Visualizing NGS with IGV and Bioconductor

In this lab, we're going to look at different ways to visualize next-generation sequencing data. The first one is outside of R, and the other with R.

We're going to use

- the sequencing experiment in the `passilaBamSubset` package, which is hosted on Bioconductor and
- the information about the genes, which are annotated in a *transcript database* which is also hosted on Bioconductor

We're going to start by looking at sequencing coverage using the Java program, IGV.

IGV is an application which is available from the Broad institute. You need to go to the Broad Institute website-- and then click download. In order to download IGV, you need to register You just need to fill a form, and then they'll send you back the download link so that you can download IGV

IGV is a Java program, so you can either run it using Java Web Start, or from the command line

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

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

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

So here's a blank IGV process. The data we're looking at, is an RNA sequencing experiment o[Drosophila. So we need to use the Drosophila melanogaster genome, and specifically the dmN genome. So we can click here to load this. And now we want to load in the data. So we're going to loade So now you need to navigate to whatever your working directory was. So, for me it was in the week O folder. And so, now here are the four files that I just created. So the Bam files, and their index. And I'j going to select just the Bam files to load into IGV. So now I have these two

new tracks which arU createde

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

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

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

So, a Bam file-- which has gigabytes of information-- because of the index file, IGV can go in an] extract only the reads that it needs for the region of interest. And so you can actually show dozens o[files at once, and compare. And look for substitutions, or for sequencing, or chip seq experiments, looZ at differences in coverage across files. So, in the next unit, I'm going to show how to generate coveragU plots-- like the ones in here-- but within Bioconductore So continuing the visualizing sequencing lab, we're going to try to make the same coverage plot iW Bioconductor using the simple plot functionp

So first we need to load the genomic ranges package if we don't already have it loaded. So I'j assuming you have the objects loaded from the beginning of the visualizing NGS data unit. ST pasillaBAM on subset, the transcript database, and these filesp

So if you're using Bioconductor version 14, you also need to load this Bioconductor library, genomiV alignments. If you are using a previous version of Bioconductor, you don't need this because th] functions we're going to use are already in the genomic ranges package. So you only need this library io you're using the newest version of Bioconductorp

So we're going to read in the reads, so the alignments, from the file fl1. And we're going to use th] readGAlignments function to do this. So now x is an object of class GAlignments. And we can just type y and see what it contains. So each row is then a readp

And if we want to look at the coverage of these reads, so the base pairwise coverage, we can use th] coverage function. And this will then generate a RleList. So an Rle, remember from week two, is a run| length encoding And this is a list which covers these five-- even more-- chromosomes. And so if w] then extract one element of the list, we get an Rle. So a run-length encoding of integer coveragep

And so what this is telling us is that we have zero coverage for the first 891 base pairs, and then w] have coverage of one for 27 base pairs, et cetera. So this is a compressed way to represent coveragep

And let's zoom in now to range which is near this gene of interest, LGS. So I just eyeballed a rang] which covers this gene. I'm going to name this z and then we can subset the coverage, this RleLisX using zp

So as I was speaking I remembered that this ability to subset an RleList using a genomic range is onli available for Bioconductor 2.14. So the equivalent way to do this in Bioconductor 2.13 was to subset th] RleList to an Rle and then to query that Rle with the I ranges. So basically the new Bioconductor 2.1x does this for you. So it looks up in the RleList that chromosome of the ranges in zp So now we have this Rle of coverage for our area of interest, we want to turn this into a numeric. ST we've uncompressed this coverage into a long vector of numbers which we can plot. So now we caW see the base pair resolution coverage of these reads in this area of interest. So we've lost, basically, th] information here, which is that we're on chromosome 4, starting at the 456,500 base pairp

We can do the same thing for another file, so file two. We read in the pairs. So because fl2 is a paired iW sequencing experiment, we now have pairs of reads. So this range is, and these ranges are, two read_ which correspond to one fragment. So these are now pairs of readsp

And we can likewise calculate the coverage using the coverage function. And then we can repeat th] same line of code to get the numeric version of the coverage. And then we can plot the first file_ coverage in blue and the second files coverage in red to compare this coverage on top of each otherp

So now we want to zoom in to a single exon, say this exon here between in the area of 6,000 bas] pairs, we can use xlim function and plot the two coverages. So we can see that both of these hav] coverage on the same exon, although it's variablep

So we also want to be able to extract information about the genes. So in the final part of this unit I'j going to show how to extract out the information about this gene LGS. So we'll use the same function_ that were described in week sevenp

So that we're going to use the BioMart package and we'll load the drosophila ensemble gene BioMartp And so we want to look up the gene with the gene name LGS. So we're going to look for what filter_ exist for this Mart which have name as part of the name. So we're going to use the grep function tT subset the list of filtersp

So these are the filters for the BioMart which have the word name in the description. And so I think LGS would be in this flybasename_gene. So this is the name of the gene from the fly base. And we can us] the getBM function and specify we're using the drosophila BioMart. And we're going to filter on flybas] gene name and ask it what the ensembl gene ID is for the gene with name LGS. We only ask for on] gene, so it's pretty quick, and it tells us that the ensemble gene name for this name is right herep So now that we have the ensembl gene ID, we can get the exons out of the transcript database. ST here's a transcript database for the drosophila melanogaster genome. And if we use the exonsBi function, we pull out the exons for every gene into a g ranges list. And then we're going to query this [ranges list using this name here, which we got from BioMartp

So now this is the gene of interest. So this is the LGS gene and we see it has the six exons. And thes] are the same exons which we saw above right here. So one, two, three, four, five, sixp

And we can we can plot these exons. So the following three lines will open up an empty plot and theW plot each exons as an arrow. So here we can see we had a medium size exon, a short, short, longe short, long exonp

But actually, if we notice here, this gene is a minus strand gene. So we'll make the following adjustmenX to the code. So arrows function has an argument code. And we're going to say, look up th] strandedness of the first exon. If it's a plus strand, then use code=2 and that means put the arrowhean at the end here. And if it's a minus strand gene, use code=1 and that means to put an arrow at th] start. So if we run this code, now we get actually the strandedness of these exons which goes in th] minus directionp

So in the next units we're going to continue. And I'm going to show two packages for visualizin[genomic data in Bioconductor which allow you to avoid rewriting all of this messy code every time yoZ want to draw exons or coveragep So we're continuing where we left off, where we had the pasillaBamSubset, RNA-seq coverage, and thW LGS gene that we want to look at the coverage of. And we're going to show two packages whic` simplify plotting coverage in Bioconductort

So both of these packages have very extensive vignettes, which describe plotting many, many kinds ob data--more than just coverage. But we're just going to show, briefly, how to use each of thesW packages to draw the coverageт

So, first we're going to use the GVIZ package from Bioconductor. And it works like so. You set up d Genome Axis Track, using the following line, and then you can specify an Annotation Trackt

So if you have, for instance, we have this gene that we want to draw as an annotation along the bottomt So you can then create another track. And so these things now don't really do us very good, they're jus\ objects floating in our session. But then when we call the PLOTTRACKS function and supply a list, it wila automatically draw these exons we had before. So if you remember, these were our exons. And thig was done with much less effort using the package. So these arrows indicating that this is a minus strane genet

So the GVIZ package also allows you to draw data. So we can look at the coverage, for instance. So wW have the coverage already as an RleList. And in order to plot coverage using the GVIZ package, yo^ need to turn the data into a GRanges object. So, luckily, Bioconductor has this AS function, which allowg you to very easily convert objects from one to the other, if this is implemented. So we can then conver\ the XCOV coverage, which was an RleList, into a GRanges objectt

If you look, now it's converted what was coverage of zero into a range. So we have zero coverage foY the first four chromosomes, and then it says, chromosome four starts out with 891 base pairs of zerT coverage. And that was the same information that we had here, 891 base pairs of zero coverage. NoV it's just a different objectt

So we'll also convert the Y coverage, and now we can create two data tracks. So I'm only interested i[plotting, say, coverage which overlaps this range, which I specified earlier, Z. So I'm going to use thig OVER function to subset the coverage. So I only want the GRanges from this XGR object, whic` So now I have these two tracks. And, like before, I just add them into a list, and it will automatically plo\ them. So now, here are my exons, here's my genome track at the top, and then here are my two filest And they have their own scales. So the first sample goes up to 50, the second sample goes up to abovW 30t

And now it's showing a dot for each base pair of coverage. So there are many types. So you can plo\ the different data tracks using different types, and so we can just simply plot it using POLYGON, whichc instead of the dots, it draws a line up and down from zero to the coverage. So one thing here to note, ig that there was a dot here, and a dot here, and it connected them with a line. But actually, there'g probably zero coverage along this region. So that's just something to note in this plot-- that yo^ probably want this to go to zero here. But this is a pretty quick and easy way to draw the coverage, andc at the same time, the gene model at the topt

So the other package I wanted to show, and I'm just going to show these four lines here, is the GGBId package. GGBIO builds off of the GGPLOT2 package, which is a whole other way of drawing plots in Rt And we don't have

time to cover it here, but I'm just going to show you, basically, how you would havW done the same thing with GGBIOt

So GGBIO has a function called AUTOPLOT. And if you give it a Bioconductor object, it will prepare thW plot that seems most reasonable. So, if I ask it to plot the gene, it draws these big black boxes, which ig the gene model. And then at the bottom, you can see it has the kilobases. So this is kind of the genomW track at the bottomt

One thing that's nice, if you point it to this BAM file-- so now this is just a string pointing to a BAM file-n and if you say that you want to look at this BAM file over this range, it will read in the BAM file, parse thW coverage, read the alignments, and extract the information. And draw this nice plot. So that's reallu convenient. And that was many lines of code that we had above, and it's shorthand into one singlW function. And we can also do the same for file twot

So that was an extremely brief overview of these two GGBIO and GVIZ packages. But I encourage yo^ to look over the vignettes, because there's way too much options there to cover in one labt