VIP

Vicariance Inference Program

User's reference manual

Version 11.4.28

J. Salvador Arias

CONICET, INSUE, Facultad de Ciencias Naturales e Instituto Miguel Lillo, Miguel Lillo 205, San Miguel de Tucumán (4000), Tucumán, Argentina

jsalarias@csnat.unt.edu.ar

1. Introduction

The biogeographical analysis within a single lineages (“phylogenetic biogeography”, “taxon history biogeography”, “single lineage biogeography”) has been subject of several methods based on the concept of ancestral areas (e.g. Fitch optimization [Fi71] or DIVA [Ro97]). Unfortunately, such methods rest on the idea of predefined areas that are optimized in some way into a phylogenetic trees. There are several problems on the use of predefined areas [Ho97][Ho01], that ultimately end in the doubtful implementation of such methods.

Based on insights from Rosen [Rs78], Hovenkamp [Ho97][Ho01] propose a method to deal directly with spatial data in a phylogenetic context. Although his objective is more akin to “cladistic biogeography” (a search for history of earth), ideas from Hovenkamp are useful for use in phylogenetic biogeography. The program present here, VIP (from Vicariance Inference Program) uses ideas the ideas of Hovenkamp in a optimality criterion framework allowing the user to set reconstruction costs, use heuristic searches and visualize the results in a geographically defined graphic interface.

1.1 Data requirements

VIP requires two types of primary information. First, it requires a cladogram depicting phylogenetic relationships among the study taxa. The second type of data is georeferences (latitude, longitude) from terminals. Each terminal must have a georeferenced point to be taken into account in the analysis. Additionally, in the case of time calibrated phylogenies (i.e. an analysis including fossils) is possible to assign an age to terminals and nodes.

A detailed account on data formats and data edition can be found in the following sections.

1.2 Program background

Hovenkamp [Ho97][Ho01] argued that the only evidence left from an speciation process in a geographic context are allopatric distributions. Unfortunately, although many clades show an allopatric distribution, many does not. There are some distributions that present some form of overlap, others in that a widespread terminal or clade make uninformative several of its ancestral nodes.

Some of these problems are also found in the “area-based” biogeography. Area based biogeography assumes that distributions can be resumed as few homogeneous areas. VIP does not take that approximation, but it uses some solutions developed under such paradigm [Pa94a][Pa94b]. If a distribution of a node is ignored, and it is possible to found a solution with more nodes showing allopatric distributions, this solution might be preferred as it increases the explanatory power of allopatry. But also, removing a node is a form of an *ad-hoc* rejection of conflicting evidence.

VIP uses a system of costs, in which the user can set the cost of a node distribution removal with respect to the cost of a non-vicariant node (a node that can not be explained geographically). If distribution removal has a cost of 0, the result if the maximum number of possible nodes explained by allopatry, that is “maximum vicariance” [Pa94b][Ro03]. If removal has an infinite cost, the result is the maximum number of possible nodes explained without removal, as in the original Hovenkamp's method [Ho97][Ho01] and similar to “assumption 0” [Ho97] or “paralogy free” subtrees [Ho01][NL96]. Note that in a dichotomic tree, if removals are set to a cost of 1.0, then the results will be equal (in terms of cost and reconstruction) to set the cost as 0.0, as each removal imply a non-vicariant node (and, if removals have some cost, the program does not count the removal and the non-vicariant node), but this equality does not hold in the presence of polytomies (in which a removal does not imply that the parent node became non-vicariant).

Details about the method are described in [AEip] as “spatial analysis of vicariance.”

1.3 Results

The results of an analysis with VIP is displayed in two windows, the map and the tree window. In the map window, allopatric distributions are displayed using red and blue for each descendant, if there are some overlap, it is show in green. It is possible to show a potential barrier, calculated using the Voronoi tessellation. In a tree window, nodes with allopatric distribution are marked with a black square. Nodes with removed distributions are shown as white circles. In some cases, a node is removed, but it also has an allopatric distribution, in such cases, a white square is displayed in the node.

1.4 Running VIP

VIP can be downloaded as binary or in form of source code. Binaries for Linux and Windows are available.

1.4.1 Linux binaries

To run linux version you need to install the gtk-bin package. This package is already installed in Gnome desktop. For KDE, you can install this package using synaptic, get-app or a similar tool from your preferred repository.

1.4.2 Windows binaries

To run the program you must install the gtk windows runtime. The latest version can be downloaded here (http://sourceforge.net/projects/gtk-win/).

Also, it is possible to compile the code yourself. The source is write on GNU C, using gtk. Make sure you have the latest version of gtk-dev library (available here: http://www.gtk.org/download.html, linux users can download and install it with synaptic, app-get or a similar application). In windows you need to link the program including the following libraries gdk-win32.lib, gdk_pixbux.lib and pango.lib.

1.5 License and citation

As any scientific work, if you use this program please cite the method description [AEip], Hovenkamp original proposal [Ho97][Ho01], and the program itself. Also I suggest including algorithms or methods used/modified by the program (such as [Pa94b]) where applicable.

1.6 Contact

I will be happy to hear you, for any questions, suggestions, bug reports, your VIP user experience or your code modifications. You can contact me at: jsalarias@csnat.unt.edu.ar. If you like, I also can include you in a list of VIP users, so I can send you reminders of new available versions.

For bug reports, please specify how the bug is triggered, and if possible, a data set that produce the error.

Also, I will be happy if you send me a pdf copy of your published results using VIP.

1.7 Acknowledgments

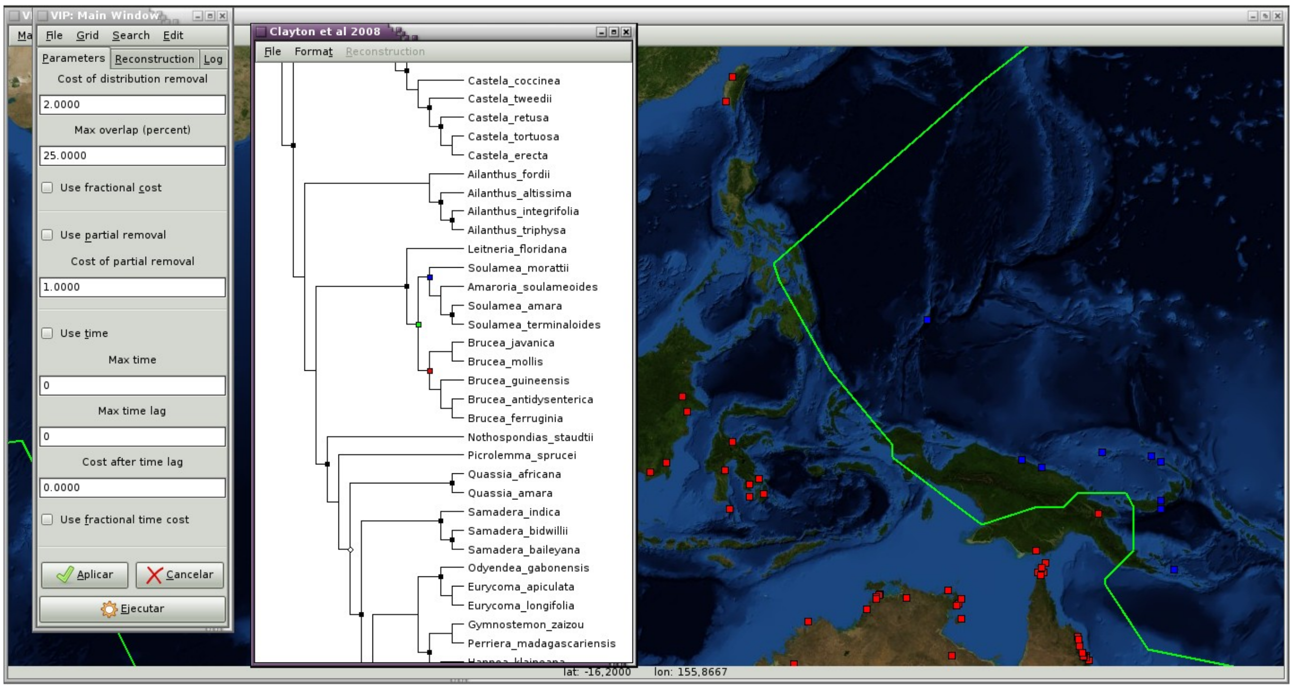
Pablo Goloboff and Claudia Szumik provide strong support, encouragement and guidance during this project. Peter Hovenkamp, provide support, interest and helpful comments in several stages of the development. Dolores Casagranda, Santiago Catalano and Marcos Mirande, always provide interest and suggestions.

This work was funded thanks to a CONICET doctoral fellowship, a FONCyT grant (PICT 1314 to P.A. Goloboff).

Maps displayed on this manual were taken from NASA blue marble web site (http://earthobservatory.nasa.gov/Newsroom/BlueMarble/).

2. General layout

VIP is a program that uses a Graphic User Interfaz (GUI) based on different windows (Fig. 2.1). There are three basic windows. The main window has the menus and parameter entries for basic operations, such as data loading, searches, reconstructions and data edition. The tree window displayed the actual tree and node, it has menus for basic tree display. The map window displayed the position of records for the selected node, and a map, if any; it has menus for map configuration, and record display.



Cost of distribution removal
2.0000
Max overlap (percent)
25.0000
☐ Use fractional cost
☐ Use partial removal
Cost of partial removal
1.0000
☐ Use time
Max time
0
Max time lag
0
Cost after time lag
0.0000
☐ Use fractional time cost

lat: -16.2000 lon: 155.8667

Figure 2.1 VIP interface under linux. Map from NASA blue marble.

In all windows, there are keyboard shortcuts for all menu operations.

In this section, a general overview of the layout is provided, more detailed accounts (based mostly on menu actions) are given in the following chapters.

2.1 Main Window

The main window is used for user input (files and search/reconstruction parameters), and for a basic report of the results.

It has four menus (Fig 2.1). **File menu** is similar to most program's file menu, it is used to load input data, export data and exit the program. **Grid menu** is activated when there are some data, it is used to set the grid. **Search menu** is activated when a grid is active, it is used for searches and reconstruction operations. **Edit menu** is activated when there some data, it is used to some basic edit operations for the records and reconstructions.

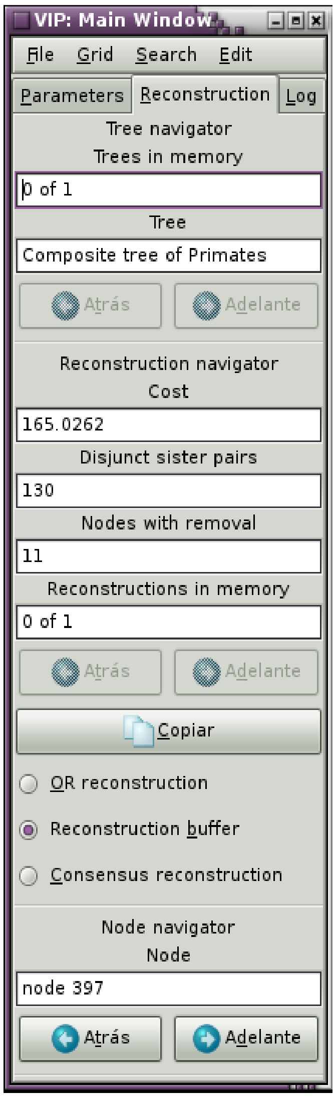
The windows has three tabs. The first tab is label as **parameters**. It is used to set the parameters used during the reconstruction (Fig 2.2). The box **cost of distribution removal** is used to set the cost of a distribution removal. This cost are set in relationship to non vicariant nodes (always with a cost of 1.00). The cost of the reconstruction is the sum of the cost assigned to each node. The **max overlap** box sets the maximum (as percentage) acceptable overlap for a node to be counted as allopatric. The check box **use fractional cost** is used to give more grain to the reconstruction. If fractional costs are used, then an allopatric node might be with a cost if it has some overlapping, the cost is proportional to the overlap, for example, with a an overlap of the 10% the cost of a vicariant node is 0.1. Removals, as well as non-vicariant nodes will have always counted using their respective costs.

The box **cost of partial removal** is used to set the cost of a partial distribution removal.

The **use partial removal** check box allow the program to search using a partial removal, instead of a full removal. That is, in an allopatric node, the distribution of one of its descendants is removed from the distribution of its ancestor, then the node remains allopatric. This is an experimental option, and just implemented for basic searches. The cost of this late overlap is set with the **cost of partial removal** box.

If data is time calibrated, then the **use time** check box is activated. With this option checked, the searches take into account the age difference between the descendants (based on [HU01], Arias, in prep.). The **max time** box, is used to set the maximum acceptable age difference between the descendants and the ancestor for a vicariant distributed (beyond that, the node is counted as non-vicariant). The **max time lag** box, is used to set the maximum age difference between the descendants and the ancestor to count the node as vicariant. Greater differences (that are smaller than max time) will be penalized with a cost set in the **Cost after time lag** box. This cost must be between 0 (no extra cost after time lag) to less than 1 (the node is almost non-vicariant). With **Use fractional cost check box**, the extra cost is assigned as the rate between timelag and max time.

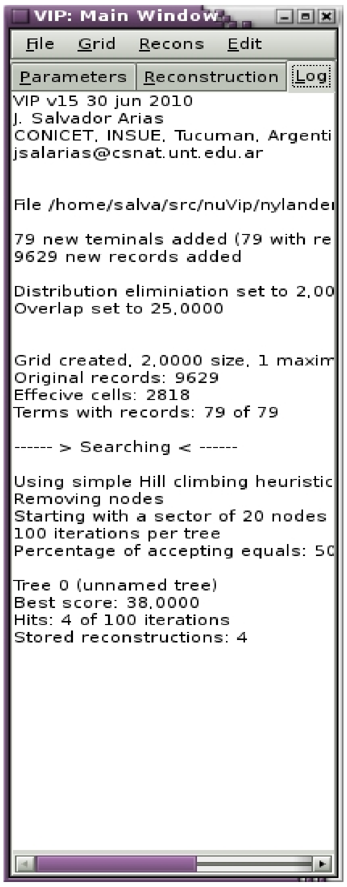
Apply and **cancel** buttons are used to set the values, or the return to the previous ones. **Execute** button is equivalent to the menu Recons\Search... and perform the search using the actual parameters (equivalent to apply and then search).



Trees in memory: 0 of 1
Tree: Composite tree of Primates
Cost: 165.0262
Disjunct sister pairs: 130
Nodes with removal: 11
Reconstructions in memory: 0 of 1
OR reconstruction [✓ Reconstruction buffer] Consensus reconstruction
Node: node 397

Figure 2.2 Parameters tab on Main Window

The **reconstruction** tab (Fig. 2.1) is used to show some basic statistics of the actual reconstruction, and to navigate across the data. The **tree navigator** shows the number of **trees in memory**, and the name of the **actual tree**. With the buttons **back** and **forward** is possible to change the tree. The **reconstruction navigation** box shows a **cost** box with the actual cost of the reconstruction, the **disjunct sister pairs** box show the number of nodes in which descendants are disjunct in the reconstruction, the box **nodes with removal** indicates the number of nodes with a removed distribution. The **reconstructions in memory** shows the number of reconstructions found with the searches, and if a searched reconstruction is selected indicates the id (number) of the **actual reconstruction**, in this case, the **back** and **forward** button allows the user to change the reconstruction. The button **copy** allows the user to copy the actual reconstruction. The option entries **OR reconstruction**, **Searched reconstruction** and **Consensus reconstruction** are used to select among reconstruction kinds. The **node navigator** shows the label of the **actual node**, it can be changed using the **back** and **forward** buttons.



```
VIP v15 30 jun 2010
J. Salvador Arias
CONICET, INSUE, Tucuman, Argenti
jsalarias@csnat.unt.edu.ar


File /home/salva/src/nuVip/nylande

79 new teminals added (79 with re
9629 new records added

Distribution eliminiation set to 2,00
Overlap set to 25,0000


Grid created, 2,0000 size, 1 maxim
Original records: 9629
Effecive cells: 2818
Terms with records: 79 of 79

------ > Searching < ------

Using simple Hill climbing heuristic
Removing nodes
Starting with a sector of 20 nodes
100 iterations per tree
Percentage of accepting equals: 50

Tree 0 (unnamed tree)
Best score: 38,0000
Hits: 4 of 100 iterations
Stored reconstructions: 4
```

Fig 2.3 Log tab on Main Window

The **log** tab, stores the log of a running season in text format, so it can be copied into the clipboard. The log is helpful to keep notice of a particular running season, so it can be repeated in any other time.

2.2 Tree view window

The tree view window (Fig 2.1) is used to visualize the actual tree, and the reconstructions, if any. The title of the window change to reflect the name of the actual tree.

To move the tree inside the window it is possible to use arrow keys, or holding the mouse left button (drag), also, if there are a mouse wheel, it can be used to scroll the tree vertically. Tree size can be changed with arrows keys and Ctrl button. To quick size Ctrl + F fit the tree to the actual window, and with Ctrl + T fit the space between terminals to be readable.

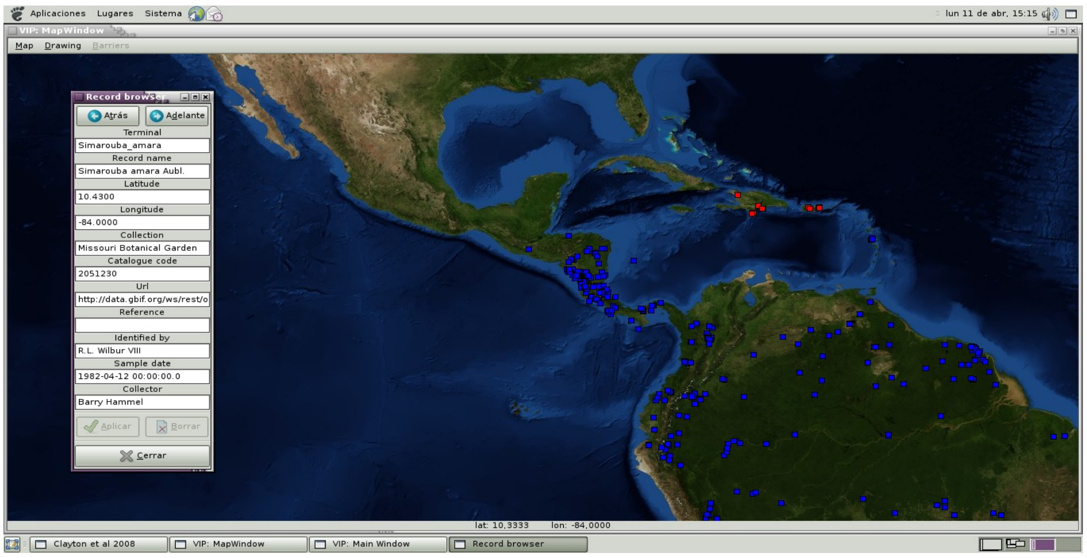
With the mouse left button is possible to select nodes by a click. Also using page up and page down, the selected node change. The selected node is marked with a green square. If the node is a vicariant node, then their descendants are marked with a red and blue square. With a click of the right button, information about the node is displayed, and the name of the node can be edited. Clicking the right button with the shift key will prompt a dialog box to set the fill of the descendants of this particular node.

If there are a reconstruction, then the vicariant nodes are displayed as black squares. If the node span is removed is displayed with a white circle. If the node span is removed but also is a vicariant node, then it is show with a white square.

2.3 Map window

The map window (Fig. 2.1) is used to display the records of the selected node.

The records are displayed as green squares. If the node is a vicariant node, then the records of each descendant are displayed in red and blue, according to each descendant (in coordination with the colors used in the tree view window), in such cases, the overlap is show in green. In the case of consensus reconstructions, not all records are included, so they are showed in white.



Terminal	Simarouba_amara
Record name	Simarouba amara Aubl.
Latitude	10.4300
Longitude	-84.0000
Collection	Missouri Botanical Garden
Catalogue code	2051230
Url	http://data.gbif.org/ws/rest/o
Reference	
Identified by	R.L. Wilbur VIII
Sample date	1982-04-12 00:00:00.0
Collector	Barry Hammel

lat: 10,3333 lon: -84,0000

Fig 2.4 Record browser. Map from NASA blue marble.

With a click on the left button of mouse over a record, information of the particular record is displayed (Fig. 2.4). By holding the right button, it is possible to move the map. At the bottom of the mouse window, there is a label showing the latitude and longitude at the mouse pointer, if it is on the map.

3. File operations (Main Window > File menu)

Basic file operations are packed in the File menu, at Main Window, this operations behave nearly identical to file operations in most GUI applications.

3.1 Open (Ctrl + O)

This menu shows a file chooser dialog, that allows the user to select a file to be open. VIP open their own XML files, and KML files that use folders as nodes. Although just only one file can be open each time, several files might be open, allowing the user to use multiple trees.

3.2 Format input

A simple XML files are used as input by VIP. It aslo can read some KML (Google earth) files if the trees are in a set of folders. It is not expected that the user edit directly this files. A simple TNT [GE08] macro is provided (toxml.run) to write a phylogenetic tree into a XML file. Then using VIP, the point localities of each terminal or the feed records menu can be use to input the georeferenced data.

A full specification of the XML format used is explained in the appendix.

3.3 Feed records (Ctrl + X)

It is possible to load full batches of data into VIP using this option. Data can be loaded if they are on the same XML format used by VIP, in some KML files, in which records are stored as placemarks, XYD format used by NDM [Go08] and tab delimited text files. The condition is that the name of the terminal to be feed match exactly (i.e. case sensitive, same length) with a terminal name or synonym.

A sample table is included with the examples, the user can edit it with any spreadsheet program, and then save it as tab delimited text.

After a file is feed, VIP stores two files, unmatch.xml and unfeed.xml. The first file contains the names of unmatched records. VIP match records by a perfect matching with their names, this file is produced so it can be easily edited to be feed again, and allowing VIP to recognize other names. The second file, unfeed, stores the records that are not feed, this allows the user to re-feed them again, without duplicating data.

3.4 Save data as (Ctrl + Shift + S)

It is possible to save the data in the actual session of VIP in XML format (for example, after edition) with all terminals or only the terminals with records, in KML (to be open with

Google earth), in XYD (to be open with NDM), and as tab delimited tables. Also is possible to save an alphabetic list of the used terminals and an alphabetic list of source collections (if available) in plain text format.

3.5 Open results (Ctrl + Shirf + O)

This menu is activated in presence of a grid. It reads the results (i.e. the reconstructions) from a previous run. It uses a particular XML format detailed in the appendix.

3.6 Save results (Ctrl + S)

This menu is activated in presence of a grid. It save the actual reconstructions in memory in a XML file.

3.7 Close (Ctrl + W)

This option removes all the loaded data from memory, without closing the program.

3.8 Quit (Ctrl + Q)

Exit the program.

4. Grid operations (Main Window > Grid menu)

VIP uses a grid to represent the spatial distribution of taxa. The grid menu is used to set the basic properties of the grid.

4.1 Grid settings (Ctrl + G)

This menu sets the size of each grid, and the maximum filling. The grid always start at -180 lon, 90 lat. The fill is the number of cells around the record to be counted as presences (similar to [AE09][Go08]). For example, a fill of 0, only the cells in which a record fall are counted with a presence, a fill of 1, the eight cells surrounded the cell in which the record fall as presences. Fill values (smaller or equal than the maximum fill) might be defined to each node (an its descendants) by pressing the shift and the right button of mouse in the tree view window.

Instead of trying several starting points of the grid, it is better to use an small grid size, and a fill. For example, instead of a 5 degree cells, using a 1.66 degree cells and a fill of 1, make a virtual 5 degree cell for each record.

The filling neighborhood can de defined with the **Moore's neighborhood** and **Von Neumann's neighborhood** radio buttons. Moore neighborhood uses the Chebychev's distance to fill the grid (like the movements of a king in a chess board), whereas von Neumann neighborhood uses the Manhattan distance.

The settings of the grid can only be modified if there is no grid created.

4.2 Set fill in all nodes (Ctrl + I)

By default, all nodes are set using the maximum fill, with this option all nodes will be set to the fill indicated (if the value is smaller or equal to the maximum fill). Its use is to remove the user defined fills in several nodes.

Note that VIP uses an algorithm to compress the data, but this compression is performing with the maximum fill value. So using a fill for all nodes smaller than maximum fill will produce a waste of memory that slows down the searches. An advantage, if that it is not necessary to destroy the grid (and losing reconstructions).

4.3 Create grid (Ctrl + A)

The grid set does not creates the grid, just set the rules for the grid creation. The create grid menu creates a grid in memory. As the grid is compressed to save memory and computation time, record edition is unavailable after the grid is created. When the grid is created, a default OR reconstruction is performed on the data, activating several of the information boxes in the reconstruction tab.

4.4 Redo grid (Ctrl + B)

If some fill values are changed, this menu updates the grid to reflect the new fill values, and automatically check the reconstructions in memory.

4.5 Destroy Grid (Ctrl + D)

Removes the grid and the reconstructions from memory. The absence of grid is reflected on reconstruction tab showing “no grid” in several fields. Without a grid, it is possible to edit records.

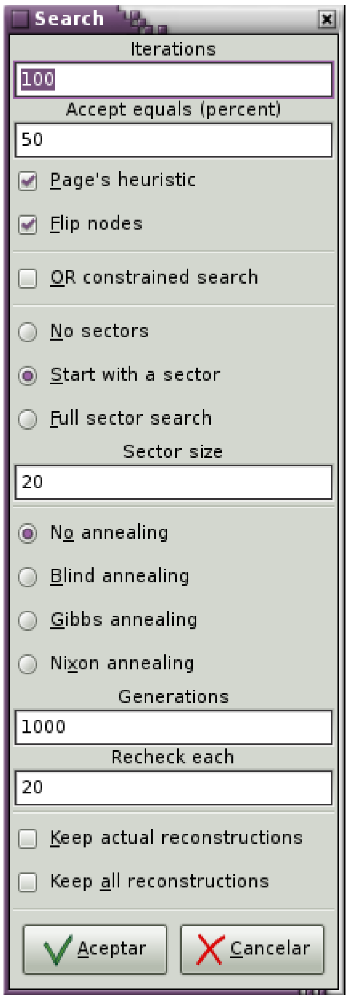
4.6 Save grid table (Ctrl + M)

After the grid was created is possible to save the information of the grid in form of text (tab delimited) tables. This include an NDM matrix [Go08] for endemism analysis (although, I recommend to save direct coordinates, with File>Save data, 3.4). Also, KML and Excel files can be saved to show the amount of species per grid square.

5. Reconstruction operations (Main Window > Recons menu)

This is the basic menu of Vip. It encapsulates the basic operations about reconstructions: searching, consensus and filtering reconstructions.

5.1 Heuristic search (Button Execute; Ctrl + E)



Iterations
100
Accept equals (percent)
50
☑ Page's heuristic
☑ Flip nodes
☐ QR constrained search
○ No sectors
● Start with a sector
○ Full sector search
Sector size
20
● No annealing
○ Blind annealing
○ Gibbs annealing
○ Nixon annealing
Generations
1000
Recheck each
20
☐ Keep actual reconstructions
☐ Keep all reconstructions

Fig 5.1 Search dialog

This menu displays the search dialog (Fig. 5.1). During an heuristic search a determinate algorithm is used the indicated number of iterations setting with the **Iterations** text box. Searches stop when no better reconstruction is found, it is possible to accept solutions with the same score using a probability, that is set in **Accept equal (percent)** text box, this probability is setting as a percentage (i.e. between 0 and 99). Take into account that not accepting equals make the finding of the optimal harder, and being to flexible, just slow down the search.

The searches were based on the successive removal of nodes [Pa94a],[Pa94b], that are setting with the **Page's heuristic** check box (After the algorithm in [Pa94b]). By default the box is checked so, all possibilities are checked before accepting a new reconstruction to start the next round. If not checked, the program runs quicker, but the search is more softer, because when a new reconstruction is better it is accepted immediately.

The successive removal, might be set with the **Flip nodes** check box. if the box is checked, instead or just removing nodes, all nodes are examined in each round, if the node is removed, then it is restored as non removed, if the node is not removed, then it is removed. Otherwise, it just remove a node, and then examines if the removal produce a better score. Note that just removing nodes and using Page's heuristic is exactly the full algorithm of Page [Pa94b]. Flip nodes is more slower (as it checks all nodes in each round) but produce a more exhaustive search that just removing nodes.

There are several search strategies. By default **no sectors** and **no annealing** is used. If **start with a sector** is checked, instead of removing all nodes in the first round of the search, a group of nodes is removed, when the results of this sector can not be improved the search continues in the traditional manner. This is useful if there are some particular combinations of nodes that quickly produce a local optimum, making that the same set of nodes are selected in each replication. With a **full sector search**, the whole data set is examined with sectors. The **sector size** text box allows the user to set the size of the sector. The selection of the nodes on a sector is at random. This procedure is based on sectorial searches proposed for parsimony by [Go99].

Annealing searches accept non-optimal solutions with the hope to scape of a local optimum. In a **blind annealing**, the difference between the actual solution and the potential new one is used to determinate the probability to accept the new solution. In VIP that probability is based on the probability of accept equals. This procedure is based on traditional annealing [KE83] and tree drifting [Go99]. With **Gibb's annealing**, several possibilities are tried, and their probability of acceptance stored, and summed. This cumulative probability is used to determinate the new state of the reconstruction, the procedure is based on [HE08] implementation of Gibb's sampling. **Nixon's annealing**, is based on the idea of parsimony ratchet [Ni99], a group of nodes is removed, then the best reconstruction is search, the removed nodes are restored, and a full round of node flipping is performed. With annealing the number of modifications are set with **Generations** text box. It is important to note, that when using annealing, the flip nodes is used automatically (otherwise reconstructions will be always worse than the originals).

By default, each search cleans the reconstructions in memory, if **keep actual reconstructions** is checked, they are preserved (unless a new optimal was found). By default, only optimal solutions are stored, if **keep all reconstructions** is checked then a reconstruction from each iteration is saved in the reconstruction's buffer.

5.2 Consensus (Ctrl + C)

After a search, the consensus of the found reconstructions can be calculated. Note that before each search, the consensus is cleaned. After every search consensus is only calculated the first time that this menu (or the consensus reconstruction option on reconstruction flap) is selected, any other selection of this options just shows the previous consensus.

5.3 Filter (Ctrl + F)

Removed all reconstructions with cost greater than an extra-cost assigned by the user, and clean the consensus. If the extra cost is set as 0, then all suboptimal reconstructions are removed. It can be helpful, for example, how is the best reconstruction found, under other cost parameter setting.

5.4 Clean (Ctrl + N)

Removes all reconstructions in memory.

5.5 Clean consensus (Ctrl + U)

Removes the actual consensus from memory, so a new consensus will be calculated the next time.

6. Edit operations (Main Window > Edit menu)

This menu contains some basic edition options. This instructions are only available under certain circumstances.

6.1 Hot mouse (Ctrl + H)

Checking this option allows the user to edit manually the records for each terminal. It is only active when there are no active grid. A terminal must be selected. Records can be added on the map window by pressing the right mouse button holding the shift key. By clicking the right mouse button and holding the Ctrl key over a record, a georeference browser dialog is displayed, showing the selected point, it is possible to change the coordinates or delete records using this dialog box.

6.2 Edit reconstruction (Ctrl + T)

When a searched reconstruction is selected, pressing the right mouse button holding the Ctrl key over a node (in the tree window) change its status (removed or not removed and vice verse). Two points must to be taken into account. First, the selected recostruction is lost, and second partial overlapping is not calculated (so if the changed node has some partial removal, this information is lost).

6.3 Georeference browser (Ctrl + R)

Displays the georeference browser (Fig 2.4). If the grid is not active and Hot mouse option is selected is possible to edit georeferenced records.

6.4 Save log (Ctrl + L)

Save the content of the actual log in a text file.

6.5 Clean log (Ctrl + K)

Clean the log, this is automatically performed when closing data.

7. Tree file operations (Tree window > File menu)

This menu allows the user to save the tree.

7.1 Tree name (Ctrl + I)

Change the name of the tree displayed on screen.

7.2 Save tree (Ctrl + Shift + T)

Save the tree displayed on the screen as a xml file using PhyloXML [HZ09] format. Note, just the tree (and no any other associated data) is saved with this option.

7.3 Save tree graphic (Ctrl + Shift + G)

Save an SVG file of the actual tree, following the actual format options. The SVG file is an open format for vector graphics (as a tree), allowing an easy way to produce high quality graphics. There are several programs to edit/export/render this files.

8. Tree window menus (Tree window > Format menu)

This menu allows the user to configure the shape and sorting of the tree. This options only affects how the tree is drawn. It can be used to show the tree with **aligned terms** (Ctrl + A) or **ragged terms** (Ctrl + R). When the tree is aligned it is possible to **pull nodes to root** (Ctrl + E) or **pull nodes to terminals** (Ctrl + O).

The tree can be sorted using the “weight” of its descendants (the number of terminals of the node). With **sort up** (Ctrl + U), lighter terminas are shown at the top, where as with **sort down** (Ctrl + D) they are shown at the bottom. With the **mixed sorting** option (Ctrl + M) some nodes are sorted up, where other are sorted down. With the **default sorting** (Ctrl + G), the tree is sorted as it was read.

As the size of the tree might be larger than the window, there are quick ways to fit it to a particular size. With **fit tree to window** (Ctrl + F) the tree is resized to the window size. On the other hand, with **fit tree to terms** (Ctrl + T), the tree is sized to allow a clear reading of the terminal names.

9. Tree reconstruction operations (Tree window > Reconstruction menu)

This menu is used to show reconstructions when the consensus reconstruction is selected.

After making the consensus, the program check if there are other available nodes that share the same vicariant distribution than the selected node. To be available, a node must be on an independent clade to the selected node (i.e. it can not be a descendant or an ancestor). This is equivalent to search for a “supported vicariance event” [Ho97].

But different from [Ho97], the search is node based, so instead of finding an event, it find other nodes that share a common allopatric pattern with the selected node. This in that sets might be no reciprocal. For example, a node A share an allopatric pattern with node B and node C, but node C is a descendant of B, so node C will not be valid on the common allopatric set of B.

9.1 All reconstructions (Ctrl + L)

The default option, shows the reconstruction of all nodes.

9.2 Common reconstructions (Ctrl + N)

This option shows only the reconstructions that share the same common allopatric pattern as the selected node.

9.3 Supported reconstructions (Ctrl + S)

This options shows only the reconstruction off all nodes that share a common allopatric pattern with another node. Note that given the nature of the set reconstruction, the showed node might not be simultaneously supported.

9.4 Bremer's support (Ctrl + B)

When the consensus reconstruction is activated, this option shows the Bremer values for each disjunction. This values were calculated during searches, so to save them, use the graphic save of the tree (7.3). This are Bremer values, because are based on Bremer support [Br88], that is, they show the difference in fit among the best reconstruction that shows the descendants of the node as disjunct vs the fit of the best reconstruction in which descendants of the node are overlapped. The greater the value the best is the support for the disjunction. Sometimes it was impossible to found a reconstruction with a particular node as overlapped, then it is indicated with the symbol of greather than and a question mark.

10. Map operations (Map Window > Map menu)

This menu is used to basic map operations, such as loading, and scale.

10.1 Open (Ctrl + O)

Load an image to be used as map. Most popular graphic formats as jpg, gif, bmp and png are supported. The map must be in a cylindrical and isometric projection (i.e. equirectangular projection), to coincide with the window coordinate system.

10.2 Map limits (Ctrl + L)

This menu display a map limit dialog. It is used to set the limits of the actual map, so the map will be put on the right coordinates, and in adequate the scale. Note that the limits are just for the loaded image. Records can be scattered in any valid geographic point of the earth.

In absence of loaded image, it can be used to change the scale of the displaying window (it is assumed than an image of 720 x 360 pixels is loaded).

10.3 Centre on (Ctrl + C)

Center the window at the selected coordinates.

10.4 Save map (Ctrl + Shirf + S)

Save an image (in jpeg format) or a KML map of the actual (visualized) distribution. In the case of KML map files, if it a reconstruction, all vicariant nodes of the actual reconstruction can be saved in the same file.

11. Drawing operations (Map Window > Drawing menu)

This menu is used to change the display on the map.

11.1 View records (Ctrl + R)

When the grid is active, this option toggle the visualization of records. It is the only option available if there is no grid.

11.2 View cells (Ctrl + E)

When the grid is active, it is possible to see the actual cells. If there are a huge amount of points, It would produce faster redrawing.

11.3 Record radius (Ctrl + U)

Allows the user to change the size used to draw records. The radius is measured in pixels. It also change the mouse radius sensibility above records.

11.4 All records (Ctrl + A)

If checked (the default) shows all records, records that did not pertains to the reconstruction are showed as void boxes. If unchecked, just the records assigned to the actual reconstruction are showed.

11.5 Removed records (Ctrl + M)

If checked, it shows records that altough associated with the node are not included in the reconstruction. This only happens if partial removal is active, or when examining a consensus recontruction. The removed records are drawn in white.

11.6 Barrier (Ctrl + B)

If checked, shows the potential barrier associated with the selected node, if the node is reconstructed as vicariant (see Barriers menu).

11.7 Grid (Ctrl + G)

If checked, display the grid based on the actual grid settings.

12. Barrier operations (Map Window > Barriers)

This menu is active only when the barrier option (Map Window > Drawing > Barrier; 11.5) is checked. It change the display and properties of barrier drawing.

Barriers are calculated using the Voronoi-Delaunay dual graph on the records reconstructed for the selected node. But instead of storing all triangles/cells, just the triangles that contain points from each descendant are kept. Points on overlapping areas are not taken into account for the triangulation.

12.1 Voronoi lines (Ctrl + V)

When this option is selected, a line emulating the Voronoi tessellation is showed. Note that the line did not necessary fall in the "correct" barrier, it is only a devise to easily locate the potential geographic barrier associated with the disjunction. As the Voronoi segments are mid-distance lines, farther points produce less meaningful barriers.

12.2 Delaunay trinangles (Ctrl + D)

When this option is selected, the triangles of the Delaunay triangulation are showed. This is equivalent to show the space between the disjoint distributions.

12.3 Mid triangles (Ctrl + I)

Instead of using voronoi lines, the line that split the triangles is shown.

12.4 Barriers on ancestors (Ctrl + N)

If checked, display the barriers detected on parent nodes of the selected node. The ancestral barriers are displayed as broken and slim lines.

12.5 Common barriers (Ctrl + S)

If checked, and a consensus reconstruction is selected, it display the barrier associate with the common allopatric set associated with the node (see under Tree window > Reconstruction menu) as a thin line. It uses a very conservative procedure, using the disjunctions associated with all nodes in the allopatric set, even if these nodes are not simultaneously supported.

Appendix 1. Keyboard shortcuts

Keyboard shortcuts in alphabetic order. Note that shortcuts are only available in the window with the focus (the active window).

In parenthesis it is indicated the section that explain the particular menu entry.

Main window

Ctrl + A	Create grid	(4.3)
Ctrl + B	Redo grid	(4.4)
Ctrl + C	Consensus	(5.2)
Ctrl + D	Destroy grid	(4.5)
Ctrl + E	Heuristic search	(5.1)
Ctrl + F	Filter	(5.3)
Ctrl + G	Set grid	(4.1)
Ctrl + H	Hot mouse	(6.1)
Ctrl + I	Set fill in all nodes	(4.2)
Ctrl + K	Clean log	(6.5)
Ctrl + L	Save log	(6.4)
Ctrl + M	Save grid table	(4.6)
Ctrl + N	Clean	(5.4)
Ctrl + O	Open	(3.1)
Ctrl + Shitf + O	Open results	(3.5)
Ctrl + R	Georeference browser	(6.3)
Ctrl + Q	Quit	(3.8)
Ctrl + S	Save results	(3.6)
Ctrl + Shift + S	Save data as	(3.4)
Ctrl + T	Edit reconstruction	(6.2)
Ctrl + U	Clean consensus	(5.5)
Ctrl + W	Close	(3.7)
Ctrl + X	Feed records	(3.3)

Tree window

Ctrl + A	Aligned terms	(8)
Ctrl + B	Bremer's support	(9.4)
Ctrl + D	Sort down	(8)
Ctrl + E	Pull nodes to terms	(8)
Ctrl + F	Fit tree to window	(8)
Ctrl + G	Default sorting	(8)
Ctrl + Shirf + G	Save tree graphic	(7.3)
Ctrl + I	Tree name	(7.1)
Ctrl + L	All reconstructions	(9.1)
Ctrl + M	Mixed sorting	(8)
Ctrl + N	Common reconstructions	(9.2)
Ctrl + O	Pull nodes to root	(8)
Ctrl + R	Ragged terms	(8)
Ctrl + S	Supported reconstructions	(9.3)
Ctrl + T	Fit tree to terms	(9)
Ctrl + Shift + T	Save tree	(7.2)

Ctrl + U	Sort up	(9)
Ctrl + (arrow)	Resize the three	(2.2)

Map window

Ctrl + A	All records	(11.4)
Ctrl + B	Barrier	(11.6)
Ctrl + C	Centre on	(10.3)
Ctrl + D	Delaunay triangles	(12.2)
Ctrl + E	View cells	(11.2)
Ctrl + G	Grid	(11.7)
Ctrl + I	Mid triangles	(12.3)
Ctrl + L	Map limits	(10.2)
Ctrl + M	Removed records	(11.5)
Ctrl + N	Barriers on ancestors	(12.4)
Ctrl + O	Open	(10.1)
Ctrl + R	View records	(11.1)
Ctrl + S	Common barriers	(12.5)
Ctrl + Shift + S	Save map	(10.4)
Ctrl + U	Record radius	(11.3)
Ctrl + V	Voronoi lines	(12.1)

Appendix 2. XML reference

This appendix contains an alphabetical reference of the XML elements used and recognized by VIP. It is important to note, that although the files generated by VIP are valid XML files, the parser used by VIP is not as strict as “real” XML parsers and do not make any external validation. Therefore there is no an specific Document type definition or similar schema for the XML elements described here.

The described elements are the ones that VIP can interpret. It is possible to include more elements than the one described here, and as long as they remain as a valid XML elements, VIP ignores them silently.

As the VIP parser is very simple, it does not support namespaces. That is, it takes the whole name inside the element as a single name. In the case of files downloaded from GBIF as DarwinCore, that uses namespaces, they are integrated as part of the keyword in the parser, so this files can be read without problem.

Several elements are borrowed from KML (http://schemas.opengis.net/kml/), PhyloXML (http://www.phyloxml.org/), and DarwinCore (http://rs.tdwg.org/dwc/). That elements can be inserted on VIP documents, and VIP would interpret it as their own file, VIP interpreted them as synonyms of the elements described here. If fact, it is possible to combine several types without harm (unless, of course, that that might leave the file unreadable for other programs). Nevertheless VIP output uses exclusively the elements referred here.

A2.1 <Clade>

See <Node>

Remarks

Element borrowed from PhyloXML. Never used by VIP (unless a PhyloXML file was saved), and read it as a <Node>.

A2.2 <Document>

See <Vip>

Remarks

Element borrowed from KML. Never used by VIP (unless a KML file was saved), and read it as <SpecimenInfo>.

A2.3 <Folder>

See <Node>

Remarks

<Folder> is a synonym of <Node>. In general, VIP save the data using <Node>, if a tree is also saved. When just geographical/specimen information is saved, it uses <Folder>.

A2.4 <Node>

Syntax

```
<Node id=”unsigned integer”>
        <name>...</name>                <!-- string -->
        <age>...</age>                          <!-- unsigned integer -->
        <synonym>...</synonym>          <!-- string -->

        <!-- 0 or 2 or more Folder/Node/Clade elements -->
        <!-- 0 or more SpecimenInfo, Placemark, Point elements -->
</Node>
```

Description

<Node> is the basic node of a phylogenetic tree, that is, keep hierarchical information. In case that a <Node> element appears in the file outside of a <Tree>, <Phylogeny>, <Folder>, <Clade> or <Node> element, a new tree will be created to store the node. If an id is not indicated, or has an error, then an automatic id is assigned. If no tree element is open, then a new tree from the scratch will be created. The first node read works as the root of the tree.

Elements

<name>
The name assigned to the node. It is assumed (but not checked) that the name is unique. Otherwise, when feeding data, feed data will be assigned to the first node found.

<age>
The age assigned to the node. It start from 0 (present) and increase in the past direction. For example, if units are MYA, then the age of the end of Cretaceous is 65.

<synonym>
A node element can contain 0 or more <synonym> elements. When feed the data, a name will be search on the name list and on the synonym list. Then, it is assumed (but not checked) that strings on <synonym> element are unique, and also different from any valid <name> element. Otherwise, when feed the data it will be assigned to the first string found. Note that <synonym> are not taxonomic synonyms, but local synonyms in the context of the actual data.

Example

```
<?xml version="1.0" encoding="UTF-8" ?>

<Vip>
```

```
    <Tree>
        <name>Some dinosaurs</name>
        <Node id=”3”>
            <name>Saurischia</name>
        <Node id=”0”>
            <name>Aves</name>
        </Node>
        <Node id =”4”>
            <name>Sauropoda</name>
        <Node id=”1”>
            <name>Diplodocus</name>
            <age>150</age>
            <synonym>Seismosaurus</synonym>
        </Node>
        <Node>
            <name>Apatosaurus</name>
            <age>150</age>
            <synonym>Apatosaurus ajax</synonym>
            <synonym>Apatosaurus excelsus</synonym>
            <synonym>Brontosaurus</synonym>
            <SpecimenInfo>
                <name>Apatosaurus ajax</name>
                <Point>
                    <coordinates>-105.2,39.7</coordinates>
                </Point>
            </SpecimenInfo>
        </Node>
        </Node>
        </Node>
    </Tree>
</Vip>
```

Synonyms
<Node>, <Folder>, <Clade>.

Contains

<Node>, <Folder>, <Clade> elements. When none of this elements is included, the node is interpreted as a terminal, otherwise it require 2 or more descendants (i.e. an internat node of a phylogeny).

<SpecimenInfo>, <Placemark> elements, if the node is a terminal (i.e. no <Node>, <Folder>, <Clade> elements present).

A2.5 <Phylogeny>

See <Tree>

Remarks

Element borrowed from PhyloXML. Never used by VIP (unless a PhyloXML file was saved), and read it as a <Node>.

A2.6 <Placemark>

See <SpecimenInfo>

Remarks

Element borrowed from KML. Never used by VIP (unless a KML file was saved), and read it as <SpecimenInfo>.

A2.7 <Point>

Syntax

```
<Point>
        <coordinates>...</coordinates>            <!-- lon,lat -->
</Point>
```

Description

The spatial position of a record.

Elements

<coordinates> (required)

A tuple of real valued numbers that represent the geographic location of the point. The values accepted are longitude, and latitude (in that order) separated by a comma, without spaces. Values uses degrees with decimal point (without minutes, seconds notation). Longitude must be between -180 and 180 (negative values for West hemisphere), and Longitude between -90 and 90 (negative values for South hemisphere).

Example

```
<?xml version="1.0" encoding="UTF-8" ?>

<Vip>
        <Tree>
                <Node>
                <Node>
                        <name>Aves</name>
                </Node>
                <Node>
                        <name>Sauropoda</name>
                <Node>
                <name>Diplodocus</name>
            </Node>
            <Node>
                <name>Apatosaurus</name>
                <SpecimenInfo>
                    <name>Apatosaurus ajax</name>
                    <Point>
                        <coordinates>-105.187,39.653</coordinates>
                    </Point>
                </SpecimenInfo>
            </Node>
            </Node>
            </Node>
        </Tree>
</Vip>
```

Contained by

<SpecimenInfo>, <Placemark> elements.

A2.8 <SpecimenInfo>

Syntax

```
<SpecimenInfo>
    <name>...</name>                        <!-- string -->
    <collection>...</collection>            <!-- string -->
    <catalog>...</catalog>                  <!-- string -->
    <url>...</url>                          <!-- string -->
    <reference>...</reference>                  <!-- string -->
    <identifiedBy>...</identifiedBy>        <!-- string -->
    <collectionDate>...</collectionDate>        <!-- dateTime -->
    <collector>...</collector>              <!-- string -->

    <!-- 1 Point element -->
</SpecimenInfo>
```

Description

Specimen info contains the geographical datum of a particular specimen. Apart of the geographical data, most parts of this element are used for user's bookkeeping. So the user will be able to trace back each distribution point to an specific source.

Elements

<name>

The name assigned to specimen. If this name is different from the name of the parent, then it will be added to the list of synonyms of the node.

<collection>

The name of the collection where the specimen is deposited.

<catalog>

The catalog number/code used to refer the specimen in the collection.

<url>

A URL for the record in question, if any (note: VIP does not validate the url or its syntax).

<reference>

A reference to a publication source of the record (for example, a taxonomic revision).

<identifiedBy>

The name of the person/team that identify the specimen.

<collectionDate>

Date of sample collection (note: VIP does not validate the syntax).

<collector>

The name of the person/team that collect the specimen.

Example

```
<?xml version="1.0" encoding="UTF-8" ?>

<Vip>
        <Tree>
                <Node>
                <Node>
                        <name>Aves</name>
                </Node>
                <Node>
                <Node>
                        <name>Diplodocus</name>
                </Node>
                <Node>
                        <name>Apatosaurus</name>
                        <SpecimenInfo>
                                <name>Apatosaurus ajax</name>
                                <collection>Paleobiology database</collection>
                                <catalog>408913</catalog>
                        <url>http://data.gbif.org/35342819</url>
                                <reference>Ostrom, McIntosh, 1999</reference>
                                <identifiedBy>Marsh, O.</identifiedBy>
                                <collectionDate>1877</collectionDate>
                                <collector>Lakes, A.</collector>
                                <Point>
                                        <coordinates>-105.187,39.653</coordinates>
                                </Point>
                        </SpecimenInfo>
                </Node>
                </Node>
                </Node>
        </Tree>
</Vip>
```

Synonyms

<Placemark> works as a synonym of <SpecimenInfo>, although the <Placemark> element as defined in KML has a more narrow scope for collection metadata.

Contains

<Point> element (1 and just 1 required).

Contained by

<Node>, <Folder>, <Clade> elements.

A2.9 <Tree>

Syntax

```
<Tree>
        <name>...</name>                  <!-- string -->
        <!-- 1 Folder/Node/Clade element -->
</Tree>
```

Description

<Tree> is a container of explicit phylogenies, so the hierarchy of the folders is taken into account when data is read. When data is feed, the internal hierarchy of the tree is ignored (but not their elements). If a <Tree>, <Phylogeny>, <Folder> or <Clade> element is open, it is an error.
There must be a <Folder>, <Node> or <Clade> element inside a tree. Otherwise, the tree is ignored.

Elements

<name>
A string assigned to name the tree. If no string is assigned, VIP automatically set it as "unnamed tree x", where x is the number of the actual number of trees in memory (starting from 0).

Example

```
<?xml version="1.0" encoding="UTF-8" ?>

<Vip>
        <Tree>
                <name>A simple tree, (a (b c))</name>
                <Node>
                <Node>
                        <name>a</name>
                </Node>
                <Node>
                <Node>
                        <name>b</name>
                </Node>
                <Node>
                        <name>c</name>
                </Node>
                </Node>
                </Node>
        </Tree>
</Vip>
```

Synonyms

<Node>, <Folder> or <Clade> if no <Tree>, <Phylogeny>, <Node> or <Folder> element is open.

Contains

<Node>, <Folder>, <Clade> elements (1 and just 1 required).

A2.10 <Vip>

Syntax

```
<Vip>
        <!-- 0 or more Tree/Phylogeny/Forder/Node/Clade elements -->
</Vip>
```

Description

<Vip> is a container of the phylogenies and distributions that VIP program can read. It is the root element of the file. This element is required.

Example

```
<?xml version="1.0" encoding="UTF-8" ?>

<Vip>
        <Folder>
        <Folder>
                <name>a</name>
        </Folder>
        <Folder>
        <Folder>
                <name>b</name>
        </Folder>
        <Folder>
                <name>c</name>
        </Folder>
        </Folder>
        </Folder>
</Vip>
```

Synonyms

<Document>

Contains

<Tree>, <Phylogeny>, <Folder>, <Node>, <Clade> elements.

Appendix 3. VRX reference

Apart from the XML schema used for input the data, VIP also uses a XML format to store and read results of an analysis. Although such files are not intended to be read by a human, it might be useful to understand them (for example for feeding it to a third party application). The actual limitations of the XML parser of VIP (explained in the appendix 2) are also applicable in this context.

A3.1 <DistroElimination>

Syntax

```
<DistroElimination>
        <nodeID>...</nodeID>            <!-- number -->
</DistroElimination>
```

Description

<DistroElimination> is an element to indicate that a particular node is removed in a reconstruction.

Elements

<nodeID>
The id of the node to be eliminated in the reconstruction. This must be coincident with the id used to identify the node in VIP.

Example

```
<?xml version="1.0" encoding="UTF-8"?>

<VipResults>

<Reconstruction>
        <treeID>0</treeID>
        <DistroElimination>
                <nodeID>10</nodeID>
        </DistroElimination>
</Reconstruction>

</VipResults>
```

A3.2 <PartialElimination>

Syntax

```
<PartialElimination>
        <nodeID>...</nodeID>              <!-- number -->
</PartialElimination>
```

Description

<PartialElimination> is an element to indicate that a particular node is removed in a partial way in the reconstruction.

Elements

<nodeID>

The id of the node to be eliminated (partially) in the reconstruction. This must be coincident with the id used to identify the node in VIP.

Example

```
<?xml version="1.0" encoding="UTF-8"?>

<VipResults>

<Reconstruction>
        <treeID>0</treeID>
        <PartialElimination>
                <nodeID>7</nodeID>
        </PartialElimination>
</Reconstruction>

</VipResults>
```

A3.3 <Reconstruction>

Syntax

```
<Reconstruction>
        <treeID>...</treeID>              <!-- number -->
        <!-- 0 or more DistroElimination elements -->
        <!-- 0 or more PartialElimination elements -->
</Reconstruction>
```

Description

<Reconstruction> is a container of a particular reconstruction in a given tree.

Elements

<treeID>

The number used to identify the tree, this number must be coincident with the number used to store the trees in VIP. This element is obligatory.

Example

```
<?xml version="1.0" encoding="UTF-8"?>

<VipResults>

<Reconstruction>
        <treeID>0</treeID>
        <DistroElimination>
                <nodeID>11</nodeID>
        </DistroElimination>
        <PartialElimination>
                <nodeID>7</nodeID>
        </PartialElimination>
</Reconstruction>

</VipResults>
```

Contains

<DistroElimination>, <PartialElimination> elements.

A3.4 <VipResults>

Syntax

```
<VipResults>
        <!-- 0 or more Reconstruction elements -->
</VipResults>
```

Description

<VipResults> is a container of the reconstructions used by VIP. It is the root element of the VRX file. This element is required.

Example

```
<?xml version="1.0" encoding="UTF-8"?>

<VipResults>

<Reconstruction>
        <treeID>0</treeID>
        <DistroElimination>
                <nodeID>10</nodeID>
        </DistroElimination>
</Reconstruction>
```

```
<Reconstruction>
        <treeID>0</treeID>
        <DistroElimination>
                <nodeID>11</nodeID>
        </DistroElimination>
        <PartialElimination>
                <nodeID>7</nodeID>
        </PartialElimination>
</Reconstruction>
```

</VipResults>

Contains

<Reconstruction> elements.

References

[AEip] J.S. Arias, C.A. Szumik, P.A. Goloboff. In press. Spatial analysis of vicariance: a method for using direct geographical information in historical biogeography. Cladistics. Doi: 10.1111/j.096-0031.2011.00353.x.

[AE09] L. Aagesen, C.A. Szumik, F.O. Zuloaga, O. Morrone (2009) Quantitative biogeography in the South America highlands - recognizing the Altoandina, Puna and Prepuna through the study of Poaceae. Cladistics 25, 295-310.

[Br88] K. Bremer (1988) The limits of amino-acid sequence data in angiosperm phylogenetic reconstruction. Evolution 42, 795-803.

[Fi71] W.M. Fitch (1971) Toward defining the course of evolution: Minimum change for a specific tree topology. Systematic Zoology 20, 406-416.

[GE08] P.A. Goloboff, J.S. Farris, K.C. Nixon (2008) TNT, a free program for phylogenetic analysis. Cladistics 24, 774-786. Program available at: http://www.zmuc.dk/public/phylogeny/tnt/

[Go99] P.A. Goloboff (1999) Analyzing large data sets in reasonable times: solutions for composite optima. Cladistics 15, 415-428.

[Go08] P.A. Goloboff. (2008) NDM, program distributed by the author, available at: http://www.zmuc.dk/public/phylogeny/Endemism/

[HE08] J.P. Huelsenbeck, C. Ané, B. Larget, F. Ronquist (2008) A bayesian perspective on a non-parsimonious parsimony model. Systematic Biology 57, 406-419.

[Ho97] P. Hovenkamp (1997) Vicariance events, not areas, should be used in biogeographical analysis. Cladistics 13, 67-79.

[Ho01] P. Hovenkamp (2001) A direct method for the analysis of vicariance patterns. Cladistics 17, 260-265.

[HU01] C.A. Hunn, P. Upchurch (2001) The importance of time/space in diagnosing the causality of phylogenetic events: Towards a "Chronobiogeographical" paradigm? Systematic Biology 50, 391-407.

[KE83] S. Kirkpatrick, C.D. Gelatt, M.P. Vecchi (1983) Optimization by simulated annealing. Science 220, 671-680.

[Ro03] F. Ronquist (2003) Parsimony analysis of coevolving species associations. In: R.D.M. Page (Ed.) Tangled trees: phylogeny, cospeciaton, and coevolution, Chicago: Chicago Univ. pp. 22-64.

[Ro97] F. Ronquist (1997) Dispersal-vicariance analysis: A new approach to the quantification of historical biogeography. Systematic Biology 46, 195-203.

[Rs78] D.E. Rosen (1978) Vicariant patterns and historical explanation in biogeography. Systematic Zoology 27, 159-188.

[Pa94a] R.D.M. Page (1994) Maps Between Trees and Cladistic Analysis of Historical Associations among Genes,Organisms, and Areas. Systematic Biology 43, 58-77.

[Pa94b] R.D.M. Page (1994) Parallel phylogenies: Reconstructing the history of host-parasite assemblages Cladistics 10, 155-173.

[Ni99] K.C. Nixon (1999) The parsimony ratchet, a new method for rapid parsimony analysis. Cladistics 15, 407-414.

[NL96] G.J. Nelson and P. Ladiges (1996) Paralogy in cladistic biogeography and analysis of paralogy-free subtrees. American Museum Novitates 3167, 1-58.