Ethan O’Keefe
Gen 711
5/13/2022
Gen 711 Final Project Report

Identifying Genetic Variance Linkage to Increased Alcohol Consumption in Wistar Rats

Background

I do drug addiction research at UNH with Dr. Charntikov in the psychology department. In these investigations we often use animal models such as rats to measure behavioral economic demand and quantify addictive propensity. Our current study is attempting to identify individuals susceptible to alcohol abuse. Our study is using Wistar rats as they are bred with a higher propensity to consume alcohol, making it easier to habituate them to the generally unenjoyable taste of ethanol[1]. Our study is focusing on variation amongst certain behaviors and neuronal activation. In an attempt to understand the Wistar's inbred increase in consumption propensity, I have decided to identify variance within its genes previously linked to alcohol abuse as a means of correlating the phenotypical consumption to genotypic factors that could be analyzed further in future studies. A project pipeline can be found in the .readme file of this github page.

Script / Troubleshooting References

The majority of this project was completed using command line WSL and its supplementary tools. The script used for this project contains a combination of used, unused, and troubleshooting commands, to avoid confusion as to which parts were used for what, I will refer to each relevant command by the line it can be found on in the master script on Github as such; command$^{ln\#}$. There is a troubleshooting appendix at the end of this paper which will detail referenced portions of the project that encountered significant troubleshooting, and will detail the methods and eventual success or failure of the attempts. Instances that encountered such roadblocks will be referenced as such; Process$^{tr\#}$

Genome Collection

The first step in this project was to obtain a Wistar rat genome, and a reference Brown rat genome. The Brown rat was chosen for its neutral (unselected for) propensity to consume, meaning it should do well as a reference to illuminate any significant variation in the Wistar genome. The most up to date, widely used rat reference genome, the BN7.2 genome, was assembled from Brown rat sequences, making it easily accessible and very thorough. It was downloaded from NCBI[2]. The Wistar sequences were gathered from the European Nucleotide Archive[3] where they were submitted by a project performing a comparative analysis on GK and Wistar rats[4]. The project had uploaded 80 fastq reads total, 20 forward and reverse for each the

GK and Wistar rats. A series of corresponding forward and reverse Wistar reads were downloaded as fastq.gz files. The files were decompressed using gzip[tr1, ln20-44]

```
'gzip -d *.fastq.gz'
```

Resulting decompressed files were readable as text based fastq files and were used for sequence alignment.

FastQC

A fastQC analysis was performed to determine base call quality, and if the sequences contained enough high quality reads to be used in an analysis. The software was downloaded from Babraham Bioinformatics[5] as an executable .jar file that ran independently from the command line in its own browser window. To analyze call quality, 8 forward reads were concatenated and run through the software.

```
'cat fastq1 fastq2 fastq3… > assembled fastq1-8.fastq'
```

This was done to ensure reported call quality was representative of the entirety of the sequenced reads, and not biased by the particularly high or low quality sample used. The program reported average individual base quality across all the reads (Figure 1) , and overall read quality distribution (Figure 2). Both were found to be a satisfying quality, with the majority of bases, and reads, being above the 28-32 phred score required for confident analysis.

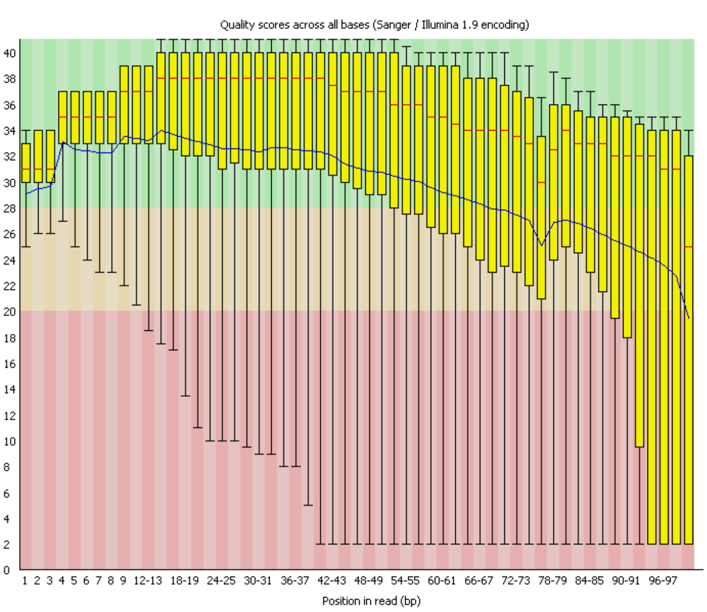


Figure 1. Average base call quality

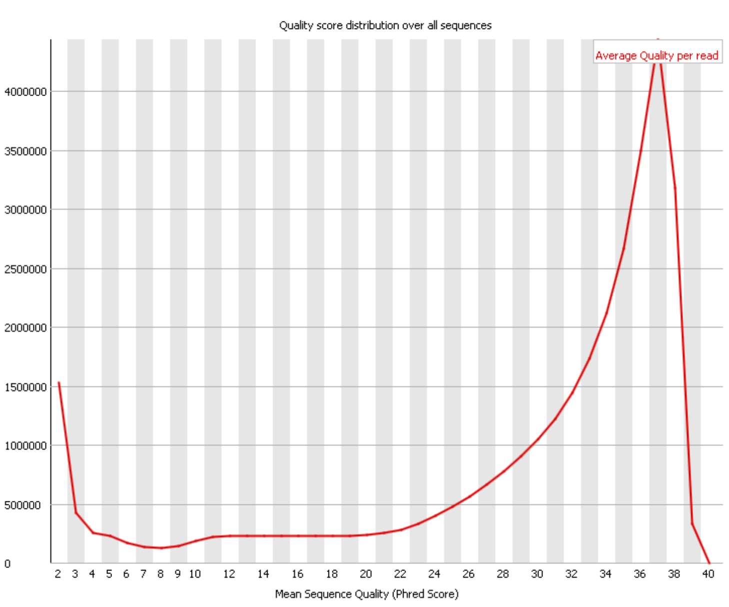


Figure 2. Overall read quality distribution

Sequence Alignment

Sequence alignment was accomplished using Bowtie2[6], a fast and sensitive command line alignment tool. The program was installed to the command line using Conda[ln63]. The software functions by aligning paired or unpaired reads to an indexed reference genome. The BN7.2 reference genome was indexed by first decompressing from its TAR format[ln65].

```
'tar -x -f file.tar'
```

Followed by using bowtie2's built in indexing software to create an index set for bowtie2 to refer to[ln70].

```
'bowtie2-build reference_genome.fna bowtie'
```

This created a set of .bt2 indexing files saved to their own folder that bowtie could use to align the input sequences to the reference. For the alignment itself, there were a series of options regarding the type of alignment desired[tr2,ln100-104]. Default settings were used for speed and sensitivity, the '-p 4' command was added to increase the CPU threads used during the analysis and speed up the process. The reads were formatted to bowtie2 as unpaired as this gave the fewest problems. The finished alignment was output as a sequence alignment map (.sam)

```
'bowtie2 -p 4 \
        -x reference in index \
        -U read1.fastq  \
        -U read2.fastq \
        -S output_file.sam'
```

As quick as the alignment software claims to be, individual paired reads took about 10-15 minutes per alignment. Aligning multiple reads at once took up to several hours. However, this alignment step was one of the few that was not hampered by large file sizes. Even 8 combined forward and reverse reads were aligned by bowtie in time, and produced .sam files upwards of 12Gb. This is worth mentioning as file sizes became a recurring issue.

Sam and Bam Formats

The output data from the bowtie2 alignment was in .sam file format. The overwhelming majority of bioinformatic tools work with .bam files. Samtools[7] was installed using conda[ln64] to an environment created specifically for this project[ln116]. As a result of this, everytime samtools was used, the environment had to be activated. Samtools had a relatively simple method for converting sam files into bam files, which involved viewing the sam file in bam format, and placing that output in a .bam file[ln108].

```
‘conda activate FinProjEnv
	samtools view -S -b file.sam > newfile.bam
conda deactivate’
```

The resulting .bam files were significantly smaller as is to be expected. The .bam file is a binary format of the .sam file, making it smaller and easier to store.

Sorting and Indexing

While not necessary for the final variant analysis, I wanted to view the aligned sequences in the Integrative Genome Viewer[8] to ensure successful alignment and make note of any visual oddities in the data. IGV was installed from the Broad Institute as a non-command line program. Before aligned .bam files could be opened in the viewer they needed to be sorted, then indexed. IGV has a built in toolkit; igvtools, that could sort, index, count, and transfer to TDF. Bam files were first sorted, requiring a source and destination file, as well as preset max reads count. I set the ‘max reads’ value to above however many reads were in the file I was sorting[tr3]. Sorted files were then indexed using the same igvtools program to create a complementary .sorted.bam.bai file that the viewer could reference when opening the .sorted.bam file.

Visualizing Alignments with IGV

Once the .bam files were sorted and indexed, they could be opened in IGV. Each forward and reverse aligned read represented relatively little coverage of the Rnor_7.0 reference genome (another brown rat reference). As such it was difficult to get a clear view of consistently mismatched bases and exon coverage. The figure below is of 6 aligned reads (Figure 3). The first 4 each being of a different set of forward and reverse reads, and the 5th being of two forward and reverse reads, hence its increased depth and coverage. The desired visualization in this step was to be of several combined forward and reverse reads aligned, but due to file size restrictions, the maximum IGV was able to process was two[tr3]. The completed alignments had partial coverage on the genes of interest. However it was too sparse to allow an analysis, especially on the exons. This step was ultimately used to just ensure alignments occurred correctly.

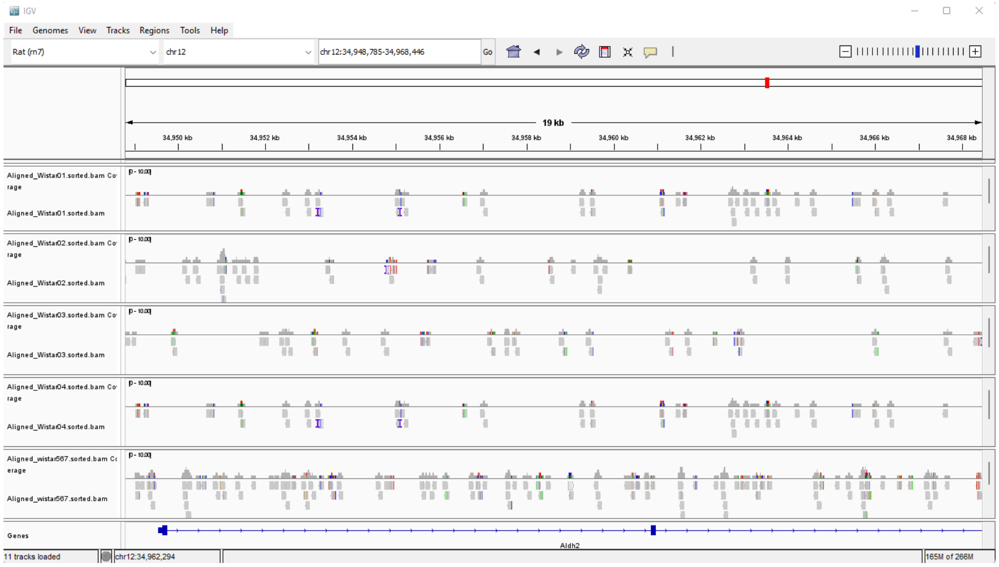

IGV
File
Genomes
View
Tracks
Regions
Tools
Help
Rat (rn7)
chr12
chr12:34,948,785-34,968,446
Go
19 kb
34,950 kb
34,952 kb
34,954 kb
34,956 kb
34,958 kb
34,960 kb
34,962 kb
34,964 kb
34,966 kb
34,968 kb
Aligned_Wistar01.sorted.bam Coverage
[0 - 10.00]
Aligned_Wistar01.sorted.bam
Aligned_Wistar02.sorted.bam Coverage
[0 - 10.00]
Aligned_Wistar02.sorted.bam
Aligned_Wistar03.sorted.bam Coverage
[0 - 10.00]
Aligned_Wistar03.sorted.bam
Aligned_Wistar04.sorted.bam Coverage
[0 - 10.00]
Aligned_Wistar04.sorted.bam
Aligned_wistar567.sorted.bam Coverage
[0 - 10.00]
Aligned_wistar567.sorted.bam
Genes
Aldh2
11 tracks loaded
chr12:34,962,294
165M of 266M

Figure 3. IGV visualization of 6 aligned Wistar sequences.

Normalizing .sorted.bam Files

For variant calling, files needed to be converted into .vcf or .vcf.gz files, which then needed to be normalized. Samtools was used to convert the files into .vcf.gz formats with the ‘mpileup’ command. This command uses the previously indexed BN7.2 reference genome to assemble the .vcf data from the .bam files, then bgzip to compress it into a .vcf.gz file[tr4, ln217].

```
‘samtools mpileup -f  reference_sequence.fasta  input_file.bam | bgzip > destination_file.vcf.gz’
```

The resulting .vcf.gz file was then normalized using bcftools. Bcftools[9] is an add-on for samtools designed for working with .bcf and .vcf file formats. The ‘norm’ command normalizes the file, and can output un/compressed .vcf or .bcf file types. I used the command to output a compressed .vcf.gz file[ln221].

```
‘bcftools norm -O z -o output_file.vcf.gz input_file.vcf(.gz)’
```

Variant Calling

Bcftools was again used, in this step, for variant calling. The ‘call’ command searches the normalized vcf.gz files for single nucleotide polymorphisms (SNP’s). This ultimately is the variation that is used for the analysis. The ‘-mv’ command was used to search for multiallelic variants only[ln220].

```
‘bcftools call -O z -mv -o output_file.vcf.gz  input_file.vcf.gz’
```

The output file was a .vcf.gz file with SNP variant calls within it that could be used for variant visualization.

Variant Analysis and Visualization

The SNP called .vcf.gz files needed to be viewed in a genome browser to identify variants in the genes of interest. Once again IGV was used. To view the files in IGV they were again first sorted and indexed with igvtools. Being .vcf files they were much smaller than the bam files initially viewed in IGV so they passed through the igvtools step quickly and without incident. The file was loaded into the viewer to visualize variants (Figure 4).

This is where the file size issue created an immovable roadblock. For proper variant calling, a full coverage genome must be used. Due to the size limitations of igvtools, no more than two reads were assembled. In fact the only file that could be properly normalized and called was a truncated .vcf file of the two sets of forward and reverse reads. This truncation likely occurred during a halted mpileup step, where the file was still present and occupied, but did not contain any calls past the first chromosome. As seen in Figure. 4, there are no calls past partway through the first chromosome, where none of the genes of interest are present. This leads this project to an unfortunate conclusion, in which the actual analysis cannot take place.

What I hoped to see in this analysis was significant variation in genes previously linked to alcohol abuse in the wistar rat genome. Some of the genes I planned to look at include; ALDH2, ADH1B-C, FTO, SLC39A8, VRK2, DCLK2, ISL1,[10,11] as well as others included in the papers cited below. As I previously stated, none of these genes are present in the first half of the first chromosome, and I have exhausted all options to obtain a full coverage variant call. There inevitably is variation in the genome that causes the Wistar's increased propensity as it is not an environmentally acquired trait and as such must source from gene variance. Hopefully, I can revisit this project in the future with hardware capable of working with such large file sizes to complete this analysis.

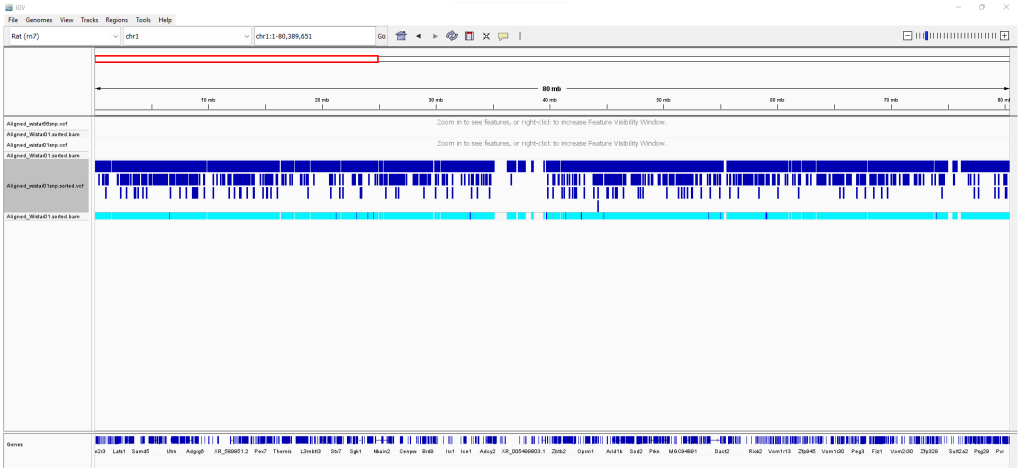

IGV
File Genomes View Tracks Regions Tools Help
Rat (rn7)
chr1
chr1:1-80,389,651
Go
80 mb
10 mb
20 mb
30 mb
40 mb
50 mb
60 mb
70 mb
80 mb
Aligned_wistar56snp.vcf
Aligned_Wistar01.sorted.bam
Aligned_wistar01snp.vcf
Aligned_Wistar01.sorted.bam
Aligned_wistar01snp.sorted.vcf
Aligned_Wistar01.sorted.bam
Zoom in to see features, or right-click to increase Feature Visibility Window.
Zoom in to see features, or right-click to increase Feature Visibility Window.
Genes
n2r3 Lats1 Samd5 Utrn Adgrg6 XR_589851.2 Pex7 Themis L3mbtl3 Stx7 Sgk1 Nkain2 Cenpw Brd9 Irx1 Ice1 Adcy2 XR_005499803.1 Zbtb2 Oprm1 Arid1b Sod2 Prkn MGC94891 Dact2 Riok2 Vom1r13 Zfp945 Vom1r30 Peg3 Fiz1 Vom2r30 Zfp329 Sult2a2 Psg29 Pvr

Figure 4. IGV variant visualization of chromosome 1 truncated .vcf

Troubleshooting

Appendix of significant roadblocks encountered in the project, plenty more issues arose and were solved but were not deemed important enough to include here.

1. This was early in the project pipeline, I had trouble finding the correct command to unzip gz files. It was solved with a relatively simple set of google searches until the gzip -d command was found.
2. I encountered significant formatting errors when attempting to align sequences. I would receive errors claiming to exit with value 1. Google searches did not present anything usable. Eventually I got in contact with Professor Miller, who separated the command segments across several lines and changed the input reads from paired to unpaired. This resolved the issue.
3. I had issues with sorting large bam files. I found the igvtools sort command could not process more than 22 million reads. My first attempt at solving this problem was using different sorting softwares. Igvtools had a command line version of the program. Attempting the process in WSL resulted in even fewer reads read before the program quit. I then tried using samtools, which quit part way through as well. I realized the issue was in the RAM memory pre-allocated to java, which was the language igvtools was

running in. I did not have a java console and was unfamiliar with the language so attempts to increase its memory allocation were unsuccessful. I manually increased the RAM allocation in my settings but it had no effect. I attempted to use temporary directories, to store the files when the program couldn't, but for whatever reason igvtools failed to utilize it. This left me with the unfortunate realization that I could not sort files larger than 22 million reads. Each aligned set of sequences had 8,000,000 reads, meaning I could not sort a combination greater than 2 forward and reverse aligned reads which held 16,000,000 reads total. This is largely where hope for a significant final analysis halted, with only partial coverage it was unlikely meaningful conclusions could be drawn.

4. There were significant issues encountered when attempting to normalize .vcf or .vcf.gz files. The only file that was recognized by either the normalizing or calling commands was the chromosome 1 truncated .vcf. All other .vcf files, whether compressed to .gz or not, could not be recognized by the commands. The terminal would give the same output; 'unknown file type'. Again, with the assistance of Professor Miller, attempts were made to solve this issue by using various combinations of file types, rearranging the compression order, and indexing .vcf files. This issue was unresolved, and as such the only viewable variant call is the Wistar .vcf truncated in the first chromosome.

Citations

1. Vengeliene, Valentina. "A Comparative Study on Alcohol-Preferring Rat Lines: Effects of Deprivation and Stress Phases on Voluntary Alcohol Intake." *Alcoholism, Clinical and Experimental Research*, U.S. National Library of Medicine, https://pubmed.ncbi.nlm.nih.gov/12878910/.
2. "MRATBN7.2 - Genome - Assembly - NCBI." *National Center for Biotechnology Information*, U.S. National Library of Medicine, https://www.ncbi.nlm.nih.gov/assembly/GCF_015227675.2.
3. Embl-Ebi. *Ena Browser*, https://www.ebi.ac.uk/ena/browser/view/PRJEB6678?show=reads.
4. 1.Liu, Tiancheng, et al. "Comparative Genome of GK and Wistar Rats Reveals Genetic Basis of Type 2 Diabetes." *PloS One*, Public Library of Science, 3 Nov. 2015, https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4631338/.
5. *Babraham Bioinformatics - FastQC a Quality Control Tool for High Throughput Sequence Data*, https://www.bioinformatics.babraham.ac.uk/projects/fastqc/.
6. "Bowtie 2." *Bowtie 2: Fast and Sensitive Read Alignment*, http://bowtie-bio.sourceforge.net/bowtie2/index.shtml.
7. *Samtools*, http://www.htslib.org/.
8. *Home | Integrative Genomics Viewer*, https://software.broadinstitute.org/software/igv/.
9. "Bcftools." *Bcftools by Samtools*, https://samtools.github.io/bcftools/.

10. .Edenberg, Howard J, and Tatiana Foroud. “Genetics and Alcoholism.” *Nature Reviews. Gastroenterology & Hepatology*, U.S. National Library of Medicine, Aug. 2013, https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4056340/.
11. “Study Reveals Genes Associated with Heavy Drinking and Alcoholism.” *ScienceDaily*, ScienceDaily, 2 Apr. 2019, https://www.sciencedaily.com/releases/2019/04/190402124314.htm.

Appendix

- Full scale images of the figures references above, these can also be found as downloadable files in the ‘Appendix Images’ folder on this projects github

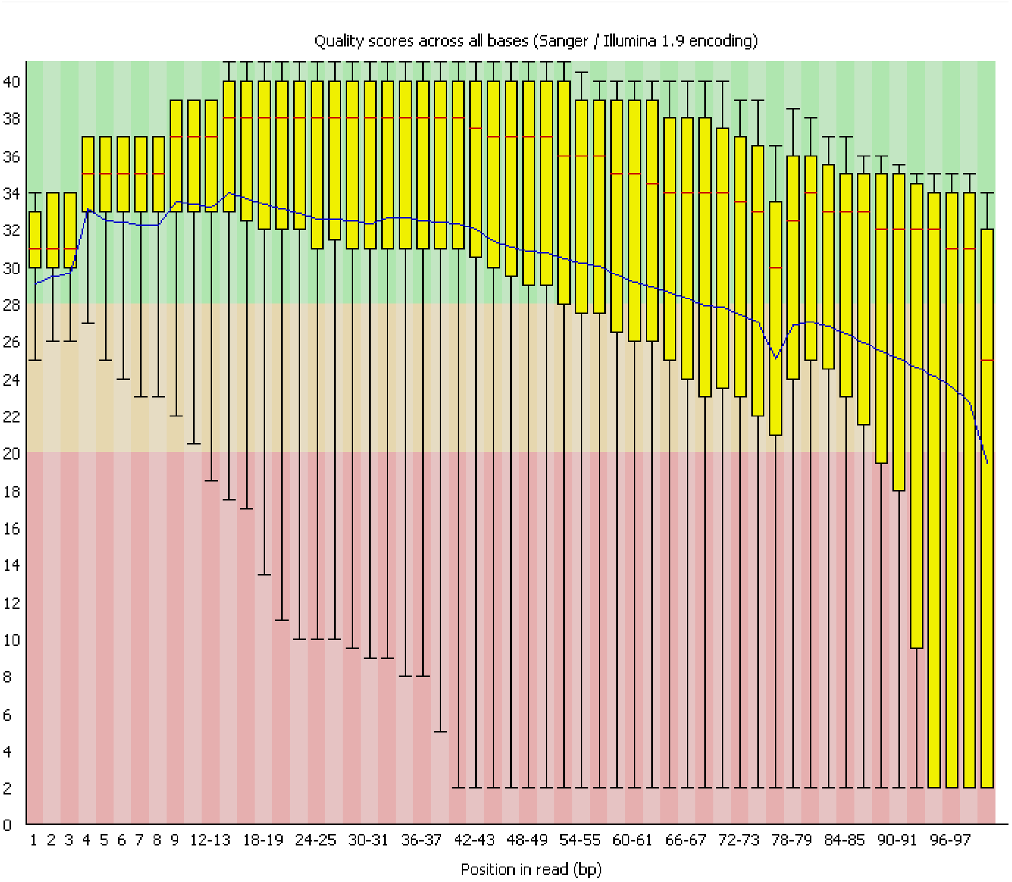


Figure 1. Average base call quality

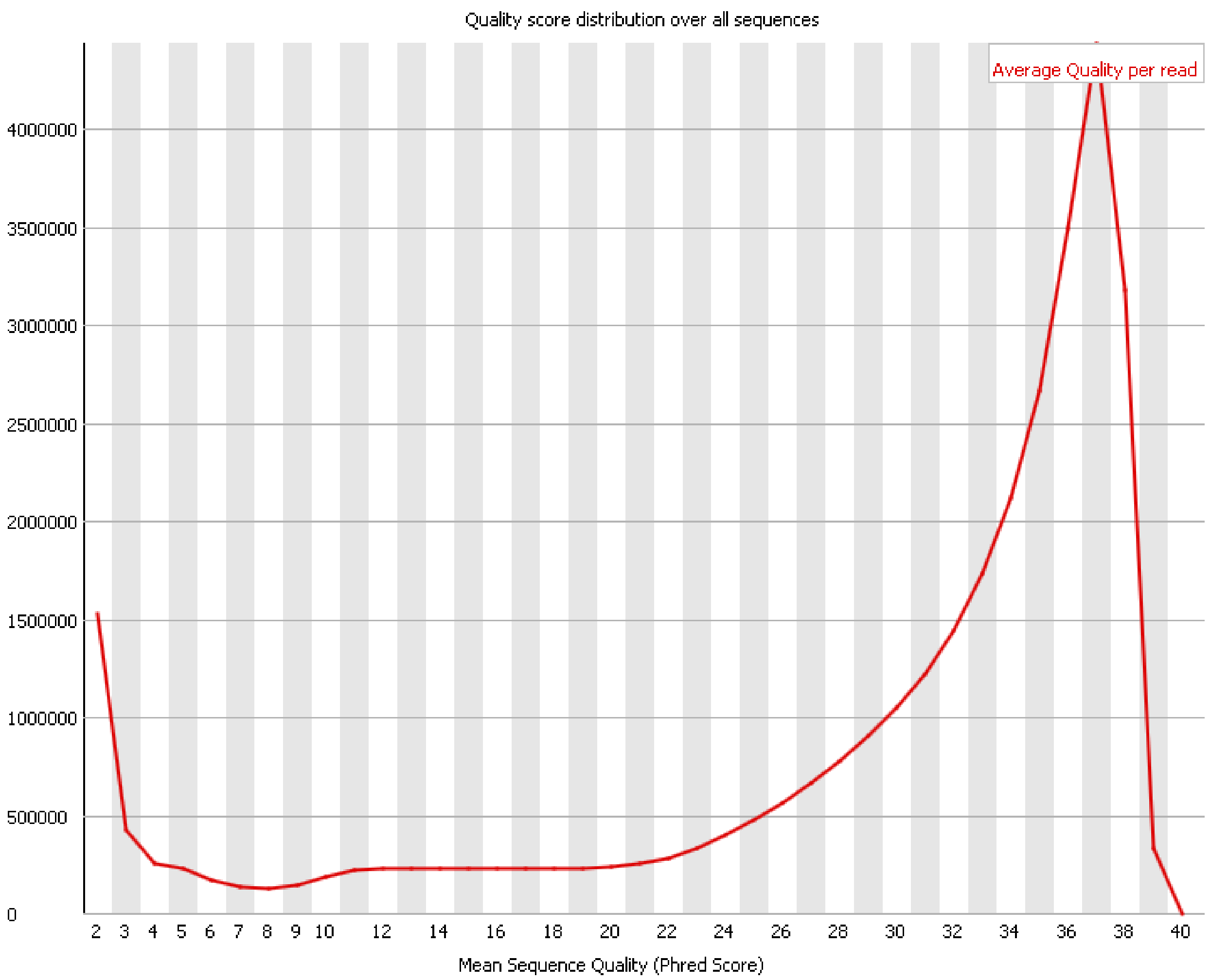


Figure 2. Overall read quality distribution

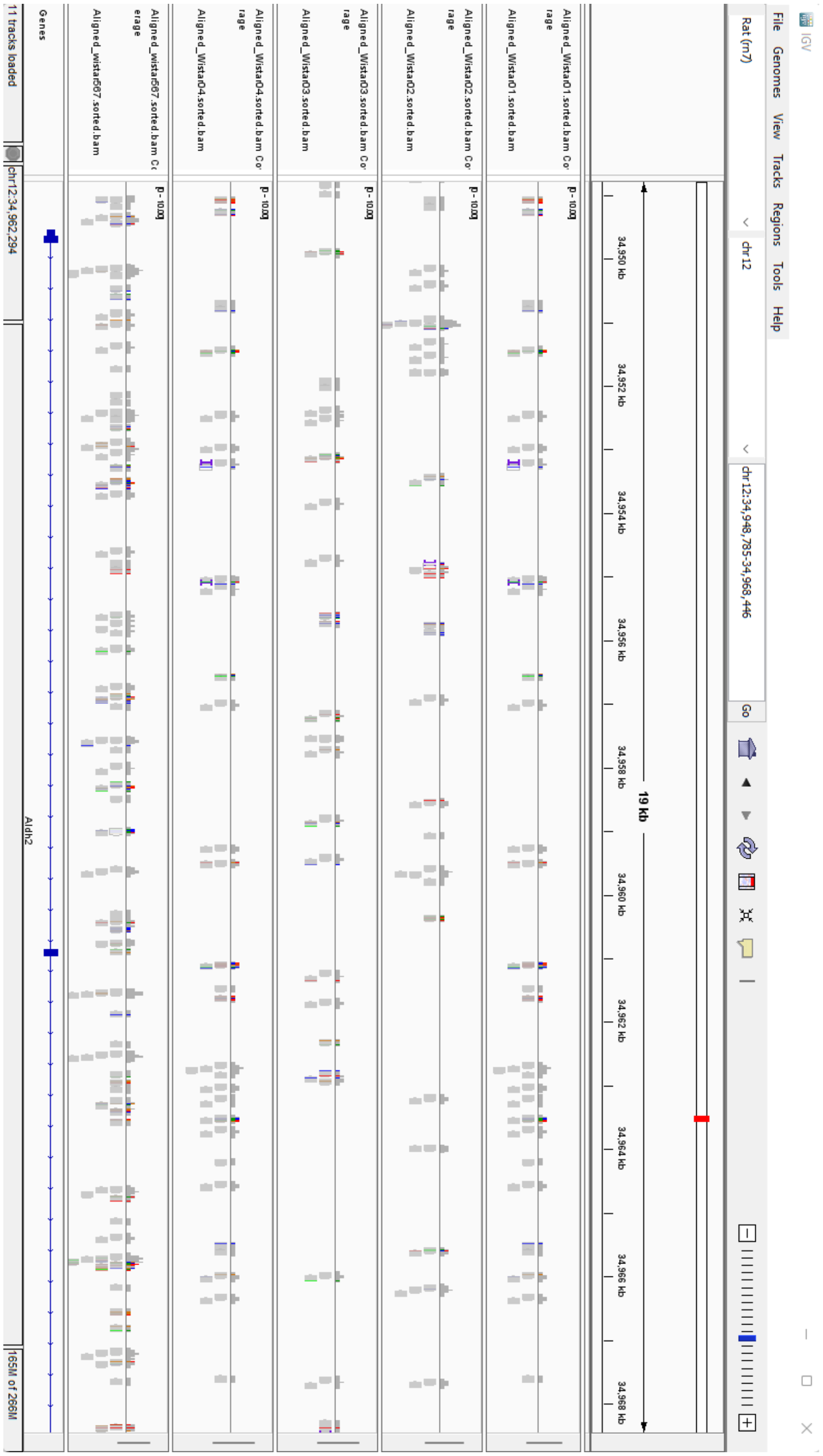
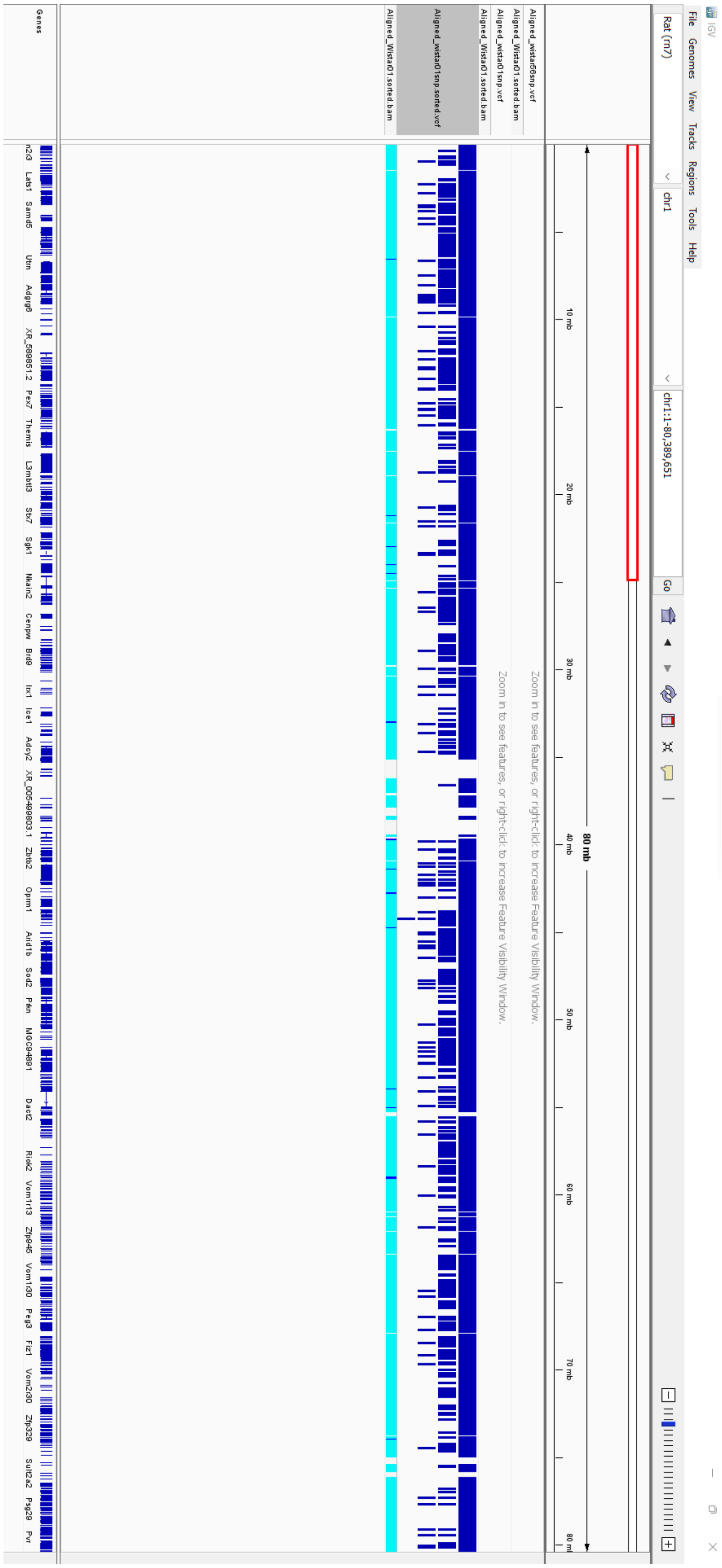

IGV
File Genomes View Tracks Regions Tools Help
Rat (rn7)
chr12
chr12:34,948,785-34,968,446
Go
19 kb
34,950 kb
34,952 kb
34,954 kb
34,956 kb
34,958 kb
34,960 kb
34,962 kb
34,964 kb
34,966 kb
34,968 kb
Aligned_Wistar01.sorted.bam Coverage
[0 - 10.00]
Aligned_Wistar01.sorted.bam
Aligned_Wistar02.sorted.bam Coverage
[0 - 10.00]
Aligned_Wistar02.sorted.bam
Aligned_Wistar03.sorted.bam Coverage
[0 - 10.00]
Aligned_Wistar03.sorted.bam
Aligned_Wistar04.sorted.bam Coverage
[0 - 10.00]
Aligned_Wistar04.sorted.bam
Aligned_wistar567.sorted.bam Coverage
[0 - 10.00]
Aligned_wistar567.sorted.bam
Genes
Aldh2
11 tracks loaded
chr12:34,962,294
165M of 266M

Figure 3. IGV visualization of 6 aligned Wistar sequences.


IGV
File Genomes View Tracks Regions Tools Help
Rat (rn7)
chr1
chr1:1-80,389,651
Go
Aligned_wistar56snp.vcf
Aligned_Wistar01.sorted.bam
Aligned_wistar01snp.vcf
Aligned_Wistar01.sorted.bam
Aligned_wistar01snp.sorted.vcf
Aligned_Wistar01.sorted.bam
Genes
10 mb
20 mb
30 mb
40 mb
50 mb
60 mb
70 mb
80 mb
80 mb
n2r3 Lats1 Samd5 Utrn Adgrg6 XR_5909512 Pex7 Themis L3mbtl3 Stx7 Sgk1 Nkain2 Cenpw Brd0 Inx1 Ice1 Adoy2 XR_005490803.1 Zbtb2 Oprm1 Arid1b Sod2 Pkm MGC94891 Dact2 Rlok2 Vom1r13 Zfp945 Vom1r30 Peg3 Fiz1 Vom2r30 Zfp329 Sult2a2 Psg29 Pvr

Zoom in to see features, or right-click to increase Feature Visibility Window.

Zoom in to see features, or right-click to increase Feature Visibility Window.

Figure 4. IGV variant visualization of chromosome 1 truncated .vcf