Project Proposal: Gamifying the transcriptome

Chidube Ezeozue and Joel Brooks

November 26, 2012

Introduction

The reviewers of our first proposal primarily concerned with the specifics with our game implementation. In this revised proposal, we added more detail about specific elements of the game in order to address these concerns. Furthermore, we clarified the main use case of our game in the *Innovation* section of this proposal, as it seemed we weren't clear in our first proposal about how the game could be best utilized for contributing to the field.

Specific Aims

Our primary goal for this project is to create a game that can potentially utilize human powered computation to reconstruct transcripts and their relative abundance from short RNA sequence reads. We hope to accomplish this by creating a puzzle game that allows users to try and create transcripts and assign abundances to them to explain levels of exon abundance found by RNA-seq technology. In this game, the underlying biology would be abstracted away by representing relative levels of exon abundance within a gene as colored blocks, and thus the game would be designed to be appealing to those with no knowledge of the underlying biological problem they are solving. This game would not be a complete replacement for existing computational methods for transcript assembly as these algorithms have been shown to work well in many cases, but instead utilizing human powered computation to explore the solution space of transcript isoforms in cases where existing algorithmic techniques may be lacking in accuracy.

Research Strategy

Significance

Alternative splicing is an important functional element in eukaryotes (Pan et al., 2008). These splicing events allow a single gene to produce multiple mRNA transcripts and thus increased biological complexity, as different isoforms may lead to different proteins or different

regulation of the same protein (Trapnell et al., 2010). These events are quite common, and evidence suggests they occur in roughly 95% of multiexonic human genes.

The development of RNA-seq technology has greatly advanced the potential effectiveness of mRNA transcript assembly, but detecting isoforms from the millions of short reads generated by RNA-seq is a computationally daunting problem. Currently existing methods are able to asses relative exon abundance for a gene from RNA-seq data with relatively high accuracy (Trapnell et al., 2009), however reconstructing the set of mRNA isoforms that produced those exon abundances is a computationally complex problem. Cufflinks uses weighted bipartite graphs in order to find the minimum set of transcripts that explain the set of reads, however, we would like to explore the possibility of humans finding alternative sets of transcripts through a game like interface. The human solutions could then be scored against existing databases of transcripts for genes.

Innovation

Human powered computation has been shown to be of use in biologically related problems (Kawrykow et al., 2012; Cooper et al., 2010). For example, Phylo allowed humans with no understanding of biology to perform multiple sequence alignment (MSA). Phylo was designed to abstract the underlying biology away by representing nucleotide bases as colored blocks. The intuition behind such an abstraction is humans are quite good at visual pattern recognition problems. Using Phylo, humans were indeed able to outperform state of the art MSA algorithms for certain sequences. We'd like to use similar motivation for exploring the idea of humans reconstructing transcript isoforms from relative abundances of exons. We believe that this problem can effectively be conveyed in a simple block-based puzzle, and therefore a person with no understanding of computation or biology can play but still potentially find useful solutions. Thus, by abstracting the problem of transcript assembly into a simple puzzle form, we can hopefully use human powered computation to help parse RNA-seq data. Due to the overhead of human powered computation, utilizing our game for complete transcriptome assembly would be difficult. However, we envision this project being utilized in a similar manner to Phylo, where solutions to complex regions can be explored by humans and compared to those found by the current gold standards.

Approach

Data preprocessing

For our game we intend to use genomic data from the mouse to ease comparison with related work given that most other authors in this area worked with the mouse genome (Trapnell et al., 2010; Guttman et al., 2010; Feng et al., 2010; Li et al., 2011). Since read alignment and exon detection is outside the scope of this project, we would use the tool TopHat (Trapnell et al., 2009) to perform these tasks. The input to the game would therefore be expression levels for the different exons and junctions in the mouse genes.
However, to be able to stratify the game into difficulty levels we would organize the genes

by number of exons and amount of disagreement on genes' isoforms using other tools. After pre-processing the reads with TopHat, we would push exon and junction expression level data to a database for easy retrieval. We chose to use a database since the voluminous read data is not relevant to our problem and therefore not useful to store.

Game Development

Our game has two major components:

1. A backend for supplying challenges to players and aggregating their solutions. We would build this backend using a Python/Django webserver and a MySQL database or Google App Engine datastore.

2. The actual game is built as a browser-based game using HTML 5.

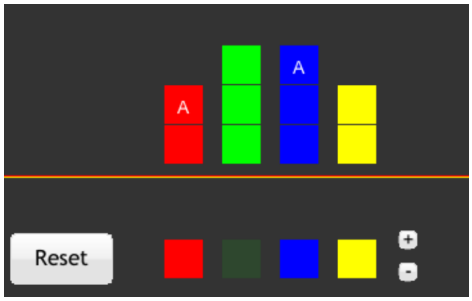

Figure 1: User interface of transcriptome assembly game.

When the game starts, players will be presented with a gene where each column of blocks represent exons discovered by TopHat, and column heights represent their relative abundances. Furthermore, columns will contain "linked" boxes that contain a unique identifier shared with a box in another column. These represent reads that mapped to more than one exon and spanned a splice junction. Players will then be able to copy this entire gene into a transcript, delete exons and vary the relative expression levels of their transcripts in order to remove boxes from the puzzle. "Linked" boxes can only be removed by transcripts that contain both exons corresponding to the "linked" boxes without the exons in between them.

An example puzzle is shown in Figure 1. Users can toggle exons to create transcripts and then increase or decrease abundance the transcripts they create. Increasing abundance for a transcript will remove one block from each exon column included in the transcript in the puzzle. Notice that the first and third column have "linked" blocks that can only be removed by a transcript that contains the red and blue exon, but not the green exon. Players will be scored using the following criteria:

- Agreement with exon and junction read data
- Number of transcripts created

- Length of transcripts created

If time permits, we would include a feature that allows players split and merge exons to discover new splice sites and better explain read data.

We will use the LimeJS[1] HTML 5 framework to build the game, as this allows us to focus on the core gameplay elements and not on small coding details. Furthermore, an HTML 5 implementation means the game will be desktop and mobile compatible, and should therefore make it easy to present the game to our testing user base. We intend to develop this game as an open-source project released under the GNU public license.

Testing

We intend to recruit volunteer testers for our game from within the MIT community; classmates, dorm mates etc. We hope to have at least 50 different people test the game before the project deadline. Testing would focus on discovering bugs, improving the user interface to make the most of the players' ability, evaluating likeability and performing the primary purpose of the game; discovering transcripts and their expression levels. There would be two primary sources of test data:

- Real read data obtained from mouse skeletal muscle cells; C2C12 mouse myoblast cell line (NCBI SRA accession number SRR037947) (Trapnell et al., 2010)
- Simulated read data obtained by passing known transcripts from the UCSC mm9 gene annotation (Karolchik et al., 2008) through Flux Simulator (Sammeth et al., 2010) to enable us compare the performance of our tool to some ground truth.

Evaluation

We would evaluate the success of our game by investigating the intersection of the our transcript discoveries with those from annotated data and popular tools like Cufflinks (Trapnell et al., 2010), Scripture (Guttman et al., 2010) , IsoInfer (Feng et al., 2010) and IsoLasso (Li et al., 2011).
We would also be interested in the time it took players to finish a round and if we have the time, we would incorporate features that will allow us study the thought process of the players.

Progress and Preliminary Results

Data Processing

We were able to successfully obtain the same read data used by Trapnell et al. (Trapnell et al., 2010) and use TopHat to map reads to NCBI build 37.1 of the mouse genome. This resulted

[1] http://www.limejs.com/

in 13,378,594 reads mapped to the mouse genome and 103,748 possible splice junctions identified.

We also simulated reads using Flux Simulator. We obtained annotated data from the mouse genome mm9. To make the problem more tractable, we eliminated all instances of intron-skipping and used only genes with at least 2 transcripts. We also capped the number of exons to 200,000 because our computing memory resources could not simulate much more exons than that. We simulated 15,000,000 reads of 75 nucleotides each and pushed information about read count and mapped exons to the database.

Server and Database

We utilize a Python/Django server to support the front end communicating using Javascript Object Notation (JSON) format. The database is a MySQL database running on the internal CSAIL MySQL servers. The server-front end communication will allow the front-end pull game data from the database and send player results to the server.

Game Front End

The game front end is currently being built using the LimeJS HTML 5 framework. We have designed the main layout for the game scenes where the main interaction between players and the game will occur. Currently the game can load a list of exons and their corresponding abundances and display the corresponding puzzle. Users can then create new transcripts by toggling which exons they want in their new transcript, and setting the abundance of that particular transcript. Some core functionality like scoring the user's solution and server communication are currently being added.

1 Updated Timeline

Activity	Start Date	Completion Date	Responsibility
Data collection[1]	October 23rd	October 27th	J.B.
Run TopHat, extract exon and junction expression levels[1]	October 27th	November 5th	J.B.
Create web server skeleton and design database schema[1]	November 1st	November 10th	C.E.
Setup front-end UI framework and initial game layout[1]	November 5th	November 17th	J.B.
Add core gameplay functionality[2]	November 17th	November 28th	C.E. and J.B.
Filter simulated transcriptome data to eliminate intron-skipping isoforms and run RNA-seq simulation[1]	November 17th	November 28th	C.E.
Add front end to server communication[2]	November 17th	November 28th	C.E. and J.B.
Mid-course report preparation[2]	November 23rd	November 25th	C.E. and J.B.
Push simulated data to the database [2]	November 25th	November 28th	C.E
Push real read data to the database	November 27th	December 2nd	C.E
hliner Test on users	November 28th	December 2nd	C.E.
Investigate intersection with Cufflinks	December 2nd	December 4th	J.B.
Investigate intersection with Scripture	December 2nd	December 4th	C.E.
Investigate intersection with IsoInfer[3]	December 2nd	December 4th	J.B.
Investigate intersection with IsoLasso[3]	December 2nd	December 4th	C.E.
Report writing	October 27th	November 30th	C.E. and J.B.
Complete final report	December 4th	December 6th	C.E. and J.B.
Prepare final presentation	December 6th	December 10th	C.E. and J.B.

[1]Completed
[2]Work in progress
[3]If time permits

Resources

Data sources

- C2C12 mouse myoblast cell line (NCBI SRA accession number SRR037947) (Trapnell et al., 2010)
- UCSC mm9 gene annotation (Karolchik et al., 2008)

These datasets are publicly available.

Relevant lectures

- L5: Genome Assembly: Consensus-alignment-overlap, Graph-based assembly
- L9: Transcript structure analysis, Differential Expression, Significance Testing

Computational resources

We do not anticipate requiring any additional computing resources apart from our personal and work computers. In the event that we require additional computing resources, we intend to use the internal CSAIL cloud.

Personal resources

We will get periodic advice and guidance from PhD students, Deniz Yorukoglu and Po-Ru Loh. We also intend to reach out to Manuel Garber, one of the authors of Scripture, and Cole Trapnell, one of the authors of TopHat and Cufflinks for high-level direction and feedback.

Bibliography

Cooper, S., Khatib, F., Treuille, A., Barbero, J., Lee, J., Beenen, M., Leaver-Fay, A., Baker, D., Popović, Z., et al. (2010). Predicting protein structures with a multiplayer online game. *Nature*, 466(7307):756–760.

Feng, J., Li, W., and Jiang, T. (2010). Inference of isoforms from short sequence reads. In *Research in Computational Molecular Biology*, pages 138–157. Springer.

Guttman, M., Garber, M., Levin, J., Donaghey, J., Robinson, J., Adiconis, X., Fan, L., Koziol, M., Gnirke, A., Nusbaum, C., et al. (2010). Ab initio reconstruction of cell type-specific transcriptomes in mouse reveals the conserved multi-exonic structure of lincrnas. *Nature biotechnology*, 28(5):503–510.

Karolchik, D., Kuhn, R., Baertsch, R., Barber, G., Clawson, H., Diekhans, M., Giardine, B., Harte, R., Hinrichs, A., Hsu, F., et al. (2008). The ucsc genome browser database: 2008 update. *Nucleic acids research*, 36(suppl 1):D773–D779.

Kawrykow, A., Roumanis, G., Kam, A., Kwak, D., Leung, C., Wu, C., Zarour, E., Sarmenta, L., Blanchette, M., and Waldispühl, J. (2012). Phylo: a citizen science approach for improving multiple sequence alignment. *PloS one*, 7(3):e31362.

Li, W., Feng, J., and Jiang, T. (2011). Isolasso: a lasso regression approach to rna-seq based transcriptome assembly. *Journal of Computational Biology*, 18(11):1693–1707.

Pan, Q., Shai, O., Lee, L., Frey, B., and Blencowe, B. (2008). Deep surveying of alternative splicing complexity in the human transcriptome by high-throughput sequencing. *Nature genetics*, 40(12):1413–1415.

Sammeth, M., Lacroix, V., and Ribeca, P. (2010). The flux simulator. Available at: http://flux.sammeth.net. Accessed October 22, 2012.

Trapnell, C., Pachter, L., and Salzberg, S. (2009). Tophat: discovering splice junctions with rna-seq. *Bioinformatics*, 25(9):1105–1111.

Trapnell, C., Williams, B., Pertea, G., Mortazavi, A., Kwan, G., Van Baren, M., Salzberg, S., Wold, B., and Pachter, L. (2010). Transcript assembly and quantification by rna-seq reveals unannotated transcripts and isoform switching during cell differentiation. *Nature biotechnology*, 28(5):511–515.