BRNO UNIVERSITY OF TECHNOLOGY
VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
FAKULTA ELEKTROTECHNIKY
A KOMUNIKAČNÍCH TECHNOLOGIÍ

DEPARTMENT OF BIOMEDICAL ENGINEERING
ÚSTAV BIOMEDICÍNSKÉHO INŽENÝRSTVÍ

ADVANCED COMPUTATIONAL METHODS FOR CNV DETECTION IN BACTERIAL GENOMES
POKROČILÉ VÝPOČETNÍ METODY PRO DETEKCI CNV V BAKTERIÁLNÍCH GENOMECH

SUMMARY OF DOCTORAL THESIS
TEZE DIZERTAČNÍ PRÁCE

AUTOR PRÁCE AUTHOR	Ing. Robin Jugas
ŠKOLITEL SUPERVISOR	Ing. Helena Škutková, Ph.D.

BRNO 2023

Abstract

The focus in the field of structural variations is mainly focused on human genomes. Thus, detecting copy number variation (CNV) in bacteria is a less developed field. Commonly used CNV detection methods do not consider the features of bacterial circular genomes and generally, there is a space to improve performance metrics. This thesis presents a CNV detection method called CNproScan focused on bacterial genomes. CNproScan implements a hybrid approach combining read depth and read pair signals. It considers all bacteria features and depends only on NGS data. Based on the benchmarking results, the CNproScan achieved very well in various conditions. Using the read pair information, the CNVs are classified into several categories. Also, compared with other methods, CNproScan can detect much shorter CNV events. Because of the necessity of merging not only the various feature signals but also the results of different algorithms, the thesis also introduces a pipeline called ProcaryaSV developed to easily employ five CNV detection tools and merge their results. ProcaryaSV handles the whole procedure from quality check, reads trimming, and alignment to the CNV calling.

Keywords

Next-generation sequencing, Structural variation, Copy number variation, Bacteria

Contents

Introduction 4

1. Structural Variations 6

1.1 Classification 6

1.2 Structural Variations in Bacteria 7

2. Detection of Structural Variants 8

2.1 Using single approach 8

2.2 Limitations of using a single approach 12

3. Objectives of the thesis 14

4. CNproScan 15

4.1 The Algorithm Design 15

4.1.1 Data preparation 15

4.1.2 Main function 16

4.1.3 Biases Normalization 17

4.1.4 Outliers as CNV candidates 18

4.1.5 Extending CNV boundaries 19

4.1.6 Read-pair information 20

4.2 Benchmarking 21

4.2.1 Results for coverage 100× 22

4.2.2 Results for coverage 10, 20, 200× 23

4.2.3 CNV length analysis 25

5. ProcaryaSV 27

5.1 Pipeline Design 27

5.2 Merging algorithm 28

5.3 Benchmarking 30

Conclusion 33

References 35

Curriculum vitae 40

List of own publications 41

INTRODUCTION

The topic of this thesis is the detection of copy number variations in bacterial genomes. The copy number variations (CNVs) are a subgroup of a large field of structural variations (SVs). The structural variations are largely studied, yet there are still many gaps in the knowledge about them. This is even more factual for structural variations in bacteria. Despite that the first gene amplification was observed in Escherichia coli back then in 1963, this field of research is less developed in bacteria compared to the advancements in human or other eukaryotic genomes.

However, CNVs play an important role in the bacteria. They have a direct impact on protein production. In the long term, this has an impact on evolution and specialization. The short-term adaptive gene duplication can cause antibiotic resistance, which is an emerging issue.

Sequencing is a common way how to study these organisms and became substantially cheap. Two ways of sequencing bacterial genomes are being done. The sequencing of bacterial isolates or whole bacterial communities. The thesis deals with the first one as it enables the detection of structural changes in the genome such as copy number variations.

The lesser attention paid to structural variations in bacteria could be partly caused by technical difficulties detecting small rearrangements with short-read sequencing. Right now, we are at the breaking point between the massively used short-read next-generation sequencing, and the long-read third-generation sequencing. However, the inertia in the field is large and the next-generation sequencers are abundantly present and used in the labs. Furthermore, next-generation sequencing produces high throughput data necessary for copy number detection.

Firstly, I define briefly what structural variations and copy number variations are. The specifics of bacterial genomes are described in a special subchapter.

In the second chapter, I describe the field of detection of structural variations. This chapter is focused on bioinformatical aspects of structural variations detection.

The practical part of the thesis follows in two chapters. The first one is a novel algorithm for CNV detection named CNproScan. There were several reasons to create it. First, the majority of tools are aimed at large, mainly human, genomes. They require specific types of inputs and dominantly rely on paired sample-reference samples, e.g., tumor-normal tissues. Also, they are intended to detect large rearrangements, and they are not scaled to small copy-number events. However, large CNVs are rare in prokaryotes. Second, there are not enough detection tools aimed at bacteria genomes, and some of the already published ones are already deprecated. Also, based on the reviews, there is only a small overlap between the results of various tools. A high false positive discovery is a common issue. Third, bacterial pathogens pose still a highly deadly risk. In 2019, they caused 13.6% of all global deaths. Five bacteria – *S. aureus, E. coli, S. pneumoniae, K. pneumoniae, and P. aeruginosa*, were responsible for more than half of all cases. The bacteria pathogens were the second leading cause of death after ischemic heart failure. As mentioned previously, the CNVs can play a role in antibiotic resistance, bacteria adaptation, and specialization. The issue of bacteria drug resistance is present and emerging. Thus, there is a serious need to develop tools aimed at the detection of bacterial CNVs. All these aspects lead to the development of a new tool which was called CNproScan, derived from the words Copy Number prokaryotic Scanning.

The second tool is a pipeline for the alignment and detection of CNVs and SVs, named ProcaryaSV. The reasons to create the ProcaryaSV pipeline were two. It was more convenient to create a reproducible workflow than running the various scripts every time some parameter changed. Secondly, during the literature research, I came across the topic of merging not only the detection approaches but also the standalone detection tools. This idea origins in the results of multiple reviews which show how little CNV and SV overlap across multiple detection tools

The presented tools extend the scope of tools for a microbiologist to study bacterial organisms. While CNproScan detects deletions and duplications, the ProcaryaSV pipeline enables the detection of inversions and insertions by combining multiple detection tools.

This summary of the doctoral thesis is a shortened version of the doctoral thesis. The theoretical chapters are limited to a minimum while keeping a large portion of the practical chapters.

1. STRUCTURAL VARIATIONS

1.1 Classification

The SVs can be classified into several categories. The most common one is classification regarding copy numbers into **balanced** and **unbalanced events**. Another criterion classifies SVs into **single** and **complex SVs** which consist of more underlying simple SVs. The SVs can be classified based on their size as **fine-scale**, **intermediate-scale**, or **large-scale**. SVs can also be categorized based on the process of creation as **cut-and-paste** and **copy-and-paste**. Structural variation is observed as a junction between two breakpoints in the genome. When the sequencing read spans over a breakpoint junction, it leads to discordant features compared to the other read alignment features. This junction is defined by its orientation, space between breakpoints, etc. [1]

The canonical types of SVs are **deletions, insertions, duplications, inversions, and translocations.** The minimum length of such events is not exactly specified. The initial size threshold was 1 kbp, later decreased to 50 bp but nowadays the SVs are all variants that are not single nucleotide variants (SNV). A more accurate definition than by size could be by a mechanism of creation of that SV. Small indels are created by replication slippage, while larger CNVs are created by homology recombination [2].

The inversions and translocations classify as a balanced type of SVs, whereas the rest as unbalanced SVs. The deletions and duplications are also called **copy number variations** (CNVs) especially when they include gene regions. [3]

The basic illustration of various SV types is in Figure 1.1. The upper boxes represent the reference genome, while the lower boxes represent the sample genome situation. As you can realize, the definition of SV is tied to some reference situation. This reference is another genome, another sample of a different location or time, or a pool of samples merged.

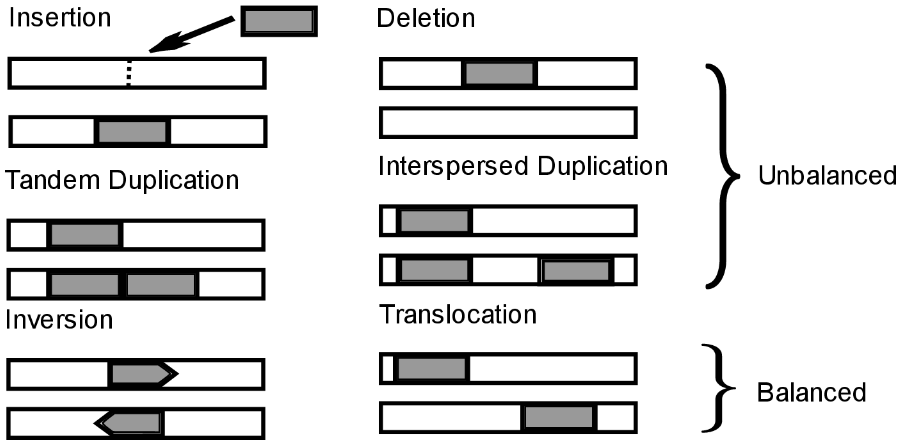


Figure 1.1 – SV types, condition between referential and analyzed genome

1.2 Structural Variations in Bacteria

Although the first gene amplification was observed in the model organism *Escherichia coli K-12* in 1963 [4], the later major effort regarding structural variants and copy number variants was focused on human genomes or generally eukaryotic organisms. However, that was an underestimation of the importance of prokaryotic genome rearrangements as was later discovered. Bacteria are an omnipresent and essential part of nature. There is an estimation of 5×10^{30} bacteria present on the earth. Also, they belong to the most deadly pathogens and multiple issues related to bacterial pathogens emerged, namely growing antimicrobial resistance. [5]

The sequencing of bacterial samples is done in two ways, by cultivating and sequencing **bacterial isolates** or by sequencing **communities**, e.g., **microbiomes**, by shotgun metagenomic sequencing, or by targeted amplicon sequencing. The focus of this thesis lies in the sequencing of bacterial isolates. [6]

Prokaryotic genomes differ in multiple ways from eukaryotic ones. The genome is composed usually of a single double-stranded DNA formed into a circular shape. There can be additional independent circular genomes called plasmids carrying less important though beneficial genes. In some species, e.g. *Shigella*, the plasmids are responsible for virulence [7]. Because of the small size, the bacterial genome is dense. Genes lack introns and are almost next to each other without a significant gap. Some genes are organized in operons, adjacent genes belonging to the same pathway and expressed together.

Most importantly, bacterial genomes are free of large repetitive regions, yet they contain some repetitive elements. These repetitions then serve as a substrate for genome rearrangements. They can also be incorporated through **horizontal gene transfer** (HGT).

It is important to mention that there is a negative relationship between genome stability and repetitive sequences. The bacterial genomes are limited to a finite number of genes they can harbor. They dispose of less-worthy genes to balance new gene gain from HGT. This bacterial continuous gene gain and loss makes them adaptable [8]. [9]–[11]

Generally, rearrangements over 50 bp are considered SVs in Bacteria [6]. The role of SVs in the prokaryotic domain is different compared to eukaryotic genomes. Both evolutionary and phenotypic implications are extensively studied. The prokaryotic genomes are stable between subsequent generations (due to binary fission), but on the evolutionary timeline, they are plastic, shaped by HGT, genome rearrangements, prophages (bacteriophages), and **mobile genetic elements** (MGE). These can all participate in genome rearrangements [5], [12]. Furthermore, the mechanisms of SVs creation are similar to those described in Eukaryotic genomes [10].

The symmetrical design of the genome leads to biased **symmetrical structural variations**. Three forces were described as creating this bias. First, the distance of a gene from the replication origin (oriC) is a large force. More important genes were observed to be close to oriC. Second, there is a difference in replication between the leading and lagging strands. Third, the limitation to having symmetrically sized replichores (halves of a circular chromosome) leads to symmetrical inversions. Symmetrical inter-replichore inversions are the most commonly detected SV in bacteria. [10], [13]–[15]

Structural variations in bacteria can change the distance of a gene from **the replication origin** (oriC) which can have an extensive impact [10]. The SVs and CNVs are part of pathogenesis evolution and antibiotic resistance [16].

2. DETECTION OF STRUCTURAL VARIANTS

2.1 Using single approach

The read-pair approach employs one of the biggest advantages of sequencing – paired-end reads. This approach observes **the position, distance, and orientation of read pairs** in the alignment. The reads which differ from expectations are called '**discordant'**. These discordant reads are mapped closer or further than expected, mapped in inverted orientation, mapped in the incorrect order, or mapped on different chromosomes. [17]

Several **signatures** (features of mapped reads) are defined for classes of structural variations. The easiest signatures for detection are basic insertion and deletion. Pair of reads that span over isolated deletion are mapped in the correct orientation of forward-to-reverse, but the insert size between reads is longer than the expected library insert size.

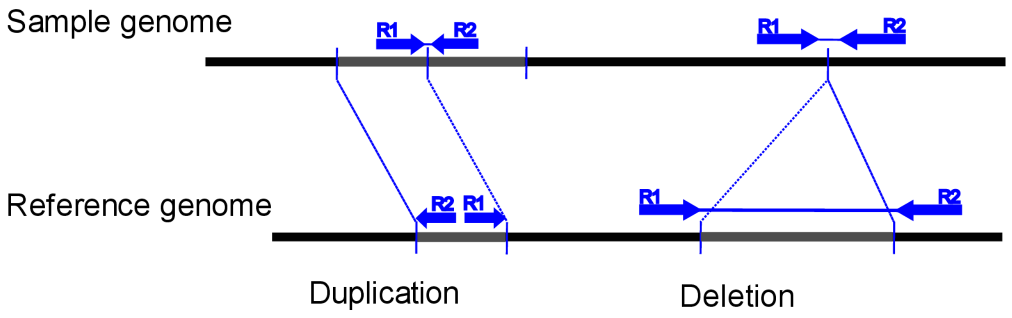


Figure 2.1 - Read-pair signature for deletion and duplication

The read-pair approach can distinguish between tandem and interspersed duplications. The signatures of reads from tandemly duplicated segments include lower insert size (as reads are mapping closer than expected) and reversed both orientation and order of the reads (upstream read location mapping to the reverse and downstream read location mapping to the forward, i.e. -/+). The interspersed duplication signatures include increased insert size and reads mapping to the opposing strands but with reversed order (+/- and -/+). [18]

Another case of duplication is inverted duplication, which shares signatures with inversion. Contrary, direct duplication (unchanged orientation) shares a signature with deletion. These similarities make the detection challenge. The basic signature for duplication and deletion is in Figure 2.1 - Read-pair signature for deletion and duplication. [18]

The data distribution of the insert size is expected to be Gaussian [19], [20]. An interesting case represents sequencing with two different insert sizes libraries. This is designed to overcome the limitation of small insert sizes for detecting larger genome rearrangements. [21]

The split-read approach takes full advantage of mapping properties to the reference genome. It enables single-base resolution. Firstly it was used in the project of human genome indels detection from Sanger sequencing [22]. The signatures are based on an incorrect alignment of mapped reads which is gapped or split. The approach to detect split reads is through **soft clipping**. The soft clip of the read represents a continuous mismatch at the 5’ or 3’ end of the read. The sources of soft clips can be sequencing errors, chimeric reads, reference errors but also structural variants. Another mechanism included in this approach is the **anchor and orphan reads** illustrated in Figure 2.2. This mechanism overlaps with the pair-read approach. [23]

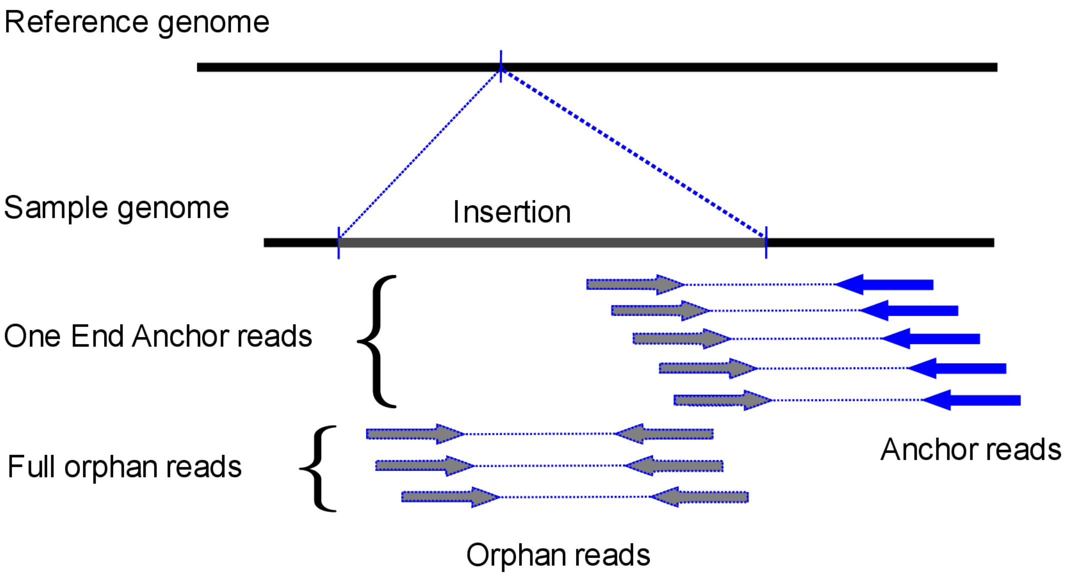


Figure 2.2 – Orphan and anchor reads

Read sequenced over a deletion breakpoint will map with a split mapping signature, where both ends of reads (prefix and suffix) will map to different regions in reference. If its mate pair is uniquely mapped, the split read is masked as a so-called soft-clipped read. This signature is well used by long reads platforms but with short-read data, there can be too much false mapping of read halves. This can be mitigated by limiting the candidate reads or setting conditions. [24], [25]

The decision of what is tagged as soft-clipped and what is tagged as an alignment mismatch depends on the mapping algorithm. The illustration of the soft clipping is in Figure 2.3.

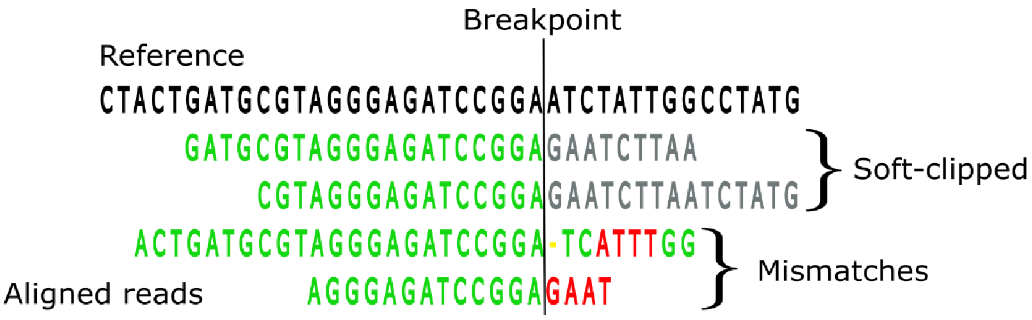

Breakpoint
Reference
CTACTGATGCGTAGGGAGATCCGGAATCTATTGGCCTATG
GATGCGTAGGGAGATCCGGAGAATCTTAA
CGTAGGGAGATCCGGAGAATCTTAATCTATG
Soft-clipped
ACTGATGCGTAGGGAGATCCGGA-TCATTTGG
AGGGAGATCCGGAGAAT
Mismatches
Aligned reads

Figure 2.3 – Soft-clipping illustration

The read-depth approach evaluates the **coverage**, i.e. a number of reads that cover a certain position. The terms **read-depth** and coverage are often used interchangeably unless defined specifically. The distribution of coverage is assumed to be Poisson random distribution. In the presence of biases and sequencing errors, the observed coverage distribution differs from the expected Poisson and is wider [26]. The basic hypothesis is that duplicated regions will manifest significantly elevated coverage. Oppositely, the deleted regions will manifest zero or decreased coverage. Thus, only two signatures are created by the read-depth approach illustrated in Figure 2.4. The essential factor for successful detection is appropriate sequencing read-depth because the read-depth approach assumes that read depth is proportional to copy number. The average read counts in regions correlate very well with DNA copy numbers for Illumina and pyrosequencing platforms, while not for SOLID sequencing [27]. [28]–[30]

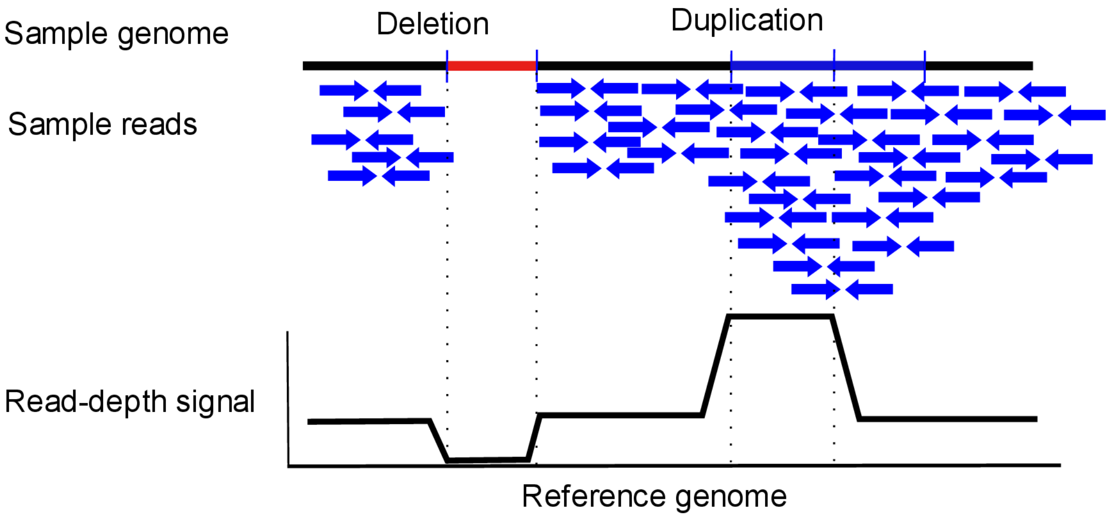


Figure 2.4 – Read-depth signatures and read-depth signal

Generally, the advent of read-depth was easier as the methods applied to CNV detection in array-CGH, e.g. circular binary segmentation, can be used with some modifications on NGS data [27]. However, they are differences: variance in probes is lowest for the normal state (equal copy numbers), and the variance increases for copy number changes. Contrarily, the lowest variance of read-depth is for the deletion state, and it further increases proportionally with increasing copy number [31].

Data processing steps are the following: data preparation (extracting read depth from a pre-filtered set of reads), data normalization (minimizing the influence of sequencing biases), reading read-depth in non-overlapping windows, detection of same copy-number regions (segmentation) and merging them, and estimating the copy-number. [27]

The observed values of read depth can be converted into logarithm, log-ratio (for paired or pooled samples), or Z-scores depending on the algorithm [32], [33]. The CNVs can be detected at the visible read-depth level, implicating statistical testing for significant changes from the global average or neighboring regions. Or it can be detected at the read-depth distribution level, observing the significant deviations from the expected parameters of the distribution. [32]

The results of the read-depth approach are influenced by many external factors. Several biases affecting read depth exist, e.g., PCR amplification bias, GC bias, mappability bias, repetitive segments, and artifacts created through multi-mapping reads. Coping with GC and mappability biases is an essential step of the read-depth approach. Multi-mapping reads represent another ambiguity. This phenomenon emerges when a read can be mapped into multiple positions at the same score (the same uniqueness).

The de-novo assembly approach requires genome assembly. If read length and the amount would be sufficient for a de-novo assembly of the genome, it should be theoretically possible to detect all structural variants including copy numbers, content, and their structure. Such detection would not be based on inference from read signatures but would be directly visible in comparison to the reference, i.e. in self-dot-plot. However, whole-genome sequencing is still costly to perform at parameters that would enable de-novo assembly. Assembly approaches include a whole-genome de-novo assembly and also a local re-assembly to produce contigs which are then compared to the reference genome. The latter is often used in combination with the split-read approach or with the orphaned reads. Generally, the de-novo assembly is limited by required coverage and sequencing costs. The required coverage is about 50× compared to the sufficient 15× coverage required for mapping-based methods. [17]

2.2 Limitations of using a single approach

It is necessary to mention that NGS methods have difficulties to detect SVS in repetitive regions, thus the detection of microsatellites, transposable elements, heterochromatin, and segmental duplications is challenging. This limitation is not possible to overcome with algorithm design, but rather a combination of other sequencing platforms overcomes this. [34]

Each detection method itself has limitations. Split-read is the most precise in exact boundaries of SV, but on the other hand, is very limited to the length of the reads and short reads affect accuracy and precisions. Also, it works only in unique regions of the genome. [35]

Read-pair can detect all types of SVs but is not precise in establishing boundaries. The accuracy of read-pair methods depends on the insert size and its distribution. Small SVs can be skipped in detection with large insert libraries. Similarly to split-read methods, the power is limited in nonunique regions of the genome. [35]

Assembly methods have poor detection power against duplications or repeats and require high coverage. Read-method works well on duplications and can detect the copy numbers as the only method. However, the boundaries resolution is poor. [35]

The limitations of using a single approach are overcome by implementing multiple approaches or even tools together.

Hybrid algorithms were the first to overcome the limitations of distinctive approaches. That is achieved by a combination of more approaches and overlapping their outputs or by increasing the support of SV events by multiple signatures. The breakpoint resolution can be increased by a hybrid approach, which leads to more precise detection of SV boundaries. This is enabled by integrating the split-read approach. The copy number can be calculated by integrating the read-depth approach. The spectrum of detected SVs can also be extended by integrating more approaches. The read-depth method can only detect deletions and duplications and by integrating them with other approaches we can detect a wider spectrum of SV or subtype them. The performance metrics such as sensitivity and specificity can also be improved by a hybrid approach.

3. Objectives of the thesis

The purpose of the thesis is to bring novel multidisciplinary approaches to bacterial genome analysis of copy number variations. CNVs play an important role in bacteria in processes of antibiotic resistance, bacteria adaptation, evolution, and specialization. The issue of bacteria drug resistance is present and emerging. Thus, there is a serious need to develop tools aimed at the detection of bacterial CNVs.

Large structural variations are rare in bacteria because of their small and densely packed genomes. Thus, the detection of small CNVs is more important. Special features of bacterial genomes should be taken into consideration and could theoretically improve performance. Multiple bacterial genomes are not annotated. Therefore, the developed method should rely merely on sequencing reads, and a reference.

Developing a standalone method for CNV detection in bacteria is the first objective. Incorporating this method into a pipeline is the second objective. The sub-objectives were set as follows:

1. Develop a novel method for CNV detection (CNproScan)
    1.1 Using signal-based computational methods
    1.2 Not requiring apriori known genome annotation
    1.3 Targeting bacterial genomes
    1.4 Statistically evaluated and tested

2. Develop a CNV detection pipeline (ProcaryaSV)
    2.1 Targeting bacterial genomes
    2.2 Implementing an efficient merging algorithm
    2.3 Statistically evaluated and tested
    2.4 Enabling the reproducibility and scalability

4. CNPROSCAN

4.1 The Algorithm Design

The CNproScan uses the ‘sandwich’ design with partial blocks stacked vertically, as illustrated in Figure 4.1. The program consists of several main blocks. The first one is coverage normalization, the second is outliers detection to determine CNVs, the third is the application of the read-pair approach and signature rules to narrow the CNV subtype and the last is formatting the output. Each block is here described more from the implementation aspect. [36]

4.1.1 Data preparation

The preparation of sequencing reads before the CNproScan's detection of CNVs is carried out by the usual procedure. After trimming and quality check, the reads are mapped by the aligner. The BWA-MEM was used in the testing [37]. The samtools package is used to handle the rest of the work [38]. The alignment is sorted and written as a binary BAM file. The coverage signal is obtained by the command "samtools depth" with parameter -a which includes zero coverage positions.

For the optional mappability correction, the required genome mappability file is obtained from GenMap [39]. The settings -K 30 -E 2, meaning the size of unique k-mers and allowed mismatches, was generally used for all analysis.

If the user is interested in the correction of the origin of replication bias, then, the location or multiple locations of *oriC* is necessary. This information can be searched for in the DoriC database [40]. The record from DoriC must match with the corresponding genome reference used for alignment.

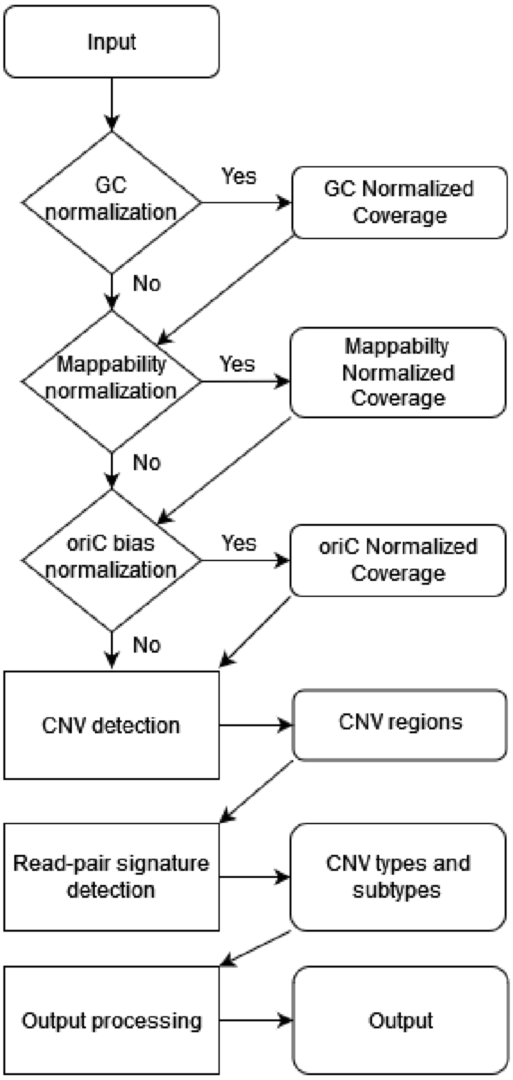


Figure 4.1 – CNproScan workflow

4.1.2 Main function

The CNproScan was developed initially as a set of MATLAB functions and during the peer review, it was rewritten into R. Both versions share the same methodology. Both versions are hosted in GitHub repositories (Table 4.1). All normalizations are optional, but the GC and the mappability normalization are recommended as commonly used.

Table 4.1 – CNproScan GitHub repositories

R version	https://github.com/robinjugas/CNproScan
MATLAB version	https://github.com/robinjugas/CNproScanMatlab

4.1.3 Biases Normalization

The GC normalization is done with the use of the modified Yoon approach [31]. The normalization requires to use of a sliding window. Benjamini et Speed notes that a window size of at least fragment length should be used [41]. Yoon ties the GC normalization and read-depth approach as it is common with the use of a 100bp window[31].

The mappability normalization is based on the Magi approach [27]. The approach is very similar to GC normalization. We use mappability scores calculated from an external tool named GenMap [39]. GenMap focus on the problem of finding the occurrence of a substring with length *k* in the sequence while allowing some errors *e*, when the sequence here is the reference genome sequence. It returns a mappability score, defined as the inverse of the occurrence frequency, of 1 for a unique substring and a mappability score close to 0 for repetitive substrings. GenMap was chosen because it is accessible as a conda package, based on the paper it outperforms previously published competing packages, is exact and non-heuristic, and enables the choice of a number of errors. The k=30 and e=2 were used by authors to perform analysis on *Klebsiella pn*. Thus, we take over the same settings. [39], [42]

The replication origin bias is normalized by our approach, which is based on the previously presented principles for GC and mappability normalization. The genome is binned into 100bp windows similarly to the GC normalization. Whether the bias is corrected depends on the user and the result of the Spearman correlation test p-value.

Firstly, it is required to remove any outliers. It is because the distance to *oriC* is symmetrical and deletion or duplication on one side of the symmetry could completely deflect the normalization of the regions with the same distance to the replication origin. The outliers are removed using the 1.5 times IQR (interquartile range) rule on both tails. Then, the distance to *oriC* is calculated in windows of 100 bp. The circular genome correction is applied so that the minimum value of all possible constellations is chosen.

The important parameter is the level of rounding. This parameter impacts how many windows are taken together in estimating the median read depth of a certain distance to *oriC*. Rounding to thousands means that approximately ten 100bp windows on each side are taken together, rounding to tens of thousand means a hundred windows are taken together. The table of values of *oriC* distances and median read depths is constructed. Importantly, the Spearman correlation value is computed between the *oriC* distances and estimated read depth medians. The p-value is calculated for the alternative hypothesis that Spearman's correlation coefficient Rho (-1,1) is different from zero. The normalization is further applied if the p-value of this test is less than the alpha value of 0.05. The p-values are computed via the asymptotic approximation, which means that they depended on the number of *oriC* distance values and will likely be less than the alpha level for the lower rounding level. However, it was observed that higher rounding is more robust, and rounding to tens of thousand is applied. The replication origin bias is then normalized by the formula

$$\overline{RC_i} = RC_i \cdot \frac{m}{m_{iORICdist}}, \tag{4.1}$$

where m stands for the median value of read-depth of all windows and $m_{iORICdist}$ for the median value of the windows with the same *oriC* distance.

The information about the genomic position of replication origin is accessible in the DoriC database [40]. If the *oriC* normalization is intended, it is useful to check if there is a record in DoriC for the selected genome reference or choose the different one.

4.1.4 Outliers as CNV candidates

The normalized coverage signal is sent into the outliers analysis. For this task, the CNproScan employs the GESD outlier detection algorithm described before. As this algorithm requires the upper bound of the suspected outliers. To serve a robust estimation of this upper bound, the modified Z-score outliers detection is used and values with a modified Z-score above 3.5 are labeled as candidate outliers. This usually leaves a large number of candidates, meaning several thousand and e.g., more than ten thousand candidates for real *Klebsiella pneumoniae* samples.

To reduce the performance drawback of testing thousands of values in a for-cycle, the GESD testing is done in a parallel way. This was possible because the task is possible to parallelize. This is done in the R version with the use of R packages *parallel*, *doParallel*, and *foreach*. Simply done, the whole genome is divided into *n* sections which are tested separately and parallel. The argument *cores* in R main function serve as the definition of the number *n*. After each partial segment is done, the results, which are genomic positions of significantly large coverage values, are merged into a single vector.

The results are post-processed. The vector of outliers is sorted and the gaps between outliers are detected using the lagged differences function. Then, depending on the parameter *peakDistanceThreshold*, which is set up to 20bp, the adjacent outliers closer than 20bp are merged into consecutive segments. These serve as a basis for CNV events.

In Figure 4.2 the results of outliers detection are displayed for the artificial genome. The details of the creation of the artificial genome dataset are described later in the chapter Benchmarking on simulated data. The zero coverage values are removed first (in red). The candidate outliers from the modified Z-score method are in blue and multiple of them are overlapped with blue as they were confirmed by the GESD outliers test.

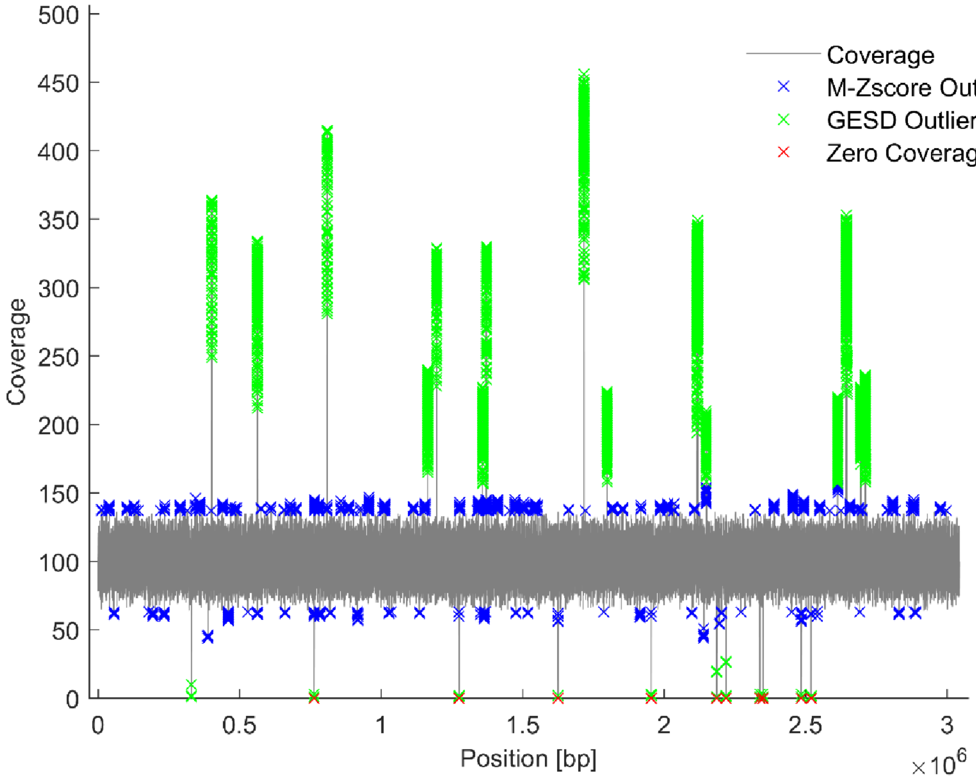


Figure 4.2 – Detecting outliers in artificial genome

4.1.5 Extending CNV boundaries

Because of the nature of outliers detection, only the most significant parts of CNVs are uncovered, i.e. for duplications, only the peaks are labeled yet. To extend the borders of the CNV down to the baseline, the slope of a line is used. The line is given as a coverage region. The slope is calculated as $m = \frac{y_2-y_1}{x_2-x_1}$ and the distance between x-axis values $x_2 - x_1$ is defined by a specified step (11 bp, optional). The slope is calculated gradually on both ends of the peak until there is a change in the numerical sign for the value of slope m. If the change of slope is detected x-times (x defined as 5, optional), then it is considered as the CNV's border. The updated version adds the condition of reaching the baseline defined as the average of the coverage. A detail of one CNV with extended boundaries is plotted in Figure 4.3. Notice how the whole depth of CNV is detected compared to the previous Figure 4.2 (third peak from the end).

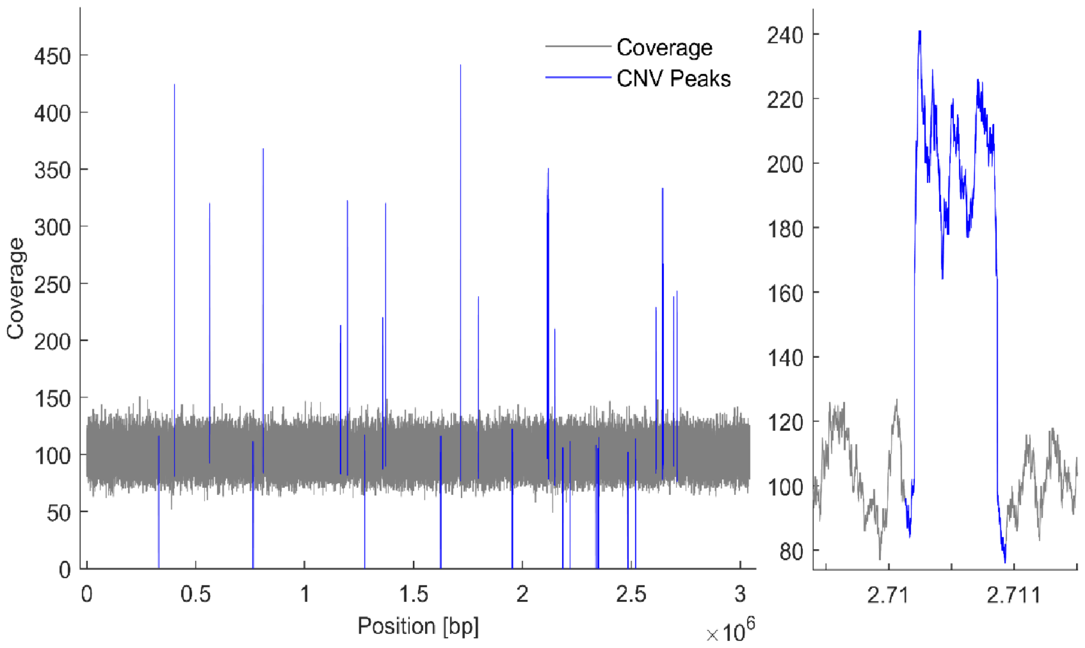


Figure 4.3 – Detail on extending the CNV boundaries

4.1.6 Read-pair information

Since CNproScan detects solely CNVs, I choose only a few signatures to use from the read-pair approach. The features of deletion, tandem duplication, and interspersed duplication are targeted. The signatures are as defined in Soylev's work [18].

Contrary to other approaches which merge the two pieces of information, the read-pair approach helps to validate and specify the CNV events detected from the read-depth approach. Both approaches are not equal, rather the read-pair information is subjugated to the main read-depth information. This is because as mentioned earlier, certain signatures are not exclusive, and specifically direct duplication signature is the same as a deletion signature. Because of this reality, the read-pair approach is subjugated to the read-depth approach which can distinguish between duplication and deletion very clearly.

I search for outliers in the signatures because the distribution of fragment sizes in a library is Gaussian and the detected signature is usually largely distant from the normal state. The outliers are defined on a simple rule of 1.5IQR, which means that insert sizes larger than the sum of the upper Q3 quartile and 1.5 times the interquartile range are labeled as an outlier.

Furthermore, the features are searched only in the regions of already detected CNVs and not genome-wide. This reduces the computational time. The genomic regions are scanned inside the detected CNV boundaries extended by the insert size on both ends.

The decision of which SV subtype will be chosen is done by selecting the most prevalent signature inside the region.

The detection of discordant reads uses the fields defined in the SAM/BAM format [38], [43], mainly TLEN (Template length), and bitwise FLAG, which contains information about the read`s relative orientation, etc. In the R version, packages *Rsamtools*, *GenomicRanges*, and *IRanges* were used to access the BAM file structure. In the Matlab version, the *bamread* function from the Bioinformatics Toolbox was used.

As already mentioned, the BAM is scanned only in the regions of detected CNVs. The boundaries are defined as the start and end of CNV's coordinates plus/minus the insert size. The insert size is defined as the median of absolute values of whole BAM reads. Similarly, the interquartile range, the first and third quartiles are defined based on the whole BAM file. The "isize" in Rsamtools (TLEN in BAM definition) is used for these estimations. For median read length, the "qwidth" is used.

The circular genome correction is used as described, and each CNV region defined in the read-depth approach part is scanned for reads defined by specified signature rules.

The theoretical signature rules were extended because of some observations from the testing and are all listed in Table 4.2.

Table 4.2 – Overview of applicated signature rules

Type	**Subtype**	**Strand orientation**	**Insert Size**
Deletion		+/- (-/+)	Higher
Tandem Duplication	Direct	+/- -/+	Lower
Tandem Duplication	Indirect	+/+ -/-	Lower
Interspersed Duplication	Direct	+/- -/+	Higher
Interspersed Duplication	Indirect	+/+ -/-	Higher

4.2 Benchmarking

The performance of CNproScan was evaluated on the dataset which had been previously used in the testing of the CNOGpro package [44]. This dataset is based on the *S. aureus* genome sequence into which were imputed 30 artificial CNVs with defined genomic coordinates, lengths, and copy-number. There are 12 deletions and 18 duplications of various lengths, mainly focused on the small events.

The dataset has two parts – one with imputed CNVs and the second one with no CNVs to evaluate the metric of true negatives. These two datasets were constructed with different coverage values – 10×, 20×, 100×, and 200×. The sequencing reads were generated with the ART reads simulator [45] and then processed by the described pipeline.

The performance of CNproScan was compared directly with LUMPY [46], CNVnator [47], Pindel [48], and DELLY [21]. And indirectly with CNOGpro [44], cnv-seq[49], and cn.MOPS [50], where I adopted the previously published results.

The main focus was on 100× coverage, then the only tools competing well were evaluated for other coverages 10×, 20×, and 200×. The results are evaluated by the metrics of the confusion matrix. The Accuracy, Sensitivity/Recall, Specificity, Precision, and F1 score are all used across the results chapters. Lastly, the results and discussion are taken from CNproScan's published paper [36].

4.2.1 Results for coverage 100×

The most emphasis was put on the 100× coverage. All 8 tools are benchmarked for this value of coverage. It is high enough to provide a sufficient signal-to-noise ratio with easily detectable CNVs. The complete results with the number of correct and false observations, and performance metrics are in Table 4.3.

Focusing on the default 100× coverage (in Table 4.3), the overall accuracy achieved was 93% and was the highest among tools. CNproScan detected 26 TP. Four FN CNVs were short regions under 26 bp in length, consisting of 2 deletions and 2 regions with a copy number of two. There was a single CNV event detected outside the original coordinates, which we consider an FN case.

CNproScan and Pindel were both able to detect shorter CNV events than other methods. Pindel has higher sensitivity as it was able to detect 27 out of 30 CNVs. However, Pindel's high sensitivity has the drawback of a high false positive rate. Pindel detected 371 CNVs, mainly deletions, in the empty reference dataset. Furthermore, there were another 418 FPs in the dataset with CNVs. A high false discovery rate in CNV detection is a common problem stated in the literature [51], however, only Pindel suffered from this.

Table 4.3 – Results for coverage 100×

	CNproScan	**CNOGpro**	**Cnv-seq**	**cn.MOPS**	**LUMPY**	**CNVnator**	**DELLY2**	**Pindel**
TP	26	22	14	7	13	21	22	27
FP	0	0	0	0	2	0	0	789
FN	4	8	16	23	17	9	8	3
TN	30	30	30	30	30	30	30	30
Accuracy	93.3	86.7	73.3	61.7	69.4	85.0	86.7	6.7
Sensitivity	86.7	73.3	46.7	23.3	43.3	70.0	73.3	90.0
Precision	100.0	100.0	100.0	100.0	86.7	100.0	100.0	3.3
Specificity	100.0	100.0	100.0	100.0	93.8	100.0	100.0	3.7
F1 score	92.9	84.6	63.6	37.8	57.8	82.4	84.6	6.4

Other tools detected fewer CNVs. Sorted from the lowest number of TPs, there was cn.MOPS, LUMPY, cnv-seq, CNVnator, and equal CNOGpro and DELLY2. Since they all detected zero or a very low number of FPs, other metrics are influenced by the number of TP and FN. Thus, precision and specificity for all tools except Pindel were high.

CNproScan achieved the highest F1 score. The close competitors in this metric were CNOGpro, CNVnator, and DELLY2.

Although DELLY2 and LUMPY are both hybrid triple method combinations, they differ significantly in the detection of CNVs. DELLY2 performed better.

The detection of short CNVs with a low copy number is the most challenging task. For 100× coverage, we can conclude that CNproScan detected duplicated CNVs longer than 37 bp. Two duplicated CNVs of 4bp and 23bp lengths were not detected. The shortest detected deletion was 4bp and then two 17bp deletions.

The performance of the other tools varied. Pindel (90%) followed by CNproScan (86.67%) achieved the highest sensitivity. The third best performing in sensitivity were CNOGpro (73.33), DELLY (73.33), and LUMPY (70.00). CNproScan achieved the highest accuracy (93.33%). CNOGpro (86.67%.), LUMPY (85.00) and DELLY (86.67%.) were close in accuracy.

4.2.2 Results for coverage 10, 20, 200×

In the evaluation of other coverage's effect on the performance, only the best performers from the previous chapter were selected to reduce the complexity of the results. Selected were: CNproScan, CNOGpro, CNVnator, LUMPY, DELLY, and PINDEL.

I benchmarked CNproScan and others at four different coverage values: 10×, 20×, 100×, and 200×. The complete performance metrics are in Table 4.4. The highest values per row are highlighted in bold font type. The CNOGpro was aborted at 200× coverage because of an under-dispersion error, so the results are missing for this coverage.

For 10× coverage, the CNproScan's sensitivity was 66.67%, and 20 out of 30 CNVs were detected. Pindel had the highest TP count of 26, while also having the highest FP rate. The second highest TP count has DELLY and CNproScan. LUMPY has 17 TPs. DELLY and LUMPY had both zero FP. Contrary, there were 19 FP and an additional 20 FP in an empty dataset detected by CNproScan. The combined metric score was the best for LUMPY and DELLY, then CNVnator followed by CNproScan. The hybrid methods LUMPY and DELLY performed very well in the shallow coverage.

For 20× coverage, CNproScan achieved the highest accuracy (86%) and detected 22 TP. CNOGpro also detected 22 TP, LUMPY 21 TP, DELLY 20, and Pindel 27 TP, thus Pindel had the highest sensitivity. There was no FP detected with CNproScan. There is a visible step in detection quality from increasing coverage from 10× to 20×. The combined metric score was the best for CNproScan followed by LUMPY and DELLY.

100× coverage was discussed in the previous chapter, the highest combined score was achieved by CNproScan followed by CNOGpro, LUMPY, and DELLY. Only Pindel detected one more TP than CNproScan but suffered from a high false positive rate across the complete artificial dataset.

Doubling the coverage to 200×, CNproScan detected 28 TP and 1 FP. The second closest was Pindel with 27 TP. The accuracy and sensitivity were the highest for CNproScan as the overall combined score.

Beginning with the 20× coverage, the CNproScan had the highest F1 score and Accuracy and kept it to 200×.

There is also Figure 4.4, where precision, recall, and F1 scores are plotted. It is visible how since reaching coverage 20×, the performance metrics for CNproScan are going up to the highest numbers.

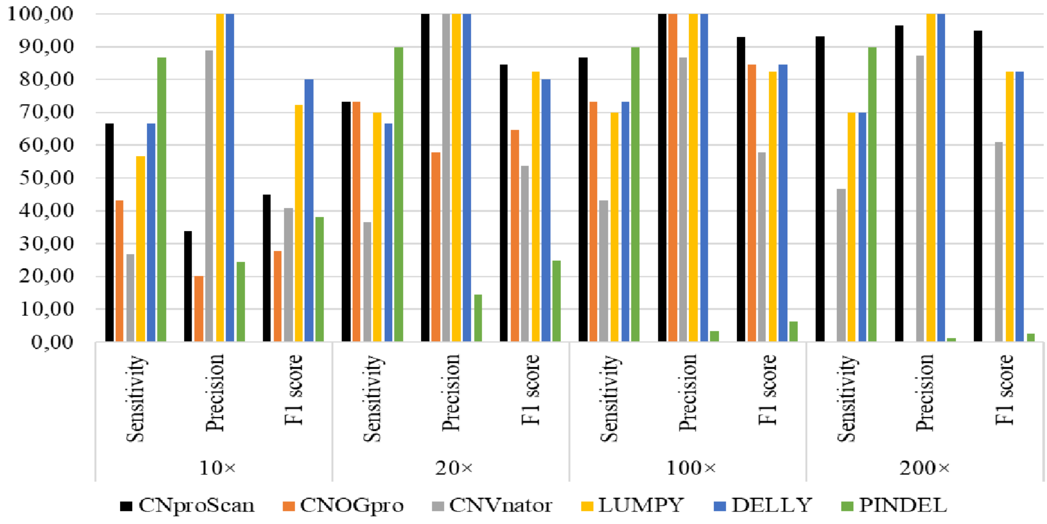


Figure 4.4 – Sensitivity, Precision and F1 scores of the simulated dataset

Table 4.4 – Results of all coverage values

10×						
	CNproScan	**CNOGpro**	**CNVnator**	**LUMPY**	**DELLY**	**PINDEL**
Accuracy	50.5	38.7	62.3	78.3	**83.3**	40.0
Sensitivity	66.7	43.3	26.7	56.7	66.7	**86.7**
Precision	33.9	20.3	88.9	**100.0**	**100.0**	24.5
Specificity	43.5	37.0	96.8	**100.0**	**100.0**	27.3
F1 score	44.9	27.7	41.0	**72.3**	**80.0**	38.2
20×						
	CNproScan	**CNOGpro**	**CNVnator**	**LUMPY**	**DELLY**	**PINDEL**
Accuracy	**86.7**	68.4	68.3	85.0	83.3	25.8
Sensitivity	73.3	73.3	36.7	70.0	66.7	**90.0**
Precision	**100.0**	57.9	**100.0**	**100.0**	**100.0**	14.4
Specificity	**100.0**	65.2	**100.0**	**100.0**	**100.0**	15.7
F1 score	**84.6**	64.7	53.7	82.4	80.0	24.8
100×						
	CNproScan	**CNOGpro**	**CNVnator**	**LUMPY**	**DELLY**	**PINDEL**
Accuracy	**93.3**	86.7	69.4	85.0	86.7	6.7
Sensitivity	86.7	73.3	43.3	70.0	73.3	**90.0**
Precision	**100.0**	**100.0**	86.7	**100.0**	**100.0**	3.3
Specificity	**100.0**	**100.0**	93.8	**100.0**	**100.0**	3.7
F1 score	**92.9**	84.6	57.8	82.4	84.6	6.4
200×						
	CNproScan	**CNOGpro**	**CNVnator**	**LUMPY**	**DELLY**	**PINDEL**
Accuracy	**95.1**	-	70.0	85.0	85.0	2.7
Sensitivity	**93.3**	-	46.7	70.0	70.0	90.0
Precision	96.6	-	87.5	**100.0**	**100.0**	1.3
Specificity	96.8	-	93.3	**100.0**	**100.0**	1.4
F1 score	**94.9**	-	60.9	82.4	82.4	2.5

4.2.3 CNV length analysis

Next, I analyzed how tools dealt with various CNV lengths. The histogram depicting the tools' ability to detect various CNV lengths for 100× coverage is in Figure 4.5. The y-axis shows the count of CNVs detected within four defined bin sizes. The numbers of CNVs in each bin are shown in the brackets in the figure legend. Only CNproScan and Pindel detected the shortest CNVs (blue color). The CNV lengths are categorized into 4 bins: 0-25bp, 26-100bp, 101-1000bp, and 1001-4000bp.

The majority of tools coped perfectly with the longest CNVs (1001-4000bp). Only Pindel and cn.MOPS did not detect a 1302bp duplicated CNV. In the 101-1000bp bin, several tools struggled to detect all CNVs – namely cnv-seq, and cn.MOPS, CNVnator.

On the contrary, only 5 tools detected some CNVs from bin 26-100bp. CNproScan (3 out of 4) and Pindel (4 out of 4) detected the most CNVs. Others were CNOGpro, LUMPY, and DELLY. In the smallest CNVs under 25bp, only CNproScan (3 out of 6) and PINDEL (4 out of 6) detected any CNVs.

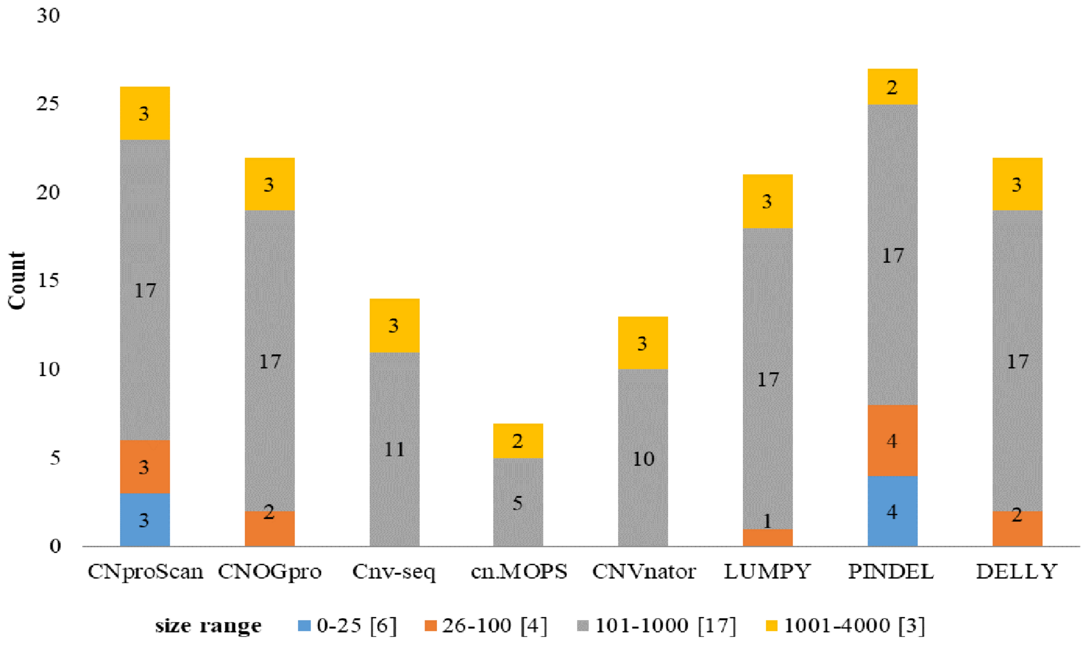


Figure 4.5 - CNV Size Histogram.

5. PROCARYASV

Such as the hybrid method removes the limits of a single approach, the integration of multiple detection tools limits their weakness and improves performance. This issue is tied to the topic of merging structural variants.

The topic of merging variant callers is more advanced in the field of SNP or SNV variant calling, where various tools are already being successfully merged based on their performance, e.g., using machine learning methods [52].

In the fields of SVs or CNVs, the problem of overlaps arises. The parameters of the minimal overlap and the type of overlap (equal, within, etc.) can be both user-defined or hard-coded. However, these parameters are usually defined somehow arbitrarily.

What reliable results are is the question to ask. Generally, the union or the intersection of results is the most common approach. It depends on the preference for higher sensitivity or specificity. Most effortlessly, the reliable results are those given by the most tools. Then, the threshold of how many tools are the most has to be set. On the other hand, rare events could be omitted. Alternatively, the union approach likely produces a high rate of false positives. A weighted approach can be applied if performance metrics are known. But for accurate performance metrics, you need a valid ground truth set, ideally validated by sequencing methods. [51]

5.1 Pipeline Design

I decided to create a CNV/SV calling pipeline based on the Snakemake framework [53]. It is a Python-based workflow management system for reproducible and scalable analysis. It consists of so-called Snakemake rules which define the inputs and outputs of a given rule. The rule serves to call a certain function, tool, package, etc. Parameters that are necessary for the called tool can be specified too inside the rule or can be adopted from the external configuration file. Scalability parameters can be defined too, such as the number of threads or memory requirements. The possibilities are multiple.

I called the pipeline ProcaryaSV denoting the focus on prokaryotic genomes. It is based on commonly used CNV and SV detection tools and state-of-the-art processes for manipulating sequencing data. All the necessary specifical inputs for each SV/CNV caller are processed as described by the caller's manuals. In some cases, I used or further modify the Snakemake Wrappers repository where the finished easy-to-use rules and wrappers (small Python scripts calling the tools) are available.

The overall simplified workflow is in Figure 5.1. Only the basic tools are pictured, without raw reads quality check or the optional trimming parts.

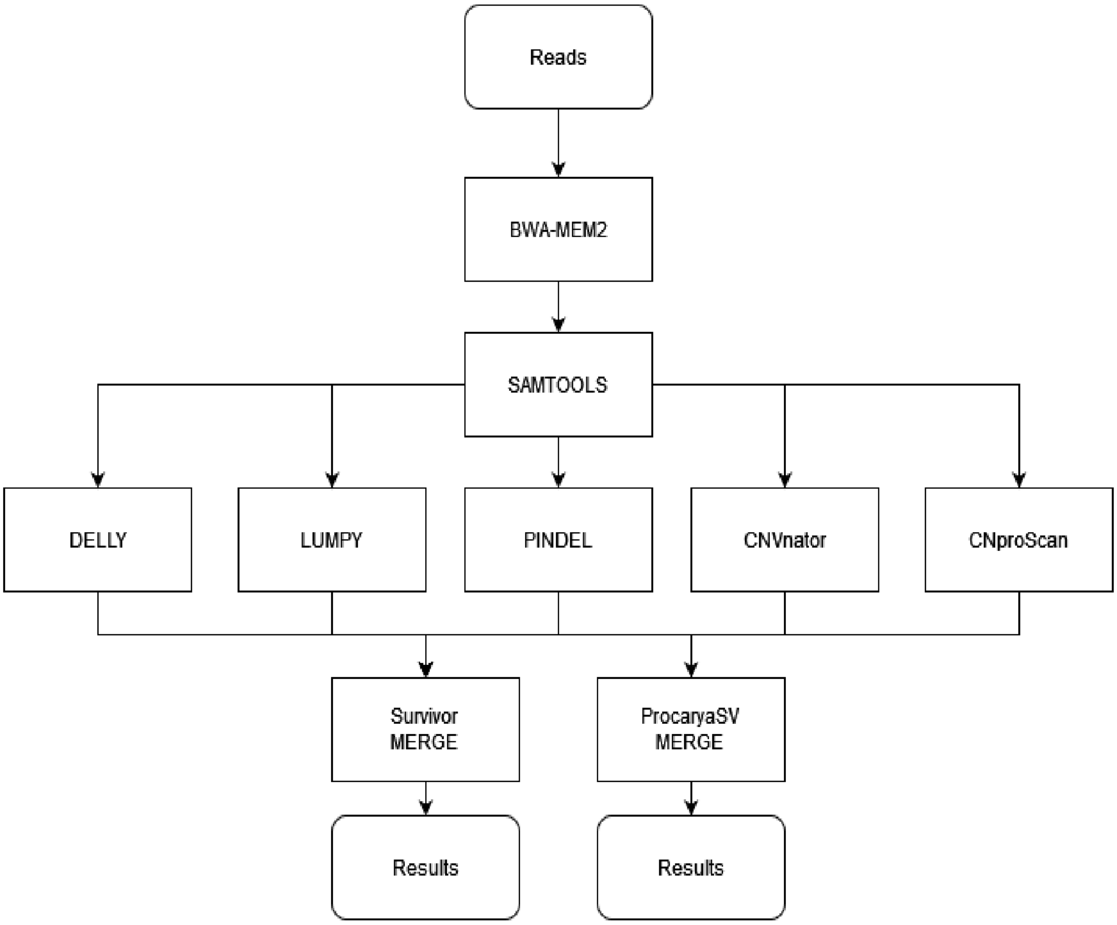


Figure 5.1 – ProcaryaSV workflow

Two CNV callers and three SV callers were used. One of the requirements was that the tools has to be placed in some conda repository to be easily installed by Snakemake. The only exception is the CNproScan which is available only in GitHub so far. Also, tools mustn't require too obscure inputs, e.g. from some deprecated packages. Most importantly, tools have to be suitable for haploid prokaryotic genomes. Thus, the CNproScan, CNVnator, LUMPY, DELLY2, and Pindel were selected.

The ProcaryaSV is available from the GitHub repository in Table 5.1.

Table 5.1 – ProcaryaSV GitHub repository

Repository	Version
https://github.com/robinjugas/ProcaryaSV	1.0

5.2 Merging algorithm

The two main inputs to merging SVs are merging BED files, using the bedtools [54], or merging the VCF files, using, for example, SURVIVOR (StructURal Variant majorIty VOte) [55] or SVDB [56].

I tried both SURVIVOR and SVDB to merge the resulting VCF files. The SVDB failed completely resulting in non-readable files because it writes the genomic sequence of the event into the file. The SURVIVOR did better compared to SVDB and thus was kept in the pipeline for user comparison.

The merging algorithm of ProcaryaSV parses the VCF outputs for all callers and respects their specifics. It separates four categories of SV calls: deletions, duplications, inversions, and insertions, and merges them separately. Insertions and inversions are called only by Pindel and Delly2, while deletions and duplications are called by all of them.

Here, I present my own approach to merging SVs based on cumulating binary vectors and then thresholding them. The user can define the value of the threshold by his or her preference. The input is VCF files from callers. The results are formatted as a TSV (tab-separated values) file, which can be imported into any spreadsheet application. The parameters of minimum and maximum SV length are to be set. All SVs not fitting into these are deleted.

For every type of SV detected (DEL, DUP, INV, INS) the simple binary vector is created for each caller separately and then these are summed up (see illustration in Figure 5.2). This means that a region called by two callers will have a value of two spanning the region where these callers overlap.

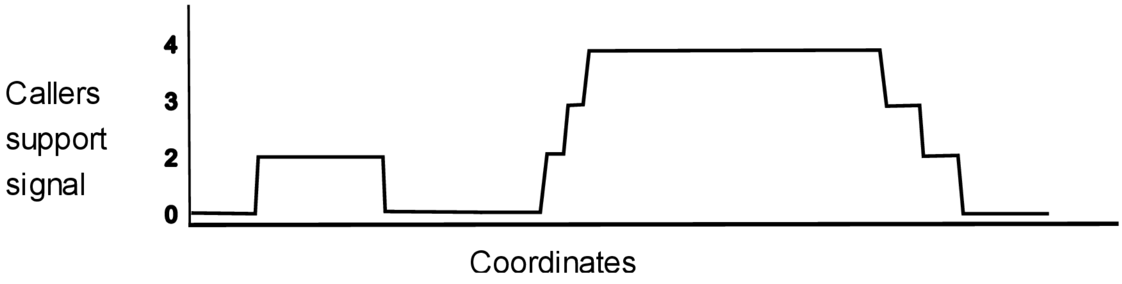
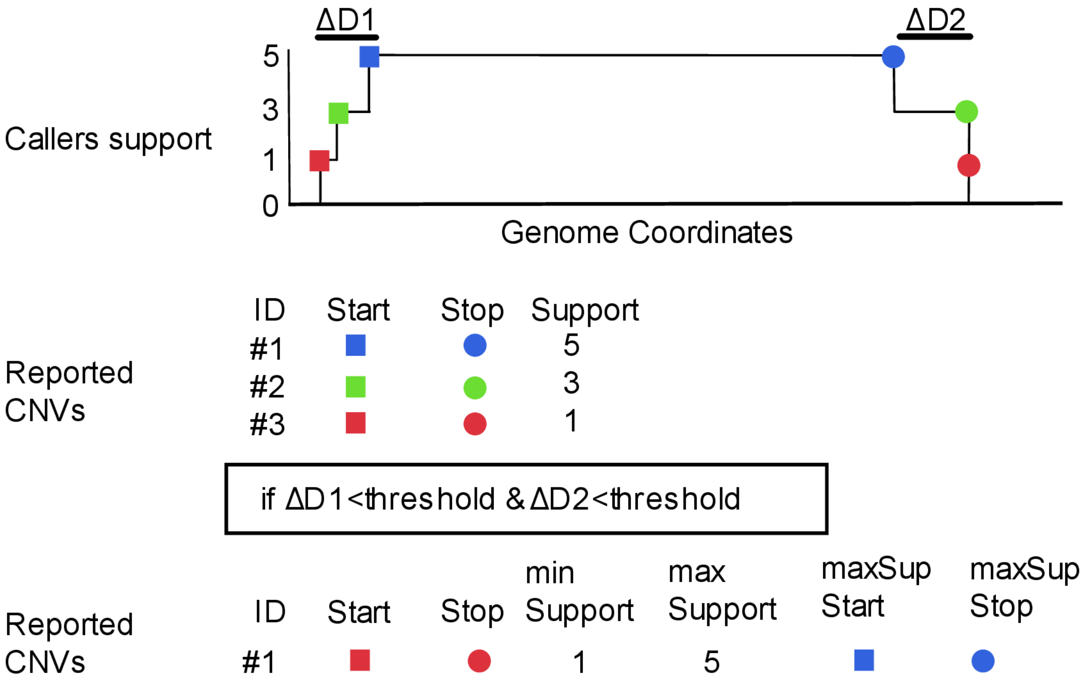


Figure 5.2 – Caller's support signal of two structural variations. First one with a support of two callers, second one with a support of four callers. The steps are created by reported different start and stop coordinates.

In the first iteration, all levels of callers support are outputted except those under the value of the user-defined caller's threshold. The regions called the most times are outputted first and then it proceeds to a lower number of callers. This also means that certain regions can be reported multiple times, once as a shorter region of higher support, and later as a longer region of lower support. The number of callers that called the regions is reported.

In the second iteration, the events are searched for overlaps with the use of the Iranges package function findOverlaps. The important parameter here is the maxgap, meaning the maximum allowed distance between the start and end coordinates. This step is to collapse overlapping events into one. The one event is reported with corresponding values and coordinates with maximum support are saved. The diagram of merging is in Figure 5.3. The threshold represents the maxgap parameter and is applied to all firstly reported events. If the condition was not met, #3 would be reported separately.



ID	Start	Stop	Support
#1	🟦	🔵	5
#2	🟩	🟢	3
#3	🟥	🔴	1

ID	Start	Stop	min Support	max Support	maxSup Start	maxSup Stop
#1	🟥	🔴	1	5	🟦	🔵

Figure 5.3 – SV merging diagram. The colored objects refer to detected SV’s coordinates.

After this, the reported CNVs are searched for overlaps again, to report potential cases where multiple shorter CNVs overlap a single long CNV. This information is stored in the output file.

In the last iteration, all events are backtracked. The information about the number of underlying events and percentual coverage by each caller is recorded and saved. A region can be supported by multiple callers, but they can contribute as multiple separately reported events merged because of the merging step.

The tabular separated file (.tsv) is the output together with informative graphs. These are the Venn diagram of callers and pie plot of different SV types' abundances.

5.3 Benchmarking

The merging algorithm of ProcaryaSV was benchmarked on the previous artificial dataset. The various values of minimum callers support (MinCallers) were used to decide the optimal value. The threshold is inclusive, the operator ‘>=’ is used.

Table 5.2 – ProcaryaSV merging performance metrics

10×					
MinCallers threshold	**1**	**2**	**3**	**4**	**5**
Accuracy	36.3	**90.0**	85.0	73.3	56.7
Sensitivity	90.0	80.0	70.0	46.7	13.3
Precision	21.8	100.0	100.0	100.0	100.0
Specificity	23.6	100.0	100.0	100.0	100.0
F1 score	35.1	**88.9**	82.4	63.6	23.5
20×					
MinCallers threshold	**1**	**2**	**3**	**4**	**5**
Accuracy	37.0	**90.0**	83.3	83.3	58.3
Sensitivity	90.0	80.0	66.7	66.7	16.7
Precision	22.3	100.0	100.0	100.0	100.0
Specificity	24.2	100.0	100.0	100.0	100.0
F1 score	35.8	**88.9**	80.0	80.0	28.6
100×					
MinCallers threshold	**1**	**2**	**3**	**4**	**5**
Accuracy	13.1	**87.3**	85.0	85.0	61.7
Sensitivity	90.0	83.3	70.0	70.0	23.3
Precision	6.7	89.3	100.0	100.0	100.0
Specificity	7.4	90.9	100.0	100.0	100.0
F1 score	12.5	**86.2**	82.4	82.4	37.8
200×					
MinCallers threshold	**1**	**2**	**3**	**4**	**5**
Accuracy	7.0	**88.9**	85.0	85.0	61.7
Sensitivity	93.3	86.7	70.0	70.0	23.3
Precision	3.5	89.7	100.0	100.0	100.0
Specificity	3.8	90.9	100.0	100.0	100.0
F1 score	6.8	**88.1**	82.4	82.4	37.8

All performance metrics are in Table 5.2, the maximum F1 scores are in bold. The maximum F1 scores are all achieved when the MinCallers threshold is set to 2. The precision-recall curves for all coverage levels are in Figure 5.4. The precision remains the same from the MinCallers threshold set to 2 (higher coverage) or 3 (lower coverage), while a threshold lower than 2 brings a lot of false positives. Recall (sensitivity) decreases to very low numbers, omitting many true positives. Following previous results, setting the minimal callers threshold to 2 is the optimal setting to balance precision and recall.

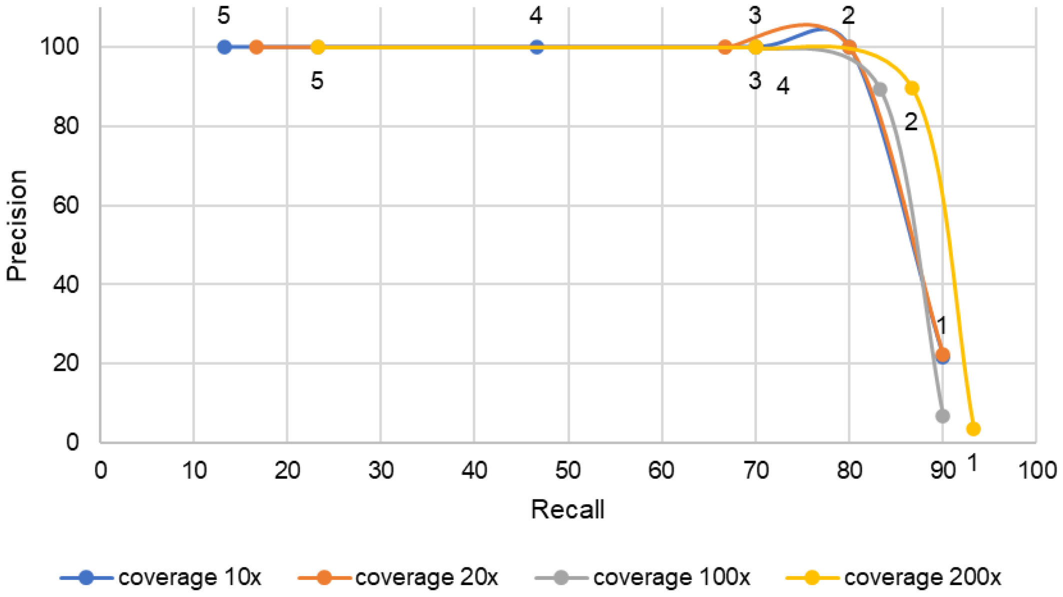


Figure 5.4 – Precision-recall curve for MinCallers threshold parameter. Digits near lines denote the parameter value

For comparison, in Table 5.3 there is an evaluation of the SURVIVOR merging algorithm. The numbers from the previous table are added for easier comparison. The SURVIVOR merge command settings were set to minimal callers 2, maximum allowed distance 1000, and minimal considered SV length 1. Looking at the results, they are almost the same looking at the same settings of minimal callers 2 for both algorithms. The ProcaryaSV had higher F1 scores for 20× and higher coverages by a few points.

This is expected based on the description of the SURVIVOR merging method. In SURVIVOR, two SVs are defined as overlapping if their start and stop coordinates are within 1 kb and of the same SV class.

Table 5.3 – SURVIVOR and ProcaryaSV performance metrics for minCallers of 2

	Coverage	**Accuracy**	**Sensitivity**	**Precision**	**Specificity**	**F1 score**
SURVIVOR	**10×**	90.0	80.0	100.0	100.0	88.9
	20×	85.0	70.0	100.0	100.0	82.4
	100×	87.1	80.0	92.3	93.8	85.7
	200×	85.5	76.7	92.0	93.8	83.6
ProcaryaSV	**10×**	90.0	80.0	100.0	100.0	88.9
	20×	**90.0**	80.0	100.0	100.0	**88.9**
	100×	**87.3**	83.3	89.3	90.9	**86.2**
	200×	**88.9**	86.7	89.7	90.9	**88.1**

CONCLUSION

The main topic of the thesis is the detection of copy number variations specifically in the prokaryotic genomes. While there is a tremendous and still increasing number of papers focused on the topic of SVs and CNVs, the resources related to bacteria are much less frequent. The work presented in the thesis aims to partially fill this gap.

In two practical chapters, I described the CNproScan algorithm and pipeline implementing it called ProcaryaSV. Both tools might be useful for microbiology research.

The CNproScan was published two years ago and was minorly updated a few times, thus I presented the original results with the updated ones. The methodology is based on read-depth navigated CNV detection combined with read-pair-based categorization. The read-pair approach is based on recent knowledge and enables the categorization of CNVs into known duplication types. The CNproScan does not consist only of detection methods. It handles the GC and mappability biases and bacteria-only related replication origin bias.

The CNproScan was tested on the artificial dataset of various coverages (10×, 20×, 100×, 200×) and compared with other seven CNV detection tools. CNproScan had the highest accuracy and F1 score for 20×, 100×, and 200× coverage. The accuracy for 100× was 93.3 % and the F1 score was 84.6 %. That is about 10 % higher than the closest competition. Also, it proved to be useful for detecting short CNVs under 25bp. The reported CNV boundaries are accurately corresponding to the specified boundaries in the majority of test cases. The accuracy of reporting a valid copy number is about 75 %.

Integration of multiple detection tools has already been done in the past. Merging two methods can easily be done, but the scalability decreases with adding more tools. Thus, I used a signal representation of genome rearrangements and summed the signals of individual detection tools. The merging algorithm was tested on the previous artificial dataset and compared with the SURVIVOR merging algorithm. Generally, the two methods are comparable, yet for coverages starting at 20×, the ProcaryaSV's merging algorithm performed slightly better. Both accuracy and F1 score are about 90 %. The parameter of minimal callers support, denoting how many callers have to detect an SV, was calculated from the precision-recall graph to be ideally 2 or higher. The ProcaryaSV pipeline employs five state-of-the-art detection tools and provides all necessary inputs and outputs for them. The pipeline enables reproducibility and is coded in the Snakemake.

Regarding the limitations of the presented methods. The CNproScan's running time is higher than the competition. This was mitigated as much as possible by implementing parallelization. Furthermore, the algorithm is based on the read-depth approach and requires a certain level of coverage, which is 20× based on results. However, coverage higher than 15× is a common requirement for the detection of any genome rearrangements. Generally, higher coverage leads to higher accuracy.

Nowadays, the topic of bacteria drug resistance is an urgent task. Genome rearrangements, including copy number variations, play a role in this issue. Other than that, genome rearrangements participate in the evolution and specialization of bacteria. Next-generation sequencing is still widely used and brings the high throughput necessary for accurate CNV detection. A reliable tool that is directly designed to detect CNVs in bacterial genomes (like the CNproScan), unlike tools designed for eukaryotic genomes, is essential. In turn, the proposed ProcaryaSV pipeline will enable CNV and SV analysis with maximum support for clinically relevant results.

REFERENCES

[1] Y. Li *et al.*, “Patterns of somatic structural variation in human cancer genomes,” *Nature*, vol. 578, no. 7793, pp. 112–121, Feb. 2020, doi: 10.1038/s41586-019-1913-9.

[2] J. K. Sehn, “Chapter 9 - Insertions and Deletions (Indels),” in *Clinical Genomics*, S. Kulkarni and J. Pfeifer, Eds., Boston: Academic Press, 2015, pp. 129–150. doi: 10.1016/B978-0-12-404748-8.00009-5.

[3] S. Kosugi, Y. Momozawa, X. Liu, C. Terao, M. Kubo, and Y. Kamatani, “Comprehensive evaluation of structural variation detection algorithms for whole genome sequencing,” *Genome Biol.*, vol. 20, p. 117, Jun. 2019, doi: 10.1186/s13059-019-1720-5.

[4] T. Horiuchi, S. Horiuchi, and A. Novick, “The Genetic Basis of Hyper-Synthesis of β-Galactosidase,” *Genetics*, vol. 48, no. 2, pp. 157–169, Feb. 1963.

[5] E. Darmon and D. R. F. Leach, “Bacterial Genome Instability,” *Microbiol. Mol. Biol. Rev.*, vol. 78, no. 1, pp. 1–39, 2014, doi: 10.1128/mmbr.00035-13.

[6] P. T. West, R. B. Chanin, and A. S. Bhatt, “From genome structure to function: insights into structural variation in microbiology,” *Curr. Opin. Microbiol.*, vol. 69, p. 102192, Oct. 2022, doi: 10.1016/j.mib.2022.102192.

[7] Z. Seferbekova *et al.*, “High Rates of Genome Rearrangements and Pathogenicity of Shigella spp.,” *Front. Microbiol.*, vol. 12, 2021, Accessed: Feb. 03, 2023. [Online]. Available: https://www.frontiersin.org/articles/10.3389/fmicb.2021.628622

[8] P. Hu *et al.*, “Comparative Genomics Study of Multi-Drug-Resistance Mechanisms in the Antibiotic-Resistant Streptococcus suis R61 Strain,” *PLoS ONE*, vol. 6, no. 9, p. e24988, Sep. 2011, doi: 10.1371/journal.pone.0024988.

[9] T. A. Brown, *Genomes*, 2nd ed. Oxford: Wiley-Liss, 2002. Accessed: Sep. 13, 2022. [Online]. Available: http://www.ncbi.nlm.nih.gov/books/NBK21128/

[10] V. Periwal and V. Scaria, “Insights into structural variations and genome rearrangements in prokaryotic genomes,” *Bioinformatics*, vol. 31, no. 1, pp. 1–9, 2015, doi: 10.1093/bioinformatics/btu600.

[11] E. P. C. Rocha, “The Organization of the Bacterial Genome,” *Annu. Rev. Genet.*, vol. 42, no. 1, pp. 211–233, 2008, doi: 10.1146/annurev.genet.42.110807.091653.

[12] S. F. Fitzgerald *et al.*, “Genome structural variation in Escherichia coli O157:H7,” *Microb. Genomics*, vol. 7, no. 11, p. 000682, doi: 10.1099/mgen.0.000682.

[13] A. E. Darling, I. Miklós, and M. A. Ragan, “Dynamics of Genome Rearrangement in Bacterial Populations,” *PLOS Genet.*, vol. 4, no. 7, p. e1000128, 7 2008, doi: 10.1371/journal.pgen.1000128.

[14] D. Hughes, “Evaluating genome dynamics: the constraints on rearrangements within bacterial genomes,” *Genome Biol.*, vol. 1, no. 6, p. reviews0006.1, Dec. 2000, doi: 10.1186/gb-2000-1-6-reviews0006.

[15] J. Repar and T. Warnecke, “Non-Random Inversion Landscapes in Prokaryotic Genomes Are Shaped by Heterogeneous Selection Pressures,” *Mol. Biol. Evol.*, vol. 34, no. 8, pp. 1902–1911, Aug. 2017, doi: 10.1093/molbev/msx127.

[16] P. J. Hastings and S. M. Rosenberg, “In pursuit of a molecular mechanism for adaptive gene amplification,” *DNA Repair*, vol. 1, no. 2, pp. 111–123, Feb. 2002, doi: 10.1016/s1568-7864(01)00011-8.

[17] C. Alkan, B. P. Coe, and E. E. Eichler, “Genome structural variation discovery and genotyping,” *Nat. Rev. Genet.*, vol. 12, no. 5, pp. 363–376, 2011, doi: 10.1038/nrg2958.

[18] A. Soylev, T. M. Le, H. Amini, C. Alkan, and F. Hormozdiari, “Discovery of tandem and interspersed segmental duplications using high-throughput sequencing,” *Bioinformatics*, vol. 35, no. 20, pp. 3923–3930, Oct. 2019, doi: 10.1093/bioinformatics/btz237.

[19] S. Lee, E. Cheran, and M. Brudno, “A robust framework for detecting structural variations in a genome,” *Bioinformatics*, vol. 24, no. 13, pp. i59–i67, Jul. 2008, doi: 10.1093/bioinformatics/btn176.

[20] J. Qi and F. Zhao, “inGAP-sv: a novel scheme to identify and visualize structural variation from paired end mapping data,” *Nucleic Acids Res.*, vol. 39, no. Web Server issue, pp. W567–W575, Jul. 2011, doi: 10.1093/nar/gkr506.

[21] T. Rausch, T. Zichner, A. Schlattl, A. M. Stütz, V. Benes, and J. O. Korbel, “DELLY: structural variant discovery by integrated paired-end and split-read analysis,” *Bioinformatics*, vol. 28, no. 18, pp. i333–i339, Sep. 2012, doi: 10.1093/bioinformatics/bts378.

[22] R. E. Mills *et al.*, “An initial map of insertion and deletion (INDEL) variation in the human genome,” *Genome Res.*, vol. 16, no. 9, pp. 1182–1190, Sep. 2006, doi: 10.1101/gr.4565806.

[23] J. Schröder *et al.*, “Socrates: identification of genomic rearrangements in tumour genomes by re-aligning soft clipped reads,” *Bioinformatics*, vol. 30, no. 8, pp. 1064–1072, Apr. 2014, doi: 10.1093/bioinformatics/btt767.

[24] B. Liu *et al.*, “Structural variation discovery in the cancer genome using next generation sequencing: computational solutions and perspectives,” *Oncotarget*, vol. 6, no. 8, pp. 5477–5489, Mar. 2015, doi: 10.18632/oncotarget.3491.

[25] P. Medvedev, M. Stanciu, and M. Brudno, “Computational methods for discovering structural variation with next-generation sequencing,” *Nat. Methods*, vol. 6, no. 11S, pp. S13–S13, 2009, doi: 10.1038/nmeth.1374.

[26] D. R. Smith *et al.*, “Rapid whole-genome mutational profiling using next-generation sequencing technologies,” *Genome Res.*, vol. 18, no. 10, pp. 1638–1642, Oct. 2008, doi: 10.1101/gr.077776.108.

[27] A. Magi, L. Tattini, T. Pippucci, F. Torricelli, and M. Benelli, “Read count approach for DNA copy number variants detection,” *Bioinformatics*, vol. 28, no. 4, pp. 470–478, 2012, doi: 10.1093/bioinformatics/btr707.

[28] L. Tattini, R. D’Aurizio, and A. Magi, “Detection of Genomic Structural Variants from Next-Generation Sequencing Data,” *Front. Bioeng. Biotechnol.*, vol. 3, p.

92, Jun. 2015, doi: 10.3389/fbioe.2015.00092.

[29] L. Zhang, W. Bai, N. Yuan, and Z. Du, “Comprehensively benchmarking applications for detecting copy number variation,” *PLoS Comput. Biol.*, vol. 15, no. 5, pp. 1–12, 2019, doi: 10.1371/journal.pcbi.1007069.

[30] J. Duan, J.-G. Zhang, H.-W. Deng, and Y.-P. Wang, “Comparative Studies of Copy Number Variation Detection Methods for Next-Generation Sequencing Technologies,” *PLoS ONE*, vol. 8, no. 3, pp. e59128–e59128, 2013, doi: 10.1371/journal.pone.0059128.

[31] S. Yoon, Z. Xuan, V. Makarov, K. Ye, and J. Sebat, “Sensitive and accurate detection of copy number variants using read depth of coverage,” *Genome Res.*, vol. 19, no. 9, pp. 1586–1592, 2009, doi: 10.1101/gr.092981.109.

[32] M. Zhao, Q. Wang, Q. Wang, P. Jia, and Z. Zhao, “Computational tools for copy number variation (CNV) detection using next-generation sequencing data: features and perspectives - Springer,” *BMC Bioinformatics*, vol. 14 Suppl 1, no. Suppl 11, pp. S1–S1, 2013, doi: 10.1186/1471-2105-14-S11-S1.

[33] Y.-C. Wei and G.-H. Huang, “CONY: A Bayesian procedure for detecting copy number variations from sequencing read depths,” *Sci. Rep.*, vol. 10, no. 1, p. 10493, Jun. 2020, doi: 10.1038/s41598-020-64353-1.

[34] M. Pendleton *et al.*, “Assembly and diploid architecture of an individual human genome via single-molecule technologies,” *Nat. Methods*, vol. 12, no. 8, pp. 780–786, Aug. 2015, doi: 10.1038/nmeth.3454.

[35] M. Pirooznia, F. S. Goes, and P. P. Zandi, “Whole-genome CNV analysis: advances in computational approaches,” *Front. Genet.*, vol. 6, p. 138, 2015, doi: 10.3389/fgene.2015.00138.

[36] R. Jugas *et al.*, “CNproScan: Hybrid CNV detection for bacterial genomes,” *Genomics*, vol. 113, no. 5, pp. 3103–3111, Sep. 2021, doi: 10.1016/j.ygeno.2021.06.040.

[37] H. Li, “Aligning sequence reads, clone sequences and assembly contigs with BWA-MEM,” *ArXiv13033997 Q-Bio*, May 2013, Accessed: Apr. 15, 2022. [Online]. Available: http://arxiv.org/abs/1303.3997

[38] H. Li *et al.*, “The Sequence Alignment/Map format and SAMtools,” *Bioinformatics*, vol. 25, no. 16, pp. 2078–2079, 2009, doi: 10.1093/bioinformatics/btp352.

[39] C. Pockrandt, M. Alzamel, C. S. Iliopoulos, and K. Reinert, “GenMap: ultra-fast computation of genome mappability,” *Bioinforma. Oxf. Engl.*, vol. 36, no. 12, pp. 3687–3692, Jun. 2020, doi: 10.1093/bioinformatics/btaa222.

[40] H. Luo and F. Gao, “DoriC 10.0: an updated database of replication origins in prokaryotic genomes including chromosomes and plasmids,” *Nucleic Acids Res.*, vol. 47, no. D1, pp. D74–D77, Jan. 2019, doi: 10.1093/nar/gky1014.

[41] Y. Benjamini and T. P. Speed, “Summarizing and correcting the GC content bias in high-throughput sequencing,” *Nucleic Acids Res.*, vol. 40, no. 10, pp. 1–14, 2012, doi: 10.1093/nar/gks001.

[42] T. Derrien *et al.*, “Fast Computation and Applications of Genome Mappability,” *PLOS ONE*, vol. 7, no. 1, p. e30377, Spring 2012, doi: 10.1371/journal.pone.0030377.

[43] “SAM/BAM and related specifications.” samtools, Dec. 19, 2022. Accessed: Dec. 29, 2022. [Online]. Available: https://github.com/samtools/hts-specs

[44] O. Brynildsrud, L. G. Snipen, and J. Bohlin, “CNOGpro: Detection and quantification of CNVs in prokaryotic whole-genome sequencing data,” *Bioinformatics*, vol. 31, no. 11, pp. 1708–1715, 2015, doi: 10.1093/bioinformatics/btv070.

[45] W. Huang, L. Li, J. R. Myers, and G. T. Marth, “ART: A next-generation sequencing read simulator,” *Bioinformatics*, vol. 28, no. 4, pp. 593–594, 2012, doi: 10.1093/bioinformatics/btr708.

[46] R. M. Layer, C. Chiang, A. R. Quinlan, and I. M. Hall, “LUMPY: a probabilistic framework for structural variant discovery,” *Genome Biol.*, vol. 15, no. 6, p. R84, Jun. 2014, doi: 10.1186/gb-2014-15-6-r84.

[47] A. Abyzov, A. E. Urban, M. Snyder, and M. Gerstein, “CNVnator: an approach to discover, genotype, and characterize typical and atypical CNVs from family and population genome sequencing,” *Genome Res.*, vol. 21, no. 6, pp. 974–984, Jun. 2011, doi: 10.1101/gr.114876.110.

[48] K. Ye, M. H. Schulz, Q. Long, R. Apweiler, and Z. Ning, “Pindel: a pattern growth approach to detect break points of large deletions and medium sized insertions from paired-end short reads,” *Bioinformatics*, vol. 25, no. 21, pp. 2865–2871, Nov. 2009, doi: 10.1093/bioinformatics/btp394.

[49] C. Xie and M. T. Tammi, “CNV-seq, a new method to detect copy number variation using high-throughput sequencing,” *BMC Bioinformatics*, vol. 10, no. 1, p. 80, Mar. 2009, doi: 10.1186/1471-2105-10-80.

[50] G. Klambauer *et al.*, “cn.MOPS: mixture of Poissons for discovering copy number variations in next-generation sequencing data with a low false discovery rate,” *Nucleic Acids Res.*, vol. 40, no. 9, p. e69, May 2012, doi: 10.1093/nar/gks003.

[51] K. Lin, S. Smit, G. Bonnema, G. Sanchez-Perez, and D. de Ridder, “Making the difference: integrating structural variation detection tools,” *Brief. Bioinform.*, vol. 16, no. 5, pp. 852–864, Sep. 2015, doi: 10.1093/bib/bbu047.

[52] L. T. Fang *et al.*, “An ensemble approach to accurately detect somatic mutations using SomaticSeq,” *Genome Biol.*, vol. 16, no. 1, p. 197, Sep. 2015, doi: 10.1186/s13059-015-0758-2.

[53] F. Mölder *et al.*, “Sustainable data analysis with Snakemake.” F1000Research, Apr. 19, 2021. doi: 10.12688/f1000research.29032.2.

[54] A. R. Quinlan and I. M. Hall, “BEDTools: a flexible suite of utilities for comparing genomic features,” *Bioinforma. Oxf. Engl.*, vol. 26, no. 6, pp. 841–842, Mar. 2010, doi: 10.1093/bioinformatics/btq033.

[55] D. C. Jeffares *et al.*, “Transient structural variations have strong effects on quantitative traits and reproductive isolation in fission yeast,” *Nat. Commun.*, vol. 8, no. 1, Art. no. 1, Jan. 2017, doi: 10.1038/ncomms14061.

[56] J. Eisfeldt, “SVDB.” Dec. 30, 2022. Accessed: Jan. 15, 2023. [Online]. Available: https://github.com/J35P312/SVDB

CURRICULUM VITAE

Personal Information	
Name / Surname/ Title	Ing. Robin Jugas
E-mail	robinjugas@gmail.com
Nationality	Czech
Date of birth	19.03.1991
Education	
Doctoral Study	2016-so far, Biomedical Technology and Bioinformatics
Name of the organization	Brno University of Technology, Department of Biomedical Engineering
Thesis theme	Utilization of Genomic Signal Processing Techniques for De-Novo Assembly and Detection of Structural Variations in Bacterial Genomes Supervisor: Ing. Helena Škutková, Ph.D.
Graduate study	2014-2016, Biomedical Engineering and Bioinformatics, Ing.
Name of the organization	Brno University of Technology, Department of Biomedical Engineering
Thesis theme	Signal Processing Based Methods for Genome Assembly Refinement Supervisor: Mgr. Ing. Karel Sedlář, Ph.D.
Undergraduate study	2011-2014, Biomedical Engineering and Bioinformatics, Bc.
Name of the organization	Brno University of Technology, Department of Biomedical Engineering
Thesis theme	Digital Processing of Plant Genomes Supervisor: Mgr. Ing. Karel Sedlář
Work experience	
Dates, Position	09/2020 – so far, Laboratory technician
Main activities and responsibilities	Bioinformatics analysis of Human methylation data, Human genomes variant and SV calling, RNA array data analysis
Name of employer	CEITEC MU, Ondřej Slabý Research Group
Dates, Position	07/2014– 08/2015 IT support
Main activities and responsibilities	Linux server administrator, network administrator, HW maintenance
Name of employer	SDB Brno (Youth free-time center)
Training courses	
	2018 NGS School
	2017 Summer school of mathematical biology IBA MU
Academic/scientific activities	
Scientific impact	WoS records: 7, h-index: 2, citations number: 15
Researcher ID	O-4299-2017
Google Scholar	https://scholar.google.com/citations?user=9BPJp_wAAAAJ
ORCID	0000-0003-4675-0985
GitHub	https://github.com/robinjugas

LIST OF OWN PUBLICATIONS

Journal articles

JUGAS R, SEDLAR K, VITEK M, NYKRYNOVA M, BARTON V, BEZDICEK M, LENGEROVA M, SKUTKOVA H. CNproScan: Hybrid CNV detection for bacterial genomes. Genomics. 2021 Sep;113(5):3103-3111. doi: 10.1016/j.ygeno.2021.06.040. Epub 2021 Jul 3. PMID: 34224809 IF 4.31 Q2 (2021)

MADERANKOVA D, JUGAS R, SEDLAR K, VITEK M, SKUTKOVA H. Rapid Bacterial Species Delineation Based on Parameters Derived From Genome Numerical Representations. Comput Struct Biotechnol J. 2019 Jan 9;17:118-126. doi: 10.1016/j.csbj.2018.12.006. PMID: 30728919; PMCID: PMC6352304. IF 6.18 Q1 (2019)

SKUTKOVA H, MADERANKOVA D, SEDLAR K, JUGAS R, VITEK M. A degeneration-reducing criterion for optimal digital mapping of genetic codes. Comput Struct Biotechnol J. 2019 Mar 19;17:406-414. doi: 10.1016/j.csbj.2019.03.007. PMID: 30984363; PMCID: PMC6444178. IF 6.18 Q1 (2019)

Conference proceedings

JUGAS, R.; VÍTEK, M.; MADĚRÁNKOVÁ, D.; ŠKUTKOVÁ, H. Signal processing based CNV detection in bacterial genomes. In Bioinformatics and Biomedical Engineering. IWBBIO 2019. Lecture Notes in Computer Science. Granada, Spain: Springer Verlag, 2019. p. 93-102. ISBN: 978-3-030-17937-3. ISSN: 0302-9743.

JUGAS, R. Application of Optimization Algorithms to the Genome Assembly. In Proceedings of the 24th Conference STUDENT EEICT 2018. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních, 2018. p. 595-599. ISBN: 978-80-214-5614-3.

JUGAS, R.; VÍTEK, M.; SEDLÁŘ, K.; ŠKUTKOVÁ, H. Cross-Correlation Based Detection of Contigs Overlaps. In MIPRO 2018 proceedings. New York: IEEE, 2018. p. 155-158. ISBN: 978-953-233-095-3.

JUGAS, R. DNA reads feature extraction analysis. In Proceedings of IEEE Student Branch Conference Mikulov 2017. 2017. p. 33-36. ISBN: 978-80-214-5526-9.

JUGAS, R.; SEDLÁŘ, K.; ŠKUTKOVÁ, H.; VÍTEK, M. Overlap detection for a genome assembly based on genomic signal processing. In 2017 IEEE 30th International Symposium on Computer-Based Medical Systems. Proceedings of the ... IEEE International Symposium on Computer-Based Medical Systems. USA: IEEE Computer Society, 2017. p. 301-305. ISBN: 978-1-5386-1710-6. ISSN: 2372-9198.

JUGAS, R. Číslicové zpracování genomických signálů pro klasifikaci rostlin. In Proceedings of the 20th conference Student EEICT 2014. Brno: LITERA, 2014. p. 132-134. ISBN: 978-80-214-4922-0.

Posters

JUGAS, R.; SEDLÁŘ, K.; VÍTEK, M.; ŠKUTKOVÁ, H. CNV Hybrid Detection in Bacteria Using Signal Processing. International Conference of the Genetics Society of Korea 2019 : New Horizons of Genetics. 1. Seoul, Republic of Korea: The Genetics Society of Korea, 2019. p. 101-101.