Speeding up NGS software development

Erwan Drezen, Guillaume Rizk, Rayan Chikhi, Charles Deltel, Claire Lemaitre, Pierre Peterlongo, Dominique Lavenier

GATB

Speeding-up NGS software development

Sequencing, Finishing and Analysis in the Future Meeting, Santa Fé, USA, May 2014

Erwan Drézen[1], Guillaume Rizk[1], Rayan Chikhi[2], Charles Deltel[1], Claire Lemaitre[1], Pierre Peterlongo[1] and Dominique Lavenier[1]

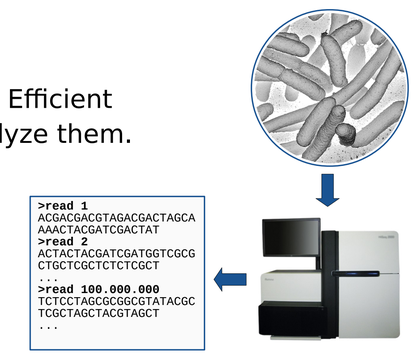
1 INRIA/IRISA/GenScale, Campus de Beaulieu, 35042 Rennes cedex

2 Department of Computer Science and Engineering, Pennsylvania State University, USA

1. What is GATB ?

Motivation

NGS technologies produce terabytes of data. Efficient and fast NGS algorithms are essential to analyze them.



Objective

The Genome Assembly Tool Box (GATB)

- is an open-source software
- provides an easy way to develop efficient and fast NGS tools
- is based on data structure with a very low memory footprint
- allows complex genomes to be processed on desktop computers

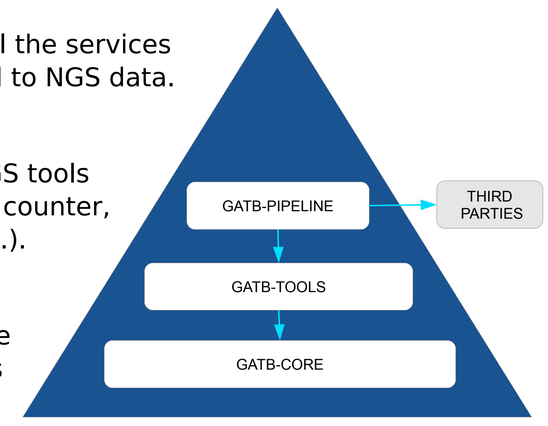
2. Software Solution

The GATB philosophy proposes a 3-layer construction to analyze NGS datasets

1. **GATB-CORE**: a C++ library holding all the services needed for developing software dedicated to NGS data.

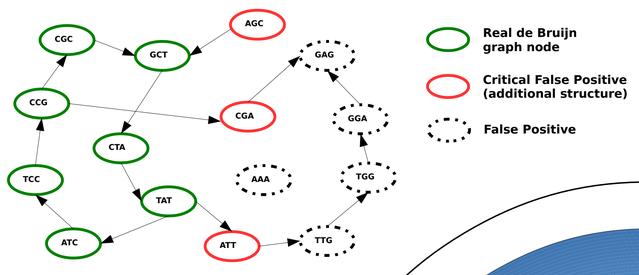
2. **GATB-TOOLS**: a set of elementary NGS tools mainly built upon the GATB library (k-mer counter, contiger, scaffolder, variant detection, etc.).

3. **GATB-PIPELINE**: a set of NGS pipeline that links together tools from the previous layer.



3. Compact de Bruijn graph data structure

The core data structure of GATB is a de Bruijn graph that encodes the main information from the sequencing reads.



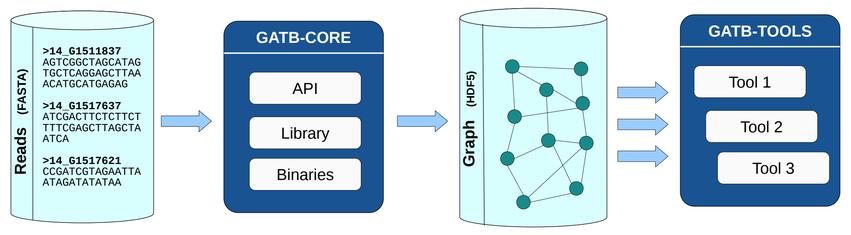
Strength of GATB

GATB makes this graph compact by using a Bloom filter (a space efficient probabilistic data structure) and by using a CFP additional structure that avoids false positive answers from the Bloom filter due to its probabilistic nature.

How to Analyze Complex Genomes on a Simple Desktop Computer ?

4. Workflow

Here is a typical workflow when working with GATB



GATB-CORE transforms the reads into a de Bruijn graph, saves it in a HDF5 file that can be opened by other tools developed with the GATB-CORE API.

5. GATB helps you as a NGS user

GATB's de Bruijn graph: a basis for families of tools

- **Data error correction**
- **Assembly**
- **Biological motif detection**

a whole human genome sequencing reads can be handled with 5 GBytes of memory

Several tools based on GATB are already available

Bloocoo K-mer spectrum based read error corrector for large datasets

Minia Short read assembler based on a de Bruijn graph. Results are of similar contiguity and accuracy to other de Bruijn assemblers (e.g. Velvet)

DiscoSNP Discover Single Nucleotide Polymorphism (SNP) from non-assembled reads

TakeABreak Detects inversion breakpoints without a reference genome by looking for fixed size topological patterns in the de Bruijn graph

6. GATB helps you as a NGS developer

The GATB C++ library gives you the opportunity to quickly develop new NGS tools that fit your needs.

Major facts about the GATB C++ library

- Object Oriented Design
- Simple and powerful graph API
- Simple and powerful multithreading model
- HDF5 usage for data storage
- Fully documented with numerous code samples
- Complete test suite

Publications

G. Rizk, D. Lavenier, R. Chikhi, **DSK: k-mer counting with very low memory usage**, Bioinformatics, 2013 Mar 1;29(5):652-3

R. Chikhi, G. Rizk. **Space-efficient and exact de Bruijn graph representation based on a Bloom filter**, Algorithms for Molecular Biology 2013, 8:22

G. Collet, G. Rizk, R. Chikhi, D. Lavenier, **Minia on Raspberry Pi, assembling a 100 Mbp genome on a Credit Card Sized Computer**, Poster at the JOBIM conference, 2013 Jul 1-4 (Toulouse) Best poster award.

K.I Salikhov, G. Sacomoto, G. Kucherov, **Using Cascading Bloom Filters to Improve the Memory Usage for de Brujin Graphs**, Algorithms in Bioinformatics, Lecture Notes in Computer Science, Volume 8126, 2013, pp 364-376

License & Web Site

For more details on GATB:

http://gatb.inria.fr

Partners