TECH NOTE

Analyzing Twist Targeted Methylation Sequencing Data Using the Twist Human Methylome Panel

DISCLAIMER

This document is meant to serve as a guideline for processing next-generation sequencing (NGS) data from the Twist NGS Methylation Detection workflow using the Twist Human Methylome panel. These are recommendations based on the Twist bioinformatics workflow for FASTQ files generated using Illumina platforms. Specifically, this technical note highlights how to align reads, perform CpG calling, and generate QC metrics. This pipeline can be modified to run any Twist Custom Methylation panel with the appropriate file substitutions, such as the reference file and the target/bait interval files. For readers with limited bioinformatics experience, we suggest starting with pre-built pipelines such as the nf-core/methylseq package referenced in the original NEBNext Enzymatic Methyl-seq (EM-seq™) paper (https://nf-co.re/methylseq/1.6.1).

OVERVIEW

Methylation analysis pipelines for NGS data are based on several open source tools and packages. This collateral describes best practices for using the BWA-meth/MethylDackel workflow to generate high quality CpG calls from the Twist Human Methylome panel. While we acknowledge that there are other aligners and methylation callers in the field (such as Bismark), the focus of this technical note is meant to serve as a reference point to generate data that is directly comparable to the Twist gold standard dataset. Additionally, internal data shows that the BWA-meth/MethylDackel workflow is the most robust in aligning reads, handling the presence of SNPs and sequencing errors, and producing the highest quality CpG calls. To request access to the gold standard dataset for the Twist Human Methylome panel, please contact customer support (customersupport@twistbioscience.com) and it will be transferred via a secure FTP link.

PACKAGES AND VERSIONS

bwameth	version: 0.2.2
cutadapt	version: 3.2
fastqc	version: 0.11.9
gatk	version: v4.1.8.1
methyldackel	version: 0.5.3
multiqc	version: 1.9
picard	version: 2.22.8
sambamba	version: 0.7.1
samtools	version: 1.10
seqtk	version: 1.3-r106
trim galore!	version: 0.6.4_dev

**Other dependencies of these main packages might be required.*

RESOURCES BUNDLE

1. **hg38_noAltHla_UCSC.fa** = hg38 reference file used by Twist. Any reference file can be used in its place.
2. **covered_targets_Twist_Methylome_hg38_annotated_collapsed.bed** = BAIT and TARGET file for the Twist Human Methylome.

   **hg19 liftover file is also available. Please note: There is always an inherent risk and inaccuracy when using liftover files. Older or newer genome builds are not identical in the regions they encompass. Therefore, some regions do not always remain the same, get deleted, or are misplaced altogether.*

3. **Internal Twist Human Methylome dataset** = (please contact Twist customer support for data transfer via secure FTP):

   NovaSeq 2x100 bp, 8-plex methylome experiment:
   - FASTQ files at 250X raw coverage
   - BAM files generated by bwa-meth alignment, hg38
   - Picard HsMetrics
   - CpG calls in bedGraph format generated by MethylDackel
   - Global methylation report

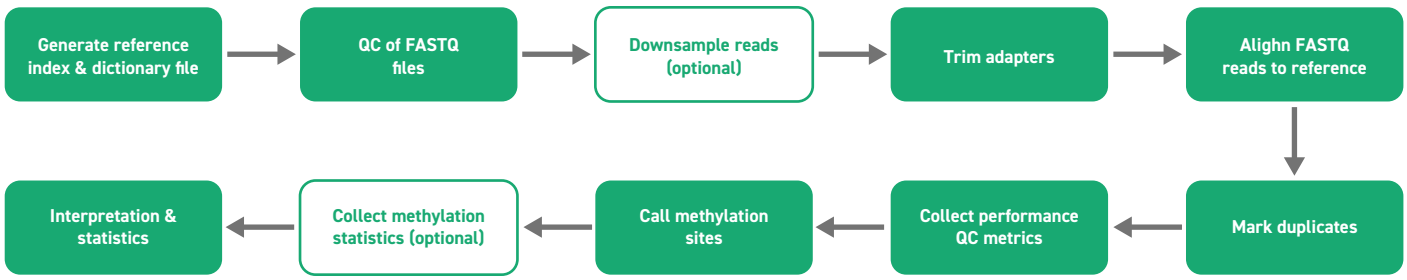
OUTLINE

Described here are the individual steps and commands for each processing step. These should be combined into a singular pipeline for analyzing multiple samples in a high throughput setting. All reference files used are provided by Twist in the Twist Human Methylome panel resources bundle.

- **Generate reference genome index and dictionary file** (bwa, samtools, Picard)
- **QC of FASTQ files** (fastqc)
- **Trim adapters** (trim galore!)
- **Align FASTQ reads to reference** (bwameth, samtools, sambamba)
- **Marking duplicates and collect QC metrics** (gatk, picard, samtools)
- **Calling methylation** (MethylDackel)
- **Interpretation and statistics** (R)

Optional steps:

- **Downsampling reads** (seqtk)
- **Collect methylation stats**



DETAILED TWIST RECOMMENDED PIPELINE AND LOGIC

The example workflow below is for an example NovaSeq 2 x1 00 bp experiment where the sample name is "**S1_methylome**" and the genome reference is "**hg38_noAltHla_UCSC.fa**". When copy and pasting from a document to the command line or shell script, certain characters can be erroneous, such as the " symbols. Please check all syntax.

1. Generate reference genome index and dictionary file (bwa, samtools, Picard)

1a. Make an index of the reference genome in FASTA format. The output will be five files: hg38_noAltHla_UCSC.fa.amb, hg38_noAltHla_UCSC.fa.ann, hg38_noAltHla_UCSCfa.bwt, hg38_noAltHla_UCSCfa.pac, hg38_noAltHla_UCSCfa.sa

```
bwa index -a bwtsw hg38_noAltHla_UCSC.fa
```

1b. Create an index for your FASTA.file.

```
samtools faidx hg38_noAltHla_UCSC.fa > hg38_noAltHla_UCSC.fa.fai
```

1c. Create a reference sequence dictionary which is required by many processing and analysis tools.

```
java -Xmx4g -Xms4g -jar picard.jar CreateSequenceDictionary REFERENCE=hg38_noAltHla_
UCSC.fa OUTPUT=hg38_noAltHla_UCSC.dict
```

2. QC of fastq files (fastqc)

FastQC can be performed on pre-trimmed or post-trimmed reads from step 4.

```
fastqc --noextract --threads 16 S1_methylome_R1_001.fq.gz S1_methylome_R2_001.fq.gz
```

3. Optional: Downsampling reads (seqtk)

When comparing datasets from different experiments or samples within the same experiment, it is useful to randomly downsample fastq files to the same number of reads. For the Twist Human Methylome Panel dataset, all samples should be downsampled to 250X raw coverage based on the merged probe file.

Total targeted region size (based on the merged probe footprint size): **123,144,327 bp**

```
awk '{SUM += $3-$2} END {print SUM}'
covered_regions_Twist_Human_Methylome_hg38.bed
```

Total wanted basepairs at 250X raw coverage: **30,786,081,750 bp**

```
total targeted region size * downsample amount at X raw coverage =
total wanted basepairs
```

Total reads wanted: 153,930,409 **final total reads**

This depends on the sequence read length, so an experiment with 2 x 100 bp would be a different downsampling amount than an experiment with 2x150 bp. Because paired-end read data is used in this example, it is divide by 2 reads because there is a read 1 (R1) and a read 2 (R2).

```
total wanted basepairs/(read length * number of reads) = final total reads
```

Unzip FASTQ files first!

```
gunzip -c  S1_methylome_R1_001.fastq.gz >  S1_methylome_R1_001.fastq
gunzip -c  S1_methylome_R2_001.fastq.gz >  S1_methylome_R2_001.fastq
```

Use the seqtk package with your downsampling value (final total reads) to downsample the samples. Set your seed when running paired-end reads.

```
seqtk sample -2 -s 42 S1_methylome_R1_001.fastq 153930409 >
S1_methylome_R1_001_downsampled.fastq
seqtk sample -2 -s 42 S1_methylome_R2_001.fastq 153930409 >
S1_methylome_R2_001_downsampled.fastq
```

4. Trim adapters (trim galore!)

For adapter trimming, Trim Galore! uses the first 13 bp of Illumina standard adapters ('AGATCGGAAGAGC') by default (suitable for both ends of paired-end libraries), but additional accepts other adapter sequences.

```
trim_galore --gzip --cores 16 --output_dir output --2colour 20 --
paired S1_methylome_R1_001_downsampled.fastq
S1_methylome_R2_001_downsampled.fastq

Output = S1_methylome_R1_001.trimmed.fq.gz
Output= S1_methylome_R2_001.trimmed.fq.gz
```

5. Align FASTQ reads to reference (bwameth, samtools, sambamba)

```
bwameth.py --reference hg38_noAltHla_UCSC.fa -t 16 --read-group
'@RG\tSAMPLE_ID:1\tPL:illumina\tLB:SAMPLE\tSM:SAMPLE'
S1_methylome_R1_001.trimmed.fq.gz
S1_methylome_R2_001.trimmed.fq.gz

sambamba view -h -t 16 -T hg38_noAltHla_UCSC.fa --filter 'not
secondary_alignment and not failed_quality_control and not
supplementary and proper_pair and mapping_quality > 0' -f bam -l 0
S1_methylome.bam -o /tmp/tmpzs73858y.bam

sambamba sort -t 16 -m 30GiB --tmpdir /tmp/ -o /dev/stdout -l 0
/tmp/tmpzs73858y.bam | sambamba view -h -t 16 -o
S1_methylome_sorted.bam -T hg38_noAltHla_UCSC.fa -f bam /dev/stdin

samtools index -@ 16 S1_methylome_sorted.bam >
S1_methylome_sorted.bam.bai
```

6. Mark duplicates and collect QC metrics (picard, samtools): NovaSeq (patterned flow cell) example

6a. Mark duplicates (We run different parameters for a random flow cell or a patterned flow cell. The optical pixel distance should be changed accordingly, random = 100 and patterned = 2500 based on GATK best practices.)

```
java -Xmx4g -Xms4g -jar picard.jar MarkDuplicates -I
S1_methylome_sorted.bam -O S1_methylome_sorted.markdup.bam -R
hg38_noAltHla_UCSC.fa -M
S1_methylome_sorted.picard_markdup_raw_metrics --CREATE_INDEX false --
MAX_RECORDS_IN_RAM 1000 --SORTING_COLLECTION_SIZE_RATIO 0.15 --
ASSUME_SORT_ORDER coordinate --OPTICAL_DUPLICATE_PIXEL_DISTANCE 2500

samtools index -@ 16
S1_methylome_sorted.markdup.bam
```

6b. Convert BED file to interval list

```
java -Xmx4g -Xms4g -jar picard.jar BedToIntervalList -I
covered_targets_Twist_Methylome_hg38_annotated_collapsed.bed
covered_targets_Twist_Methylome_hg38_annotated_collapsed.intervals
SD hg38_noAltHla_UCSC.dict
```

6c. Generate performance metrics (coverage cap setting is increased to 1000 for accurate metrics at higher sequencing depths).

Picard HsMetrics calculate important performance metrics , such as Fold-80 Base Penalty, HS Library Size, Percent Duplicates, and Percent Off Bait.

```
java -Xmx4g -Xms4g -jar picard.jar CollectHsMetrics -I
S1_methylome_sorted.markdup.bam -O
S1_methylome_sorted.markdup.picard_collecthsmetrics_raw_metrics -R
hg38_noAltHla_UCSC.fa --BAIT_INTERVALS
covered_targets_Twist_Methylome_hg38_annotated_collapsed.intervals --
TARGET_INTERVALS covered_targets_Twist_Methylome_hg38_annotated_collapsed.intervals
--MINIMUM_MAPPING_QUALITY 20 --
COVERAGE_CAP 1000 --PER_TARGET_COVERAGE
S1_methylome_sorted.markup.picard_collecthsmetrics_per_target_coverage
_raw.text --NEAR_DISTANCE 500
```

6d. Generate additional performance metrics

```
java -Xmx4g -Xms4g -jar picard.jar CollectMultipleMetrics -I
S1_methylome_sorted.markdup.bam
-O S1_methylome_sorted.markup.picard_collectmultiplemetrics_raw -R
hg38_noAltHla_UCSC.fa --PROGRAM null --PROGRAM CollectGcBiasMetrics --
PROGRAM CollectInsertSizeMetrics --PROGRAM
CollectAlignmentSummaryMetrics
```

7. Calling methylation (MethylDackel)

All quotes come directly from the MethylDackel readme documentation:
github.com/dpryan79/MethylDackel/blob/master/README.md

7a. Check for bias and clean reads before calling CpGs

"In an ideal experiment, we expect that the probability of observing a methylated C is constant across the length of any given read. In practice, however, there are often increases/decreases in observed methylation rate at the ends of reads and/or more global changes. These are termed methylation bias and including such regions in the extracted methylation metrics will result in noisier and less accurate data. For this reason, users are strongly encouraged to make a methylation bias plot. MethylDackel comes with a function for just this purpose:"

```
MethylDackel mbias hg38_noAltHla_UCSC.fa
S1_methylome_sorted.markdup.bam S1_methylome

# to get --OT and --OB (trimming of biased reads, for the methylome. Internal dataset
= --OT 0,0,0,98 --OB 0,0,3,0)
```

Data hasn't suggested that this is more than a few bases when using the Twist Human Methylome panel. The output of this gets passed into the "extract" command to avoid calling CpGs in these biased regions.

7b. Add filters and generate CpG calls

There are additional parameter recommendations that can be added to the extract command to ensure high quality methylation calls.

Excluding likely variant sites (`--maxVariantFrac 0.25`): "If your samples are not genetically homogenous, it can sometimes be advantageous to exclude likely variant sites from methylation extraction. As an example, since unmethylated Cs are read as Ts, extracting methylation from a position with a C->T mutation will cause incorrect results. In such a case, the opposite strand will have an A rather than a G (in the non-variant case, there would be a G regardless of methylation status). MethylDackel tracks the number of non-Gs on the strand opposite of Cs in the reference sequence. If the fraction of these exceeds the --maxVariantFrac option, then that position will be excluded from output."

Require a minimum depth (`--minDepth 10`). This parameter is pretty straightforward. For the majority of applications, 10x minimum depth seems to be the consensus. Internal Twist datasets use a min depth of 10X, but other values can be used.

The merging context option is used here because this produces CpG call percents that are most usable. "In many circumstances, it's desirable for metrics from individual Cytosines in a CpG to be merged, producing per-CpG metrics rather than per-Cytosine metrics. This can be accomplished with the --mergeContext option."

```
MethylDackel extract --minDepth 10 --maxVariantFrac 0.25 --OT X,X,X,X
--OB X,X,X,X --mergeContext hg38_noAltHla_UCSC.fa
S1_methylome_sorted.markdup.bam -o S1_methylome
```

```
Optional flag: --keepDupes
```

MethylDackel readme suggests that users keep duplicates for enrichment-based library preparations. This recommendation is based on old enrichment technology, RRBS, that uses restriction enzymes to fragment and size-select sequenceable fragments. That method produces very low complexity libraries and, therefore, it is likely that fragments have the same start and stop position instead of being true duplicates. This is in contrast with the EM-seq library generation technology based technology that Twist uses. Our method produces highly complex libraries, leaving little likelihood of fragments having the same start and stop position, unless they are true duplicates. Therefore, Twist recommends not keeping duplicates for your methylation calls, but ultimately this is up to the user.

7c. Optional: Create more significant figures for percent methylation

The fourth column of the bedgraph generated by MethylDackel is the percent methylation calls, for example 66 = 66%. By default , MethylDackel rounds this number down to the nearest integer. For some applications, it might be beneficial to generate percentages with more significant figures. In this example, the fourth column is transformed to have 2 additional significant figures after the decimal.

```
awk 'BEGIN {FS=OFS="\t"} NR == 1 {print $0}   NR > 1 {print
$1,$2,$3,(($5/($5+$6)*100)+0),$5,$6;}' OFMT="%.2f"
S1_methylome.bedgraph > S1_methylome_processed.bedgraph
```

Original line: chr1	10541	10543	77	14	4
Transformed line: chr1	10541	10543	77.78	14	4

7d. Add filters and generate cytosine report

This extract command is ran twice. The second command is performed to look at the cytosine report, including -CHH and -CHG sites. The outputs are used to generate global methylation stats, like the non-CpG conversion rate.

```
MethylDackel extract --minDepth 10 --maxVariantFrac 0.25 --OT 0,0,0,98
--OB 0,0,3,0 --cytosine_report --CHH --CHG hg38_noAltHla_UCSC.fa
S1_methylome_sorted.markdup.bam -o S1_methylome_report
```

If methylKit is used for downstream analysis, please add the flag '--methylkit' to generate an output file that is compatible. This flag is not compatible with '--mergeContext'.

8. Optional: Collect methylation statistics (samtools, MethylDackel)

8a. Mapping efficiency

```
samtools stats S1_methylome.bam | grep ^SN | cut -f 2-
```
The output will be = # raw total sequences

```
samtools stats S1_methylome_sorted.bam | grep ^SN | cut -f 2-
```
The output will be = # raw total sequences

Mapped reads (filtered bam) / total reads (original bam)

```
S1_methylome_sorted.bam # raw total sequences/ S1_methylome.bam # raw
total sequences = mapping efficiency
```

8b. Extracting called CpG ratio (percent)

Calculate the global average CpG percent methylation for each sample. This can serve as a quick check there are hypo- or hyper-methylated samples in the experiment. No sample will ever be 0% or 100% methylated, so always expect some degree of variability.

From the MethylDackel cytosine report

```
S1_methylome_report.txt
 x = Number of called methylated CpG sites
 y = Number of called unmethylated CpG sites

 called_cpg = (x / (x + y)) * 100
```

8c. Extracting non-CpG conversion ratio (percent)

The final report gives values for the non-CpG conversion rate, the percent of Cs that were converted to Ts, excluding any C followed by a G. Conversion rate of each base (non-CpG) can be calculated as T / (T + C) * 100, where T is thymine and C is cytosine read counts for that base.

From the MethylDackel cytosine report:

```
S1_methylome_report.txt
 a = Number of called unmethylated non-CpG sites
 b = Number of called methylated non-CpG sites

 non_cpg_conversion = (a / (a + b)) * 100
```

9. Interpretation of methylation calls - testing for differential methylated cytosines (DMCs) or regions (DMRs)

There are multiple packages and methods for interpreting methylation calls from the final bedgraph file (MethylDackel pipeline). The main goal of this analysis is to look for regions that have statistically different methylation statuses between samples or groups of samples which could point to functional, diagnostic, or prognostic methylation sites or regions.

1. **methylSig:** “Uses a beta-binomial model to test for significant differences between groups of samples. Several options exist for either site-specific or sliding window tests, combining strands, and for variance estimation.” github.com/sartorlab/methylSig
2. **deepTools:** For this tool, convert the bedgraph files into bigwig files (bedGraphToBigWig). Then, use the compute matrix function to calculate scores per genome region(s) and prepare intermediate files that can be used with plotHeatmap and plotProfiles. For example, plotProfile plots the methylation signal profile over regions of interest. deeptools.readthedocs.io/en/develop/index.html
3. **DSS (Dispersion Shrinkage for Sequencing data)**: The core of DSS is a procedure based on a Bayesian hierarchical model to estimate and shrink gene- or CpG site-specific dispersions, then conduct Wald tests for detecting differential expression/methylation. bioconductor.org/packages/release/bioc/vignettes/DSS/inst/doc/DSS.html#1_Introduction

SUPPLEMENTARY INFORMATION

Correlation analysis: Use when comparing a processed dataset to the Twist gold standard dataset to validate pipeline or workflow. (ucsc-bedgraphtobigwig, deepTools, R).

1. Create BigWig files from your final MethylDackel CpG call bedGraph files. Repeat steps 1a & 1b for all samples.

1a. Only print the first 4 columns, sort by location, and remove the track description line:

```
awk '{print $1,$2,$3,$4}' S1_methylome.bedGraph | sort -k1,1 -
k2,2n | grep -v “track” > S1_methylome_processed.bedGraph
```

1b. Next, convert your processed bedGraph file into a BigWig file:

```
bedGraphToBigWig S1_methylome_processed.bedGraph
chrom_hg38.sizes.genome S1_methylome.bw
```

2. Use deepTools multiBigwigSummary to compute the average scores over the genomic regions for each sample.

include as many samples that you want, example has all 8-plex methylome samples

```
multiBigwigSummary bins --binSize 1000 -b Methylome_01_8-
plex_Point010_OvernightFH_sorted.markdup_OT_OB_merge_CpG.bw
Methylome_02_8-plex_Point010_OvernightFH_sorted.markdup_OT_OB_merge_CpG.bw
Methylome_03_8-plex_Point010_OvernightFH_sorted.markdup_OT_OB_merge_CpG.bw
Methylome_04_8-plex_Point010_OvernightFH_sorted.markdup_OT_OB_merge_CpG.bw
Methylome_05_8-plex_Point010_OvernightFH_sorted.markdup_OT_OB_merge_CpG.bw
Methylome_06_8-plex_Point010_OvernightFH_sorted.markdup_OT_OB_merge_CpG.bw
Methylome_07_8-plex_Point010_OvernightFH_sorted.markdup_OT_OB_merge_CpG.bw
Methylome_08_8-plex_Point010_OvernightFH_sorted.markdup_OT_OB_merge_CpG.bw
--labels Sample_01 Sample_02 Sample_03 Sample_04 Sample_05 Sample_06
Sample_07 Sample_08 -o Methylome_results_bin.npz
```

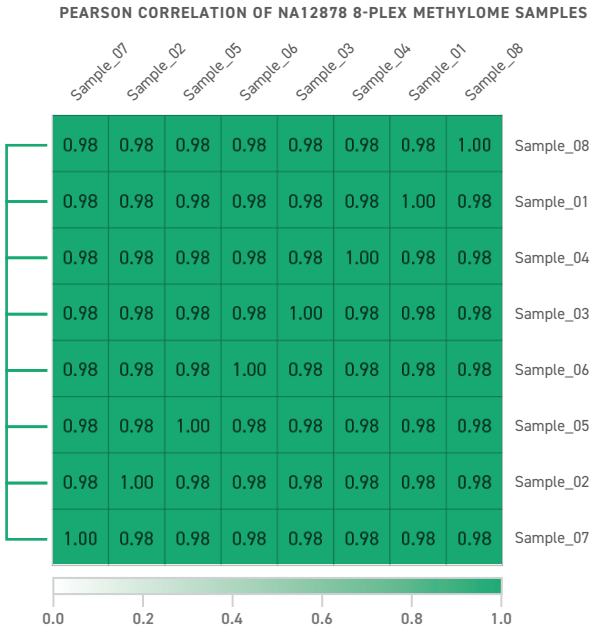
```
**ignore error: The following chromosome names did not match between the bigwig files
```

3. Correlation calculation of all samples - Pearson (deepTools).

```
plotCorrelation --corData Methylome_results_bin.npz --corMethod
pearson  --plotTitle 'Pearson Correlation of NA12878 8-plex
Methylome Samples' --whatToPlot heatmap --plotNumbers --colorMap
Greens -o heatmap_pearsonCorr_Methylome_8plex.png --
outFileCorMatrix pearsonCorr_Methylome_8plex_bin1000_heatmap.txt

plotCorrelation --corData Methylome_results_bin.npz --corMethod
pearson  --plotTitle 'Pearson Correlation of NA12878 8-plex
Methylome Samples' --whatToPlot scatterplot --plotNumbers -o
scatterplot_pearsonCorr_Methylome_8plex.png --outFileCorMatrix
pearsonCorr_Methylome_8plex_bin1000_scatter.txt
```

The CpG calls between all eight NA12878 replicate samples in this example experiment using the Twist Human Methylome panel are highly correlated (pairwise correlation coefficients > 0.98). This shows that the capture of the CpG sites is consistent between all samples and that the pipeline is calling CpG levels the same across all samples for the same regions.



To compare the Twist gold standard CpG calls of S1-S8 to real experimental data, run the code found above to calculate the correlation coefficient. All 16 samples should be ran through multiBigwigSummary and the output file goes into plotCorrelation. There should be a very high correlation coefficients for all pairwise comparisons.

4. Correlation calculation of only two samples (bedops, R - ggpubr)

BedGraph sample data must be formatted to fit a dataframe appropriate for a correlation analysis. Intersect data so that both samples have the same positions to examine.

```
bedtools intersect -a S01_Twist.bed -b S01_Customer.bed -sorted
-wa -wb > S_01_Twist_Customer_intersect_methylome.bed
```

```
awk -v OFS='\t' '{print ($1":"$2"-"$3),$4, $8}'
S_01_Twist_Customer_intersect_methylome.bed
> S_01_Twist_Customer_intersect_methylome_data_frame.txt
*Add header if needed.
```

Example: `S_01_Twist_Customer_intersect_methylome_data_frame.txt`

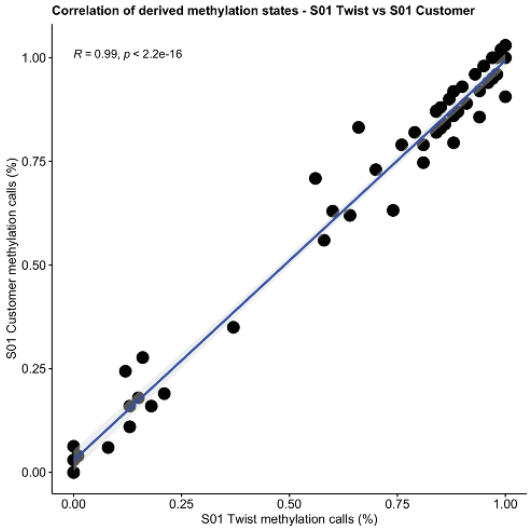
CpG_site	S01_Twist	S01_Customer
chr1:129331-129333	0.92	0.91
chr1:631189-631191	0.83	0.282

4a. Import data and randomly subsample data points. The example below subsamples 100,000 data points:

```
> S_01 <-
read.delim("S_01_Twist_Customer_intersect_methylome_data_frame.txt")
> set.seed(1000)
> index <- sample(1:nrow(S_01), 100000)
> subsample_S_01<- S_01[index, ]
```

4b. Create scatter plot and pearson correlation:

```
> S_01_plot<-ggscatter(subsample_S_01, x = "S01_Twist",
y="S01_Customer", size=0.05, add = "reg.line", conf.int = TRUE,
cor.coef = TRUE, cor.method = "pearson", title ="Correlation of
derived methylation states - S01 Twist vs S01 Customer", xlab = "S01
Twist methylation calls (%)", ylab = "S01 Customer methylation calls
(%)", add.params = list(color = "blue", fill = "lightgray"))
> ggpar(S_01_plot, font.main = c(12,"bold"))
```



Example data and plot show that S01 methylation calls generated by Twist compared to S01 methylation calls generated by the Customer are highly correlated, with a pearson R value of 0.99 and p < 2.2e-16.

How to modify the pipeline for compatibility with any Twist Custom Methylation Panel

While this technical note was written specifically for the Twist Human Methylome panel, these steps can be applied to any Twist Custom Methylation panel.

1. First, change the genome reference file to match the genome build that your panel was designed against. You will substitute that file in place of the **hg38_noAltHla_UCSC.fa** file for all steps where it is required.
2. Second, you will replace the Methylome V1 probe and target BED file (**covered_targets_Twist_Methylome_hg38_annotated_collapsed.bed**) for your custom panel files. These files come in the design folder you are provided with. The TARGET file you should use will start with the naming convention “Target_bases_covered_by_probes” and for the BAIT (probe) file you should use the file starting with the naming convention “Probes_merged_ok”.
3. Third, if you are performing the optional downsampling step, you will need to calculate the total size of your custom panel using the “Probes_merged_ok” file. That is the starting point for all subsequent calculations. Please also adjust the read length to match your sequencing run.