Statistics for Bioinformatics
Practicals - Word Count Probabilities

We will use **R** to answer a series of questions on word counts. These analyses will allow us to put in practice several topics treated in the theoretical course. We will follow a problem-driven approach starting from a question, we will see how to solve it with a **R** script.

We will start with a tutorial, where the solutions will be shown below each question. A few additional exercises are also proposed.

In the current tutorial, we will analyze the probabilities associated to word occurrences. In the next tutorial, we will fit various theoretical distributions on an observe distribution of word counts, and compare the significance calculated with these differenc distributions.

These tutorials assume that you already learned the following chapters of the course.

1. Probabilities
2. Theoretical distributions
3. Fitting
4. Goodness of fit
5. Test of significance

Tutorial: occurrence probabilities

Assuming that a 1000 bp DNA sequence has been generated with a Bernouilli schema, and the following residue probabilities:

- P(A) = 0.32
- P(T) = 0.33
- P(C) = 0.17
- P(G) = 0.18

Questions

1. What is the probability to observe an occurrence of the word GATAAG at a given position of the sequence ?

2. What is the expected number of occurrences for the word GATAAG in a sequence of this length ?

3. What is the probability to observe exactly 3 occurrences of the word GATAAG ?

4. What is the probability to observe at least 3 occurrences of this word ?

5. Plot the theoretical distribution of probability of occurrences of this word.

6. Evaluate the effect of using the Poisson distribution as an approximation of the binomial in the tests above.

Solutions

The problem can be formulated as a Bernouilli schema, where each position of the 1000 sequence corresponds to a trial. At each position, the word can either be found (success) or not (failure). We assume that the probability of occurrence (success) is constant along the sequence, and we can thus use the binomial distribution to calculate the probability to observe a certain number of occurrences (successes).

We need to calculate

- The probability of success at each trial, i.e. the probability to observe the word at a given position of the sequence.
- The number of trials, i.e. the number of possible positions for this word in a 1000 bp sequences.

Probability of occurrence at a given position

The probability of the word can be estimated as the product of probability of its letters. Let us formulate this in **R**.

```
## Initializations
p.letter <- vector() ## Initialize a vector to store residue  probabilities
p.word <- vector() ## Initialize a vector to store word probabilities

W <- 'GATAAG' ## The word
k <- nchar(W) ## Word length

## Specify residue frequencies
p.letter["A"] <- 0.32
p.letter["C"] <- 0.17
p.letter["G"] <- 0.18
p.letter["T"] <- 0.33

## Check the result
print(p.letter)
sum(p.letter)

## calculate word probability
p.word[W] <- 1 # initialize
for (i in 1:nchar(W)) {
  letter <- substr(W,i,i) ## Select the letter at position i of word W
  p.word[W] <- p.word[W]*p.letter[letter] ## update word probability
}
```

Number of possible positions

```
## Calculate number of positions for a word of length k in a sequence of length L
L <- 1000
pos <- L - k + 1
print(pos)
```

Expected number of occurrences

The expected number of occurrences is the product of the number of possible positions by the probability of occurrence at a given position.

```
E.occ <- pos*p.word[W]
```

Occurrence probabilities

The probability of occurrences can be obtained with the binomial function. Before using it, you should read the on-line help and learn how to use the **R** implementation of this distribution.

```
## Open the help for the binomial distribution
help(Binomial)

## Note: on some versions of R, this does not work.
## If this is your case, try
help(dbinom)
```

The function `dbinom(x,size,prob)` calculates the density function, i.e. the probability to observe exactly *x* matches when one performs *size* trials with a probability *prob* of success at each trial. We can now fill the parameters to answer our first question.

```
## Probabilty to observe exactly 3 occurrences
dbinom(3,size=pos,prob=p.word[W])
```

In order to calculate the probability of observing at least 3 matches, we could sum the results obtained with the same density function, for each valu between 3 and 6. This would however be inefficient in terms if calculation.

The distribution function can return the same result in one operation, but we need to be ccareful about the parameters: by default, this function returns the lower tail of the distribution, i.e. the probability to observe **at most** *x* successes.

We can use the option *lower.tail=F* to specify that we want the right tail of the distribution. However, this will return the probability to observe **more than** *x* successes.

Thus, in order to calculate the probability to observe **at least** *x*, we need to ask the probability to observe **more than** *x-1* successes. In our case, x=3 and x-1=2.

One possibility for calculating the cumultative sum would be to calculate each individual probability, with the density function `dbinom`, and sum them. However, this would be very inefficient, and **R** provides a cumulative density function `pbinom` to calculate directly the left (default) or the right tail of a distribution.

```
## Probabilty to observe at least 3 occurrences
pbinom(2,size=pos,prob=p.word[W],lower.tail=F)
```

We will now plot the probability to observe exactly x occurrences, as a function of x. for this, we will come back to the density function, since we want the individual probability for each particular occurrence value.

```
x <- 0:pos ## x is a vector taking all possible positions
y <- dbinom(x,size=pos,prob=p.word[W])
plot(x,y,type="l",col="#0000BB",lwd=2)
```

This graphic is not very informative: we used the whole range of possible values (0:1000) for the X axis, but, given the small value of the word probability, the majority of the distribution is restricted to the smaller occurrence values (between 0 and 10). We wil thus plot the distribution over a shorter range.

For this, we will select the 11 first values of the occurrence values (`x[1:11]`), and the corresponding probability values, which are found in the 11 first entries of the y vector (`y[1:11]`).

```
## Plot frequency polygon
plot(x[1:11],y[1:11],type="l",col="#0000BB",lwd=2)

## Plot with histogram type and add a few legends
plot(x[1:10],y[1:10],
    type="h",
    col="#0000BB",
    lwd=2,
    main=paste("Probability of occurrences for word", W),
    xlab="occurrences",
    ylab="probability",
    panel.first=grid(col="#00BB00"))

## Do the same plot, with a logarithmic scale on Y axis.
plot(x[1:10],y[1:10],
    type="h",
    col="#0000BB",
    lwd=2,
    main=paste("Probability of occurrences for word", W),
    xlab="occurrences",
    ylab="probability",
```

```
    panel.first=grid(col="#00BB00"),
    log="y")
```

Poisson approximation

Evaluate the effect of using the Poisson distribution as an approximation of the binomial in the tests above.

For the Poisson distribution, we need only one parameter (the expected mean) instead of two (the number of trials and the probability of success at each trial). Actually, we already calculated the expected mean above : it is the expected number of occurrences for the considered word.

```
x <- 1:10

W <- 'GATAAG'
k <- 6
pos <- L-k+1

## Calculate the P-value of occurrences with a Poisson distribution
E.occ <- pos*p.word[W]
Pval.Poisson <- ppois(x-1, E.occ,lower=F)
Pval.binomial <- pbinom(x-1, pos, p.word[W], lower=F)

## Plot the histogram of the binomial P-value
plot(x[1:10],Pval.binomial,
    type="l",
    col="#0000BB",
    lwd=2,
    main=paste("Probability of occurrences for word", W),
    xlab="occurrences",
    ylab="probability",
    panel.first=grid(col="#00BB00"))

## Superimpose the Poisson approximation
lines(x[1:10],Pval.Poisson[1:10],col="red",lwd=1)


## Plot the histogram of the binomial P-value
plot(x[1:10],Pval.binomial,
    type="l",
    col="#0000BB",
    lwd=2,
    main=paste("Probability of occurrences for word", W),
    xlab="occurrences",
    ylab="probability",
    panel.first=grid(col="#00BB00"),
    log="y")

## Superimpose the Poisson approximation
lines(x[1:10],Pval.Poisson[1:10],col="red",lwd=1)
```

Exercise : probability of occurrence with substitutions

Assuming that a DNA sequence contains the same proportion of A, C, G, T, what is the probability to observe, at a given position of this sequence, the word `CAGTGAT`

1. without substitution ?
2. with exactly one substitution ?
3. with at most one substitution ?
4. with at most 3 substitutions ?
5. with at most 6 substitutions ?
6. Plot the distribution of probabilities, as a function of the number of substitutions.

Jacques van Helden (jvhelden@ulb.ac.be)