DSB205, Winter 2025
Problem Set 4: Graphical models
Due March 14, 2025 at 11:59pm

Submission instructions

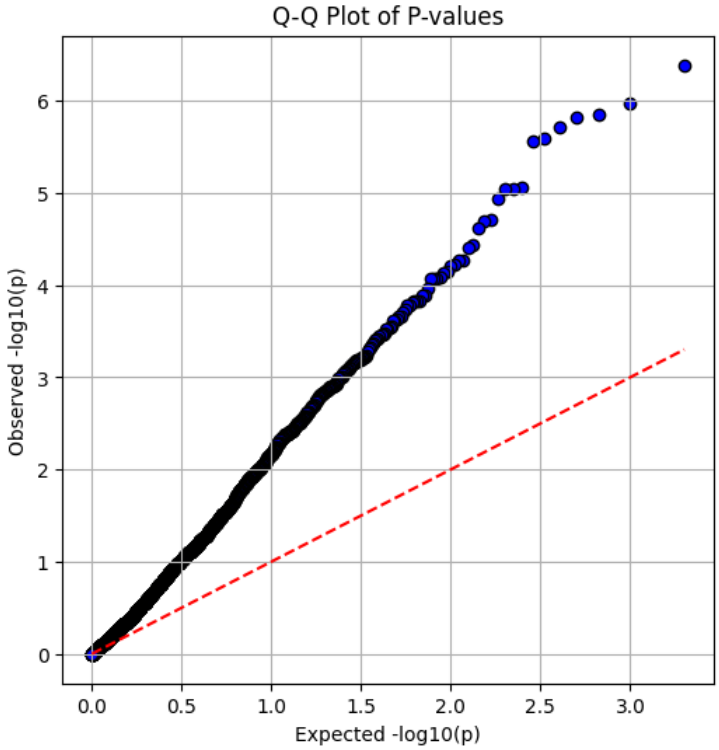
- Submit your solutions electronically on the course Gradescope site as PDF files.
- If you plan to typeset your solutions, please use the LaTeX solution template. If you must submit scanned handwritten solutions, please use a black pen on blank white paper and a high-quality scanner app.

1 Correcting for population stratification in GWAS with PCA [15 pts]

In class, we have talked about how population structure can lead to false discoveries in genome-wide association studies. One approach to correct for population stratification in GWAS is to "explain away" population stratification with principal component analysis.

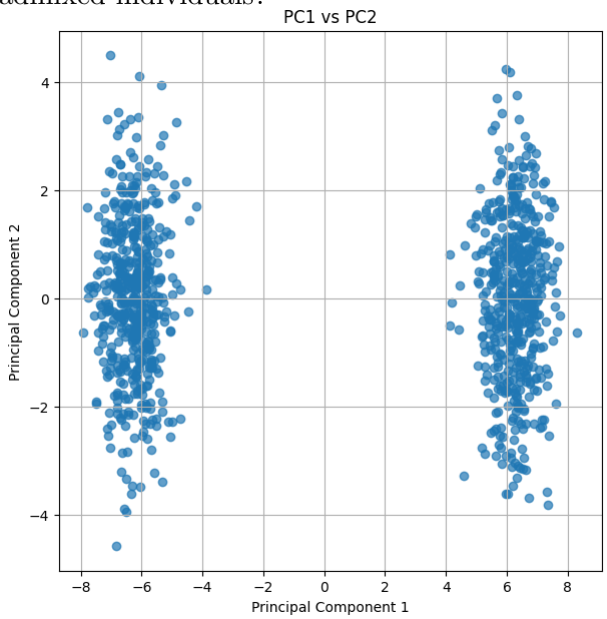
You are given a genotype matrix containing $M = 2000$ SNPs and $N = 1000$ individuals and the binary phenotype associated with each individual.

(a) Run linear regressions on each SNP and the phenotype (You may use the lm function in R for this) and report your p-values in a QQ plot. Do you observe inflation in your p-values? How many SNPs are significantly associated with the phenotype at $\alpha = 0.05$? Control FWER using Bonferroni correction.



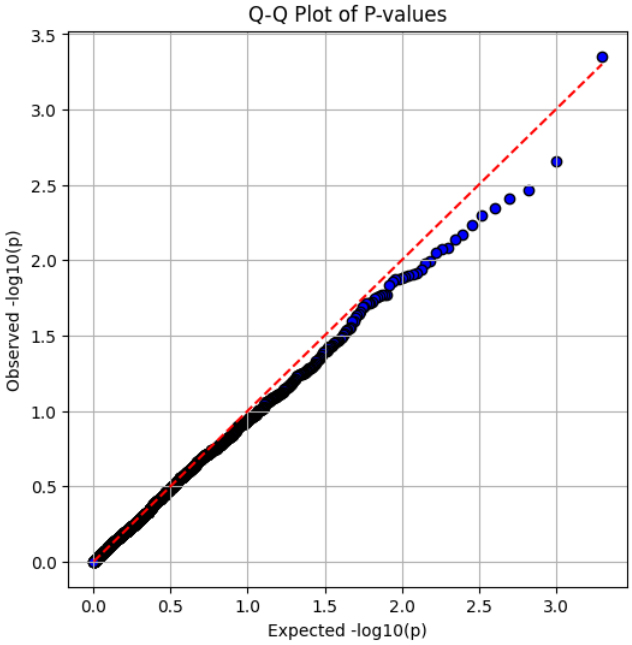
We observe inflation in our p-value distribution. There are 14 significant SNPs controlled with Bonferroni correction.

(b) Use the prcomp function in R to run PCA on the genotype matrix. Plot PC1 against PC2. How many populations do you think are represented in the dataset, given that there are no admixed individuals?



There seems to be 2 populations.

(c) Run linear regressions on each one of your SNPs again, but this time include a term for PC1 in your regression. What does the QQ plot indicate? How many SNPs are significantly associated with the phenotype at $\alpha = 0.05$? Again, control FWER using Bonferroni correction.



QQ Plot seems to show that the data does follow the uniform distribution of p values. 0 significant p values when we acconunt for ancestry in the regression models.

2 Graphical models and phylogenetic trees [25 pts]

In class, we studied HMMs as an example of a graphical model where the graph is structured as a chain. In this problem, we will study graphical models structured as trees.

Recall that we can label the two alleles at a SNP as 0 and 1. We observe the frequency of the 1 allele in a population, *i.e.*, the fraction of individuals in the population that carry the 1 allele at the SNP. We assume that we observe the frequencies of an allele in different populations.

Tree-structured graphical models are a natural model to represent populations relationships. Each node corresponds to a population. The leaves of the tree are the observed values of allele frequencies in present-day populations while the internal nodes are the allele frequencies in an ancestral population.

Let $X_1 \in [0, 1]$ denote the frequency of the 1 allele at a single SNP in population 1. Population 1 splits into 2 populations, 2 and 3. The allele frequency in 2 and 3 are independent given X_1. Population 2 then splits into 4 and 5. We can represent the joint distribution of allele frequencies in the five populations $(X_1, X_2, X_3, X_4, X_5)$ as a graphical model (Figure 1) where each $X_i \in [0, 1]$.

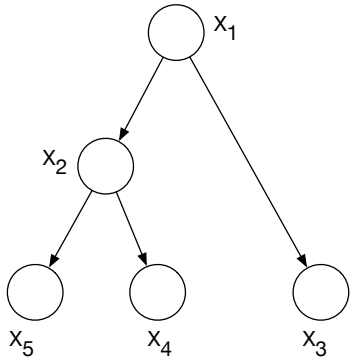


Figure 1:

(a) [4 pts] Write down the joint distribution of $P(x_1, x_2, x_3, x_4, x_5)$ as a product of conditional probabilities.

$$P(x_{1:n}) = \prod_{i=1}^{n} P(x_i | x_{Pa(i)})$$

$$P(x_1, x_2, x_3, x_4, x_5) = P(x_1)P(x_2|x_1)P(x_3|x_1)P(x_4|x_2)P(x_5|x_2)$$

(b) [6 pts] For the graphical model in Figure 1, which of the following conditional independence statements holds ?

i. $X_4 \perp\!\!\!\perp X_5$
False

ii. $X_4 \perp\!\!\!\perp X_5 | X_1$
False

iii. $X_4, X_5 \perp\!\!\!\perp X_3 | X_1, X_2$
True

(c) [7 pts] Generally, we only observe data from the leaves (X_3, X_4 and X_5) which correspond to present-day populations but not from the internal nodes which correspond to ancestral populations. One common problem in evolutionary biology is that we would like to test if the tree specified in Figure 1 could have generated the data (X_3,X_4 and X_5). Show that if the above model generated the observed data, we have: $\mathbb{E}\left[(X_5 - X_4)X_3\right] = 0$.

i. To show this, first consider the distribution over (X_3, X_4, X_5) conditioned on X_1 and X_2. Using the conditional independence propoerties, first show that $\mathbb{E}\left[(X_5 - X_4)X_3|X_1, X_2\right] = 0$. For this result, you will also need to use a property of the conditional distributions $P(x_i|x_{parent(i)})$ common in evolutionary biology, *i.e.*, $\mathbb{E}\left[X_i|X_{parent(i)}\right] = X_{parent(i)}$ (this assumption captures the idea that the average allele frequencies do not change over time).

Using independence rule of expected value, E[XY] = E[X]E[Y]. Given the parents, X_4 and X_5 are independent to X_3, and we can use that rule.

$$\mathbb{E}\left[(X_5 - X_4)X_3|X_1, X_2\right] = \mathbb{E}\left[(X_5 - X_4)|X_1, X_2\right]\mathbb{E}\left[X_3|X_1, X_2\right]$$

Focus on first term. Also, note that conditioning on additional parents doesn't increase information, so we can ignore them.

$$\mathbb{E}\left[(X_5 - X_4)|X_1, X_2\right] = \mathbb{E}\left[X_5|X_1, X_2\right] - \mathbb{E}\left[X_4|X_1, X_2\right]$$
$$\implies \mathbb{E}\left[X_5|X_2\right] - \mathbb{E}\left[X_4|X_2\right] = X_2 - X_2 = 0$$

Plugging it back in:

$$\mathbb{E}\left[(X_5 - X_4)|X_1, X_2\right]\mathbb{E}\left[X_3|X_1, X_2\right] = 0 \times \mathbb{E}\left[X_3|X_1, X_2\right] = 0$$

ii. Having shown the above result, $\mathbb{E}\left[(X_5 - X_4)X_3|X_1, X_2\right] = 0$, use the tower property of expectation to obtain the final result.

$$\mathbb{E}\left[(X_5 - X_4)|X_3\right] = \mathbb{E}\left[\mathbb{E}\left[(X_5 - X_4)X_3|X_1, X_2\right]\right] = \mathbb{E}\left[0\right] = 0$$

(d) [5 pts] We will apply this theory to real data from a European, an African and an Asian population. In this setting, X_5, X_4 and X_3 represent European, African and Asian population. hw.4-1.txt consists of 1614 SNPs (one per line) from the three populations (one per column) where for each SNP $j, j \in \{1 : 1614\}$, we observe an independent sample from the joint distribution (X_3^j, X_4^j, X_5^j). We can estimate $\mathbb{E}\left[(X_5 - X_4)X_3\right]$ by the statistic: $\hat{d} = \frac{\sum_{j=1}^{m}(x_5^j - x_4^j)x_3^j}{m}$. What is $\hat{d}$ from this data ?

From this data, $\hat{d}$ is 0.01075.

(e) [2 pts] We would like to test the null hypothesis $H_0 : \mathbb{E}[(X_5 - X_4)X_3] = 0$. Under H_0 : $\hat{d} \sim \mathcal{N}(0, \sigma^2)$ where $\sigma = 4 \times 10^{-3}$ (we won't get into the details of how we compute σ here). Compute the p-value for H_0.

P-value is 0.0072.

(f) [1 pts] Based on the p-value, can you accept the tree that relates Europeans, Africans and Asians ?

Since p-value is below the 0.05 significance level, we reject the null. We reject the tree that relates the different populations.

3 Hidden Markov Models [15 pts]

Consider a HMM with hidden variables $Z_{1:m}$ and observed variables $X_{1:m}$. Assume that each $Z_t \in \{1 : K\}$.

(a) [5 pts] In class, we discussed the forward algorithm. Now fill in the recursion for the backward algorithm. Show how we can write $\beta_{t-1}(j) = \sum_{i=1}^{K} \beta_t(i) P(Z_t = i|Z_{t-1} = j) P(X_t|Z_t = i)$.

First, let's look at $\beta_t(j)$:

$$\beta_t(j) = p(X_{(t+1):m}|Z_t = j)$$

Similarly, to get to time step t-1, we get:

$$\beta_{t-1}(j) = p(X_t, X_{(t+1):m}|Z_{t-1} = j)$$

Now, marginalizing over all possible states of Z_t, and then using chain rule:

$$\beta_{t-1}(j) = \sum_{i}^{K} p(X_t, X_{(t+1):m}, Z_t = i|Z_{t-1} = j)$$

$$\beta_{t-1}(j) = \sum_{i}^{K} p(X_t, X_{(t+1):m}|Z_t = i, Z_{t-1} = j) p(Z_t = i|Z_{t-1} = j)$$

We know that once we know Z_t, any prior info doesn't matter:

$$p(X_t, X_{(t+1):m}|Z_t = i, Z_{t-1} = j) \implies p(X_t, X_{(t+1):m}|Z_t = i)$$

Let's expand that term using conditional independence and substitute in $\beta_t(j)$

$$p(X_t, X_{(t+1):m}|Z_t = i) \implies p(X_t|Z_t = i) p(X_{(t+1):m}|Z_t = i)$$

$$\implies p(X_t|Z_t = i) \beta_t(i)$$

$$\beta_{t-1}(j) = \sum_{i}^{K} \beta_t(i) p(Z_t = i|Z_{t-1} = j) p(X_t|Z_t = i)$$

(b) [5 pts] The forward-backward algorithms allow us to efficiently compute the probabilities $\gamma_t(j) = P(Z_t = j|X_{1:m})$ for all $t \in \{1 : m\}$. It is also useful to be able to compute the probabilities $\xi_t(i, j) = P(Z_t = i, Z_{t+1} = j|X_{1:m})$. Show how ξ_t can be computed assuming we are given all the α_t and β_t, $t \in \{1 : m\}$.

$$\xi(i, j) = P(Z_t = i, Z_{t+1} = j|X_{1:m}) = \frac{P(Z_t = i, Z_{t+1} = j, X_{1:m})}{P(X_{1:m})}$$

Let's split up the X to make the computation easier, and then apply the chain rule:

$$P(Z_t = i, Z_{t+1} = j, X_{1:m}) \implies P(Z_t = i, Z_{t+1} = j, X_{1:t}, X_{(t+1):m})$$

$$P(X_{1:t}, Z_t = i, X_{(t+1):m}, Z_{t+1} = j) = P(X_{1:t}, Z_t)P(X_{(t+1):m}, Z_{t+1} = j|X_{1:t}, Z_t)$$
$$= P(X_{1:t}, Z_t)P(Z_{t+1} = j|X_{1:t}, Z_t)P(X_{(t+1):m}|Z_{t+1} = j, X_{1:t}, Z_t)$$

Now, using Markov properties of excess information, we can simplify numerator probability:

$$= P(X_{1:t}, Z_t)P(Z_{t+1} = j|Z_t)P(X_{(t+1):m}|Z_{t+1} = j)$$

Substituting in α and transition probability ρ:

$$\alpha_t(i)\rho(i, j)P(X_{(t+1):m}|Z_{t+1} = j)$$

Let's expand the final term's X, and then sub in emission probability ψ and β:

$$= \alpha_t(i)\rho(i, j)P(X_{(t+1)}|Z_{t+1} = j)P(X_{(t+2):m}|Z_{t+1} = j)$$
$$= \alpha_t(i)\ \rho(i, j)\ \psi_{t+1}(j)\ \beta_{t+1}(j)$$

Plugging back numerator to original expression:

$$\xi(i, j) = \frac{\alpha_t(i)\ \rho(i, j)\ \psi_{t+1}(j)\ \beta_{t+1}(j)}{P(X_{1:m})}$$

Denominator was derived in lecture:

$$\xi(i, j) = \frac{\alpha_t(i)\ \rho(i, j)\ \psi_{t+1}(j)\ \beta_{t+1}(j)}{\sum_j^K \beta_t(j)\alpha_t(j)}$$

(c) [5 pts] The forward-backward algorithm has computational complexity of $\mathcal{O}(K^2)$. This does not use any further assumptions about the joint probability except the conditional independence assumptions associated with the HMM. We can reduce the computational cost if we are given additional information about the probability distribution.

Consider a HMM with the following transition probability between time $t-1$ and t. We first draw R_t – a binary random variable. If $R_t = 0$, then $Z_t = Z_{t-1}$. If $R_t = 1$, then Z_t is chosen independently of Z_{t-1}. Formally:

$$\begin{aligned} R_t &\sim \text{Ber}(p) \\ Z_t|Z_{t-1}, R_t = 1 &\sim Unif(\{1, \ldots, K\}) \\ Z_t|Z_{t-1}, R_t = 0 &= Z_{t-1} \end{aligned}$$

Write the computationally efficient recursion for the forward computation. What is the new computational complexity ?

(This HMM is used often in genetics in problems such as genotype imputation or phasing. Consider the problem of genotype imputation. We have a reference panel of K individuals and a test individual that we would like to impute ("fill in missing SNPS"). We will use the idea that the genome of the test individual can be obtained by copying segments of the genomes from the reference panel. However, different positions along the genome might copy from different reference individuals because of recombination.)

From lecture, we derived the forward algorithm as:

$$\alpha_t(j) = P(x_{1:t}, z_t = j) \implies \sum_i^K \alpha_{t-1}(i) P(x_t|z_t = j) P(z_t = j|z_{t-1} = i)$$

From the given conditions, we get:

$$\begin{aligned} P(z_t = j|z_{t-1} = i) &= p \cdot P(Z_t|Z_{t-1}, R_t = 1) \cdot (1-p) P(Z_t|Z_{t-1}, R_t = 0) \\ &= p \cdot \frac{1}{K} \cdot (1-p) \mathbf{1}_{\{j=i\}} \end{aligned}$$

Plugging it back in:

$$\alpha_t(j) = \sum_i^K \alpha_{t-1}(i) \psi(j) [\, p \cdot \frac{1}{K} \cdot (1-p) \mathbf{1}_{\{j=i\}} \,]$$

$$\alpha_t(j) = \psi(j) \sum_i^K \alpha_{t-1}(i) [(1-p) \mathbf{1}_{\{j=i\}} \cdot \frac{p}{K}]$$

$$\alpha_t(j) = \psi(j)[\sum_{i}^{K} \alpha_{t-1}(i)(1-p)\mathbf{1}_{\{j=i\}} + \sum_{i}^{K} \frac{p}{K}\alpha_{t-1}(i)]$$

Notice that the term $\alpha_{t-1}(i)(1-p)\mathbf{1}_{j=i}$ will only be present when j=i. Therefore, the summation to K can be simplified as a constant.

$$\alpha_t(j) = \psi(j)[\alpha_{t-1}(j)(1-p) + \frac{p}{K}\sum_{i}^{K} \alpha_{t-1}(i)]$$

Even though there is a summation term, this summation term only needs to be performed once during the forward pass because it doesn't depend on j, so it's a constant. Thus, the time complexity comes down to O(K).