Software Requirements Specification
Phyloloc

Nicolás Bombau Carlos Castro Damián Domé
Emmanuel Teisaire

October 7, 2010

Contents

1 Introduction

1.1 Purpose

Emergent viruses and other pathogens spread out by means of chains of infection in which, basically, one host passes the pathogen to another one. In this way, pathogens gradually spread along a geographic range that usually corresponds to its host distribution. Hosts capable of moving long distances can therefore largely drive pathogen dispersal. For example, it is though that Human Immunodeficiency Virus 1 (HIV-1) spread out from Africa into remote places around the world following human movements.

Deciphering the history of pathogens' demography is of keen interest in epidemiology, as is demonstrated by many works that deal with unraveling the dispersion process of pathogens. The combination of data on the evolutionary history of the pathogen (i.e. phylogenetic trees) with information on its geographic distribution, together with other epidemiological data, shall help explaining the present distribution of infectious agents. The basic idea behind this kind of studies is that phylogenetic relatedness is indicative of geographic proximity.

Despite the obvious interest in the problem of tracing the geographic origin and dynamics of emerging diseases, a formal optimality criterion capable of integrating phylogenetic and geographic data is lacking. Phyloloc shall implement a method to estimate the geographic origin of any given clade in a rooted tree. The method produces, for each node in a given tree or set of trees, a vector of potential ancestral locations. Each potential location present in this vector has an associated value, ranging from 0 to 1, that reflects the plausibility of the corresponding location being the geographic origin of the node?s childs.

1.2 Document Conventions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119.

1.3 Intended Audience

Below are enumerated the people involved in the thesis, and their respective roles. In the short term, they shall be the audience of the present document.

- Lic Silvia Gomez: Thesis Director, ITBA
- Lic. Daniel Gutson: Thesis Tutor, FuDePAN
- PhD Leandro Jones: Thesis Colaborator, EFPU
- Nicolás Bombau: Thesist, ITBA
- Carlos Castro: Thesist, ITBA

- Damián Domé: Thesist, ITBA
- Emmanuel Teisaire: Thesist, ITBA

1.4 Project Scope

The product specified in the current document is called **“Phyloloc”**, and it’s main purpose is to, through a formal optimality criterion, integrate the phylogenetic and geographic data of a collection of phylogenetic trees.

The output of the project shall be three software components: a general and resusable phylogenetic utilities library, phyloloc itself, which uses the utilities library, and a user interface, which must also be reusable.

The final product has to let the user load phylogenetic tree or set of trees, and output an integrated tree, which in it’s nodes has a vector of potential ancestral locations and their corresponding plausibility value, ranging from 0 to 1.

The product shall also provide a *user-friendly* interface, in order to let the user visualize phylogenetic trees in a confortable way. The user interface shall also allow basic operations like node selection, highlighting, expansion and collapsation.

The project shall be deployed as a desktop application. As some algorithms may take a considerable amount of time to run, it would be interesting to provide a way to run the algorithms in a distributed way. This functionality shall be included in another future project, being out of the current project scope.

1.5 Document Structure

The structure of the present document follows the recommendations from the “Guide for the requirements specification” published by the IEEE, standard 830-1998.

Section 2 provides an overall description of the most general aspects of the product, such as the product perspective, major features, user interfaces and profiles, and operating enviroment.

Section 3 describes the interfaces of the product, such as it’s user interface and software interfaces.

Section 4 organizes the functional requirements for the product.

Section 5 enumerates the non functional requirements, such as coding style, design and coverage constraints.

2 Overall Description

2.1 Product Perspective

The product is intended to provide the cientific and amateur communities a tool that accomplishes to join the information they have in the form of phylogenetic trees, considering the geographic and phylogenetic information through the method proposed by PhD Leandro Jones.

The tool is also intended to provide quick *point-and-click* capabilities for agile tree analisys.

2.2 Product Features

This section provides a brief description of the major features of the product, which are detailed in 4.

2.2.1 Phylogenetic Trees Loading

In order to integrate trees, in the first place phyloloc must be able to load phylogenetic trees, which shall have previously been obtained using any of the methods available, such as maximum likehood and maximum parsimony methods.

2.2.2 Phylogenetic Trees Visualization and Analisys Tools

The product provides visualization capabilities, specially prepared for really large trees visualization. d for really large trees visualization. The GUI shall also provide basic node operations like searching, selecting, highlighting and coloring.

2.2.3 Phylogenetic and Geographical Data Integration

The product shall provide an analisys operation, which integrates phylogenetic and geographical data, considering branch length and geographic dispersion of locations. The method shall be explained in detail in 4.2.1.

2.2.4 User Classes and Characteristics

The final users are the ones that use the product in order to use the tools it provides. The product is intended to be used by people keen on phylogeny, such as biologists, students, bioinformatics investigators, and anyone interested in the subject.

3 Interfaces

3.1 User Interface

The user interface consists of two forms, a main menu and a configuration panel, detailed below.

3.1.1 Main Menu

The main menu is the place where the top-level commands are grouped. The menu shall at least, include the following options:

1. Load: Load data in order to visualize or analyze it.
2. Close: Closes the system.
3. Search Nodes: Search a determined node by name.
4. Clear Selection: Selects all the tree nodes.
5. Clear Selection: Clears all the selected nodes.
6. Expand Nodes: Expands all selected nodes.
7. Collapse Nodes: Collapses all selected nodes.
8. Color Nodes: Highlights the selected nodes with a color that the user can choose for each coloring.
9. Show Node Ancestors: Highlights all selected nodes' ancestors.
10. Show Node Descendants: Highlights all selected nodes' descendants.

3.1.2 Data Load Form

Form that allows the user to load the trees into the system. The user is asked to load a file, from which the information of the trees is parsed.

3.1.3 Tree Visualization Form

The visualization form allows the user to see the recently loaded trees, and also the integrated tree. This form shall provide the visualization capabilities, and showing the result of applying the basic tools such as collapsing, expanding nodes and node coloring. The user shall also visualize the details of a particular node in a friendly way.

3.1.4 Configuration Panel

The configuration panel allows the user to customize certain aspects of the execution and analisys, such as the integrating algorithm termination criterion, as well as the ways of handling missing data, and other possible system parameters.

3.2 Hardware Interfaces

No requirements specified.

3.3 Software Interfaces

The product shall be divided into three modules:

1. phylopp: Base library that provides phylogenetic utilities, that concern general phylogeny functions that may serve other purposes for future related products. This library shall be completly reusable.
2. phyloloc: The main product, which shall use the utilities provided by phylopp, in order to perform the product-specific algorithms and operations.
3. phylogui: User interface that shall provide the final user ways to interact with phyloloc. As phylogui shall provide the final user ways to visualize phylogenetic trees, that part of the gui shall be completly reusable, and loosely coupled, so that it can be used in future products.

Although phylopp and phylogui are software interfaces that shall interact with phyloloc, all of them shall be implemented as part of the current project.

3.4 Communication Interfaces

No requirements specified.

4 Functional Requirements

4.1 GUI Requirements

4.1.1 REQ01 Data Load

Objective

The objective of this requirement is to let the user load a file containing the trees that will be processed by the product.

Priority High.

Description

1. **File Loading:** The system displays a form drop down list, with the possible file and tree formats available. Also, the system allows the user to select a file to upload from his filesystem. If the file is successfully loaded, the validation of the file takes place. Otherwise, an error message is displayed.
2. **File Content Validation:** The system parses the file loaded in the previous step, using the parse method assigned to the file format selected. If there is a parsing error, an error message is displayed. If the parsing is succesfull, the system indicates the user that the file was loaded.

The tree load format shall be Newick.

4.1.2 REQ02 Terminal Node Search

Objective

The objective of this requirement is to let the user search terminal nodes by name.

Priority

High.

Description

1. **Activation:** The main manu shall have a *search* option. Once this option is selected, the user is prompted to enter the name of the node. In case there are no matches, the user is informed so. Otherwise, the matches are highlighted.
2. **Visualization:** Once the search progress is done, the found and selected nodes - if applies, are highlighted in the tree visualization.

4.1.3 REQ03 Node Selection

Objective

The objective of this requirement is to let the user select a node or set of nodes.

Priority

High.

Description

1. **Node Selection:** The user can select a node by pointing and clicking it. Once a node is selected, the system highlights it.

4.1.4 REQ04 All Nodes Selection

Objective

The objective of this requirement is to let the user select all nodes.

Priority

High.

Description

1. **Node Selection:** The main manu shall have a *select all* option. Once this option is selected, the system selects all nodes.

4.1.5 REQ05 Node Selection Clear

Objective

The objective of this requirement is to let the user clear the selected nodes.

Priority

High.

Description

1. **Selection Clear:** The main manu shall have a *clear selection* option. Once this option is selected, the system removes all highlighting from the tree nodes.

4.1.6 REQ06 Node Expanding

Objective

The objective of this requirement is to let the user expand a given node or set of nodes.

Priority

High.

Description

1. **Node Expanding:** The main manu shall have an *expand* option. Once this option is selected, the system expands the selected nodes, showing its direct children.

4.1.7 REQ07 Node Collapsing

Objective

The objective of this requirement is to let the user collapse a given node or set of nodes.

Priority

High.

Description

1. **Node Collapsing:** The main manu shall have an *collapse* option. Once this option is selected, the system collapses the selected nodes, hiding its children.

4.1.8 REQ08 Node Coloring

Objective

The objective of this requirement is to let the user colur a given node or set of nodes.

Priority

High.

Description

1. **Node Coloring:** The main manu shall have an *Color* option. Once this option is selected, the system displays a set of colors, so that the user selects one. Once a color is selected, the system colors the

4.1.9 REQ09 Node Ancestors Selection

Objective

The objective of this requirement is to let the user select the ancestors of a given node.

Priority

High.

Description

1. **Node Ancestors Selection:** The main manu shall have an *Select Ancestors* option. Once this option is selected, the system recursively selects all the ancestors of all the selected nodes.

4.1.10 REQ10 Node Descendants Selection

Objective

The objective of this requirement is to let the user select the descendants of a given nodes

Priority

High.

Description

1. **Node Descendants Selection:** The main manu shall have an *Select Descendants* option. Once this option is selected, the system recursively selects all the descendants of all the selected nodes.

4.1.11 REQ011 Tree Saving

Objective

The objective of this requirement is to let the user save a tree in the filesystem.

Priority

High.

Description

1. **Tree Saving:** The main menu shall have a *Save* option, which saves the structure of the tree that is being visualized, so that it can be shared, or loaded again later. The export format shall be Newick.

4.1.12 REQ012 Formatted Tree Saving

Objective

The objective of this requirement is the possibility of saving trees in the filesystem, mantaing colouring and expansion preferences.

Priority

Low.

Description

1. **Tree Saving:** The main menu shall have a *Formatted Save* option, which saves the structure of the tree that is being visualized, so that it can be shared, or loaded again later. This option also saves the state of the nodes, such as if it is collapsed or expanded, the colour asigned to it, and its selection.

4.2 Algorithmic Requirements

4.2.1 REQ13 Tree Data Integration

Objective

The objective of this requirement is the integration of a tree's topology, geographical data, branch length and dispersion of locations.

Priority

High

Description

For the integration of the information available, several passes have to be done along all its nodes. Here the detailed procedure is described.

1. **First Pass:** The first one is a down-pass (i.e. from tips to root), in which the nodes' vectors are given a transitory states set using the method proposed in *phyloloc (Jones, Gutson, ...).* Factors are introduced to correct geographical dispersion and branch length.

2. **Second Pass:** The second pass starts from the tree root, passing recursively through all the nodes, until it goes back to the tips.

3. **Further Passes and Convergence:** Once the completion of the first two passes through the tree, the first pass shall be repeated, and afterwards the second pass, until the end condition is true. There are many cases in which the algorithm might terminate.

4. **Convergence Criteria:** The idea is to provide different ways of deciding whether the algorithm has converged or not. The ways that shall be provided are the following:

   (a) Finish when the number of up and down passes reaches a certain number, which is received as a parameter.

   (b) Finish when each of the root node's probability vector's cells vary less than Δ, between the current and the previous pass.

   (c) Finish when each of the selected nodes' probability vectors' cells vary less than Δ, between the current and the previous pass.

   (d) Finish when each of the terminal nodes' probability vectors' cells vary less than Δ, between the current and the previous

4.2.2 REQ14 Multiple Tree Consensus

Objective

The objective of this requirement is the consensus of a set of trees that have been previously loaded by the user.

Priority

High.

Description

1. **Tree Merge Algorithm:** For the consensus of sets of phylogenetic trees, there exist many well known algorithms, which have already been tested for years. One of those algorithms shall be chosen, adapted and implemented in order to provide tree consensus functionality.

4.2.3 REQ03 Missing Data Handling

Objective

The objective of this requirement is to provide the possibility of loading trees where some of its terminal nodes have have missing data.

Priority

High.

Description

1. **Missing Data Handling:** The most common kind of terminal node that shall be loaded is one that regards to a known location. However, the system shall be able to handle when some terminal node has missing data. The interpretation of this situation is that it's known that certain virus kind arised, but it is not known where.

The system shall provide two ways of handling this kind of nodes:

1. **Every Missing Data is a diferent and unknown location.**
2. **All Missing Data are the same location**

5 Nonfunctional Requirements

5.1 Design Requirements

The product must be implemented on the basis of the object oriented design principles listed below. The first five of them, are also known by the acronym "**SOLID**". A brief description of each design principle is also listed.

- **S**ingle responsibility principle (SRP)
- **O**pen/closed principle (OCP)
- **L**iskov substitution principle (LSP)
- **I**nterface segregation principle (ISP)
- **D**ependency inversion principle (DIP)
- Law of Demeter (LoD)

Also, the product must stick to the ANSI C++ standard, and the "coding style" agreed altogether with FuDePAN.

5.2 Software Attributes

The product's code shall have an 80% coverage through tests.

5.3 Memory Restrictions

No memory constraints are registered, but the reader should notice that given the problems complexity, the system's performance will be directly related to the machine that runs it.

5.4 Operating Enviroments

Phyloloc is intended to be used on standard plataforms, as a desktop application. The supported plataforms are Windows 7 and GNU/LINUX most used distributions.

5.5 License Restrictions

The product shall be licensed as GPLv3.

5.6 GUI Usability Attributes

The GUI shall be color blind enabled.

Appendix

A Glossary

- **ITBA**: Instituto Tecnológico de Buenos Aires
- **FuDePAN**: Fundación para el Desarrollo de la Programación en Ácidos Nucleicos.
- **EFPU**: Estación de Fotobiología Playa Unión
- **API**: Application Programming Interface.
- **GPL**: General Public License.
- **IEEE**: Institute of Electrical and Electronics Engineers
- **SOLID**: Acronym introduced by Robert C. Martin, representing five basic object oriented design and programming principles.
- **Single responsibility principle** SRP states that every object should have a single responsibility, and that responsibility should be entirely encapsulated by the class. All its services should be narrowly aligned with that responsibility.
- **Open/closed principle** OCP states that software entities (classes, modules, functions, etc.) should be open for extension, but closed for modification, that is, such an entity can allow its behaviour to be modified without altering its source code.
- **Liskov substitution principle** LSP is a particular definition of a subtyping relation, called (strong) behavioral subtyping, and states that given $q(x)$ to be a property provable about objects x of type T, then $q(y)$ should be true for objects y of type S where S is a subtype of T.
- **Interface segregation principle** ISP supports the idea that many client specific interfaces are better than one general purpose interface.

- **Dependency inversion principle** DIP states that High-level modules should not depend on low-level modules, both should depend on abstractions, and that abstractions should not depend upon details, details should depend upon abstractions.
- **LoD**: Law of Demeter, object oriented principle used to accomplish loose coupling.
- **Missing Data**: When a node in a phylogenetic tree has an unknown location.

B References

- C++: Programming language.
  `http://www.cplusplus.com`
- QT: C++ library for user interface development.
  `http://qt.nokia.com/products`
- GPL: General Public License.
  `http://www.gnu.org/licenses/gpl.html`
- IEEE STD 830-1998: Guide for the Software Requirements Specification
  `http://standards.ieee.org/reading/ieee/std_public/description/se/830-1998_desc.html`
- SOLID: "Design Principles and Design Patterns", Robert C. Martin.
  `http://www.objectmentor.com/resources/articles/Principles_and_Patterns.pdf`
- "Phyloloc: Combining phylogenetic and geographic information to untangling the spread of emerging pathogens", Leandro Jones, Daniel Gutson
- RFC 2119: Standard practice to indicate requirement levels.