Visualization of the invertebrate chordate Ciona intestinalis genome using a custom genome browser

Jerry S. Chen and Robert W. Zeller

May 2010

Publication Number: CSRCR2010-03

Computational Science & Engineering Faculty and Students Research Articles

Database Powered by the Computational Science Research Center Computing Group

COMPUTATIONAL SCIENCE & ENGINEERING

SAN DIEGO STATE UNIVERSITY

Computational Science Research Center
College of Sciences
5500 Campanile Drive
San Diego, CA 92182-1245
(619) 594-3430

Visualization of the invertebrate chordate Ciona intestinalis genome using a custom genome browser

Jerry S Chen[*,1], *Robert W Zeller* [1]
[1] *Computational Science Research Center and Department of Biology*
San Diego State University, 5500 Campanile Drive, San Diego, CA 92182
[*] *Corresponding author: jchen@alumni.caltech.edu*

Abstract: With the sequencing of hundreds of genomes and the advent of high-throughput technologies able to produce high volumes of genomic information at increasingly faster rates and cheaper costs, there is a vital need for proper visualization tools for analyzing large amounts of genomic data. In this report, we implement a genome visualization tool for the model organism Ciona intestinalis. The tool is a genome browser that allows us to visualize a graphical representation of any gene in its native location within the Ciona genome. By default, each gene is displayed along with useful information such as underlying DNA content, corresponding protein sequence and location of coding regions. Our genome browser is flexible, allowing many display options, and expandable, allowing a user to upload and display additional genomic information such as conservation plots and gene expression data. The base genome browser is based on our own update of the Ciona gene models, which we also describe in this report. The Ciona intestinalis genome browser will be a vital tool for our genomics studies on this model organism.

Keywords: genome browser, Ciona intestinalis, genome visualization

1 Introduction

Why we need genome visualization tools
With the incredible volume of sequencing and genomic data that is being produced today, there is a need for proper visualization tools. To get a sense for the amount of genomic data being generated today, consider the case of our own genome. The haploid human genome contains approximately 3 billion base pairs. When sequencing a genome, covering the genome several times over is required to produce statistically reliable sequence data. The first full-scale sequencing of the human genome covered the genome eight times (8x coverage) [1]. Recent human sequencing efforts have scaled up to 36x coverage [2]. In addition, biologists will often collect sequence data from several individuals within a given species in order to account for the significant degree of inter-species variation (and even to look specifically for variation). Early sequencing of the genome [1] compiled DNA from five individuals. In these early efforts they found 2.1 million (0.1%) single-nucleotide polymorphisms (SNPs), which are variations of a single nucleotide at an identical genomic location between two individuals. Add in the vast amounts of insertions, deletions, inversions and other modifications that occur from individual to individual (for example, due to viruses and carcinogens), one can see why it is not sufficient to only produce a single copy of a genome multiple rounds of sequencing efforts are needed. To be able to analyze this variation, one needs visualization tools to compare the differences across multiple sequence runs.

In addition to raw sequence data, many technologies are being produced that generate enormous volumes of associated genomic data. For example, microarrays are commonly used to produce large-scale gene expression data. The human genome contains 25,000 protein-coding genes. Factor in the use of microarrays in capturing expression data over several time points [3-5], and again one has the need for visualization tools to aid in the subsequent data analysis. More recently, a technology called ChIP-SEQ [6-8] has been developed which attempts to comprehensively find sequence regions where a transcription factor protein of interest binds to genomic DNA. The visualization of ChIP-SEQ data with respect to the entire genome is essential for knowing the location of the binding sites and of the surrounding genomic context.

Why we need a Ciona intestinalis genome browser
In this report, we describe the implementation of a genome browser for the model organism Ciona intestinalis. The main reason for using C.intestinalis for genomic studies is that it is physiologically and genetically very similar to the human, but has a much smaller (155 million base pairs) and simpler genome that does not contain many of the complexities that make vertebrate studies extremely difficult. C.intestinalis is an invertebrate chordate with 80% of its genes having similar human counterparts. It is the considered the closest invertebrate relative to vertebrates showing similarity of many tissue types making it an ideal model organism for understanding both invertebrate and vertebrate genomics [9, 10]. The genome has been sequenced [11, 12], with over 2 million publicly available expressed-sequence tags (ESTs) available to verify this sequence information. In addition, a wealth of expression and annotation data is currently available, making C.intestinalis a good model organism for genomics studies [9, 10, 13, 14].

Although two genome browsers exist for Ciona (ANISEED and Ghost databases, see below), there is a need to update the browser for full compatibility with the latest genome assembly [12] and for the ability to store gene information in a more natural hierarchal structure. One of the existing browsers housed in the Ciona ANISEED database (http://aniseed-ibdm.univ-mrs.fr/) is based largely on older versions of the genome, and does not allow one to search for genes based on the latest genome assembly. In addition, the associated gene annotation data is outdated. The second existing genome browser at the Kyoto Ghost database (http://hoya.zool.kyoto-u.ac.jp/cgi-bin/gbrowse/kh/), although recently updated with the newest assembly data, lacks desired functionality such as multi-level gene features and updated annotation data. The existence of our own updated C.intestinalis genome browser will ensure that our analysis is based on up-to-date information and will allow us to customize it as necessary to suit our research needs.

2 Methods

Genomic software and data
The Ciona intestinalis KH gene models and KH genome assembly models [12] were obtained from the Ciona Ghost website (http://hoya.zool.kyotou.ac.jp/ download_kh.html). Source code files for the web-based Generic Genome Browser (GBrowse) software were downloaded from the GMOD GBrowse website (http://gmod.org/wiki/GBrowse). The latest production release (version 1.69) was downloaded. Initial installation of GBrowse from source code was performed with the help of a Perl installation script. Apache (version 2.2.11), MySQL (version 14.12) and Perl (version 5.10.0) were all updated and configured for use with GBrowse version 1.69. All software, data and necessary backend modules and programs were stored on a local server running Fedora Core (RedHat) release 10 powered by dual quad-core 2.3 Ghz AMD Opteron processors.

GFF gene model conversion program
The program for converting the GFF2 gene model file into a multi-level feature GFF3 gene model file was written using Python 2.6.4. First, the lengths of each scaffold of the genome assembly were calculated and used to create a chromosome feature for each scaffold. Next, a gene feature was created for each gene in the genome. This was done by parsing the old GFF2 gene model file for the widest start and stop coordinates for each gene, considering all alternative transcripts. The widest start and stop coordinates comprised the boundaries for each gene. Next, the existing mRNA, 5UTR, CDS and 3UTR feature information for each transcript in the old GFF2 file was converted into a GFF3-compatible format. This included adding parent tags to create a multi-level hierarchy of features: at the top level is the gene feature; the second level is the mRNA feature; and the third (bottom) level are the five_prime_utr, CDS and three_prime_utr features. Finally, all other feature tags were updated for compatibility with the GFF3 specification. The associated GBrowse configuration file was then updated with an aggregators tag that reflects the hierarchy in the GFF3 file.

3 Results

We have implemented and configured the latest production release of GBrowse (ver. 1.69) genome browser software on a local Linux server. One of the key functionalities of the latest GBrowse release is the allowance of multi-level nesting of genomic features using GFF3-format files. Existing genome browsers for the Ciona intestinalis genome are based on older versions of GBrowse using GFF2-format files that do not allow multi-level nesting. We wrote a program to convert the GFF2 gene model file that exists in the ANISEED Ciona intestinalis database (http://aniseed-ibdm.univ-mrs.fr/) into a GFF3 gene model file that adds multi-level genomic feature information (see Materials and Methods). Specifically, in our gene model file we have a three-level nesting of features for each gene: the first (top) level is the gene feature; the second level is the mRNA feature; and the third level contains three_prime_utr, five_prime_utr and CDS features. Figure 1 shows an example of how this information is stored in the gene model file and how this data is finally displayed on the genome browser. Multi-level feature nesting allows a more biologically relevant organization of genomic data which is not found in previous Ciona intestinalis genome browsers.

In the initial phases of using our own browser we used the sequence of only two scaffolds along with only a few genes in our gene model file. We set up our browser initially to read directly from these two files. However, with the C.intestinalis genome being 155 million base pairs in length and with the full gene model file containing information for 24,014 total transcripts corresponding to 15,249 unique genes, direct parsing of these files by the browser became infeasible. Thus, once we verified that our browser was correctly reading these files, it became necessary to load this genomic information into a SQL database. So next we populated our updated GFF3 gene model file along with the FASTA file containing the entire genomic sequence into a SQL database. We successfully integrated this database with the genome browser so that, when a query is input into the browser, the browser on the backend does not have to scan through an entire file but rather only has to process an appropriate SQL query. Integration of an SQL database with the genome browser will be very useful for features that we plan to add in the future. For example, we recently completed annotation of the updated Ciona intestinalis KH gene models [12] based on Gene Ontology (GO) terms . We plan to add these GO gene annotations into our Ciona SQL database. This will allow us to retrieve annotation information directly from the genome browser via backend SQL queries.

The basic interface of our web-based genome browser allows us to perform standard genome visualization. A search bar at the top of the browser window allows us to search for a gene either by gene name or by scaffold coordinates (Figure 2). Annotation tracks may be turned on and off and configured as desired, and new tracks may be uploaded as well. By default, the Ciona browser displays three panels: (1) an overview panel showing scaffold coordinates; (2) a region panel showing the landscape of genes within and around the region of interest; (3) a detailed panel showing genomic information within the region highlighted in the region panel. We wrote and set up our own configuration file to have the detailed panel display five tracks by default: (1) a protein-coding genes track showing gene information; (2) a CDS track showing gene coding regions; (5) a UTR track showing five prime and three prime untranslated regions; (6) a DNA/GC content track showing guanine and cytosine (GC) percentage content at low magnifications, and raw DNA sequence at high magnification; (7) a 6-frame translation track showing protein translation of all six open reading frames of the double-stranded DNA (Figure 2). A scroll/zoom panel is provided at the top with arrow keys that allow the user to scroll through the genome, and zoom-in/zoom-out keys along with a pull-down option window that allow the user to zoom to almost any desired magnification (Figure 3).

Visualization is also powerful tool for gaining qualitative information that would otherwise be difficult by raw parsing of data files. For example, the browser allows us to quickly determine the number of alternative transcripts for a particular gene and view differences between them by simple visual comparison of respective gene plots (Figure 2). By zooming in we can view actual genomic sequence on both DNA strands and six-frame protein translations of gene sequence (Figure 3). Being able to view the raw DNA sequence in line with other features such as exon and intron locations can allow us to explore questions such as, What is the nucleotide composition

at exon-intron junctions? What differences exist between alternative transcripts? In fact, we are currently using the browser to identify important differences between alternative transcripts for specific microRNA targets [15]. In the future we plan to add gene operon information so that one can quickly determine if a gene belongs to a particular operon. Also, we are concurrently working on building a conservation track to show genomic conservation of Ciona intestinalis with the closely related Ciona savignyi (Chen and Zeller, in progress). The conservation track will facilitate comparative studies between the two species. One can imagine eventually building a tool that will be able to extract all conserved or all non-conserved regions of a particular genomic region of interest simply by highlighting the region of interest and selecting the appropriate option.

4 Conclusion

In this report we have described the implementation and features of a custom genome browser for the model organism Ciona intestinalis. We are currently running the genome browser and associated software on a dual quad-core Linux server. Plans are underway to build a local cluster of machines that will enable us to expand the capabilities of our genome browser by adding related functions such as parallel BLAST and analysis and visualization tools for high-throughput sequence data. In the future we also plan to add more interactive functionality into the browser. For example, we have recently completed a full annotation of the Ciona intestinalis genome based on the most recent genome assembly [12]. The annotation gives a description of the likely functions of each gene. We hope to add a feature whereby one can click on a gene in the viewing window and have a new webpage appear showing a detailed annotation of the gene. Also, an option will be provided whereby one can view information such as ideal primers to use when cloning out the gene or for doing in-situ- hybridization. In summary, the custom genome browser we have implemented, along with the updated gene model file and gene annotation, will be an important visualization tool for doing genetics studies using Ciona intestinalis.

5 References

1. Venter, J.C., et al., The sequence of the human genome. Science, 2001. 291(5507): p. 1304-51.

2. Wang, J., et al., The diploid genome sequence of an Asian individual. Nature, 2008. 456(7218): p. 60-5.

3. Spellman, P.T., et al., Comprehensive identification of cell cycle-regulated genes of the yeast Saccharomyces cerevisiae by microarray hybridization. Mol Biol Cell, 1998. 9(12): p. 3273-97.

4. Chen, J., P Paolini, Fourier analysis of time course microarrays and its relevance to gene expression dynamics. ACSESS Proceedings, 2008.

5. Shedden, K. and S. Cooper, Analysis of cell-cycle gene expression in Saccharomyces cerevisiae using microarrays and multiple synchronization methods. Nucleic Acids Res, 2002. 30(13): p. 2920-9.

6. Fields, S., Molecular biology. Site-seeing by sequencing. Science, 2007. 316(5830): p. 1441-2.

7. Jothi, R., et al., Genome-wide identification of in vivo protein-DNA binding sites from ChIP-Seq data. Nucleic Acids Res, 2008. 36(16): p. 5221-31.

8. Johnson, D.S., et al., Genome-wide mapping of in vivo protein-DNA interactions. Science, 2007. 316(5830): p. 1497-502.

9. Passamaneck, Y.J. and A. Di Gregorio, Ciona intestinalis: chordate development made simple. Dev Dyn, 2005. 233(1): p. 1-19.

10. Donoghue, P.M.A.P., The Evolutionary Emergence of Vertebrates From Among Their Spineless Relatives. Evo Edu Outreach, 2009.

11. Dehal, P., et al., The draft genome of Ciona intestinalis: insights into chordate and vertebrate origins. Science, 2002. 298(5601): p. 2157-67.

12. Satou, Y., et al., Improved genome assembly and evidence-based global gene model set for the chordate Ciona intestinalis: new insight into intron and operon populations. Genome Biol, 2008. 9(10): p. R152.

13. Shi, W., M. Levine, and B. Davidson, Unraveling genomic regulatory networks in the simple chordate, Ciona intestinalis. Genome Res, 2005. 15(12): p. 1668-74.

14. Chen, J.S., RW Zeller, Updated annotation of the Ciona intestinalis genome and its application in operon analysis. Unpublished (in progress).

15. Chen, J., RW Zeller, Regulation of gene expression by the microRNA miR-124 in the developing nervous system of C.intestinalis. ACSESS Proceedings, 2009.

6 Figures

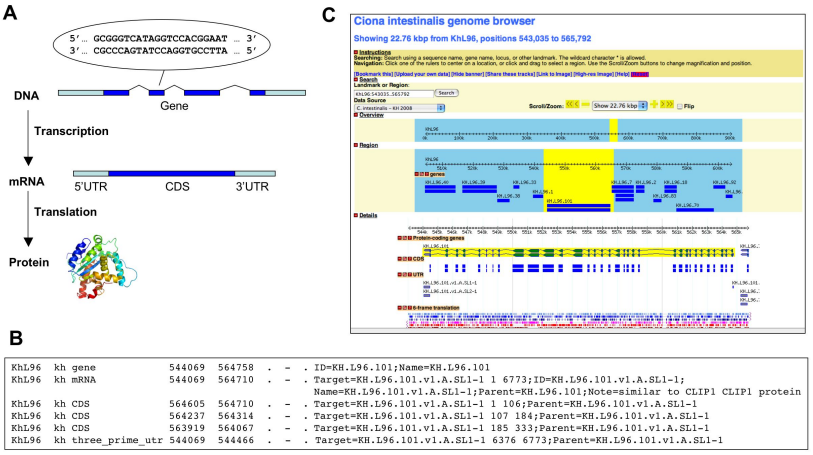

Fig. 1: Our genome browser has the capability of displaying multi-level features, which is useful for displaying genomic information such as a typical mRNA transcript comprised of three sub-regions. (A) When activated, a gene is transcribed into a messenger RNA (mRNA) transcript. The mRNA transcript can be broken down into three regions: a 5 prime untranslated region (5UTR), a 3 prime untranslated region (3UTR) and a coding sequence (CDS). The coding sequence is translated into a functional protein. (B) Example of multi-level feature data in the GFF3 gene model file. Each line of a GFF3 gene model file consists of nine distinct columns which each contain necessary genomic information so that the genome browser can probably display the gene. The GFF3 model file includes a parent tag whereby multi-level features can be stored. In (B), notice the parent hierarchy between gene (top level), mRNA (middle level) and CDS/3UTR (bottom level). The parent tag in the mRNA line refers to the gene ID (KH.L96.101), and the parent tag of the CDS and 3UTR lines refer to the ID of the mRNA (KH.L96.101.v1.SL1-1). (C) Display of this feature data within the web-based GBrowse genome browser. The 5UTR is the gray region to the left of the coding sequence, the coding sequence (CDS) is shaded in blue, and the 3UTR is the gray region to the right.

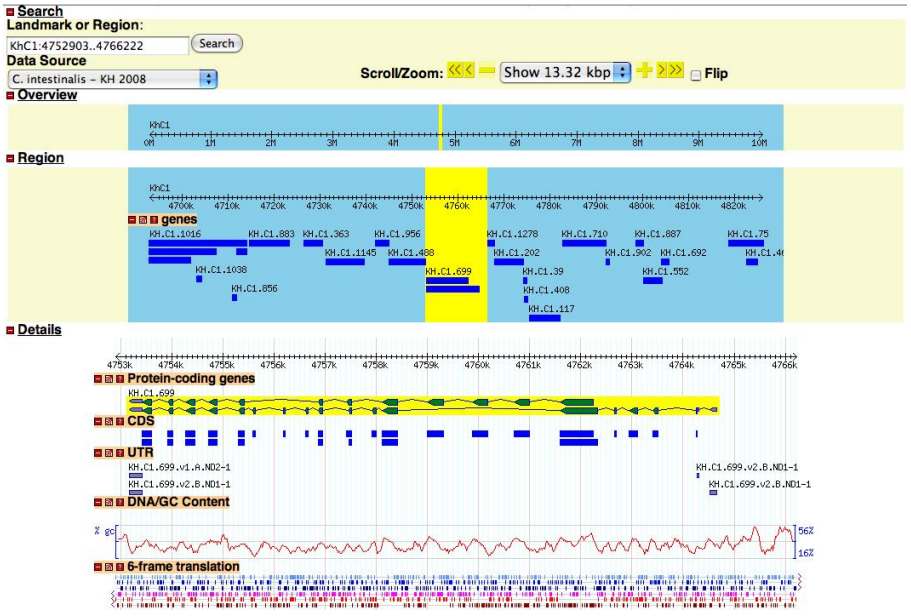

Fig. 2: The default genome browser window shows an overview panel with the scaffold coordinates, a region panel showing a landscape of the genes around and within the field of view, and five annotation tracks specific to the highlighted region of interest: a protein-coding genes track showing exon (green arrows), intron (jagged lines), and untranslated region (gray arrows) for genes; a CDS track showing gene coding regions (blue boxes); a UTR track showing untranslated regions (gray boxes); a DNA/GC content track showing the percentage of guanine and cytosine nucleotides across the region; and a 6-frame translation track showing, when zoomed in, the protein translation of the underlying DNA sequence for all six reading frames. Note that for the particular gene in view (KH.C1.699), two alternative transcripts exist. Notice that one of the transcripts in comparison (KH.C1.699.v2.B.ND1-1) contains 4 more exons and a 5UTR. The C.intestinalis genome browser allows us to quickly visualize such differences in alternative transcripts. The search bar at the top allows the user to search for genes either by scaffold coordinates (shown) or by gene ID (such as KH.C1.699 for this case). Finally, the scroll/zoom toolbar at the top-right allows the user to adjust the location and magnification of the highlighted window.

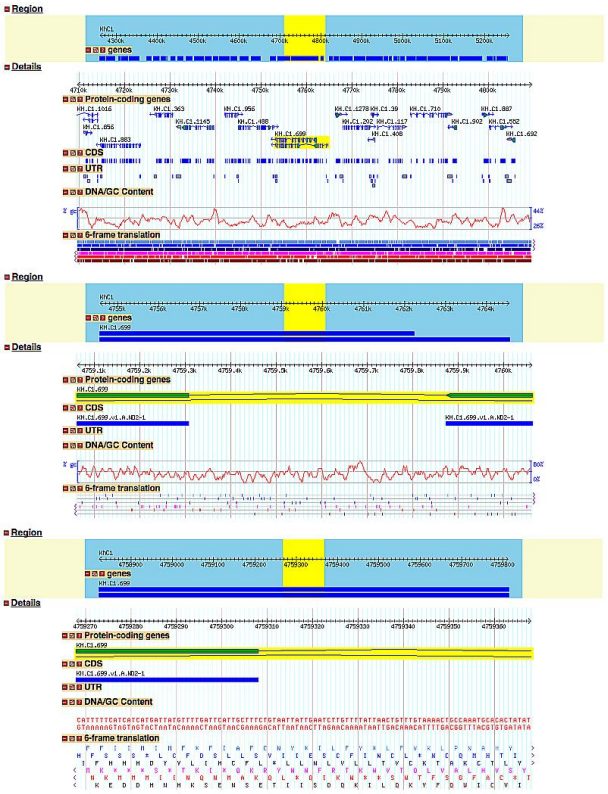

Fig. 3: The three panels show different magnifications: 10 kb window (top), 1 kb (middle) and 100 bp (bottom) - which are typically used for multi-gene, single-gene and nucleotide level viewing, respectively. Notice that when zoomed into a 100 bp window, the DNA/GC content and 6-frame translation tracks show the DNA and translated protein sequences, respectively.