GigaScience, 8, 2019, 1–11

doi: 10.1093/gigascience/giz037
Technical Note

TECHNICAL NOTE

GenPipes: an open-source framework for distributed and scalable genomic analyses

Mathieu Bourgey[1,2,*,†], Rola Dali[1,2,†], Robert Eveleigh[1,2], Kuang Chung Chen[3,4], Louis Letourneau[1,2], Joel Fillon[5], Marc Michaud[2], Maxime Caron[1,2,5], Johanna Sandoval[6], Francois Lefebvre[1,2], Gary Leveque[1,2], Eloi Mercier[1,2], David Bujold[1,2], Pascale Marquis[1,2], Patrick Tran Van[7], David Anderson de Lima Morais[8], Julien Tremblay[9], Xiaojian Shao[1,2], Edouard Henrion[1,2], Emmanuel Gonzalez[1,2], Pierre-Olivier Quirion[1,2], Bryan Caron[3,4] and Guillaume Bourque[1,2,5,*]

[1]Canadian Centre for Computational Genomics, Montréal, QC, Canada; [2]McGill University and Genome Québec Innovation Center, Montréal, QC, Canada; [3]McGill HPC Centre, McGill University, Montréal, QC, Canada; [4]Calcul Québec, QC, Canada; [5]Department of Human Genetics, McGill University, Montréal, QC, Canada; [6]Beaulieu-Saucier Université de Montréal Pharmacogenomics Centre, Montréal, QC, Canada; [7]Department of Ecology and Evolution, University of Lausanne, Lausanne, Switzerland; [8]Centre de Calcul Scientifique (CCS), Université de Sherbrooke, Sherbrooke, QC, Canada and [9]Energy, Mining and Environment, National Research Council Canada, Montréal, QC, Canada

[*]**Correspondence address.** Guillaume Bourque, Montréal, QC,740 Dr Penfield Ave, Montreal, QC. H3A 0G1, Canada. Tel: +1(514) 398–7245; Fax: +1(514) 398–1790; E-mail: guil.bourque@mcgill.ca http://orcid.org/0000-0002-8432-834X; Mathieu Bourgey, Montréal, QC,740 Dr Penfield Ave, Montreal, QC. H3A 0G1, Canada. E-mail: mathieu.bourgey@mcgill.ca http://orcid.org/0000-0002-3933-9656

[†]Co–first authors.

Abstract

Background: With the decreasing cost of sequencing and the rapid developments in genomics technologies and protocols, the need for validated bioinformatics software that enables efficient large-scale data processing is growing. **Findings:** Here we present GenPipes, a flexible Python-based framework that facilitates the development and deployment of multi-step workflows optimized for high-performance computing clusters and the cloud. GenPipes already implements 12 validated and scalable pipelines for various genomics applications, including RNA sequencing, chromatin immunoprecipitation sequencing, DNA sequencing, methylation sequencing, Hi-C, capture Hi-C, metagenomics, and Pacific Biosciences long-read assembly. The software is available under a GPLv3 open source license and is continuously updated to follow recent advances in genomics and bioinformatics. The framework has already been configured on several servers, and a Docker image is also available to facilitate additional installations. **Conclusions:** GenPipes offers genomics researchers a simple method to analyze different types of data, customizable to their needs and resources, as well as the flexibility to create their own workflows.

Received: 2 June 2018; **Revised:** 28 September 2018; **Accepted:** 10 March 2019

Keywords: genomics; workflow management systems; frameworks; workflow; pipeline; bioinformatics

Introduction

Sequencing has become an indispensable tool in our quest to understand biological processes. Moreover, facilitated by a significant decline in overall costs, new technologies and experimental protocols are being developed at a fast pace. This has resulted in massive amounts of sequencing data being produced and deposited in various public archives. For instance, a number of national initiatives, such as Genomics England and All of US, plan to sequence hundreds of thousands of individual genomes in an effort to further develop precision medicine. Similarly, a number of large initiatives, such as ENCODE [1] and the International Human Epigenome Consortium (IHEC) [2], plan to generate thousands of epigenomics datasets to better understand gene regulation in normal and disease processes. Despite this rapid progress in sequencing, genomics technologies, and available datasets, processing and analyses have struggled to keep up. Indeed, the need for robust, open source, and scalable bioinformatics pipelines has become a major bottleneck for genomics [3].

Available bioinformatics tools for genomic data can be categorized into 3 different groups: (i) analysis platforms/workbenches, (ii) workflow management systems (WMS)/frameworks, and (iii) individual analysis pipelines/workflows. Platforms of the first type, such as Galaxy [4] or DNA Nexus [5], provide a full workbench for data upload and storage and are accompanied by a set of available tools. While they provide fast and easy user services, such tools can be inconvenient for large-scale projects owing to the need to move sizeable datasets to the platform. In the second type, WMSs such as Snakemake [6], Nextflow [7], BPipe [8], and BigDataScript [9] and declarative workflow description languages such as CWL or WDL are dedicated to providing a customizable framework to build bioinformatics pipelines. Such solutions are flexible and can help in pipeline implementation but rarely provide robust pre-built pipelines that are ready for production analysis. Finally, tools of the third type are individual analysis pipelines for various applications that have been validated and published. These are useful for specific applications but can sometimes be challenging to implement and difficult to modify or scale up. They have also rarely been tested on multiple computing infrastructures.

Here we present GenPipes, an open source, Python-based WMS for pipeline development. As part of its implementation, GenPipes includes a set of high-quality, standardized analysis pipelines, designed for high-performance computing (HPC) resources and cloud environments. GenPipes' WMS and pipelines have been tested, benchmarked, and used extensively over the past 4 years. GenPipes is continuously updated and is configured on several different HPC clusters with different properties. By combining both WMS and extensively validated end-to-end analysis workflows, GenPipes offers turnkey analyses for a wide range of bioinformatics applications in the genomics field while also enabling flexible and robust extensions.

Material and Methods

Overview of the GenPipes framework

GenPipes is an object-oriented framework consisting of Python scripts and libraries that create a list of jobs to be launched as Bash commands (Fig. 1). There are 4 main objects that manage the different components of the analysis workflow, namely, Pipeline, Step, Job, and Scheduler. The main object is the "Pipeline" object, which controls the workflow of the analysis. Each specific analysis workflow is thus defined as a specific Pipeline object. Pipeline objects can inherit from one another. The Pipeline object defines the flow of the analysis by calling specific "Step" objects. The Pipeline instance could call all steps implemented in a pipeline or only a set of steps selected by the user. Each step of a pipeline is a unit block that encapsulates a part of the analysis (e.g., trimming or alignment). The Step object is a central unit object that corresponds to a specific analysis task. The execution of the task is directly managed by the code defined in each Step instance; some steps may execute their task on each sample individually while other steps execute their task using all the samples collectively. The main purpose of the Step object is to generate a list of "Job" objects, which correspond to the consecutive execution of single tasks. The Job object defines the commands that will be submitted to the system. It contains all the elements needed to execute the commands, such as input files, modules to be loaded, as well as job dependencies and temporary files. Each Job object will be submitted to the system using a specific "Scheduler" object. The Scheduler object creates execution commands that are compatible with the user's computing system. Four different Scheduler objects have already been implemented (PBS, SLURM, Batch, and Daemon; see below).

GenPipes' object-oriented framework simplifies the development of new features and its adaptation to new systems; new workflows can be created by implementing a Pipeline object that inherits features and steps from other existing Pipeline objects. Similarly, deploying GenPipes on a new system may only require the development of the corresponding Scheduler object along with specific configuration files. GenPipes' command execution details have been implemented using a shared library system, which allows the modification of tasks by simply adjusting input parameters. This simplifies code maintenance and makes changes in software versions consistent across all pipelines.

Freely distributed and pre-installed on a number of HPC resources

GenPipes is an open source framework freely distributed and open for external contributions from the developer community. GenPipes can be installed from scratch on any Linux cluster supporting Python 2.7 by following the available instructions [10]. GenPipes can also be used via a Docker image, which simplifies the set-up process and can be used on a range of platforms, including cloud platforms. This allows system-wide installations, as well as local user installations via the Docker image without needing special permissions.

Through a partnership with the Compute Canada consortium [11], the pipelines and third-party tools have also been configured on 6 different Compute Canada HPC centers. This allows any Canadian researcher to use GenPipes along with the needed computing resources by simply applying to the consortium [12]. To ensure consistency of pipeline versions and used dependencies (such as genome references and annotation files) and to avoid discrepancy between compute sites, pipeline set-up has been centralized to 1 location, which is then distributed on a

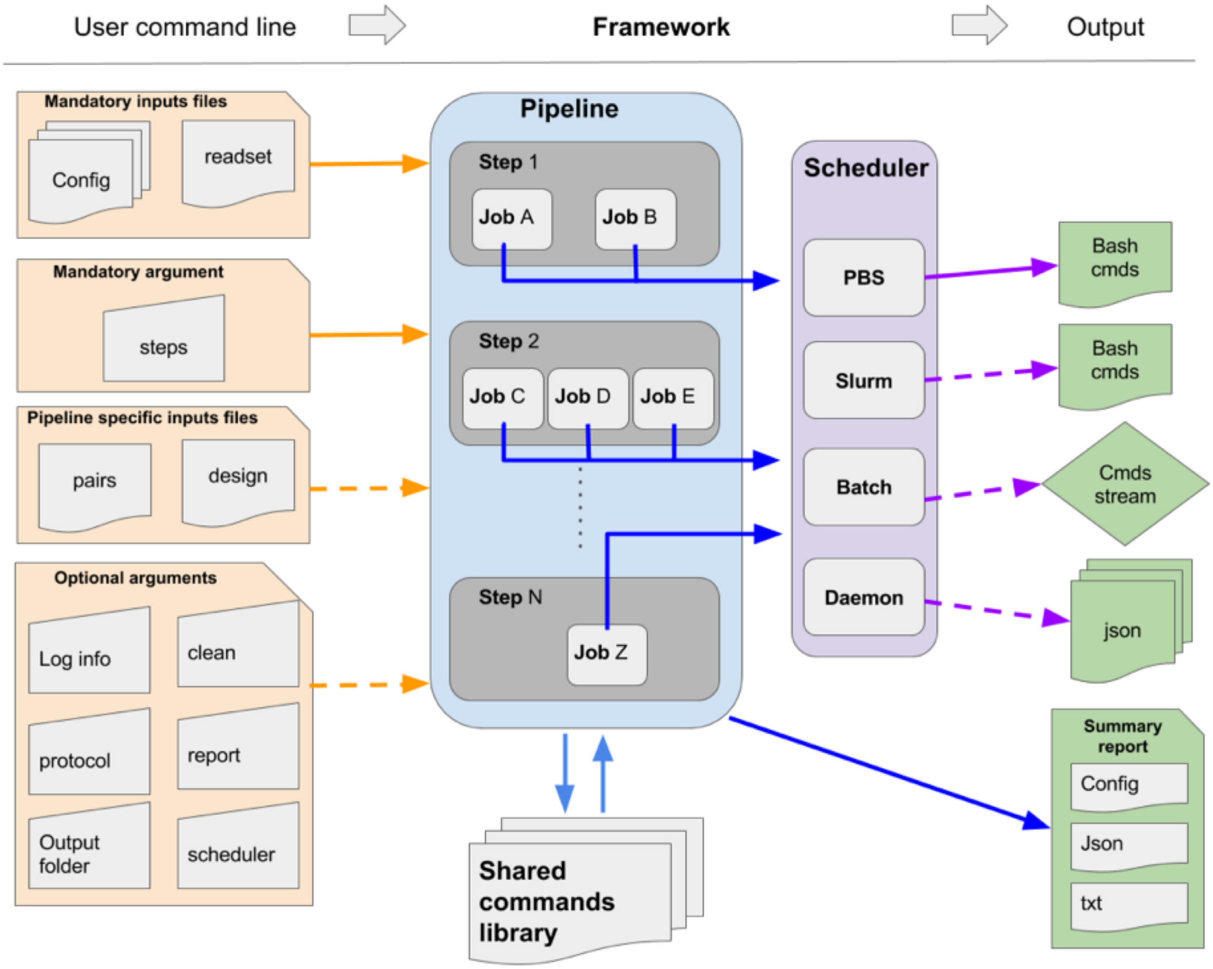

Figure 1: General workflow of GenPipes. Diagram showing how the information flows from the user command line input through the 4 different objects (Pipeline, Step, Job, and Scheduler) in order to generate system-specific executable outputs. cmds: commands.

real-time shared file system: the CERN (European Organization for Nuclear Research) Virtual Machine File System [13].

Running GenPipes

GenPipes is a command line tool. Its use has been simplified to accommodate general users. A full tutorial is available [14]. Briefly, to launch GenPipes, the following is needed:

- A readset file that contains information about the samples, indicated using the flag "-r". GenPipes can aggregate and merge samples as indicated by the readset file.
- Configuration/ini files that contain parameters related to the cluster and the third-party tools, indicated using the flag "-c". Configuration files are customizable, allowing users to adjust different parameters.
- The specific steps to be executed, indicated by the flag "-s".

The generic command to run GenPipes is:

<pipeline>.py -c myConfigurationFile -r myReadSetFile -s 1-X > Commands.txt && bash Commands.txt

where <pipeline> can be any of the 12 available pipelines and X is the step number desired. Commands.txt contains the commands that the system will execute.

Pipelines that conduct sample comparisons, such as ChIP-Seq and RNA sequencing (RNA-Seq), require a design file that describes each contrast. Custom sample groupings can be defined in the design file. Design files are indicated by the flag "-d". The tumour_pair pipeline requires normal–tumour pairing information provided in a standard comma-separated values file using the "-p" option. More information on the design file and the content of each file type can be found in the GenPipes tutorial and the online documentation.

When the GenPipes command is launched, required modules and files will be searched for and validated. If all required modules and files are found, the analysis commands will be produced. GenPipes will create a directed acyclic graph that defines job dependency based on input and output of each step. For a representation of the directed acyclic graph of each pipeline, refer to supplementary Figs S1–14. Once launched, the jobs are sent to the scheduler and queued. As jobs complete successfully, their dependent jobs are released by the scheduler to run. If a job fails, all its dependent jobs are terminated and an email notification is sent to the user. When GenPipes is rerun, it will detect which steps have successfully completed, as described in section "Smart relaunch features," and skip them but will create the command script for the jobs that were not completed successfully. To force the entire command generation, despite successful completion, the "-f" option should be added.

Results

GenPipes was first released in 2014. Since then, it has grown to implement 12 pipelines and is currently installed and maintained on 13 different clusters (Fig. 2a and b). GenPipes has been actively used for the past 4 years to quality control (QC) and analyze thousands of samples each year (Fig. 2c). It has also been used to analyze data for several large-scale projects such as IHEC [2] and eFORGE [15].

Key features of GenPipes

GenPipes' framework has been optimized to facilitate large-scale data analysis. Several features make this possible (Fig. 2a):

Multiple schedulers

GenPipes is optimized for HPC processing. It can currently accommodate 4 different types of schedulers:

- PBSScheduler creates a batch script that is compatible with a PBS (TORQUE) system.
- SLURMscheduler creates a batch script that is compatible with a SLURM system.
- BatchScheduler creates a batch script that contains all the instructions to run all the jobs one after the other.
- DaemonScheduler creates a log of the pipeline command in a JSON file.

Job dependencies

To minimize the overall analysis time, GenPipes uses a dependency model based on input files, which is managed at the Job object level. A job does not need to wait for the completion of a previous step unless it is dependent on its output. Jobs thus become active and can be executed as soon as all their dependencies are met, regardless of the status of previous jobs or of other samples. Thus, when a pipeline is run on multiple samples, it creates several dependency paths, 1 per sample, each of which completes at its own pace.

Smart relaunch features

Large-scale data analysis is subject to failure, which could result from system failure (e.g., power outage, system reboot) or user failure (errors in set parameters, or resources). To limit the micro-management and time required to relaunch the pipeline from scratch, GenPipes includes a system of reporting that provides the status of every job in the analysis in order to facilitate the detection of jobs that have failed. Additionally, a relaunch system is implemented that allows restarting the analysis at the exact state before the failure. The relaunch system uses 2 features: md5sum hash and time stamps. When GenPipes is launched, a md5sum hash is produced for each command. Upon relaunch following a failure, the newly produced hash is compared to that of the completed job to detect changes in the commands. If the hashes are different, the job is relaunched. To detect updates in input files, GenPipes compares the time stamp on the input and output files of already completed jobs. If the date stamp on the input files is more recent than that on the output files, then the job is relaunched. If neither the hash code nor the time stamp flags the job to be relaunched, then it is considered complete and up-to-date and it will be skipped in the pipeline restart process.

Configuration files

Running large-scale analyses requires a very large number of parameters to be set. GenPipes implements a superposed configuration system to reduce the time required to set up or modify parameters needed during the analysis. Configuration files, also referred to as “ini” files, are provided among the arguments of the GenPipes command. These files follow the standard INI format, which was selected for its readability and ease of use by non-expert users. Each pipeline reads all configuration files, one after the other, based on a user-defined order. The order is of major importance because the system will overwrite a parameter each time it is specified in a new ini file. The system allows the use of the default configuration files provided in GenPipes alone or in combination with user-specific configuration files. The configuration files provided with GenPipes are the result of years of experience along with intensive benchmarking. Additionally, several configuration files adjusted for different compute systems or different model organisms are available. The main advantage of this system is to reduce the users' task; only parameters that need to be modified (e.g., system parameters, genomic resources, user-specific parameters) have to be adjusted during the set-up phase of the analysis. To track and enable reproducibility, GenPipes always outputs a file containing the final list of parameters used for the analysis.

Choice among multiple inputs

GenPipes represents a series of Step objects that are interdependent based on inputs and outputs. Many of the pipeline steps implemented in GenPipes represent filtering, manipulation, or modification of specific genomics files that share common formats (e.g., bam, fastq, vcf). To ensure more flexibility in the analysis, a system of ordered list to be interpreted as input files is used. For a given Step, each Job can be given a series of inputs. The Job will browse its list of possible inputs and will consider them based on the order in the list. The first input file found either on disk or in the overall output list will be chosen as input. The chosen input will determine the dependency of the Job to the other Jobs in the pipeline. This system is flexible and allows users to skip specific steps in the pipeline if they consider them unnecessary.

Customizable workflows

Despite the benchmarking and testing made on the standard analysis procedures implemented in GenPipes, some users may be interested in modifying pipelines. To make GenPipes more flexible, a ”protocol” system is used. The system allows the implementation of different workflows into a single Pipeline object. As a result, one can replace specific steps by other user-specific ones. In that case, the user will only need to implement these new Steps and define an additional protocol that will use part of the initial Steps and the newly developed ones. As an example, this has been used to incorporate the Hi-C analysis workflow and the capture Hi-C analysis workflow into GenPipes' hicseq pipeline. A flag (-t hic or -t capture) can be used to specify the workflow to be executed. This system has been developed to reduce the amount of work for external users who decide to contribute to code development and to limit the number of Pipeline objects to maintain. This will also allow us to provide multiple workflows per pipeline to appeal to different tool preferences in each field.

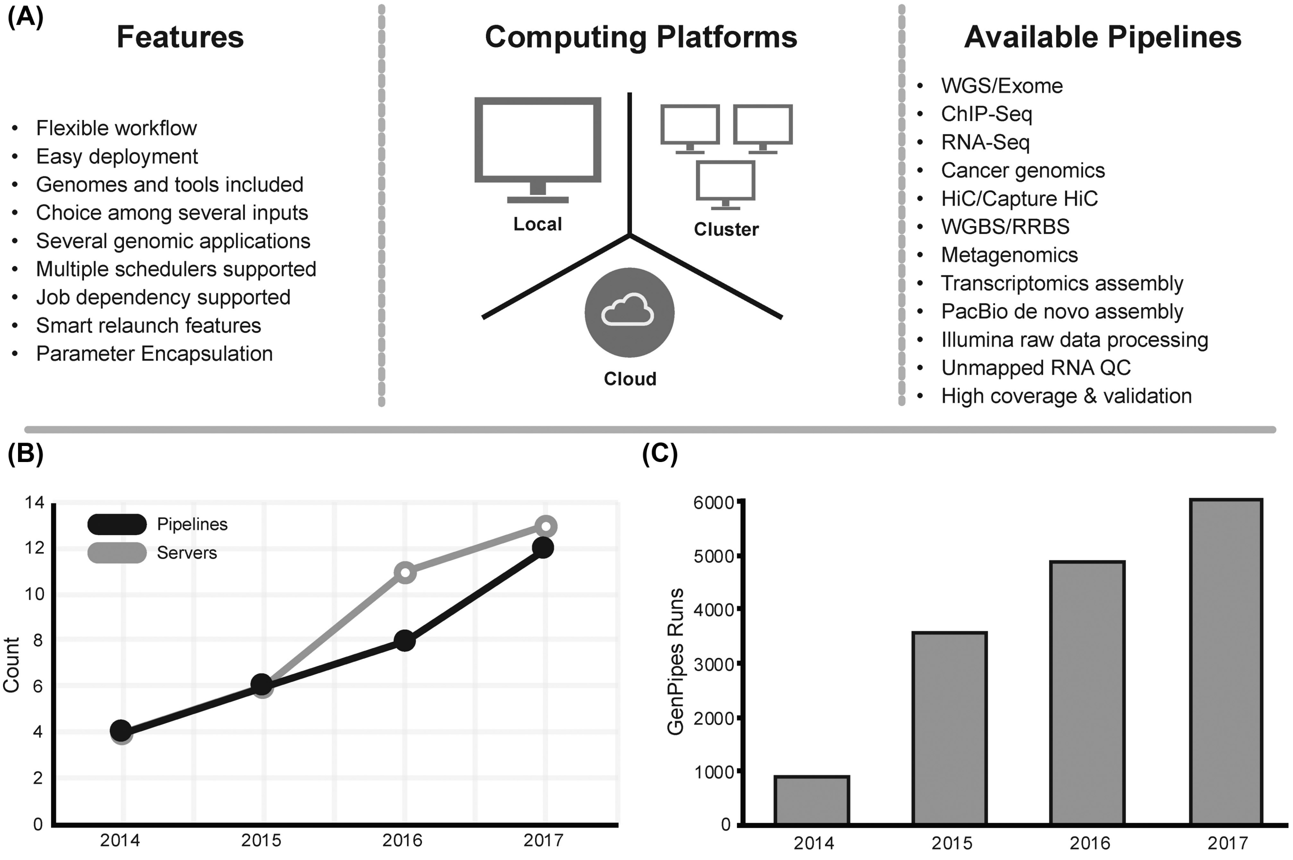

Figure 2: GenPipes' properties and growth. A, Diagram showing GenPipes' features, compatible computing platforms, and available pipelines. B, GenPipes' available pipelines and maintained servers since the release of GenPipes in 2014. C, Bar plot showing the number of GenPipes runs per year since its release. RRBS: reduced-representation bisulfite sequencing; WGS: whole-genome seqencing.

Facilitating dependency installation

Genomic analyses require third-party tools, as well as genome sequence files, annotation files, and indices. GenPipes comes configured with a large set of reference genomes and their respective annotation files, as well as indices for most aligners. It also includes a large set of third-party tools. If GenPipes is being installed from scratch on new clusters, automatic bash scripts that download all tools and genomes are included to ease the set-up process. These scripts support local installations without the need for super-user privileges. Tools and dependencies are versioned and are loaded by GenPipes in a version-specific manner. This allows different pipelines to use different software versions based on need. It also allows retention of the same parameters and tools for any given project for reproducibility. GenPipes is also provided as a container version for which no dependency installation is required.

Available workflows

GenPipes implements 12 standardized genomics workflows including DNA-Seq, tumour analysis, RNA-Seq, *de novo* RNA-Seq, ChIP-Seq, Pacific Biosciences (PacBio) assembly, methylation sequencing, Hi-C, capture Hi-C, and metagenomics (Fig. 2a). All pipelines have been implemented following a robust design and development routine by following established best practices standard operating protocols. Below we summarize GenPipes' workflows; more details are available in the GenPipes documentation. For more details concerning computational resources used by each pipeline, refer to supplementary Table S1. All workflows accept a bam or a fastq file as input.

DNA-Seq pipeline

DNA-Seq has been implemented optimizing the GATK best practices standard operating protocols [16]. This procedure entails trimming raw reads derived from whole-genome or exome data followed by alignment to a known reference, post-alignment refinements, and variant calling. Trimmed reads are aligned to a reference by the Burrows-Wheeler Aligner, bwa-mem [17]. Refinements of mismatches near insertions and deletions (indels) and base qualities are performed using GATK indels realignment and base recalibration [16] to improve read quality after alignment. Processed reads are marked as fragment duplicates using Picard MarkDuplicates [16] and single-nucleotide polymorphisms and small indels are identified using either GATK haplotype callers or SAMtools mpileup [18]. The Genome in a Bottle [19] dataset was used to select steps and parameters minimizing the false-positive rate and maximizing the true-positive variants to achieve a sensitivity of 99.7%, precision of 99.1%, and F1 score of 99.4% (for more details, refer to Supplementary Materials). Finally, additional annotations are incorporated using dbNSFP [20] and/or Gemini [21] and QC metrics are collected at various stages and visualized using MulitQC [22]. This pipeline has 2 different protocols, the default protocol based on the GATK variant caller, haplotype_caller, ("-t mugqic"; Fig. 3) and one based on the mpileup/bcftools caller ("-t mpileup"; Fig. S1).

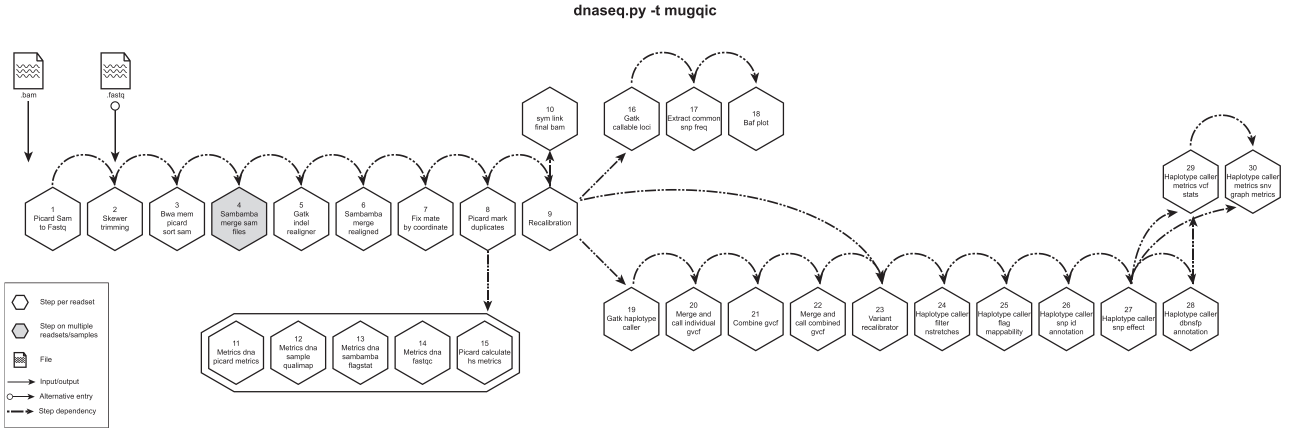
dnaseq.py -t mugqic
.bam
.fastq
1 Picard Sam to Fastq
2 Skewer trimming
3 Bwa mem picard sort sam
4 Sambamba merge sam files
5 Gatk indel realigner
6 Sambamba merge realigned
7 Fix mate by coordinate
8 Picard mark duplicates
9 Recalibration
10 sym link final bam
11 Metrics dna picard metrics
12 Metrics dna sample qualimap
13 Metrics dna sambamba flagstat
14 Metrics dna fastqc
15 Picard calculate hs metrics
16 Gatk callable loci
17 Extract common snp freq
18 Baf plot
19 Gatk haplotype caller
20 Merge and call individual gvcf
21 Combine gvcf
22 Merge and call combined gvcf
23 Variant recalibrator
24 Haplotype caller filter nstretches
25 Haplotype caller flag mappability
26 Haplotype caller snp id annotation
27 Haplotype caller snp effect
28 Haplotype caller dbnsfp annotation
29 Haplotype caller metrics vcf stats
30 Haplotype caller metrics snv graph metrics
Step per readset
Step on multiple readsets/samples
File
Input/output
Alternative entry
Step dependency

Figure 3: GenPipes DNASeq pipeline diagram. Schematic representation of GenPipes' dnaseq.py pipeline. Hexagons represent steps in the pipeline. White hexagons represent steps that process input from a single sample, while grey ones represent steps that process input from several samples. Arrows show step dependencies.

Another pipeline that is optimized for deep coverage samples, dnaseq_high_coverage, can be found in Fig. S2.

RNA-Seq pipeline

This pipeline aligns reads with STAR [23] 2-passes mode, assembles transcripts with Cufflinks [24], and performs differential expression with Cuffdiff [25]. In parallel, gene-level expression is quantified using htseq-count [26], which produces raw read counts that are subsequently used for differential gene expression with both DESeq [27] and edgeR [28]. Several common quality metrics (e.g., ribosomal RNA content, expression saturation estimation) are also calculated through the use of RNA-SeQC [29] and in-house scripts. Gene Ontology terms are also tested for over-representation using GOseq [30]. Expressed short single-nucleotide variants (SNVs) and indels calling is also performed by this pipeline, which optimizes GATK best practices to reach a sensitivity of 92.8%, precision of 87.7%, and F1 score of 90.1%. A schema of pipeline steps can be found in Fig. S3. Another pipeline, rnaseq_light, based on Kallisto [31] and used for quick QC, can be found in Fig. S4.

De-Novo RNASeq pipeline

This pipeline is adapted from the Trinity-Trinotate suggested workflow [32, 33]. It reconstructs transcripts from short reads, predicts proteins, and annotates, leveraging several databases. Quantification is computed using RSEM, and differential expression is tested in a manner identical to the RNA-seq pipeline. We observed that the default parameters of the Trinity suite are very conservative, which could result in the loss of low-expressed but biologically relevant transcripts. To provide the most complete set of transcripts, the pipeline was designed with lower stringency during the assembly step in order to produce every possible transcript and not miss low-expressed messenger RNA. A stringent filtration step is included afterward in order to provide a set of transcripts that make sense biologically. A schema of pipeline steps can be found in Fig. S5.

ChIP-Seq pipeline

The ChIP-Seq workflow is based on the ENCODE [1] workflow. It aligns reads using the Burrows-Wheeler Aligner. It creates tag directories using Homer [34]. Peaks are called using MACS2 [35] and annotated using Homer. Binding motifs are also identified using Homer. Metrics are calculated based on IHEC requirements [36]. The ChIP-Seq pipeline can also be used for assay for transposase-accessible chromatin using sequencing (ATAC-Seq) samples. However, we are developing a pipeline that is specific to ATAC-Seq. A schema of pipeline steps can be found in Fig. S6.

The Tumour Analysis pipeline

The Tumour Pair workflow inherits the bam processing protocol from DNA-seq implementation to retain the benchmarking optimizations but differs in alignment refinement and mutation identification by maximizing the information utilizing both tumour and normal samples together. The pipeline is based on an ensemble approach, which was optimized using both the DREAM3 challenge [37] and the CEPH mixture datasets to select the best combination of callers for both SNV and structural variation detection. For SNVs, multiple callers such as GATK mutect2, VarScan2 [38], bcftools, and VarDict [39] were combined to achieve a sensitivity of 97.5%, precision of 98.8%, and F1 score of 98.1% for variants found in ≥2 callers. Similarly, SVs were identified using multiple callers: DELLY [40], LUMPY [41], WHAM [42], CNVkit [43], and Svaba [44] and combined using MetaSV [45] to achieve a sensitivity of 84.6%, precision of 92.4%, and F1 score of 88.3% for duplication variants found in the DREAM3 dataset (for more details, refer to Supplementary Material). The pipeline also integrates specific cancer tools to estimate tumour purity and tumour ploidy of sample pair normal–tumour. Additional annotations are incorporated to the SNV calls using dbNSFP [20] and/or Gemini [21], and QC metrics are collected at various stages and visualized using MulitQC [22]. This pipeline has 3 protocols (sv, ensemble, or fastpass). Schemas of pipeline steps for the 3 protocols can be found in Figs S7–S9.

Whole-genome bisulfite sequencing pipeline (WGBS or methylation sequencing)

The methylation sequencing workflow is adapted from the Bismark pipeline [46]. It aligns paired-end reads with bowtie2 default mode. Duplicates are removed with Picard, and methylation calls are extracted using Bismark [46]. Wiggle tracks for both read coverage and methylation profile are generated for visualization. Variant calls can be extracted from the whole-genome bisulfite sequencing (WGBS) data directly using bisSNP [47]. Bisulfite conversion rates are estimated with lambda genome or from human non-CpG methylation directly. Several metrics based on IHEC requirements are also calculated. Methylation sequencing can also process capture data if provided with a capture bed file. A schema of pipeline steps can be found in Fig. S10.

Hi-C pipeline

The HiC-Seq workflow aligns reads using HiCUP [48]. It creates tag directories, produces interaction matrices, and identifies compartments and significant interactions using Homer. It identifies topologically associating domains using TopDom [49] and RobusTAD [50] (bioRxiv 293175). It also creates ".hic" files using JuiceBox [51] and metrics reports using MultiQC [22]. The HiC-Seq workflow can also process capture Hi-C data with the flag "-t capture" using CHICAGO [52]. Schemas for the Hi-C and capture Hi-C protocols of this pipeline can be found in Figs S11 and S12, respectively.

The metagenomic pipeline (ribosomal RNA gene amplification analysis)

This pipeline is based on the established QIIME procedure [53] for amplicon-based metagenomics. It assembles read pairs using FLASH [54], detects chimeras with uchime [55], and picks operational taxonomic units using vsearch [56]. Operational taxonomic units are then aligned using PyNAST [57] and clustered with FastTree [58]. Standard diversity indices, taxonomical assignments, and ordinations are then calculated and reported graphically. A schema of pipeline steps can be found in Fig. S13.

The PacBio pipeline

The PacBio whole-genome assembly pipeline is built following the HGAP method [33], including additional features, such as base modification detection [59] and genome circularization [60]. *De novo* assembly is performed using PacBio's SMRT Link software [61]. Assembly contigs are generated using HGAP4. Alignments are then corrected and used as seeds by FALCON [62] to create contigs. The resulting contigs are then polished and processed by "Arrow" [63], which ultimately generates high-quality consensus sequences. An optional step allowing assembly circularization is integrated at the end of the pipeline. A schema of pipeline steps can be found in Fig. S14.

Comparison with other solutions for next-generation sequencing analysis

Data collected for select tools modified from Griffith et al. [64] (Table 1) show that GenPipes' strength lies in its robust WMS that comes with one of the most diverse selection of analysis pipelines that have been thoroughly tested. The pipelines in the framework cover a wide range of sequencing applications (Fig. 2a). The pipelines are end-to-end workflows running complete bioinformatics analyses. While many available pipelines conclude with a bam file or run limited post-bam analysis steps, the pipelines included in GenPipes are extensive, often having as many as 40 different steps that cover a wide range of post-bam processing. It is important to note that GenPipes, as well as several other WMSs, like Nextflow [65] and SnakeMake [66], support community-developed pipelines; however, those have not been included in the comparison.

GenPipes is compatible with HPC computing, as well as cloud computing [67], and includes a workflow manager that can be adapted to new systems. GenPipes also provides job status tracking through JSON files that can then be displayed on a web portal (an official portal for GenPipes will be released soon). GenPipes' available pipelines facilitate bioinformatics processing, while the framework makes it flexible for modifications and new implementations.

GenPipes developers offer continuous support through a Google forum page [68] and a help desk email address (pipelines@computationalgenomics.ca). Since the release of version 2.0.0 in 2014, a community of users has run GenPipes to conduct approximately 3,000 analyses processing ∼100,000 samples (Fig. 2b and c).

Discussion and Conclusion

GenPipes is a workflow management system that facilitates building robust genomic workflows. GenPipes is a unique solution that combines both a framework for development and end-to-end analysis pipelines for a very large set of genomics fields. The efficient framework for pipeline development has resulted in a broad community of developers with >30 active branches and >10 forks of the GenPipes repository. GenPipes has several optimized features that adapt it to large-scale data analysis, namely:

- Multiple schedulers: GenPipes is optimized for HPC processing. It currently accommodates 4 schedulers.
- Job dependencies: GenPipes establishes dependencies among its different steps. This enables launching all the steps at the same time and minimizes queue waiting time and management.
- Smart relaunch: GenPipes sets and detects flags at each successful step in the pipeline. This allows the detection of successfully completed steps and easy relaunch of failed steps.
- Parameter encapsulation: Genpipes uses a superposed configuration system to parse all required parameters from configuration files. This simplifies the use of the framework and makes it more flexible to user adjustments. Tested configuration files that are tailored to different clusters and different species are included with GenPipes.
- Diverse inputs: GenPipes has been developed to launch using different starting inputs, making it more flexible.
- Flexible workflows: GenPipes implements a workflow in steps. Users can choose to run specific steps of interest, limiting waste of time and resources.

GenPipes is under continuous development to update established pipelines and to create new pipelines for emerging technologies. For instance, new genomics pipelines are being developed for ATAC-Seq, single cell RNA-Seq, and HiChIP. GenPipes is also being redeveloped to use the CWL to provide a cloud-compatible version more seamlessly, and more Scheduler objects, like DRMAA, are being added to expand compatibility with more platforms. GenPipes has become a reliable bioinformatics solution that has been used in various genomics publications for DNA-Seq [69–76], RNA-Seq [77], and ChIP-Seq [78] analyses. GenPipes is currently available as source code, as well as a Docker image for easy installation and use. GenPipes has been optimized for HPC systems but can run on a laptop computer on small datasets.

Availability of Source Code and Requirements

- Project name: GenPipes
- Project home page: http://www.c3g.ca/genpipes
- Operating system(s): Linux; can be used on Windows and Mac OS using Docker
- Programming language: Python
- Other requirements: Workflow-dependent; detailed in documentation
- License: GNU GPLv3
- SciCrunch RRID:SCR_016376

Table 1: Comparisonof available solutions for NGS analysis.

		Features												Pipelines								
Solution	Language	Software license	Published	Free	Open source	Cloud/Container	HPC	Workflow manager	Progress Monitoring	Package Manager	GUI	Reports	Config Validation	Germline	Somatic	RNA-Seq	RNA-Seq De novo	ChIP-seq	Metagenome	Methyl-Seq	Hi-C	PacBio assembly
GenPipes	Python	GNU LGPL	Pending	✓	✓	✓	✓	✓	✓	✗	✗	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Genome Modeling System	Perl	GNU LGPLv3	[64]	✓	✓	✓	✓	✓	✓	✗	✗	✓	✓	✓	✓	✓	✗	✗	✗	✗	✗	✗
Galaxy	Python	Academic Free L3.0	[4]	✓	✓	✓	✓	✓	✓	✓	✓	✗	✗	✓	✓	✓	✓	✓	✓	✗	✗	✗
bcbio-nextgen	Python	MIT License	No	✓	✓	✓	✓	✓	✗	✗	✗	✓	N/A	✓	✓	✓	✗	✓	✗	✗	✗	✗
Omics Pipe	Python	MIT License	[79]	✓	✓	✓	✓	✓	✗	✓	✗	✓	✗	✓	✓	✓	✗	✓	✗	✗	✗	✗
Gene Pattern	Java	Custom	[80]	✓	✓	✓	✓	✓	✓	✗	✓	✓	✗	✓	N/A	✓	✗	✗	✗	✗	✗	✗
Illumina BaseSpace	bash	Custom	No	✗	✗	✓	✓	✓	✓	✗	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗	✗
BINA Genomic Analysis System	Java/Python	Custom	No	✗	✗	✓	✓	✓	✓	✗	N/A	N/A	N/A	✓	✓	✓	N/A	✗	✗	✗	✗	✗
SeqWare	Java	GNU GPLv3	[81]	✓	✓	✓	✓	✓	✓	✗	✓	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
DNA Nexus Platform	Python/bash	Custom	No	✓	Partial	✓	✗	✓	✓	✓	✓	✓	N/A	✓	✓	✗	✗	✗	✗	✗	✗	✗
gkno	Python	MIT License	No	✓	✓	✓	✓	✓	✗	✗	✗	✗	✓	✓	✗	✗	✗	✗	✗	✗	✗	✗
NGSANE	bash	BSD3	[82]	✓	✓	✓	✓	✓	✓	✗	✗	✓	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗
GATK's Queue	Scala	MIT License & Broad Institute	No	Partial	Partial	✗	N/A	✓	✓	✗	✗	N/A	N/A	✓	✓	✗	✗	✗	✗	✗	✗	✗
CGA's Firehose	Java	N/A	No	✓	✗	N/A	✓	✓	✓	✗	✓	✓	N/A	N/A	✓	✗	✗	✗	✗	✓	✗	✗
MIT STAR	Python	GNU GPLv3	[83]	✓	✓	✓	✗	✓	✗	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
CromWell/WDL	Scala	BSD 3-Clause	No	Partial	✓	✓	✓	✓	✓	✗	✗	N/A	✓	✓	✓	✗	✗	✗	✓	✗	✗	✗
BigDataScript	BDS	Apache License V2	[9]	✓	✓	✓	✓	✓	✓	✗	✗	✓	✗	✗	✓	✗	✗	✗	✗	✗	✗	✗
Kronos	Python	MIT license	[84]	✓	✓	✓	✓	✓	✗	✓	✗	✗	✗	✓	✓	✓	✗	✗	✗	✗	✗	✗
Nextflow	Java	GNU GPLv3	[7]	✓	✓	✓	✓	✓	✗	✓	✗	✓	✗	✗	✗	✓	✗	✗	✗	✓	✗	✗
SnakeMake	Python	MIT License	[6]	✓	✓	✓	✓	✓	✓	✓	✗	✓	✓	✗	✓	✓	✗	✗	✗	✗	✗	✗

Modified from Griffith & Griffith et al. [64]. Note that community-built pipelines are not considered in the Pipelines section of the table. It is also worth noting that the following table is meant to provide the reader with an overview of the features of several tools in the field but not necessarily an exhaustive list. For a full description of each tool's capabilities, please consult their official documentation.

Availability of Supporting Data and Materials

Snapshots of the code are available in the *GigaScience* GigaDB repository [85].

Additional Files

Supplementary_Materials.pdf includes:

Table 1: Performance of GenPipes' dnaseq pipeline on HG001 dataset with and without base recalibration.

Table 2: Performance of GenPipes' dnaseq pipeline on HG001 dataset using GATK3 and GATK4.

Table 3: Performance of GenPipes' dnaseq pipeline on HG001 dataset using various tools.

Table 4: Performance of GenPipes' dnaseq pipeline on HG001 dataset at various GATK variant recalibration step sensitivity levels.

Table 5: Performance of GenPipes' tumour pair pipeline on the DREAM3 dataset using various tools.

Table 6: Performance of GenPipes' tumour pair pipeline on the DREAM3 dataset using various tools followed by tool recommended filtering.

Table 7: Performance of GenPipes' tumour pair pipeline on the Ceph mixture dataset using various tools.

Table 8: Performance of GenPipes' tumour pair pipeline on the DREAM3 dataset using various structural variants callers.

FigureS1.pdf includes:

Figure S1: Schematic representation of GenPipes' dnaseq mpileup pipeline.

Figure S2: Schematic representation of GenPipes' dnaseq high coverage pipeline.

Figure S3: Schematic representation of GenPipes' rnaseq pipeline.

Figure S4: Schematic representation of GenPipes' rnaseq light pipeline.

Figure S5: Schematic representation of GenPipes' rnaseq de novo assembly pipeline.

Figure S6: Schematic representation of GenPipes' chipseq pipeline.

Figure S7: Schematic representation of GenPipes' tumour pair sv pipeline.

Figure S8: Schematic representation of GenPipes' tumour pair ensemble pipeline.

Figure S9: Schematic representation of GenPipes' tumour pair fastpass pipeline.

Figure S10: Schematic representation of GenPipes' methylseq pipeline.

Figure S11: Schematic representation of GenPipes' hicseq hic pipeline.

Figure S12: Schematic representation of GenPipes' hicseq capture pipeline.

Figure S13: Schematic representation of GenPipes' ampliconseq pipeline.

Figure S14: Schematic representation of GenPipes' pacbio assembly pipeline.

Abbreviations

ATAC-Seq: assay for transposase-accessible chromatin using sequencing; CHIP-Seq: chromatin immunoprecipitation sequencing; CWL: Common Workflow Language; DRMAA: Distributed Resource Management Application API; eFORGE: Experimentally Derived Functional Element Overlap Analysis of Regions from EWAS; ENCODE: Encyclopedia of DNA Elements; GATK: Genome Analysis Tool Kit; HPC: high-performance computing; IHEC: International Human Epigenome Consortium; indel: insertion and deletion; JSON: JavaScript Object Notation; PacBio: Pacific Biosciences; QC: quality control; RRBS: reduced-representation bisulfite sequencing; RNA-Seq: RNA sequencing; SMRT: Single Molecule, Real-Time; SNV: single-nucleotide variant; TORQUE: Terascale Open-source Resource and Queue Manager; WDL: Workflow Description Language; WGBS: whole-genome bisulfite sequencing; WGS: whole-genome seqencing; WMS: workflow management system.

Competing Interests

The authors declare that they have no competing interests.

Funding

This work was supported by CANARIE, Compute Canada, and Genome Canada. Additional support came from a grant from the National Sciences and Engineering Research Council (NSERC-448167-2013) and a grant from the Canadian Institute for Health Research (CIHR-MOP-115090). G.B. is also supported by the Fonds de Recherche Santé Québec (FRSQ-25348).

Authors' Contributions

MB, GB, MC, LL and JF designed GenPipes. MB, RD, RE, LL, JF, MM, MC, JS, FL, GL, EM, DB, PM, PTV, DALM, JT, XS, EH, EG and POQ developed GenPipes. MB, RD, LL, JF, EH, KCC, POQ and BC contributed to the portability of GenPipes over the different HPC servers. MB, RD and GB wrote the manuscript. All authors revised and approved the manuscript.

Acknowledgements

Data analyses were enabled by compute and storage resources provided by Compute Canada and Calcul Québec. We would also like to acknowledge Romain Gregoire and Tushar Dubey for their contribution to the code and Patricia Goerner-Potvin for her help in planning the report content.

References

1. ENCODE. The ENCODE (ENCyclopedia Of DNA Elements) Project. Science 2004;**306**(5696):636–40.
2. Stunnenberg HG, Hirst M. The International Human Epigenome Consortium: a blueprint for scientific collaboration and discovery. Cell 2016;**167**(5):1145–9.
3. Mardis ER. The $1,000 genome, the $100 000 analysis? Genome Med 2010;**2**(11):84.
4. Afgan E, Baker D, van den Beek M et al. The Galaxy platform for accessible, reproducible and collaborative biomedical analyses: 2016 update. Nucleic Acids Res 2016;**44**(W1):W3–W10.
5. DNANexus website. https://www.dnanexus.com/. Accesed September 2018.
6. Koster J, Rahmann S. Snakemake–a scalable bioinformatics workflow engine. Bioinformatics 2012;**28**(19):2520–2.
7. Di Tommaso P, Chatzou M, Floden EW, et al. Nextflow enables reproducible computational workflows. Nat Biotechnol 2017;**35**(4):316–9.
8. Sadedin SP, Pope B, Oshlack A. Bpipe: a tool for running and managing bioinformatics pipelines. Bioinformatics

2012;**28**(11):1525–6.

9. Cingolani P, Sladek R, Blanchette M. BigDataScript: a scripting language for data pipelines. Bioinformatics 2015;**31**(1):10–6.
10. GenPipes. https://bitbucket.org/mugqic/genpipes/src/master/. Accesed May 2019.
11. Compute Canada. https://www.computecanada.ca. Accesed May 2019.
12. Compute Canada: Apply for an account. https://www.computecanada.ca/research-portal/account-management/apply-for-an-account/. Accessed May 2019.
13. Buncic P, Aguado Sanchez C, Blomer J, et al. CernVM - a virtual software appliance for LHC applications. J Phys A 2010;**219**:042003.
14. GenPipes tutorial. http://www.computationalgenomics.ca/tutorials/. Accessed May 2019.
15. Breeze CE, Paul DS, van Dongen J, et al. eFORGE: a tool for identifying cell type-specific signal in epigenomic data. Cell Rep 2016;**17**(8):2137–50.
16. Van der Auwera GA, Carneiro MO, Hartl C, et al. From FastQ data to high confidence variant calls: the Genome Analysis Toolkit best practices pipeline. Curr Protoc Bioinformatics 2013;**43**:11.10.1–33.
17. Li H, Durbin R. Fast and accurate short read alignment with Burrows-Wheeler transform. Bioinformatics 2009;**25**(14):1754–60.
18. Li H, Handsaker B, Wysoker A, et al. The Sequence Alignment/Map format and SAMtools. Bioinformatics 2009;**25**(16):2078–9.
19. Zook JM, Catoe D, McDaniel J, et al. Extensive sequencing of seven human genomes to characterize benchmark reference materials. Sci Data 2016;**3**:160025.
20. Liu X, Wu C, Li C, et al. dbNSFP v3.0: a one-stop database of functional predictions and annotations for human nonsynonymous and splice-site SNVs. Hum Mutat 2016;**37**(3):235–41.
21. Paila U, Chapman BA, Kirchner R, et al. GEMINI: integrative exploration of genetic variation and genome annotations. PLoS Comput Biol 2013;**9**(7):e1003153.
22. Ewels P, Magnusson M, Lundin S, et al. MultiQC: summarize analysis results for multiple tools and samples in a single report. Bioinformatics 2016;**32**(19):3047–8.
23. Dobin A, Davis CA, Schlesinger F, et al. STAR: ultrafast universal RNA-seq aligner. Bioinformatics 2013;**29**(1):15–21.
24. Trapnell C, Williams BA, Pertea G, et al. Transcript assembly and quantification by RNA-Seq reveals unannotated transcripts and isoform switching during cell differentiation. Nat Biotechnol 2010;**28**(5):511–5.
25. Trapnell C, Hendrickson DG, Sauvageau M, et al. Differential analysis of gene regulation at transcript resolution with RNA-seq. Nat Biotechnol 2013;**31**(1):46–53.
26. Anders S, Pyl PT, Huber W. HTSeq–a Python framework to work with high-throughput sequencing data. Bioinformatics 2015;**31**(2):166–9.
27. Anders S, Huber W. Differential expression analysis for sequence count data. Genome Biol 2010;**11**(10):R106.
28. Robinson MD, McCarthy DJ, Smyth GK. edgeR: a Bioconductor package for differential expression analysis of digital gene expression data. Bioinformatics 2010;**26**(1):139–40.
29. DeLuca DS, Levin JZ, Sivachenko A, et al. RNA-SeQC: RNA-seq metrics for quality control and process optimization. Bioinformatics 2012;**28**(11):1530–2.
30. Young MD, Wakefield MJ, Smyth GK, et al. Gene ontology analysis for RNA-seq: accounting for selection bias. Genome Biol 2010;**11**(2):R14.
31. Bray NL, Pimentel H, Melsted P, et al. Near-optimal probabilistic RNA-seq quantification. Nat Biotechnol 2016;**34**(5):525–7.
32. Grabherr MG, Haas BJ, Yassour M, et al. Full-length transcriptome assembly from RNA-Seq data without a reference genome. Nat Biotechnol 2011;**29**(7):644–52.
33. Chin CS, Alexander DH, Marks P, et al. Nonhybrid, finished microbial genome assemblies from long-read SMRT sequencing data. Nat Methods 2013;**10**(6):563–9.
34. Heinz S, Benner C, Spann N, et al. Simple combinations of lineage-determining transcription factors prime cis-regulatory elements required for macrophage and B cell identities. Mol Cell 2010;**38**(4):576–89.
35. Zhang Y, Liu T, Meyer CA, et al. Model-based analysis of ChIP-Seq (MACS). Genome Biol 2008;**9**(9):R137.
36. IHEC standards. https://github.com/IHEC/ihec-assay-standards. Accessed May 2019.
37. Ewing AD, Houlahan KE, Hu Y, et al. Combining tumor genome simulation with crowdsourcing to benchmark somatic single-nucleotide-variant detection. Nat Methods 2015;**12**(7):623–30.
38. Koboldt DC, Zhang Q, Larson DE, et al. VarScan 2: somatic mutation and copy number alteration discovery in cancer by exome sequencing. Genome Res 2012;**22**(3):568–76.
39. Lai Z, Markovets A, Ahdesmaki M, et al. VarDict: a novel and versatile variant caller for next-generation sequencing in cancer research. Nucleic Acids Res 2016;**44**(11):e108.
40. Rausch T, Zichner T, Schlattl A, et al. DELLY: structural variant discovery by integrated paired-end and split-read analysis. Bioinformatics 2012;**28**(18):i333–9.
41. Layer RM, Chiang C, Quinlan AR, et al. LUMPY: a probabilistic framework for structural variant discovery. Genome Biol 2014;**15**(6):R84.
42. Kronenberg ZN, Osborne EJ, Cone KR, et al. Wham: identifying structural variants of biological consequence. PLoS Comput Biol 2015;**11**(12):e1004572.
43. Talevich E, Shain AH, Botton T, et al. CNVkit: genome-wide copy number detection and visualization from targeted DNA sequencing. PLoS Comput Biol 2016;**12**(4):e1004873.
44. Wala JA, Bandopadhayay P, Greenwald NF, et al. SvABA: genome-wide detection of structural variants and indels by local assembly. Genome Res 2018;**28**(4):581–91.
45. Mohiyuddin M, Mu JC, Li J, et al. MetaSV: an accurate and integrative structural-variant caller for next generation sequencing. Bioinformatics 2015;**31**(16):2741–4.
46. Krueger F, Andrews SR. Bismark: a flexible aligner and methylation caller for Bisulfite-Seq applications. Bioinformatics 2011;**27**(11):1571–2.
47. Liu Y, Siegmund KD, Laird PW, et al. Bis-SNP: combined DNA methylation and SNP calling for Bisulfite-seq data. Genome Biol 2012;**13**(7):R61.
48. Wingett S, Ewels P, Furlan-Magaril M, et al. HiCUP: pipeline for mapping and processing Hi-C data. F1000Res 2015;**4**:1310.
49. Shin H, Shi Y, Dai C, et al. TopDom: an efficient and deterministic method for identifying topological domains in genomes. Nucleic Acids Res 2016;**44**(7):e70.
50. Dali R, Bourque G, Blanchette M, A Tool for Robust Annotation of Topologically Associating Domain Boundaries.2018. bioRxiv:293175, doi.org/10.1101/293175.
51. Durand NC, Shamim MS, Machol I, et al. Juicer provides a one-click system for analyzing loop-resolution Hi-C experiments. Cell Syst 2016;**3**(1):95–8.
52. Cairns J, Freire-Pritchett P, Wingett SW, et al. CHiCAGO: ro-

bust detection of DNA looping interactions in Capture Hi-C data. Genome Biol 2016;**17**(1):127.

53. Kuczynski J, Stombaugh J, Walters WA, et al. Using QIIME to analyze 16S rRNA gene sequences from microbial communities. Curr Protoc Bioinformatics 2011;**Chapter 10**: p. Unit 10.7.
54. Magoc T, Salzberg SL. FLASH: fast length adjustment of short reads to improve genome assemblies. Bioinformatics 2011;**27**(21):2957–63.
55. Edgar RC, Haas BJ, Clemente JC, et al. UCHIME improves sensitivity and speed of chimera detection. Bioinformatics 2011;**27**(16):2194–200.
56. Rognes T, Flouri T, Nichols B, et al. VSEARCH: a versatile open source tool for metagenomics. PeerJ 2016;**4**:e2584.
57. Caporaso JG, Bittinger K, Bushman FD, et al. PyNAST: a flexible tool for aligning sequences to a template alignment. Bioinformatics 2010;**26**(2):266–7.
58. Price MN, Dehal PS, Arkin AP. FastTree: computing large minimum evolution trees with profiles instead of a distance matrix. Mol Biol Evol 2009;**26**(7):1641–50.
59. Methylome Analysis Technical Note. https://github.com/PacificBiosciences/Bioinformatics-Training/wiki/Methylome-Analysis-Technical-Note.
60. Hunt M, Silva ND, Otto TD, et al. Circlator: automated circularization of genome assemblies using long sequencing reads. Genome Biol 2015;**16**:294.
61. https://www.pacb.com/products-and-services/sequel-system/software/. Accessed May 2019.
62. FALCON wiki. https://github.com/PacificBiosciences/FALCON/wiki/. Accessed Septmeber 2018.
63. GenomicConsensus.s https://github.com/PacificBiosciences/GenomicConsensus. Accessed May 2019.
64. Griffith M, Griffith OL, Smith SM, et al. Genome modeling system: a knowledge management platform for genomics. PLoS Comput Biol 2015;**11**(7):e1004274.
65. NextFlow Community Pipelines. https://github.com/nf-core. Accessed Septmeber 2018.
66. SnakeMake Community Pipelines. https://github.com/snakemake-workflows. Accessed Septmeber 2018.
67. GenPipes Cloud. http://www.computationalgenomics.ca/genpipes-in-the-cloud/. Accessed May 2019.
68. GenPipes GoogleForum. https://groups.google.com/forum/#!forum/GenPipes. Accessed May 2019.
69. Buczkowicz P, Hoeman C, Rakopoulos P, et al. Genomic analysis of diffuse intrinsic pontine gliomas identifies three molecular subgroups and recurrent activating ACVR1 mutations. Nat Genet 2014;**46**(5):451–6.
70. Scelo G, Riazalhosseini Y, Greger L, et al. Variation in genomic landscape of clear cell renal cell carcinoma across Europe. Nat Commun 2014;**5**:5135.
71. Le Guennec K, Quenez O, Nicolas G, et al. 17q21.31 duplication causes prominent tau-related dementia with increased MAPT expression. Mol Psychiatry 2017;**22**(8):1119–25.
72. Torchia J, Golbourn B, Feng S, et al. Integrated (epi)-genomic analyses identify subgroup-specific therapeutic targets in CNS rhabdoid tumors. Cancer Cell 2016;**30**(6):891–908.
73. Oliazadeh N, Gorman KF, Eveleigh R, et al. Identification of elongated primary cilia with impaired mechanotransduction in idiopathic scoliosis patients. Sci Rep 2017;**7**:44260.
74. Bellenguez C, Charbonnier C, Grenier-Boley B, et al. Contribution to Alzheimer's disease risk of rare variants in TREM2, SORL1, and ABCA7 in 1779 cases and 1273 controls. Neurobiol Aging 2017;**59**:220.e1–9.
75. Hamdan FF, Myers CT, Cossette P, et al. High rate of recurrent de novo mutations in developmental and epileptic encephalopathies. Am J Hum Genet 2017;**101**(5):664–85.
76. Monlong J, Girard SL, Meloche C, et al. Global characterization of copy number variants in epilepsy patients from whole genome sequencing. PLoS Genet 2018;**14**(4):e1007285.
77. Manku G, Hueso A, Brimo F, et al. Changes in the expression profiles of claudins during gonocyte differentiation and in seminomas. Andrology 2016;**4**(1):95–110.
78. Deblois G, Smith HW, Tam IS, et al. ERRalpha mediates metabolic adaptations driving lapatinib resistance in breast cancer. Nat Commun 2016;**7**:12156.
79. Fisch KM, Meißner T, Gioia L, et al. Omics Pipe: a community-based framework for reproducible multi-omics data analysis. Bioinformatics 2015;**31**(11):1724–8.
80. Reich M, Liefeld T, Gould J, et al. GenePattern 2.0. Nat Genet 2006;**38**(5):500–1.
81. O'Connor BD, Merriman B, Nelson SF. SeqWare Query Engine: storing and searching sequence data in the cloud. BMC Bioinformatics 2010;**11**(Suppl 12):S2.
82. Buske FA, French HJ, Smith MA, et al. NGSANE: a lightweight production informatics framework for high-throughput data analysis. Bioinformatics 2014;**30**(10):1471–2.
83. Ceraj I, Riley JT, Shubert C, StarHPC - Teaching Parallel Programming within Elastic Compute Cloud. In: Proceedings of the ITI 2009 31st International Conference on Information Technology Interfaces, 2009, Cavtat, Croatia. 2009.
84. Taghiyar MJ, Rosner J, Grewal D, et al. Kronos: a workflow assembler for genome analytics and informatics. Gigascience 2017;**6**(7):1–10.
85. Bourgey M, Dali R, Eveleigh R, et al. Supporting data for "GenPipes: an open-source framework for distributed and scalable genomic analyses." GigaScience Database 2019. http://dx.doi.org/10.5524/100575.