Nextflow: a tool for deploying reproducible computational pipelines

Paolo Di Tommaso[1,2], Maria Chatzou[1,2], Pablo Prieto Barja[1,2], Emilio Palumbo[1,2], Cedric Notredame[1,2]

[1] Bioinformatics and Genomics Program, Centre for Genomic Regulation (CRG), Dr. Aiguader 88, 08003 Barcelona, Spain. Email: paolo.ditommaso@crg.eu

[2] Universitat Pompeu Fabra (UPF), 08003 Barcelona, Spain.

Project Website: http://nextflow.io
Source Code: https://github.com/nextflow-io/nextflow
License: GPLv3 - http://www.gnu.org/copyleft/gpl.html

Main Text of Abstract

Genomic pipelines usually rely on a combination of several pieces of third party research software. Academic software applications tend to be prototypes and are often difficult to install, configure and deploy. Furthermore their experimental nature can result in frequent updates, thus raising serious reproducibility issues.

Despite the fact that many tools have been developed to tackle these problems none of them have so far provided a comprehensive solution. For this reason we developed Nextflow, a tool that is specifically designed to address the reproducibility problem in computational pipelines by allowing researchers to easily write parallel and distributed data analysis applications. The three main strengths of Nextflow are:

- Its capacity to integrate any existing tools and scripts.
- Its support of a high-level parallelization model for complex task interactions and easy deployment.
- Increased reproducibility thanks to its reliance on Docker containers technology.

A Nextflow pipeline is made by putting together several processes. Each process can be written in any scripting language that can be executed by the Linux platform (BASH, Perl, Ruby, Python, etc.). Parallelization is automatically managed by the framework and it is implicitly defined by the processes input and output declarations.

Moreover Nextflow provides an abstraction over the underlying execution platform. Thus, the resulting pipeline can run on a single workstation, on different grid infrastructures or in a cloud environment.

The integration with Docker containers technology and the Github sharing platform enables pipelines to be deployed, along with all their dependencies, across multiple platforms without any modifications, making it possible to share them and replicate their results in a predictable manner. In addition Nextflow provides a rich set of built-in functions for recurrent operations on common bioinformatics data formats (FASTA, FASTQ, etc.) such as split, count, filter, combine, etc.

Finally the Nextflow programming model greatly simplifies writing large scalable pipelines, by utilizing a scatter-process-gather parallelization strategy that is quite common in bioinformatics applications.

We used this approach in PIPER, a pipeline for the detection and mapping of long non-coding RNAs. We managed to speed-up the overall pipeline execution by 6 times and to reduce the application code base in a significant manner when compared to the previous PERL based implementation.