Proceedings of the 11th International Symposium on
Computer Applications in Biotechnology (CAB 2010),
Leuven, Belgium, July 7-9, 2010
Julio R. Banga, Philippe Bogaerts, Jan Van Impe,
Denis Dochain, Ilse Smets (Eds.)

Merlin: Metabolic Models Reconstruction using Genome-Scale Information ⋆

Oscar Dias * **Miguel Rocha** ** **Eugénio C. Ferreira** *
Isabel Rocha *

* *IBB Institute for Biotechnology and Bioengineering, Centre of Biological Engineering, University of Minho, Campus de Gualtar, 4710-057 Braga, Portugal (e-mail: [odias, ecferreira, irocha]@ deb.uminho.pt).*
** *CCTC -Computer Science and Technology Centre, University of Minho, Campus de Gualtar, 4710-057 Braga, Portugal (e-mail: mrocha@ di.uminho.pt)*

Abstract: This article describes Merlin, a user-friendly program that performs functional genomic annotations of lists of genes. Merlin retrieves information of each homologue and automatically scores the results, allowing the user to change the score selection, and dynamically (re-)annotate the genome. Merlin expedites the transition from genome-scale data to SBML metabolic models, allowing the user to have a preliminary view of the biochemical network.

Keywords: Systems Biology, Genome-Scale Reconstruction, BLAST, SBML, Metabolic Engineering.

1. INTRODUCTION

Genome-scale reconstructed metabolic models are based on the well-known stoichiometry of biochemical reactions and can be used for simulating *in silico* the phenotypic behaviour of a microorganism, under different environmental and genetic conditions, thus representing an important tool in Metabolic Engineering [Rocha et al. (2008)]. The reconstruction of a metabolic network associates the genome of a given organism to its physiology, through the replication of the biochemical reactions and molecular mechanisms taking place in a given organism [Francke et al. (2005)].

The genome-scale reconstruction of metabolic networks encompasses several steps, such as genome annotation, reactions identification and stoichiometry determination, compartmentation, determination of the biomass composition, energy requirements and additional constraints. The first step (genome annotation) is essential to this type of reconstruction, because precursory data can be extracted for the model reconstruction. Information such as gene or open reading frame (ORF) names, assigned cellular functions, sequence similarities, and, for the enzyme coding genes, the Enzyme Commission (EC) number(s) should be retrieved to accomplish the first stage of the mathematical model development [Rocha et al. (2008)].

According to the Integrated Microbial Genomes (IMG) system [Markowitz et al. (2006)] there are currently more than 4.000 genomes (4.368 as of December 2009) fully sequenced with more than 700 genomes (747 as of December 2009) being drafted right now. Sequence similarities between genes and genomes can be established using well known algorithms such as *BLAST* [Altschul et al. (1990)] or *FASTA* [Lipman and Pearson (1985)].

⋆ This work is supported by a PhD grant from the portuguese Fundação para a Ciência e a Tecnologia: SFRH/BD/47307/2008.

2. GENOME ANNOTATION

Genome Annotation encompasses both "gene finding", on the sequenced genome, and the assignment of biological functions to the recently found genes [Medigue and Moszer (2007); Salzberg (2007)].

Gene finding in eukaryotic genomes is different than in the prokaryotic ones, as about 90% of the bacterial genome are coding sequences. On the other hand, higher eukaryotes have less than 10% of coding sequences. Moreover, eukaryotes generally have two or more overlapping open reading frames, and it is difficult to identify the start of translation and find regulatory signals such as promoters and terminators [Salzberg et al. (1998)].

There are several software tools for *gene finding*. Almost all use probabilistic methods, such as Hidden Markov Models (HMM), to identify coding sequences within the open reading frames. Examples of such applications are GLIMMER [Salzberg et al. (1998)], GenMark [Borodovsky and Mcininch (1993)], EuGène [Foissac and Schiex (2005)]. Alternatively, there are some tools that use methods other than HMM, such as Gismo [Krause et al. (2007)]. A list of some of these, and some other, applications is available at *www.genefinding.org/software.html*.

Some of the software applications listed above also attach biological data (functional annotation) to the recognised genes. Other tools that annotate the genome at the protein level, are GOAnno [Chalmel et al. (2005)], or GeneFAS [Joshi et al. (2004)] which uses Bayesian probability of function similarity between two connected genes and

several other tools. These applications try to assign one or more proteins to each gene product.

The gene functional annotation procedure can be defined as the assignment of functional information to a specific gene. Such information is often obtained by similarity to formerly characterized sequences, found in several online or local databases [Ouzounis and Karp (2002)]. At the time of the annotation, a given gene product may be unknown, and labelled as hypothetical protein. Even if a "real" protein is assigned to a gene, such protein may not be the correct one, leading to a misclassification.

2.1 Re-Annotation

As the ever-increasing knowledge of genomes grows, the annotation of genes becomes outdated over time [Salzberg (2007)]. Thus, the re-annotation of a genome, especially for genes classified as hypothetical proteins, is very important for assuring an up-to-date gene list and not compromising future similarity alignments for newly sequenced genes. The creation of a gene-function *wiki* is proposed by Wang (2006). Such *wiki* would also simplify the selection of a similarity search result. Salzberg (2007) goes even further, and proposes the development of genome *wiki's*, where each genome should have a *wiki* and biologists would help the re-annotation process in a familiar environment. There are already some genome *wiki's* for several organisms such as the *Drosophila wiki*, the fungal genomes *wiki* or the *E. coli wiki*. However, the *wiki* solution can be very demanding at the curation level, thus it can only be adopted for organisms which have already been systematically studied, either automatically or manually.

Nevertheless, after the initial annotation of the genome, there are several circumstances that can lead to a genome-wide re-annotation. Whether new genes or protein functions are discovered, a research group tries to determine the reproducibility of an existing annotation, or just because the information associated to a specific organism is known to be outdated, a genome-wide re-annotation will update the data assigned to such genome [Ouzounis and Karp (2002); Tamaki et al. (2007)].

There are already some applications that perform semi-automatic or manual functional annotation of a genome. However, most of these tools are aimed at genome projects and do not provide outputs that allow easy genome-wide (re-)annotations for the development of genome-scale metabolic models.

2.2 BLAST

BLAST [Altschul et al. (1990)] is used to compare two sequences (pairs of genes or proteins) and to search for locally similar regions on such sequences. Initially, the sequence is matched to nucleotide or amino acid sequences, throughout the defined target databases, containing millions of sequences. Afterwards, the statistical significance for each sequence match is calculated. The results of a *BLAST* analysis provide functional and evolutionary relationships between related sequences. Though being similar to *FASTA*, *BLAST* is quicker be-cause it searches only for rarer, more significant patterns in sequences. Nevertheless, these two algorithms are the most reliable ones to find close homologues between genes [Salzberg et al. (1998)].

3. *MERLIN'S* SYSTEM AND METHODS

We introduce Merlin for the reconstruction of genome-scale metabolic models. Some features within Merlin overlap several other software tools. Merlin allows the user to: perform similarity searches for any organism that has its genome sequenced, perform semi-automated dynamic (re-)annotation of the genome, and generate new *GenBank* genome annotated files (.gbk) from the existing ones, for submission to *NCBI, EMBL* and/or *DDBJ*. The user may also combine the similarity data with the information previously loaded into a local database and export the results to a metabolic model in the Systems Biology Markup Language (*SBML*) [Hucka and *et al* (2003)] format.

3.1 Specifications

Merlin is composed by two modules: the Dynamic Annotation Tool and the Model Reconstruction Tool, each of which will be further described in the next sections.

The Dynamic Annotation Tool automatically annotates lists of genes, properly provided in the *FASTA* format (files containing either nucleotide or amino acid sequences). This module allows the user to define the *BLAST* similarity searches initial parameters such as the e-value, maximum number of hits, remote database, etc. The results of the *BLAST* search are then scored, allowing the user to dynamically (re-)annotate each gene, either by accepting the scorer selection or selecting another entry, supported by a quantifiable confidence level. If none of the presented results satisfies the user, a manual record can also be added.

The Models Reconstruction Tool allows the user to load information from the Kyoto Encyclopedia of Genes and Genomes (KEGG) [Kanehisa and Goto (2000)], integrate it with information from the previous module and later build the metabolic model storing it in the well accepted *SBML* implementation.

Merlin was developed using the Java and Perl programming languages and is supported by the AIBench framework(http://www.aibench.org).

Merlin is available for Windows and distributed at *http://sysbio.uminho.pt/merlin* merlin under the GNU General Public License.

3.2 Merlin Architecture

The (re-)annotation process in Merlin is based on similarity searches to the online *GenBank* databases, as illustrated in figure 1 where Merlin's architecture is depicted. From this process, a list of files is generated, one for each gene, containing similarity information. After that, information for all the homologues present in each file is retrieved from the *Entrez Protein* database and loaded into a local relational database.

The acquired information is shown for user appraisal and interaction. The user can then select the information based

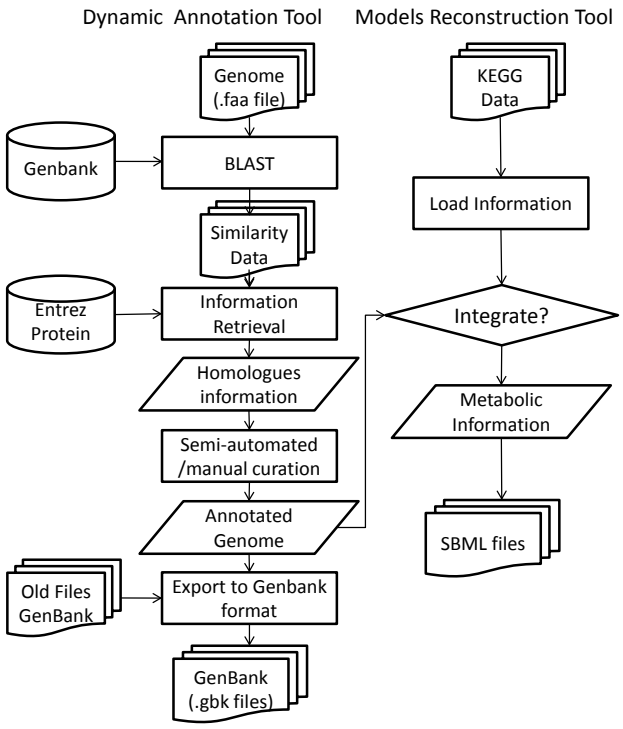

Fig. 1. Schematic representation of *Merlin's* architecture.

on the confidence level scores, provided by Merlin. After the manual curation, the user can export a new annotated file and/or integrate the information with the previously loaded *KEGG* information.

The last stage is the *SBML* model generation. Since the only metabolic information retrieved from a BLAST search is the EC number, the similarity information is integrated with the *KEGG* data, providing new reactions to be added to the metabolic model. Hence, the reactions stored in the local database, which are catalysed by the enzymes identified in the similarity search, along with the reactions already assigned for the case study by *KEGG*, are accepted for the generation of the metabolic model.

3.3 Operations implementation

Merlin's (re-)annotation: The purpose of this operation is the inference of candidate functions that could be assigned, by homology, to the proteins encoded by each gene in the genome. Such process is depicted in figure 2 and will be next described.

First of all, the genome files are downloaded from the *GenBank* ftp website. Next, the application uses a fasta file per chromosome of the organism that is being studied. These files contain the amino acids or nucleotides sequences for each gene.

Then, such files are submitted to a Perl routine that runs *BLAST*, remotely, to one of the *GenBank* databases. This routine was developed amid the *BioPerl* project [Stajich and *et al* (2002)]. For each gene, the result of the *BLAST* is kept in a file that contains homology information. For each homologue identified, the returned homology data is the following: locus identifier, e-value, *BLAST* score and organism.

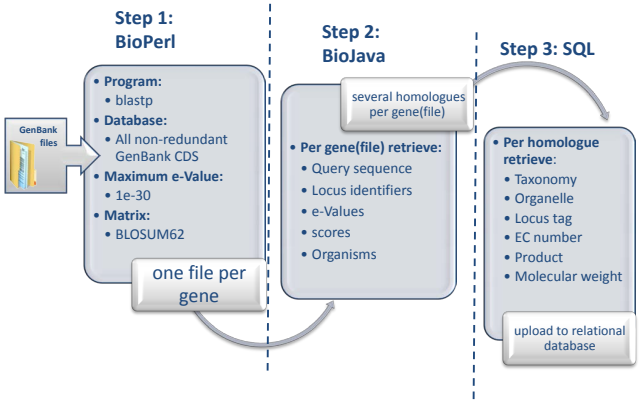

Fig. 2. Path from genome to homology data.

After that, a java tool, developed upon the *BioJava* project [Holland et al. (2008)], is run to collect information about each of the homologues identified for every gene. The data is retrieved remotely from the *Entrez Protein* database. The information to be downloaded is the following: taxonomy, organelle (if available), chromosome (if available), locus tag, product (protein name), EC number (if available) and molecular weight.

Finally, the downloaded information is stored in a *MySQL* relational local database, where it is available to be used for the reconstruction of the metabolic model.

Merlin's Load Database: This operation loads several *KEGG* data files (compound, glycan, compound.inchi, reaction, ec.list, enzyme, organism_enzyme.list and organism.ent) and builds a local database that allows the user to later assemble a genome-scale model, selecting and editing reactions, to be included in the model. This feature is handled by a Perl routine that parses and loads data from the files listed above, which were downloaded from the *KEGG's* ftp website [1]. Some of the information downloaded from *KEGG* is generic (ligand and pathway databases - compound, glycan, compound.inchi, reaction, ec.list, and enzyme). That means that it is the same for all organisms. On the other hand, data downloaded from the *KEGG* genes database (organism_enzyme.list, organism.ent) is organism specific. Such data specifies which enzymes are assigned to each gene and the gene itself.

Merlin's Views and Edition: The views of the local database enable the edition of any loaded information, except the compounds information. Therefore, the user can edit genes, proteins and reactions. Moreover, new genes, proteins and/or reactions that are not available can be added to the local database. For example, a new reaction can be added by selecting existing compounds and assigning them with stoichiometric coefficients.

Merlin's Integrate: This operation compares the enzyme information retrieved by similarity with the data already available in the local database. The common unique identifier used for cross-referencing information is the locus tag. In case of conflict between the local database information and the *BLAST* data, the user can select which data should be automatically preferred or if the data should

[1] ftp://ftp.genome.jp/pub/kegg/

be merged. In the later case, the user will have to resolve, manually, each conflict that arises from the data integration.

Merlin's SBML Builder: This operation allows the user to export the model, currently stored in a relational database to the well accepted *SBML* format. This feature allows the user to employ the model in other applications, such as OptFluxRocha et al. (2010), very easily.

3.4 BLAST information classifier(scorer)

The BLAST information viewer allows the user to visualise and edit the information in a table format, where the candidate protein names and EC numbers are automatically scored and displayed in "*drop down boxes*". In this work a BLAST hits scorer is proposed, where the candidate EC numbers of each gene are ranked according to the following proposition: the global confidence level S for the assignment of an EC number to a certain gene, can be assessed from the frequency score (S_1) and the taxonomy score (S_2) according to equation 1.

$$S = \alpha S_1 + (1 - \alpha) S_2 \tag{1}$$

Where α ($0 < \alpha < 1$) determines the relative weight given to the frequency and taxonomy scores. S_1 is the frequency of each EC number divided by the global number of hits for such gene, S_2 is the taxonomic score for the first j hits (default value $j = 3$) of each distinct EC number, as depicted in equation 2.

$$S_2 = \sum_{i=0}^{j} ti \left(\frac{1}{t_{max} j (1 - (j - k)\beta)} \right) \tag{2}$$

Where ti is the taxonomic frequency determined by counting up the number of common taxa between the case study and the homologue organism, t_{max} is the maximum taxonomic frequency value, obtained by adding up the case study taxa, and (k) is the number of available hits of a given EC number ($0 < k < j$). β (default value $\beta = 0,15; 0 < \beta < 1$) is the penalty cost, implemented in equation 2, for the difference between the defined number of hits j and the number of available ones k. β is used to avoid false positives, as it penalises the taxonomy score for EC numbers that may have been falsely annotated, since there may exist very few hits for such assignment.

Using *S. cerevisiae* as an example for the target taxa:

- *Eukaryota; Fungi; Dikarya; Ascomycota; Saccharomycotina; Saccharomycetes; Saccharomycetales; Saccharomycetaceae; Saccharomyces; Saccharomyces cerevisiae*

The t_{max} value is 10.

Consider the taxonomic classification of the following possible BLAST hits which assign the same EC Number to a specific *S. cerevisiae* gene:

- ***Eukaryota; Fungi; Dikarya; Ascomycota; Saccharomycotina; Saccharomycetes; Saccharomycetales; Saccharomycetaceae;*** *Kluyveromyces; Kluyveromyces lactis*
- ***Eukaryota; Fungi; Dikarya; Ascomycota; Saccharomycotina; Saccharomycetes; Saccharomycetales; Saccharomycetaceae;*** *Kluyveromyces; Kluyveromyces marxianus*
- *Proteobacteria; Gammaproteobacteria; Enterobacteriales; Enterobacteriaceae; Escherichia; Escherichia coli K-12*

The taxonomic frequencies calculated for the above hits by adding up the bold taxa, are $8, 8, and 0$. Hence, according to equation 2 and using the defaults, $S_2 = 0.53$.

If, for the same *S. cerevisiae* gene, only the eukaryotic organisms were retrieved by similarity, the number of available hits would be $k = 2$. Therefore, as the default number of hits would be $j = 3$, the penalty cost β would influence the value of the taxonomic score S_2, which would be $S_2 = 6.27$. Hence, despite the penalty cost, the taxonomy score value would be higher than in the previous case.

Although the default α value is set to $\alpha = 0.5$, it can be modified (in the BLAST data viewer). Several factors may influence the value that the user assigns to this parameter, such as the studied organism being a prokaryote or eukaryote. As there are more sequenced and annotated prokaryotes than eukaryotes, it is expected that a higher α value is assigned to prokaryotes (based on data not shown), since the frequency value S_1 will be higher. On the other hand, well studied organisms (either prokaryotes or eukaryotes) may also have high α values, as they will probably get more BLAST hits (*e.g.* several strains of the same specie), increasing the frequency score S_1.

At last, Merlin automatically selects for (re-)annotation (ticks the Select check box on the BLAST information viewer) the results that have the highest global confidence values (S), but only if S is higher than the confidence level threshold of $S = 0,5$. Nevertheless, the user can perform a dynamic annotation since all the results are available, for user evaluation, in drop down boxes.

4. METHODOLOGY

Two of the most well studied organisms were selected for Merlin's validation: *E. coli str. K-12 substr. W3110* and *S. cerevisiae*. For each organism a project was created within Merlin, and each project's local database was loaded with the information available on *KEGG* for such organism (using Merlin's load database operation).

The relevance of the results automatically selected by Merlin was assessed by evaluating the integration of the information retrieved by homology with the data loaded from the *KEGG* database. Therefore, the enzymes selected by the *BLAST* scorer were matched to the information loaded from *KEGG*.

As *KEGG* only assigns enzymes to genes, only proteins that have an EC number assigned could be compared to the proteins available at the local database, hence only the enzymes were matched. Names of genes that encode other proteins could not be integrated. However, this is compatible with the fact that metabolic models use only metabolic genes.

Table 1. BLAST parameters.

Parameter	Value
Matrix	BLOSUM62
E-Value (maximum)	1e-30
Word Size	3
Algorithm	blastp
Remote database	All non-redundant GenBank CDS
Max hits number	100
j (minimum number of hits)	3
β (penalty)	0.15

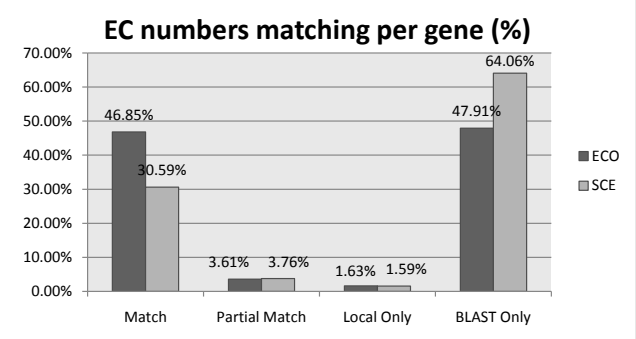

Fig. 3. Percentage of EC numbers matching.

To assess the accuracy of the software, when retrieving homology information, the assignment of the gene names was evaluated. All genes were selected instead of using only genes that encode enzymes, seeing as these data are retrieved from the genes themselves when using the non-redundant database.

For *E. coli* the α value used to score the *BLAST* hits was $\alpha = 0.4$. However, since *S. cerevisiae*, though being also a well studied organism, is a Eukaryote, the α value was set to $\alpha = 0, 2$.

Table 1 characterises the parameters used to perform similarity searches for the case studies. The presented values are Merlin's default standards.

5. RESULTS AND DISCUSSION

The matching of the *BLAST* data with the *KEGG* database is shown in Table 2:

Table 2. Number of genes (un)matched for *E. coli K-12* and *S. cerevisiae*.

	EC numbers		protein names	
	E. coli	*S. cerevisiae*	*E. coli*	*S. cerevisiae*
Match	903	842	766	674
Partial Match	22	31	-	-
Distinct	68	73	227	272
Only KEGG	44	64	44	64
Only BLAST	395	727	395	727
Total	1432	1732	1432	1732

As demonstrated on Table 2 the studied organisms had similar distributions for gene matching between both databases, either on EC numbers or protein names.

As depicted in Figure 3 and Table 2, the results for the EC numbers integration are similar for *E. coli* and *S. cerevisiae*. For the bacterium there were a total of 1432 genes which encoded enzymes. More than 60% of those genes were assigned with the same protein by *KEGG* and by similarity. For the yeast, 1732 enzyme encoding genes were identified, with almost 50% of the genes being assigned with the same enzyme in both databases. For the two organisms less than 2% of the genes assigned by *KEGG* were only partially matched by Merlin's similarity search for homologues.

For both organisms less than 5% of the genes were assigned with different EC numbers by similarity and on *KEGG*. Most of the cases were genes that encoded an incomplete EC number on one database and the complete EC umber on the other database

Over 3% of the genes were assigned as enzyme coding genes in the local database, but no similarity was found by Merlin when *BLAST* was performed. The most unexpected results were obtained on the genes that were only appraised by Merlin's similarity search. For *E. coli*, Merlin identified 395 candidate genes, which may encode enzymes, with scores beyond the confidence level threshold. Moreover, for the yeast, Merlin identified (42%) 727 candidate genes from the 1732 total enzyme coding genes. Such high percentage is explained by the low α value used for eukaryotes. As explained before, eukaryotes should be scored with a lower α value since there are less sequenced organisms with complex structures, thus the taxonomy score should be preferred over the frequency score.

The candidate genes should be verified with organism specific databases such as EcoCyc [2] for *E. coli* or SGD [3] for *S. cerevisiae*. Nevertheless, the enzymes assigned by the candidate genes may also be helpful for filling gaps or find alternative pathways in the metabolic model. Hence, the information provided by homology, if confirmed by experimental evidences, can be useful for the development of a more complete and robust metabolic model.

Table 3. Genes names matching for *E. coli K-12* and *S. cerevisiae*.

	E. coli	*S. cerevisiae*
Match	4830	4148
Distinct	0	9
Only Local	2	337
Only BLAST	5	48
Total	4837	4542

As expected the gene names matching was very straightforward with over 90% matches for both bacterium and yeast (91.33% and 99.86% respectively). The distinct gene names cases are possibly synonyms not available in the local database. However, the gene names only available in one of the databases (*BLAST* or *KEGG*) are related to genes that are missing from the other database. This issue concerns the version of the annotation, as Merlin retrieves the most up to date homology information and allows the user to employ the most up to date *GenBank* fasta files for similarity search. Yet *KEGG* annotations may be outdated. This problem may also have been an issue for the EC numbers and protein names integration.

[2] https://www.ecocyc.org
[3] http://www.yeastgenome.org/

6. CONCLUSIONS

With the ever increasing amount of genomic data becoming available, every tool developed to interpret and make sense of such data is useful, as appraising such bulk loads of data can be very tedious and time consuming.

Merlin is proposed as a user-friendly tool, which allows to attain comprehensible information and perform a semi-automated dynamic annotation, relying in the most up to date information, available in the *GenBank* database, and integrate such data with the information already available at the well accepted *KEGG* database, for the development of a more robust metabolic model. In this paper it was shown that Merlin identifies almost all of the information provided by *KEGG*, and also specifies candidate EC numbers for other genes. Moreover, such model may be retrieved in the Systems Biology Markup Language for *in silico* processing.

Merlin obtains the most up-to-date information from online databases, allowing the user to perform regular similarity searches and update the genome annotation.

7. FUTURE WORK

Merlin will be embedded with other tools such as PSORT-B, SignalP or TargetP, for subcellular protein localization and will be able to integrate other databases, namely UniProt. Reactions for which the EC number is not available will also be added to the metabolic model. A specific operation will be added for the inclusion of the biomass formation equation, using the compounds available at the local database.

REFERENCES

Altschul, S., Gish, W., Miller, W., Myers, E., and Lipman, D. (1990). Basic local alignment search tool. *JOURNAL OF MOLECULAR BIOLOGY*, 215(3), 403–410.

Borodovsky, M. and Mcininch, J. (1993). Genmark - parallel gene recognition for both dna strands. *COMPUTERS & CHEMISTRY*, 17(2), 123–133. 2ND INTERNATIONAL WORKSHOP ON OPEN PROBLEMS OF COMPUTATIONAL MOLECULAR BIOLOGY, TELLURIDE, CO, JUL 19-AUG 02, 1992.

Chalmel, F., Lardenois, A., Thompson, J., Muller, J., Sahel, J., Leveillard, T., and Poch, O. (2005). Goanno: Go annotation based on multiple alignment. *BIOINFORMATICS*, 21(9), 2095–2096. doi: 10.1093/bioinformatics/bti252.

Foissac, S. and Schiex, T. (2005). Integrating alternative splicing detection into gene prediction. *BMC BIOINFORMATICS*, 6. doi:10.1186/1471-2105-6-25.

Francke, C., Siezen, R., and Teusink, B. (2005). Reconstructing the metabolic network of a bacterium from its genome. *TRENDS IN MICROBIOLOGY*, 13(11), 550–558. doi:10.1016/j.tim.2005.09.001.

Holland, R.C.G., Down, T.A., Pocock, M., Prlic, A., Huen, D., James, K., Foisy, S., Draeger, A., Yates, A., Heuer, M., and Schreiber, M.J. (2008). Biojava: an open-source framework for bioinformatics. *BIOINFORMATICS*, 24(18), 2096–2097. doi:10.1093/bioinformatics/btn397.

Hucka, M. and *et al* (2003). The systems biology markup language (sbml): a medium for representation and exchange of biochemical network models. *BIOINFORMATICS*, 19(4), 524–531. doi: 10.1093/bioinformatics/btg015.

Joshi, T., Chen, Y., Becker, J., Alexandrov, N., and Xu, D. (2004). Function prediction for hypothetical proteins in yeast saccharomyces cerevisiae using multiple sources of high-throughput data. *OMICS: A Journal of Integrative Biology*, 17–22. 8th World Multi-Conference on Systemics, Cybernetics and Informatics, Orlando, FL, JUL 18-21, 2004.

Kanehisa, M. and Goto, S. (2000). Kegg: Kyoto encyclopedia of genes and genomes. *NUCLEIC ACIDS RESEARCH*, 28(1), 27–30.

Krause, L., McHardy, A.C., Nattkemper, T.W., Puehler, A., Stoye, J., and Meyer, F. (2007). Gismo - gene identification using a support vector machine for orf classification. *NUCLEIC ACIDS RESEARCH*, 35(2), 540–549. doi:10.1093/nar/gkl1083.

Lipman, D. and Pearson, W. (1985). Rapid and sensitive protein similarity searches. *SCIENCE*, 227(4693), 1435–1441.

Markowitz, V.M., Korzeniewski, F., Palaniappan, K., Szeto, E., Werner, G., Padki, A., Zhao, X., Dubchak, I., Hugenholtz, P., Anderson, I., Lykidis, A., Mavromatis, K., Ivanova, N., and Kyrpides, N.C. (2006). The integrated microbial genomes (img) system. *NUCLEIC ACIDS RESEARCH*, 34(Sp. Iss. SI), D344–D348. doi: 10.1093/nar/gkj024.

Medigue, C. and Moszer, I. (2007). Annotation, comparison and databases for hundreds of bacterial genomes. *RESEARCH IN MICROBIOLOGY*, 158(10), 724–736. doi:10.1016/j.resmic.2007.09.009.

Ouzounis, C. and Karp, P. (2002). The past, present and future of genome-wide re-annotation. *Genome Biology*, 3(2). doi:10.1186/gb-2002-3-2-comment2001. URL `http://genomebiology.com/2002/3/2/comment/2001`.

Rocha, I., Förster, J., and Nielsen, J. (2008). Design and application of genome-scale reconstructed metabolic models. *Methods in molecular biology (Clifton, NJ)*, 416, 409.

Rocha, I., Maia, P., Evangelista, P., Vilaca, P., Soares, S., Pinto, J., Nielsen, J., Patil, K., Ferreira, E., and Rocha, M. (2010). OptFlux: an open-source software platform for in silico metabolic engineering. *BMC Systems Biology*, 4(1), 45.

Salzberg, S., Delcher, A., Kasif, S., and White, O. (1998). Microbial gene identification using interpolated markov models. *NUCLEIC ACIDS RESEARCH*, 26(2), 544–548.

Salzberg, S.L. (2007). Genome re-annotation: a wiki solution? *GENOME BIOLOGY*, 8(1). doi:10.1186/gb-2007-8-1-102.

Stajich, J. and *et al* (2002). The bioperl toolkit: Perl modules for the life sciences. *GENOME RESEARCH*, 12(10), 1611–1618. doi:10.1101/gr.361602.

Tamaki, S., Arakawa, K., Kono, N., and Tomita, M. (2007). Restauro-G: a rapid genome re-annotation system for comparative genomics. *Genomics, Proteomics & Bioinformatics*, 5(1), 53–58.

Wang, K. (2006). Gene-function wiki would let biologists pool worldwide resources. *NATURE*, 439(7076), 534. doi:10.1038/439534a.