Palette Assistant

Generating Nominal Color Encodings with Simulated Annealing

CS 294-10: Visualization
UC Berkeley
Fall 2008

Calvin Ardi
Simon Tan
Ketrina Yim

http://sybak.com/NCEG

Problem

How can unique, quality color palettes for encoding nominal data be produced with minimal human intervention, while taking into account a designer's preferences?

Motivation

Color is an important component of most visualizations, especially as it is often used for encoding nominal (categorical) data. Much research has gone into developing automatic generation of color palettes for **quantitative** data. However, approaches to producing **qualitative** palettes (such as Cynthia A. Brewer's ColorBrewer) rely on experienced human designers to manually hand-pick colors that work well together.

ColorBrewer showing a designer-created color palette

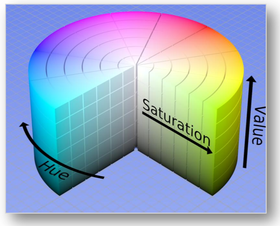
Nominal encodings using color require a well-designed color palette to effectively show data. However, the task of selecting a palette is often difficult for those who lack visualization design experience because there are a number of issues to address:

- Optimizing brightness and contrast between the colors
- Choosing a palette with "color harmony" (warm, cool, etc.)
- Considering compatibility with colorblind people

We seek to explore the use of simulated annealing (with heuristics based on sound color rules) in order to provide an automated way of generating palettes for nominal encodings without the need of a color expert.

Approach

We used simulated annealing to iteratively generate color palettes, traversing the color space by perturbing the Hue, Saturation, and Value aspects of each color from a starting palette. (The starting palette would vary depending on the "color harmony" preference.)



Each step in the simulated annealing was taken by randomly incrementing or decrementing the Hue, Saturation, or Value of a color in the palette

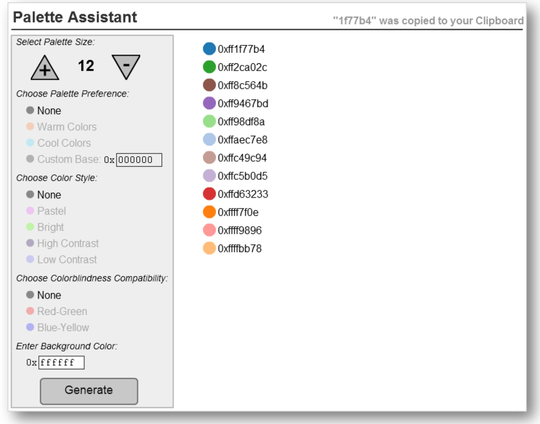
To evaluate the quality of each generated palette, we turned to Brewer's Color Use Guidelines (*Brewer, C. A. 1999*), which includes such principles as:

- Any combinations among red-orange-yellow-green are potentially confusing for colorblind people.
- Qualitative schemes benefit from small variations in lightness and saturation.
- Colors should be similar in contrast with the background and each other.

We wrote a heuristic function that would take in a color palette of arbitrary size and assign a score based on how well it fit with these guidelines and the preferences set by the user. Each guideline or option was represented as a module in the heuristic, and the modules were weighted based on our interpretation of how important they were. (See bottom.)

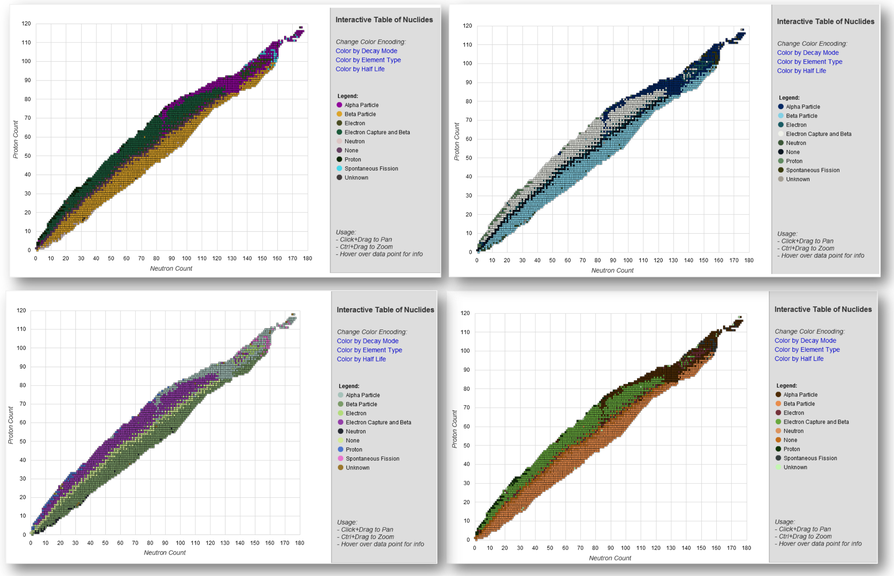
The scores were used to decide whether to accept or reject each new generated palette at each step of the simulated annealing. After many (thousands) of iterations, we returned the resulting colors formatted for use in our wrapper application or any other visualization built on the Flare library.

Weighted heuristic modules combine to produce a score for any given palette



Our generator was developed as a library alongside the Flare visualization library, and we built a Flex application around it that allowed users to input their preferences with a graphical user interface

Sample Results



Future Work

Although our system does produce reasonably fine palettes, the results could still be improved and the system could be made more useful by:

- Considering actual samples of data in order to incorporate heuristics such as color assignment based on the relative size of a category
- Applying machine or supervised learning techniques to determine optimal weights for each of the heuristic modules
- Allowing the ability to "lock" a subset of desirable colors on a palette and algorithmically re-generate the rest