Input filtering

The following flags allow you to exclude samples and/or variants from an analysis batch based on a variety of criteria.

Some of these criteria are based on statistics such as estimated MAF that may vary through multiple filtering passes. If variation is problematic, use --freqx to export initial statistics, and then include --read-freq in all filtering passes where you want to refer back to the initial statistics.

ID lists

```
--keep <filename>
--remove <filename>
--keep-fam <filename>
--remove-fam <filename>
```

--keep accepts a space/tab-delimited text file with family IDs in the first column and within-family IDs in the second column, and removes all unlisted samples from the current analysis. **--remove** does the same for all listed samples.

Similarly, **--keep-fam** and **--remove-fam** accept text files with family IDs in the first column, and keep or remove entire families.

When operating on multiple ID lists, you may want to use these flags in conjunction with Unix text manipulation utilities (e.g. `cat`, `cut`, `sort`, `uniq`).

```
--extract ['range'] <filename>
--exclude ['range'] <filename>
```

--extract normally accepts a text file with a list of variant IDs (usually one per line, but it's okay for them to just be separated by spaces), and removes all unlisted variants from the current analysis.

With the '**range**' modifier, the input file should be in set range format instead.

--exclude does the same for all listed variants. Note that this is slightly different from PLINK 1.07's behavior when the main input fileset contains duplicate variant IDs: PLINK 1.9 removes all matches, while PLINK 1.07 just removes one of the matching variants. If your intention is to resolve duplicates, you should now use --bmerge instead of --exclude.

Cluster membership

```
--keep-clusters <filename>
--keep-cluster-names <name(s)...>
--remove-clusters <filename>
--remove-cluster-names <name(s)...>
```

If samples are assigned to clusters (via --within/--family), **--keep-clusters** and **--keep-cluster-names** can be used individually or in combination to define a list of clusters to keep; all samples not in one of those clusters are then removed from the current analysis. --keep-clusters accepts a text file with one cluster name per line, and --keep-cluster-names takes a space-delimited sequence of cluster names on the command line.

Similarly, **--remove-clusters** removes all samples in clusters named in a file, and **--remove-cluster-names** removes all samples in clusters named on the command line.

Set membership

```
--gene <set ID(s)...>
--gene-all
```

If variants have been assigned to sets (via --set/--make-set), **--gene** takes a space-delimited sequence of set names on the command line and removes all variants not in one of the named sets, while **--gene-all** only removes variants which aren't in any set (this used to happen automatically in some situations).

Attribute-based

```
--attrib <attrib file> [boolean condition description]
--attrib-indiv <sample attrib file> [boolean condition]
```

Given a (possibly gzipped) file assigning attributes to variants, and a comma-delimited list (**with no whitespace**) describing a boolean condition on the attributes, **--attrib** excludes all variants which are either missing from the attribute file or don't satisfy the condition. The attribute file is expected to have variant IDs in the first column of each line, followed by zero or more space-separated attribute names applying to the variant. (Variant IDs are not allowed to appear multiple times.) See snp129.attrib.gz on the resources page for an example.

--attrib-indiv expects an attribute file which starts with FID and IID columns instead of a variant ID column, and filters samples instead of variants.

The boolean condition is of the form

([has attribute p_1 or p_2...] AND [lacks attributes n_1 and n_2...])

where if there are no p_i's, the first predicate is true, and if there are no n_i's, the second predicate is true. (When there are multiple negative match conditions, PLINK 1.9 builds before 27 Jun 2015 incorrectly required only one attribute to be missing.) As mentioned above, the boolean condition description is expected to be in the form of a comma-delimited list; entries starting with '-' are added to the n_i attribute name list ("negative match conditions"), and the rest join the p_i list ("positive match conditions"). For example,

```
--attrib snps.txt exonic,-failed,-candidate
```

keeps variants with the 'exonic' attribute which also lack the 'failed' and 'candidate' attributes.

If the first entry in the filter description is a negative match, **you now must precede the '-' with a comma**, e.g.

```
--attrib snps.txt ,-failed
```

Without the comma, the PLINK 1.9 command line parser would interpret `-failed` as another flag. (We apologize for this incompatibility with PLINK 1.07.) If you are programmatically generating the second --attrib[-indiv] parameter, it is safe to always include a leading comma.

Chromosomes

```
--chr <number(s)/range(s)...>
--not-chr <number(s)/range(s)...>
```

--chr excludes all variants not on the listed chromosome(s). Normally, valid choices for humans are 0 (i.e. unknown), 1-22, X, Y, XY (pseudo-autosomal region of X; see --split-x/--merge-x), and MT. Separate multiple chromosomes with spaces or commas, and use dashes to specify ranges. Spaces are not permitted immediately before or after a range-denoting dash.

For example, the following are all valid and equivalent:

```
--chr 1-4, 22, xy
--chr 1-4 22 XY
--chr 1,2,3,4,22,25
```

You might wonder about the '25'. Non-autosomal chromosomes can also be identified by numeric code: if there are **n** autosomes, **n+1** is the X chromosome, **n+2** is Y, **n+3** is XY, and **n+4** is MT.

--not-chr is the reverse of --chr: variants on listed chromosome(s) are excluded. So

```
--not-chr 0 5-21 x y mt
```

is equivalent to the three --chr examples above (assuming human data).

If you specified --allow-extra-chr, you can refer to the extra chromosome codes by name. For example,

```
--allow-extra-chr --not-chr chr1_gl000191_random
```

Quick index search

```
--autosome
--autosome-xy
```

--autosome excludes all unplaced and non-autosomal variants, while **--autosome-xy** does not exclude the pseudo-autosomal region of X. They can be combined with --not-chr, e.g.

```
--autosome-xy --not-chr 5-21
```

is also equivalent to the three --chr examples.

Keep only SNPs

```
--snps-only ['just-acgt']
```

--snps-only excludes all variants with one or more multi-character allele codes. With '**just-acgt**', variants with single-character allele codes outside of {'A', 'C', 'G', 'T', 'a', 'c', 'g', 't', <missing code>} are also excluded.

Simple variant window

```
--from <variant ID>
--to <variant ID>
```

--from excludes all variants on different chromosomes than the named variant, as well as those with smaller base-pair position values. **--to** is similar, excluding variants with larger position values instead. If they are used together but the --from variant is after the --to variant, they are automatically swapped.

```
--snp <variant ID>
--window <total window size, in kb>
--exclude-snp <variant ID>
```

--snp specifies a single variant to load by name. If it's combined with **--window**, all variants with physical position no more than half the specified kb distance (decimal permitted) from the named variant are loaded as well.

Similarly, **--exclude-snp** specifies a single variant to exclude; this can also be combined with --window.

```
--from-bp <pos>
--to-bp <pos>
--from-kb <kb pos>
--to-kb <kb pos>
--from-mb <mb pos>
--to-mb <mb pos>
```

These flags let you use physical positions to specify a variant range to load. Kilobase and megabase values can include decimals. You are required to specify a single chromosome when using these.

Multiple ranges

```
--snps <variant ID(s)/range(s)...>
--exclude-snps <variant ID(s)/range(s)...>
```

--snps accepts a collection of individual variant IDs and variant ranges. For example,

```
--snps rs1111-rs2222, rs3333, rs4444
```

tells PLINK to load all variants between rs1111 and rs2222 inclusive, as well as rs3333 and rs4444. (Syntax works the same way as --chr. If your variant IDs contain dashes, you'll want to use the --d flag as well.) If rs1111 and rs2222 are on different chromosomes **i** < **j**, then all variants on chromosomes numbered between **i** and **j** are loaded, as well as the last variants on chromosome **i** and the first variants on chromosome **j**. (You can exclude some intermediate chromosomes by combining --snps with --not-chr.)

--exclude-snps excludes all the specified variants/ranges instead.

Arbitrary thinning

```
--thin <p>
--thin-count <n>
--bp-space <bp count>
--thin-indiv <p>
```

```
--thin-indiv-count <n>
  (alias: --max-indv)
```

--thin removes variants at random by retaining each variant with probability **p**, **--thin-count** removes variants at random until only **n** remain, and **--bp-space** excludes one variant from each pair closer than the given bp count. (Yes, --bp-space is equivalent to VCFtools --thin; we can't do much about this mixup without breaking backward compatibility.) Note that LD-based pruning also has a variant thinning effect, and is normally more useful than these three commands.

Similarly, **--thin-indiv** removes samples at random by retaining each sample with probability **p**, while **--thin-indiv-count** removes samples at random until only **n** remain.

Covariates

```
--filter <filename> <value(s)...>
--mfilter <n>
```

--filter accepts a space/tab-delimited text file with family IDs in the first column, within-family IDs in the second column, and a covariate in the third column. All samples either missing from the table, or with a covariate value which doesn't match any of the --filter parameters past the first, are removed from the analysis. Covariate values do not need to be numeric.

--mfilter causes the --filter parameter(s) to be compared with the covariate in the (**n**+2)th column instead.

Missing genotype rates

```
--geno [maximum per-variant]
--mind [maximum per-sample]
--oblig-missing <variant x block file> <block definition file>
```

--geno filters out all variants with missing call rates exceeding the provided value (default 0.1) to be removed, while **--mind** does the same for samples.

--oblig-missing lets you specify blocks of missing genotype calls for --geno and --mind to ignore. The first file should be a text file with variant IDs in the first column and block names in the second, while the second file should be in .clst format. See the PLINK 1.07 documentation for examples. (--oblig-clusters is a deprecated way to specify --oblig-missing's second parameter.)

If any genotype calls in a block are not actually missing, PLINK now reports an error; use --zero-cluster if you want to force those calls to missing instead.

Missing phenotypes

```
--prune
```

--prune filters out all samples with missing phenotypes.

Minor allele frequencies/counts

```
--maf [minimum freq]
--max-maf <maximum freq>
--mac <minimum count>
  (alias: --min-ac)
--max-mac <maximum count>
  (alias: --max-ac)
```

--maf filters out all variants with minor allele frequency below the provided threshold (default 0.01), while **--max-maf** imposes an upper MAF bound. Similarly, **--mac** and **--max-mac** impose lower and upper minor allele count bounds, respectively.

Only founders are normally considered by these filters; use --nonfounders to change this.

```
--maf-succ
```

--maf-succ causes primary minor allele frequencies to be estimated via the "rule of succession" employed by EIGENSOFT. I.e.,

q_{hat} := (1 + <observed minor allele count>) / (2 + <total observations>)

instead of the usual

q_{hat} := <observed minor allele count> / <total observations>.

This flag does not affect stratified MAF computations.

Hardy-Weinberg equilibrium tests

```
--hwe <p-value> ['midp'] ['include-nonctrl']
```

--hwe filters out all variants which have Hardy-Weinberg equilibrium exact test p-value below the provided threshold. We recommend setting a low threshold—serious genotyping errors often yield extreme p-values like 1e-50 which are detected by any reasonable configuration of this test, while genuine SNP-trait associations can be expected to deviate slightly from Hardy-Weinberg equilibrium (so it's dangerous to choose a threshold that filters out too many variants).

--hwe's '**midp**' modifier applies the mid-p adjustment described in Graffelman J, Moreno V (2013) The mid p-value in exact tests for Hardy-Weinberg equilibrium. The mid-p adjustment tends to bring the null rejection rate in line with the nominal p-value, and also reduces the filter's tendency to favor retention of variants with missing data. We recommend its use.

Because of the missing data issue, you should not apply a single p-value threshold across a batch of variants with highly variable missing call rates. A warning is now given whenever observation counts vary by more than 10%.

Only founders are considered by this test; use --nonfounders to change this. Also, with case/control data, cases and missing phenotypes are normally ignored; override this with '**include-nonctrl**'.

Mendel error rates

```
--me <max per-trio error rate> <max per-variant error rate> ['var-first']
--me-exclude-one [parent error ratio threshold]
```

--me filters out variants and samples/trios with **M**endel **e**rror rates exceeding the given thresholds. Haploid and mitochondrial data are currently ignored.

- By default, samples with only one parent in the dataset are not considered, and when parental genotype data is missing, (great-)grandparental data is not checked. This behavior can be changed with --mendel-duos/--mendel-multigen.
- By default, variants and trios are considered simultaneously. To filter out variants first, use the '**var-first**' modifier.
- By default, when a trio's Mendel error rate exceeds the given threshold, *all* members of the trio are excluded; to only exclude one member of each trio, use the **--me-exclude-one** flag. If a ratio is provided to --me-exclude-one, a parent is excluded whenever both parents are present and the ratio between the two parental .imendel error counts, considering only the immediate trio, exceeds the given ratio (see the "Samples implicated" column in the --mendel error type table); otherwise the child is excluded.
- When PLINK 1.07 --me was used either with --set-me-missing or without --make-bed/--recode, it would set some Mendel errors to missing *before all errors were identified*, and as a consequence some other errors were not noticed at all if overlapping trios were present. This no longer happens.

Quality scores

```
--qual-scores <filename> [quality score col.] [variant ID col.] [skip]
--qual-threshold <minimum score>
--qual-max-threshold <maximum score>
```

Given a file with variant IDs in the first column and quality scores in the second, **--qual-scores** removes all named variants with out-of-range or nonnumeric quality scores. The positions of the quality score and variant ID columns can now be adjusted with the second and third parameters. The optional fourth 'skip' parameter is either a nonnegative integer, in which case it indicates the number of lines to skip at the top of the file, or a single nonnumeric character, which causes each line with that leading character to be skipped. (Note that, if you want to specify '#' as the skip character, you need to surround it with single- or double-quotes in some Unix shells.)

For example, if `qual.vcf` is a well-formed VCF file,

```
--qual-scores qual.vcf 6 3 '#'
```

filters on the QUAL column.

The default range is [0, ∞). (This is a change from PLINK 1.07's [0, 1].) **--qual-threshold** changes the lower bound, and **--qual-max-threshold** lets you set an upper bound. Exact matches with the --qual-max-threshold value are not filtered out.

Note that these flags can be used to perform range-based filtering on other per-variant numeric values (e.g. average read depth) as well.

The related --qual-geno-scores family of flags has been provisionally retired, since they cannot be extended in a VCF-friendly manner. (We plan to provide VCF-friendly alternatives in the future.) If you would prefer to continue using them, contact us.

Miscellaneous

```
--allow-no-sex
--must-have-sex
```

By default, unless the input is loaded with --no-sex[1], samples with ambiguous sex have their phenotypes set to missing when analysis commands are run. Use **--allow-no-sex** to prevent this. (This setting is no longer ignored when --make-bed or --recode is present.)

However, phenotypes are normally retained for --make-bed, --recode, and --write-covar; use **--must-have-sex** to force phenotypes of ambiguous-sex samples to missing in this context.

1: --allow-no-sex was also unnecessary when using --pheno in PLINK 1.07. We believe that edge case just creates confusion, so it has been eliminated.

```
--filter-cases
--filter-controls
--filter-males
--filter-females
--filter-founders
--filter-nonfounders
```

Given case/control data, **--filter-cases** causes only cases to be included in the current analysis, while **--filter-controls** does the same for controls.

--filter-males and **--filter-females** behave analogously for males and females.

--filter-founders excludes all samples with at least one known parental ID from the current analysis (note that it is not necessary for that parent to be in the current dataset), while **--filter-nonfounders** does the reverse.

```
--nonfounders
```

By default, nonfounders are not counted by --freq[x] or --maf/--max-maf/--hwe. Use the **--nonfounders** flag to include them.

```
--make-founders ['require-2-missing'] ['first']
```

By default, if parental IDs are provided for a sample, they are not treated as a founder even if neither parent is in the dataset. With no modifiers, **--make-founders** clears both parental IDs whenever at least one parent is not in the dataset, and the affected samples are now considered founders. The '**require-2-missing**' modifier causes this to only happen when both parents are missing.

This normally happens after all sample-affecting filters have been applied (so it's too late to affect e.g. --filter-founders). If you want this to happen before all filters instead, add the '**first**' modifier.

Data management >>