A Scalable Indexing Solution to Mine Huge Genomic Sequence Collections

Eric Rivals, Nicolas Philippe, Mikael Salson, Martine Léonard, Thérèse Commes, Thierry Lecroq

Number 89, April 2012

ERCIM NEWS

www.ercim.eu

Special theme:

Big Data

Also in this issue:

RESEARCH AND INNOVATION

This section features news about research activities and innovative developments from European research institutes

EVENTS, BOOKS, IN BRIEF

A Scalable Indexing Solution to Mine Huge Genomic Sequence Collections

by Eric Rivals, Nicolas Philippe, Mikael Salson, Martine Leonard, Thérèse Commes and Thierry Lecroq

With High Throughput Sequencing (HTS) technologies, biology is experiencing a sequence data deluge. A single sequencing experiment currently yields 100 million short sequences, or reads, the analysis of which demands efficient and scalable sequence analysis algorithms. Diverse kinds of applications repeatedly need to query the sequence collection for the occurrence positions of a subword. Time can be saved by building an index of all subwords present in the sequences before performing huge numbers of queries. However, both the scalability and the memory requirement of the chosen data structure must suit the data volume. Here, we introduce a novel indexing data structure, called Gk arrays, and related algorithms that improve on classical indexes and state of the art hash tables.

Biology and its applications in other life sciences, from medicine to agronomy or ecology, is increasingly becoming a computational, data-driven science, as testified by the launch of the Giga Science journal (http://www.gigasciencejournal.com). In particular, the advent and rapid spread of High Throughput Sequencing (HTS) technologies (also called Next Generation Sequencing) has revolutionized how research questions are addressed and solved. To assess the biodiversity of an area, for instance, instead of patiently determining species in the field, the DNA of virtually all species present in collected environmental samples (soil, water, ice, etc.) can be sequenced in a single run of a metagenomic experiment. The raw output consists of up to several hundred million short sequencing reads (eg from 30 to 120 nucleotides with an Illumina sequencer). These reads are binned into classes corresponding to species, which allow to reliable estimation of their number and relative abundance. This becomes a computational question.

In other, genome-wide, applications, HTS serve to sequence new genomes, to catalogue active genes in a tissue, and soon in a cell, to survey epigenetic modifications that alter our genome, to search for molecular markers of diseases in a patient sample. In each case, the read analysis takes place in the computer, and users face scalability issues. The major bottleneck is memory consumption. To illustrate the scale, currently sequences accumulate at a faster rate than the Moore law, and large sequencing centres have outputs of gigabases a day, so large that even network transfer becomes problematic.

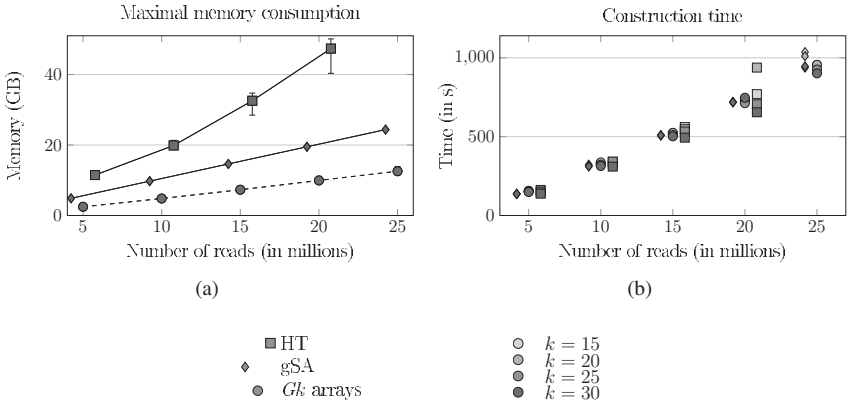

Figure 1: Comparison of Gk arrays with a generalised Suffix Array (gSA) and a Hash Table solutions on the construction time and memory usage.

Let us take an example. Consider the problem of assembling a genome from a huge collection of reads. Because sequencing is error prone and the sequenced DNA vary between cells, the read sequences are compared pairwise to determine regions of approximate matches. To make it feasible, potentially matching regions between any read pair are selected on the presence of identical subwords of a given length k (k-mer). For the sake of efficiency, it is advantageous, if not compulsory, to index once for all the positions of all distinct k-mers in the reads. Once constructed, the index data structure is kept in main memory and repeatedly accessed to answer queries like 'given a k-mer, get the reads containing this k-mer (once/at least once)'. The question of indexing k-mers or subwords has long been addressed for large texts, however classical solutions like the generalized suffix tree, or suffix array require too much memory for a read collection. Even state of the art implementations of sparse hash tables (Google sparse hash) hit their limits with such data volumes.

To address the increasing demand for read indexing, we have developed a compact and efficient data structure, dubbed Gk arrays, that is specialized for indexing huge collections of short reads (the term 'collection', rather than 'set', underlines that the same read sequence can occur multiple times in the input). An in-depth evaluation has shown that Gk arrays, can be constructed in a time similar to the best hash tables, but outperform all concurrent solutions in term of memory usage (Figure 1). The Gk arrays combine three arrays: one for storing the sorted positions where true k-mers start, an inverted array that allows finding the rank of any k-mer from a position in a read, and a smaller array that records the intervals of positions of each distinct k-mer in sorted order. Although reads are concatenated for construction, Gk arrays avoid storing the positions of (artificial) k-mers that overlap

two adjacent reads. For instance, the query for counting the read containing an existing k-mer takes constant time. Several types of queries have been implemented and Gk arrays accommodate fixed as well as variable length reads. Gk arrays are packaged in an independent C++ library with a simple and easy to use programming interface (http://www.atgc-montpellier.fr/ngs/). They are currently exploited in a read mapping and RNA-sequencing analysis program; their scalability, efficiency, and versatility made them adequate for read error correction, read classification, k-mer counting in assembly program, or other HTS applications. Gk arrays can be seen as an indexing layer that is accessed by higher level applications. Future developments are planned to devise direct construction algorithms, or a compressed version of Gk arrays that, like other index structures, stores only some sampled positions and reconstruct the others at runtime, hence enabling the user to control the balance between speed and memory usage.

Gk arrays library is available on the ATGC bioinformatics platform in Montpellier: http://www.atgc-montpellier.fr/gkarrays

Link:
Gk arrays library:
http://www.atgc-montpellier.fr/gkarrays

Please contact:
Eric Rivals
LIRMM, CNRS, Univ. Montpellier II, France
E-mail: rivals@lirmm.fr
http://www.lirmm.fr/~rivals

A-Brain: Using the Cloud to Understand the Impact of Genetic Variability on the Brain

by Gabriel Antoniu, Alexandru Costan, Benoit Da Mota, Bertrand Thirion and Radu Tudoran

Joint genetic and neuroimaging data analysis on large cohorts of subjects is a new approach used to assess and understand the variability that exists between individuals. This approach, which to date is poorly understood, has the potential to open pioneering directions in biology and medicine. As both neuroimaging- and genetic-domain observations include a huge number of variables (of the order of 106), performing statistically rigorous analyses on such Big Data represents a computational challenge that cannot be addressed with conventional computational techniques. In the A-Brain project, researchers from Inria and Microsoft Research explore cloud computing techniques to address the above computational challenge.

Several brain diseases have a genetic origin, or their occurrence and severity is related to genetic factors. Genetics plays an important role in understanding and predicting responses to treatment for brain diseases like autism, Huntington's disease and many others. Brain images are now used to understand, model, and quantify various characteristics of the brain. Since they contain useful markers that relate genetics to clinical behaviour and diseases, they are used as an intermediate between the two. Currently, large-scale studies assess the relationships between diseases and genes, typically involving several hundred patients per study.

Imaging genetic studies linking functional MRI data and Single Nucleotide Polyphormisms (SNPs) data may face a dire multiple comparisons issue. In the genome dimension, genotyping DNA chips allow recording of several hundred thousand values per subject, while in the imaging dimension an fMRI volume may contain 100k-1M voxels. Finding the brain and genome regions that may be involved in this link entails a huge number of hypotheses, hence a drastic correction of the statistical significance of pair-wise relationships, which in turn results in a crucial reduction of the sensitivity of statistical procedures that aim to detect the association. It is therefore desirable to set up techniques that are as sensitive as possible to explore where in the brain and where in the genome a significant link can be detected, while correcting for family-wise multiple comparisons (controlling for false positive rate).

In the A-Brain project, researchers of the Parietal and KerData Inria teams jointly address this computational problem using cloud computing techniques on Microsoft Azure cloud computing environment. The two teams bring their complementary expertise: KerData (Rennes) in the area of scalable cloud data management and Parietal (Saclay) in the field of neuroimaging and genetics data analysis. The Map-Reduce programming model has recently arisen as a very effective approach to develop high-performance applications over very large distributed systems such as grids and now clouds. KerData has recently proposed a set of algorithms for data management, combining versioning with decentralized metadata management to support scalable, efficient, fine-grain access to massive, distributed Binary Large OBjects (BLOBs) under heavy concurrency. The project investigates the benefits of integrating BlobSeer with Microsoft Azure storage services and aims to evaluate the impact of using BlobSeer on Azure with large-scale application experiments such as the genetics-neuroimaging data comparisons addressed by Parietal. The project is supervised by the Joint Inria-Microsoft Research Centre.

Sophisticated techniques are required to perform sensitive analysis on the targeted large datasets. Univariate studies find an SNP and a neuroimaging trait