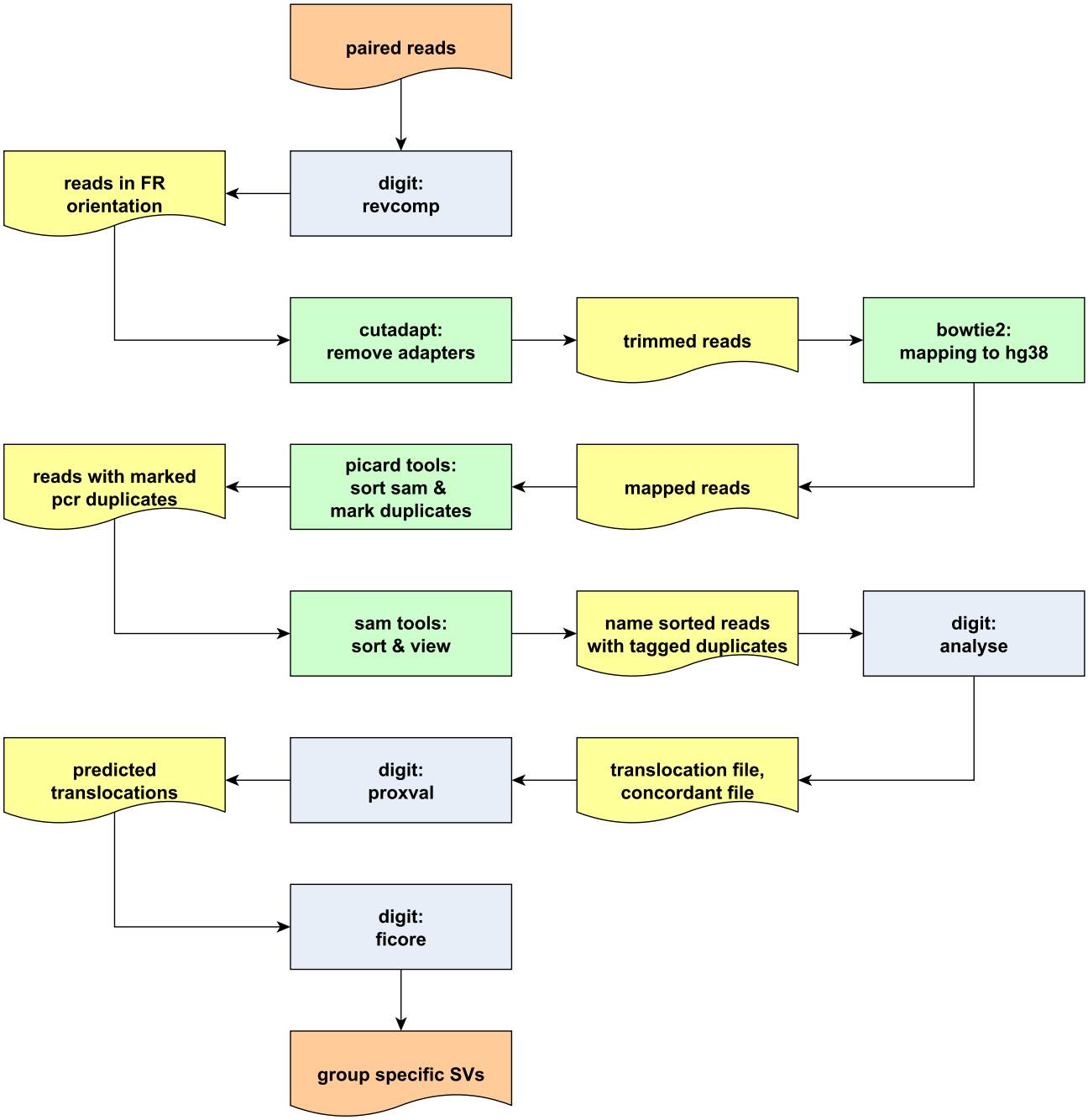
digit

Detection and **I**dentification of **G**enom**i**c **T**ranslocations

User Manual

Richard Meier

1 Workflow of the Pipeline



This workflow was designed for processing mate-pair samples utilizing Bowtie2.0 as primary alignment software. Note that configuration and utilization of modules can change according to the properties of the sample. Regular paired end samples will not require the revcomp step, for example. We thus recommend to read through the documentation and look at the example scripts in order to determine which configuration is appropriate for your analysis. A complete, exemplary script of the depicted workflow can be found in the last chapter of this manual.

2 Requirements and Recommended Software

In terms of computer ressources we recommend 16 gigabytes of ram or more but no less than 8 gigabytes in order to properly process genomic samples with digit. This predominantly concerns the proxval module. If memory related issues arrise when trying to utilize the proxval module for large samples, we recommend splitting up read pairs into separate files by their associated chromosomes with the chrospl module.

Java The pipeline was built in the Java programming language an requires Java SE 7 (or Java JDK 1.7) or higher to run.

Adapter Trimming Adapter trimming was performed with the tool "**cutadapt**" for which a documentation can be found on the project's web site (`http://code.google.com/p/cutadapt/`). This is an optional step that we recommend including into the user's workflow. It had minor effects during test runs but can vary in its impact according to the properties and quality of each sample.

Alignment The tool was built to work with "**Bowtie2.0**" (`http://bowtie-bio.sourceforge.net/bowtie2/index.shtml`) as a primary alignment tool to map the paired reads to the reference genome. Initially Bowtie2.0 was found to perform better than the BWA aln algorithm. However, it should be noted that strong evidence suggests that the BWA mem algorithm significantly outperforms both.
"**BWA mem**" is the recommented as primary alignment method if read lengths ≥ 70 are available (`http://bio-bwa.sourceforge.net/`). Note that when utilizing BWA mem, a low MVM threshold (≤ 0.0005) or disabling of the MVM filter is advisable in order to maintain high sensitivity.

Duplicate Removal One of the most important preparation steps that has to be executed after the reads have been mapped to the reference and before the pipeline can be run is marking of pcr duplicates. For this task we use **Picard Tools'** (`http://picard.sourceforge.net/`) MarkDuplicates function and SAM Tools `http://samtools.sourceforge.net/` to convert the data back into our target format. Note that removal of marked PCR duplicates is handled by the analyse module.

```
java -jar /home/picard-tools/SortSam.jar INPUT=unsorted.sam OUTPUT=sorted.sam
        SORT_ORDER=coordinate
java -jar /home/picard-tools/MarkDuplicates.jar INPUT=sorted.sam OUTPUT=marked.sam
        ASSUME_SORTED=true
/home/sam-tools/samtools sort -n makred.sam nmsorted.bam
/home/sam-tools/view sort -o final.sam nmsorted.bam
```

3 Setup

The software is provided as an executable jar file, that contains all modules. The ready to use jar file can either directly be downloaded from the GitHub project page (*.../digit-trl/executables/digit.jar*) or manually be built from source.

In order to build the software from source the Apache Ant software tool and the Java SE Developement Kit 7 or higher is required. After cloning or downloading the project repository from GitHub open your system's command line interpreter and navigate to the project source code folder (*.../digit-trl/digit*). Make sure that the directory *.../digit-trl/executables* exists on your local machine before building the source. After executing the following statement:

```
ant -buildfile build.xml
```

...the code wil be compiled and a runnable jar file will be created in the *executables* subdirectory.

4 Modules

The pipeline is divided into different modules that can be utilized by executing the main program and specifying the module's name and subsequent parameters.

```
java -jar /home/programs/digit/digit.jar moduleName [parameters]
```

Parameters for the module are specified by a tag that always starts with a dash followed by its argument. All options are separated by space characters.

```
java -jar /home/programs/digit/digit.jar moduleName -p1 arg1 -p2 arg2 -p3 arg3 ...
```

4.1 index - build digit index

Creates a digit specific index file of a genome that helps the pipeline to grab subsequences quickliy via random file access. This index is required for the "proxval" module. The following files are created:

file name	description
indexFile.idx	digit index file
indexFile.idx.ChrLen.txt	chromosome length file

```
index -g genomeFile.fa -o indexFile.idx
```

parameters:

-g genome file: fasta file that contains the sequence information of all chromosomes of a target genome

-o output file: created index of the target genome

4.2 revcomp - build reverse complements

Most aligners require read pairs to be in forward-reverse (FR) orientation. Since mate-pair reads usually are in RF orientation, mate pair data requires reads to be turned into their reverse complements. This issue can be resolved by using the "revcomp" module.

```
revcomp -i fileWithReads.fastq -o fileWithReads_rv.fastq
```

parameters:

-i input file: fastq file with reads that have to be turned into reverse complements

-o output file: created reverse complements in fastq format

4.3 analyse - divide mapped reads into categories

This module divides mapped reads into different categories. Low quality reads, discordants and concordants are separated and calculations that are necessary for clustering are performed. The following files will be created:

file name	description
prefix_conc.sam	concordant read pairs
prefix_del.sam	potential deletion read pairs
prefix_ins.sam	potential insertion read pairs
prefix_inv.sam	potential inversion read pairs
prefix_trl.sam	potential translocation read pairs
prefix_softclips.sam	soft clipping read pairs (legacy feature)
prefix_softsummary.fq	soft clipping read pairs in fastq format with additional information coded in tags. (legacy feature)
prefix_summary.txt	summary with statistics about the separation distance distribution
MAPQ-Dist.txt	contains MAPQ values of the first 1,000,000 reads.
SEP-Dist	read pair separation distance values of the first 1,000,000 reads.

```
analyse -i mappedReads.sam -o /home/results/sample01/ -s 2.33 -a false
        -l lowComplexity.bed
```

parameters:

-i input file: mapped reads in SAM format. Reads of the same pair have to have the same QNAME and follow one another in subsequent lines. Note that read orientation information has to be present in the SAM FLAG values in order for discordant pairs to be correctly picked up and categorized.

-o output path: directory in which all output files will be stored

-s sigma factor: factor multiplied with the standard deviation of the separation distance between read pairs in order to estimate proximity thresholds for clustering [2.33]

-a aln subsetting: if true, subset to reads with the XT:A:U tag. This legacy setting for bwa aln should be false in most workflows. [true, false]

-lc lc file: a bed file that contains the annotation for low complexity regions in the genome. Any read pair that contains at least one read mapping to such an area will be excluded from the analysis. Note that this can also be used for custom workflows in order to exclude any set of target regions from the analysis.
We chose the format provided by the repeat masker track of the ucsc genome browser (`http://genome.ucsc.edu/`). In order to obtain the same file used in our workflow navigate to the table browser, chose your target reference genome and select...
group: Repeats, track: RepeatMasker, region:genome
and add a filter with "repClass does match Low_complexity" as restriction.
Low complexity filtering can be disabled by utilizing an empty text file.

-q cut-off mapping quality: all read pairs that contain reads with a mapping quality (MAPQ) lower than this value will be discarded as low quality reads. Utilizing mapping quality as filter for SV discovery was highly conservative in our experiments. We recommend to not filter based on mapping quality (cut-off value of 0 or less) in most cases and to chose thresholds carefully in custom workflows. [0]

-r chromosome length file: file specifying the names and lengths of all chromosomes in the target reference genome (this file will also be generated by the index module). Each line corresponds to a chromosome where names and lengths are separated by a whitespace character. If no file is specified a hardcoded table of lengths from the human reference genome hg38 will be used.

4.4 proxval - build proximity clusters

This module tries to find clusters of read pairs that represent the same structural variation. The module performs filters based on cluster size, MVM and origin duplicity in order to remove false positive events. Input of the method is the "prefix_trl.sam" file that is created by the analyse module. Structural variations of the type specified by the file will be stored in the output directory. The following files will be created:

file name	description
outFile	output file: containing the remaining reads after the module has finished
outFile.circos.txt	cluster file: reported structural variations after the module has finished in circos format
outFile.reclustered.txt	pairs after the second clustering cycle
outFile.reclustered.txt.circos.txt	clusters after the second clustering cycle
outFile.preclustered.txt	pairs after the first clustering cycle
outFile.preclustered.txt.circos.txt	clusters after the first clustering cycle
approved_pairs_cX.txt	mvm values of all approved, preclustered read pairs
pairs_remapped_cX.txt	mvm values of all preclustered, discordant read pairs
pairs_remapped_conc.txt	mvm values of the first 100k concordant read pairs
rejected_pairs_cX.txt	mvm values of all rejected, preclustered read pairs
prox_log.txt	log file of the module process

```
proxval -i prefix_trl.sam -o clusteredReads.sam -t 4854 -c 3 -g genome.fa -x genome.idx
        -T 100 -M 0.005
```

parameters:

-i input file: target sv category sam file created by the "analyse" module

-o output file: file that stores all reads that contribute to called structural variations. All other output files will also be created in the same folder as the output file.

-t proximity threshold: estimated separation distance that is bigger than most other separation distances. Used to determine when the clustering algorithm stops to extend search regions. It is also used to grab extended mapping target regions from the reference in order to calculate MVM values. The optimal value for this parameter varies according to the properties of the sample. An estimate of the optimal value is calculated by the "analyse" module and can be found in the last line of the "prefix_summary.txt" file. We recommend to always use this estimated value.
Thus, instead of specifying a numeric value "file=prefix_summary.txt" can be used as argument to automatically parse out the estimated optimal value from the generated summary file.

-c cluster size threshold: Determines the required number of read pairs that support a cluster in order to call a structural variation. A value of 3 was sufficient in most of our experiments but high coverage or low coverage samples may require either increasing or decreasing the threshold. [3]

-g genome file: target reference genome the reads were mapped to

-x index file: digit index file created by the "index" module for the genome specified by -g

-T template length: maximum length of the sequenced reads (for example: 50, 75, 100).

-M MVM threshold: percentage of the concordant MVM cumulative distribution function that is used as threshold to reject discordant read pairs with a low MVM. Specifying a negative number as argument will lead to the MVM filtering step being skipped. [0.005]

-an anchor span: specifies the maximum cluster extension distance. Each cluster is built around one anchor read pair and extended in both directions from both reads in said pair. The anchor span sets a limit to the size of the extension. This is specifically relevant when processing high coverage datasets for which there may never be a sufficiently big gap for the window extension to terminate. In this case the anchor span should be reduced to increase processing speed. [10000000]

-r chromosome length file: file specifying the names and lengths of all chromosomes in the target reference genome (this file will also be generated by the index module). Each line corresponds to a chromosome where names and lengths are separated by a whitespace character. If no file is specified a hardcoded table of lengths from the human reference genome hg38 will be used.

memory issues:

Running this module requires a lot of memory. When using reference genome hg38 we recommend to allocate at least 3GB of memory or else the program is likely to terminate prematurely and throw errors. The maximum amount of memory Java is allowed to allocate can be increased via the "java -Xmx" option:

java -Xmx3g -jar /scratch/digit/digit.jar proxval ...	allow up to 3GB of RAM
java -Xmx2560m -jar /scratch/digit/digit.jar proxval ...	allow up to 2560MB (i.e. 2.5GB) of RAM

Potential errors related to memory allocation:

```
Exception in thread "main" java.lang.OutOfMemoryError: Java heap space
at ... at ... at ... ...
```

The program reached the maximum amount of memory that Java is allowed to allocate. Increase the maximum with the "-Xmx" option and rerun the module.

```
Error occured during initialization of VM
Could not reserve enough space for ...KB object heap
```

Java failed to allocate the specified amount of memory. Check your computer's current memory usage and close programs with high memory use. You should always allocate less memory than is currently available. Even if the specified amount of memory does not surpass the computer's currently available amount, allocation can sometimes still fail. Java always tries to allocate a continous block of memory, but memory often is fragmented into used and available blocks. If none of the free blocks is large enough allocation fails. Close other currently running programs and/or try restarting your computer.

4.5 ficore - find common regions

This module compares predicted SVs from different samples to detect and separate common events associated with the target disease from common, unspecific events that frequently occur in normal samples. Results are directed to standard output.

```
ficore -D sampleD1.circos.txt,sampleD2.circos.txt,sampleD3.circos.txt
       -N sampleN1.circos.txt,sampleN2.circos.txt,sampleN3.circos.txt
```

parameters:

-D disease files: comma delimited list of sv call files associated with the same target disease

-N normal files: comma delimited list of files that are unrelated to the target disease

-r chromosome names file: file specifying the names of all chromosomes in the target reference genome. Each line corresponds to a chromosome where a name is always the first entry of the line and entries are separated by a whitespace character. The chromosome length file from the previous modules can be used. If no file is specified a hardcoded list of names from the human reference genome hg38 will be used.

-lof library output file: A library of common, unspecific events that occur either exclusively in the normal files or in normal and disease files at the same time will be outputted to this target destination. No library will be generated if this option is not specified. Library files can be shared and utilized as normal files in future runs of the module. (SIGNIFICANTLY SLOWS DOWN THE PROCESS.)

-C merging categories: category list for file merging. This option is useful when sv calling data from multiple files is to be merged into and then processed as a single super sample. It was specifically designed to merge samples of multiple sequencing runs from the same individual or organism. It allows such samples to be individually processed until right before the ficore analysis and only then to be merged, instead of combining them all upstream. This is meaningful since sequencing quality and sample properties may not only differ between batches but also depend on the sequencing technology and other methods employed.
List elements are separated by colons. Each element represents a merged super sample and is a list of files that are separated by commas (analogous to the -D and -N option). The category list is thus for all intends and purposes a list of lists of files.

An additional option for files in the -D -N -C lists is providing a library file containing multiple samples. Libraries are a set of super clusters in modified circos format consisting of merged, overlapping clusters from various samples. Libraries of common, unspecific events can be created by the ficore module. A library file is specified by using the "@" character as a prefix before specifying the file name.

4.6 mvmfo - independent mvm filter

This module filters read pairs according to their MVM.

```
mvmfo -i sample_trl.sam -o sample.circos.txt -T 100 -t 4101 -g genome.fa
      -x genome.idx -M 0.005 -r genome.idx.ChrLen.txt -p 3
```

parameters:

-i input file: discordant read pair sam file

-o output file: file that all filtered read pairs are written to

-t proximity threshold: Same recommendations go as for the -t parameter of the proxval module

-g genome file: target reference genome the reads were mapped to

-x index file: digit index file created by the "index" module for the genome specified by -g

-T template length: maximum length of the sequenced reads (for example: 50, 75, 100).

-M MVM threshold: percentage of the concordant MVM cumulative distribution function that is used as threshold to reject discordant read pairs with a low MVM. [0.005]

-p process thread number: the number of threads that will be used to calculate MVM thresholds. [1]

-r chromosome length file: file specifying the names and lengths of all chromosomes in the target reference genome (this file will also be generated by the index module). Each line corresponds to a chromosome where names and lengths are separated by a whitespace character.

4.7 chrospl - split files by chromosomes

This optional module splits up read pair files into two files according to a set of chromosomes. All read pairs that are associated with a chromosome in the set go into the first file, while the other read pairs go into the second file. This is useful when dealing with samples for which large amounts of discordant pairs are called. Subsetting the discordant read pair files and individually processing them with the proxval module makes the workflow more performant and require less memory.
It is important that in the input file reads that belong to the same pair are in consecutive lines. Input and output of the module are headerless sam files.

```
chrospl -i trl.sam -o1 trl_sp1.sam -o2 trl_sp2.sam -s chr1,chr2,chr3,chr4
```

parameters:

-i input file: translocation read pair file

-o1 output file 1: first split

-o2 output file 2: second split

-s chomosome set: list of chromosomes that all read pairs in the first split are associated with and read pairs in the second split are not associated with.

5 Exemplary Scripts

5.1 Bowtie2.0 + MVM filtering

This is an exemplary workflow for a mate-pair dataset utilizing Bowtie2 and MVM filtering.

```
index -g /scratch/hg38/genome.fa -o /scratch/hg38/genome.idx

java -jar /scratch/digit/digit.jar revcomp -i /scratch/data/s01/reads_R1.fastq -o /scratch/data/s01/reads_R1_rv.fastq
java -jar /scratch/digit/digit.jar revcomp -i /scratch/data/s01/reads_R2.fastq -o /scratch/data/s01/reads_R2_rv.fastq

/scratch/cutadapt/cutadapt -b ACCCAAATTTAAAGGGC -o /scratch/data/s01/reads_R1_at.fastq /scratch/data/s01/reads_R1_rv.fastq
/scratch/cutadapt/cutadapt -b CAATAAGAATGTGTGTG -o /scratch/data/s01/reads_R2_at.fastq /scratch/data/s01/reads_R2_rv.fastq

/scratch/bowtie2/bowtie2 -q -x /scratch/hg38/genome.fa -1 /scratch/data/s01/reads_R1_at.fastq
        -2 /scratch/data/s01/reads_R2_at.fastq -S /scratch/results/s01/mappedReads.sam

java -jar /scratch/picard-tools/SortSam.jar INPUT=mappedReads.sam OUTPUT=sorted.sam
          SORT_ORDER=coordinate

java -jar /scratch/picard-tools/MarkDuplicates.jar INPUT=sorted.sam OUTPUT=marked.sam
          ASSUME_SORTED=true

/scratch/sam-tools/samtools sort -n marked.sam nmsorted.bam
/scratch/sam-tools/view sort -o mapped_rmd.sam nmsorted.bam

rm sorted.sam
rm marked.sam
rm nmsorted.bam

java -jar /scratch/digit/digit.jar analyse -i /scratch/results/s01/mapped_rmd.sam -o /scratch/results/s01/ -s 2.33
        -a false -lc /scratch/hg38/lowComplexity.bed -r /scratch/refs/hg38.idx.ChrLen.txt -q 0

java -Xmx8g -jar /scratch/digit/digit.jar proxval -T 100 -c 3 -i /scratch/results/s01/mapped_rmd_trl.sam
        -o /scratch/results/s01/trl_clusters -g /scratch/hg38/genome.fa -x /scratch/hg38/genome.idx -M 0.005
        -t file=/scratch/results/s01/mapped_rmd_summary.txt -r /scratch/refs/hg38.idx.ChrLen.txt

java -jar /scratch/digit/digit.jar ficore
        -D /scratch/results/s01/trl_clusters.circos.txt,/scratch/results/s02/trl_clusters.circos.txt,...
        -N /scratch/results/normal01/trl_clusters.circos.txt,/scratch/results/normal02/trl_clusters.circos.txt,...
        -r /scratch/refs/hg38.idx.ChrLen.txt > /scratch/results/ficor_eval.txt
```

Alternatively, the proxval parameter -t 4372 could be obtained by doing the following:

```
more /scratch/results/s01/mapped_rmd_summary.txt
...
number of read-pairs: 21537794
recommended cluster threshold: 4372
```

5.2 Robust, high coverage (large sample) configuration

For very large samples or if the required memory of the proxval module is too high, the translocation file can be split up into sub files by chromosomes associated with read pairs. After structural variation calls have been made the cluster file can be concatenated into a single file again.

```
java -jar /scratch/digit/digit.jar analyse -i /scratch/results/s01/mapped_rmd.sam -o /scratch/results/s01/ -s 2.33
        -a false -lc /scratch/hg38/lowComplexity.bed -r /scratch/refs/hg38.idx.ChrLen.txt -q 0

java -jar /scratch/digit/digit.jar chrospl -i /scratch/results/s01/mapped_rmd_trl.sam
        -o1 /scratch/results/s01/splits/mapped_rmd_trl_sp1.sam
        -o2 /scratch/results/s01/splits/mapped_rmd_trl_sp2.sam
        -s chr1,chr2,chr3,chr4

java -jar /scratch/digit/digit.jar chrospl -i /scratch/results/s01/mapped_rmd_trl_sp1.sam
        -o1 /scratch/results/s01/splits/mapped_rmd_trl_sp11.sam
        -o2 /scratch/results/s01/splits/mapped_rmd_trl_sp12.sam
        -s chr5,chr6,chr7,chr8,chr9

rm /scratch/results/s01/mapped_rmd_trl_sp1.sam

java -jar /scratch/digit/digit.jar chrospl -i /scratch/results/s01/mapped_rmd_trl_sp2.sam
        -o1 /scratch/results/s01/splits/mapped_rmd_trl_sp21.sam
        -o2 /scratch/results/s01/splits/mapped_rmd_trl_sp22.sam
        -s chr5,chr6,chr7,chr8,chr9

rm /scratch/results/s01/mapped_rmd_trl_sp2.sam

for arg in "sp11" "sp12" "sp21" "sp22"
do
        mkdir /scratch/results/s01/${arg}
        java -Xmx8g -jar /scratch/digit/digit.jar proxval -T 100 -i /scratch/results/s01/splits/mapped_rmd_trl_${arg}.sam
                -o /scratch/results/s01/${arg}/trl_clusters -g /scratch/hg38/genome.fa -x /scratch/hg38/genome.idx -M 0.005
                -t file=/scratch/results/s01/mapped_rmd_summary.txt -r /scratch/refs/hg38.idx.ChrLen.txt -an 500000  -c 3
done

cat /scratch/results/s01/sp11/trl_clusters.circos.txt /scratch/results/s01/sp12/trl_clusters.circos.txt
        /scratch/results/s01/sp21/trl_clusters.circos.txt /scratch/results/s01/sp22/trl_clusters.circos.txt
        > /scratch/results/s01/trl_clusters_merged.circos.txt
```

5.3 Workflow utilizing BWA mem

BWA mem's built in local realignment seemed to outperform digit's MVM filter. For read lengths of ≥ 70 we thus recommend using BWA mem and either disabling the MVM filter or reducing the MVM threshold.

```
index -g /scratch/hg38/genome.fa -o /scratch/hg38/genome.idx

java -jar /scratch/digit/digit.jar revcomp -i /scratch/data/s01/reads_R1.fastq -o /scratch/data/s01/reads_R1_rv.fastq
java -jar /scratch/digit/digit.jar revcomp -i /scratch/data/s01/reads_R2.fastq -o /scratch/data/s01/reads_R2_rv.fastq

/scratch/cutadapt/cutadapt -b ACCCAAATTTAAAGGGC -o /scratch/data/s01/reads_R1_at.fastq /scratch/data/s01/reads_R1_rv.fastq
/scratch/cutadapt/cutadapt -b CAATAAGAATGTGTGTG -o /scratch/data/s01/reads_R2_at.fastq /scratch/data/s01/reads_R2_rv.fastq

/scratch/bwa/bwa mem -t 7 /scratch/hg38/genome.fa /scratch/data/s01/reads_R1_at.fastq
        /scratch/data/s01/reads_R2_at.fastq > /scratch/results/s01/mappedReads.sam

java -jar /scratch/picard-tools/SortSam.jar INPUT=mappedReads.sam OUTPUT=sorted.sam
          SORT_ORDER=coordinate

java -jar /scratch/picard-tools/MarkDuplicates.jar INPUT=sorted.sam OUTPUT=marked.sam
          ASSUME_SORTED=true

/scratch/sam-tools/samtools sort -n marked.sam nmsorted.bam
/scratch/sam-tools/view sort -o mapped_rmd.sam nmsorted.bam

rm sorted.sam
rm marked.sam
rm nmsorted.bam

java -jar /scratch/digit/digit.jar analyse -i /scratch/results/s01/mapped_rmd.sam -o /scratch/results/s01/ -s 2.33
        -a false -lc /scratch/hg38/lowComplexity.bed -r /scratch/refs/hg38.idx.ChrLen.txt -q 0

java -Xmx8g -jar /scratch/digit/digit.jar proxval -T 100 -c 3 -i /scratch/results/s01/mapped_rmd_trl.sam
        -o /scratch/results/s01/trl_clusters -g /scratch/hg38/genome.fa -x /scratch/hg38/genome.idx -M -1
        -t file=/scratch/results/s01/mapped_rmd_summary.txt -r /scratch/refs/hg38.idx.ChrLen.txt
```