MCB 5472

Types of selection

Peter Gogarten
Office: BSP 404
phone: 860 486-4061,
Email: gogarten@uconn.edu
Old assignment

Write script from exam

Work of Student project
Simple: dinucleotide sequence from single nucleotide sequence

Part I: Reading data

```perl
#!/usr/bin/perl
# hw10.pl modified from Erin Duffy
use warnings;
die "usage: hw10.pl <limit>\n" unless @ARGV == 1;
my $filename=ARGV[0];
open(IN, "< $filename") or die "cannot open $filename:$!";
print "The file $filename is being analyzed:\n\n\n";
open(OUT, ">> table.out");
my $inseq = "";
my $ref = ""
my %nmer = 0;
while(defined(my $line=<IN>)){
    chomp($line);
    if ($line=~/>/){
        $line =~ s/>ref\(\S+\)///g;
        $ref=$1;
        print "$ref\n";
    } else{
        $inseq .= $line;
    } 
}
$inseq =~ tr/atgc/ATGC/;
$inseq =~ s/\s+/\//g;
$inseq =~ s/\N/\//g;
# print "$gi \n\n$inseq \n"
print "$inseq \n";
```
Simple: dinucleotide sequence from single nucleotide sequence

Part 2: Analyzing data

```perl
####input done, start analysis
my @seqarray = split('//', $inseq);
    my $count = @seqarray;
    for (my $i = 0; $i<$count-1; $i++) {
        my $subseq1 = @seqarray[$i].@seqarray[$i+1];
        # my $nmer = '

        $nmer{$subseq1} += 1;
    }
my @nucs1 = ("A","T","G","C");
my @nucs2 = ("A","T","G","C");
foreach my $nuc (@nucs1) {
    foreach (@nucs2) {
        $dimer=$nuc."
        print "$dimer\t$nmer{$dimer}\n"
    }
}
    # foreach (@sorted_by_value) {
    #     print "dimer\t\toccurred\nmer{$_} times \n"

# print "@seqarray\n"
# print "@subseq\n"
# my $count = @seqarray;
# print "There were $count sequences read in.\n"

close(IN);
# close(OUT);
```
Subroutine: dinucleotide sequence from single nucleotide sequence

```perl
sub diffreq {
    my $inseq = $_[0];
    my %nmer = ();
    $inseq =~ tr/atgc/ATGC/;
    $inseq =~ s/\S//g;
    $inseq =~ s/N//g;

    my @seqarray = split(\/, $inseq);
    my $count = @seqarray;
    for (my $i = 0; $i < $count-1; $i++) {
        my $subseq1 = $seqarray[$i].$seqarray[$i+1];
        $nmer{$_}$subseq1 += 1;
    }

    my @nucs1 = ("A", "T", "G", "C");
    my @nucs2 = ("A", "T", "G", "C");
    foreach my $nuc (@nucs1) {
        foreach my $temp (@nucs2) {
            my $dimer=$nuc.$temp;
            if (!exists $nmer{$_}$dimer) {$_}$nmer{$_}$dimer=0; #defeats the purpose of using a hash, but we want a complete table
                print "\t$nmer{$_}$dimer";
            }
            print "\n";
        }
    }
}
```
Analyze file with multiple ORFs

Part 1: Open file and print header

```perl
#hw10.pl modified from Erin Duffy
use warnings;

die "usage: script.pl <limit> \n" unless @ARGV == 1;
my $filename=$ARGV[0];
open(IN, "< $filename") or die "cannot open $filename: !";
print "$filename is being analyzed:\n\n\n";
open(OUT, "> table.out");
my $inseq = "";
my $ref = "";
my @nmer = ();
my @nucs1 = ("A","T","G","C");
my @nucs2 = ("A","T","G","C");

##Print header##
print "REF";
foreach my $nuc (@nucs1) {
    foreach (@nucs2) {
        $dimer=$nuc.$_
        print "\t$dimer";
    }
}

print "\n";
```

Add OUT to print to file
Analyze file with multiple ORFs

Part 2: Program Flow

```
my $flag="F";

while(defined(my $line=<IN>)){
    chomp($line);
    if (($line=~/>/ )&& ($flag eq "T")) { #this is the second or more time a sequence is encountered
        $line =~ s/\:\\:(c?\d+)/g;
        $ref=$1;
        dimfreq ($inseq); #calls subroutine to analyze sequence
        $inseq=""; #reset sequence assembled from infile to an empty string
        print "$ref"; } # print ref of next sequence to analyze
    
    if (($line=~/>/ )&& $flag eq "F") { #this is the first sequence in the multiple fasta file
        $line =~ s/\:\\:(c?\d+)/g;
        $ref=$1;
        print "$ref"; #prints seqname to table
        $flag="T";#
    }
    
    else {
        $inseq .= $line;
    }
}
```
Subroutine: dinucleotide sequence from single nucleotide sequence

```perl
sub dimfreq { #output the last sequence into table
    $inseq = "";
    my $inseq = $_[0];
    my %nmer = ();
    $inseq =~ tr/atgc/ATGC/;
    $inseq =~ s/\s+///g;
    $inseq =~ s/\N//g;
    my @seqarray = split(//, $inseq);
    my $count = @seqarray;
    for (my $i = 0; $i < $count-1; $i++) {
        my $subseq1 = $seqarray[$i].$seqarray[$i+1];
        $nmer{$subseq1} += 1;
    }
    my @nucs1 = ("A", "T", "G", "C");
    my @nucs2 = ("A", "T", "G", "C");
    foreach my $nuc (@nucs1) {
        foreach my $temp (@nucs2) {
            my $dimer = $nuc.$temp;
            if (!exists $nmer{$dimer}) {$nmer{$dimer} = 0;}
            print "\t$nmer{$dimer}"
        }
        print "\n";
    }
    close(IN);
    close(OUT);
```
<table>
<thead>
<tr>
<th>REF</th>
<th>AA</th>
<th>AT</th>
<th>AG</th>
<th>AC</th>
<th>TA</th>
<th>TT</th>
<th>TG</th>
<th>TC</th>
<th>GA</th>
<th>GT</th>
<th>GG</th>
<th>GC</th>
<th>CA</th>
<th>CT</th>
<th>CG</th>
<th>CC</th>
<th>Sum</th>
</tr>
</thead>
<tbody>
<tr>
<td>55</td>
<td>26</td>
<td>14</td>
<td>70</td>
<td>45</td>
<td>16</td>
<td>25</td>
<td>49</td>
<td>77</td>
<td>69</td>
<td>43</td>
<td>188</td>
<td>101</td>
<td>43</td>
<td>85</td>
<td>95</td>
<td>181</td>
<td>1127</td>
</tr>
<tr>
<td>1235</td>
<td>36</td>
<td>26</td>
<td>80</td>
<td>68</td>
<td>14</td>
<td>37</td>
<td>61</td>
<td>70</td>
<td>91</td>
<td>47</td>
<td>223</td>
<td>116</td>
<td>68</td>
<td>73</td>
<td>113</td>
<td>148</td>
<td>1271</td>
</tr>
<tr>
<td>2490</td>
<td>27</td>
<td>39</td>
<td>78</td>
<td>68</td>
<td>8</td>
<td>27</td>
<td>68</td>
<td>99</td>
<td>102</td>
<td>63</td>
<td>227</td>
<td>122</td>
<td>73</td>
<td>74</td>
<td>142</td>
<td>208</td>
<td>1425</td>
</tr>
<tr>
<td>c4895</td>
<td>28</td>
<td>4</td>
<td>55</td>
<td>38</td>
<td>14</td>
<td>33</td>
<td>38</td>
<td>67</td>
<td>53</td>
<td>23</td>
<td>146</td>
<td>96</td>
<td>30</td>
<td>92</td>
<td>80</td>
<td>185</td>
<td>982</td>
</tr>
<tr>
<td>c5506</td>
<td>20</td>
<td>8</td>
<td>32</td>
<td>28</td>
<td>11</td>
<td>21</td>
<td>26</td>
<td>40</td>
<td>35</td>
<td>21</td>
<td>110</td>
<td>49</td>
<td>24</td>
<td>46</td>
<td>47</td>
<td>101</td>
<td>619</td>
</tr>
<tr>
<td>c7340</td>
<td>64</td>
<td>50</td>
<td>107</td>
<td>112</td>
<td>18</td>
<td>45</td>
<td>97</td>
<td>116</td>
<td>144</td>
<td>62</td>
<td>273</td>
<td>148</td>
<td>106</td>
<td>120</td>
<td>150</td>
<td>236</td>
<td>1848</td>
</tr>
<tr>
<td>c7993</td>
<td>15</td>
<td>4</td>
<td>40</td>
<td>25</td>
<td>13</td>
<td>28</td>
<td>25</td>
<td>40</td>
<td>39</td>
<td>32</td>
<td>110</td>
<td>54</td>
<td>18</td>
<td>42</td>
<td>60</td>
<td>101</td>
<td>646</td>
</tr>
<tr>
<td>c8699</td>
<td>39</td>
<td>11</td>
<td>69</td>
<td>26</td>
<td>4</td>
<td>19</td>
<td>27</td>
<td>27</td>
<td>72</td>
<td>10</td>
<td>92</td>
<td>69</td>
<td>28</td>
<td>37</td>
<td>57</td>
<td>76</td>
<td>663</td>
</tr>
<tr>
<td>8799</td>
<td>9</td>
<td>2</td>
<td>13</td>
<td>8</td>
<td>4</td>
<td>5</td>
<td>8</td>
<td>6</td>
<td>8</td>
<td>6</td>
<td>24</td>
<td>20</td>
<td>11</td>
<td>10</td>
<td>14</td>
<td>24</td>
<td>172</td>
</tr>
<tr>
<td>9022</td>
<td>1</td>
<td>2</td>
<td>4</td>
<td>3</td>
<td>4</td>
<td>10</td>
<td>2</td>
<td>9</td>
<td>2</td>
<td>2</td>
<td>11</td>
<td>10</td>
<td>4</td>
<td>11</td>
<td>8</td>
<td>11</td>
<td>94</td>
</tr>
<tr>
<td>12493</td>
<td>34</td>
<td>27</td>
<td>56</td>
<td>55</td>
<td>17</td>
<td>24</td>
<td>49</td>
<td>67</td>
<td>68</td>
<td>35</td>
<td>127</td>
<td>87</td>
<td>52</td>
<td>72</td>
<td>85</td>
<td>158</td>
<td>1013</td>
</tr>
<tr>
<td>c14311</td>
<td>4</td>
<td>12</td>
<td>26</td>
<td>33</td>
<td>12</td>
<td>33</td>
<td>48</td>
<td>75</td>
<td>27</td>
<td>26</td>
<td>114</td>
<td>84</td>
<td>32</td>
<td>97</td>
<td>63</td>
<td>144</td>
<td>830</td>
</tr>
<tr>
<td>14329</td>
<td>24</td>
<td>12</td>
<td>58</td>
<td>48</td>
<td>14</td>
<td>24</td>
<td>43</td>
<td>54</td>
<td>70</td>
<td>17</td>
<td>147</td>
<td>105</td>
<td>34</td>
<td>82</td>
<td>91</td>
<td>171</td>
<td>994</td>
</tr>
<tr>
<td>15508</td>
<td>17</td>
<td>1</td>
<td>46</td>
<td>10</td>
<td>4</td>
<td>19</td>
<td>13</td>
<td>47</td>
<td>35</td>
<td>14</td>
<td>99</td>
<td>59</td>
<td>19</td>
<td>49</td>
<td>49</td>
<td>117</td>
<td>598</td>
</tr>
<tr>
<td>16106</td>
<td>37</td>
<td>16</td>
<td>68</td>
<td>54</td>
<td>10</td>
<td>14</td>
<td>63</td>
<td>49</td>
<td>74</td>
<td>47</td>
<td>159</td>
<td>90</td>
<td>54</td>
<td>59</td>
<td>80</td>
<td>148</td>
<td>1022</td>
</tr>
<tr>
<td>17121</td>
<td>36</td>
<td>17</td>
<td>78</td>
<td>41</td>
<td>12</td>
<td>29</td>
<td>55</td>
<td>55</td>
<td>87</td>
<td>33</td>
<td>171</td>
<td>96</td>
<td>38</td>
<td>72</td>
<td>82</td>
<td>126</td>
<td>1028</td>
</tr>
<tr>
<td>c18677</td>
<td>19</td>
<td>6</td>
<td>44</td>
<td>29</td>
<td>8</td>
<td>23</td>
<td>22</td>
<td>41</td>
<td>46</td>
<td>7</td>
<td>94</td>
<td>59</td>
<td>25</td>
<td>58</td>
<td>47</td>
<td>106</td>
<td>634</td>
</tr>
<tr>
<td>c20001</td>
<td>24</td>
<td>17</td>
<td>74</td>
<td>65</td>
<td>30</td>
<td>50</td>
<td>67</td>
<td>67</td>
<td>87</td>
<td>47</td>
<td>246</td>
<td>113</td>
<td>38</td>
<td>100</td>
<td>107</td>
<td>190</td>
<td>1322</td>
</tr>
<tr>
<td>c20374</td>
<td>7</td>
<td>2</td>
<td>22</td>
<td>16</td>
<td>9</td>
<td>16</td>
<td>19</td>
<td>22</td>
<td>23</td>
<td>15</td>
<td>79</td>
<td>25</td>
<td>8</td>
<td>33</td>
<td>23</td>
<td>54</td>
<td>373</td>
</tr>
<tr>
<td>c21072</td>
<td>10</td>
<td>8</td>
<td>43</td>
<td>22</td>
<td>7</td>
<td>28</td>
<td>40</td>
<td>41</td>
<td>55</td>
<td>28</td>
<td>140</td>
<td>55</td>
<td>10</td>
<td>53</td>
<td>55</td>
<td>77</td>
<td>672</td>
</tr>
<tr>
<td>c21767</td>
<td>9</td>
<td>6</td>
<td>39</td>
<td>26</td>
<td>16</td>
<td>16</td>
<td>29</td>
<td>46</td>
<td>32</td>
<td>23</td>
<td>119</td>
<td>63</td>
<td>23</td>
<td>62</td>
<td>51</td>
<td>128</td>
<td>688</td>
</tr>
<tr>
<td>c23014</td>
<td>32</td>
<td>35</td>
<td>68</td>
<td>88</td>
<td>29</td>
<td>43</td>
<td>60</td>
<td>76</td>
<td>77</td>
<td>57</td>
<td>143</td>
<td>102</td>
<td>83</td>
<td>74</td>
<td>109</td>
<td>169</td>
<td>1245</td>
</tr>
<tr>
<td>23080</td>
<td>45</td>
<td>13</td>
<td>146</td>
<td>77</td>
<td>19</td>
<td>55</td>
<td>92</td>
<td>133</td>
<td>138</td>
<td>75</td>
<td>362</td>
<td>201</td>
<td>78</td>
<td>157</td>
<td>176</td>
<td>327</td>
<td>2094</td>
</tr>
</tbody>
</table>
Table.out in excel
<table>
<thead>
<tr>
<th>AA</th>
<th>AT</th>
<th>AG</th>
<th>GG</th>
<th>GT</th>
<th>GC</th>
<th>CA</th>
<th>CT</th>
<th>CG</th>
<th>CC</th>
<th>Sum of Squares</th>
<th>SCRRT of (error^2)</th>
</tr>
</thead>
<tbody>
<tr>
<td>7.5962</td>
<td>5.041</td>
<td>2.6353</td>
<td>2.4148</td>
<td>5.4637</td>
<td>0.0055</td>
<td>0.0055</td>
<td>0.0055</td>
<td>0.0055</td>
<td>0.0055</td>
<td>9.8816</td>
<td>0.0200079</td>
</tr>
<tr>
<td>0.006667</td>
<td>3.3116</td>
<td>0.004316</td>
<td>3.3976</td>
<td>0.0027792</td>
<td>0.0012107</td>
<td>0.005047</td>
<td>0.0037488</td>
<td>0.0019892</td>
<td>0.0039406</td>
<td>6.2476606</td>
<td>0.0152079</td>
</tr>
<tr>
<td>4.0866</td>
<td>0.0018194</td>
<td>0.0018348</td>
<td>2.42960</td>
<td>0.03676</td>
<td>0.0018206</td>
<td>0.0018206</td>
<td>0.0018206</td>
<td>0.0018206</td>
<td>0.0018206</td>
<td>0.0018206</td>
<td>0.0018206</td>
</tr>
<tr>
<td>4.9355</td>
<td>8.8887</td>
<td>1.21937</td>
<td>1.90263</td>
<td>0.0019892</td>
<td>0.0019892</td>
<td>0.0019892</td>
<td>0.0019892</td>
<td>0.0019892</td>
<td>0.0019892</td>
<td>0.0019892</td>
<td>0.0019892</td>
</tr>
<tr>
<td>1.2019</td>
<td>5.0007</td>
<td>0.0010443</td>
<td>5.00009</td>
<td>0.0010443</td>
<td>0.0010443</td>
<td>0.0010443</td>
<td>0.0010443</td>
<td>0.0010443</td>
<td>0.0010443</td>
<td>0.0010443</td>
<td>0.0010443</td>
</tr>
<tr>
<td>4.9355</td>
<td>8.8887</td>
<td>1.21937</td>
<td>1.90263</td>
<td>0.0019892</td>
<td>0.0019892</td>
<td>0.0019892</td>
<td>0.0019892</td>
<td>0.0019892</td>
<td>0.0019892</td>
<td>0.0019892</td>
<td>0.0019892</td>
</tr>
<tr>
<td>1.2019</td>
<td>5.0007</td>
<td>0.0010443</td>
<td>5.00009</td>
<td>0.0010443</td>
<td>0.0010443</td>
<td>0.0010443</td>
<td>0.0010443</td>
<td>0.0010443</td>
<td>0.0010443</td>
<td>0.0010443</td>
<td>0.0010443</td>
</tr>
</tbody>
</table>

Table.out in excel
Illustration of a biased random walk
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. This reasoning (with many more details) is known as the modern synthesis.

Note: this is not in contradiction to 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
• Mutationism, hopeful monsters (see here for a critical discussion by Arlin Stolzfus)
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.
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 \textbf{generated} in diploid population size $N$: $u*2N$

Probability of \textbf{fixation} for each allele = $1/(2N)$

Substitution rate (the rate with which mutations are fixed in a lineage) =

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

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

Therefore:

\textbf{If $f s=0$, the substitution rate is independent of population size, and equal to the mutation rate} !!!!

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

\textbf{Fixation time due to drift alone:}

$t_{av} = 4*N_e$ generations

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

$N_e = n/(1/N_1 + 1/N_2 + \ldots + 1/N_n)$
\( s > 0 \)

Time till fixation on average:
\[ t_{av} = \frac{2}{s} \ln (2N) \text{ generations} \]
(also true for mutations with negative “s”! discuss among yourselves)

E.g.: \( N = 10^6 \),
- \( s = 0 \): average time to fixation: \( 4 \times 10^6 \) 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

\( \Rightarrow \) substitution rate of mutation under positive selection is larger than the rate with which neutral mutations are fixed.
Random Genetic Drift

Selection

advantageous

disadvantageous

Allele frequency

Modified from from 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

Modified from from www.tcd.ie/Genetics/staff/Aoife/GE3026/GE3026_1+2.ppt
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 ...
```

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
```

Modified from 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

Modified from www.tcd.ie/Genetics/staff/Aoife/GE3026/GE3026_1+2.ppt
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

Mutant allele
(Montreal 440 Family)

4 base pair deletion
Causes frameshift

Stop codon

Modified from 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 \( d_n \)
Genetic Code – Note degeneracy of 1\textsuperscript{st} vs 2\textsuperscript{nd} vs 3\textsuperscript{rd} position sites

<table>
<thead>
<tr>
<th>1\textsuperscript{st} Position</th>
<th>2\textsuperscript{nd} Position</th>
<th>3\textsuperscript{rd} Position</th>
</tr>
</thead>
<tbody>
<tr>
<td>UUU UUC phenyl alanine</td>
<td>UCU UCC UCA UCG serine</td>
<td>UAU UAC tyrosine</td>
</tr>
<tr>
<td>UUA UUG leucine</td>
<td>UUC UCC UCA UCG serine</td>
<td>UAA UAG stop</td>
</tr>
<tr>
<td>CUU CUC CUA CUG leucine</td>
<td>CCC CCA CCG proline</td>
<td>CAU CAC histidine</td>
</tr>
<tr>
<td>CAG arginine</td>
<td>CAU CAC histidine</td>
<td>CAA CAG glutamine</td>
</tr>
<tr>
<td>AUG methionine</td>
<td>AAA AAG lysine</td>
<td>AGU AGC serine</td>
</tr>
<tr>
<td>AUG methionine</td>
<td>AAA AAG lysine</td>
<td>AGA AGG arginine</td>
</tr>
<tr>
<td>GUU GUC GUA GUG valine</td>
<td>GCU GCC GCA GCG alanine</td>
<td>GAU GAC aspartic acid</td>
</tr>
<tr>
<td>GAU GAC aspartic acid</td>
<td>GAA GAG glutamic acid</td>
<td>GGU GGC GGA GGG glycine</td>
</tr>
</tbody>
</table>
Genetic Code

Four-fold degenerate site – Any substitution is synonymous

From: mentor.lscf.ucsb.edu/course/spring/eemb102/lecture/Lecture7.ppt
Two-fold degenerate site – Some substitutions synonymous, some non-synonymous
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 $\#s/\#a$

<table>
<thead>
<tr>
<th>Species1</th>
<th>TGA</th>
<th>TGC</th>
<th>TGT</th>
<th>TGT</th>
<th>TGT</th>
<th>TGT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Species2</td>
<td>TGT</td>
<td>TGT</td>
<td>TGT</td>
<td>TGT</td>
<td>GGT</td>
<td></td>
</tr>
</tbody>
</table>

$\#s = 2$ sites

$\#a = 1$ site

$\#a/\#s=0.5$

To assess selection pressures one needs to calculate the rates ($K_a$, $K_s$), 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.
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 straightforward, 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 Ka and Ks distances from codon based aligned sequences.
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.
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 \texttt{CodonFreq}). Under this model, the relationship holds that \( \omega = \frac{d_N}{d_S} \), 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 aproaches 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;
mcmcp samplefreq=500 printfreq=500;
mcmc ngen=500000;
sump burnin=50;
sumpt burnin=50;
end;
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.
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)?
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 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 average probability

**Microsoft Excel - Book1**

![Excel formula](image1)

<table>
<thead>
<tr>
<th>BR</th>
<th>BS</th>
<th>BT</th>
<th>BU</th>
<th>BV</th>
<th>BV</th>
</tr>
</thead>
<tbody>
<tr>
<td>199</td>
<td>0.020322</td>
<td>0.025016</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>200</td>
<td>0.018418</td>
<td>0.028381</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>201</td>
<td>0.018418</td>
<td>0.028381</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>202</td>
<td>0.018418</td>
<td>0.028381</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>203</td>
<td>0.018418</td>
<td>0.028381</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>204</td>
<td>average</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

**Plot row**

**Copy paste formula**

**Enter formula**
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 way 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:

```
5   855   I
human  goat-cow  rabbit  rat
marsupial

1
GTG  CTG  TCT  CCT  GCC  GAC  AAG  ACC  AAC  GTC  AAG  GCC  CTG  GAG  AGG  ATG  TTC  CTG  TCC  TTC  CCC  ACC  ACC  AAG

```

```
6    467
61
```

```
PKPRYADYLPGFDADGSVLDQGIALWFPGPNSFTGEDVLELQGHGGPVIL
PKPRYADYLPGFDVDSGLDQGIALYFPGPNSTFTGEDVLELQGHGGPVIL
LRPRYAEYLPFTTEDGQQLDQGIALFFNPNSFTGEDVLELQGHGGPVVM
LKNPRHAHYGFLDAGGQVLDEGSLYFGPNSTFTGEDVLELQGHGGPVVL
LQSRHARYARFRDAQGEVIDDGIAVWFAPNSFTGEVVELQGHGSPVLL
LRPRYAHYTRFLDQDEVIDGGLALWFPAPNSFTGEDVLELQGHGSPVLL
```
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

dS -tree

dN -tree
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/
Results of an analysis using the SLAC approach

<table>
<thead>
<tr>
<th>Codon</th>
<th>dN-dS</th>
<th>Normalized dN-dS</th>
<th>p-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>45</td>
<td>2.80935</td>
<td>1.57283</td>
<td>0.171148</td>
</tr>
<tr>
<td>51</td>
<td>2.94546</td>
<td>1.64923</td>
<td>0.109144</td>
</tr>
<tr>
<td>65</td>
<td>2.62064</td>
<td>1.46734</td>
<td>0.197579</td>
</tr>
<tr>
<td>70</td>
<td>3.37001</td>
<td>1.88693</td>
<td>0.124660</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Codon</th>
<th>dN-dS</th>
<th>Normalized dN-dS</th>
<th>p-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>-2.5</td>
<td>-1.33979</td>
<td>0.111111</td>
</tr>
<tr>
<td>9</td>
<td>-4.5</td>
<td>-2.51963</td>
<td>0.0178326</td>
</tr>
<tr>
<td>19</td>
<td>-5.94249</td>
<td>-3.32728</td>
<td>0.0243467</td>
</tr>
<tr>
<td>22</td>
<td>-2.5</td>
<td>-1.33979</td>
<td>0.111111</td>
</tr>
<tr>
<td>41</td>
<td>-2.84041</td>
<td>-1.59039</td>
<td>0.193214</td>
</tr>
<tr>
<td>48</td>
<td>-2.45744</td>
<td>-1.37597</td>
<td>0.0793724</td>
</tr>
<tr>
<td>59</td>
<td>-4.95667</td>
<td>-2.78093</td>
<td>0.0236379</td>
</tr>
<tr>
<td>60</td>
<td>-2.96058</td>
<td>-1.65768</td>
<td>0.108898</td>
</tr>
<tr>
<td>66</td>
<td>-2.51831</td>
<td>-1.41004</td>
<td>0.15211</td>
</tr>
<tr>
<td>71</td>
<td>-2.45417</td>
<td>-1.37413</td>
<td>0.129462</td>
</tr>
<tr>
<td>72</td>
<td>-2.31427</td>
<td>-1.2958</td>
<td>0.162177</td>
</tr>
<tr>
<td>75</td>
<td>-5.47043</td>
<td>-3.06299</td>
<td>0.038673</td>
</tr>
<tr>
<td>86</td>
<td>-2.54472</td>
<td>-1.42483</td>
<td>0.151309</td>
</tr>
</tbody>
</table>

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 β-sheets (panel II) and 2 α-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/
Example testing for dN/dS in two partitions of the data -- John’s dataset

Set up two partitions, define model for each, optimize likelihood
Example testing for dN/dS in two partitions of the data -- John’s dataset

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

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

<table>
<thead>
<tr>
<th>Parameter ID</th>
<th>Value</th>
<th>Constraint</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tree_1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Tree_12</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BEGINNING_Shared_AC</td>
<td>0.187238</td>
<td></td>
</tr>
<tr>
<td>BEGINNING_Shared_CT</td>
<td>0.891995</td>
<td></td>
</tr>
<tr>
<td>BEGINNING_Shared_R</td>
<td>0.19809</td>
<td></td>
</tr>
<tr>
<td>END_Shared_AC</td>
<td>0.126137</td>
<td></td>
</tr>
<tr>
<td>END_Shared_CT</td>
<td>0.770683</td>
<td></td>
</tr>
<tr>
<td>END_Shared_R</td>
<td>0.19809</td>
<td>BEGINNING</td>
</tr>
<tr>
<td>Tree_1.AB256718.synRate</td>
<td>0.309711</td>
<td></td>
</tr>
<tr>
<td>Tree_1.AF001672.synRate</td>
<td>0.0364501</td>
<td></td>
</tr>
<tr>
<td>Tree_1.AF009898.synRate</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Tree_1.AF055424.synRate</td>
<td>0.0414451</td>
<td></td>
</tr>
<tr>
<td>Tree_1.AF074267.synRate</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Tree_1.AF074279.synRate</td>
<td>0.131262</td>
<td></td>
</tr>
<tr>
<td>Tree_1.AF084286.synRate</td>
<td>0.0419524</td>
<td></td>
</tr>
<tr>
<td>Tree_1.AF144307.synRate</td>
<td>0.129191</td>
<td></td>
</tr>
<tr>
<td>Tree_1.AF256183.synRate</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Tree_1.AF256188.synRate</td>
<td>0.0415364</td>
<td></td>
</tr>
</tbody>
</table>

Name and save as Null-hyp.
Example testing for dN/dS in two partitions of the data -- John’s dataset

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 \( \text{dN/dS} \) in two partitions of the data -- John’s dataset

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

---

**Model 1**

**Model 2**
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.