BioSeD - **Bio**logical **Se**quences **D**atabase
User Manual

February 20, 2010

Contents

1 Installation 2

2 Labels 3

2.1 Creating new labels 5

2.2 Import / Export 6

3 Sequences 6

3.1 Labels page 7

3.2 Batch 9

4 Taxonomies 10

4.1 Managing trees 10

4.1.1 Tree browsing 12

4.2 Managing ranks 13

4.2.1 Import / Export 13

4.3 Managing taxonomies 14

5 Search 15

5.1 Query input 16

5.2 Operations 18

5.3 Sub-sequences 19

5.4 Histograms 19

5.5 Export 19

5.6 Batch labels 19

5.7 Delete 20

5.8 Preview 20

5.9 Written queries 22

6 File formats 23

7 Administration 23
7.1 User management 23
7.1.1 New user 23
7.1.2 User settings 23
7.1.3 Managing other users 24
7.2 Import / Export database 24
7.3 Application customization 25

8 File formats 25
8.1 FASTA 25
8.2 XML 27

9 Query language 32

List of Figures

1 System home page. 3
2 Listing all labels 4
3 Import labels report 6
4 Sequence page. 7
5 Example labels page. 8
6 Annotating a new object label. 9
7 Upload sequences page. 10
8 Batch sequence report. 11
9 Tree listing. 11
10 Viewing a tree. 12
11 Browsing the NCBI tree. 13
12 Import rank report 14
13 Editing taxonomy information. 15
14 Other names section. 15
15 Search query input. 16
16 Search operations. 19
17 Plotting histograms. 20
18 Export search page. 21
19 Add label to multiple sequences. 21
20 Search preview results. 21
21 Using view label. 22
22 Page with user information. 24
23 Query language written in BNF. 32

List of Tables

1 Label types and operators. 18
2 Operators and values in query expressions. 31

Listings

1 Plain FASTA format. 26
2 Complex FASTA format example. 26
3 An example Label XML file. 27
4 An Sequence XML file. 28
5 An example Rank XML file. 28
6 An example Taxonomy tree XML file. 29
7 Database in XML skeleton. 30

In this appendix we will cover the main use cases for the application, from a user point of view. After reading this section, the user should be able to install and use the application.

1 Installation

Before installing the application we will need to download the install package from the project webpage. Two methods are available to download the application: downloading a release or through Subversion, getting the most recent source code updates and fixes.

First, point your browser to `http://code.google.com/p/ibmc-bio-db/`.

If you want to use Subversion, select **Source** and once the page has loaded you should see the Subversion checkout command. Run:

```
$ svn checkout http://ibmc-bio-db.googlecode.com/svn/trunk/ biosed
```

If you want to download a stable release, select **Downloads** and click on the most up-to-date version. Once the file as downloaded, run:

```
$ tar zxvf biosed-version.tar.gz
```

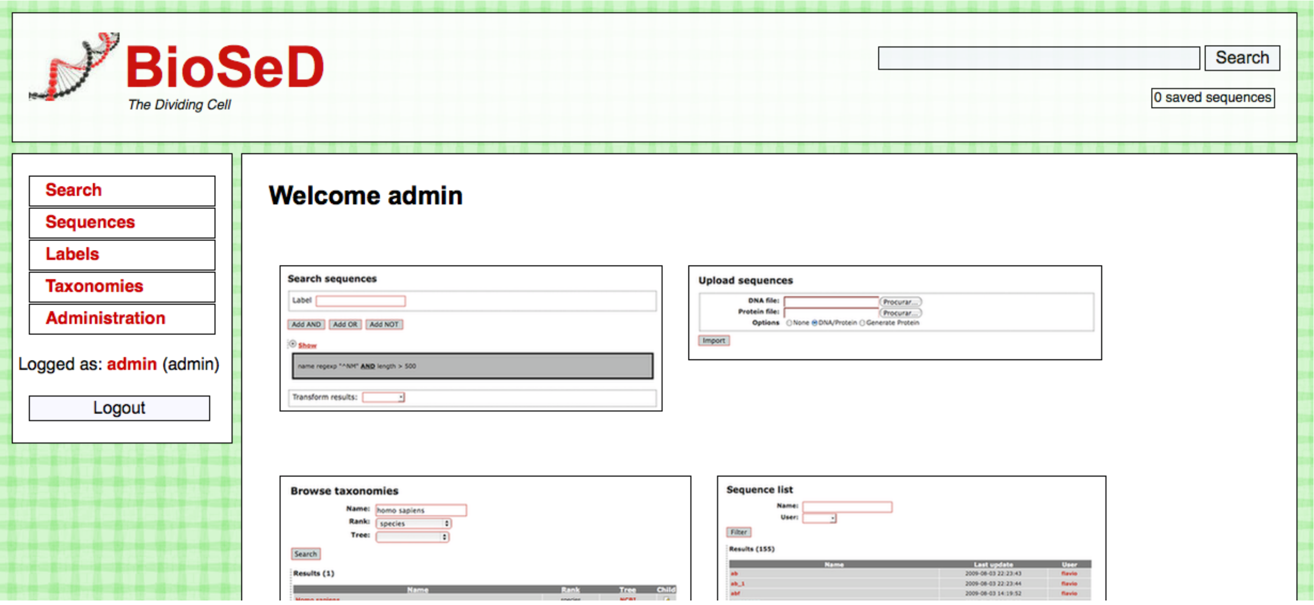
For both methods, enter into the **biosed** directory. Take a look at the file **README** and follow the instructions.

Once done, the application should be installed. Congratulations!

The install scripts creates one user for using the application, the *admin* user, and, as name says has more rights than *normal* users. Right now, you should point your browser to the application's location and login with this user.

Once the page has loaded, you should now see the main application's page. On the left is the main menu, in the header there is a search box and at the main box is located the main application's work area.

Right below the main menu there is a login box. Put *admin* in the user field and use the password you provided during the installation process.



Logged as: admin (admin)

Welcome admin

name regexp "^NM" AND length > 500

Options ○ None ● DNA/Protein ○ Generate Protein

Name: homo sapiens
Rank: species
Tree:

Results (1)

Name	Rank	Tree	Child

Name:
User:

Results (155)

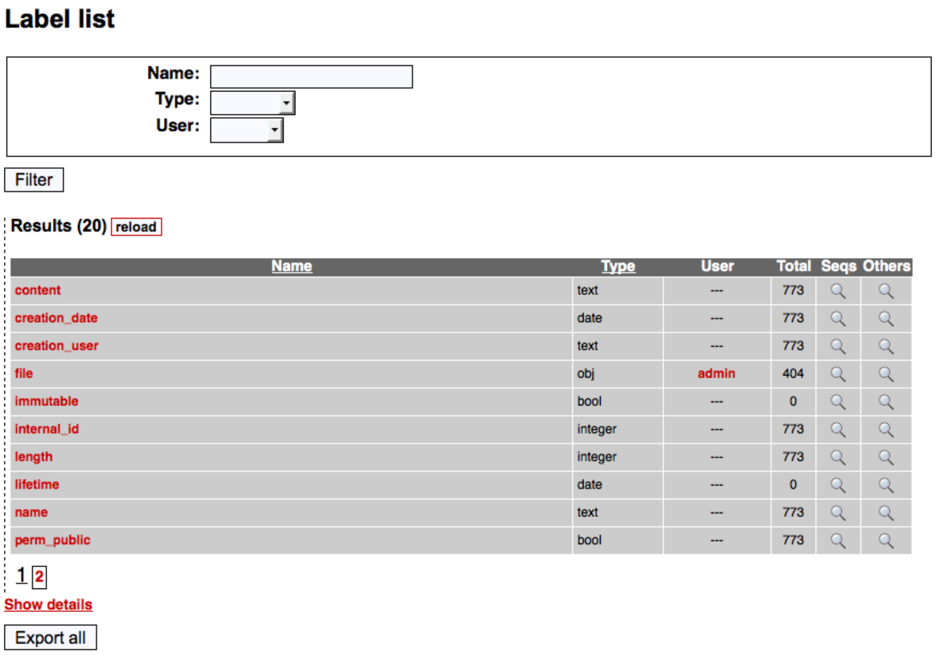
Name	Last update	User
ab	2009-08-03 22:23:43	flavio
ab_1	2009-08-03 22:23:44	flavio
abf	2009-08-03 14:19:52	flavio

Figure 1: System home page.

If you entered the correct password, the new page should look like Figure 1.

2 Labels

Label related options are found in the **Labels** submenu. For normal users only the options **Labels - List** and **Labels - Export** are available. The **admin** user has access to a few more options: **Labels - Add/New** and **Labels - Import**.



Label list

Name:
Type:
User:

Filter

Results (20) reload

Name	Type	User	Total	Seqs	Others
content	text	---	773	🔍	🔍
creation_date	date	---	773	🔍	🔍
creation_user	text	---	773	🔍	🔍
file	obj	admin	404	🔍	🔍
immutable	bool	---	0	🔍	🔍
internal_id	integer	---	773	🔍	🔍
length	integer	---	773	🔍	🔍
lifetime	date	---	0	🔍	🔍
name	text	---	773	🔍	🔍
perm_public	bool	---	773	🔍	🔍

1 2

Show details

Export all

Figure 2: Listing all labels

When listing labels, a page like Figure 2 should appear. The traditional filter form is available and it is also possible to filter by label type.

The label types available are:

- Integer: integer value.
- Float: float value.
- Text: text value.
- Position: pair of start and length values, representing a sequence's segment.
- Reference: pointer to another sequence.
- Taxonomy: pointer to a taxonomy.
- URL
- Bool: true/false value.
- Date: day, month and year triplet.
- Object: an uploaded file.

Still in this page, one can push the button **Export all** to export the current set of labels being listed to XML. The button **Add new** redirects to the new label page, where one can create a new label.

In the label list grid, special notes should be said about the columns:

- The column **Total** tells how many sequences contain that label.

- Clicking on the label name redirects the application to the label's page.
- The column **Seqs** creates a new search page with sequences that contain that label.
- The column **Others** creates a new search page with sequences that do not contain that label.

2.1 Creating new labels

If you are an administrator, pushing the **Add new** button a new form will be presented. The form contains all the fields needed to create a new label. They are:

- **Name**: the label's name.
- **Type**: the label's type.
- **Code**: the code to generate a new label value using the sequence information as input.
- **Validation code**: code to validate a new label instance. Should return true if the value is valid, false otherwise.
- **Modification code**: block of code run after the sequence's content is updated. Should not return anything.
- **Comment**: comment about the label.
- **Must exist**: if true, every sequence must be annotated with this label, when that doesn't happen the label goes to the sequence's missing list.
- **Generate on creation**: if true when a sequence is created, the sequence will be automatically annotated with this label using the **Code** field.
- **Generate on modification**: if true and if the sequence is already annotated with this label, the label value is automatically changed using the **Code** field when the sequence's content is altered.
- **Deletable**: if true a specific label instance can be user deleted.
- **Editable**: if true a specific label instance can be user edited.
- **Default**: if true the label will me made system default and cannot be edited thereafter.
- **Public**: if true the label can be made part of public (no login) searches.

All the code fields must be written in PHP [1].

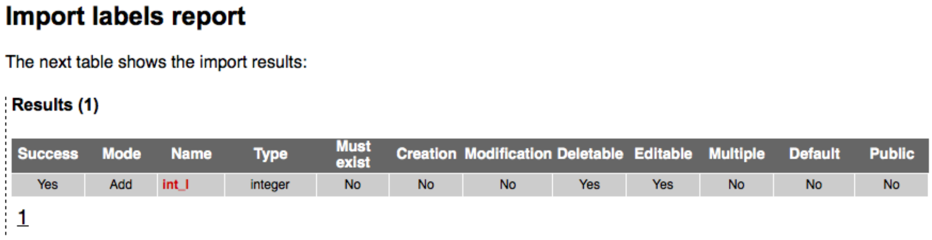
Once created, you will be redirect to the label's page, where you can view or edit information about the label. Each field can be changed by clicking on it.

Other options are present: **Delete** prompts you to delete the label, **Export** exports the label to XML and **List labels** redirects you to the label list.

2.2 Import / Export

To export all labels you can use the option **Labels - Export** from the main menu. This operation can be performed by any user.

Only the administrator is entitled to import files with labels. The option for this is **Labels - Import**. There you should upload a XML file containing labels. Once the file is processed a new report page is shown, as in Figure 3.



Import labels report

The next table shows the import results:

Results (1)

Success	Mode	Name	Type	Must exist	Creation	Modification	Deletable	Editable	Multiple	Default	Public
Yes	Add	int_l	integer	No	No	No	Yes	Yes	No	No	No

1

Figure 3: Import labels report

The **Success** column tells if the label was successfully installed into the system and the **Mode** column if the label is new to the system (mode *add*) or the label was already present and it was only updated (mode *edit*).

3 Sequences

All sequence related options can be found in the **Sequences** submenu. Three options can be found there: **List**, **Add/New** and **Batch**.

The **Sequences - List** option is used to list all system sequences. This page can also be accessed without logging in, where only the labels annotated with the label **perm_public** as true will appear. This page also features a filter form.

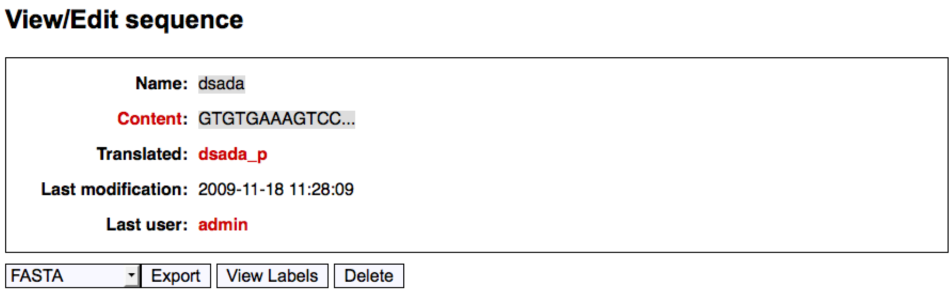
More options are available in that page, namely:

- **Export all**: exports all sequences in the grid to one of the following formats: FASTA, XML, Simple FASTA, phylip, phylip3, nexus, nexusnon, mega, meganon, paup, and paupnon.
- **Search**: launches a new search page with the sequences present in the grid.
- **Add new**: redirects to a new page, where one can insert a new sequence.

To insert a new sequence one can use the **Add new** button or the **Sequences - Add/New** option from the main menu. In the new sequence page, three fields are available: **Name**, for the sequence name; **Content**, for the sequence content and **Generate protein**, that when the sequence being insert is a DNA sequence, a protein sequence will be generated from it and the two sequences will be automatically linked using the **translated** label.

Once the sequence is inserted, the user will be redirected to the sequence page (Figure 4). In this page, basic information about the sequence is shown, namely its name, content, a link to the translated sequence, and, in the case of sub-sequences, a link to the original sequence. In this page it is also possible to export the sequence, by pushing the **Export** button.

The **Delete** button prompts you to delete the sequence and the **View labels** button redirects you the sequence's label page.



View/Edit sequence

Name: dsada

Content: GTGTGAAAGTCC...

Translated: dsada_p

Last modification: 2009-11-18 11:28:09

Last user: admin

Figure 4: Sequence page.

To edit sequence name or content just click in the respective name and content.

3.1 Labels page

Through the **View labels** button we can access the sequence's label page.

The labels page displays the sequence's annotated labels, labels available to add, missing labels and non-multiple labels that have more than one instance for this sequence, which we name the **bad multiple** labels.

The annotated labels list is always shown. The **Available** labels list is only shown if the sequence is not annotated with every system label, which is the most frequent case. The missing labels list is only shown if the sequence has not been annotated with mandatory labels, like **type** or **length**. The bad multiple labels list is only shown when, for some reason, the sequence is annotated with various instances of the same label that is not multiple, this can happen when a label, once multiple, no longer is.

An example labels page in shown in Figure 5. Some sets of labels are not shown by default and must be displayed by clicking **Show**. Filtering of labels is also available as shown in the example page.

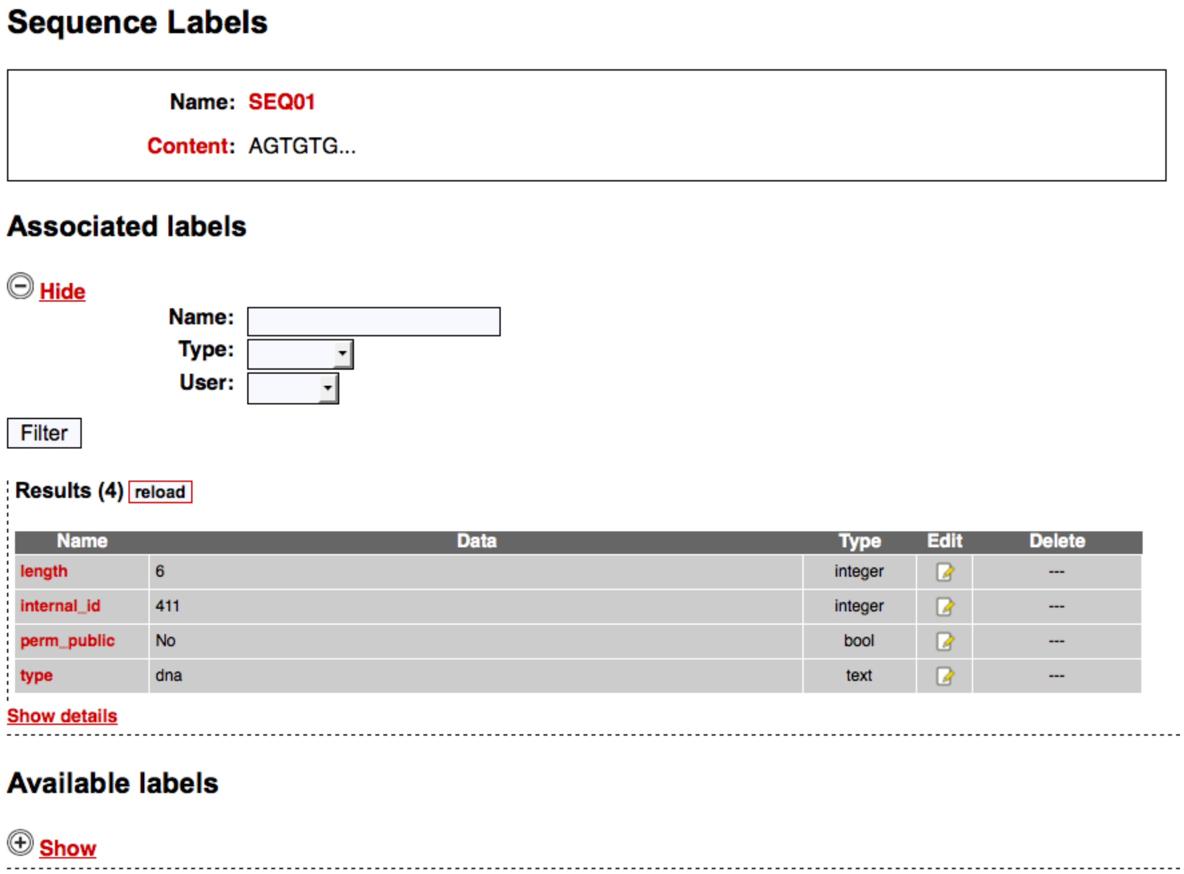
Some useful interactions were implemented: clicking in one missing label opens the available pages separator and highlights the label there, easing the process of annotating the sequence with missing labels; clicking in one bad-multiple label highlights the specific label instance in the annotated labels list.

Clicking on the **Show details** link forces the annotated labels list to display more details about the labels.

Multiple labels are shown as *label_name[parameter]*, where *parameter* is the multiple label parameter.

Clicking **Add** the icon from the first column of the **Available labels** list popups a new window that will be used to create a new label instance. To delete any annotated label, just use the icons from the **Delete** column from the same list. The **Data** column displays the label value and for label types like reference, object, taxonomy or URL, a link is rendered that redirects you to the resource.

The **Edit** icon from the annotated labels list launches a new window to edit the current label value. This window is similar to the one used to insert new label instances.



Sequence Labels

Name: SEQ01

Content: AGTGTG...

Associated labels

⊖ Hide

Name:

Type:

User:

Results (4)

Name	Data	Type	Edit	Delete
length	6	integer	✎	---
internal_id	411	integer	✎	---
perm_public	No	bool	✎	---
type	dna	text	✎	---

Available labels

Figure 5: Example labels page.

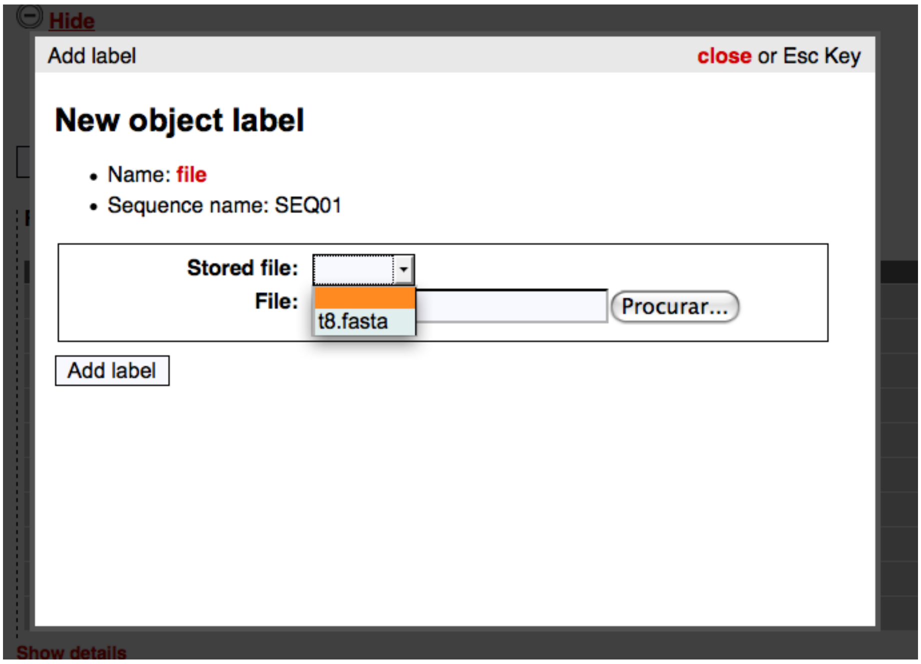
The edit or add label popup windows display forms where the user can insert or edit label values.

If the label supports automatic generation a checkbox **Generate default value** is displayed. If checked in, the result will be a new label instance generated from the label code.

For each label type, the popup window is slightly different. Next's a summary is enumerated:

- **integer**, **float**: text field with numeric validation.
- **text**: simple text field.
- **url**: text field with URL validation.
- **bool**: a checkbox.
- **position**: two text fields with numeric validation.
- **taxonomy**: a searchable grid with taxonomies that can be selected.
- **reference**: a searchable grid with sequences that can be selected.
- **date**: text field with calendar widget.

- **object** (Figure 6): if the label has files attached a select list is displayed containing them. An upload field is also available.



Add label

New object label

- Name: file
- Sequence name: SEQ01

Stored file:

File:

t8.fasta

Figure 6: Annotating a new object label.

Once a label instance is added or edited the labels page is updated to reflect the changes.

3.2 Batch

Instead of inserting sequences one by one using the page described above, you can upload a file with several sequences in either XML or FASTA format. These files can be also annotated with labels as described in the section 8.

When uploading the file, the system will try to insert or update the stored sequences. A sequence is only updated if the application can found a previously stored sequence with the same name and content, everything else is treated as a new sequence.

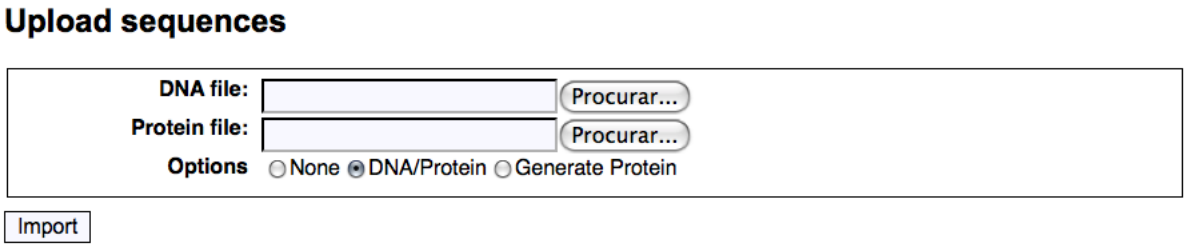
To access this functionality use the **Sequences - Batch** main menu option.

In this page (Figure 7) three options are available: upload a single file; upload both DNA and protein file, linking the sequences along the process; upload a DNA file and also generate protein sequences.

When the file is being processed a loading screen appears displaying the progress. When everything is done a report page like the Figure 8 should appear.

If the **none** option was chosen, only a list of sequences is shown, for everything else, a list of DNA and protein sequences are shown.

If the files sent were annotated with labels, a label report is shown, indicating the state for each label. For example, if a label present in the file is not installed in the system, a warning indicating the label is not installed will be shown. Only previously installed labels will be used when importing label values.



Upload sequences

DNA file: [ ] Procurar...

Protein file: [ ] Procurar...

Options ○ None ● DNA/Protein ○ Generate Protein

Import

Figure 7: Upload sequences page.

In each sequence list the **New** column indicates if the sequence is new and was inserted, or if it is old and it is being updated. The **Comment** label displays various kinds of informations and can tell when the sequence was only updated.

If the file was annotated with labels, a column named **Status** will appear in the sequence list. Clicking on the green arrow will popup a window with a grid, indicating for each label, the status for this sequence. For example, if a label is automatically generated when a sequence is created and its value was specified in the file, the label will not be updated and the text **Already inserted** will be shown.

Clicking on the **batch manipulated** link will redirect you to a new search page with only the imported sequences, which is useful to run various operations for all those sequences in batch mode.

4 Taxonomies

To manage taxonomies we should pay attention to three things: trees, taxonomy ranks and taxonomies.

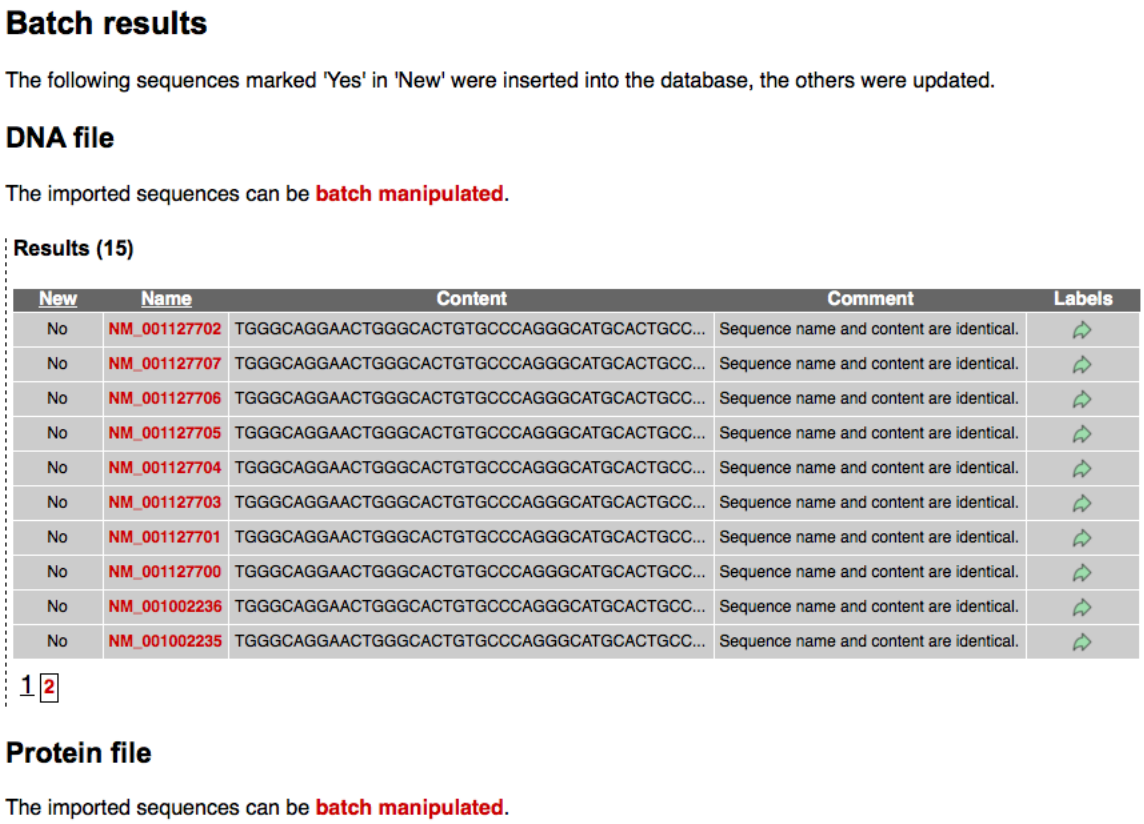
With taxonomy trees one can have multiple trees of taxonomies, which is useful to have custom taxonomies and more scientific trees like the NCBI taxonomy tree.

Ranks is one way to categorize taxonomies. The system installs a rich set of ranks, with parent/child relationships already defined.

To access taxonomy related features, use the **Taxonomies** submenu from the main menu.

4.1 Managing trees

To list the currently defined taxonomy trees, select **Taxonomies - Trees - List**, there you should at least see the NCBI tree (Figure 9).



Batch results

The following sequences marked 'Yes' in 'New' were inserted into the database, the others were updated.

DNA file

The imported sequences can be **batch manipulated**.

Results (15)

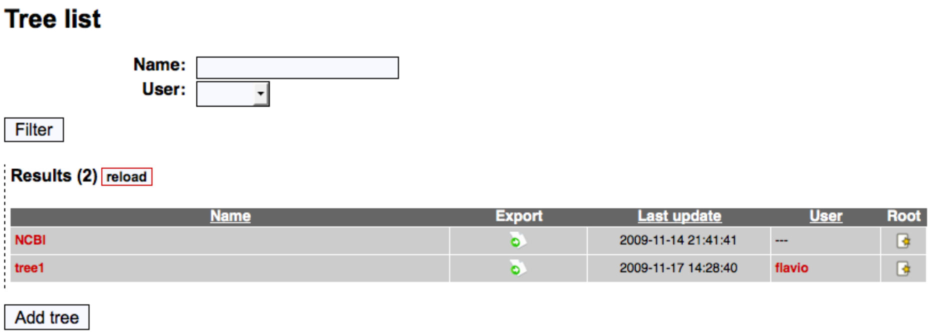
New	Name	Content	Comment	Labels
No	NM_001127702	TGGGCAGGAACTGGGCACTGTGCCCAGGGCATGCACTGCC...	Sequence name and content are identical.	➦
No	NM_001127707	TGGGCAGGAACTGGGCACTGTGCCCAGGGCATGCACTGCC...	Sequence name and content are identical.	➦
No	NM_001127706	TGGGCAGGAACTGGGCACTGTGCCCAGGGCATGCACTGCC...	Sequence name and content are identical.	➦
No	NM_001127705	TGGGCAGGAACTGGGCACTGTGCCCAGGGCATGCACTGCC...	Sequence name and content are identical.	➦
No	NM_001127704	TGGGCAGGAACTGGGCACTGTGCCCAGGGCATGCACTGCC...	Sequence name and content are identical.	➦
No	NM_001127703	TGGGCAGGAACTGGGCACTGTGCCCAGGGCATGCACTGCC...	Sequence name and content are identical.	➦
No	NM_001127701	TGGGCAGGAACTGGGCACTGTGCCCAGGGCATGCACTGCC...	Sequence name and content are identical.	➦
No	NM_001127700	TGGGCAGGAACTGGGCACTGTGCCCAGGGCATGCACTGCC...	Sequence name and content are identical.	➦
No	NM_001002236	TGGGCAGGAACTGGGCACTGTGCCCAGGGCATGCACTGCC...	Sequence name and content are identical.	➦
No	NM_001002235	TGGGCAGGAACTGGGCACTGTGCCCAGGGCATGCACTGCC...	Sequence name and content are identical.	➦

1 2

Protein file

The imported sequences can be **batch manipulated**.

Figure 8: Batch sequence report.



Tree list

Name:
User:

Results (2)

Name	Export	Last update	User	Root
NCBI	●	2009-11-14 21:41:41	---	▣
tree1	●	2009-11-17 14:28:40	flavio	▣

Figure 9: Tree listing.

The user has the possibility to filter the tree list using a tree name or the user who made the last update.

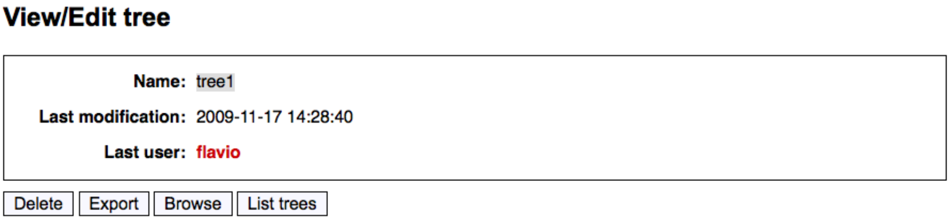
From here you can select a tree to view or edit or select **Add tree** to create a new tree. To create a new tree you can also use **Taxonomies - Trees - Add/New**.

Each tree can also be exported to a XML file using the green **Export** button. The yellow **Root** button helps creating a new root taxonomy for this tree.

Selecting **Add tree**, you will be prompted for the tree name. Once created, you will be redirected to the tree's page, as in Figure 10.

In it you can see history information and also edit the tree name by clicking on it. There is also four operation buttons:

- List trees: redirects you back to the tree list.
- Export: exports the tree to a XML file.
- Delete: prompts you to delete the tree (not available for the NCBI tree).
- Browse: enables you to browse the tree.



View/Edit tree

Name: tree1
Last modification: 2009-11-17 14:28:40
Last user: flavio

Figure 10: Viewing a tree.

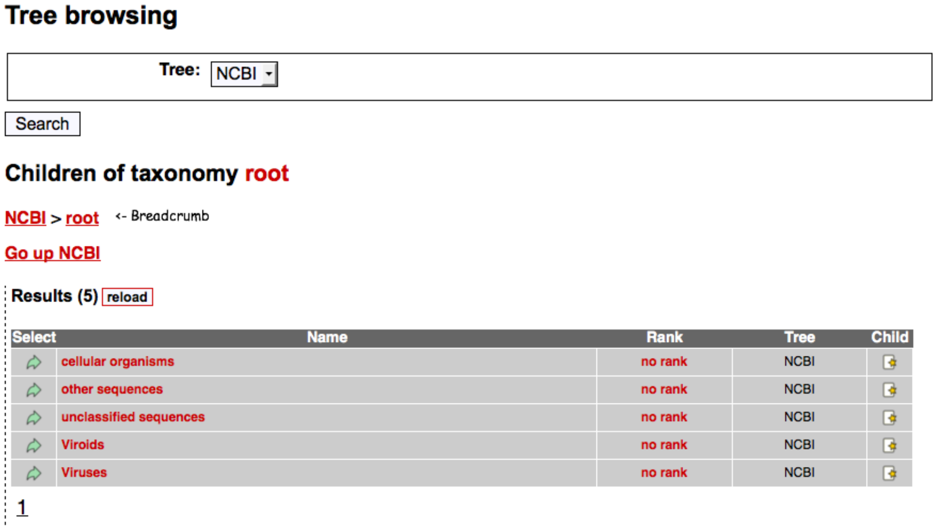
If you are an administrator you can make use of the option **Taxonomies - Trees - Import** to upload a taxonomy tree XML file. Once the file is uploaded, the system tries to insert all taxonomies from that tree into the system. If some taxonomies are found, updates are done.

4.1.1 Tree browsing

The tree browser page provides an easy to use interface to navigate through the target tree.

In Figure 11 we are navigating the NCBI tree, currently at node **root** and the grid is filled with **root**'s children.

We can go up in the hierarchy by clicking **Go up *node name***. To navigate into a taxonomy we click the green arrow in the select column for that taxonomy. We can also add a new taxonomy child, selecting the last icon from the **Child** column, it redirects us to a new taxonomy form, with the parent taxonomy already setup.



Tree browsing

Tree: NCBI

Search

Children of taxonomy root

NCBI > root <- Breadcrumb

Go up NCBI

Results (5) reload

Select	Name	Rank	Tree	Child
➦	cellular organisms	no rank	NCBI	📄
➦	other sequences	no rank	NCBI	📄
➦	unclassified sequences	no rank	NCBI	📄
➦	Viroids	no rank	NCBI	📄
➦	Viruses	no rank	NCBI	📄

1

Figure 11: Browsing the NCBI tree.

Another important aspect is the breadcrumb component (in the example **NCBI** > **root**). Clicking in one breadcrumb name will make the current taxonomy change.

4.2 Managing ranks

To list all system ranks use the option **Taxonomies - Ranks - List** from the main menu. This page gives you a filter enabled rank list, being possible to filter the rank list by rank name, parent rank name or the user who made the last rank change.

You can also order the rank names by alphabetical order, ascending or descending.

To export the current set of ranks just push the **Export all** button.

For each rank in the grid, there are two action icons: one named **Taxonomy**, that redirects us to the new taxonomy form where the rank is already selected, and the other, **Child** opens a new page with a form to create a new rank with the parent rank already set.

To create ranks, there's also the **Taxonomies - Ranks - Add/New** menu option. On this page, you should input the rank name and select the parent rank from the list of inserted ranks.

Once a rank is created, the system redirects you to the rank page where you can edit the rank name or parent rank by clicking in the respective name. Editing by clicking on the name, changes the rank name to a text field, where you can input the new name and then, when pushing the **OK** button, sends the changes back to the server.

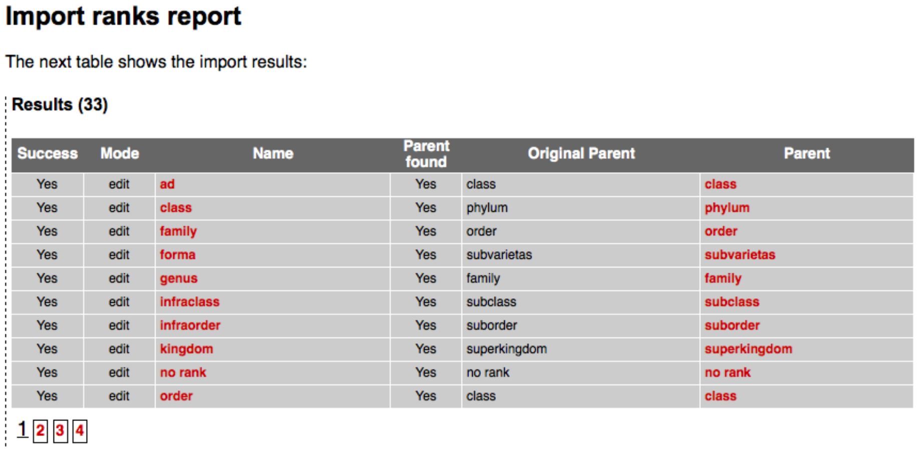
Also, when on the rank page, there are two buttons at the end of the page: **Delete** prompts you to delete the rank and **List ranks** redirects you back to the rank list.

4.2.1 Import / Export

When logged in as **admin** you can export all system ranks through the option **Taxonomies - Ranks - Export**. The output file is XML.

The other way around, you can import a XML file with ranks, through the option **Taxonomies - Ranks - Import**. Once the file is processed by the server a page like Figure 12 is shown.

The column **Success** tells if the new rank was installed into the system or not, the column **Mode** indicates if the new rank was added or edited. The **Parent found** column tells if the parent was found in the system (if the system can't find it, it is created). The column **Original parent** was the rank parent before the import operation if the rank was already in the system and the column **Parent** just indicates the new parent rank.



Import ranks report

The next table shows the import results:

Results (33)

Success	Mode	Name	Parent found	Original Parent	Parent
Yes	edit	ad	Yes	class	class
Yes	edit	class	Yes	phylum	phylum
Yes	edit	family	Yes	order	order
Yes	edit	forma	Yes	subvarietas	subvarietas
Yes	edit	genus	Yes	family	family
Yes	edit	infraclass	Yes	subclass	subclass
Yes	edit	infraorder	Yes	suborder	suborder
Yes	edit	kingdom	Yes	superkingdom	superkingdom
Yes	edit	no rank	Yes	no rank	no rank
Yes	edit	order	Yes	class	class

Figure 12: Import rank report

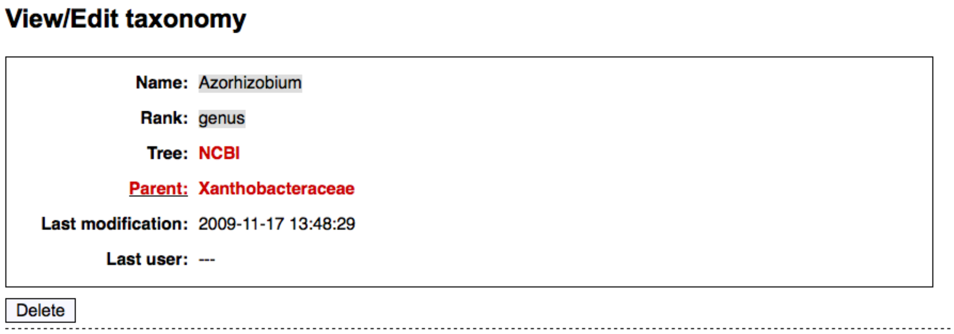
4.3 Managing taxonomies

We have seen that there are lots of ways to get to the new taxonomy form. The standard way is to choose the option **Taxonomies - Add/New** from the main menu. There you should enter the taxonomy name, choose the rank and tree from a list of stored ranks and trees, respectively.

The rank and tree can be left empty, but as a recommendation, you should define them right from the beginning.

Once a taxonomy is created, the system redirects you to the taxonomy page. Each taxonomy page is composed of: a form where you can edit basic taxonomy information (Figure 13), a list of optional taxonomy names (Figure 14) and a list of children taxonomies.

In the first part (Figure 13), you can edit the name and rank by clicking in the respective name. Changing the tree is not allowed. To change the parent you should click the red **Parent:** link and when a window popup appears you should search for your parent taxonomy, select the name from the grid and then push the **Select** button.



View/Edit taxonomy

Name: Azorhizobium

Rank: genus

Tree: NCBI

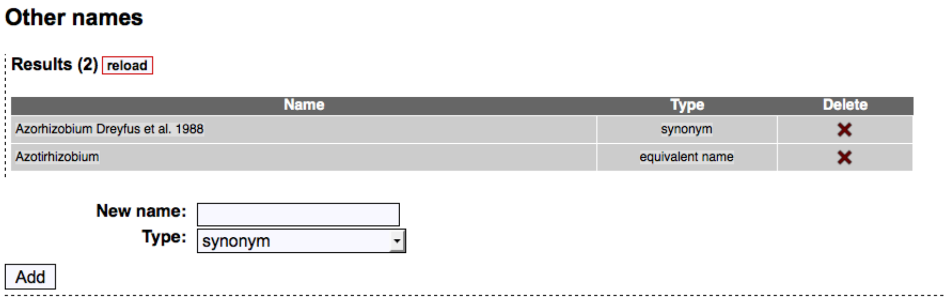
Parent: Xanthobacteraceae

Last modification: 2009-11-17 13:48:29

Last user: ---

Figure 13: Editing taxonomy information.

The **Other names** section provide a list of other names for the taxonomy. Each name can be deleted using the **Delete** column. To edit a name you should click on the name cell and then push the **OK** button. To edit the name type the process is identical. To add a name, just use the provided form.



Other names

Results (2)

Name	Type	Delete
Azorhizobium Dreyfus et al. 1988	synonym	✗
Azotirhizobium	equivalent name	✗

New name:

Type: synonym

Figure 14: Other names section.

The final section displays a grid with the children taxonomy. You can add new ones to the list by pushing the **Add child** button.

As the number of taxonomies can get very large, we provided a page where one can search by taxonomies by just using the name, tree or rank, or any combination of these.

To use this interface go to **Taxonomies - Browse**.

5 Search

To search sequences using the annotated label information, one can use the search pages available from the main menu. Three pages are available:

- **Search - ALL**: to search all sequences.
- **Search - DNA**: search only DNA sequences.
- **Search - Protein**: search only protein sequences.

All the three search pages look the same, so everything we will describe for the rest of this section will apply for all of them.

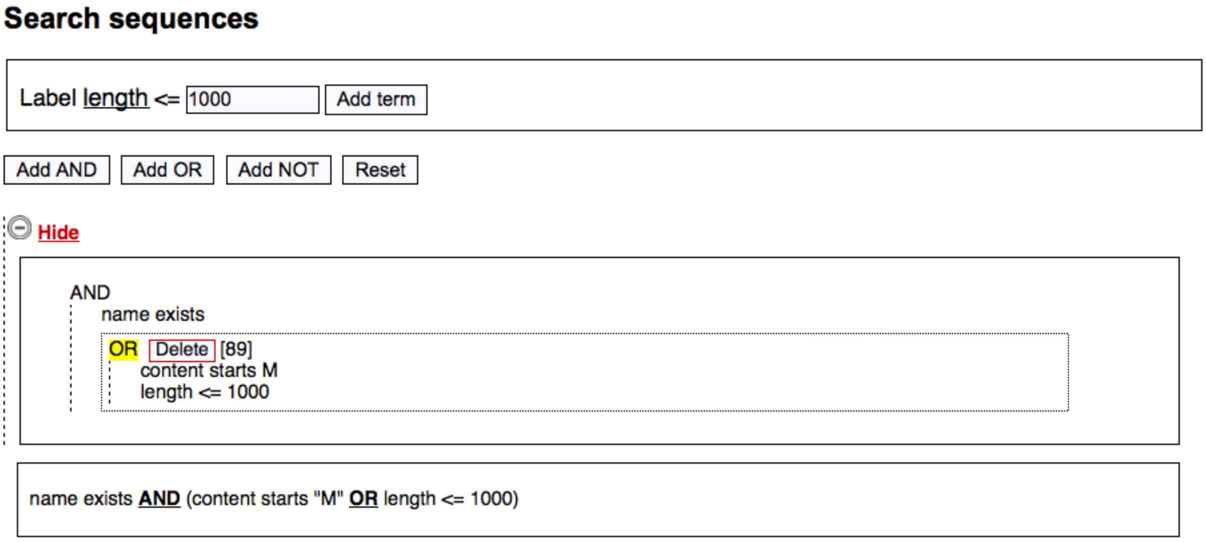
Accessing any search page, we will rapidly discover three main sections in this page: the query input section, the operations section and the preview section.

5.1 Query input

The query input section (as shown in Figure 15) presents you controls to display and manipulate the query expression.

The query is presented in two formats: the tree, where you can select parts of the expression and preview in real time how many sequences are filtered using that sub-expression and the query text, where the query is presented in an human readable format. In the query tree view, apart from selecting where to insert sub-expressions, you can also delete parts of the expression by selecting an AND, OR, NOT or sub-expression and pressing **Delete**.

To insert a new query expression, you should choose where the expression will be put. That is possible selecting one of the previously inserted OR, AND or NOT expressions. If you simply want to create a simple AND query expression, press the **Reset** button and start inserting expressions. But if you want to create complex queries you should know how to insert expressions at different positions.



Search sequences

Label length <= 1000

```
AND
  name exists
  OR Delete [89]
    content starts M
    length <= 1000
```

name exists **AND** (content starts "M" **OR** length <= 1000)

Figure 15: Search query input.

When creating a new sub-expression, the process involves various steps:

- First, input the label name you want to use for this term in the **Label** field. The system will autocomplete the label names as you type.
- When a label is chosen, you must select the search operator. Please see the table 1 for information about each operator.
- After the operator is chosen, you must, optionally, input the value for the operator.

  In resume, for each label type:

  - Text, URL and Object: Text field.

- - Integer, position and float: Numeric text field.
  - Bool: Checkbox.
  - Date: The date is input in a calendar widget. Please click on the text field to activate it.
  - Taxonomy: Text field for the **like** operator. For the **equal** operator you should click on the **Find taxonomy** link, search a taxonomy within the popup window and then click on the chosen taxonomy name.
  - Reference: Text field for the **like** operator. For the **equal** operator you should click on the **Find sequence** link, search a sequence within the popup window and then click on the chosen sequence.

- Press the **Add term** button. The new term should appear on the query views, and the preview section will be automatically updated.

Please remember that you can use the **exists** and **not exists** operators. These operators will filter sequences that are annotated with a label or not, respectively, and do not need values.

If the label is multiple, you can, optionally, input the multiple parameter. If the multiple parameter is not given, the search is done for all label instances. If given, the query will only use the specific label instance.

You can also insert AND, OR and NOT terms. Use the respective buttons.

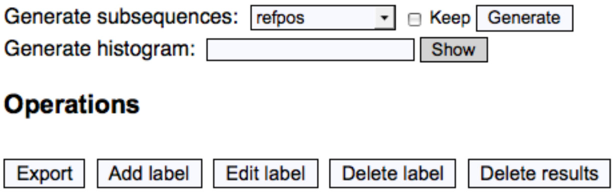
Label type	**Operators**
URL, text and object	• equal: Equal comparison. • contains: If the label contains a substring. • starts: If the instance starts with. • ends: Starts counterpart. • regexp: Regular expression matching.
Bool	• equal: Equal comparison.
Integer and float	• = • > • < • >= • <=
Position	• = • > • < • >= • <= You should also select the position component, start or length for the term.
Date	• equal: Equal comparison. • before: Date is before some date. • after: Date is after some date.
Taxonomy	• equal: Equal comparison. • like: A taxonomy name to search for. Using this operator will make system search for all taxonomies in the database with this name and then the query will match if a sequence points to any of them.
Reference	• equal: Equal comparison. • like: A sequence name to search for. Works the same way as for taxonomies.

Table 1: Label types and operators.

5.2 Operations

While you create your complex query you can see the preview list getting shorter and shorter, giving you immediate feedback. And now that you have your results, you can do things with them.

The operations section as it can be seen in Figure 16, contains various operations you can do to the search result list.



Generate subsequences: refpos □ Keep

Generate histogram:

Figure 16: Search operations.

5.3 Sub-sequences

To begin, we can generate sub-sequences using a position label from the result list. For example, if your sequences contain a position label for a specific sequence segment, you can generate sub-sequences for the complete set of sequences. To do that, you must select the position label from the **Generate subsequences** select list and then push the **Generate button**. If you want to keep your sub-sequences around longer than a few days, you should check the **Keep** checkbox. When not keeping the sequences around, the system will delete them after some hours.

Once the sub-sequences are generated, a report page will be shown.

5.4 Histograms

Another operation is the **Generate histogram** (Figure 17). This option can generate an histogram for that label distribution across the result list. For example, you could generate a length distribution for a given sequence set and then the system will generate the frequency histogram and display the distribution total and number of classes. For numeric labels the smallest class, largest class, average, median and mode values are also shown.

If your label is multiple and numeric, you can chose what value will be representative for each sequence. You can use the average value for all label instances from a sequence, the minimum or the maximum value. If the label is not numeric but multiple, all values will be considered.

In the popup window that appears when generating the histogram, you can also copy the distribution values to use with programs like Microsoft Excel.

5.5 Export

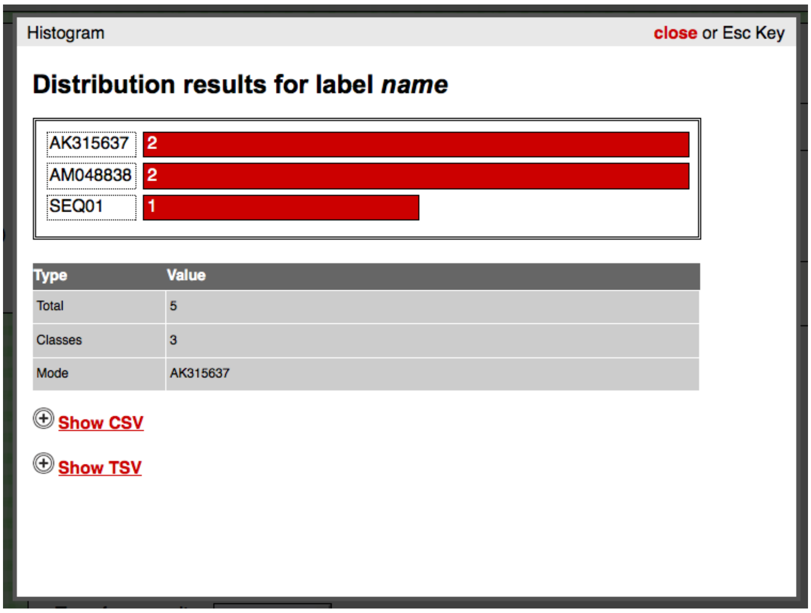
Another important operation is the **Export** button. It can export your result sequences (Figure 18) into the file formats mentioned before and supported by the system.

For formats like FASTA or XML, you can select the labels that will appear on the file, only exporting annotation that is important to the task at hand.

5.6 Batch labels

Another common action to do is, for example, annotate a list of sequences with the same label instance. For this you can use the button **Add label**, to add, or the button **Edit label**, to edit.

Using the **Add label** option you will be redirected to a page that looks like Figure 19. First you should enter the label name into the **Label** text field, optionally you can check the



Distribution results for label *name*

Type	Value
Total	5
Classes	3
Mode	AK315637

Figure 17: Plotting histograms.

Update checkbox. When checked and if the sequence already contains that label instance, the current value will updated; if not checked, nothing will be done.

When ready press the **Next...** button. A popup window should appear. This window is very similar to the one we used to add a label to a sequence, so, it works the same way.

Once you input the label value, a report on the process should appear, right below the **Next...** button.

The **Edit label** operation works in a similar fashion, but instead of having the **Update** checkbox, it contains the **Add new** checkbox. This checkbox, when checked and when the sequence does not contain the label, forces the system to add a new label value. It is the counterpart of the **Update** option, but for the designed for the edit mode.

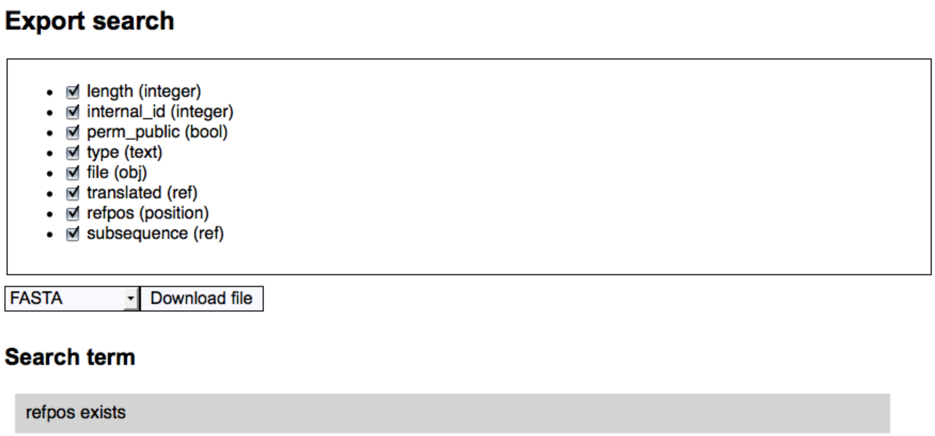
There is still another button, the **Delete label** button. This button gives you the possibility to delete a label from the set of sequences. After you put the label, press the **Next...** button and answer **Yes** and all annotations related to that label will disappear from the sequence list.

5.7 Delete

If you want to delete your sequence list just select the button **Delete** and everything will be deleted. The action is irreversible, so use it with care!

5.8 Preview

The last section (Figure 20) present in the search page is the preview section. In it we can see the results (as a list of sequences) for the query being build.



Export search

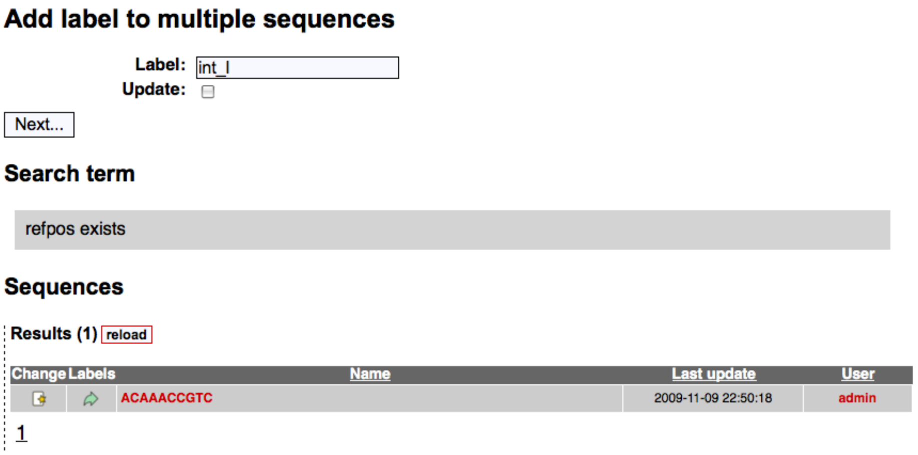
- ☑ length (integer)
- ☑ internal_id (integer)
- ☑ perm_public (bool)
- ☑ type (text)
- ☑ file (obj)
- ☑ translated (ref)
- ☑ refpos (position)
- ☑ subsequence (ref)

FASTA | Download file

Search term

refpos exists

Figure 18: Export search page.



Add label to multiple sequences

Label: int_l
Update: □

Next...

Search term

refpos exists

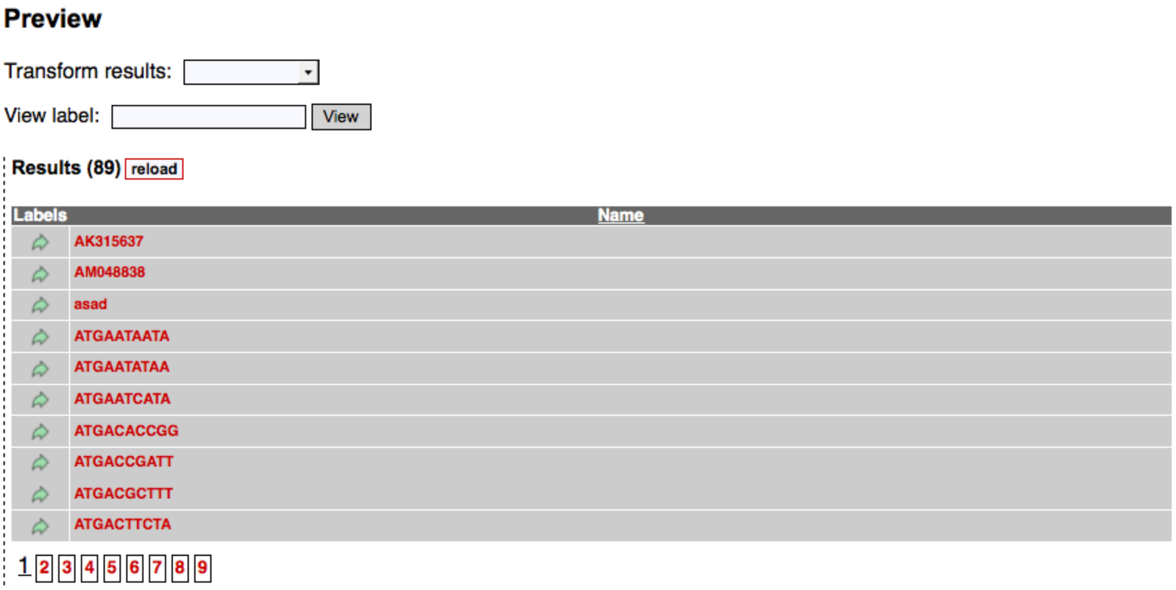
Sequences

Results (1) reload

Change	Labels	Name	Last update	User
		ACAAACCGTC	2009-11-09 22:50:18	admin

1

Figure 19: Add label to multiple sequences.



Preview

Transform results:

View label: View

Results (89) reload

Labels	Name
	AK315637
	AM048838
	asad
	ATGAATAATA
	ATGAATATAA
	ATGAATCATA
	ATGACACCGG
	ATGACCGATT
	ATGACGCTTT
	ATGACTTCTA

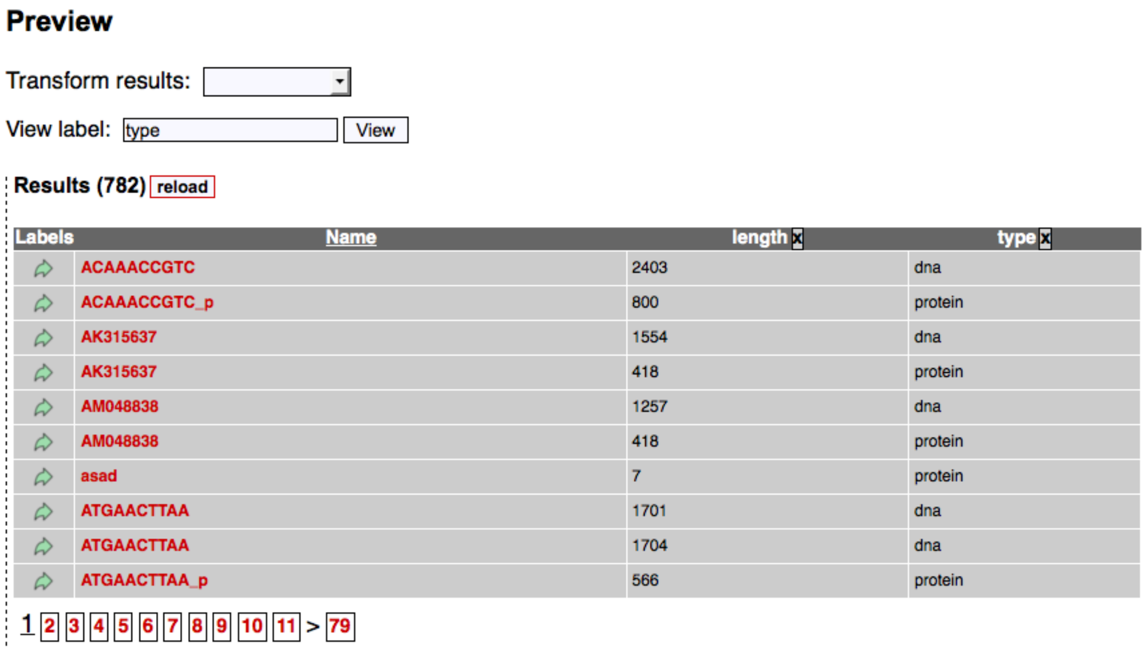
1 2 3 4 5 6 7 8 9

Figure 20: Search preview results.

One important option in this section is **Transform results**. Here you can select a reference label, and the current results will be transformed using the annotated reference label in each sequence. If not all sequences are annotated with that label, the new result set will be smaller than the original. If the reference label is multiple, the new result set can be potentially bigger.

Please note that the new, transformed, set will be used for the batch operations mentioned before.

The other option, **View label**, can add new label columns to the result grid, showing the label values for each sequence, as shown in Figure 21.



Preview

Transform results:

View label: type View

Results (782) reload

Labels	Name	length x	type x
↪	ACAAACCGTC	2403	dna
↪	ACAAACCGTC_p	800	protein
↪	AK315637	1554	dna
↪	AK315637	418	protein
↪	AM048838	1257	dna
↪	AM048838	418	protein
↪	asad	7	protein
↪	ATGAACTTAA	1701	dna
↪	ATGAACTTAA	1704	dna
↪	ATGAACTTAA_p	566	protein

1 2 3 4 5 6 7 8 9 10 11 > 79

Figure 21: Using view label.

Each new label column added can be removed by clicking the **X** on the column header.

5.9 Written queries

Instead of using the query input section from the search page, you can use the search field on the top of each page to input arbitrarily complex queries.

The search field can also search for anything in the system: labels, taxonomies, ranks and sequences. When the specified search expression is not valid, the system will fallback to a wide search for the former objects, displaying a page with a grid for each entity where the query matched. So, for example, if you input 'homo sapiens', the system will display a grid with taxonomies with 'homo sapiens' in the name.

But the more interesting case is when using valid query expressions. A query expression is composed of terminal expressions and composed expressions.

A terminal expression contains a label name, an operator and, optionally, a value. The following expression is a terminal expression: *length > 500*. The operators that do not need a value, are the **exists** and **notexists** operators, like *species exists*. For everything else the form is *label_name operator value*. Operators and value information is shown in table 2.

Another note: if you want to write values with spaces, wrap the value around ' or ".

Now, composed expressions can combine various other expressions, recursively, and use the special operators: AND, OR or NOT. AND and OR can be used like *expression1 AND expression2 AND expression3 ...*. The NOT operator can only have an expression as argument, like this: *NOT expression*.

You can also use parenthesis to group expressions. So for example, you can have things like: *<(length > 500 and name exists) or content regexp AGTG>*.

Once you input the expression, the system will analyze it and build a tree for it, redirecting you to the search page, where you can run batch operations against the resulting sequences.

6 File formats

For file format information please read the section 8 from the main technical report.

7 Administration

There are a few functionalities to do maintenance or administration tasks. These tasks can only be used when logged in as **admin**.

7.1 User management

One of the key areas in administration is user management. Lets see how we can add new users, update user settings and so forth.

7.1.1 New user

To create a new user select **Administration - Users - Register** and input the user information. The username field is the name that should be used to login. You must also input the user's password twice. Click **Do register**. If everything went well, a new user has been registered and a list of all system users is shown.

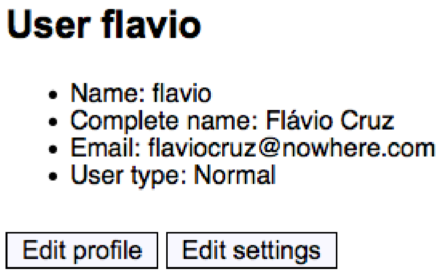
This list can be accessed through **Administration - Users - List**.

Create more users as needed. You can also logout and try the new registered users if you want.

7.1.2 User settings

Given that only administrators can register new users, normal users can modify their information, namely, the complete name, password, email and other settings.

To edit profile data, click on the user's name, below the main menu. The new page presents user information and two buttons, as shown in Figure 22.



User flavio

- Name: flavio
- Complete name: Flávio Cruz
- Email: flaviocruz@nowhere.com
- User type: Normal

Figure 22: Page with user information.

Selecting **Edit profile** enables us to edit the complete name, email or password. When editing any information here, you should input your old password. If you want to change the current password fill the two text fields for that, if not, leave them empty.

The other button, **Edit settings** enables you to change other, aspect related settings, like the number of items per grid.

7.1.3 Managing other users

If you are an administrator using the **admin** account, you can edit other user profiles, through **Administration - Users - List** and then selecting the target user. When editing one user you should enter the **admin** password and not the user's current password.

For some reason, if you want to disable one user, go to the users list, select the user name and click the **Delete** option.

One very destructive feature is the **Database reset**. It can be accessed through the menu option **Administration - Reset Database** and removes all custom data from the database, which is:

- All taxonomy trees, except the NCBI tree.
- All ranks except the system defaults.
- All non-default labels.
- All sequences.
- All normal users.
- All files in the **file** table.

7.2 Import / Export database

Another useful feature is the database export / import facilities. One can export the whole database as a XML file and then import it somewhere else, literally copying the source database.

What is exported?

- labels

- ranks
- taxonomy trees, except the NCBI tree
- sequences

To export the database, use the option **Administration - Export Database**.

To import a database XML file, go to **Administration - Import Database** from the main menu. There you should upload the file and, once processed, an import report is shown. The report is similar to the individual ones, but over various entities.

7.3 Application customization

There are two ways of customizing the application:

- Changing the text that appears on the header: use the option **Administration - Database Description**.
- Changing the application's background: use the option **Administration - Database Background**. You should upload an JPG or PNG file.

8 File formats

We use files to import or export data in order to interchange information between different kinds of systems or among various instances of the application. These files can be in two different kinds of formats: XML or FASTA.

Among other things, those files are used throughout the system to: copy entire databases, import sequences, install new labels, import whole taxonomy trees or heterogenous integration.

8.1 FASTA

The FASTA format [2] is very well known in the bioinformatics field as it is used to store a specific set of DNA or protein sequences.

In our system, this format is used to export stored sequences or to import new ones.

We have designed two FASTA-like formats:

- Plain format

  In the plain format we just store the sequence name followed by its content.

- Complex format

  In this format we also store label instance information along the sequence data.

  The format starts by including one line comment, followed by a line telling which labels are included for each sequence. Those labels are separated by the character '|'.

Listing 1: Plain FASTA format.

```
>AK315637
ELRLRYCAPAGFALLKCNDADYDGFKTNCSNVSVVHCTNLMNTTVTTGLLLNGSYSENRT
QIWQKHRTSNDSALILLNKHYNLTVTCKRPGNKTVLPVTIMAGLVFHSQKYNLRLRQAWC
HFPSNWKGAWKEVKEEIVNLPKERYRGTNDPKRIFFQRQWGDPETANLWFNCHGEFFYCK
MDWFLNYLNNLTVDADHNECKNTSGTKSGNKRAPGPCVQRTYVACHIRSVIIWLETISKK
TYAPPREGHLECTSTVTGMTVELNYIPKNRTNVTLSPQIESIWAAELDRYKLVEITPIGF
APTEVRRYTGGHERQKRVPFVXXXXXXXXXXXXXXXXXXXXXXVQSQHLLAGILQQQKNL
LAAVEAQQQMLKLTIWGVK
<...more sequences...>
```

Listing 2: Complex FASTA format example.

```
;flavio - Monday 19th October 2009 07:06:33 PM - sequence id 465
#name|length|internal_id|perm_public|type|translated|url
>AK315637|1554|465|0|dna|AK315637_p|[google -> http://google.pt Â§ ncbi -> http
    ://www.ncbi.nlm.nih.gov/]
ELRLRYCAPAGFALLKCNDADYDGFKTNCSNVSVVHCTNLMNTTVTTGLLLNGSYSENRT
QIWQKHRTSNDSALILLNKHYNLTVTCKRPGNKTVLPVTIMAGLVFHSQKYNLRLRQAWC
HFPSNWKGAWKEVKEEIVNLPKERYRGTNDPKRIFFQRQWGDPETANLWFNCHGEFFYCK
MDWFLNYLNNLTVDADHNECKNTSGTKSGNKRAPGPCVQRTYVACHIRSVIIWLETISKK
TYAPPREGHLECTSTVTGMTVELNYIPKNRTNVTLSPQIESIWAAELDRYKLVEITPIGF
APTEVRRYTGGHERQKRVPFVXXXXXXXXXXXXXXXXXXXXXXVQSQHLLAGILQQQKNL
LAAVEAQQQMLKLTIWGVK (...)
<...more sequences...>
```

For each sequence line we put all the label instances separated by the character '|'. The order of the label instances must be equal to the label's order at the file's header.

If the sequence does not have a specific label instance the string in that column should be empty "".

If the label instance value is empty and that label is not editable and can be generated from code it will be automatically generated when imported.

For multiple labels, the label value is enclosed by square brackets '[]' and each instance, represented as *param* -> *value*, is separated by the character 'Â§'.

The special label 'name' is treated like any other label. If it is not included in the label's header, the first 10 sequence's content characters will be used by omission.

An example of this format can be seen in Listing 2.

Listing 3: An example Label XML file.

```
<labels>
  <label>
      <name>length</name>
      <type>integer</type>
      <comment></comment>
      <default>1</default>
      <must_exist>1</must_exist>
      <auto_on_creation>1</auto_on_creation>
      <auto_on_modification>1</auto_on_modification>
      <code>return strlen($content);</code>
      <valid_code>return $data &gt; 0;</valid_code>
      <editable>0</editable>
      <deletable>0</deletable>
      <multiple>0</multiple>
      <public>1</public>
   </label>
   <...more labels...>
</labels>
```

8.2 XML

The XML format is widely used to export and import lots of different kinds of data throughout the system. This format can handle labels, sequences, taxonomy trees, ranks and the database itself.

- **Labels**

  Using the XML format we can export a set of labels. This file can then be imported in another system resulting in label installation or update.

  An example of this kind of file is shown in Listing 3 and as it can be seen, we store each label property as a XML tag.

  All the rules concerning empty label instances from the complex FASTA format are also present in this format.

- **Sequences**

  Besides the FASTA format, sequences can also be stored in XML files. The main difference between the FASTA format is that, given the structured and flexible nature of XML, it is easier to describe the sequence contents and its label instances.

  The same sequence represented in FASTA (Listing 2) can be seen formatted as XML in Listing 4.

- **Ranks**

  To manage ranks across multiple application instances we designed a XML format to store taxonomy ranks.

  As it can be seen in Listing 5, for each rank we register its name and parent rank. This type of files is useful to copy rank sets around systems.

Listing 4: An Sequence XML file.

```
<sequences>
<author>flavio</author>
<date>Tuesday 20th October 2009 12:59:53 AM</date>
<what>sequence id 465</what>
<labels>
   <label>length</label>
   <label>internal_id</label>
   <label>perm_public</label>
   <label>type</label>
   <label>translated</label>
   <label>url</label>
</labels>
<sequence>
   <name>AK315637</name>
   <content>ELRLRYCAPAGFALLKCNDADYDGFKTNCSNVSVVHCTNLMNTTVTTGLLLNGSYSENRT
  QIWQKHRTSNDSALILLNKHYNLTVTCKRPGNKTVLPVTIMAGLVFHSQKYNLRLRQAWC
  HFPSNWKGAWKEVKEEIVNLPKERYRGTNDPKRIFFQRQWGDPETANLWFNCHGEFFYCK
  MDWFLNYLNNLTVDADHNECKNTSGTKSGNKRAPGPCVQRTYVACHIRSVIIWLETISKK
  TYAPPREGHLECTSTVTGMTVELNYIPKNRTNVTLSPQIESIWAAELDRYKLVEITPIGF
  APTEVRRYTGGHERQKRVPFVXXXXXXXXXXXXXXXXXXXXXXVQSQHLLAGILQQQKNL
  LAAVEAQQQMLKLTIWGVK (...)</content>
   <label name="length">1554</label>
   <label name="internal_id">465</label>
   <label name="perm_public">0</label>
   <label name="type">dna</label>
   <label name="translated">AK315637_p</label>
   <label name="url" param="google">http://google.pt</label>
   <label name="url" param="ncbi">http://www.ncbi.nlm.nih.gov/</label>
</sequence>
</sequences>
```

Listing 5: An example Rank XML file.

```
<ranks>
   <rank>
      <name>class</name>
      <parent>phylum</parent>
   </rank>
   <rank>
      <name>family</name>
      <parent>order</parent>
   </rank>
   <...more ranks...>
</ranks>
```

Listing 6: An example Taxonomy tree XML file.

```
<tree>
   <name>example</name>
   <nodes>
      <taxonomy>
         <name>root_taxonomy</name>
         <rank>family</rank>
         <taxonomy>
            <name>child_taxonomy</name>
            <rank>genus</rank>
         </taxonomy>
         <taxonomy>
            <name>child_taxonomy2</name>
            <rank>genus</rank>
         </taxonomy>
      </taxonomy>
   </nodes>
</tree>
```

- **Taxonomy trees**

  We designed a XML format to store taxonomy trees, which is very useful to easily copy an entire taxonomy tree from one system to another.

  In this format, we store the tree name followed by a 'nodes' tag which will store, starting by the root taxonomies, the taxonomies from this tree. Each 'taxonomy' tag may contain an arbitrary number of 'taxonomy' tags which represent taxonomy's children.

- **Database**

  We designed another XML based format, this time to store the entire database. The skeleton for this format is presented in Listing 7 and it is organized as follows:

  - **labels**

    This section is exactly the same as the Label XML file.

  - **ranks**

    Idem, but for ranks.

  - **trees**

    A special tag containing all the taxonomy trees. Each tree is represented the way it is shown for the Taxonomy tree XML file.

  - **sequences**

    This section follows the Sequence XML file structure.

Listing 7: Database in XML skeleton.

```xml
<biodata>
  <...labels...>
  <...ranks...>
  <trees>
    <...all taxonomy trees...>
  </trees>
  <...sequences...>
</biodata>
```

Label type	Operators	Values
URL, text and object	• equal: Equal comparison. • contains: If the label contains a substring. • starts: If the instance starts with. • ends: Starts counterpart. • regexp: Regular expression matching.	—
Bool	• equal: Equal comparison.	The value should be ”true” or ”false”.
Integer and float	• = • > • < • >= • <=	The value should be a number.
Position	• = • > • < • >= • <=	Before the operator you should indicate the position component to compare: ’start’ or ’length’.
Date	• equal: Equal comparison. • before: Date is before some date. • after: Date is after some date.	Values should be in the form *dd-mm-yyyy* like *03-11-2009.*
Taxonomy	• like: A taxonomy name to search for.	The **like** operator gets a taxonomy name and then searches all taxonomies with that name in the system and if the sequence points to any of them the query succeeds.
Reference	• like: A sequence name to search for.	The values and operators work just like the taxonomy labels, but applied for sequences.

Table 2: Operators and values in query expressions.

9 Query language

A simple, yet arbitrarily complex, query language was designed to search stored sequences using annotated information present in label instances.

A simplified grammar in BNF format for this language is shown in Figure 23. Note that every label supports two basic unary operators: **exists** and **notexists**, when used they filter sequences that contain any value label or no value at all, respectively. Queries can be nested using the AND, OR and NOT operators. Parenthesis can also be used to group expressions.

⟨expression⟩→⟨expression⟩ AND ⟨expression⟩| ⟨expression⟩ OR ⟨expression⟩| NOT ⟨expression⟩|(⟨expression⟩) | ⟨terminal⟩
⟨terminal⟩→⟨label name⟩⟨unary operators⟩|⟨bool terminal⟩|⟨integer terminal⟩|⟨float terminal⟩|⟨position terminal⟩|⟨taxonomy terminal⟩|⟨text terminal⟩|⟨url terminal⟩| ⟨obj terminal⟩| ⟨date terminal⟩
⟨bool terminal⟩→⟨label name⟩|⟨label name⟩⟨bool operators⟩⟨bool value⟩
⟨bool operators⟩→⟨base operators⟩
⟨bool value⟩→ **true**| **false**
⟨unary operators⟩→**exists**|**notexists**
⟨base operators⟩→**is**| =| **eq**| **equal**
⟨integer terminal⟩→⟨label name⟩⟨numeric operators⟩⟨integer value⟩
⟨float terminal⟩→⟨label name⟩⟨numeric operators⟩⟨float value⟩
⟨position terminal⟩→⟨label name⟩⟨position type⟩⟨numeric operators⟩⟨integer value⟩
⟨numeric operators⟩→⟨base operators⟩|>| >=| <| <=
⟨position type⟩→**start**|**length**
⟨taxonomy terminal⟩→⟨label name⟩⟨taxonomy operators⟩⟨label value⟩
⟨taxonomy operators⟩→⟨base operators⟩| **like**
⟨url terminal⟩→⟨label name⟩⟨text operators⟩⟨url⟩
⟨text terminal⟩→⟨label name⟩⟨text operators⟩⟨label value⟩
⟨text operators⟩→⟨base operators⟩|**contains**| **starts**| **ends**| **regexp**
⟨obj terminal⟩→⟨text terminal⟩
⟨date terminal⟩→⟨label name⟩⟨date operators⟩⟨date value⟩
⟨date operators⟩→⟨base operators⟩| **after**| **before**
⟨date value⟩→⟨day⟩**-** ⟨month⟩ **-** ⟨year⟩
⟨label name⟩→ ⟨base label name⟩|⟨base label name⟩ [⟨string⟩]
⟨base label name⟩→"⟨string⟩"| ⟨string⟩
⟨label value⟩→"⟨string⟩"| ⟨string⟩

Figure 23: Query language written in BNF.

All labels support a basic set of operators: **is**, **=**, **eq** and **equal**. All those operators do the same thing and, depending on the label type, they filter sequences which contain the specified label value.

We can also specify a multiple label instance with the parameter selector, using *label_name[parameter]*. If an expression involves a multiple label that is not parameter specific, all label instances will be considered, instead of only one.

The following list specifies the differences for each label type:

- **Bool**

  Labels of this type can use the equal operation on values *true* or *false*. We can also skip the operator and value altogether and only keep the label name, as the example: *dna and length > 5* instead of *dna is true and length > 5*.

- **Integer and float**

  Numeric labels use the basic comparison operators: =, >, >=, <, <=.

- **Position**

  For position labels we must first select between the start or the length component, and then an integer operator. Example: *label_name start > 5*.

- **Taxonomy and reference**

  For these kinds of labels we can also use the operator **like**, which has the same effect as the standard equal operator. Those operators work by searching all sequences or taxonomies where the name matches the provided regular expression and then filtering the result list of sequences who have at least one label instance point to the same sequence or taxonomy of the former search result.

- **Url and text**

  For these label types the operators provided are: **starts** (if the string starts with the provided value), **ends** and **regexp**, for regular expression matches.

- **Object**

  Object labels can use the same text operators to search the filename associated with the label instance.

- **Date**

  Date labels provide day based operators: **equal** (in the same day), **after** (after the day), **before** (before the day).

References

[1] PHP: HyperText Preprocessor, http://www.php.net/

[2] FASTA format description, http://www.ncbi.nlm.nih.gov/blast/fasta.shtml