# The Multi-Dimensional Landscape of Graph Drawing Metrics

Gavin J. Mooney, Helen C. Purchase, Michael Wybrow, and Stephen G. Kobourov

**Abstract**—Any graph drawing can be characterised by a range of computational aesthetic metrics. For example, a given drawing might be described as having eight crossings, a mean angular resolution of 0.34, and an edge orthogonality value of 0.72. However, without knowing the distribution of these metrics it is hard to compare the quality of drawings of different graphs, nor know whether a given drawing is typical or an outlier within the space of all possible drawings. This paper explores the range and distribution of ten normalised graph drawing layout metrics, based on graphs created by six graph generation algorithms and drawings created by six popular layout algorithms. We include the "Rome" and "North" graph repositories in our analysis. Our exploration of the multi-dimensional aesthetics space allows for comparisons between the graph drawing algorithms, highlighting those that cover larger or smaller volumes of the aesthetics space. We calculate the correlation coefficients between the metrics, indicating those that may conflict with each other (negatively correlated), and those that may be redundant (positively correlated). Our results will be useful as the basis for simulated annealing or gradient descent layout algorithms, for identifying the best layout algorithms for producing a specified combination and range of aesthetics, and for informing experimental controls in human empirical studies.

**Index Terms**—Graph layout aesthetics, Graph metrics, Graph layout algorithms

## 1 Introduction

Research papers in the graph drawing literature tend to selectively choose graphs and graph drawings to illustrate their results; these are chosen specifically for the purposes of demonstrating the worth of the research contribution. What is not clear is how representative these chosen graphs and graph drawings are—within the space of all possible graphs and possible graph drawings. A multi-dimensional space of all possible graphs can be envisioned, where the axes are common graph metrics: density, centrality, diameter, etc. Similarly, the multi-dimensional space of all possible graph drawings would have axes based on the common layout principles (or 'aesthetics'): prevalence of edge crossings, number of bends, average angle of edge crossings, etc. This paper considers the latter: if we have a graph drawing and its set of aesthetic measurements, where does it lie in this multi-dimensional space? Are some (or all) of its metric values at an extreme, or could this drawing be considered 'typical'? If typical, we would expect all its metric values to lie around their respective medians in the distributions. Determining the distributions of such metrics requires the generation of a large number of graph drawings, based on a large number of graphs. The more varied the graphs and the more varied the drawings, the better, since we are more likely to get coverage of the multi-dimensional space.

We have tackled the problem by using a range of random graph generation algorithms (creating graphs of different sizes) and a range of common graph layout algorithms, producing a total of 447,934 graph drawings. We then produce descriptive statistics in the form of distribution charts and correlations, presenting an overall summary of these graph drawings with respect to common quantifiable layout principles. The ten aesthetic criteria considered in our paper span the entire spectrum of measurable layout statistics commonly used in related work (e.g., [6, 27, 29, 35, 36, 40]).

### 1.1 Motivation

At the very heart of this endeavour is the fundamental ability to compare the aesthetic worth of two graph drawings when the graphs are a different structure or size. By defining distributions along each of the aesthetic metric axes, the ability to perform three common research tasks becomes easier: generating graph drawings through optimisation, defining graph drawing stimuli for experimental purposes, and identifying the most appropriate layout algorithm for satisfying particular layout metrics. We explain each below.

First, simulated annealing, hill-climbing or gradient descent systems can be used to generate graph drawings. Typically, the objective function for these methods tries to optimise one or more layout features, aiming, for example, for zero edge crossings as well as for maximum angular resolution. Metric distributions and correlations between layout features can indicate where aiming for extremes is unreasonable, and where it is better to aim for a range of values instead. If the metric distribution is skewed to the bottom (the median is less than 0.5), then specifying an extreme value of 1.0 is unreasonable. Correlations can also show us where it would be foolish to attempt to maximise two metrics simultaneously. If the edge length metric is negatively correlated with the extent to which edges align to an underlying grid, then it does not make sense to try to create a graph drawing that maximises both, and including both in the objective function will increase the likelihood of local minima. Similarly, if two metrics are highly positively correlated, then only one needs to be included in the objective function.

Second, human experiments with graph drawings require the production of visual stimuli. Since we know that the extent of layout features in a graph drawing affects human perception [29], experimenters need to know where their experimental stimuli lie within the metrics distributions landscape. For example, a drawing has an edge length variation metric value of 0.25; is this a 'high' value or a 'low' one, when considered within the whole population of graph drawings? We need to know the distribution and typical values of the edge length distribution metric to answer this question. If it is impossible (or highly unlikely) that a metric value will exceed 0.6, then a drawing that measures 0.5 for this metric can be considered 'good'; conversely, if the median metric value for another metric is 0.8, then 0.5 would be considered poor. These relative judgements on the suitability of graph drawings in human experiments are particularly important when the layout factors need to be deliberately varied or controlled.

Finally, by looking at the different metric landscapes created by different layout algorithms, we can see to what extent a layout algorithm covers the multi-dimensional metric space. Some algorithms will cover some (and more) metrics better than others. Therefore, if there is a requirement that a given graph be drawn so as to maximise a set of particular metrics, the landscape can indicate which layout algorithm would be best to use.

- Gavin J. Mooney, Helen C. Purchase and Michael Wybrow are with Monash University. E-mail: [firstname].[lastname]@monash.edu.
- Stephen G. Kobourov is with University of Arizona. E-mail: kobourov@cs.arizona.edu

Manuscript received xx xxx. 201x; accepted xx xxx. 201x. Date of Publication xx xxx. 201x; date of current version xx xxx. 201x. For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org. Digital Object Identifier: xx.xxxx/TVCG.201x.xxxxxxx

## 2 Background

### 2.1 Graph Generation Algorithms & Common Repositories

Researchers often require a set of graphs for their research, e.g., for graph mining or graph drawing. Finding appropriate real-world graphs may be impossible or existing repositories may not be suitable, justifying the use of synthetically generated graphs. We use six graph generators commonly found in the literature (see Sec. 3 for implementation details).

- Erdős–Rényi (ER) generates random graphs, with a given fixed probability of any two nodes being connected [10].
- Barabási–Albert (BBA) generates random graphs with node degrees following a power law, to model real-world networks [2]. New nodes added to the graph are preferentially connected to nodes with a high degree.
- Newman–Watts–Strogatz (NWS) generates small world graphs, that is, graphs with high clustering but small path lengths, commonly used in social network analysis [24, 37].
- Lancichinetti–Fortunato–Radicchi (LFR) generates graphs with community structures, attempting to model social networks; it requires that the minimum and maximum community size be specified [23].
- Stochastic Block Model (SBM) generates random clustered graphs, where nodes are initially assigned to communities and connections between nodes in two different communities are determined by a given probability [15].
- Random Geometric (GEO) generates graphs by placing nodes uniformly at random within a unit square, and adding an edge if two nodes are within a pre-specified distance of each other [26].

These graph generation algorithms are frequently used in experiments attempting to model real-world graphs. ER models edge density, NWS models edge density and small world properties, BBA models edge density, small world properties, and power-law degrees, etc.

We also include the North [7] and Rome [8] graph collections, since they are frequently referred to and used in the graph drawing literature.

- The Rome graphs: A set of 11,582 graphs constructed for experimental analysis of graph drawing algorithms. The graphs were created by taking 112 real-world graphs and generating variations of them.
- The North graphs: A collection of directed graphs by North at AT&T Bell Labs collected from Draw DAG, an e-mail graph drawing service [20].

### 2.2 Graph Drawing Algorithms

In choosing our layout algorithms, we identified commonly used methods, covering a range of algorithmic categories; (Sec. 3 for implementation details).

- Fructerman & Reingold. (FR) This is a standard force-directed method, with attractive forces between adjacent nodes and repulsive forces between all node pairs [11].
- Kamada & Kawai. (KK) A force-directed method that attempts to place pairs of nodes at Euclidean distances that are proportional to the underlying graph-theoretic distances [17]. This is a global graph embedding algorithm, similar to multi-dimensional scaling embedding [22, 31].
- DRGraph. (DRG) This is a local graph embedding method, where only the distances between nearby nodes affect the layout [42]. Local algorithms attempt to preserve the local neighbourhoods, while global algorithms such as MDS and K K attempt to retain all pairwise distances. DRG provides a faster implementation of the original t-SNET graph layout method [21].
- Sugiyama. (Sugi) This is a standard graph layout method that places nodes layer by layer. It is frequently used for trees, hierarchies and directed acyclic graphs, but can be used for other types of graphs [32].
- Human-like Orthogonal Network Layout. (HOLA) This is an orthogonal graph layout method where vertices lie on integer grid points and edges follow grid lines. Unlike prior algorithms of this type which optimise compactness or bends, HOLA was informed by human empirical studies [18].
- Circular. (Circ) This is a common layout algorithm which places nodes randomly around the circumference of a circle.
- Random. (Ran) While not a layout algorithm per se, we also consider a random layout as a baseline, since we assume it performs worse than layout algorithms designed for readability.

### 2.3 Graph Drawing Metrics

Purchase [28] defined several graph layout metrics; we use some of them, improve others, and add new ones. All metrics are normalised to lie between 0 and 1, where 1 represents the extreme that is intuitively assumed to be 'good', for example, a low number of edge crossings. Such normalisation is essential for two reasons: so as to define the bounds of our multi-dimensional metric space, and to facilitate comparison between drawings of different graphs. All the metrics apply to straight line drawings of connected graphs, with no self-loops or multiple edges, and we apply the same initial transforms as described in Purchase [28].

A graph drawing, denoted $D(G)$ is the assignment of every node of a graph, $G$, to a pair of coordinates on the 2-D plane. The position of a node $u_i$ is $(x_i, y_i)$. Edges are drawn as polylines or straight lines between their nodes. 'bends promotion' converts any polyline drawing $D(G)$ into a straight line drawing $D'(G)$ by introducing dummy nodes at the bends of the polyline edges in $D(G)$ and replacing the associated edge segments with new straight-line edges in $D'(G)$ [28]. Similarly 'crosses promotion' removes edge crossings by introducing a dummy node at each edge crossing, and replacing the two edges with four shorter ones. A drawing with both bends promotion and crosses promotion applied is denoted $D''(G)$. The number of nodes and edges in these drawings are denoted by $n'$, $m'$, and $n''$, $m''$ for bends and crosses promotion respectively.

Here we investigate ten metrics which form a focus of interest in a series of papers about graph statistics [3, 4] and graph drawing statistics [1] detailed formulae are available in the supplemental materials.

- Angular Resolution (AR). The larger the angle between two adjacent edges, the easier it is to distinguish between them and to follow them visually to their end nodes [25]. After calculating the angular deviation of each edge from the ideal angle (based on node degree), the AR metric calculates the mean over all nodes (excluding degree-1 nodes).
- Aspect Ratio (Asp). Too wide or too tall layouts are harder to display and work with. The Aspect Ratio is the ratio of the height of the drawing's bounding box to its width (or vice versa, depending on which is greater).
- Crossing Angle (CA). Edges which cross at 90-degree angles are more readable [36]. Using the same approach as Angular Resolution we measure the average deviation of angles at which edges cross from an ideal.
- Edge Crossings (EC). Edge crossings make it difficult to follow paths visually [25, 27, 29]. The Edge Crossings metric counts the number of edge crossings and normalises with respect to an estimate of the maximum number of crossings.
- Edge Length Deviation (EL). Edge lengths should be uniform for unweighted graphs [25]. Following Ahmed et al. [1] this metric defines an 'ideal' edge length as the mean of all edges, and calculates the mean deviation of all edges from this ideal.

- Edge Orthogonality (EO). Aligning edges to a grid is a popular layout principle for some domains (e.g., electrical circuits). The Edge Orthogonality metric takes the mean of the angular deviation of all edges from the horizontal or vertical axis (whichever is closest).
- Gabriel Ratio (GR). Having nodes too close to edges makes them hard to distinguish [25]. The principle underlying a Gabriel graph is that no node is placed within the circle formed by using any edge as its diameter. Our Gabriel Ratio metric is calculated by determining how many nodes violate this rule, normalised against the number of nodes that could do so.
- Neighbourhood Preservation (NP). Nodes that are close in terms of graph-theoretic distance should be near each other in the layout. The Neighbourhood Preservation metric [1] uses the Jaccard similarity index between the k nearest neighbours in graph-theoretical sense and in the layout.
- Node Resolution (NR). Nodes that are too close together are difficult to distinguish, but the scale of the drawing might require such closeness. The Node Resolution metric is the ratio between the smallest distance between two nodes and the largest distance between two nodes.
- Node Uniformity (NU). Nodes should be well distributed in the layout, so that they are easier to distinguish [25]. The Node Uniformity metric measures the distribution of nodes in the bounding box, by splitting it into grid cells based on the number of nodes in the graph, counting the number of nodes in each cell, and comparing them with an ideal distribution.

We notably exclude stress as a metric due to the difficulty in creating a normalised metric to quantify it (so that graphs of different sizes can be compared). Unlike most other measures, stress is affected by geometric scaling [38]. This poses a difficult problem when comparing outputs from different algorithms (where one algorithm may draw a graph within a unit square and another may position the nodes thousands of units apart).

### 2.4 Prior Work using Metrics

Many graph drawing algorithms optimise some kind of aesthetic metric. For example, the Kamada-Kawai method [17] explicitly optimises global distance preservation, t-SNET [21] and DRGraph [42] optimise local distance preservation, and Radermacher et al. [30] optimise edge crossings. There are also approaches which more explicitly optimise multiple metrics, such as Davidson and Harel's [5] graph drawing by simulated annealing, which optimises edge crossings, edge lengths, and node uniformity. Wang et al. [35] optimise ideal edge directions in addition to ideal edge lengths. Devkota et al. [6] minimise edge crossings and maximize crossing angles. Eades et al. [9] optimise multiple geometric criteria, including the Gabriel graph property. More recently, Ahmed et al. [1] propose a general framework for multi-criteria optimisation using stochastic gradient descent (stress, neighbourhood preservation, edge lengths, the Gabriel graph property, edge crossings, aspect ratio, crossing angle, and angular resolution).

Aesthetic metrics have also been employed in human experiments to evaluate the subjective perception of graph layouts. These experiments involve presenting participants with different graph layouts and collecting their feedback on the aesthetics, readability, and overall quality of the visualisations. By incorporating human evaluations, researchers can validate the effectiveness of aesthetic metrics and refine their algorithms accordingly [27, 36].

## 3 Methodology

### 3.1 Generating Graphs

We generate graphs using the six techniques described in Sec. 2.1. For each method, we generate 60 graphs for each size in the range of $n = [10, 20..110]$ nodes, for 660 per method. We limit the graphs we generate to those with $n \leq m \leq 3n$ edges to create sparse graphs. We limit our data set to small and sparse graphs, such as those in the Rome and North graph collections, as they are predominant in human-subject studies [41]. Furthermore, Ghoniem et al. [12] argue that the readability of node-link diagrams deteriorates as the size and density of the graph increases. We generate graphs until we have one which is connected and has an acceptable number of edges. Graphs within each generator are also checked for isomorphism and discarded if they are isomorphic with a previously generated graph.

The ER, BBA, NWS, SBM and GEO graphs were generated using the respective NetworkX implementations [13]. The choice of parameters is made in order to allow the generators to create the sparse graphs we desired, whilst ensuring the graph remained connected. Randomness is incorporated to extend the range of graphs which could be generated. The parameters were decided as follows (initial probabilities are chosen at random in the range [0,1]):

- ER: Each edge is added with probability $p \in (0, 1)$. If the generated graph is not connected we increase p, and if the graph had too many edges we decrease p.
- BBA: The number of edges to attach from a new node to existing nodes, m, is a random integer in the range $[1, 4]$.
- NWS: The number of nearest neighbours joined in the ring topology, k, is a random integer in the range $[2, 8]$. The probability of adding a new edge for each edge, p, is set to 0.5.
- SBM: The list of floats specifying edge densities between different groups, p, was generated by first creating a random list of integers (and a random size of the list) which sum to the number of nodes, in order to have random group sizes. The probabilities between groups were then assigned randomly.
- GEO: The dimension d is set to 2. The radius distance threshold, r, was chosen as a random float in the range $[0, 1]$ (since the implementation places nodes randomly in a unit square, for $d = 2$). As with ER we increase or decrease r until we get a graph which satisfies our requirements.
- LFR: We use the implementation by Lancichinetti and Fortunato,[1] setting the average degree to 4, the maximum degree to 8, the mixing parameter for the weights to 0.1, the exponent for the weight distributions to 1, and the minimum community size to 2.

In total, our data set consists of 16,768 graphs.

### 3.2 Drawing Graphs and Calculating Metrics

The KK, FR, Circular, and Random drawings were created using the respective implementations in NetworkX [13]. The HOLA drawings were created using the Adaptagrams implementation with SWIG for Python.[2] The DRGraph drawings were created using the original implementation and default parameters.[3] Each layout algorithm generates five drawings for each graph in our data set, with the exception of Circular and HOLA, which only generate one. Specifically, we use one drawing:

- from Circular, as multiple runs produce the same layout
- from HOLA, due to its long execution time

We use the default parameters for each layout algorithm to ensure valid comparison.

Due to the size and structure of some graphs, some layout algorithms produce the same drawings. Such duplicates are determined by checking for duplicate metric values: if two graph drawings have the same value for all ten metrics, then they are assumed to be the same drawing. We found 4,802 such cases and excluded these duplicates from our data set. As a result, the total number of graph drawings in our experiment ended up being 447,934.

All aesthetic metrics were implemented in Python using standard libraries and the packages NetworkX [13], NumPy [14], and SciPy

[1] https://github.com/eXascaleInfolab/LFR-Benchmark_UndirWeightOvp

[2] https://github.com/mjwybrow/adaptagrams/

[3] https://github.com/ZJUVAI/DRGraph

[34]. The detailed descriptions of each metric can be found in the supplemental materials. A Python script was used to read and calculate the aesthetic metrics for each graph drawing, saving the results in a CSV file. We then used Pandas [33, 39] and matplotlib [16] to analyse the data and generate the visualisations.

For the polyline HOLA drawings, the following aesthetic metrics were calculated before performing bends promotion: NP, NR, NU. Then Asp, AR, CA, GR, EC, EL, and EO were calculated after performing bends promotion. An implication of this is that EL is being calculated for edge segments (where bends occur) rather than the full edge.

## 4 The Metric Landscape: Description and Discussion

The metric landscape is presented in Fig. 1, Fig. 2, Fig. 3, and Table 1.

### 4.1 Distributions

Fig. 2 presents the distributions separated by graphs that have been randomly generated by our 6 algorithms, and by those which came from the North and Rome graph sets. We can compare the distributions to ensure that the large size of the Rome graph set does not bias the distributions in Fig. 1. We can see that the distributions for the generated and curated graphs are similar, with the exception of Angular Resolution. Our observations of the distributions relate to each metric in Fig. 1.

Sugiyama performs worse than the other layouts for Angular Resolution; this is not unexpected as the layering technique creates edges with acute angles to nodes on nearby layers. HOLA also performs poorly for Angular Resolution, due to the fact that edges leave nodes at orthogonal angles, which inevitably yields poor Angular Resolution for high degree nodes. The distribution is roughly bi-modal for all algorithms (including random to some extent); further work could determine whether this is due to the secondary factors of size or graph generator.

For Aspect Ratio, all layouts except Sugiyama have distributions skewed towards the higher end. This is not unexpected: the layered approach creates a bounding box that is typically wide and short.

In the Crossing Angle distributions, Sugiyama also performs worse, since the layers create acute angles. The spike at (or close to) 1.0 reflects drawings with few (or no) edge crossings. We would expect HOLA to perform well here. The fact that it does not could be a consequence of bends promotion and requires further investigation.

All layouts, except from Circular, are seemingly successful in minimising Edge Crossings and show distributions skewed towards 1.0. While this phenomenon is somewhat due to the overestimation of the maximum number of edge crossings as used in the metric, we note that the extent of the distribution for Random demonstrates that the metric is capable of taking on nearly the full range of values. HOLA is particularly successful here as the algorithm deliberately routes edges to avoid crossings.

For the Edge Length metric we expect Sugiyama to perform badly (a variety of edge lengths to children on adjoining layers, and long edges spanning multiple layers). DRGraph separates nodes that are not close in a neighbourhood sense and likely produces a mixture of short and long edges differing from the 'ideal'. HOLA performs well here since the metric is calculated on edge segments.

As expected, HOLA performs well on Edge Orthogonality; the other layouts have distributions similar to the Random layout. This shows that non-orthogonal layouts are not better than random when considering Edge Orthogonality.

The Gabriel Ratio distributions are skewed to the top for all layouts. Like the Edge Crossings metric, this is partly due to overestimation of the upper bound; however, the lower distribution for Random layouts suggests that this is also because the layouts successfully maintain the Gabriel distance throughout the drawing.

The Neighbourhood Preservation metric is better for DRG than others since they work on the same principle. HOLA also performs slightly better on this metric, likely due to the way it lays out trees independently and places them back into the core layout.

Node Resolution tells us very little as most drawings have a small value. While this does not suggest that the two closest nodes cannot be distinguished visually, it does suggest that this metric is not particularly useful.

Each layout performs similarly for the Node Uniformity metric, except the Random layout, which has low Node Uniformity values (as expected).

We note that some metrics show multi-modal distributions. The bi-modal distribution for the Crossing Angle metric can be attributed to the presence of several graph drawings with no crossings. In particular, small planar graphs are often drawn with very few crossings by most layout algorithms and this accounts for one of the two modes. The other mode covers drawings with crossings where there's more variability in the layouts by different algorithms. The other multi-modal patterns appear to be attributable to the different graph generators and graph sizes used, as seen when the analysis and visualisation are computed for each generator (see supplemental material). For example, the Barabasi-Albert and Newman-Watts-Strogatz generators exhibit bi-modal distributions for Aspect Ratio, which we attribute to the varied parameters used when generating the graphs. On the other hand, the multi-modal distribution of Node Uniformity on the Circular layout can be attributed to the different (discrete) graph sizes.

### 4.2 Correlations

Fig. 3 presents the correlations between pairs of metrics.

We could interpret the correlations between pairs of properties in the random layouts as a baseline. Indeed, there are only two correlations greater than 0.5 and no negative correlations smaller than -0.5. The high correlation (0.652) between EC and GR could be due to the similar way in which these metrics are calculated. The high correlation (0.585) between NP and NR is unclear.

The standard force-directed method FR has four more correlations above 0.5 and still no negative correlations stronger than -0.5. Specifically (AR, GR), (AR, EL), (AR, EC), (NR, CA). This confirms some folkloric knowledge that for small graphs, force-directed methods tend to distribute edges well (good AR), and this is correlated with uniform edge lengths (EL), with good Gabriel Ratio (GR), and few edge crossings (EC).

The global embedding method (KK) performs similarly to MDS layouts that optimise stress. There are ten correlations stronger than 0.5 and no notable negative correlations. For small graphs the KK algorithm performs similarly to force-directed methods such as FR, confirmed by several of the correlations involving AR and EL.

The local embedding method (DRGraph) tries to preserve local neighbourhoods. It is the only layout that does not show a strong correlation between NR and NP.

The standard layered method (Sugiyama), adds several correlations above 0.5, but more notably two negative correlations below -0.5: (EO, CA) and (EO, NR). In fact, every property is negatively correlated with Edge Orthogonality (EO), as forcing orthogonality negatively impacts every other property.

HOLA shows strong positive correlations between (AR, EL), indicating that similar edge lengths may be necessary for good AR in orthogonal drawings. There are strong negative correlations between (AR, EC), and (GR, NR), which could be a result of the generally high values for GR and low values for NR.

We observe that the correlations between EC and GR are similar for all algorithms apart from HOLA; this can be useful when optimising one or the other of these two criteria. We also note major differences between correlations across the different algorithms. While some of these differences are not clear and are worth exploring, others make sense. For example, the correlation between GR and EL in DRGraph is $-0.36$ while in KK it is 0.68: the local clustering approach of DRGaph implicitly encourages edge length variation, unlike KK which attempts to maintain similar edge lengths.

### 4.3 Typical, Best and Worst Graph Drawings

The 'typical' drawings in Table 2 have metric values close to the medians of our specific ten metrics. It should not be surprising that they do not look very 'nice'; most published graph drawings are the output of algorithms that attempt to optimise criteria—rather than just find a

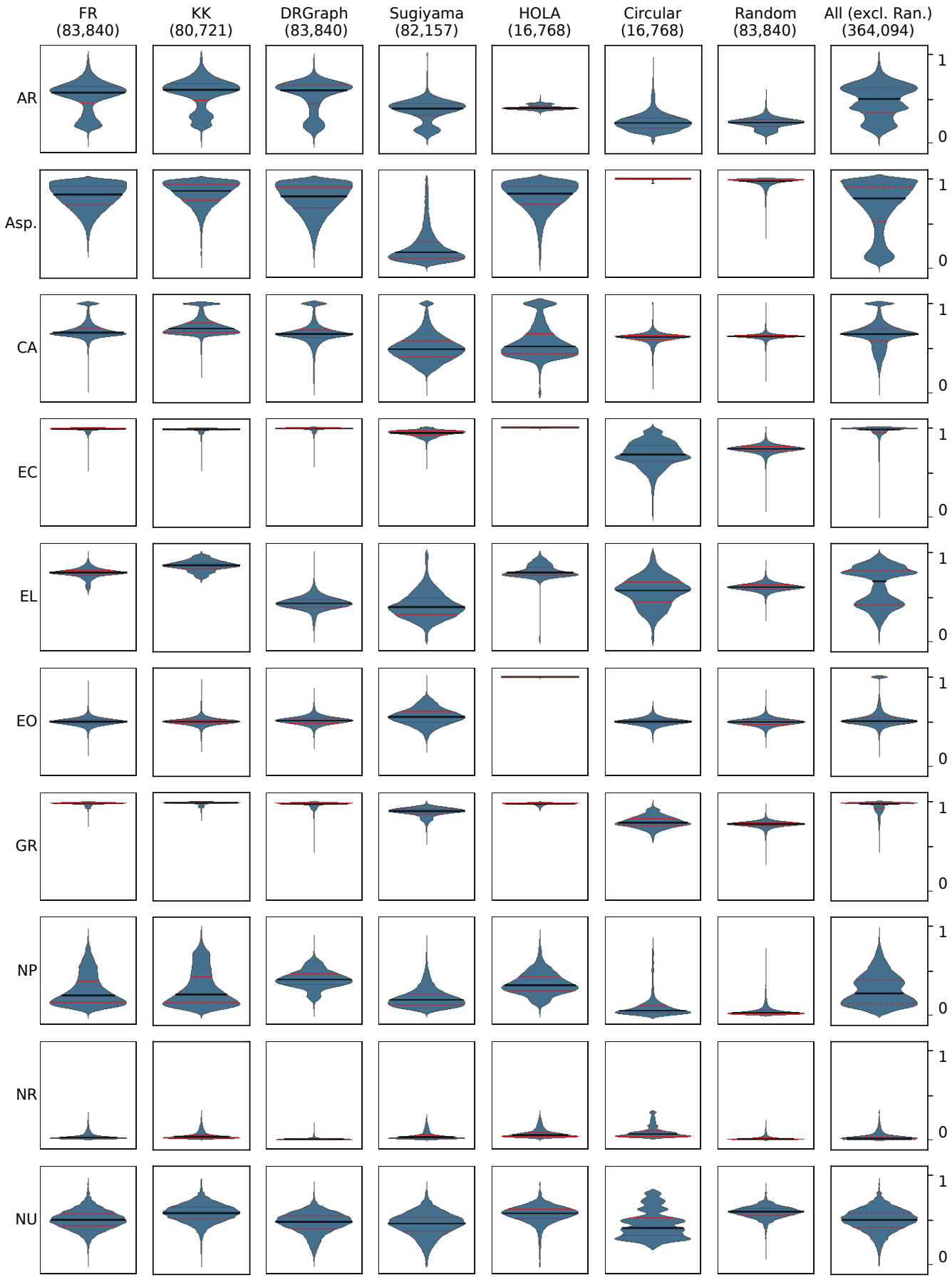



Fig. 1: The distributions for each metric, and each layout algorithm. The value in brackets is the number of drawings for each layout. The medians and inter-quartile ranges for each distribution are shown in Table 1.

| | | FR | KK | DRGraph | Sugiyama | HOLA | Circular | Random | All (excl. Random) |
|---|---|---|---|---|---|---|---|---|---|
| Angular Resolution | 1st | 0.457 | 0.493 | 0.447 | 0.32 | 0.389 | 0.198 | 0.178 | 0.355 |
| | Med. | 0.571 | 0.607 | 0.601 | 0.398 | 0.401 | 0.24 | 0.232 | 0.508 |
| | 3rd | 0.64 | 0.678 | 0.661 | 0.452 | 0.425 | 0.271 | 0.288 | 0.629 |
| Aspect Ratio | 1st | 0.708 | 0.763 | 0.673 | 0.118 | 0.714 | 0.949 | 0.998 | 0.527 |
| | Med. | 0.823 | 0.863 | 0.806 | 0.178 | 0.833 | 0.976 | 0.999 | 0.784 |
| | 3rd | 0.913 | 0.935 | 0.907 | 0.293 | 0.92 | 0.99 | 1 | 0.911 |
| Crossing Angle | 1st | 0.65 | 0.681 | 0.623 | 0.405 | 0.442 | 0.621 | 0.599 | 0.586 |
| | Med. | 0.677 | 0.722 | 0.659 | 0.492 | 0.521 | 0.634 | 0.626 | 0.662 |
| | 3rd | 0.729 | 0.786 | 0.709 | 0.589 | 0.658 | 0.644 | 0.644 | 0.723 |
| Edge Crossings | 1st | 0.978 | 0.983 | 0.984 | 0.918 | 0.998 | 0.738 | 0.621 | 0.952 |
| | Med. | 0.986 | 0.989 | 0.991 | 0.943 | 0.999 | 0.763 | 0.699 | 0.985 |
| | 3rd | 0.991 | 0.994 | 0.995 | 0.96 | 0.999 | 0.786 | 0.8 | 0.993 |
| Edge Lengths | 1st | 0.747 | 0.815 | 0.39 | 0.306 | 0.742 | 0.586 | 0.446 | 0.423 |
| | Med. | 0.772 | 0.852 | 0.43 | 0.389 | 0.772 | 0.61 | 0.572 | 0.689 |
| | 3rd | 0.796 | 0.889 | 0.474 | 0.492 | 0.834 | 0.635 | 0.667 | 0.803 |
| Edge Orthogonality | 1st | 0.477 | 0.48 | 0.488 | 0.495 | 1 | 0.473 | 0.477 | 0.484 |
| | Med. | 0.5 | 0.501 | 0.513 | 0.553 | 1 | 0.497 | 0.5 | 0.511 |
| | 3rd | 0.523 | 0.521 | 0.539 | 0.614 | 1 | 0.521 | 0.524 | 0.549 |
| Gabriel Ratio | 1st | 0.975 | 0.982 | 0.964 | 0.868 | 0.974 | 0.732 | 0.733 | 0.922 |
| | Med. | 0.982 | 0.988 | 0.979 | 0.896 | 0.98 | 0.751 | 0.768 | 0.977 |
| | 3rd | 0.988 | 0.993 | 0.985 | 0.919 | 0.985 | 0.767 | 0.814 | 0.987 |
| Neighbourhood Preservation | 1st | 0.141 | 0.137 | 0.342 | 0.116 | 0.27 | 0.016 | 0.025 | 0.144 |
| | Med. | 0.221 | 0.234 | 0.398 | 0.17 | 0.336 | 0.028 | 0.05 | 0.255 |
| | 3rd | 0.381 | 0.433 | 0.464 | 0.24 | 0.429 | 0.05 | 0.103 | 0.404 |
| Node Resolution | 1st | 0.018 | 0.026 | 0.003 | 0.018 | 0.038 | 0.006 | 0.042 | 0.014 |
| | Med. | 0.028 | 0.039 | 0.007 | 0.03 | 0.056 | 0.011 | 0.065 | 0.028 |
| | 3rd | 0.045 | 0.061 | 0.012 | 0.052 | 0.084 | 0.021 | 0.101 | 0.05 |
| Node Uniformity | 1st | 0.431 | 0.516 | 0.404 | 0.371 | 0.517 | 0.551 | 0.325 | 0.424 |
| | Med. | 0.5 | 0.58 | 0.479 | 0.459 | 0.573 | 0.591 | 0.408 | 0.506 |
| | 3rd | 0.57 | 0.643 | 0.545 | 0.536 | 0.624 | 0.627 | 0.52 | 0.584 |

Table 1: 1st Quartile, Median, and 3rd Quartile for each metric and layout. The 'All' column excludes random layouts.

'middle-ground'. We note that the largest graph appears to have two visual sections: a dense untidy center surrounded by several neatly placed leaf nodes.

The first two of the 'best' drawings (Table 2) are unsurprising—a tree and a planar graph—noting that these are both from the curated graph repositories. The larger graph drawing is surprising; over all ten metrics, the best possible mean was only 0.623, and the visual quality of the drawing attests to this: it is not what we might call a 'nice' drawing. We conclude that for larger graphs (n= around 100), attempting to satisfy all ten metrics would be ambitious.

For the 'worst' diagrams (Table 2), we particularly note that the Angular Resolution metric is poor for each of them, and it is not surprising that for the larger graph we get a typical 'hairball'.

## 5 Conclusion

This data serves as a valuable resource for researchers seeking to evaluate and compare the quality of graph drawings, allowing them to position graph drawings in a metric landscape.

For graph layout approaches based on optimisation (e.g., simulated annealing, gradient descent), the distributions suggest appropriate target values. If, for example, the aim is to maximise both the Crossing Angle and Gabriel Ratio metrics, we can look at the median and quartiles for these two metrics: (CA:[0.586, 0.662, 0.723]; GR:[0.922, 0.977, 0.987]). Thus, while it might make sense for the target GR value to be set at 1.0 in the optimisation process, 1.0 would be an unreasonable target for CA (when perhaps a value of 0.8 might be more appropriate). We can also see from the correlation matrix that it would be foolish to also include maximising Edge Orthogonality in this same optimisation, since it has a negative correlation with Crossing Angle (-0.3).

When preparing experimental stimuli intended to compare the comprehension effect of a particular layout aesthetic, controlling the other known metrics is important. The first column in Table 3 shows a set of three stimuli designed to test the effect of the Edge Crossing metric: the Edge Crossing metric has three values (which we know are roughly 'low', 'medium', and 'high', since we know the metric distribution), but note that the Edge Orthogonality metric also has L/M/H values for these same three stimuli. This means that if the 'high' stimulus pro-duces better results than the other two, it could be because of the Edge Orthogonality variation, rather than the variation in the Edge Crossing metric, thus, a 'confounding factor'. The second column in Table 3

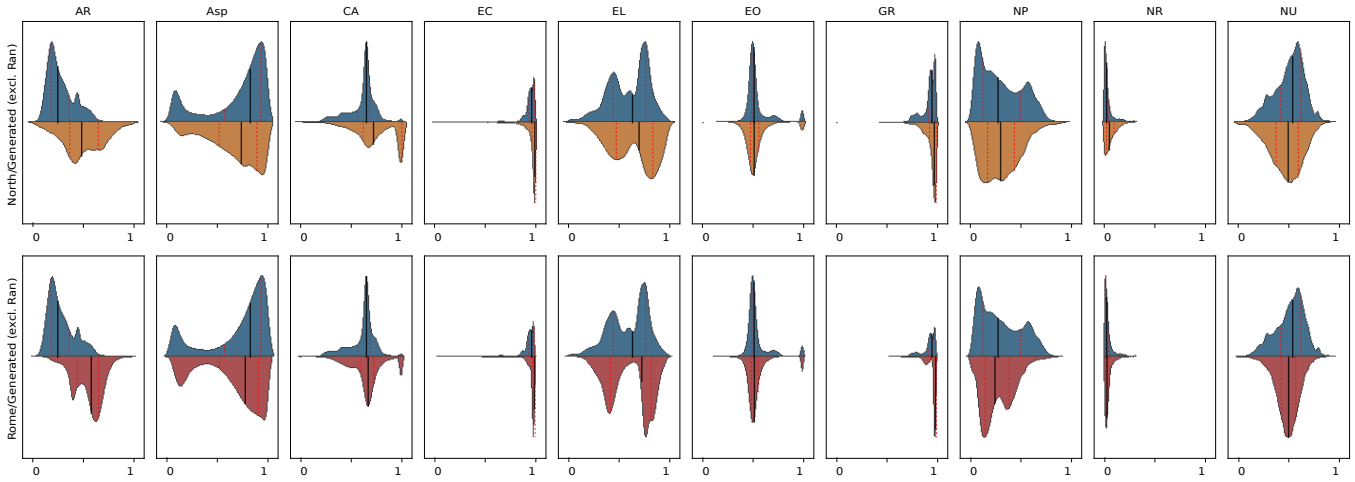



Fig. 2: Violin plots for each metric, across all drawings, excluding Random. The first row shows the distributions for the generated graphs (blue) (excluding North and Rome graphs, and all Random layouts) compared to the distributions for the North (orange) drawings (excluding Random); the same is shown for the second row with the Rome drawings (red). The data includes 249,530 drawings created from Rome graphs, 27,825 drawings created from North graphs, and 86,739 drawings created from the generated graphs.

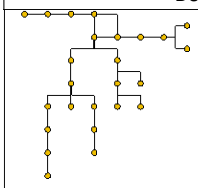
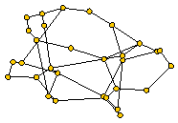
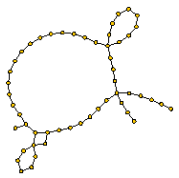
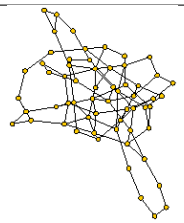
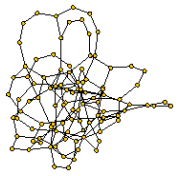
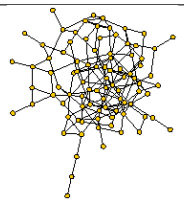
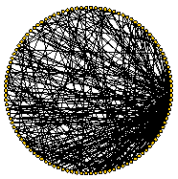
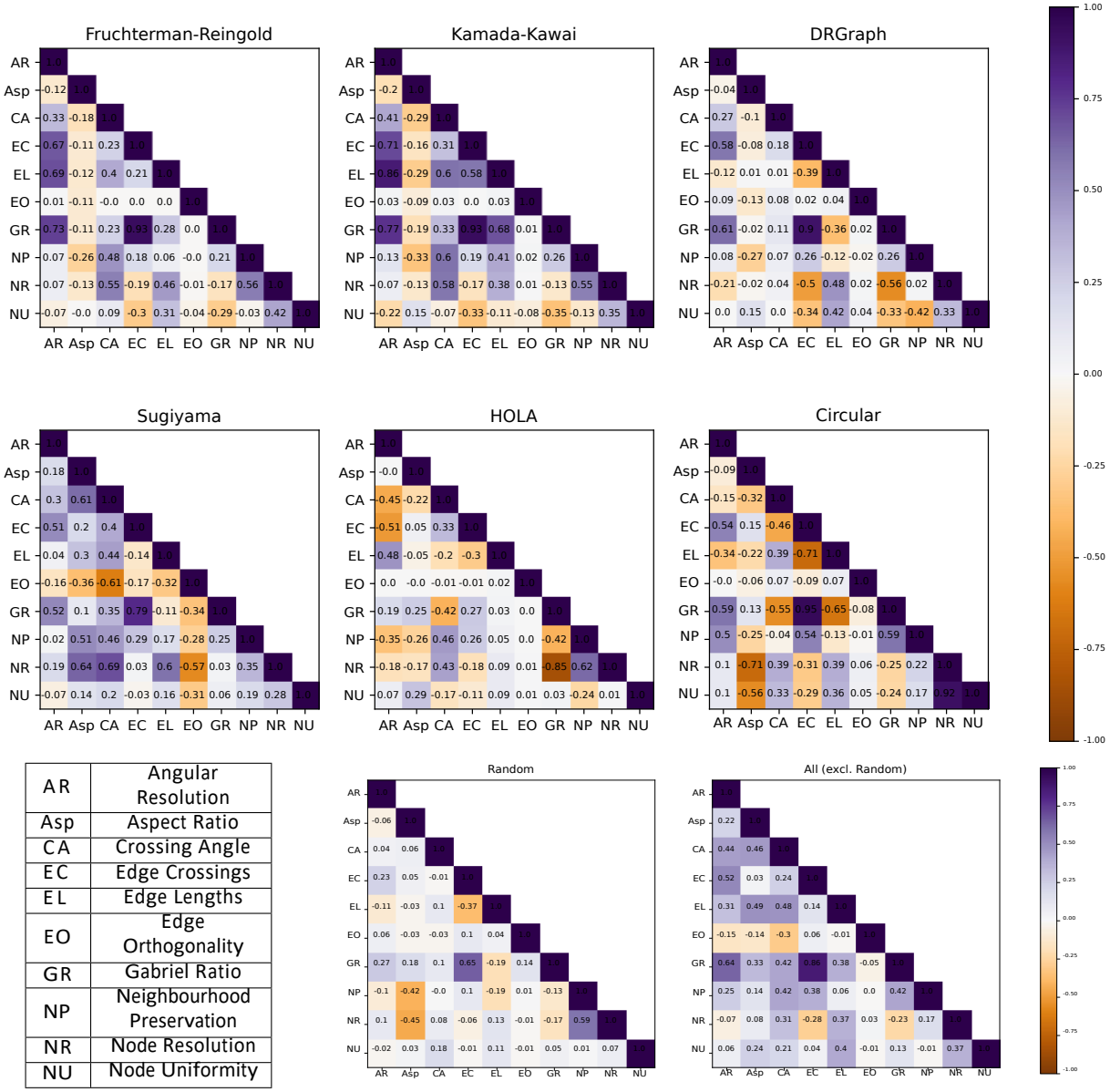
| Best | Typical | Worst |
|---|---|---|
|  Nodes: 25<br>Edges: 25<br>Type: Rome<br>Layout: HOLA<br>Mean: 0.762 |  Nodes: 30<br>Edges: 44<br>Type: NWS<br>Layout: DRG<br>Mean: 0.593 |  Nodes: 10<br>Edges: 24<br>Type: GEO<br>Layout: DRG<br>Mean: 0.443 |
|  Nodes: 52<br>Edges: 55<br>Type: North<br>Layout: K K<br>Mean: 0.777 |  Nodes: 60<br>Edges: 93<br>Type: NWS<br>Layout: FR<br>Mean: 0.592 |  Nodes: 32<br>Edges: 117<br>Type: North<br>Layout: Sugi<br>Mean: 0.334 |
|  Nodes: 100<br>Edges: 144<br>Type: NWS<br>Layout: FR<br>Mean: 0.623 |  Nodes: 90<br>Edges: 155<br>Type: ER<br>Layout: K K<br>Mean: 0.584 |  Nodes: 110<br>Edges: 321<br>Type: BBA<br>Layout: Circ<br>Mean: 0.438 |

Table 2: Examples of three “best", “typical", and “worst" graph drawings, chosen to cover a range of generators, sizes and layouts. The “mean" refers to the mean of the metric values across the ten metrics for each drawing; the mean of medians for each metric across all drawings is 0.591.

shows better stimuli, where the three Edge Orthogonality metric values are comparable within the context of its distribution.

Observation of the distributions can help when deciding on a graph layout algorithm that favours some aesthetic criteria over others. For example, if Neighbourhood Preservation is important, DRGraph and HOLA look best; if the Edge Lengths criterion is also added, then HOLA would be preferred. We would not choose Sugiyama if Crossings Angles are important; Kamada and Kawai is likely to give better Node Uniformity results.

In calculating the range of metric coverage for each algorithm, we find that Sugiyama has the greatest range (mean 0.78/metric), followed by DRGraph (mean 0.75/metric). As expected, the ranges for Kamada and Kawai and Fruchterman-Reingold are similar (0.68 and 0.70 respectively). When we consider the mean value over all metrics for all algorithms, we find that HOLA produces ‘the best’ layouts (0.65), followed by Kamada and Kawai (0.64) and then Fruchterman-Reingold (0.61). HOLA’s high score is a result of the fact that it has favourable Edge Crossing metric scores—it minimises edge crossings by laying




| | |
|---|---|
| AR | Angular Resolution |
| Asp | Aspect Ratio |
| CA | Crossing Angle |
| EC | Edge Crossings |
| EL | Edge Lengths |
| EO | Edge Orthogonality |
| GR | Gabriel Ratio |
| NP | Neighbourhood Preservation |
| NR | Node Resolution |
| NU | Node Uniformity |

Fig. 3: Correlations between each pair of metrics, grouped by layout algorithms.

out the core of the network (i.e., cycles with sub-tree removed) and subsequently expands the core to reintroduce sub-trees without introducing additional crossings—and the Edge Orthogonality metric favours grid-based layouts. This means that two of the ten metrics are close to 1.0 for the majority of HOLA drawings. Two of the algorithms perform even worse than Random (0.51): Sugiyama (0.46) and Circular (0.50). This is not surprising, since the layer-based approach of Sugiyama is very different from the other algorithms, and our choice of metrics does not advantage a layered directed acyclic method (unlike, as noted above, the Edge Orthogonality metric gives HOLA an advantage). The Circular algorithm is not optimised to reduce Edge Crossings, and the value of the Angular Resolution metric will naturally be low since, for each node, half of the surrounding area is off-limits with respect to placement of edges.

This paper presents the first attempt to define the landscape of graph layout aesthetic properties. We have provided distribution and correlation data that can be used to determine the comparative aesthetic worth of graph drawings, showing where they are positioned within a much larger sample of drawings. The dataset of half a million layouts is available, as is all the source code for the metric computations and the data analysis.

### 5.1 Limitations and Future Work

The distribution and correlation data presented here is limited to the graph generators, layout algorithms, and aesthetic metrics we have chosen. However, this does not invalidate our approach, but rather provides opportunities for the data to be extended and refined over time.

As a starting point, we chose to focus on sparse graphs to ensure that reasonably readable drawings could be produced (rather than 'hairballs'). Of course, the inclusion of larger, denser, or larger and denser graphs might change the data substantially (or perhaps not)—we leave this for future work. This may also require many of the metric calculations to be optimised for speed. Currently, the computation of all metrics in the paper is prohibitively expensive for larger and denser graphs. Furthermore, experimental papers which verify that the chosen metrics are beneficial in improving human understanding of graphs typically use small graphs. Further scrutiny is required to ensure these principles hold for larger graphs. Some metrics may have a 'tipping point' where the difference between metric values becomes insignifi-

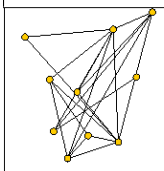
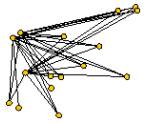
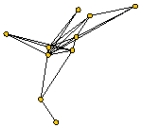
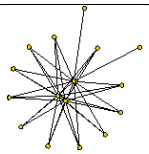
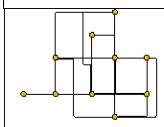
| Stimuli 1 | Nodes: 10<br>Edges: 20<br>Type: North | Stimuli 2 | Nodes: 16<br>Edges: 35<br>Type: North |
|---|---|---|---|
|  | Layout: Ran<br>EC: 0.648<br>EO: 0.387 |  | Layout: Ran<br>EC: 0.282<br>EO: 0.55 |
|  | Layout: DRG<br>EC: 0.824<br>EO: 0.469 |  | Layout: K K<br>EC: 0.632<br>EO: 0.502 |
|  | Layout: HOLA<br>EC: 0.991<br>EO: 1 |  | Layout: Sugi<br>EC: 0.737<br>EO: 0.526 |

Table 3: Example of how the data can be used to ensure appropriate metric controls in determining experimental stimuli. The first set of stimuli has a confounding factor; the second set controls for Edge Orthogonality.

cant. For example, a graph drawing with 1,000 crossings may be no more or less understandable than one with 1,500 crossings [19].

Similarly, we have not specifically focused on graph types of particular interest to the graph drawing community (e.g., trees, planar graphs, beyond planar graphs (like k-planar), 3D meshes, etc.). Again, this does not negate the contribution made here, and we look forward to extending the data with such specific considerations. Although the generators chosen are intended to model real-world graphs, we also plan to expand the dataset to include real-world graphs of interest to the graph visualization community.

We have confined our exploration to a limited set of ten metrics. Extending the landscape with more metric dimensions will enrich it. We are particularly interested in the potential for including stress and symmetry, since these are properties that are often (implicitly or explicitly) optimised in layout algorithms. Both present challenges with respect to implementation and/or normalisation.

## Supplemental Materials

- The code for generating graphs and layouts, calculating metrics, creating visualisations, and analysing results is available at https://github.com/gavjmooney/graph_metrics.
- The graph drawings and the CSV file of metric values used to create Fig. 1, Fig. 2, Fig. 3, and Table 1 are available at https://drive.google.com/drive/folders/1z3xwPZmx_ZlUhuXEkVdxrGUDSvT_QqoT?usp=sharing.
- The detailed metric formulae are available at https://drive.google.com/drive/folders/1GKytyTn6U02R7BX4CGQ1bZKdXTJrtWor?usp=sharing.

## References


[1] R. Ahmed, F. De Luca, S. Devkota, S. G. Kobourov, and M. Li. Multicrite-ria scalable graph drawing via stochastic gradient descent, (SGD)$^2$. IEEE Transactions on Visualization and Computer Graphics, 28(6):2388–2399, 2022. doi: 10.1109/TVCG.2022.3155564 2, 3, 11, 12

[2] A.-L. Barabasi and R. Albert. Emergence of scaling in random networks. Science, 286(5439):509–512, Oct. 1999. doi: 10.1126/science.286.5439.509 2

[3] H. Chen, V. Huroyan, S. Kobourov, and M. Kryven. On Random Graph Properties, June 2022. arXiv:2003.02673 [physics]. 2

[4] H. Chen, U. Soni, Y. Lu, V. Huroyan, R. Maciejewski, and S. Kobourov. Same stats, different graphs: Exploring the space of graphs in terms of graph properties. IEEE transactions on visualization and computer graphics, 27(3):2056–2072, 2019. 2

[5] R. Davidson and D. Harel. Drawing graphs nicely using simulated annealing. ACM Transactions on Graphics, 15(4):301–331, Oct. 1996. doi: 10.1145/234535.234538 3

[6] S. Devkota, R. Ahmed, F. De Luca, K. Isaacs, and S. Kobourov. Stress-Plus-X (SPX) Graph Layout. In Graph Drawing and Network Visualization: 27th International Symposium, GD 2019, Prague, Czech Republic, September 17–20, 2019, Proceedings, pp. 291–304. Springer, 2019. 1, 3

[7] G. Di Battista, A. Garg, G. Liotta, A. Parise, R. Tamassia, E. Tassinari, F. Vargiu, and L. Vismara. Drawing directed acyclic graphs: An experimental study. pp. 76–91, 1997. 2

[8] G. Di Battista, A. Garg, G. Liotta, R. Tamassia, E. Tassinari, and F. Vargiu. An experimental comparison of four graph drawing algorithms. Computational Geometry, 7(5):303–325, 1997. 11th ACM Symposium on Computational Geometry. doi: 10.1016/S0925-7721(96)00005-3 2

[9] P. Eades, S.-H. Hong, A. Nguyen, and K. Klein. Shape-based quality metrics for large graph visualization. J. Graph Algorithms Appl., 21(1):29–53, 2017. 3

[10] P. Erdös and A. Rényi. On random graphs I. Publicationes Mathematicae Debrecen, 6:290–297, 1959. 2

[11] T. M. J. Fruchterman and E. M. Reingold. Graph drawing by force-directed placement. Software: Practice and Experience, 21(11):1129–1164, Nov. 1991. doi: 10.1002/spe.4380211102 2

[12] M. Ghoniem, J.-D. Fekete, and P. Castagliola. On the readability of graphs using node-link and matrix-based representations: A controlled experiment and statistical analysis. Information Visualization, 4(2):114–135, 2005. doi: 10.1057/palgrave.ivs.9500092 3

[13] A. A. Hagberg, D. A. Schult, and P. J. Swart. Exploring network structure, dynamics, and function using networkx. In G. Varoquaux, T. Vaught, and J. Millman, eds., Proceedings of the 7th Python in Science Conference, pp. 11 – 15. Pasadena, CA USA, 2008. 3

[14] C. R. Harris, K. J. Millman, S. J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N. J. Smith, R. Kern, M. Picus, S. Hoyer, M. H. van Kerkwijk, M. Brett, A. Haldane, J. F. del Río, M. Wiebe, P. Peterson, P. Gérard-Marchant, K. Sheppard, T. Reddy, W. Weckesser, H. Abbasi, C. Gohlke, and T. E. Oliphant. Array program-ming with NumPy. Nature, 585(7825):357–362, Sept. 2020. doi: 10.1038/s41586-020-2649-2 3

[15] P. W. Holland, K. B. Laskey, and S. Leinhardt. Stochastic blockmodels: First steps. Social Networks, 5(2):109–137, June 1983. doi: 10.1016/0378-8733(83)90021-7 2

[16] J. D. Hunter. Matplotlib: A 2D graphics environment. Computing in Science & Engineering, 9(3):90–95, 2007. doi: 10.1109/MCSE.2007.55 4

[17] T. Kamada and S. Kawai. An algorithm for drawing general undirected graphs. Information Processing Letters, 31(1):7–15, 1989. doi: 10.1016/0020-0190(89)90102-6 2, 3

[18] S. Kieffer, T. Dwyer, K. Marriott, and M. Wybrow. HOLA: Human-like Orthogonal Network Layout. IEEE Transactions on Visualization and Computer Graphics, 22(1):349–358, Jan. 2016. doi: 10.1109/TVCG.2015.2467451 2

[19] S. G. Kobourov, S. Pupyrev, and B. Saket. Are crossings important for drawing large graphs? In Graph Drawing: 22nd International Symposium, GD 2014, Würzburg, Germany, September 24-26, 2014, Revised Selected Papers 22, pp. 234–245. Springer, 2014. 9

[20] E. Koutsofios, S. North, S. Intset, and S. Sparcmcemit. Drawing graphs with dot. 01 1999. 2

[21] J. F. Kruiger, P. E. Rauber, R. M. Martins, A. Kerren, S. G. Kobourov, and A. C. Telea. Graph layouts by t-SNE. Comput. Graph. Forum, 36(3):283– 294, 2017. doi: 10.1111/cgf.13187 2, 3

[22] J. B. Kruskal. Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. Psychometrika, 29(1):1–27, 1964. 2

[23] A. Lancichinetti, S. Fortunato, and F. Radicchi. Benchmark graphs for testing community detection algorithms. Physical Review E, 78(4):046110, Oct. 2008. arXiv:0805.4770 [physics]. doi: 10.1103/PhysRevE.78.046110 2

[24] M. E. J. Newman and D. J. Watts. Renormalization group analysis of the small-world network model. Physics Letters A, 263(4-6):341–346, Dec. 1999. doi: 10.1016/S0375-9601(99)00757-4 2

[25] D. S. Parker. Aesthetics-based graph layout for human consumption. Softw.

Pract. Exp., 26:1415–1438, 1996. 2, 3
[26] M. Penrose. Random geometric graphs, vol. 5. OUP Oxford, 2003. 2
[27] H. Purchase. Which aesthetic has the greatest effect on human understanding? In G. DiBattista, ed., Graph Drawing, pp. 248–261. Springer Berlin Heidelberg, Berlin, Heidelberg, 1997. 1, 2, 3
[28] H. C. Purchase. Metrics for graph drawing aesthetics. Journal of Visual Languages & Computing, 13(5):501–516, 2002. doi: 10.1006/jvlc.2002.0232 2
[29] H. C. Purchase, R. F. Cohen, and M. James. Validating graph drawing aesthetics. In G. Goos, J. Hartmanis, J. van Leeuwen, and F. J. Brandenburg, eds., Graph Drawing, vol. 1027, pp. 435–446. Springer Berlin Heidelberg, Berlin, Heidelberg, 1996. Series Title: Lecture Notes in Computer Science. doi: 10.1007/BFb0021827 1, 2
[30] M. Radermacher, K. Reichard, I. Rutter, and D. Wagner. A geometric heuristic for rectilinear crossing minimization. In 2018 Proceedings of the Twentieth Workshop on Algorithm Engineering and Experiments (ALENEX), pp. 129–138. SIAM, 2018. 3
[31] R. N. Shepard. The analysis of proximities: multidimensional scaling with an unknown distance function. i. Psychometrika, 27(2):125–140, 1962. 2
[32] K. Sugiyama, S. Tagawa, and M. Toda. Methods for visual understanding of hierarchical system structures. IEEE Transactions on Systems, Man, and Cybernetics, 11:109–125, 1981. 2
[33] The Pandas development team. pandas-dev/pandas: Pandas, Feb. 2020. doi: 10.5281/zenodo.3509134 4
[34] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett, J. Wilson, K. J. Millman, N. Mayorov, A. R. J. Nelson, E. Jones, R. Kern, E. Larson, C. J. Carey, I. Polat, Y. Feng, E. W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E. A. Quintero, C. R. Harris, A. M. Archibald, A. H. Ribeiro, F. Pedregosa, P. van Mulbregt, and SciPy 1.0 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. Nature Methods, 17:261– 272, 2020. doi: 10.1038/s41592-019-0686-2 4
[35] Y. Wang, Y. Wang, Y. Sun, L. Zhu, K. Lu, C.-W. Fu, M. Sedlmair, O. Deussen, and B. Chen. Revisiting stress majorization as a unified framework for interactive constrained graph visualization. IEEE transactions on visualization and computer graphics, 24(1):489–499, 2017. 1, 3
[36] C. Ware, H. Purchase, L. Colpoys, and M. McGill. Cognitive Measurements of Graph Aesthetics. Information Visualization, 1(2):103–110, June 2002. doi: 10.1057/palgrave.ivs.9500013 1, 2, 3
[37] D. J. Watts and S. H. Strogatz. Collective dynamics of 'small-world' networks. 393, 1998. 2
[38] E. Welch and S. Kobourov. Measuring Symmetry in Drawings of Graphs. Computer Graphics Forum, 36(3):341–351, June 2017. doi: 10.1111/cgf.13192 3
[39] Wes McKinney. Data Structures for Statistical Computing in Python. In Stéfan van der Walt and Jarrod Millman, eds., Proceedings of the 9th Python in Science Conference, pp. 56 – 61, 2010. doi: 10.25080/Majora-92bf1922-00a 4
[40] M. Xue, Z. Wang, F. Zhong, Y. Wang, M. Xu, O. Deussen, and Y. Wang. Taurus: Towards a unified force representation and universal solver for graph layout. IEEE Transactions on Visualization and Computer Graphics, 29(1):886–895, 2023. doi: 10.1109/TVCG.2022.3209371 1
[41] V. Yoghourdjian, D. Archambault, S. Diehl, T. Dwyer, K. Klein, H. C. Purchase, and H.-Y. Wu. Exploring the limits of complexity: A survey of empirical studies on graph visualisation. Visual Informatics, 2(4):264–282, 2018. 3
[42] M. Zhu, W. Chen, Y. Hu, Y. Hou, L. Liu, and K. Zhang. Drgraph: An efficient graph layout algorithm for large-scale graphs by dimensionality reduction. IEEE Transactions on Visualization and Computer Graphics, 27(2):1666–1676, 2021. doi: 10.1109/TVCG.2020.3030447 2, 3

## A Appendix: Metric formulae in detail

- Angular Resolution (AR): The average deviation of minimum angles to an ideal angle of adjacent edges around a node, excluding nodes of degree one.

$$AR = 1 - \frac{1}{n^{deg>1}} \sum_{i=1}^{n^{deg>1}} \frac{\vartheta^i - \theta^{i\,min}}{\vartheta^i} \tag{1}$$

where $\vartheta_i$ is the ideal minimum angle at the node $u_i$,

$$\vartheta_i = \frac{360^\circ}{\text{degree}(u_i)} \tag{2}$$

and $\theta_{i\,min}$ is the actual minimum angle between the incident edges at node $u_i$.

- Aspect Ratio (Asp): Aspect ratio is the aspect ratio of the bounding box surrounding the graph drawing. That is, the height of the bounding box divided by the width, or vice versa if the height is larger than the width. If the height or width is zero then we set the aspect ratio to one.

- Crossing Angle (CA). Using the same approach as Angular Resolution we measure the average deviation of angles at which edges cross from an ideal. Crosses promotion is first performed on the graph drawing, then the same method for AR is used only on the newly created crossing nodes.

- Edge Crossings (EC): The number of edge crossings in the drawing, scaled against the total possible crossings.

$$EC = 1 - \begin{cases} \frac{c}{c_{mx}}, & \text{if } c_{mx} > 0 \\ 0, & \text{otherwise} \end{cases} \tag{3}$$

Where c is the number of crossings and $c_{mx}$ is the upper bound on the number of possible crossings,

$$c_{mx} = \frac{m(m-1)}{2} - c_{deg} \tag{4}$$

$$c_{deg} = \frac{1}{2} \sum_{j=1}^{n} \text{degree}(u_j)(\text{degree}(u_j) - 1) \tag{5}$$

and $c_{deg}$ is the number of crossings which are impossible due to the fact that adjacent edges cannot cross.

We expand upon this definition to reduce $c_{mx}$ further by accounting for triangles and 4-cycles in the graph. Pairs of triangles can only cross at most six times, as opposed to the nine calculated by only using $c_{deg}$. We do however have to account for triangles with shared edges and nodes, as these cases are partially handled by $c_{deg}$. Additionally non-adjacent edges to a triangle can only cross at most two of the triangle's edges. We call the number of crossings which are impossible due to triangle interactions $c_{tri}$ which is calculated using Algorithm 1.

We can also reduce $c_{mx}$ by the number of 4-cycles in the graph, $c_{4cyc}$, due to the fact that if two edges in a 4-cycle cross, it is impossible for the other two edges to cross.

The final calculation for $c_{mx}$ becomes:

$$c_{mx} = \frac{m(m-1)}{2} - c_{deg} - c_{tri} - c_{4cyc} \tag{6}$$

Giving a tighter upper bound on the number of possible crossings.

Algorithm 1 Algorithm to calculate $C_{tri}$

```
procedure REDUCE TRIANGLES(Graph : G)
    c_tri ← 0
    for each triangle, t, in G do
        for each edge, m, in G do
            if m is not part of, or adjacent to any triangle in G then
                c_tri ← c_tri + 1
            end if
        end for
    end for
    for each pair of triangles, t, u, in G do
        if t and u share an edge then
            c_tri ← c_tri + 1
        else if t and u share a node then
            c_tri ← c_tri + 2
        else
            c_tri ← c_tri + 3
        end if
    end for
end procedure
```

- Edge Length (EL): Ahmed et al. [1] minimise the average deviation of edge lengths from a set of ideal edge lengths. We take a similar approach and define EL in a similar fashion to AR, taking the average deviation of edge lengths from an ideal and normalising against the number of edges. We define the ideal edge length as the average length of all edges in the existing drawing.

- Edge Orthogonality (EO): The average deviation of edge angles to the horizontal and vertical axes.

$$EO = 1 - \frac{1}{m} \sum_{i=1}^{m} \delta_i, \text{ where,} \tag{7}$$

$$\delta_i = \frac{\min(\theta_i, |90^\circ - \theta_i|, 180^\circ - \theta_i)}{45^\circ} \tag{8}$$

and $\theta_i$ is the positive angle between node $u_i$ and the x-axis.

- Gabriel Ratio (GR): A graph is a Gabriel graph if it can be drawn in such a way that no node is placed within the circle formed by using any edge as its diameter. Ahmed et. al [1] incorporate this idea by adding repulsive forces around the midpoints of edges. We propose a new metric called the Gabriel Ratio (GR) which measures the extent to which a graph drawing conforms to this principle. In a similar fashion to EC, we count the number of nodes which fall within a circle formed using an edge as its radius and weigh this against an estimate for the total possible nodes which could do so.

$$GR = 1 - \begin{cases} \frac{g}{g_{mx}}, & \text{if } g_{mx} > 0 \\ 0, & \text{otherwise} \end{cases} \tag{9}$$

Where g is the number of non-conforming nodes and $g_{mx}$ is the upper bound on the number of possible non-conforming nodes,

$$g_{mx} = m(n-2) \tag{10}$$

We also reduce $g_{mx}$ when a non-conforming node is found—by one if the non-conforming node is adjacent to one of the endpoints of the edge it violates, and by two if it is adjacent to both endpoints. We do so because the endpoints of two adjacent edges cannot simultaneously be inside the circles formed by both edges. This gives us a better estimate of the upper bound and hence a more accurate distribution of values when the metric is calculated.

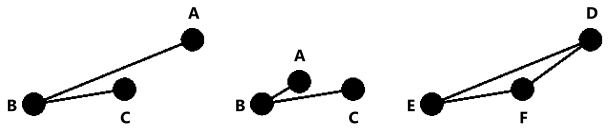



Fig. 4: An example showing the need to lower the upper bound of non-conforming nodes in the GR metric.

For example, consider Fig. 4. Looking at node C (on the left), we can see that it does not conform to the Gabriel property for the edge (A,B). However, since node C is also connected to node B, it is not possible for node A to break the property for the edge (B,C), as moving node A (in the middle) to do so would mean node C is no longer a violating node for (A,B).

Looking at nodes D, E, and F, we can draw similar conclusions, though there are now three edges where the Gabriel property cannot be violated simultaneously, instead of two.

- Neighbourhood Preservation (NP): In understanding the overall structure of a graph, it makes sense to put those nodes that are close together in terms of graph-theoretic distance near to each other with respect to geometric (Euclidean) distance on the plane. The Neighbourhood Preservation metric [1] uses the Jaccard similarity index between the adjacency matrix and the matrix that shows whether each pair of nodes is in the set of k-nearest (geometric) neighbours.
- Node Resolution (NR): The ratio of the minimum and maximum Euclidean distances between any pair of nodes.
- Node Uniformity (NU): Node Uniformity measures how uniformly distributed nodes are inside the bounding box of the drawing. We calculate it by splitting the box into cells and counting the number of nodes in each cell, then comparing this to an ideal distribution. The size of cells (and hence number of cells) is scaled by the square root of the number of nodes in the graph.

## B Appendix: Distributions for each layout, grouped by graph generator

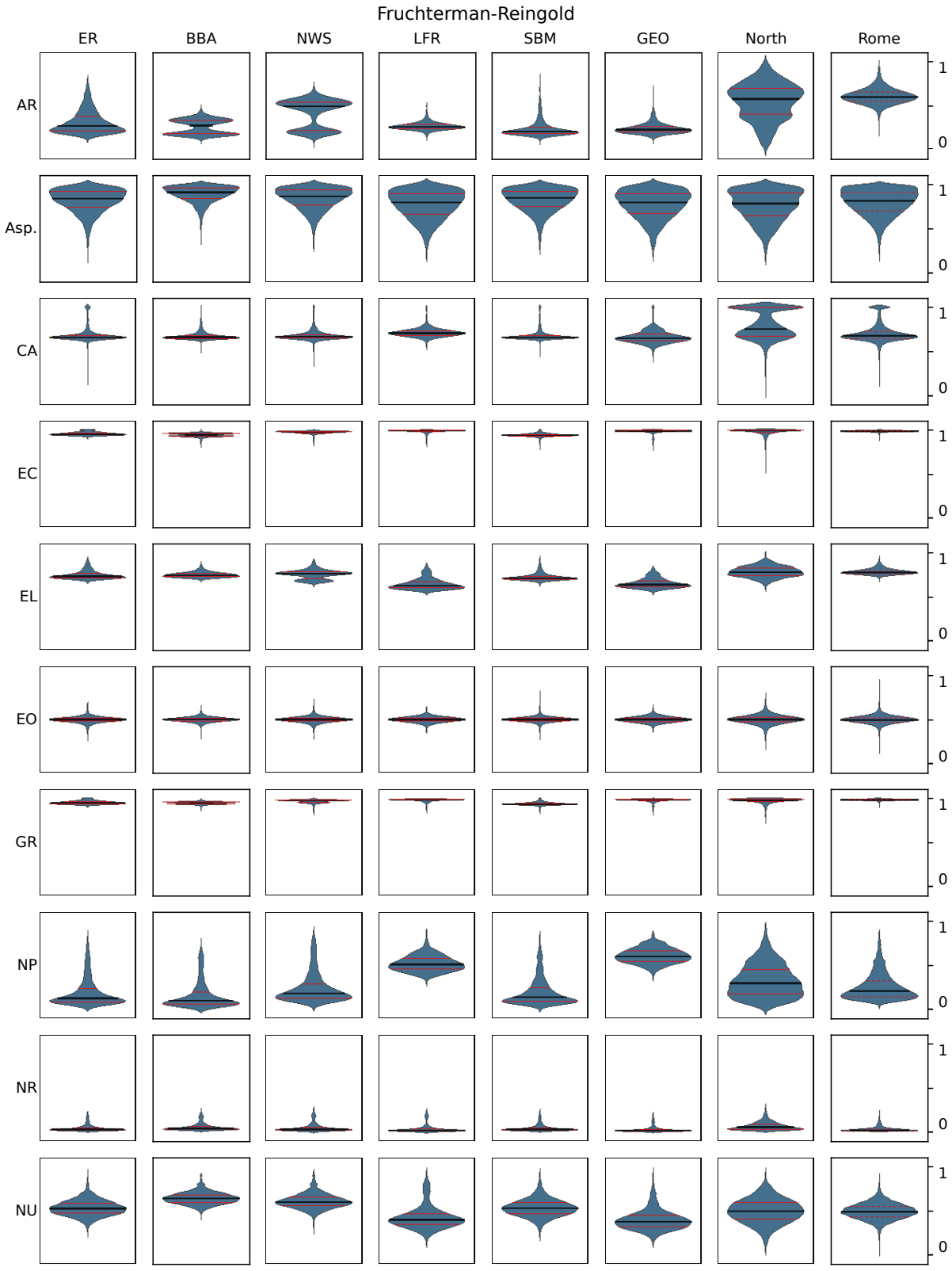



Fig. 5: Distributions for Fruchterman-Reingold drawings, grouped by graph generation method.

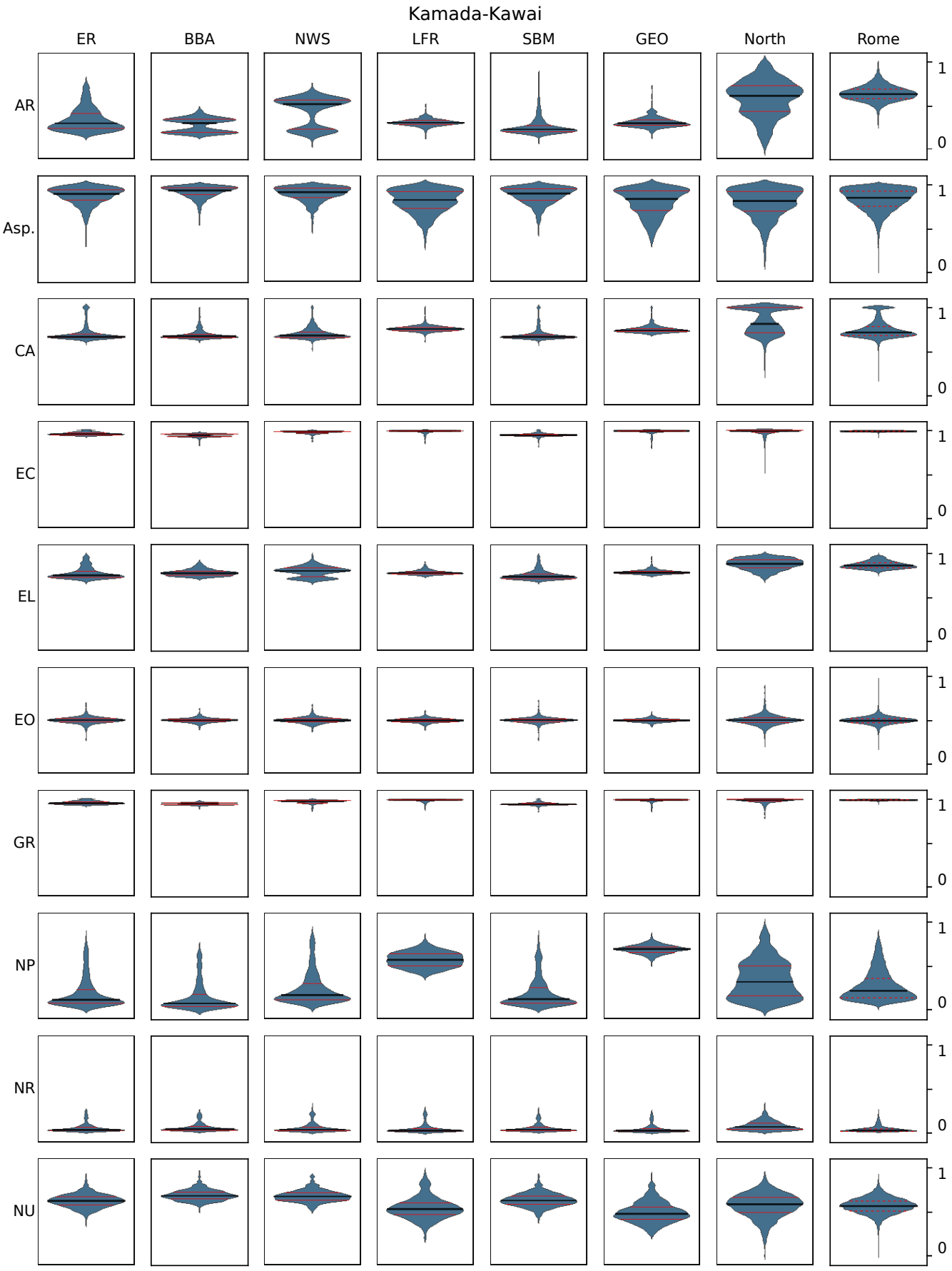



Fig. 6: Distributions for Kamada-Kawai drawings, grouped by graph generation method.

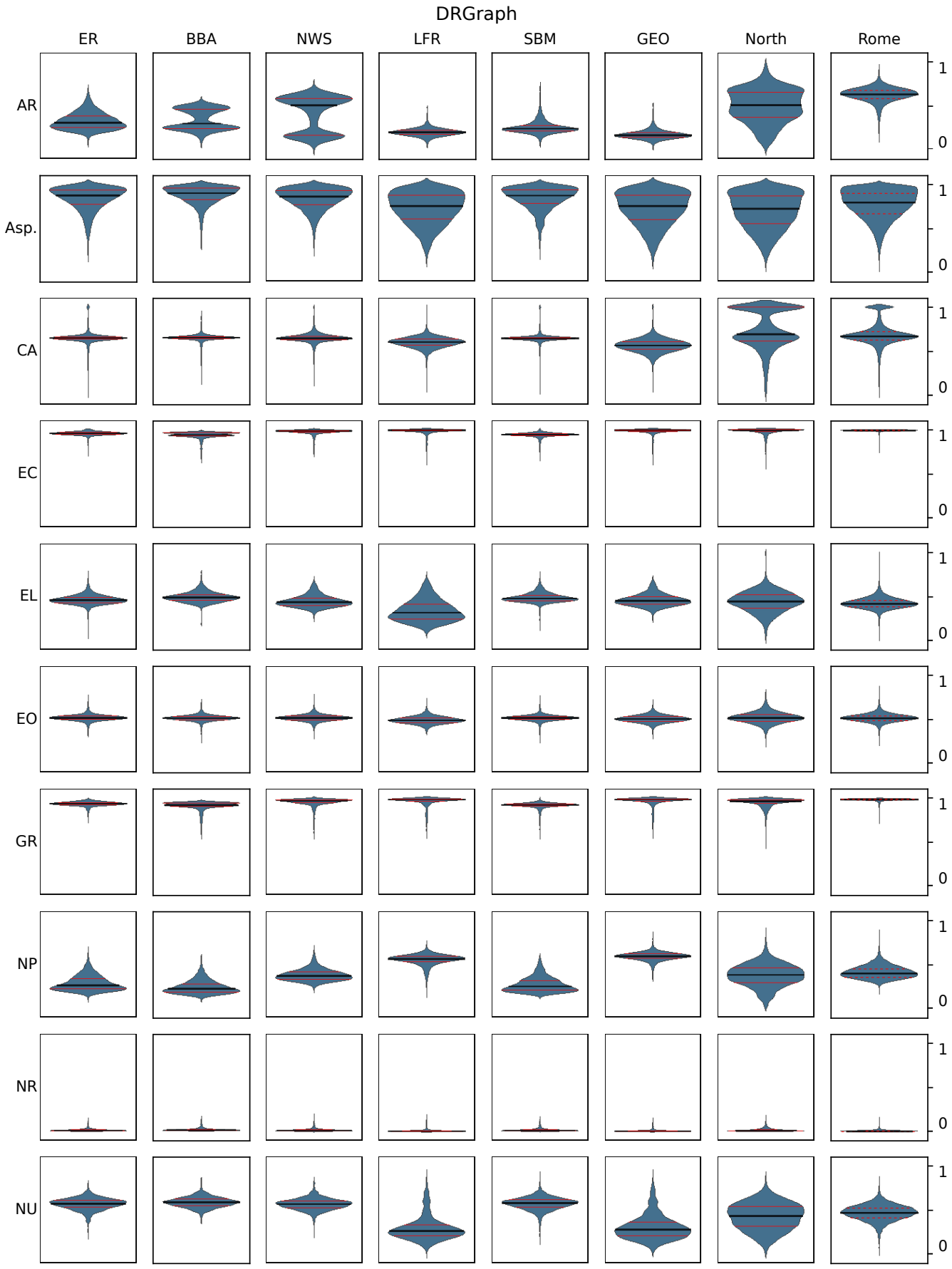



Fig. 7: Distributions for DRGraph drawings, grouped by graph generation method.

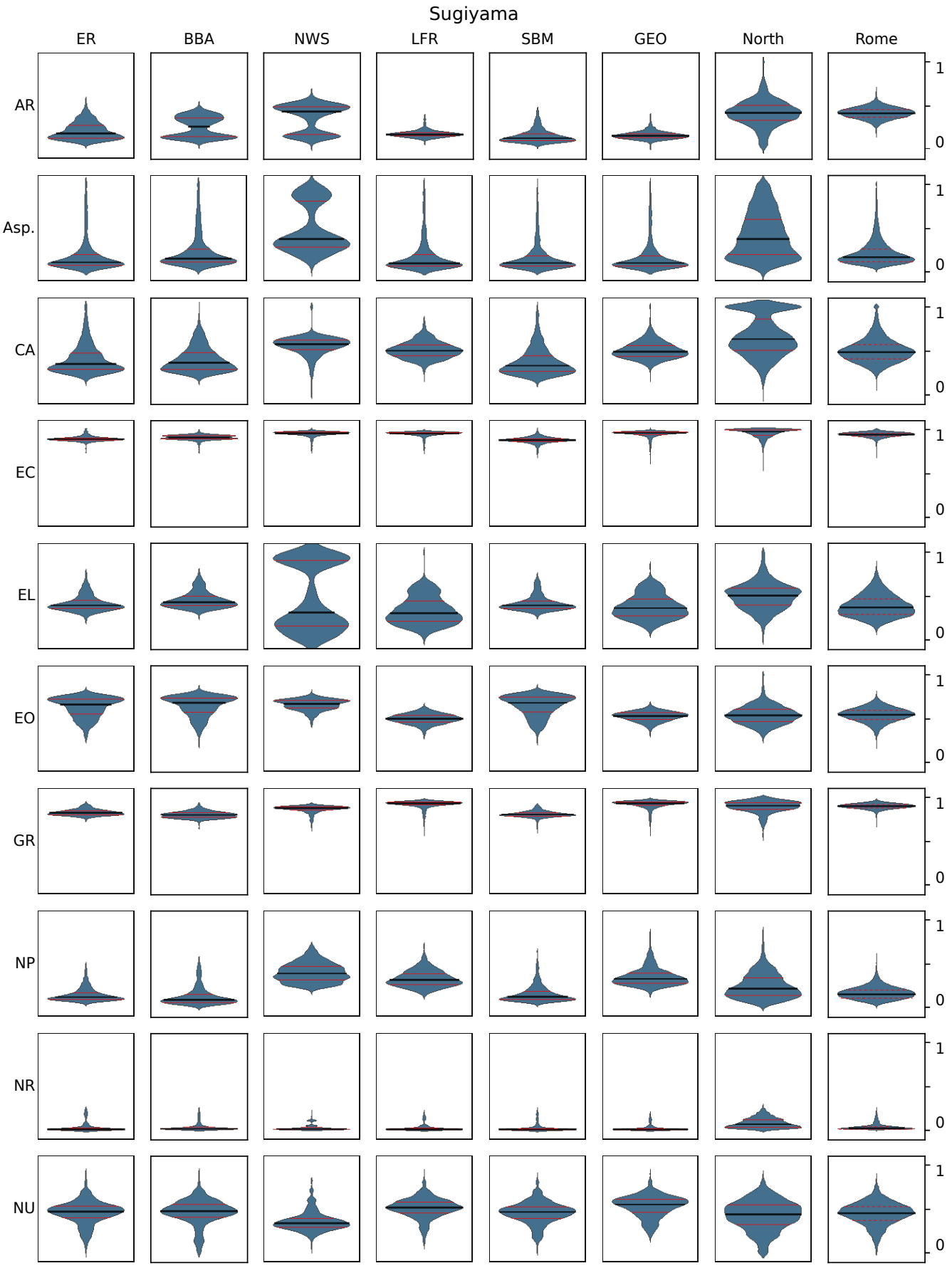



Fig. 8: Distributions for Sugiyama drawings, grouped by graph generation method.

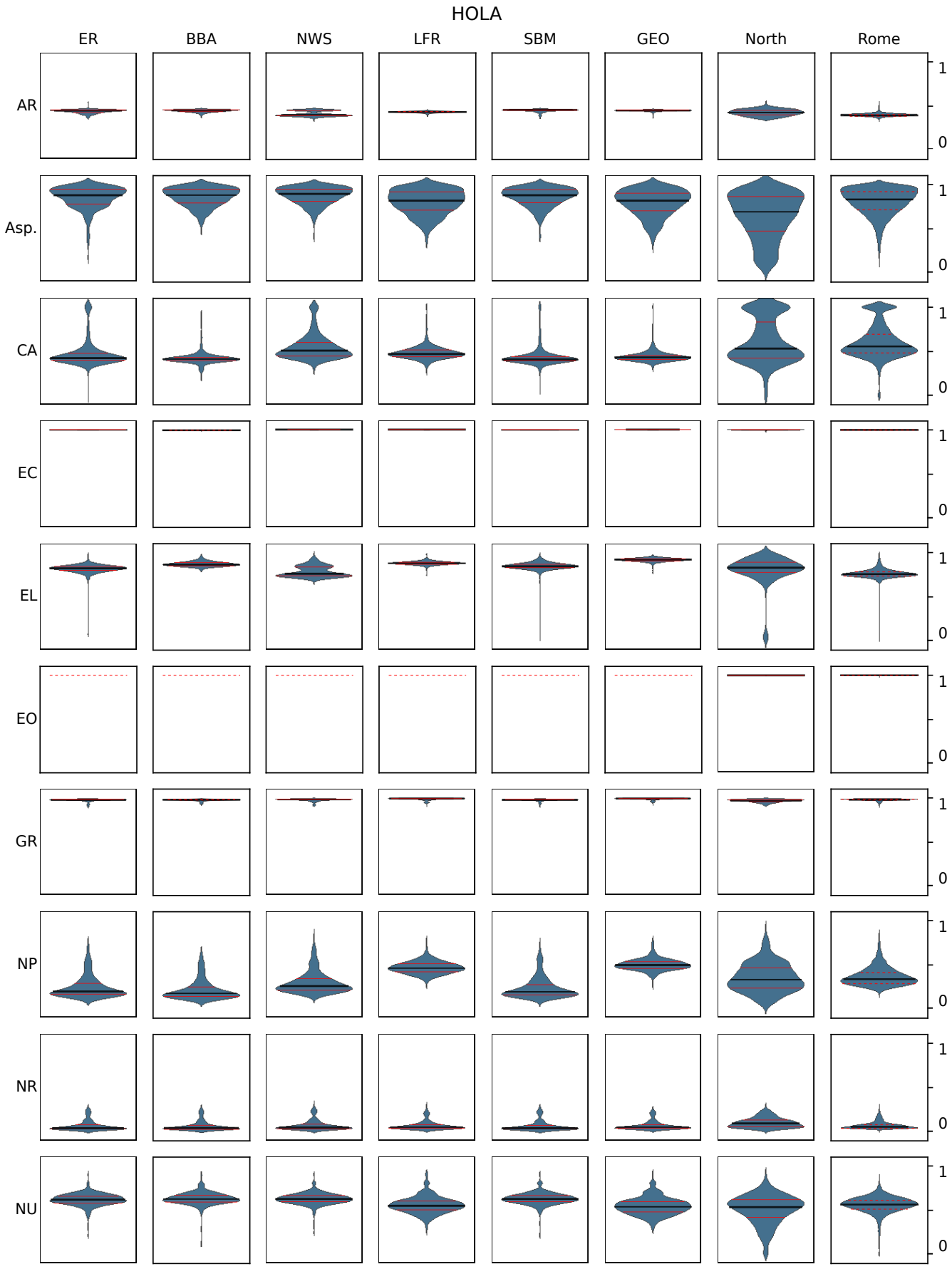



Fig. 9: Distributions for HOLA drawings, grouped by graph generation method.

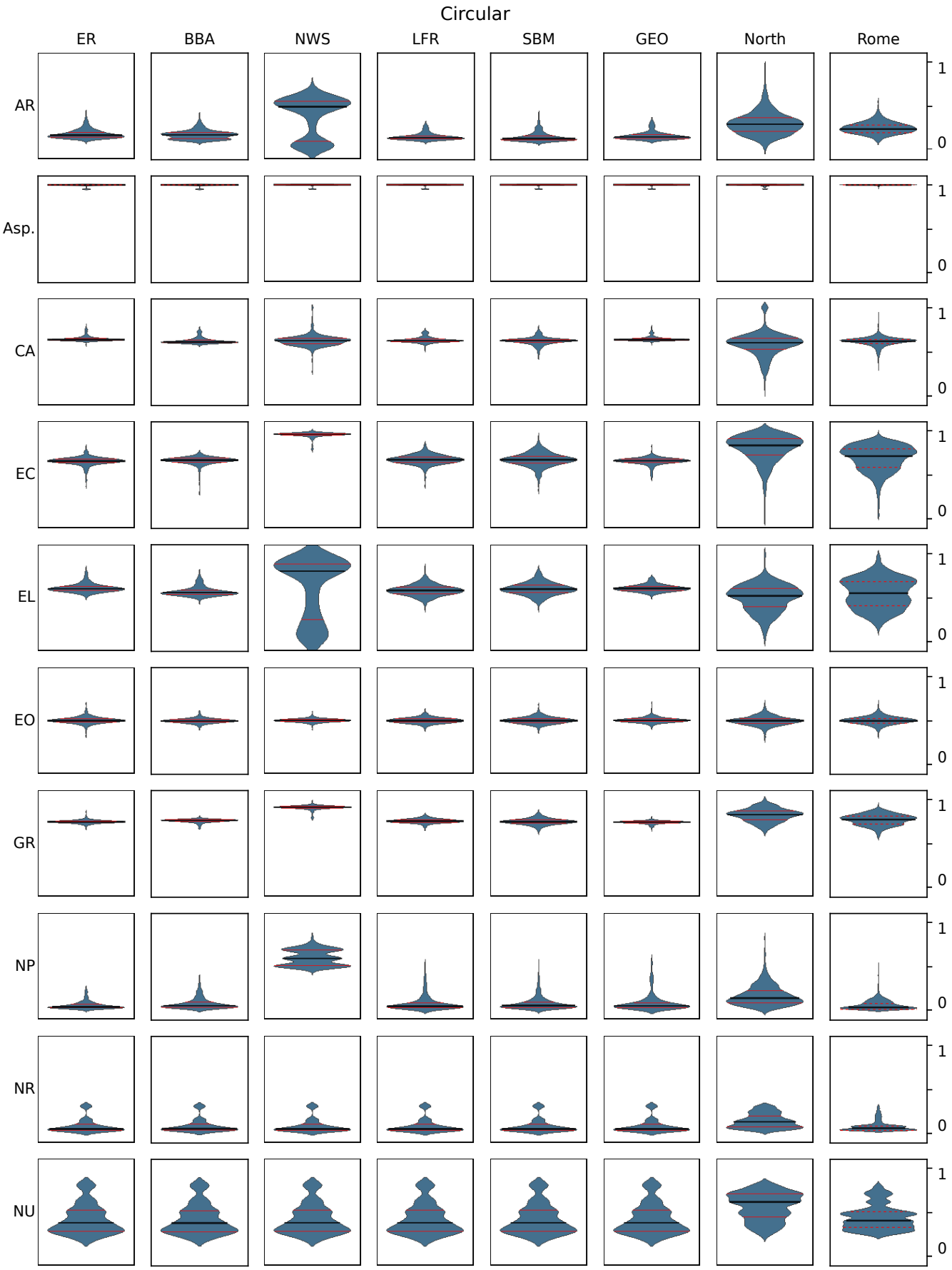



Fig. 10: Distributions for Circular drawings, grouped by graph generation method.

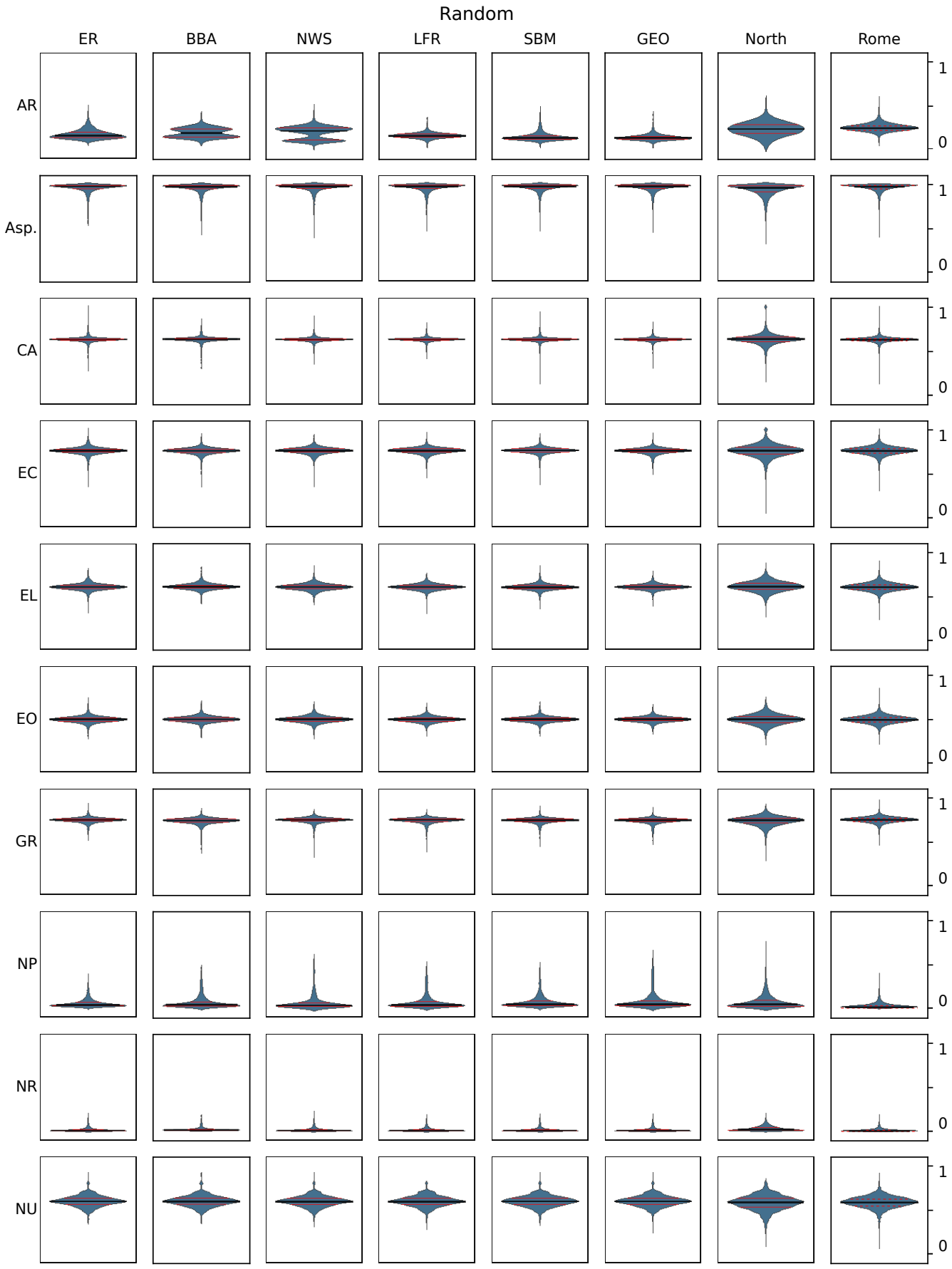



Fig. 11: Distributions for Random drawings, grouped by graph generation method.

## C Appendix: Distributions for each layout, grouped by number of nodes

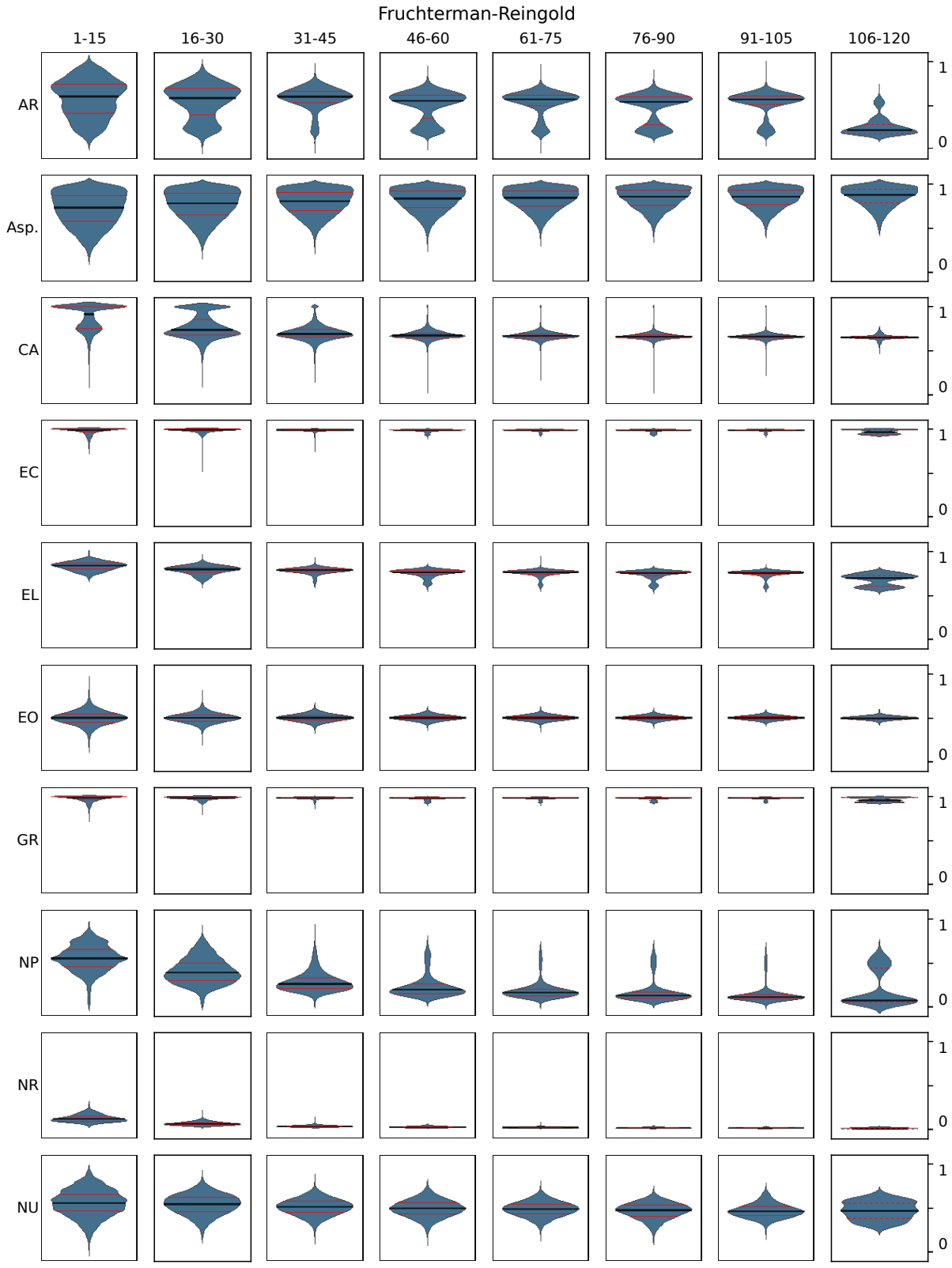



Fig. 12: Distributions for Fruchterman-Reingold drawings, grouped by number of nodes.

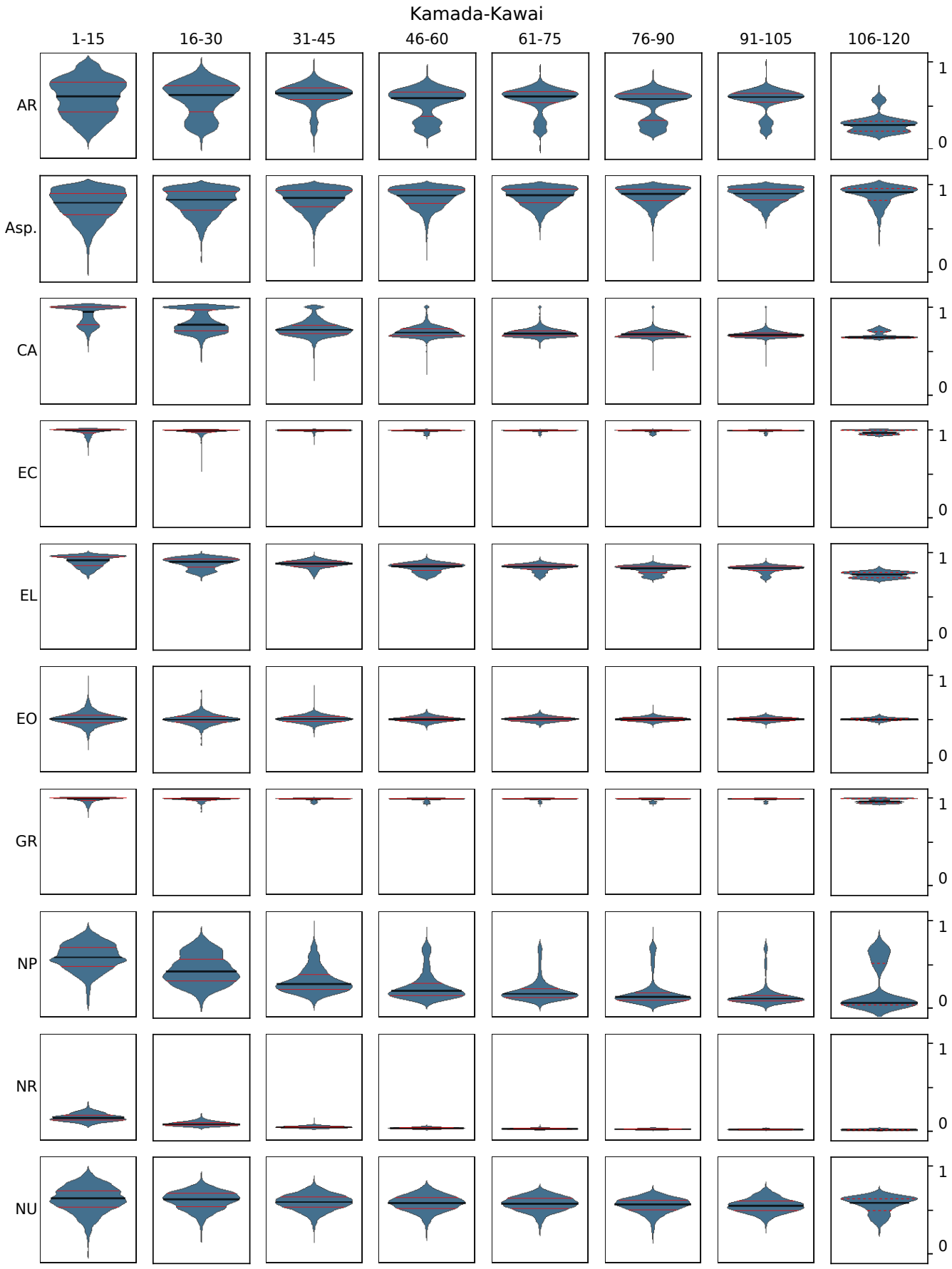



Fig. 13: Distributions for Kamada-Kawai drawings, grouped by number of nodes.

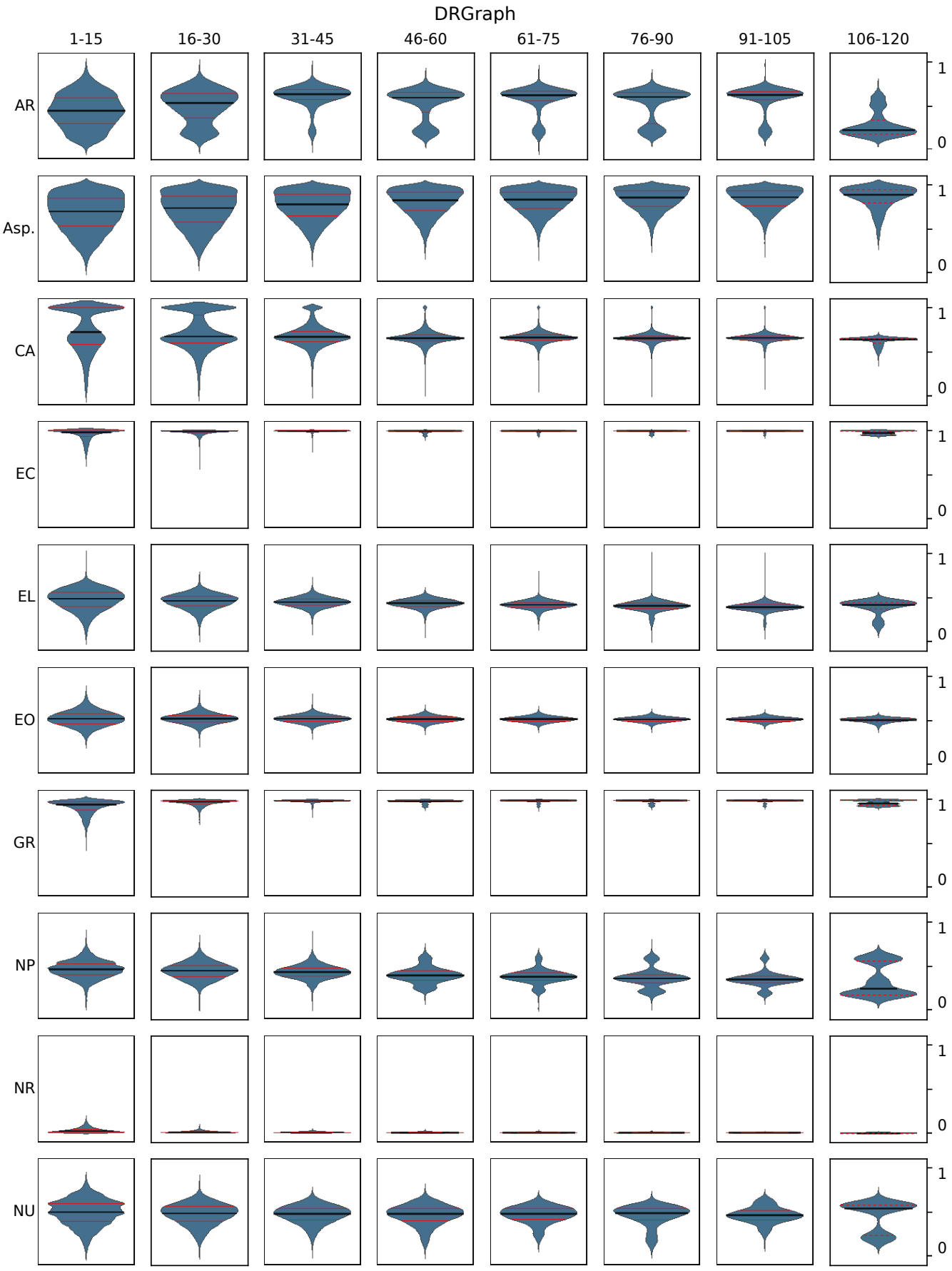



Fig. 14: Distributions for DRGraph drawings, grouped by number of nodes.

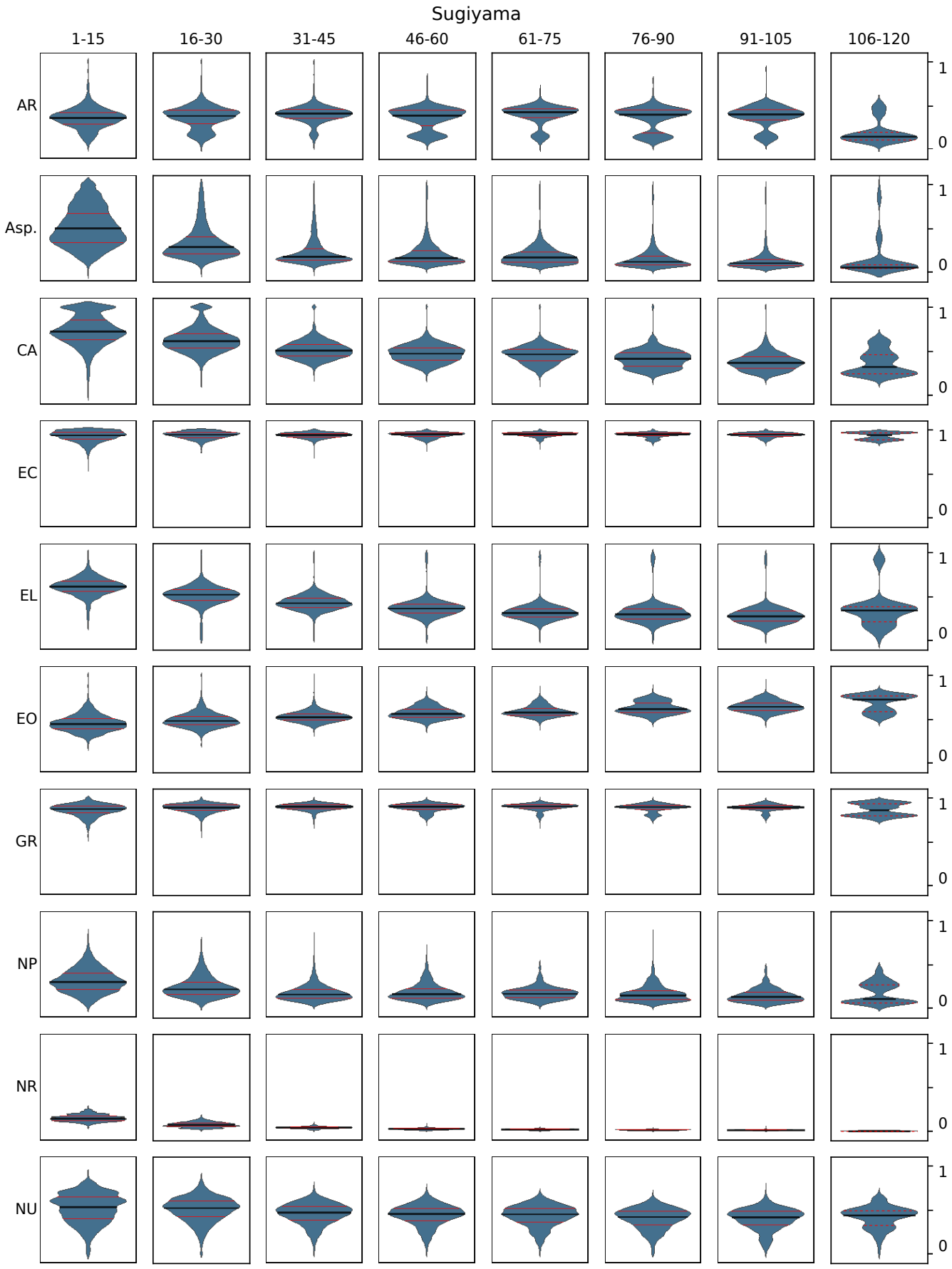



Fig. 15: Distributions for Sugiyama drawings, grouped by number of nodes.

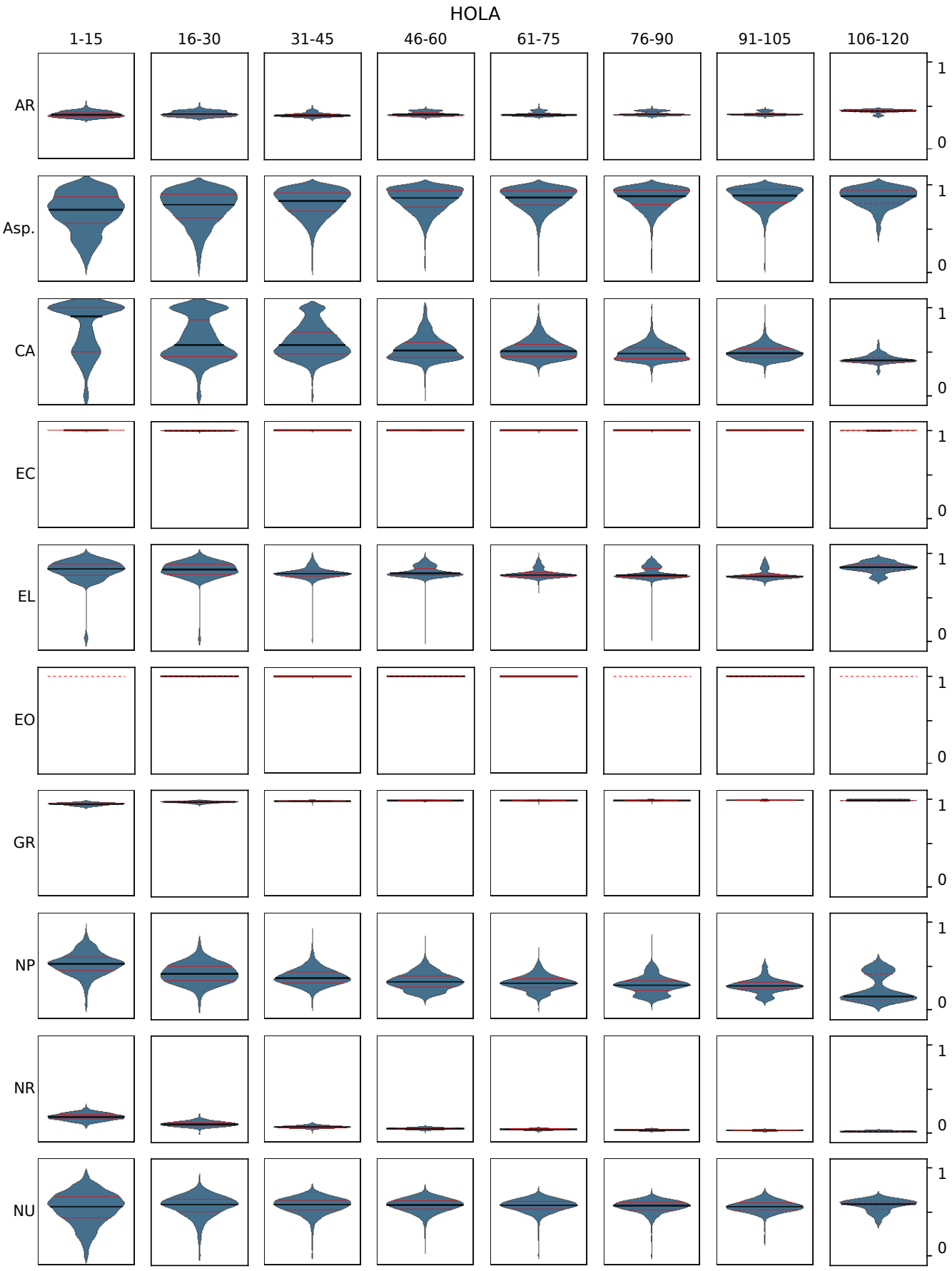



Fig. 16: Distributions for HOLA drawings, grouped by number of nodes.

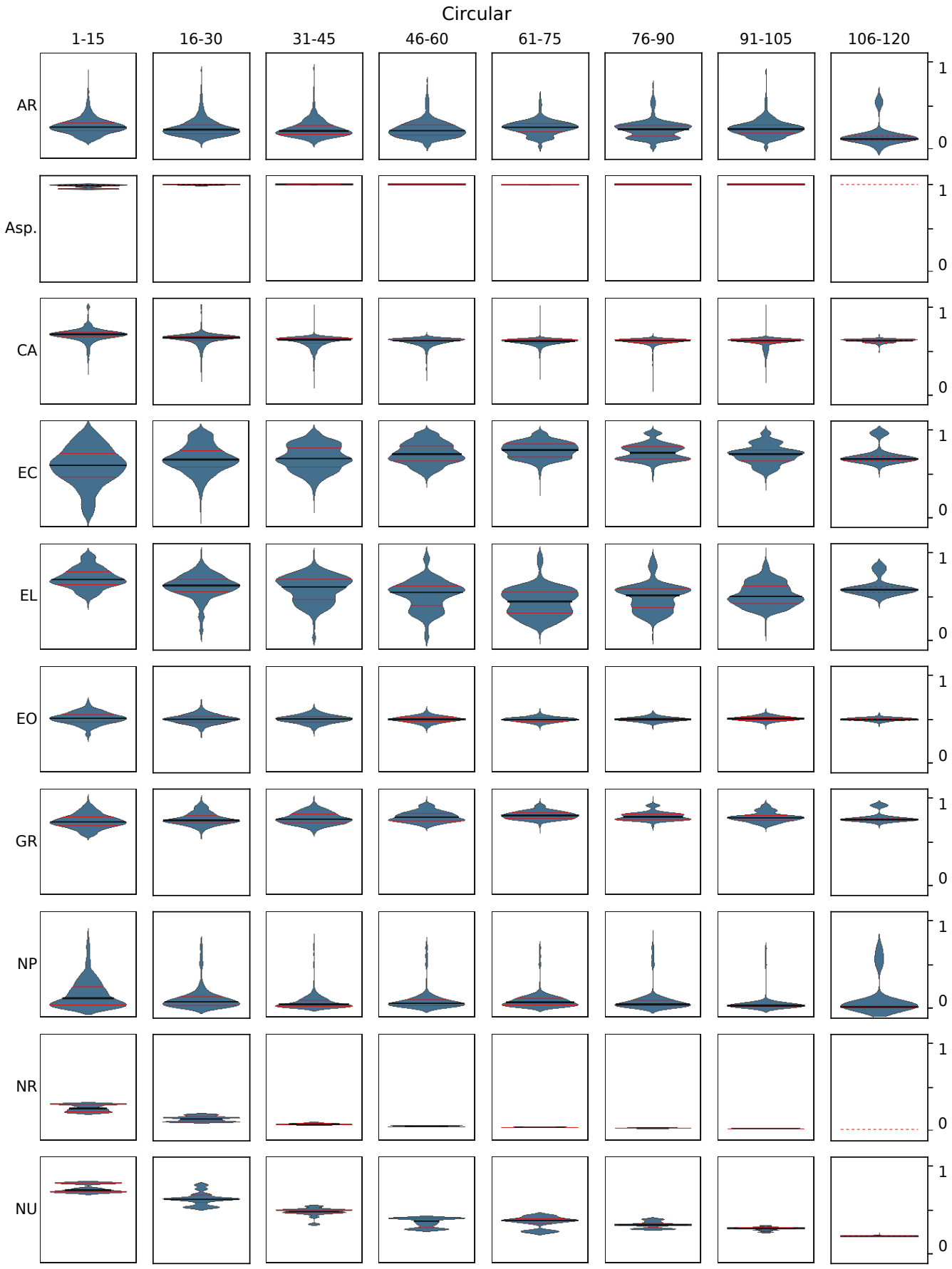



Fig. 17: Distributions for Circular drawings, grouped by number of nodes.

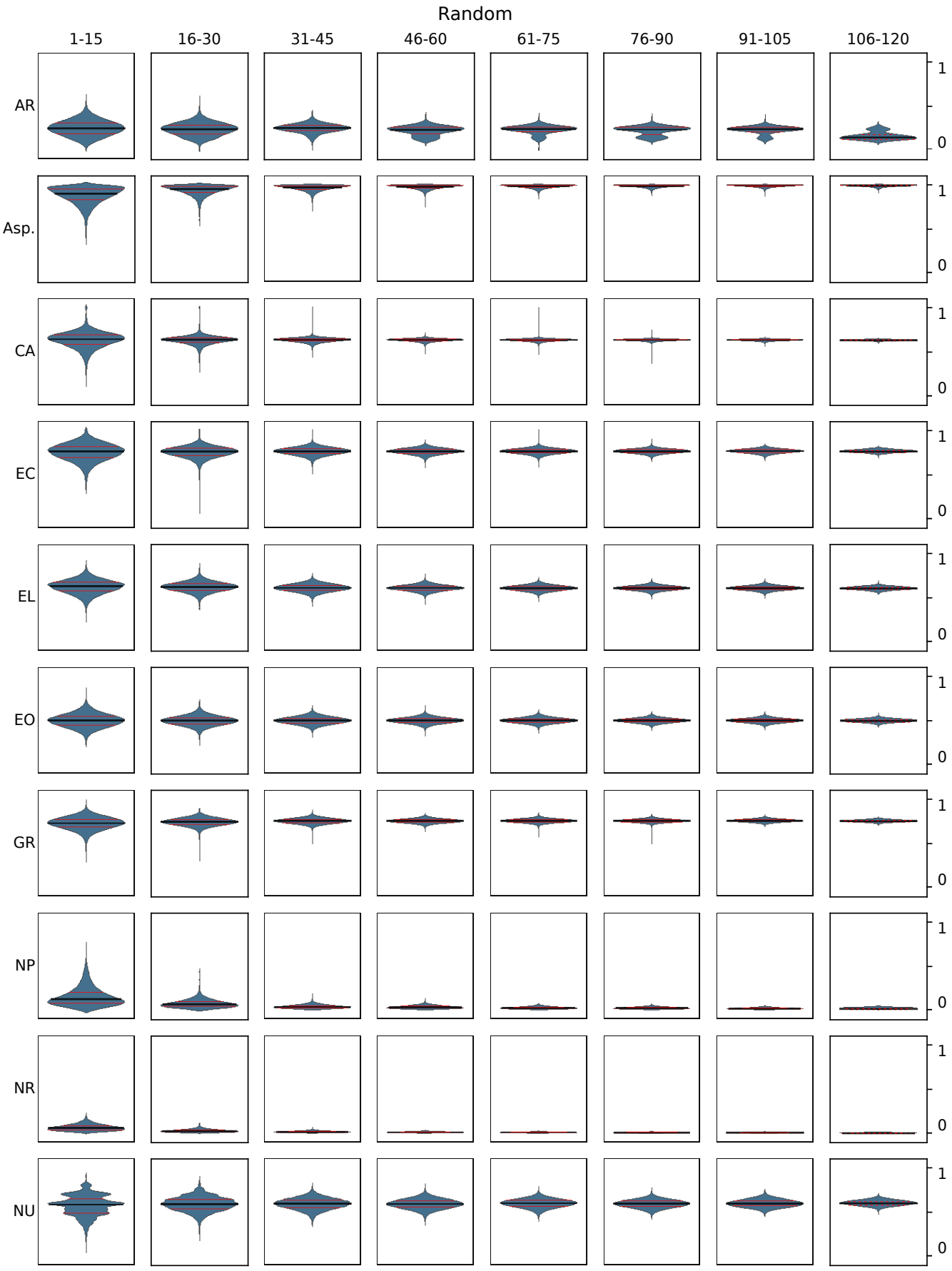



Fig. 18: Distributions for Random drawings, grouped by number of nodes.