Bioinformatics, 31(11), 2015, 1708–1715
doi: 10.1093/bioinformatics/btv070
Advance Access Publication Date: 1 February 2015
Original Paper

OXFORD

Genome analysis

CNOGpro: detection and quantification of CNVs in prokaryotic whole-genome sequencing data

Ola Brynildsrud[1,*], Lars-Gustav Snipen[2] and Jon Bohlin[3]

[1]Section for Biostatistics and Epidemiology, Norwegian University of Life Sciences (NMBU), Oslo, [2]Department of Chemistry, Biotechnology and Food Science, Norwegian University of Life Sciences (NMBU), Ås and [3]Norwegian Institute of Public Health, Division of Epidemiology, 0403 Oslo, Norway

*To whom correspondence should be addressed.

Associate Editor: Alfonso Valencia

Received on April 22, 2014; revised on January 7, 2015; accepted on January 28, 2015

Abstract

Motivation: The explosion of whole-genome sequencing (WGS) as a tool in the mapping and understanding of genomes has been accompanied by an equally massive report of tools and pipelines for the analysis of DNA copy number variation (CNV). Most currently available tools are designed specifically for human genomes, with comparatively little literature devoted to CNVs in prokaryotic organisms. However, there are several idiosyncrasies in prokaryotic WGS data. This work proposes a step-by-step approach for detection and quantification of copy number variants specifically aimed at prokaryotes.

Results: After aligning WGS reads to a reference genome, we count the individual reads in a sliding window and normalize these counts for bias introduced by differences in GC content. We then investigate the coverage in two fundamentally different ways: (i) Employing a Hidden Markov Model and (ii) by repeated sampling with replacement (bootstrapping) on each individual gene. The latter bypasses the complex problem of breakpoint determination. To demonstrate our method, we apply it to real and simulated WGS data and benchmark it against two popular methods for CNV detection. The proposed methodology will in some cases represent a significant jump in accuracy from other current methods.

Availability and implementation: *CNOGpro* is written entirely in the R programming language and is available from the CRAN repository (http://cran.r-project.org) under the GNU General Public License.

Contact: ola.brynildsrud@nmbu.no

Supplementary information: Supplementary data are available at *Bioinformatics* online.

1 Introduction

Copy number variation (CNV) is a type of structural variation that refers to any abnormality in the frequency at which a DNA sequence occurs in a genome. It is a critical component of the genetic variability of organisms (Alkan *et al.*, 2011). CNVs can include duplications (sometimes referred to as amplifications) or deletions of a particular stretch of sequence. Considerable research has been carried out on the relatively easy problem of determining whether a strain contains a gene or not, but less work has been devoted to the more complex problem of measuring non-zero variation in the gene copy number. The phenotypical implications of CNV are also less clear than those resulting from a functional deletion, although CNVs are known to be a source of important genetic variation in both humans (Stranger *et al.*, 2007) and bacteria (Riehle *et al.*, 2001). Research in humans has typically focused on the role of CNVs in cancer and inherited disease (Hastings *et al.*, 2009), and there is also evidence of adaptive duplications (Cooper *et al.*, 2007). Less work has been devoted to exploring the role of CNVs in prokaryotes. However, bacteria display substantial variation in gene copy numbers, and the extra cost of maintaining a redundant gene comes with the payoff of a selective

advantage under certain environmental conditions (Klappenbach *et al.*, 2000; Kondrashov, 2012).

The most common way of testing the hypothesis of no CNV is by examining the read counts (RCs) along the chromosome after alignment of the sequenced reads to a reference genome or de novo assembly. The argument is that the RC in any non-overlapping, equally sized bin can be considered as a stochastic variable with a particular probability distribution and that there is an inherent proportionality between the expected value (mean) of this variable and the underlying copy number (Alkan *et al.*, 2009; Campbell *et al.*, 2008; Medvedev *et al.*, 2009), analogous to principles of CNV estimation in array Comparative Genome Hybridization technology. However, RCs are affected by biases that must be corrected for before any valid conclusions can be made.

There are currently at least 48 available tools for the discovery of CNVs from next-generation sequencing data (for a detailed runthrough, see Zhao *et al.*, 2013). Of these, few attempt to quantify the number of copies of any particular chromosomal segment (Klambauer *et al.*, 2012). Furthermore, most of these tools are designed with a diploid, human setting in mind. This presents a number of problems when applying these tools to prokaryotes. Prokaryotes are organisms of indefinite ploidy, and there are some additional allowances on the valid copy number outcomes when compared with diploid genomes. Although for human genomes a copy number of 1.5 would mean that the segment in question had two copies on one chromosome but one on the other, in bacteria the interpretation would vary from species to species depending on the number of sets of chromosomes, both complete and incomplete, it maintains. Complicating matters even further, the copy number result may vary due to bacterial growth mechanics. In fact, any non-negative decimal copy number could make sense for bacteria, because bacteria growing under exponential growth conditions are able to replicate their chromosomes faster than they can divide (Pecoraro *et al.*, 2011). This is reflected in DNA sequencing data, with copy numbers representing a mixture of the discrete number of chromosomes that the bacterium maintains at the stationary, non-dividing phase and a fractional number that stems from the fact that cells are in different phases of binary fission cycle.

This article presents a tool for the discovery of CNVs from prokaryote-origin whole-genome sequencing (WGS) data. We have developed an R package called *CNOGpro*, which is an acronym for 'Copy Numbers of Genes in prokaryotes.' The main purpose of the tool is to quickly estimate the number of copies of any gene or intergenic segment in a resequencing experiment. *CNOGpro* supports rapid calling of copy number using a hidden Markov model and additionally allows for the construction of confidence intervals around copy number estimates by bootstrapping. Although several publicly available CNV analysis tools designed for work on human-origin data can, with varying amounts of tinkering, also accept prokaryote data, to our knowledge no existing tool for CNV analysis focuses specifically on prokaryote data.

2 Materials and methods

2.1 Data preparation

The first step of RC-based CNV discovery consists of quality control of the sequencing data, including filtering of poor and uninformative data, thereafter mapping the reads onto the backbone of some related genome. [If the reference and the test organisms are too distantly related, we are introducing bias (Nijkamp *et al.*, 2012).] The filtering should be informed by such parameters as total coverage, average quality of reads, frequency of ambiguous characters (e.g. 'N') and quality distribution according to read length [Easily checked with programs such as FASTQC (Babraham) Bioinformatics. http://bioinformatics.babraham.ac.uk/projects/fastqc]. It is especially important to remove polymerase chain reaction (PCR) duplicates introduced in the sequencing, easily performed by the rmdup command in samtools (Li *et al.*, 2009) or the DupRecover script of Zhou *et al.* (2014). As for aligning reads from the resequencing experiment to a reference sequence, Bowtie (Langmead *et al.*, 2009) or Maq (http://maq.sourceforge.net) are both reliable and recommended third-party software solutions. In this article, we use a complete genome, but *CNOGpro* should also accept draft genomes as reference, as long as they adhere to the GenBank flat file format and are parsed contig-wise into the pipeline.

Next, one needs to apply some counting scheme on the coverage metrics. For independent observations in a number series, we can only count each read once. One way of doing this is by counting each read at its leftmost end, i.e. the lowest chromosomal coordinate to which the read maps. This information is available in the default output for the SAMtools binary alignment/map (.bam) format and can be used to create best-hit read location files. (Details about this procedure can be found in *CNOGpro's* manual, provided in the R package.) Figure 1 shows the workflow of *CNOGpro*'s methods.

2.2 Counting reads

RCs are then made in neighboring, non-overlapping windows. (Referred to as RCs, observations or coverage interchangeably in the following chapters.) Each count represents the number of reads that have start points within that respective window. The average coverage and the desired sensitivity/specificity ratio should determine the window size. An average number of reads equal to 20–30 in each window is appropriate (Abyzov *et al.*, 2011). We have noted only a very slight drop in specificity when the average coverage is lowered from 100× to 20× and a more moderate drop when coverage is lowered to 10× (respective numbers: 99.96%, 99.7% and 99.0%; see Supplementary Table ST3). Long windows will make sure few false-positive CNVs are called, but one may also miss local coverage variation, which could be suggestive of small CNVs. The reverse is true for short windows. As a rule of thumb, shortening window length improves sensitivity at only marginal specificity cost and improving average coverage increases specificity and may increase sensitivity

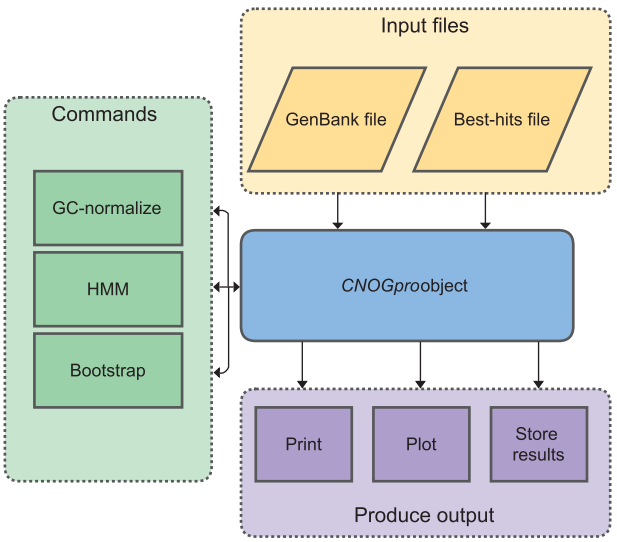

Fig. 1. Workflow diagram of *CNOGpro*

slightly. This is at least valid within window ranges 30–200 and coverage ranges 10×–1000×.

2.3 Biases and normalization

If the null hypothesis of no CNV between the reference organism and the data was true and there were no sources of bias, the expected RC in each window throughout the chromosome could be considered as independent and identically distributed random variables. Because of a number of known and unknown biases that influence the coverage, this is not the case. Therefore, before statistical inference is attempted, normalization must be performed on the RC metric. With flawless normalization, only the underlying CNV number of a genomic segment determines the expected RC value in a window. [The term segment is hereafter used to describe any continuous stretch of DNA with associated RC observations, parsed according to genes (including RNA genes) and intergenic stretches in the reference organism.]

Significant sources of bias in RC observations are local GC content of the chromosome (Dohm *et al.*, 2008), GC content of the sequencing probes (Diskin *et al.*, 2008), genomic mappability (Derrien *et al.*, 2012; Lee and Schatz, 2012) and copy number bias stemming from the fact that cells are in different stages of the replication cycle (Skovgaard *et al.*, 2011; Zomer *et al.*, 2012). On top of this, one should expect to see batch-specific effects from different platforms, runs and laboratories (Khrameeva and Gelfand, 2012).

GC content has a major effect on coverage on all current sequencing platforms (Aird *et al.*, 2011; Dohm *et al.*, 2008). The bias is thought to be a PCR-related sequencing artifact, arising especially during the library preparation step. *CNOGpro* uses Charif and Lobry's R package SeqinR (Charif and Lobry, 2007) to calculate the local GC content in each window. Normalization is done in accordance with the method proposed by Yoon *et al.* (2009), who suggested calculating median RCs for windows with GC content $(0, 1, 2, \ldots, 100\%)$ and weighting the individual RC of each window i with the overall median RC (mRC) divided by the median RC corresponding to that window's GC content mRC_{GCi}.

$$\mathrm{RC}_i^{\mathrm{corr}} = \mathrm{RC}_i \times \frac{\mathrm{mRC}}{\mathrm{mRC}_{\mathrm{GCi}}} \tag{1}$$

2.4 CNV detection and quantification

Using graphical techniques, it is simple to detect areas of the chromosome where the median normalized RCs deviates from the juxtapositional ones. CNVs are identified as significantly increased or decreased RCs over multiple consecutive windows. Other methods recommend the application of a segmentation algorithm to determine the breakpoints at where the coverage changes at this stage. However, this approach discards important information such as the starts and stops of individual genes. Typically in re-sequencing experiments, this information is available from the reference genome. We therefore propose a method wherein the coverage of each individual gene (or intergenic segment) is modeled separately. This has a number of advantages: first, the problem of breakpoint determination disappears. (An assumption here is that duplications and deletions work at the gene level.) Second, genomic wave patterns will not be called as CNVs, because these signals will cancel each other out, and the mean coverage will be an unbiased estimator of the true copy number (assuming that there are a certain number of independent and identically distributed observations within each gene). Third, we are not as concerned with outliers and observations taking extreme values just by chance. The RC observation should behave as a stochastic variable following a distribution in the Poisson family, usually with a certain degree of overdispersion (Sepúlveda *et al.*, 2013; Smith *et al.*, 2008) due to uncorrected biases. In segmentation algorithms, extreme observations could influence the breakpoint estimation, but in our method such observations are attributed to random variation of the stochastic variable.

2.4.1 Bootstrapping approach to gene-wise copy number estimates

Having assigned each RC observation to a gene or intergenic region, we attempt to estimate the copy number of each individual segment. According to our assumptions, observations in the same segment represent samples from a single underlying count distribution belonging to the Poisson family. Regardless of overdispersion, the expected value of any such distribution is equal to a rate parameter λ, also known as the mean. The mean of the observed RCs is representative of the underlying distribution from which the observations are drawn. *CNOGpro* computes this parameter for each gene or intergenic region by simply taking the mean of that region's associated observations. Confidence intervals around the estimate are constructed by repeated sampling with replacement (bootstrapping) of the associated observations. The means directly reveal the underlying copy number if they are subsequently normalized to be expressed as a multiple of what we know is the 'true' mean for non-CNV regions. This leaves us with one problem: How do we determine which regions are non-CNV prior to any copy number inferences? We will suggest two possible approaches: (i) using a priori information on gene copy numbers or (ii) using the method implemented in *CNOGpro*'s hidden Markov model method.

2.4.2 Hidden Markov Model

CNOGpro also implements the Viterbi algorithm (Viterbi, 1967) for detecting the most probable sequence of copy number states in the input (normalized) RC data. In brief, an emission matrix is created wherein the log probabilities of each possible RC token in each possible copy number state are stored. The probabilities in each state are taken from negative binomial probability distributions (commonly used for overdispersed count data) whose parameters are indirectly inferred by sampling from the input RC data. First, RCs are sampled from the input data to establish the mean and variance. The most common results are averaged to represent the true mean and variance of the counts of segments with copy number equal to one, building upon the assumption that most genomic segments in a resequencing assembly to a reference organism will not be duplicated or deleted. The sampling distribution is parameterized as a negative binomial distribution with parameters p and r by solving the following equations, inherent to the negative binomial distribution, with respect to p and r:

$$\mathrm{mean} = \frac{\mathrm{pr}}{(1-\mathrm{p})} \tag{2}$$

$$\mathrm{var} = \frac{\mathrm{pr}}{(1-\mathrm{p})^2} \tag{3}$$

The probability distribution of each possible state over the various output tokens (represented as $k = \{0, 1, 2 \ldots\}$ in the following) are calculated with the following formula:

$$\Pr(X = k) = \binom{k+r-1}{k} p^k (1-p)^r \tag{4}$$

To prevent arithmetic overflow in normal computer systems, the binomial coefficient in (4) needs to be calculated using the alternative formulation using the gamma function:

$$\Pr(X = k) = \frac{\Gamma(k+r)}{\Gamma(k+1)\Gamma(r)} p^k (1-p)^r \tag{5}$$

which can be further modified by exponentiation of the logarithmic expression:

$$\Pr(X = k) = e^{(\ln\Gamma(r+k)-\ln\Gamma(k+1)-\ln\Gamma(r)+x\ln(p)+r\log(1-p))} \quad (6)$$

CNOGpro assumes that the mean and variance scales linearly with copy number. For the special case of a deletion, i.e. where the copy number is zero, the probabilities are taken from a geometric distribution with the parameter p representing the rate of erroneously mapped reads:

$$\Pr(X = k) = (1 - p)^k p \quad (7)$$

The second parameter in the Hidden Markov model (HMM), the transition matrix, holds log-probabilities of switching between the different possible states in the chain. *CNOGpro* currently only accepts a single subparameter as input to the transition matrix, namely the probability q of switching states. The probability of remaining in the same state is calculated as $1 - q$, whereas all remaining transitions are considered equally probable and share the probability q between them. A Viterbi path is then calculated as the most likely sequence of states in the chain, each step being only dependent on the one immediately before it as well as the emission and transition matrices.

The user is allowed control of the most important parameters such as the number of states to include, the probability of changing states and the fraction of erroneously mapped reads. Figure 2 presents a closer look at the relation between the HMM and bootstrap methods.

3 Results

3.1 Application to WGS data

To demonstrate its utilities, we tested *CNOGpro* on WGS data from *Staphylococcus aureus* TW20, the details of which can be found in the Supplementary File SD1. Results can be seen in Supplementary Table ST1 and isolate data in Supplementary Table ST2.

3.2 Considerations on sensitivity and specificity

To validate our protocol, we used the ART sequencing simulator of Huang *et al.* (2011), to create simulated datasets of the Illumina paired-end protocol. The sequence data were created with FN433596 as a template. We set the parameters of the simulator to best approximate the real sequence data we had used; average coverage was set to 100 and read length to 76×2 nt with an insert size of 500 and a standard deviation of 100. We discarded the first and final 500 nucleotides in the assembly, because to our knowledge, ART does not support circular chromosomes, leading to no representation of reads across the origin. To test sensitivity (ratio of the number of true positives to the number of positives), we introduced 30 CNVs by deleting or duplicating randomly chosen ORFs or intergenic segments in FN433596, as shown in Supplementary Table ST3. A random number generator chose the segments and corresponding CNV levels. To test the specificity (ratio of the number of true negatives to the number of negatives), we also used FN433596 directly (i.e. with no introduced CNVs). Non-mapping and

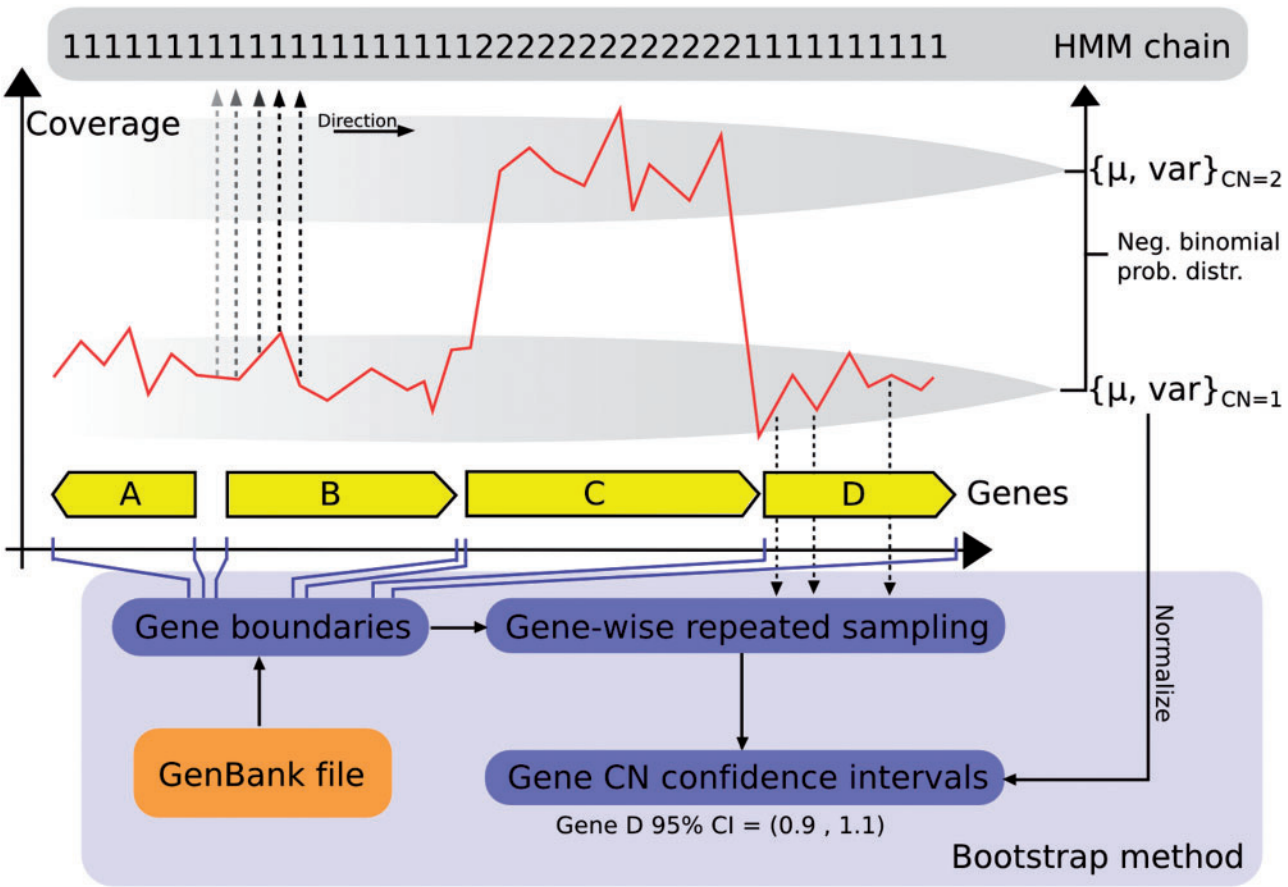

Fig. 2. The relation between *CNOGpro*'s principal methods of CNV quantification. Hidden Markov modeling and gene-wise bootstrapping, as well as their relation to the boundaries of genomic features needed in the analysis

low-quality reads (average PHRED score < 20) were filtered out before alignment, and reads were aligned with Maq (Version 0.7.1. Available at http://maq.sourceforge.net) against the reference sequence. Only reads that mapped in pairs were kept, and in cases where reads mapped to multiple possible sites, one was chosen at random.

Our algorithm called no CNVs for the non-altered sequence data, indicating a specificity of 100%. For the data with CNVs introduced, we encountered two false-positive calls: one in an IS256 region between coordinates 1 955 731 and 1 956 099, a repeat region with many occurrences in the chromosome, which would suggest that it was called because of a mapping problem and the other in a 230 bp intergenic region between coordinates 2 613 756 and 2 613 985. With a total of 5437 true-negative segments, this points to a specificity of 99.96%.

As expected, our algorithm performed poorly for CNV regions with a length of less than 100 bp, our window length for this analysis, calling only 2 out of 10 regions. All CNVs with a length >100 bp were detected, and of these, 11 out of 20 were called with the correct copy number using bootstrapping (true copy number represented in a 99% confidence interval), whereas the HMM called five CNVs correctly and unambiguously and 14 partly correct, meaning that the correct copy number were among the suggested solutions. (Because the HMM calculates breakpoints independently of gene starts and stops, gene-wise results from the HMM tend to reflect mixtures of copy numbers.) There was a strong correlation between CNV length and the accuracy of the copy number estimate. The results point to an overall sensitivity of 73% but significantly higher for longer CNV regions. Our quantification algorithms are less accurate, with 5/30 and 11/30 correctly called CNVs for the HMM and bootstrap approaches, respectively. Altogether, 19/30 regions were quantified with the correct or partly correct copy number. However, we note that simulated data do not suffer from identical biases as those introduced in real sequencing data.

We also benchmarked our algorithm against those of cnv-seq (Xie and Tammi, 2009) and cn.MOPS (Klambauer *et al.*, 2012), two current standards in CNV analysis. Similar to our algorithm, neither cnv-seq nor cn.MOPS called any false positives. They did, however, perform considerably worse than our algorithm when it comes to sensitivity. Cnv-seq detected 14 out of the 30 CNV regions, which would indicate a sensitivity of 46% for this dataset. cn.MOPS correctly detected seven CNV regions and indicated the correct copy number for four of these, which would result in a sensitivity of 23% for this dataset. It must be noted here that cn.MOPS is designed to accept more than two samples as an input, and so the results might differ if we had included additional samples. Results from these analyses can be found in Supplementary Table ST3 and are visualized in Figure 3.

We additionally redid the above analyses in *CNOGpro* while letting the window length vary between 30 and 200 nt. At an average coverage of 100×, there was a trend of increased sensitivity with the lower window lengths (17/30 = 57% for window size 30 versus 12/30 = 40% for window size 200) without any apparent loss in specificity. Dropping the coverage to 20× did not impact sensitivity at all but led to a slightly higher rate of false-positive CNV calls (16/5437, Sp = 99.7%). At an average coverage of 10, sensitivity is 43% (13/30) and specificity 99.0% (5386/5437). In summary, it is possible to call CNVs even from runs with an average coverage as low as 10×, and in fact, there is only a moderate drop in sensitivity when moving from 100× to 10× coverage.

4 Discussion

4.1 CNV calling in prokaryotes versus in humans

The departures from CNV calling problems on human data are multifold. First, the levels of non-coding and repetitive DNA are much lower in prokaryotes. Consequently, genomic mappability (Lee and Schatz, 2012, Derrien *et al.*, 2012) is of diminished importance as a source of bias when compared with eukaryote data. The bias does not seem prominent for the Illumina GA platform, and normalizing may in fact introduce *more* bias to the RCs than what was already there [Evident from the results of Magi *et al.* (2011).] We, therefore, suggest that correcting for this bias is unnecessary for most prokaryotic sequencing experiments, at least those sequenced on Illumina GA machines. Second, the sizes of prokaryotic genomes are a fraction of eukaryotic ones. This relation typically translates to CNV regions as well, with regions being up to several kilobases long. This is the lower threshold of the range of known human CNVs, which can be many megabases long (Redon *et al.*, 2006). *CNOGpro* investigates coverage on a gene-by-gene basis, providing a higher sensitivity to detect short CNV regions. Third, although human-origin data are nearly always from diploid cell types, prokaryotes can have a varying number of copies of each chromosome, as well as plasmids, as well as partial copies and chimeras. This affects copy number quantification attempts by algorithms designed with a diploid setting in mind. Note, however, that the nature of cell replication and division in exponentially growing prokaryotes allows many copies of the chromosome to pile up in a single cell (Pecoraro *et al.*, 2011). Origin-proximal regions are often the most amplified in this growing phase, with a downward sloping gradient towards the more distant parts of the genome (Chao *et al.*, 2013; Gallagher *et al.*, 2011; Skovgaard *et al.*, 2011; Zomer *et al.*, 2012). The result is that one will often find non-integer copy numbers of a gene, which actually represents a mixture of the base copy number and the consensus of the replication cycles of sequenced cells.

4.2 Strengths and weaknesses

The primary strength of *CNOGpro* lies in its ability to inform the user of copy number called through two fundamentally different approaches: through gene-wise bootstrapping of RCs and by cycling through RCs chromosome-wise in an HMM. The former allows high sensitivity, whereas the latter is faster and only returns integer results. Both methods have weaknesses that can be alleviated by the other. For example, consider the fact that genomic coverage usually follows wave-like patterns, the causes and significance of which are not known, although it is known that the pattern correlates with GC content, not only of the genomic region but also of the probes used in the sequencing (Diskin *et al.*, 2008). This wave pattern confound methods that use segmentation algorithms to try to properly assign breakpoints, such as HMM-based approaches, but when each gene's coverage is investigated individually, this becomes less of a problem because the tips of the waves cancel each other out while the median, on average, remains more or less the same. We tested this in our simulated TW20 data multiplying observed coverage by a sine function with a 1000-bp period (roughly the size of an average gene), allowing the amplitude to randomly vary between periods, but with a maximum of 30% of the mean coverage. This changed the copy number calls for two segments out of the tested 5466, indicating that it had little overall impact in the analysis.

Using the bootstrapping method, it is also possible to predict CNV in genes that are present in multiple copies in the reference. If, for example, the reference has two copies of a gene and we find that our test organism presents with a bootstrap copy number result of

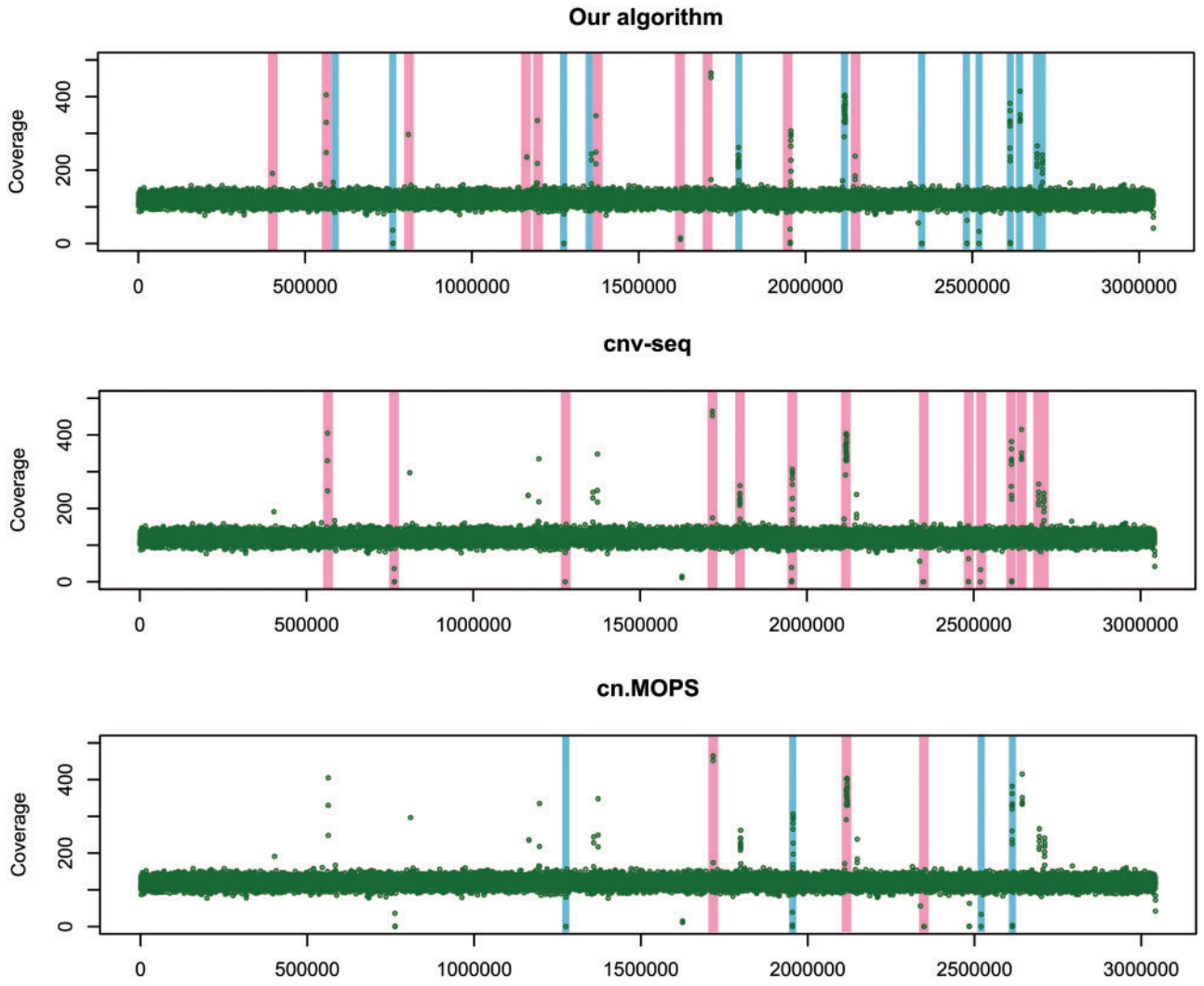

Fig. 3. Comparison of the present algorithm, cnv-seq and cn.MOPS in detecting and quantifying CNVs in simulated data with artificially introduced CNVs. Red color indicates that the algorithm correctly detected a CNV region, but that it was either not quantified (as in cnv-seq) or it was quantified incorrectly. Segments that were quantified correctly are highlighted in blue. The *x*-axes correspond to genomic position

1.5, we would expect the test organism to host three copies of the gene in question. Conversely, the expected bootstrap copy number result in a situation of 'de-duplication' where the reference has three copies and the test organism 2 would be 0.67. It is helpful to have knowledge of multi-copy occurrences of genes in the reference, because such information may help distinguishing genuine CNV from noise. In any case, the sensitivity towards CNV in multi-copy genes will be somewhat lower than what we have estimated in the previous section because such variation have lower signal intensities than variation in single-copy genes.

CNOGpro only compares coverage internally in a chromosome. This is helpful because significant biases have been demonstrated between identical isolates sequenced at different laboratories, on different machines and even with slightly differing protocols (Aird *et al.*, 2011; Khrameeva and Gelfand, 2012). [There are other potential causes of bias that we have not accounted for that may also play a part. For example, supercoiling of the genome has been shown to affect transcription under *in vivo* conditions (Pruss and Drlica, 1989).]

A few weaknesses with the gene-by-gene method of analyzing CNVs must also be noted: First, we will not discover intragenic duplications or deletions such as, for example, the duplication of some intragenic sequence motif. (However, the HMM model might still discover it if the duplication signal is strong. In this case, multiple possible copy numbers will be suggested for the gene in question.) Second, for very small genes, we may not have the required amount of RC observations to reject the null hypothesis of no variation in copy number, even if the gene is in fact present in more or less copies than in the reference sample. These weaknesses underscore the need for careful inspection of the data and the generous application of graphical methods for control.

4.3 Conclusion

We have presented a simple, quick and effective tool for detecting and quantifying CNVs in WGS data of prokaryotic organisms. When comparing similar prokaryotic genomes where details about the genomic layout in the reference are available, it represents a considerable jump in accuracy over other methods. It additionally has functions for creating high-quality informative plots and figures, an example of which can be seen in Figure 4. Our method starts with WGS data in the SAM format. Data are easily accessible through the Sequence Read Archive (SRA) (https://www.ncbi.nlm.nih.gov/Traces/sra), from

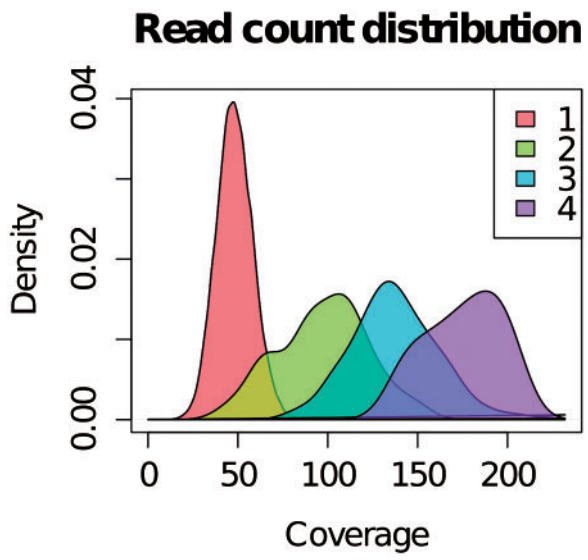

Fig. 4. Output of *CNOGpro*'s plot method, showing density curves of read count observations partitioned according to assigned copy number state

where one can also download the SRA toolkit and create SAM-format files from sequencing experiments. There are probably datasets from thousands of sequencing experiments freely available in the different sequence banks, just waiting for someone to analyze the clues that have been hidden in the frequencies with which each sequence occurs. *CNOGpro* is written entirely in the R programming language and is freely available under the GNU public license GPL-2. We believe it will be a valuable addition to the toolbox of every researcher conducting resequencing experiments to study copy number variance.

Acknowledgements

We thank the Computational Life Science initiative at the University of Oslo for helpful comments related to the analysis. We would also like to thank the reviewers of this article for suggestions to improvements and in particular the encouragement to publish this algorithm as an R package.

Funding

This work was funded by the Norwegian University of Life Sciences (NMBU).

Conflict of Interest: none declared.

References

Abyzov,A. *et al.* (2011) CNVnator: an approach to discover, genotype, and characterize typical and atypical CNVs from family and population genome sequencing. *Genome Res.*, **21**, 974–984.

Aird,D. *et al.* (2011) Analyzing and minimizing PCR amplification bias in Illumina sequencing libraries. *Genome Biol.*, **12**, R18.

Alkan,C. *et al.* (2009) Personalized copy number and segmental duplication maps using next-generation sequencing. *Nat. Genet.*, **41**, 1061–1067.

Alkan,C. *et al.* (2011) Genome structural variation discovery and genotyping. *Nat. Rev. Genet.*, **12**, 363–376.

Campbell,P.J. *et al.* (2008) Identification of somatically acquired rearrangements in cancer using genome-wide massively parallel paired-end sequencing. *Nat. Genet.*, **40**, 722–729.

Chao,M.C. *et al.* (2013) High-resolution definition of the Vibrio cholerae essential gene set with hidden Markov model-based analyses of transposon-insertion sequencing data. *Nucleic Acids Res*, 1–16.

Charif,D. and Lobry,J.R. (2007) SeqinR 1.0-2: a contributed package to the r project for statistical computing devoted to biological sequences retrieval and analysis. In: D.U.,Bastolla *et al* (eds.) *Structural Approaches to Sequence Evolution, Biological and Medical Physics, Biomedical Engineering*. Springer-Verlag, Berlin, Germany, pp. 207–232.

Cooper,G.M. *et al.* (2007) Mutational and selective effects on copy-number variants in the human genome. *Nat. Genet.*, **39**, S22–S29.

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

Diskin,S.J. *et al.* (2008) Adjustment of genomic waves in signal intensities from whole-genome SNP genotyping platforms. *Nucleic Acids Res.*, **36**, e126.

Dohm,J.C. *et al.* (2008) Substantial biases in ultra-short read data sets from high-throughput DNA sequencing. *Nucleic Acids Res.*, **36**, e105.

Gallagher,L.A. *et al.* (2011) Genome-scale identification of resistance functions in *Pseudomonas aeruginosa* using Tn-seq. *mBio*, **2**, e00315–e00310.

Hastings,P.J. *et al.* (2009) Mechanisms of change in gene copy number. *Nat. Rev. Genet.*, **10**, 551–564.

Huang,W. *et al.* (2011) ART: a next-generation sequencing read simulator. *Bioinformatics*, **28**, 593–594.

Khrameeva,E.E. and Gelfand,M.S. (2012) Biases in read coverage demonstrated by interlaboratory and interplatform comparison of 117 mRNA and genome sequencing experiments. *BMC Bioinformatics*, **13**, S4.

Klambauer,G. *et al.* (2012) cn.MOPS: mixture of Poissons for discovering copy number variations in next-generation sequencing data with a low false discovery rate. *Nucleic Acids Res.*, **40**, e69.

Klappenbach,J.A. *et al.* (2000) rRNA operon copy number reflects ecological strategies of bacteria. *Appl. Environ. Microbiol.*, **66**, 1328–1333.

Kondrashov,F.A. (2012) Gene duplication as a mechanism of genomic adaptation to a changing environment. *Proc. R. Soc. B Biol. Sci.*, **279**, 5048–5057.

Langmead,B. *et al.* (2009) Ultrafast and memory-efficient alignment of short DNA sequences to the human genome. *Genome Biol.*, **10**, R25.

Lee,H. and Schatz,M.C. (2012) Genomic dark matter: the reliability of short read mapping illustrated by the genome mappability score. *Bioinformatics*, **28**, 2097–2105.

Li,H. *et al.* (2009) The Sequence Alignment/Map format and SAMtools. *Bioinformatics*, **25**, 2078–2079.

Magi,A. *et al.* (2011) Read count approach for DNA copy number variants detection. *Bioinformatics*, **28**, 470–478.

Medvedev,P. *et al.* (2009) Computational methods for discovering structural variation with next-generation sequencing. *Nat. Methods*, **6**, S13–S20.

Nijkamp,J.F. *et al.* (2012) De novo detection of copy number variation by co-assembly. *Bioinformatics*, **28**, 3195–3202.

Pecoraro,V. *et al.* (2011) Quantification of ploidy in proteobacteria revealed the existence of monoploid, (mero-)oligoploid and polyploid species. *PLoS One*, **6**, e16392.

Pruss,G.J. and Drlica,K. (1989) DNA supercoiling and prokaryotic transcription. *Cell*, **56**, 521–523.

Redon,R. *et al.* (2006) Global variation in copy number in the human genome. *Nature*, **444**, 444–454.

Riehle,M.M. *et al.* (2001) Genetic architecture of thermal adaptation in *Escherichia coli*. *Proc. Natl Acad. Sci. USA*, **98**, 525–530.

Sepúlveda,N. *et al.* (2013) A Poisson hierarchical modelling approach to detecting copy number variation in sequence coverage data. *BMC Genomics*, **14**, 128.

Skovgaard,O. *et al.* (2011) Genome-wide detection of chromosomal rearrangements, indels, and mutations in circular chromosomes by short read sequencing. *Genome Res.*, **21**, 1388–1393.

Smith,D.R. *et al.* (2008) Rapid whole-genome mutational profiling using next-generation sequencing technologies. *Genome Res.*, **18**, 1638–1642.

Stranger,B.E. *et al.* (2007) Relative impact of nucleotide and copy number variation on gene expression phenotypes. *Science*, **315**, 848–853.

Viterbi,A. (1967) Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Trans. Inf. Theory*, **13**, 260–269.

Xie,C. and Tammi,M.T. (2009) CNV-seq, a new method to detect copy number variation using high-throughput sequencing. *BMC Bioinformatics*, **10**, 80.

Yoon,S. *et al.* (2009) Sensitive and accurate detection of copy number variants using read depth of coverage. *Genome Res.*, **19**, 1586–1592.

Zhao,M. *et al.* (2013) Computational tools for copy number variation (CNV) detection using next-generation sequencing data: features and perspectives. *BMC Bioinformatics*, **14**, S1.

Zhou,W. *et al.* (2014) Bias from removing read duplication in ultra-deep sequencing experiments. *Bioinformatics*, 1073–1080.

Zomer,A. *et al.* (2012) ESSENTIALS: software for rapid analysis of high throughput transposon insertion sequencing data. *PLoS One*, **7**, e43012.