1 Statistical Genetics

1. Overview of latent variable approach to genetic mapping

   Review of existing methods of genetic mapping with sequencing data:

   - Generic model: given p SNPs in a gene, $X_1, \cdots, X_p$ and response variable Y (binary or continuous). The simple regression model:

     $$g(\mu) = X\beta \tag{1}$$

     where g is the link function and $\mu = \mathrm{E}(Y|X)$. What is interesting, however, is the gene-level effect (some kind of aggregate over β_j's). To see this, we could assume a model:

     $$\{X_1, \cdots, X_p\} \to Z \to Y \tag{2}$$

     where Z is a latent variable representing the state of gene (e.g. whether gene function is disrupted by X_j's). Assume the effect of X_j on Z is ξ_j, and the effect of Z on Y is τ, and assume linear model, then we have the linear model in terms of X_j's and Y:

     $$\mathrm{E}(Y|X_1, \cdots, X_p) = \sum_j \tau\xi_j X_j \tag{3}$$

     This is a linear model where we assume $\beta_j = \tau\xi_j$. Different existing methods correspond to different ways of parametrize β_j or ξ_j.
   - Multiple-Marker Test: testing the hypothesis: $H_0 : \beta_1 = \cdots = \beta_p = 0$, e.g. via Hotelling's T^2 test. No assumption of β_j is made, thus we are comparing a simple H_0 vs. a complex alternative model: loss of power.
   - Collapsing method: typically multiple rare variants (RVs) are grouped (e.g. those with the same MAF), and a dummy variable is used. This is equivallent to assuming that $\xi_j = 1$ if the frequency is within a threshold, and 0 otherwise. The drawbacks are: (1) ξ_j's are fixed (actually unknown), sensitive to misclassification of variants. (2) Also suffers if variants with different signs (some beneficial, some harmful) are included in the same group.
   - Lin-Tang method (EREC) and VAAST:
     - EREC method: fix ξ_j (called the weight of SNP), and perform regression with $\beta_j = \xi_j\tau$, and make inference on τ. In the EREC method, $\xi_j = \hat{\beta}_j$ (plus some small constant).
     - VASST method: similarly, fix ξ_j at MLE and infer τ, but done with a generative model. β_j is related to the variant frequencies in cases and in controls, and MLE of these frequencies are used in the paper.
     - Assessment: the method is similar to the no-pooling option in a hierarchical normal model: suppose θ_j is the mean of the j-th group, estimate θ_j from the observations of the j-th group, and estimate the population level parameter, μ, by assuming:

       $$\theta_j \sim N(\mu, \tau^2) \tag{4}$$

       The drawbacks are: (1) high variance of $\hat{\beta}_j$; (2) prior information of variants are not included.

- Penalized regression/LASSO: penalty $\|\beta\|_1$ or $\|w\beta\|_1$, where w further weights the SNPs (e.g. favoring SNPs with low MAF). Then a gene is selected if any of $\beta_j \neq 0$ in LASSO. The problems:
  - No pooling: when LASSO is deciding between $\beta_j = 0, \forall j$ vs. some $\beta_j \neq 0$, it is performing a task similar to the multiple-marker test, and this suffers from model complexity. Instead, in a collapsing/hierarchical model, SNPs are assumed to share β_j's.
  - Penalty: the original LASSO penalizes the coefficients β_j, this may unfaily punish the rare variants with large effects but low MAF. Weighting only partially overcome this problem.

Overview of latent variable model:

- Motivation: gene function/activities are natural variables that determine the response variable (phenotype). Furthermore, a large body of knowledge is expressed in terms of genes. By working at the level of genes (or other molecular traits), the dimensionality of the problem is significantly reduced. The difficulty is that gene function/activities are not observed, thus a model involving these latent variables may not be identifiable.
- Information of SNPs: the effect of a SNP on a gene can be estimated by using several lines of data - evolutionary conservation, the amino acid change, the allele frequency (in general, very low frequency means the SNP is probably under some constraint, thus probably affecting the gene), etc. Using these information would allow one to have a reasonable guess of whether a SNP influences a (latent) gene variable.
- External data: e.g. if genotype-transcriptome data is available, the effects of SNPs on gene expression level can be directly estimated from data. Thus in a different dataset of genotype-phenotype, the latent variable of gene level can be estimated using the SNP-to-gene model learned before.
- Learning gene-level effect from latent variables: learning accurately the effect of individual SNPs on phenotype is difficult because of the small sample size per SNP. However, we could use all SNPs of the gene to learn the effect at the gene-level.
- Learning gene-gene interactions from latent variables: SNP-SNP interactions are generally hard to find in practice because of the many large number of pairs to test. At the gene-level, the number of pairs is much smaller; and we could do similar information aggregration over all SNPs of each gene to estimate the extent of gene-gene interaction.
- Using prior knowledge of genes: e.g. whether a gene is implicated to a disease before, or whether two genes belong to the same pathway. These knowledge can be encoded in a latent model as prior of gene effects, or gene-gene interactions.
- Molecular diagnosis in personalized medicine: given a patient, the problem of personalized medicine is to determine the cause of the disease, and in our framework, the problem is to infer the function/activity of genes that are known to be related to disease. Furthermore, this can incorporate more information, e.g. age-of-onset, family history, other diagostic measures, etc.
- Summary: a unified framework to integrate various sources of data: SNP information (conservation, biochemical properties of amino acids, etc.), external genetic data (e.g.

expression QTL), and prior knowledge of genes and gene networks, for better estimation of gene effects and gene-gene interactions.

Simple latent variable model:

- Model : suppose we have the model for one gene:

$$\{X_1, \cdots, X_p\} \to Z \to Y \tag{5}$$

where X_j are SNPs, Y is the response variable (case/control state), and Z is the gene state variable (latent), representing the extent the gene function/activity is perturbed by the SNPs. Z is related to X_j's by logistic or linear regression (see below). The coefficients of this regression, γ_j, measures the effect of the j-th SNP on the gene, and is parameterized by w_j, known constants (see below). Y is related to Z through:

$$g(\mathrm{E}\,(Y|Z)) = \beta_1 Z + \beta_0 \tag{6}$$

where $g(\cdot)$ is the link function. For linear regression, there is an additional parameter σ^2. The goal is to estimate $\beta = (\beta_1, \beta_0)$ given the data or infer the posterior distribution $P(\beta|X, Y, w)$, with γ be additional parameters and Z be missing data.

- Fixed vs. random effect: we have an option of having γ as functions (deterministic) of w, or γ follows prior distributions parameterized by w. The fixed effect model, however, is probably too constrained: the model would have only 2 dof. (β_1 and β_0) plus a few variance parameters. So it is important to have a model where γ_j's are random.
- Assessing damaging mutations: programs such as PolyPhen report the probability that a mutation disrupts the gene function. w_j's can be set as these probabilities. Alternatively, the population frequencies of SNPs can be used: e.g. we could have $w_j = 1$ if the frequency $p_j < \phi_0$ where ϕ_0 is some threshold (e.g. 0.01), and $w_j = 0$ otherwise. In all cases, we choose w_j to be a number in $[0, 1]$.
- Prior distribution of γ: we assume that γ follows normal distribution with variance σ_γ^2 (to be estimated). Csonsider two cases:
  - Binary latent variable: suppose Z is binary (0 if normal, 1 if perturbed), then Z is related to X through:

$$P(Z = 1|X, \gamma) = \mathrm{logit}^{-1}\left(\sum_j \gamma_j X_j\right) \tag{7}$$

Assume w_j is interpreted as the probability that the risk allele of X_j affects the function of the gene. Since γ_j is the log odds ratio, we have

$$\mathrm{E}\,(\gamma_j) = \log\frac{w_j}{1 - w_j} + \delta \tag{8}$$

where δ is a constant (the base-line log-odds of the reference allele).

  - Continuous latent variable: if Z is continuous, Z is related to X through a linear model:

$$E(Z|X, \gamma) = \sum_j \gamma_j X_j \tag{9}$$

We need to fix the scale of γ, otherwise the model is not identifiable (we could scale Z and γ simultaneously). Also note that there is no intercept term γ_0 again for the identifiability reason. Given w_j, we assume that $\gamma_j \sim N(w_j, \sigma_\gamma^2)$.

- Continuous vs. binary latent variables: we prefer the continuous latent variables for several reasons:
  - Sign of effect: while in most cases, a mutation damages (decreases) the function of a gene, in rare cases, a variant may increase the gene activity. This is especially true for SNPs with regulatory effects (e.g. making transcription faster, or mRNA more stable). The sign is not easily captured with binary latent variables.
  - Ease of inference: with linear regression of Z on γ, it is easy to show that the predictive distribution of Z is also normal. This make inference much easier. In contrast, under the binary latent variable model: integration of γ is not analytically solvable.
  - Generality: to extend the model to the case where Z is a molecular trait, e.g. level of gene expression or metabolite, the continous model is easier.

Evaluation of the simple latent variable model:

- Analysis of model identifiablility: because of latent variables, identifiability is a major issue, i.e. very different parameter settings may lead to equally good fit of data. To see this, we consider several naive way of assigning latent variables:
  - We could choose z_i to be close to y_i (s.t. τ is very large): but the fitting of X to Z may be poor then.
  - Alternatively, we could choose Z to fit X_j's (effectively some kind of average of X_j's), but then Z may be poorly correlated with Y.

  The optimal way of setting latent variables should thus balance the fitting of both X to Z and Z to Y.
- Comparison with collapsing methods:
  - The collapsing method: we use the SNP information to fix $Z = \sum_j x_j w_j$, where w_j is the damaging potential of the j-th SNP. And then it estimates β using the regression of Y on Z. Collapsing method is thus a special case of our model, with $\sigma_\gamma^2 = 0$ and $\sigma_z^2 = 0$ (no uncertainty of γ and Z).
  - Benefits of latent variable model: effectively, given a random value of β, the model infers best γ (linear regression of Y on X_j); then given γ, one obtains a better estimate of Z, which is then be used to have a better estimate of β, etc. In the end, the estimate of γ_j is better than w_j (prior mean), and the estimate of Z is better than Xw (the naive collapsed variable), and as a result, a better $\hat{\beta}$.
  - Remark: as with collapsing method, the latent variable method enjoys the benefit of pooling: β is learned from all samples where $Z = 1$ ($Z = 0$ is not informative, i.e. no genetic perturbation), and the information of SNPs are integrated into Z. Meanwhile, it further improves the naive collapsing method, by allowing it to update the naive collapsed variable.
- Comparison with the EREC method by Lin-Tang: assume γ is random, but obtain crude estimate of γ by performing regression of Y on X directly, and then fix γ and estimate

β. This is an approximation of our latent variable model, where inference is not done iteratively (γ is never updated once estimated in a crude way).

Latent variable model with interactions:

- Motivation: the main challenge is that there are many possible interactions between genetic mutations. This may include:
  - Gene-level interaction: SNPs interaction within genes.
  - Gene-gene interaction: SNPs of one gene interact with SNPs of another gene.
  - Pathway-level interaction: gene interaction within pathways.
  - Between-pathway interaction: genes of one pathway interact with genes of another pathway.
- Simple method for pathway-level test: we consider a case where we want to associate a pathway with a phenotype, allowing interactions. Our model would be:

$$Y = \sum_j \beta_j X_j + \sum_{i,j} \beta_{ij} X_i X_j + \epsilon \tag{10}$$

  with the null hypothesis to be tested: $H_0 : \beta_{ij} = 0$ for any i, j. While it is possible to do the test, it suffers from a very high level of dof. $p + p^2$.
- Benefit of latent variable model: to associate a pathway with a phenotype, or in general p features with the response, we would group the features into K groups, with each group of features potentially affecting one latent variable, and the respond depends on the K latent variables, with possible interactions. The number of parameter is thus:

$$\text{dof} = p + K + K^2 \tag{11}$$

  In general, this is much less than $p + p^2$ if $K << p$.
- Alternative strategy: hierarchical model. In Equation 10, we could introduce priors on β_{ij} to reduce dof. Example: $\beta_{ij} = 0$ if i, j in the same group, and $\beta_{ij} \sim N(\mu_{kl}, \sigma^2_\beta)$ if i in the group k and j in the group l.
- Alternative strategy: feature expansion (collapsing at group level). Assume the p features can be partitioned into K groups, introduce additional variables that represent each of the K group. Example, for any potential interacting pair of variables (e.g. pair of PPI genes), X_j and X_k, define a new variable $Z = X_j \wedge X_k$.
  - The model is flexible, allowing many different features to be added, and it allows X_j individually affecting Y (without going through the intermediate latent variable).
  - The limitation: new variables are pre-defined, unlike the latent variable model, where there could be additional (unknown) parameters associating the observed and unobserved variables.

Using latent variables in integrating multiple datasets:

- Motivation: in a genetic dataset, there are potentially many more latent variables: gene expression level, metabolite level, physiological traits (blood pressure, glucose level, etc). Introducing these variables (including their interactions with genetic changes) may allow one to better explain phenotypes.

- Using GWAS data to improve genetic analysis: GWAS of many physiological and molecular traits have been performed, including eQTL and metabolite-QTL. To use these datasets for a NGS dataset, we could extract all loci from GWAS for a trait, and map these loci to genes. Then from sequencing data, predict the functional disruption of these genes; or alternatively, simply treat the trait-associated genes as a GWAS-defined gene group, and apply the group-level analysis.

2. Simple latent variable model with quantitative phenotypes

   Model of quantitative phenotypes:

   - Model: given the variants, $X_1, \cdots, X_p$, the gene activity (latent variable) follows:

   $$Z = \sum_j X_j \gamma_j + \delta \tag{12}$$

   where $\delta \sim N(0, \sigma_z^2)$. The prior of the coefficients follow normal distribution:

   $$\gamma_j \sim N(w_j, \sigma_\gamma^2) \tag{13}$$

   The phenotype is a linear model of the gene activity:

   $$Y = \beta_1 Z + \beta_0 + \epsilon \tag{14}$$

   where $\epsilon \sim N(0, \sigma_y^2)$ is the error term. Given the data X, Y, the prior knowledge w, our goal is to infer $\beta, \sigma_\gamma^2, \sigma_z^2, \sigma^2 | X, Y, w$. We ignore the constants w in the notations. In practice, we may also want to fix σ_γ (the accuracy of our prior knowledge, may not be identifiable).

   - Model simplification attempt: one is tempted to simplify the model in this way: we first marginalize γ in $Z|X, \phi$, and the result is normal distribution, with mean:

   $$\mathrm{E}\,(Z|X,\phi) = \mathrm{E}_\gamma(Z|X,\phi,\gamma) = \mathrm{E}_\gamma(X\gamma) = X\mathrm{E}_\gamma(\gamma) = Xw \tag{15}$$

   and variance:

   $$\begin{aligned} \mathrm{Var}\,(Z|X,\phi) &= \mathrm{Var}_\gamma[\mathrm{E}\,(Z|\gamma)] + \mathrm{E}_\gamma[\mathrm{Var}\,(Z|\gamma)] \\ &= \mathrm{Var}_\gamma[X\gamma] + \mathrm{E}_\gamma[\sigma_z^2] = \sigma_\gamma^2 \textstyle\sum_j x_j^2 + \sigma_z^2 \end{aligned} \tag{16}$$

   Therefore, the conditional distribution of Z given X:

   $$Z|X,\phi \sim N(Xw, \sigma_\gamma^2 \|X\|_2^2 + \sigma_z^2) \tag{17}$$

   We have thus a regression model of Y on Z, where the explanatory variables are not observed, but follow normal distribution. Note that when X has more SNP, Z will have large variance: this is understandable because we will introduce more uncertainty with the effect of each SNP. To summarize, our model is (for the i-th observation):

   $$z_i \sim N(\mu_i, \sigma_i^2(\phi)) \tag{18}$$

   where

   $$\mu_i = x_i w \qquad \sigma_i^2(\phi) = \|x_i\|_2^2 \sigma_\gamma^2 + \sigma_z^2 \tag{19}$$

   And the response variable:

   $$y_i = \beta_1 z_i + \beta_0 + \epsilon_i \qquad \epsilon_i \sim N(0, \sigma_y^2) \tag{20}$$

- Errors of model simplification: we note that we have n iid. samples, and the error in the above analysis is that for different sample, different values of γ may be used, in other words:

$$\int p(z_1, \cdots, z_n|\gamma)d\gamma = \int \prod_i p(z_i|\gamma)d\gamma \neq \prod_i \int p(z_i|\gamma)d\gamma \tag{21}$$

- Remark: in Bayesian inference, be careful to integrate out the parameters (or variables) in the middle of the model. In particular, be careful when integrating out parameters/variables for indivudal data points (because they may be shared by multiple data points). Better to do this at the level of the final posterior distribution or morginal likelihood (see below).

Likelihood and posterior:

- Likelihood: we first decompose the likelihood function of the parameters $(\beta, \gamma, \sigma_y^2, \sigma_z^2)$. We could do this because z_i is not shared by other data points.

$$p(y|x, \beta, \gamma) = \int \prod_i p(y_i|z_i, \beta)p(z_i|x_i, \gamma)dz_1 \cdots dz_n = \prod_i \int p(y_i|z_i, \beta)p(z_i|x_i, \gamma)dz_i \tag{22}$$

  Note that for the notational simplicity, we ignore σ_z^2 and σ_y^2 in the function. The likelihood of the i-th observation is (the i-th term in the product) is a mixture of normal distribution, and thus $y_i|x_i$ also follows normal distribution with mean:

$$\mathrm{E}\,(y_i) = \mathrm{E}_{\,z_i}(y_i|z_i) = \mathrm{E}_{\,z_i}(\beta_1 z_i + \beta_0) = \beta_1 x_i \gamma + \beta_0 \tag{23}$$

  and variance:

$$\mathrm{Var}\,(y_i) = \mathrm{E}_{\,z_i}[\mathrm{Var}\,(y_i|z_i)] + \mathrm{Var}_{\,z_i}[\mathrm{E}\,(y_i|z_i)] = \sigma_y^2 + \mathrm{Var}_{\,z_i}(\beta_1 z_i + \beta_0) = \beta_1^2\sigma_z^2 + \sigma_y^2 \tag{24}$$

  The likelihood is thus:

$$p(y|x, \beta, \gamma) = \prod_{i=1}^{n} N(y_i|\beta_1 x_i \gamma + \beta_0, \beta_1^2\sigma_z^2 + \sigma_y^2) \tag{25}$$

- Model identification: note that:

$$\beta_1 x_i \gamma = \sum_j (\beta_1 \gamma_j) x_{ij} \tag{26}$$

  The parameters β_1 and γ_j are coupled, so without prior, the model is not identified. By defining a prior distribution of γ (assuming the hyperparameters are known), the posterior distribution is identifiable. Second, we note that we have two free parameters, σ_z and σ_y, for the variance, thus not identifiable. So we define:

$$\sigma^2 = \beta_1^2\sigma_z^2 + \sigma_y^2 \tag{27}$$

  as the free parameter.

- Posterior distribution: using the new notation, our likelihood function is:

$$y|\beta,\gamma,\sigma^2 \sim N(\beta_1 x\gamma + \beta_0, \sigma^2 I) \tag{28}$$

We use normal prior for γ and the noninformative prior for other parameters:

$$p(\beta_1,\beta_0) \propto 1 \qquad p(\gamma_j) \sim N(w_j,\sigma_\gamma^2) \qquad p(\sigma^2) \propto 1/\sigma^2 \tag{29}$$

Or we can write the prior of γ in vector form: $\gamma \sim N(w,\sigma_\gamma^2 I)$. The posterior distribution:

$$p(\beta,\gamma,\sigma^2|y) \propto \frac{1}{\sigma^2} N(\gamma|w,\sigma_\gamma^2 I) N(y|\beta_1 x\gamma + \beta_0, \sigma^2 I) \tag{30}$$

- Remark: the model has some similarity, but different from error-in-variable model. In our model, the latent variable can be integrated out, and we are interested in the conditional distribution $p(y|x)$; while in the EIV model, needs to consider the joint distribution $p(x,y)$. EIV analysis is thus based on e.g. covariance between the variables.

Approximate inference:

- Inference strategy: we are interested in $p(\beta|y)$, and so we need to integrate out other parameters $\gamma,\sigma^2,\sigma_z^2$, in the posterior distribution. Even if $p(\beta|y)$ has no simple form, we could sample from the distribution easily if for any β, we can compute its value easily.
- Integrating out γ: similar to computing model evidence in Bayesian linear regression.
- Crude estimation of parameters: we could obtain a crude estimate of z_i as its mean $x_i w$, then do regression of y_i on $x_i w$ to obtain β and σ^2. To obtain a crude value of ϕ, we could do 1D maximization of the posterior distribution.
- Posterior mode and normal approximation: could use Conditional Maximization (CM) algorithm. First, we could define:

$$\phi = \beta_1^2 \sigma_z^2 + \sigma^2 \tag{31}$$

Since σ^2 is a free parameter, we could do maximization wrt. parameters, β,γ,σ_z^2 and ϕ, then solve σ^2 from the estimated values of all the parameters. If the variance parameters (ϕ and σ_z^2) are known, we need to minimize:

$$\min_{\beta,\gamma} \frac{1}{2\sigma_\gamma^2}\sum_j (\gamma_j - w_j)^2 + \frac{1}{2\phi}\sum_{i=1}^n \left(y_i - \beta_1 \sum_j \gamma_j x_{ij} - \beta_0\right)^2 \tag{32}$$

When γ is given, this is an ordinary least square problem. When β is given, this is similarly a Bayesian linear regression with informative prior and γ can be solved easily as well.

- EM algorithm for posterior mode: let $\theta = (\beta,\phi,\sigma_z^2)$ be the parameters. At the E-step, we compute the expected log-likelihood averaging over z_i:

$$Q(\theta|\theta^{(t)}) = \sum_i \mathrm{E}_{z_i|\theta^{(t)}}[\log p(y_i,z_i|\theta)] = \sum_i \mathrm{E}_{z_i|\theta^{(t)}}\left[\log p(z_i|\phi) + \log p(y_i|z_i,\beta,\sigma^2)\right] \tag{33}$$

Let $b_i = \|x_i\|_2^2 \sigma_\gamma^2$. The two terms are given by:

$$\log p(z_i|\phi) = -\frac{1}{2}\log(b_i+\phi) - \frac{1}{2}\frac{(z_i-\mu_i)^2}{b_i+\phi} \tag{34}$$

$$\log p(y_i|z_i,\beta,\sigma^2) = -\log\sigma - \frac{1}{2}\frac{(y_i-\beta_1 z_i-\beta_0)^2}{\sigma^2} \tag{35}$$

Given $\theta^{(t)}$, z_i follows normal distribution, and this allows to take expectation of the two terms above (TO DO). In the M-step, we could then solve β and σ analytically, but ϕ has to be solved numerically because of the $(b_i+\phi)$ term.

- Enumeration on grids: the model has 4 parameters, thus we could (1) first define a rough range of values for all parameters and divide the intervals into grids; (2) enumerate the grids: compute the (unnormalized) posterior density at each grid.

Gibbs sampling:

- $\beta|\gamma,\sigma^2,y$: when γ is given, let $\tilde{x} = x\gamma$, then we have an univariate regression of y on x':

$$y \sim N(\beta_1\tilde{x}+\beta_0,\sigma^2) \tag{36}$$

The prior is noninformative: $p(\beta_1,\beta_0) \propto 1$. Let $\hat{\beta}_{LS}$ be the least square estimator of β (2-dim. vector), and V_{LS} be the covariance matrix of the estimators under ordinary regression, then the posterior follows from standard Bayesian regression with noninformative prior:

$$\beta|\sigma^2,y \sim N(\hat{\beta}_{LS},\sigma^2\cdot V_{LS}/\text{MSE}) \tag{37}$$

- $\gamma|\beta,\sigma^2,y$: when β is given, we define $\tilde{y} = y-\beta_0$, and $\tilde{x} = \beta_1 x$, then we have the regression:

$$\tilde{y} \sim N(\tilde{x}\gamma,\sigma^2) \tag{38}$$

The prior distribution $\gamma \sim N(w,\sigma_\gamma^2 I)$. Applying the standard results of Bayesian regression with semi-conjugate prior:

$$\gamma|\beta,\sigma^2,y \sim N(\hat{\gamma}_n,V_n) \tag{39}$$

where

$$\hat{\gamma}_n = V_n\left(\frac{1}{\sigma_\gamma^2}w + \frac{1}{\sigma^2}\tilde{X}^T\tilde{y}\right) \tag{40}$$

$$V_n = \left(\frac{1}{\sigma_\gamma^2}I + \frac{1}{\sigma^2}\tilde{X}^T\tilde{X}\right)^{-1} \tag{41}$$

- $\sigma^2|\beta,\gamma,y$: follows from Bayesian inference of normal distribution with known mean but unknown variance. Let v be the mean squared error:

$$v = \frac{1}{n}\sum_{i=1}^{n}(y_i-\beta_1 x_i-\beta_0)^2 \tag{42}$$

Then σ^2 follows:

$$\sigma^2|\beta,\gamma,y \sim \text{Inv-}\chi^2(n,v) \tag{43}$$

Latent variable model with gene-environment interactions:

- Model: suppose we have a covariate U for some environmental variable, and we wish to test if the gene has interaction with U. Our model of Z is the same as before, but for phenotype, we have:
$$Y_i = \beta_1 Z_i + \beta_2 U_i + \beta_3 Z_i U_i + \epsilon_i \tag{44}$$
Following the same procedure, we integrate out Z_i:
$$y_i | x_i, u_i, \beta, \gamma \sim N\left((\beta_1 + \beta_3 u_i) x_i \gamma + \beta_2 u_i + \beta_0, (\beta_1^2 + \beta_3^2 u_i^2)\sigma_z^2 + \sigma^2\right) \tag{45}$$

3. Inference of latent variable model: binary phenotypes

Model of binary phenotypes:

- Model:
$$P(Y = 1 | Z, \beta) = \sigma(\beta_1 Z + \beta_0) \tag{46}$$
- Reformulating the model: follow the common notations, we are solving a logistic regression problem of $X \to Y$:
$$P(Y = 1 | X, \beta) = \sigma(\beta_1 x + \beta_0) \tag{47}$$
We assume β_1, β_0 are known, and $X \sim N(\mu, \sigma^2)$. Our goals are: (1) the posterior distribution $X|Y, \beta$; (2) the model evidence:
$$p(y) = \int p(x) p(y|x) dx \tag{48}$$
We note that the problem is very similar to Bayesian logistic regression, where parameters follow normal prior.
- Posterior distribution $X|Y, \beta$: the log. posterior:
$$\begin{aligned} \ln p(x|y, \beta) &= \ln p(x) + \ln p(y|x, \beta) + \text{const} \\ &= -\tfrac{1}{2\sigma^2}(x - \mu)^2 + y(\beta_1 x + \beta_0) - \ln(1 + \exp(\beta_1 x + \beta_0)) + \text{const} \end{aligned} \tag{49}$$
We use the normal approximation of the posterior distribution. Suppose $\hat{x}$ maximizes the log-posterior function above. We next find the Fisher information at $\hat{x}$: taking the second derivative of x:
$$-\frac{d^2}{dx^2} \ln p(x|y, \beta) = \frac{1}{\sigma^2} + \pi(1 - \pi)\beta_1^2 \tag{50}$$
where $\pi = P(y = 1|x, \beta) = \sigma(\beta_1 x + \beta_0)$. Thus $x|y, \beta$ is approximately normal:
$$x|y, \beta \sim N\left(\hat{x}, (1/\sigma^2 + \hat{\pi}(1 - \hat{\pi})\beta_1^2)^{-1}\right) \tag{51}$$
- Theorem: we have the following approximation for sigmoid function of normal random variable (see [Bishop, Bayesian logistic regression]). Suppose $X \sim N(\mu, \sigma^2)$, and $\sigma(\cdot)$ is the sigmoid function, then
$$\int \sigma(x) p(x) dx \approx \sigma\left(\frac{\mu}{\sqrt{1 + \pi\sigma^2/8}}\right) \tag{52}$$

- Evidence/predictive distribution: we want to solve:

$$p(y=1) = \int_{-\infty}^{+\infty} N(x|\mu,\sigma^2)\sigma(\beta_1 x + \beta_0)dx \tag{53}$$

We perform variable substitution: $t = \beta_1 x + \beta_0$, then $x = (t-\beta_0)/\beta_1$, and $dx = dt/\beta_1$. Plug in these terms:

$$p(y=1) = \int_{-\infty}^{+\infty} N(t|\beta_1\mu + \beta_0, \beta_1^2\sigma^2)\sigma(t)dt \approx \sigma\left(\frac{\beta_1\mu + \beta_0}{\sqrt{1+\pi\beta_1^2\sigma^2/8}}\right) \tag{54}$$

Approximate inference of latent variable model:

- Posterior mode: Let $\phi = (\sigma_\gamma^2, \sigma_z^2)$ be parameters of the model of Z, the posterior distribution:

$$p(\phi,\beta|X,Y,w) \propto p(\phi)p(\beta)\prod_{i=1}^{n}\int p(z_i|X_i,w,\phi)p(y_i|z_i,\beta)dz_i \tag{55}$$

We assume that we have noninformative prior of (ϕ,β). The integral can be calculated using the results above, and so given any (ϕ,β), the posterior can be efficiently calculated. Any numerical optimization algorithm can be used to find the posterior mode.

- Normal approximation based on posterior mode: compute the second partial derivative of $\ln p(\phi,\beta|X,Y,w)$ numerically and apply the normal approximation.
- Gibbs sampling: we are interested in the conditional distributions (let $D = (X,Y,w)$ be the observations):
  - $p(\beta|D,Z,\phi) = p(\beta|X,Z)$: the usual logistic regression
  - $p(\phi|D,\beta,Z) = p(\phi|X,w,Z)$: normal distribution of unknown variance but known mean
  - $p(Z|D,\beta,\phi)$: logistic regression with uncertain data. Approximate normal distribution.
- EM algorithm for posterior mode: in the E-step, suppose we have $\theta^{(t)} = (\phi^{(t)},\beta^{(t)})$, we compute:

$$\begin{aligned} Q\left(\theta|\theta^{(t)}\right) &= \log P(\phi,\beta) + \textstyle\sum_{i=1}^{n} \mathrm{E}_{z_i|y_i,X_i,w_i,\theta^{(t)}}\left[\log P(z_i|X_i,y_i,w_i,\theta)\right] \\ &= \log P(\phi,\beta) + \textstyle\sum_{i=1}^{n} \mathrm{E}_{z_i|y_i,X_i,w_i,\theta^{(t)}}\left[\log P(z_i|X_i,w,\phi) + \log P(y_i|z_i,\beta)\right] \end{aligned} \tag{56}$$

Note that $\log P(y_i|z_i,\beta)$ term is not quadratic, and its expectation over z_i (normal distribution) cannot be obtained analytically. We could apply the approximation with logistic regression of uncertain data to estimate the expectation, however, then the maximization of $Q(\cdot)$ still has to be done numerically.

Issues:

- The sign problem: a variant may occasionally have protective effect on the phenotype. How should this be modeled? In general, we can define another hidden (binary) variable per variant, U_j, and U_j can be part of the inference. Heuristically, we could estimate the effect direction of each SNP, and then fix U_j.

  - Remark: protective mutations are rare in practice for human diseases.
- Diploid genotype: each individual has two copies of a gene, may need to model the dominance and dosage effect. Ex. if a gene is haplosufficient, then even if one copy is disrupted, the other may still be functional.
- Hypothesis testing approach: to evaluate the approach (type I error and power), need to have a test of the key coefficients (equal to 0 or not). For Bayesian approach, this is effectively using the posterior interval of the coefficients (decision rule: reject H_0 if 0 is not in the posterior interval at level α). It is also possible to develop a traditional test of the coefficients β.
  - See Errors-in-variables models.
- Correction for multiple hypothesis testing (MHT).

4. Latent variable model with interactions

   Latent variable model with gene-gene interactions:

   - Model: suppose we have p SNPs, $X_1, \cdots, X_p$, some of which belong to the first group, G_1, (e.g. Domain 1 of a gene), and the rest belong to the second group, G_2 (e.g. Domain 2). Suppose SNPs in G_1 affect a latent variable Z_1, and similarly G_2 SNPs affect Z_2:

   $$\mathrm{E}\left(Z_k|X,\gamma\right) = \sum_{j\in G_k} \gamma_j X_j \qquad k = 1,2 \tag{57}$$

   The response variable Y is a GLM of Z_1 and Z_2 with possible interaction:

   $$\mathrm{E}\left(Y|Z_1,Z_2\right) = g^{-1}\left(\tau_0 + \tau_1 Z_1 + \tau_2 Z_2 + \mu Z_1 Z_2\right) \tag{58}$$

   where μ is the interaction term. Similar to the previous models, priors on γ_j can be defined, based on the information of SNPs (conservation, MAF, etc.).
   - Remark: one can also define a hierarchical model which eliminate the latent variables. However, considerably more coefficients will be involved, for every j, k, where $X_j \in G_1$ and $X_k \in G_2$, we will need an coefficient β_{jk} (regularized, prior distribution depends on the effect of j on Z_1, k on Z_2 and the interaction). It is probably computationally difficult to solve this model.

   Inference of latent variable model with interactions:

   - Logistic regression with uncertain data: we consider the model of logistic regression with two explanatory variables:

   $$P(Y = 1|X_1, X_2, \beta) = \sigma(\beta_1 X_1 + \beta_2 X_2 + \beta_{12} X_1 X_2 + \beta_0) \tag{59}$$

   where $X_1 X_2$ is the interaction term. The explanatory variables have prior distributions:

   $$X_k \sim N(\mu_k, \sigma_k^2) \qquad k = 1,2 \tag{60}$$

   And our goal is similar to before: (1) posterior distribution $X_1, X_2|Y, \beta$; and (2) the model evidence.

- Posterior distribution: apprxoimate the posterior by normal distribution. First, we find the posterior mode.

$$\ln p(x_1, x_2|y, \beta) \quad = \ln p(x_1) + \ln p(x_2) + \ln p(y|x_1, x_2, \beta) + \text{const} \tag{61}$$

  Note that when x_1 is given, the distribution of x_2 can be reduced to the case we have before; and similarly for the distribution of x_1. This implies a conditional maximization algorithm for maximization; and Gibbs sampling.

- Evidence/predictive distribution: the model evidence is:

$$p(y=1) = \int\int N(x_1|\mu_1, \sigma_1^2) N(x_2|\mu_2, \sigma_2^2) \sigma(\beta_1 x_1 + \beta_2 x_2 + \beta_{12} x_1 x_2 + \beta_0) dx_1 dx_2 \tag{62}$$

  We could use repeated integration:

$$p(y=1) = \int N(x_2|\mu_2, \sigma_2^2) dx_2 \int N(x_1|\mu_1, \sigma_1^2) \sigma((\beta_1 + \beta_{12} x_2) x_1 + \beta_2 x_2 + \beta_0) dx_1 \tag{63}$$

  The latter integral is a function of x_2, and can be found using our previous result:

$$\int N(x_1|\mu_1, \sigma_1^2) \sigma((\beta_1 + \beta_{12} x_2) x_1 + \beta_2 x_2 + \beta_0) dx_1 \approx \sigma\left(\frac{(\beta_1 + \beta_{12} x_2)\mu_1 + \beta_2 x_2 + \beta_0}{\sqrt{1 + \pi(\beta_1 + \beta_{12} x_2)^2 \sigma_1^2 / 8}} \right) \tag{64}$$

Predicting phenotypes from network states:

- Background: suppose we are given the states of all genes (whether a gene is disrupted or not) for each sample: they can be predicted from sequencing data, or are treated as latent variables. And we also have a phenotype state of each sample. Can we develop a predictive model from gene states to the phenotype?
- Problem: let X be the gene states, X_i is 0 if a gene is normal, 1 if a gene is disrupted. The genes are organized into pathways/networks, G, thus we want to predict Y from the network state $X(G)$.
- Remark: while similar models are studied in other regression problems (features are structured), the focus here is to learn the non-linear aspect of the model (as oppose to e.g. adjacent genes should have similar coefficients). The informative patterns are, for example, two pathways are disrupted at the same time; the number of disrupted genes in a pathway exceeding a threshold; etc.

5. Experiment design

   Tasks:

   - Gene effect test: one gene only, and test whether the gene has any effect on phenotype.
   - Interaction test: two or more genes, and test whether there is gene-gene interaction.
     - Two genes: no marginal effect
     - Two genes: marginal effect
     - Mulitple genes: single interacting pair

  - Multiple genes: multiple interacting pairs

Simulation of sequences/variants:

- Simple: similar to [Lin11], 10-20 rare variants in a locus, in LE.
- Complex: population genetics simulation of sequences, and limit to variants using MAF and MAF_{total} cutoff [Morris10]
- Complex: take real sequences from 1000 Genome or any real data used for evaluation [Yu12].
- Question: some studies simulate the descendents of real sequences [Zhang07], why is this necessary?

Genetic model:

- Simple: γ_j (impact of a SNP) is either 0 (neutral) or 1 (functional), with the ratio of 0 vs. 1 determined by some population genetic estimate.
- Complex: a SNP is either neutral (0) or functional; if functional, then γ_j follows some distribution in $[0, 1]$, e.g. uniform distribution. Then we have, e.g. 10 levels $(0.1, 0.2, \cdots, 2.0)$, with 2 SNPs per level [Wu11].
- Question: possible to get a distribution of γ of all functional variants? See [1000 Genome Project] and other papers that estimate the percent of neutral vs functional mutations in population genetic data.

Prior knowledge: need to specify w for all the variants to be used as part of the input of the program

- Under the simple genetic model: w is either 0 or 1 and the misclassification rate (when $w_j \neq \gamma_j$) matches the rate from practical estimate.
  - If w is based on MAF: e.g. $w_j = 1$ iff $p_j < 0.01$. It might be possible to get the estimate of the misclassification rate using population genetics.
  - If w is based on PolyPhen scores: the misclassification rates reported by PolyPhen.
- Under the complex genetic model: $w_j \sim N(\gamma_j, \sigma_\gamma^2)$, different levels of variance corresponding to how accurate our prior knowledge is. Apply truncation s.t. $w_j \in [0, 1]$.

Comparison with other methods:

- Collapsing method: use presence/absence or the number of rare variants. Could use this for both gene test and interaction test.
- Weighted collapsing method: use $\sum_j w_j X_j$ as the explanatory variable and do regression. Find the reference [Wei Pan's work?]
- minP-SNP test: for gene effect, use the min. p-values of all SNPs in a gene.
- minP-pair test: for interaction effect, use the min. p-values of all SNP pairs between two genes.
- Question: what is the decision rule? Bonferroni correction [Morris10]: reject H_0 if $minP < \alpha/p$, where p is the number of SNPs. This is too conservative. Do permutation/resampling and obtain the null distribution of $minP$.

Evaluation:

- Type I error:
  - For Bayesian approach: obtain the posterior interval (I) of τ at level α, if $0 \notin I$, then reject H_0.
  - Question: really need to evaluate type I error at certain level α?
- Power:

Real data:

- Comparison with other methods: suppose we have one pair (or a single gene) that is biologically valid
  - Comparison of p-values: compare the p-values of this pair (gene) under different methods.
  - Power assessment: resampling of data, and assess the power of different methods [Yandell11].
- Pancreatitis: inflmmation at pancreas, but not autoimmune disease. Main function affected: secretion of trpsin for digesting protein (not insulin production). Existing studies
  - GWAS: identify two genes with genome-wide significance. CFTR and CLDN2 genes. A few other genes slightly below the threshold may be interesting. Related to calcium metabolism (?)
  - Sequencing data: of about 5 candidate genes.

6. Combined analysis of case-control and de novo mutation data

   Motivation:

   - Trio data: suppose we have sequencing data from N_{trio} trios (normal parents and one disease child). The sequencing data allow us to identify all the de novo mutations, i.e. mutations in the child but not present in the parents. If a de novo mutation is common in the families, then it is likely that the gene would be causal to the disease.
   - Combining trios with case-control data: the same de novo mutations may appear in the case-control data, and the enrichment in the cases is another sign of the association of the gene to the disease. Thus in both datasets, the frequency of the mutations depend on the (common) effect size of the mutations, and we can use all these data to infer the effects.
   - Example: 500 autism trios with 2 loss-of-function (LOF) mutations in some given gene, and in 1000 cases and 6000 controls, the same gene may have 6 and 3 LOF mutations respectively. Is this gene associated with autism?

   Model:

   - Genetic model: for LOF mutations, the homozygotes are extremely rare, so we assume there are only two genotypes: wild type (A) and heterozygote LOF mutations (a). Note that we collapsed all the LOF mutations of a gene into a single genotype, as all these

mutations presumably have the same functional effect. Let f_A and f_a be the penetrance of the two genotypes respectively, and g_A, g_a be the population frequency of the two respectively. The disease prevalence is given by:

$$K = f_A g_A + f_a g_a \tag{65}$$

- Model of case-control data: this follows from the standard 2×2 table analysis. Suppose p_i and q_i are the frequency of the i-th genotype in cases and in controls, respectively. They are related to the population frequency and penetrance by (using Bayes Theorem):

$$p_i = \frac{f_i g_i}{K} \qquad q_i = \frac{(1-f_i)g_i}{1-K} \tag{66}$$

  Let X_{case} be the number of a genotypes in the cases, and X_{control} be the number of a genotypes in the controls, then they follow binomial distributions:

$$X_{\text{case}} \sim \text{Bin}(N_{\text{case}}, p_a) \qquad X_{\text{control}} \sim \text{Bin}(N_{\text{control}}, q_a) \tag{67}$$

  where N_{case} and N_{control} be the total number of cases and controls respectively.
- Model of trio data: suppose we are given a normal pair of parents with a disease child. Let x be genotype and y be phenotype of an individual (Disease or Unaffected). We are interested in the probability of de novo mutaiton given the trio:

$$p_{dn} = P(x_p = AA, x_c = a | y_p = UU, y_c = D) \tag{68}$$

  where c and p represent child and parents respectively. Given normal parents, the genotypes of parents and child follow into four cases, and conditioned on $y_p = UU$, their probabilities are given by (assuming the mutation rate per gene copy is μ):
  - $x_p = AA, x_c = A$: no mutation. Probability $q_A(1-2\mu)$.
  - $x_p = AA, x_c = a$: de novo mutation. Probability $2q_A\mu$.
  - $x_p = Aa, x_c = A$: no mutation. Probability $q_a(1-\mu) \cdot \frac{1}{2} = \frac{q_a}{2}(1-\mu)$.
  - $x_p = Aa, x_c = a$: inherited mutation or de novo mutation from one of the parent. Probability $q_a(1-\mu) \cdot \frac{1}{2} + q_a\mu = \frac{q_a}{2}(1+\mu)$.

  Note that we assume $x_p = aa$ is extermely rare, and we ignore this case. So the probability of $x_p = AA$ is q_A instead of q_A^2. This allows us to write the probability:

$$\begin{aligned} P(y_c = D | y_p = UU) &= \textstyle\sum_{x_p, x_c} P(x_p, x_c, y_c = D | y_p = UU) \\ &= \textstyle\sum_{x_p, x_c} P(x_p | y_p = UU) P(x_c | x_p) P(y_c = D | x_c) \end{aligned} \tag{69}$$

  Plug in the relevant terms, we have:

$$P(y_c = D | y_p = UU) = q_A(1-2\mu)f_A + 2q_A\mu f_a + \frac{q_a}{2}(1-\mu)f_A + \frac{q_a}{2}(1+\mu)f_a \tag{70}$$

  The second term out of the four terms above gives the desired de novo probability:

$$p_{dn} = \frac{2q_A\mu f_a}{q_A(1-2\mu)f_A + 2q_A\mu f_a + \frac{q_a}{2}(1-\mu)f_A + \frac{q_a}{2}(1+\mu)f_a} \tag{71}$$

  We could use the relationship between q and f to write the de novo probability in terms of f and g (not shown). Suppose we observed among N_{trio} trios, there are X_{trio} de novo mutations in the children, then we have:

$$X_{\text{dn}} \sim \text{Bin}(N_{\text{trio}}, p_{dn}) \tag{72}$$

Inference:

- Parameterization and approximation: we could parameterize using q_a and f_a/f_A (relative risk). For simplicity of notations, we will call the two parameters q and γ respectively. The other parameters can be related to these two parameters. First, we have $f_a = \gamma f_A$. We assume f_A is known, approximately equal to K. Next, we could write p_a as (using $f_A = K$):

$$p_a = q_a \frac{f_a}{1 - f_a} \frac{1 - K}{K} = q\gamma \frac{1 - K}{1 - \gamma K} \tag{73}$$

  And:

$$q_A = 1 - q \tag{74}$$

  Finally, we approximate p_{dn}, noting that $1 - \mu \approx 1 + \mu \approx 1$:

$$p_{dn} \approx \frac{2(1 - q)\mu\gamma}{(1 - q) + 2(1 - q)\mu\gamma + q(\gamma + 1)/2} \tag{75}$$

  For LOF mutations or rare variants with frequency below 0.01, $1 - q \approx 1$. And the relatively risk is typically less than 100, thus $2\mu\gamma << 1$. We have:

$$p_{dn} \approx \frac{2\mu\gamma}{1 + 2\mu\gamma + q\gamma/2} \approx \frac{2\mu\gamma}{1 + q\gamma/2} \tag{76}$$

- Hypothesis testing: we assume that q can be estimated from the control data (usually many more). Our model can be summarized as:

$$X_{\text{case}} \sim \text{Bin}\left(N_{\text{case}}, q\gamma \frac{1 - K}{1 - \gamma K}\right) \qquad X_{\text{dn}} \sim \text{Bin}\left(N_{\text{trio}}, \frac{2\mu\gamma}{1 + q\gamma/2}\right) \tag{77}$$

  We are testing $H_0 : \gamma = 1$ vs. $H_1 : \gamma > 1$. Under H_0:

$$X_{\text{case}} \sim \text{Bin}\left(N_{\text{case}}, q\right) \approx \text{Poisson}(N_{\text{case}} q) \tag{78}$$

$$X_{\text{dn}} \sim \text{Bin}\left(N_{\text{trio}}, 2\mu\right) \approx \text{Poisson}(2N_{\text{trio}}\mu) \tag{79}$$

  Note that under H_0, the gene is not associated with the phenotype, thus X_{dn} should only be determined by mutation rate 2μ. Since the sum of two independent Poisson random variables is Poisson, we choose our test statistic as:

$$T = X_{\text{case}} + X_{\text{dn}} \sim \text{Poisson}(N_{\text{case}} q + 2N_{\text{trio}}\mu) \tag{80}$$

  under H_0.

- Estimating relative risk: from Equation 77, we could do maximum likelihood parameter estimation (no closed form). When q is small and γ is not too large, we could approximate the binomial distributions using Poisson, and assume $1 - \gamma K \approx 1 - K$:

$$X_{\text{case}} \sim \text{Poisson}(N_{\text{case}} q\gamma) \qquad X_{\text{dn}} \sim \text{Poisson}(2N_{\text{trio}}\mu\gamma) \tag{81}$$

  The sum of the two is Poisson, and we can get the estimator of γ as:

$$\hat{\gamma} \approx \frac{X_{\text{case}} + X_{\text{dn}}}{N_{\text{case}} q + 2N_{\text{trio}}\mu} \tag{82}$$

- Examples: suppose we have 1 LOF mutation in 2000 controls, thus $q \approx 5.0 \cdot 10^{-4}$, and the mutation rate is about 10^{-5}. Suppose $N_{\text{case}} = 1000$ and $N_{\text{trio}} = 500$. We compute the p-values under different settings:
  - De novo data only: 1 de novo LOF mutation, $p = 10^{-2}$; 2 de novo LOF mutations, $p = 5.0 \cdot 10^{-5}$; 3 de novo LOF mutations, $p = 1.6 \cdot 10^{-7}$.
  - Case-control data only: 3 LOF, $p = 0.01$; 4 LOF, $p = 1.7 \cdot 10^{-3}$; 5 LOF, $p = 1.7 \cdot 10^{-4}$; 6 LOF, $p = 1.4 \cdot 10^{-5}$.
  - Both types of data: we only consider the problem if having 1 de novo LOF mutation adds the evidence in case-control data. Suppose there are 4 LOF in case-control, then having 1 de novo LOF makes $p = 1.7 \cdot 10^{-4}$; 5 LOF in case-control, having 1 de novo LOF makes $p = 1.4 \cdot 10^{-5}$; etc.
- Remark: the model assumes q is given, how to relax this assumption?

Real genes:

- Settings: we analyze some genes in real data: $N_{\text{case}} = 1000$, $N_{\text{trio}} = 500$. The gene mutation rate is calculated from point mutation rate and gene size, adjusted by the local GC content. The fraction of LOF mutations in point mutations is 10 / 161. We call μ the de novo LOF mutation rate, and p_{CC}, p_{dn}, p_{combine} the p-values of case-control, de novo and combined test, respectively.
- CHD8: $X_{\text{case}} = 3$, $X_{\text{control}} = 0$ (pseudocount 0.5) in 6000 controls, $X_{\text{dn}} = 1$, $\mu = 1.4 \cdot 10^{-5}$, we have: $p_{\text{CC}} = 9.0 \cdot 10^{-5}$, $p_{\text{dn}} = 0.014$, and $p_{\text{combine}} = 3.4 \cdot 10^{-6}$.
- KATNAL2: $X_{\text{case}} = 3$, $X_{\text{control}} = 3$ in 6000 controls, $X_{\text{dn}} = 2$, $\mu = 2.2 \cdot 10^{-6}$, we have: $p_{\text{CC}} = 0.014$, $p_{\text{dn}} = 3.2 \cdot 10^{-6}$, and $p_{\text{combine}} = 1.7 \cdot 10^{-4}$.
- COL25A1: $X_{\text{case}} = 3$, $X_{\text{control}} = 0$ (pseudocount 0.5) in 1000 controls, $X_{\text{dn}} = 1$, $\mu = 3.5 \cdot 10^{-6}$, we have: $p_{\text{CC}} = 0.014$, $p_{\text{dn}} = 3.5 \cdot 10^{-3}$, and $p_{\text{combine}} = 1.7 \cdot 10^{-3}$.

Mutation rate:

- The mutation rate: for LOF mutations, we only consider point mutations (stop codons or splice sites), as frameshift mutations are less frequent and harder to model. The LOF point mutations are calculated from: (1) the number of point mutations per gene is equal to the basic mutation rate per bp, multiplied by the gene length and corrected for local GC content. (2) multiply this number by the fraction of LOF events over all point mutations (estimated from all genes).
- Remark: Mutation-selection balance may be useful for the mutation rate estimation and incorporating selection. Given a gene with two genotypes, at equilibrium, the population frequency of th disadvantegous genotype is:

$$q_e = \mu / s \tag{83}$$

  where μ is the mutation rate and s is the selection coefficient (the two genotypes have selection 1 and $1 - s$ respectively).
- Reference: [Rate, molecular spectrum, and consequences of human mutation, PNAS, 2010]

7. Using severity scores for gene test

   Severity score distribution:

   - Motivation: suppose we have a severity score for every rare variant. Then we could have a total severity score per individual (the sum of the scores, but most people have at most one severe mutation anyway). If we draw the distribution of the severity scores of cases and controls, we should expect that cases tend to have a distribution that is skewed towards more severe mutations than controls. In addition, most people would have zero-scores (zero-inflation). Can we develop a test based on this idea?
   - Modeling the severity score distribution: suppose Y is the phenotype label (1 for case, 0 for control), and Z is the severity score. We want to compare the distribution $P(Z|Y = 1)$ vs. $P(Z|Y = 0)$. However, it does not have an obvious parametric form, so we use Bayes Theorem, as $P(Y = 1|Z)$ is the penetrance of the mutation:

     $$P(Z|Y = 1) = P(Y = 1|Z)\frac{P(Z)}{P(Y = 1)} \tag{84}$$

     And similarly for $P(Z|Y = 0)$. We assume a simple model of the penetrance:

     $$P(Y = 1|Z) = \sigma(\beta_1 Z + \beta_0) \tag{85}$$

     The likelihood is given by:

     $$\prod_i p(z_i|y_i) = \prod_i \frac{p(z_i)}{p(y_i)} \prod_i p(y_i|z_i) \propto \prod_i p(y_i|z_i) \tag{86}$$

     The term $p(y_i)$ is simply disease prevalance, and the term $p(z_i)$ is the same for cases and controls. Thus the model is equivalent to logistic regression of y on z.

   Modeling genotype distributions:

   - An alternative model of the distribution of Z is:

     $$Z = \sum_j w_j X_j \tag{87}$$

     where X_j is the genotype at the j-th SNP, and w_j is the severity score of the mutation X_j. Thus Z is a sum of Bernoulli random variables. The zero-inflation comes from the fact that for any individual, X_j's are mostly equal to 0. However, since X_j's are observed, all the information in Z is in X_j, and we may just want to model X_j distribution.
   - Genotype model: the simplest model assumes that X_j follows Bernoulli distributions (two alleles per site):

     $$X_j|Y = 1 \sim \text{Bernoulli}(p_j) \qquad X_j|Y = 0 \sim \text{Bernoulli}(q_j) \tag{88}$$

     where p_j and q_j are the frequency of the rare variants in cases and controls respectively. Suppose N^{case} and N^{control} are the number of cases and controls, respectively, and for

the j-th site, let N_j^{case} and N_j^{control} be the number of rare variants in caseas and controls, respectively. The likelihood:

$$P(D|p,q) = \prod_j p_j^{N_j^{\text{case}}} (1-p_j)^{N^{\text{case}} - N_j^{\text{case}}} q_j^{N_j^{\text{control}}} (1-q_j)^{N^{\text{control}} - N_j^{\text{control}}} \tag{89}$$

The simplest approach is to test $H_0 : p_j = q_j, \forall j$ (the approach by [VAAST]). However, this test suffers from a high dof. We can use the severity scores to reduce the complexity of the model. We know that:

$$\frac{p_j}{q_j} = \frac{f_j}{1-f_j}\frac{1-K}{K} = \frac{1-K}{K}\frac{f_{j0}}{1-f_{j0}} \cdot \text{OR}_j \tag{90}$$

where f_j is the penetrance of the rare variant of the j-th site, f_{j0} is the penetrance of the reference allele at the j-th site, K is the disease prevalence, and OR_j is the odds ratio of the rare variant. Clearly, OR_j depends on the severity w_j, and this can be modeled as a simple linear function (assuming w_j is properly scaled):

$$\text{OR}_j = \beta w_j \tag{91}$$

Since w_j is given, the only parameter of the model is β, a great reduction of model complexity.

- Remark: this model effectively tests if the penetrance f_j (or odds ratio) has a monotonic relationship with the severity score w_j. When f_j (or OR_j) is observed, this is a standard problem (since w_j is known); but since they are unknown parameters, we need a new model.
- Remark: the VAAST model also uses the (kind of) severity score in the model. The model defines the fraction of a given mutation in neutral vs. disease proteins, and use the ratio in the likelihood. However, this apporach only weighs the different sites (s.t. the more severe sites are weighed more heavily), but does not achieve reduction of model complexity.

2 Genomics

1. Predicting TF targets from PBM data

   Lessons from PBM data: from the original PBM paper, the Linear model [Annala et al.] and BEEML-PBM papers.

   - Positional independence and PWM-based model: relaxing independence assumption is probably important. The simple linear model and the model using E value or median intensity outperform methods based on PWM. The finding of alternative binding motif also supports the positional dependence.
   - Multiple sites (flanking sequences): this is important, supported by (1) the observation that the same K-mer is associated with a broad range of intensity; (2) the results from linear model and BEEML. However, the binding of multiple sites at the same time is probably not important (BEEML).
   - Positional effect: supported by the BEEML paper (Figure S6).
   - Background noise: for many probes, intensity mostly due to background.
   - Gapped motifs: very small influence on top of the Linear model. However, this does not mean the gapped motifs are not important, as the linear model already includes 4-mers and 5-mers, which may cover some gapped motifs.
   - Short K-mers: in the Linear model paper, show that if only include 6-mers and 7,8-mers, the model does not perform as well as the full model (however, the effect is not reported).

   Review of existing methods:

   - E-value or HMIK (highest median intensity K-mer): large variation of intensities of probes containing a K-mer, thus not reliable estimation; in particular, low affinity K-mers may receive high E-values or HMIK scores.
   - Linear model: very flexible with K-mers of differnet lengths, thus capturing secondary motifs, gapped motifs, etc, and possibly background signals (i.e. some short K-mers may increase/decrease binding in a TF-nonspecific fashion). The main drawback is that the learned coefficients have no physical interpretation, and difficult to apply the model to longer sequences.
   - BEEML: positional independence assumption. The benefit is that it can be easily generalized/extended to longer sequences.

   Strategies for predicting TF targets using PBM data: may need to apply a model learned from PBM to longer sequences, because a target sequence may harbor multiple sites of the TF (homotypic clustering, a well-known phenomenon).

   - Single site-based model: suppose we have some statistics of single sites/K-mers, e.g. E-values. When applying the K-mer models, we score a region based on all putative binding sites in the region, i.e. combining the E-values of top K-mers. If the distribution of the statistics under the alternative model (true site) is known, then some form of LRT can be developed; if not, then use methods such as Fisher's product method.

- Probe model: scan a putative region with the probe model (35-mer). Options of combining the probe-level scores: (1) best window (probe); (2) the "average" of the top d (d is a parameter) windows; etc.
- Biophysical model: relax the positional independence assumption. The model learned can be directly applied to longer sequences: if binding affinity of any K-mer is known, then the affinity to the entire sequence can be computed easily.

Baseline methods:

- Best site: the score of a sequence is the E-value of the best K-mer.
- Best probe: scan the sequence with the probe model, and choose the best one.
- Summing sites by intensities: [Zhu & Bulyk, Genome Res, 2009] consider all K-mers with E-score above a threshold, and sum the median intensities of all the K-mers.
- Summing sites by E or p-values: suppose we first transform E-values to p-values, then define the test statistic on the top m K-mers:

$$T = -2\sum_{i=1}^{m} \log p_i \tag{92}$$

  where p_i is the p-value of the i-th K-mer. T follows χ^2 distribution with dof. m.

Biophysical modeling of PBM data without independence assumption:

- BEEML-PBM model [Zhao & Stormo, NBT, 2011]: suppose $F(i)$ is the binding probability of the i-th probe, we have:

$$F(i) = \sum_{j=1}^{L} F_{\text{pos}}(j)P(j) \tag{93}$$

  where $P(j)$ is the probability of binding at the j-th position of the probe and it is related to the position weight matrix. Suppose S_j is the K-mer at the j-th position, and $\bar{S}_j$ is its reverse complement (in the other strand), then:

$$P(j) = P(S_j) + P(\bar{S}_j) - P(S_j)P(\bar{S}_j) \tag{94}$$

  where $P(S_j)$ is the binding probability of the sequence S_j. This follows from the Principle of Inclusion-Exclusion (a TF molecule cannot occupy both strands at the same time). The model aims to minimize the objective function (of the parameters of the PWM):

$$O(\epsilon, \mu) = \sum_i W_i(Y_i - a - cF(i))^2 + \lambda \sum_{b=A}^{T} \sum_k \epsilon(b,k)^2 \tag{95}$$

  where W_i and Y_i are the weights and the observed intensity of the i-th probe, respectively.
- Linearization of the BEEML-PBM model: note that the model contains quadratic terms, $P(S_j)P(\bar{S}_j)$ if we use $P(S_i)$ as basic parameters. Instead, for a given K-mer S_j, we define the following coefficient:

$$\beta_j = P(S_j) + P(\bar{S}_j) - P(S_j)P(\bar{S}_j) \tag{96}$$

where $P(S_j)$ is defined as before. Clearly, we have the constraint $0 \leq \beta_j \leq 1$ for any j. Then we should have the constraint:

$$\beta_j = \beta_{C(j)} \tag{97}$$

where $C(j)$ is the (index of) the reverse complement of the j-th K-mer. This means that for 4^K possible K-mers, we only need $4^K/2$ free parameters. For the i-th probe, we define the features X_{ij} for any K-mer S_j. Suppose S_j occurs in k positions in the i-th probe, with positions $p_1, \cdots, p_k$ (one strand only), then the feature is defined as:

$$X_{ij} = \sum_k F_{\text{pos}}(p_k) \tag{98}$$

where $F_{\text{pos}}(p_k)$ is the positional correction. When S_j does not occur in the i-th probe, $X_{ij} = 0$. We have the following regression for the i-th probe:

$$Y_i = a + c\left(\sum_j \beta_j X_{ij}\right) + \epsilon_i \tag{99}$$

Note that c and β_j are coupled, and we cannot estimate them individually (for any estimated parameters, we can always scale them without violating the constraints). So we simply write the regression as a linear model:

$$Y_i = \beta_0 + \sum_j \beta_j X_{ij} + \epsilon_i \tag{100}$$

subject to the constraint $\beta_j \geq 0$. To fit the model, we would need weighted least square (plus regularization, below).

- Relating to the physical parameters: Suppose a model is fitted, we could define $c = \max_j \beta_j$, and then $\beta'_j = \beta_j/c$ is the binding probability of the j-th K-mer (averaging over both strands). However, it is not possible to distinguish S_j and its reverse complete $\bar{S}_j$ from β_j alone, thus not possible to solve $P(S_j)$ from β_j.
- Prior of β: suppose we fit the model with the independence assumption, i.e. the original BEEML-PBM model. From this, we can estimate the binding probabilities $P(S_j)$ and $P(\bar{S}_j)$, and:

$$\alpha_j = c\left[P(S_j) + P(\bar{S}_j) - P(S_j)P(\bar{S}_j)\right] \tag{101}$$

When $j = 0$, we have $\alpha_j = a$. We have a regularization term to model the idea that β_j should be close to α_j in general. This leads to the following problem, minimize:

$$f(\beta) = \sum_i W_i(Y_i - X_i\beta)^2 + \lambda \sum_j (\beta_j - \alpha_j)^2 \tag{102}$$

subject to $\beta_j \geq 0$. Note that $\alpha_j \geq 0$, and we want β_j to be close to α_j, so the positivity constraint of β is not important. Also we could define $X'_i = \sqrt{W_i}X_i$ and $Y'_i = \sqrt{W_i}Y_i$, so the weight terms can be easily incorporated into X' and Y'. Ignoring the positivity constrait, we have:

$$\frac{\partial f(\beta)}{\partial \beta} = 2X'^T(Y' - X'\beta) + 2(\beta - \alpha) = 0 \tag{103}$$

Solving this equation:

$$\beta = (X'^T X' - I)^{-1}(X'^T Y' - \alpha) \tag{104}$$

- Applying the model: the simple approach is to apply the linear model, $\beta_0 + X\beta$, where X is the feature vector of the new sequence. If sites are non-overlapped, suppose there are n putative sites, and the occupancy status per site is $X_1, \cdots, X_n$, then:

$$\mathrm{E}(N) = \mathrm{E}(X_1 + \cdots + X_n) = \sum_i \mathrm{E}(X_i) = \sum_i \beta(S_i) \tag{105}$$

  For longer sequence (e.g. 100-200 bp), however, it may be important to consider the case where multiple sites in the sequence are occupied by the TF molecules simultaneously. Example: a sequence with two strong sites. The probability of each site occupied by the TF is β_1 and β_2 respectively. Then the total affinity of the sequence is the expected number of TF molecules occupied:

$$F(S) = \beta_1 + \beta_2 + 2\beta_1\beta_2 \tag{106}$$

  This problem can be solving using dynamic programming [He et al, A Biophysical Model for Analysis of Transcription Factor Interaction and Binding Site Arrangement from Genome-Wide Binding Data, PLoS ONE, 2009]

- Alternative ways of regularization:
  - L_1 regularization: biologically, many K-mers could have non-zero coefficients (binding affinity is a continuous variable). And experiments seem to suggest that. So L_1 regularization may not be the most appropriate.
  - Neighboring K-mers: if two K-mers have similar sequences, their binding affinity should be close to each other. To model this, we could have a regularization term that is similar to fused-lasso. Define a graph G, where two K-mers are adjacent if they differ by a single base pair. Then we have the following problem, minimize:

$$f(\beta) = \sum_i W_i(Y_i - X_i\beta)^2 + \lambda \sum_{(j,k)\in G} (\beta_j - \beta_k)^2 \tag{107}$$

    subject to $\beta_j \geq 0$. The drawback of this approach is: (1) difficult to optimize; (2) the assumption that the neighbors should have similar affinity is only partially true: the critical position could have a large effect on the binding affinity.

- Optimization: in general, the objective function and constraints (under different scenarios) are all quadratic, so efficient algorithm using convex optimization is always available.
  - Positivity constraint: if the only constraint is $\beta_j \geq 0$, in addition to L_1 regularization, then we could use a modification of the original LARS algorithm. This is reported as "positive LASSO" in [Efron et al, Least Angle Regression, 2003].
  - Reference: [Model-based deconvolution of genome-wide DNA binding, Bioinformatics, 2008].

The effect of TF concentration:

- Background: given a binding site S_j, its probability of being occupied is proportional to the Boltzmann weight q_j, defined as:

$$q_j = [TF]K(S_{\max})e^{-E_j} \tag{108}$$

where $[TF]$ is the TF concentration, K is the association constant, $S_{\max}$ is the consensus site, E_j is the mismatch energy of the site (nonnegative). The probability of binding is thus: $P(S_j) = q_j/(1+q_j)$.

- In vivo experiments: the PBM data is obtained in vitro. The coefficients (binding probabilities) β_j are generally different in vivo because of difference in $[TF]$, among other things. Let r be the relative TF concentration:

$$r = \frac{[TF]_{\text{vivo}}}{[TF]_{\text{vitro}}} \tag{109}$$

and let the binding probability in PBM be $\beta^{(0)}$. We could obtain the binding probability in vivo $\beta(r)$ as a function of $\beta^{(0)}$ and r:

  - Boltzmann weight in vitro: β depends on the affinity of the site and its reverse complement. For most TFs, however, only one of the two makes significant contribution, so we assume β_j is equal to the binding probability of the stronger one. The Boltzmann weight of the site in vitro is given by:

$$q^{(0)} = \frac{\beta^{(0)}}{1-\beta^{(0)}} \tag{110}$$

  - Binding probability in vivo: q is proportional to $[TF]$, thus the weight in vivo is given simply by $rq^{(0)}$. The binding probability:

$$\beta(r) = \frac{rq^{(0)}}{1+rq^{(0)}} = \frac{r\beta^{(0)}}{1+(r-1)\beta^{(0)}} \tag{111}$$

This means when applying the model in vivo, we should correct for β using r. However, r is usually unknown, and we need to fit r using the in vivo data (e.g. using the ChIP-seq data).

- Variation of $[TF]$: when applying the model to data where $[TF]$ may change, e.g. time-series TF-binding data, the relative change of $[TF]$ can be obtained from data, and r is no longer unknown.

2. Combining sequence information and DNase hypersensitivity data for TFBS prediction

Treating DNase hypersensitivity (HS) data as a measure of chromatin accessibility:

- Chromatin accessibility: among different types of chromatin data, DNase I hypersensitivity (HS) data is found to be the most informative, so we will only consider this type of data. We assume we have measurements along the genome, $D(x)$, where x is the genome position. And larger values of $D(x)$ means the position is more accessible, and very large values of $D(x)$ corresponds to open chromatin (completely accessible).

- Incorporating accessibility: suppose we have the binding probability β^0 for a given site from PBM experiments. In PBM experiment, DNA is not bound by nucleosomes, so β^0 corresponds to the maximum binding of the site. For a genomic position with accessibility measurement D, its binding probability can be modeled as a logistic regression of β^0:

$$\beta(D) = \beta^0 \frac{1}{1 + e^{-\mu_0 - \mu_1 D}} \tag{112}$$

  The parameters μ_0 and μ_1 need to be fit from data. For a longer sequence, we sum over all putative sites, with the above correction for each site. If D does not vary much along the sequence, we simply multiple the correction constant to the total binding affinity of the sequence.
- Drawbacks of this approach:
  - The parameters μ_0 and μ_1 have to be estimated from additional data. If they are independent of the TF, then we could estimate them from the ChIP-seq data of a small number of TFs, however, this is unlikely to be true. Empirically, the optimal parameters vary with TFs; biologically, TF binding may modify the chromatin accessibility.
  - Not clear how additional information is incorporated, e.g. the distance to TSS, sequence conservation, etc.

Biophysical model of TF occupancy and DNA accessibility: simple model

- Motivation:
  - DNase data (tag counts) are determined by the local chromatin accessibility.
  - Chromatin accessibility depends on the basal level of accessibility at that position, and may be modified by TF binding. It is well known that TF binding may recruit histone modifying enzymes and chromatin remodeling complex to change chromatin structure.
  - The need of modeling both accessibility and TF occupancy: the two variables are related, but may differ significantly. E.g. A region may be completely open (by the action of other TFs), but may not be bound by the TF of interest at all.
- Notations: suppose we have n sites (e.g. a large number of sites randomly sampled from the genome). We have measurements, the tag count D_i of the i-th site (in the nearby 200bp window). Suppose the predicted affinity (probability of occupancy) of the i-th site is β_i, assuming β_i is estimated from in vitro PBM data (thus known). For simplicity, we use γ_i as a measure of TF binding in naked DNA:

$$\beta_i = \frac{\gamma_i}{1 + \gamma_i} \tag{113}$$

  Let Z_i be the (latent) indicator variable of whether TF is bound to the i-th site, and A_i be the indicator variable of whether the chromatin is accessible (to DNase I). Our goal is to estimate $P(Z_i = 1 | D_i, \gamma_i)$, the in vivo occupancy of the TF.
- Simple model: assuming TF binding does not modify chromatin accessibility, A_i is independent of Z_i. For any site i, we have

$$P(A_i = 1) = \frac{q}{1 + q} \tag{114}$$

where q is the average weight (unnormalized probability) of the open state, and indepdent of sites. Given A_i, the distribution of D_i is given by some parameterized distribution, e.g. Poisson or Negative Binomial distribution [CENTIPEDE paper, Pique-Regi, GR, 2011]. For simplicity, we assume:

$$P(D_i|A_i = 1) = \text{Poisson}(\phi_1) \qquad P(D_i|A_i = 0) = \text{Poisson}(\phi_0) \tag{115}$$

Our parameters are $\theta = (q, \phi)$. We thus have the likelihood:

$$\begin{array}{ll} P(D|q,\phi) & = \prod_i \left[P(A_i = 1)P(D_i|A_i = 1) + (1 - P(A_i = 1))P(D_i|A_i = 0)\right] \\ & = \prod_i \left[q/(1+q)\text{Poisson}(D_i|\phi_1) + 1/(1+q)\text{Poisson}(D_i|\phi_0)\right] \end{array} \tag{116}$$

- Inference: The parameters can be estimated using EM algorithm (treating A_i as latent variables). Once we have the parameters, the latent variable A_i can be inferred:

$$P(A_i|D_i, \hat{\theta}) \propto P(A_i|\hat{q})P(D_i|A_i, \hat{\phi}) \tag{117}$$

  Once we have $P(A_i|D_i)$, we could infer the occupancy at each site as (ignoring the parameters in the equation):

$$\begin{array}{ll} P(Z_i = 1|D_i) & = P(Z_i = 1|A_i = 1)P(A_i = 1|D_i) + P(Z_i = 1|A_i = 0)P(A_i = 0|D_i) \\ & = \gamma_i/(1+\gamma_i) \cdot P(A_i = 1|D_i) \end{array} \tag{118}$$

  where we assume: (1) under $A_i = 1$, i.e. for open chromatin the occupancy of TF is equal to the occupancy of TF in naked DNA; (2) TF cannot bind to closed chromatin, i.e. $P(Z_i = 1|A_i = 0) = 0$.

- Remark:
  - This is similar to the model before where we simply multiply the occupancy in vitro by a ratio that depends on the DNase tag counts. The difference is that the parameters can be estimted by MLE (fitting a mixture model).
  - The problem of this simple approach is that DNA accessibility does not depend on TF binding affinity, in other words, the parameters of the model are not TF-specific. In practice, we could check whether TF binding affects accessibility by comparing the distribution of DNase tag count in high-affinity (predicted) regions vs. background regions.

Extending the simple model with TF binding:

- Model: consider the i-th site, it may exist in three states:
  - Closed: $A_i = 0, Z_i = 0$;
  - Open but not bound by the TF: $A_i = 1, Z_i = 0$
  - Open and bound by the TF: $A_i = 1, Z_i = 1$

  The weights of the former two states are: 1 and q respectively (from our previous model). The weight of the last state is $\lambda\gamma_i q$, where $\lambda \geq 1$ represents how strongly the TF may modify the chromatin accessibility. Indeed, if the TF has no effect, $\lambda = 1$, the weight

is simply the product of q (the probability of open state) and γ_i (the probability of TF binding in open chromatin). From this, we have:

$$P(A_i = 1) = \frac{q(1 + \lambda\gamma_i)}{1 + q(1 + \lambda\gamma_i)} \tag{119}$$

The interpretation is: if TF may modify chromatin state, then we simply replace q in the simple model above (Equation 114), by $q(1 + \lambda\gamma_i)$, so that the site with strong TF binding is also likely to be open.

- Iinference: similar as before, the parameters $\theta = (q, \lambda, \phi)$ can be inferred by Maximum Likelihood using EM. Once we have the parameters, and $P(A_i|D_i)$, the occupancy:

$$P(Z_i = 1|A_i, D_i, \hat{\theta}) = P(Z_i = 1|A_i = 1)P(A_i = 1|D_i, \hat{\theta}) = \frac{\hat{\lambda}\gamma_i}{1 + \hat{\lambda}\gamma_i} P(A_i = 1|D_i, \hat{\theta}) \tag{120}$$

When $\lambda = 1$, this equation reduces to the simple case before.

Incorporating additional information:

- Model: the basal level of chromatin accessibility, q, may vary with chromatin region. Ex. suppose d_i represents a measure of the distance of the i-th site to the TSS (we transform the distance s.t. it is close to 0 when the site is far from TSS), we may assume that q is larger if it is closer to TSS. We could then replace q by $q(1 + \eta d_i)$. Taking both TF binding and distance into account:

$$P(A_i = 1) = \frac{q(1 + \lambda\gamma_i + \eta d_i)}{1 + q(1 + \lambda\gamma_i + \eta d_i)} \tag{121}$$

Similarly, we could add terms representing the effect of other TFs that may facilitate chromatin changes, or sequence conservation (conserved sequences are more accessible, i.e. having larger q).

- Comparing with CENTIPEDE model:
  - In the CENTIPEDE model, no distinction is made between A_i and Z_i, it is assumed that D_i depends on Z_i (TF occupancy). The consequence is: given a site with very high tag count, the model will predict that $Z_i = 1$ given D_i, regardless of whether the site actually matches the TF binding specificity. To avoid this problem, CENTIPEDE imposes a hard cutoff on all putative PWM matches.
  - In contrast, in our approach, the TF occupancy is the posterior probability of $A_i = 1$ (similar to CENTIPEDE), multiplied by the constant $\lambda\gamma_i/(1 + \lambda\gamma_i)$, which favors strong sites. Thus, for a site with extremely high DNase tag count, the model will predict $A_i = 1$, but if it does not match the TF, γ_i will be close to 0 and our model will predict low TF occupancy.

Predicting TF binding with integrative model:

- Motivation: an integrative model that predicts TF binding using sequence features and functional data (measurements).

- Model overview: Let X be the sequence features, including TF binding, possible co-factors, distance to TSS, GC content, etc., and Y be functional data, including DNase HS data, histone patterns, gene expression level, and conservation data (indication of functionality of the sequence). Let Z be the latent variable of TF occupancy at the sequence. The model:

$$X \to Z \to Y \tag{122}$$

  The $X \to Z$ model is called the upstream model, and the $Z \to Y$ model is called the downstream model. The upstream model describes how sequence features may determine TF binding; and the downstream model specifies how TF binding modifies the epigenetic patterns of a sequence.

- The upstream model: the binding of a TF to a sequence primarily depends on the physical interaction between the TF and its motif. This is modified by other features. We define X_{TF} as the main feature, e.g. the motif score or the PBM score of the sequence, and X_{other} as other features, e.g. the motif/PBM scores of other TFs, the distance to TSS, the GC content of sequence, etc. Our model:

$$P(Z = 1|X_{\text{TF}}, X\text{other}) = P(Z = 1|X_{\text{TF}}, A)P(A|X_{\text{other}}) = \gamma(X_{\text{TF}}) \cdot \sigma(X_{\text{other}}\beta) \tag{123}$$

  where A describes the event that the chromatin is accessible, and γ is the function that maps X_{TF} to the probability of occupancy, and σ is the logistic regression.

- The downstream model: if there are multiple measurements of Z, we could assume they are independent. In some cases, however, we may want to model the relationship between these variables (e.g. expression level depends on the epigenetic states). Consider two types of data, Y_{HS}, the DNase HS data, and Y_{C}, the conservation data (e.g. PhastCons scores). These can be modeled as:

$$Y_{\text{HS}}|Z \sim \text{Poisson}(\lambda_Z) \tag{124}$$

$$Y_{\text{C}}|Z \sim N(\mu_Z, \sigma_Z^2) \tag{125}$$

  where we assume that the conservation scores are normally distributed (Z scores).

Predicting TF binding using integrative model (II):

- Model: let D be the DNase HS data, C be the conservation data (e.g. PhastCons score) of a sequence to be tested. Define two latent variables, A - whether chromatin is accessible and Z - whether the sequence is bound by the TF of interest. Our model:

$$D \leftarrow A \rightarrow Z \rightarrow C \tag{126}$$

  We assume A prior distribution is Bernoulli(π), and

$$P(D|A=k) = \text{Poisson}(\lambda_k) \qquad k = 0, 1 \tag{127}$$

  The conditional distribution of $Z|A$ is determined by physics of binding:

$$P(Z=1|A=0) = 0 \qquad P(Z=0|A=0) = 1 \tag{128}$$

$$P(Z=1|A=1) = p \qquad P(Z=0|A=1) = 1-p \tag{129}$$

  where p is the probability of binding in vitro. We assume p is known up to a constant c (the constant may need to be estimated). The distribution of C:

$$P(C|Z=k) = N(\mu_k, \sigma_k^2) \qquad k = 0, 1 \tag{130}$$

  assuming the conservation score follows normal distribution.

- Likelihood: the parameters $\theta = (\lambda_1, \lambda_0, \mu_1, \sigma_1^2, \mu_0, \sigma_0^2, c)$, the likelihood:

$$P(D, C|\theta) = \sum_{A,Z} P(A, Z, D, C) = \sum_{A,Z} P(A)P(D|A)P(Z|A)P(C|Z) \tag{131}$$

  Plug-in the various terms, we have the likelihood for one data point:

$$\begin{aligned} P(D_i, C_i|\theta) &= (1-\pi)\text{Poisson}(D_i|\lambda_0)N(C_i|\mu_0, \sigma_0^2) + \pi p_i \cdot \text{Poisson}(D_i|\lambda_1)N(C_i|\mu_1, \sigma_1^2) \\ &+ \pi(1-p_i) \cdot \text{Poisson}(D_i|\lambda_1)N(C_i|\mu_0, \sigma_0^2) \end{aligned} \tag{132}$$

- Remark: the model can be extended to incorporate more data. In particular, we could define a prior on $P(A)$ that depends on the distance to TSS, the presence of other motifs (which may facilitate TF binding).

3. Combining quantitative predictions from multiple species

   Phylogenetic averaging with switching event:

   - Problem: suppose we have quantitative traits (e.g. TF occupancy) in n species, $x_1, \cdots, x_n$, related by a tree T. We are interested in obtaining the average over n species. However, the evolutionary process is not always homogeneous, in particular, the selection pressure on the trait may be applied to only a subset of species (we are only interested in the average under constraint).
   - Averaging over partial tree: suppose we have a reference species x^*, wlos, we can assume that it is the root of the tree (time reversibility). Starting from the reference species, intially it is under selection; however, during evolution, the constraint may be lost (a switch event). Let T^* be the partial tree where selection operates, and $\bar{\mu}(T^*)$ be its tree average (computed from the phylogenetic average algorithm). We need to compute:

     $$\bar{\mu} = \mathrm{E}_{T^*}[\bar{\mu}(T^*)] \tag{133}$$

     We assume the switch event follows a Poisson process with rate λ, where λ is relatively low. When no switch event happens (the probability $1-\lambda T$), the average is $\bar{\mu}(T)$. When switch event does happen, suppose the time of the event is τ, we have:

     $$\bar{\mu}_{\text{switch}} = \sum_{i=1}^{2n-3} \int_0^{t_i} \lambda \bar{\mu}(T_\tau) d\tau \tag{134}$$

     where $\bar{\mu}(T_\tau)$ is the average over the tree induced by τ, T_τ. We approximate the integral by:

     $$\int_0^{t_i} \lambda \bar{\mu}(T_\tau) d\tau \approx \lambda t_i \bar{\mu}(T_{t_i/2}) \tag{135}$$

     i.e. the switch even only occurs in the mid-point of any branch. The total average is given by:

     $$\bar{\mu} = (1-\lambda T)\bar{\mu}(T) + \lambda \sum_{i=1}^{2n-3} t_i \bar{\mu}(T_{t_i/2}) \tag{136}$$

     Thus we need to compute the phylogenetic average over the whole tree, $\mu(T)$, and the phylogenetic average over the subtree, $T(t_i/2)$, induced by the switch event at the mid-point of the i-th branch.
   - Example: we consider a two-branch tree with leaf nodes X_1, X_2 and branch length t_1, t_2. Suppose X_1 is the reference species. According to our equation above:

     $$\bar{\mu} = (1-\lambda T)\frac{X_1+X_2}{2} + \lambda t_1 X_1 + \lambda t_2 X_1 = (1-\lambda T)\frac{X_1+X_2}{2} + \lambda T X_1 \tag{137}$$

     where $T = t_1 + t_2$. The weights of the two species are:

     $$w_1 = \frac{1-\lambda T}{2} + \lambda T \qquad w_2 = \frac{1-\lambda T}{2} \tag{138}$$

     So the weighting would favor X_1, the reference species; but when λ is very small, $w_1 \approx w_2$, this reduces to the case of no switch event.

- Dealing with missing variables: in the calculation of $\bar{\mu}(T_{t_i/2})$, the tree $T_{t_i/2}$ involves the mid-point of the i-th branch, which is not observed. This is the problem of computing phylogenetic average over a tree, where some leaf nodes are not observed. It is easy to see that we only need to remove the branch of the missing leaf node - all the leaf nodes now are observed.
  - Nodes with only one child: if we remove a branch in the tree, its parent node now has only one child. We could modify the dynamic programming algorithm to deal with non-binary tree, or modify the tree, i.e. remove this parent node, so that the tree is binary.
- Tree extraction: the algorithm involves extracting the tree $T_{t_i/2}$ induced by a switch event at the mid-point of the i-th branch. To do this, we first re-root the tree at the reference species, then remove the subtree rooted at the switch point.
- Alternative model: define M_0 as the null model of no natural selection on the trait, and M_1 be the alternative model. The two models differ by the rate of Brownian motion: with σ_0^2 the rate of M_0, and σ_1^2 the rate of M_1. In addition, we may have branch-specific change under M_1. The model is selected by $P(x_1, \cdots, x_n|M_0)$ vs. $P(x_1, \cdots, x_n|M_1)$.
  - Limitations: need to estimate the different Brownian motion rates σ_0^2 and σ_1^2. Need additional assumptions/data, e.g. use many orthologous random sequences to estimate σ_0^2. The advantage of the phylogenetic average method is: not dependent on the σ parameter, and is robust to the normality assumption.
  - Reference: [Gu, Statistical framework for phylogenomic analysis of gene family expression profiles. Genetics, 2004]

Functional conservation of TFBSs in multiple species:

- Motivation:
  - Compensatory change of TFBSs: TFBSs may turnover in a nearby position. From the liver ChIP-seq study [Schmidt & Odom, Science, 2010], a significant fraction turnover between human and mouse occurs within 1-2kb. The challenge is within this range, it is easy to see a motif match by chance.
  - Sequence conservation: while we rely on conservation of affinity, sequence conservation provides additional information: e.g. if a putative site appears in a conserved region, then even if motif match is not found in the orthologous sequences, the site is likely to be real (the motif match may be moved to an adjacent region).
- Strategy: given a long sequence of 1-2Kb in the reference species, our strategy is to estimate the affinity of this sequence in every of K species, and use phylogenetic averaging to evaluate the extent the affinity is conserved. We will estimate the true affinity (not just sequence match), taking epigenetic information into account. When DNase HS data is available in any species, we will use it to estimate chromatin accessibility, which can be then used for affinity prediction; if no HS data, use sequence-level conservation to infer chromatin state.
- Model: suppose A_k is the chromain state of the k-th species (assume that the state is constant in a 1-2 KB sequence window). The occupancy of a site, i, in the k-th species

is related to A_k by:

$$P(Z_k^i|D) = P(Z_k^i|A_k)P(A_k|D) = \frac{\lambda\gamma_k^i}{1+\lambda\gamma_k^i}P(A_k|D) \tag{139}$$

where D is data, and λ is the TF effect on accessibility (assuming constant over species) and γ_k^i is the (predicted) in vitro binding of the i-th site in the k-th species. Our data is D_1, the DNase HS data of the reference species. To obtain $P(A_k|D)$, we have:

$$P(A_k|D_1) = P(A_k|A_1 = 1)P(A_1 = 1|D_1) + P(A_k|A_1 = 0)P(A_1 = 0|D_1) \tag{140}$$

The term $P(A_1|D_1)$ is obtained from the DNase I data in the reference species. The probability $P(A_k = 1|A_1 = 1)$ can be estimated from sequence-level conservation: if sequence is conserved, then we have a high probability that the chromatin state is also conserved. The probability $P(A_k|A_1 = 0)$ can be assummed to be the genome-wide average of $A_k = 1$.

4. Finding gene regulators from expression data

   Problem and background:

   - Problem: given the putative relevance of regulators, generally TFs, to genes (motif score, ChIP-chip/seq data, etc.), and given the expression data of the genes in a certain context, e.g. development, or across multiple tissues, identify the active or true regulators of each gene.
   - Motivation:
     - Motif-based scores may lead to a high fraction of false positives; and even in vitro experiments (e.g. PBM) can not predict well the in vivo binding.
     - ChIP-chip or ChIP-seq data are often obtained in cell lines or conditions/cell types different from the expression data, thus do not directly suggest the biologically active regulators.
   - A special case: expression data can be represented as binary class of genes.
     - Application: similarly expressed genes vs. the other genes, find the relevant regulators.
     - Application: genes are up- or down-regulated, find the regulators responsible for stimulation or inhibition. Ex. TF finding in DREM; TF/motif finding in genes differentially expressed in cancer cells vs. normal cells.

   Strategies:

   - Basic strategy: treat the TF binding or motif scores as features, and expression data or class as response variables, a regression problem.
   - Features: a TF-binding feature could be the summary statistic incorporating homotypic clustering, conservation and other features (e.g. histone modification).
   - Combinatorial TF interactions: a single match to TF may be false positive, but putative binding sites of two related TFs in neighborhood are much more likely to be true positives. Model these as additional features, potentially favoring adjacent sites.

- Gene relationship: genes form groups/pathways and are related to each other (e.g. PPI). The genes in the same pathway or closely related are more likely to share the same regulators.
- Class structure: suppose genes are grouped into clusters and the response variables are cluster membership, then there is additional structure in the responsible variables, e.g. two closely related clusters are likely to share the same regulator. Thus pooling multiple clusters could increase the power of detecting regulators (an idea used in DREM).
- Remark:
  - The main challenge is to model gene structure and class structure.
  - Combinatorial features: the promising features can be learned without using the label data. The simplest strategy is to do a co-occurrence test of pairwise TFs. This can be improved by: (1) gene structure (search for combinatorial features in related genes); (2) subnetworks instead of pairs, e.g. $A-B$, $B-C$ and $C-A$ all have weak statistics, but the triplet may be a good feature.

Incorporating gene groups with single response variable:

- Motivation: There is probably substantial heterogeneity in the gene regulatory network, e.g. two genes may have similar expression profiles, but their regulators may be quite different. Incorporating gene groups could increase the power:
  - Example: a TF may control a small set of genes in a large cluster, if we do an enrichment test of this TF, it is not significant; but if this small number of genes fall in the same pathway, then this is highly unlikely due to chance.
  - Alternative approach: for each pathway, separately test if the regulator is enriched. However, this loses power, as a regulator may act in multiple pathways; and in particular, the weak effects in multiple pathways could be strong evidence.
- Model: given the features $X_1, \cdots, X_p$ of n genes, and class variables $Y_i, 1 \leq i \leq n$. In the simplest model, we have:
$$Y = X\beta + \epsilon \tag{141}$$
(Replace with logistic regression for binary classes). Suppose genes can be divided into K disjoint pathways, then to model heterogeneity, we may assume that each pathway has its own $\beta_k, 1 \leq k \leq K$ (i.e. regulators are pathway specific). To reduce the model complxity, we need to model how these β_k's are related. We could also assume that these pathways may be related in some way (associated with a graph).
- Interactions: introduce pathway variables of genes, Z, thus $Z_i = 1$ if a gene belongs to the pathway (suppose we study one pathway a time). Then we have the regression to model the dependence of β on pathways (consider only one feature):
$$Y = \beta X + \gamma X \cdot Z + \epsilon \tag{142}$$
where $X \cdot Z$ is the interaction term, and the null hypothesis that the regulator is not relevant is expressed as: $\beta = \gamma = 0$.
- Hierarchical model: we could assume that β_k is from a common distribution, e.g. $N(\lambda, \tau^2)$, and the null hypothesis that the regulator is irrelevant corresponds to $\lambda = 0$,

which can be inferred from the posterior distribution of λ. A yet better approach to model heterogeneity is to use a mixture distribution: $\beta_k \sim \text{Mixture}(0, N(\lambda, \tau^2))$, thus the regulator has no effect in most pathways, but may have non-zero effect on a small number of pathways.

- Lasso: we could use the idea of fused Lasso that penalizes the difference of β_k's between two groups: this has the effect of regularizing β_k's. In particular, the penalty can be weighted by using the pathway relationship: the more related pathways are penalized more heavily if their β_k's are different.
- Kernel smoothing: this is the idea of varying coefficient model, applied to the discrete index variable k. We'll learn a smooth function β_k over k, thus related pathways tend to have similar values of β_k.
- Comparison of different methods:
  - Regulator-pathway interaction: The drawbacks are (1) there are many pathways to test (multiple testing correction); (2) no simple way to incorporate pathway relationship; (3) one pathway is tested at each time, thus losing the power.
  - Hierarchical model: modeling pathway relationship is difficult; computationally expensive.
  - Lasso: not clear how to do hypothesis testing.
  - Kernel smoothing: not clear how the kernel should be defined.
- Remark:
  - Many genes are not assigned to any known pathway, thus the power is reduced if we use only known pathways. Possibly strategies: (1) use gene modules derived from functional gene networks; (2) assign any such genes to known pathways using gene network data.
  - Overlapping pathways can be incoporated by: if a gene belongs to multiple pathways, then β for this gene is a mean of β_k of all related pathways.

Incorporating gene networks with single class variable:

- Model: similar to above, but now we have a graph G that relates genes, and we want related genes to share regulators. Suppose β_{ij} is the effect of the j-th feature on the i-th gene, we need to regularize on β_{ij}.
- Hierarchical model with latent variable: suppose we have a latent variable Z_{ij}, it is 1 if i-th gene is regulated by the j-th feature. We could define a prior distribution of Z_{ij} using Markov random field: if two genes are adjacent, then it is more likely their Z_{ij} terms share the same sign. Then for the i-th gene: if $Z_{ij} = 0$, the j-th feature is irrelevant; if $Z_{ij} = 1$, the j-th feature has effect β_j.
- Lasso: the idea of fused lasso can be applied easily here: we learn parameters β_{ij} with penalty for difference between parameter values of adjacent genes.
- Kernel smoothing: similar idea as above, we could define the objective function s.t. β_{ij} are smoothed.