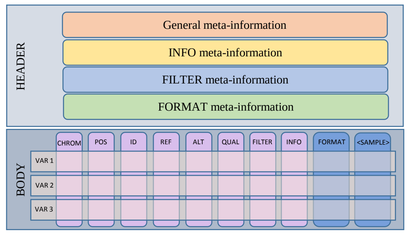
Efficient and iterative Genome Analytics using a graph-based model

Sanna Aizad and Ashiq Anjum
{s.aizad, a.anjum}@derby.ac.uk

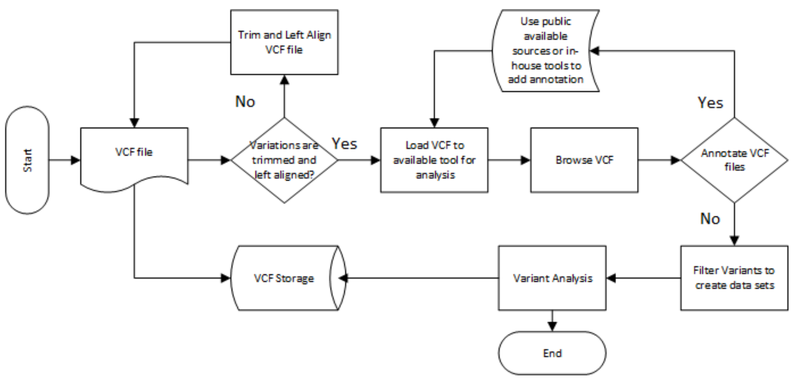
Abstract: The human genome is represented as a linear sequence of 3.2 billion base pairs as a "Reference Genome". The variations within individuals are stored in Variant Calling Format (VCF) text files. These files are the primary format choice for genome analysis and can reach a huge size. For example, the file sizes of 1700 participants from the 1000 Genomes Project is approximately 200TB. Consequently, reading and processing these text files takes a lot of time and resources. Here we look at a new way of representing the data with in these text files as a graph-based model. This new representation will allow for quick and efficient iterative analytics. This work aims to motivate representing the VCF and FASTA files as Graphs to run on a cloud to exploit the high-performance capabilities provided by cloud computing.

♦ Background



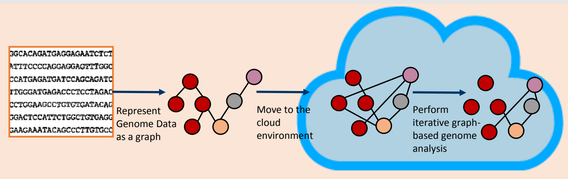
VCF file containing Variations of the Genome

FASTA file containing the Reference Genome of Humans

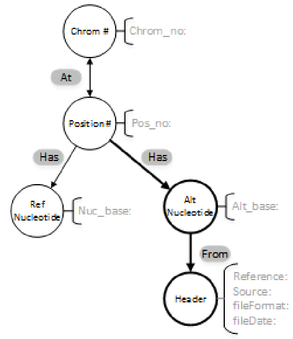


Processing a VCF file

♦ Basic Concept

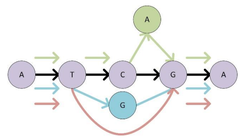


♦ Graph Model: VCF to Graph

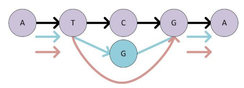


Graph Model of VCF file

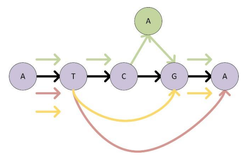
#CHROM	POS	ID	REF	ALT	QUAL	FILTER	INFO
20	3	.	C	G	.	PASS	DP=100
20	2	.	TC	T	.	PASS	DP=100
20	2	.	TC	TCA	.	PASS	DP=100



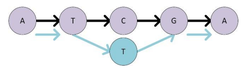
#CHROM	POS	ID	REF	ALT	QUAL	FILTER	INFO
20	2	.	TC	TG,T	.	PASS	DP=100



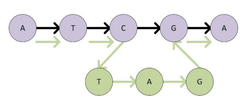
#CHROM	POS	ID	REF	ALT	QUAL	FILTER	INFO
20	2	.	TCG	TG,T,TCAG	.	PASS	DP=100



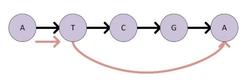
#CHROM	POS	ID	REF	ALT	QUAL	FILTER	INFO
20	3	.	C	T	.	PASS	DP=100

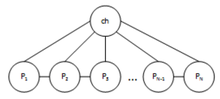


#CHROM	POS	ID	REF	ALT	QUAL	FILTER	INFO
20	3	.	C	CTAG	.	PASS	DP=100



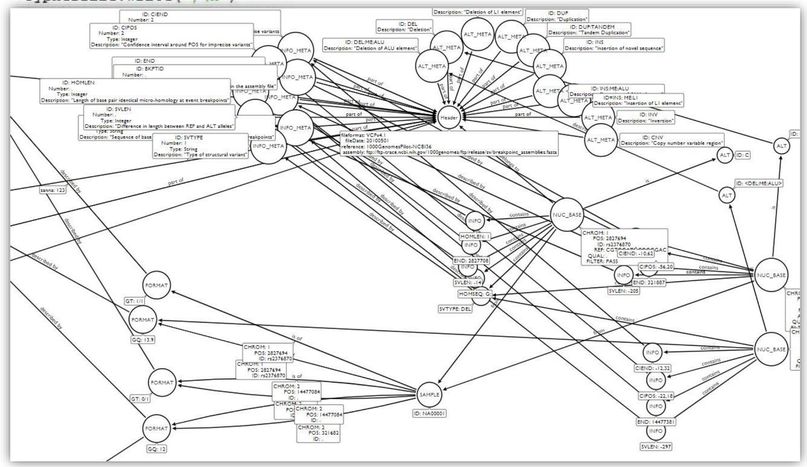
#CHROM	POS	ID	REF	ALT	QUAL	FILTER	INFO
20	2	.	TCG	T	.	PASS	DP=100





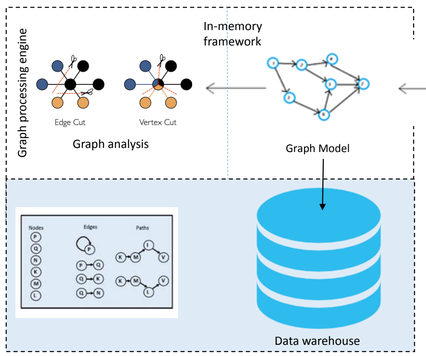
Graph Model of FASTA file


```
cypherfile.write('CREATE(Pos')
cypherfile.write('{}'.format(pos_no))
cypherfile.write(')-[:has {chrom:%a, ID: %a, Ref_base: %a}]->(Alt_' %(ch, ID, Ref))
cypherfile.write('{}'.format(alt_id))
cypherfile.write(')\n')
```



Graph in neo4j

♦ Future Work



In the proposed architecture, the Graph model created from the VCF and Reference Genome would be moved to a graph-processing engine such as GraphX.

The native graph environment will allow for quick graph analysis.

♦ Conclusion

We have proposed a different data structure for genomic datasets as well as an in-memory architecture which allows quick, iterative processing without compromising data from the file-based datasets. Biological information hidden in text becomes apparent is the form of semantics in the graph model.

UNIVERSITY of DERBY