

PhyloclassTalk v 1.0

Hernán Morales

Guillermo Giovambattista

26/11/2015

Contents

Introduction	3
Platforms	3
Privacy	4
Getting Started	5
What you need	5
Installation	5
Windows	5
Linux	5
Subsystems	6
Projects	6
Blast Query Builder	7
Loading a Data set	7
Building the query	8
Executing Queries	10
Exporting Results	11
Debugging Queries	12
Rule-Base	13
Species Repository	14
Browsing Repository Data	14

Updating Repository Data	14
Adding a Custom Repository	14
Territory Builder	17
Introduction	17
Creating a new territory	17
Open territory files	20
Classifier	22
Introduction	22
Classifier Parameters	22
Classification Matches	23
Classification Mismatches	23
GenBank Browser	24
GenBank Feature Tables	24
Files	26
Export Formats	26
Using Reflective Facilities of Smalltalk	26
Credits	27
Troubleshooting	27
Communication Channels	29
Contributing	29
Developing PhyloclassTalk	30
Building PhyloclassTalk in UNIX-like	30
Building PhyloclassTalk in Windows	30
Known Issues	30
License	31

Introduction

PhyloclassTalk is an open-source bioinformatics platform that aims to help providing information about wealth and geographical distribution of studied species. Its main features are:

- A novel graphical user-interface to build BLAST queries (Blast Query Builder).
- A territorial builder framework and user-interface based in a new geopolitical library (Territorial).
- A species repository module enabling species name search and selection for later recognition.
- Geographical classification of retrieved sequences through feature extraction and text-mining from the GenBank database.

PhyloclassTalk is developed using the following technologies:

- Pharo Smalltalk (<http://www.pharo.org>)
- Spec UI Framework (<http://spec.st>)
- Roassal Visualization Engine (<http://objectprofile.com>)
- Fuel Serialization Engine
- BioSmalltalk (<http://biosmalltalk.github.io/web/>), a pure object bioinformatics library.

Platforms

PhyloclassTalk is distributed as a desktop application, developed, maintained and tested on Pharo Smalltalk, and therefore it should be able to execute under Microsoft Windows, GNU/Linux and MacOS X platforms. To access latest fixes and features, click in the Configuration icon and the button "Update PhyloclassTalk" from the settings list. PhyloclassTalk has been developed and tested in the following Operating Systems

- Microsoft Windows XP (SP2, SP3), 7, 8.0, 8.1
- GNU/Linux CentOS 6, 7

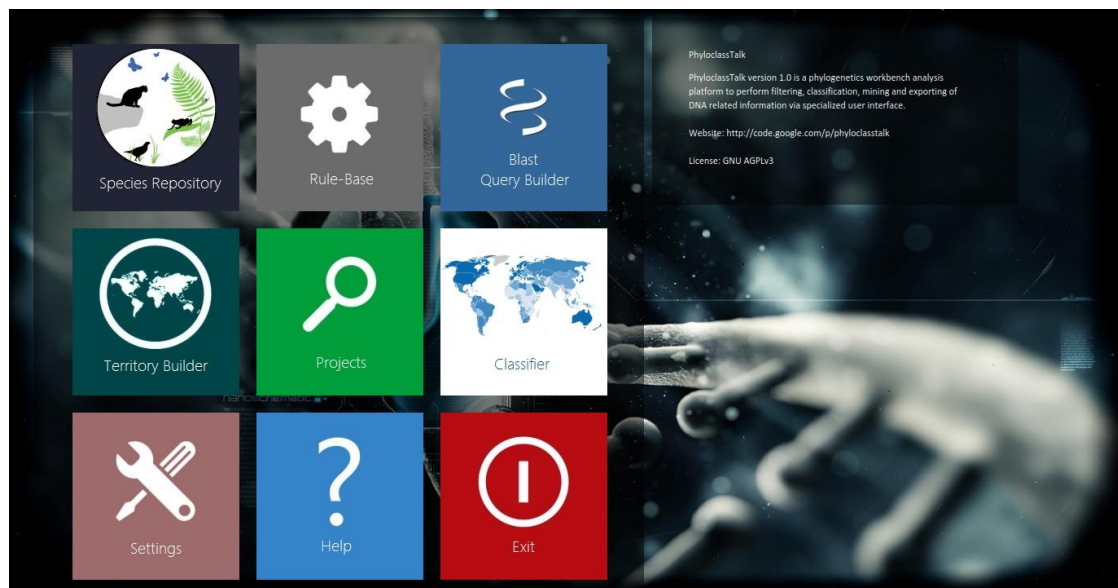


Figure 1: PhyloclassTalk Main Window

- GNU/Linux Debian

Privacy

Project data and user working data sets are not sent to any on-line server.

Getting Started

What you need

- A BLAST results file in XML format exported from the NCBI WebSite.
- A group of territories or breeds to label retrieved sequence records.

Installation

Windows

On Windows, simply run the self-installer file. The PhyloclassTalk executable Windows installer can handle required DLL files installation and decompressing the executables and data files into your preferred location. It also includes the sample files.

Linux

On Linux, untar and uncompress the downloaded distribution:

```
tar zxvf phyloclasstalk-1.0.0.11.tar.gz  
cd phyloclasstalk-1.0.0.11  
./pharo phyloclasstalk-1.0.0.11.image
```

Note: Version numbers can differ according to the current PhyloclassTalk version.

Subsystems

The PhyloclassTalk application's architecture and behavior can be broken up into two separate function-groups:

- Configuring and running a classifier over downloaded information.
- Curating the classified results while updating dictionaries.

Both tasks will be covered in more detail later in this document. PhyloclassTalk enables you to easily filter BLAST hits, associate sequences with (currently only GenBank) annotations, and perform classifications on these sequences.

Projects

A project must be created to do anything useful beyond reading help and configuring global parameters. To create a PhyloclassTalk project, click the main **Projects** button and the **Create New Project** button. Projects are not saved on disk until the **Save Current Project** button is clicked. Currently projects includes basic information such as Date of creation, Author name, Version number, and specific information related with text-mining and classification:

- The classifier used.
- Selected species/organism repositories to support text-mining and classification.
- Territorial area of interest.
- Blast Query Manager, which maintains Blast filters and query results.

Once a Project is created, all the remaining tools became available to work in the context of the current project. One could start by defining a territorial area to narrow the text-mining or by filtering a BLAST XML file downloaded from the NCBI's web site.

Blast Query Builder

Blast is a family of programs (BLASTn, BLASTp, BLASTx, TBLASTn and TBLASTx) to compare a short bio-sequence (called the "query" sequence) against sequences in a large database. There is a huge literature about sequence (and string) matching which can be queried through a simple search in Internet. For such reason an explanation of Blast concepts are beyond the scope of this manual, however we assume a basic understanding of Blast parameters for proper manipulation of the Blast Query Builder.

The Blast Query Builder is an user interface to create and execute queries dynamically against a blast result dataset previously downloaded in XML format. Such input file can be obtained from the NCBI Blast Web Site, exporting the resulting alignment in XML format.

Loading a Data set

To build a query first load XML results exported from the NCBI website. Select File -> Open BLAST XML from the main menu.

If Blast XML results loading is successful, you should see an information dialog as confirmation. A basic validation check is performed to assure a Blast XML file was actually selected. However, XML files which are truncated or not in the expected format (NCBI's BLAST XML) will not be detected until query execution. Blast Query Builder validation process only reads the first bytes of the loaded file, but does not process the entire file until necessary.

You must create and execute at least an initial query filter to work with the results.

Note: Sample XML data sets are included in the PhyloclassTalk files sub-directory.

Building the query

The BLAST query builder works by selecting BLAST properties of interest along with an operator and a query value. Such "triplet" forms a filter. A query cannot be executed until at least a filter is provided. The steps for creating and executing a query are:

- Select property, operator and value to create a filter: Operators are associated with the property selected. Selecting a numeric property (like "Alignment Length") will present corresponding comparators in the operator list. Selecting a string property (like "Hit Definition") will present appropriate operations for comparing Strings.
- Hit "+" button to add a new filter (optional).
- Click the "Execute" button.
- Select nodes of interest in the resulting palette: Blast hits information is taken from the DTD definition at http://www.ncbi.nlm.nih.gov/dtd/NCBI_BlastOutput.mo

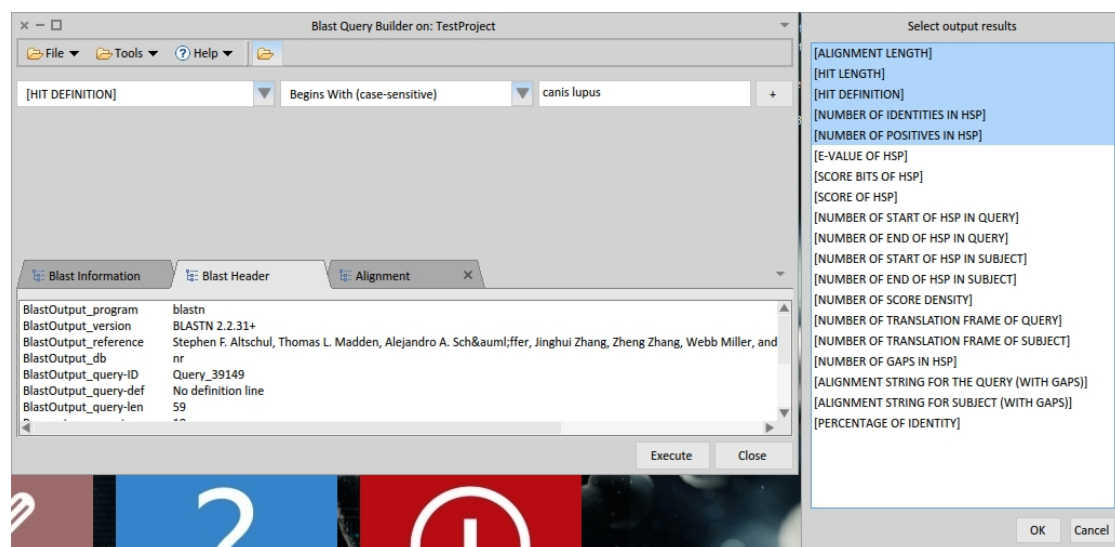


Figure 2: Blast Query Results Selection

From <https://www-bimas.cit.nih.gov/blastinfo/blaststrategy.html>

The High-scoring Segment Pair (HSP) is the fundamental unit of BLAST algorithm output. An HSP consists of two sequence fragments of arbitrary but equal length whose alignment is locally maximal and for which the alignment score meets or exceeds a thresh-

old or cutoff score. A set of HSPs is thus defined by two sequences, a scoring system, and a cutoff score; this set may be empty if the cutoff score is sufficiently high. In the programmatic implementations of the BLAST algorithm described here, each HSP consists of a segment from the query sequence and one from a database sequence. The sensitivity and speed of the programs can be adjusted via the standard BLAST algorithm parameters *W*, *T*, and *X* (Altschul et al., 1990); selectivity of the programs can be adjusted via the cutoff score.

Supported properties of BLAST results are:

- Alignment Length: Length of the alignment used
- Hit Definition: Scan the definition line for subject sequence.
- Hit Length: Length of the subject sequence.
- Number of Identities in HSP (high-scoring segment pair, or match): The number and fraction of total residues in the HSP which are identical.
- Number of Positives in HSP: The number and fraction of residues for which the alignment scores have positive values.
- E-Value of HSP: Expect value of the HSP
- Score Bits of HSP: A Bit-Score is a normalized log-scaled version of a score expressed in bits. From the NCBI web site: *The bit score, S' , is derived from the raw alignment score, S , taking the statistical properties of the scoring system into account. Because bit scores are normalized with respect to the scoring system, they can be used to compare alignment scores from different searches.*
- Score of HSP: The HSP score represents the overall quality of the alignment. Higher numbers correspond to higher similarity. The score scale depends on the scoring system used (substitution matrix, gap penalty)
- Number of Start of HSP in Query
- Number of End of HSP in Query
- Number of Start of HSP in Subject
- Number of End of HSP in Subject
- Number of Score Density
- Number of Translation Frame of Query
- Number of Translation Frame in Subject
- Number of Gaps in HSP

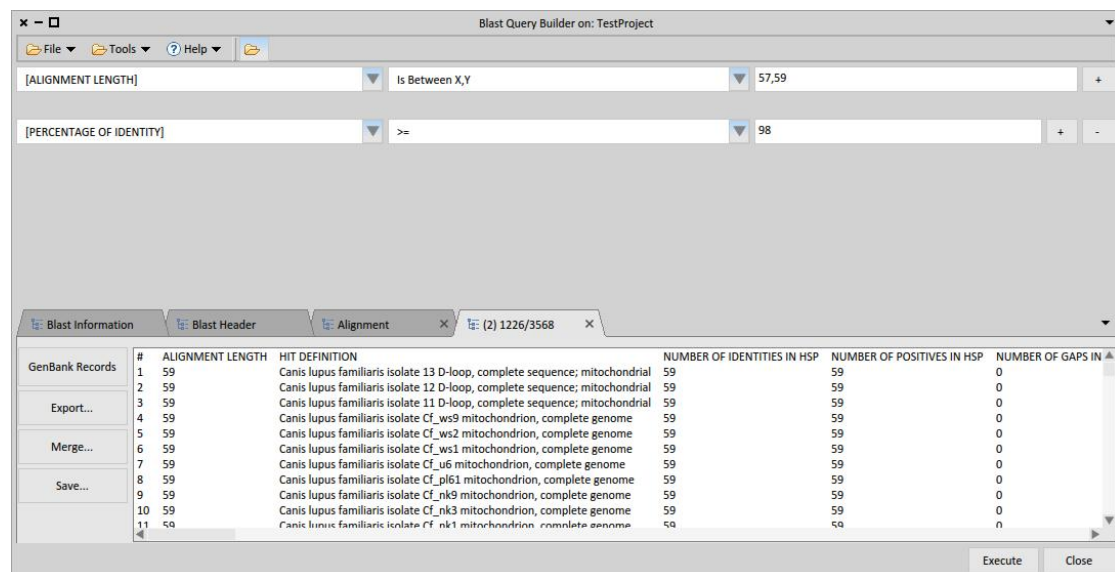
- Alignment String for the Query (with gaps)
- Alignment String for the Subject (with gaps)

Executing Queries

Query execution works by walking each filter and executing in turn. Execution can be cumulative (default behavior) or non-cumulative. In the cumulative configuration, each filter is executed acting on the previous filtered results. Non-cumulative execution generates a result set for each filter, which is useful to compare filters performance.

Cumulative execution and other useful settings can be changed by selecting Tools -> Options from the toolbar menu.

Each execution result is displayed in a new palette in the Results pane, however it is not automatically added to the user project. To add a blast results in a palette click in the Save button. All saved BLAST filtered results can be accessed through Tools -> Saved Blast Results window.



The screenshot shows the 'Blast Query Builder' window with the following filters:

- [ALIGNMENT LENGTH] Is Between X,Y 57,59
- [PERCENTAGE OF IDENTITY] >= 98

The results pane shows a table with the following columns: #, ALIGNMENT LENGTH, HIT DEFINITION, NUMBER OF IDENTITIES IN HSP, NUMBER OF POSITIVES IN HSP, and NUMBER OF GAPS IN A. The table contains 11 rows of results, all with an alignment length of 59 and 0 gaps.

#	ALIGNMENT LENGTH	HIT DEFINITION	NUMBER OF IDENTITIES IN HSP	NUMBER OF POSITIVES IN HSP	NUMBER OF GAPS IN A
1	59	Canis lupus familiaris isolate 13 D-loop, complete sequence; mitochondrial	59	59	0
2	59	Canis lupus familiaris isolate 12 D-loop, complete sequence; mitochondrial	59	59	0
3	59	Canis lupus familiaris isolate 11 D-loop, complete sequence; mitochondrial	59	59	0
4	59	Canis lupus familiaris isolate Cf_ws9 mitochondrion, complete genome	59	59	0
5	59	Canis lupus familiaris isolate Cf_ws2 mitochondrion, complete genome	59	59	0
6	59	Canis lupus familiaris isolate Cf_ws1 mitochondrion, complete genome	59	59	0
7	59	Canis lupus familiaris isolate Cf_u6 mitochondrion, complete genome	59	59	0
8	59	Canis lupus familiaris isolate Cf_pl61 mitochondrion, complete genome	59	59	0
9	59	Canis lupus familiaris isolate Cf_nk9 mitochondrion, complete genome	59	59	0
10	59	Canis lupus familiaris isolate Cf_nk3 mitochondrion, complete genome	59	59	0
11	59	Canis lupus familiaris isolate Cf_nk1 mitochondrion, complete genome	59	59	0

Figure 3: Blast Query Results

Note: If you do not save your filtered results clicking the Save button, they will be lost if you close the Blast Query Builder window.

Exporting Results

Results can be saved and exported to FASTA and CSV formats. Exporting as CSV report includes all the available information previously selected in the resulting nodes window. To export to FASTA format, select, in the output result window, both the HIT_DEFINITION and the aligned sequences items.

Debugging Queries

Debugging implies the execution of each line of a script step by step. The Blast Query Builder runs into Pharo, a full Smalltalk object programming environment which enables the option of debugging queries with programming scripts by opening a Workspace (a tool used to write or paste scripts).

To debug a script, open a "World Menu" by clicking in any empty area of Phylo-classTalk, then click Workspace and replace the organismName and blastXML variable, in the following script, with your organism name (as shown in the Hit definition line in the XML output) and exported Blast XML file name:

```
| qm prop cmp organismName outputProperty blastXML |
organismName := 'homo sapiens'.
blastXML := 'V9T8NT6S01R-Alignment.xml'.
prop := BlastQBProperty new
    name: 'Hit_def';
    propertySelector: #selectHitDefinitions:
    yourself.
outputProperty := BlastQBProperty new
    propertySelector: #selectHspAlignLengths;
    name: 'Hsp_align-len'.
cmp := QBPropertyComparator new selector: #beginsWith:.
qm := BlastQBQueryManager new
    addBlast: blastXML asFileReference readStream;
    addNewQuery: { prop . cmp . organismName };
    selectedNodes: (OrderedCollection with: outputProperty);
    executeQuery.
```

Select the code and bring the contextual menu, then click Debug it to access the Debugger. An explanation of how to debug code is beyond this manual but any search in Internet can reveal short tutorials to get started.

Rule-Base

The Rule-Base Module is an UI for browsing and editing rules in a hierarchical fashion. Rules are displayed as a binary tree enabling to explore how the classification algorithm is evaluated. Each selected rule is displayed along with the Smalltalk code which contains its condition, the true and false branches.

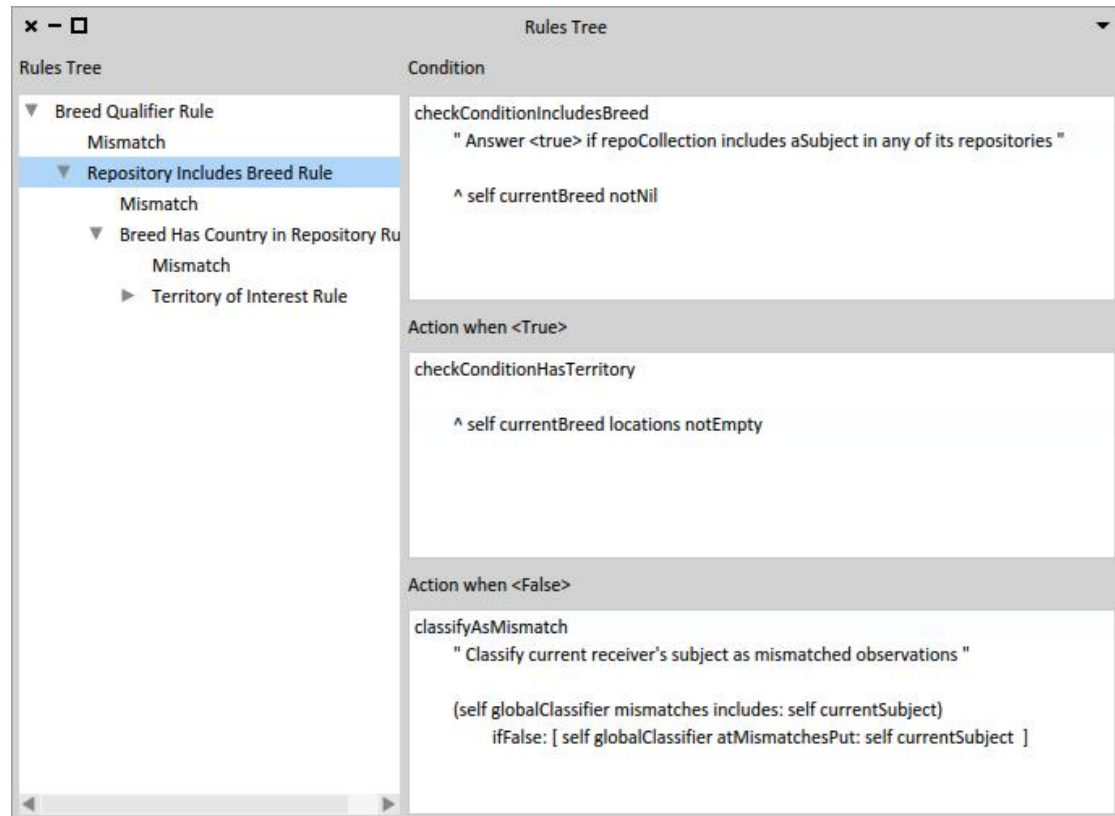


Figure 4: PhyloclassTalk Rules Tree

Species Repository

Browsing Repository Data

The species repository contains curated dictionaries of species names, synonyms, locations and additional information like date of extinction (if extinct). Each species is shown in a tree, and could contain several repositories.

Our species model does not integrate species data in a unique dictionary, nor normalize identifiers across databases. This is done on purpose to be able to measure performance of specific data sources, and because different species/-pedigrees dictionaries have different models (and would require an ontology disambiguation or mapping between different fields). The classifier module allows to combine different dictionaries but this feature is experimental and currently has not been tested.

A species item is displayed along with the number of repositories currently imported. Opening a species tree item will show a list of repositories for that species, with the corresponding size of the dictionary.

The repository information panel displays metadata about the currently selected item in the species repository tree.

The repository preview panel displays the selected dictionary, with the possibility of filtering by specific term.

-

Updating Repository Data

All repositories can be updated from remote (currently only through **Google Drive**) by clicking Update button.

Adding a Custom Repository

A custom repository could be created directly through PhyloclassTalk by clicking the "Add Repository" button. All mandatory fields are marked with *

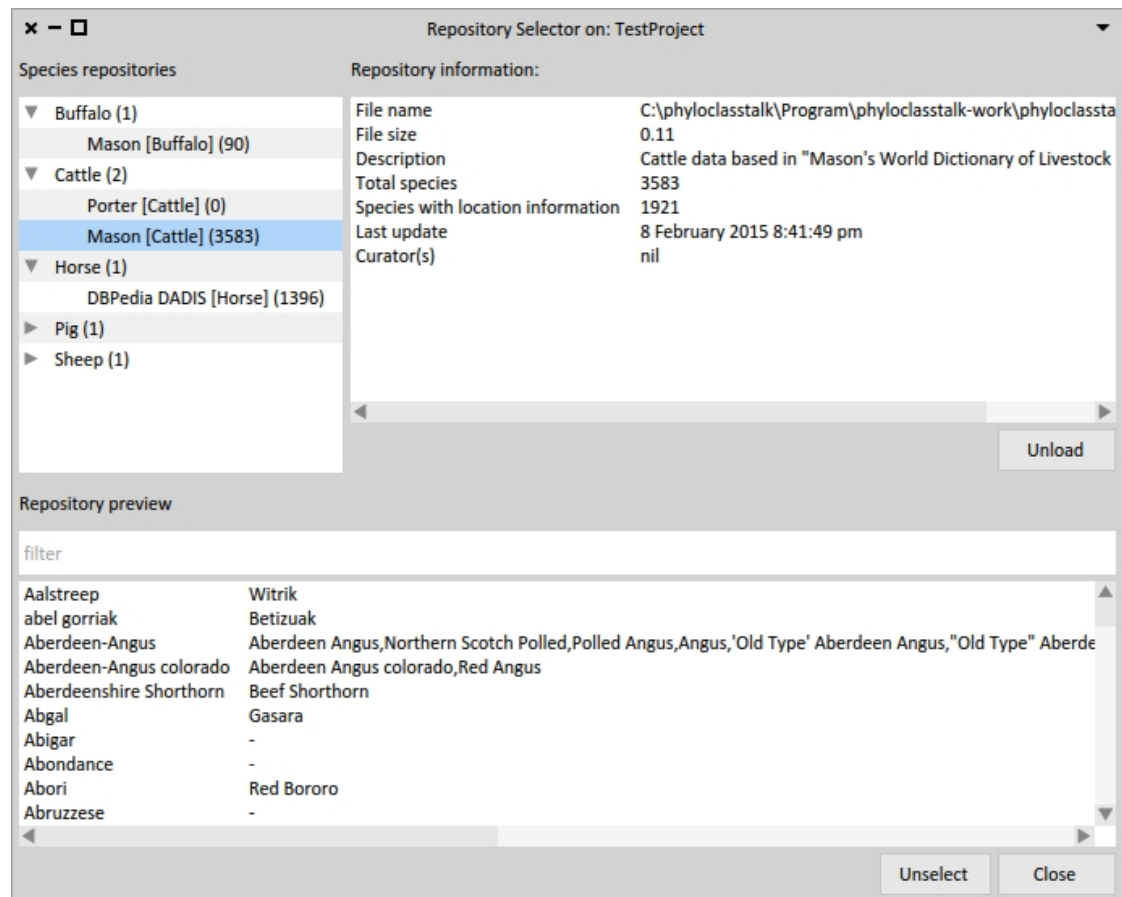


Figure 5: Repository Selector

If you create or curate a dictionary, you could have integrated in the next Phylo-classTalk distribution. Please register in the project page to be sure your changes will be reviewed and integrated in the next software version.

A species repository format has a CSV format with the following specification:

- Field Separator should be semicolon ";"
- Column 1: Unique name of pedigree (mandatory) in English language.
- Column 2: Synonyms (if available) delimited by commas (,) in English language. If no synonyms are entered, this field should be empty.
- Column 3: Territory of origin in English language. If several territories are available, they must be delimited by commas (,). If no territories are entered, this field should be empty. Territory of origin could be, for example, a country name.
- Column 4: Sub-territory or sub-region of territory of origin.
- Column 5: If extinct, date of extinction.
- Column 6: Unique name of pedigree in Spanish language.
- Column 7: Unique name of pedigree in French language.

If a pedigree name or synonym is duplicated, only the first detected (from the beginning of file) will be used during a classification.

Territory Builder

Introduction

The Territory Builder supports building custom geopolitical territories for population studies. Any containment relationship could be defined, for example, you may want to define your territory of interest by creating artificial groupings like "Developed Countries" and "Developing Countries".

The usage of the Territory Builder is driven by a user-interface (UI) wizard. You can open the Territory Builder wizard by clicking the "Territory Builder" icon in the main application screen. When the process finishes, an "Explorer" window is opened. In the Explorer you may see the properties in your new territory, and save it to a file.

The Territory API is better described in its own manual.

Creating a new territory

To create a new territory, select the "New Territory" option in the wizard, and click the Next button. A territory name could be composed of any character supported in the host file system, this usually means that characters as `,` `/` are not allowed. The entered territory name will be used as the file name when saving the territory to a file. A territory is a file with `.tty` extension.

A new territory could be "Composite" or "Simple":

- A "Simple Territory" consist in a single atomic location, which will not be subdivided logically in your study. This means it could be a Country or even a Continent, but it will not be composed of other sublocations.
- A "Composite Territory" is composed by one or several Simple or Composite territories itself.

Enter a territory name, and hit Next.

Next window will allow to compose territories based on different types of common organizations in the world. New territory groups can be created by using

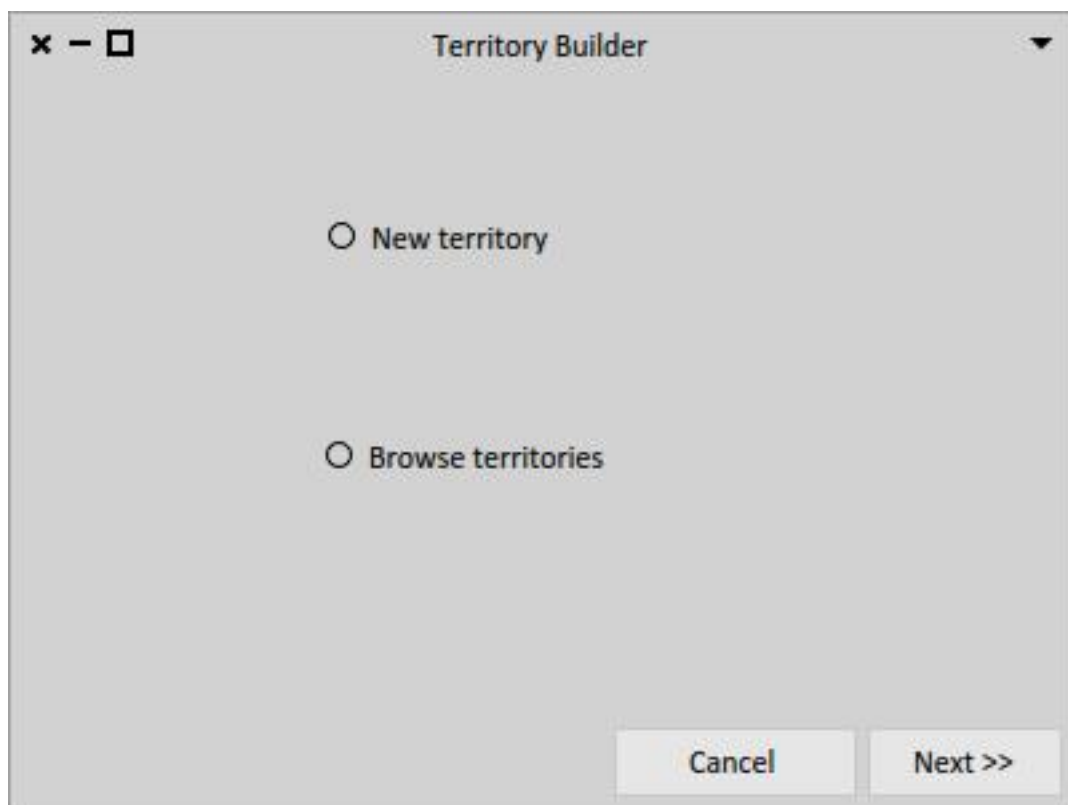


Figure 6: Territory Builder

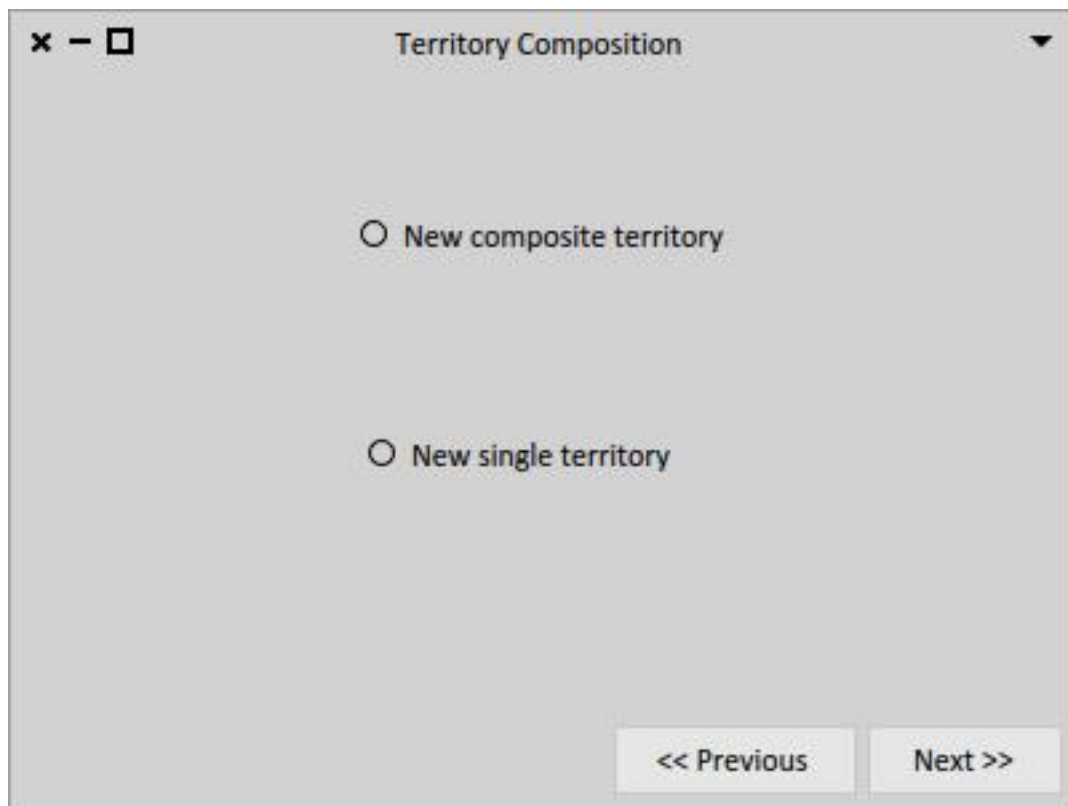


Figure 7: Territory Composition

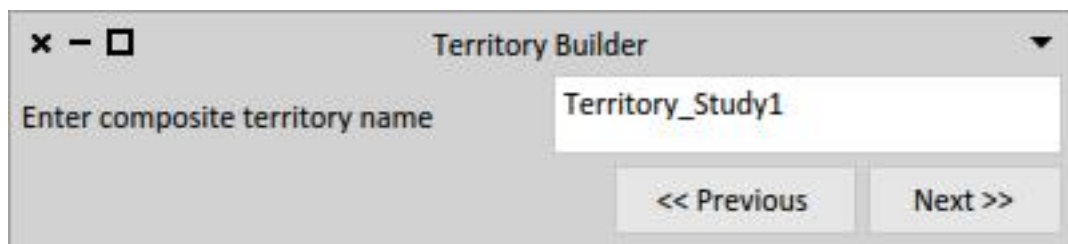


Figure 8: Territory Name

the Territorial Application Programming Interface.

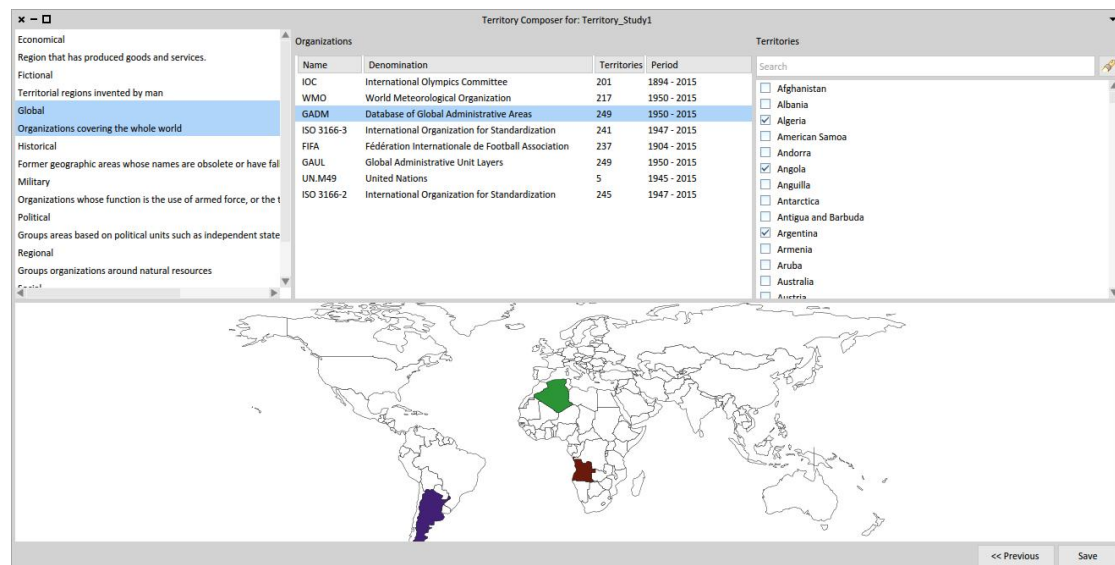


Figure 9: Territory Name

Once finished, the wizard will popup an Explorer window to review or make modifications to the new territory. To view the possible operations over the territory, select the top node (identified by the Territory name) and right-click to bring the contextual menu.

To add territories to a "Composite Territory", select the option "Add territory..." in the menu item.

Open territory files

A territory is a file with .tty extension. You can open a previously created territory by:

- Click in the Territory Builder icon and select Territory Viewer. A window with list of saved territories is displayed at the top. The bottom of the window displays a map for the currently selected territory.
- Open the "World menu", then Tools, and File Browser. Navigate to your project folder, and selecting the territory file. Once found, select the .tty file and then right-click to bring the contextual menu. Select the menu item

"BioSmalltalk: Materialize". An explorer window will open displaying the territory containment structure.

Classifier

Introduction

The PhyloclassTalk Classifier includes a friendly user-interface window to perform rule-based classification of observations (or instances). Observations are based in GenBank Records but other type of records (for example, EMBL) could be possible, although not currently implemented.

The classifier window is divided in three main panels. Left-most panel - named Classifier Parameters - contains the classifier settings stored in the current project. Both the middle and right-panels contains the classification results once finished.

Classifier Parameters

The Classifier Parameters includes all necessary settings to change the Classifier behaviour. All settings are required to have non-empty values.

Classifier Rules

Rules are the core of the classifier, they define the conditions and actions to be applied to the project observations. Such rules can be configured from the Edit button in the Classifier Parameters panel, and different rules can be selected in the drop list. The currently selected rule name correspond to a *Smalltalk class name* which is an entry point for specifying the actual rule code.

PhyloclassTalk includes two pre-built rules:

- **PCTBreedRuleTree:** Contains rules for recognizing pedigree names, including synonyms.
- **PCTTerritorialRuleTree:** Contains rules for recognizing territorial names, including demonyms and synonyms.

Observations

Observations are the user instances to be classified. Observations can be filtered by clicking the Change... button in the Observations field. The current implementation supports browsing GenBank Records. We refer to the GenBank Browser Help to learn how to use the UI.

The Selected Species settings is intended to display a list of repositories selected in the current project. The Selected Species UI is described in the Species Repository window.

Classification Matches

Under development

Classification Mismatches

Mismatches can be queried from the "Mismatches View" by selecting an item, a feature of interest, and launching a search provider through an external web browser.

The search provider enables to narrow the query to a specific search engine. For example, selecting the feature for DOI will search using the <http://doi.org/> search engine, selecting an accession number feature (GBSeq_accession-version), will launch a search through Entrez e-Utils service.

Notice the features available to select are generated dynamically from the currently available features for the selected item in the Mismatches View data grid.

GenBank Browser

The GenBank Browser allows to browse downloaded GenBank records from the NCBI GBSeq specification (http://www.ncbi.nlm.nih.gov/dtd/NCBI_GBSeq.dtd). From the GenBank DTD comment:

GBSeq represents the elements in a GenBank style report of a sequence with some small additions to structure and support for protein (GenPept) versions of GenBank format as seen in Entrez. While this represents the simplification, reduction of detail, and flattening to a single sequence perspective of GenBank format (compared with the full ASN.1 or XML from which GenBank and this format is derived at NCBI), it is presented in ASN.1 or XML for automated parsing and processing. It is hoped that this compromise will be useful for those bulk processing at the GenBank format level of detail today. Since it is a compromise, a number of pragmatic decisions have been made.

The GenBank Browser works by parsing the nodes found in a downloaded GenBank XML tree. You can download such GenBank XML tree through the Blast Query Module. After filter execution, each results palette includes a button to download the GenBank records for the retrieved/filtered hits.

GenBank Feature Tables

The GenBank Browser contains a list of features to filter results in a Data View grid. Once a feature is selected, features (columns) are updated for the current loaded data set. This is useful to narrow which nodes are significant for your needs by improving space and processing time. Notice that the grid view displays additional/different columns not displayed in the "Profiles List". This is because GenBank/EMBL/DDBJ provides a Feature Table (see http://www.insdc.org/files/feature_table.html for details) to include additional extensible features. Columns like

- "organism"
- "isolate"

- "country"
- "breed"

etc. are contained in **GBQualifier_name** and **GBQualifier_value** nodes in the NCBIGBSeq DTD. This means that to view such "hidden" features, you should select GBQualifier_name and GBQualifier_value in the list of features.

Files

PhyloclassTalk creates a project file `.phylocst` in the (pharo) virtual image directory, which contains different objects related to an user project.

Extension	Description
<code>.phylocst</code>	PhyloclassTalk Project File (Fuel format)
<code>.bq</code>	Blast Query Builder query file (Fuel format)

Export Formats

Extension	Description
<code>.arp</code>	Arlequin Software Project File (ASCII)
<code>.csfasta</code>	FASTA File Format (ASCII)
<code>.csv</code>	Comma-Separated Values (ASCII)
<code>.meg</code>	MEGA Software (ASCII)

Using Reflective Facilities of Smalltalk

Pharo Smalltalk is usually described as a system of living objects, where almost everything can be easily explored and modified. PhyloclassTalk is implemented on top of Pharo Smalltalk and a group of supporting libraries.

Several parts of the system includes an option for inspecting or exploring the object under focus. For example the Blast Query Builder includes an option to explore the saved or merged results.

For details on the Smalltalk language and the Pharo environment, we encourage users to read the Deep Into Pharo book.

Credits

PhyloclassTalk has been originally developed by Hernán Morales and Guillermo Giovambattista. The work is sponsored by the Institute of Veterinary Genetics (IGEVET) - National Scientific and Technical Research Council (CONICET) in Argentina. To find more information about PhyloclassTalk, please visit the official web page <http://phyloclasstalk.github.io/>

This User manual was written by Hernán Morales hernan.morales@gmail.com and is a concise guide explaining the features as well as giving step-by-step instructions on using PhyloclassTalk. Please send all comments and suggestions regarding this manual to the BioSmalltalk Mailing List by sending an email to biosmalltalk-users@googlegroups.com.

Troubleshooting

Please try to ascertain that the behavior you see really is a bug.

- If PhyloclassTalk crashes, it's a bug.
- If PhyloclassTalk does not behave as documented, it's a bug.
- If things work strange, but you are not sure about the way they are supposed to work, it might well be a bug, but you might want to doublecheck the documentation and the mailing lists.

Before you contact mailing list, gather the background information that you will need to describe your problem. When describing a problem, be as specific as possible and include all relevant background information. To save time, know the answers to these questions:

- Which PhyloclassTalk version were you running when the problem occurred?
 - Information provided in the upper right box in the application, or in the user manual if application couldn't be opened.
- Do you have logs, traces, or messages that are related to the problem? Logs are found in these locations:

- The directory where the .image file is found, file named : PharoDebug.log
 - The Transcript window inside the application: Click in empty space, Tools, Transcript.
- Can you reproduce the problem? If so, what steps do you take to reproduce it?
- Is there a workaround for the problem? If so, be prepared to describe the workaround.

Communication Channels

Contributing

PhyloclassTalk is actively in development and constantly improving. If you have an interest in PhyloclassTalk, the Source Code Repository is the place you want to go right away. The main developer of PhyloclassTalk may be reached by sending email to hernan.morales@gmail.com

PhyloclassTalk is under active development by members of "Instituto de Genética Veterianaria (CCT - La Plata)".

Developing PhyloclassTalk

Building PhyloclassTalk in UNIX-like

To build latest image of PhyloclassTalk in UNIX-like environment, execute the following script:

```
#!/bin/sh(wget -O- http://get.pharo.org/50+vm | bash)
&wait./pharo-vm/pharo Pharo.image config
"http://smalltalkhub.com/mc/hernan/PhyloclassTalk"
"ConfigurationOfPhyloclassTalk" -printVersion -install=bleedingEdge
```

Building PhyloclassTalk in Windows

To build latest image of PhyloclassTalk in Windows environments you need to install MinGW with MSYS, and execute the following script from the MSYS shell (usually `C:\MinGW\msys\1.0\msys.bat`):

```
#!/bin/sh(wget -O- http://get.pharo.org/50+vm | bash)
&wait./pharo-vm/pharo Pharo.image config
"http://smalltalkhub.com/mc/hernan/PhyloclassTalk"
"ConfigurationOfPhyloclassTalk" -printVersion -install=bleedingEdge
```

Known Issues

Please visit the source code repository at <https://github.com/PhyloclassTalk> to view the current issues

License

Copyright 2016 by Hernán Morales Durand and Guillermo Giovambattista.

The contents of this book are protected under the Creative Commons Attribution-ShareAlike 3.0 Unported license.

You are free:

- to **Share**: to copy, distribute and transmit the work,
- to **Remix**: to adapt the work,

Under the following conditions:

Attribution. You must attribute the work in the manner specified by the author or licensor (but not in any way that suggests that they endorse you or your use of the work).

Share Alike. If you alter, transform, or build upon this work, you may distribute the resulting work only under the same, similar or a compatible license.

For any reuse or distribution, you must make clear to others the license terms of this work. The best way to do this is with a link to this web page: <http://creativecommons.org/licenses/by-sa/3.0/>

Any of the above conditions can be waived if you get permission from the copyright holder. Nothing in this license impairs or restricts the author's moral rights.

Your fair dealing and other rights are in no way affected by the above. This is a human-readable summary of the Legal Code (the full license): <http://creativecommons.org/licenses/by-sa/3.0/legalcode>