PIMMS 1.9 Handbook
March 2015.

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

Transposon-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

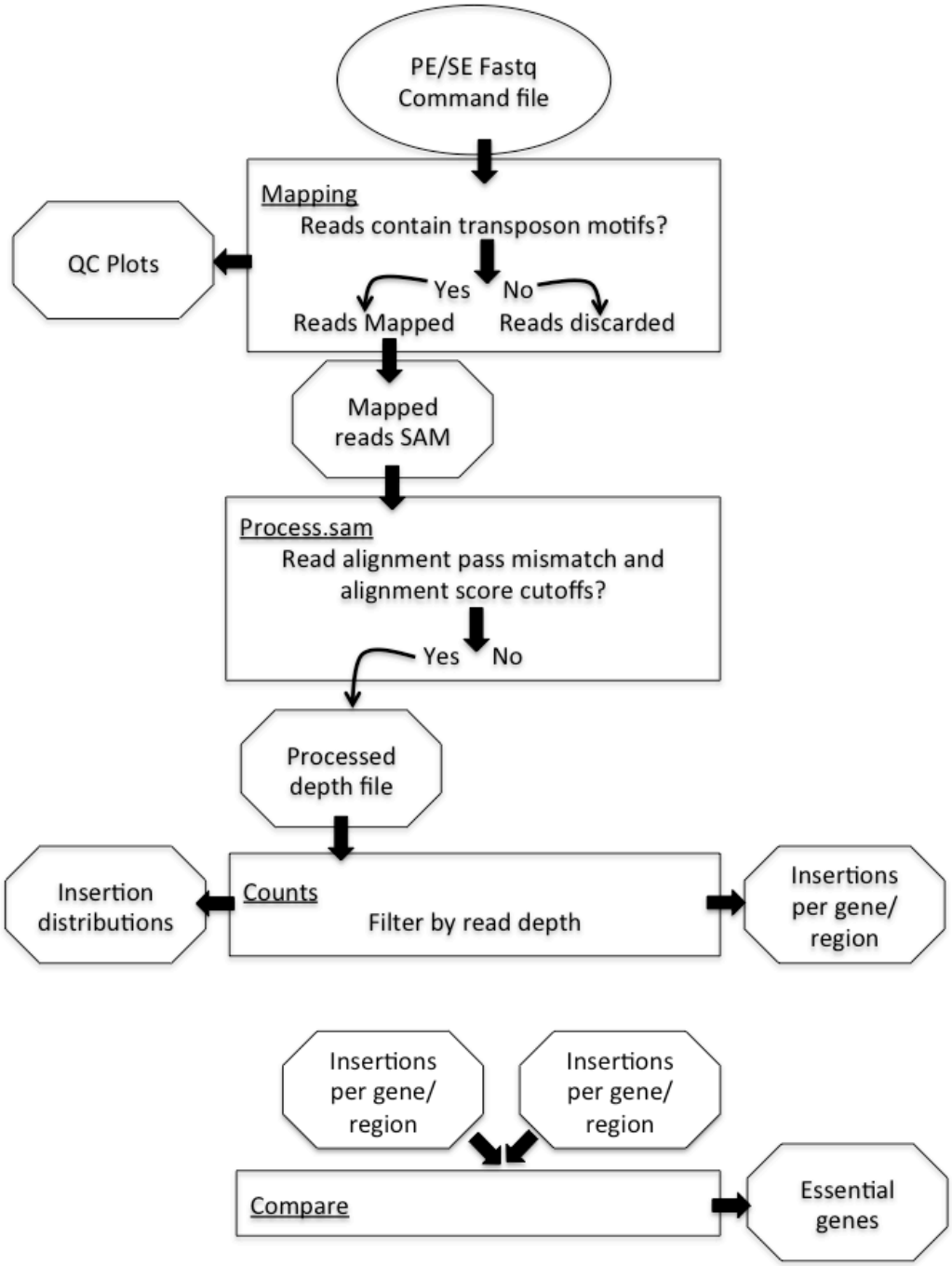
5 DEPENDENCIES

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"
-q generate quality plots Y/N
-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.
-q	Generate quality plots. This uses the fastxtoolkit and can be slow for large input files.
-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 (the share of reads mapped at a location) 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	Observed insert proportion in input file = (read count at position /

proportion of reads	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

11 TUTORIAL

All scripts and example datasets can be obtained from
https://github.com/ADAC-UoN/PIMMS

The test dataset comprises two short subsamples of real data generated from *S. uberis*. These have arbitrarily been labelled as IN and OUT as input and output pool of a phenotypic screen. Each consists of paired reads (R1 & R2).

Commands used for this tutorial are provided in the text file example.command.line.README.txt.

We assume that an aligner (we suggest BWA) is available and in your path. You should also install
1) Fastxtookit available from http://hannonlab.cshl.edu/fastx_toolkit/
2) To generate plots from fastxtoolkit gnuplot is also required.
3) The statistical package R
4) Perl packages Getopt::Long, and Statistics::Descriptive

Following the complete tutorial the results should be located as shown in figure 1.

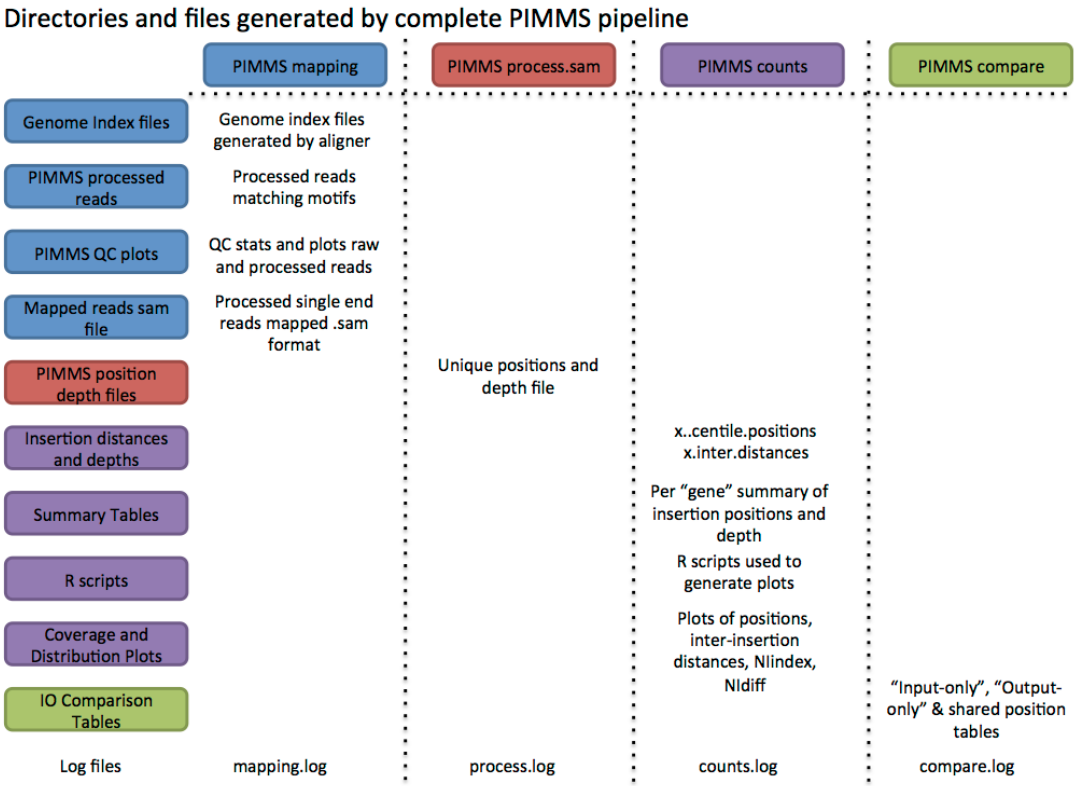


Figure 1.
Directories and files contained which are generated by each module of PIMMS. If the tutorial is followed this should be the resulting data structure.

11.1 Step 1 Mapping reads to a reference genome.

```
./PIMMS.pl -m mapping -c PIMMS.commands.txt -i test.IN.R1.fastq -j test.IN.R2.fastq -g
0140J.dna -e sanger –q N -r @M01661 -min 20 -max 50 -n testIN
```

In the example dataset test.IN.Rn.fastq and test.OUT.Rn.fastq are generated on a MIseq run with the identifier @M01661. For other data this can be determined by looking at the head of the fastq file in a text editor or terminal. See http://en.wikipedia.org/wiki/FASTQ_format for encoding type. The reference genome is in standard fasta format.

The PIMMS.commands.txt file looks like this:
TCAGAAAACTTTGCAACAGAACC
GGTTCTGTTGCAAAGTTTAAAAA
bwa mem -t 4

Lines one and two are the two motifs to search for (the last 23 bases of both ends of the transposon, these will be different if a different transposon is used). Line three is the command for the aligner (BWA mem with 4 threads, if the thread count is changed this will naturally affect the mapping speed).

The produced log file tells us lots of information about the run.
For example number of reads matched.

```
PIMMS.matched.min20.max50.fastq.summary
File        Total Reads         Sequences matched   Percent matched (Seqeuences matched/Total Reads)
./test.IN.R1.fastq      100000    50952      50.95
=============================================================================
Motif_matched_count
Motif 1(TCAGAAAACTTTGCAACAGAACC) or reverse complement 21011 (41.24% of 50952)
Motif 2(GGTTCTGTTGCAAAGTTTAAAAA) or reverse complement 29033 (56.98% of 50952)
Motif 1 AND Motif 2 in single read 1287 (2.53% of 50952)
```

How many of these are single and paired end

```
Count matched pairs
Single End Reads:94332
Paired End Reads:2246
Combined single end output File:testIN.PIMMS.processed.se.fastq
```

And the location of files and plots

```
=============================================================================
Quality control plots of Processed Data see directory PIMMS.QC.plots
Quailty of reads per base position:PIMMS.processed.se.fastq.Quality.png
Nucleotide Distribution:PIMMS.processed.se.fastq.Nucl.dist.png
=============================================================================
```

If an experiment with input and output pools are to be compared this is repeated for the output pool

```
./PIMMS.pl -m mapping -c PIMMS.commands.txt -i test.OUT.R1.fastq -j test.OUT.R2.fastq
-g 0140J.dna -e sanger –q N -r @M01661 -min 20 -max 50 -n testOUT
```

11.2 Step 2 Processing the SAM file.

```
./PIMMS.pl -m process.sam -s
Mapped.reads.sam.file/testIN.PIMMS.processed.se.fastq.mapped.sam -n testIN -col 8 -mis
0 -a 50
```

```
./PIMMS.pl -m process.sam -s
Mapped.reads.sam.file/testOUT.PIMMS.processed.se.fastq.mapped.sam -n testOUT -col 8 -
mis 0 -a 50
```

The SAM file produced in step 1 is processed. This is found in the directory "Mapped.reads.sam.file". As the transposon used in these experiments inserts by inclusion of a 8 bp repeat, the positions are collapsed if they are exactly 8bp apart. The –mis parameter controls for the number of mismatches in an alignment using the "MD" tag of the SAM file and alignment score controlled by the –a option by filtering on the "AS" tag of the SAM file. Reads with mismatches greater than that chosen by the user or alignment scores less than requested are ignored. For those that exceed user cutoffs the position of the initial base of the alignment is recorded.
In this example a fixed integers for mismatches is used here. However if a decimal is used this is interpreted as the maximum number of mismatches as a proportion of the read length i.e –mis 0.5 is interpreted as 50% of each read can be a mismatch.

The processing log file again gives information on the number of reads mapped (+/-) collapsing of positions.

```
================================================================================================
Sam File processed: testIN.PIMMS.processed.se.fastq.mapped.sam
Total Reads in testIN.PIMMS.processed.se.fastq.mapped.sam: 42392
Total Reads mapped to reference genome: 31983
Percent of reads mapped to reference genome: 75.45%
Unique mapped positions: 6876
Mean read depth per mapped postion: 4.65
================================================================================================
Collapsed Reads at 8 (N = collapse option off)
Total reads mapped after positions collapsing: 31983
Reads collapsed: 378
Unique mapped positions after collapsing: 6825
Mean depth at position after collapsing: 4.69
Percent of total positions collapsed: 0.74%
Percent of total reads collapsed: 1.18%
================================================================================================
```

11.3 Step 3. Count insertions and determine positions of essential genes.

```
./PIMMS.pl -m counts -d
PIMMS.position.depth.files/testIN.m0.a50.PIMMS.processing.collapse.8.positions.depths
-g 0140J.dna -gtf S.uberis.gtf -cov 2 -n testIN

./PIMMS.pl -m counts -d
PIMMS.position.depth.files/testOUT.m0.a50.PIMMS.processing.collapse.8.positions.depths
-g 0140J.dna -gtf S.uberis.gtf -cov 2 -n testOUT
```

The counts module uses the depths file produced by step 2 filters by minimum coverage (-cov) at a particular site and maps to known gene positions as determined by the GTF file. In this example positions with < 2 reads mapped at that position i.e singleton insertion events are ignored. The requirments for the GTF file are described in section 9.1 above.

The log file provides information on the calculations used in the analysis and also locations of files.

```
Position depth file processed:testIN.m0.a50.PIMMS.processing.collapse.8.positions.depths
Genome size (bp):1852352
Expected insert count: 6.26979472140762 (29932 / 4774)
Expected insert proportion: 0.000209467951403435 = (1 / 4774)
================================================================================================
Coverage Descriptive statistics for positions >= 2 depth:
```

Coverage count: 4774
Mean coverage: 6.26979472140762
Median coverage: 4
Minimum coverage: 2
Maximum coverage: 1084
==========

Details of the base composition at the point of insertion are given and the total composition of the genome are also determined.

==========
Base composition of unique insertion sites
Insertion base A 1572 32.9%
Insertion base C 950 19.9%
Insertion base G 1116 23.4%
Insertion base T 1136 23.8%
Base composition of Genome
Genome base A 586754 31.7%
Genome base C 338988 18.3%
Genome base G 339562 18.3%
Genome base T 587048 31.7%

The distribution of distances between unique insertions are reported

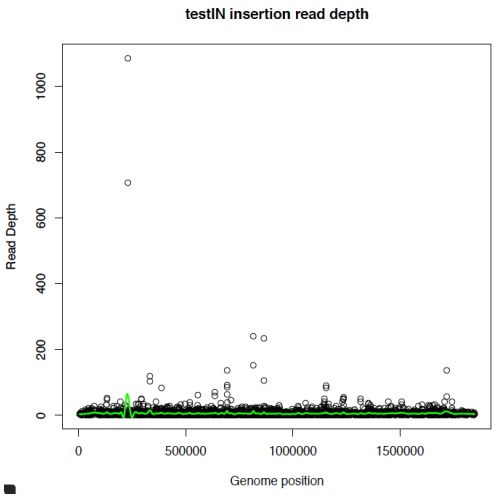
==========
Distribution of distances between unique insertion positons above 2 depth (bp):
Count:4773
Mean inter unique insertion distance: 386.675256652001
Median inter unique insertion distance: 90
Minimum inter unique insertion distance: 1
Maximum inter unique insertion distance: 16414
==========

OUTPUT TABLES
Table of all CDS, tRNA and rRNA with insertion number etc in
Summary.Tables/testIN.PIMMS.counts.min_cov.2.summary.table
Normalised Reads Mapped (NRM_score) is calculated as (total number of reads mapped/length of gene in KB)/(total mapped read count/10^6)
Normalised Insertions Mapped (NIM_score) is calculated as (total number of unique insertions mapped/length of gene in KB)/(total mapped read count/10^6)
Table of all per position insertions NIindex etc in
Insertion.distances.and.depths/testIN.PIMMS.counts.min_cov.2.insertion.positions.depths
==========

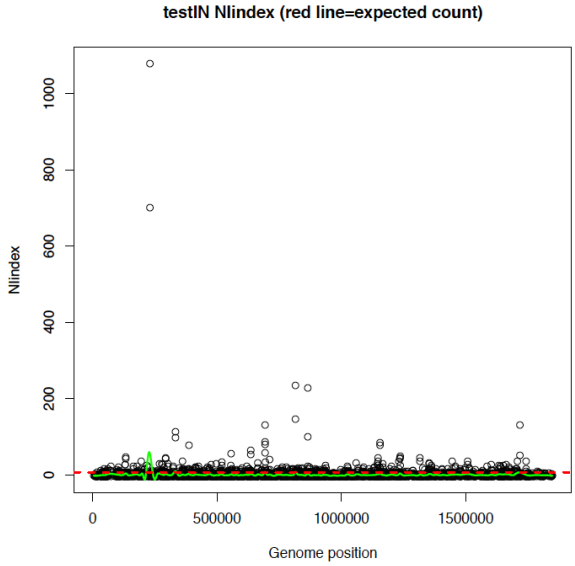
The summary table is a tab delimited file providing all information on a gene-by-gene basis.

Plots generated are located in the Coverage.Distribution.plots directory.

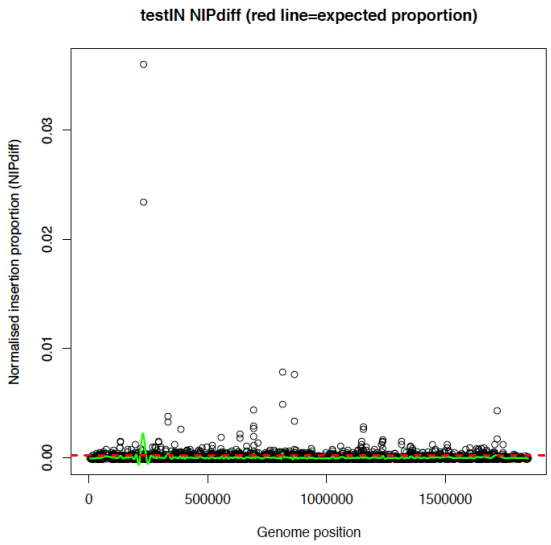
PIMMS.counts.min_cov.2.insertion.positions.depths.pdf – depth at each position across genome. Green line = smoothed depth.



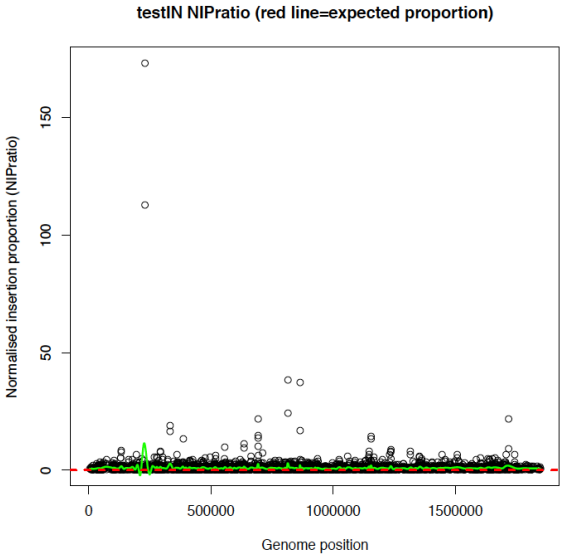
PIMMS.counts.min_cov.2.insertion.positions.NIindex.pdf- as above but includes dotted line of expected numbers of insertions (red dotted line).
PIMMS.counts.min_cov.2.insertion.positions.NIindex.Zoom.pdf (not shown) is the same plot but zoomed in to show differences close to NIindex = 0.



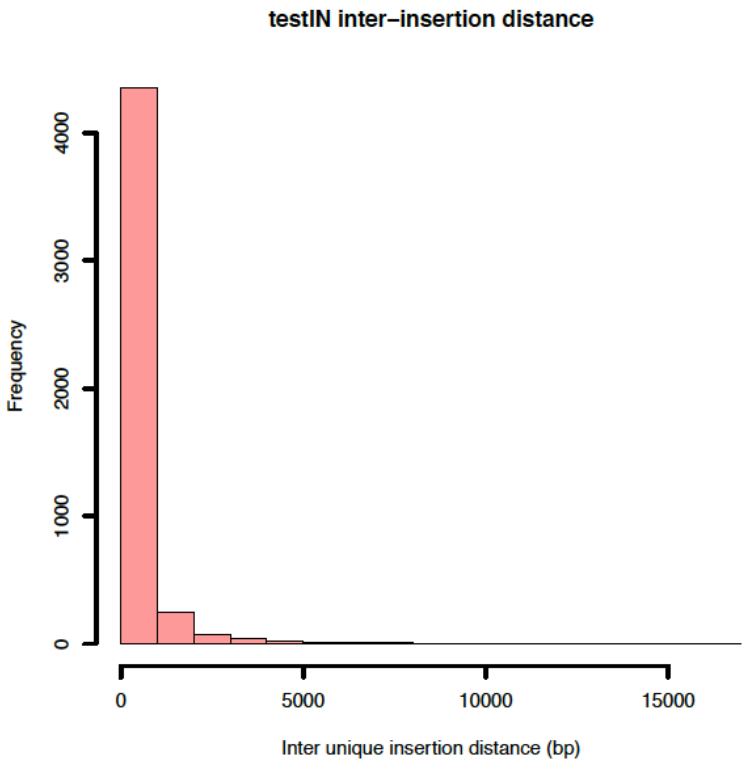
PIMMS.counts.min_cov.2.insertion.positions.NIPdiff.pdf Normalised insertion proportion (NIdiff), green = smoothed NIPdiff, red = expected.
PIMMS.counts.min_cov.2.insertion.positions.NIPdiff.zoom.pdf (not shown) Zoomed version of above.



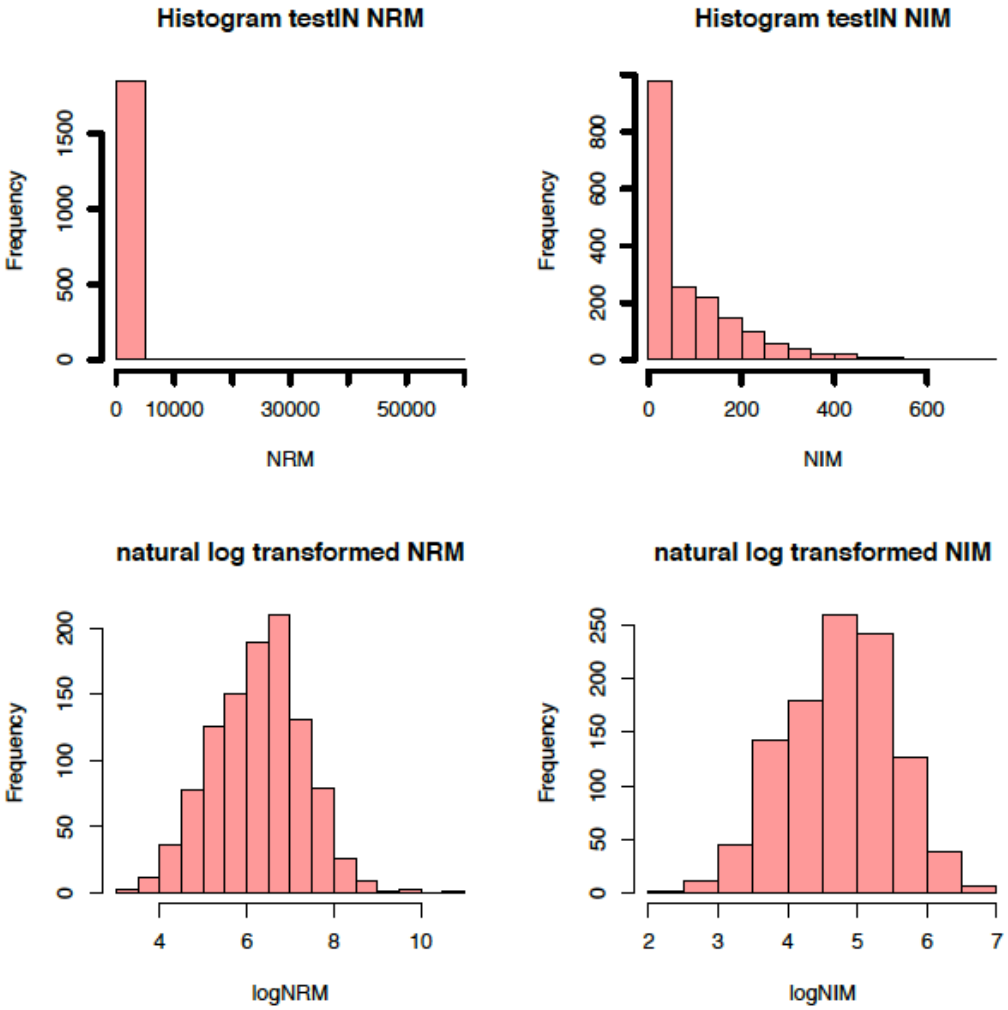
PIMMS.counts.min_cov.2.insertion.positions.NIPratio.pdf. Normalised insertion proportion NIPratio. green = smoothed NIPdiff, red = expected.
PIMMS.counts.min_cov.2.insertion.positions.NIPdiff.zoom.pdf (not shown) Zoomed version of above.



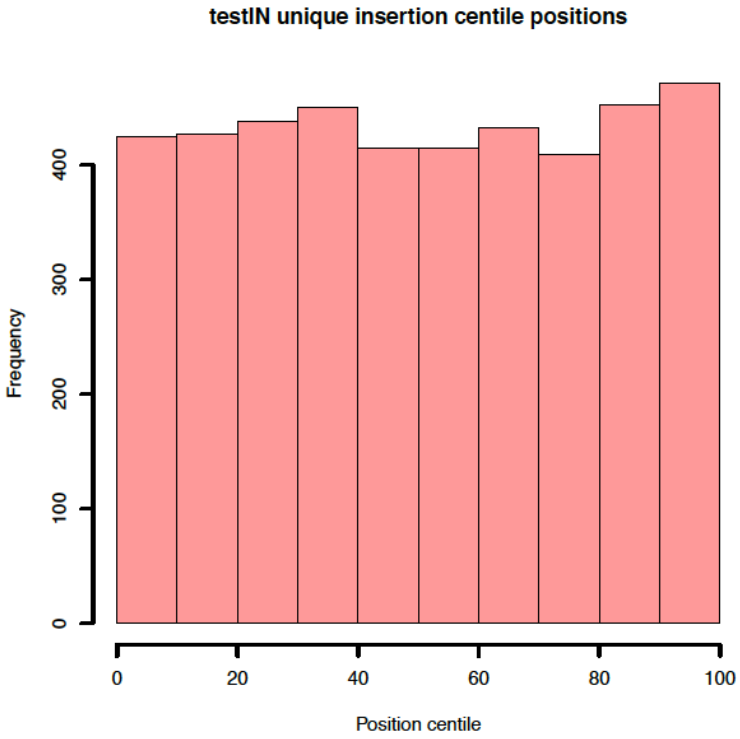
PIMMS.counts.min_cov.2.inter.insertion.distances.histogram.pdf. Histogram of the distance between unique insertion points.



PIMMS.counts.min_cov.2.summary.plots.pdf. Summary plots of NRM NIM and these values log transformed.



PIMMS.counts.min_cov.2.unique.insertion.centile.positions.density.plot.pdf. Density plot of the centile positions of insertions into genes. For each insertion the normalised position in a gene is calculated as the centile position. A histogram of these positions are plotted.



11.4 Step 4. Compare phenotypes or experiments

```
./PIMMS.pl -m compare -in
Insertion.distances.and.depths/testIN.PIMMS.counts.min_cov.2.insertion.positions.depth
s -out
Insertion.distances.and.depths/testOUT.PIMMS.counts.min_cov.2.insertion.positions.dept
hs -dist 8 -n testIO
```

Two experiments or a phenotypic challenge can be conducted using PIMMS compare. Two insertion depth files following PIMMS counts processing are compared.

The log file reports important information

```
======================================================================
Input file:
        Insertion.distances.and.depths/testIN.PIMMS.counts.min_cov.2.insertion.positio
ns.depths
Output file:
        Insertion.distances.and.depths/testOUT.PIMMS.counts.min_cov.2.insertion.positi
ons.depths
Total positions:5924
Shared positions:3690 (62.29%)
Of 3690 those included after collapse at 8 bp:11
Input only positions:1084 (18.30%)
Output only positions:1150 (19.41%)
======================================================================
Of shared positions:
Mean of log transformed proportion ratio:-0.00343202774319904
Standard Deviation of log transformed proportion ratio:0.53114591770895
Number of positions with Zscore >= 3.291 (p < 0.001): 3
Number of positions with Zscore >= 2.579 (p < 0.01): 21
Number of positions with Zscore >= 1.960 (p < 0.05): 66
Number of positions with Zscore <= -3.291 (p < 0.001): 3
Number of positions with Zscore <= -2.579 (p < 0.01): 30
Number of positions with Zscore <= -1.960 (p < 0.05): 72
======================================================================
```

Additionally the three tables in directory “IO comparison tables” contains tab-delimited files of sites shared or unique to each experiment. The compare script allows processing of data obtained from phenotypic studies. Following use of the counts module, PIMMS compare, compares two pools termed “input” and “output” to identify common and unique mutation events between experimental conditions. Three output tables are produced (input only, output only and shared positions). Tables consist of insertion position normalised observed and expected number of reads and associated gene information. For the shared positions an indication of the magnitude of deviation from an expected norm is determined. Within an experiment the natural logarithm (base *e*) transformed proportion ratio (the share of reads mapped at a location) 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). To provide an approximation of statistical importance, Zscores > standard deviations equivalent to a p value of 0.001, 0.01, 0.05 are flagged as |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) and if |Zscore| < 1.960, Flag = "".