Supplementary Methods

BS-tagging protocol

Step 1: Tagmentation of genomic DNA

This step generates fragmented DNA by tagmentation. The ratio of Tn5:DNA determines fragment size. Fluorometric measurement of input DNA should therefore be performed to quantify input DNA.

1. Set up the following tagmentation reaction in a deep well MIDI plate, and incubate at 37 ^{o}C for 5 minutes in a Hybex oven:

Genomic DNA (100 ng)		19 µl
TD buffer (2x)		25 µl
TDE	1	5 µl
Unmethylated lambda DNA (0.5 ng/µl)		1 µl
		50 µl

Gently pipette up and down 10 times to mix. Change tips between samples.
Alternatively: mix on a thermomixer at 1,500 RPM for 30 seconds. Spin briefly prior to 37 ^{o}C incubation. Change tips between samples.
2. Once the 5 minute incubation is complete, place the MIDI plate on ice and proceed immediately to **Step 2**.

Step 2: Silane bead reaction cleanup

1. To each well of the 96 well MIDI plate containing the tagmented product, add the following:

i.	Dynabeads® MyOne™ Silane	15 µl
ii.	Qiagen Buffer PB	200 µl
		215 µl

2. The total reaction volume will be 265 µl. Seal the plate and mix well by placing the MIDI plate on a thermomixer and shaking at 1,350 RPM for 2 minutes, then incubate the plate at room temperature for 5 minutes.
3. Place the plate on a magnetic stand for 3 minutes or until clear.
4. Carefully remove and discard the supernatant.
5. Remove the plate from the magnet and add 200 µl of 80% ethanol*.
6. Seal the plate and mix well by placing the MIDI plate on a thermomixer and shaking at 1,350 RPM for 30 seconds.
7. Place the MIDI plate back onto the magnet for 2 minutes or until clear.
8. Carefully remove and discard the supernatant when clear and proceed to repeat steps 5-7 for a total of two ethanol washes.
9. After the second wash, place the plate on magnet for 2 minutes or until clear.
10. Remove and discard the supernatant then briefly spin down to collect any residual ethanol at the bottom of the well.
11. Place the plate back on the magnet and use pipette to remove any remaining ethanol.
12. Dry the beads at room temperature for 4-6 minutes.
13. Once dry, remove the MIDI plate from the magnetic stand and elute in 21 µl elution buffer.
14. Seal the plate and mix by shaking at 1,850 RPM for 4 minutes.
15. Briefly spin the plate and place it back onto a magnetic rack for 3-5 minutes or until clear.
16. Carefully transfer 19 µl of the eluted material into a clean PCR plate.

17. Proceed immediately to **Step 3**.

    * Use freshly prepared 80% ethanol for the washing step

Step 3: End -filling with 5-methyl-dCTP mix

1. Set up end filling reaction in a clean PCR plate and incubate at 37 °C for 30 minutes in a thermal cycler:

NEB buffer 2 (10x)	2.5 µl
Klenow fragment (3'->5' exo -)	1.5 µl
Tagmented DNA	19 µl
End-repair nucleotide solution	2 µl
	25 µl

2. Clean up the product with 50 µl (2:1 ratio) of AMpureXP beads according to the manufacturer's instructions. The elution volume is 21 µl in elution buffer.

Step 4: Bisulphite treatment

1. Perform the bisulphite treatment reaction according to the manufacturer's instruction. **CRITICAL STEP** | ensure that the conversion mix is performed for at least 10 minutes while protected from light. There should be no visible solids in the conversion reagent. If solids are present, a fresh conversion mix should be made to avoid conversion issues.
2. The final elution volume is 18 µl

Step 5: Adapter attachment and PCR amplification

1. Set up the following reaction in a PCR plate:

KAPA HiFi HotStartUracil+ Ready Mix (2x)	25 µl
10 µM Illumina i5-adapter (IDT)	1.5 µl
10 µM custom i7-adapter (IDT)	1.5 µl
PCR primer cocktail (PPC, Illumina)	5 µl
Bisulphite-treated DNA	17 µl
Total	50 µl

2. Place the sample plate in a thermal cycler and perform the following steps:

```
98 °C        30 seconds
10 cycles of
      98 °C  10 seconds
      63 °C  30 seconds
      72 °C  3 minutes
72 °C        5 minutes
4 °C storage
```

Step 6: Size selection of library

1. Set up the purification reaction in a deep well, MIDI plate and purify the products according to the manufacturer's instructions:

PCR product	49 µl
H_2O	151 µl
AMpureXP	120 µl
	320 µl

* Use freshly prepared 80% ethanol for the washing step.

CRITICAL STEP | The use of a 0.6X AmpureXP cleanup will exclude fragments <400 bp

2. Elute in 15 µl of elution buffer.

Step 7: Library quality check

Before running the libraries on a massively parallel sequencer, check the length distribution of library using the Bioanalyzer and quantify the product with Qubit Fluorometric Quantitation.

Step 8: Massively parallel sequencing on the Illumina HiSeq X system

Custom sequencing primer oligos should be used for read 2 sequencing and i7 index sequence reading. The Illumina-provided sequencing oligos are used for read 1. If using recent versions of HiSeq X software (RTA 2.7.7 and HCS 3.4.0.38), a spike-in of 5% PhiX DNA should be enough to generate high-quality data, but an even lower proportion of *K. radiotolerans* should generate data of equivalent quality.

Stage 9: Analyzing BS-tagging data

Adapter sequence are first N-masked from raw FASTQ files using cutadapt v1.9.1 (Martin 2011). Because the library protocol produces R1 reads with G>A transitions and R2 reads with C>T transitions after bisulphite conversion, short-read alignment is performed with bwa-meth (Pedersen et al. 2014) with R2 reads mapped as a typical C>T converted R1 read and R1 reads mapped as a typical G>A converted read. Additionally, we modified bwa-meth's default minimum longest match length for a read (0.44*read-length) to greater than 30 bp (0.2*read-length) before marking the alignment in sequence alignment/map (SAM) format (Li and Durbin 2009) with the 0x200 bitwise flag indicating not pass filter, failed platform/vendor quality control. The resulting alignments are marked for duplicates using Picard v2.4.1. To calculate strand-specific duplicates, alignments are separated based on their orientation to the reference, then MarkDuplicates is run on the two alignment sets separately. Our use of methylated cytosines in end-repair following tagmentation revealed a rare artifact (~1-2% of reads) where incorporation of the methylated cytosines extended deeper into the duplex fragment than the typical ~9 bases expected from the transposition footprint. Because this obscures the original methylation state, we designed a custom Perl script, filterFillIn2, which marks reads with 4 consecutive methylated CHH's outside of the first 9 bases and excludes those reads from downstream analysis by appending the 0x200 bitwise flag, marking them as qc failed.

Custom oligonucleotides and primers

Index sequences

BSTAG_i701
CAAGCAGAAGACGGCATACGAGATTCGCCTTAGTTTTGTGGGTTTGGAGATGTGTATAAGAGATAG

BSTAG_i710
CAAGCAGAAGACGGCATACGAGATCAGCCTCGGTTTTGTGGGTTTGGAGATGTGTATAAGAGATAG

BSTAG_i703
CAAGCAGAAGACGGCATACGAGATTTCTGCCTGTTTTGTGGGTTTGGAGATGTGTATAAGAGATAG

BSTAG_i704
CAAGCAGAAGACGGCATACGAGATGCTCAGGAGTTTTGTGGGTTTGGAGATGTGTATAAGAGATAG

BSTAG_i501 (same as Illumina N501)
AATGATACGGCGACCACCGAGATCTACACTAGATCGCTCGTCGGCAGCGTC

BSTAG_i502 (same as Illumina N502)
AATGATACGGCGACCACCGAGATCTACACCTCTCTATTCGTCGGCAGCGTC

BSTAG_i503 (same as Illumina N503)
AATGATACGGCGACCACCGAGATCTACACTATCCTCTTCGTCGGCAGCGTC

BSTAG_i504 (same as Illumina N504)
AATGATACGGCGACCACCGAGATCTACACAGAGTAGATCGTCGGCAGCGTC

Sequencing primers (Exiqon)

Read2_sequencing primer
GTTTTGTGGGTT+TGGA+GATGT+GTATAA+GA+GATAG
+LNA base, estimated TM 72 ^{o}C

Index_i7_reading primer
CTATCTCTTATACA+CAT+CTC+CAAAC+CCACA+AAAC
+LNA base, estimated TM 72 ^{o}C

Code sources

filterFillIn2

https://github.com/will-NYGC/bstag (and below).

CutAdapt

https://cutadapt.readthedocs.io/en/stable/

bwa-meth

https://github.com/brentp/bwa-meth

Picard

http://broadinstitute.github.io/picard/

Trim Galore!

https://www.bioinformatics.babraham.ac.uk/projects/trim_galore/

phantompeakqualtools

https://github.com/kundajelab/phantompeakqualtools

deepTools

https://deeptools.readthedocs.io/en/latest/

Analytical pipeline modules (**Supplemental Fig S2**):

MethylDackel

https://github.com/dpryan79/MethylDackel

MethPipe (Song et al. 2013)

https://github.com/smithlabcode/methpipe

RADmeth (Dolzhenko and Smith 2014)

https://github.com/smithlabcode/radmeth

methylKit (Akalin et al. 2012)

http://bioconductor.org/packages/release/bioc/html/methylKit.html

Biscuit

https://github.com/zwdzwd/biscuit

filterFillIn2 code

```
#!/bin/env
perl
             # Needs samtools in the path
             use strict;
             use warnings;
             use Bio::DB::Bam::Alignment;
             use Getopt::Long;
             use File::Basename;
             use IPC::Cmd qw(can_run);
             use List::Util qw(min max);
             use POSIX;




             my $script_version = 'filterFillIn2.pl v0.2';


             my $cmd = join(" ",$0,@ARGV);


             my($bam,$ref_fa,$prefix,$keep_string,$verbose,$help);
             my $fillIn_dist = 11;
             GetOptions("b=s" => \$bam,
                       "r=s" => \$ref_fa,
                       "p=s" => \$prefix,
                       "k"   => \$keep_string,     # Keep string of mC/C on reads for
             downstream QC
                       "d:i" => \$fillIn_dist,
                       "v"   => \$verbose,
                       "h"   => \$help);


             &usage if $help or !$bam or !$ref_fa;




             # Test for samtools
             (print STDERR "***ERROR: \'samtools\' must be in path, exiting...\n\n" and
             &usage) if !(can_run('samtools'));
```

```perl
# Define output prefix and files
if (!$prefix) {
    $prefix = $bam;
    $prefix =~ s/\.bam$/\.filterFillIn/;
    print STDERR "Using prefix: " . basename($prefix) . "\n";
}


my $output_bam = "$prefix.bam";
open(OUTPUT,"| samtools view -hb - > $output_bam");


my $chh_file = "$prefix.chhString.txt" if $keep_string;
open(CHH_STRING,">$chh_file") if $keep_string;




# Define minimum value dependent on machine-type
my $min = (`samtools view $bam | head -1 | cut -f 1` =~ /^E/) ? "#" : "!";  # 
If ^E, then XTen min = 2/#; else 2500 and min = 0/!


# Read in BAM file
my $sam = Bio::DB::Sam -> new(-fasta=>$ref_fa,-bam=>$bam,-autoindex=>1);




# Print header with additional filterFillIn.pl line
my $header = $sam -> header -> text;
print OUTPUT "$header";
print OUTPUT 
"\@PG\tID:filterFillIn.pl\tPN:filterFillIn.pl\tVN\:$script_version\t";
print OUTPUT "DS:[Reads with 4+ mCHH outside of 1st ${fillIn_dist}bp of read 
are flagged as \"not passing filters\" by marking with 512/0x200 ";
print OUTPUT "bitwise flag with the CHH string included (with \"|\" after 
${fillIn_dist}bp) in the \"YF:Z:C|G{#}\" tag. ";
```

```
print OUTPUT "When R1 overlaps R2 ${fillIn_dist}bp fill-in region, set R2 base
qualities to 0 (2500) or 2 (X Ten) and report count in flag \"YO:i:#\"]";
print OUTPUT "\tCL\:$cmd\n";




# Parse reads by chromosome
my $mapped = 0;
my $unmapped = 0;
my $filtered = 0;


my $i = $sam -> features(-iterator=>1);


my $iter = 0 if $verbose;
while (my $read = $i->next_seq) {

    if ($verbose) {
       printf("#\t%s | Filtering %s ..... %d\n",strftime("%F,
%r",localtime),basename($bam),$iter) if $iter % 10e6 == 0;
       ++$iter;
    }


    # Reconstruct SAM format fields
    my @bam_line = split(/\t/,$read -> tam_line);


    if ( $bam_line[1] & 4 ) {
       ++$unmapped;
       print OUTPUT join("\t",@bam_line) . "\n";
       next;
    } else {
       ++$mapped;
    }


    my $isRevComp = $read -> reversed;
    # Get read from bitflag (note, actual R2's are flagged as R1 because
swapped furing bwameth.sh)
    # Tried to flip but breaks PileOmeth b/c that code expects R1 to have C>T
and R2 to have G>A
    my $r12 = ($bam_line[1] & 64) ? 'R1' : 'R2';

    # Change R1 (actually R2) overlapping base qualities to min (#|!) when
insert size less than readlength add change YO:i flag to 1
    # Min base quality will cause it to get skipped on pile up
    my($insert,$seq,$qual) = @bam_line[8..10];
    my $clip_n = min(length($seq),max(0,($fillIn_dist + length($seq)) -
abs($insert)));


    # Define last INCLUDED offset in read and use later to remove readthrough
fillIn effects on R1
    my $last_index = length($seq)-1 - $clip_n;


    # R1 (actually R2) may capture fillIn artifact at its 3' end which could
have higher base quality and get used over R2 (actually R1)
    if ($r12 eq 'R1' and $clip_n > 0) {  # R1 is actually R2
        my @qual = ($isRevComp) ? reverse(split("",$qual)) : split("",$qual);
        $bam_line[10] = ($isRevComp) ?
join("",reverse(@qual[0..$last_index],$min x min($clip_n,length($seq)))) :
            join("",@qual[0..$last_index],$min x min($clip_n,length($seq)));


        push(@bam_line,"YO:i:$clip_n");
    }



    my($ref,$matches,$query) = $read -> padded_alignment;
    if ($isRevComp) {
        $ref = &rev_comp($ref);
        $query = &rev_comp($query);
        $matches = reverse($matches);
    }

    my @ref = split('',$ref);
    my @query = split('',$query);
    # Establish boundary index in context of query--matches--reference strings
which may or may not be length of read (indels will expand it)
    my $boundary_index;
    if ($r12 eq 'R1') {
       if ($clip_n > 0) {
           my $counter = 0;
           for (my $i = $#query; $i >= 0; $i--) {
              ++$counter if $query[$i] ne '-';
              $boundary_index = $i and last if $counter == $clip_n;
           }
       } else {
           $boundary_index = $#query + 1;
       }
    } else {
        my $counter = 0;
        for (my $i = 0; $i < length($query); $i++) {
            ++$counter if $query[$i] ne '-';
            $boundary_index = $i and last if $counter == $fillIn_dist;
        }
    }


    # Declare arrays for CHH sequences
    # @chh_before: before fill-in region, <=11bp (index=10), typically
    # @chh_after: after fill-in region, >11bp (index=10)
    # @chh_filter: precise string after fill-in, including '-' for deleted
reference C's
    my(@chh_before,@chh_after);

    my $base = ($r12 eq 'R1') ? 'C' : 'G';
    my $context = ($base eq 'C') ? '[ACT]' : '[AGT]';
    my $regex = ($r12 eq 'R1') ? "$base$context\{2\}" : "$context\{2\}$base";



    my $offset = 0;
    my $result = index($ref,$base,$offset);  # Initiate, then while loop
indexes

    while ($result != -1) {  #As long as there's a match and offset has room
for trinucleotide
```

```
        my $tri = $ref[$result];

        if ($r12 eq 'R1') {
            my $j = $result + 1;
            while (length($tri)==1 and $j <= length($ref)-2) {
               $tri .= $ref[$j] if ($ref[$j] =~ /[ACGT]/);  # This is to skip
"-" when deletion and get next letter
               ++$j;
            }
            while (length($tri)==2 and $j <= length($ref)-1) {
               $tri .= $ref[$j] if ($ref[$j] =~ /[ACGT]/);
               ++$j;
            }
        } else {
            my $j = $result - 1;
            while (length($tri)==1 and $j >= 1) {
               $tri = "$ref[$j]$tri" if ($ref[$j] =~ /[ACGT]/);
               --$j;
            }
            while (length($tri)==2 and $j >= 0) {
               $tri = "$ref[$j]$tri" if ($ref[$j] =~ /[ACGT]/);
               --$j;
            }
        }


        # Keep CHH's before and after 9th base separate
        if ( $tri =~ /$regex/ ) {


            # R1 can read into fill-in artifact, so must remove from 3' end, R2
is 5' end
            # Will be reversed for R1, chh_string will read from end to front
so consistent with R2 direction
            if ($r12 eq 'R1') {
               ($result >= $boundary_index) ? unshift(@chh_before,$result) :
unshift(@chh_after,$result);
            } else {
               ($result <= $boundary_index) ? push(@chh_before,$result) :
push(@chh_after,$result);
            }
        }
        $offset = $result + 1;
        $result = index($ref,$base,$offset);
    }
    my $chh_string = join("",@query[@chh_before],"|",@query[@chh_after]);
    my $chh_filter_string = join("",@query[@chh_after]);

    push(@bam_line,"YF:Z:$chh_string");  # Adds this flag indicating CHH
string, with "|" marking break before and after fill-in

    if ($chh_filter_string =~ /^$base{4}/) {  # 4 or more mCHH outside of
fill-in is artifact
                                              # n=4; 1 - pbinom(n,p=0.3,q=n-1)
= 0.0081
                                             # highest reported meth ratios of
CHH are in brain, 20-25%
                                             # so used conservative p=0.3,
although p.value<0.01 if p=0.2 or p=0.25
        $bam_line[1] += 512 if !( $bam_line[1] & 512 );
        ++$filtered;
    }
    print OUTPUT join("\t",@bam_line) . "\n";
    print CHH_STRING "$chh_string\n" if $keep_string and $chh_string;
}
close(CHH_STRING);







my $filtered_rate = ($mapped > 0) ? $filtered/$mapped : 'NA';


# Print this to stout so can redirect to $QC_DIR instead of $prefix directory
my $stats = "$prefix.stats";
open(STATS,">$stats");
print STATS join("\t","#Mapped",qw( Unmapped Map.filtered Map.filtered.rate ))
. "\n";
print STATS join("\t",$mapped,$unmapped,$filtered,$filtered_rate) . "\n";
```

```
# SUBS ###########
sub usage {
    print STDERR "$0 -b <BAM> -r <REF FASTA> -c <CHROMS>\n";
    print STDERR "\t-b  BAM alignment\n";
    print STDERR "\t-r  Reference FASTA\n";
    print STDERR "\t-p  Output prefix [Default: trims *.bam]\n";
    print STDERR "\t-k  Keep CHH state strings [Outputs to
<PREFIX>.chhStates.txt\n";
    print STDERR "\t-d  Fill-in distance [Default: 11bp]\n";
    print STDERR "\t-v  Verbose mode\n";
    print STDERR "\t-h  Print this message\n";
    exit;
}




sub rev_comp {
    my $dna = shift;


    # reverse the DNA sequence
    my $revcomp = reverse($dna);


    # complement the reversed DNA sequence
    $revcomp =~ tr/ACGTacgt/TGCAtgca/;
    return $revcomp;
}
```

Supplemental references

Akalin A, Kormaksson M, Li S, Garrett-Bakelman FE, Figueroa ME, Melnick A, Mason CE. 2012. methylKit: a comprehensive R package for the analysis of genome-wide DNA methylation profiles. *Genome Biol* **13**: R87.

Dolzhenko E, Smith AD. 2014. Using beta-binomial regression for high-precision differential methylation analysis in multifactor whole-genome bisulfite sequencing experiments. *BMC Bioinformatics* **15**: 215.

Li H, Durbin R. 2009. Fast and accurate short read alignment with Burrows-Wheeler transform. *Bioinformatics* **25**: 1754–1760.

Martin M. 2011. Cutadapt removes adapter sequences from high-throughput sequencing reads. *EMBnet j* **17**: 10.

Pedersen BS, Eyring K, De S, Yang IV, Schwartz DA. 2014. Fast and accurate alignment of long bisulfite-seq reads.

Song Q, Decato B, Hong EE, Zhou M, Fang F, Qu J, Garvin T, Kessler M, Zhou J, Smith AD. 2013. A reference methylome database and analysis pipeline to facilitate integrative and comparative epigenomics. *PLoS ONE* **8**: e81148.