1. Copy number variation

In the last module we looked at probabilistic models for SNV data. In this module we will switch gears and consider copy number variants (CNVs). The main difference between the two data types from a modelling perspective is spatial correlation. If we look at total read counts between adjacent positions in the genome they will be strongly correlated. This requires us to consider models which can account for the spatial correlation. In contrast, it fairly reasonable to model SNVs as independent observations.

In this module we will explore how to model CNVs. We start by discussing some data representations. Next we will construct a simple model for inferring total copy number (TCN) from homogeneous cell populations. We will then look at some of the practical limitations of this model and how they can be addressed.

1.1. Data representation

Again we will assume we have aligned data from a high throughput sequencing experiment. When dealing with CNVs there are a few different ways to summarise the aligned read data for modelling purposes. The obvious one is to use allele counts in the same way as SNV data. The main downside of this approach is that we will typically be interested in every position in the genome, whereas we usually consider far fewer positions when dealing with SNVs. This can be computationally expensive, and make model fitting impractically slow. One solution is to restrict the set of positions we consider. A common choice is to look at heterozygous SNPs identified from matched normal tissue. This reduces the number of positions from billions to millions, which is significantly more tractable. The other benefit of heterozygous SNPs is that they can be used infer allele specific copy number as we will discuss later.

An alternative way to represent the data is to divide the genome into bins, typically of equal size though this is not essential. We can then count the total number of reads within the bin and use this as our data. This approach is attractive computationally as we typically use bins with lengths on the order of $10^3 - 10^6$ which means we will have between $10^3 - 10^6$ data points. Beyond the computational savings, binning the data can also be useful when the sequencing coverage is low. This has become a more relevant consideration in recent years with the development of single cell whole genome sequencing platforms. Single cells are often sequenced at low coverage (typically 0.001x-0.1x) as opposed to the higher coverage of bulk (typically 30x-100x). The main reason to do this is cost, though technical issues such as library diversity also factor in. The most delicate issue in this case is choosing an appropriate bin size.

A variation on the binning strategy is to use bins defined by haplotype blocks. A haplotype block is segment of the genome where we are able to *phase* heterozygous SNPs. The point of using these blocks is to combine the read counts from a larger set of heterozygous SNPs in order to improve allele specific copy number inference. For each block we then report three values: the total read count, the number of reads supporting the a allele at heterozygous positions, and the number of reads supporting the b allele at heterozygous positions.

1.2. Modelling spatial correlation

Regardless of the data representation, we need to consider spatial correlation when modelling CNV data. The rational is that copy number changes typically affect large regions of the genome that will span many data points. Thus it more likely the copy number of adjacent points is the same. There are a few commonly used strategies to deal with this problem.

The simplest strategy is to *segment*, that is identify regions in the same state, before performing any other analysis. This can be readily done by standard tools such as circular binary segmentation (CBS). Once the data has been segmented we can ignore the spatial correlation and treat the data as independent. This is the approach used by tools such as ASCAT and the battenberg algorithm. This makes the modelling task easier and generally leads to faster methods. The downside is that we cannot leverage additional information from the model during the segmentation step. This in turn means that we cannot improve the segmentation as we learn other features of the data such as the tumour content.

The other major approach is to use probabilistic models with spatial correlation. By far the most common approach in this category is to use hidden Markov models (HMM). Alternative models such as Kalman filters and more general state space models are not widely used in cancer genomics. The appeal of HMMs is that they are relatively simple but still able to capture some notion of spatial correlation. The key assumption is that a data point only depends on the point immediately preceding it. This is the Markov assumption of the model and is important for developing efficient inference algorithms. The main downside of HMMs is that they have a fairly limited ability to model *state duration*. Informally this means that very long segments tend to be discouraged. This will manifest as the inference of rapid changes from a state and back into it. Despite this, HMMs generally work well and will be the focus of the remainder of this module.

1.3. Hidden Markov models

1.3.1. Description

An HMM is a latent variable model, where the latent (or hidden) variables have a dependency structure. The latent variables take values in some discrete state space $\mathcal{X}$. The observed values depend on the latent variables in much the same way as a mixture model. In fact an alternative way to define an HMM is as a dynamic mixture model, where the probability of belonging to a cluster changes as we move along the positions. To define an HMM we need three pieces:

1. $\boldsymbol{\pi}$ - The initial state vector. This is a vector of length $|\mathcal{X}|$ where π_x is the probability the first hidden variable in the sequence takes value x.
2. A - The transition matrix. This is a $|\mathcal{X}| \times |\mathcal{X}|$ matrix where A_{ij} is the probability that the current hidden value takes on state j given the previous one was in state i.
3. F - The emission distribution. This is the distribution for the observed data which depends on the associated hidden state. The hidden state is thus like the cluster indicator in mixture models, selecting which parameter is used for F to generate the data point.

Here we assume that $K = |\mathcal{X}|$ is known and fixed. This assumption can be relaxed in much the same way as mixture models using non-parametric Bayesian priors. However, the resulting models are usually computationally demanding to fit.

Remark 1. Though non-parametric HMMs have not been widely used for studying cancer, they could solve some issues. First, like we did in SNVs we could use them to infer the number of clones. Second, we could relax the maximum copy number constraint that HMM based methods require.

The basic model for a Bayesian HMM is as follows

$$\begin{aligned}
\boldsymbol{\pi}|\boldsymbol{\kappa} &\sim \text{Dirichlet}(\cdot|\boldsymbol{\kappa}) \\
A_{k\cdot}|\boldsymbol{\gamma}_k &\sim \text{Dirichlet}(\cdot|\boldsymbol{\gamma}_k) \\
z_1|\boldsymbol{\pi} &\sim \text{Categorical}(\cdot|\boldsymbol{\pi}) \\
z_n|z_{n-1}, A &\sim \text{Categorical}(\cdot|A_{z_{n-1}\cdot}) \\
\theta_k &\sim G \\
x_n|z_n, \boldsymbol{\theta} &\sim F(\cdot|\theta_{z_n})
\end{aligned}$$

where $A_{k\cdot}$ is used to denote the k^{th} row of A, F and G are arbitrary distributions with densities f and g. We can write down the joint distribution for this model as follows.

$$\begin{aligned}
p(\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{\pi}, A, \boldsymbol{\theta}) &= \underbrace{p(\boldsymbol{\pi}) \prod_{k=1}^{K} p(\theta_k)\, p(A_{k\cdot}|\boldsymbol{\gamma}_k)}_{\text{prior}} \times \underbrace{p(x_1|z_1, \boldsymbol{\theta}) p(z_1) \prod_{n=2}^{N} p(x_n|z_n, \boldsymbol{\theta}) p(z_n|z_{n-1})}_{} \\
&= p(\boldsymbol{\pi}) \prod_{k=1}^{K} p(\theta_k)\, p(A_{k\cdot}|\boldsymbol{\gamma}_k) \times \\
&\quad \prod_{k=1}^{K} [\pi_k f(x_1|\theta_k)]^{\mathbb{I}(z_1=k)} \prod_{n=2}^{N} \prod_{\ell=1}^{K} f(x_n|\theta_\ell)^{\mathbb{I}(z_1=k)} \prod_{k=1}^{K} A_{k\ell}^{\mathbb{I}(z_{n-1}=k, z_n=\ell)}
\end{aligned}$$

1.3.2. Inference

One of the most useful features of HMMs is that we can efficiently compute the conditional distribution of the hidden states. This is done using the forward-backward (FB) algorithm, which is a dynamic programming algorithm. Applying the FB we can easily implement an expectation maximisation algorithm to estimate the MAP parameters. Once we have those parameters we can use another recursive algorithm, the Viterbi algorithm, to find the most probable sequence of hidden states.

Remark 2. The forward and backward recursions are also useful when using MCMC methods. They can be used to develop a simple Gibbs sampler that sequentially updates the hidden states. Another approach is to use the *forward filtering backward sampling* algorithm which updates the entire chain.

In what follows we will suppress the dependencies on the model parameters to keep the notation simple. We will just state the required recursion formulas here. Interested readers can find the detailed derivation in [cite bishop]. We have two recursions the forward, $\alpha(z_n)$, and the reverse, $(\beta(z_n))$.

$$\begin{aligned}
\alpha(z_n = k) &= p(x_n|z_n = k) \sum_{\ell=1}^{K} \alpha(z_{n-1} = l) A_{\ell k} \\
\beta(z_n = k) &= \sum_{\ell=1}^{K} \beta(z_{n+1} = \ell)\, p(x_{n+1}|z_{n+1} = \ell)\, A_{k\ell}
\end{aligned}$$

The main use of these recursions is to compute the following quantities

$$\begin{aligned}
p(\boldsymbol{x}) &= \sum_{k=1}^{K} \alpha(z_N = k) \\
\mathbb{E}_{\boldsymbol{z}}[\mathbb{I}(z_n = k)] &= \frac{p(z_n = k|\boldsymbol{x})}{p(\boldsymbol{x})} \\
&= \frac{\alpha(z_n = k)\beta(z_n = k)}{p(\boldsymbol{x})} \\
\mathbb{E}_{\boldsymbol{z}}[\mathbb{I}(z_{n-1} = k, z_n = \ell)] &= p(z_{n-1} = k, z_n = \ell|\boldsymbol{x}) \\
&= \frac{\alpha(z_{n-1} - k) p(x_{n-1}|z_{n-1} = k) p(x_n|z_n = \ell) \beta(z_n = \ell)}{p(\boldsymbol{x})}
\end{aligned}$$

These become useful in the EM algorithm during the maximisation step when we seek to optimise $\log p(\boldsymbol{x}, \boldsymbol{\pi}, A, \boldsymbol{\theta})$.

$$\begin{aligned}
\log p(\boldsymbol{x}, \boldsymbol{\pi}, A, \boldsymbol{\theta}) &= \mathbb{E}_{\boldsymbol{z}}[\log p(\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{\pi}, A, \boldsymbol{\theta})] \\
&= \sum_{k=1}^{K} \mathbb{E}_{\boldsymbol{z}}[\mathbb{I}(z_1 = k)] \log \pi_k + \\
&\quad \sum_{n=1}^{N} \sum_{k=1}^{K} \mathbb{E}_{\boldsymbol{z}}[\mathbb{I}(z_n = k)] \log f(x_n|\theta_k) + \\
&\quad \sum_{n=2}^{N} \sum_{k=1}^{K} \sum_{\ell=1}^{K} \mathbb{E}_{\boldsymbol{z}}[\mathbb{I}(z_{n-1} = k, z_n = \ell)] \log A_{k\ell} + \\
&\quad \log p(\boldsymbol{\pi}|\boldsymbol{\kappa}) + \sum_{k=1}^{K} \log p(A_{k\cdot}|\boldsymbol{\gamma}) + \sum_{k=1}^{K} \log p(\theta_k)
\end{aligned}$$

This optimisation takes a closed form for A and $\boldsymbol{\pi}$

$$\begin{aligned}
\hat{\pi}_k &\propto \kappa_k + \mathbb{E}_{\boldsymbol{z}}[\mathbb{I}(z_1 = k)] \\
\hat{A}_{k\ell} &\propto \gamma_{k\ell} + \sum_{k=1}^{K} \sum_{\ell=1}^{K} \mathbb{E}_{\boldsymbol{z}}[\mathbb{I}(z_{n-1} = k, z_n = \ell)]
\end{aligned}$$

Computing the MAP value of the parameters θ_k will not be a closed form in general so often numerically optimise using the gradient given by

$$\frac{\partial}{\partial \theta_k} \log p(\boldsymbol{x}, \boldsymbol{\pi}, A, \boldsymbol{\theta}) \;=\; \sum_{n=1}^{N} \mathbb{E}_{\boldsymbol{z}}[\mathbb{I}(z_n = k)] \frac{\partial}{\partial \theta_k} \log f(x_n|\theta_k) + \frac{\partial}{\partial \theta_k} \log p(\theta_k)$$

Remark 3. Thus far we consider only a single sequence. We will see in the next section that we will treat each chromosome as a separate sequence. The above equations are simple to modify to accommodate this variation.

1.4. Homogeneous sample model

We now turn our attention to model building. Here we will consider the problem of inferring copy number profiles from a sample that has no clonal population structure and no normal contamination. This model would also apply equally well to single cell data. We assume that the data is binned total read counts. Given this we will not try to infer allele specific copy number, but instead focus on total copy number. Let's introduce some notation. Let

- m index the chromosomes
- M be the number of chromosomes
- n index the bins in a chromosome
- N_m be the number of bins for chromosome m
- x_n^m denote the number of reads in the n^{th} bin on chromosome m
- z_n^m denote the hidden state for the n^{th} bin on chromosome m

1.4.1. Model description

The first step is think about an appropriate distribution to model the read counts. Given the observed data takes non-negative integer values, a simple choice is the Poisson distribution. Next we need to think about our hidden states. We would like them to encode the copy number of a bin, so we will assume they take values in the set $\{1, ..., C\}$ where C is some maximum possible copy number.

Remark 4. We will ignore homozygous deletions for the simplicity.

Now we need to think about the prior distribution for the parameters of the Poisson data distribution. The parameter of a Poisson needs to be a positive real value. A very common choice for this type of parameter is the Gamma distribution. If we sample the parameter for each state from an arbitrary Gamma there is no relationship between the copy number and data. We will take a different approach here. We will sample a single value r from a Gamma and let the Poisson parameter for state c be $\theta_c = cr$. The parameter r can be interpreted as the haploid bin coverage i.e. the coverage of a bin with copy number 1. Thus we are assuming the total number of reads in a bin is a linear function of copy number. We can set the parameters a and b in the Gamma prior so the distribution has a mean equal to the haploid depth we expect. Alternatively they can be set to values which lead to a *vague* distribution i.e. one with high variance.

We finish off by setting the prior parameters for $\boldsymbol{\pi}$ and A. There is no obvious reason to prefer any particular initial copy number on a chromosome, so we let $\boldsymbol{\kappa} = (1, ..., 1)$ which leads to the uniform prior on the simplex (vectors that sum to one). We would like to encode a preference for adjacent bins to share the same copy number. To achieve this we let $\boldsymbol{\gamma}_{k\ell} = \zeta$ for $\ell \neq k$ and $\gamma_{kk} = \zeta + \xi$ for some values ζ and ξ. To be concrete we will let $\zeta = 1$ and $\xi = 10$. This prior will favour the hidden variables to stay in the same state as their neighbour. The full model is then

$$\begin{aligned}
\boldsymbol{\pi}|\boldsymbol{\kappa} &\sim \text{Dirichlet}(\cdot|\boldsymbol{\kappa}) \\
A_{k\cdot}|\boldsymbol{\gamma}_k &\sim \text{Dirichlet}(\cdot|\boldsymbol{\gamma}_k) \\
z_1^m|\boldsymbol{\pi} &\sim \text{Categorical}(\cdot|\boldsymbol{\pi}) \\
z_n^m|z_{n-1}^m, A &\sim \text{Categorical}(\cdot|A_{z_{n-1}^m \cdot}) \\
r &\sim \text{Gamma}(\cdot|a, b) \\
\theta_c|r &= rc \\
x_n^m|z_n^m, \boldsymbol{\theta} &\sim \text{Poisson}(\cdot|\theta_{z_n^m})
\end{aligned}$$

1.4.2. Inference

We can use the EM algorithm as discussed above. Minor modification need to be made to account for the fact we have multiple sequences (chromosomes).

1.4.3. Limitations

The main practical issue this model will run into is the assumption that the read counts only depend on the total copy number number. In reality issues such as mappability and GC content will also impact read depth. One approach is to perform some form of normalisation to the read counts. This typically changes the values to be non-integer, so we would need to abandon the Poisson assumption. A standard way forward is then to work with log transformed counts. We can then normalise those and model the data as following a Normal distribution instead of a Poisson.

If we do not want to abandon the current model, then we could try incorporating the additional data about covariates such as GC content into the model. Let g_n^m denote some measure of the GC content for bin n on chromosome n. A very simple way to incorporate this information is to modify the data distribution as follows.

$$x_n^m | g_n^m, z_n^m, \boldsymbol{\theta} \sim \text{Poisson}(\cdot | g_n^m \theta_{z_n^m})$$

We have simply scaled the Poisson parameter by the GC content information. Often incorporating observed data directly into the likelihood causes problems. One reason is the observed data maybe on the wrong scale or it may be error prone itself. A simple solution is to introduce an additional layer to the model. For example we could pick some distribution and set its mean to g_n^m denoted by $H(\cdot | g_n^m)$. Then we augment the model as follows

$$\begin{aligned} h_n^m &\sim H(\cdot | g_n^m) \\ x_n^m | h_n^m, z_n^m, \boldsymbol{\theta} &\sim \text{Poisson}(\cdot | h_n^m \theta_{z_n^m}) \end{aligned}$$

Now the observed value g_n^m only impacts the likelihood indirectly, and we could tune additional parameters for H to fit the data better. A reasonable choice for H in this case would again be a Gamma distribution with appropriate parameters.

The other major deficiency of the model is the assumption of a Poisson likelihood. The Poisson only has a single parameter which controls its mean and variance. As a result it can often be a poor fit to real data that has more variability than it can model. This is called overdispersion and we touched on in the first module. The standard solution is to switch to an overdispersed distribution such as the Negative-Binomial. We keep the mean of this distribution the same as for the Poisson, but introduce an additional variance parameter. It may be useful to introduce separate variance parameters for each copy number state, or parameterise the variance to depend on the state in way analogous to the mean.

1.5. Non-homogeneous sample model

The previous model was quite simplistic in assuming a homogeneous population of cells. In patient data the most major violation of this assumption would be normal contamination. Clonal population structure will also be an issue. In this section we will look at how we can improve the basic model to address these issues.

1.5.1. Normal contamination

Normal contamination is actually relatively simple to address. Let t be the tumour content of the sample. Then for any bin we have proportion t of cells with copy number c and proportion $(1-t)$ of cells with copy number 2 (assuming autosomes). We do not know the tumour content so we will need to specify a prior distribution. A simple choice is of course the continuous Uniform distribution. However, an informative Beta distribution maybe more useful if there is information from pathology estimates or other sources. Here we will use the simple Uniform.

The updated model is then as follows.

$$
\begin{aligned}
\boldsymbol{\pi}|\boldsymbol{\kappa} &\sim \text{Dirichlet}(\cdot|\boldsymbol{\kappa}) \\
A_{k\cdot}|\boldsymbol{\gamma}_k &\sim \text{Dirichlet}(\cdot|\boldsymbol{\gamma}_k) \\
z_1^m|\boldsymbol{\pi} &\sim \text{Categorical}(\cdot|\boldsymbol{\pi}) \\
z_n^m|z_{n-1}^m, A &\sim \text{Categorical}(\cdot|A_{z_{n-1}^m\cdot}) \\
r &\sim \text{Gamma}(\cdot|a,b) \\
t &\sim \text{Uniform}(\cdot|[0,1]) \\
\theta_c|r,t &= r(2\,(1-t)+t\,c) \\
x_n^m|z_n^m,\boldsymbol{\theta} &\sim \text{Poisson}(\cdot|\theta_{z_n^m})
\end{aligned}
$$

Inference will get slightly more complicated as we need to infer t. But the same basic steps will work.

1.5.2. Ploidy and identifiability

One issue we have ignored thus far is whether our model is *identifiable*. Roughly speaking a model is identifiable if every combination of parameters maps to a unique likelihood value in the frequentist setting, or joint probability value in the Bayesian setting. If a model is unidentifiable then we can have multiple MLE or MAP solutions. This raises the issue of how to select the best solution. In a fully Bayesian analysis, unidenitifiability is not theoretically an issue. We will simply observe a multi-modal posterior distribution. In practice unidenitifiability can make it hard to implement efficient MCMC algorithms as we need to be able *hop* between modes to explore the posterior. As a general rule of thumb it is best to avoid constructing models which are unidentifiable.

In the case of the simple model we defined for CNV analysis this unindentifiability crops up in the term $\theta_c = r\,c$ which paramaterises the Poisson observation distribution. The issue we have is that we can double the copy number of all segments and half the value of r to obtain the same value of θ_c. The likelihood of these solutions is thus identical. Because we have a prior on r the joint probability of these two different solution may be slightly different. However, the data is not informing the solution only the prior. In this case we say the model is *weakly identifiable*. This situation is worrying because we need to trust the prior for r if we are to trust the MAP solution. If we adopted the frequentist viewpoint, things would worse still as we would have no way to choose a solution. It may seem that adding tumour content into the model can help with this issue as our value of θ now depends on t as well. Unfortunately this is not the case, as we can also adjust the value of t to derive equally likely solutions.

In the field of CNV analysis the term *ploidy* is loosely used to refer to the average copy number of a sample. The exact definition varies slightly depending on the model, but the basic idea is the same. To the best of the author's knowledge all models for CNV analysis have the problem of unidentifiability caused by ploidy i.e. scaling the total copy number of all segments. The common approach employed is to manually select a solution from the of MLE or MAP estimates that have been found. This issue afflicts both bulk and single CNV analysis. Automating the selection of ploidy solution remains an open problem, and in the absence of any additional information or assumptions on the model appears very difficult. This is one reason that many tools for CNV analysis will give very different results when analysing the same sample.

There are a few avenues of research that could be pursued to address this issue. If we have additional information about the ploidy of the samples, for example from flow cytometry, we could leverage this information. This is only applicable to bulk analysis however, and is not full proof. We may also have domain specific knowledge about the cancer type we are analysing. For example we may know a priori that the genomes are relatively stable, so a near diploid solution is preferred. Alternatively we may have cancers which are known to undergo genome doubling early, in which case a near tetraploid solution would be preferred. Again this is not a general solution, as most cancers fall on a spectrum of genomic instability. Some single cell sequencing platforms image the cells before sequencing. It is possible this could be used to identify cells which are larger and more likely to have higher ploidy, though it is unclear whether this will work. The final idea would be to model the variance of each copy number state and try to extract information from this. Under certain modelling assumptions cells with higher overall copy number will have more variability in the observed reads counts.

In summary CNV analysis is challenging because of the ploidy problem. Currently there is no full proof automated way to address this issue. In practice this means that either manual curation or ad-hoc post-processing needs to be performed to identify a solution.

1.6. Allele specific copy number model

We will briefly outline the steps needed to produce an allele specific copy number model. We will assume the data comes in the form of haploid blocks. Let x_n^a be the number of reads supporting the a allele in the block n, x_n^b be the number of reads supporting the b allele, x_n^t be the total read depth and x_n^l the length of the block. Note that we no longer assume the blocks are of equal length. This will impact the expected number of reads in a block.

We know let the state space of the HMM be $\{(c_a, c_b) \colon c_a, c_b \leq C\}$ i.e. all pairs of allele specific copy numbers. Again we set a maximum number of copies C. Our state space has now grown from size C to C^2, so inference will be much slower.

We will stick with the Poisson distribution to model total read depth, x_n^t, where we scale the haploid depth by total copy. But we will now include the bin length. This alters the interpretation of r, which now is the haploid depth of a position not a bin.

We also need to model the allele specific counts. A simple choice is to use a Binomial for the number reads supporting the b allele. We will let the probability of success be $\frac{c_b}{c_a + c_b}$ so the probability depends on the relative copy number of the b allele.

We suppress the index for chromosomes for clarity. The full model is then as follows.

$$
\begin{aligned}
\boldsymbol{\pi}|\boldsymbol{\kappa} &\sim \text{Dirichlet}(\cdot|\boldsymbol{\kappa}) \\
A_{k\cdot}|\boldsymbol{\gamma}_k &\sim \text{Dirichlet}(\cdot|\boldsymbol{\gamma}_k) \\
z_1|\boldsymbol{\pi} &\sim \text{Categorical}(\cdot|\boldsymbol{\pi}) \\
z_n|z_{n-1}, A &\sim \text{Categorical}(\cdot|A_{z_{n-1}\cdot}) \\
r &\sim \text{Gamma}(\cdot|a, b) \\
\theta_n|z_n = (c_a, c_b), x_n^l &= r\,(c_a + c_b)\,x_n^l \\
x_n^t|z_n, \boldsymbol{\theta} &\sim \text{Poisson}(\cdot|\theta_n) \\
x_n^b|x_n^a, z_n = (c_a, c_b) &\sim \text{Binomial}\left(\cdot|x_n^a + x_n^b, \frac{c_b}{c_a + c_b}\right)
\end{aligned}
$$

Fitting this model is no harder than the original model, as we have not gained any additional parameters. Evaluating the likelihood will become more expensive due to the added Binomial term.

1.7. Discussion

1.7.1. Summary

In this module we saw how to construct probabilistic models for CNV data. We discussed different data representations that are commonly used. We then explored using HMM models to capture the spatial correlation of the data. We constructed a simple model for inferring total copy number profiles from homogeneous populations. We then explored how this model could be altered to address more complex issues. We note that all of these approaches could be combined to produce a more general model.

1.7.2. Other approaches

We focused on using HMMs in this module, primarily to illustrate how to handle spatial correlation in data. We could have avoided the use of HMMs, and the computational complexity this entails, by segmenting the data first. This is a widely used approach the effectively makes the data independent. This data can then easily be modelled using ideas similar to those in the previous module.