infectionAgeHIV Workflow

Erik Lundgren

July 2023

Overview

infectionAgeHIV is a small package that can produce probability distributions for infection times based on HIV biomarkers

Workflow

- format patient data into two data frames: one for data where there is one value per patient and one where there may be multiple
- use `find.infection.ages` to determine which biomarker values are usable in the multiple biomarker model and find infection age distributions

Load package and example data set pRNG seed

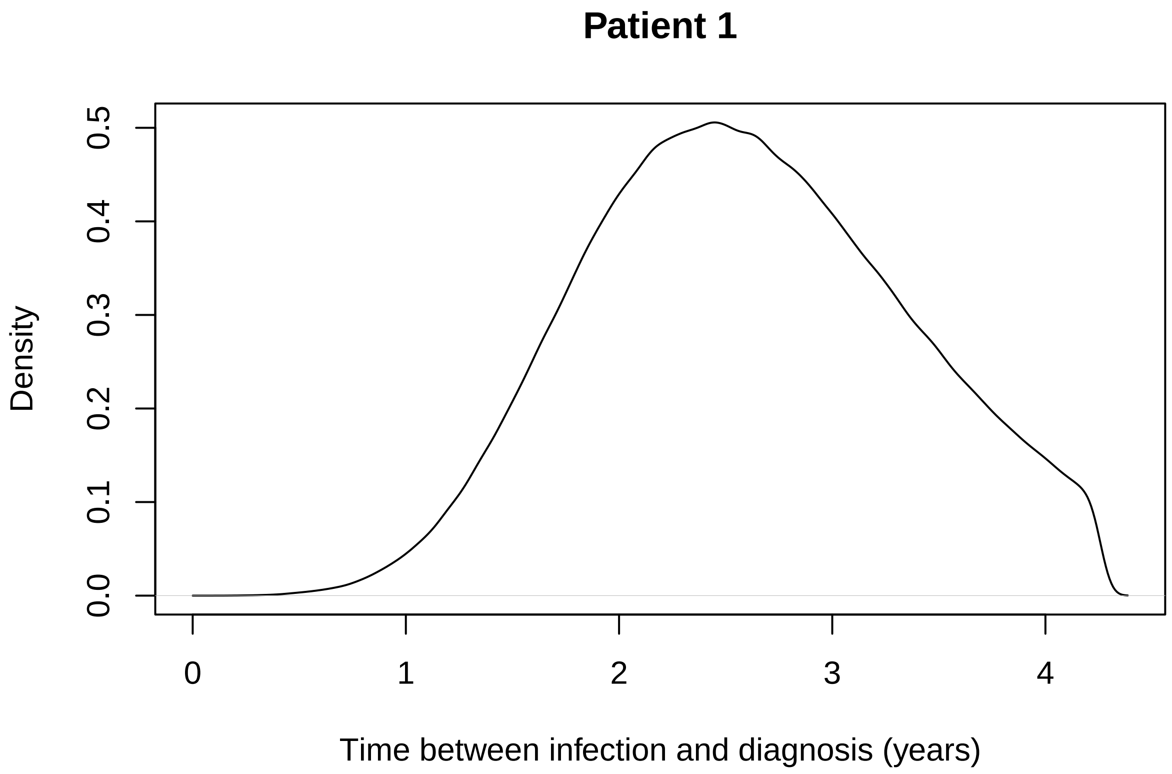
```
library(infectionAgeHIV)
df.single <- df.single.example
df.multiple <- df.multiple.example
set.seed(12345)
```

Using biomarker data to predict infection times

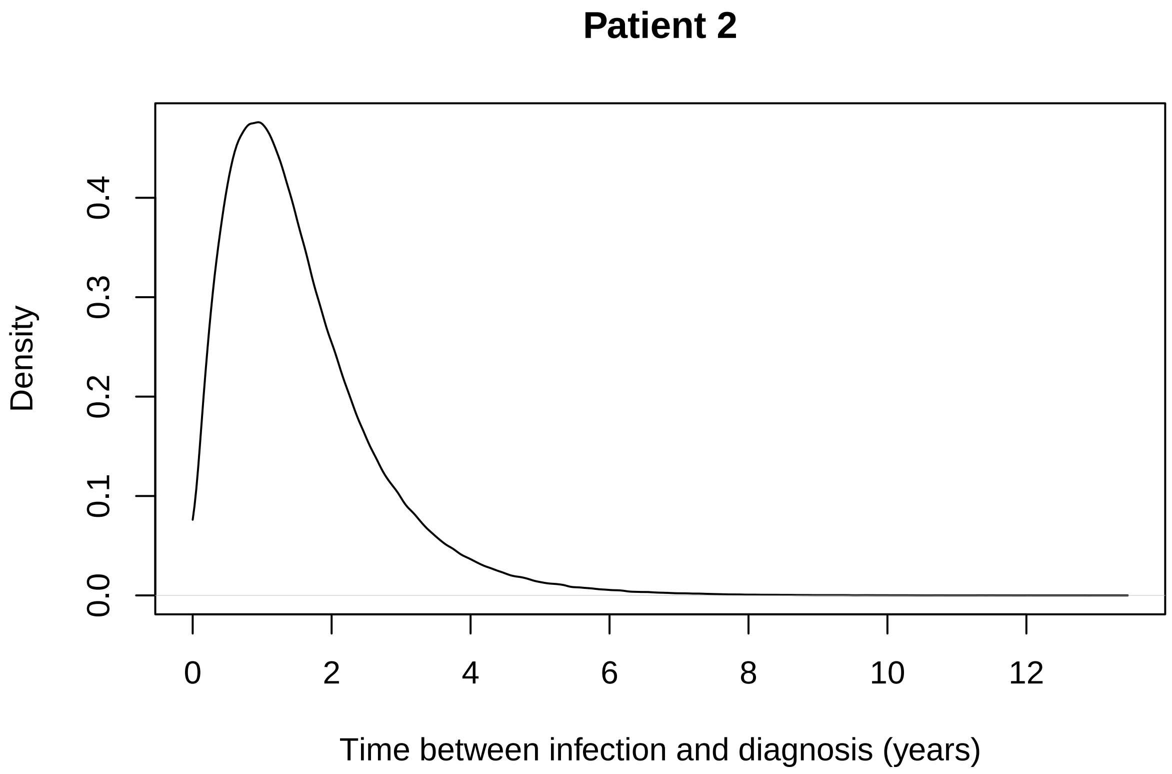
```
df <- find.infection.ages(df.single = df.single,
                          df.multiple = df.multiple,
                          n.adapt = 1e4, n.burn = 1e5, n.iter = 1e6)
```

We can plot the inferred distributions for the amount of time between infection and diagnosis.

```
plot(df$infection_age_dists_diag[[1]],
     xlab = "Time between infection and diagnosis (years)",
     main = paste0("Patient ", df.single$patient_id[[1]]))
```

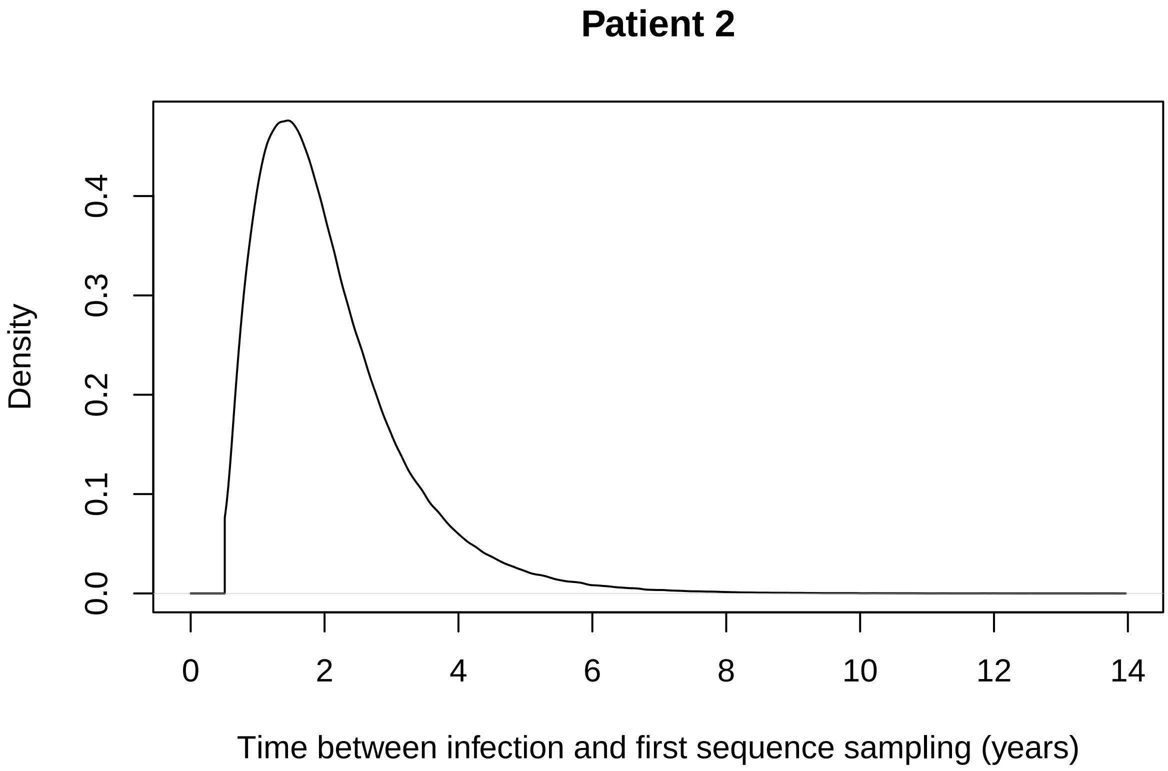


```r
plot(df$infection_age_dists_diag[[2]],
     xlab = "Time between infection and diagnosis (years)",
     main = paste0("Patient ", df.single$patient_id[[2]]))
```

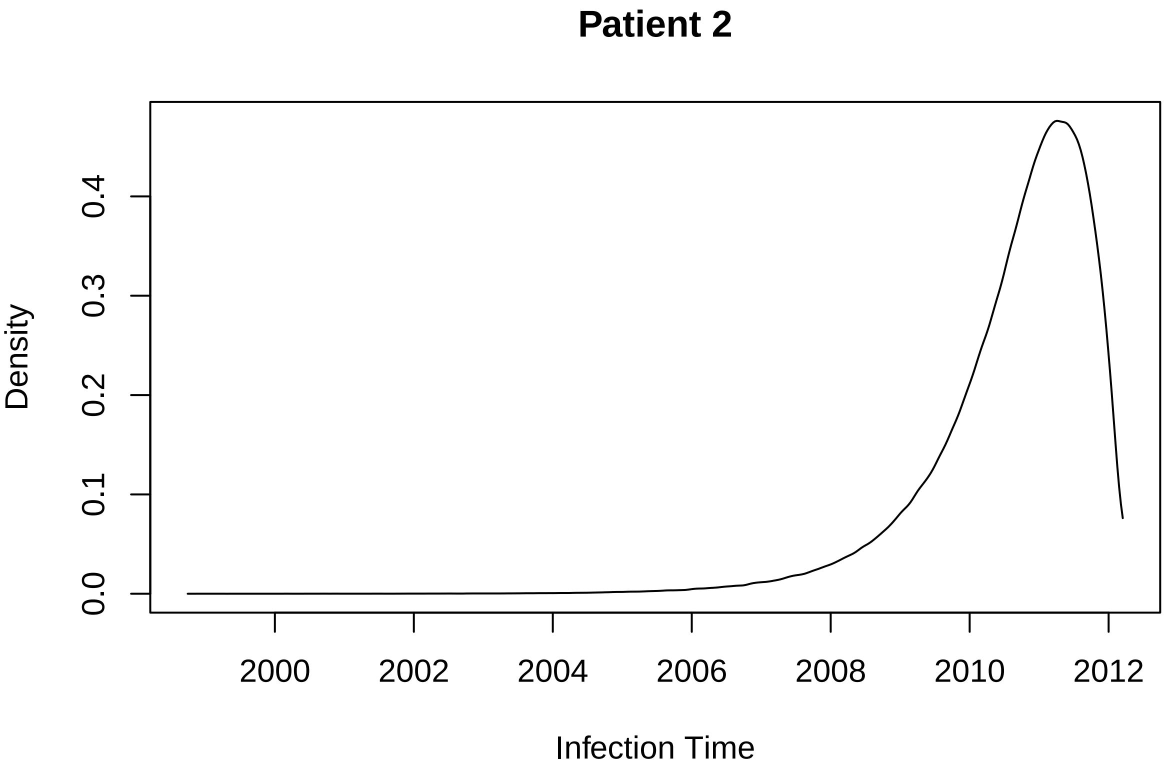


The distributions for the amount of time between infection and taking sequence samples and the infection time density in terms of real time are also returned.

```
plot(df$infection_age_dists_seq[[2]],
     xlab = "Time between infection and first sequence sampling (years)",
     main = paste0("Patient ", df.single$patient_id[[2]]))
```



```
plot(df$pdf[[2]], type = "l",
     xlab = "Infection Time",
     ylab = "Density",
     main = paste0("Patient ", df.single$patient_id[[2]]))
```



Drawing random infection ages

We can also draw from the inferred distributions by supplying a uniform [0,1] random variable to the inverse CDF transform.

```
drawn.infection.ages <- df$icdf[[2]](runif(10))
print(drawn.infection.ages)
```

```
##  [1] 3.6748076 2.6203018 2.1758194 0.8924798 1.3945458 0.6968928 4.2245902
##  [8] 0.6775484 1.3361903 0.6565581
```

Using `prepare.HIV.data`

If more control is needed, `prepare.HIV.data` can be called without the wrapper `find.infection.ages`