In this lab, we're going to look at four different ways to visualize next-generation sequencing data. So, one of which is N, outside of R, and the other three, which will be NR. So we're going to use the sequencing experiment in the passilaBamSubset package, which is hosted on Bioconductor. You can install with this line above. And we also need the genes, which are annotated in the transcript database here-- which is also hosted on Bioconductor.

So the two files that we're going to look at, is specified by these two functions. So we're going to start by looking at sequencing coverage using the Java program, IGV. So, IGV is an application which is available from the Broad institute. You need to go to the Broad Institute website-- which we'll link to in this script-- and then click download. And in order to download IGV, you need to register.

So, you just need to fill out this form, and then they'll send you back the download link, and then you can download. And IGV is a Java program, so you can either run it using one of these methods here-- so either Java Web Start, or from the command line. So first, we're going to specify which files that we want to use. So, we're going to look at the files which are contained in the passilaBamSubset library. So what I'm going to do, is just copy these files from within the library, to some convenient location.

So first, you should set the working directory in our studio to the source file location. So now I'm going to copy these files-- these Bam files-- into my current working directory. So this is a call to copy on the command line, file one, into the current working directory, using the base name.

And then, we also need to index these Bam files, because they don't necessarily already have an index. Which is a way that IGV can use to pull out the coverage for an arbitrary genomic range. So, the index Bam function in the Rsamtools library. We'll then create these index files, which have a .bai ending.

So now, here's the link for downloading IGV. And using IGV, we're going to look up the gene, LGS. So, this is my downloaded version of IGV. And I'm going to use the igv.comand script, which will launch terminal, and it will open up an IGV window for me.

So here's a blank IGV process. The data we're looking at, is an RNA sequencing experiment of Drosophila. So we need to use the Drosophila melanogaster genome, and specifically the dm3 genome. So we can click here to load this. And now we want to load in the data. So we're going to load.

So now you need to navigate to whatever your working directory was. So, for me it was in the week 8 folder. And so, now here are the four files that I just created. So the Bam files, and their index. And I'm going to select just the Bam files to load into IGV. So now I have these two new tracks which are created.

So the passilaBamSubset package is a subset of the reads from the passila package, which map to chromosome 4. So let's zoom into chromosome 4. And now, after a couple seconds of loading, we have all of these reads. It's very dense, so we can't see individual reads very well. But what we can do is, we're interested in the gene LGS, so just some arbitrary gene on chromosome 4.

So you can type the gene name here, and click Go. And now it'll zoom in to a single gene, here, shown on the bottom. And here we have the reads, and here on the top we have a little preview of the coverage. So, we can see that the coverage only is on the axons, mostly. Here's some stray reads which map to the Intron.

And if we zoom in even further-- so you can hold Shift, and drag on this top part to zoom-- you can see the individual reads. So here's a read on the plus strand, here's a read on the minus strand. And these lines indicate bases which do not match the reference. So, IGV is a very useful program for visualizing reads from sequencing experiments. Mostly because you can do this very quickly.

So, a Bam file-- which has gigabytes of information-- because of the index file, IGV can go in and extract only the reads that it needs for the region of interest. And so you can actually show dozens of files at once, and compare. And look for substitutions, or for sequencing, or chip seq experiments, look at differences in coverage across files. So, in the next unit, I'm going to show how to generate coverage plots-- like the ones in here-- but within Bioconductor.