Uppsala University

Applied Bioinformatics

Project Report

Local Visualization Platform
for
Manual Gene Annotation Curation

Authors:

Efthymia Chantzi
Email: Efthymia.Chantzi.0787@student.uu.se
MSc programme in Bioinformatics

Stephen Omondi Otieno Anyango
Email:Stephenomondiotieno.Anyango.4647@student.uu.se
MSc programme in Bioinformatics

Supervisor:

Katarzyna Zaremba-Niedzwiedzka
Email:katarzyna.zaremba@icm.uu.se
Post-Doc, Ettema Lab

October 16, 2015

Acknowledgements

This project was proposed and supported by Thijs Ettema's Lab. We are grateful to Thijs as the PI of the lab for the opportunity to work a project in the Lab. We also are most grateful for Katarzyna for the mentorship and continued assistance in making us feel comfortable and for provision of one of the in-house scripts used in the project. We also thank Felix, Jimmy and Anja, who provided insight and expertise that greatly assisted our effort.

Abstract

Manual curation requires integrated and highly customizable visualization of one or more genomes. Apart from high-quality visualization, the ability of research labs to have locally available tools is fundamental in increasing the amount of security and customizability of such platforms. In this project, we install and configure a locally-hosted, in-house mirror of the UCSC genome browser and also add features for the visualization of KEGG metabolic pathways on track details. This work extends the information available to an expert curator to complement results of automated annotation by constructing an organism-specific KEGG metabolic pathway for a gene of interest. The current implementation can serve as the basis for the development of a long-term annotation platform that offers full-featured genome annotations and high-quality visualization. We present a functional installation and extension accessible only to members of the "*Thijs Ettema's*" Lab.

Contents

List of Figures

1 Introduction

Novel sequencing technologies are delivering a huge number of new sequences, both finished and draft genomes, all of which call for continuous improvement of genome annotation procedures. However, platforms that aid in high quality automated annotations are still in infancy and drive the need for integrated, flexible and high-quality visualization tools to assist manual curators and domain experts in the verification process [2]. The use of automated annotation alone can be misleading and hence, there is great demand for expert curators. These curators sift through relevant literature for supporting experimental evidence of the annotation. They also leverage good visualization tools for comparative analysis. Identification of synteny, homology and protein families [3] provided by high-level bioinformatic algorithms, such as Hidden Markov Models and Profile-based methods, constitute major parts of the manual curation process. In addition, the integration of gene ontology, metabolic pathways and biological networks[3], make fundamental contributions to the quality of this process. However, this leads to data-intensive workload that spans from a single gene to a set of selected genes either in one or multiple genomes. Thus, there is great need for tools that are customizable and integrated.

1.1 The "ASGARD" project

The "ASGARD" project constitutes an ongoing large-scale genome annotation project that is currently being held by the researchers in *"Thijs Ettema-Lab"* [4], which is established in Uppsala University. The underlying purpose of their research lies in the exploration of microbial diversity using novel cultivation-independent approaches, such as metagenomics and single-cell genomics, which are expected to shed light on the reconstruction of the origin and evolutionary history of the eukaryotic cell.

1.1.1 TACK archaea

"While surveying microbial diversity in deep marine sediments influenced by hydrothermal activity from the Arctic Mid-Ocean Ridge, gene sequences belonging to uncultivated archaeal candidate lineages were identified" [1]. More specifically, this recently published work, revealed *Lokiarchaeota*; a novel candidate archaeal phylum, which forms a monophyletic group with eukaryotes in phylogenomic analyses, and whose genomes encode a wide spectrum of eukaryotic signature proteins. As shown in figure 1, eukaryotes emerge as a sister group to or from within the archaeal lineage TACK superphylum, based on phylogenetic analysis of universal protein data. These results provide strong evidence for an archaeal ancestor of eukaryotes and a different viewpoint on the domains of life as they are known.

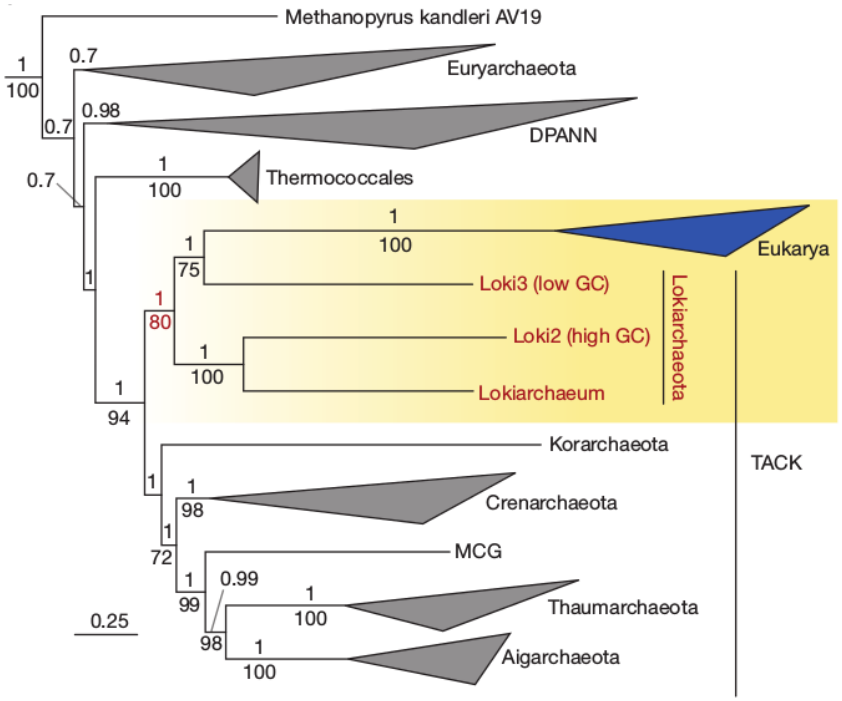

Figure 1: *Complex archaea that bridge the gap between prokaryotes and eukaryotes [1]*

1.2 Annotation Curation Platform

This project aims at the development of an annotation curation platform that would serve as a solution to the aforementioned "*ASGARD*" project. Given that there is great demand from the research group with reference to a long-term annotation platform, the development of a locally installed, customizable, integrated and well-documented visualization tool is of the utmost importance. This would serve as a solution for the manual curation of various genes that are of particular interest for the research group. Sensitive unpublished genomic data are distributed only over the local network and hence, they are protected. Other than that, a customizable design enables flexible potential modifications, since requirements and needs change over time.

1.2.1 Tools considered

Depending on the fact that there is need for a locally installed genome browser that would allow the visualization of KEGG [5] metabolic pathway maps, the group considered some of the available open-source options; Generic Genome Browser (GBrowse) [6], Artemis [7], Ensemble [8] and UCSC[1] Genome Browser [9].

Although all of these genome browsers were developed in response to the problem of visualizing genome assemblies and associated data, each one of these has different properties as a software system. This fact indicates that one of them may be more appropriate than the others on different research occasions. In this case, our client settled on the UCSC Genome Browser [9].

[1] University of California Santa Cruz

1.3 UCSC Genome Browser

Launched in 2001, the UCSC Genome Browser [9] is a web-based graphical viewer for genomic data, presenting an integrated visualization of a wide variety of genomic data mapped to genomic coordinates. It was created to provide a graphical visualization of the very large amount of genomic sequencing data produced by the Human Genome Project. However, the UCSC Genome Browser has grown considerably over the years, mainly because of the addition of other genome assemblies. Apart from the simple coordinate-based interface, a lot of supporting tools, such as BLAT [10], have also been developed around it, after the respective requests from the user community. As it stands now, it offers a fast and multifunctional experience by consisting of many different programs that communicate between the user and database.

The properties that led the research group to decide on further extension of the UCSC Genome Browser are essentially the ability to collate annotations of many types and eligible visualization of gene predictions, mRNA alignments, epigenomic data, conservation scores and variation data. Moreover, it supports various formats used in high-throughput sequencing data analysis, such as Binary Alignment Format (BAM), Variant Calling Format (VCF), bigBed and bigWig, allowing rapid visualization of large datasets via local hosting [11]. It is mainly for these reasons that the UCSC Genome Browser was selected for extension; to incorporate metabolic pathways from KEGG [5], as being the first additional view that facilitates the crucial curation process.

In this work, we present the methods employed, challenges encountered, milestones achieved and propose potential future work regarding the installation, configuration and extension of the UCSC Genome Browser.

2 Materials and Methods

The Lab provided an option to use UCSC Genome Browser and our major tasks were to do a local installation, load some of their data as proof of concept, and then make the necessary changes in the application to visualize KEGG pathways.

2.1 Installation and Configuration

First and foremost, the UCSC Genome browser was installed and configured locally. Its source code was cloned from GitHub [12] at `https://github.com/ucscGenomeBrowser/kent.git`. At this point, it is important to mention that this tool is a web-based application implemented in *C* and running on *Apache2* and *MySQL* back-end. The source code includes a lot of neat utilities used for data processing and automation of frequent data loading tasks. Secondly, the web documents (HTML, Javascript, CSS) for the application were downloaded into the local Apache web root directory from their FTP site at `ftp://hgdownload.cse.ucsc.edu/htdocs`. Furthermore, the configuration of Apache2 was necessary, in order to allow execution of *CGI*[2] *includes* [13] and following of *symbolic links*. The functionality of the genome browser is provided by the compilation of the C code into *CGI* [13] programs, which in turn are executed during navigation around the web application.

[2]Common Gateway Interface

2.2 Loading Archaeon Loki genome

Assembly genomic data for the *Lokiarcheota* (NCBI Accession: JYIM00000000) was downloaded from *NCBI GenBank FTP* and loaded into a *MySQL* database, using a non-automated pipeline consisting of several utilities of the UCSC Genome Browser [9]. All GenBank formatted data was converted to *Browser-Extensible Data* (BED) format using in-house Perl scripts provided by the client. The data was split into four different categories for track loading; contigs, genes, CDS[3] and RNAs. It should be mentionted that CDS data was further converted into *GenePred* format, using one of the utilities provided by the UCSC Genome Browser, named *bedToGenePred*. Furthermore, a track file including the four different track categories and GC percent was created. It was loaded by the provided *hgLoadTrackDb* utility.

2.3 KEGG MySQL Tables

The visualization of KEGG [5] pathways necessitates the addition of two extra MySQL tables in the currently locally installed database of the UCSC Genome Browser [9]. More precisely, the first one of them pertains to the required KEGG data, including names of the general pathways, identifiers of the orthology groups (KOs) and enzyme numbers (ECs) that respond to the respective reactions. This table is named *KEGG_PathwayList* and the schema of its parent database "*hgcentral*" is attached to Appendix A.

The second table, named as *org_kegg_data*, refers to the data that are specific to the studied organism; *Lokiarcheota*. It includes the name of the genes as well as identifiers of the orthology groups (KOs), where each one of them belongs to. This list of data is provided by the client. The schema of its parent database is also attached to Appendix A.

2.4 Visualization of KEGG pathways

The client-requested visualization of KEGG metabolic pathway maps is performed by the contribution of the open-source Biopython toolset [14]. This is necessary due to the fact that KEGG was recently commercialized and would not suffice for our client's requirements.

More specifically, three different packages from Biopython were used: *KEGG.REST*, *KEGG.KGML* and *Graphics.KGML_vis*. The first one enables the downloading of a KGML[4] [15] file and an image of the general pathway via the REST-style KEGG API [16]. This file contains the positions of the various graphical objects in the image of the general pathway map. The second package allows the parsing of such a KGML file. Since, the acquisition and parsing of the KGML are guaranteed, the third package provides the ability to intervene in the properties of the graphical objects and thus, reconstruct the organism-specific pathway image, by overlaying the general image with the information of interest, which stems from the added MySQL tables described in the previous section (2.3).

Given the specific requirements suggested by the client, we modified the source code of the package *Graphics.KGML_vis*, so that it produces the desirable graphical results. In addition, we fixed some bugs before utilizing the package. All these interventions from our side, are thoroughly described in the wiki documentation page that is privately hosted by the client.

[3] Coding Sequences
[4] KEGG Markup Language

2.5 Creation of additional web pages

The visualization of the KEGG pathways is built on two different web pages that are incorporated in the local installation of the UCSC Genome Browser. These pages are implemented by using Python CGI scripts [17]. The combination of Python with a web server leads to dynamic web sites that are not based on files in the file system, but rather on programs, which are run by the server after a request. These programs generate the content that is returned to the user. This is the content of our web pages, which has been produced using *HTML*, *Javascript* and *Bootstrap 3* for the styling of the user interface.

We have created in total two additional web pages. The first one lists all the KEGG pathways that corresponds to the user's selection of a particular gene from the track viewer. Additionally, there is a list of all the other genes that are part of the same genome and participate in the same pathway. As far as the second web page is concerned, it is triggered by clicking the button of a pathway from the first page. It is obtained as a separate window and displays the currently selected pathway. There is also a zoom option, which can be activated/deactivated by the user. The pages are publicly available on BitBucket [18] at `https://bitbucket.org/steveomondi/ucscgb/src` under the "*cgi-bin*" subdirectory.

2.5.1 Integration into UCSC Genome Browser

The last part is the integration of the previously described web pages into the UCSC Genome Browser. For this , an additional C module, *thijslab.c*, is created, compiled and installed along with the whole source code. It is executed when a gene is clicked on the genome viewer. In this way, it passes a number of parameters to the respective python script, which prints the content of the KEGG pathway section.

2.5.2 Performance

Given performance in terms of loading time is key, the two additional MySQL tables were indexed (B-tree option). This ensured faster data retrieval. Also, the reconstructed KEGG pathway maps that are adjusted to the requested organism-specific information, are stored and viewed locally. We have enabled disk-caching of these images on first access, so that the performance is improved on multiple views of them. In case a pathway that has already been cached is requested, its loading is a lot faster, as there is no network latency overhead.

2.5.3 Extensibility

It is true that there are expectations on a long-term annotation platform from the side of our client. Therefore, we have considered the importance of extensibility of the system, even if in its infancy. This accounts for the fact that we have chosen a template-based development that allows the flexible incorporation of additional sections.

3 Results

The implemented KEGG visualization platform comprises a part of the UCSC Genome Browser, which is locally installed and loaded only with *Lokiarchaeota* genomic data that correspond to the needs of our clients. It can be accessed either only within the local network of their lab, or by their personal machines after the appropriate installation and configuration. Hence, the results of this project can only be demonstrated by following one of the two aforementioned ways. However, we attach screenshots of the developed web application that demonstrate the most important results. These can be found in Appendix B.

Briefly, our clients are able to submit genomic data with putative genes to KEGG, which runs domains searches against its own database and provides an orthology identifier for each matched gene. This data is used by our extension of UCSC Genome Browser to reconstruct the organism-specific metabolic maps. We highlight in, red background, the graphical position of the requested gene and in yellow background, the positions of all the other genes of the same genome that belong to the reconstructed pathway. The purple-colored elements are other positions in the pathway for which there are no genes in the organism's genomic data.As for the labeling, each one of the highlighted elements is accompanied with the first fetched EC (enzyme) number a gene that is found there. In case, there is no EC number provided, the KEGG orthology identifier (KO) is displayed instead. Moreover, if more than one genes are found in a position, the total count of them is shown in parenthesis after the label tag.

4 Discussion

The present genome viewer enables the local visualization of KEGG metabolic pathways, which are adjusted to Loki Archeon genome data. It is fully functional on the local network of our clients. There is extensive documentation on a wiki page that is necessary for the installation, configuration, use and maintenance of the system.

Given we had only five weeks to develop this system, it is quite reasonable that there will be several improvements to be made and extensions to be included. It is important to highlight the challenges that we dealt with, so that they could be avoided in similar future approaches.

The evaluation of an existing software that is to be extended constitutes one of most significant tasks during the pre-developing phase. This process should always be done thoroughly, so that all alternatives and their evaluations are considered. In this way, future problems that would withhold the implementation are less likely to arise. Keen consideration should be taken in usability, ease of extension, portability and robustness of the underlying architecture. In our case, one of the major challenges was working with an existing web-application, which was developed in the C programming language in 2001. Nowadays, there exist several different languages that would render the same task a less cumbersome procedure and highly decrease the learning curve. Furthermore, although the structure of the application was modular, it took us time to follow its design in a level that we would be able to build our own extension on the top of it. In addition, it is true that we were missing test data, since Microbe or Archaea data are not available. This practically leads to an other challenging task, which is the manual loading of genomes. Therefore, the platform in its current version requires database management, which can be a complicated procedure for users who are not familiar with databases.

Last but not least, we encountered a significant problem regarding the second part of this

project, which is the visualization of KEGG pathways. Unfortunately, KEGG is no longer open-source. This problem was solved by using Biopython packages that allowed us to reconstruct the general pathway maps and overlay them with available organism-specific data.

5 Future Work

Provided that there are expectations from our client for the development of a long-term annotation platform, the future work could overwhelm the currently achieved work. However, it is always important to have a flexible and extensible basic implementation, which could facilitate the future work. Since we mostly worked on the basic core of a genome browser that incorporates the local visualization of KEGG pathways, we have considered some aspects that would be useful for the near future.

Optional editing of gene names, and possibly the addition of notes to provide support for such editing would add flesh toward a wholesome curation. Additionally, it is of the utmost importance to improve the usability concerning the database part of the system. More precisely, it is necessary to enable simpler loading of tracks and genomes via in-house scripts that would replace the cumbersome user involvement. Lastly, it would be useful to aim at better and more effective data organization that would boost the performance of the system.

In general, the group will have to evaluate the cost of loading genomic data and configuring tracks *vis-a -viz* the resources they have.

6 Division of Work

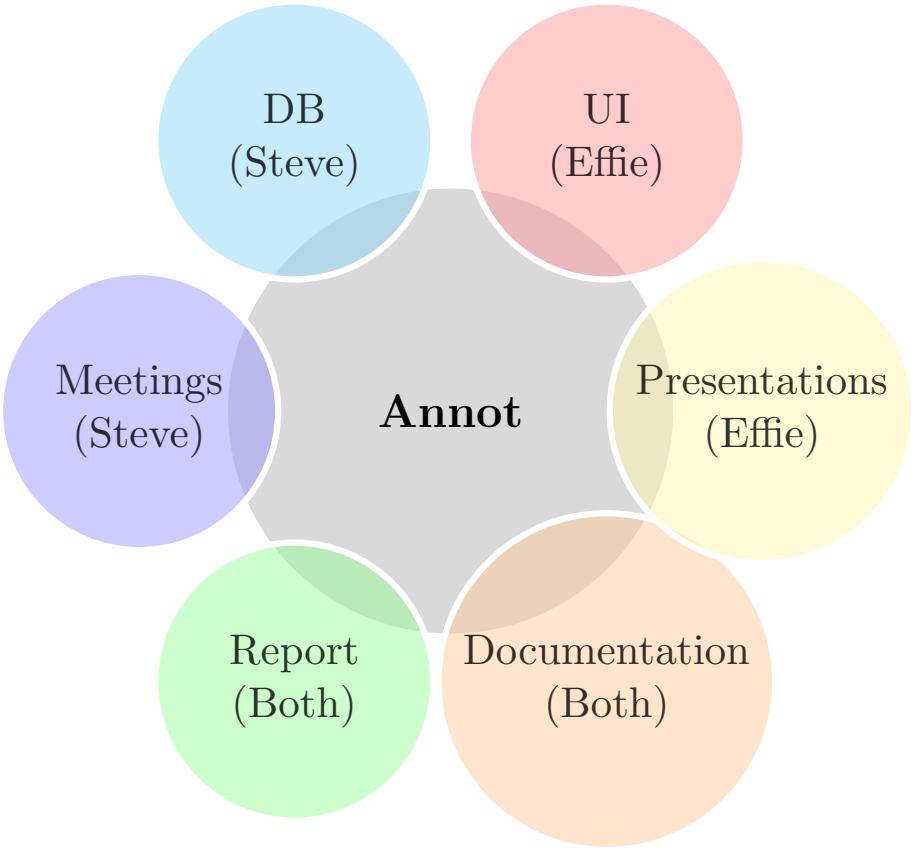

Figure 2: *Work division according to the submitted plan*

Our contributions to the project consist of two major parts; *database management* and *user interface development*, which co-interact. As indicated by our initial division of work in figure 2, each one of us embarked on a separate part. This was necessary to increase efficiency in a bid to complete the project in time. However, we were both actively involved in each other's part for questions, proposals, help and support. We were working together on a daily basis at a project room that our clients so graciously provided us with in their corridor. This helped our efforts a lot, since we had constant interaction and feedback from them. Also, the final presentation was written by both of us. We discussed its content and then it was split in two parts.

In general, there was very good communication and effective cooperation. Despite the challenges we encountered and the limited available time, we managed to fulfill our goals for this project.

References

[1] A. Spang *et al.*, "Complex archaea that bridge the gap between prokaryotes and eukaryotes," *Nature*, vol. 521, pp. 173–179, May 2015.

[2] W. Klimke *et al.*, "Solving the Problem: Genome Annotation Standards before the Data Deluge," *Stand Genomic Sci*, vol. 5, pp. 168–193, Oct 2011.

[3] A. Pujar *et al.*, "From manual curation to visualization of gene families and networks across Solanaceae plant species," *Database (Oxford)*, vol. 2013, p. bat028, 2013.

[4] "Thijs Ettema." http://www.scilifelab.se/researchers/thijs-ettema/. Accessed: October 6, 2015.

[5] "Kegg pathway Database." http://www.genome.jp/kegg/pathway.html. Accessed: October 6, 2015.

[6] L. D. Stein *et al.*, "The generic genome browser: a building block for a model organism system database," *Genome Res.*, vol. 12, pp. 1599–1610, Oct 2002.

[7] T. Carver *et al.*, "Artemis: an integrated platform for visualization and analysis of high-throughput sequence-based experimental data," *Bioinformatics*, vol. 28, pp. 464–469, Feb 2012.

[8] F. Cunningham *et al.*, "Ensembl 2015," *Nucleic Acids Res.*, vol. 43, pp. D662–669, Jan 2015.

[9] "Ucsc Genome Bioinformatics." http://genome.ucsc.edu/. Accessed: October 6, 2015.

[10] "Ucsc Genome Bioinformatics." https://genome.ucsc.edu/FAQ/FAQblat.html. Accessed: October 6, 2015.

[11] R. M. Kuhn *et al.*, "The UCSC genome browser and associated tools," *Brief. Bioinformatics*, vol. 14, pp. 144–161, Mar 2013.

[12] "GitHub." https://github.com/. Accessed: October 6, 2015.

[13] "CGI - Common Gateway interface." http://www.webopedia.com/TERM/C/CGI.html. Accessed: SOctober 6, 2015.

[14] "Biopython." http://biopython.org/wiki/Main_Page. Accessed: October 6, 2015.

[15] "KGML (kegg markup language)." http://www.kegg.jp/kegg/xml/. Accessed: October 6, 2015.

[16] "Kegg api." http://www.kegg.jp/kegg/docs/keggapi.html. Accessed: October 6, 2015.

[17] "HOWTO Python in the web." https://docs.python.org/2/howto/webservers.html. Accessed: October 6, 2015.

[18] "Bitbucket." https://bitbucket.org/. Accessed: October 6, 2015.

A Appendix A

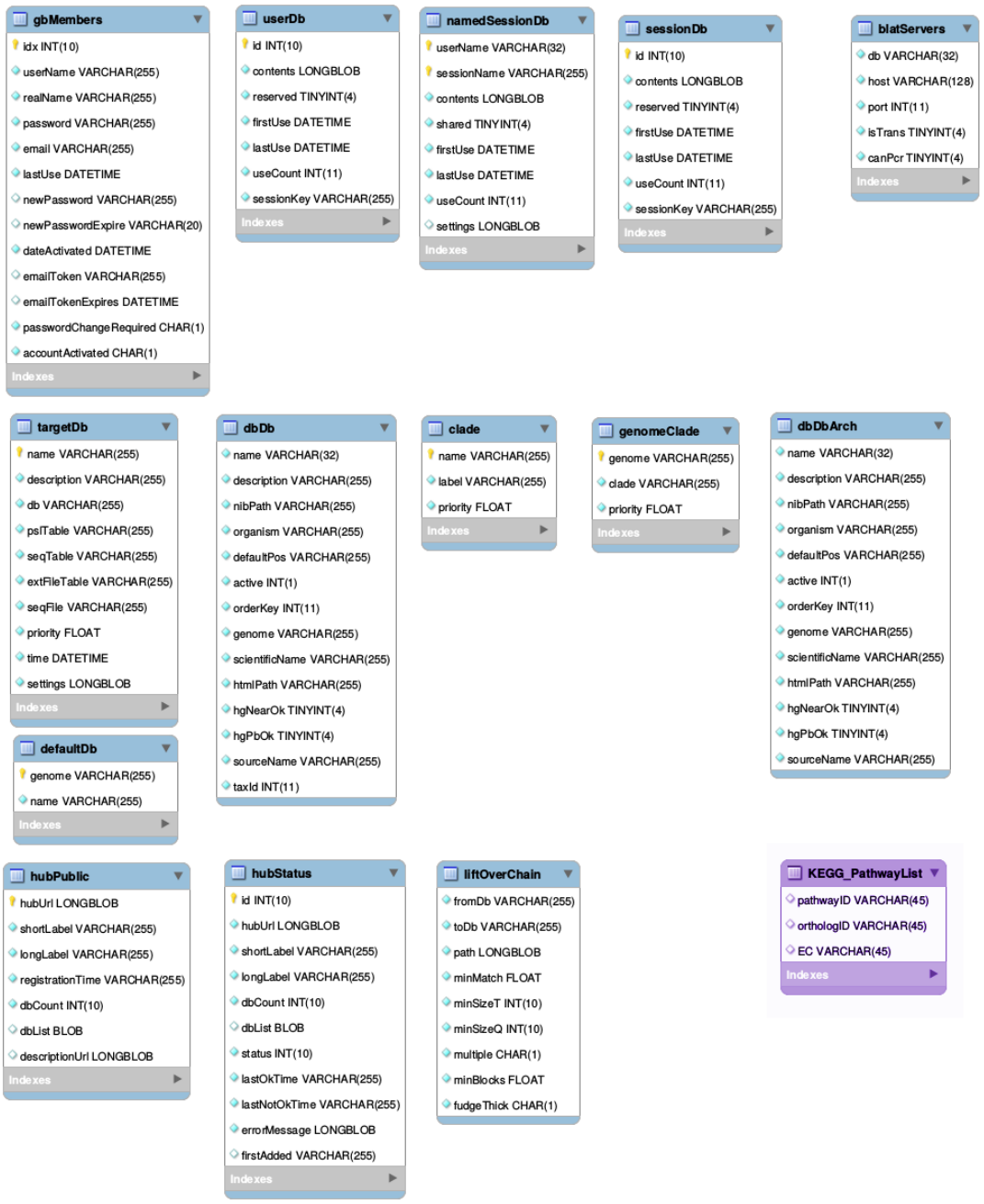

Figure 3: *Schema of "hgcentral" Database*

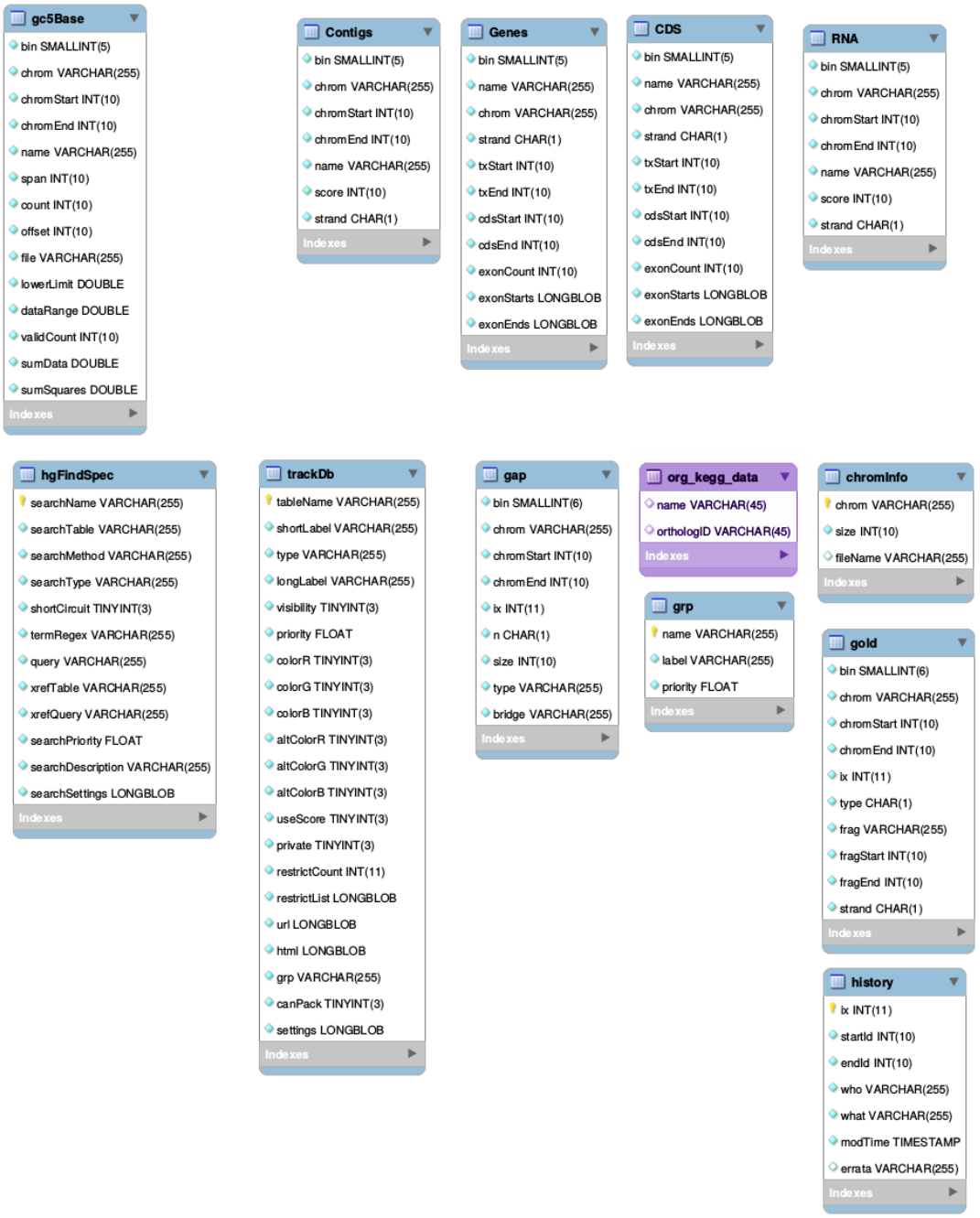

Figure 4: *Schema of Archeon Loki genome "arcLok1" Database*

B Appendix B

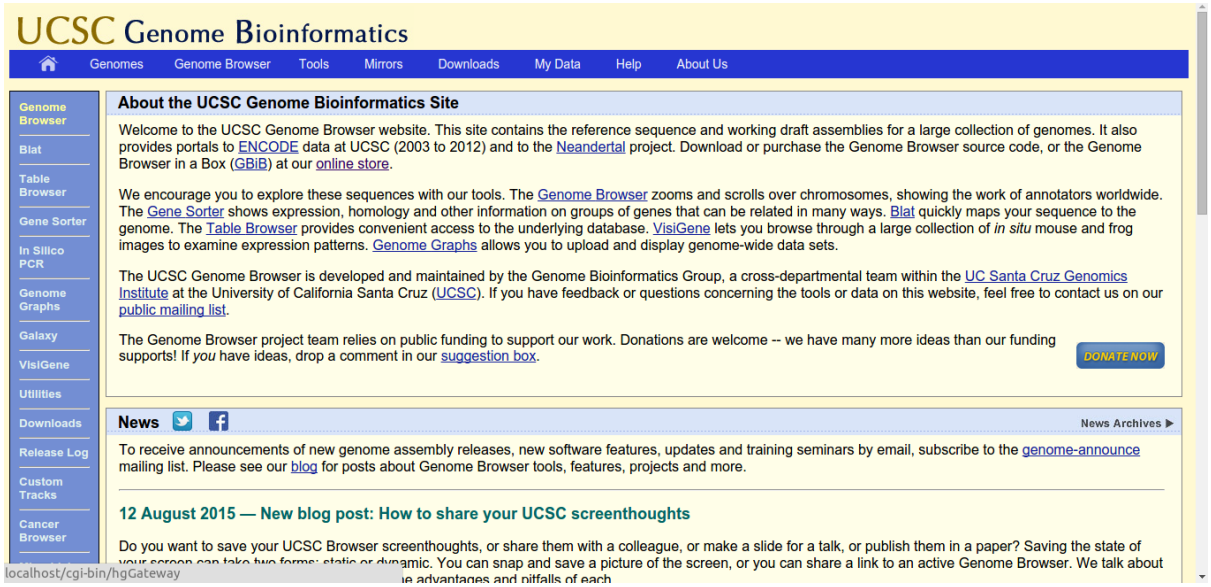

Figure 5: *Index page*

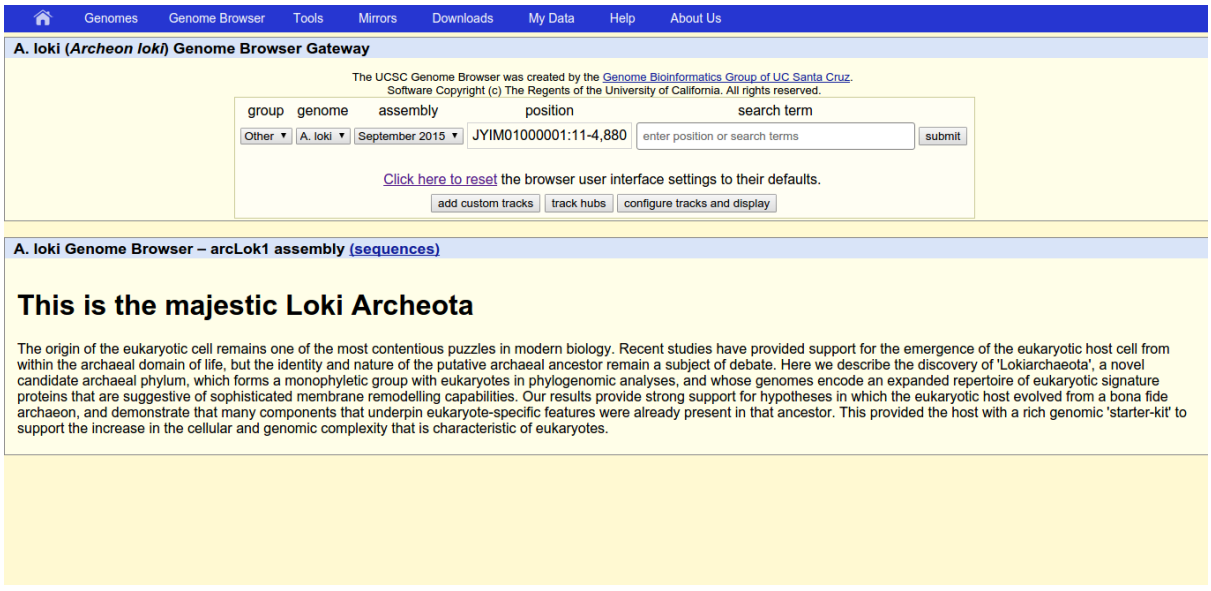

Figure 6: *Loki Archaeota Gateway page*

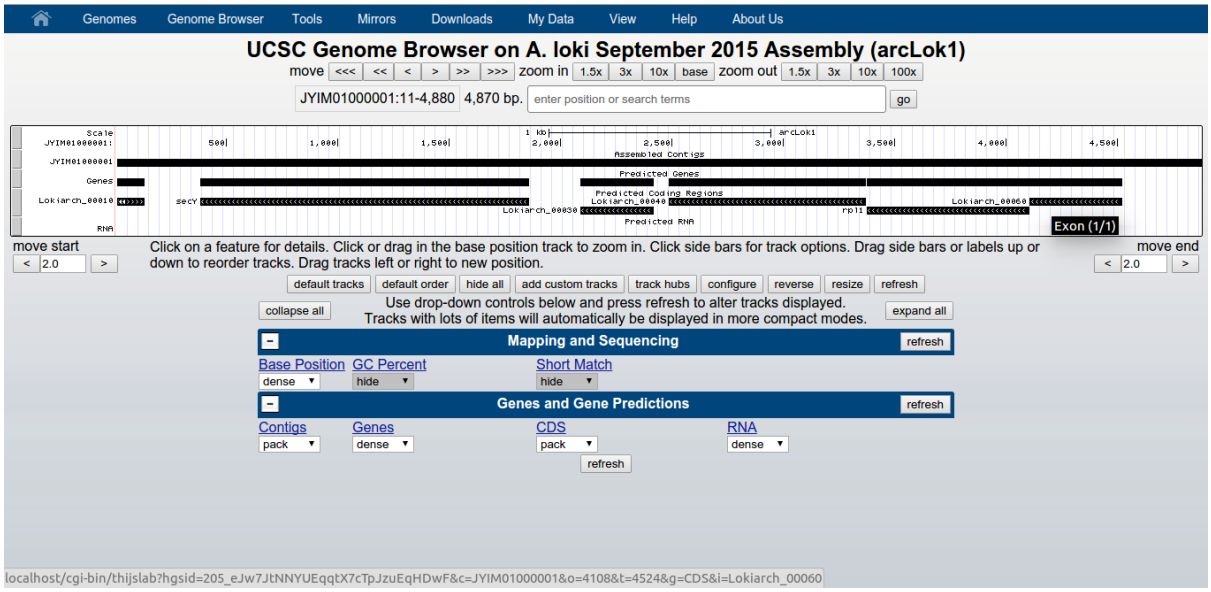

Figure 7: *Loki Archaeota tracks page*

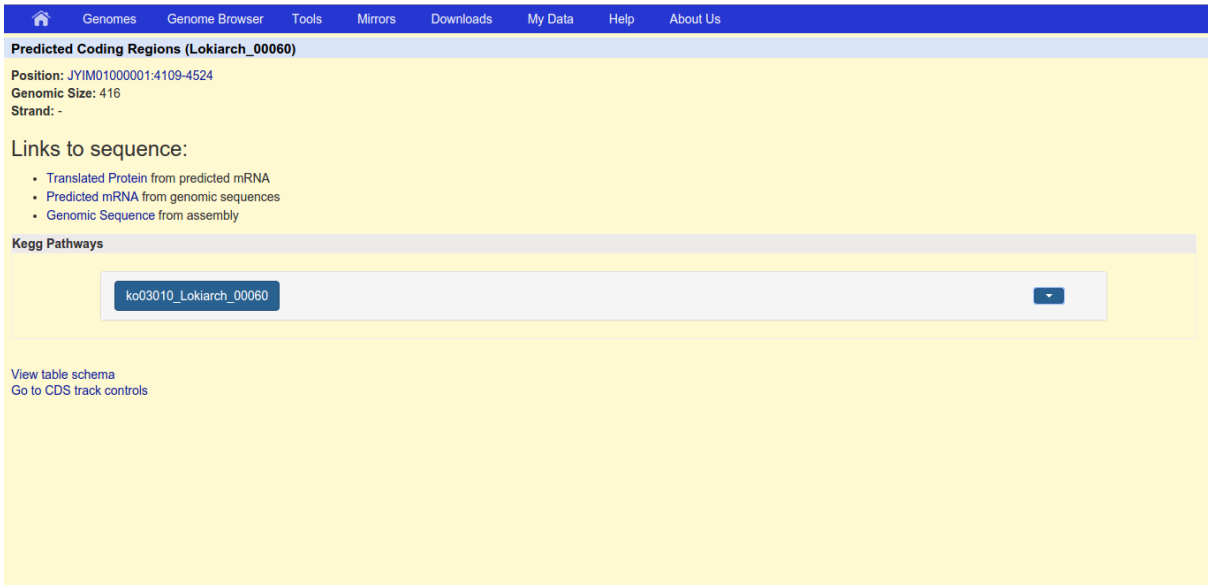

Figure 8: *Thijslab page*

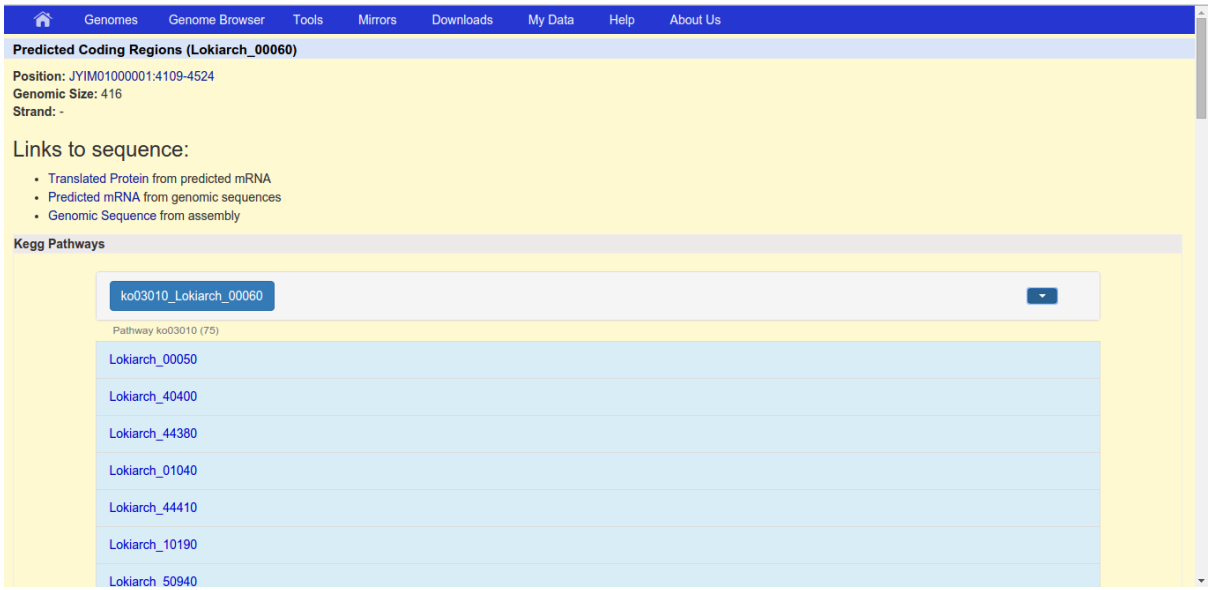

Figure 9: *Thijslab page ⟼ listing all Loki Archaea genes in pathway*

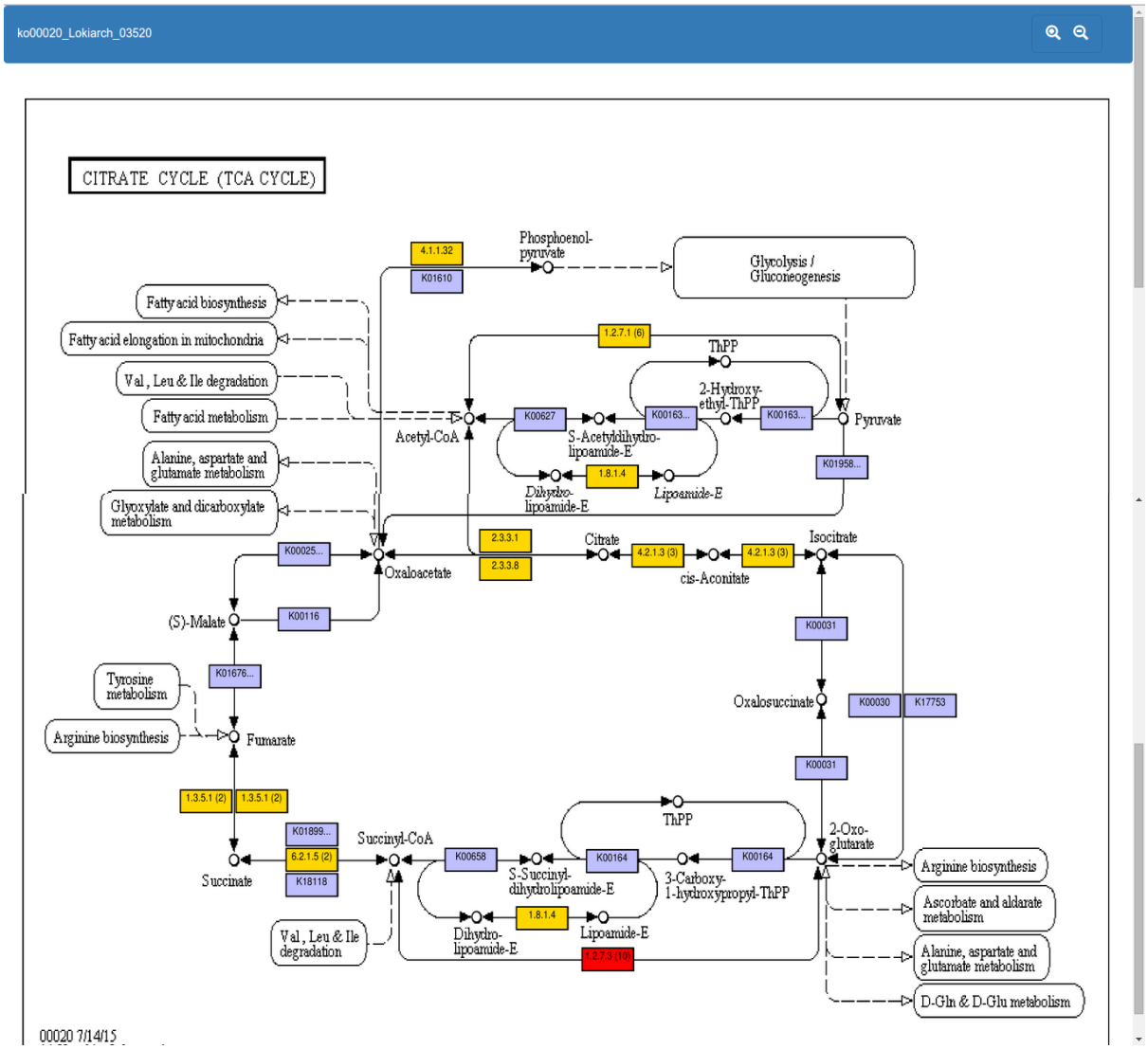

Figure 10: *Pathway ko00020 for gene Lokiarch_03520*

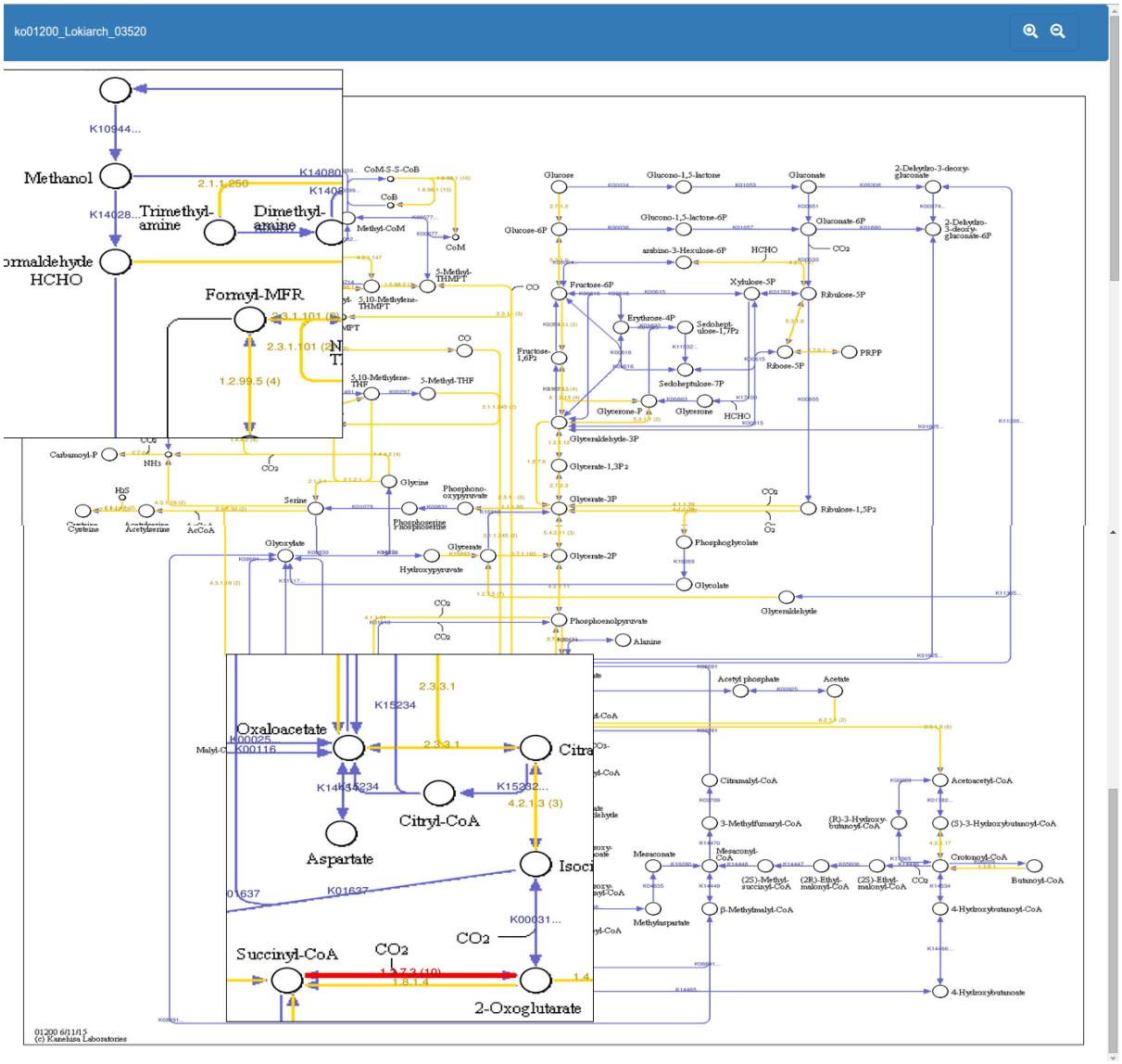

Figure 11: *Pathway ko01200 for gene Lokiarch_03520*