**Advanced Genetic Risk Scores – Tutorial**

**Prerequisites**

Genetic data in Oxford BGEN+SAMPLE, VCF (imputed dosages) or PLINK format (hard calls)

PLINK software

QCTOOL software (GEN files)

Unix terminal with Bash or similar shell language

AWK

**1. Background**

1. **Genetic risk scores**

A genetic risk score **(GRS)**, polygenic risk score or genomic risk score is a way to quantify the cumulative effect of multiple single nucleotide polymorphisms **(SNPs)** associated with a **polygenic** trait or disease**.** Polygenic means risk is increased by the contribution of many sites or loci in the genome instead of just a mutation (monogenic). Genome wide association studies **(GWAS)** typically use frequentist inference to test the association **(P value)** and effect size **(beta)** of SNPs against a trait or disease of interest *(Figure 1)*.

Chart

Description automatically generated

Figure 1: GWAS of Type 1 diabetes in T1DGC case-control data, p-values>1x10^-300.

Most GRS use a simple logarithmic additive risk model – *sum(risk allele count \* allele beta)* where *allele beta =* *log(allele odds ratio)*. The odds ratio **(OR)** is the most easily “interpretable” statistic, for example an OR of 10 means that the corresponding allele increases your odds of the measured outcome 10-fold over the regular population that has been compared, usually healthy controls but not always. These statistics are generated using linear regression (continuous trait e.g. measured auto antibody levels) or logistic regression (binary trait e.g. Type 1 diabetes yes or no). It is a linear model. *(Figure 2)*

![Table

Description automatically generated]()

Figure 2: Linear log-additive model of genetic risk.

1. **Risk allele math (log additive)**

Risk effects - it is important to note that if we study a locus where the possible allele is, for example, A/T and we have *ORA=10, betaa=log(10)=1* (i.e. the A allele increases your odds 10-fold) then homozygous for the risk allele *betaA/A=2\*betaa=2* and *betaT/T=0,* therefore *betaA/T=1*.

Protective effects – Let’s say we prefer to say “the T allele is protective” rather than “the A allele is risky”, then we can invert our model *ORT=1/ORa=0.1, betaT=-1* or in words, the T allele decreases risk 10-fold over baseline.

This may be hard to conceptualise but whether we choose to code an allele as risk-increasing (risky) or risk-decreasing (protective) is completely arbitrary and has no bearing on the shape of GRS distribution, aside from altering the “raw values”. Typically, when we talk about a protective allele, we are saying this because we have some background knowledge that we expect the biological mechanism is risk-decreasing compared to our control population.

1. **Risk distributions**

Each sample will be assigned a raw number quantifying their risk, this value is arbitrary and specific to the exact risk distribution, it is meaningless outside of the context of the entire risk scale. We can compare values across populations only if we are certain the characteristics of the populations won’t bias our results and that we have generated the exact same model, for example genotyping errors could cause issue. Sometimes we can normalise the values by dividing by the number of alleles included in the model, though much caution is urged to ensure we are comparing the same model and haven’t introduced bias.

Chart, histogram

Description automatically generated

Figure 2: Frequency density plot of a genomic risk score split by case (red) and blue (control), the quantile lines are coloured to each distribution respectively.

1. **Other models of risk**

Log additive models make a key assumption, that the effect scales linearly with the number of alleles, which may not always be true. A prime example of where a linear model is not good is in classic dominant disease where you only need one risk allele to have the maximum risk. Thinking back to our above example if the A allele was dominant, we would want *betaA=2, betaA/A=2, betaA/T=2, betaT/T=0* which is clearly not a linear relationship. We don’t necessarily need a non-linear model though, instead we can change how we represent our genotypes. For allele specific relationships such as dominant alleles we can split our genotypes into additional variables (one-hot encoding), it is not necessary to understand this in detail.

1. **Interaction and interaction terms**

Similarly, we may think that two or more of our SNPs interact to modulate risk in a non-linear fashion, typically these relationships are detected and modelled using a multiplicative model of interaction although alternatives such as exponential models do exist. We call this SNP x SNP or gene x gene interaction **epistasis**. Again, we don’t need a non-linear model, we can simply add “interaction terms” into our linear log-additive model.

Table

Description automatically generated

Figure : Linear log-additive model with SNP x SNP interaction terms.

**This tutorial will explain how to generate log-additive genetic risk scores with additional interaction terms such as that shown above.**

**2. Generating the Type 1 Diabetes Genetic Risk Score 2 (GRS2)**

**This section will explain how to generate the GRS2 published in DiabetesCare (Sharp et al, 2018) which is presently the most discriminative score for identifying Type 1 diabetes (T1D).** The principles however apply to other log-additive risk scores including interaction terms including those published by us (celiac disease CD-GRS) and others.

**A - Background**

The human leukocyte antigen **(HLA)** region of the genome lies on chromosome 6 in 6p21-22 though this notation isn’t helpful for working with genetic data, so it is easier from a bioinformatics perspective to talk about base position. While HLA is often stated as only around 4 million base pairs in length, we often include the flanking region out of interest so the “extended HLA region” lies between bases 28,000,000 - 35,000,000 roughly. The HLA region is unique from the rest of the genome in its complexity and linkage disequilibrium structures which means it is poorly modelled by simple log-additive SNP models without interaction terms. It is very important in autoimmune disease as these genes are directly implicated in immune function.

1. **HLA alleles and what they really mean**

You may have seen HLA alleles represented in any of these forms: DR3, DR3-DQ2, DQ2.5, DR2-DQ2.5 which actually all have the same meaning. I will attempt to explain what an HLA allele is and what the different notations mean but it is okay to continue on if you do not fully understand. In the past HLA alleles were discovered by reacting one patient’s serum with white blood cells from another patient, as different reactions were observed different alleles were documented such as DR3, DR4, DR15 etc. Later molecular study identified genes within these serotypes such as HLA-A, DRB1, DQB1 and subtypes of these alleles were named based on both serological reaction and genomic variation for example DRB1\*03:01 where the 03 represents a DR3 serotype coding an 01 type protein. This can actually expand to 8 digits as we increase resolution to look at coding and non-coding changes at each base within the gene.

HLA-DRB1\*[02] (DR2)

ATATATTTTTTAGTCCCCGTGAGTAAATAGGGCTATTTTAGTCCCCGTGAGTCTGTTAGGGGTCGGTAGTCGTGAGTAAATATTAGTCCCCCCGTAAATATTAGTCGTAAATATTAGTCTAGTCCCCGTGATAGTTTGGCTCTCTACTCTACTACTAGTCGTGAGTAAATCATAGTGTGTAGTCGTGAGTAAATATCCCCCCTACTGTCGTGAGTAAATATGTGTA

At the 2-digit resolution the made-up sequence above we will use as an example, imagine this sequence is found on chromosome 6 on one of the two chromosomes every sample has. The yellow area represents the base pairs that correspond to the DRB1 gene with serotype DR2. These base pairs can vary but the serotype can stay the same.

HLA-DRB1\*02:[01] (DR2)

ATATATTTTTTAGTCCACGTGAGTAAATAGGGCTATTTTAGTCCCCGTGAGTCTGTTAGGAGTCGGTAGTCGTGAGTAAATATTAGTCCCCCCGTAAATATTAGTCGTAAATATTAGTCTAGTCCCCGTGATAGTT

TGGCTCTCTACTCTACTACTAGTCGTGAGTAAATCATAGTGTGTAGTCGTGAGTAAATATCCCCCCTACTGTCGTGAGTAAATATGTGTA

Now at 4-digit resolution looking at the DRB1\*02 sequence we can identify variation that is specific to encoding a protein. The sequence can still vary but we can be sure that we will be coding the same protein for all 02:01 and this will be different from 02:02 etc. Past this level the biological implications are much less, thus for genetic study we normally only care about 4-digit allele types.

Out of interest we’ll go down to 6/8-digit resolution. Let’s say the first part of our sequencing is in a non-coding region and the second part is in a coding region. At 6 digits we’re talking about the coding region being a fixed sequence, at 8 digits we’re talking about the non-coding part as a fixed sequence. You may also see a letter as a 9th digit but this describes changes in expression rather than genomic sequence.

ATATATTTTTTAGTCCACGTGAGTAAATAGGGCTATTTTAGTCCCCGTGAGTCTGTTAGGAGTCGGTAGTCGTGAGTAAATATTAGTCCCCCCGTAAATATTAGTCGTAAATATTAGTCTAGTCCCCGTGATAGTT

TGGCTCTCTACTCTACTACTAGTCGTGAGTAAATCATAGTGTGTAGTCGTGAGTAAATATCCCCCCTACTGTCGTGAGTAAATATGTGTA

HLA-DRB1\*02:01:[01:02]

1. **But what about the different nomenclature?**

Now you hopefully understand the relevance of serotype and what that means in terms of genetic variation you will understand the different nomenclature better. HLA alleles are subdivided by class of protein produced, for example class 1 proteins are encoded by HLA-A,B,C genes and class 2 by HLA-DR, DQ, DP, DO, DM. Class 3 encodes a less important system called the complement system. Biologically a particular section of HLA-DR and DQ has demonstrated the most importance in disease, this is the genes DRB1 – DQA – DQB1.

So when people talk about DR3 in Type 1 diabetes technically they could mean DRB1\*03:XX:XX:XX so any protein, not a fixed sequence just the same serotype. Similarly, DR3-DQ2 tells us a bit more but still isn’t perfect, now we know the DQB part of the chain is serotype 2. Then expanding to DR3-DQ2.5 tells us the DQA part of the chain is serotype 5. Now we know we’re dealing with DRB1\*03:XX – DQA1\*05:XX – DQB1\*02:XX which is a haplotype i.e. a “block” of genetic variation inherited together.

Table : An example of the kind of language you might hear talked about in T1D and what is intended versus what is actually meant.

|  |  |  |
| --- | --- | --- |
| **What you hear** | **What it actually means** | **What they meant** |
| DR3 | DRB1\*03:XX – DQA1\*XX:XX – DQB1\*XX:XX | DRB1\*03:01 – DQA1\*05:01 – DQB1\*02:01 |
| DR3-DQ2 | DRB1\*03:XX – DQA1\*XX:XX – DQB1\*02:XX | DRB1\*03:01 – DQA1\*05:01 – DQB1\*02:01 |
| DR3-DQ2.5 | DRB1\*03:XX – DQA1\*05:XX – DQB1\*02:XX | DRB1\*03:01 – DQA1\*05:01 – DQB1\*02:01 |
| DQ2 | DRB1\*03:XX – DQA1\*XX:XX – DQB1\*02:XX | DRB1\*03:01 – DQA1\*05:01 – DQB1\*02:01 |

What’s the deal with the DQX.X nomenclature? It is a kind of lazy shorthand to describe what’s going on with DQA1 and DQB1, traditional immunogeneticists don’t like it as it’s not consistent at all with their carefully developed naming conventions. You will still see it frequently in genetics publications as bioinformaticians like it because it’s lazy.

Back to type 1 diabetes, fortunately we know from modern genetic study that what all of these different nomenclatures are trying to describe is DRB1\*03:01 – DQA1\*05:01 – DQB1\*02:01. However, people who have been in the field a while will still talk in serotypes such as just saying DR3 as they were taught the serology only and don’t know the genetic implications. *(Table 1)*

1. **Proxy variants and HLA imputation**

So two important points from above to keep in mind (1) most of the biological insight can be determined by 4 digit HLA alleles e.g. DRB1\*02:01 and (2) at 4 digits a lot of the sequence is fixed but not all of it and specific parts of sequence correspond to specific 4 digit alleles e.g 02:01 vs 02:02 etc (c) a haplotype describes a chain of multiple gene types inherited together.

So given a set of SNPs how do we determine 4-digit HLA allele type or even better full haplotypes? We can combine information from many SNPs to infer the HLA allele type, this is HLA imputation such as SNP2HLA, HIBAG, HLA\*IMP etc. Accuracy can be good, but the downside is we then need lots of SNPs just to identify one gene type, that makes it practically really difficult. We also need to determine which chromosome each allele type is on to determine the full haplotypes which we call **phasing** and can be really difficult.

What we can do instead is pull out single SNPs that correlate highly with HLA haplotypes or specific alelles. We call these SNPs proxy variants. Let’s take rs2187668 *(Figure 3),* it tags the haplotype DRB1\*03:01 – DQA1\*05:01 – DQB1\*02:01 with correlation r2=0.98 in white European populations, that’s exceptionally accurate and we only had to genotype a single SNP in a huge block of sequence!

**B – Method**

1. **Checking SNPs are present and extracting them**

You have 3 basic considerations to start with

* Are the SNPs I need present in my data set and what should I do if they aren’t?
* Are the SNPs present well-imputed enough to use?
* Are my palindromic SNPs coded correctly?

Are the SNPs I need present in my data set and what should I do if they aren’t

We use GEN so I will describe the procedure using QCTOOL. If you are not comfortable with this or have VCF files you can use other tools or shell command “grep” which will be very slow but will work.

1. Make a tab delimited table with “SNPID RSID chromosome position A1 A2” as the columns and add your SNPs, you can download DBSNP as a text file from the UCSC ftp. This way you can use your preferred coding language to look up the SNPs by RSID and pull out the corresponding position and alleles. The SNPID column can vary but please check your GEN file to figure out exactly how it looks. If you have weird RSIDs then sometimes it’s easier to just replace the RSID in the GEN file with CHR:POSITION:A1:A2 and you can be sure QCTOOL will match it
2. Use QCTOOL -incl-variants flag along with the table of SNPs to extract the SNPs.
3. Check everything looks right and you have the right number of SNPs in a single file.

If you are missing SNPs then you need to find alternative SNPs (proxies). The SNPs were discovered using a combination of 1000Genomes, HRC, UK10K and T1DGC (HLA) reference panels, if you have imputed to the TOPMED panel then you will likely not have any issues. If you have imputed to 1000Genomes or similar, then you will likely be missing some SNPs. I’ve provided a list of alternative SNPs that are in 1000Genomes reference panel in the file “T1D\_GRS67\_1000G\_nopalin\_pos\_hg19.xlsx”. Alternatively, you can use the NCI LD-proxy tool found at <https://ldlink.nci.nih.gov/?tab=ldproxy> but your input SNP must be in 1000Genomes.

Are the SNPs present well-imputed enough to use

If you have the metrics from your imputation, then you can simply look up the GRS2 SNPs and find the INFO(R2) column. I recommend INFO>0.7 for HLA DR-DQ SNPs and INFO>0.4 for the rest. If the SNPs aren’t well imputed, you will also need to replace them with proxies as above. If they are poorly imputed and there is no suitable proxy - for DR-DQ add a column for the corresponding haplotype with all 0, for non DR-DQ that’s okay you can still include them as \*some\* information is better than none.

Are my palindromic SNPs coded correctly

A palindromic SNP is a SNP that has the alleles A/T or G/C, this matters because when we read SNP variation sometimes we read the positive strand and sometimes the negative. For example, if we read a G/A SNP on the positive strand then reading the negative strand it would be a C/T SNP (as A->T G->C and vice versa). With SNP array data it is common to have different SNPs read on different strands and not know which is correct. This presents a problem with A/T or G/C SNPs as inverting them to the other strand they will still be A/T or G/C.

You can keep these SNPs in (I recommend at least trying) or replace them with the alternative in the 1000Genomes non-palindromic spreadsheet described 2 paragraphs up. If you keep them in you need to take GREAT CAUTION checking the frequencies in your cases and controls match up with the scores you are assigning, for example a positive beta SHOULD match to the allele more common in your cases than in your controls. On rare occasions this might not be wise - if the characteristics of your cohort may bias the allele frequencies or if the frequencies are close to 0.5 for both alleles.

1. **Using tag SNPs to call HLA haplotypes in DR-DQ**

In section B(iii) above I described the use of proxy variants to mark the presence of specific HLA haplotypes. This is a really important concept in the GRS2. We use 14 SNPs to tag corresponding haplotypes that are evidenced to be important in T1D risk. Firstly, we need to extract these SNPs, then call the haplotypes using the SNPs on a sample-by-sample basis and finally assign a beta score for their genotype to each sample.

Table : HLA DR-DQ haplotypes and their corresponding tag SNPs used in the interaction part of the model.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Haplotype** | **DQA1** | **DQB1** | **Freq\*** | **Marker\*\*** | **r2** |
| DQ8.1 | 03:XX | 03:02 | 0.154 | rs9275490 | 0.987 |
| DQ6.2 | 01:XX | 06:02 | 0.1437 | rs17843689 | 0.968 |
| DQ2.5 | 05:01 | 02:01 | 0.1314 | rs9273369 | 0.998 |
| DQ2.2 | 02:01 | 02:02 | 0.1108 | rs17211699 | 0.985 |
| DQ7.5 | 05:05 | 03:01 | 0.1106 | rs9469200 | 0.975 |
| DQ5.1 | 01:01 | 05:01 | 0.1088 | rs10947332 | 0.994 |
| DQ7.3 | 03:XX | 03:01 | 0.0648 | rs1281935 | 0.882 |
| DQ6.3 | 01:03 | 06:03 | 0.0567 | rs62406889 | 0.968 |
| DQ9.2 | 02:01 | 03:03 | 0.0366 | rs28746898 | 0.955 |
| DQ4.2 | 04:01 | 04:02 | 0.0227 | rs12527228 | 0.956 |
| DQ5.3 | 01:XX | 05:03 | 0.0208 | rs1794265 | 0.990 |
| DQ9.3 | 03:02 | 03:03 | 0.0079 | rs9405117 | 0.956 |
| DQ6.9 | 01:02 | 06:09 | 0.0071 | rs16822632 | 0.954 |
| DQ6.1 | 01:03 | 06:01 | 0.0066 | rs117806464 | 0.986 |

Assuming you have all of your SNPs extracted in a GEN or VCF file with imputed dosages you will need to pull only the 14 SNPs above (or your replacement SNPs) out into a separate file using the relevant tool (see section (i) above). Now run the convert\_gen\_to\_single\_A1\_dosage.sh shell script to convert your GEN file into a table of single dosage columns. It should look like below.

HLA haplotypes are either present or not, so we can’t be dealing with dosages, we need to round the dosage values. You may come across “NA” entries for missing genotypes, replace these with 0. Use your preferred scripting language to iterate over the output and round the variables.

The variables are rounded but as you can see above we have a whole lot of 2’s for each sample, but that’s physically impossible, you only have 2 of each chromosomes so you can only have 2 HLA haplotypes. We’ve got the wrong alleles coded in our dosage – but that’s OK. By default haplotypes should be rare (more 0’s than 2’s in each column) but you might have to take particular caution if your population was selected based on HLA. Use your preferred scripting language to flip the columns. You can invert the dosage to the other allele by using

*dosage(A2)=–dosage(A1))+2*

Now you should have a table that looks like this.

Go ahead and change the genotype names to match the haplotype each is tagging for example rs9275490 becomes “DQ81” omitting the dot.

It’s time for a bit of a quality check, use your preferred scripting language to check the sum of each row is in most cases <=2. Having more than 2 HLA alleles is physically impossible, however, remember the SNPs are not in perfect correlation so there may be a handful of samples where this is the case. If it’s many people then something has gone wrong in the previous steps. If it’s only a few, this is OK and we can remove the least likely haplotype dosage. You can do this manually using both the frequency and r2 values in the table above, if it’s rare and or not well tagged it’s less likely to be correct. This is more common in non-White populations and if in this case you have a fair few and don’t suspect an error then use one of the many great HLA allele frequency databases to figure out what is most likely for your particular cohort. You can also just drop these people out the same way you would with missing genotypes as long as you keep note of it for any future publication.

The next step is to process this into a list of haplo-genotypes for each sample which you can do with the script X.sh which should output the following

1. **Generating and combining the scored components**

For the DR-DQ interaction component simply apply the script score\_with\_awk.sh to the list of haplo-genotypes you previously generated.

For other HLA and non-HLA you can simply use PLINK with the --score flag. It’s also useful to note if you want a marginally more accurate score you can load dosage values into PLINK with:

--dosage <dosage\_file.txt> --fam <fam file> -- score <scorefile.txt> --double-dosage

For more information on how to use any of these commands refer to PLINK documentation.

Now you have a .profile file from PLINK and a list of DR-DQ scores use your preferred scripting language to combine the two scores. This is the complete GRS2.