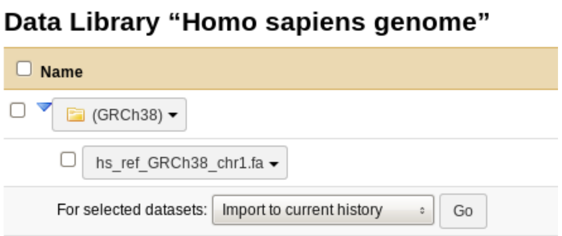
REFERENCE DATA IN GALAXY

Reference data

Fasta file


Data Library “Homo sapiens genome”

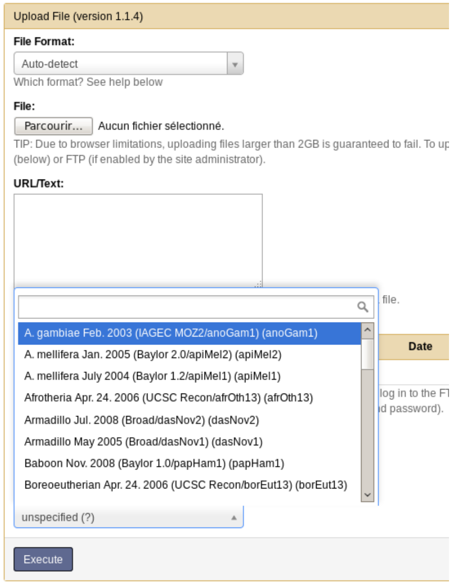
□ Name
□ (GRCh38)
□ hs_ref_GRCh38_chr1.fa

For selected datasets: Import to current history — Go

Build names (e.g. hg19)

Indexes
(len, blast, bowtie, ...)

Data upload


Upload File (version 1.1.4)

File Format:
Auto-detect
Which format? See help below

File:
Parcourir... Aucun fichier sélectionné.
TIP: Due to browser limitations, uploading files larger than 2GB is guaranteed to fail. To up
(below) or FTP (if enabled by the site administrator).

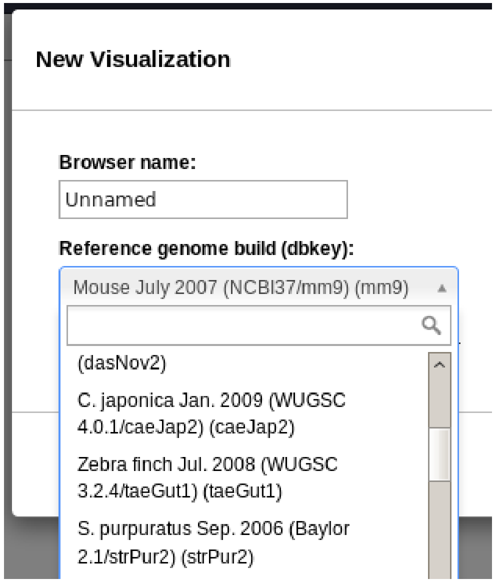
URL/Text:

A. gambiae Feb. 2003 (IAGEC MOZ2/anoGam1) (anoGam1)
A. mellifera Jan. 2005 (Baylor 2.0/apiMel2) (apiMel2)
A. mellifera July 2004 (Baylor 1.2/apiMel1) (apiMel1)
Afrotheria Apr. 24. 2006 (UCSC Recon/afrOth13) (afrOth13)
Armadillo Jul. 2008 (Broad/dasNov2) (dasNov2)
Armadillo May 2005 (Broad/dasNov1) (dasNov1)
Baboon Nov. 2008 (Baylor 1.0/papHam1) (papHam1)
Boreoeutherian Apr. 24. 2006 (UCSC Recon/borEut13) (borEut13)

unspecified (?)

Execute

Visualizations

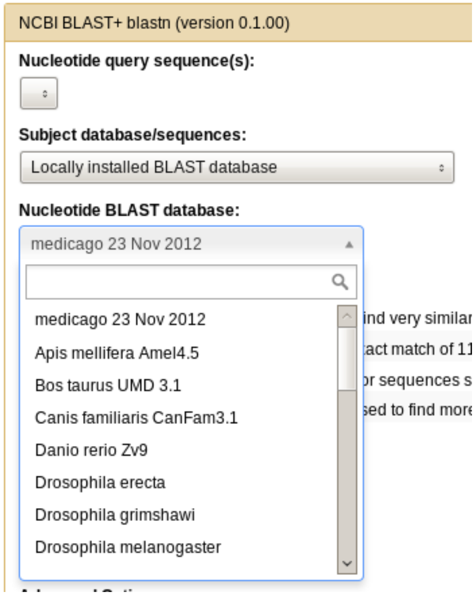

New Visualization

Browser name:
Unnamed

Reference genome build (dbkey):
Mouse July 2007 (NCBI37/mm9) (mm9)

(dasNov2)
C. japonica Jan. 2009 (WUGSC 4.0.1/caeJap2) (caeJap2)
Zebra finch Jul. 2008 (WUGSC 3.2.4/taeGut1) (taeGut1)
S. purpuratus Sep. 2006 (Baylor 2.1/strPur2) (strPur2)

Tools


NCBI BLAST+ blastn (version 0.1.00)

Nucleotide query sequence(s):

Subject database/sequences:
Locally installed BLAST database

Nucleotide BLAST database:
medicago 23 Nov 2012

medicago 23 Nov 2012
Apis mellifera Amel4.5
Bos taurus UMD 3.1
Canis familiaris CanFam3.1
Danio rerio Zv9
Drosophila erecta
Drosophila grimshawi
Drosophila melanogaster

Reference data: loc files

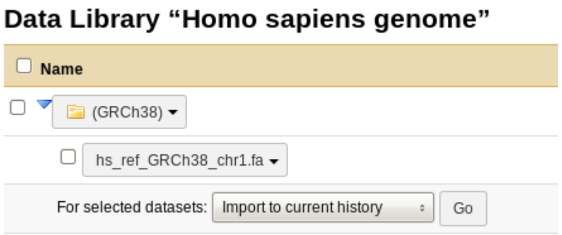
- The good old way
- *.loc files in tool-data dir

```
nr_05Jun2010 NCBI NR (non redundant) 05 Jun 2010       /data/blastdb/05Jun2010/nr
nr_15Aug2010 NCBI NR (non redundant) 15 Aug 2010       /data/blastdb/15Aug2010/nr
```

- Pros:
  - Simple and it works
- Cons:
  - Need to restart galaxy
  - Manual intervention (error-prone, easy to forget)
  - Need to generate indexes manually

Reference data: data libraries

- Management from the admin interface
- Pros:
  - Permissions support
- Cons:
  - Pregenerate indexes manually or let user do it
  - Import to user history: tools need to support it (indexes)
  - Visualization: need to create “custom builds”
  - Manual intervention (less error-prone, easy to forget)



Data Library “Homo sapiens genome”

☐ Name

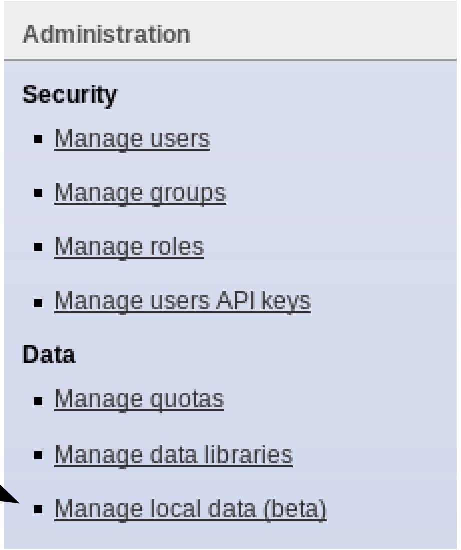
☐ (GRCh38)

☐ hs_ref_GRCh38_chr1.fa

For selected datasets: Import to current history

Reference data: data tables

- New: “data managers”!
- Type of tool, only accessible to admin
  - Download/index files
  - Fill “tool data tables” (~ *.loc files)
  - Available in the toolshed
- Web UI or API



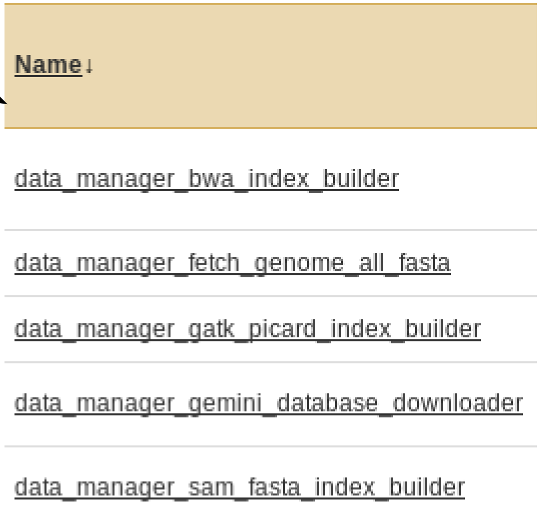
Administration

Security

- Manage users
- Manage groups
- Manage roles
- Manage users API keys

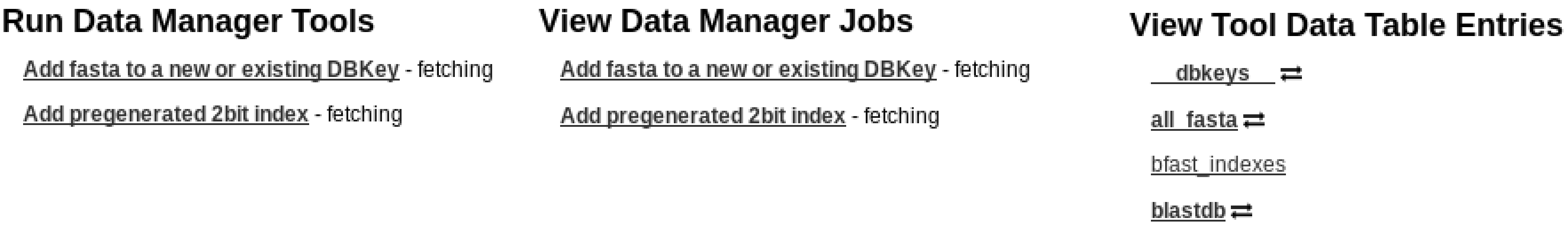
Data

- Manage quotas
- Manage data libraries
- Manage local data (beta)



Name↓
data_manager_bwa_index_builder
data_manager_fetch_genome_all_fasta
data_manager_gatk_picard_index_builder
data_manager_gemini_database_downloader
data_manager_sam_fasta_index_builder

Reference data: data tables



Run Data Manager Tools

Add fasta to a new or existing DBKey - fetching

Add pregenerated 2bit index - fetching

View Data Manager Jobs

Add fasta to a new or existing DBKey - fetching

Add pregenerated 2bit index - fetching

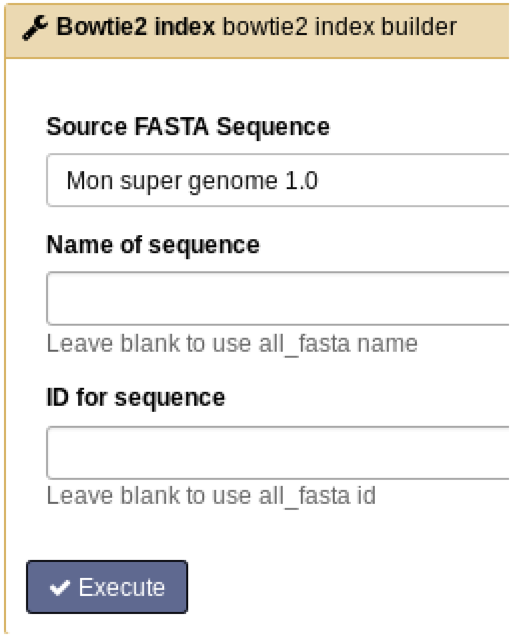
View Tool Data Table Entries

dbkeys ⇄

all_fasta ⇄

bfast_indexes

blastdb ⇄



Bowtie2 index bowtie2 index builder

Source FASTA Sequence

Mon super genome 1.0

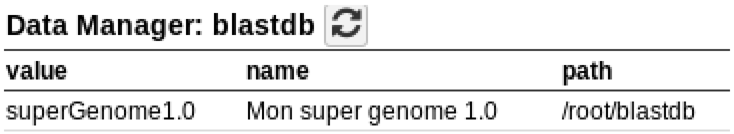
Name of sequence

Leave blank to use all_fasta name

ID for sequence

Leave blank to use all_fasta id

✔ Execute



Data Manager: blastdb

value	name	path
superGenome1.0	Mon super genome 1.0	/root/blastdb

- Pros:
  - Much less error-prone
- Cons:
  - Manual: still easy to forget…
  - Galaxy-centric (files not available outside)

USING IT

Scenarii

- **Using a loc file**
  - **just a db, not a genome**
- Configuring genomes with loc files
  - More than just a db (upload, visualization)
- Using data tables
- Configuring data tables, data managers

Using a loc file ncbi_blastp_wrapper.xml

```
<conditional name="db_opts">
    <param name="db_opts_selector" type="select" label="Subject database/sequences">
      <option value="db" selected="True">BLAST Database</option>
      <option value="file">FASTA file</option>
    </param>
    <when value="db">
        <param name="database" type="select" label="Protein BLAST database">
            <options from_file="blastdb_p.loc">
              <column name="value" index="0"/>
              <column name="name" index="1"/>
              <column name="path" index="2"/>
            </options>
        </param>
        <param name="subject" type="hidden" value="" />
    </when>
    <when value="file">
        <param name="database" type="hidden" value="" />
        <param name="subject" type="data" format="fasta" label="Protein FASTA file
to use as database"/>
    </when>
</conditional>
```

Using a loc file tool-data/blastdb_p.loc

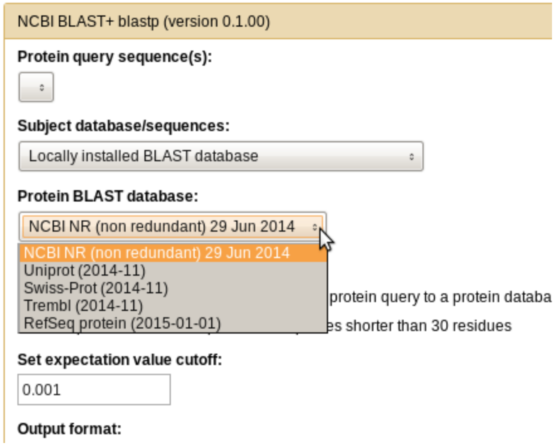
```
#This is a sample file distributed with Galaxy that is used to define a
#list of protein BLAST databases, using three columns tab separated
#(longer whitespace are TAB characters):
#
#<unique_id>    <database_caption> <base_name_path>
#
# [...]
nr_29Jun2014    NCBI NR (non redundant) 29 Jun 2014     /db/nr/NR_2014-06-29/blast/All/nr

uniprot Uniprot (2014-11)     /db/uniprot/UniProt_2014_11/blast/All/uniprot

swissprot Swiss-Prot (2014-11)    /db/uniprot/UniProt_2014_11/blast/Swiss-Prot/uniprot_sprot

trembl  Trembl (2014-11)        /db/uniprot/UniProt_2014_11/blast/TrEMBL/uniprot_trembl

refseq_protein  RefSeq protein (2015-01-01)     /db/refseq_protein/RefSeq_protein_2015-01-01/blast/All/refseq_protein
```



NCBI BLAST+ blastp (version 0.1.00)

Protein query sequence(s):

Subject database/sequences: Locally installed BLAST database

Protein BLAST database: NCBI NR (non redundant) 29 Jun 2014

- NCBI NR (non redundant) 29 Jun 2014
- Uniprot (2014-11)
- Swiss-Prot (2014-11)
- Trembl (2014-11)
- RefSeq protein (2015-01-01)

protein query to a protein databa

es shorter than 30 residues

Set expectation value cutoff: 0.001

Output format:

Scenarii

- Using a loc file
  - just a db, not a genome
- **Configuring genomes with loc files**
  - **More than just a db (upload, visualization)**
- Using data tables
- Configuring data tables, data managers

Configuring genomes with loc files

- config/galaxy.ini

```
# File containing old-style genome builds
#builds_file_path = tool-data/shared/ucsc/builds.txt
```

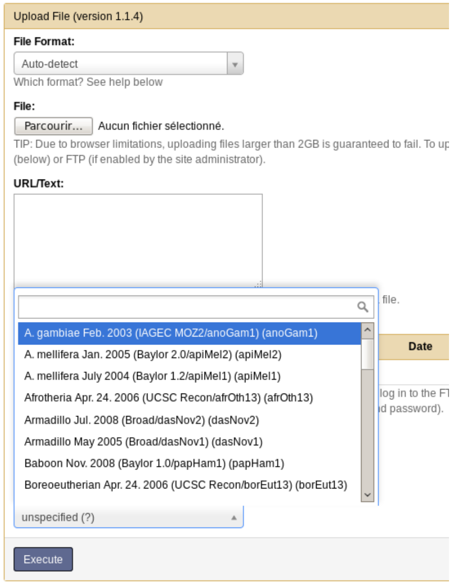
- builds.txt

```
#Harvested from http://genome-test.cse.ucsc.edu/cgi-bin/das/dsn
?   unspecified (?)
hg19Haps        hg19Haplotypes Feb. 2009 (GRCh37/hg19Haps) (hg19Haps)
hg19    Human Feb. 2009 (GRCh37/hg19) (hg19)
hg18    Human Mar. 2006 (NCBI36/hg18) (hg18)
hg17    Human May 2004 (NCBI35/hg17) (hg17)
hg16    Human July 2003 (NCBI34/hg16) (hg16)
hg15    Human Apr. 2003 (NCBI33/hg15) (hg15)
venter1 J. Craig Venter Sep. 2007 (HuRef/venter1) (venter1)
panTro2 Chimp Mar. 2006 (CGSC 2.1/panTro2) (panTro2)
panTro1 Chimp Nov. 2003 (CGSC 1.1/panTro1) (panTro1)
gorGor2 Gorilla Aug. 2009 (Sanger 4/gorGor2) (gorGor2)
gorGor1 Gorilla Oct. 2008 (Sanger 0.1/gorGor1) (gorGor1)
ponAbe2 Orangutan July 2007 (WUGSC 2.0.2/ponAbe2) (ponAbe2)
rheMac2 Rhesus Jan. 2006 (MGSC Merged 1.0/rheMac2) (rheMac2)
```

Configuring genomes with loc files

- Len file: length of each chrom (for visualization)
- tool-data/shared/ucsc/chrom/hg19.len

```
chr1    249250621
chr2    243199373
chr3    198022430
chr4    191154276
chr5    180915260
chr6    171115067
chr7    159138663
chrX    155270560
chr8    146364022
chr9    141213431
chr10   135534747
chr11   135006516
chr12   133851895
chr13   115169878
chr14   107349540
chr15   102531392
```



Upload File (version 1.1.4)

File Format:

Auto-detect

Which format? See help below

File:

Parcourir... Aucun fichier sélectionné.

TIP: Due to browser limitations, uploading files larger than 2GB is guaranteed to fail. To up
(below) or FTP (if enabled by the site administrator).

URL/Text:

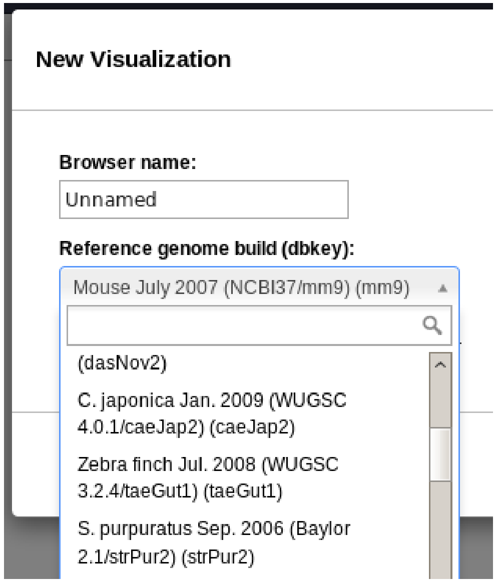
A. gambiae Feb. 2003 (IAGEC MOZ2/anoGam1) (anoGam1)
A. mellifera Jan. 2005 (Baylor 2.0/apiMel2) (apiMel2)
A. mellifera July 2004 (Baylor 1.2/apiMel1) (apiMel1)
Afrotheria Apr. 24. 2006 (UCSC Recon/afrOth13) (afrOth13)
Armadillo Jul. 2008 (Broad/dasNov2) (dasNov2)
Armadillo May 2005 (Broad/dasNov1) (dasNov1)
Baboon Nov. 2008 (Baylor 1.0/papHam1) (papHam1)
Boreoeutherian Apr. 24. 2006 (UCSC Recon/borEut13) (borEut13)

unspecified (?)

file.

Date

log in to the F
d password).



New Visualization

Browser name: Unnamed

Reference genome build (dbkey): Mouse July 2007 (NCBI37/mm9) (mm9)

(dasNov2)
C. japonica Jan. 2009 (WUGSC 4.0.1/caeJap2) (caeJap2)
Zebra finch Jul. 2008 (WUGSC 3.2.4/taeGut1) (taeGut1)
S. purpuratus Sep. 2006 (Baylor 2.1/strPur2) (strPur2)

Configuring genomes with loc files

- Build ids used in other loc files
- Result of mapping associated with selected genome + visu
- e.g. tool-data/bowtie2_indices.loc:

```
#This is a sample file distributed with Galaxy that enables tools
#to use a directory of Bowtie2 indexed sequences data files. You will
#need to create these data files and then create a bowtie_indices.loc
#file similar to this one (store it in this directory) that points to
#the directories in which those files are stored. The bowtie2_indices.loc
#file has this format (longer white space characters are TAB characters):
#
#<unique_build_id>   <dbkey>   <display_name>   <file_base_path>
#
#So, for example, if you had hg18 indexed stored in
#/depot/data2/galaxy/bowtie2/hg18/,
#then the bowtie2_indices.loc entry would look like this:
#
hg18   hg18    hg18    /depot/data2/galaxy/bowtie2/hg18/hg18
```

Scenarii

- Using a loc file
  - just a db, not a genome
- Configuring genomes with loc files
  - More than just a db (upload, visualization)
- **Using data tables**
- Configuring data tables, data managers

Using data tables example: BWA

```
    <conditional name="genomeSource">
      <param name="refGenomeSource" type="select" label="Will you select a reference
genome from your history or use a built-in index?">
        <option value="indexed">Use a built-in index</option>
        <option value="history">Use one from the history</option>
      </param>
      <when value="indexed">
        <param name="indices" type="select" label="Select a reference genome">
          <options from_data_table="bwa_indexes">
            <filter type="sort_by" column="2" />
            <validator type="no_options" message="No indexes are available" />
          </options>
        </param>
      </when>
      <when value="history">
        <param name="ownFile" type="data" format="fasta" metadata_name="dbkey"
label="Select a reference from history" />
      </when>
    </conditional>
```

Scenarii

- Using a loc file
  - just a db, not a genome
- Configuring genomes with loc files
  - More than just a db (upload, visualization)
- Using data tables
- **Configuring data tables**, data managers

Configuring data tables config/tool_data_table_conf.xml

- Load old-style *.loc files in corresponding data tables

```
<tables>
    <!-- Locations of all fasta files under genome directory -->
    <table name="all_fasta" comment_char="#">
        <columns>value, dbkey, name, path</columns>
        <file path="tool-data/all_fasta.loc" />
    </table>
    <!-- Locations of indexes in the BFAST mapper format -->
    <table name="bfast_indexes" comment_char="#">
        <columns>value, dbkey, formats, name, path</columns>
        <file path="tool-data/bfast_indexes.loc" />
    </table>
    <!-- Locations of nucleotide (mega)blast databases -->
    <table name="blastdb" comment_char="#">
        <columns>value, name, path</columns>
        <file path="tool-data/blastdb.loc" />
    </table>
[...]
```

Configuring data tables: adding entries

- Modify loc files, then restart/reload
- Use data managers
  - Available in toolshed.genouest.org or official test toolshed
- TP
  - Install data_manager_fasta_dbkeys (genouest)
  - Add a test fasta file
  - Create a simple tool (head, cat…) that reads the all_fasta data table

Scenarii

- Using a loc file
  - just a db, not a genome
- Configuring genomes with loc files
  - More than just a db (upload, visualization)
- Using data tables
- Configuring data tables, **data managers**

Creating a data manager

- my_data_manager/

  |--data_manager/

  |--my_wrapper.py

  |--my_wrapper.xml

  |--tool-data/

  |--example.loc.sample

  |--tool_data_table_conf.xml.sample

  |--data_manager_conf.xml

```
<?xml version="1.0"?>
<data_managers>
    <data_manager
tool_file="data_manager/my_wrapper.xml"
id="add_example">
        <data_table name="example_indexes">
            <output>
                <column name="value" />
                <column name="dbkey" />
                <column name="name" />
                <column name="path" />
            </output>
        </data_table>
    </data_manager>
</data_managers>
```

- Describe output of the wrapper

Creating a data manager

- my_data_manager/

```
 |--data_manager/
   |--my_wrapper.py
   |--my_wrapper.xml
 |--tool-data/
   |--example.loc.sample
   |--tool_data_table_conf.xml.sample
 |--data_manager_conf.xml
```

```
<tables>
    <table name="bwa_indexes" comment_char="#">
        <columns>value, dbkey, name, path</columns>
        <file path="tool-data/bwa_index.loc" />
    </table>
</tables>
```

- Added automatically in shed_tool_data_table_conf.xml during installation from toolshed (loc file path modified)

Creating a data manager

```
#This is a sample file distributed with Galaxy that enables tools
#to use a directory of BWA indexed sequences data files. You will need
#to create these data files and then create a bwa_index.loc file
#similar to this one (store it in this directory) that points to
#the directories in which those files are stored. The bwa_index.loc
#file has this format (longer white space characters are TAB characters
#
#<unique_build_id>   <dbkey>   <display_name>   <file_path>
#
#So, for example, if you had phiX indexed stored in
#/depot/data2/galaxy/phiX/base/,
#then the bwa_index.loc entry would look like this:
#
#phiX174   phiX   phiX Pretty   /depot/data2/galaxy/phiX/base/phiX.fa
#
```

- my_data_manager/

  |--data_manager/

    |--my_wrapper.py

    |--my_wrapper.xml

  |--tool-data/

    |--**example.loc.sample**

    |--tool_data_table_conf.xml.sample

Creating a data manager

- my_data_manager/

|--data_manager/

|--my_wrapper.py

|--**my_wrapper.xml**

|--tool-data/

|--example.loc.sample

|--tool_data_table_conf.xml.sample

|--data_manager_conf.xml

```
<tool id="my_data_manager" name="Add pregenerated XXX
index" version="0.0.1" tool_type="manage_data">
    <description>fetching</description>
    <command interpreter="python">my_wrapper.py "$
{out_file}"</command>
    <inputs>

        <param name="dbkey" type="select" label="DBKey">
            <options from_data_table="__dbkeys__"/>
        </param>
        <param name="i_n" type="text" label="Index Name"/>
        <param name="i_id" type="text" label="Index ID"/>
        <param type="text" name="i_p" label="Path" />
    </inputs>
    <outputs>
        <data name="out_file" format="data_manager_json"/>
    </outputs>
</tool>
```

Creating a data manager

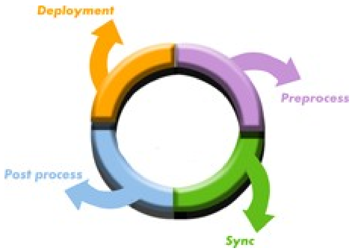
- my_data_manager/

```
|--data_manager/
   |--my_wrapper.py
   |--my_wrapper.xml
|--tool-data/
   |--example.loc.sample
   |--tool_data_table_conf.xml.sample
|--data_manager_conf.xml
```

  - Generate data table content in JSON format
  - Use existing python script as template

BIOMAJ

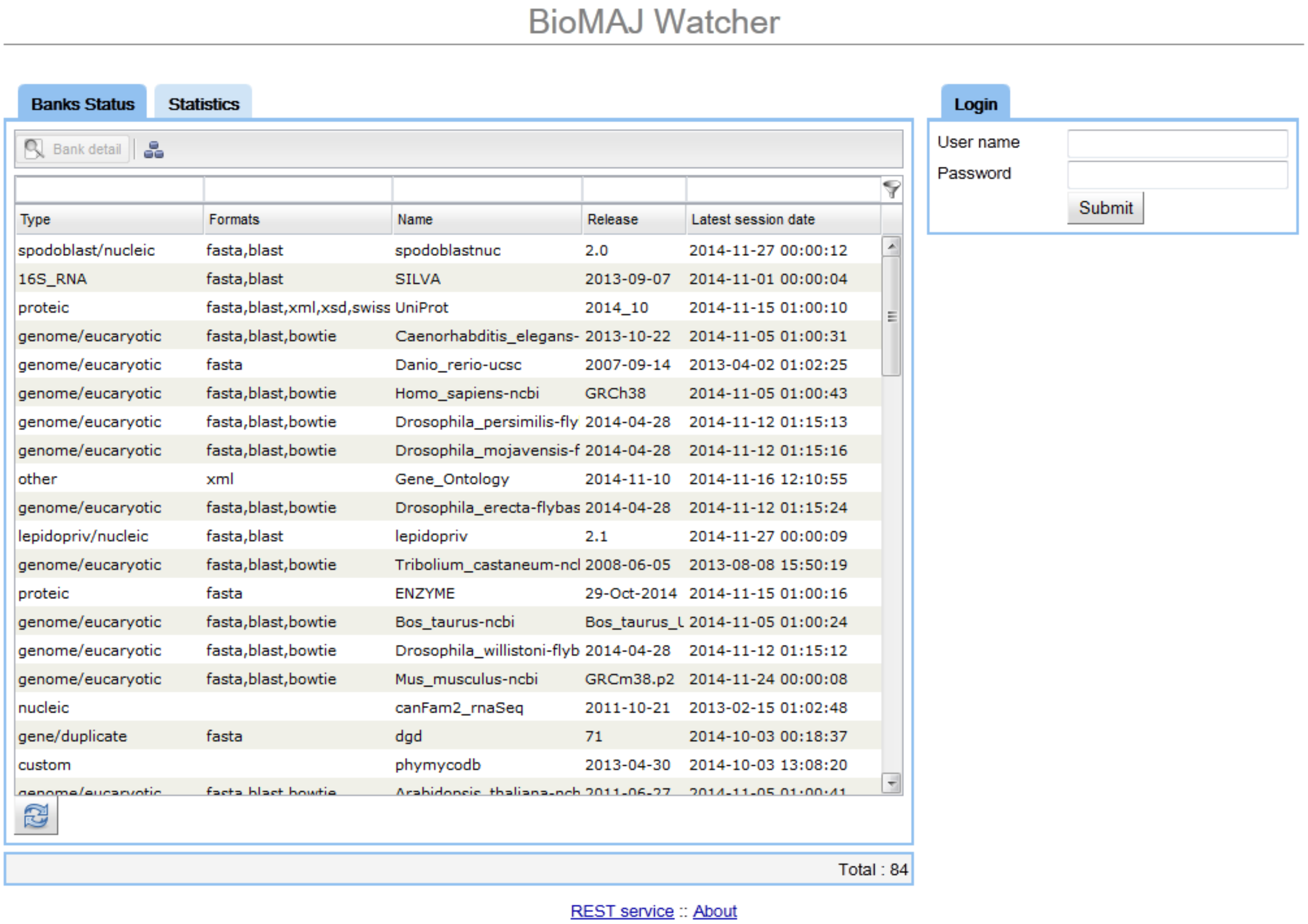
BioMAJ



- Workflow engine for data synchronization and processing
  - Find new data releases
  - Download files (ftp, http, …)
  - Process data (indexes, format conversions, twitter, …)
- Scheduling
- Web UI, REST API
- Widely used
  - French bioinfo platforms, debian/rpm packages, …
- New version coming! (complete rewrite in python)

BioMAJ

http://biomaj.genouest.org



BioMAJ Watcher

User name

Password

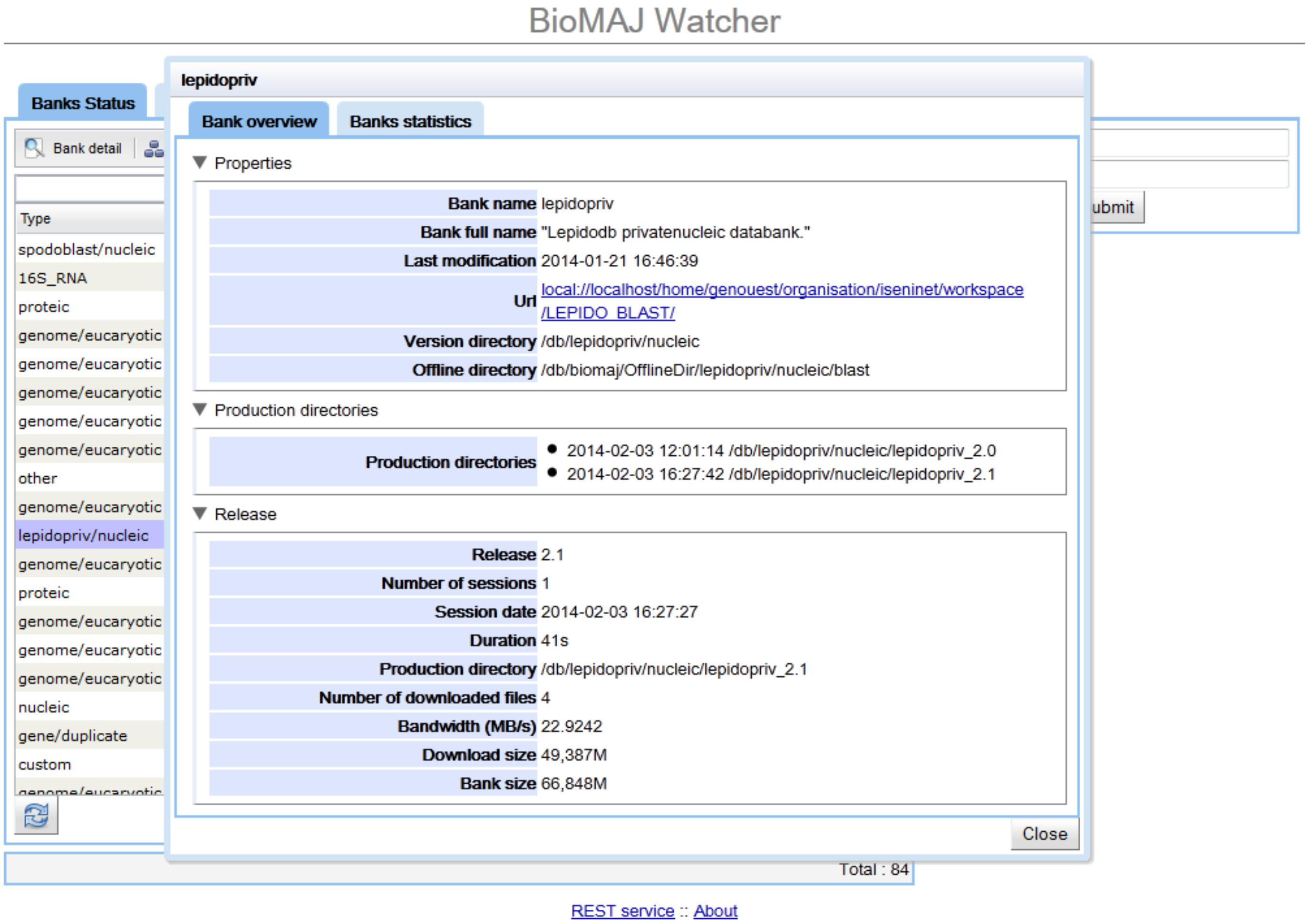
Type	Formats	Name	Release	Latest session date
spodoblast/nucleic	fasta,blast	spodoblastnuc	2.0	2014-11-27 00:00:12
16S_RNA	fasta,blast	SILVA	2013-09-07	2014-11-01 00:00:04
proteic	fasta,blast,xml,xsd,swiss	UniProt	2014_10	2014-11-15 01:00:10
genome/eucaryotic	fasta,blast,bowtie	Caenorhabditis_elegans-	2013-10-22	2014-11-05 01:00:31
genome/eucaryotic	fasta	Danio_rerio-ucsc	2007-09-14	2013-04-02 01:02:25
genome/eucaryotic	fasta,blast,bowtie	Homo_sapiens-ncbi	GRCh38	2014-11-05 01:00:43
genome/eucaryotic	fasta,blast,bowtie	Drosophila_persimilis-fly	2014-04-28	2014-11-12 01:15:13
genome/eucaryotic	fasta,blast,bowtie	Drosophila_mojavensis-f	2014-04-28	2014-11-12 01:15:16
other	xml	Gene_Ontology	2014-11-10	2014-11-16 12:10:55
genome/eucaryotic	fasta,blast,bowtie	Drosophila_erecta-flybas	2014-04-28	2014-11-12 01:15:24
lepidopriv/nucleic	fasta,blast	lepidopriv	2.1	2014-11-27 00:00:09
genome/eucaryotic	fasta,blast,bowtie	Tribolium_castaneum-ncl	2008-06-05	2013-08-08 15:50:19
proteic	fasta	ENZYME	29-Oct-2014	2014-11-15 01:00:16
genome/eucaryotic	fasta,blast,bowtie	Bos_taurus-ncbi	Bos_taurus_U	2014-11-05 01:00:24
genome/eucaryotic	fasta,blast,bowtie	Drosophila_willistoni-flyb	2014-04-28	2014-11-12 01:15:12
genome/eucaryotic	fasta,blast,bowtie	Mus_musculus-ncbi	GRCm38.p2	2014-11-24 00:00:08
nucleic		canFam2_rnaSeq	2011-10-21	2013-02-15 01:02:48
gene/duplicate	fasta	dgd	71	2014-10-03 00:18:37
custom		phymycodb	2013-04-30	2014-10-03 13:08:20
genome/eucaryotic	fasta,blast,bowtie	Arabidopsis_thaliana-ncb	2011-06-27	2014-11-05 01:00:41

Total : 84

REST service :: About

BioMAJ

http://biomaj.genouest.org



BioMAJ Watcher

Type
spodoblast/nucleic
16S_RNA
proteic
genome/eucaryotic
genome/eucaryotic
genome/eucaryotic
genome/eucaryotic
genome/eucaryotic
other
genome/eucaryotic
lepidopriv/nucleic
genome/eucaryotic
proteic
genome/eucaryotic
genome/eucaryotic
genome/eucaryotic
nucleic
gene/duplicate
custom
genome/eucaryotic

Total : 84

lepidopriv

Bank overview | Banks statistics

▼ Properties

Bank name	lepidopriv
Bank full name	"Lepidodb privatenucleic databank."
Last modification	2014-01-21 16:46:39
Url	local://localhost/home/genouest/organisation/iseninet/workspace/LEPIDO_BLAST/
Version directory	/db/lepidopriv/nucleic
Offline directory	/db/biomaj/OfflineDir/lepidopriv/nucleic/blast

▼ Production directories

Production directories	● 2014-02-03 12:01:14 /db/lepidopriv/nucleic/lepidopriv_2.0 ● 2014-02-03 16:27:42 /db/lepidopriv/nucleic/lepidopriv_2.1

▼ Release

Release	2.1
Number of sessions	1
Session date	2014-02-03 16:27:27
Duration	41s
Production directory	/db/lepidopriv/nucleic/lepidopriv_2.1
Number of downloaded files	4
Bandwidth (MB/s)	22.9242
Download size	49,387M
Bank size	66,848M

BIOMAJ ❤ GALAXY

BioMAJ ❤ Galaxy

- Let BioMAJ and Galaxy be friends!
- BioMAJ post process to update reference data in Galaxy
- It brings:
  - Automatization: scheduled updates
  - Reliability: no more dead entries in loc files
  - Data reuse: each index is generated and stored 1 time and can be used from command line, galaxy, mobyle, …

The Project

- BioMAJ “post processes” to inject reference data in Galaxy
    - Using data managers
    - Or data libraries (permissions support)
- BioMAJ “remove processes” to remove old reference data
- Supported : fasta, blast, bowtie(2), bwa, 2bit
    - Easy to extend to other formats

What you need

- Up-to-date Galaxy instance (tested with galaxy-central)
- Galaxy patch to allow removal from data tables: PR #577
- Install some data managers
    - http://toolshed.genouest.org
- BioMAJ processes
    - Python scripts
    - https://github.com/genouest/biomaj2galaxy
- Configure BioMAJ databank: config file or web UI

Example: human genome (NCBI ftp)

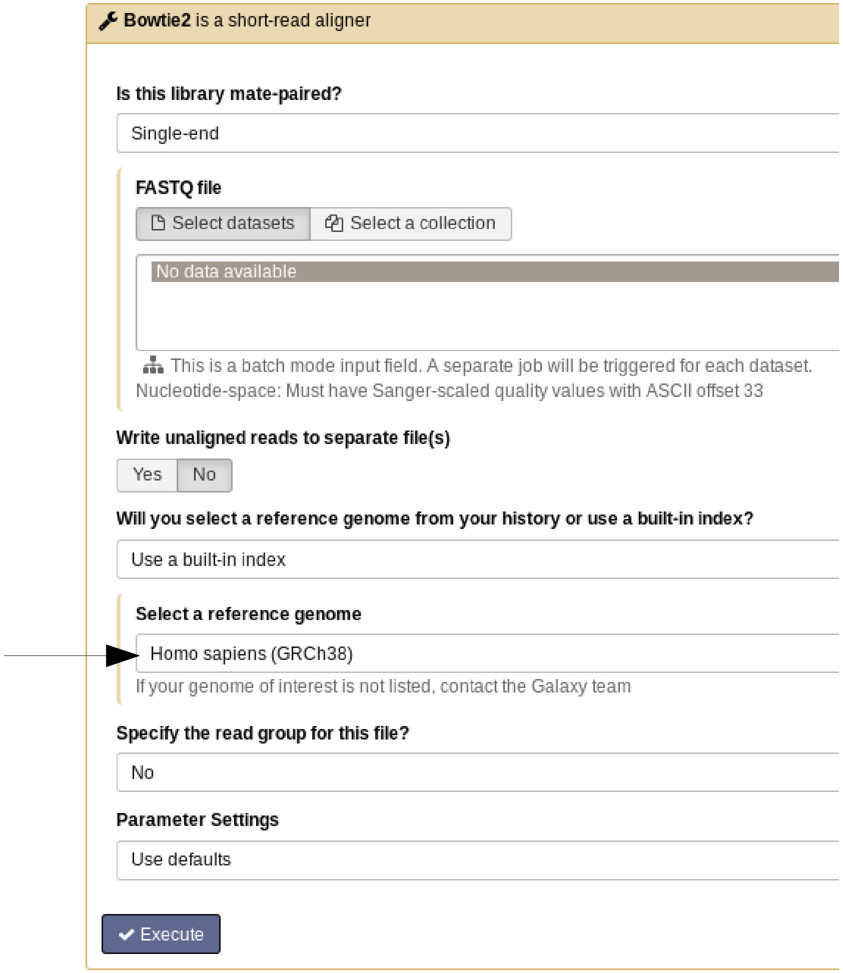
```
B2.db.post.process=GALAXY
GALAXY=galaxy_dm

galaxy_dm.name=galaxy_dm
galaxy_dm.desc=Add files to Galaxy tool data tables
galaxy_dm.type=galaxy
galaxy_dm.exe=add_galaxy_data_manager.py
galaxy_dm.args=-u http://example.org/galaxy/ -k my_api_key -d "${remote.release}"
-n "Homo sapiens (${remote.release})" -g ${data.dir}/${dir.version}/${db.name}_$
{remote.release}/fasta/all.fa --bowtie2 ${datadir}/${dir.version}/${db.name}_$
{remote.release}/bowtie/all --blastn ${data.dir}/${dir.version}/${db.name}_$
{remote.release}/blast/Homo_sapiens-ncbi_testing

db.remove.process=RM_GALAXY
RM_GALAXY=rm_galaxy_dm

rm_galaxy_dm.name=rm_galaxy_dm
rm_galaxy_dm.desc=Remove from Galaxy tool data tables
rm_galaxy_dm.type=galaxy
rm_galaxy_dm.exe=remove_galaxy_data_manager.py
rm_galaxy_dm.args=-u http://example.org/galaxy/ -k my_api_key -d "$
{remote.release}" -f --blastn --bowtie2 --delete
```

Example: human genome (NCBI ftp)



Bowtie2 is a short-read aligner

Is this library mate-paired?

Single-end

FASTQ file

No data available

This is a batch mode input field. A separate job will be triggered for each dataset.

Nucleotide-space: Must have Sanger-scaled quality values with ASCII offset 33

Write unaligned reads to separate file(s)

Yes [✓ No]

Will you select a reference genome from your history or use a built-in index?

Use a built-in index

Select a reference genome

Homo sapiens (GRCh38)

If your genome of interest is not listed, contact the Galaxy team

Specify the read group for this file?

No

Parameter Settings

Use defaults