MGEScan Documentation

Release 0.1

Hyungro Lee

May 15, 2018

Contents

1 QuickStart 3

2 MGEScan Workflow 15

3 MGEScan Command Line Interface 21

4 MGEScan on Galaxy Installation 25

5 MGEScan ToolShed 31

6 MGEScan Software Process 37

7 MGEScan on Amazon Cloud (EC2) 41

8 MGEScan-LTR 49

9 MGEScan-nonLTR 53

10 Visualization 55

11 Test Results (New) 57

12 Test Results 61

13 Test Results with Previous MGEScan 1.3.1 65

14 Source code 67

MGEScan on Galaxy is the latest version of MGEScan to identify long terminal repeats (LTR) and non-LTR retroelements in eukaryotic genomic sequences on a web interface or on a command line. HMMER v3.1b1 and openMPI are supported for MGEScan-LTR and MGEScan-nonLTR programs so the better performance is guaranteed than previous version of MGEScan. Cloud image is available on Amazon Cloud (EC2) to utilize on-demand computing resources for data analysis.

The documentation provides basic tutorials of using MGEScan on Galaxy Workflow system and additional information such as installation and use of MGEScan on Amazon Cloud (AWS EC2).

CHAPTER 1

QuickStart

MGEScan, identifying LTR and non-LTR in genome sequences are available on the Galaxy scientific workflow which is a web-based workflow software to support data analysis with various tools.

1.1 Overview

This tutorial demonstrates a quick start of using MGEScan on Galaxy workflow with a sample dataset, D. melanogaster genome.

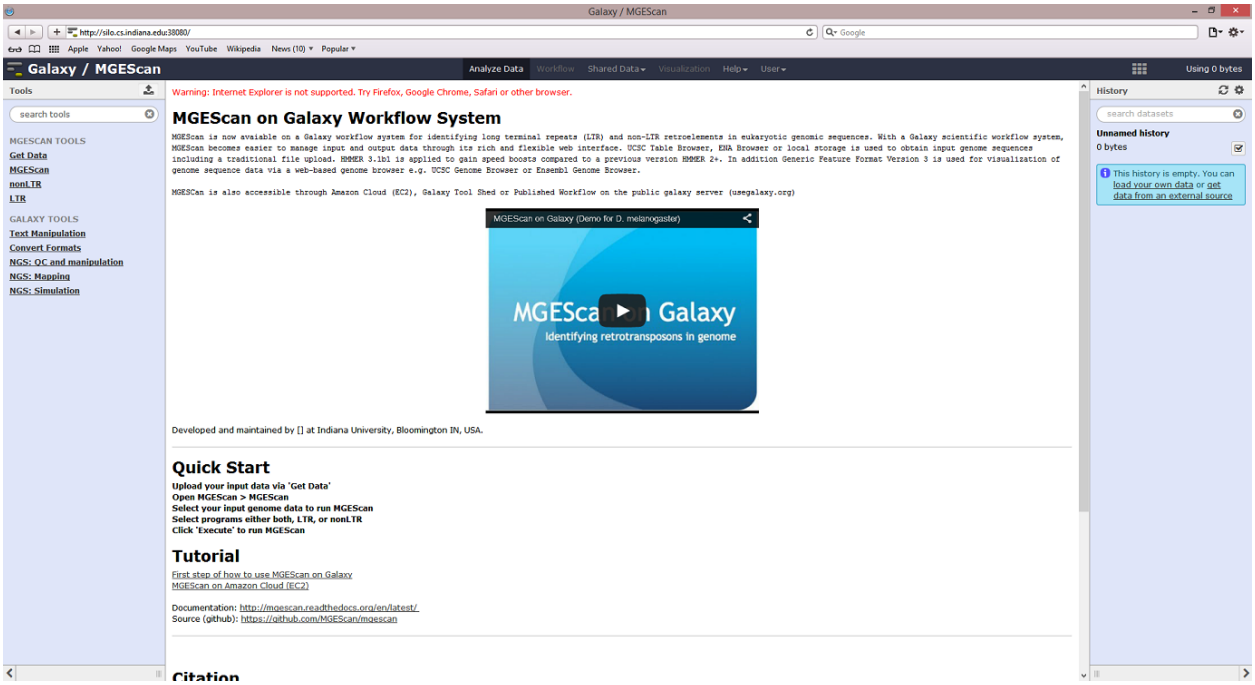
Tip: Approximate 3 hours and 30 minutes (including 3 hours computation time)

1.2 Run MGEScan-LTR and MGEScan-nonLTR for D. melanogaster

In this tutorial, we will try to run both MGEScan-LTR and MGEScan-nonLTR with D. melanogaster genome dataset. You can find the dataset at the `Shared Data` menu on top and MGEScan tools on the left frame.

1.2.1 Access to Galaxy/MGEScan

Run Galaxy/MGEScan at your machine:

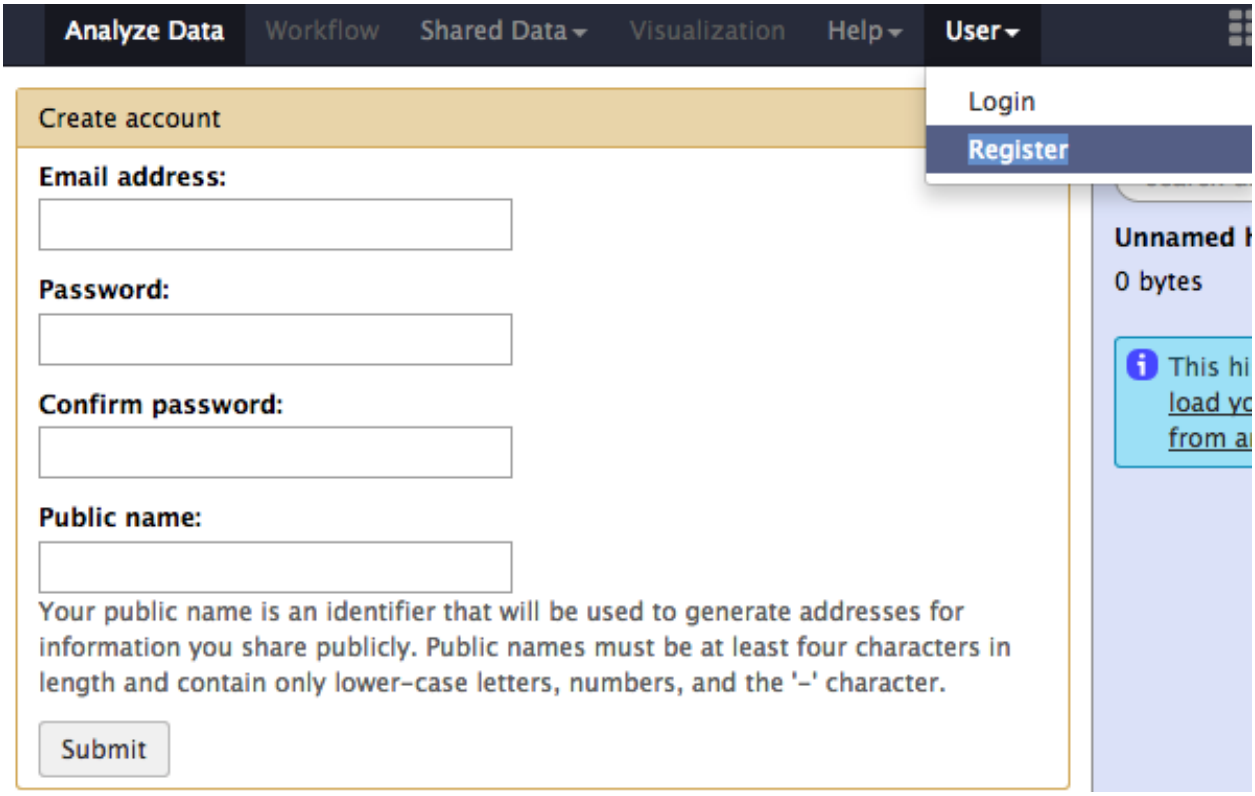


1.2.2 Login or Register (Optional)

You can save your work if you have account on Galaxy workflow. The user-based history in Galaxy/MGEScan stores your data and launched tasks. The guest user account is able to run the MGEScan tools without the login but results or history data won't be saved if the web browser session is closed.

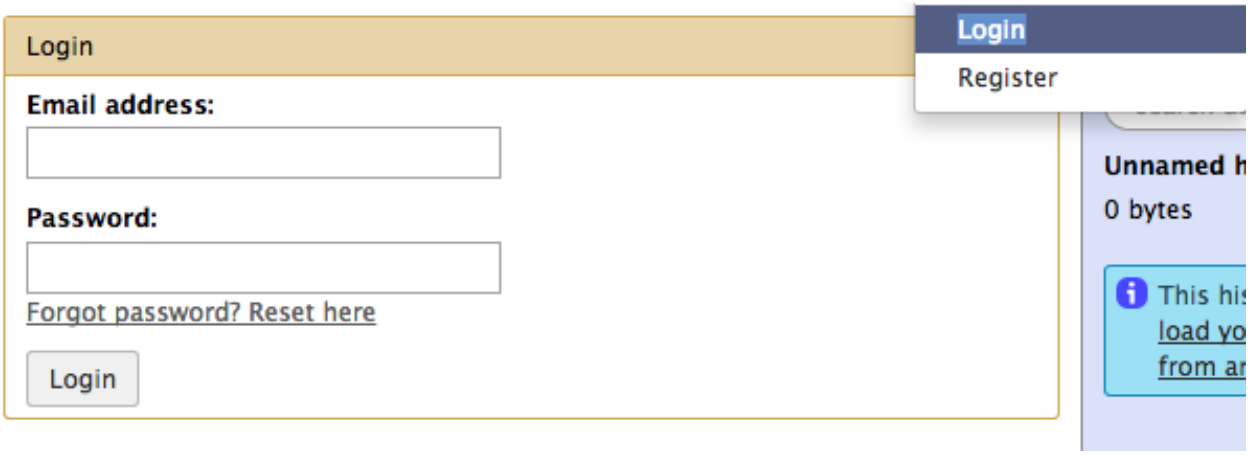
Register

Email address is required to sign up.



Login

If you already have an account, you can use your user id and password at the *User* > *Login* page.

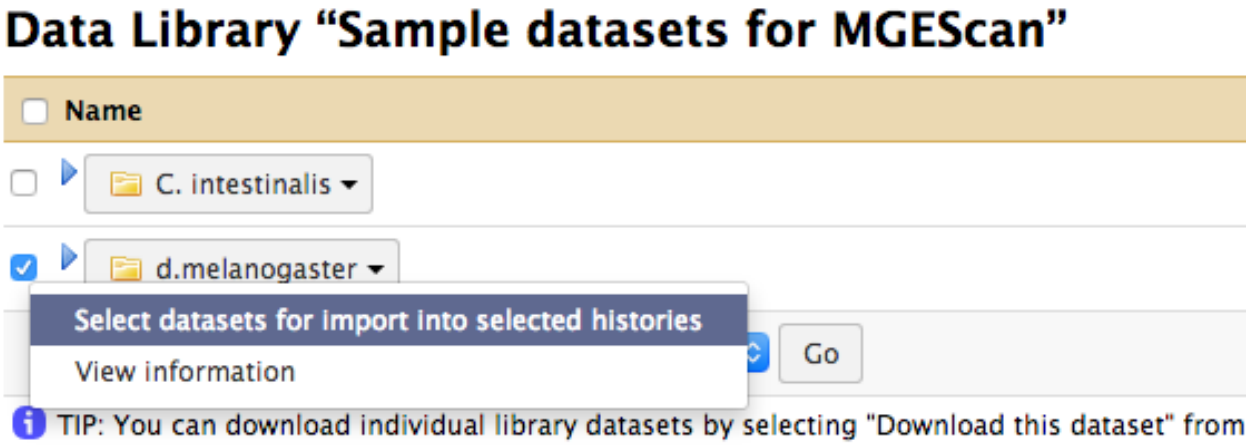


1.2.3 Get Dataset from `Shared Data`

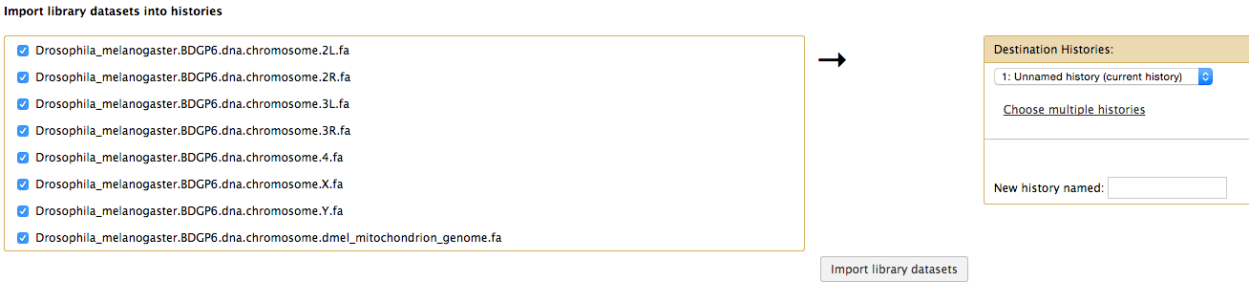
You can find sample datasets (e.g. D.melanogaster) at `Shared Data` menu on top. Click "Shared Data" > "Data Libraries" and find "Sample datasets for MGEScan".

1.2.4 Example: Drosophila melanogaster

In the Data Library, enable the checkbox for `d.melanogaster` and click "Select datasets for import into selected histories" from the down arrow at the end.



You will find 8 fasta files are available. We need to import all of them, make them all checked and click "Import library datasets" in the middle of the page.



Once you imported the D. melanogaster datasets into your history, you are ready to run MGEScan tools on Galaxy. Go to the main page, and checkout imported datasets (8 files) on the right frame of the page.

Note: You can select where datasets to be imported.

1.2.5 Run MGEScan for LTR and nonLTR

In the new version of MGEScan, two programs, MGEScan-LTR and MGEScan-nonLTR, can be ran at the same time with a merged result. Open the page at **"MGEScan > MGEScan"**, a simple tool is available for LTR and nonLTR executions with MPI option for parallel processing.

Note: Find **LTR** or **nonLTR** page if you'd like to choose other options to run MGEScan tools in detail.

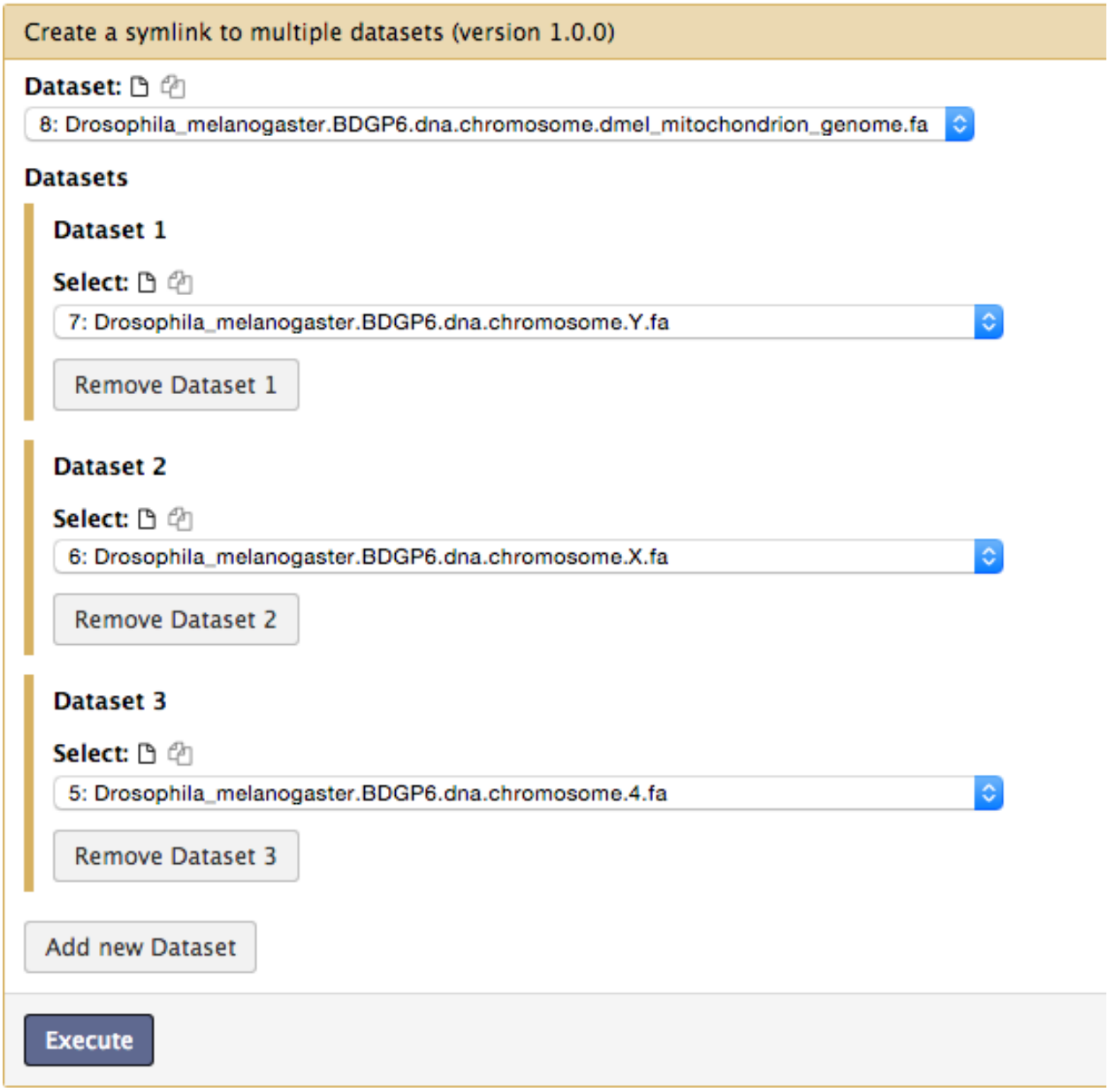
1.2.6 Create a single link to multiple inputs

In the example of `d. melanogaster`, we have 8 fasta files as its sequences. To analyze them all at the same time, we create a single link to the files prior to running MGEScan tool on Galaxy. One archive file to many files (e.g.

file.tar) will be used as an input of MGEScan tool on Galaxy. Note that Galaxy workflow does not support multiple arbitrary inputs but this symlink tool allows you to have dynamic inputs as a Galaxy input dataset.

- FInd "Tools > Create a symlink to multiple datasets" on the left frame.

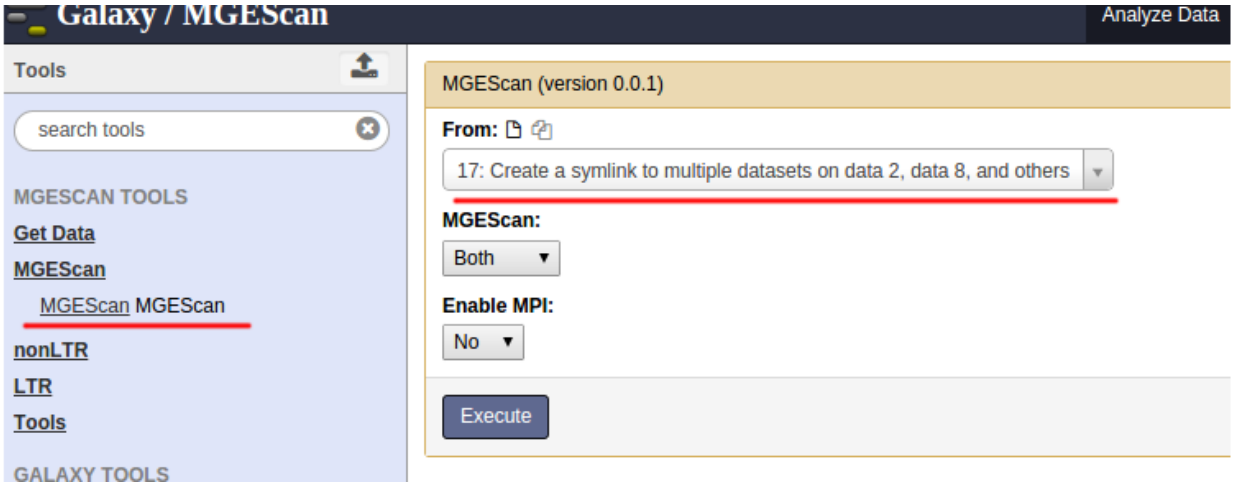
We will add 8 fasta files each by clicking "Add new Dataset" from "8: Drosophila_melanogaster.BDGP6.dna.chromosome.dmel_mitochondrion_genome.fa" to "1: Drosophila_melanogaster.BDGP6.dna.chromosome.2L.fa" like so:



Make sure you have added all the files without duplication. The added order is not important though. File(s) will be placed in a same directory without order.

1.2.7 MGEScan Tool

MGEScan runs both LTR and nonLTR with a selected input genome sequence. Find "MGEScan > MGEScan" tool on the left frame and confirm that the symlink dataset we created in the previous step is loaded in "From" select form like so:



Enable MPI

To accelerate processing time, select "Yes" at "Enable MPI" select form and specify "Number of MPI Processes". If you have a multi-core system, use up to the number of cores.

Our options are:

- From: Create a symlink to multiple datasets on data 2 and data 8, and others
- MGEScan: Both
- Enable MPI: Yes
- Number of MPI Processes: 4

And click "Execute".

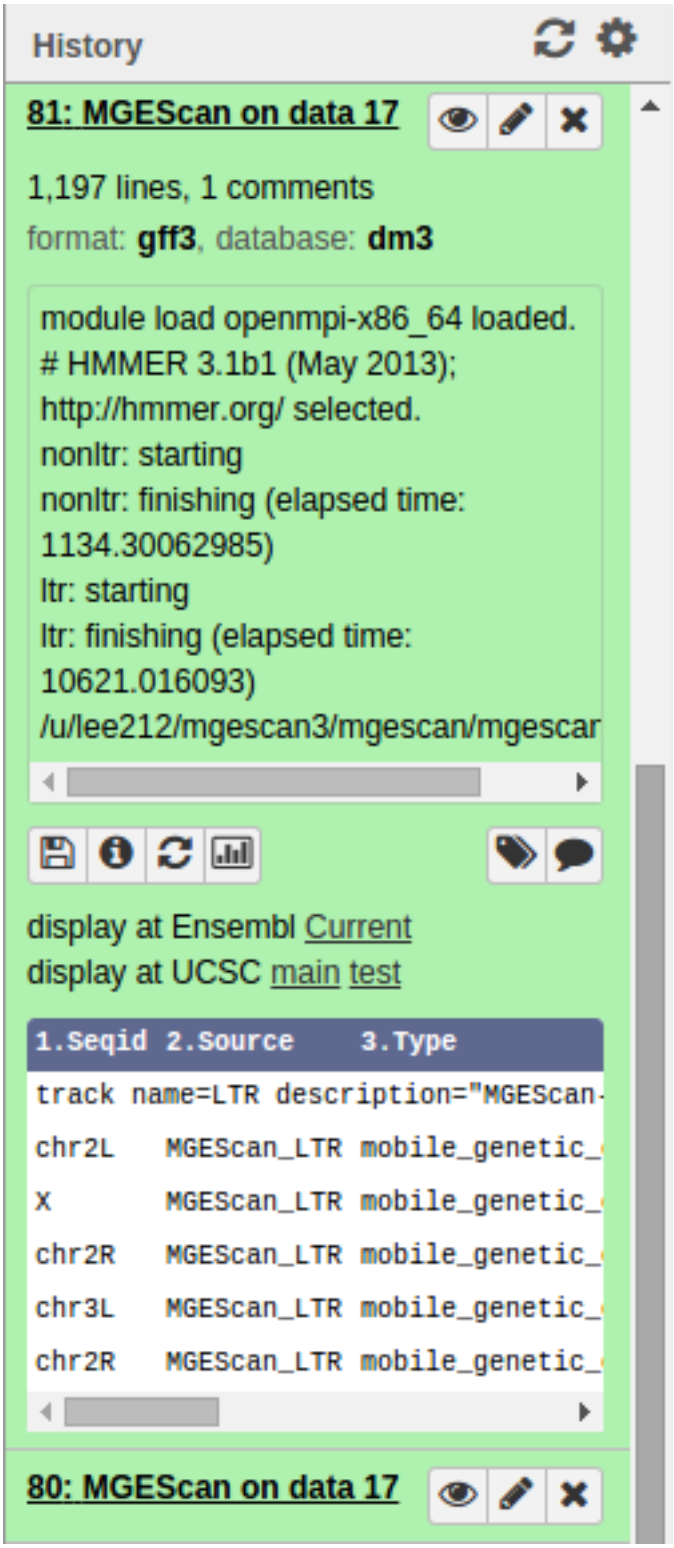
1.3 Computation Time

Our test case took 3 hours for analyzing LTR and nonLTR of `D. melanogaster`:

- nonLTR: 19 minutes
- LTR: 3 hours
- Total: 3 hours

1.4 Results

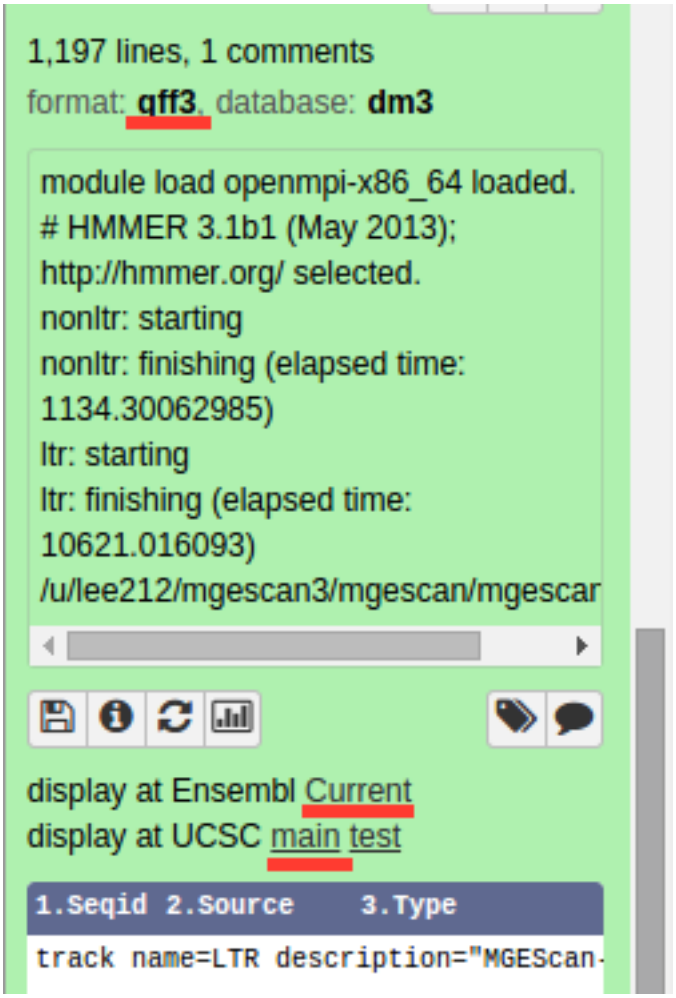
Upon the MGEScan tools completion, the output files are accessible via Galaxy in *gff3* format, a plain text, or an archived (e.g. tar.gz) file. You will notice that the color of your tools has been changed to green like so:



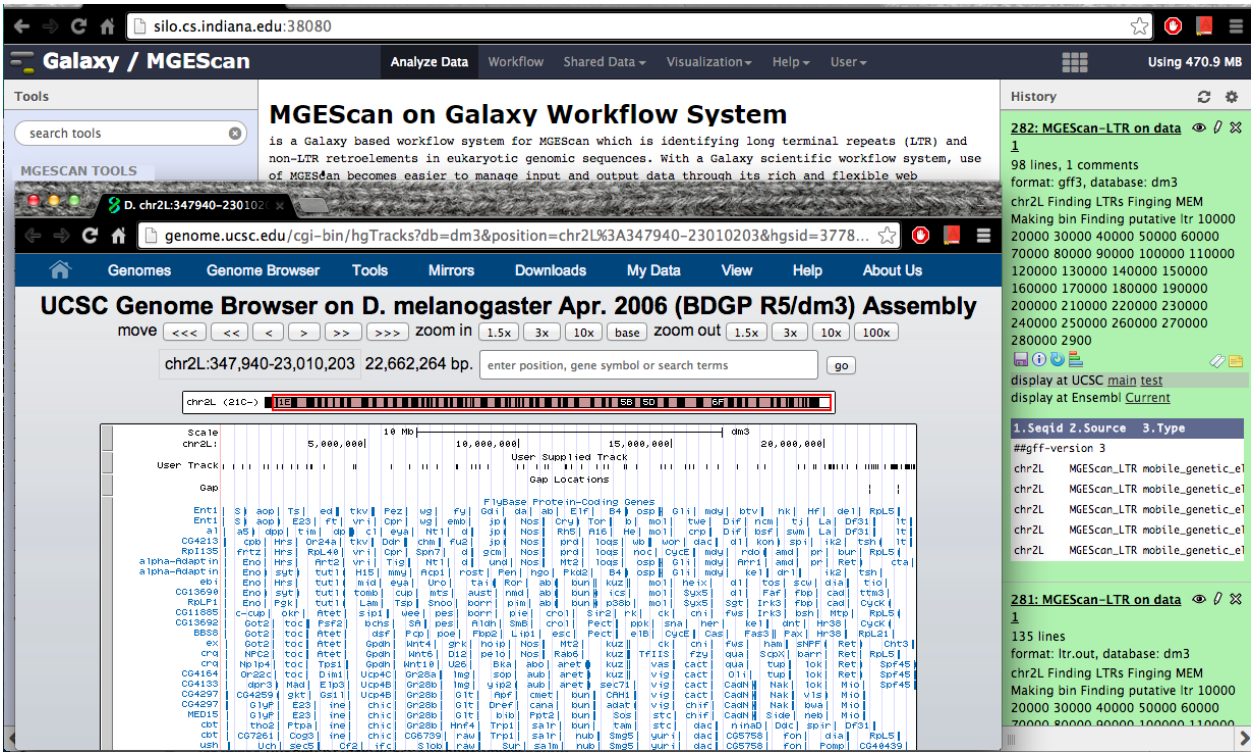
You can download the output files to your local storage, or get access to Genome Browser with provided links.

1.4.1 Visualization: UCSC or Ensembl Genome Browser

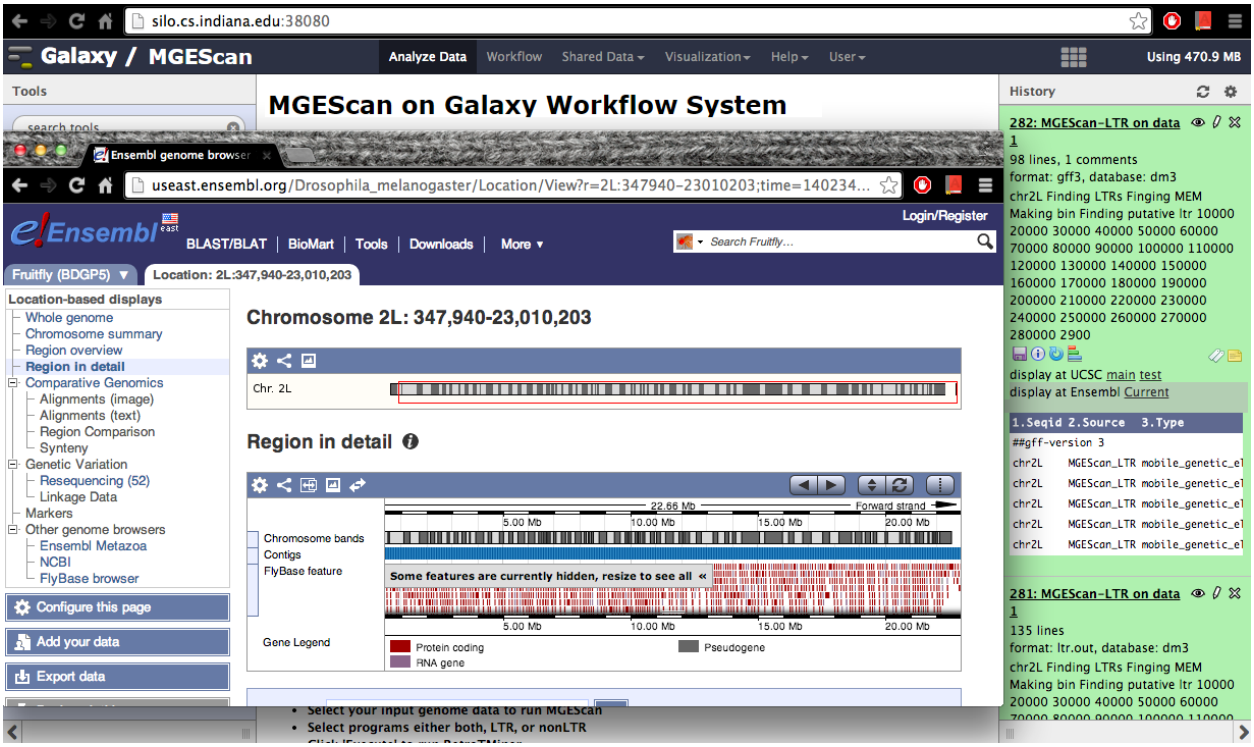
Your genomic data in a Generic Feature Format Version 3 (gff3) can be displayed by a well known visualization tool such as UCSC or Ensembl Genome Browser on Galaxy with custom annotations of MGEScan for LTR and nonLTR. Find the link provided for gff3 to view interactive graphical display of genome sequence data.



UCSC Genome Browser (Example View)



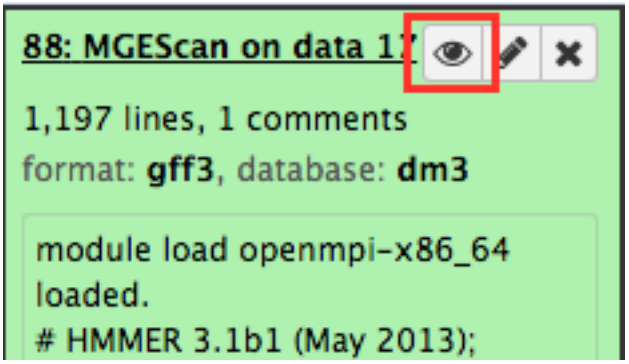
Ensembl (Example View)



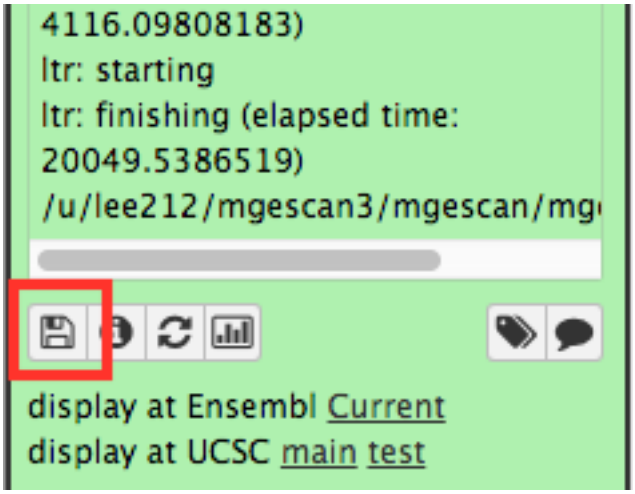
1.4.2 Additional Options

There are other options to view results on a web interface or local.

- View data: Content of the result file

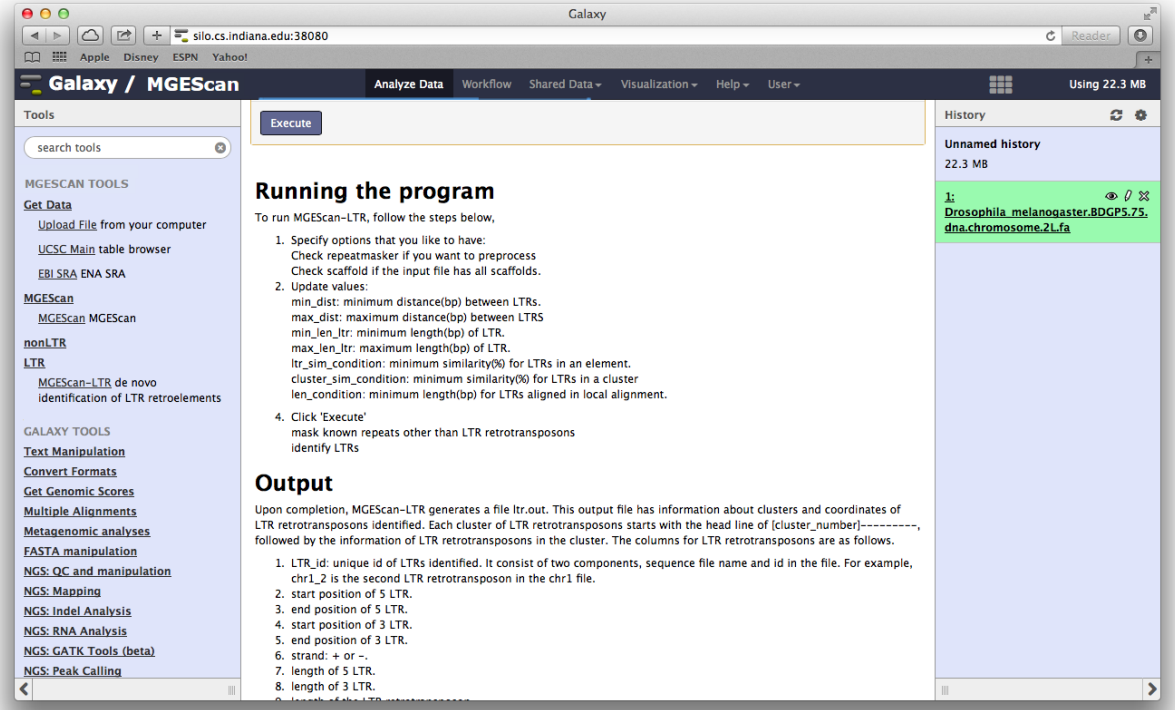


- Download: Download the file



Description of tools

Each tool in Galaxy has its description to explain how to use.


Galaxy
silo.cs.indiana.edu:38080
Reader
Apple
Disney
ESPN
Yahoo!
Galaxy / MGEScan
Analyze Data
Workflow
Shared Data
Visualization
Help
User
Using 22.3 MB
Tools
search tools
MGESCAN TOOLS
Get Data
Upload File from your computer
UCSC Main table browser
EBI SRA ENA SRA
MGEScan
MGEScan MGEScan
nonLTR
LTR
MGEScan-LTR de novo identification of LTR retroelements
GALAXY TOOLS
Text Manipulation
Convert Formats
Get Genomic Scores
Multiple Alignments
Metagenomic analyses
FASTA manipulation
NGS: QC and manipulation
NGS: Mapping
NGS: Indel Analysis
NGS: RNA Analysis
NGS: GATK Tools (beta)
NGS: Peak Calling
Execute
Running the program
To run MGEScan-LTR, follow the steps below,
1. Specify options that you like to have:
Check repeatmasker if you want to preprocess
Check scaffold if the input file has all scaffolds.
2. Update values:
min_dist: minimum distance(bp) between LTRs.
max_dist: maximum distance(bp) between LTRS
min_len_ltr: minimum length(bp) of LTR.
max_len_ltr: maximum length(bp) of LTR.
ltr_sim_condition: minimum similarity(%) for LTRs in an element.
cluster_sim_condition: minimum similarity(%) for LTRs in a cluster
len_condition: minimum length(bp) for LTRs aligned in local alignment.
4. Click 'Execute'
mask known repeats other than LTR retrotransposons
identify LTRs
Output
Upon completion, MGEScan-LTR generates a file ltr.out. This output file has information about clusters and coordinates of LTR retrotransposons identified. Each cluster of LTR retrotransposons starts with the head line of [cluster_number]---------, followed by the information of LTR retrotransposons in the cluster. The columns for LTR retrotransposons are as follows.
1. LTR_id: unique id of LTRs identified. It consist of two components, sequence file name and id in the file. For example, chr1_2 is the second LTR retrotransposon in the chr1 file.
2. start position of 5 LTR.
3. end position of 5 LTR.
4. start position of 3 LTR.
5. end position of 3 LTR.
6. strand: + or -.
7. length of 5 LTR.
8. length of 3 LTR.
History
Unnamed history
22.3 MB
1: Drosophila_melanogaster.BDGP5.75.dna.chromosome.2L.fa

CHAPTER 2

MGEScan Workflow

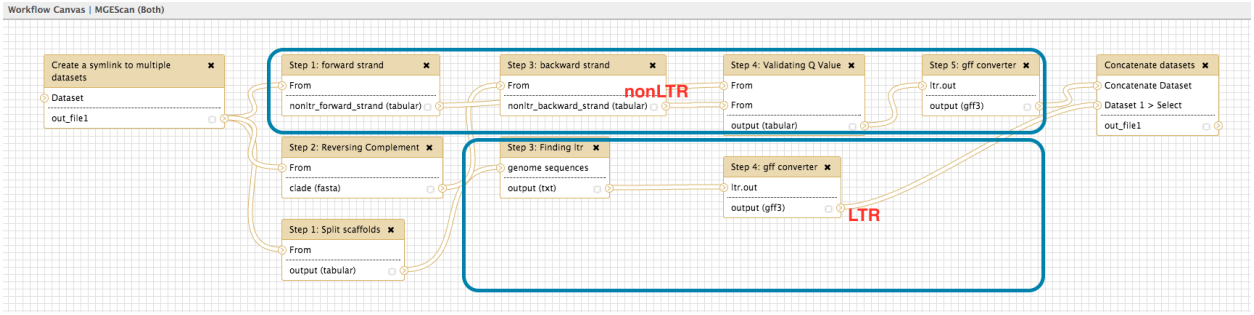
MGEScan tools for LTR and nonLTR consist of a series of computational steps in Galaxy Workflow. With the drawing canvas, you can compose sub-processes of MGEScan with other Galaxy tools and run entire workflow applications (steps) or just find out the details of processes of MGEScan tools. Each application normally has both input and output connected to the input of the next.

We provide three workflows:

- MGEScan (Both) for identifying LTR and nonLTR
- MGEScan-LTR
- MGEScan-nonLTR

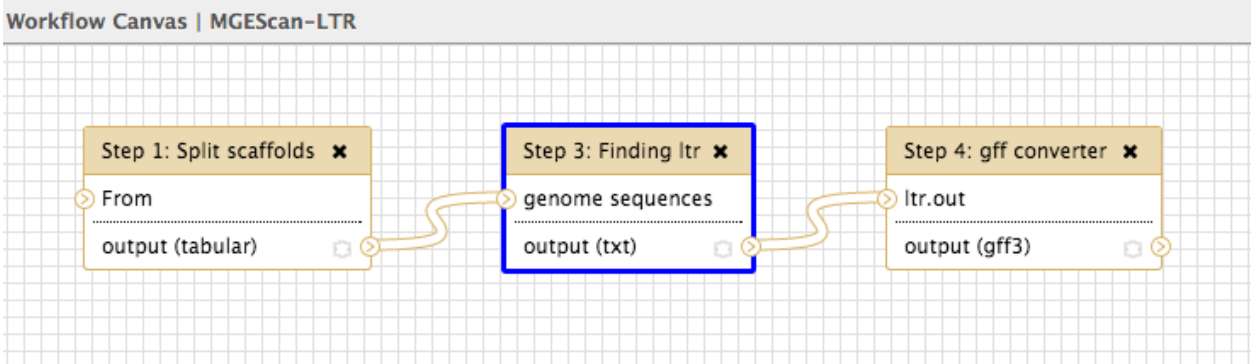
2.1 MGEScan (Both)

This workflow contains 10 steps to run both LTR and nonLTR programs in parallel. Find "MGEScan (Both)" at Workflow menu on top.



2.2 MGEScan-LTR

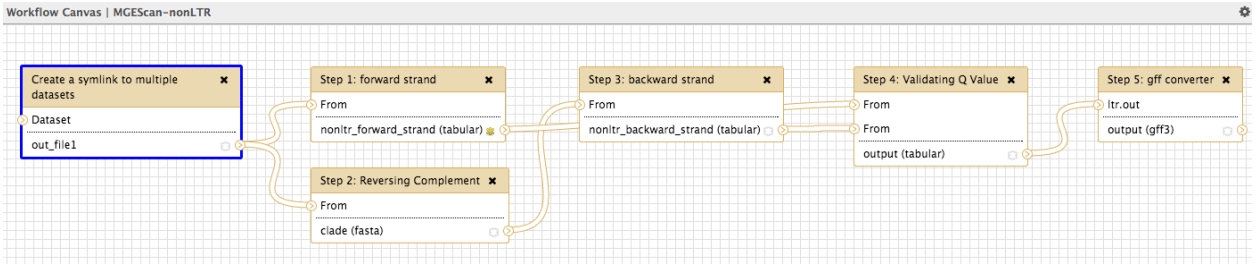
This workflow contains 3 steps to run the LTR program.



- Step 1: Split scaffolds
- Step 2: RepeatMasker (optional)
- Step 3: Finding ltr
- Step 4: gff converter

2.3 MGEScan-nonLTR

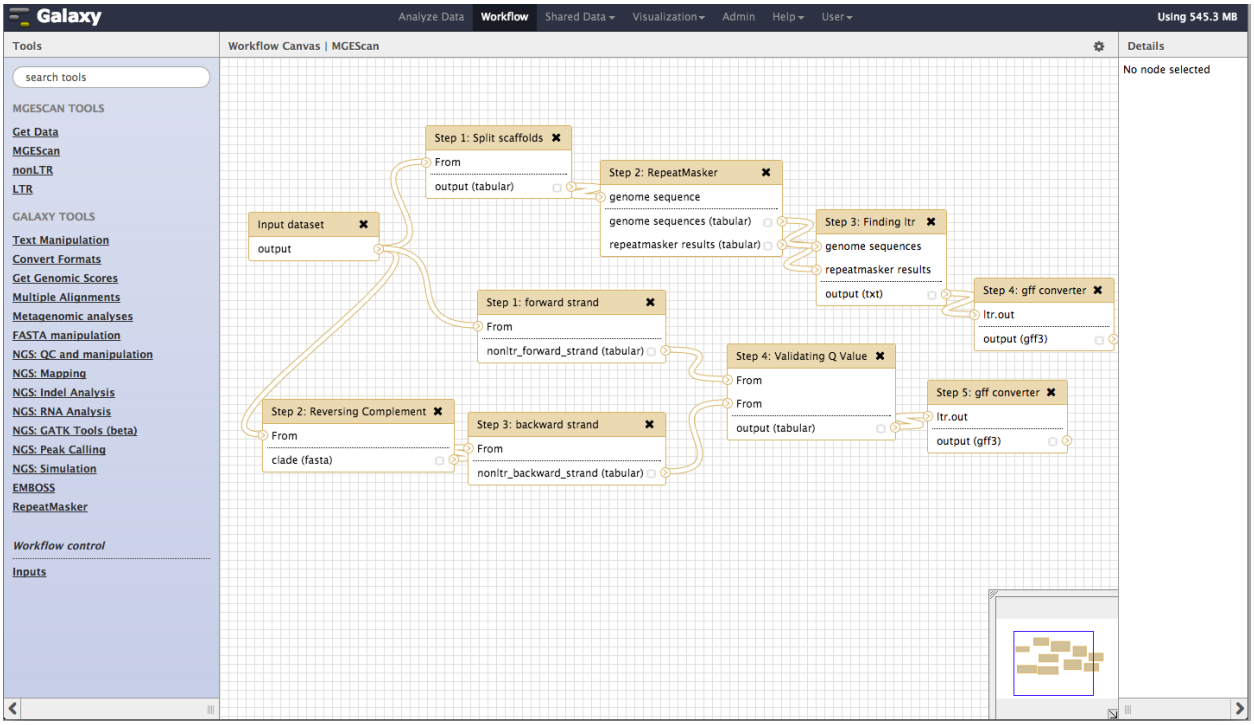
This workflow contains 6 steps to run the nonLTR program.



- Create a symlink to multiple datasets
- Step 1: forward strand
- Step 2: Reversing Complement
- Step 3: backward strand
- Step 4: Validating Q Value
- Step 5: gff converter

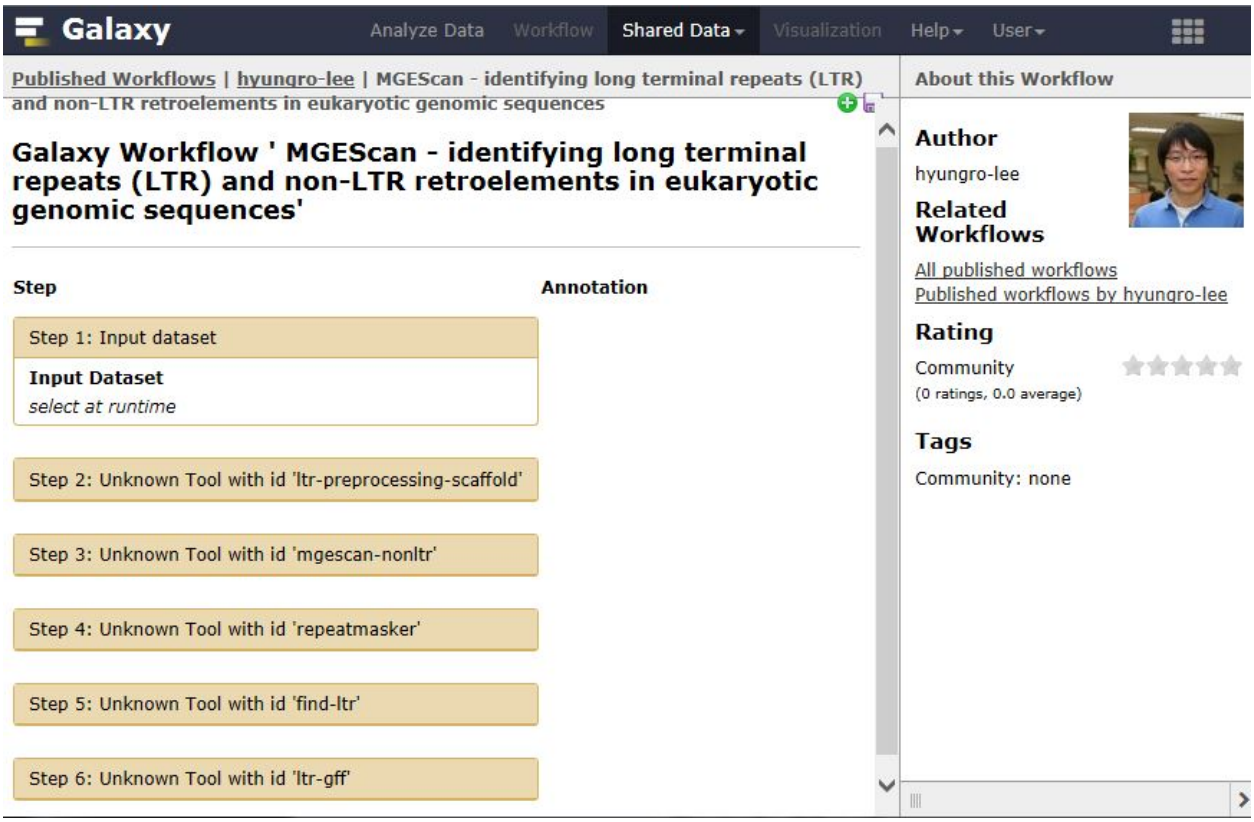
2.4 Workflow Canvas

In Galaxy > Workflow > Edit, you can modify or update the MGEScan workflow on Galaxy Workflow Canvas.



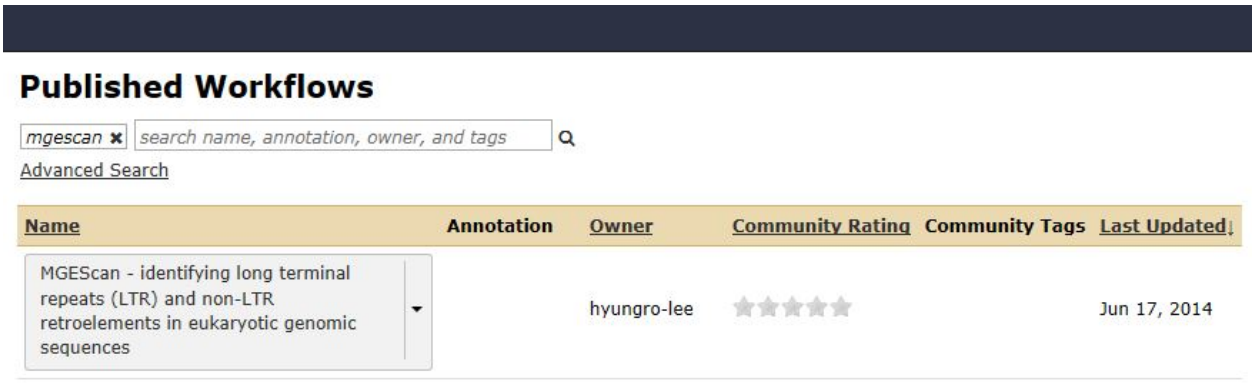
2.5 Registered Workflow in Local

Once you completed composing/updating workflow, you can save your work on local. You can download and store workflow file on your storage.



2.6 Registered Workflow in Public Server (usegalaxy.org)

Through Galaxy Public Workflow Website, your workflow can be shared with other scientists and researchers. MGEScan workflow has been registed on https://usegalaxy.org/workflow/list_published.



2.7 Overview of MGEScan Workflow (Draft)

The published MGEScan workflow consists of LTR and non-LTR programs in parallel. LTR has four components including splitting scaffolds, pre-processing by repeatmasker, finding LTRs, and converting results in gff3 format.

Quick Start

CHAPTER 3

MGEScan Command Line Interface

MGEScan provides Command Line Interface (CLI) along with Galaxy Web Interface. You can run MGEScan-LTR and MGEScan-nonLTR programs on your shell terminal.

3.1 Installation

If you have installed MGEScan on Galaxy, MGEScan CLI tools are available on your system.

Note: Do you need to install MGEScan? See here for *Installation*. Follow the instructions except the Galaxy. You can skip the Galaxy installation if you need MGEScan CLI tools only.

3.1.1 Installation in Userspace

It is possible to install MGEScan on userspace without root permission. Please follow the instructions below. `virtualenv` is required. Create your virtualenv and activate it like:

```
virtualenv $HOME/virtualenv/mgescan
source $HOME/virtualenv/mgescan/bin/activate
```

Once your virtualenv is activated, you will see `(mgescan)` label in your prompt.

Note: Don't forget to activate your virtualenv when you open a new session. source $HOME/virtualenv/mgescan/bin/activate

```
git clone https://github.com/MGEScan/mgescan.git
cd mgescan
python setup.py install
```

You will see a (Y/n) prompt for your input like:

```
$MGESCAN_HOME is not defined where MGESCAN will be installed.
Would you install MGESCAN at /$HOME/mgescan3 (Y/n)?
```

`$HOME/mgescan3` is a default path to install MGEScan. Proceed to install MGEScan in the default directory `$HOME/mgescan3`. If you like to install MGEScan in other location, define MGESCAN_HOME environment variable like this:

```
export MGESCAN_HOME=<desired location to install mgescan>
e.g.
export MGESCAN_HOME=/home/abc/program/mgescan
```

3.2 Usage

Try `mgescan -h` on your terminal:

```
(mgescan)$ mgescan -h
MGEScan: identifying ltr and non-ltr in genome sequences

Usage:
        mgescan both <genome_dir> [--output=<data_dir>] [--mpi=<num>]
        mgescan ltr <genome_dir> [--output=<data_dir>] [--mpi=<num>]
        mgescan nonltr <genome_dir> [--output=<data_dir>] [--mpi=<num>]
        mgescan (-h | --help)
        mgescan --version

Options:
        -h --help   Show this screen.
        --version   Show version.
        --output=<data_dir> Directory results will be saved
```

3.3 MGEScan Programs

`mgescan` CLI tool provides options to run `ltr`, `nonltr` or both programs.

3.3.1 How to Run

If you need to run MGEScan program to indentify both LTR and non-LTR for certain genome sequences, simply specify the path where your input genome files (FASTA format) exist with `both` sub-command.

For example, if you have DNA sequences (FASTA) for Fruitfly (Drosophila melanogaster) under `$HOME/dmelanogaster` directory, and want to save results in the `$HOME/mgescan_result_dmelanogaster`, your may run `mgescan` command like so:

```
(mgescan)$ mgescan both $HOME/dmelanogaster --output=$HOME/mgescan_result_
↪dmelanogaster
```

The expected output message is like so:

```
ltr: starting
nonltr: starting
nonltr: finishing (elapsed time: 306.881129026)
ltr: finishing (elapsed time: 1306.881129026)
```

MPI Option

If your system supports a MPI program, you can use `--mpi` option with a number of processes. Use half number of your cores.

Input Files (FASTA)

The input can be a single file with a single sequence or multiple sequences. Store your input DNA sequences in a same folder and specify the path when you run MGEScan program. For example, if you run the program for D. melanogaster, you may have sequence files like so:

```
$ ls -al dmelanogaster
total 167564
drwx------  2 mgescan mgescan     4096 Jan 28 23:23 .
drwx------ 13 mgescan mgescan     4096 Apr  7 18:45 ..
-rw-------  1 mgescan mgescan 23395126 Dec 18  2014 2L.fa
-rw-------  1 mgescan mgescan 21499210 Dec 18  2014 2R.fa
-rw-------  1 mgescan mgescan 24952673 Dec 18  2014 3L.fa
-rw-------  1 mgescan mgescan 28370194 Dec 18  2014 3R.fa
-rw-------  1 mgescan mgescan  1374441 Dec 18  2014 4.fa
-rw-------  1 mgescan mgescan 22796595 Dec 18  2014 X.fa
-rw-------  1 mgescan mgescan  2796595 Dec 18  2014 Y.fa
```

3.3.2 Results

Upon the succeessful completion of MGEScan program, several output files are stored in the destination directory that you specified with `--output` parameter. It includes plain text and gff3 files.

`ltr.out`

MGEScan LTR generates `ltr.out` to describe clusters and coordinates of LTR retrotransposons identified. Each cluster of LTR retrotransposons starts with the head line of [cluster_number]———, followed by the information of LTR retrotransposons in the cluster. The columns for LTR retrotransposons are as follows.

1. LTR_id: unique id of LTRs identified. It consist of two components, sequence file name and id in the file. For example, chr1_2 is the second LTR retrotransposon in the chr1 file.
2. start position of 5 LTR.
3. end position of 5 LTR.
4. start position of 3 LTR.
5. end position of 3 LTR.
6. strand: + or -.
7. length of 5 LTR.
8. length of 3 LTR.

9. length of the LTR retrotransposon. 10.TSD on the left side of the LTR retotransposons. 11.TSD on the right side of the LTR retrotransposons. 12.di(tri)nucleotide on the left side of 5LTR 13.di(tri)nucleotide on the right side of 5LTR 14.di(tri)nucleotide on the left side of 3LTR 15.di(tri)nucleotide on the right side of 3LTR

Sample output of `ltr.out` for D. melanogaster

`ltr.out`

CHAPTER 4

MGEScan on Galaxy Installation

MGEScan on Galaxy can be installed on a local machine or on the cloud e.g. Amazon EC2. The local installation is for Ubuntu 14.04+ distribution. Others (e.g. OpenSUSE, Fedora) are not verified.

Tip: approximate time: 20 minutes

4.1 Preparation

There are required software to be installed prior to run MGEScan. You need to install system packages with `sudo` command (admin `root` privilege is required). `virtualenv` is used for Python package installation.

- `root` privilege to install packages with **sudo**

4.2 Quick Installation

One-liner command provides a quick installation of required software and configuration.

Warning: This one-liner installation script runs several commands without any further confirmation from you. If you'd like to verify each step, skip this quick installation and follow the installation instuctions below.

```
curl -L https://raw.githubusercontent.com/MGEScan/mgescan/master/one-liner/ubuntu |␣
↪bash
```

Start a Galaxy/MGEscan web server with a default port `38080`.

```
source ~/.mgescanrc
cd $GALAXY_HOME
nohup sh run.sh &
```

Note: RepeatMasker is not included.

Note: Default admin account is `mgescan_admin@mgescan.com`. Sign up with this account name and your password.

4.3 Normal Installation

4.4 Software for Python

If `virtualenv`, `git`, and `python-dev` are available on your system, you can skip this step.

Ubuntu

```
sudo apt-get update
sudo apt-get install python-virtualenv python-dev git -y
```

Fedora

```
sudo yum update
sudo yum install python-virtualenv python-devel git -y
```

4.5 Environment Variables

MGEScan will be installed on a default directory `$HOME/mgescan3`. You can change it if you prefer other location to install MGEScan.

```
export MGESCAN_HOME=$HOME/mgescan3
export MGESCAN_SRC=$MGESCAN_HOME/src
export GALAXY_HOME=$MGESCAN_HOME/galaxy
export TRF_HOME=$MGESCAN_HOME/trf
export RM_HOME=$MGESCAN_HOME/RepeatMasker
export MGESCAN_VENV=$MGESCAN_HOME/virtualenv/mgescan
```

Tip: MGEScan on Galaxy uses version 3 in the naming like mgescan3.

Create a MGESCan start file `.mgescanrc`

```
cat <<EOF > $HOME/.mgescanrc
export MGESCAN_HOME=\$HOME/mgescan3
export MGESCAN_SRC=\$MGESCAN_HOME/src
export GALAXY_HOME=\$MGESCAN_HOME/galaxy
export TRF_HOME=\$MGESCAN_HOME/trf
export RM_HOME=\$MGESCAN_HOME/RepeatMasker
export MGESCAN_VENV=\$MGESCAN_HOME/virtualenv/mgescan
EOF
```

Then include it to your startup file (i.e. `.bash_profile`).

```
echo "source ~/.mgescanrc" >> $HOME/.bash_profile
```

Create a main directory.

```
source ~/.mgescanrc
mkdir $MGESCAN_HOME
```

4.6 Software for MGEScan

Galaxy Workflow, HMMER (3.1b1), EMBOSS Suite and TRF are required. RepeatMasker is optional.

4.6.1 Galaxy

Tip: Make sure that $MGESCAN_HOME is set by `echo $MGESCAN_HOME` command. If you don't see a path similar to `/home/.../mgescan3/`, you have to define environment variables again.

From Github repository (source code):

```
cd $MGESCAN_HOME
git clone https://github.com/galaxyproject/galaxy/
```

4.6.2 HMMER and EMBOSS

If you have `HMMER` and `EMBOSS` on your system, you can skip this step.

Ubuntu

```
sudo apt-get install hmmer emboss -y
```

Fedora

- HMMER v3.1b2

```
sudo yum install gcc -y
wget ftp://selab.janelia.org/pub/software/hmmer3/3.1b2/hmmer-3.1b2-linux-intel-x86_64.
↪tar.gz
tar xvzf hmmer-3.1b2-linux-intel-x86_64.tar.gz
cd  hmmer-3.1b2-linux-intel-x86_64
./configure
make
make check
make install
```

- EMBOSS 6.6.0 (latest)

```
wget ftp://emboss.open-bio.org/pub/EMBOSS/emboss-latest.tar.gz
tar xvzf emboss-latest.tar.gz
cd EMBOSS-*
./configure
make
```

```
make check
make install
```

4.6.3 Open MPI

Ubuntu

```
sudo apt-get install openmpi-bin libopenmpi-dev -y
```

4.6.4 Virtual Environments (virtualenv) for Python Packages

It is recommended to have an isolated environment for MGEScan Python libraries. virtualenv creates a separated space for MGEScan, and issues from dependencies and versions of Python libraries can be avoided. Note that you have to be in the virtualenv of MGEScan before to run any MGEScan command line tools. The following commands create a virtualenv for MGEScan and enable it on your account.

```
mkdir -p $MGESCAN_VENV
virtualenv $MGESCAN_VENV
source $MGESCAN_VENV/bin/activate
echo "source $MGESCAN_VENV/bin/activate" >> ~/.bash_profile
```

Note: Skip the last line `echo "source ..."`, if you'd like to enable `mgescan` virtualenv manually.

4.6.5 Tandem Repeats Finder (trf)

`trf` is a single binary executable file to locate and display tandem repeats in DNA sequences. MGEScan-LTR requires `trf` program.

```
mkdir -p $TRF_HOME
wget http://tandem.bu.edu/trf/downloads/trf407b.linux64 -P $TRF_HOME
```

4.6.6 RepeatMasker (Optional)

RepeatMasker is a program that screens DNA sequences for interspersed repeats and low complexity DNA sequences. MGEScan-LTR has an option to use RepeatMasker.

```
mkdir $RM_HOME
wget http://www.repeatmasker.org/RepeatMasker-open-4-0-5.tar.gz
tar xvzf RepeatMasker-open-4-0-5.tar.gz
mv RepeatMasker/* $RM_HOME
ln -s $RM_HOME/RepeatMasker $MGESCAN_VENV/bin/
```

4.7 MGEScan Installation

MGEScan can be installed from Github repository (source code):

```
cd $MGESCAN_HOME
git clone https://github.com/MGEScan/mgescan.git
ln -s mgescan src
cd $MGESCAN_SRC
python setup.py install
```

4.8 Configuration

4.8.1 Virtual Environments (virtualenv)

Make sure you have loaded your virtual environment for MGEScan by:

```
source $MGESCAN_VENV/bin/activate
```

You will see `(mgescan)` label on your prompt.

4.8.2 Galaxy Configurations for MGEScan

MGEScan github repository contains codes and toolkits for MGEScan on Galaxy. Prior to run a Galaxy Workflow web server, the codes and toolkits should be installed in the `galaxy` main directory.

```
cp -pr $MGESCAN_SRC/galaxy-modified/* $GALAXY_HOME
```

4.8.3 trf

To run `trf` anywhere under `mgescan` virtualenv, we create a symlink in the `bin` directory.

```
ln -s $TRF_HOME/trf407b.linux64 $MGESCAN_VENV/bin/trf
chmod 700 $MGESCAN_VENV/bin/trf
```

4.8.4 RepeatMasker

RepeatMasker also requires configuration.

Ubuntu

```
cd $RM_HOME
$RM_HOME/configure
```

Fedora

```
sudo yum install perl-Data-Dumper perl-Text-Soundex -y
cd $RM_HOME
$RM_HOME/configure
```

Outputs like so:

```
RepeatMasker Configuration Program

This program assists with the configuration of the
```

```
RepeatMasker program.  The next set of screens will ask
you to enter information pertaining to your system
configuration.  At the end of the program your RepeatMasker
installation will be ready to use.

 <PRESS ENTER TO CONTINUE>
```

4.8.5 Galaxy Admin User

Declare your email address as a Galaxy admin user name.

```
export GALAXY_ADMIN=mgescan_admin@mgescan.com
```

Warning: REPLACE `mgescan_admin@mgescan.com` with your email address. You also have to sign up Galaxy with this email address.

```
sed -i "s/#admin_users = None/admin_users = $GALAXY_ADMIN/" $GALAXY_HOME/universe_
↪wsgi.ini
```

4.9 Start Galaxy

Simple `run.sh` script starts a Galaxy web server. First run of the script takes some time to initialize database.

```
cd $GALAXY_HOME
nohup sh run.sh &
```

Note: Default port number : 38080 http://{[}IP ADDRESS]:38080

CHAPTER 5

MGEScan ToolShed

MGEScan is available in Galaxy ToolShed to install MGEScan tools and dependencies from the public Galaxy Tool Shed (https://toolshed.g2.bx.psu.edu). A few clicks allow you to install MGEScan and required software easily e.g. HMMER, Tandem Repeat Finder, and EMBOSS. The following installation guide explains how to apply MGEScan to your existing or brand new Galaxy server using ToolShed.

5.1 Installation Guide

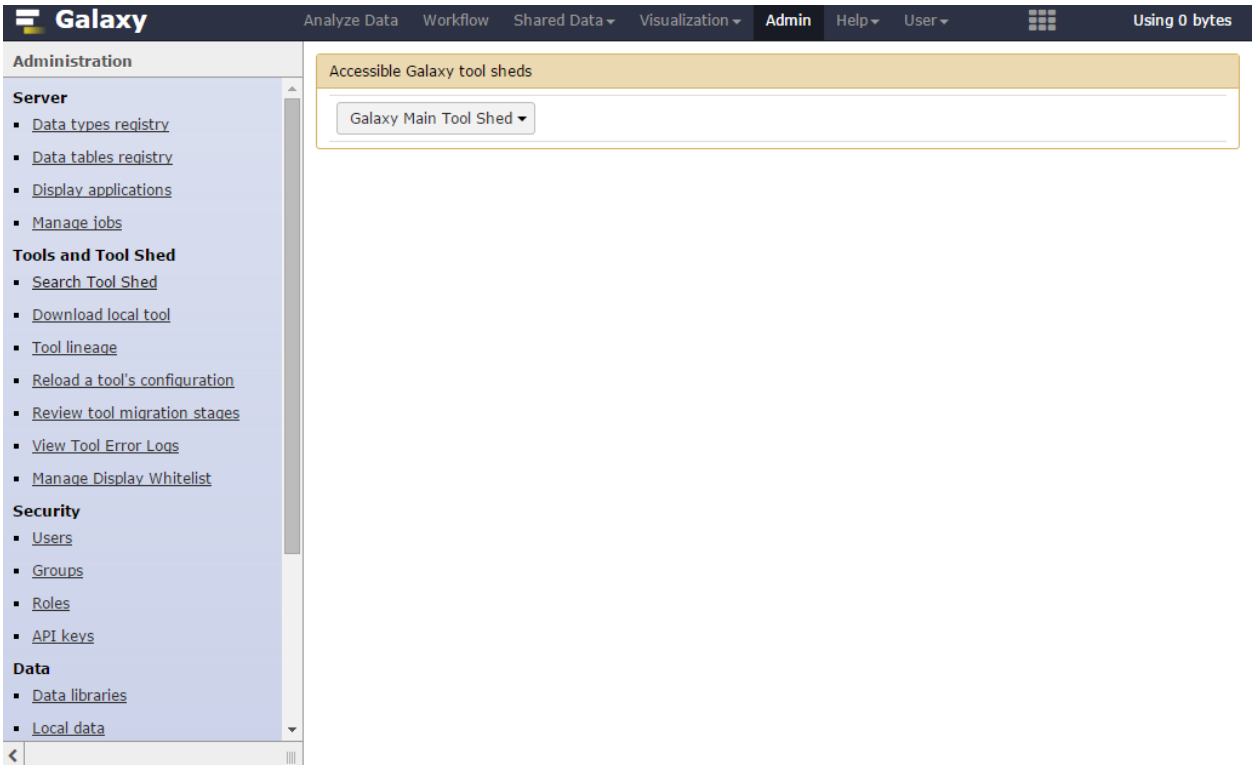
5.1.1 Prerequite

You need to make sure the following system packages are available on your system prior to install MGEScan.

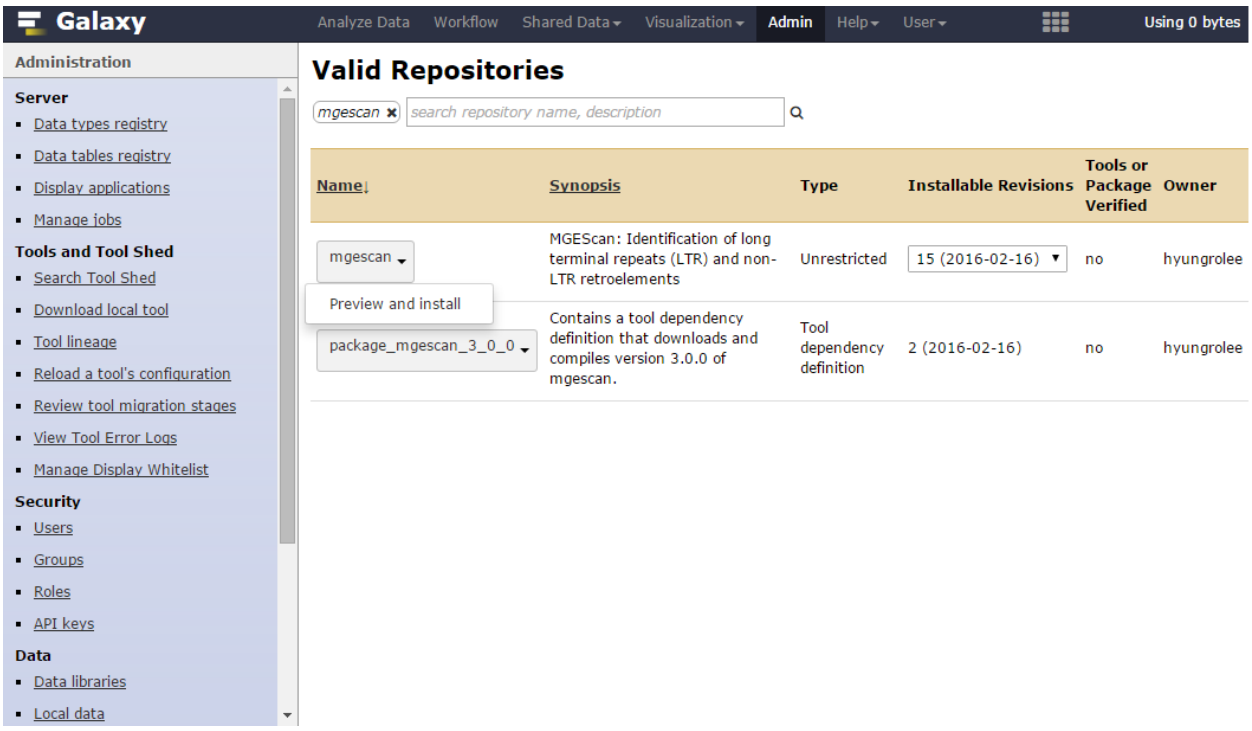
- Python pip
- Python setuptools
- Python dev package
- MPI for parallel processing (i.e. openmpi-bin, libopenmpi-dev on Ubuntu)

5.1.2 Admin Page

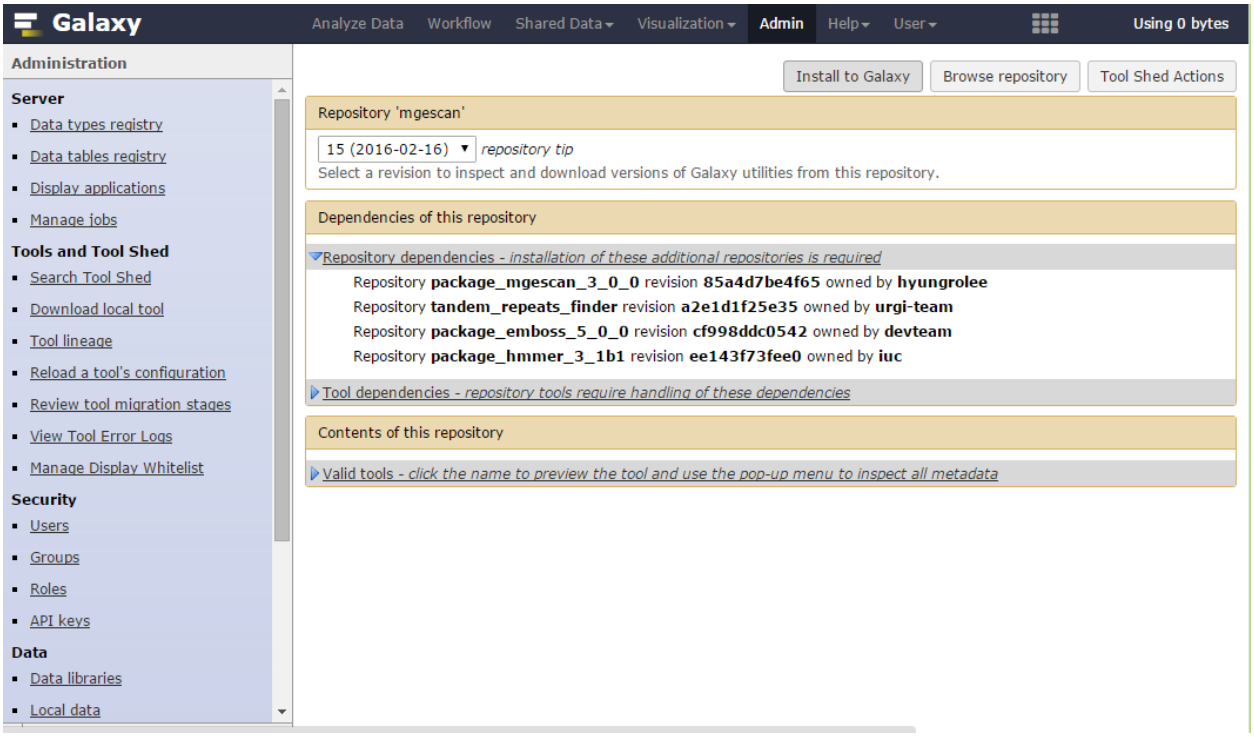
Admin user only is able to add a new Galaxy Tool from ToolShed. Find **Admin** link from the top menu tab. Click **Search Tool Shed** on the left menu tab of the Admin page.



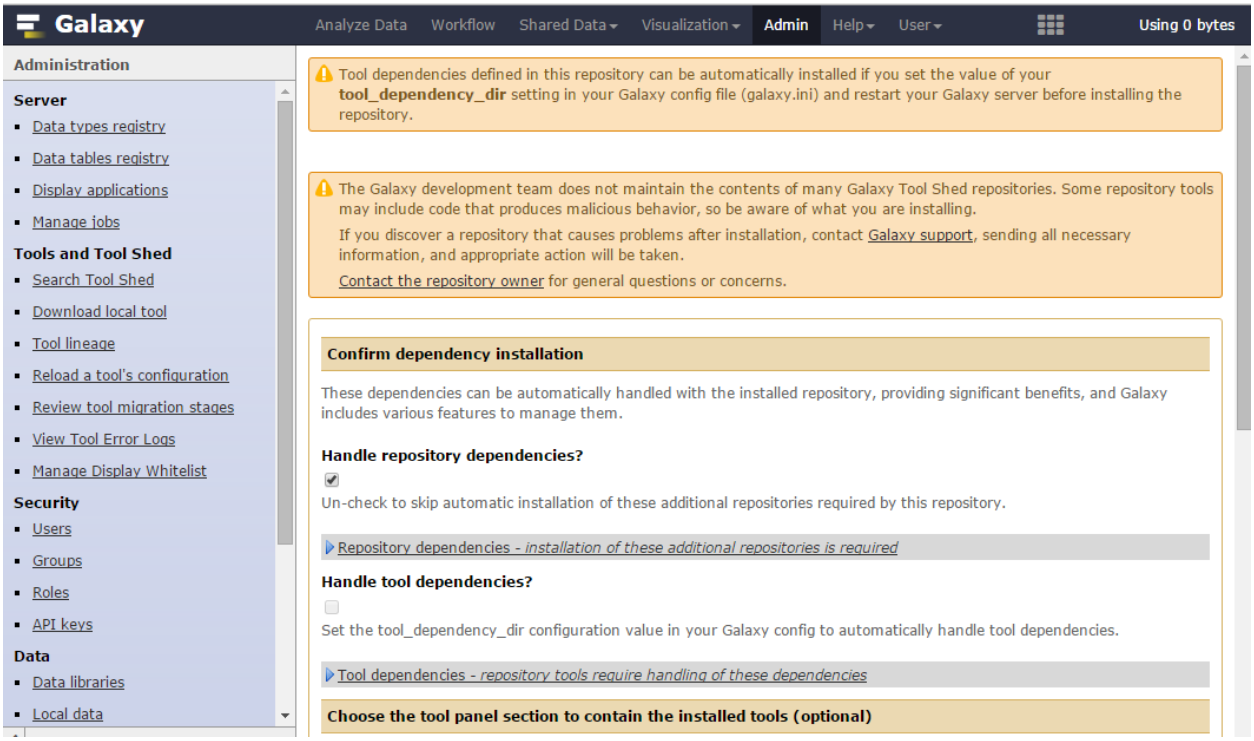
If you can find 'Galaxy Main Tool Shed' select button on the right page, click **Browse valid repositories**. It redirects the page to the public Galaxy toolshed page in which 3,728 tools available in 2016.



Type `mgescan` in the search box. Choose `mgescan`, not `package_mgescan_3_0_0` to preview and install.

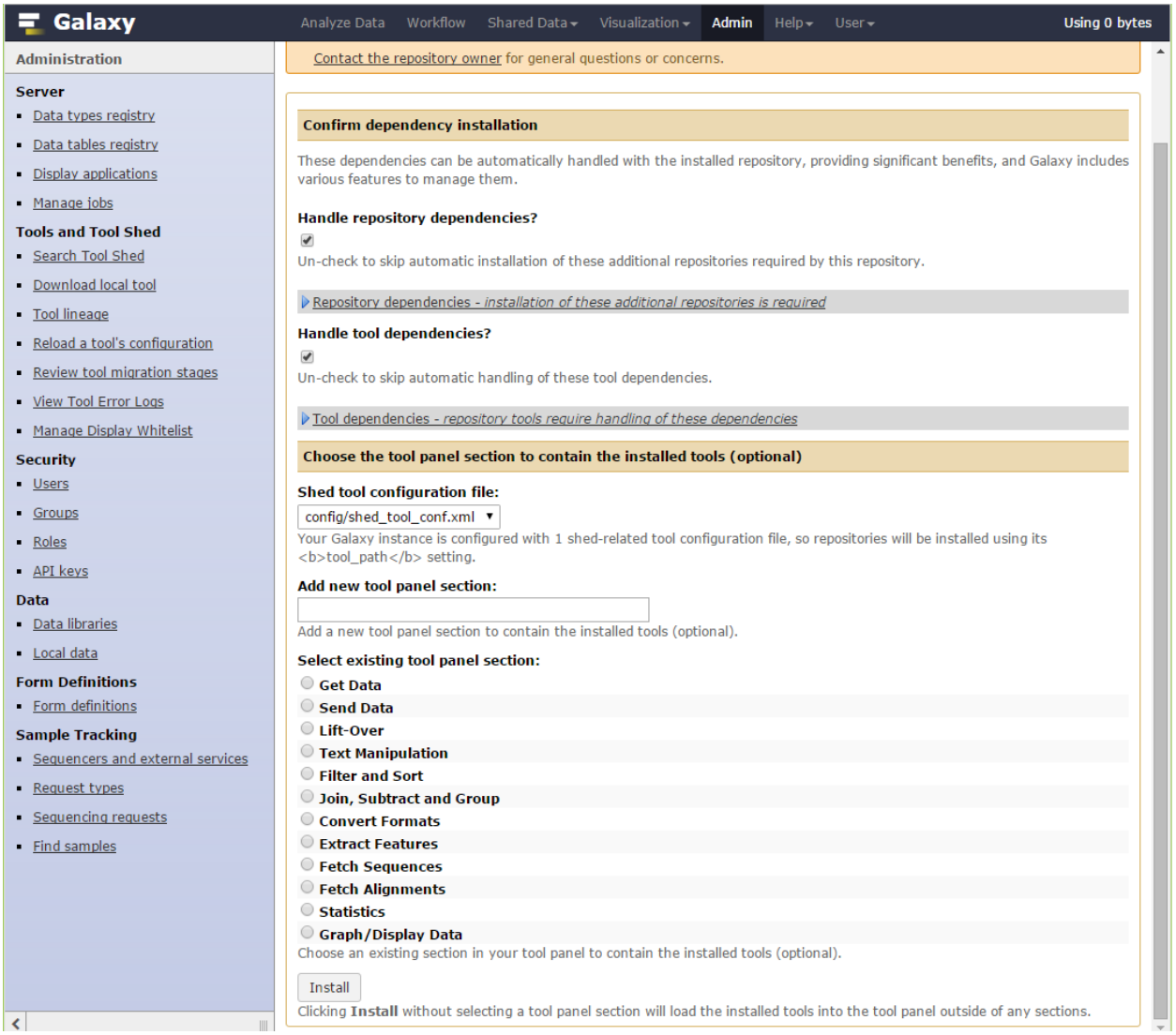


You may find there are other dependencies to be installed as well. If you are ready to install, find **Install to Galaxy** button on top of the page. It goes to the confirmation page.

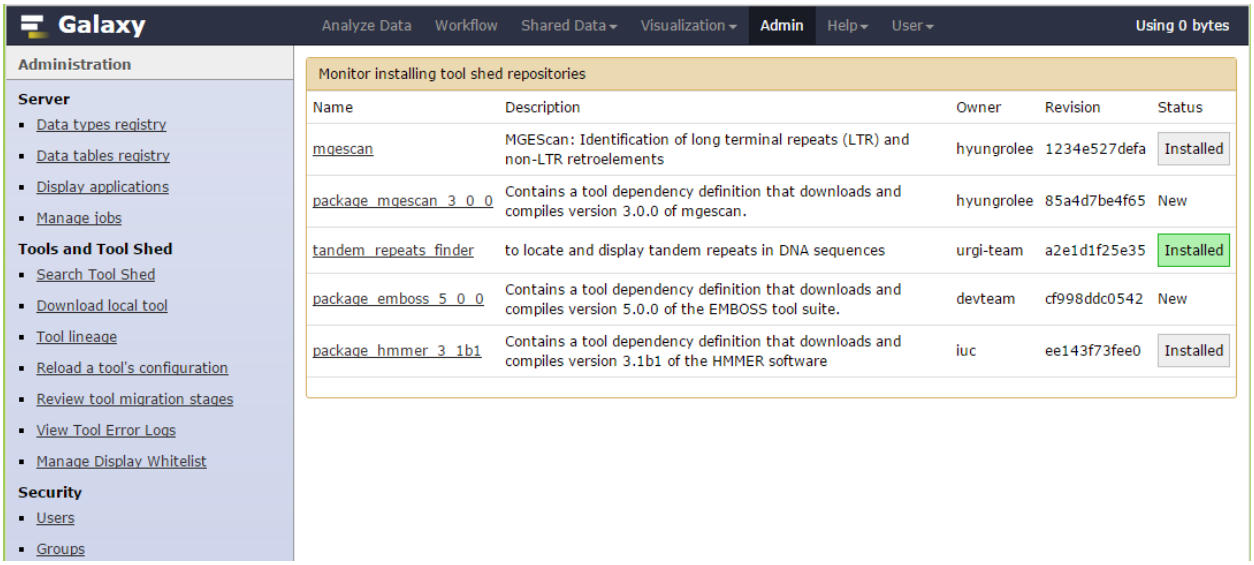


Note: You need to make sure *repository dependencies* and *tool dependencies* are checked in the page. Otherwise, necessary tools or repositories may not be installed properly.

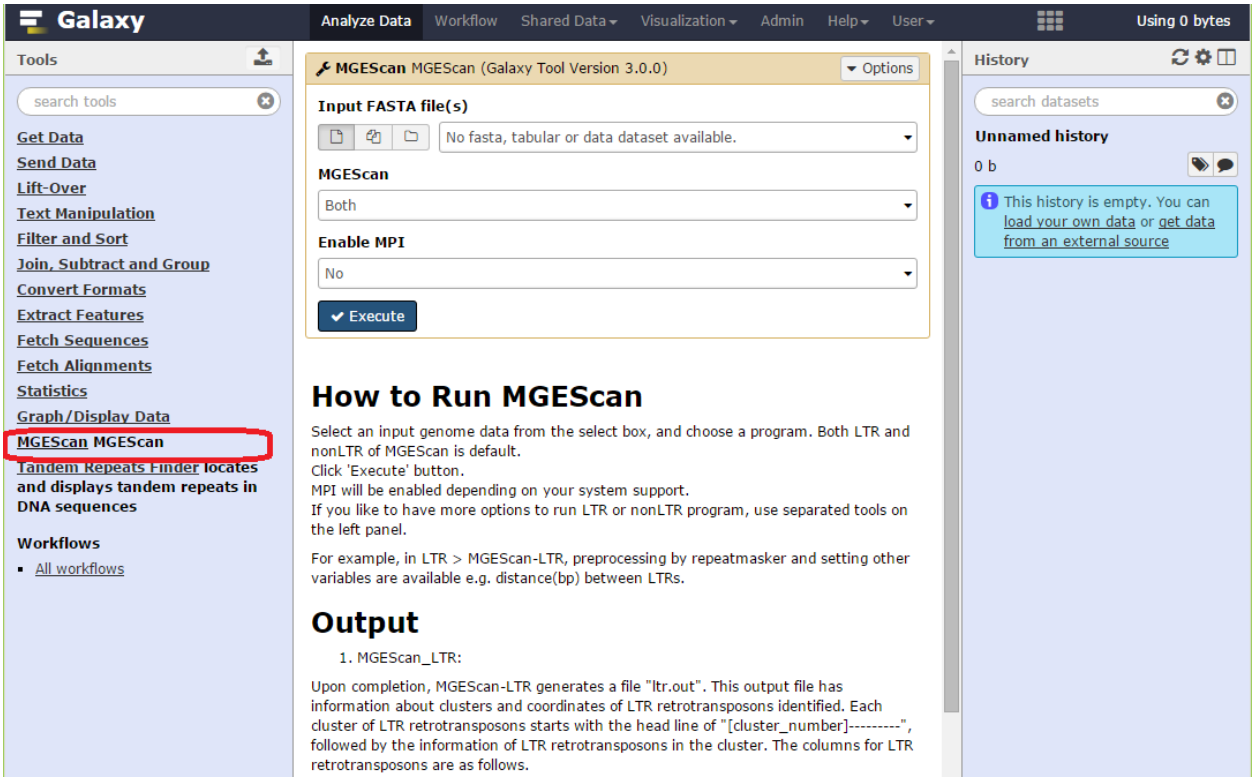
You will find **Install** button at the bottom of the page. Once you click the button, your Galaxy server starts to download tools and repositories and install MGEScan on your Galaxy.



You can find MGEScan from **Manage installed tools** page from the left menu tab in the admin page. `mgescan` tool adds EMBSS, HMMER, Tandem Repeat Finder and MGEScan packages. You need to find all these tools are successfully installed. *Installation Status* indicates whether is is installed properly with colors. *Installed* with light green box indicates the tool or package installation is succeeded, if you see grey box, there is some issue in the installation.

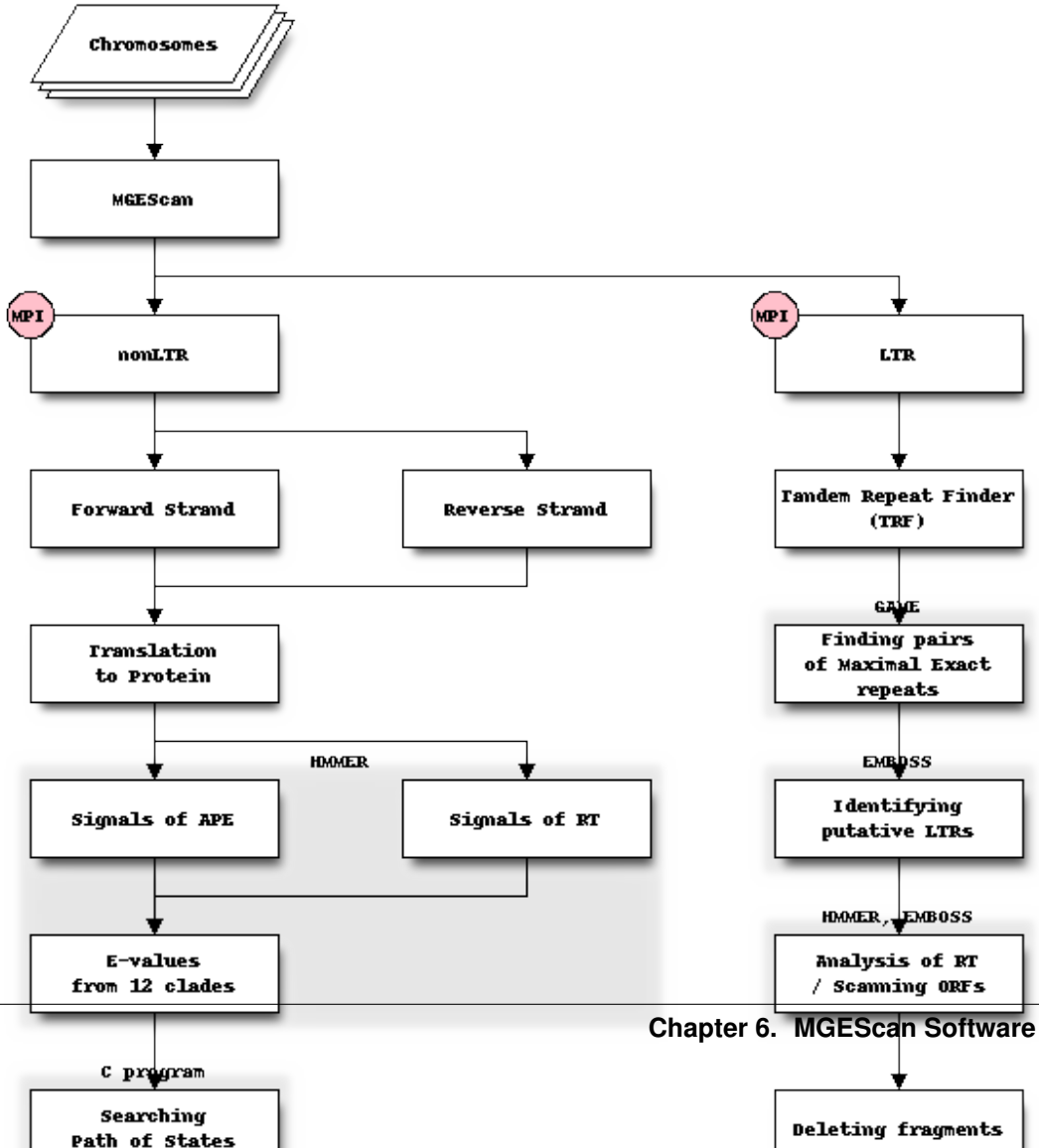


Go to the main page of your Galaxy. The new **MGEScan MGEScan** tool is available on your left tool menu tab.



CHAPTER 6

MGEScan Software Process



- HMMER: hmmsearch 3.1b1 http://hmmer.org/
- EMBOSS: matcher, transeq http://emboss.sourceforge.net/, http://www.ebi.ac.uk/Tools/emboss/
- GAME: Choi, Jeong-Hyeon, Hwan-Gue Cho, and Sun Kim. "GAME: a simple and efficient whole genome alignment method using maximal exact match filtering." Computational biology and chemistry 29.3 (2005): 244-253.
- Tandem Repeats Finder: https://tandem.bu.edu/trf/trf.html
- MGEScan: https://github.com/mgescan/mgescan

CHAPTER 7

MGEScan on Amazon Cloud (EC2)

With Amazon Cloud Web Services, a virtual single or distributed system for MGEScan can be easily deployed. MGEScan (Amazon machine image ID: ami-10672b7a on 'US East-Ohio' region) is available to create our Galaxy-based system for MGEScan which is identifying long terminal repeats (LTR) and non-LTR retroelements in eukaryotic genomic sequences. More cloud options will be available soon including Google Compute Engine, Microsoft Windows Azure or private cloudplatforms such as OpenStack and Eucalyptus.

Note: ami-10672b7a was created in 2015. To apply new updates of MGEScan and Galaxy, follow the instructions below after launching the image on AWS EC2.

- Stop Galaxy server first - processs looks like *python ./scripts/paster.py serve universe_wsgi.ini*
- Update system packages `sudo yum update -y`
- Update mgescan code `cd $MGESCAN_SRC;git pull;python setup.py install`
- Update Galaxy code `cd $GALAXY_HOME;git pull`
- Migrate Galaxy DB, if necessary `cd $GALAXY_HOME;./run.sh;sh manage_db.sh -c ./universe_wsgi.ini upgrade`
- Update Galaxy tools `cp -pr $MGESCAN_SRC/galaxy-modified/* $GALAXY_HOME`
- Start Galaxy server `cd $GALAXY_HOME;nohup bash run.sh &`

Command lines only

```
kill `ps -ef|grep universe_wsgi|grep -v grep|awk '{print $2}'`
sudo yum update -y
cd $MGESCAN_SRC;git pull;python setup.py install
cd $GALAXY_HOME;git pull
cd $GALAXY_HOME;./run.sh;sh manage_db.sh -c ./universe_wsgi.ini upgrade
cp -pr $MGESCAN_SRC/galaxy-modified/* $GALAXY_HOME
cd $GALAXY_HOME;nohup bash run.sh &
```

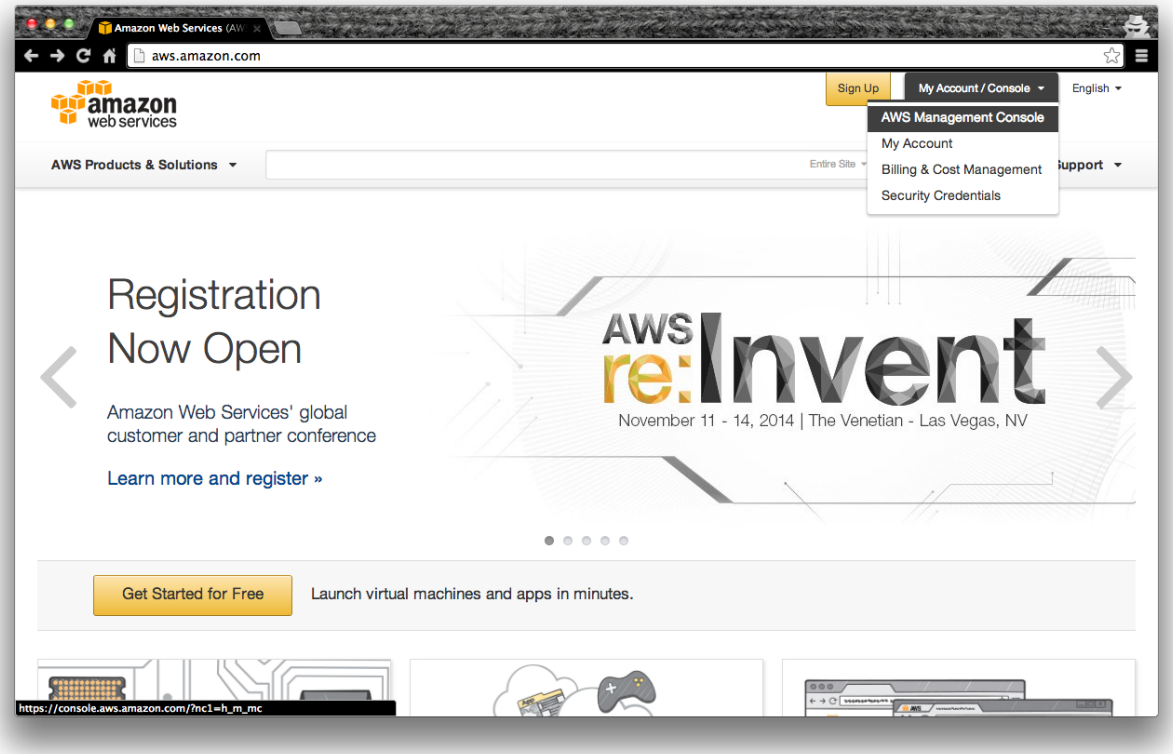
7.1 Deploying MGEScan on Galaxy

First step is getting an Amazon account to launch virtual instances on Amazon IaaS platform EC2.

7.1.1 AWS EC2 Account

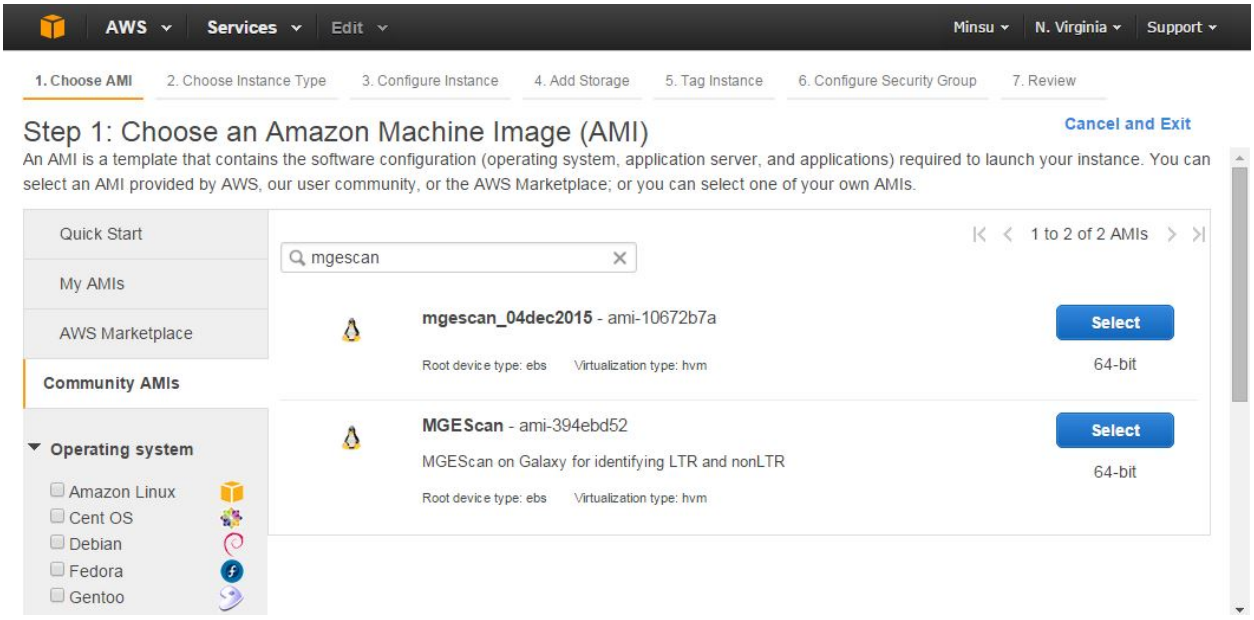
If you already have an account of Amazon AWS EC2, open AWS Management Console to launch our MGEScan image on EC2. Otherwise, create an AWS Account.

- http://aws.amazon.com



7.1.2 MGEScan Machine Image

In AWS Management Console, open *EC2 Dashboard* > *Launch Instance*. To choose an Amazon Machine Image (AMI) of MGEScan, select *Community AMIs* on the left tab, and search by name or id, e.g. mgescan or ami-10672b7a. (US East-Ohio Region Only)



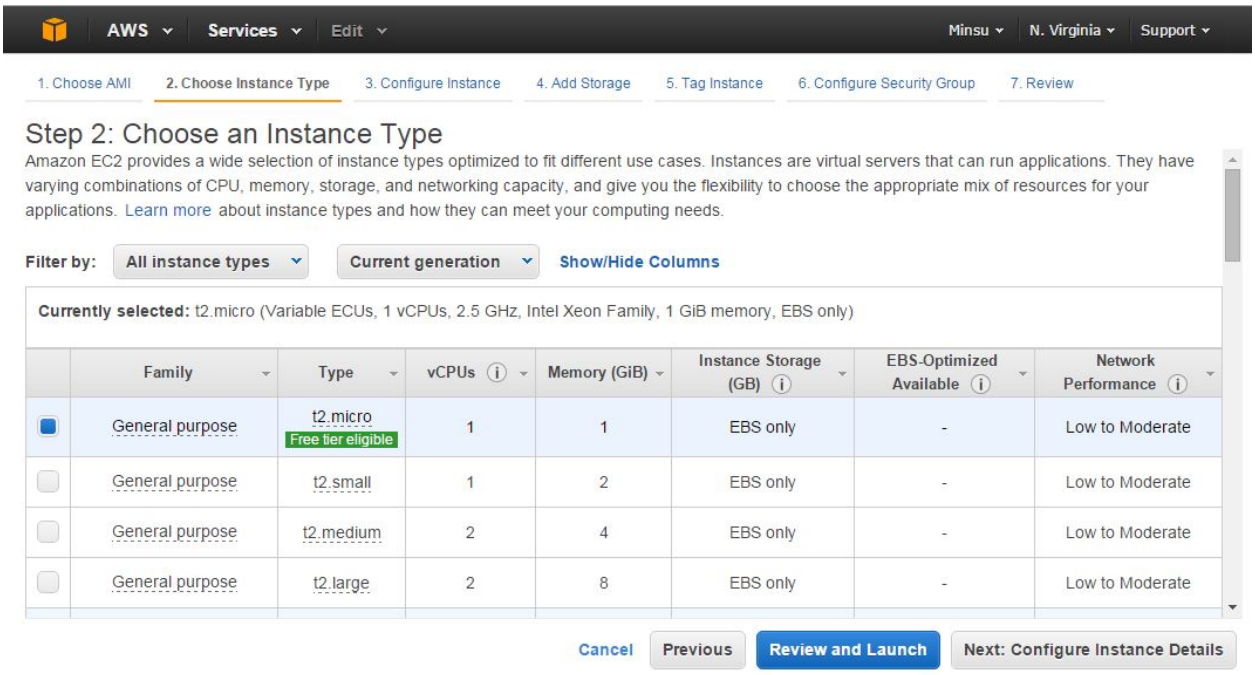
MGEScan EC2 Image Information

- Region: US East
- Image Name: MGEScan
- ID: ami-10672b7a
- Server type: 64bit
- Description: MGEscan on Galaxy for identifying LTR and nonLTR
- Root device type: ebs
- Virtualization type: hvm

7.1.3 Choose an Instance Type for `MGEScan` Instance

Once you choose **MGEScan** image as a base image, you need to select the size of instance. `t2.micro` uses 1 vCPUs and 1 GB memory which is in free tier. Ohter options are available to have large instance e.g. 40 vCPUs. Click **Review and Launch** icon at bottom of the page.

Tip: t2.micro: (Variable ECUs, 1 vCPUs, 2.5 GHz, Intel Xeon Family, 1 GiB memory, EBS only)



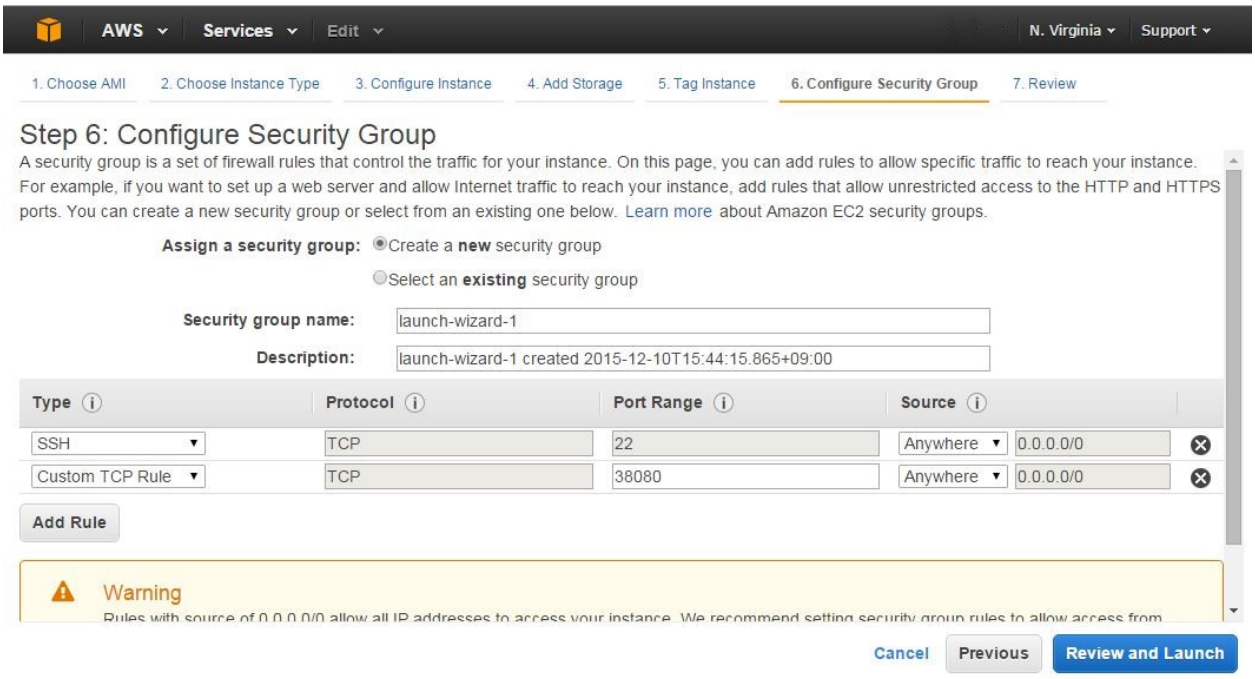
7.1.4 Security Group for Web

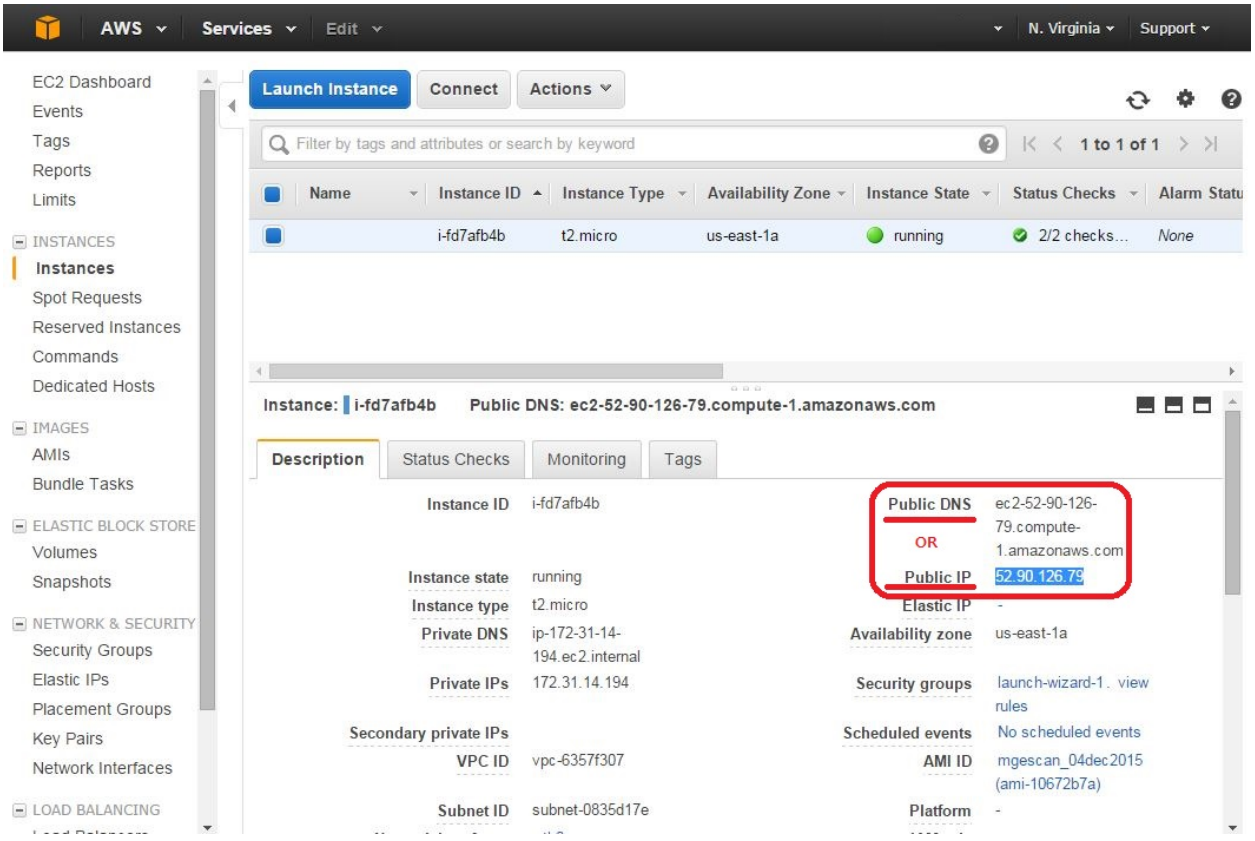
MGEscan / Galaxy uses `38080` default web port. We need to add a rule to have this port opened on the new instance. There are a few steps you have to follow.

- Find "Security Groups" section and click "Edit security groups". "Create a new

security group" is selected as a default with a 22 SSH port opened to anywhere.

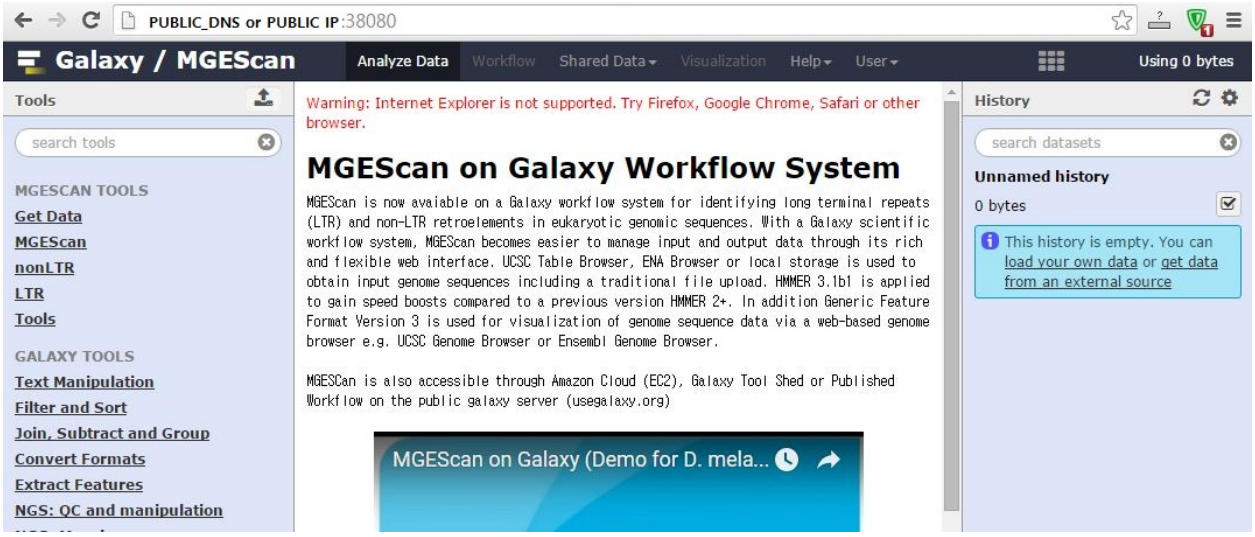
- We will add `38080` tcp port. Click "Add Rule" and type `38080` in the "Port Range" input box.
- Don't forget to update "Source" to **"Anywhere"** from "Custom IP".
- Once you're done, click "Reivew and Launch".
- Click "Launch" again.
- Choose a SSH keypair from existing or new one.
- Click "Launch Instance" and wait.
- Find out public IP address and open a web browser with the address. e.g. http://{[}IP address]:38080 *Don't forget the port number 38080*





7.1.5 Access to MGEScan Instance

Once the MGEScan instance is launched and accessible, galaxy scientific workflow system for MGEScan and SSH connection are avabilable through given dns name.

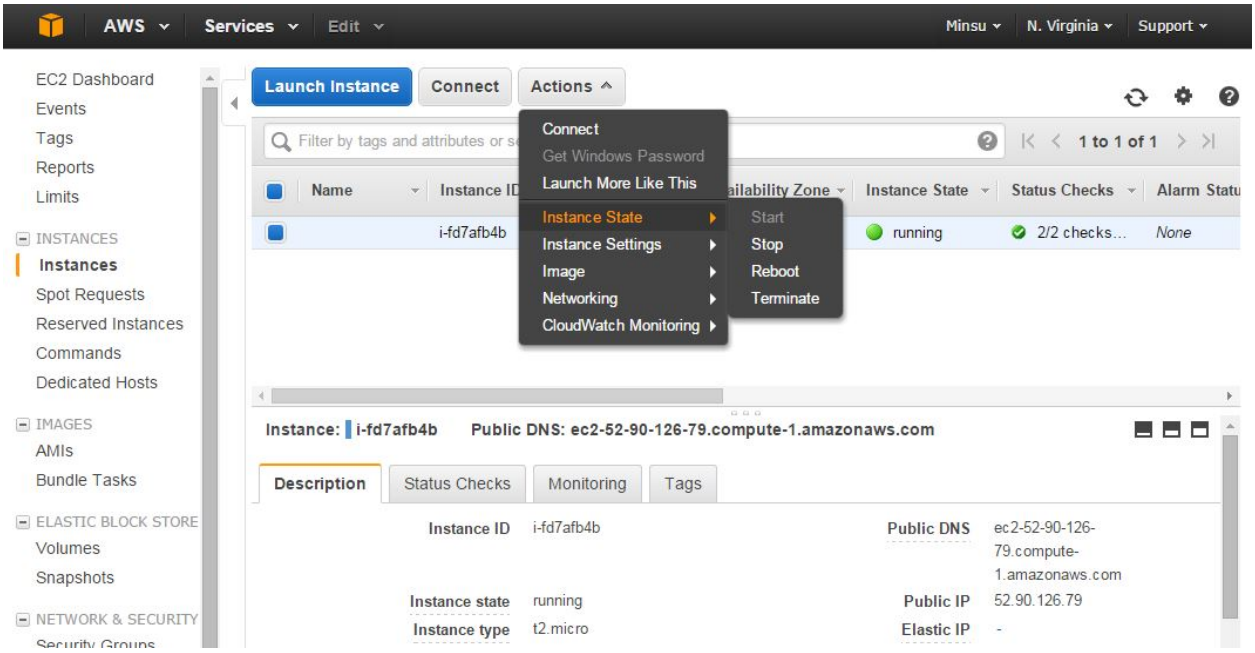


7.1.6 Ready To Use

The MGEScan is now ready to conduct your experiment on Amazon EC2.

Note: Do not forget to terminate your virtual instance after all analysis completed. Amazon Cloud charges use of VM instances hourly.

Terminating AWS Instance:



7.2 Note

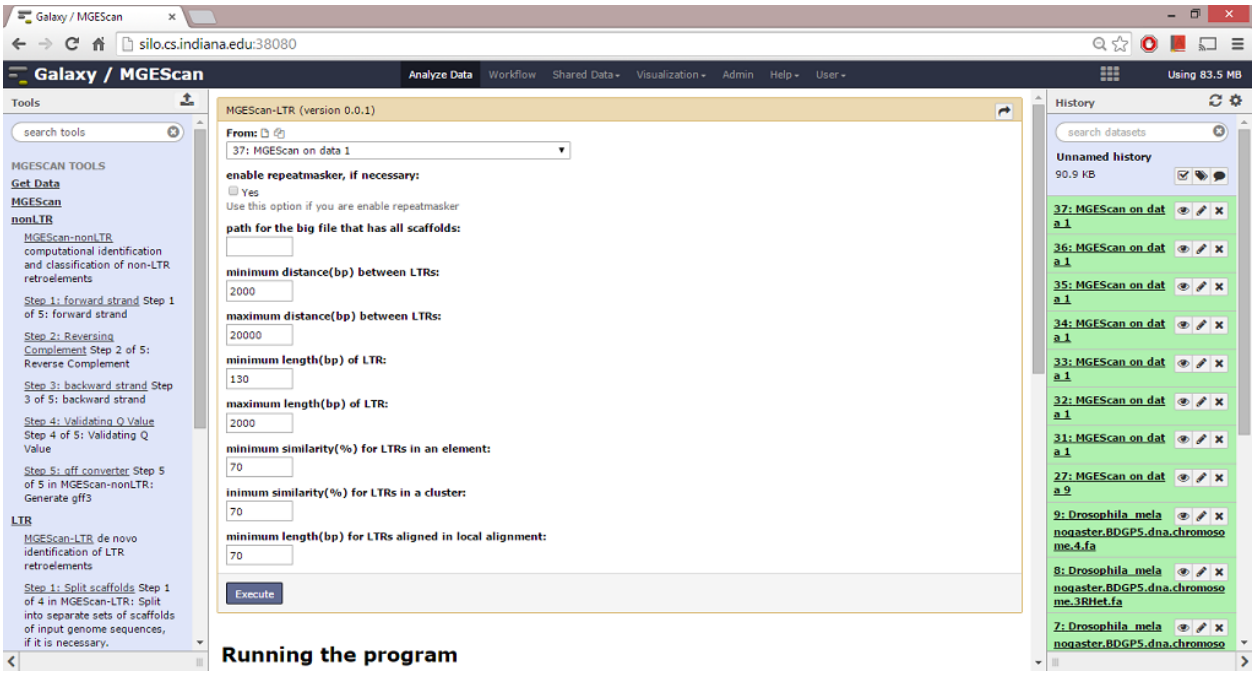
Add a script to auto-start Galaxy after reboot in `/etc/rc.local`

```
su ec2-user -c 'source ~/.mgescanrc;cd $GALAXY_HOME;nohup sh run.sh &'
```

CHAPTER 8

MGEScan-LTR

MGEScan-LTR program identifies long terminal repeats (LTR). RepeatMasker can be used to identify repetitive elements in genomic sequences.



8.1 Description

MGEScan-LTR identifies all types of LTR retrotransposons, i.e., young intact, old intact, and solo LTR retrotransposons, without relying on a library of known elements. It uses approximate string matching, protein domain analysis, and profile Hidden Markov Models to identify intact LTR retrotransposons.

For details, please read following references.

- Rho, M., et al. (2007) De novo identification of LTR retrotransposons in eukaryotic genomes. BMC Genomics, 8, 90.
- Rho, M., et al. (2010) LTR retroelements in the genome of Daphnia pulex. BMC Genomics, 11, 425.

8.2 Running the program

To run MGEScan-LTR, follow the steps below,

- Specify options that you like to have:
  - Check repeatmasker if you want to preprocess
  - Check scaffold if the input file has all scaffolds.
- Update values:
  - min_dist: minimum distance(bp) between LTRs.
  - max_dist: maximum distance(bp) between LTRS
  - min_len_ltr: minimum length(bp) of LTR.
  - max_len_ltr: maximum length(bp) of LTR.
  - ltr_sim_condition: minimum similarity(%) for LTRs in an element.
  - cluster_sim_condition: minimum similarity(%) for LTRs in a cluster
  - len_condition: minimum length(bp) for LTRs aligned in local alignment.
- Click 'Execute'

8.3 Options

- RepeatMasker: Yes / No
- file path for multiple sequences to divide
- settings for LTRs
  - minimum distance(bp) between LTRs
  - maximum distance(bp) between LTRs
  - minimum length(bp) of LTR
  - maximum length(bp) of LTR
  - minimum similarity(%) for LTRs in an element
  - minimum similarity(%) for LTRs in a cluster
  - minimum length(bp) for LTRs aligned in local alignment

8.4 Results

Upon completion, MGEScan-LTR generates a file ltr.out. This output file has information about clusters and coordinates of LTR retrotransposons identified. Each cluster of LTR retrotransposons starts with the head line of [cluster_number]———, followed by the information of LTR retrotransposons in the cluster. The columns for LTR retrotransposons are as follows.

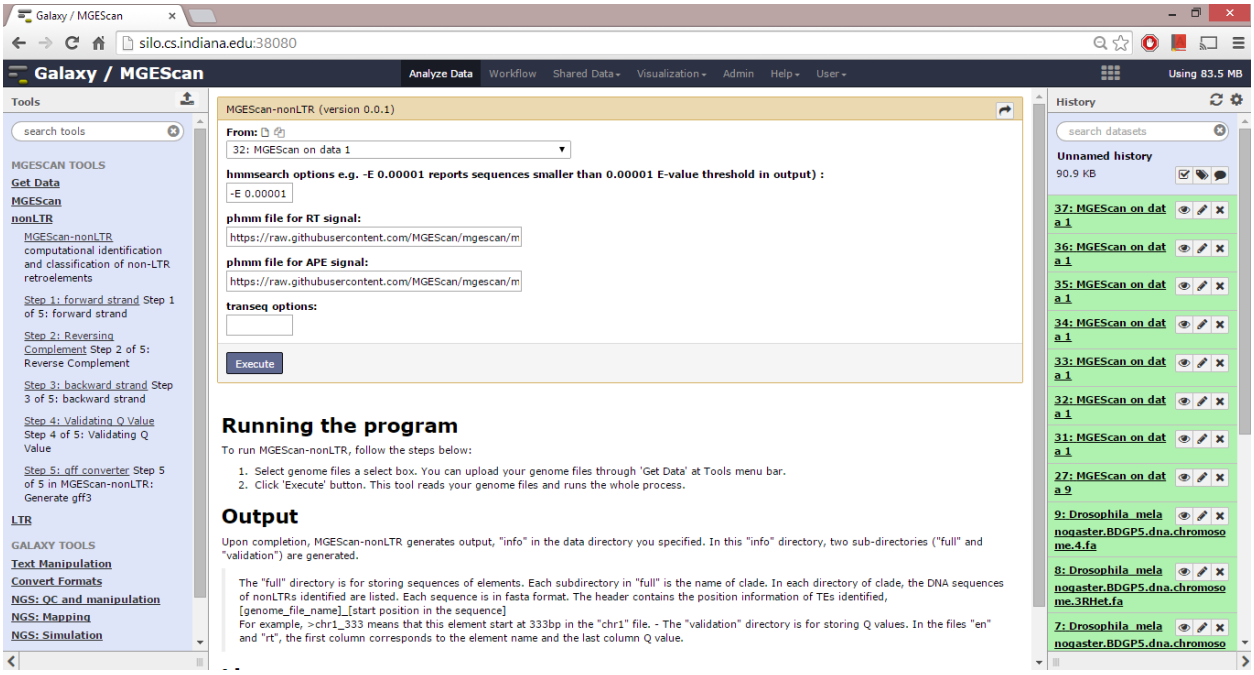
- LTR_id: unique id of LTRs identified. It consist of two components, sequence file name and id in the file. For example, chr1_2 is the second LTR retrotransposon in the chr1 file.
- start position of 5 LTR.
- end position of 5 LTR.
- start position of 3 LTR.
- end position of 3 LTR.
- strand: + or -.
- length of 5 LTR.
- length of 3 LTR.
- length of the LTR retrotransposon.
- TSD on the left side of the LTR retotransposons.
- TSD on the right side of the LTR retrotransposons.
- di(tri)nucleotide on the left side of 5LTR
- di(tri)nucleotide on the right side of 5LTR
- di(tri)nucleotide on the left side of 3LTR
- di(tri)nucleotide on the right side of 3LTR

8.5 License

CHAPTER 9

MGEScan-nonLTR

MGEScan-nonLTR is a program to identify non-long terminal repeat (non-LTR) retrotransposons in genomic sequences. A few options are available in the Galaxy workflow system to configure the program settings, e.g. hmmsearch of protein sequence database with a profile hidden Markov model (HMM).



9.1 Description

MGEScan-nonLTR identifies non-LTR retrotransposons based on Gaussian Bayes classifiers and generalized hidden Markov models consisting of twelve super states that correspond to different clades or closely related clades.

For details, please read following reference.

- Rho, M., Tang, H. (2009) MGEScan-non-LTR: computational identification and classification of autonomous non-LTR retrotransposons in eukaryotic genomes. Nucleic Acids Research, 37(21), e143.

9.2 Running the program

To run MGEScan-nonLTR, follow the steps below:

- Select genome files a select box. You can upload your genome files through 'Get Data' at Tools menu bar.
- Click 'Execute' button. This tool reads your genome files and runs the whole process.

9.3 Options

- hmmmsearch options e.g. -E 0.00001 : reports sequences smaller than 0.00001 E-value threshold in output
- URL of the profile files for RT and APE
- EMBOSS transeq options

9.4 Results

Upon completion, MGEScan-nonLTR generates output, "info" in the data directory you specified. In this "info" directory, two sub-directories ("full" and "validation") are generated.

The "full" directory is for storing sequences of elements. Each subdirectory in "full" is the name of clade. In each directory of clade, the DNA sequences of nonLTRs identified are listed. Each sequence is in fasta format. The header contains the position information of TEs identified, [genome_file_name]_[start position in the sequence] For example, >chr1_333 means that this element start at 333bp in the "chr1" file. - The "validation" directory is for storing Q values. In the files "en" and "rt", the first column corresponds to the element name and the last column Q value.

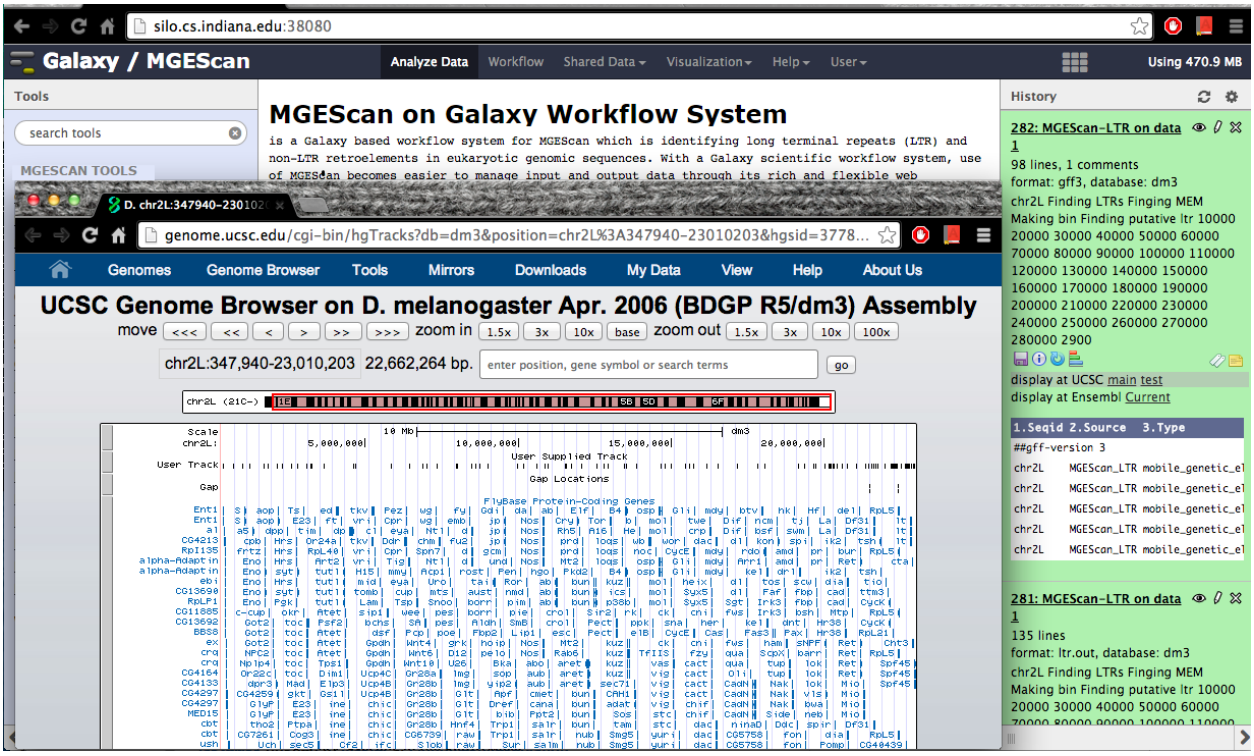
9.5 License

CHAPTER 10

Visualization

Galaxy Workflow System helps display results using genome browsers such as UCSC or Ensembl. MGEScan supports General Feature Format (GFF) to describe genes of MGEScan results so both ltr and non-ltr results can be views via UCSC Genome Browser or Ensembl.

10.1 UCSC Genome Browser



10.2 Source Code

In MGEScan source code, ltr/toGFF.py and nonltr/toGFF.py are used to convert results to GFF format developed by Wazim Mohammmed Ismail.

CHAPTER 11

Test Results (New)

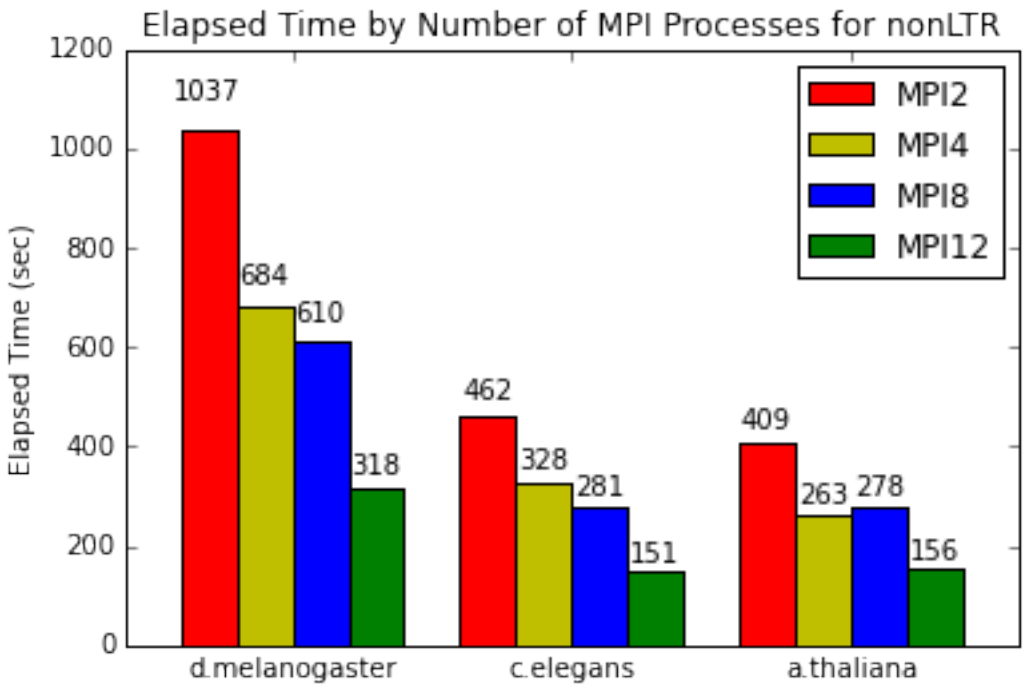
Page Contents

- *D. melanogaster (dm3)*
  - *Evaluation*
- *C. elegans*
- *A. thaliana*

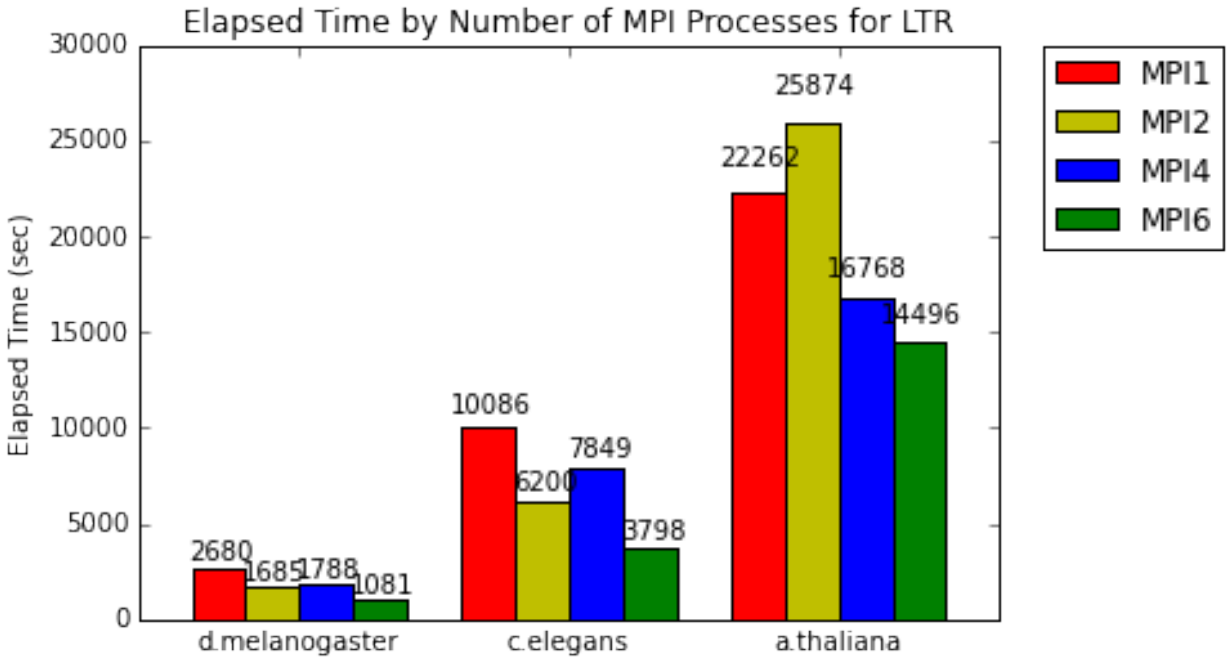
Three genomes were tested with MGEScan-LTR and MGEScan-nonLTR programs.

- **Test genome sequences:**
  - 4. melanogaster (dm3): fruitfly.org
  - 3. elegans: wormbase.org
  - 1. thaliana: nih.gov
- **Test Environment:** chameleoncloud.org
- **Hardware Spec:**
  - Intel Xeon E5-2670 v3 "Haswell" processors (each with 12 cores @ 2.3GHz)
  - 48 vCPUs
  - 128 GiB
- **Operating System:**
  - Ubuntu 14.04 LTS

Performance nonLTR with MPI



Performance LTR with MPI



ipynb file

11.1 D. melanogaster (dm3)

- results (gff3): https://github.com/MGEScan/evaluation/tree/accuracy-evaluation/results/dm3

11.1.1 Evaluation

Table 11.1: Elapsed time for MGEScan-nonLTR (dm3)

Elapsed Time	Options
5 mins (318 secs)	12 MPI Processes
11 mins (610 secs)	8 MPI Processes
12 mins (684 secs)	4 MPI Processes
18 mins (1037 secs)	2 MPI Processes

Table 11.2: Elapsed time for MGEScan-LTR (dm3)

Elapsed Time	Options
18 mins (1081 secs)	6 MPI Processes
30 mins (1788 secs)	4 MPI Processes
28 mins (1685 secs)	2 MPI Processes
45 mins (2680 secs)	1 MPI Process

11.2 C. elegans

- results (gff3): https://github.com/MGEScan/evaluation/tree/accuracy-evaluation/results/c_elegans

11.3 A. thaliana

- results (gff3): https://github.com/MGEScan/evaluation/tree/accuracy-evaluation/results/a_thaliana

CHAPTER 12

Test Results

Page Contents

- *D. melanogaster (dm3)*
  - *Evaluation*
  - *Extra Files*
- *C. intestinalis (KH)*
  - *Evaluation*
  - *Extra Files*
  - *Extra Files*
- *D. pulex (GCA_000187875.1)*
  - *Evaluation*
  - *Extra Files*

Four sample genomes were tested with MGEScan-LTR and MGEScan-nonLTR programs.

- **Test genome sequences:**
  - 4. melanogaster (dm3): ucsc
  - 3. intestinalis (KH): Ensembl
- **Test Environment:** Cloud instances of FutureSystems at Indiana University (http://futuresystems.org).
- **Hardware Spec:**
  - Intel Xeon X5550 2.66GHz
  - 8 vCPUs
  - 16 GB DDR3 1333 MHz
  - 160GB 7200RPM SATA
- **Operating System:**

- Ubuntu 14.04 LTS

Test Genome Sequences

Dataset	Ver. of MGEScan	Elapsed Time (hr:min:sec) [Speed up]		
		Total	Non-LTR	LTR
d. melanogaster	1	3:40:20	0:55:20	2:45:00
	2	2:35:04 [1.42]	0:19:30 [2.84]	2:35:04 [1.06]
	2 (MPI with 4 CPUs)	1:48:22 [2.03]	0:15:29 [3.57]	1:48:22 [1.52]
d. pulex	1	4:05:57	1:08:47	2:57:10
	2	2:36:54 [1.56]	0:46:20 [1.48]	2:36:54 [1.12]
	2 (MPI with 4 CPUs)	1:03:43 [3.84]	0:14:38 [4.7]	1:03:43 [2.76]
c. intestinalis	1	5:18:36	0:34:47	4:43:49
	2	4:05:27 [1.29]	0:09:23 [4.03]	4:05:27 [1.16]
	2 (MPI with 4 CPUs)	1:22:37 [3.89]	0:03:02 [11.14]	1:22:37 [3.48]
Ave. Speed Up	2	1.43	2.78	1.11
	2 (MPI with 4 CPUS)	3.26	6.47	2.59

12.1 D. melanogaster (dm3)

- `dm3.gff3`
- `dm3.ltr.out`
- `dm3.en`
- `dm3.rt`

12.1.1 Evaluation

Table 12.1: Elapsed time for MGEScan (dm3)

Program	Total	nonLTR	LTR	Options
MGEScan1.3.1	3 hrs 40 mins (13,220 secs)	55 mins (3,320 secs)	2 hrs 45 mins (9,900 secs)	HMMER2, no MPI
MGEScan2	2 hrs 35 mins (9,304 secs)	19 mins (1,170 secs)	2 hrs 35 mins (9,304 secs)	HMMER3.1b1, no MPI
MGEScan2 with MPI	1 hr 48 mins (6,502 secs)	15 mins (929 secs)	1 hr 48 mins (6,502 secs)	HMMER3.1b1, MPI with 4 processors

12.1.2 Extra Files

- `dm3.tar.gz` (Compressed file)

12.2 C. intestinalis (KH)

- `KH.gff3`
- `KH.ltr.out`
- `KH.en`
- `KH.rt`

12.2.1 Evaluation

Table 12.2: Elapsed time for C. intestinalis

Program	Total	nonLTR	LTR	Options
MGES-can1.3.1	5 hours 18 minutes 36 seconds	34 minutes 47 seconds	4 hours 43 minutes 49 seconds	HMMER 2.3.2, no MPI
MGEScan2	4 hours 5 minutes 27 seconds	9 minutes 23 seconds	4 hours 5 minutes 27 seconds	HMMER 3.1b1, no MPI
MGEScan2 with MPI	1 hour 22 minutes 37 seconds	3 minutes 2 seconds	1 hour 22 minutes 37 seconds	HMMER 3.1b1, MPI with 4 processors

12.2.2 Extra Files

- `KH.tar.gz`

12.2.3 Extra Files

- `strPur2.tar.gz`

12.3 D. pulex (GCA_000187875.1)

- `dpulex.gff3`
- `dpulex.ltr.out`
- `dpulex.en`
- `dpulex.rt`

12.3.1 Evaluation

Table 12.3: Elapsed time for MGEScan (dpulex)

Program	Total	nonLTR	LTR	Options
MGEScan1.3.1	4 hrs 5mins (14,697 secs)	1hr 8mins (4,127 secs)	2 hrs 57 mins (10,570 secs)	HMMER 2.3.2, no MPI
MGEScan2	2 hrs 36 mins (9,414 secs)	46 mins (2,780 secs)	2 hrs 36 mins (9,414 secs)	HMMER 3.1b1, no MPI
MGEScan2 with MPI	1hr 3mins (3,823 secs)	15 mins (878 secs)	1 hr 3mins (3,823 secs)	HMMER 3.1b1, MPI with 4 processors

12.3.2 Extra Files

- `dpulex.tar.gz`

CHAPTER 13

Test Results with Previous MGEScan 1.3.1

- 4. melanogaster: `dmelanogaster.old.tar.gz`
- 4. pulex: `dpulex.old.tar.gz`
- 3. intestinalis: `KH.old.tar.gz`
- 19. purpuratus: `strPur2.old.tar.gz`

CHAPTER 14

Source code

Source code is available at https://github.com/mgescan/mgescan