Genome-wide association study (GWAS) is a methodology that focus on the associations between single nucleotide polymorphisms (SNPs) and a phenotypic trait (or traits). In brief, a large cohort of individuals are characterized by their SNPs makeup using SNPs arrays (DNA microarray), and the collected data are used to test if specific SNP or cluster of SNPs, correlate with a specific phenotypic characteristic, for example a disease or his predisposition. This approach has been very successful so far to identify thousands of SNP association with multiple diseases (1). Despite GWAS does not give specific information about the gene (or genes) involved in a phenotype it offers a broad overview of potentials targets loci associated with the trait(s) that can be eventually screened afterword. This is particularly useful when the studied trait (or disease) is the result of multiple gene products and interactions that cannot be easily explained using classical approaches. The most common setup in GWAS is the case-control study, which compare the SNPs of 2 groups, for example healthy controls vs disease affected individuals. The odds of disease for individuals having a specific allele and the odds of disease for individuals who don t have the same allele are then express as a ratio and a P-value significance is derived by a simple chi-square test. The P-values are then plot as negative log of the P value creating the well know **Manhattan plot** where the most significant SNPs associations stand out on the plot. The conventional significance p value threshold is **$5x10^{-8}$**. There are multiple variations at the case-control setup in GWAS, for example quantitative phenotypic data can be used as traits (for example height or weight), gene expression, and biomarkers (for example blood or urine metabolites) (1-5).

This last one concern this specific Case study in bioinformatic (6).

The CoLaus cohort was used for the association study between SNPs genotyping and H-NMR urine metabolic profiles (metabolome). Indeed, the concentration and variety of small molecules (metabolites) in the urine is the result of the complex relationships between genetic makeup, environment and metabolism of each individual. The goal of metabolome-wide genome-wide association studies is to discover how genetic variation affects metabolome phenotypes. Those studies can at the end uncover clinical phenotypes and eventually predict disease progression.

For this study the genotyping was performed using the Affymetrix GeneChip Human Mapping 500 K array set. Genotypes were called using BRLMM software and PLINK was used to detect and correct for population stratification, and only individuals with call rate over 90% were included. The software IMPUTE 0.2.0. was used for genetic imputation (to deduce unknown genotypes based on known ones). Expected allele dosages were computed for 2,557,249 SNPs. A total of 835 individuals were part of the study.

H-NMR spectra for the urines were acquired using a Bruker Avance II 700 MHz spectrometer. Spectra were split in bins of chemical shift increments of 0.005 ppm (as average intensity per bin), resulting in metabolic profiles of 2,200 **metabolome features**. Those were filtered removing features with more than 5% of missing values. Bin intensities were log-averaged across replicate samples for each individual. For each individual, it was applied a Z-score transformation in order to achieve zero mean and unit variance.

In this study were first identified the genetic variants that correlates with any of the metabolome features (**untargeted approach**) and after was derived a profile of significance for association with all features.
Then a new method was used to identify the underlying metabolites: **metabomatching**.

The NMR spectrum for most metabolites presents multiple peaks, and the genetic effect of a SNP on a metabolite usually results in associations of that SNP with multiple metabolome features. It is then generated a **pseudo-spectrum** of a SNP, consisting of a set of significant P values (−log(P-values)) for its associations with each feature. When genetic effect (association) is strong enough, the pseudo-spectrum is quite similar to the NMR spectrum for the underlying metabolite, allowing eventually its identification (6, 7).

At the end, for each SNP association, metabomatching assigns scores to all genetically associated metabolites with known NMR spectrum. The scores are computed using the significance values of the features that correspond to peaks in the known spectra. The metabolites are then ranked, based on these scores, to identify the possible candidate metabolites.

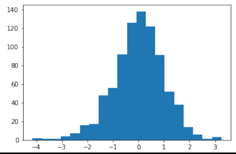
This approach was very successful to identify new SNPs associations and confirm previously know ones, suggesting it potential utility in any GWA-metabolome association studies (6, 7).

Results:

Code :	*Comment :*
Import Modules	
import pandas as pd; import matplotlib.pyplot as plt import scipy.stats as stats import numpy as np %matplotlib inline	• *pandas as pd for file I/O* • *numpy as np for scientific computing* • *scipy.stats as stats for statistical functions, and* • *matplotlib.pyplot as plt for*
Data Import: *Import metabolome data and verify their consistency*	

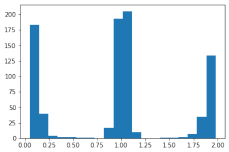
i1 = pd.read_csv('inpf.metabolome.csv') # i1: input 1 - metabolome
metabolome = i1.values
	Load data for HNMR using panda pd. The metabolome file has multiple columns (1017 =metabolite features) and multiple rows (835= number of individuals). The header of the columns is the metabolome feature express in ppm (chemical shift 0.005 ppm) after data filtration. Along each single column the MNR data for the specific feature for each individual (835). The MNR data are saved as metabolome variable (as a data frame).

feature_ppms_tmp__strings = i1.columns.tolist()
feature_ppms_tmp__map = map(float,feature_ppms_tmp__strings)
feature_ppms = list(feature_ppms_tmp__map)
print("Number of features: " + str(len(feature_ppms)))
print("Chemical shifts ranging from: " + str(min(feature_ppms)))
print("to: " + str(max(feature_ppms)))
 	
Number of features: 1017 Chemical shifts ranging from: 0.7025 to: 8.8425
	The column.tolist get from pandas DataFrame column headers; map apply the function float to all items and then the items are saved as a list. In the end the columns header is saved as feature ppms variable (as a list). Some info about it are printed out (number of features (spectra bins), min and max).

metabolome_feature = metabolome[:,3]
plt.figure()
plt.hist(metabolome_feature,20)
	Checking normally distribution of the data. Necessary for running linear regression later. In this case feature 3 and bins size 20.
	

Data Import: *Import genome SNPs data and verify their consistency*	

i2 = pd.read_csv('inpf.genome.csv')
genome = i2.values
snp_of_interest_name = 'rs2287921'
snp_of_interest_index = 1
	Load data for genotyping using panda pd. The file has 6 columns and the first top element of each column contains the rs number, which identifies the genetic variant (SNP); subsequent elements contain the corresponding genotype. Only a part of the whole data is shared for this exercise. Indeed, having the whole SNPs per each individual would allow us students to pin point any donor and this is not contemplated in the ethical rules for this study. Again, in line 2 is performed the split of header from the data. To the second column of the data frame is given the full name rs2287921.

genome_snp = genome[:,1]
plt.figure()
plt.hist(genome_snp,20)
	Each SNP can have 3 genotypes (AA, AB, BB) and for the rs2287921 we are building the histogram for the allele dosage.
	

Regression *For a random feature f and SNP rs2287921, run a simple linear regression with f as the response variable, and rs2287921as the explanatory variable. Collect association statistics, that is: the effect size ßfs, the standard error Sfs, the pfs-value.*	
# Compute - 1 feature 1 SNP association_statistics_simple=np.zeros([1,3]) *# numpy create an empty array to store 3 values.* snp_values = genome[:,snp_of_interest_index] *# collecting the genomic info for the rs2287921 previously defined*	

```
feature_values = metabolome[:,snp_of_interest_index] # collecting the metabolome info for the rs2287921 previously defined
slope, _, _, p_value, std_err = stats.linregress(snp_values, feature_values)   # calculate a linear regression using scipy.stat
between the last 2 variables
association_statistics_simple=[slope,p_value,std_err] # collect the stats of the linear regression
print (association_statistics_simple) # print them
[-0.004183056475831504, 0.9396402341289458, 0.055225579926 74742]
```

```
# Compute- ALL features 1 SNP

number_features = len(feature_ppms) # calculate the number of features
association_statistics=np.zeros([number_features,3]) # numpy create an empty array to store 3 values for each feature.
snp_values = genome[:,snp_of_interest_index] # collecting the genomic info for the rs2287921 previously defined
for feature_index in range(number_features): # calculate the linear regression and saving the stats in this case using a for loop
    feature_values = metabolome[:,feature_index]
    slope, _, _, p_value, std_err = stats.linregress(snp_values, feature_values)
    association_statistics[feature_index,:]=[slope,p_value,std_err]
```

# Compute - ALL features ALL SNP: same as before but nesting for loops in order to compute for each feature for all the SNPs. Because we have 6 SNP and 3 values for each are calculated (slope ,p_value, std_err), at the end we will have 18 values to save.

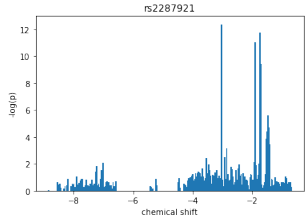
```
number_snps = genome.shape[1]
number_features = len(feature_ppms)
association_statistics__all_snps=np.zeros([number_features, 3*number_snps])

for snp_of_interest_index in range(number_snps):
    snp_values = genome[:,snp_of_interest_index]
    for feature_index in range(number_features):
        feature_values = metabolome[:,feature_index]
        slope, _, _, p_value, std_err = stats.linregress(snp_values, feature_values)
        start = 3 * snp_of_interest_index
        until = start + 3
        association_statistics__all_snps[feature_index,
start:until]=[slope,p_value,std_err]
```

For each feature f and SNP rs2287921, run a simple linear regression with f as the response variable, and rs2287921 as the explanatory variable. For each regression, collect association statistics, that are: the effect size βfs, the standard error Sfs, the pfs-value.

Visualize *pseudo-spectrum of a SNP*

```
x_axis = -np.array(feature_ppms);
plt.figure()
pval = association_statistics__all_snps[:,1]
y_axis = -np.log10(pval)
plt.ylabel('-log(p)'); plt.xlabel('chemical shift')
plt.title(snp_of_interest_name)
plt.bar(x_axis, y_axis, 0.05)
```



For SNP rs2287921, plot the −log10(pvalue) of the SNP's association with each feature. ***This will generate the*** **pseudo-spectrum** *of a SNP, consisting of a set of significance values (−log(P-values)) of its associations with each feature.*

Data Export *Write the associations statics to file, conforming to the format required by metabomatching within phenomenal. The format is described in the documentation, which is available for your perusal here (section 2.1.1)*

The pseudospectrum is loaded in metabomatching as a tab-separated file name   .tsv

The file has 4 columns, the header labels are shift, beta (effect size), se (standard error) and p. The next N rows have the stat data for the association for the N specific metabolome features, indexed by chemical shift.

```
# Export ALL features 1 SNP ##############################################################

pseudospectrum_label = ['shift'] # create the label shift
pseudospectrum_label.append('beta/' + snp_of_interest_name) # create the label beta
pseudospectrum_label.append('p/' + snp_of_interest_name) # create the label p
pseudospectrum_label.append('se/' + snp_of_interest_name) # create the label se
pseudospectrum = np.column_stack((np.array(feature_ppms),association_statistics)) #creation of the dataframe data
pseudospectrum_df = pd.DataFrame(pseudospectrum,columns=pseudospectrum_label) # merging dataframe data and labels

pseudospectrum_df.to_csv('pseudospectrum.tsv',sep='\t',index=False,float_format="%.4g")  # export the file
```

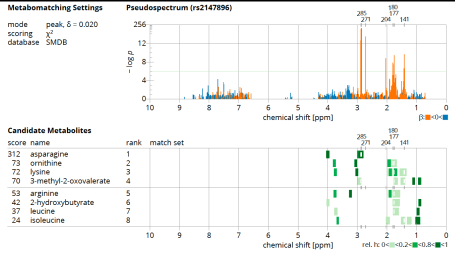
```
# Export ALL features ALL SNPs ###########################################################
# Same as before but for all SNPs using a for loop

snp_names = i2.columns
pseudospectrum_label = ['shift'];
for snp_of_interest_index in range(number_snps):
    snp_of_interest_name = snp_names[snp_of_interest_index]
    pseudospectrum_label.append('beta/rs' + snp_of_interest_name)
    pseudospectrum_label.append('p/rs' + snp_of_interest_name)
    pseudospectrum_label.append('se/rs' + snp_of_interest_name)
pseudospectrum = np.column_stack((np.array(feature_ppms),association_statistics__all_snps))
pseudospectrum_df = pd.DataFrame(pseudospectrum,columns=pseudospectrum_label)
pseudospectrum_df.to_csv('pseudospectrum_all_snps.tsv',sep='\t',index=False,float_format="%.4g")
```

Metabomatching

We will use metabomatching to identify the metabolites underlying significant SNP feature associations. A few ways to do this are described on https://www2.unil.ch/cbg/index.php?title=Metabomatching. The simplest way, and the way on which we'll focus, is to use PhenoMeNal https://public.phenomenal-h2020.eu/

In brief, after loading our data, metabomatching will assigns scores to all metabolites with known NMR spectrum. The scores are computed using the significance values of the features that correspond to peaks in the known spectra. The metabolites are then ranked, based on these scores, to identify the candidate metabolites most likely to underlie the association. We will get also a graphic output, with the summary of score and rank for metabolite.



Metabomatching Settings

mode peak, δ = 0.020
scoring χ^2
database SMDB

Pseudospectrum (rs2147896)

Candidate Metabolites

score	name	rank
312	asparagine	1
73	ornithine	2
72	lysine	3
70	3-methyl-2-oxovalerate	4
53	arginine	5
42	2-hydroxybutyrate	6
37	leucine	7
24	isoleucine	8

Discussion

Here presented the methodology applied for a metabolome- and genome-wide association study on H-NMR urine metabolic profiles. The aim of the code here presented was to generate the association between the SNPs and the metabolome features, to generate the pseudo-spectrum associations and to export them in a valid format in order metabomatching can be applied.

Normally those studies use a target approach: metabolome data are acquired, identified and their concentration quantified and tested for association with SNPs. In this case the metabolome data are binned in metabolome features and tested against genetic associations. The genetically associates features tend to be similar to the NMR peaks for a specific metabolite. In brief Metabomatching is a methodology that use genetic spiking information to identify metabolite candidates, as long as those last ones are present in a spectral database.
The features that presented a genetic association were subjected to both manual (using public databases) and automated metabolite annotation.
The automated annotation in particular was performed using the *metabomatching* approach, using the association *p*-values, effect sizes (*β*), and standard errors (*s*) generated from the simple linear regressions analysis between a SNP and all metabolome features (*pseudospectrum* of the SNP).

$\sum_{f \in F_\delta(m)} \frac{\beta_{rf}^2}{s_{rf}^2},$ The comparison of speudospectra and metabolome reference spectra was performed generating a feature match set $F_\delta(m)$ for every metabolite *m* in the reference database. $F_\delta(m)$ contain all features *f* within a neighborhood of *δ* ppm of any spectrum peak listed in the peak description of *m*. For the pseudospectrum of a given SNP *r* and the spectrum of every metabolite *m*, it was computed the match sum with β_{rf} the effect size and s_{rf} the standard error of the association between SNP *r* and feature *f*. The match sum was considered to be χ^2-distributed with $|F_\delta(m)|$ degrees of freedom to define the score for the tested metabolite as the negative logarithm of the corresponding *p*-value. The high-ranking metabolites are most likely to underlie the SNP-feature associations.

This strategy has multiple advantages: because the metabolite identification is performed after the association study (the normalized H-NMR data themselves were used as phenotypes), this allow to not discard any metabolites from the study; only the metabolites with a proven genetic component are at the end taken in consideration, reducing the burden associated with the metabolic identification. Very importantly the metabomatching method can be applied to any similar study, indeed the GWAs signals can overlap the NMR signal identification itself. The method can then even detect association with unknow metabolites that can be identified afterword as soon as the NMR databases for individual metabolites became more and more accurate. The assignment of a metabolite to all associated features can be difficult (but essential to give a direct mechanistic interpretation) but not necessary if the aim is to find new genetic loci relevant to metabolic variability and eventually relevant as clinical result.
The performance of metabomatching is clearly linked to the strength of genetic spiking and the quality of spectral databases but was robust enough at the end to discover and confirm new SNPs association across 2 different cohorts (6). Metabomatching is therefore likely to become an important tool in metabolome- and genome-wide association studies.

1) **10 Years of GWAS Discovery**: **Biology**, **Function, and Translation**. Visscher PM, et al.; Am J Hum Genet. (2017) 6;101(1):5-22. Review
2) **Population genomics of human gene expression.** Stranger BE et al.; Nat Genet (2007) 39: 1217-1224
3) **A genome-wide perspective of genetic variation in human metabolism.** Illig T. et al.; (2010) Nat Genet 42: 137-141
4) **A genome-wide association study of metabolic traits in human urine.** Suhre K, et al.; (2011) Nat Genet 43: 565–569.
5) **A description of large-scale metabolomics studies: increasing value by combining metabolomics with genome-wide SNP genotyping and transcriptional profiling.** Homuth G, et al. (2012) J Endocrinol 215: 17–28
6) **Genome-wide association study of metabolic traits reveals novel gene-metabolite-disease links.** Rueedi R, et al. PLoS Genet. (2014) 20;10(2)
7) **Metabomatching: Using genetic association to identify metabolites in proton NMR spectroscopy.** Rueedi R, et al. PLoS Comput Biol. (2017) 1;13(12)