

Compréhension de l'influence des variables l (taux de spéciation), m (taux d'extinction), spp (nombre d'espèces observées au temps zéro) et T (temps) par la génération d'arbres phylogénétiques

Voici tout d'abord les commandes que nous a transmis Nicolas Salamin et qu'il est nécessaire d'introduire dans « R » afin de pouvoir utiliser la fonction permettant de générer des arbres phylogénétiques variables (possibilité de sourcer ces lignes de codes se un fichier de type R les contenant existe à l'aide du chemin **Fichier**→**Sourcer du code R...**).

```
createTree<-function(t=1, l=2, m=1, spp=20, plot=FALSE) {
  if(l<m) {
    stop("Pas d'arbre possible. Le taux de spéciation est plus
petit que le taux d'extinction.\n",call.=FALSE);
  }

  if(l==0) {
    stop("Pas d'arbre possible. Le taux de spéciation est de
zéro.\n",call.=FALSE);
  }

  require(ape);
  tree<-rtree(spp);
  nonodes<-(spp*2)-1;
  times<-genTimes(t, l, m, spp-1);

  j<-2;
  set<-c(1:spp, (spp+2):nonodes);
  for(i in set) {
    k<-which(tree$edge[,2]==i);
    if(i>spp) {
      tree$edge.length[k]<-times[j];
      j<-j+1;
    }
    else {
      tree$edge.length[k]<-0;
    }
  }
}

edge<-tree$edge.length;
for(i in 1:(nonodes-1)) {
  j<-tree$edge[i,1]; #it's the ancestor, I need to find
which line as j as descendant to get its branch length

  if(j==(spp+1)) { #if j is the root, its time is times[1]
    anc<-times[1];
  }
  else { #otherwise, search the corresponding line
    k<-which(tree$edge[,2]==j);
    anc<-edge[k];
  }
}
```

```
    }  
  
    tree$edge.length[i]<-anc-edge[i];  
  }  
  
  if(plot) {  
    plot(tree);  
    axisPhylo();  
  }  
  
  return(list(tree=tree, T=times[1]));  
}  
  
genTimes<-function(t=1, l=2, m=1, spp=20) {  
  if(l==m) {  
    l<-l+0.000000001;  
  }  
  
  lt<-l*t;  
  mt<-m*t;  
  
  myexp<-exp(lt-mt);  
  
  a<-0;  
  b<-(myexp - 1)/(l*myexp - m);  
  
  times<-sort(runif(spp, a, b), decreasing=TRUE);  
  
  return((1/(l-m))*log((1-m*times)/(1-l*times)));  
}
```

On peut maintenant générer des arbres phylogénétiques en faisant varier les variables l (taux de spéciation), m (taux d'extinction), spp (nombre d'espèces observées au temps zéro) et t^1 (temps séparant le temps zéro du premier nœud ancestral n'étant pas compris dans l'arbre considéré).

```
createTree(t=..., l=..., m=..., spp=..., plot=T)
```

¹ Attention, contrairement aux codes que nous avons écrits concernant le « Birth and Death process » ainsi que le « Maximum likelihood estimation of speciation rates » pour lesquels la variable t représentait les temps séparant chaque nœud de notre arbre du sommet des branches de ce dernier, dans la commande « createTree » la variable t représente le temps séparant le premier nœud ancestral n'étant pas compris dans l'arbre considéré du temps zéro (il s'agit donc de l'équivalent de notre T).

Exemple :

```
createTree(t=4, l=2, m=1.3, spp=17, plot=T)  
$tree
```

Phylogenetic tree with 17 tips and 16 internal nodes.

Tip labels:

```
t8, t3, t4, t10, t5, t13, ...
```

Rooted; includes branch lengths.

```
$T  
[1] 2.336935
```

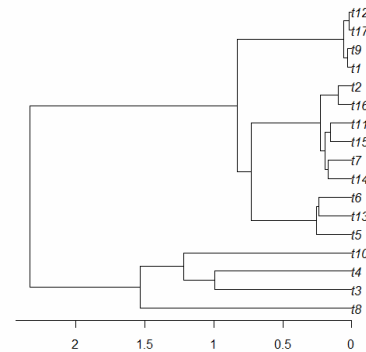
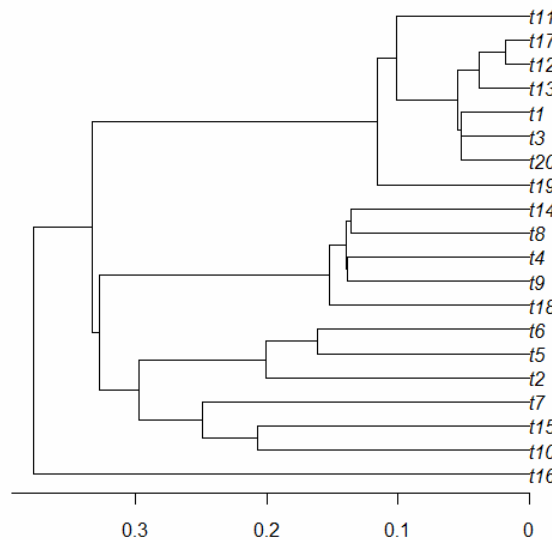


Illustration de l'influence de l et m sur la structure des arbres phylogénétiques

I. `createTree(t=4, l=5, m=0.1, spp=20, plot=T)`



Si l est beaucoup plus grand que m, alors les événements de spéciations ont du se produire relativement récemment. En effet, si tel n'était pas le cas et que les premières spéciations s'étaient produites beaucoup plus en arrière, de nombreux autres événement similaires auraient eu le temps de se produire générant ainsi un nombre significativement plus grand d'espèces observables au temps zéro que celui effectivement observé.

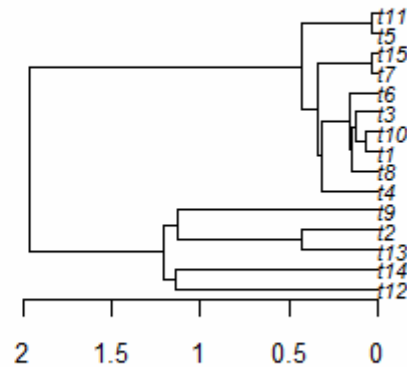
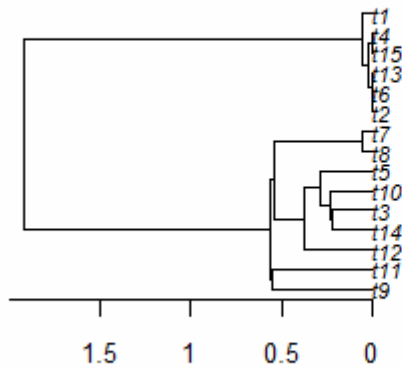
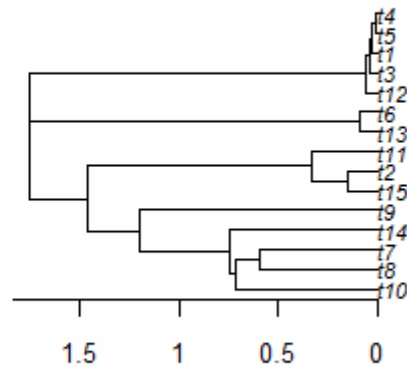
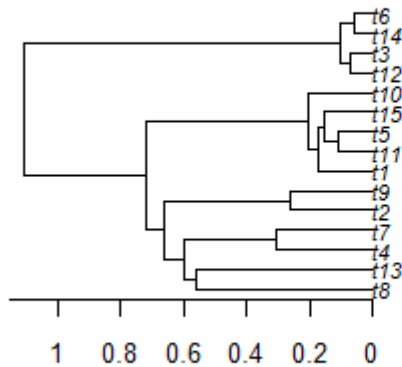
II. `createTree(t=4, l=1, m=3, spp=20, plot=T)`

Erreur : Pas d'arbre possible. Le taux de spéciation est plus petit que le taux d'extinction.

IV. `createTree(t=4, l=0, m=0, spp=20, plot=T)`

Erreur : Pas d'arbre possible. Le taux de spéciation est de zéro.

V. `createTree(t=2, l=3, m=1.5, spp=15, plot=T)`



Il est intéressant de remarquer qu'en générant plusieurs arbres à l'aide de la même commande, on obtient des arbres différents. Cela met en évidence l'aspect stochastique des processus d'évolution.

Il est également possible, à partir d'un arbre phylogénétique généré à l'aide de la fonction `createTree`, d'utiliser les valeurs de `t` qui en sont issues afin de les introduire dans notre système du maximum de vraisemblance. Voici quelques exemples :

```
I. arbre<-createTree(l=3,m=1.5,t=2,spp=23,plot=T)
arbre
$tree
```

Phylogenetic tree with 23 tips and 22 internal nodes.

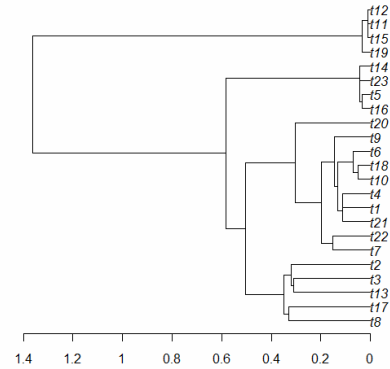
Tip labels:

```
t8, t17, t13, t3, t2, t7, ...
```

Rooted; includes branch lengths.

```
$T
```

```
[1] 1.363510
```



```
arbre$tree
```

Phylogenetic tree with 23 tips and 22 internal nodes.

Tip labels:

```
t8, t17, t13, t3, t2, t7, ...
```

Rooted; includes branch lengths.

```
t<-branching.times(arbre$tree)
```

```
t
```

	24	25	26	27	28	29
30	1.36350979	0.58223469	0.50548960	0.35123400	0.32862170	0.32109104
31	0.30888106					
	31	32	33	34	35	36
37	0.30502023	0.19750356	0.15373642	0.14509655	0.13166369	0.11246129
38	0.11139910					
	38	39	40	41	42	43
44	0.07031668	0.04927957	0.04364270	0.04237863	0.03325612	0.03285059
45	0.01152181					
45	0.01145393					

A ce stade, nous avons donc le vecteur t correspondant à l'arbre que nous avons généré, et nous connaissons également la valeur de T puisque c'est nous-même qui l'avons définie lors de l'utilisation de la fonction createTree. On peut donc produire les graphes correspondant à cet arbre à l'aide des formules que nous avons encodées concernant le maximum de vraisemblance.

```
T<-2
```

```
mat<-matrix(numeric(10000),ncol=100)
```

```
l<-seq(0.1,10,length.out=100)
```

```
m<-1
```

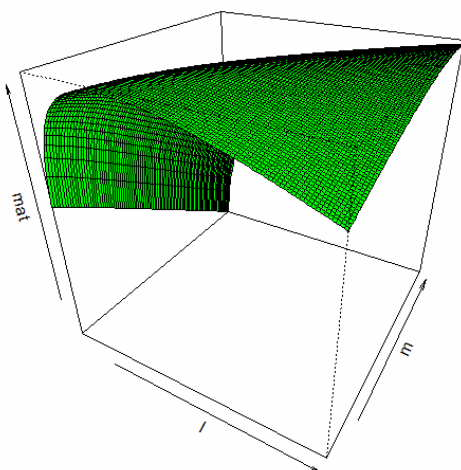
```
for(i in 1:100) {
for(j in 1:100) {
mat[i,j]<-vrais(l[i], m[j], t, T)
}}
```

```
persp(l,m,mat,main="Représentation tridimensionnelle de la
vraisemblance des valeurs de l et m concernant
arbre",col="green",theta=30,phi=30)
```

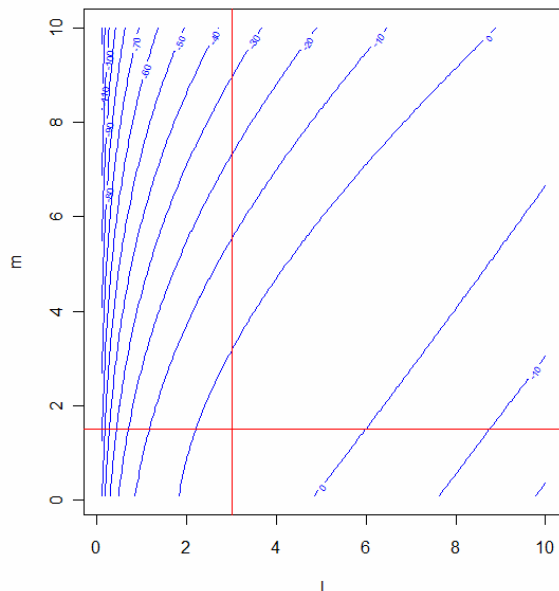
```
contour(l,m,mat,xlab="l",ylab="m",main="Représentation en
relief de la vraisemblance des valeurs de l et m concernant
arbre",col="blue")
```

```
abline(h=1.5,v=3,col="red")
```

Représentation tridimensionnelle de la vraisemblance des valeurs de l et m concernant arbre



Représentation en relief de la vraisemblance des valeurs de l et m concernant arbre



Lors de la génération de cet arbre avec la commande createTree, nous avons nous-même défini les valeurs de m et l. Sachant cela et à l'aide de la commande abline(), nous pouvons donc représenter sur la graphique contour() la position réelle du maximum de vraisemblance (intersection des droites rouges) et le comparer au résultat approximatif que nous fournit ce graphe (courbes bleues).

```
II. arbre<-createTree(l=3,m=1,spp=25,plot=T,t=2)
T<-2
t<-branching.times(arbre$tree)
mat<-matrix(numeric(10000),ncol=100)
l<-seq(0.1,10,length.out=100)
m<-1
for(i in 1:100) {
for(j in 1:100) {
mat[i,j]<-vrais(l[i], m[j], t, T)
```

```

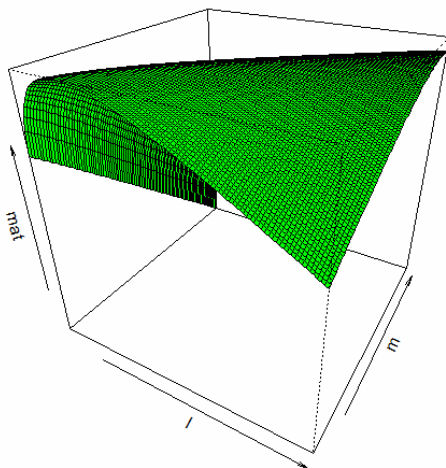
}}
persp(l,m,mat,main="Représentation tridimensionnelle de la
vraisemblance des valeurs de l et m concernant
arbre",col="green",theta=30,phi=30)

v<-c(0,-2,-4,-6,-8,-10,-20,-30,-40,-50,-60,-70,-80,-90,-100,-
120,-140,-160)

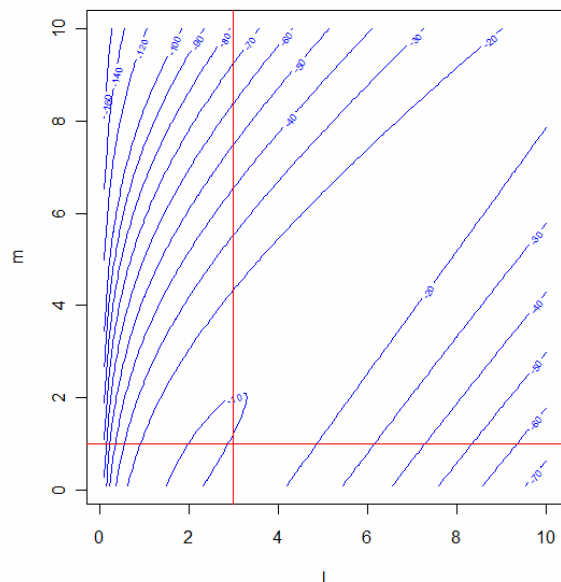
contour(l,m,mat,xlab="l",ylab="m",main="Représentation en
relief de la vraisemblance des valeurs de l et m concernant
arbre",col="blue",levels=v)

abline(h=1,v=3,col="red")
  
```

Représentation tridimensionnelle de la vraisemblance
 des valeurs de l et m concernant arbre



Représentation en relief de la vraisemblance
 des valeurs de l et m concernant arbre



```

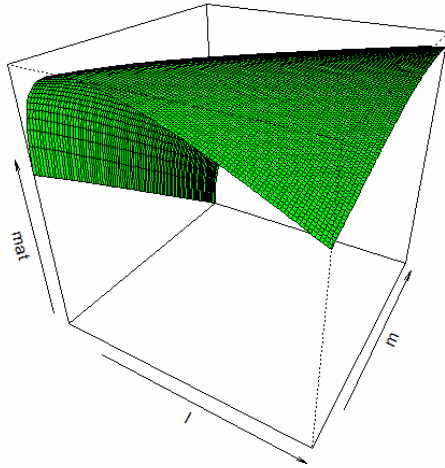
III. arbre<-createTree(l=3,m=1,spp=250,plot=T,t=2)
T<-2
t<-branching.times(arbre$tree)
mat<-matrix(numeric(10000),ncol=100)
l<-seq(0.1,10,length.out=100)
m<-1
for(i in 1:100) {
for(j in 1:100) {
mat[i,j]<-vrais(l[i], m[j], t, T)
}}
persp(l,m,mat,main="Représentation tridimensionnelle de la
vraisemblance des valeurs de l et m concernant
arbre",col="green",theta=30,phi=30)

v<-c(0,-10,-20,-30,-40,-50,-100,-200,-400,-600,-800,-1000,-
1200,-1400)
  
```

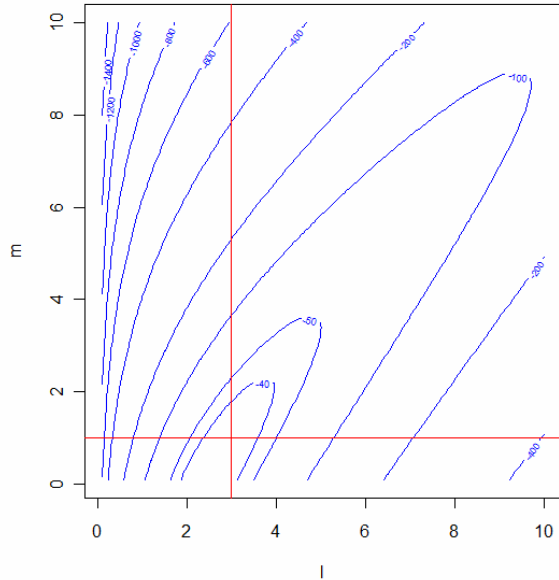
```
contour(l,m,mat,xlab="l",ylab="m",main="Représentation en relief de la vraisemblance des valeurs de l et m concernant arbre",col="blue",levels=v)
```

```
abline(h=1,v=3,col="red")
```

Représentation tridimensionnelle de la vraisemblance des valeurs de l et m concernant arbre



Représentation en relief de la vraisemblance des valeurs de l et m concernant arbre



```
IV. arbre<-createTree(l=3,m=1,spp=2500,plot=T,t=2)
```

```
T<-2
```

```
t<-branching.times(arbre$tree)
```

```
mat<-matrix(numeric(10000),ncol=100)
```

```
l<-seq(0.1,10,length.out=100)
```

```
m<-l
```

```
for(i in 1:100) {
  for(j in 1:100) {
    mat[i,j]<-vrais(l[i], m[j], t, T)
  }
}
```

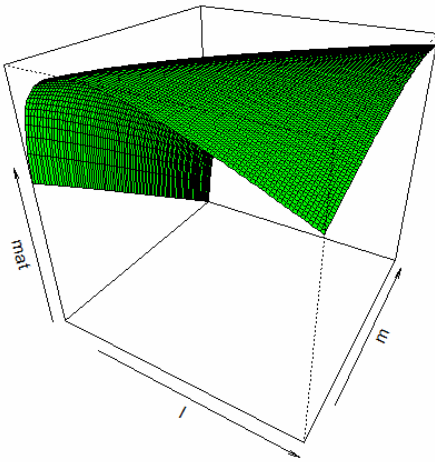
```
persp(l,m,mat,main="Représentation tridimensionnelle de la vraisemblance des valeurs de l et m concernant arbre",col="green",theta=30,phi=30)
```

```
v<-c(0,-10,-20,-30,-40,-50,-100,-200,-300,-400,-500,-600,-700,-800,-900,-1000,-2000,-4000,-8000)
```

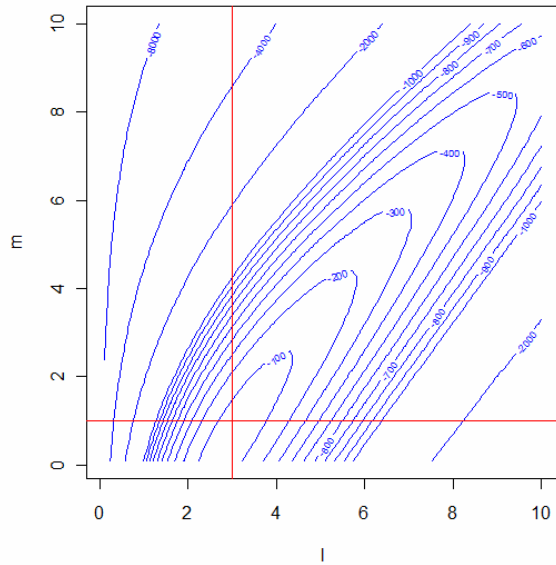
```
contour(l,m,mat,xlab="l",ylab="m",main="Représentation en relief de la vraisemblance des valeurs de l et m concernant arbre",col="blue",levels=v)
```

```
abline(h=1,v=3,col="red")
```


Représentation tridimensionnelle de la vraisemblance
des valeurs de l et m concernant arbre



Représentation en relief de la vraisemblance
des valeurs de l et m concernant arbre



En comparant les exemples 2, 3, et 4 entre lesquels le nombre d'espèces considérées est accru d'un facteur 10 à chaque fois, on peut constater que plus le nombre d'espèce est grand et plus le maximum de vraisemblance apparaît avec précision.