Covar.cpp:117: LOGGER.e(0, "covariates can't have duplicate FID+IID.");

Covar.cpp:127: LOGGER.e(0, "covariates can't have duplicate FID+IID.");

Covar.cpp:138: LOGGER.e(0, "covariates can't have duplicate FID+IID.");

Covar.cpp:217: LOGGER.e(0, "something unexpected happened. Please check your covariate file.");

Covar.cpp:224: LOGGER.e(0, "0 covariate to be included.");

Covar.cpp:245: LOGGER.e(0, err\_string);

Covar.cpp:285: LOGGER.e(0, "column " + to\_string(i) + ", " + err\_string);

Covar.cpp:297: LOGGER.e(0, "column " + to\_string(i) + ", " + err\_string);

Covar.cpp:468: LOGGER.e(0, "can't read " + err\_string);

Covar.cpp:493: LOGGER.e(0, "less than " + to\_string(least\_col) + " columns in " + err\_string);

Covar.cpp:496: LOGGER.e(0, "can't read " + to\_string(last\_keep) + "th column from " + err\_string);

Covar.cpp:541: LOGGER.e(0, "can't read " + to\_string(last\_keep) + "th column of line " + to\_string(line\_number) + " from " + err\_string);

Covar.cpp:568: LOGGER.e(0, "too many levels in covariate #" + to\_string(col\_index + 1) + ". You may fit it as a quantitative covariate using --qcovar.");

Covar.cpp:576: LOGGER.e(0, "line " + to\_string(line\_number) + " contains non-numeric values in " + err\_string);

Covar.cpp:644: LOGGER.e(0, "No main function in covariate yet.");

FastFAM.cpp:255: //LOGGER.e(0, "debug");

FastFAM.cpp:330: LOGGER.e(0, "Some sample IDs in the model file do not exist in genotype file!");

FastFAM.cpp:345: LOGGER.e(0, "can't open file [" + bin\_file + "] to read.");

FastFAM.cpp:349: LOGGER.e(0, "can't read magic number.");

FastFAM.cpp:352: LOGGER.e(0, "wrong header in [" + bin\_file + "]. This file can only be generated from GCTA model.");

FastFAM.cpp:357: LOGGER.e(0, "can't read sample size.");

FastFAM.cpp:360: LOGGER.e(0, "wrong number of samples in [" + bin\_file + "]. This file can only be generated from GCTA model.");

FastFAM.cpp:365: LOGGER.e(0, "can't read number of covariates in [" + bin\_file + "]. this file can only be generated from GCTA model.");

FastFAM.cpp:371: LOGGER.e(0, "can't read covariate flag");

FastFAM.cpp:375: LOGGER.e(0, "can't tao value in [" + bin\_file + "]. This file can only be generated from GCTA model.");

FastFAM.cpp:379: LOGGER.e(0, "can't c\_inf in [" + bin\_file + "]. This file can only be generated from GCTA model.");

FastFAM.cpp:388: LOGGER.e(0, "failed to read mu.");

FastFAM.cpp:396: LOGGER.e(0, "failed to read phenotype.");

FastFAM.cpp:407: LOGGER.e(0, "failed to read covariates from model.");

FastFAM.cpp:417: LOGGER.e(0, "failed to read H from model.");

FastFAM.cpp:448: LOGGER.e(0, "Some sample IDs in the model file do not exist in genotype file!");

FastFAM.cpp:458: LOGGER.e(0, "can't open file [" + bin\_file + "] to read.");

FastFAM.cpp:461: LOGGER.e(0, "can't read header in [" + bin\_file + "].");

FastFAM.cpp:464: LOGGER.e(0, "wrong header in [" + bin\_file + "]. This file can only be generated from GCTA model.");

FastFAM.cpp:467: LOGGER.e(0, "the sample IDs in model is different from the ID file!");

FastFAM.cpp:495: LOGGER.e(0, "failed to read phenotype from model.");

FastFAM.cpp:510: LOGGER.e(0, "failed to read covariates from model.");

FastFAM.cpp:524: LOGGER.e(0, "failed to read covariates from model.");

FastFAM.cpp:558: LOGGER.e(0, "failed to read the V inverse from model.");

FastFAM.cpp:623: LOGGER.e(0, "Did you forget to specify --pheno?");

FastFAM.cpp:636: LOGGER.e(0, "No overlapping individual with non-missing data to be included from the covariate file(s).");

FastFAM.cpp:762: if(!pheno\_w) LOGGER.e(0, "failed to write " + options["out"]+".cphen");

FastFAM.cpp:817: LOGGER.e(0, "The Vp is below 1e-5. Please check: 1. Is there a scaling issue with the phenotype? 2. Can the covariates explain all the Vp (e.g., phenotype is included as a covariate by accident)? 3. If it is a binary trait, is the prevalence very rare?");

FastFAM.cpp:863: LOGGER.e(0, "Unknown method to estimate the Vg");

FastFAM.cpp:904: if(!inv\_id) LOGGER.e(0, "failed to write " + options["out"]+".grm.id");

FastFAM.cpp:917: if(!inv\_id) LOGGER.e(0, "failed to write " + options["out"]+".grm.id");

FastFAM.cpp:930: LOGGER.e(0, "can't write to [" + options["out"] + ".grm.inv]");

FastFAM.cpp:948: LOGGER.e(0, "can't read file [" + id\_file + "].");

FastFAM.cpp:968: LOGGER.e(0, "sample IDs are not same from line " + to\_string(cur\_index + 1) + " in [" + id\_file + "].");

FastFAM.cpp:976: LOGGER.e(0, "File is empty or the line number is not consistent in inverse V [" + id\_file + "].");

FastFAM.cpp:985: LOGGER.e(0, "can't open the file.");

FastFAM.cpp:998: LOGGER.e(0, "can't read file in pos: " + to\_string(cur\_pos));

FastFAM.cpp:1014: if(!inv\_id) LOGGER.e(0, "failed to write " + options["out"]+".grm.id");

FastFAM.cpp:1043: LOGGER.e(0, "can't open " + bin\_file + " to write.");

FastFAM.cpp:1047: LOGGER.e(0, "can't write header to " + bin\_file + ".");

FastFAM.cpp:1053: LOGGER.e(0, "can't write phenotype to " + bin\_file + ".");

FastFAM.cpp:1057: LOGGER.e(0, "can't write phenotype to " + bin\_file + ".");

FastFAM.cpp:1065: LOGGER.e(0, "can't write covariates to " + bin\_file + ".");

FastFAM.cpp:1071: LOGGER.e(0, "can't write covariates to " + bin\_file + ".");

FastFAM.cpp:1084: LOGGER.e(0, "can't write V inverse to " + bin\_file + ".");

FastFAM.cpp:1219: LOGGER.e(0, "can't invert the V");

FastFAM.cpp:1428: LOGGER.e(0, "can't invert the V");

FastFAM.cpp:1442: LOGGER.e(0, "can't invert XtViX");

FastFAM.cpp:1472: LOGGER.e(0, "Hi can't be inverted.");

FastFAM.cpp:1509: LOGGER.e(0, "Error to open the " + options["out"] + ".reml");

FastFAM.cpp:1605: LOGGER.e(0, "fastGWA-REML can't converge.");

FastFAM.cpp:1615: LOGGER.e(0, "fastGWA-REML can't converge - hit upper limit!");

FastFAM.cpp:1670: LOGGER.e(0, "can't solve the regression");

FastFAM.cpp:1716: LOGGER.e(0, "can't read [" + filename + ".grm.sp]");

FastFAM.cpp:1832: LOGGER.e(0, "can't invert the V matrix");

FastFAM.cpp:1845: LOGGER.e(0, "can't read " + to\_string(num\_marker\_rand) + " SNPs from autosome for tuning.");

FastFAM.cpp:1901: LOGGER.e(0, "some SNPs could not be read successfully!");

FastFAM.cpp:1940: LOGGER.e(0, "Not enough valid null SNPs (<100). \nUsers may check if there are too many signals or the MAF/INFO/genotype-missing-rate criteria is too stringent.");

FastFAM.cpp:1952: LOGGER.e(0, "Inconsistent sample size - there may be some unknown bugs.");

FastFAM.cpp:1958: LOGGER.e(0, "can't write fastGWA MLM parameters to the file.");

FastFAM.cpp:1991: LOGGER.e(0, "can't invert the sparse GRM");

FastFAM.cpp:1996: LOGGER.e(0, "Unknown inverse methods");

FastFAM.cpp:2071: LOGGER.e(0, "can't write allele frequency to [" + sFileName + ".bin].");

FastFAM.cpp:2075: LOGGER.e(0, "can't write beta to [" + sFileName + ".bin].");

FastFAM.cpp:2078: LOGGER.e(0, "can't write se to [" + sFileName + ".bin].");

FastFAM.cpp:2081: LOGGER.e(0, "can't write p to [" + sFileName + ".bin].");

FastFAM.cpp:2084: LOGGER.e(0, "can't write Padj to [" + sFileName + ".bin].");

FastFAM.cpp:2088: LOGGER.e(0, "can't write N to [" + sFileName + ".bin].");

FastFAM.cpp:2092: LOGGER.e(0, "can't write INFO score to [" + sFileName + ".bin].");

FastFAM.cpp:2140: LOGGER.e(0, "can't write allele frequency to [" + sFileName + ".bin].");

FastFAM.cpp:2144: LOGGER.e(0, "can't write beta to [" + sFileName + ".bin].");

FastFAM.cpp:2147: LOGGER.e(0, "can't write se to [" + sFileName + ".bin].");

FastFAM.cpp:2150: LOGGER.e(0, "can't write p to [" + sFileName + ".bin].");

FastFAM.cpp:2154: LOGGER.e(0, "can't write N to [" + sFileName + ".bin].");

FastFAM.cpp:2158: LOGGER.e(0, "can't write INFO score to [" + sFileName + ".bin].");

FastFAM.cpp:2295: LOGGER.e(0, "can't open [" + sFileName + "] to write.");

FastFAM.cpp:2303: LOGGER.e(0, "can't open [" + sFileName + ".snpinfo] to write.");

FastFAM.cpp:2308: LOGGER.e(0, "can't open [" + sFileName + ".bin] to write.");

FastFAM.cpp:2393: LOGGER.e(0, "Obsoleted flag. Try --fastGWA-mlm or --fastGWA-mlm-exact");

FastFAM.cpp:2402: LOGGER.e(0, "Obsoleted flag. Try --fastGWA-mlm or --fastGWA-mlm-exact");

FastFAM.cpp:2410: LOGGER.e(0, curFlag + "can't deal with 0 or > 1 files");

FastFAM.cpp:2420: LOGGER.e(0, "--fastGWA-mlm must run with --grm-sparse");

FastFAM.cpp:2431: LOGGER.e(0, "--fastGWA-mlm-binary must run with --grm-sparse");

FastFAM.cpp:2440: LOGGER.e(0, curFlag + " can only work with --fastGWA-mlm");

FastFAM.cpp:2450: LOGGER.e(0, "--fastGWA-mlm-exact must run with --grm-sparse");

FastFAM.cpp:2460: LOGGER.e(0, "--fastGWA-lr can't run with --grm-sparse currently");

FastFAM.cpp:2472: LOGGER.e(0, curFlag + " can't handle more than 2 values");

FastFAM.cpp:2483: LOGGER.e(0, curFlag + "can't deal with covar other than 1");

FastFAM.cpp:2551: LOGGER.e(0, "--load-inv can't load multiple files");

FastFAM.cpp:2570: LOGGER.e(0, "can't read " + options["model\_file"] + ".mdl.id");

FastFAM.cpp:2578: LOGGER.e(0, "can't read the model binary file.");

FastFAM.cpp:2582: LOGGER.e(0, "--load-model can't load multiple files");

FastFAM.cpp:2588: LOGGER.e(0, "can't generate and load model at the same time");

FastFAM.cpp:2601: LOGGER.e(0, "can't read " + options["geneset"]);

FastFAM.cpp:2609: LOGGER.e(0, "set based test can only be applied to the 2nd step with --load-model flag.");

FastFAM.cpp:2612: LOGGER.e(0, "set based test should load the set list by --set-list flag");

FastFAM.cpp:2752: LOGGER.e(0, "can't find the region set. Specify it by --set-list");

FastFAM.cpp:2764: LOGGER.e(0, "can't open [" + sFileName + "] to write.");

FastFAM.cpp:2773: LOGGER.e(0, "can't find a valid gene in the genotype file");

FastFAM.cpp:2941: LOGGER.e(0, "XtWX is not invertible!");

FastFAM.cpp:2972: LOGGER.e(0, "can't invert the V matrix!");

FastFAM.cpp:2977: LOGGER.e(0, "can't get the ViX matrix!");

FastFAM.cpp:2985: LOGGER.e(0, "XtViX is not invertible!");

FastFAM.cpp:3026: LOGGER.e(0, "can't invert the V matrix!");

FastFAM.cpp:3033: LOGGER.e(0, "XtViX is not invertible!");

FastFAM.cpp:3056: LOGGER.e(0, "Error to open the " + options["out"] + ".reml");

FastFAM.cpp:3234: LOGGER.e(0, "can't invert the V matrix!");

FastFAM.cpp:3241: LOGGER.e(0, "XtViX is not invertible!");

FastFAM.cpp:3289: LOGGER.e(0, "XtWX is not invertible!");

FastFAM.cpp:3301: LOGGER.e(0, "can't invert the V matrix!");

FastFAM.cpp:3308: LOGGER.e(0, "XtViX is not invertible!");

FastFAM.cpp:3355: if(!inv\_id) LOGGER.e(0, "failed to write " + options["out"]+".grm.id");

FastFAM.cpp:3425: LOGGER.e(0, "can't read " + to\_string(num\_marker\_rand) + " SNPs from autosome for tuning.");

FastFAM.cpp:3463: LOGGER.e(0, "failed to estimate gamma. There are too many signals! You may use --cv-threshold to lower the threshold");

FastFAM.cpp:3475: LOGGER.e(0, "some SNPs were not read successfully!");

GRM.cpp:60: LOGGER.e(0, "can't open " + grm\_file + ".grm.bin");

GRM.cpp:63: LOGGER.e(0, "The IDs in GRM and the IDs in the GRM binary file do not match [" + grm\_file + "]");

GRM.cpp:106: LOGGER.e(0, "can't open " + mgrm\_file + " to read.");

GRM.cpp:126: LOGGER.e(0, out\_err);

GRM.cpp:129: LOGGER.e(0, "only 2 GRMs are supported currently.");

GRM.cpp:138: LOGGER.e(0, "The sample IDs in the two GRMs are not same. Try --unify-grm first.");

GRM.cpp:141: if(!o\_id) LOGGER.e(0, "can't write to [" + options["out"] + ".grm.id]");

GRM.cpp:154: LOGGER.e(0, "The size of [" + files[0] + ".grm.bin] is not correct.");

GRM.cpp:157: LOGGER.e(0, "The size of [" + files[0] + ".grm.N.bin] is not correct.");

GRM.cpp:160: LOGGER.e(0, "The size of [" + files[1] + ".grm.bin] is not correct.");

GRM.cpp:163: LOGGER.e(0, "The size of [" + files[1] + ".grm.N.bin] is not correct.");

GRM.cpp:191: LOGGER.e(0, "can't write to [" + out\_file + ".grm.bin].");

GRM.cpp:194: LOGGER.e(0, "can't write to [" + out\_file + ".grm.N.bin].");

GRM.cpp:211: LOGGER.e(0, "can't open " + mgrm\_file + " to read.");

GRM.cpp:231: LOGGER.e(0, out\_err);

GRM.cpp:234: LOGGER.e(0, "not enough valid GRMs to be unified");

GRM.cpp:319: LOGGER.e(0, "can't read " + file\_name + ".");

GRM.cpp:324: LOGGER.e(0, "can't write to " + wfile\_name + ".");

GRM.cpp:342: LOGGER.e(0, "Error reading [" + file\_name + "], in position " + to\_string(ftell(h\_grm)), ".");

GRM.cpp:355: LOGGER.e(0, "Error reading [" + file\_name + "], in position " + to\_string(bytes\_offset), ".");

GRM.cpp:364: LOGGER.e(0, "writing to [" + wfile\_name + "], pos: " + std::to\_string(ftell(h\_wgrm)) + ".");

GRM.cpp:382: if(!o\_id) LOGGER.e(0, "can't write to [" + options["out"] + ".grm.id]");

GRM.cpp:387: if(!o\_fam) LOGGER.e(0, "can't write to [" + options["out"] + ".grm.sp]");

GRM.cpp:416: LOGGER.e(0, "Failed to read GRM between line " + to\_string(index\_grm\_pairs[part\_index].first + 1) + " and "

GRM.cpp:467: LOGGER.e(0, "Failed to write the output");

GRM.cpp:527: LOGGER.e(0, "Failed to write the GRM N");

GRM.cpp:549: LOGGER.e(0, "can't write to [" + options["out"] + ".grm.id]");

GRM.cpp:560: LOGGER.e(0, "can't write to [" + options["out"] + ".family.txt, .singleton.txt]");

GRM.cpp:573: LOGGER.e(0, "Failed to read GRM between line " + to\_string(index\_grm\_pairs[part\_index].first + 1) + " and "

GRM.cpp:669: LOGGER.e(0, "can't open [" + options["out"] + ".grm.bin] to write");

GRM.cpp:702: LOGGER.e(0, "Failed to read GRM between line " + to\_string(index\_grm\_pairs[part\_index].first + 1) + " and "

GRM.cpp:723: LOGGER.e(0, "Failed to write the output file, please check the disk condition or permission");

GRM.cpp:757: LOGGER.e(0, "cannot calculate when" + to\_string(part) + " is larger than " + to\_string(this->num\_parts));

GRM.cpp:776: LOGGER.e(0, "can't allocate enough memory for genotype buffer.");

GRM.cpp:794: LOGGER.e(0, "can't allocate enough memory to store the mask.");

GRM.cpp:810: LOGGER.e(0, "can't allocate enough memory to store the (parted) GRM: " + to\_string(fill\_grm\*sizeof(double) / 1024.0/1024/1024) + "GB required.");

GRM.cpp:816: LOGGER.e(0, "can't allocate enough memory to store (parted) N: " + to\_string(fill\_grm\*sizeof(uint32\_t) / 1024.0/1024/1024) + "GB required");

GRM.cpp:872: if (!grm\_id) { LOGGER.e(0, "cannot open the file [" + o\_grm\_id + "] to write"); }

GRM.cpp:1380: LOGGER.e(0, "can't open " + o\_name + ".grm.sp to write");

GRM.cpp:1388: LOGGER.e(0, "can't open " + o\_name + ".grm.bin or .grm.N.bin to write");

GRM.cpp:1704: LOGGER.e(0, "can't handle multiple GRM files");

GRM.cpp:1711: LOGGER.e(0, "it is not allowed to have the same file name for the input and the output files");

GRM.cpp:1744: LOGGER.e(0, "can't specify --make-grm-part, --make-grm-d-part or --make-grm-xchr-part together");

GRM.cpp:1753: LOGGER.e(0, part\_grm\_symbol + " can only deal with integer value");

GRM.cpp:1756: LOGGER.e(0, part\_grm\_symbol + "arguments should be >= 1");

GRM.cpp:1759: LOGGER.e(0, part\_grm\_symbol + "the 1st parameter (number of parts) can't be smaller than the 2nd parameter");

GRM.cpp:1778: LOGGER.e(0, part\_grm\_symbol + " takes two arguments: the number of total parts and the part number to be calculated currently");

GRM.cpp:1879: LOGGER.e(0, "--grm-cutoff can't deal with more than one value currently");

GRM.cpp:1882: LOGGER.e(0, "can't find the --grm flag that is essential to --grm-cutoff");

GRM.cpp:1897: LOGGER.e(0, opitem + " can't deal with more than one value currently");

GRM.cpp:1900: LOGGER.e(0, "can't find the --grm flag that is essential to " + opitem);

GRM.cpp:1926: LOGGER.e(0, "can't find the --grm flag that is essential to " + curFlag);

GRM.cpp:1938: LOGGER.e(0, curFlag + " can't deal with more than one value currently");

GRM.cpp:1981: LOGGER.e(0, "can't allocate enough memory for genotype buffer.");

GRM.cpp:1989: LOGGER.e(0, "the original version has been deleted. Please use GCTA >= 1.92.4");

GRM.cpp:2019: LOGGER.e(0, "can't allocate enough memory for genotype buffer.");

GRM.cpp:2027: LOGGER.e(0, "the original version has been deleted. Please use GCTA 1.92.4");

Geno.cpp:162: LOGGER.e(0, "No genotype file is specified");

Geno.cpp:400: LOGGER.e(0, "the third column should be numeric");

Geno.cpp:403: LOGGER.e(0, "frequency values should range from 0 to 1");

Geno.cpp:456: if (!o\_freq) { LOGGER.e(0, "cannot open the file [" + name\_frq + "] to write"); }

Geno.cpp:515: LOGGER.e(0, message);

Geno.cpp:536: LOGGER.e(0, "can't open [" + cur\_bed\_file + "] to read.");

Geno.cpp:560: LOGGER.e(0, "read error for [" + geno\_files[cur\_file\_index] + "].\nThere might be some problems with your storage, or have you changed the file?");

Geno.cpp:568: LOGGER.e(0, "read error for [" + geno\_files[cur\_file\_index] + "].\nThere might be some problems with your storage, or have you changed the file?");

Geno.cpp:649: LOGGER.e(0, "can't allocate enough memory to read genotype.");

Geno.cpp:674: LOGGER.e(0, "can't allocate enough memory to read genotype.");

Geno.cpp:703: LOGGER.e(0, "inconsistent sample sizes between the .bgen file [" + geno\_files[i] + "] and the .sample file (specified by --sample).");

Geno.cpp:714: LOGGER.e(0, "can't allocate enough memory to read genotype.");

Geno.cpp:846: LOGGER.e(0, err);

Geno.cpp:897: LOGGER.e(0, "found heterozygote coding (=1) on ChrX (23) for male samples. GCTA follows the convention from PLINK, treats ChrX (23) as non-PAR and codes male genotype on ChrX as either 0 or 2. \nPlease check the gender information in your genotype file, or use PLINK --split-x to separate the ChrX into non-PAR (chr23) and PAR (chr25) regions for further analysis.\nThe non-PAR region is assumed to have the dosage compensation problem, and you can specify it with the --dc flag, with 0: no compensation, and 1: dosage compensation (the default is 1 in GCTA).");

Geno.cpp:1051: LOGGER.e(0, "can't read " + to\_string(lag\_index) + "th SNP in [" + geno\_files[fileIndex] + "].");

Geno.cpp:1163: LOGGER.e(0, "decompress genotype file error in " + error\_promp);

Geno.cpp:1170: LOGGER.e(0, "not compressed by zstd in " + error\_promp);

Geno.cpp:1173: LOGGER.e(0, "original size unknown in " + error\_promp);

Geno.cpp:1177: LOGGER.e(0, "size stated in the compressed file is different from " + error\_promp);

Geno.cpp:1182: LOGGER.e(0, "decompress genotype file error: " + string(ZSTD\_getErrorName(dSize)) + " in " + error\_promp);

Geno.cpp:1185: LOGGER.e(0, "unknown compress format in " + error\_promp);

Geno.cpp:1193: LOGGER.e(0, "inconsistent number of samples in " + error\_promp);

Geno.cpp:1197: LOGGER.e(0, "multi-allelic SNPs detected in " + error\_promp);

Geno.cpp:1205: LOGGER.e(0, "multiploidy detected in " + error\_promp);

Geno.cpp:1273: LOGGER.e(0, "multiploidy detected in " + error\_promp);

Geno.cpp:1517: LOGGER.e(0, "reach the end of the genotype file, but still cannot finish the reading process (EOF error).");

Geno.cpp:1641: LOGGER.e(0, "The size of the extracted markers is larger than the buffer size.");

Geno.cpp:1672: LOGGER.e(0, "inconsistent sample sizes between the .bgen file [" + geno\_files[i] + "] and the .sample file (specified by --sample).");

Geno.cpp:1797: LOGGER.e(0, "decompress genotype data error in " + error\_promp);

Geno.cpp:1804: LOGGER.e(0, "not compressed by zstd in " + error\_promp);

Geno.cpp:1807: LOGGER.e(0, "original size unknown in " + error\_promp);

Geno.cpp:1811: LOGGER.e(0, "the size stated in compressed data is different from " + error\_promp);

Geno.cpp:1816: LOGGER.e(0, "decompress genotype error: " + string(ZSTD\_getErrorName(dSize)) + " in " + error\_promp);

Geno.cpp:1819: LOGGER.e(0, "unknown compress format in " + error\_promp);

Geno.cpp:1839: LOGGER.e(0, "inconsistent number of samples in " + error\_promp );

Geno.cpp:1843: LOGGER.e(0, "multi-allelic SNPs detected in " + error\_promp);

Geno.cpp:1851: LOGGER.e(0, "multi-ploidy detected in " + error\_promp);

Geno.cpp:1860: LOGGER.e(0, "GCTA currently cannot support phased data in " + error\_promp);

Geno.cpp:1898: LOGGER.e(0, "multi-allelic SNPs detected in " + error\_promp);

Geno.cpp:2037: LOGGER.e(0, "multi-allelic SNPs detected in " + error\_promp);

Geno.cpp:2068: LOGGER.e(0, "multi-allelic SNPs detected in " + error\_promp);

Geno.cpp:2152: LOGGER.e(0, message);

Geno.cpp:2189: if(lag\_index < 0)LOGGER.e(0, "strange index in " + to\_string(curRawIndex)

Geno.cpp:2197: LOGGER.e(0, "read [" + geno\_files[curFileID] + "] error.\nThere might be some problems with your storage, or have you changed the file?");

Geno.cpp:2355: LOGGER.e(0, "reach the end of the BED file, but still but still cannot finish the reading process (EOF error).");

Geno.cpp:2366: LOGGER.e(0, "requested marker number exceeded the buffer size.");

Geno.cpp:2377: if(lag\_index < 0)LOGGER.e(0, "strange index in " + to\_string(curRawIndex)

Geno.cpp:2382: LOGGER.e(0, "read [" + geno\_files[curFileID] + "] error.\nThere might be some problems with your storage, or have you changed the file?");

Geno.cpp:2509: LOGGER.e(0, "open genotype [" + cur\_filename + "], " + string(strerror(errno)));

Geno.cpp:2651: if (!out) { LOGGER.e(0, "cannot open the file [" + options["out"] + ".sum" + "] to write"); }

Geno.cpp:2814: LOGGER.e(0, "decompress genotype data error in " + to\_string(raw\_index) + "th SNP.");

Geno.cpp:2819: LOGGER.e(0, "inconsistent number of samples in " + to\_string(raw\_index) + "th SNP." );

Geno.cpp:2823: LOGGER.e(0, "multi-allelic SNPs still detected, the bgen file might be malformed.");

Geno.cpp:2836: LOGGER.e(0, "can't support phased data currently.");

Geno.cpp:2842: LOGGER.e(0, "can't support probability bits other than in byte unit.");

Geno.cpp:2846: LOGGER.e(0, "malformed data in " + to\_string(raw\_index) + "th SNP.");

Geno.cpp:2898: LOGGER.e(0, "multi-allelic SNPs detected in the " + to\_string(raw\_index) + "th SNP.");

Geno.cpp:2943: LOGGER.e(0, " multi-allelic SNPs detected in the " + to\_string(raw\_index) + "th SNP.");

Geno.cpp:2986: LOGGER.e(0, err\_string);

Geno.cpp:2993: LOGGER.e(0, err\_string);

Geno.cpp:3003: LOGGER.e(0, err\_string);

Geno.cpp:3285: LOGGER.e(0, " reach the end of the BED file, but still but still cannot finish the reading process (EOF error).");

Geno.cpp:3424: LOGGER.e(0, "MAF can't be negative: " + to\_string(val));

Geno.cpp:3444: LOGGER.e(0, "value specified by --max-maf can't be smaller than that by --min-maf");

Geno.cpp:3456: LOGGER.e(0, "max MAF can't be negative or larger than 0.5");

Geno.cpp:3480: LOGGER.e(0, "no " + key\_name + " parameter found");

Geno.cpp:3484: LOGGER.e(0, key\_name + " " + options[key\_store] + " not found");

Geno.cpp:3510: LOGGER.e(0, "illegal value in --maf");

Geno.cpp:3513: LOGGER.e(0, "--maf can't be smaller than 0 or larger than 0.5");

Geno.cpp:3519: LOGGER.e(0, "cannot support multiple values in --maf currently");

Geno.cpp:3530: LOGGER.e(0, "illegal value in --maf");

Geno.cpp:3533: LOGGER.e(0, "--max-maf can't be smaller than 0 or larger than 0.5");

Geno.cpp:3536: LOGGER.e(0, "cannot support multiple values in --maf currently ");

Geno.cpp:3542: LOGGER.e(0, "value specified by --max-maf can't be smaller than that by --min-maf");

Geno.cpp:3566: LOGGER.e(0, "illegal value in " + flag);

Geno.cpp:3569: LOGGER.e(0, "multiple values in " + flag + ", not supported currently.");

Geno.cpp:3628: LOGGER.e(0, "can't recognize recode method: " + options\_in["--recodet"][0]);

Geno.cpp:3802: if (!osOut) { LOGGER.e(0, "cannot open the file [" + name\_out + "] to write."); }

Geno.cpp:3833: if (!osOut) { LOGGER.e(0, "cannot open the file [" + name\_out + "] to write."); }

Geno.cpp:3994: if (!o\_freq) { LOGGER.e(0, "cannot open the file [" + name\_frq + "] to write"); }

LD.cpp:35: LOGGER.e(0, "can't open " + options["out"] + " for writing.");

LD.cpp:106: LOGGER.e(0, "can't write to " + options["out"] + ".");

LD.cpp:109: LOGGER.e(0, "can't write to " + options["out"] + ".");

LD.cpp:171: LOGGER.e(0, "LD window is not an integer.");

Marker.cpp:91: LOGGER.e(0, "no marker exists");

Marker.cpp:119: LOGGER.e(0, "0 SNP remained!");

Marker.cpp:129: LOGGER.e(0, "can't open [" + filename + "] to read]");

Marker.cpp:413: LOGGER.e(0, "the SNP index is too large in" + to\_string(raw\_index));

Marker.cpp:451: LOGGER.e(0, "The .bim file only has " + to\_string(ncol) + " columns, which is not a valid.");

Marker.cpp:471: LOGGER.e(0, "can't find all the essential columns in PVAR file.");

Marker.cpp:474: LOGGER.e(0, "invalid PVAR file. It should start with #CHROM.");

Marker.cpp:505: LOGGER.e(0, "Line " + to\_string(nHeader + i + 1) + " contains illegal distance value.");

Marker.cpp:542: LOGGER.e(0, "invalid PVAR file.");

Marker.cpp:550: LOGGER.e(0, "cannot open the file [" + bim\_file + "] to read");

Marker.cpp:567: LOGGER.e(0, "the bim file [" + bim\_file + "], line " + to\_string(line\_number)

Marker.cpp:579: LOGGER.e(0, "Line " + to\_string(line\_number) + " of [" + bim\_file +

Marker.cpp:588: LOGGER.e(0, "Line " + to\_string(line\_number) + " of [" + bim\_file +

Marker.cpp:621: LOGGER.e(0, "cannot open the file [" + gfile + "] to read");

Marker.cpp:634: LOGGER.e(0, "invalid gene list file");

Marker.cpp:652: LOGGER.e(0, " line " + to\_string(line\_number)

Marker.cpp:659: LOGGER.e(0, " Line " + to\_string(line\_number) + " of [" + gfile +

Marker.cpp:669: LOGGER.e(0, " Line " + to\_string(line\_number) + " of [" + gfile +

Marker.cpp:698: //LOGGER.e(0, " Line " + to\_string(line\_number) + ", chromosome information can't be found in the genotype file");

Marker.cpp:740: LOGGER.e(0, "bad magic number in the bgen file.");

Marker.cpp:747: LOGGER.e(0, "strange header length - might be an invalid bgen file.");

Marker.cpp:782: LOGGER.e(0, "GCTA only supports bgen version 1.2 and 1.3. Use QCTOOLv2 to convert files with < v1.2 into newer version.");

Marker.cpp:811: LOGGER.e(0, "can't open index file: " + string(sqlite3\_errmsg(db)) +

Marker.cpp:821: LOGGER.e(0, "bad index file: " + string(sqlite3\_errmsg(db)) +

Marker.cpp:840: LOGGER.e(0, "can't read bgen file [" + bgen\_file + "], " + string(strerror(errno)));

Marker.cpp:845: LOGGER.e(0, "can't read bgen file [" + bgen\_file + "], " + string(strerror(errno)));

Marker.cpp:849: LOGGER.e(0, "bad index file, first 1000 bytes aren't consistent."

Marker.cpp:855: LOGGER.e(0, "bad index file, file size isn't consistent."

Marker.cpp:866: LOGGER.e(0, "bad index file: " + string(sqlite3\_errmsg(db)) +

Marker.cpp:881: LOGGER.e(0, "bad index file, the indexed SNPs are different from those in bgen file."

Marker.cpp:893: LOGGER.e(0, "bad index file: " + string(sqlite3\_errmsg(db)) +

Marker.cpp:901: LOGGER.e(0, "bad index file: " + string(sqlite3\_errmsg(db)) +

Marker.cpp:924: LOGGER.e(0, "bad index file: " + string(sqlite3\_errmsg(db)) +

Marker.cpp:971: LOGGER.e(0, string(sqlite3\_errmsg(db)) + "\nBad index file. Regenerate it by " + prompt\_index + ".");

Marker.cpp:991: LOGGER.e(0, "The first variant in bgen file is not consistent with that in index file."

Marker.cpp:996: LOGGER.e(0, "The variants in bgen file aren't consistent with those in index file."

Marker.cpp:1066: LOGGER.e(0, "The marker parameters are out of range.");

Marker.cpp:1076: LOGGER.e(0, "can't open bgen file to read");

Marker.cpp:1096: LOGGER.e(0, "strange header length - it might be an invalid bgen file.");

Marker.cpp:1131: LOGGER.e(0, "GCTA only supports bgen version 1.2 and 1.3. Use QCTOOL to convert files with < v1.2 into newer version.");

Marker.cpp:1135: LOGGER.e(0, "non-zlid compressed files are not supported currently");

Marker.cpp:1174: //LOGGER.e(0, "test")

Marker.cpp:1225: LOGGER.e(0, "0 SNP remains for further analysis.");

Marker.cpp:1314: LOGGER.e(0, "0 SNP remains.");

Marker.cpp:1337: LOGGER.e(0, "0 SNP remains.");

Marker.cpp:1406: LOGGER.e(0, "the SNP list file [" + snplist\_file + "], line " + to\_string(line\_number) +

Marker.cpp:1430: LOGGER.e(0, "no " + key\_name + " parameter found");

Marker.cpp:1434: LOGGER.e(0, key\_name + " " + options[key\_store] + " not found");

Marker.cpp:1461: LOGGER.e(0, "Invalid autosome number: " + options\_in["--autosome-num"][0]);

Marker.cpp:1464: LOGGER.e(0, "Multiple values in --autosome-num are not supported currently");

Marker.cpp:1486: LOGGER.e(0, "The chromosome number specified by --chr flag is not within the autosome range (i.e., 1 to 22)");

Marker.cpp:1502: LOGGER.e(0, "The chromosome number specified by --chr flag is not within the autosome range (i.e., 1 to 22)");

Marker.cpp:1521: LOGGER.e(0, "non-numeric value was specified by --chr");

Marker.cpp:1529: LOGGER.e(0, "non-numeric value was specified by --chr");

Marker.cpp:1532: LOGGER.e(0, "multiple --chr flags are not supported currently");

Marker.cpp:1536: LOGGER.e(0, "--chr is out of the chromosome range");

Marker.cpp:1556: LOGGER.e(0, "Marker has no main process this time");

OptionIO.cpp:20: LOGGER.e(0, "no " + key\_name + " parameter found");

OptionIO.cpp:23: LOGGER.e(0, key\_name + " " + options[key\_store] + " not found");

OptionIO.cpp:37: LOGGER.e(0, "no " + key\_name + " parameter found");

OptionIO.cpp:85: LOGGER.e(0, "no item in [" + list\_filename + "].");

OptionIO.cpp:155: LOGGER.e(0, "the number of fields in [" + fileName + "] is less than " + to\_string(minFields) + ".");

OptionIO.cpp:169: LOGGER.e(0, "the file [" + fileName + "] contains different number of elements in line " + to\_string(line\_number) + ".");

OptionIO.cpp:181: LOGGER.e(0, "can't read [" + fileName + "].");

OptionIO.cpp:184: LOGGER.e(0, "found blank header line or can't find valid text data.");

OptionIO.cpp:186: LOGGER.e(0, "text data is different from header.");

Pheno.cpp:73: LOGGER.e(0, "no phenotype file presents");

Pheno.cpp:93: LOGGER.e(0, "found duplicate IDs in phenotype data.");

Pheno.cpp:101: LOGGER.e(0, "a non-numeric value is specified by --mpheno");

Pheno.cpp:106: LOGGER.e(0, "The value specified by --mpheno can't be less than 0 or larger than the total number of columns in .pheno file");

Pheno.cpp:121: LOGGER.e(0, "found duplicate IDs in gender information.");

Pheno.cpp:142: LOGGER.e(0, "found duplicated IDs in sample IDs.");

Pheno.cpp:191: LOGGER.e(0, "0 individual remains for further analysis.");

Pheno.cpp:222: LOGGER.e(0, "can't read [" + sublist\_file + "]");

Pheno.cpp:241: LOGGER.e(0, err\_file + " has less than 3 columns, where the first 2 columns should be FID, IID");

Pheno.cpp:254: LOGGER.e(0, err\_file + " does not have enough columns to read");

Pheno.cpp:258: LOGGER.e(0, err\_file + " has less than 2 columns.");

Pheno.cpp:262: LOGGER.e(0, err\_file + " is empty.");

Pheno.cpp:283: LOGGER.e(0, err\_file + ", line " + to\_string(line\_number) +

Pheno.cpp:351: LOGGER.e(0, "GCTA requires all files have the same sample information.");

Pheno.cpp:374: LOGGER.e(0, "only found " + to\_string(ncol) + " columns, which is invalid for FAM file (at least 6 columns).");

Pheno.cpp:389: if(!foundIID) LOGGER.e(0, "can't find IID or #IID in header, which is invalid for PSAM file.");

Pheno.cpp:393: if(!found) LOGGER.e(0, "SEX column in PSAM file is required by GCTA.");

Pheno.cpp:438: LOGGER.e(0, "invalid PSAM file.");

Pheno.cpp:447: LOGGER.e(0, "cannot open sample file to read.");

Pheno.cpp:466: LOGGER.e(0, "invalid sample file");

Pheno.cpp:469: LOGGER.e(0, "invalid sample file");

Pheno.cpp:494: LOGGER.e(0, "Line " + to\_string(line\_number + 3) + " has different number of columns.");

Pheno.cpp:512: LOGGER.e(0, "cannot open the file [" + fam\_file + "] to read");

Pheno.cpp:524: LOGGER.e(0, "the fam file [" + fam\_file + "], line " + to\_string(line\_number)

Pheno.cpp:604: LOGGER.e(0, "can't write to [" + filename + "].");

Pheno.cpp:966: LOGGER.e(0, "--mpheno has to be used with --pheno");

Pheno.cpp:982: LOGGER.e(0, "Phenotype has no main process this time");

main.cpp:148: LOGGER.e(0, "found multiple options: " + cur\_string);

main.cpp:152: LOGGER.e(0, "the option must start with \"--\": " + cur\_string);

main.cpp:163: LOGGER.e(0, "no output file name is specified in the \"--out\" option");

main.cpp:168: LOGGER.e(0, "missing the --out option");

main.cpp:194: LOGGER.e(0, "can't get the thread number from --thread-num option.");

main.cpp:197: LOGGER.e(0, "can't set the number of multiple threads.");

main.cpp:204: LOGGER.e(0,"can't set the number of threads using both --thread-num and --threads");

main.cpp:211: LOGGER.e(0, "can't get the thread number from --threads option.");

main.cpp:214: LOGGER.e(0, "can't set the number of multiple threads.");

main.cpp:259: LOGGER.e(0, "multiple main functions are not supported currently.");

main.cpp:297: if(mains.size() > 1) LOGGER.e(0, "multiple main functions are not supported currently.");

main.cpp:309: LOGGER.e(0, err\_msg);

main.cpp:311: LOGGER.e(0, string(err\_msg));