GENE-SWitCH

The regulatory GENomE of SWine and CHicken: functional annotation during development

Deliverable D2.1
Reproducible, scale-able workflows for FAANG data analysis

Deliverable leader: UEDIN

Authors: M. Watson (UEDIN), A. Law (UEDIN), A. Archibald (UEDIN), R. Kuo (UEDIN), P. Harrison (EMBL), F. Martin (EMBL), A. Sokolov (EMBL), S. Foissac (INRAE), S. Djebali (INSERM), C. Kurylo (INRAE), O. Madsen (WU), J. de Wos (WU)

Version: 1.1

Due date of deliverable (as in DoA):	M6
Actual submission date:	03/08/2021, month M26

Dissemination level:

PU Public	**X**
CO Confidential, only for members of the consortium (including the Commission Services)	

Research and Innovation Action, SFS-30-2018-2019-2020 Agri-Aqua Labs
Duration of the project: 01 July 2019 – 30 June 2023, 48 months

Revision History

This document contains a revision history log. When changes occur, the document's revision history log will reflect an updated version number, the date of the new version, the author making the change, and a summary of the changes.

Version	Date	Author	Summary of changes
V1	31 January 2020 (M7)	M. Watson (UEDIN), A. Law (UEDIN), A. Archibald (UEDIN), R. Kuo (UEDIN), P. Harrison (EMBL), F. Martin (EMBL), A. Sokolov (EMBL), S. Foissac (INRAE), S. Djebali (INSERM), C. Kurylo (INRAE), O. Madsen (WU), J. de Wos (WU)	Original
V1.1	22 July 2021 Submitted on 3 August 2021 (M26)	M. Watson (UEDIN), A. Law (UEDIN), A. Archibald (UEDIN) S.Guizard (UEDIN), P. Harrison (EMBL), F. Martin (EMBL), A. Sokolov (EMBL), S. Foissac (INRAE), S. Djebali (INSERM), C. Kurylo (INRAE), O. Madsen (WU), J. de Wos (WU)	Full description of Reproducible, scale-able workflows for FAANG data analysis and some examples of their use

Table of contents

1 Summary

Objectives

The overall aim of Work Package 2 (New annotation maps of the pig and chicken genomes) is to identify and annotate the elements in the pig and chicken genome that are functional. Such functional elements include DNA sequences that encode proteins. These coding sequences are transcribed into messenger RNA molecules that are in turn translated into proteins and are commonly known as genes or protein-coding genes. However, the final product from some genes are RNA molecules and these genes are known as non-coding genes. Functional elements also include DNA sequences that are regulatory in nature and their functions include determining whether, when and where genes are expressed. Inferences about the functional elements of a genome can be drawn from analysis of a range of assays of both genomic DNA and transcribed RNA molecules. These assays generate large complex DNA / RNA sequence datasets.

The Work Package 2 objective, for which this D2.1 is the deliverable, is to produce the data analysis tools to analyse these large sequence datasets to facilitate the identification of the functional elements in the pig and chicken genomes.

Method

There are a range of data analysis or bioinformatics tools and software packages available in the public domain for the analysis of DNA and RNA sequence data. We agreed a set of principles for software development within the project. As GENE-SWitCH is one of a number of projects that will contribute to the global Functional Annotation of Animal Genomes (FAANG) initiative, we have also consulted key players from other FAANG projects. We identified relevant packages and analysis pipelines for assessment, modification as necessary and deployment for analysis of the datasets that are being generated by other Work Packages (WP1, WP4 and WP5) in the GENE-SWitCH project. As described below, the software pipelines that we are deploying comprise nf-core pipelines or modifications thereof.

Main Results

We have agreed a document summarising the principle for software development within the project (Annex 1).

We have established a shared hardware platform on the EMBL-EBI Embassy Cloud (https://www.embassycloud.org/) on which the software development and deployment is enabled (MS7; D3.4 in progress).

We have established a collection of datasets for testing and benchmarking the software packages.

Following earlier identification and evaluation of existing pipelines (publicly available, widely used and maintained) we have established a collection of software packages for the analysis of short read RNA sequence data (RNA-seq), long read RNA sequence data (Iso-seq), small RNA sequence data (smrna-seq) methylation sequence data (methyl-seq) and ATAC-seq data. These software packages include software developed by others, primarily software from the nf-core framework of community-curated bioinformatics pipelines, further modifications of such software and software developed by the GENE-SWitCH partner organisations.

We have posted the relevant software packages on the FAANG GitHub (https://github.com/FAANG), using the GENE-SWitCH project acronym ‘proj-gs-‘ , for example the GENE-SWitCH RNA-Seq pipeline at https://github.com/FAANG/proj-gs-rna-seq.

Teams involved

UEDIN, EMBL, INRAE, INSERM, WU

2 Introduction

Bioinformatics is the process of converting DNA/RNA sequence data into information and knowledge. Bioinformatics tools include algorithms, software packages and pipelines for the analysis of sequence data. As sequencing technologies have evolved over the past decade and more, it has been necessary to develop new tools to analyse the resulting sequence datasets. The rapidly evolving sequencing technologies present significant challenges for data analysis, including exponential growth in the number and size of datasets and differences in the accuracy and error models of the different technologies. These challenges are being met by parallel rapid development of new algorithms, software packages and pipelines.

Reproducibility is a cornerstone of the scientific approach. Thus, for experimental laboratory work it is critical that the methods used to generate experimental data are described in sufficient detail to allow replication of the experiment. Similarly, the data analysis methods and tools need to be described in sufficient detail to allow the analyses to be repeated by others. The sharing of sequence data has been part of the ethos of genome projects since the advent of the Human Genome Project and is a key principle of the global Functional Annotation of ANimal Genomes (FAANG) initiative of which GENE-SWitCH is a contributing project. Arguably, this sharing of data allows the replication of data analyses on a scale unprecedented in the history of science. However, the capacity to replicate / reproduce sequence data analyses is critically dependent on access to the relevant bioinformatics tools.

The aims of the FAANG initiative are being delivered by a series of independently funded projects around the world, including three H2020 projects (GENE-SWitCH, BovReg, Aqua-FAANG) rather than as a single enterprise. In order to integrate data and results for the same species and to allow comparative analyses across species it is highly desirable that common software pipelines are used for the data analyses.

This deliverable is concerned with enabling reproducible consistent data analysis and subsequent data integration and comparative analyses.

3 Results

3.1 Pipeline development guidelines, standards and coding principles

Agreeing guidelines for software pipeline development is an essential first step towards establishing a set of standard shared software tools. The processing and analysis of sequence data typically involves the use of multiple software tools and programs. Outputs from one analysis program become input for a subsequent program and so on. Whilst the passing of output data from one program to the next program for further processing and analysis can be undertaken manually, it is more efficient, robust and reproducible to link the programs into semi-automated pipelines. Workflow managers are software tools for linking programs in this manner. The successful operation of software tools typically have dependencies with respect to the environment, operating system and the availability of various files and libraries. Many of these environmental factors change over time potentially rendering the program unworkable in the new environment or potentially changing the result of the analysis in an unpredictable or undetectable manner. In order to preserve the capability to reproduce analyses it is highly desirable to isolate the software programs and pipelines from such changes in the computing environment. Containerisation offers a solution to this problem. A software container includes all the code and its dependencies so that the software always runs the same regardless of the computing infrastructure and environment used. There are multiple computing languages used in bioinformatics software. In seeking to establish a set of standard analysis tools for GENE-SWitCH and wider FAANG it is desirable that the range of languages used is minimised.

We agreed a set of guidelines, standards and coding principles for the development of analysis pipelines (Annex 1). These guidelines address the issues highlighted above. Briefly, the preferred languages to be used are Python 3 and Java 8, the Nextflow workflow manager

(https://www.nextflow.io/) should be used to link the analysis tools into pipelines and the pipelines should be containerised in Docker (https://www.docker.com/). The guidelines also address the issue of sharing the analyses tools via the FAANG public GitHub repository (https://github.com/FAANG), containerisation and the use of open source software.

3.2 Pipeline development and software catalogue

There are multiple software tools available in the public domain for the analysis of sequence data. These tools have been deployed in projects similar to FAANG and GENE-SWitCH including ENCODE and Blueprint (http://www.blueprint-epigenome.eu/). Some of these tools are already available as pipelines and workflows.

The software pipelines, workflows and tools developed or compiled for analysis of GENE-SWitCH (FAANG) data are available via the FAANG GitHub (https://github.com/FAANG). The workflows / pipelines currently available are:

3.2.1 TAGADA: Transcripts And Genes Assembly, Deconvolution, Analysis

Summary

The TAGADA RNA-seq pipeline, for Transcripts And Genes Assembly, Deconvolution, Analysis, has been specifically developed for the GENE-SWitCH project. While many RNA-seq pipelines are currently available to build *de novo* gene models or quantify gene expression levels using a provided gene annotation, few allow both transcript reconstruction and expression assessment from RNA-seq data in a reproducible way. This is why a dedicated tool was needed.

In brief, TAGADA combines several reference RNA-seq bioinformatics tools into a containerized pipeline. It maps reads with STAR, reconstructs and quantifies genes and transcripts with StringTie and detects and characterizes long non-coding RNAs (lncRNAs) with FEELnc.

TAGADA uses the Nextflow framework in line with the nf-core specifications. It provides a containerized environment that makes it compatible with a variety of high-performance computing platforms and workload orchestrators. TAGADA is designed to be easy to use and flexible. As such, TAGADA requires a minimal set of inputs: a set of RNA-seq read files, a reference genome and its gene annotation. Optionally, a simple tabulated metadata file can also be provided to describe the experimental design and seamlessly merge samples according to specified factors.

The pipeline automatically generates a large variety of quality controls in the form of interactive charts and tables with statistics and metrics for various steps of the workflow. Expression tables are also provided with read counts for annotated and predicted genes and transcripts allowing further comparative expression analyses. We believe that the TAGADA pipeline offers a useful, powerful and easy-to-use way to process RNA-seq data, nicely complementing the existing nf-core catalogue of bioinformatics tools and contributing to the FAANG global action.

Analysis workflow

The TAGADA pipeline executes the following main processes:

1. Control reads **quality** with FastQC (https://github.com/s-andrews/FastQC).
2. **Trim** adaptators from reads with Trim Galore (https://github.com/FelixKrueger/TrimGalore).
3. Estimate **overhang** length of splice junctions, and **index** the genome sequence with STAR (https://github.com/alexdobin/STAR).
   The indexed genome is saved to “output/index”.
4. **Map** reads to the indexed genome with STAR.
   The mapped reads are saved to “output/maps”.
5. Estimate **direction** and **length** of mapped reads, and compute genome **coverage** with Bedtools (https://github.com/arq5x/bedtools2).

The coverage files are saved to “output/coverage”.

6. **Merge** mapped reads by factors with Samtools (https://github.com/samtools/samtools).
   See the merge factors section for details.
7. **Assemble** transcripts and **combine** them into a novel annotation with StringTie (https://github.com/gpertea/stringtie).
   The novel annotation is saved to “output/annotation”.
8. **Detect** long non-coding RNAs with FEELnc (https://github.com/tderrien/FEELnc).
   The annotations of long non-coding RNAs are saved to ”output/annotation”.
9. **Quantify** genes and transcripts with StringTie, and **format** them into tabulated files.
   The TPM values and read counts for each annotation are saved to “output/quantification”.
10. Aggregate quality controls into a **report** with MultiQC (https://github.com/ewels/MultiQC).
    The report is saved to “output/control”.

To run the pipeline a command-line execution is required in the form: ./nextflow-run FAANG/analysis-TAGADA --revision 1.0.1 --output directory --profile test docker

Code repository

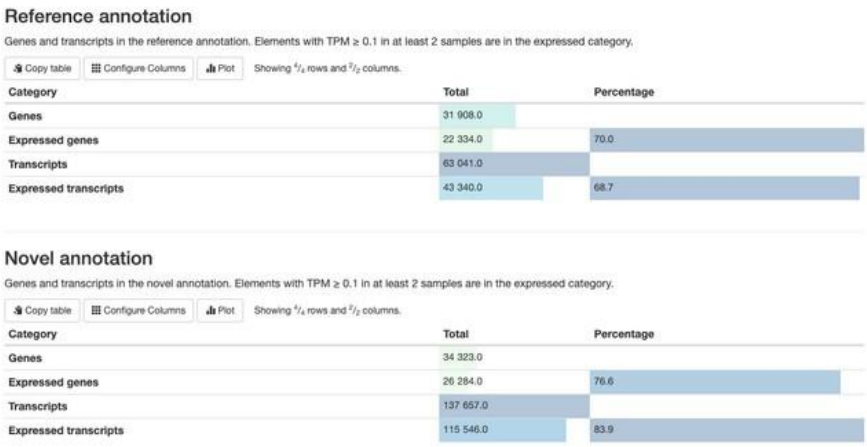
The TAGADA analysis pipeline is developed in the form of an open-source software. Its code is publicly available on the Github code repository at **https://github.com/FAANG/analysis-TAGADA**. It is developed under the Apache-2-0 License. It is mainly coded using the NextFlow language (https://www.nextflow.io/) and offers a containerized execution, as per the GENE-SWitCH WP2 Code and Procedure’s agreement.

Results

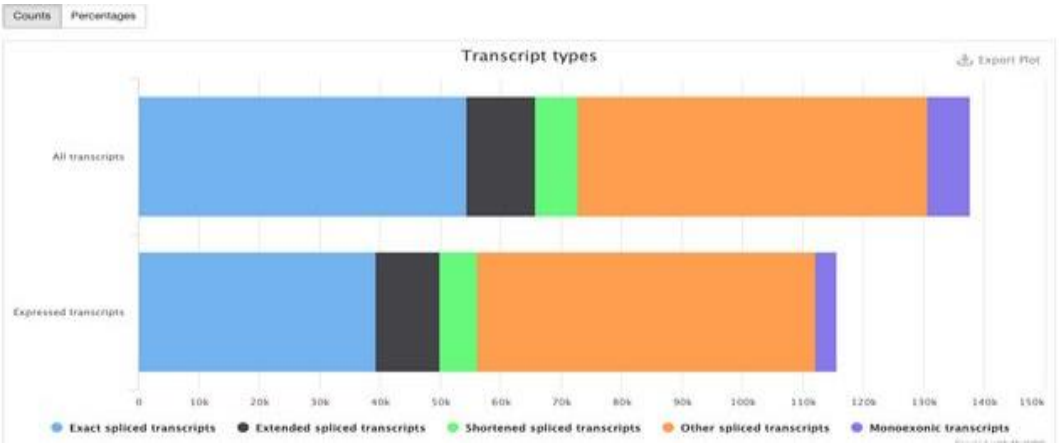
In addition to the results and log files, an interactive html-based report is provided for each execution to summarize the main Quality Controls (QC) and analysis results. This report is a stand-alone file that can easily be shared with the project’s partners, sent to external collaborators, or published online.

The report presents the following QCs and results, corresponding to successive analysis steps. The illustrations were generated from analysis of RNA-Seq data produced within GENE-SWitCH WP1:

- A comparison between genes and transcript counts in the reference annotation provided to the pipeline and the novel annotation generated by the pipeline:



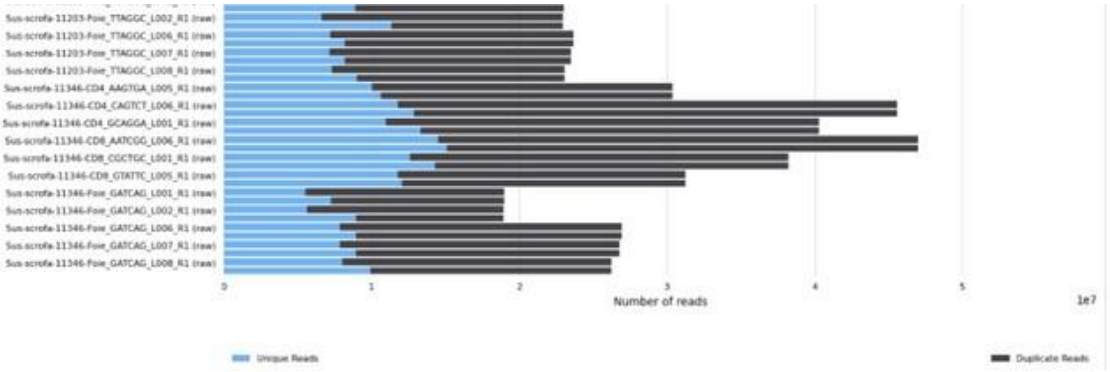
- How transcripts from the novel annotation differ from the transcripts of the reference annotation:



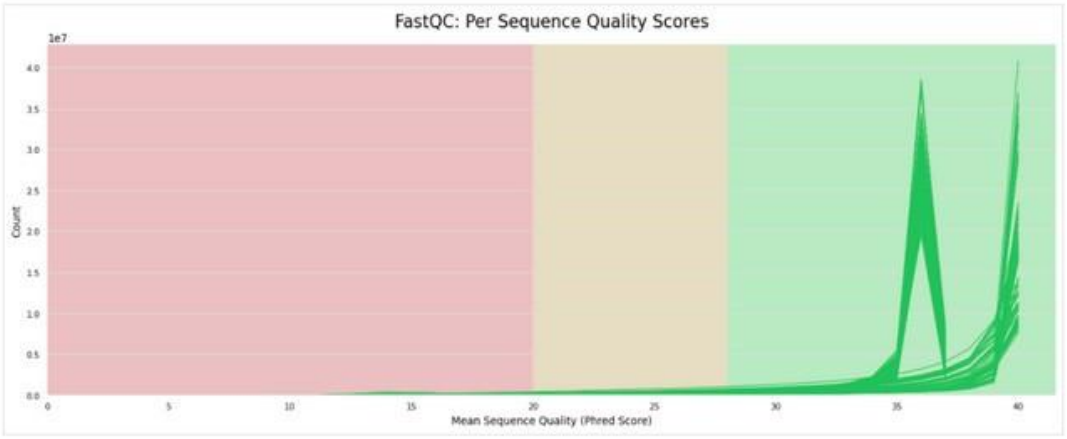
- A summary of the metadata describing the samples:

input	batchno	animal	tissue
Sus-scrofa-11346-CD4_AAGTGA_L005	1.0	pig2	cd4
Sus-scrofa-11346-CD4_CAGTCT_L006	2.0	pig2	cd4
Sus-scrofa-11346-CD4_GCAGGA_L001	3.0	pig2	cd4
Sus-scrofa-10886-CD4_ATTCCG_L006	1.0	pig3	cd4
Sus-scrofa-10886-CD4_CTGGTT_L001	2.0	pig3	cd4
Sus-scrofa-10886-CD4_GTCTGG_L005	3.0	pig3	cd4
Sus-scrofa-10999-CD4_CAAAAA_L001	1.0	pig4	cd4
Sus-scrofa-10999-CD4_CCTTTT_L005	2.0	pig4	cd4
Sus-scrofa-10999-CD4_CGTGTG_L006	3.0	pig4	cd4
Sus-scrofa-11203-CD4_AGTCGC_L006	1.0	pig1	cd8
Sus-scrofa-11203-CD4_GATTCA_L001	2.0	pig1	cd8
Sus-scrofa-11203-CD4_GCCCTG_L005	3.0	pig1	cd8
Sus-scrofa-11203-CD8_CACGTT_L001	4.0	pig1	cd8
Sus-scrofa-11203-CD8_CTTCCA_L005	5.0	pig1	cd8
Sus-scrofa-11203-CD8_TCCCCC_L006	6.0	pig1	cd8

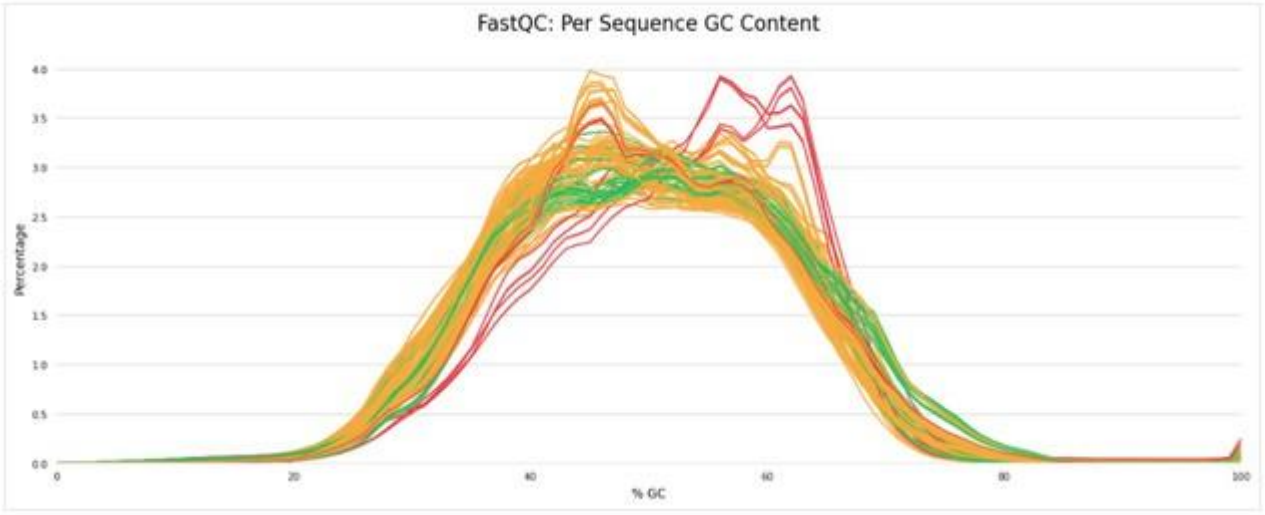
- The counts of unique (blue) and duplicated (black) reads:



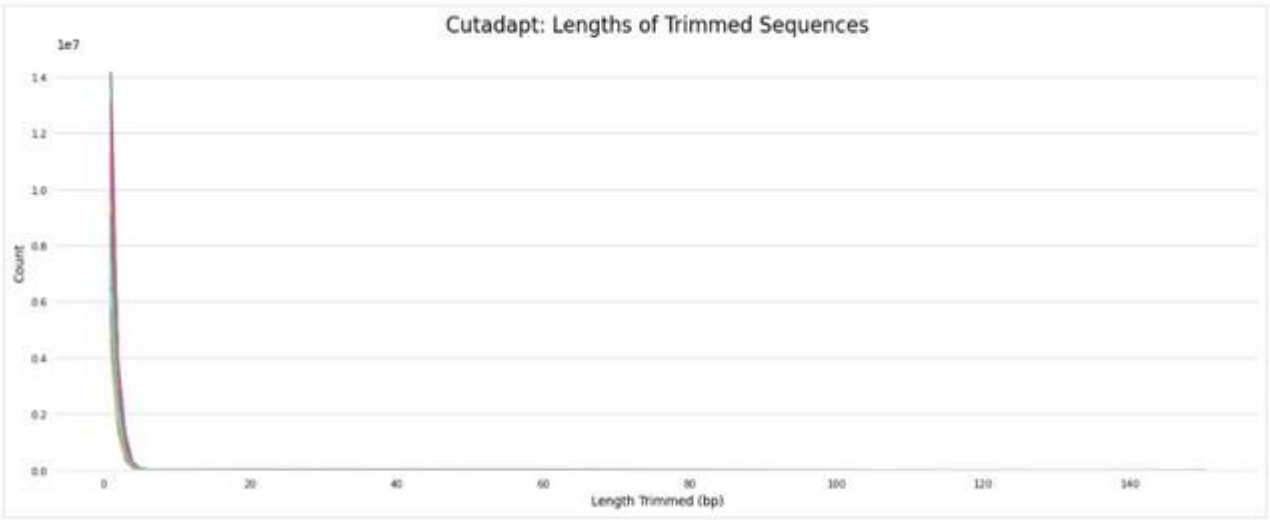
- The distribution of average sequence quality scores:



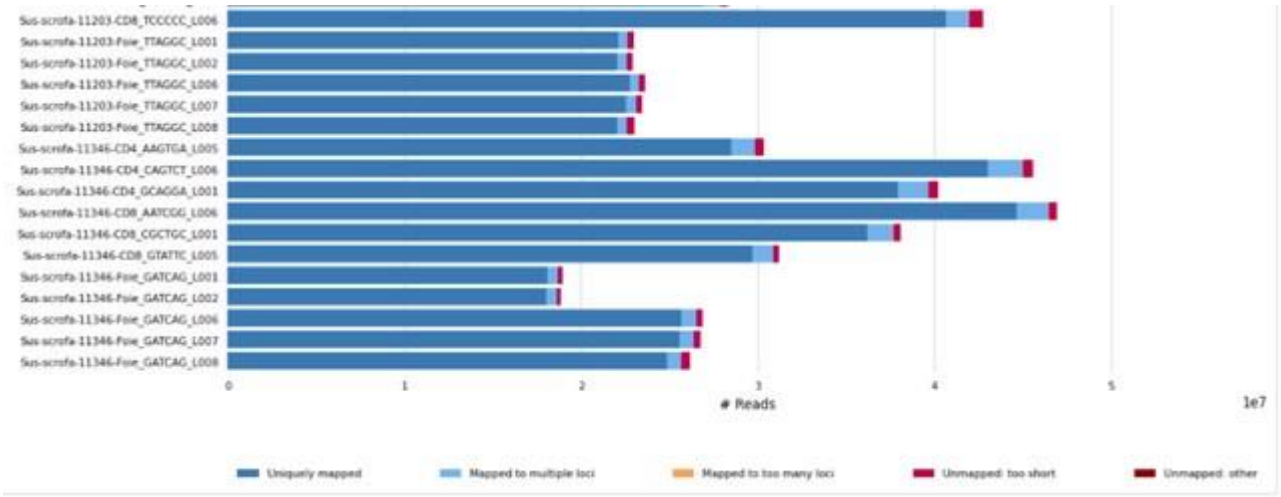
- The percentage of GC bases per sequence:



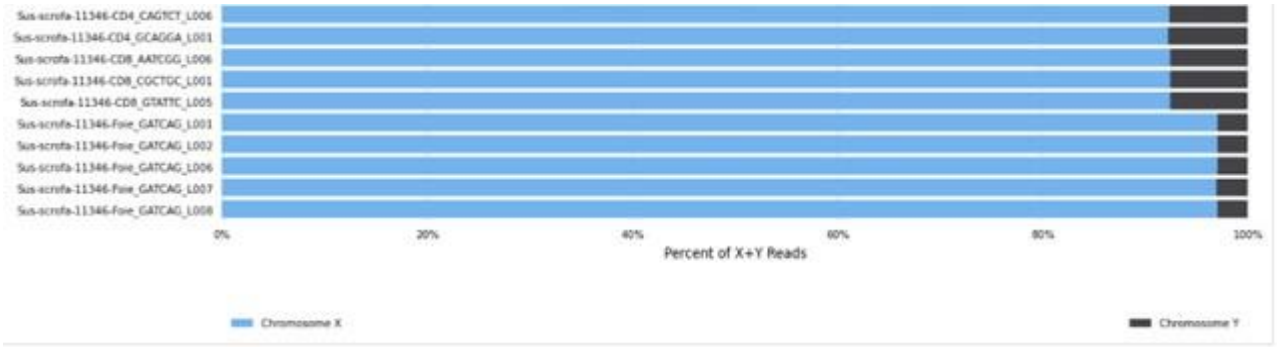
- The length of trimmed sequences (adapters):



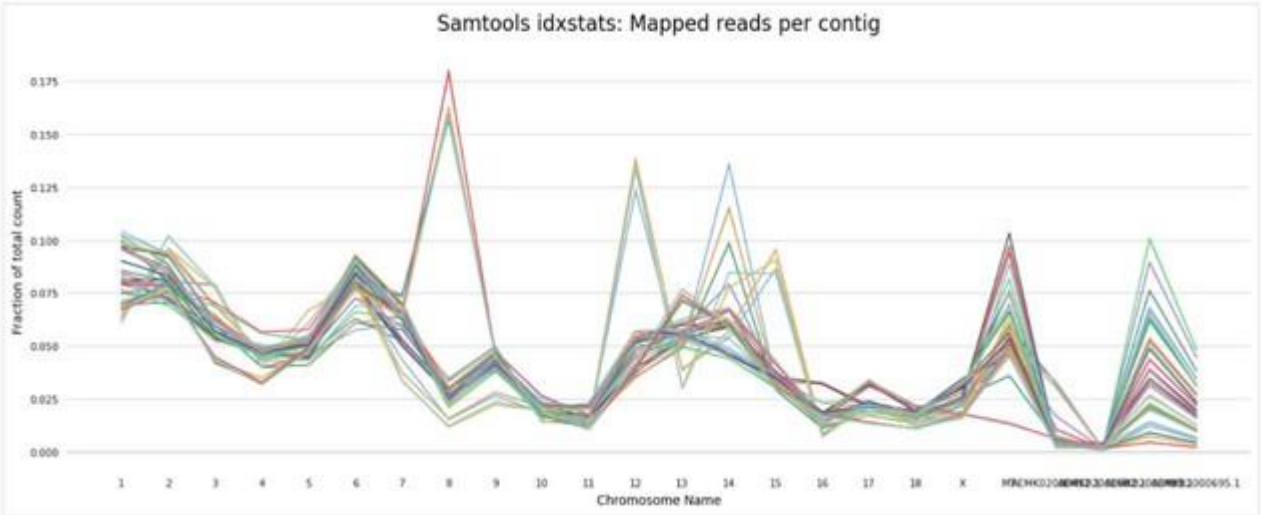
- The proportions of mapped and unmapped reads:



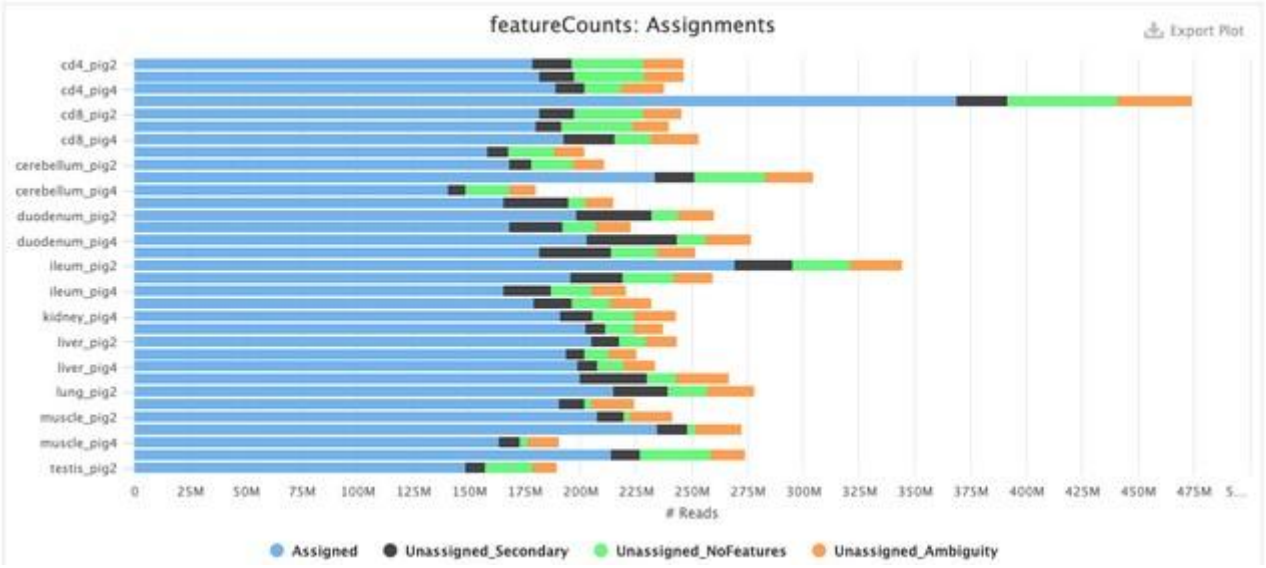
- The proportions of X chromosome and Y chromosome reads:



- The fraction of mapped reads per contig:

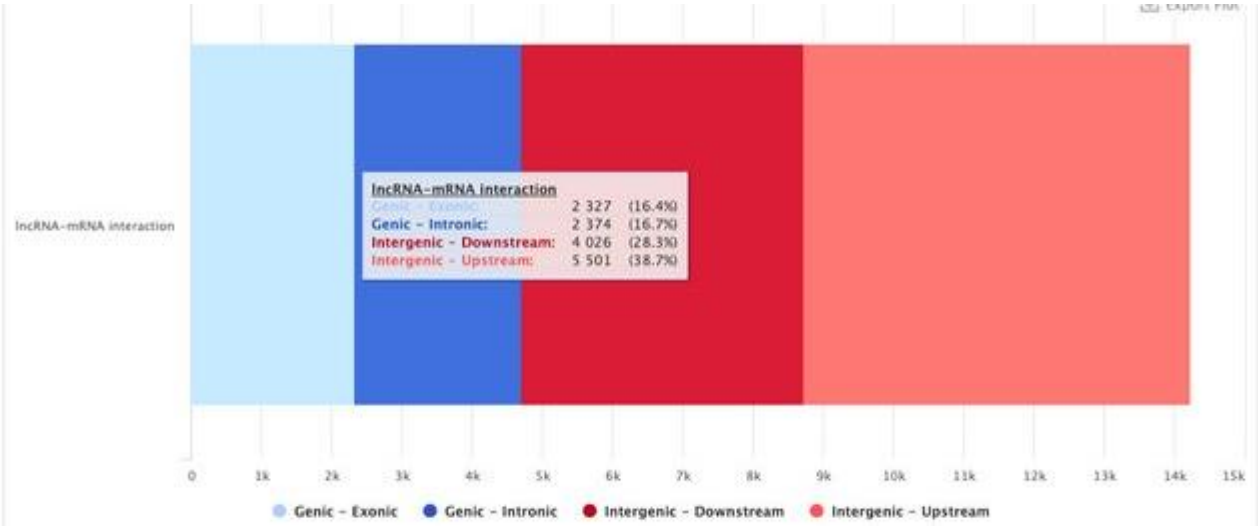


- The number of exonic reads (blue) per experiment group:



- The number of long non-coding transcripts in the novel annotation:

- The positions of long non-coding transcripts relative to the closest coding transcript:



Communications

The TAGADA pipeline has been presented during the BovReg Nextflow workshop organized by the CRG on 17-20 November 2020: https://www.bovreg.eu/nextflow-and-nf-core-workshop-by-crg/ and at the JOBIM Bioinformatics conference (July 2021): https://jobim2021.sciencesconf.org/resource/page/id/11

3.2.2 Iso-seq analysis pipeline

The Proj-gs-iso-seq pipeline was developed using the last nf-core template files. DSL2 version of nextflow language allows the isolatation of each program into modules that can be tested and shared separately. Each module uses a conda package or a container to run the analyses.

The proj-gs-isos-seq pipeline is dedicated to the analysis of long read RNA sequence (or Iso-Seq) data as generated on Pacific Biosciences (PacBio) sequencing platforms. A schematic overview of the pipeline is shown below. The pipeline can be decomposed into three phases:

- Highly accurate transcripts sequence creation
- Transcript mapping on the reference genome
- Post processing of alignments for gene model cleaning (TAMA framework)

The first phase of the program uses the official PacBio isoseq3 tools to process the raw sequencing data:

- CCS: Transcripts consensus sequence creation
- LIMA: Primer removal from consensus
- ISOSEQ REFINE: Trimming of polyA tail from transcripts, chimeric and polyA tailless transcript filtration
- PBMERGE: merging of all transcripts into one sample dataset
- ISOSEQ CLUSTER: Polishing of the consensus to reach highly accurate transcripts
- SAMTOOLS FASTQ: Creation extraction of a FASTQ file describing base level quality

The second phase of the pipeline generates gene models by mapping the transcripts onto the reference genome. First, it applies minimap2 (https://github.com/lh3/minimap2) on the FASTQ files. Then it converts resulting alignments from SAM format to BAM format with SAMTOOLS VIEW and SAMTOOLS SORT commands (https://github.com/samtools/samtools).

Finally, the TAMA framework (https://github.com/GenomeRIK/tama) is used to produce one annotation per sample:

- The alignments are split by chromosomes for parallelised analysis using BAMTOOLS
- Genes models are cleaned with TAMA collapse which fine-tunes merging and keeps track of the contributing components
- The data are merged back by sample using TAMA merge

The pipeline allows rapid and simple generation of whole genome annotations. Moreover, thanks to nextflow language and the mandatory containerisation of programs, no installation is required (expect nextflow and a container framework).

Code repository

The pipeline is available and accessible in the FAANG GitHub at Proj-gs-iso-seq (https://github.com/FAANG/proj-gs-isoseq).

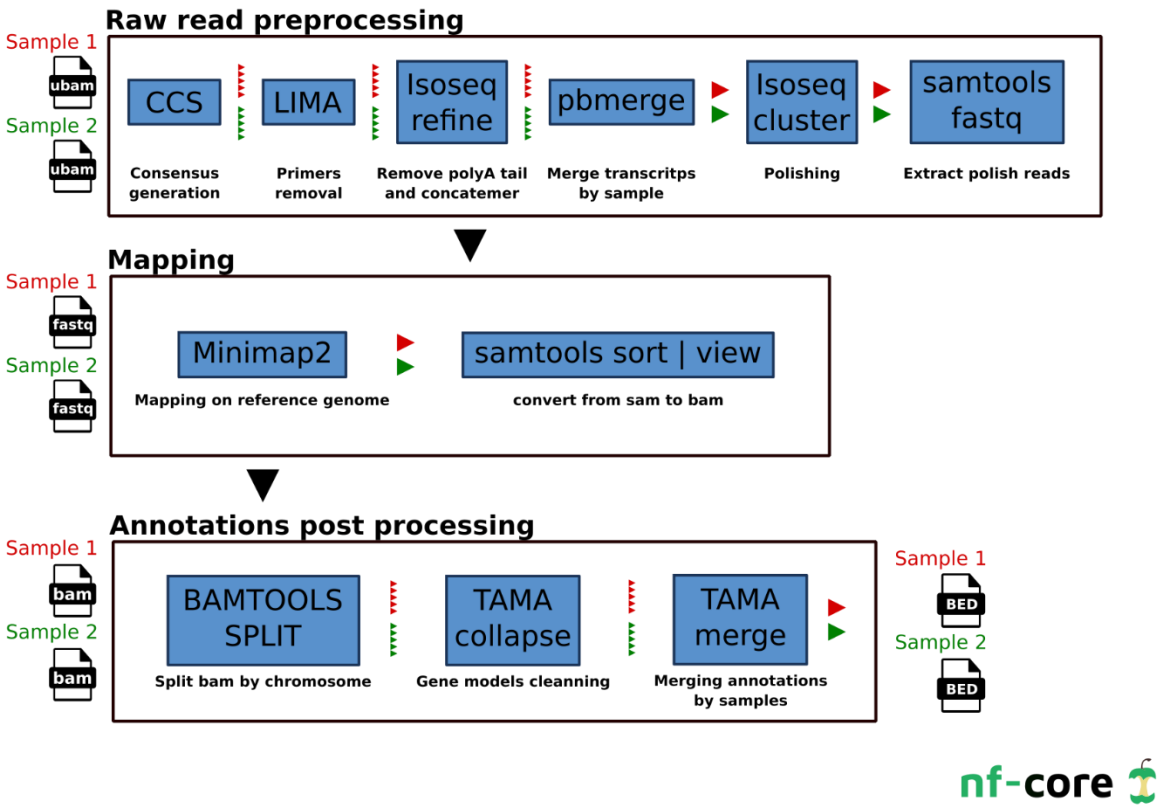


Figure I: Schematic representation of the Iso-Seq analysis pipeline (Proj-gs-iso-seq)

3.2.3 GSM-pipeline: GENE-SWitCH project methylation analysis pipeline

The GENE-SWitCH project methylation analysis pipeline (GSM-pipeline) is based on the pre-existing nf-core pipeline for analysis of methylation data (https://nf-co.re/methylseq) to which the GENE-SWitCH team has added extensions (Figure GSM).

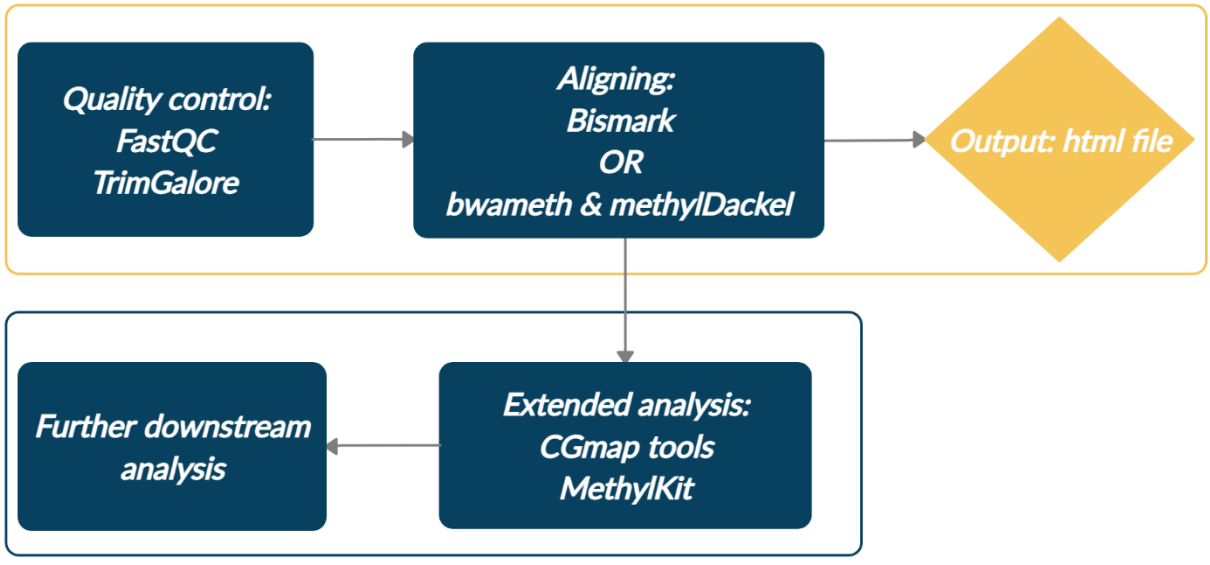


Figure GSM: Schematic of GSM-pipeline. The pre-existing nf-core pipeline is shown above, outlined in yellow and generates an html file as output. The GENE-SWitCH developed outlines are shown, outlined in blue.

The nf-core/methylseq pipeline (https://nf-co.re/methylseq https://github.com/nf-core/methylseq/tree/1.6.1) is an analysis pipeline for Bisulfite-Sequencing data that has the

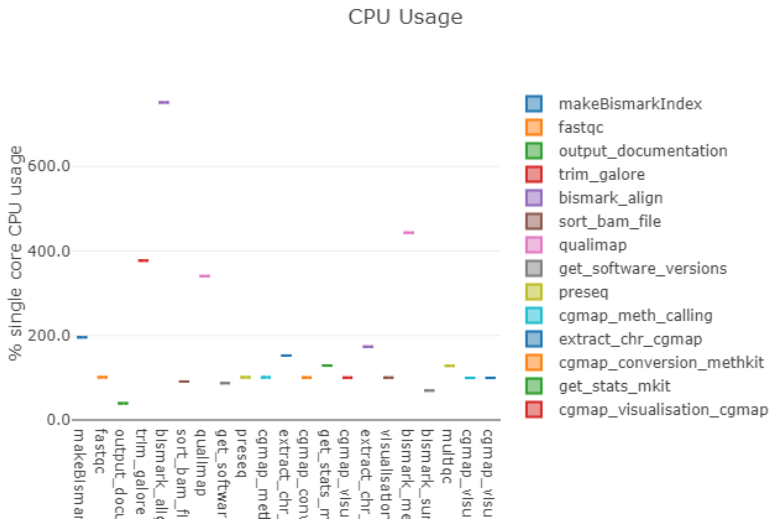
ability to trim raw data (both RRBS and WGBS data) using TrimGalore (http://www.bioinformatics.babraham.ac.uk/ projects/trim_galore/), thereafter the quality of raw and trimmed data is controlled and visualised using FastQC (https://github.com/s-andrews/FastQC). Alignment of the data can then be completed using two different aligners (which can be specified by the user): Bismark (https://github.com/FelixKrueger/Bismark) or bwa-meth (https://github.com/brentp/bwa-meth) with MethylDackel (https://github.com/dpryan79/methyldackel) for methylation calling. Methylation calling is included together with the alignment step in Bismark, and for this reason MethylDackel is necessary when aligning with bwa-meth. Additionally, as a part of the extension of this pre-existing pipeline methylation calling for further downstream analysis is achieved using CGmaptools (https://github.com/guoweilong/cgmaptools). CGmaptools is a bisulfite sequencing analysis toolset with improved methylation calling, visualisation of methylation data, allele specific methylation, and SNV calling. Finally MethylKit (https://github.com/al2na/methylKit) is a R packaged which is specific for RRBS and WGBS methylation data, and provides useful DNA methylation statistics. The GSM-pipeline gives many different figures and tables, such as overall methylation coverage across the genome, methylation levels per chromosome across the genome, diagrams showing distribution of methylation statistics (CpG, and non-CpG methylation). This provides more insight into methylation data, specifically on a chromosome level and methylation. The GSM-pipeline is containerized with all tools used to complete this analysis, ensuring reproducibility of the analysis. General statistics regarding pipeline, quality, alignment and methylation are provided as an output from the pipeline and can easily be shared with WP2 partners. Figures produced as output from the pipeline can be used in publications and provides correct file types for easy additional downstream analysis. Instructions for running the pipeline can be followed from the nf-core methyl-seq homepage, and the newest version of Nextflow needs to be used when running the pipeline. Some example results and steps for running the pipeline are shown below.

Code repository

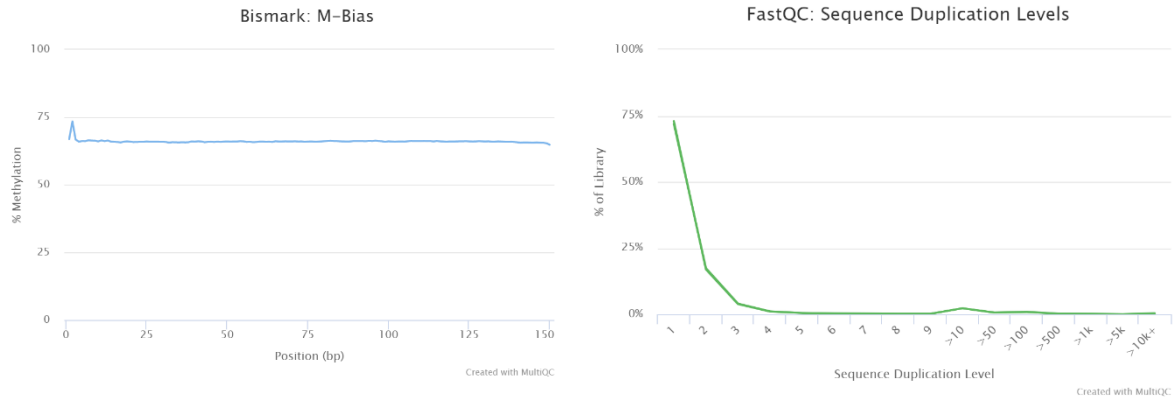
The code was developed using Nextflow workflow and with tools containerized using Docker image (following the coding principles agreed within WP2). Pipeline was developed under Apache 2.0 licence. Pipeline can be downloaded on Github in the following repository: https://github.com/FAANG/GSM-pipeline.

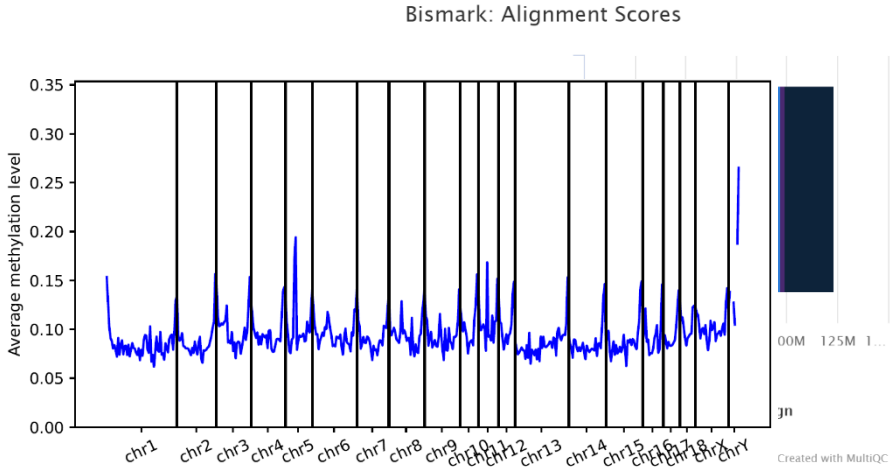
Example results:

Pipeline has been extensively tested on RRBS and WGBS data types as generated within the GENE-SWitCH project WP1. The following figure shows an example html file providing details about the pipeline.

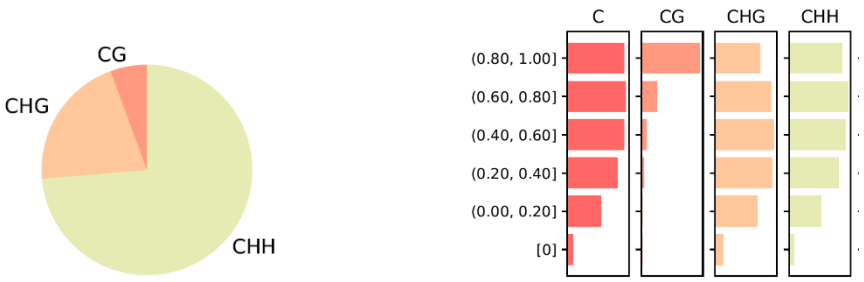


Example of different pipeline information given as output from the pipeline.





HTML file output showing examples of quality control and alignment statistics.



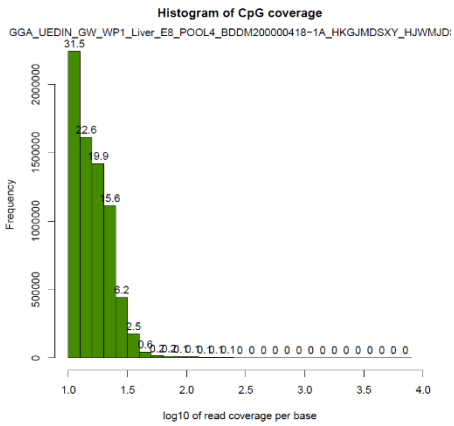
Example results from pipeline for RRBS and WGBS data

Communication

This pipeline has been presented at the BovReg Nextflow workshop in 2020.

3.2.4 Small RNA-seq (smRNA-seq)

For the analysis of small RNA-Seq (smRNA-Seq) data we will use the available nf-core pipeline (https://nf-co.re/smrnaseq). The nf-core/smrnaseq pipeline was built using Nextflow (https://www.nextflow.io/) and uses docker containers in order to facilitate the use of the



pipeline across a range of computing infrastructures.

The nf-core/smrnaseq pipeline as summarised at (https://nf-co.re/smrnaseq) performs the following tasks:

1. Raw read QC with FastQC (https://www.bioinformatics.babraham.ac.uk/projects/fastqc/)
2. Adapter trimming with Trim Galore ! (https://www.bioinformatics.babraham.ac.uk/projects/trim_galore/)
    1. Insert size calculation
    2. Collapse reads with seqcluster (https://seqcluster.readthedocs.io/mirna_annotation.html#processing-of-reads)
3. Alignment against miRBase mature miRNA with Bowtie1 (http://bowtie-bio.sourceforge.net/index.shtml)
4. Alignment against miRBase hairpin
    1. Unaligned reads from step 3 with Bowtie1 (http://bowtie-bio.sourceforge.net/index.shtml)
    2. Collapsed reads from step 2.2 with Bowtie1 (http://bowtie-bio.sourceforge.net/index.shtml)
5. Post-alignment processing of miRBase hairpin
    1. Basic statistics from step 3 and step 4.1 with SAMtools (https://sourceforge.net/projects/samtools/files/samtools/)
    2. Analysis on miRBase hairpin counts with edgeR (https://bioconductor.org/packages/release/bioc/html/edgeR.html) to produce
        - TMM normalization and a table of top expressed hairpin
        - MDS plot clustering samples
        - Heatmap of sample similarities
    3. miRNA and isomiR annotation from step 4.1 with mirtop (https://github.com/miRTop/mirtop)
6. Alignment against host reference genome with Bowtie1 (http://bowtie-bio.sourceforge.net/index.shtml)

    1. Post-alignment processing of alignment against host reference genome with SAMtools (https://sourceforge.net/projects/samtools/files/samtools/)
7. Novel miRNAs and known miRNAs discovery with MiRDeep2 (https://www.mdc-berlin.de/content/mirdeep2-documentation)
    1. Mapping against reference genome with mapper module
    2. Known and novel miRNA discovery with mirdeep2 module
8. miRNA quality control with mirtrace (https://github.com/friedlanderlab/mirtrace)
9. Present QC for raw read, alignment and expression results with multiQC (https://multiqc.info/)

The small RNA-seq analysis pipeline is linked to the FAANg GitHub repository at (https://github.com/FAANG/analysis-smrnaseq).

3.2.5 ATAC-Seq

For the analysis of ATAC-Seq data we will use the available nf-core ATAC-Seq pipeline (https://nf-co.re/atacseq). The nfcore/atacseq pipeline was built using Nextflow (https://www.nextflow.io/) and uses docker containers in order to facilitate the use of the pipeline across a range of computing infrastructures.

The nf-core/atacseq pipeline as summarised at (https://nf-co.re/atacseq) performs the following tasks:

1. Raw read QC with FastQC (https://www.bioinformatics.babraham.ac.uk/projects/fastqc/)
2. Adapter trimming with Trim Galore ! (https://www.bioinformatics.babraham.ac.uk/projects/trim_galore/)
3. Alignment with BWA (https://sourceforge.net/projects/bio-bwa/files/)
4. Mark duplicates with picard (https://broadinstitute.github.io/picard/)
5. Merge alignments from multiple libraries of the same sample with picard (https://broadinstitute.github.io/picard/)
    1. Re-mark duplicates with picard (https://broadinstitute.github.io/picard/)
    2. Filtering to remove:
        - Reads mapping to mitochondrial DNA with SAMtools (https://sourceforge.net/projects/samtools/files/samtools/)
        - Reads mapping to blacklisted regions with SAMtools (https://sourceforge.net/projects/samtools/files/samtools/) and (BEDTools (https://github.com/arq5x/bedtools2/)
        - Reads that are mapped as duplicates with SAMtools (https://sourceforge.net/projects/samtools/files/samtools/)
        - Reads that are not marked as primary alignments with SAMtools (https://sourceforge.net/projects/samtools/files/samtools/)
        - Reads that are unmapped with SAMtools (https://sourceforge.net/projects/samtools/files/samtools/)
        - Reads that map to multiple locations with SAMtools (https://sourceforge.net/projects/samtools/files/samtools/)
        - Reads containing > 4 mismatches with BAMTools (https://github.com/pezmaster31/bamtools)
        - Reads that are soft-clipped with BAMTools (https://github.com/pezmaster31/bamtools)
        - Reads that map have an insert size > 2 kb with BAMTools (https://github.com/pezmaster31/bamtools) (paired-end reads only)
        - Reads that map to different chromosomes with Pysam (https://pysam.readthedocs.io/en/latest/installation.html) (paired-end reads only)

- Reads that are not in FR (forward) orientation with Pysam (https://pysam.readthedocs.io/en/latest/installation.html) (paired-end reads only)
- Reads where only one read of the pair fails the above criteria with Pysam (https://pysam.readthedocs.io/en/latest/installation.html) (paired-end reads only)

3. Alignment-level QC and estimation of library complexity with picard (https://broadinstitute.github.io/picard/) and Preseq (http://smithlabresearch.org/software/preseq/)
4. Create normalised bigwig files scaled to 1 million mapped reads with BEDTools (https://github.com/arq5x/bedtools2/) and bedGraphToBigWig (http://hgdownload.soe.ucsc.edu/admin/exe/)
5. Generate gene-body meta-proflie from bigWigfiles with deepTools (https://deeptools.readthedocs.io/en/develop/content/tools/plotProfile.html)
6. Calculate genome-wide enrichment with deepTools (https://deeptools.readthedocs.io/en/develop/content/tools/plotProfile.html)
7. Call broad/narrow peaks with MACS2 (https://github.com/macs3-project/MACS)
8. Annotate peaks relative to gene features with HOMER (http://homer.ucsd.edu/homer/download.html)
9. Create consensus peakset across all samples and create tabular file to aid in the filtering of the data with BEDTools (https://github.com/arq5x/bedtools2/)
10. Count reads in consensus peaks with featureCounts (http://bioinf.wehi.edu.au/featureCounts/)
11. Differential accessibility analysis, PCA and clustering with R (https://www.r-project.org/) and DESeq2 (https://bioconductor.org/packages/release/bioc/html/DESeq2.html)
12. Generate ATAC-seq specific QC html report with atagv (https://github.com/ParkerLab/ataqv)

6. Merge filtered alignments across replicates with picard (https://broadinstitute.github.io/picard/)
    1. Re-mark duplicates with picard (https://broadinstitute.github.io/picard/)
    2. Remove duplicate reads with SAMtools ()
    3. Create normalised bigwig files scaled to 1 million mapped reads with BEDTools () and bedGraphToBigWig (http://hgdownload.soe.ucsc.edu/admin/exe/)
    4. Call broad/narrow peaks with MACS2 (https://github.com/macs3-project/MACS)
    5. Annotate peaks relative to gene features with HOMER (http://homer.ucsd.edu/homer/download.html)
    6. Create consensus peakset across all samples and create tabular file to aid in the filtering of the data with BEDTools ()
    7. Count reads in consensus peaks relative to merged library-level alignments with featureCounts (http://bioinf.wehi.edu.au/featureCounts/)
    8. Differential accessibility analysis, PCA and clustering with R (https://www.r-project.org/) and DESeq2 (https://bioconductor.org/packages/release/bioc/html/DESeq2.html)
7. Create IGN session file containing bigwig tracks, peaks and differential sites for data visualisation with the Integrative Genome Viewer (IGV) (https://software.broadinstitute.org/software/igv/)

8. Present QC for raw read, alignment, peak-calling and differential accessibility results with ataqv (https://github.com/ParkerLab/ataqv) and MultiQC (https://multiqc.info/)

The ATAC-seq pipeline can be accessed on the nf-core at https://nf-co.re/atacseq.

3.3 Datasets for testing and benchmarking

Shared datasets are invaluable for the development of software analysis tools, especially in multipartner teams. As the GENE-SWitCH project has progressed to data production we now using project data for testing our pipelines. These data are available via the FAANG Data Portal where the data are already in the public domain or via GENE-SWitCH project specific directories in the EMBL-EBI Embassy Cloud infrastructure.

3.4 Shared computing infrastructure

Whilst our objective is to develop bioinformatics pipelines that can be deployed in a consistent manner across any computing infrastructure, the access to a shared computing environment facilitates the work of the WP2 partners. Thus, the software development and testing of the pipelines are being conducted on the EMBL-EBI Embassy cloud infrastructure (https://www.embassycloud.org/) (MS7; D3.4 in progress).

4 Conclusion

The capability to reproduce experiments and the analysis of experimental data is critical to the scientific approach. There are multiple bioinformatics tools and pipelines for the analysis of functional genomic data in the public domain. We have established the principles and means to ensure that the bioinformatics pipelines used for the analyses of GENE-SWitCH data will be available to others in order that they can reproduce our analyses and check the validity of our conclusions. Through coordination with the other H2020-funded FAANG projects (BovReg, Aqua-FAANG) and other major FAANG projects worldwide we will maximise the opportunities for comparative analyses.

5 Deviations or delays

Delays in recruiting bioinformaticians at Partner UEDIN means that the completion of the Iso-Seq bioinformatics pipeline to be used for analyses of the GENE-SWitCH data was delayed. However, we have established the principles for development and systems to share the pipelines widely and the infrastructure through which to deploy the pipelines. The joint FAANG workshop at EMBL-EBI on 25-27 February 2020 has been a good opportunity to ensure that our pipeline developments are in step with those of other H2020 projects (BovReg and AquaFAANG). The pipelines for analysis of RNA-seq, Iso-Seq, smRNA-Seq, methyl-Seq and ATAC-seq data are in place and are being used for analysis of GENE-SWitCH data and the primary annotation of the pig and chicken genomes.

6 References

The ENCODE Project Consortium. An integrated encyclopedia of DNA elements in the human genome. Nature. 2012; 489:57–74.

Giuffra E, Tuggle C, and the FAANG Consortium. FAANG: Current Achievements and Roadmap. Annual Review of Animal Biosciences. 2019, Vol. 7, pp 65-88.

The FAANG Consortium. Coordinated international action to accelerate Genome to Phenome, The Functional Annotation of ANimal Genomes (FAANG) Project. Genome Biology 2015, 16:57.

Harrison P, Fan J, et al. FAANG, establishing metadata standards, validation and best practices for the farmed and companion animal community. Anim Genet. 2018; 49:520–6.

Foissac S, Djebali S, et al. Multi-species annotation of transcriptome and chromatin structure in domesticated animals. BMC Biology. 2019, Vol 17(1): 108. doi:10.1186/s12915-019-0726-5.

Barrett T, Clark K, et al. BioProject and BioSample databases at NCBI: facilitating capture and organization of metadata. Nucleic Acids Res. 2012; 40:D57–63. https://doi.org/10.1093/nar/gkr1163.

Ewels PA, Peltzer A, Fillinger S, Patel H, Alneberg J, Wilm A, Garcia MU, Di Tommaso P, Nahnsen S. The nf-core framework for community-curated bioinformatics pipelines. Nat Biotechnol. 2020 Mar;38(3):276-278. doi: 10.1038/s41587-020-0439-x.

Danecek P, Bonfield JK, Liddle J, Marshall J, Ohan V, Pollard MO, Whitwham A, Keane T, McCarthy SA, Davies RM, Li H. Twelve years of SAMtools and BCFtools. Gigascience. 2021 Feb 16;10(2):giab008. doi: 10.1093/gigascience/giab008.

7 Annexes

7.1 Annex 1: GENE-SWitCH principles for software development

GENE-SWitCH: Pipeline development guidelines, standards and coding principles

Authors: Alan Archibald, Sarah Djebali, Sylvain Foissac, Peter Harrison, Andy Law, Alexey Sokolov, Mick Watson.

This document details the guidelines, standards and coding principles that must be adhered to for the development of bioinformatic pipelines for the GENE-SWitCH project.

1. All pipelines will be openly developed with code freely available in public GitHub (https://github.com) repositories.
2. All pipelines will be licensed under Apache 2.0 (see (https://github.com/FAANG/dcc-portal-frontend/blob/master/LICENSE)). A license file must be available in the root directory of the repository in a file named 'LICENSE'.
3. All pipelines will be stored in the FAANG public GitHub repository (https://github.com/FAANG). Each pipeline must follow the naming convention of hyphens '-' for spaces, all lowercase and have the prefix 'proj-gs-'. New pipelines should match the style of existing pipelines in the project.
4. Wherever possible pipelines should utilise open source software dependencies. Where proprietary software has to be included in a pipeline this must be clearly stated in the pipelines readme file, with instructions on how a user can obtain the software and license.
5. You must only include code inside your repository that has been licensed for this usage, otherwise it must be linked to externally.
6. All repositories should have a readme file in the root directory. That describes the pipelines intended use, installation instructions and usage instructions.
7. The readme file must contain the following statements 'The GENE-SWitCH project has received funding from the European Union's Horizon 2020 (https://ec.europa.eu/programmes/horizon2020/) research and innovation program under Grant Agreement No 817998.' and 'This repository reflects only the listed contributors views. Neither the European Commission nor its Agency REA are responsible for any use that may be made of the information it contains.'.
8. Workflows should utilise the Nextflow workflow manager (https://www.nextflow.io/).
9. Workflows should be containerised in Docker to handle all dependencies (https://www.docker.com/).
10. Prefered languages for pipeline development are Python 3 and Java 8.
11. Python 3 development should follow PEP8 style guide.
12. Where options are required, provide default options in YAML configuration files, and instructions in the readme for all potential configuration options.
13. Comply with latest security recommendations, patching vulnerable dependencies.
14. Pipelines should have appropriate testing from defined test datasets, preferably from Chicken and Pig samples. Test data must be held within the repository, be publicly available in the archives, and referenced in the root readme.

15. Pipelines should include benchmarking, particularly if multiple options exist for certain stages of the pipeline, for example different alignment options. Benchmarking datasets should be available in the public archives and described in the root readme file.