README

Jessica Bonnie

10/06/2015

IMCHIP

Throughout this **README**, the calls to the code in this directory will be proceeded with `${imchipfolder}` which refers to the path to this folder, currently set to the following location with this line:

```
imchipfolder=/h2/t74/cphgdesk/share/cphg_OnengutLab/Jessica/Farewell/IMCHIP
```

Cleaning Immunochip data requires the following scripts:

- `check_export.sh`
- `makedata.sh`
- `pheno_inc.sh`
- `qc1.sh`
- `relatedness_qc.sh`
- `structure.sh`
- `gender.R`
- `rawrel_relat.R`
- `rawrel_imchip.R`
- `projpca_plus.R`
- `projpca.R`
- `qc2_imchip.sh`

I have also included two scripts from other Immunochip projects which call all of the scripts described:

- `caroline_20150817.sh`
- `trialnet_20150407.sh`

Here are the steps for cleaning data from the Immunochip array.

Preparing the Raw Data

1. **check_export.sh** – this script checks the integrity of the genome studio export. It is called like so:

```
bash ${imchipfolder}/check_export.sh ${DATAFILE} ${tracking} "Top Alleles"
```

where `${DATAFILE}` is the path to the raw export in the cphgcore, `${tracking}` is the path to the sample tracking sheet, and "Top Alleles" refers to the suffix of the columns from the export which will be used to create the plink files.

2. **makedata.sh** – this script produces a non-phenotyped raw plink file from the genome studio export. It is called like so:

```
bash ${imchipfolder}/makedata.sh ${DATAFILE} ${plink_nopheno} "Top Alleles" > ${logfile}
```

where `${DATAFILE}` is the path to the raw export in the cphgcore, `${plink_nopheno}` is the name/path where the output PLINK file should be written, and "Top Alleles" refers to the suffix of the columns from the export which will be used to create the plink files. If you wish to record a log of the process you can designate a `${logfile}`, otherwise remove the redirection `>` to it in every command.

3. If there are duplicate individual IDs PLINK will complain when it is called during the last script. If that happens, PLINK's log file will provide a list of those IDs. The duplicates *must* be renamed before proceeding. Bad things will happen otherwise.

```
grep "Duplicate individual found:" ${plink_nopheno}.log | sed 's/Duplicate individual found: \[ //g' |\
     sed 's/ \]//g' | cut -d' ' -f1 | awk '{print $1,$1,$1"_2",$1"_2"}' > duplicate_update.txt

plink --update-ids duplicate_update.txt --bfile ${plink_nopheno} --out ${plink_nophenoB} --make-bed --n
```

4. Now the phenotype information must be added. There is a small hitch about how it is done, because later scripts will expect the existence of certain files that can be easily produced by using the **pheno_inc.sh** script. This script will only add STATUS and SEX information to the plink file which it is given (make sure the column names are recognizable in the phenotype file.) The "covariable count" is a number indicating how many columns will be of interest for coloring graphs later (Cohort is the first one, and is standard; the next is Race. The titles are hardcoded into the script, but can be easily changed to match the title of the columns in the phenotype table.)

```
bash ${imchipfolder}/pheno_inc.sh ${phenofile} ${plink_nophenoB} ${plink_raw} ${covariablecount} T >> $
```

where `${phenofile}` is the phenotype file (with "Sex" and "Status" and "SampleID" columns – n.b. both columns must contain numeric values, `${plink_nophenoB}` is the input PLINK file, `${plink_raw}` is the name of the output PLINK file, `${covariablecount}` is the number of covariables in the file (recall that the list of titles is hardcoded, so if there is a covariable other than "Cohort", the column title must be put into the list), and the "T" indicates that there is status information to be incorporated.

5. There will also need to be a color file in the project folder (which will contain the folders created by the scripts). You can write one with this line:

```
echo "red blue limegreen purple orange magenta purple4 deepskyblue peru yellow chartreuse4 steelblue lau
```

6. Phenotype information that was not included through the **pheno_inc.sh** script can be added using PLINK. See example scripts for more detail.

QC

Once the raw data is complete with phenotypic information, the QC can begin.

1. The first script is **qc1.sh**. It can be run like so. Some of the output files need to be moved because the other scripts are a bit brittle:

```
bash ${imchipfolder}/qc1.sh ${plink_raw} ${nickname} F > ${logfile2}
cp ${plink_raw}.covariable ${project_folder}/QC1/${nickname}.cov
cp covariables1.list ${project_folder}/.
```

where `${nickname}` is a short alias that will be used to name the files, `${plink_raw}` is the raw PLINK file with all phenotype information included.

2. **relatedness_qc.sh** must be run in order to calculate relatedness between the samples.

```
bash ${imchipfolder}/relatedness_qc.sh ${nickname} ${covariablevalue} I >> ${logfile2}
```

where `I` indicates that data was run on the Immunochip.

4. **qc_pdf.sh** is run to illustrate the findings of **qc1.sh** and **relatedness_qc.sh**. Please note that any script that draws graphs will be calling R and an embedded R script. It is possible that your path to that script will be different from mine, so if the script fails, you should check that the path to the R script is correct. You can call **qc_pdf.sh**

```
bash ${imchipfolder}/qc_pdf.sh ${overall_title} ${nickname} "IMCHIP" ${covariablevalue} >> ${logfile2}
```

where `${overall_title}` is a title to be added to the graphs and `${covariablevalue}` refers to the number of the covariable to be graphed (according to the hard coded list in **pheno_inc.sh**. e.g. if you want to color by Cohort, the covariablevalue would be 1.)

5. **gender_basic.R** is also included. It will draw a gender graph for whatever population it is given using KING –bySample output.

```
R CMD BATCH "--args ${bySample}  ${nickname} ${overall_title} TRUE" ${imchipfolder}/gender_basic.R
ps2pdf ${nickname}gender.ps ${nickname}gender.pdf
```

where `${bySample}` is most likely *QC1/*nickname4bySample.txt*, ‘{overall_title}`is the project name that should be included on the graph, and`TRUEorFALSE‘ indicates whether the boundary lines should be drawn.

6. **structure.sh** looks at population structure and draws graphs. Please see note regarding R scripts location in #4.

```
bash ${imchipfolder}/structure.sh ${nickname} ${overall_title} ${chip} ${covariablevalue} >> ${logfile2
```

7. **qc2_imchip.sh** produces a graph and a list of SNPs which are in Hardy-Weinberg Disequilibrium in the European population. This list will need to be added to the one produced by qc1.sh in order to produce a complete list of SNPs to be removed.

```
bash ${imchipfolder}/qc2_imchip.sh ${nickname} ${overall_title} ${chip} ${covariablevalue} >> ${logfile
```

This script will write a list of SNPs to be removed because of Hardy-Weinberg Disequilibrium to a file in the QC2_HWE folder: *QC2_HWE/hweSNP.txt*. It will need to be appended to the original list of SNPs to be removed from **qc1.sh**.

```
cat QC1/snptoberemoved.txt > release_snptoberemoved.txt

##To reduce confusion, Monomorphic2 will be recoded to Monomorphic
sed -i 's/Monomorphic2/Monomorphic/g' release_snptoberemoved.txt

## Now include HWD snps
awk '{print $1, "HWDinEUR"}' QC2_HWE/hweSNP.txt >> release_snptoberemoved.txt
```

8. If the SNP map needs to be updated to hg19, the included imchip_mapupdate.txt file can be used to update the SNPs. This file was derived using the steps in the /h2/t74/cphdesk/share/cphg_OnengutLab/Jessica/Farwel folder.

```
plink --noweb --bfile ${plinkraw} --update-map ${imchipfolder}/imchip_mapupdate.txt --make-bed --out ${
```