Benchmarking makeblastdb

Armand

22 January 2019

Benchmark time analysis for makeblastdb

The python script included in this repo was used to sample pseudo random sequences from the Los Alamos sequences as described in our paper, Table 2.

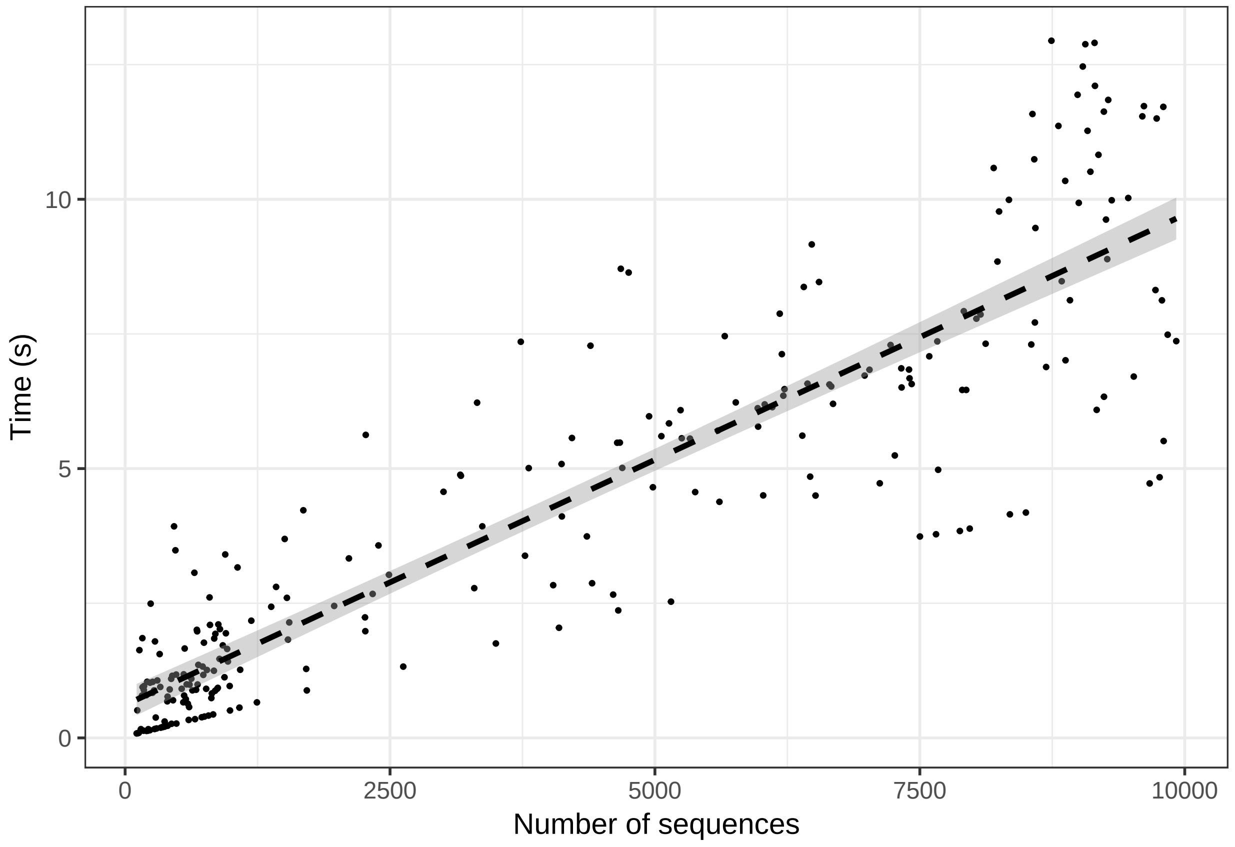
The code below plots the results and fits a linear model where the number of sequences is the independent variable and the time it takes is the dependent variable.

```
library(ggplot2)


#system("scp pi@196.254.115.64:/home/pi/blastDataBaseTiming/makeBlastTimes.txt .")


timeFile <- read.csv("makeBlastTimes.txt", header = F)

colnames(timeFile) <- c("n", "s")


lin <- lm(s ~ n, data = timeFile)

summary(lin)
```

```
##
## Call:
## lm(formula = s ~ n, data = timeFile)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.6897 -0.7049 -0.1006  0.7336  4.3736
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 6.117e-01  1.498e-01   4.082 6.02e-05 ***
## n           9.104e-04  2.858e-05  31.856  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.565 on 248 degrees of freedom
## Multiple R-squared:  0.8036, Adjusted R-squared:  0.8028
## F-statistic:  1015 on 1 and 248 DF,  p-value: < 2.2e-16
```

```
p <- ggplot(timeFile, aes(x = n, y = s))+
  geom_point(size = 0.5)+
  geom_smooth(method = 'lm', color = "black", linetype = "dashed")+
  theme_bw()+
  xlab("Number of sequences")+
  ylab("Time (s)")
```

```r
p
```



```r
ggsave(filename = "makeblastdb.tiff", plot = p, dpi = 300)
```

```
## Saving 6.5 x 4.5 in image
```

From the linear model summary shown above, we can see that it takes approximately 0.0009 seconds for every sequence.