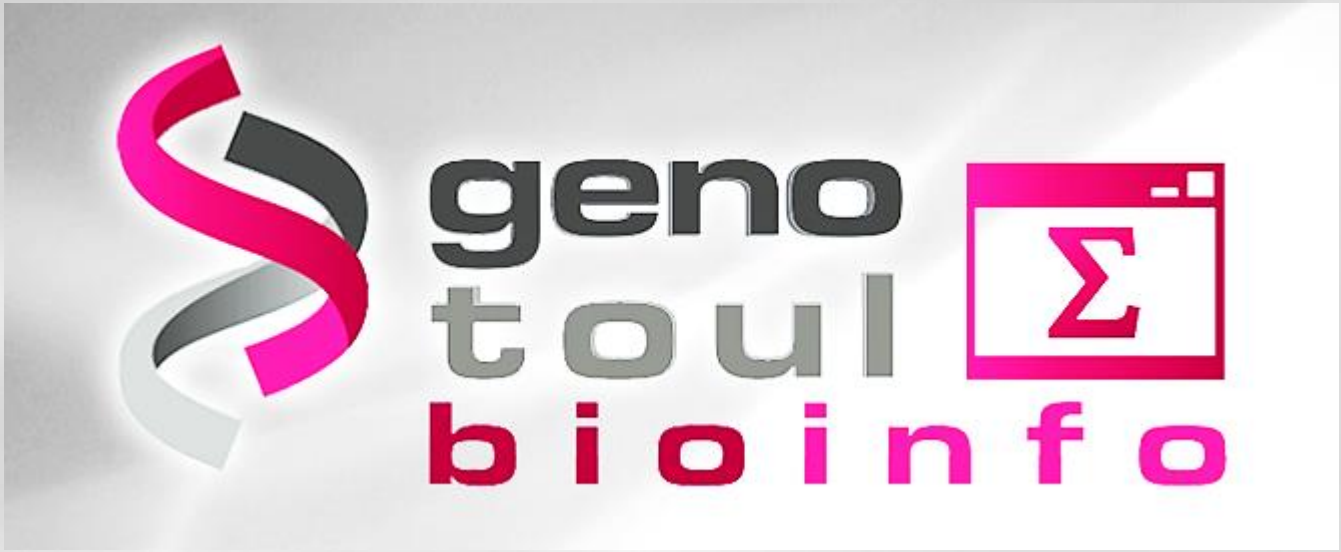
Plateforme Bioinformatique Midi-Pyrénées



Module 4: les pressions de sélection

Géraldine PASCAL

FeedBack: Pourquoi vous intéressez-vous où aves-vous besoin des calculs de pressions de sélection ?

Qu'allons-nous voir ?

1. **Introduction général sur les pression de sélection**
2. **Présentation du logiciel codeml**
3. **Le fichier de contrôle codeml.ctl**
4. **La vraisemblance**
5. **Les modèles**
    1. **Exo 1**
    2. **Exo 2**
6. **Calculs sites / branch / branch-sites**
7. **LRT**
    1. **Exo 3**
    2. **Exo 4**
8. **BEB**

On évolue

Les variations phénotypiques héréditaires émergent dans les populations par mutation génétique.

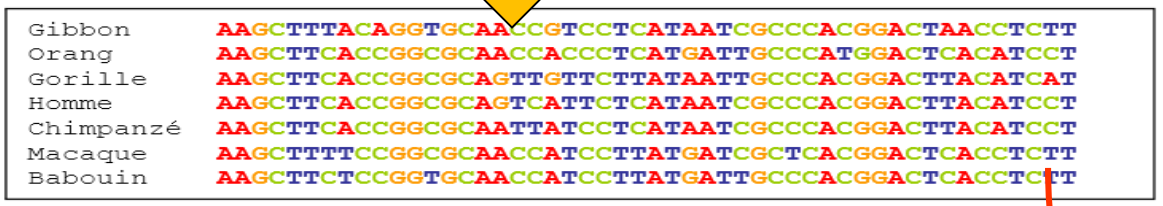
Et on évolue, parce qu'on mute.

La plupart du temps les gènes (→ protéines) restent stables mais parfois on observe de nombreuses modifications:
gènes immunitaires, reproduction, pathogénicité, symbiose.

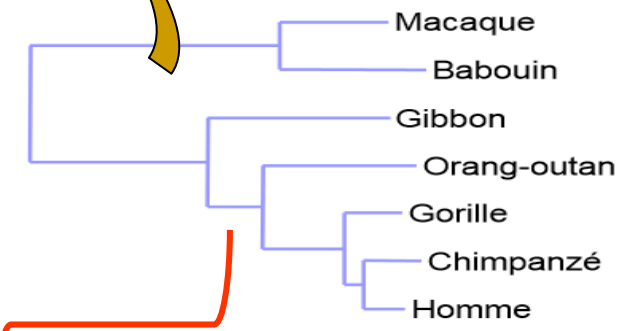
Mutation avantageuse ou désavantageuse.
Mutation maintenue ou perdue.

Schéma général

Fichier fasta de séquences

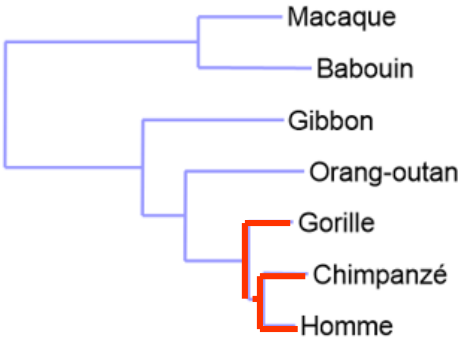


Alignement



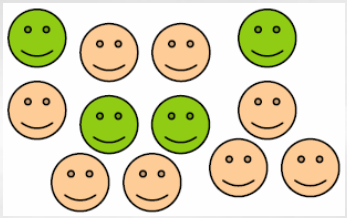
Arbre

Modèle + tests

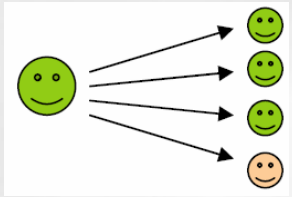


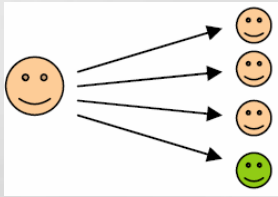
Qu'est-ce que la sélection positive ?

I. trait variable

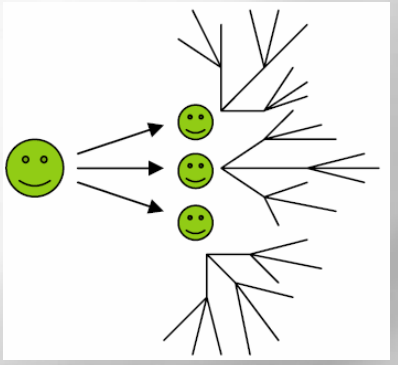


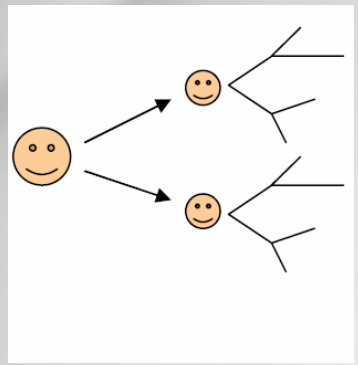
II. héritable





III. conférant un plus grand succès reproducteur





Exemple de sélection positive

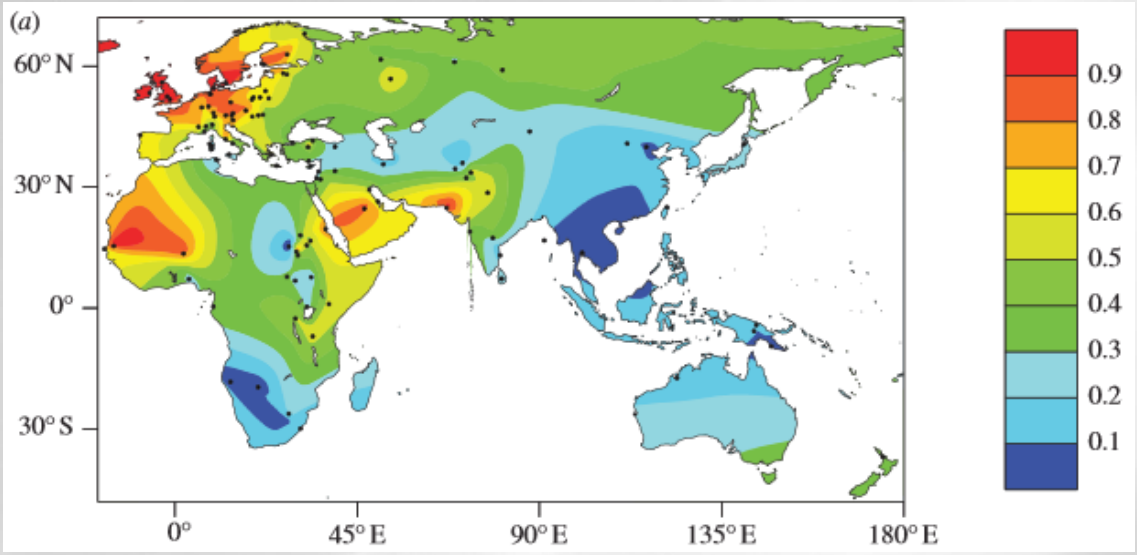
L'exemple probablement le mieux documenté dans l'évolution récente de l'homme est le cas de la persistance de la lactase (Bersaglieri et al. 2004; Tishkoff et al. 2007).

La persistance à la lactase

Le gène codant la lactase sur le chromosome 2 humain s'exprime chez la plupart des individus africains et asiatiques seulement chez le nouveau né, mais s'éteint après le sevrage.

L'enzyme lactase exprimée permet au nouveau né de métaboliser le lactose contenu dans le lait maternel.

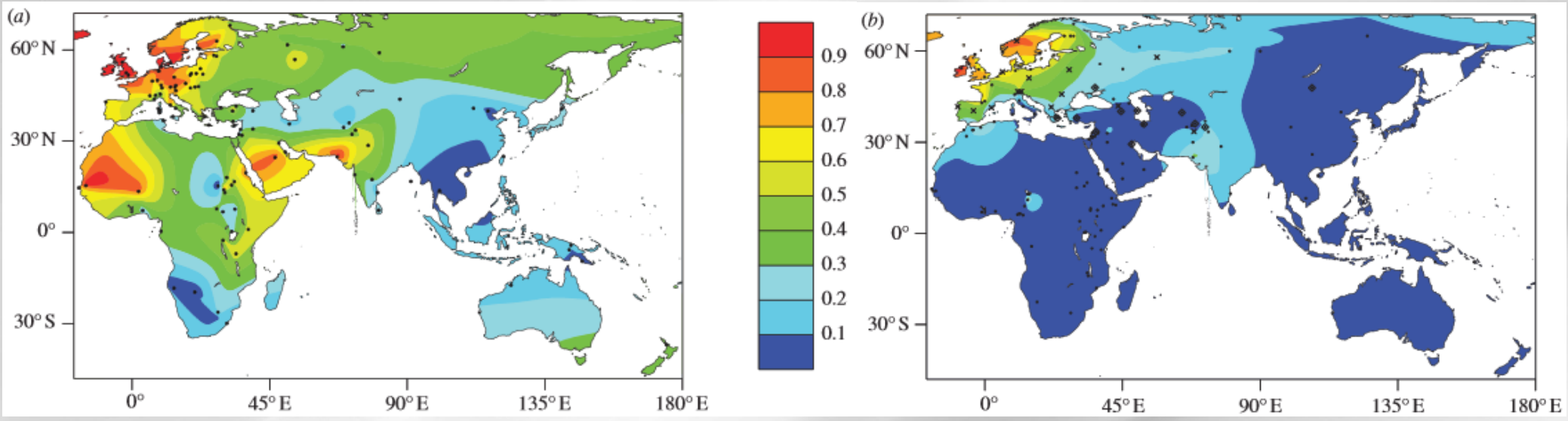
Distribution de la « lactose persistance » dans le monde



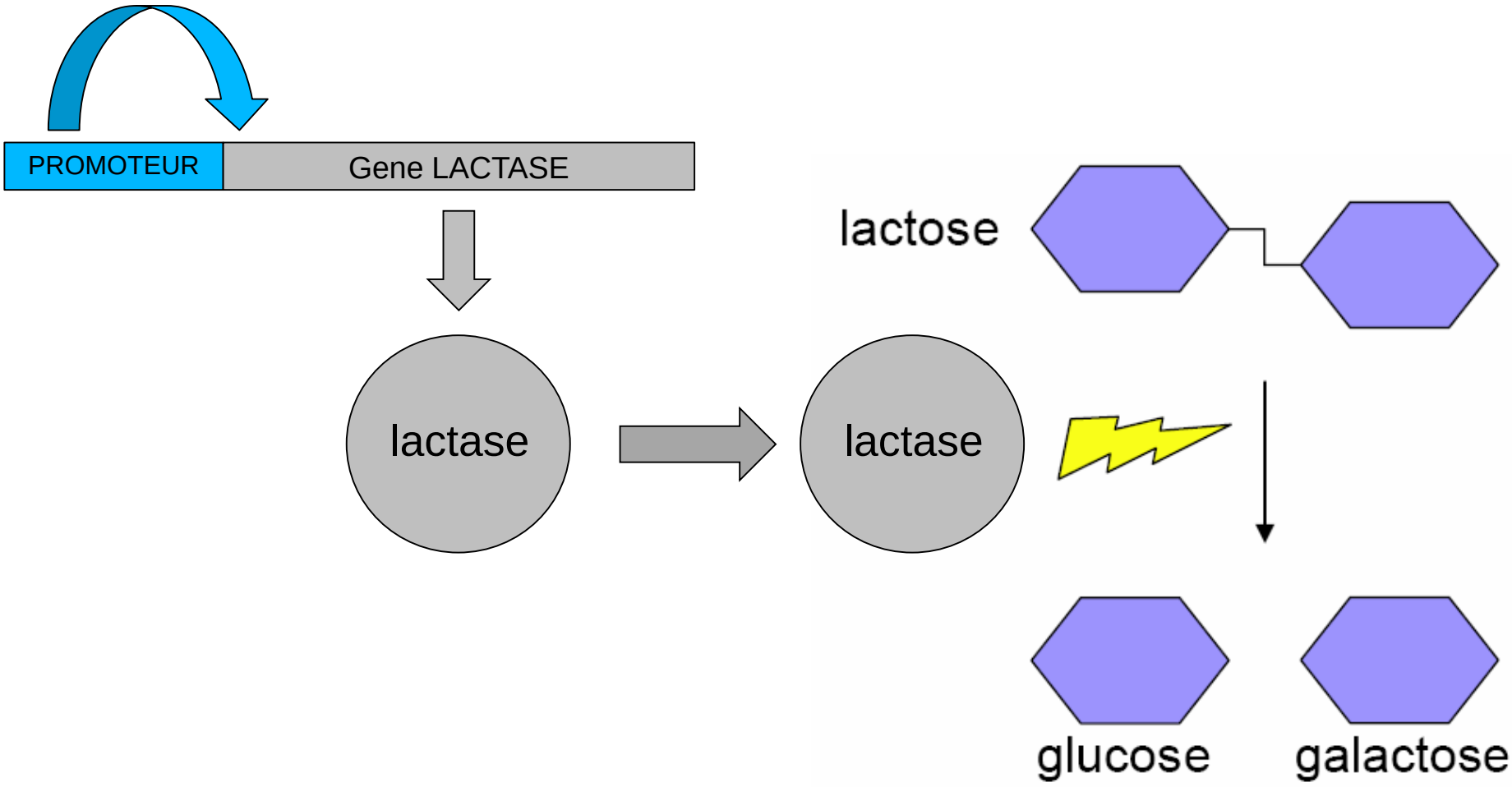
Chez 75% des individus européens et chez certaines populations africaines, on observe une persistance de l'expression de la lactase tout au long de la vie, et par conséquent de la capacité à métaboliser le lactose.

Comment l'expliquer ?

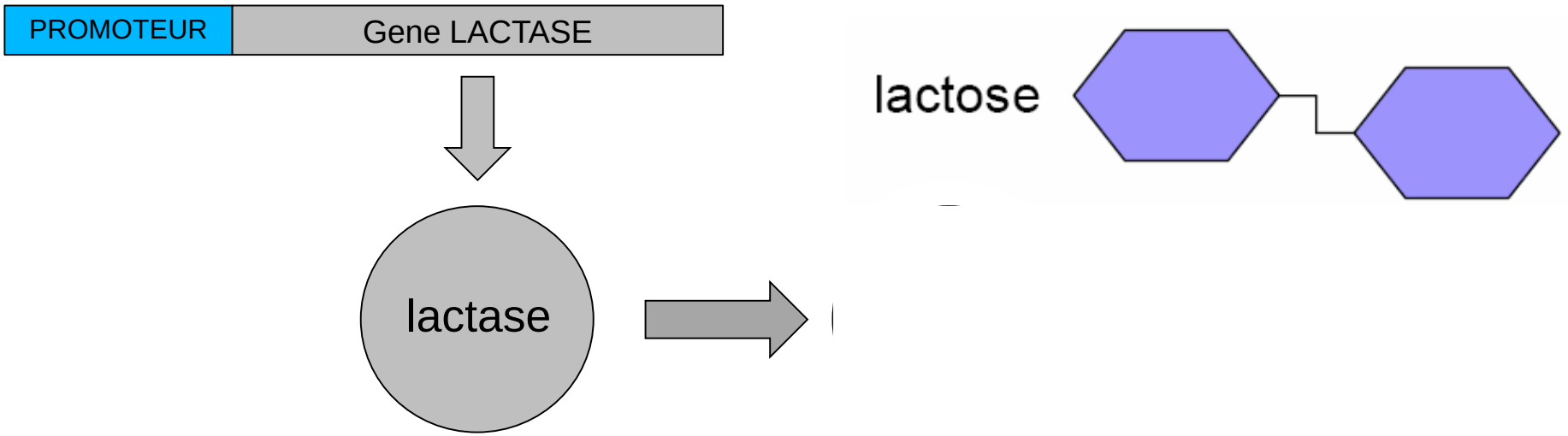
Distribution de la lactose persistance dans le monde et de l'allèle -13910T (associée à la LP)



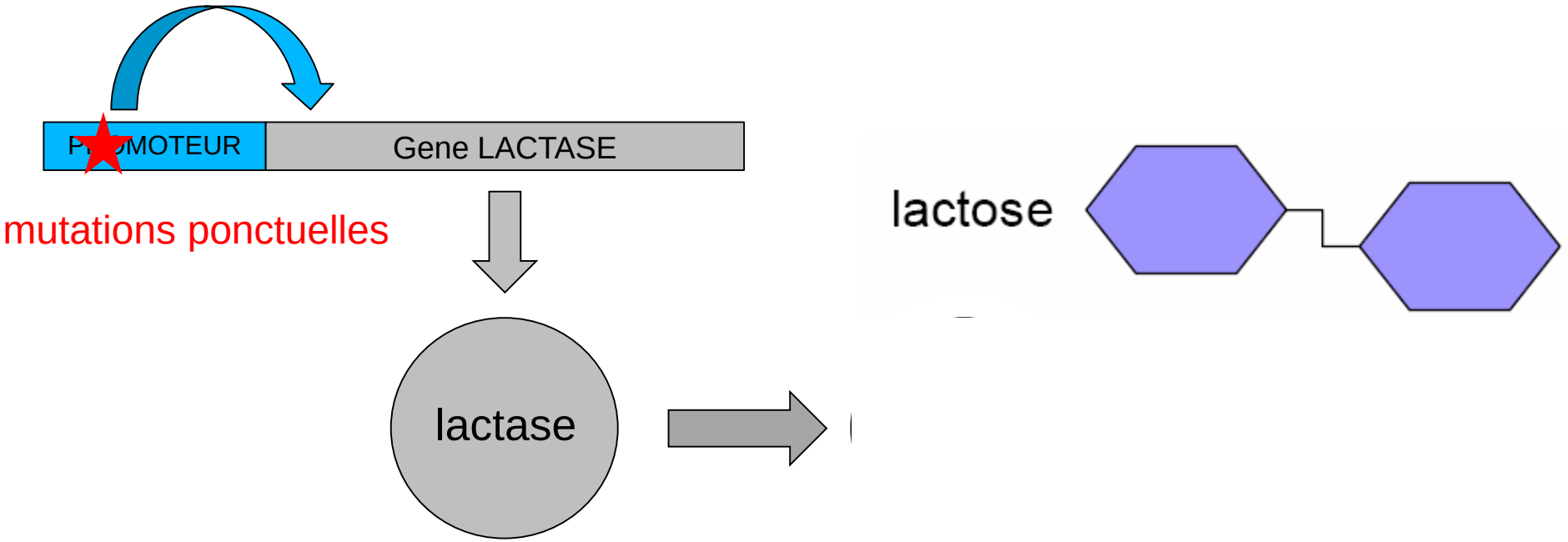
Action de la lactase sur le lactose jusqu'au sevrage



Action de la lactase sur le lactose après sevrage



Dans le cas d'une mutation sur le promoteur après sevrage



Le gène de la lactase sous sélection positive

On sait actuellement que le lait a eu une importance capitale dans la survie des humains, lors de leur arrivée dans certaines régions du monde.

La distribution atypique du trait ainsi que la relation entre cette distribution, les habitudes culturelles des populations et la distribution géographique des élevages laitiers suggèrent que la sélection positive est responsable de la persistance de l'enzyme de lactase.

Sélection positive par pression environnementale

Différentes hypothèses sélectives ont été proposées [Holden and Mace, 1997, Mace et al., 2003].

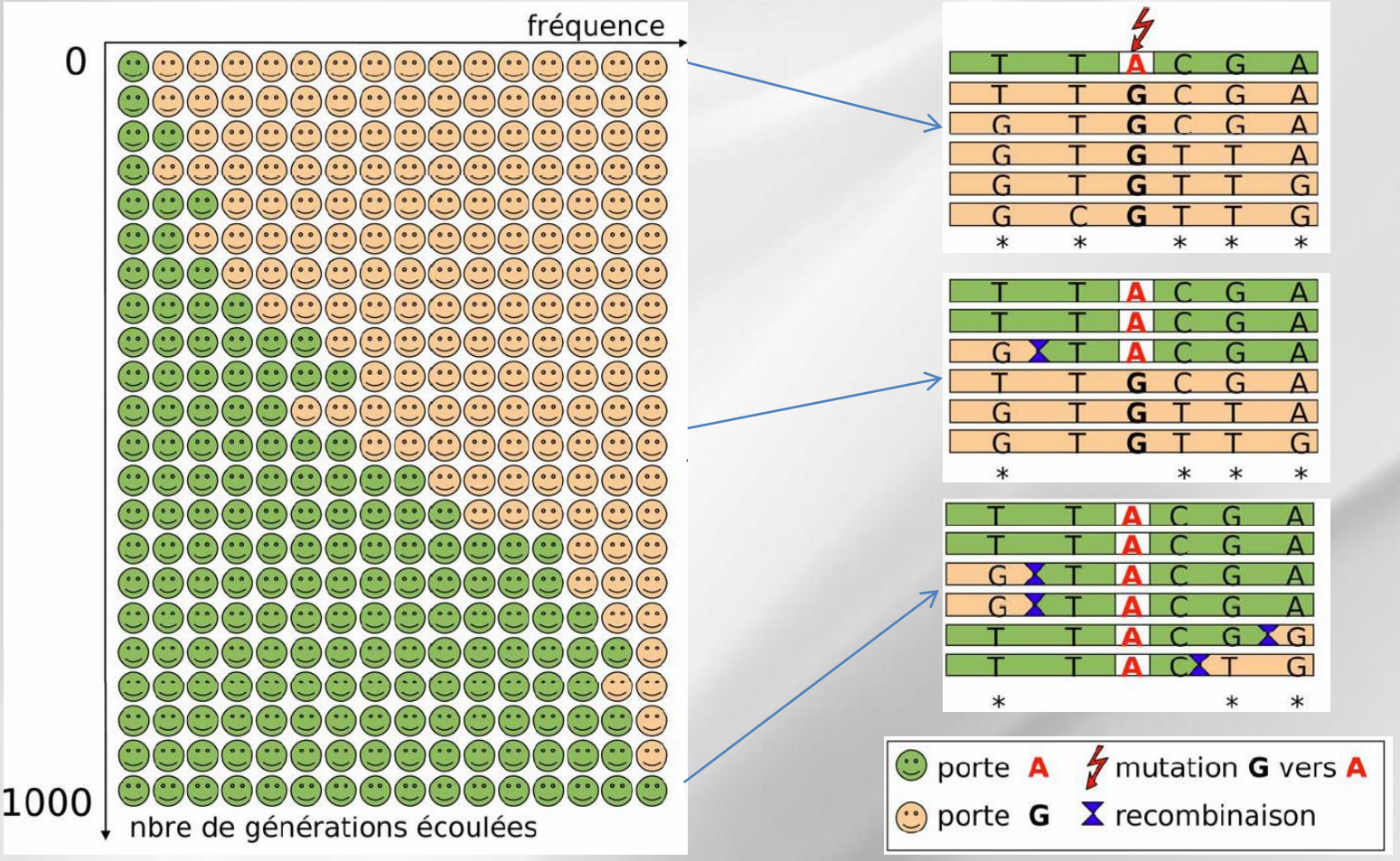
Les rares éleveurs initialement dotés de la persistance ont en consommant le lait de leurs animaux profité d'un apport énergétique qui s'est traduit par une descendance plus abondante.

Les mutations conférant la persistance à la lactase sont ainsi passées d'une fréquence très faible au début de l'élevage il y a 10000 ans à une fréquence très importante dans la population européenne et certaines populations africaines aujourd'hui.

Ces mutations ne sont pas fixées, elles sont en cours d'installation dans les populations

Il est important d'ajouter que la persistance de la lactase en Europe et dans certaines populations africaines est un cas tout à fait remarquable de convergence évolutive: les mutations sélectionnées en Europe et en Afrique sont apparues indépendamment et ne sont pas situées aux mêmes positions dans les séquences régulatrices (Tishkoff et al. 2007).

Qu'est-ce que la sélection positive ?

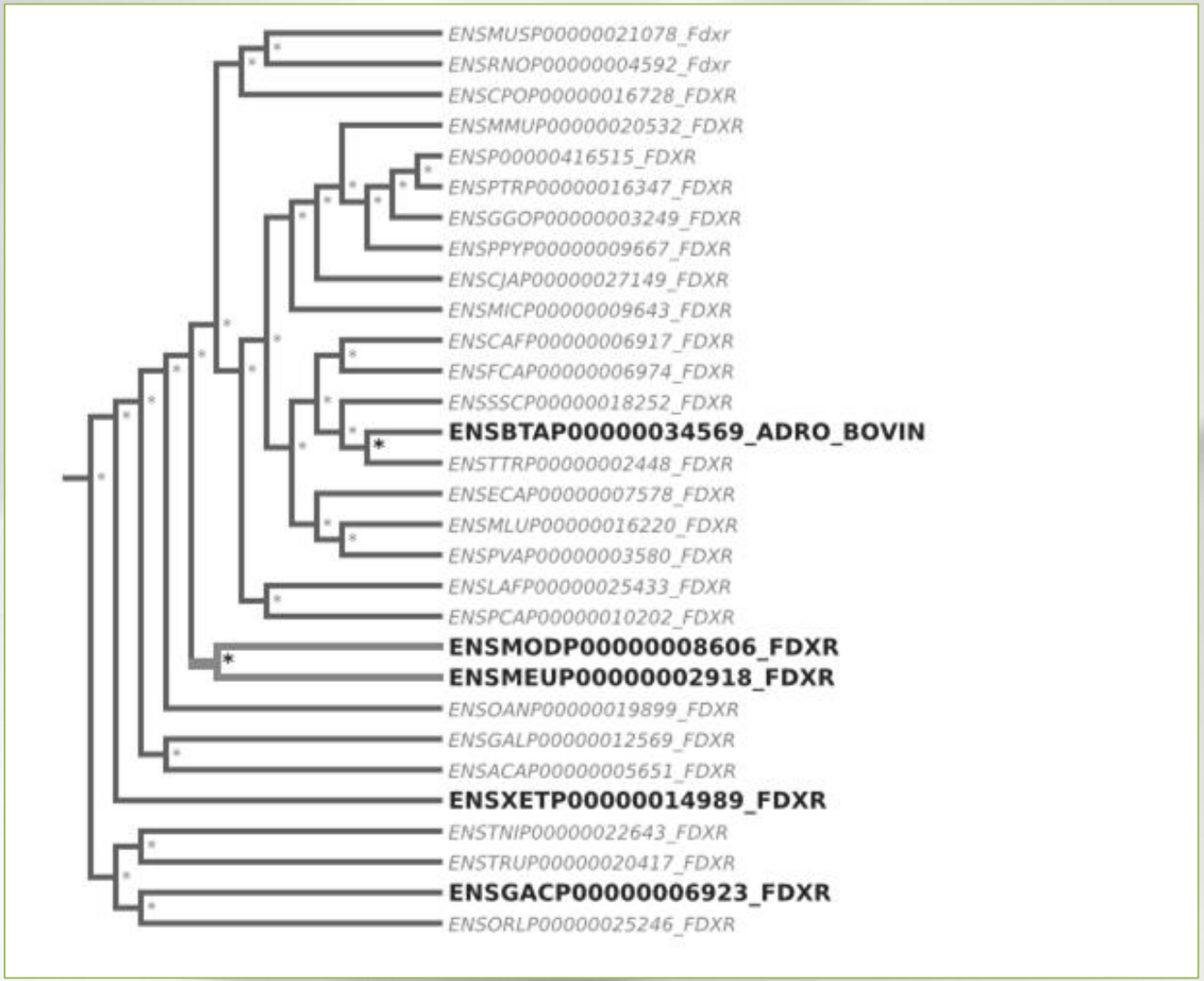


A mesure qu'un trait avantageux est sélectionné, la mutation causative du trait augmente en fréquence et peut le cas échéant remplacer l'état originel.

Sélection positive: principe

Point de départ :
1- un arbre phylogénétique
et
2- l’alignement multiple qui a servi à sa construction

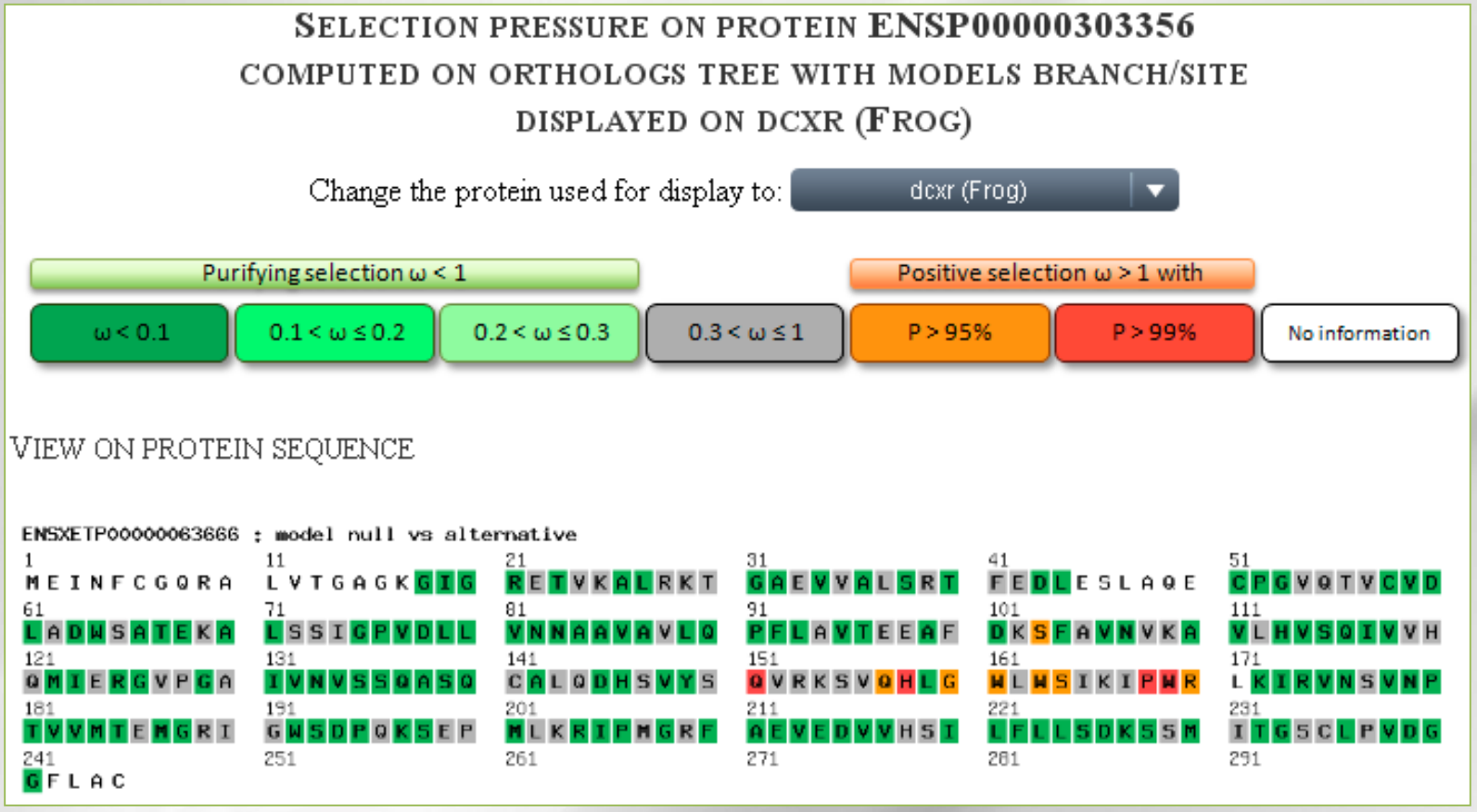
Sélection positive: principe



Sélection positive: principe

Après calculs, repérage des branches de l'arbre
qui sont sous sélection (statistiquement significatif)

Sélection positive: principe



Sélection positive: principe

VIEW ON 3D STRUCTURE

View : without residues positive selection residues purifying selection residues

Back to original view | Turn around X axis

Le calcul de sélection positive au niveau moléculaire

Basé sur l'estimation du ratio ω

$$\omega = \frac{\text{Taux de substitutions non-synonymes}}{\text{Taux de substitutions synonymes}} = \frac{dn}{d_S} = \frac{Ka}{Ks}$$

Substitution **non-synonyme** : **changement** de l'acide aminé (TAC_Tyr → TCC_Ser)

Substitution **synonyme** : **pas de changement** de l'acide aminé (TAC_Tyr→ TAT_Tyr)

Le calcul de sélection positive au niveau moléculaire

$$\omega = \frac{\text{Taux de substitutions non-synonymes } d_N}{\text{Taux de substitutions synonymes } d_S}$$

ω < 1 **ω = 1** **ω > 1**

Sélection purifiante

les changements des acides aminés sont souvent pénalisés par la sélection naturelle.

Evolution neutre

les mutations se répartissent aléatoirement sans pression (pseudogénéisation)

Sélection positive

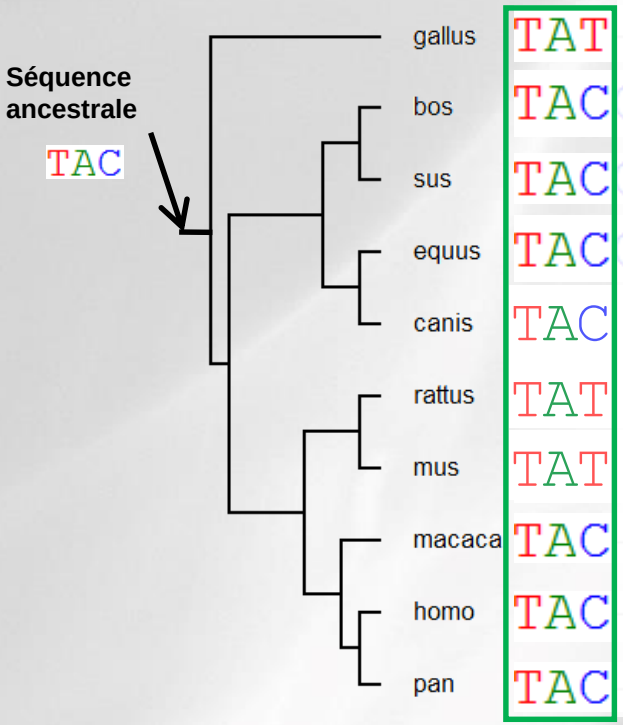
Les modifications des acides aminés sont favorisées par la sélection naturelle qui tend à les conserver au cours de l'évolution plus vite qu'elle ne conserve les mutations synonymes.

Le calcul de sélection positive au niveau moléculaire

- Basé sur un alignement multiple de séquences, alignées par codons, et sur un arbre phylogénétique
- Estimation de ω par maximum de vraisemblance

Calculs sur site

Sélection purifiante



First Position	Second Position of Codon: T	C	A	G	Third Position
T	TTT Phe [F]	TCT Ser [S]	TAT Tyr [Y]	TGT Cys [C]	T
	TTC Phe [F]	TCC Ser [S]	TAC Tyr [Y]	TGC Cys [C]	C
	TTA Leu [L]	TCA Ser [S]	TAA *Ter* [end]	TGA *Ter* [end]	A
	TTG Leu [L]	TCG Ser [S]	TAG *Ter* [end]	TGG Trp [W]	G
C	CTT Leu [L]	CCT Pro [P]	CAT His [H]	CGT Arg [R]	T
	CTC Leu [L]	CCC Pro [P]	CAC His [H]	CGC Arg [R]	C
	CTA Leu [L]	CCA Pro [P]	CAA Gln [Q]	CGA Arg [R]	A
	CTG Leu [L]	CCG Pro [P]	CAG Gln [Q]	CGG Arg [R]	G
A	ATT Ile [I]	ACT Thr [T]	AAT Asn [N]	AGT Ser [S]	T
	ATC Ile [I]	ACC Thr [T]	AAC Asn [N]	AGC Ser [S]	C
	ATA Ile [I]	ACA Thr [T]	AAA Lys [K]	AGA Arg [R]	A
	ATG Met [M]	ACG Thr [T]	AAG Lys [K]	AGG Arg [R]	G
G	GTT Val [V]	GCT Ala [A]	GAT Asp [D]	GGT Gly [G]	T
	GTC Val [V]	GCC Ala [A]	GAC Asp [D]	GGC Gly [G]	C
	GTA Val [V]	GCA Ala [A]	GAA Glu [E]	GGA Gly [G]	A
	GTG Val [V]	GCG Ala [A]	GAG Glu [E]	GGG Gly [G]	G

Le calcul de sélection positive au niveau moléculaire

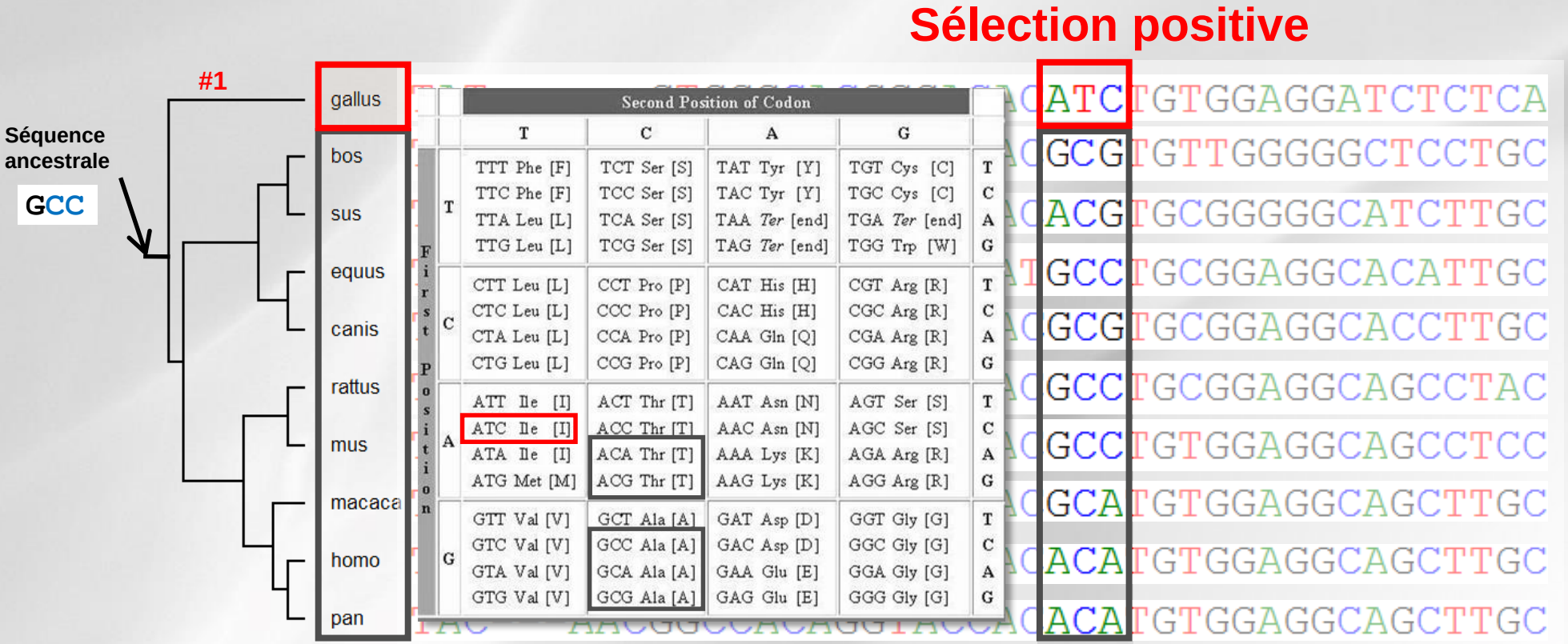
- Basé sur un alignement multiple de séquences, alignées par codons, et sur un arbre phylogénétique
- Estimation de ω par maximum de vraisemblance

Calculs sur site

Le calcul de sélection positive au niveau moléculaire

- Basé sur un alignement multiple de séquences, alignées par codons, et sur un arbre phylogénétique
- Estimation de ω par maximum de vraisemblance

Calculs sur branche-site



Le calcul de sélection positive au niveau moléculaire - codeml

PAML 4: Phylogenetic Analysis by Maximum Likelihood

*Ziheng Yang**

*Department of Biology, Galton Laboratory, University College London, London, United Kingdom

PAML, currently in version 4, is a package of programs for phylogenetic analyses of DNA and protein sequences using maximum likelihood (ML). The programs may be used to compare and test phylogenetic trees, but their main strengths lie in the rich repertoire of evolutionary models implemented, which can be used to estimate parameters in models of sequence evolution and to test interesting biological hypotheses. Uses of the programs include estimation of synonymous and nonsynonymous rates (d_N and d_S) between two protein-coding DNA sequences, inference of positive Darwinian selection through phylogenetic comparison of protein-coding genes, reconstruction of ancestral genes and proteins for molecular restoration studies of extinct life forms, combined analysis of heterogeneous data sets from multiple gene loci, and estimation of species divergence times incorporating uncertainties in fossil calibrations. This note discusses some of the major applications of the package, which includes example data sets to demonstrate their use. The package is written in ANSI C, and runs under Windows, Mac OSX, and UNIX systems. It is available at http://abacus.gene.ucl.ac.uk/software/paml.html.

Introduction

Phylogenetic methods for comparative analysis of DNA and protein sequences are becoming ever more important with the rapid accumulation of molecular sequence data, spearheaded by numerous genome projects. It is now common for phylogeny reconstruction to be conducted using large data sets involving hundreds or even thousands of genes. Similarly, phylogenetic methods are widely used to estimate the evolutionary rates of genes and genomes to detect footprints of natural selection, and the evolutionary information is used to interpret genomic data (Yang 2005). For example, both evolutionary conservation indicating negative purifying selection and accelerated evolution driven by positive Darwinian selection have been employed to detect functionally significant regions of the genome (e.g., Thomas et al. 2003; Nielsen et al. 2005;

- Likelihood ratio tests (LRTs) of hypotheses through comparison of nested statistical models (BASEML, CODEML, CHI2);
- Estimation of synonymous and nonsynonymous substitution rates and detection of positive Darwinian selection in protein-coding DNA sequences (YN00 and CODEML);
- Estimation of empirical amino acid substitution matrices (CODEML);
- Estimation of species divergence times under global and local clock models using likelihood (BASEML and CODEML) and Bayesian (MCMCTREE) methods;
- Reconstruction of ancestral sequences using nucleotide, amino acid, and codon models (BASEML and CODEML);
- Generation of nucleotide, codon, and amino acid sequence alignments by Monte Carlo simulation (EVOLVER).

http://abacus.gene.ucl.ac.uk/software/paml.html

CODEML du package PAML

strength of PAML is in its rich collection of sophisticated substitution models, useful when our focus is on understanding the process of sequence evolution. Examples of analyses that can be performed using the package include

- Comparison and tests of phylogenetic trees (BASEML and CODEML);
- Estimation of parameters in sophisticated substitution models, including models of variable rates among sites and models for combined analysis of multiple genes (BASEML and CODEML);

Key words: codon models, likelihood, PAML, phylogenetic analysis, software.

E-mail: z.yang@ucl.ac.uk.

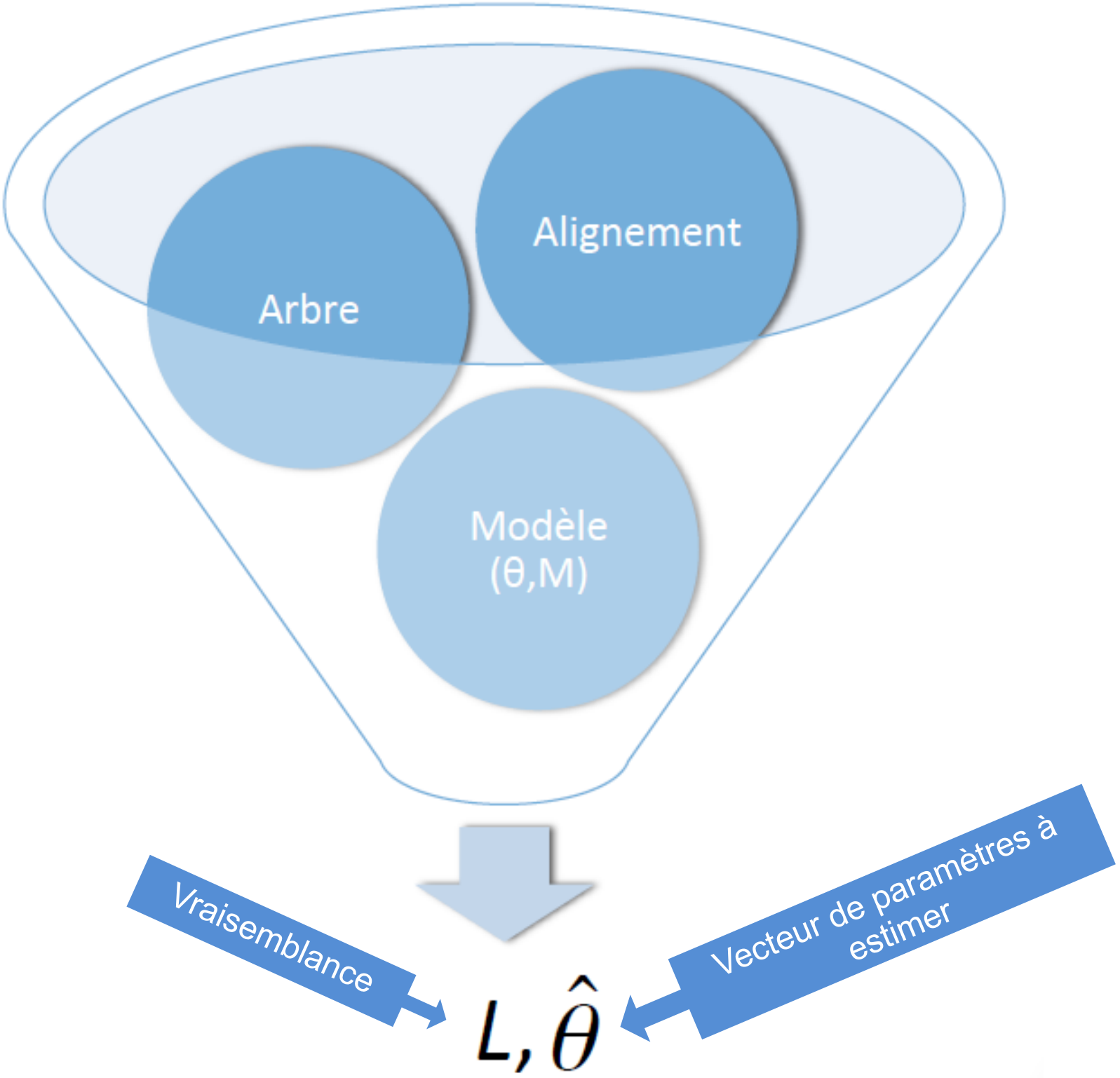
Mol. Biol. Evol. 24(8):1586–1591. 2007
doi:10.1093/molbev/msm088
Advance Access publication May 4, 2007

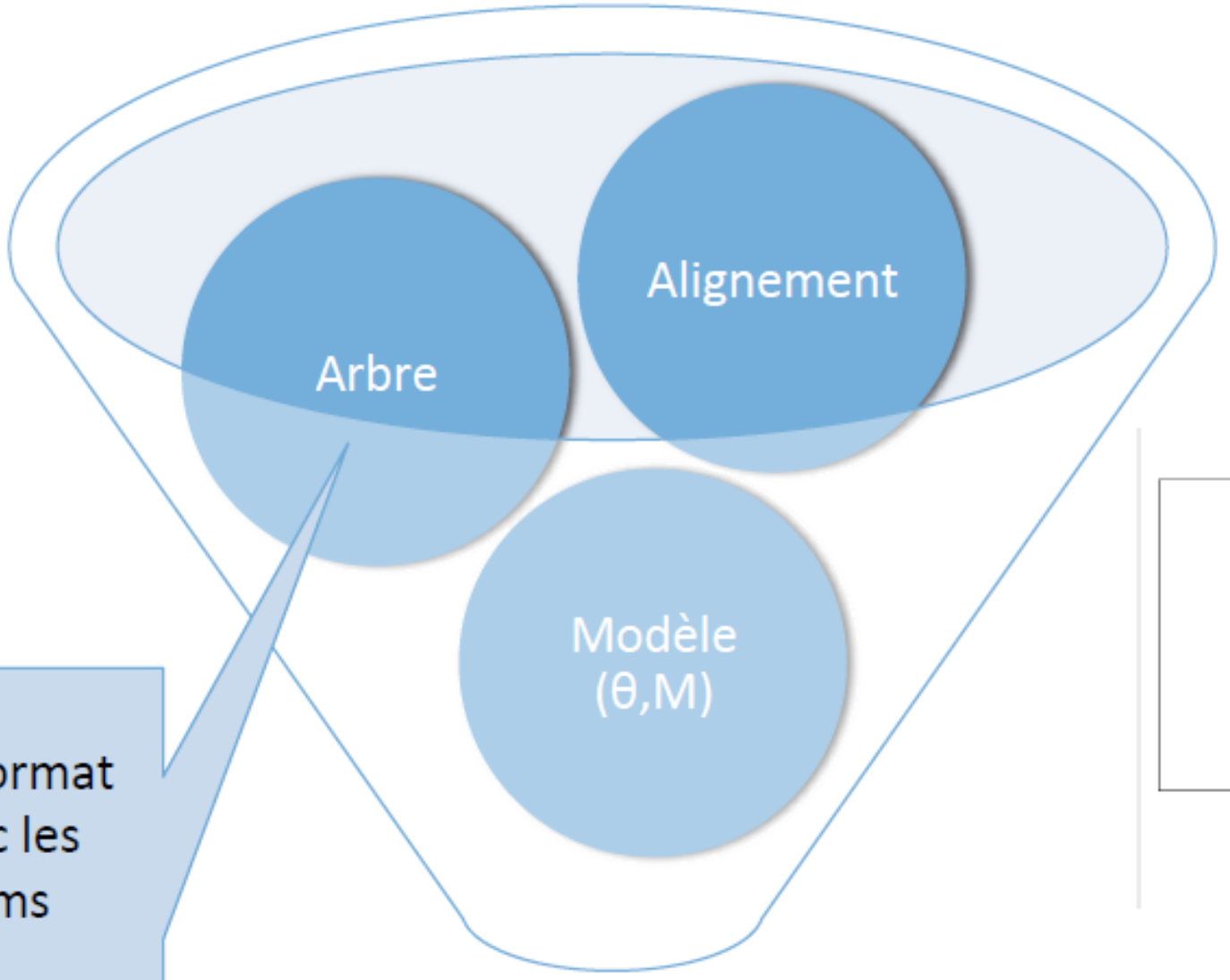
Major Applications of the Software Package

Comparison and Tests of Trees

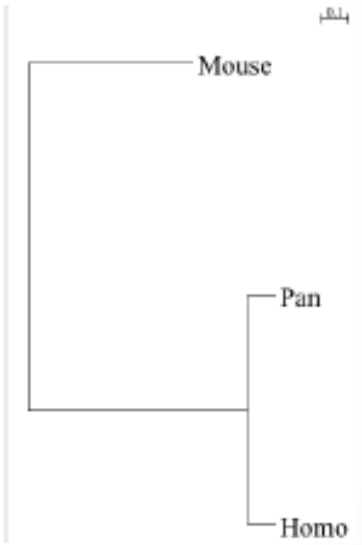
The programs BASEML and CODEML can take a set of user trees and evaluate their log likelihood values under a variety of nucleotide, amino acid, and codon substitution models. When more than one tree is specified, the programs automatically calculates the bootstrap proportions for trees using the RELL method (Kishino and Hasegawa 1989), as well as p values using the K-H test (Kishino and Hasegawa 1989) and S-H test (Shimodaira and Hasegawa 1999). See Goldman, Anderson, and Rodrigo (2000) for a critical review of those methods.

In particular, a number of likelihood models are implemented in BASEML and CODEML for combined analysis of heterogeneous data sets from multiple gene loci (Yang 1996). These models allow estimation of common parameters


Alignement
Arbre
Modèle
(θ,M)
Vraisemblance
Vecteur de paramètres à estimer
$L, \hat{\theta}$

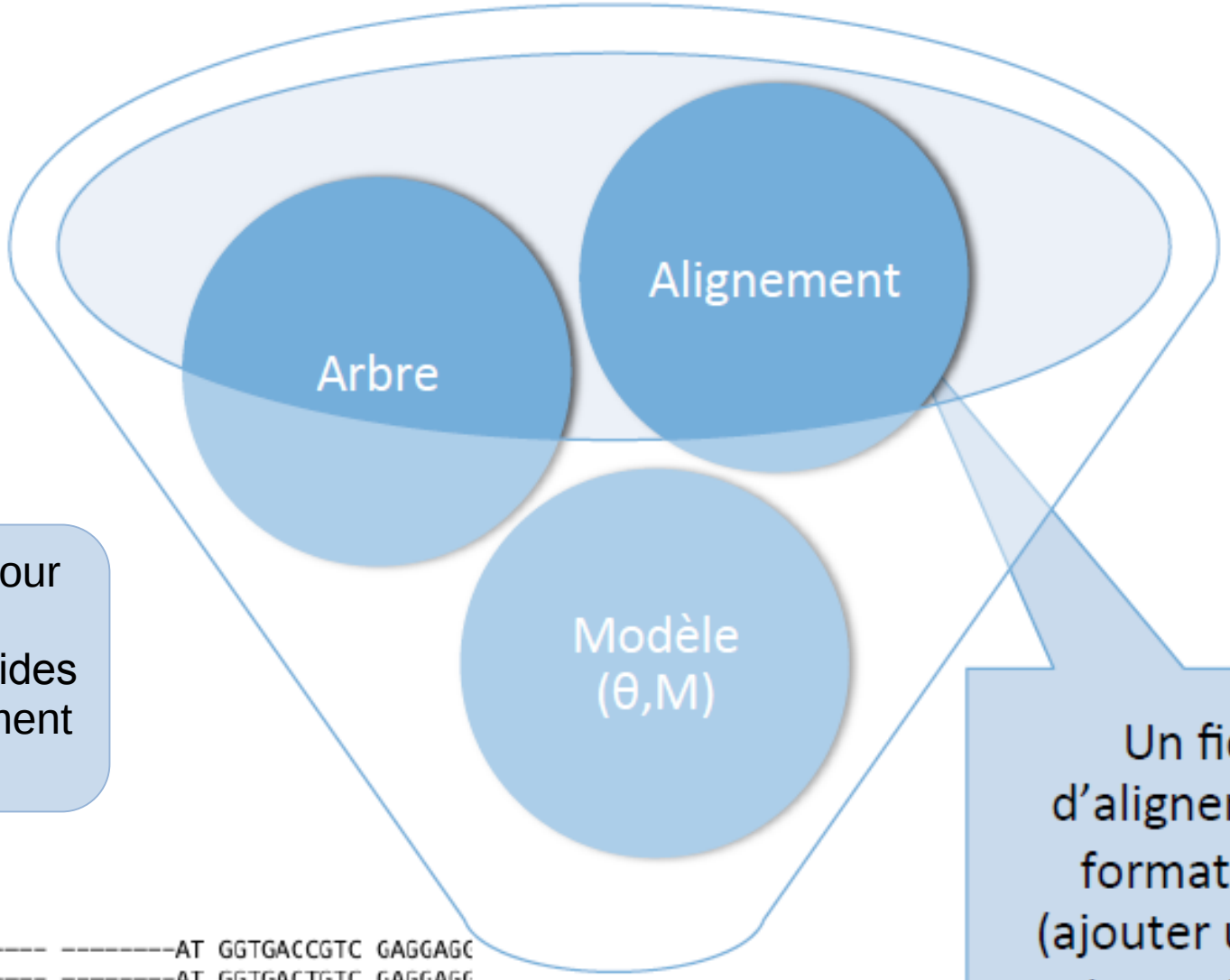


Un arbre au format newick (avec les mêmes noms d'espèces), binaire , sans bootstrap et avec les étiquettes adéquates




```
((Homo:0.1,Pan:0.1):
0.8,Mouse:0.6);
```

$$L, \hat{\theta}$$

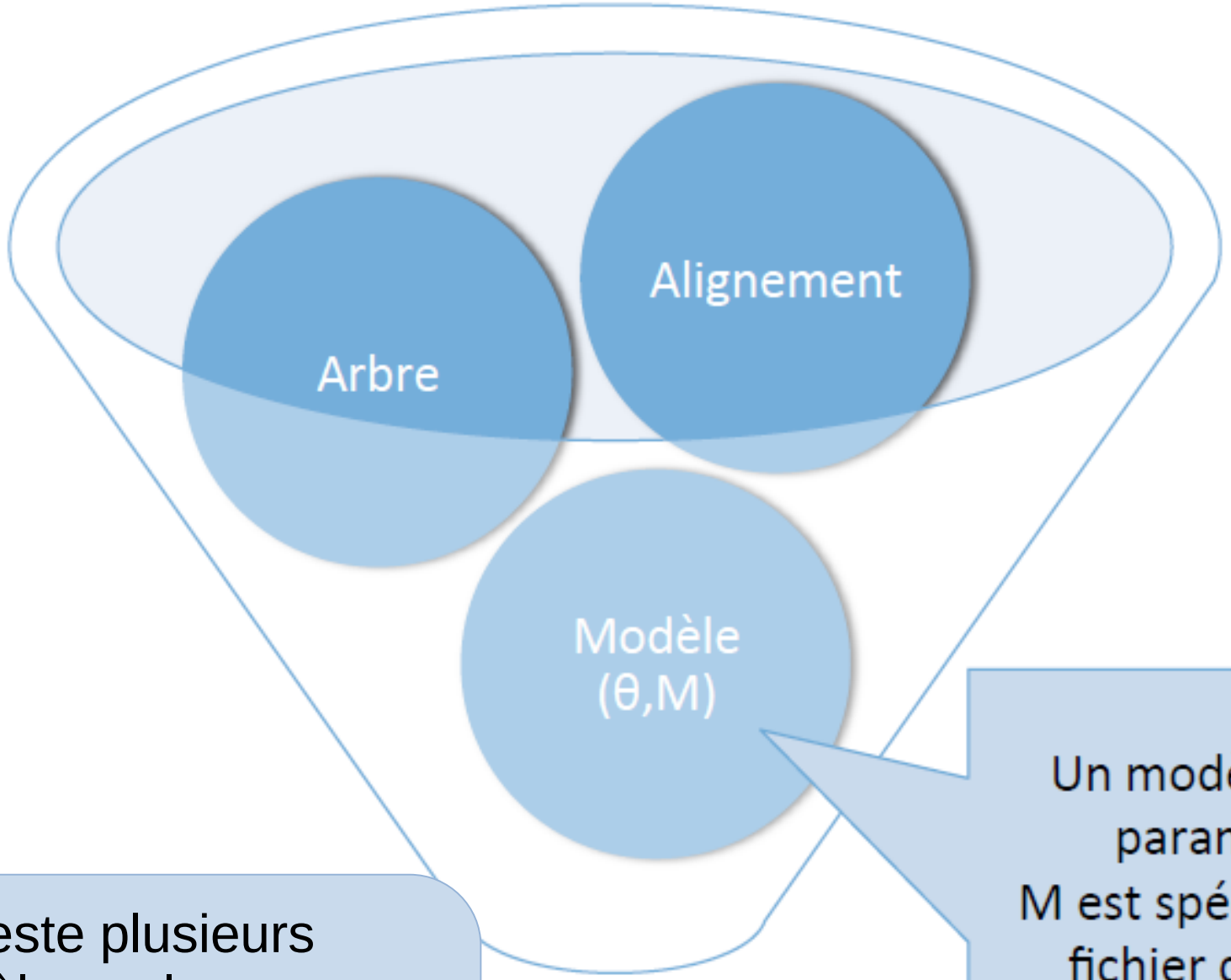


Pal2Nal: logiciel pour le passage de l'alignement en acides aminés en alignement en codons

```
45   1305   I
Ipomoea_n_CHSD      ---------- --------AT GGTGACCGTC GAGGAGC
Ipomoea_n_CHSE      ---------- --------AT GGTGACTGTC GAGGAGC
Ipomoea_pu_CHSD     ---------- --------AT GGTGACCGTC GAGGAGC
Ipomoea_pu_CHDE     ---------- --------AT GGTGACCGTC GAGGAAC
Perilla_f_CHS       ---------- --------AT GGTGACCGTC GAGGACA
Pinus_d_CHS         ---ATGGCTG AGACTTTGGG TTTGGATCTA GAGGCAT
Psilotum_n_CHS      ---------- ---------- ----ATGTCA ATGTCAT
Ipomoea_b_CHSDI     ---------- --------AT GGTGACCGTC GAGGAGC
```

$$L,\ \hat{\theta}$$

Un fichier d'alignement au format phylip (ajouter un I pour indiquer le format « interleaved »), nb sites et gaps multiples de 3, pas de codons stop.



Un modèle M et ses paramètres θ.
M est spécifié grâce au fichier de contrôle (codeml.ctl) :

On teste plusieurs modèle, on les compare statistiquement et on choisit le meilleur, le plus vraisemblable

$L,\ \hat{\theta}$

Fonctionnement de CODEML

En très très gros…

Fixe des valeurs discrètes de ω (0,2 - 1 - 1,5)

Calcule toutes des probabilités de passage d'un codon à un autre.

Calcule la probabilité que les mutations réelles aient eu lieu : vraisemblance.

Fait évoluer les valeurs de ω jusqu'à maximiser la vraisemblance.

Le fichier de contrôle codeml.ctl

```
     seqfile = stewart.aa * sequence data file name
     outfile = mlc * main result file name
    treefile = stewart.trees * tree structure file name

       noisy = 9  * 0,1,2,3,9: how much rubbish on the screen
     verbose = 0  * 1: detailed output, 0: concise output
     runmode = 0  * 0: user tree;  1: semi-automatic;  2: automatic
                  * 3: StepwiseAddition; (4,5):PerturbationNNI; -2: pairwise

     seqtype = 2  * 1:codons; 2:AAs; 3:codons-->AAs
   CodonFreq = 2  * 0:1/61 each, 1:F1X4, 2:F3X4, 3:codon table
*       ndata = 10
       clock = 0   * 0:no clock, 1:clock; 2:local clock; 3:TipDate

      aaDist = 0  * 0:equal, +:geometric; -:linear, 1-6:G1974,Miyata,c,p,v,a
                  * 7:AAClasses
  aaRatefile = wag.dat * only used for aa seqs with model=empirical(_F)
                * dayhoff.dat, jones.dat, wag.dat, mtmam.dat, or your own

       model = 2
                  * models for codons:
                      * 0:one, 1:b, 2:2 or more dN/dS ratios for branches
                  * models for AAs or codon-translated AAs:
                      * 0:poisson, 1:proportional,2:Empirical,3:Empirical+F
                      * 6:FromCodon, 8:REVaa_0, 9:REVaa(nr=189)

     NSsites = 0  * 0:one w;1:neutral;2:selection; 3:discrete;4:freqs;
                  * 5:gamma;6:2gamma;7:beta;8:beta&w;9:beta&gamma;
                  * 10:beta&gamma+1; 11:beta&normal>1; 12:0&2normal>1;
                  * 13:3normal>0

       icode = 0  * 0:universal code; 1:mammalian mt; 2-11:see below
       Mgene = 0  * 0:rates, 1:separate;

   fix_kappa = 0  * 1: kappa fixed, 0: kappa to be estimated
       kappa = 2  * initial or fixed kappa
```

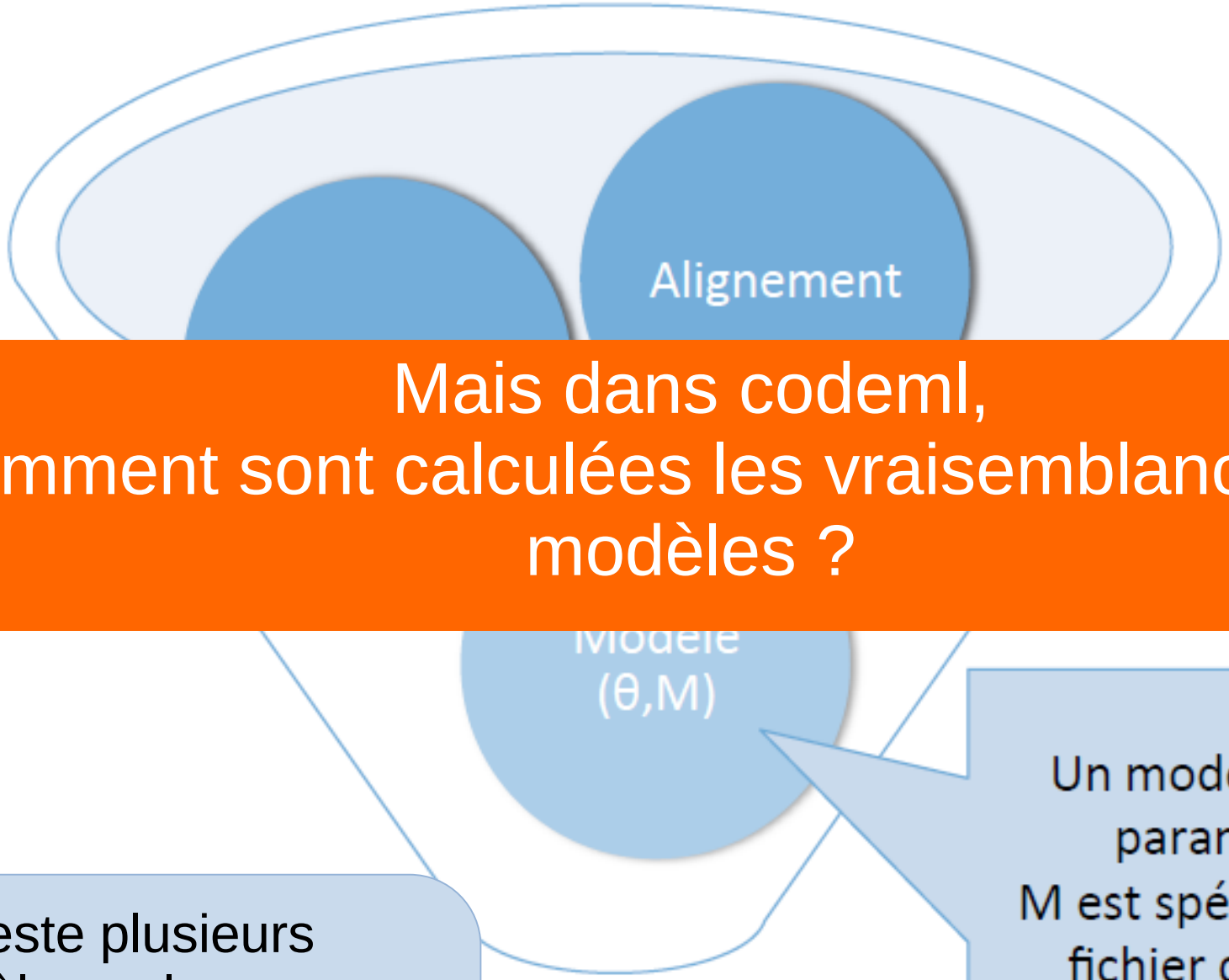
```
    fix_omega = 0   * 1: omega or omega_1 fixed, 0: estimate
        omega = .4 * initial or fixed omega, for codons or codon-based AAs

    fix_alpha = 1   * 0: estimate gamma shape parameter; 1: fix it at alpha
        alpha = 0. * initial or fixed alpha, 0:infinity (constant rate)
       Malpha = 0   * different alphas for genes
        ncatG = 3   * # of categories in dG of NSsites models

      fix_rho = 1   * 0: estimate rho; 1: fix it at rho
          rho = 0. * initial or fixed rho,   0:no correlation

       getSE = 0   * 0: don't want them, 1: want S.E.s of estimates
 RateAncestor = 0   * (0,1,2): rates (alpha>0) or ancestral states (1 or 2)

   Small_Diff = .5e-6
*   cleandata = 0   * remove sites with ambiguity data (1:yes, 0:no)?
* fix_blength = 0    * 0: ignore, -1: random, 1: initial, 2: fixed
       method = 0    * 0: simultaneous; 1: one branch at a time
```



Mais dans codeml,
comment sont calculées les vraisemblances des modèles ?

Un modèle M et ses paramètres θ.
M est spécifié grâce au fichier de contrôle (codeml.ctl) :

On teste plusieurs modèle, on les compare statistiquement et on choisit le meilleur, le plus vraisemblable

$$L,\ \hat{\theta}$$

Mais dans codeml, comment sont calculées les vraisemblances des modèles ?

- Les modèles sont des modèles de codons : matrice Q de taille 61x61 qui donne les taux de substitution d'un codon *i* vers un codon *j* :

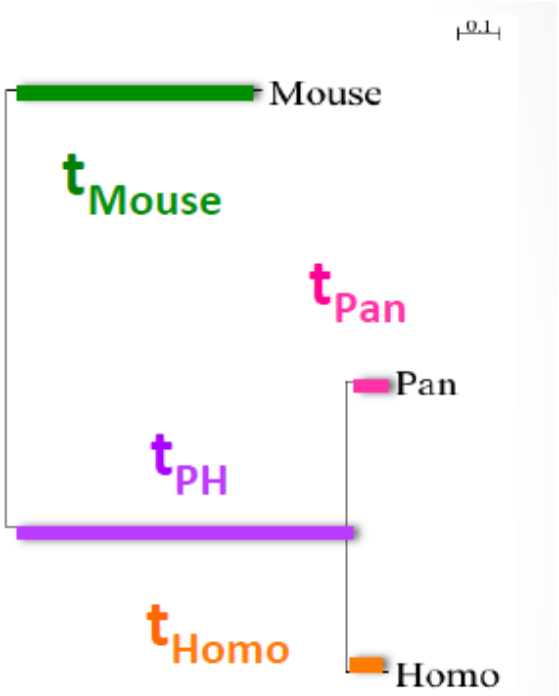
$$Q = \{q_{i \to j}\} \begin{cases} 0 & \text{Si } i \text{ et } j \text{ diffèrent par 2 ou 3 positions} \\ \pi_j & \text{Si } i \text{ et } j \text{ diffèrent par 1 transversion synonyme} \\ \kappa\pi_j & \text{Si } i \text{ et } j \text{ diffèrent par 1 transition synonyme} \\ \omega\pi_j & \text{Si } i \text{ et } j \text{ diffèrent par 1 transversion non-synonyme} \\ \kappa\omega\pi_j & \text{Si } i \text{ et } j \text{ diffèrent par 1 transition non-synonyme} \end{cases}$$

Mais dans codeml, comment sont calculées les vraisemblances des modèles ?

- On transforme Q en P(t) : matrice des probabilités de changement sur un temps t. $P(t) = e^{Qt}$
- Sur une branche de longueur t on a donc la probabilité de substitution d'un codon i vers un codon j : $p_{ij}(t)$
- D'abord le programme propose un set de paramètres $\theta=(\omega_i,t,p_i,p,q,\varkappa,\pi_i)$ ⇔ beaucoup de paramètres !

Mais dans codeml, comment sont calculées les vraisemblances des modèles ?

- Puis, pour un site k (sur les n sites de l'alignement), la vraisemblance conditionnelle à un nœud est le produit de toutes les probabilités de transitions (sur les branches qui découlent de ce nœud) intégré pour tous les états ancestraux possibles (61 codons).



	site k
Mouse	ATT
Pan	ACT
Homo	AGT

$$L_k = \sum_{i}^{61 codons} \pi_i p_{i \to ATT}(t_{Mouse}) \times \left[\sum_{j}^{61 codons} p_{i \to j}(t_{PH}) p_{j \to AGT}(t_{Homo}) p_{j \to ACT}(t_{Pan}) \right]$$

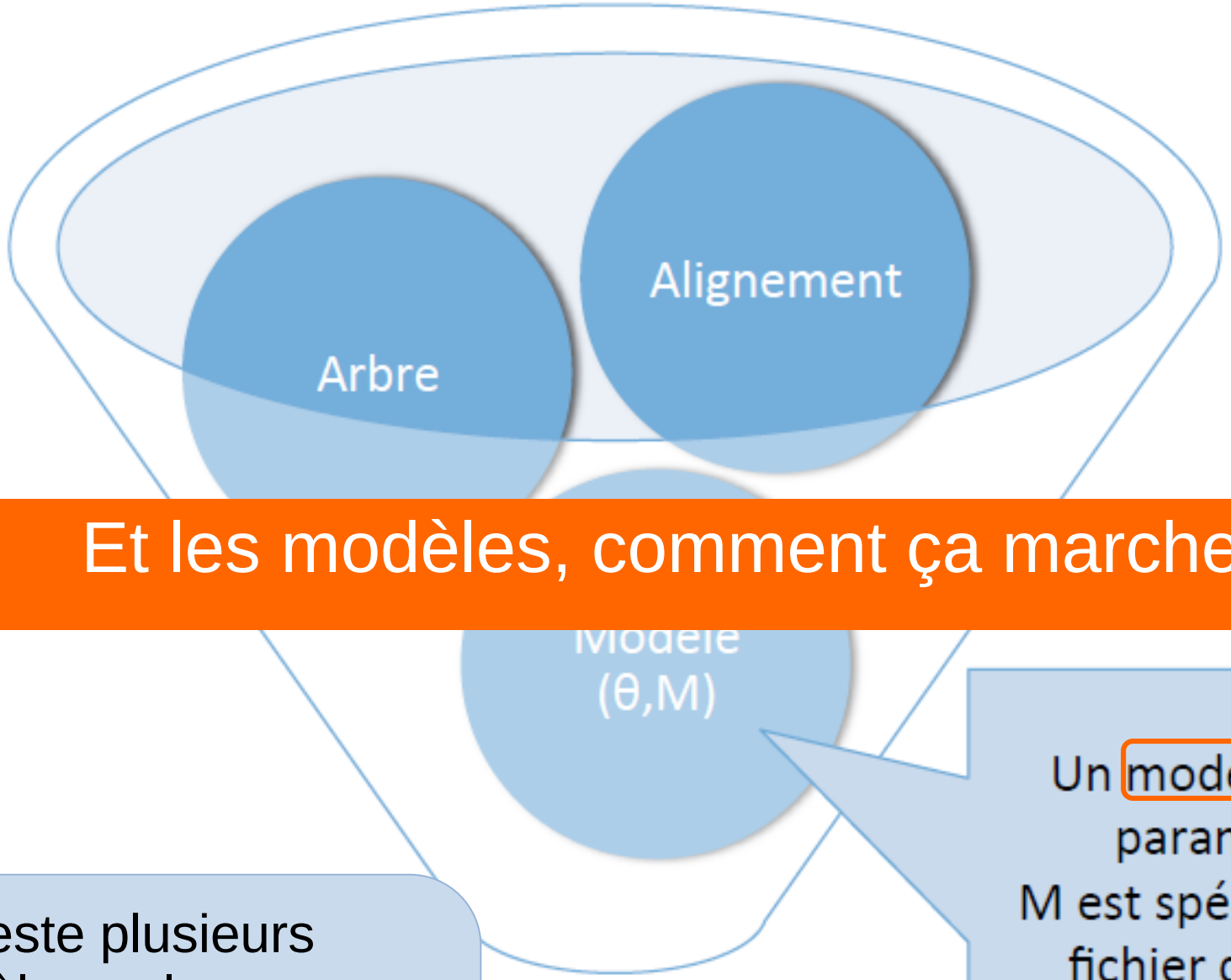
Mais dans codeml, comment sont calculées les vraisemblances des modèles ?

- Puis la vraisemblance globale du modèle sous ce set de paramètres est calculée (*n* sites indépendants)

$$L = \prod_{k=1}^{n} L_k \qquad \ln(L) = \sum_{k=1}^{n} \ln(L_k)$$

- Puis le programme essaye un nouveau set de paramètres, calcule la vraisemblance et ainsi de suite jusqu'à ce que la vraisemblance atteigne un plateau => maximum de vraisemblance.

- Il s'agit d'une méthode exploratoire => elle n'est pas certaine à 100% car il peut exister des maxima locaux de vraisemblance.



Et les modèles, comment ça marche ?

Un modèle M et ses paramètres θ.
M est spécifié grâce au fichier de contrôle (codeml.ctl) :

On teste plusieurs modèle, on les compare statistiquement et on choisit le meilleur, le plus vraisemblable

$$L,\ \hat{\theta}$$

```
      seqfile = stewart.aa * sequence data file name
      outfile = mlc * main result file name
     treefile = stewart.trees * tree structure file name

        noisy = 9  * 0,1,2,3,9: how much rubbish on the screen
      verbose = 0  * 1: detailed output, 0: concise output
      runmode = 0  * 0: user tree;  1: semi-automatic;  2: automatic
                   * 3: StepwiseAddition; (4,5):PerturbationNNI; -2: pairwise

      seqtype = 2  * 1:codons; 2:AAs; 3:codons-->AAs
    CodonFreq = 2  * 0:1/61 each, 1:F1X4, 2:F3X4, 3:codon table
*       ndata = 10
        clock = 0   * 0:no clock, 1:clock; 2:local clock; 3:TipDate

       aaDist = 0  * 0:equal, +:geometric; -:linear, 1-6:G1974,Miyata,c,p,v,a
                   * 7:AAClasses
   aaRatefile = wag.dat * only used for aa seqs with model=empirical(_F)
                  * dayhoff.dat, jones.dat, wag.dat, mtmam.dat, or your own

        model = 2
                   * models for codons:
                       * 0:one, 1:b, 2:2 or more dN/dS ratios for branches
                   * models for AAs or codon-translated AAs:
                       * 0:poisson, 1:proportional,2:Empirical,3:Empirical+F
                       * 6:FromCodon, 8:REVaa_0, 9:REVaa(nr=189)

      NSsites = 0  * 0:one w;1:neutral;2:selection; 3:discrete;4:freqs;
                   * 5:gamma;6:2gamma;7:beta;8:beta&w;9:beta&gamma;
                   * 10:beta&gamma+1; 11:beta&normal>1; 12:0&2normal>1;
                   * 13:3normal>0

        icode = 0  * 0:universal code; 1:mammalian mt; 2-11:see below
        Mgene = 0  * 0:rates, 1:separate;

     fix_kappa = 0  * 1: kappa fixed, 0: kappa to be estimated
        kappa = 2  * initial or fixed kappa
```

Codeml de PAML: les modèle sous lesquels peut évoluer ω

Les différents modèles évolutifs sont paramétrables sous la variables NSsites du fichier CTL

#p = nb de paramètres à estimer

Model	NSsites	#p	Parameters
M0 (one ratio)	0	1	ω
M1a (neutral)	1	2	p_0 ($p_1 = 1 - p_0$), $\omega_0 < 1$, $\omega_1 = 1$
M2a (selection)	2	4	p_0, p_1 ($p_2 = 1 - p_0 - p_1$), $\omega_0 < 1$, $\omega_1 = 1$, $\omega_2 > 1$
M3 (discrete)	3	5	p_0, p_1 ($p_2 = 1 - p_0 - p_1$) ω_0, ω_1, ω_2
M7 (beta)	7	2	p, q
M8 (beta&ω)	8	4	p_0 ($p_1 = 1 - p_0$), $p, q, \omega_s > 1$

p_1 = proportion des sites sous ω discrétisé

Modèle	p	Paramètres	Notes
M0 (one-ratio)	1	ω	One ω ratio for all sites
M1 (neutral)	1	p_0	$p_1 = 1 - p_0$, $\omega_0 = 0$, $\omega_1 = 1$
M2 (selection)	3	p_0, p_1, ω_2	$p_2 = 1 - p_0 - p_1$, $\omega_0 = 0$, $\omega_1 = 1$
M3 (discrete)	$2K - 1$ ($K = 3$)	p_0, p_L, … , p_{K-2}, ω_0, ω_1, . . ω_{K-1}	$p_{K-1} = 1 - p_0 - p_1 - \ldots - p_{K-2}$
M4 (freqs)	$K-1$ ($K = 5$)	p_0, p_L, … , pK_{-2}	The ω_K are fixed at 0, ⅓>, ⅔, 1, and
M5 (gamma)	2	α, β	From G(α,β)
M6 (2gamma)			
M7 (beta)			
M8 (beta&ω)			
M9 (beta&gamma)			
M10 (beta&gamma + 1)	5	p_0, P, Q, α, β	p_0 from B(P, Q) and $1 - p_0$ from 1 +
M11 (beta&normal>1)	5	p_0, P, Q, μ, σ	p_0 from B(P, Q) and $1 - p_0$ from N(M … runcated to $\omega > 1$
M12 (0&2normal>1)	5	p_0, p_1, μ_2, σ_1, σ_2	p0 with $\omega_0 = 0$ and $1 - p_0$ from the … e: p_1 from $N(1, \sigma_1^2)$, and $1 - p_1$ from $N(\mu_2, \sigma_2^2)$ … normals truncated to $\omega > 1$
M13 (3normal>0)	6	p_0, p_1, μ_2, σ_0, σ, σ_2	p_0 from $N(0, \sigma_0^2)$, p_1 from $N(1, \sigma_1^2)$ … $= 1 - p_0 - p_1$ from $N(\mu_2, \sigma_2^2)$, all normals truncated to $\omega > 1$

Modèles de complexes

Entre parenthèses = noms dans le fichier codeml.ctl :

```
NSsites = 0  * 0:one w;1:neutral;2:selection; 3:discrete;4:freqs;
             * 5:gamma;6:2gamma;7:beta;8:beta&w;9:beta&gamma;
             * 10:beta&gamma+1; 11:beta&normal>1; 12:0&2normal>1;
             * 13:3normal>0
```

Exemple modèles 7 et 8

ATG AAT AAC GAT TGC AAA GTC CCT GTC AAT

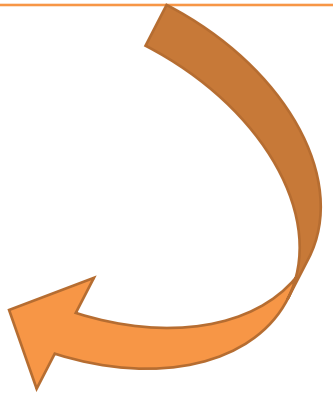
$\omega_1, \omega_2, \omega_3, \omega_4, \omega_5, \omega_6, \omega_7, \omega_8, \omega_9, \omega_{10}$

Le modèle M7 fait varier ω selon une loi béta discrétisé K fois

Modèle M7 : ω_i -> β(p,q)

$i \in \{1,2,3,\ldots,K\}$

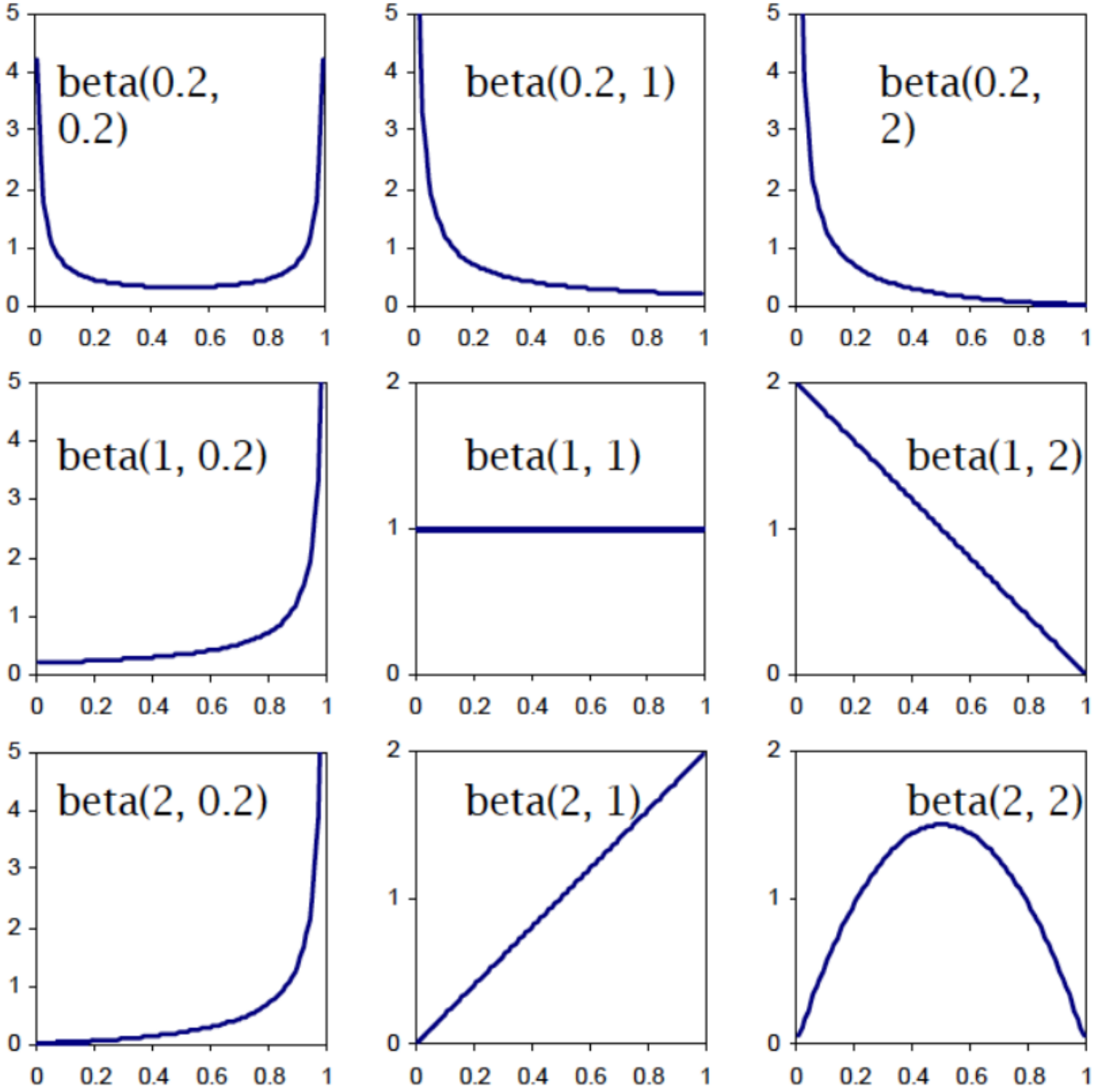
K=10 par defaut, peut être changé grâce à la variable **nCatG**



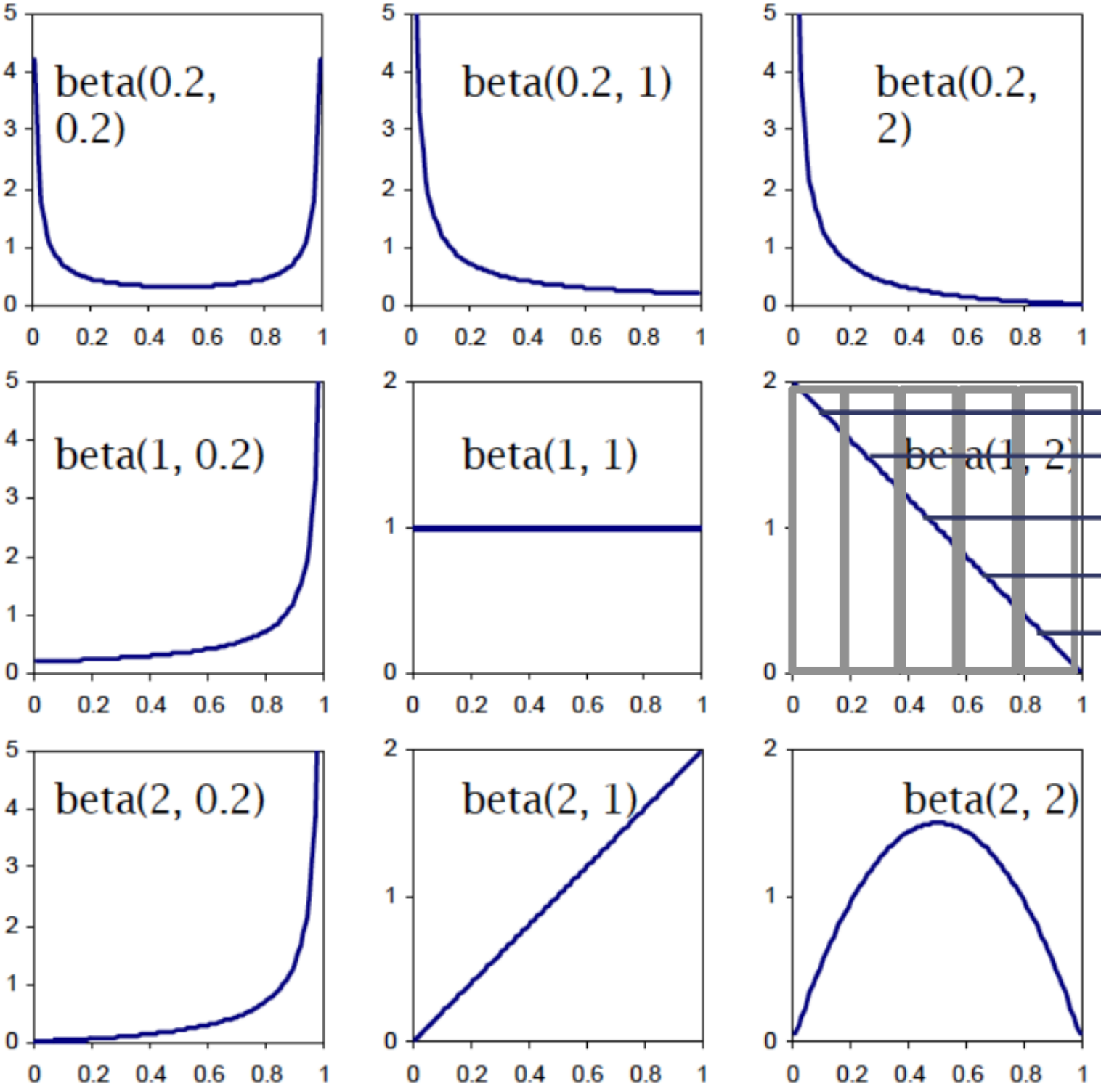
Sélection négative ω<1

Evolution neutre ω=1

Sélection positive ω>1



La loi beta est très flexible !



La loi beta est très flexible !

Exemple de discrétisation **en K=5 catégories**

Exemple modèles 7 et 8

ATG AAT AAC GAT TGC AAA GTC CCT GTC AAT

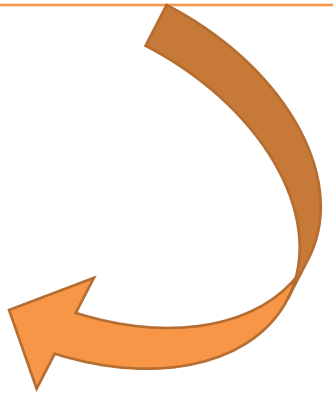
$\omega_1, \omega_2, \omega_3, \omega_4, \omega_5, \omega_6, \omega_7, \omega_8, \omega_9, \omega_{10}$

Le modèle M7 fait varier ω selon une loi béta discrétisé K fois

Modèle M7 : ω_i -> β(p,q)

$i \in \{1,2,3,\ldots,K\}$

K=10 par defaut, peut être changé grâce à la variable **nCatG**



Sélection négative ω<1

Evolution neutre ω=1

Sélection positive ω>1

Exemple modèles 7 et 8

ATG AAT AAC GAT TGC AAA GTC CCT GTC AAT

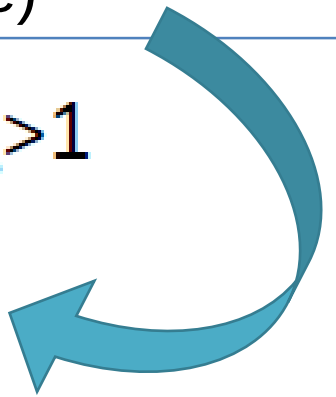
$\omega_1, \omega_2, \omega_3, \omega_4, \omega_5, \omega_6, \omega_7, \omega_8, \omega_9, \omega_{10}, \omega_S$

Le modèle M8 fait varier ω selon une loi béta discrétisé K fois
+ un ω > 1 (autorise la sélection positive)

Modèle M8 : ω_i -> β(p,q) et ω_S>1

$i \in \{1,2,3,\ldots,K\}$

K=10 par defaut, peut être changé grâce à la variable **nCatG**



Sélection négative ω<1

Evolution neutre ω=1

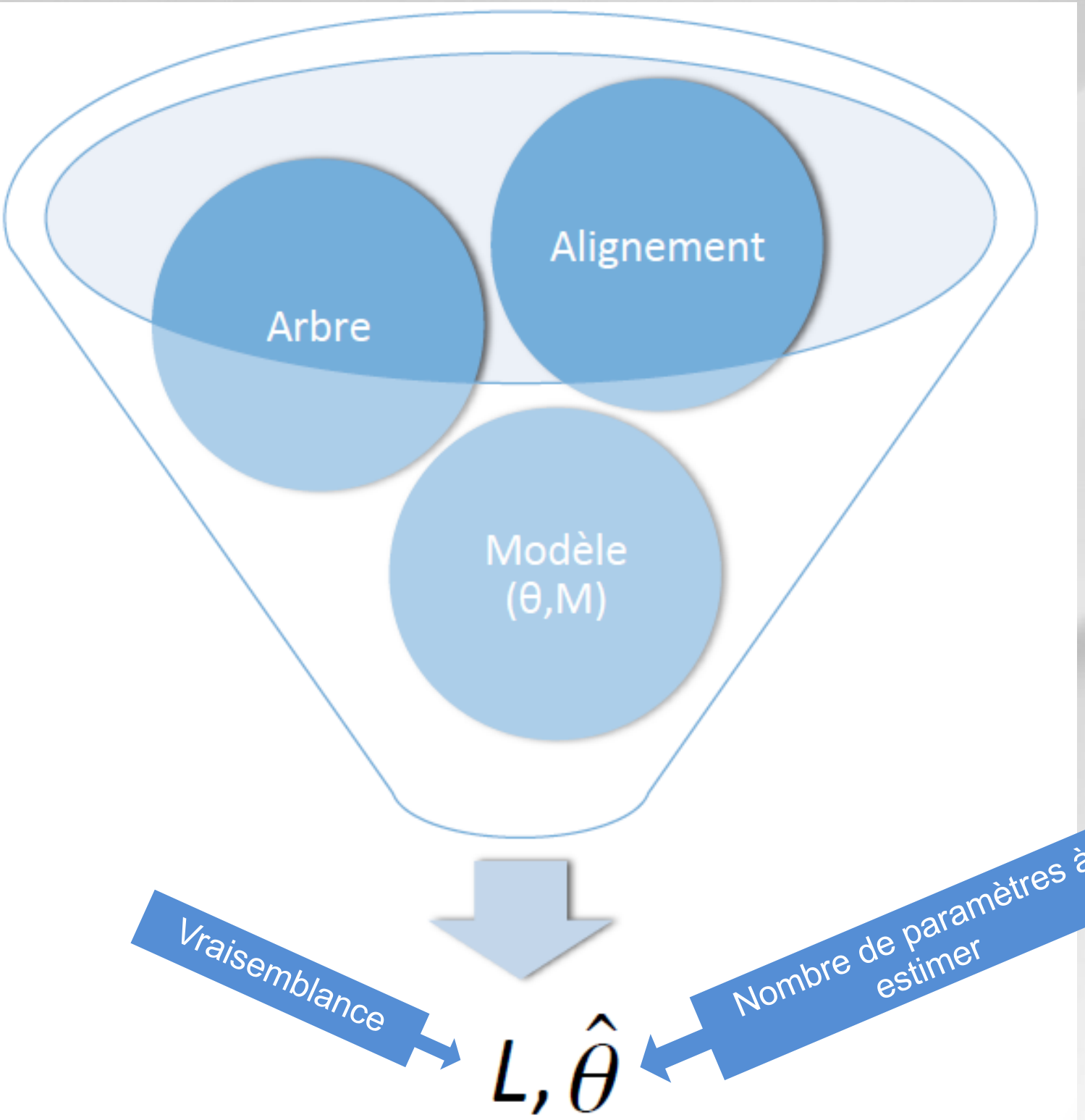
Sélection positive ω>1

Exo 1

ML estimation of the dN/dS (ω) ratio “by hand” for GstD1

Exo 2

Investigating the sensitivity of the dN/dS ratio to assumptions


Arbre
Alignement
Modèle
(θ,M)
Vraisemblance
Nombre de paramètres à estimer
$L, \hat{\theta}$

3 différents type de calculs

Calcul sur site

Calcul sur Branch

Calcul sur Branch-Site

Colonisation d'une nouvelle niche écologique, divergence fonctionnelle de gènes dupliqués, colonisation d'un hôte par un parasite… devient une point sur le fil du temps particulier dans l'histoire évolutive

Se choisit au travers du paramètre **model** dans le fichier de contrôle .ctl

```
      seqfile = stewart.aa * sequence data file name
      outfile = mlc * main result file name
     treefile = stewart.trees * tree structure file name

        noisy = 9  * 0,1,2,3,9: how much rubbish on the screen
      verbose = 0  * 1: detailed output, 0: concise output
      runmode = 0  * 0: user tree;  1: semi-automatic;  2: automatic
                   * 3: StepwiseAddition; (4,5):PerturbationNNI; -2: pairwise

      seqtype = 2  * 1:codons; 2:AAs; 3:codons-->AAs
    CodonFreq = 2  * 0:1/61 each, 1:F1X4, 2:F3X4, 3:codon table
*       ndata = 10
        clock = 0   * 0:no clock, 1:clock; 2:local clock; 3:TipDate

       aaDist = 0  * 0:equal, +:geometric; -:linear, 1-6:G1974,Miyata,c,p,v,a
                   * 7:AAClasses
   aaRatefile = wag.dat * only used for aa seqs with model=empirical(_F)
                  * dayhoff.dat, jones.dat, wag.dat, mtmam.dat, or your own

        model = 2
                   * models for codons:
                       * 0:one, 1:b, 2:2 or more dN/dS ratios for branches
                   * models for AAs or codon-translated AAs:
                       * 0:poisson, 1:proportional,2:Empirical,3:Empirical+F
                       * 6:FromCodon, 8:REVaa_0, 9:REVaa(nr=189)

      NSsites = 0  * 0:one w;1:neutral;2:selection; 3:discrete;4:freqs;
                   * 5:gamma;6:2gamma;7:beta;8:beta&w;9:beta&gamma;
                   * 10:beta&gamma+1; 11:beta&normal>1; 12:0&2normal>1;
                   * 13:3normal>0

        icode = 0  * 0:universal code; 1:mammalian mt; 2-11:see below
        Mgene = 0  * 0:rates, 1:separate;

    fix_kappa = 0  * 1: kappa fixed, 0: kappa to be estimated
        kappa = 2  * initial or fixed kappa
```

3 différents type de calculs

Sur site: model = 0 means one ω ratio for all lineages (branches),

Sur branch: model = 1 means one ratio for each branch (the free-ratio model), and

Sur branch-site: model = 2 means an arbitrary number of ratios (such as the 2-ratios or 3-ratios models).
When model = 2, you have to group branches on the tree into branch groups using the symbols # or $ in the tree.

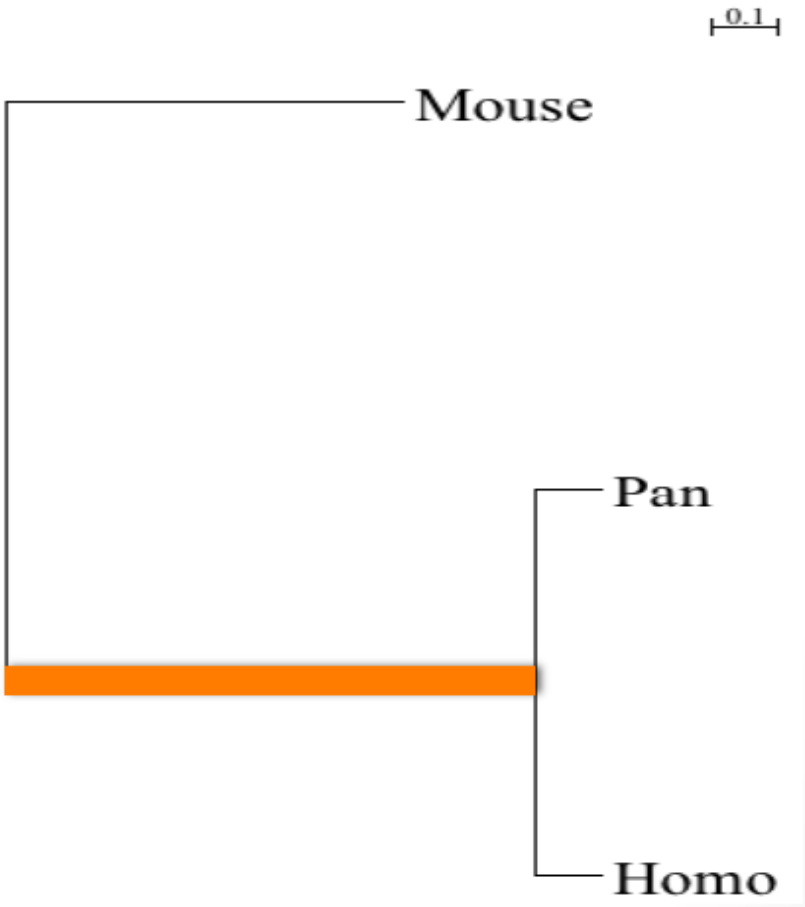
Calcul branch-site

```
((Homo:0.1,Pan:0.1)#1:0.8,Mouse:0.6);
```

ω_0

ω_1

Pose une hypothèse d'évolution sur cette branche



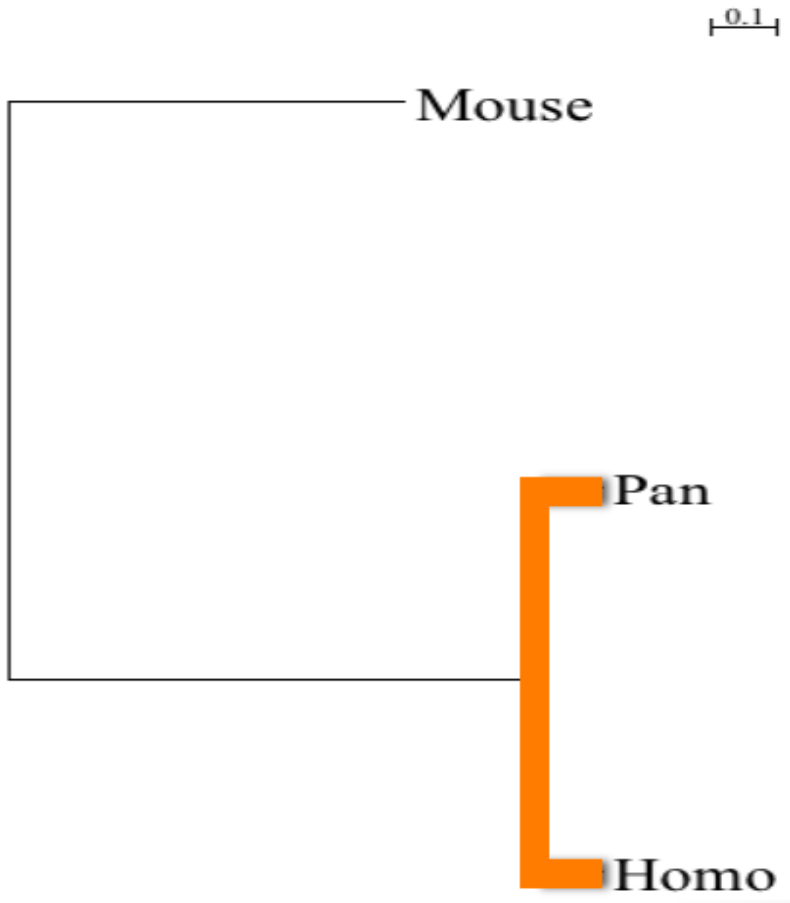
Calcul branch-site

```
((Homo:0.1,Pan:0.1)$1:0.8,Mouse:0.6);
```

ω_0

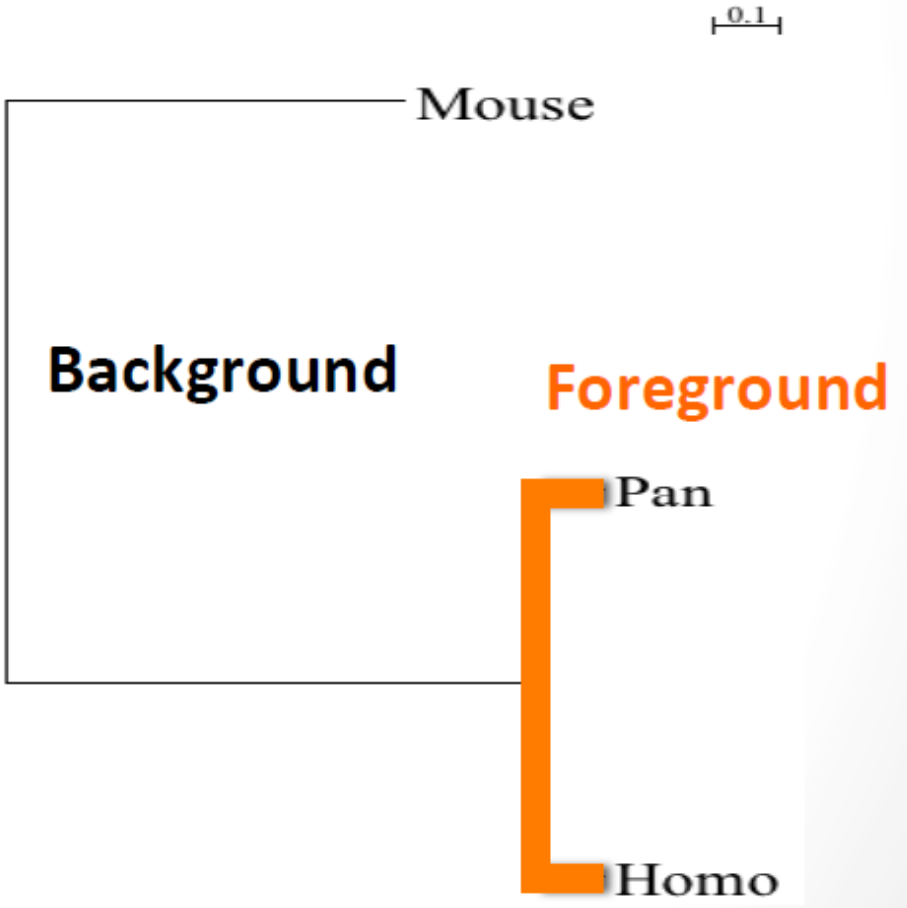
ω_1

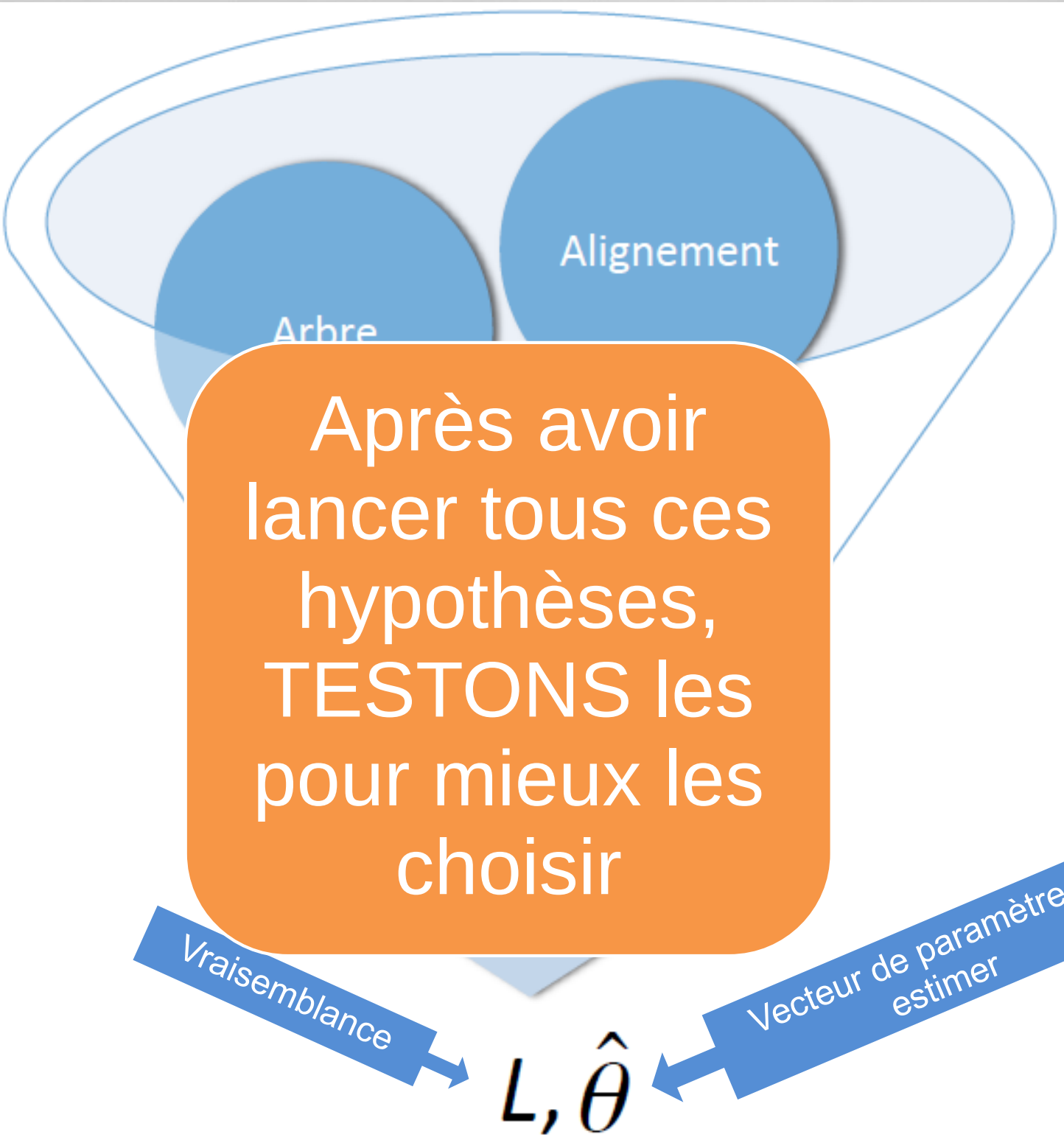
Pose une hypothèse d'évolution sur cette branche et ses descendantes



Calcul branch-site

```
((Homo:0.1,Pan:0.1)$1:0.8,Mouse:0.6);
```




Alignement
Arbre
Après avoir lancer tous ces hypothèses, TESTONS les pour mieux les choisir
Vraisemblance
Vecteur de paramètres à estimer
$L, \hat{\theta}$

Tester les résultats

On compare toujours un modèle alternatif (M_1) par exemple) à son équivalent nul (ici M_0).

Les 2 modèles doivent avoir la même « structure », M0 étant une version « plus simple » de M1 i.e. avec moins de paramètres.

M2 sera comparé à M1

M8 sera comparé à M7

M8a servira de second modèle de comparaison pour M8

Tester les résultats

Ainsi M0 et M1 sont des modèles emboîtés.

Dans ce cas, leurs vraisemblances se comporte de la manière suivante:

$$\Delta = 2 \times \ln\left(\frac{L_1}{L_0}\right) \rightarrow \chi^2_\delta$$

δ = différence du nombre de paramètres de M_1 par rapport à M_0

Les ln*L* sont donnés dans le fichier de sortie rst

=> M_1 est significativement plus vraisemblable que M_0 si Δ est supérieur au χ^2 seuil correspondant. Ceci est un LRT : Likelihood Ratio Test.

Tester les résultats: exemple

- Prenons un exemple… On lance 100 fois une pièce de monnaie, on obtient 65 faces et 35 piles.

- Notre hypothèse nulle est que la pièce n'est pas truquée. Calculons la vraisemblance L_0 de ce modèle.

Tester les résultats: exemple

- Prenons un exemple… On lance 100 fois une pièce de monnaie, on obtient 65 faces et 35 piles.

- Notre hypothèse nulle est que la pièce n'est pas truquée. Calculons la vraisemblance L_0 de ce modèle. Ici M_0=loi binomiale de paramètres $\theta=(n=100, p=0.5)$ i.e. aucun paramètre à estimer.

Probabilité d'avoir face est de 50%

$$L_0 = P(D / M, \theta) =$$

Tester les résultats: exemple

- Prenons un exemple… On lance 100 fois une pièce de monnaie, on obtient 65 faces et 35 piles.

- Notre hypothèse nulle est que la pièce n'est pas truquée. Calculons la vraisemblance L_0 de ce modèle. Ici M_0=loi binomiale de paramètres $\theta=(n=100, p=0.5)$ i.e. aucun paramètre à estimer.

$$L_0 = P(D/M,\theta) = \binom{n}{65} p^{65}(1-p)^{35} = \binom{100}{65} 0.5^{65}(1-0.5)^{35} = 0.000864$$

$$\ln(L_0) = -7.054$$

Tester les résultats: exemple

- Prenons un exemple… On lance 100 fois une pièce de monnaie, on obtient 65 faces et 35 piles.

- Notre hypothèse nulle est que la pièce n'est pas truquée. Calculons la vraisemblance L_0 de ce modèle. Ici M_0=loi binomiale de paramètres $\theta=(n=100, p=0.5)$ i.e. aucun paramètre à estimer.

$$L_0 = P(D/M,\theta) = \binom{n}{65} p^{65}(1-p)^{35} = \binom{100}{65} 0.5^{65}(1-0.5)^{35} = 0.000864$$

$$\ln(L_0) = -7.054$$

Maintenant, continuons

Tester les résultats: exemple

- Prenons un exemple… On lance 100 fois une pièce de monnaie, on obtient 65 faces et 35 piles.

- Notre hypothèse de travail est que la pièce est truquée (avec $0 \leq p \leq 1$). La valeur la plus vraisemblable pour p est

Tester les résultats: exemple

- Prenons un exemple… On lance 100 fois une pièce de monnaie, on obtient 65 faces et 35 piles.

- Notre hypothèse de travail est que la pièce est truquée (avec $0 \leq p \leq 1$). La valeur la plus vraisemblable pour p est

$$\hat{p} = 65/100 = 0.65$$

- C'est l'estimation au maximum de vraisemblance. Calculons la vraisemblance L_1 de ce modèle. Ici M_1=loi binomiale de paramètres $\theta=(n=100, p=0.65)$ i.e. 1 paramètre à estimé.

Tester les résultats: exemple

- Prenons un exemple… On lance 100 fois une pièce de monnaie, on obtient 65 faces et 35 piles.

- Notre hypothèse de travail est que la pièce est truquée (avec $0 \leq p \leq 1$). La valeur la plus vraisemblable pour p est

$$\hat{p} = 65/100 = 0.65$$

- C'est l'estimation au maximum de vraisemblance. Calculons la vraisemblance L_1 de ce modèle. Ici M_1=loi binomiale de paramètres $\theta=(n=100, p=0.65)$ i.e. 1 paramètre à estimé.

$$L_1 = P(D/M,\theta) = \binom{n}{65} p^{65}(1-p)^{35} = \binom{100}{65} 0.65^{65}(1-0.65)^{35} = 0.08340$$

$$\ln(L_1) = -2.484$$

C'est mieux, mais est-ce significativement mieux?

Tester les résultats: exemple

- Que donne le LRT ?

LRT = Log Ration Test

$$\ln(L_0) = -7.054$$
$$\ln(L_1) = -2.484$$

$$\Delta = 2(\ln(L_1) - \ln(L_0)) = 9.14$$

Le LRT est supérieur au seuil du Chi2 donc ?

Nombre de paramètres:

$\delta = 1-0 = 1$

D'où

$\chi^2_{\text{seuil}\,;\,\alpha=0.05} = 3.84$

Seuil trouvé dans la table du Chi2

Le modèle M1 est significativement plus vraisemblable que le modèles M0 au risque α de première espèce égale à 5%

Exo 3

Test hypotheses about molecular evolution of Ldh

Exo 4

To test for sites evolving under positive selection in the nef gene.

On a vu dans l'exo4 :

La detection de site sous selection positive par "Naive Empirical Bayes (NEB) analysis" notamment sous les modèles 2, 3 et 8.

NOTONS: on ne regarde ces sites si et seulement si le LRT est statistiquement significatif !

Mais il existe une méthode plus précise " Bayes Empirical Bayes (BEB) analysis" notamment sous les modèles 2 et 8.

BEB dans le fichier de résultats

Dans l'analyse par site et l'analyse branche-site, une méthode bayésienne est implantée pour détecter précisément les résidus sous sélection.

Cette méthode BEB (Bayes Empirical Bayes) renvoie une liste de résidus sous sélection positive avec une **probabilité postérieure** associée.

```
Bayes Empirical Bayes (BEB) analysis (Yang, Wong & Nielsen 2005. Mol. Biol. Evol. 22:1107-1118)
Positively selected sites (*: P>95%; **: P>99%)
(amino acids refer to 1st sequence: MBCCFI)

            Pr(w>1)     post mean +- SE for w

    51 I      0.949         3.367 +- 0.571
    52 P      1.000**       3.498 +- 0.042
    54 E      0.957*        3.388 +- 0.525
    97 N      0.625         2.515 +- 1.270
    99 T      0.998**       3.492 +- 0.132
   100 D      0.990**       3.474 +- 0.255
   101 T      1.000**       3.498 +- 0.040
   102 K      1.000**       3.498 +- 0.060
   103 T      1.000**       3.498 +- 0.044
   104 N      0.999**       3.496 +- 0.084
```

Voilà en (très) gros, comment ça marche

L'optimisation des paramètres par codeml est donc l'étape la plus couteuse en temps cpu.

Elle est d'autant plus couteuse que le nombre de paramètres est grand.

Sources

http://evomics.org/learning/phylogenetics/paml/

http://www.molecularevolution.org/resources/activities/paml_activity

http://mbe.oxfordjournals.org/content/30/12/2723.abstract.html?etoc

http://abacus.gene.ucl.ac.uk/software/pamlDOC.pdf

http://abacus.gene.ucl.ac.uk/software/paml.html

Une publication et un ouvrage de **Ziheng Yang** :

Yang (2007) PAML 4: phylogenetic analysis by maximum likelihood. Mol. Biol. Evol. 24(8):1586-91.
Yang (2006) Computational Molecular Evolution. Ed. Oxford University Press.

Un point rapide sur le KA/KS en 2 pages :

Hurst LD (2002) The Ka/Ks ratio: diagnosing the form of sequence evolution. Trends in Genetics 18: 486.

Exo 1

The objective of this activity is to use CODEML to evaluate the likelihood of the GstD1 sequences for a variety of ω values.

Dataset: *GstD1* genes of *Drosophila melanogaster* and *D. simulans* (600 codons).

Glutathione S transferase D1
Conjugation of reduced glutathione to a wide number of exogenous and endogenous hydrophobic electrophiles. Has DDT (pesticide) dehydrochlorinase activity.

Objective: Use codeml to evaluate the likelihood the *GstD1* sequences for a variety of fixed ω values.

1- Plot log-likelihood scores against the values of ω and determine the maximum likelihood estimate of ω.

2- Check your finding by running codeml's hill-climbing algorithm.

Exercice 1

1.Find the files for exercise 1 on the workshop web-site (ex1_codeml.ctl, ex1_seqfile.txt) and familiarize yourself with them. Pay close attention to the modified control file called ex1_codeml.ctl. When you are ready to run CODEML, delete the ex1_ prefix from the control file and the seq file (e.g., the control file must be called **codeml.ctl**).

2.Create a directory where you want your results to go, and place all your files within it. Now open a terminal, move to the directory that contains your files, and run CODEML.

3.Familiarize yourself with the results (ex1_HelpFile.pdf). If you have not edited the control file the results will be written to a file called results.txt. Identify the line within the results file that gives the likelihood score for the example dataset.

Exercice 1

4. Now change the control files and re-run CODEML. The objective is to compute the likelihood of the example dataset given a fixed value of omega.
    1. Change the name of your result file (via **outfile=** in the control file) or you will overwrite your previous results!
    2. Change the fixed value for omega by changing the value for **omega=** in the control file. The values for this exercise are provided as comments at the bottom of the example control file that has been provided to you.

5. Repeat step 4 for each value of omega given in the comments of the example control file.

Exercice 1

6. Use your favorite spread sheet or statistical package to plot the likelihood score (y-axis) against the fixed value for omega (x-axis). Use a **logarithmic scale** for the x-axis (do not transform omega). Your figure should look something like this: FigE1.pdf.

7. From the plot, try to guess the value of omega that will maximize the likelihood score (i.e., the MLE).

8. Now change the control file so that CODEML will use its hill-climbing algorithm to find the MLE; set **fix_omega=0** in the control file. Compare the result to your guess from step 7.

Exercice 1

Le maximum de vraisemblance à trouver sur le plot doit être ~0,07

Quand on fixe fix_omega = 0
(c.-à-d. qu'on demande a codeml d'estimer lui-même le oméga),
il le fixe à 0,0670
(dN/dS dans /home/gpascal/Formation_Plateforme/Module4/exo1/results_0.11.txt)

Exo 2

Exercice 2

Dataset:	*GstD1* genes of *Drosophila melanogaster* and *D. simulans* (600 codons).
Objective:	1- Test effect of transition / transversion ratio (κ) 2- Test effect of codon frequencies (π_i's) 3- Determine which assumptions yield the largest and smallest values of *S*, and what is the effect on ω

Exercice 2

Find the files for Exercise 2 on the workshop web-site (ex2_codeml.ctl, ex2_seqfile.txt) and familiarize yourself with them. It would be best to create a new directory for exercise 2.

Run CODEML using the settings in the control file for exercise 2. Familiarize yourself with the results (ex2_HelpFile.pdf). In addition to the likelihood score you must be able to identify the part of the result file that provides estimates of the following:

Number of synonymous or nonsynonymous sites (S and N)

Synonymous and nonsynonymous rates (dS and dN)

Exercice 2

As in exercise 1, you will need to change the control files and re-run CODEML. The objective is to compute the likelihood of the example dataset under different model assumptions. To do this you must:

Change the name of the main result file (via outfile= in the control file) or you will overwrite your previous results

Change the model assumptions about codon frequencies (via CodonFreq=) and kappa (via kappa= and fix_kappa=).

Repeat step 3 for each set of assumptions about codon frequencies and kappa given as comments at the bottom of the example control file.

Exercice 2

In your favorite spreadsheet program create a table like “Table E2” in the slides (TableE2.pdf) and fill it in with your results.

Use your table to determine which assumptions yield the largest and smallest values of S.

What is the effect on omega?

Exercice 2

With this exercise we explore the effects of

(i) ignoring the transition to transversion rate ratio (fix_kappa = 1; kappa = 1)

(ii) ignoring codon usage bias (CodonFreq = 0)

(iii) alternative treatments of unequal codonvfrequencies (CodonFreq = 2 and CodonFreq = 3)

Note that for these data, transitions are occurring at higher rates than transversions and codon frequencies are very biased, with average base frequencies of 6%(T), 50% (C), 5% (A) and 39% (G) at the third position of the codon.

Thus, we expect estimates accounting for both biases will be the most reliable.

Exercice 2

"CodonFreq=" is used to specify the equilibrium codon frequencies

Fequal: - 1/61 each for the standard genetic code
- CodonFreq = 0
- number of parameters in the model = 0

F3x4: - caluclated from the average nucleotide frequencies at the three codon positions
- CodonFreq = 2
- number of parameters in the model = 9

F61 - also called "ftable"; empirical estimate of each codon frequency
- CodonFreq = 3
- number of parameters in the model = 61

Exercice 2

```
      seqfile = seqfile.txt   * sequence data filename
      outfile = results.txt   * main result file name

        noisy = 9       * 0,1,2,3,9: how much rubbish on the screen
      verbose = 1       * 1:detailed output
      runmode = -2      * -2:pairwise

      seqtype = 1       * 1:codons
    CodonFreq = 0       * 0:equal, 1:F1X4, 2:F3X4, 3:F61 [CHANGE THIS]
        model = 0       *
      NSsites = 0       *
        icode = 0       * 0:universal code

    fix_kappa = 1       * 1:kappa fixed, 0:kappa to be estimated [CHANGE THIS]
        kappa = 1       * fixed or initial value

    fix_omega = 0       * 1:omega fixed, 0:omega to be estimated
        omega = 0.5     * initial omega value

* Codon bias = none (equal); Ts/Tv bias = none (fixed at 1)
* Codon bias = none (equal); Ts/Tv bias = Yes (estimate by ML)

* Codon bias = yes (F3x4); Ts/Tv bias = none (fixed at 1)
* Codon bias = yes (F3x4); Ts/Tv bias = Yes (estimate by ML)

* Codon bias = yes (F61); Ts/Tv bias = none (fixed at 1)
* Codon bias = yes (F61); Ts/Tv bias = Yes (estimate by ML)
```

Exercice 2

Further details for about the assumptions tested in ACTIVITY 2

Assumption set 1: (Codon bias = none; Ts/Tv bias = none)
CodonFreq=0; kappa=1; fix_kappa=1

Assumption set 2: (Codon bias = none; Ts/Tv bias = Yes)
CodonFreq=0; kappa=1; fix_kappa=0

Assumption set 3: (Codon bias = yes [F3x4]; Ts/Tv bias = none)
CodonFreq=2; kappa=1; fix_kappa=1

Assumption set 4: (Codon bias = yes [F3x4]; Ts/Tv bias = Yes)
CodonFreq=2; kappa=1; fix_kappa=0

Assumption set 5: (Codon bias = yes [F61]; Ts/Tv bias = none)
CodonFreq=3; kappa=1; fix_kappa=1

Assumption set 6: (Codon bias = yes [F61]; Ts/Tv bias = Yes)
CodonFreq=3; kappa=1; fix_kappa=0

Complete this table

Table E2: Estimation of d_S and d_N between *Drosophila melanogaster* and *D. simulans GstD1* genes

Assumptions	κ	S	N	d_S	d_N	ω	ℓ
Fequal + $\kappa = 1$	1.0	?	?	?	?	?	?
Fequal + κ = estimated	?	?	?	?	?	?	?
F3×4 + $\kappa = 1$	1.0	?	?	?	?	?	?
F3×4 + κ = estimated	?	?	?	?	?	?	?
F61 + $\kappa = 1$	1.0	?	?	?	?	?	?
F61 + κ = estimated	?	?	?	?	?	?	?

κ = transition/transversion rate ratio

S = number of synonymous sites

N = number of nonsynonymous sites

$\omega = d_N/d_S$

ℓ = log likelihood score

Estimation of d_S and d_N between *Drosophila melanogaster* and *D. simulans GstD1* genes

Method	κ	S	N	d_S	d_N	ω	ℓ
ML methods							
Fequal, $\kappa = 1$	1	152.9	447.1	0.0776	0.0213	0.274	-927.18
Fequal, κ estimated	1.88	165.8	434.2	0.0221	0.0691	0.320	-926.28
F3x4, $\kappa = 1$	1	70.6	529.4	0.1605	0.0189	0.118	-844.51
F3x4, κ estimated	2.71	73.4	526.6	0.1526	0.0193	0.127	-842.21
F61, $\kappa = 1$	1	40.5	559.5	0.3198	0.0201	0.063	-758.55
F61, κ estimated	2.53	45.2	554.8	0.3041	0.0204	0.067	-756.57

Exo2: conclusion

Results of our exploratory analyses (TableE2) indicate that model assumptions are very important for these data.

For example, ignoring the transition to transversion ratio almost always led to underestimation of the number of synonymous sites (S), overestimation of dS, and underestimation of w.

This is because transitions at the third codon positions are more likely to be synonymous than transversions are (Li 1985).

Exo2: conclusion

Similarly, biased codon usage implies unequal substitution rates between the codons, and ignoring it also leads to biased estimates of synonymous and nonsynonymous substitution rates.

In real data analysis, codon usage bias was noted to have an even greater impact than the transition/transversion rate ratio, and is opposite to that of ignoring transition bias.

This is clearly indicated by the sensitivity of S to codon bias, where S in this gene (45.2) is less than one third the expected value under the assumption of no codon bias (S = 165.8).

The estimates of ω differ by as much as 4.7 fold (Table 1). Note that these two sequences differed at just 3% of sites.

Exo 3

Dataset:	The *Ldh* gene family is an important model system for molecular evolution of isozyme multigene families. The rate of evolution is known to have increased in *Ldh*-C following the gene duplication event
Objective:	Use LRTs to evaluate the following hypotheses:

1- The mutation rate of *Ldh-C* has increased relative to *Ldh-A*,

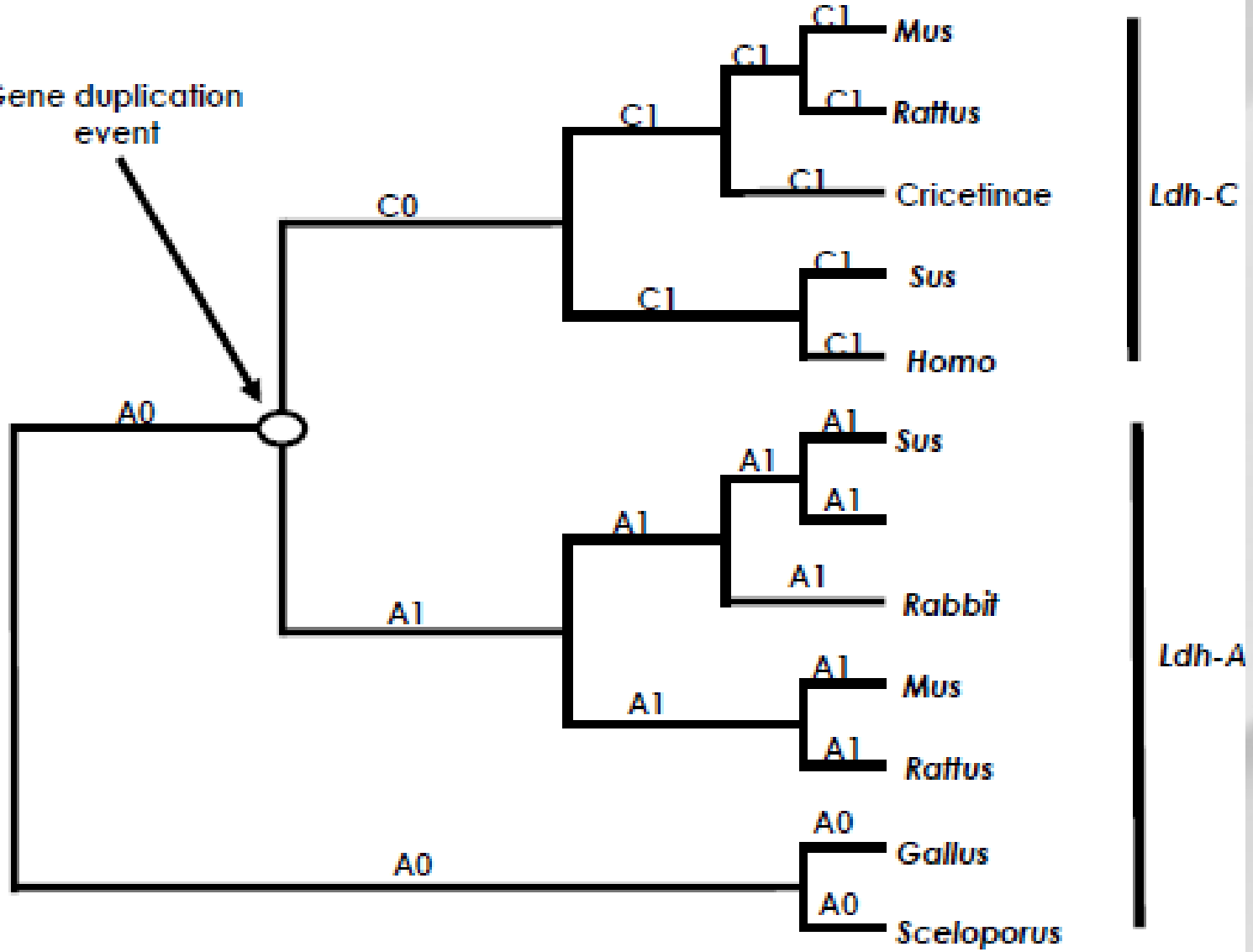
2- A burst of positive selection for functional divergence occurred following the duplication event that gave rise to *Ldh-C*

3- There was a long term shift in selective constraints following the duplication event that gave rise to *Ldh-C*

Exercise 3

Obtain the files for Exercise 3 from the course web-site (ex3_codeml.ctl, ex3_seqfile.txt, treeH0.txt, treeH1.txt, treeH2.txt, treeH3.txt).

The tree files represent different hypotheses denoted H0, H1, H2 & H3 (LDH_tree.pdf). These hypotheses represent the following concepts:



$$H_0: \omega_{A0} = \omega_{A1} = \omega_{C1} = \omega_{C0}$$
$$H_1: \omega_{A0} = \omega_{A1} = \omega_{C1} \neq \omega_{C0}$$
$$H_2: \omega_{A0} = \omega_{A1} \neq \omega_{C1} = \omega_{C0}$$
$$H_3: \omega_{A0} \neq \omega_{A1} \neq \omega_{C1} = \omega_{C0}$$

Exercise 3

H0: homogeneous selection pressure over the tree.

H1: episodic change in selection pressure in Ldh-C (only in the branch that immediately follows the gene duplication event).

H2: Long term shift in selection pressure in Ldh-C only; Ldh-C has a permanent change in selection pressure (as compared to its ancestors) whereas Ldh-A remains subject to the ancestral level of selection pressure.

H3: Long term shift in selection on both Ldh-C and Ldh-A; those lineages are subject to selection pressures different from each other and from the ancestor.

Exercise 3

Run CODEML using the settings in the control file for Exercise 3. Familiarize yourself with the results (ex3_HelpFile.pdf)

In addition to the likelihood score you must be able to identify the branch-specific estimates of the omega parameter.

(In the first run, the branch specific values for omega will all be the same. In later runs there will be differences among some branches).

Exercise 3

As in the previous exercises, you will need to change the control files and re-run CODEML.

The objective is to compute the likelihood, and estimate omega parameters, under different models of how selection pressure changes in different parts of the tree.

Because the relevant model information is contained in the tree file, you will need several tree files (obtained from the course web site) and change the control file so that it reads the different tree files.

Exercise 3

As always, you should change the name of the main result file (via outfile= in the control file) or you will overwrite your previous results.

Change the model assumptions about branch specific omega values by changing the tree files (via treefile= and model=) set within the control file.

```
      seqfile = seqfile.txt   * sequence data filename
     treefile = tree.txt      * tree structure file name [CHANGE THIS]
      outfile = results.txt   * main result file name

        noisy = 9      * 0,1,2,3,9: how much rubbish on the screen
      verbose = 1      * 1:detailed output
      runmode = 0      * 0:user defined tree

      seqtype = 1      * 1:codons
    CodonFreq = 2      * 0:equal, 1:F1X4, 2:F3X4, 3:F61

        model = 0      * 0:one omega ratio for all branches
                       * 1:separate omega for each branch
                       * 2:user specified dN/dS ratios for branches

      NSsites = 0      *

        icode = 0      * 0:universal code

    fix_kappa = 0      * 1:kappa fixed, 0:kappa to be estimated
        kappa = 2      * initial or fixed kappa

    fix_omega = 0      * 1:omega fixed, 0:omega to be estimated
        omega = 0.2    * initial omega


*H0 in Table 3:
*model = 0
*(X02152Hom,U07178Sus,(M22585rab,((NM017025Rat,U13687Mus),
*(((AF070995C,(X04752Mus,U07177Rat)),(U95378Sus,U13680Hom)),(X53828OG1,
* U28410OG2)))));

*H1 in Table 3:
*model = 2
*(X02152Hom,U07178Sus,(M22585rab,((NM017025Rat,U13687Mus),(((AF070995C,
*(X04752Mus,U07177Rat)),(U95378Sus,U13680Hom))#1,(X53828OG1,U28410OG2))
* )));

*H2 in Table 3:
*model = 2
* (X02152Hom,U07178Sus,(M22585rab,((NM017025Rat,U13687Mus),(((AF070995C
* #1,(X04752Mus #1,U07177Rat #1)#1)#1,(U95378Sus #1,U13680Hom #1)
* #1)#1,(X53828OG1,U28410OG2)))));
```

Exercise 3

Repeat step 3 for each of the four tree files that have been provided to you.

Again, keep track of your results by using a table like “Table E3” shown in the slides (TableE3.pdf).

In addition, carry out likelihood ratio tests (LRT) of the hypotheses below.

Use 1 degree of freedom for each LRT.
Helpful: Chi-Square Calculator.

H0 vs. H1

H0 vs. H2

H2 vs. H3

Table E3: Parameter estimates under models of variable ω ratios among lineages and LRTs of their fit to the *Ldh-A* and *Ldh-C* gene family.

Models	ω_{A0}	ω_{A1}	ω_{C1}	ω_{C0}	ℓ	LRT
H_0: $\omega_{A0} = \omega_{A1} = \omega_{C1} = \omega_{C0}$	?	$= \omega_{A.0}$	$= \omega_{A.0}$	$= \omega_{A.0}$	?	?
H_1: $\omega_{A0} = \omega_{A1} = \omega_{C1} \neq \omega_{C0}$	?	$= \omega_{A.0}$	$= \omega_{A.0}$	?	?	?
H_2: $\omega_{A0} = \omega_{A1} \neq \omega_{C1} = \omega_{C0}$	?	$= \omega_{A.0}$	?	$= \omega_{C.1}$	?	?
H_3: $\omega_{A0} \neq \omega_{A1} \neq \omega_{C1} = \omega_{C0}$	?	?	?	$= \omega_{C.1}$	?	?

The topology and branch specific ω ratios are presented in Figure 5.

H_0 v H_1: df = 1

H_0 v H_2: df = 1

H_2 v H_3: df = 1

$\chi^2_{df=1,\ \alpha=0.05} = 3.841$

Models	w_{A0}	w_{A1}	w_{C1}	w_{C0}	ℓ
$H_0 : w_{A0} = w_{A1} = w_{C1} = w_{C0}$	0.14	$= w_{A0}$	$= w_{A0}$	$= w_{A0}$	-6018.63
$H_1 : w_{A0} = w_{A1} = w_{C1} \neq w_{C0}$	0.13	$= w_{A0}$	$= w_{A0}$	0.19	-6017.57
$H_2 : w_{A0} = w_{A1} \neq w_{C1} = w_{C0}$	0.07	$= w_{A0}$	0.24	$= w_{A1}$	-5985.63
$H_3 : w_{A0} \neq w_{A1} \neq w_{C1} = w_{C0}$	0.09	0.06	0.24	$= w_{A1}$	-5984.11

Note: The topology and branch specific ω ratios are presented in Fig5. The d.f. is 1 for the comparisons of H0 vs. H1, H0 vs. H2, and H2 vs. H3.

Dans resultH1.txt trouver la ligne:
w (dN/dS) for branches: 0.13188 0.19203

Models	w_{A0}	w_{A1}	w_{C1}	w_{C0}	ℓ
$H_0 : w_{A0} = w_{A1} = w_{C1} = w_{C0}$	0.14	$= w_{A0}$	$= w_{A0}$	$= w_{A0}$	-6018.63
$H_1 : w_{A0} = w_{A1} = w_{C1} \neq w_{C0}$	0.13	$= w_{A0}$	$= w_{A0}$	0.19	-6017.57
$H_2 : w_{A0} = w_{A1} \neq w_{C1} = w_{C0}$	0.07	$= w_{A0}$	0.24	$= w_{A1}$	-5985.63
$H_3 : w_{A0} \neq w_{A1} \neq w_{C1} = w_{C0}$	0.09	0.06	0.24	$= w_{A1}$	-5984.11

Note: The topology and branch specific ω ratios are presented in Fig5. The d.f. is 1 for the comparisons of H0 vs. H1, H0 vs. H2, and H2 vs. H3.

LRT HO vs. H1	2,12
LRT HO vs. H2	66
LRT H2 vs. H3	3,04

$$\chi^2_{df=1,\ \alpha=0.05} = 3.841$$

Seule l'hypothèse H2 vs. H0 est statistiquement significative.

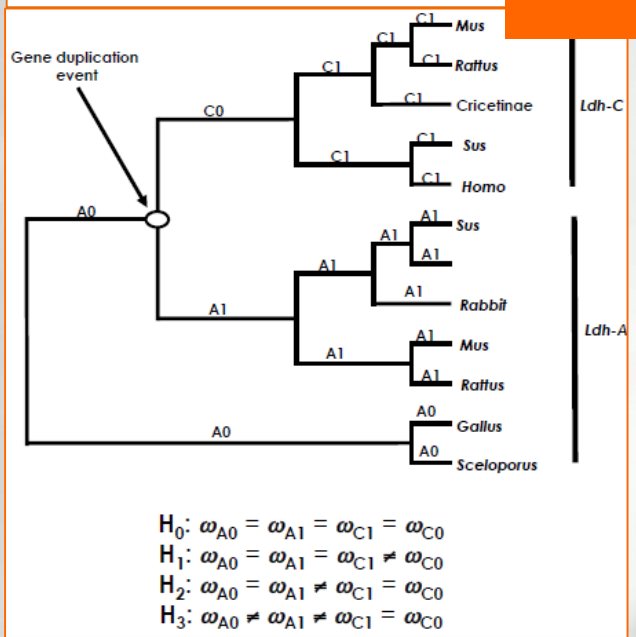
Note that if functional divergence of Ldh-A and Ldh-C evolved by positive selection for just one or a few amino acid changes, we would not observe a large difference in ω ratios among branches.

Models	w_{A0}	w_{A1}	w_{C1}	w_{C0}	ℓ
$H_0 : w_{A0} = w_{A1} = w_{C1} = w_{C0}$	0.14	$= w_{A0}$	$= w_{A0}$	$= w_{A0}$	-6018.63
	0.13	$= w_{A0}$	$= w_{A0}$	0.19	-6017.57
	0.07	$= w_{A0}$	0.24	$= w$	
$H_3 : w_{A0} \neq$					

Ldh-A and Ldh-C was dominated by purifying selection

no evidence for functional divergence of Ldh-A and Ldh-C by positive selection

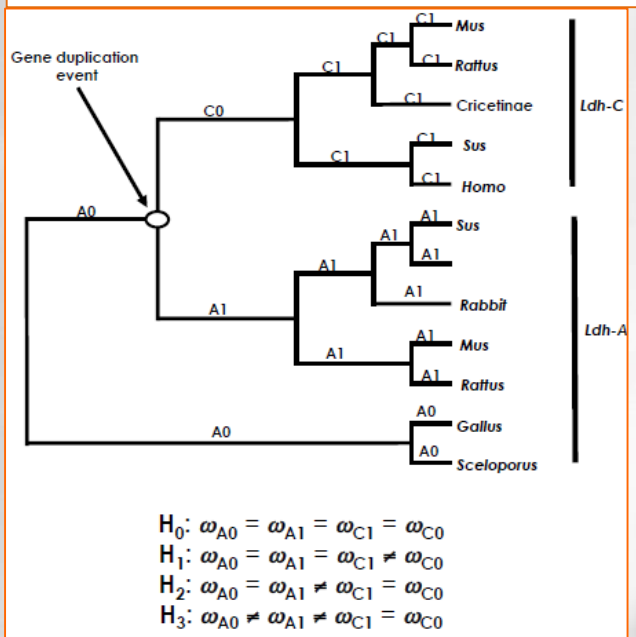
LRT suggests that selective pressure in Ldh-C immediately following gene duplication (0.19) was not significantly different than the average over the other branches



LRT HO vs. H1	2,12
LRT HO vs. H2	66
LRT H2 vs. H3	3,04

We want to test H1: episodic change in selection pressure in Ldh-C (only in the branch that immediately follows the gene duplication event).

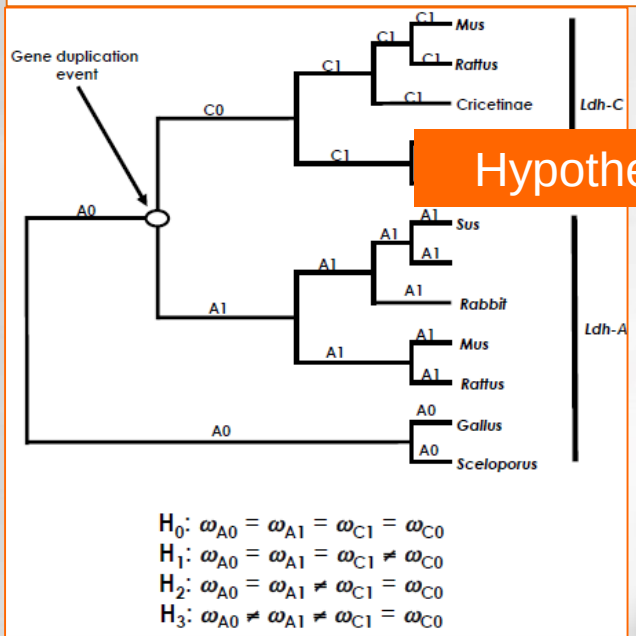
Models	w_{A0}	w_{A1}	w_{C1}	w_{C0}	ℓ
$H_0 : w_{A0} = w_{A1} = w_{C1} = w_{C0}$	0.14	$= w_{A0}$	$= w_{A0}$	$= w_{A0}$	-6018.63
$H_1 : w_{A0} = w_{A1} = w_{C1} \neq w_{C0}$	0.13	$= w_{A0}$	$= w_{A0}$	0.19	-6017.57
$H_2 : w_{A0} = w_{A1} \neq w_{C1} = w_{C0}$	0.07	$= w_{A0}$	0.24	$= w_{A1}$	-5985.63
$H_3 : w_{A0} \neq w_{A1} \neq w_{C1} = w_{C0}$	0.09	0.06	0.24	$= w_{A1}$	-5984.11



LRT HO vs. H1	2,12
LRT HO vs. H2	66
LRT H2 vs. H3	3,04

The ω rate increase in Ldh-C was due to an increase in the nonsynonymous substitution rate over all lineages of the Ldh-C clade.
LRT highly significant

Models	w_{A0}	w_{A1}	w_{C1}	w_{C0}	ℓ
$H_0 : w_{A0} = w_{A1} = w_{C1} = w_{C0}$	0.14	$= w_{A0}$	$= w_{A0}$	$= w_{A0}$	-6018.63
$H_1 : w_{A0} = w_{A1} = w_{C1} \neq w_{C0}$	0.13	$= w_{A0}$	$= w_{A0}$	0.19	-6017.57
$H_2 : w_{A0} = w_{A1} \neq w_{C1} = w_{C0}$	0.07	$= w_{A0}$	0.24	$= w_{A1}$	-5985.63
$H_3 : w_{A0} \neq w_{A1} \neq w_{C1} = w_{C0}$	0.09	0.06	0.24	$= w_{A1}$	-5984.11



Hypothesis no significant

LRT HO vs. H1	2,12
LRT HO vs. H2	66
LRT H2 vs. H3	3,04

we tested the hypothesis that selective pressure differed in both Ldh-A and Ldh-C following gene duplication

Exo 4

The objective of this exercise is to use a series of LRTs to test for sites evolving under positive selection in the nef gene. If you find significant evidence for positive selection, then identify the involved sites by using empirical Bayes methods.

Dataset: 44 *nef* alleles from a study population of 37 HIV-2 infected people living in Lisbon, Portugal. The *nef* gene in HIV-2 has received less attention than HIV-1, presumably because HIV-2 is associated with reduced virulence and pathogenicity relative to HIV-1

Objectives: 1- Learn to use LRTs to test for sites evolving under positive selection in the *nef* gene.

2- If you find significant evidence for positive selection, then identify the involved sites by using empirical Bayes methods.

The role of the nef gene in differing phenotypes of HIV-1 infection has been well studied, including identification of sites evolving under positive selective pressure (Zanotto et al. 1999).

Padua et al. (2003) sequenced 44 nef alleles from a study population of 37 HIV-2 infected people living in Lisbon, Portugal.

They found that nucleotide variation in the nef gene, rather than gross structural change, was potentially correlated with HIV-2 pathogenesis.

In order to determine if the nef gene might also be evolving under positive selective pressure in HIV-2, we analysed those same data here with models of variable *w* ratios among sites (Yang et al. 2000).

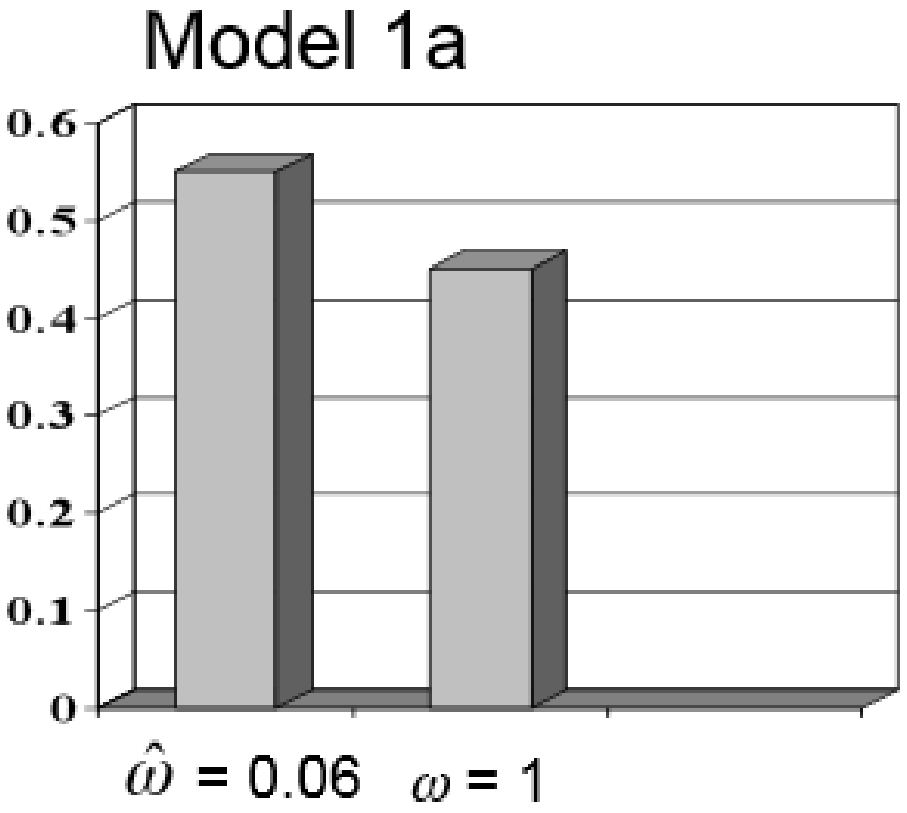
Exercice 4

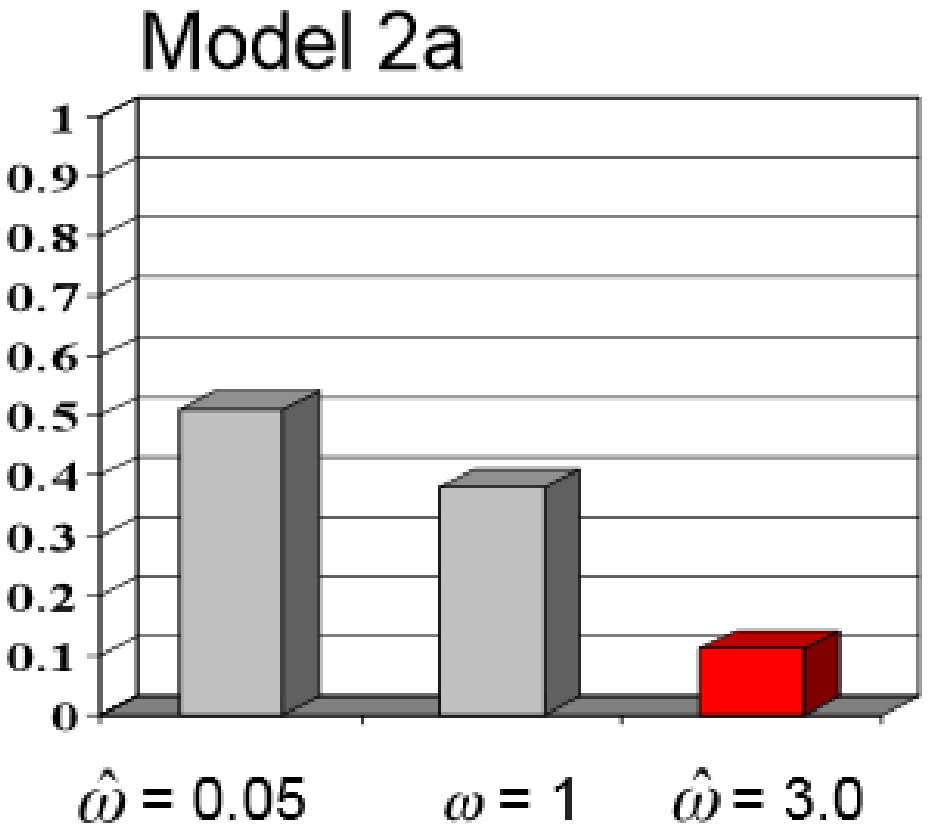
1. Obtain all the files for exercise 4 from the course website (ex4_codeml.ctl, ex4_seqfile.txt, treeM0.txt, treeM1.txt, treeM2.txt, treeM3.txt, treeM7.txt, treeM8.txt).

2. If you plan to run two or more models at the same time, then create a separate directory for each run and place a sequence file, control file and tree file in each one.

H_0: variable selective pressure but NO positive selection (M1)
H_1: variable selective pressure with positive selection (M2)

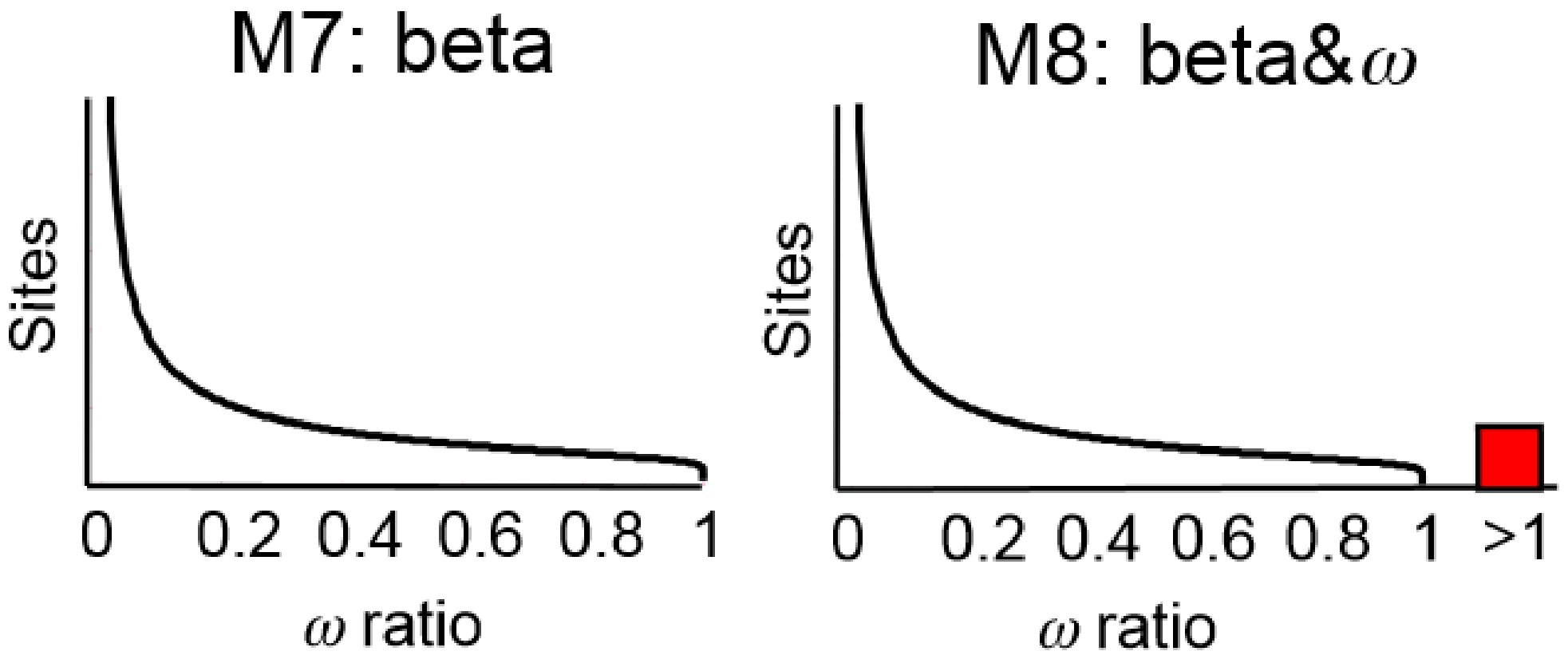
Compare $2\Delta l = 2(l_1 - l_0)$ with a χ^2 distribution





H_0: Beta distributed variable selective pressure (M7)
H_1: Beta plus positive selection (M8)

Compare $2\Delta l = 2(l_1 - l_0)$ with a χ^2 distribution



```
      seqfile = seqfile.txt          * sequence data filename

    * treefile = treefile_M0.txt     * SET THIS for tree file with ML branch lengths under M0
    * treefile = treefile_M1.txt     * SET THIS for tree file with ML branch lengths under M1
    * treefile = treefile_M2.txt     * SET THIS for tree file with ML branch lengths under M2
    * treefile = treefile_M3.txt     * SET THIS for tree file with ML branch lengths under M3
    * treefile = treefile_M7.txt     * SET THIS for tree file with ML branch lengths under M7
    * treefile = treefile_M8.txt     * SET THIS for tree file with ML branch lengths under M8

      outfile = results.txt          * main result file name
        noisy = 9                    * lots of rubbish on the screen
      verbose = 1                    * detailed output
      runmode = 0                    * user defined tree
      seqtype = 1                    * codons
    CodonFreq = 2                    * F3X4 for codon ferquencies
        model = 0                    * one omega ratio for all branches

    * NSsites = 0                    * SET THIS for M0
    * NSsites = 1                    * SET THIS for M1
    * NSsites = 2                    * SET THIS for M2
    * NSsites = 3                    * SET THIS for M3
    * NSsites = 7                    * SET THIS for M7
    * NSsites = 8                    * SET THIS for M8

        icode = 0                    * universal code
    fix_kappa = 1                    * kappa fixed
      * kappa = 4.43491              * SET THIS to fix kappa at MLE under M0
      * kappa = 4.39117              * SET THIS to fix kappa at MLE under M1
      * kappa = 5.08964              * SET THIS to fix kappa at MLE under M2
      * kappa = 4.89033              * SET THIS to fix kappa at MLE under M3
      * kappa = 4.22750              * SET THIS to fix kappa at MLE under M7
      * kappa = 4.87827              * SET THIS to fix kappa at MLE under M8

    fix_omega = 0                    * omega to be estimated
        omega = 5                    * initial omega

      * ncatG = 3                    * SET THIS for 3 site categories under M3
      * ncatG = 10                   * SET THIS for 10 of site categories under M7 and M8

  fix_blength = 2                    * fixed branch lengths from tree file
```

Exercice 4

3. As in all the previous exercises, you will need to change the control file and re-run CODEML several times. In this case you will be fitting six different codon models (M0, M1a, M2a, M3, M7 & M8) to the example dataset.
    a) If you are running your analyses sequentially in the same directory, then you should change the name of the main result file (via outfile= in the control file) or you will overwrite your previous results.
    b) Set the tree file with treefile=. I have supplied tree files pre-loaded with the ML branch lengths for each model (hence you need to set a different tree for each model). This will greatly speed up your analyses, giving you more “beer time”. See the example control file for more details about treefile names.
    c) Set the codon model with NSsites=.
    d) Fix the value of kappa at the ML estimate with kappa=. Again, this will help speed up the analysis. See the control file for the value of kappa for each model.
    e) For some models you will also need to set the number of categories (ncatG) in the omega distribution:
        a) For M3 set ncatG=3
        b) For M7 set ncatG=10
        c) For M8 set ncatG=10
    f) Once the analysis is complete, rename the rst file because subsequent runs will overwrite it!
    g) Repeat steps a. through f. for each of the six codon models listed above.

Exercice 4

4. Keep track of your results (ex4_HelpFile.pdf) by using a table like “Table E4” shown in the slides (TableE4.pdf).
5. In addition, carry out the following likelihood ratio tests:
   a) M0 vs. M3 (4 degrees of freedom)
   b) M1a vs. M2a (2 degrees of freedom)
   c) M7 vs. M8 (2 degrees of freedom)

Table E4: Parameter estimates and likelihood scores under models of variable ω ratios among sites for HIV-2 *nef* genes.

Nested model pairs	d_N/d_S [b]	Parameter estimates [c]	PSS [d]	ℓ
M0: one-ratio (1) [a]	?	ω = ?	N.A.	?
M3: discrete (5)	?	p_0 = ?, p_1 = ?, (p_2 = ?) ω_0 = ?, ω_1 = ?, ω_2 = ?	? (?)	?
M1: neutral (1)	?	p_0 = ?, (p_1 = ?) ω_0 = ?, (ω_1 = 1)	N.A.	?
M2: selection (3)	?	p_0= ?, p_1 = ?, (p_2 = ?) ω_0 = ?, (ω_1 = 1), ω_2 = ?	? (?)	?
M7: beta (2)	?	p = ?, q = ?	N.A.	?
M8: beta&ω (4)	?	p_0 = ? (p_1 = ?) p = ?, q = ?, ω = ?	? (?)	?

[a] The number after the model code, in parentheses, is the number of free parameters in the ω distribution.

[b] This d_N/d_S ratio is an average over all sites in the HIV-2 *nef* gene alignment.

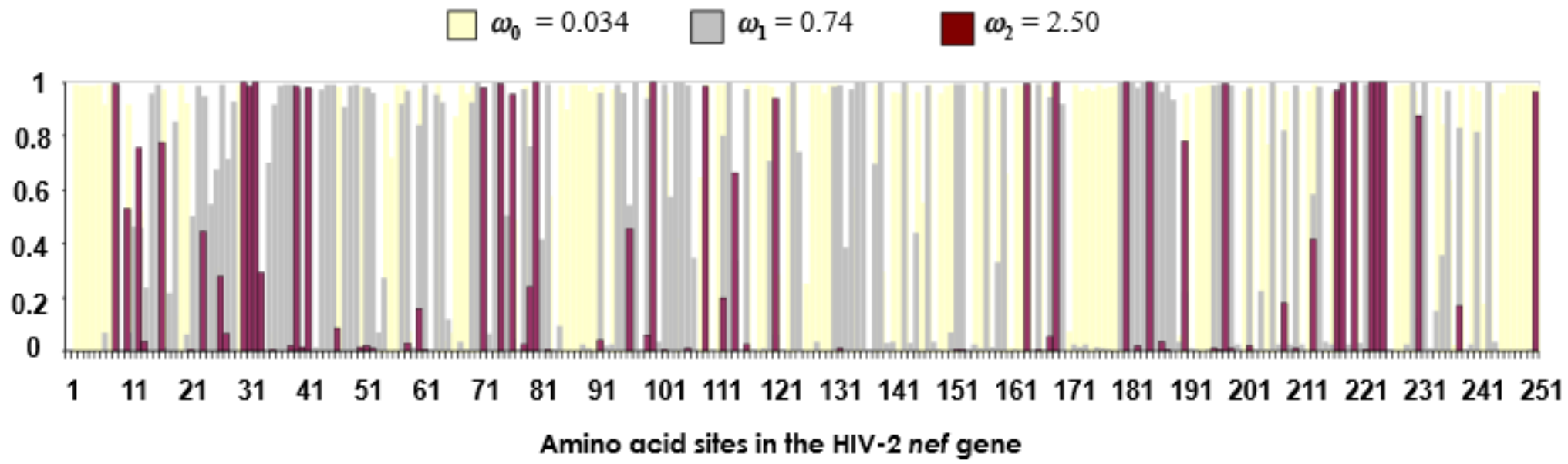
[c] Parameters in parentheses are not free parameters.

[d] PSS is the number of positive selection sites (NEB). The first number is the PSS with posterior probabilities > 50%. The second number (in parentheses) is the PSS with posterior probabilities > 95%.

Exercice 4

4. Lastly, open the rst file generated when you ran model M3 (ex4_rst_HelpFile.pdf). Locate the columns of posterior probabilities for each site under the three site-categories of this model. Use these data to reproduce the plot shown in the slides.

Reproduce this plot



Posterior probabilities for sites classes under M3 (K = 3) along the HIV-2 nef gene alignment.

Model	d_N/d_S	Parameter estimates	PSS	ℓ
M0: one-ratio (1)	0.51	$\omega = 0.505$	none	-9775.77
M3: discrete (5)	0.63	$p_0 = 0.48, p_1 = 0.39, (p_2 = 0.13)$	31 (24)	-9232.18
		$\omega_0 = 0.03, \omega_1 = 0.74, \omega_2 = 2.50$		
M1: neutral (1)	0.63	$p_0 = 0.37, (p_1 = 0.63)$	not allowed	-9428.75
		$(\omega_0 = 0), (\omega_1 = 1)$		
M2: selection (3)	0.93	$p_0 = 0.37, p_1 = 0.51, (p_2 = 0.12)$	30 (22)	-9392.96
		$(\omega_0 = 0), (\omega_1 = 1), \omega_2 = 3.48$		
M1a: NearlyNeutral (2)	0.48	$p_0 = 0.55, (p_1 = 0.45)$	not allowed	-9315.53
		$(\omega_0 = 0.06), (\omega_1 = 1)$		
M2a: PositiveSelection (4)	0.73	$p_0 = 0.51, p_1 = 0.38, (p_2 = 0.11)$	26 (15)	-9241.33
		$(\omega_0 = 0.05), (\omega_1 = 1), \omega_2 = 3.00$		
M7: beta (2)	0.42	$p = 0.18, q = 0.25$	not allowed	-9292.53
M8: beta&ω (4)	0.62	$p_0 = 0.89, (p_1 = 0.11)$	27 (15)	-9224.31
		$p = 0.20, q = 0.33, \omega = 2.62$		

Model	d_N/d_S	Parameter estimates	PSS	ℓ
M0: one-ratio (1)	0.51	$\omega = 0.505$	none	-9775.77
M3: discrete (5)	0.63	$p_0 = 0.48, p_1 = 0.39, (p_2 = 0.13)$	31 (24)	-9232.18
		$\omega_0 = 0.03, \omega_1 = 0.74, \omega_2 = 2.50$		
M1: ne				
M2: selection (3)	0.93	$p_0 = 0.37, p_1 = 0.51, (p_2 = 0.12)$	30 (22)	-9392.96
		$(\omega_0 = 0), (\omega_1 = 1), \omega_2 = 3.48$		
M1a: NearlyNeutral (2)	0.48	$p_0 = 0.55, (p_1 = 0.45)$	not allowed	-9315.53
		$(\omega_0 = 0.06), (\omega_1 = 1)$		
M2a: PositiveSelection (4)	0.73	$p_0 = 0.51, p_1 = 0.38, (p_2 = 0.11)$	26 (15)	-9241.33
		$(\omega_0 = 0.05), (\omega_1 = 1), \omega_2 = 3.00$		
M7: beta (2)	0.42	$p = 0.18, q = 0.25$	not allowed	-9292.53
M8: beta&ω (4)	0.62	$p_0 = 0.89, (p_1 = 0.11)$	27 (15)	-9224.31
		$p = 0.20, q = 0.33, \omega = 2.62$		

M0 vs M3 LRT = 1087.2, with P< 0.01, df = 4
these indicate that the selective pressure is highly variable among site

Model	d_N/d_S	Parameter estimates	PSS	ℓ
M0: one-ratio (1)	0.51	$\omega = 0.505$	none	-9775.77
M3: discrete (5)	0.63	$p_0 = 0.48, p_1 = 0.39, (p_2 = 0.13)$	31 (24)	-9232.18
		$\omega_0 = 0.03, \omega_1 = 0.74, \omega_2 = 2.50$		
M1: neutral (1)	0.63	$p_0 = 0.37, (p_1 = 0.63)$	not allowed	-9428.75
		$(\omega_0 = 0), (\omega_1 = 1)$		
M2: selection (3)	0.93	$p_0 = 0.37, p_1 = 0.51, (p_2 = 0.12)$	30 (22)	-9392.96
M1a: NearlyNeut				
		$(\omega_0 = 0.06), (\omega_1 = 1)$		
M2a: PositiveSelection (4)	0.73	$p_0 = 0.51, p_1 = 0.38, (p_2 = 0.11)$	26 (15)	-9241.33
		$(\omega_1 = 1), \omega_2 = 3.00$		
M7: beta (2)	0.42	$p = 0.18, q = 0.25$	not allowed	-9292.53
M8: beta&ω (4)	0.62	$p_0 = 0.89, (p_1 = 0.11)$	27 (15)	-9224.31
		$p = 0.20, q = 0.33, \omega = 2.62$		

M1 vs M2 LRT = 223.58 with P< 0.01, df = 2
these suggest that about 12% of sites in the nef gene of HIV-2 are elvoving under positive selective pressure, with ω between 2 and 3.

M1a vs M2a
LRT is significant

M7 vs M8
LRT is significant

The figure shows the posterior probabilities for the K = 3 site classes at each site of nef under model M3.

24 sites were identified as having very high posterior probability (P>0.95) of evolving under positive selection (site class with ω>1).

Interestingly none of these sites matched the two variable sites in a proline-rich motif that is strongly associated with an asymptomatic disease profile (Padua et al. 2003).

In fact, only 4 of the 24 sites were found in regions of nef considered important for function.

Disruption of the important nef regions is associated with reduced pathogenicity in HIV-2 infected individuals (Switzer et al. 1998; Padua et al. 2003).

Our results suggest that selective pressure at such sites is fundamentally different than selection acting at the 24 positive selection sites predicted using the Bayes theorem.

To be identified with such high posterior probabilities, the predicted sites must have been evolving under long-term positive selective pressure, suggesting that they are more likely subjected to immune-driven diversifying selection at epitopes.