

Data visualization in the neurosciences: overcoming the curse of dimensionality

Elena A. Allen^{1,2}, Erik B. Erhardt^{1,3}, and Vince D. Calhoun^{1,4}

¹*The Mind Research Network, Albuquerque, New Mexico, USA*

²*Departments of Biological and Medical Psychology and the K.G. Jebsen Center for Research on Neuropsychiatric Disorders, University of Bergen, Jonas Lies vei 91, 5009 Bergen, Norway*

³*Department of Statistics, University of New Mexico, Albuquerque, New Mexico, USA*

⁴*Departments of Electrical and Computer Engineering, Psychiatry, and Neuroscience, University of New Mexico, Albuquerque NM, USA*

1 Survey details

Characteristics of surveyed figures are provided in Table S1. Survey questions were: 1) is the dependent variable or quantity of interest labeled? 2) Is the scale of the dependent variable indicated? 3) Where applicable, is a measure of uncertainty displayed? (We consider uncertainty or variability to be appropriate for descriptions of data distributions or parameter estimates, such as a population mean.) 4) Is the type of uncertainty (e.g., standard error bars or confidence intervals) defined in the figure or accompanying legend? While a definition of uncertainty may be provided elsewhere in the text, we believe it should be readily available to readers without having to scour the manuscript from beginning to end.

Examples of these graphical features are shown in Figure 1A for synthetic 2-dimensional (2D) and 3D datasets. For the 2D data (left), features 1 and 2 describing the dependent variable and its scale are found along the y-axis. Feature 3 is represented by error bars surrounding the parameter estimates, and feature 4 is found as the error bar label on the graph. For the 3D data (right), the outcome of interest is plotted as a function of two independent variables using a color map, which maps the values of the dependent variable to a continuous progression of color hue, or in this case, brightness. We define a color map as a display where the mapping between data values and color is *intentionally* performed by a researcher (in contrast to many photographic images and reconstructions where mapping is implicit). In the articles we sampled, color maps were by far the most common design used to display higher dimensional datasets. In the example, surveyed features 1 and 2 are indicated on the color-axis, or color bar. In this case, feature 3 is not shown, though generally uncertainty may be portrayed in a second color map (e.g., a map of standard errors), or integrated into a single map by plotting a test-statistic or using a dual-coding system as discussed in the main text. Because of the extra intention and notation required to display uncertainty surfaces, feature 4 does not apply to 3D axes. (Essentially, if it is clear that uncertainty is displayed, then it must have been labeled somewhere.)

For each article, binary ratings (yes or no) were assigned to all figures that displayed data in 2D or 3D graphs; schematics and other expository graphics were not evaluated. For multi-paneled figures, a single rating was assigned for each type of display (2D or 3D) based on the features present in the majority of panels. For results reported in Figures 1 and S1, proportions were determined first within each article then averaged over articles to remove the influence of the number of figures per article. Separation of survey

results by journal in Figure S1A shows that the prevalence of graphical features varies considerably, and that 2D and 3D graphical features covary slightly (Fig. S1B).

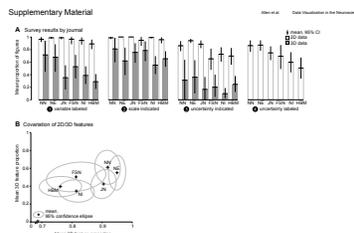


Figure S1: Survey results separated by journal. (A) Mean proportion of 2D (white) and 3D (dark gray) figures displaying each feature. Error bars denote 95% non-parametric confidence intervals (10,000 resamples). NN = *Nature Neuroscience*; NE = *Neuron*; JN = *Journal of Neuroscience*; FSN = *Frontiers in Systems Neuroscience*; NI = *Neuroimage*; HBM = *Human Brain Mapping*. (B) Scatter plot of the mean 2D feature proportion (averaged over features 1--4) versus the mean 3D feature proportion (averaged over features 1--3) for each journal. Crosses indicate mean data values; gray ellipses denote 95% confidence regions. See Table S1 for more information on the articles and figures surveyed.

Table S1: Frequencies and proportions of figures surveyed

	Articles	Figures	Figures with 2D data	Figures with 3D data	Figures with 2D and 3D data	Figures with 2D categorical data	Figures with 2D continuous data	Figures with 2D categorical & continuous data
NN	51	283	272 (0.96 ± 0.12)	42 (0.13 ± 0.27)	31 (0.10 ± 0.22)	184 (0.61 ± 0.36)	103 (0.35 ± 0.33)	45 (0.14 ± 0.19)
NE	50	344	332 (0.96 ± 0.13)	44 (0.14 ± 0.27)	32 (0.10 ± 0.23)	236 (0.69 ± 0.33)	141 (0.42 ± 0.32)	75 (0.22 ± 0.25)
JN	50	278	252 (0.89 ± 0.20)	61 (0.25 ± 0.34)	35 (0.14 ± 0.27)	144 (0.51 ± 0.38)	86 (0.33 ± 0.37)	29 (0.11 ± 0.23)
FSN	38	156	100 (0.55 ± 0.43)	64 (0.43 ± 0.42)	11 (0.06 ± 0.13)	40 (0.23 ± 0.34)	44 (0.24 ± 0.30)	7 (0.03 ± 0.11)
NI	49	190	141 (0.71 ± 0.33)	88 (0.50 ± 0.36)	39 (0.20 ± 0.31)	45 (0.28 ± 0.32)	59 (0.26 ± 0.30)	7 (0.04 ± 0.14)
HBM	50	200	121 (0.59 ± 0.35)	122 (0.66 ± 0.31)	43 (0.25 ± 0.28)	57 (0.31 ± 0.33)	31 (0.11 ± 0.25)	3 (0.01 ± 0.06)
all	288	1451	1218 (0.79 ± 0.32)	421 (0.35 ± 0.38)	191 (0.14 ± 0.26)	706 (0.45 ± 0.38)	464 (0.29 ± 0.33)	166 (0.09 ± 0.19)

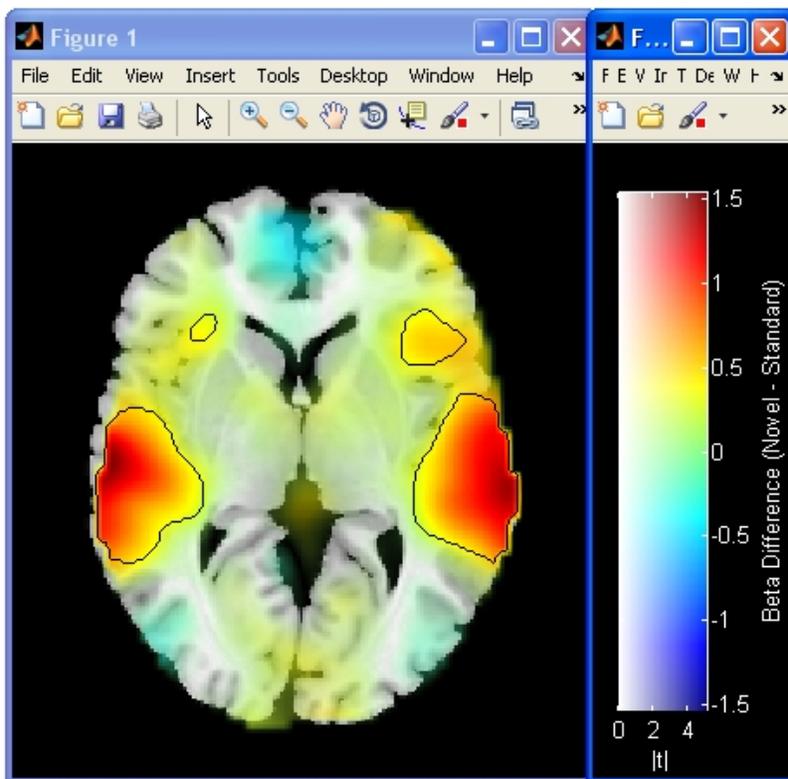
Data presented as number of occurrences (mean proportion over articles ± s.d.). NN = *Nature Neuroscience*; NE = *Neuron*; JN = *Journal of Neuroscience*; FSN = *Frontiers in Systems Neuroscience*; NI = *Neuroimage*; HBM = *Human Brain Mapping*.

2 Creating dual coded images

Here we provide sample MATLAB scripts to encourage the generation of rich displays that portray uncertainty. We use example data from an fMRI auditory oddball (AOD) experiment to demonstrate dual color-mapping. Quantitative effects are mapped to color hue and effect uncertainty is mapped to color transparency. To run the code and see an example of dual coding, simply un-zip `sample_code.zip` in its own folder and run `dualcodeImage.m` from the command line:

```
>> cd sample_code  
>> dualcodeImage
```

You should see something this:



2.1 dualcodeImage.m

The steps to produce this image are detailed in dualcodeImage.

```

1  %-----
2  % Sample Matlab code for creating images with hue and alpha color-mapping.
3  %
4  % Notes for using this code:
5  % You must have OpenGL available on your system to use transparency (alpha).
6  % When rendering transparency MATLAB automatically uses OpenGL if it is
7  % available. If it is not available, transparency will not display.
8  % See the figure property RendererMode for more information.
9  %
10 % EA Allen August 30, 2011
11 % eallen@mrn.org
12 %-----
13
14 %% 1. Load the AOD_data.mat file with sample data from the fMRI AOD experiment
15 %-----
16 load AOD_data.mat
17 % For a single axial slice (Z = 2 mm) of data, you should have:
18 % Bmap_N_S: 'Difference between Novel and Standard betas averaged over 28 subjects'
19 % Tmap_N_S: 'T-statistics for the paired t-test comparing Novel and Standard betas'
20 % Pmap_N_S: 'Binary map indicating significance at P<0.001 (fdr corrected)'
21 % Underlay: 'Structural image ch2bet from MRICron, warped to functional data'
22 %-----
23
24 %% 2. Set some defaults that will affect the appearance of the image
25 %-----
26 % Set the Min/Max values for hue coding
27 absmax = max(abs(Bmap_N_S(:)));
28 H_range = [-absmax absmax]; % The colormap is symmetric around zero
29
30 % Set the Min/Max T-values for alpha coding
31 A_range = [0 5];
32 % Voxels with t-stats of 0 will be completely transparent;
33 % voxels with t-stat magnitudes greater or equal than 5 will be opaque.
34
35 % Set the labels for the colorbar
36 hue_label = 'Beta Difference (Novel - Standard)';
37 alpha_label = '|t|';
38
39 % Choose a colormap for the underlay
40 CM_under = gray(256);
41
42 % Choose a colormap for the overlay
43 CM_over = jet(256);
44 %-----
45
46 %% 3. Do the actual plotting
47 %-----
48 % Make a figure and set of axes
49 F = figure('Color', 'k', 'Units', 'Normalized', 'Position', [0.3, 0.4, 0.2, 0.35]);
50 axes('Position', [0 0 1 1]);

```

```

51
52 % Transform the underlay and beta map to RGB values, based on specified colormaps
53 % See function convert_to_RGB() for more information
54 U_RGB = convert_to_RGB(Underlay, CM_under);
55 O_RGB = convert_to_RGB(Bmap_N_S, CM_over, H_range);
56
57 % Plot the underlay
58 layer1 = image(U_RGB); axis image
59 hold on;
60 % Now, add the Beta difference map as an overlay
61 layer2 = image(O_RGB); axis image
62
63 % Use the T-statistics to create an alpha map (which must be in [0,1])
64 alphamap = abs(Tmap_N_S);
65 alphamap(alphamap > A_range(2)) = A_range(2);
66 alphamap(alphamap < A_range(1)) = 0;
67 alphamap = alphamap/A_range(2);
68
69 % Adjust the alpha values of the overlay
70 set(layer2, 'alphaData', alphamap);
71
72 % Add some (black) contours to annotate nominal significance
73 hold on;
74 [C, CH] = contour(Pmap_N_S, 1, 'k');
75 %-----
76
77 %% 4. Create a 2D colorbar for the dual-coded overlay
78 %-----
79 G = figure('color', 'k', 'Units', 'Normalized', 'Position', [0.5, 0.4, 0.06, 0.35]);
80 x = linspace(A_range(1), A_range(2), 256);
81 % x represents the range in alpha (abs(t-stats))
82 y = linspace(H_range(1), H_range(2), size(CM_over,1));
83 % y represents the range in hue (beta weight difference)
84 [X,Y] = meshgrid(x,y); % Transform into a 2D matrix
85 imagesc(x,y,Y); axis xy; % Plot the colorbar
86 set(gca, 'Xcolor', 'w', 'Ycolor', 'w')
87 colormap(CM_over);
88 alpha(X);
89 alpha('scaled');
90 xlabel(alpha_label)
91 set(gca, 'YAxisLocation', 'right')
92 ylabel(hue_label)
93 %-----

```

2.2 convert_to_RGB.m

The script above is dependent on `convert_to_RGB`, which is used to transform pixel values with arbitrary scaling into a truecolor RGB image.

```

1 function IMrgb = convert_to_RGB(IM, cm, cmLIM)
2 % convert_to_RGB - converts any image to truecolor RGB using a specified colormap
3 %
4 % USAGE: IMrgb = convert_to_RGB(IM, cm, cmLIM)
5 %
6 % INPUTS:
7 %   IM       = the image [m x n]
8 %   cm       = the colormap [p x 3] <optional; default = jet(256)>
9 %   cmLIM    = the data limits [min max] to be used in the color-mapping
10 %             <optional; default = [min(IM) max(IM)]>
11 % OUTPUTS:
12 %   IMrgb    = the truecolor RGB image [m x n x 3]
13 %
14 % Based on ind2rgb from the Image Processing Toolbox
15 % EA Allen August 30, 2011
16 % eallen@mrn.org
17 %-----
18 if nargin < 2
19     cm = jet(256);
20 end
21
22 if nargin < 3
23     cmLIM = [min(IM(:)) max(IM(:))];
24 end
25
26 IM = IM-cmLIM(1);
27 IM = IM/(cmLIM(2)-cmLIM(1));
28 nIND = size(cm,1);
29 IM = round(IM*(nIND-1));
30
31 IM = double(IM)+1;
32 r = zeros(size(IM)); r(:) = cm(IM,1);
33 g = zeros(size(IM)); g(:) = cm(IM,2);
34 b = zeros(size(IM)); b(:) = cm(IM,3);
35
36 IMrgb = zeros([size(IM),3]);
37 % Fill in the r, g, and b channels
38 IMrgb(:, :, 1) = r;
39 IMrgb(:, :, 2) = g;
40 IMrgb(:, :, 3) = b;

```