Data Formats in RevBayes

Reading, Manipulating and Writing Data

1 Overview

This tutorial describes the data formats that are used in `RevBayes`.

Requirements

We assume that you have read and hopefully completed the following tutorials:

- RB_Getting_Started

2 Molecular Sequence Data

2.1 Getting Started

- Download data and output files from: http://revbayes.github.io/tutorials.html

- Open the file **`primates_cytb.nex`** in your text editor. This file contains the nucleotide sequences of the cytochrome B gene sampled from 13 species (Box 1). The elements of the **`DATA`** block indicate the data type, number of taxa, and sequence length.

Box 1: A fragment of the NEXUS file containing the ITS sequences for this exercise.

```
#NEXUS

Begin data;
Dimensions ntax=13 nchar=673;
Format datatype=DNA missing=? gap=-;
Matrix
Trig_excelsa
TCGAAACCTG...
Fagus_engleriana
TCGAAACCTG...
Fagus_crenata1
TCGAAACCTG...
Fagus_japonica2
TCGAAACCTG...
Fagus_japonica1
TCGAAACCTG...
Fagus_orientalis
TCGAAACCTG...
Fagus_sylvatica
TCGAAACCTG...
Fagus_lucida1
TCGAAACCTG...
Fagus_lucida2
TCGAAACCTG...
Fagus_crenata2
TCGAAACCTG...
Fagus_grandifolia
TCGAAACCTG...
Fagus_mexicana
TCGAAACCTG...
Fagus_longipetiolata
TCGAAACCTG...
       ;
End;
```

2.2 Loading Molecular Sequence Data

We can read the data into `RevBayes` using the **`readDiscreteCharacterData()`** function.

```
data <- readDiscreteCharacterData("data/primates_cytb.nex")
```

2.3 Querrying Dataset Attributes

When a dataset has been loaded into `RevBayes`, we can query relevant `Rev` variables. To report the current value of any variable, simply type the variable name and press enter. For example, the **`data`** variable returns general information about the sequence alignment:

```
data
   DNA character matrix with 23 taxa and 1141 characters
   =====================================================
   Origination:                   primates_cytb.nex
   Number of taxa:                23
   Number of included taxa:       23
   Number of characters:          1141
   Number of included characters: 1141
   Datatype:                      DNA
```

The **`data`** variable has *member functions* that we can use to retrieve specific attributes of the dataset. These member functions include the number of taxa (**`data.ntaxa()`**), the sequence length (**`data.nchar()`**), etc.

```
data.ntaxa()
   23
```

Available *member functions* for the data variable are listed in Table 1.

2.4 Concatenating Sequences

We can combine two or more datasets using the **`concatenate`** function. First, we will read in two datasets; the first is an alignment of primate cytb sequences, the second is an alignment of cox2 sequences:

```
data_cytb <- readDiscreteCharacterData("data/primates_cytb.nex")
data_cox2 <- readDiscreteCharacterData("data/primates_cox2.nex")
```

Next, we will concatenate these two alignments using the **`concatenate`** function. This returns a single data matrix that combines the sequences of both gene regions.

```
data <- concatenate(data_cytb, data_cox2)
```

We can confirm this by querying the **`data`** variable:

Table 1: Available member functions for the data variable.

Function name	Type
chartype	String function ()
excludeAll	void function ()
excludeCharacter	void function (Natural)
excludeCharacter	void function (Natural [ ])
getEmpiricalBaseFrequencies	Simplex function ()
getNumInvariantSites	Natural function ()
includeAll	void function ()
includeCharacter	void function (Natural)
includeCharacter	void function (Natural [ ])
ishomologous	Bool function ()
methods	void function ()
names	String [ ] function ()
nchar	Natural function ()
ntaxa	Natural function ()
removeTaxa	void function (String)
removeTaxa	void function (String [ ])
setCodonPartition	void function (Natural)
setCodonPartition	void function (Natural [ ])
setNumStatesPartition	void function (Natural)
setTaxonName	void function (String current, String new)
show	void function ()
size	Natural function ()

```
data
     DNA character matrix with 23 taxa and 1852 characters
 =====================================================
 Origination:                 primates_cytb.nex
 Number of taxa:              23
 Number of included taxa:     23
 Number of characters:        1852
 Number of included characters: 1852
 Datatype:                    DNA
```

2.5 Excluding/Including Taxa

We can exclude species from an alignment that is currently in memory using the **removeTaxa** function. For example, we could exclude the outgroup species *Saimiri sciureus* from our concatenated primate alignment (**data**) by typing:

```
data.removeTaxa("Saimiri_sciureus")
```

We can then confirm the removal of a species by checking the number of remaining taxa:

```
data.ntaxa()
   22
```

The number of species has decreased by one, as expected. We can confirm that we have excluded *Saimiri sciureus* by typing:

```
data.names()
    [ "Callicebus_donacophilus", "Cebus_albifrons", "Alouatta_palliata", ...]
```

2.6 Excluding/Including Sites or Genes

We can exclude a single site (or set of sites) from an alignment that is currently in memory using the **excludeCharacter** function. For example, we could exclude the first site in our concatenated primate alignment (**data**) by typing:

```
excludeCharacter([1])
```

[Note that sites of an alignment are indexed from 1 to N.] We can confirm the removal of a site by checking the number of remaining sites:

```
data.nchar()
   1851
```

The number of sites has decreased by one, as expected. We can return the excluded site to our alignment using the **includeCharacter** function:

```
includeCharacter([1])
```

We can similarly exclude/include a range of sites, *e.g.*, corresponding to a gene region. Here, we will exclude all 1141 sites comprising the cytb gene region from our concatenated alignment:

```
data.excludeCharacter(1:1141)
```

We can check the number of remaining sites, which comprise the cox2 gene region:

```
data.nchar()
   711
```

We can easily return the excluded cytb sequences by typing:

```
data.includeCharacter(1:1141)
```

It is also possible to exclude/include all sites using the **`excludeAll`** and **`includeAll`** commands.

3 Biogeographical Data

3.1 Nexus file

The data file contains a matrix of binary characters corresponding to the observed ranges of the study taxa.

→ Open the file `examples/psychotria_range.txt`.

```
#NEXUS

begin data;
  dimensions ntax=19 nchar=4;
  format datatype=standard symbols = "01";
  matrix
    P_mariniana_Kokee2 1000
    P_mariniana_Oahu  0100
    ...
    P_hexandra_Oahu   0100
  ;
end;

Begin trees;
       TREE tree1 = ((((((((P_hawaiiensis_WaikamoiL1:0.9656850499,
       P_mauiensis_Eke:0.9656850499):0.7086257935,(P_fauriei2:1.23
       0218511,P_hathewayi_1:1.230218511):0.4440923324):0.17671155
       ...
       89):0.4630447802,P_hexandra_Oahu:2.826939991):2.372081244);
End;
```

Geographic range data is stored in standard Nexus format. In the `data` block, the first line gives the dimensions of the data matrix and the second line indicates we will be using binary characters. The four characters correspond to areas defined by the geography file (next subsection). Rows in the `matrix` block correspond to taxa and their geographic range data, while columns give in which areas each taxon is present (1) or absent (0). For example, the geographic range of taxon `P_hexandra_Oahu` is restricted to area 2.

The `trees` block gives the tree describing the shared ancestry of the study species. Because range evolution occurs in units of geological time, the analysis in this tutorial requires a high-quality, time-calibrated phylogeny. This typically requires a multiple sequence alignment over several loci and fossils for calibration. Since the availability of such data typically limits the taxa we can include in a biogeographic analysis, it is best to estimate the time-calibrated phylogeny first. Only afterwards should you begin assembling range data for your biogeographic study. If your phylogeny cannot be calibrated (e.g., if fossils are unavailable) your best alternative is to proceed with a time tree resulting from a divergence time estimation analysis. For this tutorial, the phylogeny is assumed to contain no uncertainty.

3.2 Atlas file

Here, we focus our tutorial on the Hawaiian archipelago. Beneath Hawaii, currently the largest and youngest island, is a volcanic hotspot that periodically creates new islands. The ages of these islands are fairly well known, meaning we can model range availability as a function of time. Following **?**, we will lump groups of smaller islands into single areas to simplify the analysis, leaving us with four areas: Hawaii (H), Oahu (O), Maui (M; this includes Molokai and Lanai), and Kauai (K; this includes Niihau). These areas are modeled have arisen 0.5, 1.9, 3.7, and 5.5 million years ago, respectively.

Although the model will use discrete-state biogeographic ranges, geographical area is naturally continuous. This means we must impose some discretization upon the geography to designate a set of biogeographically meaningful characters called areas. Different methods use different criteria for this discretization, so it is best to perform the discretization yourself rather than blindly using the discretization given from a previous study or method (but do blindly use the dataset included in this tutorial). Some geographic problems are more amenable to discretization: for instance, the Hawaiian archipelago forms naturally discrete areas on the basis of islands. For many geographic problems, however, it may unclear how to perform this discretization. Much like morphological analyses, you might decide to choose areas based on expert opinion, based on some model, or using some "naive" uniform discretization. This procedure is not part of the tutorial, but you should be aware that area definitions are not always obvious or objective.

→ Open the file `examples/hawaii_dynamic.atlas.txt`.

This is called the Atlas file, which uses a file format called JSON (JavaScript Object Notation). JSON is a lightweight format used to assign values to variables in a hierarchical manner. There are three main tiers to the hierarchy in the Atlas file: the atlas, the epoch, and the area. In the lowest tier, each area corresponds to a character in the model and is assigned it's own properties. In the middle tier, each epoch contains the set of homologous areas (characters) that may be part of a species' range, but importantly the properties of these areas may take on different values during different intervals of time, as given by the `start_age` and `end_age` variables. Because the tree and range evolution model also operate on units of geological time, the rates of area gain and loss can condition on areas' properties as a function of time. Sometimes these models are called stratified models or epochal models. Finally, the atlas contains the array of epochs in the highest tier.

Each area is assigned a `latitude` and `longitude` to represent its geographical coordinates, ideally the centroid of the area. If a centroid does not represent the distance between areas, splitting the area into multiple smaller areas is reasonable. The data augmentation approach used in this analysis allows you to effectively specify as many areas as desired, whereas matrix exponentiation methods are limited to approximately ten areas. The distance between the coordinates of any two areas informs the distance-dependent dispersal parameter (β from the $\eta(\cdot)$ function) for range expansion events, so coordinates roughly close to the center of the area suffice. Here, the `latitude` and `longitude` change in each of the four epochs, where they begin at the current location of Hawaii and drift northwesterly until they reach their current positions.

In addition, each area is marked as habitable or not using the `dispersalValues` array. The elements in the array correspond to the other areas defined in the analysis. For example, in `epoch1`, Kauai's `dispersalValues` is equal to `[ 1,0,0,0 ]`, which indicates Kauai exists at that point in time but it is not in contact with any other areas, i.e. the range in that area cannot expand into other areas. The `dispersalValues` for Oahu, Maui, and Hawaii are all equal to `[ 0,0,0,0 ]`, meaning no species may be present in that area during the time interval of e͡poch1 during ages from 10.0 to 3.7. In contrast, `epoch4`, from ages 0.5 to the present, range expansions may occur between any pair of areas and any area may be included in a species' range.

Questions about this tutorial can be directed to:

- Tracy Heath (email: tracyh@berkeley.edu)
- Michael Landis (email: mlandis@berkeley.edu)
- Sebastian Höhna (email: sebastian.hoehna@gmail.com)
- Brian R. Moore (email: brianmoore@ucdavis.edu)

Version dated: February 1, 2015