Supplemental Material for "Detection of simple and complex *de novo* mutations without, with, or with multiple reference sequences"

Table of Contents

Introduction

Here, we provide further detail on the motivation, methods, data acquisition, processing, and analysis steps underlying the results presented in this work. Software, manuscript source, and additional resources (e.g. genome assemblies, machine-readable text files with paths to data downloads) are available at the following locations:

Purpose	URL	License
Genome assembly	https://github.com/mcveanlab/mccortex	MIT
Variant assembly	https://github.com/mcveanlab/Corticall	Apache 2.0
Additional resources	https://github.com/mcveanlab/Corticall/tree/master/manuscript	-

S1 Missing novelty in reference-based scans for *de novo* mutations

Our main manuscript presents an analysis wherein genomic novelty in several *P. falciparum* isolates is quantified using two separate methods and compared. This analysis is detailed below.

Consider a scenario in which genomic sequence sampled from an individual is absent or sufficiently disparate from a canonical reference genome. Many sequenced reads may fail to align to the reference correctly. Variants spanned by those reads may thus be rendered undiscoverable by reference-based approaches to variant detection. However, quantifying this missingness — the number of *de novo* mutations (DNMs) in a sample that remain to been discovered — is not straightforward.

We hypothesized that this missingness could be quantified by comparing a reference-*free* DNM novel sequence set (i.e. one constructed by performing *de novo* assembly on all members of the pedigree and emitting haplotype sequences unique to the children) to a reference-*based* novel sequence set (i.e. one constructed by aligning pedigree reads to a reference sequence, identifying mutations, and emitting haplotype sequences implied by those mutations). In an ideal scenario where divergence between samples and the reference is very low, the reference-free and reference-based analyses should be equivalent. With real data, the lack of equivalence can provide insight into how much novelty within a sample remains to be identified.

We performed such an analysis on *P. falciparum* progeny from the 3D7xHB3 experimental cross (20 progeny, 76-bp reads, ∼100x coverage, see Table S4 for further detail). The analysis workflow follows and is additionally diagrammed for clarity in Figure S1.

S1.1 Reference-free novel sequences

DNMs can be considered generators of novel genomic sequence — sequence that is present in an individual's genome but absent from their parents. Rather than counting mutational events explicitly, we instead count sequences unique to the genome of the child. We used the McCortex software (Turner et al., 2018) to split sequenced reads into fixed-length subsequences of length $k = 47$ ("k-mers"), aggressively removed sequencing errors and other artifacts using McCortex's k-mer-frequency-based cleaning and additional filters described in the main text, and compute the disjoint between child's k-mers and parental k-mers (Figure S1a-c). This provides a conservative reference- and alignment-free method of computing the *total* amount of novelty in a child's genome.

S1.2 Reference-based novel sequences

We aligned parental and progeny reads to the canonical *P. falciparum* reference of the 3D7 isolate using BWA-MEM (Li, 2013), called SNVs and small indels using HaplotypeCaller (Poplin et al., 2017) (GATK v4.0.2.1 with the `-ploidy 1` setting) and larger indels and structural variants using Delly (Rausch et al., 2012) (v0.7.8). We did not apply any filtration steps in an effort to generate a callset with maximum sensitivity.

To generate k-mers that could be compared with the reference-free analysis, we sought to permute the reference sequence with mutations from the callset and emit k-mers spanning the mutations. Given the high false discovery rate of the unfiltered callset, we could not rely on a single haplotype with all variants incorporated. Instead, we generated nearly all possible variant haplotypes within small genomic windows. At each putative mutation, we collected the previous

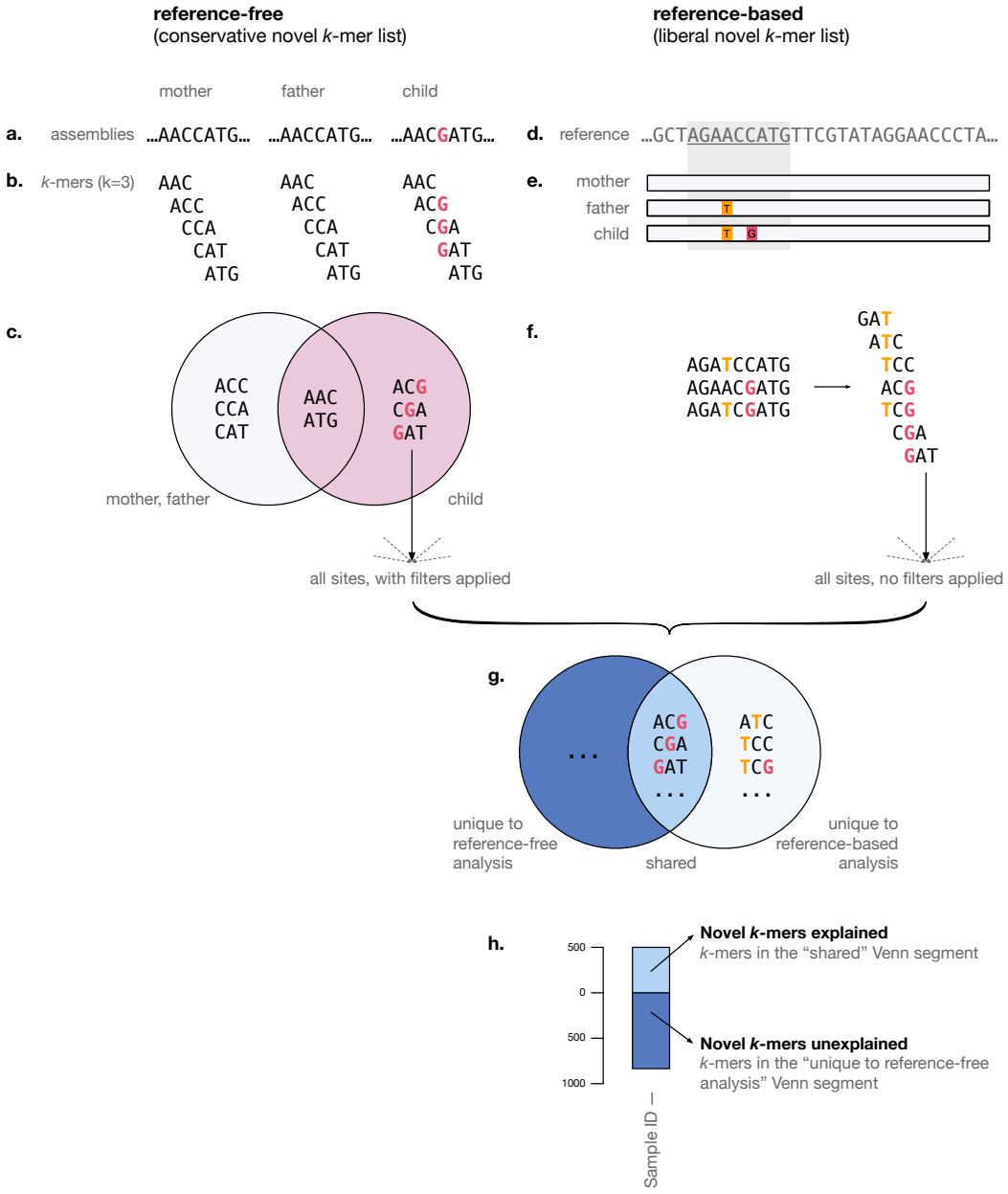

Figure S1. Procedure for comparing reference-free and reference-based scans for genomic novelty. *Reference-free pipeline:* (**a**) Reads are assembled *de novo* and error-cleaned into a multi-color de Bruijn graphs using the McCortex software. (**b**) Error-cleaned assemblies are converted to lists of parental and child k-mers ($k = 3$ depicted). (**c**) Lists of k-mers are compared, and novel k-mers (the child's disjoint) are extracted and subjected to additional filtering. *Reference-based pipeline:* (**d**) Reads are aligned to a reference sequence and variants are called using maximum sensitivity settings and no application of filters. (**e**) Windows around putative variants are chosen. (**f**) Variants within window are combinatorically used generate all possible haplotypes, then distilled to k-mers of the same size used in the reference-free pipeline. No further filtering is applied. *Comparison:* (**g**) All k-mers from other sites in the genome are aggregated. The Venn diagram of k-mers from the reference-free and reference-based analyses is computed. (**h**) Shared k-mers reflect those that are present in the reference-free analysis and recapitulated by the reference-based analysis. Those unique to the reference-free analysis reflect those that could not be recapitulated by the reference-based analysis. The k-mers unique to the reference-based analysis are discarded.

50-bp of reference sequence and a maximum of 10 variants within the immediately following 50-bp window. We combinatorically constructed all haplotypes from every possible subset of variants collected. Assuming all sites are biallelic, this effectively imposed a theoretical limit of $2^{10} = 1024$ haplotypes from any given site. Additionally, as Delly emits a consensus haplotype sequence for structural variants precisely identified, we included this sequence in the list of haplotypes to be processed. We then emitted *k*-mers from these haplotypes, and advanced to the next unprocessed variant. These steps are depicted in Figure S1d-f.

In our *P. falciparum* data, the restriction of processing the first 10 variants seen within each 50-bp window affected $0.26\% \pm 0.06\%$ of windows processed per sample (23277 ± 4325 windows per sample, 62 ± 19 windows reduced per sample).

S1.3 Comparison

Finally, we examined the Venn diagram of reference-free and reference-based novel *k*-mer lists, identifying shared *k*-mers (present in both analyses) and reference-free-unique *k*-mers (novel *k*-mers not recapitulated by the reference-based analysis). Novel *k*-mers unique to the reference-based analysis were discarded as these largely originate from the combinatoric haplotype construction procedure and were not true genomic sequences. Figure S1g-h show these final comparison steps.

S2 Data processing: long-read sequencing, assembly, and alignment

S2.1 Data availability

PacBio RSII raw read ERA accession numbers, processed assembly download paths, and annotation file download paths are provided below:

Table S1. Pacbio read and assembly metadata and download paths.

source code	secondary sample accession	origin	culturing	sequencing	chemistry	fasta
HB3	ERS712858	Honduras	SI	SI	P6C4	FASTA
DD2	ERS639545	Vietnam	SI	SI	P6C4	FASTA
7G8	ERS686280	Brazil	SI	SI	P6C4	FASTA
GB4	ERS3566948	Ghana	NIH	CSHL	P6C4	FASTA
803	ERS3566949	Cambodia	NIH	CSHL	P6C4	FASTA
36F11	ERS3119776	803xGB4 progeny	NIH	CSHL	P6C4	FASTA

A machine-readable version of this manifest is available at `https://github.com/mcveanlab/Corticall/blob/master/manuscript/manifest.pacbio.txt`.

S2.2 Library preparation

To facilitate draft reference genome construction, we obtained high molecular weight genomic DNA (HMW gDNA) for seven *P. falciparum* parasites: all six parental clones spanning the four experimental crosses (3D7, HB3, DD2, 7G8, GB4, 803) and one progeny clone from the 803xGB4 cross (36F11). Parasite lines for 3D7, HB3, DD2, and 7G8 were produced by the Kwiatkowski lab at the Wellcome Trust Sanger Institute (SI). Lines for 803, GB4, and 36F11 were produced by the National Institute of Allergy and Infectious Diseases at the NIH (NIH). All cultures were maintained under standard conditions (Trager and Jensen, 1976).

We obtained $6-20$ μg of HMW gDNA from each haploid parasite culture. QC was performed with NanoDrop spectrophotometers (Thermo Fisher Scientific) to verify gDNA purity. 20 kbp insert SMRTbell libraries were generated per sample with a Blue-Pippin (Sage Science) size selection range of $10-50$ kbp.

For additional assembly polishing of the HB3, DD2, and 7G8 genomes, we generated short fragment libraries using 0.5 μg of DNA and a PCR-free library construction method (Kozarewa et al., 2009). These libraries were sequenced using on MiSeq Illumina instruments, generating 250 bp paired-end reads and mean fragment length of 500 bp.

PacBio RSII sequencing of each library took place at two facilities. 3D7, GB4, 803, and 36F11 were sequenced at Cold Spring Harbor Laboratory's Next Generation Genomics Shared Resource (CSHL). HB3, DD2, and 7G8 were sequenced at the Sanger Institute.

S2.3 *De novo* assembly

3D7, HB3, DD2, and 7G8 assemblies on the libraries described above have been previously reported and publicly released (Otto et al., 2018). Summarizing, raw reads were assembled using

the HGAP2 software using default parameters and a genome size of 23.5 Mbp. The HGAP assemblies were further improved by removing contaminating sequences, short contigs, and overlapping contigs; merging contigs with substantial overlap length; and polishing using the 250 bp Illumina reads aligned to the assembly.

Remaining samples GB4, 803, and 36F11 were assembled with the HGAP3 software.

S2.4 Contaminant removal

To remove possible contaminants from all assemblies, we ran contigs through BLAST (Altschul et al., 1990), excluding any contig with a match to an organism other than *P. falciparum* in the `nt` database (updated Oct 2017).

S2.5 Pseudochromosome contiguation and gene annotation

To reorient contigs to match canonical reference orientation, build scaffolds representing whole chromosomes, and annotate genes, we ran all assemblies through the Companion webserver (Steinbiss et al., 2016). We set the `Reference organism` setting as `Plasmodium falciparum 3D7`. We specified that both *ab initio* gene finding and existing gene model transfer be performed, specifying the `Strain` option for the RATT (Otto et al., 2011) gene model transfer tool. All other settings remained as software defaults. Note that Companion produced scaffolds representing all 14 autosomes in the *P. falciparum* genome, but does not automatically recognize and circularize the mitochondria and apicoplast genomes. These linear contigs, along with contigs that could not be placed in an autosome, are concatenated in a separate contig named with a "_00" suffix. Hence, scaffolding by pseudochromosome contiguation results in 15 total scaffolds per assembly.

S2.6 Annotation of repetitive regions and core/non-core compartments

We annotated repetitive sequences by applying the RepeatMasker (Smit et al., 2013) software to each genome using the maximum sensitivity `-s` option and the `-species 'plasmodium falciparum'` argument. We annotated core and non-core regions of each genome with the (Ozer et al., 2014) software, processing each chromosome across all parental genomes simultaneously.

S2.7 Quality assessment

We first estimated the overall quality of our assemblies by performing a scaffold-to-chromosome alignment of our proof-of-principle 3D7 draft to the finished reference genome of the same parasite (Gardner et al., 2002) using MUMmer (Kurtz et al., 2004). The alignments are visualized as a multi-dotplot in Figure S2, an extension of a dot plot that depicts alignments as two dimensional matrices with target and query sequences on the x and y axes respectively (Gibbs and Mcintyre, 1970). Most chromosomes are assembled completely, and the overwhelming majority of the assembly appears on-diagonal (indicating successful one-to-one reconstruction). Elements appearing off-diagonal could represent misassembly. However, note that most of these off-diagonal elements occur towards the extremes of each chromosome. Given that the reference genome was constructed with Sanger reads an order of magnitude shorter than the PacBio reads, it is possible some repetitive regions have been collapsed or misplaced, contributing to this nominal error rate.

We called variants between the two assemblies to quantify errors using MUMmer, finding 3,357 SNPs, 11,620 insertions, and 4,603 deletions. Overall, the SNP, insertion, and deletion rates are exceedingly low: amounting to 19,580 events in a 23 Mbp genome (0.17%). The insertion rate is much higher than that of deletions and SNPs, perhaps due to the dominant insertion error mode of the PacBio sequencing instrument. All chromosomes appear reasonably similar in performance. Based on these measurements of the error rate, we estimate the quality of the PacBio assembly of the 3D7 isolate to be approximately Q31[1], or less than one error per thousand bases.

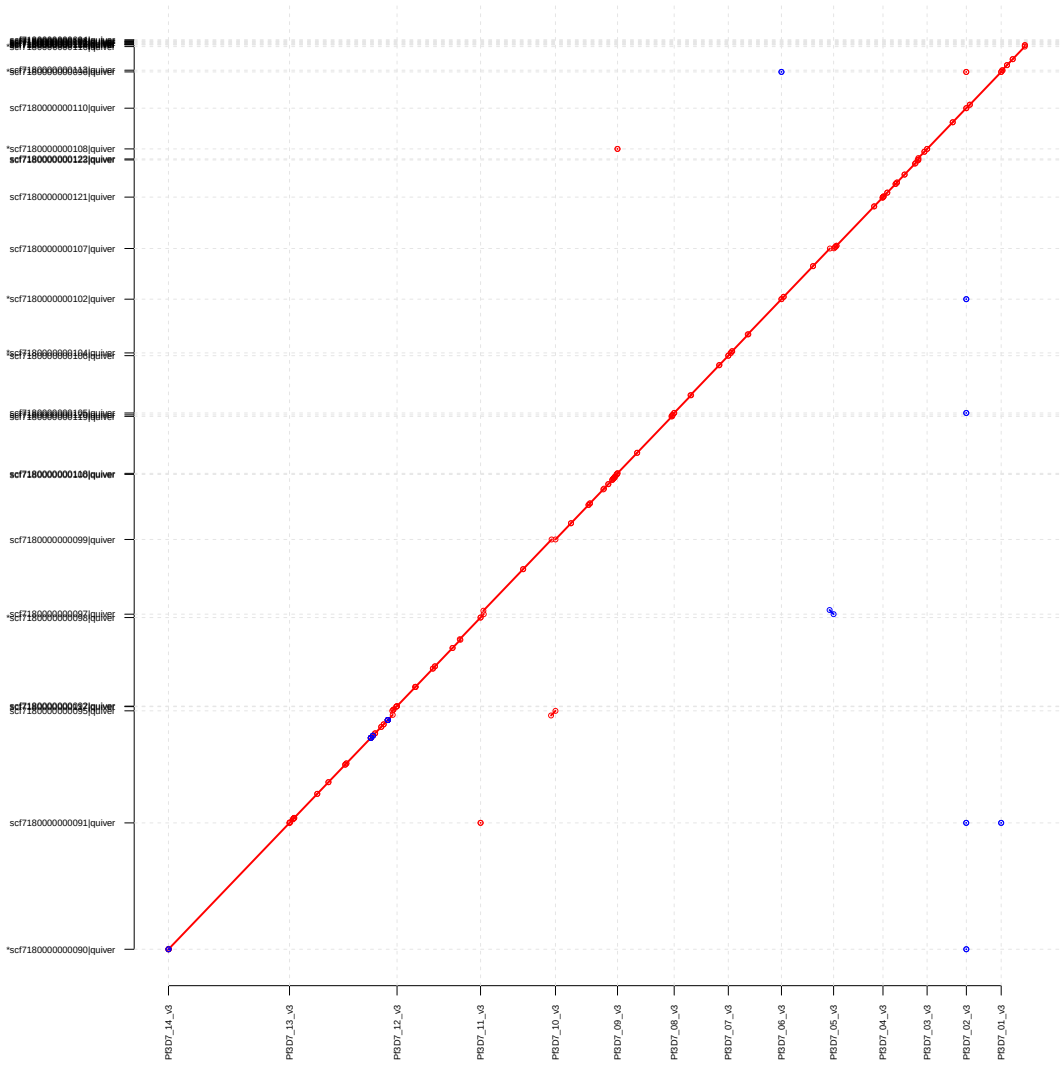

Figure S2. Alignment of contigs from the 3D7 draft to the finished reference assembly, with sequences from reference and draft assembly on the x and y-axis, respectively. Each contig from the draft assembly is represented by a line segment terminated at either end with circles. Red denotes forward alignment of contigs to the reference, blue denotes reverse alignment.

[1] $Q = -10\log_{10}(q) = -10\log_{10}((11,620 + 4,603 + 3,357)/23,332,831)$

Table S2. Metrics on PacBio sequencing and assemblies

	3D7 (ref)	3D7 (draft)	HB3	DD2	7G8	GB4	803	36F11
Assembler	–	HGAP2	HGAP2	HGAP2	HGAP2	HGAP3	HGAP3	HGAP3
Contigs	–	34	28	16	17	104	35	33
Scaffolds[a]	16	15	15	15	15	15	15	15
Length (bp)	23332839	23729641	22813863	22682439	22832395	27105702	23074973	23429077
Length (autosomes) (bp)	22671410	23194002	22671410	22641998	22775193	24613844	22772320	22989675
Core genome (bp)	20574502	–	21047328	21058329	21137184	21514228	21093156	–
Accessory genome (bp)	2096908	–	1624082	1583669	1638009	3099616	1679164	–
Genes (autosomes)	5561	5141	5130	5166	5219	5041	4501	4174
Quality (MUMmer)	–	31	–	–	–	–	–	–
Quality (Graph)	–	28	29	31	27	23	28	25

[a] In the draft reference sequences, the circular mitochondrial and apicoplast genomes are grouped together in a non-autosomal scaffold, resulting in one less scaffold than the canonical reference.

As we did not have high-quality, finished genomes against which to compare the other assemblies, we devised a quality metric approximation based on *k*-mer counting. We used our graph processing software to find *k*-mers present in the PacBio assembly and absent in the Illumina data for the same sample. We retained only one *k*-mer per variant event, discarding adjacent *k*-mers that tag the same event. We computed assembly quality using the number of retained *k*-mers as an estimate of the number of bases different between the PacBio and Illumina data.

We note that both measurements of assembly quality are pessimistic estimates. Our MUMmer-based estimates are predicated on the assumption that any differences between our 3D7 assembly and the canonical reference assembly indicate errors in our draft. In long, repetitive regions of the genome, this assumption may not be accurate as the Sanger reads used to assemble the reference were an order of magnitude shorter than the PacBio reads, potentially leading to collapsed repeats. Similarly for the *k*-mer counting estimate, while the PacBio RSII reads are known to have a much higher error rate than Illumina reads, it is not necessarily the case that all discrepancies between PacBio assemblies and Illumina reads should be adjudicated in favor of the latter.

Per-sample sequencing and final assembly details are presented in Table S2.

S3 Data processing: short-read sequencing, alignment, and assembly

Table S3: Illumina metadata and download paths.

cross	sample	source code	accession	fastq end1 ftp	fastq end2 ftp
3D7xHB3	PG0051-C	3D7_Glasgow	ERR019061	FASTQ (END1)	FASTQ (END2)
3D7xHB3	PG0052-C	HB3_Glasgow	ERR019054	FASTQ (END1)	FASTQ (END2)
3D7xHB3	PG0053-C	XP3	ERR019067	FASTQ (END1)	FASTQ (END2)
3D7xHB3	PG0054-C	XP4	ERR019062	FASTQ (END1)	FASTQ (END2)
3D7xHB3	PG0055-C	XP5	ERR019066	FASTQ (END1)	FASTQ (END2)
3D7xHB3	PG0056-C	XP8	ERR019068	FASTQ (END1)	FASTQ (END2)
3D7xHB3	PG0057-C	XP9	ERR019069	FASTQ (END1)	FASTQ (END2)
3D7xHB3	PG0058-C	XP24	ERR019063	FASTQ (END1)	FASTQ (END2)
3D7xHB3	PG0060-C	XP52	ERR019058	FASTQ (END1)	FASTQ (END2)
3D7xHB3	PG0061-C	X2	ERR019059	FASTQ (END1)	FASTQ (END2)
3D7xHB3	PG0062-C	X4	ERR019070	FASTQ (END1)	FASTQ (END2)
3D7xHB3	PG0063-C	X5	ERR019060	FASTQ (END1)	FASTQ (END2)
3D7xHB3	PG0064-C	X6	ERR019071	FASTQ (END1)	FASTQ (END2)
3D7xHB3	PG0065-C	X10	ERR019064	FASTQ (END1)	FASTQ (END2)
3D7xHB3	PG0066-C	X11	ERR019072	FASTQ (END1)	FASTQ (END2)
3D7xHB3	PG0067-C	X12	ERR019073	FASTQ (END1)	FASTQ (END2)
3D7xHB3	PG0068-C	X30	ERR019065	FASTQ (END1)	FASTQ (END2)
3D7xHB3	PG0069-C	X33	ERR019055	FASTQ (END1)	FASTQ (END2)
3D7xHB3	PG0070-C	X35	ERR019056	FASTQ (END1)	FASTQ (END2)
3D7xHB3	PG0071-C	X39	ERR019074	FASTQ (END1)	FASTQ (END2)
HB3xDD2	PG0004-CW	HB3_Ferdig	ERR012788	FASTQ (END1)	FASTQ (END2)
HB3xDD2	PG0008-CW	DD2_Ferdig	ERR012840	FASTQ (END1)	FASTQ (END2)
HB3xDD2	PG0015-C	B1SD	ERR019044	FASTQ (END1)	FASTQ (END2)
HB3xDD2	PG0016-C	QC13	ERR012895	FASTQ (END1)	FASTQ (END2)
HB3xDD2	PG0017-C	QC01	ERR019050	FASTQ (END1)	FASTQ (END2)
HB3xDD2	PG0018-C	B4R3	ERR019042	FASTQ (END1)	FASTQ (END2)
HB3xDD2	PG0019-C	SC05	ERR019051	FASTQ (END1)	FASTQ (END2)
HB3xDD2	PG0020-C	TC08	ERR019052	FASTQ (END1)	FASTQ (END2)
HB3xDD2	PG0021-C	GC03	ERR015447	FASTQ (END1)	FASTQ (END2)
HB3xDD2	PG0024-C	3BD5	ERR019053	FASTQ (END1)	FASTQ (END2)
HB3xDD2	PG0027-C	TC05	ERR015450	FASTQ (END1)	FASTQ (END2)
HB3xDD2	PG0030-C	7C188	ERR019046	FASTQ (END1)	FASTQ (END2)
HB3xDD2	PG0031-C	7C408	ERR015458	FASTQ (END1)	FASTQ (END2)
HB3xDD2	PG0034-C	7C3	ERR019047	FASTQ (END1)	FASTQ (END2)
HB3xDD2	PG0037-C	7C20	ERR015451	FASTQ (END1)	FASTQ (END2)
HB3xDD2	PG0038-C	7C111	ERR015457	FASTQ (END1)	FASTQ (END2)
HB3xDD2	PG0039-C	7C140	ERR015454	FASTQ (END1)	FASTQ (END2)
HB3xDD2	PG0040-Cx	7C159	ERR107475	FASTQ (END1)	FASTQ (END2)
HB3xDD2	PG0041-C	7C170	ERR015446	FASTQ (END1)	FASTQ (END2)

HB3xDD2	PG0042-C	7C183	ERR015448	FASTQ (END1)	FASTQ (END2)
HB3xDD2	PG0043-C	7C421	ERR015459	FASTQ (END1)	FASTQ (END2)
HB3xDD2	PG0044-C	7C424	ERR019043	FASTQ (END1)	FASTQ (END2)
HB3xDD2	PG0045-C	QC23	ERR012892	FASTQ (END1)	FASTQ (END2)
HB3xDD2	PG0046-Cx	7C46	ERR107476	FASTQ (END1)	FASTQ (END2)
HB3xDD2	PG0047-C	7C126	ERR015452	FASTQ (END1)	FASTQ (END2)
HB3xDD2	PG0048-C	7C7	ERR019049	FASTQ (END1)	FASTQ (END2)
7G8xGB4	PG0077-C	JC3	ERR027112	FASTQ (END1)	FASTQ (END2)
7G8xGB4	PG0077-CW	JC3	ERR045636	FASTQ (END1)	FASTQ (END2)
7G8xGB4	PG0078-C	QF5	ERR029092	FASTQ (END1)	FASTQ (END2)
7G8xGB4	PG0078-CW	QF5	ERR045638	FASTQ (END1)	FASTQ (END2)
7G8xGB4	PG0079-C	JF6	ERR027102	FASTQ (END1)	FASTQ (END2)
7G8xGB4	PG0079-CW	JF6	ERR045637	FASTQ (END1)	FASTQ (END2)
7G8xGB4	PG0080-C	TF1	ERR027103	FASTQ (END1)	FASTQ (END2)
7G8xGB4	PG0081-CW	DEV_18_05_11	ERR045633	FASTQ (END1)	FASTQ (END2)
7G8xGB4	PG0082-C	WC4	ERR029093	FASTQ (END1)	FASTQ (END2)
7G8xGB4	PG0083-C	7G8_NIH	ERR027099	FASTQ (END1)	FASTQ (END2)
7G8xGB4	PG0084-C	GB4_NIH	ERR027100	FASTQ (END1)	FASTQ (END2)
7G8xGB4	PG0087-C	JB8	ERR029091	FASTQ (END1)	FASTQ (END2)
7G8xGB4	PG0088-C	KH7	ERR027111	FASTQ (END1)	FASTQ (END2)
7G8xGB4	PG0090-C	KC2	ERR027116	FASTQ (END1)	FASTQ (END2)
7G8xGB4	PG0091-C	KA6	ERR027117	FASTQ (END1)	FASTQ (END2)
7G8xGB4	PG0093-C	XB3	ERR029105	FASTQ (END1)	FASTQ (END2)
7G8xGB4	PG0094-C	D2	ERR027106	FASTQ (END1)	FASTQ (END2)
7G8xGB4	PG0094-CW	D2_18_05_11	ERR045632	FASTQ (END1)	FASTQ (END2)
7G8xGB4	PG0095-C	NIC	ERR027107	FASTQ (END1)	FASTQ (END2)
7G8xGB4	PG0096-C	NF10	ERR027108	FASTQ (END1)	FASTQ (END2)
7G8xGB4	PG0097-C	WF12	ERR027109	FASTQ (END1)	FASTQ (END2)
7G8xGB4	PG0098-C	DAN	ERR027110	FASTQ (END1)	FASTQ (END2)
7G8xGB4	PG0099-C	JB12	ERR029146	FASTQ (END1)	FASTQ (END2)
7G8xGB4	PG0100-C	JE11	ERR029404	FASTQ (END1)	FASTQ (END2)
7G8xGB4	PG0101-C	KC5	ERR029147	FASTQ (END1)	FASTQ (END2)
7G8xGB4	PG0102-C	XF12	ERR029143	FASTQ (END1)	FASTQ (END2)
7G8xGB4	PG0102-CW	XF12_18_05_11	ERR045635	FASTQ (END1)	FASTQ (END2)
7G8xGB4	PG0104-C	KB8	ERR029148	FASTQ (END1)	FASTQ (END2)
7G8xGB4	PG0105-C	XD8	ERR029144	FASTQ (END1)	FASTQ (END2)
7G8xGB4	PG0105-CW	XD8_13_05_11	ERR045628	FASTQ (END1)	FASTQ (END2)
7G8xGB4	PG0107-C	JON	ERR029408	FASTQ (END1)	FASTQ (END2)
7G8xGB4	PG0109-C	XG10	ERR029405	FASTQ (END1)	FASTQ (END2)
7G8xGB4	PG0110-C	LC12	ERR171454	FASTQ (END1)	FASTQ (END2)
7G8xGB4	PG0111-C	JC9	ERR029409	FASTQ (END1)	FASTQ (END2)
7G8xGB4	PG0111-CW	JC9_18_05_11	ERR045634	FASTQ (END1)	FASTQ (END2)
7G8xGB4	PG0112-C	AUD	ERR029406	FASTQ (END1)	FASTQ (END2)
7G8xGB4	PG0113-CW	JH6_12_05_11	ERR045626	FASTQ (END1)	FASTQ (END2)

803xGB4	PG0443-C	803	ERR570006 FASTQ (END1) FASTQ (END2)
803xGB4	PG0050-CX2	GB4	ERR570014 FASTQ (END1) FASTQ (END2)
803xGB4	PG0445-C	11H5	ERR570030 FASTQ (END1) FASTQ (END2)
803xGB4	PG0446-C	36F11	ERR570038 FASTQ (END1) FASTQ (END2)
803xGB4	PG0447-C	36H9	ERR656280 FASTQ (END1) FASTQ (END2)
803xGB4	PG0448-C	37D9	ERR656283 FASTQ (END1) FASTQ (END2)
803xGB4	PG0450-C	39A4	ERR656289 FASTQ (END1) FASTQ (END2)
803xGB4	PG0451-C	39C5	ERR656292 FASTQ (END1) FASTQ (END2)
803xGB4	PG0453-C	43H3	ERR656298 FASTQ (END1) FASTQ (END2)
803xGB4	PG0455-C	46G9	ERR570015 FASTQ (END1) FASTQ (END2)
803xGB4	PG0458-C	61A12	ERR570039 FASTQ (END1) FASTQ (END2)
803xGB4	PG0459-C	61D3	ERR656281 FASTQ (END1) FASTQ (END2)
803xGB4	PG0460-C	61E6	ERR656284 FASTQ (END1) FASTQ (END2)
803xGB4	PG0461-C	61E8	ERR656287 FASTQ (END1) FASTQ (END2)
803xGB4	PG0462-C	71D6	ERR656290 FASTQ (END1) FASTQ (END2)
803xGB4	PG0463-C	76H10	ERR656293 FASTQ (END1) FASTQ (END2)
803xGB4	PG0464-C	85D3	ERR656296 FASTQ (END1) FASTQ (END2)
803xGB4	PG0465-C	87A11	ERR656299 FASTQ (END1) FASTQ (END2)
803xGB4	PG0466-C	87E7	ERR570008 FASTQ (END1) FASTQ (END2)
803xGB4	PG0467-C	40G11	ERR570016 FASTQ (END1) FASTQ (END2)
803xGB4	PG0469-C	50C5	ERR570032 FASTQ (END1) FASTQ (END2)
803xGB4	PG0470-C	24G11	ERR570040 FASTQ (END1) FASTQ (END2)
803xGB4	PG0471-C	36D5	ERR656282 FASTQ (END1) FASTQ (END2)
803xGB4	PG0472-C	36E5	ERR656285 FASTQ (END1) FASTQ (END2)
803xGB4	PG0473-C	38G5	ERR656288 FASTQ (END1) FASTQ (END2)
803xGB4	PG0474-C	39C3	ERR656291 FASTQ (END1) FASTQ (END2)
803xGB4	PG0475-C	4E8	ERR656294 FASTQ (END1) FASTQ (END2)
803xGB4	PG0476-C	34F5	ERR656297 FASTQ (END1) FASTQ (END2)
803xGB4	PG0492-C	34B1	ERR905451 FASTQ (END1) FASTQ (END2)
803xGB4	PG0494-C	35C2	ERR905453 FASTQ (END1) FASTQ (END2)
803xGB4	PG0495-C	38A6	ERR905454 FASTQ (END1) FASTQ (END2)
803xGB4	PG0496-C	38E11	ERR905455 FASTQ (END1) FASTQ (END2)
803xGB4	PG0498-C	39G5	ERR905457 FASTQ (END1) FASTQ (END2)
803xGB4	PG0499-C	39H5	ERR905458 FASTQ (END1) FASTQ (END2)
803xGB4	PG0500-C	40A6	ERR905459 FASTQ (END1) FASTQ (END2)
803xGB4	PG0501-C	40B12	ERR905460 FASTQ (END1) FASTQ (END2)
803xGB4	PG0502-C	40F4	ERR905461 FASTQ (END1) FASTQ (END2)
803xGB4	PG0503-C	40G2	ERR905462 FASTQ (END1) FASTQ (END2)
803xGB4	PG0504-C	43E5	ERR905463 FASTQ (END1) FASTQ (END2)
803xGB4	PG0505-C	44D4	ERR905464 FASTQ (END1) FASTQ (END2)
803xGB4	PG0508-C	76H10-Tk13	ERR905468 FASTQ (END1) FASTQ (END2)
803xGB4	PG0509-C	85G7	ERR905469 FASTQ (END1) FASTQ (END2)
803xGB4	PG0510-C	88C9	ERR905470 FASTQ (END1) FASTQ (END2)
803xGB4	PG0512-C	37F12	ERR905473 FASTQ (END1) FASTQ (END2)

S3.1 Data availability

Raw reads for *P. falciparum* parents and progeny were provided to us by the MalariaGen project (Miles et al., 2016). ERA accession numbers and fastq download paths are provided in Table S3.

A machine-readable version of this manifest is available at https://github.com/mcveanlab/Corticall/blob/master/manuscript/manifest.illumina.txt.

S3.2 Library preparation

We obtained short-read, paired-end whole genome sequence data for parent and progeny clones of the 3D7xHB3 (Walliker et al., 1987), HB3xDD2 (Wellems et al., 1990), 7G8xGB4 (Hayton et al., 2008), and 803xGB4 (Sá et al., 2018) crosses. All samples were sequenced on Illumina platforms between 2010 and 2014 using a PCR-free library preparation protocol intended to reduce coverage biases associated with AT-rich templates. Avoiding PCR during library construction also removes the issue of replication errors that occur in early cycles being propagated to all subsequent copies, thus masquerading as DNMs. Across all four crosses, and excluding parasite clones, we obtained 127 samples (8 parents, 119 progeny). The data is summarized in Table S4.

S3.3 Alignment

Table S4. Summary of sequencing data for four *P. falciparum* crosses

	3D7xHB3	HB3xDD2	7G8xGB4	803xGB4
Parents	2	2	2	2
Progeny	18	24	35	42
Read length (bp)	76	76	76	100
Insert size (bp)	300 ± 29	253 ± 48	293 ± 21	222 ± 10
Coverage	99 ± 38	121 ± 91	110 ± 40	205 ± 106
Platform	Illumina GAII	Illumina GAII	Illumina GAII	Illumina HiSeq 2000
Sequencing date	2010	2009-2012	2010-2011	2014

We obtained the *P. falciparum* reference sequence (Gardner et al., 2002) obtained from PlasmoDB (Aurrecoechea et al., 2009) (build 32.0, April 2017). For reference-based analyses, we aligned all reads with BWA-MEM to this reference using default settings. We marked duplicate reads using the `MarkDuplicates` tool in the Picard suite, and applied the GATK base quality score recalibrator (BQSR) to improve the accuracy of the base quality scores (DePristo et al., 2011). We computed coverage metrics with the GATK's `DepthOfCoverage` tool, and insert size metrics with Picard's `CollectInsertSizeMetrics` tool[2]. A summary of the sample data is presented in Table S4.

S3.4 Reference-based variant calls

For later comparison to our de Bruijn graph-based DNM calls, we applied two software packages to the 20 samples in the 3D7xHB3 cross. For SNVs and small indels, we applied the GATK's

[2] http://broadinstitute.github.io/picard/

HaplotypeCaller (Poplin et al., 2017), specifying a ploidy of 1 and leaving all other parameters as software defaults. For larger and more complex structural variants (insertions, deletions, tandem duplications, inversions, and translocations), we applied the Delly2 software (Rausch et al., 2012). No additional filtering was applied to these callsets in order to preserve maximum sensitivity for later comparison to our *k*-mer-counting analyses.

S3.5 *De novo* assembly

We constructed multi-color linked de Bruijn graphs (LdBGs) for all 127 samples using McCortex (Turner et al., 2018) with a four-step pipeline. First, to build raw (un-error-cleaned) graphs, we applied the `build` command with a *k*-mer size parameter of 47 bp for all samples. Next, we ran the `clean` command with default settings to produce error-cleaned graphs. Substantial coverage variation in the *P. falciparum* samples caused the default error-cleaning process to erroneously remove true genomic *k*-mers. Unaddressed, this lead to gapped assemblies with unusually short contigs, as *k*-mers necessary for haplotype threading were not present in the graph. We implemented a repair module (`RecoverExcludedKmers`) in Corticall, recovering *k*-mers that were present in the raw graph, absent in the clean graph, and present in the canonical reference or either parental PacBio draft assembly for the cross. We ran the McCortex `inferedges` tool on the error-recovered graph, ensuring adjacencies between records in the graph were properly recorded. Finally, the parents, children, PacBio draft references, and reference genome were joined into a single multi-color graph to facilitate downstream processing using the Corticall `Join` command (a low-memory equivalent to McCortex's `join` command). All graphs were output in sorted order to facilitate random access over the dataset in Corticall.

We constructed three link annotation sets for each sample graph using the McCortex `thread` command: one for each parental PacBio draft and one for a sample's paired end reads. For PacBio assemblies and reads, we constructed links using the two-way gap-filling option. For paired-end reads, we further improved these links by re-threading links using `thread`'s paired-end mode and minimum (maximum) fragment size parameters 0 bp (1000 bp). We then filtered out low-coverage links likely arising from sequencing errors according to McCortex's documentation.

S3.6 Validation of DNM calling strategy

We generated DNM calls for all progeny based on the procedure detailed in S4 and performed in silico validation according to the detail in S5. We sought to verify our effectiveness in real data as well. We first verified our filtration strategy by comparing the number of novel *k*-mers found in the progeny 803xGB4 sequenced on both PacBio RSII and Illumina instruments (isolate 36F11, corresponding to Illumina sample PG0446-C). For the PacBio assembly, we considered a *k*-mer novel if it was present in the PacBio assembly and the cleaned Illumina data with coverage greater than $6x$. We computed the overlap between this set and the novel *k*-mers determined by the aforementioned filters. The results are presented in Table S5. All novel *k*-mers present in the PacBio assembly are recapitulated by the Illumina analysis. The Illumina sample also contains 53 additional novel *k*-mers not seen in the PacBio assembly.

We then manually validated the five resulting DNM calls in the 36F11 progeny sample by aligning short reads with BWA-MEM and the PacBio draft assembly with `minimap2` (Li, 2018) to both our 803 and GB4 draft references. We visually inspected each variant in IGV (Thorvaldsdóttir et al., 2013), adjudicating each call based on whether evidence for the variant was

Table S5. Comparison of novel *k*-mers between PacBio and Illumina sequencing for 36F11/PG0446-C

PacBio ID	Illumina ID	Novel *k*-mers unique to PacBio	Novel *k*-mers overlap	Novel *k*-mers unique to Illumina
36F11	PG0446-C	0	48	53

(1) present in the short read data, (2) present in the long read assembly, and (3) absent in the draft reference. These results are summarized in Table S6. Three variants had clear evidence in both the short read data and long read assembly and were considered validated. While variant 3 had inconsistent evidence between the short and long read data, its validation status could not be confidently adjudicated. This particular variant allele represents a single-base expansion of a nine-base homopolymer expansion, a common error mode for both short and long read sequencing instruments. Finally, variant 5 had substantial evidence in the short read data but was absent from the long read assembly. This comports with our finding above regarding novel *k*-mers not present in the PacBio assembly. We hypothesize that this represents a polymorphic site that can be found in the Illumina data but was likely have been removed during the PacBio haploid assembly process.

Table S6. Manual validation of *de novo* mutations in 803xGB4 progeny 36F11.

id	chromosome	start	ref allele	alt allele	validates
1	PfGB4_NIH_CSHL_08	1535876	C	G	yes
2	PfGB4_NIH_CSHL_14	288542	A	AA	uncertain (homopolymer)
3	PfGB4_NIH_CSHL_14	616500	TATATATATA	TT	yes
4	Pf803_NIH_CSHL_13	1971356	A	G	yes
5	Pf803_NIH_CSHL_14	1389386	C	T	no (polymorphic)

S4 Further details of the Tesserae model

S4.1 Transition and emission matrices

The Tesserae pair-HMM is specified by a transition matrix and emission matrix, provided below. Terms are defined as follows:

Table S7. Definitions and defaults in the Tesserae pair-HMM.

Term	Definition	Default
δ	probability of indel initiation	0.025
ϵ	probability of indel extension	0.75
ρ	probability of recombination	0.0001
π_M	probability of starting in match (M) state	0.75
π_I	probability of starting in insert (I) state	$1 - \pi_M$
τ	probability of termination	0.001
$\|Y\|$	$\Sigma_{k=1}^{n} l_k$ where l_k is the length of sequence k	-

The transition matrix is given by:

	B	M_x	I_x	D_x	M_k	I_k	D_k	T
B	0	$\frac{\pi_M}{\|Y\|}$	$\frac{\pi_I}{\|Y\|}$	0	$\frac{\pi_M}{\|Y\|}$	$\frac{\pi_I}{\|Y\|}$	0	0
M_x	0	$1-2\delta-\rho-\tau$	δ	δ	$\frac{\rho\pi_M}{\|Y\|}$	$\frac{\rho\pi_I}{\|Y\|}$	0	τ
I_x	0	$1-\epsilon-\rho-\tau$	ϵ	0	$\frac{\rho\pi_M}{\|Y\|}$	$\frac{\rho\pi_I}{\|Y\|}$	0	τ
D_x	0	$1-\epsilon$	0	ϵ	0	0	0	0
M_k	0	$\frac{\rho\pi_M}{\|Y\|}$	$\frac{\rho\pi_I}{\|Y\|}$	0	$1-2\delta-\rho-\tau$	δ	δ	τ
I_k	0	$\frac{\rho\pi_M}{\|Y\|}$	$\frac{\rho\pi_I}{\|Y\|}$	0	$1-\epsilon-\rho-\tau$	ϵ	0	τ
D_k	0	0	0	0	$1-\epsilon$	0	ϵ	0
T	0	0	0	0	0	0	0	1

The emission probability of a nucleotide emitted from an insertion state is $e(x_i) = 0.2$, while the nucleotide emission matrix $e(x_i, y_j)$ (where x_i is the nucleotide at site i in sequence x and y_j is the nucleotide at site j in sequence y) is given by:

$$e(x_i, y_j) = \begin{cases} 0.9 & x_i = y_j \\ 0.05 & \text{if nucleotide transition} \\ 0.025 & \text{if nucleotide transversion} \end{cases}.$$

The trellis diagram for the model is provided in Figure S3.

S4.2 Variant decoding

By keeping pointers through the recursion process of the Viterbi algorithm, we trace back the most probable path through h. This path through the resulting traceback matrix can be interpreted as specifying the background haplotype for subsequences and variants in the query

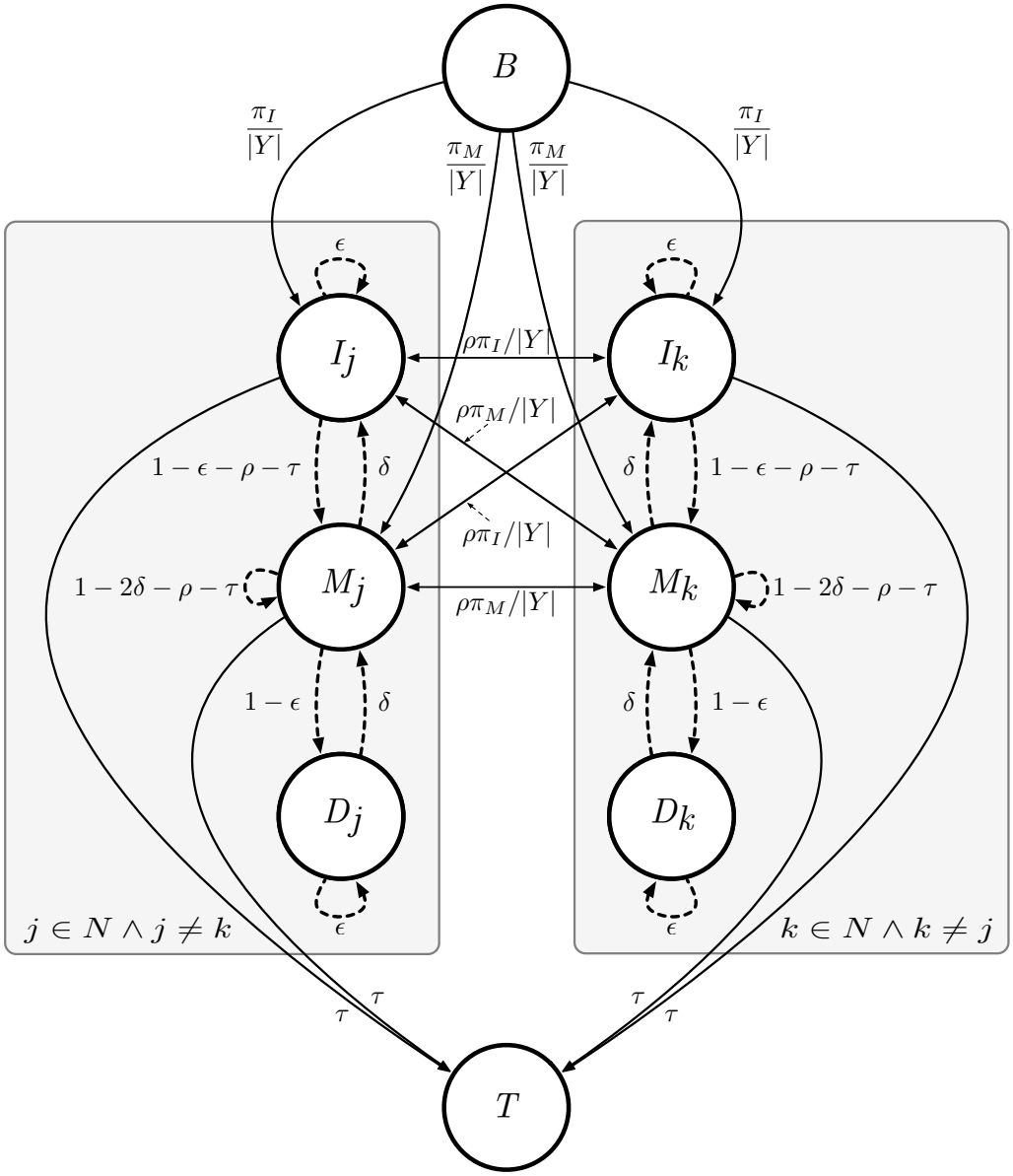

Figure S3. The Tesserae model trellis diagram for simultaneous alignment with recombination. Each plate denotes a global alignment model for input query and source sequences. Match, insert, and delete states are connected within single sequences. To permit recombination, all sequences are exhaustively connected through match and insert (but not delete) states. Dashed lines indicate internal (intra-plate) transitions. Solid lines indicate external (inter-plate) transitions.

sequence against this background. Figure S4 depicts the overall process on a toy example consisting of a short query sequence and two candidate source sequences. The query sequence is constructed by initiating assembly on the LdBG in the query color at one or more novel *k*-mers. The two candidate source haplotypes are constructed by initiating assembly at non-novel *k*-mers found in the query sequence. Redundant contigs are removed (this step is not shown). The sequence set is processed with the Tesserae model, and decoding of the resulting traceback matrix reveals two recombination breakends and an insertion.

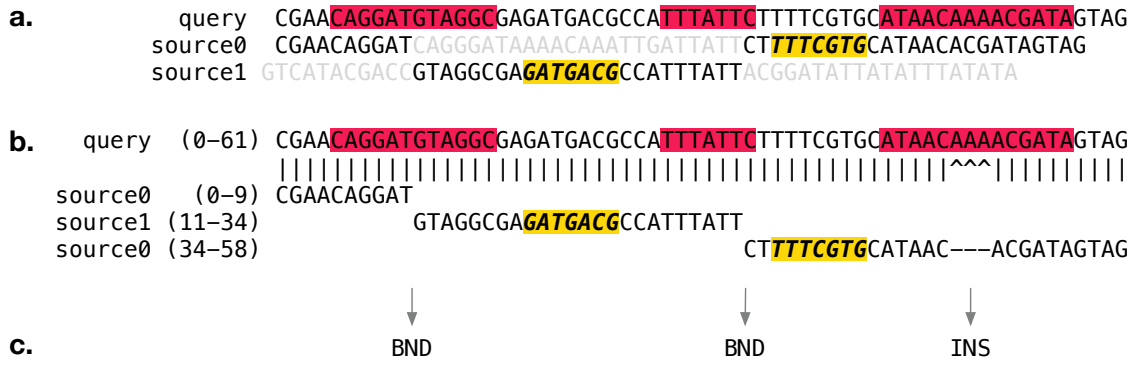

Figure S4. Assembly, alignment, and variant decoding process for a toy example. (**a**) A 61-bp query sequence with three regions of sequence novelty (nucleotides with a red background). To assemble candidate source sequences, *k*-mers in the non-novel regions are used to seed contig traversals on the LdBG in the source colors (in this example, $k = 7$). Sequences with yellow backgrounds indicate the seed *k*-mers for these traversals. For visual clarity, sequences shown here are positioned manually, and source subsequences that are not present in the query sequence are shown in grey. (**b**) All sequences in the sequence set aligned with the Tesserae model, with breakends depicted as linebreaks and variants shown on their haplotypic background. (**c**) Putative mutations spanned by novel *k*-mers are emitted as breakend ("BND") variant calls.

S5 Simulation of progeny genomes

We simulated progeny genomes from a synthetic crossing of HB3 and DD2 parasites with known allelic recombinations and DNMs using the Corticall module `SimulateHaploidChild`. Our simulations output a VCF describing recombination operations and other permutations to the appropriate reference sequences. The newly generated progeny reference genomes were then used to simulate reads for *de novo* assembly and DNM calling.

S5.1 Homologous recombination

We began with the pseudochromosome contiguated assemblies of HB3 and DD2 from section S2.5, for which homologous chromosomes suitable for recombination were easily paired. Crossover number c was taken to be Poisson-distributed, parameterized with values as determined in Miles *et al.* 2016 (Miles et al., 2016) and shown in Equation 1 with rate parameter $\mu = \lambda d$, average crossover rate $\lambda = 0.0135\ cM^{-1}$ and map length $d(\ell) = 0.74\ell - 0.11$, where ℓ is chromosome length. Map length and crossover probability distributions are shown in Figure S5.

$$P\{c|\ell\} = \frac{\mu^c}{c!}e^{-\mu} = \frac{(\lambda d(\ell))^c}{c!}e^{-\lambda d(\ell)} \tag{1}$$

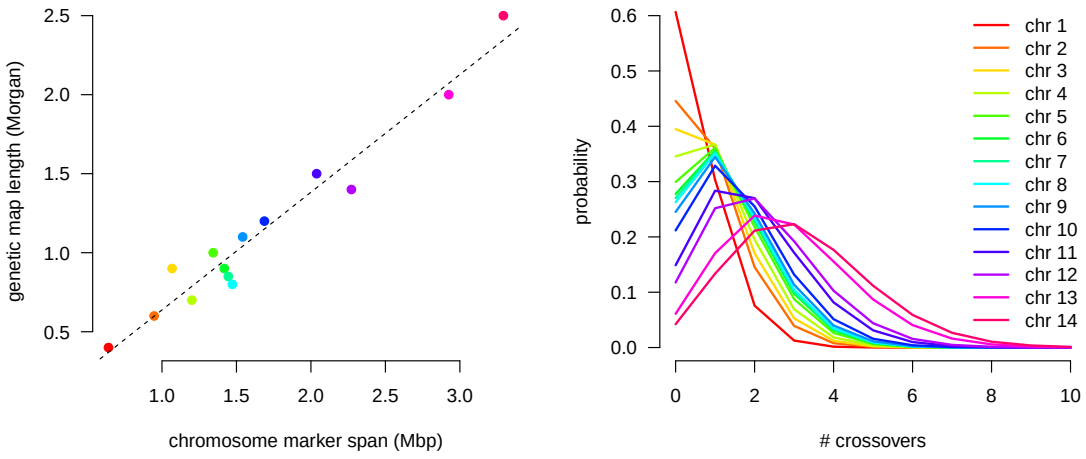

Figure S5. Simulated map lengths per chromosome (left) and crossover probability distribution per chromosome (right).

S5.2 Non-homologous allelic recombination

We simulated non-homologous allelic recombination (NAHR) events between members of the *var* gene family annotated in the HB3 and DD2 draft genome assemblies. While true NAHR events appear to be mediated by microhomology and possibly 5′ upstream promoter sequence, we deemed random pairings sufficient for our needs. For each simulated progeny, we chose at

random two *var* genes from each genome, computed exact homology maps at $k = 21$ between the two sequences, and simulated a random number (between 1 and 6) of crossover events between the two sequences constrained to the homologous regions. We emitted two VCF entries for each NAHR event, the first replacing one *var* sequence with our new sequence, the other removing the second *var* gene from the progeny reference.

S5.3 Other variants

We simulated SNVs, short indels and inversions, multi-nucleotide variants. For the non-point mutations, we simulated a range of allele sizes up to 1,000 bp. To evaluate our capabilities in expansions or contractions of repetitive sequence, we scanned the parental reference assemblies for series of repeating sequence units up to length 6 bp, adding or removing units up to the maximum number of units present in the repeat. All events were added to the VCF used for draft reference sequence permutation.

S5.4 Read simulation

Reads were simulated using 76 bp reads, insert size of 250 bp, a mean read depth of $100x$, and per-base error rate of 0.5%. All processing steps proceeded identically to the procedure specified in section S3.5.

S5.5 Variant detection with other algorithms

To facilitate comparison of our mutation calls to those of other algorithms, we applied Delly (SVs, Rausch et al. (2012)), GRIDSS (SVs, Cameron et al. (2017)), GATK's HaplotypeCaller (SNVs and small indels, Poplin et al. (2017)), and Manta (SVs, Chen et al. (2016)) to our simulated genomes.

While Corticall is designed specifically to operate using multiple reference sequences, the remaining reference-based callers were not. Initial testing on simulated reads aligned to the canonical 3D7 reference revealed extremely low sensitivity, suggesting that attempting to characterize variation in simulated HB3 and DD2 haplotypes absent from the 3D7 reference would not provide an effective demonstration of the reference-based algorithms' capabilities. To compensate for this issue, we developed a procedure to identify DNMs in the child by aligning reads and calling variants against each parental reference sequence in succession. We simulated perfect reads (76 bp paired-end, 250 bp insert, 100× coverage, zero errors) for both parents. Parental and progeny read sets were aligned twice: first to our HB3 draft reference sequence and then to our DD2 draft.

To select candidate DNMs, we identified variants in both parents and progeny using each of the aformentioned callers and excluded variants in the child proximal to a variant in the parent. We merged each progeny's callset into a single file. To mitigate double-counting resulting from the identification of the same variant in the child against two different reference sequences, we excluded variants found in syntenic regions of the parental reference genomes by permuting a 200 bp window of the draft reference using each detected variant, aligning the sequence to the counterpart parental reference, and rejecting the variant if the permuted sequence (1) aligned to the expected chromosome, (2) aligned with mapping quality > 10, and (3) matched the reference with no errors. This filter was useful in reducing redundancy in core regions of the genome, but had little to no effect in non-core regions.

Upon inspect of the callsets, additional filtering was applied based on read depth ($> 80\times$, $< 200\times$), mapping quality (> 9.0), Fisher's exact test for strand bias (< 1.0), strand odds ratio (< 3.0).

S5.6 Evaluation

To evaluate performance of our and other mutation callers, and to handle variant representational differences (e.g. quantify recovered *de novo* mutations that are identified on the wrong parental background), we developed the `CompareCalls` module in Corticall. For "concrete" variants (SNVs and indels with precise position and allele information), we required the simulated and recalled variant types to be identical and computed the reciprocal overlap of simulated versus variant haplotypes in a 200 bp window centered on the variant, requiring at least an 80% overlap. For symbolic variants (large SVs and translocations), we required recalled events to be discovered on the correct parental background and no more than $5,000$ bp away from the expected position. We also developed a "lenient" evaluation strategy wherein the variant types were not required to be identical, the incorrect haplotypic background was permitted, and the variant positioning was relaxed to within $50,000$ bp. We report both the strict and lenient F_1 score for DNMs.

References

Altschul SF, Gish W, Miller W, Myers EW, and Lipman DJ. 1990. Basic local alignment search tool. *Journal of molecular biology* **215**: 403–410.

Aurrecoechea C, Brestelli J, Brunk BP, Dommer J, Fischer S, Gajria B, Gao X, Gingle A, Grant G, Harb OS, et al.. 2009. PlasmoDB: a functional genomic database for malaria parasites. *Nucleic acids research* **37**: D539–D543.

Cameron DL, Schröder J, Penington JS, Do H, Molania R, Dobrovic A, Speed TP, and Papenfuss AT. 2017. GRIDSS: sensitive and specific genomic rearrangement detection using positional de Bruijn graph assembly. *Genome Research* **27**: 2050–2060.

Chen X, Schulz-Trieglaff O, Shaw R, Barnes B, Schlesinger F, Källberg M, Cox AJ, Kruglyak S, and Saunders CT. 2016. Manta: rapid detection of structural variants and indels for germline and cancer sequencing applications. *Bioinformatics* **32**: 1220–1222.

DePristo MA, Banks E, Poplin R, Garimella KV, Maguire JR, Hartl C, Philippakis AA, del Angel G, Rivas MA, Hanna M, et al.. 2011. A framework for variation discovery and genotyping using next-generation DNA sequencing data. *Nature genetics* **43**: 491–498.

Gardner MJ, Hall N, Fung E, White O, Berriman M, Hyman RW, Carlton JM, Pain A, Nelson KE, Bowman S, et al.. 2002. Genome sequence of the human malaria parasite Plasmodium falciparum. *Nature* **419**: 498–511.

Gibbs AJ and Mcintyre GA. 1970. The Diagram, a Method for Comparing Sequences. *European Journal of Biochemistry* **16**: 1–11.

Hayton K, Gaur D, Liu A, Takahashi J, Henschen B, Singh S, Lambert L, Furuya T, Bouttenot R, Doll M, et al.. 2008. Erythrocyte Binding Protein PfRH5 Polymorphisms Determine Species-Specific Pathways of Plasmodium falciparum Invasion. *Cell Host & Microbe* **4**: 40–51.

Kozarewa I, Ning Z, Quail MA, Sanders MJ, Berriman M, and Turner DJ. 2009. Amplification-free Illumina sequencing-library preparation facilitates improved mapping and assembly of (G+C)-biased genomes. *Nature methods* **6**: 291–295.

Kurtz S, Phillippy A, Delcher AL, Smoot M, Shumway M, Antonescu C, and Salzberg SL. 2004. Versatile and open software for comparing large genomes. *Genome Biology* **5**: R12.

Li H. 2013. Aligning sequence reads, clone sequences and assembly contigs with BWA-MEM. *arXiv.org* p. arXiv:1303.3997.

Li H. 2018. Minimap2: pairwise alignment for nucleotide sequences. *Bioinformatics* **34**: 3094–3100.

Miles A, Iqbal Z, Vauterin P, Pearson R, Campino S, Theron M, Gould K, Mead D, Drury E, O'Brien J, et al.. 2016. Indels, structural variation, and recombination drive genomic diversity in Plasmodium falciparum. *Genome Research* **26**: 1288–1299.

Otto TD, Böhme U, Sanders M, Reid A, Bruske EI, Duffy CW, Bull PC, Pearson RD, Abdi A, Dimonte S, et al.. 2018. Long read assemblies of geographically dispersed Plasmodium falciparum isolates reveal highly structured subtelomeres. *Wellcome Open Research* **3**: 52.

Otto TD, Dillon GP, Degrave WS, and Berriman M. 2011. RATT: Rapid Annotation Transfer Tool. *Nucleic acids research* **39**: e57–e57.

Ozer EA, Allen JP, and Hauser AR. 2014. Characterization of the core and accessory genomes of Pseudomonas aeruginosa using bioinformatic tools Spine and AGEnt. *BMC genomics* **15**: 737.

Poplin R, Ruano Rubio V, DePristo MA, Fennell TJ, Carneiro MO, Van der Auwera GA, Kling DE, Gauthier LD, Levy-Moonshine A, Roazen D, et al.. 2017. Scaling accurate genetic variant discovery to tens of thousands of samples. *bioRxiv* p. 201178.

Rausch T, Zichner T, Schlattl A, Stütz AM, Benes V, and Korbel JO. 2012. DELLY: structural variant discovery by integrated paired-end and split-read analysis. *Bioinformatics* **28**: i333–i339.

Sá JM, Kaslow SR, Krause MA, Melendez-Muniz VA, Salzman RE, Kite WA, Zhang M, Moraes Barros RR, Barros RRM, Mu J, et al.. 2018. Artemisinin resistance phenotypes and K13 inheritance in a Plasmodium falciparum cross and Aotus model . *Proceedings of the National Academy of Sciences* **2**: 201813386–6.

Smit A, Hubley R, and Green P. 2013. *RepeatMasker Open-4.0*.

Steinbiss S, Silva-Franco F, Brunk B, Foth B, Hertz-Fowler C, Berriman M, and Otto TD. 2016. Companion: a web server for annotation and analysis of parasite genomes. *Nucleic acids research* **44**: W29–W34.

Thorvaldsdóttir H, Robinson JT, and Mesirov JP. 2013. Integrative Genomics Viewer (IGV): high-performance genomics data visualization and exploration. *Briefings in bioinformatics* **14**: 178–192.

Trager W and Jensen JB. 1976. Human malaria parasites in continuous culture. *Science* **193**: 673–675.

Turner I, Garimella KV, Iqbal Z, and McVean G. 2018. Integrating long-range connectivity information into de Bruijn graphs. *Bioinformatics* **34**: 2556–2565.

Walliker D, Quakyi I, Wellems T, McCutchan T, Szarfman A, London W, Corcoran L, Burkot T, and Carter R. 1987. Genetic analysis of the human malaria parasite Plasmodium falciparum. *Science* **236**: 1661–1666.

Wellems TE, Panton LJ, Gluzman IY, do Rosario VE, Gwadz RW, Walker-Jonah A, and Krogstad DJ. 1990. Chloroquine resistance not linked to mdr-like genes in a Plasmodium falciparum cross. *Nature* **345**: 253–255.