Supplementary data

Comparison of alignment software for genome-wide bisulphite sequence data

Aniruddha Chatterjee[1, 3#], Peter A. Stockwell[2# *], Euan J.Rodger[1], Ian M Morison[1, 3]

[1]Department of Pathology, Dunedin School of Medicine, University of Otago, 270 Great king street, Dunedin 9054, New Zealand

[2]Department of Biochemistry, University of Otago, 710 Cumberland street, Dunedin 9054, New Zealand

[3] National Research Centre for Growth and Development, New Zealand

# These authors contributed equally to this work.

Supplementary Table S1: Percentage of contamination by adaptor sequences in the dataset.

Dataset	Contamination percentage (bp)	No of reads containing adaptor sequence	% of reads containing adaptor sequence
100 bp	12.3	7987905	43.2
75 bp	3.8	3583521	19.4
60 bp	1.0	1240640	6.7

Supplementary Table S2: Percentage sequences trimmed by dynamic trimming[$].

Dataset	Percentage of the data (base pairs) trimmed by dynamic trimming	No of dynamically trimmed reads	% of reads that were dynamically trimmed
100 bp	12.9	12599267	68.2
75 bp	6.9	8027626	43.4

[$]The dynamic trimming was performed by *fastq_quality_trimmer,* a publicly available program distributed as a part of the fastx toolkit.

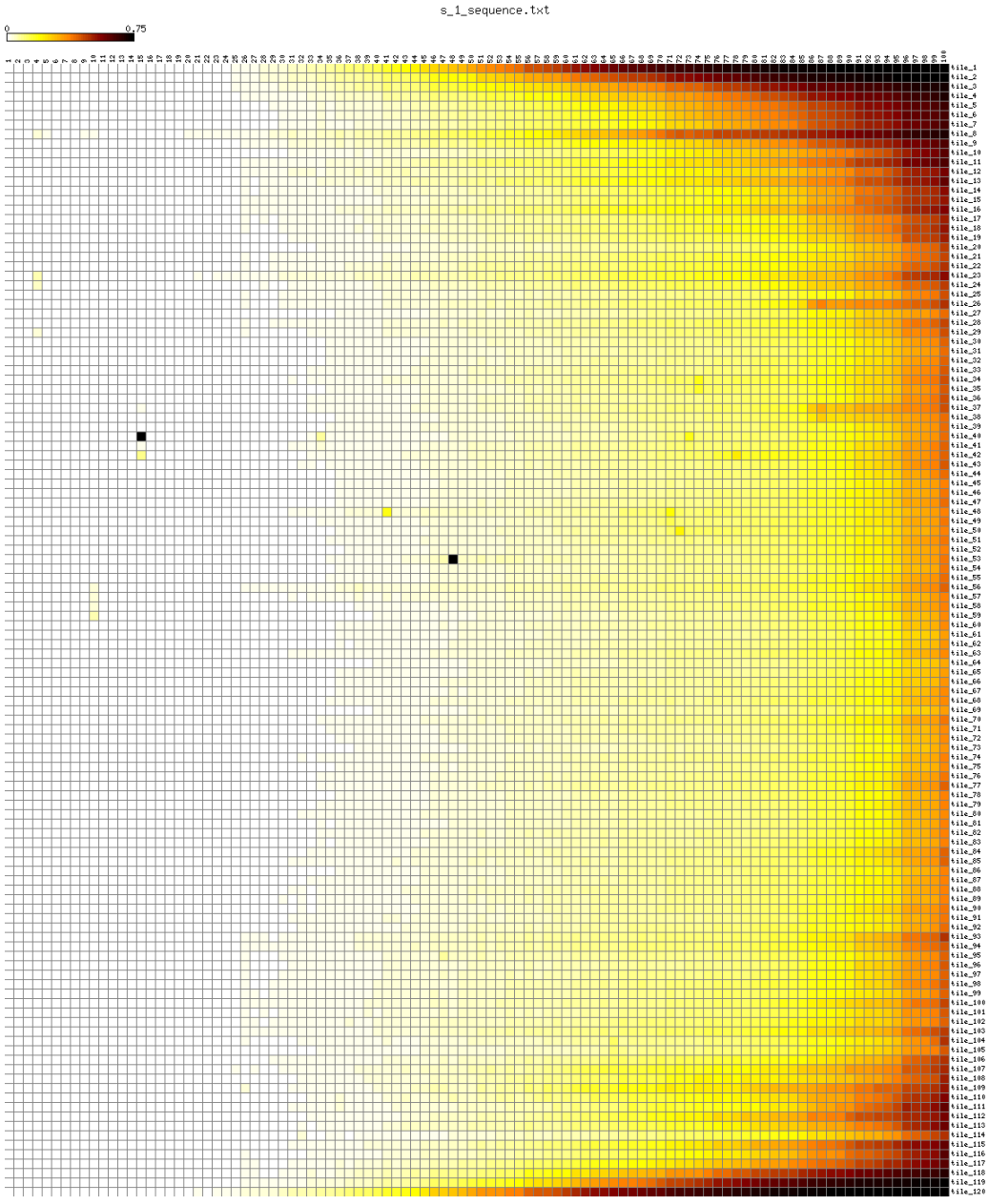
Supplementary Table S3: Effect of adaptor trimming and dynamic trimming on alignment efficiency .

Program	Percentage of all sequence reads showing unique alignment against unprocessed 100 bp dataset, then adaptor trimmed or dynamic trimmed			Percentage of all sequence reads showing unique alignment after hard trimming to 75 bp, then adaptor trimmed or dynamic trimmed		
	Before trim	After adaptor trim	After dynamic trim	Before trim	After adaptor trim	After dynamic trim
Bismark	30.9	58.8	43.4	42.2	60.8	54.2
BSMAP v1.2	36.0	66.1	48.7	55.5	68.5	61.3
RMAPBS*	44.3	71.0	-	65.1	73.0	-

*Note that RMAPBS requires all reads to have the same length for mapping. Our program *cleanadaptors*, trimmed the adaptor sequences from the reads and then padded them to full length with 'N's in order to perform the mapping. A consequence of this is that most were then rejected as being of low quality and failed alignment. Hence, only 10,917,041 reads were processed for the 100 bp trimmed run and 15,786,663 for 75 bp trimmed. For the same reason mapping with dynamically trimmed reads were not performed for RMAPBS.

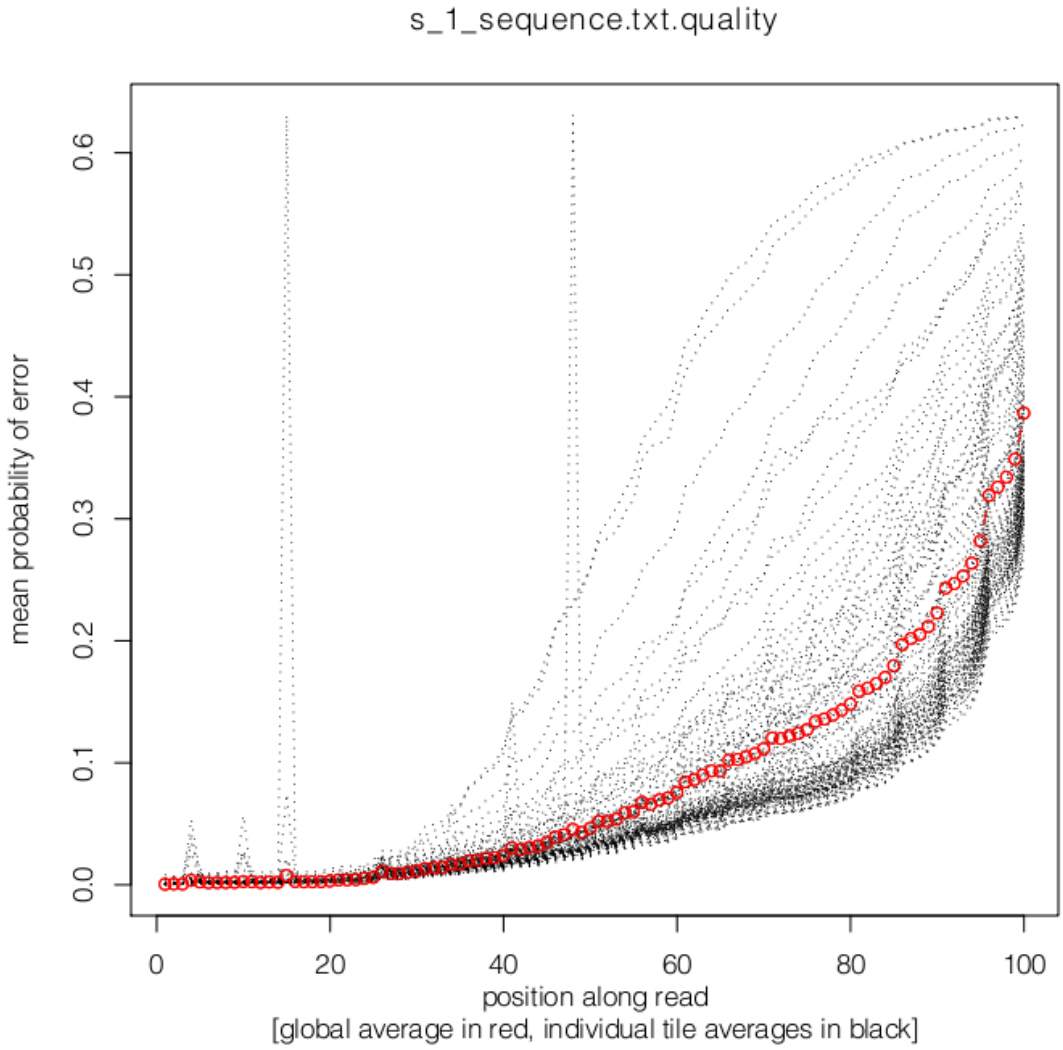
Supplementary Figure S1: SolexaQA heat map display of run quality.

100 bp reads are displayed horizontally, tiles vertically. The depth of colour represents the uncertainty of base calls with black equivalent to complete uncertainty. Tile 40 cycle 16 and tile 53 cycle 48 both show evidence of air bubbles while the poorer quality tiles at the top and bottom of the display probably result from flowcell dynamic effects. Note: SolexaQA analyses a portion of the data, by default 10,000 reads/tile are processed.



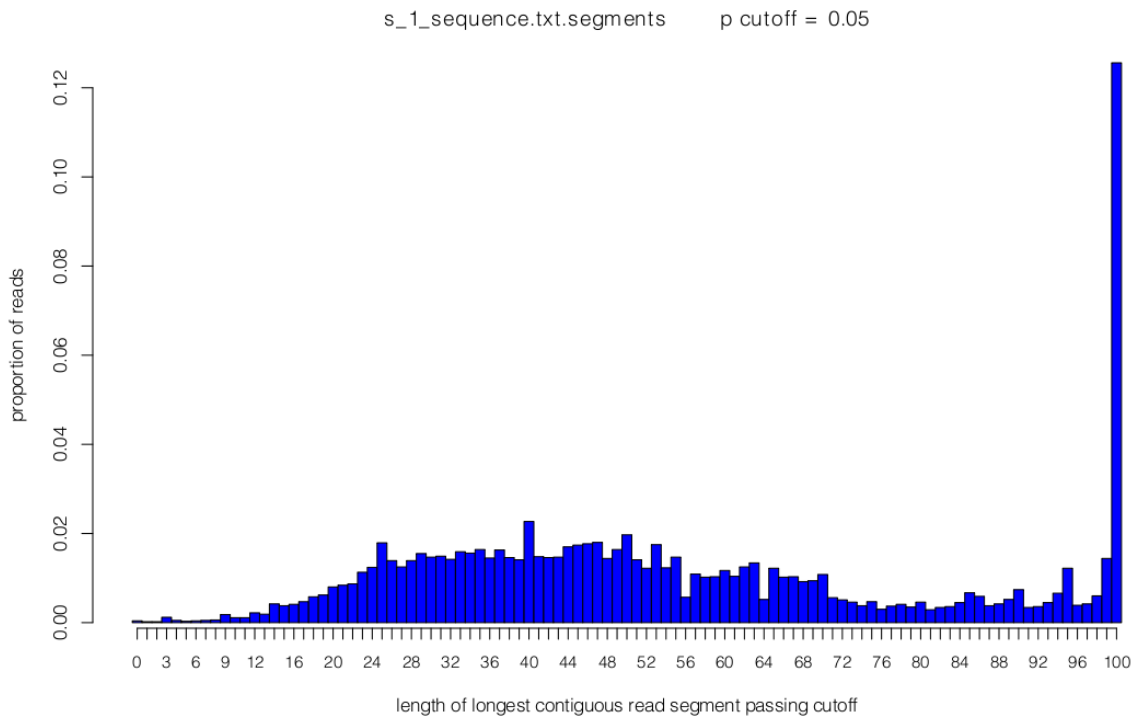
Supplementary Figure S2: SolexaQA plot of read errors *vs* cycle number.

The aberrant tiles from Figure S1 are displayed clearly, as is the progressive decline of read quality with cycle number. On the basis of this plot and that of Figure S1, the reads were trimmed to 75bp for further analysis.



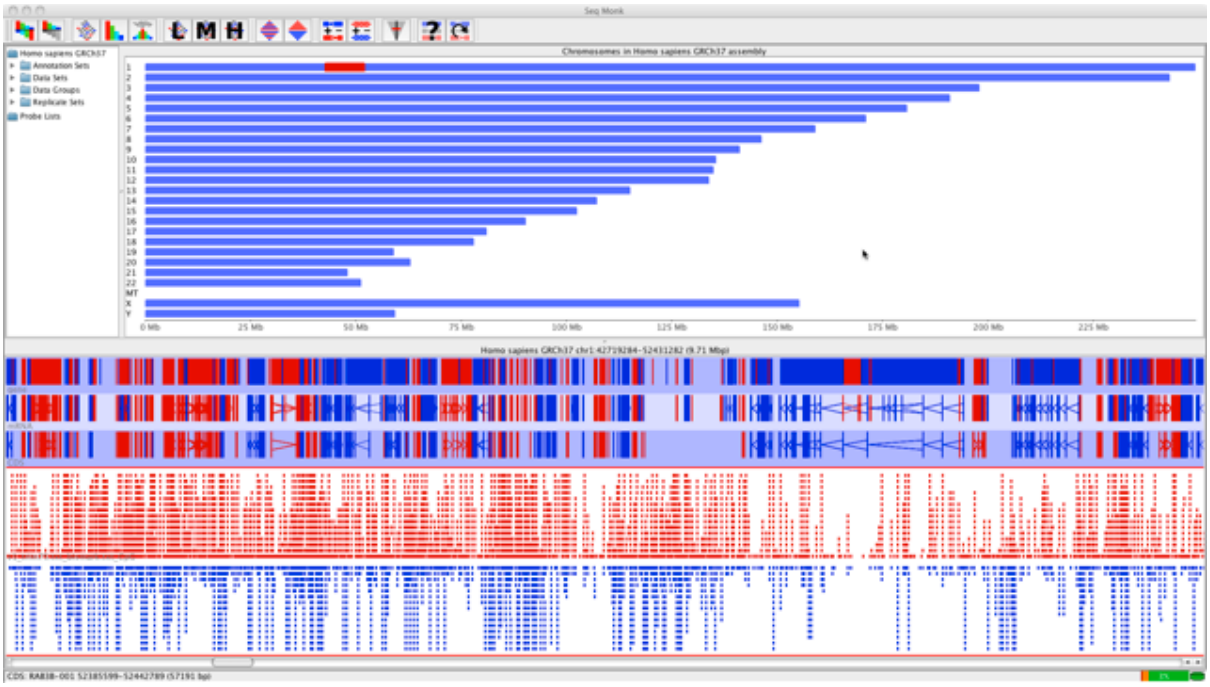
Supplementary Figure S3: SolexaQA histogram of proportion of reads passing a quality threshold

Bars represent the proportion of reads reaching various continuous read lengths with a quality score of 0.05 or better. At this level, there is a 1 in 20 chance that a base is misscalled.



Supplementary Figure S4: Seqmonk displaying Bismark data against Human genome GRCh37 build.

A 9.7 Mbp region of chromosome 1 is displayed. Methylated CpG positions are shown in the red pane below the gene, mRNA and CDS panes. Unmethylated CpGs are in the blue pane.



Supplementary Figure S5: Flow diagram of the analysis pipeline

The individual steps of the bioinformatics pipeline for methylation analysis (from bisulphite sequencing data) is described. *Cleanadaptors, rmapbsbed2cpg* and *mkrrgenome* (in bold) are our own in house developed programs. SolexaQA.pl , FastQC and Seqmonk are freely available tools. The pipeline as written is for a single, unindexed run : for indexed runs the pipeline would be applied to each indexed dataset separately.