Identification of functionally active genomic features relevant to phenotypic diversity and plasticity in cattle

Deliverable 3.2

FIRST PROTOTYPE OF A NORMALIZED ENCODE PIPELINE

Grant agreement no°: 815668

Due submission date

2020-08-31

Actual submission date

2020-08-11

Responsible author(s)

J. Espinosa Carrasco, CRG (joseantonio.espinosa@crg.eu)

C. Notredame, CRG (cedric.notredame@crg.eu)

Confidential: No

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 815668. The content of this report reflects only the author's view. The European Commission is not responsible for any use that may be made of the information it contains.

DOCUMENT CONTROL SHEET

Deliverable name	First prototype of a normalized ENCODE pipeline
Deliverable number	D3.2
Partners providing input to this Deliverable	CRG, FMV, INRAE
Draft final version circulated by lead party to: On date	FMV, INRAE (2020-07-27)
Approved by (on date)	FBN as Coordinator (2020-07-28)
Work package no	3
Dissemination level	Public (PU)

REVISION HISTORY

Version number	Version date	Document name	Lead partner
V1	2020-07-27	D3.2_v1_CRG	CRG
V2	2020-07-28	D3.2_v2_CRG_FMV_INRAE	CRG
Final	2020-08-10	D3-2-final	CRG

Changes with respect to the DoA (Description of Action)

No changes

Dissemination and uptake

This deliverable is for public use.

Table of Content

1. Summary of Results

The nf-core project partners, a community-based effort to collect Nextflow best-practice bioinformatics pipelines implemented with Nextflow as workflow manager, have established a set of standards to deliver reproducible and interoperable data analyses pipelines. In this BovReg deliverable (D3.1), we describe how these standards provide a perfect framework to develop BovReg reference pipelines for the annotation of the functional genome of cattle. Besides, the nf-core community leads an effort to collect a set of best-practice bioinformatics pipelines following these standards. We show in this deliverable, how, as part of this effort, pipelines are available on the nf-core website for many of the bioinformatics analyses to be implemented within BovReg. In this manner, these pipelines can be used as the basis for the development of the BovReg reference pipelines. Together with following nf-core standards, this framework should guarantee that BovReg pipelines adhere to current computational best practices and thus, represent a long-lasting resource for the community.

2. Introduction

BovReg aims to characterize the cattle functional genome. For this purpose, there is a need to implement several genomic analysis pipelines. The goal of BovReg is for these pipelines to become the reference (or normalized) computational resource for the reannotation of cattle genomes. Thus, the uptake of FAIR (findable, accessible, interoperable and reusable) research principles for scientific reproducibility (Wilkinson et al. 2016) will permit to develop pipelines following best practices regarding computational reproducibility, interoperability. It will furthermore enable the reuse of these pipelines when new data becomes available.

Flagship human genomic projects such as ENCODE (Dunham et al. 2012) and GTEX (Ardlie et al. 2015) have previously implemented reference bioinformatics analyses. Nevertheless, since the time when these projects were carried out the field of computational biology has seen an explosion of powerful new standards for reproducibility and interoperability (Grüning et al. 2019; Lamprecht et al. 2019). Hence, the porting of several of these pipelines becomes the first step towards our goal of implementing fully normalized bioinformatics workflows within the framework of BovReg.

Workflow manager systems have recently become the basis to implement interoperable reproducible-analysis pipelines (Baichoo et al. 2018). This is achieved thanks to features such as the native capability to enable pipeline portability between different computational infrastructures and the integration of container technologies (Di Tommaso et al. 2015). Thus, workflow managers like Snakemake (Köster and Rahmann 2012) or Nextflow (Di Tommaso et al. 2017) constitute the backbone of normalized workflows. The Nextflow community has been especially active in this regard and as a natural consequence, nf-core has emerged as a prime example of a community-based effort to collect and curate best-

practices analysis pipelines (Ewels et al. 2020). Since the early stages of the project, the Comparative Bioinformatics group at BovReg partner CRG has been part of this initiative as much by the creation and maintenance of Nextflow itself as by its active contribution to nf-core.

3. Core Report

The pipelines collected by nf-core guarantee its full portability, reproducibility, and interoperability by enforcing a set of minimal requirements (Ewels et al. 2020). BovReg shares this same objective since its reference pipelines should be normalized to:

1) allow its deployment in most popular cloud providers as well as HPC clusters,

2) enable the consistent reproduction of its results and,

3) guarantee the interoperability between the different yielded results.

To this end, during the FAANG Shared Workshop meeting at Hinxton (UK), which took place in February 2020, BovReg agreed on adopting nf-core as the standard for the implementation of reference BovReg analysis pipelines. nf-core requirements are summarized in Annex 1 to this document. BovReg reference pipelines will adhere to these guidelines to ensure that they follow best practices for scientific-computing (Möller et al. 2017; Grüning et al. 2018).

The points covered by nf-core guidelines will guarantee that BovReg reference pipelines achieve the above-enumerated standardization goals. Briefly, the implementation of the pipelines using Nextflow makes them platform agnostic (addressing point 1 above) enabling its deployment in virtually any computational environment from a local machine (testing mode) to the cloud. Besides, Nextflow natively integrates container technologies, such as Docker (http://www.docker.com) and Singularity (https://sylabs.io/), and package and software environment management systems such as Conda (https://conda.io).

Enforcing the use of these solutions ensures that the same computational results are reproduced by running the analysis exactly in the same computational environment and thus provides a solution to point 2 above (Di Tommaso et al. 2015; da Veiga Leprevost et al. 2017).

Furthermore, nf-core guidelines recommend using standard file formats, to share a common pipeline structure, and to thoroughly document the pipelines, three key aspects to enable the interoperability of the results of the different pipelines (point 3 above). Finally, further points of the guidelines such as the requirement of a minimal test dataset and the need to pass the automated continuous-integration test using these data, among others, provide the basis to develop pipelines following existing computational best practices. (See Annex 1 for a more detailed description).

We set as the first step towards the standardization of BovReg reference pipelines, the porting of ENCODE project analysis pipelines following current best practices in terms of computational reproducibility and interoperability. In this manner, the ported pipelines should become the backbone to build BovReg normalized pipelines adapted for our specific needs and data (i.e. cattle). Notably, as a result of nf-core community effort to collect and curate a set of best-practice analysis pipelines, many of the ENCODE analysis pipelines (https://www.encodeproject.org/pipelines/) find their counterpart implementation on nf-core (https://nf-co.re/pipelines) as summarized in Table 1. These standardized pipelines could thus represent the backbone for the development of BovReg reference pipelines.

Type of analysis	Data subtype	ENCODE link	Equivalent nf-core pipeline	nf-core status
RNA-seq	RNA-seq of long RNAs (paired-end, stranded)	https://www.encodeproject.org/pipelines/ENCPL002LPE/	https://nf-co.re/rnaseq	Released v1.4.2
	RNA-seq of long RNAs (single-end, unstranded)	https://www.encodeproject.org/pipelines/ENCPL002LSE/		
	Small RNA-seq single-end pipeline	https://www.encodeproject.org/pipelines/ENCPL337CSA/	https://nf-co.re/smrnaseq	Released v1.0.0
	microRNA-seq pipeline	https://www.encodeproject.org/pipelines/ENCPL444CYA/		
CAGE-seq	RAMPAGE or CAGE (paired-end)	https://www.encodeproject.org/pipelines/ENCPL122WIM/	https://nf-co.re/cageseq	Under development
ChIp-seq	Histone ChIP-seq (unreplicated)	https://www.encodeproject.org/pipelines/ENCPL841HGV/	https://nf-co.re/chipseq	Released v1.2.0
	Histone ChIP-seq (replicated)	https://www.encodeproject.org/pipelines/ENCPL272XAE/		
	Transcription factor ChIP-seq	https://www.encodeproject.org/pipelines/ENCPL138KID/		
	Transcription factor ChIP-seq (unreplicated)	https://www.encodeproject.org/pipelines/ENCPL493SGC/		
ATAC-seq	Dnase-seq (single-end)	https://www.encodeproject.org/pipelines/ENCPL201DNS/	Not available	Not available
	Dnase-seq (paired-end)	https://www.encodeproject.org/pipelines/ENCPL202DNS/	Not available	Not available
	ATAC-seq	https://www.encodeproject.org/pipelines/ENCPL792NWO/	https://nf-co.re/atacseq	Released v1.2.0
WGBS	Whole-Genome Bisulfite Sequencing (WGBS)	https://www.encodeproject.org/pipelines/ENCPL210QWH/	https://nf-co.re/methylseq	Released v1.5.0
HiC-seq	HiC-seq	NA	https://nf-co.re/hic	Released v1.2.1
GWAS	GWAS	NA	https://nf-co.re/gwas	

Table 1 - Equivalence between the ENCODE project and nf-core pipelines

As depicted on the table, only Dnase-seq ENCODE analysis pipelines do not find their respective equivalent pipeline on nf-core, likely due to the displacement of Dnase-seq by ATAC-seq as the preferred chromatin accessibility assay (Buenrostro et al. 2013). It should be noted that nf-core includes a pipeline designed for ATAC-seq analysis (https://nf-co.re/atacseq), which is a key method within the BovReg project. All the remaining ENCODE reference pipelines are also available at nf-core, although not always on a one-to-one relationship. This is due to the fact that in some cases ENCODE analyses, although listed separately, are performed by the same pipeline parameterized in a different manner. This is the case of the so-called "RNA-seq of long RNAs analyses", where in fact, both "paired-end" and "single-end" ENCODE pipelines lead to the same GitHub repository (https://github.com/ENCODE-DCC/long-rna-seq-pipeline). This also holds for ChIp-seq ENCODE analyses which are released on the following repository (https://github.com/ENCODE-DCC/chip-seq-pipeline).

Task 3.1 identified the bioinformatics analyses that will be required to be implemented within BovReg. As we summarized in Table 1, most of these bioinformatics analyses can use a corresponding ENCODE and nf-core pipeline. However, this is not the case for the Hi-C seq and GWAS analyses, which, although identified in our survey, were not released in the framework of the original ENCODE effort. Notably, the nf-core community provides a stable release of a pipeline for the analysis of Hi-C seq data (https://nf-co.re/hic) and a development version of a GWAS pipeline (https://nf-co.re/gwas). Finally, the methylseq pipeline (https://nf-co.re/methylseq) can not only analyse Whole-Genome Bisulfite Sequencing (WGBS), as depicted in the table, but also reduced representation Bisulfite sequencing (RRBS). This is important since these are sequencing data to be produced in the framework of the BovReg project.

4. Conclusions

In this document we show, how most of the computational analyses planned within the BovReg project can use corresponding nf-core pipelines. They will therefore constitute the basis for BovReg reference pipelines development. nf-core was chosen, because it adheres to best practices in terms of computational reproducibility, interoperability, and distributability. Such high scientific-computing standards are achieved by following a set of requirements and recommendations recently published (Ewels et al. 2020) as summarized in Annex 1. Following nf-core guidelines, as agreed during the FAANG Shared Workshop meeting at Hinxton, will ensure that BovReg reference pipelines result in a permanent resource for the community for the reannotation of cattle functional genome.

5. References

Ardlie, K. G., D. S. Deluca, A. V. Segre, T. J. Sullivan, T. R. Young, E. T. Gelfand, C. A. Trowbridge, et al. 2015. “The Genotype-Tissue Expression (GTEx) Pilot Analysis: Multitissue Gene Regulation in Humans.” Science 348 (6235): 648–60.

Baichoo, Shakuntala, Yassine Souilmi, Sumir Panji, Gerrit Botha, Ayton Meintjes, Scott Hazelhurst, Hocine Bendou, et al. 2018. “Developing Reproducible Bioinformatics Analysis Workflows for Heterogeneous Computing Environments to Support African Genomics.” BMC Bioinformatics 19 (1): 457.

Buenrostro, Jason D., Paul G. Giresi, Lisa C. Zaba, Howard Y. Chang, and William J. Greenleaf. 2013. “Transposition of Native Chromatin for Multimodal Regulatory Analysis and Personal Epigenomics.” Nature Methods 10 (12): 1213.

Di Tommaso, Paolo, Maria Chatzou, Evan W. Floden, Pablo Prieto Barja, Emilio Palumbo, and Cedric Notredame. 2017. “Nextflow Enables Reproducible Computational Workflows.” Nature Biotechnology 35 (4): 316–19.

Di Tommaso, Paolo, Emilio Palumbo, Maria Chatzou, Pablo Prieto, Michael L. Heuer, and Cedric Notredame. 2015. “The Impact of Docker Containers on the Performance of Genomic Pipelines.” PeerJ 3 (September): e1273.

Dunham et al. 2012. “An Integrated Encyclopedia of DNA Elements in the Human Genome.” Nature 489 (7414): 57–74.

Ewels, Philip A., Alexander Peltzer, Sven Fillinger, Harshil Patel, Johannes Alneberg, Andreas Wilm, Maxime Ulysse Garcia, Paolo Di Tommaso, and Sven Nahnsen. 2020. “The Nf-Core Framework for Community-Curated Bioinformatics Pipelines.” Nature Biotechnology 38 (3): 276–78.

Grüning, Björn A., Samuel Lampa, Marc Vaudel, and Daniel Blankenberg. 2019. “Software Engineering for Scientific Big Data Analysis.” GigaScience 8 (5). https://doi.org/10.1093/gigascience/giz054.

Grüning, Björn, John Chilton, Johannes Köster, Ryan Dale, Nicola Soranzo, Marius van den Beek, Jeremy Goecks, Rolf Backofen, Anton Nekrutenko, and James Taylor. 2018. “Practical Computational Reproducibility in the Life Sciences.” Cell Systems 6 (6): 631–35.

Köster, Johannes, and Sven Rahmann. 2012. “Snakemake—a Scalable Bioinformatics Workflow Engine.” Bioinformatics 28 (19): 2520–22.

Lamprecht, Anna-Lena, Leyla Garcia, Mateusz Kuzak, Carlos Martinez, Ricardo Arcila, Eva Martin Del Pico, Victoria Dominguez Del Angel, et al. 2019. “Towards FAIR Principles for Research Software.” Data Science, no. Preprint: 1–23.

Möller, Steffen, Stuart W. Prescott, Lars Wirzenius, Petter Reinholdtsen, Brad Chapman, Pjotr Prins, Stian Soiland-Reyes, et al. 2017. “Robust Cross-Platform Workflows: How Technical and Scientific Communities Collaborate to Develop, Test and Share Best Practices for Data Analysis.” Data Science and Engineering 2 (3): 232–44.

Veiga Leprevost, Felipe da, Björn A. Grüning, Saulo Alves Aflitos, Hannes L. Röst, Julian Uszkoreit, Harald Barsnes, Marc Vaudel, et al. 2017. “BioContainers: An Open-Source and Community-Driven Framework for Software Standardization.” Bioinformatics 33 (16): 2580–82.

Wilkinson, Mark D., Michel Dumontier, I. Jsbrand Jan Aalbersberg, Gabrielle Appleton, Myles Axton, Arie Baak, Niklas Blomberg, et al. 2016. “The FAIR Guiding Principles for Scientific Data Management and Stewardship.” Scientific Data 3 (March): 160018.

6. Annexes

Annex 1 - Coding Guidelines for BovReg Reference Pipelines

BovReg agreed to use nf-core as the standard for the implementation of the reference bioinformatics pipelines during the FAANG Shared Workshop meeting at Hinxton. With these guidelines, we aim to ensure that BovReg production pipelines follow a standard best-practice implementation. The Guidelines are listed in the following requirements:

Built using Nextflow. Pipelines will be implemented using Nextflow to enable data analysis reproducibility and portability.

Have a double OSS license Apache 2.0 and MIT. We agreed on the use of both Apache 2.0 and MIT licenses, since the consortium agreed to use the former and nf-core requires the latter.

Software bundled using Docker / Singularity. The software should be containerized using either Docker or Singularity. The preferred solution will be to provide an environment Conda file listing all pipeline software requirements and use this file to create the Conda environment within the container. In this manner, pipelines can support users running Conda, Docker or Singularity.

Continuous integration testing. All the workflows should include continuous integration testing such as Travis, GitHub Actions, etc.

Pass nf-core lint test. These tests guarantee that nf-core pipelines adhere to a common file structure. Of note, nf-core provides a command-line tool to evaluate that pipelines are compliant with this structure.

Stable release tags. Each stable version of the pipeline should be released with its corresponding tag and DOI to ensure code traceability and should be made openly available on GitHub FAANG repository.

Include a minimal test dataset. Project repository should include a minimal dataset that allows the quick testing of workflow execution.

Common pipeline structure and usage. Reference pipelines should maintain a common code organization.

Run in a single command. This should simplify automatic pipeline deployment.

Excellent documentation. Pipelines should be distributed with comprehensive documentation, i.e., pipelines documentation should allow users to run pipelines after spending a reasonable amount of time reading the documentation.

A responsible point of contact / GitHub username. Reference pipelines should be maintained.

And the following recommendations:

Software bundled using Bioconda. Use a Conda environment script to list all tools used by the pipeline as explained in point "**Software bundled using Docker / Singularity**".

Require as little input metadata as possible. Pipelines will include optional pipeline steps to convert file types and to build reference indexes.

Optimized output file formats. When possible, reference pipelines should use standard file formats.

Explicit support for running in cloud environments. Workflows should be portable and this implies that it should be possible to run reference pipelines in different computational environments from a local computer to HPC environments, including the Embassy cloud and widely used cloud solutions.

Benchmarks from running on cloud environments. It is recommendable to include tests that ensure that the reference pipelines are deployable in cloud environments when possible.

Annex 1 references:

Ewels, P.A., Peltzer, A., Fillinger, S. et al. The nf-core framework for community-curated bioinformatics pipelines. Nat Biotechnol 38, 276–278 (2020). https://doi.org/10.1038/s41587-020-0439-x