Research Artifact: Color Schemes

Conor O'Brien

November 7, 2023

Contents

1 Introduction

As developers on the CSCI 435 Project 4 Astxplainer Visualization team, we need to research potential color schemes for use in our project.

2 Color Schemes

We have 4 primary potential use cases for color schemes:

1. Node probabilities (0.00 to 1.00) are *continuous*.
2. Node types (e.g. at least leaf node vs tree with children) are *categorical*.
3. Node focus (default, hover, and highlighted) is *categorical*.
4. Overall color scheme (light vs dark) is *categorical*, but determined.

Our color schemes should be readable and accessible (e.g. colorblind safe, grayscale friendly).

2.1 Continuous color schemes

Pelacio et al. give an unclassed diverging quantitative color scale (featured pp. 5-6, 8) for coloring node probabilities [6]. This scale roughly goes from dark red (0.0), to light red (0.25), to white (0.5), to light blue (0.75), to dark blue (1.0), and appears to correspond with the RdBu diverging colormap Matplotlib provides [2].

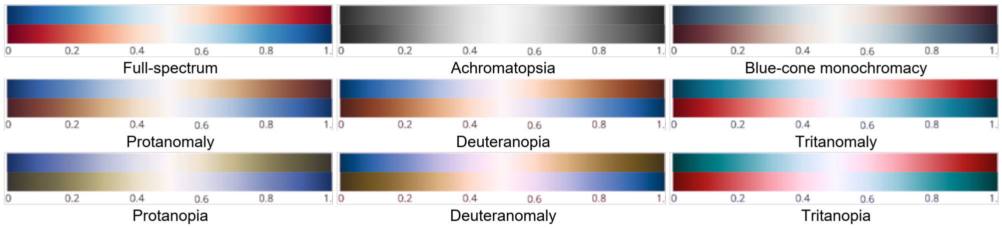

Figure 1: Coblis tests on RdBu colormap, with reversed scale arrayed above scale for ease of comparing distinctness.

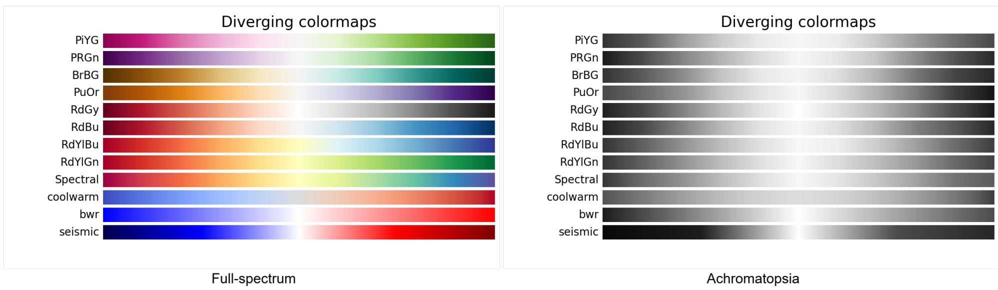

Figure 2: An achromatopsia vision view of the diverging colormaps Matplotlib provides.

Under the Coblis colorblindness tests [1], RdBu is distinguishable at all presented simulated visual perceptions besides achromatopsia (Figure 1). Practically, this means while the color spectrum is mostly accessible, it becomes inaccessible in achromatic environments, such as achromatopsia affected individuals, as well as grayscale printings. This makes it suboptimal for our purposes, and a major interest of our research.

This is primarily due to the two primary divergent tones having indistinguishable saturations. This can be fixed by two approaches: Either using a sequential color scale, or varying saturation continuously throughout the scale. While there are a plethora of accessible sequential color scales available already, there are no popular diverging colormaps shipped with Matplotlib that are accessible to achromatopsia vision (Figure 2). We therefore shall consider a novel class of colormaps which seeks to address these issues.

2.1.1 Achromatopsia Linear Diverging Colormaps

To investigate our second option more thoroughly, we devise diverging colormaps which, under an achromatopsia vision view, transform to linear colormaps.

The nature of a diverging colormap is to have two defined extrema with some more muted middleground to differentiate them [5]. Naturally, this is not possible in grayscale. Although a classical diverging colormap cannot be mapped directly to grayscale, a sequential grayscale colormap can still communicate the same information.

For the purposes of this investigation, we approximate achromatopsia using Coblis's mapping function over RGB colorspace (ranging 0.0 to 1.0) [3]:

$$e = A(r, g, b) = 0.299r + 0.587g + 0.114b$$
$$(r', g', b') = (e, e, e)$$

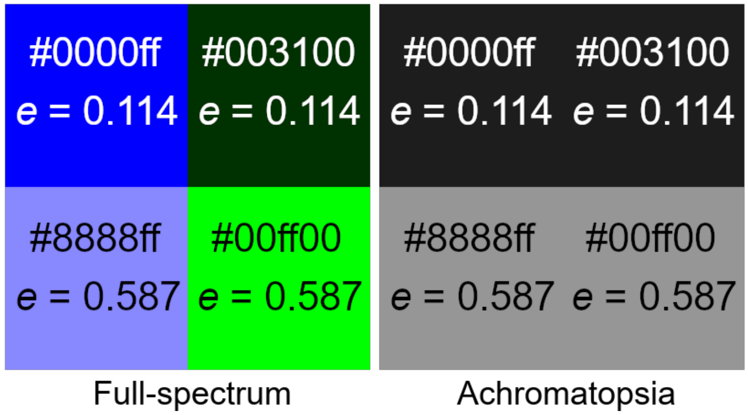

Figure 3: A full intensity blue and a dark, near-black green map to the same perceived color ($e = 0.114$) under simulated achromatopsia vision.

We wish to investigate satisfying both the characteristics of a full-spectrum diverging colormap and an achromatopsia sequential colormap simultaneously. Thus, we wish to pick appropriate colors for our diverging colormap for whom the e values increase consistently. For example, for a 3 color diverging colormap, we will have three colors C_1, C_2, and C_3 for which the corresponding e_1, e_2, and e_3 with some positive constant Δ such that $e_{n+1} = e_n + \Delta$.

Note that the color that contributes the least in achromatopsia vision is blue. In that context, a bright blue is comparable to a dark green (Figure 3). This makes shades of blue ideal for our lower extremum. Likewise, as green contributes the most, we get a bright color in both full-spectrum and anchromatopsia vision.

In general, there is an entire plane of solutions which map to a specific e-value. We are interested especially in a full-spectrum vision bright color in the center with $e \approx 0.5$. Additionally, we are interested in equally dark, differing full-spectrum vision colors on the extrema with low and high e values respectively.

To begin, let us arbitrarily choose $e_k = (0.114, 0.5, 0.886)$ with $\Delta = 0.386$, which lets us choose $C_1 = (0, 0, 1.0)$. (We do not want to choose much lower e_1 or much higher e_3, since we want to use *colors*, rather than eccentric shades of black or white.) For our first attempt, let us mirror Matplotlib's RdBu, and arbitrarily choose C_3 as some shade of red. Since our target $e_3 = 0.886$, and we want a shade of red (i.e. $r \gg g = b$), we have to solve

$$\begin{aligned} 0.886 &= A(r, g, g) = 0.299r + 0.587g + 0.114g \\ &= 0.299r + 0.701g \\ 0.701g &= 0.886 - 0.299r \\ g &= \frac{0.886 - 0.299r}{0.701} \\ &= 1.26391 - 0.426534r \end{aligned}$$

Maximizing $r = 1$ gives $g = b = 0.837375$, which is a bright, pale peach (`#ffd5d5`). We may also employ linear programming to generate other candidates. Minimizing $r + g + b$ with $A(r, g, b) = 0.886$ gives $(r, g, b) = (1, 1, 0)$, which is bright yellow. Instead minimizing relative luminance ($L = 0.2126r + 0.7152g + 0.0722b$) gives $(r, g, b) = (1, 0.8058, 1)$, a shade of pink. An example minimizing luminance but instead targeting $e_3 = 0.7$ for a *de facto* darker shade, also with the additional constraint $g = b$, gives $(r, g, b) = (1, 0.572, 0.572)$; this is included for illustration purposes in Figure 4.

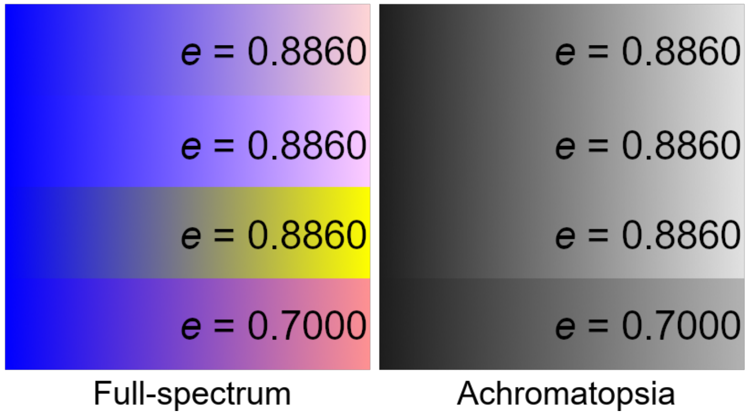

Figure 4: Bicolor colormap schemes targeting $e_k = (0.114, 0.886)$. From top to bottom: Ach-Blue-Peach, Ach-Blue-Pink, Ach-Blue-Yellow, and Ach-Blue-Salmon.

Any linear RGB interpolation between two colors also interpolates the corresponding e values. Let $I(C_1, C_2, p)$ be color which lies at proportion p between the linearly interpolated colors C_1 and C_2; representing each color as a vector, we have $I(C_1, C_2, p) = C_1 + p(C_2 - C_1)$, $A(C) = W \cdot C$ for $W = [0.299, 0.587, 0.114]$, and

$$\begin{aligned} A(I(C_1, C_2, p)) &= W \cdot (C_1 + p(C_2 - C_1)) \\ &= W \cdot C_1 + W \cdot p(C_2 - C_1) \\ &= W \cdot C_1 + pW \cdot C_2 - pW \cdot C_1 \\ &= A(C_1) + pA(C_2) - pA(C_1) \\ &= A(C_1) + p(A(C_2) - A(C_1)) \\ &= I(A(C_1), A(C_2), p). \end{aligned}$$

Thus, any choice of RGB colors will always smoothly interpolate under our simulated achromatopsia vision, and any smooth RGB interpolation is also smooth under achromatopsia vision.

While these appear to be adequate bicolor colormap schemes, looking ahead to introducing the center color C_2, results are not promising. We need to now target midpoints e_2 using similar approaches to before. Supposing we now want a maximal green, $g = 1$ is actually too large, so our previous analytical method fails here; with a slight adjustment, simple algebra reveals $0.587g = 0.5 \iff g = 0.852$. Minimizing $r + g + b$ targeting $e_2 = 0.5$ via linear programming, agnostic to our desire of green, however, gives the same solution: $(r, g, b) = (0, 0.852, 0)$, an acceptable forest green. Instead minimizing luminance as before, we get $(r, g, b) = (1, 0.148, 1)$, a strong magenta. Instead targeting $e_2 = 0.407$ to compensate for the lower $e_3 = 0.7$ value we used illustratively, we get $(r, g, b) = (0, 0.693, 0)$ when minimizing $r + g + b$, a slightly darker forest green, and $(r, g, b) = (0.980, 0, 1)$ when minimizing luminance, an even stronger magenta.

Holding fixed C_1 and e_k, we have our choice between three right endpoint colors (peach, pink, yellow) and two midpoint colors (green and magenta); with our compromise on $e_3 = 0.7$, we investigated one choice of endpoint color (salmon) and two choices of midpoint color (darker shades of green and magenta); all for a total of 8 possible combinations under consideration. The results of this colormap investigation are contained in Table 1 and Figure 5.

Colormap name	C_2	C_3	e_2	e_3
A-BGP: Ach-Blue-Green-Peach	$(0.000, 0.852, 0.000)$	$(1.000, 0.837, 0.837)$	0.500	0.886
A-BMP: Ach-Blue-Magenta-Peach	$(1.000, 0.148, 1.000)$	$(1.000, 0.837, 0.837)$	0.500	0.886
A-BGK: Ach-Blue-Green-Pink	$(0.000, 0.852, 0.000)$	$(1.000, 0.806, 1.000)$	0.500	0.886
A-BMK: Ach-Blue-Magenta-Pink	$(1.000, 0.148, 1.000)$	$(1.000, 0.806, 1.000)$	0.500	0.886
A-BGY: Ach-Blue-Green-Yellow	$(0.000, 0.852, 0.000)$	$(1.000, 1.000, 0.000)$	0.500	0.886
A-BMY: Ach-Blue-Magenta-Yellow	$(1.000, 0.148, 1.000)$	$(1.000, 1.000, 0.000)$	0.500	0.886
A-BGS: Ach-Blue-Green-Salmon	$(0.000, 0.693, 0.000)$	$(1.000, 0.572, 0.572)$	0.407	0.700
A-BMS: Ach-Blue-Magenta-Salmon	$(0.980, 0.000, 1.000)$	$(1.000, 0.572, 0.572)$	0.407	0.700

Table 1: Textual description of achromatopsia linear diverging colormaps for $C_1 = (0, 0, 1)$ and $e_1 = 0.114$.

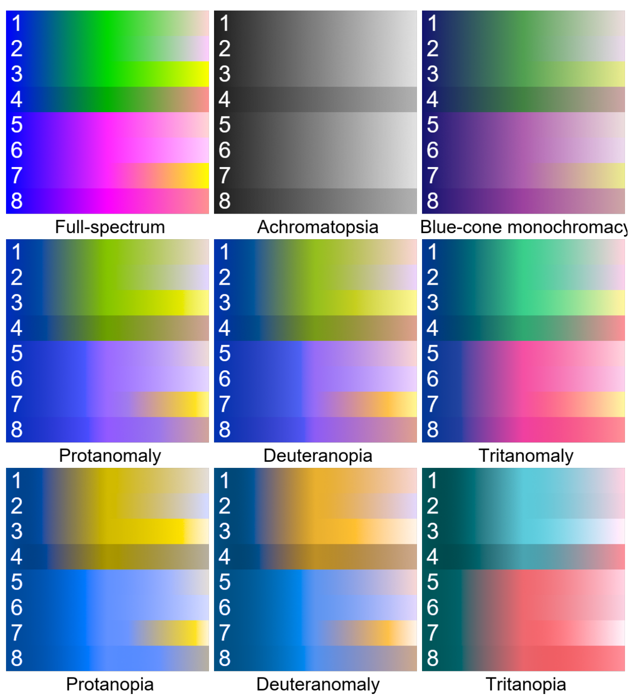

Figure 5: Coblis tests on our achromatopsia linear diverging colormaps for $C_1 = (0, 0, 1)$ and $e_1 = 0.114$. Numbered: (1) Ach-Blue-Green-Peach, (2) Ach-Blue-Green-Pink, (3) Ach-Blue-Green-Yellow, (4) Ach-Blue-Green-Salmon, (5) Ach-Blue-Magenta-Peach, (6) Ach-Blue-Magenta-Pink, (7) Ach-Blue-Magenta-Yellow, and (8) Ach-Blue-Magenta-Salmon.

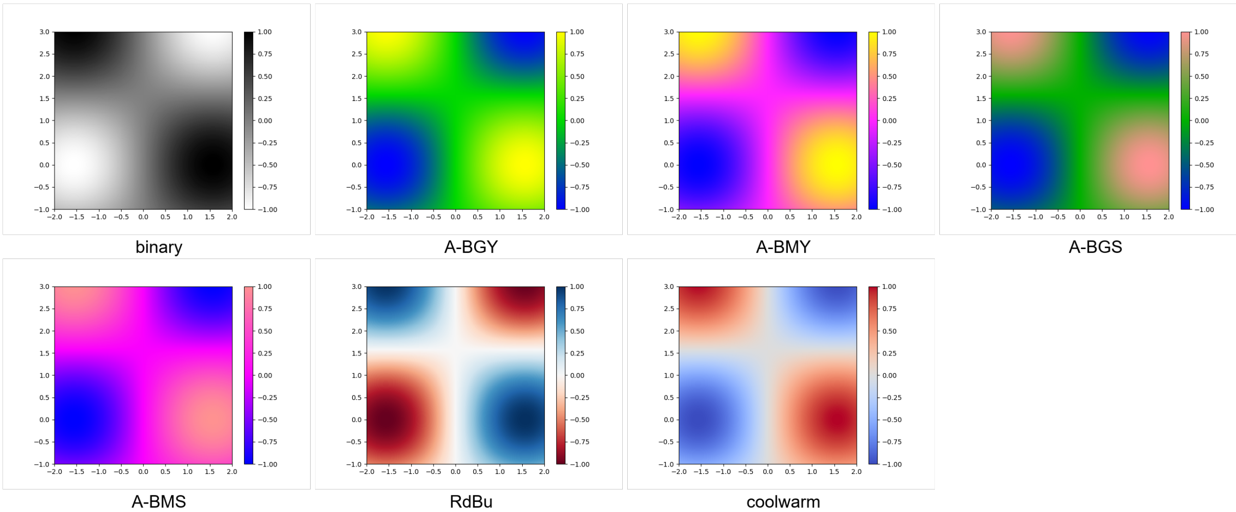

Figure 6: Tested colormaps visualizing $z = \sin x \cos y$ over $x \in [-2, 2]$, $y \in [-1, 3]$, and $z \in [-1, 1]$ given to participants.

2.1.2 Evaluation

Of the 8 candidate colormaps, we arbitrarily select 4: Ach-Blue-Green-Yellow, Ach-Blue-Magenta-Yellow, Ach-Blue-Green-Salmon, and Ach-Blue-Magenta-Salmon. We judged these to have the widest breadth of range across all 9 simulated vision spectra. To further evaluate these achromatopsia linear diverging colormaps, we distributed a survey online asking recipients to rank these 4 colormaps, as well as 3 control colormaps from Matplotlib: `binary`, `RdBu`, and `coolwarm`. The survey was open for 24 hours, from November 5th, 2023 to November 6th, 2023, and reached 60 participants with a response rate of 93.33% ($N = 56$).

The main bulk of the survey itself consisted of 7 questions asking participants to rate the 7 aforementioned colormaps on a discrete linear scale from 1 ("Awful") to 10 ("Perfect"). Each colormap was projected on the same graph Figure 6. After that, we collected one mandatory demographic datum, whether or not the respondent was colorblind or colordeficient, and if so, in what way; we also collected three optional demographic data: Age band (18 to 24, 25 to 44, 45 to 64, and 65 or older), self-described technology proficiency (linear discrete scale from 1 ("Not proficient") to 5 ("Proficient")), self-described familiarity with data visualization (linear discrete scale from 1 ("Not very familiar") to 5 ("Very familiar")), and gender (1 or more from Male, Female, Non-binary, Transgender, Genderqueer, Genderfluid, Two-Spirit, Agender, Bigender, Prefer not to say, and a fillable Other option).

We predicted that the control colormaps would significantly outperform our novel colormaps, due to their lack of aesthetic appeal compared to conventional diverging colormaps. Of our colormaps, we predicted A-BMY (Ach-Blue-Magenta-Yellow) would perform the best, given that its colors are the most distinct and harmonious.

Of the 60 polled individuals, $N = 567$ assented to the informed consent of the survey and provided responses. 2 individuals identified themselves as having Protanomaly (Red-Weak) colorblindness; the rest identified as having full-spectrum vision. Of our polled demographic data, the average identified technology proficiency was 4.18 (scale 1 to 5), and the average identified data visualization familiarity was 2.69 (scale 1 to 5). As for age, 23 respondents identified as being in the 18 to 24 age range, 14 in the 25 to 44 age range, 12 in the 45 to 64 age range, 6 in the 65 or older age range, and 1 preferred not to answer.

Colormap name	Average Rating	Mode Rating	Proportion of ratings ≥ 5
`binary`	5.66	**8**	0.6786
`RdBu`	6.14	5	**0.7857**
`coolwarm`	**6.48**	5	**0.7857**
A-BGY	4.93	4	0.4821
A-BMY	**5.55**	5	0.6429
A-BGS	3.82	4	0.3393
A-BMS	5.14	**6**	**0.7857**

Table 2: Survey results. $N = 57$, excluding non-respondents.

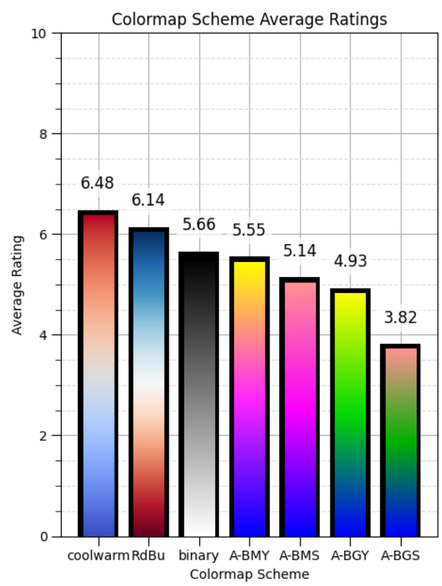

Figure 7: Average ratings per color scheme.

Our survey results (Table 2) show a slight preference amongst participants for the control colormaps, although the difference between the control colormap `binary` and our novel colormap A-BMY is minuscule (5.66 to 5.55) (see Figure 7). There is a significant dislike of A-BGS, and comparing averages of `binary`, A-BMY, A-BMS, and A-BGY all lie within an interval of size 0.73.

However, we can get a clearer idea of the distribution of preferences through the more precise view of the rating distribution Figure 8 offers. From this, we can extract that the distributions participants appraised most often with at least a rating of 5 were `RdBu` and `coolwarm` in the control group and A-BMS in the novel group. Generally, all distributions besides the green-centred novel distributions are appraised with 5 or higher by around at least two-thirds of the participants.

2.2 Categorical color schemes

Choosing accessible categorical color schemes is a relatively simpler task. Choosing N different colors that both look good in full-spectrum vision as well as distinct in other kinds of vision is simple for relatively small N.

One thing to keep in mind is that the most accessible color schemes use non-color markings to

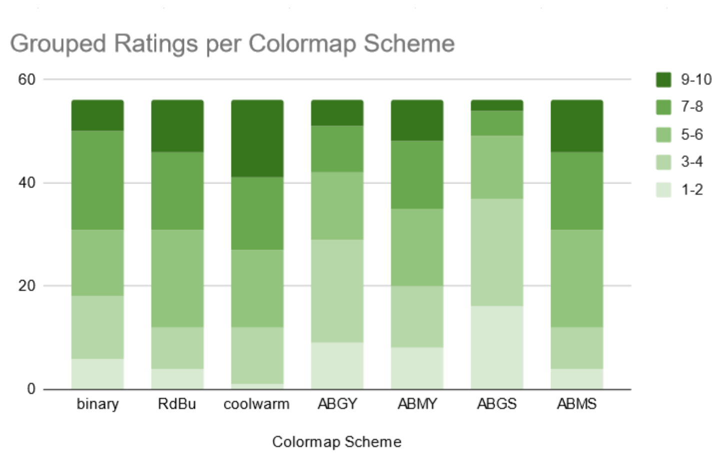

Figure 8: Distribution of votes, grouped in consecutive buckets of size 2.

aid visualization, such as designing with shapes [4]. The benefits reaped from visualizations where this is applicable applies also to full-spectrum vision, increasing readability.

Thus, ideally, for each category, there is some accompanying non-color aid to help differentiate. Even when the information is technically parseable without the use of color, where color helps, colorblind people would still appreciate some non-color aid to help them parse the information easier.

For our three categorical use cases, this means:

1. Node types may have different colors to help users distinguish between different node types at a glance. Importantly, this means there should be accompanying symbols.

2. Node focus may have different colors to help draw attention to focused elements, or prospective focused elements. This can be made accessible with underlines.

3. Dark mode versus light mode is intrinsically accessible, done correctly.

Most important to figure out for the moment is a prospective categorical color palette and an accompanying set of symbols. First, let us begin by evaluating the qualitative colormaps Matplotlib provides. See Figure 9. Note that no colormap is fully accessible, and even limiting our categories to 4 or fewer, the only ones that succeed universally are the initially monochrome schemes `tab20b` and `tab20c`; `tab20` almost succeeds but for anchromatopsia vision.

While we could fix this with a similar approach we applied for achromatopsia, an easier fix would be to integrate symbols into our color scheme visualization. Such symbols are readily available, depending on context. Differing border styles, a color-agnostic symbolic tag in addition to a color, etc. There are no shortage of such strategies, but it is hard to forecast which symbols will be most useful at this time.

3 Discussion

Our methodology and conclusions are limited by having to rely on simulated accessibility metrics. In a perfect world, we would be able to survey individuals with each of at least the most common vision types besides full-spectrum vision. A survey on that level is certainly beyond our resources.

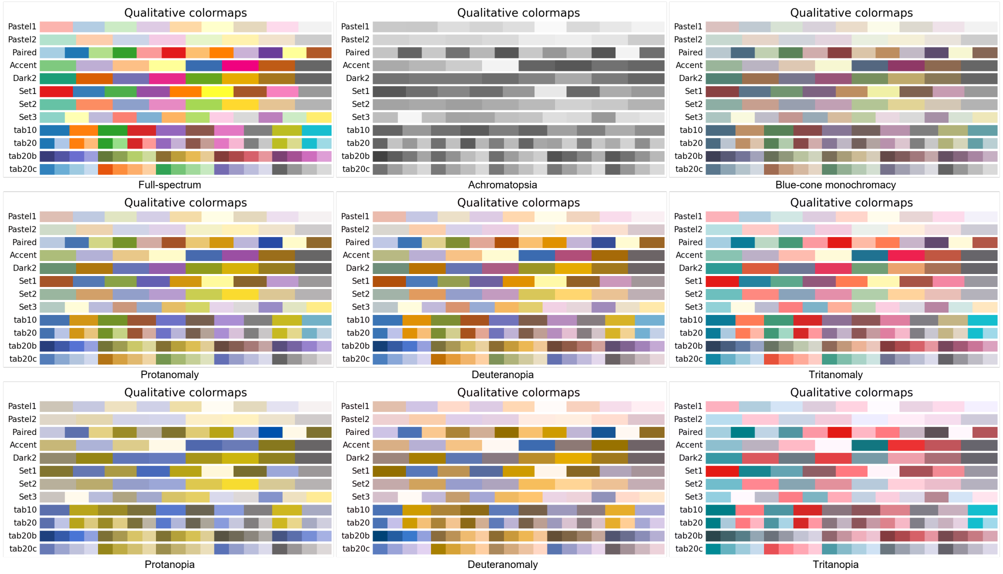

Figure 9: Coblis tests on the qualitative colormaps Matplotlib offers.

While our novel diverging colormaps are not the most aesthetically pleasing, they are useful inclusions in a set of color palettes. They also provide for an interesting framework for creating accessible color schemes which double as readable in full-spectrum views. More advanced linear programming and solution finding could be used to handle simulated vision environments other than achromatopsia, and could provide additional dual-accessible solutions.

4 Conclusion

We introduced a family of novel diverging colormaps which are linear under an achromatopsia view. We chose 4 colormaps from this family and evaluated them against 2 standard diverging colormaps and one linear colormap, all available by default in Matplotlib. We evaluated the chosen colormaps through a survey, which showed that participants generally preferred the standard colormaps, but also found two of our novel colormaps to be acceptable, being A-BMS and A-BMY; participants decidedly did not generally like A-BGS and A-BGY.

We identify and reiterate the need for non-color indicators in the context of color used to differentiate different categories.

References

[1] Color-Blindness.com. COBLIS Color Blindness Simulator, Accessed on 2023-11-05.

[2] The Matplotlib development team. Choosing Colormaps in Matplotlib, Accessed on 2023-11-05. Documentation. Accessed on 2023-11-05.

[3] MaPePeR. jsColorblindSimulator, August 16, 2022. GitHub Repository. Accessed on 2023-11-05.

[4] Federico Monaco. Color blind accessibility manifesto. *Commun. ACM*, 65(8):7, jul 2022.

[5] Lisa Charlotte Muth. Which color scale to use when visualizing data, March 16th, 2021. Blog Post. Accessed on 2023-11-05.

[6] David N Palacio, Alejandro Velasco, Daniel Rodriguez-Cardenas, Kevin Moran, and Denys Poshyvanyk. Evaluating and explaining large language models for code using syntactic structures, 2023.