Analysis and Development of Tools for Genome and Transcriptome Manipulation

Dhananjay Raman, d.raman@iitb.ac.in

Introduction

Tools for manipulating transcriptome data in the form of GTF files usually specialize in various types of queries, such as interval arithmetic, aggregate queries, and data filtering, but their performance can vary depending on the specific task. This project aims to systematically benchmark these tools across a set of typical workloads to identify their strengths and weaknesses. Targeted optimizations are implemented to enhance their efficiency at bottlenecks. The improved tools will streamline data processing in high-throughput studies, enabling faster discovery and better handling of large datasets.

Performance Metrics

- **Data Loading**: Time to load GTF files into the tool's data structure
- **Aggregate Queries**: Counting exons per gene, calculating total exon length, and identifying chromosomes with the most transcripts
- **Interval Queries**: Merging exon intervals, finding overlaps with a specific interval, and subtracting specified repetitive intervals

The SQL-PyBedTools Method

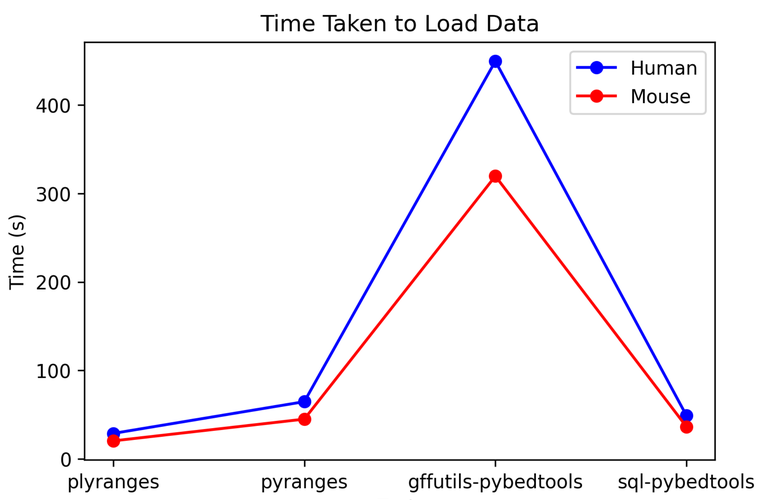
- Combine SQL and PyBedTools to optimize aggregate and interval queries
- Convert pyranges data frame into an in-memory SQL database for aggregate operations
- SQL is more efficient for large-scale aggregation tasks due to optimized database queries
- For interval operations, PyBedTools is more efficient due to its specific data structure

Tools and Benchmarks

- **PyRanges**: A Python library designed for efficient genomic range operations, focusing on interval arithmetic and aggregations
- **Plyranges**: An R package that facilitates genomic data manipulation using a grammar-based approach for transformation and analysis
- **gffutils-pybedtools**: A Python tool that combines gffutils for GTF/GFF file handling with PyBedTools for interval operations, enabling advanced genomic analysis
- **sql-pybedtools**: A hybrid method that stores genomic data in an in-memory SQL database, using SQL for aggregate queries and PyBedTools for interval operations
- The performance of each tool was evaluated on both the human (**GRCh38.p13**) and mouse (**GRCm39**) Ensembl GTF datasets

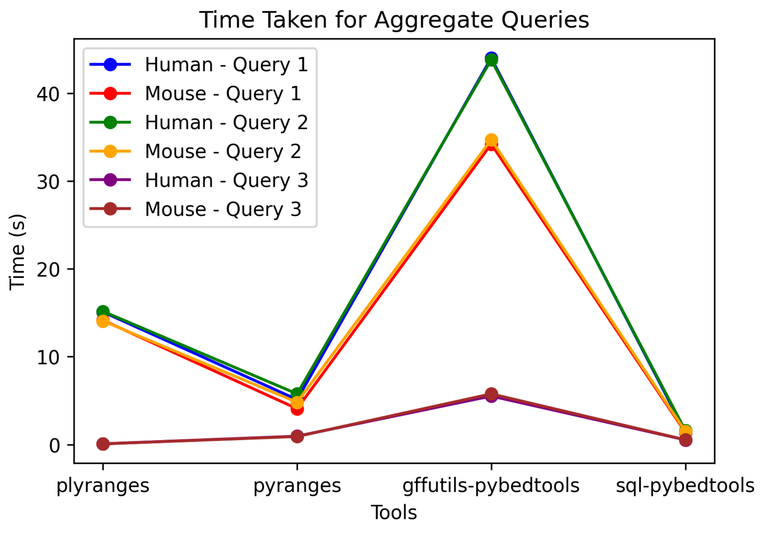
Data Loading Results

- **Human Data**: Plyranges was the fastest (28.8 seconds), followed by sql-pybedtools (49.1 seconds), pyranges (64.7 seconds), and gffutils-pybedtools (449.5 seconds)
- **Mouse Data**: Plyranges remained the fastest (20.4 seconds), pyranges took 45 seconds, sql-pybedtools took 36.5 seconds, and gffutils-pybedtools took 319.6 seconds
- **Conclusion**: Plyranges is the most efficient for data loading, while gffutils-pybedtools shows significant delays, likely due to unnecessary SQL conversions and indexing



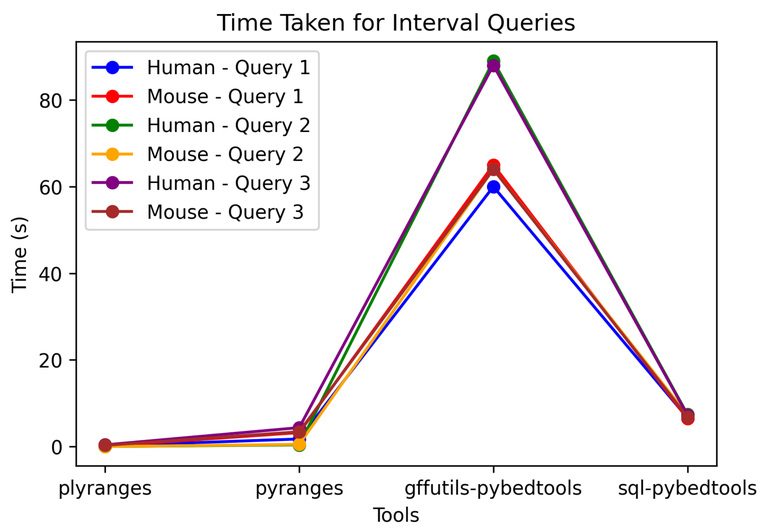
Aggregate Query Results

- **Human Data**: Sql-pybedtools was the fastest for all aggregate queries, followed by pyranges and plyranges, with gffutils-pybedtools lagging behind
- **Mouse Data**: Sql-pybedtools maintained the fastest performance, with plyranges and pyranges showing competitive results, while gffutils-pybedtools was slower
- **Query 3**: Plyranges was the fastest for query 3 on the human dataset, followed by pyranges, with gffutils-pybedtools and sql-pybedtools showing slower times
- **Conclusion**: Sql-pybedtools was the most efficient overall, with plyranges and pyranges excelling on specific queries



Interval Query Results

- **Human Data**: Plyranges was the fastest for all interval queries, followed by pyranges, with sql-pybedtools and gffutils-pybedtools showing significant delays
- **Mouse Data**: Plyranges again outperformed all tools for query 1, with pyranges performing well, while gffutils-pybedtools showed slow times
- **Conclusion**: Plyranges excelled in interval queries, while gffutils-pybedtools struggled, and sql-pybedtools was less efficient



Conclusions

Plyranges consistently outperformed other tools in both aggregate and interval queries, offering the fastest execution times, while sql-pybedtools excelled in aggregate queries and gffutils-pybedtools showed significant delays.
Future work could focus on optimizing gffutils-pybedtools for faster query performance, exploring multi-threading or parallel processing capabilities in all tools, and testing with larger genomic datasets for scalability.

References

- Stovner, E. B., Sætrom, P. (2019). PyRanges: efficient comparison of genomic intervals in Python. Bioinformatics, 36(3), 918-919. https://doi.org/10.1093/bioinformatics/btz615.
- Lawrence, M., Huber, W., Pagès, H., Aboyoun, P., Carlson, M., Gentleman, R., Morgan, M. T., Carey, V. J. (2013). Software for Computing and Annotating Genomic Ranges. PLOS Computational Biology, 9(8), 1-10. https://doi.org/10.1371/journal.pcbi.1003118.
- Lee, S., Cook, D., Lawrence, M. (2019). Plyranges: A grammar of genomic data transformation. Genome Biology, 20, 1-8. https://doi.org/10.1186/s13059-018-1597-8.
- Dale, R. (2011). gffutils: GFF and GTF file manipulation and interconversion. Retrieved from https://github.com/daler/gffutils.
- Dale, R. (2010). PyBedTools: Python wrapper -- and more -- for BEDTools (bioinformatics tools for "genome arithmetic") Retrieved from https://github.com/daler/pybedtools