

The β -Globin Gene Exercise

Extensions

You can suggest the following activities to your really motivated students who seem to be interested in *bioinformatics*:

- Challenge your students to think of how they could develop a computer algorithm that would automatically search through a nucleotide sequence to find a gene. Remind them that Microsoft Word will allow them to set up *macros* to find words in documents. Your students

If you cannot speak of it with numbers, then your knowledge of the subject is of a trivial and unsatisfying type. — Lord William Thomson Kelvin, British Physicist

should determine what features to look for when identifying the beginning of a gene, and what feature to look for when identifying an exon-intron junction.

You or your students could contact your local research university and ask for a *bioinformaticist* who would visit your class (or host your class at his or her lab). Ask the bioinformaticist to explain how the algorithms in their software work to find genes.

- We should always be looking for opportunities to have our students calculate and measure things. Here is an interesting calculation dealing with the average length of an **ORF (Open Reading Frame)** in a non-coding region of DNA.

Imagine that a monkey is sitting at a keyboard with only four keys (ATGC) and is randomly entering a nucleotide sequence. Then if you translate that nucleotide sequence into amino acids using the familiar triplet codons, what would be the average length of the ORFs in your sequence? Another way to ask this question is, What is the average number of amino acids between STOP codons?

One group of students could take a theoretical approach to this problem and calculate a number based on the statistical frequency at which the monkey will type one of the three STOP codons¹. Another group of students could take an experimental approach and manually count and calculate the average number of amino acids between the STOP codons in all of the non-coding reading frames of the β -globin gene.

Do the two approaches result in similar answers?

Can your students propose an approach to develop the computer algorithm to find a gene?

¹**Calculating how frequently a monkey will type a STOP codon:** In a given reading frame, the chance that the monkey will type a U at the start of a triplet code is $\frac{1}{4}$. The chance that the monkey will type a UA is $\frac{1}{4} \times \frac{1}{4}$, or $\frac{1}{16}$. The chance that the monkey will type UAA is $\frac{1}{4} \times \frac{1}{4} \times \frac{1}{4}$, or $\frac{1}{64}$. Therefore, if you let the monkey type 64 triplet codons, statistics suggest that one of the codons will be UAA. Find a monkey and try this. But, since there are three stop codons (UAA, UAG, and UGA), the chance of finding ANY stop codon in this 64 codon sequence is $\frac{3}{64}$, or approximately once every 21 codons. Thus, you would predict that the average size of an open reading frame (ORF) in a completely random, non-coding region of DNA would be ~21 codons. Compare this statistical prediction with the average number of amino acids between STOP codons in Intron II (non-coding) of the β -globin gene.

