

# MCB 5472

## Types of Selection

*Peter Gogarten*

Office: *BSP 404*

phone: *860 486-4061,*

Email: [gogarten@uconn.edu](mailto:gogarten@uconn.edu)

# Very old exercises:

Write a program that it uses hashes to calculates mono-, di-, tri-, and quartet-nucleotide frequencies in a genome.

Go over tetraA.pl

# Old exercises:

modify tetraA.pl so that the user (or another program) can assign the size of the nmer as a variable!

Input size on nmer

```
print "\ngive the size of the nmer whose frequencies you want to calculate\n";  
chomp ($n=<>);  
print "\n$n mers are tabulated\n";
```

One possible solution is to add a loop as follows:

```
#####  
#count nmer frequencies  
  
for($i = 0; $i < $count-($n-1); $i++){ #big loop going through genome  
    for($k = 0; $k < $n; $k++){ #loop to assemble nmer starting in position $i  
        $subseq .= $bases[$i+$k];  
    }  
    $nmer{$subseq}++;  
    #Make rev complement of subseq:  
    $subseq=~ tr/ATGCYRatcgyr/TACGRYTAGCRY/;  
    $subseq= reverse $subseq;  
    $nmer{$subseq}++;  
}  
#####  
#output table
```

For large nmers this is inefficient. Rather than concatenating the nmer from beginning to end every time, it would be faster to assemble the nmer only once, delete the first base and add the next to the end.

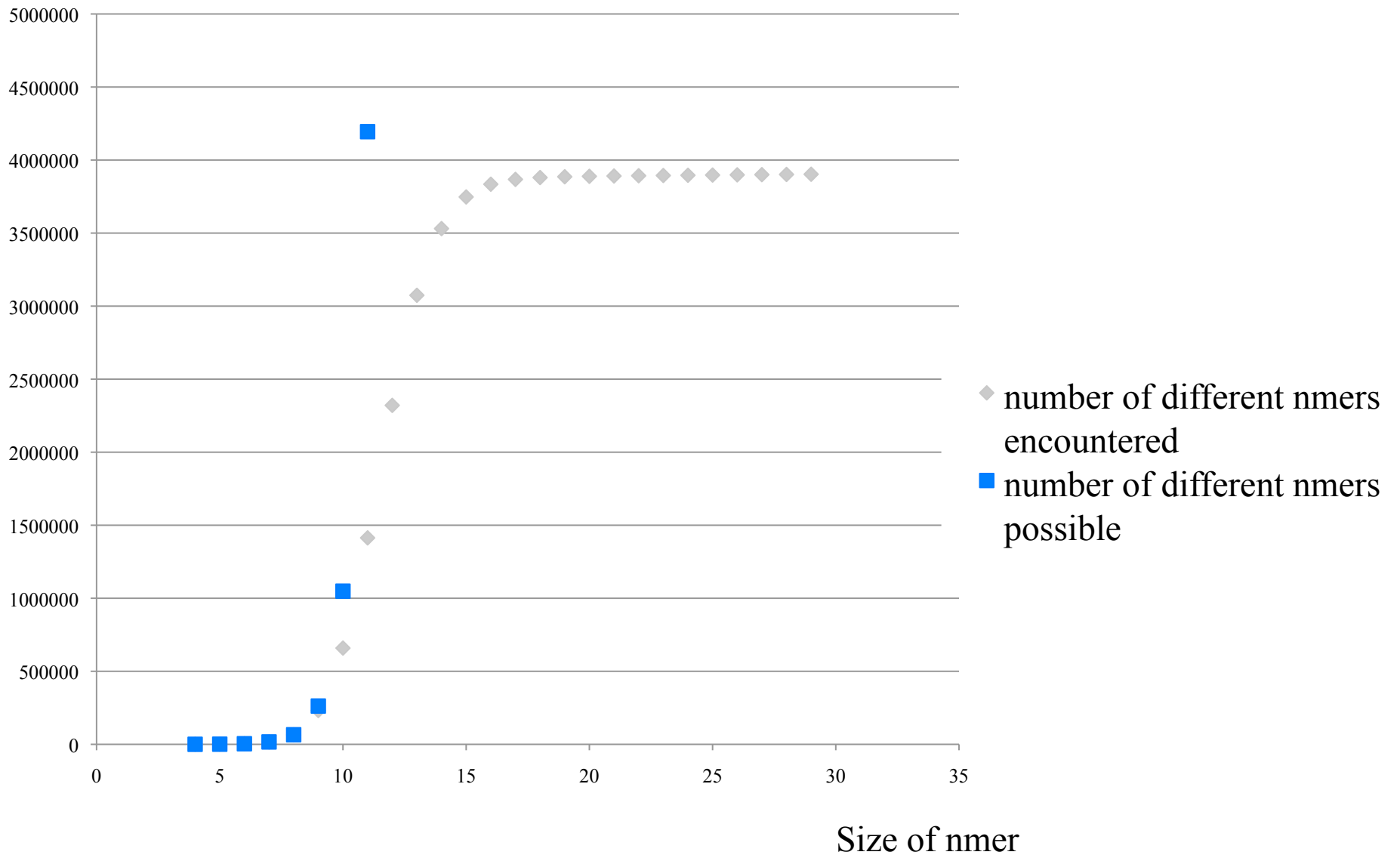
To run many nmer sizes, one could add another loop (check nmer\_many.pl):

```
#####
#count nmer frequencies

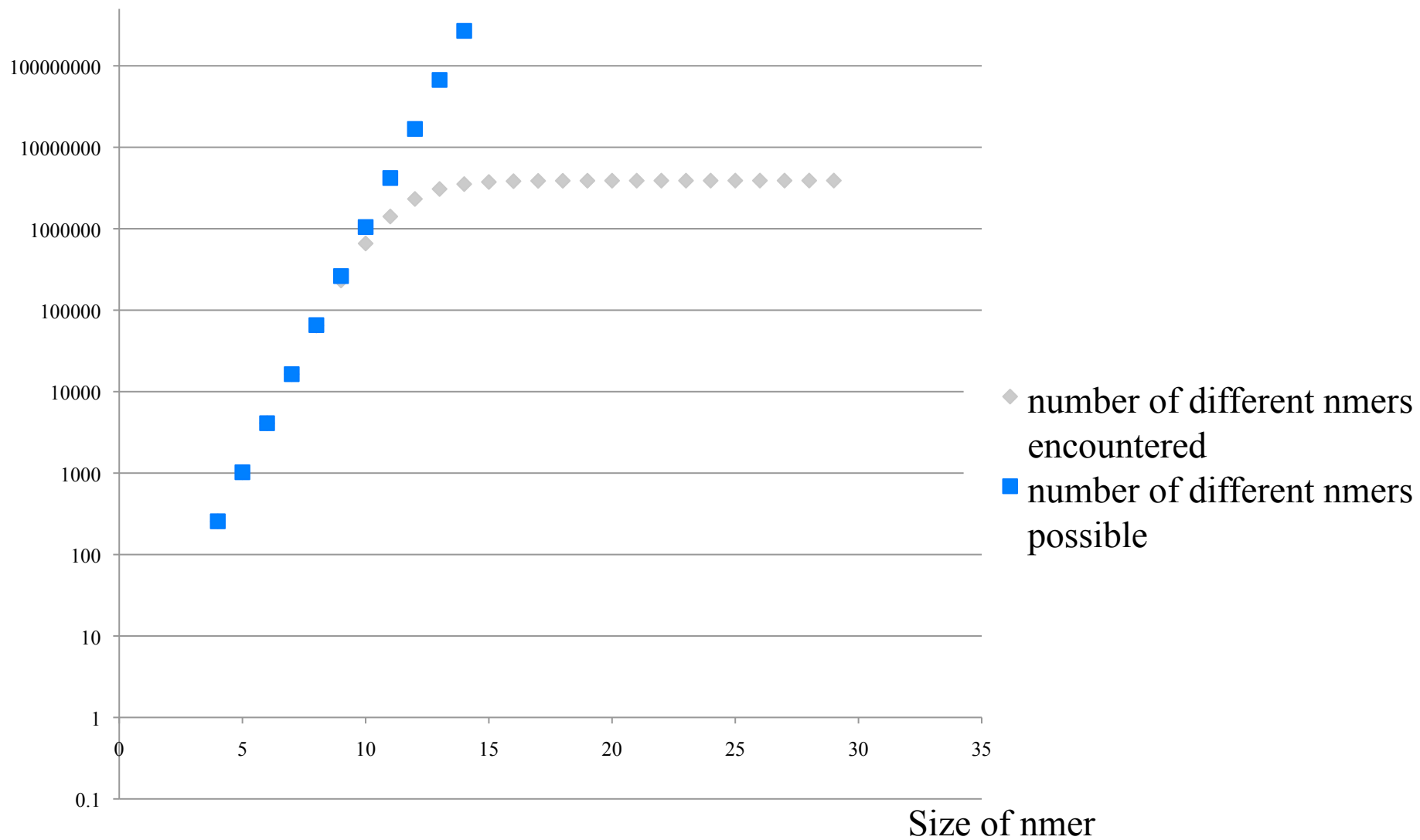
for($n = 4 ; $n < 30; $n++) ← Goes through nmers

{print "\n$n mers are tabulated\n";
  foreach (@sorted_by_value){delete $nmer{$_}}; #reset %nmer Should work too
  @sorted_by_value=(); ← Important to reset values from last loop
  for($i = 0; $i < $count-($n-1); $i++){
    $subseq=""; #reset subseq
    for($k = 0; $k < $n; $k++){
      $subseq .= $bases[$i+$k];
    }
    # print "$subseq\n";#left over from error checking
    $nmer{$subseq}++;
    #Make rev complement of subseq:
    $subseq=~ tr/ATGCYRatcgyr/TACGRYTAGCRY/;
    $subseq= reverse $subseq;
    $nmer{$subseq}++;
  }
#####
#output table
@sorted_by_value = sort { $nmer{$a} <=> $nmer{$b} or $a cmp $b} keys %nmer;
$nmercount=@sorted_by_value;
foreach (@sorted_by_value) {
  # print "$n mer \t $_ \t occurred $nmer{$_} times\n";
  # print OUT "$_\t$nmer{$_}\n" ;
}
$maxcount=4**$n;
$maxcountp=100*$nmercount/$maxcount;
print"$nmercount different $n mers were encountered, out of $maxcount possible\n this is $maxcountp % of the maximum possible\n";

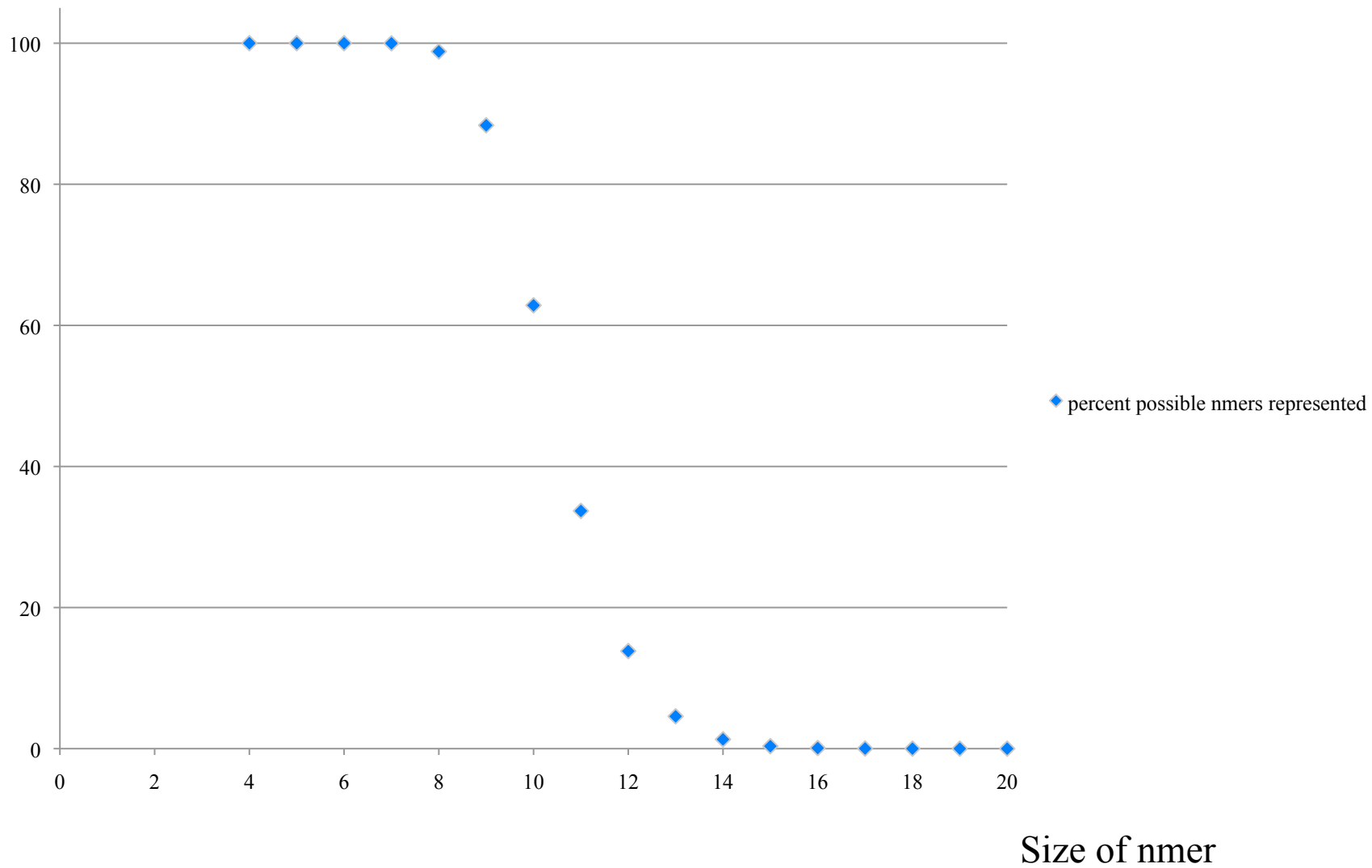
open (OUT2, ">>my_summary_table2" ) or die "cannot open my_table";
print OUT2 "$n\t$nmercount\t$maxcount\t$maxcountp\n";
close (OUT2);
} ← Nmer loop ends here
close (OUT);
```



Same with logarithmic y axis



# percent possible nmers represented



## Very Old Assignments:

- Re-read chapter P16-P18 in the primer
- Given a multiple fasta sequence file\*, write a script that for each sequence extract the gi number and the species name. and rewrites the file so that the annotation line starts with the gi number, followed by the species/strain name, followed by a space. (The gi number and the species name should not be separated by or contain any spaces – replace them by \_. This is useful, because clustalw will recognize the number and name as handle for the sequence.)
- Work on your student project
- Assume that the annotation line follows the NCBI convention and begins with the > followed by the gi number, and ends with the species and strain designation given in []

Example:

```
>gi|229240723|ref|ZP_04365119.1| primary replicative DNA  
helicase; intein [Cellulomonas flavigena DSM 20109]
```

Example multiple sequence file is [here](#).



an error prone solution is at [convertannotationline.pl](#)

## Old Assignment:

Rewrite this script so that it uses the `$&` variable to extract the gi number and the species name.

Symbol	Meaning
.	any character
\w	alphanumeric and _
\W	any non-word character
\s	any whitespace
\S	any non-whitespace
\d	any digit character
\D	any non-digit character
\t	tab
\n	newline
*	match 0 or more times
+	match 1 or more times
?	match 1 or 0 times
{n}	match exactly n times
{n,m}	match n to m times
^	match from start
\$	match to end

```

#!/usr/bin/perl
unless(@ARGV==1) {die "please provide file name in command line \n
file should contain multiple sequences in fasta format \n\n";}
$filename=$ARGV[0];
open(IN, "< $filename") or die "cannot open $filename:$!";
$outfile=$filename.".giSpec";
open(OUT, "> $outfile") or die "cannot open $outfile:$!";

while(<IN>){
    $line = $_;
    if($line =~ m/^>/){ #find annotation line
        if ($line =~ m/gi\\|\\d*/) # find gi number
            {$gi=$& ; # assign match to $gi
            $gi =~ s/gi\\|//g; #sub gil with nothing
            # $gi =~ s/\\|//g;} #sub | with nothing no longer needed reg ex does not include second|
            else {$nogi++;
            $gi="noGInumber".$nogi}; #in case no match to gi\\|\\d* found
        if ($line =~ m/^[.*/]){ #look for species/strain name
            $name = $&; # assign match to $name
            $name =~ s/^[//g; #sub [ with nothing - note \ before [ in Reg Ex
            $name =~ s/\\//g} #sub ] with nothing
            else {$name="NoNameFound"}; #report that no name was included
        $id="$gi_"."$name";
        $id =~ tr/ /_/;
        chomp($id);
        print OUT ">$id\n";
    }else{
        print OUT $line;
    }
}
}

```

Check convert2.pl

# the gradualist point of view

**Evolution occurs within populations where the fittest organisms have a selective advantage. Over time the advantages genes become fixed in a population and the population gradually changes.**

**Note: this is not in contradiction to the the theory of neutral evolution. (which says what ?)**

## **Processes that MIGHT go beyond inheritance with variation and selection?**

- Horizontal gene transfer and recombination
- Polyploidization (botany, vertebrate evolution) see [here](#)
- Fusion and cooperation of organisms (Kefir, lichen, also the eukaryotic cell)
- Targeted mutations (?), genetic memory (?) (see [Foster's](#) and [Hall's](#) reviews on directed/adaptive mutations; see [here](#) for a counterpoint)
- Random genetic drift
- [Gratuitous complexity](#)
- Selfish genes (who/what is the subject of evolution??)
- Parasitism, altruism, [Morons](#)

# selection versus drift

see Kent Holsinger's java simulations at

<http://darwin.eeb.uconn.edu/simulations/simulations.html>

The law of the gutter.

compare drift versus select + drift

The larger the population the longer it takes for an allele to become fixed.

**Note:** Even though an allele conveys a strong selective advantage of 10%, the allele has a rather large chance to go extinct.

**Note#2:** Fixation is faster under selection than under drift.

**BUT**

$$s=0$$

Probability of fixation,  $P$ , is equal to frequency of allele in population.

Mutation rate (per gene/per unit of time) =  $u$  ;

freq. with which allele is generated in diploid population size  $N = u * 2N$

Probability of fixation for each allele =  $1/(2N)$

**Substitution rate** =

frequency with which new alleles are generated \* Probability of fixation =

$$u * 2N * 1/(2N) = u$$

Therefore:

If  $s=0$ , the substitution rate is independent of population size, and equal to the mutation rate !!!! (NOTE: Mutation unequal Substitution! )

This is the reason that there is hope that the molecular clock might sometimes work.

**Fixation time due to drift alone:**

$$t_{av} = 4 * N_e \text{ generations}$$

( $N_e$  = effective population size; For  $n$  discrete generations

$$N_e = n / (1/N_1 + 1/N_2 + \dots + 1/N_n)$$

$$s > 0$$

Time till fixation on average:

$$t_{\text{av}} = (2/s) \ln(2N) \text{ generations}$$

(also true for mutations with negative “s” ! discuss among yourselves)

E.g.:  $N=10^6$ ,

$s=0$ : average time to fixation:  $4 \cdot 10^6$  generations

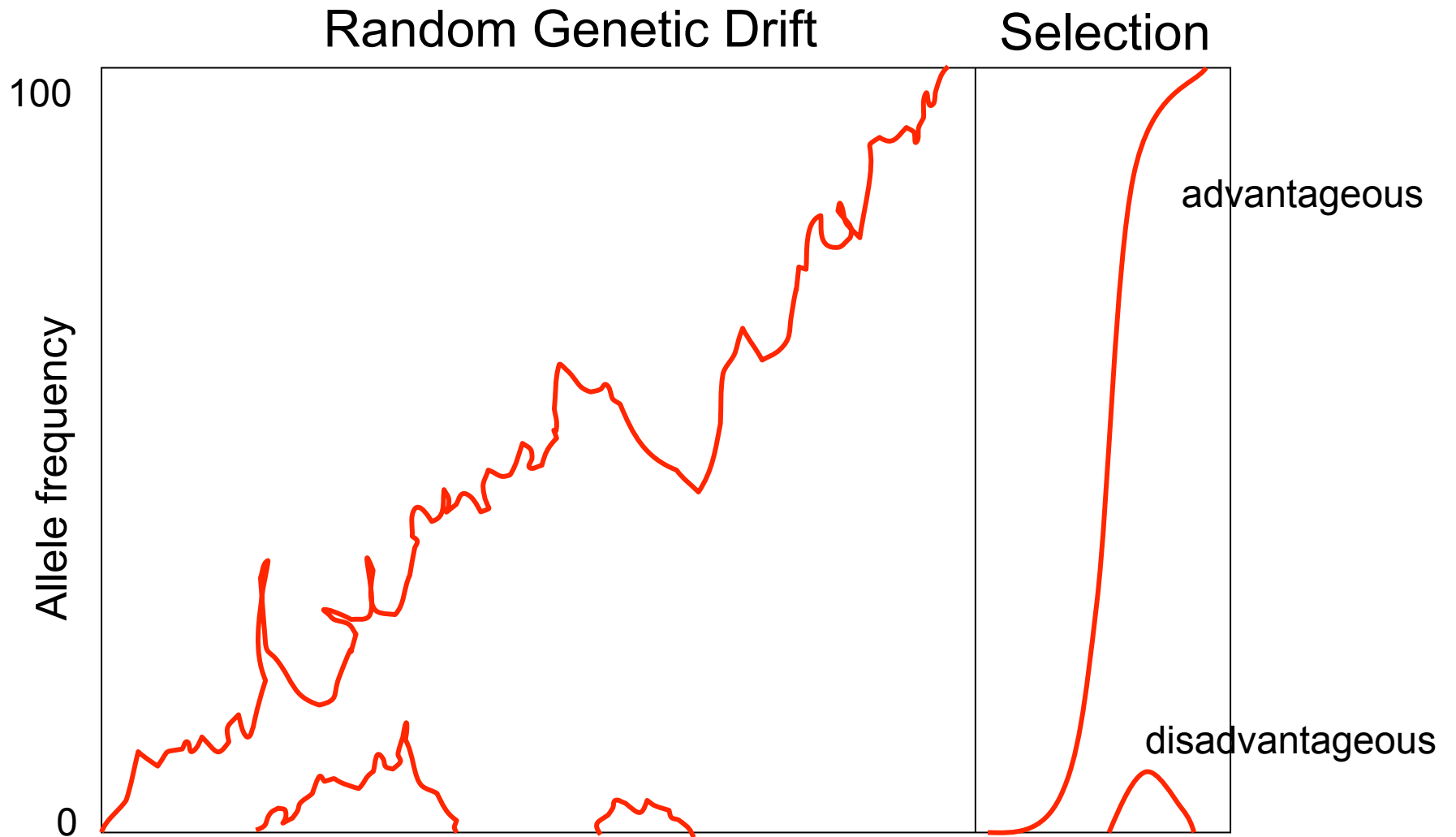
$s=0.01$ : average time to fixation: 2900 generations

$N=10^4$ ,

$s=0$ : average time to fixation: 40.000 generations

$s=0.01$ : average time to fixation: 1.900 generations

**=> substitution rate of mutation under positive selection is larger than the rate with which neutral mutations are fixed.**



Modified from from [www.tcd.ie/Genetics/staff/Aoife/GE3026/GE3026\\_1+2.ppt](http://www.tcd.ie/Genetics/staff/Aoife/GE3026/GE3026_1+2.ppt)

# Positive selection

- A new allele (mutant) confers some increase in the **fitness** of the organism
- Selection acts to favour this allele
- Also called adaptive selection or Darwinian selection.

NOTE: **Fitness** = ability to survive and reproduce



# Advantageous allele

Herbicide resistance gene in nightshade plant

*Solanum nigrum* (nightshade) psbA gene:

Normal sequence:

... R L I F Q Y A **S** F N N S  
... CGA TTG ATC TTC CAA TAT GCT **AGT** TTC AAC AAC TCT ...

Serine

Atrazine-resistant mutant:

... CGA TTG ATC TTC CAA TAT GCT **GGT** TTC AAC AAC TCT ...  
... R L I F Q Y A **G** F N N S

Glycine

Modified from [www.tcd.ie/Genetics/staff/Aoife/GE3026/GE3026\\_1+2.ppt](http://www.tcd.ie/Genetics/staff/Aoife/GE3026/GE3026_1+2.ppt)

# Negative selection

- A new allele (mutant) confers some decrease in the fitness of the organism
- Selection acts to remove this allele
- Also called purifying selection

# Deleterious allele

Human breast cancer gene, BRCA2

5% of breast cancer cases are familial

Mutations in BRCA2 account for 20% of familial cases

## Normal (wild type) allele

```
2780      2790      2800      2810      2820      2830      2840      2850      2860      2870      2880      2890      2900
fhrMetValLeuTyrGlyAspThrGlyAspLysGlnAlaThrGlnValSerIleLysLysAspLeuValTyrValLeuAlaGluGluAsnLysAsnSerValLysGlnHisIleLysMetThrLeu
ACCATGGTTTTATATGGAGACACAGGTGAT---AAGCAACCCAAGTGTCAATTAATAAAGATTTGGTTTATGTTCTTGCAGAGGAGAACAAAAATAGTGTAAAGCAGCATATAAAAATGACTCTC . . . .
```

```
ACCATGGTTTTATATGGAGACACAGGTGAT---AAGCAACCCAAGTGTCAATTAATAAAGATTTGGTTTATGTTCTTGCAGAGGAGAACAAAAATAGTGTAAAGCAGCATATAAAAATGACTCTC
fhrMetValLeuTyrGlyAspThrGlyAsp  LysGlnProLysCysGlnLeuLysLysIleTrpPheMetPheLeuGlnArgArgThrLysIleVal *
```

**Mutant allele  
(Montreal 440  
Family)**

Stop codon

4 base pair deletion  
Causes frameshift

Modified from from [www.tcd.ie/Genetics/staff/Aoife/GE3026/GE3026\\_1+2.ppt](http://www.tcd.ie/Genetics/staff/Aoife/GE3026/GE3026_1+2.ppt)

# Neutral mutations

- Neither advantageous nor disadvantageous
- Invisible to selection (no selection)
- Frequency subject to ‘drift’ in the population
- **Random drift** – random changes in small populations

# Types of Mutation-Substitution

- Replacement of one nucleotide by another
- Synonymous (Doesn't change amino acid)
  - Rate sometimes indicated by  $K_s$
  - Rate sometimes indicated by  $d_s$
- Non-Synonymous (Changes Amino Acid)
  - Rate sometimes indicated by  $K_a$
  - Rate sometimes indicated by  $d_n$

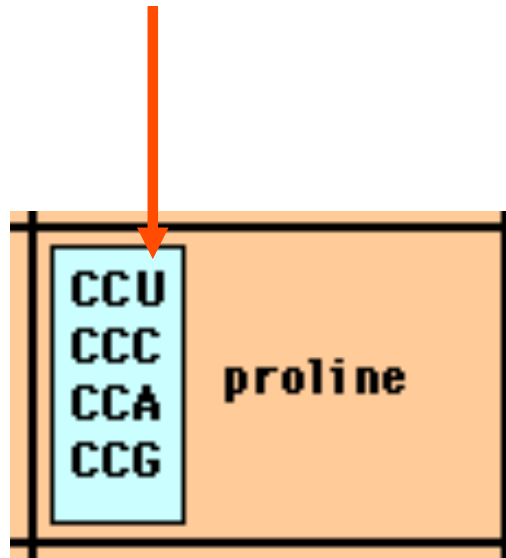
(this and the following 4 slides are from

[mentor.lscf.ucsb.edu/course/spring/eemb102/lecture/Lecture7.ppt](http://mentor.lscf.ucsb.edu/course/spring/eemb102/lecture/Lecture7.ppt))

# Genetic Code – Note degeneracy of 1<sup>st</sup> vs 2<sup>nd</sup> vs 3<sup>rd</sup> position sites

<p>UUU phenylalanine UUC alanine</p> <p>UUA leucine UUG leucine</p>	<p>UCU serine UCC serine UCA serine UCG serine</p>	<p>UAU tyrosine UAC tyrosine</p> <p>UAA stop UAG stop</p>	<p>UGU cysteine UGC cysteine</p> <p>UGA stop</p> <p>UGG tryptophan</p>
<p>CUU leucine CUC leucine CUA leucine CUG leucine</p>	<p>CCU proline CCC proline CCA proline CCG proline</p>	<p>CAU histidine CAC histidine</p> <p>CAA glutamine CAG glutamine</p>	<p>CGU arginine CGC arginine CGA arginine CGG arginine</p>
<p>AUU isoleucine AUC isoleucine AUA isoleucine</p> <p>AUG methionine</p>	<p>ACU threonine ACC threonine ACA threonine ACG threonine</p>	<p>AAU asparagine AAC asparagine</p> <p>AAA lysine AAG lysine</p>	<p>AGU serine AGC serine</p> <p>AGA arginine AGG arginine</p>
<p>GUU valine GUC valine GUA valine GUG valine</p>	<p>GCU alanine GCC alanine GCA alanine GCG alanine</p>	<p>GAU aspartic acid GAC aspartic acid</p> <p>GAA glutamic acid GAG glutamic acid</p>	<p>GGU glycine GGC glycine GGA glycine GGG glycine</p>

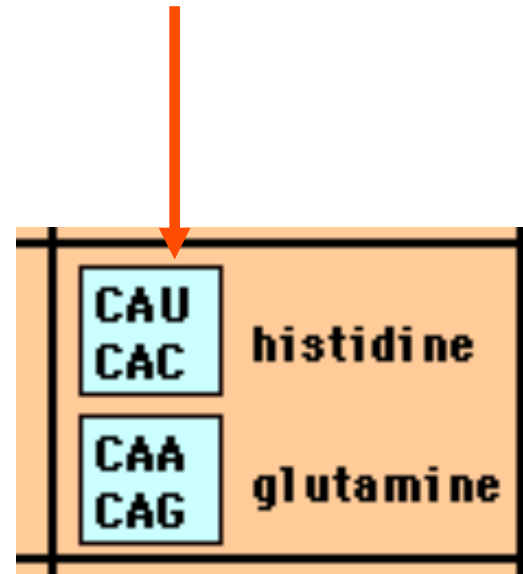
# Genetic Code



*Four-fold degenerate site* – Any substitution is synonymous

From: [mentor.lscf.ucsb.edu/course/spring/eemb102/lecture/Lecture7.ppt](http://mentor.lscf.ucsb.edu/course/spring/eemb102/lecture/Lecture7.ppt)

# Genetic Code



*Two-fold degenerate site* – Some substitutions synonymous, some non-synonymous

From: [mentor.lscf.ucsb.edu/course/spring/eemb102/lecture/Lecture7.ppt](http://mentor.lscf.ucsb.edu/course/spring/eemb102/lecture/Lecture7.ppt)



# Measuring Selection on Genes

- Null hypothesis = neutral evolution
- Under neutral evolution, synonymous changes should accumulate at a rate equal to mutation rate
- Under neutral evolution, amino acid substitutions should also accumulate at a rate equal to the mutation rate

# Counting #s/#a

Species1	Ser TGA	Ser TGC	Ser TGT	Ser TGT	Ser TGT
Species2	Ser TGT	Ser TGT	Ser TGT	Ser TGT	Ala GGT

#s = 2 sites

#a = 1 site

#a/#s=0.5

**To assess selection pressures one needs to calculate the rates (Ka, Ks), i.e. the occurring substitutions as a fraction of the possible syn. and nonsyn. substitutions.**

**Things get more complicated, if one wants to take transition transversion ratios and codon bias into account. See chapter 4 in Nei and Kumar, Molecular Evolution and Phylogenetics.**

Modified from: [mentor.lscf.ucsb.edu/course/spring/eemb102/lecture/Lecture7.ppt](http://mentor.lscf.ucsb.edu/course/spring/eemb102/lecture/Lecture7.ppt)

# dambe

Two programs worked well for me to align nucleotide sequences based on the amino acid alignment,

One is DAMBE (only for windows). This is a handy program for a lot of things, including reading a lot of different formats, calculating phylogenies, it even runs codeml (from PAML) for you.

The procedure is not straight forward, but is well described on the help pages. After installing DAMBE go to HELP -> general HELP -> sequences -> align nucleotide sequences based on ...->

If you follow the instructions to the letter, it works fine.

DAMBE also calculates  $K_a$  and  $K_s$  distances from codon based aligned sequences.

# dambe (cont)

The screenshot shows the DAMBE Help window with the following structure:

- Window Title: Data Analysis in Molecular Biology and Evolution
- Menu Bar: File Edit Alignment Sequences Seq. Analysis Graphics Phylogenetics Control Tools Help
- Sub-window Title: DAMBE Help
- Sub-window Menu Bar: File Edit Bookmark Options Help
- Sub-window Controls: Help Topics Back Print << >>
- Sub-window Navigation: Contents Index Search
- Left Panel (Table of Contents):
  - Overview
  - Main Menu
    - File
    - Edit
    - Sequences
      - Align sequence
      - Align nuc. s
      - Sequences
      - View Sequ
      - Get Rid of
      - Delete seq
      - Delete dup
      - Work on Co
      - Work on A
      - Work on cc
      - Work on cc
      - Work on cc
      - Work on cc
      - Work on cc
      - Restore seq
      - Change seq
      - Get Comple
  - Seq. Analysis

- Main Content Area:
- Section: **Align nuc. seq. against aligned aa. seq.**
- Why:** One frustrating experience I have often had with aligning protein-coding nucleotide sequences is the introduction of many frameshift indels in the aligned sequences, even if the protein genes are known to be all functional and do not have these frameshifting indels. In other words, the introduced frameshifting indels in the aligned sequences are alignment artefacts, and the correctly aligned sequences should have complete codons, not one or two nucleotides, inserted or deleted.
- One way to avoid the above alignment problem is to align the protein-coding nucleotide sequences against amino acid sequences. This obviously requires amino acid sequences which can be obtained in two ways. First, if you have nucleotide sequences of good quality, then you can translate the sequences into amino acids. Second, if you are working on nucleotide sequences deposited in GenBank, then typically you will find the corresponding translated amino acid sequences. DAMBE can read both the nucleotide sequence and the corresponding amino acid sequence in a GenBank sequence.
- How:** Here I illustrate the use of this special feature by assuming that you already have a file containing unaligned protein-coding nucleotide sequences, say **unaligned.fas**, in your hard disk.
- Open the **unaligned.fas** file. When asked whether to align the sequences, click **No**. The unaligned sequences will then be read into DAMBE's buffer. Now click **Sequences/Work on Amino Acid Sequences** to translate the protein-coding nucleotide sequences into amino acid sequences. If the translation results in a number of termination codons embedded in the sequences (represented by "\*\*"), then either your nucleotide sequences are of poor quality or they might be from pseudogenes. In either case you should give up aligning your nucleotide sequences against these junky amino acid sequences.
- If the translation looks good, then click **Sequence/Align sequences with Clustal** to align the translated amino acid sequences. Once this is done, you have a set of aligned amino acid sequences in the DAMBE buffer for you to align your nucleotide sequences against.
- Click **Sequence/Align nuc. seq. against aligned aa seq.** A standard file **Open/Save** dialog box will appear. Choose the **unaligned.fas** file again, which contains the unaligned nucleotide sequences. DAMBE will align the nucleotide sequences against the aligned amino acid sequences in the buffer. This procedure ensures that no frameshifting indels are introduced as an alignment artefact.
- If your sequences were retrieved from GenBank, then most protein-coding genes will already have translated amino acid sequences included in the FEATURES table of GenBank files. You can use DAMBE to first read in all amino acid sequences, align these amino acid sequences, and then ask DAMBE to splice out the corresponding CDS, and align the CDS sequences against aligned amino acid sequences in DAMBE buffer.

If you will be included in the electronic database and do wish to

If you inform also a assist

Citation  
Xia, X  
e  
Xia, X  
p

File: No file

## aa based nucleotide alignments (cont)

An alternative is the *tranalign* program that is part of the *emboss* package. On `bbcxsrv1` you can invoke the program by typing *tranalign*.

Instructions and program description are [here](#).

If you want to use your own dataset in the lab on Monday, generate a codon based alignment with either *dambe* or *tranalign* and save it as a nexus file **and** as a phylip formatted multiple sequence file (using either *clustalw*, *PAUP* (*export* or *tonexus*), *dambe*, or [readseq](#) on the web)

# PAML (codeml) the basic model

$$q_{ij} = \begin{cases} 0, & \text{if the two codons differ at more than one position,} \\ \pi_j, & \text{for synonymous transversion,} \\ \kappa\pi_j, & \text{for synonymous transition,} \\ \omega\pi_j, & \text{for nonsynonymous transversion,} \\ \omega\kappa\pi_j, & \text{for nonsynonymous transition,} \end{cases}$$

The equilibrium frequency of codon  $j$  ( $\pi_j$ ) can be considered a free parameter, but can also be calculated from the nucleotide frequencies at the three codon positions (control variable `CodonFreq`). Under this model, the relationship holds that  $\omega = d_N/d_S$ , the ratio of nonsynonymous/synonymous substitution rates. This basic model is fitted by specifying `model = 0` `NSsites = 0`, in the control file `codeml.ctl`. It forms the basis for more sophisticated models implemented in `codeml`.

# sites versus branches

**You can determine omega for the whole dataset; however, usually not all sites in a sequence are under selection all the time.**

**PAML (and other programs) allow to either determine omega for each site over the whole tree, *Branch Models* , or determine omega for each branch for the whole sequence, *Site Models* .**

**It would be great to do both, i.e., conclude codon 176 in the vacuolar ATPases was under positive selection during the evolution of modern humans – alas, a single site does not provide any statistics ....**

# Sites model(s)

work great have been shown to work great in few instances.  
The most celebrated case is the influenza virus HA gene.

A talk by Walter Fitch (slides and sound) on the evolution of this molecule is [here](#) .

This [article by Yang et al, 2000](#) gives more background on ml approaches to measure omega. The dataset used by Yang et al is here: [flu data.paup](#) .



## sites model in MrBayes

**The MrBayes block in a nexus file might look something like this:**

```
begin mrbayes;  
set autoclose=yes;  
lset nst=2 rates=gamma nucmodel=codon omegavar=Ny98;  
mcmc samplefreq=500 printfreq=500;  
mcmc ngen=500000;  
sump burnin=50;  
sumt burnin=50;  
end;
```

# Vincent Daubin and Howard Ochman: Bacterial Genomes as New Gene Homes: The Genealogy of ORFans in *E. coli*. *Genome Research* 14:1036-1042, 2004

The ratio of non-synonymous to synonymous substitutions for genes found only in the *E. coli* - *Salmonella* clade is lower than 1, but larger than for more widely distributed genes.

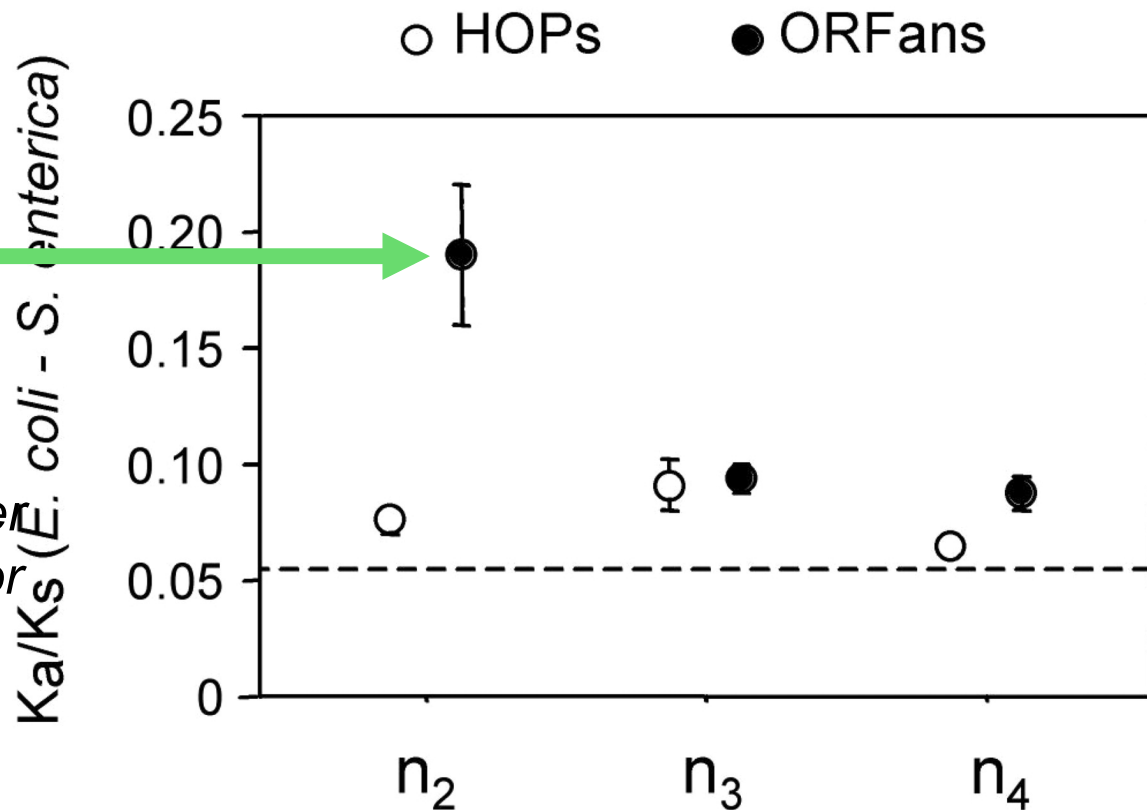


Fig. 3 from Vincent Daubin and Howard Ochman, *Genome Research* 14:1036-1042, 2004

Trunk-of-my-car analogy: Hardly anything in there is the result of providing a selective advantage. Some items are removed quickly (purifying selection), some are useful under some conditions, but most things do not alter the fitness.

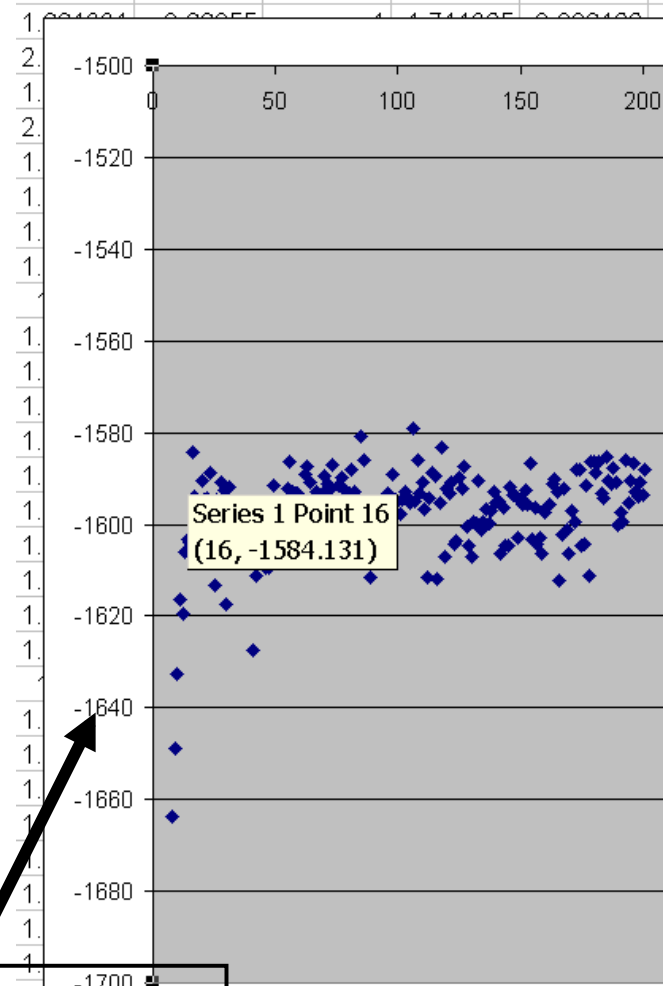
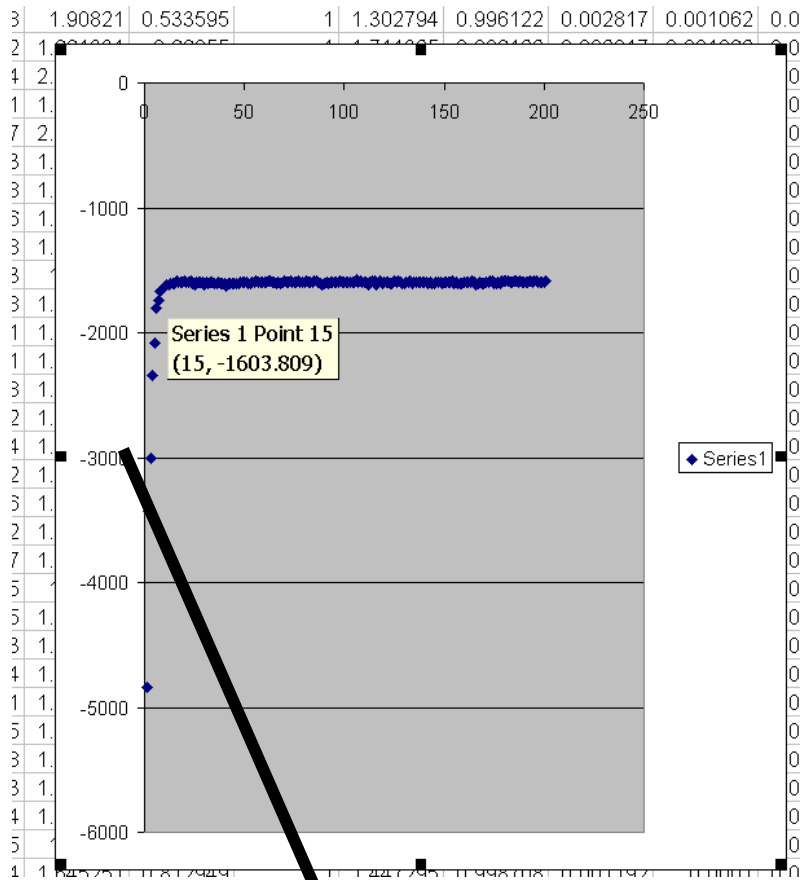


*Could some of the inferred purifying selection be due to the acquisition of novel detrimental characteristics (e.g., protein toxicity)?*

## MrBayes analyzing the \*.nex.p file

- 1. The easiest is to load the file into excel (if your alignment is too long, you need to load the data into separate spreadsheets – see [here](#) exercise 2 item 2 for more info)**
- 2. plot LogL to determine which samples to ignore**
- 3. for each codon calculate the the average probability (from the samples you do not ignore) that the codon belongs to the group of codons with  $\omega > 1$ .**
- 4. plot this quantity using a bar graph.**

# plot LogL to determine which samples to ignore



the same after rescaling the y-axis

for each codon calculate the the average probability

Microsoft Excel - Book1

File Edit View Insert Format Tools Data Window

BT204 = =AVERAGE(BT51:BT203)

	BR	BS	BT	BU	BV	BW
199	0.020322	0.025016	0	0	0	
200	0.018418	0.028381	0	0	0	
201	0.018418	0.028381	0	0	0	
202	0.018418	0.028381	0	0	0	
203	0.018418	0.028381	0	0	0	
204		average	0			
205						
206						

Microsoft Excel - Book1

Edit View Insert Format Tools Data Window Help Adobe PDF

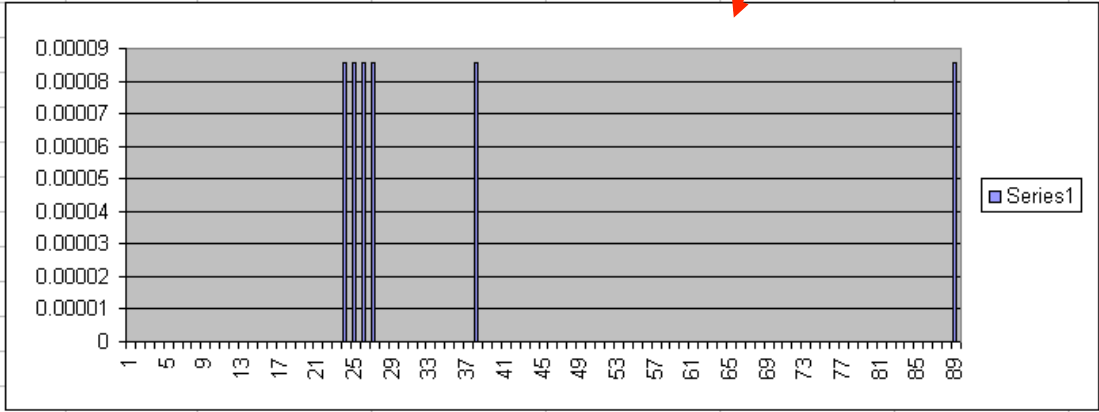
CP204 = =AVERAGE(CP51:CP203)

CO	CP	CQ	CR	CT	CU	CV
0	0	0.000215	0.000215	0.000215	0.000215	0
0	0	0.000018	0.000018	0.000018	0.000018	0
0	0	0.000124	0.000124	0.000124	0.000124	0
0	0	0.000044	0.000044	0.000044	0.000044	0
0	0	0.00033	0.00033	0.00033	0.00033	0
0	6.53595E-09	8.55948E-05	8.55948E-05	8.55817E-05	8.55752E-05	0

enter formula

copy paste formula

plot row



## MrBayes on bbcxrv1

**If you do this for your own data,**

- run the procedure first for only 50000 generations (takes about 30 minutes) to check that everything works as expected,**
- then run the program overnight for at least 500 000 generations.**
- Especially, if you have a large dataset, do the latter twice and compare the results for consistency. ( I prefer two runs over 500000 generations each over one run over a million generations.)**

**The preferred wa to run mrbayes is to use the command line:**

**>mb**

**Do example on threonlyRS**

# PAML – codeml – sites model

the paml package contains several distinct programs for nucleotides (baseml) protein coding sequences and amino acid sequences (codeml) and to simulate sequences evolution.

The input file needs to be in phylip format.

By default it assumes a sequential format (e.g. [here](#)).

If the sequences are interleaved, you need to add an “I” to the first line, as in these example headers:

```

        6      467      I
gi|1613157 ----- MSDNDTIVAQ ATPPGRGGVG ILRISGFKAR EVAETVLGKL
gi|2212798 ----- MSTTDTIVAQ ATPPGRGGVG ILRVSGRAAS EVAHAVLGKL
gi|1564003 MALIQSCSGN TMTTDTIVAQ ATAPGRGGVG IIRVSGPLAA HVAQTVTGRG
gi|1560076 -----M QAATETIVAI ATAQGRGGVG IIVRSGPLAG QMAVAVSGRQ
gi|2123365 -----MN--- -ALPSTIVAI ATAAGTGGIG IIVRLSGPQSV QIAAALGIAG
gi|1583936 -----MSQRS TKMGDTIAAI ATASGAAGIG IIRLSGSLIK TIATGLGMTT

```

```

5  855  I
human
goat-cow
rabbit
rat
marsupial
1
GTG CTG TCT CCT GCC GAC AAG ACC AAC GTC AAG GCC GCC TGG GGC AAG GTT GGC GCG CAC
... ..G.C ... ..T ..T ... .. ..GC A..
... ..C ..T ... ..A.. ..A.T ... ..AA ... A.C ... AGC ...
... ..C ... G.A .AT ... ..A ... ..A.. ..AA. TG. ... ..G ... A.. ..T .GC ..T
... ..C ..G GA. ..T ... ..T C.. ..G ..A ... AT. ... ..T ... ..G ..A .GC ...
61

GCT GGC GAG TAT GGT GCG GAG GCC CTG GAG AGG ATG TTC CTG TCC TTC CCC ACC ACC AAG
... ..A .CT ... ..C ..A ... ..T ... .. ..AG. ... .. .. ..
.G. ... ..C ..C ... ..G.. ... ..T.. GG. ... .. .. ..
.G. ..T ..A ... ..C ..A. ... ..A C.. ... ..GCT G.. ... .. ..
..C ..T ..CC ..C .CA ..T ..A ..T ..T ..CC ..A ..CC ... ..C ... ..T ... ..A

```



## **PAML – codeml – sites model (cont.)**

**the program is invoked by typing codeml followed by the name of a control file that tells the program what to do.**

**paml can be used to find the maximum likelihood tree, however, the program is rather slow. Phyml is a better choice to find the tree, which then can be used as a user tree.**

**An example for a codeml.ctl file is [codeml.hv1.sites.ctl](#)**

**This file directs codeml to run three different models:**

**one with an omega fixed at 1, a second where each site can be either have an omega between 0 and 1, or an omega of 1, and third a model that uses three omegas as described before for MrBayes.**

**The output is written into a file called [Hv1.sites.codeml\\_out](#) (as directed by the control file).**

**Point out log likelihoods and estimated parameter line (kappa and omegas)**

**Additional useful information is in the [rst](#) file generated by the codeml**

**Discuss overall result.**

## **PAML – codeml – branch model**

**For the same dataset to estimate the dN/dS ratios for individual branches, you could use this file [codeml.hv1.branches.ctl](#) as control file.**

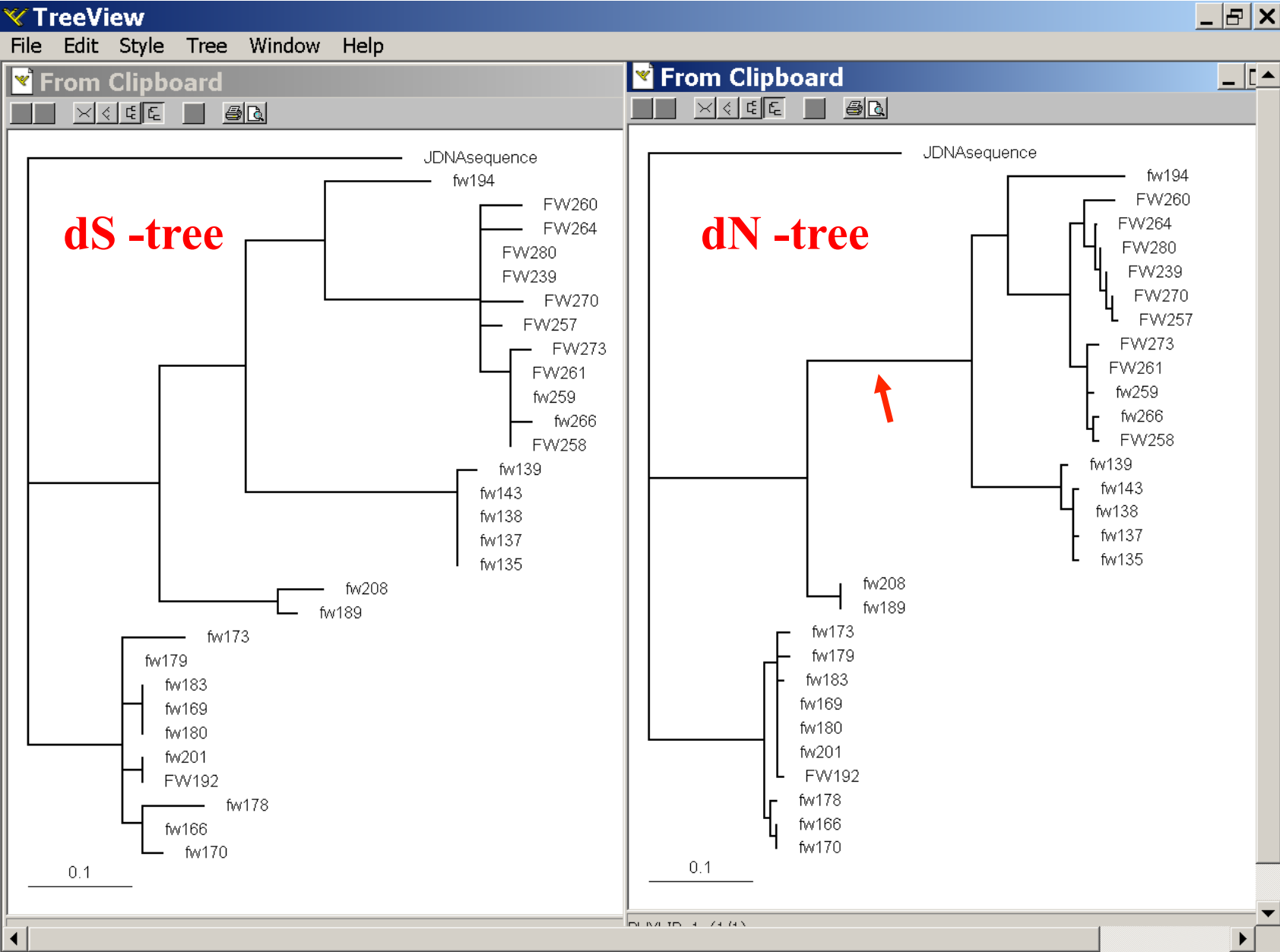
**The output is written, as directed by the control file, into a file called [Hv1.branch.codeml\\_out](#)**

**A good way to check for episodes with plenty of non-synonymous substitutions is to compare the dn and ds trees.**

**Also, it might be a good idea to repeat the analyses on parts of the sequence (using the same tree). In this case the sequences encode a family of spider toxins that include the mature toxin, a propeptide and a signal sequence (see [here](#) for more information).**

**Bottom line: one needs plenty of sequences to detect positive selection.**

# PAML – codeml – branch model



# where to get help

read the manuals and help files

check out the discussion boards at <http://www.rannala.org/phpBB2/>

## else

there is a new program on the block called [hy-phy](#)  
(=hypothesis testing using phylogenetics).

The easiest is probably to run the analyses on the authors [datamonkey](#).



## Discussion: Other ways to detect positive selection?

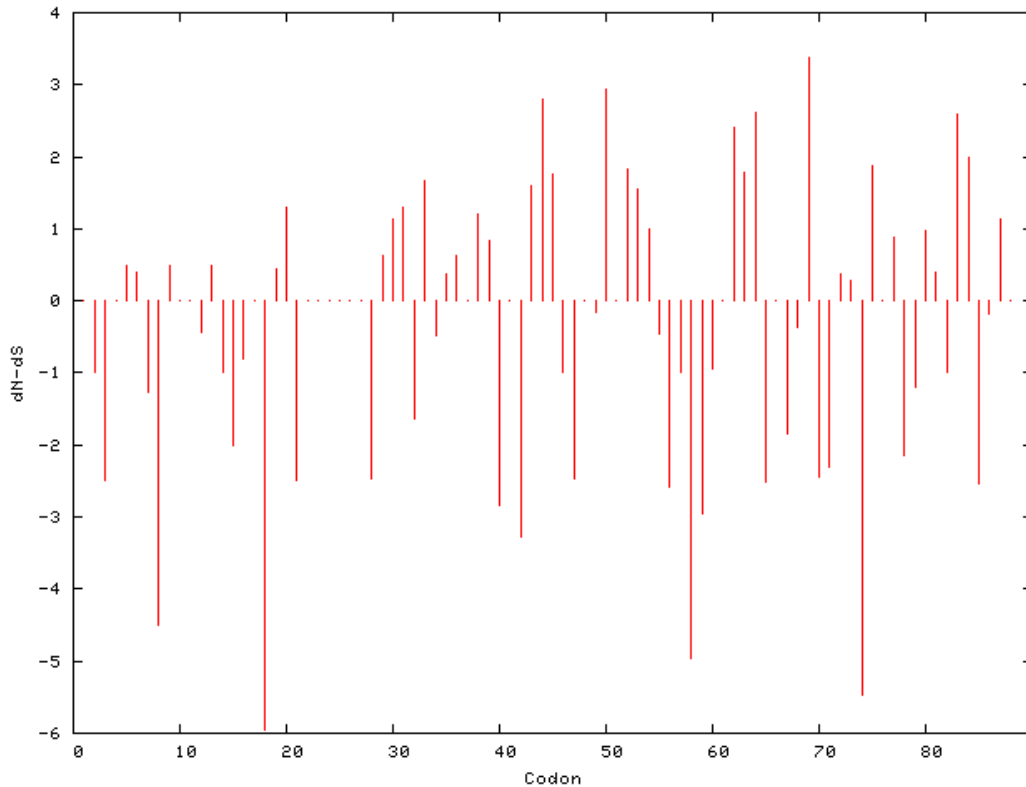
Selective sweep -> fewer alleles present in population

Repeated episodes of positive selection -> high dN

If time discuss <http://online.itp.ucsb.edu/online/infobio01/fitch1/>

# hy-phy

## Results of an analysis using the SLAC approach



FOUND 4 POSITIVELY SELECTED SITES (0.2 significance level)

Codon	dN-dS	Normalized dN-dS	p-value
45	2.80905	1.57283	0.174148
51	2.94548	1.64923	0.109144
65	2.62064	1.46734	0.197579
70	3.37001	1.88693	0.124868

FOUND 13 NEGATIVELY SELECTED SITES (0.2 significance level)

Codon	dN-dS	Normalized dN-dS	p-value
4	-2.5	-1.39979	0.111111
9	-4.5	-2.51963	0.0178326
19	-5.94245	-3.32728	0.0243467
22	-2.5	-1.39979	0.111111
41	-2.84041	-1.59039	0.193214
48	-2.45744	-1.37597	0.0793724
59	-4.96667	-2.78093	0.0236379
60	-2.96058	-1.65768	0.108898
66	-2.51831	-1.41004	0.15211
71	-2.45417	-1.37413	0.129462
72	-2.31427	-1.2958	0.162177
75	-5.47043	-3.06299	0.0388673
86	-2.54472	-1.42483	0.151309

more output might still be [here](#)

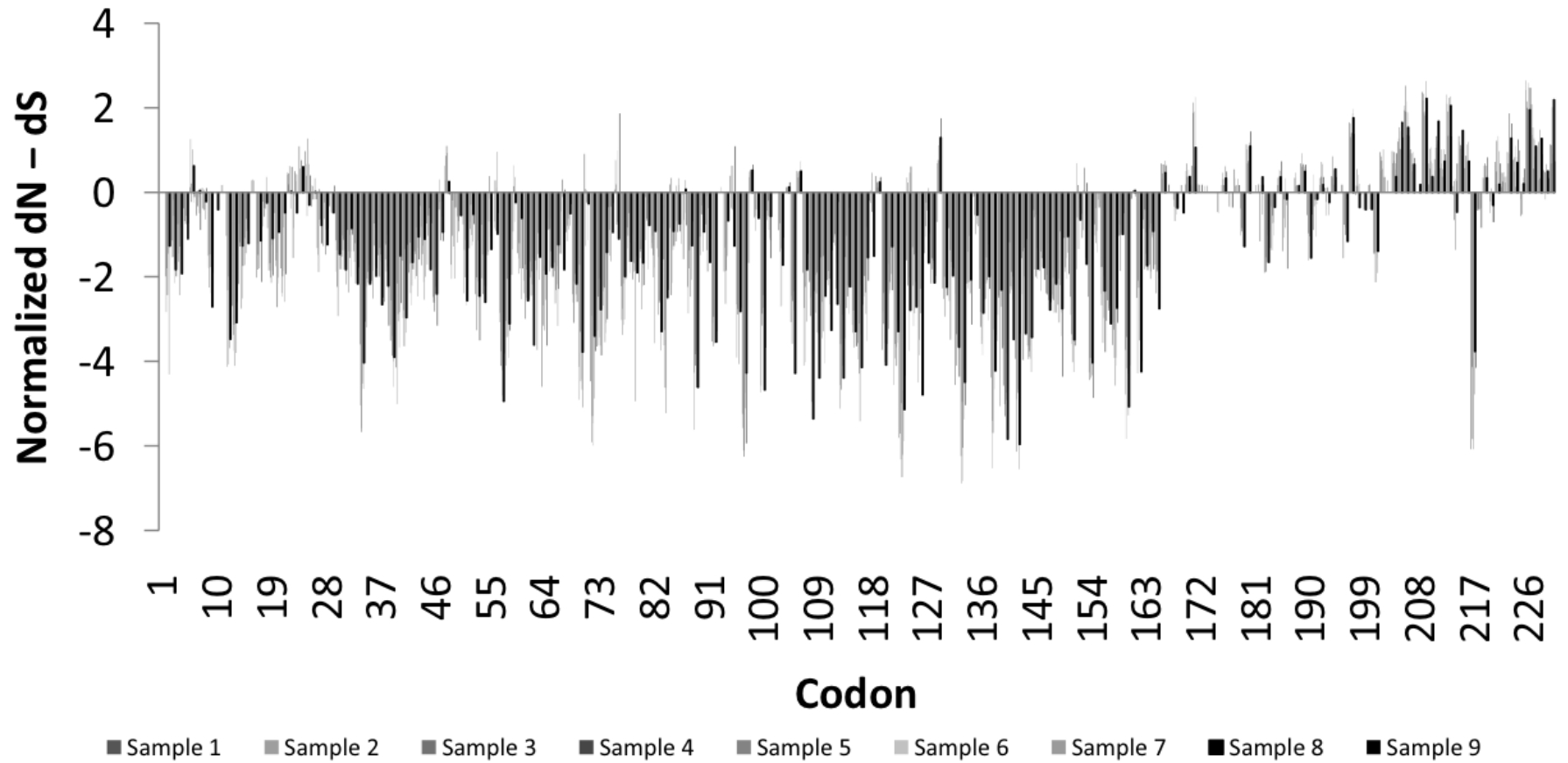


Fig 1. Patterns of substitutions: Bars represent  $dN > dS$  (positive) or  $dN < dS$  (negative) in random samples of 148 – 150 sequences (A) and the whole dataset of 1312 viruses (B). Included in B are regions of mapped activity and 3D structures of the RNA-binding domain (RBD, panel I) [21] and Effector domain (ED, rotated to expose the 7  $\beta$ -sheets (panel II) and 2  $\alpha$ -helices (panel II)) [7] with residues under negative (yellow/brown), neutral (gray) or positive (red) selection highlighted. Residues 208-230 not included in the 3D structure of the ED are disordered (compare with figure 5). Note sites with  $dN > dS$  map on the helix motifs of the ED or the linkers flanking them or the disordered region.

Hy-Phy -



Hypothesis Testing using Phylogenies.

Using Batchfiles or GUI

Information at <http://www.hyphy.org/>

Selected analyses also can be performed online at <http://www.datamonkey.org/>

A screenshot of the Datamonkey website homepage. The header includes navigation links: ANALYZE YOUR DATA, HOME, HELP, CITATIONS, JOB QUEUE, STATS, HYPHY PACKAGE, and DATAMONKEYS WELCOME. The main banner features the text "DATAMONKEY" in large, bold letters, with "RAPID DETECTION OF POSITIVE SELECTION" below it. A cartoon monkey character is visible on the left. Below the banner, a welcome message states: "Welcome to the free public server for detecting signatures of positive and negative selection from coding sequence alignments using state-of-the-art statistical models. This service is brought to you by the viral evolution group at the Antiviral Research Center of the University of California, San Diego. The methods and software tools are developed and maintained by Sergei L. Kosakovsky Pond, Simon Frost and Art Poon." A red box contains an announcement: "April 14th, 2008: We have implemented 4 queues for jobs of different types on datamonkey.org. This will prevent a situation when complex long-running jobs (e.g. GABranch) hold up the entire queue for many hours. Model Selection/FEL/IFEL (queue 1), REL/PARRIS (queue 2), GABranch (queue 3) and Spidermonkey/BGM (queue 4) each receive their own scheduling and a job of each type can run concurrently with jobs of other types." Below this, a section titled "Datamonkey.org can help you answer the following questions (publications citing datamonkey.org) :" lists three questions: "Which codon sites are under positive or negative selection?", "Is there evidence of selection in my alignment?", and "Which codon sites are under positive or negative selection at the population level?". Each question is followed by a brief description of the methods used.



# Example testing for dN/dS in two partitions of the data -- John's dataset

The screenshot displays the HyPhy software interface. The main window shows a multiple sequence alignment of nucleotide data for 150 species across 690 sites. The alignment is viewed in a window titled "DataSet ns1\_all\_nt\_8\_sample". The alignment is color-coded by site, with sites 460-525 in blue and sites 526-690 in orange. The alignment is viewed in a window titled "DataSet ns1\_all\_nt\_8\_sample".

Partition Name	Partition Type	Tree Topology	Substitution Model	Parameters	Equilibrium Freqs.	Rate Classes
END	Codon	Tree_1	MG94xTN93_3x4	Global	Partition	
BEGINNING	Codon	Tree_12	MG94xTN93_3x4	Global	Partition	

Nucleotide Data. 690 sites (403 distinct patterns), 150 species. Current Selection:525-525

Set up two partitions, define model for each, optimize likelihood

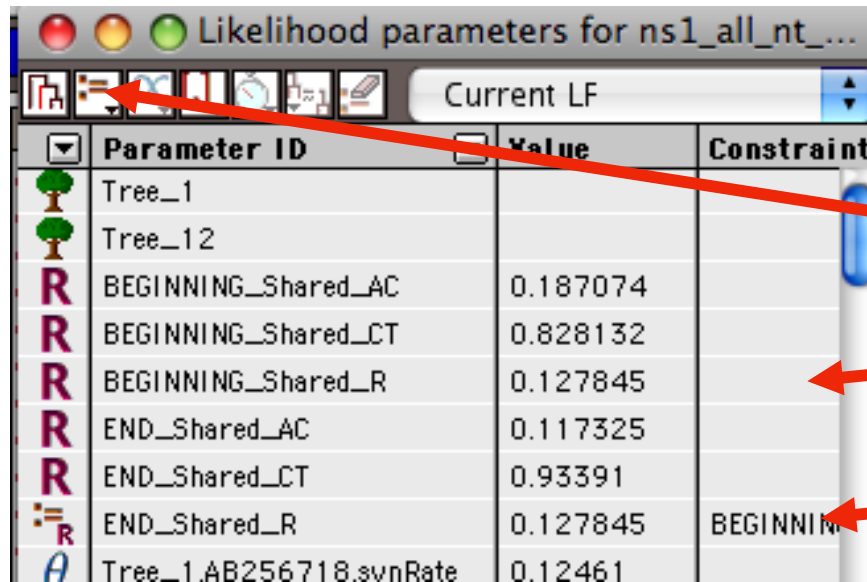
# Example testing for dN/dS in two partitions of the data -- John's dataset

Parameter ID	Value	Constraint
Tree_1		
Tree_12		
BEGINNING_Shared_AC	0.187074	
BEGINNING_Shared_CT	0.828132	
BEGINNING_Shared_LR	0.127845	
END_Shared_AC	0.117325	
END_Shared_CT	0.93391	
END_Shared_LR	0.946316	
$\theta$ Tree_1.AB256718.synRate	0.12461	
$\theta$ Tree_1.AF001672.synRate	0.016737	
$\theta$ Tree_1.AF009898.synRate	0	
$\theta$ Tree_1.AF055424.synRate	0.017357	
$\theta$ Tree_1.AF074267.synRate	0	
$\theta$ Tree_1.AF074279.synRate	0.0527182	
$\theta$ Tree_1.AF084286.synRate	0.0176037	
$\theta$ Tree_1.AF144307.synRate	0.0528252	
$\theta$ Tree_1.AF256183.synRate	0	
$\theta$ Tree_1.AF256188.synRate	0.0174124	
$\theta$ Tree_1.AF523503.synRate	0.0527042	
$\theta$ Tree_1.AJ344036.synRate	0	
$\theta$ Tree_1.AJ410594.synRate	0.0350104	
$\theta$ Tree_1.AJ410598.synRate	0.0174538	
$\theta$ Tree_1.AM502792.synRate	0.0174516	

Save Likelihood Function  
then  
select as alternative

The dN/dS ratios for the  
two partitions are  
different.

# Example testing for dN/dS in two partitions of the data -- John's dataset



Parameter ID	Value	Constraint
Tree_1		
Tree_12		
BEGINNING_Shared_AC	0.187074	
BEGINNING_Shared_CT	0.828132	
BEGINNING_Shared_R	0.127845	
END_Shared_AC	0.117325	
END_Shared_CT	0.93391	
END_Shared_R	0.127845	BEGINNING
Tree_1.AB256718.svnRate	0.12461	

Set up null hypothesis, i.e.:

The two dN/dS are equal

(to do, select both rows and then click the define as equal button on top)

Example testing for dN/dS in two partitions of the data --  
John's dataset

The screenshot displays the HyPhy interface with a table of likelihood parameters. The table has columns for Parameter ID, Value, and Constraint. The parameters listed include tree structures (Tree\_1, Tree\_12) and various shared parameters (BEGINNING\_Shared\_\*, END\_Shared\_\*). Two red arrows point to the BEGINNING\_Shared parameters. A HYPHY Console window is open at the bottom, showing a command line with node IDs and branch lengths.

Parameter ID	Value	Constraint
Tree_1		
Tree_12		
BEGINNING_Shared_AC	0.187238	
BEGINNING_Shared_CT	0.891995	
BEGINNING_Shared_R	0.19809	
END_Shared_AC	0.126137	
END_Shared_CT	0.770683	
END_Shared_R	0.19809	BEGINNING
Tree_1.AB256718.synRate	0.309711	
Tree_1.AF001672.synRate	0.0364501	
Tree_1.AF009898.synRate	0	
Tree_1.AF055424.synRate	0.0414451	
Tree_1.AF074267.synRate	0	
Tree_1.AF074279.synRate	0.131262	
Tree_1.AF084286.synRate	0.0419524	
Tree_1.AF144307.synRate	0.129191	
Tree_1.AF256183.synRate	0	
Tree_1.AF256188.synRate	0.0415364	
Tree_1.AF523		
Tree_1.AJ34475		
Tree_1.AJ410		
Tree_1.AJ410		
Tree_1.AM503		
Tree_1.AM503		
Tree_1.AY028		
Tree_1.AY210		
Tree_1.AY241		

Example testing for dN/dS in two partitions of the data --  
John's dataset

The screenshot shows the HyPhy software interface. The main window displays a table of likelihood parameters for a dataset named 'ns1\_all\_nt\_8'. The table has columns for 'Parameter ID', 'Value', and 'Constraint'. The parameters listed include tree structures (Tree\_1, Tree\_12), shared rates (BEGINNING\_Shared\_AC, BEGINNING\_Shared\_CT, BEGINNING\_Shared\_R, END\_Shared\_AC, END\_Shared\_CT, END\_Shared\_R), and site-specific synon rates (Tree\_1.AB256718.synRate, etc.). A dropdown menu labeled 'Current LF' is open, showing a list of parameter names. A 'HYPHY Console' window is also visible at the bottom.

Parameter ID	Value	Constraint
Tree_1		
Tree_12		
BEGINNING_Shared_AC	0.187238	
BEGINNING_Shared_CT	0.891995	
BEGINNING_Shared_R	0.19809	
END_Shared_AC	0.126137	
END_Shared_CT	0.770683	
END_Shared_R	0.19809	BEGINNING
Tree_1.AB256718.synRate	0.309711	
Tree_1.AF001672.synRate	0.0364501	
Tree_1.AF009898.synRate	0	
Tree_1.AF055424.synRate	0.0414451	
Tree_1.AF074267.synRate	0	
Tree_1.AF074279.synRate	0.131262	
Tree_1.AF084286.synRate	0.0419524	
Tree_1.AF144307.synRate	0.129191	
Tree_1.AF256183.synRate	0	
Tree_1.AF256188.synRate	0.0415364	
Tree_1.AF523		
Tree_1.AJ34475		
Tree_1.AJ410		
Tree_1.AJ410		
Tree_1.AM500		
Tree_1.AM500		
Tree_1.AY028		
Tree_1.AY210		
Tree_1.AY241		

Name and save as Null-hyp.

# Example testing for dN/dS in two partitions of the data -- John's dataset

The screenshot shows the HyPhy software interface. The main window is titled "Likelihood parameters for ns1\_all\_nt\_..." and has a dropdown menu set to "Null Hyp (no partitions)". A red arrow points from the text on the right to this dropdown menu. Below the menu is a table of parameters:

Parameter ID	Value	Constraint
Tree_1		
Tree_12		
BEGINNING_Shared_AC		
BEGINNING_Shared_CT		
BEGINNING_Shared_R		
END_Shared_AC		
END_Shared_CT		
END_Shared_R		
Tree_1.AB256718.synR		
Tree_1.AF001672.synR		

Overlaid on the bottom right is a "HYPHY Console" window showing the following output:

```
Time taken = 21606.9 seconds  
LF evaluations/second = 4.31552  
  
Likelihood Ratio Test  
  
2*LR = 225.881  
DF = 1  
P-Value = 0
```

After selecting LRT (= Likelihood Ratio test), the console displays the result, i.e., **the beginning and end of the sequence alignment have significantly different dN/dS ratios.**

# Example testing for dN/dS in two partitions of the data -- John's dataset

Alternatively, especially if the the two models are not nested, one can set up two different windows with the same dataset:

The screenshot displays two windows from a software application, likely a phylogenetic analysis tool, showing nucleotide data and partition settings for a dataset named 'ns1\_all\_nt\_8\_sample'.

The top window, titled "DataSet ns1\_all\_nt\_8\_sample\_finished", shows a sequence alignment with columns numbered 530 to 620. The bottom window, titled "DataSet ns1\_all\_nt\_8\_sample", shows a sequence alignment with columns numbered 460 to 530.

Below the sequence alignments are two tables for partition settings. The top table, labeled "Model 1", shows two partitions: "END" and "BEGINNING". Both partitions use a "Codon" partition type, "Tree\_1" and "Tree\_12" topologies, and "MG94xTN93\_3x4" substitution models. The "Equilibrium Freqs." column is highlighted with a red arrow.

The bottom table, labeled "Model 2", shows a single partition named "ns1\_all\_nt\_8\_san". It uses a "Codon" partition type, "ns1\_all\_nt\_8\_san" topology, and "MG94xTN93\_3x4" substitution model. The "Equilibrium Freqs." column is highlighted with a red arrow.

Both tables also include columns for "Parameters" and "Equilibrium Freqs.". The status bar at the bottom of each window indicates "Nucleotide Data. 690 sites (403 distinct patterns), 150 species. Current Selection: 479-479" for the top window and "Current Selection: 1-690" for the bottom window.

# Example testing for dN/dS in two partitions of the data -- John's dataset

Simulation under model 1, evaluation under model 2, calculate LR  
Compare real LR to distribution from simulated LR values. The result might look something like this

or

this

