



Plant genome annotation using bioinformatics

ghorbani mandolakani Hossein^{**}, khodarahmi manouchehr
darvish farrokh, taeb mohammad

ghorbani24sma@yahoo.com

islamic azad university of science and research branch

Abstract

Large amounts of genome sequence data are available and much more will become available in the near future. A DNA sequence alone has, however, limited use. Genome annotation is required to assign biological interpretation to the DNA sequence. The aim of genome annotation is to describe the biological function of every single nucleotide during the life span of an organism. This requires the help of bioinformatics. Bioinformatics is a multidisciplinary approach that combines several areas of expertise in the automated analysis of bio-molecular data. To achieve the goal of proper annotation of a genome, close cooperation between bioinformaticians and (genome) biologists is required at several levels. Communication in bioinformatics for genome annotation is a major challenge on several levels: communication between computers and communication between researchers are both at stake, as well as the communication between computers and human beings. The global bioinformatics community is moving towards a (web) service-based infrastructure. The second major issue in bioinformatics and genome annotation is the quality of annotation data. Most annotation depends in some way or another on previous annotations. Obviously the quality of such prediction relies on the quality of the underlying data. The issue of error propagation is an important issue in the field of genome annotation and needs much future attention.

Key words: bioinformatics, genom annotation, DNA sequence



Introduction

Genome annotation is the process of assigning biological interpretation to a DNA sequence. A DNA sequence, as a string of nucleotides, has limited use in application and research. Various analyses are required to assign biological interpretation to a DNA sequence. The goal of genome annotation is to describe the function of every single nucleotide, in any cell or cell compartment, during the reproduction and the life span of an organism. The need for bioinformatics in genome annotation became evident upon the completion of the first genomes (4, 17). Bioinformatics is the multidisciplinary approach that combines, amongst others, molecular biology, information technology, mathematics and statistics in the automated analysis of biomolecular data. The term 'bioinformatics' appeared in scientific literature somewhere in the 1980's. It has its roots in fields as theoretical and computational biology. Nowadays over 300 genomes have been sequenced (6). The annotation of a single genome is an intensive task, from both a computational as a biological perspective, confirming the importance of bioinformatics in genome annotation. A genome is not annotated by bioinformaticians alone, but in close cooperation with biologists. Biologists deliver the raw data and biological context for the annotation of a sequence. Often this results in new hypotheses that lead to more experiments by both biologists and bioinformaticians and ultimately contribute to the advancement of biological understanding.

Computational genome annotation

Complete and in-depth annotation of a genome requires the application of many different software tools. The number of separate computational executions that needs to be performed can be extremely high, generating complex data flows and requiring large amounts of CPU-time as well as data storage capacity. Data must always be in a form that can intelligibly be presented to and used by researchers.

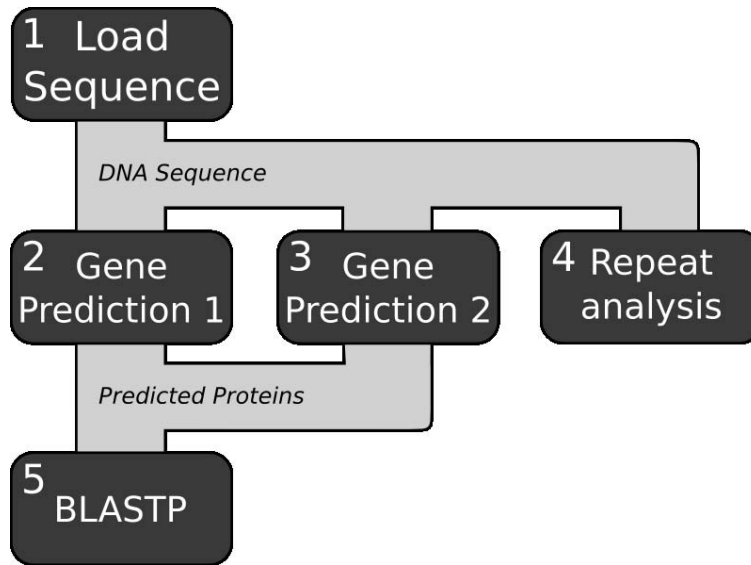


Figure 1: Schematic representation of a small annotation project in the form of a pipeline. The input sequences (1) are analyzed by two gene prediction tools (2&3) and an analysis of the repetitive regions (4). The predicted proteins are subsequently compared against a protein database by BLASTP (5).

Workflow management

A simple genome annotation project will consist of several gene predictors, a tool to consolidate gene predictions, a repeat-finding tool, and a BLAST analysis on predicted genes (Figure 1). This set of tools and the order in which they should be executed is called a “pipeline”. Execution of the pipeline will involve a large number of separate jobs in which the output of one job serves as input for a subsequent job. The results of all analyses need to be stored in a database and made available to an end user.

On the scale of a complete genome analysis, it will quickly become impossible to perform all analyses manually. Computational pipelines require the use of specialized software that schedules and keeps track of jobs as well as of the creation and storage of data and results. Several systems are available for this kind of pipeline or workflow management in a genome annotation environment(12, 15, 18). Any pipeline system that is able to handle complex, elaborate and configurable pipelines requires extensive computing. There are several possibilities for the scaling of computing capacity. One is to use a single multiprocessor system but these tend to be expensive and difficult to scale. A second, more scalable, solution is to use a cluster of distributed small to midrange systems. In 1993, the Beowulf project (2) was the first to implement such a cluster using commodity hardware and brought such systems into the reach of many. Many different systems have since been developed. These can be divided roughly into two types; clusters that operate as a



single multiprocessor computer and clusters in which separate computers (nodes) are directed by a central control unit. Clusters of the first type simulate a multiprocessor computer by using a middle layer that handles communication between the nodes. This type of cluster is particularly designed to handle large, computationally demanding jobs but requires specially adapted software. Two well known examples are PVM (14) and MPI (10). HMMer (8) and BLAST (11) are examples of bioinformatics applications that are able to work on PVM or MPI clusters, respectively. In the second type of clusters, parallel computing is achieved through distribution of separate jobs by the central control unit (or “master”) over independent nodes (or “slaves”) that execute the job and return the results back to the master. This type of clusters does not require specially adapted software and are well-suited for the execution of large numbers of jobs that require relatively little computing power. Common job management software for cluster computing includes Sun Grid Engine (SGE) (21), openPBS (13) and Condor (3). The latter is aimed at heterogeneous, non-dedicated, hardware and is able to run on, for example, idle office desktops. A complete Linux distribution that includes different job management is Rocks (16). If computational facilities are distributed over different physical locations, it is commonly named a “grid”. Implementation of a grid can provide a level of throughput which is not achievable for a single cluster of super-computer. A common used toolkit for developing grids is Globus (7) but Condor and SGE also contain grid-like features. The choice for a system depends strongly on the requirements of an application. As most genome annotation pipelines need to execute large numbers of separate applications the second type is in most cases better suited. Especially as several of the mentioned solutions are able to run a “sub-cluster” of the first type.

Data management and data exchange

A second aspect in automated execution of genome annotation pipelines is concerned with data management: exchange and storage. The various bioinformatics tools available use a wide variety of data input and output formats. This hampers communication between separate analyses. One tool may deliver an output that can not be used as input for the next tool without a translation. A notorious example is the output of a BLAST analysis (1) which is a human readable text document that has undergone many changes. Small changes which are easy to understand for a human usually breaks software attempting to automate a task. It is extremely difficult to design a single, generic data format, due to the inherent diversity of biomolecular data types (reviewed in Stein (20)). However, a number of solutions have been proposed for the standardization of genome annotation data, including the General Feature Format (GFF) (5) and XML based BioMOBY (23, 22). GFF is widely used in the bioinformatics community, although it is limited in scope and difficult to extend. BioMOBY provides a more flexible solution for the exchange of biomolecular data. BioMOBY distinguishes itself by not attempting to describe data, but describes how to describe data, BioMOBY is a meta-format. This makes the exchange of data formats much easier and as a direct result, the exchange



of data itself. Before data can be exchanged however, a decision on the BioMOBY must still be made. Similar problems apply to data storage. Raw output from can be stored as such but this renders the data inaccessible for subsequent inspection and integration. The database underlying the Generic Genome Browser (19) stores GFF data and is thus limited to what GFF can describe. Again, the use of a meta-format such as BioMOBY can help in storing data and making it more accessible.

Visualization

The final step of an annotation program is to present the results of all analyses and biological interpretations in an intelligible and easily accessible form to the biologist. This is a not a trivial task as the amount of data generated is often enormous. Several strategies can be taken to help explore and understand genome annotation data, the best among which is through visualization (“a picture is worth a thousand words”). Among the widely used tools available for the visualization and exploration of an annotated genome are the Generic Genome Browser (Gbrowse) (19) and Ensembl (9). Most visualization tools have been developed as web interfaces to underlying annotation databases. Therefore, they generally perform poorly with respect to interactivity.

Reference

- 1- Altschul et al. Basic local alignment search tool. *J Mol Biol*, 215(3). 1990. 403–10.
- 2- Beowulf website. URL <http://www.beowulf.org/>.
- 3- Condor website. URL <http://www.cs.wisc.edu/condor/>.
- 4- Fleischmann et al. Whole-genome random sequencing and assembly of *Haemophilus influenzae* Rd. *Science*, 269(5223). 1995. 496–512.
- 5- Generic Feature Format website. URL <http://www.sanger.ac.uk/Software/formats/GFF/>.
- 6 - Genomes online database (gold) website. URL <http://genomesonline.org>.
- 7- Globus toolkit website. URL <http://www.globus.org/>.
- 8- HMMer website. URL <http://hmmer.wustl.edu>.
- 9- Hubbard et al. Ensembl. *Nucleic Acids Res*, 33(Database issue): D447–53, 2005.
- 10- Message Passing Interface (mpi) website. URL <http://www-unix.mcs.anl.gov/mpi/>
- 11- mpiBLAST website. URL <http://mpiblast.lanl.gov/>.
- 12- Oinn et al. Taverna: a tool for the composition and enactment of bioinformatics workflows. *Bioinformatics*, 20(17). 2004. 3045–54.
- 13- OpenPBS: Portable Batch System website. URL <http://www.openpbs.org/>.
- 14- Parallel Virtual Machine website. URL [http://www.csm.ornl.gov/pvm/pvm home.html](http://www.csm.ornl.gov/pvm/pvm%20home.html).
- 15- Potter et al. The Ensembl analysis pipeline. *Genome Res*, 14(5). 2004. 934–41.
- 16- Rocks Cluster Distribution website. URL <http://www.rocksclusters.org/>.



- 17- Sanger et al. Nucliotide sequence of bacteriophage phi X174 DNA. *Nature*, 265(5596). 1977. 687-95.
- 18- Shah et al. Pegasys: software for executing and integrating analyses of biological sequences. *BMC Bioinformatics*, 5:40, 2004.
- 19- Stein et al. The generic genome browser: a building block for a model organism system database. *Genome Res*, 12(10). 2002. 1599-610.
- 20- Stein. Integrating biological databases. *Nat Rev Genet*, 4(5). 2003. 337-45.
- 21- Sun Grid Engine website. URL <http://gridengine.sunsource.net/>.
- 22- Wilkinson and Links. BioMOBY: an open source biological web services proposal. *Brief Bioinform*, 3(4). 2002. 331-41.
- 23- Wilkinson et al. BioMOBY successfully integrates distributed heterogeneous bioinformatics Web Services. The PlaNet exemplar case. *Plant Physiol*, 138(1). 2005. 5-17.