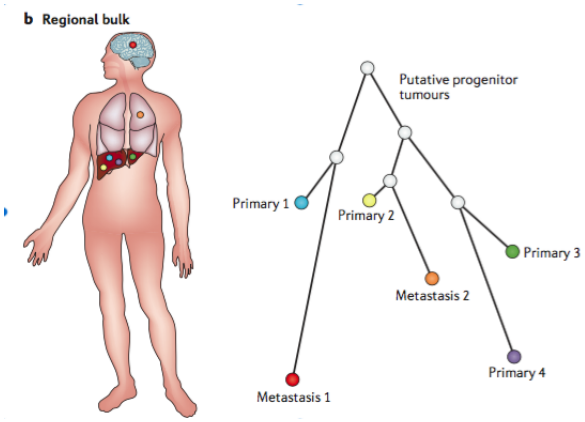
Assessment of heterogeneity in multiple samples of bulk tumor DNA

Emily Simons, Margaux Hujoel, Alexander Munoz

April 5, 2017

1 Background

Tumors are genetically heterogenous populations of stromal and cancer cells. The genetic heterogeneity of the cancer cell fractions of tumors can be defined by phylogenic distance of the clones and dominant subclones observed when a patient has more than 1 sample from their primary tumor and/or metastases (as in figure below.) The phylogenies of tumor clones can be derived from differences in somatic single nucleotide variants across the cancer samples. An increase in tumor genetic heterogeneity has been associated with treatment failure and decreased survival time in some cancer types (Landau et al, Cell. 2013.) Characterizing degree of tumor heterogeneity has the potential to help identify patients with increased tumor heterogeneity that are likely to have a shorter progression-free survival compared to patients with lower tumor heterogeneity. These patients could be selected for more intense monitoring to enable quicker transitions to second-line therapies.



2 Aims

Our overarching aims for this project are to develop a clustering method to identify most parsimonious phylogenic tree for simulated data and then to compare this method to existing methods used to identify phylogenic trees from multiple samples of tumor DNA (such as Canopy or the expectation maximization method outlined by Zare et al. 2014). Implicit in this aims statement is the ability to generate data from a known tree in order to assess our method.

3 Methods

We will first attempt to develop a clustering method assuming perfect phylogeny and ignoring copy-number alterations (although if we have the time we would like to extend our method to encompass copy-number alterations). Our method will take as inputs the number of "normal" reads at each location as well as the number of alternate reads. We will input reads for multiple samples. We will then first test our method with the DNA mixing experiment data.

Following this test we will simulate data from a tree of our choosing (see section on simulated data on our approach of how we will do this). We will most likely first generate the data with no noise and then introduce noise in order to determine how robust our (and other) methods are. For example, given raw reads, we can tell that the number of clusters lies between 1 and the number of samples. We can optimize then optimize this cluster count for our problem looking at the scaling of the distance metric with k.

We then, having specified this tree, know how many clusters there are in our tree. Therefore, if our method has to take in the number of clusters as a prespecified parameter, we will analyze what happens if we give our method the incorrect number of clusters. As the number of clusters is an unknown parameter (although in our simulations it is fixed and known) we want to determine (either ourselves or through a literature review) some way to decide on the optimal number of clusters.

Finally, we will compare our method to Canopy or another pre-existing method. We will compare the trees that are derived to each other, as well as to the true underlying tree from which our simulated data was drawn from. We, time permitting, will also compare the methods with respect to robustness to noise.

4 Simulated data

We intend to simulate data from our tree as follows:

We can construct a given tree we wish our data to emulate. From this tree we can construct a genotype matrix, denoted $Z_{N.C}$, where N is the number of

mutated loci and C is the number of clones in the tumor (we use the notation used by Zare et al.). This genotype matrix is a matrix of 0s and 1s: if $Z_{i,c} = 1$ then clone c has the variant allele at locus i (it is 0 otherwise). According to Zare et al., column 1 of Z represents the "normal" and thus it should be constructed of all zeros. However, because of possible contamination, matrix Z can be represented as a random variable, where each entry is an independent Bernoulli trial:

$$Z_{i,c} \sim \text{Bern}(\mu_{i,c})$$

where $\mu_{i,c}$ is the probability of reading a mutation at locus i in clone c. We also need a clone frequency matrix $P_{C \cdot M}$ where M is now the number of samples we have (including 1 normal, so we have $M - 1$ tumor samples). $P_{c,j}$ is the proportion of cells of clone c in subsection j: that is it represents the tissue composition of the C clones in our samples (of which we have M). As such, we can easily note that each column of P must sum to 1. Letting the first column be the normal sample we have the first column as $(1, 0, ..., 0)$. Given our tree, we can construct $Z_{N \cdot C}$ easily. We can then decide upon a $P_{C \cdot M}$, just ensuring the columns sum to 1. Zare et al then derive how the matrix $X_{N \cdot M}$ of variant allele counts ($X_{i,j}$ is the number of reads supporting the mutation at locus i in subsection j) is related to our Z and P matrices as well as the $R_{N \cdot M}$ matrix which is just the matrix of total counts ($R_{i,j}$ total number of reads at locus i in subsection j).

As such, we can "back track" to go from our known Z matrix to the X matrix. The R and P matrixes can drawn at time of data simulation. To simulate R, we assume a constant read depth (a fair assumption given that sequencing machines usually produce a fixed number of reads). Thus, the conditional distribution of R given the sample is well defined by the multinomial. For simplicity, we will assume a flat probability mass for the multinomial probability vector over the loci for a fixed subpopulation. Now, let $\mathfrak{D}_{\text{total}}$ be the read depth of our sequencing machine. Then, the total read depth of each sample is well-approximated by:

$$\mathfrak{D}_{\text{sample}} = \frac{\mathfrak{D}_{\text{total}}}{M}$$

Now with this read depth, we place each read in a bin of the locus it maps to. This bin placement follows the multinomial distribution. Assuming a flat probability vector, a column of R follows:

$$R|\text{sample} \sim \text{Mult}\left(\mathfrak{D}_{\text{sample}}, \left(\frac{1}{N}, \frac{1}{N}, ..., \frac{1}{N}\right)\right)$$

Because we assumed a flat probability vector, this model excludes any copy number variations. We can change this model in our project if we would want to account for copy number variations as well.

We can also simulate P, the clone frequency matrix. We can simply draw

these from a uniform and then normalize such that the conditional distribution of a column (a sample) is a valid probability mass function and sums to 1.

$$P_{i,j}|j \neq 0 \propto \text{Unif}(0,1)$$

This assumption of the Zare et al paper also provides us an opportunity to improve as we could use a beta distribution using the maximum likelihood α and β parameters (estimated after clustering) instead of the flat uniform assumed by Zare et al:

$$P_{i,j}|j \neq 0 \propto \text{Beta}(\hat{\alpha}_{\text{MLE}}, \hat{\beta}_{\text{MLE}})$$

Because each $Z_{i,j} = 1$ (i.e. mutant in clone j at locus i) represents half of the subsection's proportion, this implies a probability of a variant allele in clone c subsection j of

$$\pi_{i,j} = \frac{1}{2} Z_i \cdot P_j$$

We can thus conclude that the number of reads will follow a binomial distribution since we are summing independent bernoullis (again, each read is an independent bernoulli, variant read count is just the sum of these bernoullis). So:

$$X_{i,j} \sim \text{Bin}(R_{i,j}, \pi_{i,j})$$

where $X_{i,j}$ are variant allele counts. Now understanding the distribution of the draws, the introduction of noise is evident.

If the above simulation method does not work as we desire, we can refer to the Zare et al. paper which talks about the method they used to simulate data, although they did not build their Z from a known tree, as we are attempting to do.