BIIGLE Tools – A Web 2.0 Approach for Visual Bioimage Database Mining

Timm Schoening, Nils Ehnert, Jörg Ontrup, Tim W. Nattkemper
Bielefeld University, Faculty of Technology, Biodataining & Applied Neuroinformatics Group
{ tschoeni | jontrup | tnattkem }@techfak.uni-bielefeld.de

Abstract

In this paper we want to discuss the usage of web 2.0 techniques to realize information visualization based exploration and annotation of huge volume, semi-structured data, and in particular high throughput bioimage series. To this end, we developed a toolbox for a graphical representation of different displays in a browser context which can be used for image database exploration in a link & brush fashion. The web based approach proved to be capable of information visualization tasks and supports collaboration of several users at arbitrary locations.

1. Introduction

During the past several years, the Internet has undergone some remarkably big steps forward. It developed from the first basic information retrievals and data sharing services to a much more convenient and wide-spread institution for the daily use. For some years now, this process is described by the buzz word Web 2.0.

One basic feature of the new Internet is collaboration of many users [1]. Almost all of the information provided on the Internet nowadays is created by users and not operators of web pages. Hence another key phrase occurred: the *"User Generated Content"* (UGC).

The second key feature of the Web 2.0 is the look and feel of the created applications. During the last decade internet browser and programming languages evolved, to be able to create and show multimedia web pages which respond and look like normal desktop software. These applications are called: *"Rich Internet Applications"* (RIAs).

Following the first experimental business and social applications, the domain of Science has entered the Web 2.0., described by the keyword "Science 2.0" [2,3].

Based on this development we wanted to figure out the power and usefulness of this new environment for the processing of high throughput image series of underwater habitats. The first domain of our web based approach was the examination of coral growth in the North Sea and North Atlantic.

The first step of this examination was to create a Web 2.0 interface to allow online browsing of the captured images, combined with an annotation tool to specify the species found in this habitat ("Biigle").

This application was realized using Adobe Flex [4] which is a software development kit for the creation of RIAs.

Following this successful and encouraging achievement, the idea arose, to develop some tools for the exploration of the created annotation data.

This data is stored in a database. To support users in gaining a mental model of the structure hidden in this data and to simplify further investigations, these tools should use Link & Brush techniques to allow easy interaction of users with data ("Biigle Tools").

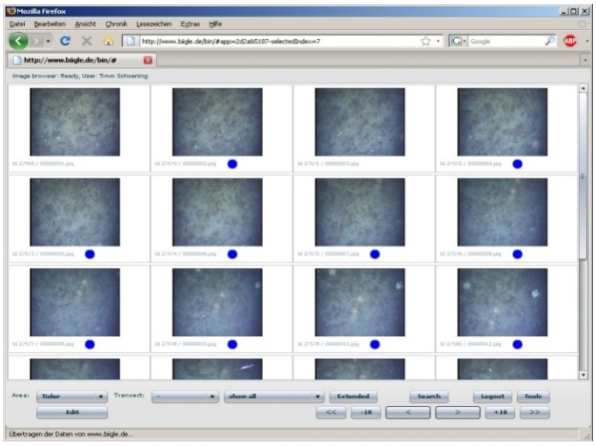

Figure 1: Biigle in a Firefox browser window. You can see the thumbnail matrix as well as different dropdown menus and buttons to limit the fundamental pool of data.

2. Biigle

Biigle is an acronym for *"BioImage Indexing and Graphical Labeling Environment"*. The current data basis is a pool of more than 30.000 underwater images of the seafloor at cold water coral reefs. The images show an amazing variety of life in this biosphere. But by now Biigle is also used for other tasks as well and is currently used e.g. for microscopy images.

The main component is the *"Biigle browser"* to select image groups/series of interest. Images can be selected by area, transect and added label types. The resulting selection is displayed in a thumbnail matrix (Figure 1).

By selecting a single item from the matrix, the user gets a full screen view of the image. An additional tool box used for adding labels to the image is displayed.

To label a region in an image, the user selects a region contour type (e.g. rectangle, circle) and a biological label. Labels set by other users in the same image can be displayed as well.

These labels are the UGC created by this RIA. All information extracted by humans from the previously unstructured images is then stored in an online database. The combination of the pool of images and this database results in a heap of then semi-structured data. This data can be used to train learning algorithms to perform the labeling task faster and without effects of fatigue.

3. Motivation

The Biigle database currently contains more than 30.000 images connected to almost 10.000 labels of around 100 types. This roughly corresponds to more than 2000 labeled images with an average amount of 5 labels per image, since ca. 8000 images contain no labeled region yet. By examination of random images it is apparent that neither have all labels been found nor do the unlabeled images contain no objects.

Hence two tasks have to follow the initial efforts. First, the labeling has to be continued until as much labels as possible are added to the images. Second, the semi-structured data (digital images plus semantic annotations) have to be explored.

To find relationships between different species apparent on the images, being a main research interest in this project, the database has to be filtered and dynamic queries according to a specific task have to be performed. To information technology specialists who are familiar with databases, such queries are not an obstructive task. The users of the Biigle environment though, are professionals from other disciplines and most times do not have the knowledge to perform this task. Hence it is crucial for successful progress to create a transparent view on the data and enable users to find connections via a graphical representation of information.

To increase the ease and usability, the question arose, whether these visualizations could be implemented using Flex as well. This would allow integrating them directly into Biigle.

4. Biigle Tools

The main task was to develop a tool box to interactively create graphical displays, called *"Biigle Tools"*. In difference to other frameworks for visualization of data in a web context, i.e. Flare [5], which uses Flex too, our plan was to create tools, which allowed interaction with the displayed data. The main goal of these tools was a graphical representation of the data stored in the database to prevent the user from actually working with a database query language.

To help and guide the finding of relations, these graphical displays have to follow basic principles of information visualization. Each tool was designed to be operated via mouse input and stands for an abstraction step to keep the user away from the basic data.

The tools should be designed to be able to deal with different stages of dimensionality and a lot of data items.

All tools had to be integrated into the Biigle browser to allow live limitation of the displayed data. The user should be able to browse all images stored in the database or limit the data by selecting a requested area, transect or label status.

Biigle Tools use dynamic visualization by enabling the user to manipulate the style of the display [6]. Interactive visualization is made possible by allowing the user to select and change the pool of input data for a display [6]. Based on the change of the data or the display, the user can operate with the new screen content in a feedback loop.

One reason to choose Flex was its ability to render any desired graphical object as well as being able to handle several different types of user input. This allowed the implementation of tools using link & brush functionality which is the fundamental analysis principle used in Biigle tools. None of the displays is static and interaction with the data and display is the goal of each of them.

The data to be analyzed are quantities of regions, labeled by a particular class, per image or in a set of images.

If only one biological class, i.e. one parameter is considered, two straightforward ways to present the data would be a Tukey box plot or a histogram [7,8]. Because of the focus on relation finding tasks, one-dimensional displays are just necessary to get a first impression on single variables. Based on a histogram

view, and the graphical component selected, the ordinal data can be binned into intervals, chosen by the user.

To find correlations between two or more parameters several other graphical displays were implemented: scatter plot, table lens, parallel coordinates, generalized drafters plot and netmap display. Some of these will be explained in detail later.

5. Curse of dimensionality (and quantity)

Curse of dimensionality is a term from the field of machine learning and usually describes the problem to characterize a structure in a high-dimensional feature space by a low number of data points [9].

The Biigle tools do not deal with machine learning, but during the development of the displays we experienced another Curse of dimensionality (and quantity). The Adobe Flex Environment was designed to create RIAs that should feel like desktop applications. But even desktop applications need some processing time in tasks including 30.000 items and several parameters. The browser based approach does not support a fast calculation and thus the tools sometimes need a bit of time to perform the requested job. This occurs mainly in displays for a large number of parameters d and images n. The size of d and n determines the number of graphical objects in a display. To create a graphical representation instantly after changing the setting of the chosen visual component, d should be limited to $d_{max} = 3$ and the pool of images should not contain more than $n_{max} = 1.000$ images.

6. Examples

6.1 Generalized Drafters Plot (GDP)

A matrix of scatter plots is called generalized drafters plot [6,7]. Each position in this matrix shows a single scatter plot of two parameters. Since the matrix is symmetric it can be limited to the upper triangle (Figure 2 (A)).

Each position on a single chart describes the correlation of one parameter with another. We implemented a GDP that displays the amount of images which contain this combination of parameters by dots on the chart. The encoding of image amounts can be set to (dot-) color or radius.

To select dots in a GDP, an active chart has to be specified first. This is realized by a single click on a chart. To select items on a chart, three selection modes are provided, which can differ from chart to chart. One can use a rectangle while on another chart a polygon is used and a third one uses a circle. Images are set active if they are selected on every chart. Images on charts without a selection are handled as active.

If the encoding is set to radius, active items are highlighted. This makes it easy to see how many images lie on a position and how many of them are currently active. The total amount of active images is shown, and these active images can be displayed in the Biigle browser.

The generalized drafters plot becomes huge for lots of parameters. It is almost impossible to display charts for more than four parameters on current desktop screens when using them for link & brush tasks. The usage of color codes for image amount allows smaller charts so it is possible to display up to seven parameters. The major drawback is that a lot of screen is wasted, where no charts are displayed.

6.2 Parallel Coordinates

In parallel coordinates all parameters are normalized to fit a scaling bar of a given size which is the same for each parameter. The parameter vectors are displayed as compound lines from the parallel coordinate on the left to the one on the right [6,7,10].

The position of an intersection of a parallel coordinate with a parameter line is defined by the parameter value encoded as height.

A static parallel coordinates plot is useless in most cases. Dealing with lots of parameter vectors, the plot consists of a packed clutter of lines forming an inscrutable cobweb.

Usually several lines intersect on a data point on a parallel coordinate. A bend occurs to draw the subsequent line to the next coordinate. Because of the resulting discontinuities it is impossible to follow a single line. This could be avoided by drawing the parameter vectors not as linear connections between data points on the coordinates but as curves [6]. A hindrance for this is the performance. Another solution is to assign a color code to the lines. On the other hand huge amounts of parameter vectors will lead to such a high number of color scaling steps, making it impossible to interpret the scale.

The parallel coordinates plot evolves to a mighty display if highlighting is implemented to filter the data and to integrate it into a link & brush context. Therefore we extended the static plot and replaced each parallel coordinate with a vertical slider containing two thumbs to define an active interval. Only parameters, whose values all lie within the intervals, are drawn. All other parameters are either hidden or blurred (Figure 2 (B)).

The thumbs positions can easily be modified by mouse drag and the display instantly redraws the cobweb according to the new set of intervals.

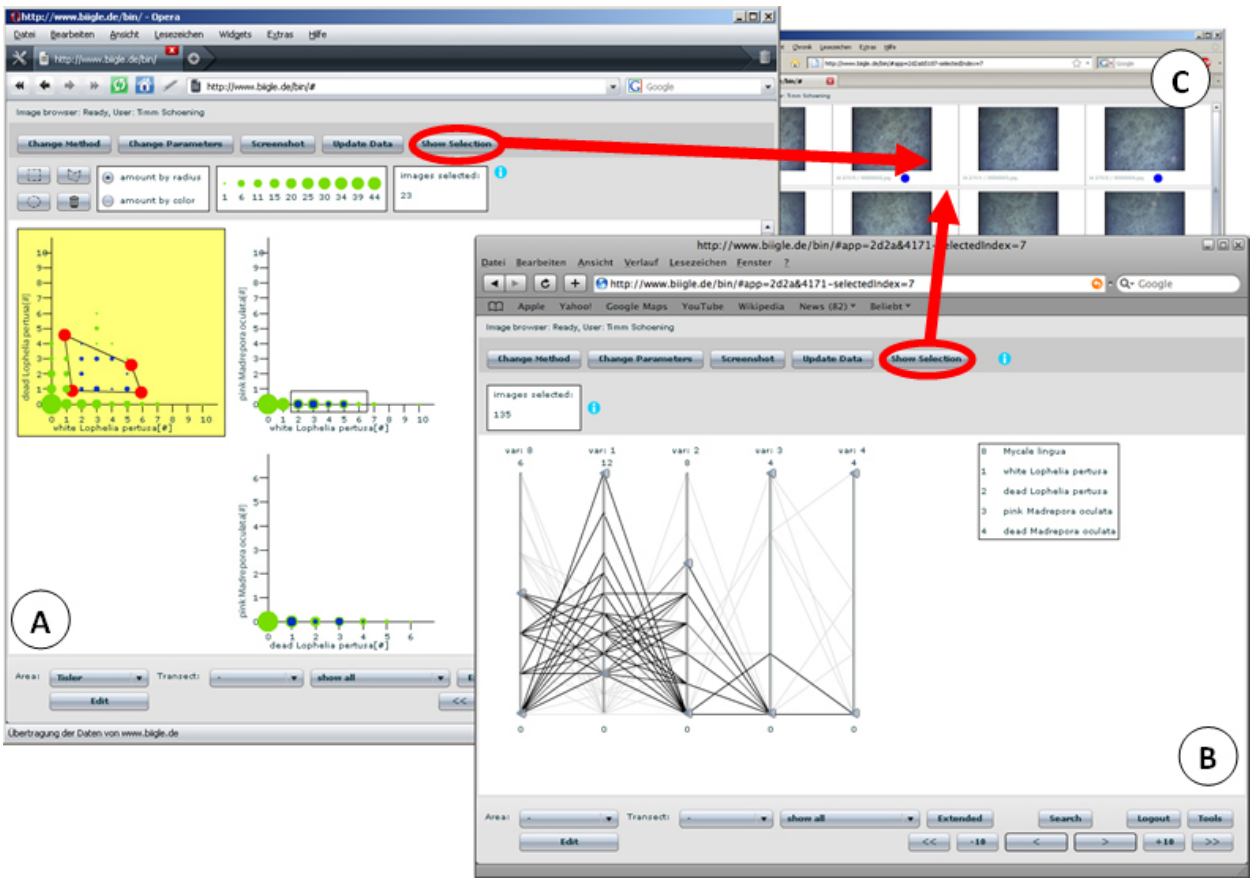

Figure 2: (A) The generalized drafters plot. You can see three charts that form the upper triangle of a matrix. Each chart shows the connection of two parameters. The encoding is set to radius. On some charts active items have been specified either by a polygon or rectangular selection tool. Items that are active on all screens are highlighted in blue. (B) The parallel coordinates plot. Each vertical coordinate encodes one parameter. Two thumbs on each coordinate can be used to define an active interval. Lines between coordinates stand for images, where black lines are active and grey lines inactive. The highlighted "Show Selection" button returns you to the Biigle thumbnail matrix (C or Figure 1). The shown images are those that match all selections made in the Biigle tools.

6.3 Netmap Display

All graphical tools described so far either cannot display data of any given dimensionality (i.e. histogram, scatter plot) or do need increasing screen size to do so. The size of a table lens or parallel coordinates plot increases linear with dimensionality, the size of a generalized drafters plot even squared.

The netmap display solves both problems as its size is always the same for data of any dimensionality. The data is therefore displayed on a circles rim. The circle is divided into regions for different parameters (i.e. species) which are encoded by color. All regions are subdivided into bins of growing frequencies (Figure 3 B). Each bin B_i can be described as $B_i = \{\Omega_i, l_i\}$ where Ω_i is a parameter and l_i is the frequency of labeled regions in the preselected image set.

A single object (i.e. an image) is displayed by a cobweb between these bins. A connection from bin B_1 to another bin B_2 is drawn if an object contains l_1 labels of Ω_1 and l_2 labels of Ω_2 (Figure 3 B). All objects together form an inscrutable huddle within the circle. Single objects cannot be traced throughout the plot anymore. Our Biigle tool version of a netmap can handle this problem in two ways. It is possible to show just the surrounding polygon of each object which results in a decreased amount of connections in the circle (Figure 3 C). Anyway the netmap display still remains confusing for high amounts of objects.

The second option is to encode the thickness of each connection of two bins by the amount of all objects that feature the parameters of those bins. Important connections which are frequent are instantly visible (Figure 3 A).

All characteristics described so far just deal with the static display, but we used the netmap as a link & brush component as well.

The first option to interact with the display is to modify the fundamental bins. Each parameter region features a button to show a histogram of the underlying distribution of this parameter. The amount of bins can be specified and the size of each bin can be modified (Figure 3 E). In the beginning there is a bin for any existent label amount l_i. The user can pool these bins to groups which can contain any combination of connected bins, i.e.:

$$L_1 = \{l_1,\ldots,l_{k-1}\}\ L_2 = \{l_k\}\ L_3 = \{l_{k+1},\ldots,l_n\}$$

The look of the netmap display does not change whether its bins encode single label amounts or pools of these. A reduced amount of bins can result in a better overview of the data.

The second and more important option to interact with the netmap display is to enable and disable single bins by mouse click. Connections from a disabled bin are blurred and all objects which contain this (pool of) label amount(s) are set inactive. This results in narrower connections between other bins, as all connections for an inactive object are not taken into account for the calculation of the connections thickness.

7. Results

None of the implemented components can be said to be the "right one" for image database exploration. To gain a first overview, only a histogram is useful. None of the other displays can show this information in an easily understandable way.

To find correlations between two parameters a single scatter plot has proven to be the best option. The distribution of data items is instantly perceived [6]. It is understandable without a lot of training.

The GDP can be useful for three-dimensional data. But on the whole, none of the components for high-dimensional data is as useful and powerful as the netmap display.

All components act as a first proof of concept for their utility to the Biigle context. None of them can be said to be in a final state. Each of them should undergo further development.

Generally just a little part of information visualization techniques has been used to create the tools. To deskill the handling, principles like Gibson's affordance should be applied [6].

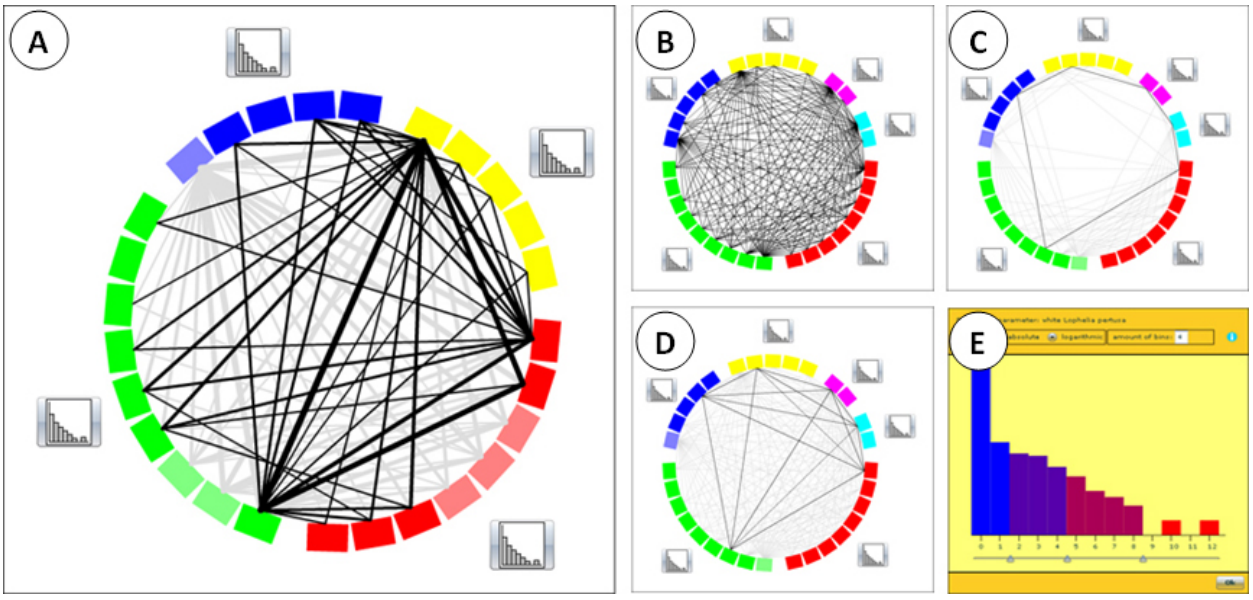

Figure 3: The netmap display. (A) shows a Netmap display which encodes object amounts by connection thickness. Each parameter's region is encoded by color. Note the disabled bins and the blurred connections from and to these bins as well as the histogram buttons adjacent to each parameter region. (B) shows a netmap display without connection encoding. All connections are drawn and form an inscrutable huddle. (C) and (D) show a netmap display with only one active object and the object encoding set to polygon in (C) and set to normal mode with all connections drawn in (D). (E) shows a histogram of a given parameter. Note that there are fewer bins (e.g. 4) than bars (e.g. 13, where the height of bars 9 and 11 is zero).

8. Conclusion

The development of Biigle Tools has proven the context of Web 2.0 being capable of information visualization tasks. The combination of data annotation and exploration in the same environment is possible and makes sense as it describes two steps of the same task. Based on the insights from the exploration of the semi-structured data, correlations might be found, which can lead to improvements of the structuring itself. This creates another large scale iteration loop with the professional user at its center.

Due to the flexibility and Web 2.0 approach, other image pools can be integrated. It is also possible to use the technique for completely other image and label based research projects.

We gained insight to the possibilities as well as the hindrances of information visualization within the Web 2.0. Generally any visualization can be thought to be transformed to a web application. The software development with Flex is oriented to create graphical displays and allow easy interaction with them and is therefore outstandingly suitable in this task. One downside, as described in chapter 5 is a limitation in dimensionality and quantity. This problem can be solved by using different graphical displays or outsourcing calculation intense tasks to a powerful server. The RIA will then just be used to display the results of the calculations.

The main upside of the Web 2.0 approach is the flexibility in the usage of these applications. It is possible to collaborate with colleagues anywhere in the world. It is unnecessary to maintain the software on the user side. The current release will always be used when surfing to the application site.

This makes it easy for software developers as well because they don't need to modify their application for any operating system and users are kept away from annoying version update/cross platform issues (Apple, Linux, Windows, etc.).

To work with RIAs designed with Flex just a web browser with a Flash plug-in, which is installed on most computers anyway, and an internet connection is needed (note that there are display results for three most popular browsers in Figures 1 and 2). That means research can be done being in a bureau at work or in the kitchen at home, even in any coffee store or in a deckchair on the beach (assuming there is WLAN).

9. Acknowledgements

This work was part of the StatoilHydro-funded research programme CORAMM (Coral Risk Assessment, Monitoringand Modelling) co-ordinated by Laurenz Thomsen (Jacobs University, Bremen). Special thanks go to Melanie Bergmann and Autun Purser for outstanding efforts in data labeling and consultation. J.H.Spencer, A. Larsson, L. Jonsson and M. Roberts kindly labeled fauna in BIIGLE. Data was provided by the Alfred-Wegener-Institut für Polar- und Meeresforschung (Bremerhaven, Germany) and by Tomas Lundälv from the Tjarnö Marine Biology Laboratory (Sweden).

10. References

[1] **Röttgers, Janko.** Am Ende der Flegeljahre. *c't*. 2007, 25.

[2] **Shneiderman, Ben.** Science 2.0. *Science*. 2008, p. 1349f.

[3] **Waldrop, Mitchell.** Science 2.0 – Is Open Access Science the Future? *www.sciam.com* [Online] 2008 http://www.sciam.com/article.cfm?id=science-2-point-0.

[4] **Adobe – Flex 3.** [Online] http://www.adobe.com/products/flex

[5] Flare [Online] http://flare.prefuse.org/

[6] **Ware, Colin.** *Information Visualization – Perception for Design.* San Francisco : Elsevier, 2004. ISBN 978-1-44860-819-1.

[7] **Spence, Robert.** *Information Visualization – Design for Interaction.* Harlow : Pearson Education Limited, 2007. ISBN 978-0-132-0655-04

[8] **Tufte, Edward.** *The Visual Display of Quantitative Information.* Cheshire : Graphic Press LLC, 2001. ISBN 978-0961392147.

[9] **Wikipedia curse of dimensionality [Online] http:**//en.wikipedia.org/wiki/Curse_of_dimensionality.

[10] **Inselberg, A.** Parallel coordinates: a tool for visualizing multi-dimensional geometry http://www.ifs.tuwien.ac.at/~mlanzenberger/informatica-feminale03/se188095/auth/00146402.pdf