TJHSST Senior Research Project
SyntenyChecker: quality control from syntenic regions
2007-2008

Edward Stallknecht Rice

October 10, 2007

Abstract

When sequencing or mapping a genome, many steps of quality control are necessary to insure the accuracy of the sequence or map. One such control often used involves comparing the sequence or map in question to a previously-created, well-established sequence or map of a different but closely-related organism. Certain regions of genes are closely conserved when organisms diverge through evolution and thus are near-identical on related organisms. Finding these regions on both genomes and comparing the order of genes on each can therefore be used to find potential errors on the genome in question. Currently, this process is usually performed either by hand or by a program written for a specific comparison which cannot be easily used for other comparisons. The aim of this project is to write a highly-configurable program which uses a generalized approach to finding potential errors and can therefore be used as a quality control on any sequence or map being compared to any other sequence or map.

1 Introduction - Purpose and Scope

The subject of this project is the use of syntenic regions between two related organisms to find potential errors in a sequence or map. The goal is to create a highly-configurable program which uses a generalized approach to finding potential errors and can therefore be used as a quality control on any sequence or map being compared to any other sequence or map. This project is worth doing because the current methods of using synteny as a quality control are inconsistent and time-consuming. It is a good topic for the Computer Systems Lab because it applies computer science to a real scientific problem. Scientists making maps or sequences for an organism ought to be interested in the results of the project because it will provide a fast and consistent quality control.

This project will require research on current methods of using synteny as a quality control as well as algorithms used to find regions of synteny. I have already done this research. Preexisting maps are required to test the program. Some (the previously published, accepted maps) are freely available on NCBI's website, while others (the unpublished, more erroneous ones) are available to me because I have previously contributed to their creation. I have already obtained and acquainted myself with all the maps I will need. While creating a complete and documented application usable in any circumstance and ready for distribution is outside my ability considering the time constraints of the semester, I am confident that I will be able to create a program that will at least work on a horse map vs. human sequence comparison and have a structure that will allow it to be easily modified and improved by others until it is ready for distribution.

2 Background

One of the fundamental tasks in bioinformatics is the creation of genome maps of different organism. A map of a genome is a sequential list of markers on each chromosome in the genome along with estimates of distances between markers. Each marker is a unique segment of DNA that acts as a landmark. A marker can, but need not, be a gene. Many approaches to creating maps exist, including genetic mapping and radiation hybdrid, both of which require an initial physical experiment followed by complex calculations usually performed by a computer program [Schaffer, 2006, Applegate et al., 2006].

When creating a map of a genome, it is necessary to perform multiple tests on the output of such a computer program in order to ensure its accuracy and find potential problems. One of the tests often used to perform this quality control is comparative gene mapping. Comparative gene mapping involves comparing the map of the genome in question to an existing published map of a different but closely-related organism using knowledge about synteny across homologous markers [O'Brien and Graves, 1990].

Synteny is the relationship between similar genes on different organisms. Generally, certain properties of the order of similar genes on different organisms are conserved. For example, linked groups of genes on a mammal tend to be linked on other mammals [Rettenberger et al., 1995].

Comparative gene mapping is not a new concept, and it has been used as a quality control in the assembly or mapping of a variety of genomes. A committee at the Human Genome Meeting in 1990 used it to measure the progress being made on maps for certain mammals by comparing them to human and mouse genetic maps [O'Brien and Graves, 1990]. The time required to create a physical map of the mouse genome in 2002 was greatly reduced by comparative genomics [Gregory et al., 2002]. In 2007, a map of bovine chromosome 14 was checked by comparing it to human chromosome 8 [Marques et al., 2007]. Comparative gene mapping has also been used with rice and sorghum [Bowers et al., 2005].

Several applications currently exist which find regions of synteny between either two marker maps or two assembled sequences, but do not use this information for the purpose of finding potential errors. ADHoRe [Vandepoele et al., 2002], DAGchainer [Haas et al., 2004], PatternHunter [Ma et al., 2002], BLASTZ [Schwartz et al., 2003], and SyMAP [Soderlund et al., 2006] are examples. Other applications, such as Carthagene [Faraut et al., 2007], AMOS [Pop et al., 2004], and the Atlas Genome Assembly System [Havlak et al., 2004], use comparative genomics to make some decisions in their processes.

DAGchainer uses a novel algorithm involving graph theory and dynamic programming to find syntenic regions between two sequences. It only outputs the information and does not use it to make decisions. It cannot compare two marker maps or a sequence and a marker map because it relies on BLAST protein alignment, which can only be performed on sequenced genomes, to find homologous regions [Haas et al., 2004]. However, this algorithm is still notable because finding syntenic regions is the first step of using comparative gene mapping as a quality control, and the algorithm has the potential to be generalized and used in other situations besides the narrow set of inputs DAGchainer can take.

The algorithm used by DAGchainer to find syntenic regions creates a directed acyclic graph of the data [Haas et al., 2004]. A directed acyclic graph is a graph in which all edges have a direction and cycles are impossible, so there is a finite number of paths in the graph [Cormen et al., 1992]. In this implementation, each node N represents a pair of homologous genes n1 and n2 where n1 is a gene on the first genome and n2 is a gene on the second genome. The pair (n1,n2) is used to represent its node. The node (a1,a2) preceeds node (b1,b2) if and only if a1 preceeds b1 in the first genome and a2 preceeds b2 in the second genome. Each node has a score and each edge has a cost [Haas et al., 2004]. Dynamic programming can be used to find the maximum-cost path in the graph [Sedgewick, 1983], which represents the most probable region of synteny between the two genomes [Haas et al., 2004].

3 Procedures

I have already performed most of the required research and implemented and tested the DAGchainer algorithm to find regions of synteny. My focus is now to find errors in the regions of synteny. I have a plan about various checks I can do on the data to find potential errors. For example, interleaving regions of synteny (e.g. for marker order ABCDEF on genome 1, ACE are found to be a different region of synteny than BDF) usually indicate a problem, so finding and reporting these regions is a good check on the output. My tasks will be thinking of different indicators of categories of failures of synteny, such as interleaving regions, creating diagnostic algorithms that can find these indicators, implementing these algorithms, and giving outputs for each diagnostic which report the problem and give information about it in a way that will help the user fix the problem as easily as possible. The process of gradually adding diagnostics will evolve as it happens, guided by what problems the previously-implemented diagnostics find and do not find, and therefore cannot be planned in detail at this point.

The only resources I need are a computer with a text editor and various language interpreters. I will need to use python, perl, and bash to create this program. I will construct visuals representing overlapping erroneous gene segments to show the types of errors my program is finding. I will also construct models of simple graphs to show the data structure setup of my program. The input data necessary are maps of various mammalian genomes, some of which are freely available from the NCBI website. I will use a horse map which has not yet been published or put on the NCBI site because it still contains errors and therefore is more useful for testing than a published map which contains few errors. This map is available to me because I worked on it last summer and NCBI is involved in its creation. I will test my program by running data through it for which I know where the errors are and comparing the known errors to the errors found by the program.

3.1 Software

Computer languages I'll use:

1. Python
2. Perl
3. Bash Scripts

Development environment:

1. Bash command line shell
2. vim
3. Command line interpreters for python and perl

3.2 Algorithms/Data Structures

I'll be using the following algorithms/data structures, in addition to designing my own:

1. Dynamic programming
2. Directed acyclic graphs
3. Lists

4 Schedule

During the first half of the semester, I will focus on writing a program that can find potential errors in the horse map by comparing it to the human sequence in order to have interesting output to report for the Intel Science Talent Search.

During the second half of the semester, I will focus on improving the program to make it more generalized and configurable, documenting it well, and presenting it with a paper, oral presentation, etc.

5 Expected Results

I expect to create a program that will find regions of potential error in a horse map by comparing it to the human sequence. I will present the final results by showing example outputs of the program, and graphically representing what they mean. These results will contribute to the work of future researchers by giving them a consistent means of performing syntenic quality control on their data.

References

[Applegate et al., 2006] Applegate, D. L., Bixby, R. E., Chvatal, V., and Cook, W. J. (2006). *The Traveling Salesman Problem: A computational study*, page 63. Princeton University Press.

[Bowers et al., 2005] Bowers, J. E., Arias, M. A., Asher, R., Avise, J. A., and et al (2005). Comparative physical mapping links conservation of microsynteny to chromosome structure and recombination in grasses. *Proceedings of the National Academy of Science*, 102(37):13206–13211.

[Cormen et al., 1992] Cormen, T. H., Leiserson, C. E., and Rivest, R. L. (1992). *Introduction to Algorithms*, pages 86–90. McGraw-Hill.

[Faraut et al., 2007] Faraut, T., de Givry, S., Chabrier, P., Derrien, T., Galibert, F., Hitte, C., and Schiex, T. (2007). A comparative genome approach to marker ordering. *Bioinformatics*, 23(2):e50–e56.

[Gregory et al., 2002] Gregory, S. G., Sekhon, M., Schein, J., Zhao, S., Osoegawa, K., Scott, C. E., Evans, R. S., Burridge, P. W., Cox, T. V., Fox, C. A., and et al (2002). A physical map of the mouse genome. *Nature*, 418(6899):743–750.

[Haas et al., 2004] Haas, B. J., Delcher, A. L., Wortman, J. R., and Salzburg, S. L. (2004). DAGchainer: a tool for mining segmental genome duplications and synteny. *Bioinformatics*, 20(18):3643–3646.

[Havlak et al., 2004] Havlak, P., Chen, R., Durbin, K. J., Egan, A., Ren, Y., Song, X.-Z., Weinstock, G. M., and Gibbs, R. A. (2004). The atlas genome assembly system. *Genome Research*, 14(4):721–732.

[Ma et al., 2002] Ma, B., Tromp, J., and Li, M. (2002). PatternHunter: faster and more sensitive homology search. *Bioinformatics*, 18(3):440–445.

[Marques et al., 2007] Marques, E., de Givry, S., Stothard, P., Murdoch, B., Wang, Z., Womack, J., and Moore, S. S. (2007). A high resolution radiation hybrid map of bovine chromosome 14 identifies scaffold rearrangement in the latst bovine assembly. *BMC Genomics*, 8:254.

[O'Brien and Graves, 1990] O'Brien, S. and Graves, J. M. (1990). Report of the committee on comparative gene mapping. *Cytogenetics and Cell Genetics*, 55(1-4):406–433.

[Pop et al., 2004] Pop, M., Phillippy, A., Delcher, A. L., and Salzberg, S. L. (2004). Comparative genome assembly. *Briefings in Bioinformatics*, 5(3):237–248.

[Rettenberger et al., 1995] Rettenberger, G., Klett, C., Zechner, U., Kunz, J., Vogel, W., and Hameister, H. (1995). Visualization of the conservation of synteny between humans and pigs by heterologous chromosomal painting. *Genomics*, 26:372–378.

[Schaffer, 2006] Schaffer, A. (2006). *Handbook of Computational Molecular Biology*, chapter Chapter 17: Human Genetic Linkage Analysis. Chapman and Hall.

[Schwartz et al., 2003] Schwartz, S., Kent, W. J., Smit, A., Zhang, Z., Baertsch, R., Hardison, R. C., Haussler, D., and Miller, W. (2003). Human-mouse alignments with blastz. *Genome Research*, 13(1):103–107.

[Sedgewick, 1983] Sedgewick, R. (1983). *Algorithms*, pages 483–494. Addison-Wesley.

[Soderlund et al., 2006] Soderlund, C., Nelson, W., Shoemaker, A., and Paterson, A. (2006). SyMAP: A system for discovering and viewing syntenic regions of FPC maps. *Genome Research*, 16(9):1159–1168.

[Vandepoele et al., 2002] Vandepoele, K., Saeys, Y., Simillion, C., Raes, J., and de Peer, Y. V. (2002). The automatic detection of homologous regions (ADHoRe) and its application to microcolinearity between arabidopsis and rice. *Genome Research*, 12(11):1792–1801.