Exploring Phylogenetic Data:

> Simon J. Greenhill (simon@simon.net.nz).
>
> ARC Centre of Excellence for the Dynamics of Language, Australian National University.
>
> Department of Linguistic and Cultural Evolution, Max Planck Institute for the Science of Human History.

In this section, I want you to learn:

1. About the major file format you'll encounter in phylogenetics: the Nexus file.
2. How to do some simple data quality control checks.

What is a Nexus File?

The most common file format you'll encounter in phylogenetics is a *nexus* file. The filename will usually end in .nex or .trees. Nexus is a very simple text format, that starts with "#NEXUS", and then contains a series of one or more 'blocks'. The blocks start with "BEGIN" … and terminate with "END":

```
#NEXUS

BEGIN DATA;

DIMENSIONS NTAX=3 NCHAR=5;
FORMAT MISSING=? GAP=- DATATYPE=BINARY;

[my dataset]

MATRIX
Tzeltal                   10100
Chontal                   11110
Akateco                   100?1
END;
```

The above excerpt shows a small nexus data block. This tells us that in this datafile we have:

- 3 taxa – here languages (NTAX=3).
- 5 characters (NCHAR=5).
- A definition of the DATATYPE. Here we have binary data, other values here include "DNA",

“NUCLEOTIDE”, “STANDARD”, “MORPH”(ological), etc. The correct value to use here will be determined by the program you want to use. BEAST wants ‘binary’ for our data.

- The character “?” is a missing value (i.e. data we don’t have)
- The names of the taxa are Tzeltal, Chontal, Akateco.
- The data for these languages e.g. Tzeltal = 10100
- Comments are included [within square brackets].

Uppercase or lowercase does not matter.

Tree blocks:

Another block you will come across are tree blocks. This one contains one tree, called “mytree” for three languages:

```
BEGIN trees;
    tree mytree = (Akateco:2,(Chontal:1,Tzeltal:1):1);
END;
```

The format of this tree is Newick (https://en.wikipedia.org/wiki/Newick_format), which has a nested structure of parentheses. This one tells us that Chontal and Tzeltal are more closely linked to each other then Akateco (i.e. they are nested within an extra set of ()’s).

Open the nexus file I gave you in a text editor and have a look:

`[ ] How many taxa/languages?`

`[ ] How many characters?`

`[ ] How many trees? (hint: scroll down)`

Exploring our data in R:

We’re going to do some basic checks and quality control on the data I’ve given you.

Remember: Phylogenetic methods are not magic – they’re only as good as the data you give them: Garbage in = Garbage out.

`[ ] Got it.`

1. Load your data:

Open up `RStudio` (or just `R` if you prefer), and load the nexus data using the `APE` library in R:

```r
library(ape)
library(ggplot2)

nex <- read.nexus.data("cpacific.nex")
tree <- read.nexus("cpacific.nex")
```

`[ ] Load the nexus data into R`

Let's see what that tree looks like. It's the tree that `glottolog` suggests is the most accepted classification for these languages.

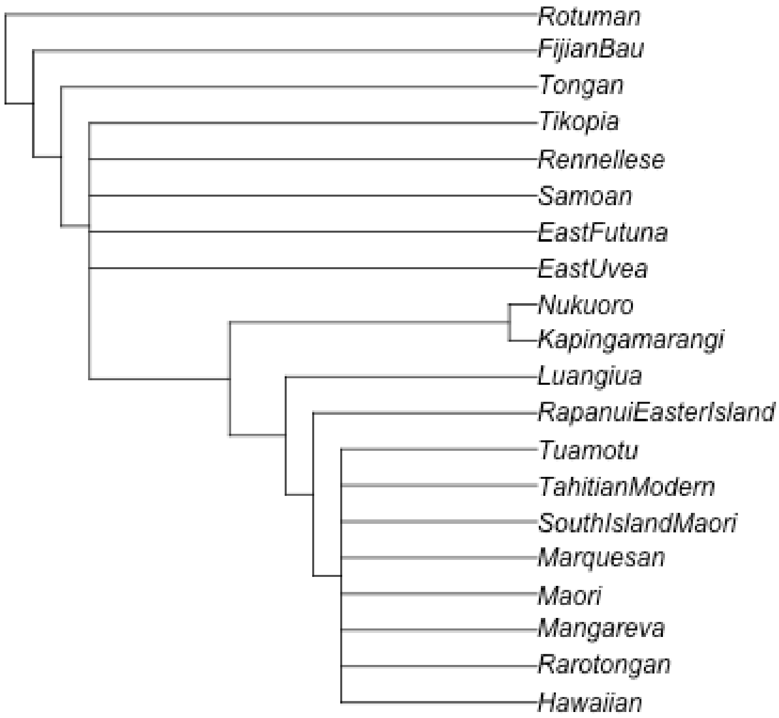
```r
# look at the R representation of the tree. Does it look right?
tree

# we will "ladderize" the tree which just makes it a bit easier to read.
tree <- ladderize(tree)

# plot the tree
plot(tree)
```

`[ ] Plot the tree in R`

It should look something like this:



```
[ ] Which language is most closely related to Kapingamarangi?
```

The biggest subfamily of languages in our data is East Polynesian. It includes Rapanui (Easter Island) and all the other languages in that group.

```
[ ] What are the other East Polynesian languages?
```

2. Look for languages that have low data:

We want to make sure that the languages we're analysing have a good amount of data. Missing data is not necessarily a problem for phylogenetic analysis, it depends on *what's* missing more than *how much* (Wiens '06). However, it's good to take a look and see where problems might be.

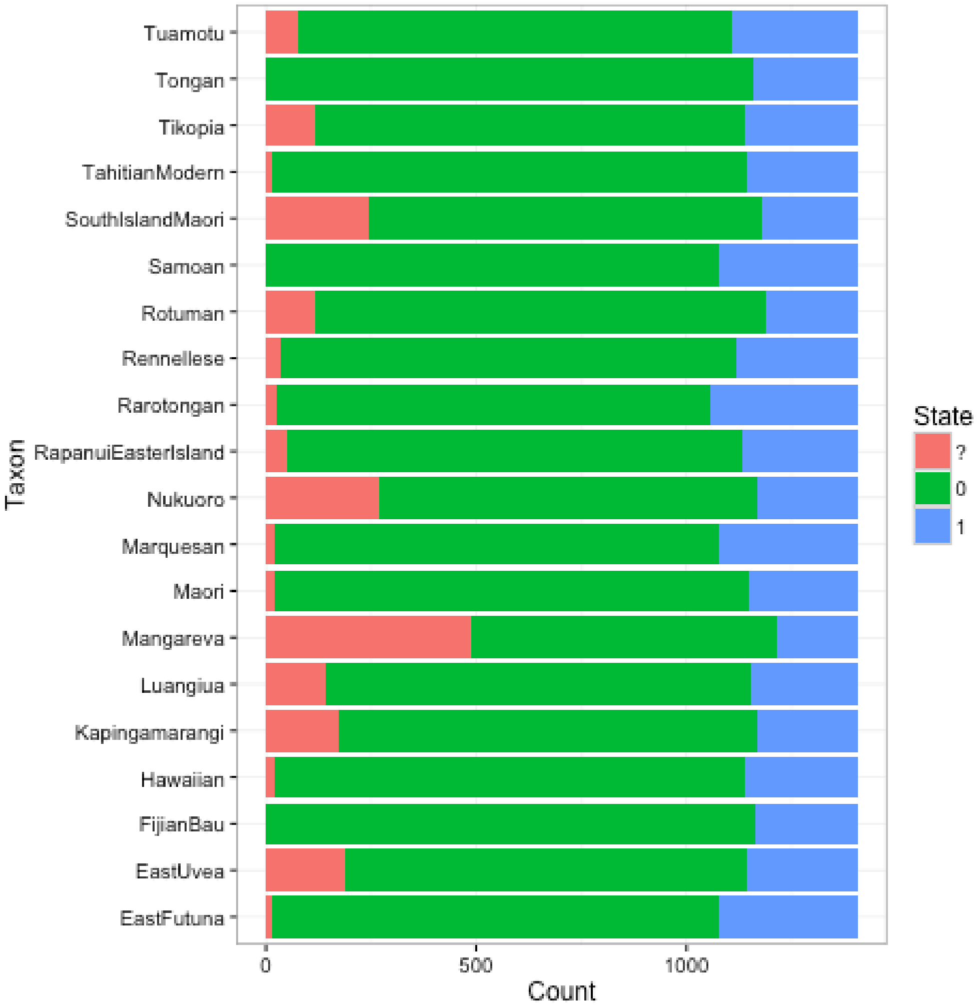
Run this:

```r
#' Taking a nexus file from _APE_'s read.nexus.data
#' function, calculates how many states of the given
#' value are present for each Taxon.
#'
#' Returns a data frame.
#'
#' @param nexus a nexus.data object from read.nexus.data.
#' @return A dataframe of \code{Taxon}, \code{State}, and \code{Count}.
#' @examples
#' nex <- read.nexus.data('filename.nex')
#' statecounts(nex)
statecounts <- function(nexus) {
    # homework problem - figure out a more elegant way to do this.
    out <- data.frame(Taxon=c(), State=c(), Count=c())
    for (taxon in names(nexus)) {
        f = as.data.frame(table(nexus[[taxon]]))
        out <- rbind(out,
            data.frame(Taxon=taxon, State=f$Var1, Count=f$Freq)
        )
    }
    out
}

sc <- statecounts(nex)
ggplot(sc, aes(x=Taxon, y=Count, fill=State)) + geom_bar(stat="identity") + coord_flip
```

`[ ] Plot the state counts for each language`

… You should get something like this:



You will probably see that there are lots more 0's than everything else as there's more ways for a given site to not share the same value than for it to share the same value.

The state values should be reasonably banded (i.e. in this example, most of the languages have around 250 ones (in blue), ~2000 zeros (in green). The missing data state is "?" in red.

Look at the number of '1's – i.e. the amount of cognates in each language.

`[ ] Which languages have the most attested cognates?`

`[ ] Which languages have the fewest attested cognates?`

`[ ] Which languages have the most missing data?`

I think there's a couple of languages I'd worry about here – the ones that have a lot of missing data *and* few cognates.

`[ ] Which languages would Simon worry about?`

Should you remove languages with low data? if you remove them then you lose any information they provide and they may be interesting cases or vital for calibrations etc. If you don't remove them, they may 'float' around the tree and break up other groups.

The solution is to keep an eye on them and see what they do. Often if the languages are poorly attested because it's a bad wordlist or salvage data then you will probably have many of the deep cognates but fewer of the shallow cognate sets, so you will see them drag down to the base of the tree. Or if they've been poorly studied, then the sound changes might be badly understood, so they may have only shallow cognates and will glue onto whichever taxon happens to have the most similarities (hopefully a sister language, if you're lucky).

If these low data taxa don't behave, then delete them, unless you *need* them. If you need them, then consider adding a monophyly constraint to enforce their position in the tree.

Let's leave them in for now, because a bit of risk makes life more fun.

3. Plot the distribution of cognate sizes

How big are our cognate sets?

If they're all really small, then we won't have any deep signal (i.e. we've only got the rapidly changing features). If they're all big, then we won't have any signal for sub(sub-sub) grouping the tips correctly (i.e. we've only got the highly stable features).

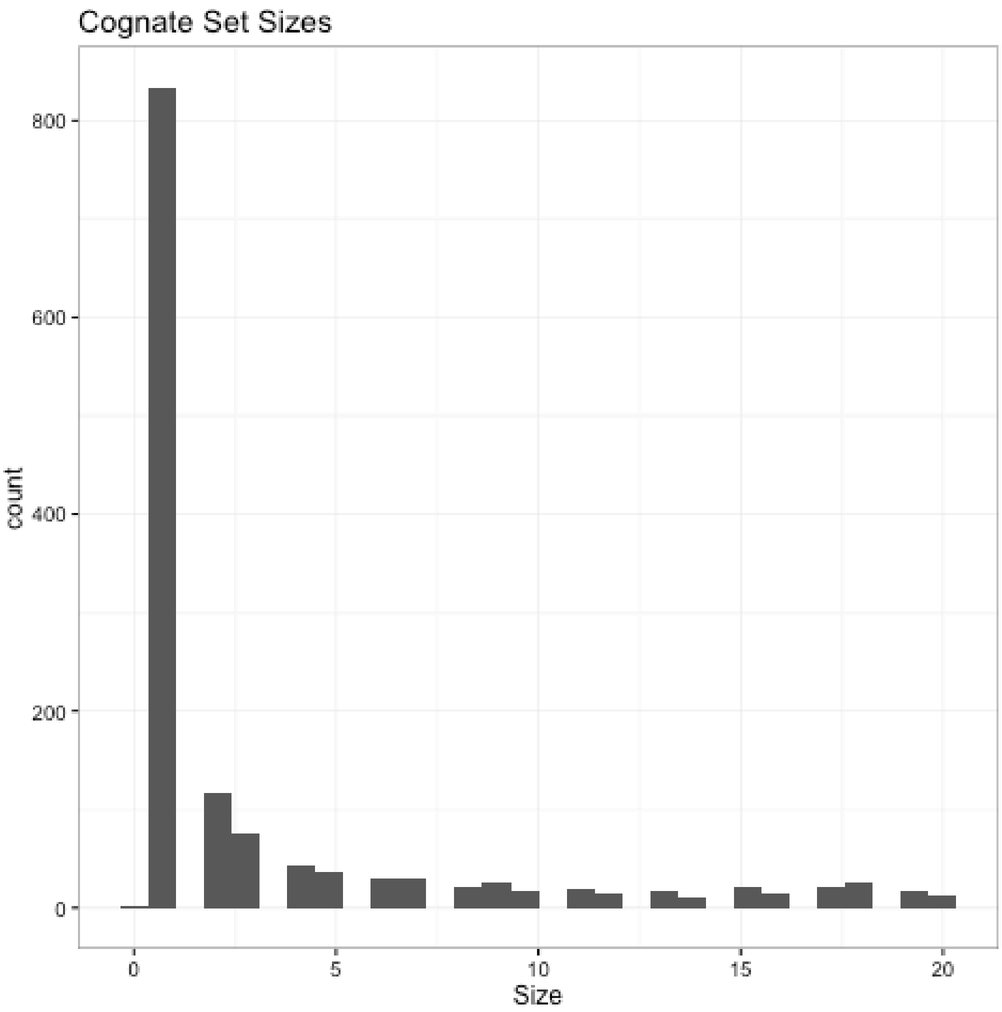
```r
cognatesizes <- function(nex) {
    df <- as.data.frame(nex)

    count <- function(arow) {
        length(arow[arow == '1'])
    }
    out <- data.frame(
        Site=row.names(df), Size=apply(df, 1, count)
    )
    out
}

sizes <- cognatesizes(nex)
qplot(Size, data=sizes, geom="histogram", main="Cognate Set Sizes") + theme_bw()
```

`[ ] Plot cognate size distribution`

You should see something like this:



What type of distribution would we expect to see? I think it should look a bit like the one above, where there are lots of cognates which are shared by one or two sister languages, and fewer cognates that stretch across many languages.

You may see other patterns. If you see very few cognates up the right side of the graph, then you have few cognates that are shared across most/all languages. That is, you have little deep signal and you will find it hard to resolve the deepest groupings in the phylogeny. This will happen if you've got lots of weakly related languages.

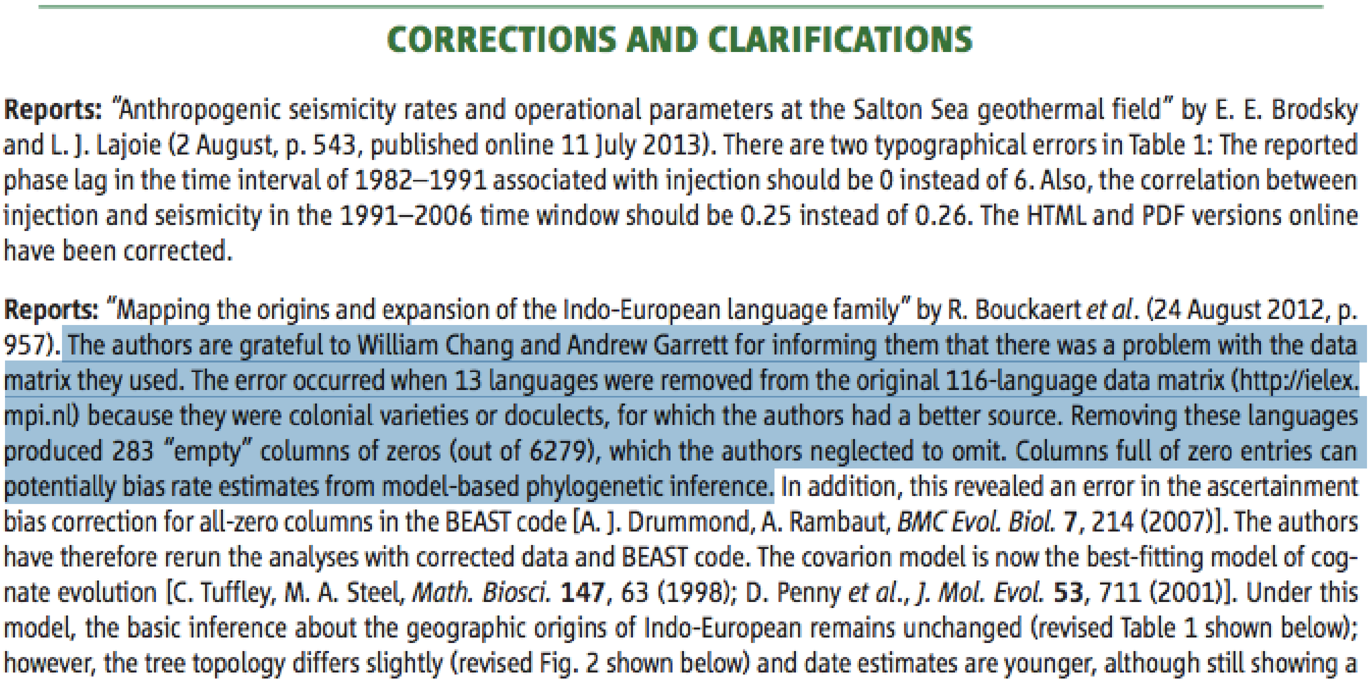
If you see very few cognates on the left side of the distribution, then you've got few cognates between sister languages and small groups. This will be a much bigger problem – you may be able to recover the deeper groupings but your younger relationships will be flakey.

Note that there are formal tests of signal in your data which you should try out if you've got some unusual data. Two good old fashioned tests are the `g1` (Hillis & Huelsenbeck '92) and Permutation Tail Probability tests (`PTP` , Archie '89). You should also read Revell et al. ('08) as phylogenetic signal is quite a complex problem. As far as I know these tests are only implemented in *PAUP** ("pop-star"). Ask me about this if you are interested.

(For the record, this dataset does have significant phylogenetic signal.)

Do we have any *empty* cognate sets?

If we have any sites that are completely empty, these can be problems e.g. in a recent paper by Bouckaert et al. '13 dating the spread of Indo-European, we had to issue a correction when some empty sites were left in:



CORRECTIONS AND CLARIFICATIONS

Reports: "Anthropogenic seismicity rates and operational parameters at the Salton Sea geothermal field" by E. E. Brodsky and L. J. Lajoie (2 August, p. 543, published online 11 July 2013). There are two typographical errors in Table 1: The reported phase lag in the time interval of 1982–1991 associated with injection should be 0 instead of 6. Also, the correlation between injection and seismicity in the 1991–2006 time window should be 0.25 instead of 0.26. The HTML and PDF versions online have been corrected.

Reports: "Mapping the origins and expansion of the Indo-European language family" by R. Bouckaert *et al*. (24 August 2012, p. 957). The authors are grateful to William Chang and Andrew Garrett for informing them that there was a problem with the data matrix they used. The error occurred when 13 languages were removed from the original 116-language data matrix (http://ielex.mpi.nl) because they were colonial varieties or doculects, for which the authors had a better source. Removing these languages produced 283 "empty" columns of zeros (out of 6279), which the authors neglected to omit. Columns full of zero entries can potentially bias rate estimates from model-based phylogenetic inference. In addition, this revealed an error in the ascertainment bias correction for all-zero columns in the BEAST code [A. J. Drummond, A. Rambaut, *BMC Evol. Biol.* **7**, 214 (2007)]. The authors have therefore rerun the analyses with corrected data and BEAST code. The covarion model is now the best-fitting model of cognate evolution [C. Tuffley, M. A. Steel, *Math. Biosci.* **147**, 63 (1998); D. Penny *et al.*, *J. Mol. Evol.* **53**, 711 (2001)]. Under this model, the basic inference about the geographic origins of Indo-European remains unchanged (revised Table 1 shown below); however, the tree topology differs slightly (revised Fig. 2 shown below) and date estimates are younger, although still showing a

By why is this a problem? It means the analysis has proportion of data that are absent, are always absent, and don't ever change so it messes with the rates.

Note the magnitude of this – 4.5% of the data were absent. This lead to about a 10% over-estimation in the age of the Indo-European family.

This was embarrassing. So, make sure you don't have lots of empty cognates:

```
sizes[sizes==0]
```

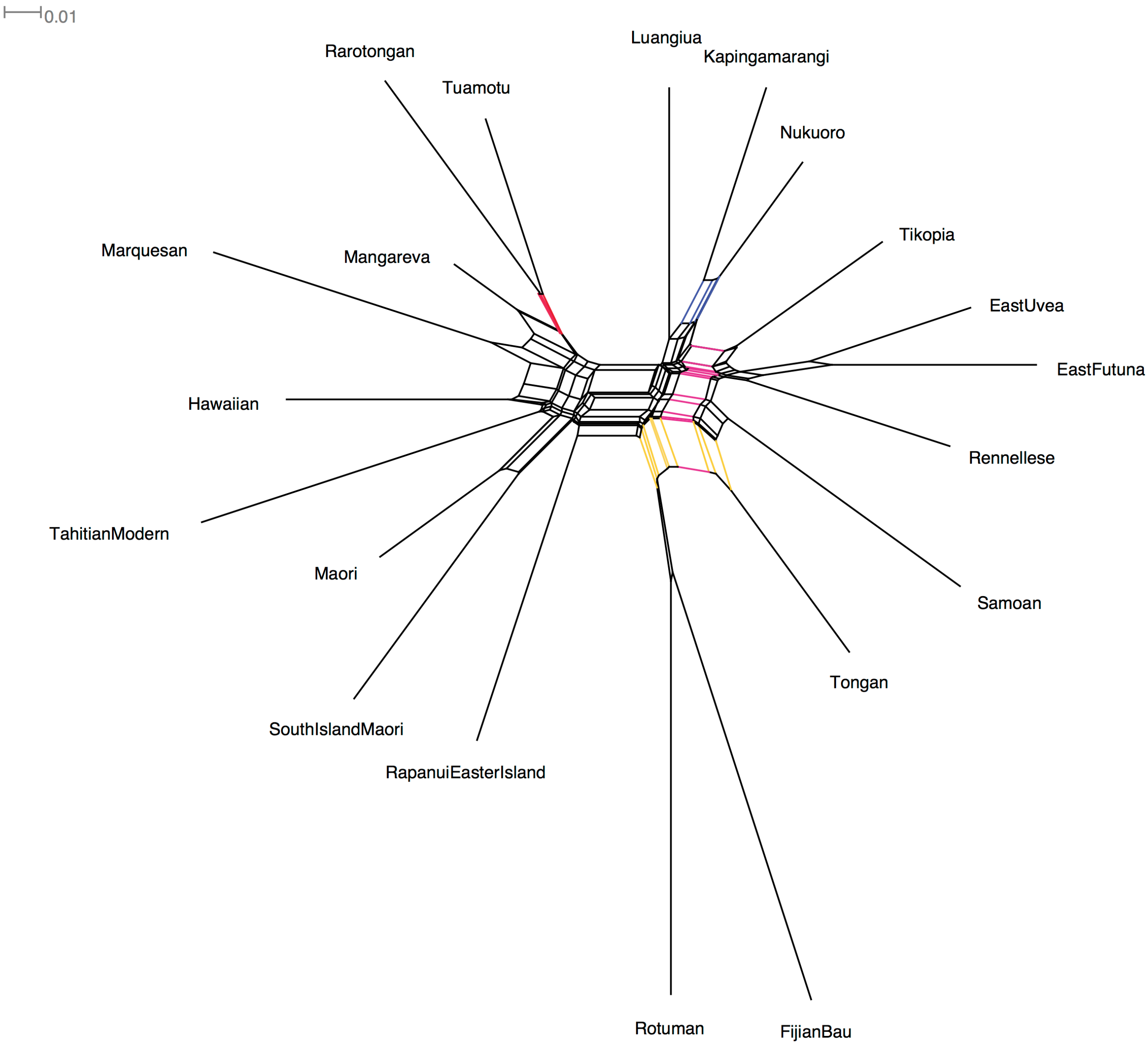
.. if your data has some then you will need to remove them.

Our dataset will have ONE. We need this because of the way `BEAST` implements a correction for `Ascertainment Bias`. See the section on `Ascertainment Correction` later on.

Networks:

Another good way to visualise the data is a `Neighbor-Net` network. This shows the conflicting signal in the data using a `splits graph`. Here, bigger boxes mean more conflict, longer lines means more signal in the data supporting that 'split'.

Creating a `Neighbor-Net` is a good topic for an “Hour of Power”. Here’s the Network of our data:



How to read these things is to look for the parallel lines as these represent the groupings in our data. I’ve colored a few of them to show you the groups.

1. A nice ‘tree-like’ split with little conflict looks like the tiny box grouping. A good example of this is the *red* split that groups Rarotongan and Tuamotu with each other against everything else.

2. A slightly more conflicting group is the *blue* split that groups Kapingamarangi and Nukuoro.

3. There’s a really big conflicting split for Tongan. You can see that the *yellow* splits place it with Rotuman and Fijian, while the *pink* split places Tongan with the Samoic-Outlier languages (Tikopia, East Uvea, East Futuna, Rennellese, Samoan). At a guess I’d expect this to be an outcome of Tongan influencing the Samoic-Outlier languages during the Tongan Empire period

https://en.wikipedia.org/wiki/Tu%CA%BBi_Tonga_Empire.

Have a look at the tree we drew above and compare. Do we see any other groupings from the tree in the network?

```
[ ] Are there any groupings we can see in both the tree and the network?
```

Overall this is quite a messy dataset, so it should be fun to see what we can do with trees…

Other methods:

There is ongoing development into tools for assessing data quality. Remco Bouckaert's working on an approach to visualise cognate data across a map (implemented in `Babel` . This figure below shows the cognates for "we" in these languages and what they group. Interestingly there's a cognate linking NZ Maori to Kapingamarangi. This is a little surprising, and suggests we need to look at that data again.