

Selective Angular Brushing of Parallel Coordinate Plots

R. Sahann¹ , I. Gajic¹, T. Möller^{1,2} , J. Schmidt³ 

¹University of Vienna, Faculty of Computer Science, Austria

²Data Science @ Uni Vienna, Austria

³VRVis Zentrum für Virtual Reality und Visualisierung Forschungs-GmbH, Austria

Abstract

Parallel coordinates are an established technique to visualize multivariate data. Since these graphs are generally hard to read, we need interaction techniques to judge them accurately. Adding to the existing brushing techniques used in parallel coordinate plots, we present a triangular selection that highlights lines with a single click-and-drag mouse motion. Our selection starts by clicking on an axis and dragging the mouse away to select different ranges of lines. The position of the mouse determines the angle and the scope of the selection. We refined the interaction by running and adapting our method in two small user studies and present the most intuitive version to use.

CCS Concepts

• **Human-centered computing** → **Visualization**; **Interaction techniques**;

1. Motivation

The visualization of multivariate data brings several challenges in terms of screen space and interaction. Parallel coordinates [Ins09] have become a well-known and increasingly used visualization technique for the visualization of multivariate data [JF16]. When using parallel coordinates in data analysis, interaction plays an important role [FL03] in enhancing analysis. Besides interacting with the axes (e.g., sorting), brushing is considered one of the most important interaction concepts in parallel coordinates plots. Brushing enables the users to select individual or multiple lines (i.e., data items) in a parallel coordinates plot. This is done by setting a *brush* (e.g., rectangle or lasso) and computing its intersection with the underlying data lines.

Multiple brushes [HW13] can be combined with logical operations (AND and OR) to further filter the data. For example, one could imagine defining two brushes—one for a specific range on axis x_1 and another for a specific range on axis x_2 . By combining them with a logical AND, the user will see all data defined within both ranges (i.e., all lines that “go through” both brushes). The definition of filter operations over multiple domains (i.e., axes) usually requires a sequence of different mouse interactions. In the previous example, users will have to define two brushes on two different axes, which require the same operation (click and drag) twice.

We present a novel method for selecting a subset of lines in a parallel coordinates plot. Our method allows users to specify the subset based on two axes but only one mouse operation. While a lasso selects all lines within the selected region, our techniques only selects lines in a particular region with a particular direction.

2. Related Work

Parallel coordinates are a well-studied technique [HW13], yet they are difficult to comprehend. Hence, several improvements have been suggested. For example, researchers proposed to apply illustrative representations [MM08] and focus+context techniques [NH06] to parallel coordinates to enhance perception. Density-based representations [HW09] and edge-bundling techniques [PBO*14] are proposed to deal with overplotting for large datasets. Effective interaction techniques, like selections, are crucial for users to deal with dense representations [Sii00].

In this paper, we specifically concentrate on brushing in parallel coordinates. Classical approaches include

- selecting values on an axis,
- range selection on axes, and
- brushing (e.g., using a line or lasso) in between axes.

In multi-dimensional settings, brushes can also be combined [MW95] to allow higher-order selections [War94, War97]. Multi-dimensional and higher-order brushing is an active research area. Here, researchers include additional guidance [RLS*19] to deal with high-density parallel coordinates [REB*16]. Another brushing technique is described by angular selection [HLD02] where users can define line subsets based on an angle at a specific point in the plot. Previous work suggests that interaction with parallel coordinates is intuitive and combining different methods for brushing can lead to increased performance for users [SR06]. In line with these findings, our new selection method can be seen as a combination of angular selection [HLD02] and multi-range selection [War97].

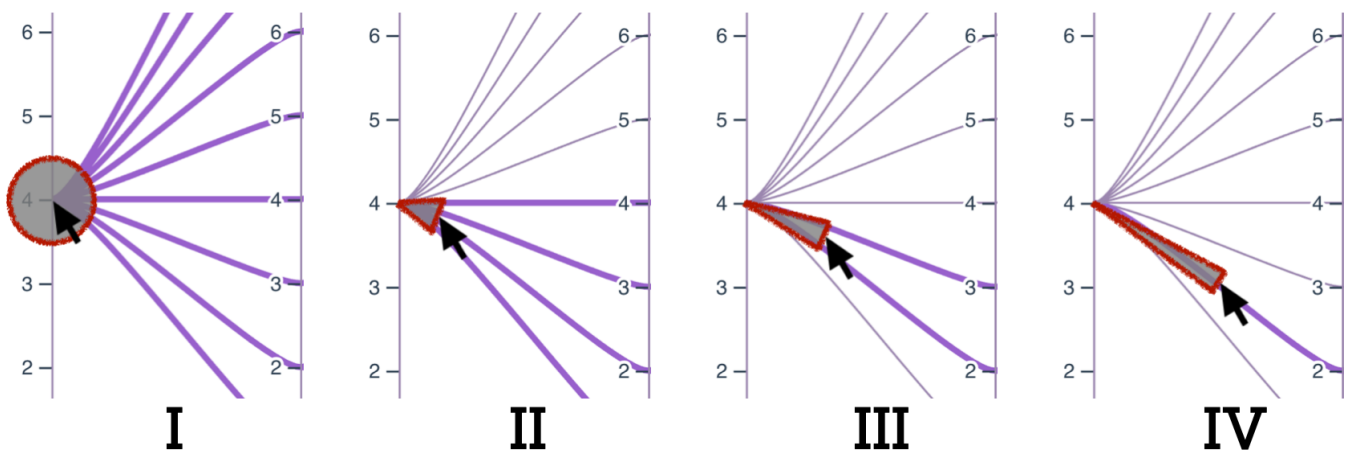


Figure 1: Selective Angular Brushing (basic concept). The selection method is shown in four steps—the red outlines have been added for clarity and are not part of the implementation. I) The interaction starts with a click on the axis, where all lines going through that point are highlighted. II) By dragging the mouse away from the axis, the circle selection changes to a triangle that highlights only lines beneath the triangle. III) Dragging the mouse even further narrows the region and in this way selects fewer lines. IV) Changing the angle is still possible anytime and also changes the selection.

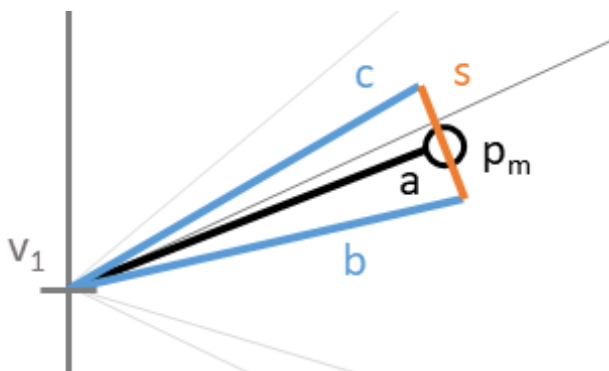


Figure 2: Basic concept. Starting from a value v_1 an imaginary line a can be drawn to the current mouse position p_m . Based on the angle between a and the axis, the lines b and c can be drawn. The endpoints of these lines define the line s , which is perpendicular to a and is located at the current mouse position.

3. Selection Design

Our novel method *Selective Angular Brushing* is based on two main concepts:

- Allow line selection based on the angle between two axes (similar to Hauser et al. [HLD02]).
- Enable users to do the selection with one single mouse operation.

The idea evolved out of working with data on student grades where the axes defined grades for specific lectures. In our case, users wanted to quickly verify whether students improved or deteriorated from one lecture to another. Basically, they wanted to verify "which lines move up from one specific point", "which lines are parallel", and "which lines go downwards". Using the classical concept of multi-range selections, it was necessary to operate with

two ranges on two axes, which was tedious. Therefore, we came up with a new solution to answer these questions more quickly.

3.1. Basic Concept

Selective Angular Brushing is based on the mouse operation click-and-drag. The different steps of the selection process are outlined in Figure 1. Selective Angular Brushing starts with the user clicking on one axis in a parallel coordinates plot (step I). This initially selects all lines going through this point, shown by a circle centered on the current value. By dragging the mouse away in a certain direction, only the lines matching the current angle are selected (step II). The mouse marker, which was originally a circle, changes its shape to a triangle. Moving the mouse further away from the initial click shortens the triangle's base, narrowing the selection (step III). An isosceles triangle with the apex at the initial click position and the center of the base at the mouse position shows the highlighted section. The mouse position can be changed at any time, which changes the angle to the original point and the triangle shape and, therefore, the selection (step IV). By now, releasing the mouse button applies the current selection, although other operations (e.g., filtering) may also be possible.

The main parameters for defining the selection are the distance and the angle between the axis and the current mouse position. Figure 2 shows the geometric concept. All data lines crossing the line s closest to the mouse position will be selected. In other words, it can be said that the triangle side s which is closest to the mouse pointer defines the selection interval. The whole selection area (consisting of the lines b , c , and s) is shown as an indicator. It should be noted, however, that the selection is not made by selecting all lines that somehow touch this area (compare step III in Figure 1).

The selection area is narrowed (i.e., the line s is shortened) the further the user moves the mouse away from the original click (compare step II and IV in Figure 1). When staying close to the

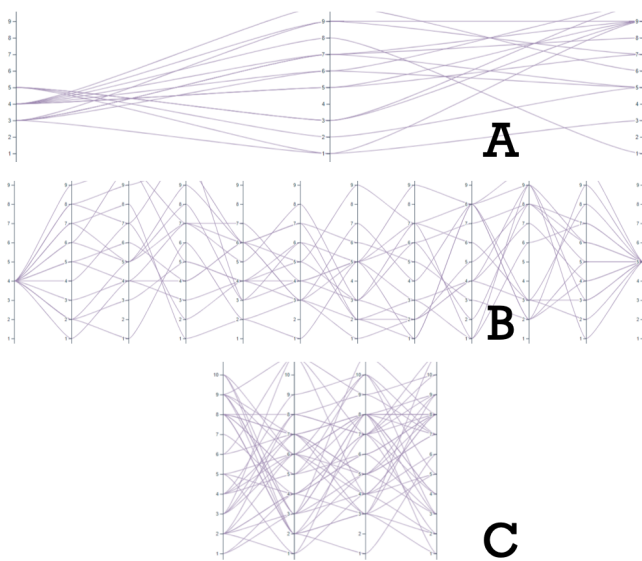


Figure 3: The three test datasets we used in the second evaluation, which were also used the first evaluation. A was previously called (viii) wide, B (ii) twelve axes, and C was referred to as (iv) 36 lines in the first evaluation. The focus of this evaluation was comparing the selection method with and without the triangular guide.

original click position, all lines going through this position are selected. When moving further away, only lines crossing the selection area stay selected. This means that more coarse selections are possible close to the original point, and more fine-grained selections are possible when moving further away. The change in the size of the selection area is up to the configuration done by the visualization designer and developer.

While this concept is easy to implement, we also wanted to evaluate whether Selective Angular Brushing is at the same time intuitive for users when interacting with parallel coordinates. We, therefore, conducted a user study on this (see Section 3.2). The results were promising and indicated that the concept is understandable even if the mouse markers (circle and triangle) are not shown.

3.2. Evaluation

Our evaluation of the intuitiveness of Selective Angular Brushing consisted of two steps. In our first evaluation we conducted ten interviews of users with at least some experience using parallel coordinate plots. Important to note, in this first iteration we did not include a visualization of the selection area, but rather just let the users use the click-and-drag mouse operation to highlight lines directly. We used nine different parallel coordinate plots with i. two axes, ii. twelve axes, iii. five lines, iv. 36 lines, a plot with three axes and 15 lines that was v. large, vi. small, vii. tight but high, viii. wide, and ix. a plot with actual student course data.

In individual digital interviews, which lasted approximately half an hour each, we tested whether selecting lines based on click-and-drag operations is intuitive to use. Participants were asked to share their screen. They received an initial explanation of the vi-

sualization and selection, and then interacted with the plots independently. No explicit tasks were given, so the testers were free to explore the interface and different data sets for up to 15 minutes. Afterward, we asked the users to rate the intuitiveness of using this technique in various parallel coordinate plots on a five-point Likert scale and if the plot's shape influenced their experience. Finally, we ended the interview with questions from the standard usability scale (SUS) [Bro13] test.

Our participants found the selection method generally intuitive to use, independent of size, shape, or number of lines (average 4.3 out of 5 points), but also reported, to some extent, that they would need the support of a technical person to be able to use the system (2.1 out of 5). Some users suggested that it would be more intuitive to use the system if some visual guidance would show the current selection. As a result, we adapted our implementation to include the triangular-shaped visual representation of selection area.

For the second evaluation, we recruited four participants, two of which already took part in the first round. Since the plots' size did not make a difference previously, we reduced the number of different charts to three, shown in Figure 3. Apart from the changes in charts the study setup remained identical to the first one. We also had the users compare the original version without visual guidance versus the selection area highlighting (see Figure 1). This evaluation scored slightly higher marks for usability when the sample included the visualization of the selection area. Interestingly, some users suggested that they liked using the visualization without the selection area being highlighted better—but only after they had a few minutes to use it with the visual guide to understanding how the mouse position is linked to the selected range.

We, therefore, conclude that Selective Angular Brushing is most intuitively used when the selection area is shown. In cases where the additional overlay might clutter the visualization, a short tutorial to familiarize the users with the selection method might be sufficient to use the selection without guidance.

3.3. Extension to axes range selections

When working with dense datasets, a lot of lines are shown in the plot, and then a simple click on an axis may not be enough for users to select the desired range of values. We, therefore, decided to extend Selective Angular Brushing to also work with range selectors [War97].

In this case, the selection with Selective Angular Brushing starts with a click on a previously set range selector placed on one of the plot axes. Again, a selection area is drawn when dragging the mouse away from the axes, and all lines following the same angle as the current selection are selected. Since we now start from a range selector, the selection area now is not a triangle any more, but rather a trapezoid. The area close to the mouse pointer becomes more narrow when moving away from the axis, so that more fine-grained selections are possible.

4. Implementation

We implemented two prototypes in web-based frameworks. To demonstrate the basic concept (as described in Section 3.1) we

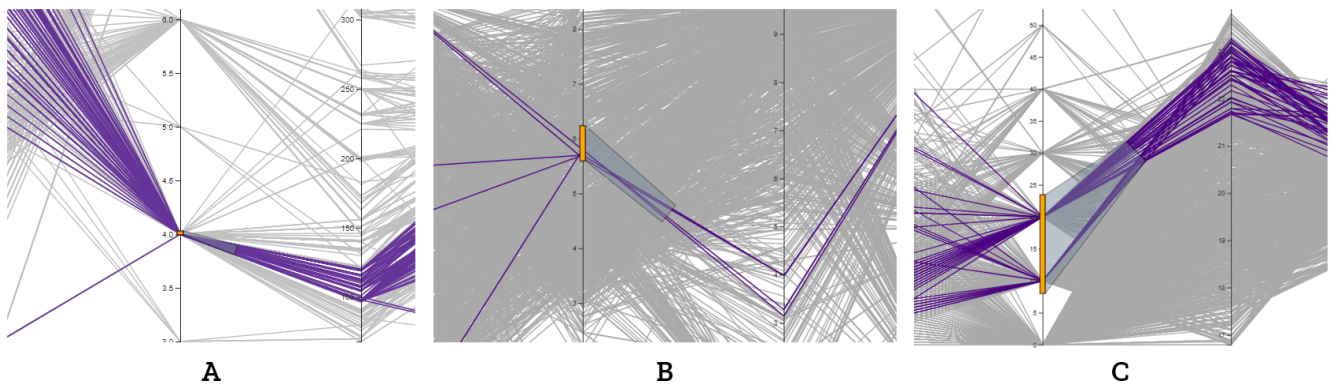


Figure 4: Selective Angular Brushing (range selectors). In case of dense data to be shown, range selection might be used instead of a single click on the axis. We could show that also in case of large datasets (A: 406 lines, B: 1,458 lines, C: 19,735 lines), Selective Angular Brushing provides an intuitive way to interact with the data.

used Vue.js [You21] and D3 [Bos21] version 6. This prototype was also used for the user evaluation (described in Section 3.2). We then implemented a second prototype to demonstrate the extension for range selections (described in Section 3.3) with D3 [Bos21] version 6 (for the interactive elements), WebGL [Khr21] (for the line rendering) and Angular [Goo21] version 11. The source code is available at <https://github.com/johanna-schmidt/selective-angular-brushing>.

5. Results

We tested Selective Angular Brushing with different datasets. The results can be seen in Figure 4. We used three different datasets to test our approach. In A a very small range selector is used, and therefore the selection is very similar to our initial point-based basic approach. The dataset used here contains specifications of cars from the 70s and 80s [Eva21] and consists of 406 lines. In B the range selector covers a larger area on the axis, which also affects the shape of the selection area. Even in a very dense data area, the data points with a specific angle can be nicely selected. In this case, we used data of the human freedom index [Kag21] which consisted of 1,458 lines. In C a large range selector is used, and again, with Selective Angular Brushing it is possible to extract the data items with specific characteristics based on the angle. Here we used a dataset consisting of different parameters for energy prediction [Kag18], which consisted of 19,735 lines. The results indicate that Selective Angular Brushing is a useful selection mechanism for both sparse and dense datasets.

Because of the simplicity of the interaction, there are a number of ideas on how Selective Angular Brushing can be extended and further evaluated:

- **Line rendering mechanisms** We would like to note that for our technique to work it is important that every polyline in the parallel coordinates plot touches its exact value at every axis. However, in other representations where, for example, smoothed curves [GK03] are used, it cannot be guaranteed that the curves touch the exact values. The initial click (basic concept) or a range selector might then not reach all intended lines.

- **Combinations** Selective Angular Brushing always starts at an axis, and could therefore easily be combined with a lasso brush—which can be applied whenever the click-and-drag begins in the space between two axes.
- **Scalability** We tested our approach with dense datasets (as seen in Figure 4). In case of many lines being drawn in the parallel coordinates plot, the performance of the technique depends on how fast the intersection of the selector and the lines can be calculated. In case of 19,735 lines, the calculation was still smooth in a JavaScript setting. In case larger datasets are used (e.g., more than 50,000 lines), parallel threads to compute the intersection, and faster drawing mechanisms (e.g., WebGL instead of SVG) are beneficial. The number of axes does not influence the performance, since only two axes are considered for calculating the selection.
- **Mobile devices** In the future we would like to explore the applicability of our approach on mobile devices with touch-and-drag functionalities.
- **Further evaluation** The user study we conducted was based on rather small and sparse datasets. In the future we would like to conduct further studies to evaluate the usefulness of our approach for the analysis of large datasets.
- **Other applications** Selective Angular Brushing works on line-based visualizations and could potentially also be useful for node-link diagrams and line charts.

6. Conclusion

This paper presents *Selective Angular Brushing*, a novel selection mechanisms for parallel coordinates based on a single click-and-drag mouse operation. Starting from a point or range on an axis, our technique allows to select all lines that follow a certain angle. This makes it possible to select all lines that ”go up” or ”go downwards” with one single operation, where previously multiple clicks (e.g., setting two ranges on two axes) were necessary. We conducted a user study to evaluate the intuitiveness of our approach. In the future, we would like to further study the applicability for mobile devices, and further evaluate the combination with other interaction and rendering techniques.

References

- [Bos21] BOSTOCK M.: D3.js - Data-Driven Documents . <https://d3js.org/>, 2021. [Accessed 2021-02-28]. 4
- [Bro13] BROOKE J.: Sus: A retrospective. *J. Usability Studies* 8, 2 (Feb. 2013), 29–40. 3
- [Eva21] EVANS C.: Parallel Coordinates. http://www.columbia.edu/~cme2126/datavisuals/bigdata_parallelcoordinates.html, 2021. [Accessed 2021-02-26]. 4
- [FL03] FERREIRA DE OLIVEIRA M., LEVKOWITZ H.: From visual data exploration to visual data mining: a survey. *IEEE Transactions on Visualization and Computer Graphics* 9, 3 (2003), 378–394. doi:10.1109/TVCG.2003.1207445. 1
- [GK03] GRAHAM M., KENNEDY J.: Using curves to enhance parallel coordinate visualisations. In *Proceedings of the 7th International Conference on Information Visualization* (London, UK, July 16–18 2003), IV '03, pp. 10–16. doi:10.1109/IV.2003.1217950. 4
- [Goo21] GOOGLE I.: Angular - The modern web developer's platform. <https://angular.io/>, 2021. [Accessed 2021-02-28]. 4
- [HLD02] HAUSER H., LEDERMANN F., DOLEISCH H.: Angular brushing of extended parallel coordinates. In *Proceedings of the IEEE Symposium on Information Visualization* (Boston, MA, USA, Oct. 28–29 2002), INFOVIS '02, pp. 127–130. doi:10.1109/INFVIS.2002.1173157. 1, 2
- [HW09] HEINRICH J., WEISKOPF D.: Continuous Parallel Coordinates. *IEEE Transactions on Visualization and Computer Graphics* 15, 6 (2009), 1531–1538. doi:10.1109/TVCG.2009.131. 1
- [HW13] HEINRICH J., WEISKOPF D.: State of the Art of Parallel Coordinates. In *STAR Proceedings of 34th Annual Conference of the European Association for Computer Graphics* (Girona, Spain, May 6–10 2013), Eurographics '13. doi:10.2312/conf/EG2013/stars/095–116. 1
- [Ins09] INSELBERG A.: *Parallel Coordinates: Visual Multidimensional Geometry and Its Applications*. Springer, 2009. 1
- [JF16] JOHANSSON J., FORSELL C.: Evaluation of parallel coordinates: Overview, categorization, and guidelines for future research. *IEEE Transactions on Visualization and Computer Graphics* 22, 1 (2016), 579–588. 1
- [Kag18] KAGGLE: Appliances Energy Prediction - Data driven prediction of energy use of appliances. <https://www.kaggle.com/loveall/appliances-energy-prediction>, 2018. [Accessed 2021-02-25]. 4
- [Kag21] KAGGLE: The Human Freedom Index - A global measurement of personal, civil, and economic freedom. <https://www.kaggle.com/gsutters/the-human-freedom-index>, 2021. [Accessed 2021-02-24]. 4
- [Khr21] KHRONOS M.: WebGL Overview. <https://www.khronos.org/webgl/>, 2021. [Accessed 2021-03-01]. 4
- [MM08] MCDONNELL K., MUELLER K.: Illustrative parallel coordinates. *Computer Graphics Forum* 27, 3 (2008), 1031–1038. doi:10.1111/j.1467-8659.2008.01239.x. 1
- [MW95] MARTIN A., WARD M.: High dimensional brushing for interactive exploration of multivariate data. In *Short Paper Proceedings of the IEEE Visualization Conference* (Atlanta, GA, USA, Oct. 30–31 1995), VIS '95, pp. 271–278. doi:10.1109/visual.1995.485139. 1
- [NH06] NOVOTNÝ M., HAUSER H.: Outlier-preserving focus+context visualization in parallel coordinates. *IEEE Transactions on Visualization and Computer Graphics* 12, 5 (2006), 893–900. doi:10.1109/TVCG.2006.170. 1
- [PBO*14] PALMAS G., BACHYNSKYI M., OULASVIRTA A., SEIDEL H., WEINKAUF T.: An Edge-Bundling Layout for Interactive Parallel Coordinates. In *Proceedings of the IEEE Pacific Visualization Symposium* (Yokohama, Japan, Mar. 4–7 2014), PACIFICVIS '14, pp. 57–64. doi:10.1109/PacificVis.2014.40. 1
- [REB*16] RAIDOU R., EISEMANN M., BREEUWER M., EISEMANN E., VILANOVA A.: Orientation-Enhanced Parallel Coordinate Plots. *IEEE Transactions on Visualization and Computer Graphics* 22, 1 (2016), 589–598. doi:10.1109/TVCG.2015.2467872. 1
- [RLS*19] ROBERTS R., LARAMEE R., SMITH G., BROOKES P., DCRUZE T.: Smart brushing for parallel coordinates. *IEEE Transactions on Visualization and Computer Graphics* 25, 3 (2019), 1575–1590. doi:10.1109/TVCG.2018.2808969. 1
- [Sii00] SIIRTOLA H.: Direct manipulation of parallel coordinates. In *Proceedings of the Conference on Human Factors in Computing Systems* (The Hague, The Netherlands, Apr. 1–6 2000), CHI '00, pp. 119–120. doi:10.1145/633292.633361. 1
- [SR06] SIIRTOLA H., RÄIHÄ K.: Interacting with parallel coordinates. *Interacting with Computers* 18, 6 (2006), 1278–1309. doi:10.1016/j.intcom.2006.03.006. 1
- [War94] WARD M.: XmdvTool: integrating multiple methods for visualizing multivariate data. In *Proceedings of the IEEE Conference on Visualization* (Washington, DC, USA, Oct. 17–21 1994), Visualization '94, pp. 326–333. doi:10.1109/VISUAL.1994.346302. 1
- [War97] WARD M.: Creating and Manipulating N-dimensional Brushes. In *Proceedings of the Joint Statistical Meeting* (Anaheim, CA, USA, Aug. 1997), ASA Proceedings, pp. 6–14. 1, 3
- [You21] YOU E.: Vue.js - The Progressive JavaScript Framework. <https://vuejs.org/>, 2021. [Accessed 2021-02-28]. 4