

A Generic Framework and Library for Exploration of Small Multiples Through Interactive Piling

Supplementary Material

Fritz Lekschas

Harvard School of Engineering and Applied Sciences

Xinyi Zhou

State Key Lab of CAD&CG, Zhejiang University

Wei Chen

State Key Lab of CAD&CG, Zhejiang University

Nils Gehlenborg

Harvard Medical School

Benjamin Bach

University of Edinburgh

Hanspeter Pfister

Harvard School of Engineering and Applied Sciences

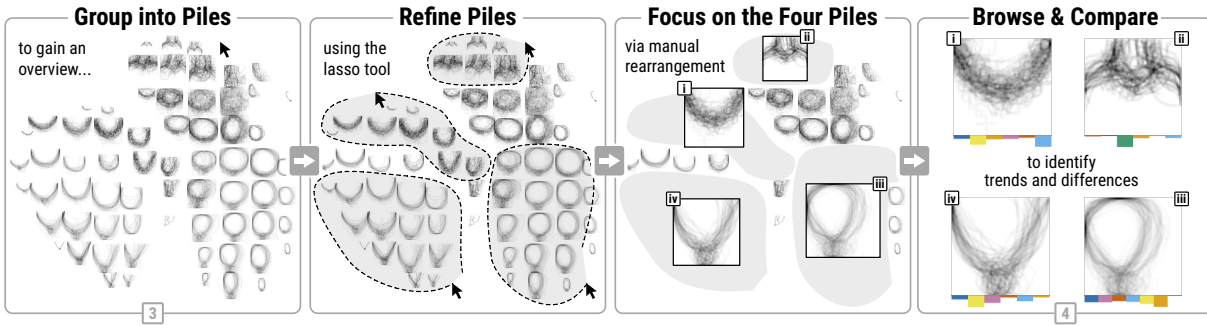


Figure S1: **Grouping and Arrangement Refinement From the Teaser Figure.** After grouping items in close proximity into piles in Fig. 1A3, we further refine the grouping and isolate four piles via manual rearrangement to show overarching concepts in how people think of necklaces. Please also see our Supplementary Video for a recorded exploration of this dataset.

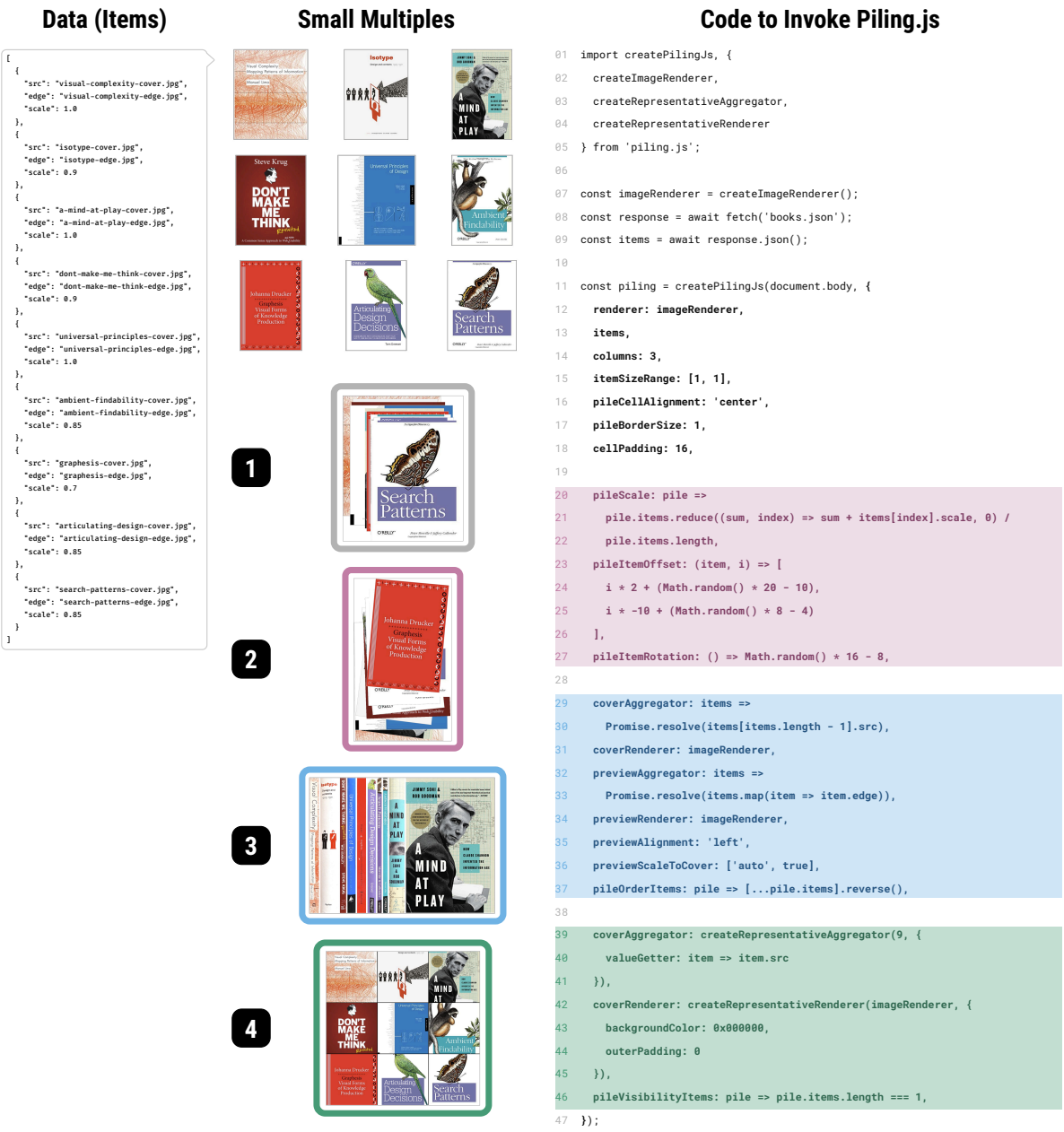


Figure S2: **View Specification.** We exemplify how the declarative view specification (bold source code) enables different pile encodings. In (1), we show the default pile encoding. In (2), we randomized the item offset for partial previewing. In (3), we implemented foreshortened previews. And in (4), we visualize the pile as a gallery preview.

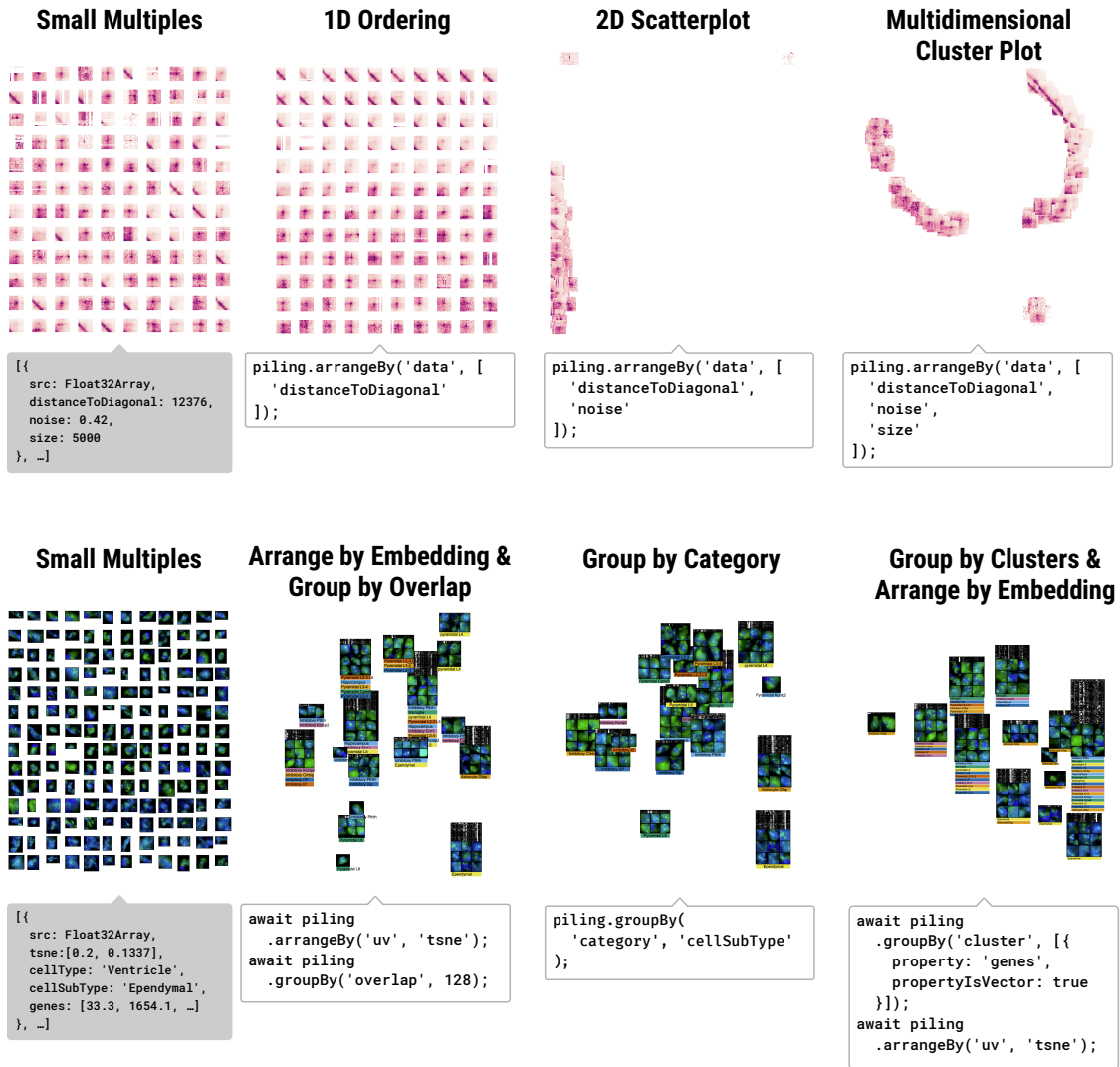


Figure S3: **Arrangement and Grouping in PILING.JS.** We demonstrate how PILING.JS' `arrangeBy` and `groupBy` methods work. (Top) The `arrangeBy('data')` subroutine expects an array of item property names (e.g., `distanceToDiagonal`, `noise`, and `size`) to arrange the piles by one, two, or more dimensions. Multidimensional cluster plots are realized with UMAP. (Bottom) The `groupBy` can take the current layout (e.g., grouping by overlap), item properties (e.g., grouping by `cellSubType` category), or similarity (e.g., grouping by cluster) into consideration. The cluster subroutine uses k-means clustering with $k = \max(2, \lceil \sqrt{|items|/2} \rceil)$ by default.

The image shows a graphical user interface for parameterization, divided into four main sections:

- View Properties:** Includes an 'Undo' button, 'Pile/Item' settings (itemSize, pileItemOffset), 'Layout' settings (itemSize, cellSize, cellPadding, columns, rowHeight, cellAspectRatio, pileCellAlignment), and 'Label' settings (pileLabel, pileLabelText, pileLabelAlign, pileLabelStackAlign, pileLabelFontSize, pileLabelHeight, pileSizeBadge, pileSizeBadgeAlign).
- Arrangement Operations:** Includes 'Arrangement' settings (arrangeBy, navigationMode), 'Pile By...' settings (Row, Column, Grid, Overlap, Distance, Category), and 'Split By' settings (Split All).
- Grouping Operations:** Includes 'Subset' settings (items) and 'Coloring' settings (poly-T hue, nuclei).
- Custom Settings:** A code block showing a JSON configuration for 'poly-T hue' with parameters like dtype, min, max, numSteps, defaultValue, onInput, and a setter function.

Figure S4: **Graphical User Interface for Parameterization.** PILING.JS implements several default settings and allows to define custom use-case specific settings. (Scaled-up version of Fig. 10.)

Table S1: **Piling.js Coverage of the Visual Piling Design Space.** An overview of piling.js' current support for the five dimensions of the visual piling design space.

| Grouping | |
|--------------------|---|
| Manual | Sequential grouping via a drag & drop gesture Sequential grouping via multi-selections (multiple mouse clicks while holding down the shift key) |
| Automatic | Parallel grouping via a lasso gesture Layout-driven grouping via <code>groupBy('row')</code> , <code>groupBy('column')</code> , or <code>groupBy('grid')</code> Proximity-based grouping via <code>groupBy('overlap')</code> or <code>groupBy('distance')</code> Similarity-based grouping via <code>groupBy('category')</code> or <code>groupBy('cluster')</code> |
| Arrangement | |
| Item | Random offset arrangement Orderly rule-based arrangement Orderly data-driven arrangement |
| Pile | Gridded linear ordering arrangement via <code>arrangeBy('index')</code> Gridded 2D spreadsheet-like arrangement via <code>arrangeBy('ij')</code> Precise arrangement via <code>arrangeBy('xy')</code> or <code>arrangeBy('uv')</code> Precise data-driven arrangement via <code>arrangeBy('data')</code> |
| Previewing | |
| Partial | Preview via partial item overlap |
| Gallery | Preview gallery of 2, 3, 4, 6, 8, or 9 thumbnails |
| Foreshortened | Preview via aggregation or a custom preview renderer |
| Combining | Preview via aggregation or a custom cover renderer |
| Indicating | Preview via custom preview renderer |
| Browsing | |
| In-place | Full item will appear at the top of the pile upon hovering the associated preview |
| Dispersive | Temporary dispersion into a regular 1D or 2D grid |
| Layered | Browse a pile in isolation via the context menu |
| Hierarchical | Nested browsing of sub-piles in isolation via the context menu |
| Aggregation | |
| Synthetic | Summary statistics (e.g., min, max, mean, median, or sum) Custom aggregator with a preview or cover renderer |
| Representative | Kmeans cluster centroids Custom aggregator |
| Simplistic | Custom aggregator with custom preview or cover renderer |