

Finding a Gene in DNA

What is a String	2 min
Understanding Strings	3 min
Developing an Algorithm	5 min
Positions in Strings	8 min
Translating into Code	11 min
Java Math	8 min
Programming Exercise: Finding a Gene and Web Links	10 min
Practice Quiz: Finding a Gene in DNA	6 questions

Finding All Genes in DNA

Debugging Code

Using the StorageResource Class

Review

BlueJ

Class

Edit

Tools

Options

BlueJ: FindGene

FindGeneSimpleAndTest - FindGene

Source Code

```
/**
 * Write a description of FindGeneSimpleAndTest here.
 *
 * @author (your name)
 * @version (a version number or a date)
 */
public class FindGeneSimpleAndTest {

    public String findGeneSimple(String dna) {
        // start codon is "ATG"
        // stop codon is "TAA"
        String result = "";

        return result;
    }
}
```

We're going to do a very simple algorithm in order to find our gene.

Translating into Code

Have a question? Discuss this lecture in the week forums.

Interactive Transcript

Search Transcript

English

0:03
Hi, we're going to find a gene in a DNA strand. [We're going to do a very simple algorithm in order to find our gene.](#) So I've started some code here. We have an algorithm method, called findGeneSimple.

0:21
And we haven't done much with it yet, but what I've also done down here is I've written a couple of DNA strands that we will use to test the code that we write. So you can see here we have String dna. We're going to print our DNA stand. We're going to call findGeneSimple and then print out our gene. And then I just have done this with four different strands of DNA.

0:47
So let's write the code now.

0:52
So I've started by just saying, the resulting gene is just going to be called a variable called result. And I've set it to null, the null string. And if you remember from the video, we are going to look for the start codon in the strand of DNA. And the start codon is ATG, the string ATG. So we can use the new string functions that we learned about. So I'm going to start by creating a variable that will hold the index position of the start codon. So we'll call it startIndex, and then we are going to look in the strand of DNA and we'll use the indexOf function to look for ATG.

1:50
And the indexOf function is going to go through the string and it's going to stop when it finds ATG and it's going to return the index location of where it starts. So that'll be the starting position of our gene. We also have to look for the stop codon, and that is TAA. So we're going to create a variable for that called stopIndex,

2:15
For the position of where the stop codon is. Again we will look in our DNA strand, DNA and we'll use indexOf. But if you just look for TAA, it will start at the beginning of the string. And we want to look for the stop codon after the start codon. So we know where the start codon is. It's in the variable startIndex. And we want to start looking past that. So you can add a second parameter to the indexOf function, and so we will add that and say, start looking where ATG is, which is startIndex.

3:00
+3, which is the length of ATG.

3:09
So now we have the startIndex and the stopIndex, startIndex of ATG and the stopIndex of TAA. And now we want the strand that includes those two and everything in between it. So I'm going to call that our result.

3:26
Again we'll use a string function. So we'll use the string function called substring.

3:34
And the way substring works is you have to say I want a piece of a string and I want to start, where do we want to start? We want to start where ATG is, so we'll start at the startIndex and then the rest of our gene is going to be everything past ATG up until TAA, which is at the stopIndex.

3:58
But we also want to include TAA so we'll add 3.

4:03
So that means take our substring from where ATG starts, go all the way to the stopIndex +3, which is the first character after TAA. And the way substring works is it says start at the first StartIndex and then go all the way up to, but not including, the second parameter. And then let's compile this and see if it works.

4:33
Okay, so it compiles fine.

4:36
So let's go over here, and we'll create a new findGene.

4:45
And now we'll run our test method that we wrote.

4:49
And there it is. So we can see, here's the first DNA strand. And you look for ATG, there it is. And then look for TAA and there's the first TAA that's after it. And you can see there's the gene that we found, right here. And then the next example here's ATG. We look for TAA all the way here. And you can see here that's the gene we found. Here's the third example, ATG all the way to TAA. You can see that we've got a big gene there. And then here's the last example. ATGTAA so that works good too.

5:27
But what happens if ATG is not there? Or what happens if TAA is not there? So let's look at our example here, and let's change one of our strings.

5:42
Let's change this string.

5:45
We'll just save this one.

5:47
And let's change it to,

5:59
A string that does not have ATG in it. Let's see what happens if we run that example.

6:06
So we're going to come back over here.

6:13
And we'll run our testFind, and we got an error down here. You can see, it says StringIndexOutOfBoundsException, - 1. String index out of range. What happened? Let's go look at our code.

6:28
So we looked for ATG and we didn't find it. And what the indexOf function does if it doesn't find the string you are looking for, it returns -1. So startIndex has the value -1 and then we are trying to build a string starting at -1 and -1 doesn't exist. So we got an error. So what are we going to do? We need to fix this code.

6:51
So let's check right after we ask for ATG, let's put an if statement in there to check to make sure that ATG was there.

7:02
So we can ask if startIndex is equal to -1, that means there is no ATG.

7:15
In that case, there is no gene. There's no gene if there's no ATG, there's no start. So what we'll do is we'll just return the empty string.

7:32
So if you compile this now and if we run it.

7:44
So we'll create findGene.

7:47
And then we'll run testFind, and it ran, but there's no gene here. That's because there's no ATG. Let's try another example.

7:59
We'll change our code again to give another example.

8:05
Let's just change this one here.

8:09
Let's create a string where we have ATG but we don't have TAA.

8:14
So we have the start codon, but not the stop codon.

8:19
So I'll create a string, CGATGGTTTAAAAGT. So there's no, there is, let's get rid of it. G, there we go. So now we have ATG right here and to the right of it there is no TAA. So let's compile and we'll run it.

8:57
And you can see it's not there.

9:02
There's no gene for the second one because ATG, but there's no TAA. All right, so just like we put a check for if there's no ATG, we should do the same for if there's no stop codon, so no TAA. So let's add that code, too.

9:22
We would add it right after we get the value for stop codon.

9:28
So right here we can check and see if stop.

9:35
If the stopIndex equals -1, at that point we know there's no stop codon and so there's no gene. And so we can again, just return the empty string.

9:54
Return, Empty string. So that's just a safe way, if there's no start codon, or no stop codon. Let's add a comment here. This is a case for no TAA. We'll compile our code and we'll just double check and make sure that works.

10:22
So it compiles and we'll run it.

10:31
And now we have, our first string is good, we find a gene. Our second string, there is no gene. We have an ATG but we don't have a TAA. This string looks good.

10:44
The string has no ATG. It actually does have a TAA but that doesn't matter because it has no ATG. And so there is no gene for that one either. So anyway that is the way to find a simple gene and a strand of DNA. We look for the start codon and then if there's a start codon, we look past it for the stop codon. And if we find both of those, then we can return the start codon, the stop codon, and everything in between it as the gene. Thank you.

Downloads

Lecture Video	mp4
Subtitles (English)	WebVTT
Transcript (English)	txt

Would you like to [help us translate](#) the transcript and subtitles into additional languages?