The GAIA Software Framework for Genome Annotation

G. Christian Overton, Charles Bailey, Jonathan Crabtree, Mark Gibson,
Stephen Fischer and Jonathan Schug
Center for Bioinformatics
University of Pennsylvania
Philadelphia, Pennsylvania 19104, USA

Abstract

We describe a software framework, GAIA, that supports semi-automated annotation of uncharacterized sequence data. The annotation framework incorporates annotation by data source integration, data analysis, and manual data entry. Components of the system include a configurable, open data analysis pipeline, a relational information storage manager, and Java-based graphical user interfaces. We discuss design decisions and tradeoffs in building such a system, and policies and strategies for producing consistent, uniform, high quality annotation.

1 Introduction

Within the next seven years, the first reference sequence of the human genome will be complete. Achieving this goal will require an average of roughly 2 MB of finished sequence per day of world-wide output. High-throughput, large-scale sequencing efforts now underway in a number of sequencing centers in academia and industry should approach the requisite rate of sequence generation in the next few years. In contrast, automation of data analysis leading to consistent and comprehensive annotation of the sequence has not yet been attained except in restricted cases. Moreover, sequence annotation is not static: to remain current, the sequence must be continually re-visited and re-analyzed in light of improved and new sequence analysis algorithms, and updates to relevant online data sources such as GenBank and SwissProt. Fully automated sequence analysis remains a challenging task. Over the past few years, several groups have assembled systems that produce automatic annotation for sequences from prokaryotes and lower eukaryotes. The MAGPIE system is now widely in use for genome sequencing projects in prokaryotes,[1,2] and GeneQuiz has been used for both prokaryotes and yeast.[3,4] While many of the ideas developed in these systems are quite general, the genomes of higher eukaryotes, and vertebrates in particular, are substantially more difficult to interpret especially as regards tasks such as gene finding.

By sequence annotation, we mean the prediction and archiving of landmarks and biological features (*e.g.* genes), putative biological signals (*e.g.* transcription elements and matrix attachment sites), sequence characteristics (*e.g.* CpG islands and isochores), and gene products as a step in the functional characterization of the sequence. Annotation can be generated through three principal methods: data analysis such as computational gene finding; integration of data, information and knowledge from existing online resources; and encoding data, information and knowledge directly from the literature. Comprehensive analysis will take advantage of all three methods.

We have implemented a software framework, GAIA (Genome Annotation and Information Analysis) designed to explore the issues involved in automating the annotation of the human genome as well as those of other organisms.

2 System Design Criteria

At this stage of development, GAIA is primarily an engineering prototype designed to explore and reveal the issues that will arise in automating annotation of genomic sequence; however, it is also meant to be a practical system at each stage of development that will deliver first minimal and then increasingly comprehensive, high quality and consistent annotation. We began by defining a set of initial engineering requirements that should be satisfied when GAIA is viewed either as an experimental or practical system:

1. Information should be explicitly represented in the annotation database in a machine accessible form (*i.e.* queryable), except where totally infeasible.
2. The strategy defining the annotation process should be recorded as part of a policy statement describing the system, and represented

explicitly in the annotation database as well.

3. The system should support annotation computed on-the-fly where feasible. For example, restriction enzyme patterns or base composition of a region of sequence could be computed dynamically rather than archived.
4. The annotation strategy should be conservative, using a consistent application of well-characterized techniques.
5. The system should support "plug-and-play" where new application software, new data sources and new annotation strategies can be easily incorporated in the system.
6. The system should be designed to expedite the experimental confirmation of computational predictions, especially in high-throughput formats.
7. The system should be able to present data to the user at several levels of resolution, so that an appropriate balance between detail and scope may be selected for different tasks.

3 Implementation

Figure 1 depicts the flow of information through the primary components of the GAIA system. The architecture supports all three modes of annotation — data analysis, data integration, and manual data entry — although at different levels of sophistication. Obviously, only data analysis and data integration could in principle be fully automated. In practice though, data integration requires many decisions about the semantics of the data which make fully automating the process difficult at this time. To the extent that data analysis depends on information made available as the result of data integration, it too cannot yet be fully automated.

Data can be submitted to the annotation engine either from a local file or interactively through the GAIA WWW page.[a] Any of the popular formats for sequence data are acceptable. The raw sequence data along with ancillary information (*e.g.* references, map location, organism) are deposited in the working database. There is no current limit on the size

[a] `http://www.cbil.upenn.edu/gaia`

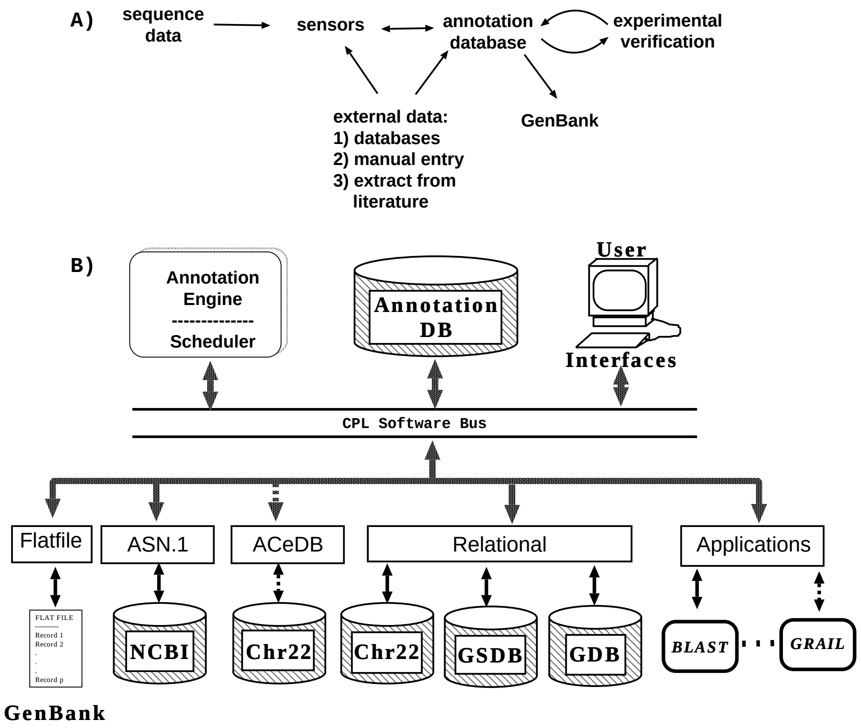


Figure 1: Information flow through system components.

of the input sequence; typically the system handles sequence of 40KB (cosmid inserts) or larger (BAC, P1 inserts).

3.1 Annotation Engine

The annotation engine comprises a series of autonomous components, called sensors, each performing a specific analysis, which communicate with each other via the annotation database. Each sensor may use as input the sequence itself, ancillary data provided when the sequence was deposited, and annotation produced by other sensors. Results are then deposited back into the database, along with a description of the annotation process. In the current version, sensors also use data from

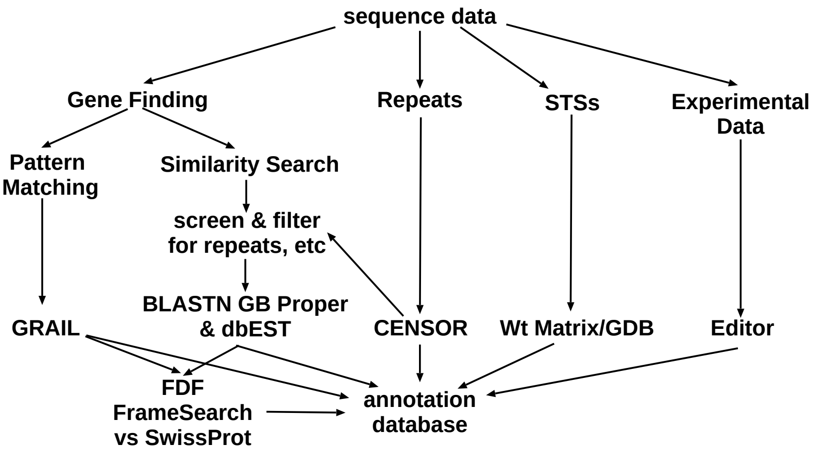


Figure 2: Annotation strategy.

GenBank, dbEST, GenPept, and GDB. The scheduling of different sensors may be done manually, or may be handled by the default scheduler. Currently, this scheduler is a Perl program which simply executes the annotation strategy, while insuring that the order of execution properly resolves dependencies between sensors.

In the present annotation strategy (Figure 2), the sensors are: gene finding by pattern recognition using GRAIL[5]; gene finding by sequence similarity using BLAST[6]; identification and characterization of repetitive elements using CENSOR[7]; identification of STSs recorded in GDB; and characterization of gene products using Framesearch on the Paracel FDF machine (not yet available publically). However, the architecture of the system is open in the sense that any application program or data source can be readily incorporated in the annotation strategy. To illustrate in more detail, the strategy for gene finding by sequence similarity to ESTs is depicted in Figure 3 (details can be found in Bailey *et al.*[8]).

Where there are no dependencies in the annotation strategy, analysis can be carried out in parallel. For example, the repetitive sequence sensor and the STS sensor are entirely independent and can be scheduled to run concurrently. Obviously, this substantially reduces the wall

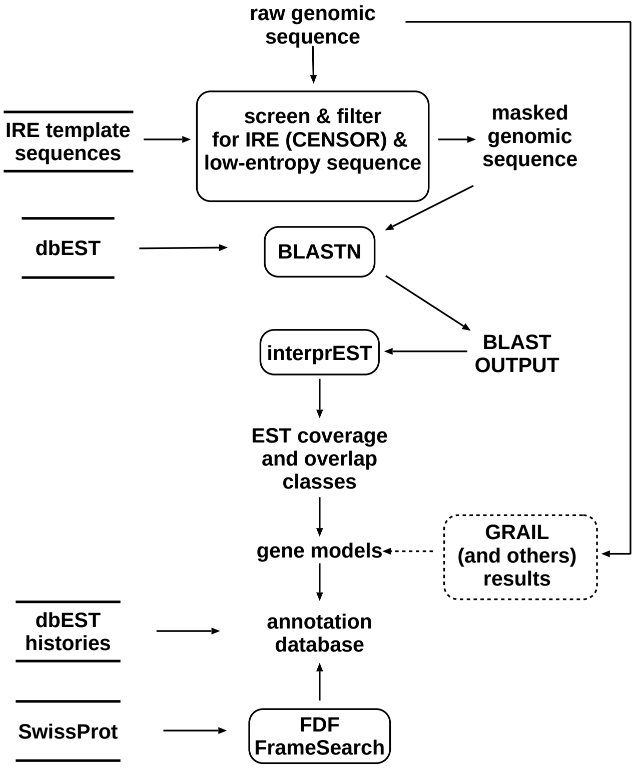


Figure 3: Strategy for gene finding by sequence similarity.

clock time for annotation, especially when special-purpose data analysis hardware such as the Paracel FDF is employed. The rest of the annotation process is carried out on a farm of Sun Unix workstations which includes a six processor E4000 UltraSparc.

3.2 Annotation Database

As annotations are computed by the sensors, they are deposited in the annotation database. The schemas for these data are described in an object-oriented extension of CPL,[9,10] a rich language in which collection types such as sets, bags, and lists as well as record, union, and primitive types can be expressed. CPL objects are currently stored in a Sybase database as text types. This language is also the data exchange format for application software inter-operation on the GAIA "software bus."

The foundation schema for GAIA extends the GSDB relational schema for GenBank along several lines. The goals in further schema development are to (1) reduce and where possible eliminate the free text descriptions and ambiguities which plague current public sequence databases; and (2) simplify the schema conceptually by transforming it into an object-oriented form. We have made several enhancements to the schema:

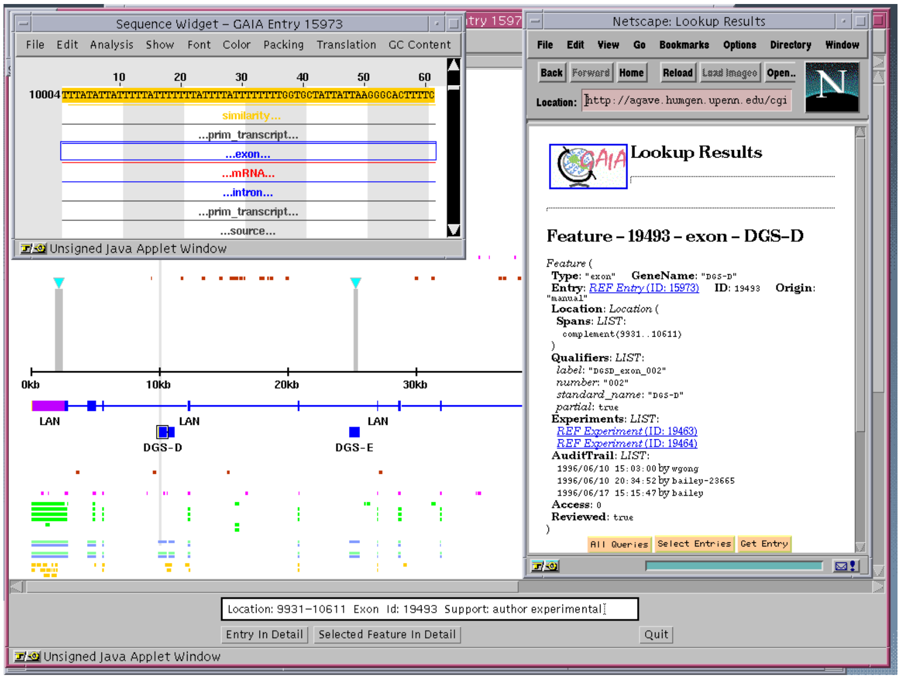
1. The representation of sequence features is guaranteed to be in a canonical form that permits uniform access to all feature types. For example, all of a gene's exons and introns are explicitly represented and thus can be efficiently and exhaustively retrieved from the database.
2. Information about genes is organized around a "transcription unit" which supports the representation of alternative splice forms of a transcript.
3. The method(s) of prediction for each feature along with parameter settings of the application software, the version of the software, the versions of contributing data sources, and the date the analysis was performed are recorded in the database. This is compact, since it is principally done by reference to the annotation strategy information.
4. Experimental method(s) for feature "prediction" are also recorded in the database. A controlled vocabulary for experimental types is being constructed.
5. We are developing a virtual sequence representation, so that a gene and its transcription units can be described without the necessity of reference to a complete and contiguous interval of DNA sequence.
6. Data is versioned. The "current" state of the database is not monotonic, meaning that features can be eliminated from the database if the underlying supporting evidence changes, *e.g.* as a consequence of new versions of an algorithm or updates to contributing databases.

3.3 bioWidget Interface

Once a sequence has been annotated and the information deposited in the annotation database, it can accessed by three methods: direct queries against the CPL objects stored in Sybase, HTML forms, and a Java-based graphical user interface application built with the bioWidget GUI toolkit. Here we will examine only the bioWidget GUI, which is the most intuitive to use. Figure 4 shows a typical display of information for sequence from the DiGeorge Critical Region (DGCR) on human chromosome 22q. Three intercommunicating components, which are created dynamically from information in the annotation database, are used in this application. The map applet displays a schematic of the sequence, with experimental and computational annotation color coded with respect to the method of analysis. The polarity of the annotation is indicated by location above (left-to-right polarity) or below the scale bar. The sequence applet displays sequence information along with a more detailed information on the annotation. The region of the sequence applet visible is indicated as a gray bar in the map widget. Each feature is active, and clicking on it will highlight the feature in both the map and sequence applets. More detailed information on each feature with hot links to WWW pages is displayed in an HTML page linked to the bioWidget applications.

3.4 Initial Data Analysis

GAIA has been used to annotate nearly 2 MB of sequence on human chromosome 22q generated by Bruce Roe (University of Oklahoma, Norman, OK), in collaboration with Marcia Budarf and Beverly Emanuel (Children's Hospital of Philadelphia, Philadelphia, PA). A nearly contiguous region of almost 1.3MB surrounding the DGCR was found to have 25 genes, more than 1400 repetitive elements, and 75 mapped STSs. The density of genes was not uniform, with over half of the genes concentrated in a 200 KB segment identified as the DGCR by genetic and physical mapping. In addition, sequence from human chromosome 7 and mouse have been analyzed through GAIA, and a version is being designed specifically to handle sequence generated for the Arabidopsis


Sequence Widget – GAIA Entry 15973
File Edit Analysis Show Font Color Packing Translation GC Content
10 20 30 40 50 60
10004 TTTATATTATTTTTATTTTTTTATTTTATTTTTTTGGTGCTATTATTAAGGGCACTTTTC
similarity...
...prim_transcript...
...exon...
...mRNA...
...intron...
...prim_transcript...
...source...
Unsigned Java Applet Window
0kb
10kb
20kb
30kb
LAN
LAN
LAN
DGS-D
DGS-E
Netscape: Lookup Results
File Edit View Go Bookmarks Options Directory Window
Back Forward Home Reload Load Images Open..
Location: http://agave.humgen.upenn.edu/cgi
All Queries
Select Entries
Get Entry
Entry In Detail
Selected Feature In Detail
Quit
Unsigned Java Applet Window

Lookup Results

Feature – 19493 – exon – DGS-D

Feature (
Type: "exon" **GeneName**: "DGS-D"
Entry: *REF Entry (ID: 15973)* **ID**: 19493 **Origin**: "manual"
Location: *Location* (
Spans: *LIST*:
complement(9931..10611)
)
Qualifiers: *LIST*:
label: "DGSD_exon_002"
number: "002"
standard_name: "DGS-D"
partial: true
Experiments: *LIST*:
REF Experiment (ID: 19463)
REF Experiment (ID: 19464)
AuditTrail: *LIST*:
1996/06/10 15:03:00 by wgong
1996/06/10 20:34:52 by bailey-23665
1996/06/17 15:15:47 by bailey
Access: 0
Reviewed: true
)

Location: 9931–10611 Exon Id: 19493 Support: author experimental

Figure 4: bioWidget interface to genomic sequence data.

genome sequencing project.

4 Concluding Remarks

While GAIA is an experimental prototype rather than a production system, it already fills a gap in the availability of practical software systems capable of at least semi-automating framework computational annotation. In addition, because of the flexibility of the GAIA architecture, new methods can be tested and evaluated in a rapid prototyping mode quickly leading to the identification of problem areas and trial of improvements.

With regard to data analysis, our approach has been to incorporate application software developed outside our group into the annotation engine infrastructure. (The one exception to this is our work on EST-driven gene identification.[8]) Perhaps the major barrier we have encountered is the lack of robust methods supporting inter-operation of application programs. Perl is an efficient language for mediating between software applications, but it requires us to implement an interface to each software package we plan to use. Recent efforts to implement CORBA interfaces to standard software packages may greatly simplify this task, if adequate performance can be assured.

Integration of distributed, heterogeneous data sources remains perhaps the greatest technical and scientific challenge. The diversity and growth rate of data sources (InfoBiogen lists more than 380 biology related online resources as of August, 1997) relevant to the annotation process is impressive. Data source integration can be broken down into a series of steps as described in Davidson *et al.*[11] The University of Pennsylvania Database Group has developed a system, Kleisli, which handles data transformation (re-structuring) and integration extremely efficiently for very diverse data sources. [9,10] Developments in Kleisli and other multi-database integration systems [12] are attempting to address the still substantial technical challenges to generalized data resource integration. However, a significant remaining impediment is the problem of matching instances of information between data sources. In part, this problem must be addressed by the development of ontologies of standardized nomenclatures and terminology, including synonym tables, for biology similar to what has been accomplished in the Unified Medical Language System.[b]

For the foreseeable future, manual entry of annotation will continue to be a critical source of information. Our goal is to permit investigator input of annotation in an environment which is simple, interactive, and able to perform basic consistency checking (*e.g.* controlled vocabularies), and then follow up with data entry verification by trained annotators. More sophisticated checks are also possible, such as structural constraints on gene models.[13] The annotation editor for the Mouse Gene Expression Database (GXD)[c] is an excellent model of the type of system

[b] `http://www.nlm.nih.gov/pubs/factsheets/umls.html`

[c] `http://www.informatics.jax.org`

needed for the genome annotator.

Further development of GAIA will be closely coordinated with the DOE funded Genome Annotation Collaboratory (GAC). Details can be found at the GAC WWW site.[d]

Acknowledgments

We would like to thank Paracel for the gift of the Fast Data Finder and Sun Microsystems for the gift of additional processors for the E4000. Work on GAIA was supported by NIH grant R011HG01539, and on bioWidget components by NIH grant RO1HG01450 and and DOE grant DE-FG02-94ER6192.

References

1. T. Gaasterland and E. Selkov. Reconstruction of metabolic networks using incomplete information. In *Proceedings of the Third International Conference on Intelligent Systems for Molecular Biology*, pages 127–135, Menlo, CA, 1995. AAAI Press.
2. T. Gaasterland and C. W. Sensen. MAGPIE: A multipurpose automated genome project investigation environment for ongoing sequencing projects. Available as Postscript document at http://www.mcs.anl.gov/home/gaasterl/magpie.html, 1996.
3. M. Scharf, R. Schneider, G. Casari, P. Bork, A. Valencia, C. Ouzounis, and C. Sander. GeneQuiz: A workbench for sequence analysis. In *Proceedings of the Second International Conference on Intelligent Systems for Molecular Biology*, volume 2, pages 248–253, Menlo, CA, 1994. AAAI Press.
4. G. Casari, C. Ouzounis, A. Valencia, and C. Sander. GeneQuiz II: automatic function assignment for genome sequence analysis. In *Proceedings of the First Annual Pacific Symposium on Biocomputing pp 707-709. Hawaii, USA - World Scientific.*, volume 1, pages 707–709. World Scientific, 1996.

[d] `http://avalon.epm.ornl.gov`

5. E. C. Uberbacher and R. J. Mural. Locating protein-coding regions in human DNA sequences by a multiple sensor-neural network approach. *Proc. Nat. Acad. Sci. USA*, 88:11261–11265, 1991.
6. S. F. Altschul, W. Gish, W. Miller, E. W. Myers, and D. J. Lipman. Basic local alignment search tool. *J. Mol. Biol.*, 215:403–410, 1990.
7. J. Jurka, P. Klonowski, V. Dagman, and P. Pelton. CENSOR–a program for identification and elimination of repetitive elements from DNA sequences. *Computers & Chemistry*, 20(1):119–121, March 1996.
8. C. L. Bailey, D. B. Searls, and G. C. Overton. Analysis of EST-driven gene identification in genomic DNA sequence. 1997. Submitted.
9. P. Buneman, S. B. Davidson, K. Hart, G. C. Overton, and L. Wong. A data transformation system for biological data sources. In *Proceedings of the 21st International Conference on Very Large Data Bases*, volume 21, pages 158–169. Morgan-Kaufmann Publishers, Inc., 1995.
10. S. Davidson, C. Overton, V. Tannen, and L. Wong. Biokleisli: A digital library for biomedical researchers. *International Journal of Digital Libraries*, 1(1), November 1996.
11. S. B. Davidson, G. C. Overton, and P. Buneman. Challenges in integrating biological data sources. *Journal of Computational Biology*, 2(4):557–572, 1995.
12. V.M. Markowitz, I.A. Chen, and A. Kosky. Exploring heterogeneous molecular biology databases in the context of the Object-Protocol Model. In S. Suhai, editor, *Theoretical and Computational Genome Research.* Plenum Press, 1996.
13. G. C. Overton, J. Aaronson, J. Haas, and J. Adams. QGB: A system for querying sequence database fields and features. *Journal of Computational Biology*, 1(1):3–13, 1994.