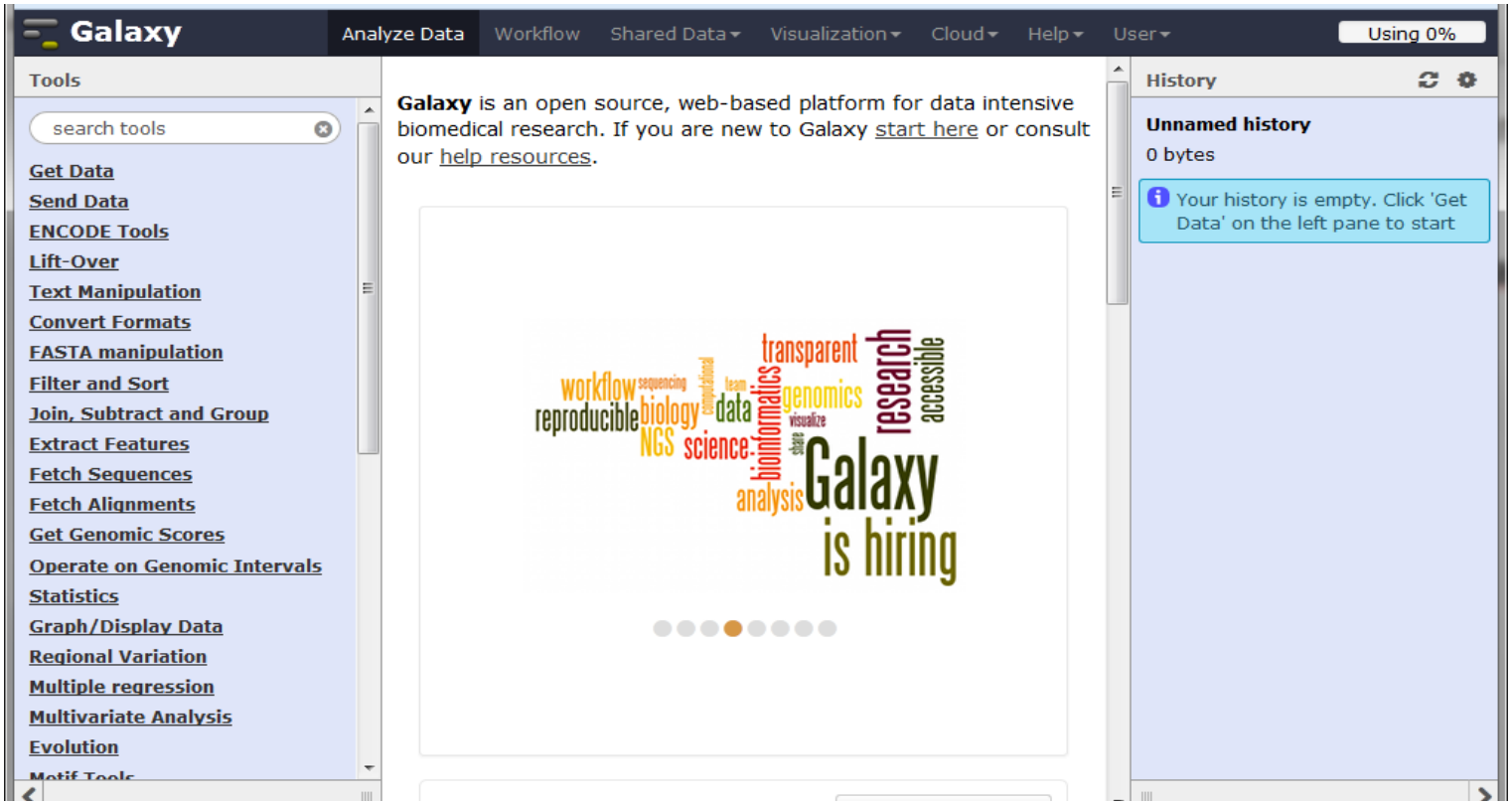
Installation d'une instance GALAXY

Sarah Maman

Instance publique du Galaxy Project

Sur le serveur public du Galaxy Project :https://usegalaxy.org/



De nombreuses autres instances publiques sont disponibles.

Instance SlipStream du Galaxy Project

Edité par Galaxy project

Solution plug-and-play

Inutile d'installer, de configurer et d'administrer une instance, ou de gérer les mises à jour code et outils.

Solution adaptée pour les biologistes non informaticiens.

Quelques infos sur la performance :

TOOLS	TASK	DATA	RUN-TIME
Bowtie 2	Mapping whole human genome	204 million paired-end 100bp Illumina reads	2 Hours 44 Minutes
SAMTools	SAM-BAM conversion	127GB SAM (41GB resulting BAM)	2 Hours 7 Minutes
TopHat 2	RNA-Seq mapping	24 million 100bp Illumina reads	1 Hours 24 Minutes
Cufflinks 2	Differential Expression Analysis	4.3 GB SAM File	0 Hours 11 Minutes

Source : http://bioteam.net/slipstream/galaxy-edition/

Instance sur un VM ou en local

VirtualBox est un logiciel de virtualisation de systèmes d'exploitation.

En utilisant les ressources matérielles de l'ordinateur (*système hôte*), VirtualBox permet la création d'un ou de plusieurs ordinateurs virtuels dans lesquels s'installent d'autres systèmes d'exploitation (*systèmes invités*). Source : http://doc.ubuntu-fr.org/virtualbox

Voici les principales étapes d'installation de Galaxy sur votre vm

1. Préparer votre espace de travail

   mkdir ~/galaxy-python
   ln -s /path/to/python2.7 ~/galaxy-python/python
   export PATH=~/galaxy-python:$PATH

2. Télécharger les sources de Galaxy (Mercurial ?)

   hg clone https://bitbucket.org/galaxy/galaxy-dist/
   cd galaxy-dist
   hg update stable

3. Lancer Galaxy

   sh run.sh

4. Paramétrer Galaxy : universe ini file
5.

Intérêts d'un vm

6. Faire fonctionner plus d'un système d'exploitation en même temps en toute sécurité
7. Possibilité de cloner une vm donc de partager des machines Galaxy
8. vm sauvegardée donc restaurable.

Architecture de l'instance toulousaine

Galaxy est installée sur une machine virtuelle qui envoie les calculs à un cluster.

Utilisateur de Galaxy (biologiste)

Envoi de données

Récupération des résultats

Serveur Web Galaxy

Envoi les jobs

Gère la file d'attente

Gestionnaire de tâches

Exécute

Cluster de calculs

Comparaison des différentes instances

En fonction du contexte d'utilisation et des ressources IT disponibles :

	NO WAIT TIMES	NO STORAGE QUOTAS	NO JOB SUBMISSION LIMITS	NO DATA TRANSFER BOTTLENECKS	NO IT EXPERIENCE REQUIRED	NO REQUIRED INFRASTRUCTURE
GALAXY MAIN	✗	✗	✗	✗	✓	✓
LOCAL GALAXY	?	?	?	✓	✗	✗
CLOUD GALAXY	✓	✓	✓	✗	✗	✓
SLIPSTREAM GALAXY	✓	✓	✓	✓	✓	✓

Source : http://bioteam.net/slipstream/galaxy-edition/

Configuration et administration d'une instance GALAXY

Ibouniyamine Nabihoudine - Sarah Maman
Sigenae

Configuration à l'aide du fichier galaxy.yml (ex universe_wsgi.ini)

- Fichier de configuration .ini :
  - Déploiement
  - Répertoire de travail
  - Base de donnée de travail, etc.
  -
- Ce fichier est organisé en sections :
  - **[server:main]** : configuration du serveur de déploiement
  - **[app:main]** : configuration de l'application Galaxy
  - Autres thèmes : Files and directories, Logging and Debugging, Job execution, Users and Security

- Configuration de l'adresse de déploiement de l'instance
  - Section **[server:main]**
  - Paramètres **host** et **port**

```
[server:main]

port = 8090
host = 127.0.0.1
```

... déploiement de l'instance Galaxy à l'adresse **http://127.0.0.1:8080**

Autres configurations

Configuration des fichiers et répertoires de travail

Job_working_directory : Scripts sh générés par Galaxy et lançés sur le cluster, répertoires/Fichiers de travail de Galaxy.

datatypes_conf_file : Fichier permettant de configurer les types de données

tool_conf.xml : Fichier de configuration et description des outils disponibles

Gestion des utilisateurs

Plusieurs méthodes de sécurisation d'une instance :

- Gestion des utilisateurs interne à Galaxy (universe.ini)
- Gestion externe des utilisateurs (LDAP)

Les paramètres à modifier pour la gestion interne à Galaxy :

- **require_login** : **True** pour limiter l'accès aux utilisateurs ayant un compte galaxy
- **allow_user_creation** : **True** pour autoriser un utilisateur à se créer lui-même un compte

Intégrer un outil (bio)informatique dans une instance GALAXY

Sarah Maman
Sigenae

Une galaxy de poupées russes

Mr Galaxy sur le web	• Instance locale • Partage ToolShed
Wrapper xml	• I/F web • Avec ou sans Cheetah
Wrapper perl	• Facultatif • Adaptable à chaque instance
(Bio)Info tool ou ligne de commande	• Localisation Path • Evolution

Fichier xml

Ce fichier XML est :

- un formulaire de saisie,
- visible depuis l'interface web Galaxy.

```
<tool id="fa_gc_content_1" name="Compute GC content">
  <description>for each sequence in a file</description>
  <command interpreter="perl">toolExample.pl $input $output</command>
  <inputs>
    <param format="fasta" name="input" type="data" label="Source file"/>
  </inputs>
  <outputs>
    <data format="tabular" name="output" />
  </outputs>

  <help>
This tool computes GC content from a FASTA file.
  </help>

</tool>
```

Nom du tool affiché sur l'interface web, dans le menu de gauche.

Nom interne à Galaxy

Label affiché sur l'interface web

Help : reStructuredText Markup Specification : http://docutils.sourceforge.net/docs/ref/rst/restructuredtext.html

Tags de base de votre XML

```
<tool id = "id_outil" name = "nom_outil "   version =
"version_outil">
        <description> description de l'outil </description>
        <command> perl ou cheeta </command>
        <inputs>
                <param ... />
        </inputs>
        <outputs>
                <data ... />
        </outputs>
        <help> commentaires, contact, citation </help>
</tool>
```

Tag <tool> et <description>

```
<tool id="sm_mothur_preprocess_V-1-0_date" name="* Nom de l'outil visible dans l'interface galaxy">
   <description>plus de détails</description>
       <command interpreter="perl">sm_wrapper_version_date.pl
                   $input
                   $param
                   "$int "
                   $output
        </command>
        <inputs> .. . </inputs>
        <outputs> ... </outputs>
        <help> ... </help>
</tool>
```

•L'id du tool doit être unique (integrated_tool_conf.xml)

•Le fichier sm_wrapper_version_date.pl est appelé dans la balise <command> du xml.

•Le xml et le wrapper perl (ou autre langage) doivent être dans le même répertoire : tools/vosInitiales/fwrapper.pl et file.xml

•Les outils Galaxy sont en anglais.

Tag <command>

Exécution directe :

- Ligne de commande :

```
<command>sed -r '$pattern' $input > $outfile  </command>
```

- CHEETA (http://www.cheetahtemplate.org/): langage puissant mais limité. D'où le passage par un wrapper.

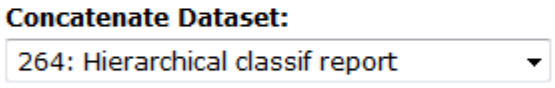
```
<command interpreter="python">gatk_wrapper.py
        --max_jvm_heap_fraction "4"
        --stdout "${output_log}"
        #if str( $ref ) != "None":
            -d "gatk_input"
        #end if   <command>
```

Exécution indirecte avec un wrapper :

- un script ini_tool_wrapper.pl est appelé dans la balise <command>
- Nous appellerons ce script : wrapper.
- Peu importe le langage de ce wrapper. Ils sont tous supportés par Galaxy.
- Il est cependant nécessaire de préciser le langage utilisé dans « interpreter ».

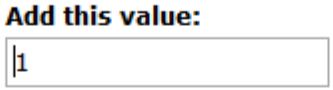
```
<command interpreter="perl">ini_tool_wrapper.pl … </command>
```

Fichier xml : principaux composants dans <inputs></inputs>



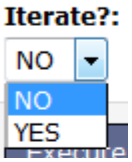
Fichier entrant :

<param name = "nom_interne" type = "data" format = "fasta" label = "affichage" /



Champs de saisie :

<param name = "nom_interne" type = "integer " value = "10.0" label = "affichage"/>



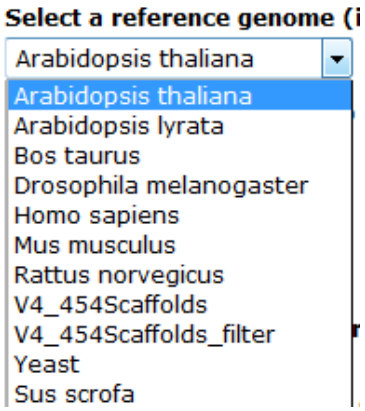
Liste :

<param name = "nom_interne" type = "select " label = « Iterate ?">
<option value = "T">NO</option>
<option value = "F">YES</option>
</param>

Boutons radio :

<param name = "nom_interne" type = "select " display = "radio" label = "affichage">
<option value = "megablast">megablast</option>
<option value = "blastn">blastn</option>
<option value = "blastn-short">blastn-short</option>
<option value = "dc-megablast">dc-megablast</option>
</param>

Fichier xml : principaux composants dans <inputs></inputs>

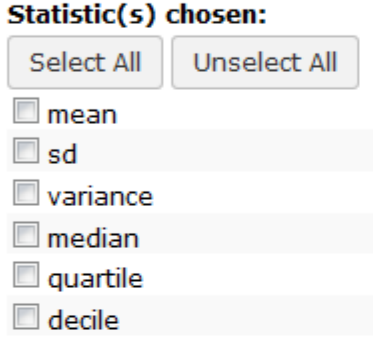


1 - XML avec le composant select pour le “Génome de référence ”

```
<param name="input_ref_genome" type="select" label="Select a reference genome (if  your genome
of interest is not li
sted, please contact Sigenae)">
        <options from_file="mirdeep2_indexes.loc">
            <column name="name" index="0"/>
            <column name="value" index="1"/>
        </options>
 </param>
```

2 - **mirdeep2_indexes.loc** dans tool-data/ :

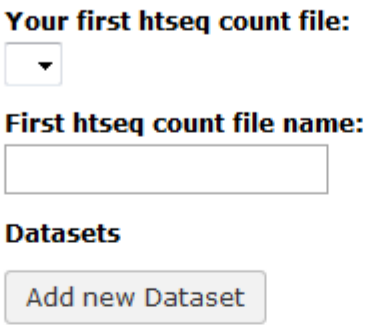
```
more mirdeep2_indexes.loc
Arabidopsis thaliana    /save/galaxy-dev/bank/mirdeep2/Arabidopsis_thaliana
Arabidopsis lyrata    /save/galaxy-dev/bank/mirdeep2/ensembl_arabidopsis_lyrata_genome
Bos taurus    /save/galaxy-dev/bank/mirdeep2/ensembl_bos_taurus_genome
```



Cases à cocher:

```
 <param name="stat_chosen" type="select" display="checkboxes" multiple="True"
label="Statistic(s ) chosen">
       <option value="mean">mean</option>
       <option value="sd">sd</option>
       <option value="variance">variance</option>
       <option value="median">median</option>
       <option value="quartile">quartile</option>
       <option value="decile">decile</option>
       <validator type="empty_field" message="Please choose a statistic representation" />
 </param>
```

Fichier xml : principaux composants dans <inputs></inputs>



1 - Répéter un paramètre dans le tag <inputs> du XML :

```
<param format="tab" name="input_htseqcount" type="data" label="Your first htseq count file"/>
<param name="name1" size="20" type="text" value="" label="First htseq count file name"/>
    <repeat name="queries" title="Dataset">
        <param name="inputs_count" type="data" format="tab" label="Other htseq count files" />
        <param name="names" size="20" type="text" value="" label="htseq count file name"/>
    </repeat>
```

1 - Répéter un paramètre dans le tag <command> du XML :

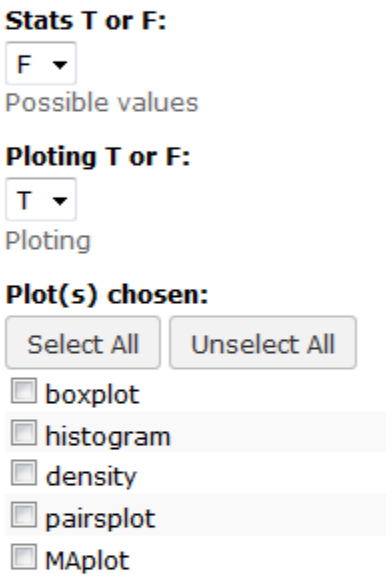
```
<command interpreter="perl">sm_htseqcount_merge.pl
...   #for $q in $queries
      ${q.inputs_count}
      ${q.names}
  #end for
</command>
```

Plus d'infos : https://wiki.galaxyproject.org/Admin/Tools/ToolConfigSyntax?action=show&redirect=Admin%2FTools%2FTool+Config+Syntax

Fichier xml : principaux composants dans <inputs></inputs>

1 – Une condition dans le tag <inputs> du XML :

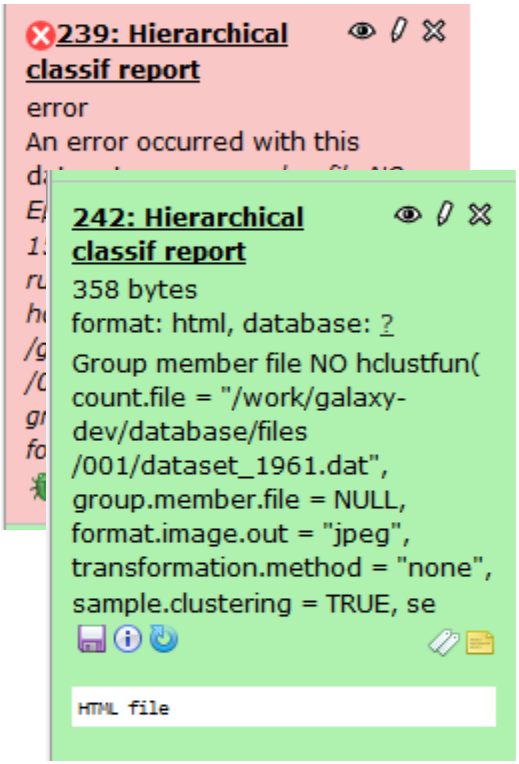
```
<conditional name="stat_cond">
        <param name="stat" type="select" help="Possible values" label="Stats T or F">
                <option value="T">T</option>
                <option value="F">F</option>
        </param>
        <when value="F" />
        <when value="T">
                <param name="stat_chosen" type="select" display="checkboxes" multiple="True" label="Statistic(s
                        <option value="mean">mean</option>
                        ........
                         <validator type="empty_field" message="Please choose a statistic representation" />
                </param>
        </when>
</conditional>

<conditional name="ploting_cond">
        <param name="ploting" type="select" help="Ploting" label="Ploting T or F">
                <option value="T">T</option>
                <option value="F">F</option>
        </param>
        <when value="F" />
        <when value="T">
                <param name="plot_chosen" type="select" help=""  display="checkboxes" multiple="True" label="Plot(s) chosen">
                        <option value="boxplot">boxplot</option>
                        <option value="histogram">histogram</option>
                        <option value="density">density</option>
                        <option value="pairsplot">pairsplot</option>
                        <option value="MAplot">MAplot</option>
                </param>
        </when>
</conditional>
```



2 – Une condition dans le tag <command> du XML :

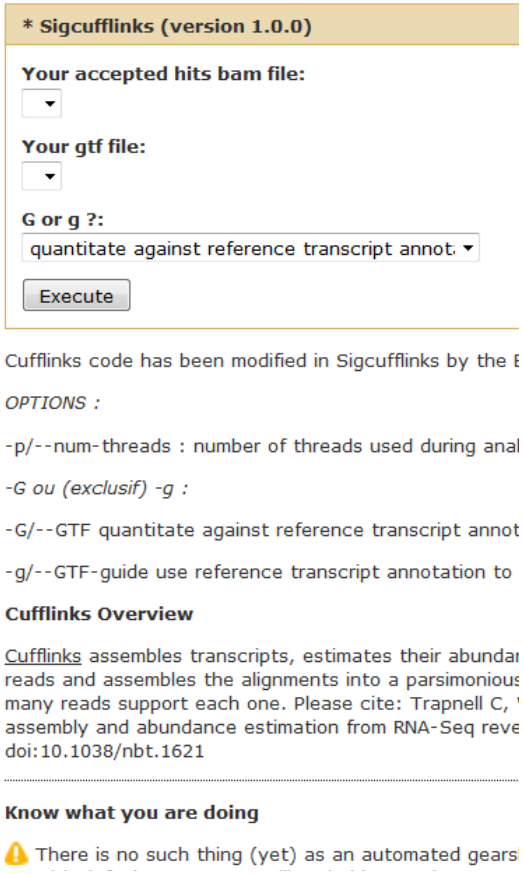
```
<command interpreter="perl">Graphics_desc.pl
        ...
        $stat_cond.stat
        $stat_cond.stat_chosen
        $ploting_cond.ploting
        $ploting_cond.plot_chosen
        ....
 </command>
```

Fichier xml : principaux composants dans <outputs></outputs>




```
<outputs>
        <data format = "pdf" name = " result" label " result of ${tool.name} on
        ${input.name} />
        <filter>pdf is True</filter>
        <!--Ne pas afficher cet output s'il n'est pas généré - - >

     <data name="blast_out" format="tabular" label="BLAST Report">
    <change_format>
    <- - format variable en fonction des conditions d'éxécution - - >
            <when input="view_options" value="0" format="txt"/>
            <when input="view_options" value="7" format="blastxml"/>
            <when input="view_options" value="8" format="tabular"/>
            <when input="html_output" value="T" format="html"/>
     </change_format>
</data>
</outputs>
```

Fichier xml : Commentaires de l'outil avec la balise <help></help>



<help>

****Titre en gras****

Pour plus de détails, cliquer ici_..._ ici: http://www.google.fr

.. class:: warningmark
Warning

How to cite

</help>

Wrapper type

Ajouter la licence

```
Copyright (C) 2009 INRA #
This program is free software: you can redistribute it and/or modify
it under the terms of the GNU General Public License as published by
the Free Software Foundation, either version 3 of the License, or
(at your option) any later version. #
This program is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
GNU General Public License for more details. #
You should have received a copy of the GNU General Public License
along with this program. If not, see <http://www.gnu.org/licenses/>
```

Precider la version de l'outil informatique ou bioinfo sous-jacent

```
<version_command>tophat -version</version_command>
```

Wrapper type

Exemple de code du xml

Le nom du wrapper commence par une * lorsqu'il s'agit d'un tool ajouté par Sigeane ou PF BioInfo à l'instance Galaxy. L'objectif étant de distinguer façilement les tools de base de Galaxy Project des tools ajoutés dans l'instance.

```
<tool id="unique_wrapper_id" name="* nom du wrapper">
    <description></description>
          <command interpreter="perl">vosInitiales_nomWrapper.pl
    --input_1 $input1
    --param $param
    --output1 $output1
    --selector $conditional.selector
    #if $conditional.selector =="1":
       --conditionalparam1 $conditional.param1
       --conditionalparam2 $conditional.param2
    #else
       --conditionalparam3 $conditional.$param3
       --conditionalparam4 $conditional.$param4
    #end if
          </command>
                <inputs>
                      <param format="txt" name="input1" type="data" label="First input file"/>
                      <param name="param" type="select" label="Param">
                            <option value="fastq">fastq</option>
                            <option value="fq">fq</option>
                        </param>
                    <conditional name="conditional">
                        <param name="selector" type="select" label="Question for biologists ?">
                          <option value="0">No</option>
                          <option value="1">Yes</option>
                        </param>
                        <when value="1">
                            <param name="param1" type="data" format="fasta" label="Path to file 1"/>
                            <param name="param2" type="data" format="fasta" label="Path to file 2"/>
                        </when>
                        <when value="0">
                            <param name="param3" size="20" type="text" value="0" label="param 3"/>
                            <param name="param4" size="20" type="text" value="0" label="param 4"/>
                        </when>
                    </conditional>
                </inputs>
                <outputs>
                   <data format="fasta" name="output1" label ="file.fasta"/>  <!-- choisir un label le plus court possible -->
                </outputs>
  <help>
     Voir la section du wiki Galaxy consacrée à la rédaction des tools Galaxy.
  </help>
</tool>
```

Description du fichier tool_conf.xml

Ce fichier décrit la toolbox : le menu de gauche de l'interface Galxy.

```
<?xml version="1.0"?>
<toolbox>

    <label text="Label1" id="feat" />

    <section name="Get Data" id="get_data">
        <tool file="data_source/upload.xml"/>
        <tool file="sm_upload/genotoul_upload.xml" />
        <tool file="data_source/ucsc_tablebrowser.xml" />
        <tool file="data_source/ucsc_tablebrowser_test.xml" />
        <tool file="data_source/ucsc_tablebrowser_archaea.xml" />
        <tool file="data_source/microbial_import.xml" />
        <tool file="data_source/biomart.xml" />
        <!--<tool file="sm_save/save.xml"/>-->
    </section>

    <section name="Text Manipulation" id="textutil">
        <tool file="stats/column_maker.xml" />
        <tool file="filters/fixedValueColumn.xml" />
        <tool file="filters/catWrapper.xml" />
        <tool file="filters/cutWrapper.xml" />
        <tool file="filters/mergeCols.xml" />
        <tool file="filters/convert_characters.xml" />
        <tool file="filters/CreateInterval.xml" />
        <tool file="filters/cutWrapper.xml" />
        <tool file="filters/changeCase.xml" />
        <tool file="filters/pasteWrapper.xml" />
        <tool file="filters/remove_beginning.xml" />
        <tool file="filters/randomlines.xml" />
        <tool file="filters/headWrapper.xml" />
        <tool file="filters/tailWrapper.xml" />
        <tool file="filters/trimmer.xml" />
        <tool file="filters/wc_gnu.xml" />
        <tool file="filters/secure_hash_message_digest.xml" />
        <tool file="new_operations/tables_arithmetic_operations.xml" />
    </section>

</toolbox>
```

Label descriptif

L'outil est placé dans une section.
Le path est relatif.

Configuration de l'exécution des tools avec job_conf.xml

```
<?xml version="1.0"?>
<job_conf>

    <!-- Plateformes d'execution des programmes -->
    <plugins>
        <plugin id="local" type="runner" load="galaxy.jobs.runners.local:LocalJobRunner" workers="4"/>
        <plugin id="sge" type="runner" load="galaxy.jobs.runners.drmaa:DRMAAJobRunner" workers="4"/>
    </plugins>

    <handlers>
        <handler id="main"/>
    </handlers>

    <!-- Chaque destinattion doit correpondre a une plateforme d'execution-->
    <destinations default="real_user_cluster">
        <destination id="local" runner="local"/>
        <destination id="real_user_cluster" runner="sge">
            <param id="galaxy_external_runjob_script">scripts/drmaa_external_runner.py</param>
            <param id="galaxy_external_killjob_script">scripts/drmaa_external_killer.py</param>
            <param id="galaxy_external_chown_script">scripts/external_chown_script.py</param>
        </destination>
    </destinations>

    <!-- Forcer des outils s'executer sur unse seule destination -->
    <tools>
        <tool id="genotoul_upload" destination="local"/>
    </tools>

</job_conf>
```

Plateformes d'éxecution

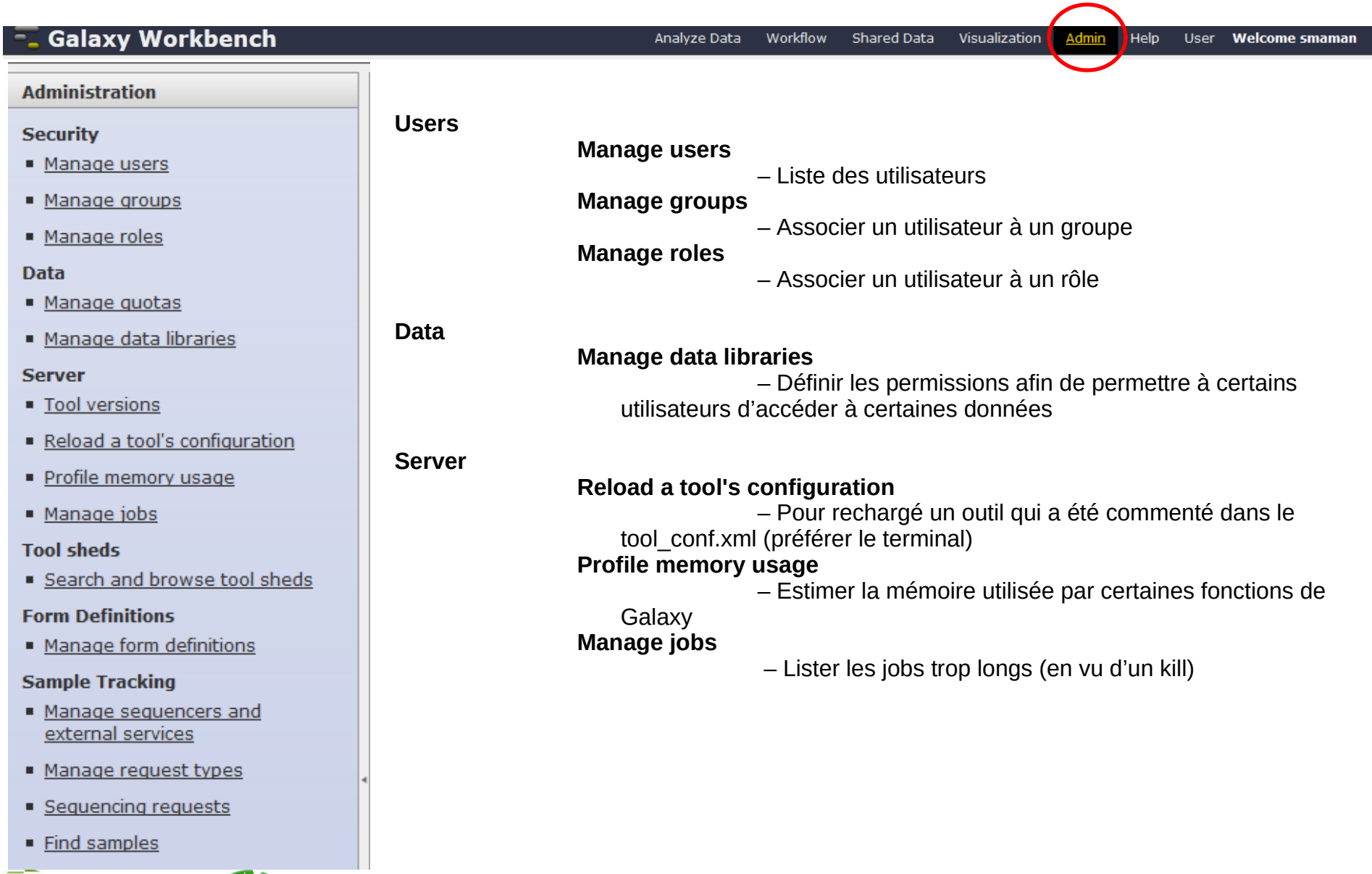
Configuration des destinations

1outil = 1 destination unique

Le fichier job_conf.xml permet de spécifier les paramètres du qsub finement :

```
<destination id="gatk_variant_select_job" runner="drmaa">
    <param id="galaxy_external_runjob_script">scripts/drmaa_external_runner.py</param>
    <param id="galaxy_external_killjob_script">scripts/drmaa_external_killer.py</param>
    <param id="galaxy_external_chown_script">scripts/external_chown_script.py</param>
    <param id="nativeSpecification">-clear -q workq -l mem=10G -l h_vmem=50G -pe parallel_smp
4</param>
</destination>
```

Administration de Galaxy via l'interface web



Users

Manage users
– Liste des utilisateurs

Manage groups
– Associer un utilisateur à un groupe

Manage roles
– Associer un utilisateur à un rôle

Data

Manage data libraries
– Définir les permissions afin de permettre à certains utilisateurs d'accéder à certaines données

Server

Reload a tool's configuration
– Pour rechargé un outil qui a été commenté dans le tool_conf.xml (préférer le terminal)

Profile memory usage
– Estimer la mémoire utilisée par certaines fonctions de Galaxy

Manage jobs
– Lister les jobs trop longs (en vu d'un kill)