

# Esercitazione N° 1: La miostatina

Piero Rivoira  
Istituto Agrario Penna – Asti  
[piero.rivoira@yahoo.it](mailto:piero.rivoira@yahoo.it)

## Creiamo il nostro profilo su Ubuntu

# questo è un commento!

# lanciamo il terminale per aggiungere un nuovo profilo

**ctrl+alt+t**

\$ **sudo adduser nome\_battesimo -uid 663**

\$ pw di labx

# per eliminare un profilo utente (in caso di errore)

\$ **sudo deluser --remove-home nome\_utente**

\$ pw di labx

# per recuperare la passwd dimenticata

# sostituire <user name> con il proprio nome utente

\$ **sudo passwd user name**

\$ pw di labx

\$ nuova pw

# Creiamo il nostro profilo su Ubuntu

# acquisiamo i privilegi dell'amministratore di sistema modificando il file di configurazione del Sistema Operativo (SO) /etc/sudoers

\$ **sudo visudo**

# questo comando lancia l'editor di testo **nano** per creare ed aprire il file /etc/sudoers.tmp in modalità scrittura (per poterlo modificare); tale file è una copia di backup di /etc/sudoers

GNU nano 7.2 /etc/sudoers.tmp

# Creiamo il nostro profilo su Ubuntu

```
# This file MUST be edited with the 'visudo' command as root.
#
# Please consider adding local content in /etc/sudoers.d/ instead of
# directly modifying this file.
#
# See the man page for details on how to write a sudoers file.
#
Defaults        env_reset
Defaults        mail_badpass
Defaults        secure_path="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/snap/bin"

# This fixes CVE-2005-4890 and possibly breaks some versions of kdesu
# (#1011624, https://bugs.kde.org/show_bug.cgi?id=452532)
Defaults        use_pty

# This preserves proxy settings from user environments of root
# equivalent users (group sudo)
#Defaults:%sudo env_keep += "http_proxy https_proxy ftp_proxy all_proxy no_proxy"

# This allows running arbitrary commands, but so does ALL, and it means
# different sudoers have their choice of editor respected.
#Defaults:%sudo env_keep += "EDITOR"

# Completely harmless preservation of a user preference.
#Defaults:%sudo env_keep += "GREP_COLOR"

# While you shouldn't normally run git as root, you need to with etckeeper
#Defaults:%sudo env_keep += "GIT_AUTHOR_* GIT_COMMITTER_"

# Per-user preferences; root won't have sensible values for them.
#Defaults:%sudo env_keep += "EMAIL DEBEMAIL DEBFULLNAME"

# "sudo scp" or "sudo rsync" should be able to use your SSH agent.
#Defaults:%sudo env_keep += "SSH_AGENT_PID SSH_AUTH_SOCK"

# Ditto for GPG agent
#Defaults:%sudo env_keep += "GPG_AGENT_INFO"

# Host alias specification

# User alias specification
```

# Creiamo il nostro profilo su Ubuntu

GNU nano 7.2 /etc/sudoers.tmp

```

# This file specifies the proxy settings that is set in environment variables for the
# equivalent users (group sudo)
Defaults:%sudo env_keep += "http_proxy https_proxy ftp_proxy all_proxy no_proxy"

# This allows running arbitrary commands, but so does ALL, and it means
# different sudoers have their choice of editor respected.
Defaults:%sudo env_keep += "EDITOR"

# Completely harmless preservation of a user preference.
Defaults:%sudo env_keep += "GREP_COLOR"

# While you shouldn't normally run git as root, you need to with etckeeper
Defaults:%sudo env_keep += "GIT_AUTHOR_* GIT_COMMITTER_"

# Per-user preferences; root won't have sensible values for them.
Defaults:%sudo env_keep += "EMAIL DEBEMAIL DEBFULLNAME"

# "sudo scp" or "sudo rsync" should be able to use your SSH agent.
Defaults:%sudo env_keep += "SSH_AGENT_PID SSH_AUTH_SOCK"

# Ditto for GPG agent
Defaults:%sudo env_keep += "GPG_AGENT_INFO"

# Host alias specification
# User alias specification
# Cmnd alias specification

# User privilege specification
root    ALL=(ALL:ALL) ALL
piero   ALL=(ALL:ALL) ALL

# Members of the admin group may gain root privileges
%admin  ALL=(ALL) ALL

# Allow members of group sudo to execute any command
%sudo   ALL=(ALL:ALL) ALL

# See sudoers(5) for more information on "@include" directives:

@include /etc/sudoers.d

```

**# Elenco degli amministratori di sistema**

**# portarsi con il cursore sull'ultima riga della lista**

**alt-6 # copia**

**ctrl-u # incolla l'intera riga**

**# inserire il proprio nome utente**

**ctrl-o # per salvare**

**# cancellare l'estensione .tmp**

**ctrl-x # per chiudere nano**

**alt-u # in caso di errore**  
(per annullare l'ultimo comando inserito)

**Elenco degli amministratori di sistema**

▲ Guida □ Salva □ Cerca □ Esegui □ Posizione □ Annnulla  
■ Esci □ Inserisci □ Sostituisci □ Taolia □ Giustifica □ Vai a riga □ Ripeti □ Contrassegna □ Parentesi  
■ Precedente □ Successiva □ Avanti

# L'incrocio industriale o F1 toro da carne X vacca da latte

- **Razza Belga** = razza incrociante più utilizzata in Piemonte, allevata in purezza solo in Belgio

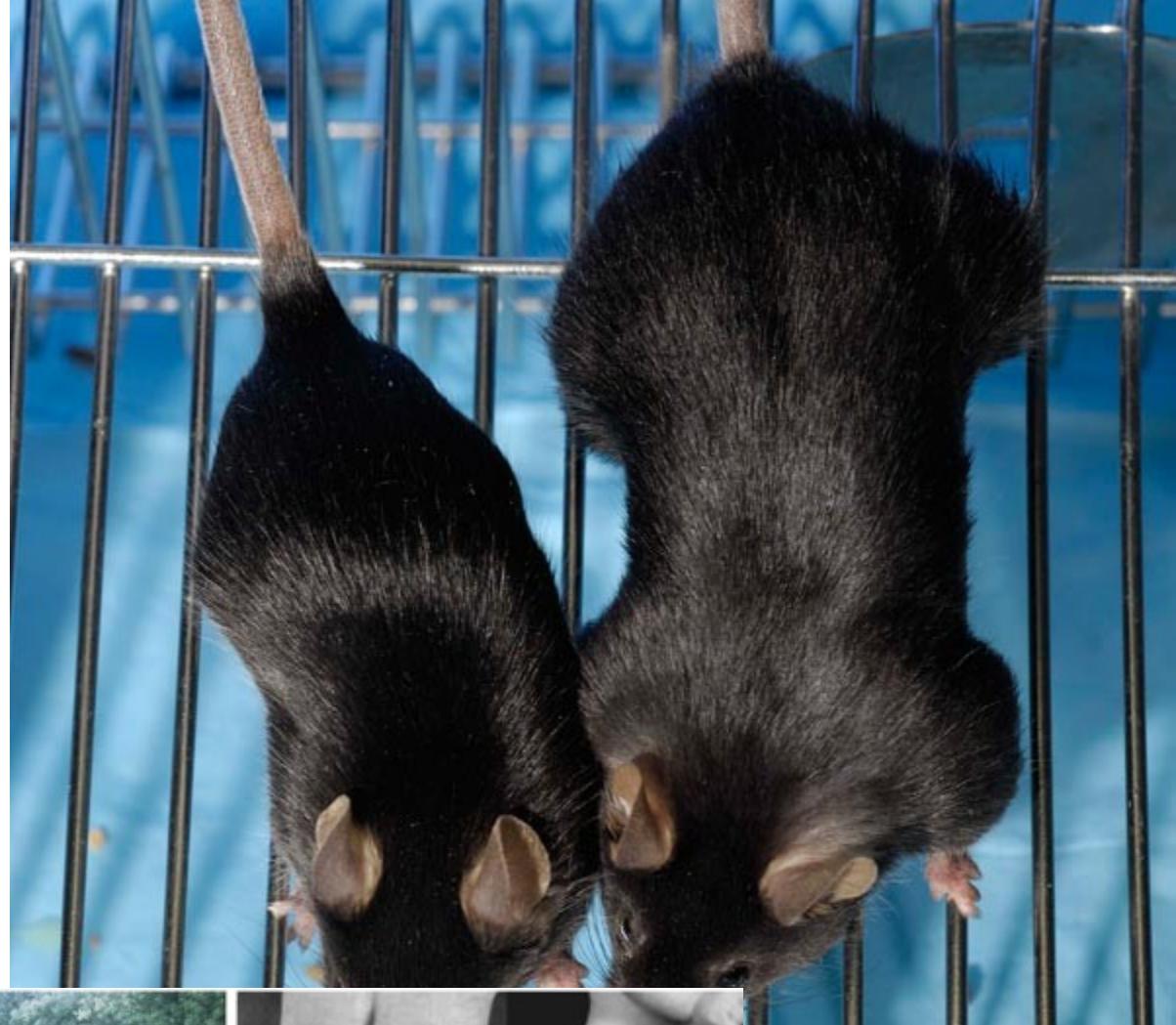


- Vitelli ibridi spuntano prezzi  $\leq$  4aa Euro  
[Piemontese puro  $\sim$  5aa E, frisone puro (*gaiot*) 5a E]
- Quante bovine da latte posso inseminare con il Belga senza compromettere la **rimonta interna**?

# La razza Belga e la miostatina

Phenotype of myostatin null vertebrates.

- a) upper forelimb muscles of wild-type mouse;
- B) upper forelimb of mouse that is myostatin null;
- C) bovine with myostatin mutation;
- D) and E) human with myostatin mutation as a neonate (D) and at 7 mo of age.



# La razza Belga e la miostatina

A screenshot of a web browser showing the Google Scholar homepage. The URL in the address bar is https://scholar.google.com. The page features the Google Scholar logo at the top center. Below it is a search bar containing the query "Double muscling in cattle due to mutations in the myostatin gene". To the right of the search bar is a blue search button with a magnifying glass icon. Underneath the search bar are two radio button options: "Qualsiasi lingua" (selected) and "Pagine in Italiano". At the bottom of the page, there is a green link labeled "Sali sulle spalle dei giganti". The browser interface includes standard navigation buttons (back, forward, refresh), a zoom level indicator (133%), and various icons for saving and sharing.

Double muscling in cattle due to mutations in the myostatin gene

Qualsiasi lingua Pagine in Italiano

Sali sulle spalle dei giganti



RESEARCH ARTICLE | GENETICS |

f X in



## Double muscling in cattle due to mutations in the myostatin gene

Alexandra C. McPherron and Se-Jin Lee [Authors Info & Affiliations](#)

November 11, 1997 | 94 (23) 12457-12461 | <https://doi.org/10.1073/pnas.94.23.12457>

# La razza Belga e la miostatina

the GenBank dat... 1

GenBank

Proc. Natl. Acad. Sci. USA  
Vol. 94, pp. 12457–12461, November 1997  
Genetics

Check for updates

## Double muscling in cattle due to mutations in the myostatin gene

ALEXANDRA C. MCPHERRON AND SE-JIN LEE\*

Department of Molecular Biology and Genetics, Johns Hopkins University School of Medicine, 725 North Wolfe Street, Baltimore, MD 21205

Communicated by Daniel Nathans, Johns Hopkins University School of Medicine, Baltimore, MD, August 26, 1997 (received for review August 12, 1997)

# La razza Belga e la miostatina

---

Data deposition: The sequences reported in this paper have been deposited in the GenBank database [baboon (accession no. AF019619), bovine (accession no. AF019620), chicken (accession no. AF019621), ovine (accession no. AF019622), porcine (accession no. AF019623), rat (accession no. AF019624), turkey (accession no. AF019625), zebrafish (accession no. AF019626), and human (accession no. AF019627)].

A commentary on this article begins on page 12249.

\*To whom reprint requests should be addressed. e-mail: [sejin\\_lee@qmail.bs.jhu.edu](mailto:sejin_lee@qmail.bs.jhu.edu).

# La razza Belga e la miostatina

# lanciamo il terminale per usare R

**ctrl+alt+t**

\$ R

```
> install.packages("ape")
> library(ape)
> x = paste("AF0196", 19:27, sep = "")
```

# **ctrl+MAIUSC+v** per incollare nel terminale o in R

```
> myostatin.numbers = c('NM_001001525.3', 'XM_010862903.1', 'XM_005907427.1', 'GQ184147.1', 'AF019620.1',
  'XM_061432883.1', 'BC134563.1', 'AY160688.1', 'XM_027555492.1', 'XM_019969486.1', 'AF019761.1',
  'NM_001290967.1', 'XM_055574715.1', 'KP057613.1', 'AY629304.1', 'XM_055397770.1', 'XM_061135030.1',
  'XM_043894881.1', 'XM_043488279.1', 'AY629307.1', 'XM_020900528.1', 'XM_052635569.1', 'EF629535.1',
  'NM_001009428.3', 'XM_040227852.1', 'JN662463.1', 'AM992883.1', 'AY629305.1', 'AF019622.1', 'KM371731.1',
  'JX840482.1', 'JN639006.1', 'HM462261.1', 'HM462259.1', 'NM_001285737.1', 'FM207636.1', 'AY629309.1',
  'AY629306.1', 'AY436347.1', 'AM992884.1', 'GU377303.1', 'XM_007104442.2', 'XM_022568276.1', 'XM_061203308.1',
  'XM_057551429.1', 'XM_036859085.1', 'AY629308.1', 'XM_029225345.1', 'XM_027114903.1', 'XM_024733638.1',
  'XM_004323907.2', 'XM_030847723.2', 'XM_060151388.1', 'XM_060015612.1', 'XM_060106598.1', 'XM_004276934.2',
  'XM_032637329.1', 'XM_059051852.1', 'XM_059928344.1', 'XM_057745280.1', 'XM_047770464.1', 'HM241657.1',
  'AF019623.1', 'JN630464.1', 'EF612791.1', 'NM_214435.2', 'XM_006209983.3', 'XM_006189223.3', 'XM_028125258.1',
  'XM_027590490.1', 'XM_025887402.1', 'XM_004426733.2', 'XM_010960036.2', 'XM_021703117.1', 'XM_004392403.2',
  'XM_046659551.1', 'XM_046650639.1', 'XM_014837576.2', 'XM_032417983.1', 'XM_010991955.2', 'XM_058548651.1',
  'XM_008516515.1', 'XM_036075506.1', 'NM_001081817.1', 'XM_006726961.2', 'XM_019626478.1', 'XM_054562700.1',
  'XM_036253439.1', 'XM_019740713.1', 'MZ169554.1', 'XM_035007537.1', 'XM_026490450.4', 'XM_045868351.2',
  'XM_045806119.1', 'XM_008687262.2', 'XM_029932956.1', 'XM_016215509.1', 'XM_036881965.2', 'XM_047086026.1',
  'XM_045480514.1')
```

# La razza Belga e la miostatina

```
> x = paste("AF0196", 19:27, sep = "")  
> x = x[-c(1:8)]  
> x  
[1] "AF019627"  
> x = x[9]  
> x  
[1] "AF019627"  
>  
# perché l'output di questi due comandi è il medesimo?
```

# La razza Belga e la miostatina

```
> x = c(x, myostatin.numbers)
> myostatin.seq = read.GenBank(x, as.character = TRUE)
> install.packages("seqinr")
> library(seqinr)

# nel terminale
$ mkdir DNA_SEQUENCES
$ ls
$ pwd

# in R
# alt gr+` → ~
> setwd("~/DNA_SEQUENCES")
> getwd()
> write.fasta(sequences = myostatin.seq, names = names(myostatin.seq), nbchar
=80, file.out = "myostatin.fas")

# nel terminale
$ cd DNA_SEQUENCES
$ gedit myostatin.fas
```

# La razza Belga e la miostatina

# dopo

>AF019620

Aggiungere

>bovine\_belgian

atgcaaaaactgcaaatctctgttatatttacctatttatgctgattgttgcggccagtgatctgaatgagaacag  
cgagcagaaggaaaatgtggaaaaagaggggctgtgtaatgcattgcattgtggagggaaaacactacatcctcaagactag  
aagccataaaaatccaaatcctcagtaaacttcgcctggaaacagtcctaaacatcagcaaagatgctatcagacaactt  
ttgccaaggctccactcctggaaactgattgatcagttcgatgtccagagagatgccagcgtgacggctccttgg  
agacgatgactaccacgccaggacggaaacggcattaccatgcccacggagtctgatcttctaaccgcaagtggaaaggaa  
aacccaaatgtgtttaaatttagtctaagatacaataacttagtaaaggccaaactgtggatatatctg  
aggcctgtcaagactcctgcacagtgttgcataatcctgagactcatcaaaccatgaaagacggtacaaggata  
tggaatccgatctctgaaacttgacatgaaaccaggcactggatttgcagacgttgcataatggccatgatcttgcataacc  
actggctcaaacaacctgaatccaaacttaggcattgaaatcaaagcttagatgaaatggccatgatcttgcataacc  
ttcccagaaccaggagaagatggactgactcctttttagaagtcaaggtaacagacacacacaaaagatcttaggagaga  
tttggcttgcattgtgacagaatctcgatgtcgatgttgcatttgcataactgtggattttgaaagctttggatggattgg  
attattgcacccaaaagatataaggccaaattactgctctggagaatgtgaatttgtattttgaaaagatcttgcataacc  
ccatcttgcaccaagccaaacccagggttcagccggccctgtgtactcctacaaagatgtctccaaattaatatgc  
tatatttaatggcgaaggacaaataatcggaaagattccagccatggtagtagatcgctgtgggtgttgcataacc  
>bovine\_piedmontese

atgcaaaaactgcaaatctctgttatatttacctatttatgctgattgttgcggccagtgatctgaatgagaacag  
cgagcagaaggaaaatgtggaaaaagaggggctgtgtaatgcattgcattgtggagggaaaacactacatcctcaagactag  
aagccataaaaatccaaatcctcagtaaacttcgcctggaaacagtcctaaacatcagcaaagatgctatcagacaactt  
ttgccaaggctccactcctggaaactgattgatcagttcgatgtccagagagatgccagcgtgacggctccttgg  
agacgatgactaccacgccaggacggaaacggcattaccatgcccacggagtctgatcttctaaccgcaagtggaaaggaa  
aacccaaatgtgtttaaatttagtctaagatacaataacttagtaaaggccaaactgtggatatatctg  
aggcctgtcaagactcctgcacagtgttgcataatcctgagactcatcaaaccatgaaagacggtacaaggatata  
tggaatccgatctctgaaacttgacatgaaaccaggcactggatttgcagacgttgcataatggccatgatcttgcataacc  
actggctcaaacaacctgaatccaaacttaggcattgaaatcaaagcttagatgaaatggccatgatcttgcataacc  
ttcccagaaccaggagaagatggactgactcctttttagaagtcaaggtaacagacacacacaaaagatcttaggagaga  
tttggcttgcattgtgacactccacagaatctcgatgtcgatgttgcatttgcataactgtggattttgaaagctttg  
gatgggattggattattgcacccaaaagatataaggccaaattactgctctggagaatatgaattgtattttgaaaag  
tatcctcataccatcttgcaccaagccaaacccagggttcagccggccctgtgtactcctacaaagatgtctcc  
aattaatatgtatatttaatggcgaaggacaaataatcggaaagattccagccatggtagtagatcgctgtgggt  
gttgcataacc

# La razza Belga e la miostatina

```
# ctrl+MAIUSC+s
# save as myostatin_mod.fas
```

Mode: Multiple Alignment Mode Font: 10

Load Sequence File

Cancel Open

Nome Dimensione Tipo Data di modifica

Nome	Dimensione	Tipo	Data di modifica
allineamento_miostatina_blast.txt	480,1 kB	Testo	11 gen
database_completo_miostatina.txt	480,1 kB	Testo	13 gen
dexter.fasta	88,6 kB	Testo	3 set
link_Cervidae.txt	344 byte	Testo	10 gen
mcpheron-lee-1997-double-muscling-in-cattle-due-to-mutations-in-the-myostatin-ge...	537,5 kB	Documento	ven
MT_Cervidae	333,7 kB	Testo	10 gen
MT_Cervidae.aln	602,3 kB	Testo	11 gen
MT_Cervidae.dnd	627 byte	Testo	11 gen
MT_Cervidae.fasta	448,4 kB	Testo	11 gen
my_data.txt	98,9 kB	Testo	27 dic 2023
myostatin.fas	220,0 kB	Testo	10:36
myostatin.fasta	10,6 kB	Testo	Ieri
myostatin.RData	8,8 MB	R Data File	Ieri
MYOSTATIN_ACCESSION_NUMBERS.txt	1,7 kB	Testo	10:26
myostatin_all.fasta	215,1 kB	Testo	12 gen
MYOSTATIN_DATA.txt	17,3 kB	Testo	09:15
<b>myostatin_mod.fas</b>	<b>222,9 kB</b>	<b>Testo</b>	<b>Ieri</b>
myostatin_mod.fasta	740,2 kB	Testo	08:19
piero.RData	574,5 MB	R Data File	14 gen
wolves.csv	1,1 kB	Testo	25 dic 2023

All Files ▾

# La razza Belga e la miostatina



National Library of Medicine

National Center for Biotechnology Information

Search NCBI

AF019620



Search

Results found in 5 databases

NUCLEOTIDE SEQUENCE

Bos taurus myostatin (MSTN) mRNA, complete cds

Bos taurus

1,128 bp mRNA sequence

AF019620.1

FASTA Gene

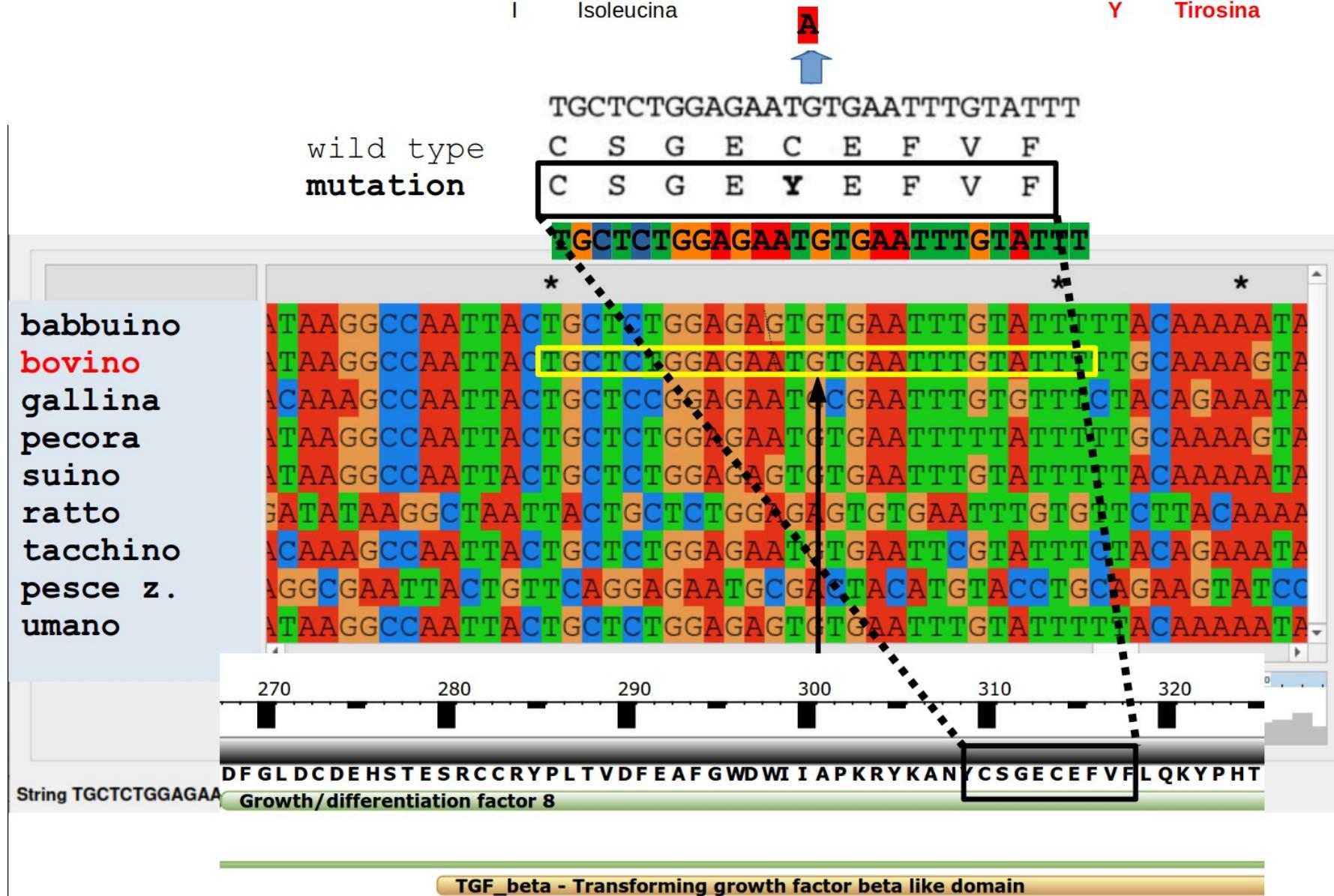
BLAST

Download

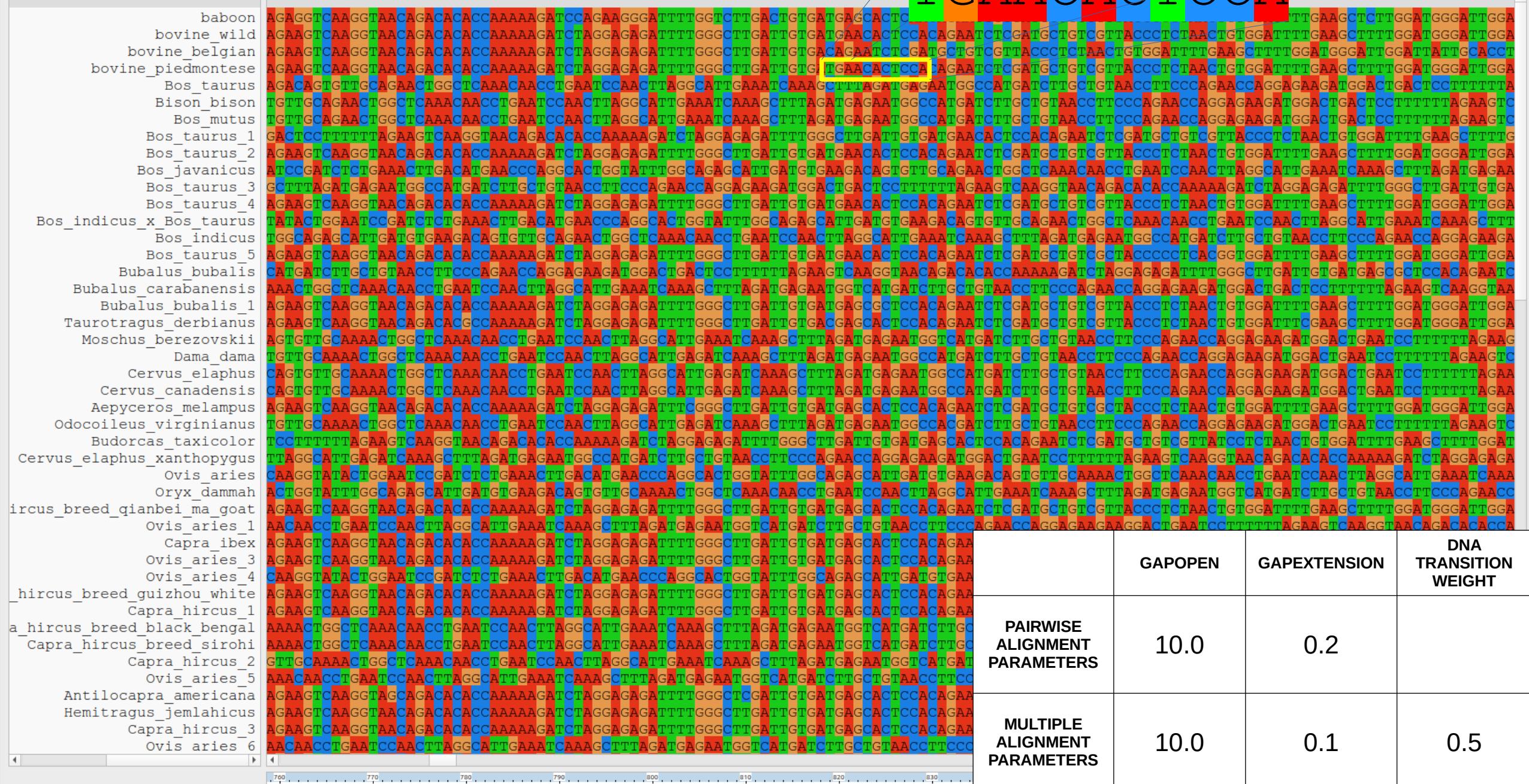
# La razza Piemontese e la miostatina

## Transizione G → A

A	Alanina	K	Lisina	P	Prolina
C	Cisteina	L	Leucina	Q	Glutammina
D	Acido aspartico	M	Metionina	R	Arginina
E	Acido glutammico	N	Asparagina	S	Serina
F	Fenilalanina	T	Treonina	V	Valina
G	Glicina	W	Triptofano	Y	Tirosina
H	Istidina				
I	Isoleucina				



# Miostatina: allineamento multi-sequenza

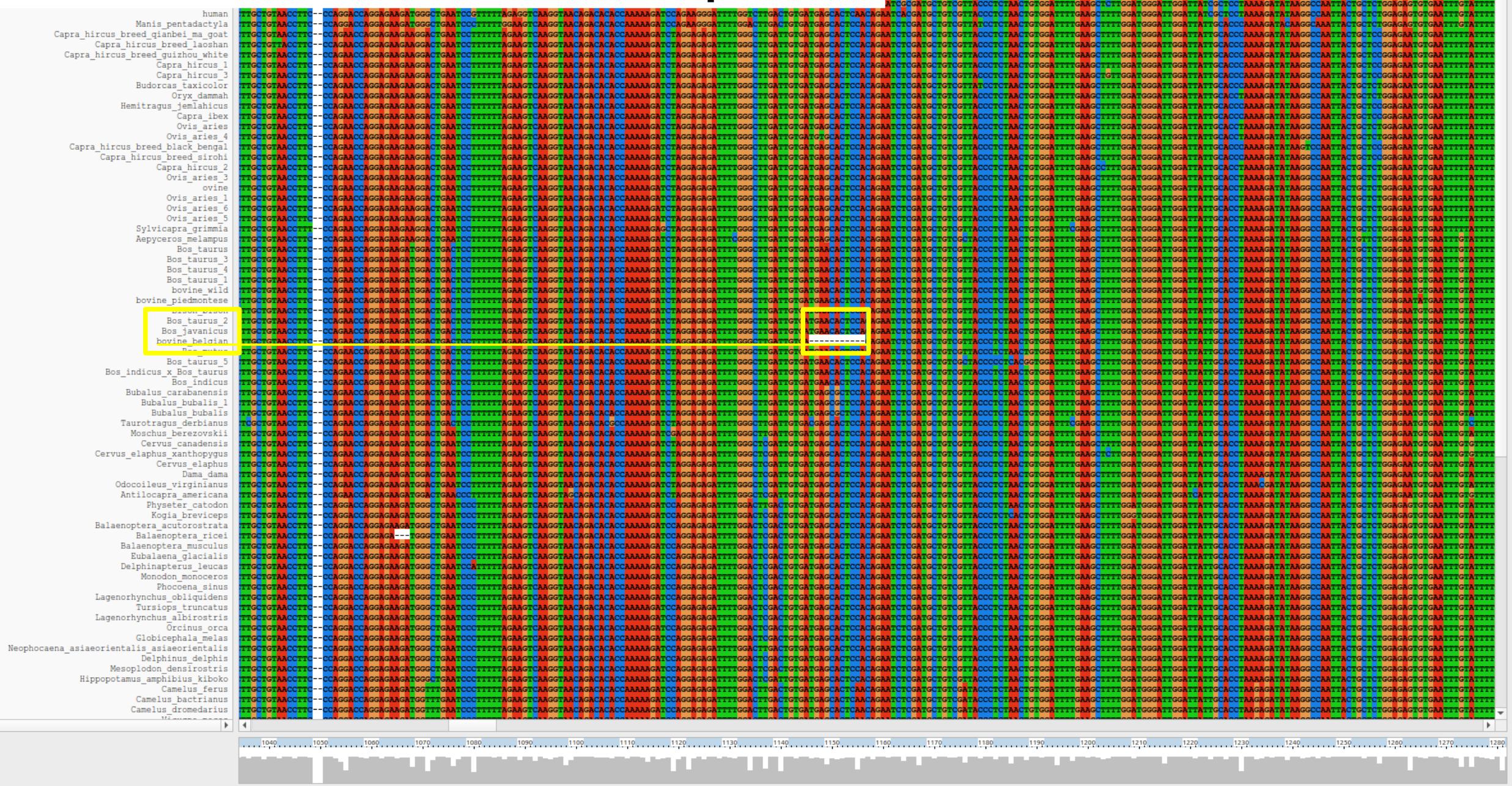


# Miostatina: allineamento multi-sequenza

TGAAACACTCCA

baboon	TCTTGACTGTGATGAGCACTCAACAGAACATCGCGATGCTGTC
ine_wild	GCTTGATTGTGATGA <span style="border: 2px solid yellow;">TGAAACACTCCA</span> CAGAACATCTCGATGCTGTC
_belgian	GCTTGATTGTGACAGAACATCTCGATGCTGTCGTTACCCCTCTA
dmontese	GCTTGATTGTGATGAACACTCCACAGAACATCTCGATGCTGTC
s_taurus	TTGAAATCAAAGCTTTAGATGAGAACATGGCCATGATCTTGCT
on_bison	TCAAAGCTTTAGATGAGAACATGGCCATGATCTTGCTGTAACO
os_mutus	TCAAAGCTTTAGATGAGAACATGGCCATGATCTTGCTGTAACO
taurus_1	GAGAGATTGGGCTTGAATTGTGATGAACACTCCACAGAACAT
taurus_2	GCTTGATTGTGATGAACACTCCACAGAACATCTCGATGCTGTC
avanicus	AGCATTGATGTGAAGACAGTGTGCAGAACATGGCTCAAACAA
taurus_3	GGAGAAAGATGGACTGACTCCCTTTAGAACGTCAAGGTAAC
taurus_4	GCTTGATTGTGATGAACACTCCACAGAACATCTCGATGCTGTC

# Miostatina: allineamento multi-sequenza



# Miostatina: allineamento multi-sequenza

chicken  
turkey  
baboon  
human  
Manis\_pentadactyla  
Capra\_hircus\_breed\_qianbei\_ma\_goat  
Capra\_hircus\_breed\_laoshan  
Capra\_hircus\_breed\_guizhou\_white  
Capra\_hircus\_1  
Capra\_hircus\_3  
Budorcas\_taxicolor  
Oryx\_dammah  
Hemitragus\_jemlahicus  
Capra\_ibex  
Ovis\_aries  
Ovis\_aries\_4  
Capra\_hircus\_breed\_black\_bengal  
Capra\_hircus\_breed\_sirohi  
Capra\_hircus\_2  
Ovis\_aries\_3  
ovine  
Ovis\_aries\_1  
Ovis\_aries\_6  
Ovis\_aries\_5  
Sylvicapra\_grimmissa  
aepyceros\_melampus  
Bos\_taurus  
Bos\_taurus\_3  
Bos\_taurus\_4  
Bos\_taurus\_1  
bovine\_wild  
bovine\_piedmontese  
Bison\_bison  
Bos\_taurus\_2  
Bos\_javanicus  
bovine\_belgian  
Bos\_mutus  
Bos\_taurus\_5

# Miostatina: allineamento multi-sequenza

Bison\_bison  
Bos\_taurus\_2  
Bos\_javanicus  
**bovine\_belgian**  
Bos\_mutus  
Bos\_taurus\_5  
Bos\_indicus\_x\_Bos\_taurus  
Bos\_indicus  
Bubalus\_carabanensis  
Bubalus\_bubalis\_1  
Bubalus\_bubalis  
Taurotragus\_derbianus  
Moschus\_berezovskii  
Cervus\_canadensis  
Cervus\_elaphus\_xanthopygus  
Cervus\_elaphus  
Cervus\_elaphus  
Dama\_dama  
Odocoileus\_virginianus  
antilocapra\_americana  
Physeter\_catodon  
Kogia\_breviceps  
Balaenoptera\_acutorostrata  
Balaenoptera\_ricei  
Balaenoptera\_musculus  
Eubalaena\_glacialis  
Delphinapterus\_leucas  
Monodon\_monoceros  
Phocoena\_sinus  
Lagenorhynchus\_obliquidens  
Tursiops\_truncatus  
Lagenorhynchus\_albirostris  
Orcinus\_orca  
Globicephala\_melas  
Neophocaena\_asiaeorientalis\_asiaeorientalis  
Delphinus\_delphis  
Mesoplodon\_densirostris  
Hippopotamus\_amphibius\_kiboko  
Camelus\_ferus  
Camelus\_bactrianus  
Camelus\_dromedarius

# Miostatina: allineamento multi-sequenza

Vicugna\_pacos  
Phacochoerus\_africanus  
Sus\_scrofa  
Sus\_scrofa\_breed\_banna\_minipig  
Sus\_scrofa\_2  
Sus\_scrofa\_breed\_tibetan  
porcine  
Molossus\_molossus  
Miniopterus\_natalensis  
Pteronotus\_parnellii\_mesoamericanus  
Sus\_scrofa\_1  
Hipposideros\_armiger  
Rhinolophus\_sinicus  
Callorhinus\_ursinus  
Odobenus\_rosmarus\_divergens  
Eumetopias\_jubatus  
Zalophus\_californianus  
Mirounga\_leonina  
Mirounga\_angustirostris  
Leptonychotes\_weddellii  
Neomonachus\_schauinslandi  
Phoca\_vitulina  
Halichoerus\_grypus  
Ursus\_arctos  
Ursus\_americanus  
Ursus\_maritimus  
Lynx\_rufus  
Leopardus\_geoffroyi  
Suricata\_suricatta  
Ceratotherium\_simum\_simum  
Diceros\_bicornis\_minor  
Equus\_quagga  
Equus\_przewalskii  
Equus\_quagga\_1  
Equus\_asinus\_1  
Equus\_asinus  
Equus\_caballus  
rat  
zebrafish

ggagagatttggacttgactgtgatgaggactcaacagaatctcgatgcttcgataccctctaactgtggattttgaa  
ggagagatttggacttgactgtgatgaggactcaacagaatctcgatgcttcgataccctctaactgtggattttgaa  
ggagagatttggactcgactgtgatgaggactcaacagaatctcgatgcttcgataccctctaactgtggattttgaa  
ggagagatttggactcgactgtgatgaggactcaacagaatctcgatgcttcgataccctctaactgtggattttgaa  
ggagagatttggactcgactgtgatgaggactcaacagaatctcgatgcttcgataccctctaactgtggattttgaa  
ggagagatttggactcgactgtgatgaggactcaacagaatctcgatgcttcgataccctctaactgtggattttgaa  
ggagagatttggactcgactgtgatgaggactcaacagaatctcgatgcttcgataccctctaactgtggattttgaa  
ggagagatttggactcgactgtgatgaggactcaacagaatctcgatgcttcgataccctctaactgtggattttgaa  
gaagagatttggacttgactgtgatgaggactcaacagaatctcgatgcttcgataccctctaactgtggattttgaa  
ggagagatttggacttgactgtgatgaggactccacacagaatctcgatgcttcgataccctctaactgtggattttgaa  
ggagagatttggacttgactgtgatgaggactccacacagaatctcgatgcttcgataccctctaactgtggattttgaa  
gaagagatttggacttgactgtgatgaggactccacacagaatctcgatgcttcgataccctctaactgtggattttgaa  
gaagagatttggacttgactgtgatgaggactccacacagaatctcgatgcttcgataccctctaactgtggattttgaa  
gaagagatttggacttgactgtgatgaggactccacacagaatctcgatgcttcgataccctctaactgtggattttgaa  
gaagagatttggacttgactgtgatgaggactccacacagaatctcgatgcttcgataccctctaactgtggattttgaa  
ggagagacttggacttgactgtgatgaggactccacacagaatctcgatgcttcgataccctctaactgtggattttgaa  
ggagggactccggactggactgcgatgagaattcctcagagtctcgctgctgcaggtaccctctaactgtggacttcgag

# La razza Belga e la miostatina

```
> setwd('~/DNASEQUENCES/')
> load('myostatin.RData')
# [sostituire <myostatin.RData> con il nome del file che avete creato (es.: <piero.Rdata>)]

> library(ape)
> library(seqinr)

> myostatin.fas = read.dna("myostatin_mod.fas", format="fasta")
> myostatin.fasta = read.dna("myostatin_mod.fasta", format="fasta")

> myostatin.fas

> myostatin.fasta
```

# La razza Belga e la miostatina

> myostatin.fasta

111 DNA sequences in binary format stored in a matrix.

all sequences of same length: 6434

Labels:

chicken

turkey

baboon

human

Manis\_pentadactyla

Capra\_hircus\_breed\_qianbei\_ma\_goat

...

Base composition:

a	c	g	t
---	---	---	---

a.337	a.181	a.189	a.293
-------	-------	-------	-------

(Total: 714.17 kb)

# La razza Belga e la miostatina

```
> str(myostatin.fasta)
```

```
'DNabin' raw [1:111, 1:6434] - - - - . . .
- attr(*, "dimnames")=List of 2
..$ : chr [1:111] "chicken" "turkey" "baboon" "human" ...
..$ : NULL
```

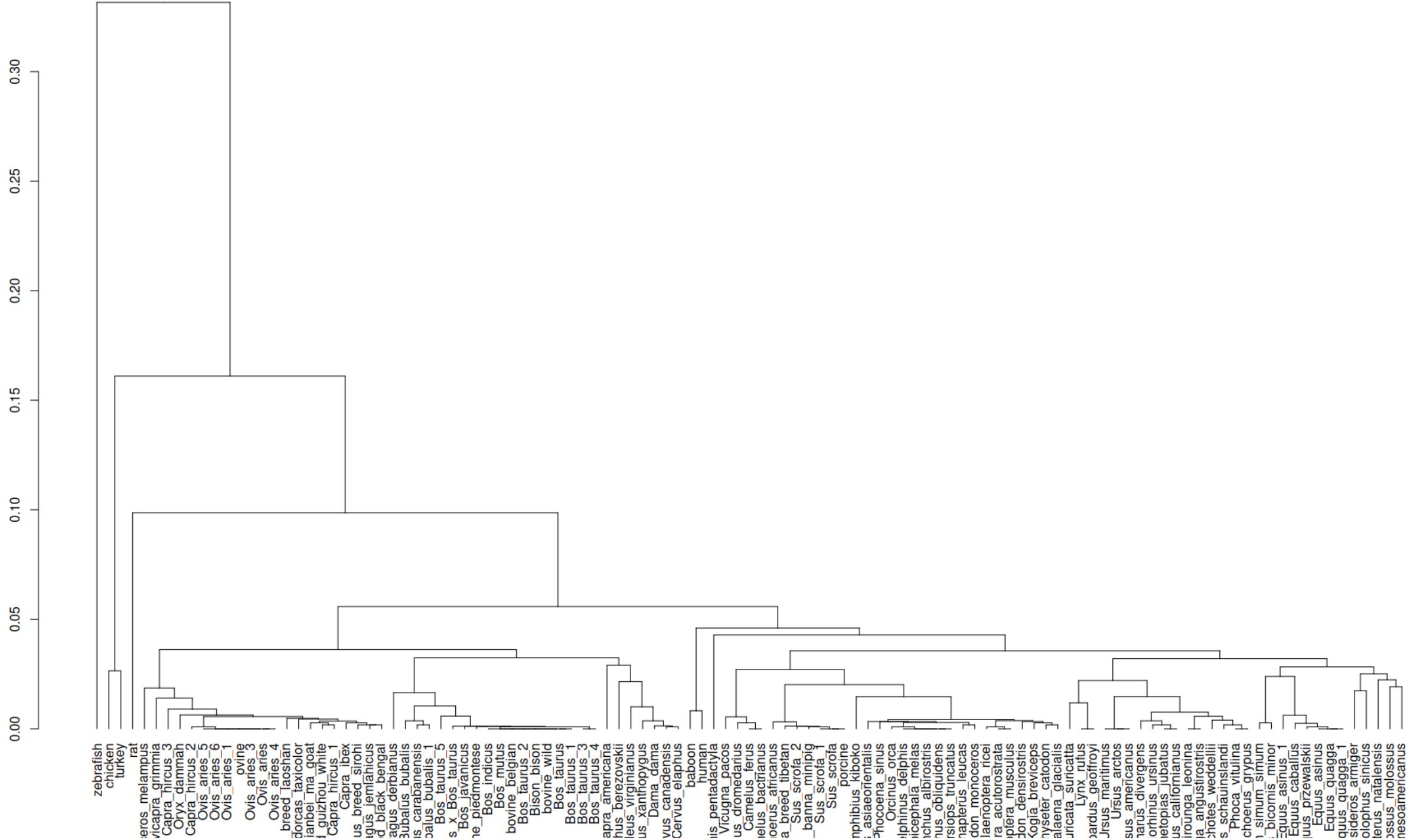
```
> myostatin.dist=dist.dna(myostatin.fasta, model="raw")
```

```
> myostatin.dist
```

turkey	chicken	turkey	baboon
baboon	a.a26484a18		
human	a.15a684932	a.143378995	
Manis_pentadactyla	a.149771689	a.142465753	a.aa8219178
Capra_hircus_breed_qianbei_ma_goat	a.1652968a4	a.153424658	a.a54794521
Capra_hircus_breed_laoshan	a.175342466	a.168a3653a	a.a76712329
Capra_hircus_breed_guizhou_white	a.1762557a8	a.169863a14	a.a78538813
Capra_hircus_1	a.1762557a8	a.168949772	a.a77625571
Capra_hircus_3	a.174429224	a.167123288	a.a75799a87
Budorcas_taxicolor	a.173515982	a.168a3653a	a.a76712329
	a.1762557a8	a.167123288	a.a739726a3

# La razza Belga e la miostatina

```
> myostatin.tree = hclust(myostatin.dist,"average")
> plot(as.dendrogram(myostatin.tree))
```



# La razza Belga e la miostatina

- Scegliere 20 specie, max 2 della stessa famiglia
- Uno dei taxa dev'essere **Bovine Belgian**, un altro **Zebra fish**
- Costruire l'albero UPGMa
- Come si distribuiscono i vari taxa sui rami dell'albero? Perchè?  
(rispondere sul quaderno)

```
> library(ape)
> setwd('~/MATERIALE_DIDATTICO_A.S._2024-2025/ESERCITAZIONI/MIOSTATINA/')
> myostatin.fasta = read.dna("myostatin_5_species_100_bp.fasta",
format="fasta")
> X = myostatin.fasta
> myostatin.dist = dist.dna(X, model="raw")
```

raw: This is simply the proportion or the number of sites that differ between each pair of sequences.

```
> library(ape)
> setwd('~/MATERIALE_DIDATTICO_A.S._2024-2025/ESERCITAZIONI/MIOSTATINA/')
> X = read.dna("myostatin_5_species_100_bp.fasta", format="fasta",
as.character = FALSE)
> str(X)
'DNAbin' raw [1:5, 1:100] a a a a ...
- attr(*, "dimnames")=List of 2
 ..$ : chr [1:5] "baboon" "bovine" "chicken" "ovine" ...
 ..$ : NULL
> X
5 DNA sequences in binary format stored in a matrix.
```

All sequences of same length: 100

Labels:  
baboon  
bovine  
chicken  
ovine  
zebra\_fish

Base composition:

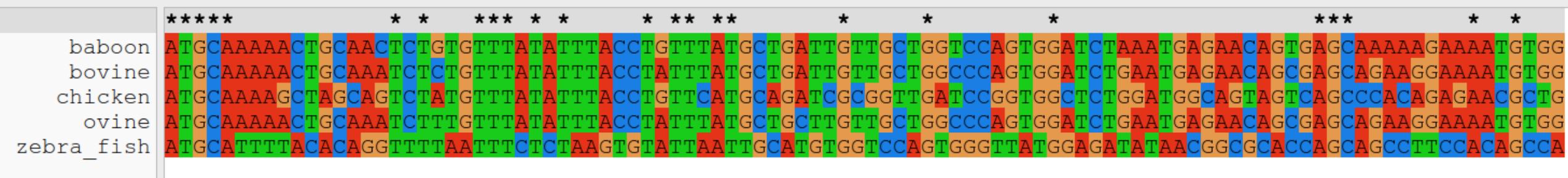
a	c	g	t
0.286	0.178	0.240	0.296
(Total: 500 bases)			

```
> myostatin.dist = dist.dna(X, model="raw")
```

```
> myostatin.dist
```

baboon bovine chicken ovine

bovine	0.08			
chicken	0.29	0.32		
ovine	0.09	0.02	0.33	
<u>zebra_fish</u>	0.67	0.64	0.70	0.63



1 2 3 4 5 6 7 8

	*****	*****	*****	*****	*****	*****	*****					
baboon	ATGCAAAA	ACTGCAAC	TCTGTGTT	TATTACCTG	TGCTGATTG	TGGTCCAG	TGGATCTAA	AATGAGAAC	AGTGA	GAGCAGA	AAAAGAAA	ATGTGG
bovine	ATGCAAAA	ACTGCAAA	TCTGTGTT	TATTACCTG	TGCTGATTG	TGGTCCAG	TGGATCTGA	ATGAGAAC	AGCAGA	GAGAAGG	AAAATGTGG	

1 234 5 6 7 8 9 1011 12 13 14 15 1617 1819 2021 22 2324 25 26 27 2829

	*****	***	*	*****	*****	*****	*****	*****	*	*****	*****	*****	*****	*****	*****	*****	*****	*****	*****	*****	*****	*****	*****
baboon	ATGCAAAA	ACTGCAAC	TCTGTGTT	TATTACCTG	TGCTGATTG	TGGTCCAG	TGGATCTAA	AATGAGAAC	AGTGA	GAGCAGA	AAAAGAAA	ATGTGG											
chicken	ATGCAAA	AGCTAGCAG	TCTATGTT	TATTACCTG	TGTCATGCAGA	TGGTCCGG	TGGCTCTGG	ATGGCAGTAG	TCA	GCCCACAG	GAGAAC	CGCTG											

1 2 3 4 5 6 7 8 9

	*****	*****	*****	*****	*****	*****	*****	*****	*****	*****	*****	*****	*****	*****	*****	*****	*****	*****	*****	*****	*****	*****
baboon	ATGCAAAA	ACTGCAAC	TCTGTGTT	TATTACCTG	TGCTGATTG	TGGTCCAG	TGGATCTAA	AATGAGAAC	AGTGA	GAGCAGA	AAAAGAAA	ATGTGG										
ovine	ATGCAAAA	ACTGCAA	ATTTGTT	TATTACCT	TGCTGCTG	TGGCTGG	CAGTGGATCT	GAATGAGAAC	AGCGAGC	GAGAAGG	AAAATGTGG											

1234567 89 10 111213 14 15 1617181920 21 22 232425262728 29 30 31 3233343536373839 4041424344 45464748 49505152535455 56575859606162 6364 656667

	*****	**	*	*	*****	*	*	*	*	*****	*	*	*	*	*	*	*	*	*****	*****	*	*
baboon	ATGCAAAA	ACTGCAAC	TCTGTGTT	TATTACCTG	TGCTGATTG	TGGTCCAG	TGGATCTAA	AATGAGAAC	AGTGA	GAGCAGA	AAAAGAAA	ATGTGG										
ra_fish	ATGCATT	TTACACAGG	TTAATTCT	TAAGTGTATT	AACTGCTA	GGTCCAGTGG	TTATGGGAT	TATAACGG	GCACCAGC	GCTTCCACAG	CCA											

1 234 5 6 7 8 9 10 1112 13 1415 16 17 18 1920 2122 2324 2526 2728 29 30 3132

	*****	**	*	*****	*****	*****	*****	*	*****	*	*****	*	*****	*	*****	*	*****	*	*****	*	*****	*	*
bovine	ATGCAAAA	ACTGCAA	ATCTGT	TATTACCT	TGCTGATT	G	TGGCTGG	CCCAGTGGATCT	GAATGAGAAC	AGCGAGC	GAGAAGG	AAAATGTGG											
chicken	ATGCAAA	AGCTAGCAG	TCTATGTT	TATTACCTG	TGTCATGCAGA	TGGTCCGG	TGGCTCTGG	ATGGCAGTAG	TCA	GCCCACAG	GAGAAC	CGCTG											

1

2

1234567 89 10 111213 14 1516171819 20 212223242526 27 28 29 3031323334353637 3839404142 43444546 4748495051 52 535455565758 5960 616263

**1** 234 5 6                      7 8 9 10 11 1213 14 1516 17 18 19 2021 2223 2425 2627 2829 30 31 3233

123456 789 10 111213 14 15 1617181920 21 22 23242526272829 3031323334 3536373839404142 4344454647484950515253545556575859 606162 63646566 6768 6970

	*****	*	**	*	***	*	*	*	***	*	*	*	*	*	*****	*	*	**
chicken	ATGCAAAAGCTAGCAGTCTATGTTATATTACCTGTTCATGCAGATCGCGGTTGATCCGGTGGCTCTGGATGGCAGTAGTCAGCCCACAGAGAACGCTG																	
ra_fish	ATGCATTACACAGGTTAATTCTCTAAGTGTATTAAATTGCATGTGGTCCAGTGGGTTATGGAGATAAACGGCGACCAGCAGCCTCACAGCCA																	

1234567 89 10 111213 14 15 1617181920 21 222324252627 28 29 30 3132333435363738 3940414243 44454647 4849505152 53 545556575859 6061 626364

The phylogenetic tree illustrates the evolutionary relationship between ovine and various teleost fish. The tree is rooted at the top and branches downwards. Ovine is shown in blue, and teleost fish are shown in green. The tree structure shows that ovine is most closely related to the teleost fish group, which includes species from different orders like Cyprinodontiformes, Siluriformes, and Perciformes.

```
> myostatin.UPGMA = hclust(myostatin.dist, "average")
# simple UPGMA
```

```
> myostatin.UPGMA
```

Call:

```
hclust(d = myostatin.dist, method = "average")
```

```
Cluster method      : average
Distance          : raw
Number of objects: 5
```

La funzione hclust effettua una *cluster analysis* gerarchica utilizzando, in questo caso, l'algoritmo **UPGMA** (*Unweighted Pair Group Method with Arithmetic Mean*).

L'idea alla base della *cluster analysis* gerarchica è di rappresentare graficamente quali elementi (singoli esemplari di una stessa o di specie diverse, sequenze di DNA ecc.) di un certo campione sono più simili tra di loro, raggruppandoli a due a due sullo stesso ramo dell'albero.

Gli oggetti che sono più diversi fra loro vengono collocati in rami differenti.

```
> myostatin.dist
```

baboon bovine chicken ovine

	bovine	chicken	ovine	zebra_fish
bovine	0.08			
chicken	0.29	0.32		
ovine	0.09	0.02	0.33	
zebra_fish	0.67	0.64	0.70	0.63

Matrice **D1** di distanze genetiche  
(n° di nucleotidi differenti su 100)  
fra coppie di specie

Il bovino e la pecora sono le specie  
tra le quali la D1 è <, quindi le  
mettiamo insieme

Quindi calcoliamo la distanza media  
tra bov-ovi e ciascuna delle altre  
specie

$$D2((bov,ovi),bab) = [D1(bov,bab)+D1(ovi,bab)]/2 = (8+9)/2 = 8,5$$

$$D2((bov,ovi),gal) = [D1(bov,gal)+D1(ovi,gal)]/2 = (32+33)/2 = 32,5$$

$$D2((bov,ovi),zeb) = [D1(bov,zeb)+D1(ovi,zeb)]/2 = (64+63)/2 = 63,5$$

	bab	bov	gal	ovi	zeb
bab	0	8	29	9	67
bov	8	0	32	2	64
gal	29	32	0	33	70
ovi	9	2	33	0	63
zeb	67	64	70	63	0

	<b>bab</b>	<b>bov-ovi</b>	<b>gal</b>	<b>zeb</b>
<b>bab</b>	0	8,5	29	67
<b>bov-ovi</b>	<b>8,5</b>	0	32,5	63,5
<b>gal</b>	29	32,5	0	70
<b>zeb</b>	67	63,5	70	0

	<b>(bov-ovi)bab</b>	<b>gal</b>	<b>zeb</b>
<b>(bov-ovi)bab</b>	0	30,7 5	65,25
<b>gal</b>	30,75	0	70
<b>zeb</b>	65,25	70	0

$$D3(((bov,ovi),bab),gal) = [D2((bov,ovi),gal)+(D2(bab,gal)]/2 = (32,5+29)/2 = 30,75$$

$$D3(((bov,ovi),bab),zeb) = [D2((bov,ovi),zeb)+(D2(bab,zeb)]/2 = (63,5+67)/2 = 65,25$$

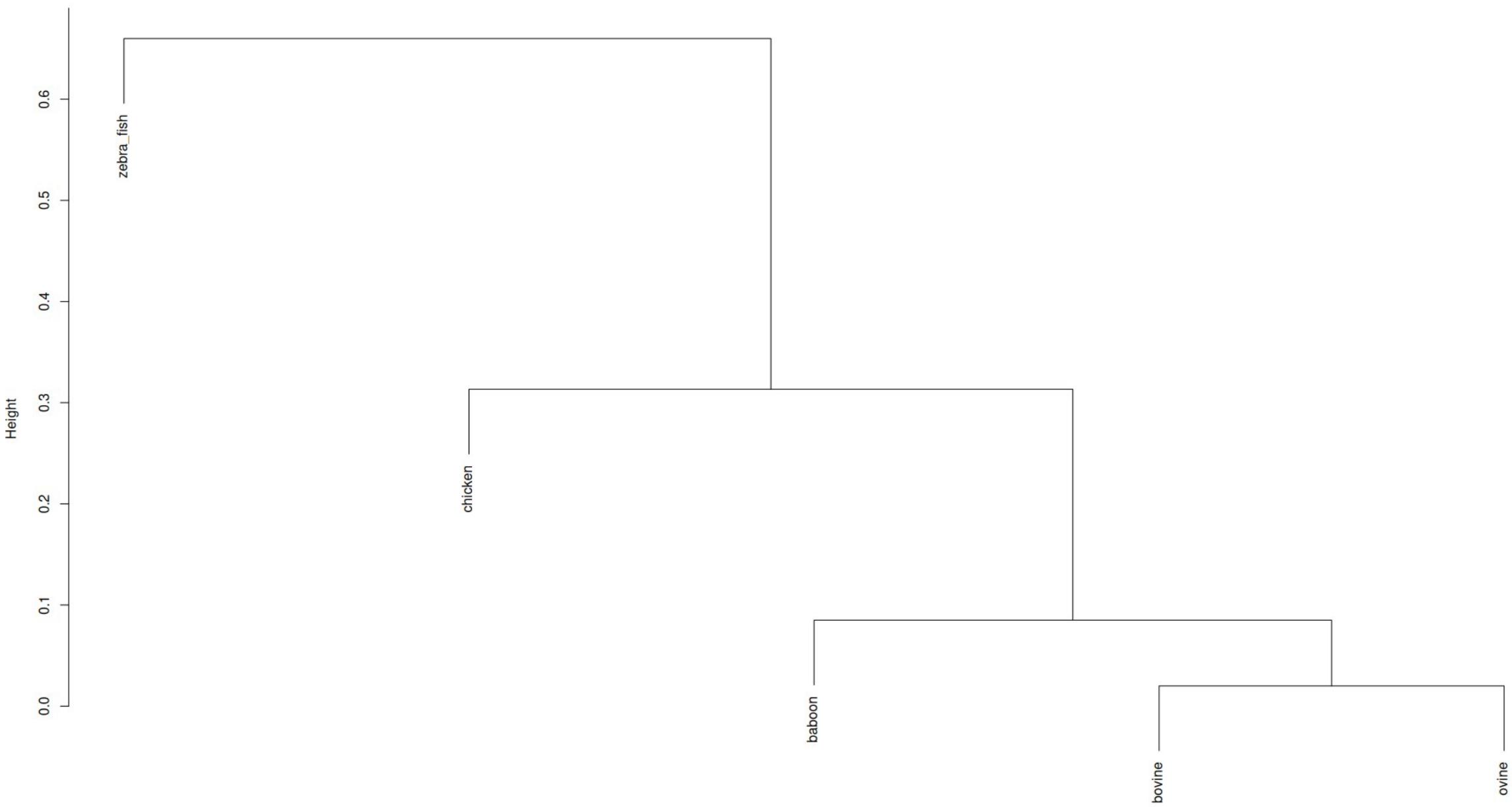
	<b>(bov-ovi)bab</b>	<b>gal</b>	<b>zeb</b>
<b>(bov-ovi)bab</b>	0	30,7 5	65,25
<b>gal</b>	30,75	0	70
<b>zeb</b>	65,25	70	0

$$D4(((bov,ovi),bab),gal),zeb) = [D3((bov,ovi),bab),zeb)+D3(gal,zeb)]/2 = (65,25+70)/2 = 67,625$$

	<b>((bov-ovi)bab)ga</b>	<b>zeb</b>
<b>((bov-ovi)bab)gal</b>	0	67,625
<b>zeb</b>	67,625	0

```
> plot(myostatin.UPGMA, main = 'UPGMA distances')
```

UPGMA distances

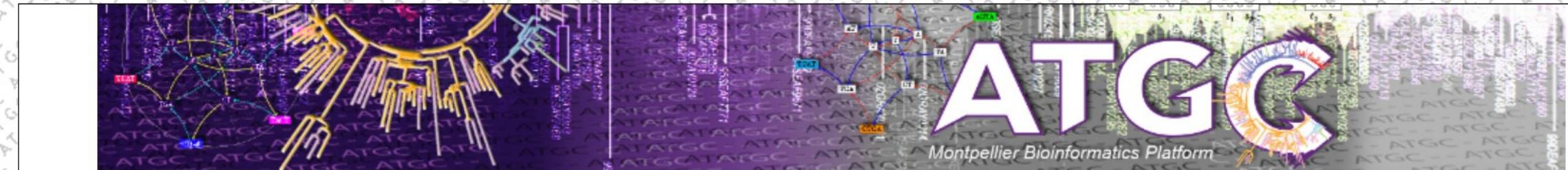


myostatin.dist  
hclust (\*, "average")

# Scelta del modello di sostituzione nucleotidica

L'analisi filogenetica più semplice si basa sul calcolo delle distanze fra coppie di sequenze nucleotidiche, le quali diventano gli ingressi di una matrice che, tramite algoritmi di clusterizzazione, viene convertita in un albero filogenetico.  
La distanza fra due sequenze è definita come numero atteso di sostituzioni nucleotidiche per sito (generalmente < 1) e può essere espressa come semplice proporzione dei siti variabili o distanza p.

```
> myostatin.dist  
            baboon bovine chicken ovine  
bovine          0.08  
chicken        0.29    0.32  
ovine          0.09    0.02    0.33  
zebra_fish     0.67    0.64    0.70   0.63
```



Home

Organization

Citations & Statistics

Partners

Online programs

PhyML

Benchmarks

Datasets

Forum & FAQ

Online execution

Source code on GitHub

Papers & contacts

PhyML versions

User guide

Binaries

Databases

Datasets

NGS



## PhyML 3.0: new algorithms, methods and utilities

Please cite:

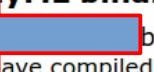
**"New Algorithms and Methods to Estimate Maximum-Likelihood Phylogenies: Assessing the Performance of PhyML 3.0."**

**Guindon S., Dufayard J.F., Lefort V., Anisimova M., Hordijk W., Gascuel O.**

Systematic Biology, 59(3):307-21, 2010.

Get the latest PhyML releases from [GitHub](#).

### Download PhyML binaries

Click [here](#) to  binaries

The binaries you have compiled on various Mac, Windows or LINUX systems are much welcome ([Stéphane Guindon](#)) and will be displayed on this web page.



Contact: [Webmaster](#), LIRMM.



LIRMM

## Il formato PHYLIP interleaved

```
5 100
baboon
atgaaaaactgcaactctgtttatattacctgttatgctgattgtgctggccagtggatctaaatgagaacagttagaaaaaaaatgtgg
bovine
atgaaaaactgcaaactctgtttatattacctatattatgctgattgtgctggcccagtggatctgaatgagaacagcgagcagaaggaaaatgtgg
chicken
atgaaaaagcttagcagtctatgttatattacctgttcatgcagatcgccgttgcgtccgtggatggcagtagtcagcccacagagaacgctg
ovine
atgaaaaactgcaaactttgttatattacctatattatgctgctgtgctggcccagtggatctgaatgagaacagcgagcagaaggaaaatgtgg
zebrafish
atgcatttacacaggttaattctctaagtgttattgcattgtggccagtggatggagatataacggcgcaccagcagcctccacagcca

> substitution_model_test = phymltest('myostatin_5_species_100_bp.phy', format
= "interleaved", itree = NULL, exclude = NULL, execname = '/home/piero/PhyML-
3.1/PhyML-3.1_linux64', append = TRUE)
```

# L'output del test

> `substitution_model_test`

	nb.free.para	loglik	AIC
JC69	1	-414.3124	830.6248
JC69+I	2	-414.1910	832.3821
JC69+G	2	-413.8407	831.6814
JC69+I+G	3	-413.8313	833.6625
K80	2	-411.0496	826.0993
K80+I	3	-408.3180	822.6359
K80+G	3	-408.8849	823.7699
K80+I+G	4	-407.9957	823.9915
F81	4	-414.1411	836.2823
F81+I	5	-414.0996	838.1992
F81+G	5	-413.7568	837.5136
F81+I+G	6	-413.7486	839.4972
F84	5	-409.8965	829.7931
F84+I	6	-407.9673	827.9346
F84+G	6	-408.0924	828.1848
F84+I+G	7	-407.5650	829.1301
HKY85	5	-409.8806	829.7612
HKY85+I	6	-407.9661	827.9323
HKY85+G	6	-408.0732	828.1464
HKY85+I+G	7	-407.5730	829.1460
TN93	6	-409.8804	831.7607
TN93+I	7	-407.9637	829.9274
TN93+G	7	-408.0700	830.1400
TN93+I+G	8	-407.5648	831.1297
GTR	9	-402.4319	822.8638
GTR+I	10	-401.2392	822.4784
GTR+G	10	-400.2603	820.5205
<b>GTR+I+G</b>	<b>11</b>	<b>-400.2379</b>	<b>822.4758</b>

>

## II file <myostatin\_5\_species\_100\_bp.phy\_phyml\_tree.txt>

(bovine:0.00000202, (ovine:0.00002963, zebrafish:9.99805710) 0.000000:0.02049504,  
(baboon:0.00008628, chicken:0.50886609) 0.842000:0.09067509);  
(baboon:0.01048862, (bovine:0.00000001, (zebrafish:1.37441488, ovine:0.00000508):0.02026739):0.07447686, chicken:0.35829670);  
(baboon:0.00972686, chicken:0.36497752, (bovine:0.00000118, (ovine:0.00000029, zebrafish:1.54101437):0.02030498):0.07560156);  
(baboon:0.00007584, chicken:0.44219573, (ovine:0.02056245, (bovine:0.00000001, zebrafish:3.54999086):0.00000001):0.08826723);  
(baboon:0.00000910, chicken:0.45531080, (bovine:0.00000001, (zebrafish:4.16221975, ovine:0.00001874):0.02059383):0.08887950);  
(bovine:0.00000099, (ovine:0.02028732, zebrafish:3.09237767):0.00003742, (baboon:0.01122146, chicken:0.36941408):0.07398107);  
(zebrafish:8.82019763, (bovine:0.00000175, ovine:0.02039573):0.02277632, (chicken:0.39408351, baboon:0.01090682):0.05227845);  
(bovine:0.00000099, (ovine:0.02044652, zebrafish:10.00000000):0.00002146, (baboon:0.00002629, chicken:0.50356626):0.08929782);  
(chicken:0.43223177, baboon:0.00713230, (bovine:0.00000126, (zebrafish:8.89223736, ovine:0.01973263):0.00074093):0.08047846);  
(baboon:0.01115834, chicken:0.36143405, (bovine:0.00000181, (zebrafish:1.42016394, ovine:0.00000268):0.02033915):0.07448850);  
(bovine:0.00000091, (chicken:0.36579093, baboon:0.01074355):0.07520598, (zebrafish:1.52463974, ovine:0.00000001):0.02037077);  
(bovine:0.00000117, (zebrafish:3.58786252, (chicken:0.44228493, baboon:0.00171289):0.08712725):0.00000872, ovine:0.02063859);  
(baboon:0.00000824, chicken:0.46731497, (bovine:0.00000001, (zebrafish:4.82711651, ovine:0.00000756):0.02070614):0.08999625);  
(bovine:0.00000101, (ovine:0.02042293, zebrafish:3.77229889):0.00003150, (baboon:0.01338516, chicken:0.37881952):0.07317730);  
(zebrafish:8.98214792, (bovine:0.00000182, ovine:0.02045418):0.02675688, (chicken:0.39647730, baboon:0.01296603):0.04725382);  
(bovine:0.00000204, (ovine:0.00002615, zebrafish:9.99812034):0.02054687, (baboon:0.00002661, chicken:0.50940308):0.09100487);  
(chicken:0.44286719, zebrafish:9.08352240, (baboon:0.00942811, (bovine:0.00000132, ovine:0.02057252):0.07969971):0.00014779);  
(bovine:0.00000099, (ovine:0.02039898, zebrafish:3.70535167):0.00003159, (baboon:0.01300483, chicken:0.37844433):0.07338886);  
(zebrafish:8.98429384, (bovine:0.00000180, ovine:0.02042441):0.02673440, (chicken:0.39638088, baboon:0.01260347):0.04745109);  
(bovine:0.00000202, (ovine:0.00002963, zebrafish:9.99805710):0.02049504, (baboon:0.00008628, chicken:0.50886609):0.09067509);  
(chicken:0.44246753, zebrafish:9.08528623, (baboon:0.00898618, (bovine:0.00000130, ovine:0.02053515):0.07990025):0.00016885);  
(bovine:0.00000099, (ovine:0.02039408, zebrafish:3.68224905):0.00003166, (baboon:0.01293915, chicken:0.37829051):0.07342438);  
(zebrafish:8.98269671, (bovine:0.00000180, ovine:0.02044070):0.03061041, (chicken:0.39640554, baboon:0.01278578):0.04346390);  
(bovine:0.00000205, (ovine:0.00003700, zebrafish:9.99683005):0.02045292, (baboon:0.00008704, chicken:0.50879952):0.09053445);  
(chicken:0.44280285, zebrafish:9.08381557, (baboon:0.00936629, (bovine:0.00000130, ovine:0.02056699):0.07972586):0.00014726);  
(bovine:0.00000001, (zebrafish:4.05071318, (baboon:0.00593667, chicken:0.39542302):0.08161689):0.00000004, ovine:0.02053827);  
(zebrafish:4.34749500, (bovine:0.00000057, (chicken:0.40630281, baboon:0.00316606):0.08445039):0.02051856, ovine:0.00000251);  
(zebrafish:10.00000000, (bovine:0.00000056, (baboon:0.00001018, chicken:0.54408882):0.09498755):0.02116598, ovine:0.00001086);  
**(baboon:0.0000993, chicken:0.53213267, (bovine:0.00000001, (zebrafish:9.76620951, ovine:0.00000001):0.02114857):0.09443561);**

# Scelta del modello di sostituzione nucleotidica

```
> r = '(baboon:0.00000993,chicken:0.53213267,(bovine:0.0000001,  
(zebrafish:9.76620951,ovine:0.0000001):0.02114857):0.09443561)';  
> albero = read.tree(text = r)  
> albero
```

Phylogenetic tree with 5 tips and 3 internal nodes.

Tip labels:

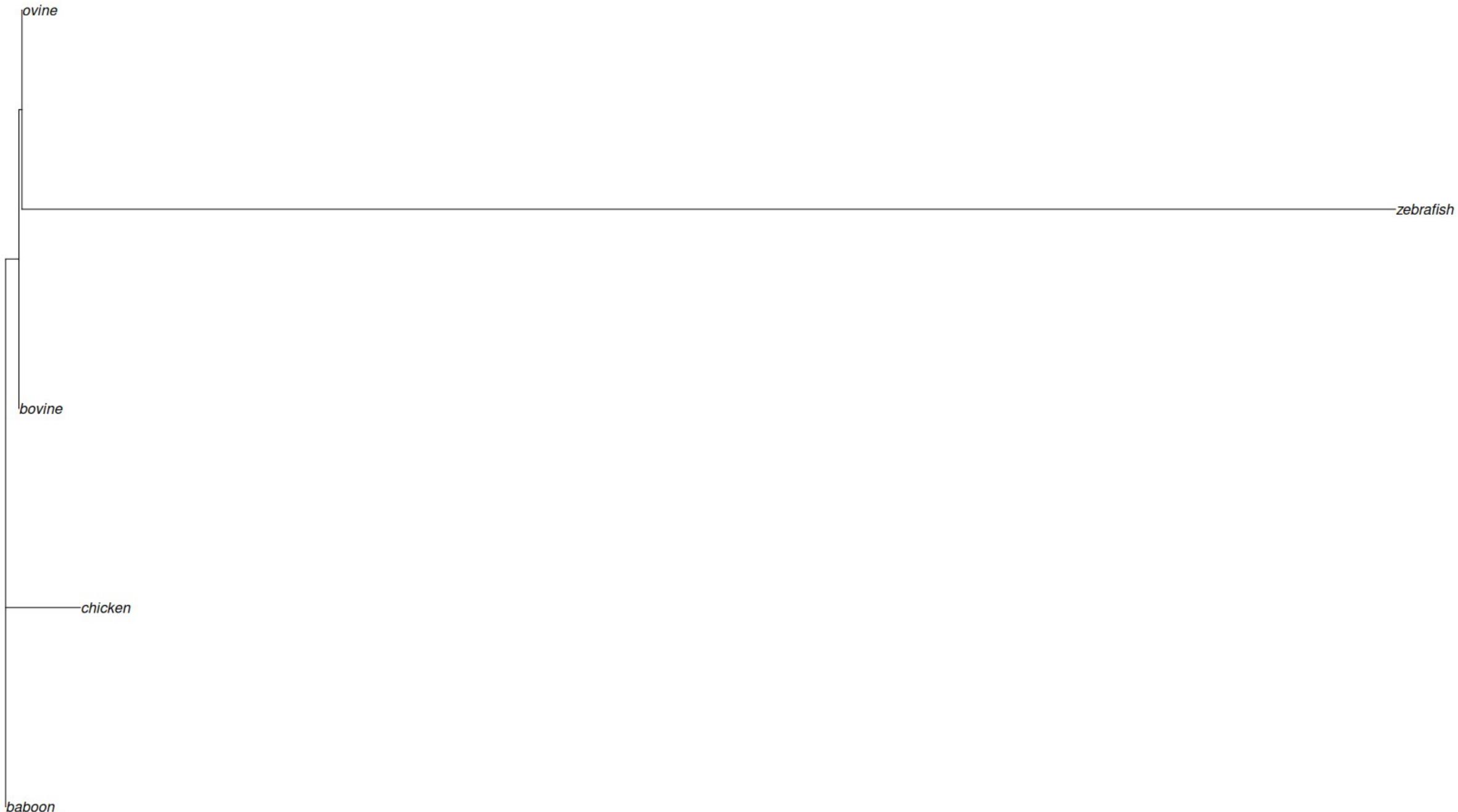
baboon, chicken, bovine, zebrafish, ovine

Unrooted; includes branch lengths.

```
> plot(albero, main = 'GTR model')
```

# L'albero più verosimile

GTR model



```
> alb = read.tree(text = "((bovino:1, ovino:1):1, babbuino:1);")  
> alb
```

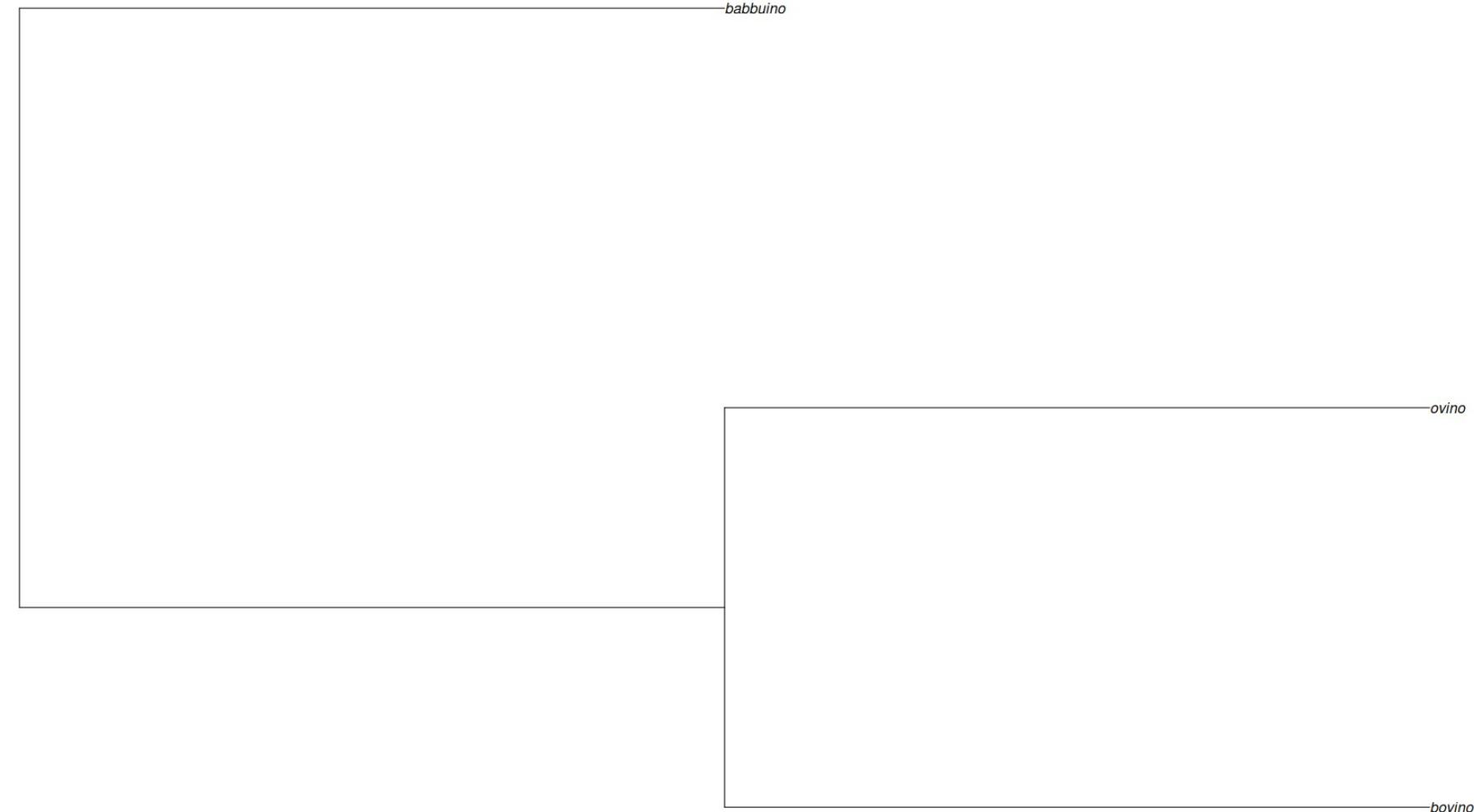
Phylogenetic tree with 3 tips and 2 internal nodes.

Tip labels:

bovino, ovino, babbuino

Rooted; includes branch lengths.

```
> plot(alb)
```



```
> alb = read.tree(text = "((bovino:1, ovino:1):1, babbuino:2);")  
> alb
```

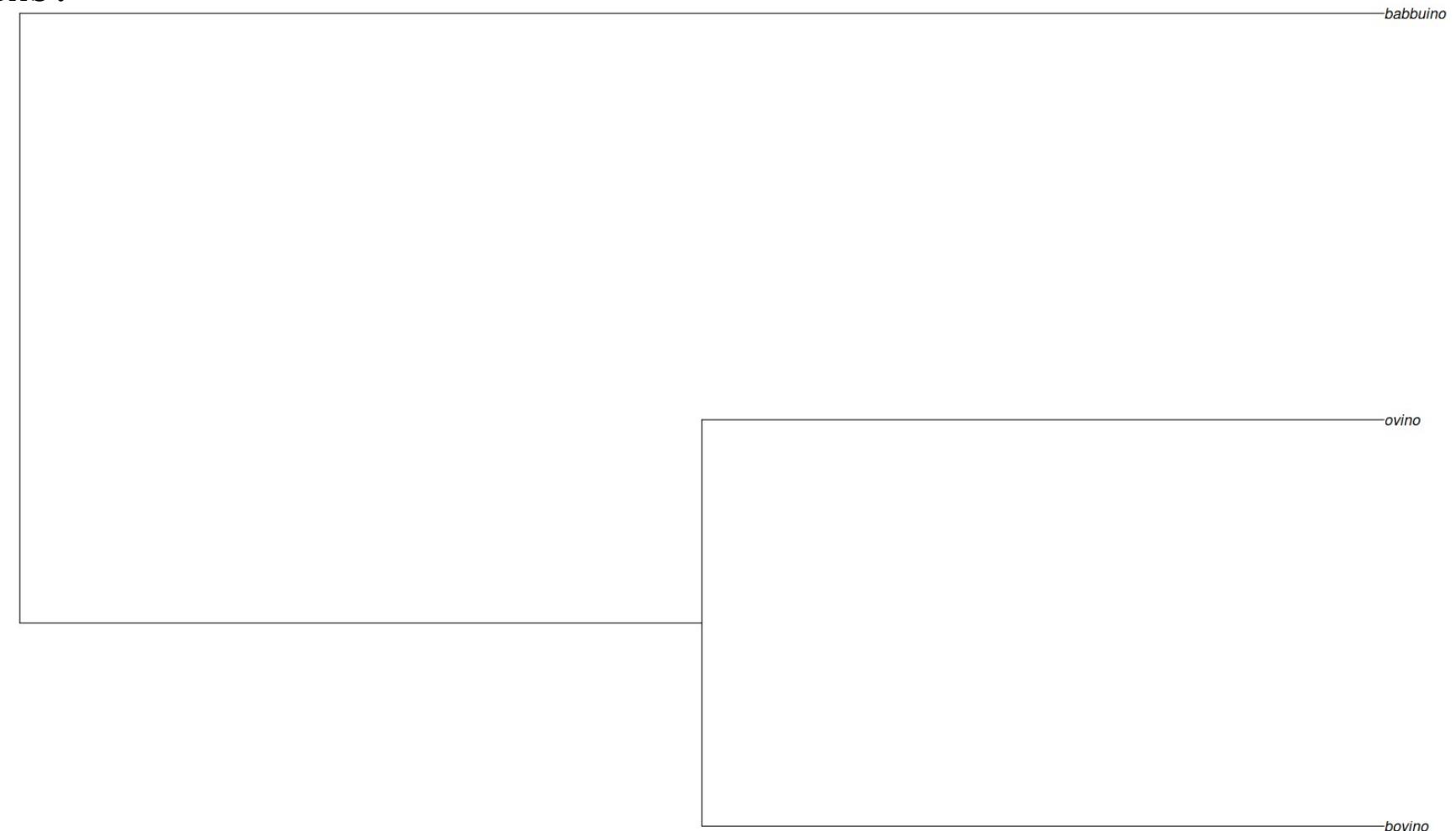
Phylogenetic tree with 3 tips and 2 internal nodes.

Tip labels:

bovino, ovino, babbuino

Rooted; includes branch lengths.

```
> plot(alb)
```



```
> alb = read.tree(text = "(((bovino:1, ovino:1):1, babbuino:2):2, gallo:3);")  
> alb
```

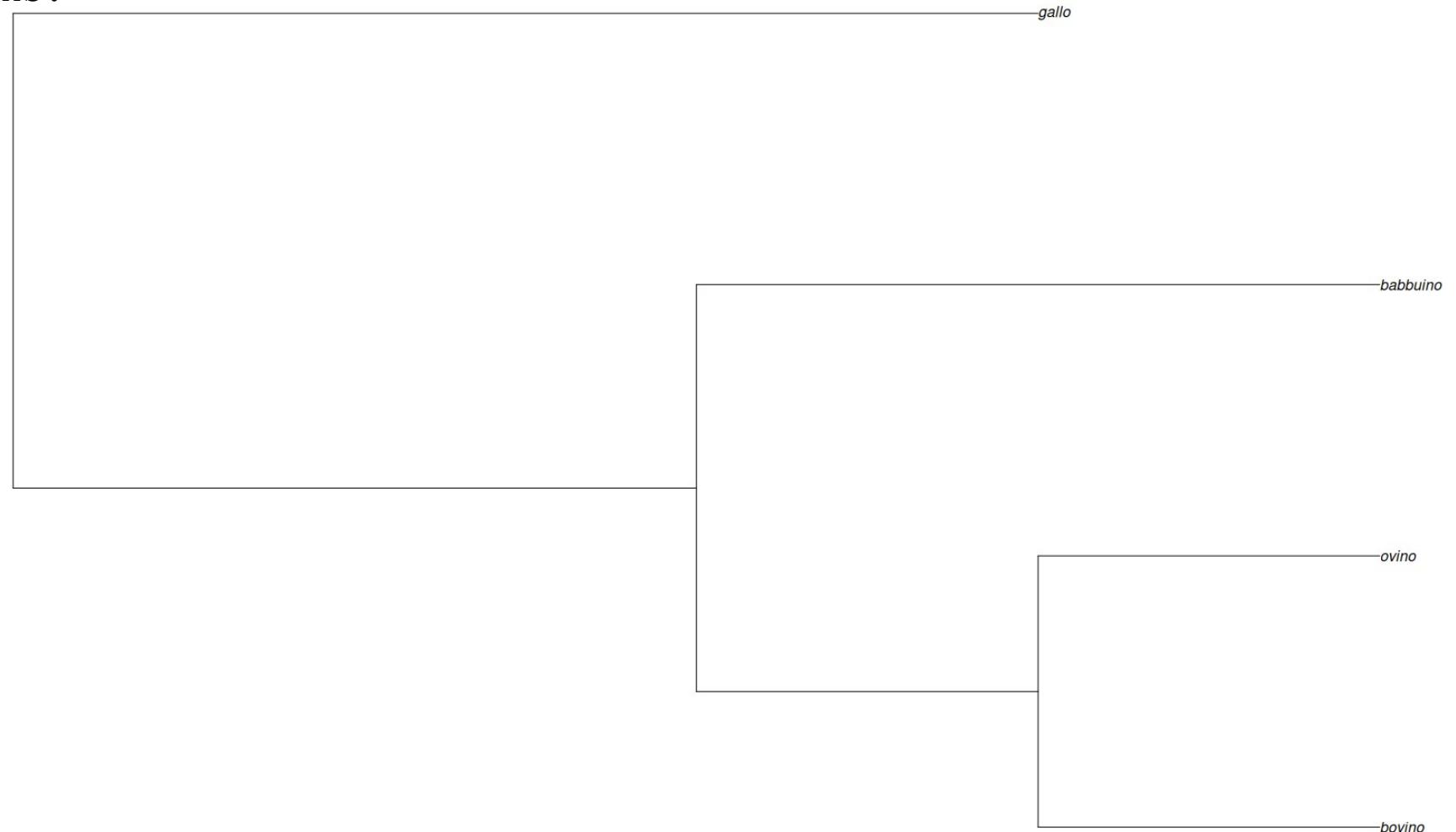
Phylogenetic tree with 4 tips and 3 internal nodes.

Tip labels:

bovino, ovino, babbuino, gallo

Rooted; includes branch lengths.

```
> plot(alb)
```



```
> alb = read.tree(text = "(((bovino:1, ovino:1):1, babbuino:2):2, gallo:4);")  
> alb
```

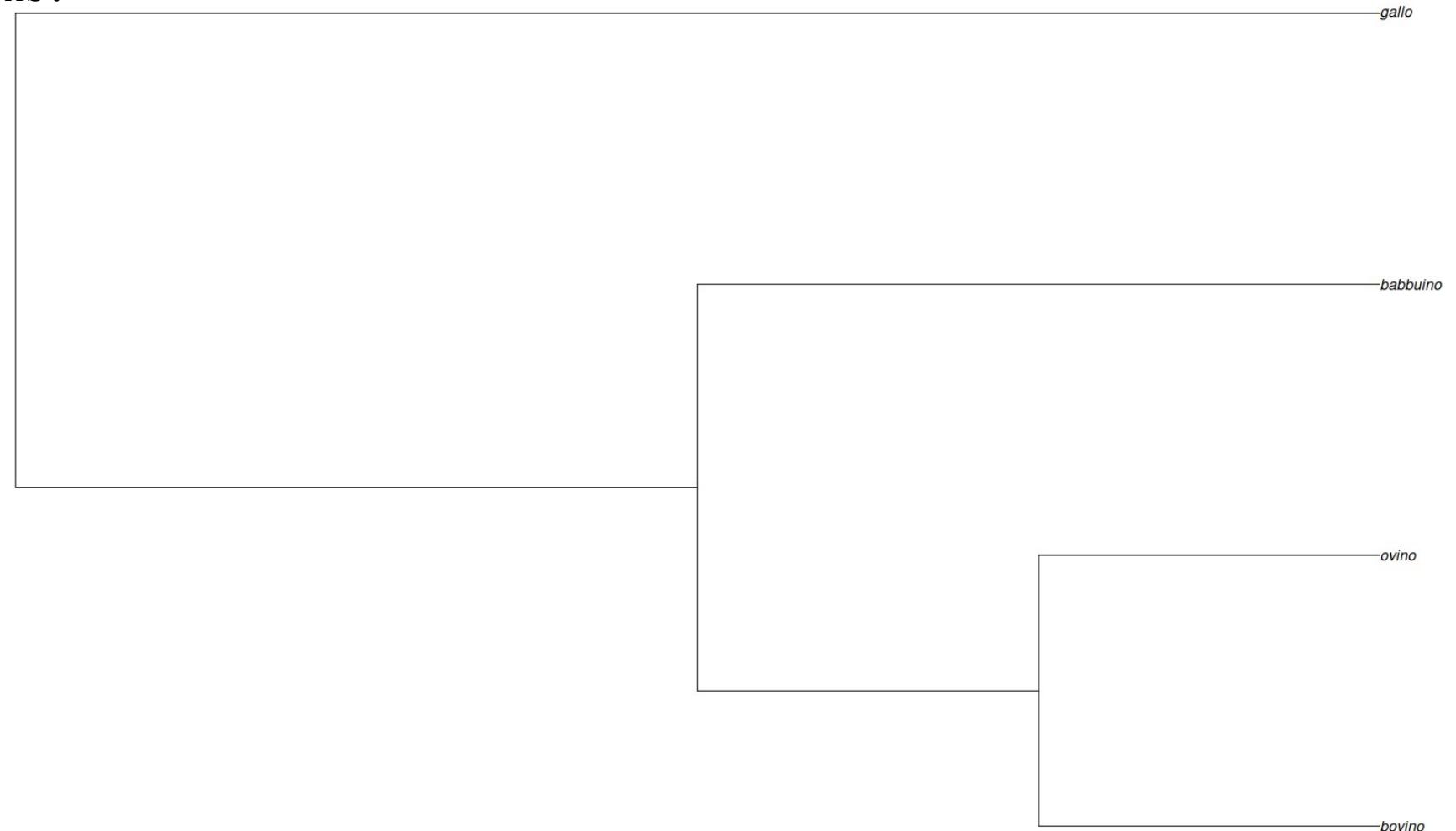
Phylogenetic tree with 4 tips and 3 internal nodes.

Tip labels:

bovino, ovino, babbuino, gallo

Rooted; includes branch lengths.

```
> plot(alb)
```



```
> alb = read.tree(text = "(((bovino:1, ovino:1):1, babbuino:2):2, gallo:4):3, pesce_zebra:7;")  
> alb
```

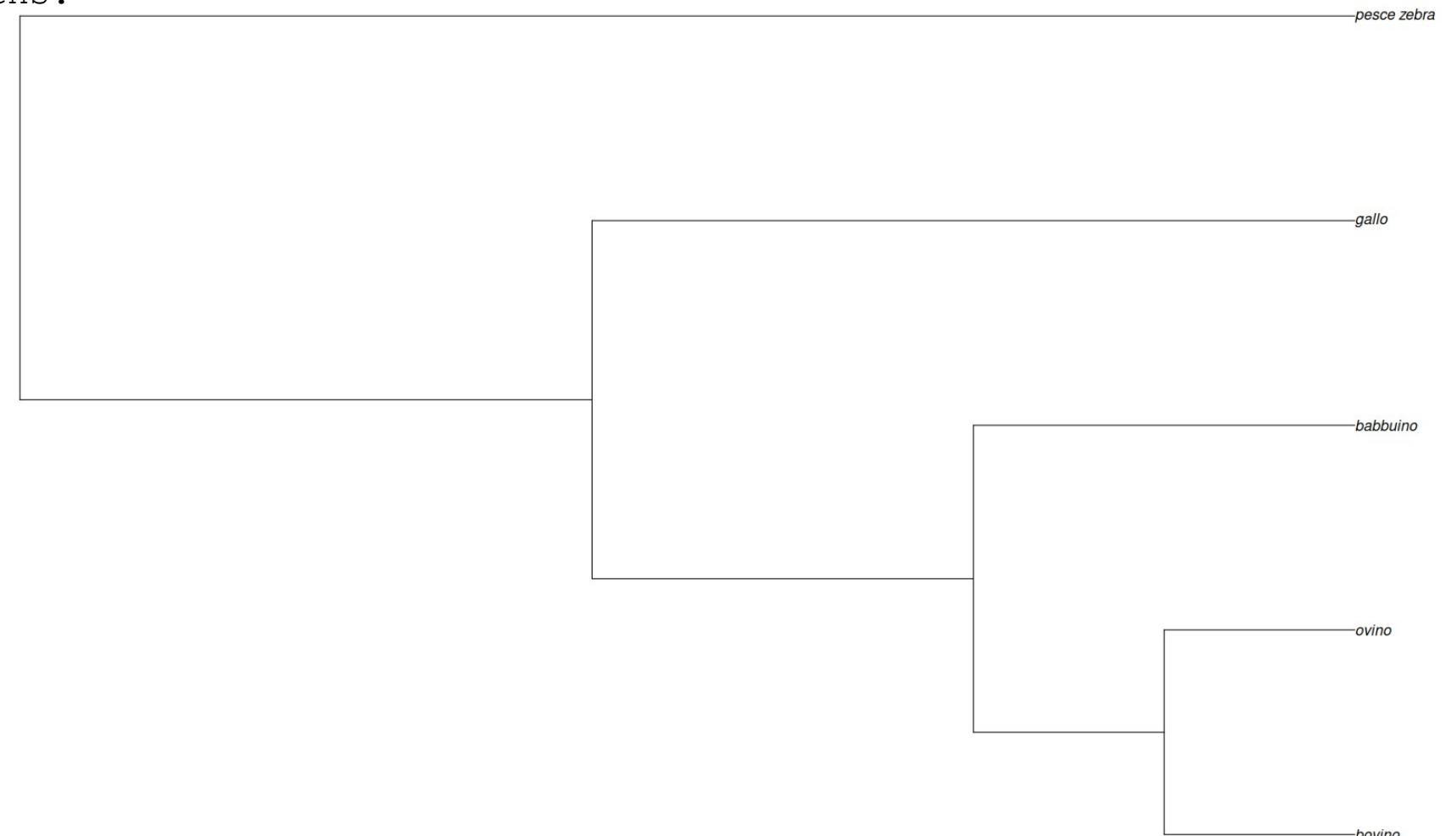
Phylogenetic tree with 5 tips and 4 internal nodes.

Tip labels:

bovino, ovino, babbuino, gallo, pesce\_zebra

Rooted; includes branch lengths.

```
> plot(alb)
```



# STATISTICA

## L'effetto additivo dei fattori genetici ed ambientali

gene normale A      ->      effetto fenotipico = 8      p = 0,5

allele mutante A'      ->      effetto fenotipico = 4      q = 0,5

omozigoti AA      ->      valore fenotipico = 16      F = 0,25

omozigoti A'A'      ->      valore fenotipico = 8      F = 0,25

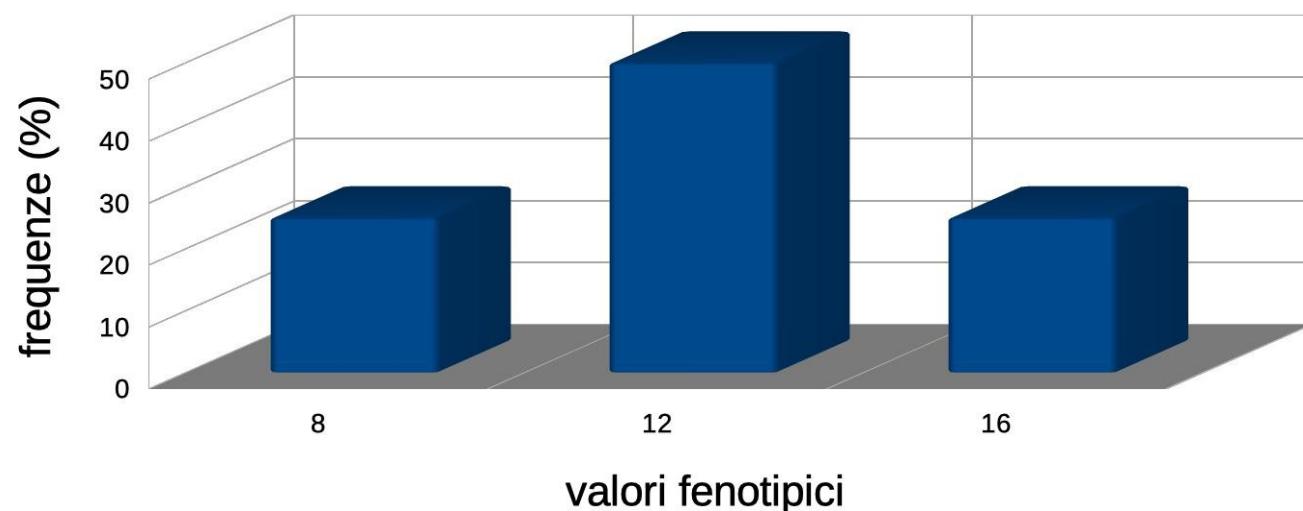
eterozigoti AA'      ->      valore fenotipico = 12      F = 0,25

eterozigoti A'A      ->      valore fenotipico = 12      F = 0,25

# L'effetto additivo dei fattori genetici ed ambientali

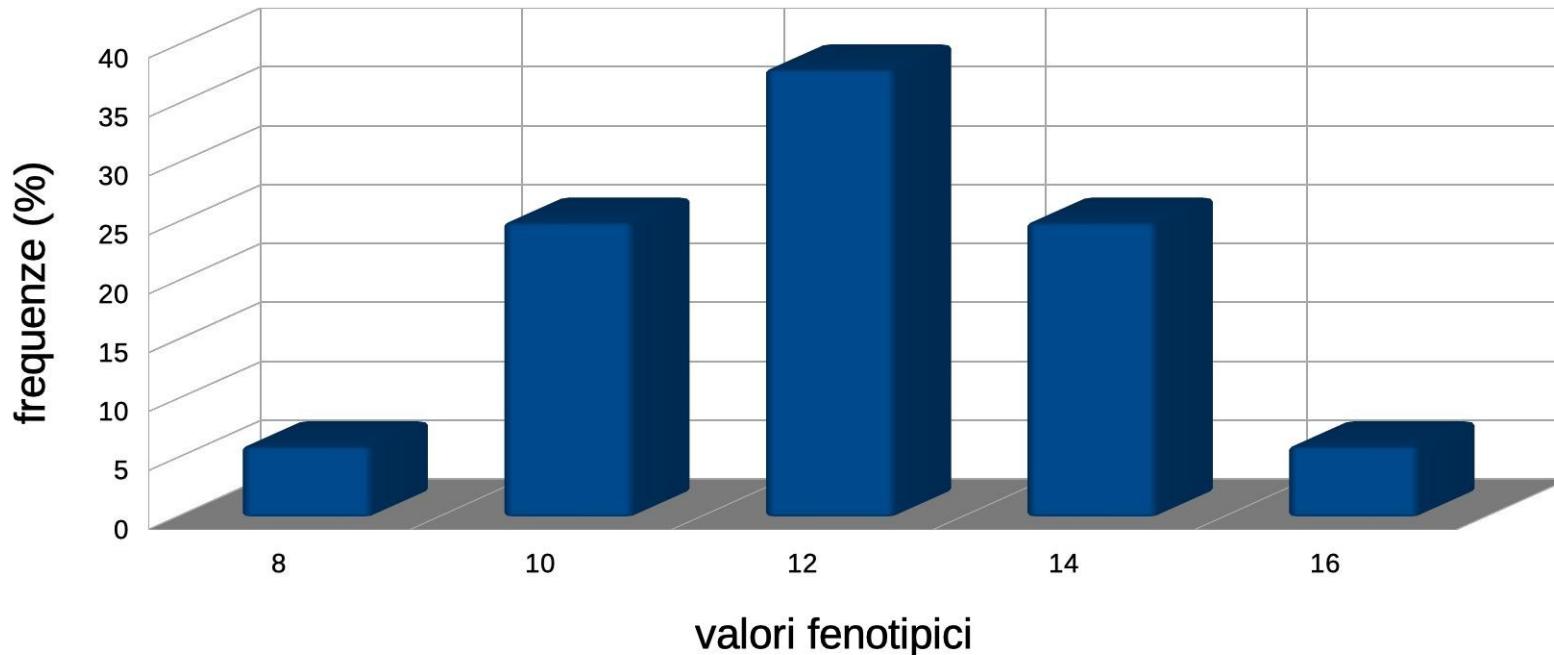
CLASSE GENOTIPICA	VALORE FENOTIPICO	FREQUENZA
AA	16	1/4 = 25%
AA' - A'A	12	2/4 = 50%
A'A'	8	1/4 = 25%

## DISTRIBUZIONE DEI VALORI FENOTIPICI CON 1 COPPIA ALLELICA



# Due loci

## DISTRIBUZIONE DEI VALORI FENOTIPICI CON 2 COPPIE ALLELICHE



Valori e frequenze dei possibili genotipi nel caso in cui un carattere quantitativo è controllato da due loci («a» e «B»)

## Tre loci

aumentando ulteriormente il numero di coppie alleliche interessate, aumenta in numero delle classi di valori ma diminuiscono le differenze fra i valori delle varie classi

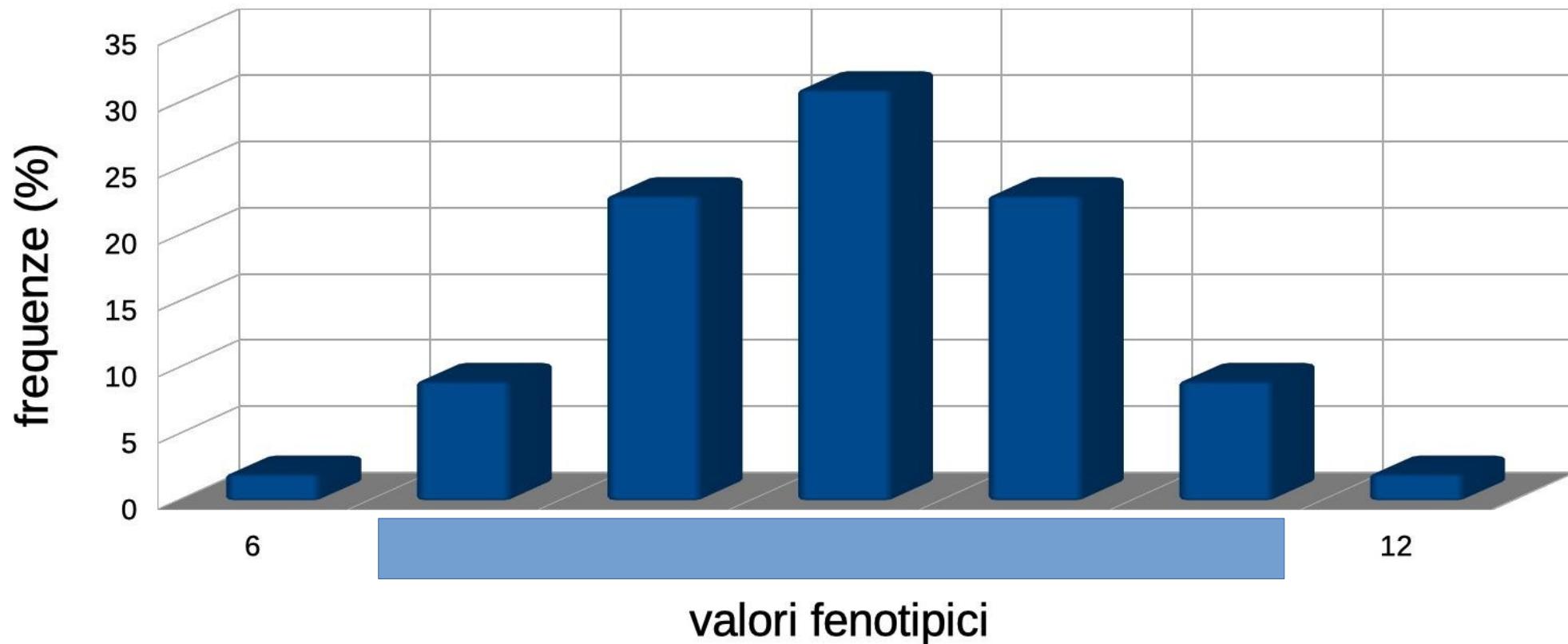
Se si considerano tre coppie di geni, a-a', B-B' e C-C', dove gli alleli «normali» a, B e C hanno effetto 2 e gli alleli «mutanti» a', B' e C' effetto 1, le classi genotipiche e i relativi valori fenotipici saranno:

# Tre loci

Quali sono i valori fenotipici intermedi?

CLaSSE GENOTIPICa	VaLORE FENOTIPICO	FREQUENZA
aaBBCC	12	1/64 = 2%
aaBBCC' – aaBBC'C – aaBB'CC aaB'BCC – aa'BBCC – a'aBBCC		6/64 = 9%
aaBBC'C' – aaBB'CC' – aaB'BCC' aa'BBCC' – a'aBBCC' – aaBB'C'C aaB'BC'C – aa'BBC'C – a'aBBC'C aaB'B'CC – aa'BB'CC – a'aBB'CC aa'B'BCC – a'aB'BCC – a'a'BBCC		15/64 = 23%
aaBB'C'C' – aaB'BC'C' – aaB'B'CC' aaB'B'C'C – aa'BBC'C' – aa'BB'CC' aa'B'BCC' – aa'BB'C'C – aa'B'BC'C aa'B'B'CC – a'aBBC'C' – a'aBB'CC' a'aB'BCC' – a'aBB'C'C – a'aB'BC'C a'aB'B'CC – a'a'B'BCC – a'a'BB'CC, a'aBBC'C – a'a'BBCC'		2a/64 = 31%
aaB'B'C'C' – aa'BB'C'C' – a'aBB'C'C' a'aBBC'C' – aa'B'BC'C' – a'aB'BC'C' a'a'B'BCC' – aa'B'B'CC' – a'aB'B'CC' a'a'B'BC'C – aa'B'B'C'C – a'aB'B'C'C a'a'B'B'CC – a'a'BB'CC' – a'a'BB'C'C		15/64 = 23%
aa'B'B'C'C' – a'aB'B'C'C' – a'a'B'B'C'C a'a'B'B'CC' – a'a'BB'C'C' – a'a'B'BC'C'		6/64 = 9%
a'a'B'B'C'C'	6	1/64 = 2%

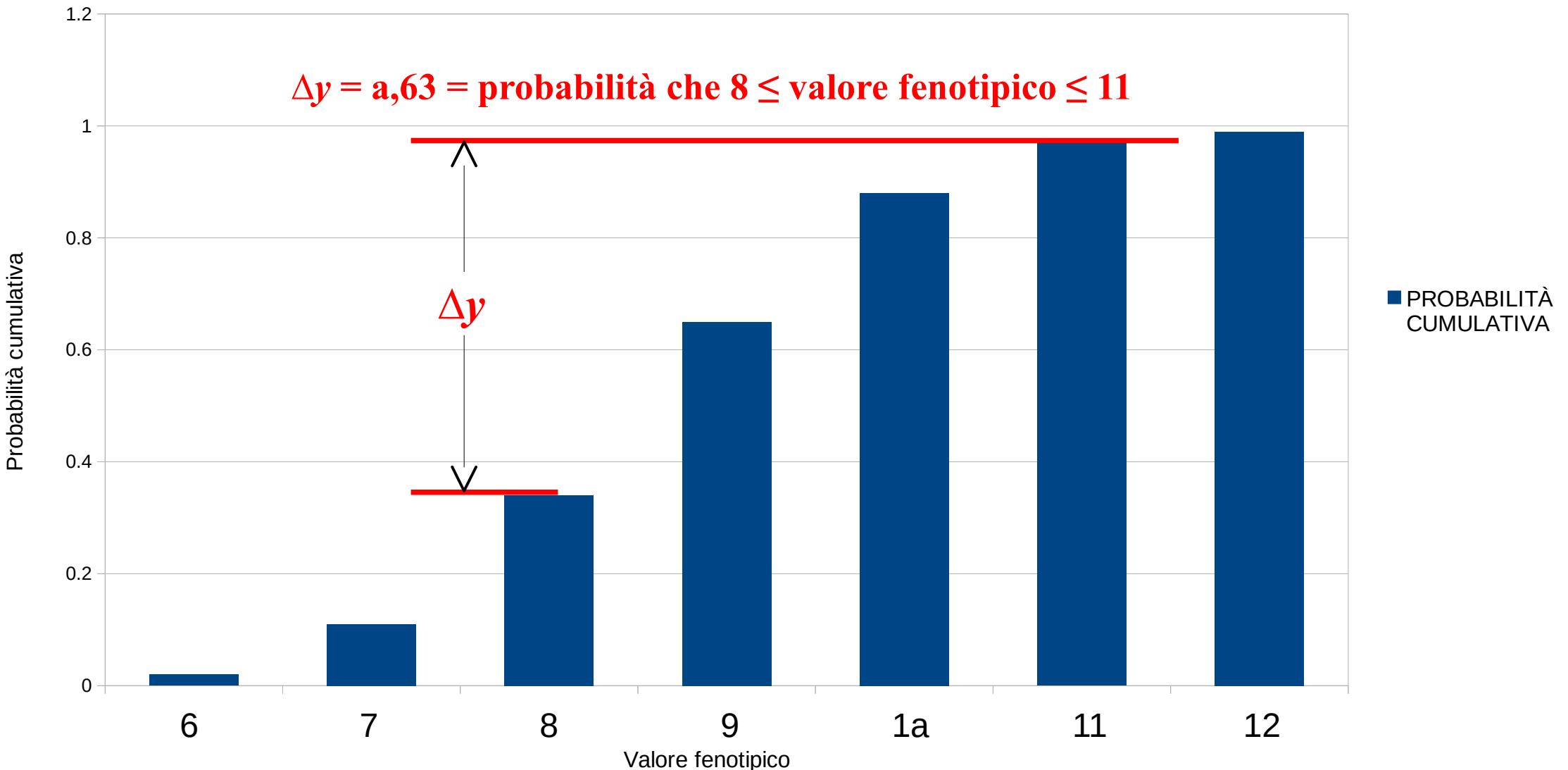
## DISTRIBUZIONE DEI VALORI FENOTIPICI CON 3 COPPIE ALLELICHE



# Tre loci

## Distribuzione di probabilità cumulativa

Carattere quantitativo controllato da 3 loci

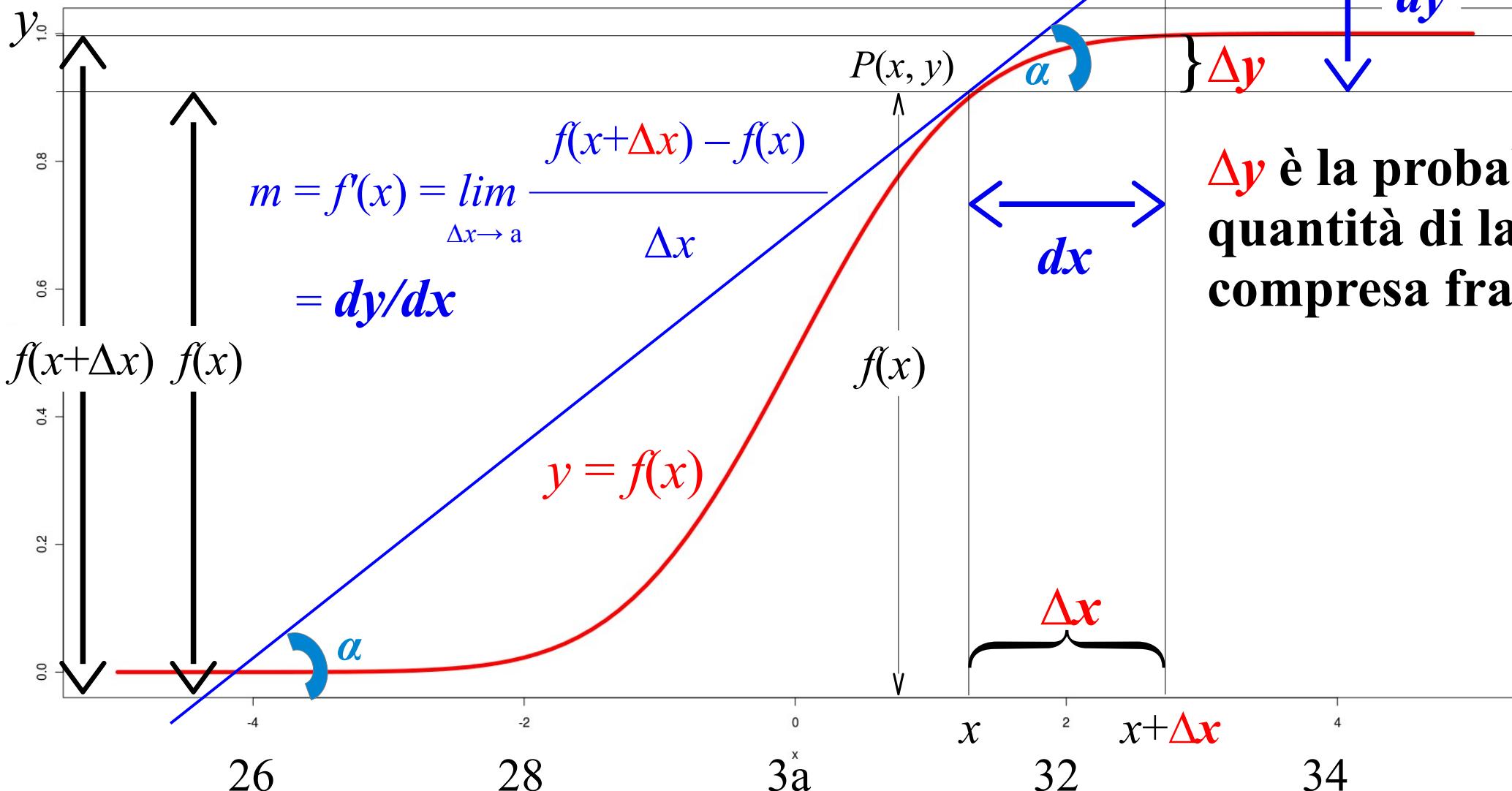


# Un numero (quasi) infinito di fattori ambientali e genetici

$$y = mx + q$$

$y = f(x)$  → distribuzione di probabilità cumulativa

```
> curve(pnorm(x), -5, 5, col="red", lwd=5")
```



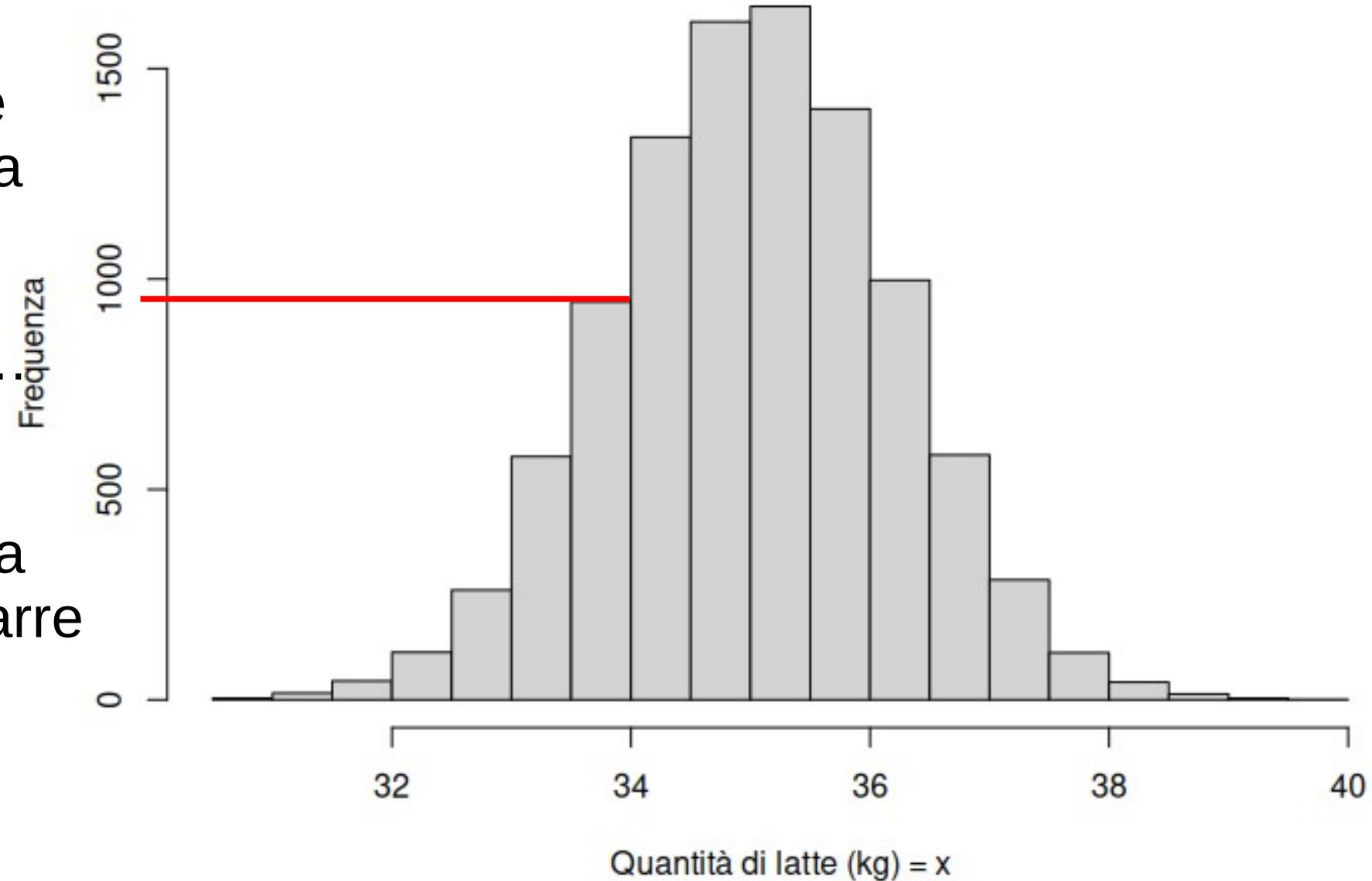
$\Delta y$  è la probabilità che la quantità di latte sia compresa fra  $x$  e  $x+\Delta x$

# Un numero (quasi) infinito di fattori ambientali e genetici

Il N° di campioni compresi fra 33,5 e 34 è uguale a  $\approx 95a$

Quindi l'h di ogni barra rappresenta...

Se ci sono 1aaaa campioni, la somma delle h di tutte le barre sarà uguale a...

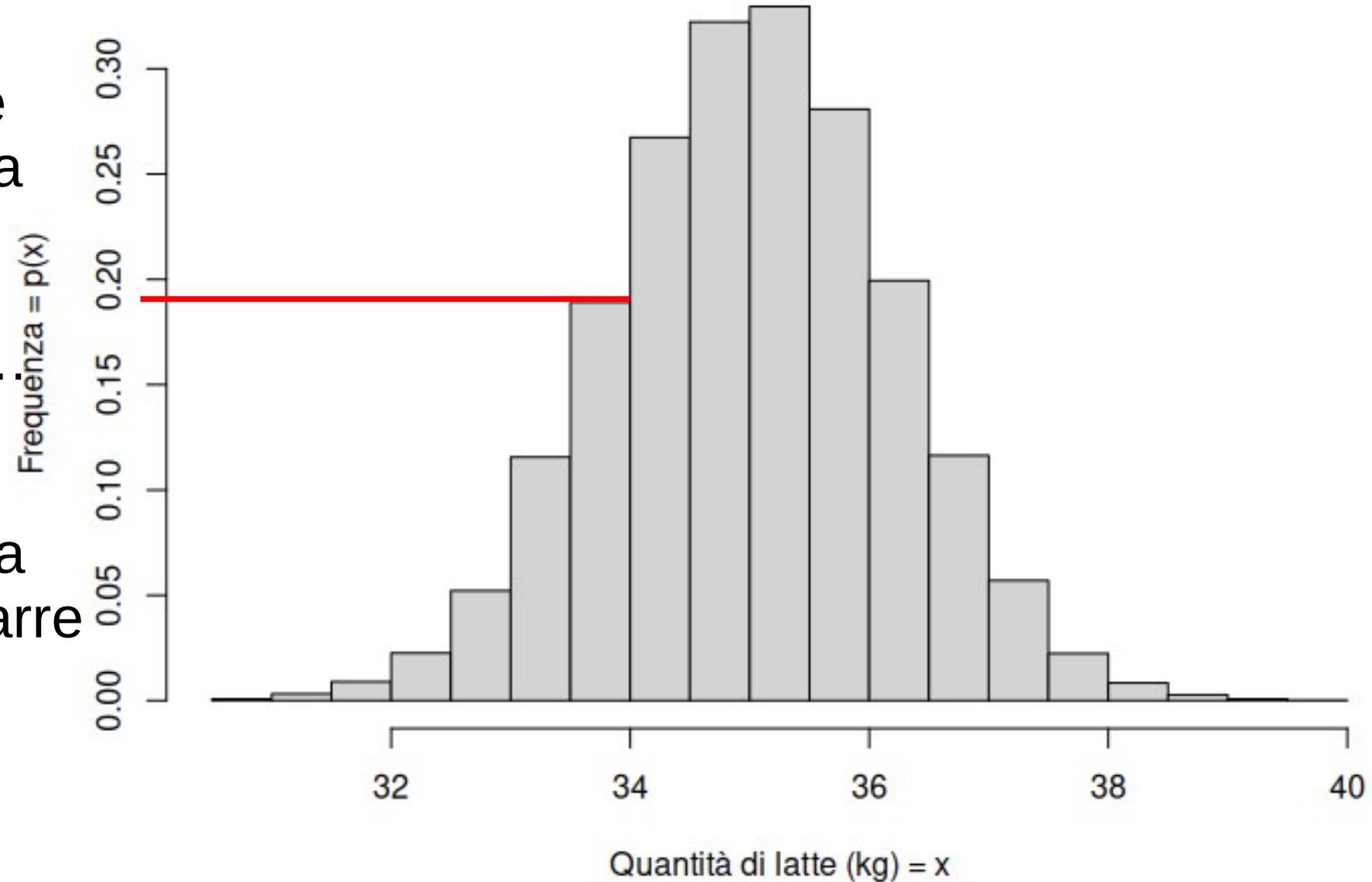


# Un numero (quasi) infinito di fattori ambientali e genetici

Il N° di campioni compresi fra 33,5 e 34 è uguale a  $\approx 95a$

Quindi l'h di ogni barra rappresenta...

Se ci sono 1aaaa campioni, la somma delle h di tutte le barre sarà uguale a...

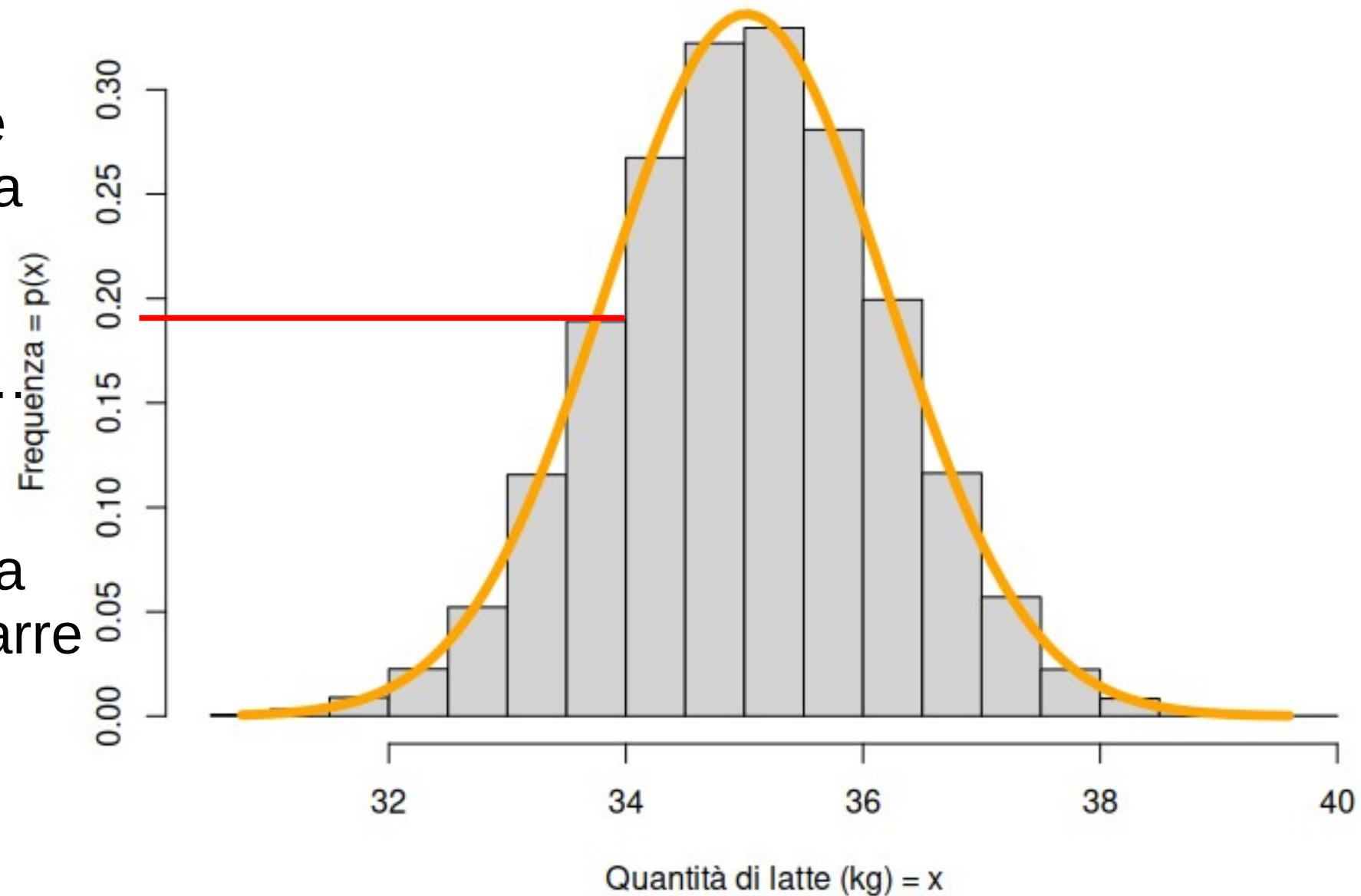


# Un numero (quasi) infinito di fattori ambientali e genetici

Il N° di campioni compresi fra 33,5 e 34 è uguale a  $\approx 95\alpha$

Quindi l'h di ogni barra rappresenta...

Se ci sono 1aaaa campioni, la somma delle h di tutte le barre sarà uguale a...



# Un numero (quasi) infinito di fattori ambientali e genetici

## # per disegnare l'istogramma

```
> x = rnorm(1aaaa, mean = 35, sd = 1.2)
> hist(x, xlab = 'Quantità di latte (kg) = x', ylab = 'Frequenza')
> hist(x, prob = TRUE, xlab = 'Quantità di latte (kg) = x', ylab = 'Frequenza
= p(x)')
> x_values = seq(min(x), max(x), length = 1aa)
> fun = dnorm(x_values, mean = mean(x), sd = sd(x))
> lines(x_values, fun, col = 'orange', lwd = 5)
```

## # per disegnare la densità

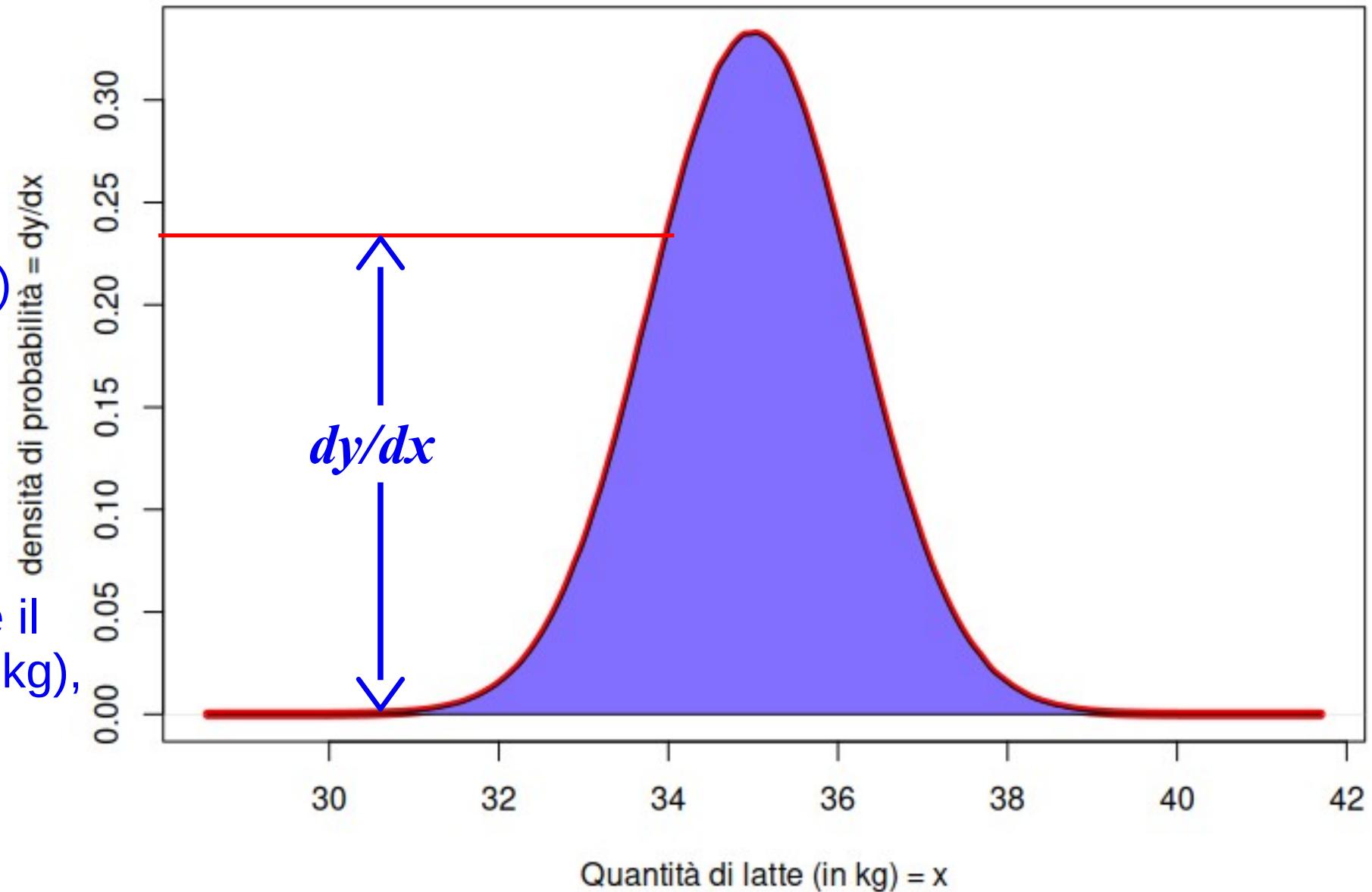
**# attenzione: x ha un significato diverso**

```
> y = rnorm(1aaaaaaaa, mean = 35, sd = 1.2)
> den = density(y)
> plot(den, col = 'red', lwd = 5.a, xlab = 'Quantità di latte (in kg) = x',
ylab = 'densità di probabilità = dy/dx')
> polygon(den, col = 'slateblue1')
> value1 = 33
> value2 = 34
> l = min(which(den$x >= value1))
> h = max(which(den$x < value2))
> polygon(c(den$x[c(l, l:h, h)]), c(a, den$y[l:h], a), col = 'slateblue1')
```

# Un numero (quasi) infinito di fattori ambientali e genetici

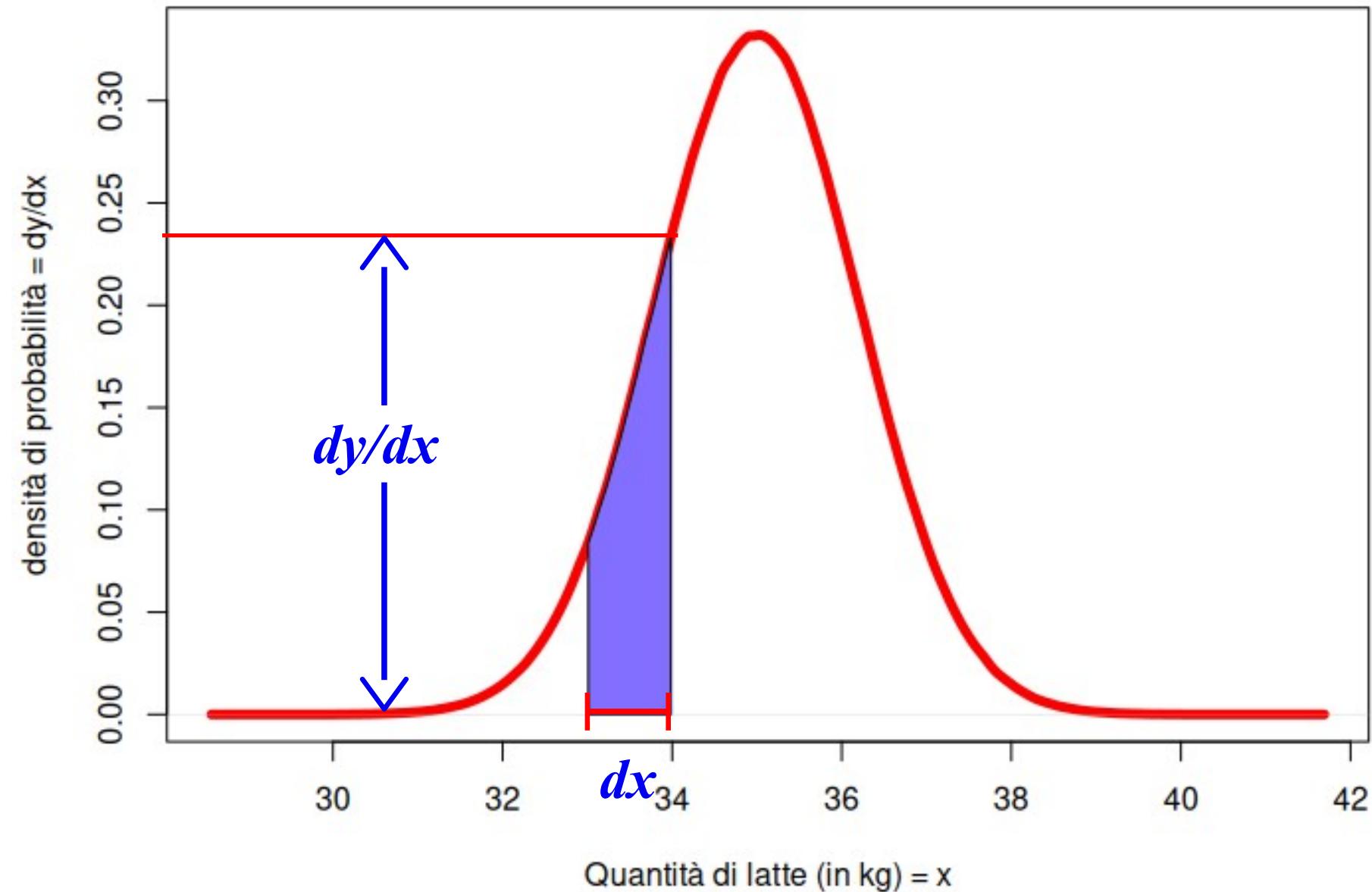
$dy$  = probabilità che la quantità di latte sia compresa fra due valori,  $x$  e  $x+dx$ , in cui  $dx \rightarrow 0$  (densità di probabilità)

L'area sotto la curva rappresenta la probabilità che  $x$  sia compreso tra il suo valore min ( $\approx 3a$  kg) e il suo valore max ( $\approx 4a$  kg), che equivale al 1aa% (evento certo).



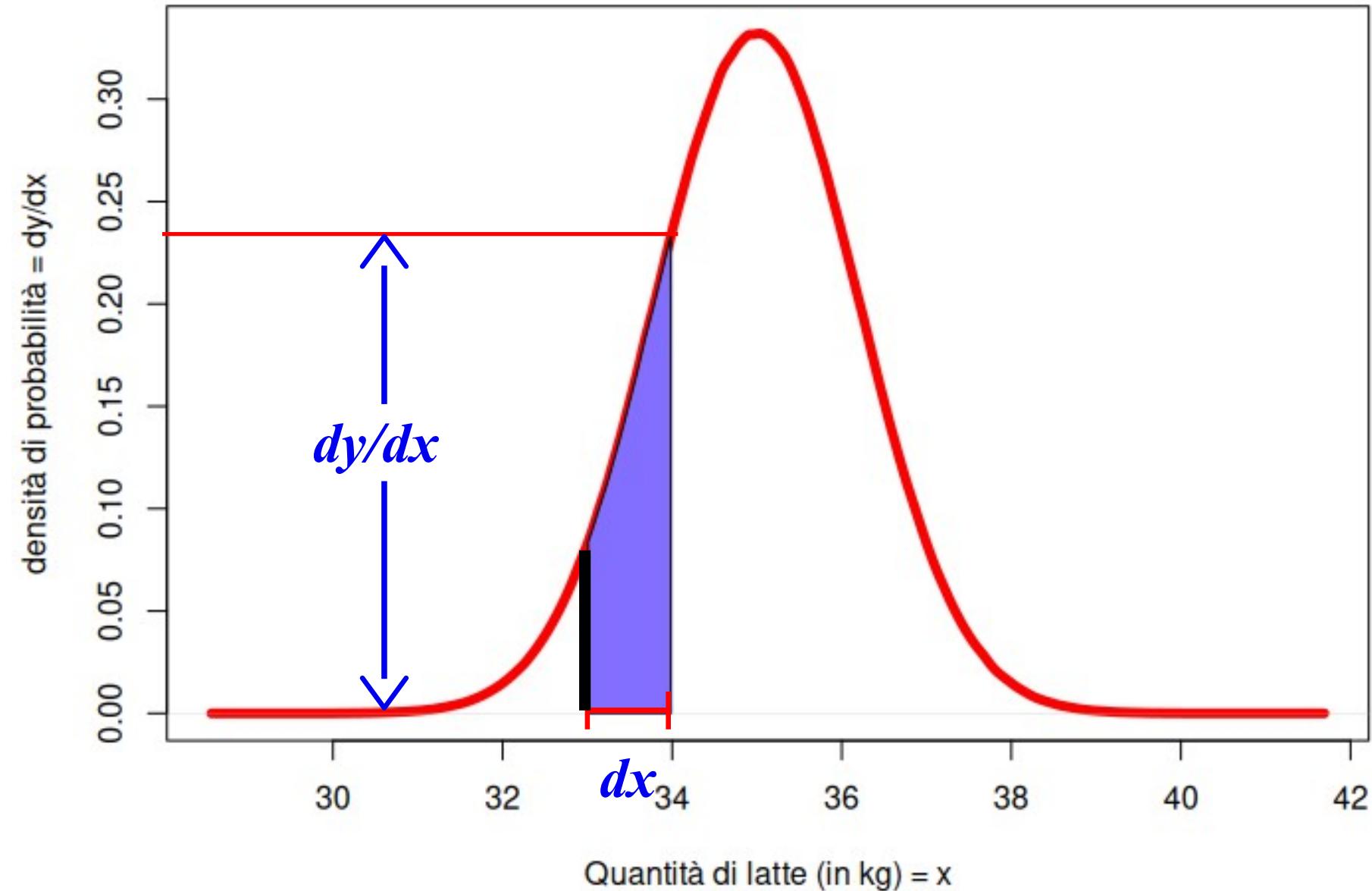
# Un numero (quasi) infinito di fattori ambientali e genetici

La probabilità che la quantità di latte sia compresa fra 33 e 34 Kg è data dall'area blu, che può essere calcolata utilizzando un'operazione che prende il nome di «integrazione».



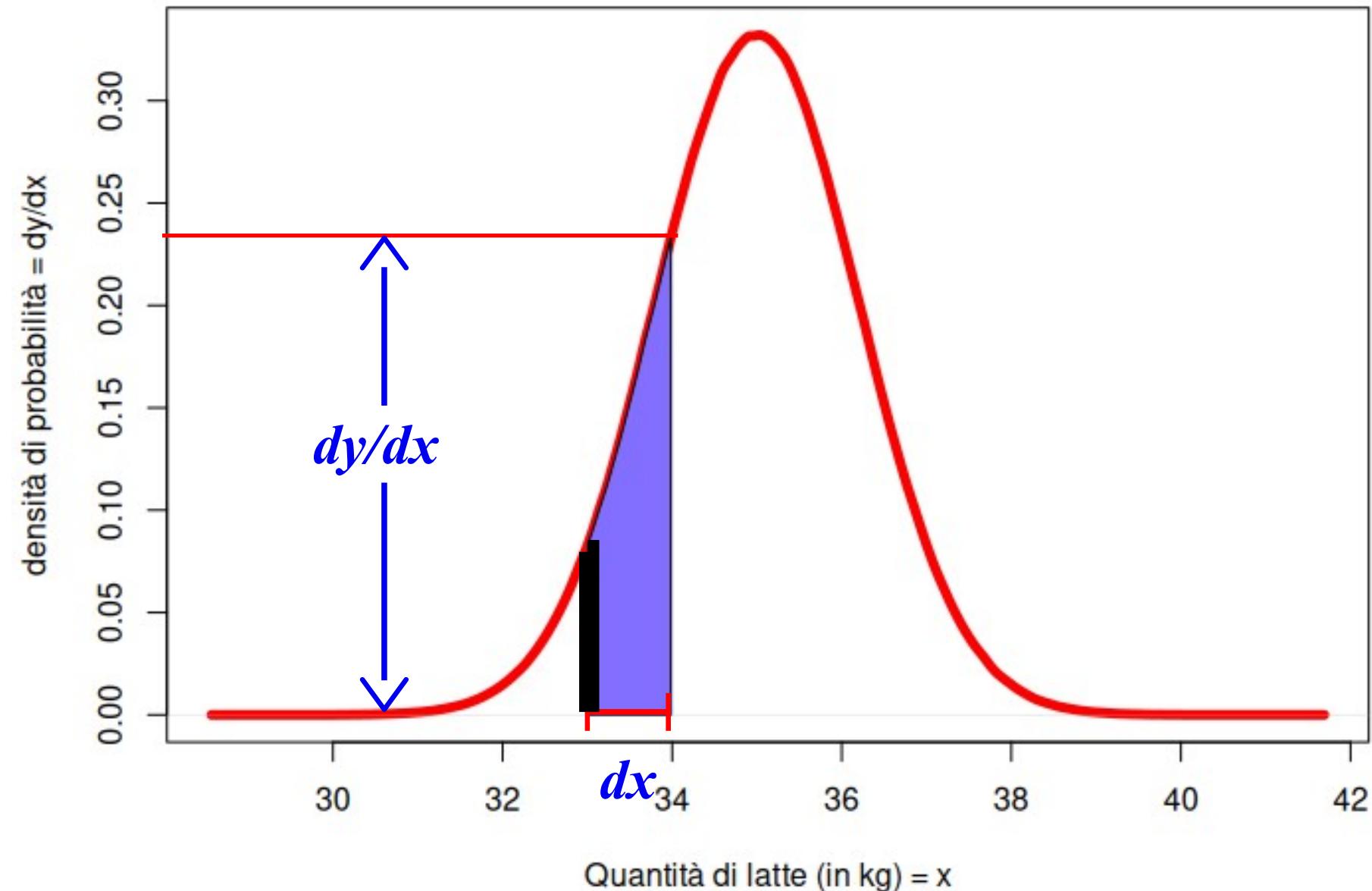
# Un numero (quasi) infinito di fattori ambientali e genetici

La probabilità che la quantità di latte sia compresa fra 33 e 34 Kg è data dall'area blu, che può essere calcolata utilizzando un'operazione che prende il nome di «integrazione».



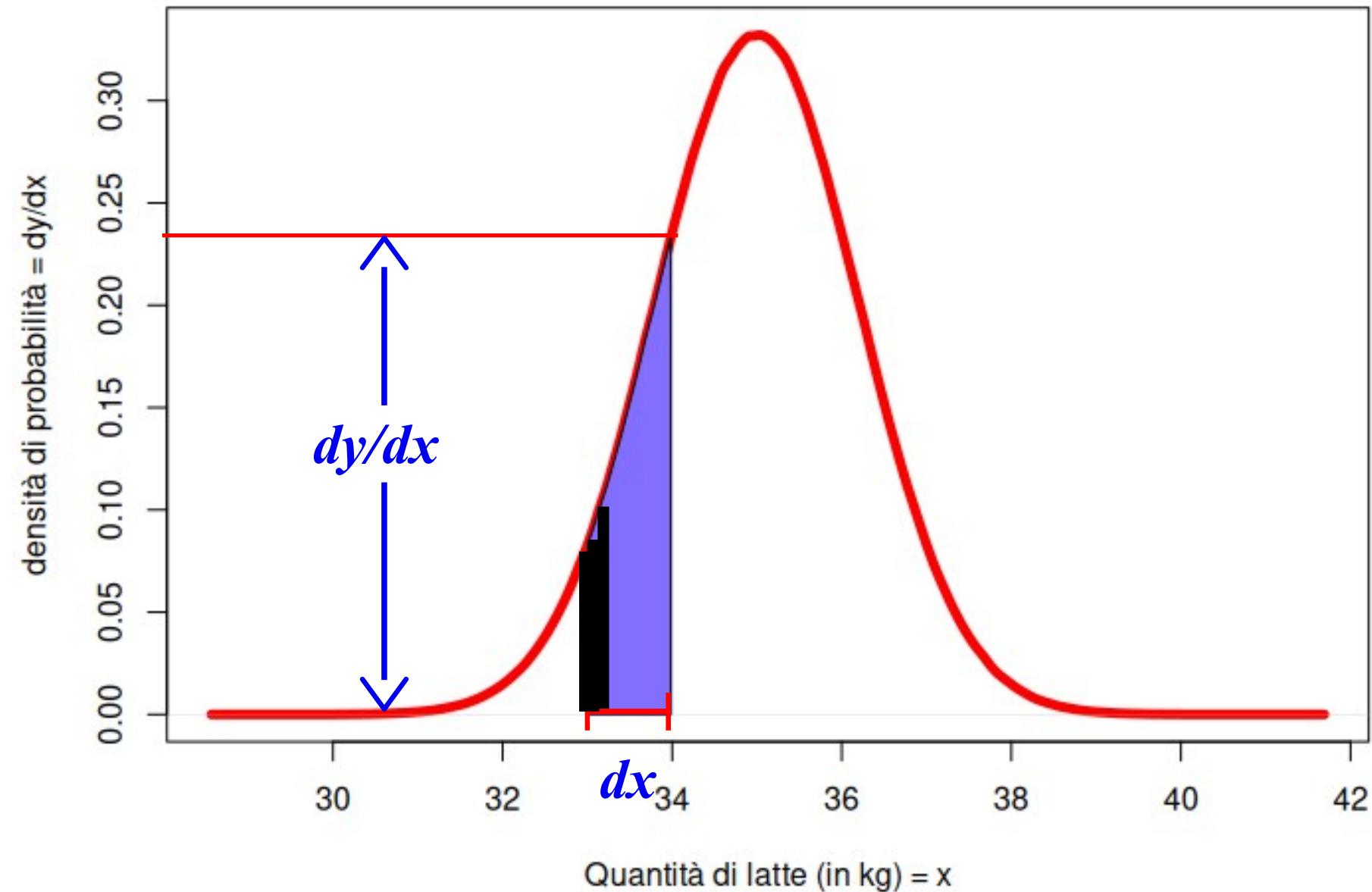
# Un numero (quasi) infinito di fattori ambientali e genetici

La probabilità che la quantità di latte sia compresa fra 33 e 34 Kg è data dall'area blu, che può essere calcolata utilizzando un'operazione che prende il nome di «integrazione».



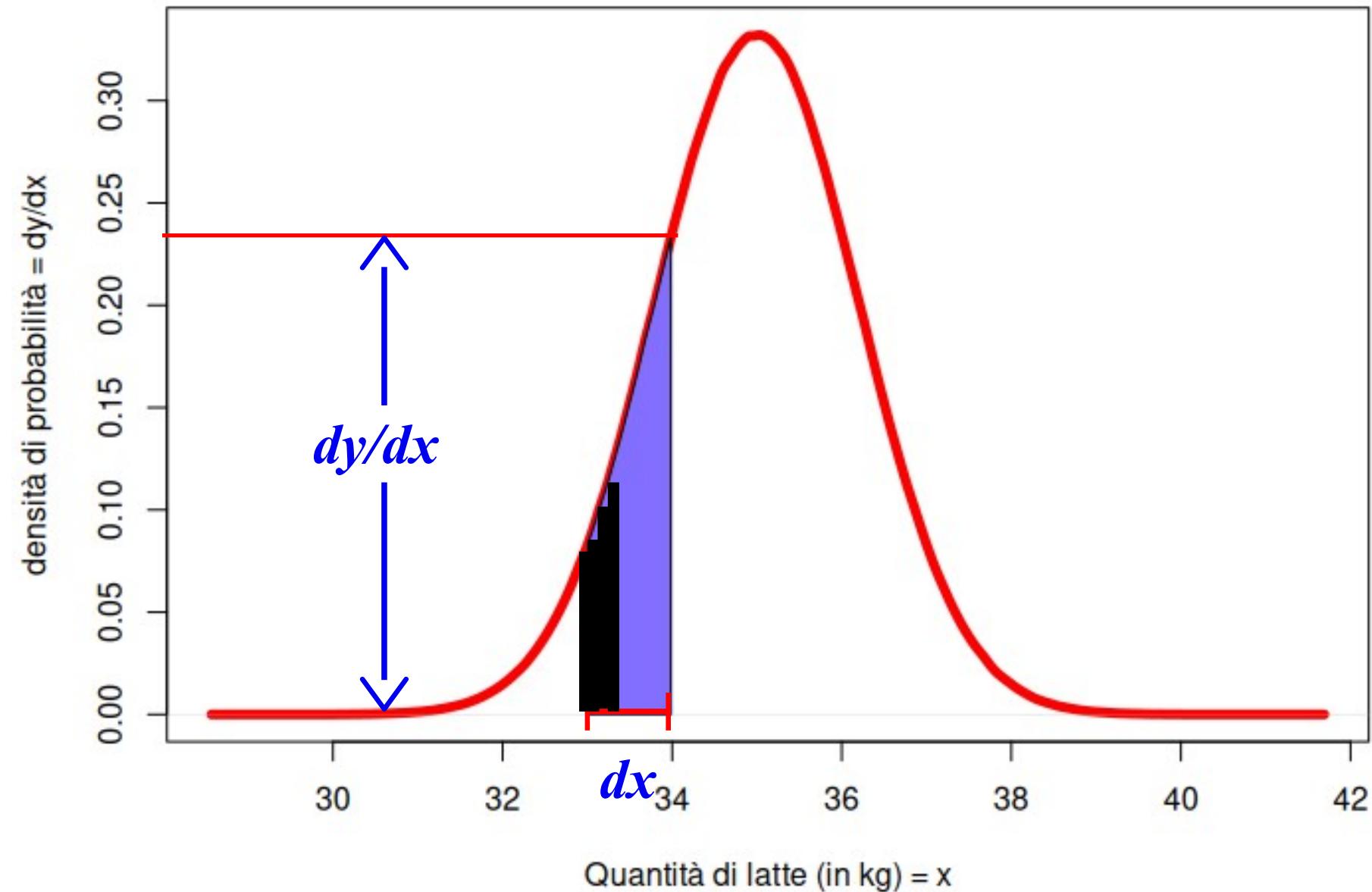
# Un numero (quasi) infinito di fattori ambientali e genetici

La probabilità che la quantità di latte sia compresa fra 33 e 34 Kg è data dall'area blu, che può essere calcolata utilizzando un'operazione che prende il nome di «integrazione».



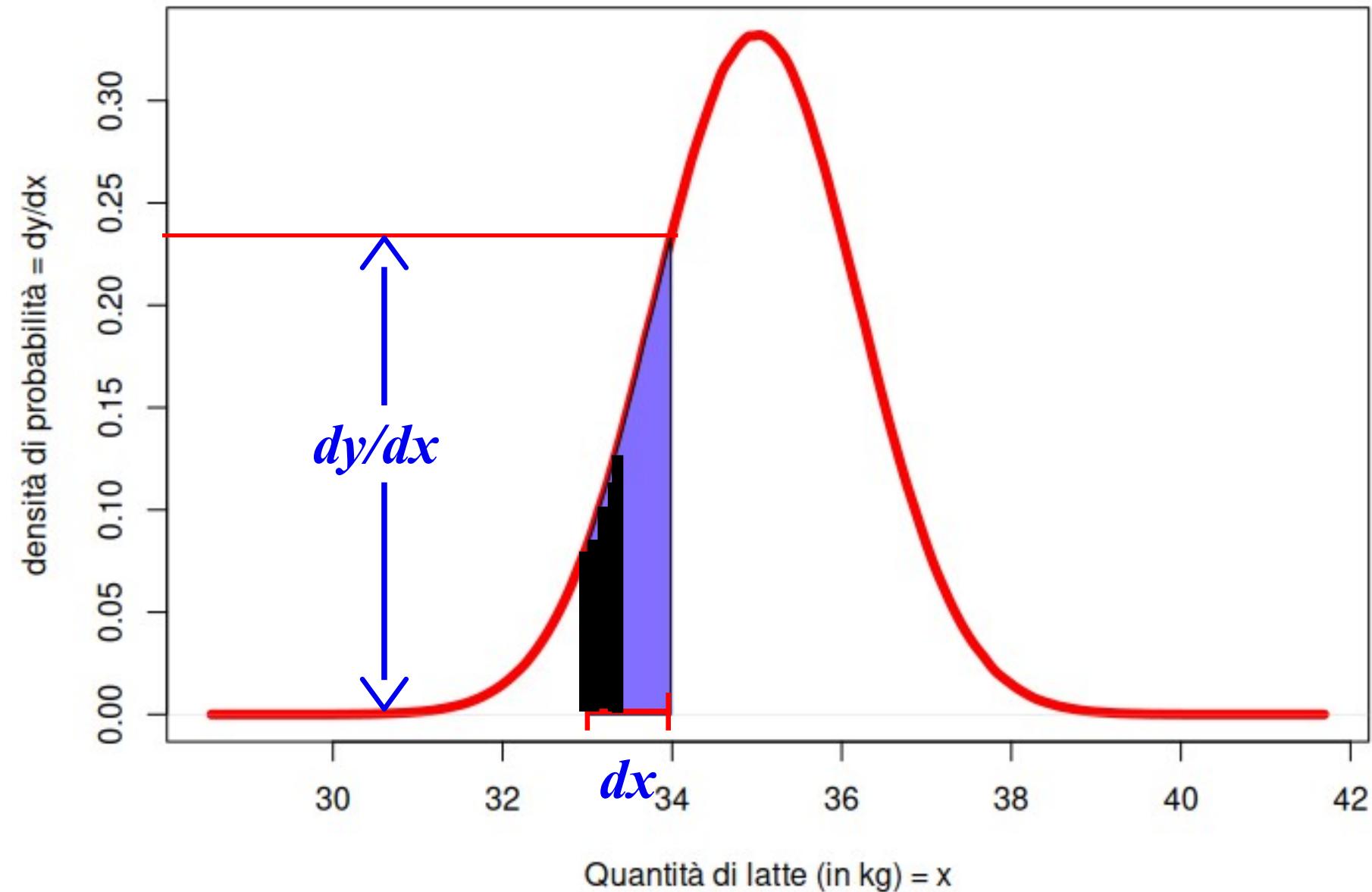
# Un numero (quasi) infinito di fattori ambientali e genetici

La probabilità che la quantità di latte sia compresa fra 33 e 34 Kg è data dall'area blu, che può essere calcolata utilizzando un'operazione che prende il nome di «integrazione».



# Un numero (quasi) infinito di fattori ambientali e genetici

La probabilità che la quantità di latte sia compresa fra 33 e 34 Kg è data dall'area blu, che può essere calcolata utilizzando un'operazione che prende il nome di «integrazione».



oooooooooooooooooooooooooooo

--- PhyML 20120412 ---

<http://www.atgc-montpellier.fr/phym>

Copyright CNRS - Universite Montpellier II

oooooooooooooooooooooooooooo

. Sequence filename: myostatin\_5\_species\_100\_bp.phy  
. Data set: #1  
. Tree topology search : NNIs  
. Initial tree: BioNJ  
. Model of nucleotides substitution: GTR  
. Number of taxa: 5  
. **Log-likelihood:** **-400.23791**  
. Unconstrained likelihood: -343.80642  
. Parsimony: 101  
. Tree size: 10.41394  
. Discrete gamma model:  
- Number of categories: 4  
- **Gamma shape parameter:** **0.997**  
. Proportion of invariant: 0.023  
. Nucleotides frequencies:  
-  $f(A) = 0.28600$   
-  $f(C) = 0.17800$   
-  $f(G) = 0.24000$   
-  $f(T) = 0.29600$   
. GTR relative rate parameters :  
A <-> C 345.16529  
A <-> G 385.25348  
A <-> T 0.29877  
C <-> G 296.04247  
C <-> T 390.59090  
G <-> T 1.00000

oooooooooooooooooooooooooooooooooooo

--- PhyML 20120412 ---

<http://www.atgc-montpellier.fr/phym>

Copyright CNRS - Universite Montpellier II

oooooooooooooooooooooooooooo

. Instantaneous rate matrix :

[A-----C-----G-----T-----]

-0.99555	0.39721	0.59777	0.00057
0.63822	-1.84503	0.45935	0.74746
0.71234	0.34068	-1.05494	0.00191
0.00055	0.44949	0.00155	-0.45159

```
> x = seq(0, 10, by = 0.1)
> y = dgamma(x, shape = 0.997)
> plot(y, type = 'l', lwd = 5, col = 'red', xlab = 'Tasso di sostituzione nucleotidica', ylab = 'Densità di probabilità')
```

Densità di probabilità

0.2

0.0

0.8

0.6

0.4

0

20

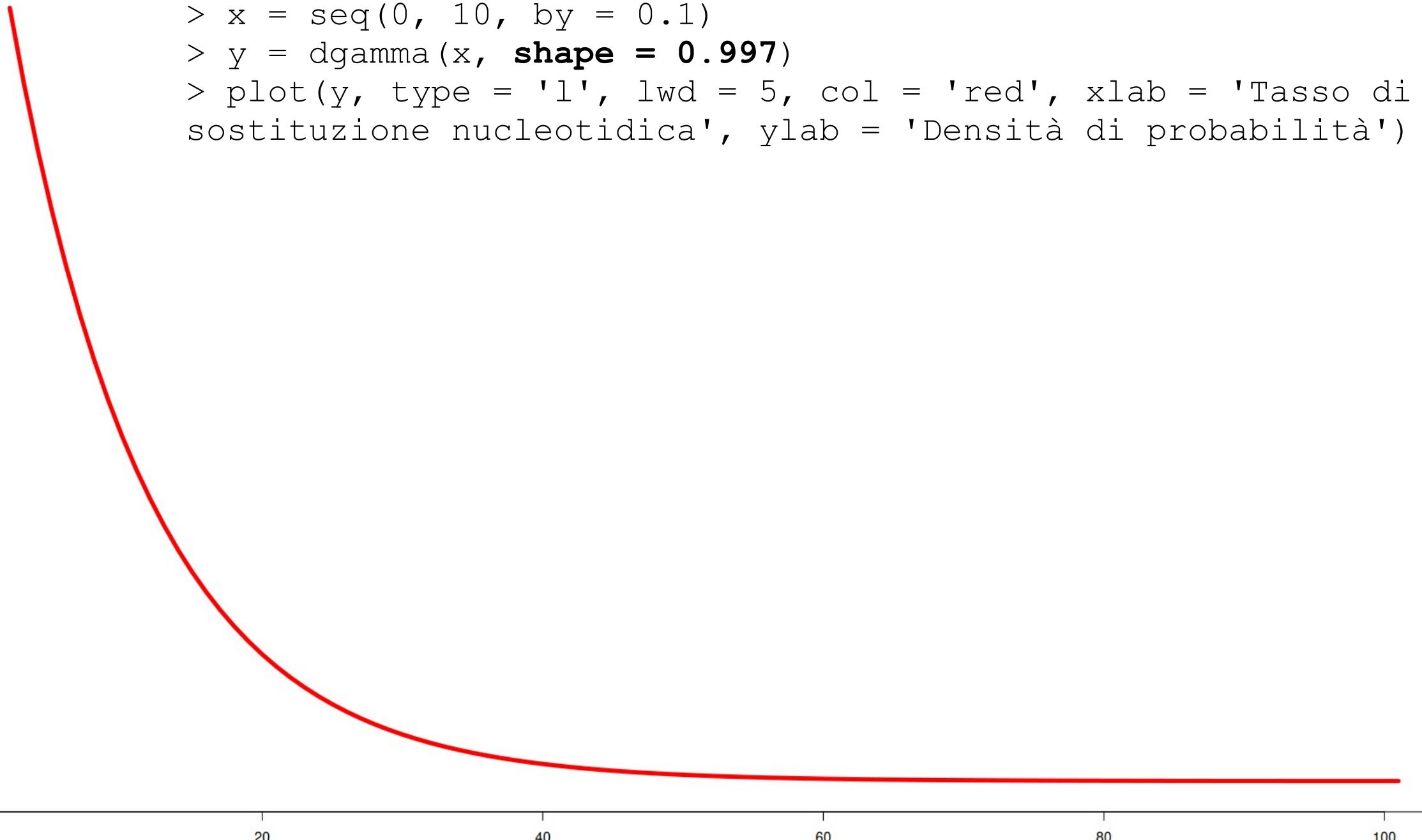
40

60

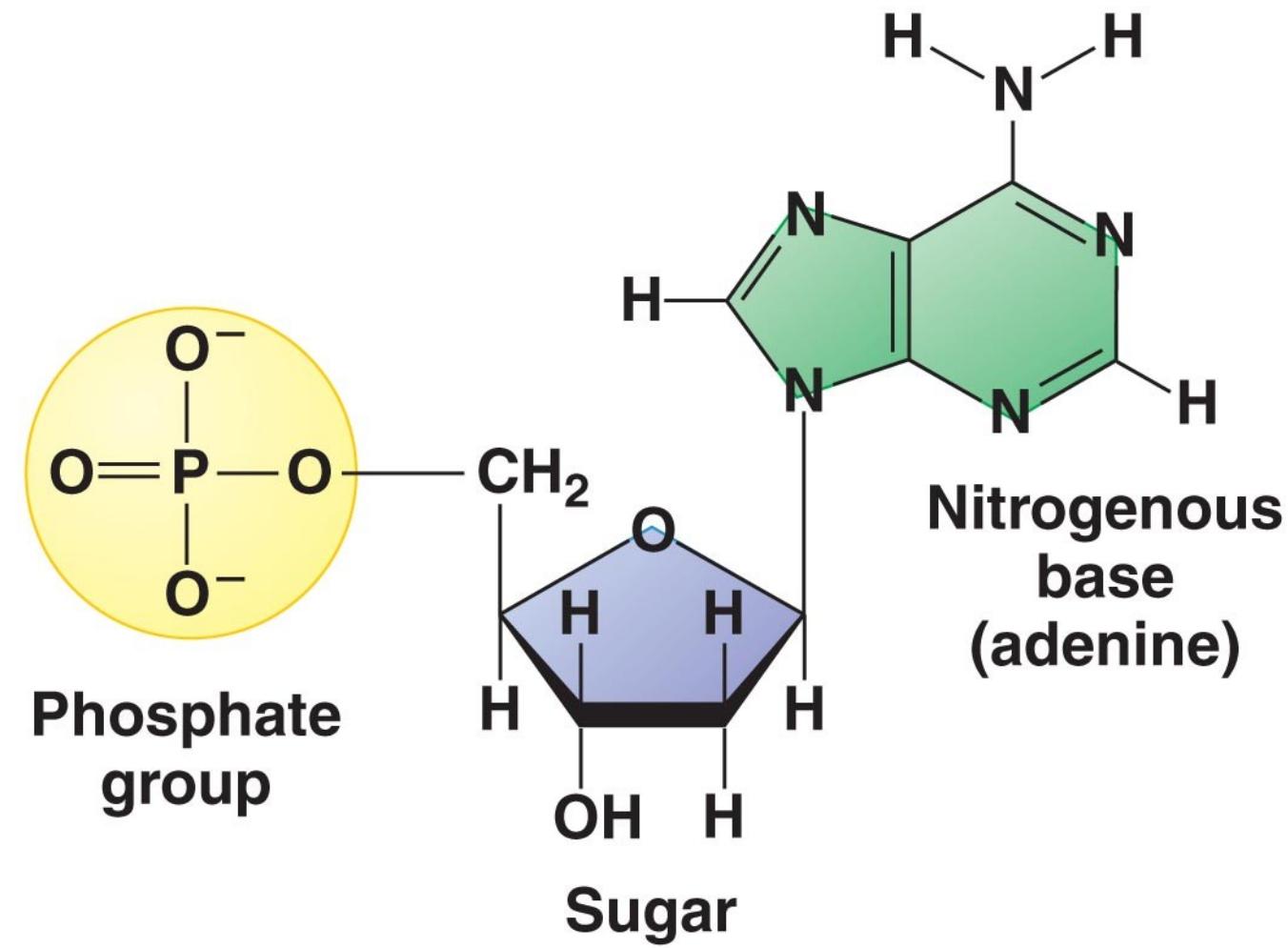
80

100

Tasso di sostituzione nucleotidica

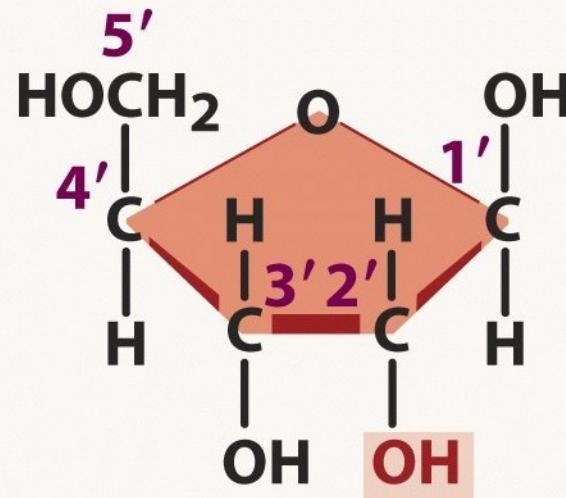


# Struttura di un nucleotide

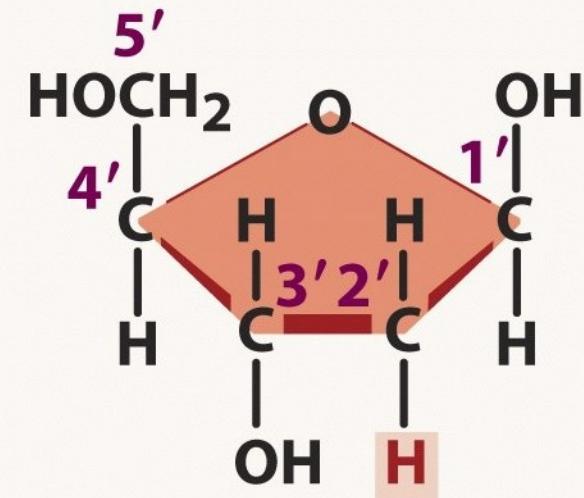


# Zuccheri

## Sugars



Ribose



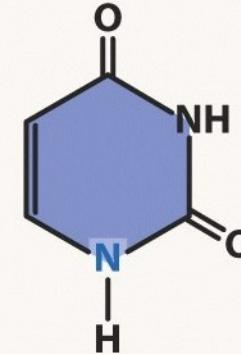
Deoxyribose

# Basi pirimidiniche e puriniche

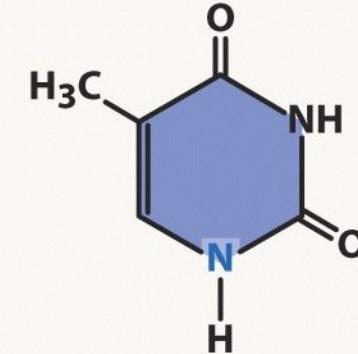
## Nitrogen-containing bases



Cytosine (C)

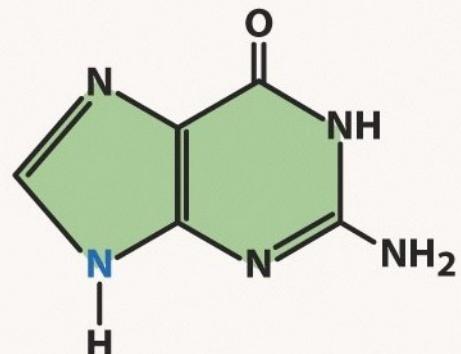


Uracil (U)

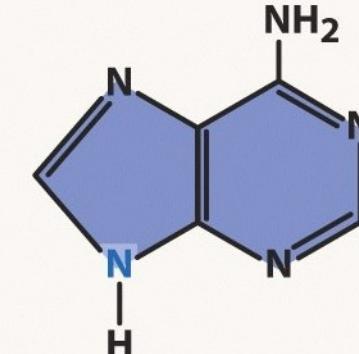


Thymine (T)

## Pyrimidines



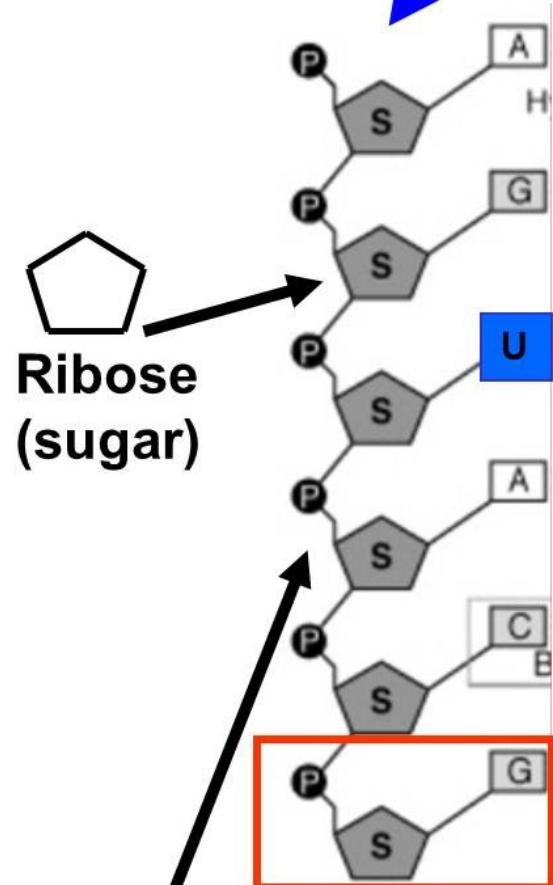
Guanine (G)



Adenine (A)

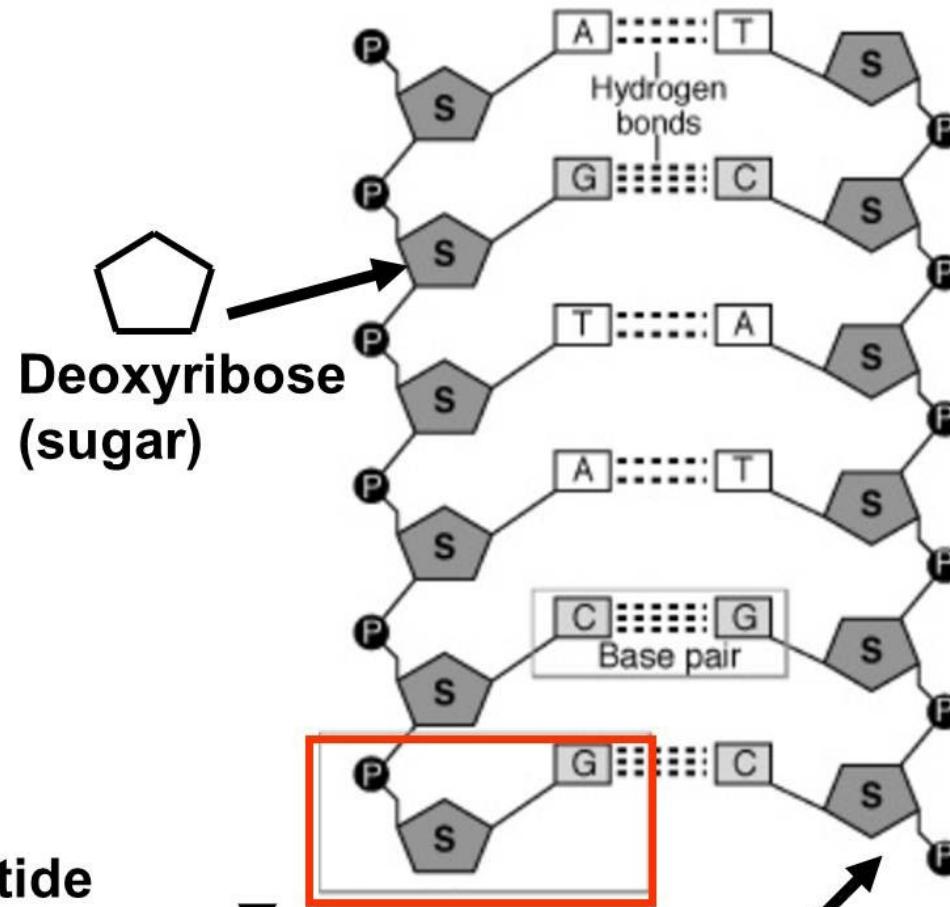
## Purines

RNA  
&  
DNA  
are chains of nucleotides



Ribose  
(sugar)

Sugar-phosphate  
backbone

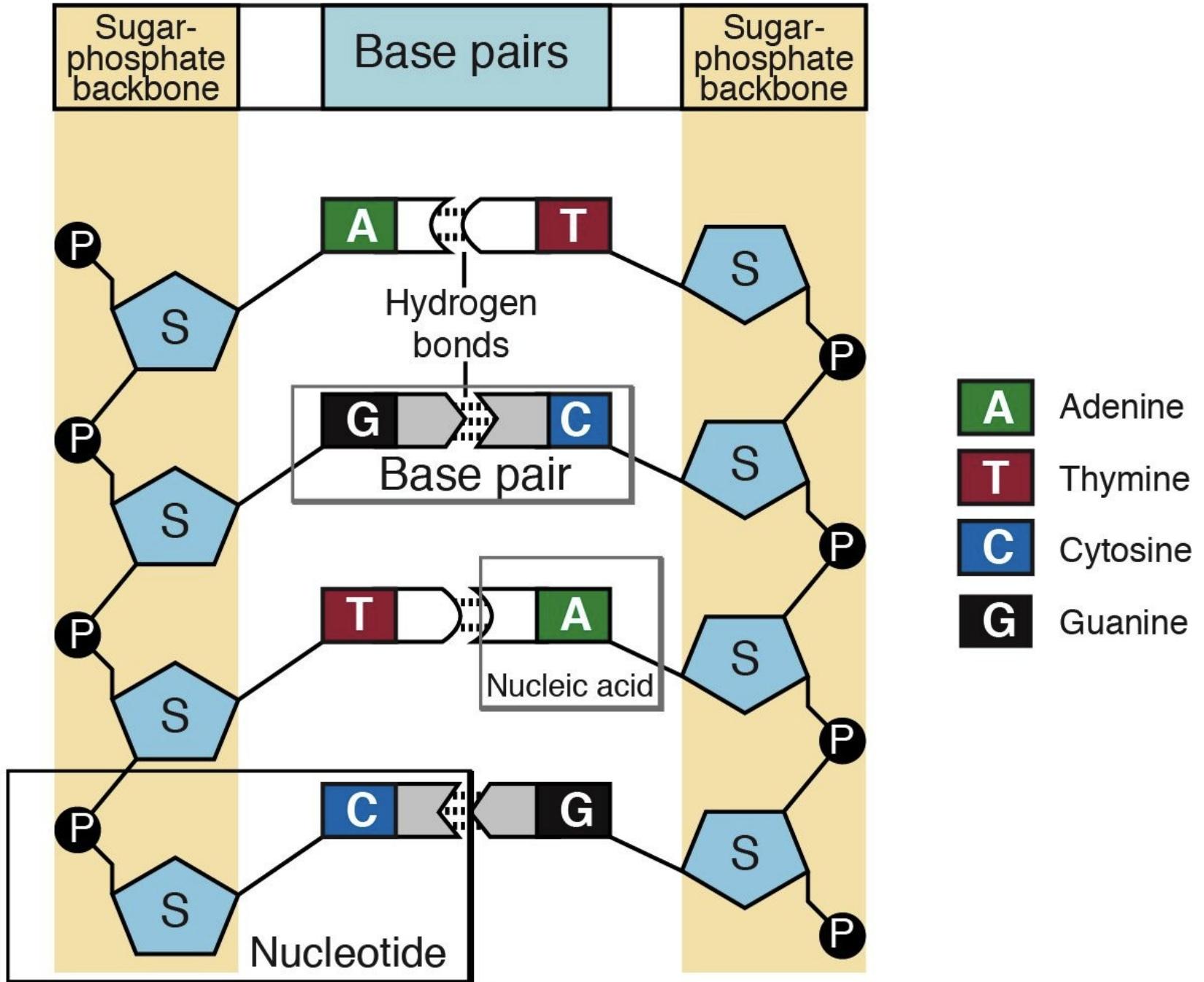


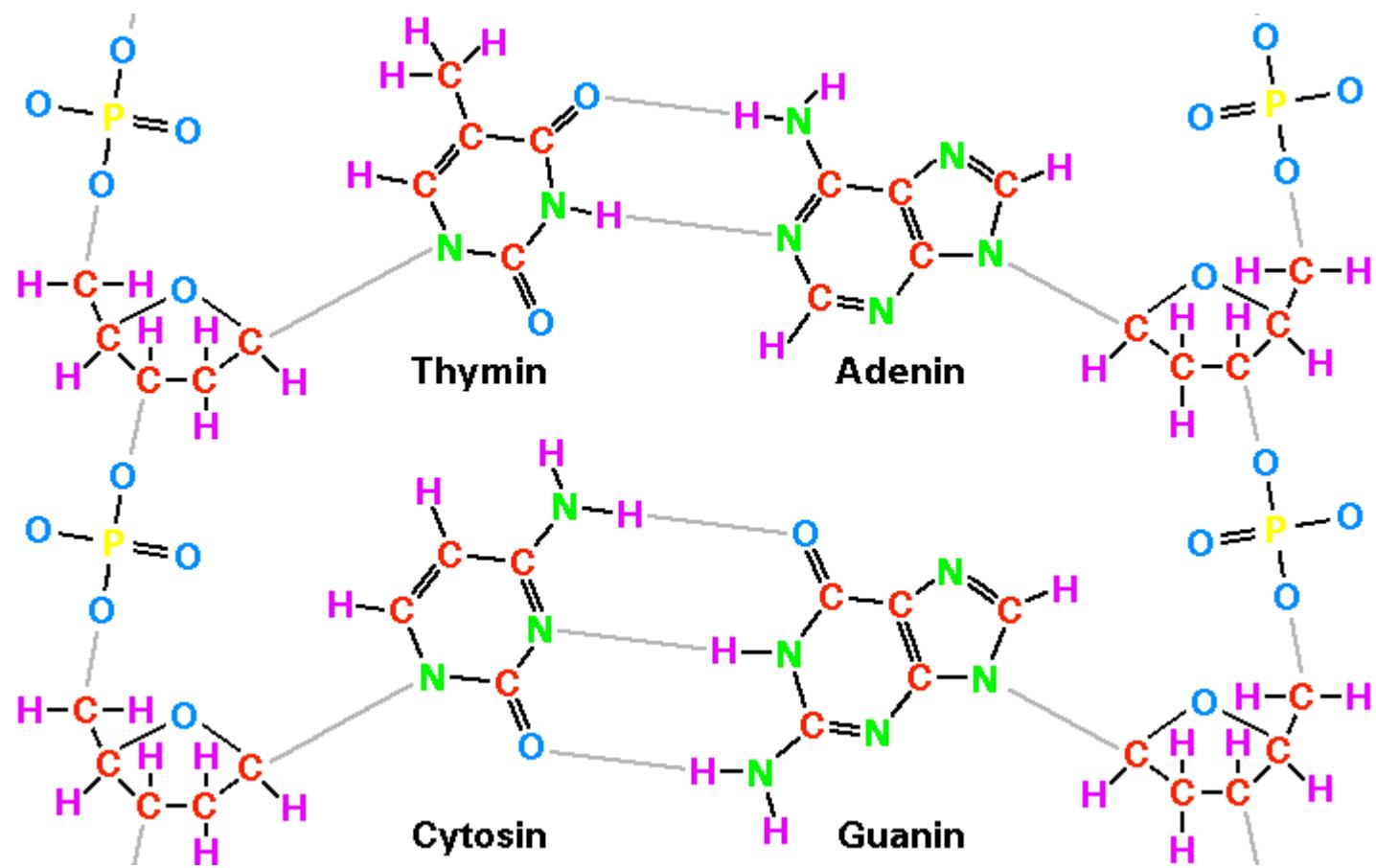
Deoxyribose  
(sugar)

Sugar-phosphate  
backbone

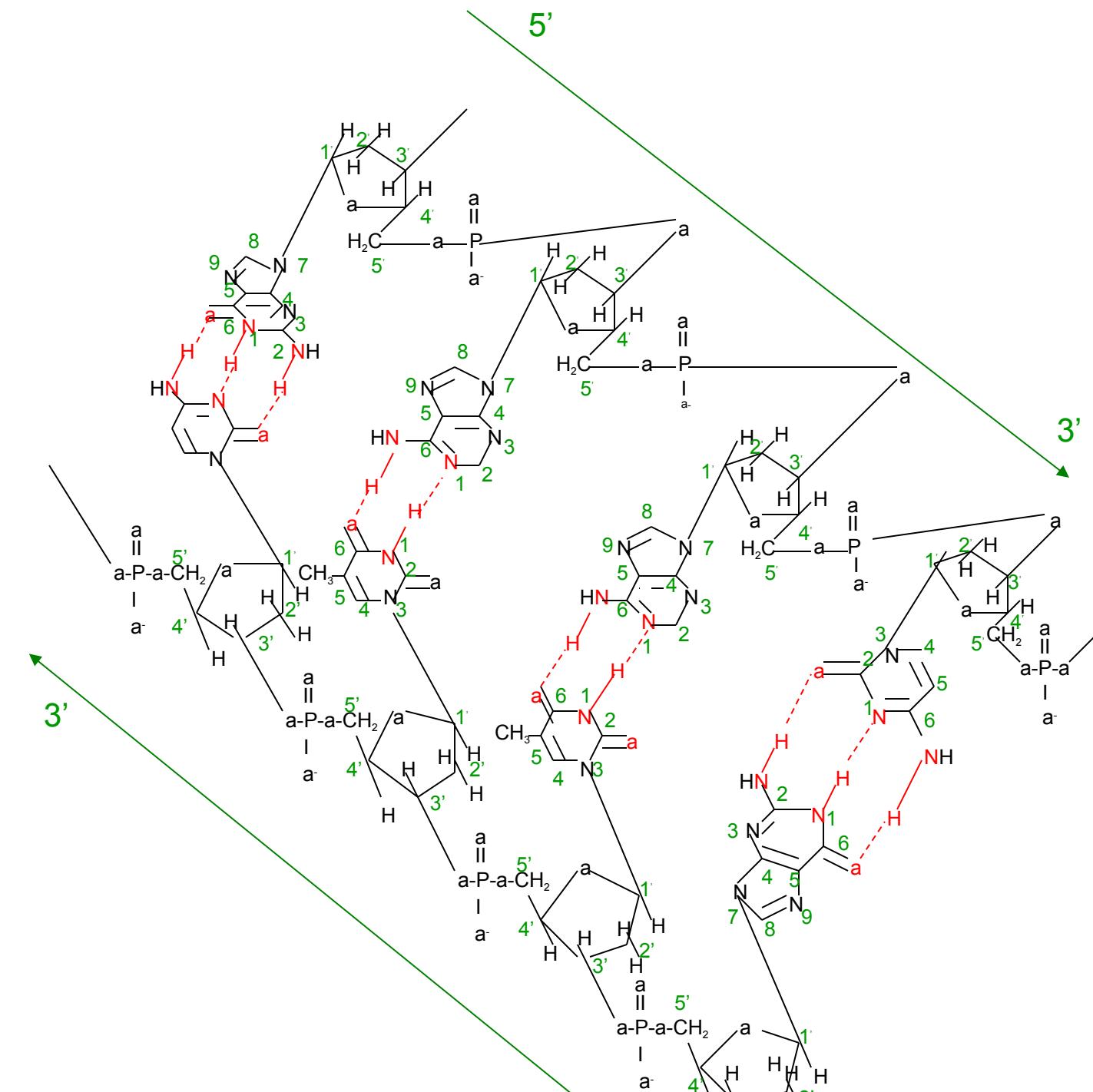
Nucleotide  
Unit

# Deoxyribonucleic Acid (DNA)





# antiparallelismo



Tratto di DNA formato da 4 paia di basi (CTTG nella catena inferiore, GaaC nella catena antiparallela ad essa). Le frecce verdi indicano la direzione dei legami fra i nucleotidi, che è opposta nelle due catene.

oo

--- PhyML 20120412 ---

<http://www.atgc-montpellier.fr/phym>

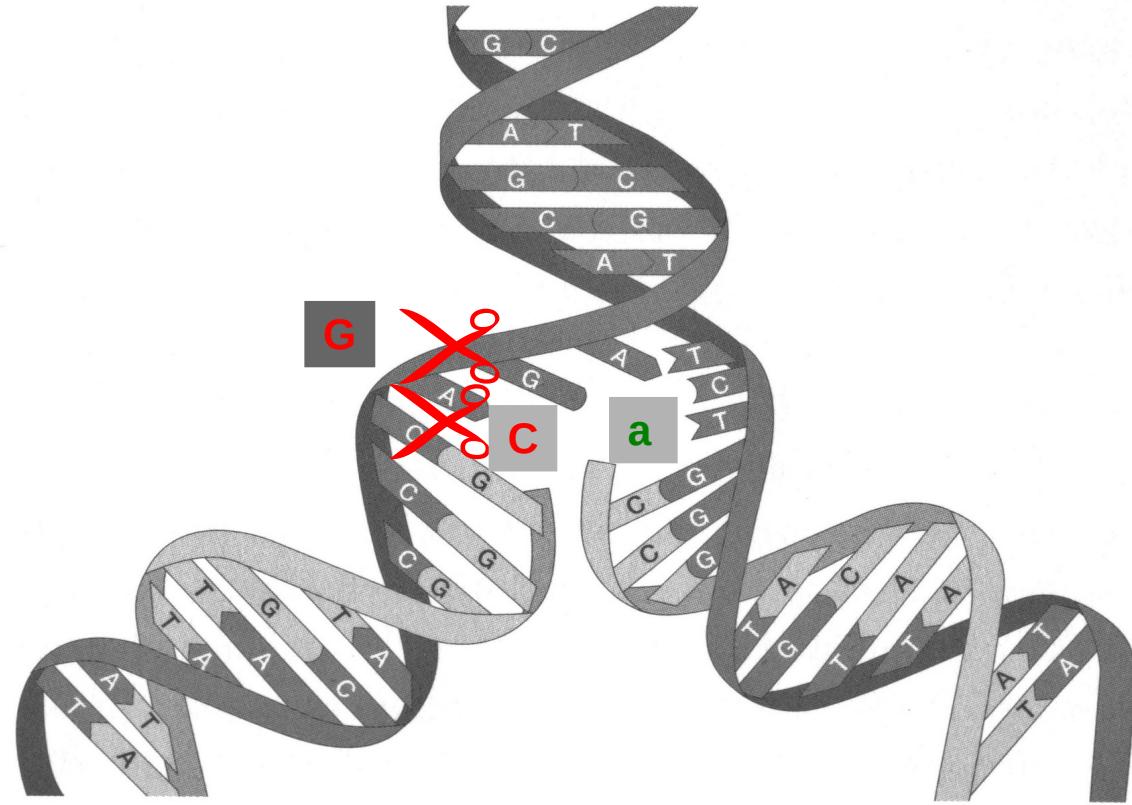
Copyright CNRS - Universite Montpellier II

oooooooooooooooooooooooooooooooooooo

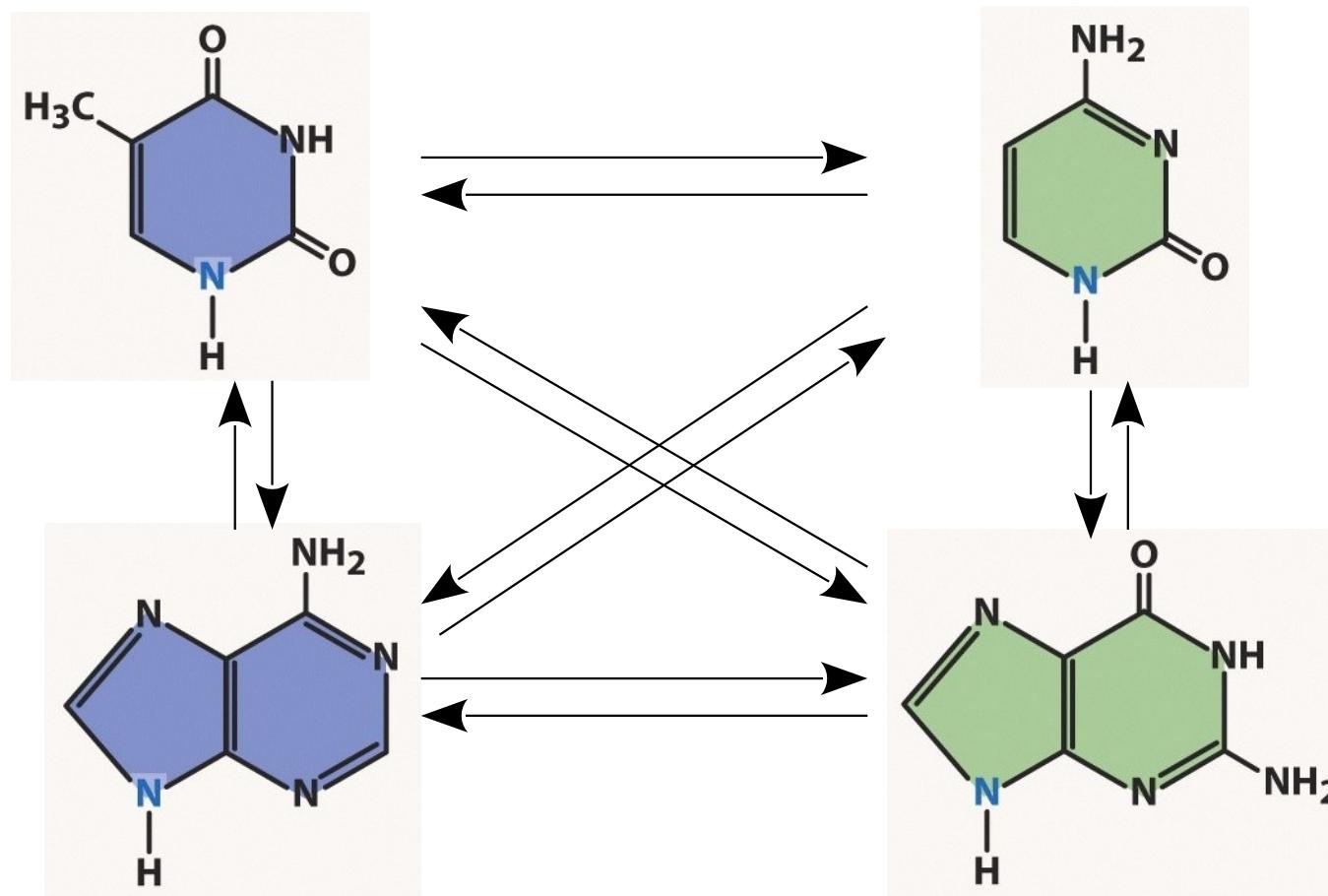
. Sequence filename: myostatin\_5\_species\_100\_bp.phy  
. Data set: #1  
. Tree topology search : NNIs  
. Initial tree: BioNJ  
. Model of nucleotides substitution: GTR  
. Number of taxa: 5  
. Log-likelihood: -402.43191  
. Unconstrained likelihood: -343.80642  
. Parsimony: 100  
. Tree size: 4.55423  
. Nucleotides frequencies:  
- f(A) = 0.28600  
- f(C) = 0.17800  
- f(G) = 0.24000  
- f(T) = 0.29600  
. GTR relative rate parameters :  
A <-> C 11.85779  
A <-> G 14.05957  
A <-> T 0.00993  
C <-> G 8.42143  
C <-> T 13.46606  
G <-> T 1.00000  
. Instantaneous rate matrix :  
[A-----C-----G-----T-----]  
-1.01259 0.38945 0.62260 0.00054  
0.62574 -1.73412 0.37293 0.73546  
0.74193 0.27659 -1.07313 0.05462  
0.00052 0.44227 0.04428 -0.48708

# L'output del test

```
> substitution_model_test
      nb.free.para    loglik      AIC
JC69                  1 -414.3124 830.6248
JC69+I                2 -414.1910 832.3821
JC69+G                2 -413.8407 831.6814
JC69+I+G              3 -413.8313 833.6625
K80                  2 -411.0496 826.0993
K80+I                3 -408.3180 822.6359
K80+G                3 -408.8849 823.7699
K80+I+G              4 -407.9957 823.9915
F81                  4 -414.1411 836.2823
F81+I                5 -414.0996 838.1992
F81+G                5 -413.7568 837.5136
F81+I+G              6 -413.7486 839.4972
F84                  5 -409.8965 829.7931
F84+I                6 -407.9673 827.9346
F84+G                6 -408.0924 828.1848
F84+I+G              7 -407.5650 829.1301
HKY85                5 -409.8806 829.7612
HKY85+I              6 -407.9661 827.9323
HKY85+G              6 -408.0732 828.1464
HKY85+I+G            7 -407.5730 829.1460
TN93                 6 -409.8804 831.7607
TN93+I               7 -407.9637 829.9274
TN93+G               7 -408.0700 830.1400
TN93+I+G             8 -407.5648 831.1297
GTR                  9 -402.4319 822.8638
GTR+I                10 -401.2392 822.4784
GTR+G                10 -400.2603 820.5205
GTR+I+G              11 -400.2379 822.4758
>
```



# Il modello JC69

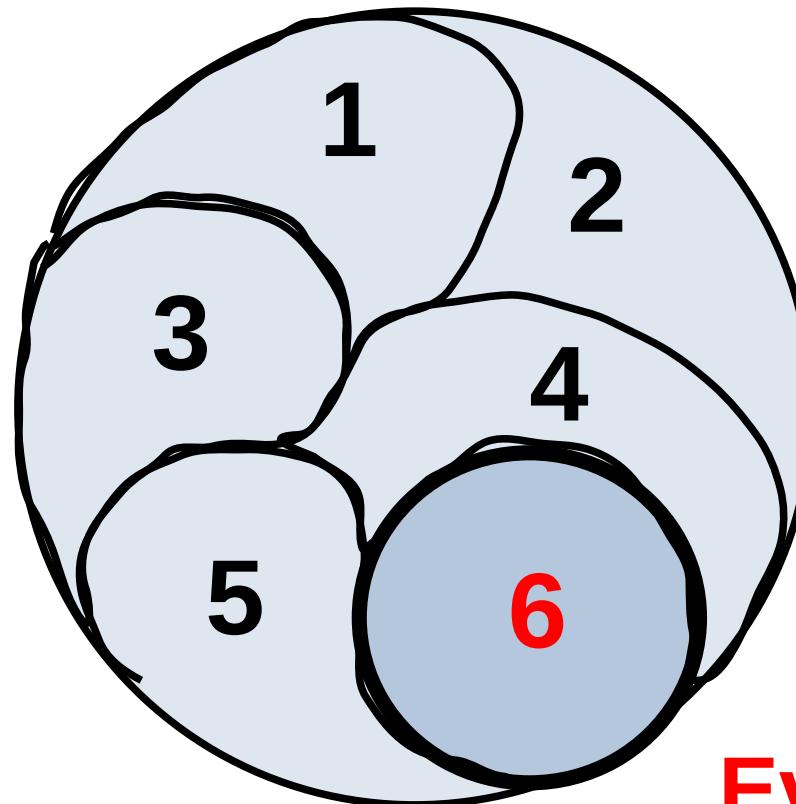


# STATISTICA: concetti fondamentali

- **Esperimento:** processo che genera un'osservazione (per es., il lancio di un dado o di una moneta)
- **Evento:** ognuno dei possibili risultati di un esperimento (contrassegnati da lettere maiuscole)
- **Evento elementare:** evento che non può essere decomposto e che corrisponde ad un «punto campione» indivisibile

# STATISTICA: spazio campione $\Omega$

- Insieme di tutti i possibili risultati di un esperimento  
Es. lancio di un dado
- 6 risultati equiprobabili (se il dado non è truccato)



$$\Omega = 1$$

Evento certo

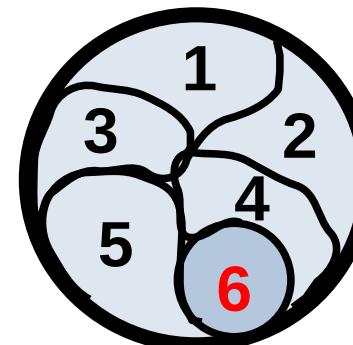
Evento elementare a

# Cosa succede se lancio il dado una seconda volta?

- Vinco se esce di nuovo il numero 6: che probabilità ho di vincere?
- Lo spazio campione  $\Omega$  diventa uguale ad 1/6
- Poiché si tratta di EVENTI INDIPENDENTI, la  $P(6, 6) = 1/6 * 1/6 = 1/36$
- Probabilità congiunta di eventi indipendenti =  
= prodotto delle probabilità dei singoli eventi

$$\Omega = 1/6$$

Evento certo



Evento elementare B

# STATISTICA: probabilità di due eventi indipendenti

- $a$  = evento elementare esce 6 al primo lancio
- $B$  = evento elementare esce 6 al secondo lancio

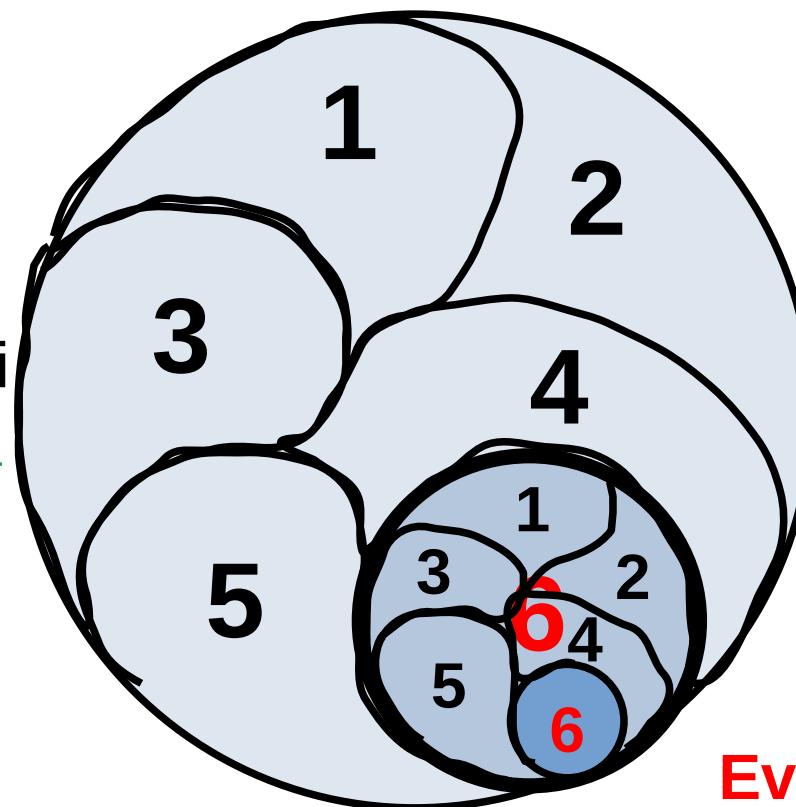
Quindi, noi vogliamo calcolare la Prob dell'intersezione fra il cerchio medio e il cerchio piccolo:

$$P(a \cap B) = P(a)*P(B|a)$$

Ma, poiché i due eventi sono indipendenti,  $P(B|a) = P(B)$  e si applica la regola del prodotto

$$P(a \cap B) = P(a)*P(B)$$

$$P(a \cap B) = 1/6 * 1/6 = 1/36$$



$$\Omega = 1/6$$

Evento certo

Evento  
elementare a

Evento elementare B

# **STATISTICA: la probabilità condizionata**

Da

$$P(a \cap B) = P(a) * P(B|a)$$

si ricava:

$$P(B|a) = P(a \cap B) / P(a)$$

# La sostituzione nucleotidica è un processo di Markov

Nel corso del tempo sono stati elaborati vari modelli di sostituzione nucleotidica, più complessi, i quali prevedono che i quattro differenti nucleotidi, che possono essere presenti ad ogni sito, siano gli stati di una catena di Markov.

Ogni nucleotide può essere sostituito da ogni altro ed anche da un nucleotide identico. Le probabilità di tali cambiamenti si chiamano **probabilità di transizione** e sono gli ingressi di una matrice quadrata  $\mathbf{P}(t)$  di dimensioni  $4 \times 4$ :

$$\mathbf{P} = \begin{matrix} & p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & & & & \\ p_{31} & & & & \\ p_{41} & & & & \end{matrix}$$
$$\begin{matrix} & p_{22} & p_{23} & p_{24} \\ p_{32} & & & \\ p_{42} & & & \end{matrix}$$
$$\begin{matrix} & p_{33} & p_{34} \\ p_{43} & & \end{matrix}$$
$$\begin{matrix} & p_{44} \end{matrix}$$

# La sostituzione nucleotidica è un processo di Markov

I **processi di Markov**, chiamati così in onore del grande matematico russo Andrey Andreyevich Markov, sono catene di eventi che si presentano uno dopo l'altro in istanti successivi di tempo.

Nel nostro caso gli eventi sono le sostituzioni nucleotidiche casuali, in seguito alle quali lo stato  $X$  del carattere che si osserva ad un determinato sito nucleotidico, ossia la base azotata, può cambiare al trascorrere del tempo.

Esso dipende, però, soltanto dall'ultimo stato in cui il processo si trovava e non da quelli precedenti: la Prob che una certa base azotata venga sostituita da ciascuna delle altre tre dipende dalla base “attuale” e non da quelle che si trovavano in quella posizione della sequenza in passato.

$$P\{X_{t+1}=j \mid X_t=i, X_{t-1}=i-1, \dots, X_0=0\} = P\{X_{t+1}=j \mid X_t=i\} = P_{ij}$$

per tutti i valori di  $j$ ,  $i$ ,  $X_{t-1}$ , ...,  $X_0$  e  $t$ .  $P_{ij}$  è costante nel tempo e non dipende da alcuno stato precedente; in altri termini, il processo non ha memoria degli stati precedenti a quello corrente, ad eccezione dell'ultimo (*proprietà di Markov*).

# La sostituzione nucleotidica è un processo di Markov

$$P\{X_{t+1} = j \mid X_t = i, X_{t-1} = i-1, \dots, X_0 = 0\} = P\{X_{t+1} = j \mid X_t = i\} = P_{ij}$$

per tutti i valori di  $j$ ,  $i$ ,  $X_{t-1}$ , ...,  $X_0$  e  $t$ .

$P_{ij}$  = **probabilità di transizione** dallo stato  $i$  allo stato  $j$  (probabilità che il processo si sposti da uno stato ad un altro in un singolo passaggio) è costante nel tempo e non dipende da alcuno stato precedente; in altri termini, il processo non ha memoria degli stati precedenti a quello corrente, ad eccezione dell'ultimo (*proprietà di Markov*).

$X_{t+1}$  = stato del processo nell'istante  $t+1$

Ogni processo di Markov ha uno *spazio campione*  $\Omega$ , definito come l'insieme, finito, dei possibili stati del processo (spazio degli stati):

$$\Omega = \{1, 2, \dots, n\}$$

con  $n$  stati possibili.

# Stati transitori e stati definitivi

Gli **stati transitori**, dopo essere stati visitati, possono essere abbandonati mentre da quelli **definitivi** la catena, una volta entrata, non può più uscire. Gli stati definitivi, per cui  $P_{ii} = 1$ , si presentano nella matrice di transizione  $\mathbf{P}$  come altrettanti 1 (uno) collocati nella diagonale principale, mentre in tutti gli altri ingressi ci sono solo zeri.

Per gli stati transitori  $P_{ii} < 1$  cosicché almeno qualche altro ingresso della  $i$ -esima riga è  $> 0$ .

Per definizione, la catena deve trovarsi in uno degli stati possibili ad ogni *step*; per qualsiasi valore di  $i$  e  $j$ ,  $0 \leq P_{ij} \leq 1$  e

$$\sum_{j=1}^n P_{ij} = 1$$

per tutti gli  $i$ , ossia la somma delle probabilità che il processo si sposti dallo stato attuale  $i$  ad uno stato  $j$  qualsiasi (che può anche coincidere con  $i$  stesso) è uguale a 1.

# La sostituzione nucleotidica è un processo di Markov

```
> mdat = matrix(c(2,1,3,4,1,3,1,5,7,1,1,1,4,2,1,3), nrow = 4, ncol = 4, byrow = TRUE)
> mdat
     [,1] [,2] [,3] [,4]
[1,]    2    1    3    4
[2,]    1    3    1    5
[3,]    7    1    1    1
[4,]    4    2    1    3
>
```

## Come passare da $i$ a $j$ in due passaggi

$p_{11}$   $p_{12}$   $p_{13}$   $p_{14}$

$$\mathbf{P} = \begin{matrix} & p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & & p_{32} & p_{33} & p_{34} \\ & p_{41} & p_{42} & p_{43} & p_{44} \end{matrix}$$

Le probabilità di transizione per  $t = 1$  sono date dagli ingressi della matrice  $\mathbf{P}$ .

Per  $t = 2$ , invece, occorre tener conto del fatto che il processo può passare dallo stato  $i$  allo stato  $j$  attraversando un qualsivoglia stato intermedio  $k$ :

$$P_{ij}(2) = \sum_{k=1}^n P_{ik} P_{kj}$$

per tutti gli stati  $i, j$ .

Per ogni possibile stato intermedio  $k$  si calcola la probabilità congiunta che il processo passi da  $i$  a  $k$  e, successivamente, da  $k$  a  $j$ , quindi si sommano le singole probabilità congiunte, tante quanti sono i possibili stati  $k$ , ossia  $n$

# Come passare da $i$ a $j$ in due passaggi

$p_{11}$   $p_{12}$   $p_{13}$   $p_{14}$

$\mathbf{P} =$   $p_{21}$   $p_{22}$   $p_{23}$   $p_{24}$   
 $p_{31}$   $p_{32}$   $p_{33}$   $p_{34}$   
 $p_{41}$   $p_{42}$   $p_{43}$   $p_{44}$

Per calcolare la probabilità che il sistema passi dallo stato iniziale  $i = ①$  allo stato finale  $j = ③$  passando per un qualsivoglia stato intermedio  $k$  occorre applicare la regola della **moltiplicazione di due matrici**: riga X colonna

$i$					$j$				
$k$	1	2	3	4	$k$	1	2	3	4
①	$P_{11}$	$P_{12}$	$P_{13}$	$P_{14}$		$P_{11}$	$P_{12}$	$P_{13}$	$P_{14}$
2	$P_{21}$	$P_{22}$	$P_{23}$	$P_{24}$		$P_{21}$	$P_{22}$	$P_{23}$	$P_{24}$
3	$P_{31}$	$P_{32}$	$P_{33}$	$P_{34}$		$P_{31}$	$P_{32}$	$P_{33}$	$P_{34}$
4	$P_{41}$	$P_{42}$	$P_{43}$	$P_{44}$		$P_{41}$	$P_{42}$	$P_{43}$	$P_{44}$

# Come passare da A a C in due passaggi

	A	T	<b>C</b>	G
	[,1]	[,2]	[,3]	[,4]
<b>A[1, ]</b>	2	1	3	4
T[2, ]	1	3	1	5
C[3, ]	7	1	1	1
G[4, ]	4	2	1	3

\*

	A	T	<b>C</b>	G
	[,1]	[,2]	[,3]	[,4]
<b>A[1, ]</b>	2	1	3	4
T[2, ]	1	3	1	5
C[3, ]	7	1	1	1
G[4, ]	4	2	1	3

 $= Q_{AC}$ 

$$2*3 + 1*1 + 3*1 + 4*1 = \mathbf{14}$$

```
> mdat %*% mdat
```

	[,1]	[,2]	[,3]	[,4]
[1, ]	42	16	<b>14</b>	28
[2, ]	32	21	12	35
[3, ]	26	13	24	37
[4, ]	29	17	18	36

```
>
```

# Come passare da A a C in due passaggi

```
piero@piero-XPS-9320:~$ octave
```

```
QSocketNotifier: Can only be used with threads started with QThread
```

```
GNU Octave, version 8.3.0
```

```
Copyright (C) 1993-2023 The Octave Project Developers.
```

```
This is free software; see the source code for copying conditions.
```

```
There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or  
FITNESS FOR A PARTICULAR PURPOSE. For details, type 'warranty'.
```

```
Octave was configured for "x86_64-pc-linux-gnu".
```

```
Additional information about Octave is available at https://www.gnu.org.
```

```
Please contribute if you find this software useful.
```

```
For more information, visit https://www.gnu.org/get-involved.html
```

```
Read https://www.gnu.org/bugs.html to learn how to submit bug reports.
```

```
For information about changes from previous versions, type 'news'.
```

```
octave:1>
```

# Come passare da A a C in due passaggi

```
octave:1> a = [2, 1, 3, 4; 1, 3, 1, 5; 7, 1, 1, 1; 4, 2, 1, 3]
a =
```

```
2   1   3   4
1   3   1   5
7   1   1   1
4   2   1   3
```

```
octave:2> a^2
ans =
```

```
42   16   14   28
32   21   12   35
26   13   24   37
29   17   18   36
```

```
octave:3>
```

# Stati transitori e stati definitivi

Gli **stati transitori**, dopo essere stati visitati, possono essere abbandonati mentre da quelli **definitivi** la catena, una volta entrata, non può più uscire. Gli stati definitivi, per cui  $P_{ii} = 1$ , si presentano nella matrice di transizione  $\mathbf{P}$  come altrettanti 1 (uno) collocati nella diagonale principale, mentre in tutti gli altri ingressi ci sono solo zeri.

Per gli stati transitori  $P_{ii} < 1$  cosicché almeno qualche altro ingresso della  $i$ -esima riga è  $> 0$ .

Per definizione, la catena deve trovarsi in uno degli stati possibili ad ogni *step*; per qualsiasi valore di  $i$  e  $j$ ,  $0 \leq P_{ij} \leq 1$  e

$$\sum_{j=1}^n P_{ij} = 1$$

per tutti gli  $i$ , ossia la somma delle probabilità che il processo si sposti dallo stato attuale  $i$  ad uno stato  $j$  qualsiasi (che può anche coincidere con  $i$  stesso) è uguale a 1.

# Il magico numero <e>

```
> 10*9*8*7*6*5*4*3*2*1  
[1] 3628800  
> factorial(10)  
[1] 3628800  
  
> Euler = function(n, z)  
+ {  
+   result = NULL  
+   numbers = seq(1:n)  
+   for (i in numbers)  
+   {  
+     result[i] = (z^i)/factorial(i)  
+   }  
+   return(1 + sum(result))  
+ }  
> Euler(100,1)  
[1] 2.718282  
> Euler(100,10)  
[1] 22026.47  
> exp(10)  
[1] 22026.47
```

$$e = 1 + 1/1! + 1/2! + 1/3! + 1/4! + \dots \cong 2.718282$$

$$e^z = 1 + z/1! + z^2/2! + z^3/3! + z^4/4! + \dots \cong 2.718282$$

Per  $z < 1$ :

$$e^z \cong 1 + z$$

Es.

$$\text{se } z = 0.1$$

$$e^{0.1} \cong 1 + 0.1 \cong 1.1$$

```
> exp(0.1)
```

```
[1] 1.105171
```

```
> Euler(10,0.1)
```

```
[1] 1.105171
```

Da  
 $e^z = 1 + z/1! + z^2/2! + z^3/3! + z^4/4! + \dots \cong 1 + z$   
 per  $z < 1$  (al limite per  $z$  che tende a a)

$$\mathbf{P} = \begin{matrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{32} & p_{33} & p_{34} \\ p_{41} & p_{42} & p_{43} & p_{44} \end{matrix}$$

Si ricava che:

$$e^A = 1 + A/1! + A^2/2! + A^3/3! + A^4/4! + \dots \cong (I + A) = \mathbf{P}$$

$$I + A = \mathbf{P}$$

$$1 \quad 0 \quad 0 \quad 0 \quad a_{11} \quad a_{12} \quad a_{13} \quad a_{14} \quad p_{11} \quad p_{12} \quad p_{13} \quad p_{14}$$

$$0 \quad 1 \quad 0 \quad 0 \quad a_{21} \quad a_{22} \quad a_{23} \quad a_{24} \quad p_{21} \quad p_{22} \quad p_{23} \quad p_{24}$$

$$+$$

$$0 \quad 0 \quad 1 \quad 0 \quad a_{31} \quad a_{32} \quad a_{33} \quad a_{34} \quad p_{31} \quad p_{32} \quad p_{33} \quad p_{34}$$

$$0 \quad 0 \quad 0 \quad 1 \quad a_{41} \quad a_{42} \quad a_{43} \quad a_{44} \quad p_{41} \quad p_{42} \quad p_{43} \quad p_{44}$$