Visualizing Gene Co-Expression as Google Maps

No Author Given

No Institute Given

Abstract. We visualize gene co-regulation patterns by creating 2D embeddings from microarray data corresponding to complete gene sets from the mouse genome, across large numbers of cell types. We use google maps and client-side graphics to dissiminate pre-rendered such visualizations with a small but intuitive set of interactions. We conduct an anecdotal evaluation with domain specialists and demonstrate that biologists appreciate this approach because it facilitates low-overhead access to readily analyzable perspectives of unfamiliar datasets and because it offers a convenient way of disseminating large datasets in visual form.

1 Introduction

Visualization of biological data ranges from websites that provide sparse, keyhole representations of database stored data, to complex stand-alone visualization systems with many options and analysis features. Both approaches have merit and are widely used, yet both have task-specific limitations. In terms of usability, the former have low visual expressivity, avoid complex computations and do not show large data volumes at once, while the latter have significant overhead associated with setting up and learning to control the environments. A drawback shared by both approaches is in disseminating data and results: biologists producing data lack the expertise required to set up and maintain a database-driven website; publishing raw data requires interested researchers to learn and operate specialized analysis software, increasing the overall invested effort.

In this context we introduce gene co-expression maps which are pre-rendered 2D embeddings of genomic multidimensional data. We show complete gene sets of around twenty-three thousand genes with expression measurements across tens of cell types (e.g. all members of a cell family). These images are served through the Google Maps API, and have a simple and intuitive set of interactions that can be learned with minimal overhead. An example of such a visualization is shown in Figure 1. We run an anecdotal user study and show that this approach is a viable solution for understanding genomic co-regulation patterns especially in large unfamiliar datasets, as well as for disseminating large micro-array datasets by data-intensive labs.

The motivation behind our work is to let labs publish data and results in visual form along with raw textual data so that users can access readily analyzable perspectives on the data without additional overhead. Specifically, we are involved in the PROJECT (removed for anonymity, referred to as PROJECT

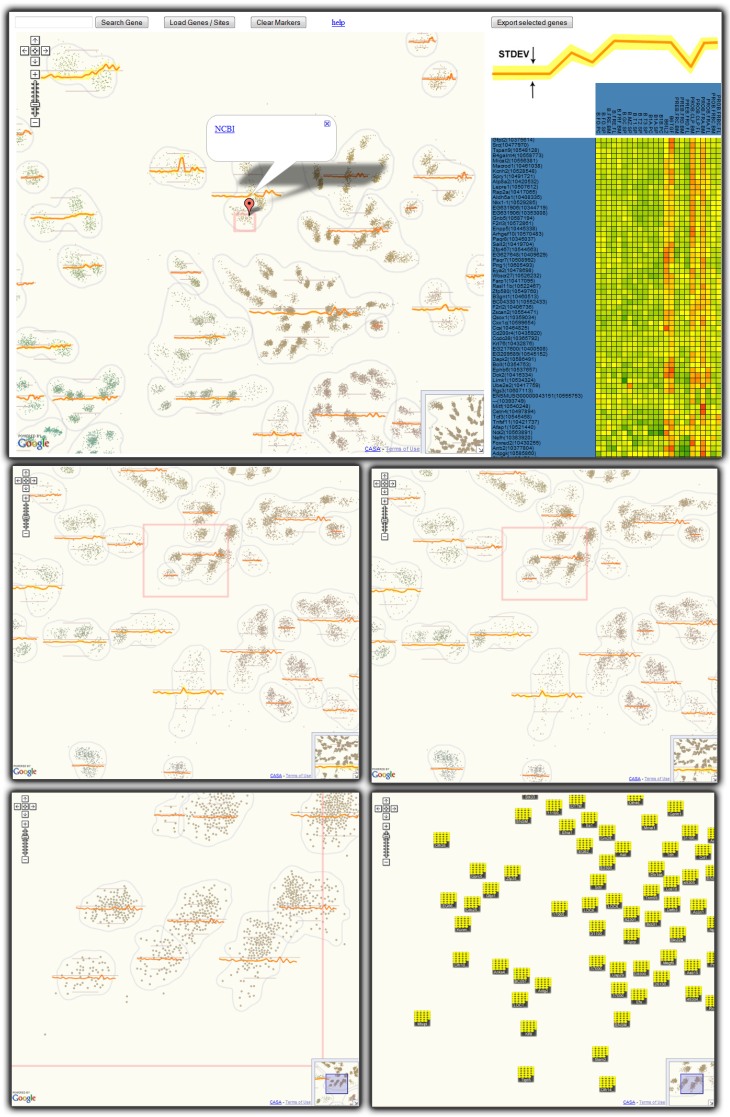

Fig. 1. Co-expression map of 23k genes over 24 cell types of the B-cell family exemplifies map concept. The top view illustrates how maps are combined with client-side graphics: the map is at the center of the display while selecting genes by drawing an enclosing rectangle generates a heatmap on the right. Maps have multiple levels of zooming (bottom 2 rows), each with a potentially different representation. For example, genes are drawn as heatmap glyphs at the high zoom (lower right), and as dots at low zoom. Expression profiles of collocated genes are aggregated and displayed as yellow glyphs over the map. As zoom increases, expression profiles are computed for increasingly smaller regions. Interactions are not limited to zooming and panning; pop-up boxes link out to extra data sources, and selections of genes bring up a heat map (top panel)

throughout the paper), a collaborative effort aimed at generating a complete microarray dissection of gene expression in the immunological system of the mouse. The data-map concept lets us disseminate the project's microarray data as precomputed visualizations that can be accessed on the project website.

The key differences between traditional approaches and our maps are as follows. Instead of the data-query-specification/recompute paradigm, co-regulation maps contain all accessible data, with data query and specification being done through zooming and panning during visualization. Traditionally, it is the end user's job to construct a visualization (query specification and parameter definition), while our visualizations are built by bioinformatics staff in larger labs. Finally, the goal of complex visualization systems is to give users complex functionality that answers a large array of questions. Co-regulation maps, on the other hand, aim to provide fast intuitive access to visual data; their functionality is therefore balanced with a sparse set of interactions, close to what is available in regular Google Maps. For users, including scientists browsing and analyzing data as well as those producing data, visualizations become easy to access, learning time is significantly reduced, users worry only about the data, and disseminating visual results is simplified.

Contributions We introduce co-regulation maps served through the Google Maps API for visually disseminating large microarray datasets. We present design elements, challenges and opportunities that became apparent in our project, and an evaluation demonstrating the usefulness of the approach as well as suggesting design guidelines.

2 Related Work

Many advanced systems for biological data analysis have been developed over the past decade. Examples targeting microarray expression data include free software packages such as Clusterview [1], TimeSearcher [2], and Hierarchical Clustering Explorer (HCE) [3] or commercial systems such as Spotfire [4] and GeneSpring [5]. GenePattern [6] is a broad effort aiming to facilitate the integration of heterogeneous modules and data into a unitary, web-managed framework for microarray data analysis. Our goal is to offer no-overhead visualizations that will be used primarily for casual data exploration by users unable to spend time learning advanced systems. In that regard, our work comes closer to applications providing primarily look-up functionality such as tools provided on the NCBI website or the genome browser at USCS [7]. In contrast to these efforts, we aim to provide visualizations that include more computation and visual cues and less complicated query specifications.

In terms of web-accessible visualization ManyEyes [8, 9] paved the way for everyday data visualization and demonstrates the usefulness of the web as a dissemination and collaboration medium. Unfortunately, while web-development toolkits such as Protovis [10] greatly aid web development, large scale web visualization is hampered by inherent browser capabilities [11]. Alternatively, stand-alone systems have been made available as applets or to be run as client

applications directly from websites [6, 12]. However, users still have to control the parameters involved in producing visualizations, specify their data queries and learn system features. This often constitutes an undesired overhead. Yet another approach, more similar to our work from an implementation standpoint, is to use Ajax (asynchronous JavaScript and XML) technology to do the rendering on the server side and serve images asynchronously to the client browser. A specific call for Ajax-based application in bioinformatics is made in [13] and [14] and [15] exemplify this approach. There is however only one essential element that differentiates this approach from traditional offline visualization systems: control and display happens in a separate place from rendering and computation. Our research differs by attempting to limit regular users' effort in creating visualizations and assigning this task to experienced personnel, by introducing visualizations that contain most of the data associated with a problem, and by using the Google Maps API, a readily available Ajax implementation of prerendered images. Closest to our work are X:MAP [16] and Genome Projector [17] which present implementations of genome browsers using the Google Maps API. We extend this idea to 2D embeddings and provide an evaluation that suggests design guidelines.

In our work we use multidimensional scaling (MDS), the process by which multi-dimensional data points are projected in a space with lower dimensionality. We use MDS to represent gene expression similarity over multiple biological conditions. Keim [18] provides a good overview of multidimensional visualization. Non-linear MDS methods, as in our work, use the similarity distance between data points to define an error measure that quantifies the amount of distance information lost during the embedding. Gradient descent or force simulation is then used to position the points in the low-dimensional space so as to minimize the error measure. A good example of such an approach is force directed placement (FDP) [19] which simulates a system of masses connected by springs of lengths equal to the distances to be embedded. Because an iteration of the original FDP model is $O(n^3)$, acceleration techniques have been proposed [20–22]. We use the last approach, an algorithm with linear iteration time proposed by Chalmers. Finally, relevant to our work is HiPP [23], an algorithm using a hierarchical clustering to drive a 2D embedding. In our work we use a combination of the original FDP, Chalmlers' acceleration technique and HiPP.

3 Methods

Given genes with expression measurements over multiple biological conditions, we construct a 2D map where genes are placed so that their proximity is proportional to the similarity of their expression profiles. Scientists can use the B-cell co-regulation map in Figure 1 to find other genes that co-regulate with genes of interest and to understand how their genes of interest co-regulate given the set of conditions described by the map. Immunologists can browse co-regulation maps to understand expression patterns in the featured conditions. Finally, scientists

interested in downloading unfamiliar data can perform a preliminary investigation using maps hosted on the project website.

Our embedding algorithm was inspired by HiPP [23] but employs a different layout technique. As in HiPP, we use bisecting k-means to create a hierarchical clustering of the data. We then compute the clustering distance of two genes as the length of the path between their nodes in the clustering tree. We multiply this distance by the Euclidian distance between genes in the high-dimensional space described by the biological conditions. Finally, we use Chalmer's embedding [22] to project this combined distance in 2D. The discrete component introduced by the clustering tree is responsible for the clear demarcations between clusters observable in Figure 1. We initially used a standard projection but user feedback indicated that the lack of visible clusters detracted from analysis. Users considered the modified version preferable even when made aware that cluster boundaries were introduced artificially.

In rendering, glyphs are drawn over map regions, showing the aggregated expression profile of genes in that particular region along with the standard deviation. The size of aggregated regions is zoom-dependent; as zoom level increases averaging are performed over smaller sets for increased averaging accuracy. This is achieved by linking zoom to cluster-cutting of a hierarchical clustering of 2D projected distances. For obtained clusters an iso-contour is drawn around the members of the cluster by using the method in [24] to achieve a rough enclosing curve and then refining it using active contours [25].

In low-level zooms, genes are represented by heatmap glyphs that color-code the expression value of that gene at each condition, giving users access to individual data values. The color scheme chosen was blue-green-yellow-red to maximize the perceived expression difference, following our users' request. To minimize the chance of overlapping gene-glyphs we apply a repulsive force between nodes at the end of the embedding stage. The force decays exponentially with inter-node distance and only affects the layout locally.

We use the Google Maps API, an Ajax framework used to render large maps, to display our visualizations. It receives as input image data in the form of a set of small images, called tiles, that when assembled together form the different zoom levels of the map. Each zoom level consists of a rectangular grid of tiles of size $2^{zoom} X 2^{zoom}$. The API decodes the zoom level and coordinates of the currently viewed map region to retrieve and display the visible tiles. The developer can load a custom set of tiles in the API by implementing a callback function that translates numerical tile coordinates and zoom level into unique paths to the custom tiles. The API provides basic functionally such as zooming and panning and allows programmatic extension or customization with markers and polyline overlays, information pop-ups and event management. The API can be easily integrated into any Javascript-powered web-page.

Our 2D embeddings are rendered to tiles, gene positions are exported to a text file, and gene expressions are coded as one-byte values to limit size and exported to a text file. These elements are used in the Javascript + Google Maps + Protovis map implementation in Figure 1. Users can search for a single gene

and highlight it via a marker. Alternatively, an entire set of genes can be loaded as well through copy and paste. Genes can also be selected directly on the map by drawing a selection rectangle. If the selection is small enough (100 genes in our implementation), a heatmap representation is rendered using the Protovis library. The list of selected genes can be exported for further analysis.

4 Results

We conducted an anecdotal evaluation of our co-regulation visualization with the help of the PROJECT coordinator and four geneticists working on regulation patterns in T-cells, B-cells and NK cells. The four geneticists were selected so that their computer operating abilities spanned a broad range, from active involvement in the lab's bioinformatics efforts to limited familiarity with analysis software. As part of the evaluation we introduced the approach and explained its limitations, then demonstrated our prototype while asking questions and invited users to comment. Two users interacted with the prototypes themselves.

All subjects decided that the co-expression map is useful. The primary workflow that our users identified was to project their own genes of interest onto one or more cell spaces. One subject would also look for global patterns of co-regulation, possibly over multiple maps and suggested we link multiple maps in separate browser tabs, such that selections of genes performed on one map are mirrored onto the others. One subject suggested using this application to create customized datasets by selecting subsets of co-regulated genes from explored datasets and exporting them in a convenient tabular form.

All users rated ease of use as higher than other systems they have worked or experimented with. They were excited to be able to access visualizations in a browser and several stated that this makes them more likely to use the visualizations. One of our subjects thought data maps could be useful for researchers new to a lab since they could start analyzing data right away. She then extended this idea to non-PROJECT members and mentioned she would like such visualizations to be present in other data sources as well. She added that her particular lab has good technical support, but that since she is close to graduating and considering doing research on her own, this approach seems very appealing.

Most subjects said the available features are enough for quick data analysis. Two users explicitly complimented the superposed expression profiles, stating that they summarize data well and can guide exploration. All users were happy with the heatmap upon-selection mechanism, with the ability to export selected sets of genes and highlight personal genes of interest. Several users asked for more hyperlinking and metadata features.

A majority of our subjects identified the static nature of the maps as a non-issue. Two of them expressed the desire to customize the cell types over which genes are projected. However, they agreed that there are relatively few cell subsets that they would choose from and that multiple maps covering these possibilities would probably work. We note that these two users were the ones most comfortable using analysis software in their daily research and were highly

familiar with the PROJECT data, explaining the desire for increased flexibility. The PROJECT coordinator commented about the benefits of being able to accompany raw data with relevant visualizations and the minimal overhead in deploying and maintaining the map system by simply copying a directory structure. He has since decided to switch the lab's database-driven distribution system to a map oriented one.

5 Discussion

5.1 Design

Instead of the traditional visualization flow of data specification followed by visualization recomputation, our co-regulation maps suggest a different approach: all data is shown at once and data specification/abstraction is done at the time of visualization through zooming and panning. Zooming can be used to summarize data at different abstraction levels, such that relevant information is available to the user at all zoom levels. Since the visual information conveyed is itself spatial, co-regulation visualization is suited for this approach.

Static maps can be synergistically coupled with interactive web elements implemented in Protovis. Focus+context visualizations can be created so that maps offer the context while focus views are implemented in Protovis. We note that we advocate for simplicity: merely replicating the complexity of stand-alone systems on the web was not our goal.

5.2 Uses

Co-regulation maps are not intended to compete with advanced micro-array analysis systems. While biologists often work intensely on specific data-sets for which the flexibility of an advanced analysis system is warranted, they often need to browse through related data-sets where the relevance cannot be clearly assessed. Running a time-consuming visualization on each piece of such data is an overhead which our maps eliminate. As our evaluation indicated, this might be especially relevant for researchers lacking access to a strong computational infrastructure. Similarly, users often want to relate their own data to existent data volumes, a task made easy by loading genes of interest on existent co-regulation maps. Finally, data intensive projects want to distribute readily available visualizations along with raw data so that their users can gain insight into the data without having to run their own analysis. The fact that our collaborator, a coordinator of a data intensive lab, has decided to replace his database-centric data distribution with a map setup supports this claim.

5.3 Opportunities

Linking multiple co-regulation maps together (e.g. for different cell families) could answer questions about conservation of gene function over multiple conditions, a question raised by one of our subjects. In Figure 2 we show an example of

a preliminary implementation of this function using cookies to pass information between multiple browser tabs. This functionality was not evaluated yet.

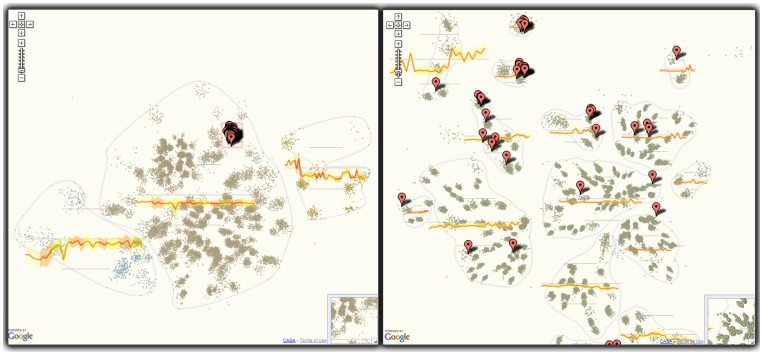

Fig. 2. Linked co-regulation maps of the T-cell (left) and B-cell (right) families. A selection in the T-cell map is reflected onto the B-cell map. A few groups of genes that are co-regulated in both cell families are noticeable by inspecting the upper part of the B-cell map.

During our evaluation, users were excited about the opportunities of collaboration offered by maps. Exchanging interactive images rather than static ones and sending links rather than datasets was positively received. Maps also support more integrated collaborative work, such as annotations, well since the static nature of maps ensures that each user has the same view of the data and that shared comments target the same visualization elements.

Finally, co-regulation maps can be extended to display more complex gene relationships while Google Maps implementation can be applied to other visualizations as well, such as standard heatmaps. For example, employing machine learning techniques on gene mircoarray data, a collaborator separated genes into functional modules and submodules with associated regulators. In Figure 3 we show a visual representation of this module space. Instead of the bisecting K means clustering hierarchy described in section 3, we use the two-level module/submodule hierarchy to draw genes belonging to the same submodules and modules closer together. Enclosing rectangles are then drawn over the modules and submodules. Information about the module, derived from our collaborator's analysis is shown on the right together with a complete heatmap of the module's genes and regulators. The analysis was performed on 346 cell types making dynamically generated heatmaps slow to render using client graphics. However, since genes in a module are predefined, heatmaps can be computed as browsable Google Maps themselves. Protovis implemented axes that are linked to the heatmap's panning and zooming and thus stick with the map, are attached on the sides ensuring that users know what genes and cell types they are focused on.

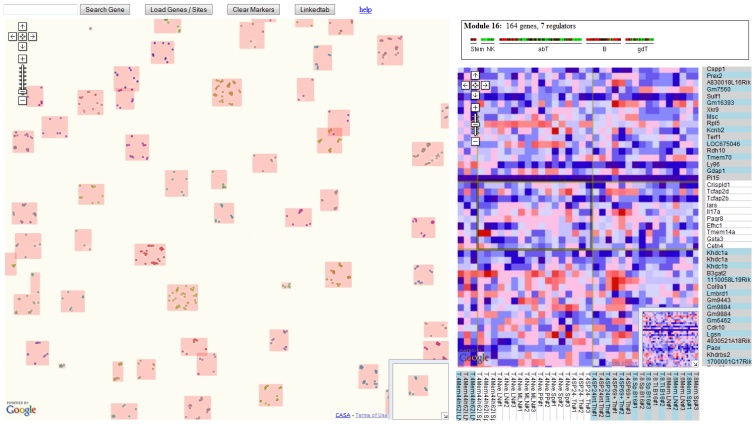

Fig. 3. Google map of gene modules and submodules embedded in 2D on the left. Information about a selected module, including a browsable heatmap, on the right.

6 Conclusion

We presented a low-overhead approach for browsing through large, unfamiliar micro-array datasets. We construct pre-computed planar embeddings of genes' expression values over multiple conditions such as cell types. We then render them as static images and display them using the Google Maps API along with an intuitive set of interactions. The contributions of this work include design elements, uses and opportunities for this type of visualization, and an evaluation that indicates that such visualizations are desirable for exploring novel data, casual browsing, disseminating results and data, and relating small data-sets to existent data volumes.

References

1. Eisen, M., Spellman, P., Brown, P., Botstein, D.: Cluster analysis and display of genome-wide expression patterns. Proceedings of the National Academy of Sciences **95** (1998) 14863
2. Hochheiser, H., Baehrecke, E., Mount, S., Shneiderman, B.: Dynamic querying for pattern identification in microarray and genomic data. In: Proceedings of IEEE International conference on Multimedia and Expo. Volume 3., Citeseer (2003) 453–456
3. Seo, J., Shneiderman, B.: Interactively exploring hierarchical clustering results. Computer (2002) 80–86
4. : (Decision site for functional genomics) `http://www.Spotfire.com`.
5. : (Cutting-edge tools for expression analysis) `www.silicongenetics.com`.
6. Kuehn, H., Liberzon, A., Reich, M., Mesirov, J.: Using GenePattern for gene expression analysis. Current protocols in bioinformatics/editoral board, Andreas D. Baxevanis...[et al.] (2008)

7. Kent, W., Sugnet, C., Furey, T., Roskin, K., Pringle, T., Zahler, A., et al.: The human genome browser at UCSC. Genome research **12** (2002) 996
8. Viegas, F., Wattenberg, M., Van Ham, F., Kriss, J., McKeon, M.: Manyeyes: a site for visualization at internet scale. IEEE Transactions on Visualization and Computer Graphics **13** (2007) 1121
9. Viégas, F., Wattenberg, M., McKeon, M., Van Ham, F., Kriss, J.: Harry potter and the meat-filled freezer: A case study of spontaneous usage of visualization tools. In: Proc. HICSS. (2008)
10. Bostock, M., Heer, J.: Protovis: A Graphical Toolkit for Visualization. IEEE Transactions on Visualization and Computer Graphics **15** (2009) 1121–1128
11. Johnson, D., Jankun-Kelly, T.: A scalability study of web-native information visualization. In: Proceedings of graphics interface 2008, Canadian Information Processing Society Toronto, Ont., Canada, Canada (2008) 163–168
12. Shannon, P., Markiel, A., Ozier, O., Baliga, N., Wang, J., Ramage, D., Amin, N., Schwikowski, B., Ideker, T.: Cytoscape: a software environment for integrated models of biomolecular interaction networks. Genome research **13** (2003) 2498
13. Aravindhan, G., Kumar, G., Kumar, R., Subha, K.: (AJAX Interface: A Breakthrough in Bioinformatics Web Applications)
14. Berger, S., Iyengar, R., Ma'ayan, A.: AVIS: AJAX viewer of interactive signaling networks. Bioinformatics **23** (2007) 2803
15. Gretarsson, B., Bostandjiev, S., ODonovan, J., H
"ollerer, T.: (WiGis: A Framework for Scalable Web-based Interactive Graph Visualizations)
16. Yates, T., Okoniewski, M., Miller, C.: X: Map: annotation and visualization of genome structure for Affymetrix exon array analysis. Nucleic Acids Research **36** (2008) D780
17. Arakawa, K., Tamaki, S., Kono, N., Kido, N., Ikegami, K., Ogawa, R., Tomita, M.: Genome Projector: zoomable genome map with multiple views. BMC bioinformatics **10** (2009) 31
18. Keim, D.: Information visualization and visual data mining. IEEE transactions on Visualization and Computer Graphics **8** (2002) 1–8
19. Fruchterman, T., Reingold, E., of Computer Science, D., of Illinois at Urbana-Champaign, U.: Graph drawing by force-directed placement. Software: Practice and Experience **21** (1991) 1129–1164
20. Tejada, E., Minghim, R., Nonato, L.: On improved projection techniques to support visual exploration of multi-dimensional data sets. Information Visualization **2** (2003) 218–231
21. Morrison, A., Chalmers, M.: A pivot-based routine for improved parent-finding in hybrid MDS. Information Visualization **3** (2004) 109–122
22. Chalmers, M.: A linear iteration time layout algorithm for visualising high-dimensional data. In: Proceedings of the 7th conference on Visualization'96, IEEE Computer Society Press Los Alamitos, CA, USA (1996)
23. Paulovich, F., Minghim, R.: HiPP: A novel hierarchical point placement strategy and its application to the exploration of document collections. IEEE transactions on visualization and computer graphics **14** (2008) 1229–1236
24. Watanabe, N., Washida, M., Igarashi, T.: Bubble clusters: an interface for manipulating spatial aggregation of graphical objects. In: Proceedings of the 20th annual ACM symposium on User interface software and technology, ACM (2007) 182
25. Kass, M., Witkin, A., Terzopoulos, D.: Snakes: Active contour models. International journal of computer vision **1** (1988) 321–331