Supplementary Materials

Batch effects in population genomic studies with low-coverage whole genome sequencing data: causes, detection, and mitigation

R. Nicolas Lou[1] and Nina O. Therkildsen[1]

1. Department of Natural Resources, Cornell University, Ithaca, New York, USA

Supplementary Methods

"Batch-effect-naive" bioinformatic pipeline

To convert raw fastq files into bam format, we first trimmed adapters from sequencing reads using Trimmomatic-0.39 (NextSeq-150PE: `PE -phred33 'ILLUMINACLIP:'$ADAPTERS':2:30:10:1:true'`, HiSeq-125SE: `SE -phred33 'ILLUMINACLIP:'$ADAPTERS':2:30:10'`). We then used fastp-0.19.7 to trim poly-G tails with the NextSeq-150PE batch of data only (`--trim_poly_g -Q -L -A`), with the default setting on minimum poly-G length threshold (`--poly_g_min_len 10`). We mapped reads to the gadMor3 reference genome using bowtie2-2.3.5.1 (`-q --phred33 -- very-sensitive -I 0 -X 1500 –fr` for NextSeq-150PE and `-q --phred33 -- very-sensitive` for HiSeq-125SE). We used samtools-1.11 to convert the resulting sam files to bam format and sorted them (`view -buS` and `sort`), Picard tools-2.9.0 to remove duplicated reads (`MarkDuplicates VALIDATION_STRINGENCY=SILENT REMOVE_DUPLICATES=true`), and BamUtil-1.0.14 to clip overlapping read pairs (`clipOverlap`) with the NextSeq-150PE batch of data only. Lastly, we performed indel realignment with GATK-3.7 (`-T RealignerTargetCreator` followed by `-T IndelRealigner --consensusDeterminationModel USE_READ`, with default options). Lastly, we counted the number of bases with mapping quality higher than 20 in the indel-realigned bam files using Samtools, and calculated per-sample sequencing depth (Table 1, Figure S1).

To estimate individual heterozygosity, we first estimated sample allele frequency (SAF) likelihoods with ANGSD-0.931 across the entire genome (including the non-variable sites) for each of the 163 samples included in this paper (`-doSaf 1 -GL 1 -doCounts 1 -setMinDepth 2 -setMaxDepth 10 -minQ 20 -minmapq 30`). We then used the realSFS module in ANGSD-0.931 to estimate genome-wide site frequency spectrum (SFS) for each individual, from which individual heterozygosity can be calculated (Figure 1A "before").

Presence/absence of poly-G tails

Instead of only applying the poly-G tail trimming functionality in fastp-0.19.7 as we did in the "batch-effect-naïve" pipeline, we also used the sliding window quality trimming

functionality in fastp-0.19.7 (`--cut_right --trim_poly_g -L -A`) to further eliminate poly-G tails in the adapter trimmed fastq files in the NextSeq-150PE batch. Default window length (`--cut_right_window_size 4`) and mean base quality threshold (`--cut_right_mean_quality 20`) were used. We randomly selected a single read with poly-G tails to demonstrate the effectiveness of poly-G tail removal with and without the sliding window quality trimming functionality (Figure 2A). In addition, we randomly selected three samples, and used FastQC-0.11.8 to calculate the base composition of each read position for each individual, before poly-G removal, after poly-G trimming, and after poly-G trimming + sliding window quality trimming (Figure 2B). These FastQC results were then summarized and visualized using custom R scripts: https://github.com/therkildsen-lab/batch-effect/blob/main/markdown/polyg.md We also used FastQC-0.11.8 to calculate the base composition of each read position after poly-G trimming, read alignment, and quality control in order to demonstrate the poly-G tails can persist after read alignment (Figure S2)

As we found that poly-G tails cannot be completely removed with the poly-G tail trimming functionality fastp-0.19.7 alone, we applied sliding window quality trimming to all of our adapter trimmed fastq files in the NextSeq-150PE batch, remapped them to the reference genome, and repeated our deduplication, overlap clipping, and indel-realignment procedure. All following analyses are based on these files from which poly-G tails are removed. We also estimated heterozygosity from these bam files without correction for other causes of batch effect in order to demonstrate the strong impact poly-G tail has on heterozygosity estimation (Figure 1A "before" vs. Figure S3).

After the poly-G issue is resolved, we used ANGSD to identify SNPs in the data (`-GL 1 -doGlf 2 -doMaf 1 -doMajorMinor 1 -doCounts 1 -doDepth 1 -dumpCounts 1 -doIBS 1 -makematrix 1 -doCov 1 -P 32 -SNP_pval 1e-6 -setMinDepth 46 -setMaxDepth 184 -minInd 20 -minQ 20 -minMaf 0.05 -minMapQ 20`). This generated a total of 5,204,764 SNPs.

We then estimated the sample allele frequency (SAF) likelihoods and the minor allele frequencies (MAF) in each batch of data (pooling all populations together) at this set of SNPs (`-dosaf 1 -GL 1 -doGlf 2 -doMaf 1 -doMajorMinor 3 -doCounts 1 -doDepth 1 -setMinDepth 20 -setMaxDepth 184 -minInd 20 -minQ 20 -minMapQ 20 -sites $SNP_LIST`). The MAFs estimated in this step were used later to extract a list of private alleles in each batch of data (alleles with frequencies between 10% and 90% in one batch but smaller than 1% or larger than 99% in the other batch) (custom R script: https://github.com/therkildsen-lab/batch-effect/blob/main/markdown/degradation.md#extract-private-alleles-and-examine-proportion-of-base-substitutions). This step also generates the read count at each SNP location with a minimum mapping quality of 20. To generate the read count without a mapping quality filter, we used `-doCounts 1 -doDepth 1 -dumpCounts 1 -setMinDepth 2 -minInd 2 -minQ 20`. These read counts were later used to identify region affect by reference bias (Figure 4). Lastly, using the SAF likelihoods as input, we ran the realSFS module in ANGSD to estimate a genome-wide two-dimensional SFS, which we used as a prior to estimate per-SNP F_{ST} between the two batches (Figure 1C "before").

To perform PCA, we first need a list of LD-pruned SNPs. Due to computational limitations in LD estimation, we kept 1 SNP in every 5 SNPs in our SNP list, and also filtered out the SNPs within four large inversions known to be polymorphic in Atlantic cod. We then used ngsLD-1.1.0 to estimate pairwise LD between SNPs from genotype likelihoods

(`--n_ind 163 --n_sites 944554 --probs --rnd_sample 1 --max_kb_dist 10`) and to perform LD pruning (`--max_kb_dist 10 --min_weight 0.5`). After LD pruning, we obtained a "batch-effect-naïve" SNP list with 715,468 unlinked SNPs. With this set of unlinked SNPs, we performed PCA using ANGSD (`-GL 1 -doGlf 2 -doMaf 1 -doMajorMinor 3 -doCounts 1 -doDepth 1 -dumpCounts 1 -setMinDepth 2 -setMaxDepth 661 -minInd 2 -minQ 20 -minMapQ 20 -minMaf 0.05 -doIBS 2 -makematrix 1 -doCov 1 -sites $LD_PRUNED_SNP_LIST`). Eigendecomposition was then performed on the resulting covariance matrix using custom R scripts (https://github.com/therkildsen-lab/batch-effect/blob/main/markdown/figures.md) (Figure 1B "before").

Difference in levels of base quality score miscalibration

To detect base quality score miscalibration in the data, we estimated heterozygosity using ANGSD-0.931 from the poly-G -free bam files (`-doSaf 1 -GL 1 -doCounts 1 -setMinDepth 2 -setMaxDepth 10 -minmapq 30`) with either a relaxed (`-minQ 20`, Figure S3) or a stringent base quality filter (`-minQ 33`, Figure 1A "after"). If base quality is not biased, we expect to see no systematic differences in heterozygosity estimates between the two base quality filter settings. We visualized the change in heterozygosity estimates after applying the more stringent base quality filter in both batches, and used the paired samples t-test to evaluate whether this change is significantly different from zero in either batch (Figure 3), using custom R scripts (https://github.com/therkildsen-lab/batch-effect/blob/main/markdown/base_quality.md#figure-3). To mitigate the batch effects caused by base quality score miscalibration in individual heterozygosity (Figure S3), we used the estimates generated with a more stringent base quality filter, resulting in Figure 1A "after" and Figure 5A "before".

Difference in levels of reference bias / alignment error

To detect reference bias / alignment error in the data, we first used the Integrative Genomics Viewer to spot check read alignment at randomly selected Fst outliers between the two batches of data (Figure 1C), either with or without a minimum mapping quality filter of 20. The position LG23:6170006 is shown in Figure 4A, but other outlier loci exhibited similar patterns (e.g., LG07:9272785, LG16:25656225, LG17:19229736) where a position appears to be polymorphic in the HiSeq-125SE batch without the mapping quality filter but appears to be fixed for the reference allele when such filter is applied.

For each SNP, we then calculated the proportion of reads with mapping quality lower than 20, and tested the difference in this value between F_{ST} outliers and all other SNPs (two-sample t-test, Figure 4B).

To mitigate the batch effects caused by reference bias in F_{ST} (Figure 1C "before"), we filtered out all SNPs with more than 10% of reads having mapping quality scores lower than 20, resulting in Figure 1C "after". To mitigate the batch effects caused by reference bias in PCA, we filtered out all private alleles in either batch of the data, and performed PCA with the same setting as Figure 1B "before", resulting in Figure 1B "after" and Figure 5B "after" (see https://github.com/therkildsen-lab/batch-effect/blob/main/markdown/degradation.md#come-up-with-a-new-snp-list). Lastly, we also

ran PCA after filtering out the private SNPs in either one of the two batches to illustrate the extent of ascertainment bias this approach causes (Figure S4).

Difference in levels of DNA degradation

To detect DNA degradation in the data, we first estimated heterozygosity using ANGSD from the poly-G -free bam files (`-doSaf 1 -GL 1 -doCounts 1 -setMinDepth 2 -setMaxDepth 10 -minmapq 30`) with a stringent base quality filter (`-minQ 33`) to control for the base quality bias issue, and either including (`-noTrans 0`) or excluding transitions (`-noTrans 1`) in the data. If all samples are similarly degraded, we expect to see no systematic differences in the change of heterozygosity estimates after excluding the transitions. We visualized the change in heterozygosity estimates after excluding the transitions in well-preserved samples in both batches and degraded samples (as identified with gel electrophoresis) in the HiSeq-125SE batch, and used the ANOVA test to evaluate whether this change is significantly different between the three groups of samples (Figure 5D), using custom R scripts: https://github.com/therkildsen-lab/batch-effect/blob/main/markdown/degradation.md#estimate-heterozygositing-without-transitions).

In addition, we used custom R scripts to visualize the proportion of different base substitutions in private alleles in each batch: https://github.com/therkildsen-lab/batch-effect/blob/main/markdown/sfs.md (Figure 5C). We expect to see an enrichment of C-to-T and G-to-A substitutions in the batch of data that is has degraded samples (i.e. HiSeq-125SE).

To mitigate the batch effects caused by DNA degradation in individual heterozygosity (Figure 5A "before", we used the heterozygosity estimates generated excluding transitions , resulting in Figure 5A "after". To mitigate the batch effects caused by DNA degradation (and reference bias) in PCA (Figure 5B "before"), we filtered out all private alleles in either batch of the data before performing PCA, resulting in Figure 5B "after".

Difference in sequencing depth

To demonstrate that difference in sequencing depth can cause batch effect, we first used simulated data. The simulation pipeline is based on the ones that were used for Lou et al. (in revision), and associated scripts are available on GitHub: https://github.com/therkildsen-lab/lcwgs-simulation). In SLiM3, we randomly created a starting sequence on a 30Mbp chromosome, created nine populations, each with population size (N) of 500. These nine populations are distributed on a three-by-three grid, with a constant bidirectional migration rate (m) equal to 0.002 connecting each pair of adjacent populations. We scaled up the neutral mutation rate (μ) to 2x10-7 per bp per generation, and recombination rate (r) to 50cM/Mbp. We ran the simulation for 10,000 generations, resulting in a metapopulation that has achieved mutation-drift-migration equilibrium. This metapopulation consists of nine populations, each with population genetic parameters resembling a diploid animal population with effective population size (Ne) on the order of 10^4. We used ART-MountRainier to simulate the sequencing process, and subsampled the bam files to create different per-population sample sizes (5, 10, 20, 40, 60, 80). For each sample size, we gave half of the samples in each population a coverage of 0.125x, and gave the other half of samples a coverage of 4x. We called SNPs and estimated genotype likelihoods with the nine populations combined using -

`GL 1 -doGlf 2 -doMaf 1 -doMajorMinor 5 -doCounts 1 -doDepth 1 -dumpCounts 1 -doIBS 2 -makematrix 1 -doCov 1 -P 6 -SNP_pval 1e-6 -rmTriallelic 1e-6 -setMinDepth 2 -minInd 1 -minMaf 0.05 -minQ 20` in ANGSD-0.931. This step outputs a covariance matrix (`-doCov 1`) and a distance matrix (`-doIBS 2`) among individuals, and in addition to these, we also used PCAngsd-0.98 to generate another covariance matrix using the estimated genotype likelihoods from ANGSD (`-minMaf 0.05 -iter 200 -maf_iter 200`). Using the `eigen()` function and the `cmdscale()` function in R, we conducted principal component analysis (PCA) and principal coordinate analysis (PCoA) with these covariances matrices and distance matrix, respectively, and plotted the samples on the first two principal components / principal coordinates (Figure 6).

In addition, we evaluated the performance of PCAngsd-0.98 in comparison with that of ANGSD-0.931with our empirical data. Using the same SNP list with which Figure 1B "after" was generated (i.e., private alleles in both batches of data were filtered out), we ran PCAngsd-0.98 with the default setting (Figure S5). We found that although batch effect is not observed when ANGSD is used, the PCA generated by PCAngsd still exhibit batch effect after reference bias and DNA degradation are controlled for. Therefore, it is likely that this batch effect is caused by differences in sequencing coverage, to which PCAngsd is more susceptible.

Supplementary Figures

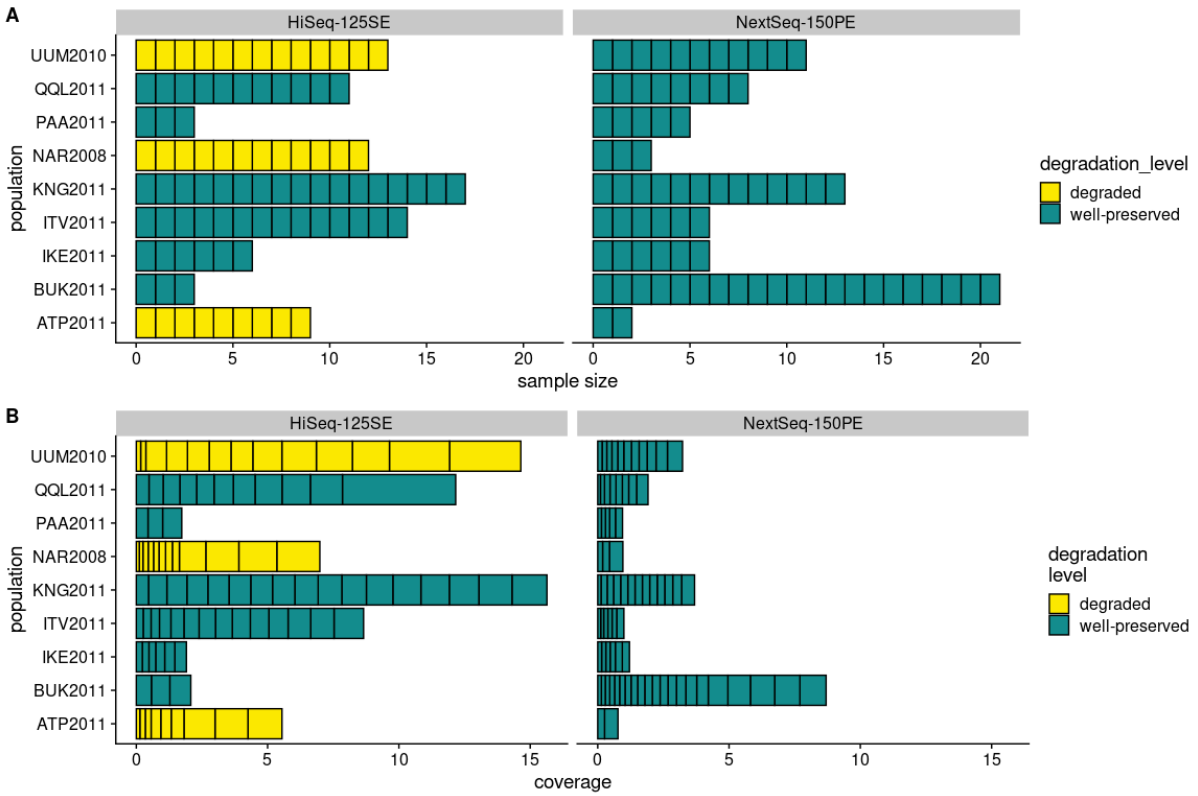

Figure S1. An overview of samples included in this study. **(A)** Sample size and **(B)** depth of coverage in different sequencing batches, grouped by population and colored by DNA degradation level. Note that we refer to ITV2011 as "pop 1", KNG2011 as "pop 2", QQL2011 as "pop 3" in Figure 1, S3, S4, S5. They are chosen for these plots because all their samples well-preserved (so that DNA degradation does not become a confounding factor), and because of their (relatively) larger sample size and higher coverage in both sequencing batches. Also note that we refer to ATP2011 as "pop 4", NAR2008 as "pop 5", UUM2010 as "pop 6" in Figure 5. These are the three populations for which samples are split to different batches based on their DNA degradation levels.

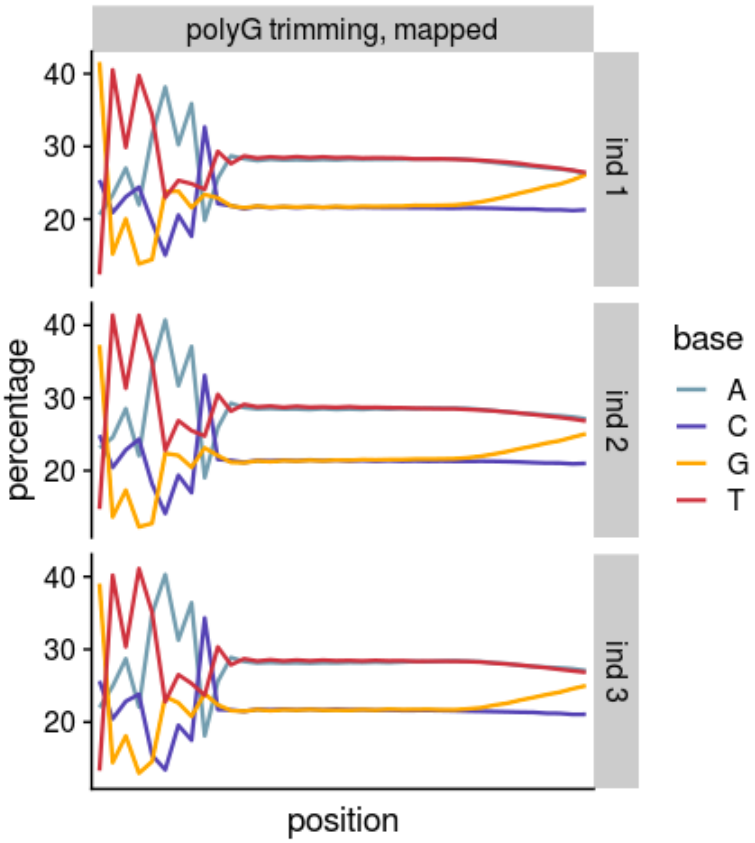

Figure S2. Persistence of poly-G tails after read alignment and quality check (i.e. minimum mapping quality filter of 20, deduplication, overlapping read end clipping) in three randomly chosen samples in the NextSeq-150PE batch if only poly-G trimming is performed.

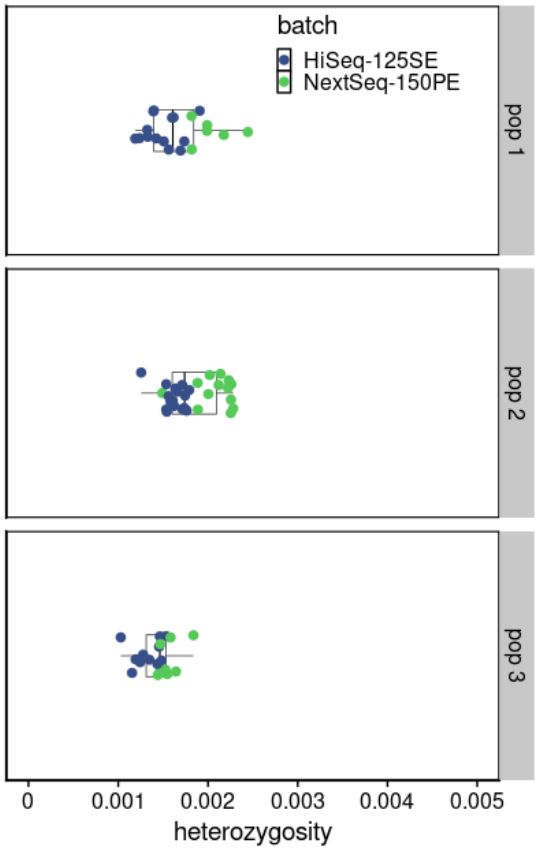

Figure S3. Heterozygosity estimates with sliding window trimming but not with stringent base quality filtering. Comparing this figure with Figure 1A, we conclude that poly-G tail is a more important factor than base quality score miscalibration in causing batch effects in heterozygosity estimation; nonetheless, sliding window trimming alone is not sufficient to resolve the issue, since batch effects are still strong in this figure, presumably caused by base quality score miscalibration.

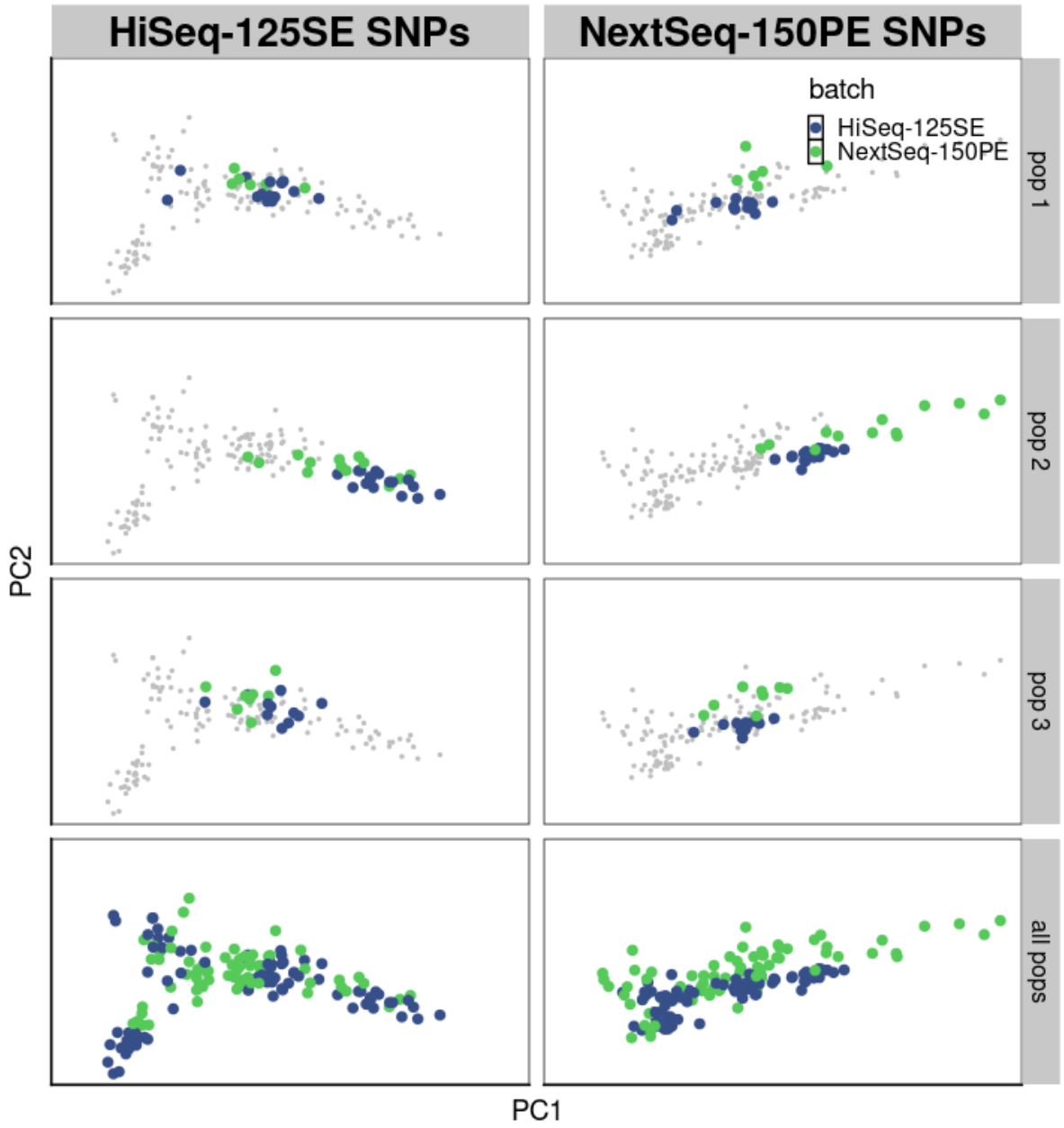

Figure S4. Calling SNPs with only one batch of data does not resolve batch effect but instead causes strong ascertainment bias. Samples from the batch with which SNPs are called appear at more extreme positions on a PCA plot due to the ascertainment bias.

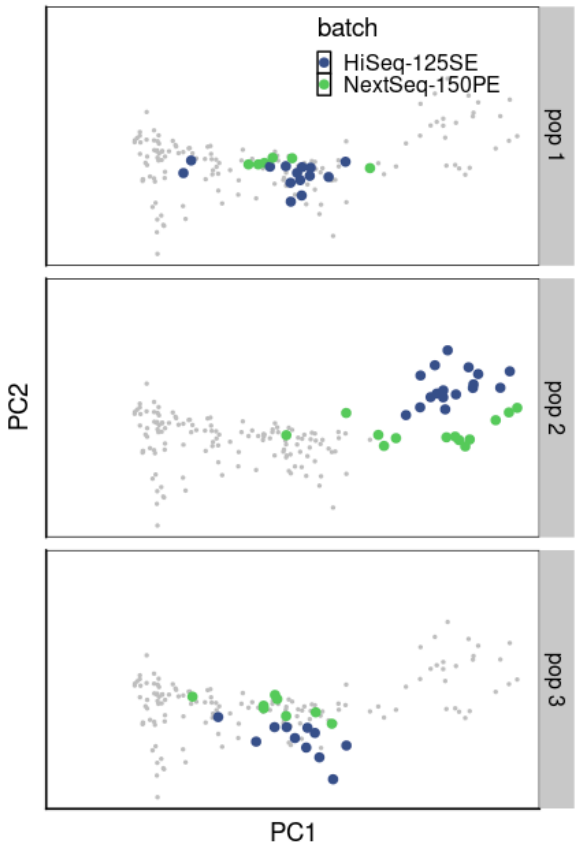

Figure S5. PCA result from PCAngsd after excluding SNPs that are invariable in one batch of samples but are at intermediate frequencies in the other batch, and excluding SNPs that have a high number of low-mapping-score reads mapped to them (i.e., same data as in Figure 1B "After", but PCAngsd was used to generate the covariance matrix for this figure instead of ANGSD). This might reflect the higher susceptibility of PCAngsd to batch effect caused by sequencing coverage difference.