U-CIE [/juː 'siː/]: Color encoding of high-dimensional data

Mikaela Koutrouli[1], John H. Morris[2] and Lars J. Jensen[1]

[1] Novo Nordisk Foundation Center for Protein Research, Faculty of Health and Medical Sciences, University of Copenhagen

[2] Resource on Biocomputing, Visualization, and Informatics, University of California, San Francisco, California

Abstract

U-CIE is a visualization method that encodes arbitrary high-dimensional data as colors using a combination of dimensionality reduction and the CIELAB color space to retain the original structure to the extent possible. We illustrate its broad applicability by visualizing single-cell data on a protein network and metagenomic data on a world map and on scatter plots. U-CIE is available as a web resource at https://u-cie.jensenlab.org/ and as an R package.

Main

Today large, high-dimensional datasets are abundant in biomedicine. Data visualization is thus crucial both for discovering patterns in data and for subsequently communicating the insights. With this motivation, we present U-CIE [/juː 'siː/] an open-source software tool that translates high-dimensional data into colors. As color space is inherently three-dimensional, U-CIE does this in two steps: a three-dimensional approximation of the high-dimensional data is produced using a dimensionality reduction method, and this approximation is next fitted into a suitable color space (Figure 1). Each high-dimensional input vector, be it the expression profile of a gene or the abundance profile of an organism, is thereby converted into a color. These colors can subsequently be used to visualize the, e.g., genes or organisms in the context of other data, such as gene networks, phylogenetic trees, or longitude and latitude.

Dimensionality reduction techniques have improved much in recent years, becoming better at preserving both local and global structure of the high-dimensional data. Transformations like t-SNE [1] and UMAP [2] as well as generative deep-learning models, such as variational autoencoders [3], have become particularly popular within biology, especially for visualization of single-cell data and data from metagenomics studies. In this work, we have opted to use UMAP as it partially preserves global Euclidean distances; however, one can trivially replace it with any other dimensionality reduction technique.

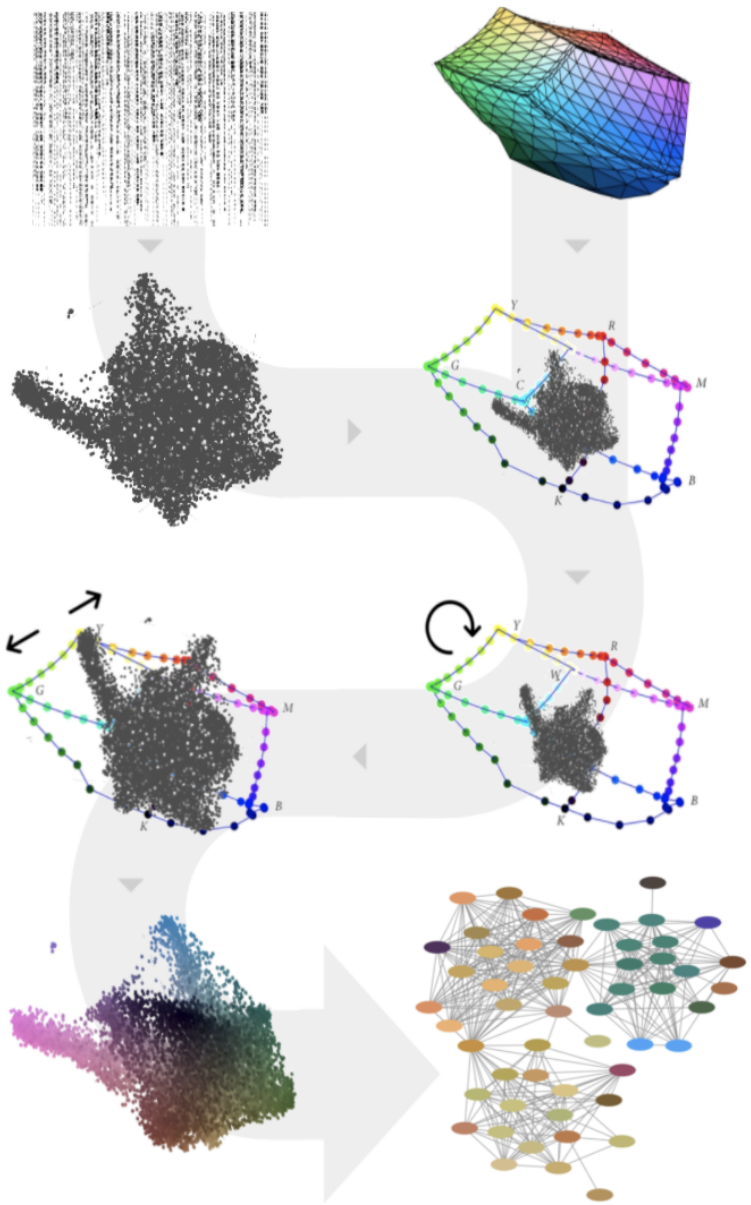

Figure 1. **Overview of the U-CIE algorithm and application to single-cell RNAseq data.** U-CIE uses a two-step process to encode high-dimensional data as colors. The first step is to use a state-of-the-art dimensionality reduction technique (UMAP) to reduce the data to three dimensions, since human color perception is inherently three-dimensional. The second step is to fit the resulting point cloud into the three-dimensional polygon that represents the displayable part of CIELAB color space. This is done as an optimization process, which shifts, rotates, and uniformly scales the point cloud to make it as large as possible while penalizing for points protruding outside the polygon. Finally, we show how the colors summarizing gene expression can be visualized on a gene network, e.g. from the STRING database [4]. The network visualization was done in Cytoscape [5].

Colors are everywhere and we all have an intuitive understanding of colors; however, it is surprisingly difficult to represent them in a way that accurately reflects how similar humans perceive any two colors to be. None of the most commonly used color representations, such as RGB, approximate human vision. A notable exception is the CIELAB color model [6], which was designed to be largely perceptually uniform, meaning that the Euclidean distance between any two points should correspond to their relative perceptual difference. This makes it particularly useful for our application, since we want to visualize positions as colors, similar to what has previously been tried for self-organizing maps [7]. CIELAB represents

colors using three values: perceptual lightness (L*) and two pairs of complementary colors, namely green–red (a*) and blue–yellow (b*). To take advantage of the properties of CIELAB, we need to convert the three-dimensional coordinates obtained from dimensionality reduction into L*a*b* coordinates. This is nontrivial, because not all combinations of L*, a*, and b* result in colors that can be displayed on a computer monitor. For this reason, we use an optimization algorithm to shift, rotate, and uniformly scale the point cloud to fit it within the displayable part of CIELAB (Figure 1). The result is an encoding of the input data, where each data point has been assigned a color, and similarity between colors reflect the distance between the original points.

Having a color encoding of high-dimensional data is useful whenever you want to visualize it in the context of some other information, since it frees up the spatial coordinates. For example, transcriptomics and proteomics data are commonly visualized onto gene/protein networks. Figure 1 exemplifies the U-CIE method by applying it to network visualization of single-cell RNA-seq (scRNA-seq) data. The scRNA-seq dataset provides data for 19,097 genes across 1018 cells, which were converted to a color for each gene and visualized on a physical protein interaction network from STRING v11.5 [4]. This allows us to graphically summarize the transcriptional regulation of the individual subunits within protein complexes. Figure 2 shows a completely different use case for U-CIE, namely visualization of microbiome compositions on a world map and on scatter plots. Specifically, we used U-CIE to color the sampling stations of the Tara Oceans project [8] based on their 16S taxonomic profiles from surface water samples. Comparing these colors with the longitude and latitude of the sampling stations on a world map shows that the samples from some oceans, e.g. the Mediterranean Sea, have quite similar microbiome composition (Figure 2A). The colors can also be interpreted in terms of environmental factors that drive microbiome composition. The lightness axis (L*) correlates most strongly with dissolved oxygen concentration, the green–red axis (a*) with dissolved nitrite/nitrate concentration, and the blue–yellow axis (b*) with water temperature (Figure 2B–D).

To allow anyone to use U-CIE, we have made it available as an interactive web resource (https://u-cie.jensenlab.org) along with a guided tutorial. It offers four different tracks: ‘Single cells’, ‘High dimensional’, ‘Distance matrix’, and ‘3D data’. All use UMAP to first reduce dimensionality, except from the last track, which allows the user to upload data that is already three-dimensional, possibly produced using another dimensionality reduction method. To exemplify the functionality of the web interface, we show how the analysis in Figure 1 can be performed (Supplementary Figure S1). First, we upload the matrix with expression counts to the ‘Single cells’ track, which assigns colors to genes. A preview of the uploaded data is first shown, allowing us to verify that it was parsed correctly before starting the analysis. As soon as the analysis has finished, we gain access to interactive visualization panels that show the color encoding in 3D and as 2D projections, controls to alter the color encoding, and the option to download a tab-delimited file with the gene names and their colors.

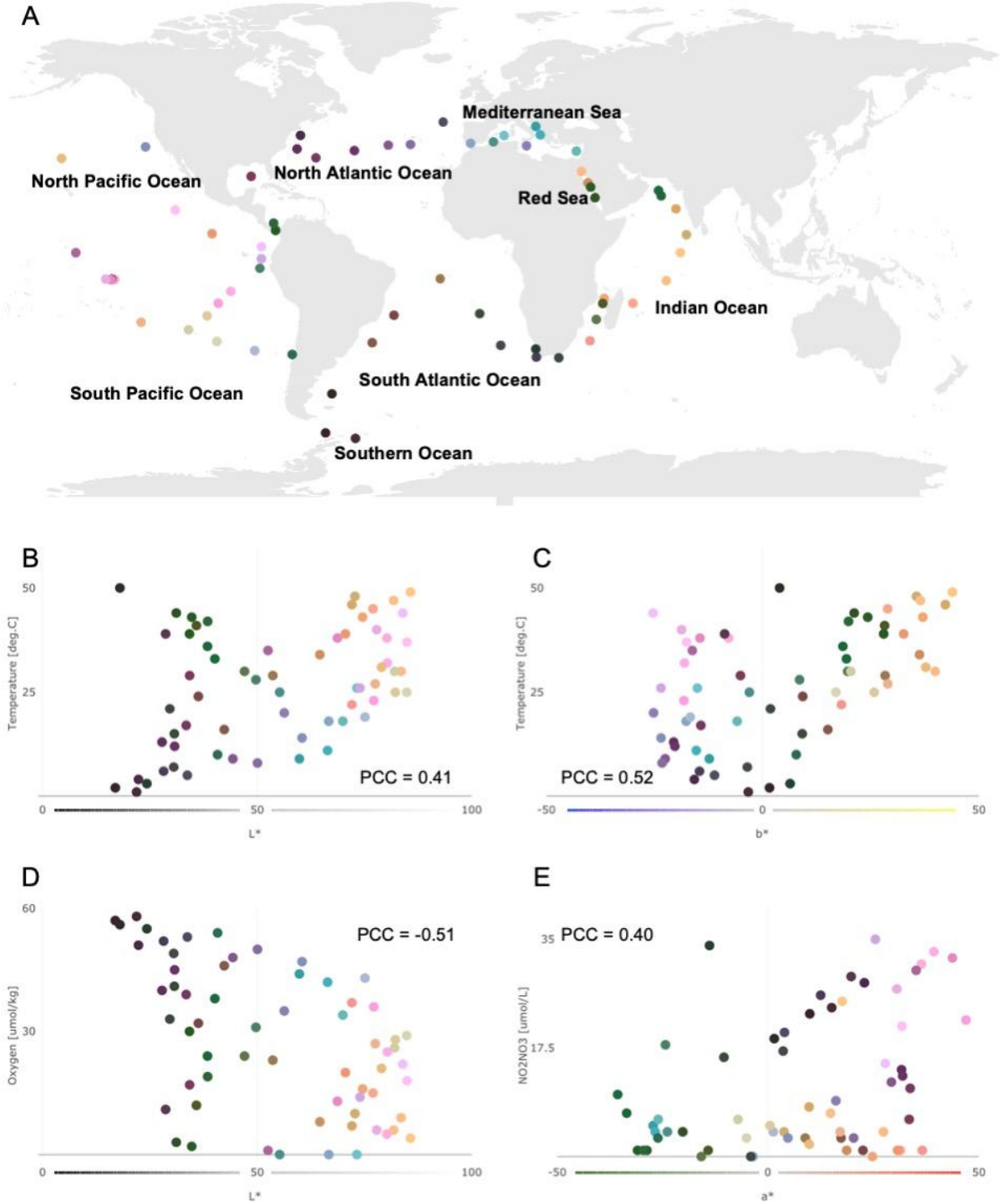

Figure 2. **Color encoding the microbiome compositions of ocean surface water samples.** 16S rRNA microbiome composition of surface water samples from the Tara Oceans [8] sampling stations were encoded as colors using U-CIE. The resulting colors were mapped onto the geographic locations of the sampling stations (panel A). The lightness (L*) and specially blue–yellow (b*) axes correlate with water temperature (panels B and C), which agrees with the observation in the original publication that temperature correlates strongly with microbiome composition. However, lightness has an even stronger (negative) correlation with dissolved oxygen concentration (panel D). Together, these axes thus correctly capture, from 16S abundance profiles, that the Indian ocean is warm with low oxygen, the North Atlantic Ocean is cold with high oxygen, and the Pacific Ocean has low oxygen and highly varying temperatures. Meanwhile, the green–red axis (a*) correlates with dissolved nitrite/nitrate concentration (panel E); for example, the middle of the Pacific Ocean is a high-nitrogen environment, whereas the waters around Panama have low nitrogen.

U-CIE is also available as an R package through CRAN (`install.packages("ucie")`) and devtools (`devtools install_github("mikelkou/ucie")`). Since users of this package will be working in the R environment, we give them full flexibility to use any dimensionality reduction method for creating the three-dimensional representation of the input data. The R package thus takes this as input and performs the optimization to fit the data into CIELAB coordinates. The output is a data frame with the names of the input data and the hex codes of the colors or the CIELAB coordinates. This data frame can be directly used with R or saved as a file locally for use in other visualization tools. Both the R package and the web resource are available under the open source MIT License.

Methods

Algorithm overview

U-CIE color encodes high-dimensional data using CIELAB color space in two stages: i) dimensionality reduction (can be skipped if data are already three-dimensional) and ii) fitting the resulting point cloud inside the displayable part of CIELAB (Fig. 1). The latter is a precomputed polygon, constructed by converting the RGB cube to CIELAB coordinates.

Dimensionality reduction

Unless the input data is already three-dimensional, U-CIE will start by reducing the data to three dimensions. There are three different tracks for doing so: ‘Single cells’, ‘High dimensional data’, and ‘Distance matrix’. The ‘Single cells’ track follows the Seurat [9] pipeline to reduce dimensionality and produce 3D UMAP coordinates for the genes of the dataset. To do so, we first transpose the input dataset to have genes as columns and cells as rows and create a Seurat Object from the counts. Without first scaling or centering the data, we use Seurat to log2 transform our data using the (option "LogNormalize"), run PCA with 50 dimensions, calculate 50 neighbors for each gene, and find clusters with the Louvain algorithm. Finally, we apply UMAP to end up with three dimensions. The ‘High dimensional’ track uses the uwot library to again run PCA with 50 dimensions and apply UMAP. Finally, the ‘Distance matrix’ track takes a square matrix and uses Python umap-learn package.

Fitting the point cloud inside the CIELAB RGB polygon

To be able to handle large datasets with many points, we first use the R ‘chull’ function to construct the convex hull of the point cloud. This dramatically speeds up the subsequent optimization step, typically by several orders of magnitude, as it only has to consider the few points on the convex hull rather than all points in the input dataset. Next, we use the Nelder–Mead simplex optimization algorithm [10] to fit the convex hull of the point cloud inside the CIELAB RGB polygon. The algorithm is allowed to shift along and rotate around three axes (L*, a*, and b*) and uniformly scale the point cloud convex hull. The objective function to be optimized is the size of the point cloud minus a penalty term for points falling outside the polygon. In other words, it aims to make the point cloud as large as possible while still fitting within the color space. To avoid local optima, we run the algorithm with 25 different sets of initial rotations. The user can optionally provide different weights for the axes in the objective, thus prioritizing spreading out the points along certain axes.

User Interface

The web interface was constructed using R/Shiny [11] and JavaScript. An interactive guided tutorial is available through the web interface as well as a YouTube video explaining the idea of U-CIE.

After uploading and processing data via one of the tracks described above, the data can be visualized in two ways. The '3D view' shows the main result of following the U-CIE pipeline, namely the 3D cloud of points colored according to the best solution found by the optimization algorithm. The view also has a table of alternative color solutions from the optimizations, which can be selected and displayed instead. The '2D projections' tab provides two interactive 2D plots, showing the same 3D point cloud projected onto the L*a* and L*b* axes, respectively. There users can see how the points are spread within the polygon. The view also has a table, which allows the user to select regions within the plots to identify the data points or search the data points by name to locate them within the cloud. Finally, the 'Download' tab shows a table with both RGB hex codes and CIELAB coordinates, which can be downloaded to use the colors for further data visualization, e.g. in Cytoscape [8]. The plots and tables are made using the Plotly [12] and DataTables [13] libraries, respectively.

CRAN package 'ucie'

Our algorithm is also available as a CRAN package in R. Users can download it from R with the command install.packages("ucie") or from the packages and the command devtools::install_github("mikelkou/ucie"). The package returns a data frame with the names of the input data points and RGB hex colors or CIELAB coordinates. The package contains only one function with 6 arguments, 5 of which are optional. Users must provide a 3-column dataset that will be encoded as colors. The other optional arguments allow users to alter the axis weights, a final scaling factor, and whether they want RGB hex codes or CIELAB coordinates. The output is a data frame with the names of the input data points and the corresponding colors.

Data availability

Each track offers a different small example dataset. The 'Single cells' track uses a single-cell RNA-seq dataset of human embryonic stem cells (GEO accession code GSE75748) [14]. The 'High dimensional' track uses an Affymetrix time course dataset of *Drosophila melanogaster* embryogenesis [15]. The 'Distance matrix' example uses the bacterial species tree available from the iTOL web resource (https://itol.embl.de) [16]. For the '3D Data' track we used the same dataset as in the 'High dimensional' track, reduced to three dimensions with locally linear embedding. Finally, the dataset used to create Figure 2 comes from [8] and can be downloaded from http://ocean-microbiome.embl.de.

Code availability

U-CIE source code is available at https://github.com/mikelkou/U-CIE_Web_Resource. R package 'ucie' code is available at https://github.com/mikelkou/ucie.

References

[1] J. van der Maaten, L., M. G. Hinton, G., and Geoffrey, “Visualizing Data using t-SNE,” *J. Mach. Learn. Res.*, no. 9, pp. 2579--2605, 2008.

[2] L. McInnes, J. Healy, and J. Melville, “UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction,” *ArXiv180203426 Cs Stat*, Sep. 2020, Accessed: Apr. 08, 2021. [Online]. Available: http://arxiv.org/abs/1802.03426

[3] G. E. Hinton and R. R. Salakhutdinov, “Reducing the Dimensionality of Data with Neural Networks,” *Science*, vol. 313, no. 5786, pp. 504–507, Jul. 2006, doi: 10.1126/science.1127647.

[4] D. Szklarczyk *et al.*, “The STRING database in 2021: customizable protein–protein networks, and functional characterization of user-uploaded gene/measurement sets,” *Nucleic Acids Res.*, vol. 49, no. D1, Art. no. D1, Jan. 2021, doi: 10.1093/nar/gkaa1074.

[5] P. Shannon, “Cytoscape: A Software Environment for Integrated Models of Biomolecular Interaction Networks,” *Genome Res.*, vol. 13, no. 11, pp. 2498–2504, Nov. 2003, doi: 10.1101/gr.1239303.

[6] Internationale Beleuchtungskommission, Ed., *Colorimetry*, 3rd ed. Wien: Comm. Internat. de l’éclairage, 2004.

[7] F. Y. Wang and M. Takatsuka, “Self-organizing Map (SOM) Based Data Navigation for Identifying Shape Similarities of Graphic Logos,” *Neural Process. Lett.*, vol. 41, no. 3, pp. 325–339, Jun. 2015, doi: 10.1007/s11063-014-9375-4.

[8] S. Sunagawa *et al.*, “Structure and function of the global ocean microbiome,” *Science*, vol. 348, no. 6237, p. 1261359, May 2015, doi: 10.1126/science.1261359.

[9] Y. Hao *et al.*, “Integrated analysis of multimodal single-cell data,” Genomics, preprint, Oct. 2020. doi: 10.1101/2020.10.12.335331.

[10] John A. Nelder and Roger Mead, “A simplex method for function minimization,” *Computer Journal*, vol. 7, pp. 308--313, 1965.

[11] Winston Chang and Joe Cheng and JJ Allaire and Carson Sievert and Barret Schloerke and Yihui Xie and Jeff Allen and Jonathan McPherson and Alan Dipert and Barbara Borges, “shiny: Web Application Framework for R,” 2021, [Online]. Available: https://CRAN.R-project.org/package=shiny

[12] C. Sievert, *Interactive web-based data visualization with R, plotly, and shiny*. Boca Raton, FL: CRC Press, Taylor and Francis Group, 2020.

[13] Matt Dowle and Arun Srinivasan, “data.table: Extension of `data.frame`,” 2021, [Online]. Available: https://CRAN.R-project.org/package=data.table

[14] L.-F. Chu *et al.*, “Single-cell RNA-seq reveals novel regulators of human embryonic stem cell differentiation to definitive endoderm,” *Genome Biol.*, vol. 17, no. 1, Art. no. 1, Dec. 2016, doi: 10.1186/s13059-016-1033-x.

[15] P. Tomancak *et al.*, “Systematic determination of patterns of gene expression during Drosophila embryogenesis,” *Genome Biol.*, vol. 3, no. 12, p. research0088.1, 2002, doi: 10.1186/gb-2002-3-12-research0088.

[16] I. Letunic and P. Bork, “Interactive Tree Of Life (iTOL): an online tool for phylogenetic tree display and annotation,” *Bioinformatics*, vol. 23, no. 1, pp. 127–128, Jan. 2007, doi: 10.1093/bioinformatics/btl529.