Health Figures: An Open Source JavaScript Library for Health Data Visualization

Andres Ledesma[*], Hannu Nieminen and Mohammed Al-Musawi

Abstract

Background: The way we look at data has a great impact on how we can understand it, particularly when the data is related to health and wellness. Due to the increased use of self-tracking devices and the ongoing shift towards preventive medicine, better understanding of our health data is an important part of improving the general welfare of the citizens. Electronic Health Records, self-tracking devices and mobile applications provide a rich variety of data but it often becomes complex and difficult to understand. Recent insight-based studies have shown that normalized visualizations of health data can provides a better understanding on health behaviors and their relationship with the overall wellness. Based on the hGraph visualization developed earlier, we implemented an open source library that addresses some usability issues found on the hGraph and extends its functionality to further enhance health data understanding and analysis including a time-based visualization of the data. We named this library hFigures. The hFigures library provides features to improve the understanding of health data. The library enables comparison of multiple health measurements organizing the measurements in groups separated by a structured layout. The purpose is to enable a visual representation of the evolution of health measurements in a comprehensive and holistic manner.

Results: We built an application using the hFigures library for health data visualization of a modelled patient in a health coaching program. We performed a user evaluation with Heuristic Evaluation, Controlled User Testing and Usability Questionnaires. The results were satisfactory, users found the library informative and usable with a high score in user satisfaction.

Conclusions: We developed hFigures, an open source library for visualizing a holistic, normalized and comprehensive graphical representation of health data. The idea is based on the concept of the hGraph but it provides additional key features, including a comparison of multiple health measurements over time. We evaluated the library in an application and obtained positive results.

Keywords: data visualization; health data; health informatics; javascript

Introduction

The ongoing shift from reactive to preventive medicine requires that citizens have the skills and means to take an active role in developing and maintaining their wellness. Individuals have greater access to self-tracking devices and applications, from sports tracking application to genome sequence analysis. These services and devices produce large amounts of data. In addition, Electronic Health Records are increasingly replacing paper records in hospitals and clinics around the world. The combination of these large and heterogeneous data sources can provide a "predictive, preventive, person-

alized and participatory" ecosystem to the benefit of the general welfare [1].

To better understand our health, we need to combine heterogeneous data sources and present the information to the user with a comprehensive and holistic approach. In order to accomplish this, "new hybrid professionals" need to integrate health information technologies and visualization design [2]. The goal is to provide tools for individuals to take better decisions regarding their health. Similarly, doctors and other medical experts need tools and solutions for getting a comprehensive view of the patients health, combining togehter patient's own measurements and clinical data.

Graphical perception is the ability of an individual to decode the information displayed as graphi-

[*]Correspondence: andres.ledesma@tut.fi
Department of Signal Processing, Tampere University of Technology, Korkeakoulunkatu 10, 33720 Tampere, Finland
Full list of author information is available at the end of the article

cal objects [3] and it is a widely researched field [4]. Graphical perception affect how we understand visualized information. In the context of health data, it remains a challenge to design graphical representations for non-medical experts. Graphical representations in this context should enhance the users' ability to understand their health situation and take informed decisions. With this "deep" understanding on the health situation, individuals can move towards healthier behaviors.

An innovative approach for health data visualization is the Health Graph (hGraph), released publicly by MITRE corporation [5]. Based on a previous study [6], we found that the hGraph enhances deep understanding of health data and enables the user to create meaningful health insights based on the interrelationships between the measurements. However, the hGraph shows a static overview of a persons wellness. That is, the values of multiple measurements at one point in time. While this visualization is useful for the purpose of understanding with a quick glance the overall situation, it lacks features such as a clear division of measurements according to their category, a distribution of the labels to avoid clutter and other aspects. In our previous study [6], users provided feedback on what the hGraph was missing to help them understand in an easier and more comprehensive manner the data in the experiment.

We used the feedback from the study along with a user-centered design approach to build our visualization library. We added features that users considered helpful in understanding the health data better. In order to build a visualization tool that enables users to take informed decisions regarding their health habits, we addressed these requests and extended the original design and functionality of the hGraph. We open sourced our library under the name Health Figures, or hFigures for short.

This article presents the research background on health data visualizations and an overview on the hGraph. The article details the design and the implementation implications of the user feedback in building the hFigures library. We summarize the features we implemented and discuss how this library can be used in Personal Health Informatics and in Health Care processes.

Background
Related Work
Visualization tools have mostly focused on Healthcare Information Systems and Electronic Health Records (EHR) [2]. For instance, TimeLine is a software developed to retrieve data from several sources and presented in a hierarchical and timeline based structure where clinicians can browse chronologically through existing EHRs including MRI [7].

Additionally, the growing market for mobile health applications (mHealth) have drawn the attention of researchers, developers and investors [8]. These applications provide large volumes of personal health data. While the market and demand are expected to grow, the use of the data has the potential to contribute to a better understanding of our health.

Goetze [9] demonstrated the impact of data visualization as means to represent health data in a comprehensive manner. He conducted a project that redesigned laboratory test results from numerical tables into colored graphics. He demonstrated that the patients were able to understand better their health situation when presented with the new designs.

Data integration for health monitoring as a Big Data process for personalized medicine has been approached by [10]. The visualization of this information uses traditional bar and pie charts to report to the user a historical view of a variety of data including mental, social, physical aspects. The novelty of this work is the integration of heterogeneous data sources while the presentation of the information was done following existing graphical representations.

An extensive choice of graphical representation is listed and explained by S. Few [11]. These techniques have been studied and used widely among researchers and individuals alike. Examples include: bar, stacked bar, line and bullet graphs. These visualizations can be combined to provide a personalized wellness indicator system, as proposed by Soomlek and Benedicenti [12].

DeRidder *et al.* developed a combined approach that retrieves data from Personal Health Records (PHRs), and presents them to individual patients using a "3D medical graphical avatar" [13]. The solution is built using HTML5 and WebGL to render 3D graphics using the web browser. Patients can browse "regions of interests" on their avatar and explore further the information contained in EHRs as well as in PHRs.

However, as stated by Shneiderman *et al.* [14], new visual representations are needed for "systematic yet flexible visual analytics processes". We present an existing tentative solution known as the hGraph, released publicly by MITRE corporation. We describe its main advantages and how they can address these challenges. In a previous study we identified possible improvements and based on our own implementation we addressed these issues and extended the features of the hGraph to better address complex data sources.

The Health Graph
The health graph, or hGraph was developed by MITRE Corporation and released to the public in 2010

under the Apache v2.0 license. The design intention of the hGraph is to facilitate the graphical representation and understanding of health data. The data can come from a wide-range of sources such as laboratory tests, physical activity, nutrition, sleep monitors and other sources. The domain of this visualization technique includes personal health informatics, EHR and Personal Health Record (PHR) visualization [5].

Design
The design consists of a circular space with an area defined by to circumferences. The area represents the minimum and maximum recommended values for a given measurement. For instance, the minimum and maximum fat percentage of a person in a given age. The measurements are represented as circles and their position in the circular space represents how far or close they are from the recommended values. The position is normalized according to the recommended values. The values are distributed in a circular space.

Graph A graph is formed by joining the data points around the circular area. This polygon or graph reveals a pattern and its shape provides a quick overview of the general situation of all the values and how they deviate from the recommendations.

Highlight Deviations The hGraph design highlights values outside of the recommendation by using the red color on the data points and by modifying the shape of the graph.

Rationale The rationale of the hGraph is that if the same measurements are plotted in the same order for various cases, then the graph patterns can reveal similar shapes associated with certain health conditions.

Web-based Solution
Web-based solutions for data visualization provide flexibility, as they can be accessed by any web browser, either from mobile devices or personal computers. The hGraph uses a web approach via HyperText Markup Language (HTML) and Scalable Vector Graphics (SVG). The programming language of the library is JavaScript and is built using the Data-Driven Documents library. Data-Driven Documents (D3.js) library provides free access to the Document Object Model (DOM), which is the substrate that enables the programmer to interface with the graphical representations in a web browser [15].

hGraph as an Insightful Visualization
We conducted a study based on the approach proposed by C. North [16] which compares visualizations based on how well users can derive meaningful insights from the data based on the visualization. We evaluated the hGraph visualization along with four alternatives using the Graphical Perception Framework proposed by Cleveland McGill [4]. The evaluation followed the insight-based methodology similar to other experiments [17]. Additionally we interviewed the hGraph users on any issues and problems they experienced in understanding the graphical representation of the data. The experiment results are discussed in [6]. The objective of the experiment was to compare different visualizations and determine how they can enable users to better understand their overall health situation, as well as the possible causes behind health issues. Out of the compared solutions, the hGraph was found to be the most effective solutions for creating meaningful insights and to help users to better understand the data.

The hGraph scales to large amounts of data entries since the measurements, represented as circles, are distributed among a circumference. Other visualizations, such as bar charts, lines or scattered plots, can quickly take a large space when plotting large datasets. As more measurements are added, the circular space utilized in the hGraph will eventually stretch to accommodate large datasets, but the distribution stays uniform along the circle and the space required is still less than with other alternative visualizations.

Figure 6 is an example of the hGraph visualization. The figure was extracted as a snapshot as the library generates an SVG document structure that cannot be exported outside the browser. We address this issue in the next section of the article. The hGraph hides the measurements when the zoom level is low, meaning that the user has zoomed out. The shape is an average of the deviation of each measurement under the same category. When the user zooms in, the details are revealed and the rest of the information becomes visible, that is the numerical values and positions (with respect to the recommendation) of the measurements.

Implementation
Throughout the design and implementation of our library, we adopted a user-centered approach. Invited users gave us continuous feedback on how they would understand the data better and what features would improve their experience with the library. Based on iterative design sessions and implementation of prototypes we developed the features based on how they enhance the usability and utility of the library. We named our library Health Figures (hFigures) based on the fact that we obtained our inspiration from the hGraph but eventually extended the library to include multiple graphs giving the notion of several intertwined figures.

This chapter explains the implementation process in our approach based on user feedback and on previous experiments [6].

Data Source

We use a JSON (JavaScript Object Notation) format to read the data, in which the measurements are grouped according to their categories. The groups contain an array of samples, which represent the values obtained from a measurement (steps per day, cholesterol, triglycerides, blood sugar or depression level using [18]). The samples contain a timestamp in Unix Epoch format and the value of the measurement. The Unix Epoch format is the number of seconds since the first of January 1970, Greenwich Meridian Time (GMT). An example of the data source is in figure 1, it shows the first measurement of the group "Blood Pressure" which in this case is comprised by Systolic and Diastolic measurements and each of them have two samples taken at two particular times, Friday 9th of January 2015 at 10:10:24 GMT (1420798224 — seconds) and Thursday 12th of February 2015 at 12:05:20 GMT (1423742720 — seconds).

SVG Document Export

The SVG document structure we designed in our implementation can be exported to a separate file outside the web browser. The short-term objective is to build a tool-kit that enables researchers to visualize their data with our implementation so they can use the generated SVG file in articles, posters, presentations or other applications. For instance, figure 7 has been exported as an SVG document from the browser into this article. SVG export is possible due to the rendering of our algorithm which does not depend on JavaScript or Cascade Style Sheet (CSS) styling properties to produce a finalized document. The library build the entire image as a stand-alone document. The hGraph library unfortunately does not produce a complete document but instead depends on CSS and JavaScript code to make the image visible.

Constant Graph Shape

Figure 6 shows an hGraph example and figure 7 shows an hFigures example. The hGraph computes the average of the deviation of the measurements in order to show the polygon or graph, as a representation of the overall health assessment. However, some measurement might deviate towards a lower value and while others towards a higher one, thus the average position would be roughly the middle recommended area. For this reason, hFigures does not change the shape of the graph if the user zooms in or out. Showing and hiding the measurement labels is the only reaction to

the zooming events from the user at the moment. This avoids clutter when the user wished to have a quick glance at the picture but keeps the graph with the same shape.

Layout Construction

The measurement groups are represented using a circular layout divided in sectors. The goal is to have a clear division between the groups as they represent the different aspects of the overall health. hFigures uses the d3 pie chart layout and modifies the data source provided to the layout. All the measurements have the same numerical value and at the end of the group, we insert an extra value in order to leave a blank space between the circular area sectors. The result is visible in the hFigures example shown in figure 7 and the code that produced this visualization is in figure 2. The pie layout constructs the sectors of the circular area based on a data source. When we provide an array of numbers, the layout uses the numbers to calculate the proportions of the area. In order to achieve the layout construction that we have designed, the array has the same constant number multiple times, the number of measurements plus an additional number for each group.

Color-coded Entries

The data source can contain additional sets of value ranges. For instance a warning range of values can let the users know when a value has reached a level that requires attention but has not yet reached a critical point. We followed the users' feedback that recommends a traffic light-based approach. The green color means that the values are within the recommended, yellow suggest a warning or follow-up action needed and the red indicated a critical threshold has been passed.

In the implementation, the library verifies if these additional ranges are present in the measurement definition. In order to verify if the property of the object exists, JavaScript provides a qualifier method, `typeof`. The returned value must be compared with the keyword definition for properties that are not present in an object, the keyword `undefined` has been suggested by Mozilla Developer Network [19], a highly reputable source for Web development. The code is shown in figure 5.

Multiple Graph

The dataset is structured as a set of measurements where each has its own collection of samples. In order to compare the evolution of these measurements, the hFigures library allows the graphical representation of any number of samples. The result is a set of graphs

or polygons overlapping or stacking with each other. In order to differentiate them, we use a lighter set of colors so that the users can see the difference between two points in time. As an example, figure 7 shows two different samples for each measurement. This example portrays a modelled person that has been active in a health coach program. Some measurements have improved and are closer to the recommendation. Users repeatedly expressed that it would be very helpful to visualize two or more different points in time so as to compare how the person has evolved.

Including multiple graphs has implications in the structure and procedures of the visualization construction. For instance, we structured the SVG document such that each measurement includes one or many plotted circles that map to each sample. The measurement labels need to be positioned considering that a plotted circles can (and probably will) overlap. This is challenge that we address in the next section by finding an optimal label space distribution to avoid labels from overlapping and also to reduce the clutter in the visualization space.

Label Space Distribution
After the measurements are plotted, the labels are added to increase readability. The position of the label needs to be defined within a given range to avoid overlaps and clutter.

Labels need to avoid overlapping with each other and with their measurements. To solve the label overlapping problem we implemented an algorithm that starts by ordering the labels by their angular position, that is the angle at which the measurement is positioned. The next step is to calculate the height of the label and position it over the previous one in the direction that goes from the center of the visualization area upwards or downwards (depending on the angle). The idea is to begin with the center of the area, either to the left or to the right of the circles, then we work our way up or down drawing the labels into the SVG document. We add the labels as SVG elements and the use the `transform` property to position them in the corresponding place. Figure 9 shows the spacing between the labels using the algorithm when drawing the labels from the center to the upper right corner. For each of the four quadrants, the library calls the method shown in figure 4 which computed the position of the label as we described.

As mentioned before, labels can also overlap with measurement circles. To avoid this problem we calculate the maximum radius from the center of the visualization area to the highest value of a measurement sample. From that starting point, we place the label in that position. In other words, for each measurement,

we find the largest value of the samples. Figure 8 shows a sector of the hFigures where the sugar measurement label has been pushed out for a few pixels in order to avoid overlapping it with the red circle. The rest of the labels adjust to that position by leaving a user-defined margin.

$$r_{label} = max(\bigcup_{i}^{n} \{r_i\}) + margin \qquad (1)$$

The radius for the label is the maximum value of the samples translated as graphical coordinates plus a margin. Equation 1 obtains the label radius r_{label} given the radii of the samples of a measurement plus the default margin m.

Evaluation
In order to test the hFigures library, we designed an application for health monitoring and evolution of a patient's status over time. The application visualized a health coaching program of a modelled patient. The application has thee components: activity timeline, the hFigures visualization and longitudinal measurements.

The **activity timeline** represents the health interventions the modelled patient has done over a period of time. At certain times, the modelled patient has a snapshot of a set of measurements visualized as **longitudinal measurements** also represented in the **hFigures** visualization. The changes in the status of the patient are displayed by two plots of the hFigures. Adjustable points in time animate these two hFgures. The adjustable points are represented as lines in the activity timeline and the longitudinal measurements. Users are allowed to drag and drop these lines to see the status of the patient at those specific times.

Methods
Nielsen suggests that "'usability has multiple components and is traditionally associated with the five usability attributes, which are learnability, efficiency, memorability, errors, and satisfaction" [20]. Following these attributes, we tested the application using heuristic evaluation, controlled user testing and usability questionnaires. These methods address the attributes mentioned by Nielsen.

Heuristic Evaluation
Usability testing requires at least one expert in the area of human-computer interaction. For our evaluation we recruited 3 experts who assess the application using Nielsen's heuristics [20]. The evaluation has 11 metrics evaluated using a seven point Likert scale. Experts assess the dialogue and use of language, estimate the memory load required to use the system, verify the error messages and assess errors avoidance of the application, among other aspects of the heuristics.

Controlled User Testing

Rubin and Chisnell [21] as well as Nielsen [20] recommend to design a set of tasks for users to perform in a controlled environment. We can then measure the time to complete the tasks, errors (crucial and non-crucial) and whether the task was successfully completed. Controlled user testing allows us to find out any obstacles the application that prevent users from completing a set of tasks. These tasks are the intended use of the application, thus the goal is to minimize errors and the time required to complete them. The tasks corresponding to the hFigures component are listed below, the objective was to determine if users understand the visualization and obtained an understanding on the overall health condition of the modelled patient.

1 How many areas of health are displayed in the hFigures?
2 Choose one of these areas and point to its measurements.
3 Identify one measurement inside the recommended values and another one outside.
4 Identify the measurement that is the furthest from the recommended values.
5 What does the green, yellow and red circles mean?
6 Has the overall health improved after coaching?
7 Which area of health has improved the most after health coaching?
8 Which measurements show the biggest improvement?
9 Understand the difference between the points inside and outside the hFigures.

Usability Questionnaires

In order to measure a broader spectrum of usability aspects, we used three questionnaires. These are Computer System Usability Questionnaire (CSUQ), After Scenario Questionnaire and Usefullness (ASQU), Satisfaction and Ease-of-Use Questionnaire (USE). All three questionnaires use a seven point Likert scale.

Computer System Usability Questionnaire

After Scenario Questionnaire

Usefulness, Satisfaction and Ease of Use Questionnaire

Data Model

Similar to the study we conducted in the insight-based methodology [6], we modeled a patient using clinical expertise of a physician along with the most common symptoms for developing Type II Diabetes. The modelled patient consisted of a set of measurements over time comprised of the following parameters:

- Nutrition
- Physical Activity
- Emotional Wellbeing

Results

The user-centered approach provided us timely feedback to enhance the implementation of features requested by the users. Table 1 lists the supported features of hFigures and hGraph. This section summarized the key differences and improvements of hFigures.

The key improvement of hFigures is the addition of multiple graphs as a mechanism to compare the values of the health measurements over time.

The immutable shape of the graph presents the same information (values of the measurements in respect to the recommended target) regardless of zooming. This feature shows the data "as is" without calculating average, mean or deviation. Users stressed the importance of graphically representing the information without any calculations such as mean or accumulated values. The users expressed that showing the measurement values in hFigures helped them to derive valuable insights with just a quick glance at the data, for instance they identified measurement that fall outside of the recommended range easier and without requiring them to zoom in or out.

The hFigures library does not calculate an overall score since the users considered that this task should be the sole responsibility of a health care professional. An overall score also depends on each person under a case by case basis. For instance, the hGraph allows the user to assign weights to each measurements' value, the score is then calculated summing the value of the weights times the measurement's deviation from the recommendation. The users participating in the design process of our library expressed that an overall score would complicate the integration of the library into daily health care processes as specialists would need to review case by case to find the adequate score formula, which means specifying the weights (importance) of each measurement for a given person.

Measurements in the hFigures library are grouped in sectors which represent the category they belong to. Grouped measurements allow a clear division of categories resulting in enhanced understanding on how certain areas of wellness have changed and how, if any, they affect each other. The sectors remain visible regardless of the zoom level, users expressed that this feature provides an informative approach as the categories are always showing to which category the measurements belong to. Hiding the category labels and displaying the measurements without divisions would

complicate understanding the status of health categories, such as sleep, nutrition, physical activity and others. Users expressed their confusion when they were unable to determine when a category starts or ends after zooming in and out of the hGraph.

The possibility to export the generated figure as an SVG file, allows the integration into research articles, presentations, websites, posters and other Software applications to further enhance the utility of the hFigures.

Discussion

The value of a data visualization depends on the knowledge that it can convey to the public. In this section, we claim that hFigures has the potential to be used both in the clinical and personal wellness applications. Large amounts of measurements do not clutter the visualization area as a result of our implementation, figure 10 shows an example of a complex holistic visualization of a modelled patient. The comparison of multiple graphs can provide a meaningful visualization to individuals and clinicians alike. The implementation of hFigures follows an extensible approach and even though it was designed to be used for health data visualization, any dataset that has a target range of values as a reference, can be visualized using this library. As an example, hFigures could be utilized for example for visualizing status of measurements in industrial process monitoring.

Translation to Health Care

Nowadays, the formats of EHR and application data follow a large variety of formats. It remains an open challenge to incorporate these data sources into a holistic repository that portrays a patient using every possible measurement, from laboratory tests to activity trackers. Currently, most of the data sources contain the date when the sample was extracted, whether it is a tracking device or a blood test. This sample date already provides the timestamp required by the hFigures data source file. The values of the samples are the main object of study in a measurement, for instance the levels of cholesterol or sugar at a given time, the number of steps per day, the percentage of body fat and several others. This information can be transformed in a simple process to build the hFigures data source following the JSON structure.

hFigures is a visualization library based on Web technologies, it uses a Web browser and the rendering of the SVG is compatible with current HTML standards. Thus providing interoperability across multiple devices including tablets, smartphones, workstations or laptops is indeed feasible. The D3.js library that hFigures is built on, enables compatibility with Internet Explorer versions 8 and higher using a component named *Aight* [22]. Internet Explorer 8 is prevalent in hospitals and clinics due to the restrictions in installing custom software.

Patient Evolution

Multiple graphs plotted on the layout of recommended values shows the change in the data over time. This could allow clinicians to understand the evolution of certain aspects in the health of the patient. For instance, health professionals would be able to look at the effect of a trail drug treatment over time. Possibly, the collected samples of a patient before and after starting the trial would be plotted as the multiple graphs portraying the evolution of the patient. Figure 10 shows an example of a more complex dataset. The labels for each individual measurement are usually hidden when a full zoom out is performed by the user. For the purpose of demonstrating the visualization library we have made all the labels visible.

Personal Health Monitoring

As an example, Fitbit provides activity trackers and a wide-range of devices. The data collected can be obtained through their API. In most cases, providers such as Fitbit follow an HTTP REST interface. Figure 3 shows an example of the data Fitbit provides through its interface. The data has `"activities"` as properties of a JavaScript object. These entries have in turn a property `"startTime"` which provides the timestamp required for the hFigures data source file.

In this case, a step counter contains the property `"steps"` with the number of steps registered by the device. This and other measurement can be plotted in the hFigures following the data source file structure.

We can also use multiple graphs to show the user-defined goals as a reference in addition to the actual values of the measurement (steps per day). Figure 3 only show steps per day, however the documentation of the API specifies that additional data is available. This data includes distance travelled, sedentary activity, floors climbed, calories burnt and more.

Other device manufacturers such as Withings or Jawbone provide their users the possibility to use their APIs to extract collected data in a similar way. As in the case of Fitbit, all data has at least a timestamp (date, time or both) and a set of values.

In order to specify the recommended range for these measurements in the hFigures data source file, it is possible to use websites such as WebMD or Mayo Clinic to obtain their recommendation for such values.

Future Work

We plan to add features to the library and use it in future studies. Animation effects and other graphical elements will be incorporated so as to facilitate the understanding of how the measurements have changed over time. A potential addition to the library includes persuasive design elements to enhance the attitudes towards behavior change.

Conclusion

A user-centered approach was followed to develop a health data visualization library, hFigures. It is based on the design idea of the hGraph but provides additional features. The goal is to enhance users' understanding of health data. By comparing multiple graphs, users get a graphical representation of the overall evolution of health measurements with respect to the recommended targets. The label spacing algorithm reduces clutter and visually separates the data into adequate spaces. Groups are clearly divided in the visualization resulting in reduced clutter. This helps users to focus on separate regions of the dataset. The hFigures library also generates an SVG structure embedded in the HTML document. The SVG can be extracted and can be directly utilized as an image file for illustrations in scientific articles and as input for other software applications.

hFigures is an open source library that aims to improve the understanding of health data through visualization techniques. The goal is to enhance individuals to take informed decisions in order to change their health behaviors towards a healthier society.

Availability and Requirements

- **Project name:** hFigures
- **Project home page and source code repository:** https://github.com/ledancs/hFigures
- **Operating System:** Platform independent.
- **Programming language:** JavaScript.
- **Other requirements:** Developers willing to deploy the application need to serve the files via a Web server. Users require a Web browser to visualize the application.
- **License:** Text for this section.
- **Any restrictions to use by non-academics:** No.

Abbreviations

hGraph: Health graph; hFigures: Health figures; JSON: JavaScript object notation; EHR: Electronic health record; PHR: Personal health record; HTML: Hypertext markup language; SVG: Scalable vector graphics; D3: Data-driven documents; API: Application programming interface.

Competing Interests

The authors declare that they have no competing interests.

Author's Information

AL and HN are part of the Personal Health Informatics research group from the Department of Signal Processing of Tampere University of Technology. AL is a Software Engineer pursuing PhD studies in Health Data Visualization with a strong background in Web Engineering and Health Information Systems. HN is a PhD senior researcher with background in biomedical engineering, signal processing, electrical engineering, user interface design and service design.

References

1. Hood, L., Flores, M.: A personal view on systems medicine and the emergence of proactive {P4} medicine: predictive, preventive, personalized and participatory. New Biotechnology **29**(6), 613–624 (2012). doi:10.1016/j.nbt.2012.03.004. Molecular Diagnostics and Personalised Medicine
2. Lesselroth, B.J., Pieczkiewicz, D.S.: Data visualization strategies for the electronic health record. In: Berhardt, L.V. (ed.) Advances in Medicine and Biology vol. 16, pp. 107–140. Nova Science Publishers, Inc, New York (2012)
3. Baird, J.C., Noma, E.J.: Fundamentals of Scaling and Psychophysics. John Wiley & Sons Canada, Limited, New Jersey (1978)
4. Cleveland, W., McGill, R.: Graphical perception: Theory, experimentation, and application to the development of graphical methods. Journal of the American statistical association **79**(387), 531–554 (1984)
5. Follett, J., Sonin, J.: hGraph: An Open System for Visualizing Personal Health Metrics. Technical report, Involution Studios, Arlington, Massachusetts (April 2012). Accessed 25 Mar 2015
6. Ledesma, A., Nieminen, H., Valve, P., Ermes, M., Jimison, H., Pavel, M.: The shape of health: A comparison of five alternative ways of visualizing personal health and wellbeing. In: Engineering in Medicine and Biology Society (EMBC), 2015 37th Annual International Conference of the IEEE (2015). doi:10.13140/RG.2.1.1321.6725
7. Bui, A.A.T., Aberle, D.R., Kangarloo, H.: Timeline: Visualizing integrated patient records. Information Technology in Biomedicine, IEEE Transactions on **11**(4), 462–473 (2007). doi:10.1109/TITB.2006.884365
8. research2guidance: mHealth App Developer Economics 2014. Berlin (2014). research2guidance
9. Goetz, T.: It's Time to Redesign Medical Data. http://www.ted.com/talks/thomas_goetz_it_s_time_to_redesign_medical_data?language=en
10. Idris, M., Hussain, S., Ahmad, M., Lee, S.: Big data service engine (bise): Integration of big data technologies for human centric wellness data. In: Big Data and Smart Computing (BigComp), 2015 International Conference On, pp. 244–248 (2015). doi:10.1109/35021BIGCOMP.2015.7072838
11. Few, S.: Information Dashboard Design. O'Reilly, North Sebastopol, California (2006)
12. Soomlek, C., Benedicenti, L.: An agent-based wellness indicator: Experimental results and future directions. J. Inf. Technol. Res. **6**(2), 1–23 (2013). doi:10.4018/jitr.2013040101
13. de Ridder, M., Constantinescu, L., Bi, L., Jung, Y.H., Kumar, A., Kim, J., Feng, D.D., Fulham, M.: A web-based medical multimedia visualisation interface for personal health records. In: Computer-Based Medical Systems (CBMS), 2013 IEEE 26th International Symposium On, pp. 191–196 (2013). IEEE

14. Shneiderman, B., Plaisant, C., Hesse, B.W.: Improving health and healthcare with interactive visualization methods. Technical report, Citeseer (2013)
15. Bostock, M., Ogievetsky, V., Heer, J.: D^3 data-driven documents. Visualization and Computer Graphics, IEEE Transactions on **17**(12), 2301–2309 (2011)
16. Chris, N.: Toward measuring visualization insight. Computer Graphics and Applications, IEEE **26**(3), 6–9 (2006)
17. Saraiya, P., North, C., Duca, K.: An evaluation of microarray visualization tools for biological insight. In: Information Visualization, 2004. INFOVIS 2004. IEEE Symposium On, pp. 1–8 (2004). IEEE
18. Poutanen, O., Koivisto, A.-M., Kääriä, S., Salokangas, R.K.: The validity of the depression scale (deps) to assess the severity of depression in primary care patients. Family practice **27**(5), 527–534 (2010)
19. The Mozilla Developer Network (MDN): The Typeof Operator - JavaScript Reference. https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Operators/typeof
20. Nielsen, J.: Usability Engineering. Elsevier, ??? (1994)
21. Rubin, J., Chisnell, D.: Handbook of Usability Testing: How to Plan, Design and Conduct Effective Tests. John Wiley & Sons, ??? (2008)
22. Allen, S.: Aight – JavaScript Shims and Shams for Making IE8-9 Behave Reasonably

Figures

Figure 1 JSON data source file. The data source file structured as a JSON file.

Figure 2 Layout construction code. The layout for distributing the measurements is build using d3 pie layout leaving spaced between the measurement groups.

Figure 3 Fitbit API example response. An example JSON response from a Fitbit activity sensor that can be transformed to a data source to be visualized by hFigures.

Figure 4 JavaScript code for label positioning. The function in JavaScript distributed the positioning of the label to avoid overlapping and clutter.

Figure 5 JavaScript code to determine if additional ranges are provided. In JavaScript the data source could contain additional ranges, these are properties in an object that need to be checked beforehand and if the exists, compare the values accordingly.

Figure 6 An example of the hGraph. The hGraph shows an overview at a low zoom level. When the zoom increases, the details of each measurement are revealed.

Figure 7 A simple example of hFigures. The measurements are separated by groups in sectors.

Figure 8 Label positioning avoiding overlapping with measurements. Some measurements can be positioned outside of the recommended range, the labels are adjusted to avoid overlapping.

Figure 9 Label positioning distribution. The labels are positioned according the their height and margin, as a result the labels do not overlap and clutter is avoided.

Figure 10 A complex hFigures example. A holistic health assessment of a modelled person comprised of several measurements with two time snapshots showing the person's evolution.

Table 1 Comparative Functionality Table. The table lists the features supported by hFigures and hGraph.

Functionality	hFigures	hGraph
Color-coded measurements	✓	Partially
Recommended area	✓	✓
Circular distribution of measurements	✓	✓
Plot values at specific times	✓	—
Time series embedded in the data source	✓	—
Immutable shape	✓	—
Weighted score	—	✓
Multiple graphs	✓	—
Visible category labels	✓	—
Label space distribution	✓	Partially
Grouped measurements	✓	—
SVG export	✓	—

Tables