Documentation of analysis for AHA abstract currently titled, ”Lipid-related genetic variants and lipid outcomes in a cohort of Chilean children (SLCS)”

Ann Von Holle

February 27, 2017

Contents

1 Statistical Analysis Plan

All analyses below are according to the plan at https://avonholle.github.io/ms-201608-1/StatisticalAnalysisPlan.html.

Note: See the SAP for more detail on variant selection, power, etc..

2 Table 1

Table 1: Descriptive Statistics by Sex

		Female	Male
		$N = 263$	$N = 283$
TG	mmol/l	3.24 4.20 5.50 (4.71±2.14)	3.10 3.99 5.62 (4.80±2.46)
LDL	mmol/l	4.56 5.26 6.11 (5.38±1.24)	4.27 5.02 5.81 (5.11±1.30)
HDL	mmol/l	1.955 2.301 2.724 (2.369±0.586)	1.731 2.053 2.387 (2.116±0.576)
TC	mmol/l	7.65 8.55 9.44 (8.70±1.29)	7.24 7.96 8.89 (8.19±1.37)
Age	years	16.634 16.767 16.934 (16.826± 0.266)	16.583 16.756 16.897 (16.791± 0.243)
BMI	kg/m2	20.85 23.25 26.17 (24.15± 4.71)	20.41 22.31 25.53 (23.46± 4.39)
log(TG)	log(mmol/l)	1.176 1.435 1.704 (1.463±0.406)	1.130 1.383 1.727 (1.466±0.435)

$_a\ b\ _c$ represent the lower quartile a, the median b, and the upper quartile c for continuous variables. $x \pm s$ represents $\bar{X} \pm 1$ SD.

Table 2: Descriptive Statistics by total

		total
		$N = 546$
Sex : Male		52% (283)
TG	mmol/l	3.16 4.12 5.60 (4.75±2.31)
LDL	mmol/l	4.35 5.12 5.97 (5.24±1.28)
HDL	mmol/l	1.830 2.172 2.577 (2.238±0.594)
TC	mmol/l	7.43 8.17 9.24 (8.43±1.36)
Age	years	16.594 16.760 16.921 (16.808± 0.255)
BMI	kg/m2	20.56 22.87 25.75 (23.79± 4.55)
log(TG)	log(mmol/l)	1.150 1.415 1.723 (1.464±0.421)

$_a\ b\ _c$ represent the lower quartile a, the median b, and the upper quartile c for continuous variables. $x \pm s$ represents $\bar{X} \pm 1$ SD.Numbers after percents are frequencies.

2.1 Extra descriptive detail for sample data

```
##
##      ### Summary of continuous variables ###
##
## sex: 0
##       n miss p.miss mean  sd median p25 p75 min max skew  kurt
## bmi 263    0      0   24 4.7     23  21  26  16  42  1.2  1.99
## ldl 263    0      0    5 1.2      5   5   6   2   9  0.3  0.06
## hdl 263    0      0    2 0.6      2   2   3   1   4  0.4 -0.10
## tc  263    0      0    9 1.3      9   8   9   5  12  0.5 -0.06
## ------------------------------------------------------------
## sex: 1
##       n miss p.miss mean  sd median p25 p75 min max skew kurt
## bmi 283    0      0   23 4.4     22  20  26  16  40  1.2  1.7
## ldl 283    0      0    5 1.3      5   4   6   2  10  0.6  0.7
## hdl 283    0      0    2 0.6      2   2   2   1   4  0.9  1.1
## tc  283    0      0    8 1.4      8   7   9   5  13  1.0  1.6
##
## p-values
##           pNormal   pNonNormal
## bmi 7.604706e-02 5.263439e-02
## ldl 1.380209e-02 5.328507e-03
## hdl 4.913534e-07 8.429538e-08
## tc  1.137038e-05 7.600696e-07
##
## Standardize mean differences
##        1 vs 2
## bmi 0.1520434
## ldl 0.2117608
## hdl 0.4359054
## tc  0.3799322
## NULL
```

2.2 Tikkanen results [1] (snapshot from paper)

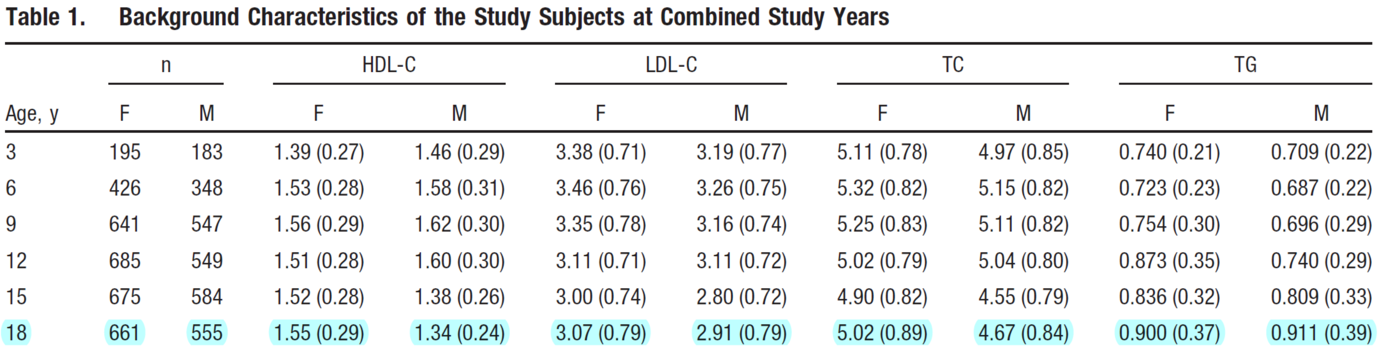


Table 1. Background Characteristics of the Study Subjects at Combined Study Years

	n		HDL-C		LDL-C		TC		TG	
Age, y	F	M	F	M	F	M	F	M	F	M
3	195	183	1.39 (0.27)	1.46 (0.29)	3.38 (0.71)	3.19 (0.77)	5.11 (0.78)	4.97 (0.85)	0.740 (0.21)	0.709 (0.22)
6	426	348	1.53 (0.28)	1.58 (0.31)	3.46 (0.76)	3.26 (0.75)	5.32 (0.82)	5.15 (0.82)	0.723 (0.23)	0.687 (0.22)
9	641	547	1.56 (0.29)	1.62 (0.30)	3.35 (0.78)	3.16 (0.74)	5.25 (0.83)	5.11 (0.82)	0.754 (0.30)	0.696 (0.29)
12	685	549	1.51 (0.28)	1.60 (0.30)	3.11 (0.71)	3.11 (0.72)	5.02 (0.79)	5.04 (0.80)	0.873 (0.35)	0.740 (0.29)
15	675	584	1.52 (0.28)	1.38 (0.26)	3.00 (0.74)	2.80 (0.72)	4.90 (0.82)	4.55 (0.79)	0.836 (0.32)	0.809 (0.33)
18	661	555	1.55 (0.29)	1.34 (0.24)	3.07 (0.79)	2.91 (0.79)	5.02 (0.89)	4.67 (0.84)	0.900 (0.37)	0.911 (0.39)

3 Table 2 results

These SNPS were selected a priori according to the power section in the statistical analysis plan listed in the section 1 above.

Note: All models adjusted for first five PC representing ancestry.

3.1 HDL individual association tests

Notes: Using the additive effect test the **rs3764261** snp is significant (p=7.3e-05) and **rs1532085** is not significant (p = 0.17).

In the Tikkanen paper [1]

```
## Summary for top 10 results, sorted by P1df
##           Chromosome   Position Strand   A1   A2   N      effB    se_effB
## rs3764261         16 2433051751      u    C    A 538 0.1577342 0.03978182
## NA              <NA>         NA   <NA> <NA> <NA>  NA        NA         NA
## NA.1            <NA>         NA   <NA> <NA> <NA>  NA        NA         NA
## NA.2            <NA>         NA   <NA> <NA> <NA>  NA        NA         NA
## NA.3            <NA>         NA   <NA> <NA> <NA>  NA        NA         NA
## NA.4            <NA>         NA   <NA> <NA> <NA>  NA        NA         NA
## NA.5            <NA>         NA   <NA> <NA> <NA>  NA        NA         NA
## NA.6            <NA>         NA   <NA> <NA> <NA>  NA        NA         NA
## NA.7            <NA>         NA   <NA> <NA> <NA>  NA        NA         NA
## NA.8            <NA>         NA   <NA> <NA> <NA>  NA        NA         NA
##           chi2.1df         P1df     effAB     effBB chi2.2df        P2df
## rs3764261 15.72108 7.340143e-05 0.0870423 0.4310441 20.21537 4.07651e-05
## NA              NA           NA        NA        NA       NA          NA
## NA.1            NA           NA        NA        NA       NA          NA
## NA.2            NA           NA        NA        NA       NA          NA
## NA.3            NA           NA        NA        NA       NA          NA
## NA.4            NA           NA        NA        NA       NA          NA
## NA.5            NA           NA        NA        NA       NA          NA
## NA.6            NA           NA        NA        NA       NA          NA
## NA.7            NA           NA        NA        NA       NA          NA
## NA.8            NA           NA        NA        NA       NA          NA
##                  Pc1df
## rs3764261 7.340143e-05
## NA                  NA
## NA.1                NA
## NA.2                NA
## NA.3                NA
## NA.4                NA
## NA.5                NA
## NA.6                NA
## NA.7                NA
## NA.8                NA
```

3.2 LDL individual association tests

Notes: Using the additive effect test the **rs6511720** snp is not significant (p > 0.93).

```
## Summary for top 10 results, sorted by P1df
##           Chromosome   Position Strand   A1   A2   N        effB
## rs4420638         19 2775446584      u    A    G 538  0.48812546
## rs629301           1  109818306      u    T    G 538 -0.17541594
## rs6511720         19 2741225944      u    G    T 538 -0.01349274
## NA              <NA>         NA   <NA> <NA> <NA>  NA          NA
## NA.1            <NA>         NA   <NA> <NA> <NA>  NA          NA
## NA.2            <NA>         NA   <NA> <NA> <NA>  NA          NA
## NA.3            <NA>         NA   <NA> <NA> <NA>  NA          NA
## NA.4            <NA>         NA   <NA> <NA> <NA>  NA          NA
## NA.5            <NA>         NA   <NA> <NA> <NA>  NA          NA
## NA.6            <NA>         NA   <NA> <NA> <NA>  NA          NA
##              se_effB     chi2.1df          P1df       effAB      effBB
## rs4420638 0.13316638 13.436114984 0.0002468262  0.56803498  0.4716794
## rs629301  0.09174878  3.655421541 0.0558870111 -0.26959520 -0.1367078
## rs6511720 0.15758264  0.007331347 0.9317658868 -0.02872452  0.2068510
## NA                NA           NA           NA          NA         NA
## NA.1              NA           NA           NA          NA         NA
## NA.2              NA           NA           NA          NA         NA
## NA.3              NA           NA           NA          NA         NA
## NA.4              NA           NA           NA          NA         NA
## NA.5              NA           NA           NA          NA         NA
## NA.6              NA           NA           NA          NA         NA
##              chi2.2df         P2df        Pc1df
## rs4420638 14.70127090 0.0006421842 0.0002468262
## rs629301   5.35989255 0.0685668377 0.0558870111
## rs6511720  0.08436532 0.9586946487 0.9317658868
## NA                 NA           NA           NA
## NA.1               NA           NA           NA
## NA.2               NA           NA           NA
## NA.3               NA           NA           NA
## NA.4               NA           NA           NA
## NA.5               NA           NA           NA
## NA.6               NA           NA           NA
```

3.3 TG individual association tests

Notes: Using an additive test both the **rs1260326** and **rs964184** snps are significant (p=0.016 and 0.028, respectively).

```
## Summary for top 10 results, sorted by P1df
##           Chromosome   Position Strand   A1   A2   N       effB    se_effB
## rs964184          11 1968684492      u    C    G 537 0.04215448 0.01846388
## rs1260326          2  304085769      u    C    T 537 0.04181070 0.01870433
## NA              <NA>         NA   <NA> <NA> <NA>  NA         NA         NA
## NA.1            <NA>         NA   <NA> <NA> <NA>  NA         NA         NA
## NA.2            <NA>         NA   <NA> <NA> <NA>  NA         NA         NA
## NA.3            <NA>         NA   <NA> <NA> <NA>  NA         NA         NA
## NA.4            <NA>         NA   <NA> <NA> <NA>  NA         NA         NA
## NA.5            <NA>         NA   <NA> <NA> <NA>  NA         NA         NA
## NA.6            <NA>         NA   <NA> <NA> <NA>  NA         NA         NA
## NA.7            <NA>         NA   <NA> <NA> <NA>  NA         NA         NA
##           chi2.1df       P1df       effAB      effBB chi2.2df       P2df
## rs964184  5.212447 0.02242575 0.055305780 0.07363344 5.695439 0.05797639
## rs1260326 4.996786 0.02539443 0.006439567 0.12635164 8.889282 0.01174132
## NA              NA         NA          NA         NA       NA         NA
## NA.1            NA         NA          NA         NA       NA         NA
## NA.2            NA         NA          NA         NA       NA         NA
## NA.3            NA         NA          NA         NA       NA         NA
## NA.4            NA         NA          NA         NA       NA         NA
## NA.5            NA         NA          NA         NA       NA         NA
## NA.6            NA         NA          NA         NA       NA         NA
## NA.7            NA         NA          NA         NA       NA         NA
##                Pc1df
## rs964184  0.02242575
## rs1260326 0.02539443
## NA                NA
## NA.1              NA
## NA.2              NA
## NA.3              NA
## NA.4              NA
## NA.5              NA
## NA.6              NA
## NA.7              NA
```

3.4 TC individual association tests

Notes: Using an additive test the **rs6511720** snp was not significant (p=0.45)

```
## Summary for top 10 results, sorted by P1df
##           Chromosome   Position Strand   A1   A2   N      effB   se_effB
## rs6511720         19 2741225944      u    G    T 538 0.1253222 0.1652135
## NA              <NA>         NA   <NA> <NA> <NA>  NA        NA        NA
## NA.1            <NA>         NA   <NA> <NA> <NA>  NA        NA        NA
## NA.2            <NA>         NA   <NA> <NA> <NA>  NA        NA        NA
## NA.3            <NA>         NA   <NA> <NA> <NA>  NA        NA        NA
## NA.4            <NA>         NA   <NA> <NA> <NA>  NA        NA        NA
## NA.5            <NA>         NA   <NA> <NA> <NA>  NA        NA        NA
## NA.6            <NA>         NA   <NA> <NA> <NA>  NA        NA        NA
## NA.7            <NA>         NA   <NA> <NA> <NA>  NA        NA        NA
## NA.8            <NA>         NA   <NA> <NA> <NA>  NA        NA        NA
##            chi2.1df      P1df     effAB     effBB  chi2.2df      P2df
## rs6511720 0.5753939 0.4481235 0.1123978 0.4490585 0.6258517 0.7313041
## NA               NA        NA        NA        NA        NA        NA
## NA.1             NA        NA        NA        NA        NA        NA
## NA.2             NA        NA        NA        NA        NA        NA
## NA.3             NA        NA        NA        NA        NA        NA
## NA.4             NA        NA        NA        NA        NA        NA
## NA.5             NA        NA        NA        NA        NA        NA
## NA.6             NA        NA        NA        NA        NA        NA
## NA.7             NA        NA        NA        NA        NA        NA
## NA.8             NA        NA        NA        NA        NA        NA
##               Pc1df
## rs6511720 0.4481235
## NA               NA
## NA.1             NA
## NA.2             NA
## NA.3             NA
## NA.4             NA
## NA.5             NA
## NA.6             NA
## NA.7             NA
## NA.8             NA
```

4 Table 3 results

Proportion of variance explained by selected SNPS.

gender	hdl	ldl	tg
female	0.14	0.05	0.12
male	0.20	0.04	0.07

Note: the largest differences between these results and Tikkanen is the LDL (lower proportions for males and females in this sample) and TG (lower proportion for males). Males in this sample also show a higher proportion of variance explained for HDL than the Tikkanen sample.

4.1 Tikkanen results [1] (snapshot from paper)

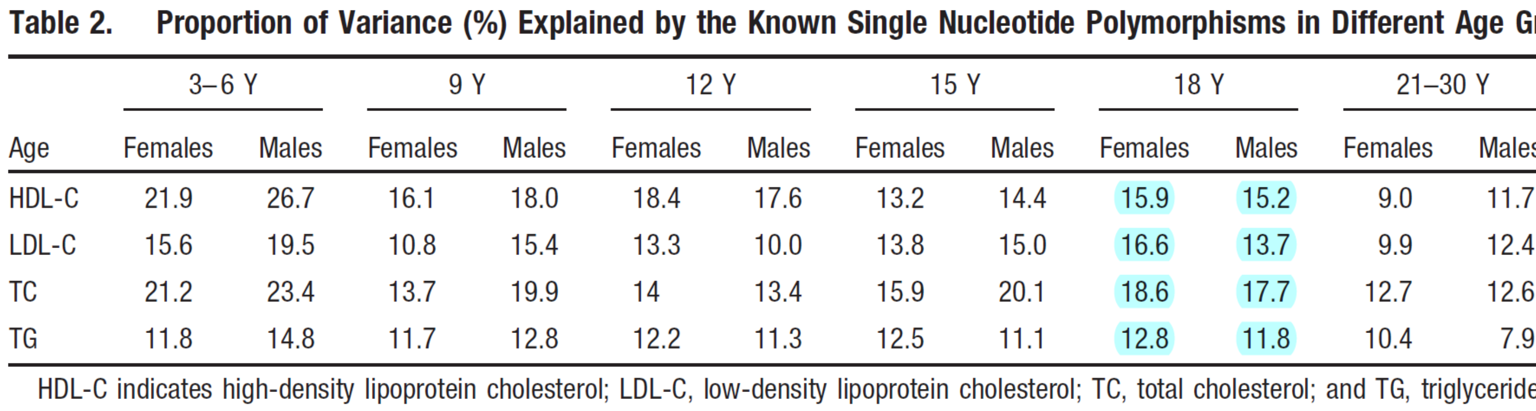


Table 2. Proportion of Variance (%) Explained by the Known Single Nucleotide Polymorphisms in Different Age Gr

	3–6 Y		9 Y		12 Y		15 Y		18 Y		21–30 Y	
Age	Females	Males	Females	Males	Females	Males	Females	Males	Females	Males	Females	Male
HDL-C	21.9	26.7	16.1	18.0	18.4	17.6	13.2	14.4	15.9	15.2	9.0	11.7
LDL-C	15.6	19.5	10.8	15.4	13.3	10.0	13.8	15.0	16.6	13.7	9.9	12.4
TC	21.2	23.4	13.7	19.9	14	13.4	15.9	20.1	18.6	17.7	12.7	12.6
TG	11.8	14.8	11.7	12.8	12.2	11.3	12.5	11.1	12.8	11.8	10.4	7.9

HDL-C indicates high-density lipoprotein cholesterol; LDL-C, low-density lipoprotein cholesterol; TC, total cholesterol; and TG, triglyceride

5 Table 4 results

Note: The polygenic risk score is derived from the Buscot 2016 paper [2], which relies on variants found in the Teslovich 2010 paper [3]. I started with the Tikkanen 2011 [1] paper, but the Buscot paper reduces the number of snps,'To avoid redundancy and overlap of genetic information, in each lipid wGRS, we chose to include only the SNPs with which it showed the strongest independent associations among the 3 lipid traits in the meta-analysis' [2]. More detail available in statistical analysis plan (in section 1).

5.1 Coefficient (se)

Association between the genetic risk score and serum lipid levels, coefficient (se)

	Outcome	Female (no adj)	Female (adj BMI)	Male (no adj)	Male (adj BMI)
1	HDL	0.13 (0.03)	0.13 (0.03)	0.12 (0.04)	0.12 (0.04)
2	LDL	0.16 (0.08)	0.17 (0.08)	0.23 (0.08)	0.24 (0.08)
3	TG	0.34 (0.15)	0.08 (0.02)	0.24 (0.13)	0.05 (0.02)

Note: All models adjusted for first five PC representing ancestry.

Note: The effect sizes here are given for a one unit SD increase in the GRS. The results given in the Tikkanen paper are given for a one unit change in the standard deviation of the GRS.

5.2 p-values

Association between the genetic risk score and serum lipid levels

Outcome	Female (no adj)	Female (adj BMI)	Male (no adj)	Male (adj BMI)
HDL	0.0001	0.0000	0.0010	0.0010
LDL	0.0399	0.0271	0.0032	0.0017
TG	0.0239	0.0006	0.0736	0.0644

5.3 Descriptive statistics for the risk scores

Descriptive statistics for the risk scores

	Female	Male
	$N = 259$	$N = 279$
Risk score, HDL	30.790 33.020 35.230 (33.141± 3.488)	31.115 33.040 35.440 (33.213± 3.393)
Risk score, LDL	35.665 40.190 44.685 (39.957± 6.384)	35.370 39.810 43.810 (39.809± 6.403)
Risk score, TG	127.29 137.68 149.34 (138.84± 17.33)	126.06 138.03 151.24 (138.32± 17.40)

$_{a}\ b\ _{c}$ represent the lower quartile a, the median b, and the upper quartile c for continuous variables. $x \pm s$ represents $\bar{X} \pm 1$ SD.

Note: The genetic risk score medians are very similar to the means listed in the table of Buscot results below.

5.4 Buscot results [2] (snapshot from paper)

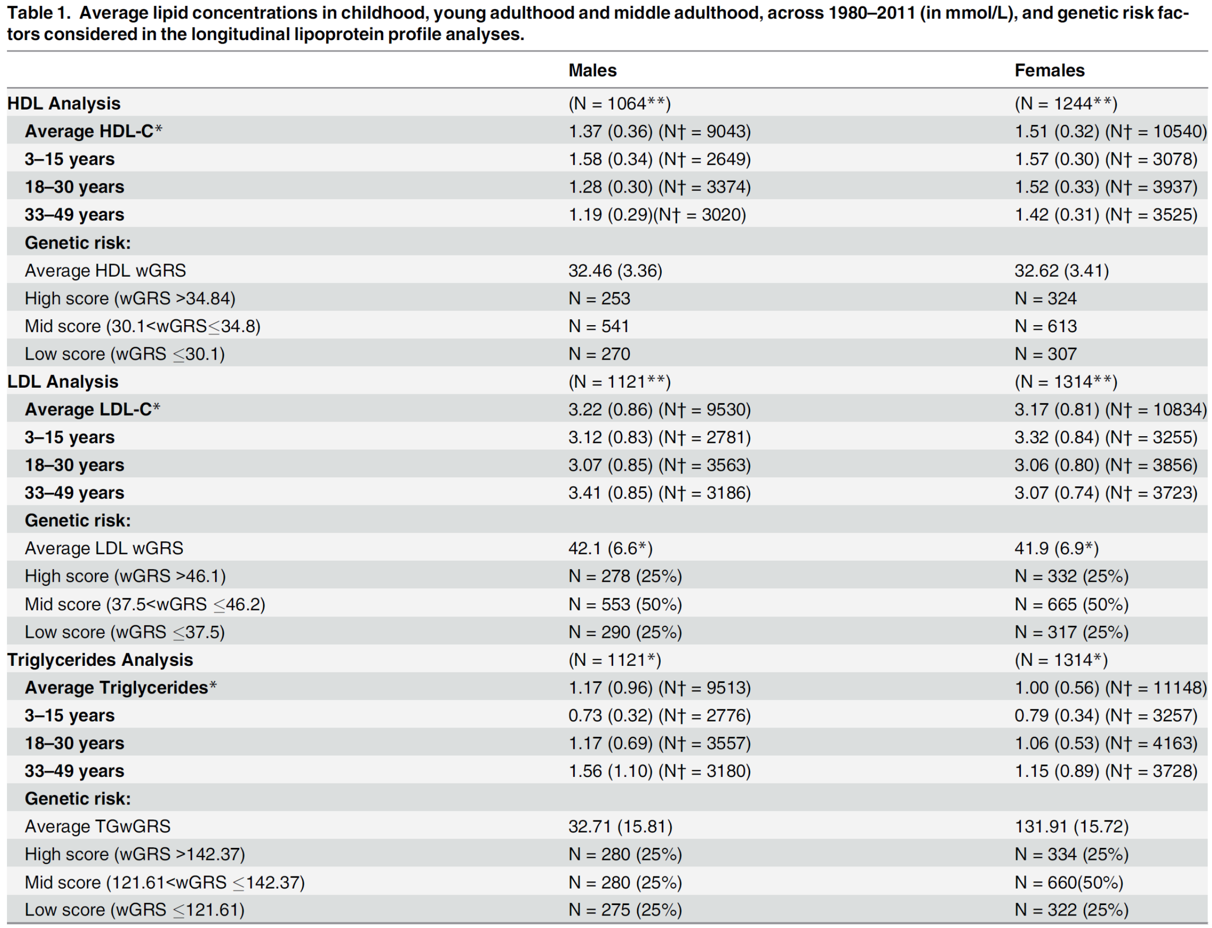


Table 1. Average lipid concentrations in childhood, young adulthood and middle adulthood, across 1980–2011 (in mmol/L), and genetic risk factors considered in the longitudinal lipoprotein profile analyses.

	Males	Females
HDL Analysis	(N = 1064**)	(N = 1244**)
Average HDL-C*	1.37 (0.36) (N† = 9043)	1.51 (0.32) (N† = 10540)
3–15 years	1.58 (0.34) (N† = 2649)	1.57 (0.30) (N† = 3078)
18–30 years	1.28 (0.30) (N† = 3374)	1.52 (0.33) (N† = 3937)
33–49 years	1.19 (0.29)(N† = 3020)	1.42 (0.31) (N† = 3525)
Genetic risk:		
Average HDL wGRS	32.46 (3.36)	32.62 (3.41)
High score (wGRS >34.84)	N = 253	N = 324
Mid score (30.1<wGRS≤34.8)	N = 541	N = 613
Low score (wGRS ≤30.1)	N = 270	N = 307
LDL Analysis	(N = 1121**)	(N = 1314**)
Average LDL-C*	3.22 (0.86) (N† = 9530)	3.17 (0.81) (N† = 10834)
3–15 years	3.12 (0.83) (N† = 2781)	3.32 (0.84) (N† = 3255)
18–30 years	3.07 (0.85) (N† = 3563)	3.06 (0.80) (N† = 3856)
33–49 years	3.41 (0.85) (N† = 3186)	3.07 (0.74) (N† = 3723)
Genetic risk:		
Average LDL wGRS	42.1 (6.6*)	41.9 (6.9*)
High score (wGRS >46.1)	N = 278 (25%)	N = 332 (25%)
Mid score (37.5<wGRS ≤46.2)	N = 553 (50%)	N = 665 (50%)
Low score (wGRS ≤37.5)	N = 290 (25%)	N = 317 (25%)
Triglycerides Analysis	(N = 1121*)	(N = 1314*)
Average Triglycerides*	1.17 (0.96) (N† = 9513)	1.00 (0.56) (N† = 11148)
3–15 years	0.73 (0.32) (N† = 2776)	0.79 (0.34) (N† = 3257)
18–30 years	1.17 (0.69) (N† = 3557)	1.06 (0.53) (N† = 4163)
33–49 years	1.56 (1.10) (N† = 3180)	1.15 (0.89) (N† = 3728)
Genetic risk:		
Average TGwGRS	32.71 (15.81)	131.91 (15.72)
High score (wGRS >142.37)	N = 280 (25%)	N = 334 (25%)
Mid score (121.61<wGRS ≤142.37)	N = 280 (25%)	N = 660(50%)
Low score (wGRS ≤121.61)	N = 275 (25%)	N = 322 (25%)

5.5 Tikkanen results [1] (snapshot from paper)

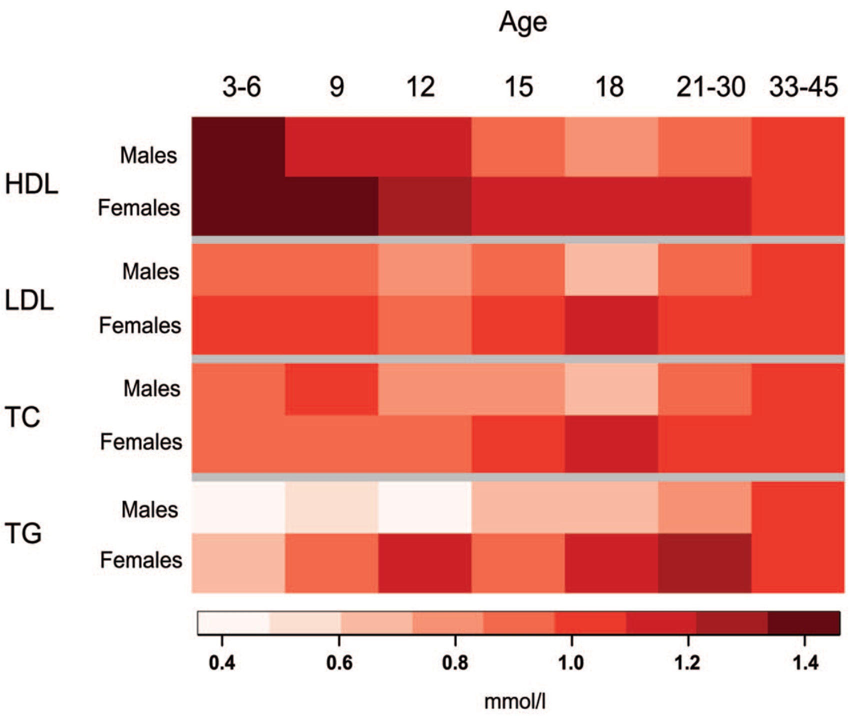
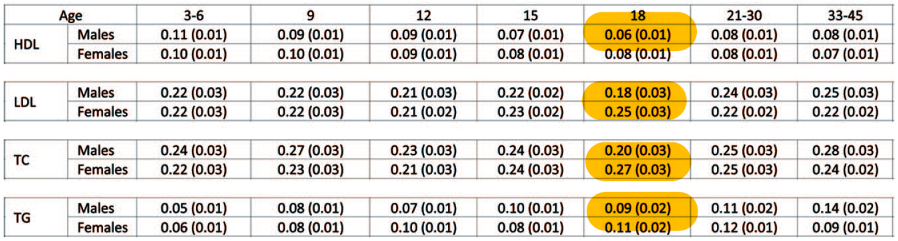


Figure 3. Association between the genetic risk scores and serum lipid levels in age groups. Colors in different age groups correspond to the effect sizes proportional to adulthood effect size. Absolute effect sizes (SD) per 1 SD change in GRS are reported in the table. All *P* values are $<8.4\times10^{-5}$. HDL indicates high-density lipoprotein cholesterol; LDL, low-density lipoprotein cholesterol; TC, total cholesterol; and TG, triglycerides.



Age		3-6	9	12	15	18	21-30	33-45
HDL	Males	0.11 (0.01)	0.09 (0.01)	0.09 (0.01)	0.07 (0.01)	0.06 (0.01)	0.08 (0.01)	0.08 (0.01)
	Females	0.10 (0.01)	0.10 (0.01)	0.09 (0.01)	0.08 (0.01)	0.08 (0.01)	0.08 (0.01)	0.07 (0.01)
LDL	Males	0.22 (0.03)	0.22 (0.03)	0.21 (0.03)	0.22 (0.02)	0.18 (0.03)	0.24 (0.03)	0.25 (0.03)
	Females	0.22 (0.03)	0.22 (0.03)	0.21 (0.02)	0.23 (0.02)	0.25 (0.03)	0.22 (0.02)	0.22 (0.02)
TC	Males	0.24 (0.03)	0.27 (0.03)	0.23 (0.03)	0.24 (0.03)	0.20 (0.03)	0.25 (0.03)	0.28 (0.03)
	Females	0.22 (0.03)	0.23 (0.03)	0.21 (0.03)	0.24 (0.03)	0.27 (0.03)	0.25 (0.03)	0.24 (0.02)
TG	Males	0.05 (0.01)	0.08 (0.01)	0.07 (0.01)	0.10 (0.01)	0.09 (0.02)	0.11 (0.02)	0.14 (0.02)
	Females	0.06 (0.01)	0.08 (0.01)	0.10 (0.01)	0.08 (0.01)	0.11 (0.02)	0.12 (0.01)	0.09 (0.01)

6 Code

6.1 Descriptive statistics for sample

```r
library(Hmisc)
library(tableone)
library(data.table)
library(ztable)

# Note: the next few sections are taken from create-dat1.R

ids = read.csv("IDsToMatch.csv", header=T) # ids from Anne that have the .ped/.fam file ids
colnames(ids) = tolower(colnames(ids))
ids$id_v1 = ids$id # make an id to merge onto the phenotype file
head(ids)


# Get phenotype data file --------------------------------------------
phen = read.csv("slcs-lipid-phen-2016.csv")
colnames(phen) = tolower(colnames(phen))
colnames(phen)

# Determine how many rows with any non-missing lipid data -----------
phen$miss.lip = apply(phen[colnames(phen) %in% c("a_tg",
                                                 "a_ldlc",
                                                 "a_hdlc",
                                                 "a_total_cholesterol")],
                      1, anyNA) # go through each row and indicate if any lipid values are non-missing
summary(phen$miss.lip) # 672 rows have all non-missing lipid data


# merge phenotype to genotype to select people (before .tped generation) -------------------------------------
phen.2a = merge(phen[phen$miss.lip==F,], ids, by="id_v1", all.x=T) # left outer join
colnames(phen.2a)
nrow(phen.2a) # now file only has the 672 individuals with all non-missing lipid values.
head(phen.2a)

phen.2b = phen.2a[!(is.na(phen.2a$fid)==T),]
nrow(phen.2b) # 546 individuals with non-missing fid and lipid values. Keep these individuals for the analyses
head(phen.2b)

# make a phenotype file for use with GenAbel -------------------------

# create .dat file to include for genabel package
phendat = phen.2b[c("iid", "a_sex", "a_tg", "a_ldlc", "a_hdlc", "a_total_cholesterol",
                    "a_age", "a_bmi_derived")]

colnames(phendat)
sapply(phendat,class)
colnames(phendat) = c("id", "sex", "tg", "ldl", "hdl", "tc", "age", "bmi") # re-order names for use in GenAbel
phendat$sex = ifelse(phendat$sex=="male", 1, 0)
table(phendat$sex)

# now create summary data table using Hmisc package
# see http://biostat.mc.vanderbilt.edu/wiki/Main/HmiscSummaryFormulaFunction
summary(phendat)

# for this table convert units from mg/dL to mmol/L (multiply by 0.00555=1/18)
phendat[c("tg", "ldl", "hdl", "tc")] = apply(phendat[c("tg", "ldl", "hdl", "tc")], 2,
                                                     function(x) (1/18)*x)
head(phendat)

# export data to make tables of data on local drive
write.csv(phendat, file="phendat.csv") # write the phendat values to table

phendat.2 = phendat[,-1]
phendat.2$log.tg = log(phendat.2$tg)

x<-upData(phendat.2, # get rid of id for summary table
          labels=c(sex="Sex", log.tg="log(TG)", tg="TG",
                   ldl="LDL", hdl="HDL", tc="TC",
                   age="Age", bmi="BMI"),
          levels=list(sex=c("Female", "Male")),
          units=c(age="years", bmi="kg/m2", log.tg="log(mmol/l)",
                  tg="mmol/l", ldl="mmol/l",
                  hdl="mmol/l", tc="mmol/l"))
contents(x)

#tg="mg/dL", ldl="mg/dL",hdl="mg/dL", tc="mg/dL"

# make table
summary(sex ~ ., data=x, method="reverse")
```

```
myvars = c("sex", "bmi", "log.tg", "ldl", "hdl", "tc")
catvars = c("sex")
myvars2 = myvars[!(myvars %in% catvars)]
tab1 = CreateTableOne(vars=myvars2, data=phendat, strata="sex")

## Warning in ModuleReturnVarsExist(vars, data):  The data frame does not have:  log.tg  Dropped

x$total="total"
```

6.2 Create data set for analysis using GenABEL package in R

Make a data file reading in plink .bed and .fam files from Anne Justice

```
ids = read.csv("IDsToMatch.csv", header=T) # ids from Anne that have the .ped/.fam file ids
colnames(ids) = tolower(colnames(ids))
ids$id_v1 = ids$id # make an id to merge onto the phenotype file
head(ids)

# extract out digits from the tubodna id
# colnames(ids) = tolower(colnames(ids))
# head(ids)
#
# new.ids = do.call('rbind', strsplit(as.character(ids£tubodna),'-',fixed=TRUE))
# head(new.ids)
#ids£id_v1 = as.numeric(new.ids[,2])

# Get PC for ancestry
```

6.3 More data handling

```
load("dat1.Rda") # load dat1 object from create-dat1.R

# quality control (doesn't need quality control because already done.)

qc = check.marker(dat1, maf=0.01, p.lev = 1e-6)
summary(qc)

# clean data

dat1.qc = dat1[qc$idok, qc$snpok]

nids(dat1.qc)
nsnps(dat1.qc)

# complete summaries for clean data

smr <- summary(gtdata(dat1.qc))

# look at data, dat1.qc is same as dat1

dat1.qc@phdata[1:10,]
dat1.qc@gtdata[1:10,1:12]
nids(dat1.qc) # how many people
nsnps(dat1.qc) # how many snps
coding(dat1.qc)[1:25] # coding where the second allele is the 'effect' or 'coded' one (see documentation at ht

refallele(dat1) # check reference allele for all snps in analysis
```

```
ea = effallele(dat1) # check effect allele for all snps in analysis
length(ea) # note that if I used the qc version it omits one SNP. NEED TO FOLLOW UP!!

# check that the effect allele specified in the genotype file is the same as specified in the Tikkanen paper
# ^^^^^^---------------------------------------------

# read in effect allele from tikkanen (see snp-list.Rmd)
eff.a.tikk = read.csv("lipid-snps.csv")
head(eff.a.tikk)
gt.tikk = data.frame(snp=eff.a.tikk$snp2,
                     eff.a.tikk = eff.a.tikk$effect.allele,
                     beta.tikk = eff.a.tikk$beta)
head(gt.tikk)

# make a data frame with genotype data and reference allele/snp rsid
gt.df1 = data.frame(snp=names(ea), gt.eff.a = ea)
head(gt.df1)
nrow(gt.df1)
nrow(gt.tikk)

# merge the two data frames together and determine if effect allele is the same
check.1 = merge(gt.tikk, gt.df1, by='snp')
check.1
nrow(check.1)

# Change direction of beta if the effect allele is different between tikkanen and genotype file
# Note: all effects are positive in Tikkanen paper so they changed the effect allele to match the direction
check.1 = within(check.1, {
  beta.2 = ifelse((as.character(substr(gt.eff.a,1,1))==
                   as.character(substr(eff.a.tikk,1,1)))==T,
                  beta.tikk,
                  -1*beta.tikk)
})



# Now make sure that the direction of these snps now matches that of the Teslovich 2010 paper
# doi:10.1038/nature09270
# I read in table 1 from this paper into phantompdf and saved as a Word file. then cut and pasted
# the word table into excel. Then saved as a .csv file. Re-checked all effect sizes for all snps in csv file
# with the pdf to make sure I have the correct effect sizes as listed in the paper.
# ^^^^^^---------------------------------------------------------------
#setwd("C:/Users/vonholle/Dropbox/unc.grad.school/my-papers/ms-201608-1/programs/kure-analysis/")

tes = read.csv(file="teslovich-snps.csv", header=T)
head(tes)

tes = within(tes, {

  major.allele.tes = substr(Alleles.MAF, 1, 1)
  minor.allele.tes = substr(Alleles.MAF, 3, 3)

  # Match the all positive values from Buscot (meant for polygenic risk score)
  # if the beta is negative then the effect allele will be the major allele after
  # inverse sign (negative to positive)
  e.size.tes = abs(Effect.size)
  e.allele.tes = ifelse(Effect.size>0, minor.allele.tes,
                             major.allele.tes)
  e.size = as.numeric(substr(as.character(tes$Effect.size), 2,
                             nchar(as.character(Effect.size))))
})

check3 = tes[c("Lead.SNP", "e.size.tes", "e.allele.tes")] # Checking here with the table 1 in paper.
check3
colnames(check3)[1] = 'snp'
nrow(check3)

check.4 = merge(check3, check.1, by='snp', all.y=T)
check.4$same.effect.check = ifelse((round(check.4$e.size.tes,2)==round(check.4$beta.2,2))==T, 1, 0)
check.4
nrow(check.4)

# the snps that have different values in Teslovich than the Tikkanan -- due to different effect allele.
# keep the effect size as the Tikkanen effect size because the effect allele is different than what is indicat
# ^^^^^^--------------------------------------------------
check.4[check.4$same.effect.check==0,]

check.4 = within(check.4, {
  effect.fixed = ifelse(same.effect.check==0 & (gt.eff.a==e.allele.tes)==T,
                        e.size.tes,
                        beta.2)
})

check.4[check.4$same.effect.check==0,] # now recheck with Teslovich table 1 to make sure effect propertly matc
# effect allele. it does.


df.fix.e = data.frame(snp=check.4$snp, correct.effect.vec = check.4$effect.fixed) # should have 76 values
df.fix.e

# Now coerce genetic data into data frame I can use to make risk score
# ^^^^^^-----------------------------------------
gt.df = as.numeric(gtdata(dat1.qc))
ph.df = dat1.qc@phdata

head(gt.df)
head(ph.df)

dim(gt.df)
dim(ph.df)

combo.dat = cbind(gt.df, ph.df)
str(combo.dat)


# Get a list of snpnames so I can figure out which ones to select for the GRS
# ^^^^^^--------------------------------------------------------

spn = snpnames(dat1)
save(spn, file="spn.Rda") # Save list of snp names so I can figure out which ones belong in the grs

#setwd("C:/Users/vonholle/Dropbox/unc.grad.school/my-papers/ms-201608-1/programs/kure-analysis")
#load("spn.Rda")

head(spn)
spn # all snps in genotype data
```

```
# rs2126259 is not in the list of snps, but is a snp of interest and in Teslovich and Tikkanen. Why not here?
"rs2126259" %in% spn
"rs12678919" %in% spn
c('rs10321548', 'rs17149780', 'rs386558067', 'rs60484430') %in% spn #these are archive snps from dbsnp that ar




# Explore data ------------------------------------------
# ^^^^^^----------------------------------------------------

# Note: can subset an object of the gwaa.data class by [i,j]: i=index of study subject and j is index of snp
summary(gtdata(dat1[1:3, 1:5]))

# Note: a1=allele frequency for allele 1, a2=allele frequency for allele 2, q.2=coded allele frequencies

perid.summary(dat1[1:10,])


# make a list of snp positions in the dat1 object so I can make the genetic risk score
# ^^^^^^----------------------------------------------------------------------
getwd()

# first get list of snps -----------------------
#setwd("C:/Users/vonholle/Dropbox/unc.grad.school/my-papers/ms-201608-1/programs")

lipids = read.csv("lipid-snps.csv") # get file from snp-list.Rmd program
lipids

lip.ldl = lipids[lipids$trait=="ldl", c("snp2")]
lip.ldl

ldl.logical = (spn %in% lip.ldl); ldl.logical

# position number of ldl snps in the spn object, representing the GenAbel object
ldl.position = match(lip.ldl, spn); length(ldl.position)
ldl.position

ldl.single.position = match(c('rs6511720'), spn)
ldl.single.position

summary(dat1[,ldl.single.position])

hdl.position = match(lipids[lipids$trait=="hdl", c("snp2")], spn)
hdl.position
length(hdl.position)
hdl.logical = (spn %in% lipids[lipids$trait=="hdl", c("snp2")]); hdl.logical

tg.position = match(lipids[lipids$trait=="tg", c("snp2")], spn)
tg.position
length(tg.position)
tg.logical = (spn %in% lipids[lipids$trait=="tg", c("snp2")]); tg.logical


# total number of snps is 76
length(ldl.position) + length(hdl.position) + length(tg.position)


# Save data into .Rda file so I can use it in another program
```

```r
# ^^^^^^----------------------------------------------------------

save(ldl.logical, ldl.position,
     hdl.logical, hdl.position,
     tg.logical, tg.position,
     combo.dat, df.fix.e,
     dat1, file="a1.Rda")

# ldl.position is index number of snps related to ldl risk score
# hdl.position is same as ldl.position but for hdl
# tg.position is same as above
# combo.dat is combined pheno and geno data in a data.frame for risk score analysis
#   Can use index number in vectors above to locate appropriate snps
# df.fix.e is a data frame that contains correct.effect.vec with the correct effect as matched with what is th
# dat1 is the gwaa.data object for use in genABEL (lipid phenotypes and genotype info)
```

6.4 Table 2 data analysis

```r
load("a1.Rda") # load a1 object from a1.R

# ^^^^^^^^^^^^---------------------------------------------
# contents of object -----------------
# ldl.position is index number of snps related to ldl risk score
# hdl.position is same as ldl.position but for hdl
# tg.position is same as above
# combo.dat is combined pheno and geno data in a data.frame for risk score analysis
#   Can use index number in vectors above to locate appropriate snps
# dat1 is the gwaa.data object for use in genABEL (lipid phenotypes and genotype info)
# ^^^^^^^^^^^^---------------------------------------------

descriptives.trait(dat1)
descriptives.marker(dat1)

# Do tests for table 1 shell
# ^^^^^^^^^^^^-------------------------------------------

# get snp subset from power calcs in
# ~/Documents/dissertation/unc-dissertation-markdown/includes/scripts/power-calcs-ind-assoc.html#371_hdl-relat

test.1 = qtscore(hdl.e ~ sex + pc1 + pc2 + pc3 + pc4 + pc5, data=dat1, trait.type="gaussian",
                 snpsubset = c('rs3764261'))

## Warning in qtscore(hdl.e ~ sex + pc1 + pc2 + pc3 + pc4 + pc5, data = dat1, : no. observations < 10;
Lambda set to 1

test.2 = qtscore(ldl.e ~ sex + pc1 + pc2 + pc3 + pc4 + pc5, data=dat1, trait.type="gaussian",
                snpsubset=c(    'rs4420638', 'rs629301', 'rs6511720'))

## Warning in qtscore(ldl.e ~ sex + pc1 + pc2 + pc3 + pc4 + pc5, data = dat1, : no. observations < 10;
Lambda set to 1

test.3 = qtscore(log(tg.e) ~ sex + pc1 + pc2 + pc3 + pc4 + pc5, data=dat1, trait.type="gaussian",
                 snpsubset = c('rs1260326', 'rs964184')) # tg.e is log transformed

## Warning in log(tg.e): NaNs produced
## Warning in qtscore(log(tg.e) ~ sex + pc1 + pc2 + pc3 + pc4 + pc5, data = dat1, : 1 observations deleted
due to missingness
## Warning in qtscore(log(tg.e) ~ sex + pc1 + pc2 + pc3 + pc4 + pc5, data = dat1, : no. observations <
10; Lambda set to 1
```

```
test.4 = qtscore(tc.e ~ sex + pc1 + pc2 + pc3 + pc4 + pc5, data=dat1, trait.type="gaussian",
                 snpsubset=c('rs6511720'))

## Warning in qtscore(tc.e ~ sex + pc1 + pc2 + pc3 + pc4 + pc5, data = dat1, :  no.  observations < 10; Lambda
set to 1
```

6.5 Table 3 data analysis

```
library(GenABEL)
#library(PredictABEL)
library(knitr)
#library(snpStats)
#biocLite("snpStats")
library(data.table)
library(xtable)


##### ^^^^^^^^^^^^^^^- get data -------------

load("a1.Rda") # load a1 object from a1.R

# contents of object -----------------
# ldl.position is index number of snps related to ldl risk score
# hdl.position is same as ldl.position but for hdl
# tg.position is same as above
# combo.dat is combined pheno and geno data in a data.frame for risk score analysis
#   Can use index number in vectors above to locate appropriate snps
# dat1 is the gwaa.data object for use in genABEL (lipid phenotypes and genotype info)

# get sign of correct direction for effects
correct.effect.vec
signs.eff = sign(correct.effect.vec)

descriptives.trait(dat1)
descriptives.marker(dat1)


# Do tests for table 2 shell

# use data.frame for these analyses, combo.dat

# first get all snp names
spn = snpnames(dat1)

#setwd("C:/Users/vonholle/Dropbox/unc.grad.school/my-papers/ms-201608-1/programs/kure-analysis")
#load("spn.Rda")
spn # all snps in genotype data

# ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^______________________________
# Function to run linear regression model to get heritability estimate
# ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^______________________________

get.h = function(df, outcome.position, outcome.left.text){
  # df: data framed
  # outcome.position: vector of positions for snps related to outcome (from a1.R)
  # outcome.left.text: left hand side of equation with outcome variable

  m.null = lm(as.formula(paste(outcome.left.text,
                               "scale(age, center=T, scale=F)")),
                  data=df)

  # now make a formula with all hdl-related variants
  names = spn[outcome.position]
  names

  names = names[!(names %in% 'rs13238203')] # take out this snp name. not in genotype file.}
  # to do:ask about this snp (and the other one)

  m.v =  lm(as.formula(paste(outcome.left.text,
                                 paste0(names, collapse= " + "),
                                 "+ scale(age, center=T, scale=F)")),
                data=df) # note that the sign of effect doesn't matter for r.squared calcs
  summary(m.v)

  # difference in r.squared between null and model with all variants
  summary(m.null)$r.squared
  summary(m.v)$r.squared
  prop.var.exp = summary(m.v)$r.squared - summary(m.null)$r.squared
  return(list(var=list(prop.var.exp), names=list(names)))

}

# create parameters for the function above

dfs = list(combo.dat[combo.dat$sex==1,], combo.dat[combo.dat$sex==0,],
           combo.dat[combo.dat$sex==1,], combo.dat[combo.dat$sex==0,],
           combo.dat[combo.dat$sex==1,], combo.dat[combo.dat$sex==0,])

positions = list(hdl.position, hdl.position,
                 ldl.position, ldl.position,
                 tg.position, tg.position)
texts = list(rep("hdl ~",2),
             rep("ldl ~",2),
             rep("tg ~", 2))

# Run function with all parameters --------------------

list.outcomes = mapply(get.h, df=dfs,
                       outcome.position=positions,
                       outcome.left.text=texts) # apply the function over all these values to get the heritabi

# put all the heritability estimates together into one data frame
vals = lapply(list.outcomes, "[[", 1) #extract out all items from list
#vals[seq(1,12,2)]
#vals

all.h = do.call(rbind.data.frame, lapply(list.outcomes, "[[", 1)[seq(1,12,2)])
head(all.h)

names(all.h) = "h2"
all.h$gender = rep(c("male", "female"), 3)
all.h$outcome = c(rep("hdl",2), rep("ldl",2), rep('tg', 2))

head(all.h)

# Convert from long to wide with outcome in rows and gender as col headers
t2 = dcast(setDT(all.h),
          gender ~ outcome,
          value.var=c("h2"))
t2 # values for table 2 shell


write.csv(t2, file="t2.csv") # write the h2 values to table
```

6.6 Table 4 data analysis

```
library(GenABEL)
#library(PredictABEL)
library(knitr)
#library(snpStats)
#biocLite("snpStats")
library(data.table)
library(Hmisc)


#### >>>>>>>>>>----- get data -------------

load("a1.Rda") # load a1 object from a1.R

# contents of object -----------------
# ldl.position is index number of snps related to ldl risk score
# hdl.position is same as ldl.position but for hdl
# tg.position is same as above
# combo.dat is combined pheno and geno data in a data.frame for risk score analysis
#   Can use index number in vectors above to locate appropriate snps
# df.fix.e is a data frame that contains correct.effect.vec with the correct effect as matched with what is th
# dat1 is the gwaa.data object for use in genABEL (lipid phenotypes and genotype info)

descriptives.trait(dat1)
descriptives.marker(dat1)

# Demo area for risk score calcs ------------------------------------
# Test out how to transform the snps with negative betas to positive and switch allele count for risk score
df2 = c(1,-2,2) # betas
df1 = data.frame(rs1=c(2,2,1),rs2=c(0,1,2), rs3=c(1,1,1)) # risk allele frequencies

vec.neg = (sign(df2)<0); vec.neg # vector indicating which columns/snps have negative betas
vec.pos = (sign(df2)>0); vec.pos

df1.neg = apply(df1[vec.neg], 2, function(x) 2-x) # reverse direction so number of effect alleles matches a po
df1.neg

df1.rev = cbind(df1[vec.pos], df1.neg)
df1.rev = df1.rev[colnames(df1)]

rs.prep = sweep(df1.rev, MARGIN=2, abs(df2), `*`) # multiply each sum of allele frequencies by the vector of n
rs.prep
rs = rowSums(rs.prep)
rs # risk scores for individuals

# End of demo area -----------------------
# >>>>>>>>>>------------------------------------

gt.df = as.numeric(gtdata(dat1)) # risk allele frequencies according to genotype file in dat1, a genabel gwaa
# NOTE: in qc genabel omits one snp. need to follow up
# Function to make weighted risk scores for all variants
# >>>>>>>>>>------------------------------------------------------------
make.risk.scores = function(orig.dat, orig.betas) {

  # get vector of position for negative and positive betas
  vec.neg = (sign(orig.betas)<0); vec.neg # vector indicating which columns/snps have negative betas
  vec.pos = (sign(orig.betas)>=0); vec.pos

  gt.neg = apply(data.frame(orig.dat)[vec.neg], 2, function(x) 2-x) # reverse direction so number of effect al
  head(gt.neg)

  gt.rev = cbind(data.frame(orig.dat)[vec.pos], gt.neg)
  gt.rev = gt.rev[colnames(orig.dat)] # re-order the columns back to original order

  rs.prep = sweep(gt.rev, MARGIN=2, abs(orig.betas), `*`) # multiply each sum of allele frequencies by the vec
  head(rs.prep)
  rs = rowSums(rs.prep)
  head(rs) # risk scores for individuals
  return(rs) # vector of risk scores for individuals
}

# Append risk scores to phenotype data
# >>>>>>>>>>--------------------------------------------------
snp.order = data.frame(snp=names(coding(dat1)))
head(snp.order)
head(df.fix.e)

df.fix.e = df.fix.e[match(snp.order$snp, df.fix.e$snp),] # need the snp order to match the genotype file
cev = df.fix.e$correct.effect.vec; cev
snp.order$snp
rs = make.risk.scores(gt.df, cev)
head(rs)


# get vector of position for negative and positive betas
vec.neg = (sign(cev)<0); vec.neg # vector indicating which columns/snps have negative betas
vec.pos = (sign(cev)>=0); vec.pos
#head(data.frame(gt.df)[vec.neg]) # gt.df is a matrix so need to convert to data frame for indexing to work
#length(colnames(gt.df)) # how many genotypes in the gt data file?

gt.neg = apply(data.frame(gt.df)[vec.neg], 2, function(x)  2-x) # reverse direction so number of effect allele
#head(gt.neg)

gt.rev = cbind(data.frame(gt.df)[vec.pos], gt.neg)
gt.rev = gt.rev[colnames(gt.df)] # re-order the columns back to original order

rs.prep = sweep(gt.rev, MARGIN=2, abs(cev), `*`) # multiply each sum of allele frequencies by the vector of nu
#head(rs.prep)
rs = rowSums(rs.prep)
head(rs) # risk scores for individuals, double check with function


# >>>>>>>>>>------------------------
all.dat = cbind(dat1@phdata, make.risk.scores(gt.df, cev))
head(all.dat)

# >>>>>>>>>>---------------------------------------------
# Now run models
# >>>>>>>>>>---------------------------------------------
# get trait-specific score --------------------

# HDL ----------------------------------

hdl.position

# check
coding(dat1)[hdl.position] # coding where the second allele is the 'effect' or 'coded' one (see documentation
refallele(dat1)[hdl.position] # check reference allele for all snps in analysis
effallele(dat1)[hdl.position] # check effect allele for all snps in analysis

# make the risk score here
gt.df.hdl = gt.df[,hdl.position] # genotype data for hdl
cev.hdl = cev[hdl.position]
cev.hdl

colnames(gt.df.hdl) # check snps

rs.hdl = make.risk.scores(gt.df.hdl, cev.hdl)
summary(rs.hdl)

# LDL ----------------------------------
ldl.position
gt.df.ldl = gt.df[,ldl.position]
cev.ldl = cev[ldl.position]
colnames(gt.df.ldl) #check snps

rs.ldl = make.risk.scores(gt.df.ldl, cev.ldl)
summary(rs.ldl)

# TG  -----------------------------------------------
tg.position
gt.df.tg = gt.df[,tg.position]
cev.tg = cev[tg.position]
cev.tg # check
colnames(gt.df.tg) # check

rs.tg =  make.risk.scores(gt.df.tg, cev.tg)
summary(rs.tg)

# Convert gwaa genabel object to data frame of phenotype data
# >>>>>>>>>>-----------------------------------------------------

phdat = dat1@phdata

# Convert outcome variables to mmol/L (multiply mg/dL by 0.0555 to get mmol/L)
# >>>>>>>>>>------------------------------------------------------

phdat = within(phdat, {
  hdl.e = 0.0555*hdl
  ldl.e = 0.0555*ldl
  tg.e = 0.0555*tg
})

# make lipid specific data frames
all.dat.hdl = cbind(phdat, rs.hdl=rs.hdl)
head(all.dat.hdl)
```

```r
all.dat.ldl = cbind(phdat, rs.ldl=rs.ldl)
head(all.dat.ldl)

all.dat.tg = cbind(phdat, rs.tg=rs.tg)
head(all.dat.tg)


# Run models
# >>>>>>>>>>---------------------------------------------

model.1.female = lm(hdl.e ~ scale(rs.hdl) + pc1 + pc2 + pc3 + pc4 + pc5,
                    data=all.dat.hdl[all.dat.hdl$sex==0,])
summary(model.1.female)

model.1.male = lm(hdl.e ~ scale(rs.hdl) + pc1 + pc2 + pc3 + pc4 + pc5,
                  data=all.dat.hdl[all.dat.hdl$sex==1,])
summary(model.1.male)

model.2.female = lm(ldl.e ~ scale(rs.ldl) + pc1 + pc2 + pc3 + pc4 + pc5,
                    data=all.dat.ldl[all.dat.ldl$sex==0,])
summary(model.2.female)

model.2.male = lm(ldl.e ~ scale(rs.ldl) + pc1 + pc2 + pc3 + pc4 + pc5,
                  data=all.dat.ldl[all.dat.ldl$sex==1,])
summary(model.2.male)

model.3.female = lm(tg.e ~ scale(rs.tg) + pc1 + pc2 + pc3 + pc4 + pc5,
                    data=all.dat.tg[all.dat.tg$sex==0,])
summary(model.3.female)

model.3.male = lm(tg.e ~ scale(rs.tg) + pc1 + pc2 + pc3 + pc4 + pc5,
                  data=all.dat.tg[all.dat.tg$sex==1,])
summary(model.3.male)

# Adjusting for BMI --------------------

model.1.female.bmi = lm(hdl.e ~ scale(rs.hdl) + bmi + pc1 + pc2 + pc3 + pc4 + pc5,
                        data=all.dat.hdl[all.dat.hdl$sex==0,])
summary(model.1.female)

model.1.male.bmi = lm(hdl.e ~ scale(rs.hdl) + bmi + pc1 + pc2 + pc3 + pc4 + pc5,
                      data=all.dat.hdl[all.dat.hdl$sex==1,])
summary(model.1.male)

model.2.female.bmi = lm(ldl.e ~ scale(rs.ldl) + bmi + pc1 + pc2 + pc3 + pc4 + pc5,
                        data=all.dat.ldl[all.dat.ldl$sex==0,])
summary(model.2.female)

model.2.male.bmi = lm(ldl.e ~ scale(rs.ldl) + bmi + pc1 + pc2 + pc3 + pc4 + pc5,
                      data=all.dat.ldl[all.dat.ldl$sex==1,])
summary(model.2.male)

model.3.female.bmi = lm(log(tg.e) ~ scale(rs.tg) + bmi + pc1 + pc2 + pc3 + pc4 + pc5,
                        data=all.dat.tg[all.dat.tg$sex==0,])
summary(model.3.female)

model.3.male.bmi = lm(log(tg.e) ~ scale(rs.tg) + bmi + pc1 + pc2 + pc3 + pc4 + pc5,
                      data=all.dat.tg[all.dat.tg$sex==1,])
summary(model.3.male)
```

```r
# Take results from linear regressions and put in table

list.results = list(m1f = model.1.female, m1m = model.1.male,
                    m2f = model.2.female, m2m = model.2.male,
                    m3f = model.3.female, m3m = model.3.male,
                    m1f.b = model.1.female.bmi, m1m.b = model.1.male.bmi,
                    m2f.b = model.2.female.bmi, m2m.b = model.2.male.bmi,
                    m3f.b = model.3.female.bmi, m3m.b = model.3.male.bmi)

results = lapply(list.results, function(x) summary(x)$coefficients) #extract out coefficients from linear regr

# put all the coefficient estimates together into one data frame
all.est = do.call(rbind.data.frame, results)

all.est

# now output slope coefficients into table
# >>>>>>>>>>-------------------------------------
colnames(all.est)[2] = "std.err"
colnames(all.est)[4] = "pval"
all.est$rn = rownames(all.est)
grepl(c("Intercept|bmi|pc1|pc2|pc3|pc4|pc5") , all.est$rn) # nuisance params

all.est = all.est[!(grepl(c("Intercept|bmi|pc1|pc2|pc3|pc4|pc5") , all.est$rn)), ] # select out slope coeffici

all.est = within(all.est, {
  est.se = paste0(round(Estimate,2), " (", round(std.err,2), ")")
  round.p = round(pval,4)
  bmi = ifelse(grepl(".b.", rn)==T, "yes", "no")
  outcome = ifelse(grepl("m1", rn)==T, "HDL",
                   ifelse(grepl("m2", rn)==T, "LDL",
                          ifelse(grepl("m3", rn)==T, "TG", NA)))
  female = ifelse(grepl("f.", rn)==T, 1, 0)
})

all.est

# Convert from long to wide with bmi in rows and outcome/gender as col headers
t3 = dcast(setDT(all.est),
                    outcome ~ female + bmi,
                    value.var=c("est.se", "round.p"))
t3 # values for table 3 shell

write.csv(t3, file="t3.csv")

# prep t3 to output as table

t3.est = t3[,1:5, with=F] # estimates w/ se
t3.est
cnames.t3 =  c("Outcome",
            "Female (no adj)", "Female (adj BMI)",
            "Male (no adj)", "Male (adj BMI)")
colnames(t3.est) = cnames.t3

t3.pval = t3[,c(1, 6:9), with=F] # p-values
t3.pval
colnames(t3.pval)=cnames.t3
```

6.7 Check coding of SNPS in analyses

```r
library(GenABEL)
#library(PredictABEL)
library(knitr)
#library(snpStats)
#biocLite("snpStats")

# ^^^^^^^^^^^^^-- get data -------------

load("a1.Rda") # load a1 object from a1.R

tes = read.csv(file="teslovich-snps.csv", header=T) # Teslovich effect sizes
head(tes)


tes = within(tes, {

  major.allele.tes = substr(Alleles.MAF, 1, 1)
  minor.allele.tes = substr(Alleles.MAF, 3, 3)

  # Match the all positive values from Buscot (meant for polygenic risk score)
  # if the beta is negative then the effect allele will be the major allele after
  # inverse sign (negative to positive)
  e.size.tes = abs(Effect.size)
  e.allele.tes = ifelse(Effect.size>0, minor.allele.tes,
                        major.allele.tes)

  e.size = as.numeric(substr(as.character(tes$Effect.size), 2,
                             nchar(as.character(Effect.size))))
})

tes.2 = tes[c("Lead.SNP", "e.size.tes", "e.allele.tes")] # Checking here with the table 1 in paper.
colnames(tes.2)[1] = 'snp'


# HDL ----------------------------------

hdl.position # position in snp vector in hdl-related snps

# check
check.1 = function(lp){
  code = coding(dat1)[lp] # coding where the second allele is the 'effect' or 'coded' one (see documentation a
  ref = refallele(dat1)[lp] # check reference allele for all snps in analysis
  eff = effallele(dat1)[lp] # check effect allele for all snps in analysis
  eff.corr = correct.effect.vec[lp]

  dat = data.frame(gt.code=code,
                       gt.ref=ref,
                       gt.eff = eff,
                       snp = names(code))
  dat = merge(dat, tes.2, by='snp', all.x=T)
  dat = merge(dat, df.fix.e, by='snp', all.x=T)
  return(dat)
}

df.hdl = check.1(hdl.position)
df.hdl$outcome = 'hdl'
df.hdl
```

```
# hdl.code = coding(dat1)[hdl.position] # coding where the second allele is the 'effect' or 'coded' one (see d
# hdl.ref = refallele(dat1)[hdl.position] # check reference allele for all snps in analysis
# hdl.eff = effallele(dat1)[hdl.position] # check effect allele for all snps in analysis
# hdl.eff.corr = correct.effect.vec[hdl.logical]
# df.fix.e£correct.effect.vec # check
#
# hdl.dat = data.frame(gt.code=hdl.code,
#                      gt.ref=hdl.ref,
#                      gt.eff = hdl.eff,
#                      snp = names(hdl.code))
# hdl.dat = merge(hdl.dat, tes.2, by='snp', all.x=T)
# hdl.dat = merge(hdl.dat, df.fix.e, by='snp', all.x=T)
# hdl.dat

# LDL ----------------------------------

df.ldl = check.1(ldl.position)
df.ldl$outcome='ldl'
df.ldl

# TG  -----------------------------------------------

df.tg = check.1(tg.position)
df.tg$outcome = 'tg'
df.tg

a.df = rbind(df.hdl, df.ldl, df.tg)
a.df

colnames(a.df)
a.df = a.df[c("snp",
              "gt.code",
              "gt.ref",
              "e.size.tes",
              "e.allele.tes",
              "correct.effect.vec",
              "gt.eff",
              "outcome")] # reorder columns

colnames(a.df) = c("snp",
                   "genotype code",
                   "genotype ref",
                   "Teslo beta",
                   "Teslo effect",
                   "Analysis beta",
                   "genotype effect",
                   "Outcome")

write.csv(a.df,
          file="check-snp-betas.csv")
```

7 Appendix

7.1 Table 2: Check SNPS used in regression

```
# check the snp names used for each outcome
# ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
```

```
names.h = lapply(list.outcomes, "[[", 1)[seq(2,12,2)]
names.h[[1]] # hdl, males

##  [1] "rs4660293"  "rs2814944"  "rs4731702"  "rs2923084"  "rs7134375"
##  [6] "rs7134594"  "rs1532085"  "rs3764261"  "rs2925979"  "rs4148008"
## [11] "rs4129767"  "rs737337"   "rs1800961"  "rs6065906"  "rs1689800"
## [16] "rs4846914"  "rs12328675" "rs2972146"  "rs6450176"  "rs605066"
## [21] "rs1084651"  "rs9987289"  "rs2293889"  "rs581080"   "rs1883025"
## [26] "rs3136441"  "rs4759375"  "rs4765127"  "rs838880"   "rs2652834"
## [31] "rs16942887" "rs11869286" "rs7241918"  "rs12967135" "rs7255436"
## [36] "rs386000"   "rs181362"   "rs13107325"

names.h[[3]] # ldl, males

##  [1] "rs4299376"  "rs3757354"  "rs1800562"  "rs1564348"  "rs11220462"
##  [6] "rs8017377"  "rs6511720"  "rs2479409"  "rs629301"   "rs1367117"
## [11] "rs11136341" "rs7206971"  "rs4420638"  "rs6029526"

names.h[[5]] # tg, males

##  [1] "rs10195252" "rs1042034"  "rs10761731" "rs11613352" "rs11649653"
##  [6] "rs11776767" "rs1260326"  "rs12678919" "rs1495741"  "rs17145738"
## [11] "rs174546"   "rs2068888"  "rs2131925"  "rs2247056"  "rs2412710"
## [16] "rs2929282"  "rs2954029"  "rs439401"   "rs442177"   "rs5756931"
## [21] "rs645040"   "rs9686661"  "rs964184"

names.h[[2]] # hdl, females

##  [1] "rs4660293"  "rs2814944"  "rs4731702"  "rs2923084"  "rs7134375"
##  [6] "rs7134594"  "rs1532085"  "rs3764261"  "rs2925979"  "rs4148008"
## [11] "rs4129767"  "rs737337"   "rs1800961"  "rs6065906"  "rs1689800"
## [16] "rs4846914"  "rs12328675" "rs2972146"  "rs6450176"  "rs605066"
## [21] "rs1084651"  "rs9987289"  "rs2293889"  "rs581080"   "rs1883025"
## [26] "rs3136441"  "rs4759375"  "rs4765127"  "rs838880"   "rs2652834"
## [31] "rs16942887" "rs11869286" "rs7241918"  "rs12967135" "rs7255436"
## [36] "rs386000"   "rs181362"   "rs13107325"

names.h[[4]] # ldl, females

##  [1] "rs4299376"  "rs3757354"  "rs1800562"  "rs1564348"  "rs11220462"
##  [6] "rs8017377"  "rs6511720"  "rs2479409"  "rs629301"   "rs1367117"
## [11] "rs11136341" "rs7206971"  "rs4420638"  "rs6029526"

names.h[[6]] # tg, females

##  [1] "rs10195252" "rs1042034"  "rs10761731" "rs11613352" "rs11649653"
##  [6] "rs11776767" "rs1260326"  "rs12678919" "rs1495741"  "rs17145738"
## [11] "rs174546"   "rs2068888"  "rs2131925"  "rs2247056"  "rs2412710"
## [16] "rs2929282"  "rs2954029"  "rs439401"   "rs442177"   "rs5756931"
## [21] "rs645040"   "rs9686661"  "rs964184"
```

7.2 Check coding of SNPS in analyses

List of SNPS and how coded in analyses

snp	genotype code	genotype ref	Teslo beta	Teslo effect	Analysis beta	genotype effect	Outcome
rs1084651	GA	G	1.95	A	1.95	A	hdl
rs11869286	CG	C	0.48	C	-0.48	G	hdl
rs12328675	TC	T	0.68	C	0.68	C	hdl

rs12967135	GA	G	0.42	G	-0.42	A	hdl
rs13107325	CT	C	0.84	C	-0.84	T	hdl
rs1532085	GA	G	1.45	A	1.45	A	hdl
rs1689800	AG	A	0.47	A	-0.47	G	hdl
rs16942887	GA	G	1.27	A	1.27	A	hdl
rs1800961	CT	C	1.88	C	-1.88	T	hdl
rs181362	CT	C	0.46	C	-0.46	T	hdl
rs1883025	CT	C	0.94	C	-0.94	T	hdl
rs2293889	GT	G	0.44	G	-0.44	T	hdl
rs2652834	GA	G	0.39	G	-0.39	A	hdl
rs2814944	GA	G	0.49	G	-0.49	A	hdl
rs2923084	AG	A	0.41	A	-0.41	G	hdl
rs2925979	CT	C	0.45	C	-0.45	T	hdl
rs2972146	TG	T	0.46	G	0.46	G	hdl
rs3136441	TC	T	0.78	C	0.78	C	hdl
rs3764261	CA	C	3.39	A	3.39	A	hdl
rs386000	CG	C	0.83	C	-0.83	G	hdl
rs4129767	AG	A	0.39	A	-0.39	G	hdl
rs4148008	GC	G	0.42	C	0.42	C	hdl
rs4660293	AG	A	0.48	A	-0.48	G	hdl
rs4731702	CT	C	0.59	T	0.59	T	hdl
rs4759375	CT	C	0.86	T	0.86	T	hdl
rs4765127	GT	G	0.44	T	0.44	T	hdl
rs4846914	AG	A	0.61	A	-0.61	G	hdl
rs581080	CG	C	0.65	C	-0.65	G	hdl
rs605066	TC	T	0.39	T	-0.39	C	hdl
rs6065906	TC	T	0.93	T	-0.93	C	hdl
rs6450176	GA	G	0.49	G	-0.49	A	hdl
rs7134375	CA	C	0.40	A	0.40	A	hdl
rs7134594	CT	C	0.44	T	0.44	T	hdl
rs7241918	TG	T	1.31	T	-1.31	G	hdl
rs7255436	CA	C	0.45	A	0.45	A	hdl
rs737337	TC	T	0.64	T	-0.64	C	hdl
rs838880	TC	T	0.61	C	0.61	C	hdl
rs9987289	GA	G	1.21	G	-1.21	A	hdl
rs11136341	AG	A	1.40	G	1.40	G	ldl
rs11220462	GA	G	1.95	A	1.95	A	ldl
rs1367117	GA	G	4.05	A	4.05	A	ldl
rs1564348	TC	T	0.56	T	-0.56	C	ldl
rs1800562	GA	G	2.22	G	-2.22	A	ldl
rs2479409	GA	G	2.01	G	-2.01	A	ldl
rs3757354	CT	C	1.43	C	-1.43	T	ldl
rs4299376	TG	T	2.75	G	2.75	G	ldl
rs4420638	AG	A	7.14	G	7.14	G	ldl
rs6029526	AT	A	1.39	A	-1.39	T	ldl
rs629301	TG	T	5.65	T	-5.65	G	ldl
rs6511720	GT	G	6.99	G	-6.99	T	ldl
rs7206971	GA	G	0.78	A	0.78	A	ldl
rs8017377	GA	G	1.14	A	1.14	A	ldl
rs10195252	TC	T	2.01	T	-2.01	C	tg
rs1042034	TC	T	5.99	T	-5.99	C	tg
rs10761731	AT	A	2.38	A	-2.38	T	tg
rs11613352	CT	C	2.70	C	-2.70	T	tg
rs11649653	GC	G	2.13	C	2.13	C	tg
rs11776767	GC	G	2.01	C	2.01	C	tg
rs1260326	CT	C	8.76	T	8.76	T	tg
rs12678919	AG	A	13.64	A	-13.64	G	tg
rs13238203	CT	C	7.91	C	-7.91	T	tg
rs1495741	AG	A	2.85	G	2.85	G	tg
rs17145738	CT	C	9.32	C	-9.32	T	tg

rs174546	TC	T	3.82	T	-3.82	C	tg
rs2068888	AG	A	2.28	G	2.28	G	tg
rs2131925	TG	T	4.94	T	-4.94	G	tg
rs2247056	CT	C	2.99	C	-2.99	T	tg
rs2412710	GA	G	7.00	A	7.00	A	tg
rs2929282	AT	A	5.13	T	5.13	T	tg
rs2954029	AT	A	5.64	A	-5.64	T	tg
rs439401	TC	T	5.50	C	5.50	C	tg
rs442177	TG	T	2.25	T	-2.25	G	tg
rs5756931	TC	T	1.54	T	-1.54	C	tg
rs645040	TG	T	2.22	T	-2.22	G	tg
rs964184	CG	C	16.95	G	16.95	G	tg
rs9686661	CT	C	2.57	T	2.57	T	tg

References

[1] E. Tikkanen et al. "Association of Known Loci With Lipid Levels Among Children and Prediction of Dyslipidemia in Adults". In: *Circulation: Cardiovascular Genetics* 4.6 (Dec. 1, 2011), pp. 673–680. ISSN: 1942-325X, 1942-3268. DOI: 10.1161/CIRCGENETICS.111.960369. URL: http://circgenetics.ahajournals.org/cgi/doi/10.1161/CIRCGENETICS.111.960369 (visited on 08/15/2016).

[2] Marie-jeanne Buscot et al. "The Combined Effect of Common Genetic Risk Variants on Circulating Lipoproteins Is Evident in Childhood: A Longitudinal Analysis of the Cardiovascular Risk in Young Finns Study". In: *PLOS ONE* 11.1 (Jan. 5, 2016). Ed. by David Fardo, e0146081. ISSN: 1932-6203. DOI: 10.1371/journal.pone.0146081. URL: http://dx.plos.org/10.1371/journal.pone.0146081 (visited on 08/15/2016).

[3] Tanya M. Teslovich et al. "Biological, clinical and population relevance of 95 loci for blood lipids". In: *Nature* 466.7307 (Aug. 5, 2010), pp. 707–713. ISSN: 0028-0836, 1476-4687. DOI: 10.1038/nature09270. URL: http://www.nature.com/doifinder/10.1038/nature09270 (visited on 08/15/2016).