ATCGes – A Tool For Curating Genome Expression Signatures

Louie Dinh

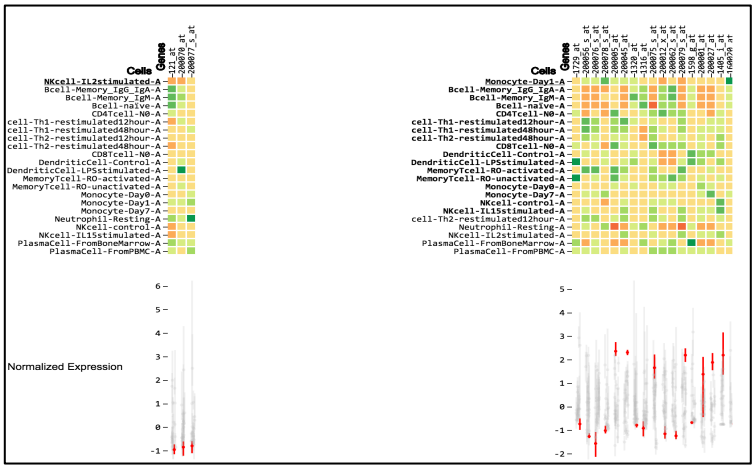

Figure 1: An example of ACTGes in action.

Abstract—In this paper we present A Tool for Curating Genome Expression Signatures (ATCGes). Understanding biological principles and diseases require accurate observation of cells in their native tissue environment. Such observations of the dynamic cellular environment can be carried out with technologies like microarrays. Currently these tools measure the raw abundance of biological molecules and cell proportions act as a confounder for many statistical tests. There are techniques, so called computational deconvolution methods, that allow us to retrieve the relative cell proportions and control for them in analyses. All such techniques require a signature matrix of gene expressions. This tool is designed to ingest microarray data, and allow the user to iteratively produce such a signature matrix. ATCGes focuses on information density, visual cues for candidate pruning, and fast iteration on signature matrix production.

Index Terms—Microarrays, RNA, Deconvolution, Signature Matrix, Gene Expression, Flow Cytometry, FACS

1 Introduction

Studying genomics data is difficult because techniques for observing cellular processes in-vitro are severely limited. However, the capabilities of tools for gathering data ex-vivo are exponentially increasing. Since the advent of sequencing, we have progressed to processing an entire human genome for approximately $1000. Furthermore, microarrays allow us to measure the gene activity of all of our 20,000 genes simultaneously.

A major deficiency in microarray gene expression data is the inability to measure relative cell proportions. We are capable of saying that gene A is more highly expressed, but unable to separate the causes of increased expression versus increased cell proliferation. These are two very different phenomenons. Understanding cell population size is crucial to disease etiology. For example an increase in different subsets of immune cells can either mean your body is fighting or helping cancer progression [Ostrand-Rosenberg 2008]. Another reason for measuring cell proportion is to control for its effects as a confounder in statistical analyses comparing gene expression between samples.

Models that don't take cell proportions into account lose much of their statistical power. Since measuring cell proportions is costly and time intensive, we would like to computationally determine the relative cell proportions.

Given the confluence of factors like cheaper sequencing, gene expression microarrays, and methylation microarrays, we'd like to deconvolute the mixture of cells back to their original proportions. This is known as the source separation problem. For any given matrix of expression, this is easy if it has an inverse and can be solved analytically.

However, in cells, there are a host of issues to confront. The main ones being noisy measurements, linearly dependent rows, and the large size of the expression matrix. Inverting the matrix is difficult when dealing with tens of thousands of measurements because this process is very computationally expensive O(N^3) even if all

• *Louie Dinh. Department of Computer Science at The University Of British Columbia. E-mail: louiedinh@gmail.com*

measurements were perfect and linearly independent.

Many deconvolution techniques can quantify cell proportions in tissues where each cell subset has been individually measured. Examples include constrained projection [Koestler 2013], least squares [Abbas 2009], support vector regression [Newman 2015] and quadratic programming [Houseman 2012, Gong 2011]. All of these techniques require a known expression matrix derived from isolated cell subsets to estimate the relative proportion of cells in an unknown mixed sample. The crux of all of these deconvolution techniques is identifying a signature expression matrix that has nice properties. Essentially, we want differentially expressed regions that allow us to tease apart the different cell proportions. Manually looking through thousands of genes is not feasible. We want a way to quickly select a candidate gene signature matrix, deconvolute using one of many techniques described above, and then iterate if necessary.

2 Related Work

To the author's knowledge, there are no other tools aimed at curating a signature expression matrix. Instead, existing tools are aimed at visualizing the expression profiles and showing how the expression patterns are related. These tools tackle both between cell types and within cell type comparisons. In a single cell type, the other tools are trying to surface relationships between genes. Through the noise, we'd like to identify whether they work together, in which case they will be expressed together, or inhibiting, in which case one will be highly expressed when the other is not. The other task is to find relationships between samples for the same gene. In this setting, the sample is examined under several treatments or exists in different disease states. The task then is to identify how the patterns of expression for a single gene change across treatments or states.

ATCGes is different because it facilitates a different goal, the identification of genes that have a profile unique to a particular cell type. These so-called biomarkers would be identified as outliers when approached with the perspective described above. Concretely, the other tools would identify a gene with a characteristic partition of samples where one sample expresses the gene differentially from the rest. While it would be theoretically possible to use the other tools to carry out such a task, in practice the resolution of the other tools make such identification infeasible. Additionally, ATCGes supports the production of a signature matrix that uniquely identifies each cell under inspection. Such an operation would require an external recording mechanism and doesn't allow the gestalt of the entire matrix to be examined.

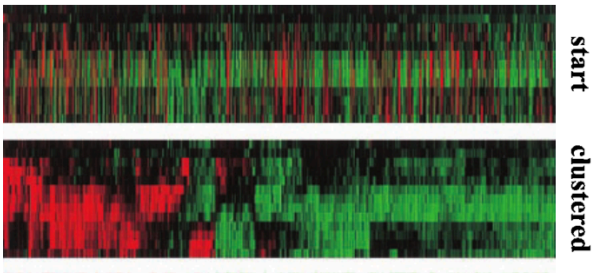

Figure 2: Pairwise clustering from Eisen (1998)

We briefly summarize the main approaches to visualizing gene expressions below.

2.1 Clustering And Heat Maps

Clustering is a technique for grouping objects based upon a similarity score. In this case, we calculate the similarity based upon gene expression. Clustering has been shown to group together genes that are co-functional or genes that have similar functions. Usually the cluster is encoded as a heat map with a red-green diverging colour map, a staple of biological visualizations.

Eisen [1998], demonstrated a pairwise linkage clustering visualization that is adapted from phylogenetic tree reconstruction algorithms (See Figure 2). A tree of similarity is built between the genes and then they are colour coded to reflect their normalized expression levels

Another method of clustering, called bi-clustering, attempts to cluster both the genes and experimental conditions simultaneously (See Figure 3). Gonçalves [2009], introduced a bi-clustering technique that could be used to understand gene expression time-series data. The output is a filtered block of genes whose expression patterns move in sync over time. This is all displayed in a SPLOM-like fashion, with each component represented by a line chart and visualizes a particular cluster of genes.

2.2 Self Organizing Maps

Another technique that is used to understand expression profiles is the self-organizing map (See Figure 5). To create a self-organizing map, one starts with a simple geometry like a rectangular grid. These points are then projected k-dimensional space occupied by the gene expression data and then shifted iteratively towards the data points. Each point moves towards the data point in proportion to the distance away from the data point. After many thousands of iterations, the map will identify clusters of genes with similar expression patterns. Once again, this can be displayed in a SPLOM-like fashion to visualize the per-cluster expression patterns.

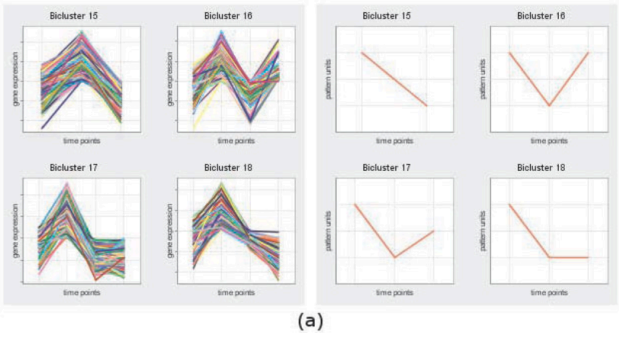

Figure 3: Biclustering example from Gonçalves (2009)

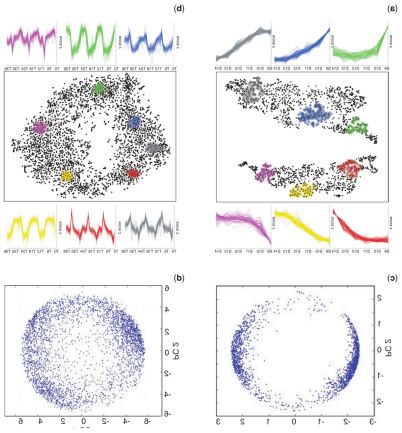

Figure 4: t-SNE from Bushati (2011)

2.3 Dimensionality Reduction

Since the expression of k genes can be interpreted as a vector in k-dimensional space, it is natural to attempt dimensionality reduction. Bushati [2011] demonstrates the use of t-SNE and PCA in projecting the k-dimensional space down to 2D that is amenable to a scatter plot (See Figure 4). The data is grouped by treatment, in this case embryo development stage, and overlaid onto the graph. There are also embedded line charts showing the expression of particular gene subsets in different experimental conditions.

3 Data and Task

ATCGes is designed to work on gene expression data from microarrays. Microarrays measure the quantity of a biological molecule, in this case RNA, present in a sample. A cell expresses genes in the form of RNA and the microarray allows us to translate this into a luminance measure based on the fluorescence of particular wells.

In microarray experiments, we will measure the expression levels of approximately 20,000 genes in a few dozen cells. Each cell will have a few (< 10) replicates to quantify the variance in expression within the same cell type.

Our problem is to use these measurements to choose a signature matrix that consists of a set of genes, called biomarkers, which are differentially expressed between the cell types. The identification of such biomarkers can proceed in stepwise fashion because each gene can be considered independently. Once a set of biomarkers is identified it would be saved and used as input for various deconvolution algorithms.

3.1 Abstracted Data

We are dealing with tabular data. The expression matrix consists a 4 dimensional cube. The axes are cell type, gene identifier and replicate identifier and the value is the luminance measure. The scale of each axis is as follows: dozens of cells types, tens of thousands of genes, and between one and ten replicates. The keys are categorical (gene identifier, cell type identifier, replicate identifier) triplets and the values are quantitative between 0 and 120,000 representing luminance of the microarray spot.

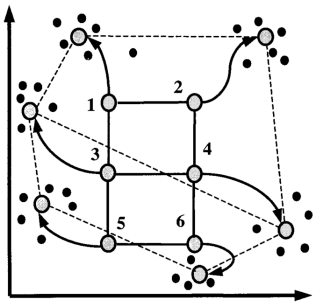

Figure 5: Self Organizing Map from Tamayo [1999]

The task is attempting to produce a subset of the gene identifiers (categorical) that will act as the signature matrix able to distinguish between all these cell types.

3.2 Abstracted Task

At the high level, the user is attempting to produce a set of gene identifiers by annotating genes whose expression can distinguish between the different cell types. These genes must be saved, along with their expression data, into a candidate solution in the system.

During the search, the target is known. The user is looking for a specific feature, a gene that is differentially expressed in one or two cells against the background cells. A good biomarker is a gene that is differentially expressed with low variance between cells of the same type but with a mean expression level that is easily distinguishable from cells of other types. The location is unknown and we are searching through approximately 20,000 genes.

In attempting to identify a gene that acts as a good marker for the cell a user would compare the expression of a single gene across the different cell types.

Once a gene is found, it is recorded into the basis matrix for export and use in downstream analyses.

4 Solution

From the high level perspective [See Figure 6], the system partitions the visualization into two juxtaposed views. The view on the left shows the candidate genes for the cell currently under consideration, termed the current cell of interest. The view on the right records the current partial solution. Upon completion of the task, the right view will contain a visual encoding of the full solution to be used in downstream analyses.

4.1 Data Retrieval And Preprocessing

All sample data used for the initial design and implementation of ATCGes comes from the Gene Expression Omnibus (GEO), a public repository for microarray experiments. In particular, we used the experimental data by Abbas et al. [2005] from their study of whole blood deconvolution.

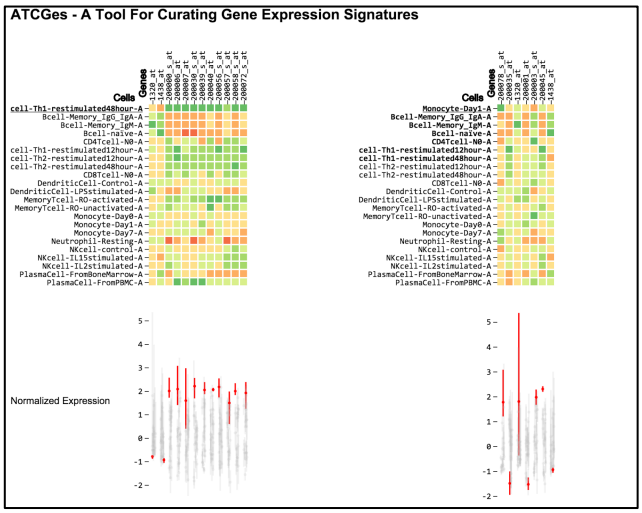

Figure 6: High level overview of ATCGes

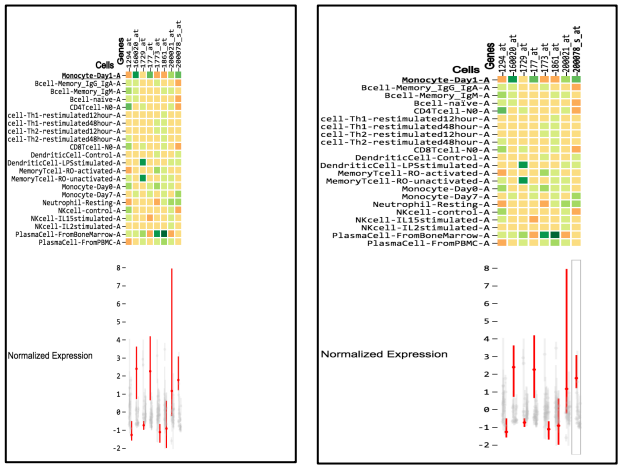

Figure 7: (Left) Before mouse over of gene. (Right) After mouse over. Notice the bolded gene label on top and the box around the relevant variance bars.

The dataset comes from GEO [Edgar et. Al 2002, accession GSE22886] and consists of microarray measurements of 12 immune cell types (B cells, CD14+ cells, CD4+ CD45RO+ CD45RA- T cells, CD4+ T cells, CD8+ T cells, IgG/IgA memory B cells, IgM memory B cells, Monocytes, NK cells, Neutrophils, Plasma cells from bone marrow, and Plasma cells from PBMC). Each cell type has multiple replicates totalling 228 samples. Each sample is assayed for the expression of 22,283 genes.

From the raw dataset, we used custom R scripts to pre-process the data as follows. Expression data from GEO consists of cell identifier, gene identifier and replicate number. Each replicate has a cell type and gene. We group by (cell type, gene identifier) pairs and have multiple replicates in each group. The expression values in each group are averaged to get a mean expression for that gene in that cell type. We also compute the standard deviation of each group. Expression levels are then normalized on a per-gene basis using Mahalanobis distance (Equation 1) to bring them into a common scale for visualization.

$$(x - \mu) / \sigma$$

Equation 1: x is raw gene expression measure, μ for mean expression data for that gene, σ for variance of expression.

4.2 Visualization

ATCGes works in a stepwise fashion where the user concentrates on selecting the signature gene for one cell at a time. When the user initially starts the tool, it will have a preselected cell of interest, in this case monocyte-Day1-A, that is displayed at the top of the cell labels in the left view [See figure 7].

The tool focuses heavily on information density. A main design consideration was to keep all information relevant to the next decision within the field of view. We chose the heat map and stylized boxplot, reducing boxes to just lines, because it scales well to the dimensions present in our problem.

4.2.1 Initial Screen and Mean Expression Encoding

The initial screen displays genes identifiers along the top axis, and cell identifiers along the side. The heat map encodes normalized expression data on a per-gene basis, i.e. vertically. Strongly expressed genes are green and strongly inhibited expressed genes are orange with yellow being the neutral ground. The exact expression values are unimportant because we are focusing on the relative difference of expression per gene group. Thus, the legend of colour to expression value is purposefully omitted to reduce the cognitive load of the user.

4.2.2 Variance Encoding

Directly below the heat map is an aligned stylized box and whisker plot. The vertically grouped plots show the expression levels for the gene directly above it across the different cell types. The bar for the cell of interest is highlighted in red and all other cells are in grey. In each box and whisker plot, the greyed out bars have reduced opacity and are also horizontally jittered. This is to prevent visual occlusion and guarantee that the expression levels of each gene for the cell of interest are always prominently displayed.

The level of visual occlusion encodes one aspect of a good biomarker candidate. Namely, we are attempting to identify a gene whose expression level for the cell of interest is well separated from the background expression of other cells. This directly maps to a low level of visual occlusion of the red bar in the box plots.

Furthermore, the size of the variance bars indicates how much the expression fluctuates between cells of the same type. The other metric that determines a good biomarker is a good separation from other cell types. This can be seen via both the heat map and the variance bars. In the heat map, the expression should be at the extremes of the spectrum, either bright green or bright orange. In the boxplot, this would be encoded as the red bar being at either the top or bottom edge of the plot, with not many grey bars overlapping.

4.2.3 Interaction Idioms

Several interaction idioms were added to improve the usability of the system. Due to the high information density, the propensity to misclick became readily apparent in early iterations.

We added responsive elements to the visualization to indicate the exact position of the mouse pointer. Upon hovering over clickable labels, we change the cursor and highlight the label that is to be activated. Clicking a cell label will change the current cell of interest and redraw the entire left panel. The cell of interest will move to the top and candidate genes for that cell will be filtered and displayed to the user. The colour encoding is also remapped to the new range of values.

Similarly, hovering over a gene label will highlight the gene of interest. Clicking on a gene is an indication that we will be using it as the biomarker for the current cell of interest. The gene is added to the partial solution in the view on the right

In the variance plots, we found that the horizontal jitter reduces the discernibility of the variance plots between gene groups. To ameliorate this, we use linked highlighting to emphasize the corresponding variance plots associated with that gene. This manifests as a grey border that appears when the user's cursor is hovering over a particular gene, and then disappears once the cursor is moved. See (See Figure 7 Left and Right).

4.2.4 Partial Solution Encoding

The partial solution, displayed on the right of Figure 1, shows the current genes that we have in our current expression signature. This encoding is very similar to the encoding of the candidate genes on the left. The only notable difference is that in the candidate gene view, the variance bars in red are always for the cell of interest. In the partial solution, the variance bar in red is for the cell at the matching index. For example, the variance bar that is highlighted in red that is second from the left is encoding the expression variance for BCell-Memory-IgG-IgA,-A rather than Monocyte-Day1-A. This is because the partial solution is interested in the differential expression of particular gene/cell pairs as opposed to just the differential expression of the genes themselves holding the cell constant.

1. Initialize candidates array to empty and cell of interest to the current cell of interest.
2. Group data triplets of (cell identifier, gene identifier, expression value) by gene identifier.
3. In each group, sort by gene expression value descending.
4. If the top cell identifier or bottom cell identifier in each group matches the cell of interest and passes a certain difference threshold to the second candidate, add to candidate pool.
5. Return candidates

Equation 2: Candidate Gene Filtering Algorithm

4.2.5 Candidate Gene Filtering Algorithm

Microarrays measure tens of thousands of genes, but not many can serve as biomarkers. When attempting to identify a biomarker for a cell type, we look for genes that have distinct mean expression values for that cell type along with low variance to minimize the chance of overlapping expression values. To allow the user to focus on only the viable candidates, we implemented a filtering algorithm that removes genes that are unlikely to be good biomarker candidates. The algorithm is described in Equation 2.

4.3 Alternatives

During the design of ATCGes, several alternative visual encodings were considered.

Initial implementation relied upon variable sized view panes for each heat maps to account for the differing number of candidates of each cell. This resulted in an unacceptable level of visual flux during transitions and was phased out for a fixed width approach.

Our strongest visual encoding, the variance plots with transparency, was initially just a standard boxplot. We found the visual occlusion to be intolerable and collapsed each box into a line. In addition, we added the transparency effect that coincided well with the separability of a biomarker.

Furthermore, we initially started with arbitrary ordering of the labels. This was confusing because each step results in a complete rendering of both the candidate genes on the left and the partial solution matrix on the right. This was very disorienting. To keep some visual anchors and minimize the label re-orders, we modify the partial solution matrix labels by removing the current cell of interest and inserting it below the previously bolded cells. This gives the appearance of constancy for the top portion of the partial solution and a shifting down of the bottom incomplete portion.

5 Implementation

This project was implemented using R [Gentleman 2009] for the data processing and D3.js [Bostock 2011] for the visualization system itself. All data was retrieved from the Gene Expression Omnibus [Edgar 2002] and several helper libraries including underscore.js, colour brewer [Brewer 2001] and Bioconductor [Gentleman 2004] were used.

5.1 Data Retrieval And Cleaning

To retrieve the data from GEO, we used the Bioconductor package available through the CRAN repository. The GEO ID for the data set is GSE22886. The data is loaded into a Bioconductor data structured called a Large Expression Set, optimized for microarray data. Using Hadly Wickam's excellent data cleaning tool packages tidyr and dplyer [Wickam 2014], we aggregated and calculated summary statistics as described in section 4.1. The data was then serialized and saved as JSON using the rjson [Couture-Beil 2013] library.

5.2 Browser Implementation

The visualization was implemented using Mike Bostock's [2011] D3 library. Initial inspiration was taken from Tom May's day/hour

heat map [May 2015]. The starting code was examined for implementation idioms and styling themes. We then proceeded to code the heat map de-novo. The bulk of implementation work involved properly modularizes the code to minimize recoding many of the idioms.

During the coding phase, there was a heavy emphasis on creating high-level procedures that lays out large portions of the visualization. For example, the same code draws both the left view and the right view of the visualization. This was enabled by the relative orientation styling options supported by both CSS and SVG. The idea is to make the rendering code extensible enough to support the modifications unique to each piece. A particularly tricky piece of implementation involved the relative ordering of the genes and cell labels on the axis. Both the left and right views required these labels but in different ways. In the left view, the cell labels only needs to specify the cell of interest at the top, and all other labels can be in arbitrary positions. In contrast, the cell labels on the right must maintain their ordering because it corresponds to the order in which the user makes progress on the task. Similarly, the gene ordering in the left view doesn't matter because they are all equal candidates for a particular cell of interest. Once again, on the right the ordering is significant because the index or relative position corresponds to the relative position of the cell for which it is the biomarker.

Other design considerations included the use of CSS styling to keep a unified theme throughout the visualization. In addition, we used Colour Brewer [Brewer 2001] to create a diverging colour map for the expression signatures. We chose the green/orange colour map because it is close to the ubiquitous green/red colour map used by biologists and provides sufficient visual separation for distinguishing the candidates.

6 Results

Here we present a sample walkthrough of ATCGes in action. A user will have performed a microarray experiment and wants to perform analyses on the resulting data. Directly using the microarray data has many drawbacks, specifically the confounding factors of cellular proportion differences in samples [Shen-Orr 2010]. To control for these factors, the user would run the data through a deconvolution pipeline within which ATCGes would be the first step.

The data is first cleaned and reformatted into JSON as described in section 5.1. Then the data is loaded into ATCGes to produce a visualization that would look similar to Figure 7 (Left). The user examines the current gene candidates by hovering the mouse over genes (Figure 7 Right) and examining their expression distributions expressed in both the heat map and the variance plots. When a good gene candidate is identified (See Section 4.2.2), the user will click on the gene, adding the first biomarker to the partial solution (See Figure 8).

The user then selects the next cell in the list to consider. In this case it would be Bcell-Memory_IgG_IgA-A, which is second. After a series of steps, each identified biomarker is added to solution (See Figure 9 and 10). When the user is finished, he will have a bolded line for each gene of interest and a full set of biomarkers to be used in the deconvolution pipeline (See Figure 1 for an example).

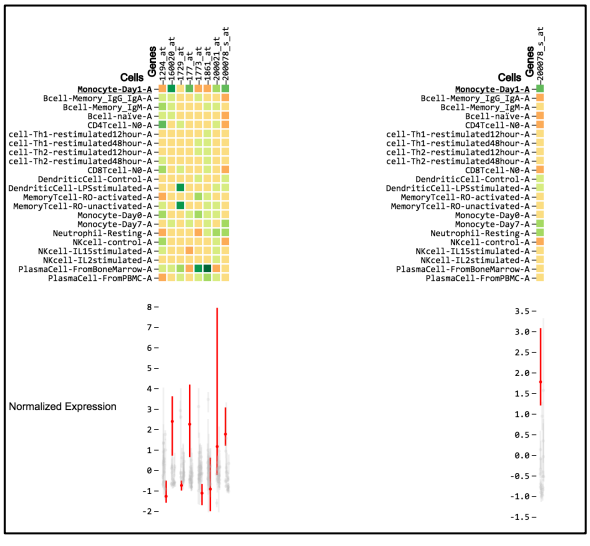

Figure 8: After adding the first gene to the partial

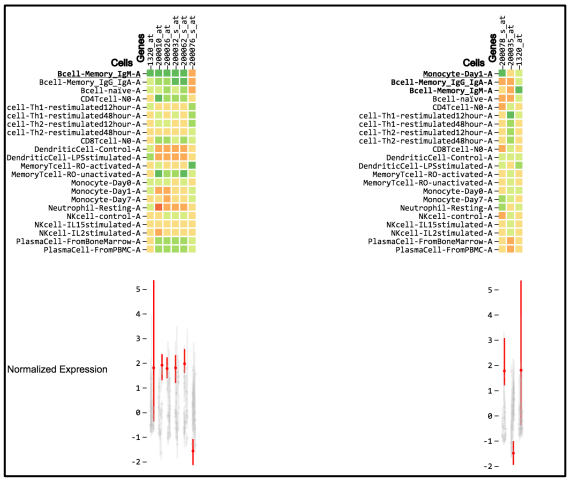

Figure 9: ATCGes after adding 3 biomarkers, one of which is a terrible marker.

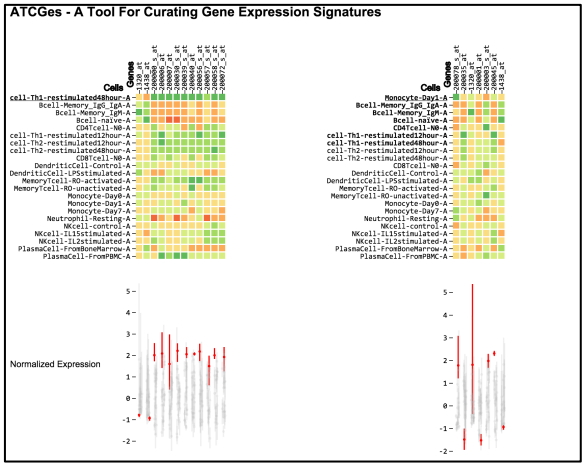

Figure 10: ATCGes halfway through the process. Notice the strongly coloured diagonal encoding the markers.

7 Discussion And Future Work

7.1 Strengths

After completing the project, we found that the signature selection problem is strongly amenable to being cast as an information visualization task. The iterative nature of the problem lends itself well to a high information density encoding and interaction with step-wise candidate selection. We were very pleased with how the entire partial solution can always be kept within the frame of view.

Our initial implementation of the variance bars used a more classic rendering of box-plots and suffered heavily from visual occlusion. This led to our stylized boxplots where the boxes are reduced to just lines. After dialling the opacity down on the background genes and colour encoding the gene of interest, we arrived at a very powerful visual cue for biomarkers. The perception of a good marker as well separated from the group with small range, matches our intuition on the traits of a good biomarker.

Another aspect that caught us by pleasant surprise is how much interactivity adds to the experience. The authors have previously curate genes for expression signatures and the interactivity allows backtracking out of bad solutions. Often, the greedy approach will select a gene that is a good marker for cell A, which might have dozens of candidates, but then leaves cell B with no candidates available. With ATCGes, we can quickly backtrack and swap out the biomarker for cell A and freeing up the gene for use as cell B's biomarker.

7.2 Weaknesses

We are still unsatisfied with the encoding of biomarkers in the right view of ATCGes. Once a biomarker has been selected, the colouring in the variance plots is no longer consistent. Each red bar denotes a different cell's expression of the vertically aligned gene. This encoding makes sense in the left panel because we only consider one cell at a time. Given more time, we would further iterate this encoding to avoid any confusion for the user. Another issue is that the markers are either expressed at the bottom or top of the range for each gene. This results in our sight line bouncing up and down. A better approach would be selective inversion along with a glyph to indicate the orientation rather than the current approach.

Furthermore, the current interaction flow requires too much clicking and mouse navigation. This is certainly not the ideal user experience and definitely merits some improvement. Another UX issue pertains to the colour mappings. Due to the nature of the problem, the expression values in the map skew much further to the high end because there is a physical limitation that a gene isn't expressed and the value is 0. Thus the normalized bins don't spread the data equally across the colour bins. A better approach would take this skew into account to make better use of the colour space.

7.3 Future Work

Currently, the most pressing issue is to integrate ATCGes into a full deconvolution tool chain. That would allow us to benchmark the output signatures against manually curated signatures without the tool and provide an objective measure of improvement. In addition, each deconvolution technique has unique properties and our perception of a good signature matrix with ATCGes may not be well matched to the deconvolution tool. Thus, it is imperative that we measure absolute performance in a production setting.

Additionally, we would like to add a heuristic to sort the candidates in order of relevance. We believe that a human is required to make the trade-offs between separability and variance in expression. However, we can certainly take steps to minimize the number of comparisons a human would have to make. This would have the added benefit of improving the user experience through a confirm and modify approach to signature selection rather than the currently implemented hunt and peck navigation.

7.4 Conclusions

We believe that ATCGes is a strong addition to the cell deconvolution tool chain. To the author's knowledge, no such tool exists and signature matrices are currently curated through a heuristic approach on the raw data. This has yielded reasonable results but doesn't allow for the rapid iteration on signature matrices that ATCGes enables. Previous work on deconvolution often only considers well-separated mean expression values and ignores the variance measures available. ATCGes allows the consideration of both separability and variance of expression of the entire signature at once.

Furthermore, current deconvolution methods rely on an accurate and well-chosen signature matrix to perform their computations. Still, we are only capable of understanding very well characterized tissues like whole blood. As we stretch our methods and attempt to understand more complex tissues, the need for good signature matrices becomes more pressing. For the foreseeable future, highly complex tissues may never be fully characterized and deconvolution must move towards a more iterative model than at present.

In conclusion, we hope that ATCGes enables users to create better gene expression signatures for the cell deconvolution problem. There is still much work to be done on improving the scalability, user interface and integration of our tool. In the grand scheme of things, ATCGes is just one tiny step on the way to fully understanding the dynamic environment inside of all cellular life.

Acknowledgments

The authors wish to thank Dr. Tamara Munzner for her feedback throughout the process and Dr. Sara Mostafavi for guiding us to the problem in the first place.

References

[1] "M. Wattenberg. A note on space-filling visualizations and space-filling curves. In Proc. IEEE Symp. Information Visualization (InfoVis), p 181-186, 2005"

[2] Ostrand-Rosenberg S. Immune surveillance: a balance between protumor and antitumor immunity. Curr Opin Genet Dev 18:p11–18, 2008

[3] Abbas AR, Baldwin D, Ma Y, Ouyang W et al. Immune response in silico (IRIS): immune-specific genes identified from a compendium of microarray expression data. Genes Immun. ;6(4):p319-31, 2005.

[4] Koestler DC, Christensen BC, Karagas MR, et al. Blood-based profiles of DNA methylation predict the underlying distribution of cell types: A validation analysis. Epigenetics. 8(8):p816-826, 2013.

[5] Abbas, Alexander R., Kristen Wolslegel, Dhaya Seshasayee, Zora Modrusan, and Hilary F. Clark. "Deconvolution of blood microarray

data identifies cellular activation patterns in systemic lupus erythematosus." *PloS one* 4, no. 7: e6098, 2009.

[6] A. M. Newman et al., “Robust enumeration of cell subsets from tissue expression profiles,” Nature Methods 12, no. 5: p. 1–10, 2015.

[7] Gong, Ting, Nicole Hartmann, Isaac S. Kohane, Volker Brinkmann, Frank Staedtler, Martin Letzkus, Sandrine Bongiovanni, and Joseph D. Szustakowski. "Optimal deconvolution of transcriptional profiling data using quadratic programming with application to complex clinical blood samples."*PloS one* 6, no. 11: e27156-e27156, 2011.

[8] Houseman EA, Accomando WP, Koestler DC, Christensen BC, Marsit CJ, Nelson HH, et al. DNA methylation arrays as surrogate measures of cell mixture distribution. BMC bioinformatics 13:p86, 2012.

[9] Eisen,M.B., Spellman,P.T., Brown,P.O. and Botstein,D. Cluster analysis and display of genome-wide expression patterns. Proc. Natl. Acad. Sci. USA, 95: p14863–14868, 1998.

[10] Joana P. Gonçalves, Sara C. Madeira and Arlindo L. Oliveira, BiGGEsTS: integrated environment for biclustering analysis of time series gene expression data, BMC Research Notes 2:p124, 2009.

[11] Tamayo,P., Slonim,D., Mesirov,J., Zhu,Q., Kitareewan,S., Dmitrovsky,E., Lander,E.S. and Golub,T.R. Interpreting patterns of gene expression with self-organizing maps: methods and application to hematopoietic differentiation. Proc. Natl Acad. Sci. USA, 96: p2907–2912, 1999.

[12] Bushati N, Smith J, Briscoe J, Watkins C. An intuitive graphical visualization technique for the interrogation of transcriptome data.Nucleic Acids Res 39:p7380–7389, 2011.

[13] Edgar R, Domrachev M, Lash AE. Gene Expression Omnibus: NCBI gene expression and hybridization array data repository Nucleic Acids Res. 30(1):p207-210, 2002.

[14] Brewer, Cynthia A., 2002. http://www.ColorBrewer.org, accessed *2015-12-01*.

[15] Gentleman, Robert, Ross Ihaka, and D. Bates. "The R project for statistical computing." *URL: http://www. r-project. org/254* (2009).

[16] Gentleman, Robert C., Vincent J. Carey, Douglas M. Bates, Ben Bolstad, Marcel Dettling, Sandrine Dudoit, Byron Ellis et al. "Bioconductor: open software development for computational biology and bioinformatics."*Genome biology* 5, no. 10: R80, 2004.

[17] Wickham, H. "Tidyr: easily tidy data with spread and gather functions." (2014).

[18] Wickham, Hadley, and Romain Francois. "dplyr: A grammar of data manipulation." *R package version 0.3. 0.2* (2014).

[19] Couture-Beil, Alex. "rjson: JSON for R." *R package version 0.2* 13 (2013).

[20] Tom, May. “Day / Hour Heatmap”. URL: http://bl.ocks.org/tjdecke/5558084 (2015).

[21] Shen-Orr SS, Tibshirani R, Khatri P, Bodian DL, Staedtler F, Perry NM, Hastie T, Sarwal MM, Davis MM, Butte AJ: Cell type-specific gene expression differences in complex tissues. Nat Methods 6(2):p287–289, 2010.