



Mise en place de tests automatiques des outils dans l'environnement Galaxy



Etudiant : EYMARD Thomas

Maître de stage : BATUT Berenice

Stage de 2^e année de DUT Bio-informatique

Galaxy

- Plate-forme ouverte et open source
- Utilisation de galaxy :
 - Galaxy dispose de serveurs publics
 - Il existe de nombreuses instances galaxy « fermées »
 - Il est possible d'installer galaxy sur n'importe quel ordinateur
- Permet de lancer des outils bioinformatiques afin d'effectuer des analyses complètes et reproductibles via des wrappers

Galaxy

- Les outils bioinformatiques marchent en ligne de commande

```
graphlan.py input.xml image.png --format pdf --dpi 100 --size 7 --pad 2
```

- Le wrapper sert d'interface entre l'utilisateur et l'outil
 - Apelle une version de l'outil donnée
 - Gère les fichiers et les paramètres
 - Gère les dépendances ainsi que les packages

Galaxy

The screenshot displays the Galaxy web interface. At the top, a dark navigation bar contains the 'Galaxy' logo and several menu items: 'Analyze Data', 'Workflow', 'Shared Data', 'Visualization', 'Help', and 'User'. On the far right of this bar, it indicates 'Using 0 bytes'.

The main interface is divided into three vertical panels:

- Left Panel (Tools):** Features a search bar and a list of tool categories. Under 'COMMON TOOLS', there are links for 'Get Data', 'Manipulate files', 'Manipulate sequence files', and 'Manipulate BAM/SAM files'. Under 'SEQUENCE PREPARATION TOOLS', there are links for 'Assemble paired-end sequences', 'Control quality', 'Chimera detection, dereplication, clustering, ... with VSEARCH', 'Cluster sequences', 'Manipulate RNA', 'Search similarity', and 'Map against reference genomes'. Under 'METAGENOMIC TOOLS', there are links for 'Assign taxonomy on all sequence type', 'Analyze metabolism', 'Combine functional and taxonomic results', and 'Visualize data'. Under 'STATISTICS AND VISUALIZATION TOOLS', there are links for 'Export to GraPhlAn', 'GraPhlAn to produce graphical output of an input tree', 'Generation, personalization and annotation of tree for GraPhlAn', 'Krona pie chart from taxonomic profile', 'Plot barplot with R', 'Plot grouped barplot with R', 'Plot generic X-Y plot with R', and 'Histogram of a numeric column'.
- Center Panel (GraPhlAn tool):** The title is 'GraPhlAn to produce graphical output of an input tree (Galaxy Version 0.9.7)'. It includes an 'Options' dropdown. The 'Input tree' section has a file upload icon and a text box stating 'No txt dataset available.' with a note: 'The tree must be in PhyloXML, Newick or text format.' The 'Output format' is set to 'PNG' with a note '(--format)'. The 'Dpi of the output image (Optional)' is an empty text box with a note 'For non vectorial formats (--dpi)'. The 'Size of the output image (in inches)' is set to '7' with a note '(--size)'. The 'Distance between the most external graphical element and the border of the image (Optional)' is an empty text box with a note '(--pad)'. At the bottom of this section is a blue 'Execute' button with a checkmark icon.
- Right Panel (History):** Features a search bar and a section titled 'Unnamed history' showing '0 b'. A blue information box states: 'This history is empty. You can load your own data or get data from an external source'.

Below the tool configuration, there is a section titled 'What it does' which describes GraPhlAn as a software tool for producing high-quality circular representations of taxonomic and phylogenetic trees. It mentions that GraPhlAn focuses on concise, integrative, informative, and publication-ready representations of phylogenetically- and taxonomically-driven investigation. It also provides a link to the 'user manual' for more information.

Test logiciel

« Le test est une technique de contrôle consistant à s'assurer, au moyen de son exécution, que le comportement d'un programme est conforme à des données préétablies »

- Applique sur tout ou une partie du système informatique un échantillon de données d'entrées et d'environnement
- Vérifie si le résultat obtenu est conforme aux attentes
- Intérêts du test :
 - Garantit que le programme fonctionne
 - Garantit que des changements n'impactent pas les résultats

Test logiciel

- Tests Unitaires: tester un seul élément ou un seul module du logiciel à la fois
- Tests fonctionnels : tester un logiciel entier après avoir sélectionné les données en fonction du code du logiciel

Le test de wrappers

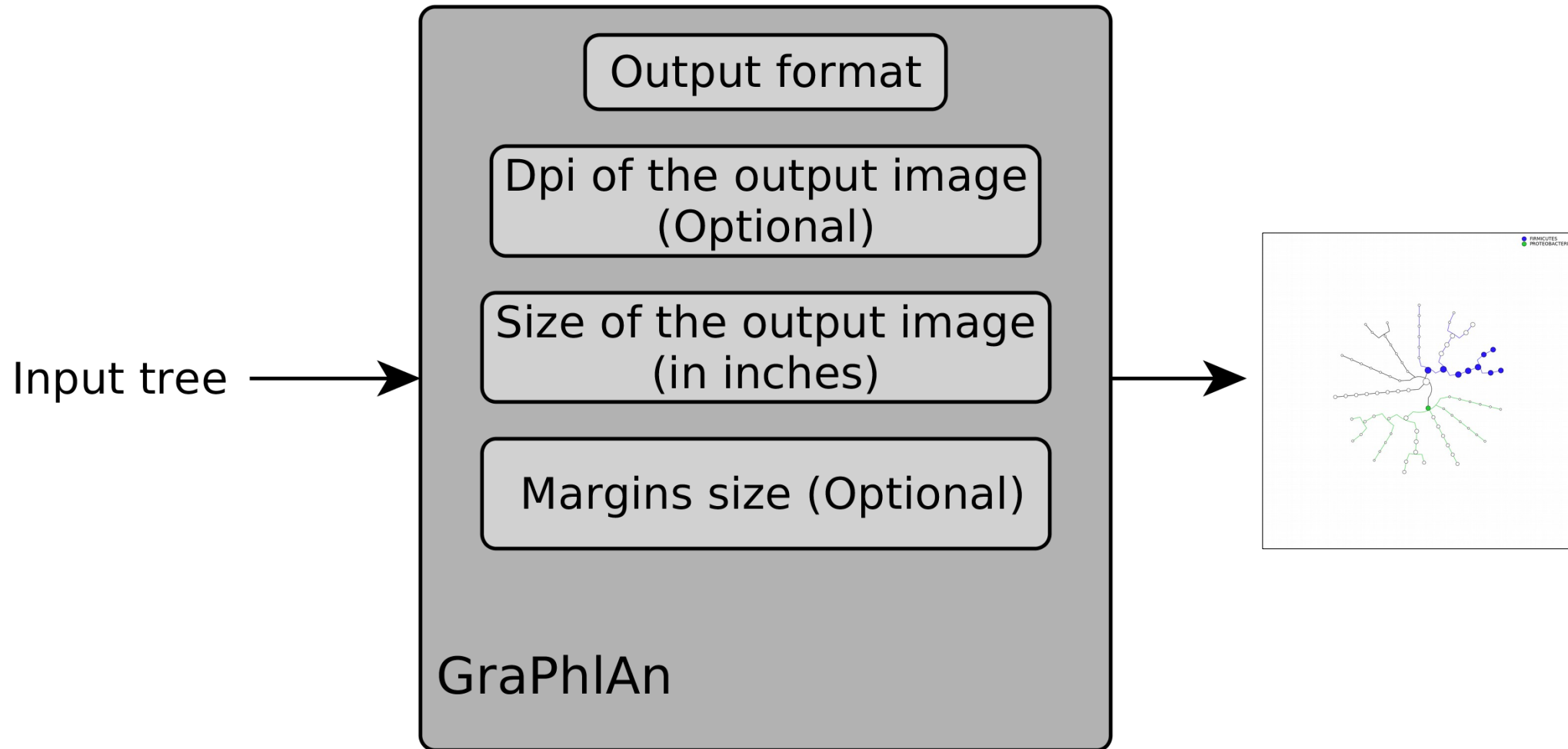
- Code des wrappers simple
- Paramètres des outils souvent particuliers
 - Erreures et inatentions fréquentes
- Test des wrappers nécessaire pour un code correct
- Permettent aussi de tester les limites fixées

Processus de tests des wrappers

Familiarisation avec l'outil bioinformatique

- Identification des fichiers entrées et sorties de l'outil (nombre, format, contenu)
- Identification des paramètres de l'outil
- Génération des entrées/sorties nécessaires pour le test

Familiarisation avec l'outil bioinformatique



Familiarisation et modification du wrapper

```
<tool id="graphlan" name="GraPhlAn" version="0.9.7">
  <description>to produce graphical output of an input tree</description>
  <macros>
    <import>graphlan_macros.xml</import>
  </macros>
  <expand macro="requirements"/>
  <stdio>
    <regex match="Warning"
           source="stderr"
           level="warning"
           description="" />
  </stdio>
  <version_command>
<![CDATA[
graphlan.py -v
]]>
  </version_command>
```

Familiarisation et modification du wrapper

```
<command>
<![CDATA[
    graphlan.py
        --format $format

        #if $dpi
            --dpi $dpi
        #end if

        --size $size

        #if $pad
            --pad $pad
        #end if

        $input_tree

        #if str($format) == "png"
            $png_output_image
        #else if str($format) == "pdf"
            $pdf_output_image
        #else if str($format) == "ps"
            $ps_output_image
        #else if str($format) == "eps"
            $eps_output_image
        #else
            $svg_output_image
        #end if
]]>
</command>
```

Familiarisation et modification du wrapper

```
<inputs>
  <param name="input_tree" type="data" format="txt" label="Input tree" help="The tree must be
in PhlyloXML, Newick or text format."/>

  <param name='format' type="select" label="Output format" help="--format">
    <option value="png" selected="true">PNG</option>
    <option value="pdf">PDF</option>
    <option value="ps">PS</option>
    <option value="eps">EPS</option>
    <option value="svg">SVG</option>
  </param>

  <param name="dpi" type="integer" label="Dpi of the output image
(Optional)" help="For non vectorial formats (--dpi)" optional="True"/>

  <param name="size" type="integer" value="7" label="Size of the output image
(in inches)" help="--size"/>

  <param name="pad" type="integer" label="Distance between the most external
graphical element and the border of the image (Optional)"
help="--pad" optional="True"/>
</inputs>
```

Familiarisation et modification du wrapper

```
<outputs>
  <data format="png" name="png_output_image"
    label="${tool.name} on ${on_string}: Image">
    <filter>format=="png"</filter>
  </data>
  <data format="pdf" name="pdf_output_image"
    label="${tool.name} on ${on_string}: Image">
    <filter>format=="pdf"</filter>
  </data>
  <data format="ps" name="ps_output_image"
    label="${tool.name} on ${on_string}: Image">
    <filter>format=="ps"</filter>
  </data>
  <data format="eps" name="eps_output_image"
    label="${tool.name} on ${on_string}: Image">
    <filter>format=="eps"</filter>
  </data>
  <data format="svg" name="svg_output_image"
    label="${tool.name} on ${on_string}: Image">
    <filter>format=="svg"</filter>
  </data>
</outputs>
```

Familiarisation et modification du wrapper

```
<tests>
  <test>
    <param name="input_tree" value="intermediary_tree.txt"/>
    <param name="format" value="png"/>
    <param name="dpi" value="100"/>
    <param name="size" value="7"/>
    <param name="pad" value="2"/>
    <output name="png_output_image" file="png_image.png" />
  </test>
</tests>
```

Test localement avec planemo

- Planemo développé pour faciliter la création de wrappers pour Galaxy
- Dispose de commandes de test :
 - `planemo lint` : teste la bonne forme du code xml
 - `planemo shed_lint` : Vérifie si le wrapper est théoriquement bon dans sa forme générale
 - `planemo test` : Vérifie la correspondance des tests pour le wrapper

Intégration dans le dépôt GitHub

- Outils à tester présents sur le dépôt Github
- Dépôt dépend de Git un logiciel de gestion de versions
- Intérêts de Git :
 - Permet de créer des branches (versions d'un même dossier indépendantes les unes des autres)
 - suit l'évolution des fichiers
 - stocke les anciennes versions de tous les fichiers

Intégration dans le dépôt GitHub

Addtest for `combine_methaph_lan2_human2` #16

Edit

Merged bebatut merged 6 commits into ASaiM:master from themyard1:br_combine_methaphLan2_human2 on 15 Apr

Conversation 4

Commits 6

Files changed 7

+33,293 -29,435



themyard1 commented on 13 Apr

+ 😊

to resolve #1

Labels

None yet

Milestone

No milestone

Assignees

No one assigned

Notifications

Unsubscribe

You're receiving notifications because you authored the thread.

2 participants



themyard1 added some commits on 12 Apr



Add missing test

b44e727



Remove some lines

✗ a5222f1



bebatut commented on an outdated diff on 14 Apr

Show 1 comment



bebatut commented on 14 Apr

+ 😊

I will add this test for CI



themyard1 added some commits on 14 Apr



Merge branch 'master' of https://github.com/ASaiM/galaxytools into br...

a34c544



Add good outputs and edit tt_blacklist

✓ e79736b



update .tt_blacklist

✓ 3f792eb



bebatut commented on 15 Apr

+ 😊

The test files are not good ones. I can send you some if you want



Change inputs and outputs files

✓ a34b11a



bebatut commented on 15 Apr

+ 😊



Test dans un environnement "propre"

- Travis CI permet de créer un environnement totalement vierge et n'y installe que ce qui lui est demandé
- Intérêts :
 - Les logiciels à installer doivent être marqués dans le fichier `.travis.yml`
 - Le test n'est pas pollué par d'anciennes versions ou des logiciels tiers

Travail effectué

Problèmes rencontrés et leurs solutions

Des paramètres particuliers : humann2_split_table

Particularité : L'outil renvoie un nombre de fichiers dépendant du fichier d'entrée, on ne peut déterminer ce nombre à l'avance

```
<tests>
  <test>
    <param name="input_file" value="joined_pathway_coverage_abundance.tsv"/>
    <output_collection name="split_tables" type="list" >
      <element name="humann2_Abundance" file="split_joined_table_abundances.tsv" />
      <element name="humann2_Coverage" file="split_joined_table_coverage.tsv" />
    </output_collection>
  </test>
</tests>
```

Solution : Rechercher des exemples et des conversations sur le sujet

Problèmes rencontrés et leurs solutions

Des erreurs parfois incompréhensibles :

group_humann2_uniref_abundances_to_GO

| 5 - Group abundances of UniRef50 gene families obtained with HUMAnN2 on data 4, data 3, and
| others: Molecular function abundance (HID - NAME)

| Dataset Blurb:

| error

| Dataset Info:

| discarding /home/cidam/conda/bin from PATH

| prepending /tmp/tmpxjziTr/job_working_directory/000/5/conda-env/bin to PATH

| Fatal error: Exit code 1 ()

| Option -p requires an argument.

| Dataset Job Standard Output:

| discarding /home/cidam/conda/bin from PATH

| prepending /tmp/tmpxjziTr/job_working_directory/000/5/conda-env/bin to PATH

| Dataset Job Standard Error:

| Fatal error: Exit code 1 ()

| Option -p requires an argument.
|

Problèmes rencontrés et leurs solutions

Des dépendances n'étant pas installées

Souvent les dépendances ne sont pas installées ou ne sont plus à la bonne version, entraînant une erreur.

Solution : Il faut les mettre à jour ou les installer correctement

Problèmes rencontrés et leurs solutions

Des outils mal documentés

humann2_merge_abundance_tables

9. Merge abundance tables

```
$ humann2_merge_abundance_tables --input-genes $INPUT_GENES.tsv --input-pathways $INPUT_PATHWAYS.tsv --output $OUTPUT.tsv
```

- \$INPUT_GENES.tsv = a file containing the gene families (or EC) abundance table (tsv format)
- \$INPUT_PATHWAYS.tsv = a file containing the pathways abundance table (tsv format)
- \$OUTPUT.tsv = the file to write the new merged abundance table (tsv format)
- Optional: `--remove-taxonomy` remove the taxonomy from the output file

Solution : effectuer des recherches sur internet par exemple les Google Groups

Wrappers testés

Outils humann2 :

- humann2_join_tables
- humann2_reduce_table
- humann2_regroup_table
- humann2_regroup_table
- humann2_rename_table
- humann2_renorm_table
- humann2_renorm_table
- humann2_split_table
- compare_humann2_output

Outil metaphlan2 :

- format_metaphlan2_output

Outil :

- combine_metaphlan2_humann2

Wrappers testés

Outils graphlan:

- graphlan
- graphlan_annotate
- export2graphlan

Outils plot:

- plot_barplot
- plot_generic_x_y_plot
- plot_grouped_barplot


Outils cdhit:

- cd_hit_est
- cd_hit_protein
- format_cd_hit_output

Autres outils:

- extract_min_max_lines
- fasta_add_barcode
- normalize_dataset
- compute_wilcoxon_test
- extract_sequence_file

Et après?

 **Galaxy Tool Shed**

Repositories Groups Help User

3904 valid tools on May 15, 2016

Search

- Search for valid tools
- Search for workflows

Valid Galaxy Utilities

- Tools
- Custom datatypes
- Repository dependency definitions
- Tool dependency definitions

All Repositories

- Browse by category

Available Actions

- Login to create a repository

Repositories by Category

search repository name, description

Name	Description	Repositories
Assembly	Tools for working with assemblies	74
ChIP-seq	Tools for analyzing and manipulating ChIP-seq data.	40
Combinatorial Selections	Tools for combinatorial selection	6
Computational chemistry	Tools for use in computational chemistry	21
Convert Formats	Tools for converting data formats	64
Data Managers	Utilities for Managing Galaxy's built-in data cache	32
Data Source	Tools for retrieving data from external data sources	35
Epigenetics	Tools for analyzing Epigenetic/Epigenomic datasets	3
Fasta Manipulation	Tools for manipulating fasta data	76
Fastq Manipulation	Tools for manipulating fastq data	54
Genome-Wide Association Study	Utilities to support Genome-wide association studies	20
Genomic Interval Operations	Tools for operating on genomic intervals	42
Graphics	Tools producing images	42
Imaging	Utilities to support imaging	1
Metabolomics	Tools for use in the study of Metabolomics	25
Metagenomics	Tools enabling the study of metagenomes	58
Micro-array Analysis	Tools for performing micro-array analysis	8
Next Gen Mappers	Tools for the analysis and handling of Next Gen sequencing data	105
Ontology Manipulation	Tools for manipulating ontologies	10
Phylogenetics	Tools for performing phylogenetic analysis	19
Proteomics	Tools enabling the study of proteins	78
RNA	Utilities for RNA	84
SAM	Tools for manipulating alignments in the SAM format	80

Conclusion