Whole model (for thesis) overview

Brendan Law

August 2, 2019

I finally figured it all out, at a high-level, and how it fits together! Then I jumped for joy, and took a long break and played Zelda. Here's a type-up of a brief summary of how everything i've done fits in, including:

1. Model in Servin and Stephens, key derivations and results
2. Shim and Stephens – key results and findings
3. HMT extension onto it
4. multiseq – key results and findings
5. How to do the HMT extension onto it

1 Basic goals and procedures

1.1 Goals

We are going to build a model describing the relationship between a dependent variable and a covariate of interest (may be continuous, but the focus will be on discrete covariates, such as membership in a group or class of people). The dependent variable may be a quantity which proxies some quantity of interest (eg: chromatin accessibility, other counts from other sequencing techniques which relate to some cellular-level trait).
Want to be able to do two things from the model we build:

1. **Inference** – is there sufficient evidence to reject the null hypothesis of no association between the covariate and the dependent variable? ($H_0 : \boldsymbol{\beta} = 0 vs H_1$).
   - As the γ parameter, and the respective hyperparameter, π, determines whether β is non-zero or not, we have two quantities of interest:
   - Bayes Factor: $BF_{s,l}(y, g) = \frac{P(y_{s,l}|g,\gamma_{s,l}=1)}{P(y_{s,l}|g,\gamma_{s,l}=0)}$
     - Does $\pi_s = 0 \Leftrightarrow \boldsymbol{\pi} \equiv 0$?
   - Likelihood ratio for $\boldsymbol{\pi}$ across all scales and locations: $\Lambda(\pi; y, g) = \frac{P(\boldsymbol{Y}|\boldsymbol{g},\boldsymbol{\pi})}{P(\boldsymbol{Y}|\boldsymbol{g},\boldsymbol{\pi}\equiv 0)}$
   - With $\boldsymbol{\pi}$ parameterised by an empirical bayes approach (and using an expectation maximisation algorithm)
   - Is BF used only for calculations? Does it have no use here in the actual inference, as the likelihood ratio takes care of that?
   - Is BF used only for calculations? Does it have no use here in the actual inference, as the likelihood ratio takes care of that?
2. **Estimate** – What is the estimated size of any association effect between the covariate and dependent variable at any given base location along the genome? A few quantities of interest here:
   - Posterior distribution of $\beta_{s,l}$
   - Mean and Variance of $\beta_{s,l}$
   - Ability to translate estimates of β (wavelet space) to α's in the data space, for more interpretable estimates.
3. May be used to model either associations or differences (models are applicable to both, method used will depend on size and nature of dataset, how noisy it is, etc.)

1.2 Data

We have the following data:

- High-throughput sequencing data (for n individuals, at a base-pair level, with counts or proxies of some desired behaviour we want to model, for each base $b = 1, \ldots, J$, where J is a power of 2)
- Some covariate for each individual, n, such as whether they are in group A or group B of the experiment we are using to detect effects in. Perhaps it's something else, like a continuous (non-categorical) variable.
- The data is pretty noisy (see all of HJ's slides for examples of how noisy such data is along the genome).
- We are treating *this data is a noisy measurement of an underlying "function" (which is much smoother).*
- Hence, the underlying strategy is to infer association, and perform estimation of effect size against a *smoothed, underlying function.* To that end, we could either:
  - Smooth the signal at an individual level, **THEN** model the differences between smoothed functions, OR
  - Use the noisy data at an individual level, **THEN** smooth the modelled differences between the functions. **This is the approach we adopt.** WHY? What would happen if we denoised, then modelled, instead? Would we lose too much signal regarding the (volatile) spatial structure of the data?

1.3 Key steps and procedures, and why

There are a few fairly important steps and components in our model:

- **Multiscale transform through wavelets (and later, using properties of independent Poissons)**:
  - "Whitening" property of wavelet transform, such that the transformed wavelet coefficients (WCs) can be treated as independent of each other
  - Expectation of spatial structure in data space $\Rightarrow$ sparse structure in transformed (multiscale) space, with local spatial effects seen as concentrations of non-zero coefficients
  - Sparsity easier to model (distributions centred around/'based on' 0)
  - 1:1 data $\leftrightarrow$ wavelet space transform $\Rightarrow$ modelled effects in multiscale space can be translated back into data space
- **Denoising the estimated coefficient through shrinkage/spike and slab priors**
  - A lot of the WCs will be close to (but not exactly, due to noise) 0, if no signal is present around that area.
  - Shrink those small values to 0
  - "Smooth" out/denoise estimates
  - Here, β is shrunk by imposing a hierarchical structure on β, more specifically, a 'spike-and-slab' prior (either normally distributed around 0, or 0 w.p. 1)
  - Effect size estimates (in the wavelet space) are smoother, denoised (as mentioned above, this is as opposed to denoising the data or WCs themselves)
  - Is it true to say that noisy WCs $\Rightarrow$ noisy β's, and then we shrink the β's to smooth them right at the end as they're our primary quantity of interest?
- **Model response-covariate relationship through hierarchical linear model**

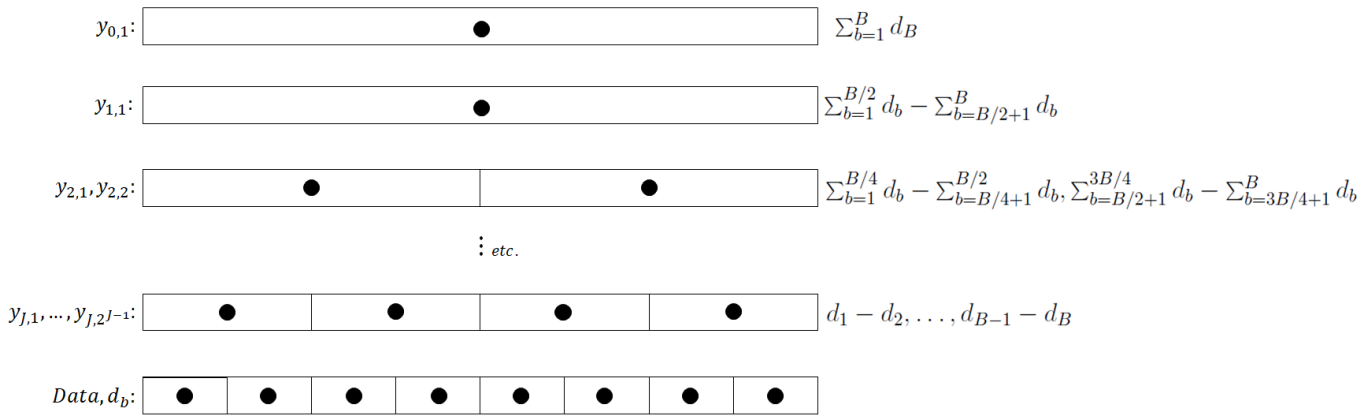
$$
\begin{aligned}
y^i_{s,l} &= \mu_{s,l} + \beta_{s,l} g^i + \epsilon^i_{s,l} \\
\epsilon^i_{s,l} &\sim N(0, \sigma^2_{s,l}) \\
\beta_{s,l} &\sim N(0, \sigma^2_{(s,l),\beta}) \gamma_{s,l} + \delta_0 (1 - \gamma_{s,l}) \\
\gamma_{s,l} &\sim Bern(\pi_s)
\end{aligned}
$$

  - Refer to 'Shim_Stephens_HMT_Beta_Properties.pdf' for the equations in matrix form (and their dimensions, etc)
- **A whole heap of Bayesian work behind the scenes – obtaining closed form formulations**
  - We'll get to this in section 2

1.4 Wavelets – what we need to know

We only discuss the Haar Discrete Wavelet Transform (DWT) here, as it is the method used in the original paper. Plus, this is only a very small/short summary.

- If no signal, most WCs should be close to 0. If signal, area of concentrated non-zero.
- This diagram briefly explains how the Haar DWT works. It's the difference of sums of a region, with the sums covering a larger region as we get less granular.
- There is also a WC at the top which is the sum of all the data.
- This ensures the transform is 1:1 (captures all the information from the data - all differences are captured at different levels, as well as the final sum).
- Finest scale has half the number of coefficients as the data, as it is generated by differencing adjacent pairs of the data.
- Need to figure out what WCs look like if a function is smooth vs if it is noisy.



2 Servin and Stephens

Servin and Stephens' paper traced out the origins of the closed-form, hierarchical Bayesian Linear Regression that forms the basis of both the subsequent papers – it's vital that we understand its key components. Here's a quick summary:

- Data: More or less the same as described, before
- Parameters:
  - Precision, $\tau \sim \Gamma(\frac{\kappa}{2}, \frac{\lambda}{2})$, with $\lambda, \kappa \to 0$ later so that the posterior has the desired behaviour and is more tractable to use, whilst still being proper
  - Mean, $\mu \mid \tau \sim N(0, \frac{\sigma_\mu^2}{\tau})$, with $\sigma_\mu^2 \to \infty$, but large values used in practice
  - Effect size, $\beta \mid \tau \sim N(0, \frac{\sigma_\beta^2}{\tau})$
  - Distribution of $\mu, \beta \mid \tau$ can be represented as the bivariate normal of an uncorrelated random vector with their respective variances on the diagonals.
  - The model is of form $y_i = \mu + \sum_j x_{ij}\beta_j + \epsilon_i$
- Hyperparameters: $\boldsymbol{\theta} := (\kappa, \lambda, \sigma_\mu^2, \sigma_\beta^2)$
- Parameters: $\mu, \beta_1, \ldots, \beta_j$
- Likelihood: $y_i \mid \mu, \beta_j, \tau, \boldsymbol{\theta}, x \sim N(\mu + \sum_j x_{ij}\beta_j, \frac{1}{\tau})$
- Note that despite finding the posteriors and Bayes Factors in the limit, the hyper-parameter σ_β^2 was not taken in the limit.

There were two main outcomes from this paper:

1. Find the Bayes Factor that (in this example, the single) SNP, s is a QTN, ie. the evidence supporting that SNP s is a QTN. This is denoted as $BF(s)$, and defined as the probability of data (under the model with single SNP, s) given the covariates (after all parameters integrated out), divided by the "null" model, which only has one parameter – the intercept (and does not depend on the value of the covariates):

$$BF(s) = \frac{P_s(y \mid G)}{P_0(y)}$$

   - Requires finding the relevant likelihood quantities (data given parameters)

2. Finding the distribution for β for inference – involves finding the posterior distribution for β and integrating out all the parameters.

In brief, the following was done, (most of this derived from the supplementary material)

- Found $P(\mu, \beta, \tau \mid y, \boldsymbol{\theta}, x))$ by finding joint prior, and combining with likelihood and using Bayes' Formula (Posterior $\propto$ Likelihood $\times$ Prior)
- Found joint beta-tau posterior $P(\beta, \tau \mid y, \boldsymbol{\theta}, x))$ by integrating over μ
- Found marginal tau posterior $P(\tau \mid y, \boldsymbol{\theta}, x))$ by integrating over β
- Found marginal beta dstribution (still with tau) $P(\beta \mid \tau, y, \boldsymbol{\theta}, x))$ by dividing the joint posterior by the marginal tau posterior
- We can also get $P(\beta \mid \tau, y, \boldsymbol{\theta}, x)$, which, in conjunction with with $P(y, \beta \mid \tau, \boldsymbol{\theta}, x)$, can get us $P(y \mid \tau, \boldsymbol{\theta}, x)$. This, with the prior of tau gives us $P(y \mid x, \boldsymbol{\theta})$, which is what we need for the Bayes Factor.
- This also brings us very close to being able to find the posterior of beta, $(\beta \mid \boldsymbol{\theta}, x)$, but we'd need to integrate out τ, which is exactly what we do later, in Shim and Stephens.

3 Shim and Stephens - no HMT

Now, we can extend what was done in Servin and Stephens, and adapt it to Shim and Stephens. The model is similar, and follows similar ideas, but now incorporates shrinkage of β, and, for now, is not generalised to a sum of multiple betas (and covariates). All other notation and structure is similar, but there are some slight differences:

- Parameters:
  - Variance (not precision, but now follows an inverse gamma as a result), $\sigma^2_{sl} \sim \Gamma^{-1}(\kappa^a_{sl}, \kappa^b_{sl})$
  - Mean, $\mu_{sl} \mid \sigma^2_{sl} \sim N(0, \sigma^2_{\mu,sl}\sigma^2_{sl})$
  - Effect size, $\beta_{sl} \mid \gamma_{sl}, \sigma^2_{sl} \sim \gamma_{sl} N(0, \sigma^2_{\beta,sl}\sigma^2_{sl}) + (1 - \gamma_{sl})\delta_0$, with δ_0 a point mass at 0
  - Proportion of zero WCs at each scale, $\gamma_{sl} \sim Bernoulli(\pi_s)$
- Hyperparameters: $\boldsymbol{\theta} := (\kappa^a_{sl}, \kappa^b_{sl}, \sigma^2_{\mu,sl}, \sigma^2_{\beta,sl}, \boldsymbol{\pi})$. Bayes Factor is found in the limits, $\kappa^a_{sl}, \kappa^b_{sl} \to 0$ and $\sigma^2_{\mu,sl} \to \infty$. Again, what about $\sigma^2_{\beta,sl}$? Why is it not taken to the limit?. In practice, $\sigma^2_{\beta,sl}$ takes on a discrete uniform prior across a range of values (0.05, 0.1, 0.2, 0.4) to span a 'wide' range of effect sizes. Is it because this is a quantity of interest, we test varying values as it is sensitive? Or would taking this to the limit not provide the nice invariance properties, etc, which we are after?
- Likelihood:
- Model is of form $y^i_{sl} \sim N(\mu + \beta_{sl} g^i, \sigma^2_{sl})$

Two main goals:

1. Posterior for β, so can derive mean, variance properties of the distribution
2. Bayes Factor, $BF_{sl}(y, g) = \frac{P(y_{sl} \mid g, \gamma_{sl}=1, \boldsymbol{\theta})}{P(y_{sl} \mid g, \gamma_{sl}=0, \boldsymbol{\theta})}$

- Note that this BF is **equivalent** to the version derived in Servin and Stephens. The 'null' model there was also just the intercept (which it also is here), and their closed form was for one covariate and effect coefficient, which ours is also.
- The prior on beta doesn't effect the formulation, as, if $\gamma = 1$, then the β prior is the same anyway. So I guess another important thing to note is that if we *condition on* $\gamma = 1$, then the formulation of $y \mid g, \boldsymbol{\theta}$ is **equivalent** also.

Hence, the main takeaway from here (and the supplementary material) is how to calculate the posterior for β.

- $P(\beta_{sl} \mid y_{sl}, g^i, \boldsymbol{\theta}) = P(\beta_{sl} \mid \gamma_{sl} = 1, y_{sl}, g^i)P(\gamma_{sl} = 1 \mid, y_{sl}, g^i) + P(\beta_{sl} \mid \gamma_{sl} = 0, y_{sl}, g^i)P(\gamma_{sl} = 0 \mid, y_{sl}, g^i)$
- Consider the case where $\gamma_{sl} = 1$. The $= 0$ case is a lot more straightforward with little work.
- Joint posterior(-ish) (still including σ^2_{sl}) of (μ_{sl}, β_{sl}) can be shown to be a multivariate normal, uncorrelated. So marginalising over μ gives us univariate normal, which is neat, and a nice example of why distributions that factorise into their products are so much better to deal with (will be important later)
- Get $P(\beta_{sl} \mid \gamma_{sl} = 1, y_{sl}, g^i)$ by multiplying $\beta_{sl} \mid \gamma_{sl} = 1, y_{sl}, g^i, \sigma^2_{sl}, \boldsymbol{\theta}$ and $\sigma^2_{sl} \mid \gamma_{sl} = 1, y_{sl}, g^i, \boldsymbol{\theta}$ together, then integrate out σ^2_{sl}. Resulting distribution is 3-parameter t-distribution.
- So then, we have the $= 1$ components for the beta posterior. The $= 0$ component is just a point mass at 0.
- Use Bayes' Formula to find $P(\gamma_{sl} = m \mid y_{sl}, g^i)$ (denoted as phi_{sl} in the paper), in the no-HMT case.
- Maximum likelihood estimations of $\boldsymbol{\pi}$ are found using an EM algorithm, denoting an empirical bayes approach:
  - Find the $\boldsymbol{\pi}$ which maximises the log-likelihood of the π, given the data (ie $\mathcal{L}(\pi; y, g)$). But hard to work into analytical form.
  - Easier to use the complete log-likelihood, with the 'latent' state available here, γ_{sl}. Hence, we find the $\boldsymbol{\pi}$ which maximises the complete log-likelihood, $P(y, \gamma \mid g, \pi)$. Easier to solve.
  - This would involve us using a lot of expressions involving $P(y \mid \gamma, g, \pi)$, but we only have closed forms for the BF.
  - Luckily, this is just the distribution of y, scaled by a constant (probability of y, where gamma = 0), and so dividing everything through by $P(y \mid \gamma = 0, \dots)$ allows us to utilise the closed form of the BF in this computation instead, with an equivalent result.

4 Shim and Stephens - with HMT

4.1 What was wrong with no HMT?

- Model 'assumes conditional independence of WCs (and effect sizes, beta) given π across scales and locations'
- 'Methods that exploit dependencies between WCs should perform better'
- 'One way...exploit tree structure of WCs (and betas) in Crouse et al'

What do we man by conditional independence of y_{sl} and β_{sl} across scales and locations?

- Given we know π (which is the only hyperparameter which links observations across different scales through γ), we have otherwise independent (conditionally) y's.
- If we didn't know π, γ remains random, and there will be dependencies across y. Our knowledge of γ would influence our knowledge of probabilities of y's across scales, locations.
- γ holds all the information about dependencies. (The π's use information from across all scales and locations). Once known, there is no more information in the remaining terms which could help us change our probability of y_i given y_j.
- Again, we have conditionally independent likelihoods, and independent priors $\Rightarrow$ independent posteriors. Likelihood is iid normal, prior are iid normals too:

$$P(\beta_{sl} \mid \dots) \propto P(y_{sl} \mid \dots) \times P(\beta_{sl})$$

So the solution is motivated by the tree-based relationship between WCs – that WCs at one scale and location often have some relationship to those in a similar location proximity, but at higher scales. **Therefore, we can model these dependencies between WCs by placing tree-structured priors on the β parameter, rather than just assuming the independent priors placed upon them at the moment.**

It'd be a good idea here for me to address some alternatives for my understanding of why we went for HMT.

- Would there have been another way to capture dependencies between β at difference scales and locations through having a π which is more 'tied together', rather than just being constant across s? Or does this not really address the issue that the y remains iid across scales and locations as a consequence of the priors placed upon it? (And therefore, the β remains (conditionally) independent across scales and locations too?
- I'm guessing that any other way of doing this would need either another parameter/more complexity incorporated into the π, or alternatively (as below), re-specification of more dependent priors.

4.2 Crouse - HMT paper

Previously, WCs (this paper specifically addresses the cases of signal denoising) have been predominantly modelled as either:

- Jointly Gaussian (which has flaws – its distribution is inconsistent with the heavy tails of WCs where signal is present and sparsity of WCs/high proportion of 0s, where no signal is present), but allows for modelling of dependency
- Non-Gaussian, but independent – which is also flawed in that it is unrealistic

An alternative would be to specify the complete joint distribution of wavelet coefficients, $f(\boldsymbol{W})$, but that would be very difficult to estimate and parameterise.

A HMT structure has some compelling properties for implementing such a tree-based structure on the priors:

- Specify latent state and conditional WC independence based on state $f(\boldsymbol{W} \mid \boldsymbol{S})$, and $f_W(\boldsymbol{w}) = \sum_m f_{(W|S)}(w \mid s = m)P(S = m)$
- Captures dependency and easier to model when we impose structure on the latent states, rather than directly on the WCs. A more 'passive' method of modelling structure.
- We also have some stuff characterising/supporting use of HMMs in such a denoising scenario from Rabiner (1989):
  - HMM belongs to a 'set of statistical models in which one tries to characterise only the statistical properties of the signal...the underlying assumption of the statistical model is that the signal can be well characterised as a parametric random process, and that the parameters of the stochastic process can be determined (estimated) in a precise, well-defined manner.'
  - '...the resulting model (which is called a hidden Markov model) is a double embedded stochastic process with an underlying stochastic process that is *not* observable (it is hidden), but can only be observed through another set of stochastic processes that produce the sequence of observations.'
  - It was shown in the paper to be practically implementable and extensible to cover a wide range of scenarios and different assumptions

So then, we are imposing the mixture model HMT structure on:

- The prior distribution of the $\beta_{s,l}$'s, with $\gamma_{s,l}$ being the latent state driving its distribution:

$$\begin{aligned} \gamma_{s,l} = 1 &\Rightarrow \ \beta_{s,l} \sim N(0, \sigma^2_{(s,l),\beta}\sigma^2_{s,l}), \text{ or} \\ \gamma_{s,l} = 0 &\Rightarrow \ \beta_{s,l} \equiv \delta_0 \end{aligned}$$

- But also noting now that $\gamma_{s,l}$ is NO LONGER just a Bernoulli distribution driven by a uniform parameter across scales, but that we incorporate 'trees' of dependencies by having that, as part of the HMT:

$$P(\gamma_{s,l} \mid \boldsymbol{\theta}) = \sum_m P(\gamma_{s,l} \mid \gamma_{p(s,l)} = m, \boldsymbol{\theta})P(\gamma_{p(s,l)} = m \mid \boldsymbol{\theta})$$

where $\boldsymbol{\theta}$ represent a set of hyperparameters which parameterise the HMT relationships between each scale, location and their respective parents in the tree. This now becomes a part of the complex prior relationship behind the $\beta s, l$ prior.

4.2.1 The EM algorithm behind this

Once again, we want to use an Empirical Bayes approach (why? is the only alternative to place priors?) to parameterise the HMT, however we can no longer rely on independence assumptions of the likelihood as we did before. We now have dependencies, and need to treat the whole joint distribution represented by the likelihood, we can no longer just factorise the prior of γ into a product of independent distributions:

$$\begin{aligned} P(\boldsymbol{y}, \boldsymbol{\gamma} \mid \boldsymbol{g}, \boldsymbol{\pi}) &= P(\boldsymbol{y} \mid \boldsymbol{\gamma}, \boldsymbol{g}, \boldsymbol{\pi}) P(\boldsymbol{\gamma} \mid \boldsymbol{g}, \boldsymbol{\pi}) \\ &= \prod_{s,l} P(y_{s,l} \mid \gamma_{s,l}, \boldsymbol{g}, \boldsymbol{\pi}) P(\gamma_{s,l} \mid \boldsymbol{g}, \boldsymbol{\pi}) \end{aligned}$$

but we must treat it as a joint distribution, where the vector of γ's have a markovian relationship (dependence on parent state):

$$\begin{aligned} P(\boldsymbol{y}, \boldsymbol{\gamma} \mid \boldsymbol{g}, \boldsymbol{\pi}) &= P(\boldsymbol{y} \mid \boldsymbol{\gamma}, \boldsymbol{g}, \boldsymbol{\pi}) P(\boldsymbol{\gamma} \mid \boldsymbol{g}, \boldsymbol{\pi}) \\ &= P(\boldsymbol{\gamma} \mid \boldsymbol{g}, \boldsymbol{\pi}) \prod_{s,l} P(y_{s,l} \mid \gamma_{s,l}, \boldsymbol{g}, \boldsymbol{\pi}) \end{aligned}$$

Once again, from this algorithm, we obtain likelihood maximising parameters for this model, which account for all the structure and data implicit in the model (as the BFs contain all knowledge of priors, likelihoods and parameters, just integrated out into a convenient format). The algorithm involved (backward-forward) is *dynamic* in a sense:

- Only need to iterate/make a guess for the root probability $P(\gamma_{1,1} = m \mid \boldsymbol{\theta})$, then all the conditional probabilities flow from there.
- Along the way, we also obtain the following values:
  - $P(\gamma_{s,l} = n \mid \gamma_{p(s,l)} = m, \boldsymbol{\theta})$ for all others
  - $P(\gamma_{s,l} = m \mid \boldsymbol{y}, \boldsymbol{\theta})$
  - $P(\gamma_{s,l} = n, \gamma_{p(s,l)} = m \mid \boldsymbol{y}, \boldsymbol{\theta})$
- **Now we have β priors dependent on the $\gamma_{s,l}$, thus there are dependencies on the WCs also (through the**
- **Note that $\mu_{s,l}$ is likely to be noisy** as it is a component of (noisy) $y_{s,l}$, and is NOT 'denoised' through shrinkage priors.

4.2.2 Differences with HMT and closing remarks

- β *prior* changes, as γ now has a HMT structure, rather than simple Bernoulli distribution
- All other parameters and likelihoods remain the same
- *Bayes Factor* doesn't change as it is conditional on the values of γ, whereby we still have the same model structure for y, and thus H_0 and H_1 are still the same.
- *Posterior for* β changes slightly as it depends on $P(\gamma_{s,l} = n \mid y_{s,l}, g^i, \boldsymbol{\theta})$, which IS more complex this time around, and must be obtained from the EM algorithm, due to the dependencies on other states and WCs at other scales and locations, which are captured in the $\boldsymbol{\theta}$ parameters.
- In *both cases*, we are required to *transform statistical quantities related to* β back to the data space using the IDWT. Underlying quantities are available in closed form in the no-HMT, but will be simulated where HMT is used.

5 multiseq - no HMT

5.1 Why Poisson?

- Gaussian version is based (not mentioned here, but when implemented practically), on *quantile transformed data*
  - The transform is to 'guard' against potential departures from normality in distribution of the dependent variables, post-wavelet transform (eg: caused by large tails and errors in the WCs)
  - **Only for inference** – ie the model built for inference is built on rank-quantiled WCs to test the hypothesis of association
  - **For estimation**, we model the raw WCs (not transformed) so that the β are more interpretable, and can be transformed easily back into the data space using the IDWT for interpretation.
- There are **two main issues with this:**
  1. *Poor performance on small samples* (Is this because ranking provides little information when few samples given?)
  2. *Information loss* – Because our d_b are based on **proportions** (normalised by total read counts), quantiling means we lose information regarding whether proportions are due to low counts and sample variability (Is an example of this distinguishing 1/2 from 100/200?)
- Model counts directly - no normalising and standardising. Modelling counts also means no need to normalise or standardise to 'fit' to normal distribution assumptions.

5.2 Basics of a Poisson method

- Assume counts along genome follows an inhomogeneous Poisson process with an *underlying intensity function*, $\lambda(b)$, as it varies with location/base pair location, b.
- According to wikipedia, citation 24: 'In all settings, the Poisson point process has the property that each point is stochastically independent to all the other points in the process, which is why it is sometimes called a purely or completely random process.' (`https://en.wikipedia.org/wiki/Poisson_point_process`). Do we have this? Furthermore:
  - Independent increments (for all increments) and/or counts in different subregions are all independent random variables.
  - Given that we assume some underlying spatial structure in the counts, can we still assume this independence between Y_b as different b's?
  - Or can we assume independence between counts at different locations, provided the 'spatial structure' is captured by the underlying intensity function?
- *Goal:* Test for differences in the **underlying intensity function, NOT in the counts**.
  - Counts are just a 'vehicle' by which to motivate the (smoother and latent) intensity function. We relate differences between groups to *differences in the intensity function*, rather than *differences in the counts*.
  - Is it easier to model the underlying function rather than the counts?
  - Is a Poisson process the best/easiest/only way to model such count data?

So, a summary of the differences between the Wavelet and multiseq models are as follows:

	W/let	Multiseq
Data	Normalised count data	Raw count data and modelled underlying intensity function
Transform	DWT (Haar)	Reparameterisation to $(\boldsymbol{x}, \boldsymbol{p})$ (see below) as per Kolaczyk
Transform details	Differences of adjacent sums of data	Differences of adjacent sums of underlying intensities
Model structure	Model WC's mean response, normal dist'n	Lower scale counts (conditional on higher scale's counts), Binomial dist'n (hence logistic glm on 'splitting param', p)

Underlying model structure:

$$\log(\lambda_j^i) = \mu_j^{(d)} + \beta_j^{(d)} g^i + u_j^i$$

where (d) subscript represents 'data space' coefficients, i the ith individual $i = 1, \dots, n$, and j the jth base location $j = 1, \dots, B$, and g is a covariate of interest. To reiterate, two main goals:

1. **Estimation:** $P(\beta_j^{(d)} \mid \dots)$, an interpretable estimate of effect of grp on underlying intensity, λ_j at given locations
2. **Inference:** Bayes Factors to test for evidence supporting alternative hypotheses regarding significance of association between covariates and counts. Can be done in the multiscale space (as previously in wavelets).

5.3 Multiscale transform in the Poisson model

Data Space: $\{(X_1^i, \dots, X_B^i), (\lambda_1^i, \dots, \lambda_B^i)\}$. Note that we'll now drop the subscript 'i' for convenience.
Denote:

$$\sum_{i=1}^{n} X_i := X_{1:n}$$

Using some properties of iid Poisson distributions, we can reparameterise the joint distribution of all X_i's. Noting that, for any independent Poisson X and Y (from Kolaczyk appendix):

$$\begin{aligned} P(X, Y \mid \lambda_X, \lambda_Y) &= P(X \mid \lambda_X) P(Y \mid \lambda_Y) \\ &= P(X + Y \mid \lambda_X + \lambda_Y) P(X \mid X + Y, \frac{\lambda_X}{\lambda_X + \lambda_Y}) \end{aligned}$$

where $X + Y \mid \lambda_X + \lambda_Y \sim Poisson(\lambda_X + \lambda_Y)$ and $X \mid X + Y \sim Bernoulli(X + Y, \frac{\lambda_X}{\lambda_{X:Y}})$. (There's no detail in Kolaczyk as to the details of this decomposition, but is it to do with thinning and superposition of Poisson processes, and independent Poisson process? Something like:)

If we have two independent Poisson processes, $X \sim \text{Poisson}(\lambda_X)$ and $Y \sim \text{Poisson}(\lambda_Y)$, then, by *superposition of poisson processes*, we have that:

$$X + Y \sim \text{Poisson}(\lambda_X + \lambda_Y)$$

By *thinning of poisson processes*, we also have that, (from `https://www.stat.berkeley.edu/ blfang/STAT134/stat134_pp_notes.pdf`), if we take a Poisson process with rate α, and for each arrival (independently of other arrivals) flip a p-biased coin and create one process consisting of only the arrivals from when our coin lands heads, and another consisting of only when the coin lands tails, then these two resulting processes are independent Poisson processes with respective rates $p\alpha$ and $(1-p)\alpha$. In the rough proof for this is a the Poisson-binomial decomposition idea:

$$\begin{aligned} P(X = x, Y = y) &= P(x + y \text{ 'arrivals', and } x \text{ of these } x + y \text{ coin flips were heads}) \\ &= P(x + y \text{ 'arrivals'}) \times P(x \text{ head coin flips out of } x + y, \text{ with probability } p) \\ &= \text{Poisson}(\lambda_X + \lambda_Y) \times \text{Binomial}(x + y, p) \\ &= \vdots \quad \text{some algebra} \\ &= P(X = x) P(Y = y) \end{aligned}$$

Hence in our case, we are thinning $X + Y \sim \text{Poisson}(\lambda_X + \lambda_Y)$ with probability p, into two processes; $X \sim \text{Poisson}(\lambda_X)$ and $Y \sim \text{Poisson}(\lambda_Y)$, and so we have that:

$$\begin{aligned} \lambda_X = p(\lambda_X + \lambda_Y) \Rightarrow p &= \frac{\lambda_X}{\lambda_X + \lambda_Y} \\ \therefore P(X = x, Y = y) &= P(X = x) P(Y = y) \\ &= \text{Poisson}(\lambda_X + \lambda_Y) \times \text{Binomial}(x + y, \frac{\lambda_X}{\lambda_X + \lambda_Y}) \end{aligned}$$

Anyway, back to the decomposition. For a with a data-space vector, $(X_1, \ldots, X_B)$, working from the bottom up, and considering each adjacent 'pair' of X_i's from left to right (ie bases 1 and 2, then 3 and 4, until $B-1$ and B):

$$\begin{aligned}
P(X_1, \ldots, X_B \mid \lambda_1, \ldots, \lambda_B) &= \prod_{b=1}^{B} P(X_b \mid \lambda_b) \\
&= P(X_1 \mid X_{1:2}, \frac{\lambda_1}{\lambda_{1:2}}) P(X_{1:2} \mid \lambda_{1:2}) P(X_3 \mid X_{3:4}, \frac{\lambda_3}{\lambda_{3:4}}) P(X_{3:4} \mid \lambda_{3:4}) \times \cdots \times \\
&\quad P(X_{B-1} \mid X_{(B-1):B}, \frac{\lambda_{B-1}}{\lambda_{(B-1):B}}) P(X_{(B-1):B} \mid \lambda_{(B-1):B}) \\
&= P(X_1 \mid X_{1:2}, \frac{\lambda_1}{\lambda_{1:2}}) P(X_3 \mid X_{3:4}, \frac{\lambda_3}{\lambda_{3:4}}) \times \cdots \times P(X_{B-1} \mid X_{(B-1):B}, \frac{\lambda_{B-1}}{\lambda_{(B-1):B}}) \times \\
&\quad P(X_{1:2} \mid \lambda_{1:2}) P(X_{3:4} \mid \lambda_{3:4}) \times \cdots \times P(X_{(B-1):B} \mid \lambda_{(B-1):B})
\end{aligned}$$

Noting that we can recurse on the expressions in the second row, for example:

$$P(X_{1:2} \mid \lambda_{1:2}) P(X_{3:4} \mid \lambda_{3:4}) = P(X_{1:2} \mid X_{1:4}, \frac{\lambda_{1:2}}{\lambda_{1:4}}) P(X_{1:4} \mid \lambda_{1:4})$$

Then, eventually, our joint distribution becomes:

$$\begin{aligned}
P(X_1, \ldots, X_B \mid \lambda_1, \ldots, \lambda_B) = \\
&P(X_1 \mid X_{1:2}, \frac{\lambda_1}{\lambda_{1:2}}) P(X_3 \mid X_{3:4}, \frac{\lambda_3}{\lambda_{3:4}}) \times \cdots \times P(X_{B-1} \mid X_{(B-1):B}, \frac{\lambda_{B-1}}{\lambda_{(B-1):B}}) \times \\
&P(X_{1:2} \mid X_{1:4}, \frac{\lambda_{1:2}}{\lambda_{1:4}}) \times \cdots \times P(X_{(B-3):(B-2)} \mid X_{(B-3):B}, \frac{\lambda_{(B-3):(B-2)}}{\lambda_{(B-3):B}}) \times \\
&P(X_{1:B/2} \mid X_{1:B}, \frac{\lambda_{1:B/2}}{\lambda_{1:B}}) P(X_{1:B} \mid \lambda_{1:B})
\end{aligned}$$

It's worth noting how, despite us only parameterising one 'half' (in this case, the 'left' half) of the X_i's we are conditioning on at each stage of our multiscale decomposition, the distribution of the other 'half' can be obtained by a binomial distribution with a parameter equal to $1-p$, where p is the parameter of the binomial distribution in the left 'half':

$$\begin{aligned}
X_{1:B} &\sim Pn(\lambda_{1:B}) \\
X_{1:B/2} \mid X_{1:B} &\sim Binomial(X_{1:B}, \frac{\lambda_{1:B/2}}{\lambda_{1:B}}) \text{ and similarly,} \\
X_{((B/2)+1):B} \mid X_{1:B} &\sim Binomial(X_{1:B}, \frac{\lambda_{((B/2)+1):B}}{\lambda_{1:B}}), \text{ where} \\
\frac{\lambda_{1:B/2}}{\lambda_{1:B}} &= 1 - \frac{\lambda_{((B/2)+1):B}}{\lambda_{1:B}}
\end{aligned}$$

In summary, we have performed a complete 1:1 reparameterisation of the above, which is the Poisson distribution analogy to a Haar wavelet transform, where we have transformed the data and parameters into a multiscale space. We can represent the reparameterised data and parameters as:

$$\{(X_1, \ldots, X_n); (\lambda_1, \ldots, \lambda_n)\} \Leftrightarrow \{(X_{1:B}, X_{1:B/2}, \ldots, X_1, X_3, \ldots, X_{B-1}); (\lambda_{1:B}, \frac{\lambda_{1:B/2}}{\lambda_{1:B}}, \ldots, \frac{\lambda_1}{\lambda_{1:2}}, \frac{\lambda_3}{\lambda_{3:4}}, \ldots, \frac{\lambda_{B-1}}{\lambda_{(B-1):B}})\}$$

For convenience, we'll denote the parameters of the reparameterised space as $\boldsymbol{p}$, and bring back the subscript ^{i}:

$$\begin{aligned}
\boldsymbol{p^i} &= (\mu^i_{0,0}, p^i_{1,1}, \ldots, p^i_{J,1}, p^i_{J,2} \ldots, p^i_{J,2^{J-1}}) \\
&= (\lambda^i_{1:B}, \frac{\lambda^i_{1:B/2}}{\lambda^i_{1:B}}, \ldots, \frac{\lambda^i_1}{\lambda^i_{1:2}}, \frac{\lambda^i_3}{\lambda^i_{3:4}}, \ldots, \frac{\lambda^i_{B-1}}{\lambda^i_{(B-1):B}})
\end{aligned}$$

where $p_{s,l}$ denotes the parameter at scale $s = 1, \ldots, J$ and location $l = 1, \ldots, L_s$. In that sense, we could also relabel the data, $X_{s,l}$ similarly.

5.4 Multiscale Poisson models

Referring back to the above, our goal was to find the underlying λ function which characterises the data, and that our model, in the data space looked like a Poisson GLM:

$$\log(\lambda_j^i) = \mu_j^{(d)} + \beta_j^{(d)} g^i + u_j^i$$

After performing the multiscale transform, our new parameters represent parameters from binomial distributions, and hence we model them using logistic regressions. We define:

$$\begin{aligned}
\alpha_{0,1}^i &:= \log(\lambda_{1:B}^i) \\
\alpha_{1,1}^i &:= \text{logit}(p_{1,1}^i) = \log\left(\frac{\lambda_{1:B/2}^i / \lambda_{1:B}^i}{1 - \lambda_{1:B/2}^i / \lambda_{1:B}^i}\right) \\
&= \log(\lambda_{1:B/2}^i) - \log(\lambda_{B/2+1:B}^i)
\end{aligned}$$

This procedure continues at each scale, until we get to the last scale, J, where, for example:

$$\alpha_{J,1}^i := \text{logit}(p_{J,1}^i) = \log(\lambda_1^i) - \log(\lambda_2^i)$$

We can see the parallels between this multiscale transformation and that of the Haar wavelet transform – it contrasts adjacent portions of the underlying functional parameter to represent signals at different scales. Homogeneity in a region is represented in this space as $\log(\lambda_{s,l}) - \log(\lambda_{s,l+1}) = 0$, for any s and between two adjacent locations, $l, l+1$, hence making our parameterisation sparse, analogous to the wavelet transform. This is a one-to-one re-parameterisation of the original $(\lambda_b)_{b=1}^B$, and we represent it as:

$$\boldsymbol{\alpha^i} = (\alpha_{0,1}^i, \alpha_{1,1}^i, \ldots, \alpha_{J,2^{J-1}}^i)$$

Now we find ourselves in a similar position as in the WaveQTL situation, when we arrive at the wavelet space. We want to apply the functional mixed model approach to the $\boldsymbol{\alpha^i}$, hoping that we can perform some adaptive shrinking to denoise the underlying function, whilst capturing the important spatial signals at particular locations. Our model is now:

$$\alpha_{s,l}^i = \mu_{s,l}^{(ms)} + \beta_{s,l}^{(ms)} g^i + v_{s,l}^i$$

where I use the superscript (ms) to represent multiscale regression coefficient estimates, and $v_{s,l}^i$ is our random effect term to capture overdispersion.

Now, due to this random effect, the likelihood for this model, $\mathcal{L}((\mu_{s,l}^i, \beta_{s,l}^i); \alpha_{s,l}^i)$, is not available in closed form. This is where the laplace transform (or shall we just call it a Taylor expansion?) comes in.

The likelihood we don't have is the one of the parameters, and random effects from the linear regression expression above, where we observe the data, x. Aka, the underlying binomial (and poisson) distributions of the data, X, in terms of the parameters we have (using the $\alpha_{s,l}^i - \ldots$ expression more as a GLM):

$$\begin{aligned}
&P(X_{1:B/2} \mid X_{1:B}, \frac{\lambda_{1:B/2}}{\lambda_{1:B}}) \\
&\quad \therefore \mathcal{L}(p_{s,l}^i; x_{1:B/2}, x_{1:B}) = \binom{x_{1:B}}{x_{1:B/2}} (p_{s,l}^i)^{x_{1:B/2}} (1 - p_{s,l}^i)^{x_{((B/2)+1):B}} \\
&\quad \therefore \mathcal{L}(\alpha_{s,l}^i; x_{1:B/2}, x_{1:B}) = \binom{x_{1:B}}{x_{1:B/2}} \left(\frac{e^{\alpha_{s,l}^i}}{1 + e^{\alpha_{s,l}^i}}\right)^{x_{1:B/2}} \left(\frac{1}{1 + e^{\alpha_{s,l}^i}}\right)^{x_{((B/2)+1):B}} \\
\therefore \mathcal{L}((\mu_{s,l}^{(ms)}, \beta_{s,l}^{(ms)}); x_{1:B/2}, x_{1:B}) &= \text{ like above with } \alpha \text{ substituted for the linear regression expression}
\end{aligned}$$

The above is not available analytically, ESPECIALLY when you add in the random effect ($v_{s,l}^i$), as you would need to integrate it out, to get just the likelihood of the two parameters of interest, given the data.

5.5 Laplace transform – addressing non-independent likelihoods

Given a PDF which is a) smooth, and b) well peaked around the point of maxima, then a parameter's likelihood can be approximated by the PDF of a normal distribution using a 2-term Taylor expansion trick on the log of the PDF around $\widehat{\theta}$, the MLE estimate of that parameter. Some manipulation, matching of terms, and then taking exponential of both sides, we have that:

$$\mathcal{L}(\theta; \text{Data}) \approx g(\theta; \widehat{\theta}, I(\widehat{\theta})^{-1})$$

where g is the pdf of $\text{MVN}(\widehat{\theta}, I(\widehat{\theta})^{-1})$ (if θ is multivariable), and $I(\widehat{\theta})$ is the (observed) information matrix. Conversely, we can represent $(\widehat{\theta} \mid \theta, I(\widehat{\theta})^{-1})$ as having a pdf approximated by the distribution function of $\text{MVN}(\theta, I(\widehat{\theta})^{-1})$. In our case:

$$\theta = \begin{bmatrix} \mu^i_{s,l} \\ \beta^i_{s,l} \end{bmatrix}$$

and

$$\widehat{\theta} = \begin{bmatrix} \widehat{\mu}^i_{s,l} \\ \widehat{\beta}^i_{s,l} \end{bmatrix}$$

being the MLE estimators of the coefficients. And now, we have a likelihood in an analytical form, and also expressed as a gaussian pdf (nicer to work with), and no random effects left in the equation. Now we have a new issue, that's **non-independent likelihoods**.

Our MVN has a covariance matrix, $I(\widehat{\theta})^{-1}$, which is not guaranteed to be a diagonal matrix, hence is unlikely to be independent. This means that:

- Starting with independent priors,
- Non-independent likelihoods,
- Results in non-independent posteriors, which are not factorisable into products (inconvenient)

Hence, we have a trick from the Wakefield paper to convert such an MVN into the product of two independent Gaussians (an MVN with a diagonal covariance matrix). The goal here is to reparameterise μ to a μ^* such that (μ^*, β) are asymptotically independent.

Here is my full derivation/interpretation(?) of the Wakefield trick:

$$\begin{pmatrix} \widehat{\mu} \\ \widehat{\beta} \end{pmatrix} \overset{d}{\sim} \left(\begin{bmatrix} \mu \\ \beta \end{bmatrix}, \begin{bmatrix} I_{0,0} & I_{0,1} \\ I_{1,0} & I_{1,1} \end{bmatrix}^{-1} \right)$$

where $I(\theta) = \begin{bmatrix} I_{0,0} & I_{0,1} \\ I_{1,0} & I_{1,1} \end{bmatrix}$ and

$$\begin{aligned}
[I(\theta)]^{-1} &= \frac{1}{I_{00}I_{11} - I_{01}^2} \begin{bmatrix} I_{11} & -I_{01} \\ -I_{01} & -I_{11} \end{bmatrix} \\
\therefore \text{se}(\widehat{\mu})^2 &= \frac{I_{11}}{I_{00}I_{11} - I_{01}^2}, \text{ and} \\
\therefore \text{se}(\widehat{\beta})^2 &= \frac{I_{00}}{I_{00}I_{11} - I_{01}^2}, \text{ and} \\
\therefore \text{cov}(\widehat{\mu}, \widehat{\beta}) &= -\frac{I_{01}}{I_{00}I_{11} - I_{01}^2} \\
\Rightarrow I_{01} &= -\text{cov}(\widehat{\mu}, \widehat{\beta})(I_{00}I_{11} - I_{01}^2)
\end{aligned}$$

So,

$$\begin{aligned}
\mu^* = \mu + \frac{I_{01}}{I_{00}}\beta &= \mu - \frac{\text{cov}(\widehat{\mu}, \widehat{\beta})(I_{00}I_{11} - I_{01}^2)}{I_{00}}\beta \\
&= \mu - \frac{\text{cov}(\widehat{\mu}, \widehat{\beta})}{\text{se}(\widehat{\beta})^2}\beta
\end{aligned}$$

and similarly,

$$\widehat{\mu}^* = \widehat{\mu} - \frac{\text{cov}(\widehat{\mu}, \widehat{\beta})}{\text{se}(\widehat{\beta})^2}\widehat{\beta}$$

And it can be shown that:

$$\begin{aligned}
\text{cov}(\widehat{\mu}^*, \widehat{\beta}) &= E[(\widehat{\mu}^* - E[\widehat{\mu}^*])(\widehat{\beta} - E[\widehat{\beta}])] \\
&= E[(\widehat{\mu} - \frac{\text{cov}(\widehat{\mu}, \widehat{\beta})}{\text{se}(\widehat{\beta})^2}\widehat{\beta} - E[\widehat{\mu} - \frac{\text{cov}(\widehat{\mu}, \widehat{\beta})}{\text{se}(\widehat{\beta})^2}\widehat{\beta}])(\widehat{\beta} - \beta)] \\
&= E[(\widehat{\mu} - \frac{\text{cov}(\widehat{\mu}, \widehat{\beta})}{\text{se}(\widehat{\beta})^2}\widehat{\beta} - \mu + \frac{\text{cov}(\widehat{\mu}, \widehat{\beta})}{\text{se}(\widehat{\beta})^2}\beta])(\widehat{\beta} - \beta)] \\
&= E[[(\widehat{\mu} - \mu) - \frac{\text{cov}(\widehat{\mu}, \widehat{\beta})}{\text{se}(\widehat{\beta})^2}(\widehat{\beta} - \beta)][\widehat{\beta} - \beta]] \\
&= E[(\widehat{\mu} - \mu)(\widehat{\beta} - \beta) - \frac{\text{cov}(\widehat{\mu}, \widehat{\beta})}{\text{se}(\widehat{\beta})^2}(\widehat{\beta} - \beta)^2] \\
&= \text{cov}(\widehat{\mu}, \widehat{\beta}) - \frac{\text{cov}(\widehat{\mu}, \widehat{\beta})}{\text{se}(\widehat{\beta})^2}E[(\widehat{\beta} - \beta)^2] \\
&= \text{cov}(\widehat{\mu}, \widehat{\beta}) - \text{cov}(\widehat{\mu}, \widehat{\beta}) = 0
\end{aligned}$$

Therefore we have:

$$\begin{aligned}
\mu^*_{s,l} &= \mu_{s,l} + \frac{I_{s,l}(0,1)}{I_{s,l}(0,0)}\beta_{s,l} \\
&= \mu_{s,l} - \frac{\text{cov}(\widehat{\mu}_{s,l}, \widehat{\beta}_{s,l})}{\text{se}(\widehat{\beta}^2_{s,l})}\beta_{s,l} \\
\widehat{\mu}^*_{s,l} &= \widehat{\mu}_{s,l} + \frac{I_{s,l}(0,1)}{I_{s,l}(0,0)}\widehat{\beta}_{s,l} \\
&= \widehat{\mu}_{s,l} - \frac{\text{cov}(\widehat{\mu}_{s,l}, \widehat{\beta}_{s,l})}{\text{se}(\widehat{\beta}^2_{s,l})}\widehat{\beta}_{s,l}
\end{aligned}$$

Now, we can write (asymptotically):

$$P((\widehat{\mu}^*_{s,l}, \widehat{\beta}_{s,l}) \mid (\mu^*_{s,l}, \beta_{s,l}), [I^*(\widehat{\theta})]^{-1}) = P(\widehat{\mu}^*_{s,l} \mid \mu^*_{s,l}, \text{se}(\widehat{\mu}^*_{s,l})^2) \times P(\widehat{\beta}_{s,l} \mid \beta_{s,l}, \text{se}(\widehat{\beta}_{s,l})^2)$$

where (surely this isn't correct due to covariance term between the mu-hat and beta-hat. But it looks like what was said (or something like it) in pg 151 of Xing):

$$\text{se}(\widehat{\mu}^*_{s,l})^2 = \text{se}(\widehat{\mu}_{s,l})^2 + I^2_{s,l}\text{se}(\widehat{\beta}_{s,l})^2$$

Note that due to our Wakefield trick, this yields an asymptotic aproximate bayes factor only (This is what Wakefield's paper and Xing's paper say. I'm not quite sure where we've appealed to any asymptotics here, unless the asymptotic is with regards to some distributional assumption he's made in his paper, and a normal approximation he used for his likelihood. If that's the case, then we don't have a similar concern, as the Laplace approx for normal is not asymptotically based?) I think the asymptotic property is from the fact that the inverse of the observed information matrix is an estimator of the asymptotic covariance matrix, hence i think we can only make the statements about the standard errors being equal to parts of the information matrix, asymptotically. Hence, all of the things probably only hold asymptotically, and hence the likelihood factorising into a product only holds asymptotically. Possibly also that the likelihood-maximising parameter is only equal to the MLE, asymptotically? (Or is it by definition?) **Note that you shouldn't really mention independence, just mention that our likelihood now conveniently factorises into the product of two distributions.**

5.5.1 Notes and practicalities

There are some notes and practicalities, explained in Xing's paper, that I need clarification on:

- Section 2 is **'Modelling the sequencing profile from a single sample'**, looks similar, but seems to model the $\widehat{\alpha}_i, \text{se}(\widehat{\alpha}_i)$ directly, rather than caring about μ or β. What part of this setup is different which allows them to do this?
- Section 3 is **'Estimating differences between multiple samples'**, but more similar to ours in the way it models, in that it mentions we need estimates and standard errors for the μ and β, unlike in section two, where *ash* is applied to α directly.
- It does mention three major modifications:
  - If g is two-level categorical, compute estimate and standard errors of μ and β from $\widehat{\alpha}$ and $\text{se}(\widehat{\alpha})$ directly. If g is quantitative, something about Binomial pseudocounts.
  - Then, it mentions we do the reparameterisation of μ and β to have independent likelihoods

5.5.2 Ash

A lot of the following derivations are based on the framework of *ash*, which stands for **A**daptive **Sh**rinkage, based off the work of Stephens (2017). Key points:

- It's an empirical bayes procedure, where:
  - Some underlying effect, β, which we want to estimate, assess uncertainty, and perform hypothesis tests (like significance of difference to 0)
  - Available data: estimates, $\widehat{\beta}$, of actual effects, and corresponding standard errors, $\text{se}(\widehat{\beta})$ (perhaps from some procedure which allows us to derive MLEs)
  - **Goal:** compute posterior for β given the provided info:

$$(\beta \mid \widehat{\beta}, \text{se}(\widehat{\beta})) \propto (\widehat{\beta} \mid \beta, \text{se}(\widehat{\beta})) \times (\beta \mid \text{se}(\widehat{\beta}))$$

  - For one observation, β_j we have the following distributions (i've just dropped the β_j here cause i'm lazy):
  - Definitely can incorporate a prior which is a mix of point mass at 0, and mixture of *zero-mean* normal distributions, where the *variances are known*. I'm guessing the reason we go for an (arbitrarily large) mixture of normals with different, (but known), variances, is to capture the possibility that our data could fit into one of many shrinkage priors (ie to account for the wide range of variability our data could have). Also makes computation a lot easier.
    - This is a key differentiator to the WaveQTL situation, where σ in the prior was a parameter, which had its own distribution (an inverse gamma), which gives another 'level' to our hierarchical model, but adds extra complexity (must be integrated out)
  - From Xing: having sufficiently large and dense grid of σ allows arbitrary accuracy in approximating any mixture of normals (for shrinkage), including, as a special case, spike and slab.
  - **Prior:**
    γ is a discrete (latent) random state variable taking values $m \in 0, \ldots, M$, according to a pmf.

$$P(\gamma = m) = \pi_m$$

$$p(\beta \mid \text{se}(\widehat{\beta})) \sim \pi_0 \delta_0 + \sum_{m=1}^{M} \pi_m N(0, \sigma_m^2)$$

  - **Likelihood:**

$$p(\widehat{\beta} \mid \beta, \text{se}(\widehat{\beta})) \sim N(\beta, \text{se}(\widehat{\beta})^2)$$

  - Parameterising γ is down to EM algorithm (although i think they use interior point methods as more efficient for this type of optimisation problem)

- Benefits: flexibility in being able to incorporate a large number of unimodal prior distributions (eg a mixture of Gaussians), which is still capable of natural shrinkage estimation, adaptive to the signal versus noise content in a given observation, whilst still being analytically tractable and computationally nice
- Remaining benefits: to do with interpretations, error rates (not sure if apply to us)

So, using the Ash framework, we have, similar to before, a functional mixed model framework:
Likelihood:

$$\begin{aligned}\alpha^i_{s,l} &= \mu_{s,l} + \beta_{s,l} g^i + v^i_{s,l} \\ &= \mu^*_{s,l} - \frac{I_{s,l}(0,1)}{I_{s,l}(0,0)}\beta_{s,l} + \beta_{s,l} g^i + v^i_{s,l}\end{aligned}$$

which then gets approximated and split out to two independent components:

$$\begin{aligned}p(\widehat{\beta} \mid \beta, \text{se}(\widehat{\beta})) &\sim N(\beta, \text{se}(\widehat{\beta})^2) \\ p(\widehat{\mu}^* \mid \mu^*, \text{se}(\widehat{\mu}^*)) &\sim N(\mu^*, \text{se}(\widehat{\mu}^*)^2)\end{aligned}$$

Priors are now a bit different to WaveQTL; no σ prior, as we used a range of fixed σ. Also, we now induce a similar prior over μ^*, as it has components of β, and so it will have a similar underlying structure, and also, as this quantity is now of interest to us, as it is related to the effect size, $\beta^{(d)}$, we are interested in. Hence, to parameterise, we can run two EM algorithms on each parameter (μ^* and β) due to their independence.

To simplify the notation, from here on end, we will only consider the case of β, with μ being identical. Hence all parameters and hyperparameters relate to the parameterisation of π as it relates to βsl, and the distributions of β.

$$\begin{aligned}\gamma_{sl} &\sim \text{Some sort of multinoulli dist'n, for } \textbf{no HMT version} \\ \gamma_{sl} &\sim \text{Joint distribution structure induced by a tree, for } \textbf{HMT version} \\ \pi_{sl,m} &= P(\gamma_{sl} = m) \\ \beta_{sl} &\sim \pi_{sl,0}\delta_0 + \sum_{m=1}^{M} \pi_{sl,m} N(0, \sigma^2_m)\end{aligned}$$

Or should we write it using random variables:?

$$\beta_{sl} \sim \mathbb{1}\{\gamma_{sl} = 0\}\delta_0 + \sum_{m=1}^{M} \mathbb{1}\{\gamma_{sl} = m\} N(0, \sigma^2_m)$$

Although, realistically, we will perform some parameter tying, for example, having only one hyperparameter govern the probability for one level of the tree, meaning we get:

$$\begin{aligned}\pi_{s,m} &= P(\gamma_{sl} = m) \\ \beta_{sl} &\sim \pi_{s,0}\delta_0 + \sum_{m=1}^{M} \pi_{s,m} N(0, \sigma^2_m)\end{aligned}$$

Then comes the EM algorithm to help us find $\pi^{(\textbf{MLE})} = (\pi^{(MLE)}_{sl,0}, \ldots, \pi^{(MLE)}_{sl,M})_{s,l}$ in an empirical bayes approach. (forgot the hats on the pi as i was lazy)

5.6 EM algorithm

A lot of this has been adapted off the background of Ash, which formally, considers, for a PoissonBinomial model, a very similar setup to the above, without the point mass at 0. We can take all the workings there, convert indexing

from $i = 1, \ldots, n$ to scale-location s, l, and assume tree-level tying for now (same parameters for same scale, s). Then:

$$\begin{aligned}
\beta_{s,l} &:= \alpha_{s,l} \\
\widehat{\beta}_{s,l} &:= x_{s,l} \\
\text{se}(\widehat{\beta}_{s,l}) &:= y_{s,l} \\
\therefore D = (D_{0,1}, \ldots, D_{J,2^{J-1}}),\ & D_{s,l} := (\widehat{\beta}_{s,l}, \text{se}(\widehat{\beta}_{s,l})) \\
\gamma = (\gamma_{0,1}, \ldots, \gamma_{J,2^{J-1}}) &:= Z \\
P(\gamma_{s,l} = m) = \pi_{s,m},\ & m = 0, \ldots, M
\end{aligned}$$

with $\pi^{(l)}$ denoting the vector of π parameters after step l of the EM algorithm.

The key results are thus:

E-step:

$$\begin{aligned}
P(\gamma_{s,l} = m \mid D_{s,l}, \pi^{(l)}) &= \frac{\pi^{(l)}_{s,m} P(\gamma_{s,l} = m \mid D_{s,l})}{\sum_{n=0}^{M} \pi^{(l)}_{s,n} P(\gamma_{s,l} = n \mid D_{s,l})} \\
&= \frac{\pi^{(l)}_{s,m} \text{BF}_{s,l}(\sigma_m^2)}{\sum_{n=0}^{M} \pi^{(l)}_{s,n} \text{BF}_{s,l}(\sigma_n^2)}
\end{aligned}$$

M-step:

$$\pi^{(l+1)}_{s,m} = \frac{\sum_{j=1}^{L_s} P(\gamma_{s,j} = m \mid D_{s,j}, \pi^{(l)})}{L_s}$$

where L_s is the number of locations (different data points) in scale s.

So the key becomes **how to get an explicit expression for the Bayes Factor**, as required above. This is also from the Ash working papers:

$$\begin{aligned}
P(D_{s,l} \mid \gamma_{s,l} = m) &= \int \mathcal{L}(\beta_{s,l}; \widehat{\beta}_{s,l}, \text{se}(\widehat{\beta}_{s,l})) P(\beta_{s,l} \mid \gamma_{s,l} = m, \pi, \sigma) d\beta_{s,l} \\
&\ldots \text{ lots of lines of integrals } \ldots \\
&= \frac{C(\widehat{\beta}_{s,l}, \text{se}(\widehat{\beta}_{s,l})^2)}{\sqrt{2\pi(\sigma_m^2 + \text{se}(\widehat{\beta}_{s,l})^2)}} \exp\Big(- \frac{\widehat{\beta}^2_{s,l}}{2(\sigma_m^2 + \text{se}(\widehat{\beta}_{s,l})^2)} \Big)
\end{aligned}$$

Now in Ash, it finds its approximate BF (ABF – approximation, because of the likelihood approximation on both denom and numerator) using the following form:

$$BF_{s,l}(\sigma_m^2) = \frac{P(D_{s,l} \mid \gamma_{s,l} = m)}{P(D_{s,l} \mid \alpha_{s,l} = 0)}$$

But surely, this is the same as, in our case, doing (**confirmed, yes it is**):

$$BF_{s,l}(\sigma_m^2) = \frac{P(D_{s,l} \mid \gamma_{s,l} = m)}{P(D_{s,l} \mid \gamma_{s,l} = 0)}$$

as $\gamma_{s,l} = 0$ is the case where $\beta_{s,l} = 0$? Anyway, in my opinion, as $P(D_{s,l} \mid \gamma_{s,l} = 0)$ is the point mass at 0 situation, there is no variance in this case, and so we should get a similar result to the Ash sheet:

$$P(D_{s,l} \mid \gamma_{s,l} = 0) = \frac{C(\widehat{\beta}_{s,l}, \text{se}(\widehat{\beta}_{s,l})^2)}{\sqrt{2\pi(\text{se}(\widehat{\beta}_{s,l})^2)}} \exp\Big(- \frac{\widehat{\beta}^2_{s,l}}{2\text{se}(\widehat{\beta}_{s,l})^2} \Big)$$

meaning we get exactly the same result as in Ash for the ABF:

$$\begin{aligned}
\text{BF}_{sl}(\sigma_m^2) &= \sqrt{\lambda} \exp[T^2(1-\lambda)/2], \\
\lambda &= \frac{\text{se}(\widehat{\beta}_{sl})^2}{\text{se}(\widehat{\beta}_{sl})^2 + \sigma_m^2} \\
T &= \frac{\widehat{\beta}_{sl}}{\text{se}(\widehat{\beta}_{sl})}
\end{aligned}$$

5.7 Posterior distribution for β

For a posterior, we get:

$$P(\beta_{sl} \mid D_{sl}, \gamma_{sl} = m) \propto \mathcal{L}(\beta_{sl}) P(\beta_{sl} \mid \gamma_{sl} = m)$$

$$\dots \text{ some workings } \dots$$

$$(\beta_{sl} \mid D_{sl}, \gamma_{sl} = m) \overset{d}{\sim} N(\frac{\sigma_m^2 \widehat{\beta}_{sl}}{\sigma_m^2 + \text{se}(\widehat{\beta}_{sl})^2}, \frac{\sigma_m^2 \text{se}(\widehat{\beta}_{sl})^2}{\sigma_m^2 + \text{se}(\widehat{\beta}_{sl})^2})$$

Except, our notable case here is the case where $\gamma_{sl} = 0$, where we have a point mass at zero:

$$P(\beta_{sl} = j \mid D_{sl}, \gamma_{sl} = 0) = \begin{cases} 1 & j = 0 \\ 0 & \text{otherwise} \end{cases}$$

Then, like in WaveQTL, our posterior becomes a mix of a mixture of gaussians, and a point mass at zero, once we marginalise over γ_{sl}:

$$P(\beta_{sl} \mid D_{sl}) = \sum_{m=0}^{M} P(\gamma_{sl} = m \mid D_{sl}, \widehat{\pi}) P(\beta_{sl} \mid D_{sl}, \gamma_{sl} = m)$$

$$= P(\gamma_{sl} = 0 \mid D_{sl}, \widehat{\pi}) \text{ (point mass at 0) } + \sum_{m=1}^{M} P(\gamma_{sl} = m \mid D_{sl}, \widehat{\pi}) N(\dots)$$

where $N(\dots)$ is the normal distribution from before.

5.8 $\beta^{(d)}$ means and variances

The goal is to go from $\beta_{sl}^{(ms)}$ to $\beta_{sl}^{(d)}$. This is all from *multiseq*. We place a few simplifying assumptions to simplify our illustration to only consider the difference between two groups:

- $g^i \in \{0, 1\}$, ie there are only two groups, indicated by either 0 or 1
- No random effect considered here, ie $u_b^i = 0$ and $v_{sl}^i = 0$

Most of the ideas here work with estimating β by differencing quantities where $g^i = 1$ versus where $g^i = 0$.

- Is it straightforward to go to a situation with more groups, or a quantitative covariate? Would the exercise involve just performing this analysis, repeatedly, against the mean level where $\beta = 0$?.
- Also, how would we account for when the random effect is not equal to 0? Would the result be quantities which involve random effects in them?

There are three main steps (superscript, ^{m} denotes group, $m \in \{0, 1\}$:

1. Model relationship between α^m and λ^1.
    - This results in equations like (18) and (19) in multiseq
    - Exact form depends on which base we are looking for λ for, and how many 'left' or 'right' nodes we travel down from the top of the tree to the desired base. (Starting from the top of the tree, p for a movement down a left branch, $q = 1 - p$ for a right branch)
2. Express α, p and q quantities in the terms we estimated in our GLM; β_{sl}, μ_{sl}
    - This gives us (20) - (24) in multiseq
3. Represent $\beta_b^{(d)} = \log(\lambda_b^1) - \log(\lambda_b^0)$ as sum of α's and either p's or q's:
    - Ends up with (27) in multiseq
    - A sum of 'top-level' effects (function of top scaling coefficient, then a sum of the differences of respective p and q terms, at each scale, depending on which path down the tree is required to get to λ_b.

For example, for the leftmost base, $\beta_1^{(d)}$:

$$\beta_1^{(d)} = \alpha_{01}^1 - \alpha_{01}^0 + \sum_{s=1}^{J}[\log p_{s1}^1 - \log p_{s1}^0]$$

Any reason why the last sum is split out in multiseq? **(confirmed – HJ said, no reason, really...)**

$$\beta_1^{(d)} = \alpha_{01}^1 - \alpha_{01}^0 + \sum_{s=1}^{J-1}[\log p_{s1}^1 - \log p_{s1}^0] + (\log p_{J1}^1 - \log p_{J1}^0)$$

Anyway, supposedly there is no analytic form for the posterior of $\beta_1^{(d)}$, (why? Is it because there are too many parameters requiring integrating out, such as μ_{sl} and β_{sl}?) hence we need to approximate the posterior mean and variance of these quantities using Taylor expansions. For any other posterior-based inference (eg credible intervals, etc), we can simulate samples from the distribution of $\beta_b^{(d)}$. Due to the independence of μ^* and β, and as all the above terms are functions of these parameters, we have that means and variances are additive:

$$E[\beta_1^{(d)}] = \text{ sum of the expectations of its constituents}$$
$$V[\beta_1^{(d)}] = \text{ sum of the variances of its constituents}$$

For $E[\alpha_{sl} \mid \gamma_{sl} = m, D_{sl}]$, we can compute this directly from what we know about β_{sl} and μ_{sl}^* (we have their posteriors in closed form, and know their means and variances). For the last sum invoving p's and q's, this takes a little more work (Delta method/Taylor Expansion):

$$E[g(X)] \approx g(E[X]) + \frac{1}{2}V(X)g''(E[X])$$
$$V[g(X)] \approx (g'(E[X]))^2V(X) - \frac{1}{4}(g''(E[x]))^2(V(X))^2$$

Where the $V(g(X))$ derivation can be done by:

$$V(g(X)) = E[(g(X))^2] - E[(g(X))]^2$$

And then we can do a taylor expansion around $(g(X))^2$ to subsequently find expectations of these terms, and hence get the first term in the equation above. With all that behind us, we can evaluate, for each group, $m \in \{0, 1\}$:

$$\log(p_{sl}^m) = g(\alpha_{sl}^m), \quad \text{where } g(x) = \log(\frac{e^{\alpha_{sl}^m}}{1 + e^{\alpha_{sl}^m}})$$
$$g'(\alpha_{sl}^m) = \frac{1}{1 + e^{\alpha_{sl}^m}}$$
$$g''(\alpha_{sl}^m) = -\frac{e^{\alpha_{sl}^m}}{(1 + e^{\alpha_{sl}^m})^2}$$

Leading to some pretty ugly looking results as per the working paper. Then we use $\log(q) = \log(1-p) = \log(\frac{e^{-\alpha}}{1+e^{-\alpha}}) = g(-\alpha)$, to get the remaining quantity (a neat trick), so now we have both expectation and variance quantities of these posteriors:

$$(\log(p_{sl}^m) \mid D_{sl}, \gamma_{sl} = m),$$
$$(\log(1 - p_{sl}^m) \mid D_{sl}, \gamma_{sl} = m)$$

In the no-HMT model, as the γ are independent, each α_{sl} posterior mean and variance could be calculated independently of each other, and same for the remaining p and q quantities.

Does it matter whether we take $E(\beta_{sl} \mid D_{sl})$ (directly from the posterior, post after marginalising γ, and then try and delta method that, or do as we did above – expectation of β_{sl} conditional on γ, delta method, then marginalise over γ? I'm guessing we chose the second way as it's easier to do.

5.9 Data space Bayes Factor

In the paper, there's a data-space BF (here, notation is for the binary γ case where it's either 1 ($\beta \neq 0$) or 0 ($\beta = 0$):

$$BF_{sl}(x, g) = \frac{P(x \mid g, \gamma_{sl} = 1)}{P(x \mid g, \gamma_{sl} = 0)}$$

How exactly do we get this quantity; it involves data space x?

6 multiseq - with HMT

6.1 New EM algorithm

Similar to above, but now needing **upward-downward algorithm**. As well as parameterising transition probabilities ($\varepsilon^{mn}_{sl,p(sl)}$), the only real modification here is the need to use the ABF (derived above) in place of the gaussian density as per Crouse's paper. (Similar to how we use BF in WaveQTL). Everything else is the same.

6.2 New Posterior distribution for β

As per WaveQTL with HMT, the posteriors here are different. In the no-HMT case, they dropped out nicely from the E-step. In the HMT version, they also drop out nicely from the E-step, but after doing the backward-forward algorithm steps, and carefully computing the joint (parent-child) and marginal distributions of gamma for each sl. We then need to divide the joint by the marginal to get the marginal of the child, etc. Hence, our posterior form is the same, but the probabilities have a lot more complexity in their calculation.

6.3 New $\beta^{(d)}$ means and variances

Similar to the no-HMT case, we will find the expectations and means of $(\alpha_{sl} \mid D_{sl}, \gamma_{sl} = m)$, ie. conditional on the γ_{sl}. We'll then use delta method, and then marginalise over all the possible states of α_{sl}.

The key here is that the HMT acts upon states, scale by scale ($s = 1, \ldots, J$), so we can do the above procedure, but scale-by-scale only.

I believe that's the dynamic programming algorithm the 'wavelets poisson' working paper is detailing. Our utlimate goal is to find $E(\beta_b^{(d)})$ and $V(\beta_b^{(d)})$, and to do that in a HMT framework we need to find:

- Means, variances of top-level coefficients for different groups, $\alpha^1_{01}, \alpha^0_{01}$ (required for all)
- Means, variances of the specific location, at each scale $\log(p^1_{sl}), \log(p^0_{sl}), \log(q^1_{sl}), \log(q^0_{sl})$ which we use, from root to leaf of tree, to get to a specific base, b
- And combine all of these to get the necessary $E[\beta_b^{(d)}]$. For the leftmost base, $\beta_b^{(d)}$, for example, we need:

$$\sum_{s=1}^{J} [\log(p^1_{s1}) - \log(p^0_{s1})] = \sum_{s=1}^{J} \log(p^1_{s1}) - \sum_{s=1}^{J} \log(p^0_{s1})$$

  which boils down to finding means, variances and probabilities of $(\sum_{s=1}^{J} \log(p^m_{s1}) \mid \gamma_{sl} = m, \boldsymbol{D})$ where $\boldsymbol{D}$ is a vector of all data (estimates and std errors) at all scales and locations. This quantity is no longer just locally dependent on the D_{sl}, but also on the data at other scale-locs (similar to WaveQTL)

To simplify things, here's our example:

- Find the quantities for the leftmost base (corresponds to finding estimates of the sum of J $\log p_{s1}$ values. Finding posterior means and variances of α quantities are not done here.
- We'll drop the subscript ^{m} for group indicator for now – the process is same for both groups, 0 and 1, or however many groups there are.

- Assume the relevant p values are indexed, $s = 1, \ldots, J$, corresponding to the specific locations required, at each of the J scales. Also, to unify notation, denote $\eta_s := \log p_{s1}$. In this case, we have that:

$$\log p_{11} = \eta_1$$
$$\vdots$$
$$\log p_{J1} = \eta_J$$

Hence, our dynamic programming algorithm attempts to:

- Our goal becomes finding mean, variance, and probability distribution of $(\sum_{s=1}^{J} \eta_s \mid \gamma_{sl} = m, \boldsymbol{D})$ for each $m = 0, \ldots, M$, and we need to perform this procedure b times for each unique path, traced through the tree, by each base $b = 1, \ldots, B$. We would also need to do this for as many groups as required.
- Wow this is computationally intensive.
- Due to sequential nature of tree, the properties of the sum, $\sum_{s=1}^{J} \eta_s$ must be **performed sequentially, from the top of the tree to the bottom, as $s = 1$ has an influence on $s = 2$, etc, due to HMT structure**

7 Flowchart summary

I never got time to do this.