Small tools for Bioinformatics

Pjotr Prins[1], Artem Tarasov[2] and Konstantin Tretyakov[3]

Affiliations: 1. Medical Genetics, University Medical Center Utrecht, The Netherlands; 2. Department of Statistical Simulation, St. Petersburg State University, Russia; 3. Institute of Computer Science, University of Tartu, Estonia
Contact E-mail: j.c.p.prins@umcutrecht.nl
URL: https://github.com/pjotrp/bioinformatics
Source code: https://github.com/pjotrp

Introduction
The small tools for Bioinformatics MANIFESTO is a grass-roots wake-up call for software developers which counters recent trends in providing largish 'monolithic' software solutions for bioinformatics without true free and open source software (FOSS) licenses. In this talk we present three tools together as a case study that represent the spirit of the MANIFESTO. These performance related tools have impact on designing and running NGS sequencing pipelines and are used today in major sequencing centers around the world. After speedily running ***sambamba once-only,*** using a **pfff** checksum, we discuss the overall philosophy of writing small tools for bioinformatics pipelines linking the design of these tools to the MANIFESTO.

Sambamba
With sambamba we set out to prove we could write an incarnation of samtools that is as efficient and can also make use of fine-grained parallelism to accelerate analysis. Sambamba is written in the D programming language which is a modern compiled programming language with run-time performance similar to that of C. D has powerful abstractions for parallel computing which it possible to easily scale sambamba with the number of cores until the point that input/output (I/O) hardware gets exhausted. Where samtools takes about 9 minutes to merge 3 BAM files of 1GB, sambamba takes 1 minute by utilising 12 cores. Sambamba is not only a fast alternative to samtools, but it also comes with extra functionality, a descriptive error handling system, a sophisticated look-ahead parser and powerful filtering.

Fast probabilistic file fingerprinting for big data
Biological data acquisition is raising new challenges both in data analysis and handling. Simply transferring files can be prohibitively slow due to their size. Common usage patterns, such as comparing and transferring, are proving computationally expensive and are tying down shared resources. Probabilistic Fast File Fingerprinting (pfff) exploits the variation present in biological data and computes fingerprints by sampling randomly from the file instead of reading it in full. Consequently, it has a flat performance characteristic correlated with data variation rather than file size. For file comparison probabilistic fingerprinting is as reliable as existing hashing techniques, such as MD5, with provably negligible risk of collisions.

Once-only
Once-only is inspired by the Lisp once-only function, which wraps another function and calculates a result only once if the inputs have not changed. Once-only makes a program or script only run once, provided the inputs do not change. This is very useful when running a range of jobs on a compute cluster or GRID. It may even be useful in the context of webservices. With once-only there are no worries about submitting serial jobs multiple times and rerunning a command when an input or output file changes. A mistake, an interruption, a hardware failure, or even a parameter tweak, does not mean everything has to be run again from scratch.

Discussion
Sambamba is a drop-in replacement of samtools. This is made possible by the fact that samtools adheres to the small tools design. Likewise, pfff replaces MD5 for large files. Once-only is incrementally beneficial for tools that run on the command line and can be submitted to a compute cluster. All three tools are small tools and were written in the Unix tradition by making software solutions self contained so they become modular and pluggable and can be easily replaced by a new generation of tools. As the MANIFESTO states: “Software is software. Software should be easy to change, replace and improve.”