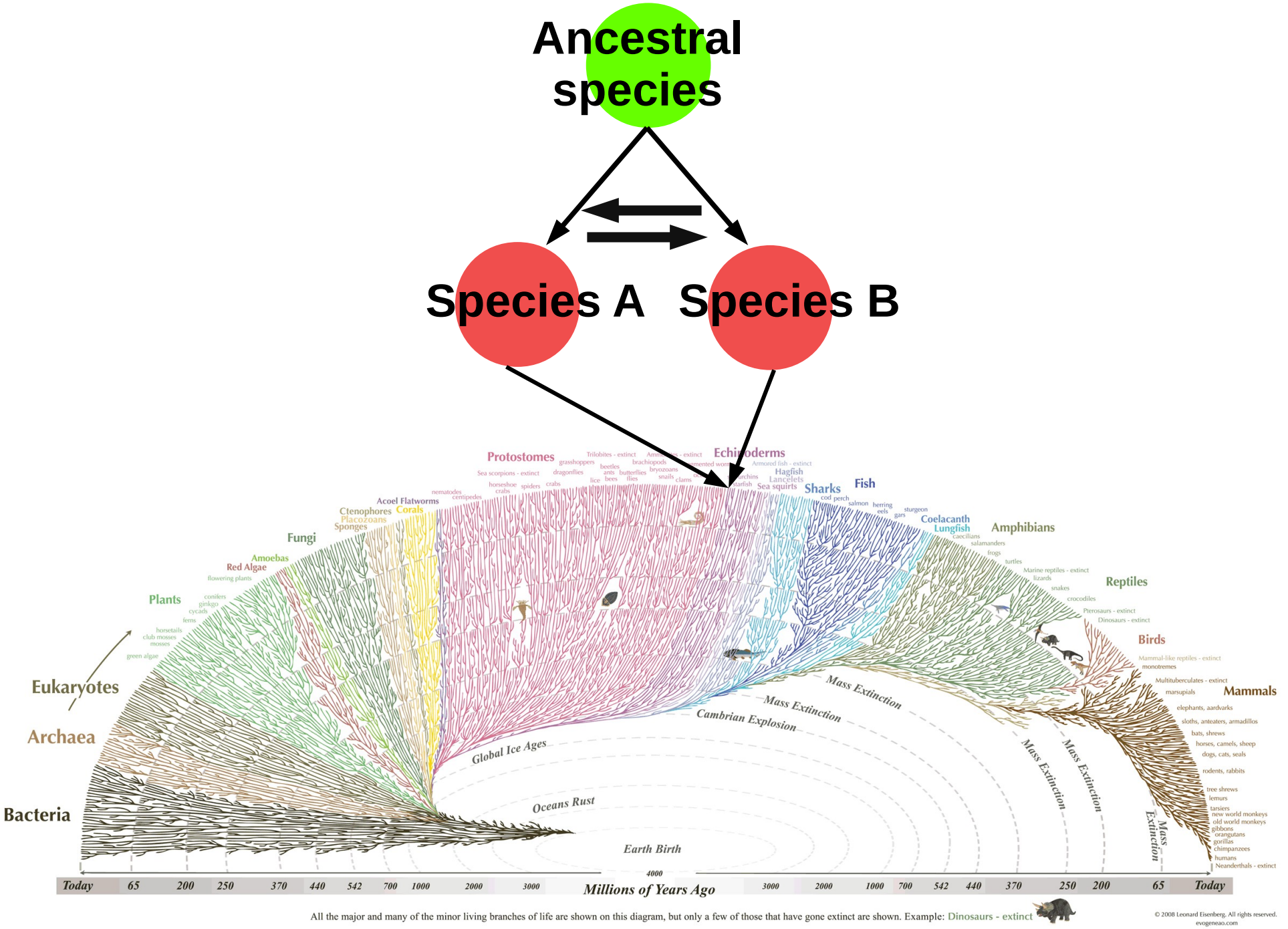
Before starting :
1) open your terminal

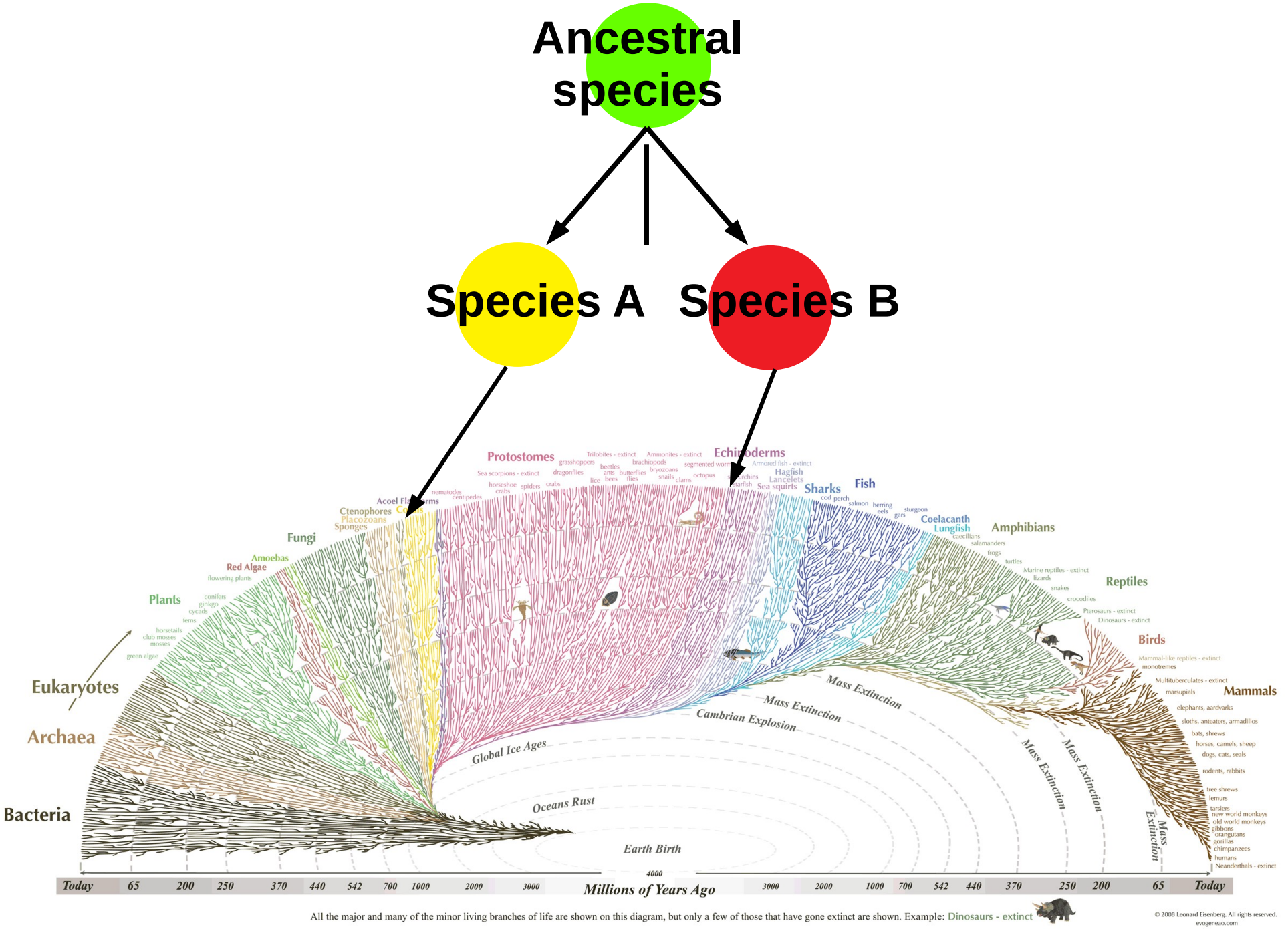
2) go to your directory named **ABC_WGD** obtained from git-hub

3) tape **git pull**

Speciation

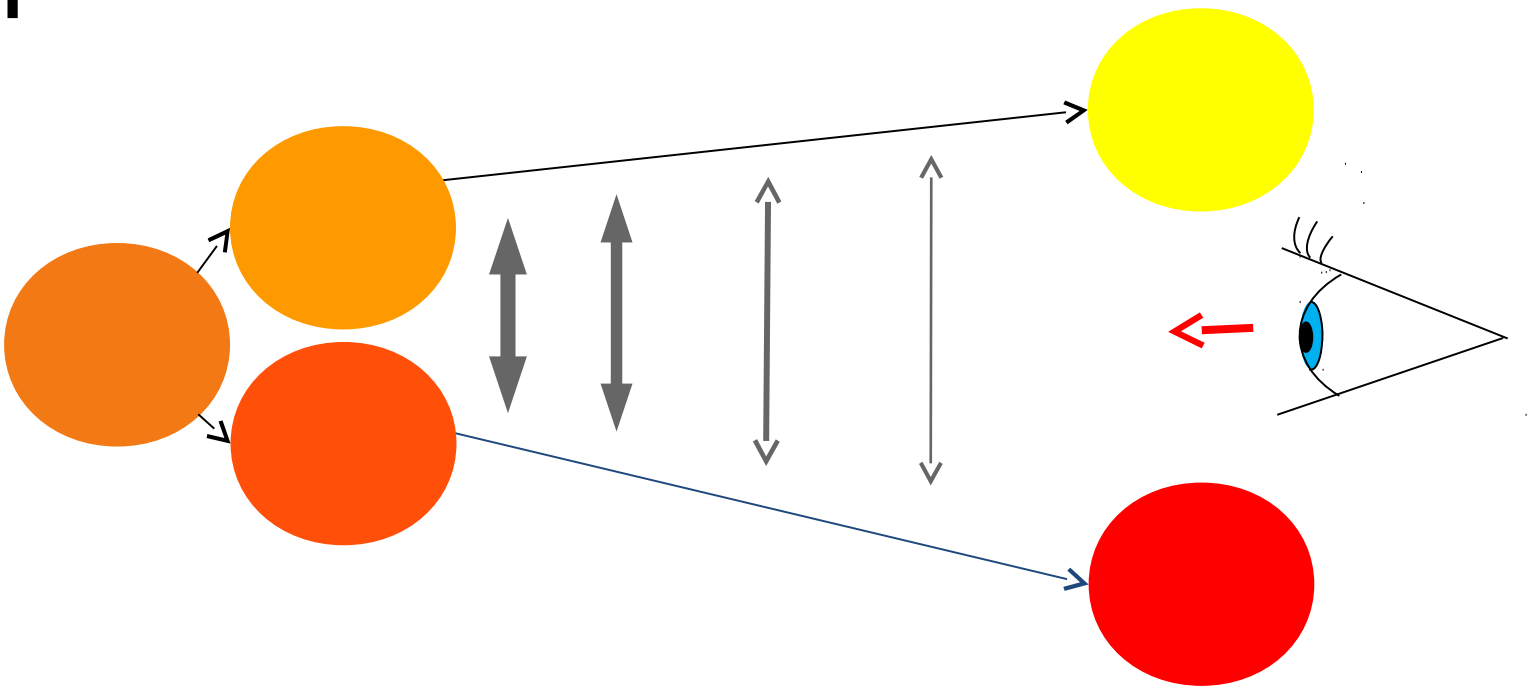

Ancestral species
Species A
Species B
Protostomes
Echinoderms
Hagfish
Lancelets
Sea squirts
Sharks
Fish
Coelacanth
Lungfish
Amphibians
Reptiles
Birds
Mammals
Acoel Flatworms
Ctenophores
Corals
Placozoans
Sponges
Fungi
Amoebas
Red Algae
Plants
Eukaryotes
Archaea
Bacteria
Cambrian Explosion
Mass Extinction
Mass Extinction
Mass Extinction
Mass Extinction
Mass Extinction
Global Ice Ages
Oceans Rust
Earth Birth
Today
65
200
250
370
440
542
700
1000
2000
3000
4000
Millions of Years Ago
All the major and many of the minor living branches of life are shown on this diagram, but only a few of those that have gone extinct are shown. Example: Dinosaurs - extinct
© 2008 Leonard Eisenberg. All rights reserved.
evogeneao.com

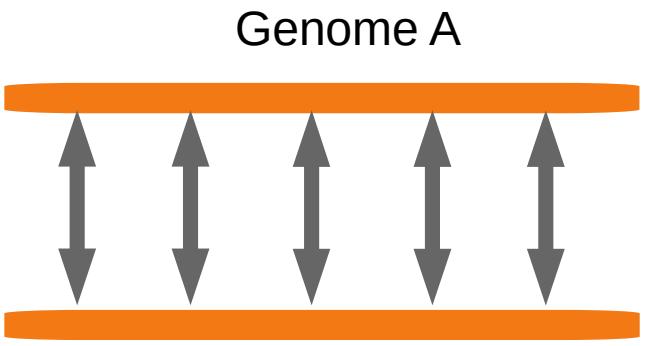
Speciation



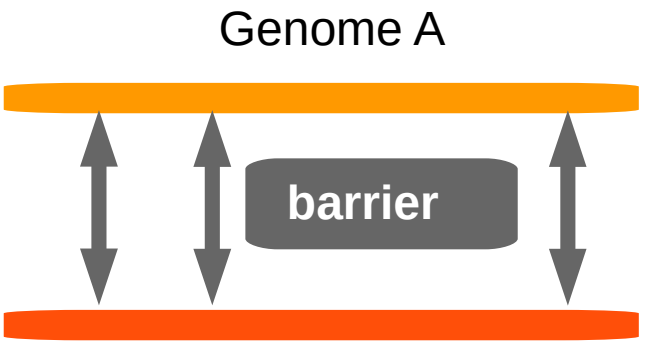
All the major and many of the minor living branches of life are shown on this diagram, but only a few of those that have gone extinct are shown. Example: Dinosaurs - extinct

Speciation

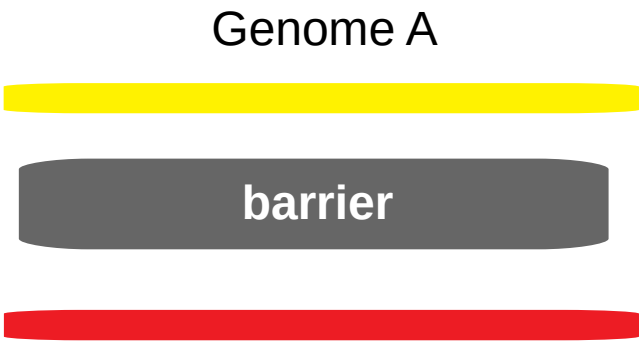




2 populations from the same species

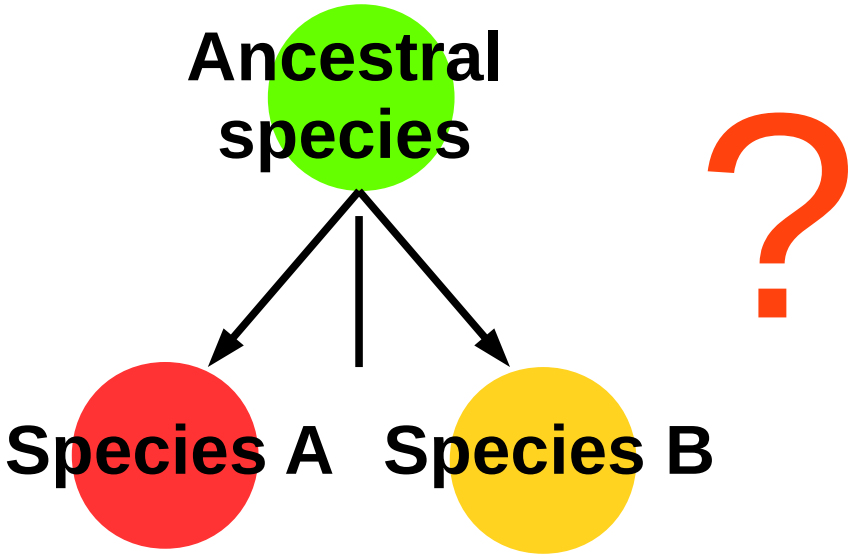


2 semi-isolated species



2 isolated species

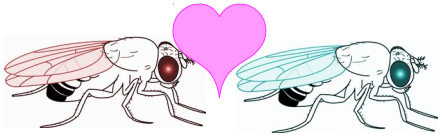
Q1 : What is the nature of species barriers ?



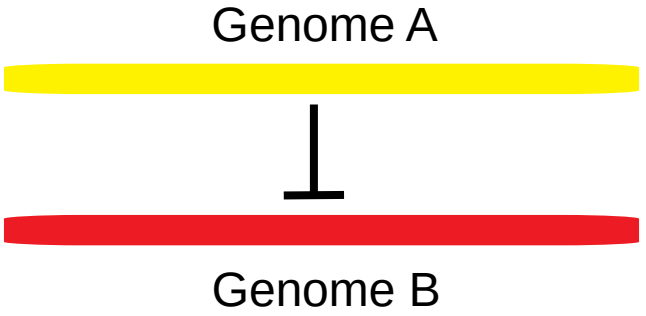
Ecological ?

Environnement

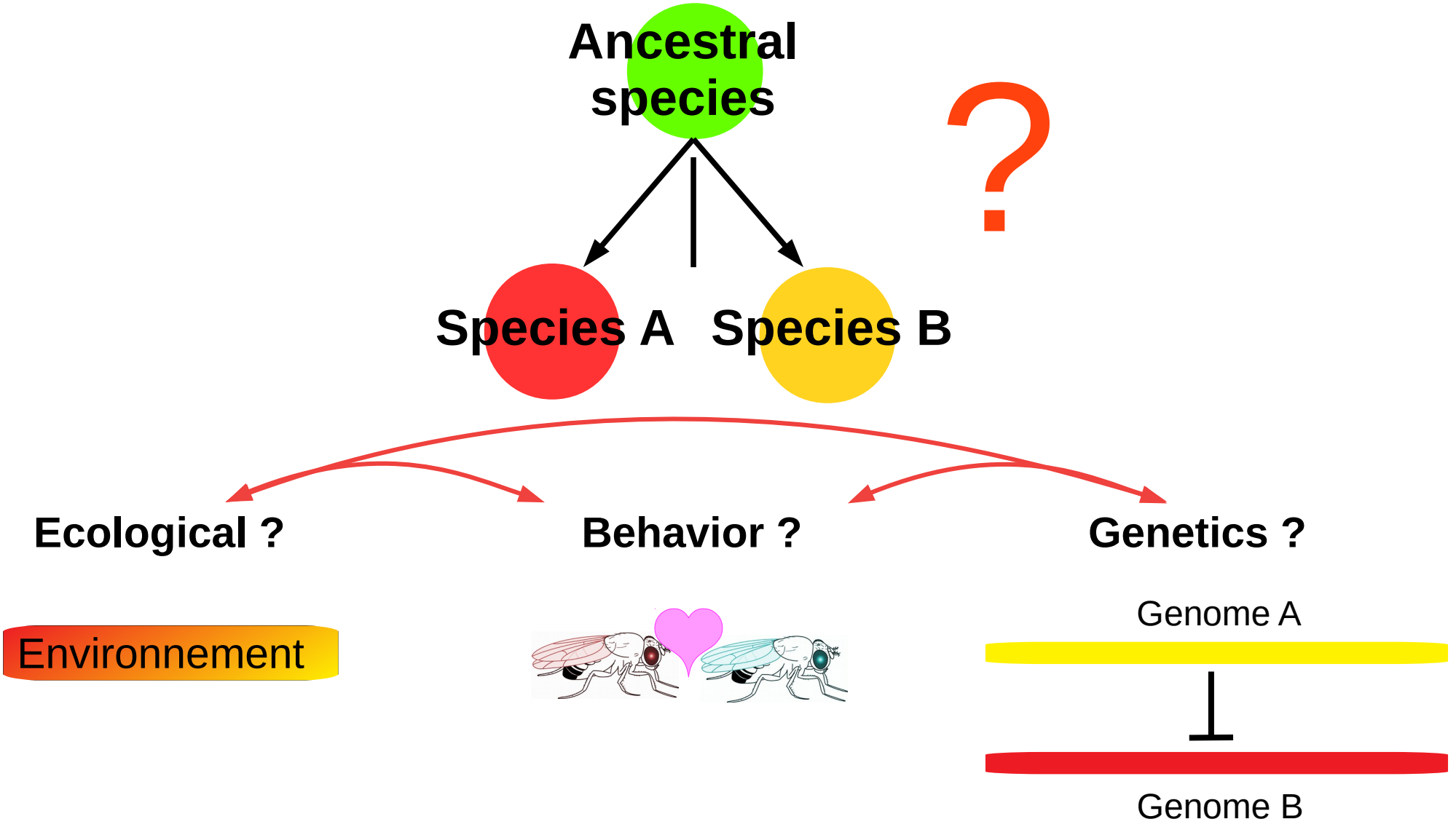
Behavior ?



Genetics ?

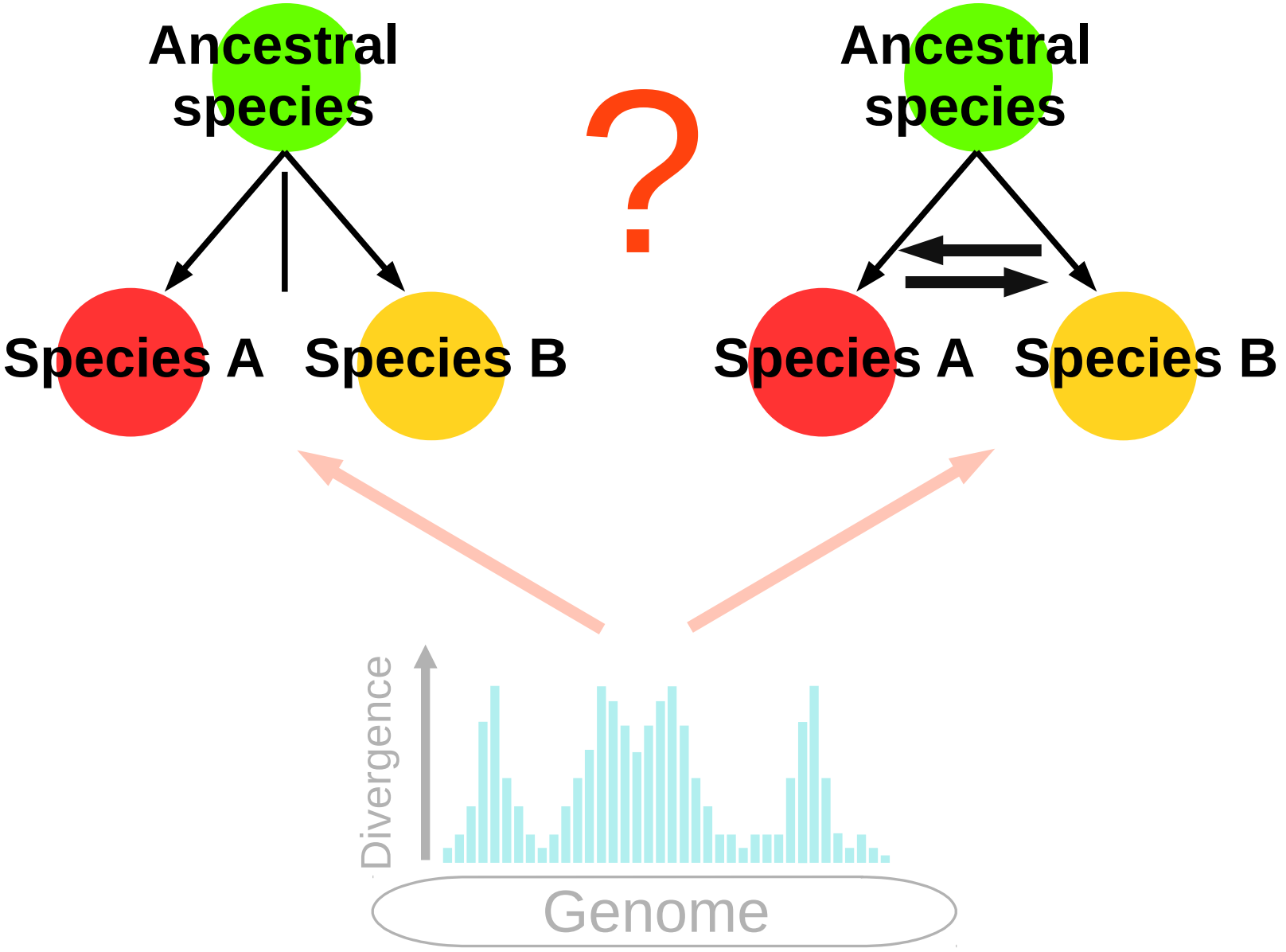


Q1 : What is the nature of species barriers ?



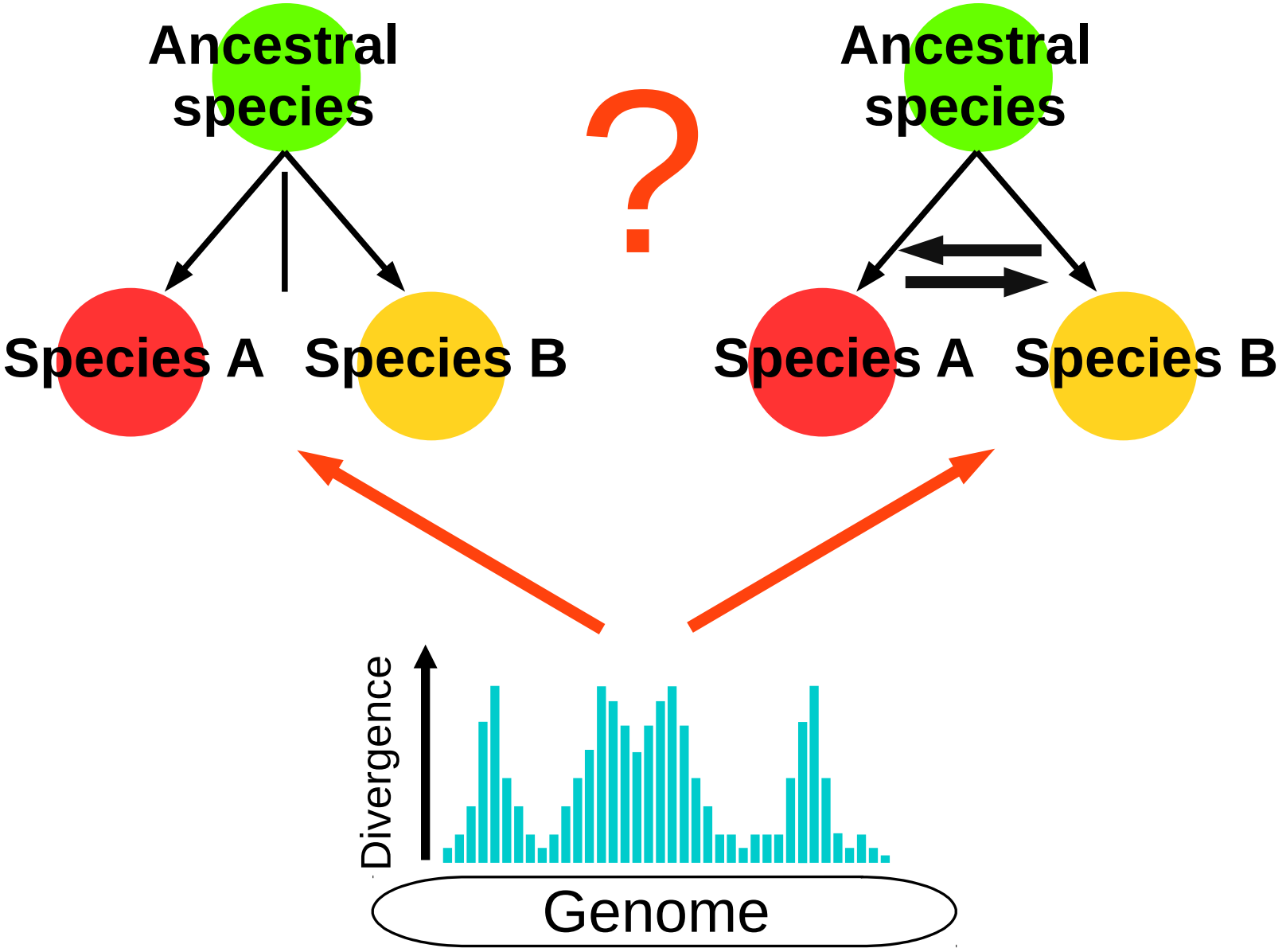
Q1 : What is the nature of species barriers ?

Q2 : What is the historical context in which barriers have been appeared ?

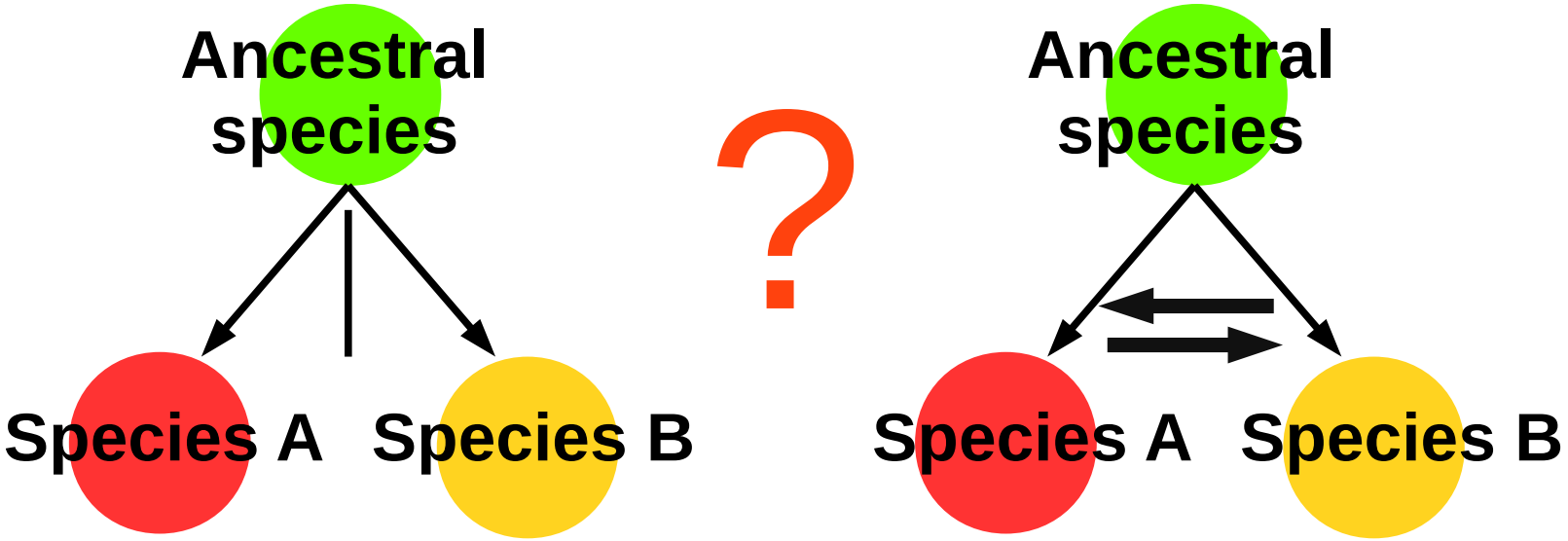


Q1 : What is the nature of species barriers ?

Q2 : What is the historical context in which barriers have been appeared ?



Goal of the day: Comparing alternative scenarios of demographic history



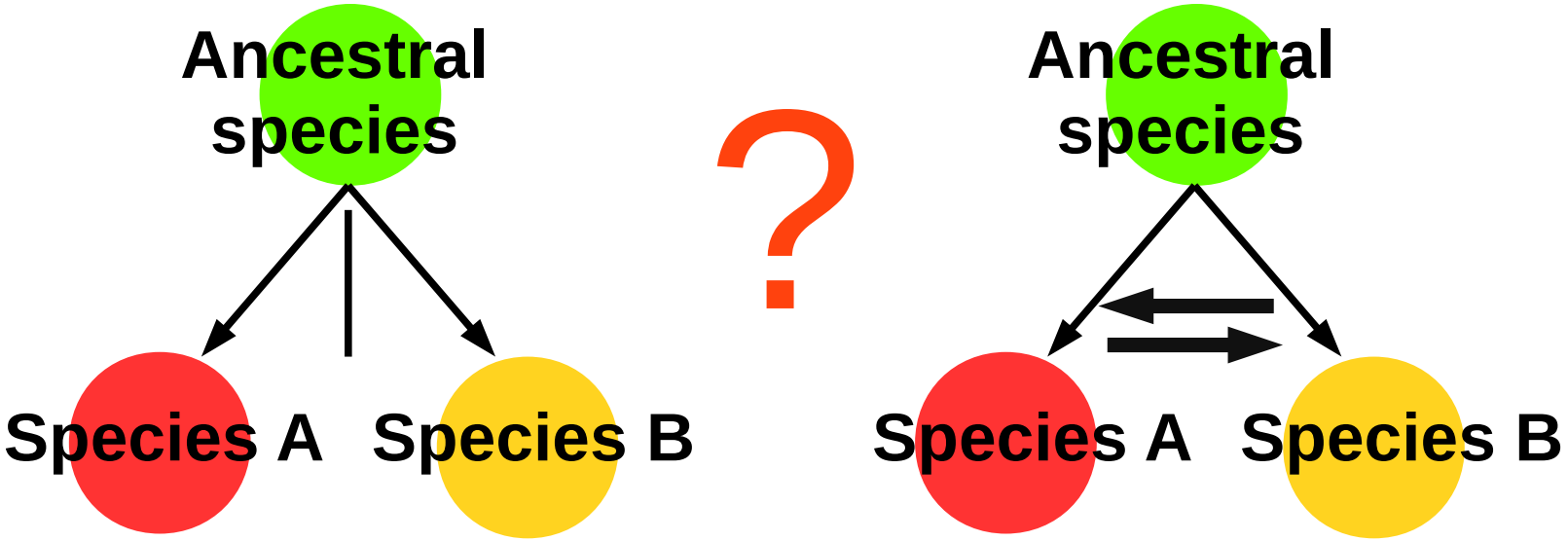
Migration *versus* isolation ?

Disomic *versus* tetrasomic *versus* heterosomic ?

Autopolyploid *versus* allopolyploid ?

Bottleneck *versus* constant population size ?

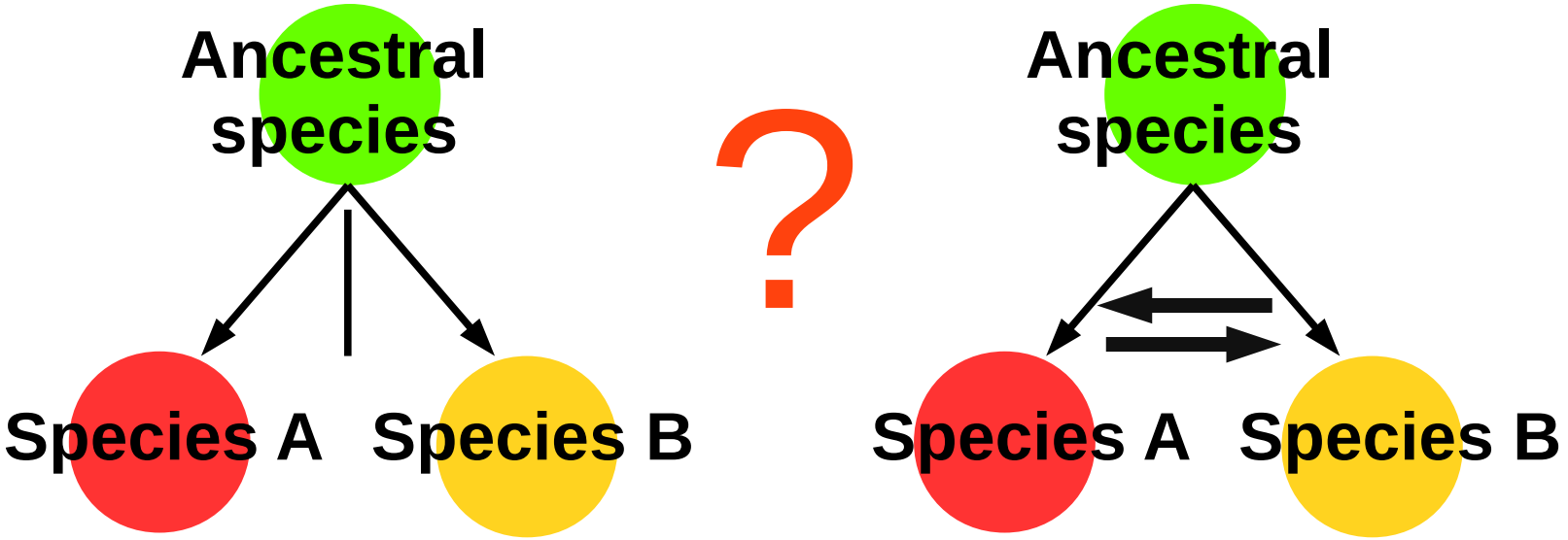
Goal of the day: Comparing alternative scenarios of demographic history



1) Basic Bayesian inferences

2) Approximate Bayesian Computation (ABC)

Goal of the day: Comparing alternative scenarios of demographic history



1) Basic Bayesian inferences

2) Approximate Bayesian Computation (ABC)

Main goal : understanding the method, not (only) the tool !

Basic Bayesian inferences

Set of 5 dice

4-sided **6-sided** **8-sided** **12-sided** **20-sided**

1) I randomly take a die among the 5
2) I throw it twice
3) observed 7 and 8

Q : what are the probabilities of each die based on these observations?

Basic Bayesian inferences

Set of 5 dice

4-sided

6-sided

8-sided

12-sided

20-sided

1) I randomly take a die among the 5
2) I throw it twice
3) observed 7 and 8

Q : what are the probabilities of each die based on these observations?

Q : what are the probabilities of each die based on these observations?

From the first observation : $X = 7$

$$P(\text{die},i \mid 7) = [\, P(7 \mid \text{die},i) \times P(\text{die},i) \,] \,/\, P(7)$$

probability of the die *i* given that the event '7' occurred

Q : what are the probabilities of each die based on these observations?

From the first observation : $X = 7$

$$P(\text{die},i \mid 7) = [\, P(7 \mid \text{die},i) \times P(\text{die},i) \,] \,/\, P(7)$$

i in [4, 6, 8, 12, 20]

probability of the die i given that the event '7' occurred

Q : what are the probabilities of each die based on these observations?

From the first observation : $X = 7$

$$P(\text{die},i \mid 7) = [P(7 \mid \text{die},i) \times P(\text{die},i)] / P(7)$$

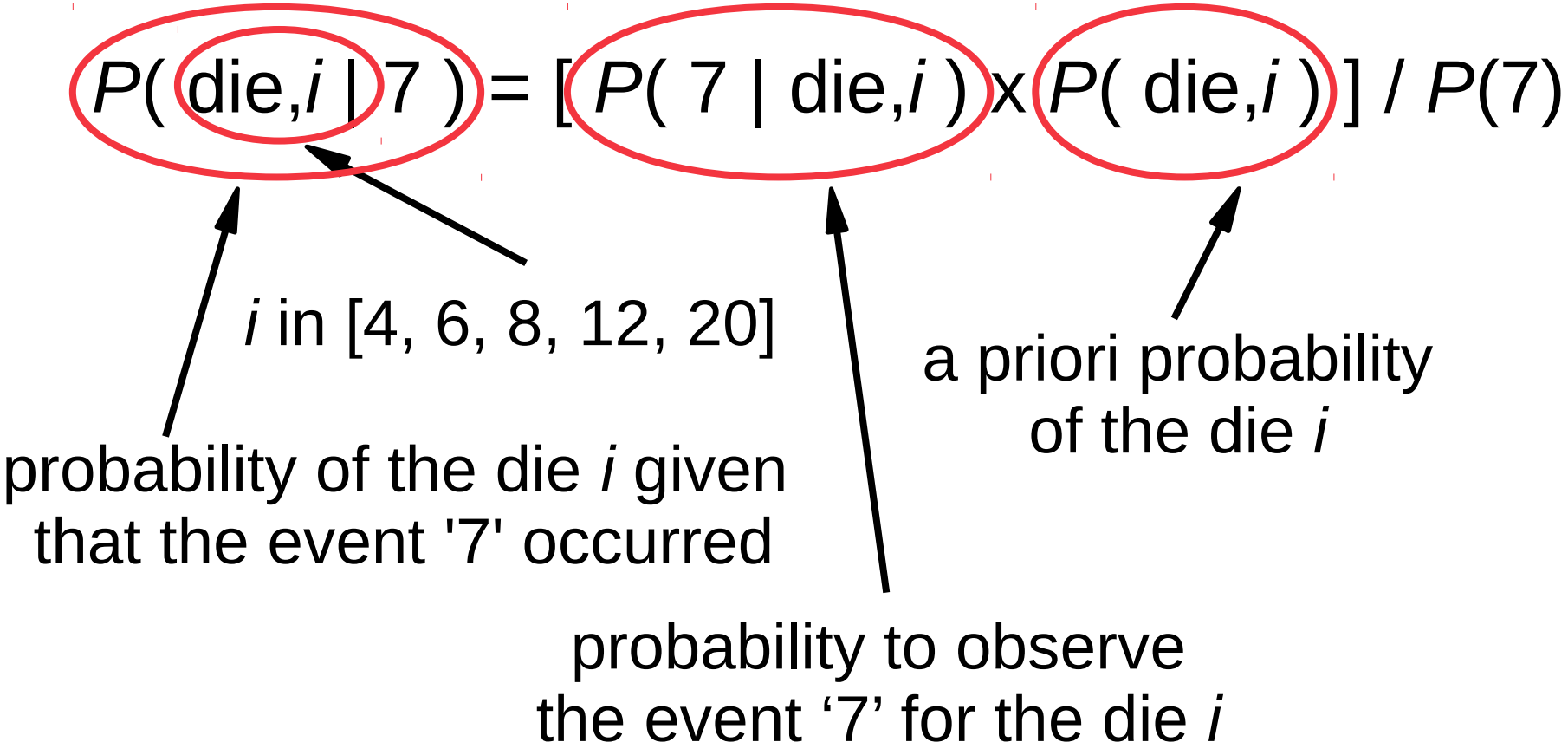
i in [4, 6, 8, 12, 20]

probability of the die i given that the event '7' occurred

probability to observe the event ‘7’ for the die i

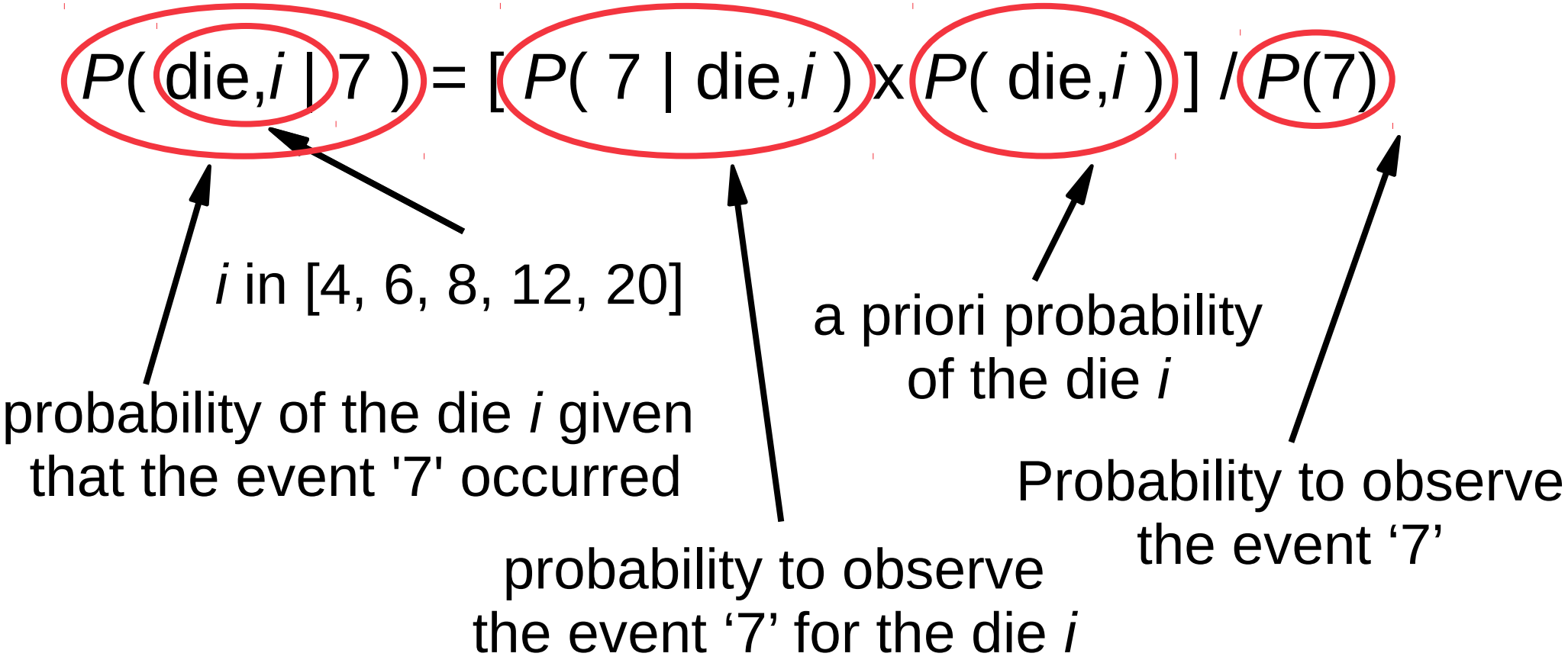
Q : what are the probabilities of each die based on these observations?

From the first observation : $X = 7$



Q : what are the probabilities of each die based on these observations?

From the first observation : $X = 7$



Q : what are the probabilities of each die based on these observations?

From the first observation : $X = 7$

$$P(\text{die},i \mid 7) = [\, P(7 \mid \text{die},i) \times P(\text{die},i) \,] \,/\, P(7)$$

?

?

?

?

?

Q : what are the probabilities of each die based on these observations?

From the first observation : $X = 7$

$$P(\text{die},i \mid 7) = [\, P(7 \mid \text{die},i) \times P(\text{die},i) \,] \,/\, P(7)$$

? 0

? 0

?

?

?

Q : what are the probabilities of each die based on these observations?

From the first observation : $X = 7$

$$P(\text{ die},i \mid 7) = [\ P(\ 7 \mid \text{die},i\ ) \times P(\text{ die},i\ )\ ] \ /\ P(7)$$

? 0

? 0

? 1/8

? 1/12

? 1/20

Q : what are the probabilities of each die based on these observations?

From the first observation : $X = 7$

$$P(\text{ die},i \mid 7) = [\ P(\ 7 \mid \text{die},i\ ) \times P(\ \text{die},i\ )\ ] \ /\ P(7)$$

? 0 1/5

? 0 1/5

? 1/8 1/5

? 1/12 1/5

? 1/20 1/5

Q : what are the probabilities of each die based on these observations?

From the first observation : $X = 7$

$P(\text{ die},i \mid 7) = [\ P(\ 7 \mid \text{die},i\ ) \times P(\ \text{die},i\ )\ ] \ /\ P(7)$

? 0 1/5

? 0 1/5

? 1/8 1/5

? 1/12 1/5

? 1/20 1/5

$$P(7) = \sum_{i=1} P(i).P(7|i)$$

Q : what are the probabilities of each die based on these observations?

From the first observation : $X = 7$

$$P(\text{die},i \mid 7) = [\, P(7 \mid \text{die},i) \times P(\text{die},i) \,] \,/\, P(7)$$

	?	0	1/5	
	?	0	1/5	
	?	1/8	1/5	0.052
	?	1/12	1/5	
	?	1/20	1/5	

In R : sum(rep(1/5, 5) * c(0,0, 1/8, 1/12, 1/20)) ~ 0.052

Q : what are the probabilities of each die based on these observations?

From the first observation : $X = 7$

$$P(\text{die},i \mid 7) = [\, P(7 \mid \text{die},i) \times P(\text{die},i) \,] \,/\, P(7)$$

$0 * 1/5 / P(7) = 0$ 0 1/5

$0 * 1/5 / P(7) = 0$ 0 1/5

$1/8 * 1/5 / P(7) = 48.39\%$ 1/8 1/5 0.052

$1/12 * 1/5 / P(7) = 32.26\%$ 1/12 1/5

$1/20 * 1/5 / P(7) = 19.35\%$ 1/20 1/5

Q : what are the probabilities of each die based on these observations?

From the second observation : $X = 8$

$P(\text{die},i \mid 7) = [\, P(7 \mid \text{die},i) \times P(\text{die},i) \,] / P(7)$

0 * 1/5 / P(7)=0

0 * 1/5 / P(7)=0

1/8 * 1/5 / P(7)=48.39%

1/12 * 1/5 / P(7)=32.26%

1/20 * 1/5 / P(7)=19.35%

How these probabilities evolve taking into account the second observation X=8?

Q : what are the probabilities of each die based on these observations?

From the second observation : $X = 8$

$$P(\text{ die},i \mid 8) = [\ P(\ 8 \mid \text{die},i\ ) \times P(\text{ die},i\ )\ ] \ /\ P(8)$$

? 0

? 0

? 1/8

? 1/12

? 1/20

Q : what are the probabilities of each die based on these observations?

From the second observation : $X = 8$

$$P(\text{ die},i \mid 8) = [\ P(\ 8 \mid \text{die},i\ ) \times P(\ \text{die},i\ )\ ] \ /\ P(8)$$

? 0 1/5 → 0

? 0 1/5 → 0

? 1/8 1/5 → 0.4839

? 1/12 1/5 → 0.3226

? 1/20 1/5 → 0.1935

Q : what are the probabilities of each die based on these observations?

From the second observation : $X = 8$

$$P(\text{die},i \mid 8) = [\, P(8 \mid \text{die},i) \times P(\text{die},i) \,] \,/\, P(8)$$

	?	0	0
	?	0	0
	?	1/8	0.4839
	?	1/12	0.3226
	?	1/20	0.1935

In R : sum(c(0,0,0.4839,0.3226,0.1935) * c(0,0,1/8,1/12,1/20))

Q : what are the probabilities of each die based on these observations?

From the second observation : $X = 8$

$$P(\text{die},i \mid 8) = [\, P(8 \mid \text{die},i) \times P(\text{die},i) \,] \,/\, P(8)$$

0

0

0

0

48.39%

62.33 %

32.26%

27.7 %

19.35%

9.97 %

Q : what are the probabilities of each die based on these observations?

From the second observation : $X = 8$

$$P(\text{die},i \mid 8) = [\, P(8 \mid \text{die},i) \times P(\text{die},i) \,] \,/\, P(8)$$

0

0

0

0

48.39%

62.33 %

32.26%

27.7 %

19.35%

9.97 %

Here we have N=2 observations

How probabilities evolve with the number of observations ?

Exploring the effects of the number of observations on inferences using the R-function **bayes**

In R :source('comparison_ABC_bayes.R')
bayes(observations=c(7, 8), dies=c(4, 6, 8, 12, 20))

The bayes function returns the probability of each of the proposed die given a series of observations.

```
bayes(observations=c(7, 8), dices=c(4, 6, 8, 12, 20))
4 6       8       12       20 best_model post_proba
0 0 0.62327 0.27701 0.09972          D8    0.62327
```

Exploring the effects of the number of observations on inferences using the R-function **bayes**

In R :source('comparison_ABC_bayes.R')
bayes(observations=c(7, 8), dies=c(4, 6, 8, 12, 20))

observations = vector containing a serie of *N* observations for a given die *i*.

dies = prior distribution of the model comparison. Vector containing a serie of *d* possible die.

Exploring the effects of the number of observations on inferences using the R-function **bayes**

In R :source('comparison_ABC_bayes.R')

bayes(observations=c(7, 8), dies=c(4, 6, 8, 12, 20))

observations = vector containing a serie of *N* observations for a given die *i*.

dies = prior distribution of the model comparison. Vector containing a serie of *d* possible die.

Bonus : simulating a die roll by using random sampling from a discrete uniform distribution (R function : sample)

```
N = 10
observations = sample(1:8, N, replace=T)
```

Exploring the effects of the number of observations on inferences using the R-function **bayes**

In R :source('comparison_ABC_bayes.R')
bayes(observations=c(7, 8), dies=c(4, 6, 8, 12, 20))

observations = vector containing a serie of *N* observations for a given die *i*.

dies = prior distribution of the model comparison. Vector containing a serie of *d* possible die.

```
> real_dice = 8
> N_observations = 3
> observations = sample(1:real_dice, N_observations, replace=T)
>
> bayes(observations = observations, dices = c(4, 6, 8, 12, 20))
  4 6       8       12       20 best_model post_proba
1 0 0 0.73513 0.21782 0.04705         D8    0.73513
```

Exploring the effects of the number of observations on inferences using the R-function **bayes**

In R :source('comparison_ABC_bayes.R')
bayes(observations=c(7, 8), dies=c(4, 6, 8, 12, 20))

observations = vector containing a serie of *N* observations for a given die *i*.

dies = prior distribution of the model comparison. Vector containing a serie of *d* possible die.

```
> real_dice = 8
> N_observations = 3
> observations = sample(1:real_dice, N_observations, replace=T)
>
> bayes(observations = observations, dices = c(4, 6, 8, 12, 20))
  4 6       8       12       20 best_model post_proba
1 0 0 0.73513 0.21782 0.04705          D8     0.73513
```

Correct inference !

Exploring the effects of the number of observations on inferences using the R-function **bayes**

In R :source('comparison_ABC_bayes.R')
bayes(observations=c(7, 8), dies=c(4, 6, 8, 12, 20))

observations = vector containing a serie of *N* observations for a given die *i*.

dies = prior distribution of the model comparison. Vector containing a serie of *d* possible die.

```
> real_dice = 8
> N_observations = 3
> observations = sample(1:real_dice, N_observations, replace=T)
> bayes(observations = observations, dices = c(4, 6, 8, 12, 20))
        4       6       8      12      20 best_model post_proba
1 0.68197 0.20207 0.08525 0.02526 0.00546         D4    0.68197
> observations
[1] 3 1 1
```

Bad inference

Exploring the effects of the number of observations on inferences using the R-function **bayes**

In R :source('comparison_ABC_bayes.R')
bayes(observations=c(7, 8), dies=c(4, 6, 8, 12, 20))

observations = vector containing a serie of *N* observations for a given die *i*.

dies = prior distribution of the model comparison. Vector containing a serie of *d* possible die.

Bad inference

```
> real_dice = 8
> N_observations = 3
> observations = sample(1:real_dice, N_observations, replace=T)
> bayes(observations = observations, dices = c(4, 6, 8, 12, 20))
        4       6       8      12      20 best_model post_proba
1 0.68197 0.20207 0.08525 0.02526 0.00546         D4    0.68197
> observations
[1] 3 1 1
```

Before applying ANY inference method

Before applying ANY inference method

know the limits of the method!

Before applying ANY inference method

know the limits of the method!

What is the error rate ?

When is the method wrong?

Exploring the effects of the number of observations on inferences using the R-function **bayes**

In R : **source('comparison_ABC_bayes.R')**

bayes(observations=c(7, 8), dies=c(4, 6, 8, 12, 20))

observations = **vector containing a serie of *N* observations for a given die *i*. Can be obtained as follow :**
observations = sample(1:8, N, replace=T)

dies = **prior distribution of the model comparison. Vector containing a serie of *d* possible die.**

Measure empirically (using loops in R) the inference errors for:

. the die of your choice

. different values of N

How much the sampling size affects inferences ?

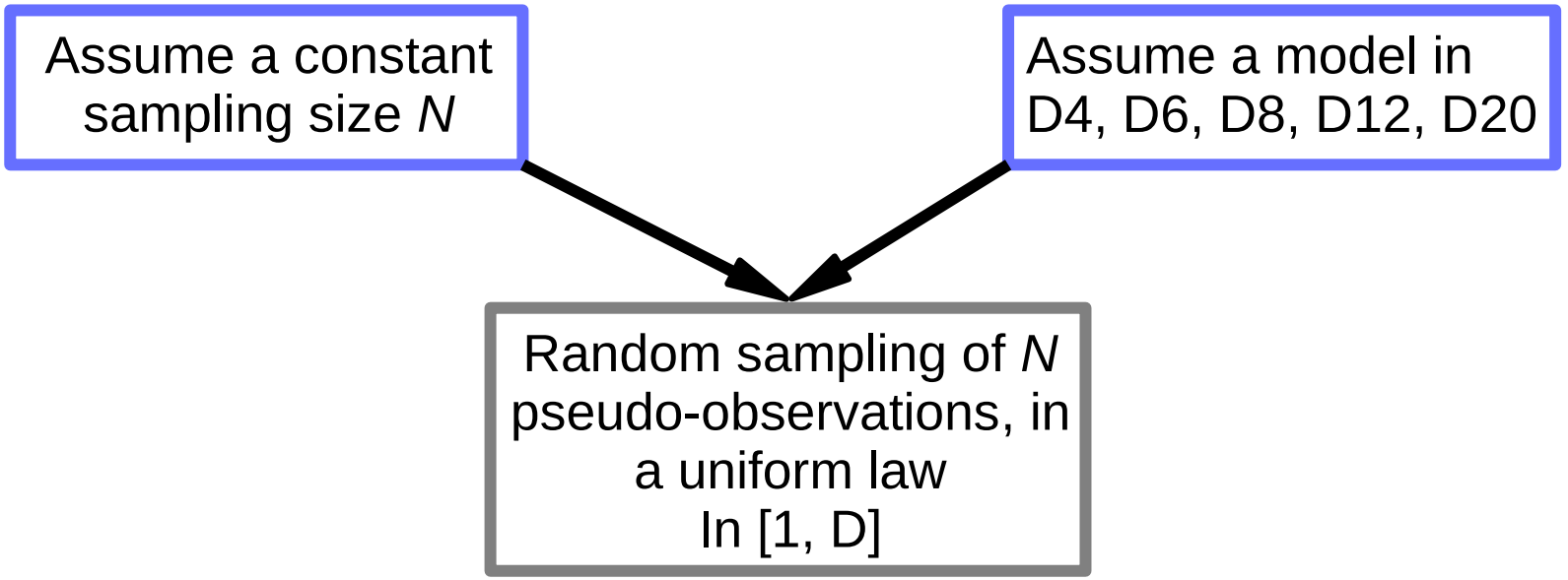
How alternative models compete each others ?

Assume a constant sampling size *N*

Assume a model in D4, D6, D8, D12, D20

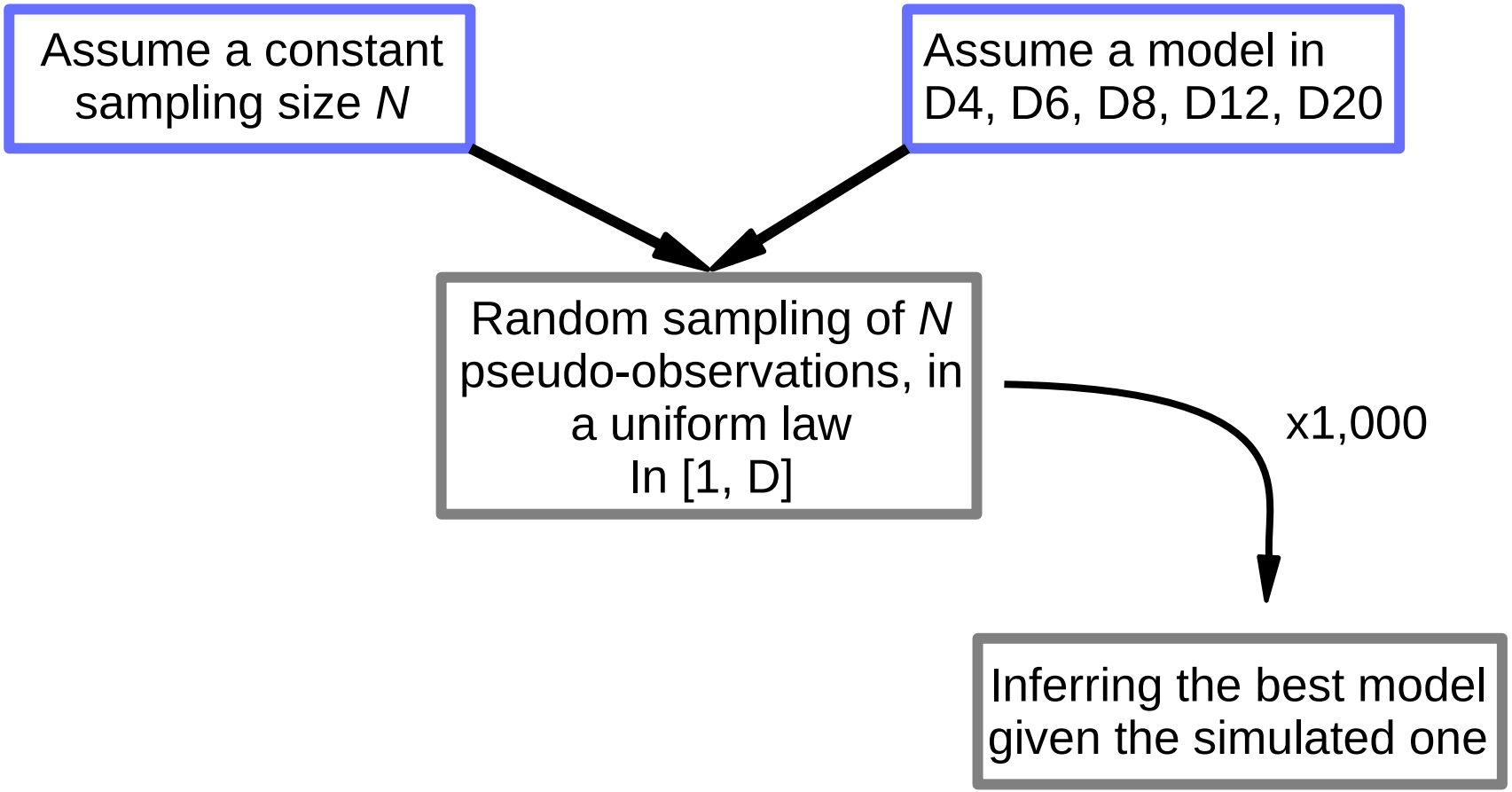
How much the sampling size affects inferences ?

How alternative models compete each others ?



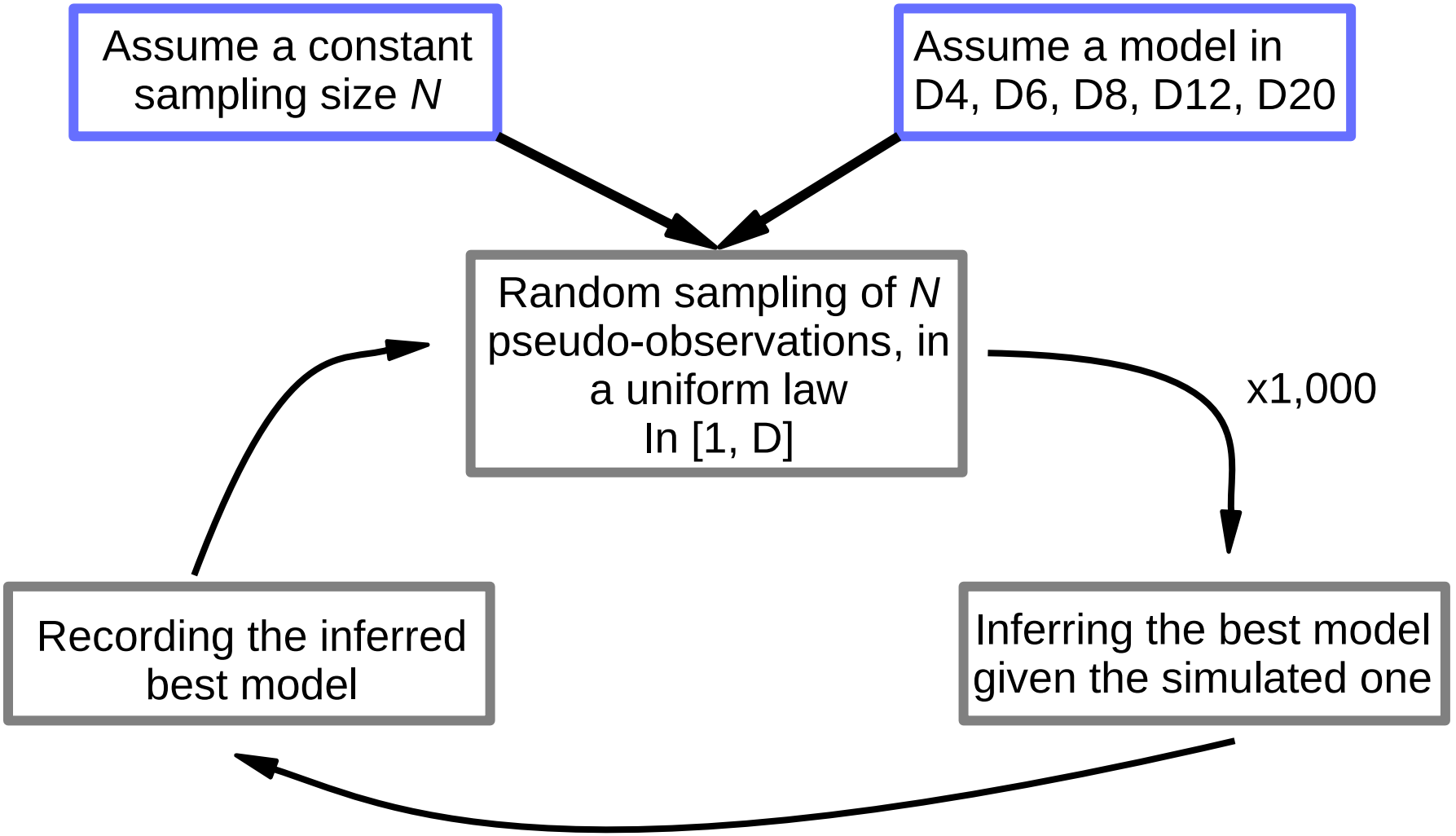
How much the sampling size affects inferences ?

How alternative models compete each others ?



How much the sampling size affects inferences ?

How alternative models compete each others ?



How much the sampling size affects inferences ?

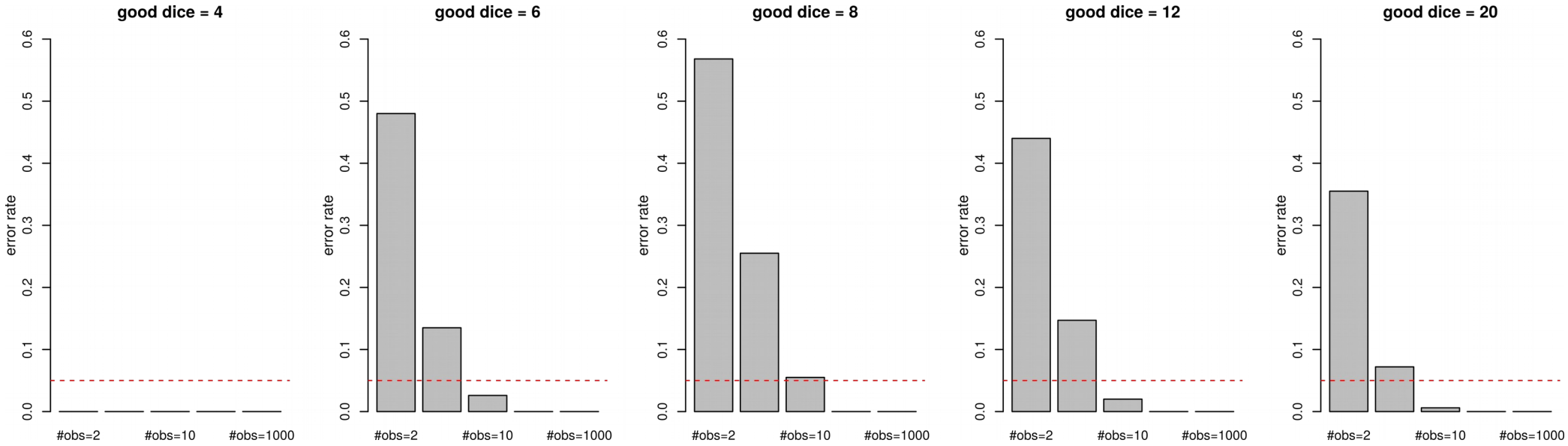
How alternative models compete each others ?

In R :source('comparison_ABC_bayes.R')
plot_bayes_error(good_die=8, nReplicates=1000)
Function made to show the error rate over Bayesian inferences

good_die = the correct die for which we know the identity that we will try to recapture using classic Bayesian inference

nReplicates = number of tests over which we will measure an error rate

How much the sampling size affects inferences ?

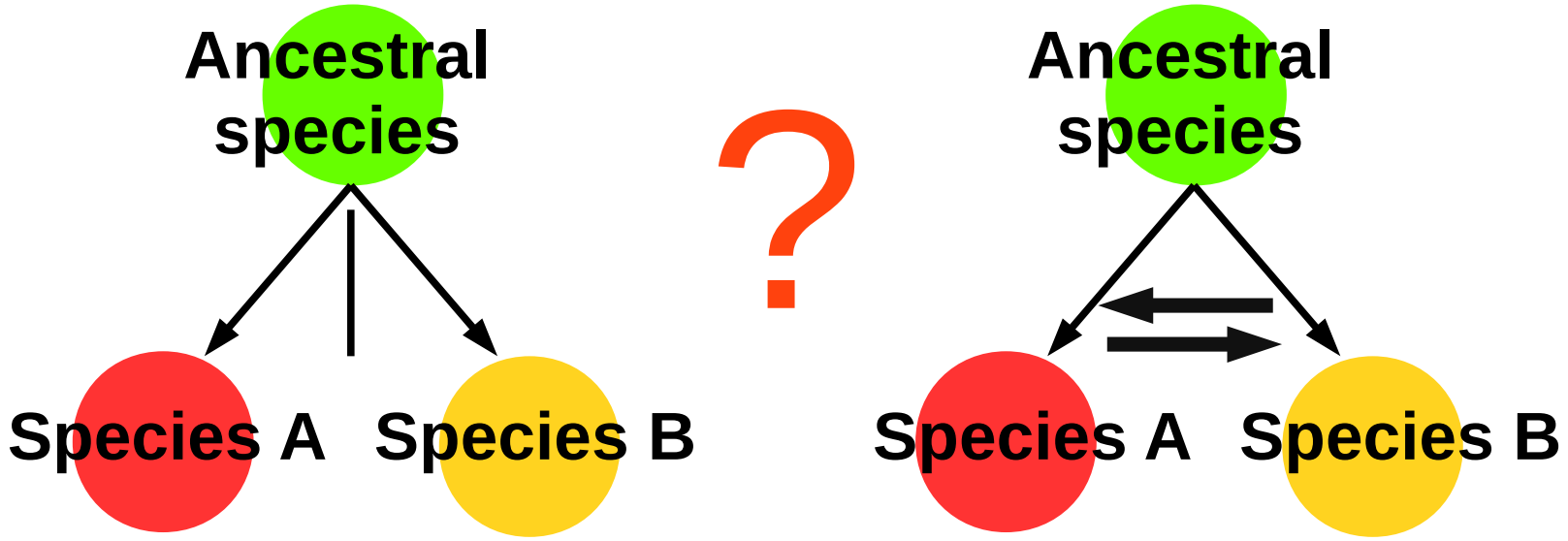


?

$P(\text{Model},i \mid \text{Data}) = [P(\text{Data} \mid \text{Model},i) \times P(\text{Model},i)] / P(\text{Data})$

Problem : the exact likelihood function is not available for all models

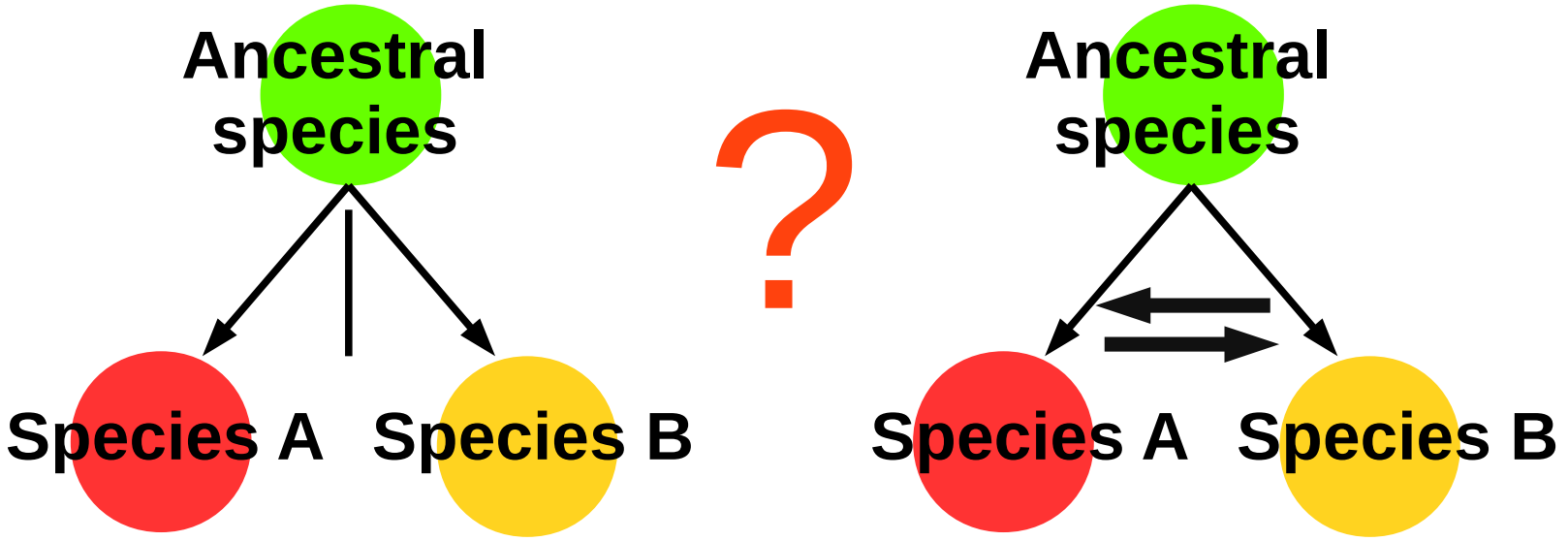
Approximated by ABC (approximate Bayesian computation)



$P(\text{Model},i \mid \text{Data}) = [P(\text{Data} \mid \text{Model},i) \times P(\text{Model},i)] / P(\text{Data})$

Problem : the exact likelihood function is not available for all models

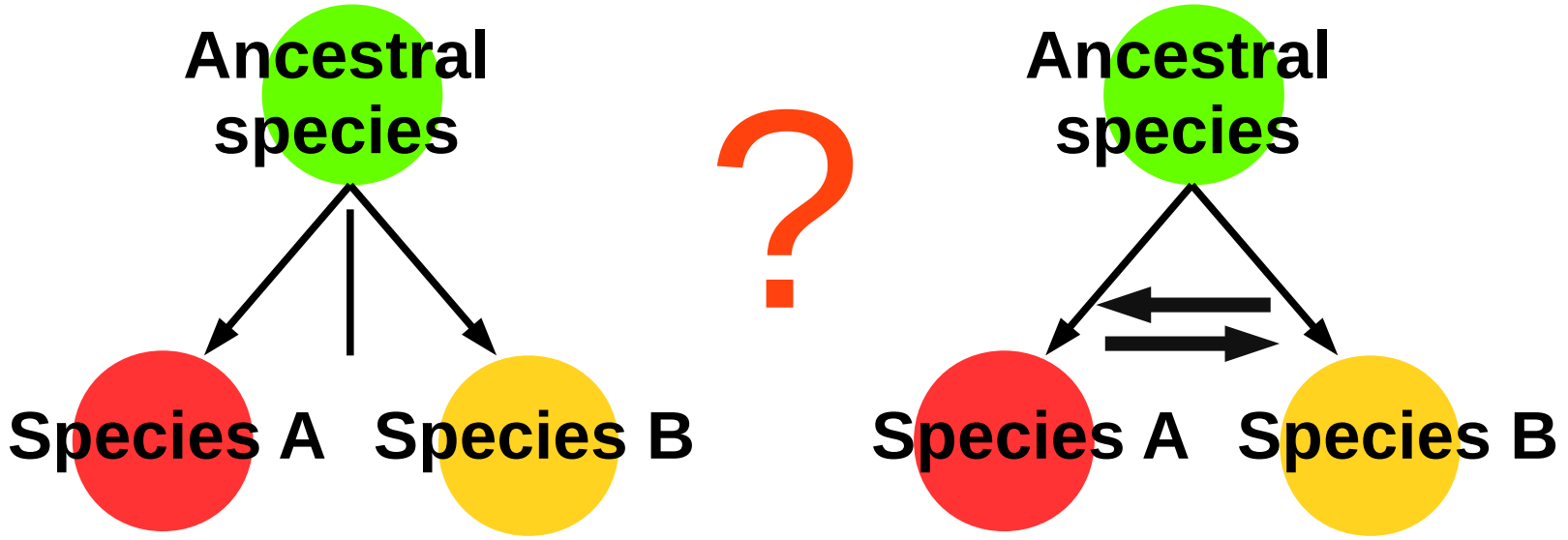
Approximated by ABC (approximate Bayesian computation)



$P(\text{Model},i \mid \text{Data}) = [P(\text{Data} \mid \text{Model},i) \times P(\text{Model},i)] / P(\text{Data})$

Problem : the exact likelihood function is not available for all models

Approximated by ABC (approximate Bayesian computation)



$P(\text{Model},i \mid \text{Data}) = [P(\text{Data} \mid \text{Model},i) \times P(\text{Model},i)] / P(\text{Data})$

Problem : the exact likelihood function is not available for all models

Approximated by ABC (approximate Bayesian computation)

ABC in 3 steps

1) summarizing the data with descriptive statistics

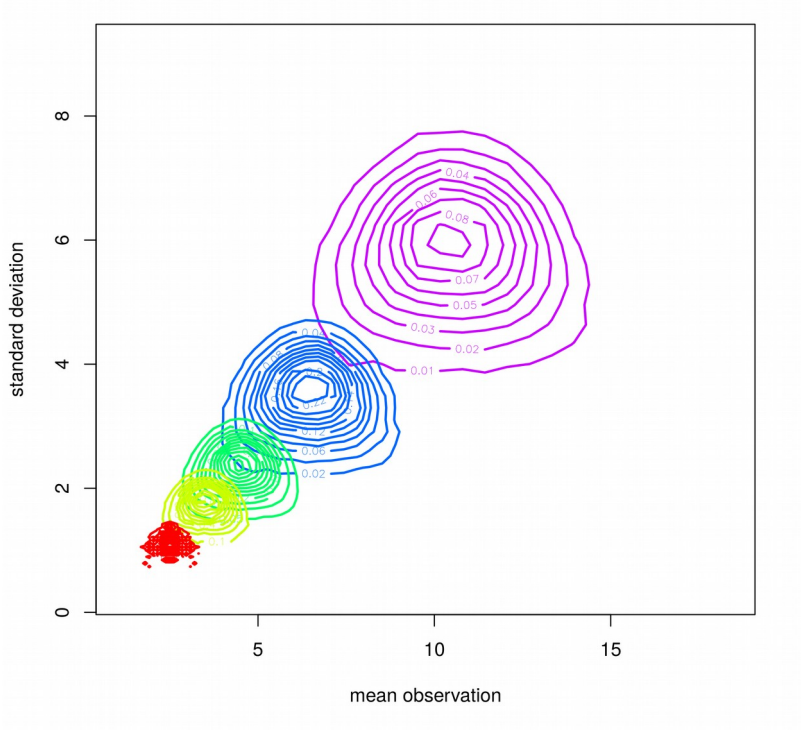

```
data = sample( 1:8, 10, replace = T )
print(data)
[1] 5 4 6 6 1 7 3 4 2 2
```

ABC in 3 steps

1) summarizing the data with descriptive statistics

```
data = sample( 1:8, 10, replace = T )
print(data)
[1] 5 4 6 6 1 7 3 4 2 2
```

2) random simulations of the statistics under the compared models



D20

D12

D8

D6

D4

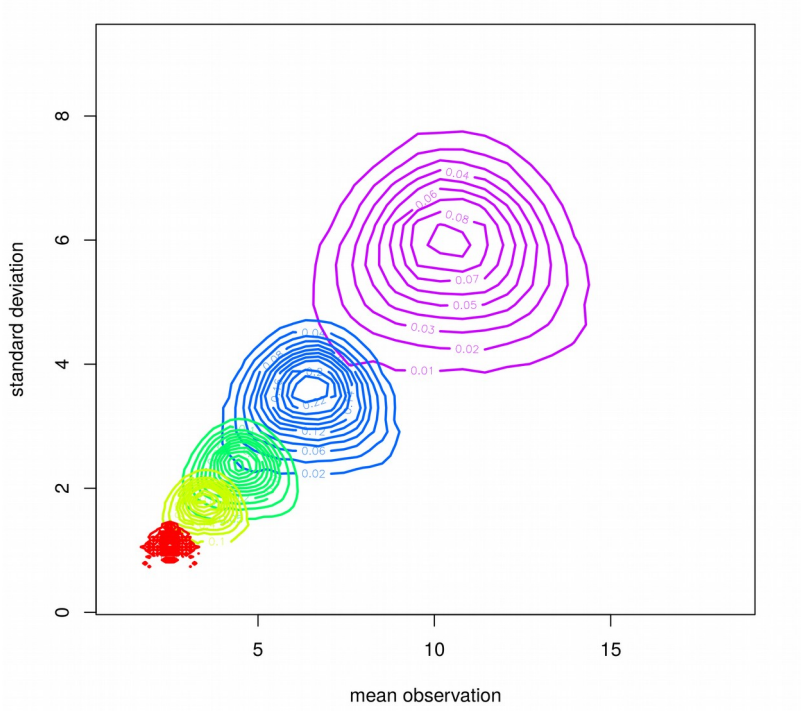
x10,000

ABC in 3 steps

1) summarizing the data with descriptive statistics

```
data = sample( 1:8, 10, replace = T )
print(data)
[1] 5 4 6 6 1 7 3 4 2 2
```

2) random simulations of the statistics under the compared models



D20

D12

D8

D6

D4

x10,000

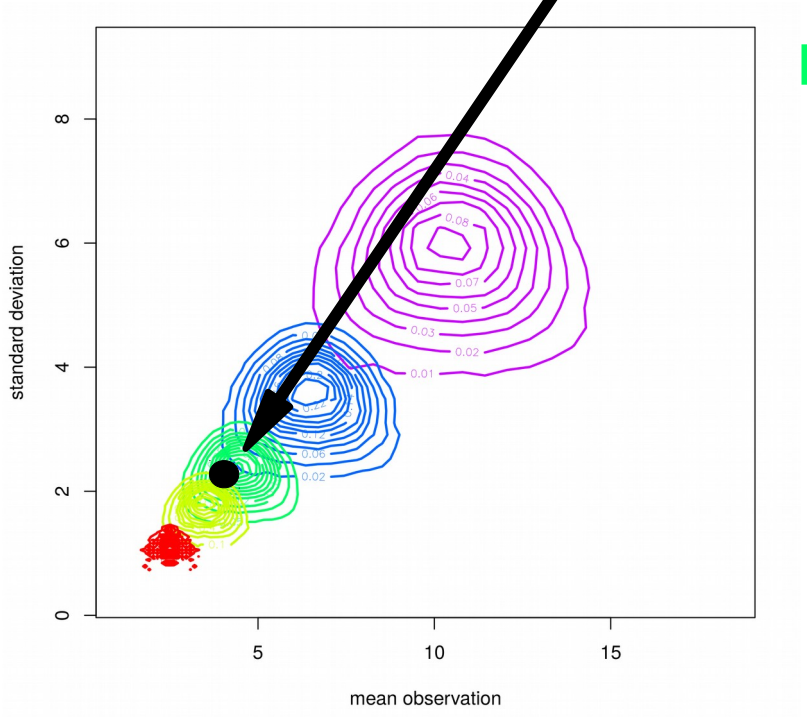
3) statistical comparison between the observation and the simulations

ABC in 3 steps

1) summarizing the data with descriptive statistics

```
data = sample( 1:8, 10, replace = T )
print(data)
[1] 5 4 6 6 1 7 3 4 2 2
```

2) random simulations of the statistics under the compared models



Best model = D8

D20

D12

D8

D6

D4

x10,000

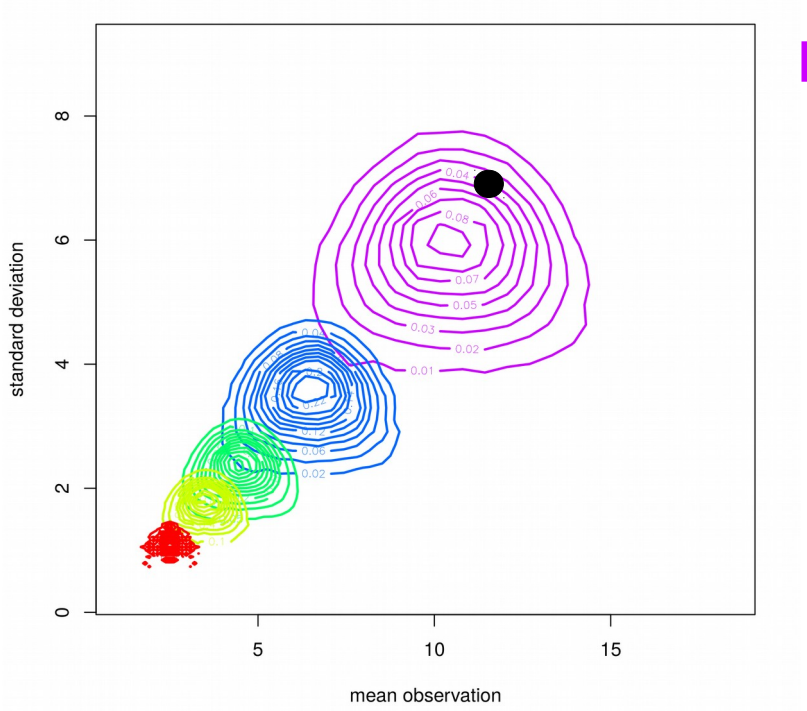
3) statistical comparison between the observation and the simulations

ABC in 3 steps

1) summarizing the data with descriptive statistics

```
data = sample( 1:8, 10, replace = T )
print(data)
[1] 5 4 6 6 1 7 3 4 2 2
```

2) random simulations of the statistics under the compared models



Best model = D20

D20

D12

D8

D6

D4

x10,000

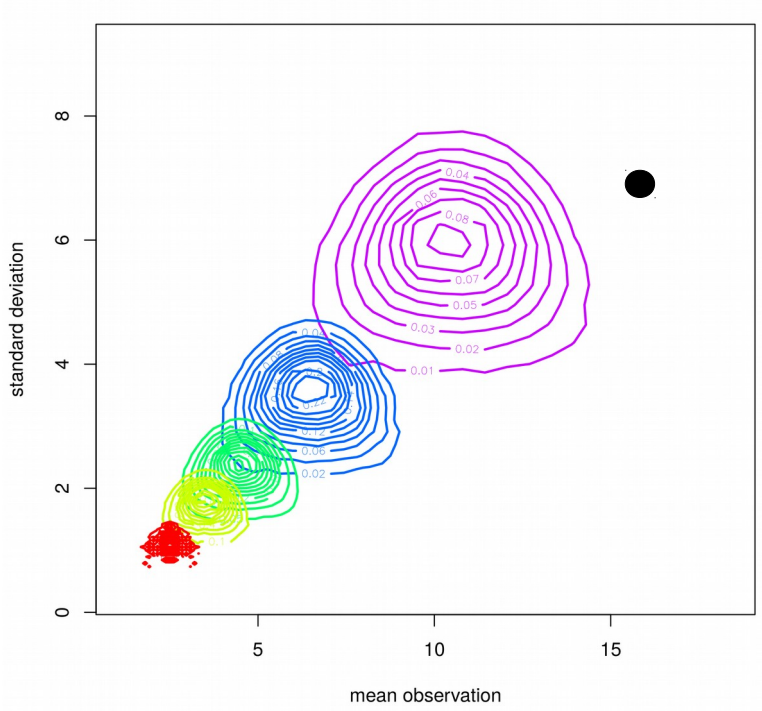
3) statistical comparison between the observation and the simulations

ABC in 3 steps

1) summarizing the data with descriptive statistics

```
data = sample( 1:8, 10, replace = T )
print(data)
[1] 5 4 6 6 1 7 3 4 2 2
```

2) random simulations of the statistics under the compared models



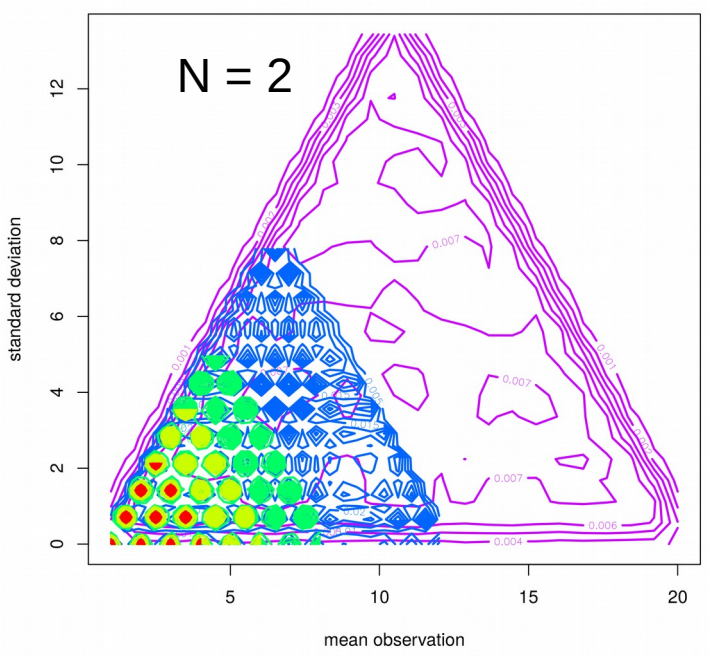
Best model = D20 (still the case)

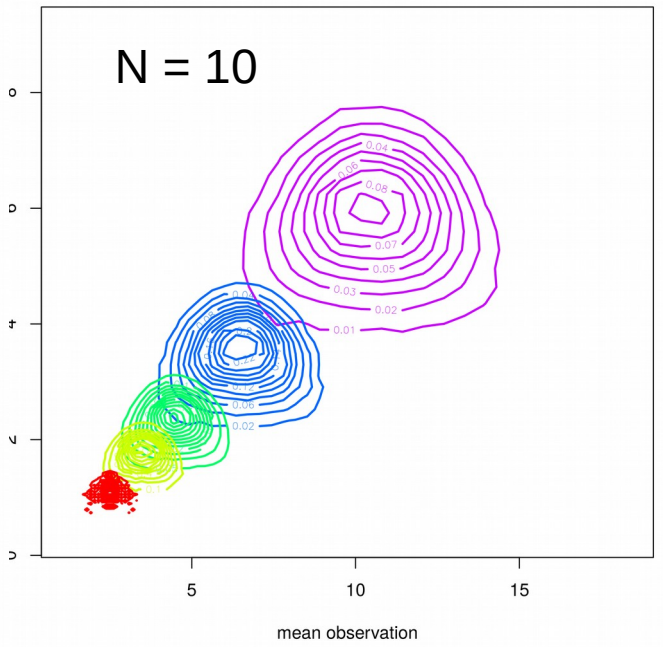
D20
D12
D8
D6
D4

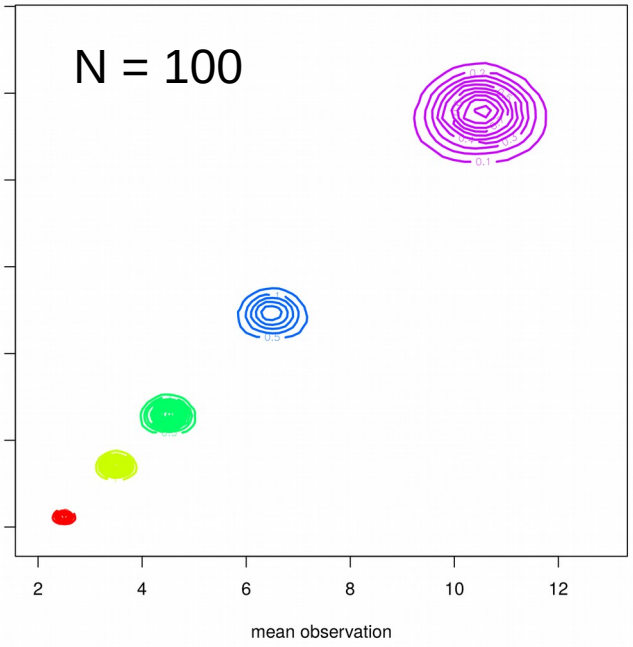
x10,000

3) statistical comparison between the observation and the simulations

Simulating the same sample size than for the observed dataset







Statistical comparison using Random Forests

One Random Forest = thousands of Decision Trees

Statistical comparison using Random Forests

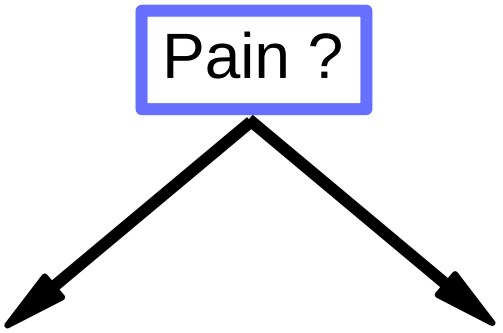
One Random Forest = thousands of Decision Trees

One Decision Tree = a succession of dichotomous criteria leading to a decision

Statistical comparison using Random Forests

One Random Forest = thousands of Decision Trees

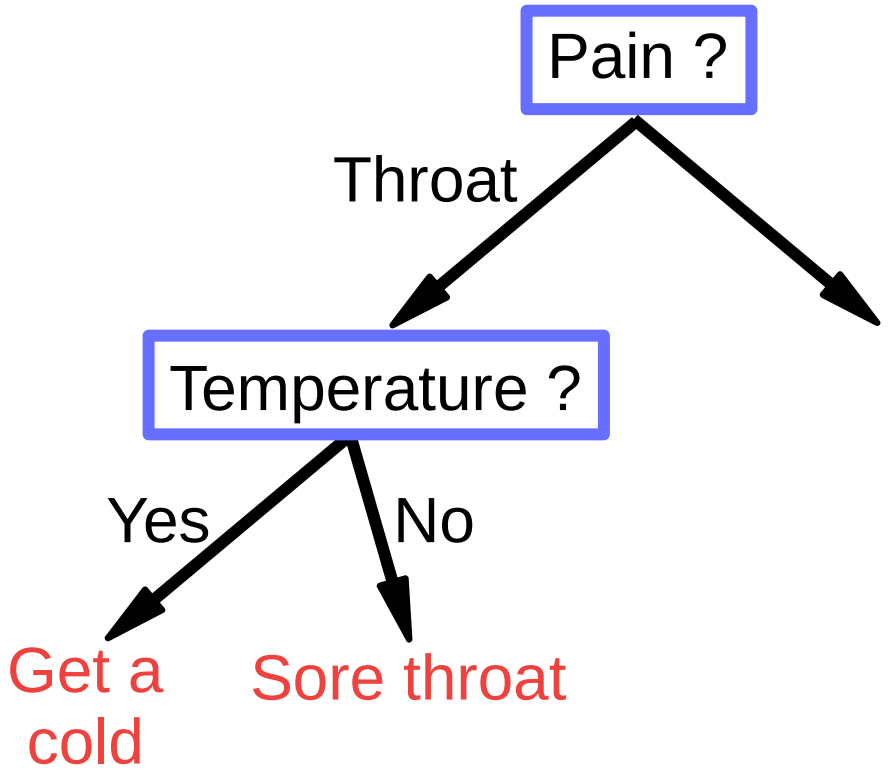
One Decision Tree = a succession of dichotomous criteria leading to a decision



Statistical comparison using Random Forests

One Random Forest = thousands of Decision Trees

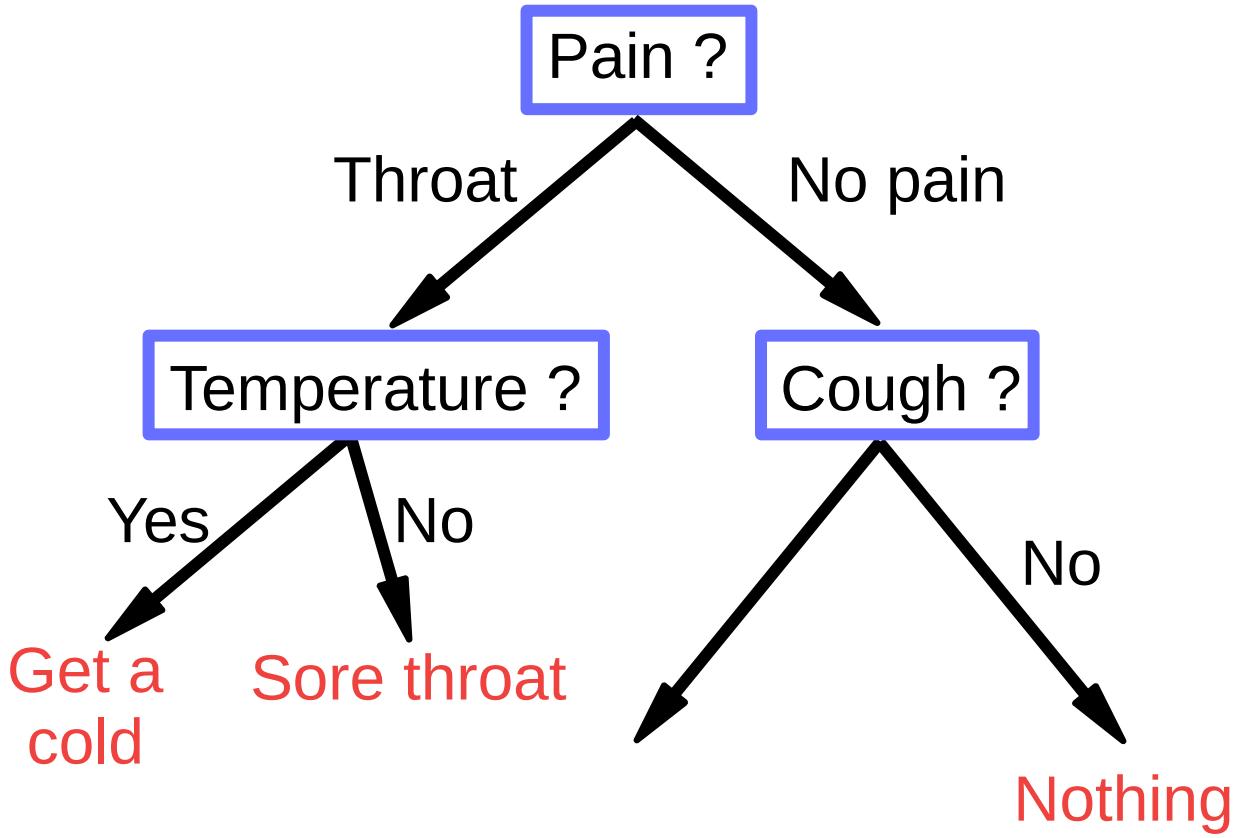
One Decision Tree = a succession of dichotomous criteria leading to a decision



Statistical comparison using Random Forests

One Random Forest = thousands of Decision Trees

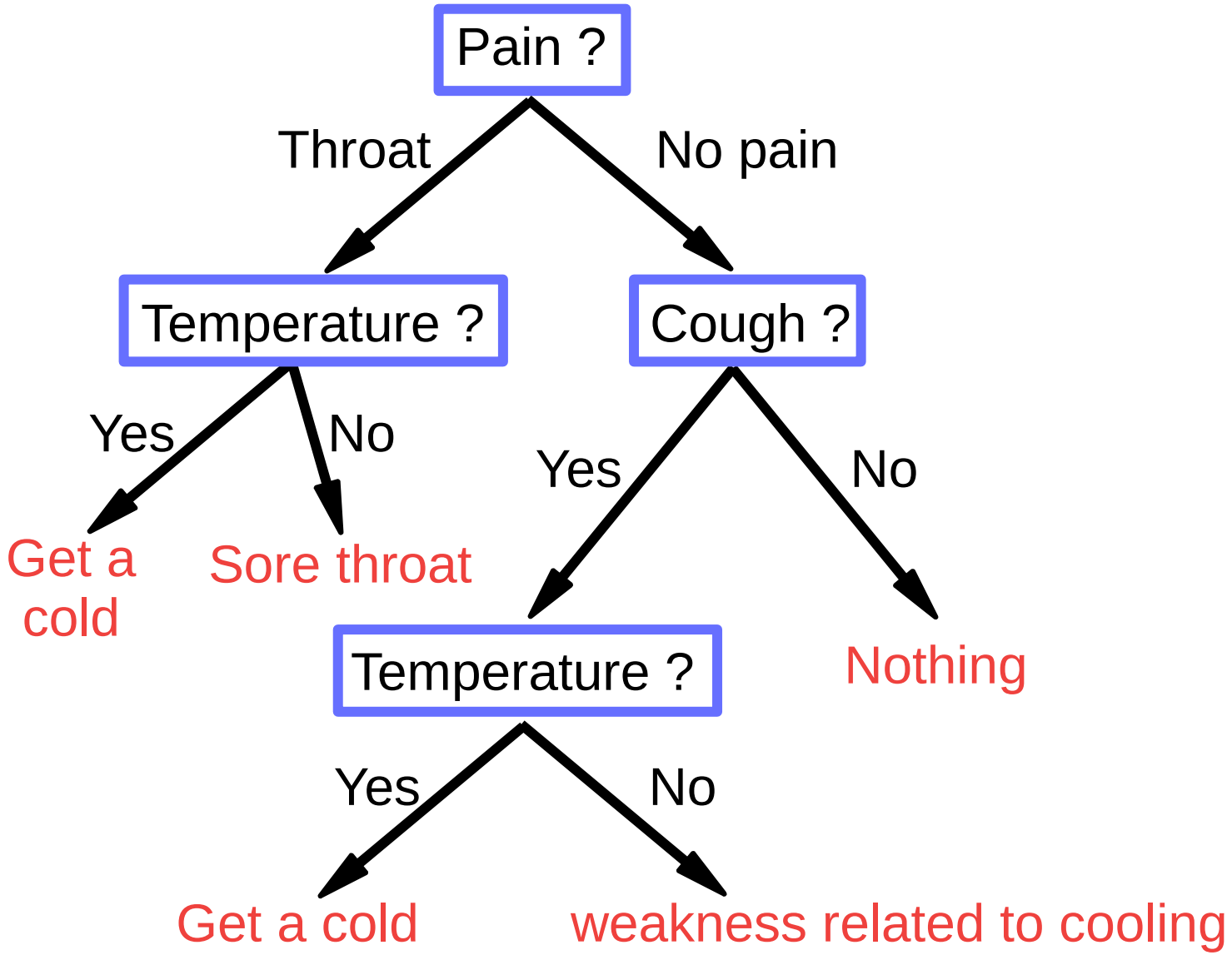
One Decision Tree = a succession of dichotomous criteria leading to a decision



Statistical comparison using Random Forests

One Random Forest = thousands of Decision Trees

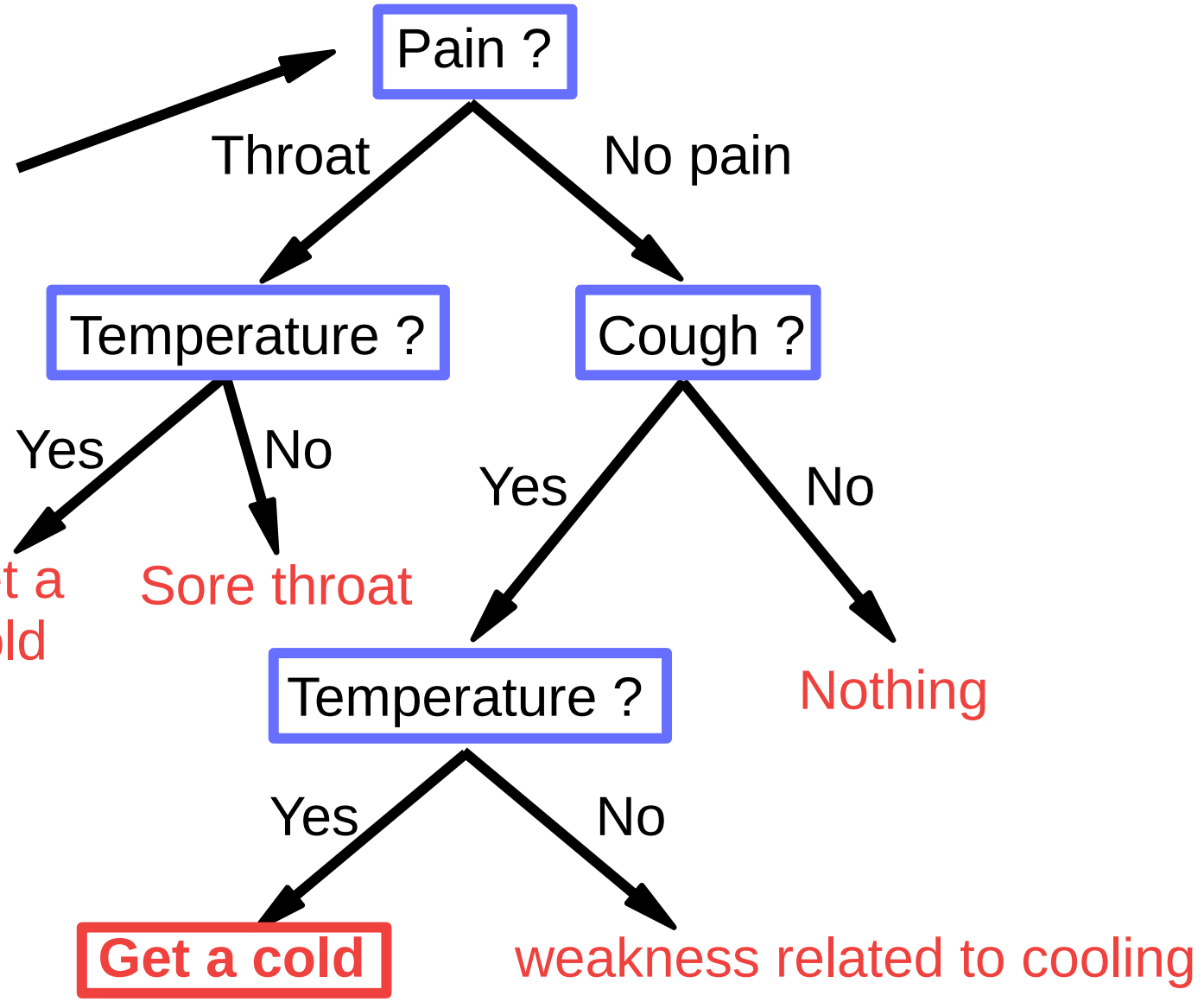
One Decision Tree = a succession of dichotomous criteria leading to a decision



Statistical comparison using Random Forests

One Random Forest = thousands of Decision Trees

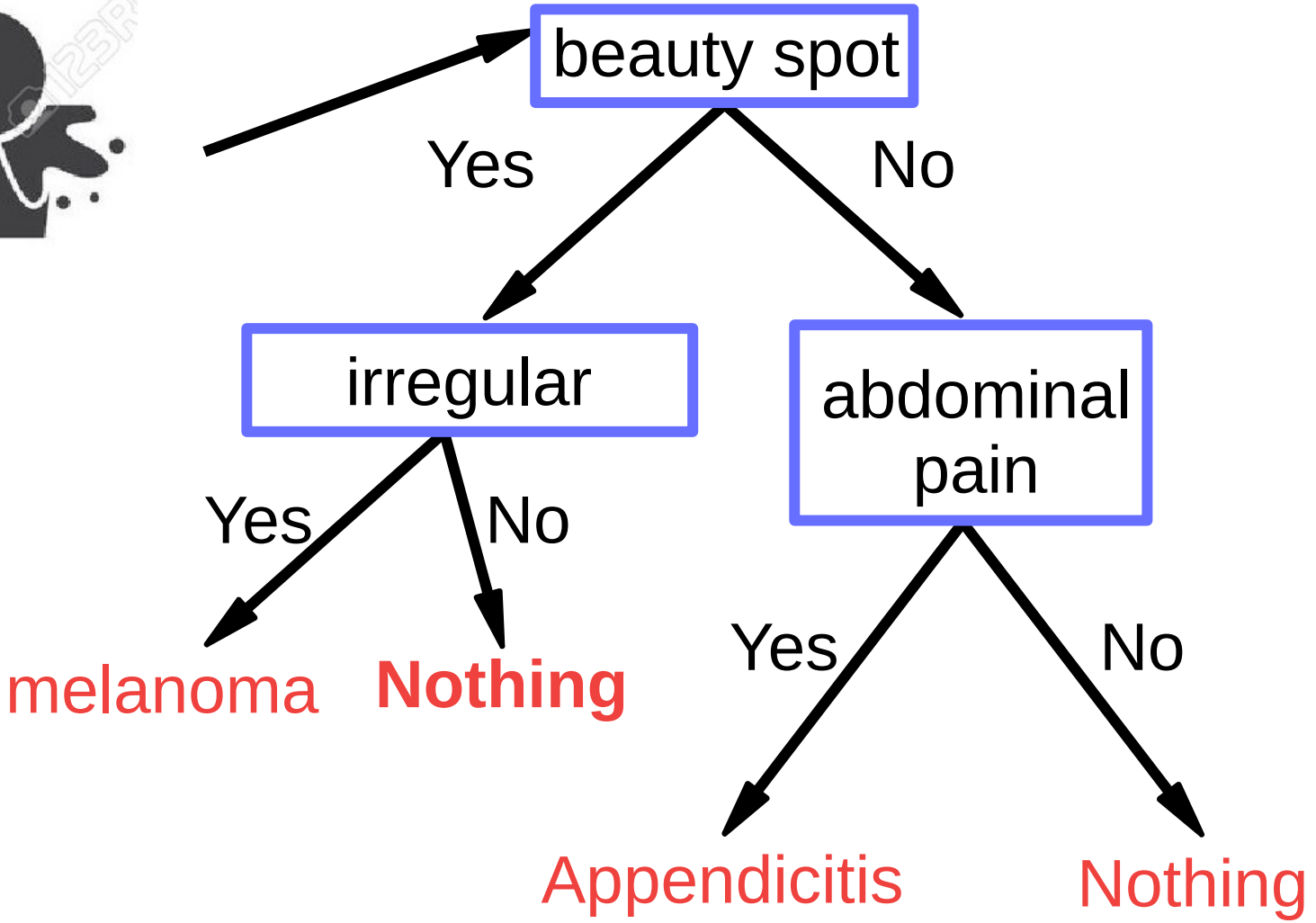
One Decision Tree = a succession of dichotomous criteria leading to a decision



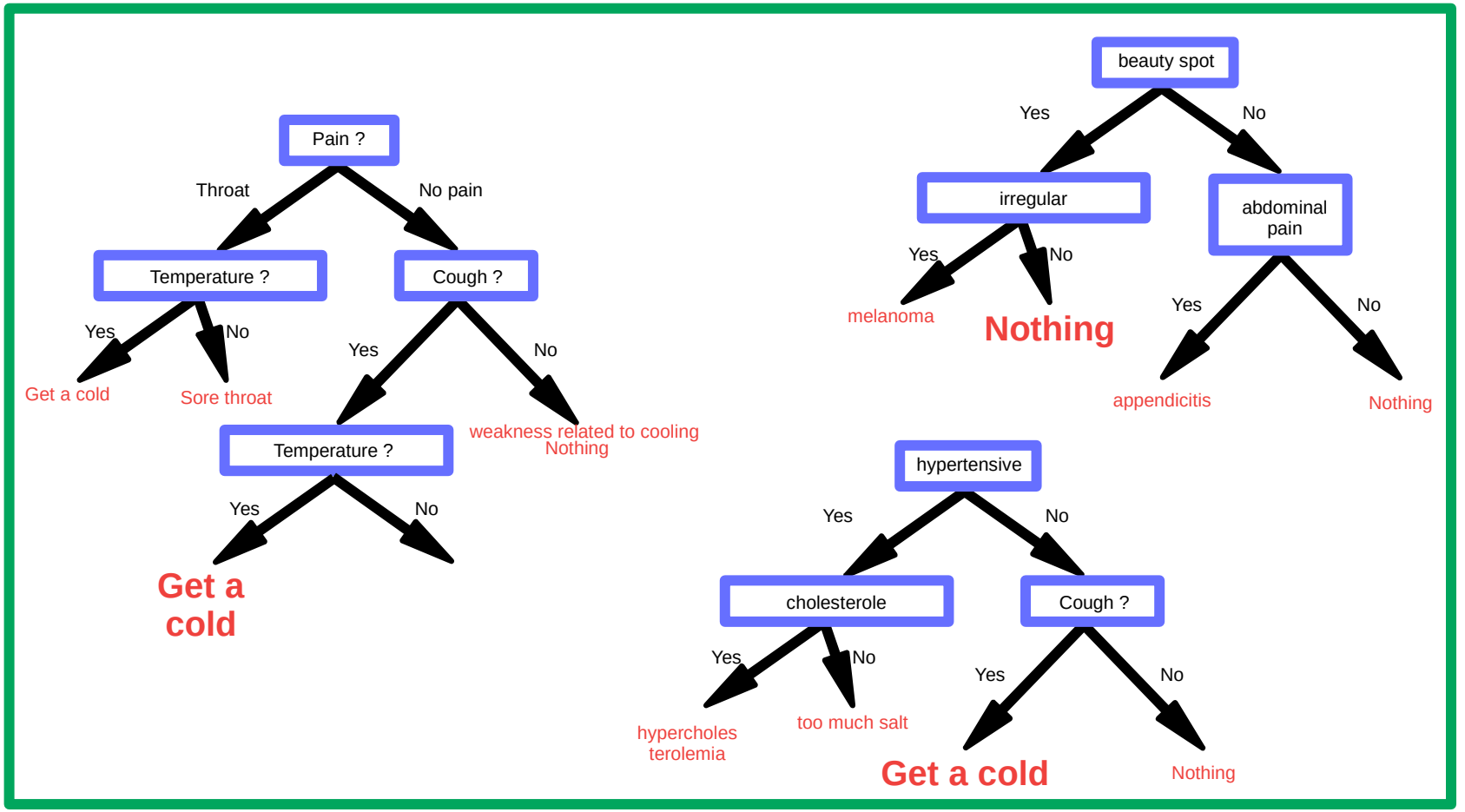
Statistical comparison using Random Forests

One Random Forest = thousands of Decision Trees

One Decision Tree = a succession of dichotomous criteria leading to a decision



Statistical comparison using Random Forests

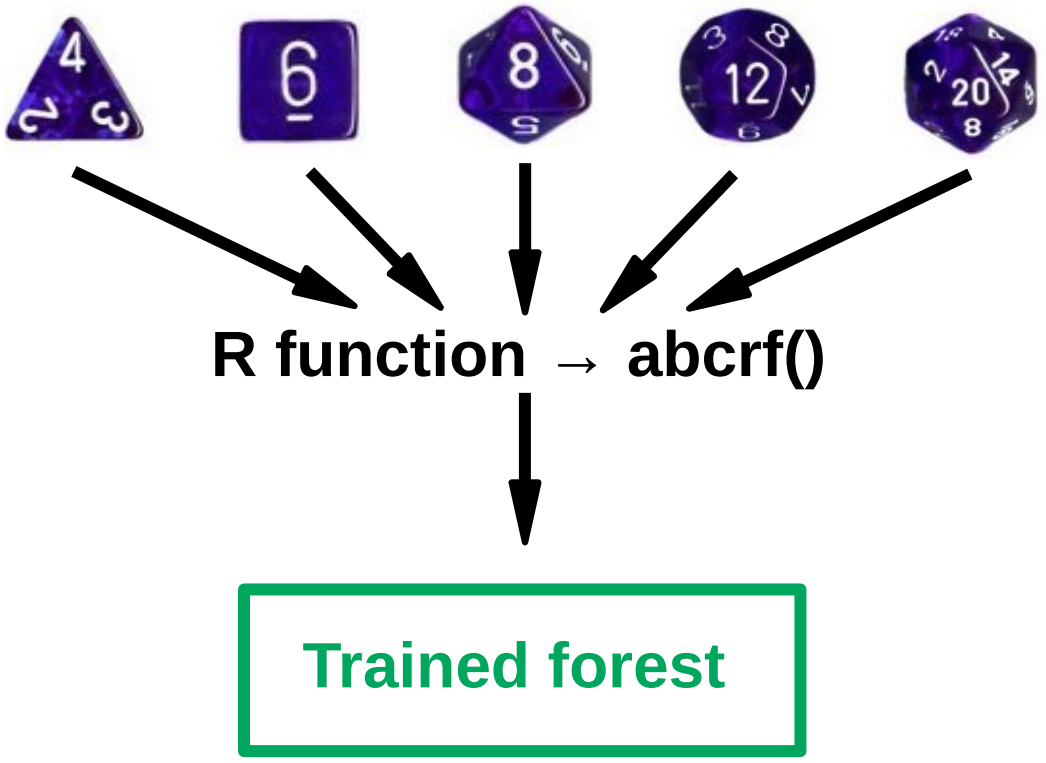


Get a cold (by majority)

Statistical comparison using Random Forests

Two steps : 1) growing **the forest** (= learning or training step) using the R function **abcrf()**

2) making the inference for the observed dataset using the R function **predict()**



Statistical comparison using Random Forests

Two steps : 1) growing **the forest** (= learning or training step) using the R function **abcrf()**

For each model, we will produce 10,000 series of *N* observations (i.e. of *N* die rolls)

Statistical comparison using Random Forests

Two steps : 1) growing **the forest** (= learning or training step) using the R function **abcrf()**

For each model, we will produce 10,000 series of *N* die rolls

```
nSimulations = 10000

simulations_die_8 = as.data.frame(matrix(NA, ncol=N, nrow=nSimulations))

for(i in 1:nSimulations){
        simulations_die_8[i,] = sample(1:8, N, replace=T)
}

colnames(simulations_die_8) = paste( rep('obs', N), 1:N, sep='')
```

Statistical comparison using Random Forests

Two steps : 1) growing **the forest** (= learning or training step) using the R function **abcrf()**

```r
N = 10

nSimulations = 10000

simulations_die_4 = as.data.frame(matrix(NA, ncol=N, nrow=nSimulations))
simulations_die_6 = as.data.frame(matrix(NA, ncol=N, nrow=nSimulations))
simulations_die_8 = as.data.frame(matrix(NA, ncol=N, nrow=nSimulations))
simulations_die_12 = as.data.frame(matrix(NA, ncol=N, nrow=nSimulations))
simulations_die_20 = as.data.frame(matrix(NA, ncol=N, nrow=nSimulations))

for(i in 1:nSimulations){
        simulations_die_4[i,] = sample(1:4, N, replace=T)
        simulations_die_6[i,] = sample(1:6, N, replace=T)
        simulations_die_8[i,] = sample(1:8, N, replace=T)
        simulations_die_12[i,] = sample(1:12, N, replace=T)
        simulations_die_20[i,] = sample(1:20, N, replace=T)
}

colnames(simulations_die_4) = paste( rep('obs', N), 1:N, sep='')
colnames(simulations_die_6) = paste( rep('obs', N), 1:N, sep='')
colnames(simulations_die_8) = paste( rep('obs', N), 1:N, sep='')
colnames(simulations_die_12) = paste( rep('obs', N), 1:N, sep='')
colnames(simulations_die_20) = paste( rep('obs', N), 1:N, sep='')
```

Statistical comparison using Random Forests

Two steps : 1) growing **the forest** (= learning or training step) using the R function **abcrf()**

Gathering all simulations in the same data.frame

```
all_simulations = rbind(simulations_die_4, simulations_die_6, simulations_die_8, simulations_die_12, simulations_die_20)
```

Statistical comparison using Random Forests

Two steps : 1) growing **the forest** (= learning or training step) using the R function **abcrf()**

Gathering all simulations in the same data.frame

```
all_simulations = rbind(simulations_die_4, simulations_die_6, simulations_die_8, simulations_die_12, simulations_die_20)
```

Getting a correspondance between each row of « all_simulations » and the associated models

```
modIndexes = rep(c('D4', 'D6', 'D8', 'D12', 'D20'), each = nSimulations)
```

Statistical comparison using Random Forests

Two steps : 1) growing **the forest** (= learning or training step) using the R function **abcrf()**

Gathering all simulations in the same data.frame

```
all_simulations = rbind(simulations_die_4, simulations_die_6, simulations_die_8, simulations_die_12, simulations_die_20)
```

Getting a correspondance between each row of « all_simulations » and the associated models

```
modIndexes = rep(c('D4', 'D6', 'D8', 'D12', 'D20'), each = nSimulations)
```

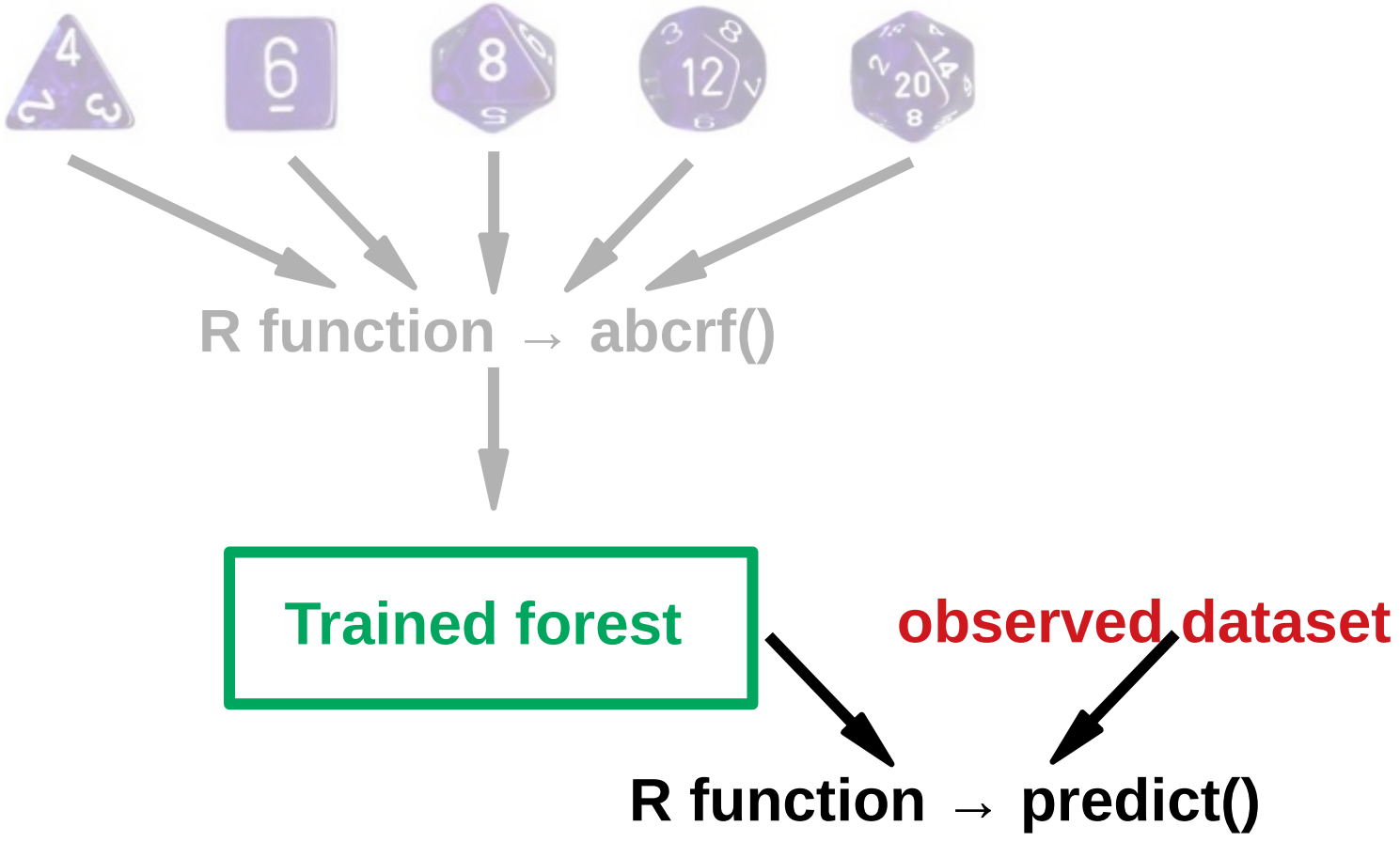
Training the forest

```
mod = abcrf(modIndexes~., data = data.frame(modIndexes, all_simulations), ntree = 1000, paral = T, ncores = 3)
```

Statistical comparison using Random Forests

Two steps : 1) growing **the forest** (= learning or training step) using the R function **abcrf()**

2) making the inference for the **observed dataset** using the R function **predict()**



Statistical comparison using Random Forests

Two steps : 1) growing **the forest** (= learning or training step) using the R function **abcrf()**

```
mod = abcrf(modIndexes~., data = data.frame(modIndexes, all_simulations), ntree = 1000, paral = T, ncores = 3)
```

Trained forest = mod

2) making the inference for the **observed dataset** using the R function **predict()**

Producing some « pseudo observations »

```
observations = matrix(sample(1:8, N, replace=T), nrow=1, ncol=N)
observations = as.data.frame(observations)
names(observations) = paste( rep('obs', N), 1:N, sep='')
```

Statistical comparison using Random Forests

Two steps : 1) growing **the forest** (= learning or training step) using the R function **abcrf()**

```
mod = abcrf(modIndexes~., data = data.frame(modIndexes, all_simulations), ntree = 1000, paral = T, ncores = 3)
```

Trained forest = mod

2) making the inference for the **observed dataset** using the R function **predict()**

Producing some « pseudo observations »

```
observations = matrix(sample(1:8, N, replace=T), nrow=1, ncol=N)
observations = as.data.frame(observations)
names(observations) = paste( rep('obs', N), 1:N, sep='')
```

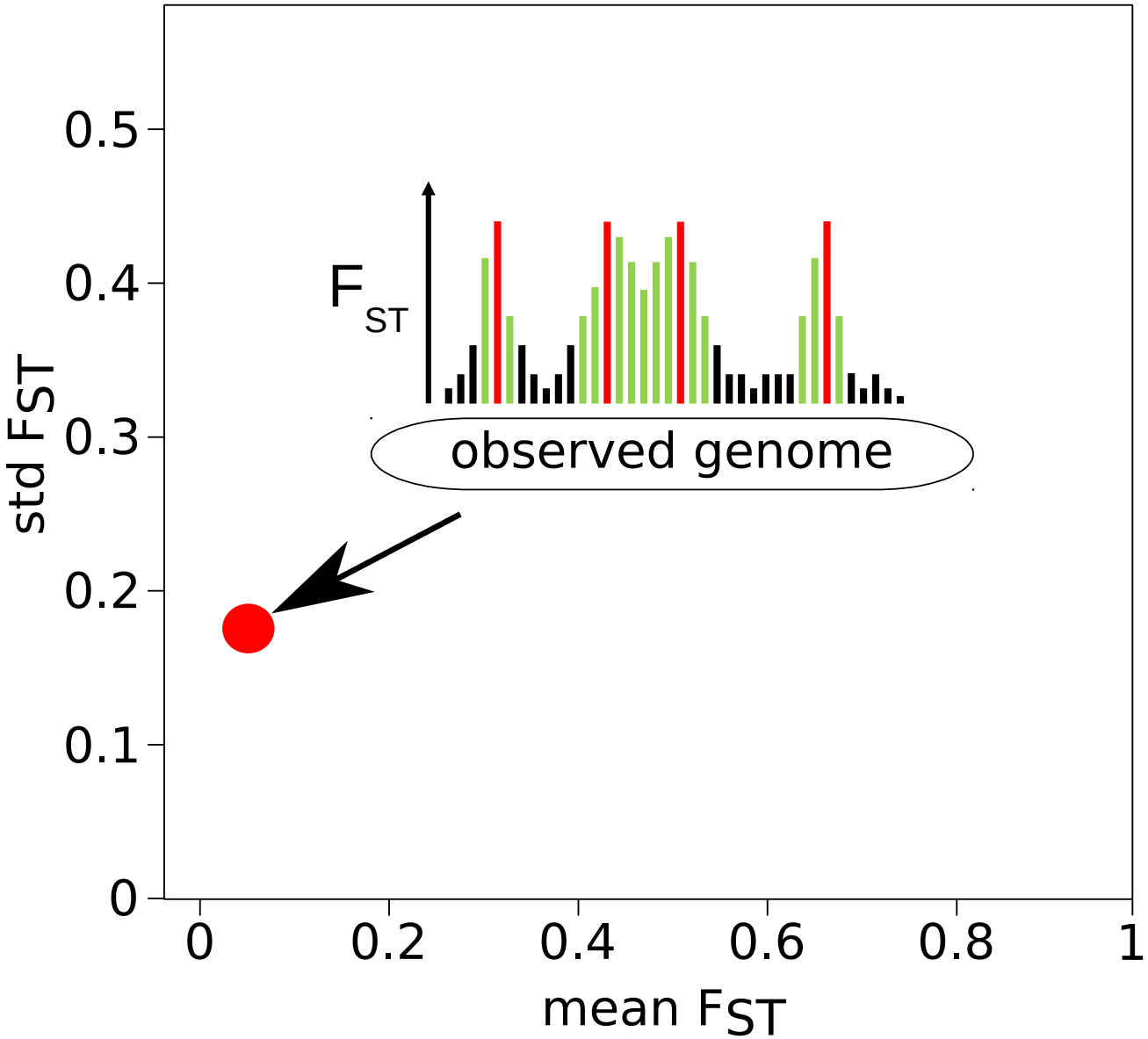
Using the trained machine learning 🍾

```
predicted_dice = predict(mod, observations, training=data.frame(modIndexes, all_simulations),
ntree = 1000, paral = T, ncores=2)
```

Before Lunch :

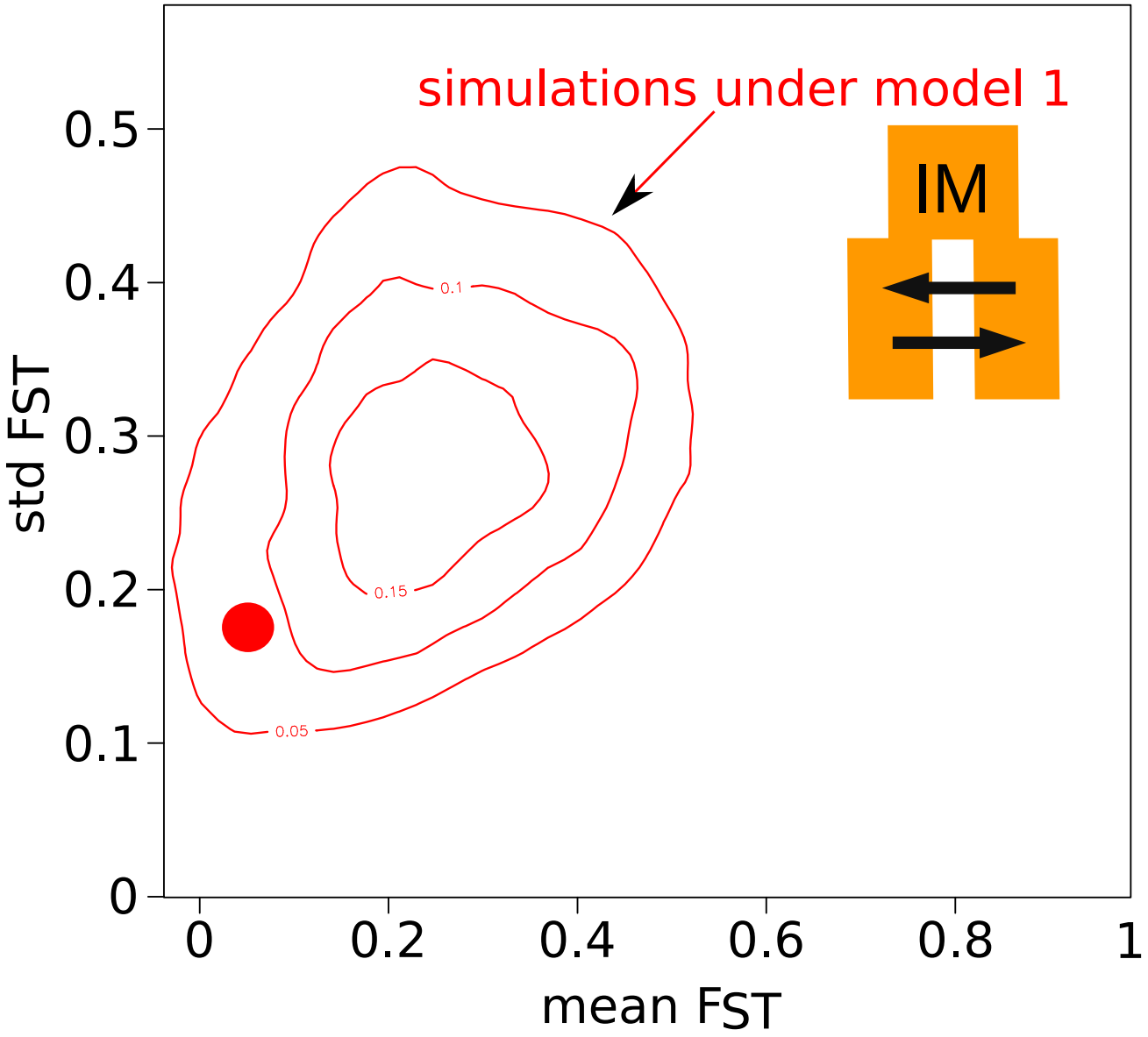
Comparing the Bayesian posterior probability with the approximated posterior probability for different conditions !

approximate Bayesian computation



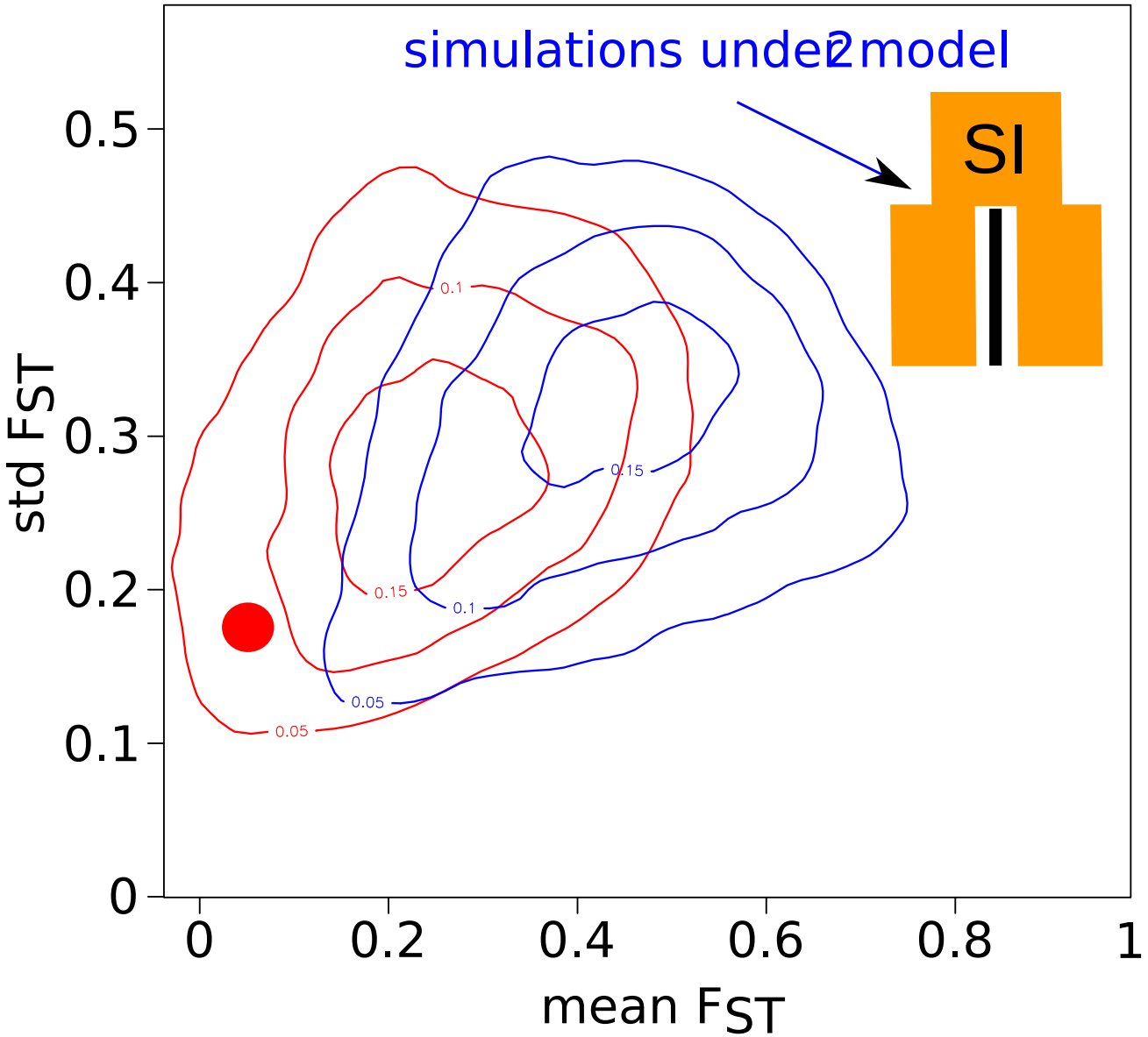
1) summarizing the data with descriptive statistics
2) random simulations under the compared models
3) statistical comparison between the observation and the simulations

approximate Bayesian computation



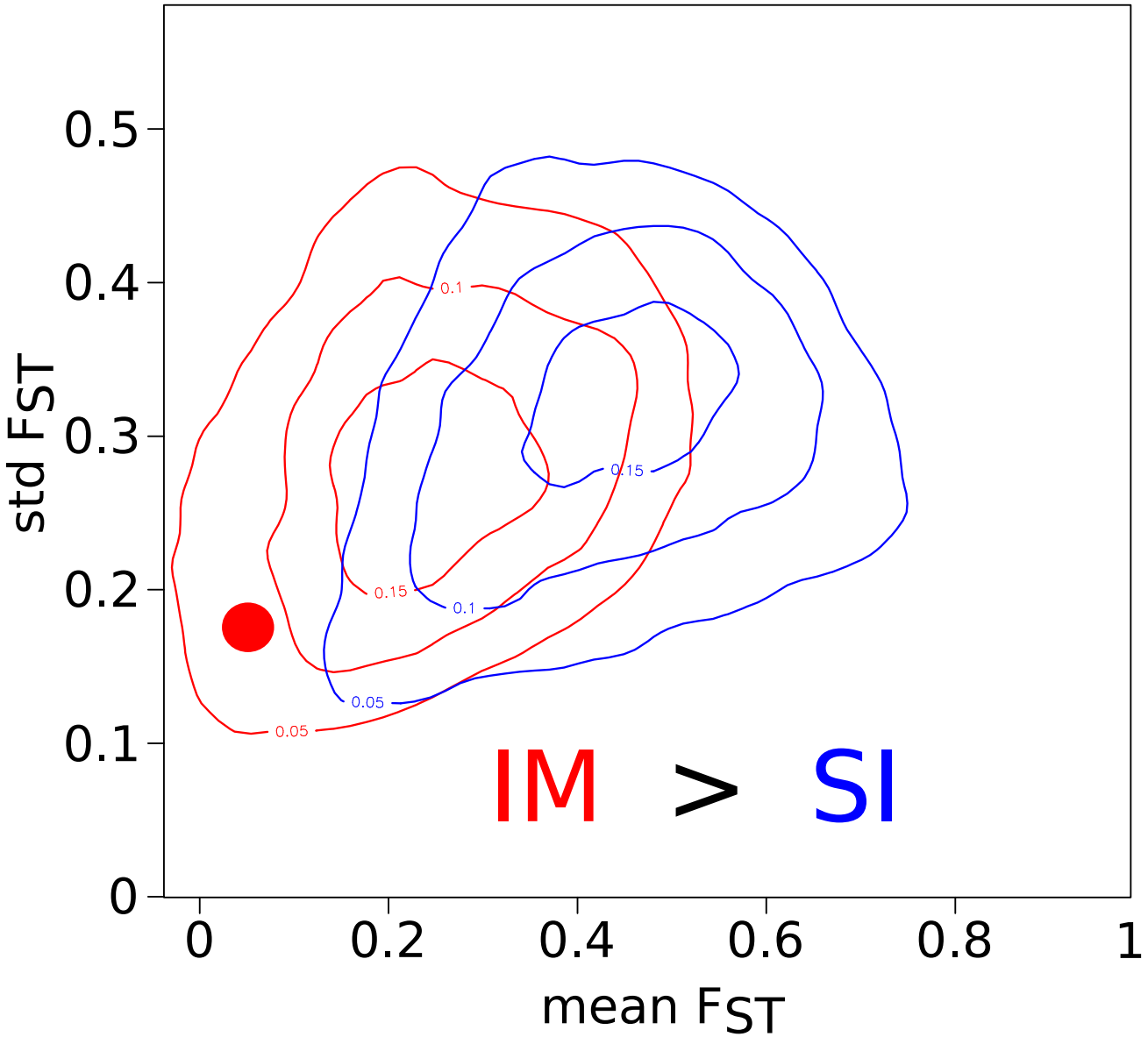
1) summarizing the data with descriptive statistics
2) random simulations under the compared models
3) statistical comparison between the observation and the simulations

approximate Bayesian computation



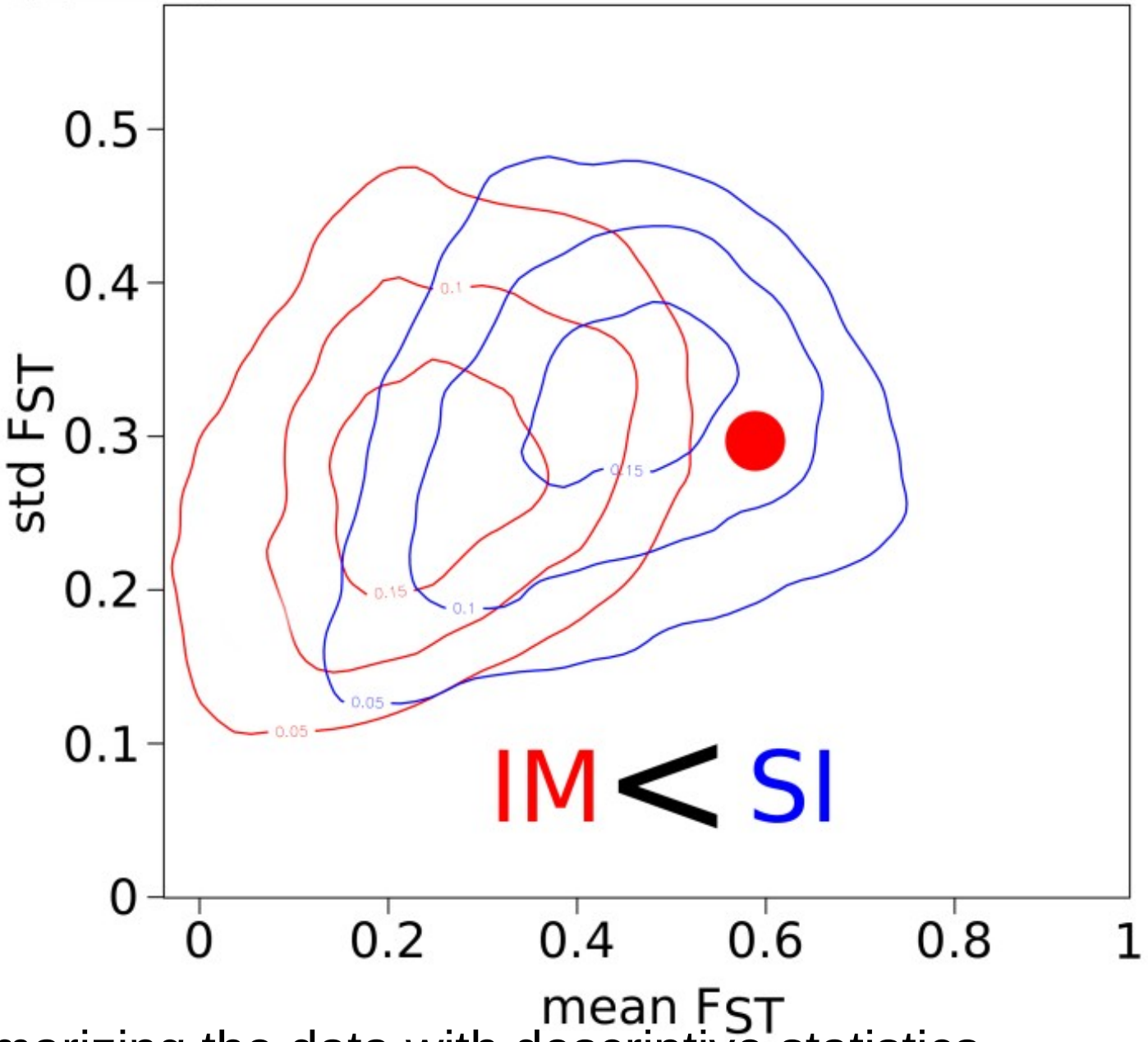
1) summarizing the data with descriptive statistics
2) random simulations under the compared models
3) statistical comparison between the observation and the simulations

approximate Bayesian computation



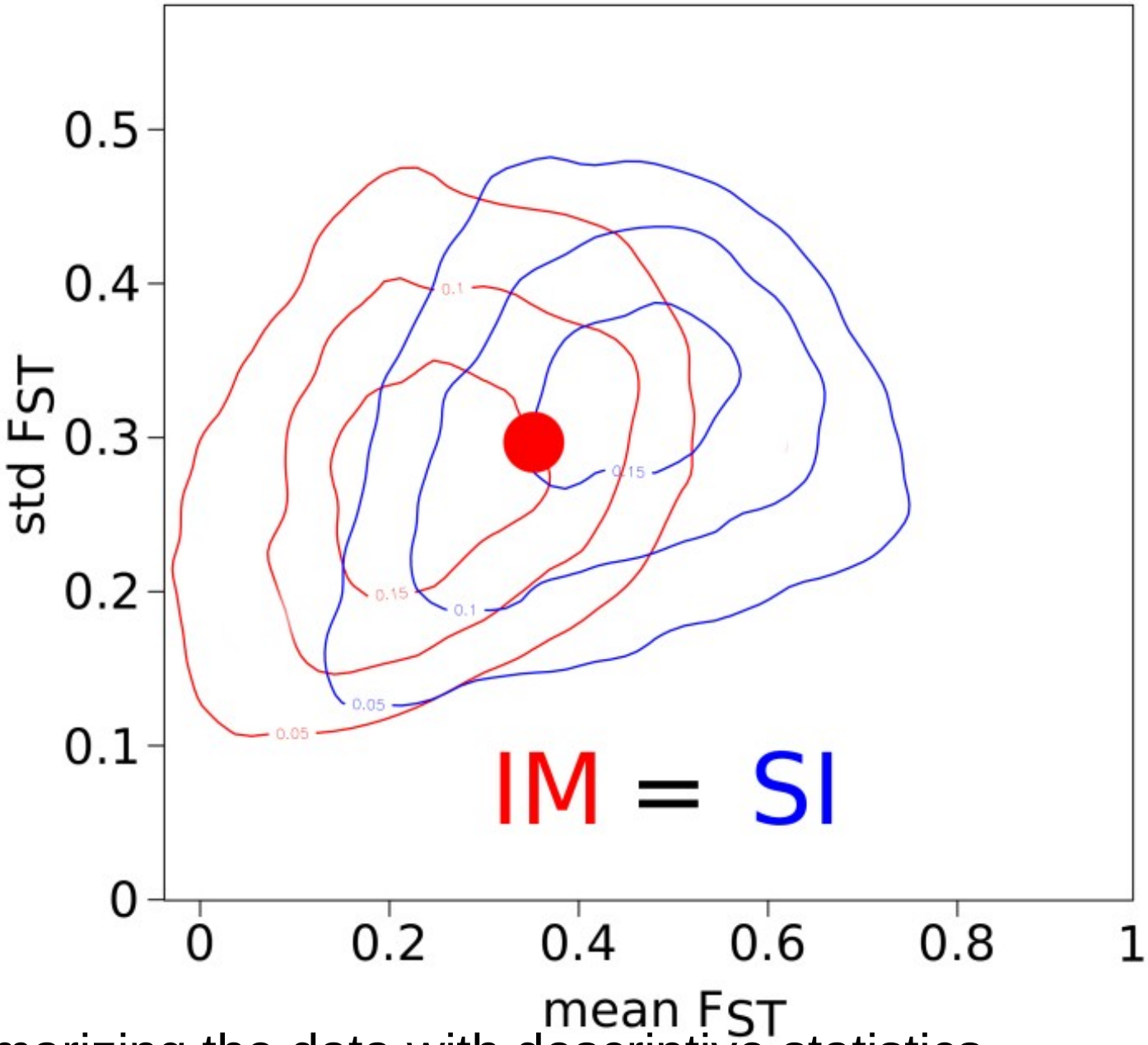
1) summarizing the data with descriptive statistics
2) random simulations under the compared models
3) statistical comparison between the observation and the simulations

approximate Bayesian computation



1) summarizing the data with descriptive statistics
2) random simulations under the compared models
3) statistical comparison between the observation and the simulations

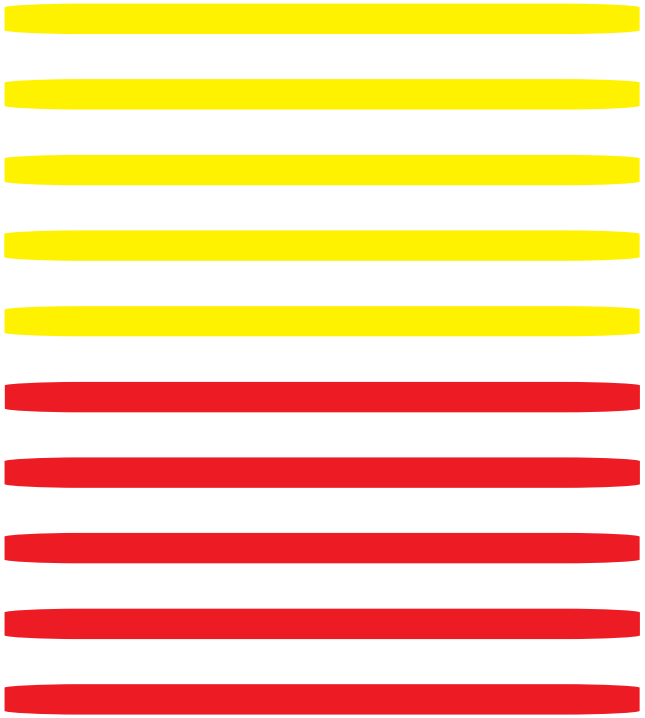
approximate Bayesian computation

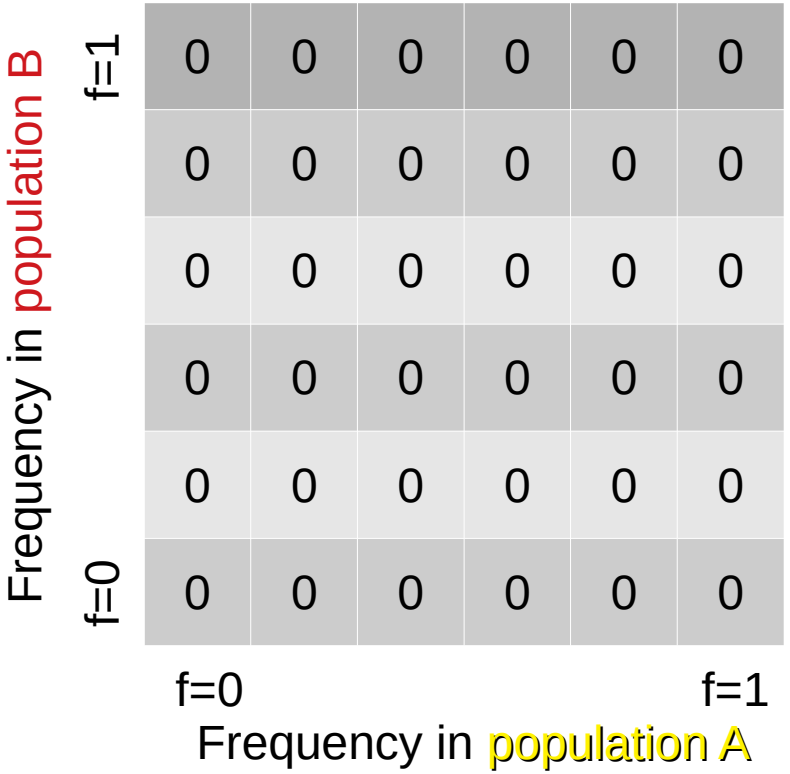


1) summarizing the data with descriptive statistics
2) random simulations under the compared models
3) statistical comparison between the observation and the simulations

Summarizing the information about history from sequenced genomes

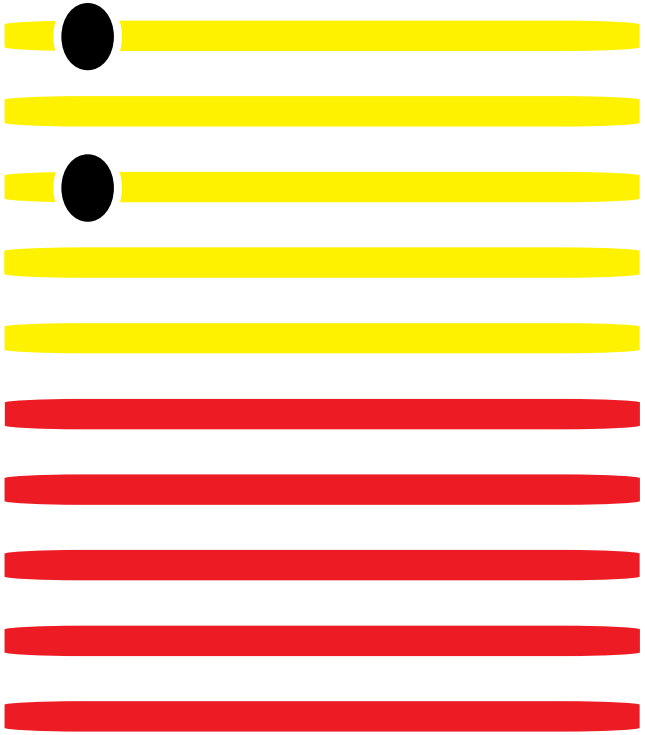
Describing the joint site frequency spectrum (**jSFS**) :

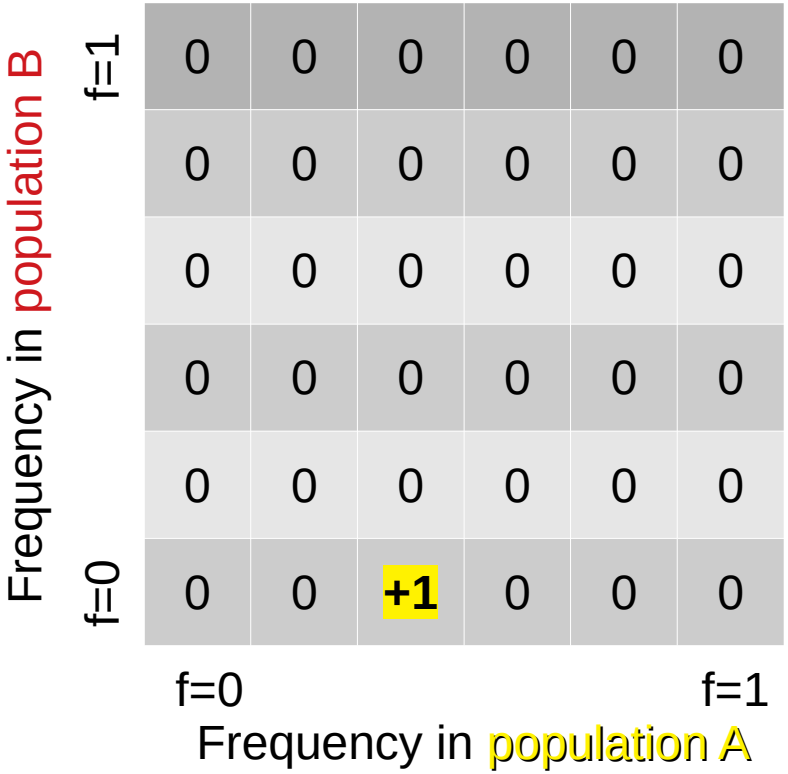


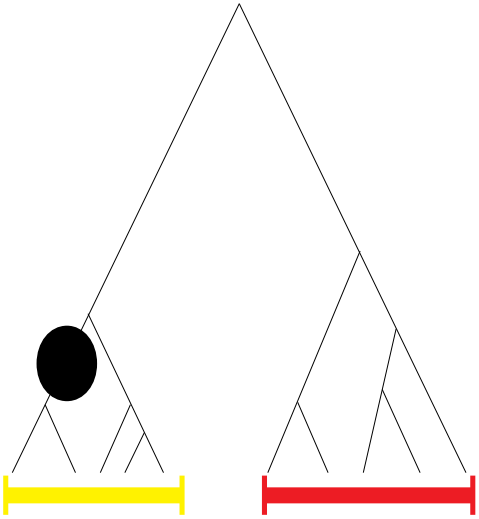


Summarizing the information about history from sequenced genomes

Describing the joint site frequency spectrum (**jSFS**) :

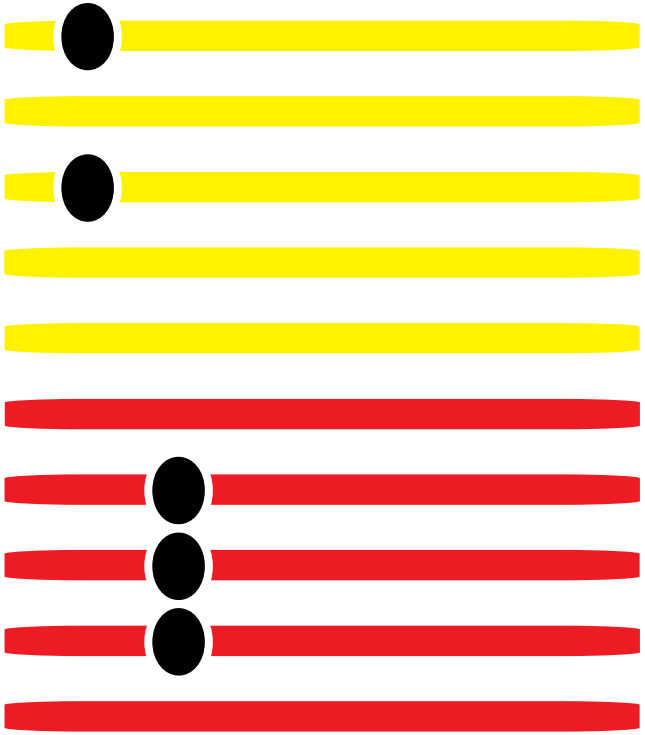






Summarizing the information about history from sequenced genomes

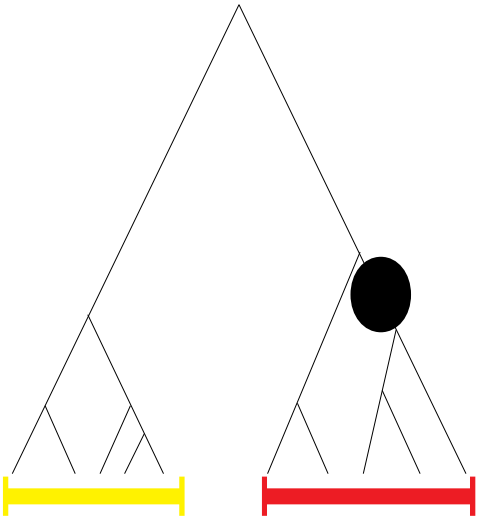
Describing the joint site frequency spectrum (**jSFS**) :



Frequency in population B f=1	0	0	0	0	0	0
	0	0	0	0	0	0
	+1	0	0	0	0	0
	0	0	0	0	0	0
	0	0	0	0	0	0
f=0	0	0	**1**	0	0	0

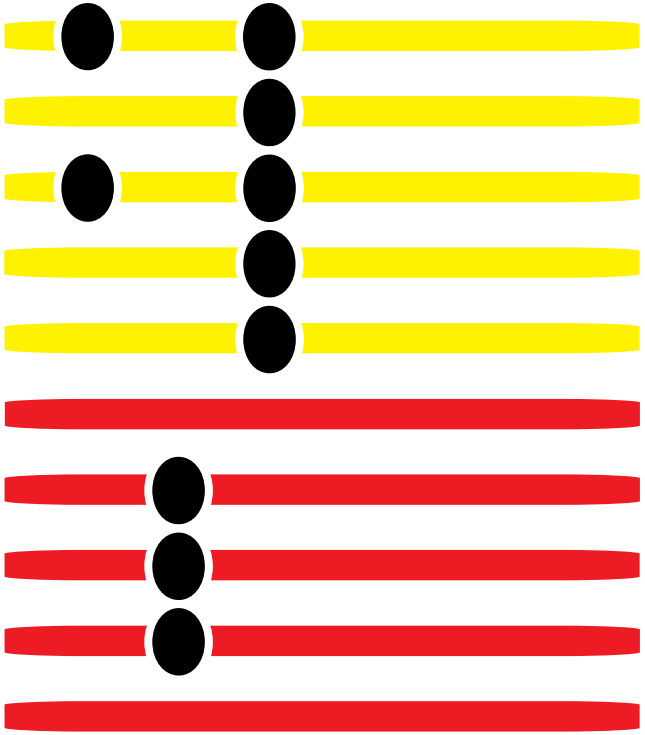
f=0 f=1

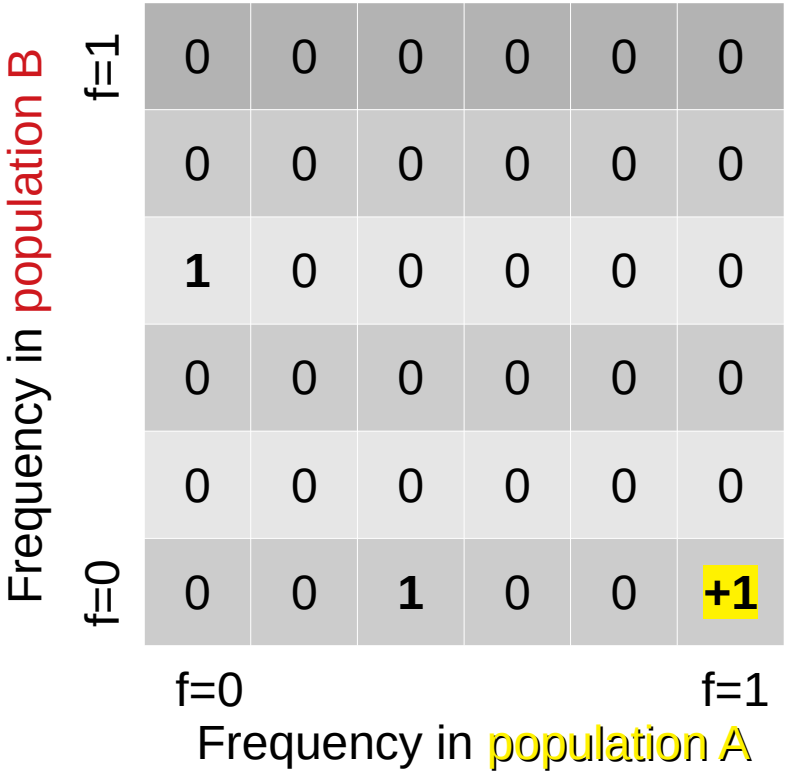
Frequency in population A

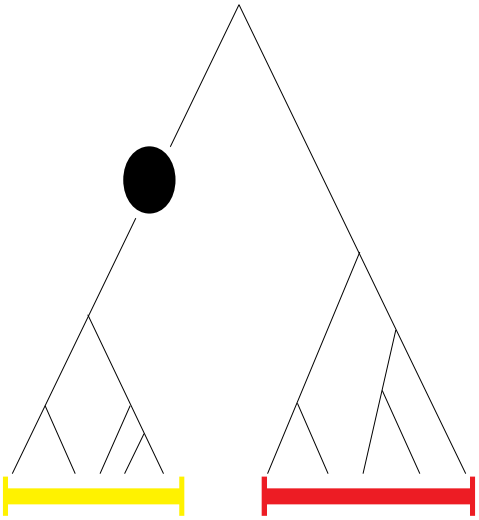


Summarizing the information about history from sequenced genomes

Describing the joint site frequency spectrum (**jSFS**) :

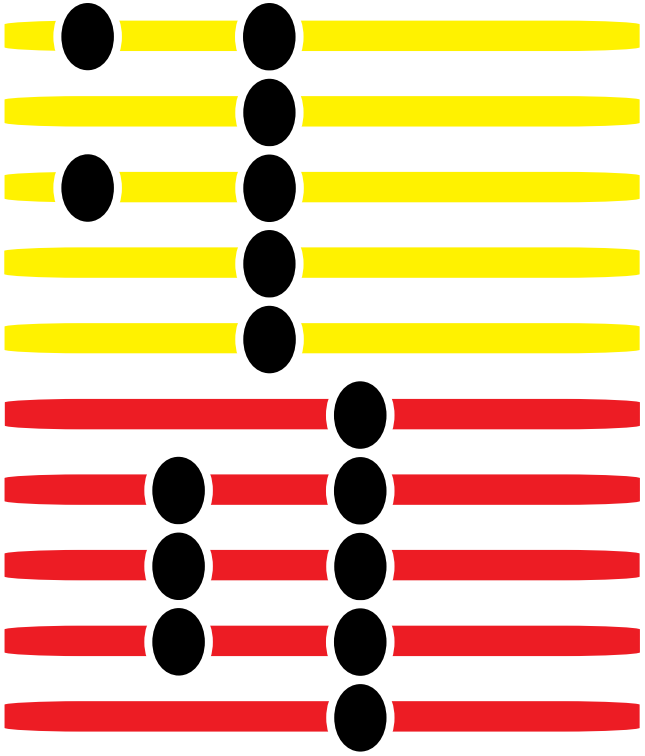






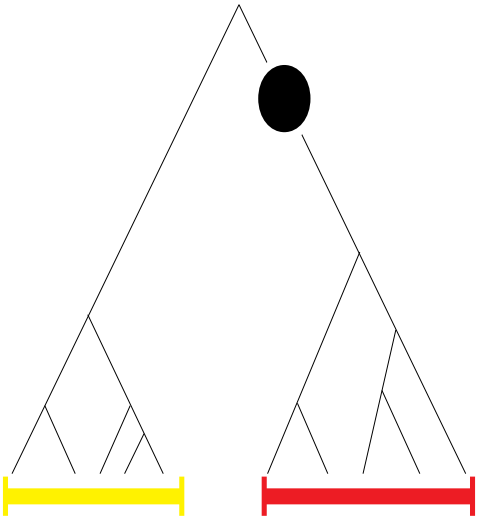
Summarizing the information about history from sequenced genomes

Describing the joint site frequency spectrum (**jSFS**) :



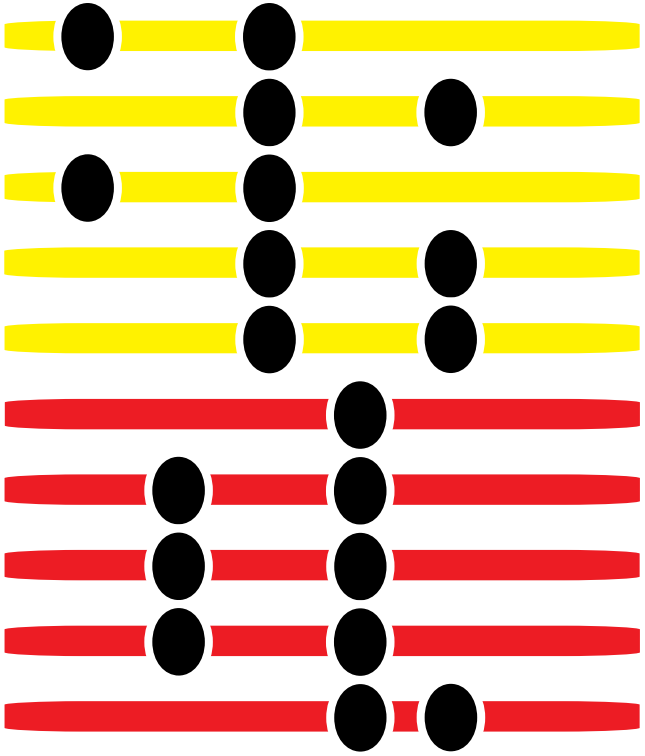
Frequency in population B						
f=1	**+1**	0	0	0	0	0
	0	0	0	0	0	0
	1	0	0	0	0	0
	0	0	0	0	0	0
	0	0	0	0	0	0
f=0	0	0	**1**	0	0	**1**
	f=0					f=1

Frequency in population A



Summarizing the information about history from sequenced genomes

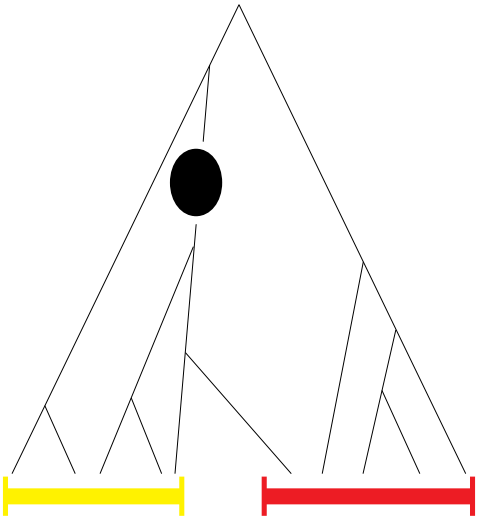
Describing the joint site frequency spectrum (**jSFS**) :



f=1	**1**	0	0	0	0	0
	0	0	0	0	0	0
	0	0	0	0	0	0
	1	0	0	0	0	0
	0	0	0	**+1**	0	0
f=0	0	0	0	**1**	0	**1**
	f=0					f=1

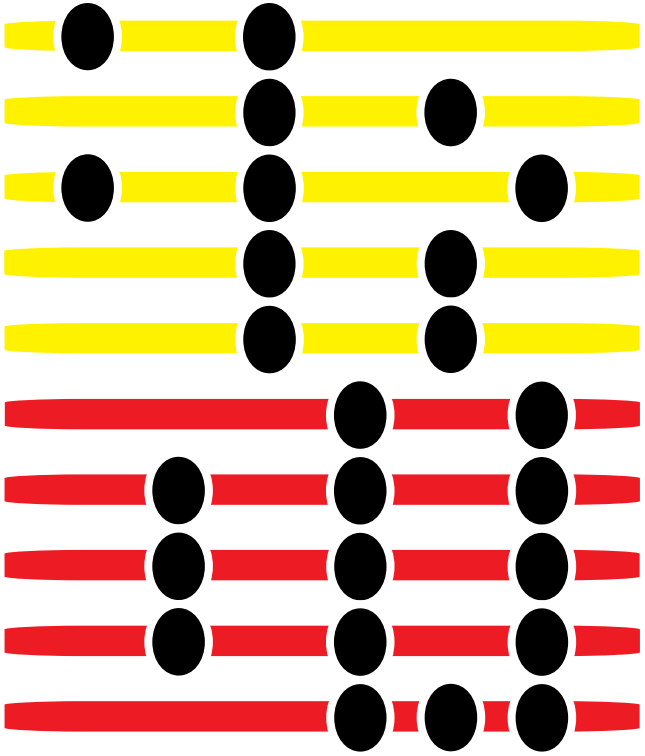
Frequency in population B

Frequency in population A



Summarizing the information about history from sequenced genomes

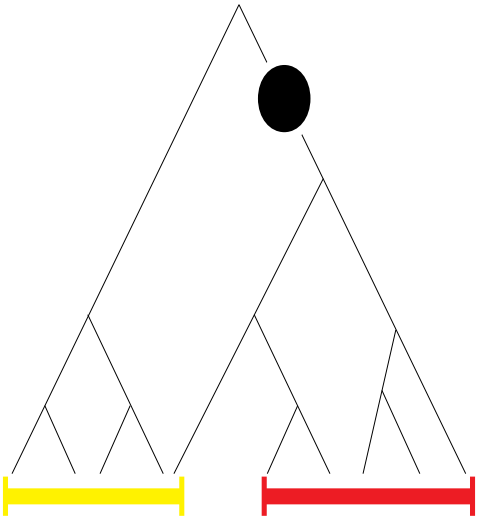
Describing the joint site frequency spectrum (**jSFS**) :

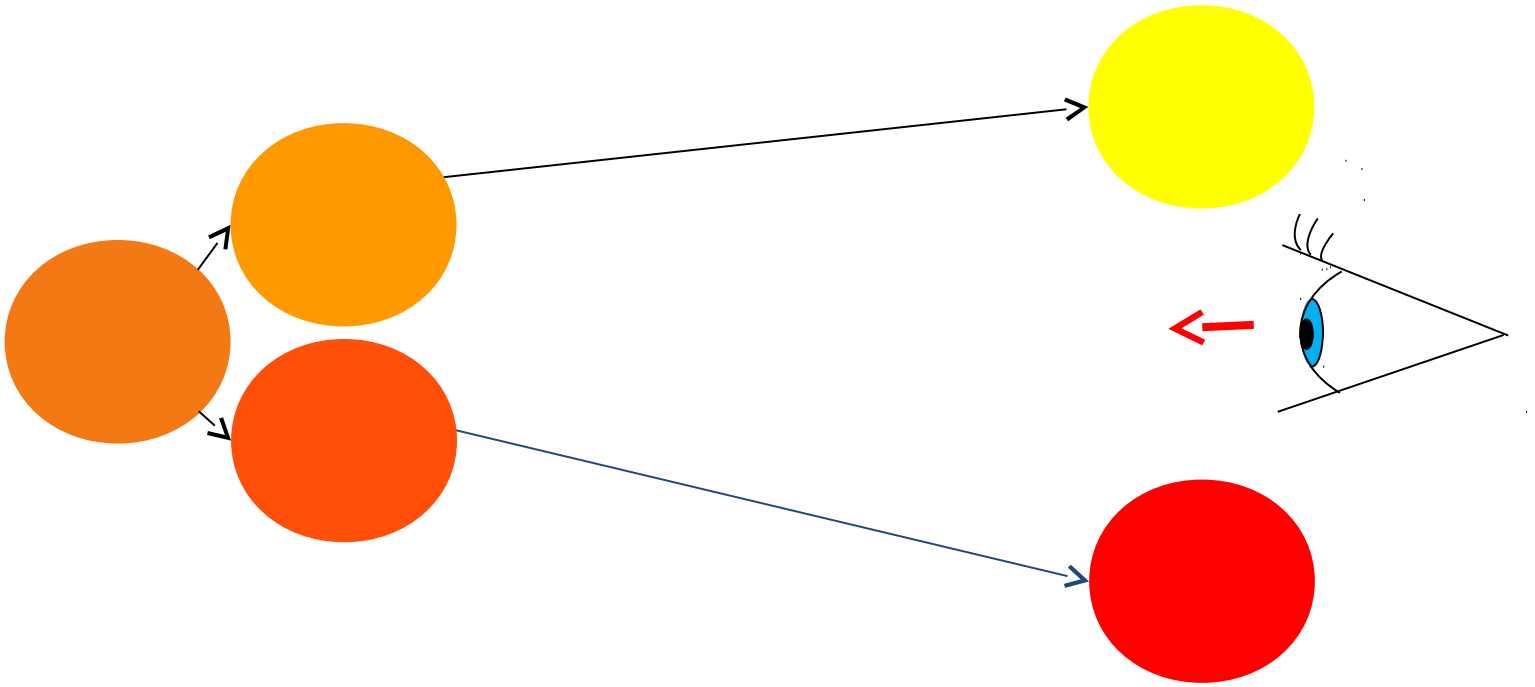


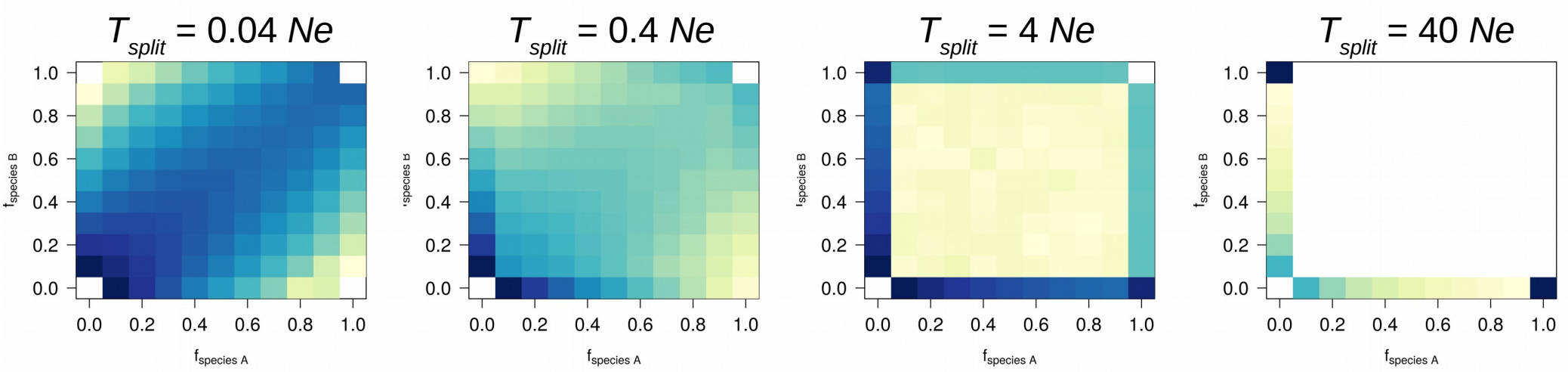
f=1	**1**	**+1**	0	0	0	0
	0	0	0	0	0	0
	0	0	0	0	0	0
	1	0	0	0	0	0
	0	0	0	**1**	0	0
f=0	0	0	0	**1**	0	**1**
	f=0					f=1

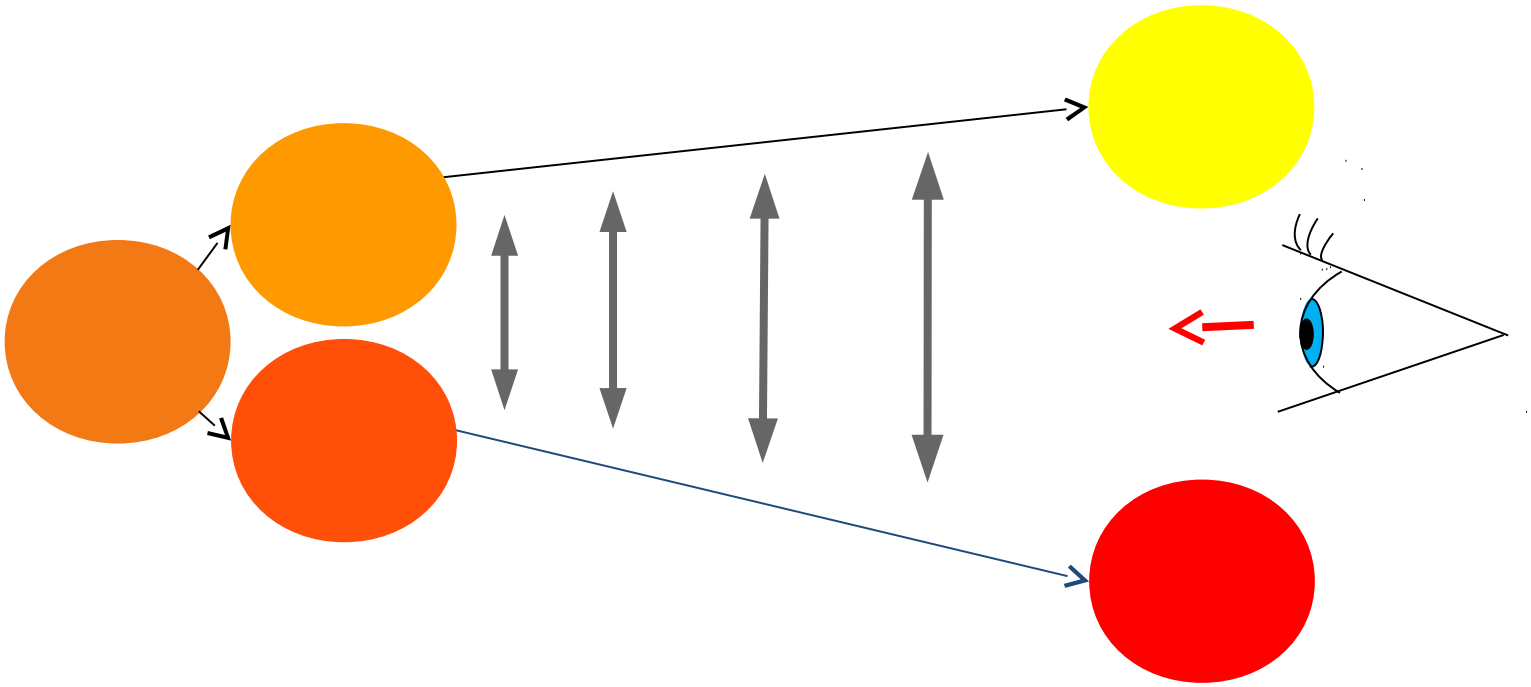
Frequency in population B

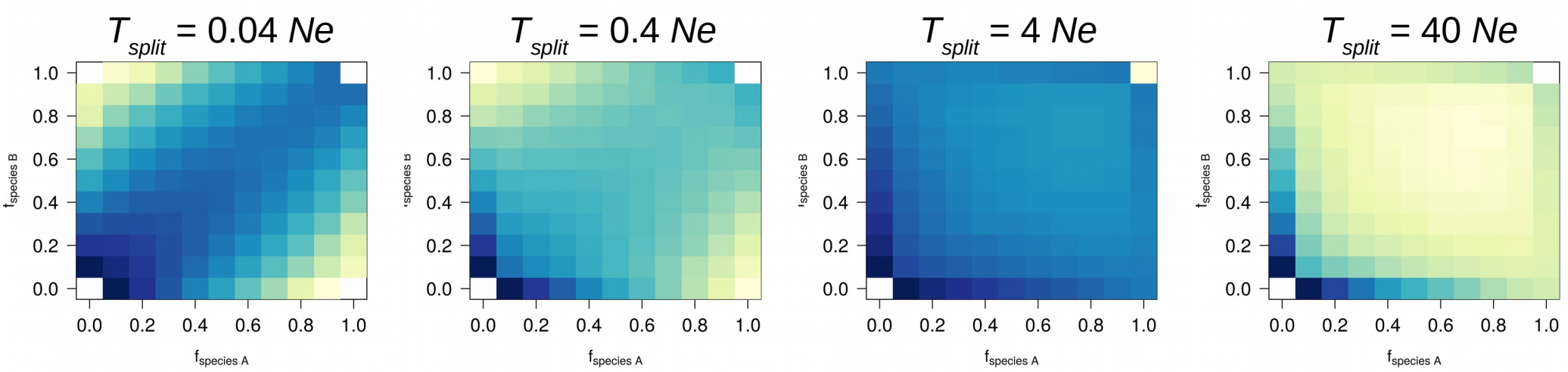
Frequency in population A



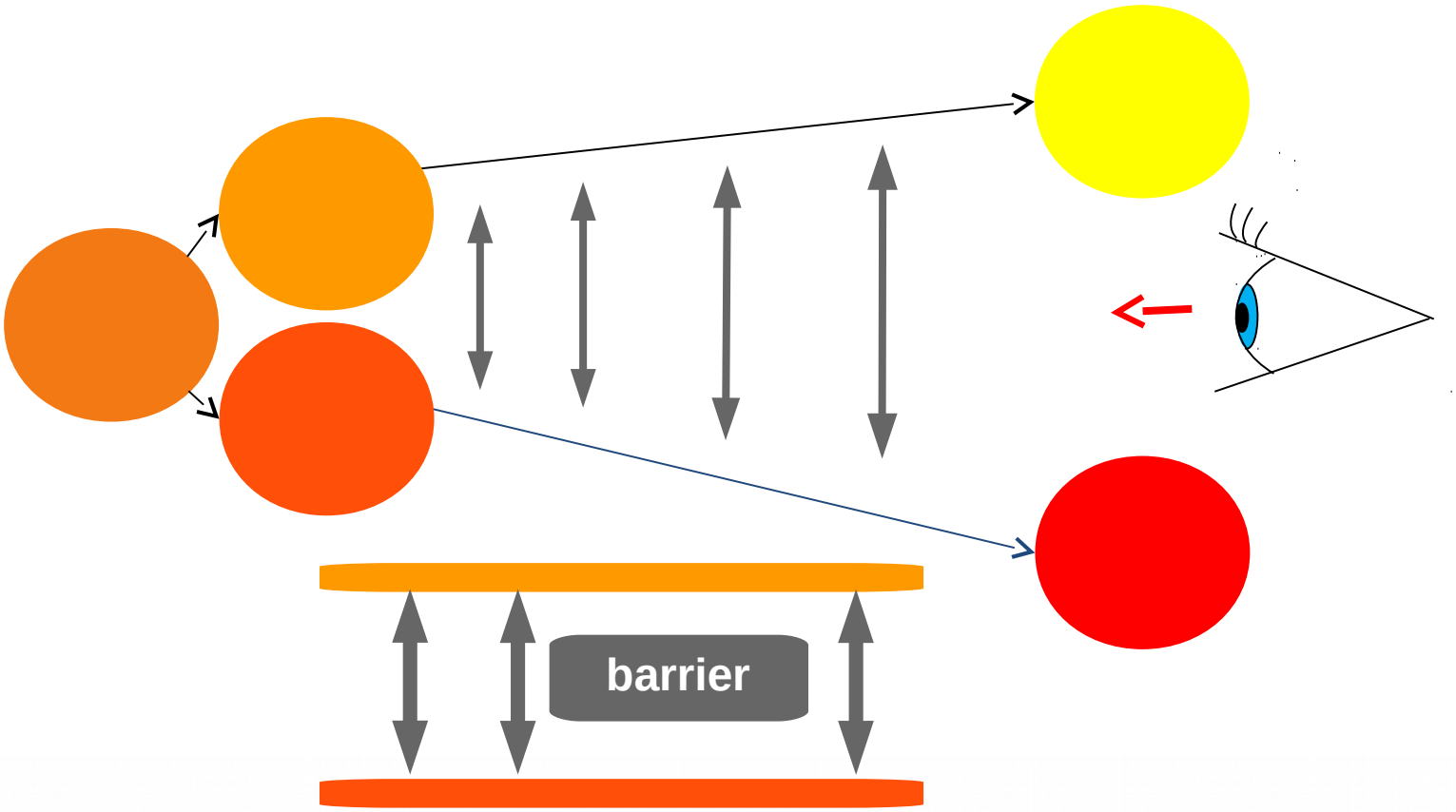


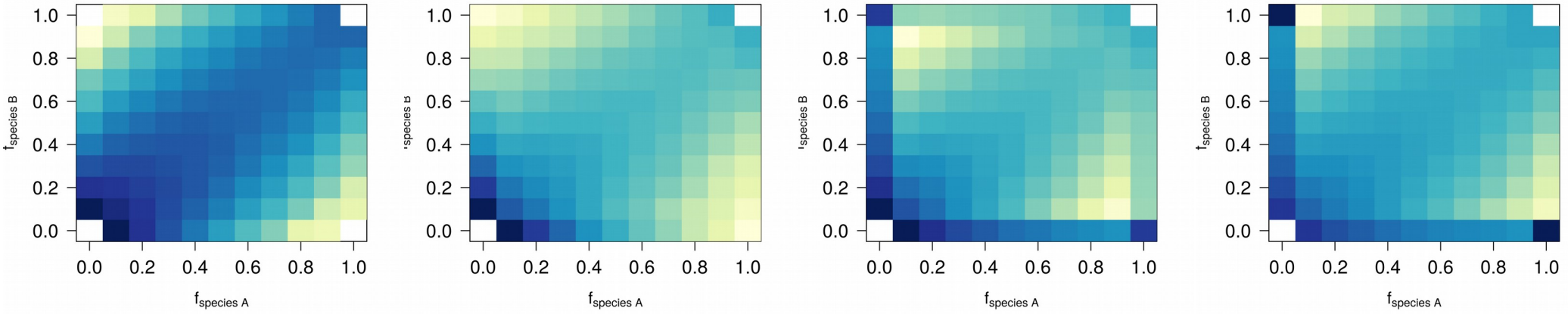

T_{split} = 0.04 *Ne*
T_{split} = 0.4 *Ne*
T_{split} = 4 *Ne*
T_{split} = 40 *Ne*
f species B
f species A
0.0
0.2
0.4
0.6
0.8
1.0





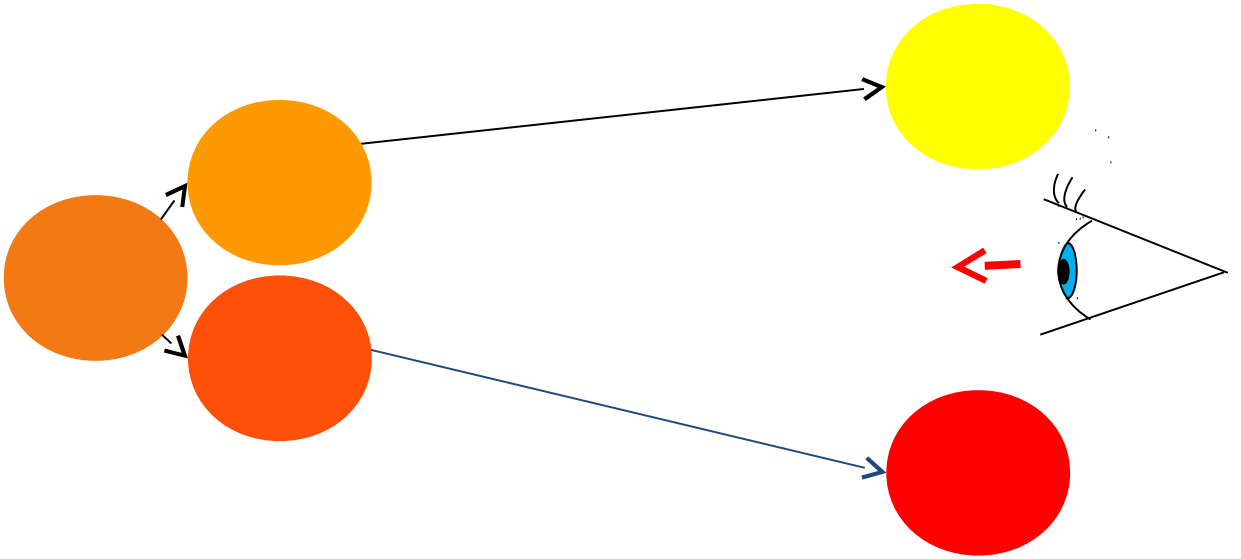
N.m = 0.25 (individual every four generations)

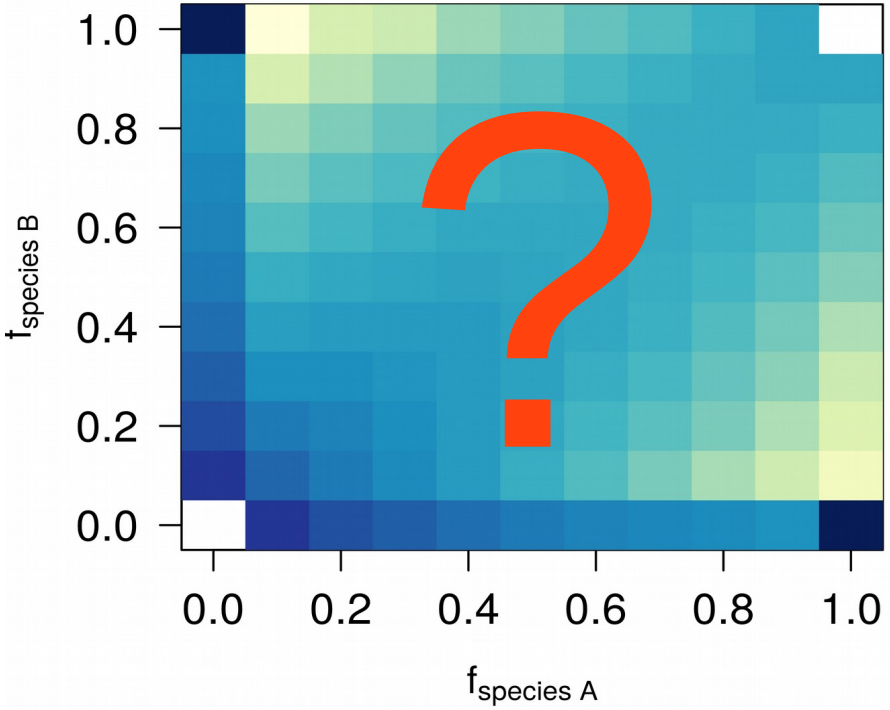




N.m = 0.25 (individual every four generations)

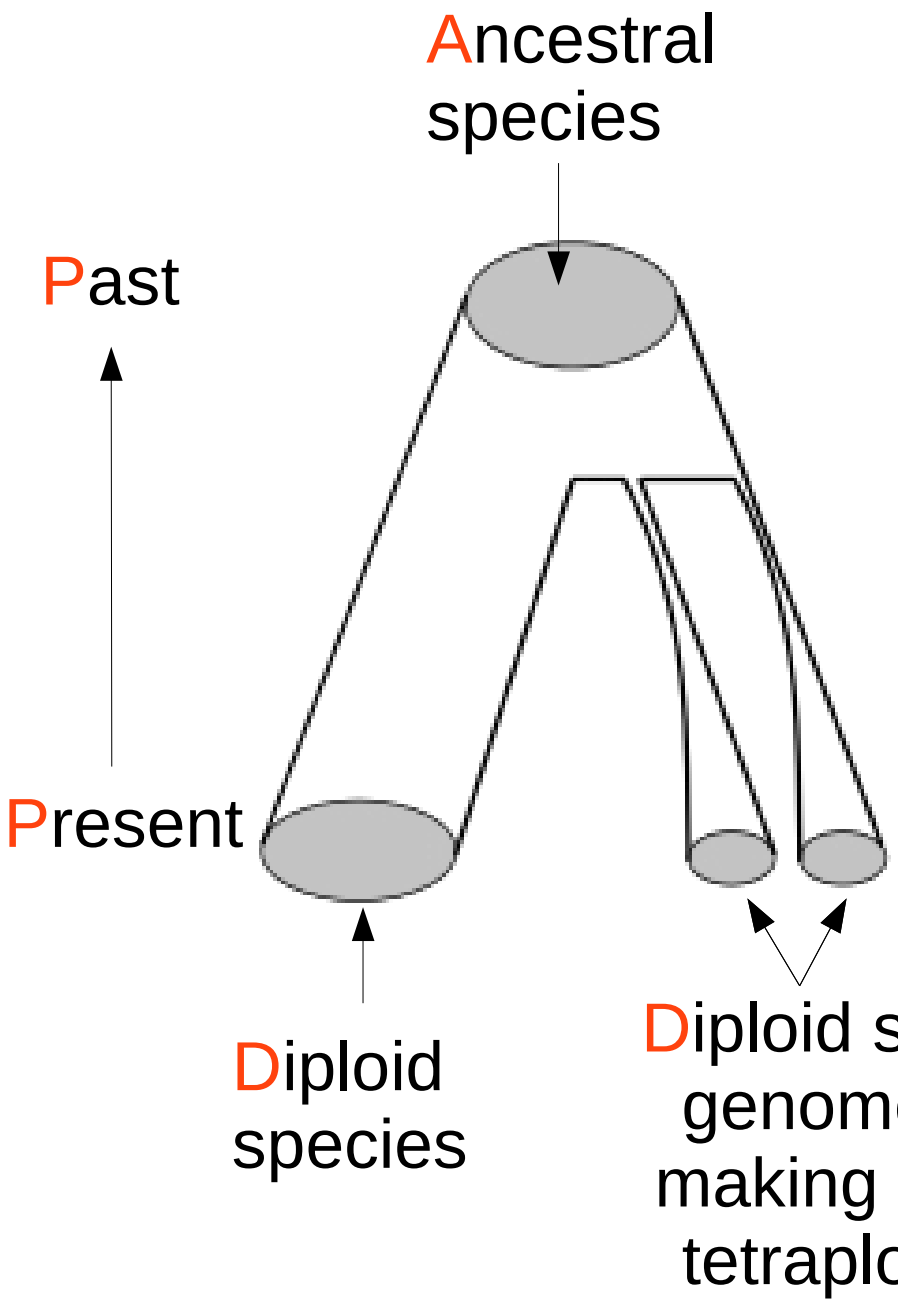
50 % of the genome linked to a barrier

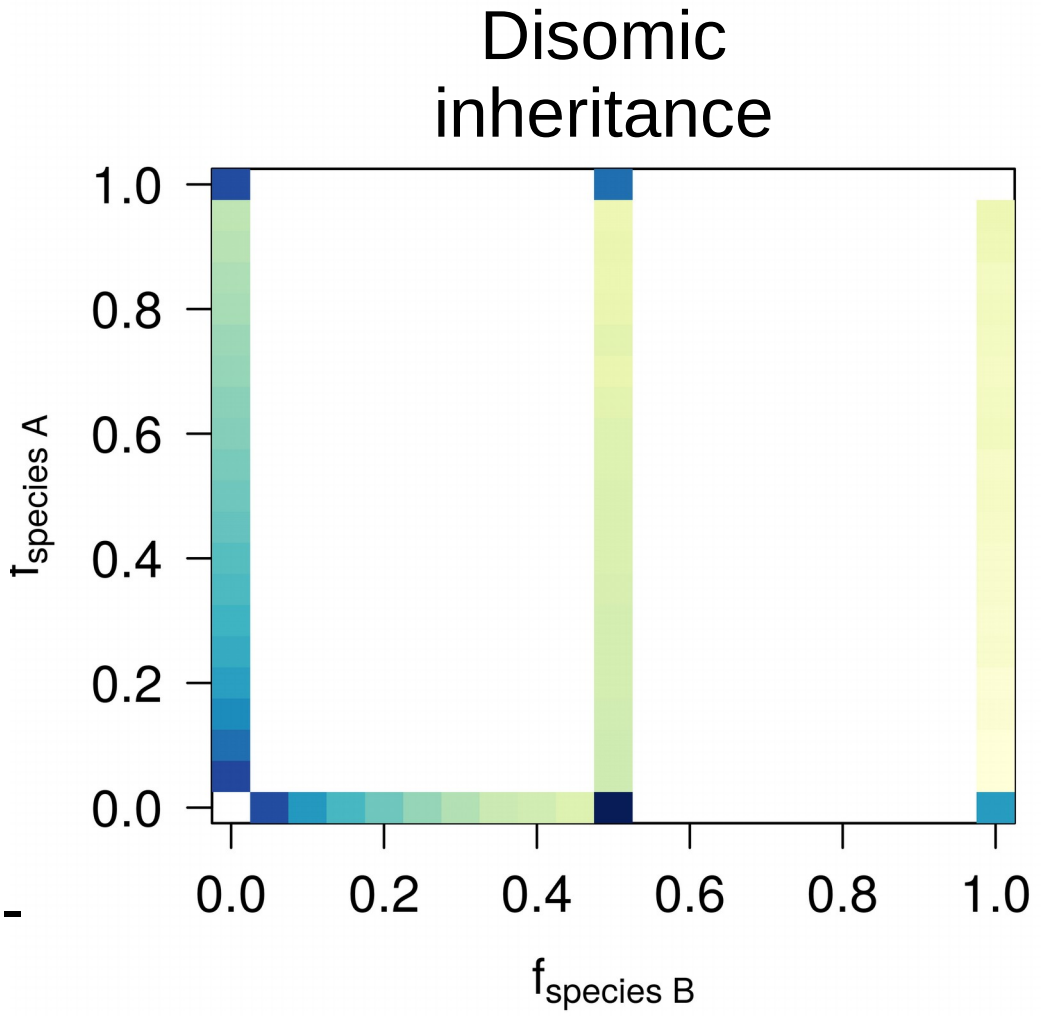


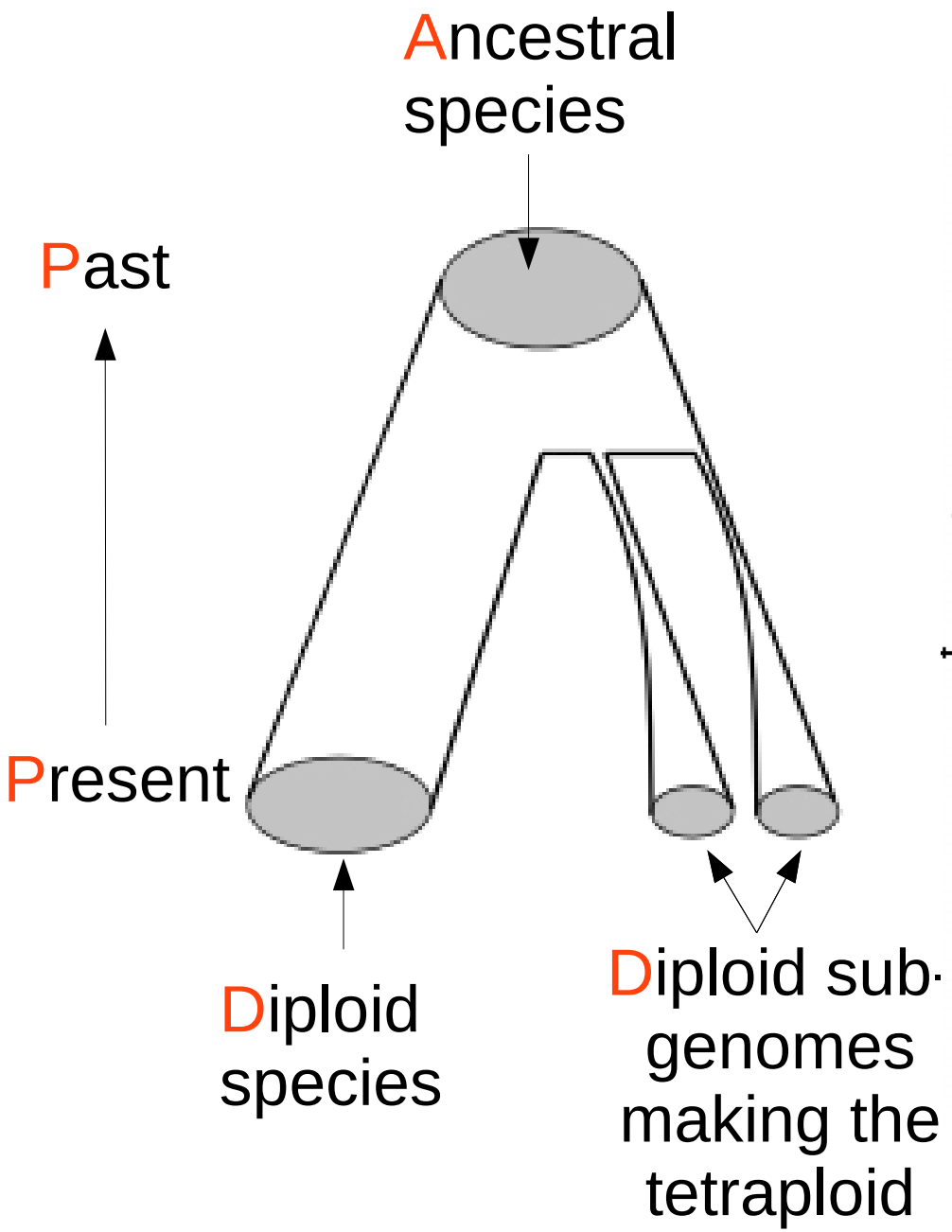


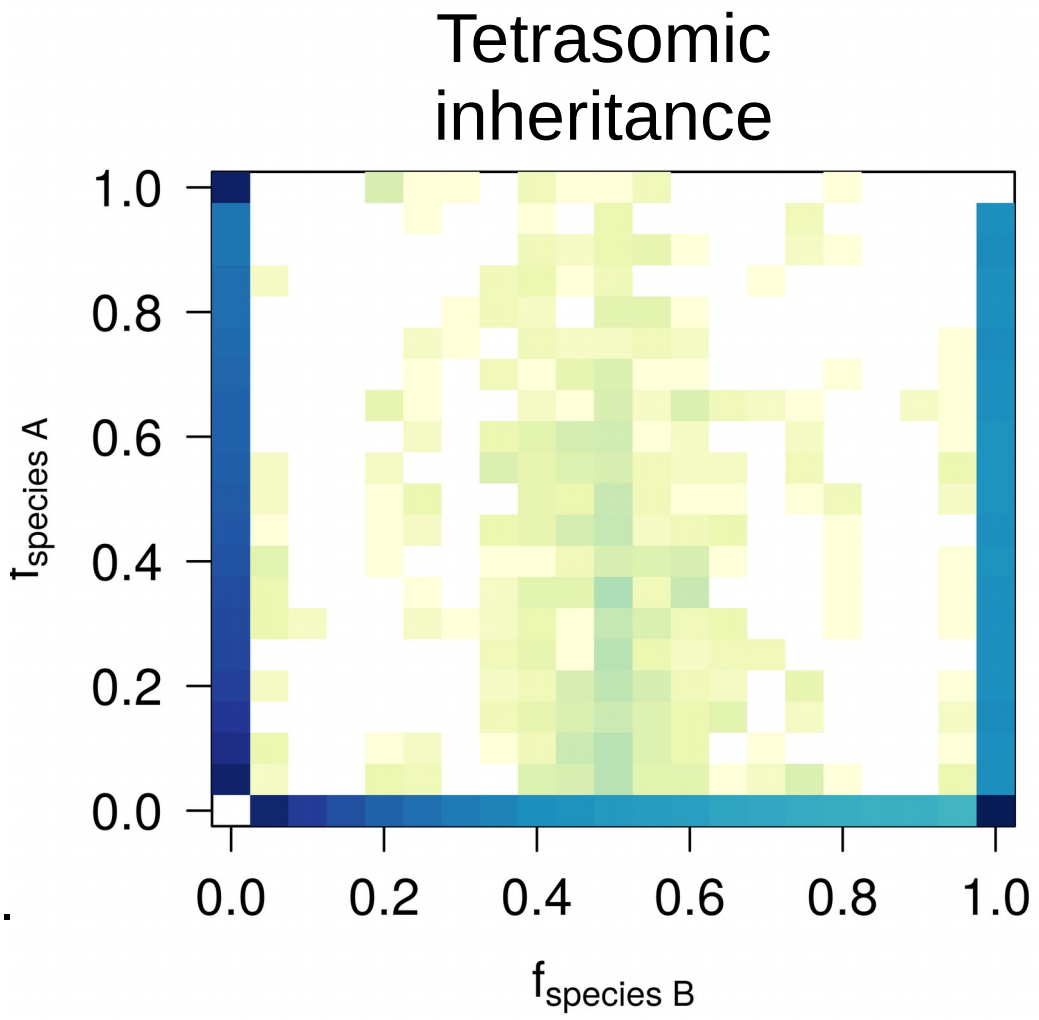
Your job → thinking about jSFS in tetraploids :

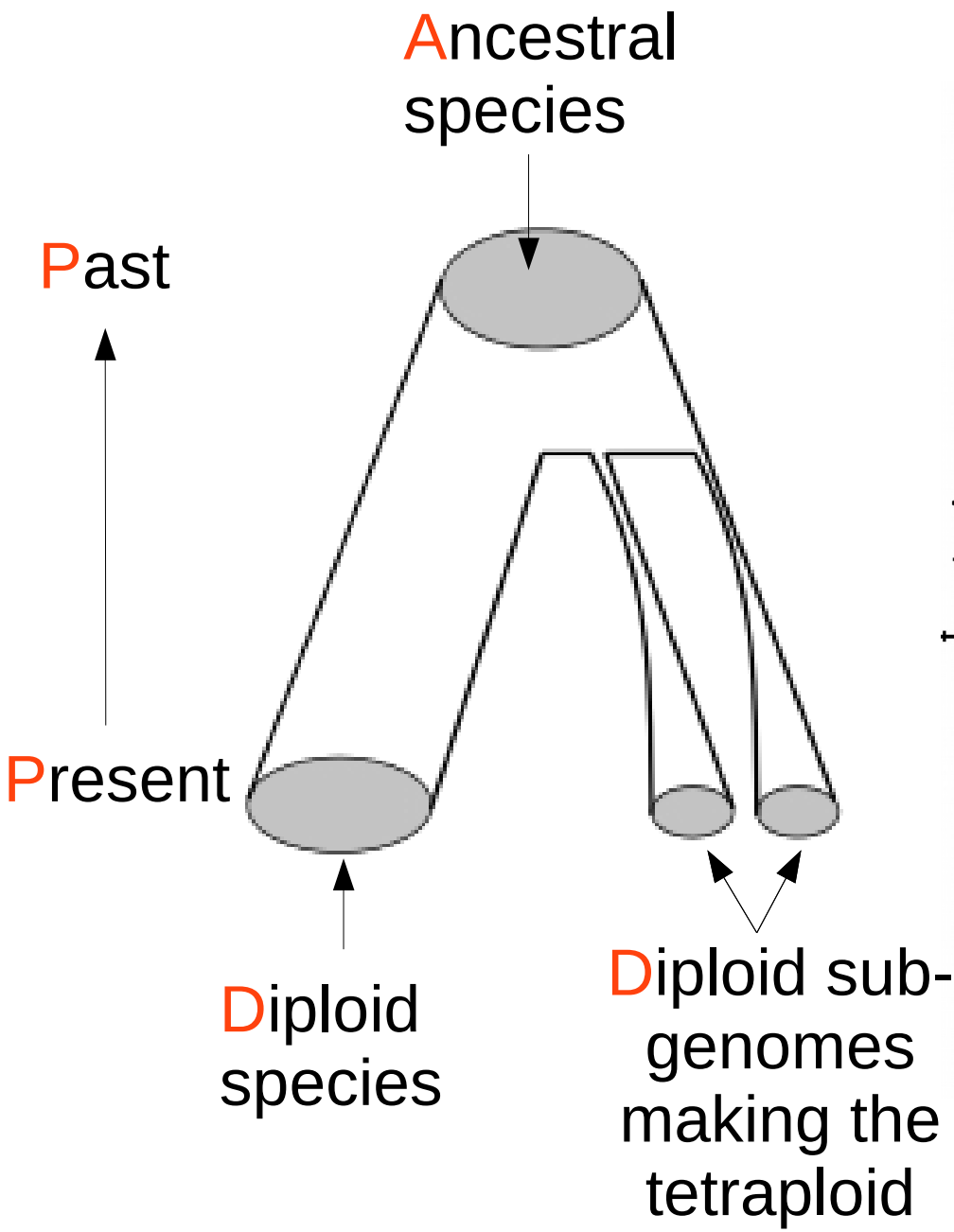
1) auto versus allo ?
2) disomic versus tetrasomic ?
3) intermediate modes of inheritance ?

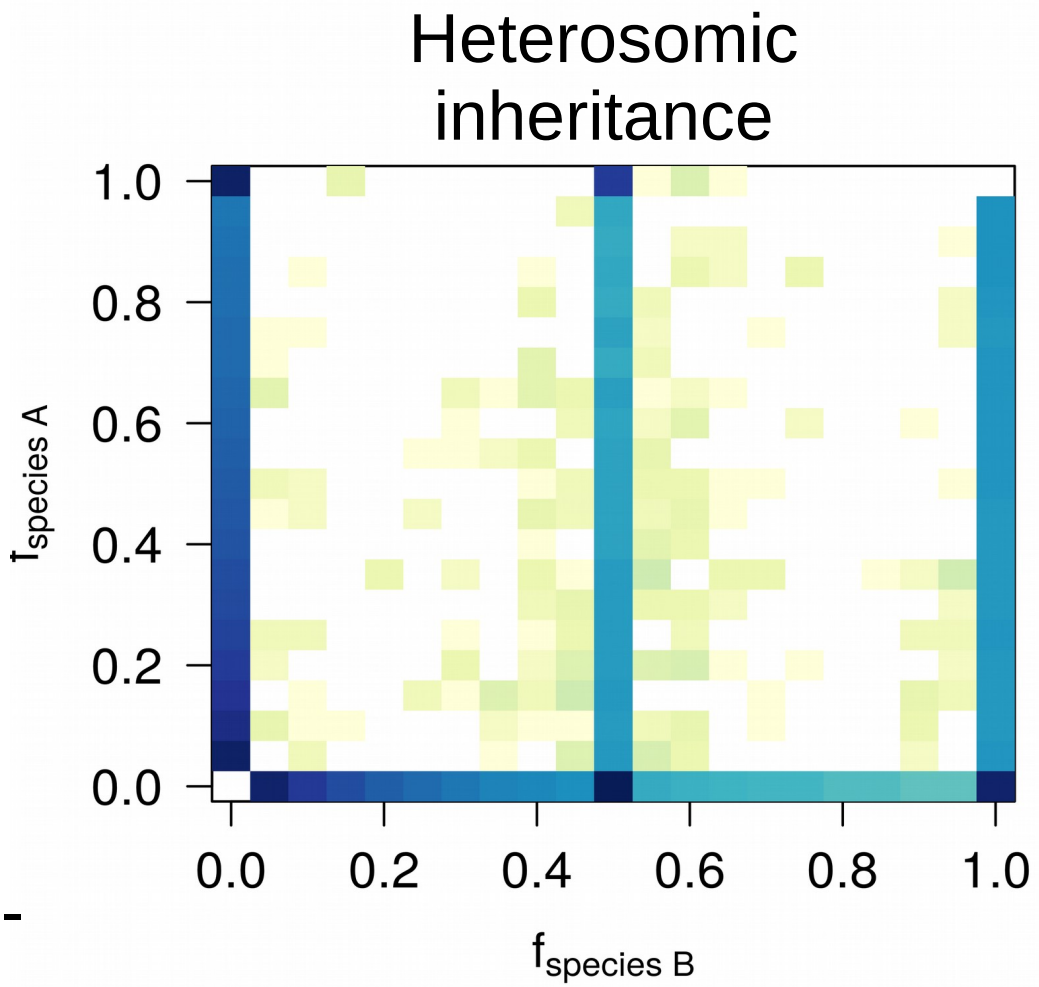

Ancestral species
Past
Present
Diploid species
Diploid sub-genomes making the tetraploid

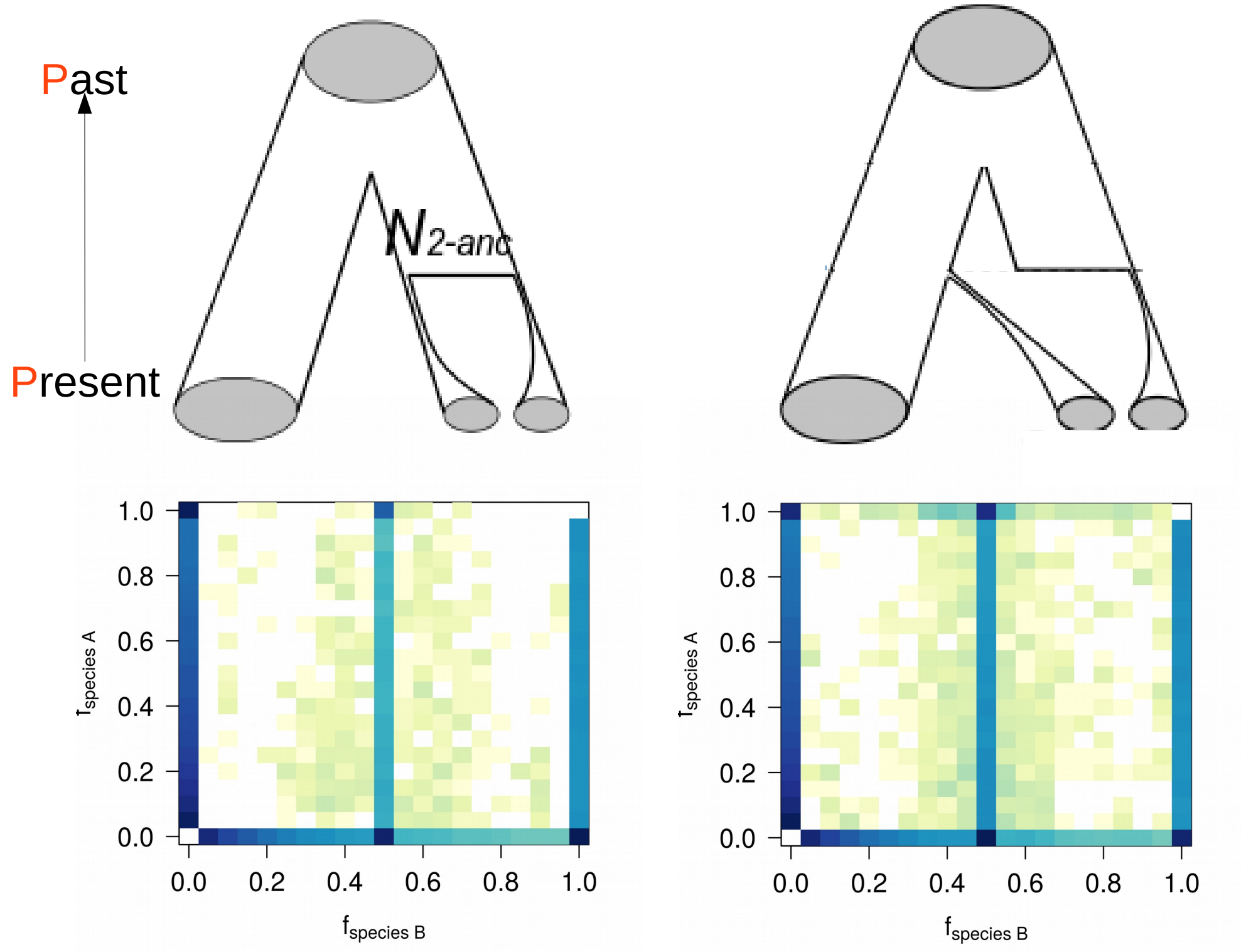

Disomic inheritance
1.0
0.8
0.6
0.4
0.2
0.0
$t_{species\ A}$
0.0
0.2
0.4
0.6
0.8
1.0
$f_{species\ B}$


Ancestral species
Past
Present
Diploid species
Diploid sub-genomes making the tetraploid


Tetrasomic inheritance
1.0
0.8
0.6
0.4
0.2
0.0
$t_{species\ A}$
0.0
0.2
0.4
0.6
0.8
1.0
$f_{species\ B}$


Ancestral species
Past
Present
Diploid species
Diploid sub-genomes making the tetraploid


Heterosomic inheritance
1.0
0.8
0.6
0.4
0.2
0.0
$f_{species\ A}$
0.0
0.2
0.4
0.6
0.8
1.0
$f_{species\ B}$


Past
Present
N2-anc
1.0
0.8
0.6
0.4
0.2
0.0
fspecies A
0.0
0.2
0.4
0.6
0.8
1.0
fspecies B

If the **current genetic data** are impacted by the **demographic history**

We can make inferences about the **demographic history** from the **current genetic data**

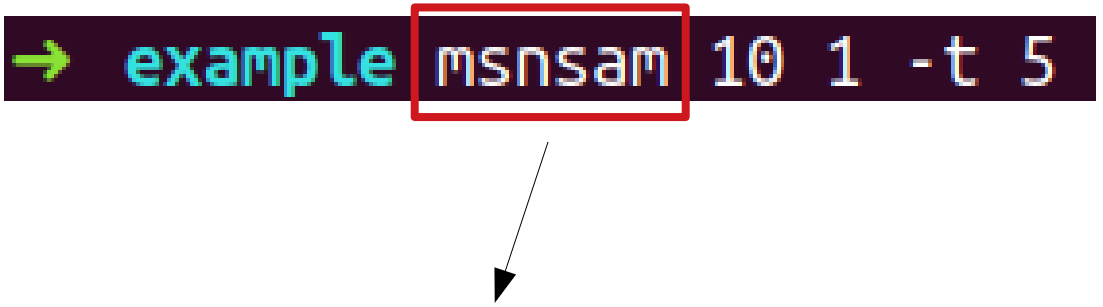
The first step : simulating genetic dataset under various demographic models

The first step : simulating genetic dataset under various demographic models

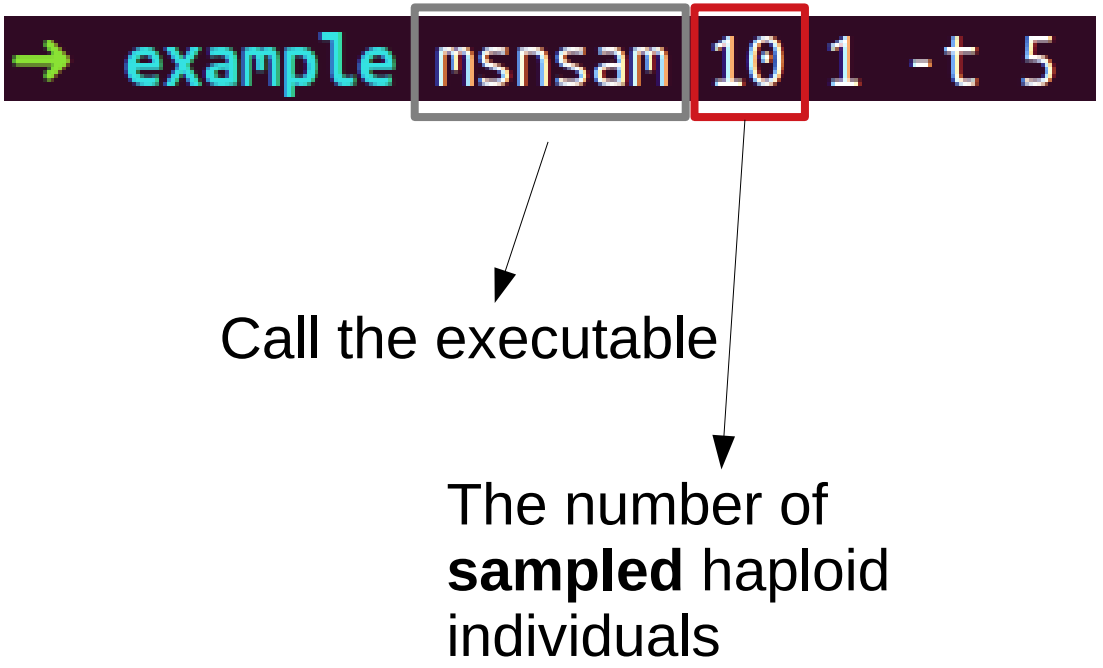
we need a **fast simulator**

Here → a coalescent one (named ms or msnsam)

The basic command line for ms



The basic command line for ms



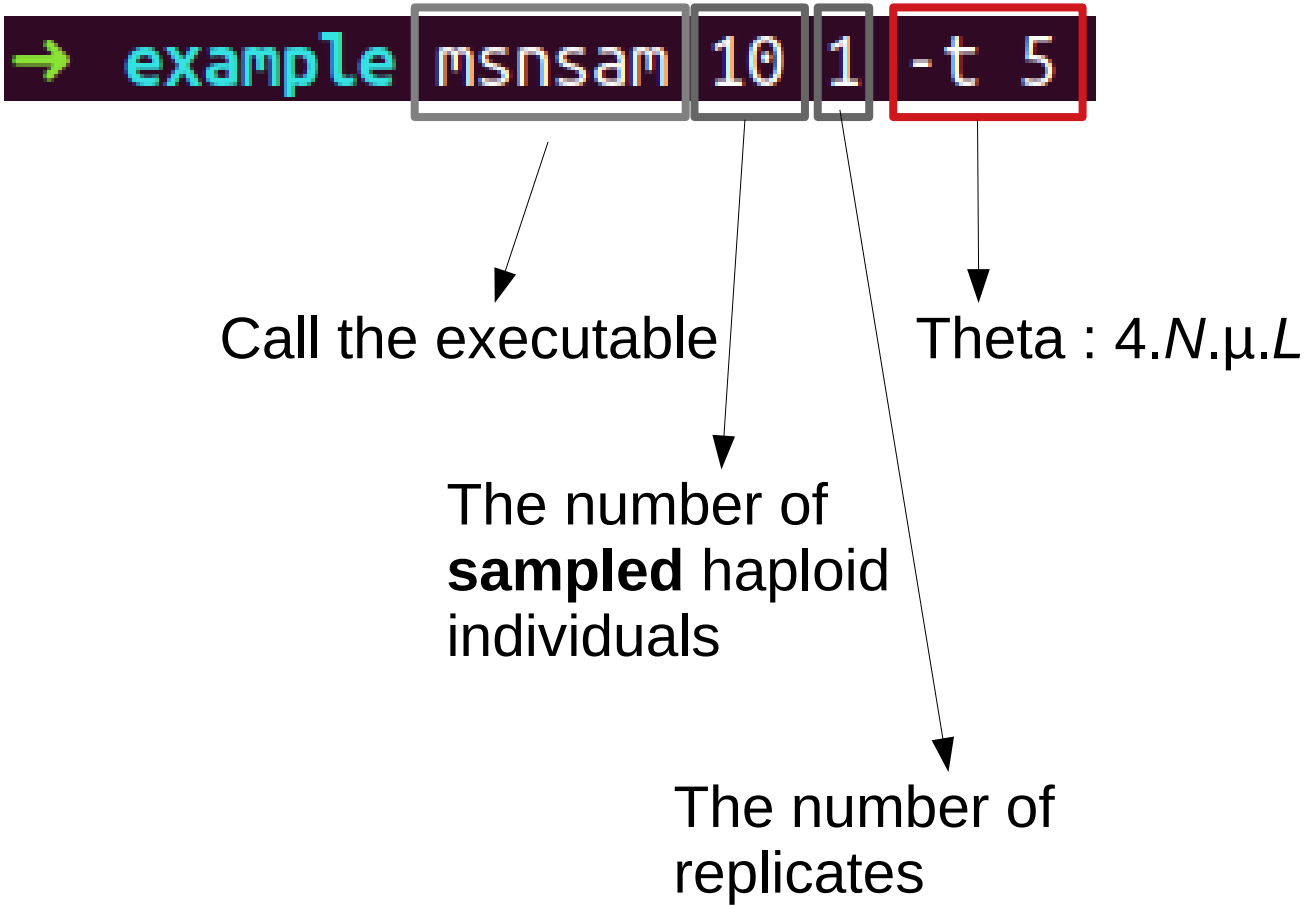
The basic command line for ms

Call the executable

The number of **sampled** haploid individuals

The number of replicates

The basic command line for ms

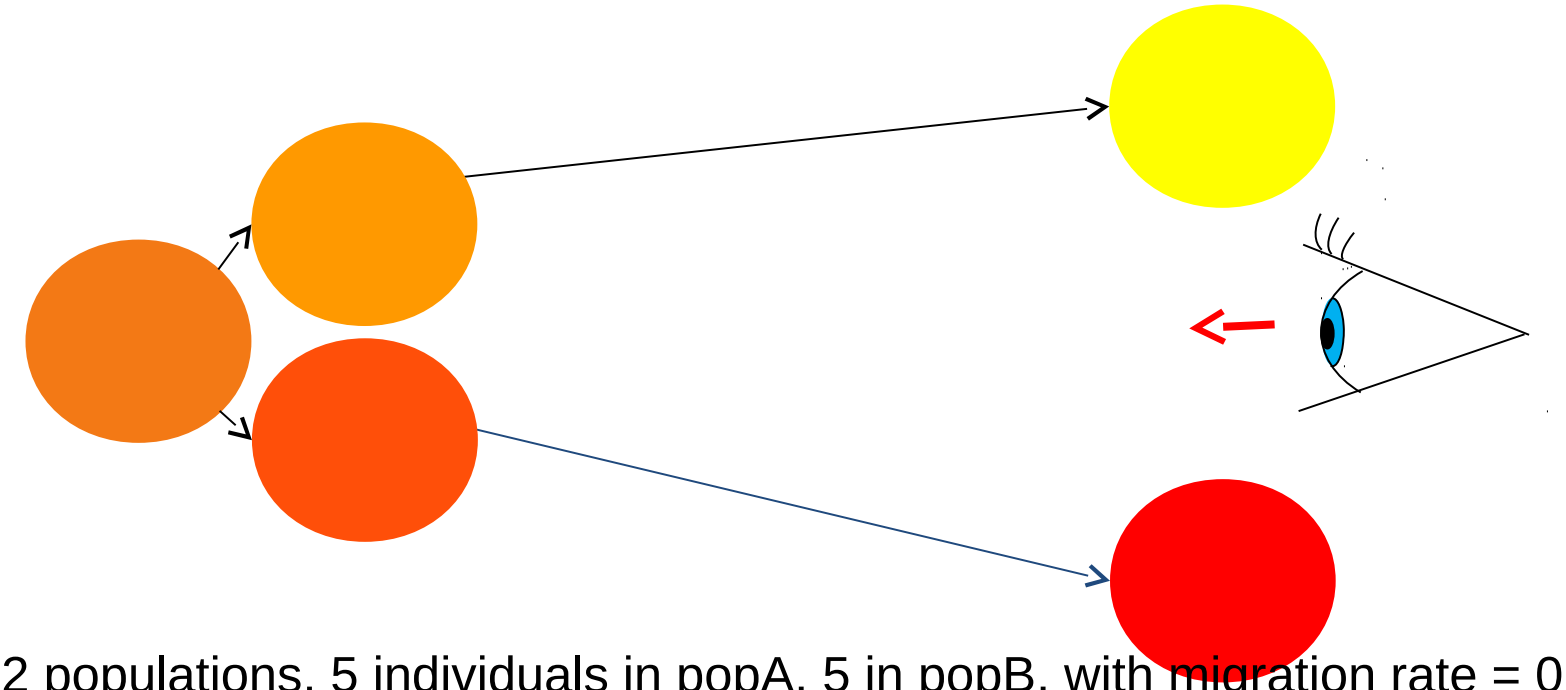


The basic command line for ms

```
→ example msnsam 10 1 -t 5
msnsam 10 1 -t 5
3579 27011 59243

//
segsites: 29
positions: 0.0095 0.0282 0.0496 0.0863 0.0906 0.1207 0.1535 0.1580 0.1861 0.2220 0.3254 0.3648 0.3797 0.3868 0.4242 0.5360 0.6244 0.632
9 0.6762 0.6978 0.6980 0.7396 0.7769 0.7870 0.7977 0.8215 0.8217 0.9178 0.9410
00000101000110011000000000000
10000000010001000101010000100
01101110001000101010000111010
00000101000110011000000000001
10000000010001000101010000100
00000101000110011000000000000
00000101100110011000000000000
00010101000110011000000000000
00000101000110011000101000000
00000101000110011000000000001
```

Adding population structure

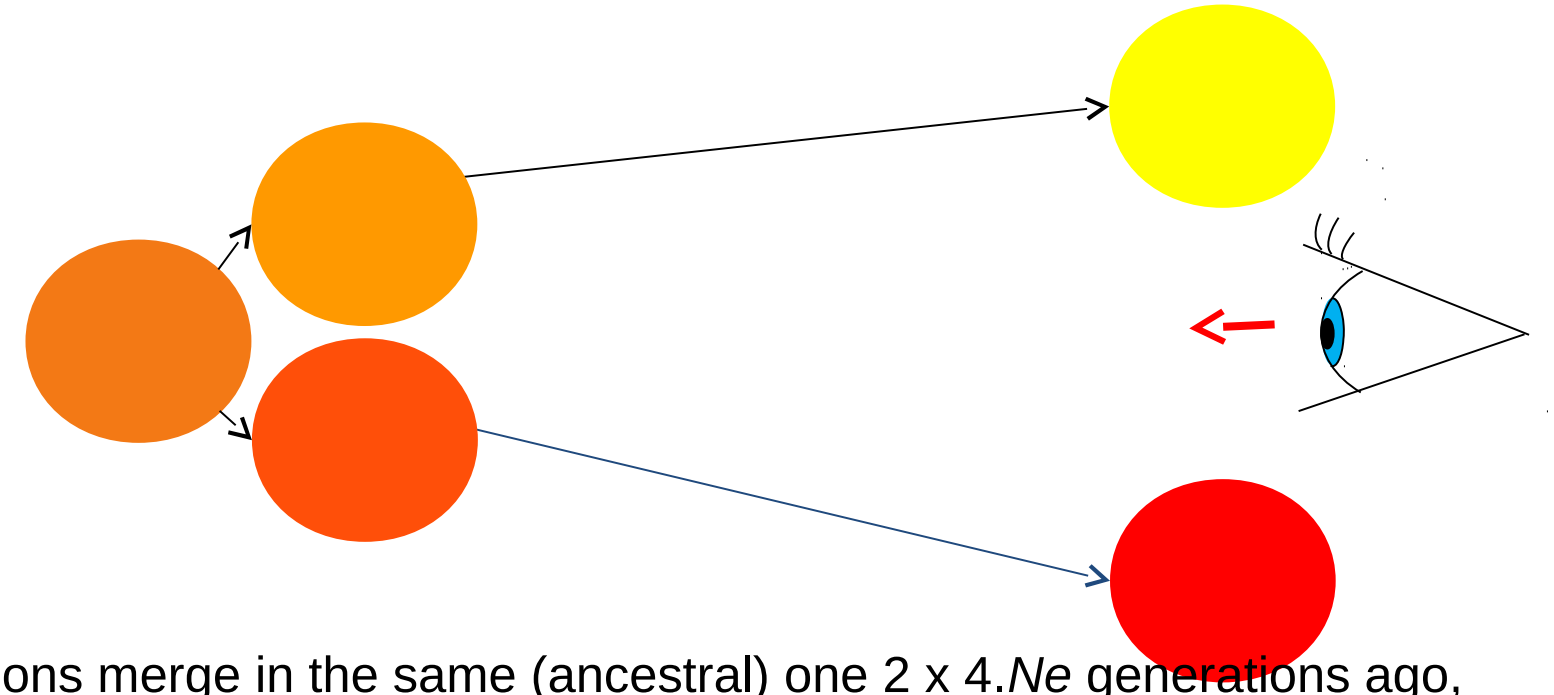


An island of 2 populations, 5 individuals in popA, 5 in popB, with migration rate = 0

```
→ example msnsam 10 1 -t 5 -I 2 5 5 0 -ej 2 1 2
msnsam 10 1 -t 5 -I 2 5 5 0 -ej 2 1 2
20019 47964 40917

//
segsites: 31
positions: 0.0322 0.0763 0.1269 0.1434 0.1535 0.1582 0.1744 0.2082 0.2146 0.3203 0.3704 0.3707 0.3824 0.4339 0.4914 0.4958 0.5401 0.570
3 0.6041 0.6540 0.6564 0.7023 0.7477 0.7960 0.8152 0.8171 0.8423 0.8722 0.8852 0.8980 0.9675
1001100000000000010001100000000
1001100000000000000001000000010
1001100100000000100011000001001
1001100000100000000001000000010
1101100000000000000001000000010
0010000010001000001000000010100
0010000001000000001000011100100
0010010000010111001000000000100
0010001001000000001100010100100
0010000001000000001000011100100
```

Adding population structure

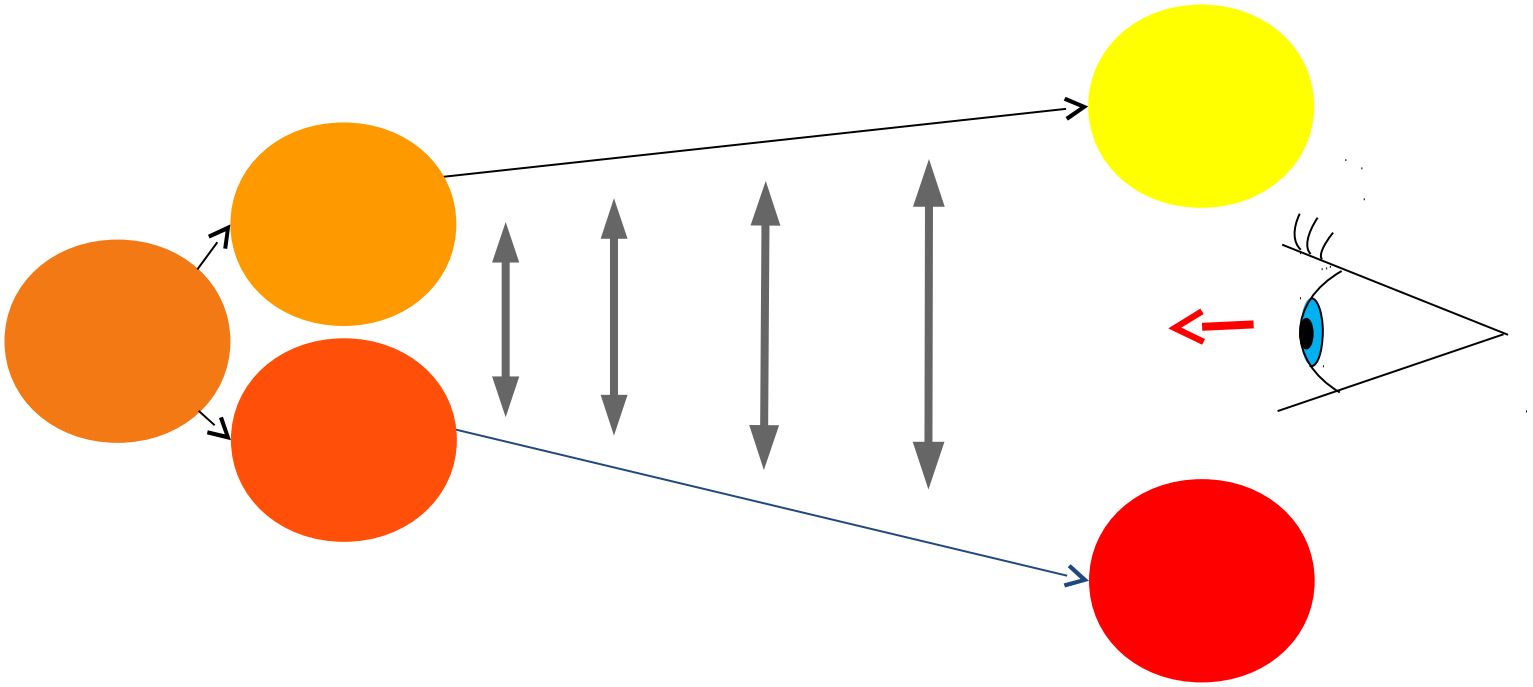


The populations merge in the same (ancestral) one 2 x 4.*Ne* generations ago,

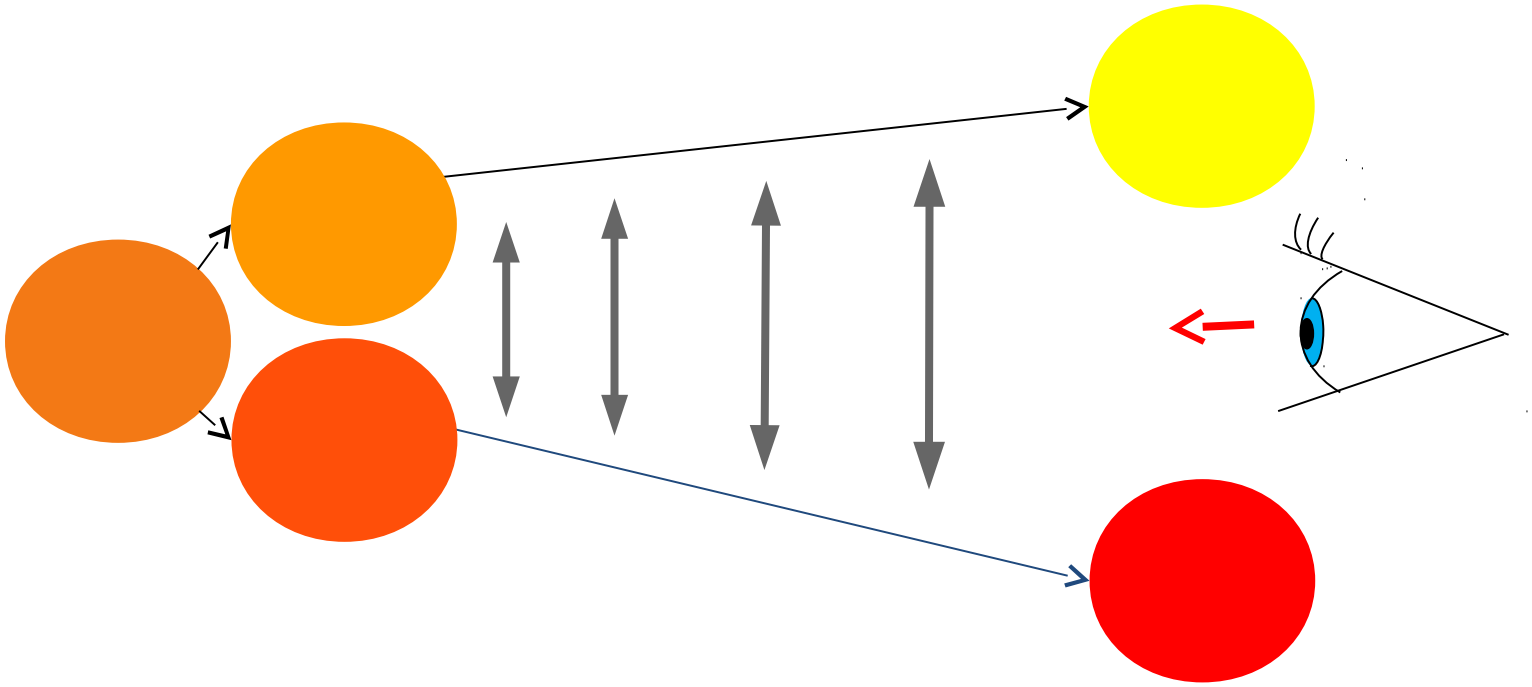
```
→ example msnsam 10 1 -t 5 -I 2 5 5 0 -ej 2 1 2
msnsam 10 1 -t 5 -I 2 5 5 0 -ej 2 1 2
20019 47964 40917

//
segsites: 31
positions: 0.0322 0.0763 0.1269 0.1434 0.1535 0.1582 0.1744 0.2082 0.2146 0.3203 0.3704 0.3707 0.3824 0.4339 0.4914 0.4958 0.5401 0.570
3 0.6041 0.6540 0.6564 0.7023 0.7477 0.7960 0.8152 0.8171 0.8423 0.8722 0.8852 0.8980 0.9675
1001100000000000010001100000000
1001100000000000000001000000010
1001100100000000100011000001001
1001100000100000000001000000010
1101100000000000000001000000010
0010000010001000001000000010100
0010000001000000001000011100100
0010010000010111001000000000100
0010001001000000001100010100100
0010000001000000001000011100100
```

Adding population structure



Adding population structure



```
→ example msnsam 10 1 -t 5 -I 2 5 5 10 -ej 2 1 2
msnsam 10 1 -t 5 -I 2 5 5 10 -ej 2 1 2
3754 64515 32205

//
segsites: 27
positions: 0.0747 0.1319 0.1320 0.1668 0.1935 0.2148 0.2855 0.3563 0.3733 0.3841 0.4010 0.4368 0.4557 0.4892 0.5066 0.5160 0.5404 0.540
6 0.5681 0.5910 0.6501 0.6986 0.7217 0.7515 0.7575 0.8053 0.9036
010010010000001000000000000
010110010000000001000010100
010010010001000000100100000
010010010010000010000010111
010010010000001000000000000
010010010001000000100100000
010010010010000010000010110
010010010000000000000010100
101001101100110100011001000
010010010010000010000010110
```

For Ubuntu us<3rs

```
github git:(master) ✗ ./run_ABC_polyploid.py

     run_ABC_polyploid.py is a python script usefull to run some random simulations of 9 models
     of speciation between a diploid and a tetraploid
     It takes 3 arguments: 1) a demographic model; 2) the name of a file containing the arguments; 3) the name of the bpfile
     The accepted models are:
             diso1
             diso2_auto
             diso2_allo
             tetra1
             tetra2_auto
             tetra2_allo
             hetero1
             hetero2_auto
             hetero2_allo

     command line: ./run_ABC_polyploid.py [model] [name of the argfile] [bpfile]



     The expected number of arguments is 3
```

For Mac u$ers

```
ub git:(master) ✗ ./run_ABC_polyploid_v2.py

run_ABC_polyploid.py is a python script usefull to run some random simulations of models
of speciation between a diploid and a tetraploid
It takes 4 arguments:
        1) a demographic model (auto, allo);
        2) a model of inheritance (disomic, heterosomic, tetrasomic);
        3) the name of the bpfile;
        4) the number of replicates (an integer)

The accepted models are:
        auto
        allo

command line: ./run_ABC_polyploid.py [model] [inheritance] [bpfile] [nreps]



The expected number of arguments is 4
```

Example : simulating « allopolyploidization » + « tetrasomique »

```
./run_ABC_polyploid.py hetero2_allo  argfile_tetra2_allo.txt bpfile

./run_ABC_polyploid_v2.py allo tetrasomic bpfile 10000
```

Ubuntu

Mac

Example : simulating « allopolyploidization » + « tetrasomic »

```
./run_ABC_polyploid.py hetero2_allo  argfile_tetra2_allo.txt bpfile

./run_ABC_polyploid_v2.py allo tetrasomic bpfile 10000
```

Ubuntu

Mac

Let’s simulate those 4 models !

{ allo ; auto } x { disomic ; tetrasomic}