PREPRINT · NOT PEER REVIEWED

Supplementary material

SwiftLD: an agent-built, byte-identical vcftools LD that is 826-fold faster than vcftools and 7.6-fold faster than PLINK 2

Joe Christensen (ORCID 0009-0007-0908-1299). Translational Neuropsychiatry Unit (TNU), Department of Clinical Medicine, Aarhus University, Denmark.

S1 Datasets

The synthetic sets are generated by gen_data.py from fixed seeds; the real regions are described in main-text Section 4.3 and in the data availability statement.

Table S1. The eleven benchmark sets. Sites are VCF records in the file, before vcftools' filters. The real regions of chromosome 22 (GRCh37) are 17.00–18.00 Mb (EUR), 18.00–18.25 Mb (all), 18.25–18.75 Mb (AFR), 19.00–19.50 Mb (EAS) and 19.50–19.80 Mb (SAS); every record in a region is kept, and the genotype sets had every | replaced by / to make unphased input for `--geno-r2`. The synthetic sets come from a haplotype mosaic (founder haplotypes with Beta(0.35, 0.35) allele frequencies, switching founder with probability $1 - \exp(-\rho \times \text{gap})$) with missing data, unphased calls, extra FORMAT fields, multi-allelic records, indels, several chromosomes and gzip input layered on top.

Set	Region / role	Samples	Sites	Options
r_eur_hap	EUR, development	503	30,104	`--hap-r2 --maf 0.05 --ld-window-bp 200000`
r_all_hap	all populations, development	2,504	8,348	`--hap-r2 --maf 0.05`
r_afr_geno	AFR, development	661	15,285	`--geno-r2 --maf 0.05 --ld-window 1000`
s_hap_full	synthetic, development	150	2,500	`--hap-r2`
s_geno_full	synthetic, development	250	2,500	`--geno-r2`
s_hap_multichr	synthetic, development	200	24,000	`--hap-r2 --ld-window-bp 150000`
rh_eas_hap	EAS, held out	504	15,268	`--hap-r2 --min-r2 0.1 --ld-window-bp 400000 --maf 0.02`
rh_sas_geno	SAS, held out	489	10,123	`--geno-r2 --ld-window-bp 300000 --maf 0.01`
h_many_samples	synthetic, held out	1,500	900	`--geno-r2`
h_hap_chr	synthetic, held out	120	9,000	`--hap-r2 --chr chr2 --ld-window 2000 --ld-window-min 3`
h_dense_geno	synthetic, held out	60	6,000	`--geno-r2 --min-r2 0.05`

S2 Other tools

Table S2. Wall times in seconds on the AVX-512 VM, one script for every tool, best of two runs for runs under 5 s; vcftools is the reference time (best of three). PLINK at two threads; scikit-allel includes Python start-up and import. PLINK 2 over SwiftLD: 4.9, 10.2, 8.7, 7.6 and 8.0-fold (geometric mean 7.6); PLINK 1.9 over SwiftLD: 5.4 to 10.9-fold; scikit-allel over SwiftLD: 103 to 252-fold. PLINK 2's LD step alone over SwiftLD: 2.2 to 4.7-fold.

Set	vcftools	SwiftLD 2 threads	PLINK 2 import	PLINK 2 LD	PLINK 2 total	PLINK 1.9 total	scikit-allel
r_eur_hap	28.66	0.0777	0.2057	0.1739	0.3796	0.4192	14.30
r_all_hap	36.49	0.0367	0.2593	0.1134	0.3727	0.4006	9.259
r_afr_geno	24.79	0.0355	0.1430	0.1647	0.3078	0.3272	5.175
rh_eas_hap	20.20	0.0294	0.1150	0.1071	0.2221	0.2259	5.408
rh_sas_geno	12.83	0.0246	0.0820	0.1147	0.1967	0.1974	2.538

Table S3. Agreement with vcftools' r^2 on the position pairs both tools reported, ranges over the five 1000 Genomes regions.

Tool and mode	Sets	Shared pairs per set	Largest abs. difference	Fraction within 10^{-6}
PLINK 2 `--r2-phased` / `--r2-unphased`	5	276,017 to 1,850,551	0 to 1.0×10^{-6}	1.0000
PLINK 1.9 `--r2`, genotype sets	2	1,090,026 to 1,497,637	0 to 1.7×10^{-29}	1.0000
PLINK 1.9 `--r2`, haplotype sets	3	267,827 to 1,850,551	0.65 to 0.79	0.0025 to 0.0588
scikit-allel `rogers_huff_r`	5	276,017 to 1,850,551	1.6×10^{-6} to 4.8×10^{-6}	0.9996 to 0.9999

PLINK 1.9 `--r2` uses an unphased composite estimator even on phased data, hence the large differences on the haplotype sets; differences of about 10^{-6} reflect vcftools' six printed digits. Commands: PLINK 2 `--vcf in.vcf.gz --max-alleles 2 --make-pgen`, then `--r2-phased` or `--r2-unphased` with `--maf` and `--ld-window-kb` or `--ld-window` matching the vcftools window and `--threads 2`; PLINK 1.9 `--vcf in.vcf.gz --biallelic-only strict --make-bed`, then `--r2` with the same window; scikit-allel `allel.read_vcf`, then `allel.rogers_huff_r` on haplotypes or dosages, squared. Versions: PLINK v2.0.0-b.1 and v1.9.1 built from plink-ng commit 03bc669; scikit-allel 1.3.13.

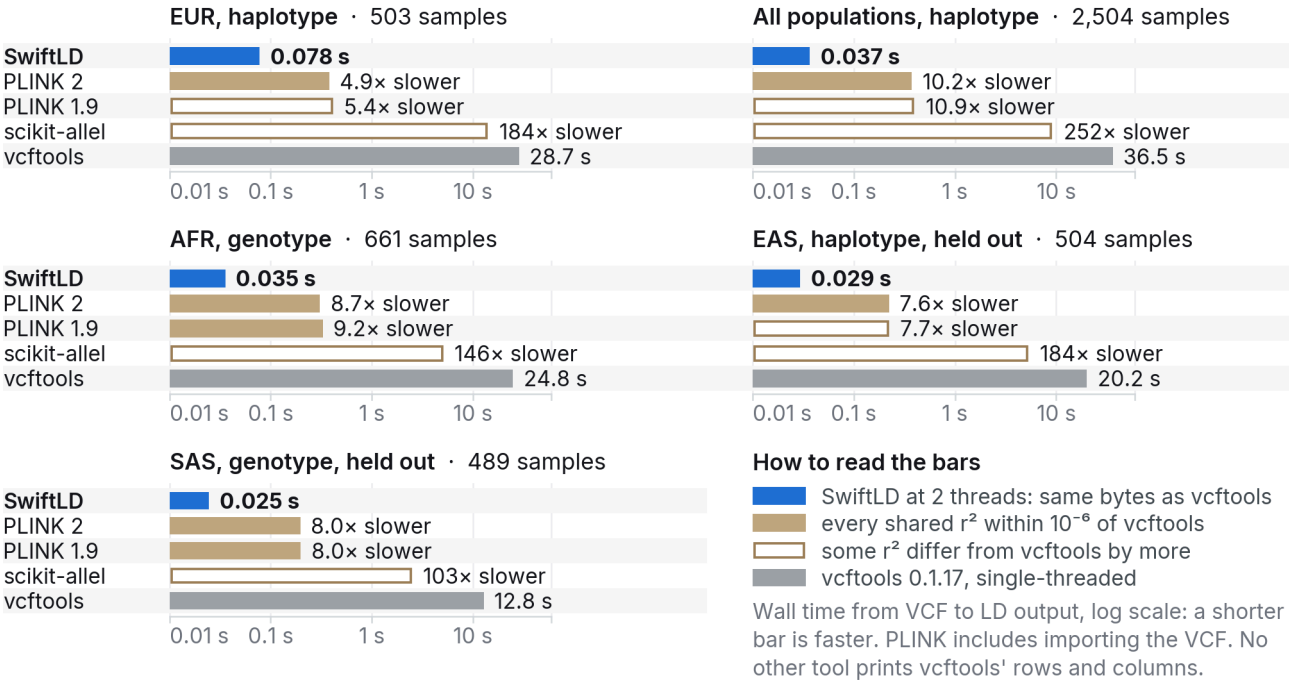


Figure S1. SwiftLD is the fastest tool on every 1000 Genomes region. Wall time from VCF to LD output on each region, log scale from 0.01 s; labels say how many times slower than SwiftLD each tool was, or give the time. PLINK times include importing the VCF; scikit-allel's include Python start-up. Solid bars: every r^2 both tools reported agreed with vcftools within 10^{-6}; open bars: some differed by more. One script timed every tool on the AVX-512 VM, PLINK at two threads (Table S2).

S3 vcftools 0.1.17 defects: reproductions and patches

All seven defects reproduce on vcftools 0.1.17 (commit 1c53c3c) built with `g++ -O2` (13.3.0, Ubuntu 24.04, `_FORTIFY_SOURCE` on by default) and zlib 1.3. With all five patches below applied, vcftools still wrote the reference output of the SAS region (sha256 4ce743d9…e27) byte for byte. The diffs below are shown with shortened context.

Table S4. Minimal reproductions of the seven defects; rows 2 to 5 use the build with patch 1 applied. Row 6: `std::getline` into member strings keeps the previous line's values. Row 7: the one-byte temporary-file encoding reads allele 2 at a biallelic site as missing, while `--freq` counts it. Neither has a patch here.

#	Input	Command	vcftools 0.1.17	With the patch
1	4 samples, 3 phased sites	`--vcf tiny.vcf --hap-r2`	"buffer overflow detected", exit 134	exit 0, 3 rows
2	150,000 records (6.5 MB text), gzip CRC byte flipped	`--gzvcf bad_crc.vcf.gz --hap-r2 --ld-window 2`	never terminates	"incorrect data check", exit 1
3	the tiny file	`--hap-r2 --no-such-option`; `--hap-r2 --geno-r2`	"Error: ...", exit 0	exit 1
4	sample D unphased at one site	`--hap-r2 --remove remove_D.txt`	"Require phased haplotypes", exit 1	exit 0, 3 rows
5	one site with every call missing (.|.)	`--hap-r2 --maf 0.05`	3 rows, two with N_CHR 0 and `-nan`	1 row
6	a record listing only sample A	`--hap-r2`	samples B to D copied from the previous record	issue only
7	genotype 0|2 at a biallelic site	`--hap-r2`; `--freq`	N_CHR 7 in LD; N_CHR 8 and frequencies summing to 0.875 in `--freq`	issue only

1. Temporary-file name buffers one byte short. Fifteen functions `strcpy` a string into `char tmpname[new_tmp.size()]`, leaving no room for the NUL; the patch adds the byte in all fifteen (first shown).

```
--- a/src/cpp/variant_file_format_convert.cpp
+++ b/src/cpp/variant_file_format_convert.cpp
@@ -26,7 +26,7 @@ void variant_file::output_as_plink(const parameters &params)
        string new_tmp = params.temp_dir+"/vcftools.XXXXXX";
-       char tmpname[new_tmp.size()];
+       char tmpname[new_tmp.size()+1];
        strcpy(tmpname, new_tmp.c_str());
```

2. Endless loop on a gzip data error. After a data error `gzgets` returns NULL while `gzeof` stays false.

```
--- a/src/cpp/vcf_file.cpp
+++ b/src/cpp/vcf_file.cpp
@@ -301,7 +301,13 @@ void vcf_file::read_line(string &out)
            tmp = gzgets(gzfile_in, gz_readbuffer, gzMAX_LINE_LEN);
            if (tmp == NULL)
+           {
+               int errnum = Z_OK;
+               const char *msg = gzerror(gzfile_in, &errnum);
+               if ((errnum != Z_OK) && (errnum != Z_BUF_ERROR))    // Z_BUF_ERROR: truncated input
+                   LOG.error("Error reading compressed input: " + string(msg));
                return;
+           }
```

3. Usage errors exit with status 0.

```
--- a/src/cpp/parameters.cpp
+++ b/src/cpp/parameters.cpp
@@ -349,7 +349,7 @@ void parameters::read_parameters()
        else
-           error("Unknown option: " + string(in_str), 0);
+           error("Unknown option: " + string(in_str), 1);
@@ -682,9 +682,9 @@ void parameters::check_parameters()
-   if (num_outputs > 1) error("Only one output function may be called.",0);
-   if (vcf_filename == "" && !stream_in) error("Input file required.", 0);
-   if (vcf_format == false && bcf_format == false) error("Must specify input file type",0);
+   if (num_outputs > 1) error("Only one output function may be called.",1);
+   if (vcf_filename == "" && !stream_in) error("Input file required.", 1);
+   if (vcf_format == false && bcf_format == false) error("Must specify input file type",1);
```

4. `--hap-r2` checks phase before excluding individuals. The patch moves the inclusion test above the phase check in three functions (first shown).

```
--- a/src/cpp/variant_file_output.cpp
+++ b/src/cpp/variant_file_output.cpp
@@ -1486,16 +1486,16 @@ void variant_file::output_haplotype_r2(const parameters &params)
            out_byte = 0x00;
-           if (e->get_indv_PHASE(ui) != '|')
-           {
-               remove(tmpname);
-               LOG.error("Require phased haplotypes for r^2 calculation (use --phased)\n");
-           }
            if (include_indv[ui] == false)
            {
                indv_miss++;
                continue;
            }
+           if (e->get_indv_PHASE(ui) != '|')
+           {
+               remove(tmpname);
+               LOG.error("Require phased haplotypes for r^2 calculation (use --phased)\n");
+           }
```

5. Sites with no called alleles pass `--maf`. Their minor allele frequency is NaN, and NaN comparisons pass the filter.

```
--- a/src/cpp/entry_filters.cpp
+++ b/src/cpp/entry_filters.cpp
@@ -817,6 +817,9 @@ void entry::filter_sites_by_frequency_and_call_rate(double min_maf, double
         max_m
    if ((maf < min_maf) || (maf > max_maf))
        passed_filters = false;

+   if ((N_non_missing_chr == 0) && ((min_maf > 0.0) || (max_maf < 1.0)))
+       passed_filters = false; // MAF undefined: no called alleles at this site
```