Our contributions to the raket/PBD packages
Supervisor: Bilderbeek, R.
Students: Damhuis, J. and Dekker, T.

What is Raket?

Raket Is an R package that uses

- PBD: to generate incipient species trees and sample a species tree
- pirouette: to convert a species phylogeny to a posterior
- nLTT: to compare a species tree to all species trees in the posterior

Steps to the pipeline

1. Create parameter files
2. Use PBD_sim to simulate a PBD tree
3. Sample a good species tree from incipient species tree
4. Generate a DNA alignment from the good species tree
5. Simulate BD tree from the DNA alignment -> posterior
6. Compare both simulated trees using nLTT statistics
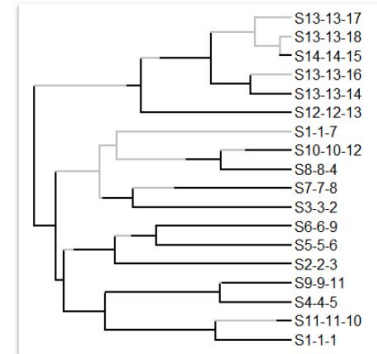7. Visualize data with violin plots

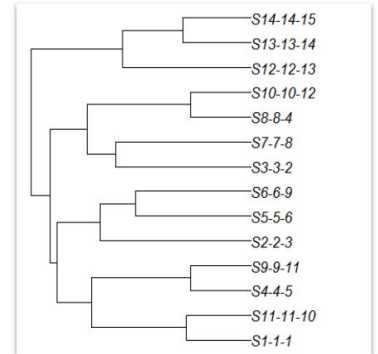## Where we worked in the pipeline

Step 1,2 and 3
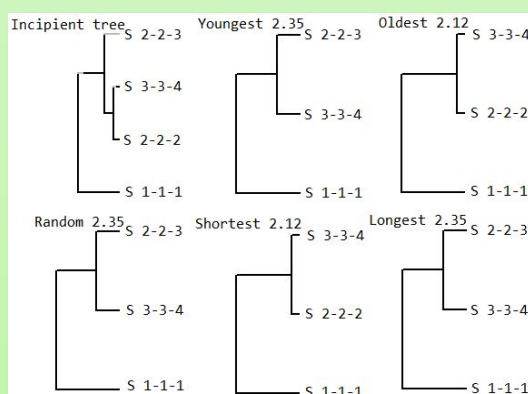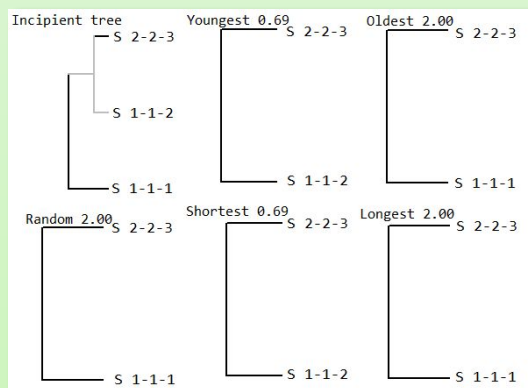

Two new ways to sample a good species tree

Added sampling methods longest and shortest


```
if (samplemethod == "shortest") {
  M <- matrix(
    c(rep(-1e10, lenL)),
    nrow = lenL,
    ncol = 1
  )
  L <- cbind(L, M)
  for (i in 2:lenL) {
    if (L[L[i, 2], 6] != L[i, 6]) {
      if (L[L[i, 2], 7] < L[i, 3]) {
        L[L[i, 2], 7] <- L[i, 3]
      }
      if (L[i, 7] < L[i, 3]) {
        L[i, 7] <- L[i, 3]
      }
    }
  }
  for (j in lenL:2) {
    if (L[i, 7] != -1e10) {
      if (L[L[i, 2], 7] < L[i, 3]) {
        L[L[i, 2], 7] <- L[i, 3]
      }
    }
  }
  for (j in 2:lenL) {
    if (L[i, 7] == -1e10) {
      if (L[L[i, 2], 7] != -1e10) {
        if (L[L[i, 2], 7] >= L[i, 3]) {
          L[i, 7] <- L[i, 3]
        } else {
          L[i, 7] <- L[L[i, 2], 7]
        }
      }
    }
  }
  neworder <- order(L[, 7], decreasing = TRUE)
  L <- L[, -7]
}
```

Getting the expected number of good and incipient lineages from a protracted pure birth model

$$A = \sqrt{\left(1 - \frac{\lambda_3}{\lambda_2}\right)^2 + 4\frac{\lambda_1}{\lambda_2}}$$

$$\mathbb{E}^g[N_g; t] = \frac{N_g(0)}{2A}\left(\left(A + 1 - \frac{\lambda_3}{\lambda_2}\right)e^{\frac{1}{2}(\lambda_3 + (A-1)\lambda_2)t} + \left(A - 1 + \frac{\lambda_3}{\lambda_2}\right)e^{\left(\frac{1}{2}(\lambda_3 - (A+1)\lambda_2)t\right)}\right)$$

$$\mathbb{E}^g[N_i; t] = \frac{N_g(0)}{A}\frac{\lambda_1}{\lambda_2}\left(e^{\left(\frac{1}{2}(\lambda_3 + (A-1)\lambda_2)t\right)} - e^{\frac{1}{2}(\lambda_3 - (A+1)\lambda_2)t}\right)$$

$$\mathbb{E}^g[N_g; t] = \frac{N_g(0)}{1 + \frac{\lambda_1}{\lambda_2}}e^{\lambda_1 t} + \frac{N_g(0)}{1 + \frac{\lambda_2}{\lambda_1}}e^{-\lambda_2 t}$$

$$\mathbb{E}^g[N_i; t] = \frac{N_g(0)}{1 + \frac{\lambda_2}{\lambda_1}}\left(e^{\lambda_1 t} - e^{-\lambda_2 t}\right)$$

```
if (ng > 0.0) {
  temp_a1 <- 1 - (siri / scr)
  temp_a2 <- temp_a1 ^ 2 + (4 * (sirg / scr))
  temp_a <- sqrt(temp_a2)
  temp_b <- temp_a + 1 - (siri / scr)
  temp_c <- exp(0.5 * (siri + (temp_a - 1) * scr) * age)
  temp_d <- exp(0.5 * (siri - (temp_a + 1) * scr) * age)
  exp_ng <- (ng / (2 * temp_a)) * (temp_b * temp_c + (temp_a - 1 +
    (siri / scr)) * temp_d)
  exp_ni <- (ng / temp_a) * (sirg / scr) * (temp_c - temp_d)
} else {
  exp_ng <- (ni / (1 + (sirg / scr))) * (exp(sirg * age) - exp(- scr * age))
  exp_ni <- (ni / (1 + (scr / sirg))) * exp(sirg * age) +
    (ni / (1 + (sirg / scr))) * exp(- scr * age)
}
```

```
test_that("pbd_get_exp_n_lineages gives right number of lineages", {

  # Excel calculation for sirg=2,scr=0.5,siri=2,age=2,ng=1,ni=0 yields:
  # Expected number of good lineages: 11.21393
  # Expected number of inicipien lineages: 43.38422
  exp_out <- list(exp_ng = 11.21393, exp_ni = 43.38422)

  # Now using pbd_get_exp_n_lineages_pb yields:
  out <- pbd_get_exp_n_lineages_pb(
    sirg = 2,
    scr = 0.5,
    siri = 2,
    age = 2,
    ng = 1,
    ni = 0
  )
```

Results from the peregrine cluster

Coming soon....

rijksuniversiteit groningen