Designing Visual Analytics Systems for Disease Spread and Evolution

VAST 2010 Mini Challenge 2 and 3 Award: Good overall design and analysis

Jo Wood* Aidan Slingsby† Jason Dykes‡

giCentre (http://gicentre.org), School of Informatics
City University London

1 Introduction

Using two of the VAST 2010 mini challenges as a case study, we report on the design decisions and software development process used to create visual analytics software for understanding disease spread and mutation. The software we developed and the analysis conducted attempted to help us understand (a) how a fictitious disease may have spread between selected cities around the globe; and (b) how genetic sequences taken from infected patients may be used to chart the evolution of the disease and changes in its severity, drug resistance and other characteristics.

Underlying our approach to the visual analytics process is the idea that *tool development and visual analysis using the tool are not separate processes*. As such we do not use off-the-shelf, high level visual analytics packages, but instead the lower level programming environment *Processing* [4] – a graphically oriented set of Java libraries suitable for data visualization [3]. As well as providing the necessary graphics and interaction facilities, Processing is engineered to ease rapid prototyping and can therefore be used for iterative tool development - data analysis tasks.

2 Design for Identifying Pandemic Disease Spread

Challenge MC2 required participants to process nearly 1Gb of text files of hospital admission data. Data related to admissions in eleven cities around the world over a ten week period and included reported symptoms and demographic details of each person admitted. The challenge involved characterising a disease and its geographic spread over time. Our approach to building software to meet this challenge was guided by a number of design principles:

- use multiply-coordinated views to afford comparisons between spatial, temporal and thematic patterns;
- use layout to maximise the information content of the representations and afford comparison between them;
- allow filtering by day, place and admission summary to enable 'detail on demand';
- pre-compute any measures necessary for rapid interaction;
- maximise the data/ink ratio of all visual representation;

Our first step was to produce a rapid visualization of the data in order to get a sense of its volume and diversity of symptoms that form the basis for analysis. We did this by using Processing to produce tag clouds of symptom terms, alphabetically ordered and sized by frequency. This revealed that some data cleaning was necessary to remove errors in the data, normalization of synonyms and stemming of verbs and adjectives. Data cleaning was achieved by iteratively writing cleaning code in Processing and visualizing the results, modifying the code and repeating. Once clean we were able to hypothesise disease symptoms by calculating the Chi-expectation value [5, 6] for each term by comparing observed fatal symptoms with all symptoms for each hospital. Results were visualised as a colored tag cloud. Storing the results of analytical processing in a database allowed more rapid prototyping and interaction during subsequent iterations.

Our next iteration of the program design/analysis involved exploring the geographic patterns of the disease. Given its global extent, our initial design involved mapping the measures of the disease (numbers of fatal admissions, numbers of patients with diagnostic symptoms etc.) to a global map base (see Figure 1 top-left). Rapid development (about an hour's programming) was enabled by using mapping and interaction libraries we had previously developed for other applications [7]. On visualising the results it was apparent that the geography of the distribution was not as rich as its temporal variation and that some some form of visual 'temporal signature' would be useful in detecting change in the pattern of admissions over time. This led to the construction of histograms of various disease measures (vertical axis) over time (horizontal axis). Aligning them vertically for each of the eleven cities enabled geographic comparisons of temporal patterns (see Figure 1 right).

The challenge required us to assess the possible relationships of symptoms with gender and age. We therefore created an expanded histogram view on which gender and age profiles could be displayed (Figure 1 bottom-left). To compare with disease symptoms, the final view we constructed provided a list of symptom terms that could be ordered by various criteria, such as frequency and TF-IDF. (Figure 1 center). All of the views were linked to allow temporal and spatial filtering ("what happened around the world on this day?"; "when was the onset and retreat of the disease in this city?").

3 Design for Identifying Genetic Mutations

Challenge MC3 required us to identify genetic sequences that corresponded to particular disease strains and the mutations that were associated with changes in their severity. The numbers of sequences involved and their length required less filtering than MC2 in order to visualise the data. As such the design of software was guided by use of alignment and sorting to explore the data. The four genetic bases were symbolised with categorical colors from ColorBrewer [1] using previously written libraries [7]. They were sufficiently short to be displayed horizontally with zooming and panning available for detail on demand. Initial visual exploration revealed that all mutations were substitutions, so aligning all bases by their position in the sequence was possible (if there had been insertions or deletions, bases would have to be aligned using common base-patterns instead). It was also clear that the majority of DNA bases were common to the set of sequences, so these could be removed from view enabling focus on the mutations between strains (see Figure 2).

Two of the challenge questions required sequences to be ranked in order to identify the most likely disease strains that satisfied particular criteria. We reflected this in the design of the interaction by allowing graphical sorting of sequences based on criteria selected by the user. This allowed common mutations to be juxtaposed in such a way to allow comparison.

*e-mail: jwo@soi.city.ac.uk
†e-mail: a.slingsby@soi.city.ac.uk
‡e-mail: jad7@soi.city.ac.uk

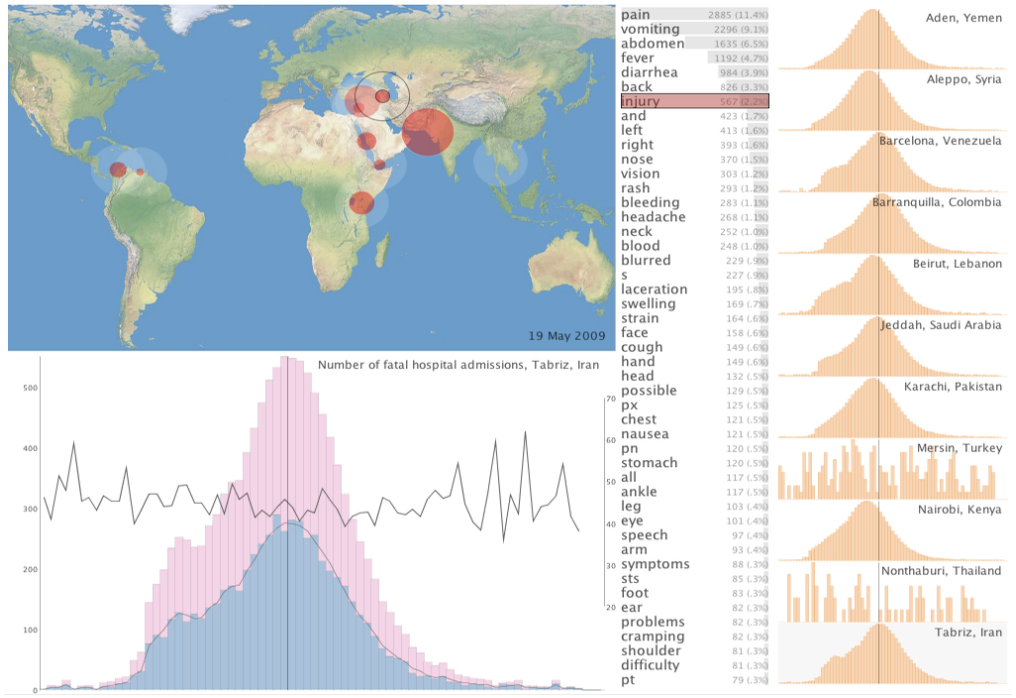

Figure 1: Final layout of PandemView showing spatial view (top left), thematic view (center), temporal views by city (right) with grey vertical time-line centred on 19th May and enlarged temporal view with gender and age profiles (bottom left)

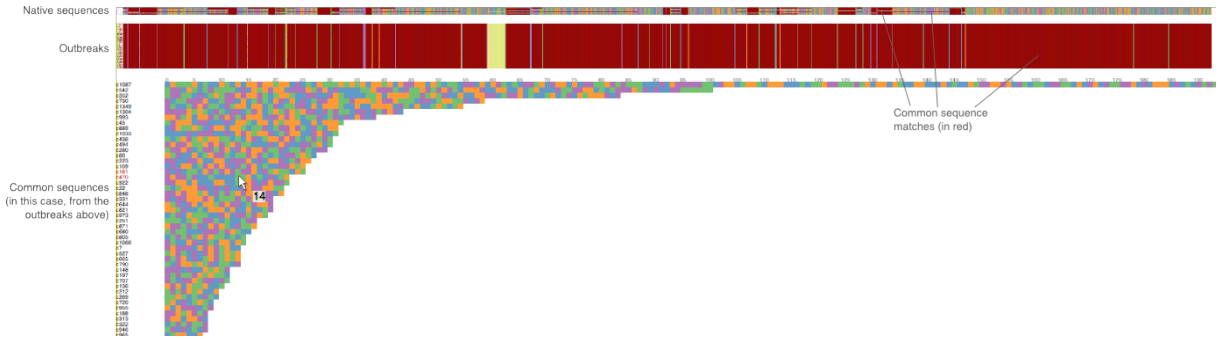

Figure 2: Final layout of SequenceView showing native sequences (top), outbreaks (middle) and longest common sequences found in the outbreaks (bottom). Common sequences are identified in red and those over which the mouse is positioned are colored yellow.

4 Conclusions

We created two bespoke applications suited to the nature of the visual analytics task at hand. Assembling and analysing the data, writing and modifying the applications each took about two person-days, which we regard as being comparable to the amount of time required using an off-the-shelf visual analytics toolkit. Unlike the use of an existing software 'tool', a lower-level programming approach affords greater flexibly in the interaction between analyst and software [2]. Importantly, it blends the distinction between software development and data analysis, allowing discoveries made about the data to influence the subsequent design of the visual tools used to explore the data. The cost in this approach to visual analytics is the skillset required to develop the applications. However with the availability of increasingly usable programming environments such as Processing and data visualization libraries, this is an approach available to an increasing number of analysts.

For detailed answers to the challenge questions and interpretation of the data see `http://gicentre.org/vast2010`.

References

[1] C. Brewer and M. Harrower. Colorbrewer: Color advice for maps. http://colorbrewer2.org, 2009.

[2] J. Dykes. Facilitating interaction for geovisualization. In J. Dykes, A. MacEachren, and M. Kraak, editors, *Exploring Geovisualization*, pages 265–291. Elsevier, Amsterdam, 2005.

[3] B. Fry. *Visualizing Data*. O'Reilly, Cambridge, 2007.

[4] B. Fry and C. Reas. Processing. http://processing.org, 2010.

[5] C. R. Unit. *People in Britain: a Census Atlas*. HMSO, London, 1980.

[6] J. Wood, J. Dykes, A. Slingsby, and K. Clarke. Interactive visual exploration of a large spatio-temporal dataset: Reflections on a geovisualization mashup. *Transactions on Visualization and Computer Graphics*, 13(6):1176–1183, 2007.

[7] J. Wood, A. Slingsby, and J. Dykes. giCentre utilities. http://gicentre.org/utils, 2010.