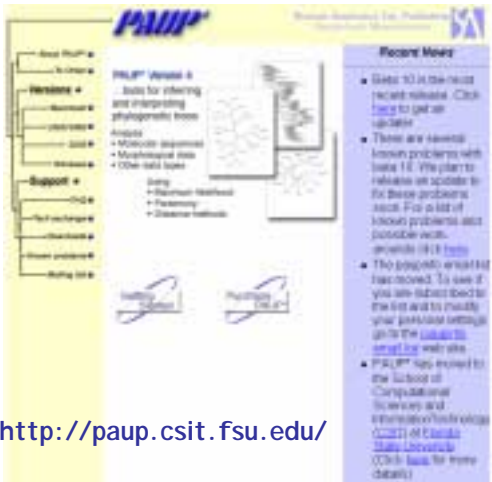


Un tutorial sobre el uso de PAUP* desde la línea de comandos
Phylogenetic analysis using parsimony and other methods



<http://paup.csit.fsu.edu/>

Un tutorial sobre el uso de PAUP* desde la línea de comandos
- Phylogenetic analysis using parsimony (and other methods) -



- PAUP* es desarrollado por David Swofford de la Univ. Estatal de Florida
- PAUP* lo comercializa Sinauer Asoc. para múltiples plataformas. Sólo la versión de Mac OS9 está carbonizada (full GUI). Las demás son de línea de comandos
- PAUP* es principalmente un programa de búsqueda de árboles bajo div. criterios de optimiz. MP, ML, ME, LS y métodos de distancia, implementando algoritmos de búsqueda exhaustivos (alltrees y BandB), así como diversos heurísticos (hsearch, NJ, UPGMA) usando diversos métodos de branch swapping (NNI, SPR, TBR). Seqs. de prots. sólo analizables bajo MP.
- Además puede inferir estados de carácter ancestrales bajo MP y ML, implementa el set completo de modelos de sustitución de DNA de la familia GTR y puede hacer una gran variedad de tests filogenéticos y contrastes de hipótesis.
- Emplea un formato propio (NEXUS) consistente en un lenguaje de scripting muy poderoso con el que se controla el comportamiento del programa a nivel de los datos, análisis y árboles.

Un tutorial sobre el uso de PAUP* desde la línea de comandos

- El formato NEXUS - taxa block

(Maddison, David R., Swofford, David L. and Maddison, Wayne P. 1997.
 NEXUS: an extensible file format for systematic information. Systematic Biology 46: 590-621)

```
#NEXUS

[!Data from: Hayasaka, K., T. Gojobori, and S. Horai. 1988. Molecular phylogeny
and evolution of primate mitochondrial DNA. Mol. Biol. Evol. 5:626-644.
]

begin taxa;
    dimensions ntax=12;
    taxlabels
    Lemur_catta
    Homo_sapiens
    Pan
    Gorilla
    Pongo
    Hylobates
    Macaca_fuscata
    M_mulatta
    M_fascicularis
    M_sylvanus
    Saimiri_sciureus
    Tarsius_syrichata
    ;
end;
```

[existen diversos (9) bloques de comandos: los que usaremos en clase son:

- 1) taxa block
- 2) characters / data block
- 3) assumptions block
- 4) PAUP block

Cada bloque comienza con begin "block"; y termina con end;

Comentarios entre corchetes son ignorados]

Un tutorial sobre el uso de PAUP* desde la línea de comandos

- El formato NEXUS - characters or data block

```
BEGIN CHARACTERS;
    [character block]
    DIMENSIONS NCHAR=100;
    FORMAT DATATYPE=DNA MISSING=? GAP=- MATCHCHAR=. INTERLEAVE=yes ;
    MATRIX
    [
        10      20      30      40      50]
    [
        *      *      *      *      *]
    R._galegae    CCGCTGGTCACCTCCGGCAAGCGCCATCCACCAGGAAGCGCCTTCCTA
    M._plurifarium  ...G.C.A.G..GT..AGCT...T.....CCG..T..GG...
    B._japonicum   ...G.CAAGT.GGAA...CT.....GA...

    [
        60      70      80      90      100]
    [
        *      *      *      *      *]
    R._galegae    CGTCGATCAGTCGACCGAAGCCAGATCCTGGTCACCGGCATCAAGTCC
    M._plurifarium T.....C.....G....CG.....T.....T.....TC
    B._japonicum   .AC...C.....C.....CTG.A..T..C.....
    ;
END;
```

- Además de DNA se pueden usar matrices binarias y de AAs para análisis bajo MP

Un tutorial sobre el uso de PAUP* desde la línea de comandos

• El formato NEXUS - assumptions block

```
begin assumptions;
  charset coding = 2-457 660-896;
  charset noncoding = 1 458-659 897-898;
  charset 1stpos = 2-457\3 660-896\3;
  charset 2ndpos = 3-457\3 661-896\3;
  charset 3rdpos = 4-457\3 662-.\3;

  usertype 2_1 = 4           [weights transversions 2 times transitions]
      a c g t
  [a] . 2 1 2
  [c] 2 . 2 1
  [g] 1 2 . 2
  [t] 2 1 2 .
      ;
  usertype 3_1 = 4           [weights transversions 3 times transitions]
      a c g t
  [a] . 3 1 3
  [c] 3 . 3 1
  [g] 1 3 . 3
  [t] 3 1 3 .
      ;

  taxset hominoids = Homo_sapiens Pan Gorilla Pongo Hylobates;

end;
```

Un tutorial sobre el uso de PAUP* desde la línea de comandos

• El formato NEXUS - paup block (es un bloque opcional; pueden darse todas las instrucciones desde la línea de comandos)

[MP analyses practice batch file: La versión completa y comentada de este archivo batch la podrán descargar de la [página del curso](#). Los números corresponden a los pasos en el análisis (el archivo original tiene muchos más pasos). Todo lo escrito entre corchetes son comentarios ignorados por PAUP* al parsear el archivo]

```
Begin paup;
  set autoclose=yes warnree=no warnreset=no;
  log start file=Paup_MP_practice.log replace;
  [1] exe primate-mtDNA.nex;
  [2] include coding/only;
  [3] delete M._mulatta M._fascicularis M._sylvanus Tarsius_syrichta;
  [4] set criterion=parsimony;
  [5] weights 2:2ndpos;
  [6] ctype 2_1:all;
  [7] cstatus;
  [8] hsearch start=stepwise addseq=random swap=spr;
  [9] outgroup Lemur_catta Macaca_fuscata Saimiri_sciureus;
  [10] showtrees all;
  [11] describtrees 1/plot=phylogram brlens=yes rootmethod=outgroup
      outroot=monophyl;
  [12] savetrees file=Primates_MP_TBR_practice.tre brlens=yes replace;
  [13] BootStrap treefile=MP_BootstrapTreeFile.tre nreps=1000 conlevel=50
      brlens=yes replace=yes format=nexus search=heuristic /addseq=random nreps=1;

quit;
```

Un tutorial sobre el uso de PAUP* desde la línea de comandos

• El formato NEXUS - paup block para NJ y LS

[Un bloque paup para reconstruir una filogenia usando **NJ** y evaluándola bajo el criterio de **LS** usando el modelo de **HKY**. El árbol resultante será descrito y presentado en forma de filograma y guardado al disco con long. de ramas. Se hace además un análisis de bootstrap con 1000 pseudoréplicas y se guarda en disco. Se guarda también un logfile "myoutput.log"]

```
Begin paup;
  log file=my_NJ_LS_boot_Analysis.log start replace;
  outgroup mi_OTU1 mi_OTU2 ...;
  set criterion=dist;
  dset distance=hky objective=ls power=2 negbrlen=prohibit;
  nj bionj=yes treefile=NJ_tree.tre brlens=yes replace=yes;
  hs start=stepwise addseq=random swap=tbr;
  describe/plot=phylo; savetrees file=my_NJ_tree.tre brlens=yes;
  bootstrap treefile=NJ_BootstrapTrees_phy.tre nreps=1000
      conlevel=50 replace=yes format=phylogram search=nj;
  savetrees file=NJboot500CnsTree.tre from=1 to=1 MaxDecimals=0
      SaveBootP=brlens replace;

  log stop;
  quit;
```

end;

Un tutorial sobre el uso de PAUP* desde la línea de comandos

• El formato NEXUS - paup block para estimar un árbol de ML bajo HKY

[Un bloque paup para reconstruir una filogenia bajo criterio de **ML** usando el modelo de **HKY** estimando ti/tv, empleando una búsqueda heurística iniciada a partir de 100 árboles de adición secuencial aleatorizada con branch swapping=TBR. El árbol resultante será descrito y presentado en forma de filograma y guardado al disco con long. de ramas. Se guarda un logfile de todo el análisis en "myoutput.log"]

```
Begin paup;
  log file=myoutput.log start replace;
  outgroup mi_OTU1 mi_OTU2 ...;
  set criterion=likelihood;
  lset nst=2 basefreq=empir rates=equal tratio=estim variant=hky;
  hsearch swap=tbr addseq=random nreps=100 start=stepwise;
  describe 1 / plot=phylogram;
  savetrees file=mytrees.tre brlens;

  log stop;
  quit;
```

end;

Un tutorial sobre el uso de PAUP* desde la línea de comandos

- Paup block para definir diversas distancias para uso bajo `criterion=distance`

Usa `dset ?` para ver todos los modelos disponibles. Los modelos con nombre (salvo custom) pueden ser usados directamente bajo `crit=dist`.

Así por ejemplo, para definir una distancia TrN93, mostrar y escribir a disco la matriz resultante y hacer una búsqueda heurística con ella bajo el criterio de mínimos cuadrados podríamos escribir los siguientes comandos en la línea de comandos de PAUP* o en un bloque de comandos PAUP, como nuestro abajo:

```
begin paup;
  set criterion=distance;
  dset distance=TamNei objective=lsfit power=2 negbrlen= prohibit;
  [me es el crit. por defecto; weighted LS, con exponente 2; evita ramas de long. negativa ]
  showdist; [despliega la matrix de distancias correspondiente, TrN93 en este caso];
  savedist file=TrNdist_myfile.nex
  hsearch;
end;
```

Un tutorial sobre el uso de PAUP* desde la línea de comandos

- Análisis de MP usando PAUP* : pasos básicos

1. Entrada de secuencias al programa

- 1.1 Start Paup * (asumiendo paup está en el camino de búsqueda: se llama con `> paup`)
- 1.2 `execute my_file.nex;`
- 1.3 alternativamente ejecutar el archivo *.nex así : `paup my_file.nex;`

2. Manejo de datos de entrada y uso de la ayuda disponible desde la línea de comandos

- 2.1 seleccionar sets de caracteres especificados en assumptions block usando `include` o `exclude`
- 2.2 se pueden definir nuevos sets de caracteres desde la línea de comandos con el comando `charset`
- 2.3 Se pueden eliminar determinados taxa con el comando `delete` o reestablecerlos con `undelete`

Un tutorial sobre el uso de PAUP* desde la línea de comandos

2. Manejo de datos de entrada y uso de la ayuda disponible desde la línea de comandos

2.4 todas los comandos y opciones de PAUP* se ven ejecutando `? ó command ?`:

```
paup> ?
ó paup> command ?
```

```
paup> ?
```

The following commands are always available:

!	Defaults	Edit	Factory	Help	Log	Quit	Time
CD	DSet	Execute	FStatus	Leave	LSet	Set	TONEXUS

The following commands require data from a DATA (or TAXA and CHARACTERS) or DISTANCES block (* = requires only TAXA block):

*Agree	Condense	Export	Jackknife	PScores	SaveAssum	*ShowTrees	*TStatus
AllTrees	*Constraints	ExSet	Lake	PSet	SaveDist	ShowUserTypes	TypeSet
AncStates	*ConTree	*Filter	*LoadConstr	Puzzle	*SaveTrees	*SortTrees	*Undelete
Assume	CStatus	GammaPlot	LScores	RandTrees	ShowAnc	StarDecomp	UPGMA
BandB	CType	*GenerateTrees	*MatrixRep	*RateSet	*ShowConstr	SufCheck	UserType
BareFregs	*Delete	*GetTrees	MPSets	Reconstruct	ShowDist	*TaxPartition	Weights
Bootstrap	*DerootTrees	HomPart	NJ	*Restore	ShowMatrix	*TaxSet	Wts
CharPartition	*DescribeTrees	HSearch	*Outgroup	RevFilter	ShowCharParts	*TreeDist	WtSet
CharSet	DScores	Include	PairDiff	Reweight	ShowRateSets	*TreeInfo	
*ClearTrees	Exclude	*Ingroup	Permute	*RootTrees	ShowTaxParts	*TreeWts	

Un tutorial sobre el uso de PAUP* desde la línea de comandos

2. Manejo de datos de entrada y uso de la ayuda disponible desde la línea de comandos

2.5 Una descripción de una línea de cada comando se obtiene con

```
paup> help cmds;
```

The following commands are always available:

!	Temporarily exit to shell
CD	Change working directory
Defaults	Set default options
DSet	Set options for distance analysis
Edit	Edit a file
Execute	Execute an input file
Factory	Reset all options to factory default settings
FStatus	Show current status of data, log, and tree files
Help	Show list of available commands
Leave	Cancel batch-processing of a file
Log	Start/stop logging of output to file
LSet	Set options for maximum likelihood analysis
Quit	Exit PAUP*
Set	Set general options
Time	Output the current time and date
TONEXUS	Convert a file to NEXUS format

...

Un tutorial sobre el uso de PAUP* desde la línea de comandos

2. Manejo de datos de entrada y uso de la ayuda disponible desde la línea de comandos

2.5 Una descripción de una línea de cada comando se obtiene con

```
paup> help cmds;
```

The following commands require data from a DATA (or TAXA and CHARACTERS) or DISTANCES block (* = requires only TAXA block):

*Agree	Calculate agreement subtrees
AllTrees	Perform exhaustive search
AncStates	Define ancestral character states
Assume	Set character assumptions
BandB	Perform branch-and-bound search
BaseFreqs	Show base frequencies for each taxon
Bootstrap	Perform bootstrap analysis
CharPartition	Define a partition of the characters
CharSet	Define a 'charset' (character set)
*ClearTrees	Empty tree buffer
Condense	Condense trees (collapse branches and/or eliminate duplicate trees)
*Constraints	Define constraint tree or specify taxon partition for convexity constraints
*ConTree	Compute consensus trees
CStatus	Show character status (types, weights, etc.)
CType	Set character types
*Delete	Delete taxa from analysis

... Y MUCHOS MÁS (ver transparencia 10)

Un tutorial sobre el uso de PAUP* desde la línea de comandos

2. Manejo de datos de entrada y uso de la ayuda disponible desde la línea de comandos

2.6 Una descripción de las opciones de cada comando se obtiene así: `paup> cmdmd ?`

```
paup> hsearch ?
```

Usage: HSearch [options...];

Available options:

Keyword	Option type	Current default setting
Swap	None NNI SPR TBR	TBR
Keep	<real-value> No	No
MultTrees	No Yes	Yes
Enforce	No Yes	No
Constraints	<constraint-name>	<none>
Converse	No Yes	No
ReconLimit	<integer-value> Infinity	Infinity
NChuck	<integer-value>	0
ChuckScore	<real-value> No	No
Start	Stepwise NJ Current <tree-number>[-<tree-number>]	*Stepwise
AddSeq	Simple Closest AsIs Random Furthest	Simple
NReps	<integer-value>	10
RSeed	<integer-value>	0
RStatus	No Yes	No
SaveReps	No Yes	No
RefTax	<integer-value>	0

Un tutorial sobre el uso de PAUP* desde la línea de comandos

3. Selección de un criterio de optimización y definición de supuestos

3.1 Selección explícita de un criterio (MP es el que usa PAUP* por defecto)

```
set criterion=parsimony; (set criterion=distance; set criterion=likelihood;)
```

3.2 Considera usar ponderación de caracteres en un contexto de MP

```
weights 2:2ndpos; otorga un peso de 2 a todas las 2as. pos. de codones definidas en el "assumptions block"
```

3.3 Considera usar una matriz de pasos o ponderación (Parsimonia generalizada)

```
ctype 3_1:all; usa una matriz de ponderación que da x3 más peso a tv que ti, definida en el "assumptions block"
```

3.4 Antes de iniciar el análisis conviene checar el estatus de definición (supuestos) el análisis de caracteres

```
cstatus; imprime las asignaciones de pesos a los distintos caracteres seleccionados para el análisis
```

Un tutorial sobre el uso de PAUP* desde la línea de comandos

4. Búsqueda y evaluación de árboles (reparar tema 6: MP y búsqueda de árboles)

4.1 Selección de una estrategia de búsqueda: **exactas** (`alltrees` y `BandB`) vs. **heurísticas** (`hsearch`, `NJ`, `UPGMA`)

4.2 Selección de opciones de búsqueda heurística: generalmente nos concentraremos en cuatro de las opciones disponibles (ver con `hs ?`) bajo `hsearch`:

- `start=stepwise|nj|current|1-89` (un rango de árboles en memoria o en file)
- `addseq=random|AsIs|furthest|simple`
- `swap=nni|spr|tbr`
- `nrep = no. de repeticiones` o puntos desde los que iniciar las búsquedas

4.3 Inicio de una búsqueda heurística mediante múltiples árboles semilla generados mediante adición secuencial aleatorizada x100 y swapping mediante TBR.

```
hsearch start=stepwise addseq=random nrep=100 swap=tbr;
```

4.4 Análisis de bootstrap

```
bootstrap nrep=500 conlevel=50 treefile=hsboot500.tre format=phyml replace search=heuristic /addseq=random nrep=1;
```

Un tutorial sobre el uso de PAUP* desde la línea de comandos

5. Impresión y salvado de árboles

5.1 Definición de secuencias outgroup

`outgroup tax1 tax2 tax3; ó outgroup 4 5 6;`

5.2 Despliegue en pantalla de cladogramas

`showtrees all;` ó `showtrees tree-list /options;` (ver opciones bajo showtr ?)

5.3 Despliegue en pantalla de filogramas e información detallada sobre árboles y caracteres con `describetrees`

`describetrees /plot=phylogram brlens=yes rootmethod=outgroup outroot=monophyl;`

5.4 Salvado de árboles en diversos formatos con `savetrees`

`savetrees file=parsTree.tre format=nexus|phylip brlens=yes;`

Un tutorial sobre el uso de PAUP* desde la línea de comandos

6. Una miscelánea de otros comandos muy útiles

6.1 Conversión de formatos foráneos de secs. alineadas a formato NEXUS con `tonexus`.

`tonexus fromfile=myfile.phy format=phylip datatype=protein tofile=myfile.nex;`
(checa `tonexus ?` para ver más opciones y formatos)

6.2 Inclusión y exclusión de sitios estándar no definidos en un "assumptions block"

usando las siguientes palabras clave junto a los comandos `include` y `exclude` :

`all`, `gapped`, `missambig`, `constant` y `uninf`

como por ejemplo: `exclude gapped;` ó `include all;`

6.3 Salir al shell desde la terminal en la que está corriendo PAUP* es fácil:

usa el comando `!`. Por ejemplo: `paup>!pwd` (para regresar a `paup>ctrl-C`)

6.4 Para reestablecer todos los settings de defecto de PAUP* usar `factory`;

6.5 Y para salir de PAUP* el comando es `quit`;

Un tutorial sobre el uso de PAUP* desde la línea de comandos

7. Ejercicio: análisis bajo MP LS y NJ del set clásico de datos de mtDNA de primates. ¿Cuál es la topología correcta entre homo, pan, gorilla y pongo con respecto a otros primates?



Pan troglodytes
(chimpancé)



Homo sapiens
(humano)



Gorilla gorilla
(gorila)



Pongo pygmaeus
(orangután)



Lemur catta
(lemur de cola anillada)



Maccaca fuscata
(macaco japonés)



Samiri sciureus
(mono ardilla)



Hylobates lar
(gibon)



Tarsius syrichta
(tarsero de Filipinas)

Un tutorial sobre el uso de PAUP* desde la línea de comandos

7. Ejercicio: análisis bajo MP LS y NJ del set clásico de datos de mtDNA de primates. ¿Cuál es la topología correcta entre homo, pan y gorilla?

7.1 Ve a la carpeta PAUP de la página del curso y descarga el archivo `mtDNA-primates.nex` y examina su estructura. ¿qué quieren decir los distintos comandos?

7.2 ejecuta el archivo `mtDNA-primates.nex`

excluye a las secs `M._mulatta` `M._fascicularis` `M._sylvanus` y `Tarsius_syrichta`;
especifica como grupo externo a `Lemur_catta` `Macaca_fuscata` `Saimiri_sciureus`;
incluye en el análisis sólo a las posiciones codificantes;

define el criterio de MP como criterio de optimización;

haz una búsqueda heurística partiendo de 500 árboles de adición secuencial aleatorizada;
describe el/los árboles y graficalos en forma de phylograma con longitudes de rama;

¿qué tipo de análisis has corrido? ¿qué ha evidenciado este análisis?

haz un análisis de bootstrap (1000 pseudoréplicas) usando una búsqueda hs;

¿qué nos dice este análisis sobre las relaciones entre homo, pan y gorilla?

Un tutorial sobre el uso de PAUP* desde la línea de comandos

7. Ejercicio: análisis bajo MP LS y NJ del set clásico de datos de mtDNA de primates. ¿Cuál es la topología correcta entre homo, pan y gorilla?

7.3 Bajo un contexto de MP y usando el assumption block, ¿qué se te ocurre hacer? prueba a usar los comandos `weights` y `ctype` en una nueva hs y bootstrap analysis; ¿qué hacen estos comandos y qué efecto tienen sobre la topología?

7.4 Cambia al criterio de distancia y haz un análisis de frecuencias de bases con `basefreq`;

- En base a este resultado, qué modelo de sustitución escogerías de estos tres: ¿JC, K2P o HKY?
- Estima un árbol de NJ bajo el modelo que creas más adecuado y haz un análisis de bootstrap con 1000 réplicas. ¿Qué topología es la favorecida?
- Estima un árbol bajo el modelo que creas más adecuado usando el criterio de ME o LS haciendo una hs. Haz un análisis de bootstrap.

7.5 En base a toda la evidencia acumulada, cuál crees que es la topología favorecida?

Un tutorial sobre el uso de PAUP* desde la línea de comandos

• Paup block para definir diversas distancias para uso bajo `criterion=likelihood`

Para definir un modelo en PAUP* bajo el criterio de máxima verosimilitud hemos de indicarle al programa las características del mismo en términos de parámetros de frecuencia, tasa y heterogeneidad de tasas de sustitución entre sitio.

Para ello hemos de explorar con `Iset` ? las posibilidades.

NO PODEMOS ESPECIFICARLE SIMPLEMENTE UN NOMBRE DE MODELO COMO JC.

La salida del programa, una vez corrido el análisis, confirmará si definimos el modelo adecuadamente.

```
[Especificación del modelo de JC69 bajo ML: un solo param de tasa e igual frec. de bases ]
begin paup;
  set criterion=likelihood; [yo escribiría más sucintamente: set crit=like; ]
  Iset nst=1 basefreq=equal; [nst es el no. de tipos o tasas de sust; basefreq= iguales ]
end;
```

```
[Especificación del modelo de F81 bajo ML: un solo param de tasa distintas frec. de bases ]
begin paup;
  set criterion=likelihood; [yo escribiría más sucintamente: set crit=like; ]
  Iset nst=1 basefreq=empirical; [nst es el no. de tipos o tasas de sust; basefreq= empíricas ]
end;
```

Un tutorial sobre el uso de PAUP* desde la línea de comandos

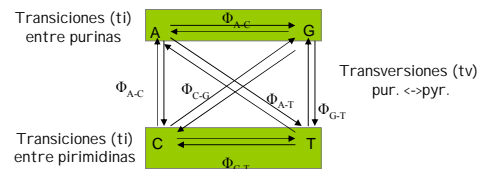
• Paup block para definir diversas distancias para uso bajo `criterion=likelihood`

```
[Especificación del modelo de K2P ó K80 bajo ML: dos params de tasa e igual frec. de bases ]
begin paup;
  set criterion=likelihood;
  Iset nst=2 basefreq=equal tratio=estimate; [estimamos el valor de ML del parámetro ti/tv ]
end;
```

```
[Especificación del modelo de HKY85 bajo ML: dos params de tasa y distintas frec. de bases ]
begin paup;
  set criterion=likelihood;
  Iset nst=2 basefreq=estimate tratio=estimate variant=hky; [ estimamos basefreq y ti/tv ratio ]
end;
```

```
[Especificación del modelo de TrN93 bajo ML: tres params de tasa (2ti, 1tv) y distintas frec. de bases ]
begin paup;
  set criterion=likelihood;
  Iset nst=6 rclass=(abaaca) rmatrix=estimate basefreq=estimate;
  [ estimamos basefreq y las tasas para las dos clases de ti y una tasa para las 4 tv ]
end;
```

[Rate classes:
a = AC
b = AG
c = AT
d = CG
e = CT
f = GT = 1]



Un tutorial sobre el uso de PAUP* desde la línea de comandos

• Paup block para definir diversas distancias para uso bajo `criterion=likelihood`

```
[Especificación del modelo de GTR+I+G bajo ML: seis params de tasa y distintas frec. de bases + proporción de sitios invariantes y parámetro alfa de la distr. gamma con 6 categorías; recuerda, GTR+I+G es el modelo más complejo de la familia GTR, con 11 parámetros libres]
```

```
begin paup;
  set criterion=likelihood;
  Iset nst=6 basefreq=estimate rmat=estimate rate=gamma shape=est ncat=6 pinv=est;
  [ estimamos basefreq, las 5 tasas libres de sustitución (a-e; f=1), parámetro alfa de la distribución gamma y proporción de sitios invariantes ]
end;
```

• Para hacer la estima del valor de ML de los parámetros de un modelo de sustitución necesitamos una topología de base o de partida, sobre la que estimar dichos parámetros.

• Recuerda que: $L(H|D) = \Pr(D|H) = \Pr(D|\nu\phi)$ Por lo tanto, para estimar ϕ , (valores de los param. del modelo de sust.) suele estimarse primero una topología de NJ o Pars o EM, las cuales representan un buen árbol de partida (muchísimo mejor que un árbol aleatorio o de adición secuencial). Fijadas la ν , es decir, topología y longitudes de rama, el cálculo de los valores de los parámetros, dada dicha combinación de ν , es relativamente rápida.

Un tutorial sobre el uso de PAUP* desde la línea de comandos

- Paup block para estimación de valores de parámetros bajo modelo GTR+I+G partiendo de un árbol HKY85-NJ

```
begin paup;
set criterion=likelihood;
lset nst=6 basefreq=estimate rmat=estimate rate=gamma shape=est ncat=6 pinv=est;

nj; [reconstrucción neighbor-joining; queda almacenado en memoria un árbol HKY-NJ,
haz un dset ? para ver las opciones por defecto ]

Scores 1; [ fuerza a PAUP* a obtener estimas de ML para basefreq, parámetro de tasa,
del parám. alfa de la distribución gamma y de proporción de sitios invariables
en base al primer árbol en memoria. Nótese que tenemos sólo 1 si hacemos un NJ]

lset basefreq=previous rmat=prev shape=prev pinv=prev;
[ indicamos a PAUP* usar los valores de ML estimados para cada parámetro previamente]

hsearch start=stepwise addseq=random nrep=10 swap=tbr;
[ hacemos una búsqueda heurística de una topología y long. de ramas bajo el criterio de ML,
usando los valores de ML estimados para cada parámetro previamente]
end;
```

Un tutorial sobre el uso de PAUP* desde la línea de comandos

- 8. Uso de PAUP* bajo ML para seleccionar el modelo más ajustado y el valor de máxima verosimilitud de los parámetros correspondientes

```
[1] exe primate-mtDNA.nex;
[2] include coding/only;
[3] delete M._mulatta M._fascicularis M._sylvanus Tarsius_syrichtha;
[4] outgroup Lemur_catta Macaca_fuscata Saimiri_sciureus;
[5] nj;
[6] set criterion=like;
[7] lset nst=1 base=eq; [settings para JC69]
[8] LScores; [nos va a dar el score global de ML del modelo JC69 usando
el árbol NJ en memoria para la estimación de parámetros]
[9] lset nst=1 base=emp; [settings para F81 usando estimación empírica de
frec. de bases. Usar estimate si queremos
estimación bajo ML de parámetros de frec.]
[10] LScores; [score global de ML bajo el modelo F81]
[11] lset nst=2 base=emp; LSc; [settings para HKY85 y eval. del modelo]
[12] lset nst=2 base=emp rates=gamma shape=est; LSc; [settings para
HKY85+G; evaluación del modelo]
[13] lset nst=2 base=emp rates=gamma shape=est pinv=est; LSc;
[settings para HKY85+I+G; evaluación del modelo]
```

Un tutorial sobre el uso de PAUP* desde la línea de comandos

- 9. Uso de PAUP* bajo ML para evaluar la hipótesis de reloj molecular bajo el modelo HKY+I+G

```
[1] exe primate-mtDNA.nex;
[2] include coding/only;
[3] delete M._mulatta M._fascicularis M._sylvanus Tarsius_syrichtha;
[4] outgroup Lemur_catta Macaca_fuscata Saimiri_sciureus;
[5] nj;
[6] set criterion=like;
[7] lset nst=2 base=emp rates=gamma shape=est pinv=est; LSc;
[settings para HKY85+I+G; evaluación del modelo]
[8] RootTrees; [enraza el árbol NJ que tenemos en memoria; necesario!!!]
[9] lset nst=2 base=emp rates=gamma shape=est pinv=est clock=yes; LSc;
[settings para HKY85+I+G bajo modelo de reloj global; evaluación
del modelo]
```