MOLECULAR ECOLOGY
RESOURCES
Molecular Ecology Resources (2015) **15**, 99–106 doi: 10.1111/1755-0998.12286

Pipeliner: software to evaluate the performance of bioinformatics pipelines for next-generation resequencing

B. NEVADO*† and M. PEREZ-ENCISO*†‡
*Centre for Research in Agricultural Genomics (CRAG), CSIC-IRTA-UAB-UB, 08193 Bellaterra, Spain, †Universitat Autònoma de Barcelona, 08193 Bellaterra, Spain, ‡Institut Català de Recerca I Estudis Avancats (ICREA), 08010 Barcelona, Spain

Abstract

The choice of technology and bioinformatics approach is critical in obtaining accurate and reliable information from next-generation sequencing (NGS) experiments. An increasing number of software and methodological guidelines are being published, but deciding upon which approach and experimental design to use can depend on the particularities of the species and on the aims of the study. This leaves researchers unable to produce informed decisions on these central questions. To address these issues, we developed PIPELINER – a tool to evaluate, by simulation, the performance of NGS pipelines in resequencing studies. PIPELINER provides a graphical interface allowing the users to write and test their own bioinformatics pipelines with publicly available or custom software. It computes a number of statistics summarizing the performance in SNP calling, including the recovery, sensitivity and false discovery rate for heterozygous and homozygous SNP genotypes. PIPELINER can be used to answer many practical questions, for example, for a limited amount of NGS effort, how many more reliable SNPs can be detected by doubling coverage and halving sample size or what is the false discovery rate provided by different SNP calling algorithms and options. PIPELINER thus allows researchers to carefully plan their study's sampling design and compare the suitability of alternative bioinformatics approaches for their specific study systems. PIPELINER is written in C++ and is freely available from http://github.com/brunonevado/Pipeliner.

Keywords: bioinformatics pipelines, experimental design, individual resequencing, next-generation sequencing, simulation

Received 15 January 2014; revision received 19 May 2014; accepted 23 May 2014

Introduction

The increasing popularity of next-generation sequencing (NGS) experiments, due to dwindling prices, has led to the proliferation of software and methodological guidelines for the analysis of NGS data sets (e.g., Metzker 2010; Nielsen *et al.* 2011). However, the fast pace with which new software is available, together with the diversity of sequencing techniques, means that the impact of experimental design and bioinformatics analysis on inferences based on NGS data is often unknown or difficult to evaluate. Furthermore, newly released software is not always extensively documented, nor is a benchmarking of newly available and previously published software always extensively and coherently performed.

A number of recent studies have addressed this issue by comparing the performance of alternative mapping software (e.g., Li & Homer 2010; Ruffalo *et al.* 2011; Fonseca *et al.* 2012) or SNP calling tools (e.g., Cheng *et al.* 2014). Likewise, the effect of alternative experimental designs and bioinformatics pipelines upon population genetics inferences has received increasing attention (Crawford & Lazzaro 2012; Fumagalli 2013; Han *et al.* 2014; Nevado *et al.* 2014). While providing relevant information on the different tools available, the plethora of available software, together with the pace with which new tools are developed, means the result of such effort needs to be constantly updated to avoid becoming obsolete. Often, researchers simply follow default procedures or options in the software, but these may not be optimized for their particular design or interests. Moreover, after the analysis, it is difficult to ascertain basic parameters such as likely false discovery rate or an estimate of the percentage of sites segregating in the population that may have been discovered.

The choice of experimental design and bioinformatics pipeline can also depend on the study system being investigated. For instance, the genetic distance to the reference genome can affect both mapping (Pool *et al.* 2010) and SNP calling steps (Nevado *et al.* 2014). Likewise, the

Correspondence: Bruno Nevado,
E-mail: brunonevado@gmail.com

complexity of the reference genome, i.e. the amount of repeats, gene duplications or inversions should be taken into account when filtering potential SNPs (Derrien *et al*. 2012; Lee & Schatz 2012). The choice of optimal experimental design further depends on the specific aims of the study. For instance, detecting rarer variants or estimating the demographic history of populations will usually require higher read depth than detecting common variants or estimating *F*st between populations (e.g. Crawford & Lazzaro 2012; Fumagalli 2013; Meynert *et al*. 2013).

These questions are difficult to address with available tools, as they are usually study specific. This means researchers are left with general ideas regarding the performance of alternative sampling designs and bioinformatics tools, but not detailed information on their suitability for their specific research questions and study system. Alternatively, researchers might perform a comparative analysis of alternative experimental designs and software tools for their own projects. This, however, entails a significant time investment and requires stronger bioinformatics skills compared to using available software.

In this work, we introduce PIPELINER, a program to evaluate numerically the performance of different bioinformatics pipelines, focusing on the recovery and sensitivity in SNP calling from resequencing data. A number of realistic input options and any combination of software tools can be compared in PIPELINER, such that researchers can find the best pipeline tailored to their study. Importantly, PIPELINER is flexible enough to accommodate new software tools that may be released in the future. It provides therefore a straightforward environment in which to evaluate new software and methodological guidelines.

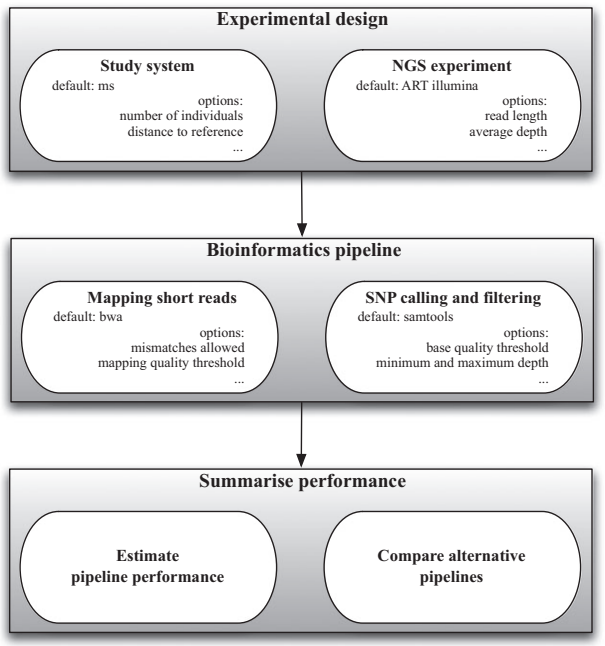

Fig. 1 Outline of the analysis performed in PIPELINER, showing the default software and options for each step.

Materials and methods

Overview of PIPELINER

PIPELINER consists of two separate executables: a graphical user interface (GUI) used for setting up the analysis parameters and plotting results and a command-line tool used during the analysis to convert file formats and summarize the performance of the pipeline used.

PIPELINER replicates an entire NGS experiment and consists of the following steps (Fig. 1): experimental design (sequence and NGS reads simulation), bioinformatics analysis (read mapping and variant calling) and evaluation of the performance of the pipeline defined. To perform the first two steps, PIPELINER calls internally different software packages: MS (Hudson 2002), ART (Huang *et al*. 2012), BWA (Li & Durbin 2009) and SAMTOOLS (Li *et al*. 2009). Other software packages can be used for any part of the analysis so long as standard file formats are employed (fasta, fastq, vcf, bam). More specifically, these steps are:

Step 1: Experimental design. (1a) Simulation of the sequence data to be used in the analysis. In this step, the user can define both the sample size of the study (how many individuals to sequence), the evolutionary history of the population under study and the genetic distance to the reference genome. File formats accepted are ms and fasta, and the default software is MS. The forward simulator SFS_CODE (Hernandez 2008) can also be used to simulate the original sequence data under more complex evolutionary scenarios, or any software able to produce a fasta or ms-like file. This flexibility allows the user to closely mimic a given study system, for example, by defining the population size and mutation rate, the selective forces operating on the genome or the demographic history of the population. The data simulated in this step must include the individuals to be sequenced and the reference genome to be used in the next steps of the analysis. When using ms format input, the user must also define an ancestral sequence upon which the SNP data will be applied. This can be a random sequence (which can be created by PIPELINER at the start of the analysis) or a sequence in fasta format provided by the user. In the latter case, the ancestral sequence can contain typical features of real genomes – such as repetitive elements – to assess the effect of these features on the SNP calling process.

(1b) Simulation of the NGS short reads using the sequence data simulated in the previous step. Here, the user can define the sequencing process itself, including the sequencing technology (Illumina or 454), depth and read length. This step takes as input the fasta files generated in the previous step and creates fastq file(s) with the raw NGS reads. Default software called is ART.

Step 2: Bioinformatics analysis. (2a) Mapping of NGS short reads to the reference genome. The user can define which software and options to use to perform the mapping of the short reads obtained in the previous step. Default software used is BWA, input format is fastq, and output format is bam.

(2b) SNP calling and filtering. From the aligned short reads files (bam format), the user defines how to obtain a list of confident variant calls (vcf format). Default software used is SAMTOOLS, BCFTOOLS and VCFUTILS (all these programs are included in SAMTOOLS distribution).

Step 3: Analysis of the results of the NGS pipeline and comparison with the true sequence data to calculate the following statistics:.

1 *Recovery*: percentage of original genotypes that are correctly identified. Genotypes missed due to low read depth or quality reduce recovery.
2 *Sensitivity*: percentage of callable genotypes – i.e., present in sites that pass the filters used – that are correctly identified. Genotypes that are missed due to low depth or quality do not reduce sensitivity.
3 *False Discovery Rate*: percentage of genotype calls performed that are incorrect.

These statistics are averaged across all individuals analysed and are reported separately for invariable genotypes (i.e., the individual is homozygous for the reference allele), homozygous SNP genotypes (the individual is homozygous for the alternative allele) and heterozygous SNP genotypes (the individual carries one reference and one alternative allele). For homozygous and heterozygous SNP genotypes, results are also reported as a function of the absolute frequency of the alternative allele in the sample.

PIPELINER also reports the frequency of different outcomes, for example, the percentage of heterozygous SNP genotypes that are incorrectly identified as homozygous for the reference allele or homozygous for the alternative allele or the percentage of invariable positions identified as variable. Detailed information regarding incorrect outcomes can optionally include the number and type of bases covering the site and their base quality, which can be used to pinpoint and correct specific errors incurred in the pipeline defined.

Finally, PIPELINER can use the program MSTATSPOP (available from http://bioinformatics.cragenomica.es/numgenomics/people/sebas/software/software.html) to estimate population genetics summary statistics. This software implements the algorithms described in Ferretti *et al.* (2012) and provides unbiased estimates even in the presence of high levels of missing data between individuals. Population genetics estimates can be obtained using both the simulated sequence data sets (obtained in step 1a above) and the data sets obtained after NGS analysis (step 2b), thus providing expectations regarding the effect of experimental designs and bioinformatics pipelines on population genetics summary statistics.

PIPELINER's GUI

PIPELINER uses only standard file formats fasta and vcf to summarize the performance of alternative pipelines, thus allowing the use of any software package(s) to simulate the original sequence data and the NGS short reads, or to perform the bioinformatics analysis. However, as the simulation and bioinformatics analysis of NGS data require several steps executed in succession, we provide a graphical user interface (GUI), which conveniently splits the analysis into their component steps.

PIPELINER's GUI contains four main sections used to define the different steps of the analysis (Fig. 2). The 'Pipeline' tab is used to set up the analysis, i.e. to define the sampling design, study system and bioinformatics analysis. The 'Run settings' tab is used to define software paths, number of replicates and other run-time options. The 'Bash file' tab allows previewing, editing and saving the pipeline commands into a bash file. The 'Plots' tab is used for summarizing and plotting the results of different pipelines.

The 'Pipeline' tab is further subdivided into five sections (Fig. 2): the 'Input' tab, to define how to obtain the original DNA sequences for the analysis; the 'NGS' tab, to obtain the NGS short reads for each diploid individual; the 'Alignment' tab, to align the NGS short reads to the reference genome; the 'SNP calling' tab, to perform SNP calling and filtering; and the 'Statistics' tab, to choose which statistics to calculate when summarizing the pipeline.

A major challenge facing a tool like PIPELINER is how to maintain up to date with the fast pace with which new software packages are developed to analyse NGS data. To deal with this issue, for the steps 'Input', 'NGS', 'Alignment' and 'SNP calling', we implement two different interfaces: a 'default' interface using the

default software and a subset of available options and a 'user-defined' interface where the user can include command-line instructions to perform any of these steps. This approach means PIPELINER can use any newly developed or nondefault software for any or all of these steps, insofar as standard file formats are used, and a command-line interface is available to run the software. The standard formats for each step are depicted in Table 1: for input, fasta and ms-like formats; for NGS simulation, the input format is fasta, and the output format is fastq; for the alignment, the input format is fastq and the output format bam; and for the SNP calling, the input format is bam, and the output format is vcf. As an example of a 'user-defined' step, Fig. 3 shows how to use BOWTIE (Langmead *et al.* 2009) to perform the mapping of short reads from a single-end NGS experiment, using default settings.

Further details on how to use the PIPELINER GUI to set up the analysis, as well as examples using different software packages for the different steps, are given in the manual included in the distribution of PIPELINER, available from http://github.com/brunonevado/Pipeliner.

Applications

PIPELINER can be used to compare the performance of alternative experimental designs, bioinformatics pipelines and software packages. Below, we give practical examples of how to use PIPELINER to address some of these issues.

Alternative experimental designs for resequencing studies

When planning a resequencing study, a central question is how to allocate the financial resources available: is it better to aim for more individuals sampled at lower read depth, or fewer individuals sequenced at higher depth? This issue was addressed in recent studies (e.g. Buerkle & Gompert 2013; Fumagalli 2013; Nevado *et al.* 2014), and a common result is that sampling many individuals at low read depth provides the most precise estimates of population genetics statistics. However, the low read depth means genotypes are not known with certainty, and inferences are based directly on genotype likelihoods without performing SNP calling (e.g. Nielsen *et al.* 2012; Buerkle & Gompert 2013). Conversely, analyses that require individual genotypes – such as tests based on haplotype information or studies aimed to characterize admixture between populations – will generally benefit from higher read depth, such that genotypes can be inferred with confidence. With PIPELINER, the effect of read depth on recovery of individual SNP genotypes obtained by resequencing can be analysed by simulation, before allocation of resources for sequencing. To illustrate this, we performed a simulation with PIPELINER where we kept the total sequencing effort constant at 200× and investigated four experimental designs: sampling 50 individuals at 4×, 20 individuals at 10×, 10 individuals at 20× and 4 individuals at 50×.

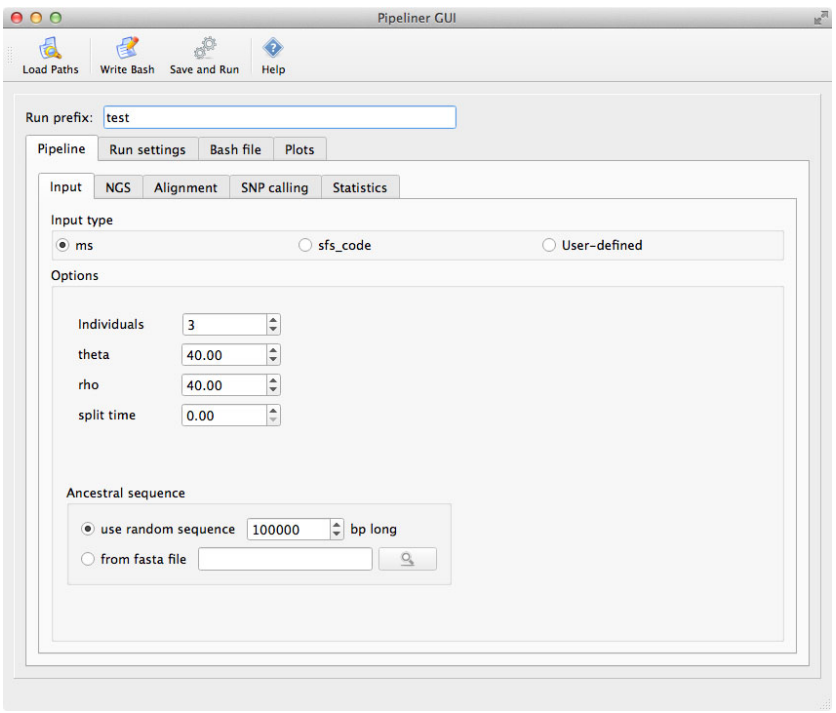

Fig. 2 Screenshot of PIPELINER's GUI showing how to define the input data using the default MS coalescent simulator.

Table 1 Default software and file formats used in each step of the analysis with PIPELINER

Step	Input format	Output format	Default software
Input	–	fasta/ms	MS, SFS_CODE
NGS simulation	fasta	fastq	ART
Alignment	fastq	bam	BWA
SNP calling	bam	vcf	SAMTOOLS

In a first step, we obtained the original SNP data by coalescent simulation using MS. We assumed a constant effective population size of 10 000 individuals, a neutral mutation rate of 10^{-8} per site per generation and simulated SNP data for a 100 kb genomic region (i.e. theta set to 40 for simulation with MS). We sampled a number of individuals according to the four experimental designs detailed above (50, 20, 10 and 4 individuals). We assumed the reference sequence comes from the same population (i.e. split time set to 0) and set as ancestral a random sequence 100 kb long. For the example with 10 individuals, the full ms command was *ms 21 1 –t 40 –r 40 100000*.

The second step in the analysis involved simulating the NGS short reads for each diploid individual, using the default software ART. ART uses empirically derived error and base-quality profiles calibrated in large empirical data sets, thus closely mimicking real

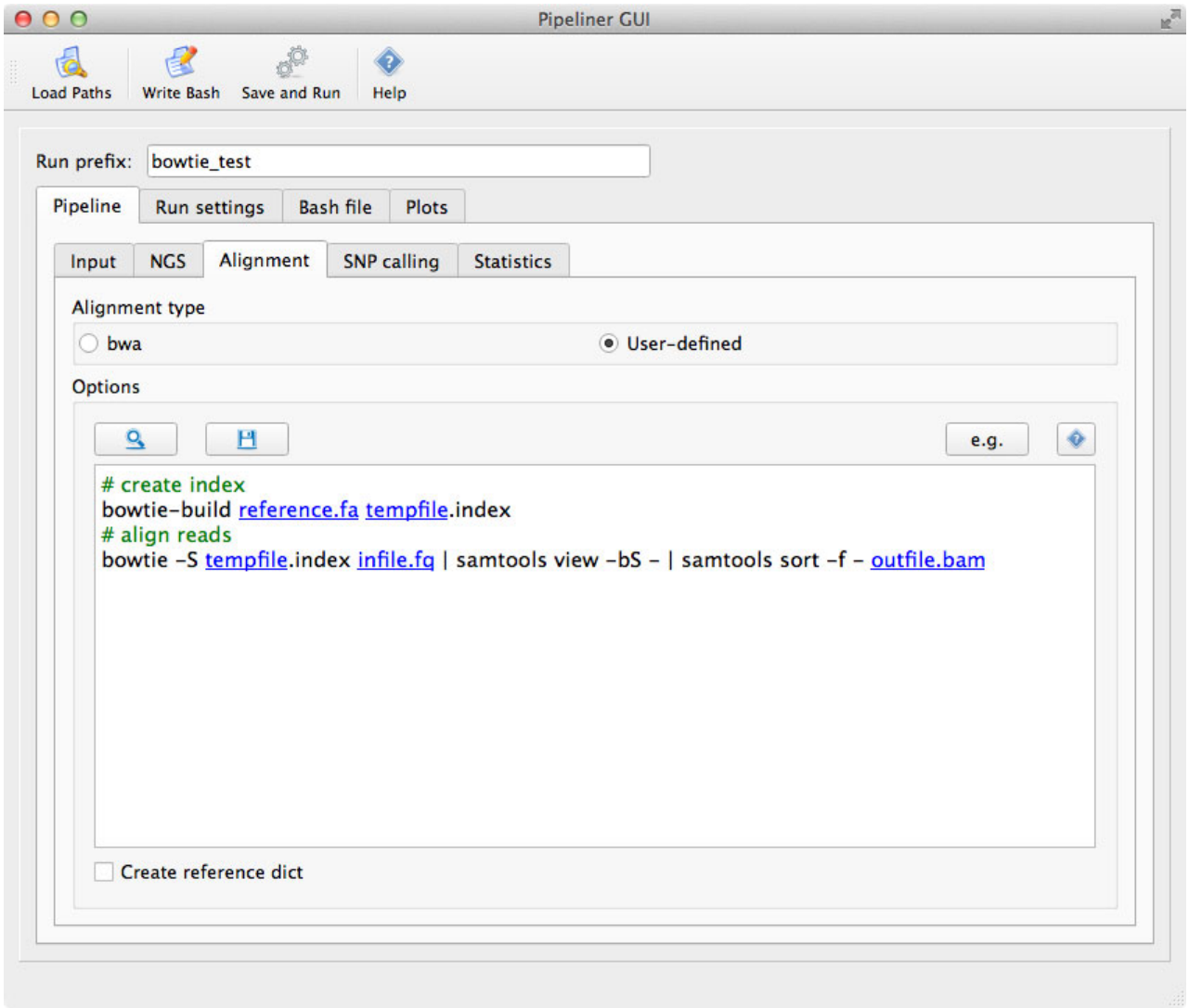

Fig. 3 Screenshot of PIPELINER' s GUI showing how to use nondefault software in PIPELINER. In this case, the mapping step is performed with BOWTIE instead of the default BWA. The underlined words are keywords expected by PIPELINER, representing input file (fastq format, *infile.fq*), output file (bam format, *outfile.bam*) and reference genome (fasta format, *reference.fa*).

NGS reads. We used the built-in profile for Illumina 75 bp, paired-ends sequencing, but different sequencing technologies and read lengths could be simulated. Average read depth per individual was set to 4×, 10×, 20× and 50× according to the four experimental designs detailed above.

Analysis of the NGS short reads was performed using the default software in PIPELINER: alignment of short reads to the reference genome using BWA (six mismatches allowed per read, minimum mapping quality of 20) and SNP calling and filtering performed with SAMTOOLS. For this example, we performed individual SNP calling with SAMTOOLS and applied the default filters of vcfutils.pl varFilter (part of SAMTOOLS package), except for depth: we used a minimum depth of half, and maximum depth of double, the expected read depth per individual.

Results of this analysis are shown in Fig. 4. For the study system investigated, and using the NGS experiment and bioinformatics analysis delineated above, increasing average read depth from 4× to 10× provided an increase in the proportion of recovered heterozygous genotypes from *c.* 20% to *c.* 80%. Increasing read depth past 10× returned smaller improvements, with 50× recovering close to 100% of all heterozygous genotypes sampled. On the other hand, the number of segregating sites identified (Fig. 4, bottom) was similar with 50 individuals sequenced at 4× or 20 individuals at 10×, but smaller when sampling fewer individuals at higher read depth. This shows that higher read depth results in a higher proportion of genotypes correctly identified, although decreasing the number of individuals results in fewer segregating sites in the sample. Depending on the aims of the study – particularly if the objective is to identify variable positions in the genome or to accurately recover the genotype information for each individual – these results could be used to infer the optimal experimental design for a given study.

Alternative bioinformatics pipelines: individual vs. multiple sample SNP calling

Multiple sample SNP calling is an option in standard SNP calling software like SAMTOOLS and GATK (DePristo *et al.* 2011), which substantially increases the performance of the SNP calling algorithm at low read depth (Nielsen *et al.* 2011). The approach used is to perform SNP calling for each site using data from all individuals sampled at that particular site. This option is expected to bias SNP calling algorithms towards medium-frequency variants (e.g. Nielsen *et al.* 2011; Han *et al.* 2014), and PIPELINER can be used to quantify this bias.

To investigate the effect of multiple sample SNP calling, we performed an analysis in PIPELINER as described in the previous example, but using average read depths of 2× and 4×, sampling five individuals and using both individual SNP calling and multiple sample SNP calling with SAMTOOLS (specified in the 'SNP calling' tab). Figure 5 shows the sensitivity (% of genotypes passing all quality filters that are correctly identified) in identifying heterozygous SNP genotypes as a function of the alternative allele count (AAC).

Overall, multiple sample SNP calling increased sensitivity as compared to individual SNP calling. However, low-frequency-derived variants became more elusive than medium- to high-frequency variants. This can lead to skewed allele frequency spectrum calculations (Han *et al.* 2014; Nevado *et al.* 2014) and illustrates how the influence of particular protocols or options can be readily investigated with PIPELINER.

Alternative software packages: SAMTOOLS vs. GATK

As a final application of PIPELINER, we evaluated the comparative performance of two widely used software packages for SNP calling: SAMTOOLS and GATK's *UnifiedGenotyper*.

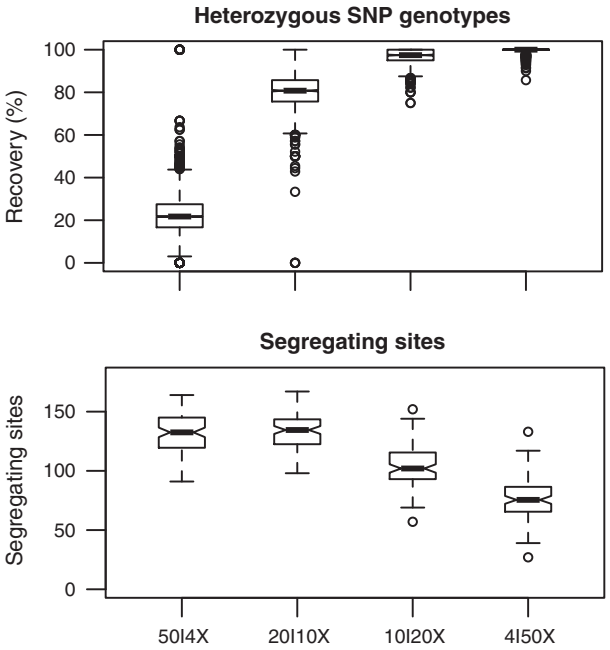

Fig. 4 Results of the analysis of alternative experimental designs with a fixed sequencing effort of 200×. Top panel shows the recovery of heterozygous SNP genotypes, that is, the percentage of original heterozygous SNP genotypes that are correctly identified after analysis of NGS reads, summarized across individuals. Bottom panel shows the number of segregating sites identified with each experimental design, summarized across replicates. Distributions shown were obtained with 100 replicates.

Simulation of original SNP data and NGS experiment was performed as described in the previous example, sampling five diploid individuals and using average read depths of 4, 8, 12, 16 and 20×. Alignment of the short reads was performed with BWA allowing for six mismatches and filtering by a minimum mapping quality threshold of 20. Individual SNP calling was performed with either SAMTOOLS – with the same settings used in previous examples – or GATK. For GATK, we used the option 'EMIT_ALL_CONFIDENT_SITES' to obtain a list of confident sites and filtered variant calls with the *VariantFiltration* tool following GATK' s best practices guide (http://www.broadinstitute.org/gatk/guide/best-practices). The settings used for SNP calling with GATK are provided as an example in PIPELINER' s distribution.

Figure 6 shows the mean sensitivity (top panel) and mean false discovery rate (FDR, bottom panel) for heterozygous genotypes, obtained with SAMTOOLS and GATK. SAMTOOLS generally exhibited higher sensitivity at the cost of higher FDR, particularly at read depths of 4× and 8×. Note that these results were obtained under a specific set of conditions and default options and that using, for example, a more stringent filtering could alter the results. Regardless, this result exemplifies how the performance of different software packages can be compared in a straightforward way using PIPELINER.

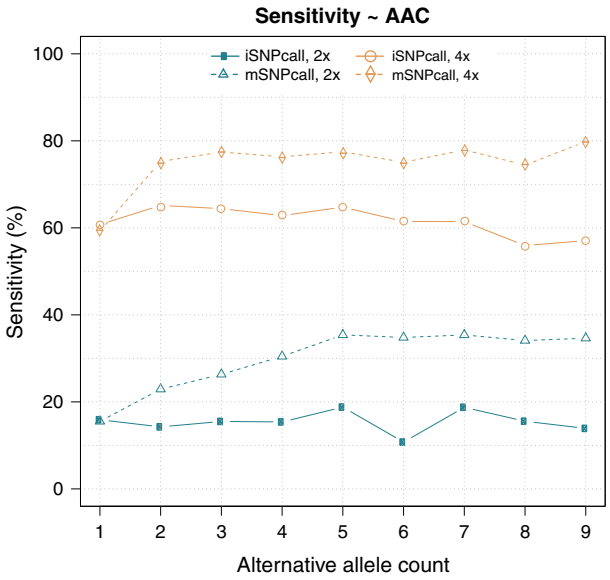

Fig. 5 The effect of multiple sample SNP calling at low read depth (2–4×). Shown is the sensitivity in identifying heterozygous SNP genotypes (% of heterozygous SNP genotypes passing all quality filters that are correctly identified) when using either individual or multiple sample SNP calling. Alternative allele count (*x*-axis) is the absolute frequency of the alternative allele in the sample. Results obtained with 100 replicates, each sampling five diploid individuals.

Limitations

PIPELINER is aimed exclusively at NGS studies using an existing reference genome – studies where NGS short reads are analysed without using a reference genome, or approaches that avoid the SNP calling process (e.g., Nielsen *et al.* 2012; Buerkle & Gompert 2013), cannot be simulated in the current version. PIPELINER considers only biallelic SNPs: sites containing more than two different alleles, or other types of variation such as indels or copy-number variants, are ignored in the analysis. Finally, the measures of performance calculated by PIPELINER are focused on the SNP calling results: the performance of the mapping step is not measured independently, but only insofar as it affects the SNP calling process.

Performance

The time and computational resources required in an analysis with PIPELINER vary depending on the software and options used – particularly during the mapping and SNP calling steps – as well as the sequence length, read depth and number of individuals simulated. As an

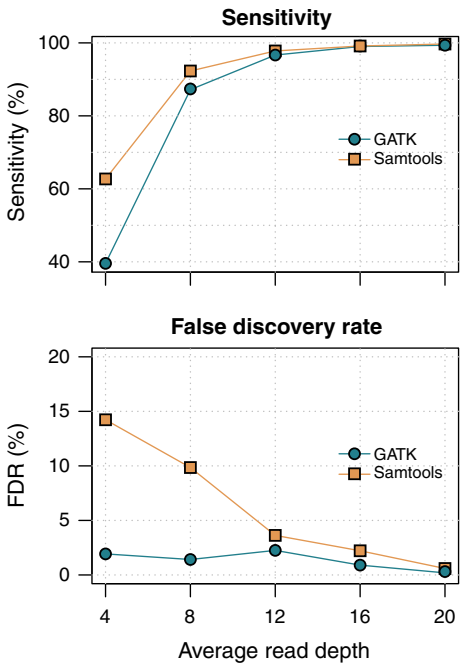

Fig. 6 Comparative performance of SAMTOOLS and GATK. Top panel shows the sensitivity (% of genotypes passing all quality filters that are correctly identified) and bottom panel the false discovery rate (% of incorrect genotype calls) for heterozygous SNP genotypes. At low read depth (2–4×), and using the default options, SAMTOOLS recovers more genotypes at the cost of more incorrect calls (i.e. higher sensitivity and higher false discovery rate). Results obtained with 100 replicates, each sampling five diploid individuals.

example, a 3.3-GHz Linux workstation took 7.5 min to run 10 replicates of 10 individuals, with 6× read depth and 100 kb sequence length. A similar analysis using a sequence length of 1 Mb took 50 min and with 10 Mb took close to 10 h. For the first example, all steps were performed with default software and options in PIPELINER, while for the latter two, the input step was performed with MACS (Chen *et al*. 2009) instead of MS.

Obtaining PIPELINER

PIPELINER is written in C++ and can be installed on Linux and Mac computers. For the command-line tool, a C++11 compiler is required, and for the GUI, the QT libraries must also be installed (www.qt-project.org). The source code, installation instructions and a detailed manual with examples can be obtained from http://github.com/brunonevado/Pipeliner.

Acknowledgements

The authors would like to thank S. Ramos-Onsins, W. Sanseverino and R. Tonda for helpful discussions, three anonymous reviewers for comments on an earlier version of this manuscript and E. Bianco for help testing the software.

Funding

This work was supported by Consolider CSD2007-00036 'Centre for Research in Agrigenomics' and AGL2010-14822 grants (Spain) to MPE.

References

Buerkle CA, Gompert Z (2013) Population genomics based on low coverage sequencing: how low should we go? *Molecular Ecology*, **22**, 3028–3035.

Chen GK, Marjoram P, Wall JD (2009) Fast and flexible simulation of DNA sequence data. *Genome Research*, **19**, 136–142.

Cheng AY, Teo YY, Ong RTH (2014) Assessing single nucleotide variant detection and genotype calling on whole-genome sequenced individuals. *Bioinformatics*. Advance access.

Crawford JE, Lazzaro BP (2012) Assessing the accuracy and power of population genetic inference from low-pass next-generation sequencing data. *Frontiers in Genetics*, **3**, 66.

DePristo MA, Banks E, Poplin R *et al.* (2011) A framework for variation discovery and genotyping using next-generation DNA sequencing data. *Nature Genetics*, **43**, 491–498.

Derrien T, Estellé J, Marco Sola S *et al.* (2012) Fast computation and applications of genome mappability. *PLoS One*, **7**, e30377.

Ferretti L, Raineri E, Ramos-Onsins S (2012) Neutrality tests for sequences with missing data. *Genetics*, **191**, 1397–1401.

Fonseca NA, Rung J, Brazma A, Marioni JC (2012) Tools for mapping high-throughput sequencing data. *Bioinformatics*, **28**, 3169–3177.

Fumagalli M (2013) Assessing the effect of sequencing depth and sample size in population genetics inferences. *PLoS One*, **8**, e79667.

Han E, Sinsheimer JS, Novembre J (2014) Characterizing bias in population genetic inferences from low-coverage sequencing data. *Molecular Biology and Evolution*, **31**, 723–35.

Hernandez RD (2008) A flexible forward simulator for populations subject to selection and demography. *Bioinformatics*, **24**, 2786–2787.

Huang W, Li L, Myers JR, Marth GT (2012) ART: a next-generation sequencing read simulator. *Bioinformatics*, **28**, 593–594.

Hudson RR (2002) Generating samples under a Wright-Fisher neutral model of genetic variation. *Bioinformatics*, **18**, 337–338.

Langmead B, Trapnell C, Pop M, Salzberg S (2009) Ultrafast and memory-efficient alignment of short DNA sequences to the human genome. *Genome Biology*, **10**, R25.

Lee H, Schatz MC (2012) Genomic dark matter: the reliability of short read mapping illustrated by the genome mappability score. *Bioinformatics*, **28**, 2097–105.

Li H, Durbin R (2009) Fast and accurate short read alignment with Burrows-Wheeler transform. *Bioinformatics*, **25**, 1754–1760.

Li H, Homer N (2010) A survey of sequence alignment algorithms for next-generation sequencing. *Briefings in Bioinformatics*, **11**, 473–483.

Li H, Handsaker B, Wysoker A *et al.* (2009) The sequence alignment/map (SAM) format and SAMtools. *Bioinformatics*, **25**, 2078–2079.

Metzker ML (2010) Sequencing technologies – the next generation. *Nature Reviews Genetics*, **11**, 31–46.

Meynert AM, Bicknell LS, Hurles ME, Jackson AP, Taylor MS (2013) Quantifying single nucleotide variant detection sensitivity in exome sequencing. *BMC Bioinformatics*, **14**, 195.

Nevado B, Ramos-Onsins SE, Perez-Enciso M (2014) Resequencing studies of nonmodel organisms using closely related reference genomes: optimal experimental designs and bioinformatics approaches for population genomics. *Molecular Ecology*, **23**, 1764–1779.

Nielsen R, Paul JS, Albrechtsen A, Song YS (2011) Genotype and SNP calling from next-generation sequencing data. *Nature Reviews Genetics*, **12**, 443–451.

Nielsen R, Korneliussen T, Albrechtsen A, Li Y, Wang J (2012) SNP calling, genotype calling, and sample allele frequency estimation from New-Generation Sequencing data. *PLoS One*, **7**, e37558.

Pool JE, Hellmann I, Jensen JD, Nielsen R (2010) Population genetic inference from genomic sequence variation. *Genome Research*, **20**, 291–300.

Ruffalo M, LaFramboise T, Koyuturk M (2011) Comparative analysis of algorithms for next-generation sequencing read alignment. *Bioinformatics*, **27**, 2790–2796.

B.N. and M.P.E. conceived the project and wrote the manuscript. B.N. coded the software.
