Exact and Approximate Area-proportional Circular Venn and Euler Diagrams

Leland Wilkinson

Abstract— Scientists conducting microarray and other experiments use circular Venn and Euler diagrams to analyze and illustrate their results. As one solution to this problem, this article introduces a statistical model for fitting area-proportional Venn and Euler diagrams to observed data. The statistical model outlined in this report includes a statistical loss function and a minimization procedure that enables formal estimation of the Venn/Euler area-proportional model for the first time. A significance test of the null hypothesis is computed for the solution. Residuals from the model are available for inspection. As a result, this algorithm can be used for both exploration and inference on real datasets. A Java program implementing this algorithm is available under the Mozilla Public License. An R function `venneuler()` is available as a package in CRAN and a plugin is available in Cytoscape.

Index Terms—visualization, bioinformatics, statistical graphics

1 Introduction

Venn diagrams are collections of n simple closed curves dividing the plane into 2^n nonempty connected regions uniquely representing all possible intersections of the interiors and exteriors of the curves [51]. The requirement that the curves be *simple* means that no more than two curves may intersect in a single point. The requirement that the curves be *closed* means that each curve may have no endpoints and each must completely enclose one or more regions. The requirement that the regions be *nonempty* means that their area must be greater than zero. The requirement that regions be *connected* means that there can be only one region resulting from the intersection of any two closed curves and that one curve may enclose only one region.

Venn diagrams are most frequently used to represent sets; in these applications, there is a one-to-one mapping from set intersections to connected regions in the diagram. Although this definition does not restrict Venn diagrams to collections of circles, the popular form of these diagrams displayed in Venn's original paper and in most applications today involves two or three intersecting circles of constant radius (circles are simple closed curves). Figure 3 shows an example.

Relaxing the restriction that all possible set intersections be represented and the restriction that curves be simple results in an Euler diagram [11]. Figure 7 shows an example. Ruskey [39] discusses various subclasses of the general definitions of Venn and Euler diagrams given here.

This paper involves Venn and Euler diagrams constructed from circles. There are some Venn and Euler diagrams that can be drawn with convex or non-convex polygons that cannot be drawn with circles, so this is a restriction. We add a further restriction in this paper, namely that the areas of polygon intersections be proportional to the cardinalities of intersections among the (finite) sets being represented by the diagram. We call these *area-proportional* Venn and Euler diagrams [5].

Venn and Euler diagrams have had wide use in teaching logic and probability. In almost all of these applications, their use has been confined to two or three circles of equal size. Venn diagrams based on circles do not exist for more than three circles [39]. Higher-order Venn and Euler diagrams can be drawn on the plane with convex or, in some cases, nonconvex polygons [10, 39].

Recently, the microarray community has discovered a new use for these diagrams [22, 33, 31, 9]. To reveal overlaps in gene lists, researchers use Venn and Euler diagrams to locate genes induced or repressed above a user-defined threshold. Consistencies across experiments are expected to yield large overlapping areas. An informal survey of 72 Venn/Euler diagrams published in articles from the 2009 volumes of *Science*, *Nature*, and online affiliated journals shows these diagrams have several common features: 1) almost all of them (65/72) use circles instead of other convex or nonconvex curves or polygons, 2) many of them (32/72) make circle areas proportional to counts of elements represented by those areas, 3) most of them (50/72) involve three or more sets, and 4) almost all of them (70/72) represent data collected in a process that involves measurement error. Figure 1 shows examples from this survey (including popular types in the left column and rare types in the right).

This paper is an attempt to provide an algorithm, called `venneuler()`, that satisfies most of these needs. We use area-proportional circles to construct Venn and Euler diagrams and we build a statistical foundation that accommodates data involving measurement error. As we show through examples and simulations in Section 5,

- The `venneuler()` algorithm produces a circular Venn diagram when the data can be fit by a circular Venn diagram.
- The `venneuler()` algorithm produces an area-proportional circular Venn diagram when the data can be fit by an area-proportional circular Venn diagram.
- It produces an area-proportional circular Euler diagram when data can be fit by that model.
- It produces a statistically-justifiable approximation to an area-proportional circular Venn or Euler diagram when the data can be fit approximately by one of these models.

2 Related Work

There have been two primary approaches to the drawing of Venn and Euler diagrams: axiomatic and heuristic. Axiomatic researchers begin with a formal definition (such as the definition of a Venn diagram given in the Introduction) and then devise algorithms for fulfilling the contract of the definition. These approaches are accompanied by proofs that the algorithm cannot violate the terms of the definition. Heuristic researchers begin with a similar definition, but devise algorithms that produce pleasing diagrams that follow the definition closely, but not provably.

2.1 Axiomatic Approaches

Although axiomatic approaches are distinguished by proofs of correctness, they do vary in their definitions. Fish and Stapleton [13, 14], for example, suggest modifying the definition of an Euler diagram given

- *Leland Wilkinson is Executive VP of Systat Software Inc., Adjunct Professor of Statistics at Northwestern University and Adjunct Professor of Computer Science at University of Illinois at Chicago. E-mail: leland.wilkinson@systat.com*

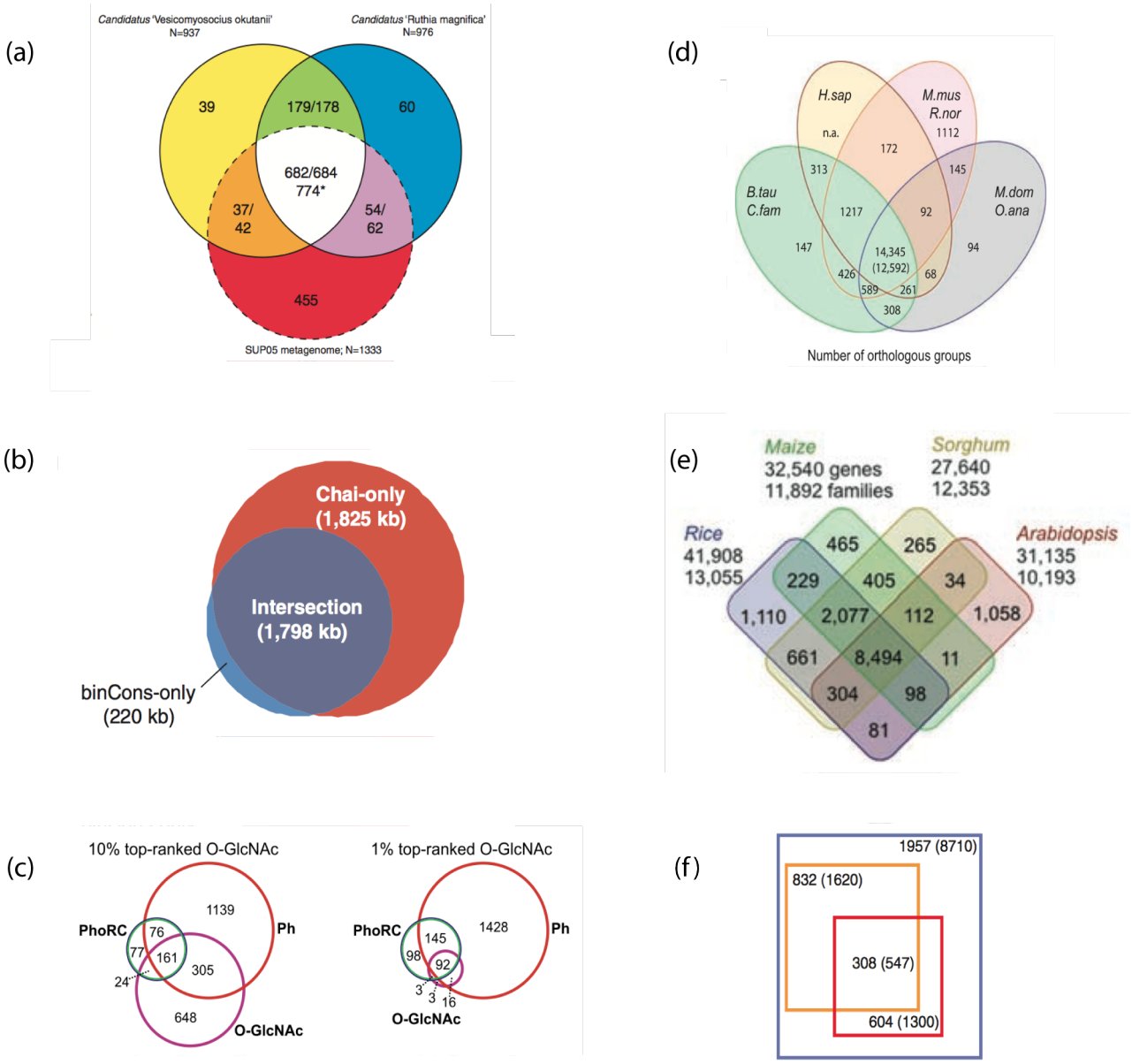

Fig. 1. Examples of Venn and Euler Diagrams excerpted from informal survey of 72 articles published in the 2009 volumes of *Science*, *Nature*, and online affiliated journals. The three diagrams in the left column use circular elements, the most popular form in the journal articles. (a) Three-ring Venn diagram from [52]. (b) Two-ring proportional-area Venn diagram from [34]. (c) Three-ring proportional-area Venn diagrams from [32]. The three diagrams in the right column use non-circular elements, relatively rare forms in the journal articles. (d) Four-ellipse Venn diagram from [2]. (e) Venn diagram using rounded rectangles from [40]. (f) Area-proportional Euler diagram using squares from [20].

above by allowing non-simple curves (curves that may cross themselves). The relaxation allows one to realize any description of set intersections in an Euler diagram. Other definitions may relax the connectedness requirement, so that two or more disjoint regions can represent a single set intersection. Other definitions may relax the requirement that all possible intersections be represented in a Venn diagram by allowing empty regions to be shaded to indicate the lack of a corresponding set intersection in the data. The algorithmic problem in all these approaches is how to satisfy the definitional contract in a single drawing. Proof of existence of a planar diagram does not always translate directly to a practical algorithm.

Chow and Ruskey [5] solved the 2-circle area-proportional Venn problem exactly by computing the area of the intersection of two circles and using this computation to arrange the circles to meet the proportionality requirement. They also solved the 3-circle area-proportional Venn problem by extension, although they show that a solution does not exist for all 3-set specifications. Variants of the Chow-Ruskey algorithm have been used in several applications [12, 21, 35, 42].

Several researchers have worked on axiomatic solutions for Euler diagrams [16, 37, 38, 46] and area-proportional Euler diagrams [36]. In [47] Rodgers and Stapleton build Euler diagrams inductively, by adding one curve at a time based on a dual graph of the diagram. They show that building well-formed Euler diagrams can be guided recursively by examining cycles in the dual graph. The result is an algorithm that in theory can represent any set description with an Euler diagram.

2.2 Heuristic Approaches

Heuristic approaches attempt to draw simple, pleasing diagrams that meet the formal requirements approximately. These methods can be useful for information visualization and informal diagramming of complex information. There have been several approaches toward achieving this goal. Most of these involve iterative refinement of a goodness criterion based on mathematical and sometimes perceptual aspects of diagrams. The most prevalent are summarized here.

Chow and Rodgers [4] fit three-circle area-proportional Venn diagrams to data by using an iterative procedure on an "ad hoc fitness function." The starting point for their solution is an approximation based on axiomatic results in [5]. We will discuss this work further in the last section of this paper.

Some have constructed Euler diagrams by working with the dual graph of Euler regions and employing graph layout algorithms to compute a solution [41]. While axiomatic ideas are involved in the development of these algorithms, the heuristic aspect stems from the use of force-reduction techniques from the graph layout literature [8]. By contrast, Flower, Fish, and Howse [15] develop an axiomatic approach to handling the graph layout itself.

In a series of papers that is most relevant to the present research, Kestler et al. [25, 26] developed an algorithm for the area-proportional generalized Euler problem (more than three sets, circles sized by set cardinality, no connectivity restriction). To deal with the complex intersection-area calculations required for dealing with more than a few sets, they use regular polygons instead of circles. They use a variety of hybrid optimization algorithms to minimize a mathematically and aesthetically based loss function. We will consider their work in more detail in the last section of this paper.

2.3 A New Statistical Approach

The present paper features an algorithm called `venneuler()` that produces generalized circular Euler diagrams for one or more sets based on a statistical goodness-of-fit function. The advantage of this approach is that data with error can be handled appropriately and the goodness-of-fit measure has a probabilistic interpretation. For data without error, the algorithm converges to a solution consistent with axiomatic definitions.

The remainder of this paper concerns this algorithm. We first introduce the algorithm itself. In the following Section, we assess its statistical characteristics. Then we present real and artificial data examples to illustrate its performance. Finally, we compare the `venneuler()` algorithm to other popular approaches to the circular area-proportional Venn and Euler problem.

3 The `venneuler()` Algorithm

The `venneuler()` algorithm is based on a simple statistical regression model, a method for computing areas of intersections of circles, and a minimization function. We present these in sequence.

3.1 Defining the Model

We begin with a list of finite data sets $X = [X_1, X_2, ..., X_n]$ varying in cardinality. Let $P = \bigcap^n \{X\}$ be a list of all possible intersections of the sets in X, including the void set and the intersections of each X_i with itself. P has $m = 2^n$ sets as entries and is ordered as

$$P = [\emptyset, X_1, X_2, X_1 \cap X_2, X_3, X_1 \cap X_3, X_2 \cap X_3, ..., X_1 \cap X_2 \cap ... X_n]$$

The ordering we use for P induces a binary n-bit pattern on each entry of X that we use to index all of our other lists of length m. In other words, each intersection structure in X is uniquely indexed by a length-n binary string that we can use to map entries of X to entries of P [17]. For three sets, this bit pattern list is

$$B = [000, 001, 010, 011, 100, 101, 110, 111]$$

Let $P^- = \texttt{Disjoint}(P)$, where the `Disjoint()` function produces disjoint entries through hierarchical set differencing, beginning with the highest-order intersections. Figure 2 contains a graphical illustration of this function. In the left panel of the figure, the seven polygons defined by the three circles and their pairwise and triple intersections represent the non-null entries in the list P. The result of the `Disjoint()` function is illustrated in the right panel.

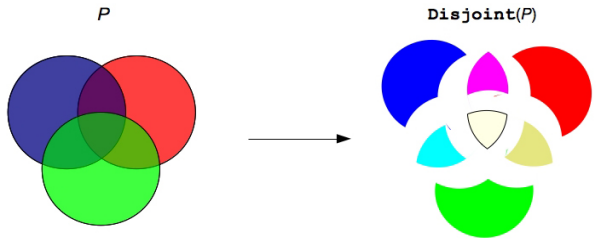

Fig. 2. Illustration of `Disjoint()` function. The function hierarchically decomposes a list of sets and their intersections into a list of disjoint subsets. The left panel represents three sets and their intersections in a Venn diagram. The three two-set intersections share a subset, namely the three-set intersection. The `Disjoint()` function partitions the three sets into seven disjoint subsets of their union.

Next, we construct a list of disks, $D = [D_1, \ldots, D_n]$, each having area $A_i = |X_i| / |\bigcup\{X\}|$. Each disk D_i is centered on coordinates (x_i, y_i). From D, we construct the corresponding list

$$Q = [\emptyset, D_1, D_2, D_1 \cap D_2, D_3, D_1 \cap D_3, D_2 \cap D_3, ..., D_1 \cap D_2 \cap ... D_n]$$

We then apply the same disjoint operation we used on P in order to produce $Q^- = \texttt{Disjoint}(Q)$. We now have a one-to-one correspondence between the entries of Q^- (disjoint disk intersections) and the entries of P^- (disjoint set intersections). Both are indexed by the same list of binary strings B.

From P^- and Q^- we make a column vector $\mathbf{c} = (|P_1^-|, \ldots, |P_m^-|)$ consisting of the counts of elements in each disjoint intersection of the sets in X and a column vector $\mathbf{a} = \texttt{Area}(Q_1^-, \ldots, Q_m^-)$ consisting of the areas of the disjoint disk intersections.

Given these entities, a Venn diagram with areas proportional to counts is defined by the equation

$$\mathbf{a} = \beta \mathbf{c} \qquad (1)$$

The parameter β is a scalar coefficient that makes areas proportional to counts.

There may not exist a set of coordinates (x_i, y_i) for which this equation is satisfied. Moreover, we will assume the elements in the data sets X_i are generated by a process having a random component. Our model is therefore

$$\mathbf{a} = \beta\mathbf{c} + \varepsilon \tag{2}$$

where ε is a random variable with zero expected value. Our ordinary least-squares estimate of β in this case is

$$\hat{\beta} = \mathbf{a}'\mathbf{c}/\mathbf{c}'\mathbf{c} \tag{3}$$

The loss in fitting this model is the sum of squared residuals:

$$\begin{aligned} SSE &= (\mathbf{a} - \hat{\beta}\mathbf{c})'(\mathbf{a} - \hat{\beta}\mathbf{c}) \\ &= (\mathbf{a} - \hat{\mathbf{a}})'(\mathbf{a} - \hat{\mathbf{a}}) \end{aligned} \tag{4}$$

3.2 Computing Areas

For a few circles, analytic computation of areas in Q^- is straightforward [4, 5]. With more than three, computations increase exponentially. Kestler et al. [25, 26] worked with regular polygons instead of circles and employed standard polygon intersection algorithms. This method is not only expensive, but it also fails to deal directly with the circles that researchers want to use.

A simple method for solving this problem is based on numerical quadrature and binary indexing. In order to compute areas on the entries of Q^-, we "draw" circles on n bit planes, each of resolution $p \times p$. Each "pixel" in a bit plane has the value 1 if it is inside a circle and 0 if not. The string of 1's and 0's derived from passing through the corresponding pixel on each bit plane yields the same binary indexing that we use for Q^- itself. We simply sum the result over all pixels to get intersection areas. The method is very fast. On the MacBook Pro used for this paper, running through a 200x200 byte array to compute these areas takes about a millisecond. Since we need to run through n such grids to detect which entries of Q^- are indexed by each cell in the grid, the complexity of this computation is $O(n)$. In practice, $p = 200$ is sufficient resolution to allow the iterations to converge. On the examples in this paper, increasing resolution beyond 200 had no effect on the visual appearance and led to changes in stress of less than .001.

3.3 Initial Circle Locations

The `venneuler()` algorithm will usually work with random starting locations for the circles. It is more efficient, however, to begin with a rational starting configuration. A rational start also reduces the likelihood of encountering a local minimum [45]. To accomplish this, we adopt an approach from classical multidimensional scaling [49]. We compute a Jaccard [23] distance matrix

$$\mathbf{D} : d_{ij} = |X_i \cap X_j| / |X_i \cup X_j| \tag{5}$$

We then choose an arbitrary row (col) in $\mathbf{D}$ and compute a matrix of scalar products on the distances conditioned on this row k. The resulting matrix is

$$\mathbf{W}_k : w_{ijk} = d_{ik} d_{jk} \cos\theta_{ikj}, \tag{6}$$

where

$$\cos\theta_{ikj} = (d_{ik}^2 + d_{jk}^2 - d_{ij}^2)/2 \tag{7}$$

We then compute the singular value decomposition

$$\mathbf{W}_k = \mathbf{U}\mathbf{V}\mathbf{U}' \tag{8}$$

The starting coordinates (x_i, y_i) are found in the rows of the first two columns of $\mathbf{U}$. We standardize these coordinates so that they have unit dispersion.

3.4 Circle Diameters

Initial circle diameters are scaled so that their areas sum to unity. Because the coordinates for the circle centers have been standardized, the initial solution tends to have overlapping circles wherever intersections occur in the data. Iterations proceed by holding diameters fixed and moving the circle centers.

3.5 Minimizing Loss

Our remaining task is to find the coordinates (x_i, y_i) that minimize the sum of squared residuals (SSE) from data fit by Equation 4. We work with a normalized loss, which we call *stress*. Stress is defined as SSE/SST (residual sum of squares divided by total sum of squares).

We use the method of steepest descent with a gradient approximation calculated from our model. The analytical gradient is a function of circle intersection areas, however, and we do not have access to these values except through numerical integration. Consequently, we work with an approximation to the gradient. For each disk D_i centered on (x_i, y_i) the descent step on each iteration, based on summing over all the areas a_k, is roughly proportional to

$$\bigtriangledown F(x,y)_i \approx \sum_{k=1}^{m} \sum_{i \neq j} \{(x_i - x_j)(a_k - \hat{a}_k), (y_i - y_j)(a_k - \hat{a}_k)\}, \tag{9}$$

where B_k (the kth element in the bit pattern list B) has nonzero bits i and j. This last condition means that, for a given disk D_i, we calculate the gradient approximation based on every lune (intersection) it contains.

We use a step size of .01 with this quasi-gradient to follow the descent path. Iterations proceed rapidly because we already have residuals on each iteration from having computed stress.

If the residuals are relatively large, this gradient approximation is relatively rough; it gets us toward the minimum, but it can overshoot the minimum and retard convergence. Consequently, we compute a final set of iterations using a closer (but more time consuming) approximation to the gradient. For this local gradient approximation, we compute stress four times for each circle center by taking small steps (.01) in a cross pattern on the plane (up, down, left, right). The gradient direction is the resultant of the lowest stress values for steps on x and y.

This use of a quasi-gradient resembles the way a gradient is approximated in stochastic gradient descent [43], but it is deterministic. Because we begin with a rational initial configuration, and because gradient descent is fairly robust to disturbances in direction, the iterations converge to a minimum in reasonable time.

3.6 Goodness of Fit

At convergence, a correlation coefficient can be computed as

$$r = \sqrt{1 - stress} \tag{10}$$

This correlation, based on regression without a constant, differs from the ordinary Pearson correlation. It tends to be larger than the Pearson in practice and needs to be interpreted with caution [29]. The next section discusses a statistical test that should be used before any interpretation.

4 The Distribution of the Stress Statistic

We computed a Monte Carlo simulation to estimate the distribution of our stress statistic. For each number of circles ($n = 3, \ldots 10$), we generated 100 simulations. For each simulation, we generated 2^n uniform random numbers to represent the areas based on the entries in Q^-. We ran `venneuler()` on the random data and computed order statistics on the resulting stress values. For the empirical stress fractiles $s_{.01}$ and $s_{.05}$, we fit the logistic function

$$s_\alpha = \exp^{b(n-c)} / (1 + \exp^{b(n-c)}) \tag{11}$$

The fit for both equations was extremely close ($r^2 > .99$). For $s_{.01}$, $c = 6.105, b = 0.909$ and for $s_{.05}$, $c = 5.129, b = 0.900$. Table 1 shows the critical values for $n = 3, \ldots, 10$.

These stress values are substantially higher than corresponding critical stress values in the multidimensional scaling literature, assuming n represents the number of points [1, 7, 27, 44, 48]. The `venneuler()` model is much more constrained than the MDS model, however. Not only are all possible pairwise intersections included in the loss function, but also all higher-order intersections are included. Moving points (disk centers) around on the plane affects 2^{n-1} areas rather than $n(n-1)/2$ distances as in MDS. Furthermore, the regression function on which loss is based has a zero intercept; MDS ordinarily includes an intercept parameter.

Table 1. Critical stress values for `venneuler()`

n	$s_{.01}$	$s_{.05}$
3	0.056	0.128
4	0.129	0.266
5	0.268	0.471
6	0.476	0.687
7	0.693	0.843
8	0.848	0.930
9	0.933	0.970
10	0.972	0.988

5 EXAMPLES

Figure 3 shows a 2-ring Venn diagram produced by the input:

$$\texttt{venneuler}(A = \{a, ab\}, B = \{b, ab\})$$

The stress value for this solution is zero, with each of the 3 areas equal to a third.

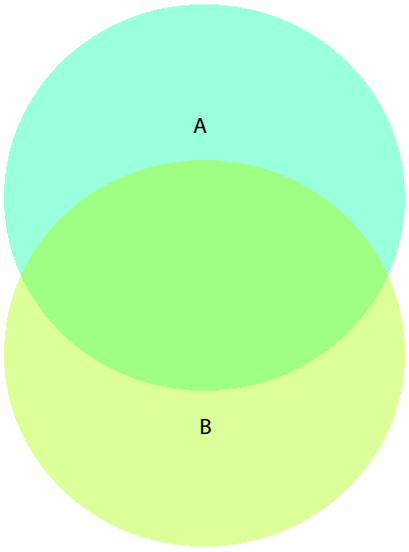

Fig. 3. Two-ring Venn diagram produced by `venneuler()` on a list of the elements in two sets. These sets share one element, namely ab.

Figure 4 shows a 3-ring Venn diagram produced by the input:

$$\begin{aligned}\texttt{venneuler}(A &= \{a, ab, ac, abc\},\\ B &= \{b, ab, bc, abc\},\\ C &= \{c, ac, bc, abc\})\end{aligned}$$

The stress value for this solution is 0.103, consistent with the fact that an equal-area solution for 3 equal-sized circles does not exist even though the ordinary Venn diagram requirement is met [5, 18]. Nevertheless, this is as close to equal-area as we can get; Figure 4 resembles aesthetically the canonical "Ballantine" charts appearing in Venn diagram tutorials [39].

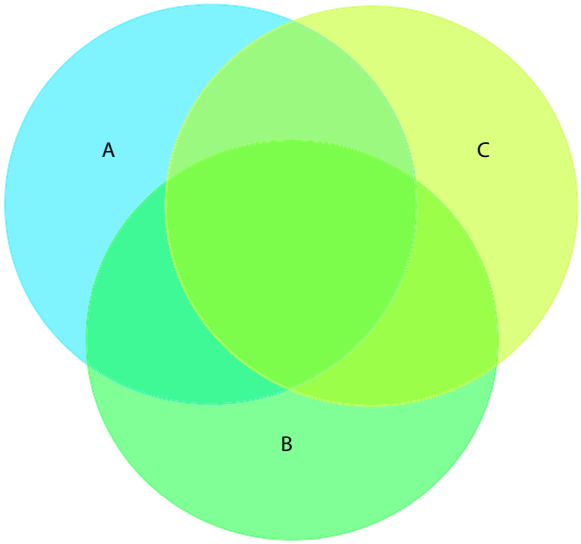

Fig. 4. Three-ring Venn diagram produced by `venneuler()` on a list of the elements in three sets.

Figure 5 shows a 4-ring diagram produced by `venneuler()` on data that lack some 3-way intersections:

$$\begin{aligned}\texttt{venneuler}(A &= \{a, ab, ac, ad, abc, abd, acd, abcd\},\\ B &= \{b, ab, bc, bd, abc, abd, bcd, abcd\},\\ C &= \{c, ac, bc, cd, abc, acd, bcd, abcd\},\\ D &= \{d, ad, bd, cd, abd, acd, bcd, abcd\})\end{aligned}$$

The resulting diagram has two 2-way intersections ($A \cap C$ and $B \cap D$) missing in the plot. It nevertheless approximates the Euler diagram for this set in the way we would expect. There is a trade-off between moving the circles outward to eliminate the 3-way areas and moving them inward to represent the 4-way area. The stress for this solution is .30.

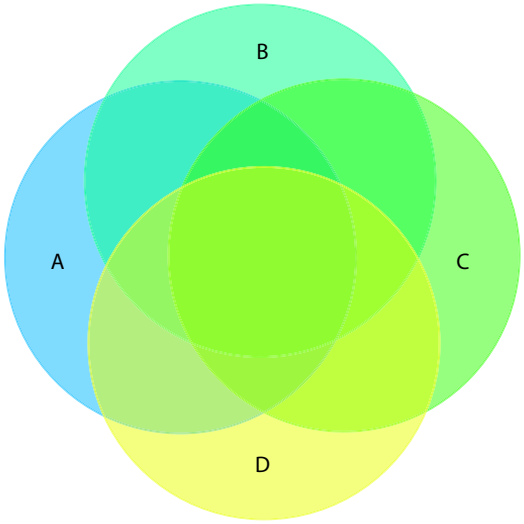

Fig. 5. Four-ring diagram produced by `venneuler()` on a list of the elements in four sets, with some intersections missing.

Figure 6 contains a residual plot from the solution in Figure 2. This

plot reveals the trade-off the `venneuler()` algorithm made. The two smallest residuals show that AC and BD are under-predicted. This happens because the 3-way intersections (for which there are no data values) are stealing area from these 2-way intersections (for which there are data values). The largest residual reveals that the four-way intersection is too large. Residual plots are a natural complement to graphics produced by `venneuler()`. They are useful for diagnosing the statistical properties of the solution, a feature unavailable in ad-hoc algorithms.

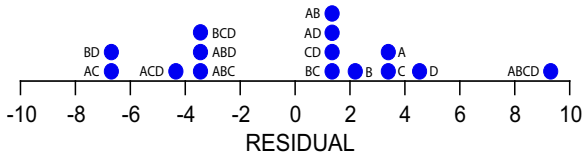

Fig. 6. Residuals for four-ring diagram in Figure 2.

Figure 7 shows a 6-ring Euler diagram produced from areas calculated from a pre-existing diagram. We drew six circles on a piece of paper, measured intersection areas, and rounded each to the nearest single-digit integer. In this version of the syntax, we directly input areas for each disjoint subset. For this type of input, the `venneuler()` program uses an ampersand to represent intersection.

$$\begin{aligned}\texttt{venneuler}(&A=4,B=6,C=3,D=2,E=7,F=3,\\ &A\&B=2,A\&F=2,B\&C=2,B\&D=1,\\ &B\&F=2,C\&D=1,D\&E=1,E\&F=1,\\ &A\&B\&F=1,B\&C\&D=1)\end{aligned}$$

The stress for this solution is .006. Although we rounded the areas to integers, the reproduced diagram closely resembles the original, confirming the ability of `venneuler()` to capture a moderately complex, low-error structure.

The question remains whether `venneuler()` can reproduce other error-free area-proportional Venn and Euler diagrams. For example, `venneuler()` reproduces exactly the area-proportional Venn diagram in Figure 3 of [5]. To test this proposition more generally, however, we generated 100 proportional Euler diagrams with number-of-circles varying from 2 to 11, diameters randomly varying from .3 to .7, and center coordinates randomly varying between .15 and .85. We then used the bitmap algorithm in Section 3.2 to calculate the areas of the disjoint polygons produced by the circles and their intersections. We then ran `venneuler()` on each input dataset. The average stress for the `venneuler()` solutions on these datasets was .006 with a standard deviation of .009.

There is a more stringent criterion we can use for this test, however. This involves a worst-case analysis. To do the analysis, we normalized the total areas of the input diagrams and the output diagrams to be 1 in order to compare inputs and outputs. We then computed for each solution the maximum discrepancy between the area of any input disjoint polygon and its corresponding output disjoint polygon. The average worst error was .013 with a standard deviation of .009 across the 100 diagrams. These errors are not significantly different from zero. Furthermore, errors of this magnitude are below the threshold of visual detectability of area differences [30].

Figure 8 shows two diagrams for data shown in Figure 1 of [24]:

$$\begin{aligned}\texttt{venneuler}(&SE=13,Treat=28,Anti\text{-}CCP=101,\\ &DAS28=91,SE\&Treat=1,SE\&DAS28=14,\\ &Treat\&Anti\text{-}CCP=6,SE\&Anti\text{-}CCP\&DAS28=1)\end{aligned}$$

These diagrams depict the overlap of genes detected in four different populations. The left panel is from the original article. The original graphic shows four circles, so the sets cannot be represented by a circular Venn diagram, Nevertheless, the Euler diagram in the right panel computed by `venneuler()` quite accurately represents the data. The stress for this solution is .001.

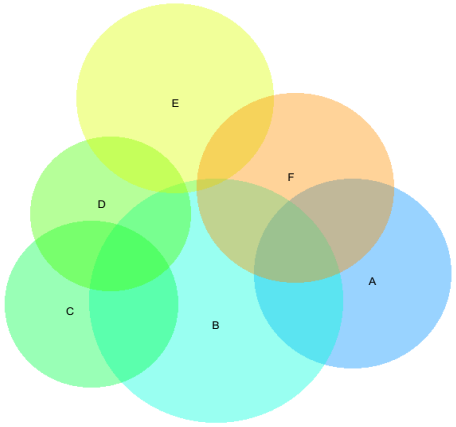

Fig. 7. Six-ring Euler diagram almost perfectly fit to an artificial dataset. The input to `venneuler()` consists of the areas of the disjoint polygons (see Figure 2) produced by this arrangement of circles. The stress for this solution is .006.

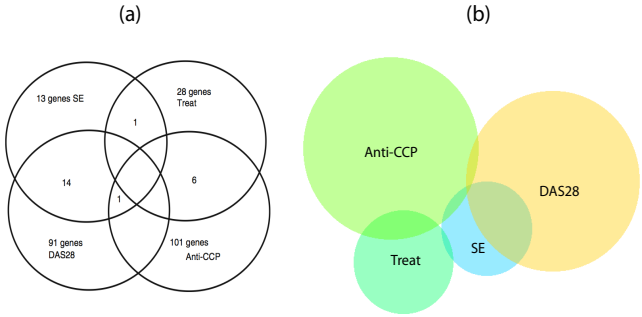

Fig. 8. Four-ring diagram of data underlying Figure 1 of [24]. Panel (a) is reproduced from the original article. Panel (b) is the `venneuler()` solution. The correlation between the areas of the circles and their intersections is greater than .99.

Figure 9 shows an Euler diagram for 12 animals based on gene lists downloaded from the Agilent DNA oligo microarray database (`http://www.chem.agilent.com`). The analysis was based on 404,528 gene symbols and 12 animal names. The stress for this solution is .01, with corresponding correlation of .99. Many genes in these lists have yet to be classified. When this task is completed, we would expect to see more overlap in genomes. Nevertheless, the `venneuler()` solution provides a reasonably accurate portrait of this work in progress.

Figure 10 shows an Euler diagram for six works of English literature (we treat the King James translation of the *Bible* as English literature in this context because it is canonical not only among English Bibles but also in English literature). We downloaded files from the Project Gutenberg Web site (`http://www.gutenberg.org/wiki/Main_Page`). Stop words (*a, and, the, ...*) were filtered and a list of distinct words was constructed for each corpus. The combined lists (totaling 65,432 words) were submitted to `venneuler()`.

The stress for this solution is .04, with a corresponding correlation of .98. The size of the circles is based on the number of unique words in each book. *Ulysses*, the King James translation of the *Bible*, and *Moby Dick* anchor the configuration; they contain the lion's share of unique words. *Ulysses* is notable for its large number of unique words – a familiar aspect to anyone struggling to read that novel. The *Bible* has a smaller number of unique words; many of these are proper names not shared by the other literature. Not surprisingly, Shakespeare's language in *Macbeth* shares much with its contemporary, the King James *Bible*.

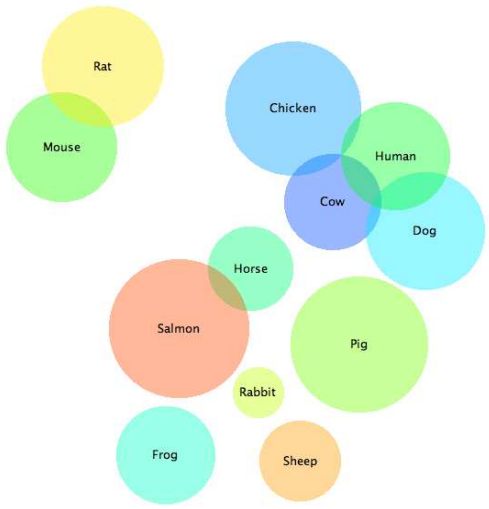

Fig. 9. Euler diagram for 12 gene lists containing 404,528 unique genes from the Agilent DNA oligo microarray database. The sizes of the circles and intersections are due to the number of genes listed for each animal in the database as opposed to the count of the genome itself. The correlation between the areas of the circles and their intersections and the gene counts is .99.

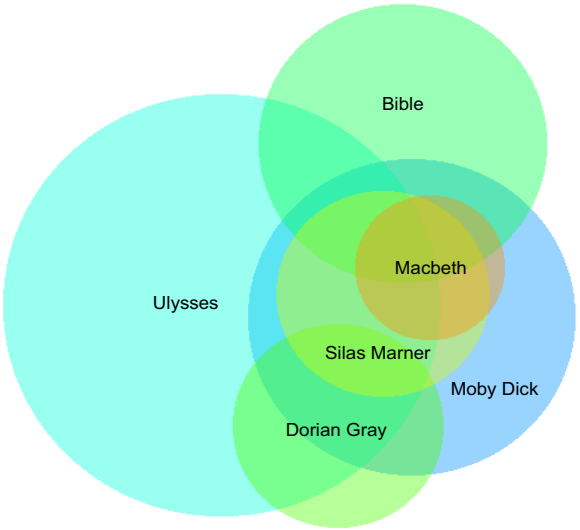

Fig. 10. Euler diagram for six works of English literature. The sizes of the circles are proportional to the number of unique words in each of the works. The areas of the intersections are proportional to the number of shared words among the works. The correlation between the areas of the circles and their intersections and the word counts is .99.

6 Comparisons with Other Venn/Euler Algorithms

In this section we compare `venneuler()` with similar generalized Venn and Euler algorithms. We will discuss first the most widely-known generalized Euler program, VennMaster [26], and show in more detail that it is unnecessarily complicated and rests on an inappropriate model. In addition, we compare `venneuler()` to a proportional-area 3-ring Venn Diagram program by Chow and Rodgers [4].

6.1 VennMaster

Several points are noteworthy.

1. VennMaster uses a complicated polygon intersection algorithm. Areas of intersections are computed for a pair of polygons in $O(m+n)$ time, where m and n are the number of vertices in each polygon. The authors note that there are several exceptions to worry about and the code to implement the algorithm is not simple. The complexity of this computation increases exponentially with the number of polygons. By contrast, the complexity of the `venneuler()` area calculation is linear in the number of polygons (circles). And instead of employing regular polygons, which reduces the precision of the solution, `venneuler()` uses high-resolution quadrature directly on circles and their intersections. Increasing the number of polygon vertices in VennMaster to approximate the resolution of the circles in `venneuler()` slows computation considerably.

2. VennMaster uses several different loss functions that appear to be governed more by aesthetic considerations than by the conventional definition of area-proportional Venn diagrams. One of these includes weighting intersections differently for small polygons than for large. However,"proportional" means $a/c = k$, where k is a constant. The `venneuler()` loss function implements this conventional definition: areas are proportional to the sizes of subsets. There is no need to weight large areas more heavily. The ordinary least squares zero-intercept regression model with equal weights gives large values greater leverage by default [3].

3. VennMaster uses stochastic optimization algorithms (an evolutionary algorithm with sensitive mutation parameters in one case, and swarm optimization in another). This choice may be due to the complexity of its loss functions, which do not lend themselves to simple gradient-based methods. These algorithms can reach different solutions for different random starts. By contrast, `venneuler()` uses ordinary steepest descent, which is a standard algorithm for multidimensional scaling and manifold learning. There is no random number generator in `venneuler()`. Repeated runs produce the same result.

4. VennMaster assumes that "all sets have at least one intersecting partner." The `venneuler()` model does not require this assumption.

5. VennMaster uses a fixed starting configuration of circles centered at one location. There are numerous studies showing that fixed and random initial configurations lead to local minima in optimization problems like this, e.g., [6, 45]. The `venneuler()` program begins with a rational starting configuration computed via a singular value decomposition. There is no need for global optimization methods such as simulated annealing, genetic algorithms, or swarm algorithms because the initial metric approximation is known to be close to the minimum [50, 28]. In addition to avoiding local minima, a rational start speeds convergence.

6. Stochastic optimization and polygon intersection calculations cause VennEuler to become unwieldy for larger problems. The VennMaster program took over 10 minutes to compute a diagram for the gene data in Figure 9. The `venneuler()` program computed this diagram in 10 seconds. Both programs were run on a

2.5 GHz MacBook Pro running the Java 1.5 Virtual Machine in 2GB of allocated memory. Despite this order-of-magnitude difference in computation time, the stress value for VennMaster was worse than that for `venneuler()` (.036 vs. .014).

A few examples suffice to illustrate the severity of these problems. Figure 11 shows two Euler diagrams based on the test dataset (example1.list) that comes with the VennMaster installation. The diagrams have been rotated and labeled to facilitate comparisons. The VennMaster (version 0.37.3) solution is in the top panel. The stress for this solution is .79. (VennMaster stress values were computed by inputting the VennMaster solution to `venneuler()` without further iterations). The `venneuler()` solution is in the bottom panel. The stress for this solution is .41. Even though the solutions seem to be similar in a cursory glance, the stress values for these solutions differ considerably and the Spiny polygon is in a completely different location. This may be due to the use of a different loss function in VennMaster, or it is possible that this program encountered a local minimum. In any case, the VennMaster solution does not come close to making areas proportional to cardinalities (except, of course, for the setwise polygon sizes).

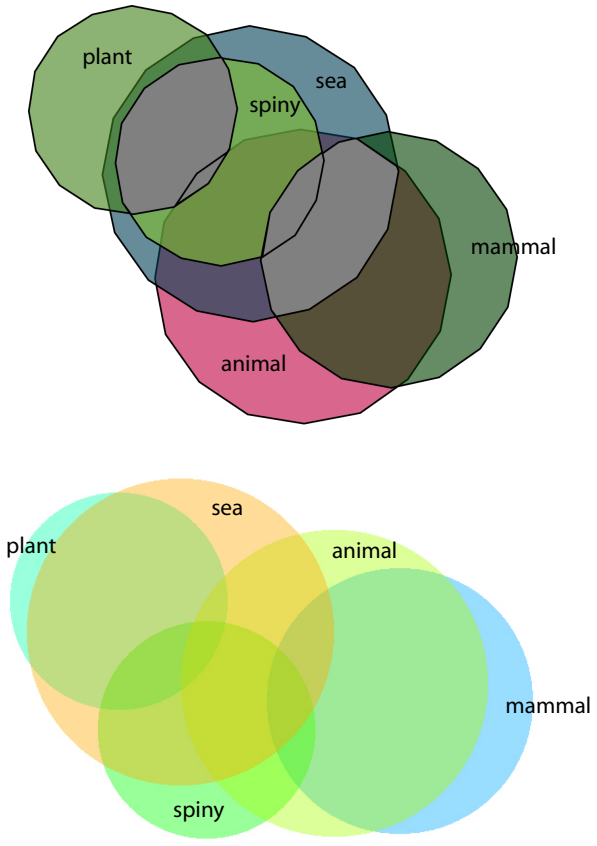

Fig. 11. VennMaster (top) and venneuler (bottom) solutions to an example dataset in the VennMaster build. VennMaster has a stress of .79; venneuler has a stress of .41.

Most importantly, however, the VennMaster program gives no indication that its solution is not acceptable. Instead, it reports "no inconsistencies." The `venneuler()` program, in contrast, prints a $stress_{05}$ value of .47 and a $stress_{01}$ value of .26. On the basis of these critical values, our most reasonable conclusion regarding the VennMaster solution is that it could have resulted from scaling random data. The `venneuler()` solution, while having a considerably lower stress, barely beats the conventional significance level itself. The `venneuler()` program warns us not to take this layout seriously.

Finally, VennMaster does not converge probabilistically to a global minimum, despite the use of global optimization. Figure 13 shows the results of 10 VennMaster solutions on the dataset used in Figure 7. Each solution was produced by initializing the random number seed in VennMaster with a uniformly distributed random integer between 0 and 10,000. Only two of the ten solutions (top left and middle right) are correct. Changing the optimizer from Particle Swarm to Evolutionary-new did not improve this poor performance. The VennMaster performance is even worse with the data in Figure 5. In ten random starts, VennMaster never came up with the minimum-loss solution shown in Figure 5. The best it could do was to overlap two of the four circles and display a three-ring Ballantine or, in a few instances, overlap three of the four and display a two-ring diagram. For these degenerate solutions, VennMaster reported no inconsistencies. However, the residual plot in Figure 6 shows that we need to worry about areas, not inconsistencies. The VennMaster solutions to this dataset are seriously wrong because they imply that two or more sets are identical.

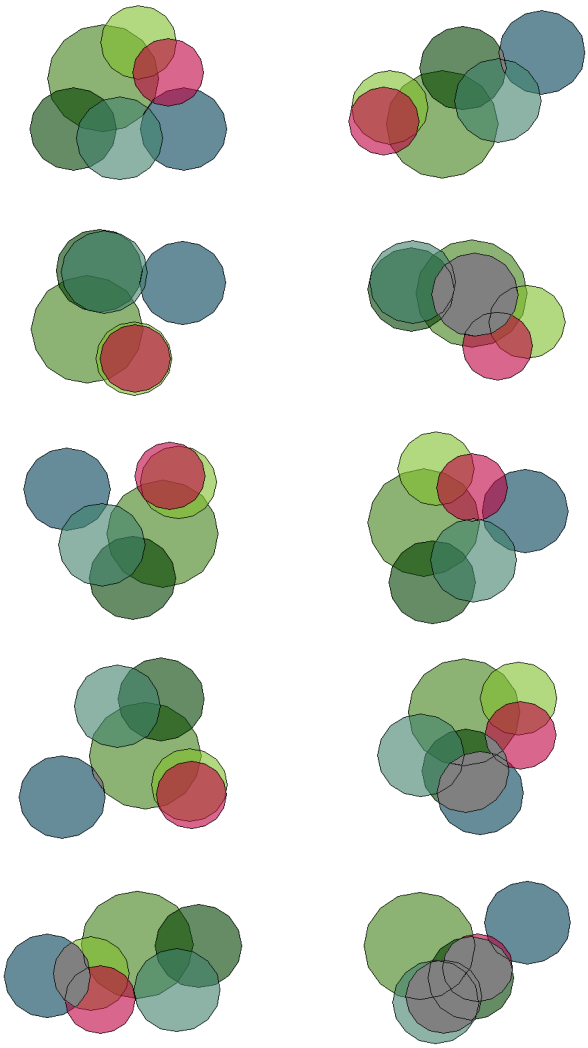

Fig. 12. Ten instances of VennMaster solutions on the dataset used in Figure 7, each based on a different random number seed. Only two solutions (top left and middle right) are correct.

6.2 Chow and Rodgers

The Chow/Rodgers algorithm is implemented in an applet at `http://theory.cs.uvic.ca/venn/EulerianCircles/`. It is discussed in [4]. The authors acknowledge that the loss function and minimization algorithm are ad hoc. Their loss differs from that in `venneuler()` in a number of respects, so it is not easy to characterize the differences. In particular, there are trade-offs between the

proportionality condition for subsets and for the circles themselves. Furthermore, the circle sizes are free to vary from iteration to iteration, so that the convergent solution may not represent set sizes accurately.

Figure 13 shows a comparison of solutions on a dataset from Figure 4 in [4]. The correlation between the areas and the data is .942 for Chow/Rodgers and .988 for `venneuler()`. The differences between solutions lead one to wonder whether the ad hoc loss function and minimization in Chow/Rodgers is worth the effort, especially because the use of an ad hoc loss function breaks the connection between the conventional model (proportional areas) and the visualization. There may be counterexamples to justify this effort, but they do not appear in [4].

We can get an idea of the absolute discrepancy between the two solutions by measuring the total absolute error in terms of counts. The total absolute count error for the Chow/Rodgers solution is 202 (by differencing the numbers across the colons in Figure 13 and summing the absolute differences) and for `venneuler()` it is 98 (by inverting the regression function of areas on counts and summing the absolute residuals). This is not an insubstantial difference. Nevertheless, more research needs to be done on whether adjusting areas via a psychometric (Stevens) function might improve the accuracy of *perception* of subset size in examples like this.

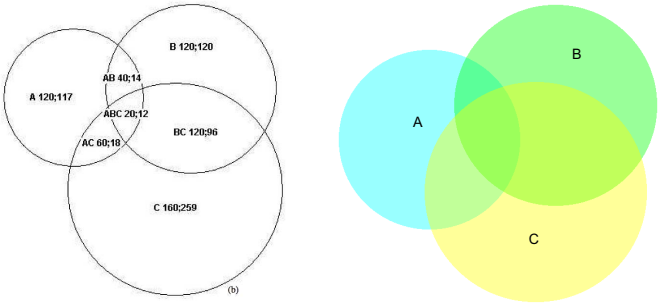

Fig. 13. Chow/Rodgers (left) and `venneuler()` (right) solutions to an example dataset in Chow and Rodgers [4]. The left diagram has a stress of .113 and the right has a stress of .024.

6.3 Discussion

All three programs compared in this section produce pretty pictures when given set-wise data. As we have seen, however, the VennMaster solutions cannot be trusted and there is no way of recognizing bad solutions by looking at them. Consequently, it is important to have a statistical basis for evaluating the quality of a given solution. The `venneuler()` loss function and its grounding in standard regression methodology makes this possible.

The Chow/Rodgers algorithm is limited to 3-ring generalized Venn diagrams, but it raises similar questions. More extensive study using a wider variety of datasets would be needed to establish a more conclusive evaluation. In our testing, we did not encounter examples for Chow/Rodgers that were as seriously wrong as the VennMaster solutions. Since `venneuler()` solves a superset of the Chow/Rodgers problem, however, there is no convincing evidence for using Chow/Rodgers on simpler problems.

7 Conclusion

The algorithm described in this report provides for the first time a statistical basis for estimating area-proportional circular Venn and Euler diagrams on real data. Its distinguishing features include a statistical loss function that accommodates data with error, the ability to evaluate probabilistically the goodness-of-fit of a solution, the ability to represent counts proportionally by areas, and the ability to accommodate unconnected sets.

Based on these results, we can suggest one reasonable strategy for producing Venn and/or Euler diagrams on setwise data. If the data are known to contain no error, then we should employ an axiomatic algorithm. First, however, it would be advisable to try `venneuler()`. If the stress for the `venneuler()` solution is less than .01, then we should consider staying with that result. The main reason for this approach is that the area-proportional circular model is widely known. Non-circular closed curves are best used for those set specifications that cannot be represented perfectly by the circular model. If the `venneuler()` stress is nonzero in the error-free-data case, then one should proceed up the hierarchy from axiomatic algorithms designed for simple curves through the more complex models discussed in [13].

If the data are known to contain error (e.g., gene expression lists, psychological and social science data), then axiomatic models are likely to be inappropriate. The reason for this lies in the fact that overfitting sample data (or, in the extreme, predicting sample data perfectly) can increase prediction error in new samples [19]. Error terms are designed to model sample error without biasing the estimates of other parameters (such as the shape, size, or location of Euler curves). In practical terms, the shape of Euler curves from an axiomatic model applied to data containing error cannot be expected to hold for new samples from the same universe.

Faced with data containing error, the researcher has to rely on a statistical measure of goodness-of-fit. The `venneuler()` stress statistic serves this role. If it is relatively small and significantly different from the stress value expected for random data, then the researcher has some confidence that a model is a good fit to the data and that the model will generalize to new samples from the same universe.

The `venneuler()` model is not the only possible statistical model for fitting Venn and Euler curves, of course. There is no reason the axiomatic models described by Ruskey, Rodgers, Stapleton, Fish, and others cannot be modified to accommodate error. Finding statistical algorithms to fit these more complex models is a nontrivial enterprise, however. In any case, the problem is important enough to merit further research. Two areas would appear to be especially promising: 1) relaxing the circle requirement in order to implement a statistical algorithm on ellipses or rectangles, and 2) embedding these algorithms in an expert system that could recognize when an axiomatic approach is more appropriate than a statistical approach on a given dataset.

Acknowledgments

Simon Urbanek (AT&T Labs) installed `venneuler()` as an R package in CRAN (`http://www.rforge.net/venneuler/`). Michael Smoot (UC San Diego) installed the same code in Cytoscape (`http://www.cytoscape.org/`). Anushka Anand, Tuan Nhon Dang, and three reviewers contributed valuable suggestions. Funding was provided by NSF/DHS grant DMS-FODAVA-0808860

Leland Wilkinson received the PhD degree from Yale University in 1975. He is a senior vice president of SPSS Inc. and an adjunct professor of statistics at Northwestern University. He wrote the SYSTAT statistical package and founded SYSTAT Inc. in 1984. He joined SPSS in a 1994 acquisition and now works on research and development of visualization systems. He is a fellow of the American Statistical Association and a member of the Committee on Applied and Theoretical Statistics of the National Academy. In addition to journal articles and the original SYSTAT computer program and manuals, He authored (with Grant Blank and Chris Gruber) Desktop Data Analysis with SYSTAT and The Grammar of Graphics.

References

[1] I. Borg and P. J. F. Groenen. *Modern Multidimensional Scaling: Theory and Applications*. Springer, 2005.

[2] Bovine Genome Sequencing and Analysis Consortium, C. G. Elsik, R. L. Tellam, and K. C. Worley. The genome sequence of taurine cattle: A window to ruminant biology and evolution. *Science*, 324:522–528, 2009.

[3] G. Casella. Leverage and regression through the origin. *The American Statistician*, 37:147–152, 1983.

[4] S. Chow and P. Rodgers. Constructing area-proportional Venn and Euler diagrams with three circles. In *Euler Diagrams Workshop 2005*, August 2005.

[5] S. Chow and F. Ruskey. Drawing area-proportional Venn and Euler diagrams. In *Graph Drawing*, volume 2912 of *Lecture Notes in Computer Science*, pages 466–477. Springer, 2004.

[6] A. K. Clark. Re-evaluation of monte carlo studies in nonmetric multidimensional scaling. *Psychometrika*, 41:401–403, 1976.

[7] T. F. Cox and M. A. A. Cox. *Multidimensional Scaling, Second Edition*. Chapman & Hall/CRC, 2000.

[8] G. Di Battista, P. Eades, R. Tamassia, and I. G. Tollis. *Graph Drawing*. Prentice Hall, Upper Saddle River, NJ, 1999.

[9] J. R. Dinneny, T. A. Long, J. Y. Wang, J. W. Jung, D. Mace, S. Pointer, C. Barron, S. M. Brady, J. Schiefelbein, and P. N. Benfey. Cell identity mediates the response of *arabidopsis* roots to abiotic stress. *Science*, 320:942–945, 2008.

[10] A. W. F. Edwards. *Cogwheels of the Mind: The Story of Venn Diagrams*. Johns Hopkins University Press, Baltimore, 2004.

[11] L. Euler. *Lettres a Une Princesse d'Allemagne*, volume 2. 1761.

[12] H. Fang, S.C. Harris, Z. Su, M. Chen, F. Qian, L. Shi, R. Perkins, and W. Tong. Arraytrack: An FDA and public genomic tool. *Methods in Molecular Biology*, 563:379–398, 2009.

[13] A. Fish and G. Stapleton. Defining Euler diagrams: Choices and consequences. In *Euler Diagrams*, 2005.

[14] A. Fish and G. Stapleton. Defining Euler diagrams: Simple or what? In *Diagrammatic Representation and Inference*, volume 4045 of *Lecture Notes in Computer Science*, pages 109–111. Springer, 2006.

[15] J. Flower, A. Fish, and J. Howse. Euler diagram generation. *Journal of Visual Languages & Computing*, 19:675–694, 2008.

[16] J. Flower and J. Howse. Generating Euler diagrams. In *Diagrammatic Representation and Inference*, volume 2317 of *Lecture Notes in Computer Science*, pages 61–75. Springer, 2002.

[17] P. Hamburger. Cogwheels of the mind: The story of Venn diagrams, a review. *Mathematical Intelligencer*, pages 36–38, 2005.

[18] P. Hamburger and R. E. Pippert. Venn said it couldn't be done. *Mathematics Magazine*, 73:105–110, 2000.

[19] T. Hastie, R. Tibshirani, and J. H. Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, New York, 2nd edition, 2009.

[20] L. J. Holt, B. B. Tuch, J. Villén, A. D. Johnson, S. P. Gygi, and D. O. Morgan. Global analysis of cdk1 substrate phosphorylation sites provides insights into evolution. *Science*, 325:1682–1686, 2009.

[21] T. Hulsen, J. deVlieg, and W. Alkema. BioVenn a web application for the comparison and visualization of biological lists using area-proportional Venn diagrams. *BMC Genomics*, 9:488, 2008.

[22] N. B. Ivanova, J. T. Dimos, C. Schaniel, J. A. Hackney, K. A. Moore, and I. R. Lemischka. A stem cell molecular signature. *Science*, 298:601–604, 2002.

[23] P. Jaccard. Étude comparative de la distribution florale dans une portion des alpes et des jura. *Bulletin de la Société Vaudoise des Sciences Naturelles*, 37:547–579, 1901.

[24] C. M. M. Junta, P. Sandrin-Garcia, A. L. Fachin-Saltoratto, S. Spano, S. Mello, R. D. R. Oliveira, D. Meyre, M. Rassi, S. Giuliatti, E. Tiemi, T. Sakamoto-Hojo, P. Louzada-Junior, E. Antonio, A. Donadi, and G. A. S. Passos. Differential gene expression of peripheral blood mononuclear cells from rheumatoid arthritis patients may discriminate immunogenetic, pathogenic and treatment features. *Immunology*, 127:365–372, 2008.

[25] H. A. Kestler, A. Müller, T. M. Gress, and M. Buchholz. Generalized Venn diagrams: a new method of visualizing complex genetic set relations. *Bioinformatics*, 21:1592–1595, 2005.

[26] H. A. Kestler, A. Müller, J. M. Kraus, M. Buchholz, T. M. Gress, H. Liu, D. W. Kane, B. Zeeberg, and J. N. Weinstein. VennMaster: Area-proportional Euler diagrams for functional GO analysis of microarrays. *BMC Bioinformatics*, 9, 2008.

[27] D. Klahr. A Monte Carlo investigation of the statistical significance of Kruskal's nonmetric scaling procedure. *Psychometrika*, 34:319–330, 1969.

[28] Y. Koren and L. Carmel. Robust linear dimensionality reduction. *IEEE Transactions on Visualization and Computer Graphics*, 10:459–470, 2004.

[29] M. H. Kutner, C. J. Nachtschiem, W. Wasserman, and J. Neter. *Applied Linear Statistical Models*. Richard D. Irwin, Inc., Homewood, IL, 1996.

[30] M. W. Levine. *Fundamentals of Sensation and Perception*. Oxford University Press, 3rd edition, 2000.

[31] C. Lu, S. S. Tej, S. Luo, C. D. Haudenschild, B. C. Meyers, and P. J. Green. Elucidation of the small RNA component of the transcriptome. *Science*, 309:1567–1569, 2005.

[32] J. Müller M. C. Gambetta, K. Oktaba. Essential role of the glycosyltransferase sxc/ogt in polycomb repression. *Science*, 325:93–96, 2009.

[33] D. T. Odom, N. Zizlsperger, D. B. Gordon, G. W. Bell, N. J. Rinaldi, H. L. Murray, T. L. Volkert, J. Schreiber, P. A. Rolfe, D. K. Gifford, E. Fraenkel, G. I. Bell, and R. A. Young. Control of pancreas and liver gene expression by HNF transcription factors. *Science*, 303:1378–1381, 2004.

[34] S. C. J. Parker, L. Hansen, H. O. Abaan, T. D. Tullius, and E. H. Margulies. Local dna topography correlates with functional noncoding regions of the human genome. *Science*, 326:389–392, 2009.

[35] M. Piroozhnia, V. Nagarajan, and Y. Deng. GeneVenn - a web application for comparing gene lists using Venn diagrams. *Bioinformation*, 1:420–422, 2007.

[36] P. Rodgers, J. Flower, G. Stapleton, and J. Howse. Some results for drawing area proportional Venn3 with convex curves. In *IV '09: Proceedings of the 2009 13th International Conference Information Visualisation*, pages 667–672, Washington, DC, USA, 2009. IEEE Computer Society.

[37] P. Rodgers, L. Zhang, and A. Fish. General Euler diagram generation. In *Diagrams 2008, LNAI 5223*, pages 13–27, Berlin, Germany, 2008. Springer Verlag.

[38] P. Rodgers, L. Zhang, G. Stapleton, and A. Fish. Embedding well-formed Euler diagrams. In *Information Visualisation, 2008. IV '08*, pages 585–593, Berlin, Germany, 2008. Springer Verlag.

[39] F. Ruskey. A survey of Venn diagrams. *Electronic Journal of Combinatorics*, 4, 1997.

[40] P. S. Schnable and D. Ware et al. The b73 maize genome: Complexity, diversity, and dynamics. *Science*, 324:1112–1115, 2009.

[41] P. Simonetto, D. Auber, and D. Archambault. Fully automatic visualisation of overlapping sets. In H-C Hege, I. Hotz, and T. Munzner, editors, *EuroVis 2009 : Eurographics/ IEEE-VGTC Symposium on Visualization 2009*, pages 967–974, Berlin, Germany, 2009. Blackwell.

[42] G. K. Smyth. Limma: linear models for microarray data. In R. Gentleman, V. Carey, S. Dudoit, R. Irizarry, and W. Huber, editors, *Bioinformatics and Computational Biology Solutions using R and Bioconductor*, pages 397–420. Springer, New York, 2005.

[43] J. C. Spall. *Introduction to Stochastic Search and Optimization*. John Wiley & Sons, Inc., New York, NY, USA, 2003.

[44] I. Spence and J.C. Ogilvie. A table of expected values for random rankings in nonmetric multidimensional scaling. *Multivariate Behavioral Research*, 8:511–517, 1973.

[45] I. Spence and F. Young. Monte carlo studies in nonmetric scaling. *Psychometrika*, 43:115–117, 1978.

[46] G. Stapleton, J. Howse, and P. Rodgers. A graph theoretic approach to general Euler diagram drawing. *Theor. Computer Science*, 411(1):91–112, 2010.

[47] G. Stapleton, P. Rodgers, J. Howse, and L. Zhang. Inductively generating Euler diagrams. *IEEE Transactions on Visualization and Computer Graphics*, 29, 2010.

[48] H.H. Stenson and R.L. Knoll. Goodness of fit for random rankings in Kruskal's nonmetric scaling procedure. *Psychological Bulletin*, 71:122–126, 1969.

[49] W. Torgerson. Multidimensional scaling: I. theory and method. *Psychometrika*, 17(4):401–419, December 1952.

[50] W. W. Torgerson. Scaling and psychometrika: Spatial and alternative representations of similarity data. *Psychometrika*, 51:57–63, 1986.

[51] J. Venn. On the diagrammatic and mechanical representation of propositions and reasonings. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 9:1–18, 1880.

[52] D. A. Walsh, E. Zaikova, C. G. Howes, Y. C. Song, J. J. Wright, S. G. Tringe, P. D. Tortell, and S. J. Hallam. Metagenome of a versatile chemolithoautotroph from expanding oceanic dead zones. *Science*, 326:578–582, 2009.