Source data

Data presented in Chapter 4 is based on analyses of the same 1000 genomes project phase 3 dataset of 2533 low coverage WGS samples as used in Chapter 3. For alignment pipeline details, pileup parsing and reference & non-reference read frequency calculation see codebase for Chapter 3.

Pipeline for context dependent sequencing error rate measurement: MATLAB

All inputs used in the key scripts associated with Chapter 4 were generated as part of the analyses described in Chapter 3, for details see the codebase for Chapter 3.

Step 1: Generate motif spaces of interest

'Motif space' refers to the complete set of all possible sequence motifs of a given length. In our analyses we are assess sequencing error rate association for 2bp to 8bp motifs. This step generates the required motif spaces.

Code: SCRIPT_1_Motif_space_generator.m

Outputs		**Dimensions & data type**		
`Motif_spaces.mat`	`N2`	16 x 1	strings (length: 2)	Struct with 7 fields, each containing full set of motifs representing one motif space of a given length from 2bp ('N2') to 8bp ('N8').
	`N3`	64 x 1	strings (length: 3)	
	`N4`	256 x 1	strings (length: 4)	
	`N5`	1024 x 1	strings (length: 5)	
	`N6`	4096 x 1	strings (length: 6)	
	`N7`	16384 x 1	strings (length: 7)	
	`N8`	65536 x 1	strings (length: 8)	

<u>Algorithm structure</u>

```
>
  >>
    >>>
```

Step 2: **Find sequence motif locations in MT genome sequences**

Find all locations of each motif from a specific motif space within our analysed sample MT consensus sequences. Each location represents the sequence position of the 3'-most base of that motif instance. Script input variable `motif_spaces` is generated in the previous step (step 1), whereas the other inputs `SampleIDs` and `MT_consensus_sequences` were produced as part of the analysis described in chapter 3.

Code: SCRIPT_2_Motif_location_finder

Inputs		**Dimensions & data type**		
`Motif_spaces`	`N2 to N8`	*4^N x 1*	string	Sets of all possible 2bp to 8bp motifs
`MT_consensus_sequences`		*2533 x 16569*	char	MT consensus sequences of 2535 samples
`SampleIDs`		*2533 x 1*	string	List of sample IDs

Outputs *for each motif space*			**Dimensions & data type**		
`Motif_counts`	`L_strand`		*4^N x 2533*	double	Number of instances of each motif (row) that were found in mtDNA of each sample (column) *N – motif length*
	`H_strand`				
	`Total`				
`Motif_locations`	`L_strand`	`[motif]`	*Structure with 4^N fields* *Each field: 2533 x 1 cell*		Motif-specific cell arrays contain indices of L or H strand locations of that motif in each sample
	`H_strand`	`[motif]`			
`Motif_table`			*Table with 4^N rows,* *1 row per motif*		Full motif list, samples each motif was found in & total motif instances found across all samples

<u>Algorithm structure</u>

```
>
  >>
    >>>
```

Step 3: Calculate sequencing error rates associated with different motifs

Using motif locations generated in the previous step, extract the read counts for reference base matches and mismatches in forward and reverse sequencing directions relative to the motif orientation, and calculate the sum forward and reverse match and mismatch counts across all motif instances, and calculate the motif associated error rates as the difference between forward and reverse read mismatch fractions. Thus motif error rate measurement approach is based on research by Allhoff et al., 2010. For details see the main thesis text.

Script input variable `motif_spaces` is generated in the earlier step 1 and variables `Motif_locations` and `Motif_table` are generated in step 2, whereas inputs `SampleIDs`, `Reads`, `Reads_Ref` and `Reads_NonRef` were produced as part of the analysis described in Chapter 3.

Code: SCRIPT_3_motif_error_rate_calculator

Inputs		**Dimensions & data type**		
`SampleIDs`		*2533 x 1*	string	List of sample IDs
`Motif_spaces`	`N2 to N8`	*4^N x 1*	string	Sets of all possible 2bp to 8bp motifs
`Motif_table` *(for each motif space)*		*Table with 4^N rows,* *1 row per motif*		Full motif list, samples each motif was found in & total motif instances found across all samples
`Motif_locations` *(for each motif space)*	`L_strand`	*Structure with 4^N fields* *Each field: 2533 x 1 cell*		Motif-specific cell arrays contain indices of L or H strand locations of that motif in each sample
	`H_strand`			
`Reads`	`Forward`	*2533 x 16569*	double	Total `A` + `C` + `G` + `T` base call counts in each sequencing direction
	`Reverse`			
`Reads_Ref`	`Forward`	*2533 x 16569*	double	Number of reads in each sequencing direction that match the consensus sequence base type
	`Reverse`			
`Reads_NonRef`	`Forward`	*2533 x 16569*	double	Total reads in each sequencing direction supporting base types other than the consensus sequence base
	`Reverse`			

Outputs *for each motif space*		**Dimensions & data type**		
`Motif_counts`	`L_strand`	4^N x 2533	double	Analysed motif instances in each sample on L and H strands and in total across both strands. Rows – motifs, columns – samples
	`H_strand`			
	`Total`			
`Readcount_totals`	`FM`	4^N x 2533	double	Sample-level FM/FMM/RM/RMM read count totals across all analysed instances of each motif within each sample. Rows – motifs, columns – samples
	`FMM`			
	`RM`			
	`RMM`			
`Sample_MER`		4^N x 2533	table	Calculated motif sample-level error rates. Rows – motifs, columns – samples
`Population_MER`		4^N x 14	table	Total motif instances analysed across samples, population-level FM/FMM/RM/RMM read count totals and the overall calculated motif error rate.

FM (Forward Match) – reads in the same orientation as the motif that support reference base
FMM (Forward MisMatch) – reads in the same orientation as the motif that support a non-reference base.
RM (Reverse Match) – reads in the opposite orientation to the motif that support reference base
RMM (Reverse MisMatch) – reads in the opposite orientation to the motif that support a non-reference base.

RER (Reverse Error Rate) = RMM / (RM + RMM) – mismatch fraction in reverse direction (relative to motif orientation)
FER (Forward Error Rate) = FMM / (FM + FMM) – mismatch fraction in forward direction (relative to motif orientation)
ERD (Error Rate Difference) = FER - RER – difference between forward and reverse mismatch fractions

MER (Motif Error Rate) – term used instead of ERD in thesis text. Both terms are used interchangeably.

'F' and 'R' used in the FM/FMM/RM/RMM notation refer to the read direction relative to the motif orientation, whereas 'L' and 'H' strand notation is used for distinguishing between forward and reverse sequencing directions relative to the mtDNA reference sequence orientation.

Algorithm structure

```
 >
   >>
     >>>
```

Step 4: **Determine whether sequencing error rates associated with different motifs**

For each analysed motif, determine the statistical significance of the association between the reference base mismatch rates and the sequencing read direction relative to motif orientation using Fisher's exact and Chi squared tests. Testing is performed on 2x2 contingency tables consisting of FM/FMM/RM/RMM read counts. Fishers exact test is used preferentially, except where contingency table values exceed 10^7. Due to the limitations of Matlab fishertest() function, Chi squared test is used for contingency tables with values above 10^7 instead.

All script inputs are variables generated in the previous step (step 3) described above.

Code: SCRIPT_4_Motif_error_rate_significance

Inputs *for each motif space*		**Dimensions & data type**		
`Readcount_totals`	`FM`	4^N x 2533	double	Sample-level FM/FMM/RM/RMM read count totals across motif instances in each sample Rows – motifs, columns – samples
	`FMM`			
	`RM`			
	`RMM`			
`Sample_MER`		4^N x 2533	table	Calculated sample-level motif error rates
`Population_MER`		4^N x 14	table	Population-level FM/FMM/RM/RMM read count and the overall calculated motif error rate

Outputs *for each motif space*		**Dimensions & data type**		
`MER_stats_summary`		4^N x 2533	table	Population-level MER and its significance testing results, plus sample MER summary statistics and sample-level significance test result summary
`MER_Sample_level_stats`	`Result`	4^N x 2533	double	Sample-level MER significance testing results: Whether the motif was analysed (TRUE/FALSE), test type (FT/X2) and significance level used, test result (1/0/NaN) and the calculated p values. Rows – motifs, columns – samples
	`P_values`	4^N x 2533	double	
	`Test_type`	4^N x 2533	string	
	`Analysed`	4^N x 2533	logical	
	`Alpha`	1 x 1	double	
`MER_Global_stats`	`Result`	4^N x 1	double	Population-level MER significance testing results: Whether motif was analysed (TRUE/FALSE), test type (FT/X2) and significance level used, test result (1/0/NaN) and the calculated p values. Rows – motifs
	`P_values`	4^N x 1	double	
	`Test_type`	4^N x 1	string	
	`Analysed`	4^N x 1	logical	
	`Alpha`	1 x 1	double	

Algorithm structure

```
>
  >>
    >>>
```