

NUT: Visualizing the Nutritional Content of What We Eat

Jessica Dawson*

University of British Columbia

ABSTRACT

I present NUT, a tool for understanding the nutritional content of what we eat. The project consists of a rapid prototyping and analysis phase, as well as a design study documenting the development of the NUT prototype. The goal of NUT is to support users in making healthy choices about what to eat by visualizing nutritional information in the context of the whole food. This report outlines the domain, task and data used in NUT, as well as an overview of the visualization solution and the components of the application. The strengths and limitations of the current implementation are discussed, along with immediate next steps and potential areas for future work. A work-in-progress version of NUT can be found at www.cs.ubc.ca/~jqdawson/533c/nut.html.

KEYWORDS: Vis for the masses, nutrition visualization, design study

1 INTRODUCTION

The goal of NUT is to help users answer the question “What’s in my food?” and to assist them in making healthy choices about what to eat. To achieve this goal NUT supports the exploration of the nutrient content of common foods in the context of the whole food through the visualization of a high-dimensional nutritional database.

A source of inspiration for this project comes from Michael Pollan’s *In Defense of Food: An Eater’s Manifesto* [19], which investigates how the ‘nutritionalism’ ideology (the term that Pollan has coined for the health fads such as ‘eat less fat’ or ‘eat more protein’), have led to poorer nutrition and health. Pollan argues against this type of reductionist science, and instead advocates for increased attempts to understand nutrients in combination and in the context of the whole food. Nutrition and the nutritional content of foods are popular topics for a wide range of people who make living a healthy lifestyle a priority. As Pollan’s documentation of the evolution of nutrition science shows, people are desperate for answers and guidance about what they should eat. With the evidence against the Western diet ever increasing in the realms of popular health and culture, more and more people are looking for new ways to inform their food choices.

A number of nutritional information systems and visualizations intended for casual use already exist. However, previous information visualization applications in this domain have typically focused on understanding foods by looking at one or two nutrients at a time. These types of applications only support users in exploring and making food choices on a nutrient-by-nutrient basis, for example, by looking for foods that have the most protein. In contrast, NUT provides the user with a high-level

understanding of the nutrient composition of the food as a whole, with the option to drill down to more detailed information if needed.

NUT is a web application designed to support tasks to help users in understanding the nutritional content of what they eat. It is intended to be approachable and easy for anyone to use quickly, while still providing details for food nerds or enthusiasts interested in spending more time exploring nutritional data. Supported tasks include helping users understand which foods contain which nutrients, as well as understanding the general overall nutrient profile of different foods. NUT also supports comparisons between two foods at a high level to guide users in choosing both food replacements and food compliments.

1.1 Data

This project visualizes the abbreviated version of the USDA National Nutrient Database for Standard Reference (USDA NNDSR), which is the main source of food composition data for the United States. The most recent version, Release 24 [12], describes a collection of 7906 different foods with 51 dimensions, which includes the mean nutrient values for 46 different nutrients. Values for foods are all calculated for 100g of food. The foods in this dataset are classified into different ‘food groups’ through a simple numbering system; 25 of these categories are explicitly defined.

2 RELATED WORK

2.1 High dimensional pixel-based visualizations

Inspiration for the overview design originally came from the Circle Segment visualization [3]. I was drawn to the idea that all the data could be shown in multiple dimensions at the same time because it seemed capable of supporting my goal of visualizing a high-level understanding of foods to users rather than focusing on details. The final implementation of NUT uses a rectangular view, rather than a radial view for the overview, and is very similar in its implementation to a pixel bar chart [4]. NUT is distinct from these pixel-based designs in that it uses a line instead of a pixel in order to improve visibility; a similar adjustment, although conceived of separately from the design of NUT, was made by Ziegel et al. for the volatility display in their visual market sector analysis for financial time series data [5]. The inspiration to use a supporting visualization to show the overview detail came from Mizbee [6].

Further discussion on the influence of the related works on the design of the NUT overview can be found in section 3.2.1.

2.2 Nutrition Visualizations

A few other visualizations have been developed using the USDA NNDSR, but none support high-level exploration of food composition or facilitate the comparison of whole foods, and instead tend to focus on the properties of individual nutrients. Lonely datum’s *Visualizing Nutrition* [7] allows the user to view a summary report for each nutrient in the data base, including visualization of the top 8 foods containing that nutrient and an

*jqdawson@cs.ubc.ca

ordered, size encoded view of all of the food groups and foods by nutrient type. A strength of this visualization, which NUT could potentially leverage in the future, is the integration of daily recommended intake values that gives users useful context for making food choices. An anonymous visualization on Many Eyes titled *Treemap of USDA National Nutrient Database* [19] visualizes the dataset in a tree map of food groups, where the user can specify a nutrient to be encoded by size, and a nutrient to be encoded by colour. This is effective in visualizing all the foods at once, but only allows users to compare two nutrients at a time.

3 ANALYSIS

In order to fulfill its role as visualization for the masses, NUT needed to be approachable to everyday users. This requirement presented a challenge in terms of deciding how best to present the large USDA NNDSRA dataset in a way that did not overwhelm users. While NUT is particularly geared towards everyday ‘food nerds’ like myself, I decided that the 47 original dimensions were likely too complex to be shown all at once in an effective overview. Further, it was likely that not all of this data would be interesting.

Thus, to inform the final design of NUT the first phase of this project involved an analysis of the USDA NNDSRA, during which I explored how and if the data could be effectively reduced. I entered this analysis with 3 primary questions: 1) Are some nutrients inherently more interesting than others, 2) can the data be reduced by filtering nutrient dimensions or deriving new dimensions, and 3) can the data be reduced by filtering rows (i.e. foods)?

3.1 Rapid prototyping for analysis

I originally attempted to use three different high-dimensional visualization and dimension reduction tools for my analysis: Ggobi [9], the Xmdv tool [10] and Dimstiller [11]. However, I quickly realized that the learning curve for each of these tools was prohibitively high for someone with my level of background knowledge. Each tool also required a different data format, and so much of the time I spent with these tools went into transforming my data into different formats.

While these tools provided very sophisticated analysis options, these challenges suggested to me that using them was outside the scope of my project. Further, the tools quickly began to seem like overkill for what, in the scope of a course project, could likely be performed in much simpler way.

I chose instead to perform a simpler analysis through rapid prototyping in Processing [12] and Processing.js [13], and my own naïve exploration of the data. I informed this analysis with my own previous knowledge and very limited further research into the domain of nutrients and nutritional guidelines.

3.2 Analysis results

I noticed that in many dimensions large groups of similar foods tended to occupy similar positions in the nutrient. In total I generated three complete processing prototypes that I was able to use to examine the data, each an evolution of what I learned from the previous one. I describe each of these prototypes and focus the majority of the discussion on the results and findings that informed the present design of NUT.

3.2.1 Early overview design

All of the overview prototypes took the form of a high-dimensional pixel-based chart overview similar to the final version I intended for NUT. The view used columns to represent the nutrient dimensions, with each food drawn as a bar instead of

a pixel, with 4 bars to a row. These bars were stacked from least to most amount of the nutrient, using a simple zig-zag layout of increasing values similar that used by Ankirst et al [3] for circle segments. I had originally planned to use a circle segments technique instead of bar charts, but I was convinced through Diehl’s discussion of the relative strengths of radial and rectangular visualizations [14] that a rectangular view would provide better support for my users in getting a sense of the whole food across multiple dimensions

Ziegler et al [5] use a similar bar chart style with lines instead of pixels in their Visual Market Sector Analysis; to avoid visual randomness they combine continuous values in order to accentuate trends. I was interested in accentuating the distribution of food groups, so I ordered my display such that the food group further ordered foods with the same value. Values with more than 2 significant digits of precision were rounded to further promote more clumping, and because that level of exactness was not required for the purposes of NUT. This helps to ensure that similarly coloured lines would clump together, making similarities and differences easier to pick out. The foods were colour-encoded by the USDA supplied category using a rotating scheme of 12 colours. A simple form of highlighting was also implemented: when a food was hovered over with the mouse, its position was highlighted in each column.

3.2.2 Evolution of prototypes and results

My first useful prototype filtered out all of the incomplete data, retaining only rows with a value for every dimension. This dramatically reduced my data to 2870 foods, but eliminated some categories of foods almost entirely. Even with the missing values eliminated, for about half of the nutrient dimensions more than 50% of the foods for any given dimension had value of zero, which indicated trace amounts of a nutrient too small to be counted. These can be clearly identified in Figure 1 as the large bands of colours rising up from the bottom of the columns.

This first prototype confirmed three main points. First, interaction on a mouse with such an overview was too cumbersome to realistically be used in a casual visualization. Second, completely eliminating foods with missing values took away too much of the dataset. More than two thirds of the foods had incomplete data and a useful nutritional analysis tool would need to communicate this fact clearly to users in some way or risk misleading users about typical nutritional content.

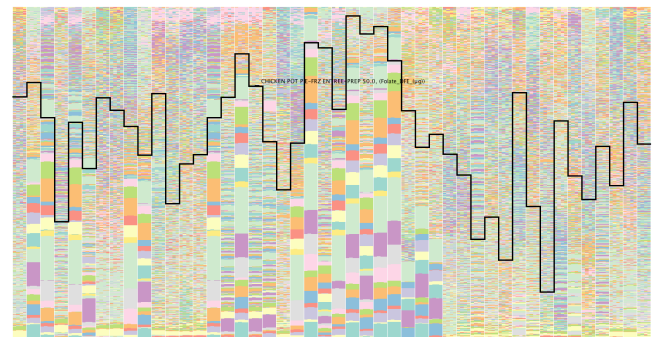


Figure 1. The first overview prototype; 46 dimensions, 2068 foods per dimension; black lines shown a mouse-over selection;

The second prototype took a similar form to the first, but incorporated the majority of the foods. A few of the USDA categories were filtered ahead of time based on simple decisions about the domain: restaurants and fast food were filtered because many of the chains were non-Canadian, and the meals tended to

cut across multiple other categories; baby food was filtered because it was interesting only to a small subgroup of my population; and a number of other categories were filtered simply because the USDA did not appear to provide a definition and one was not obvious from the foods included. The randomness shown in the first prototype encouraged me to group the USDA categories into higher-level groups of similar foods (i.e. all meats together), which I then re-encoded with new colours.

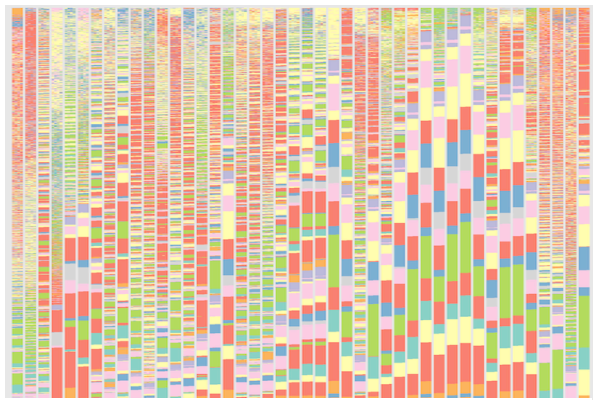


Figure 3. the second overview prototype; foods with zero or null values clearly visible by the unusually large colored bars

The high number of foods with either null or 0 values was again immediately obvious in the prototype as shown in figure 2, as these dominated even more of the view than in the first prototype.

Using this prototype I was quickly able to identify a number of unique nutrients, such as protein, ash, and Vitamins C, E, and K, which had little in common with any other nutrient dimensions. I was also able to see that a few dimensions measured the same nutrients, but simply in different ways, such as Vitamin A and Vitamin D, which were both included in micrograms and International Units (IU). With my limited highlighting, I could see a similar progression of many foods across these dimensions, which suggested they were highly correlated.

When I attempted to look for similarities across other nutrients, I found little similarity at the level of individual foods. However, I could see that in many dimensions large groups of similar foods tended to occupy similar ranges across the nutrient dimension. Based on these similarities I created loose groupings of dimensions I thought might be candidates to collapse based on commonalities in food group representation. Small sections of two such groupings are shown in figure 3. In total I created 8 such groupings from 25 of the nutrients (see Table 1 at the end for details).

Some limited research into the nutrients showed that my loose groupings were, with a few exceptions, very similar to common high-level groupings of chemically related nutrients, such as carotenoids and fats. For example, B-vitamins, fats, and carotenoids were separated in similar groups, as were a number of the trace dietary minerals.

This encouraged refinement of my groupings through further analysis. The third prototype I created went through a number of iterations as I explored which dimensions could be collapsed and how best to collapse them. Ultimately, from visual analysis alone I was not able to reduce my groupings to few enough dimensions to visualize in an overview. My final groupings shifted from my original groupings to cover broader categories such as trace minerals, and some visually dissimilar dimensions were ultimately collapsed. For most dimensions, the most unbiased and logical approach I tried was to first normalize each dimension by its

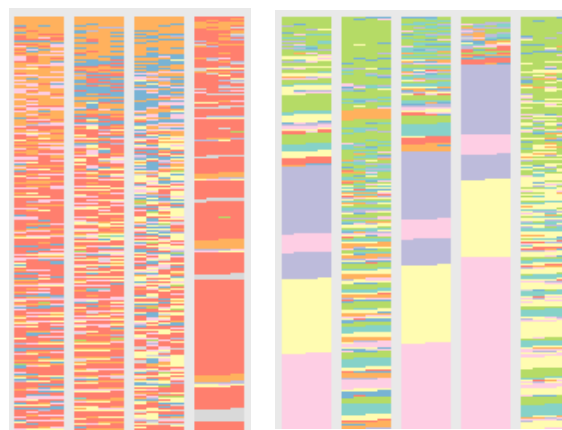


Figure 2. Portions of the second prototype. Left: three types of fat and cholesterol. Right: 5 types of carotenoids

maximum value, and then average the normalized value. This gave me a value between 0 and 1, which gave each of the collapsed nutrients an equal contribution to the new dimension. This was especially effective where collapsed nutrients used different scales.

The final version of my third prototype was used to generate the Overview image for NUT.

3.2.3 Limitations

I experienced a number of limitations in my analysis, largely due to my lack of experience and limited time available.

For example, it is possible that the reason I saw few patterns of similarity between dimensions for individual foods was because the limited highlighting in my prototype poorly supported this task. I could only highlight two foods at a time, making it difficult to look for larger trends. In retrospect, a parallel coordinates analysis in the Xmdv tool [10] could have been effective to further investigate for a small number of dimensions once I suspected they were similar.

A second limitation surrounds my calculation of the derived dimensions. I tried a number of naive approaches such as averaging and summing the values, but none gave as consistently decent results as normalizing. Likely with more time and knowledge I could come up with more sophisticated and effective derivations.

3.2.4 Conclusions and recommendations

Through my analysis I was able to come up with some reasonable, albeit not perfect, answers to my analysis questions.

Are some nutrients inherently more interesting than others? Yes. A number of unique and interesting nutrients were identified, including protein and all of the vitamins. The least interesting value is likely Ash, but given its uniqueness from the others I recommend keeping it, along with some measure of carbohydrates and fats, since together protein, carbohydrates and fat make up the big three of nutritional science.

Can the data be reduced by filtering nutrient dimensions or deriving new dimensions? Yes. A number of nutrients were shown to measure similar things, such as Carbohydrates and total sugar. Some derived dimensions were ultimately also similar to existing dimensions, such as the derived dimensions of three types of saturated fats dimensions and total lipids. From each group of such nutrients a decision can be made about the most interesting to retain, and the rest filtered. A number of new dimensions can also be somewhat effectively derived from existing dimensions.

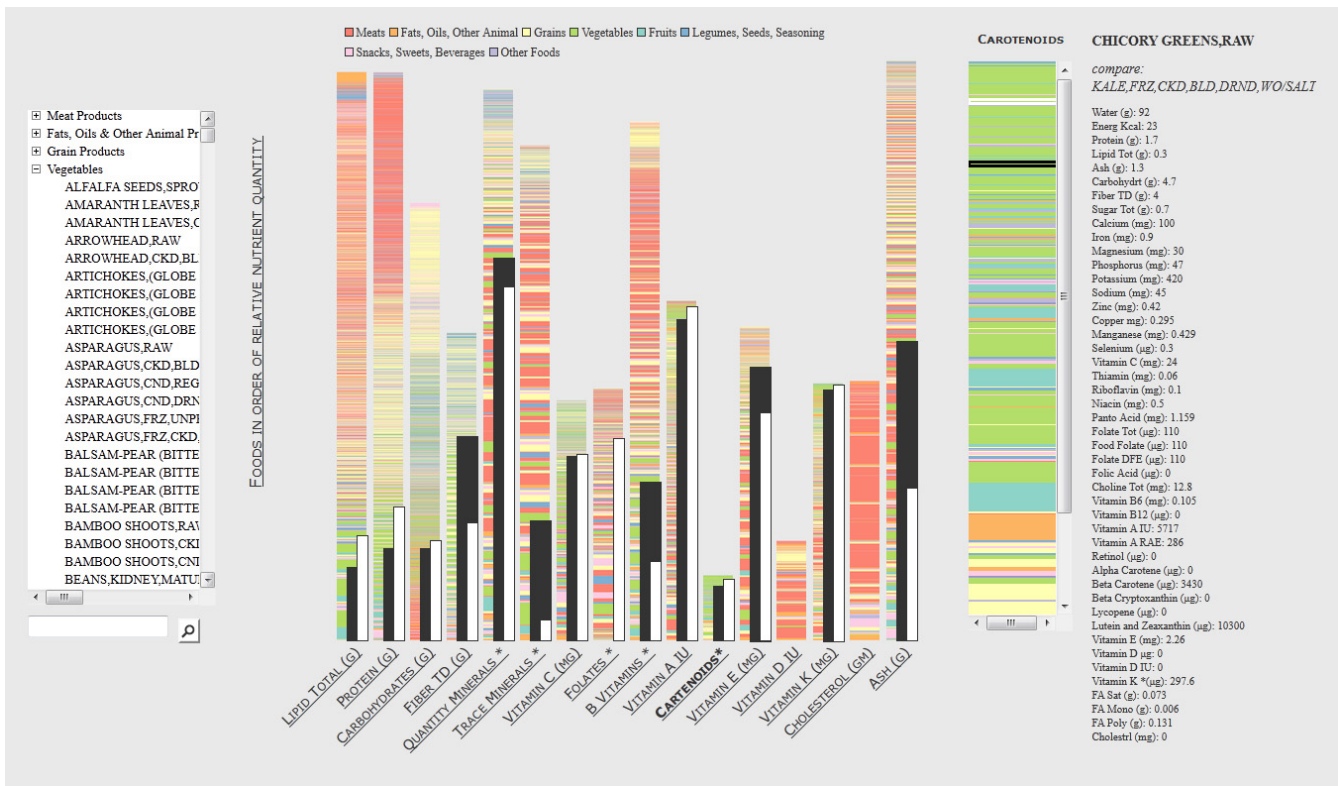


Figure 4. The NUT interface showing a primary and comparison selection. From left to right: Food List view, Overview, Nutrient view and Detail view.

Can the data be reduced by filtering rows (i.e. foods)? Yes. A large number of food values in each dimension represent null or zero values. These could be filtered out to simplify the overview.

Based on these recommendations I chose a set of derived dimensions to use in the NUT Overview, leaving me with a total of 16 dimensions (see tables 2 and 3 at the for details). I also chose to eliminate the same categories of foods that I eliminated in the second prototype, leaving me with 6014 foods.

4 DESCRIPTION OF THE SOLUTION

NUT is intended to support users in exploring the nutritional content of food through visualization. The application consists of multiple linked views centred on a high-dimensional overview of the nutrient composition of the foods in the USDA NNDSR. Data on 6014 foods is provided. NUT is intended to support users in gaining a general sense about foods as a whole through exploration of both nutrients and foods. With this knowledge, users may find themselves better equipped to make informed choices about what they eat.

The application provides both overview and detail on demand. The intention is to support users in gaining a broad understanding of foods first through the overview; the detailed summary is a secondary interest, intended for those users who want to dive deeper. A primary selection and a second food can be compared at a high-level in the overview, while details for the primary food can also be examined. The interface consists of 4 panels, with the overview providing the most visual information.

4.1 Food list view

The hierarchical tree list in the leftmost section of NUT provides the user with high-level groups of similar types of foods, such as meat products, vegetables, legumes and seeds, grain

products, fruits and fruit juices, etc. from which they can select. In the rest of the system foods are color-coded in terms of their membership to one of these eight groups. Expanding each group reveals the list of foods in that group, organized under further sub-categories as defined by the USDA NNDSRA. Grain products, for example, are further organized into breakfast cereals and baked products as well as cereal gains and pasta. This view supports click selection of a food from the list to update the other views with the primary selection. Holding shift while clicking selects a comparison food.

In the list foods are presented using the title provided in the dataset, which in some cases describes many aspects of the food and can be quite long. The foods are organized first in alphabetical order of the primary food name, i.e. Pork, and when more than one food begins with the same primary name they are further organized by similarity in further descriptive characteristics, e.g. fresh or frozen.

The search area below the list can be used to filter the items shown. When searching, only those items that contain the specified term will be shown in the list. Clearing the search box returns the list view to its unfiltered state.

4.2 Overview

The next panel displays an overview of the nutrient dimensions and the foods relevant to each nutrient. The design is based off of pixel-based displays, with horizontal lines used instead of individual pixels to encode the foods. The nutrient values for all foods are calculated as the value per 100g of food. The scales of measurement vary, using grams (g), milligrams (mg), micrograms (μ g), International Units (IU) or potentially a combination when normalized.

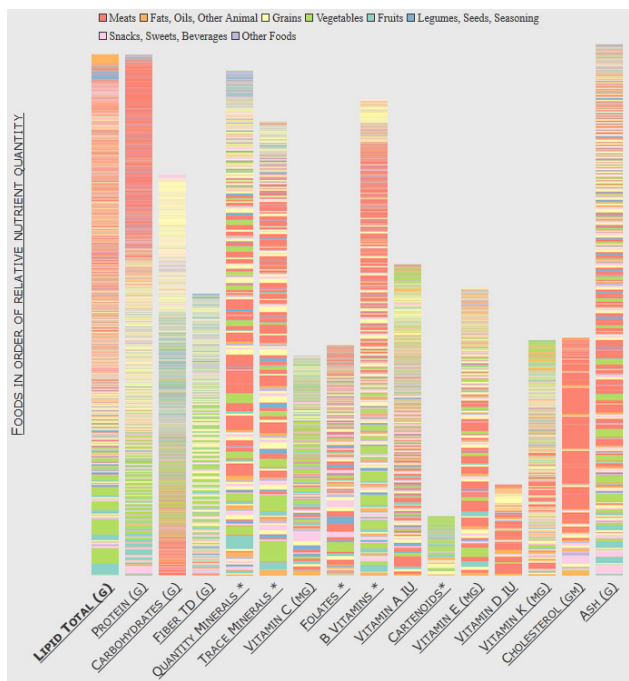


Figure 4. The Overview with no-highlighting.

4.2.1 Overview design

The overview consists of 16 nutrient dimensions, both original and derived from the full dataset of 46 nutrient dimensions. See section 3.2.2 for more information on how these dimensions were chosen. More information about related work pertaining to the design can be found in section 3.2.1

The overview takes the form of a pixel-based bar chart. A column represents each dimension, and foods are represented by color-coded lines stacked within the column in relative order from the food containing the least to the most amount of each nutrient. Only foods with a non-zero value of the nutrient of the food are displayed. The dimensions are labelled with their name and scale across the bottom of the overview, with the normalized scale for calculated dimensions indicated with a '*'.

The colour encoding corresponds to the eight high-level food group classifications that are also displayed in the Food List view. The encoding uses eight distinct colours chosen through ColorBrewer2.0 [15]. A legend defining the colours encoding each food group can be found above the overview in the interface. This encoding is used consistently throughout the interface to indicate food belonging to the same food group. Each individual food is represented by only 1/10th of a pixel. While the line encoding foods are too small to see individually, the colours tend to combine in bands and sections that give a general sense of the types of foods present in lesser or greater amounts of a nutrient (figure 5).

The ordering of the columns was determined through a combination of domain knowledge and the ordering of the original dataset. In general, the columns proceed from left to right in a loose definition of interest, with the most interesting 'big 3' (lipids, protein, carbohydrates) taking up position on the left. The remaining columns are loosely sorted by chemical similarity, such as fat-soluble or water-soluble, with ash, a unique but less interesting nutrient, taking up the right most position.

4.2.2 Interacting with the overview

Inspection of the height of each nutrient column allows the user to quickly judge the widespread availability of a nutrient: do a lot of foods have this nutrient or only a few? Using this information in combination with the general colouring at different positions within the column gives the user further information about the types of foods that tend to contain more or less of a nutrient. The overview currently only supports click interactions. Selecting a nutrient dimension by clicking on the column causes that nutrient to be displayed in the Nutrient view. The selected nutrient is indicated in the overview through the bolding of the label under the nutrient dimension column.

Selecting a food from the Food List view or the Nutrient view, highlights the relative position of that food within each nutrient dimensions in the overview through a black bar as in figure 4. The top of the bar shows the position of the food, and no bar is shown for a particular nutrient dimension when the food does not contain that nutrient. The use of a bar reinforces the relative quantity of nutrient that food has through size, making it easy to determine if the food contains a lot, a little, or none of the nutrient. The bar is positioned over only two-thirds of the column, such that the information below is still visible to the user.

Selecting a comparison highlights the relative position of that food within each nutrient dimension on top of the selected food in white, as shown in figure 4. The bar is positioned to the right, taking up only one-third of the column, in order to reinforce the height of the selected bar. This allows the user to quickly grasp the relative difference in nutrient quantity between the selected food and comparison food, as well as to judge the overall relative difference as compared to all possible foods.

4.3 Nutrient View

The Nutrient view provides a zoomed-in view of a single selected nutrient dimension from the overview. This is the second panel from the right as seen in figure 4. In this view each food is drawn as one pixel in height. This allows for more in depth exploration of the foods that make up a particular range of the nutrient and also supports looking for phenomenon that are more difficult to see in the overview, such as individual foods that do not match the dominant food group(s) in a range of values of the nutrient.

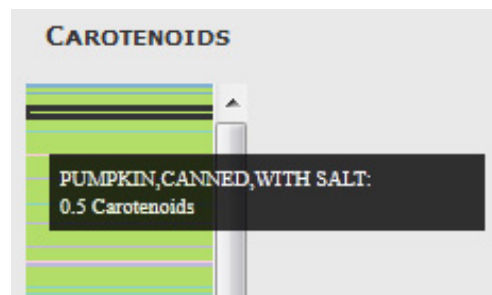


Figure 6. Close up of the Nutrient view showing the mouse-over interaction

This view supports both hover and click interactions. A dark grey box that moves with the cursor is drawn around food items as in figure 6, which allows the colour of the hovered over food to be seen through the middle of the box. A tooltip beside the cursor displays the name of the food and the nutrient value for the current nutrient.



Figure 7. Close up of the Nutrient view showing primary and comparison selections

Clicking on a food selects the food, and updates the other food with the selection. A black box is drawn around the selection, which, remains in place while the user hovers over other foods. A comparison food can be similarly selected by holding the Shift key while clicking, which updates the selection in the other view and draws a white box around the food in the Nutrient view.

4.4 Details View

The Details view is positioned at the far right of the screen as shown in figure 4 and displays the complete nutrient data for a selected food. It also serves as a reference to the user to the current state of the primary and comparison food selections.

The name of the selected food is shown at the top of the view, with the amounts of each nutrient per 100g of food listed out below. This view is updated with the information whenever a food is selected from the Food list or Nutrient views. The comparison food is shown by name directly underneath the title of the selected food, but to preserve space the details remain hidden. To access the details for the comparison food the user can either get the values for the selected nutrient by hovering over the food in the nutrient view, or by making it the primary selection.

5 MEDIUM-LEVEL IMPLEMENTATION

A prototype of NUT has been implemented as a Processing.js based web-application. Currently the application is only tested and supported for the newest versions of the Firefox and Safari web browsers. A work-in-progress version of NUT can be found at www.cs.ubc.ca/~jqdawson/533c/nut.html.

5.1 Implementation of Views

The Overview and Nutrient views were implemented as Processing sketches, which were embedded in a webpage through the use of Processing.js. The Overview was created using a scaled-down image generated ahead of time through a separately implemented Processing sketch. The Nutrient view and the highlighting in the Overview were drawn on the fly.

The remaining components were implemented through a combination of HTML, CSS, Javascript and the jQuery javascript library [16]. The Food List view was implemented using the jsTree jQuery tree plug-in [17]. All interface interactions, as well

as communication between the Processing sketches and the remaining interface elements, was achieved through JavaScript and the jQuery JavaScript library. All data was loaded into the application using Ajax support through jQuery.

5.2 Data Preparation

All data is loaded into nut using AJAX and stored as JavaScript Objects. The original USDA NNDSR dataset came in excel format. To support the different components in the application, the data had to be formatted into four different JavaScript Object Notation (JSON) formats [18]. The program architecture could possibly have been redesigned to rely on fewer data files, but I did not have enough personal expertise to accomplish this task in the time available.

The data preparation was accomplished through a combination of manual and automatic processes. Excel macros extracted data into CSV files and then the files were manually put through a CSV to JSON converter [19, 20]. The available conversion tools could only convert data from a CSV into a JSON format that was one level deep, but to achieve any reasonable program architecture required the data to be at least two levels deep. As a result of this and the limited time available to develop an automatic solution, I had to manually stitch together the outputs from the conversion tools to build the JSON files.

6 SCENARIOS OF USE

NUT supports a number of tasks to help users understand the nutritional content of their food, as illustrated by the following scenarios.

6.1 Which foods have which nutrients?

A user's first task when using NUT might be to gain a better understanding of which types of foods contain which types of nutrients (figure 8). Simply by examining the heights of the columns in the un-highlighted Overview the user can compare the nutrients found in a wide number of foods, such as lipids and protein to the nutrients found in a limited number of foods, such as carotenoids and vitamin D. By looking at the distribution of the

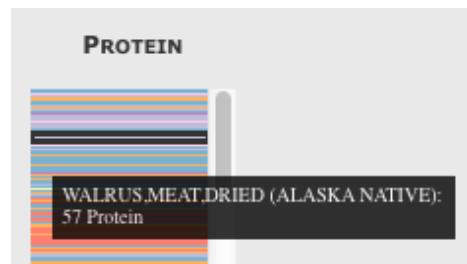


Figure 8. Illustrating a surprising find in scenario 6.1

food groups across the nutrients the user can further identify any types of food that tend to contain a lot or a little of a particular nutrient. A quick glance at the protein nutrient shows that the majority of high protein foods are meats, followed by grains, then vegetables, with things like sweets having the least protein.

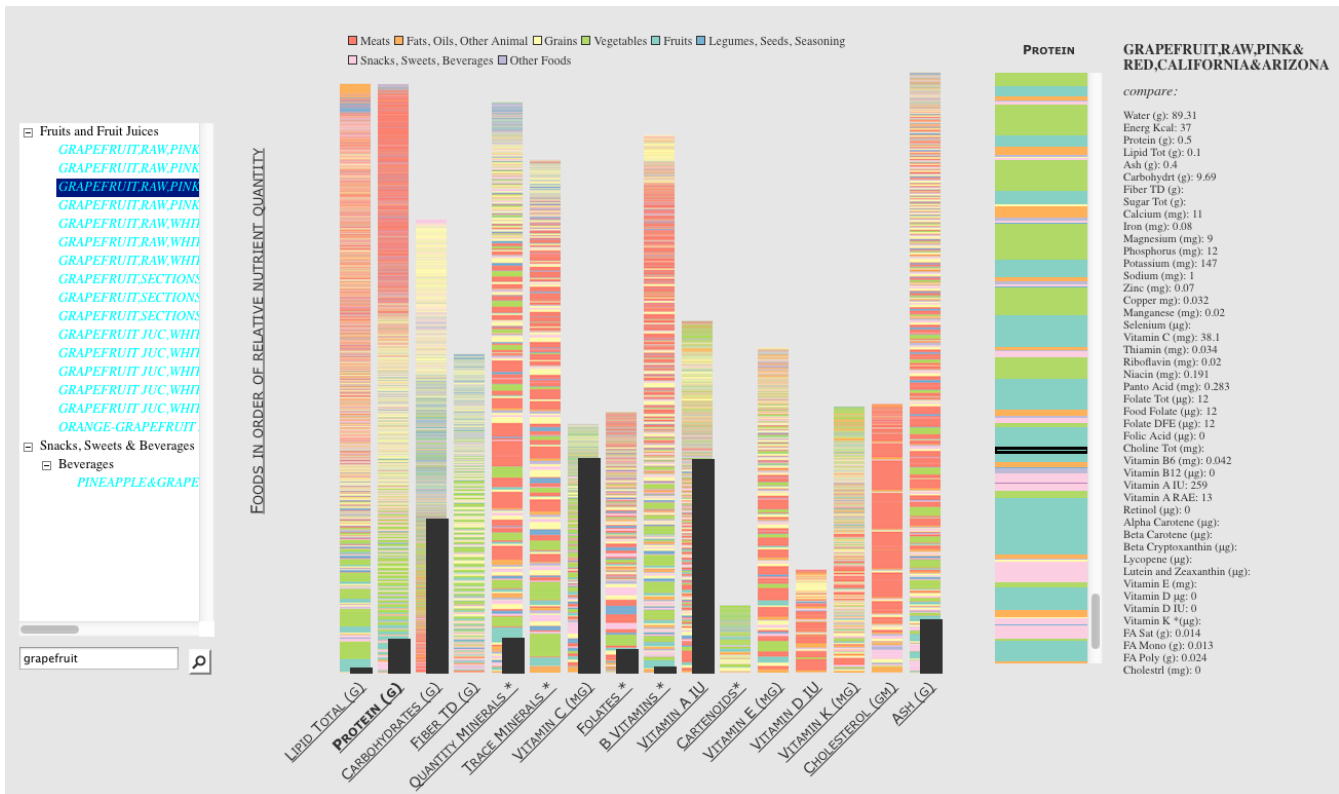


Figure 9. Scenario 6.2, understanding the nutrient profile of a food

By examining the nutrients in the Overview the user may notice some patterns they wish to investigate more deeply. For example, although the higher range of protein largely consists of meats, a mixed bar of other food types takes the place at the very top. Clicking on the column in the overview brings protein up in the Nutrient view. This reveals mostly blue and purple bars, which mouse over reveals to be primarily a combination of highly processed soy and legume products and very uncommon wild game meats like walrus and seal (which the USDA classifies as an 'Ethnic Food' rather than meat).

6.2 Understanding the nutrient profile of a food

Lets imagine now that a user's favourite fruit is grapefruit; She eats it everyday for breakfast and she wants to know what types of nutrients it contains (figure 9). To start she types grapefruit into the search bar, sees a number of possible results, and selects raw pink grapefruit from Arizona. In the overview she quickly sees that grapefruit contains large quantities of Vitamins A, C and carbohydrates, but provides relatively little of anything else. In particular, she is surprised to see how little it contains in terms of minerals. Inspection of the Details view reveals that, while grapefruit contains a fair amount of Potassium, it contains very little of any other trace or quantity minerals.

6.3 Comparing profiles of two different foods

Another common scenario for use in NUT is to compare the nutrient profiles of two foods. Imagine a lactose intolerant use that is interested in finding a suitable replacement for milk in their diet

(figure 10). The user searches for milk in the Food List and selects a 2% milk fortified with Vitamins A and D. She knows that milk is a good source of calcium and so she selects the quantity minerals dimension and hovers the mouse over foods around the highlighted milk selection. She sees there are a number of grains nearby, and so she selects one these, Kellogg's corn flakes, for comparison. To her surprise she an quickly see in the overview that the cornflakes have a similar amount of Vitamin D, and more of every type of nutrient except for fat. She continues exploring by hovering over and comparing other foods close to the milk selection, and then chooses a different nutrient dimension and does the same.

Alternately, a user may be interested in finding a combination of foods that cover a wide range of nutrients. The user starts by selecting food she eats often, such as long grain white rice. She notices that rice doesn't have a lot of vitamins, and so she selects a Vitamin A (which rice does not contain). She then decides to inspect a group of vegetables near the top of the Vitamin A column, and by selecting each in turn for comparison she finds a number of vegetables, including squash and snap peas, that could be eaten with her rice to ensure she gets a wide variety of nutrients.

7 DISCUSSION

7.1 Strengths

7.1.1 Understanding the whole food

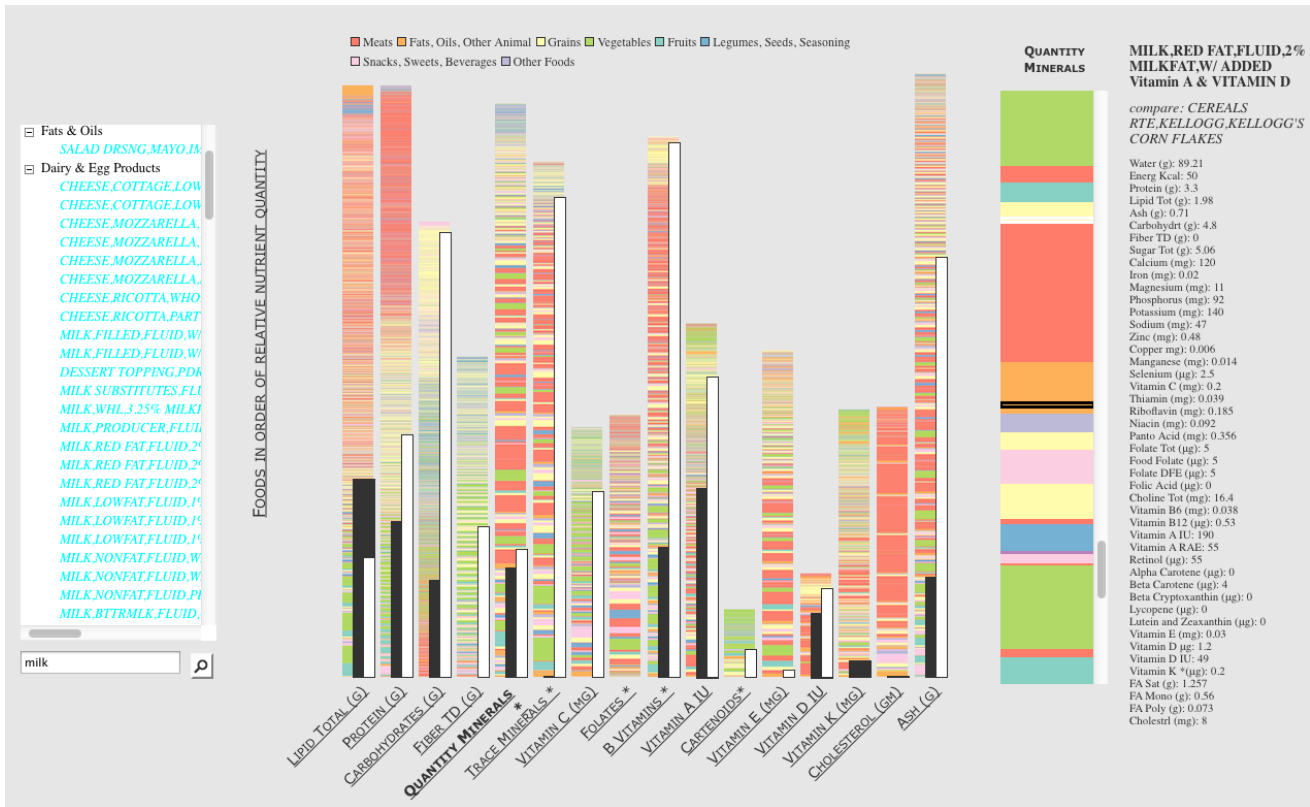


Figure 10. Illustrating scenario 6.3, comparing profiles of two foods

One of the primary strengths of NUT over the other nutritional visualizations discussed is that it succeeds in giving the user a broad understanding of whole foods. Rather than focusing on a nutrients in isolation, or a small combination of nutrients, NUT lets the user see easily how a food stacks up against other possible options they could eat in terms of it's overall nutritional value. By first presenting foods in a relative context, rather than overall quantity, it also helps the user understand their food options in the context of other foods they could be eating, helping users to think about trade-offs rather than absolutes.

The comparison view does an effective job of providing the comparisons in the context of the Overview and the whole food. The use of bars to emphasize the relative quantity allows the users to quickly get a sense of overall quantity for each food with respect to other foods, and also to easily make comparisons.

7.1.2 Understanding nutrients

Like *Visualizing Nutrition*, NUT supports the user in looking for the top foods containing the most of each nutrient. However, inspection of the nutrient values through NUT has shown, much like the example of wild game in section 6.2, these top foods also tend to be somewhat out of the ordinary. Unusual foods are not particularly helpful for ordinary users, and users typically eat multiple sources of a nutrient in a day – they don't need to get it all from one source. NUT improves on this limitation by allowing users to get a sense of the food types that dominate all the ranges of nutritional values, from food groups that tend to have a lot or a little of something, to typical foods that just fall somewhere in the middle.

7.2 Weaknesses and Limitations

7.2.1 Relative ordering

Casual input on NUT from a handful of users has shown that users tend to misinterpret the height of columns as showing absolute quantity rather than a relative ordering on first use. The use of bars for to highlight primary selections and comparisons further compounds this problem, as users are most familiar with bar charts in the context of quantitative information. This encourages the mistaken understanding that a higher bar in one column than another means that nutrient exists in a higher absolute quantity within the food.

The application is designed to allow comparisons between foods in the same column with regards to relative quantity, and I argue that making absolute comparisons between nutrients does not actually make a lot of sense. This common misinterpretation is therefore worrying, and presents a considerable limitation given that the application is intended for casual users on the web, who should find the tool straightforward to use. This problem may require drastic changes to the overview encoding to solve, but it is worth first investigating whether simply improving the labelling and supporting documentation can address it.

7.2.2 Calculated Dimensions

The choice of factors to collapse into new dimensions, as discussed in section 3.2.2 presents a trade off in terms of usability in terms of making decisions about nutrition. While users may want to maximize their intake of certain nutrients (e.g. Calcium), they may also wish to minimize their intake of others (e.g. Sodium). When nutrient dimensions are collapsed together, as in the case of Sodium and Calcium into the quantity minerals, the

resulting dimension tends to neutralize any extreme differences, making them difficult to see in the overview.

Collapsing together nutrients that users tend to want to minimize into separate groups from those that users tend to want to maximize could potentially solve this limitation. However, it is likely that no one set of dimensions will work for everyone, since these types of nutritional demands can change dramatically between users with different personal health needs. Users with low blood pressure, for example, may actually be interested in high salt foods as opposed to low salt foods.

7.2.3 Color encodings

The eight colours used to encode food groups, with the addition of the background colour, and the two colours used for highlighting foods and comparison foods, puts the total number of colours used in the interface at 11. This is at the high end of the guidelines for distinguishable hues for qualitative information, generally placed between 6 and 12 [21]. Given the small size of some lines, fewer colours might have been a more useable choice. However, experimentation with fewer colours for the food group encoding meant creating even higher-level groupings of food groups, which made it more difficult to see meaningful differences between food groups in the overview. Further, from a domain perspective the groupings of sub categories became even less intuitive, and it would have been necessary to lump even more unrelated foods together.

An evaluation of the interface using Vischeck [22], a tool for simulating colour-blindness, shows that NUT is not completely usable by those with colour vision deficiencies because of the low luminosity contrast between some of the pastel colours. The application could be improved by selecting a different palette scheme of qualitative colours from ColorBrewer2.0. Steven Gardner's evaluation of the ColourBrewer schemes [23] suggests that two of the alternative eight class schemes, while potentially still presenting problems for users with colour-vision deficiencies, are more likely to be usable than the pastel scheme currently implemented for NUT.

7.2.4 Nutrient view interaction and information hiding

The nutrient view is a necessary stand-in to allow users to interact with the columns shown in the overview display, since the overview itself is too dense to effectively navigate. The choice to show details only on demand is a common for overviews, but the Nutrient view is more of a middle step between overview and detail – it is not quite one or the other.

Through casual discussion with users I have found that once a user has chosen to inspect a nutrient, they often wish for more detailed information to be shown than simply a zoomed in view of the colours. The only interesting details that jump out at you in the current version are changes in colours showing foods from different food groups. Foods in the same food group may also have surprising or interesting differences; while differences require looking at the nutrient values, many interesting things can be found just by inspecting the name. The need to hover over a food to view these names can therefore be cumbersome, and it's easy for the user to forget what they've seen. It may be useful to allow the user to see a list of text labels corresponding to a subsection of the foods shown in the Nutrient view. This could take many forms, such as a moveable and/or adjustable lens that displays the labels beside or within the view.

7.2.5 Processing limitations

One of the largest limitations I encountered with the Processing language was the difficulty it had in handling large amounts of

data. My early implementations were extremely slow, in part due to my naïve implementation of data structures and search algorithms. I spent a small amount of time investigating optimization early on, but it was not within the scope of the project to find a way to draw the overview on the fly in my implementation. As a result, the overview in NUT is actually an image generated from a separate processing sketch.

While this approach is substantially faster than my earlier implementations, it means that the overview cannot be changed on the fly. This limits the ability of future work to update or filter the data displayed in the overview based on user-selections. In order to pursue those types of improvement the architecture would either need to be substantially optimized and reworked, or completely recoded using a faster system like d3 [24].

7.2.6 Dataset limitations

The USDA NDSR dataset has a number of limitations, the primary one being the somewhat inconsistent and complex categorization of foods. While some categories, like 'Poultry' are straightforward for a user to understand, others, like 'Ethnic foods', don't communicate a clear or consistent understanding. This makes high level classification somewhat difficult, and can potentially be confusing for users. The example of the wild game in Section 6.1 highlights this problem, as many users would expect to find such foods under 'meat'.

A second problem stems from the fact that nutrient content varies widely between individual foods items that are supposedly the same. Two carrots grown in different soils, while they may contain similar types of nutrients, do not have exactly the same amounts. A similar problem is encountered when considering individual foods as opposed to 'meals' or products made out of multiple foods, since every recipe varies. Thus, the information provided by NUT is inherently imprecise, and can only be used as a loose guideline.

Lastly, the naming conventions for foods are not consistent. Interesting information, such as whether the food is fresh or frozen, is often contained in the labels and I was interested in using this information to support filtering. Ultimately however, extracting this information automatically was not straightforward or in scope because the labels are not consistently constructed between all foods,

7.3 Lessons Learned

7.3.1 Processing and Processing.js integration

In the first parts of my analysis phase I used Processing for my implementation, but I later switched to Processing.js so that I could implement my solution for the web. Processing.js has limited support for the large number of libraries provided for Processing, one of which I relied on heavily for importing my data. As a result, I had to substantially rework my code part way through in order to use AJAX and JSON to import my data. In the future, I will spend a bit more time up front understanding the limitations of a toolkit before completely diving in.

I also significantly underestimated the complexity involved in integrating Processing sketches into a larger web application. On the one hand, my work was simplified through the use of jQuery and HTML user interface widgets and text. However, the learning curve to understand how to integrate Processing sketches so that they could be controlled through external JavaScript was quite steep, and anything beyond the simplest steps was very poorly documented. As a result, a lot of my time was spent on learning and developing through simple trial and error.

7.3.2 Scope and Time Management

At the very start of my project I did not anticipate the amount of work that could go into analysing such a simple dataset. I also did not fully appreciate the benefit that such an analysis, although simple, would have on my final design. As a result of underestimating this step, my scope ballooned while I lost significant time to actual implement my project.

Because I underestimated the time involved in integrating Processing.js and preparing the data, as discussed above and in Section 5.2, I had to rush at the end in order to complete an integrated system. A number of improvements to the UI were thus left unfinished, as discussed in the next section. Thanks to this experience, however, I feel that I am much better equipped to manage my time for such a project in the future.

7.4 Future Work

7.4.1 Improvements to current version

A number of small immediate improvements could be made to make NUT more useable. Most were always intended to be a part of the original implementation, but remain unfinished because of a lack of time.

Ideally, the food group colour encoding should be shown next to the groups in the Food list view and also in the detail view. I was prevented from implementing this in time because of the complexity of the code in the jsTree library. This would free up clutter in the interface by removing the need for a legend. Difficulties working with the jsTree also prevented me from updating primary and comparison selections in the list when made from the Nutrient view. The Food list would be improved further through the addition of amounts next to the highest-level categories to indicate the total number of food they contain.

A number of intended aspects of the Nutrient view remain unfinished. First, the scroll bar should have a marking along side it to show the user the position of the primary and comparison selections in the view when they are off-screen. Linking between the Nutrient view and the overview also needs to be improved. When a user selects a nutrient from the overview, my intention was to have the Nutrient view automatically scroll to the portion of the column that was actually clicked on. The current range in the Nutrient view could be more clearly understood in context of the overview by drawing an outline around the same portion in the related overview column.

For collapsed dimensions, the mouse-over details in the Nutrient view currently only show the normalized value that was calculated. This issue is illustrated in figure 6. Ideally, the mouse over should actually show values of each of the original dimensions because this information is significantly more informative to the user. Without it, the user has to select a food to see the values for the derived nutrient. Similarly, the membership of the nutrients shown in the detail view to the calculated groups should be shown. In the current implementation, users do not know which nutrients were used to calculate the new dimensions.

Lastly, I realized after the prototype was completed that the current interface makes it difficult to keep track of the primary selection and the comparison selection. Because these are off to the side, it is easy to forget them while exploring the rest of the interface. An immediate improvement could be to double-encode the labels in white and black, although some changes to the background might give the white enough contrast. The location could also be moved into the centre to make them more salient.

7.4.2 Drill-down into calculated dimensions

Although the calculated dimensions simplify the overview and make it easier for the user to quickly understand the food, these dimensions would be interesting to some people. Future work could add the ability for users to drill down into calculated dimensions. For example, drilling into the B Vitamins would replace the general overview dimensions with a set of all the different B Vitamin dimensions, which could then be navigated in much the same way as the current Overview.

7.4.3 Daily recommended intake

Future work could also explore adding information about the daily-recommended intake of nutrients, in order to provide context to the information displayed in the Nutrient view. This could take the form of small bars displayed a line showing the recommended values, making it easy for users to see how well a 100g portion of the food (or potentially a recommended portion) meets the requirement.

7.4.4 Currents limitations

Finally, future work should address the limitations outlined in Section 7.2. While some of the limitations may not have one solution, both the colour choices and the Nutrient view interaction are very addressable problems.

8 CONCLUSION

This report outlines NUT, the nutritional understanding tool. The primary contribution is the implementation of a high-dimensional analysis tool with multiple-linked views for the domain of nutritional information. A secondary contribution comes from the analysis and discussion of how to simplify the USDA NNDSR dataset for the casual user and the ways in which this informed the final design of NUT. The application's strengths and current limitations are discussed, and usage scenarios for NUT with respect to understanding nutrient content in the context of the whole food are presented. Using NUT casual users and food nerds alike can make informed, healthy choices about what to eat by understanding the relative nutritional content of their food.

REFERENCES

- [1] Pollan, M. In *Defense of Food: An Eater's Manifesto*, New York: Penguin Press, 2008.
- [2] U.S. Department of Agriculture, Agricultural Research Service. USDA National Nutrient Database for Standard Reference, Release 24. <http://www.ars.usda.gov/ba/bhnrc/nd1>, 2011 (accessed October 2011)
- [3] Ankerst, M., Keim, D., & Driegel, H.. Circle segments: A technique for visually exploring large multidimensional data sets. In *Proc. of Visualization'96*.
- [4] Keim, D., Hao, M. C., Ladisch, J., Meichun Hsu, & Dayal, U. Pixel bar charts: a new technique for visualizing large multi-attribute data sets without aggregation. *IEEE Symp. Information Visualization (InfoVis)*, pages 113-120, 2009.
- [5] Ziegler, H., Jenny, M., Gruse, T., & Keim, D. A. Visual market sector analysis for financial time series data. In *Proc. IEEE Conference on Visual Analytics Software and Technologies (VAST)*, pages 83-90, 2010.
- [6] Meyer, M., Munzner, T., & Pfister, H. MizBee: A Multiscale Synteny Browser. In *Proc. IEEE Symp. Information Visualization (InfoVis)*, pages 897-904, 2009.
- [7] Lonely Datum. Visualizing Nutrition. <http://lonelydatum.com/p/food/>, 2011 (accessed October 2011).

- [8] Many Eyes. Treemap of USDA National Nutrient Database. <http://www-958.ibm.com/me/visualizations/treemap-of-usda-national-nutrient->, 2010 (accessed October 2011).
- [9] gGobi. www.ggobi.org, 2008 (accessed November 2011).
- [10] Xmdv tool. <http://davis.wpi.edu/xmdv>, 2010 (accessed November 2011).
- [11] Dimstiller, Ingram, S., Munzner, T., Irvine, V., Tory, M., Bergner, s., & Möller, T. In *Proc. IEEE Conference on Visual Analytics Software and Technologies (VAST)*, pages 3-10. 2010.
- [12] Reas, C., Fry, B., & Maeda, J. *Processing: A Programming Handbook for Visual Designers and Artists*. The MIT Press, 2007.
- [13] Processing.js. <http://processingjs.org>, 2009 (accessed November 2011)
- [14] Diehl, S., Beck, F., & Burch, M. Uncovering Strengths and Weaknesses of Radial Visualizations – an Empirical Approach. In *IEEE Transactions on Visualization and Computer Graphics (TCVG)*, 16:(6), 2010.
- [15] Brewer, C. CColorBrewer2.0. <http://colorbrewer.org>, 2009 (accessed November 2011).
- [16] jQuery. <http://jquery.com>, 2010 (accessed November 2011).
- [17] jsTree: jQuery tree plugin. www.jstree.com, 2011 (accessed November 2011).
- [18] Introducing JSON. www.json.org (accessed November 2011).
- [19] Parker, C. CSV to JSON. <http://www.cpark15.com/code/utilities/csv-to-json>, 2011 (accessed November 2011).
- [20] Mr. Data Converter. http://www.shancarter.com/data_converter, 2011 (accessed November 2011).
- [21] Munzner, T. “Chapter 3 Visual Encoding Principles”, In *Information Visualization: Principles, Methods, and Practice*, Pre-Publication Draft, 2011.
- [22] Vischeck. www.vischeck.com, 2008 (accessed December 2011)
- [23] Gardner, S. D. Evaluation of the ColorBrewer color schemes for accommodation of map readers with impaired color vision. Master’s Thesis, Pennsylvania State University, 2005. http://www.personal.psu.edu/cab38/ColorBrewer/Steve_Gardner_thesis_PSU.pdf (accessed December 2011).
- [24] Bostock, M. d3.js. <http://mbostock.github.com/d3>, 2011 (accessed November 2011).

Grouping 1	Grouping 2	Grouping 3	Grouping 4	Grouping 5	Grouping 6	Grouping 7	Grouping 7
Selenium	Vitamin B ₁ (thiamine)	Iron	Vitamin B ₁₂	Saturated Fats	Beta Cryptoxanthin	Vitamin B ₉ (folic acid)	Vitamin A
Zinc	Vitamin B ₂ (riboflavin)	Copper	Vitamin B ₆	Polysaturated fats	Beta Carotene	Folates total	Retinol
	Vitamin B ₃ (niacin)		Choline	Monosaturated fats	Alpha Carotene	Folates DFE	
			Vitamin B ₅ (pantothenic acid)	Cholesterol	Lycopene (Carotenes)		
					Lutein and zeaxanthin		

Table 1. Groupings created with second prototype

Retained Dimensions	New (Collapsed) Dimensions
Lipid Totals	Carotenoids (originally 5 dim.)
Protein	Quantity minerals (from 6 dim.)
Carbohydrates	Trace Minerals (from 7 dim.)
Vitamins A, C, E, D, K	B Vitamins (from 7 dim.)
Cholesterol	Folates (from 4 dim.)
Ash	

Table 2. Retained and new derived dimensions used in the NUT Overview

Cartenoids	Quantity Minerals	Trace Minerals	B Vitamins	Folates
Beta Cryptoxanthin	Sodium	Selenium	Vitamin B ₁₂	Vitamin B ₉ (folic acid)
Beta Carotene	Potassium	Zinc	Vitamin B ₆	Folates total
Alpha Carotene	Phosphorus	Selenium	Choline	
Lycopene (Carotenes)	Magnesium	Zinc	Vitamin B ₅ (pantothenic acid)	
Lutein and zeaxanthin	Calcium	Manganese	Vitamin B ₁ (thiamine)	
Lutein and zeaxanthin	Calcium	Manganese	Vitamin B ₂ (riboflavin)	
Lutein and zeaxanthin	Calcium	Manganese	Vitamin B ₃ (niacin)	

Table 3. Composition of derived nutrients