1. Phylogenetic analysis

1.1. Overview

In this module we will consider the problem of reconstructing the evolutionary tree or *phylogeny* that relates cancer clones. First we will review the types of data we have available to address this problem. Next we do a brief review of probabilistic models in phylogenetics. Finally, we develop a probabilistic model to infer phylogenies from bulk sequence data. This model has a few unique twists that are tailored to cancer genomics data.

1.2. Data types

The model we develop in this module will use bulk sequence data. Thus we focus the bulk of our discussion on this. However, single cell sequencing is becoming increasingly common and we offer some opinions on the potential challenges.

1.2.1. Bulk sequencing

The problem of building clonal phylogenies is a very active area of research in computational cancer biology. To date most work has focused on using bulk sequencing to address this issue. The major challenge of using bulk sequencing is that we need to deal with fact samples are mixtures of cell populations. There are two different approaches that are widely employed to do this.

The first approach is to perform deconvolution using methods such as the one we developed in module 2. The key assumption that all methods of this type make is that more prevalent mutations should originate higher up the tree. Within this category there are two different strategies. The first strategy is to perform the deconvolution and tree building simultaneously. The second strategy is to first perform the deconvolution and then reconstruct trees based on cellular prevalence. In general the first strategy is more computationally demanding. The key benefit is that the tree structure can inform the deconvolution step.

The second general approach is to use classical phylogenetic methods. In standard phylogenetic problems we observe a set of species with an associated matrix of observations. The values in this matrix can either be binary, for example the presence or absence of a morphological feature. Alternatively they can be nucleotides or amino acids if we have aligned sequence data. In the context of cancer it is common to use the presence/absence of a mutation within a sample as the feature matrix. Here species correspond to samples. The obvious problem with this approach is that bulk samples represent mixtures of clonal populations. Thus there is no guarantee that the presence of a pair of mutations in a sample implies they exist within the same cell. Because of this, one need to be careful when interpreting *sample trees*. Naively constructed trees based on the approach outlined above simply represent similarities between samples, not an evolutionary relationship. Later in this module we will see one way to address this issue.

One might wonder why anyone uses the second approach? Probably the dominant reason is that it is relatively simple, and does communicate some concept of similarity between samples. A more fundamental, but perhaps under appreciated reason is that we can account for mutation loss in this setup. In contrast, the deconvolution approach hinges on the fact that higher cellular prevalence mutations appear higher in the tree. This assumption is invalidated if mutations are lost. In practice many methods are robust to this issue. The basic strategy is to treat some mutations as outliers and not fit them to the tree. While practical, it is somewhat inelegant as we are using a model we believe to be wrong.

1.2.2. Single cell sequencing

The emergence of single cell sequencing provides hope we can avoid the clonal mixture issue that afflicts bulk sequencing. In principle single cell data is ideal for classical phylogenetic methods since cells are equivalent to species or individuals within a population. The major problem, as with all things single cell, is noise. To build a phylogeny we need a set of features for the observation matrix. The question is what could these be?

We could use SNVs, but low coverage whole genome sequencing is unlikely to have even a single read covering an SNV in a cell. This leads to a very sparse observation matrix. We could potentially use CNVs which are easily detectable from binned read count data even in low coverage sequencing. However, modelling CNV evolution is a formidable problem since we can no longer make the infinite sites assumption. The main issue is that we cannot treat the copy number of a bin as a feature, as independent events may have lead to the same observed copy number. One promising approach is to use the change-points associated with CNVs. These are less likely to violate the infinite sites assumption.

Scalability is another issue for single cell data. Typical phylogenetic methods become computationally demanding with 100s of species. We will soon have data sets with thousands of cells if not more. One option is to abandon probabilistic models, and use faster methods based on distance metrics. This is not without issue as the distance matrix may be noisy. An alternative is to use more advanced inference techniques. Sequential Monte Carlo (SMC) which we will discuss in the next module is one promising approach for Bayesian phylogenetics. There is also recent work using variational inference to improve MCMC samples in a post-processing step.

1.2.3. Summary

Phylogenetic reconstruction from bulk sequencing data is challenging, primarily due to the need to deal with sample mixtures. Phylogenetics using single cell sequencing in contrast is conceptually simple, but challenging due to noise and scale. In addition, there is a great deal more bulk data currently available than single cell. Thus bulk phylogenetic methods are likely to remain important for the foreseeable future. With that in mind we will turn to the problem of reconstructing phylogenies from bulk sequence data.

1.3. Probabilistic phylogenetic models

Here we provide a brief review of probabilistic models for phylogenetics. First, we introduce the basic problem setup. Next we look at how to define a probabilistic model and compute the probability of the data efficiently. We finish with a brief discussion of Bayesian models and the computational challenges.

1.3.1. Tree topology

The basic problem of phylogenetics is as follows. Given an observation matrix X with M rows corresponding to species and N columns corresponding to features (characters), we would like to infer the evolutionary tree relating the species. For simplicity we will assume this is a binary matrix i.e. $X \in \{0,1\}^{M \times N}$. Let the topology of the tree be denoted by $\tau = (E, V)$ where E is the set of edges and V the set of vertices. In many problems there will also be branch lengths, Λ, associated with each edge, though not in all cases. If we know the ancestral sequence or can identify a distantly related species to use as an outgroup we can root a tree. While this is not easy to do when looking at species evolution, in cancer it typically straightforward as we know the genotype of the normal progenitor cell (root) of the cancer. Thus we will focus strictly on rooted trees in this module. In a rooted tree we have a well defined notion of parent child relationships, and thus which nodes are ancestral.

1.3.2. Transition probabilities

The first step to defining a probabilistic phylogenetic model is to determine the probability of transitioning from one state to the other, in the simple case from a 0 to 1 or vice-verse. If we have branch lengths the probability of this transition will need to depend on the length of the branch. The usual approach is to define a rate matrix Q where Q_{ij} is the instantaneous rate of transition from state i to j. We can then define a transition matrix as $P = \exp(Qt)$ where t is time or branch length. Here P_{ij} is the probability of a transition from state i to j along the branch with length t. If we do not have branch lengths we can define P directly.

Remark 1. Identifiability crops up again here. If we rescale the matrix Q by a constant and divide the branch length by the same value, P remains unchanged. Biologically rescaling Q corresponds to altering the mutation rate. Informally this means we would expect to see the same number of mutations if the rate is high and time interval short as we would if the rate was low and time interval long.

1.3.3. Tree probability

Given a tree and transition probabilities the question is then how to define the probability of the data? For the moment assume we observe only a single character so that our data is a vector $\boldsymbol{x}$. We would like to compute $p(\boldsymbol{x}|\tau, P)$, that is the probability of the data given the tree τ and the transition matrix P. We will assume that we know not only the character sequence of the leafs (species), but also of the unobserved internal nodes of the tree. Let

- x_v denote the value of the character at leaf node v
- y_v denote the value of the character at internal node v
- $L(\tau)$ denote the leaf nodes of τ
- $I(\tau) = V \backslash L(\tau) \bigcup \{r\}$ denote the set of internal nodes that are not the root
- $\tau(v)$ denote the subtree rooted at node v
- $\rho(v)$ denote the parent of node v
- $\gamma(v)$ denote the set of children of v
- r denote the root node of the tree

Then we have

$$\begin{aligned} p(\boldsymbol{x}|\tau, P, \boldsymbol{y}) &= \prod_{v \in L(\tau)} P_{y_{\rho(v)} x_v} \prod_{v \in I(\tau)} P_{y_{\rho(v)} y_v} \\ &= \prod_{v \in \gamma(r)} P_{y_r y_v} p(\boldsymbol{x}|\tau(v), P, \boldsymbol{y}) \end{aligned}$$

where we have developed a recursive definition in the second line. The recursion is vital to efficient computation as we will see later. We of course do not know the values of the unobserved hidden nodes so we would like to marginalise them, that is sum over all possible states of $\boldsymbol{y}$ to obtain

$$p(\boldsymbol{x}|\tau, P) = \sum_{\boldsymbol{y}} p(\boldsymbol{x}|\tau, P, \boldsymbol{y})$$

This can be done efficiently using the Felsenstein pruning algorithm, which is another example of dynamic programming. The key quantity is

$$\alpha_v(s) = \prod_{u \in \gamma(v)} \sum_{y_u} P_{s y_u} \alpha_u(y_u)$$

and we define

$$\alpha_v(s) = \begin{cases} 1 & \text{if } s = x_v \\ 0 & \text{if } s \neq x_v \end{cases}$$

for leaf nodes. In words, the pruning algorithm recursively computes $\alpha_s(v)$ from the leaf nodes upwards. In each round we sum over all possible states for the current node and $\alpha(s)$ then take the product of all the child terms.

Remark 2. We can also modify the above recursion to compute the most probable set of internal states for the tree in the same way as we use the Viterbi algorithm for HMMs.

Remark 3. As we will see later it is not necessary that we observe the character values with complete certainty. The 0/1 condition for the leaf nodes can be replaced with probabilities.

1.3.4. Bayesian phylogenetics

Now that we can define the probability for the data, we need to specify priors to complete the Bayesian model. This includes a prior for the tree topology $p(\tau)$, the branch lengths if they are included $p(\Lambda)$ and any additional parameters which govern the transition probabilities $p(\theta)$. The joint probability is then given by

$$p(X, \tau, \Lambda, \theta) = p(X|\tau, \Lambda, \theta) p(\tau) p(\Lambda) p(\theta)$$

The question now is how to fit the model? The most challenging aspect is inferring the tree topology τ. Even computing the MAP estimate is hard in most cases as it requires searching over the space of all possible trees. For rooted binary trees with n leafs there are $\frac{(2n-3)!}{2^{n-2}(n-2)!} \approx n!$ possible trees. This renders exhaustive search impossible for all but small n. This is one case where full Bayesian inference via MCMC is not much slower than other approaches. The typical way to perform inference is to use MH moves for τ. Proposing a new random tree rarely works well, so local moves such as pruning a subtree and regrafting to another part of the tree are used in practice. The other parameters are typically easier to sample, and an array of MCMC methods such as MH or Hamiltoniain Monte Carlo can be applied.

Remark 4. Implementing MCMC for phylogenetics is a bit painful due to the book keeping required for moves in the tree space. Luckily many good tools such as Mr. Bayes and BEAST exist. These software packages provide a great deal of flexibility in defining models, and are worth looking into before developing your own tools.

1.4. Probabilistic model for mutation loss

We now turn to developing our own phylogenetic method in this section. We will address the problem of building trees from bulk sequencing. We first define the problem, then discuss a suitable model and finally look at some results from real world data.

1.4.1. Problem statement

We consider the problem of building phylogenetic trees from bulk sequence data. We will assume we have multiple tissue samples collected from the same patient. We will also assume CNV analysis has been performed and estimates of tumour content are available. Our input data will be allelic counts from SNVs. We will use this data to compute an observation matrix where the entries are the probability a mutation is *clonally present*.

We will assume that there is ongoing genomic instability in the cancer that deletes mutations. This will require us to define a non-standard model which is closely related to the Stochastic Dollo process first used in linguistics. This model assumes that mutations originate only once on the tree, but can be subsequently lost any number of times.

1.4.2. Model description

We would like to infer a rooted bifuricating tree relating a set of M samples where we observe N mutations. We assume that mutations originate only once on the tree at node w, which we cannot observe. Once a mutation has originated it is propagated to its children with probability $(1-\pi_l)$ or lost with probability π_l. Once the mutation is lost it remains lost in all descendants. There are no branch lengths in the model so the only parameter governing the transition probabilities is π_l. The main assumptions are

1. Mutations originate at most once on the tree.
2. Mutations can be lost after they are acquired.
3. Mutations evolve independently i.e. our tree probability decomposes as the product of mutations.

One unusual feature of the model is that we will not consider the observation data to be perfect. Rather we will assume there is a probability that a sample contains the mutations and denote this $p(z_i|\cdot)$. Our data matrix is then a set of probabilities that a mutation is present in a sample. We can revert to the simple case by setting $p(z_i|\cdot)$ to one for the observed value and 0 for all others. For example we can threshold on the number of reads supporting the mutation.

There are two reasons for using probabilities in the observation data. First, we assume the data comes from whole genome sequencing so read depth is on the order of 30x-100x. Depending on tumour content this means there is a reasonable chance we will fail to detect the mutation when it is present. We will see when we compute the likelihood a mutation is present we explicitly correct for tumour content. The second reason is to address the problem of samples representing mixtures of cells. As discussed earlier this raises issues when interpreting sample trees. A solution to this problem would be to only build trees with mutations that are clonal in the sample. This would guarantee that the mutations co-occur within the same cell. We do not have this information in practice, but using the same approach we developed in module 2 we can compute a probability for this.

1.4.3. Probability of clonal presence

To generate our data matrix, X, for phylogenetic reconstruction we need to compute the probability a mutation is clonally present. We define this to be the probability a mutation has cellular prevalence 1.0 in the sample. To compute this probability we adopt an approach similar to module 2. Let c_b denote the number of mutated copies of a locus and c_t the total number of copies. Let t be the tumour content of the sample. Then the probability of observing a read with the mutation, if it had cellular prevalence 1.0, is

$$r = \begin{cases} \frac{c_b t}{2(1-t) + c_t t} & \text{if } c_b > 0 \\ \varepsilon & \text{if } c_b = 0 \end{cases}$$

so the probability of observing b reads with variant out of d total is

$$p(b|d, c_b, c_t, t) = \text{Binomial}(b|d, r)$$

Since we do not know the number of copies of the variant, we sum over all possible values of $c_b \in \{1, \ldots, c_t\}$ to compute the probability of presence. We assume a uniform prior for c_b so all values have equal prior weight. For the probability of absence we set $c_b = 0$.

1.4.4. Tree notation

- $V(\tau)$ denote the vertices of the tree τ
- $L(\tau)$ denote the leaves of the tree
- $D(i)$ denote the nodes descendant from node i
- $L(i)$ denote the leaves descendant from node i
- $C(i)$ denote the children of node i
- $\rho(i)$ denote the parent of node i
- $A(i)$ denote the ancestors of node i i.e. all nodes on the path from i to the root
- w denote the tree node at which a mutation originated
- z_i be an indicator if node i has the mutation
- π_l be the probability of losing a mutation along an edge
- $p(z_i|\cdot)$ be the likelihood a variant is present

1.4.5. Tree probability

We will develop a slightly non-standard recursion for computing tree probabilities. The reason for this is that the single origin constraint adds dependencies between branches in the tree. If a mutation originated in one branch it cannot originate again in another branch. Conditioned on the originating branch, $(\rho(w), w)$, the losses are Markovian on the sub-tree $D(w)$. Informally this means once we know where a mutation originates evolution is independent among the nodes. Note that once a mutation is lost it will be lost in all descendants as well due to the single origin assumption.

Mutations can originate at any point along the branch of the tree. To simplify the discussion we say a mutation originates at a node if it occurs at any point along the branch between the node and its parent. We use the same convention for lost mutations. As we will see later we can then infer at which node (clone) a mutation originated, and which clones subsequently lost the mutation.

To compute the probability of the tree we imagine picking a node w as the node at which a mutation originated. Then we would like to compute $p(\boldsymbol{x}|\tau, w)$ the probability of the data given the tree τ and origin node w. To do this we introduce the function $Q(i, \tau)$, the probability the mutation is present at node i given all possible combinations of losses on the sub-tree rooted at i. This can be computed recursively as follows

$$Q(i,\tau) \;\; = \;\; \begin{cases} \pi_l \, p(z_i = 0|\cdot) + (1-\pi_l)\, p(z_i = 1|\cdot) & \text{if} \;\; i \in L(\tau) \\ \pi_l \prod_{j \in L(i)} p(z_j = 0|\cdot) + (1-\pi_l) \prod_{j \in C(i)} Q(j,\tau) & \text{if} \;\; i \notin L(\tau) \end{cases}$$

The first line is the initial condition for the leaf nodes. The term $\pi_l \, p(z_j = 0|\cdot)$ accounts for losses on the branch above the node. The second term $(1-\pi_l)\, p(z_j = 1|\cdot)$ accounts for the probability of not being lost and observed. The second line is the internal node recursion. The first term, $\pi_l \prod_{i \in L(j)} p(z_i = 0|\cdot)$, again accounts for a loss but also enforces the loss for all children. The second term, $(1-\pi_l) \prod_{i \in C(j)} Q(i,\tau)$, accounts for the event that the mutation is not lost and then considers all possible loss patterns on the child subtrees.

With $Q(j, \tau)$ defined we then have

$$p(\boldsymbol{x}|\tau, w) \;\; = \;\; Q(w,\tau) \prod_{i \in L(\tau) \setminus L(w)} p(z_i = 0|\cdot)$$

which decomposes into a term for the subtree the mutation originates on, and term for all other nodes. Ultimately we want to compute $p(\boldsymbol{x}|\tau)$ which can be obtained by marginalizing w over all nodes in the tree. In the absence of additional information we assume a uniform prior for w i.e. $p(w) = \frac{1}{|V(\tau)|}$. Thus we have

$$p(\boldsymbol{x}|\tau) \;\; = \;\; \sum_{w \in V(\tau)} p(\boldsymbol{x}|\tau, w)\, p(w)$$

To obtain the probability for all the data we use the assumption mutations are independent. Thus the probability of the data is given by

$$p(X|\tau) \;\; = \;\; \prod_{n=1}^{N} p(\boldsymbol{x}_n|\tau)$$

where $\boldsymbol{x}_n$ is the data for the n^{th} mutation.

1.4.6. Inference

We need to infer two parameters in this model: the tree topology τ and the probability of mutation loss π_l. Fitting this model is potentially challenging because of the need to explore the space of trees. As stated earlier MCMC methods are usually favoured for addressing this problem. However, it is rare to perform multi-region sequencing on more than ten samples. This means that the number of trees is at most on the order of millions. In addition the computation of the tree likelihood can be performed in parallel for each tree topology. So we have a rare example in phylogenetics where performing MAP estimation is reasonably easy. To do this we simply enumerate all trees and optimize the mutation loss parameter for each tree. We then take the one with the highest joint probability as our solution.

Remark 5. Since we first developed this model we have generated larger datasets. Even with 12 samples the MAP estimation approach is no longer computationally feasible. An area of future work is to implement a proper MCMC sampler for this method.

1.4.7. Inferring origin, presence and loss of mutations

Given the MAP estimates for the tree $\hat{\tau}$ and the probability of loss $\hat{\pi}_l$ we can compute the most likely node where a mutation originated. We can also identify at which nodes the mutation is present and at which the mutation was lost. The strategy is to maximize the following quantity

$$p(w, z|x, \hat{\tau}, \hat{\pi}_l) \;\; = \;\; p(x|w, z, \hat{\tau}, \hat{\pi}_l) p(z|w) p(w)$$

To achieve this we modify the previous recursion for $Q(i,\tau)$ as follows

$$\hat{Q}(j,\tau) \;=\; \begin{cases} \max\left\{\pi_l\, p(z_j=0|\cdot),\,(1-\pi_l)\, p(z_j=1|\cdot)\right\} \\ \max\left\{\pi_l \prod_{i\in L(j)} p(z_i=0|\cdot),\,(1-\pi_l)\prod_{i\in C(j)} \hat{Q}(i,\tau)\right\} \end{cases}$$

In other words we replace summation with maximisation in the recursion. This is exactly the same as the relationship between the forward-backward and Viterbi algorithm. Note that in addition to $\hat{Q}(j,\tau)$ we also keep track of the choice we made to get the maximum value at each node. This allows us to label the presence of the mutation in each node, and hence identify the origin and loss points.

1.4.8. Results and limitations

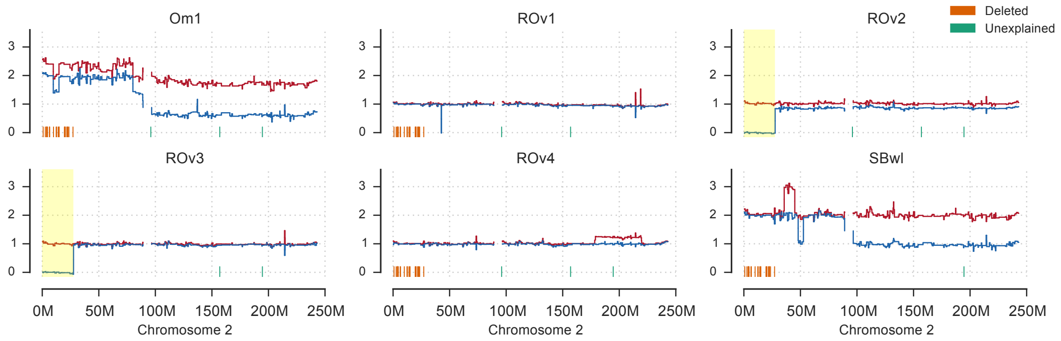

Figure 1. Results of the mutation loss model. The panels show copy number profiles from six samples from a patient with high grade serous ovarian cancer. Mutations predicted to be lost somewhere in the tree are plotted as sticks at the bottom of each panel. Highlighted in yellow is an example of a deletion event overlapping predicted losses.

With the model in hand the question is how can validate the results of the model? One ad-hoc approach we can use is to look at the predicted lost mutations. If the intuition that copy number changes are deleting mutations is valid, we should expect to see evidence of copy number changes in the samples where the mutation is lost. Figure 1 illustrates this point nicely. Here we see two samples, ROv2 and ROv3, where a focal deletion of one allele corresponds to the loss of mutations.

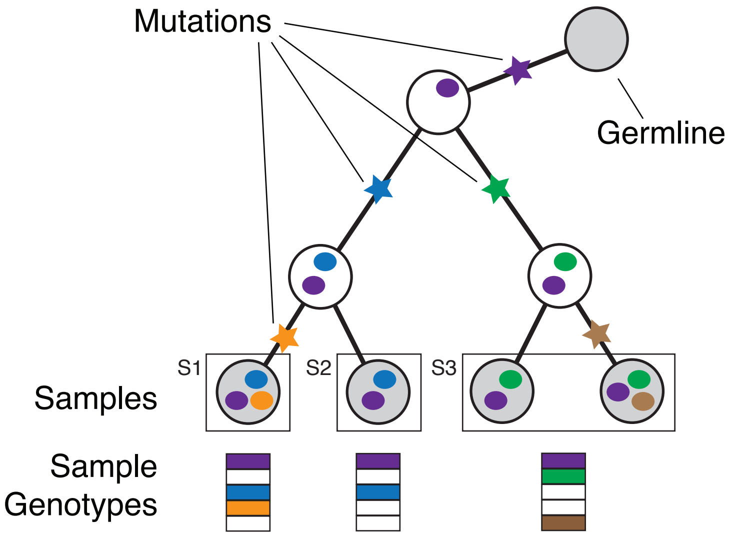

Figure 2. Simplified example of sample tree construction. On the top we show the phylogeny and evolutionary history of mutations. The observed presence/absence data is shown on the bottom.

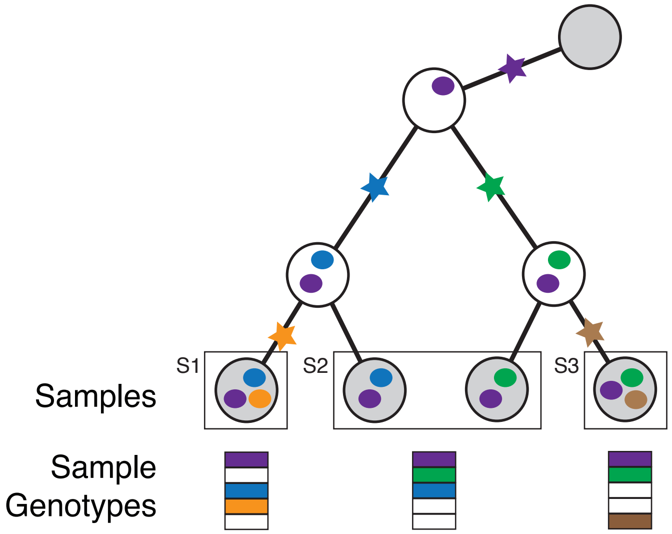

Figure 3. Simplified example of sample tree construction. On the top we show the phylogeny and evolutionary history of mutations. The observed presence/absence data is shown on the bottom.

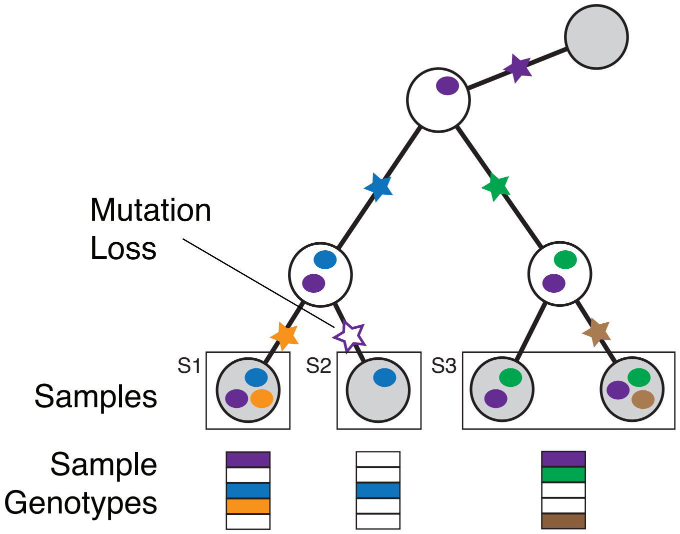

Figure 4. Simplified example of sample tree construction. On the top we show the phylogeny and evolutionary history of mutations. The observed presence/absence data is shown on the bottom.

No model is perfect, and the model described so far has one critical weakness. If any sample is a balanced mixture of two clones then it will fail badly. To understand the issue we need to consider a hidden assumption we are making. Namely, the probability of clonal presence is accurate. If we assign a high probability of presence to sub-clonal mutations from clones from different parts of the tree we have a problem.

To understand this issue first consider Figure 2 which illustrates an ideal case of building sample trees. There is no mutation loss in this example. In addition no samples are mixtures of clones from different parts of the tree. Thus we can look at the observed presence/absence data and easily determine samples S1 and S2 are more similar to each other than S3. Now consider Figure 3 where sample S2 is a mixture of clones from different parts of the tree. The presence/absence data is now ambiguous since S2 shares an equal number of mutations with S1 and S3. Finally consider Figure 4 where we have mutation loss in sample S2. Now sample S1 shares the same number of mutations with S2 as S3. The model we constructed accounts for the case in Figure 4 but not 3. As result, when we do have mixtures of divergent lineages in a sample the model is forced to use mutation loss to explain the mutations which do not fit the tree. This can lead to major failures for the method as illustrated in Figure 5. In this example we see a large number mutations predicted to be lost. Moreover, they are uniformly spread across the genome with no corroborating copy number changes in most cases. The problem is that the second sample, LOv2, is actually a mixture of clones from LOv1 and the other samples.

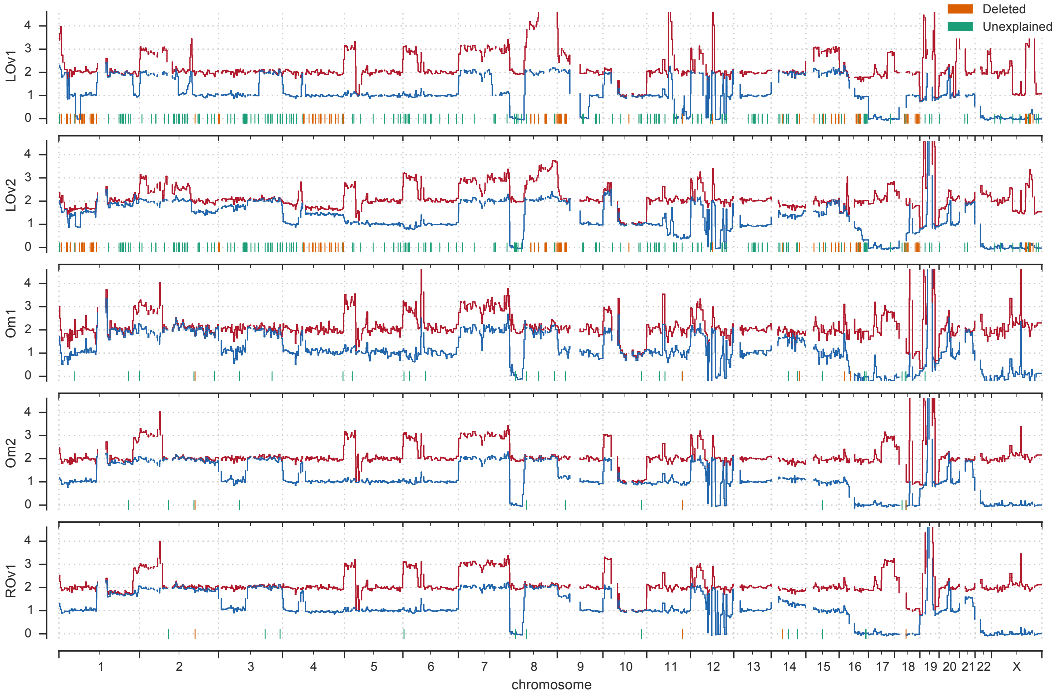

Figure 5. Example of erroneous model predictions.

The main point of this discussion is to illustrate that no model is perfect. It is important to understand the limitations of any model, and critically inspect the results. In the study where we first used this model, we were well aware of the deficiencies of the model. To address them we acknowledged we could not apply this approach to all patients in the study. For the cases we could not, we reverted to using single cell sequencing which was significantly more expensive and time consuming.

1.5. Discussion

In this module we discussed probabilistic phylogenetic models. We looked at the data types that are available and discussed the challenges associated with them. We then reviewed the basics of probabilistic phylogenetic models. Finally we developed a novel model for inferring phylogenies from bulk sequencing. We saw the success and limitations of this model when applied to real data.