Project III: Neutral versus Adaptive Evolution

Learning Objectives

The student will be able to

- build a model for the null hypothesis of a statistical test

Knowledge and Skills

- Hypothesis testing
- Neutral versus adaptive evolution

Prerequisites

- Familiarity with functions and macros in Excel

Source: Lieberman, T.D. *et al.* Parallel bacterial evolution within multiple patients identifies candidate pathogenicity genes. *Nature Genetics* 43: 1174-1176 doi:10.1038/ng.1011 (http://www.nature.com/ng/journal/v43/n12/full/ng.997.html)

A retrospective study of 112 isolates from 14 individuals with cystic fibrosis collected during an outbreak of *Burkholderia dolosa* over 16 years identifies genetic mutations.

Task: Build a model to decide whether the observed mutations are consistent with neutral evolution.

Citation: Neuhauser, C. Project III: Neutral versus Adaptive Evolution.
Created: January 1, 2012 **Revisions:**

Mutations in the Genome

The genome of *Burkholderia dolosa* can be found on the webpage of the

Burkholderia dolosa Sequencing Project, Broad Institute of Harvard and MIT:
http://www.broadinstitute.org/

The *B*. *dolosa* genome has 5,014 genes. Lieberman et al. (2011) observed 561 independent mutational events in 304 genes. They found that seventeen genes had three or more different mutations, and four genes had more than ten different mutations.

Test whether their observation is consistent with neutral evolution.

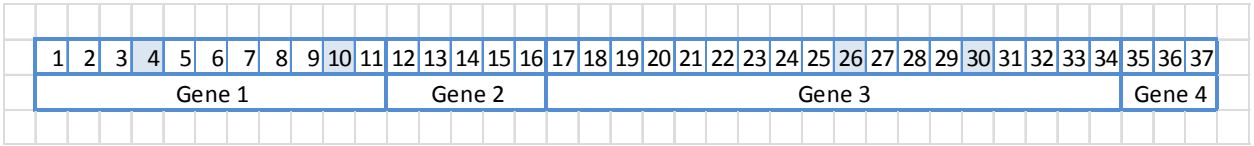
Step 1: Formulate the hypothesis

Hypothesis: The *B*. *dolosa* genome evolved under neutrality.

The hypothesis implies that we would expect the mutations to be randomly distributed among the genes. We can then ask how many genes would we expect to have 0, 1, 2, … mutations under neutral evolution. To answer this question, we build a model that mimics neutral evolution.

Step 2: Build a model of neutral evolution

We think of the genome as a sequence of nucleotides, labeled by location from 1 to the length of the genome, and assume that each nucleotide has the same likelihood of being mutated. We can simulate the 561 mutations by randomly choosing 561 sites. Since we are only interested in the number of mutations per gene, we only need to keep track which of the 5,014 genes are mutated and how often each one is mutated.



Above is an example of four genes. Each nucleotide location is labeled. The shaded boxes indicate sites with mutations. That is, gene 1 has two mutations at locations 4 and 10; gene 2 has no mutations; gene 3 has two mutations at locations 26 and 30; and gene 4 has no mutations. To mimic neutral evolution, we chose four of the 37 sites in the above genome at random, and then identify which genes each of the four sites belong to in order to determine the number of mutations per gene. For instance, if we picked the numbers 4, 10, 26, and 30, we would identify that two of the mutations belong to gene 1 and two mutations to gene 3.

Citation: Neuhauser, C. Project III: Neutral versus Adaptive Evolution.
Created: January 1, 2012 **Revisions:**

The *B*. *dolosa* genome is much larger than the 561, the number of mutations. It is therefore unlikely that the same site is picked twice, even if we allowed sites to be picked repeatedly. We can therefore simplify the model and successively pick 561 sites without checking whether a site was previously already chosen, and then determine which gene each site belongs to. This requires that we know the length of each of the 5,014 genes of the *B*. *dolosa* genome. The *B*. *dolosa* genome can be downloaded from the Broad Institute's website: http://www.broadinstitute.org/

The following Matlab script reads in the genome and determines the length of each gene.

```
%read in data
bd1g=fastaread('burkholderia_dolosa_1_genes.txt');
ngene=5014;

%calculate the length of each gene
le=zeros(ngene,1);
for i=1:ngene
    le(i,1)=length(bd1g(i).Sequence);
end
sle=sum(le,1);
```

The length of each sequence is also copied into the first column of the spreadsheet burksim.xlsm (tab Simulation), starting with Cell A3. This spreadsheet is macro enabled, which is a feature that we will need later on when we run simulations.

As in the small example we considered earlier, we imagine all genes arranged consecutively, and each nucleotide labeled from 1 to the length of the genome. To find the location where each gene "ends," we add the lengths of the genes. For instance, the first gene starts at location 1 and ends at location 1266. The second gene, which is of length 999, thus starts at location 1267 and ends at location 1266+999=2265. The third gene, which is of length 372, then starts at location 2266 and ends at location 2265+372=2637, and so on. Determine the locations of the endpoints of all genes and enter them in Column B, starting with Cell B3. In Cell B2, enter a "1" to indicate that first location.

In Column D, we entered the numbers 1 to 561 for the 561 mutations. In Column E, rows 2-562, we enter into each cell

=INT(RAND()*$B$5016)+1

This function will choose a random number between 1 and the length of the genome, stored in Cell B5016. You can enter this command into Cell E2 and then drag the cell down to Cell E562. Each time you hit the F9 key, the numbers change, indicating a new selection of sites where mutations occur.

Citation: Neuhauser, C. Project III: Neutral versus Adaptive Evolution.
Created: January 1, 2012 **Revisions:**

The next step is to identify which gene each mutation belongs to. Excel has a function, called MATCH, that searches for a specified item in a range of cells and returns the relative position of that item in the range. The syntax is

MATCH(lookup_value, lookup_array, [match_type])

The match_type argument can be -1, 0, or 1. The value 1 indicates that MATCH will look for the largest value that is less than or equal to lookup_value. The values in the lookup_array must be in ascending order. If you enter into Cell F2

=MATCH(E2,$B$2:$B$5016,1)

It will identify the relative position in the array B2:B5016 that is the largest value less than or equal to the randomly chosen location in Cell E2. For instance, if the value in Cell E2 is 2367, then the MATCH function will return 3, indicating that location 2367 is in gene 3. Drag the content of Cell F3 down to F562.

We can now count the number of times each gene was hit using the function COUNTIF. To do this, enter the numbers 1 to 5,014 into the cells H2:H5015. Enter into Cell I2

=COUNTIF($F$2:$F$562,H2)

to count the number of times gene 1 was hit by a mutation. Drag this cell down to Cell I5015. Using the COUNTIF function again, we can then count how many genes were hit 0 times, 1 time, 2 times, and so on. Enter the numbers 0 to 20 into Cells K2-K22. Enter into Cell L2

=COUNTIF($I$2:$I$5015,K2)

Drag the cell down to Cell L22. If you hit the F9 key repeatedly, you can see how many genes are hit 0 times, 1 time, 2 times, and so on.

Repeat this several times and copy the result into the spreadsheet. To simplify this task, write a macro.

Step 3: Test the Null Hypothesis

The Null Hypothesis of neutral evolution can now be tested. It was observed that seventeen genes had three or more mutations. By repeatedly running the simulation, we can check how many genes are expected to have three or more mutations. For instance, in twenty simulations, we observed the following distribution of finding three or more mutations in 1, 2, 3, 4, or 5 genes:

Citation: Neuhauser, C. Project III: Neutral versus Adaptive Evolution.
Created: January 1, 2012 **Revisions:**

Number of Genes	Number of Simulations
1	6
2	5
3	3
4	5
5	1

Write a macro to determine under the null hypothesis the distribution of the number of simulations that show three or more mutations in *x* genes, *x*=0,1,2..., and use this to decide whether the observation in Lieberman et al. (2011) is consistent with neutral evolution.

Citation: Neuhauser, C. Project III: Neutral versus Adaptive Evolution.
Created: January 1, 2012 **Revisions:**

Citation: Neuhauser, C. Project III: Neutral versus Adaptive Evolution.
Created: January 1, 2012 **Revisions:**