PIMMS 1.9 Handbook
December 2014.

1 MOTIVATION

The PIMMS (Pragmatic Insertional Mutant Mapping System) pipeline has been developed for simple essential genome discovery experiments in bacteria. Capable of using RAW Transposon-Mapping (Tn-mapping) sequence data files alongside a FASTA and GFF/GTF file, PIMMS will generate a tabulated output of each coding sequence with corresponding mapped insertions accompanied with normalised results enabling streamlined analysis. This allows for a quick assay of the genome to identify essential genes on a standard desktop computer prioritising results for further investigation.

2 AVAILABILITY

The PIMMS pipeline script is freely available at. https://github.com/ADAC-UoN/PIMMS

3 CITATION

Tn-mapping with PIMMS – Pragmatic Insertion Mutant Mapping System
Adam M. Blanchard, James A. Leigh, Sharon A. Egan and Richard D. Emes. Submitted.

4 CONTACT

Richard.Emes@nottingham.ac.uk

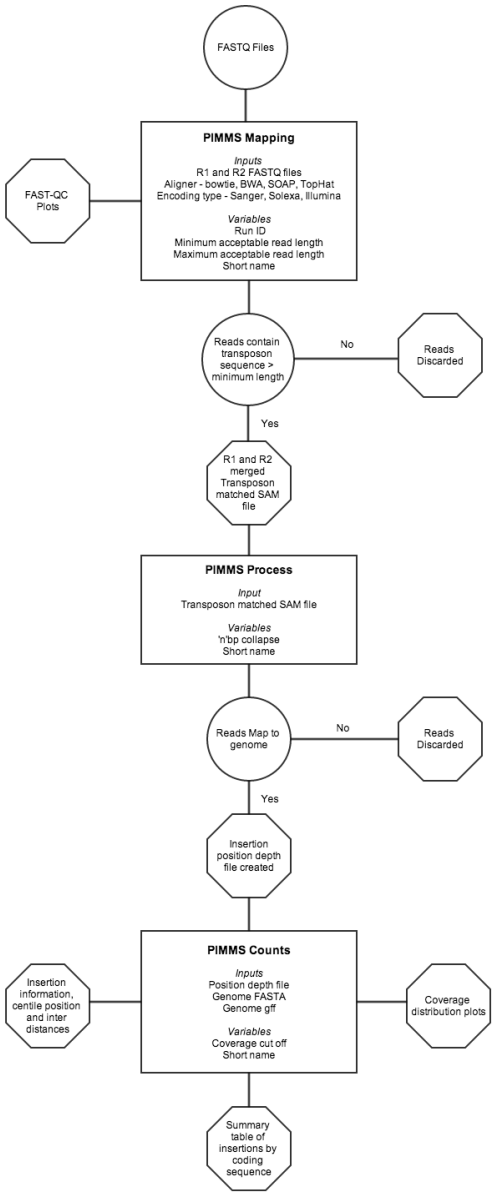
5 DEPENDENCIES

The PIMMS pipeline requires minimal additional software to operate.
The following should be installed prior to running PIMMS
1) Fastxtookit available from http://hannonlab.cshl.edu/fastx_toolkit/
2) To generate plots from fastxtoolkit gnuplot is also required.
3) bwa or bowtie2 aligners (alternatives can be used see (section PIMMS Mapping)
4) The statistical package R
5) Perl packages Getopt::Long, and Statistics::Descriptive

6 FUNCTIONALITY

The pipeline and data formats produced are shown below.

The PIMMS.pl perl script runs four modules using the commands PIMMS.pl –m [module] options where [module] is one of mapping, process.sam, counts or compare.
-v, prints version
-h, --help prints help and usage
typing PIMMS.pl -m [module] without options for module returns module specific help and usage.



7 PIMMS MAPPING

The mapping approach follows three steps
1) Read files are matched to user defined motifs and trimmed to retain potential insertion sequence. Sequence motifs that mark the end of the transposon or inserted element, this sequence must be provided by the user in the command.txt file.
2) Resulting files are mapped to given reference genome using bwa by default (alternate aligners could be used by modifying the PIMMS.commands.txt file).
3) Basic statistics of mapped read positions are produced

7.1 PIMMS mapping User parameters

-c PIMMS.command.txt
-i fastq file 1
-j fastq file 2
-g path to reference genome for bowtie mapping
-e illumina encoding type [sanger, solexa or illumina]
-r run id to be matched in fastq file e.g. "@M01661"
-min minimum length of sequence post trimming to ensure read is retained
-max maximum length of sequence post trimming that is returned
-n short name for identification of files (no spaces in name)

-c	PIMMS.command.txt user to edit to contain pair of motifs and optional aligner options
-i / -j	The fastq files should be in standard format. These files can be gzipped (with *.gz suffix) as these will be uncompressed prior to processing. If single end reads are used write "-j FALSE"
-g	path to reference genome. This is the reference genome the trimmed reads will be mapped to. The file should be in fasta format and should be the same used to build the annotations underlying the GFF file used in the PIMMS.counts part of the pipeline.
-e	illumina encoding type [sanger, solexa or illumina]. Refers to qualities encoded with the CASAVA pipeline. Illumina quality using CASAVA >= 1.8 is Sanger encoded.
-r	Run id to be matched in the fastq file. To view type “less x.fastq” and look for the “@xxxx” name at the start of each read.
-min	[Integer]. Following matching and removal of insertion motif, reads of less than –min value will not be processed further.
-max	[Integer]. Maximum length of reads to return post trimming
-n	Short name for identification of output files

7.2 Output files

1) Log file, a log file named (-n).PIMMS.mapping.log. The log file contains all steps and statistics of each step of the mapping process.
2) If a aligner index file is not present in the same directory as the reference genome, index files will be produced.
3) Raw quality plots. The fastxtoolkit is used to determine and draw plots of read quality of input fastq files.
4) Reads fulfilling match criteria are written to output file with the naming convention (-n).matched.min(-min).max(-max).fastq.
5) To avoid the potential of double counting of a single insertion from paired end reads, if both pairs of a read match the insertion motif, only a single end is retained in the file (-n).PIMMS.processed.se.fastq
6) Processed quality plots. The fastxtoolkit is used to determine and draw plots of read quality of the processed [(-n).PIMMS.processed.se.fastq] fastq file.
7) Reads mapped PIMMS.processed.se.fastq.mapped.sam

These files are saved in directories
PIMMS.processed.reads
Bowtie.index.files
Mapped.reads.sam.file
PIMMS.QC.plots

8 PIMMS PROCESS SAM FILE

Whilst we generally use the PIMMS mapping script to process reads, the PIMMS pipeline can be initiated following any read mapping that produces a standard SAM formatted output. At this processing step, insertion positions can also be collapsed if they are exactly a given distance apart. This is important if the insertional mutagenesis ystem, like the pGhost::ISS1 used to develop this protocol, incorporates a DNA repeat during insertion. The SAM processing script generates a simple text file of insertion coordinates and relative read depth at each unique insertion position.
Insertion positions are determined from the map position of reads. The Sam flags are used to determine if the reads are mapped on the forward or reverse strand and the map positions are determined from these positions. The scripts calculate depth at each unique insertion position.

8.1 PIMMS process SAM file user parameters

-s mapped reads in SAM format
-n short name for identification of output files (no spaces in name, keep short)
-col collapse and add depth of insertion EXACTLY x bp apart. (OPTIONAL use -c N to turn off).
-mis maximum mismatch in read alignment.
-a minimum read alignment quality.

-s	Standard SAM file format (output of bowtie mapping from (-n).PIMMS.processed.se.fastq.mapped.sam
-n	Short name for identification of output files
-col	[integer] unless = N counts exactly –c bp apart are collapsed. The total reads are the sum of reads at both positions. By default the smallest insertion position is retained.
-mis	Maximum mismatch allowed in alignment (filters on \"MD\" tag of sam file) if m = 0 or m >= 1 then counts mismatches if fraction is given then counts proportion of read length ie 0.1 = 10% of read length can mismatch
-a	Minimum alignment quality (this needs to be relevant for aligner used). Reads with AS score less than this are discarded. If using bowtie2 aligner - to avoid problems with using negative numbers on the commandline if you wish to have a minimum < 0 type -a neg[integer] e.g. neg10 will be interpreted as -10.

8.2 Output files

1) Log file, a log file named (–n).PIMMS.processing.collapse.(-c)_bp.log. The log file contains all steps and statistics of each step of the processing script.
2) Positions and depth file named (input.sam.file.name).PIMMS.collapse.(-c) _bp.positions.depths". This is a tab-delimited file of positions and depth at each unique position.

9 PIMMS COUNTS

The PIMMS counts script uses a given gtf/gff file to match annotation to the insertion positions. This is then used to generate tabulated output files of unique insertions, reads depths, insertions per kb, the percentile position of the first and last insert within the coding sequence and normalised read values (NRM and NIM scores). NRM – Normalised Reads Mapped (total number of reads per gene/length of gene in Kb)/(total mapped read count/10^6) and NIM – Normalised Insertions Mapped (total unique insertions mapped per gene/Length of gene in Kb)/(total mapped/10^6) provide a robust indication of gene disruption in comparison to other genes and also takes into account the variability of the number of mapped sequence reads for each sample.

9.1 PIMMS counts user parameters

-d mapped reads position depth file (output of PIMMS.process.SAM.pl)
-r reference genome in fasta format
-g reference gtf/gff file
-m minimum coverage filter (minimum coverage at insertion site to report)
-n short name for identification of files (no spaces in name, use same as for PIMMS.process.SAM.pl)

-d	Reads depth file, tab delimited positions and read depth. Generate file using the process.SAM script.
-g	path to reference genome. This is the reference genome the trimmed reads will be mapped to. The file should be in fasta format and should be the same used to build the annotations underlying the GFF file used in the PIMMS.counts part of the pipeline.
-gtf	Path to reference GTF/GFF file. Within the annotation detail column (column 9 of GFF file). Attempts to retain information for CDS using "locus_tag", " gene" and "product" flags. Assumes standard GFF file format, column 1 = genome, column 3 = data source, column 4 = genome start position, column 5 = genome end position and column 7 = strand.
-cov	[integer] minimum coverage at position to report as an insertion position
-n	Short name for identification of output files

9.2 Output files

1) Log file, a log file named (–n).PIMMS.counts.min_cov.(-m).log. The log file contains all steps and statistics of each step of the counts script.

2) (-n).PIMMS.counts.min_cov.(-m).summary.table
A tab-delimited file of a single line for each locus in the GFF file provided, containing information of

Locus	Locus information from GTF file as determined by -g
Gene	Gene information from GTF file as determined by -g
Start	Gene start position from GTF file as determined by -g
Stop	Gene stop position from GTF file as determined by -g
CDS length	Locus length
Product	Product information from GTF file as determined by -g
Number of mutations	Total number of insertions mapped within this locus
Number of unique mutations (>= [-m] x coverage)	Total number of unique insertion positions mapped within this locus
Unique mutations per1KbCDS (>= [-m] x coverage)	Total number of unique insertion positions mapped within this locus per kb of each locus
First unique insertion centile position (>= [-m] x coverage)	For each unique insertion position, the centile position of that locus is determined. The first (lowest centile) position is returned.
Last unique insertion centile position(>= [-m] x coverage)	For each unique insertion position, the centile position of that locus is determined. The last (greatest centile) position is returned.
Normalised Reads Mapped (NRM score)	= (total number of reads mapped per gene/length of gene in KB)/(total mapped read count/10^6)
Normalised Insertions Mapped (NIM score)	= (total number of unique insertions mapped per gene/length of gene in KB)/(total mapped read count/10^6)

3) (-n).PIMMS.counts.min_cov.(-m).unique.insertion.centile.positions
list of all centile positions
4) (-n).PIMMS.counts.min_cov.(-m).insertion.positions.depths
For each insertion position table of

Insertion Position	Genome position based on reference genome fasta file
Number of reads at position	total number of reads calling this as an insertion position
NIindex	Normalised insertion index (NIindex) calculated as observed insert count per position - Expected insert count per position. Expected insert count = (total mapped read count / unique

	positions count) If Observed insertions = Expected insertions NIindex would equal 0. Deviations from zero indicate greater or fewer reads mapped at position than expected by random distribution of reads.
Observed insert proportion	Observed insert proportion = (read count at position / total mapped reads).
NIPdiff	Observed insert proportion - Expected insert proportion. Expected insert proportion = (1 / number of unique insertion positions). If Observed proportion = Expected proportion NIPdiff would equal 0.
NIPratio	Observed insert proportion / expected insert proportion. If Observed proportion = Expected proportion NIPratio would equal 1.
Gene	Gene information from GTF file as determined by -g
Product	Product information from GTF file as determined by -g
Locus	Locus information from GTF file as determined by -g
Source	source information from GTF file as determined by -g
Position	Centile position

5) (-n).PIMMS.counts.min_cov.(-m).unique.insertion.inter.distances.
list of distances between unique insertion points
6) (-n).PIMMS.counts.(-m).plots.script.R
R script written for generation of plots. This is run within the counts pipeline using the command "R --vanilla --quiet < (-n).PIMMS.counts.(-m).plots.script.R"

Plots generated in pdf format

xx.insertion.positions.depths.pdf	Read depth at each position of genome
xx.insertion.positions.NIindex.pdf	Read depth at each position of genome with expected read depth plotted
xx.insertion.positions.NIindex.Zoom.pdf	Read depth at each position of genome with expected read depth plotted (with reduced scale on y axis)
xx.insertion.positions.NIPdiff.pdf	Read depth at each position of genome with expected read difference plotted
xx.insertion.positions.NIPdiff.zoom.pdf	Read depth at each position of genome with expected difference ratio plotted (with reduced scale on y axis).
xx.insertion.positions.NIPratio.pdf	Read depth at each position of genome with expected read ratio plotted.
xx.insertion.positions.NIPratio.zoom.pdf	Read depth at each position of genome with expected read ratio plotted (with reduced scale on y axis).
xx.inter.insertion.distances.density.plot.pdf	Kernel density plot of the inter-insertion distances.
xx.summary.plots.pdf	NIM and NRM plots
xx.unique.insertion.centile.positions.density.plot.pdf	Kernel density plot of centile position of unique insertion positions.

These files are saved in directories
Insertion.distances.and.depths
Summary.Tables
R.scripts
Coverage.and.Distribution.Plots

10 PIMMS COMPARE

The compare script allows processing of data obtained from phenotypic studies. Using the same output from the counts script, it compares outputs to identify common and unique mutation events between experimental conditions.

10.1 PIMMS compare user parameters

Two files are compared these are designated input and output but could be before and after selection test to identify genes essential to that selection. If the pipeline has been followed these files will be in the directory "Insertion.distances.and.depths"

-in parsed input sample x.PIMMS.counts.min_cov.x.insertion.positions.depths output from PIMMS.counts.vX.pl
-out parsed output sample x.PIMMS.counts.min_cov.x.insertion.positions.depths output from PIMMS.counts.vX.pl
-dist [interger]exact distance to collapse reads for overlap (number of bp or N to switch off) unless = N positions exactly –d bp apart in the input and output files are collapsed and considered as the same insertion. By default the smallest insertion position is retained.
-n comparison short name (keep short no spaces)

10.2 Output files

1) Log file, a log file named (–n).coverage(taken from input file)PIMMS.IO.d(-d).log. The log file contains all steps and statistics of each step of the compare script.
2) (-n).coverage(taken from input file).PIMMS.IO.d(-d).shared.position.table

For each position shared between Input and Output files, a tab-delimited table of:

Position	Genome position based on reference genome fasta file
Input insertion count	Unique number of reads at this position in input file
Output insertion count	Unique number of reads at this position in output file
Input observed proportion of reads	Observed insert proportion in input file = (read count at position / total mapped reads).
Output observed proportion of reads	Observed insert proportion in output file = (read count at position / total mapped reads).
Proportion diff	(Input observed proportion of reads)-(Output observed proportion of reads)
Proportion ratio	(Output observed proportion of reads)/(Input observed proportion of reads)
Gene	Gene information from GTF file as determined by -g
Product	Product information from GTF file as determined by -g in PIMMS.counts.pl
Locus	Locus information from GTF file as determined by -g
Source	Source information from GTF file as determined by –g in PIMMS.counts.pl
Position	Centile position
Zscore	Within an experiment The natural logarithm (base *e*) transformed proportion ratio approximates a normal distribution. Using the mean (shared mean) and standard deviation (shared sd) of this population, for each insertion the input/output proportion ratio (Proportion ratio) the Zscore is calculated as: Zscore = ((log(Proportion ratio)) - (shared mean)) / (shared sd)
Flag	To provide some approximation of statistical significance. Zscores > standard deviations equivalent to a pvalue of 0.001, 0.01, 0.05 are flagged as below. If |Zscore| >= 3.291, Flag = "***" (~ p value = 0.001) If 3.291 > |Zscore| >= 2.579, Flag = "**" (~ p value = 0.01) If 2.579 > |Zscore| >= 1.960, Flag = "*" (~ p value = 0.05) If |Zscore| < 1.960, Flag = "".

3) (-n).coverage(taken from input file).PIMMS.IO.d(-d).Input_only.position.table

For each position shared between Input and Output files, a tab-delimited table of:

Position	Genome position based on reference genome fasta file
Input insertion count	Unique number of reads at this position in input file
Output insertion count	Unique number of reads at this position in output file
Input observed proportion of reads	Observed insert proportion in input file = (read count at position / total mapped reads).
Output observed proportion of reads	Observed insert proportion in output file = (read count at position / total mapped reads).
Proportion diff	(Input observed proportion of reads)-(Output observed proportion of reads)
Proportion ratio	(Output observed proportion of reads)/(Input observed proportion of reads)
Gene	Gene information from GTF file as determined by -g
Product	Product information from GTF file as determined by -g in PIMMS.counts.pl
Locus	Locus information from GTF file as determined by -g
Source	Source information from GTF file as determined by –g in PIMMS.counts.pl
Position	Centile position

4) (-n).coverage(taken from input file).PIMMS.IO.d(-d).Output_only.position.table
For each position shared between Input and Output files, a tab-delimited table of:

Position	Genome position based on reference genome fasta file
Input insertion count	Unique number of reads at this position in input file
Output insertion count	Unique number of reads at this position in output file
Input observed proportion of reads	Observed insert proportion in input file = (read count at position / total mapped reads).
Output observed proportion of reads	Observed insert proportion in output file = (read count at position / total mapped reads).
Proportion diff	(Input observed proportion of reads)-(Output observed proportion of reads)
Proportion ratio	(Output observed proportion of reads)/(Input observed proportion of reads)
Gene	Gene information from GTF file as determined by -g
Product	Product information from GTF file as determined by -g in PIMMS.counts.pl
Locus	Locus information from GTF file as determined by -g
Source	Source information from GTF file as determined by –g in PIMMS.counts.pl
Position	Centile position

These are stored into the directory IO.comparision.Tables