TECHNICAL NOTE

DiscoSnp++: detection of all kinds of SNPs and indels from raw unassembled read set(s)

Pierre Peterlongo*, Erwan Drezen, Claire Lemaitre and Chloé Riou

*Correspondence:
pierre.peterlongo@inria.fr
GenScale, INRIA Rennes Bretagne-Atlantique, IRISA, Campus de Beaulieu, Rennes, France
Full list of author information is available at the end of the article

Abstract

Next Generation Sequencing (NGS) data provide an unprecedented access to life mechanisms. In particular, these data enable to detect polymorphisms such as SNPs and indels. As these polymorphisms represent a fundamental source of information in agronomy, environment or medicine, their detection in NGS data is now a routine task. The main methods for their prediction usually need a reference genome. However, non-model organisms and highly divergent genomes such as in cancer studies are more and more investigated. The *DiscoSnp* tool has been successfully applied to predict isolated SNPs from raw read set(s) without the need of a reference genome. We propose *DiscoSnp++*, the successor of *DiscoSnp*, which benefits from a new software design that reduces time and memory consumption, and from a new algorithmic design that detects all kinds of SNP and small indels, adds genotype information and outputs a VCF file. Results show that *DiscoSnp++* performs better than state-of-the-art methods in terms of both quality and computational resources.

Keywords: SNP; Indel; reference-free

Introduction

Next Generation Sequencing (NGS) data provides an unprecedented access to life mechanisms. In particular, these data enable to assess genetic differences between chromosomes, individuals or species. Such polymorphisms represent a fundamental source of information in many aspects of biology with numerous applications in agronomy, environment or medicine.

Within the democratization of the sequencing provided by the NGS technologies, determining genetic differences as SNPs or indels has now become a routine task. There exist numerous software designed for predicting such polymorphisms. Mostly, these methods are based on the use of a reference genome by mapping sequenced reads as this is the case for GATK [1] or SamTools [2] or by mapping partial assemblies as for DISCOVAR [3] or FERMI [4] to cite a few. Basically, they first map the data to the reference and in a second phase they scan the reference genome to analyse for each locus the differences between the reference sequence and the mapped reads.

These methods are well accepted and extensively used. However, they present severe drawbacks. First they are highly sensitive to the mapping quality. Highly repeated regions of the reference genome are difficult to map with a high degree of confidence. Polymorphism detected from these repeated regions may be erroneous as the quantification of mapped reads is erroneous and as the differences between

occurrences of the repeats can be wrongly interpreted as polymorphism. Secondly, they suffer from the fact that they need a high quality reference genome. This evident and strong condition limits the application to reference species.

In practice, biologists are more and more working on species for which there exists no confident reference genome. Additionally, despite major improvements in the sequencing techniques this last decade, reconstructing a perfect and complete genome from reads remains a highly complex task [5]. In this context, there is an important need for *reference-free* methods detecting SNPs and indels, directly from NGS reads, without requiring an assembled reference sequence. An alternative method consists in first assemble the reads before to map them back to the so obtained reference, as this is the case in [6]. However, such methods cumulate both the assembly and the mapping difficulties. In this manuscript, we refer to such methods as the *hybrid* strategy.

A few methods [7, 8, 9, 10, 11] were proposed for *de-novo* detection of polymorphism. All these methods are based on the use of the *de Bruijn graph*, i.e. a directed graph where the set of vertices corresponds to the set of words of length k (k-mers) contained in the reads, and there is an edge between two k-mers if they overlap on $k-1$ nucleotides. In this data structure, polymorphisms generate recognizable patterns called the *bubbles*. These tools detect and analyze such bubbles in order to decipher their origin (sequencing errors, inexact repeats, real SNP or indel).

We recently proposed *DiscoSnp*, a *reference-free* method for detecting isolated SNPs [12]. The *DiscoSnp* approach outperforms other *reference-free* methods in terms of both computational needs and results quality. Its main features are **1**/ its extremely low memory usage (several billion reads may be analyzed with no more than 6 GB RAM memory), **2**/ its high execution speed, **3**/ its high precision and recall, **4**/ the precision of the score assigned to each predicted SNP, **5**/ the fact that it can be applied to any number of read sets (from 1 to n), and **6**/ the kind of SNPs it detects, called *isolated*. Isolated SNPs are SNPs that are distant to the left and to the right by at least k nucleotides from any other polymorphisms, with k being one of the main parameters of *de novo* SNP detection tool. Isolated SNPs have the advantage to be easily amplified by PCR. However isolated SNPs do not represent all SNPs. Although isolated SNPs are usually the most common, in some cases, such as with highly polymorphic genomes and/or with numerous distinct genomes compared simultaneously, only a fraction of SNPs are isolated and so detected by *DiscoSnp*.

In this paper, we present *DiscoSnp++* that is an extension of the *DiscoSnp* tool. The tool was re-implemented from scratch using the GATB library [13]. The detection of isolated SNPs remains exactly the same with *DiscoSnp++*, but with a much better running time. Additionally, *DiscoSnp++* detects new kinds of variants and it can output the predicted variants in the commonly-used VCF format.

Improvements in *DiscoSnp++*

The *DiscoSnp++* tool is based on the analysis of the *de Bruijn Graph* (dBG) [**?**] for predicting *isolated SNPs*. In a dBG, a *bubble* denotes a path in the graph which diverges into two distinct paths before to meet back. In the dBG of one or several read sets, a specific motif witnesses the presence of isolated SNPs. This motif is

a bubble whose both distinct paths are composed of exactly k nodes. Figure 1.a shows a toy example of such a bubble. The original *DiscoSnp* tool detects only such isolated SNPs.

Conversely to isolated SNPs, *close* SNPs are SNPs that are close to each other or to any source of polymorphism by less than k nucleotides. In the dBG, close SNPs generate a bubble in which the two distinct paths still contain the same number of nodes, bigger than k. Figure 1.b shows a toy example of such bubble. In *DiscoSnp++*, we extended the initial model detection in order to detect close SNPs in addition to isolated ones. The user may limit the maximal number of close SNPs that can be detected in a bubble.

As presented in Figure 1.c, indels also generate bubbles in the dBG. The two paths of such bubbles are of distinct lengths. The smaller one is of size at most $k-1$ nodes. *DiscoSnp++* detects isolated indels in the dBG. The user limits the maximal size of the predicted indels.

DiscoSnp++ works as follows: **(1)** build the dBG of the input dataset(s); **(2)** detect motifs witnessing the presence of SNPs (close or not) and indels; **(3)** output their corresponding sequences together with the contig they belong to; **(4)** map back the reads of all read sets on the sequences of these motifs, mainly in order to recover the coverage and read quality per read sets; **(5)** output a Fasta file containing for each predicted SNP a couple of sequence distinct by the predicted variant(s). Among other informations, the headers of the sequences provides the coverage per allele and per read set. A score between 0 and 1 is affected to each prediction. This score is the Phi coefficient of the table of read counts, computed as follows: $\sqrt{\frac{\chi^2}{n}}$. When two or more read sets are used, this score enables to distinguish predicted SNPs due to inexact repeats from real SNPs as inexact repeats are likely to have a similar profile in each dataset. Additional, as presented in the next section, a genotype per read set is computed for each predicted variant.

Finally, during a last step **(6)**, *DiscoSnp++* generates a VCF file either using a reference genome or not. This step is described in Section "*Providing a VCF file*".

Predicting genotypes

For each prediction and each dataset, a genotype is provided assuming each dataset corresponds to a diploid individual. Genotypes are inferred independently for each individual, based on the read coverage of each allele of the bubble. To do so, the likelihoods of the three possible genotypes (homozygous 0/0 or 1/1 or heterozygous, 0/1) are computed based on a simple binomial model as described in the Nielsen 2011 review [14], see Additional File 1 for likelihood formula and details). These computations rely on only one parameter, namely the *error probability*, that is the probability that a read maps to a given allele erroneously, it was fixed to 0.01, referring to classical sequencing error rates. Finally, the genotype showing the largest likelihood is chosen and all three likelihoods are also output (-log10 transformed) as additional information. Notably, only the probability of observing the data (ie. the read counts for each allele) given the genotype is computed, no prior is used so as to compute the posterior probabilities of each genotype given the data. Users may deactivate genotyping, for instance when the input datasets are not coming from diploid individuals.

Providing a VCF file

The original *DiscoSnp* output is a fasta file. It contains a couple of sequences for each predicted polymorphism. Headers of these sequences contain the read count and average read quality per allele, and the predicted genotype, and these for each input read set. The read coverages for all datasets are used to provide additionally a trustful ranking of the predictions based on their discrimination between the datasets (see for instance the results presented Figure 4).

In *DiscoSnp++*, in addition to the original fasta format, a VCF file is output. This file provides all pieces of information contained in the original fasta headers.

Importantly, if a reference genome is provided, *DiscoSnp++* predictions are mapped to it. In this case, for the variants successfully mapped to the reference, the VCF file contains their genomic position, the reference and the alternative allele unless none of the mapped sequences correspond to the reference genome, and additional mapping information (see Additional File 1 for details about the mapping process and details about the VCF content). Notably, each polymorphism is classified according to the uniqueness of its mapping and this enables to further identify (and possibly filter out) putative false positives due to repeats in the genome. If no reference is provided, the genomic position fields are replaced by dummy value '.'.

Other improvements

The code was re-implemented from scratch using the GATB library [13]. This provides a much easier way to handle the graph file, this improves the running time, with clearer progress messages. In particular, the graph file format is common to other GATB tools, and the graph can be computed once for a given dataset and later be used with several other tools such as de novo assembly (minia [15]) and inversion discovery (TakeABreak[]).

Additionally, *DiscoSnp++* handles pair of reads during the counting phase. Simply, each pair of read files is considered as a unique read set while computing coverages, qualities, ranks and genotypes.

Results

We propose results both on synthetic and on real datasets. Synthetic datasets offer a way to exactly compute the precision and the recall of *DiscoSnp++* and of state-of-the-art methods.

Results on synthetic datasets

We first propose a bench of results based on synthetic datasets. As presented in the Additional File 1, these datasets are derived from real genomes, either from *Escherichia Coli* or from the human chromosome 1.

In these experiments, we generated the set of SNPs and indels of size no longer than ten nucleotides. Thus, we dispose of the exact and exhaustive list of variants to be found, and we are able to compute the precision and the recall of the predictions (see Additional File 1 for precision and recall computation details). We tested *DiscoSnp++*, *cortex* [8] and an hybrid method composed of SOAPdenovo2 [16] for generating the assembly, Bowtie2 [17] for mapping the reads on the assembly, and GATK [1] for calling variants. Presented results were obtained using *UnifiedGenotyper* option of GATK. As presented in the Additional File 1, when following the

GATK guidelines (including read realignment and using the *HaplotypeCaller* option), the result quality is similar but at the expense of a much longer execution time (almost 3 times longer, from approximately 19h to 54h for the human chromosome 1 experiment).

Two and more bacterial read sets

We performed an experiment on a variable number of read sets. Each read set corresponds to a simulation of the sequencing of an *Escherichia Coli* individual. As presented in the Additional File 1, we simulated SNPs and indels of size in 1 to 10 (average 5) such that the distributions of shared polymorphism between subsets of individuals are realistic.

Precision and recall results, presented in Figure 2, allow to draw the following conclusions while calling SNPs from several haploid individuals. On these data, for calling both SNPs and indels, the *cortex* precision is perfect or nearly perfect, whereas the recall strongly decreases when the number of read sets increases, until reaching less than 9% for 30 individuals.

For SNP calling, the hybrid and *DiscoSnp++* provide excellent recall (respectively at least 98.98% and 96.65%). We may however note that recall slightly decreases for *DiscoSnp++* when increasing the number of read sets while this is not the case for the hybrid approach. The precision is stable for the hybrid approach whatever the number of read sets ($\approx 91\%$), while it increases from 91% to 98% with the number of read sets for the *DiscoSnp++* approach.

About the indel calling, the hybrid strategy shows bad performances, that may be explained by the difficulty of mapping reads with indels. Conversely, *DiscoSnp++* presents high quality results both in terms of precision and in terms of recall.

In short, in term of results quality, *DiscoSnp++* outperforms the compared methods except considering the SNP precision which is similar to the hybrid approach one. Importantly, as shown in Figure 3, *DiscoSnp++* runs much faster that other methods and uses much less RAM memory. Moreover, one may also insist on the fact that *DiscoSnp++* is extremely simple to use. Table 1, showing the number of operations to perform for each method, witnesses unequivocally this simplicity.

Two human datasets

For testing *DiscoSnp++* on a diploid and more complex genome, we propose an experiment based on the human chromosome 1, assembly GRCh37. Using variants from the 1000 genomes project, we simulated two individuals, generating 25,928 indels of average size in 1 to 10 (average 2.1) and 288,069 SNPs. The data simulation protocol is presented in the Additional File 1.

Results considering all predictions are presented Table 2. The main conclusion is that, except from the hybrid approach that predicts few indels (40.97%) with a high precision (96.15%), other results do not show notable difference, even if one may notice that precision is globally higher for indels, whereas the recall is higher for SNPs.

Results presented Figure 4 provide additional pieces of information for the hybrid and the *DiscoSnp++* approaches. They show precision/recall values with respect to the ranking of the predictions (*cortex* results are not ranked in this framework).

Results show that the hybrid approach predictions are badly ranked: it appears that predictions showing the best scores are mainly false positives. Additionally, results show that the *DiscoSnp++* ranking is extremely efficient for separating false positives from true positives. Most of the predictions ranked with a score > 0.2 are true positives (97.78% of the SNPs and 98.97% of the indels).

From these experiment on a complex eukaryotic species, one may conclude that *DiscoSnp++* overall results quality is similar to the ones from other methods. However, it is the only tool with a reliable ranking of the results, enabling to select more than 50% of the predictions with a nearly perfect precision.

As previously mentioned, *DiscoSnp++* provides for each predicted variant, its genotype estimation for each individual. On this dataset, 245,690 true positive variants were predicted thus generating 491,380 predictions (one per individual). Over those predicted genotypes, 240,935 (98.06%) were correct.

DiscoSnp++ offers the possibility to map predictions on a reference genome. Among other pieces of information, the mapping provides for each mapped variant its number of mapping position from zero to n. Considering only variants having a unique mapping position in the reference genome enables to select those that are less likely to be due to inexact repeats. This is verified in this dataset as the precision of uniquely mapped SNPs is 87.6% while precision of all SNPs is 71.7%. Table 3 TODO

As shown in Figure 5, *DiscoSnp++* runs much faster that other methods (3.6x and 17.5x times faster than *cortex* and the hybrid approach respectively) and uses much less RAM memory (36.2x and 22.9x times less memory than *cortex* and the hybrid approach respectively).

Results on real datasets

We used a set of biologically validated SNPs predicted from an artificial evolution study on *Saccharomyces cerevisiae* [18]. In this study, three glucose-limited, chemostat-evolved populations of the haploid yeast strain S288c, named E1, E2 and E3, were sequenced every ≈ 70 generations, giving eight samples per population. Using a reference-based mapping approach, 110 mutations were discovered, among which only 33 have a minor allele frequency (MAF) $> 10\%$ and 32 were confirmed by Sanger sequencing. *DiscoSnp++* was run independently on populations E1, E2 and E3. For each population, *DiscoSnp++* was applied on the eight read sets corresponding to the eight time points, with the default parameters and c = 11.

This dataset enables to evaluate *DiscoSnp++* recall for SNPs only on real read datasets. Among the 32 experimentally validated SNPs, 29 were predicted by *DiscoSnp++*, leading to an estimated recall of 90.7%. Using parameter -b 2 leads to the detection of the unpredicted SNPs. The fact that these SNPs were not detected with the default (-b 1) parameter means that its bubble is symmetrically branching (see [12] for an explanation of the branching filtering strategies). This suggests that these SNPs are located in complex regions of the genome.

Note that in the [18] study, no SNP with a MAF $< 10\%$ were validated and no indel were validated, so we could not assess the precision of these types of *DiscoSnp++* predictions on this dataset.

Availability and requirements

- **Project name:** *DiscoSnp++*
- **Operating systems**: Linux and osx;
- **Programming language**: C++ (main algorithms), bash and python;
- **Other requirement**: BWA [19] if users require to map predictions to a reference genome when generating a VCF ouput;
- **License**: GNU AFFERO GENERAL PUBLIC LICENSE `gnu.org/licenses/agpl.html`
- **Any restrictions to use by non-academics**: license needed

Availability of supporting data

[TODO if suitable for review]

The data sets supporting the results of this article are available in the [repository name] repository, [unique persistent identifier and hyperlink to dataset(s) in http:// format].

Abbreviations

NGS: Next Generation Sequencing; SNP: Single Nucleotide Polymorphism; indel: insertion or deletion; PCR: Polymerase Chain Reaction; dBG: de Bruijn Graph; VCF: Variant Call Format; SAM: Sequence Alignment/Map; BAM: Binary Alignment/Map

Competing interests

The authors declare that they have no competing interests.

Author's contributions

ED implemented the GATB library and re-coded *DiscoSnp* using this library, including parallelization and optimizations. PP designed and implemented the close SNPs and deletion detection algorithms. CL designed and implemented the genotyping algorithms. CR designed and implemented the VCF generation algorithms. PP conceived and coordinated of the study, he wrote the manuscript draft. All authors participated in the writing, read and approved the final manuscript.

Acknowledgements

We thank the *GenOuest* (`genouest.org`) cluster team, who allowed us to perform all the tests. This work was supported by the French ANR-12-BS02-0008 *Colib'read* project and by the ANR-12-EMMA-0019-01 **GATB** project.

References

1. 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. Nature genetics **43**(5), 491–498 (2011)
2. Li, H., Handsaker, B., Wysoker, A., Fennell, T., Ruan, J., Homer, N., Marth, G., Abecasis, G., Durbin, R.: The Sequence Alignment/Map format and SAMtools. Bioinformatics (Oxford, England) **25**(16), 2078–9 (2009). doi:10.1093/bioinformatics/btp352
3. Weisenfeld, N.I., Yin, S., Sharpe, T., Lau, B., Hegarty, R., Holmes, L., Sogoloff, B., Tabbaa, D., Williams, L., Russ, C., Nusbaum, C., Lander, E.S., MacCallum, I., Jaffe, D.B.: Comprehensive variation discovery in single human genomes. Nat Genet **46**(12), 1350–1355 (2014)
4. Li, H.: Exploring single-sample snp and indel calling with whole-genome de novo assembly. Bioinformatics **28**, 1838–1844 (2012). doi:10.1093/bioinformatics/bts280. 1203.6364
5. Bradnam, K., Fass, J., Alexandrov, A., Baranay, P., Bechner, M., Birol, I., Boisvert, S., Chapman, J., Chapuis, G., Chikhi, R., Chitsaz, H., Chou, W.-C., Corbeil, J., Del Fabbro, C., Docking, T., Durbin, R., Earl, D., Emrich, S., Fedotov, P., Fonseca, N., Ganapathy, G., Gibbs, R., Gnerre, S., Godzaridis, E., Goldstein, S., Haimel, M., Hall, G., Haussler, D., Hiatt, J., Ho, I.: Assemblathon 2: evaluating de novo methods of genome assembly in three vertebrate species. GigaScience **2**(1), 10 (2013). doi:10.1186/2047-217X-2-10
6. Willing, E.-M., Hoffmann, M., Klein, J.D., Weigel, D., Dreyer, C.: Paired-end RAD-seq for de-novo assembly and marker design without available reference. Bioinformatics (Oxford, England), 1–8 (2011). doi:10.1093/bioinformatics/btr346
7. Peterlongo, P., Schnel, N., Pisanti, N., Sagot, M.-F., Lacroix, V.: Identifying snps without a reference genome by comparing raw reads. In: Chavez, E., Lonardi, S. (eds.) String Processing and Information Retrieval. Lecture Notes in Computer Science, vol. 6393, pp. 147–158. Springer, ??? (2010)

8. Iqbal, Z., Caccamo, M., Turner, I., Flicek, P., McVean, G.: De novo assembly and genotyping of variants using colored de bruijn graphs. Nature genetics **44**(2), 226–232 (2012)
9. Leggett, R.M., Ramirez-Gonzalez, R.H., Verweij, W., Kawashima, C.G., Iqbal, Z., Jones, J.D.G., Caccamo, M., MacLean, D.: Identifying and Classifying Trait Linked Polymorphisms in Non-Reference Species by Walking Coloured de Bruijn Graphs. PLoS ONE **8**(3), 60058 (2013). doi:10.1371/journal.pone.0060058
10. Nordström, K.J.V., Albani, M.C., James, G.V., Gutjahr, C., Hartwig, B., Turck, F., Paszkowski, U., Coupland, G., Schneeberger, K.: Mutation identification by direct comparison of whole-genome sequencing data from mutant and wild-type individuals using k-mers. Nature Biotechnology **31**(4), 325–330 (2013). doi:10.1038/nbt.2515
11. Sacomoto, G.A., Kielbassa, J., Chikhi, R., Uricaru, R., Antoniou, P., Sagot, M.-F., Peterlongo, P., Lacroix, V.: Kissplice: de-novo calling alternative splicing events from rna-seq data. BMC bioinformatics **13**(Suppl 6), 5 (2012)
12. Uricaru, R., Rizk, G., Lacroix, V., Quillery, E., Plantard, O., Chikhi, R., Lemaitre, C., Peterlongo, P.: Reference-free detection of isolated SNPs. Nucleic acids research **33**(0), 1–11 (2014). doi:10.1093/nar/gku1187
13. Drezen, E., Rizk, G., Chikhi, R., Deltel, C., Lemaitre, C., Peterlongo, P., Lavenier, D.: GATB: Genome Assembly & Analysis Tool Box. Bioinformatics (Oxford, England), 1–3 (2014). doi:10.1093/bioinformatics/btu406
14. Nielsen, R., Paul, J.S., Albrechtsen, A., Song, Y.S.: Genotype and SNP calling from next-generation sequencing data. Nature reviews. Genetics **12**, 443–451 (2011). doi:10.1038/nrg2986
15. Chikhi, R., Rizk, G.: Space-efficient and exact de Bruijn graph representation based on a Bloom filter. Algorithms Mol Biol **8**(1), 22 (2013). doi:10.1186/1748-7188-8-22
16. Luo, R., Liu, B., Xie, Y., Li, Z., Huang, W., Yuan, J., He, G., Chen, Y., Pan, Q., Liu, Y., *et al.*: Soapdenovo2: an empirically improved memory-efficient short-read de novo assembler. GigaScience **1**(1), 18 (2012)
17. Langmead, B., Salzberg, S.L.: Fast gapped-read alignment with bowtie 2. Nature methods **9**(4), 357–359 (2012)
18. Kvitek, D.J., Sherlock, G.: Whole genome, whole population sequencing reveals that loss of signaling networks is the major adaptive strategy in a constant environment. PLoS genetics **9**(11), 1003972 (2013). doi:10.1371/journal.pgen.1003972
19. Li, H., Durbin, R.: Fast and accurate short read alignment with Burrows-Wheeler transform. Bioinformatics **25**(14), 1754–1760 (2009). doi:10.1093/bioinformatics/btp324

Figures

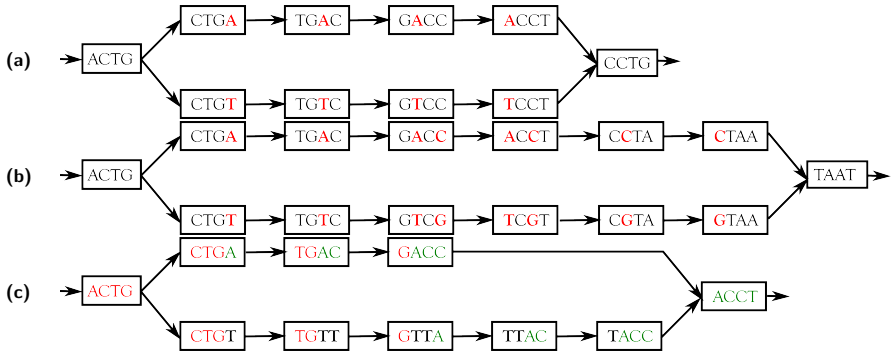

Figure 1 Examples of dBG ($k = 4$) bubbles due to SNP, close SNPs and indel. (a) Bubble generated by an isolated SNP. Detected both by *DiscoSnp* and *DiscoSnp++*. Prediction would be ACTGACCTG and ACTGTCCTG. **(b)** Bubble generated by two close SNPs. Detected only by *DiscoSnp++*. Prediction would be ACTGACCTAAT and ACTGTCGTAAT. **(c)** Bubble generated by an insertion. Detected only by *DiscoSnp++*. Prediction would be ACTGACCT and ACTGTTACCT.

Tables

Table 1 Command line complexity in term of required number of command (including file formatting when necessary) while calling variants from 2 and 30 haploid genomes. See Additional File 1 for details.

	Number of commands for two genomes	Number of commands for 30 genomes
Hybrid	19	187
cortex	8 (+2 compilations)	35 (+30 compilations)
DiscoSnp++	2	2

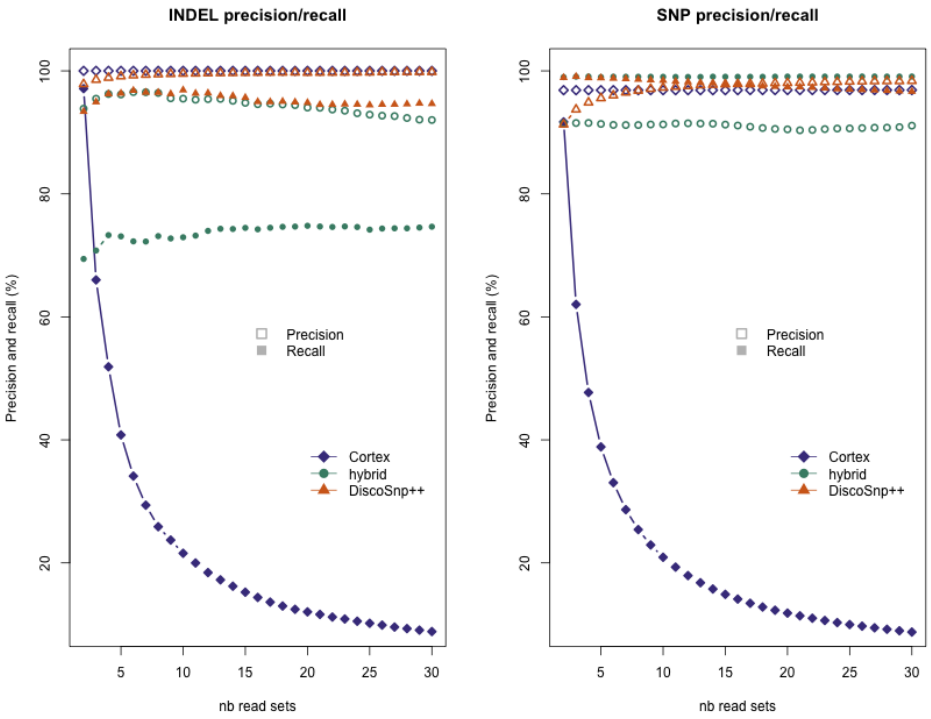

Figure 2 *DiscoSnp++*, *cortex* **and hybrid strategy (SOAPdenovo2+GATK) results, depending on the number of input haploid individuals.** SOAPdenovo2 and GATK were launched with default parameters. For *DiscoSnp++* and *cortex*, k-mers having three or fewer occurrences in all datasets were removed. Precision and recall: filled symbols represent the precision and empty symbols represent the recall. **Left:** results on indels predictions. **Right:** results on SNPs predictions.

Table 2 Human chromosome 1 results.

	SNP		indels	
	Precision (%)	Recall (%)	Precision (%)	Recall (%)
Hybrid	71.60	78.59	**96.15**	40.97
cortex	73.19	67.34	86.65	63.25
DiscoSnp++	71.71	**78.88**	75.86	**71.15**
DiscoSnp++ (rank > 2)	97.78	64.39	98.97	57.78

Table 3 Human chromosome 1 *DiscoSnp++* results (number of TP and FP) with respect to the type of variant (SNP or indel) and of the number of mapping positions on the reference genome.

	SNP		indels	
	TP	FP	TP	FP
Unique	233483	33111	10086	1075
None	516	581	0	0
Multiple	3920	45221	174	1501

Additional Files

Additional file 1 — Sample additional file title

Additional file descriptions text (including details of how to view the file, if it is in a non-standard format or the file extension). This might refer to a multi-page table or a figure.

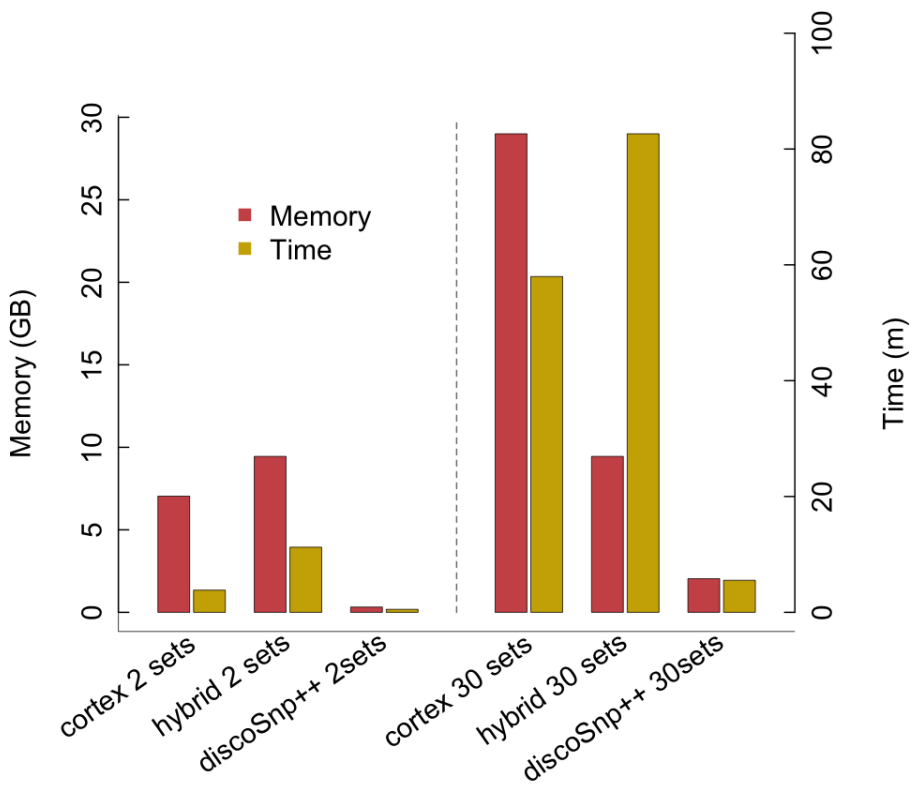

Figure 3 Comparative memory and time performances, depending on the number of input haploid individuals.

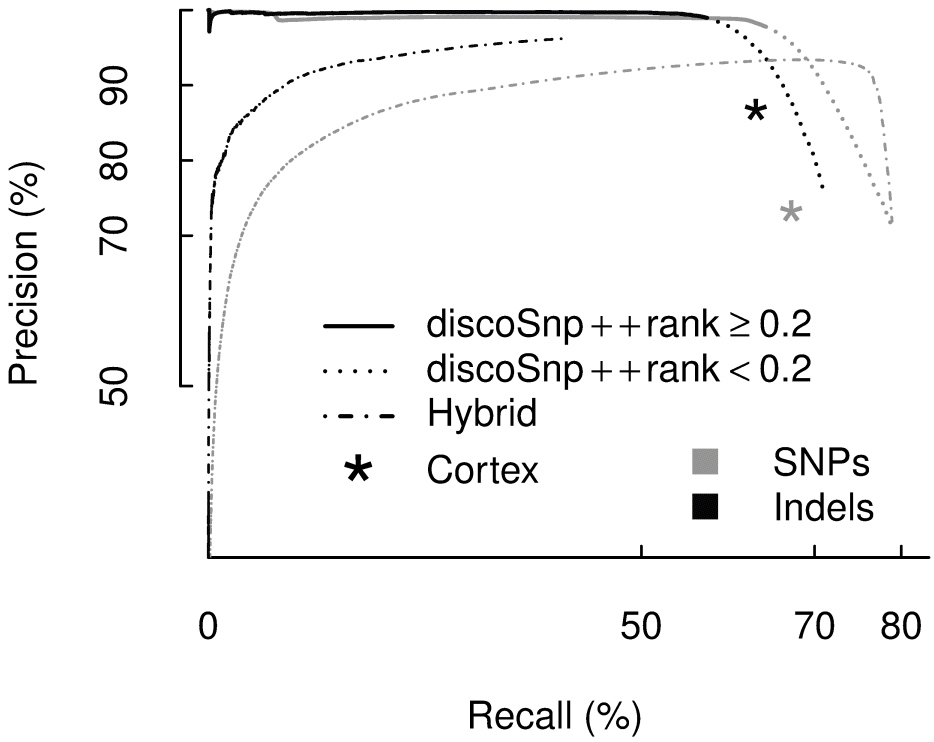

Figure 4 Comparative results of $DiscoSnp++$, $cortex$, and the hybrid SOAPdenovo2 + Bowtie2 + GATK approaches on the two diploid human chromosome 1 dataset. Precision versus recall curves are obtained by ranking the predicted SNPs and indels. Each data point is obtained at a given rank threshold, where precision and recall values are computed for all SNPs with better ranks than this threshold. The dashed tail of the two $DiscoSnp++$ curves denotes the predictions ranked with a threshold bellow 0.2. In this framework $cortex$ does not rank its predictions, its results are thus represented by a single point.

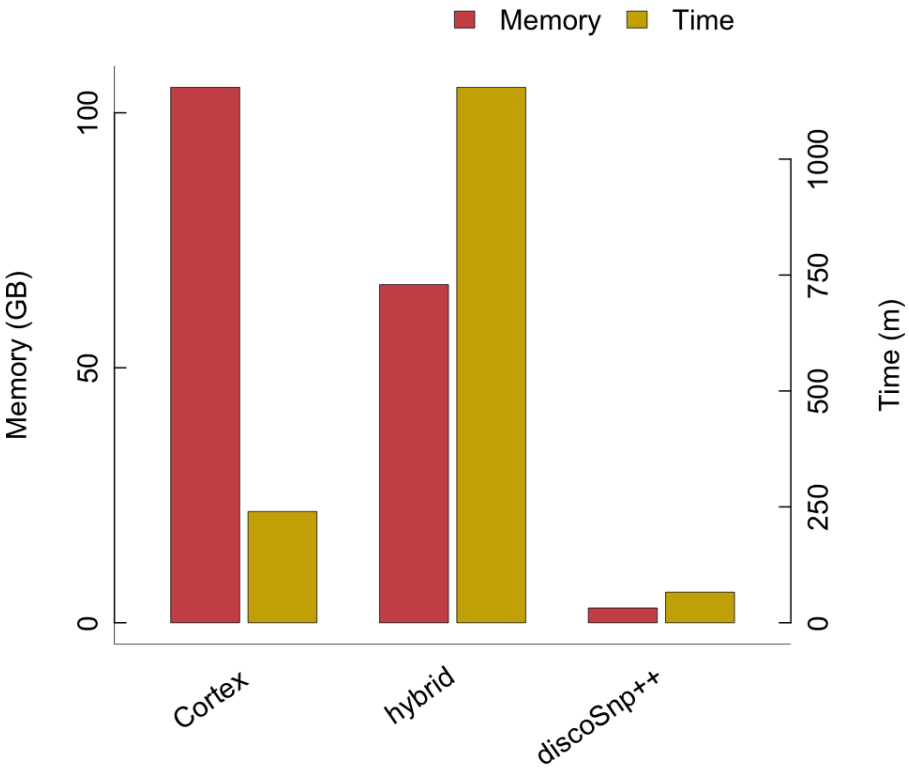

Figure 5 Comparative memory and time performances for comparing the two human datasets