GA*L*

Genome Annotator *Light*

Version 1.0

User Guide

Authors:

Arijit Panda
Narendrakumar M. Chaudhari
Sucheta Tripathy*

Contact Emails: arijpanda@gmail.com and tsucheta@gmail.com

Developed at:
Computational Genomics Lab,
Structural Biology and Bioinformatics Division,
CSIR-Indian Institute of Chemical Biology,
Kolkata, India.

*Principal Investigator

Table of Contents

Introduction 2
- Getting Started 2
- System Requirements 2
- Quick Start 2
- Additional useful Commands 3

GAL User Interface (GUI) 6
- GAL Homepage 6
- GAL Data Upload Options 7
- GAL Sample Data 8
- GAL Genome Browser 9
- Gene Sequence Page 11
- BLAST Page 12

Command Line Options 14
- How to run GAL in command line mode? 14
- Accessing host directory 14
- Running the programs 14
- Setting up the configuration file 14
- Data Format 15
- Sample organism data upload using command line mode: 16

Introduction

GAL is a software package for analyzing and visualizing a genome or a group of genomes. GAL is implemented inside Docker. Docker technology is becoming popular throughout the bioinformatics community due to its features, ease with dependencies and more efficient usage of the underlying system and resources. Docker allows deploying an application in a sandbox (called container) to run on the host operating system locally. Docker needs to be installed on the host system (Linux in this case) to proceed with GAL.

Getting Started

GAL can be installed and initiated through Docker. Docker is available in two editions: Community Edition (CE) and Enterprise Edition (EE). Docker CE and EE are available on multiple platforms, on cloud and on-premises.

- Docker website: https://www.docker.com/
- Docker Documentation for beginners: https://docker-curriculum.com/
- Docker CE and EE are available at: https://docs.docker.com/engine/installation/#supported-platforms

System Requirements

GAL can be installed on the following operating systems:

- CentOS 7.1/7.2 & RHEL 7.0/7.1/7.2/7.3 (YUM-based systems)
- Ubuntu 16.04 LTS or higher

Quick Start

1. GAL can be downloaded and installed using following docker command:

   ```
   docker pull rjit17/gal:1.0
   ```

   In 100 Mbps, network speed the entire package download takes approximately 8 minutes.

   For upcoming versions, ‘1.0’ should be replaced with respective version.

2. To run GAL use the following command:

```
docker run -it -p 8080:80 rjit17/gal:1.0
```

This will initiate GAL at port 8080 of local server or *localhost.* User may use another port to initiate another instance

[To manipulate Docker utilities refer to Docker Documentation]

3. While the GAL instance is running inside Docker container, GAL User Interface (UI) can be accessed through a web browser at following URL:

   http://localhost:port/

   In this case, it is

   http://localhost:8080/

   It can also be:

   http://<IP address of the host computer>:8080

4. GAL can now be used to upload your data through the browser.

Additional Useful Commands

List docker images

To find the pulled docker images in the system user can use the following commands:

```
docker images
```

This will list images as follows,

REPOSITORY	TAG	IMAGE ID	CREATED	VIRTUAL SIZE
rjit17/gal	0.3	8dbdefed7c21	2 days ago	5.722 GB
rjit17/gal	0.2	862e3935ccd8	2 days ago	5.722 GB
rjit17/gal	0.1	2e94bfbe45b9	9 weeks ago	5.665 GB

Set instance name

Docker by default allocates a random name and id for the running instance. User can change the instance name by adding '–-name' option in the command line. It will help the user to track an instance later.

Example:

```
docker run --name=test -it -p 8080:80 rjit17/gal:1.0
```

Here '**test'** is the name of the running instance.

Find docker instances

To find all the available docker instances use the following commands

```
docker ps –a
```

This is the output example of the above command.

```
CONTAINER ID  IMAGE           COMMAND                  CREATED
969ab10373bc  rjit17/gal:0.2  "/bin/sh -c 'service "  26 hours ago
476d22340d5f  rjit17/gal:0.3  "/bin/sh -c 'service "  47 hours ago
a5e4e47e6bdb  rjit17/gal:0.2  "/bin/sh -c 'service "  2 days ago

STATUS           PORTS                    NAMES
Up 26 hours      0.0.0.0:8080->80/tcp     hopeful_visvesvaraya
Up 47 hours      0.0.0.0:7070->80/tcp     mad_pare
Exited 2 days ago                         trusting_curie
```

Exit docker instance

To exit from a running docker instance use `exit` command.

To exit from docker command line, use **CTRL+p** followed by **CTRL+q**

Re-enter running instance

To re-enter into a running instance, use the following command

```
docker exec –it <Container_id/Name> bash
```

Example:

```
docker exec –it test bash
```

Here '**test'** is the name of the running instance.

Restart Docker instance

To start the stooped instances, use the following command:

```
docker start -i <Container_id/Name>
```

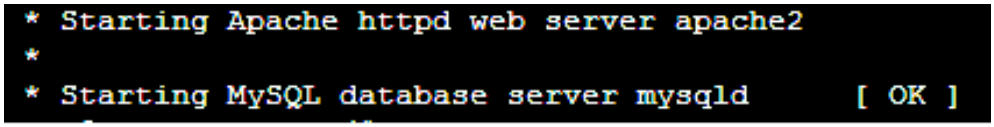
Example:

```
docker start -i test
```

Here '**test'** is the name of the running instance.

Successful GAL Start

On successful docker GAL instance start, the following message will appear.

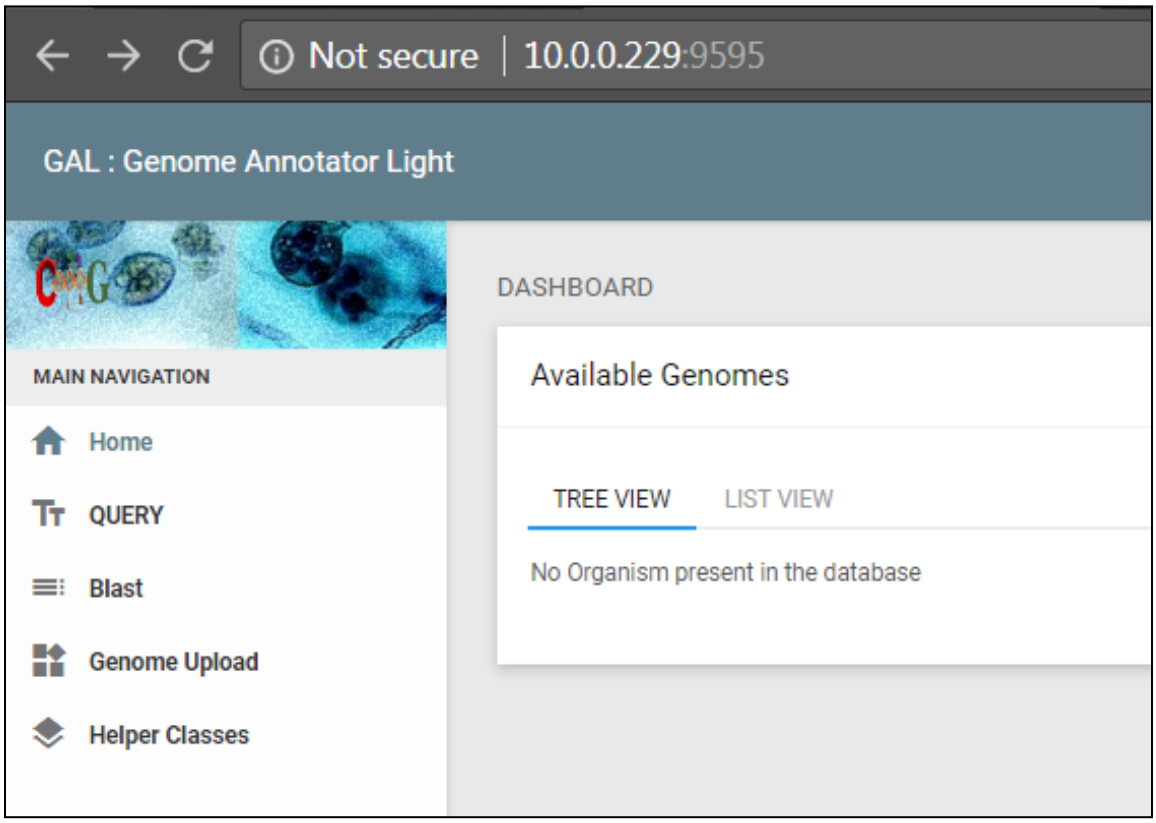


[OK] indicates successful initiation.

GAL User Interface (GUI)

GAL GUI is must for data visualization, and it includes several web pages like,

GAL Homepage



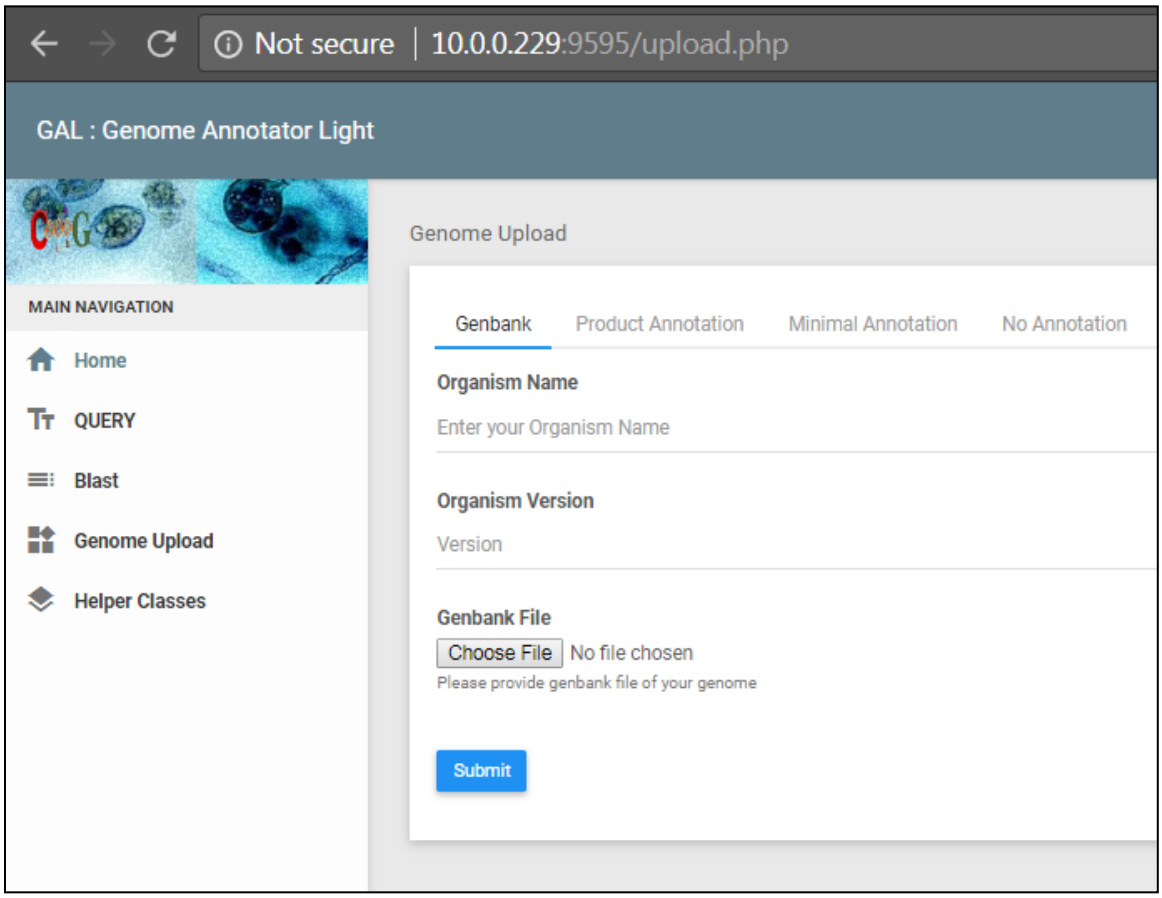
- GUI for GAL can be loaded inside a web browser for Genome Upload, Genome Browsing; downstream analyses like Blast Searches, Annotation Query and Sequence Retrieval along with analyses of all the annotated proteins using various EMBOSS tools.

- The Homepage will list the genomes only after they are processed. Until then there will be no data available in the list view or tree view.

  It approximately took 28 minutes to process ~5 Mb *E.coli* genome for Genbank Annotation as input on standard Ubuntu Desktop having 4 CPUs and 4 Gb of RAM. The same genome at various annotation levels took proportionate time. E.g. Product Annotation (31 minutes), Minimal Annotation (30 minutes), and No Annotation (175 minutes using GeneMark annotator + NCBI BLAST).

- The Navigation panel to the left will help the user to access various features like:
    - **Genome Upload:** Upload options at any stage of the annotation process.
    - **QUERY:** Gene search using gene name, primary annotation, genomic locus or HMMPFAM/Signalp/tmhmm annotations.
    - **BLAST:** Sequence search using NCBI BLAST for protein or gene sequence within the uploaded dataset.
    - **Help:** Help and documentation.

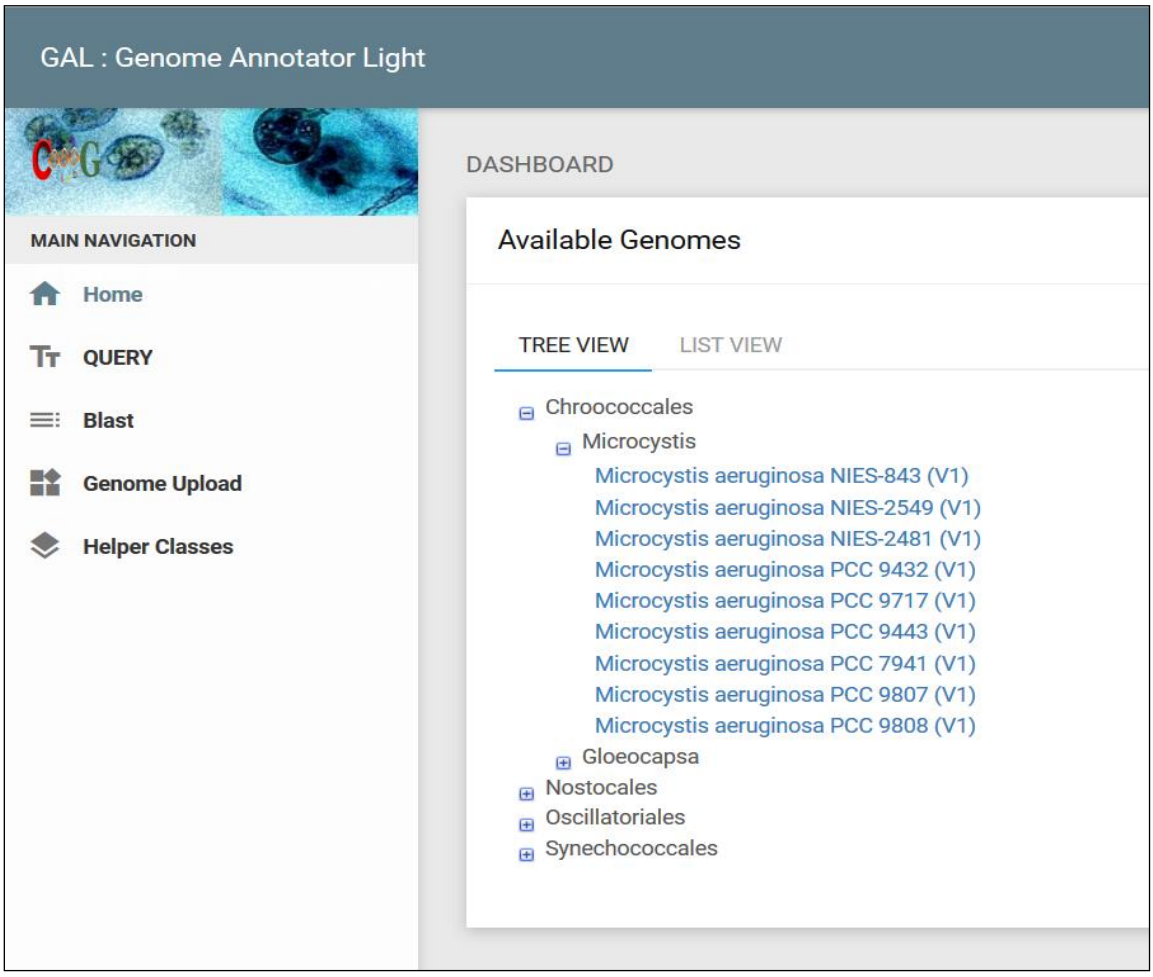
GAL Data Upload Options



The user can provide data in four ways, viz. type1: Genbank Annotation, type2: Only Genome Fasta files, type3: Genome fasta and gff files; type 4: Genome Fasta, gff files and product files

- **Genbank Annotation:** This allows data input through NCBI annotated Genbank file (GBFF).
- **Product Annotation:** This allows genome FASTA, GFF (genome feature file) and product information file.
- **Minimal Annotation:** This allows the basic annotation information provided by the user where user provides genome FASTA (FNA) file and GFF file.
- **No Annotation:** This allows data through only genome FASTA (FNA) file with annotation options using AUGUSTUS or Genmark for eukaryotic and prokaryotic genomes using related reference genomes, respectively.

GAL Sample Data



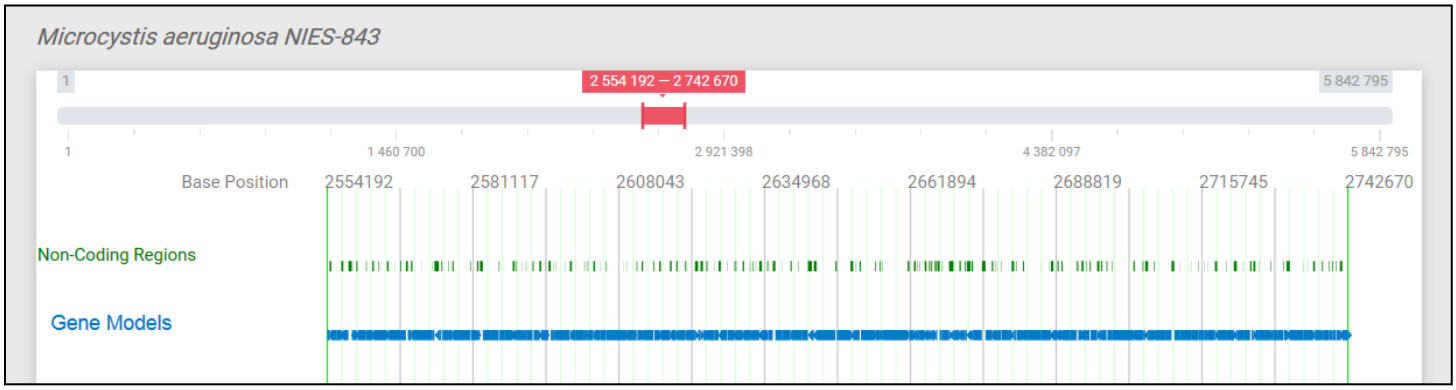
Clicking the genome name will direct the browser to Genome Summary Page for respective organism where organism details and links to the Scaffold wise Genome browser links are provided.

From genome browser, each coding and non-coding regions can be visualized in details with exon-intron boundaries along with sequence download links and analysis options.

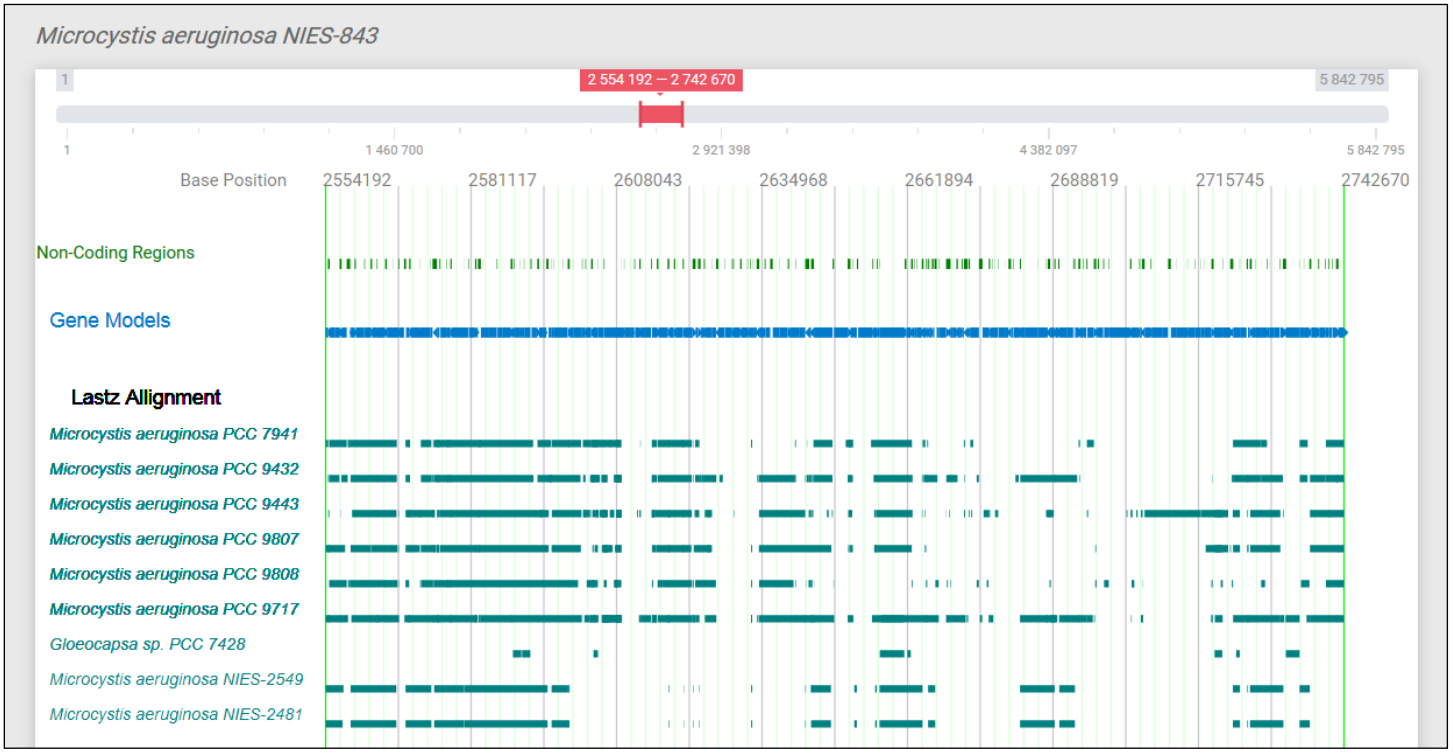
GAL Genome Browser

GAL Genome browser can visualize coding and non-coding regions in selected locus range of selected genome, as shown in the following image.

SINGLE GENOME BROWSER MODE



MULTI GENOME BROWSER MODE



Additionally, GAL can automatically visualize respective regions from multiple taxonomically related species (if present in given dataset) based on LastZ Alignments.

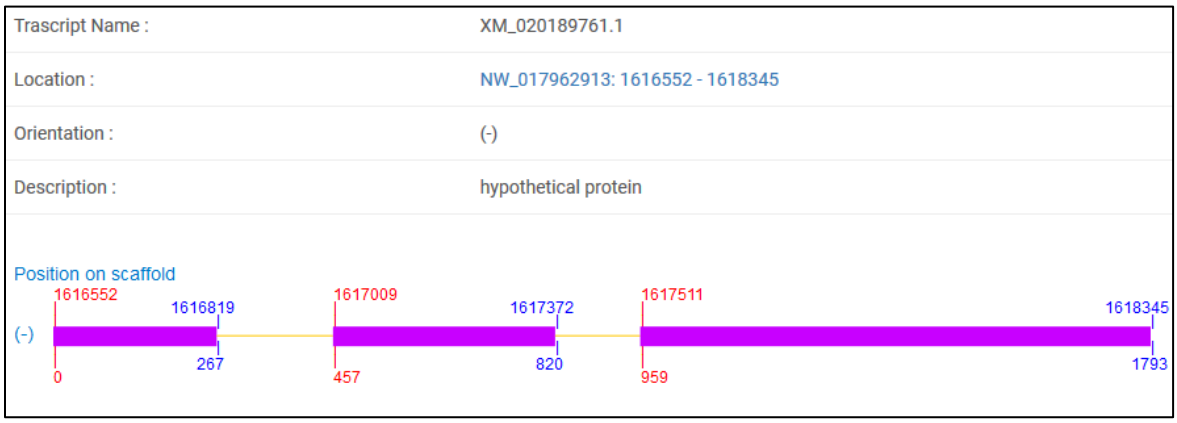
Each highlighted region links to the individual gene details page with annotation details, gene analysis options and sequence download options.

.

Gene Details Page

All the annotated genes, transcript or proteins can be analyzed separately into Gene details page,

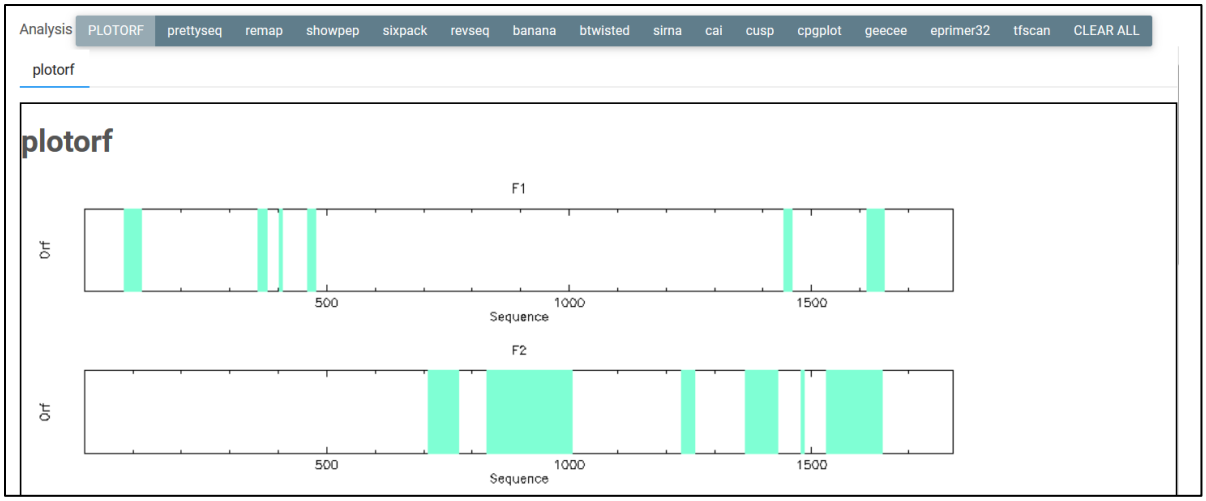
Exon Intron Boundaries for transcripts:



Annotation summary tables for various methods are also displayed on the same page for more details.

EMBOSS TOOLKIT

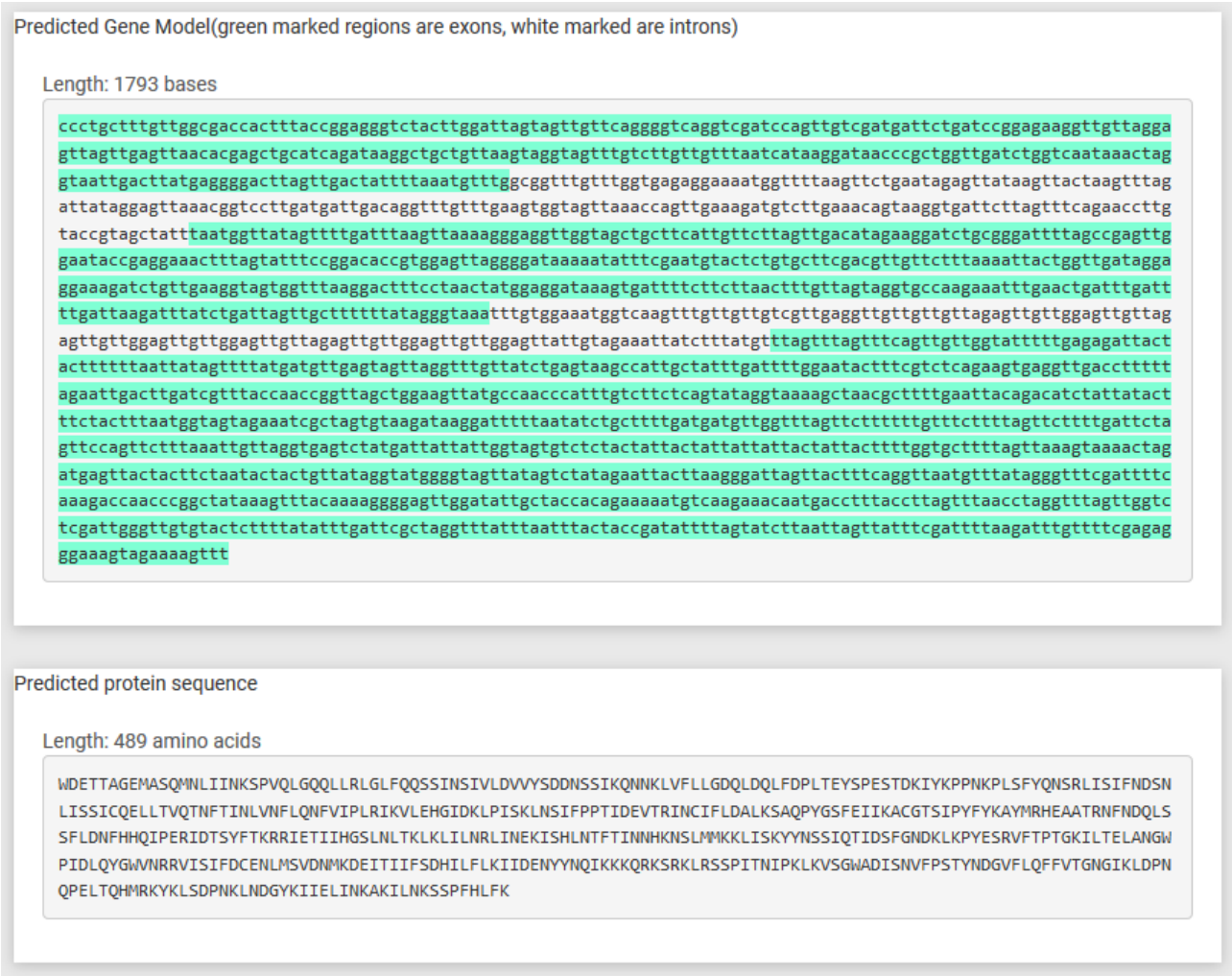
The protein analysis supported by various EMBOSS tools is available at each gene details page. The outputs can be visualized on the same page by just clicking the name of the package. All the outputs can be downloaded as image or text format wherever suitable.



The above screenshot shows various EMBOSS tools incorporated into the GAL analysis. The example output for the given transcript by plotorf tool is shown here. All the adjacent tabs with the name of these tools can generate the standard outputs. These tools include **banana, cpgplot, eprimer32, sixpack, showpep, tfscan** etc.

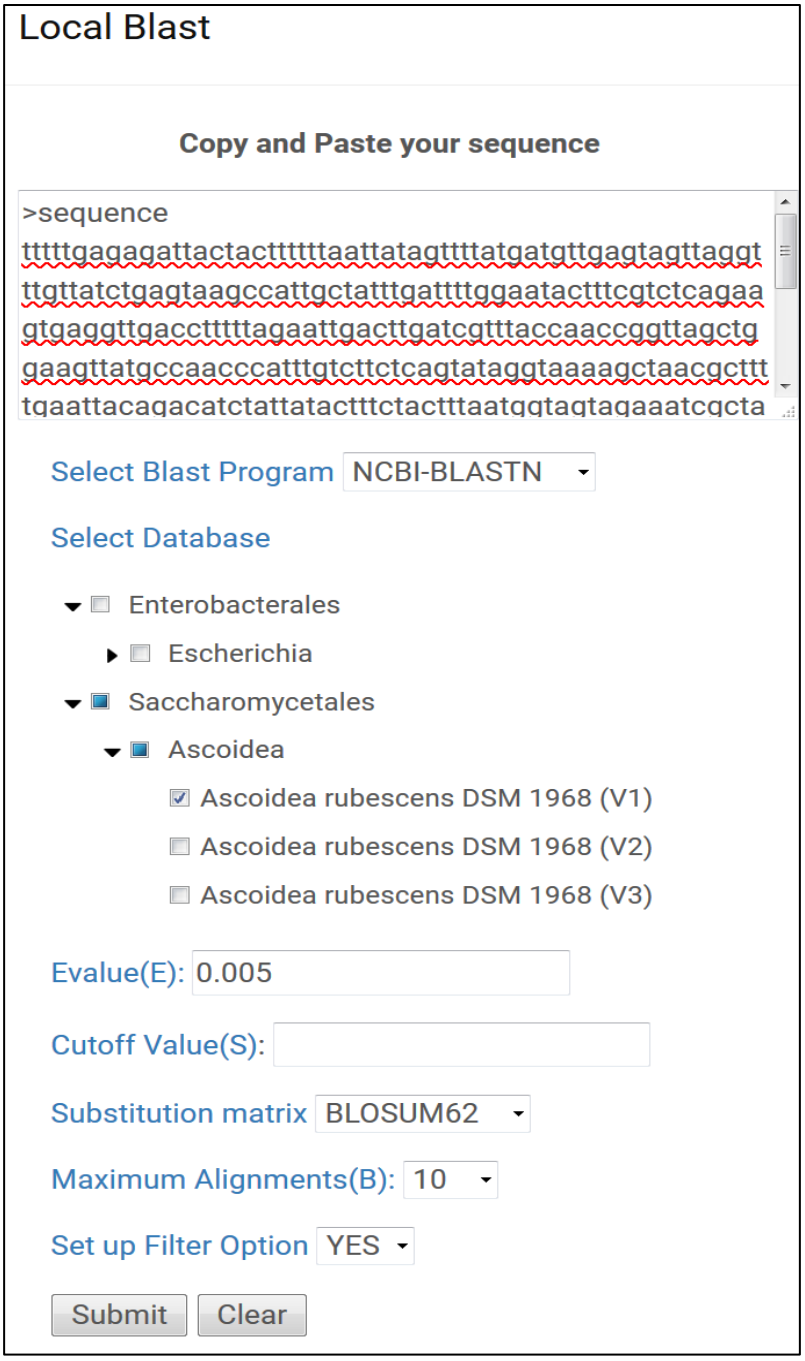
Gene Sequence Page

The gene sequence page provides the option for retrieving nucleotide sequences of the genomic region as well as protein sequence of the translated gene. The green highlighted sequence indicates the exons for easy understanding and reporting.



BLAST Page

As the genomes are available in the database after processing the genomes uploaded by the user, any nucleotide or protein sequences can be BLASTed against the available genomes. The selection of any of the genomes or all the is possible from the checkboxes near organism names. The genomes are shown as tree view for the blast options.



The screenshot of the BLAST page showing various option for sequence input and parameter as well as genome selection.

Command Line Options

How to run GAL in command line mode?

GAL can easily be run from a web browser. Optionally, for users familiar with Docker command line and Ubuntu Terminal can run GAL through command line.

Accessing host directory

The host directory can be accessed through the following command:

```
docker run -it –v [host_directory_path]:[GAL_file
 system_path] -p 8080:80 rjit17/gal:[GAL version]
```

Example:

```
docker run -it -v /home/arijit/test:/usr/GAL_data
 -p 8080:80 rjit17/gal:1.0
```

After running the above command, the host operating directory will be available to the GAL file system. In that way user can process data from the host directory. Now you will enter to GAL container.

```
root@container_id:/#
```

Running the programs

GAL is based on Python. Python 3.4 or above is required to use GAL. The main program for GAL is **`main.py`** present at: **`/usr/GAL`** path.

To run the GAL control script use following command:

```
python3 /usr/GAL/main.py --orgconfig=[config_file_path]
```

Setting up the configuration file

User needs to provide configuration file in INI format.

INI format:

```
[section]
name=value
```

Structure of the organism configuration file:

```
[OrganismDetails]
Organism:
version:
source_url:

[SequenceType]
SequenceType:

[AnnotationInfo]
Blastp:
signalp:
pfam:
tmhmm:

[filePath]
GenBank:
FASTA:
GFF:
Product:
LastZ:
SignalP:
pfam:
TMHMM:
Interproscan:

[other]
Program:
ReferenceGenome:
```

Sample configuration file is present at: **/usr/GAL/config/organism_config_format.ini**

Data Format

We have defined input data type in four ways,

Data type	Name	Input files
Type1	Genbank Annotation	Genbank Sequence File
Type2	No Annotation	Genome Fasta File
Type3	Minimal Annotation	Genome Fasta File, GFF file
Type4	Product Annotation	Genome Fasta File, GFF File, Product file

Sample organism data upload using command line mode:

Data	Commands to upload Sample genomes
Type1	`python3 /usr/GAL/main.py --orgconfig=/usr/GAL/SampleFiles/type1.Ini`
Type2	`python3 /usr/GAL/main.py --orgconfig=/usr/GAL/SampleFiles/type2.Ini`
Type3	`python3 /usr/GAL/main.py --orgconfig=/usr/GAL/SampleFiles/type3.Ini`
Type4	`python3 /usr/GAL/main.py --orgconfig=/usr/GAL/SampleFiles/type4.Ini`

END OF DOCUMENT