Discrete distributions for the analysis of Next Generation Sequencing (NGS) data

Jacques van Helden

First version: 2016-12-10; Last update: 2016-12-12

Introduction

Let us experiment first

Perfect match probability

Geometric distribution: local alignment without mismatch

Binomial: global alignment with m mismatches

Negative binomial: local alignment with at most m mismatches

Negative binomial for over-dispersed counts

Introduction

Discrete probabilities and NGS

The advent of Next Generation Sequencing (**NGS**) technologies revived the importance of discrete distributions of probabilities for biologists.
This tutorial aims at providing a rapid overview of some discrete distributions commonly used to analyse NGS data, and highlight the relationship between them.

Overview

Distribution	Applications
Geometric	Local read mapping without mismatch (read extension until first mismatch)
Binomial	Global read mapping with a given number of mismatches
Negative binomial	Local read mapping with m mismatches (waiting time for $(m+1)^{th}$ mismatch); Detection of differentially expressed genes from RNA-seq data
Poisson	ChIP-seq peak calling
Hypergeometric	Enrichment of a set of differentially expressed genes for functional classes

Let us experiment first

The Poisson distribution

The Poisson is a very simple and widely used discrete distribution.

$$P(X = x) = \frac{e^{-\lambda}\lambda^x}{x!}$$

- represents the probability to observe x successes when expecting λ (say "lambda").
- expected mean (for a sample of infinite size): $\mu = \lambda$
- expected variance: $\sigma^2 = \lambda$
- **More info**: read the help for the Poisson distribution: `help(Poisson)`

Exercise – Poisson distribution

- open collective result table
- login with the email on which you were invited
- each student has been assigned a λ comprized between 0.01 and 1000
- draw $rep = 1000$ random numbers following a Poisson with this λ value
- compute the mean and variance
- fill up the corresponding columns in the collective report

Solution – mean and variance of a Poisson random sampling

```
lambda <- 3
rep <- 1000
x <- rpois(n=rep, lambda=lambda)
mean(x)
```

```
[1] 3.08
```

```
var(x)
```

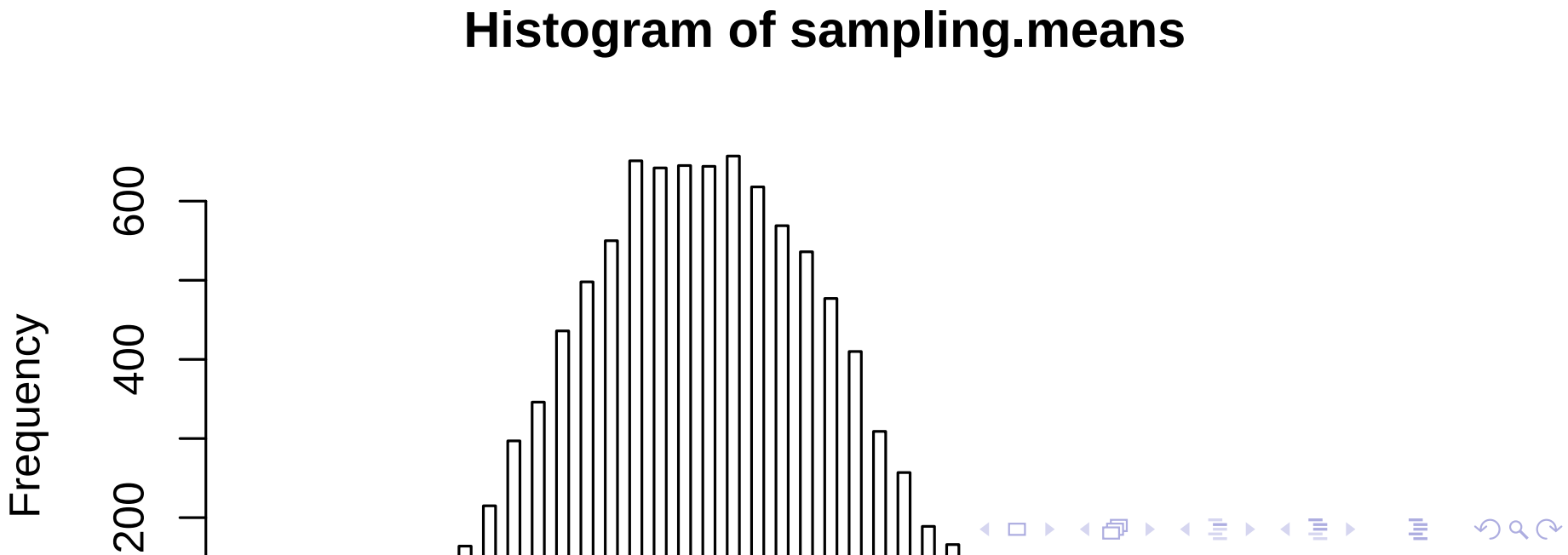
```
[1] 3.07
```

Replicating an experiment

- read the help for `runif()` and `replicate()`
- make 1000 experiments consisting of the following steps:
  - select at random a λ value between 0.5 and 1000
  - draw $n = 10$ random numbers following a Poisson with this λ
  - compute the mean and variance
- plot the relationship between mean and variance for the Poisson distribution

Solution – mean to variance relationship for the Poisson distribution

```
# ?replicate
## Example of usage of the replicate function
sampling.means <- replicate(n = 10000, mean(rpois(n=10, lam

hist(sampling.means, breaks=100)
```



Poisson mean vs variance

```
# Define a data frame with 2 columns indicating
# the mean and variance of the random Poisson samples.
rpois.stats <- data.frame(
  mean=unlist(lapply(result, mean)),
  var=unlist(lapply(result, var))*9/10
)

# Plot the relationship between mean and variance
plot(x=rpois.stats$mean,
     y=rpois.stats$var, col="grey",
     main="Random poisson sampling",
     xlab="mean", ylab="variance")
grid()
abline(a=0, b=1, col="darkgreen", lwd=2)
```

Random poisson sampling

Poisson mean vs coefficient of variation

$$V = m/s^2$$

```
# Compute the coefficient of variation
rpois.stats$V <- rpois.stats$mean / rpois.stats$var

# Check the mean of the coefficient of variation
mean(rpois.stats$V)
```

```
[1] 1.43
```

```
# Plot the relationship between mean and coefficient of va
plot(rpois.stats$mean, rpois.stats$V, col="grey",
     main="Random poisson sampling",
     xlab="mean", ylab="variance")
grid()
abline(h=1, col="darkgreen", lwd=2)
```

Perfect match probability

Perfect match probability

We align a library of 50 million short reads of 25 base pairs onto a genome that comprises 23 chromosomes totalling 3 Gigabases.
For the sake of simplicity, we assume that nucleotides are equiprobable and independently distributed in the genome.
What is the probability to observe the following events by chance?

1. A perfect match for a given read at a given genomic position.
2. A perfect match for a given read anywhere in the genome (searched on two strands).
3. A perfect match for any read of the library at any position of the genome.
4. How many matches do we expect by chance if the whole library is aligned onto the whole genome?

Perfect match - parameters

Let us define the variables of our problem. Since we assume equiprobable and independent nucleotides we can define p as probability to observe a match by chance for a given nucleotide.

$$p = P(A) = P(C) = P(G) = P(T) = 0.25$$

```
k <- 25     # Read length
L <- 50e6   # Library size
C <- 23     # Number of chromosomes
G <- 3e9    # Genome size
p <- 1/4    # Matching probability for a nucleotide
```

Exercise: use these parameters to compute the matching probability for a read (*solution is on next slide*).

Perfect match for a given read at a given genomic position

Since we assume independence, the joint probability (probability to match all the nucleotides) is the product of the individual matching probabilities for each nucleotide.

```
# Matching probabilty for a given read
# at a given genomic position
P.read <- p^k
```

$$P_{\text{read}} = P(n_1 \wedge n_2 \wedge \ldots \wedge n_k) = p^k = 0.25^{25} = 8.9e-16$$

This looks a rather small probability. However we need to take into account that this risk will be challenged many times:

- the size of the genome (3 000 000 000)
- the size of the sequencing library (50 000 000)

Number of genomic alignments

The read will be aligned to each genomic position, but we should keep in mind the following facts.

1. For each chromosome, we will skip the last 24 positions, since a 25 bp read cannot be fully aligned there.
2. We double the number of alignments since we try to map the read on two strands.

$$N = 2 \sum_{i=1}^{C} (L_i - k + 1) = 2 (G - C(k - 1))$$

```
N <- 2 * (G - C * (k - 1))
```

In total, we will thus try to align each read on 5 999 998 896 genomic positions.

Genome-wise matching probability for one read

We reason in 3 steps, by computing the following probabilities.

Formula	Rationale
$1 - P_{\text{read}} = 1 - p^k$	no match at a given genomic position
$(1 - P_{\text{read}})^N$	not a single match in the genome
$1 - (1 - P_{\text{read}})^N$	at least one match in the genome

```
P.genomic <- 1 - (1 - P.read)^N
```

This gives $P_{\text{genomic}} = 0.00000533$.

Library-wise probability

We can apply the same reasoning for the library-wise probability.

Formula	Rationale
$1 - P_{\text{genomic}} = (1 - P_{\text{read}})^N$	no genomic match for a given read
$(1 - P_{\text{read}})^{NL}$	not a single genomic match in the library
$1 - (1 - P_{\text{read}})^{NL}$	at least one genomic match in the library

```
P.library <- 1 - (1 - P.read)^(N*L)
```

This gives $P_{\text{library}} = 1$, which should however not be literally interpreted as a certainty, but as a probability so close to 1 that it cannot be distiguished from it.

Expected number of matches

The expected number of matches is the read matching probability mutliplie by the number of matching trials, i.e. $G \cdot L$ since each read will be matched against each genomic position.

$$E(X) = P_{read} \cdot N \cdot L$$

```
E <- P.read * N * L
```

In total, we expect 266 perfect matches by chance for the whole library against the whole genome.

Geometric distribution: local alignment without mismatch

Local alignment until the first mismatch

A local read-mapping algorithm starts by aligning the 5' base of a read, and extends the alignment until either the first mismatch or the end of the read. In the example below, the alignment stops after 11 nucleotides.

```
    ATGCG ACTAG CATAC GAGTG ACTAA
    11111 11111 10
... ATGCG ACTAG CGTTC GACTG ACTAA ...
```

What is the probability to obtain by chance:

1. an alignment of exactly $x = 11$ nucleotides (11 matches followed by 1 mismatch)?
2. an alignment of at least $x = 11$ nucleotides (11 matches followed by anything)?

Local alignment – parameters

```
p <- 0.25   # Matching probability for each nucleotide
x <- 11     # Number of matches before the first mismatch
P.x <- p^x * (1-p)
Pval.x <- p^x
```

$$P(X = 11) = p^x(1 - p) = 0.25^{11}0.75 = 0.000000179$$

$$P(X \geq 11) = p^x = 0.25^{11} = 0.000000238$$

Geometric distribution

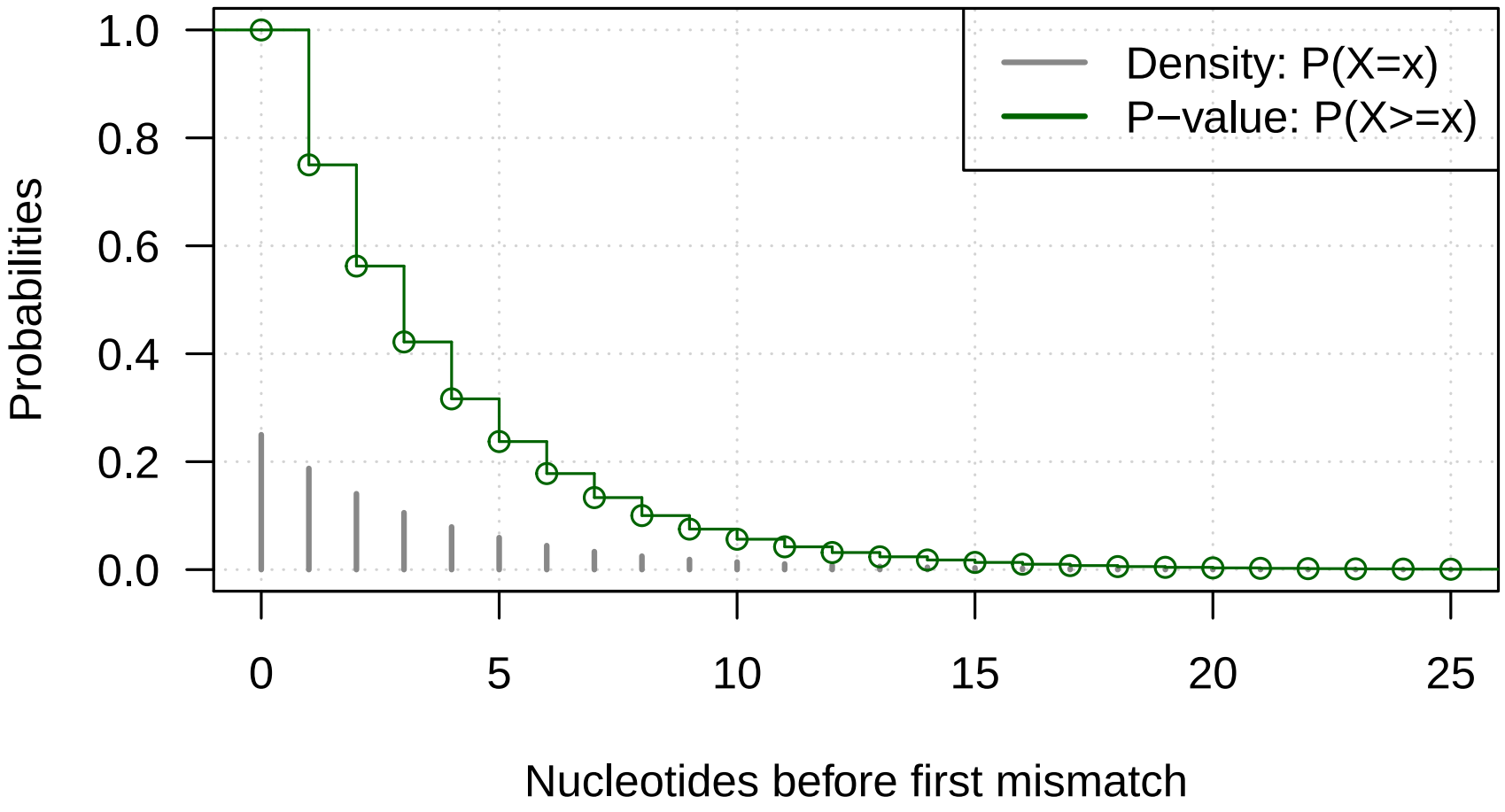


Figure 1: Geometric distribution.

Binomial: global alignment with m mismatches

Global alignment with mismatches

What is the probability to observe a global alignment with at most $m = 3$ mismatches for a given read of 25bp aligned on a particular genomic position?
This question can be formulated as a Bernoulli schema, where each nucleotide is a trial, which can result in either a success (nucleotide match between the read and the genome) or a failure (mismatch).
We can label each position of the alignment with a Boolean value indicating whether it maches (1) or not (0), as examplified below.

```
    ATGCG ACTAG CATAC GAGTG ACTAA
    11111 11111 10101 11011 11111
... ATGCG ACTAG CGTTC GACTG ACTAA ...
```

At each position, we have a probability of success $p = 0.25$, and a probability of failure $q = 1 - p = 0.75$.

Probability to observe exactly k matches

```
n <- 25      # Number of trials, i.e. the length of the alig
m <- 3       # Maximal number of accepted mismatches
k <- n -m    # Number of matches
p <- 1/4     # Matching probability for one nucleotide
```

Let us denote by k the number of matching residues. The probability to observe k successes in a Bernoulli schema with n trials and

$$P(X = k) = \mathcal{B}(k; n, p) = \binom{n}{k} p^k (1-p)^{n-k} = \frac{n!}{k!(n-k)!} p^k (1-p)^{n-k}$$

Properties of the binomial distribution

- Mean $= n \cdot p$
- Variance $= n \cdot p \cdot (1 - p)$
- Shape:
  - i-shaped when p is close to 0,
  - j-shaped when p is close to 1,
  - bell-shaped for intermediate values of p.

Binomial and perfect match

Remark: the perfect match probability seen above is a particular case of the binomial.

$$P(X = n) = \frac{n!}{n!0!} p^n (1-p)^{n-n} = p^n$$

Probability of hit with at most m mismatches

We can sum the probabilities for all possible values of matches from $k = n - m$ (m mismatches) to $k = n$ (no mismatch).

$$P(M \leq m) = \sum_{k=n-m}^{n} \binom{n}{k} p^k (1-p)^{n-k}$$

Binomial density

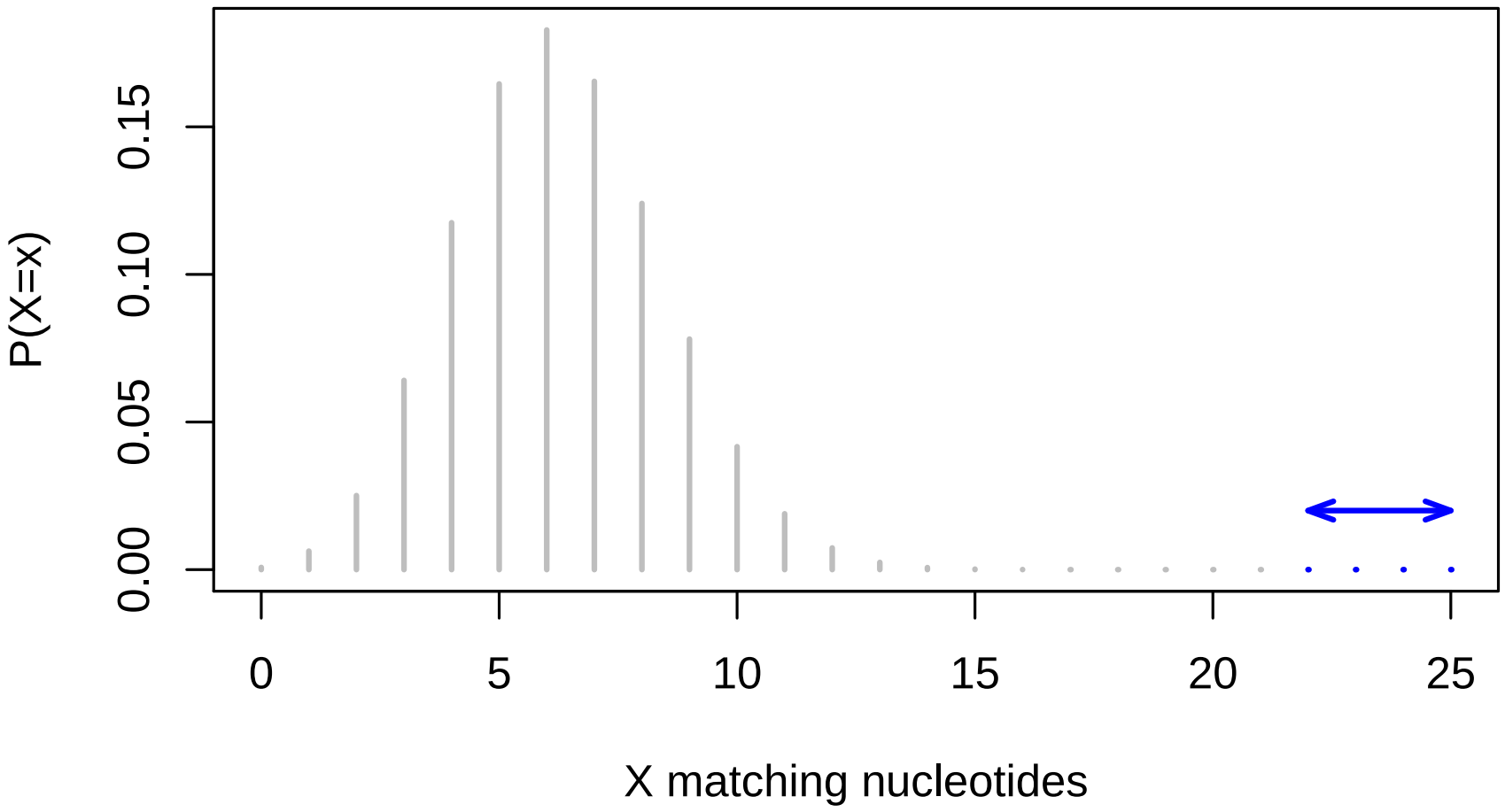


Figure 2: Binomial density function. Alignemnts with at most m mismatches are highlighted in blue.

Binomial P-value

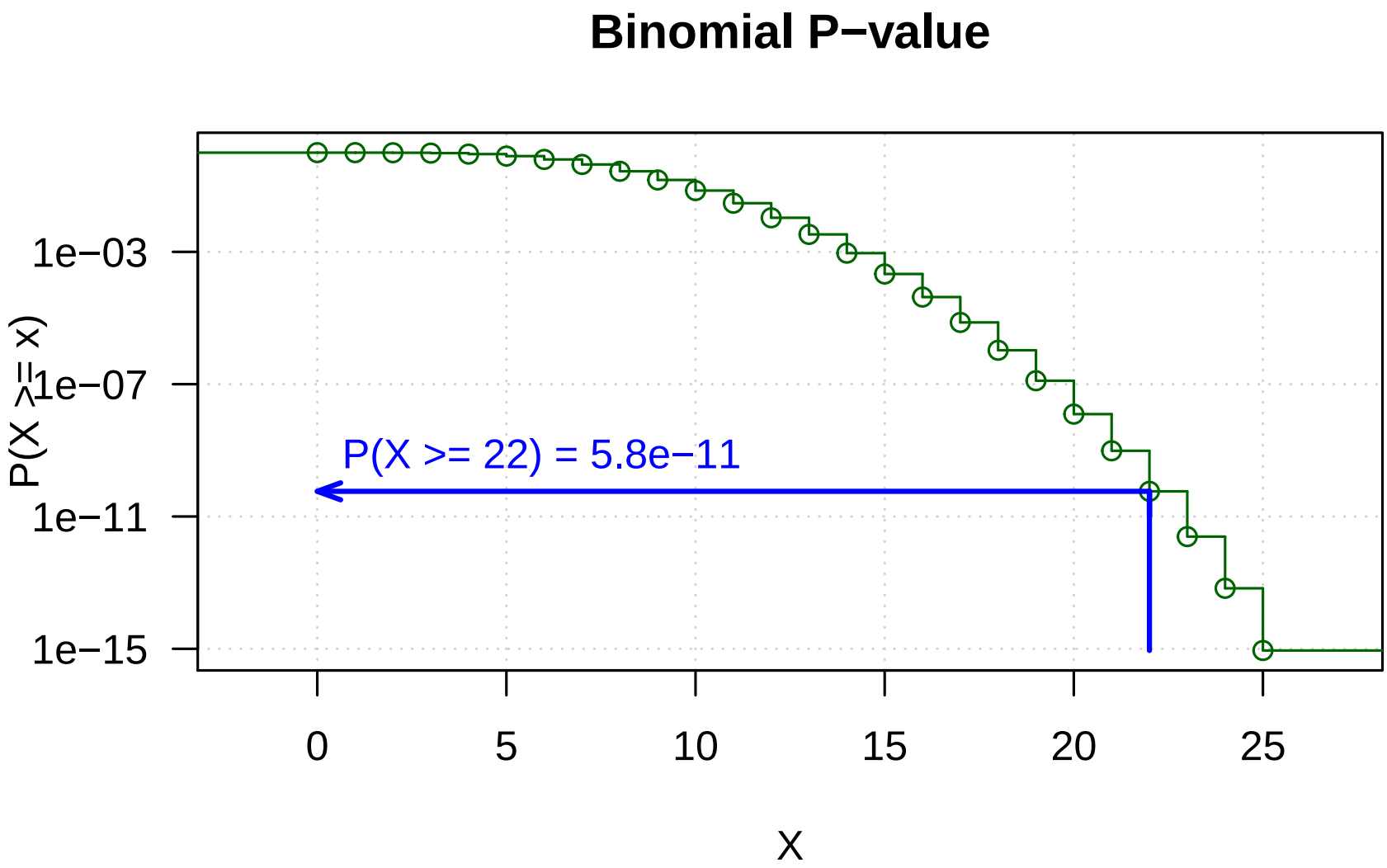


Figure 3: Binomial p-value. The ordinate indicates the probability to

Simulated sequences

We can generate random sequences with equiprobable and independent residues from the nucleotide alphabet.

$$\mathcal{A} = \{A, C, G, T\}$$

```
TATCCCCGCTATATCCTTGGAGGAG
CAGGTACTGCGTATTACGTAGTAGT
CGATGACTTTCATTGAATCACACAC
GTAAACGCTATAGGAGCTGGGCCCC
AGGACCAGGGATCCTCGGTCACCAG
...
```

Mismatch count simulation

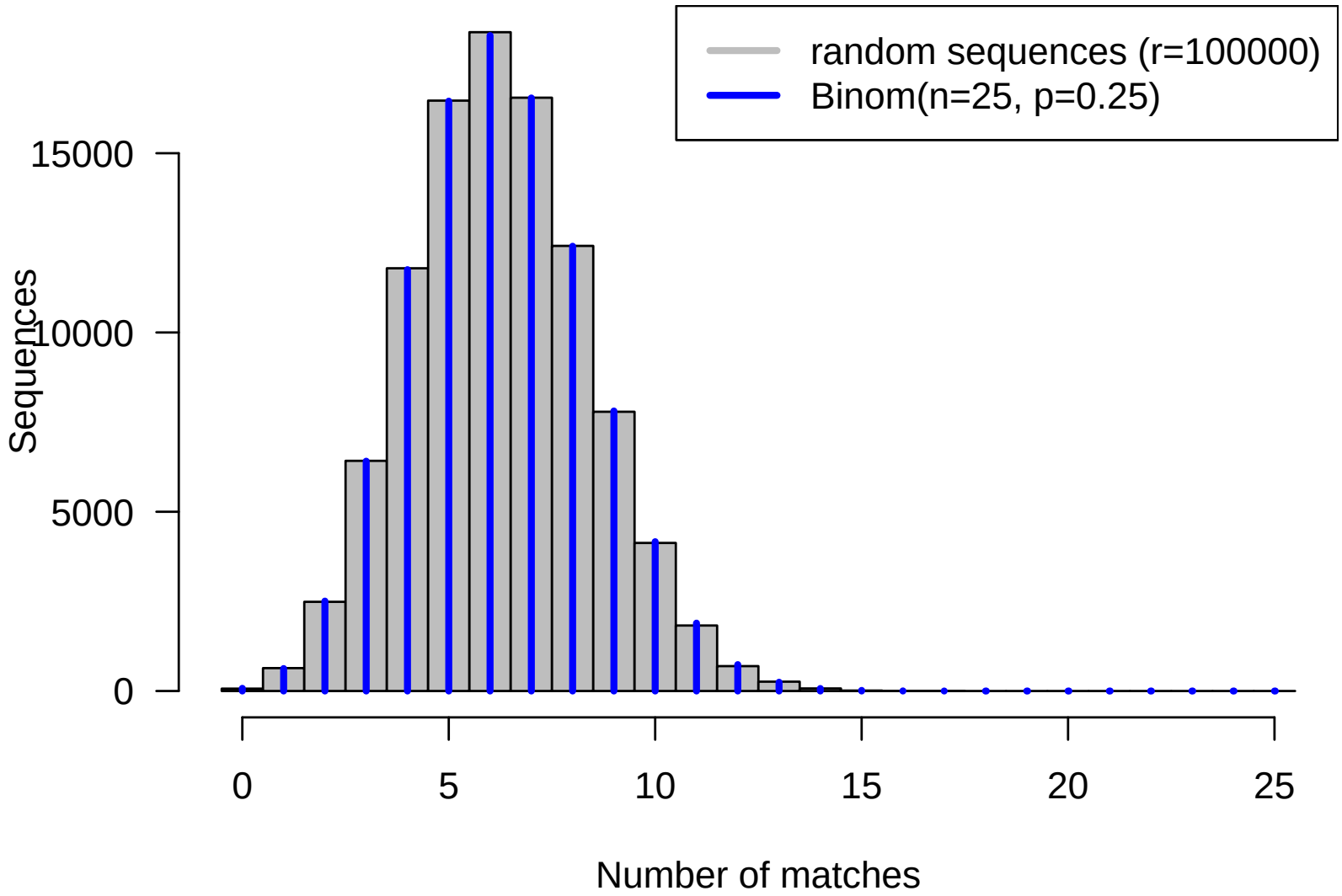


Figure 4: Global alignment simulation. A random read is aligned on

Exercise – binomial Parameters

Each student will take a custom prior probability (p) among the following values: $\{0.001, 0.01, 0.1, 0.25, 0.5, 0.75, 0.9, 0.99, 0.999\}$.

1. Draw 10000 random numbers from a binomial distribution (`rbinom()`) with the custom p and 25 trials.
2. Compute the expected mean and variance.
3. Compute the classical descriptive statistics: mean, variance, standard deviation.
4. Fill up the form on the collective result table
5. Plot an histogram of the numbers drawn.
6. Overlay the theoretical distribution and check the consistency.

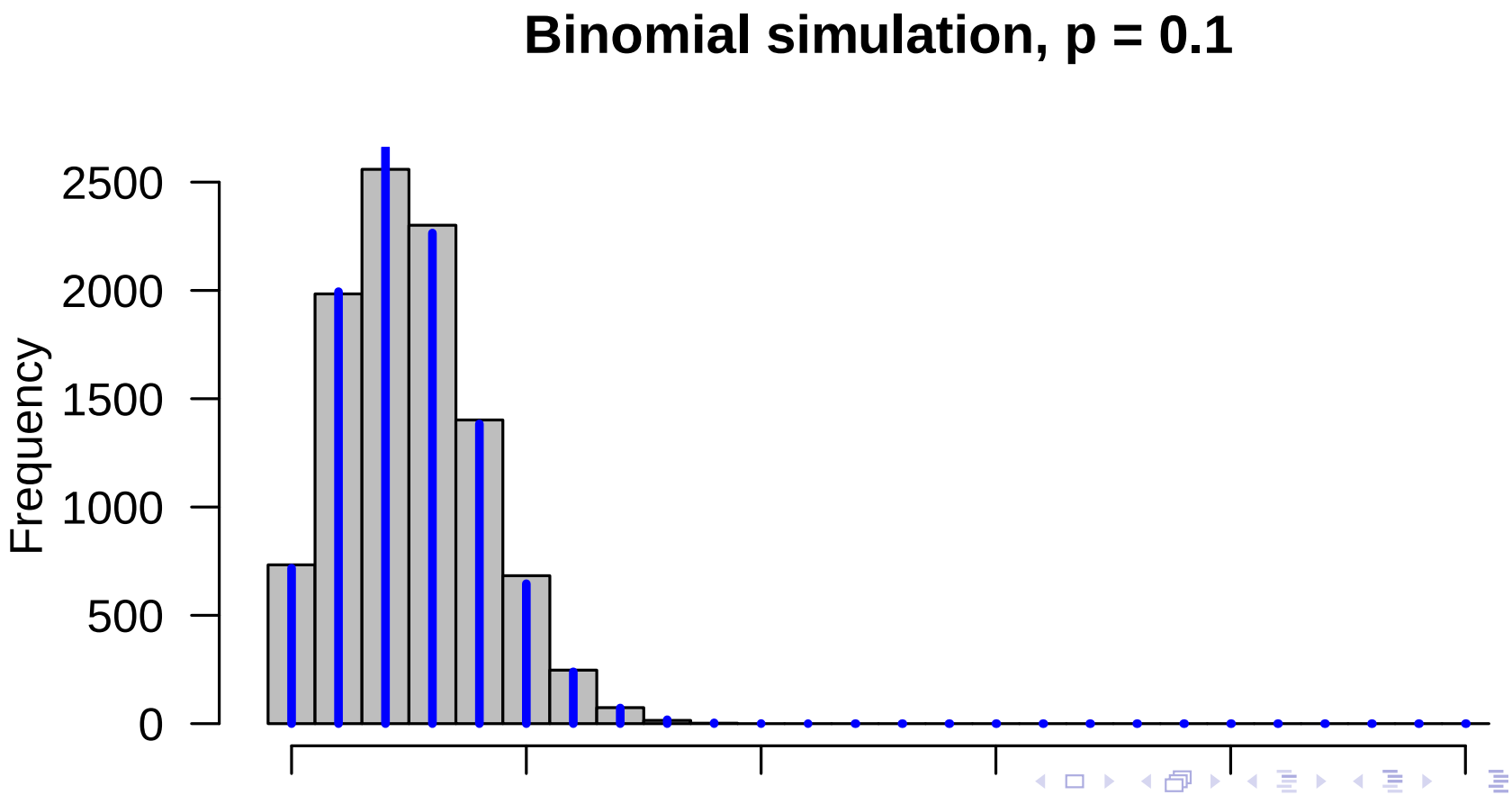
Solution – binomial

```
rand.rep <- 10000 # Random sample size
p <- 0.1          # Prior probability
n <- 25           # Number of trials for the binomial
exp.mean <- n*p   # Expected number of successes
exp.var <- n*p*(1-p)

# Generate random numbers
x <- rbinom(n = rand.rep, size = n, prob = p)

# Compute statistics
stats <- data.frame(p = p, n = n,exp.mean=exp.mean, mean=me
  exp.var = exp.var, variance=var(x), sd=sd(x))
kable(stats, digits=4)
```

p	n	exp.mean	mean	exp.var	variance	sd

Solution – binomial plot

```
hist(x, breaks=(0:26)-0.5, col="grey", main=paste("Binomial
     xlab="Successes", ylab="Frequency", las=1)
lines(0:25, rand.rep*dbinom(x = 0:25, size = n, prob = p),
```



Negative binomial: local alignment with at most m mismatches

Local alignment with mismatches: problem statement

A local alignment algorithm starts from the 5' end of a read, and stops either at the x^{th} mismatch or when the end of the read is reached. What is the probability to obtain by chance an alignemnt of exactly 25 nucleotides with exactly $m = 3$ mismatches?
This amounts to obtain exactly $k = 22$ matches and $m = 3$ mismatches (in any order), followed by a mismatch at the $(k + m + 1)^{th}$ position.
We show here some examples of local alignments with at most 5 mismatches. Note that the last residue can be either a match (uppercase) or a mismatch (lowercase).

```
AACAATAACTTATTCATGGGTCATT

tAtccc
cAggta
cgatg
```

Number of successes before the r^{th} failure

The **negative binomial** distribution (also called **Pascal distribution**) indicates the probability of the number of successes (k) before the r^{th} failure, in a Bernoulli schema with success probability p.

$$\mathcal{NB}(k|r,p) = \binom{k+r-1}{k} p^k (1-p)^r$$

This formula is a simple adaptation of the binomial, with the difference that we know that the last trial must be a failure. The binomial coefficient is thus reduced to choose the k successes among the $n-1 = k+r-1$ trials preceding the r^{th} failure.

Alternative formulation

It can also be adapted to indicate related probabilities.

- Number of **failures** (r) before the k^{th} **success**.

$$\mathcal{NB}(r|k,p) = \binom{k+r-1}{r} p^k (1-p)^r$$

- Number of **trials** ($n = k + r - 1$) before the r^{th} **failure**.

$$\mathcal{NB}(n|r,p) = \binom{n-1}{r-1} p^{n-r} (1-p)^r$$

Properties of the negative binomial

Negative binomial density

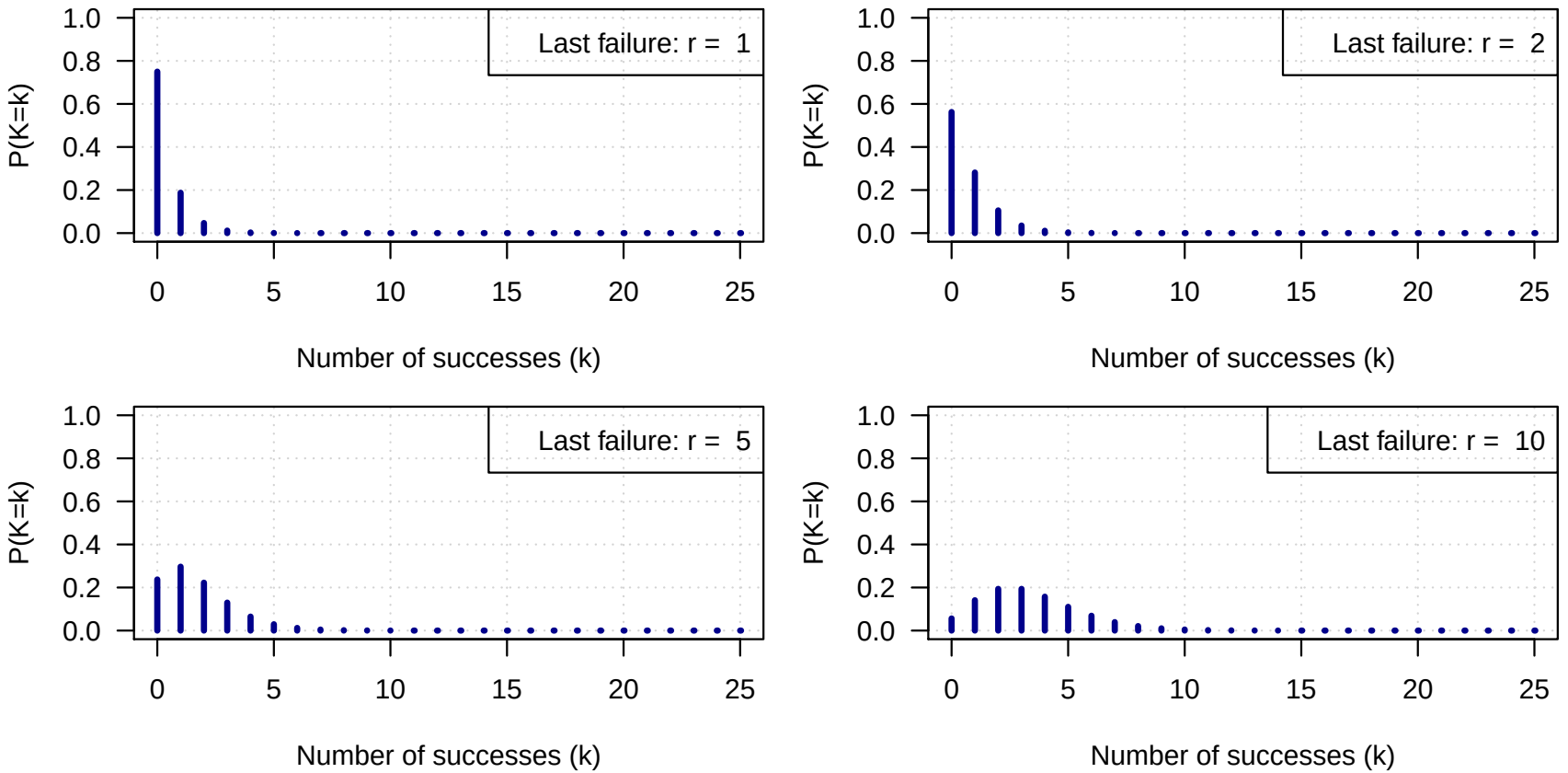
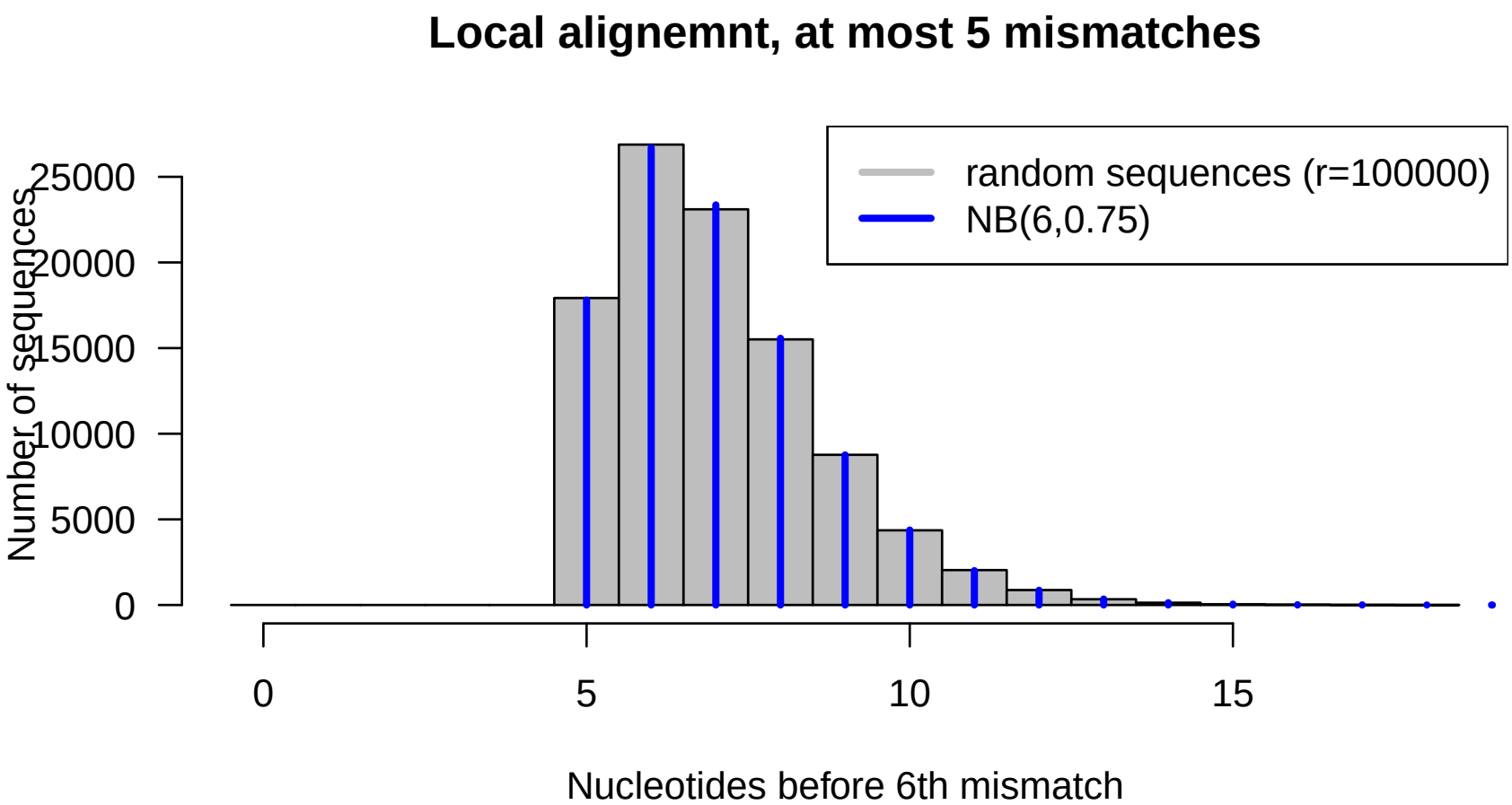


Figure 5: Negative binomial.

Local alignment with simulated sequences



Exercise – Negative binomial

Each student chooses a value for the maximal number of failures (r).

1. Read carefully the help of the negative binomial functions: `help(NegBinomial)`
2. **Random sampling**: draw of $rep = 100000$ random numbers from a negative binomial distribution (`rndbinom()`) to compute the distribution of the number of successes (k) before the r^{th} failure.
3. Compute the expected mean and variance of the negative binomial.
4. Compute the mean and variance from your sampling distribution.
5. Draw an histogram with the number of successes before the r^{th} failure.
6. Fill up the form on the collective result table

Solution to the exercise – negative binomial

```
r <- 6          # Number of failures
p <- 0.75       # Failure probability
rep <- 100000
k <- rnbinom(n = rep, size = r, prob = p)
max.k <- max(k)
exp.mean <- r*(1-p)/p
rand.mean <- mean(k)
exp.var <- r*(1-p)/p^2
rand.var <- var(k)
hist(k, breaks = -0.5:(max.k+0.5), col="grey", xlab="Numbe
     las=1, ylab="", main="Random sampling from negative bi
abline(v=rand.mean, col="darkgreen", lwd=2)
abline(v=exp.mean, col="green", lty="dashed")
arrows(rand.mean, rep/20, rand.mean+sqrt(rand.var), rep/20,
       angle=20, length = 0.1, col="purple", lwd=2)
```

Negative binomial for over-dispersed counts

To be treated in the afternoon !