Fully automated genome analysis that reflects user needs and preferences

- A detailed introduction to the MAGPIE system architecture -

Gaasterland, T.[1,3], Sensen, C.W.[2,3,*]

[1] Mathematics and Computer Science Division, Argonne National Laboratory, 9700 S. Cass Ave, Argonne, Illinois 60439 and Department of Computer Science, University of Chicago, Ryerson Hall, 100 E. 58th St., Chicago, Illinois 60637; gaasterl@mcs.anl.gov

[2] National Research Council of Canada, Institute for Marine Biosciences, Halifax, N.S., Canada B3H 3Z1; sensencw@niji.imb.nrc.ca

[3] Canadian Institute for Advanced Research; Program in Evolutionary Biology

[*] to whom correspondence should be addressed (phone 902 426 7310, fax 902 426 9413)

received:

accepted:

Summary

A system called MAGPIE (Multipurpose Automated Genome Project Investigation Environment) has been designed and implemented to meet the challenges of automated whole genome analysis. The system initiates large numbers of remote and local transactions, each depending on evolving criteria and on changing remote and local conditions. Transactions are requested from different types of remote and local resources. The remote request load is fairly balanced with other community demands on server resources. Local decision modules monitor and obey user preferences and combine evidence from multiple sources to formulate credible hypotheses about sequence function. Consistency checks from multiple types of data are integrated into the ongoing local analysis. The system performs realiably on local UNIX workstations and communicates with remote resources through standard networking protocols.

Key words

annotated logic programming, automated genome analysis

Introduction

We have recently introduced an automated system for genomic sequence analysis called MAGPIE (Multipurpose Automated Genome Project Investigation Environment) [1]. MAGPIE is designed and implemented to meet the challenges that arise from the generation of complete genome sequences, during and beyond the lifetime of a genome sequencing project. When whole genomes[1] [2] [3] are analyzed, large numbers of remote and local transactions, each depending on changing remote and local conditions, must be initiated. Decision modules must monitor and obey user preferences [4] and combine evidence from multiple sources to formulate credible hypotheses [5] about sequence function. The data volume that needs to be analyzed in a genome project prohibits the use of "by hand" techniques and even discourages semi-automatic analysis approaches [6], because these are not efficient enough to keep up with the pace of sequence production. One megabase (mbp) of microbial genome sequence contains on the order of 1000 genes. In G+C rich organisms, the number of open-reading frames (ORF) is much higher than the number of genes due to the lack of artificial stop codons. Each open reading frame must be searched against an array of tools, therefore on the order of 100,000 database searches have to be performed and analyzed per megabase of genome each year.

The results of an automated genome analysis must be served in logical units that allow researchers to access the data in the most efficient way. Investigators need the ability to interact with the automated system in order to verify, recombine or refute the automatically generated information. The strategy behind MAGPIE differs from other approaches to automated genome analysis, which are designed as remote information servers [7], in that MAGPIE operates as a local system within a particular genome project. MAGPIE automates data collection and updating; data management; and initial decisions about genome features. MAGPIE also facilitates verification of assigned features and

---

[1] A list of active genome sequencing projects, compiled by Siv G. E. Andersson (Uppsala University, Sweden), Terry Gaasterland and Christoph W. Sensen, can be accessed through the MAGPIE WWW homepage: *http://www.mcs.anl.gov/home/gaasterl/magpie.html*.

updates through wet-lab validation or refutation. In this paper we describe the system organization, how users interact with the system and the strategies behind MAGPIE which are used to address the daunting task of whole genome analysis.

Biological background

To understand how MAGPIE integrates within a genome sequencing environment, we shall use the *Sulfolobus solfataricus* genome project[2] as an example. This project is generating the complete 3 mbp genomic sequence of the thermophilic archaeon *Sulfolobus solfataricus* P2 [8] using automated sequencing techniques. The project is distributed between laboratories at the University of Ottawa, Dalhousie University in Halifax, Nova Scotia and the NRC Institute for Marine Biosciences in Halifax, Nova Scotia. All three laboratories need access to the sequence data and analysis results in real time. The project has therefore chosen to rely on the Internet as the backbone for the exchange of files and information.

The project is organized on a cosmid by cosmid basis; these are the smallest logical units in the finally mounted complete sequence. Each cosmid that enters the sequencing process is subcloned as random (nebulized) subfragments of 1-2 kilo-base pair (kbp) size into the plasmid vector *pUC*18. Sequence assembly and contig editing are performed with the Staden package [9] which produces FASTA formatted output files. However, many other sequence file formats are supported for compatibility reasons (e.g. GCG, GenBank and Phylip interleaved).

The generated sequence changes its STATE three times, before it reaches the finished state: (a) the primary sequencing state, where plasmid subclones are sequenced from the ends with M13 universal and reverse primers and the sequences are assembled into contigs (which are subcontigs of the final contig); (b) the linking state where walking primers are used to link the subcontigs generated in the primary sequencing phase into one contig, and (c) the polishing state, where single stranded regions in the completely linked contig are double stranded and ambiguities are resolved. In each state, except for the finished state, sequence gets updated frequently, as subcontigs grow or ambiguities are resolved.

---

[2] Additional information about the *Sulfolobus solfataricus* P2 genome project can be obtained from the *Sulfolobus* WWW homepage at *http://www.imb.nrc.ca/imb/sulfolob/sulhom_e.html*.

For each state, sequence analysis requires a different set of TOOLS which are applied to identify sequence features. Depending on the sequence quality, some tools also need to be employed with different parameters. To obtain the most accurate knowledge about what ORFs encode, they need to be spliced out of completely linked contigs and analyzed individually. Therefore, a GROUP of sequences is associated with one cosmid (i.e. several subcontigs of the final contig in the primary and linking phases and the whole contig and the spliced out ORFs in the polishing and finished phases).

The environment around the *Sulfolobus* project is changing rapidly, due to updates of the public sequence databases and the development of new sequence analysis tools. Therefore, even the finished sequence needs to be re-searched and re-analyzed periodically. Further, the rapidly growing number of services on the Internet provides more and more sources of information which can be linked to sequence analysis (e.g. Enzyme and Metabolic Pathway information, Phylogenetic Information, and Medline links). To analyze the sequence, the *Sulfolobus* project has decided to use any tool that will create at least partly unique information. This implies, for example, that *blast* and *fasta* are used simultaneously because the tool output will be partially non redundant.

Setting up a MAGPIE project

The creation of a MAGPIE project is a three step procedure: installation of the MAGPIE core distribution; creation of a new MAGPIE project and configuration of the new project. The MAGPIE core distribution can be installed on virtually any UNIX platform. To set up a MAGPIE core distribution, the following packages need to be installed on a UNIX computer:

Program	Source
- a C-compiler (e.g. gcc)	ftp://prep.ai.mit.edu/pub/gnu
- SICSTUS 3.0 Prolog	http://www.sicstus.se/ps/sicstus.html#Order
- Perl 5.001m (or higher)	ftp://prep.ai.mit.edu/pub/gnu
- the GD 1.2 library	http://www.boutell.com/gd/#getgd
- GD.pm	http://www-genome.wi.mit.edu/ ftp/distribution/software/WWW/GD.html
- the NCSA httpd daemon	http://hoohoo.ncsa.uiuc.edu/docs/setup/OneStep.html
- readseq	ftp://ftp.bio.indiana.edu/molbio/readseq

All of the above software is public domain except the SICSTUS 3.0 Prolog compiler. MAGPIE consists of Prolog code that needs to be compiled for the installation and a set of Perl 5 scripts that are orchestrated by the Prolog executables. For each genome project one dedicated account has to be created, MAGPIE uses the mailqueue of that account for its interaction with e-mail tool servers. If MAGPIE employs local tools, these have to be installed in a path that is visible to the dedicated account.

To create a new MAGPIE project, the Perl script **mkproject** is executed. The user is prompted for a set of questions about the new project: (1) Mailqueue for incoming responses; (2) Email address for notification messages; (3) Input file directory for the project (for new sequences and assembly information, for example); (4) Organism (for codon usage); (5) Special start codons; (6) Special stop codons; (7) Organism name and (8) WWW address for project information (optional). The answers to these questions are stored in a *.magpierc* file that is created in the home directory of the dedicated project account during the setup process. The **mkproject** script creates a hierarchical directory structure for the project and copies a set of configuration files, header templates and tables (e.g. codon-usage tables) into the project's **Config** and **Headers** directories.

The next step in the setup of a new MAGPIE project is the creation of the groups. Each group holds a set of sequences[3]. In the *Sulfolobus* project, groups are based on cosmids. The **mkgroup** script sets up a new group and adds the subdirectories for that group to the project directory structure. The entire directory structure which is created by the **mkproject** and **mkgroup** scripts is shown in figure 1.

>>>>> Insert figure 1 here. <<<<<

After the initial project setup, the configuration files are edited to fine tune the setup for the given genome. To allow editing with any UNIX editor, all configuration files are in ASCII format. Through the configuration files MAGPIE can be customized for a particular genome project without alteration of the MAGPIE program code. Together, the TOOLCONFIG, SCORECONFIG and STATECONFIG files capture a user's preferences to be reflected in the analysis of the data. Local and remote tools can be added to the TOOLCONFIG file, states for the sequence and the tools associated with a particular state to the STATECONFIG, and score levels to the SCORECONFIG. The order of tools in TOOLCONFIG indicates the priority of each tool contributing to a decision. For example, *blastx* before *blaize* indicates that within the same score strength, *blastx* calls should be given priority in the face of conflict. *Blaize* (a.k.a. *blitz* or *mpsearch*) before *blocks* indicates that full protein alignments have priority over motif hits. In SCORECONFIG, score ranges from each tool are assigned to a "level". A level corresponds to a users qualitative judgement about confidence in a tool. The STATECONFIG file contains information which tools should be applied for a particular state and the frequency at which they should be re-searched. In REPORTCONFIG, the user indicates which types of facts should be included in each report, and which tools to prefer for extracting description

[3] The set of sequences contained in a group must be defined in the context of a sequencing strategy. For example (as menitioned in the previous section), when sequencing randomly on cosmid subclones followed by primer-walking, the emerging contigs, and eventually the whole contig, for a cosmid form a group. Alternatively, when sequencing by primer walking from cosmid ends, a 100 kbp assembly project could form a natural group.

lines. The content of local databases associated with the given genome can be configured in LOCALDBCONFIG (the local databases can hold any sequence data, extracted from the given genome or sequence databases). Background colors of the html pages are specified in COLORCONFIG, and the href information for hyperlinks in the html pages in LINKCONFIG. If MAGPIE tasks can be distributed over a cluster ol local machines, the machines for each task can be listed in DISTCONFIG. The relation between finished subcontigs can be declared in CONNECTCONFIG A full set of configuration files can be accessed via the World Wide Web at: *http://nori.imb.nrc.ca/mgenit/private/mgenit.html*. The following box shows the TOOLCONFIG file as a typical example for a configuration file:

TOOLCONFIG file

```
#
#TOOLID  INPUT  EMAIL                                OUTPUT_TYPE   FACTTYPE   TOOLTYPE
#
blastx   DNA blast@ncbi.nlm.nih.gov                  fact          prosim     blast
blastp   AA  blast@ncbi.nlm.nih.gov                  fact          prosim     blast
blaize   AA  blitz@embl-heidelberg.de                fact          prosim     blaize
trnascn  DNA local                                   fact          dnasim     trnascan

blastn   DNA blast@ncbi.nlm.nih.gov                  fact          dnasim     blast
tblastn  DNA blast@ncbi.nlm.nih.gov                  fact          prosim     blast
tblastx  DNA blast@ncbi.nlm.nih.gov                  fact          prosim     blast

blocks   DNA blocks@howard.fhcrc.org                 fact          motif      blocks
prosrch  AA  local                                   fact          motif      prosearch

fastaw   DNA fasta@ebi.ac.uk                         fact          dnasim     fasta
fastan   DNA fasta@ebi.ac.uk                         fact          dnasim     fasta
fastap   AA  fasta@ebi.ac.uk                         fact          prosim     fasta
fastae   DNA fasta@ebi.ac.uk                         fact          dnasim     fasta
fastav   DNA fasta@ebi.ac.uk                         fact          dnasim     fasta

genmark  DNA genemark@ford.gatech.edu                fact          orf        genemark

prepro   AA  predictprotein@embl-heidelberg.de       text          secstr     prepro

blastxL  DNA local                                   fact          repeat     blast
blastnL  DNA local                                   fact          repeat     blast

sputnik  DNA local                                   text          repeat     sputnik
saps     AA  local                                   text          stat       saps

# 1 LETTER ABBREVIATIONS: (default is first letter, which could be ambiguous)
prosim p
dnasim d
motif  m
orf    o
secstr s
repeat r
stat   n
```

In addition to the config files, header templates are used to relay parameters to local and remote tools. The header templates are also used to define the specific input file format that is required by the tool. The following boxes show examples for header templates:

blocks Header template

```
>MAGPIE-id
MAGPIE-sequence
```

fastan Header template

```
TITLE MAGPIE-id
LIB EMALL
LIST 10
ALIGN 10
SEQ
MAGPIE-sequence
END
```

tblastn Header template

```
PROGRAM tblastn
DATALIB nr
BEGIN
>MAGPIE-id
MAGPIE-sequence-AA
```

When invoking a tool, the MAGPIE system replaces the MAGPIE-id, with a string of the form *tool--group--contig--requesttype* (the id information is used to refile incoming responses into the appropriate directories). For local tools, the command line that invokes the tool must be stored in the **LocalTools** directory. The following box shows an example for a local tool command-line file:

prosearch command line

```
/programs/ProSearch20/prosearch  -sd  MAGPIE infile  >  MAGPIE outfile
```

Here the MAGPIE system replaces “MAGPIE infile” with the full input file name and does likewise for the output file.

After the MAGPIE configuration is complete, new sequence can be entered and the MAGPIE data collection daemon can be started. The sequence data that are entered into the project can have any format supported by *readseq*. The input file format is automatically detected, which allows entering of data from a wide variety of different sequence assembly sources. MAGPIE internally handles all sequence data as FASTA formatted files.

The MAGPIE system architecture

MAGPIE is built around two Prolog daemons: the data *collection-daemon* and the *data analysis-and-report* daemon. The daemons operate independently from each other. Each daemon orchestrates a suite of MAGPIE perl scripts and local tools (these may be scripts or executables). The general MAGPIE architecture is shown in figure 2.

>>>>> Insert figure 2 here <<<<<

Users communicate with both daemons through shell interfaces, the configuration files, tables and headers. All information that is declared by the user is stored as Prolog facts in the **group/Prolog** directories and in the **project/log** directory (see Fig. 1), respectively. The daemons write log files to the **project/log** directory that account for all daemon activities.

The *data-collection* daemon monitors the MAGPIE sequence input directory, reformats the new sequences to FASTA format if necessary, refiles each new sequence to its appropriate group directory, splices out open ORFs for completely linked contigs[4], sends sequences to the configured tools and digests incoming responses into the input for the *analysis-and-report* daemon. The flow control allows the daemon to send only one request per tool and machine at a time. The *data-collection* daemon waits for the reply from the tool server to arrive before the next request is sent. This strategy allows MAGPIE to share remote resources (e.g the *blast* server) fairly with the large user

[4] As soon as a group of contigs is in a relatively linked an ambiguity-free state (as indicated in the STATECONFIG file), MAGPIE splices out ORFs, translates them by using organism-specific codon usage tables and sends ORF-specific requests to protein analysis tools. For *Sulfolobus*, a prokaryotic approach suffices: for each stop codon in each reading frame, the furthest upstream start codon is identified (that is at least some minimal distance (100 amino acids) away from the stop codon; valid *Sulfolobus* start codons are ATG, GTG, TTG). Each ORF is regarded as a potential coding region. Similarity searches, assessments of codon usage, secondary structure predictions, motif searches, and three dimensional threading techniques all contribute to an attempt to associate functional identity with the amino-acid sequence.

community. The responses are reformatted into html (the hypertext markup language). In the reformatting process, all evidence below a certain cutoff level (the level is specified in the SCORECONFIG) is erased. This shrinks the amount of data to be stored by about 80%. If reasoning about the tool output is necessary during the analysis, Prolog facts are extracted from the responses and stored in the **group/Fact** directories. MAGPIE also supports incremental updates for tools like *fastan* or *blastn*. For incremental updates, the project data are only searched against new sequence database entries (e.g. EMNEW). The facts above the cutoff level are added to the existing facts for those tools. The following box shows a set of Prolog facts that were extracted from a *blastp* response:

blastp Fact File (cutout)

```
?- multifile sim/6.
?- dynamic sim/6.
?- op(900,xfy,--).
sim('sh01e0701--c01009',['sh01e0701--c01009',688,867],[score(762),expect(1.9e-198),p(1.9e-
198)],['pir|A27339',235,499],blastp,desc("EC  2.7.1.30  glycerol  kinase  -  Escherichia
coli")).
sim('sh01e0701--c01009',['sh01e0701--c01009',367,546],[score(380),expect(1.9e-198),p(1.9e-
198)],['pir|A27339',125,232],blastp,desc("EC  2.7.1.30  glycerol  kinase  -  Escherichia
coli")).
sim('sh01e0701--c01009',['sh01e0701--c01009',13,192],[score(339),expect(1.9e-198),p(1.9e-
198)],['pir|A27339',6,119],blastp,desc("EC 2.7.1.30 glycerol kinase - Escherichia coli")).
sim('sh01e0701--c01009',['sh01e0701--c01009',400,441],[score(35),expect(1.1e-37),p(1.1e-
37)],['pir|A27339',308,321],blastp,desc("EC  2.7.1.30  glycerol  kinase  -  Escherichia
coli")).
sim('sh01e0701--c01009',['sh01e0701--c01009',688,867],[score(762),expect(1.9e-198),p(1.9e-
198)],['pdb|1GLA',234,498],blastp,desc("Glycerol  Kinase  (E.C.2.7.1.30)  Complex  With
Glycerol And The")).
sim('sh01e0701--c01009',['sh01e0701--c01009',367,546],[score(380),expect(1.9e-198),p(1.9e-
198)],['pdb|1GLA',124,231],blastp,desc("Glycerol  Kinase  (E.C.2.7.1.30)  Complex  With
Glycerol And The")).
sim('sh01e0701--c01009',['sh01e0701--c01009',13,192],[score(339),expect(1.9e-198),p(1.9e-
198)],['pdb|1GLA',5,118],blastp,desc("Glycerol Kinase (E.C.2.7.1.30) Complex With Glycerol
And The")).
...
```

The *analysis-and-report* daemon analyzes responses based on the Prolog fact files for a contig and builds the final reports. MAGPIE creates a variety of reports, which are either ASCII text files or WWW browsable reports. Each type of report was defined in collaboration with expert biologist users. The specific content of each report reflects the preferences denfined for the project through the config files. At any point, the config files can be altered and the reports regenerated in order to refine the automated view of the data.

The html reports are interconnected by hyperlinks. In the html reports, information is organized on three principal levels: (1) project-wide information, (2) group-wide

information and (3) contig-wide information. Within these levels, there are six general types of html reports: (1) Similarity reports, (2) Frameshift reports, (3) EcoVec reports, (4) Repeat reports, (5) RNA reports and (6) Metabolic pathway reports. In addition, information about the data collection status, the config files and the physical properties of ORFs and translated proteins is linked into the html pages. Table 1 gives an overview over the different types of information in the html pages and their connection points:

>>>>> insert table 1 here <<<<<

Similarity reports are a hierachical html structure that presents evidence based on the output of database search tools. MAGPIE discriminates between three different types of similarity tools: (1) Protein similarity tools [e.g. *blastp* or *blaize*], (2) DNA similarity tools [e.g. *blastn* or *trnascan*] and (3) Motiv similarity tools [e.g. *blocks* or *prosearch*]. For the similarity analysis, the *analysis-and-reports* daemon loads all Prolog facts for a particular subcontig or ORF and reasons on them with the goal to sort them into three different levels of confidence: (1) level one hits, which have a 90% or greater probability of being true, (2) level two hits, which have a 50% to 90% probability of being true and (3) level three hits which have a 50% or less probability of being true. The results of the analysis are stored in the **group** directories as a set of Prolog facts which is used to build the reports.

Frameshift reports are based on the similarity reports. Prolog reasons about neighboring ORFs on the same DNA strand which hit against identical database entries. If the 5' end of a database entry hits against the upstream ORF and the 3' end of the same database entry against the downstream ORF, a potential frameshift is given. Frameshift reports are linked from the similarity report html pages.

EcoVec reports list *Escherichia coli* and *cloning vector* contaminations for each group in the primary sequencing phase. These reports are based on *fasta* searches against the EMPRO and EMSYN subsets of the EMBL database. Hits with at least 95% identity are reported. For microbial genome projects, the EcoVec reports also include hits against

database entries from the same organism. EcoVec reports are linked to the similarity report group home pages.

Repeat reports are based on local *blastn* and *blastp* searches against the entire DNA sequence and the spliced out ORFs respectively. The repeat report for the entire sequence information is linked from the project home page, and the repeat reports for individual contigs are linked from the group home pages.

RNA reports indicate regions for which there is evidence that they code for a tRNA or some other RNA.

Pathway reports are based on enzyme identifications [10] in the similarity reports. Each identified EC-number is colored in a Pathways masterfile, according to the level of confidence for the similarity identification. EC-numbers for which no sequence information exists in the public sequence databases are highlited. This allows users to judge the likelihood of the occurrence of a particular pathway in an organism.

Additional ASCII reports contain lists and tables that can be used for a quick overview (executive reports), database submission forms, or export files that can be imported by other systems, e.g. Subtilist [11] or IGD [12].

Making Logical Decisions about Genome Features

In order to reason about features of a genome, the output from each remote and local analysis tool is digested into a set of Prolog [13] facts. Each fact represents a statement about a region of the genome. Conclusions about properties of the genome are made through deductive rules that are applied to the set of statements. The Prolog Facts are stored in ASCII flat files. This concept gives MAGPIE great flexibility by enabling reasoning on any information that is extracted from a tool's response.

To illustrate how information from multiple sources can be combined through Prolog, we shall consider a rule that combines gobal and local similarity evidence to discern a coding region[5].

The first rule is based on the following assumption: "There is a **coding_region** in a **Contig** if there is both, a **global_similarity** and a **local_similarity** against the same database sequence, and if the **Confidence** in the **coding_region** is the **maximum** of the Confidences of the similarities":

coding_region(Contig,From,To,Confidence)←

global_similarity(Contig,From1,To1,DBSequence,Confidence1),
local_similarity(Contig,From2,To2,DBSequence,Confidence2),
overlaps(From1,To1,From,To),
overlaps(From2,To2,From,To),
start(Contig,From),
stop(Contig,To),
Confidence = maximum(Confidence1,Confidence2).

The following two rules define that *blaize* supplies **global_similarity** and *blastp* supplies **local_similarity**:

[5] Rules of the form **Head** ← **Body** can be read as "**Head** is true if **Body** is true" or "**Head** if **Body**".

global_similarity(Contig,From,To,DBSequence,Confidence)←

similarity(Contig,From,To,*blaize***,DBSequence,Confidence).**

local_similarity(Contig,From,To,DBSequence,Confidence)←

similarity(Contig,From,To,*blastp***,DBSequence,Confidence).**

Additional rules can be inserted into the body of assumptions at any time. If, for example, **similarity** and **motif** information should be used in a decision about a coding region, such that strong **motifs** reinforce weak **similarity** information, the following rules could be added:

coding_region(Contig,From,To,high)←

similarity(Contig,From1,To1,Confidence1),

motif(Contig,From2,To2,*high***),**

Confidence1 > *shaky***.**

In these examples, confidence levels neet not be numeric nor completely odered. A partial ordering over symbolic values (e.g. “shaky” < “high”, “weak” < “strong”) can be used to make decisions through qualitative query-answering techniques [4].

Verification of MAGPIE results

Automated genome analysis results are only the first step in understanding complete genomes. Based on the results, researchers will design new experiments to verify or reject suggestions and theories that are presented to them automatically. MAGPIE allows users with password access to change information in form-based html pages to reflect the validation process and add additional information about genes obtained in the wet lab. The manually entered information overrides the automatically obtained results. This allows “value added” genome analysis, that finally gives correct information, even for members of large gene families or multifunctional enzymes.

References

[1] Gaasterland T, Sensen CW (1996) MAGPIE: automated genome interpretation. *Trends in Genetics* 12, 76-78

[2] Fleischmann RD, Adams MD, White O, Clayton RA, Kirkness EF, Kerlavage AR, Bult CJ, Tomb JF, Dougherty BA, Merrick JM, MeKenney K, Sutton G, FitzHugh W, Fields C, Gocayne JD, Scott J, Shirley R, Liu L-I, Glodek A, Kelley JM, Weidman JF, Phillips CA, Spriggs T, Hedbloom E, Cotton MD, Utterback TR, Hanna MC, Nguyen DT, Saudek DM, Brandon RC, Fine LD, Fritchman JL, Fuhrmann JL, Goeghagen NSM, Gnhem CL, McDonald LA, Small KV, Fraser CM, Smith HO, Venter JC (1995) Whole-genome random sequencing and assembly of *Haemophilus influenzae* Rd. *Science* 269, 496-512

[3] Fraser CM, Gocayne JD, White O, Adams MD, Clayton RA, Fleischmann RD, Bult CJ, Kerlavage AR, Sutton G, Kelley JM, Fritchman JL, Weidman JF, Small KV, Sandusky M, Fuhrman J, Nguyen D, Utterback TR, Saudek DM, Phillips CA, Merrick JM, Tomb J-F, Dougherty BA, Bott KF, Hu P-C, Lucier TS, Peterson SN, Smith HO, Hutchison CA III, Venter JC (1995) The minimal gene complement of *Mycoplasma genitalium*. *Science* 270, 397-403

[4] Gaasterland T, Lobo J (1994) Qualified answers that reflect user needs and preferences. *In: Proceedings of the International Conference on Very Large Databases*, Santiago, Chile

[5] Gaasterland T, Lobo J, Maltsev N, Chen G (1994) Assigning Function to CDS through qualified query answering. *In: Proceedings fo the Second International Conference in Intelligent Systems for Molecular Biology*, Stanford University

[6] Smith R, Worley K (1996) the BCM Search Launcher and Batch Client. *Trends in Genetics 12*, 77

[7] Scharf M, Schneider R, Casari G, Bork P, Valencia A, Ozounis C, Sander C (1994) GeneQuiz: a workbench for sequence analysis. *In: Proceedings of the Second International Conference on Intelligent Systems for Molecular Biology* (Altman R, Brutlag D, Karp P, Lathrop P, Searls D, eds) AAAI Press, Cambridge, 348-353

[8] Sensen CW, Charlebois RL, Singh RK, Klenk H-P, Ragan MA, Doolittle WF (1996) Sequencing the genome of *Sulfolobus solfataricus* P2. *In: Bacterial Genomes: Physical Structure and Analysis* (DeBruijn FJ, Lupski JR, Weinstock G, eds) Chapman and Hall, New York, in press

[9] Bonfield JK, Staden R (1995) A new DNA sequence assembly program. *Nucleic Acids Research* 23, 4992-4999

[10] Gaasterland T, Selkov E (1995) Reconstruction of Metabolic Networks using Incomplete Information. *In: Proceedings of the Third International Conference on Intelligent Systems for Molecular Biology* (Altman R, Brutlag D, Karp P, Lathrop P, Searls D, eds) AAAI Press, 127-135

[11] Moszer I, Glaser P, Danchin A (1995) SubtiList: a relational database for the *Bacillus subtilis* genome. *Microbiology* 141, 261-268

[12] Ritter O, Kocab P, Senger M, Wolf D, Suhai S (1994) Prototype implementation of the integrated genomic database. *Computational Biomedical Research* 27, 97-115

[13] Bratko I (1990) PROLOG: Programming for Artificial Intelligence. Second Edition. Addision Wesley Publishing Co, New York

Acknowledgements

This work was supported by the US Department of Energy, under Contract W-31-103-Eng-38. The *Sulfolobus* project is supported by the Canadian Genome Analysis and Technology Program (CGAT), the Canadian Institute for Advanced Research (CIAR), the National Research Council of Canada (NRC) and the Medical Research Council of Canada (MRC).

Catalogued at NRC as NRCC 39717

Tables:

Table1: Overview over the hyperlink connection points

	Project home	Group home	Contig home	Status pages	Evidence
Project statistics	x				
Project html link	x				
Group home pages	x				
Contig home pages		x			
Status pages		x			
ORF information		x			
Config file links	x	x	x	x	
Database links			x		x
EcoVec reports		x			
Evidence links			x	x	
Frameshifts reports	x	x			
Metabolism link	x				
PUMA links			x		
Repeats link	x[1]	x[2]			
Top Hit reports		x			

[1,2]= Repeat reports are split into a project wide report and contig-specific reports

Figure Legends

Figure 1: MAGPIE project directory structure.

Directories with white background are created with the **mkproject** script, directories with grey or black background with the **mkgroup** script. For directories with black background (i.e. groups B and C), the full sub-directory structure is not shown.

Figure 2: MAGPIE flowchart.

Data are entered from a sequence source(1), sent to remote (2) or local tools (3), the responses are reformatted to html format (4), Prolog facts are extracted from the responses (5), the facts are loaded into the analysis (6), the analysis results are stored as another set of Prolog facts (7), the reports are build, based on the analysis results (8), and local and remote html addresses are linked into the final reports (9). The digested responses, html reports and public resources are all html-formatted files.