

Optimize dCNS parameter for brassicaceae species

Baoxing Song

2020-08-28

notice: those parameter have not been well tested yet.

firstly, I tried to optimize the parameters

input file:

tair10.fa is the arabidopsis genome

chi_v1.fa is the *Cardamine Hirsuta* genome downloaded from <http://chi.mpiiz.mpg.de/assembly.html>

Brapa_sequence_v3.0.fasta is the *Brassica rapa* genome download from http://brassicadb.org/brad/datasets/pub/Genomes/Brassica_rapa/V3.0/

run this command to generate sequence alignment score from random extracted genome segments.

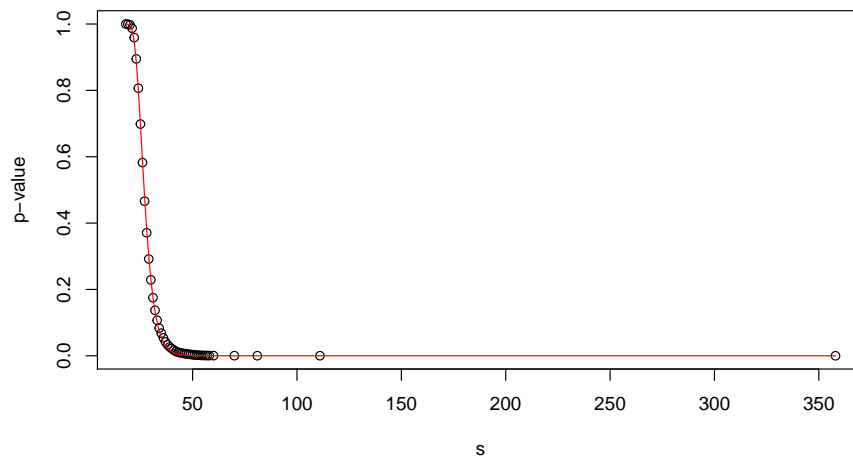
```
dCNS ranSco -r tair10.fa -i chi_v1.fa > r_arabidopsisthaliana_i_CardamineHirsuta_ranSco
dCNS ranSco -r tair10.fa -i Brapa_sequence_v3.0.fasta > r_arabidopsisthaliana_i_Brapa_ranSco
```

Run the non-linear regression R code to validate the sequence alignment scoring parameters and calculate k and lambda values.

```
data1 = read.table("r_arabidopsisthaliana_i_CardamineHirsuta_ranSco")
u = unique(data1$V1)
s = u[order(u)]

data = data.frame(x=s, y=s)
for ( i in 1:length(s) ){
  data[i, 2] = length(which(data1$V1 >= data$x[i])) / nrow(data1)
}
x=data$x
y=data$y
library(minpack.lm)
nonlin_mod=nlsLM(y~(1-exp(-1*k*1000000*exp(-1*l*x))),
  control=nls.lm.control(maxiter=550), start=list(k=0.3, l=0.2))
plot(x, y, xlab="s", ylab="p-value")
lines(x,predict(nonlin_mod),col="red")
```

Optimize dCNS parameter for brassicaceae species



```
nonlin_mod
## Nonlinear regression model
## model: y ~ (1 - exp(-1 * k * 1e+06 * exp(-1 * l * x)))
## data: parent.frame()
##      k      l
## 0.002314 0.302853
## residual sum-of-squares: 0.001673
##
## Number of iterations to convergence: 21
## Achieved convergence tolerance: 1.49e-08
```

The non-linear regression convergent well. The estimated k value is 0.002314, and the estimated lambda value is 0.302853.

Check where is the significant score threshold. We would unlikely observe the identical k and lambda values using different datasets, while those slightly different k and lambda values could produce same sequence alignment significant threshold.

```
kValue = 0.002314
lambda = 0.302853
eValue = kValue * 100000 * 100000 * exp(-1.0 * lambda * 63)
pvalue = 1.0 - exp(-1.0*eValue)
pvalue
## [1] 0.1128242

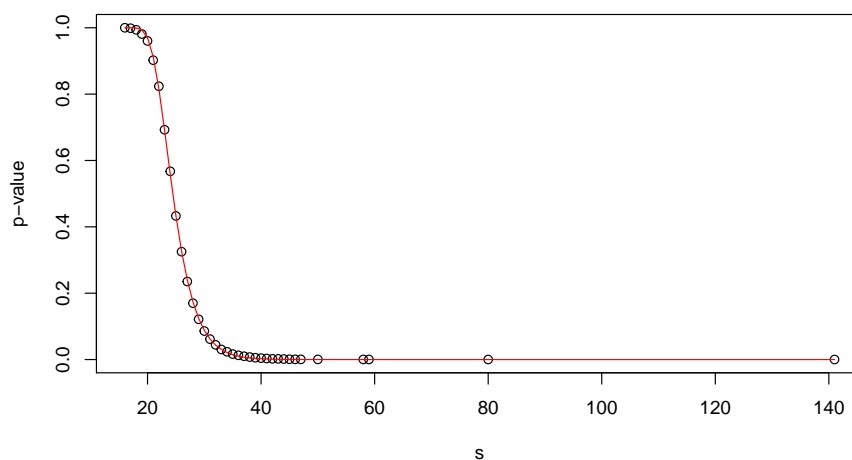
eValue = kValue * 100000 * 100000 * exp(-1.0 * lambda * 64)
pvalue = 1.0 - exp(-1.0*eValue)
pvalue
## [1] 0.08463486
```

for 100kb * 100kb sequence alignment, score 63-64 gives a p-value around 0.1 (in this implementation sequence alignment scores are always integers)

Optimize dCNS parameter for brassicaceae species

```
data1 = read.table("r_arabidopsisthaliana_i_Brapa_ranSco")
u = unique(data1$V1)
s = u[order(u)]

data = data.frame(x=s, y=s)
for ( i in 1:length(s) ){
  data[i, 2] = length(which(data1$V1 >= data$x[i])) / nrow(data1)
}
x=data$x
y=data$y
library(minpack.lm)
nonlin_mod=nlsLM(y~(1-exp(-1*k*1000000*exp(-1*l*x))),
                  control=nls.lm.control(maxiter=550), start=list(k=0.3, l=0.2))
plot(x, y, xlab="s", ylab="p-value")
lines(x,predict(nonlin_mod),col="red")
```



```
nonlin_mod
## Nonlinear regression model
## model: y ~ (1 - exp(-1 * k * 1e+06 * exp(-1 * l * x)))
## data: parent.frame()
##      k      l
## 0.005047 0.363676
## residual sum-of-squares: 0.0006118
##
## Number of iterations to convergence: 20
## Achieved convergence tolerance: 1.49e-08
```

The non-linear regression convergent well. The estimated k value is 0.005047, and the estimated lambda value is 0.363676.

```
kValue = 0.002314
lambda = 0.302853
eValue = kValue * 100000 * 100000 * exp(-1.0 * lambda * 63)
```

Optimize dCNS parameter for brassicaceae species

```
pvalue = 1.0 - exp(-1.0*eValue)
pvalue
## [1] 0.1128242

eValue = kValue * 100000 * 100000 * exp(-1.0 * lambda * 64)
pvalue = 1.0 - exp(-1.0*eValue)
pvalue
## [1] 0.08463486
```

for 100kb * 100kb sequence alignment, score 63-64 gives a p-value around 0.1, this identical with the *Cardamine Hirsuta*

Those two genome with different genome size and different ploid level gave same score threshold.

we used a smith-waterman score 60 (<64, 64 is the significant threshold) as a minimum score of seed. The minimum seed size is 30 (60/2, 60 is the minimum seed score, 2 is the match score.)

The seed window size 55 was selected to make sure there is only one seed in each window. since $55 > 30$ and < 60

The step_size value 10 to make sure there is no seed missing for each window sliding. Since $55 - 10 = 45 > 30$

For genome masking, we used 25-mer (<30 which is the minimum seed size).