**Outline**

Date of conception of idea - January 15, 2021

1. **Data Analysis**
   1. **Summary of compression softwares**

Check Summary of Compression Softwares.xlsx in CyBox

* 1. **Other Important papers**

Review - Comparison of high-throughput sequencing data compression tools

CRAM - Efficient storage of high throughput dna sequencing data using reference-based compression, The scramble conversion tool

BAM - The sequence alignment/map format and samtools Check Data Compression Conference

1. **To do**
   1. Finalize the compression algorithm for single ended reads - currently it achieves good compression ratio within an acceptable time
      1. ~~Decide on file names~~
      2. ~~Update the code to dump quality values for mismatches and insertions~~
      3. ~~Think about the optimizations possible for each of the flags~~
   2. ~~Write the program for decompression for single ended reads~~
   3. ~~Include an option for the user to save all quality scores. Then use QVZ to compress them. Too much space is needed and beats the purpose of compressing. Two modes are allowed - one to ignore the quality values and the other to store only softclips, insertions and mismatches.~~
   4. Include options to generate sequence and specific tags while decompressing
   5. ~~Write the program to split up merged files~~
   6. ~~Modify code to accommodate DNA-Seq reads~~
   7. ~~Encode the case of compressing multiple files at the same time into a single compressed file~~
   8. ~~Think about the option of merging multiple compressed files~~
   9. ~~Multi-threading the compressed file merge and the compression of multiple SAM files~~
   10. Design 2 flowcharts - one a short version (that will go in the main manuscript) and the other a longer version (for supplementary) outlining every single process in detail - This will take quite a bit of time. Design in Microsoft power point. Decide whether to put single and paired ended reads in the same diagram.
   11. ~~Draft out the compression approach for paired ended reads~~
   12. Prepare a makefile of the program and remove all the compilation steps from the main python program
   13. Prepare a table to list all the software that compress alignment data. Keep updating this table when new software is found
   14. For each os.system command track whether the command has successfully executed
   15. Write a test script to ensure everything is installed correctly
   16. Change the zpaq to a conda installation
   17. ~~For decompressed single ended, check those alignments which are not spliced and make sure they do not have XS tag with them~~
   18. ~~Clean the C code and add comments wherever necessary. Redo the header file. Put the function declarations in the header file.~~
   19. ~~Try out different techniques of optimizing the C code and compilation~~
   20. ~~Rewrite the entire code in C++.Tried that slowed things down.~~
   21. ~~Add references to Mendeley~~
   22. ~~Think about implementing random access. Build index file that can hasten the speed of operation. Have different options of generating indices - like a deep index or a shallow index~~
   23. ~~Build different indices for RNA-Seq and DNA-Seq reads~~
   24. Write a program to compare the original samfile with the decompressed samfile. Write in python. No need for time optimization - This will take time. Hence, I re-wrote the compress program to run diagnostics. It will compress and decompress the file to obtain cigar, MD string and nucleotide sequence and compare it with the original alignment.
   25. ~~Write a program in C to quickly copy paste the quality values to file. Need it when users request lossless or lossy quality values compression~~
   26. ~~Think about producing coverage files like in bedtools. Then performance of bedtools with bam or cram could be compared with abridge [Future work]~~
   27. ~~Download and test the other compression software~~
   28. ~~Move everything to 90daydata on ceres~~
   29. CRAM stores the genome sequence too
   30. ~~Generating sequencing data for comparison analysis~~
   31. ~~Rethink the decompression phase where the sequences are being extracted. It will create issues when multiple files are being decompressed at once. Recommend users to decompress one file at a time with multiple cores. [Resolved - now occupying much less memory()]~~
   32. Write a function in python to estimate the total size of the genome and determine whether the whole genome can be loaded to the memory at once
   33. Include a memory option to the program. Will be needed during decompression. Compression phase does not need the genome.
   34. Consider designing a software to transfer alignments from genomes to transcriptome
   35. Submit to conda
   36. Quality check and comparison
       1. Download the genome sequence. Decide whether to stick to Arabidopsis thaliana or should we explore other organisms too. Download and align to the unmasked genome.
       2. ~~Do not simulate reads. Use samples from RNA-Seq experiments on NCBI-SRA. Choose experiments with replicates. Choose both RNA-Seq and DNA-Seq. How many samples to process??? Check other publications. Select from a range of depths~~
       3. ~~Rerun alignments with --outBAMsortingThreadN set to 1~~
       4. Remove all alignments that have indels, mismatches and soft-clips - required for comparison with softwares that do not support indels, mismatches and soft-clips (like boiler)
   37. Write a program in python to compare the decompressed SAM file to the original SAM alignments.
2. **Main Figures**

Figures are not in the order they appear in manuscript

* 1. **WorkflowMF**

Caption - This is the caption for this figure

About - Figure will have two panels. One describing compression the other panel describing decompression. Figure will be made in Microsoft power point

* 1. **DifferentModesOfAbridgeCompressionMF**

Time and compression ratio achieved by different modes of compression by both 7z and brotli.

--skip mismatches --skip soft clips etc.

* 1. **TimeComparisonRNAMF**

Illustrate linear increase of time with increase in number of reads x-axis - Increasing size of input

y-axis - Compression time

Will have at least 4 panels - single ended reads, paired-ended reads, compression, decompression Might have to split it if number of panels exceed 4

Include retrieval from bam file too. In the manuscript mention that retrieval from bam file works faster since it is not a very good form of compression.

* 1. **CompressionRatioComparisonRNAMF**

Similar to TimeComparisonMF but for space - only compression no decompression

Space reduction from pass1 and pass2 for normal run and also for other runs with soft-clips-skipped, etc.

* 1. **CoverageDataGenerationTimeMF**

Time to generate per nucleotide coverage information from abridge compressed files compared to bedtools using bam file. Compare the correctness of both cases. Also check memory consumption.

* 1. **ReadNameNoSaveCheckMF**

For single ended reads, we will not be storing the read names. Will this impact downstream analysis like assembly and gene count generation?

* 1. **CoverageGenerationOptionsMF**

Options to generate coverage. Put two panels – one for overlapping and the other for non-overlapping

* 1. **IntegratedCIGARConstructionMF**

A figure to depict how complicated icigars are constructed

* 1. **CoverageGenerationMF**
  2. **~~MergeAbridgeCompressedRNAMF~~**

~~Compressing multiple files 1 panel for space reduction and another panel for time Select 3 best compressors (other than abridge) and compare them here~~

~~Keep each panel for different kinds of compression like skipping mismatches, skipping soft clips etc..~~

1. **Main Tables**
   1. **SummaryOfCompressionSoftwareMT**

Check Summary of Compression Softwares.xlsx in CyBox

1. **Supplementary Figures**
   1. **DetailedFlowchartSF**

Break into multiple panels

* 1. **TimeComparisonDNASF**
  2. **Conduct over multiple coverages**
  3. **TimeComparisonChIPSF**

CompressionRatioComparisonDNASF

Panel A - Conduct over multiple coverages Panel B - Conduct over multiple files

* 1. **CompressionRatioComparisonChIPSF**
  2. **transcriptomeReferenceRNASF**

Map reads to the transcriptome and store it in

* 1. **WheatSF**

Apply on wheat since it has largest genome and also draw on the homeologue thing

* 1. **RandomRetrievalTimeSF**

Time to retrieve different regions one panel for RNA-Seq and another for DNA-Seq

* 1. **IndexGenerationTimeSF**

Time taken to generate abridge index vs boiler or bai csi cram etc.

1. **Supplementary Tables**
   1. **ListOfNCBI-SRASamplesForExperimentST**

Select a mix of RNA-Seq, DNA-Seq, and other types of sequencing

* 1. **MappingTimeST**

STAR mapping times on single core along with compression and decompression

* 1. **ComparisonAmongCompressorsST**

Compare compression (and decompression) algorithms based on time, memory consumed, and compression ratio achieved

* 1. **DifferenceDueToNoNamePreserveSingleEndedST**

A list of comparisons between gene counts for RNA-Seq data for original bam file and decompressed bam file to show there is hardly any impact due to not saving of read names. Run Salmon and RSEM. Remove this table if counts are exactly same. Also do a DESeq run????

* 1. **MemoryConsumptionCompressionDecompressionST**
  2. **CompressionOptionsST**

A list of options of compression offered by ABRIDGE

1. **Supplementary Document**

How the other compression softwares were executed?

How was the NCBI-SRA data downloaded and mapped? Provide full commands

Best is to create a separate GitHub repo/Code Ocean to reproduce the entire workflow Things to mention:

How was STAR used for mapping?

Code to select only those mappings that have no soft-clips, mismatches, indels, etc.

How single ended data was generated. How time and memory was calculated