1. Inferring clonal population structure from SNV data

In this module we consider the problem of inferring clonal population structure using bulk sequencing data. We specifically consider the case of using SNVs as our clonal markers.

1.1. Problem

We consider the problems of how to use high throughput sequencing to infer the clonal population structure of a tumour. This problem is somewhat old now, but still remains relevant when looking at large cancer cohort datasets that are continuing to be generated. The basic ideas of deconvolution are also appearing in other areas so this module should provide some useful insight.

Precisely we will use read count data from SNVs to

1. Infer what proportion of cancer cells harbouring a mutation
2. Infer what mutations share the same evolutionary history i.e. originate at the same time and are lost in the same subsets of clones

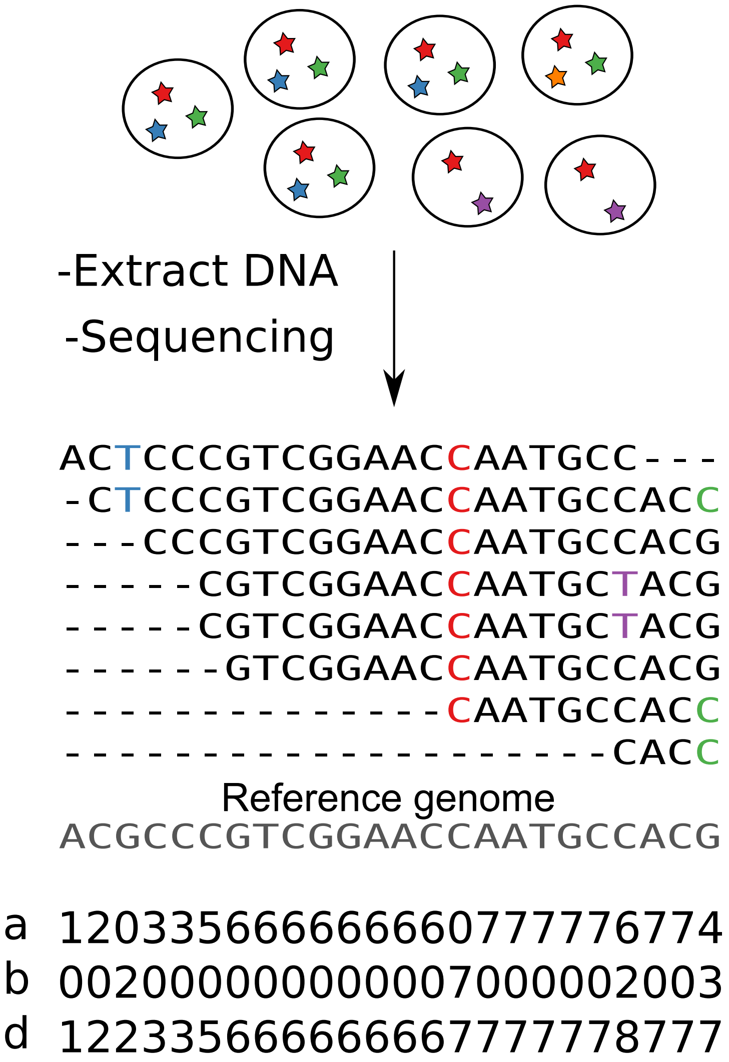

Figure 1. Schematic of bulk sequencing for a tumour experiment. At the top we have the input cell population, where stars indicate mutations. In the middle we show the aligned reads obtained from performing bulk sequencing. Positions in reads are colour coded to match the mutations at the top. Note the proportion of reads with a variant is roughly similar to the proportion of input cells with the mutation. At the bottom we show the summarised read counts which we will use for modelling.

We will assume that we have collected one or more tumour samples from a patient and performed high throughput sequencing. We assume that SNVs have already been identified using a standard variant caller tool. We summarise our aligned read count format in terms of three quantities for each SNV (see Figure 1).

- a - The number of reads which match the reference allele.
- b - The number of reads which match the variant allele.
- $d = a + b$ - The total depth of coverage at the locus.

To achieve our goals will we need to develop two parts to the model. The first part will be a way to account for the effect of *mutational genotype* and *normal contamination*. Mutational genotype refers to the fact that not all mutation will be heterozygous diploid events due to coincident copy number variation. Normal contamination refers to the fact that we also sequence non-malignant cells in real tumours. The second part will be a mechanism to cluster the mutations. Here we will use the formalism of mixture models. There is one challenging issue, which is that we do not know how many clones there are in the sample(s). We will address this problem using the Dirichlet process.

1.2. Modelling mutation genotype

The first issue we tackle is the problem of mutational genotype. Our goal in this section is to define a probabilistic model that links the observed read count data with cellular prevalence. We can then apply the standard Bayesian machinery to compute estimates of cellular prevalence.

1.2.1. Background

We will make the assumption that a point mutation only occurs once at a locus during the evolutionary history of tumour. This is often referred to as the *infinite sites assumption*. This assumption is motivated by the fact that the mutation rate is usually relatively low compared to the total size of the genome. It can break down at mutational hot spots, and this has been observed, for example when multiple substitutions are observed at a single locus. So this assumption represents our first approximation to truth.

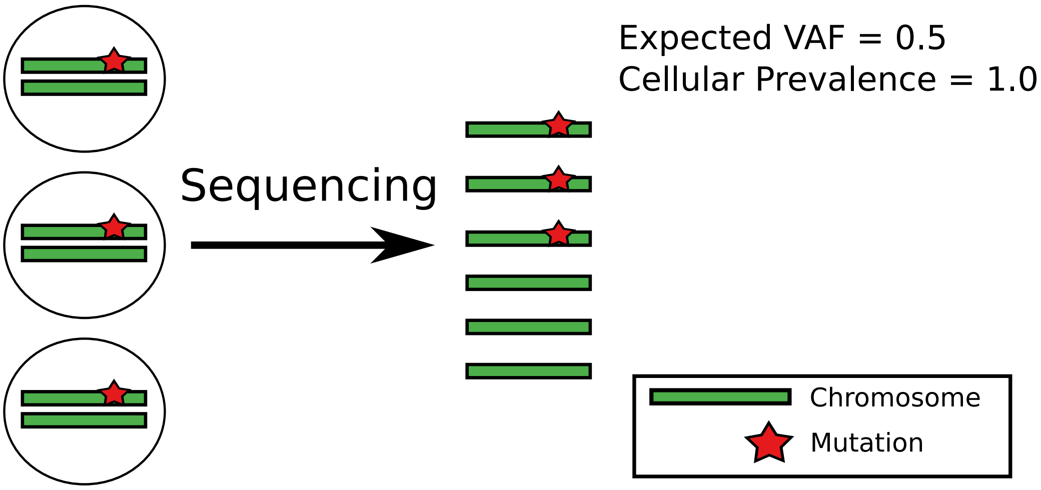

Figure 2. Example of a heterozygous diploid mutation showing why the variant allele frequency (VAF) is not the same proportion of cells harbouring the mutation (cellular prevalence). On the left we have the input population of cells which all have the mutation. On the right we have the observed sequence data where only half the reads (on average) have the mutation.

The issue we face is that tumour cells are not haploid. In the absence of copy number events, they are at least diploid at any autosomal locus. Figure 2 illustrates the basic problem. In this example we have a set of cells which are all diploid and heterozygous for a mutation. All the cells have the mutation, but only half the reads are expected to show the mutation. This suggests that if we simply double the observed variant allele frequency (VAF) we can get a reasonable estimate of the cellular prevalence (proportion of cells with the mutation). The problem is of course that cancers are aneuploid. Figure 3 illustrates the challenges that copy number variation poses.

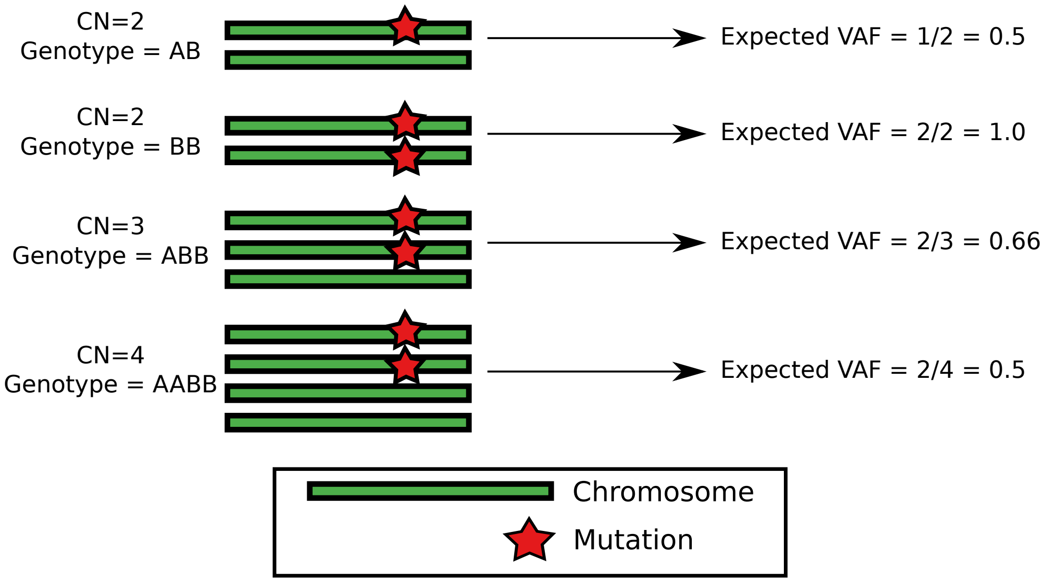

Figure 3. Effect of mutational genotype on observed VAF.

1.2.2. Population structure

To overcome the issue of mutational genotype we need to start making a model of how the observed data is generated. We break the process down into the following steps

1. We select a cell proportional to how prevalent it is in the input sample.
2. Given the cells genotype we then select a chromosome at random. Thus the probability of selecting a mutation is proportional to how many copies of the chromosome have the mutation.
3. Finally we introduce a small error probability ε. We assume the probability is the same whether we truly sampled the reference and observed an erroneous variant, or vice versa. Because of this symmetry the error term only appears in cases when all of the chromosomes have the reference or the variant allele.

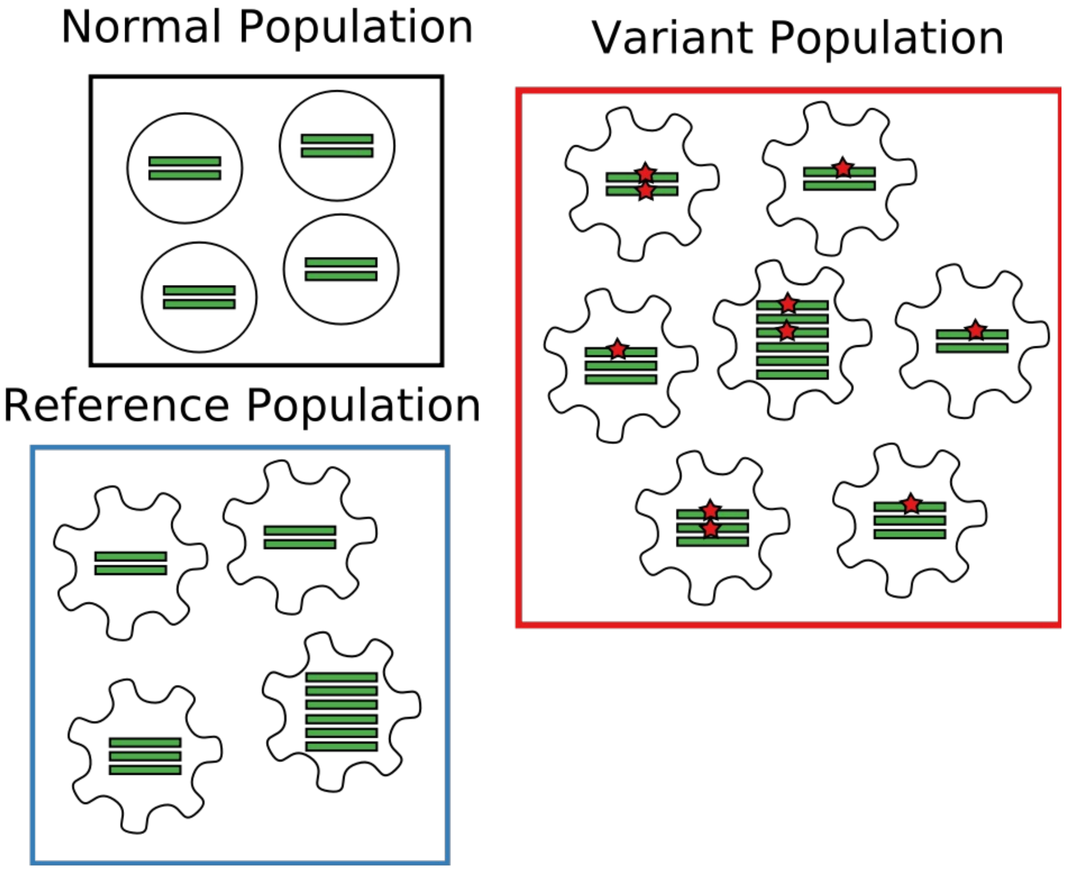

Figure 4. Illustration of the assumed population structure. Here all populations are defined with respect to a single mutation. The circular cells are non-malignant and the irregularly shaped ones are malignant.

Figure 4 illustrates the basic model. We decompose the population into three parts

- Normal population - The non-malignant cells
- Reference population - The malignant cells without the mutation
- Variant population - The malignant cells with the mutation

These populations are all defined with respect to a single mutation. If we were to pick a different mutation the cells which belong to the reference and variant population would change. This is a common point of confusion as people do not understand how a malignant cell does not have mutations. They do have mutations, just not the particular mutation we are thinking about at the moment.

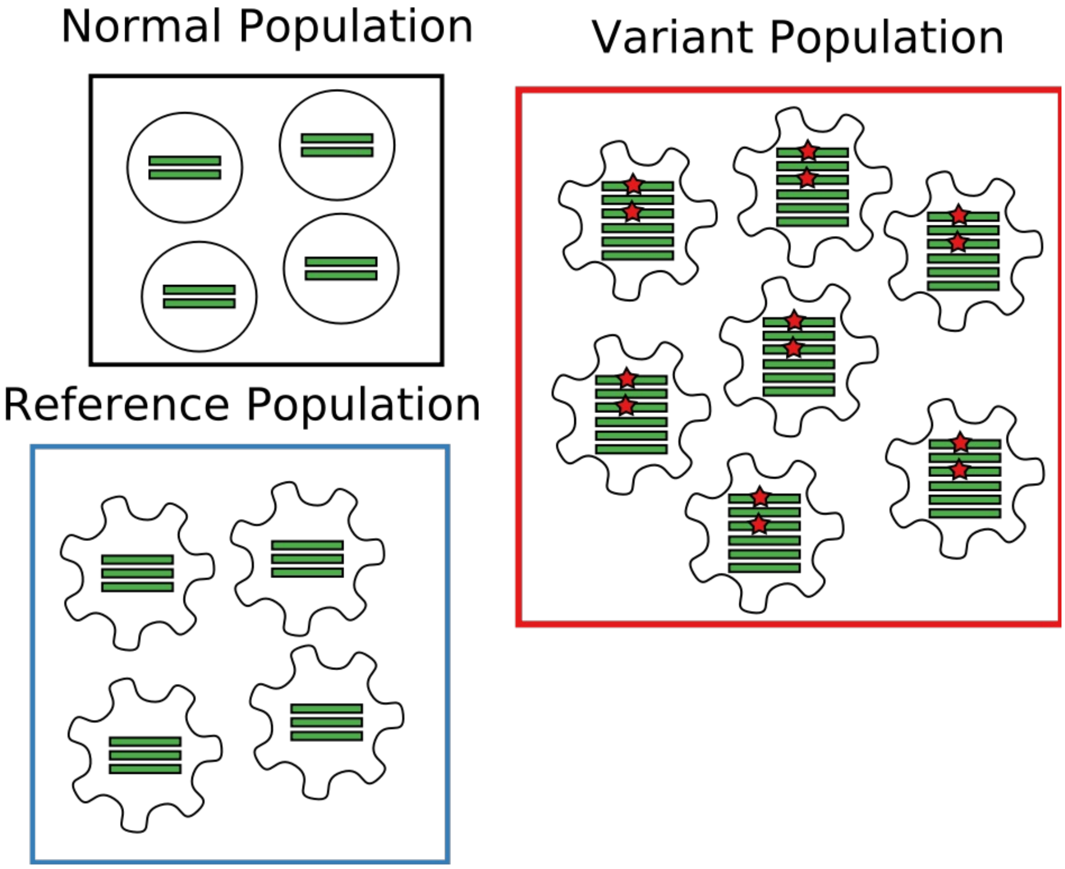

Figure 5. Illustration of the simplified population structure. In contrast to Figure 5 the mutational genotypes of all cells within the reference and variant populations are the same. Note the genotypes are different between the populations, as they must be since one population has the mutation and the other does not.

The point of the population decomposition is that the quantity of interest, the cellular prevalence of a mutation, is the proportion of cancer cells in the variant population. We are nearly ready to write down the probabilistic model but there is one issue. In Figure 4 we assume the genotypes of the malignant cells can be highly variable. While realistic, this makes it very hard to write down the probability of sampling a variant allele.

To avoid this issue we make a simplifying assumption which is that the mutational genotype of all cells within the variant population is the same. We make the same assumption for the reference population as well. Figure 5 illustrates this assumption. We emphasise this is an assumption that is likely violated in many cases. Ideally we would not need it, but the resulting model would be very complex and statistically *unidentifiable*. One way we will try to mitigate the impact of this assumption later when introduce multiple samples, is to allow the mutational genotypes to differ between samples. Roughly speaking this is then saying we only care about the mutational genotype of the dominant clone in the sample.

1.2.3. Modelling counts

With our simplified decomposition of populations and assumption we can now write down the probability of observing a variant allele. To do this we need some notation. Let

- ϕ be the cellular prevalence of the mutation
- t be the proportion of cancer cells in the sample (tumour content)
- g_N be the genotype of the normal population
- g_R be the genotype of the reference population
- g_V be the population of the variant population
- $\boldsymbol{\psi} = (g_N, g_R, g_V)$ be the vector of genotypes for notational convenience
- $c(g)$ be the copy number of genotype g

- $\mu(g)$ be the proportion of chromosomes with the mutation for genotype g. We actually make a small modification to accommodate sequence error and let $\mu(g) = \varepsilon$ when there are no mutations and $\mu(g) = 1 - \varepsilon$ when all the chromosomes are mutated.

Now we imagine we know the value of all these quantities. We then ask what is the probability of sampling a read with the mutation? We imagine that we have an infinite pool of cells which have all been lysed (cells broken apart) so that we have an infinite mixture DNA fragments. Now the probability of sampling a DNA fragment of cell depends of two things: first how common that type of cell was, second how many copies of the locus the cell has. We can easily derive these from the quantities we have defined. The probability of sampling a read from the

- normal population is proportional to $(1-t)c(g_N)$
- reference population is proportional to $t\,(1-\phi)\,c(g_R)$
- variant population is proportional to $t\phi c(g_V)$

Remark 1. It maybe surprising that we consider how many copies of the locus a cell has when determining the probability of sampling a read from that cell. To see why this necessary consider the following thought experiment. What would happen if one population had an infinite number of copies of the locus? Then it would contribute an infinite number of DNA fragments, and we would only sample these.

Remark 2. One can derive a slightly different model if assume cells are first selected before the lysis step. In this case the number of copies of the locus a cell has does not impact the probability of selecting a read from the cell. The complication is that we then need to model how we select which chromosomes from the cell get sequenced. It is simple if we pick one at random, but harder if multiple chromosomes can be sequenced.

Note we need to normalise the probabilities to sum to one. Formally, let $E_i \in \{N, R, V\}$ be a random variable indicating that read i was sample from population e. Then

$$\begin{aligned} p(E_i = e) &= \begin{cases} \frac{(1-t)c(g_N)}{Z} & \text{if } \; e = N \\ \frac{t\,(1-\phi)\,c(g_R)}{Z} & \text{if } \; e = R \\ \frac{t\phi c(g_V)}{Z} & \text{if } \; e = V \end{cases} \\ Z &= (1-t)c(g_N) + t\,(1-\phi)\,c(g_R) + t\phi c(g_V) \end{aligned}$$

Once we know which population a read comes from, then it is straightforward to compute the probability that the read has the mutation. This is simply given by $\mu(g_e)$ if $E = e$. To formalise this let F_i be a random variable indicating if read i is has the mutation. With this notation we have that

$$p(F_i = 1 | E_i = E) = \mu(g_e)$$

hence

$$\begin{aligned} p(F_i = 1) &= \sum_{a \in \{N,R,V\}} p(F_i = 1 | A_i = a) p(A_i = a) \\ &= \frac{(1-t)c(g_N)}{Z}\,\mu(g_N) + \frac{t\,(1-\phi)\,c(g_R)}{Z}\,\mu(g_R) + \frac{t\phi c(g_V)}{Z}\,\mu(g_V) \\ &:= \xi(\boldsymbol{\psi}, \phi, t) \end{aligned}$$

This implies that

$$F_i | \boldsymbol{\psi}, \phi, t \sim \text{Bernoulli}(\cdot | \xi(\boldsymbol{\psi}, \phi, t))$$

Of course we observe more than one read, and the total number of reads with a variant is $B = \sum_{i=1}^{d} F_i$. There is a basic result in probability that says the sum of Bernoulli variables follows a Binomial distribution, so we have

$$B | \boldsymbol{\psi}, \phi, t, d \sim \text{Binomial}(\cdot | d, \xi(\boldsymbol{\psi}, \phi, t))$$

This is sufficient to define the likelihood for the basic model. The only thing left is to specify a prior for ϕ. We know that $\phi \in [0, 1]$ but very little else for an arbitrary mutation. Thus we will use a Uniform continuous prior. So our simple Bayesian model is then

$$\begin{aligned} \phi &\sim \text{Uniform}(\cdot|[0,1]) \\ B|\boldsymbol{\psi},\phi,t,d &\sim \text{Binomial}(\cdot|d,\xi(\boldsymbol{\psi},\phi,t)) \end{aligned}$$

We can then write down the joint distribution

$$\begin{aligned} p(B=b,\boldsymbol{\psi},\phi,t,d) &= p(B=b|\boldsymbol{\psi},\phi,t,d)\, p(\phi) \\ &= \binom{d}{b}\xi(\boldsymbol{\psi},\phi,t)^b(1-\xi(\boldsymbol{\psi},\phi,t))^{d-b}\,\mathbb{I}(\phi\in[0,1]) \end{aligned}$$

and applying Bayes' rule we can get the posterior

$$p(\phi|=b,\boldsymbol{\psi},t,d) = \frac{p(B=b,\boldsymbol{\psi},\phi,t,d)}{\int p(B=b,\boldsymbol{\psi},\phi,t,d)\,\mathrm{d}\phi}$$

Now the integral in the bottom does not have a closed form, but it is a one dimensional integral and so is trivial to compute numerically. Thus we can now compute the posterior distribution for the cellular prevalence of a mutation. There is still some work to do, in particular we have assumed we know the values of the population genotypes $\boldsymbol{\psi}$ and the tumour content t. We can often get a reasonable estimate of t from other sources such as copy number analysis. However, we are unlikely to have any idea about $\boldsymbol{\psi}$. The genotype of the normal population is not an issue, but the other two populations are a mystery. In the next section we will discuss how to address this issue.

1.2.4. Genotype priors

In the previous section we saw that given the mutational genotypes of the populations we could infer the cellular prevalence of the mutation. In practice we do not have this information. In the Bayesian framework this issue appears frequently. We often imagine a model where we have more information then we can observe in the form of unobserved or *latent* variables. The solution to this problem is quite simple, we place a prior distribution over the variables. Once we have done this we can either infer them or *marginalise* them. It is generally preferable to marginalise the variables if possible as it reduces the dimensionality of the problem. This typically leads to faster inference algorithms as the space to be explored is smaller.

The question we face is how to specify a prior for the genotypes $\boldsymbol{\psi}$? One solution is to place a uniform prior over all possible genotypes. However, we would face two difficulties. First, our posteriors would be highly uninformative. Since we assume nothing about the genotypes, essentially any cellular prevalence would be equally likely regardless of the data. The second issue is that this is computationally infeasible as we would need to sum over an infinite set.

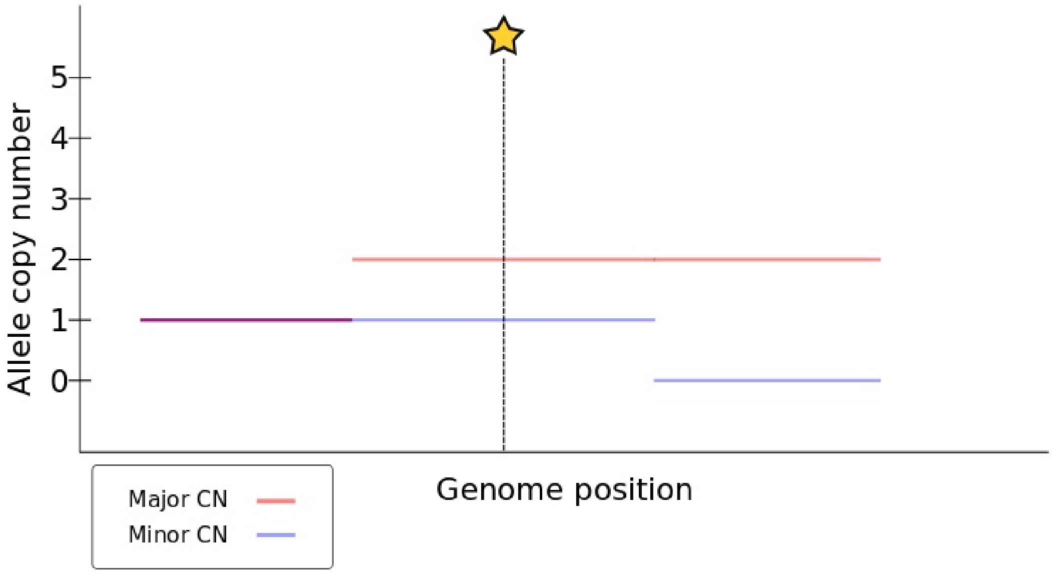

Figure 6. Illustration of allele specific copy number profile. The red line is the major copy number and the blue line is the minor copy number. The star indicates the location of an SNV. The major copy number of this SNV is 2 and the minor is 1.

To address this issue we will make use of some auxiliary information that we typically have, namely the copy number of the locus. We will assume that *allele specific* copy number profiles are available for the samples. These can be generated from micro-array, whole genome sequencing or exome sequencing. In the next module we will discuss the details of how these are actually generated. Figure 6 provides a schematic example of the information. Here we have an SNV and the copy number profile. We can see from this figure that the major copy number of the SNV is 2 and the minor copy is 1.

Remark 3. We use the terminology major/minor copy number to refer to the parental chromosome with more/less copies.

Remark 4. Many tools for copy number profiling of cancers provide sub-clonal copy number predictions. We ignore this complication, and simply take the profile of the most prevalent copy number clone.

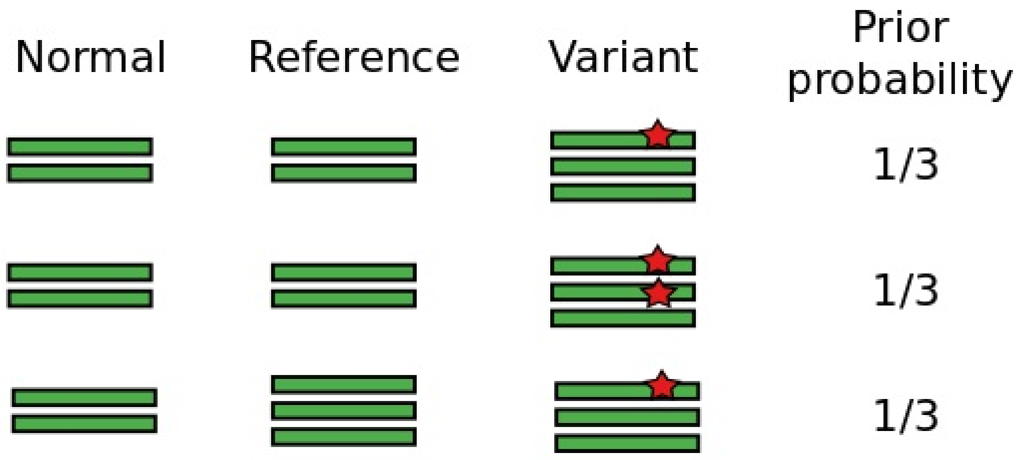

Figure 7. Schematic of how to illicit mutational genotype priors. We assume we have the information from Figure 6 and know the major copy number is 2 and the minor copy number is 1. The first two examples correspond to mutations which happen prior to the copy number change, hence the total copy number of the reference and variant population differ. The third example corresponds to the case where the mutation occurs after the copy number event. Hence, the copy number of the reference and variant population are the same. Furthermore, only a single copy can be mutated by the infinite sites assumption.

Now, the question is how to use this copy number information to develop a prior for the mutational genotypes of the populations with respect to an SNV? There are many ways to do this, and it is really a matter of personal belief. The model we will adopt breaks into two cases.

1. If the mutation occurs before the copy number event, then we need to consider all possible mutational genotypes for the variant population with mutations on one to the major copy number of the chromosomes. Since the copy number event does not affect the reference population, the total copy number of the reference population will be the same as the normal population.

2. If the mutation occurs after the copy number event, then only a single chromosome can be mutated. This follows from the infinite sites assumption. We also have that the total copy number of the reference population matches the variant population.

Applying these two rules we can list a set of possible mutational genotypes compatible with the observed copy number profile. We also need a way to assign probabilities to each of these scenarios. In the absence of any reason to prefer one over the other, we simply weight them equally.

1.2.5. The full independent model

Now we have a way to illicit priors for the mutational genotypes, we will incorporate this information into the model. Let $\boldsymbol{\pi}$ be a vector of probabilities for each mutational genotype. We will use the notation $\pi_{\boldsymbol{\psi}}$ to indicate the prior probability of the mutational genotype $\boldsymbol{\psi}$. We can then compute the probability of the observed data as follows

$$\begin{aligned} p(B=b,\boldsymbol{\pi},\phi,t,d) &= p(\phi)\sum_{\boldsymbol{\psi}} p(\boldsymbol{\psi}|\boldsymbol{\pi})p(B=b|\boldsymbol{\psi},\phi,t,d) \\ &= p(\phi)\sum_{\boldsymbol{\psi}} \pi_{\boldsymbol{\psi}} p(B=b|\boldsymbol{\psi},\phi,t,d) \end{aligned}$$

Here we have applied the law of total probability to sum over all possible genotypes we believe possible. Also note that $p(B=b|\boldsymbol{\psi},\phi,t,d)$ is just the conditional distribution we defined earlier. Thus our full model is now

$$\begin{aligned} \phi &\sim \text{Uniform}(\cdot|0,1) \\ B|\boldsymbol{\pi},\phi,t,d &\sim \sum_{\boldsymbol{\psi}} \pi_{\boldsymbol{\psi}} \text{Binomial}(\cdot|d,\xi(\boldsymbol{\psi},\phi,t)) \end{aligned}$$

where the last line is a mixture distribution over the mutational genotypes.

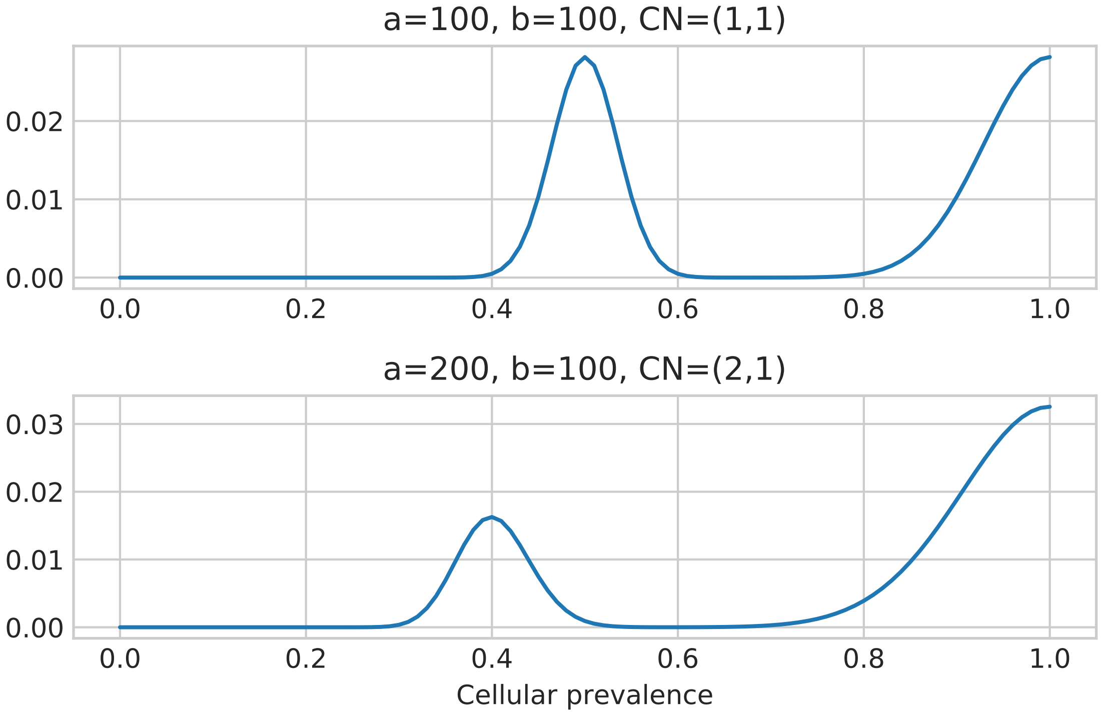

Figure 8. Example posterior densities for the cellular prevalence ϕ computed from the model. The top row shows the case for a homozygous diploid position (there is a s typo it should be CN=(2,0)). The second row shows the posterior for the case illustrated in Figures 6 and 7.

We are now at a point where we can compute the posterior of ϕ by applying Bayes' rule. We cannot analytically compute the integral to get the normalisation constants, so we use numerical approximations. The result of doing this is illustrated in Figure 8. One thing to note is that the posterior distribution is multi-modal. This corresponds to our uncertainty about the true genotype. For example in the first row of Figure 8 the VAF is 0.5 and the mode at 0.5 corresponds to the case where both copies are mutated, while the second mode at 1.0 corresponds to the case where a single copy is mutated.

To recap, we have developed a model that allows us to infer the cellular prevalence of a mutation. We assume we observe allelic count data for an SNV (a, b), have an estimate of the tumour content (t) and know the copy number profile overlapping the SNV to derive the genotype prior $(\boldsymbol{\pi})$. This model treats all mutations as completely independent. A weakness of this approach is that the posteriors tend to be multi-modal, so we are still quite uncertain about the value of ϕ. In the next section we will discuss how to fix this problem.

1.3. Clustering mutations

In the previous section we systematically developed a model to infer the cellular prevalence of a single mutation. Using this model we can easily compute the posterior distribution for the cellular prevalence ϕ of any mutation. However, these posteriors tend to be multi-modal because we have to consider a large number of mutational genotypes. We will now look at how clustering mutations can help solve this problem.

1.3.1. Motivation

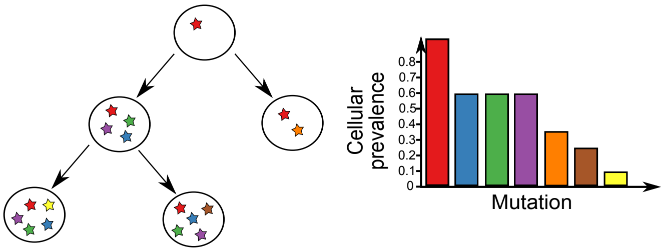

Figure 9. Illustration of the relationship between evolutionary history and cellular prevalence. On the left we have hypothetical evolutionary history, where stars indicate mutations and nodes clonal populations. On the write is a hypothetical set of cellular prevalence for the mutations.

Cancer is an evolutionary process and clonal populations are related by a phylogenetic tree. One important implication of this is that mutations which share the same evolutionary history will be at the same cellular prevalence. We illustrate this in Figure 9 where the phylogeny is shown on the left and the cellular prevalence of mutations on the right. The most interesting mutations are the green, blue and purple which all appear at the some point in the phylogeny. Assuming mutations are propagated to descendants (never lost) then these mutations will always appear in the same set of cells. Hence, their cellular prevalence will be identical. This has two important implication:

1. We should expect sets of mutations to have the same cellular.
2. If we can identify which mutations have the same cellular prevalence we can infer which ones share the same evolutionary history.

The first point tells us that we should not treat mutations independently in our model. It also means that we can share statistical strength between mutations and potentially reduce the uncertainty in our posterior distributions for ϕ. The problem we face is that we do not know which mutations belong together or what their cellular prevalence are. We also do not know how many clones are in the sample. We will see how to address this in the remainder of this module.

Remark 5. Lost mutations are not actually a problem. If mutations originate at the same point and are lost at the same point they will form a cluster of there own. This will lead to the inference of an additional clone, but not impact the cellular prevalence estimates.

If we want to reconstruct the phylogeny based on the assumption mutations at higher prevalence are further up the tree, then we have a problem. We discuss this in module 4.

1.3.2. Mixture models

Before diving into the full model, we first review mixture models. Mixture models are a very useful type of probabilistic model which posit that datapoints originate from groups or clusters. Datapoints from the same cluster share the same parameters, and thus are similar to each other in some sense. For now we assume the number of clusters, K, is known in advance and fixed. Then a standard way to construct a Bayesian mixture model is as follows.

$$\begin{aligned}
\boldsymbol{\kappa} &\in \mathbb{R}_{+}^{K} \\
\boldsymbol{\rho}|\boldsymbol{\kappa} &\sim \text{Dirichlet}(\cdot|\boldsymbol{\kappa}) \\
z_n|\boldsymbol{\rho} &\sim \text{Categorical}(\cdot|\boldsymbol{\rho}) \\
\theta_k &\sim G(\cdot) \\
x_n|\boldsymbol{\theta}, z_n &\sim F(\cdot|\theta_{z_n})
\end{aligned}$$

This model associates a latent variable, z_n, with each data point. The variable z_n takes values in the set $\{1, ..., K\}$, and acts as an indicator for which cluster a data point originates from. Each cluster has an associated parameter θ_k sampled independently from a distribution G. The observed data X_n is then generated from a distribution F with parameter θ_{z_n}. Thus whenever $z_i = z_j = k$ for data points i and j they are in the same cluster and have been generated from the same distribution $F(\theta_k)$.

1.3.3. Sharing statistical strength

We will now improve the previous model by turning it into a mixture model. For now we assume the number of clusters K is known and fixed. Then the updated model is

$$\begin{aligned}
\boldsymbol{\rho}|\boldsymbol{\kappa} &\sim \text{Dirichlet}(\cdot|\boldsymbol{\kappa}) \\
z_n|\boldsymbol{\rho} &\sim \text{Categorical}(\cdot|\boldsymbol{\rho}) \\
\phi_k &\sim \text{Uniform}(\cdot|[0,1]) \\
b_n|\boldsymbol{\pi}_n, \boldsymbol{\phi}, t, d_n, z_n &\sim \sum_{\boldsymbol{\psi}} \pi_{n\boldsymbol{\psi}} \text{Binomial}(\cdot|d_n, \xi(\boldsymbol{\psi}, \phi_{z_n}, t))
\end{aligned}$$

where we introduce the index n for data points. In this model data points are no longer independent, but will share the same cellular prevalence when they belong to the same cluster.

This example nicely illustrates the modularity of Bayesian probabilistic models. Specifically, we were able to reuse the previous model for mutational genotypes and embed it in a more complex model. This is a useful strategy in general. Begin by breaking down the problem into simpler subproblems, and then progressively extend the model.

1.3.4. MCMC inference

To fit the new model to the data, we can no longer appeal to numerical methods to compute the posterior because we have many more parameters, some of which are discrete. To address this we will use MCMC to approximate the posterior.

Remark 6. We could quite easily use the expectation maximisation (EM) algorithm to compute the MAP estimate for this model rather than MCMC. For finite mixture models this will often be faster. The main disadvantage is that we will not have a full posterior, but point estimates as discussed in module 1. Thus we cannot quantify uncertainty. Once we move to using a Dirichlet process prior, EM will no longer be applicable so we will have to use MCMC.

Before we work out the details we will write down the joint distribution.

$$p(\boldsymbol{b}, \boldsymbol{d}, \boldsymbol{\pi}, \boldsymbol{\phi}, \boldsymbol{\rho}, \boldsymbol{\kappa}, t, \boldsymbol{z}) = p(\boldsymbol{\rho}|\boldsymbol{\kappa}) \prod_{k=1}^{K} p(\phi_k) \prod_{n=1}^{N} p(b_n|\boldsymbol{\pi}_n, \boldsymbol{\phi}, t, d_n, z_n)\, p(z_n|\boldsymbol{\rho})$$

We will update the model parameters in blocks. This is typically done in MCMC methods, as designing good updates for all the parameters simultaneously is usually hard (see module 1). We will use a combination of Metropolis-Hastings (MH) and Gibbs sampling. The updates we will use are:

- $\boldsymbol{\rho}$ we will use a Gibbs update
- ϕ_k we will use an MH update with a Uniform$(\cdot|[0,1])$ proposal
- z_n we will use a Gibbs update

To implement the Gibbs step for z_n we need to compute the conditional distribution $p(z_n|-)$, where $-$ indicates all the other variables. We introduce the notation $\boldsymbol{z}^{(-n)} = (z_1, \ldots, z_{n-1}, z_{n+1}, \ldots, z_N)$ which is the vector of all cluster indicator variables except the n^{th} one. The conditional distribution is then

$$\begin{aligned} p(z_n = k|-) &= \frac{p(\boldsymbol{b}, \boldsymbol{d}, \boldsymbol{\pi}, \boldsymbol{\phi}, \boldsymbol{\rho}, \boldsymbol{\kappa}, t, \boldsymbol{z}^{(-n)}, z_n = k)}{p(\boldsymbol{b}, \boldsymbol{d}, \boldsymbol{\pi}, \boldsymbol{\phi}, \boldsymbol{\rho}, \boldsymbol{\kappa}, t, \boldsymbol{z}^{(-n)})} \\ &= \frac{p(\boldsymbol{b}, \boldsymbol{d}, \boldsymbol{\pi}, \boldsymbol{\phi}, \boldsymbol{\rho}, \boldsymbol{\kappa}, t, \boldsymbol{z}^{(-n)}, z_n = k)}{\sum_{\ell=1}^{K} p(\boldsymbol{b}, \boldsymbol{d}, \boldsymbol{\pi}, \boldsymbol{\phi}, \boldsymbol{\rho}, \boldsymbol{\kappa}, t, \boldsymbol{z}^{(-n)}, z_n = \ell)} \\ \text{after some cancellation} &= \frac{\rho_k\, p(b_n|\boldsymbol{\pi}_n, \boldsymbol{\phi}, t, d_n, z_n = k)}{\sum_{\ell=1}^{K} \rho_\ell\, p(b_n|\boldsymbol{\pi}_n, \boldsymbol{\phi}, t, d_n, z_n)} \end{aligned}$$

let $\bar{\rho}_{nk} = \frac{\rho_k\, p(b_n|\boldsymbol{\pi}_n, \boldsymbol{\phi}, t, d_n, z_n = k)}{\sum_{\ell=1}^{K} \rho_\ell\, p(b_n|\boldsymbol{\pi}_n, \boldsymbol{\phi}, t, d_n, z_n)}$ then

$$z_n|- \sim \text{Categorical}(\cdot|\bar{\boldsymbol{\rho}}_n)$$

Because we choose a Dirichlet prior for $\boldsymbol{\rho}$ the conditional distribution is easily obtained from conjugacy, and will be a Dirichlet distribution as well. Let $m_k = \sum_{n=1}^{N} \mathbb{I}(z_n = k)$ be the number of data points from cluster k. Let $\bar{\boldsymbol{\kappa}} = (\kappa_1 + m_1, \ldots, \kappa_k + m_k, \ldots, \kappa_K + m_K)$

$$\boldsymbol{\rho}|- \sim \text{Dirichlet}(\cdot|\bar{\boldsymbol{\kappa}})$$

1.3.5. Dirichlet process

The finite mixture model we defined previously has one major problem: We assume the number of clones K is known in advance. In practice this is not true, and we would like to infer the number of clones as part of the model. One strategy to address this problem is to use a Dirichlet process prior for the cellular prevalence.

The Dirichlet process (DP) prior is an example of a *non-parametric* Bayesian prior. Informally these means that it is a prior which can adapt model complexity as more data is observed. More formally the DP is a distribution over distribution (stochastic process). This means that the random variables we sample from a DP are distributions. There are two parameters for the DP: the concentration parameter $\alpha \in \mathbb{R}_+$ and the base measure G_0 a distribution. Roughly speaking α controls how many clusters we expect. While the distribution G_0 is used to sample the new values that the distribution exhibits.

To understand how a DP can be useful we need to take a slightly different view of mixture models. So far we have used the cluster indicator variables z_n to identify which data points belong to the same cluster. So if $z_i = z_j = k$ that means data points i and j come from a distribution with parameter θ_k, in other words belong to cluster k. We can change our viewpoint though, and instead of having one θ_k for each cluster, we can instead assign data point n its own parameter ζ_n. When we have $\zeta_i = \zeta_j$ we then say data points i and j belong to the sample cluster. Now if we sample ζ_n from a continuous distribution, then there is zero probability that two data points will ever share the same value. So we need to sample ζ_n from a discrete distribution if there is any chance for shared values. In the background this is what the mixture model is doing, it is creating the discrete distribution

$$G = \sum_{k=1}^{K} \rho_k\, \delta_{\theta_k}(\cdot)$$

where $\delta_x(y)$ is the Kronecker delta function that equals one when $x=y$ or zero otherwise. We then draw $\zeta_n \sim G(\cdot)$, so that $\zeta_n \in \{\theta_1, ..., \theta_K\}$.

We can re-write our current mixture model with this viewpoint as follows.

$$\begin{aligned}
\boldsymbol{\rho}|\boldsymbol{\kappa} &\sim \text{Dirichlet}(\cdot|\boldsymbol{\kappa}) \\
\theta_k &\sim \text{Uniform}(\cdot|[0,1]) \\
G &= \sum_{k=1}^{K} \rho_k \, \delta_{\theta_k}(\cdot) \\
\phi_n &\sim G \\
b_n|\boldsymbol{\pi}_n, \phi_n, t, d_n, z_n &\sim \sum_{\boldsymbol{\psi}} \pi_{n\boldsymbol{\psi}} \, \text{Binomial}(\cdot|d_n, \xi(\boldsymbol{\psi}, \phi_n, t))
\end{aligned}$$

Note that the cluster indicators are no longer present and each data point no samples its own cellular prevalence ϕ_n. However, the values must belong to the set $\{\theta_1, ..., \theta_K\}$ so we have clustering. Here the Uniform$(\cdot|[0,1])$ is used to identify the set of values G takes on. The elements of this set are often referred to as the atoms of the distribution.

The previous discussion may seem like a very confusing way to define a mixture model. The reason that it is useful is that if sample a distribution G from a DP it will be discrete. More explicitly, any distribution G sample from a DP has the following form

$$G = \sum_{k=1}^{\infty} \rho_k \, \delta_{\theta_k}(\cdot)$$

Thus we can use this distribution G in our mixture model, just the same way as the finite case. The updated version of our model now takes the following form.

$$\begin{aligned}
G_0 &= \text{Uniform}(\cdot|[0,1]) \\
G|\alpha, G_0 &\sim \text{DP}(\cdot|\alpha, G_0) \\
\phi_n &\sim G \\
b_n|\boldsymbol{\pi}_n, \phi_n, t, d_n, z_n &\sim \sum_{\boldsymbol{\psi}} \pi_{n\boldsymbol{\psi}} \, \text{Binomial}(\cdot|d_n, \xi(\boldsymbol{\psi}, \phi_n, t))
\end{aligned}$$

1.3.6. Chinese restaurant process

Before discussin how to fit the model we will take a brief digression to discuss the Chinese restaurant process (CRP). The CRP is the marginal distribution of the DP when we integrate out the distribution G. Formally, the CRP is a probability distribution on partitions of the integers. Let $[n] = \{1, ..., n\}$ be the set of all positive integers up to n. Then a partition of n is a set $c_n = \{b: b \subset [n]\}$ such that $b \cap b' = \varnothing$ and $\bigcup_{b \in c_n} b = [n]$. That is a set of disjoint sets whose union equals $[n]$. With some thought you can see this is equivalent to a clustering of the data points labelled from 1 to n. The probability mass function (pmf) of the CRP is given by

$$p(c_N|\alpha) = \frac{\Gamma(\alpha)}{\Gamma(\alpha+N)} \alpha^{|c|} \prod_{b \in c_N} (|b|-1)!$$

where α is the concentration parameter.

The CRP is often described by an analogy to customers entering a chinese restaurant, hence the name. The description goes as follows. The first customer enters the restauran and sits down at a table. The second customer then enters the restaurant makes a choice. The can either join the first customer with probability $\frac{1}{1+\alpha}$ or start a new table with probability $\frac{\alpha}{1+\alpha}$. As new customers enter they can choose to set at an existing table with probability proportional to the number of customers already there, or they can start a new table with probability proportional to α. The distribution over seatings of customers is then given by the CRP pmf.

There are a few interesting aspects of this process. First, it has a rich get richer property where new customers are more likely to join existing tables. Second, the process is *exchangeable* so the order customers enter the restaurant does not affect the distribution. This can be seen directly from the pmf of the CRP. This feature is particularly useful, as it will allow to develop a Gibbs sampler for the DP in the next section.

1.3.7. Full model inference

We can still use MH updates for ϕ_n. The new problem is to update ϕ_n. There are two approaches to performing inferences. One is to make use of the *stick breaking representation* of the DP. Using this approach we can sample the distribution G directly. Once we have this distribution, then we can perform Gibbs updates just like the finite mixture model case. The second approach which we will follow is marginalise G and in which case we have a CRP.

The key insight is that we can use exchangeability to treat the data point we want to update as the last data point added. We will re-introduce the cluster labels for this step, let $z_n \in \{1, ..., K\}$ be the cluster label for the n^{th} data point. Here K is the number of clusters with data points, after removing the n^{th} data point. The the probability of this data point joining an existing cluster depends on the size of the cluster, excluding the data point of interest. We will use $m_k^{(-n)} = \sum_n \mathbb{I}(z_n = k)$ to denote these cluster sizes. The conditional probability of z_n given the other indicator variables, $\{z_i\}_{i \neq n}$ is given by the following equation.

$$p(z_n = k | \{z_i\}_{i \neq n}) = \begin{cases} \frac{m_k^{(-n)}}{n-1+\alpha} & \text{if } k \in \{1, ..., K\} \\ \frac{\alpha}{n-1+\alpha} & \text{if } k = K+1 \end{cases}$$

Remark 7. This update is fairly easy to understand. The only subtlety when we are updating a data point that is in a singleton cluster (only member of the cluster). In this case K differs when remove the data point from the clustering.

This provides the conditional probability for the cluster label, but we will actually need a conditional probability of the form $p(z_n | \{z_i\}_{i \neq n}, \{\theta_k\}_{k=1}^K, X)$ since there is a cluster parameter associated with each cluster and this is used to generate the data point.

$$\begin{aligned} p(z_n = k | \{z_i\}_{i \neq n}, \theta, X) &= \frac{p(X | z_n = k, \{z_i\}_{i \neq n}, \theta) p(z_n = k | \{z_i\}_{i \neq n})}{p(X | \{z_i\}_{i \neq n}, \theta)} \\ &\propto p(x_n | z_n = k, \theta) \, p(z_n = k | \{z_i\}_{i \neq n}) \end{aligned}$$

Here $p(x_n | z_n = k, \theta)$ would just be the data likelihood for data point n when it has parameter θ_k. One tricky issue is that we do not have parameters for the new clusters, so we cannot easily evaluate this. One common approach is to integrate out the model parameters. Unfortunately this only works if the likelihood is conjugate to the prior, which is not the case for us. There are few ways around this. The one we will employ is to draw l values of θ from the prior to create l possible empty clusters to join. There is one last subtle point. When n was originally part of the singleton cluster we need to use that value as one of the l values of θ, so we only sample $l-1$ new values from the prior. The probability of joining these clusters is then modified and becomes $\frac{\frac{\alpha}{l}}{n-1+\alpha}$. The number of empty table paramters l is a tuning paramter for the algorithm. In practice we typically take $l = 2$. The Gibbs update is then given by the following formula.

$$p(z_n = k | \{z_i\}_{i \neq n}, \theta, X) = \begin{cases} \frac{m_k^{(-n)}}{n-1+\alpha} p(b_n | \boldsymbol{\pi}_n, \phi_k, t, d_n) & \text{if } k \in \{1, ..., K\} \\ \frac{\frac{\alpha}{l}}{n-1+\alpha} p(b_n | \boldsymbol{\pi}_n, \phi_k, t, d_n) & \text{if } k = K+1 \end{cases}$$

Remark 8. The trick for sampling parameters for the new clusters may seem a bit obscure. For more details see [cite Neal] for this (algorithm 8 from the paper) and other possible ways to peform inference for DPs.

It is also useful to use an MH step to update the values ϕ_k between the updates of z_n. This will allow us to move from values sampled from the prior, to those that are a close fit to the data. Without this MH move we would only update the cellular prevalences when we sample new tables. This is extremely slow we do not use any information about the data to generate the new ϕ values.

1.3.8. Computing consensus clustering

Thus far we have defined the model and come up with a strategy to fit the model. This involves running an MCMC algorithm for many iterations and collecting samples. At each iteration we will record the cluster paramters (cellular prevalences) and the cluster indicators (which cluster a datapoint was assigned to). Each iteration of the MCMC will have a different clustering of the data. The question is then how pick a best clustering?

One simple approach would be to take the sample with the highest joint probability i.e. the MAP estimator. This is sub-optimal as we ignore all the other samples from the MCMC chain. A better approach is to use a loss function. We are free to choose any loss function, but there are some challenges we need to consider. The biggest issue is that we can permute the labels of the cluster indicators and the cluster parameters and the likelihood is the same. So though a data point may move from cluster k to ℓ, nothing has changed because we also changing the cluster paramters from θ_k to θ_ℓ. Thus we ideally want a loss function that is invariant to label permutations. The other issue is that the number of clusters varies between iterations.

Here we will use the adjusted rand index (ARI) as the loss function. This is a measure of clustering similarity. We will then seek the clustering which minimises this lost under our approximate posterior. The details on how to do this can be found in [cite mpear]. To implement this proceedure we compute the pair-wise similarity matrix of two data points. This is the proportion of MCMC samples in which that data points belong to the same cluster. A nice feature of this summary of the MCMC trace is that it does not depend on the number of clusters or the actual labels of the data points. To optimise the ARI we then build a dendrogram from the similarity matrix and find the cut level which maximises the MPEAR score defined in [cite mpear]. This yields our estimated clustering of the data $\hat{\boldsymbol{z}} = (\hat{z}_1, ..., \hat{z}_N)$.

We would also like to obtain posterior distributions for the cellular prevalence. Once approach is to look at the full MCMC trace of the cellular prevalence associated with a mutation. We can plot these values as histogram or using a density estimator. We can also report the mean and variance of this distribution. One downside of this approach is we report a separate posterior for each mutation, which ignores our estimate clustering $\hat{\boldsymbol{z}}$. If we want to get the posterior cellular prevalence give the clustering we can use a slightly ad-hoc approach and compute the conditional posterior given $\hat{\boldsymbol{z}}$. Suppose we want the conditional posterior for cluster k, that is the posterior of ϕ_k. Then we need to computer the following quantity.

$$p(\phi_k|\hat{\boldsymbol{z}}, X) \;=\; \frac{p(\phi_k)\prod_{\{n:z_n=k\}} p(x_n|\phi_k)}{\int p(\phi_k)\prod_{\{n:z_n=k\}} p(x_n|\phi_k)\,\mathrm{d}\phi_k}$$

The integral in the denomintor does not have a closed form. But it is a one dimensional integral so we can use numerical methods to compute it.

1.3.9. Multiple samples

Thus far we have only considered data from a single sample. If multiple samples are available it would be useful to model them jointly so that clusters are coherently defined across samples. Multiple samples are very useful for this problem, as they let us identify clusters which may be missed from single sample analysis. One reason this could happen is that some clones are only present in a subset of samples. Another reason is that we may have two clones at similar prevalence in one sample that would be impossible to identify. If we have another sample where there prevalence is quite different, then we have a better chance of finding both clones.

The changes required to include multiple samples are actually fairly simple. We will need some notation for this. Let m be the index over samples, M be the number of samples, b_{nm} indicate the number of reads with variant n in sample m, d_{nm} be the corresponding read depth, $\boldsymbol{\pi}_{nm}$ be the genotype prior for mutation n in sample m and t_m be the tumour content of the m^{th} sample. Then the new model becomes

$$\begin{aligned}
G_0 &= \text{Uniform}(\cdot|[0,1]^M) \\
G|\alpha, G_0 &\sim \text{DP}(\cdot|\alpha, G_0) \\
\boldsymbol{\phi}_n &\sim G \\
b_{nm}|\boldsymbol{\pi}_{nm}, \phi_n, t_m, d_{nm}, z_n &\sim \sum_{\boldsymbol{\psi}} \pi_{nm\boldsymbol{\psi}} \text{Binomial}(\cdot|d_{nm}, \xi(\boldsymbol{\psi}, \phi_{nm}, t_m))
\end{aligned}$$

The main change is that we use Uniform prior over the M dimensional unit cube and we sample a vector of cellular prevalence $\boldsymbol{\phi}_n = (\phi_{n1}, ..., \phi_{nm})$ for each mutation. The inference procedure is largely the same. The only major differences is that we will now need to compute summaries of cellular prevalence for each sample.

We see again the modularity of the Bayesian modelling approach. It is a fairly trivial exercise to extend our simpler model to the more complex case. The most tedious part is updating to a decent notation.

1.3.10. Overdispersion

One issue with current model is that read counts often except more variability than the Binomial can model. The issue is referred to as overdispersion. At the depths commonly used for WGS (30x-100x) this problem is usually not obvious. In higher coverage data such as targeted sequencing (10^3x-10^5x) it becomes more pronounced.

The solution is to use a distribution with more parameters and more flexibility. In the case of the Binomial, a common choice is to use the Beta-Binomial distribution which is overdispersed relative to the binomial. The Beta-Binomial has to parameters a and b like a Beta distribution. We can reparameterise the Beta-Binomial in terms of mean and variance as well. Using this approach we can set the mean to ξ and fit the variance parameter. The procedure of substituting a more flexible distribution with the same mean is a common approach.

1.4. Discussion

In this module we saw how we to construct a Bayesian probabilistic model for inferring clonal population structure from bulk data. We started with a simple model to correct for mutational genotype and normal contamination. We then considered a more complex mixture model, ultimately using a Dirichlet process to infer the number of clones. We discussed how to fit the model using MCMC and summarise the resulting posterior approximation.

This module illustrates the basic technique and steps needed to construct a probabilsitic model and fit it. The most important concepts are:

1. Clearly define the problem
2. Start by solving simpler sub-problems
3. Iteratively extend the model
4. Identify a suitable method for fitting the model
5. If using MCMC, identify a way to report summaries of the posterior