$x	character [6]	1007_s_at, 1053_at, 117_at, 121_at, 1255_g_at, 1294_at
$y	character [5]	1255_g_at, 1294_at, 1316_at, 1320_at, 1405_i_at
$z	character [7]	1007_s_at, 1405_i_at, 1255_g_at, 1431_at, 1438_at, 1487_at, 1494_f_at
$x_only	character [3]	1053_at, 117_at, 121_at
$y_only	character [2]	1316_at, 1320_at
$z_only	character [4]	1431_at, 1438_at, 1487_at, 1494_f_at
$xy	character [2]	1255_g_at, 1294_at
$xz	character [2]	1007_s_at, 1255_g_at
$yz	character [2]	1255_g_at, 1405_i_at
$xy_only	character [1]	1294_at
$xz_only	character [1]	1007_s_at
$yz_only	character [1]	1405_i_at
$xyz	character [1]	1255_g_at

Table 1. Example output. This example was created by just entering three lists of IDs and setting two other parameters (subtitle and nrtype).

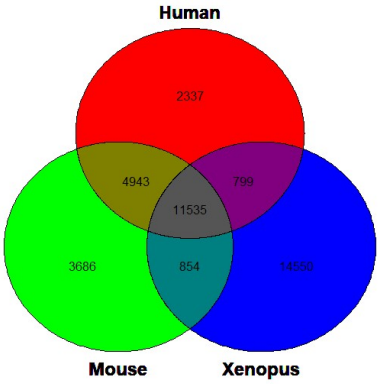
5.3. Overall comparison

To compare the different R and Python packages, we created Venn diagrams of a dataset showing orthologous genes that are present in human (Homo sapiens), mouse (Mus musculus) or the African clawed frog (Xenopus laevis) (available at the OMA Browser [16] through https://omabrowser.org/All/oma-groups.txt.gz). Since human and mouse are more closely related than human and Xenopus (and mouse and Xenopus), we expect that the circles of human and mouse have a larger overlap. Furthermore, Xenopus has more genes, so its circle should be larger than the circles of human and mouse.

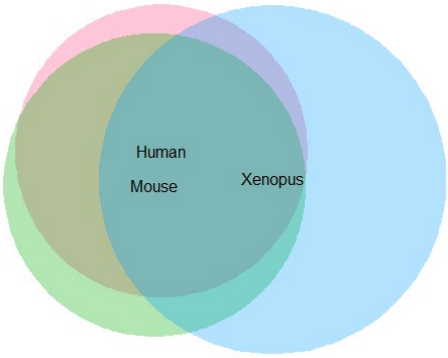
Figure 2 shows the Venn Diagrams created in each of the R packages, in alphabetical order. For each of the plots, the colours red, green and blue were used, titles were removed, and numbers were printed in the diagram (if that option was available). The code used to generate the plots can be viewed at https://www.biovenn.nl/r_python/. We can see that the packages that create area-proportional diagrams (a, c, d, g, h) give a better impression of what the data looks like: the human and mouse circles indeed have a larger overlap than with the Xenopus circle, and the Xenopus circle is larger than the other ones. The nVennR diagram (d) might be visually less appealing, but it displays the information correctly as well. The non-area-proportional diagrams (b, e, f) need some careful reading of the numbers in the figure before they can be interpreted.



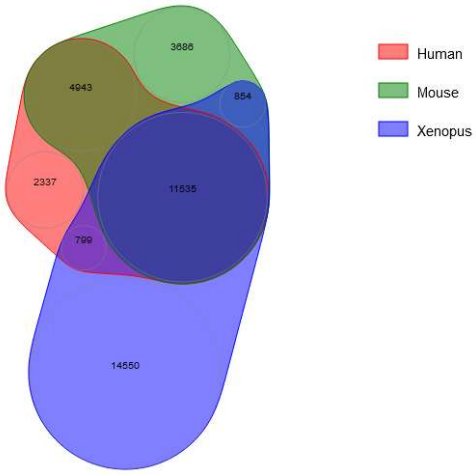
(a)



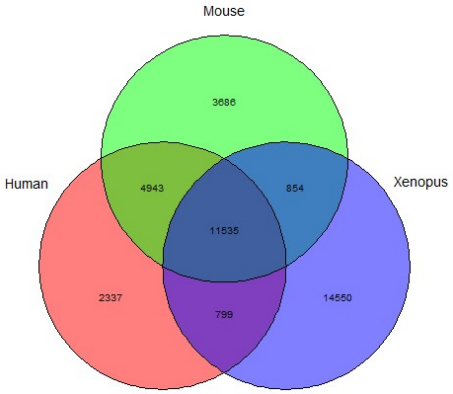
(b)



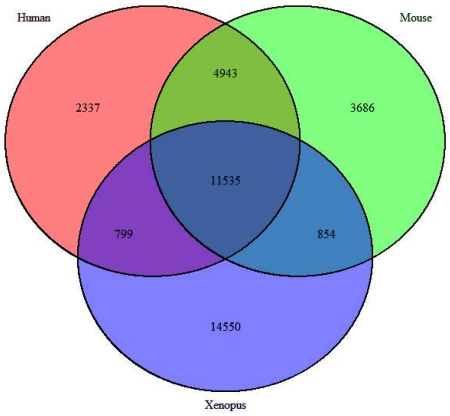
(c)



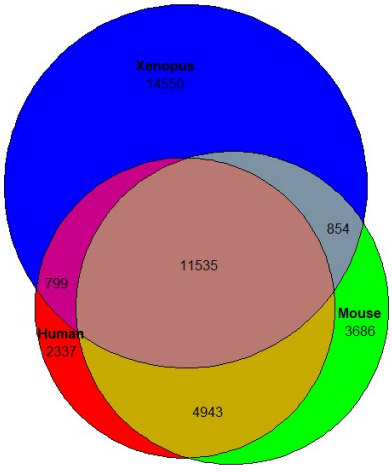
(d)



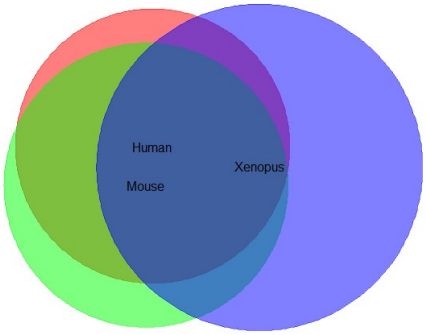
(e)



(f)



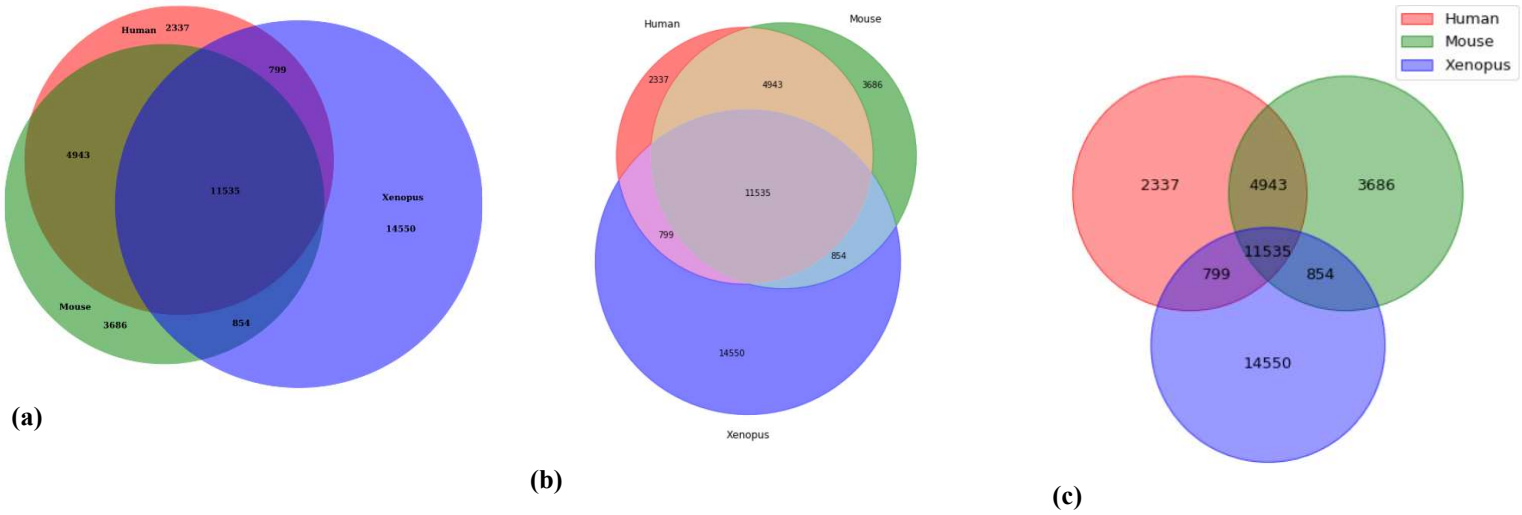
(g)



(h)

Figure 2. Venn diagrams created by each of the R packages: **a)** BioVenn, **b)** colorfulVennPlot, **c)** eulerr, **d)** nVennR, **e)** venn, **f)** VennDiagram, **g)** venneuler and **h)** vennplot.

Figure 3 shows the Venn Diagrams created in each of the Python packages, in alphabetical order, with the same method as described above. Again, the area-proportional diagrams (a, b) can be understood much more easily than the non-area-proportional diagram (c).



(a) **(b)** **(c)**

Figure 3. Venn diagrams created by each of the Python packages: **a)** BioVenn, **b)** matplotlib-venn and **c)** PyVenn.

Package name	**BioVenn**	**colorfulVennPlot**	**eulerr**	**matplotlib-Venn**	**nVennR**	**PyVenn**	**venn**	**vennDiagram**	**venneuler**	**vennplot**
Programming language	R, Python (and web)	R	R (and web)	Python	R	Python	R	R (and Cytoscape and web)	R	R
Max. number of sets	3	4 (>3 uses ellipses)	Unlimited (in theory)	3	Unlimited (in theory)	6	7	5	Unlimited (in theory)	3
Area proportionality	Automatically	Manually (only for 2-circle diagrams)	Automatically	Automatically	Automatically	No	No	Manually	Manually	Automatically
Built-in biological ID mapping	Yes	No	No	No	No	No	No	No	No	No
Input format	Sets of IDs	Sets of IDs, numbers	Sets of IDs, numbers	Sets of IDs, numbers	Sets of ID, numbers	Sets of IDs, numbers	Numbers	Sets of IDs, numbers	Sets of IDs, numbers	Sets of IDs, numbers
Output format	BMP (only in R), JPEG, PDF, PNG,	R graphics	R graphics	Python graphics	SVG, R graphics	Python graphics	R graphics	R graphics, TIFF	R graphics	R graphics

	SVG, TIFF, R/Python graphics									
Drag-and-drop of texts, nrs	Only in SVG mode	No	No	No	No	No	No	No	No	No
Shapes used	Circles	Circles/ Ellipses	Circles/ Ellipses	Circles	Polygons	Circles/ Ellipses/ Polygons	Circles/ Ellipses/ Polygons	Circles/ Ellipses	Circles	Circles/ Balls
Print absolute numbers / percentages	Both	Only absolute numbers	Both	Only absolute numbers	Only absolute numbers	Both	Only absolute numbers	Both	No	No
Set title(s)	Title and subtitle	Only title	Only title	No	No	No	No	Title and subtitle	No	No
Set circle colors	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Set circle texts	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Set background color	Yes	No	No	No	No	No	No	No	No	No
Set text colors	Yes	No	No	No	No	No	No	Yes	No	No
Set text fonts (family, face, size)	Yes	No	No	No	Only font size	Only font size	Only font size	Yes	No	No

Table 2. Venn diagram package comparison. All currently available R and Python packages that can generate Venn diagrams compared. Note that this table only lists built-in functionality; some functionality such as plotting to certain file formats might be possible by using other R or Python functions.

Table 2 shows a comparison of all features of BioVenn and the seven other packages mentioned above. BioVenn is the only package that is available in both R and Python (as well as a web interface). There are packages that can generate Venn diagrams from more than three sets, but these are either not area-proportional or inaccurate. Only BioVenn has built-in biological ID mapping functionality, which earns it the prefix 'bio'. Some programs support not only the input of IDs, but also the numbers of the sets and their overlaps. In BioVenn, these are automatically calculated from the ID lists. This also makes sure that the user cannot mathematically impossible numbers (e.g. overlaps larger than the sets themselves). BioVenn supports a large number of output formats as well. It needs to be noted here that this table only lists built-in functionality; some functionality such as plotting to certain file formats might be possible by using other R or Python functions (e.g. the 'matplotlib.pyplot' functions in Python). BioVenn is the only package that supports drag-and-drop of the texts and numbers (in SVG mode), which can be a very useful functionality when a set or overlap is very small compared to the rest of the figure, or when the circle title (e.g. 'Set X', 'Set Y', 'Set Z') overlaps with a number. BioVenn uses only circles,

whereas other packages also use ellipses, polygons or even 3D balls. There are four packages (BioVenn, eulerr, PyVenn and VennDiagram) that are able to print absolute numbers or percentages in the diagram. Finally, BioVenn offers the most flexibility in formatting: title, subtitle and circle texts can be changed (as well as their fonts and colors), and the background color and the circle colors can be set.

6. Conclusion

The BioVenn R and Python packages are a useful addition to the existing web interface, and they have some unique advantages over existing packages that can create Venn diagrams, such as the mapping of biological IDs and the drag-and-drop functionality in SVG mode. Other useful functions are the area-proportionality, printing absolute numbers or percentages, and the possibility to change all colors (including text and background) and fonts. The BioVenn R package is available in the CRAN repository [17], and can be installed by running 'install.packages("BioVenn")'. The Python package is available in the PyPI repository [18], and can be installed by running 'pip install BioVenn'. The BioVenn web interface remains available at https://www.biovenn.nl.

7. Acknowledgements

The author would like to thank the numerous people who have sent their suggestions for improvements over the past years, which have resulted in a more precise web tool (and now also an R package as well as a Python package).

8. Competing interest statement

Dr. Hulsen is employed by Philips Research.

9. References

[1] H.A. Kestler, A. Muller, J.M. Kraus, M. Buchholz, T.M. Gress, H. Liu, D.W. Kane, B.R. Zeeberg, and J.N. Weinstein, VennMaster: area-proportional Euler diagrams for functional GO analysis of microarrays, *BMC Bioinformatics* **9** (2008), 67. PubMed ID: 18230172. https://www.ncbi.nlm.nih.gov/pubmed/18230172.

[2] G. Stapleton, Z. Leishi, J. Howse, and P. Rodgers, Drawing Euler Diagrams with Circles: The Theory of Piercings, *IEEE Trans Vis Comput Graph* **17** (2011), 1020-1032. PubMed ID: 20855916. https://www.ncbi.nlm.nih.gov/pubmed/20855916.

[3] T. Hulsen, VennDiagram.tk, http://www.venndiagram.tk.

[4] T. Hulsen, J. de Vlieg, and W. Alkema, BioVenn - a web application for the comparison and visualization of biological lists using area-proportional Venn diagrams, *BMC Genomics* **9** (2008), 488. PubMed ID: 18925949. https://www.ncbi.nlm.nih.gov/pubmed/18925949.

[5] Google Scholar Citations for 'BioVenn – a web application for the comparison and visualization of biological lists using area-proportional Venn diagrams', https://scholar.google.com/scholar?cites=16587750604719531070.

[6] E. Noma and A. Manvae, colorfulVennPlot: Plot and add custom coloring to Venn diagrams for 2-dimensional, 3-dimensional and 4-dimensional data, Version 2.4, https://CRAN.R-project.org/package=colorfulVennPlot.

[7] J. Larsson, eulerr: Area-Proportional Euler and Venn Diagrams with Ellipses, Version 6.1.0, https://cran.r-project.org/package=eulerr.

[8] J.G. Perez-Silva, M. Araujo-Voces, and V. Quesada, nVenn: generalized, quasi-proportional Venn and Euler diagrams, *Bioinformatics* **34** (2018), 2322-2324. PubMed ID: 29949954. https://www.ncbi.nlm.nih.gov/pubmed/29949954.

[9] A. Dusa, venn: Draw Venn Diagrams, Version 1.9, https://CRAN.R-project.org/package=venn.

[10] H. Chen and P.C. Boutros, VennDiagram: a package for the generation of highly-customizable Venn and Euler diagrams in R, *BMC Bioinformatics* **12** (2011), 35. PubMed ID: 21269502. https://www.ncbi.nlm.nih.gov/pubmed/21269502.
[11] L. Wilkinson and S. Urbanek, venneuler: Venn and Euler diagrams, Version 1.1-0, https://cran.r-project.org/package=venneuler.
[12] Z. Xu, R.W. Oldford, and M. Lysy, vennplot: Venn Diagrams in 2D and 3D, Version 1.0, https://cran.r-project.org/package=vennplot.
[13] K. Tretyakov, Matplotlib-Venn Python package at PyPi, Version 0.11.6, https://pypi.org/project/matplotlib-venn/.
[14] K. Grigorev, PyVenn Python package at PyPi, Version 0.1.3, https://pypi.org/project/venn/.
[15] S. Briois, Biomart Python package at PyPi, Version 0.9.2, https://pypi.org/project/biomart/.
[16] A.M. Altenhoff, C.M. Train, K.J. Gilbert, I. Mediratta, T. Mendes de Farias, D. Moi, Y. Nevers, H.S. Radoykova, V. Rossier, A. Warwick Vesztrocy, N.M. Glover, and C. Dessimoz, OMA orthology in 2021: website overhaul, conserved isoforms, ancestral gene order and more, *Nucleic Acids Res* (2020). PubMed ID: 33174605. https://www.ncbi.nlm.nih.gov/pubmed/33174605.
[17] T. Hulsen, BioVenn R package at CRAN, Version 1.1.0, https://cran.r-project.org/package=BioVenn.
[18] T. Hulsen, BioVenn Python package at PyPI, Version 1.1.0, https://pypi.org/project/BioVenn/.