[Re] Local alignment statistics - Estimating K

Nathan Brouwer

11/18/2019

Estimating K

Key to E values and P values for alignments are lambda and K. From the extreme value distribution of scores we can get lambda and mu and need to calculate K.

Plain text version of equation:

mu = (ln(m x n x K))/lambda

Rendered version of equation.

$$\mu = \frac{\ln(m * n * K)}{\lambda}$$

We can re-arrange this to be

$$\lambda * \mu = \ln(m * n * K)$$

And then

$$\exp(\lambda * \mu) = \exp(\ln(m * n * K))$$

Which simplifies to

$$\exp(\lambda * \mu) = m * n * K$$

and then

$$\frac{exp(\lambda * \mu)}{m * n} = K$$

For ease let's flip that around

$$K = \frac{exp(\lambda * \mu)}{m * n}$$

We could also write it like this

$$K = \frac{e^{\lambda * \mu}}{m * n}$$

In R we can therefore get K like this (recall that m = n = 191, the length of the simulated sequences)

```
m <- 191
n <- 191

mu.param  <- 26.18856
scale.param <- 3.296625
lambda.parm  <- 1/scale.param
```

```
K <- (exp(lambda.parm*mu.param))/(m*n)
K
```

```
## [1] 0.07726649
```

An equivalent expression is worked out below, though I haven't derived it:

```
exp(lambda.parm*mu.param - log(m*n))
```

```
## [1] 0.07726649
```

Summary

Once we've simulated data we can make a histogram of all of the alignment scores. We can then use R functions to analyze the distribution of scores and estimate mu and lambda. From mu and lambda we can calculate K. In the first row of Table 1 Altchul and Gish (1996) report, for 10000 simulated sequences of n = m = 191:

- mu = 26.45
- lambda = 0.298
- K = 0.073

Using R we got

```
mu.param
```

```
## [1] 26.18856
```

```
lambda.parm
```

```
## [1] 0.3033405
```

```
K
```

```
## [1] 0.07726649
```

These values are close. The most likely reason for them not being closer is that we ran fewer simulations as as the original paper, though there could be subtle numerical differences in parts of our procedure that are different than Altchul and Gish.