Next Generation Programming: software tools for NGS tertiary analysis

U. Pozzoli

Bioinformatics Lab, Scientific Institute I.R.C.C.S E. Medea, Bosisio Parini, Italy

Motivations

NGS experiments produce a huge amount of raw data and great efforts are currently ongoing to develop algorithms and workflows to manage this data-flood and to to produce reliable results. Computational tools for primary and secondary analysis are actually part of and evolving along with sequencing technologies. Still, the amount of results (i.e. genomic variations calls) is considerable and their interpretation far from an easy task. This is the most complex, experiment-specific and time-consuming phase of the NGS data analysis pipeline. In most laboratories, with Sanger sequencing, few Kb sequences are analyzed at a time and most of the functional analysis are performed manually, filtering out known SNPs, and using annotations from genome browsers to obtain hints about the possible functional impact of candidate genomic variations. Given their relatively complex usage, computational tools are rarely used to predict variations effect on biological function. Lab people tend to apply a similar approach for the analysis of NGS results and this leads to a marked preference for small targeted experiments. Targeting of small regions can be the proper approach when sequencing of small regions in a high number of samples is needed; nevertheless by comparing target enrichment and individual genome sequencing costs it is evident that this is going to change. The role of computational methods is going to be pivotal in the interpretation of NGS experiments. Despite the great variety of extremely useful algorithms their software implementation is lacking the characteristic that can make them readily applicable to NGS results. The vast majority of them is sequence driven and analyze sequences provided by the user which has to manually deal both with variations and annotation. This is inefficient and time consuming. Workflow management platforms can partially overcome these limitation but they still lack efficiency and often require adapter software layers to incorporate new algorithms. A set of software objects able to represent genomic annotation, computational features and variations is therefore needed to implement software readily suitable for NGS results.

Methods

Based on this premises we extended the previously described GeCo++ C++ Library [1] by adding a class, namely gGenotype that can describe genotypes for one or more samples deriving both from sequencing and genotyping experiments. The class serves as an interface between the software and a properly defined relational database containing genotype information in space efficient way (i.e. only variations from some reference are inserted). Any NGS sequencing experiment result set can be stored in the database and the results accessed from any software using the gGenotype class. Another class (gElement) gives a representation of a genomic element (namely a transcript, a set of gene with isoforms etc) that can be enriched with computed features. Variations can then being applied to the element obtaining a "mutated" version for which any feature is recalculated only where needed. Furthermore we developed a set of tools to obtain annotations from the most used public databases (i.e. UCSC Genome Browser and Ensembl), to read genotype information from vcf files and to interface the software with the sequencing database. We also developed a simple but effective application framework to write applications that can easily communicate (i.e. no need for format conversions) and being called remotely using http as a transfer protocol.

Results

We briefly present here the principles of GeCo++ library usage and, more extensively, two application examples: one in the field of population genetics and the other one to predict the effects of mutations on Transcription Factor Binding Sites.

Availability

http://gecolibrary.sourceforge.net/.

References

1. Cereda M, Sironi M, Cavalleri M, Pozzoli U: GeCo++: a C++ library for genomic features computation and annotation in the presence of variants. Bioinformatics 2011, 27(9):1313-5