Wijfjes *et al. BMC Genomics* (2019) 20:818
https://doi.org/10.1186/s12864-019-6153-8

BMC Genomics

SOFTWARE Open Access

Hecaton: reliably detecting copy number variation in plant genomes using short read sequencing data

Raúl Y. Wijfjes*, Sandra Smit and Dick de Ridder

Abstract

Background: Copy number variation (CNV) is thought to actively contribute to adaptive evolution of plant species. While many computational algorithms are available to detect copy number variation from whole genome sequencing datasets, the typical complexity of plant data likely introduces false positive calls.

Results: To enable reliable and comprehensive detection of CNV in plant genomes, we developed Hecaton, a novel computational workflow tailored to plants, that integrates calls from multiple state-of-the-art algorithms through a machine-learning approach. In this paper, we demonstrate that Hecaton outperforms current methods when applied to short read sequencing data of *Arabidopsis thaliana*, rice, maize, and tomato. Moreover, it correctly detects dispersed duplications, a type of CNV commonly found in plant species, in contrast to several state-of-the-art tools that erroneously represent this type of CNV as overlapping deletions and tandem duplications. Finally, Hecaton scales well in terms of memory usage and running time when applied to short read datasets of domesticated and wild tomato accessions.

Conclusions: Hecaton provides a robust method to detect CNV in plants. We expect it to be of immediate interest to both applied and fundamental research on the relationship between genotype and phenotype in plants.

Keywords: Copy number variation, Structural variation, Plant adaptation, Machine learning

Background

Phenotypic variation between individuals of the same plant species is caused by a host of different types of genetic variation, including single nucleotide polymorphisms (SNPs), small insertions and deletions, and larger structural variation. One major class of structural variation is copy number variation (CNV), which is defined as deletions, insertions, tandem duplications and dispersed duplications of at least 50 bp. CNV comprises a large part of the genetic variation found within plant populations and is thought to play a key role in adaptation and evolution [1]. One clear example of such adaptive evolution is presented by the weed species *Amaranthus palmeri*, which rapidly became resistant to a widely used herbicide through amplification of the EPSPS gene, resulting in increased expression [2]. Similar relationships between CNV and adaptation were found in domesticated crop species [3], indicating that CNV may offer a pool of genetic variation that can be used to improve crop cultivars.

Given the increasing interest of the plant research community in CNV [1, 3, 4], the question arises whether current methods accurately detect copy number variants (CNVs) in plants. Currently, CNVs are mainly analyzed by whole genome sequencing (WGS). After a sample of interest has been sequenced and the resulting sequencing data has been aligned to a reference genome, computational methods can extract various signals from the alignments to detect CNV between the sample and the reference [5]. While long reads are better suited for detecting CNVs than short paired-end reads [6, 7], sequencing data of plants is still commonly generated using short read sequencing platforms, due to their far lower cost.

*Correspondence: raul.wijfjes@wur.nl
Bioinformatics Group, Wageningen University & Research, Wageningen, the Netherlands

Although current state-of-the-art CNV detection algorithms generally perform well when applied to human datasets [8], the typical complexity of plant data likely introduces false positive calls. First, reference genome assemblies of plants generally contain a larger number of gaps than the human reference genome, as plant genomes are difficult to assemble due to their repetitive nature. Yet, the genomic sequence contained in such gaps is still present in WGS data of samples. The reads representing this sequence generally share high similarity with other assembled regions of the reference, to which they are incorrectly aligned as a result. Second, sampled plant genomes can differ significantly from reference genome assemblies, particularly if samples represent out-bred or natural accessions. If a region in a sample genome has undergone several mutations relative to the reference, reads sequenced from this region may map to a different region than the one it is syntenic to. This is particularly likely to happen if the region that the reads originated from is highly repetitive. Third, several CNV detection algorithms erroneously process alignments resulting from dispersed duplications [9]. We expect that this issue introduces a significant number of false positives when working with plant data, as duplication and transposition of genomic sequences is considered to be one of the main drivers of adaptive evolution in plants [10].

To enable reliable and comprehensive detection of copy number variants in plant genomes, we developed Hecaton, a novel computational workflow that combines several existing detection methods, specifically tailored to detect CNV in plants. Combining methods generally results in higher recall and precision than using a single tool [11, 12], as the recall and precision of individual tools varies among different types and sizes of CNVs, depending on their algorithmic design [8]. However, determining the optimal strategy to integrate different methods is not straightforward. A suboptimal integration approach may yield only a small gain of precision, while significantly decreasing recall [8, 13]. Hecaton tackles this challenge in two ways. First, it makes use of a custom post-processing step to correct erroneously detected dispersed duplications, which are systematically mispredicted by some state-of-the-art tools. Second, it utilizes a machine-learning model which classifies detected calls as true and false positives by leveraging several features describing a detected CNV call, such as its type and size, along with concordance among the callers used to detect it. In this paper, we demonstrate that Hecaton outperforms existing individual and ensemble computational CNV detection methods when applied to plant data and provide an example of its utility to the plant research community.

Implementation

Selected CNV calling tools

To maximize the performance of Hecaton, we combine predictions of a diverse set of popular, open-source tools that complement each other in terms of the signals and strategies used to call CNVs. The selected tools include Delly [14] (version 0.7.8), GRIDSS [15] (version 1.8.1), LUMPY [16] (version 0.2.13), and Manta [17] (version 1.4.0). Delly detects CNVs using discordantly aligned read pairs and refines the breakends of detected events using split reads. LUMPY improves upon this method by integrating both of these signals to detect CNVs, as opposed to using them sequentially. Manta and GRIDSS further enhance this strategy by performing local assembly of sequences flanking breakends identified by discordantly aligned read pairs and split reads. We considered including CNVnator [18] (version 0.3.2), Control-FREEC [19] (version 10.4), and Pindel [20] (version 0.2.5b9). Pindel was dropped after showing an excessively long run time when applied to simulated high coverage datasets. CNVnator and Control-FREEC were excluded as they performed poorly during evaluations (Additional File 1: Figure S1).

Implementation of hecaton

Hecaton is a workflow specifically designed to reliably detect CNVs in plant genomes. We aimed to implement it in such a manner that it is both reproducible and easy-to-use. To this end, Hecaton is run with a single command using the Nextflow [21] framework, which provides a unified method to chain together and parallelize the different processes that are executed. It consists of three stages: calling, post-processing, and filtering (Fig. 1). Currently, Hecaton only supports the four CNV detection algorithms used during the calling stage, but can be relatively easily extended to include other tools.

Stage 1: Calling

The calling stage takes paired-end Illumina WGS data of a sample of interest and a reference genome as input and calls CNVs between the sample and reference using four different tools. First, it aligns the sequencing data to the reference using the Speedseq pipeline [22] (version 0.1.2) with default parameters. This pipeline utilizes bwa mem [23] (version 0.7.10-r789) to align reads, SAMBLASTER [24] (version 0.1.22) to mark duplicates and Sambamba [25] (version 0.5.9) to sort and index BAM files. The resulting sorted BAM file is processed by Delly, LUMPY, Manta and GRIDSS to call CNVs. Each of these tools is run with default parameters, except for the number of supporting reads required by LUMPY and Manta for a CNV to be included in the output (lowered to 1 to maximize recall). Delly and GRIDSS do not apply any filters

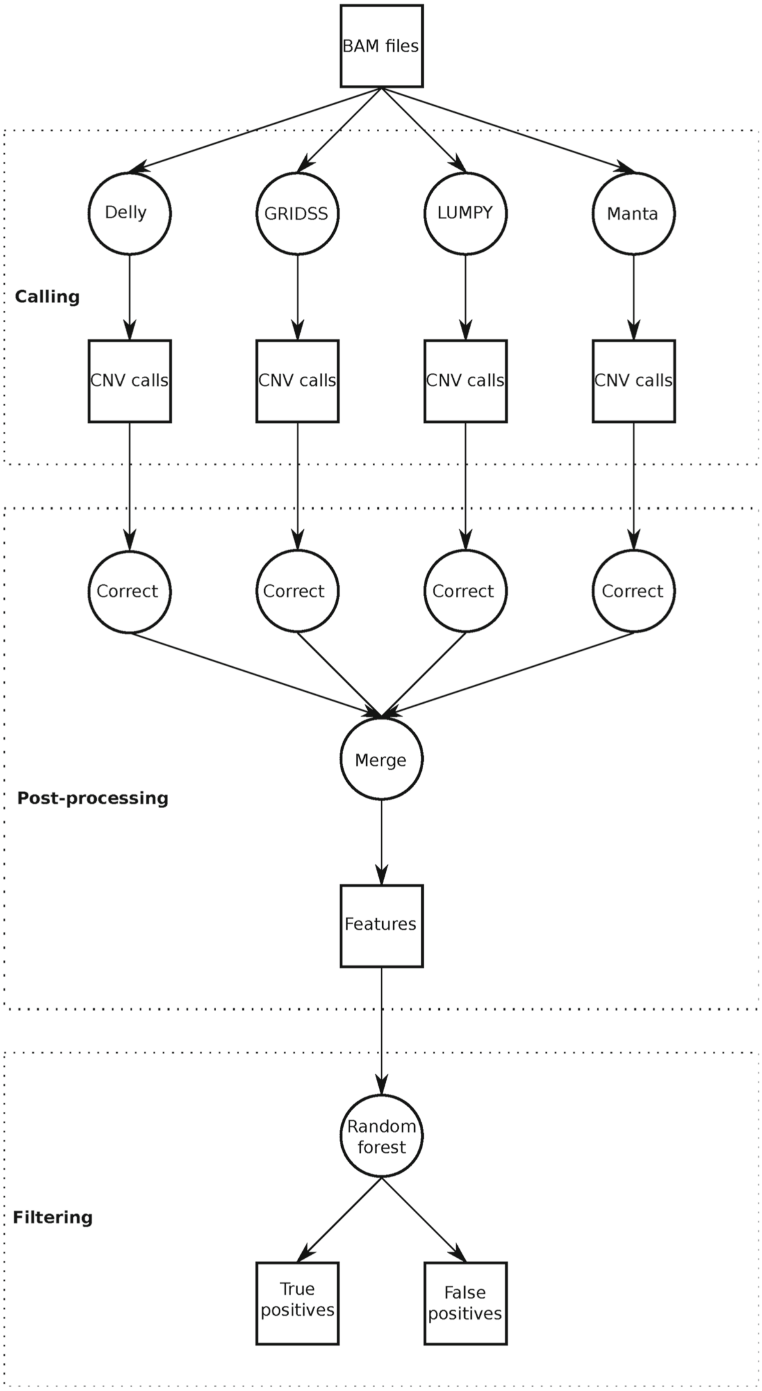

Fig. 1 Overview of Hecaton. CNVs are first called using four different tools. The resulting calls are corrected and merged into a set of features. These features are used by the random forest model to discriminate between true and false positives

by default. The final output of the calling stage consists of four VCF files containing CNVs, one for each tool.

Stage 2: Post-processing

The post-processing stage of Hecaton serves three purposes. First, it provides an automated method to process the output files of different tools using a common representation, which is necessary to properly integrate them. Second, it corrects dispersed duplications that have been detected by CNV tools as overlapping deletions and tandem duplications by mistake. Third, it merges calls produced by different tools that likely correspond to the same CNV event.

The common representation of CNVs used by Hecaton is based on the concept that each structural variant can be represented as a set of novel adjacencies. A novel adjacency is defined as a pair of bases that are adjacent to each other in the genome of a sample of interest, but not in the genome of the reference to which the sample is compared. Bases that are linked by a novel adjacency are called breakends and two breakends that corresponds to the same adjacency are referred to as mates. Although Delly, GRIDSS, LUMPY, and Manta all generate a VCF file as output, the way in which CNV calls and the evidence supporting them are represented in this file is different for each tool. For example, the output of Delly, LUMPY, and Manta contains both CNVs and breakends, while that of GRIDSS solely consists of breakends that need to be converted to CNVs by the user.

To convert the output of each tool to a common CNV format and correct erroneous dispersed duplications, Hecaton reclassifies the adjacencies underlying the CNV calls produced by each tool. First, it infers and collects adjacencies from all sets of CNVs generated during the calling stage. For example, it represents deletions as a single adjacency containing two breakends positioned on the 5' and 3' end of the deleted sequence. Next, it clusters adjacencies generated by the same tool of which the breakpoints are located within 10 bp of each other on either the 5' end or 3' end, as these are likely to be part of the same variant. Finally, it converts each cluster to a deletion, insertion, tandem duplication, or dispersed duplication, based on the relative positions of the breakends and the orientation of the sequences that are joined in a cluster. Deletions, insertions, and tandem duplications are represented by single adjacencies, while dispersed duplications are represented by two (Additional File 1: Figure S2). As the objective of Hecaton is to detect CNV and not any other form of structural variation, it excludes any set of adjacencies that cannot be classified as one of these four types from further analysis. However, Hecaton can be extended to support additional types of structural variation if needed.

Hecaton collapses calls produced by different tools that are likely to correspond to the same CNV event. Calls are merged if they fulfill all of the following conditions: they are of the same type; their breakpoints are located within 1000 bp of each other on both the 5' and 3' end; they share at least 50% reciprocal overlap with each other (does not apply to insertions); and the distance between the insertion sites is no more than 10 bp (only applies to dispersed duplications and insertions). The regions of the merged calls are defined as the union of the regions of the "donor" calls. For instance, one call that covers positions 12-30 and one call that covers positions 14-32 are merged into a call covering positions 12-32. The number of discordantly aligned read pairs and split reads supporting a merged call are both defined as the median of the numbers of the donor calls. The final result of the post-processing stage is a single BEDPE file containing all merged calls.

Stage 3: Filtering

In the filtering stage, Hecaton applies a machine-learning model to remove erroneous CNV calls. First, it generates a feature matrix that represents the set of merged calls. The rows of the matrix correspond to CNV calls and the columns correspond to features (Additional file 2: Table S1), which are extracted from the INFO and FORMAT fields of the VCF file containing the calls.

Hecaton classifies calls as true or false positives using a random forest model. We chose to implement this particular type of machine-learning model, as it outperformed a logistic regression model and a support vector machine. The model assigns a probabilistic score to each merged call based on the set of features defined for it. These scores are posterior probability estimates of calls being true positives and range between 0 and 1. Calls with scores below a specified user-defined cutoff are dropped, producing a BEDPE file containing the final output of Hecaton.

To obtain a random forest model that strikes a good balance between recall and precision, we trained it using a set of CNVs detected from real WGS data for which the labels (true or false positive) were known, based on long read data (see Additional file 3: Supplementary Methods for details on the validation procedure). We did not include CNVs obtained from simulated data in the ground truth set, as the recall and precision attained by Delly, LUMPY, Manta, and GRIDSS on such data generally does not accurately reflect their performance in real scenarios. For example, LUMPY and Manta obtained almost perfect precision when we applied them to simulated datasets with minimum filtering, if dispersed duplications were excluded from the simulation. They showed significantly lower precision in previous benchmarks when applied to real human data [16, 17].

The training and testing set were constructed by running the calling and post-processing stages of Hecaton on Illumina data of an *Arabidopsis thaliana* Col-0–Cvi-0 F1 hybrid and a sample of the *Japonica* rice Suijing18 cultivar (Additional file 2: Table S2). We detected CNVs in these samples relative to the *A. thaliana* Col-0 (version TAIR10) and *Oryza sativa Japonica* (version IRGSP-1.0) reference genome. As we aimed to maximize the performance of the model for low coverage datasets in particular, we subsampled these datasets to 10x coverage using seqtk [26]. Calls were labeled as true or false positives using long read data of the same samples (See Additional file 3: Supplementary Methods for details). To obtain a test set, we held out calls located on chromosomes 2 and 4 of *A. thaliana* and chromosomes 6, 10, and 12 of *O. sativa*, using the remaining calls as the training set. In order to obtain a model that generalizes to multiple plant species, one single model was trained using both Col-0–Cvi-0 and Suijing18 calls. The training set contained 4983 deletions, 393 insertions, 604 tandem duplications and 106 dispersed duplications, while the test set contained 2291 deletions, 174 insertions, 292 tandem duplications and 44 dispersed duplications.

We implemented the random forest model in Python using the scikit-learn package [27] (version 0.19.1). The hyperparameters of the model (*n_estimators*, *max_depth*, and *max_features*) were selected by doing a grid search with 10-fold cross-validation on the training set, using the accuracy of the model on the validation data as optimization criterion.

Benchmarking

The performance of Hecaton was compared to that of current state-of-the-art tools using short read data simulated from rearranged versions of the *Solanum lycopersicum* Heinz 1706 reference genome of tomato [28]; the testing set constructed from *A. thaliana* Col-0–Cvi-0 and rice Suijing18; and real short read data of *A. thaliana* L*er*, maize B73, and several tomato samples (Additional file 2: Table S2). We determined the recall and precision of tools with two validation methods that use long read data: VaPoR [29] and Sniffles [6]. See Additional file 3: Supplementary Methods for full details.

Results and discussion

We present Hecaton, a novel computational workflow to reliable detect CNVs in plant genomes (Fig. 1). It consists of three stages. In the first stage, it aligns short read WGS data to a reference genome of choice and calls CNVs from the resulting alignments using Delly, GRIDSS, LUMPY, and Manta, four state-of-the-art tools that complement each other in terms of their methodological set-up. In the second stage, Hecaton corrects dispersed duplications that are erroneously represented by these tools as overlapping deletions and tandem duplications. In the final stage, Hecaton filters calls by using a random forest model trained on CNV calls validated by long read data. Below, we first describe how the design of Hecaton allows it to outperform the current state-of-the-art and then we will present an application of Hecaton to crop data.

Hecaton accurately detects dispersed duplications

Dispersed duplications are defined as duplications in which the duplicated copy is found at a genomic region that is not adjacent to the original template sequence. Such variants are frequently found in plants, as plant genomes typically contain a large number of class I transposable elements that propagate themselves through a “copy and paste" mechanism. While dispersed duplications may play an important role in the adaptive evolution of plants [10], they can also introduce a significant number of false positives, if they are not taken into account while calling CNVs. To show the impact of this problem, we applied Delly, GRIDSS, LUMPY, and Manta to short read data simulated from modified versions of the *S. lycopersicum* Heinz 1706 reference genome containing different types of CNVs at known locations.

As Delly, LUMPY, and Manta systematically mispredict dispersed duplications, they attained low precision when applied to simulated data (Fig. 2a). We hypothesize that these tools misinterpret the complex patterns of signals resulting from intrachromosomal dispersed duplications during alignment (Additional file 1: Figure S2), as the false positives mostly corresponded to overlapping pairs of large deletions and tandem duplications (Fig. 2b) that cover the sequence located between the template sequence and insertion sites of simulated intrachromosomal dispersed duplications. Such signals consist of novel adjacencies, pairs of bases that are adjacent to each other in the genome of the sample of interest, but not in the genome of the reference to which the sample is compared. Deletions, insertions, and tandem duplications generate a single novel adjacency as a signal. Dispersed duplications, however, generate two novel adjacencies. Delly, LUMPY, and Manta likely process these adjacencies in isolation, resulting in overlapping deletion and tandem duplication calls.

The post-processing step of Hecaton corrects dispersed duplications that are erroneously predicted by Delly, LUMPY, and Manta, which significantly improves their performance. It recovered both intrachromosomal and interchromosomal dispersed duplications when applied to simulated data (Fig. 3a). Moreover, as the post-processing step replaces false positive deletions and tandem duplications by true positive dispersed duplications, it strongly increases the precision of Delly, LUMPY, and Manta (Fig. 3b). The post-processing step also correctly predicts dispersed duplications from the output of GRIDSS, which does not yield CNVs as output, but the adjacencies underlying them (Fig. 3). Post-processing the adjacencies reported by GRIDSS in isolation resulted in a similar trend as seen for Delly, LUMPY, and Manta, underlining the importance of correctly interpreting the signals generated by dispersed duplications.

The performance of the post-processing step improved with coverage (Fig. 3), as it fails to detect dispersed duplications if one or both of the adjacencies resulting from them are missing from the output of Delly, LUMPY, Manta, or GRIDSS. In line with this observation, the post-processing script detected a lower number of dispersed duplications simulated at low allele dosage compared to those simulated at high dosage (Additional file 1: Figure S3), as the effective coverage of variant alleles decreases

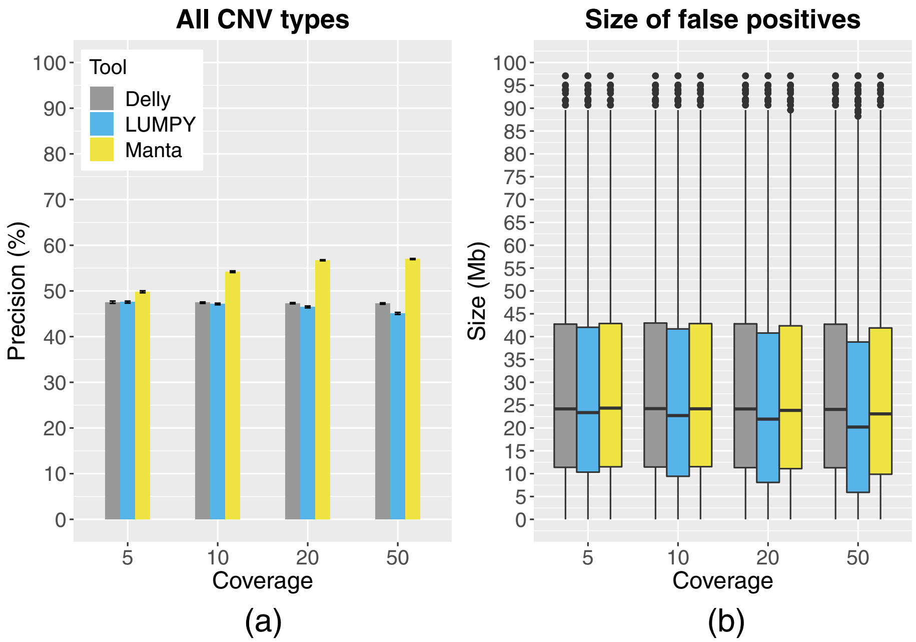

Fig. 2 Performance of Delly, LUMPY, Manta, and GRIDSS on data simulated from diploid rearranged tomato genomes. Performance metrics are reported as the mean over all 10 simulations with error bars depicting the standard error of the mean. The size distributions of the detected false positives are depicted as box plots. The overall precision of Delly, LUMPY, and Manta was low (**a**) and false positives generally consisted of large CNVs having a size of several tens of Mbs (**b**). These corresponded to pairs of large deletions and tandem duplications that covered the sequence located between the template sequence and insertion sites of intrachromosomal dispersed duplications

when they are present in few haplotypes. If only one of the two adjacencies could be detected, the post-processing script classified it as a false positive deletion, false positive tandem duplication, or generic breakend.

Hecaton generally outperforms state-of-the-art cNV detection tools

Intuitively, it makes sense to combine the output of multiple CNV detection tools, as they typically generate complementary results when applied to the same dataset [30]. However, designing a method that optimally integrates tools is not trivial. In a past benchmark, an ensemble strategy that combined tools through a majority vote did not significantly improve upon the best performing individual tool [13]. Here, we demonstrate the benefits of using a machine-learning approach, which aggregates and filters calls based on features including size, type and level of support from different tools. We trained machine-learning models using CNVs detected from 10x coverage short read data of a highly heterozygous *A. thaliana* Col-0–Cvi-0 sample and a Suijing18 rice sample. The labels (true or false positive) of these CNVs were determined using long read data of the same samples. This approach generated accurate validations of calls detected from the simulated *S. lycopersicum* Heinz 1706 datasets.

The machine-learning approach used during the filtering stage of Hecaton integrates calls of Delly, LUMPY, Manta, and GRIDSS in such a manner so that it outperforms each individual tool. When applied to *A. thaliana* Col-0–Cvi-0 and Suijing18 rice calls detected on chromosomes that were held out from model training, it generally attained a more favourable combination of recall and precision across a broad spectrum of thresholds and different CNV types (Fig. 4). For example, at a precision level of 80%, Hecaton detected 43 true positive tandem duplications, while the best performing state-of-the-art tool, GRIDSS, detected only 19. Our results agree with previous work in which a method that carefully merges calls of different CNV calling tools attained a higher precision and recall than any of the individual tools [11]. As the approach performed about equally well when using a random forest model trained on either 10x or 50x coverage data (Additional file 1: Figure S4), the random forest framework itself is the main driver of the improvement, rather than the sequencing coverage used to train the models. To check whether the improved performance held more generally, we applied Hecaton to an Illumina dataset of *A. thaliana* L*er*, a sample that was completely independent from model training. It again improved upon the performance of individual tools (Additional file 1: Figure S5), corroborating the results observed in *A. thaliana* Col-0–Cvi-0 and Suijing18 rice.

Besides outperforming individual tools, the machine-learning approach employed by Hecaton significantly

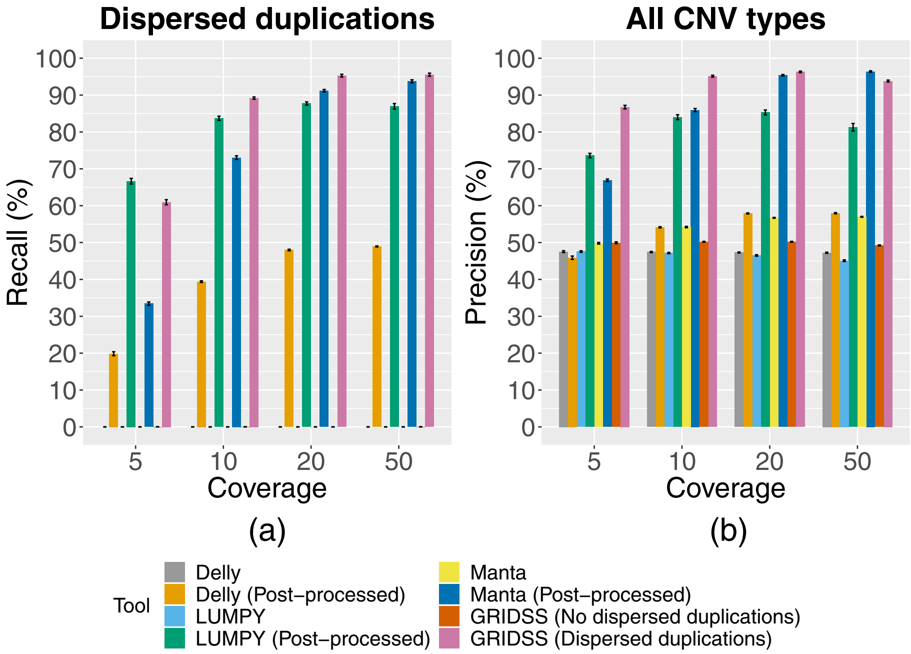

Fig. 3 Performance of the post-processing step of Hecaton on data simulated from diploid rearranged tomato genomes. Performance metrics are reported as the mean over all 10 simulations with error bars depicting the standard error of the mean. Results of GRIDSS were generated by processing adjacencies in isolation (no dispersed duplications) or by processing them in clusters (dispersed duplications). (**a**) Recall of CNV calling tools for dispersed duplications, before and after post-processing. The post-processing script of Hecaton recalled dispersed duplications not originally found in the output of Delly, LUMPY, Manta. (**b**) Overall precision of CNV calling tools, before and after post-processing. The post-processing stage of Hecaton significantly increased the precision of tools by replacing pairs of overlapping false positive deletions and tandem duplications by true positive intrachromosomal dispersed duplications

improved upon current state-of-the-art ensemble methods that are applicable to, but not specifically designed for plant data. It attained a better combination of precision and recall than MetaSV [31], SURVIVOR [32], and Parliament2 [33], three alternative approaches that aggregate the results of different CNV detection tools, when applied to datasets of Col-0–Cvi-0 and Suijing18 (Fig. 4). The poor performance of MetaSV and SURVIVOR sharply contrasts with the good performance they showed in the benchmarks of the publications describing them [31, 32]. One possible reason for this discrepancy could be that both tools were evaluated in these benchmarks using simulated data, which likely does not accurately reflect the distribution of CNVs in real data.

To evaluate Hecaton on more distantly related and repetitive genomes than those of *A. thaliana* and rice, we used it to detect CNVs between the two maize accessions Mo17 and B73. As a large fraction of calls could not be validated using long read data, due to the highly repetitive nature of the Mo17 assembly (Additional File 2: Table S3), we only report performance metrics for calls that overlap for at least 50% of their length with genes or the 5000 bp interval upstream or downstream of genes. We believe that this subset of calls still yields a representative measure of performance, as downstream analysis of CNVs detected by short reads generally focuses on genic, non-repetitive regions. Consistent with the results of our previous benchmarks, Hecaton attained a better combination of recall and precision compared to both individual state-of-the art tools and ensemble approaches (Fig. 5). For example, at a precision level of 90%, it detected a higher number of true positive deletions (13991) than LUMPY (11190), the second-most sensitive approach for deletions at that level of precision. The large number of CNVs detected by Hecaton between Mo17 and B73 confirms the extensive structural variation between the two accessions found by a whole genome alignment based approach [34].

Consistent with previous benchmarks performed with long read data [6, 7], insertions remained difficult to reliably detect using short paired-end Illumina reads in all of our test cases, even after applying the filtering stage of Hecaton. We manually investigated alignments covering tens of false positive insertions in *A. thaliana Ler* and discovered that they all resulted from alignments that were soft-clipped at the insertion site. These insertions were all reported by Hecaton to have an unknown size. With some of the insertions, the mates of the soft-clipped reads mapped to a different chromosome, indicating that some may be interchromosomal transpositions instead.

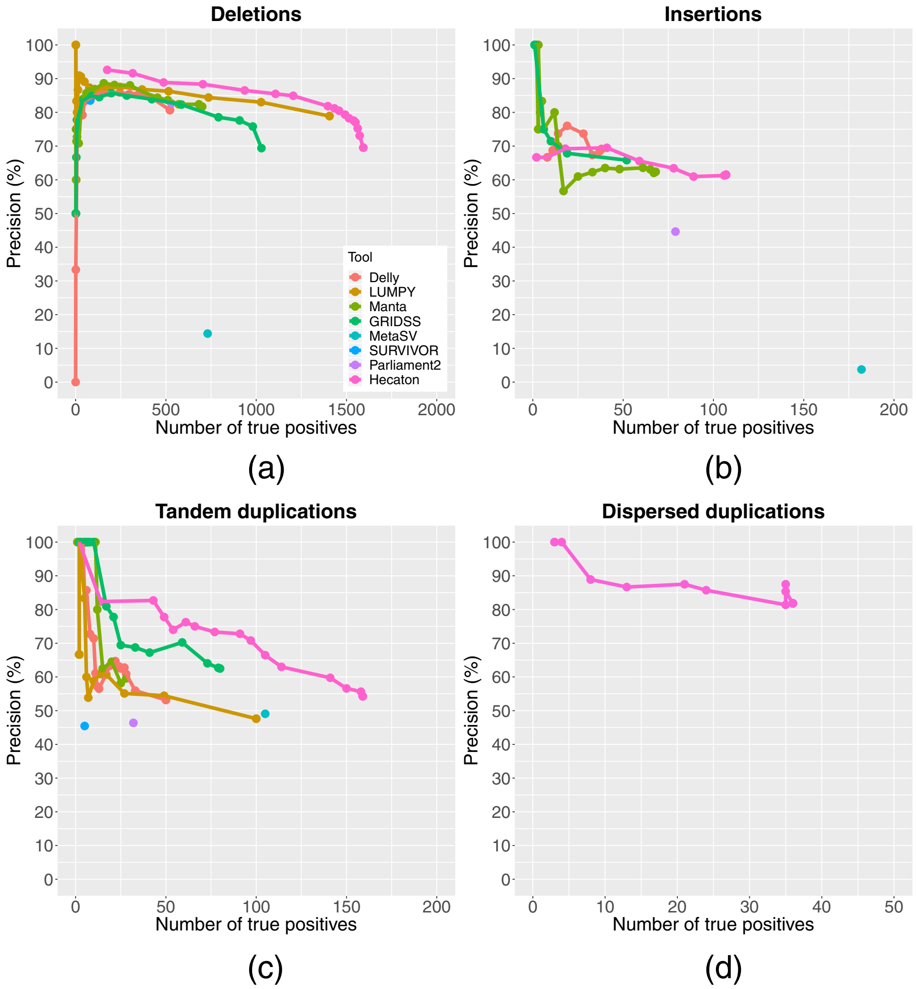

Fig. 4 Performance of different CNV detection algorithms on the test set containing Col-0–Cvi-0 and Suijing18 CNV events. Precision-recall curves of Delly, LUMPY, Manta, and GRIDSS were constructed by varying the minimum number of discordantly aligned read pairs and/or split reads supporting each call in the set. The curve of Hecaton was produced by varying the threshold of the probabilistic score used to define calls as true positives. Performance is shown separately for deletions (**a**), insertions (**b**), tandem duplications (**c**), and dispersed duplications (**d**). Curves of LUMPY and SURVIVOR are missing for insertions, as these tools are unable to detect this type of CNV. Curves are missing for all tools besides Hecaton for dispersed duplications, as Hecaton is the only tool that can detect this type of CNV, owing to its post-processing stage. Hecaton generally improved recall and precision compared to Delly, LUMPY, Manta, and GRIDSS. Moreover, it significantly outperformed MetaSV, SURVIVOR, and Parliament2, three ensemble approaches applicable to plant data

The idea of predictor combination has been already applied to improve detection of structural variants from exome sequencing data [35], detection of somatic single nucleotide variants [36], inference of gene regulatory networks [37], and predictive models of breast cancer prognosis [38]. Our results demonstrate that this concept can be used to improve CNV detection as well, contrasting a previous benchmark that combined structural variation methods through a majority vote [13]. This suggests that the aggregation approach used by Hecaton is better suited to deal with the different treatment of each tool of specific types of CNV than a majority vote. A possible reason for this is that the random forest model employed by Hecaton can capture interactions between tools and CNV types to some extent during aggregation, while a majority vote assigns an equal weight to each tool. Such an approach does not work well if most tools are ill-suited to detect a specific type of CNV.

We demonstrated that Hecaton can generalize to plant species beyond those used in its training set (Fig. 5). The performance of Hecaton can be further improved, as it is relatively easy to extend it to include other CNV detection tools or to train new random forest models using additional plant data. Nevertheless, Hecaton has some limitations. First, it has limited recall for dispersed duplications when applied to very low coverage (5x) data. Second,

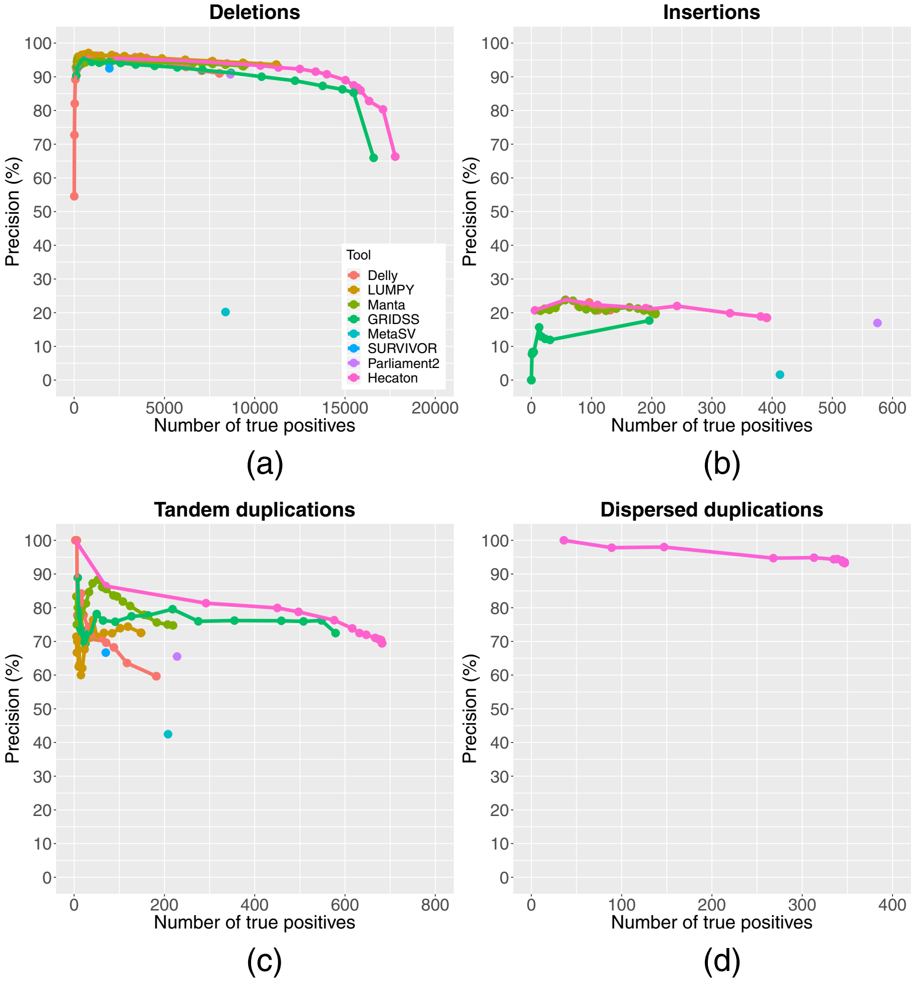

Fig. 5 Performance of different CNV detection algorithms on short read data of the maize B73 accession. CNV calls were called relative to the maize Mo17 reference assembly. Precision-recall curves of Delly, LUMPY, Manta, and GRIDSS were constructed by varying the minimum number of discordantly aligned read pairs and/or split reads supporting each call in the set. The curve of Hecaton was produced by varying the threshold of the probabilistic score used to define calls as true positives. Performance is shown separately for deletions (**a**), insertions (**b**), tandem duplications (**c**), and dispersed duplications (**d**). Curves of LUMPY and SURVIVOR are missing for insertions, as these tools are unable to detect this type of CNV. Curves are missing for all tools besides Hecaton for dispersed duplications, as Hecaton is the only tool that can detect this type of CNV, owing to its post-processing stage. Hecaton generally improved recall and precision compared to Delly, LUMPY, Manta, and GRIDSS. Moreover, it significantly outperformed MetaSV, SURVIVOR, and Parliament2, three ensemble approaches applicable to plant data

although Hecaton has no upper limit in terms of the size of CNV it can detect, we were not able to evaluate its performance on CNVs that were larger than 1 Mb, as such calls tended to be falsely validated by one of our validation methods, VaPoR (Additional file 1: Figure S6). Third, it is not able to detect insertions with both high recall and precision, a limitation it shares with other CNV detection tools designed to work with short WGS data [6]. Finally, we were not able to robustly assess the performance of Hecaton in polyploid plant species, as we could not find polyploid samples of which both short and long read data were publicly available. We expect that the performance of Hecaton on polyploids should be comparable to the performance reported in this work on diploids, if the polyploid sample does not show strong differences between haplotypes. To deal with additional biases found in more complex polyploid species, it may be worthwhile to obtain ground truth annotations in order to train random forest models specifically tailored to polyploids. Such annotations can be obtained by generating a small set of polyploid samples using both Illumina and PacBio sequencing platforms. Simulated data could serve as ground truth data as well, but we were unable to generate simulated CNVs that accurately represent the distribution of CNVs in real scenarios, a problem encountered in previous work that benchmarked CNV detection tools [8].

Hecaton provides a scalable method to detect CNVs in plant species

Hecaton scales well to crop genomes when using conventional computational server resources. By making extensive use of parallelization, it processed samples of both domesticated and wild tomatoes in reasonable time, taking a minimum of 7 h and a maximum of 40 h to process a single sample (Table 1), when using Hecaton with 13 cores (Intel® Xeon® CPU E5-2670 v3 @ 2.30 GHz) on a Linux server (Ubuntu 16.04). For comparison, it would have taken a minimum of 67 h and a maximum of 200 h to run Hecaton on a single core (Table 1).

Although we do not know the true CNVs present in these samples, we estimated that Hecaton attained lower precision in the tomato samples than in the *A. thaliana* and rice sets. We considered events to be likely false positives if they did not have a clear and uniformly lower (deletions) or higher (duplications) read depth compared to the rest of the chromosome its located on or to its flanking regions, or if they showed excessive (over 1000x) read coverage. Based on these criteria, we estimated that 30% of deletions, 20% of tandem duplications, and 80% of dispersed duplications in the wild accession LYC4 sample were false positives, based on inspection of a random sample of 20 CNVs of each type.

Additional filtering steps based on the median read depth of CNV calls and the presence of gaps in the regions flanking calls removed a significant number of CNVs from the callsets of both the domesticated and wild accessions (Table 2). However, they had little to no effect when applied to the callsets of *A. thaliana* Col-0–Cvi-0 and Suijing18 rice (Additional file 1: Figure S7). Therefore, Hecaton does not perform these steps by default. They are only meant to be used when working with samples that are distantly related to the reference genome or for which the reference genome assembly contains a significant number of gaps.

Hundreds to thousands of CNVs remained after performing additional filtering (Table 2), indicating that even a conservative, high confidence set of events (see Additional file 1: Figure S8 for an example of such an event) called by Hecaton can provide a sizable pool of genetic variation that can be further characterized. Given that the reference genome of tomato used in this work is based on a domesticated cultivar, an expectedly larger number of CNVs were found in the wild accessions (LA2157, LA0716, LYC4) than in the domesticated ones (PI158760, LA2706, TR00003, LA4451). Most of the CNVs between the tomato samples and the reference genome consisted of deletions (Table 3), following a similar trend as seen in the *A. thaliana* Col-0–Cvi-0, *A. thaliana* Ler, and Suijing18 rice samples. No insertions were reported for any of the samples (Table 3). We expect that this result does not reflect the actual underlying biology, but was rather caused by the stringent cut-off used to filter calls. Most events overlapped with repetitive elements (Table 4), which is not unexpected as low-complexity regions are thought to be one of the prime mediators of the formation of CNV [39]. A smaller, but non-negligible, fraction of CNVs overlapped with genes and coding sequences (Table 4), providing potential leads for CNV events having a functional or phenotypic impact.

Table 1 Hecaton running time and memory use

Sample	CPU time (h)	Real time (h)	Peak resident set size (Gb)
PI158760	119.5	13.1	27.8
LA2706	67.8	7.3	23.9
TR00003	96.2	11.1	24.3
LA4451	70.8	7.3	23.6
LA2157	196.1	32.5	26.6
LA0716	172.3	30.3	24.9
LYC4	189.1	39.2	25.7

Table 2 Number of CNVs detected in tomato samples before and after filtering

Sample	No filter	Read depth filter	Read depth and gap filter
PI158760	482	347	97
LA2706	934	701	403
TR00003	1487	1162	880
LA4451	1789	1482	1095
LA2157	7127	5372	4851
LA0716	10064	8472	7910
LYC4	11988	9950	9407

Conclusion

Hecaton is a computational workflow specifically designed to detect CNV from WGS data of plant genomes. It improves upon the performance of current approaches primarily developed for human genomes, indicating that such tools are less suitable to plant

Table 3 Types of CNV detected in tomato samples after filtering

Sample	Deletions	Insertions	Tandem duplications	Dispersed duplications	Total
PI158760	82	0	15	0	97
LA2706	295	0	106	2	403
TR00003	777	0	103	0	880
LA4451	901	0	193	1	1095
LA2157	4261	0	561	29	4851
LA0716	7168	0	712	30	7910
LYC4	8508	0	878	21	9407

Table 4 Overlap of filtered CNVs with repeats, genes, and coding sequences (CDS) in tomato samples

Sample	Sequence covered (Mb)	Repeats (% of total)	Genes (% of total)	CDS (% of total)
PI158760	0.21	64.9	12.38	2.32
LA2706	0.46	75.59	12.47	1.73
TR00003	1.59	84.01	9.38	1.89
LA4451	1.21	76.26	11.78	1.11
LA2157	8.59	82.79	11.01	1.25
LA0716	12.11	85.47	12.19	1.29
LYC4	12.13	81.53	13.41	1.04

data without optimization. In contrast to several state-of-the-art tools, Hecaton correctly detects dispersed duplications. Moreover, the random forest model employed by Hecaton improves upon current approaches in terms of recall and precision. Finally, the running time and memory usage of Hecaton scales well to crop genomes, demonstrating its practical utility to plant research.

We anticipate that Hecaton is of immediate interest to both applied and fundamental research regarding the relationship between genotype and phenotype in plants. CNVs have been linked with several stress-resistant phenotypes in crop species [40], including frost tolerance in wheat [41], aluminum tolerance in maize [42], and boron tolerance in barley [43]. Hecaton can extensively query crop and wild germplasm for resistant loci, that can be characterized and eventually introgressed into elite cultivars. Besides its use in agricultural research, Hecaton may help to answer more fundamental questions regarding the role of CNV, as many characteristics regarding the role of CNV in plant adaptation are still relatively unknown [44]. Such characteristics include how stress affects the rate at which CNVs accumulate, the main molecular mechanisms that govern the creation of CNVs, and the evolutionary dynamics that determine whether CNVs become fixed within a population. Populations of wild and domesticated plant species (such as the 100 Tomato Genome project [45]) may provide excellent datasets to explore these topics, given that domestication is a fairly recent phenomenon involving artificial selection for a set of well-defined and well-characterized traits.

Availability and requirements

Project name: Hecaton
Project home page: https://git.wur.nl/bioinformatics/hecaton
Operating system(s): Unix
Programming language: Python
Other requirements: Hecaton can be either installed locally or through a Docker image. All of its dependencies are listed on the project home page.
License: GNU AGPLv3
Any restrictions to use by non-academics: No

Supplementary information

Supplementary information accompanies this paper at https://doi.org/10.1186/s12864-019-6153-8.

Additional file 1: Figure S1: Performance of different CNV detection algorithms on the test set of Col-0–Cvi-0 and Suijing18 CNV events called from 10x coverage data. **Figure S2:** Interpretating the appropriate type of CNV from a set of novel adjacencies. **Figure S3:** Recall of the post-processing step of Hecaton for dispersed duplications simulated at different allele dosages in tetraploid tomato genomes, before and after post-processing. **Figure S4:** Performance of Hecaton on the test set containing Col-0–Cvi-0 and Suijing18 CNV events called from 10x coverage data, using random forest models trained on CNVs detected at different levels of sequencing coverage. **Figure S5:** Performance of different CNV detection algorithms on *A. thaliana Ler* data at 10x coverage. **Figure S6:** False positive events in simulated data that were incorrectly labeled as a true positive by VaPoR. **Figure S7:** Effect of filtering CNV calls based on read depth and presence of gaps in their flanking regions in the test set of Col-0–Cvi-0 and Suijing18 generated from 10x coverage data. **Figure S8:** Example of a high-confidence CNV called by Hecaton. **Figure S9:** Percentage of true events in simulated data incorrectly labeled as false positives by VaPoR. **Figure S10:** Percentage of true events in simulated data incorrectly labeled as false positives by Sniffles, in a set of 10 simulated versions of *S. lycopersicum* with distinct sets of CNVs. **Figure S11:** Percentage of false positive events in simulated data incorrectly labeled as a true positive by Sniffles, computed in a set of 10 simulated versions of *S. lycopersicum* with distinct sets of CNVs. **Figure S12:** Density plots of normalized median read depths of CNV events called in domesticated and wild tomato samples. **Figure S13:** Density plot of the fraction of N's in the 400 bp flanking regions of CNV events called in domesticated and wild tomato samples.

Additional file 2: Table S1: Features used for the random forest model. **Table S2:** Description of the used datasets. **Table S3:** Number of events called from B73 data that could not be validated by VaPoR. **Table S4:** Number of events simulated per size interval.

Additional file 3: Supplementary Methods: Evaluating the performance of CNV detection tools using simulated data. Evaluating the performance of CNV detection tools using real data. Applying Hecaton to tomato data.

Abbreviations

CNV: Copy number variation or copy number variant; SNP: Single nucleotide polymorphism; WGS: Whole genome sequencing

Acknowledgements

We thank Carlos de Lannoy and Mehmet Akdel for testing the installation of Hecaton.

Authors' contributions

RYW, SS, and DDR contributed to the design of Hecaton. RYW implemented Hecaton, evaluated its performance, and applied it to representative crop samples. RYW, SS, and DDR were major contributors in writing the manuscript. All authors read and approved the final manuscript.

Funding

This work was supported by the Netherlands Organisation of Scientific Research (NWO) under project grant ALWGR.2015.9. We thank the NWO and private partners Rijk Zwaan Breeding, Bejo Zaden, Genetwister Technologies, Averis Seeds, C. Meijer and HZPC Holland for their financial support through this grant. The funding bodies had no role in the design of the study; the collection, analysis, and interpretation of data; and in writing the manuscript.

Availability of data and material

To develop and evaluate Hecaton, we used publicly available short and long read datasets of an *A. thaliana* Col-0–Cvi-0 F1 hybrid [46], the *Japonica* rice variety Suijing18 [47], *A. thaliana* Landsberg *erecta* (*Ler*) [48], maize B73 [49], and domesticated and wild tomato accessions [45]. Additional file 2: Table S2 provides a full description of the datasets, including NCBI Short Read Archive accession numbers. The genomic assemblies and annotations of *A. thaliana* Col-0 (version TAIR10), *O. sativa Japonica* (version IRGSP-1.0) and maize Mo17 (Zm-Mo17-REFERENCE-CAU-1.0) can be respectively found at the European Nucleotide Archive under accession numbers GCA_000001735.2, GCA_001433935.1, and GCA_003185045.1. The genomic assembly (version SL3.0) and annotation (version ITAG3.20) of *S. lycopersicum* Heinz 1706 can be obtained from the FTP server of Sol Genomics Network (https://solgenomics.net/organism/Solanum_lycopersicum/genome).

Ethics approval and consent to participate

Not applicable.

Consent for publication

Not applicable.

Competing interests

The authors declare that they have no competing interests.

Received: 31 July 2019 Accepted: 30 September 2019
Published online: 07 November 2019

References

1. Zmieńko A, Samelak A, Kozłowski P, Figlerowicz M. Copy number polymorphism in plant genomes. Theoret Appl Genet. 2014;127(1):1–18.
2. Gaines TA, Zhang W, Wang D, Bukun B, Chisholm ST, Shaner DL, et al. Gene amplification confers glyphosate resistance in *Amaranthus palmeri*. Proc Nat Acad Sci. 2010;107(3):1029–34.
3. Gabur I, Chawla HS, Snowdon RJ, Parkin IA. Connecting genome structural variation with complex traits in crop plants. Theor Appl Genet. 2019;132(3):733–50.
4. Lye ZN, Purugganan MD. Copy Number Variation in Domestication. Trends Plant Sci. 2019;24(4):352–65.
5. Alkan C, Coe BP, Eichler EE. Genome structural variation discovery and genotyping. Nat Rev Genet. 2011;12(5):363–76.
6. Sedlazeck FJ, Rescheneder P, Smolka M, Fang H, Nattestad M, von Haeseler A, et al. Accurate detection of complex structural variations using single-molecule sequencing. Nat Methods. 2018;15(6):461–8.
7. De Coster W, De Roeck A, De Pooter T, D'hert S, De Rijk P, Strazisar M, et al. Structural variants identified by Oxford Nanopore PromethION sequencing of the human genome. Genome Res. 2019;29:1178–87.
8. Kosugi S, Momozawa Y, Liu X, Terao C, Kubo M, Kamatani Y. Comprehensive evaluation of structural variation detection algorithms for whole genome sequencing. Genome Biol. 2019;20:117.
9. Zhao X, Emery SB, Myers B, Kidd JM, Mills RE. Resolving complex structural genomic rearrangements using a randomized approach. Genome Biol. 2016;17(1):126.
10. Lisch D. How important are transposons for plant evolution? Nat Rev Genet. 2013;14(1):49.
11. Mills RE, Walter K, Stewart C, Handsaker RE, Chen K, Alkan C, et al. Mapping copy number variation by population-scale genome sequencing. Nature. 2011;470(7332):59.
12. Chaisson MJ, Sanders AD, Zhao X, Malhotra A, Porubsky D, Rausch T, et al. Multi-platform discovery of haplotype-resolved structural variation in human genomes. Nature Commun. 2019;10:1784.
13. Lee AY, Ewing AD, Ellrott K, Hu Y, Houlahan KE, Bare JC, et al. Combining accurate tumor genome simulation with crowdsourcing to benchmark somatic structural variant detection. Genome Biol. 2018;19:188.
14. Rausch T, Zichner T, Schlattl A, Stütz AM, Benes V, Korbel JO. DELLY: structural variant discovery by integrated paired-end and split-read analysis. Bioinformatics. 2012;28(18):i333–9.
15. Cameron DL, Schroeder J, Penington JS, Do H, Molania R, Dobrovic A, et al. GRIDSS: sensitive and specific genomic rearrangement detection using positional de Bruijn graph assembly. Genome Res. 2017;27:2050–60.
16. Layer RM, Chiang C, Quinlan AR, Hall IM. LUMPY: a probabilistic framework for structural variant discovery. Genome Biol. 2014;15(6):R84.
17. Chen X, Schulz-Trieglaff O, Shaw R, Barnes B, Schlesinger F, Källberg M, et al. Manta: rapid detection of structural variants and indels for germline and cancer sequencing applications. Bioinformatics. 2015;32(8):1220–2.
18. Abyzov A, Urban AE, Snyder M, Gerstein M, CNVnator: an approach to discover genotype and characterize typical and atypical CNVs from family and population genome sequencing. Genome Res. 2011;21(6):974–84.
19. Boeva V, Popova T, Bleakley K, Chiche P, Cappo J, Schleiermacher G, et al. Control-FREEC: a tool for assessing copy number and allelic content using next-generation sequencing data. Bioinformatics. 2011;28(3):423–5.
20. Ye K, Schulz MH, 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(21): 2865–71.
21. Di Tommaso P, Chatzou M, Floden EW, Barja PP, Palumbo E, Notredame C. Nextflow enables reproducible computational workflows. Nature Biotechnol. 2017;35(4):316.
22. Chiang C, Layer RM, Faust GG, Lindberg MR, Rose DB, Garrison EP, et al. SpeedSeq: ultra-fast personal genome analysis and interpretation. Nat Methods. 2015;12(10):966–8.
23. Li H. Aligning sequence reads, clone sequences and assembly contigs with BWA-MEM; 2013. Preprint at http://arxiv.org/abs/1207.3907. Accessed 23 July 2019.
24. Faust GG, Hall IM. SAMBLASTER: fast duplicate marking and structural variant read extraction. Bioinformatics. 2014;30(17):2503–5.
25. Tarasov A, Vilella AJ, Cuppen E, Nijman IJ, Prins P. Sambamba: fast processing of NGS alignment formats. Bioinformatics. 2015;31(12):2032–4.
26. Li H. seqtk, Toolkit for processing sequences in FASTA/Q formats; 2012. Available from:https://github.com/lh3/seqtk. Accessed 10th of August 2018.
27. Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, et al. Scikit-learn: Machine learning in Python. J Mach Learn Res. 2011;12(Oct):2825–30.
28. The Tomato Genome Consortium. The tomato genome sequence provides insights into fleshy fruit evolution. Nature. 2012;485(7400):635.
29. Zhao X, Weber AM, Mills RE. A recurrence-based approach for validating structural variation using long-read sequencing technology. GigaScience. 2017;6(8):gix061.
30. Lin K, Smit S, Bonnema G, Sanchez-Perez G, de Ridder D. Making the difference: integrating structural variation detection tools. Briefings in Bioinformatics. 2014;16(5):852–64.
31. Mohiyuddin M, Mu JC, Li J, Bani Asadi N, Gerstein MB, Abyzov A, et al. MetaSV: an accurate and integrative structural-variant caller for next generation sequencing. Bioinformatics. 2015;31(16):2741–4.
32. Jeffares DC, Jolly C, Hoti M, Speed D, Shaw L, Rallis C, et al. Transient structural variations have strong effects on quantitative traits and reproductive isolation in fission yeast. Nature Commun. 2017;8:14061.
33. Zarate S, Carroll A, Krashenina O, Sedlazeck FJ, Jun G, Salerno W, et al. Parliament2: fast structural variant calling using optimized combinations of callers. 2018. Preprint at https://www.biorxiv.org/content/10.1101/424267v1.abstract. Accessed 23 July 2019.
34. Sun S, Zhou Y, Chen J, Shi J, Zhao H, Zhao H, et al. Extensive intraspecific gene order and gene structural variations between Mo17 and other maize genomes. Nature Genet. 2018;50(9):1289.
35. Pounraja VK, Jayakar G, Jensen M, Kelkar N, Girirajan S. A machine-learning approach for accurate detection of copy-number variants from exome sequencing. Genome Res. 2019;29:1134–43.
36. Ewing AD, Houlahan KE, Hu Y, Ellrott K, Caloian C, Yamaguchi TN, et al. Combining tumor genome simulation with crowdsourcing to benchmark somatic single-nucleotide-variant detection. Nature Meth. 2015;12(7):623.
37. Marbach D, Costello JC, Küffner R, Vega NM, Prill RJ, Camacho DM, et al. Wisdom of crowds for robust gene network inference. Nature Meth. 2012;9(8):796.
38. Margolin AA, Bilal E, Huang E, Norman TC, Ottestad L, Mecham BH, et al. Systematic analysis of challenge-driven improvements in molecular prognostic models for breast cancer. Sci Transl Med. 2013;5(181):181re1.
39. Hastings PJ, Lupski JR, Rosenberg SM, Ira G. Mechanisms of change in gene copy number. Nature Rev Genet. 2009;10(8):551.
40. Mickelbart MV, Hasegawa PM, Bailey-Serres J. Genetic mechanisms of abiotic stress tolerance that translate to crop yield stability. Nature Rev Genet. 2015;16(4):237.

41. Würschum T, Longin CFH, Hahn V, Tucker MR, Leiser WL. Copy number variations of CBF genes at the Fr-A2 locus are essential components of winter hardiness in wheat. Plant J. 2017;89(4):764–73.
42. Maron LG, Guimarães CT, Kirst M, Albert PS, Birchler JA, Bradbury PJ, et al. Aluminum tolerance in maize is associated with higher MATE1 gene copy number. Proc Nat Acad Sci. 2013;110(13):5241–46.
43. Sutton T, Baumann U, Hayes J, Collins NC, Shi BJ, Schnurbusch T, et al. Boron-toxicity tolerance in barley arising from efflux transporter amplification. Science. 2007;318(5855):1446–9.
44. Gaut BS, Seymour DK, Liu Q, Zhou Y. Demography and its effects on genomic variation in crop domestication. Nature Plants. 2018;4:512–20.
45. Aflitos S, Schijlen E, de Jong H, de Ridder D, Smit S, Finkers R, et al. Exploring genetic variation in the tomato (*Solanum* section *Lycopersicon*) clade by whole-genome sequencing. Plant J. 2014;80(1):136–48.
46. Chin CS, Peluso P, Sedlazeck FJ, Nattestad M, Concepcion GT, Clum A, et al. Phased diploid genome assembly with single-molecule real-time sequencing. Nature Meth. 2016;13(12):1050.
47. Nie SJ, Liu YQ, Wang CC, Gao SW, Xu TT, Liu Q, et al. Assembly of an early-matured *japonica* (*Geng*) rice genome, Suijing18, based on PacBio and Illumina sequencing. Sci Data. 2017;4:170195.
48. Zapata L, Ding J, Willing EM, Hartwig B, Bezdan D, Jiao WB, et al. Chromosome-level assembly of *Arabidopsis thaliana Ler* reveals the extent of translocation and inversion polymorphisms. Proc Nat Acad Sci. 2016;113(28):E4052–60.
49. Jiao Y, Peluso P, Shi J, Liang T, Stitzer MC, Wang B, et al. Nature. 2017;546(7659):524. Improved maize reference genome with single-molecule technologies.

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.