The nz.seq package for handling genetic sequences in the Netezza Performance Server

Przemysław Biecek[1,2,*], **Paweł Chudzian**[1,3], **Peter Gee**[1], **Jeff Foran**[1], **Justin Lindsey**[1]

1. NzLabs, Netezza, an IBM company,
2. Faculty of Mathematics, Informatics, and Mechanics, University of Warsaw,
3. Faculty of Electronics and Information Technology, Warsaw University of Technology

*Contact author: przemyslaw.biecek@gmail.com

Keywords: genetic sequences, parallel processing, relational database, next generation sequencing.

In the first part of this talk we will show why it is advantageous to store genetic sequences in a relational database instead of flat files. In the second part we will present the *R* interface supporting massively parallel operations on genetic sequences which are stored in the relational database. The presented implementation of **nz.seq** package is specific to the Netezza Performance Server (NPS) database but the presented approach is general and might be easily applied to any other database with a similar architecture, i.e. the parallel share nothing environment.

DNA, RNA and amino acid sequences are usually stored in flat files instead of relational databases. The three main reasons for that are: it is easier to copy or download flat files, it is easier to load flat files into a statistical packages like *R* and it is easier to use command line tools like SAMtools, bowtie or bwa. On the other hand storing data in a database allows one to incorporate the data structure, and allows for fast access to required sequences based on their properties or metadata information. Note that the volume of genetic data may be huge, dozens or thousands of terabytes, thus the access time to a required sequence may be significant. Moreover, databases support efficient merging of different kinds of data, e.g. genetic sequences might be combined with other features like sex, age, medical history, geographical coordinates etc.

In the NPS one can take advantages of both approaches. Sequences are stored in the relational database, but some operations might be performed in all computing nodes transparently to the internal database storage model.

The package **nz.seq** leverages the communication with the database in two ways:

- From the *R* client one can easily download any sequence or set of sequences from the remote database using their names, IDs or some matching criteria. Afterwards in the local *R* client one can operate on such sequences and store results from local operations in the remote database.
- Instead of downloading the data from the remote database to local *R* client, one can upload the *R* code from the local *R* client to the remote database. The submitted *R* script is then applied to all selected sequences in a fully parallel way. That decreases the overall computation time by reducing the data download time, which can be a significant improvement due to the large volume of data. Most of the *R* functions from the *R* packages available on CRAN or Bioconductor might be used in such remote operations. One can easily apply a variety of *R* operations to a large number of sequences stored in the remote database.

At the end of this talk we will present a real life application using data from The 1000 Genomes Project.