Visualizing Genetic Variants in R using ggbio

This tutorial will cover how to do three things:

1. How to run a search using the biomaRt package to find genes that correspond to a breast cancer phenotype.
2. How to use the expanded grammar of graphics in ggbio and the autoplot feature to graph genetic data in a highly customizable manner.
3. How to import a variant call format (VCF) file using the VariantAnnotation package.

The link to the video tutorial is: https://youtu.be/Z_TX5tseW1M

The R commands used in the tutorial are found below.

Rationale

R is a powerful statistical programming language with many packages dedicated to bioinformatic analyses. R is a useful tool in the bioinformatician's toolbox since it allows for the customization of commands to the user's exact needs and running the same analysis for many files using loops. Unfortunately, R runs with the user's computing resources. If you have a computer that barely makes the specifications to run Rstudio, and you want to run variant calling on 100 different files, it will either take too long or your computer will crash. Also, R is not always intuitive, which makes it hard to learn for newer programmers. Thankfully, tutorials on specific workflows from experts exist for many biology-specific R packages and can be used to help new programmers understand the code before applying it to their own data. In this tutorial, we will be identifying genes that contribute to the breast cancer phenotype in humans using biomaRt (1). Identifying genetic variants that contribute to disease phenotypes is a key component of genomic medicine. With the increasing availability of genetic testing, identifying mutations that could increase the risk of contracting a life-threatening disease like breast cancer allows medical professionals to closely monitor at-risk individuals and catch the disease early. Monitoring measures, like yearly screenings, have been shown to increase early detection rates (4). Early detection of breast cancer, in turn, improves the 5-year survival rate of the patient, because the cancer is not given as long to spread. For breast cancer that is either localized or has only spread regionally, the 5-year relative survival rate for women tested between 2009 and 2015 was between 86% and 90%. As soon as the cancer is given the chance to spread, the survival rate plummets to 27% (5). Here, I identify ATM and BRCA1 from the list generated by the biomaRt search, and then use BRCA1 as an example gene to visualize variants using ggbio (2), an analytics package from Bioconductor.

There are several other R packages besides ggbio available to plot genomic data in R such as GenomeGraphs (3), karyoploteR (4), and RIdeogram (5), which all have slightly different approaches to plotting ideograms and other gene models. Most of these packages are a bit more involved from a coding perspective than ggbio, however, and so can be unapproachable

for newer R users. Certain packages, like karyoploteR, also specialize in certain plot types, like karyotype plots. ggbio has a convenient autoplot() function that automatically detects the type of data being passed to it and generates the most appropriate genomic plot. You can also tell it to plot the data in different ways from the autoplot() function, so you don't lose the ability to customize ggbio's graphs, but do add the convenience of a function that automatically detects which type of plot to make. ggbio works with all Bioconductor data structures, and so provides the benefit of being transferrable to all other Bioconductor packages. Subsequent steps in analysis, like using the VariantAnnotation (7) package we will use to add variants to the ideograms made in ggbio, can be easily integrated into a ggbio workflow. However, if part of your analysis pipeline cannot be done using Bioconductor data structures, you will need to convert the data into Bioconductor-ready data before being able to use ggbio. This is outside of the scope of this tutorial and will not be covered.

R Script

Note: Some packages used in this tutorial are made for R v4.0 or later. This tutorial was run using R v4.0.2.

```
#Installation instructions have been included with each library
#Installation only needs to be done once
#Accession needs to be done in every run

#Access necessary libraries
#install.packages(BiocManager)
library(BiocManager)
#install.packages(ggplot2)
library(ggplot2)
#BiocManager::install(ggbio)
library(ggbio)
#BiocManager::install(biomaRt)
library(biomaRt)
#BiocManager::install(biovizBase)
library(biovizBase)
#BiocManager::install(Homo.sapiens)
library(Homo.sapiens)
#BiocManager::install(VariantAnnotation)
library(VariantAnnotation)



#Perform a biomaRt search to determine desired genes
```

```
ensembl <- useMart("ENSEMBL_MART_ENSEMBL")
human <- useDataset("hsapiens_gene_ensembl", mart = ensembl)
GeneSelection <- getBM(attributes = "external_gene_name", filters =
"phenotype_description", values = "Breast Cancer", mart = human)

#Make Annotation Graphs for Genes we want
data(genesymbol, package = "biovizBase")
wh <- genesymbol["BRCA1"]
wh <- range(wh, ignore.strand = TRUE)
p.BRCA1 <- autoplot(Homo.sapiens, which = wh)
p.BRCA1

wh <- genesymbol["ATM"]
wh <- range(wh, ignore.strand = TRUE)
p.ATM <- autoplot(Homo.sapiens, which = wh)
p.ATM

#Import the .vcf file containing variant information
fl.vcf <- system.file("extdata", "17-1409-CEU-brca1.vcf.bgz",
package="biovizBase")
vcf <- readVcf(fl.vcf, "hg19")

#Coerce the vcf file into a VRanges option
vr <- as(vcf[, 1:3], "VRanges")

#Rename the columns of the VRanges object so that they match what we
have in wh
vr <- renameSeqlevels(vr, value = c("17" = "chr17"))

#Use autoplot on the VRanges object
p.vr <- autoplot(vr, which = wh)

#Display the autoplotted function
p.vr

#Create a genomic range (GRanges) that we can use with autoplot to
zoom in
gr17 <- GRanges("chr17", IRanges(41234400, 41234530))

#Zoom in on the desired genomic range
p.vr + xlim(gr17)
```

```r
#Zoom in even more
p.vr + xlim(gr17) + zoom()

#Using a custom geometry
autoplot(vr, which = wh, geom = "rect", arrow = FALSE)
```

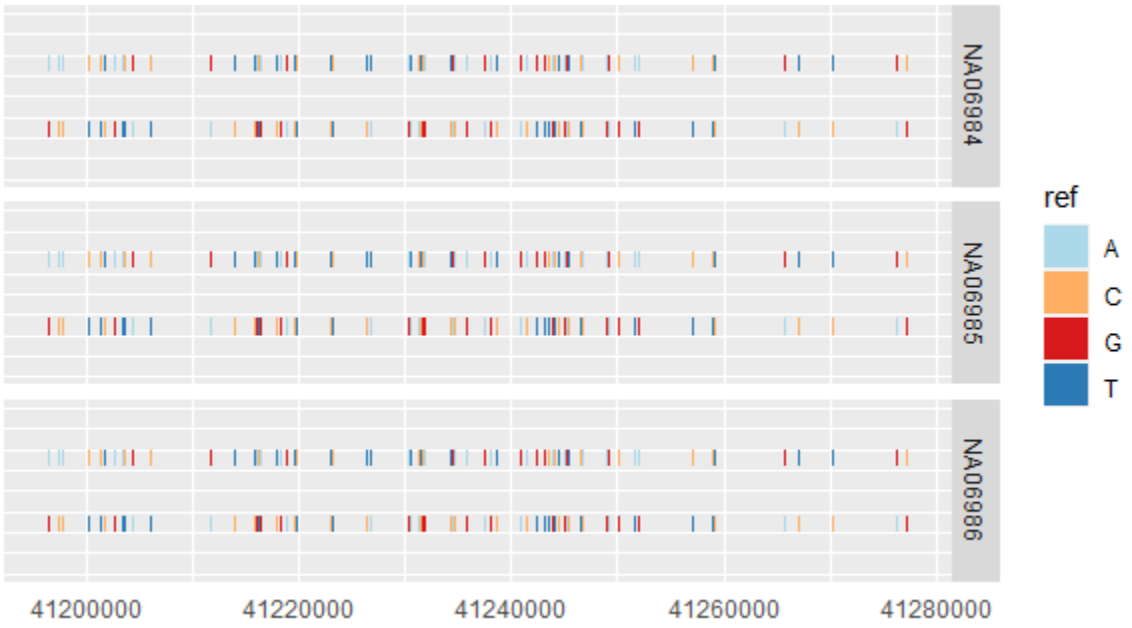

Figure 1: Custom-Geometry Variants Track: X-axis represents genomic range (bp). Y-axis contains variant information for 3 individuals (vertical axis, right) from the 1,000 genomes project with known BRCA1 mutations as compared to the GRCh38 reference assembly. Reference bases at each mutation locus are listed as the bottom track for each individual, and the set of mutations for each individual is the top track.

References

1. Durinck S, Spellman P, Birney E, Huber W (2009). “Mapping identifiers for the integration of genomic datasets with the R/Bioconductor package biomaRt.” *Nature Protocols*, **4**, 1184–1191.
2. Tengfei Yin, Dianne Cook and Michael Lawrence (2012): ggbio: an R package for extending the grammar of graphics for genomic data. Genome Biology 13:R77
3. Durinck S and Bullard J (2016). *GenomeGraphs: Plotting genomic information from Ensembl*. R package version 1.34.0.
4. Gel B, Serra E (2017). “karyoploteR : an R / Bioconductor package to plot customizable genomes displaying arbitrary data.” *Bioinformatics*, **33**(19), 3088-3090. doi: 10.1093/bioinformatics/btx346.
5. Hao Z, Lv D, Ge Y, Shi J, Weijers D, Yu G, Chen J. 2020. RIdeogram: drawing SVG graphics to visualize and map genome-wide data on the idiograms. PeerJ Comput. Sci. 6:e251 http://doi.org/10.7717/peerj-cs.251
6. **Survival Rates for Breast Cancer** *American Cancer Society*.
7. Obenchain V, Lawrence M, Carey V, Gogarten S, Shannon P, Morgan M (2014). “VariantAnnotation: a Bioconductor package for exploration and annotation of genetic variants.” *Bioinformatics*, **30**(14), 2076-2078. doi: 10.1093/bioinformatics/btu168.