

Article

Next Generation Sequence Analysis and Computational Genomics Using Graphical Pipeline Workflows

Federica Torri ^{1,2}, Ivo D. Dinov ^{2,3}, Alen Zamanyan ³, Sam Hobel ³, Alex Genco ³, Petros Petrosyan ³, Andrew P. Clark ⁴, Zhizhong Liu ³, Paul Eggert ^{3,5}, Jonathan Pierce ³, James A. Knowles ⁴, Joseph Ames ², Carl Kesselman ², Arthur W. Toga ^{2,3}, Steven G. Potkin ^{1,2}, Marquis P. Vawter ⁶ and Fabio Macciardi ^{1,2,*}

¹ Department of Psychiatry and Human Behavior, University of California, Irvine, CA 92617, USA; E-Mails: ftorri@uci.edu (F.T.); sgpotkin@uci.edu (S.G.P.)

² Biomedical Informatics Research Network (BIRN), Information Sciences Institute, University of Southern California, Los Angeles, CA 90292, USA; E-Mails: ivo.dinov@loni.ucla.edu (I.D.D.); jdames@uci.edu (J.A.); carl@isi.edu (C.K.); toga@loni.ucla.edu (A.W.T.)

³ Laboratory of Neuro Imaging (LONI), University of California, Los Angeles, CA 90095, USA; E-Mails: Alen.Zamanyan@loni.ucla.edu (A.Z.); shobel87@gmail.com (S.H.); alexgenco@gmail.com (A.G.); Petros.Petrosyan@loni.ucla.edu (P.P.); zhizhong.liu@loni.ucla.edu (Z.L.); eggert@cs.ucla.edu (P.E.); jonathan.pierce@loni.ucla.edu (J.P.)

⁴ Zilkha Neurogenetic Institute, USC Keck School of Medicine, Los Angeles, CA 90033, USA; E-Mails: clarkap@usc.edu (A.P.C.); knowles@med.usc.edu (J.A.K.)

⁵ Department of Computer Science, University of California, Los Angeles, CA 90095, USA

⁶ Functional Genomics Laboratory, Department of Psychiatry And Human Behavior, School of Medicine, University of California, Irvine, CA 92697, USA; E-Mail: mvawter@uci.edu

* Author to whom correspondence should be addressed; E-Mail: fmacciar@uci.edu; Tel.: +1-949-824-4559; Fax: +1-949-824-2072.

Received: 6 July 2012; in revised form: 15 August 2012 / Accepted: 15 August 2012 /

Published: 30 August 2012

Abstract: Whole-genome and exome sequencing have already proven to be essential and powerful methods to identify genes responsible for simple Mendelian inherited disorders. These methods can be applied to complex disorders as well, and have been adopted as one of the current mainstream approaches in population genetics. These achievements have been made possible by next generation sequencing (NGS) technologies, which require substantial bioinformatics resources to analyze the dense and complex sequence data. The

huge analytical burden of data from genome sequencing might be seen as a bottleneck slowing the publication of NGS papers at this time, especially in psychiatric genetics. We review the existing methods for processing NGS data, to place into context the rationale for the design of a computational resource. We describe our method, the Graphical Pipeline for Computational Genomics (GPCG), to perform the computational steps required to analyze NGS data. The GPCG implements flexible workflows for basic sequence alignment, sequence data quality control, single nucleotide polymorphism analysis, copy number variant identification, annotation, and visualization of results. These workflows cover all the analytical steps required for NGS data, from processing the raw reads to variant calling and annotation. The current version of the pipeline is freely available at <http://pipeline.loni.ucla.edu>. These applications of NGS analysis may gain clinical utility in the near future (e.g., identifying miRNA signatures in diseases) when the bioinformatics approach is made feasible. Taken together, the annotation tools and strategies that have been developed to retrieve information and test hypotheses about the functional role of variants present in the human genome will help to pinpoint the genetic risk factors for psychiatric disorders.

Keywords: Next Generation Sequencing (NGS); LONI pipeline; SNPs; CNVs; workflow; bioinformatics

1. Review of the Current Methodologies and Tools for NGS DNA-Sequencing Data Analysis

The power and widespread availability of next-generation sequencing (NGS) platforms, has significantly broadened the scale of many DNA-sequencing (DNA-Seq) applications, from detecting single nucleotide polymorphisms (SNPs) [1] or copy number variations (CNVs) [2], to assembling (new) genomes or transcriptomes [3], developing quantitative RNA-sequencing (RNA-Seq) analysis [4], or detecting epigenetic changes [5]. Among many various NGS applications, we focus this review on the existing methods for processing NGS DNA-Seq data.

NGS technology allows sequencing short fragments of DNA across the whole genome, producing single end (SE) or paired end (PE) reads of 50–700 base-pairs (bp). The reads might need some pre-processing conversion step (e.g., conversion between solexa and fastq format for data produced with version of the Illumina Pipeline previous than 1.8). The resulting raw DNA-Seq read data must then be analyzed following two computational macro-processes: (1) mapping and assembling, quality control, quality score re-calibration, realignment in “difficult” regions of the genome; and (2) advanced steps focused on variant calling (SNPs, insertions-deletions (Indels) and CNVs) and annotation. These macro-processes are briefly reviewed to provide a background for the software algorithms embedded in NGS analysis. The main software involved in NGS DNA-Seq are reviewed in Table 1 and briefly described below.

Table 1. Review of the most used software in next-generation sequencing (NGS) data analysis. Which includes two major computational macro-processes: (1) a primary step related to mapping and assembling, with alignment quality control, quality score re-calibration, realignment in “difficult” regions of the genome; and (2) secondary, advanced steps focused on variant (single nucleotide polymorphisms (SNPs), insertions-deletions (Indels) and copy number variations (CNVs)) calling and annotation. These macro-processes are briefly reviewed to provide a background for the software algorithms embedded in DNA-Seq analysis.

Process	Software & Algorithms	Website
Preprocessing step	homemade script	(N/A)
(1.1) Alignment	MAQ	http://maq.sourceforge.net
	BWA	http://bio-bwa.sourceforge.net/bwa.shtml
	BWA-SW (SE only)	http://bio-bwa.sourceforge.net/bwa.shtml
	PERM	http://code.google.com/p/perm/
	BOWTIE	http://bowtie-bio.sourceforge.net
	SOAPv2	http://soap.genomics.org.cn
	MOSAIK	http://bioinformatics.bc.edu/marthlab/Mosaik
	NOVOALIGN	http://www.novocraft.com/
(1.2) De novo Assembly	VELVET	http://www.ebi.ac.uk/%7Ezerbino/velvet
	SOAPdenovo	http://soap.genomics.org.cn
	ABYSS	http://www.bcgsc.ca/platform/bioinfo/software/abyss
(1.3) Basic QC	SAMTOOLS	http://sourceforge.net/projects/SAMtools/files/
	PICARD	http://picard.sourceforge.net/command-line-overview.shtml
(1.4) Advanced QC	GATK	http://www.broadinstitute.org/gsa/wiki/index.php/The_Genome_Analysis_Toolkit
	PICARD	http://picard.sourceforge.net/
	SAMTOOLS	http://sourceforge.net/projects/SAMtools/files/
	IGVtools	http://www.broadinstitute.org/igv/igvtools
(2.1a) Variant Calling and annotation		
<i>Sequence Variant Analyzer v1.0, for hg18 annotations</i>	SVA	http://www.svaproject.org/
	SAMTOOLS	http://sourceforge.net/projects/SAMtools/files/
	ERDS	http://www.duke.edu/~mz34/erds.htm
<i>SAMTOOLS and ANNOVAR for annotation</i>	SAMTOOLS	http://sourceforge.net/projects/SAMtools/files/
	ANNOVAR	http://www.openbioinformatics.org/annovar/
<i>UnifiedGenotyper and ANNOVAR for annotation</i>	GATK	http://www.broadinstitute.org/gsa/wiki/index.php/The_Genome_Analysis_Toolkit
	ANNOVAR	http://www.openbioinformatics.org/annovar/
(2.1b) CNVs		
CNVseq	CNVseq	http://tiger.dbs.nus.edu.sg/cnv-seq/
	R	http://www.r-project.org/
<i>SAMTOOLS/ERDS/Sequence variant analyzer v1.0 ERDS</i>	SVA	http://www.svaproject.org/
	SAMTOOLS	http://sourceforge.net/projects/SAMtools/files/
	ERDS	http://www.duke.edu/~mz34/erds.htm
CNVer	CNVer	http://compbio.cs.toronto.edu/CNVer/
	BOWTIE	http://bowtie-bio.sourceforge.net
	SAVANT	http://compbio.cs.toronto.edu/savant/
Simulated data generation tool	dwgsim	http://sourceforge.net/projects/dnaa/

1.1. Alignment

The first process in DNA-Seq data analysis involves *alignment* and *assembly*. *Alignment* is the process of mapping DNA-Seq reads to a reference genome. Many sequence alignment software tools that are available today use two main algorithms: the *hash-based* and the *Burrows-Wheeler Transform* methods.

Some hash-based algorithms build their hash table on the set of input reads (MAQ [6], Illumina's ELAND unpublished algorithm, SHRiMP [7], ZOOM [8]). Another set of tools build their hash table on the reference genome (SOAPv2 [9], BFAST, <http://genome.ucla.edu/bfast/>, MOSAIK <http://bioinformatics.bc.edu/marthlab/Mosaik/>, Novoalign <http://www.novocraft.com/main/index.php>, PERM [10]). After building the hash-table these methods can either use the reference genome to scan the hash table of input reads, or use the set of input reads to scan the hash table of the reference genome.

Many recent algorithms rely on the theory of string matching using Burrows-Wheeler Transform (BWT). BWT algorithms (BOWTIE [11], BWA [12], SOAPv2 [9]) typically create a suffix array from the BWT transformed sequence, rather than from the original sequence. In the first step, the sequence order of the reference genome is modified using the BWT, a reversible process (*i.e.*, the original genome sequence can be reconstructed backwards) that reorders the genome grouping together in the data structure the sequences that appear multiple times. Next, the final index is created and is used for rapid read placement on the genome. The main advantage of BWT algorithms is their speed, as they are much faster than hash-based algorithms at the same sensitivity level [3].

1.2. Assembly

Assembly starts from aligned DNA-Seq reads to reconstruct the original DNA sequence computationally, which generates large, continuous regions of DNA sequence [3]. Many alignment software provide tools to perform the assembly after the read alignment (e.g., MAQ), or standalone resources can be used (SAMTOOLS [13], Emboss [14]) or commercial packages like Geneious (<http://www.geneious.com>) and CLC-Bio (<http://www.clcbio.com>). For organisms without a sequenced reference genome, it is not possible to perform any reference genome guided assembly of the reads, thus *de novo* assembly is always an essential step for data analysis. The majority of *de novo* assemblers that have been released follow two basic approaches: overlap graphs [15] and de Bruijn graphs [16]. The overlap graph method calculates all the pair-wise overlaps between the reads and report this information in a graph. The manipulation of the same overlap graph leads to a layout of reads and then to a consensus sequence of contigs using Celera Assembler [17] or Arachne [18] among others. This traditional approach is computationally intensive as the overlap graph is extremely large even for simple organisms. De Bruijn graphs algorithm is used by most assemblers (Velvet [19], SOAPdeNOVO [20], ABySS [21]) and reduces the computational charge by breaking reads into smaller sub-sequences of DNA, called k-mers, where the k parameter describes the length in bases of these sequences [22]. The *de novo* assembly can be used also to resolve complex genomic region (e.g., rapidly evolving or rich in repetitive elements) of organisms with a reference genome. In this case the contigs are aligned back to the reference genome and can undergo all the next analytical steps here described.

1.3. Quality Control Improvement of Reads

There are many issues that must be considered when dealing with NGS data, beginning with the alignment of short reads. As an example, since each read is aligned independently, many reads spanning Indels may be misaligned. The per-base quality scores (*i.e.*, the probability that the called base in the read is the true base [23]) may also be inaccurate due to systematic errors in sequencing technology, machine cycle, and sequence context [24–26]. Thus, following the alignment and/or assembly of reads, quality control steps are implemented before continuing in the downstream analyses (in Section 2.1).

1.3.1. Basic Quality Control and File Formatting

A first basic quality control (QC) check involves formatting the aligned reads in a conventional format (e.g., Sequence Alignment/Map (SAM) or Binary Sequence Alignment/Map (BAM)). The output of this process is a clean, sorted, indexed file in BAM format that can be subjected to more advanced QC procedures, or be used directly in the downstream analyses.

1.3.2. Advanced QC

Additional advanced QC steps are strongly recommended since misaligned reads and inaccurate quality scores affect the reliability of the subsequent SNP discovery and genotyping steps, without correcting for such stochastic and systemic errors, the rate of false positive calls can be really high [27]. Even though we lack a gold standard for these procedures, there are computational tools to perform advanced QC on the data, in addition to some basic descriptive statistics and quality metrics visualization (SAMTOOLS [13], PICARD, <http://picard.sourceforge.net/>, Genome Analysis Toolkit (GATK) [27]). GATK supports locally realigning reads across regions enriched in Indels, recalibrating base quality scores of sequencing reads to correct for variation in quality with machine cycle and sequence context. This advanced QC of NGS data is probably the most important process to guarantee an accurate variant call, which is the immediate downstream analytical step.

1.4. Variant Calling and Annotation

Once the reads have been aligned and calibrated, SNPs, Indels and CNVs can be called. This step requires sensitive and specific statistical models and tools [6,9,12,13,28–48], named in Table 1.

1.4.1. SNPs and Indels Calling and Annotation

There are many algorithms that may be used to call SNPs from NGS data (SAMTOOLS [13], GATK, MAQ, SOAPv2, UnifiedGenotyperV2 within the GATK suite) and some recommended analytical and statistical frameworks [49], even if a gold standard for variant calling is still lacking as statistical methods for analyzing the data are constantly being released [49]. The SNPs and Indels are exported from these tools in variant call format (VCF), with much information related to each variant (*i.e.*, quality score, coverage, estimated genotype). Once the variants have been called they need to be

annotated, and this is a step that, until recently, only a few computational tools were able to accomplish like ANNOVAR [50], Sequence Variant Analyzer (SVA) [51], and GATK [27].

1.4.2. CNVs Calling

Furthermore, the field of computational methods for discovering structural variation on NGS data is still an open computational and bioinformatics challenge [2]. The CNVs discovery methods operate following a framework that allows detecting anomalous “signatures” or patterns, then calls the related variants using mainly four different approaches [2]: (1) read pair methods; (2) read-depth methods; (3) split read approaches; (4) *de novo* assembly.

1.5. Statistical and Variant Prioritization Analysis

Additional software such as PLINKseq (<http://atgu.mgh.harvard.edu/plinkseq/>) implement statistical models to analyze variants called from NGS experiments, testing for association with continuous or dichotomous traits and assessing an unusual distribution for rare variation across different functional categories [52]. Some other tools like PolyPhen2 [53] and VAAST [54] can be used afterwards for functional variant annotation and prioritization providing hints on the biology and pathophysiology of psychiatric disorders. Also, alternative annotation tools and strategies have been proposed [55] to retrieve information and test hypotheses about the functional role of variants present by chance in any single human genome or enriched in the genome of people affected by a psychiatric disease [56,57].

1.6. Graphical Workflows

The development and release of algorithms and software for analysis of NGS data has seen exponential growth in the last two years, requiring a huge investment in terms of time, expertise and computer infrastructure.

From this brief review of tools (Table 1), it is evident that analyzing NGS data is a challenging and time consuming operation for scientists. Ideally, these tools must be up to date and easy to use, and their sequential combination should optimize performance and accuracy, with each program producing output files compatible with the input requirements of the software performing the following operation. Such processes sequentially linked together build what is generally called a workflow. An increasingly large number of workflows are available today to manage high-throughput genomics sequencing data, from basic data processing to high-quality visualization of results. Examples include shell-scripts [58,59], tool-specific graphical interfaces [60,61], and graphical workflow environments [62–65]. The graphical workflow environment are emerging as useful for constructing, modifying, interconnecting and executing computational genomics protocols using data processing workflows, also described as “pipelines” once the processes have been connected (Table 2).

Table 2. Comparison of several Graphical Workflow Environments to manage pipelines. Most workflow environments provide graphical solutions (infrastructures) for the interactive handling of data, with several advantageous features compared to the management of the same processes via command line or scripting interfaces. When adding new software tools, some of these architectures require software recompilation and some do not. Yet, there is significant variation of the status reports generated during or after workflow execution. Data storage, internal or external, operating system and local hardware dependencies and utilization of available grid managers also vary between the different workflow environments. There are many synergies between the Pipeline and various alternative environments for software tool integration and interoperability, with also some valuable differences. The Laboratory of Neuro Imaging (LONI) pipeline infrastructure provides computational workflow execution capability with or without the use of local hardware or administrative support. Adding new software tools to the pipeline library of tools is efficient, does not require recompiling the programs, and requires only a brief description of the tool invocation syntax using the client “module description” dialog. Thus, the LONI pipeline offers a flexibility and simplicity in design of novel workflow solutions that is not available in the other two most used systems for NGS data analysis, Taverna and Galaxy. Similarly, the LONI pipeline allows workflow pausing and resuming, and provides explicit controls ensuring that processes are only instantiated when the complete upstream activities have successfully completed execution. Additionally, the available Taverna and Galaxy services have restrictive upper limits on storage (100 GB) and per-process RAM (64 GB), when they are deployed on Amazon Web-Services/Cloud creating bottlenecks with data staging to/from the servers and computational runs. The Pipeline service provides a pair of dedicated open-access servers (<http://genomics.loni.ucla.edu>) each with 40-cores and 1.4 TB of shared RAM.

Workflow Management System	Module concatenation and interoperability	Asynchronous Task Management	Requires Tool Recompiling	Data Storage	Platform Independent	Client-Server Model	Grid Enabled
LONI Pipeline [57] pipeline.loni.ucla.edu	Y	Y	N	External	Y	Y	Y
Taverna [61] taverna.sourceforge.net	Y	N	Y	Internal (MIR)	Y	N	Y
Kepler [54] kepler-project.org	Y	N	Y	Internal (actors)	Y	N	Y
Triana [66] trianacode.org	Y	N	Y	Internal data structure	Y	N	Y
Workflow Navigation System [67] wns.nig.ac.jp	N	N	N/A	External	Y	N	N
Galaxy [55] usegalaxy.org	N	N	Y	External	N	Y	N
VisTrails [55] www.vistrails.org	Y	N	Y	Internal	N	N	N

Table 2. *Cont.*

Workflow Management System: list of the compared graphical workflow environments. *Module concatenation and interoperability*: *Asynchronous Task Management*: ability to submit new workflows and report the status of executing or completed workflows asynchronously, e.g., constant interruptions of network connectivity. *Requires Tool Recompiling*: requirement to recompile new computational libraries or software tools against the graphical environment libraries, and to restart the environment when provisioning these new services. *Data Storage*: ability of environment to store data (raw, processed and derived) internally (RAM/DB) or externally (NFS/Services). *Platform Independent*: dependency of the environment on the local hardware and operating system. *Platform independence refers to the workflow environment itself, not the computational library of tools that are accessible via that environment. For environments with Client-Server architecture, this is irrelevant, as the platform-independent clients can always connect to, submit data, process protocols and monitor the status of executing pipeline workflows by connecting to (possibly platform dependent) back-end pipeline servers where specific operating systems (most commonly Linux) may be required by many informatics and genomics computing libraries. Client-Server Model*: independent server and clients that can be broadly interconnected provided. *Grid Enabled*: use of a Grid Engine/Grid Job Manager. Legend: Y = yes, N = no.

There are several additional features in graphical workflow environments that simplify the data management. Table 2 lists some of the commonly used workflow environments and compares their core features. Each of the graphical environments described in Table 2 allows design and submission of new workflows.

The most commonly used systems in NGS analysis are Taverna [68] and Galaxy [64]. The latest beta-version of the Galaxy (<http://galaxy.psu.edu/>) platform offers a NGS computational framework that embeds single processing units to be invoked as web-services but it still lacks the functionality of interlinking the outputs of one process into a subsequent module. The available Taverna and Galaxy services have restrictive upper limits on storage and per-process RAM, when they are deployed on web-services, creating bottlenecks with data staging to/from the servers and computational runs. Also, the library of available Galaxy routines doesn't allow adding new tools and is limited to a few alignment software tools (e.g., BWA and Bowtie), or on the quality control side, an incomplete beta-version of the PICARD suite.

The LONI (Laboratory of Neuro Imaging) pipeline displays unique features for the implementation of new interactive and robust NGS analysis workflow protocols using a graphical environment. The design of novel workflow solutions within the LONI pipeline environment is simple and flexible [69], and allows embedding and connecting heterogeneous software within the same computational protocol without the need of advanced bioinformatics skills (Table 2).

The LONI pipeline architecture [70,71] is a distributed environment utilizing a client-server interface for design, validation, execution and dissemination of computational protocols as graphical workflows. Individual applications are represented as modules and may be linked to form complex network implementation of the desired analytical processes. Using a flexible, user friendly and customizable data processing and visualization system, the LONI pipeline environment provides access to distributed datasets, heterogeneous software tools and diverse web-services. Additional details about the LONI pipeline environment are available in [65].

Based upon the review of the NGS software (Table 1) and the graphical workflow environments to manage workflow (Table 2), we chose to develop within the LONI environment a graphical environment for genomics, called the Graphical Pipeline for Computational Genomics (GPCG) that covers many informatics analytical steps on NGS data. This effort was a joint collaboration between

LONI at UCLA, BIRN (Biomedical Informatics Research Network) at UCI, Information Sciences Institute (ISI) at USC.

The GPCG is a set of workflows that may simplify and speed-up approaches for sequencing projects performed on Illumina/Solexa Genome Analyzer-HiSeq platform [72–74]. We have implemented workflows in GPCG that: (1.1) aligns reads (both in single and paired end) and (1.2) performs *de novo* assembly with multiple algorithms; (1.3a) performs basic formatting and quality control, followed by a (1.3b) more advanced and complex quality control to correct for sequencing biases; (2.1a) performs SNP-Indels calling and annotation, and (2.1b) CNVs calling. The Supplementary Materials S1 provides detailed information about our experiments and results.

2. Description of the GPCG

The GPCG is a collection of “ready to use” workflows covering a broad spectrum of DNA-Seq data analysis steps. We have converted each single “command line” process into a module definition, and then connected those modules involved in the same process logically to form a workflow (Figure 1). We have generated a comprehensive Supplementary Materials Chapter S2 that includes all details to deploy the GPCG infrastructure, reproduce the workflow designs, and validate the results reported in this paper. The full description of all the workflows is reported in Supplementary Materials S1. In the data source section user can specify the location of input data, a process section where modules can be linked to perform complex analysis steps, and an output section where the name and location of the output files can be specified (Figure 1).

A single workflow (Figure 1) can be run independently from others, or can be connected as illustrated by the analytical workflow protocols (Figure 2 and Table 3). For example, the user can reconnect workflows by dragging and dropping within the GPCG creating a new pipeline. Modularity, reuse and interconnectivity are key features within the LONI environment, which make the system flexible for the analytical needs of the researcher.

The current implementation of DNA-Seq workflows is summarized in Table 3. Together with its flexible connections, the GPCG incorporates alternative algorithms for the vast majority of the processes. The choice of the algorithm is critical, as different approaches might produce different results.

Users may choose in fact the most suitable analytical model for their data or develop additional module descriptions interfacing other computational tools. Additional tools, workflows and updates are available on the LONI pipeline Navigator website: <http://pipeline.loni.ucla.edu/services/library-navigator>. As a large number of different file formats are involved in these workflows, we provide a glossary and examples of all the formats encountered in Supplementary Table S1.

We have implemented a preprocessing module that allows extracting a subset of reads to perform a validation and initial testing of pipeline modules before running an entire dataset. Embedded within this module are routines to convert read quality scores from Solexa FASTQ files to the Sanger scale (for data produced with Illumina pipeline versions previous than 1.8), and to binary FASTQ as requested by some aligners like MAQ [40,75]. Once the read subset is ready, it is possible to proceed to the two main computational processes previously described. We also embedded another module in the pipeline, “dwgsim”, that allows the user to generate simulated read datasets according to their analytical needs (see Supplementary Materials S1, Figure A1b, and Supplementary Materials S2).

Figure 1. A snapshot of the general organization of a workflow within the LONI pipeline environment. This is an example of embedded modules into an alignment workflow based on BWA software. The user can simply set up the location of the input files in the data sources, manage the programs involved in the core modules, and indicate the location of the output files in the output data sink section. Every section can be interactively edited or modified through a menu of options accessed by right-clicking the mouse on the respective portion of the workflow.

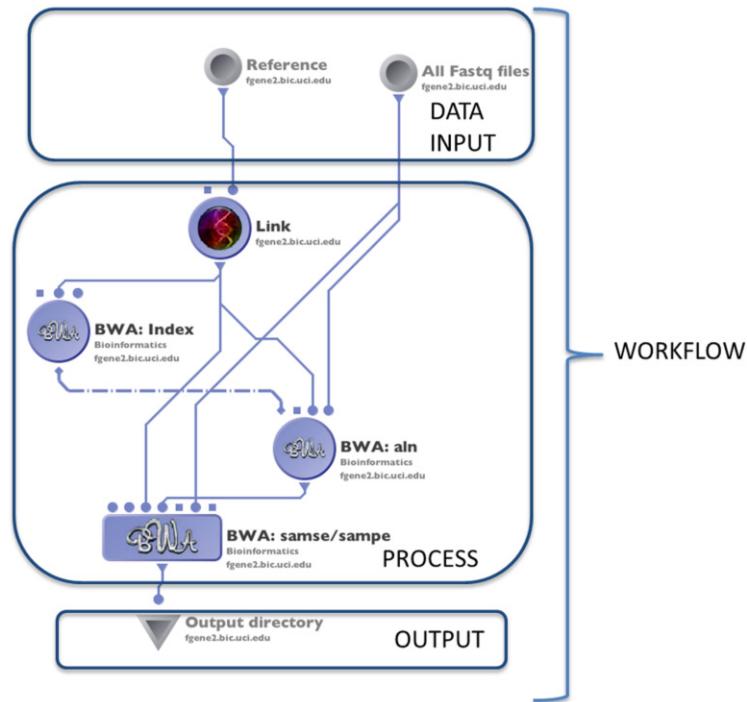


Figure 2. An example of the workflow approach to analyze DNA-Seq data in GPCG. Several alternative workflows can be run independently or connected in a logical flow. Once the reads have been pre-processed, they can be aligned (1.1), undergo (1.3) Basic and (1.4) Advanced QC, (2.1a) SNP/Indels and (2.1b) CNVs calling and annotation. The reads can also undergo (1.2) *de novo* assembly, and if a reference genome is available the contigs can be realigned back to the reference genome and then undergo the following computational processes.

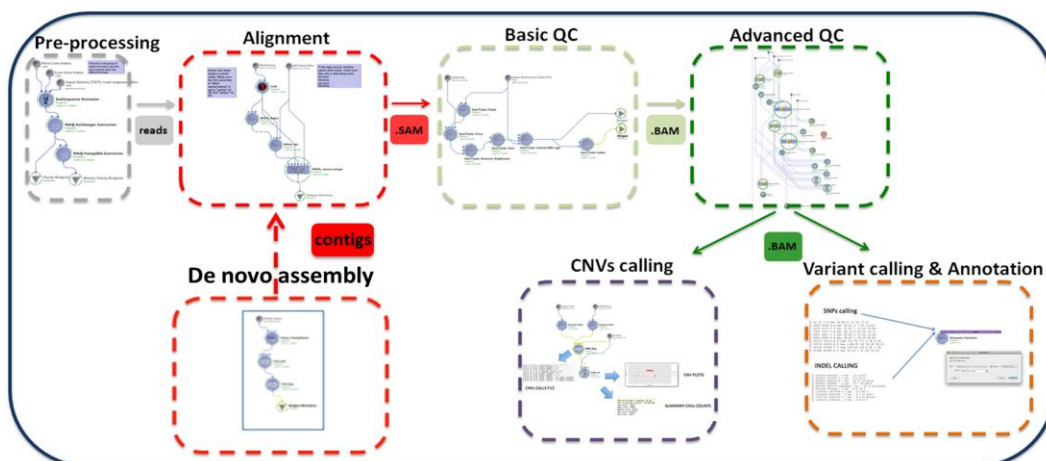


Table 3. Review of the processes and related workflows currently implemented in the NGS Pipeline. All processes and workflows have been tested and validated and are available for use by interested scientists. A single pipeline can be run independently from others, or can be connected as illustrated by the analytical workflow protocols described in this Table.

Process	Process Description	Software & Algorithms	Input *	Output (Files)	Upstream Module Dependencies	Downstream Module Dependencies
Preprocessing step	Test the NGS raw data and functionality	homemade script	reads (original solexa format)	subset of reads (fastq format)	none	(1.1) Alignment, (1.2) <i>De novo</i> assembly
(1.1) Alignment	Mapping the reads to the reference genome	MAQ	reads (binary fastq format)	SAM	Preprocessing	(1.2) Basic QC
		BWA	reads (fastq format)	SAM		
		BWA-SW (SE only)	reads (solexa format)	SAM		
		PERM	reads (fastq format)	SAM		
		BOWTIE	reads (solexa format)	SAM		
		SOAPv2	reads (fastq format)	SAM		
		MOSAİK	reads (solexa format)	SAM		
(1.2) <i>De novo</i> assembly	Build a <i>de novo</i> genome sequence	VELVET	reads (fastq format)	contigs file	Preprocessing	(1.1) Alignment or none ⁵
		SOAPdenovo	reads (fastq format)	contigs file		
		ABYSS	reads (fastq format)	contigs file		
(1.3) Basic QC	Basic Data formatting and quality control	PICARD, SAMTOOLS	SAM	BAM	(1.1) Alignment	(1.4) Advanced QC
(1.4) Advanced QC	QC for advanced issues	PICARD, SAMTOOLS, GATK	BAM	BAM clean	(1.3) Basic QC	(2.1a) Variant calling (2.1b) CNV analysis
(2.1a) Variant calling and annotation	Identify and visualize SNPs and Indels from the whole genome	Sequence Variant Analyzer v1.0	BAM clean	csv files with variants and annotation	(1.4) Advanced QC	Statistical analysis and visualization software [#]
		SAMTOOLS and ANNOVAR for annotation	BAM clean	txt files with variants		
		Unified genotyper and ANNOVAR for annotation	BAM clean	txt files with variants		
(2.1b) CNVs calling	Analysis of CNVs (ins & del > 1 Kb)	BOWTIE CNVer SAVANT	reads (solexa format)	txt file with the CNVs calls	(1.4) Advanced QC	Statistical analysis and visualization software [#]
		CNVseq	SAM	txt file with the CNVs calls		R (stat software)
		SAMTOOLS ERDS Sequence variant analyzer ERDS v1.0	BAM clean	csv file with the CNVs calls		Statistical analysis and visualization software [#]
Simulated data generation tool	Generate simulated reads according to the needs of the user	dwgsim	-	SE or PE .fastq files	-	(1.1) Alignment, (1.2) <i>De novo</i> assembly

* With solexa format we refer to the Phred quality score code used by the Illumina Pipeline version prior than 1.8 (Phred +64). The newer versions of the Illumina Pipeline produce reads file directly in Sanger format (Phred +33). To guarantee backwards compatibility with data produced by version of the Illumina Pipeline previous than 1.8 we have embedded a conversion step from Solexa FASTQ to Sanger FASTQ for the alignment software that don't support the solexa format. The user can remove this step in case the conversion is not needed; [#] External software like PLINQseq for the statistical analysis or IGV for visualization are not embedded in the workflow; ⁵ If a reference genome is not available the contigs can be used like they are for further analysis. If a reference genome is available the contigs can be aligned back to the reference genome with BWA-SW.

The GPCG pipeline environment allows users to change execution parameters for each software tool represented as a module in the pipeline workflow. This architecture allows changing parameters, adding new parameters or execution controls (e.g., flags, options), modifying values of existent parameters, removing parameters, as well as inserting new processing modules or connections to/from data objects (data-sources/inputs or data-sinks/outputs). These workflow modifications are accomplished directly in the pipeline graphical interface by using mouse-selection and keyboard entries.

2.1. Alignment

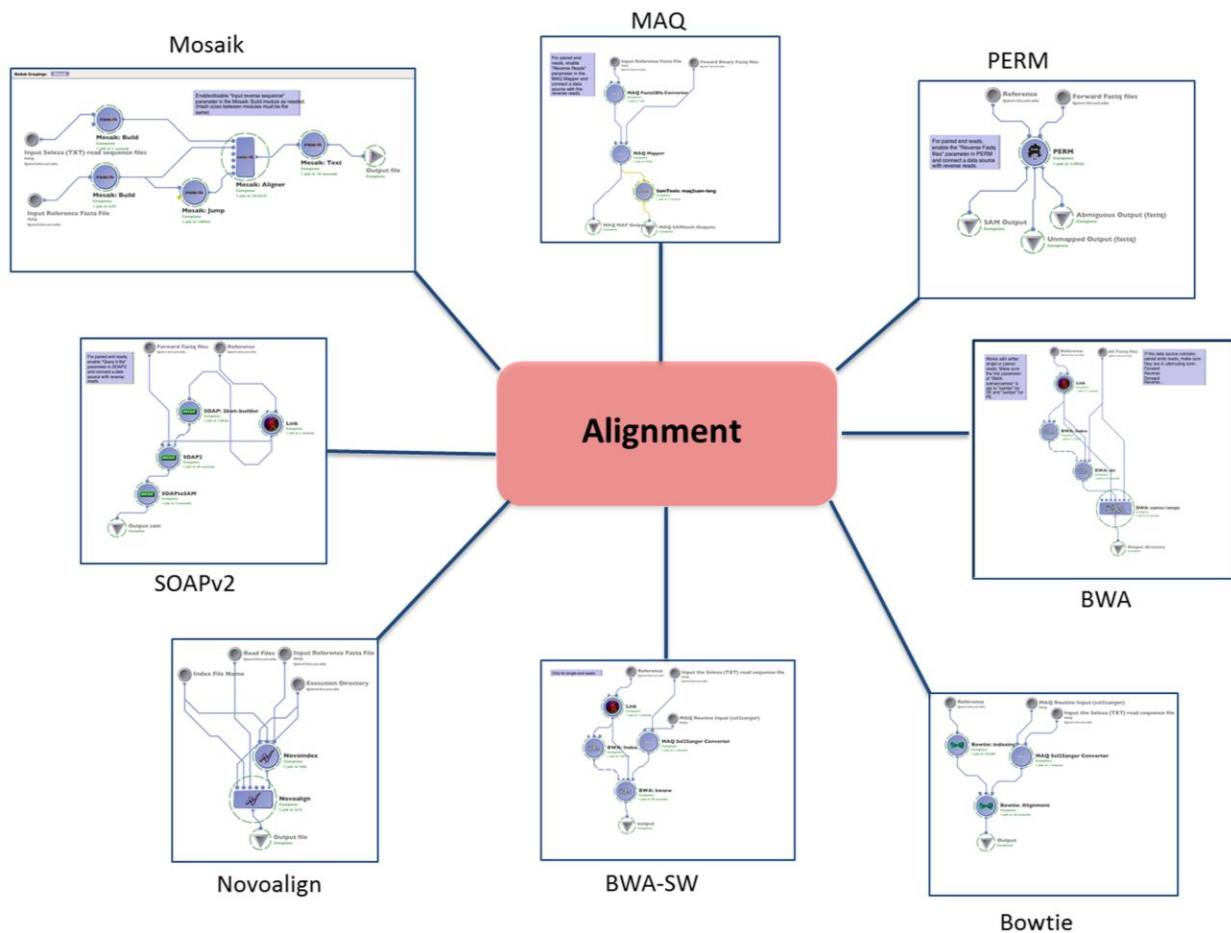
We incorporated into the LONI pipeline some of the most used alignment software tools (Figure 3). Through the “edit module” functionality of the graphical user interface (GUI) it is possible to visually manage all the parameters of the alignment software (e.g., number of allowed mismatches, read trimming, gap extension) encapsulated by the module itself. Each algorithm in a workflow (other than BWA-SW that runs only in single end) has a switch that allows the user to perform an alignment on either single end (SE) or paired end (PE) reads. The input reads are generally in FASTQ format. BWA accepts input read sequences also in BAM format. For paired-end data the pair reads need to be grouped together in one single BAM file. The final output is an alignment file in SAM format (see Supplementary Materials S1, Figure A2). This file is ready to undergo quality control procedures. The choice of the alignment software is critical as different software might lead to different results in term of mapping and downstream variants calling. It is strongly suggested to perform the alignment using more than one software and identify strategies that may provide better fit for the user’s data. BWA and BOWTIE, for example, are among the fastest alignment algorithms for both single and paired end reads, and are particularly suitable for whole genome alignments. They generally report by only one alignment hit, and in the default mode do not guarantee finding the best hit nor if the found hit is unique. In particular, for users dealing with highly repetitive genomic regions, it is possible to force these algorithms to output all the possible hits with a high cost in terms of speed, or to use a probabilistic software like PERM that outputs all the found hits. The user could perform a *de novo* assembly (see next paragraph) for a better resolution in such regions. If the main interest is the detection of CNVs with paired-end reads, Novoalign is strongly recommended because of the high mapping accuracy, even if slower than the previously mentioned Burrows-Wheeler (BW) based software.

For selecting parameters settings, users can find hints about the meaning and possible values of each single parameters by double-clicking on a parameter icon or via the execution info menu-accessible as a right-click on the module of interest. For the parameters tuning the users have to refer to the software documentation as all the algorithms works differently. The LONI Pipeline workflow environment enhances the user’s abilities to optimize the selection of the parameters for each tool by allowing specification of a list containing a range of parameters spanning the support of each parameter of interest. This *parameter-optimization* strategy is exactly the basis of the development of meta-algorithms as workflow protocols [76–78].

The quality control process we used to determine the tools to be incorporated is based on an up to date and detailed investigation of the available NGS tools, including research papers with technical and analytical details, and the online discussions panel/forums if available to better follow the

analytical trends and open issues. The testing with simulated and real data provided us further hints into the software reliability. Some information and guidelines to help in identifying the best set of software to analyze different type of data can be found in some about reference with comparisons among tools we provided [22,79].

Figure 3. Schematic representation of alignment modules available for both single and paired-end data (2.1).

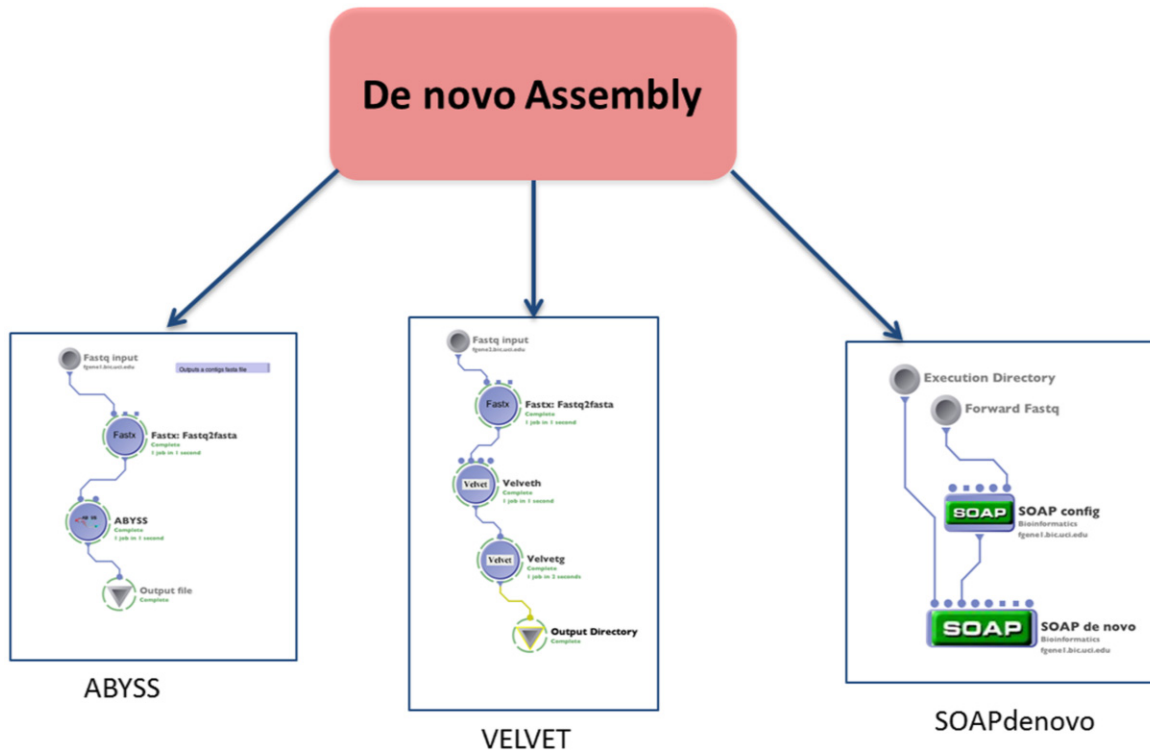


2.2. Assembly

When a reference genome is not available for the species under investigation it is necessary to perform *de novo* assembly of the reads. *De novo* assembly is also the suggested approach for reads mapping in regions prone to rearrangements, rapidly evolving, or where the reference genome might not be informative. We have developed three workflows embedding the most common *de novo* assembly software based on de Bruijn graphs (Figure 4), which works both with single and paired end data. One of the most important parameters to be tuned in *de novo* assembly is the fixed-length of the subsequences used to build the graphs length, which is abbreviated as k-mer. The most efficient k-mer size for a particular assembly is determined by the read length as well as the error rate, and it may be estimated *a priori* using “ad hoc” tools provided by the software developer or by evaluating the results obtained through varying the k-mer size across different runs. Some assemblers are also able to output scaffolds (*i.e.*, a set of contigs with known relative orientation and distance) or supercontigs (*i.e.*,

contigs in which gaps are allowed). When *de novo* assembly is used in complex region of organisms with a known reference genome, the contigs can be aligned back to it with the BWA-SW workflow, and then undergo the subsequent analytical steps.

Figure 4. Schematic representation of the *de novo* assembly workflows available (2.2).



2.3. Quality Control Improvement of Reads

2.3.1. Basic Quality Control and File Formatting

After the alignment the reads undergo first basic formatting and quality controls steps. We have developed a “Basic QC” workflow in line with the consolidated and updated procedures for DNA-Seq analyses (Figure 5).

The output of this module is a clean, sorted, indexed BAM file that can undergo advanced QC procedures, or used as it is for downstream analyses.

2.3.2. Advanced QC

Additional advanced QC issues can be addressed and fixed using the GATK toolkit (<http://www.broadinstitute.org/gsa/wiki>) [27,80]. The GATK developers suggested a core analytical framework that includes local realignment around Indels and base quality score recalibration [27,80]. We have embedded these processes in the “Advanced QC” workflows, adding other modules to produce quality control plots, statistics, and tracks useful for the visualization of the data (Figure 6).

Figure 5. A snapshot of the general organization of the Basic QC workflow (2.1.3). After an initial file cleaning that performs various fix ups, the alignment file in Sequence Alignment/Map (SAM) format is converted in Binary Sequence Alignment/Map (BAM) file and sorted. The workflow takes care of the duplicated reads removing or marking the potential PCR duplicates. If multiple read pairs have identical external coordinates, it only retains the pair with highest mapping quality. This step is particularly suited for paired end data and the user can switch between the two options simply changing the REMOVE_DUPLICATES argument in the GUI related to this module. The removal step can be excluded from a workflow run depending on the interest in studying repetitive elements. In case of paired end reads, the pipeline then ensures that all mate-pair information is in sync between each read and its mate pair, fixing any incoherent information. The BAM file undergoes MD tagging that adds string, labeling the mismatching positions. The BAM is finally indexed using the index of the reference genome.

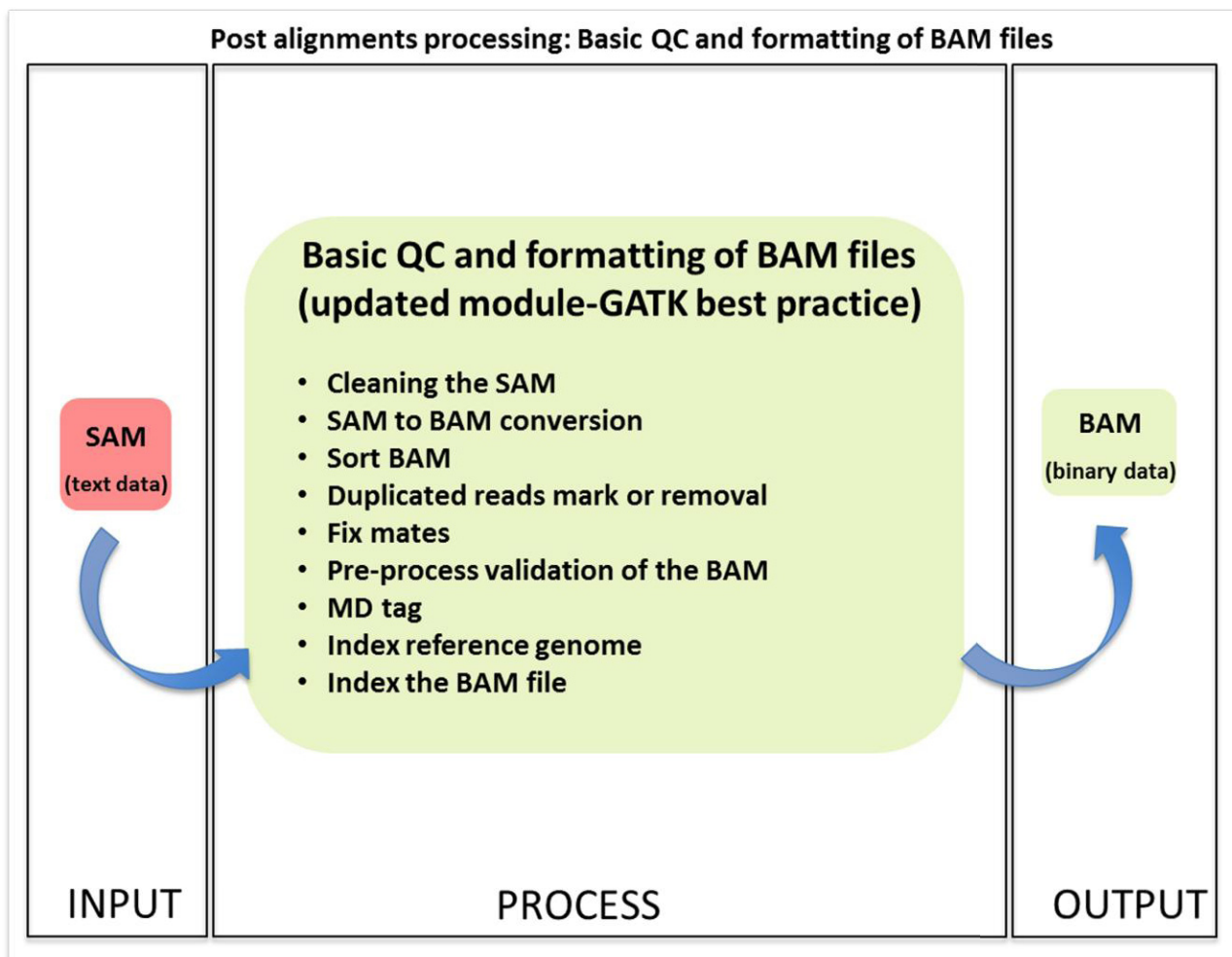


Figure 6. A snapshot of the general organization of the Advanced QC workflow (2.1.4). **(A)** After the basic QC, the reads that map within Indels in the individual's genome compared to the reference genome are locally realigned, as they may lead to alignment artifacts that can easily be misinterpreted as SNPs. The next step is the base quality score recalibration to recalibrate base quality scores of reads, by the analysis of the covariation among several features of a base (e.g., reported quality scores, the position within the read). The workflow produces plots and tables with the most important metrics for a DNA-Seq experiment (*i.e.*, mean quality by cycle, insert size metrics, quality score distribution, GC-bias metrics, main alignment metrics) with the PICARD software; **(B)** The users can then produce useful tracks for the visualization of the data in Integrative Genome Viewer (IGV). Examples are the (a) callability track (*i.e.*, evaluates how much a region can be trusted in term of coverage, accuracy and quality by GATK and can be visualized as a bar chart in IGV); (b) the sliding window coverage (*i.e.*, a computation of average alignment over a specified window size across the genome with igvtools). The main outputs of this step are: (1) a cleaned BAM file ready to be used for variant calling, (2) a set of plots and text files that can help the user to have a general picture about the general quality of the experiment and (3) a set of track files to visualize the dataset and its features. The user can upload the indexed BAM files and these tracks in IGV to visualize and annotate the reads across the whole genome with user-produced or online tracks (RefSeq, RepeatMasker, Database of Genomic Variants).

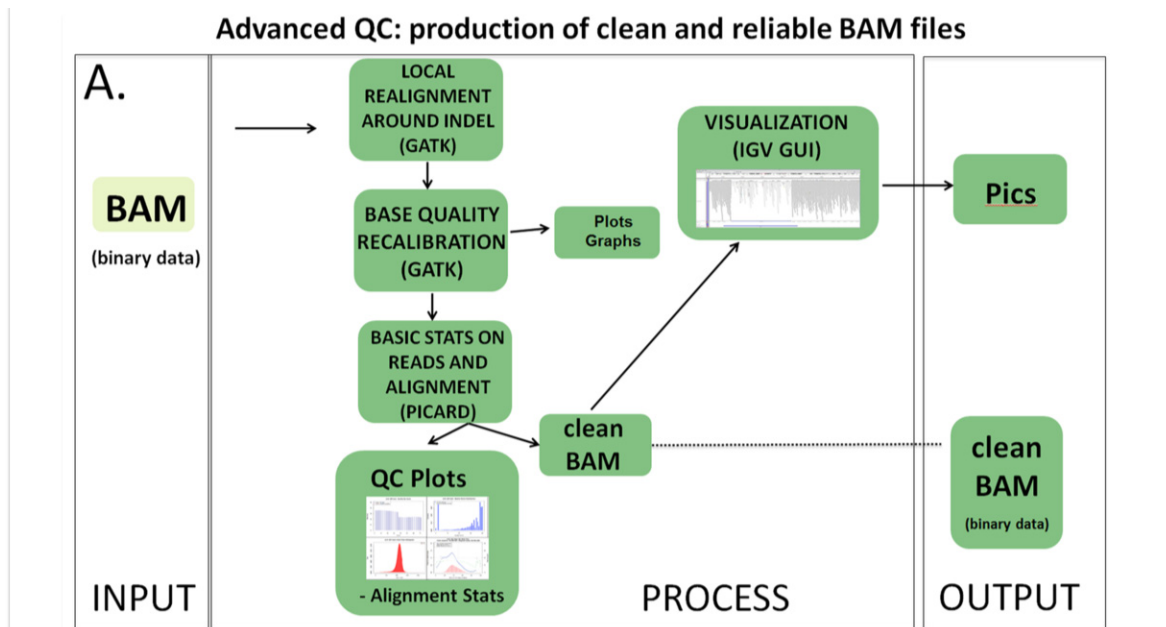
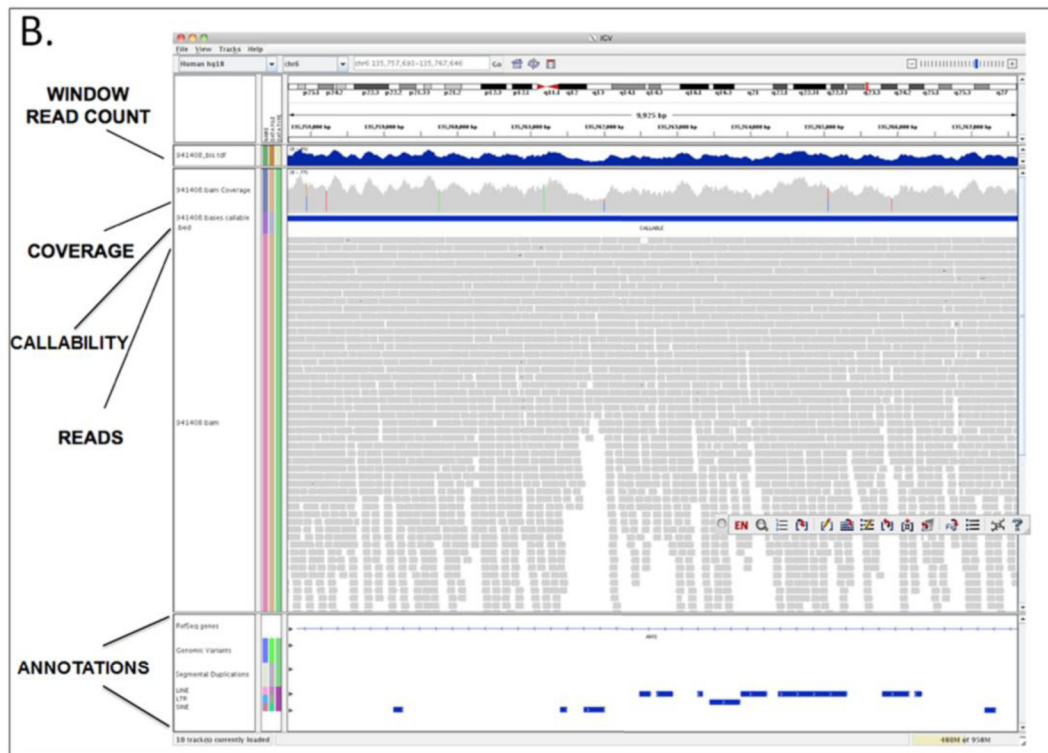


Figure 6. Cont.



2.4. Variant Calling and Annotation

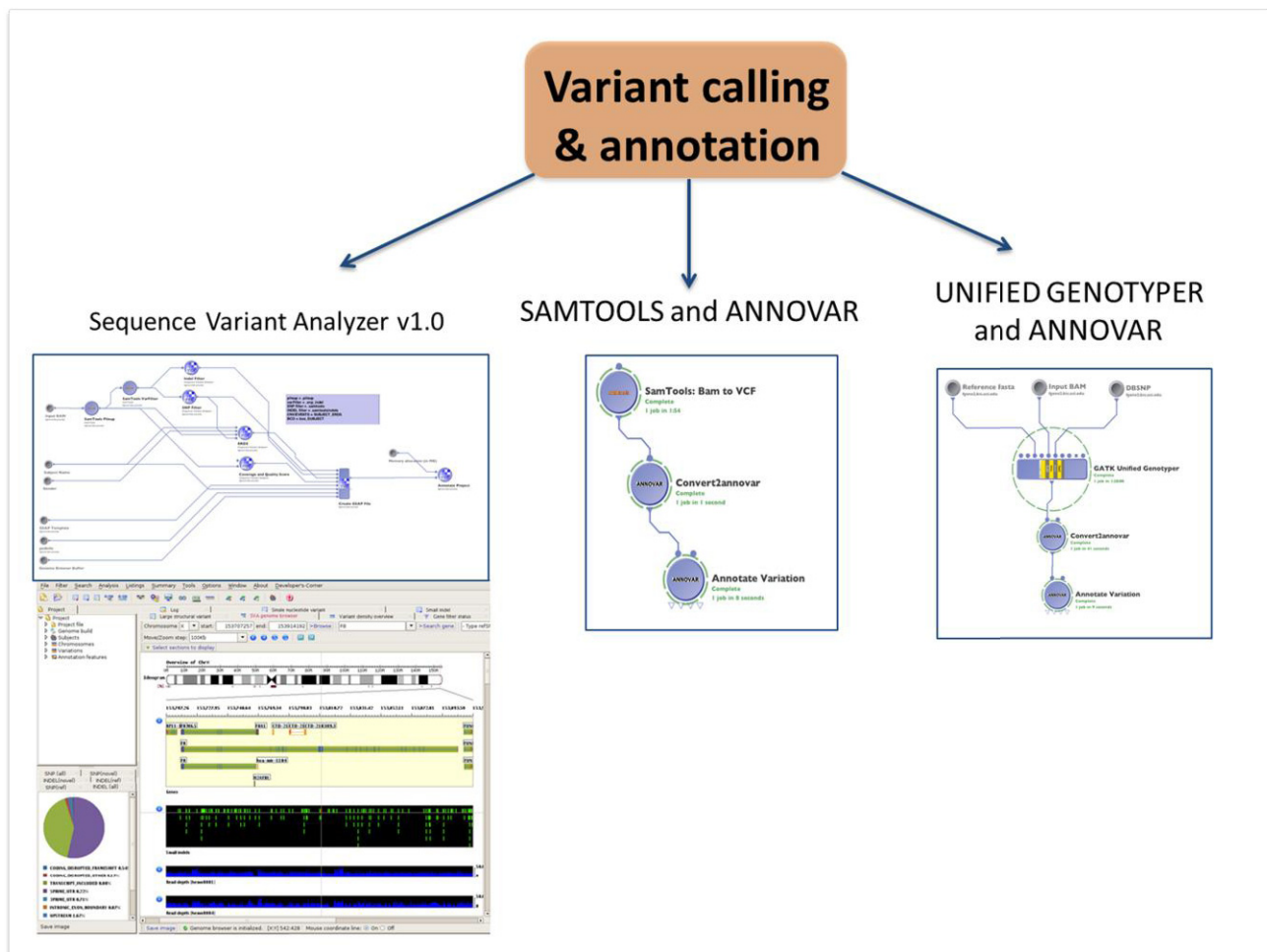
2.4.1. SNPs and Indels Calling and Annotation

We explored and embedded in workflow at least three different frameworks to call SNPs and Indels from whole genome alignment data and produce a comprehensive mutation/functional analysis report (Figure 7).

For the Sequence Variant Analyzer (SVA, <http://www.svaproject.org> [51]) v1.0 workflow, after SNPs and Indels have been called with SAMTOOLS [13] (<http://samtools.sourceforge.net/>) and CNVs have been called with “Estimation by Read Depth with Single Nucleotide Variants” (ERDS) software v1.02 (<http://web.duke.edu/~mz34/erds.htm>), they undergo annotation and visualization through SVA (Figure 7). SVA is a visualization platform for performing statistical analysis and filtering procedures as well. The workflow we have developed allows users to produce a .gsap file, which can be loaded into SVA to create a project with single or multiple annotated genomes. This version of SVA is linked to ENSEMBL hg18 annotations.

Variants detected with SAMTOOLS [13] can be also exported as a VCF file (Table S1) and comprehensively annotated through ANNOVAR [50] (Figure 7). This software allows functional annotation of genetic variants detected from diverse genomes (human genome hg18, hg19, as well as mouse, worm, fly, yeast and many others). In particular the last release of ANNOVAR retrieves variant calls and frequency information from the 1000 Genomes Project [81] or from the sixty genomes released by Complete Genomics [30]. ANNOVAR offers three different annotation options: *gene-based*, *region-based* or *filter-based annotation*, and all the three options are implemented.

Figure 7. A snapshot of the three independent workflow for variant calling, and annotation workflows available. Sequence Variant Analyzer (SVA) displays a graphical user interface (GUI) to visualize, annotate, filter and analyze the called variants.



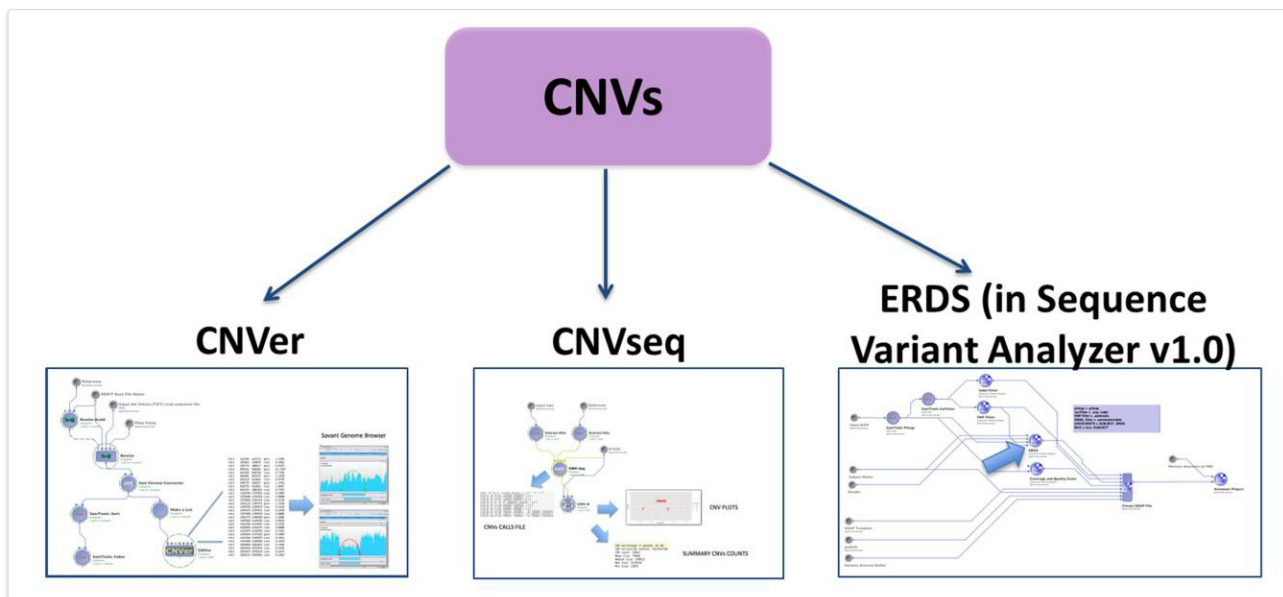
In the GPCG we also embedded the UnifiedGenotyperV2 from the GATK suite, which is a popular software to simultaneously call SNPs and Indels and produce a VCF output file [13] (Figure 7). We connected this module with ANNOVAR to perform the complete annotation of variants, as previously described.

2.4.2. CNVs Calling

While general agreement on the best analytical strategy is still lacking [2], we have implemented some workflows, as a first wave of a more comprehensive set of tools. Among the many different approaches to call CNVs, we have chosen the three approaches described in Figure 8.

CNVer relies both on read depth and read pair information [48] in a computational framework called the donor graph, that reduces the sequencing biases causing uneven local coverage (Figure 8). The most interesting feature of CNVer is the ability to compute the absolute copy counts of segments of the donor genome, and work with low coverage datasets. Moreover CNVer allows detecting CNVs without the need of a reference genome. The CNVs called by CNVer may be imported and visualized in the SAVANT genome browser [82] (Supplementary Materials S1, Figure A18).

Figure 8. A snapshot of the general organization of the CNVs modules. ERDS, CNVer and CNVseq have been implemented as a first wave of tools to call CNVs in DNA-Seq data.



CNVseq is a read depth method that detects CNVs with a robust statistical model conceptually derived from the aCGH (array Comparative genomic hybridization) analytical framework. [47]. CNVseq uses a sequence as a template and two sets of reads, one set from a reference individual (e.g., the individual expected to show a normal ploidy) and one from the test individual we want to screen for CNVs. The two sets of reads are aligned to a template genome, and then with a sliding window approach, CNVs are detected by computing the number of reads for each individual in each sliding window, yielding ratios and copy number estimations. Additional steps with this approach are performed in R (<http://www.r-project.org/>), with the possibility to get a full visualization of the \log_2 allelic ratio profiles and statistics on the detected CNVs (Supplementary Materials S1, Figure A20).

The final CNV workflow method is ERDS, which is a Hidden Markov Model (HMM) based approach that relies on read depth to infer the copy number state. It represents an extension of the method described in Bentley *et al.* (2008) [31], and described more in detail by Pelak *et al.* (2010) [45]. Expected read depth is calculated using the expectation maximization (EM) approach and corrected by GC bias. The ERDS functionality can be found embedded into the already described SVA 1.0 workflow for SNPs and Indels calling (Section 2.4.1) can then be used to visually inspect and annotate the CNVs (Supplementary Materials S1, Figure A21).

2.5. Evaluation with Simulated Data

We evaluated all the previously described GPCG workflows (Table 2 and Supplementary Materials S2) with 30 million simulated reads (both SE and PE) generated with dwgsim simulation tool, a publically available utility for simulating whole-genome Illumina reads [83]. Runtimes and information about the performances of the workflow with simulated data are summarized in Supplementary Table S2. We ran all the workflows with default parameters.

We reported for all the modules a better time performance of the GPCG compared to Galaxy, together with the absence for GPCG of the time required for the data upload.

Using the same simulated data files, we tested the common modules embedded both in our GPCG and on the Galaxy webserver interface. Due to the structure of Galaxy and the reduced number of processes available, we couldn't compare its performances on a workflow scale. The module shared by GPCG and Galaxy were: conversion of solexa into sanger format, BWA and Bowtie paired end alignment, the PICARD utilities to fix the mate information, mark the duplicates, collect the alignment, GC bias, and insert size metrics. We reported the GPCG and Galaxy time performances in Table 4.

Table 4. Runtimes and performances on simulated data for modules in common across Graphical Pipeline for Computational Genomics (GPCG) and Galaxy. The performances of GPCG in terms of run time were better than Galaxy for all the tested modules.

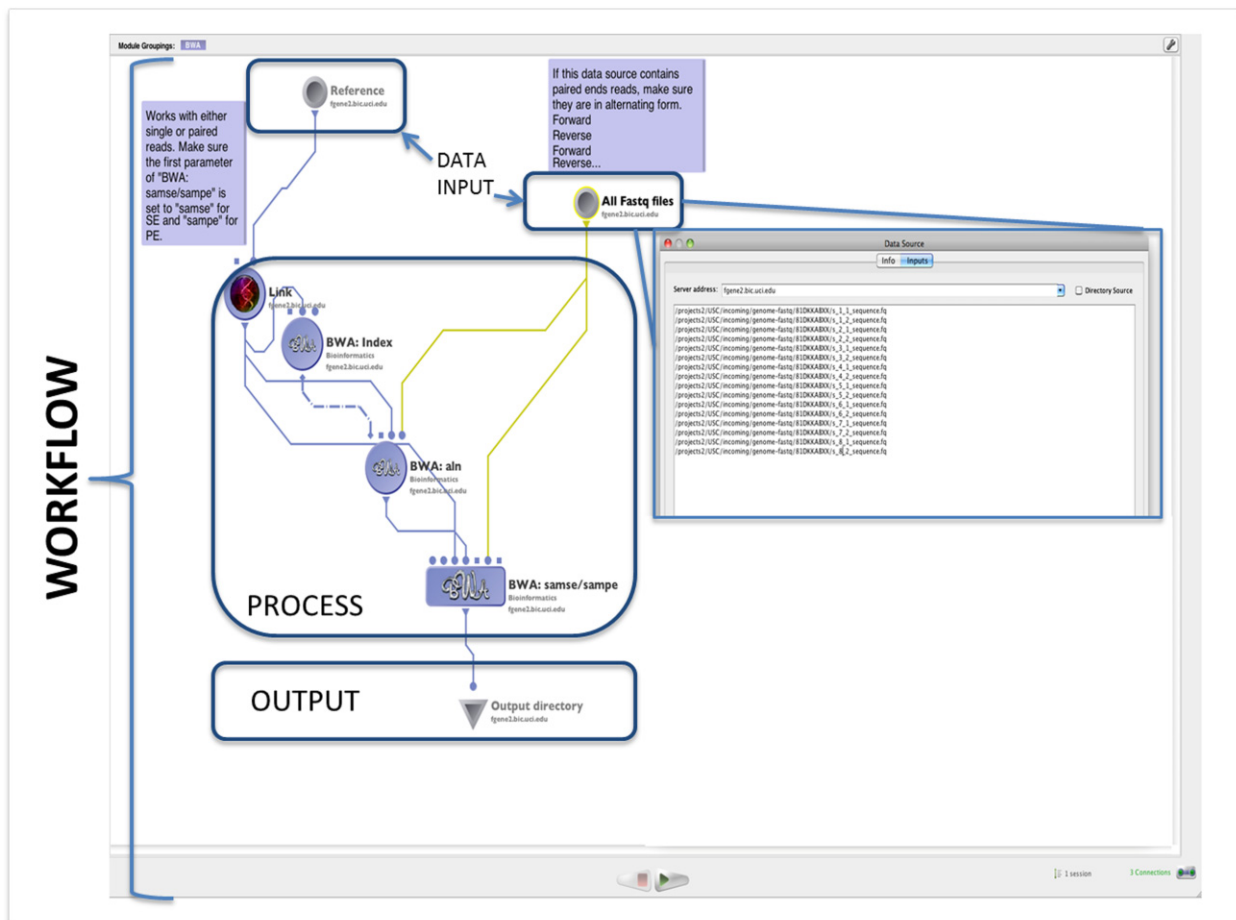
Analytical category	Input file (file size)	Job description	GPCG workflow name	Time	Galaxy module name	Time
Data upload	2.4 Gb × 2 (PE)	Upload of the data into the webserver	(N/A)	(N/A)	Upload of the data	180 min
Preprocessing	2.4 Gb fastq file	Conversion of solexa into sanger format	Preprocessing pipeline: sol2sanger	6 min	FASTQ Groomer	45 min
Alignment	2.4 Gb × 2 fastq files (PE)	BWA paired end alignment with default parameters	BWA PE (1.1)	132 min	Map with BWA for Illumina	240 min
	2.4 G × 2 fastq files (PE)	Bowtie paired end alignment with default parameters	BOWTIE PE (1.1)	205 min	Map with Bowtie for Illumina	270 min
Quality control (metrics and cleaning)	1.6 Gb SAM file	Synchronization of mate-pair information	Fix Mate Information (Basic QC, 1.3)	6 min	Paired Read Mate Fixer for paired data	30 min
	1.6 Gb SAM file	Marks duplicate reads	Mark Duplicates (Basic QC, 1.3)	2 min	Marks duplicate reads	20 min
	1.6 Gb SAM file	Reports the alignment metric of a SAM/BAM file	Collect Alignment Summary Metrics (Advanced QC, 1.4)	2 min	SAM/BAM Alignment Summary Metrics	6 min
	1.6 Gb SAM file	Reports the SAM/BAM GCbias metrics	Collect GC Bias Metrics (Advanced QC, 1.4)	3 min	SAM/BAM GC Bias Metrics	7 min
	1.6 Gb SAM file	Reports the insert size metrics	Collect Insert Size Metrics (Advanced QC, 1.4)	2 min	Insertion size metrics for PAIRED data	6 min

All of the performances of the GPCG platform were better compared to the run time of Galaxy. Furthermore, a bottleneck in the timing of analysis is the upload of to the Galaxy webserver, together with the not predictable waiting in queue time for the processes (Table 4). As the status of the process cannot be checked during the execution, the user must wait the end of the run of a process to input the results in to the next step without any hint about the elapsed time, and without the possibility to automatically keep track of the run times.

2.6. Evaluation with Real Data

We also evaluated the performance of our GPCG workflows in a “real life scenario”. Supplementary Table S2 reports the performances of the whole-genome alignment with BWA of an entire Illumina flowcell with an average of 130 million Illumina PE reads per lane. The average time required for the alignment of one lane was 6 hours, and for the whole flowcell was roughly 48 hours using GPCG. The output of the alignment was one SAM alignment file for each lane. In all large scale NGS data analysis it is important to manage the flow of both data and input/output files as they traverse complex workflows. The GPCG provides a simple way to set up the input file section (see magnified data source panel on the right, Figure 9), manage the naming of output files from individual modules using the transformation tool within a module definition (Figure 9).

Figure 9. Snapshot from the module we used to run the alignment of an entire flowcell with BWA-PE. This workflow includes the indexing of the reference genome (BWA: Index), the alignment of the two reads separately (BWA:aln) and their final combination (BWA: samse/sampe). The sixteen input files (*i.e.*, one forward and one reverse read for each one of the eight lanes of the flowcell) are shown in the data source panel magnified on the right. The pipeline allows managing all the options of the BWA alignment software through the module’s GUI without worrying about complex command lines.



3. Discussion

The GPCG was developed with a flexible graphical interface for efficient biomedical computing and distributed informatics research [65], and is intended to satisfy the needs of geneticists and computational scientists who are interested in whole genome, exome and targeted sequencing. The graphical workflows we have developed save researchers the time needed to implement and test the command lines of individual analytical steps, logically group complex operations into re-usable units, and allow these units to be aggregated into larger analytical workflows (e.g., QC following alignment).

The GPCG offers several advantages if compared to the available workflow environments for DNA-Seq data analysis. The GPCG includes a set of pipelines ready-to-go with modules logically interconnected between each other on the basis of the current analytical trends, while others offer a set of single modular routines that the user cannot connect in workflows. Moreover the LONI environment doesn't display any restrictive upper space limits on storage and on the available per-process RAM, and even more importantly doesn't require any data upload, thus eliminating bottlenecks with data staging to/from the servers. The limit of 100 GB of storage and 64 GB of RAM in Galaxy, together with the unpredictably long data upload time, don't fit well with the needs of users analyzing whole genome sequencing data, as the overall size of the original forward and reverse fastq files for a single whole 30× genome sequenced in paired end is already ~100 GB. We tested the single modules embedded both in Galaxy and GPCG with the same input files and analytical parameters (Table 4). The performance of the GPCG in term of run time was better compared to Galaxy for all the modules, with the additional time saved in GPCG (~3 hours) as no data upload on the system is required. The user of the GPCG has also the ability to disconnect and reconnect to running workflows, and to monitor at any time the progress and to check the status of a previously initiated process, with a detailed and interactive report of time performances and job execution info (e.g., output and error streams).

Since GPCG is an open source, the user can access the current version of the GPCG pipeline online at <http://pipeline.loni.ucla.edu>. The entire newly developed computational-genomics infrastructure LONI pipeline includes the pipeline server (<http://pipeline.loni.ucla.edu/DPS>), the pipeline web-start server (<http://pipeline.loni.ucla.edu/PWS>), the genomics workflows (<http://pipeline.loni.ucla.edu/services/library-navigator/>), collaborative wiki documentation for these protocols (<http://ucla.in/pbMgUm>), and community support (<http://informatics.googlecode.com/>). All these resources have been developed and are currently supported via an open and collaborative infrastructure. Constructive utilization of diverse tools and computational expertise may be shared as pipeline workflows between professionals, novice users and trainees [65,70].

In this first release of the GPCG we embedded only some popular tools for managing and analyzing DNA-Seq data from the initial raw reads to variant calling and annotation. However, to overcome this limitation, we are regularly testing and adding tools to be shared in the future releases of the genomics pipeline. Also the users can integrate new processes and implement new workflows promoting a community-based protocol validation and openly share and disseminate knowledge, tools and resources.

4. Methods

4.1. The LONI Environment and Workflow Creation

To translate DNA-Seq data analysis into a graphical pipeline solution within the LONI environment [65] we initially described all global processes in the *protocol design* or *skeletonization* step, using a top-down approach. We outlined the general classes of sequence data analysis, then the appropriate sub-classes of analyses, specific tools, test-data, invocation of concrete tools, and a detailed example of executable syntax for each step (<http://pipeline.loni.ucla.edu/support/user-guide/building-a-workflow>, [65]). The previously mentioned analytical steps are constructed from a series of command line executable processes, referred to as *modules* or *nodes*, which are connected to each other to form a visual workflow analysis protocol. All the logically concatenated *modules/nodes* involved in the same analytical step (e.g., alignment) comprise a workflow.

After all necessary modules that make up a workflow are independently defined and validated through the LONI pipeline GUI interface (<http://pipeline.loni.ucla.edu/support/user-guide/creating-modules/>), they are integrated into a coherent workflow.

4.2. Accessibility of the GPCG

The user can access the current version of the GPCG online at <http://pipeline.loni.ucla.edu>. The local Client can be set up by the users following the instructions reported at <http://pipeline.loni.ucla.edu/support/user-guide/installation/>. The GPCG can also be deployed on a server base downloading the distributed pipeline server installer (DPS) (<http://pipeline.loni.ucla.edu/support/server-guide/installation/>), and following the instructions available at <http://pipeline.loni.ucla.edu/DPS>). The GPCG workflows can also be directly launched via the pipeline web-start server (PWS) (<http://pipeline.loni.ucla.edu/PWS>). To search across the entire set of workflows the user can rely on an interactive graphical navigator (<http://pipeline.loni.ucla.edu/services/library-navigator/>), which enables not only the discovery, but also web-based utilization of this new computational-genomics infrastructure through the web, Figure 10.

4.3. Evaluation of the GPCG Workflows with Simulated and Real Data

Table 1 presents the software we have embedded in the workflows released with GPCG. We evaluated the workflows with both real and simulated data (Supplementary Table S2).

Evaluation with simulated data We used *dwgsim* [84], a utility for whole-genome Illumina reads simulation, contained in DNAA v0.1.2 (<http://sourceforge.net/projects/dnaa/>), to generate Illumina-like short sequences, using the default empirical error model illustrated on DNAA's Whole-Genome Simulation web-site (<http://dnaa.sf.net>). In total we generated 30 million reads with 100 bp length, using the complete human genome (hg18) as a reference and with default parameters. We developed also a module that allows the users to generate simulation datasets according to their needs (see Table 2 and Supplementary Materials S2).

Evaluation with real data To further compare the behavior of our workflows on real applications, we used an entire flow cell (8 lanes) with an average of 130 million Illumina PE reads per lane with

length of 100 bp (fastq produced with the Illumina Pipeline v1.8) to be aligned by BWA-PE against the whole human genome sequences (assembly: NCBI36.1/hg18).

Figure 10. The LONI pipeline computational library Navigator allows the interactive traversal, inspection, downloading and utilization of specific NGS analyses. Nested insert images illustrate the most common steps of search, selection, comparison, modification and execution of available end-to-end computational genomics workflows.

The screenshot displays the Pipeline Server Library Navigator web interface. The main view shows a large circular network diagram of workflows. A red dashed line highlights a path from a workflow in the network to a detailed view on the right. The detailed view shows 'Workflow Detail Information' for 'Integrated Bioinformatics (MAQ, SAMTools, Bowtie)', including a description and a workflow diagram. A green dashed line highlights a specific step in the workflow diagram, which is further detailed in a bottom-right inset showing the 'MAQ Mapper' module definition with its parameters and description.

These experiments were produced on an Illumina HiSeq200 DNA sequencer using v5 kits.

All the testing runs were performed on a Linux server Dual Intel Xeon X5650 2.67 Ghz 6 core processors 96 GB's of RAM 2 × 10 k RPM 150 GB WD VelociRaptor RAID 0 RHEL 5 64-bit.

5. Conclusions

The availability of workflows to manage and analyze NGS data in a straightforward way may play a role in triggering genomic advances in human health related translational research in psychiatry genetics. Currently, NGS technology is emerging as a fundamental basis on which to understand disease complexity and heterogeneity both for common and rare diseases, with benefits to clinical diagnostics and care once research findings are translated into clinical tests. Sequencing clinical

subjects is becoming a method of choice in translational studies of diseases, and genetic defects underlying several genetic disorders have been identified through whole exome or whole genome sequencing [85,86]. Nevertheless, an understanding of how genome variability leads to disease pathogenesis is still far from complete for the vast majority of genetic diseases [56], at least as to the meaning of many variants present in the genome of healthy individuals. The computational challenge for DNA-Seq data analysis is often a bottleneck, as many different tools are constantly emerging, and often requiring bioinformatics skills. We are proposing the GPCG as a handy and helpful graphical analysis platform to improve the efficiency of high-throughput data analysis in diverse applications of DNA-Seq analysis projects. There are two tiers of validations for the proposed pipeline GPCG pipeline infrastructure. The first tier is validation of the technical protocols (as we compared the results of the GPCG pipeline protocol against Galaxy). This evaluation confirms the programmatic reliability and reproducibility of the results using identical computational libraries. However, the second tier of validation is more important as it provides scientific evidence of the value added by the new GPCG pipeline infrastructure. This scientific validation includes for example the comparison of results obtained with different algorithms in term of read mapping, variant calling and quality control procedures to find the process that best fits the data, together with improvements in speed and high-throughput volume processing. For example, the GPCG framework may be used to construct a new genomics computing protocol that explicitly utilizes specific sequence analysis tools. The entire pipeline protocol may be shared with other users who can easily plug-in new data and/or swap alternative modules for analogous processing steps (employing different software tools). Such multi-investigator experimental studies would provide cues about how to select appropriate software tools and how different libraries compare in processing different type data sets (e.g., varying read length, fragments or paired ends, highly repetitive genome, *etc.*)

Acknowledgments

This work was funded in part by the National Institutes of Health through Grants U54 RR021813, 9P41EB015922-15, 2-P41-RR-013642-15, R01 MH71940, U24-RR025736, U24-RR021992, U24-RR021760, U24-RR026057 and NSF grants 0716055 and 1023115. We are also indebted to the members of the Laboratory of Neuro Imaging (LONI), the Biomedical Informatics Research Network (BIRN), and Clinical and Translational Science Award (CTSA) investigators, NIH Program officials, and many general users for their patience with beta-testing the Pipeline and for providing useful feedback about its state, functionality and usability. Benjamin Berman and Zack Ramjan from the USC Epigenome Center, University of Southern California, provided help with definition of sequence pre-processing. The authors also thank Bernard Chang for assistance with system administration of the UCI Pipeline servers.

Conflicts of Interest

The authors declare no conflict of interest.

References

1. Dalca, A.V.; Brudno, M. Genome variation discovery with high-throughput sequencing data. *Brief. Bioinform.* **2010**, *11*, 3–14.
2. Alkan, C.; Coe, B.P.; Eichler, E.E. Genome structural variation discovery and genotyping. *Nat. Rev. Genet.* **2011**, *12*, 363–376.
3. Flicek, P.; Birney, E. Sense from sequence reads: Methods for alignment and assembly. *Nat. Methods* **2009**, *6*, S6–S12.
4. Pepke, S.; Wold, B.; Mortazavi, A. Computation for chip-seq and rna-seq studies. *Nat. Methods* **2009**, *6*, S22–S32.
5. Meaburn, E.; Schulz, R. Next generation sequencing in epigenetics: Insights and challenges. *Semin. Cell Dev. Biol.* **2011**, *23*, 192–199.
6. Walsh, T.; McClellan, J.M.; McCarthy, S.E.; Addington, A.M.; Pierce, S.B.; Cooper, G.M.; Nord, A.S.; Kusenda, M.; Malhotra, D.; Bhandari, A.; *et al.* Rare structural variants disrupt multiple genes in neurodevelopmental pathways in schizophrenia. *Science* **2008**, *320*, 539–543.
7. Rumble, S.M.; Lacroute, P.; Dalca, A.V.; Fiume, M.; Sidow, A.; Brudno, M. Shrimp: Accurate mapping of short color-space reads. *PLoS Comput. Biol.* **2009**, *5*, e1000386.
8. Lin, H.; Zhang, Z.; Zhang, M.Q.; Ma, B.; Li, M. Zoom! Zillions of oligos mapped. *Bioinformatics* **2008**, *24*, 2431–2437.
9. Li, R.; Yu, C.; Li, Y.; Lam, T.W.; Yiu, S.M.; Kristiansen, K.; Wang, J. Soap2: An improved ultrafast tool for short read alignment. *Bioinformatics* **2009**, *25*, 1966–1967.
10. Chen, Y.; Souaiaia, T.; Chen, T. Perm: Efficient mapping of short sequencing reads with periodic full sensitive spaced seeds. *Bioinformatics* **2009**, *25*, 2514–2521.
11. Langmead, B.; Trapnell, C.; Pop, M.; Salzberg, S.L. Ultrafast and memory-efficient alignment of short DNA sequences to the human genome. *Genome Biol.* **2009**, *10*, R25.
12. Li, H.; Durbin, R. Fast and accurate short read alignment with burrows-wheeler transform. *Bioinformatics* **2009**, *25*, 1754–1760.
13. Chen, K.; Wallis, J.W.; McLellan, M.D.; Larson, D.E.; Kalicki, J.M.; Pohl, C.S.; McGrath, S.D.; Wendl, M.C.; Zhang, Q.; Locke, D.P.; *et al.* Breakdancer: An algorithm for high-resolution mapping of genomic structural variation. *Nat. Methods* **2009**, *6*, 677–681.
14. Olson, S.A. Emboss opens up sequence analysis. European molecular biology open software suite. *Brief. Bioinform.* **2002**, *3*, 87–91.
15. Myers, E.W. Toward simplifying and accurately formulating fragment assembly. *J. Comput. Biol.* **1995**, *2*, 275–290.
16. Pevzner, P.A.; Tang, H.; Waterman, M.S. An eulerian path approach to DNA fragment assembly. *Proc. Natl. Acad. Sci. USA* **2001**, *98*, 9748–9753.
17. Myers, E.W.; Sutton, G.G.; Delcher, A.L.; Dew, I.M.; Fasulo, D.P.; Flanigan, M.J.; Kravitz, S.A.; Mobarry, C.M.; Reinert, K.H.; Remington, K.A.; *et al.* A whole-genome assembly of drosophila. *Science* **2000**, *287*, 2196–2204.
18. Jaffe, D.B.; Butler, J.; Gnerre, S.; Mauceli, E.; Lindblad-Toh, K.; Mesirov, J.P.; Zody, M.C.; Lander, E.S. Whole-genome sequence assembly for mammalian genomes: Arachne 2. *Genome Res.* **2003**, *13*, 91–96.

19. Zerbino, D.R.; Birney, E. Velvet: Algorithms for *de novo* short read assembly using de bruijn graphs. *Genome Res.* **2008**, *18*, 821–829.
20. Li, R.; Zhu, H.; Ruan, J.; Qian, W.; Fang, X.; Shi, Z.; Li, Y.; Li, S.; Shan, G.; Kristiansen, K.; Yang, H.; Wang, J. *De novo* assembly of human genomes with massively parallel short read sequencing. *Genome Res.* **2010**, *20*, 265–272.
21. Simpson, J.T.; Wong, K.; Jackman, S.D.; Schein, J.E.; Jones, S.J.; Birol, I. Abyss: A parallel assembler for short read sequence data. *Genome Res.* **2009**, *19*, 1117–1123.
22. Miller, J.R.; Koren, S.; Sutton, G. Assembly algorithms for next-generation sequencing data. *Genomics* **2010**, *95*, 315–327.
23. Ewing, B.; Green, P. Base-calling of automated sequencer traces using phred. Ii. Error probabilities. *Genome Res.* **1998**, *8*, 186–194.
24. Brockman, W.; Alvarez, P.; Young, S.; Garber, M.; Giannoukos, G.; Lee, W.L.; Russ, C.; Lander, E.S.; Nusbaum, C.; Jaffe, D.B. Quality scores and snp detection in sequencing-by-synthesis systems. *Genome Res.* **2008**, *18*, 763–770.
25. Li, M.; Nordborg, M.; Li, L.M. Adjust quality scores from alignment and improve sequencing accuracy. *Nucleic Acids Res.* **2004**, *32*, 5183–5191.
26. Li, R.; Li, Y.; Fang, X.; Yang, H.; Wang, J.; Kristiansen, K. Snp detection for massively parallel whole-genome resequencing. *Genome Res.* **2009**, *19*, 1124–1132.
27. DePristo, M.A.; Banks, E.; Poplin, R.; Garimella, K.V.; Maguire, J.R.; Hartl, C.; Philippakis, A.A.; del Angel, G.; Rivas, M.A.; Hanna, M.; *et al.* A framework for variation discovery and genotyping using next-generation DNA sequencing data. *Nat. Genet.* **2011**, *43*, 491–498.
28. Ning, Z.; Cox, A.J.; Mullikin, J.C. Ssaha: A fast search method for large DNA databases. *Genome Res.* **2001**, *11*, 1725–1729.
29. Martin, M.V.; Rollins, B.; Sequeira, P.A.; Mesen, A.; Byerley, W.; Stein, R.; Moon, E.A.; Akil, H.; Jones, E.G.; Watson, S.J.; *et al.* Exon expression in lymphoblastoid cell lines from subjects with schizophrenia before and after glucose deprivation. *BMC Med. Genomics* **2009**, *2*, 62.
30. Drmanac, R.; Sparks, A.B.; Callow, M.J.; Halpern, A.L.; Burns, N.L.; Kermani, B.G.; Carnevali, P.; Nazarenko, I.; Nilsen, G.B.; Yeung, G.; *et al.* Human genome sequencing using unchained base reads on self-assembling DNA nanoarrays. *Science* **2010**, *327*, 78–81.
31. Bentley, D.R.; Balasubramanian, S.; Swerdlow, H.P.; Smith, G.P.; Milton, J.; Brown, C.G.; Hall, K.P.; Evers, D.J.; Barnes, C.L.; Bignell, H.R.; *et al.* Accurate whole human genome sequencing using reversible terminator chemistry. *Nature* **2008**, *456*, 53–59.
32. Koboldt, D.C.; Chen, K.; Wylie, T.; Larson, D.E.; McLellan, M.D.; Mardis, E.R.; Weinstock, G.M.; Wilson, R.K.; Ding, L. Varscan: Variant detection in massively parallel sequencing of individual and pooled samples. *Bioinformatics* **2009**, *25*, 2283–2285.
33. Wheeler, D.A.; Srinivasan, M.; Egholm, M.; Shen, Y.; Chen, L.; McGuire, A.; He, W.; Chen, Y.J.; Makhijani, V.; Roth, G.T.; *et al.* The complete genome of an individual by massively parallel DNA sequencing. *Nature* **2008**, *452*, 872–876.
34. Mokry, M.; Feitsma, H.; Nijman, I.J.; de Bruijn, E.; van der Zaag, P.J.; Guryev, V.; Cuppen, E. Accurate snp and mutation detection by targeted custom microarray-based genomic enrichment of short-fragment sequencing libraries. *Nucleic Acids Res.* **2010**, *38*, e116.

35. Shen, Y.; Wan, Z.; Coarfa, C.; Drabek, R.; Chen, L.; Ostrowski, E.A.; Liu, Y.; Weinstock, G.M.; Wheeler, D.A.; Gibbs, R.A.; *et al.* A snp discovery method to assess variant allele probability from next-generation resequencing data. *Genome Res.* **2010**, *20*, 273–280.
36. Hoberman, R.; Dias, J.; Ge, B.; Harmsen, E.; Mayhew, M.; Verlaan, D.J.; Kwan, T.; Dewar, K.; Blanchette, M.; Pastinen, T. A probabilistic approach for snp discovery in high-throughput human resequencing data. *Genome Res.* **2009**, *19*, 1542–1552.
37. Malhis, N.; Jones, S.J. High quality snp calling using illumina data at shallow coverage. *Bioinformatics* **2010**, *26*, 1029–1035.
38. Handsaker, R.E.; Korn, J.M.; Nemes, J.; McCarroll, S.A. Discovery and genotyping of genome structural polymorphism by sequencing on a population scale. *Nat. Genet.* **2011**, *43*, 269–276.
39. Kim, J.; Sinha, S. Indelign: A probabilistic framework for annotation of insertions and deletions in a multiple alignment. *Bioinformatics* **2007**, *23*, 289–297.
40. Li, H.; Ruan, J.; Durbin, R. Mapping short DNA sequencing reads and calling variants using mapping quality scores. *Genome Res.* **2008**, *18*, 1851–1858.
41. Hormozdiari, F.; Alkan, C.; Eichler, E.E.; Sahinalp, S.C. Combinatorial algorithms for structural variation detection in high-throughput sequenced genomes. *Genome Res.* **2009**, *19*, 1270–1278.
42. Lee, S.; Hormozdiari, F.; Alkan, C.; Brudno, M. Modil: Detecting small indels from clone-end sequencing with mixtures of distributions. *Nat. Methods* **2009**, *6*, 473–474.
43. Korbel, J.O.; Abyzov, A.; Mu, X.J.; Carriero, N.; Cayting, P.; Zhang, Z.; Snyder, M.; Gerstein, M.B. Pomer: A computational framework with simulation-based error models for inferring genomic structural variants from massive paired-end sequencing data. *Genome Biol.* **2009**, *10*, R23.
44. Sindi, S.; Helman, E.; Bashir, A.; Raphael, B.J. A geometric approach for classification and comparison of structural variants. *Bioinformatics* **2009**, *25*, i222–i230.
45. Pelak, K.; Shianna, K.V.; Ge, D.; Maia, J.M.; Zhu, M.; Smith, J.P.; Cirulli, E.T.; Fellay, J.; Dickson, S.P.; Gumbs, C.E.; *et al.* The characterization of twenty sequenced human genomes. *PLoS Genet.* **2010**, *6*, e1001111.
46. Ye, K.; Schulz, M.H.; Long, Q.; Apweiler, R.; Ning, Z. Pindel: A pattern growth approach to detect break points of large deletions and medium sized insertions from paired-end short reads. *Bioinformatics* **2009**, *25*, 2865–2871.
47. Xie, C.; Tammi, M.T. Cnv-seq, a new method to detect copy number variation using high-throughput sequencing. *BMC Bioinformatics* **2009**, *10*, 80.
48. Medvedev, P.; Fiume, M.; Dzamba, M.; Smith, T.; Brudno, M. Detecting copy number variation with mated short reads. *Genome Res.* **2010**, *20*, 1613–1622.
49. Nielsen, R.; Paul, J.S.; Albrechtsen, A.; Song, Y.S. Genotype and snp calling from next-generation sequencing data. *Nat. Rev. Genet.* **2011**, *12*, 443–451.
50. Wang, K.; Li, M.; Hakonarson, H. Annovar: Functional annotation of genetic variants from high-throughput sequencing data. *Nucleic Acids Res.* **2010**, *38*, e164.
51. Ge, D.; Ruzzo, E.K.; Shianna, K.V.; He, M.; Pelak, K.; Heinzen, E.L.; Need, A.C.; Cirulli, E.T.; Maia, J.M.; Dickson, S.P.; *et al.* Sva: Software for annotating and visualizing sequenced human genomes. *Bioinformatics* **2011**, *27*, 1998–2000.

52. Neale, B.M.; Rivas, M.A.; Voight, B.F.; Altshuler, D.; Devlin, B.; Orho-Melander, M.; Kathiresan, S.; Purcell, S.M.; Roeder, K.; Daly, M.J. Testing for an unusual distribution of rare variants. *PLoS Genet.* **2011**, *7*, e1001322.
53. Adzhubei, I.A.; Schmidt, S.; Peshkin, L.; Ramensky, V.E.; Gerasimova, A.; Bork, P.; Kondrashov, A.S.; Sunyaev, S.R. A method and server for predicting damaging missense mutations. *Nat. Methods* **2010**, *7*, 248–249.
54. Yandell, M.; Huff, C.; Hu, H.; Singleton, M.; Moore, B.; Xing, J.; Jorde, L.B.; Reese, M.G. A probabilistic disease-gene finder for personal genomes. *Genome Res.* **2011**, *21*, 1529–1542.
55. Zhang, J.; Chiodini, R.; Badr, A.; Zhang, G. The impact of next-generation sequencing on genomics. *J. Genet. Genomics* **2011**, *38*, 95–109.
56. Torkamani, A.; Scott-Van Zeeland, A.A.; Topol, E.J.; Schork, N.J. Annotating individual human genomes. *Genomics* **2011**, *98*, 233–241.
57. Mardis, E.R. The \$1,000 genome, the \$100,000 analysis? *Genome Med.* **2010**, *2*, 84.
58. Milano, F. Power System Architecture. In *Power System Modelling and Scripting*; Milano, F., Ed.; Springer: Berlin/Heidelberg, Germany, 2010; pp. 19–30.
59. Wang, D.; Zender, C.; Jenks, S. Efficient clustered server-side data analysis workflows using swamp. *Earth Sci. Inform.* **2009**, *2*, 141–155.
60. Ye, X.-Q.; Wang, G.-H.; Huang, G.-J.; Bian, X.-W.; Qian, G.-S.; Yu, S.-C. Heterogeneity of mitochondrial membrane potential: A novel tool to isolate and identify cancer stem cells from a tumor mass? *Stem Cell Rev. Rep.* **2011**, *7*, 153–160.
61. Yoo, J.; Ha, I.; Chang, G.; Jung, K.; Park, K.; Kim, Y. Cnvas: Copy number variation analysis system—The analysis tool for genomic alteration with a powerful visualization module. *BioChip J.* **2011**, *5*, 265–270.
62. Chard, K.; Onyuksel, C.; Wei, T.; Sulakhe, D.; Madduri, R.; Foster, I. Build Grid Enabled Scientific Workflows Using Gravi and Taverna. In *Proceedings of IEEE Fourth International Conference on eScience, 2008. eScience '08*, Indianapolis, IN, USA 7–12 December 2008; pp. 614–619.
63. Ludäscher, B.; Altintas, I.; Berkley, C.; Higgins, D.; Jaeger, E.; Jones, M.; Lee, E.A.; Tao, J.; Zhao, Y. Scientific workflow management and the kepler system. *Concurr. Comput. Pract. Exp.* **2006**, *18*, 1039–1065.
64. Goecks, J.; Nekrutenko, A.; Taylor, J.; Team, T.G. Galaxy: A comprehensive approach for supporting accessible, reproducible, and transparent computational research in the life sciences. *Genome Biol.* **2010**, *11*, R86.
65. Dinov, I.; Torri, F.; Macciardi, F.; Petrosyan, P.; Liu, Z.; Zamanyan, A.; Eggert, P.; Pierce, J.; Genco, A.; Knowles, J.; *et al.* Applications of the pipeline environment for visual informatics and genomics computations. *BMC Bioinformatics* **2011**, *12*, 304.
66. Taylor, I.; Shields, M.; Wang, I.; Harrison, A. The Triana Workflow Environment: Architecture and Applications. In *Workflows for e-Science*; Taylor, I., Deelman, E., Gannon, D., Shields, M., Eds.; Springer: Secaucus, NJ, USA, 2007; pp. 320–339.
67. Kwon, Y.; Shigemoto, Y.; Kuwana, Y.; Sugawara, H. Web API for biology with a workflow navigation system. *Nucleic Acids Res.* **2009**, *37*, W11–W16.

68. Oinn, T.; Addis, M.; Ferris, J.; Marvin, D.; Senger, M.; Greenwood, M.; Carver, T.; Glover, K.; Pocock, M.; Wipat, A.; *et al.* Taverna: A tool for the composition and enactment of bioinformatics workflows. *Bioinformatics* **2004**, *20*, 3045–3054.
69. Schatz, M. The missing graphical user interface for genomics. *Genome Biol.* **2010**, *11*, 128.
70. Dinov, I.; Lozev, K.; Petrosyan, P.; Liu, Z.; Eggert, P.; Pierce, J.; Zamanyan, A.; Chakrapani, S.; van Horn, J.; Parker, D.; *et al.* Neuroimaging study designs, computational analyses and data provenance using the loni pipeline. *PLoS One* **2010**, *5*, e13070.
71. Rex, D.E.; Ma, J.Q.; Toga, A.W. The loni pipeline processing environment. *Neuroimage* **2003**, *19*, 1033–1048.
72. Service, R.F. Gene sequencing. The race for the \$1000 genome. *Science* **2006**, *311*, 1544–1546.
73. Mardis, E.R. Next-generation DNA sequencing methods. *Annu. Rev. Genomics Hum. Genet.* **2008**, *9*, 387–402.
74. Mardis, E.R. The impact of next-generation sequencing technology on genetics. *Trends Genet.* **2008**, *24*, 133–141.
75. Li, H.; Handsaker, B.; Wysoker, A.; Fennell, T.; Ruan, J.; Homer, N.; Marth, G.; Abecasis, G.; Durbin, R.; Subgroup, G.P.D.P. The sequence alignment/map format and samtools. *Bioinformatics* **2009**, *25*, 2078–2079.
76. Leung, K.; Parker, D.S.; Cunha, A.; Dinov, I.D.; Toga, A.W. Irma: An image registration meta-algorithm—Evaluating alternative algorithms with multiple metrics. *Lect. Notes Comput. Sci.* **2008**, *5069*, 612–617.
77. Leung, K.T.K. Principal Ranking Meta-Algorithms. Ph.D. these, University of California, Los Angeles, CA, USA, 2011.
78. Rex, D.E.; Shattuck, D.W.; Woods, R.P.; Narr, K.L.; Luders, E.; Rehm, K.; Stolzner, S.E.; Rottenberg, D.E.; Toga, A.W. A meta-algorithm for brain extraction in mri. *NeuroImage* **2004**, *23*, 625–637.
79. Ruffalo, M.; LaFramboise, T.; Koyutürk, M. Comparative analysis of algorithms for next-generation sequencing read alignment. *Bioinformatics* **2011**, *27*, 2790–2796.
80. McKenna, A.; Hanna, M.; Banks, E.; Sivachenko, A.; Cibulskis, K.; Kernytsky, A.; Garimella, K.; Altshuler, D.; Gabriel, S.; Daly, M.; *et al.* The genome analysis toolkit: A mapreduce framework for analyzing next-generation DNA sequencing data. *Genome Res.* **2010**, *20*, 1297–1303.
81. Gunter, C. Genomics: A picture worth 1000 genomes. *Nat. Rev. Genet.* **2010**, *11*, 814.
82. Fiume, M.; Williams, V.; Brook, A.; Brudno, M. Savant: Genome browser for high-throughput sequencing data. *Bioinformatics* **2010**, *26*, 1938–1944.
83. Hamada, M.; Wijaya, E.; Frith, M.C.; Asai, K. Probabilistic alignments with quality scores: An application to short-read mapping toward accurate snp/indel detection. *Bioinformatics* **2011**, doi:10.1093/bioinformatics/btr537.
84. Hamada, M.; Wijaya, E.; Frith, M.C.; Asai, K. Probabilistic alignments with quality scores: An application to short-read mapping toward accurate snp/indel detection. *Bioinformatics* **2011**, *27*, 3085–3092.
85. Raffan, E.; Semple, R.K. Next generation sequencing—Implications for clinical practice. *Br. Med. Bull.* **2011**, *99*, 53–71.

86. Haas, J.; Katus, H.A.; Meder, B. Next-generation sequencing entering the clinical arena. *Mol. Cell. Probes* **2011**, *25*, 206–211.

© 2012 by the authors; licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution license (<http://creativecommons.org/licenses/by/3.0/>).