Comparing parameters using `retrocombinator`

Anindya Sharma

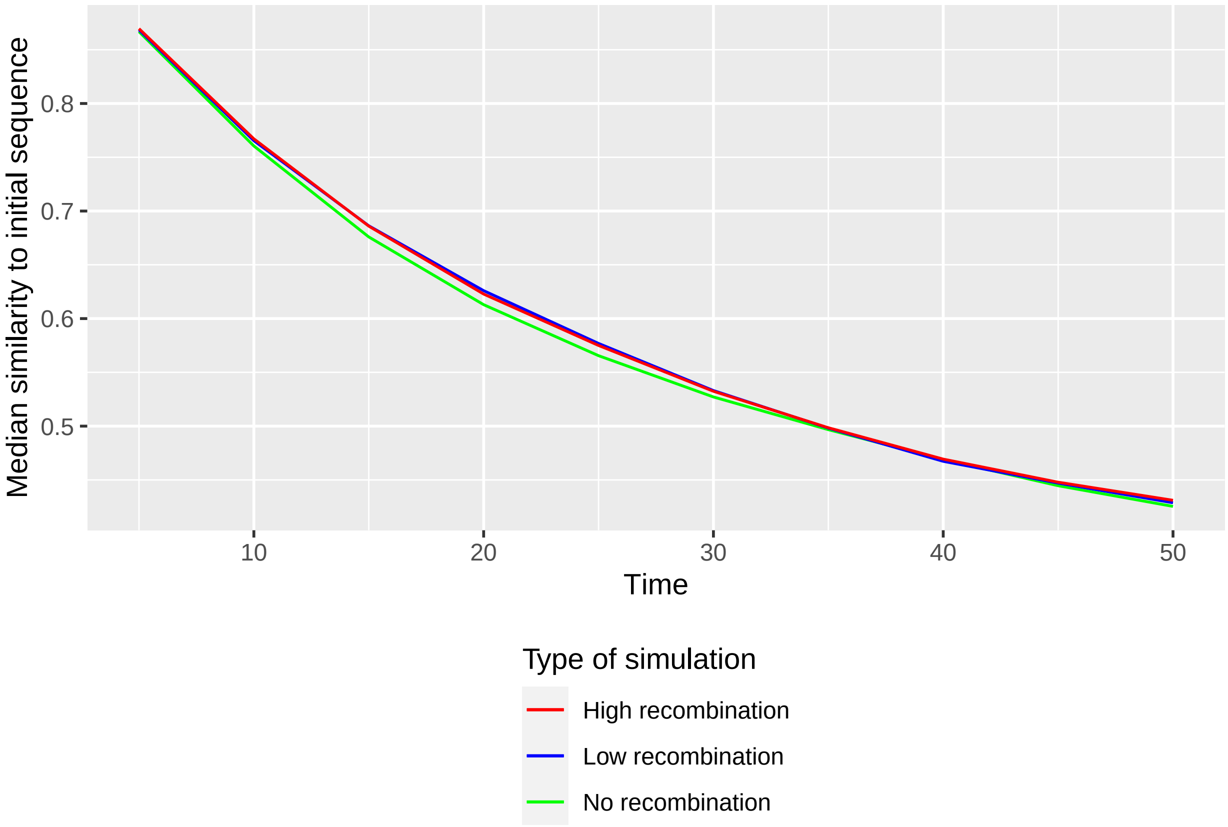
2022-04-01

```
library(ggplot2)
library(dplyr)
#>
#> Attaching package: 'dplyr'
#> The following objects are masked from 'package:stats':
#>
#>     filter, lag
#> The following objects are masked from 'package:base':
#>
#>     intersect, setdiff, setequal, union
library(magrittr)
library(retrocombinator)
```

```
sequenceParams <- SequenceParams(numInitialCopies = 20,
                                 sequenceLength = 5000
)
activityParams <- ActivityParams(lengthCriticalRegion = 50,
                                 probInactiveWhenMutated = 0.01)
burstParams <- BurstParams(burstProbability = 0.2, burstMean = 5,
                           maxTotalCopies = 100)
selectionParams <- SelectionParams(selectionThreshold = 0.3)
familyParams <- FamilyParams(familyCoherence = 0.6)
simulationParams <- SimulationParams(numSteps = 100, timePerStep = 0.5)

simulateEvolution(sequenceParams = sequenceParams,
                  activityParams = activityParams,
                  burstParams = burstParams,
                  simulationParams = simulationParams,
                  familyParams = familyParams,
                  selectionParams = selectionParams,
                  recombParams = RecombParams(recombMean = 0,
                                              recombSimilarity = 0.8),
                  outputParams = OutputParams('simulationRecomb_no.out')
)
#> [1] "simulationRecomb_no.out"
simulateEvolution(sequenceParams = sequenceParams,
                  activityParams = activityParams,
                  burstParams = burstParams,
                  simulationParams = simulationParams,
                  familyParams = familyParams,
                  selectionParams = selectionParams,
                  recombParams = RecombParams(recombMean = 2,
                                              recombSimilarity = 0.8
                                              ),
                  outputParams = OutputParams('simulationRecomb_lo.out')
)
#> [1] "simulationRecomb_lo.out"
simulateEvolution(sequenceParams = sequenceParams,
                  activityParams = activityParams,
                  burstParams = burstParams,
                  simulationParams = simulationParams,
                  familyParams = familyParams,
                  selectionParams = selectionParams,
                  recombParams = RecombParams(recombMean = 10,
                                              recombSimilarity = 0.8
                                              ),
                  outputParams = OutputParams('simulationRecomb_hi.out')
)
#> [1] "simulationRecomb_hi.out"

data_no <- parseSimulationOutput('simulationRecomb_no.out')
data_lo <- parseSimulationOutput('simulationRecomb_lo.out')
data_hi <- parseSimulationOutput('simulationRecomb_hi.out')
```

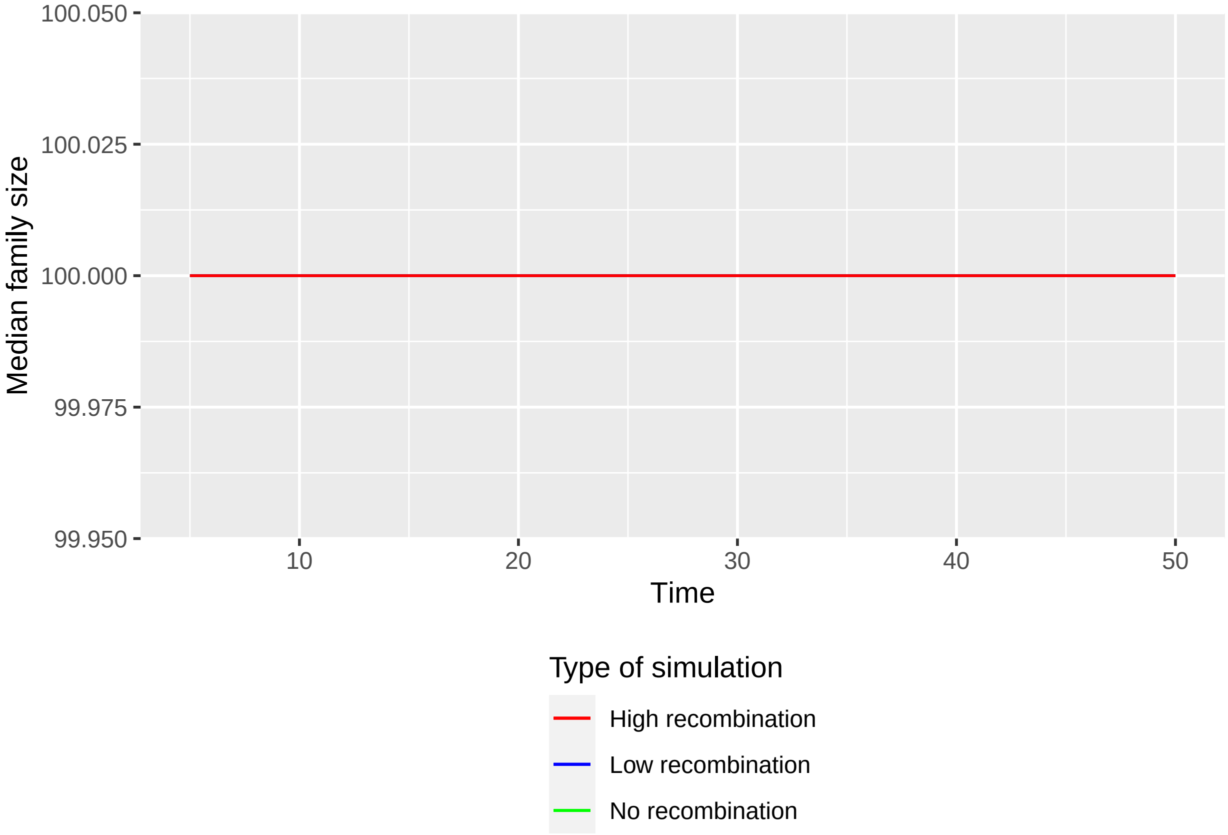
Distance to initial sequence

```
summary_no <- summariseEvolution(data_no, 'initial')
summary_lo <- summariseEvolution(data_lo, 'initial')
summary_hi <- summariseEvolution(data_hi, 'initial')

ggplot() +
  geom_line(data=summary_no, aes(x=realTime, y = divergenceMean,
                color = "No recombination")) +
  geom_line(data=summary_lo, aes(x=realTime, y = divergenceMean,
                color = "Low recombination")) +
  geom_line(data=summary_hi, aes(x=realTime, y = divergenceMean,
                color = "High recombination")) +
  scale_color_manual(
    name = "Type of simulation",
    values = c("No recombination" = "green",
               "Low recombination" = "blue",
               "High recombination" = "red"
    )
  ) +
  labs(x = 'Time', y = 'Median similarity to initial sequence') +
  retrocombinatorTheme()
```



Pairwise distances

```
summary_no <- summariseEvolution(data_no, 'families')
summary_lo <- summariseEvolution(data_lo, 'families')
summary_hi <- summariseEvolution(data_hi, 'families')

ggplot() +
  geom_line(data=summary_no, aes(x=realTime, y = familiesMedian,
                color = "No recombination")) +
  geom_line(data=summary_lo, aes(x=realTime, y = familiesMedian,
                color = "Low recombination")) +
  geom_line(data=summary_hi, aes(x=realTime, y = familiesMedian,
                color = "High recombination")) +
  scale_color_manual(
    name = "Type of simulation",
    values = c("No recombination" = "green",
               "Low recombination" = "blue",
               "High recombination" = "red"
    )
  ) +
  labs(x = 'Time', y = 'Median family size') +
  retrocombinatorTheme()
```


100.050
100.025
100.000
99.975
99.950
Median family size
10
20
30
40
50
Time
Type of simulation
High recombination
Low recombination
No recombination