Chapter 2

Methods

2.1 Principles of subclonal reconstruction

Reconstruction of the subclonal architecture of a tumour involves three main components: Estimating copy number, adjusting SNV VAFs for copy number alterations to obtain CCF values and inferring the subclonal architecture from the CCF data. This section contains a description of all methods that I use for subclonal inference. These methods form the basis of all results reported on in this thesis, sometimes as part of a much larger procedure as is detailed in Chapter 6. This chapter also contains a description of avenues that have been explored, but were deemed not an improvement. An earlier version of the text in this section has appeared in Dentro et al. (2017).

2.2 The Battenberg algorithm

Battenberg was originally developed to study the unique PD4120 sample and was briefly described in the supplement of Nik-Zainal et al. (2012a). Since then it has been adapted and extended to run with whole genome sequencing and SNP 6.0 data from 1000s of genomes and has become a standard part of the cancer genome analysis pipelines at the Sanger. This section contains a complete description of the whole genome sequencing pipeline and algorithm. In brief: Battenberg uses the 1000 Genomes SNP locations with B-allele frequency (BAF) and relative amounts of DNA (logR) as input from either whole genome sequencing or SNP 6.0 arrays. Heterozygous SNPs are identified from the matched normal sample, after which the SNPs are phased into haplotype blocks to obtain accurate BAF values. Battenberg then performs segmentation, finds an initial purity and ploidy combination before fitting a global copy number profile. Finally, it identifies segments for which the underlying BAF cannot be explained by clonal copy number and it will fit subclonal copy number as a mixture of two separate major and minor allele states.

2.2.1 Pre-processing

Battenberg starts by reading in allele counts for all 1000 genomes SNPs, which are directly obtained from the tumour and matched normal BAM. SNPs are removed from the pool if they appear on the list of unreliable SNPs (identified in a panel of 200 normal genome sequences) or when they are covered by fewer than 10 reads in the normal or 1 read in the tumour. The normal is used to identify SNPs that are heterozygous in the germline of the patient and therefore requires that the normal is from the same individual as the tumour. All SNPs then go into haplotype reconstruction, after which the germline heteroygous SNPs are used for segmentation and fitting.

2.2.2 Reconstructing haplotype blocks

Battenberg primarily uses allelic imbalances to estimate copy number. To observe these imbalances, it is helpful to look at the B-allele frequency (BAF) of a germline heterozygous SNP. For sequencing data the BAF can be calculated as:

$$BAF_i = \frac{r_{B,i}}{r_{A,i} + r_{B,i}} \tag{2.1}$$

where $r_{A,i}$ and $r_{B,i}$ represent the total reads reporting allele A and B respectively. Alternatively, the BAF can be expressed as a function of the number of chromosome copies of allele A and B (n_A and n_B respectively):

$$BAF_i = \frac{n_{B,i}}{n_{A,i} + n_{B,i}} \tag{2.2}$$

A germline heterozygous SNP will have a BAF of approximately 0.5 in the absence of any copy number changes. Deviations from 0.5 therefore can be used to detect somatic aberrations. As tumours are often admixed with normal cells, establishing the copy number state of an aberration based on the deviation of BAF requires estimating the fraction of tumour cells in the sample (the tumour purity). The number of chromosome copies in the formula above should therefore be split into a contribution of ρ tumour cells and (1-ρ) normal cells:

$$BAF_i = \frac{\rho n_{B,t} + (1-\rho) n_{B,n}}{\rho(n_{A,t} + n_{B,t}) + (1-\rho)(n_{A,n} + n_{B,n})} \tag{2.3}$$

where ρ represents the tumour purity, $n_{A,t}$ and $n_{B,t}$ the number of chromosome copies in tumour cells and $n_{A,n}$ and $n_{B,n}$ the number of chromosome copies in normal cells. Several methods have been developed to co-estimate clonal copy number states and tumour purity based on these allele-specific signals (Carter et al., 2012; Ha et al., 2014; Van Loo et al., 2010).

Tumours that exhibit much clonal genomic instability will show deviation of the BAF for large proportions of the genome. In such tumours, the BAF values show clear levels corresponding to different clonal states, which translates into more usable signal for methods that co-estimate copy number states and tumour purity. However, genomes that show large amounts of subclonal genomic instability will show a range of different BAF values and will be more difficult to fit.

Fig. 2.1 shows allele frequency values for a number of example cases that are affected by copy number changes and different normal cell admixtures. Panel A shows a region with no copy number alterations in a tumour that has no normal cell infiltration. One expects both alleles to be present in equal proportions, resulting in allele frequencies of 0.5. Panel B shows a region with a clonal gain. The bands representing allele A and B are clearly separated, with allele A representing two thirds of the total chromosome copies and allele B one third. Panel C contains a similar gain, but in a sample with 75% tumour purity, resulting in a smaller difference between the bands. Panel D shows the gain, again with 75% tumour cells, but now the coverage is reduced from 100X (as in panels A, B and C) to 40X. The bands appear to be overlapping as lowering the depth increases the noise and widens the bands. Panel E shows an example where the gain is subclonal in 60% of tumour cells resulting in further overlap of both bands. And finally panel F shows a subclonal loss in 40% of tumour cells.

Fig. 2.1 illustrates that the allele frequencies of individual SNPs are subject to statistical variation and this noise increases with lower coverage. Combining SNPs into haplotype blocks through phasing can mitigate this effect (Carter et al., 2012; Nik-Zainal et al., 2012b). Through haplotype phasing, information can be combined across multiple SNPs within a region of copy number change, by matching alleles across SNPs. For example, for SNP i, allele A may correspond to the maternal allele, while for SNP $i+1$, allele B may correspond to the maternal allele. If these are combined appropriately, smaller deviations of the BAF from the normal state can be detected, and higher precision copy number changes, including subclonal copy number changes, can be inferred.

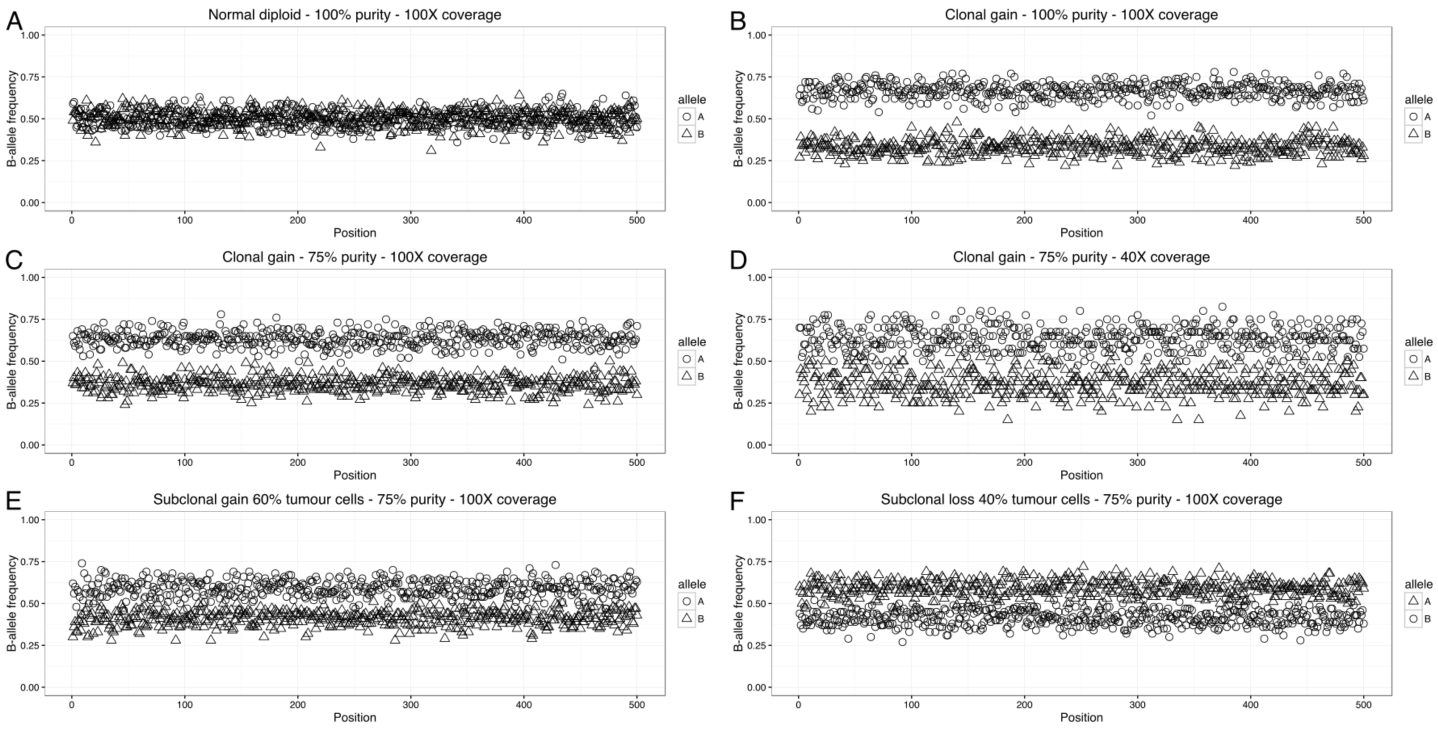


Fig. 2.1 The BAF is used to determine whether subclonal copy number exists, often in low fractions of overall cells within the sequencing sample. This figure illustrates that for alterations in low proportions of cells one should perform haplotype reconstruction to reduce noise. (A) The BAF centres around 0.5 (50% of sequenced alleles contain the SNP, as it is heterozygous in a background of two chromosome copies) when no copy number alterations have occurred and the circles (allele A) and triangles (allele B) appear exactly on top of each other. (B) When an alteration occurs, the BAF changes showing two clear ‘bands‘ (at 100% purity and 100X coverage). In this scenario an unbiased estimate of the BAF can be obtained from either of the bands separately. (C) When the tumour purity drops to 75% the two bands appear closer to each other and closer to 0.5, but at 100X they still visually separate. (D) When the gain occurs in a purity of 75%, but the tumour is sequenced at 40X instead of 100X the bands both widen considerably and start to merge. By calculating the BAF from one of the bands as both bands now contain both circles and triangles. (E) So far, only clonal gains have been simulated. This panel shows a subclonal gain in 60% of tumour cells in a tumour with 75% purity sequenced at 100X. Compared to panel C, the bands are much closer together and more difficult to separate. Panel F shows that the bands are equally difficult to separate form a loss in 40% of tumour cells occurs.

2.2.3 Fitting a global copy number profile

Two main components must be taken into account when fitting a copy number profile: Infiltration of normal cells (tumour purity) and tumour cell aneuploidy (tumour ploidy). Battenberg takes a similar approach as ASCAT (Van Loo et al., 2010) by considering a range of purity and ploidy combinations to pick a solution. After a combination is established, each segment is then assigned allele specific copy number states.

The grid search procedure is performed twice, first with large steps to find an initial optimum and then with small steps to refine the solution. The grid search procedure takes a range of purity (ρ) and ploidy (ψ_t) values and calculates the proportion of the genome fit with clonal copy number with each combination, through the steps described in the next section. It then picks the ρ and ψ_t pair that maximises the proportion of the genome with clonally altered copy number.

Finally, the copy number states of both alleles of a segment s are established through:

$$n_{A,s} = \frac{\rho - 1 - (1 - b_s)2^{l_s}(2(1-\rho) + \rho\psi_t)}{\rho} \tag{2.4}$$

$$n_{B,s} = \frac{\rho - 1 + b_s 2^{l_s}(2(1-\rho) + \rho\psi_t)}{\rho} \tag{2.5}$$

where $n_{A,s}$ is the copy number call for allele A of segment s, b_s and l_s are the BAF and logR of the segment and ψ_t is the average ploidy of all tumour cells in the sequencing sample.

2.2.4 Testing whether a segment is clonal

After fitting clonal major and minor allele copy number states, we can test whether these states accurately explain the observed BAF. If the BAF is not well explained by the best clonal states, then the segment is subclonal. This section explains the details of the test, the next section explains how the test is applied. The obtained n_A and n_B (through eqs. 2.4 and 2.5) can be non-integer values and therefore have to be rounded to obtain clonal copy number states. This can be achieved by rounding either allele up or down, yielding four possible options (explained further in the next section). For each option the expected BAF, given rounded alleles $\widehat{n}_{A,s}$ and $\widehat{n}_{B,s}$, is calculated using:

$$\widehat{b}_s = \frac{1-\rho-\rho\widehat{n}_{A,s}}{2+2\rho+\rho(\widehat{n}_{A,s}+\widehat{n}_{B,s})} \tag{2.6}$$

A choice is made between the four options by taking the combination of alleles that minimises the distance between the observed BAF b_s and the expected BAF $\widehat{b}_s$.

Finally, the $\widehat{b}_s$ value corresponding to the chosen allele combination is tested against the observed BAF through a t-test and accepted as clonal if the p-value is not significant _using 0.05 as the significance cutoff.

2.2.5 Fitting subclonal copy number

Once exact allele frequencies of segments have been calculated and a clonal copy number profile has been fit, subclonal copy number changes can be detected. As a first step, for each segment, one can determine whether the BAF value of this segment can be explained by a clonal copy number change (as detailed in the previous section). Deviation of the observed exact allele frequency from the theoretical allele frequency can be used to identify a segment having a subclonal copy number state, i.e. a combination of two or more populations of tumour cells with different copy number states, in addition to a population of normal cells.

When such a segment is fit with a clonal copy number state, the multiple subclonal states are combined into a single (integer) representation. For example, if the real copy number state of the segment is 2+1 (2 copies of one parental allele and 1 copy of the other allele) in 80% and 1+1 in 20% of tumour cells (i.e. on average 1.8+1), its clonal fit will likely be 2+1 in 100% of tumour cells (1.8+1 rounded up). The observed allele frequency will therefore deviate from the frequency expected under the clonal copy number fit, allowing us to infer that the segment cannot be explained by a clonal copy number state.

The type of subclonal copy number depends on the different copy number states at the locus and their respective fractions of tumour cells. This problem has multiple solutions, as there can be any number of subclones with distinct subclonal copy number states. However, for any given segment, the most parsimonious assumption is that there are only two distinct copy number states, and that those copy number states differ at most by one chromosome copy (i.e. are separated by only one copy number event). Battenberg therefore assumes two distinct major and minor allele states, which are separated by one copy number event.

Under this assumption, given allele-specific copy number values n_A and n_B (integer if clonal, non-integer if subclonal), there are four options for the theoretical clonal allele frequency $\widehat{h}_f$ (assuming diploid copy number in the normal cell population):

Allele A and B are both rounded down:

$$\widehat{h_f} = \frac{\rho \lfloor n_B \rfloor + 1 - \rho}{\rho(\lfloor n_A \rfloor + \lfloor n_B \rfloor) + 2(1-\rho)} \tag{2.7}$$

Allele A is rounded down and B is rounded up:

$$\widehat{h_f} = \frac{\rho \lceil n_B \rceil + 1 - \rho}{\rho(\lfloor n_A \rfloor + \lceil n_B \rceil) + 2(1-\rho)} \tag{2.8}$$

Allele A is rounded up and B is rounded down:

$$\widehat{h_f} = \frac{\rho \lfloor n_B \rfloor + 1 - \rho}{\rho(\lceil n_A \rceil + \lfloor n_B \rfloor) + 2(1-\rho)} \tag{2.9}$$

Allele A and B are both rounded up:

$$\widehat{h_f} = \frac{\rho \lceil n_B \rceil + 1 - \rho}{\rho(\lceil n_A \rceil + \lceil n_B \rceil) + 2(1-\rho)} \tag{2.10}$$

Subclonal segments can be identified by testing the observed allele frequency h_f against the theoretical $\widehat{h_f}$ of all four scenarios and accepting a segment as subclonal if the observed h_f is significantly different from $\widehat{h_f}$ in all. If the segment is deemed to be subclonal we choose one of the above four scenarios as the most likely explanation of how subclonal copy number was rounded into clonal. The scenario that explains the observed h_f best is picked, providing two combinations of major and minor allele copy number states.

Finally, having obtained the states, we estimate the proportions of tumour cells that contain each of the two major and minor allele combinations. Formally, if a fraction of tumour cells τ shows copy number state $n_{A,1}$ + $n_{B,1}$ and a fraction of tumour cells 1-τ shows copy number state $n_{A,2}$ + $n_{B,2}$, τ can be calculated as:

$$\tau = \frac{1 - \rho + \rho n_{B,2} + 2h_f(1-\rho) - h_f\rho(n_{A,2} + n_{B,2})}{h_f\rho(n_{A,1} + n_{B,1}) - h_f\rho(n_{A,2} + n_{B,2}) - \rho n_{B,1} + \rho n_{B,2}} \tag{2.11}$$

2.2.6 Extensions to segmentation

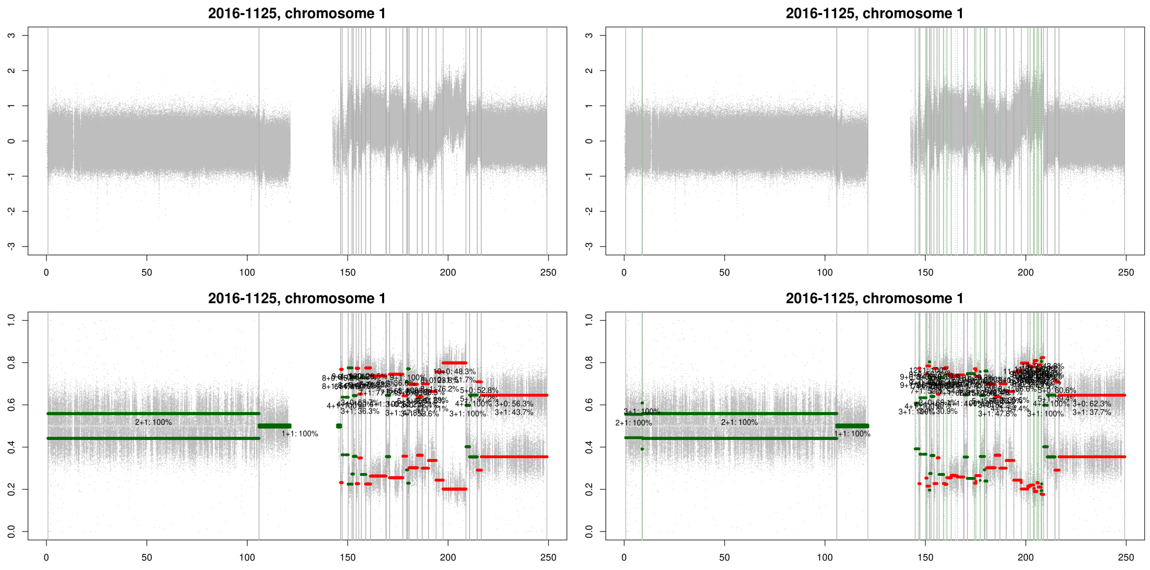
Segmentation of the phased BAF data is performed by piecewise constant fitting (PCF) in Battenberg. PCF models the data as a step-function to explain the observed data by a number of discrete copy number segments as described in (Nilsen et al., 2012). PCF is provided with BAF data for heterozygous SNPs and requires two parameters: the penalty for starting a new segment and a minimum segment length defined by the number of supporting SNPs. That means a new segment always starts with a heterozygous SNP and the startpoint may not be precise as the parameters require sufficient evidence of a step in the BAF signal before a new breakpoint is added. Finally, Battenberg does not use the logR for segmentation, which means a region in which both alleles are gained are difficult to detect as the BAF does not change.

I have therefore added the option to incorporate pre-defined breakpoints into the segmentation procedure (see Fig. 2.2 for an example). This allows for inclusion of breakpoints with base-pair resolution from SV calling. The approach starts with pre-segmenting the genome with the supplied breakpoints. It assumes the breakpoints are clean and therefore performs no further filtering. Then PCF is performed in each pre-segment to detect further breakpoints not covered by a SV, such as a chromosome arm event. However, not every structural variant constitutes a copy number change (inversions for example) and the SVs can therefore lead to spurious segments. A segment merging step is therefore added that formally tests the BAF and logR of each adjacent pair of segments through a t-test and merges the pair if the BAF and logR are not significantly different or when the major and minor allele of both segments have the same clonal values. An exception is made for segments between which there is a gap of 3Mb or larger. The assumption is made that there is either missing data or a centromere between the segments and as there is no data we make no call.

2.2.7 GC content correction

Coverage of sequencing data can be affected by artefacts that manifest themselves as a wave pattern across the genome (Diskin et al., 2008). These artefacts are correlated with local GC content and can be corrected for by a regression approach (Benjamini and Speed, 2012; Diskin et al., 2008). I observed that a substantial set of tumours reported on in this thesis are affected by this problem. Fitting an initial copy number profile was impossible as it yielded whole chromosome homozygous deletions where the profile looked generally correct for other chromosomes (Fig. 2.3, with details of chromosome 8 in Fig. 2.4). These deletions would be surprising given that about 10% of genes are thought to be essential for cell function (Wang et al., 2015), which makes it likely that every chromosome contains at least one gene required for cell survival. I have therefore implemented an approach for Battenberg that corrects the relative tumour coverage (logR) for wave patterns.

Similarly to the method implemented in ASCAT, the GC content correction function considers each SNP given in the input as the centre-point of a window. The GC content for window-sizes varying from 25kb to 10Mb have been pre-calculated. Similarly to ASCAT, we consider two window sizes to correct for high and low frequency waves. After calculating correlations the data with the GC content of the logR data we select a window $<$ 1Mb



(a) Without SV breakpoints

(b) With SV breakpoints

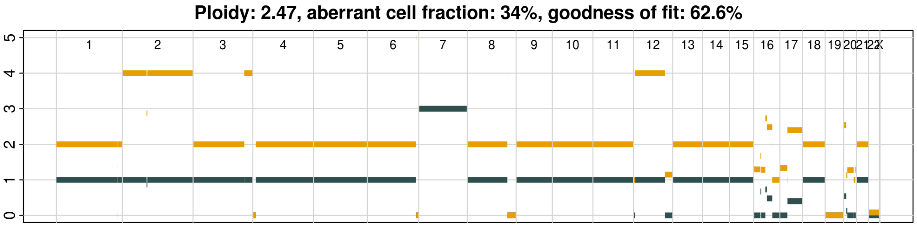
Fig. 2.2 The above figures show the segmented data with a copy number fit on a chromosome that consists of many small segments. The top plot in both figures contains the GC corrected raw logR data (grey dots) with the segment boundaries overlayed (vertical lines). The bottom plot contains the BAF with fit segments overlayed (green represents clonal copy number, while red represents subclonal). **a**) The fit without inclusion of SV breakpoints misses a series of consecutive breakpoints around 200Mb. **b**) After inclusion of the SV breakpoints (green vertical lines) Battenberg is able to call all visible segments on this complex chromosome.

(denoted as $w < 1$) and one $>= 1$Mb ($w >= 1$) and perform regression on a model that allows for both a linear and a non-linear effect of GC content:

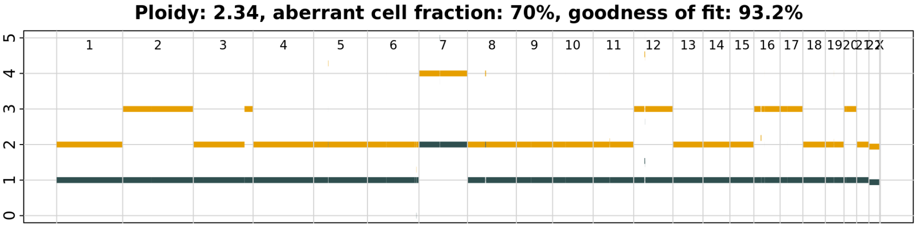
$$l = G_{w<1} + G_{w>=1} + G^2_{w<1} + G^2_{w>=1} \tag{2.12}$$

where G is the precalculated GC content data. The residuals (expected logR) are then taken as the corrected logR value and saved for use further down the pipeline.

This approach corrects for the majority of the wave effect and has allowed a substantial number of tumours to be included in the analysis described further into this thesis. It does however not completely remove the artefacts (see Fig. 2.4b), which suggest that there are additional factors that have not yet been accounted for.


Ploidy: 2.47, aberrant cell fraction: 34%, goodness of fit: 62.6%

(a) Without GC correction


Ploidy: 2.34, aberrant cell fraction: 70%, goodness of fit: 93.2%

(b) With GC correction

Fig. 2.3 Whole copy number profile for sample SA514993, with in orange the total copy number and in dark grey the minor allele. (**a**) A copy number profile with homozygous deletions on chromosomes 4, 6, 8, 19 and 22. (**b**) The homozygous deletions disappear after correction for GC content. The purity estimate also increases, which reduces the gains on chromosomes 2 and 12 by one copy and on chromosome 7 by three copies.

2.3 Subclonal architecture inference with DPClust

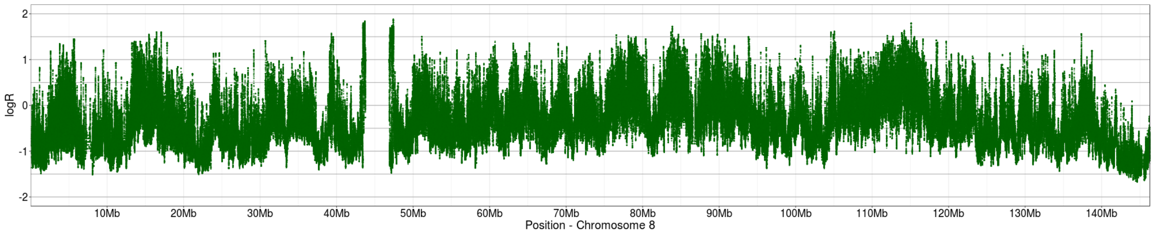
A subclone is a population of tumour cells that carry a unique subset of mutations (SNVs, indels or copy number). These mutations will appear in a similar fraction of tumour cells in the sequenced sample and can therefore be used as a marker of the population. By clustering the mutations, one can infer the existence of a subpopulation and therefore the subclonal architecture contained within the sequencing sample.

For such an approach to work one must assume that mutations occur only once during the life time of the tumour, which is referred to as the *infinite sites assumption* (Jiao et al., 2014). For SNVs and indels that assumption holds true in general given the size of the human genome, but for copy number alterations there is accumulating evidence that the same locus can be mutated on multiple occasions (Jamal-Hanjani et al., 2017).

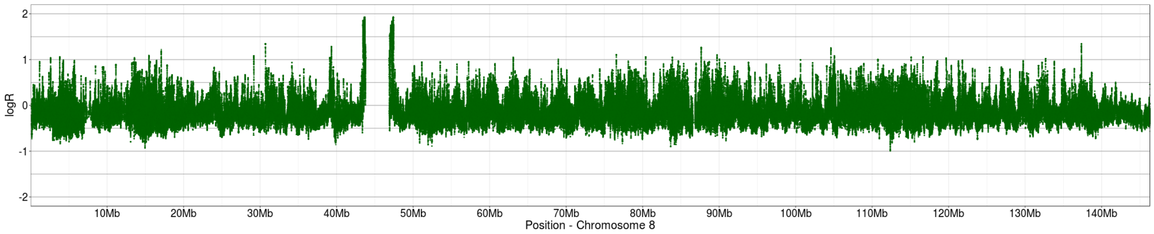
This section describes the approach implemented in the DPClust software package.

2.3.1 Estimating cancer cell fractions

To infer the subclonal architecture of a tumour one must first obtain an estimate of the fraction of tumour cells (cancer cell fraction, CCF) for each mutation, which can be inferred from VAFs of SNVs. Massively parallel sequencing results in short reads, which can then be



(a) Without GC correction



(b) With GC correction

Fig. 2.4 LogR data of chromosome 8 from sample SA514993. Smoothing was performed by applying a running median with a window-size of 101 SNPs to make the signal more visible at this scale. (**a**) Raw logR before GC correction shows a long wave pattern with a varying frequency. The homozygous deletion visible in Fig. 2.3a is situated at about 140Mb where the logR is clearly sloping downwards. (**b**) The big steps in logR are removed after correcting for GC content. The sloping at around 140Mb is reduced dramatically, now stopping Battenberg from calling a homozygous deletion (Fig. 2.3b). A light wave pattern is still visible, suggesting further improvements can be made.

aligned to a reference genome, followed by SNV calling. Both the variant and reference alleles of an SNV are supported by a number of reads, r_{mut} and r_{ref} respectively. The VAF of SNV i, f_i, can straightforwardly be calculated as:

$$f_i = \frac{r_{mut,i}}{r_{mut,i} + r_{ref,i}} \tag{2.13}$$

However, mutation clustering to identify (sub)clonal populations cannot be performed directly using VAFs, as copy number changes impact allele frequencies. Fig. 2.5 shows four SNVs in a sample that consists of 80% tumour cells and 20% normal cells. SNV 1 is clonal and occurs in a region with a normal diploid copy number state. This mutation is therefore carried by approximately half the reads that represent tumour DNA. SNV 2 is subclonal and also occurs in a region of normal diploid copy number. As both copy number and normal cell contamination are equal for both SNV 1 and 2, their allele frequencies are directly comparable and proportional to the fraction of tumour cells by which they are carried. SNV 3 falls into an area that was subclonally lost. As the subclonal loss has occurred on the other allele, this SNV's VAF is increased compared to SNV 1. SNV 4 is clonal, falls into an

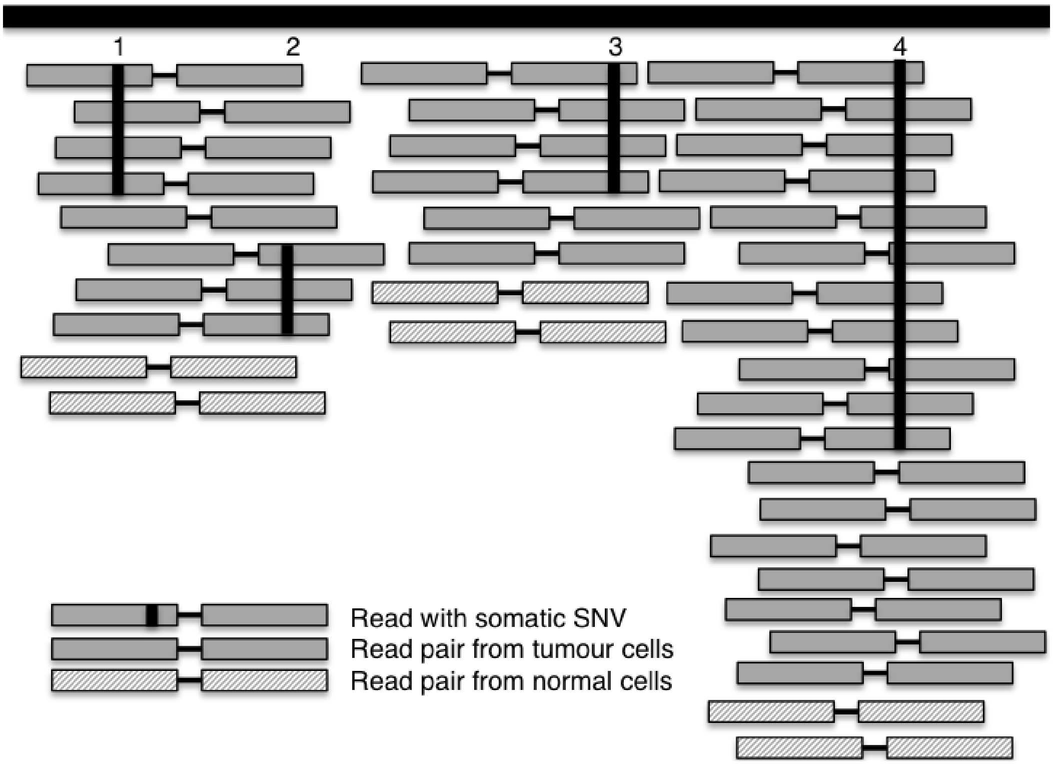


Fig. 2.5 Allele frequencies of SNVs must be transformed to Cancer Cell Fractions, accounting for copy number changes, before they can be clustered to identify subclonal populations. This illustration shows 4 SNVs in different (sub)clonal populations and in regions with different copy number states, to illustrate this principle. SNVs 1 and 2 are clonal and subclonal respectively and appear in a non-aberrated copy number state. SNV 3 coincides with a subclonal deletion, with the SNV falling on the retained allele (i.e. the other allele is subclonally deleted). SNV 4 has occurred before a gain and is therefore carried by two chromosome copies. Even though SNV 1, 3 and 4 are clonal, their allele frequencies differ due to copy number alterations.

area that is clonally gained and is on the gained allele. Its VAF is therefore higher than that of SNV 1. If these SNVs were clustered in VAF space, SNVs 3 and 4 would be mistaken for evidence of additional mutation clusters, while they in fact belong to the clonal cluster.

This example illustrates that the copy number state of an SNV, also called its multiplicity, is key to understanding VAF distributions of mutations. Estimating the multiplicity of an SNV is challenging, as it requires establishing the copy number state of a single base. Copy number callers often estimate copy number states for large stretches of DNA, which might not accurately represent the copy number state exactly at the base of the SNV. To assist with resolving this issue, it is helpful to consider the product of mutation multiplicity m_i of a mutation i and its cancer cell fraction CCF_i:

$$u_i = CCF_i m_i \tag{2.14}$$

Let us consider the properties of u_i. A clonal SNV will have a CCF of 1.0 (i.e. 100% of tumour cells) and in each cell the number of chromosome copies, m_i is an integer. It follows from the above equation that for clonal mutations $u_i \geq 1$. A subclonal mutation has a CCF less than 1.0 (for example 0.4, or 40% of tumour cells) and can only be carried by a single chromosome copy (unless also affected by a subclonal CNA), therefore m_i = 1. It follows that $u_i < 1$ for subclonal mutations. We can use these observations to obtain m_i from u_i:

$$m_i = \begin{cases} |u_i|, & \text{if } u_i \geq 1 \\ 1, & \text{if } u_i < 1 \end{cases} \tag{2.15}$$

Furthermore, u_i can be written as a function of the fraction of tumour cells ρ with a total number of chromosome copies in tumour cells at locus i, $n_{tot,t,i}$, and a fraction of normal cells 1-ρ with a total number of chromosome copies in normal cells at locus i, $n_{tot,n,i}$:

$$u_i = f_i \frac{1}{\rho}[\rho n_{tot,t,i} + (1-\rho) n_{tot,n,i}] \tag{2.16}$$

In the formula above, ρ and $n_{tot,t,i}$ can be obtained through copy number analysis, f_i can be calculated from r_{mut} and r_{ref} using Eq. 2.13, and the $n_{tot,n,i}$ values are considered known (typically 2). This equation therefore provides us with a way to calculate u_i and by extension to obtain the multiplicity of the SNV.

SNV 1 in Fig. 2.5 for example is clonal and has 4 reads reporting the variant and 6 reporting the reference allele. The purity is 0.8 (80% of total cells are tumour cells) and the total copy number of both the tumour and normal cells is 2. Its u_i therefore becomes:

$$\frac{4}{4+6} \times \frac{1}{0.8} \times [0.8 \times 2 + 0.2 \times 2] = 1.000 \tag{2.17}$$

Which translates into a CCF of 1.0 via Eq. 2.15. While for SNV 4 it yields:

$$\frac{11}{11+9} \times \frac{1}{0.8} \times [0.8 \times 3 + 0.2 \times 2] = 1.925 \tag{2.18}$$

Which also translates into a CCF of 1. SNV 4 illustrates that u_i must be rounded to obtain the multiplicity of a clonal SNV. It differs slightly from the expected value 2 because of variability in the number of reads due to limited sequencing depth. A similar mutation with 12 variant reads out of 20 would lead to an estimate of 2.100.

The accuracy of the multiplicity estimate in practice depends on the accuracy of the VAF and local copy number. Slight deviation in the VAF due to read sampling can result in minor deviation of the multiplicity estimates, as illustrated in the example above. Incorrect copy number profiles may also result in large errors if, for example, the CNA profile has been called diploid instead of tetraploid. Ambiguity in estimating whole genome duplications is a difficult problem in copy number analysis. If a copy number profile is erroneously called as diploid then SNVs carried by two chromosome copies will be estimated to have a multiplicity of 1, while SNVs on 1 chromosome copy will become subclonal as they appear to be on 0.5 copies (e.g. exactly half of tumour cells). The CCF space will therefore show an SNV cluster at exactly 0.5, while the copy number profile may also contain subclonal CNAs at exactly 50% of tumour cells. The uncertainty may be mitigated through the application of a key assumption: a CNA profile is thought to be in its normal state (diploid) unless substantial evidence of a whole genome duplication is available (i.e. the most parsimonious diploid state is assumed unless there is evidence otherwise). However in rare cases, when whole genome duplications occur late and are not followed by other copy number alterations, they leave no traces in the data and it is mathematically impossible to infer from the data available that they occurred.

We now have obtained a series of formulas to calculate CCF from a VAF and copy number profile. First, we obtain u_i through Eq. 2.16 and then calculate the multiplicity and CCF using Eqs. 2.15 and 2.14 respectively.

Finally, we adjust the multiplicity to address SNVs that may appear subclonal due to a subclonal deletion. In these cases it is unknown whether the SNV occurred first and was then deleted in a fraction of cells, or the SNV occurred after the deletion. It is important to account for such subclonal deletions (e.g. by appropriately adjusting multiplicity estimates), and ensure that these subclonal deletions do not result in the inference of spurious subclonal populations.

2.3.2 Filtering

Not all mutations that are provided as input are clustered. Mutations for which there is no copy number are removed because it is not possible to estimate their CCF value. Mutations in regions with fewer than 4 reads total coverage are also removed. Mutations in regions identified with localised somatic hypermutation (*kataegis*) are also filtered out. Short read alignment in regions with kataegis can be difficult because of the many reads carrying one or multiple variant alleles and the fact that kataegis is often observed close to a SV breakpoint. These mutations are removed to reduce the opportunity for a spurious cluster to be inferred.

2.3.3 Algorithm

DPClust clusters SNVs with a similar CCF, derived from VAF values as described in the last section. However, the VAF of a SNV - and therefore also its CCF - can be a relatively coarse measure and is a function of local sequencing depth, which should be taken into account when clustering SNVs. For example, if the SNV falls in a region of diploid copy number with a depth of 20 reads in a sample with 50% tumour cells, its CCF changes by 0.2 when a variant read is added or removed (e.g. 3 mutant reads correspond to a CCF of 0.6, while 4 mutant reads correspond to a CCF of 0.8). If the same SNV is sequenced to 80X depth, one additional variant read would change the CCF by only 0.05. Tumours are often sequenced at 30X average coverage or higher, but this coverage is not constant across the genome. Due to this discrete sampling of mutant and non-mutant reads, and the variability of the sequencing depth, CCF estimates of mutations from specific (sub)clones will show a distribution of values. For example, clonal mutations will display a range of CCF values around 1.0 (Fig. 1.1C).

A suitable error model can account for this variability. The number of variant reads can be seen as the number of successes of n independent coin tosses, where n is the total read depth. The number of successes (variant reads) can therefore be modelled through a binomial distribution with r_i the number of reads reporting the variant at location i, $r_{tot,i}$ the total depth at location i and $r_{tot,i}$ the probability of observing a mutant read:

$$r_i \sim \text{Bin}(r_{tot,i}, p_i) \tag{2.19}$$

Both r_i and $r_{tot,i}$ are observed in the data. p_i can be considered the product of two factors: the proportion of reads one expects to see if the mutation is fully clonal, ζ_i, and the true fraction of tumour cells carrying the mutation π_i:

$$p_i = \zeta_i \pi_i \tag{2.20}$$

ζ_i can be calculated from the tumour purity and the copy number state of the locus, as detailed above. Take for example a clonal SNV in a balanced diploid copy number region in a sequencing sample consisting of 80% tumour cells. The SNV is heterozygous and therefore expected to be carried by half the reads that represent tumour DNA. The expected proportion of reads is therefore 0.5 * 0.8, i.e. 0.4. If the region has three copies and the SNV is carried by two copies, one expects two thirds of the reads representing tumour DNA to be carrying the variant allele, making the expected fraction 2 * 0.8 / (3 * 0.8 + 2 * 0.2), i.e. 0.57.

The key estimate in subclonal reconstruction is the true fraction of tumour cells that are carrying mutation i, π_i. Many methods (Deshwar et al., 2015; Jiao et al., 2014; Landau et al., 2013; Roth et al., 2014) use a Dirichlet Process, which models subclonal fractions as:

$$\pi_i \sim \mathrm{DP}(\alpha P_0) \tag{2.21}$$

where $DP(P_0)$ is a Dirichlet Process with a given probability distribution P_0 and a dispersion parameter α. A realisation of a Dirichlet Process (DP) can be seen as a distribution over a (possibly) infinite sample space, or alternatively as a sampling from an unknown number of unknown distributions (Dunson, 2010). This approach allows for co-estimating both the number of contributing distributions K (the number of cellular populations) and their properties (fraction of tumour cells and number of mutations they contain). The observed sampling represents of the (possibly) infinite number of distributions and can be used to estimate K (i.e. cellular populations) through the stick-breaking representation (Sethuraman, 1994). Stick-breaking implies that the real probability distribution P can be expressed as follows:

$$P = \sum_{h=1}^{\infty} \omega_h \pi_{\theta_h}\,, \qquad \theta_h \sim P_0 \tag{2.22}$$

where π_{θ_h} is a location in CCF space and ω_h represents the probability weight of cluster h

$$\omega_h = V_h \prod_{l<h} (1 - V_h) \tag{2.23}$$

with

$$V_h \sim \mathrm{Beta}(1, \alpha) \tag{2.24}$$

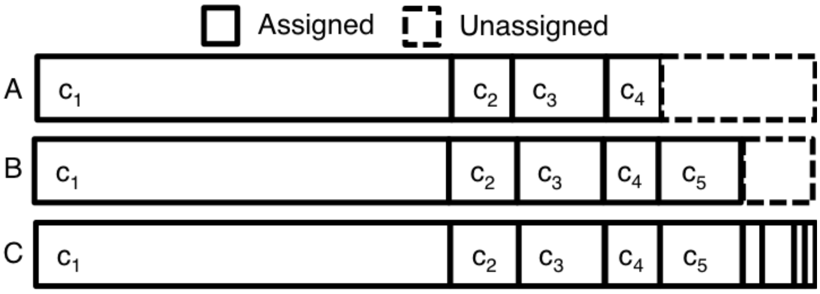


Fig. 2.6 The stick-breaking property of the Dirichlet Process is used to estimate the number of mutation clusters in the data. For each mutation, a stick of arbitrary length is broken into randomly sized bits that represent a cluster. At point A, breaks have been introduced, corresponding to clusters c1-c4. B shows the stick after introducing break 5, while C shows the completed stick-breaking procedure. The size of each broken part represents the weight associated with a cluster and influences the mutation assignments, where a high weight makes it more likely that a mutation is assigned to that cluster. These weights are updated after probabilities for each cluster have been obtained for each mutation, eventually converging on a solution.

The V_h represent parts of a unit length stick that are iteratively broken off from the remaining stick. The V_h get increasingly smaller as more parts are broken off, providing a discrete representation of an infinite space.

Fig. 2.6 symbolizes the stick at various iterations of the stick-breaking procedure. Fig. 2.6A and 2.6B show the stick after 4 and 5 breaks respectively, while Fig. 2.6C shows it after completion. Each substick represents a fraction of the total weight (number of SNVs) of a cluster and can be assigned a CCF through resampling using the assigned SNVs. Then for each SNV and for each substick, a likelihood can be calculated representing the probability that that SNV is generated by that substick, taking the characteristics of the SNV, the stick location and its associated weight into account. After assigning all SNVs, the weights are updated such that they reflect the overall likelihood across SNVs.

The DP models an appropriate number of clusters because the assigned SNVs (influenced by the cluster weight) are used to resample the cluster CCF and the weight represents the fraction of total SNVs assigned to the cluster. By repeating this process over many iterations, the weight and SNV assignments will accumulate in certain locations that correspond to the estimated clusters. Therefore, the DP has the advantage that the number of clusters does not have to be specified *a priori*, making it ideally suited to this problem.

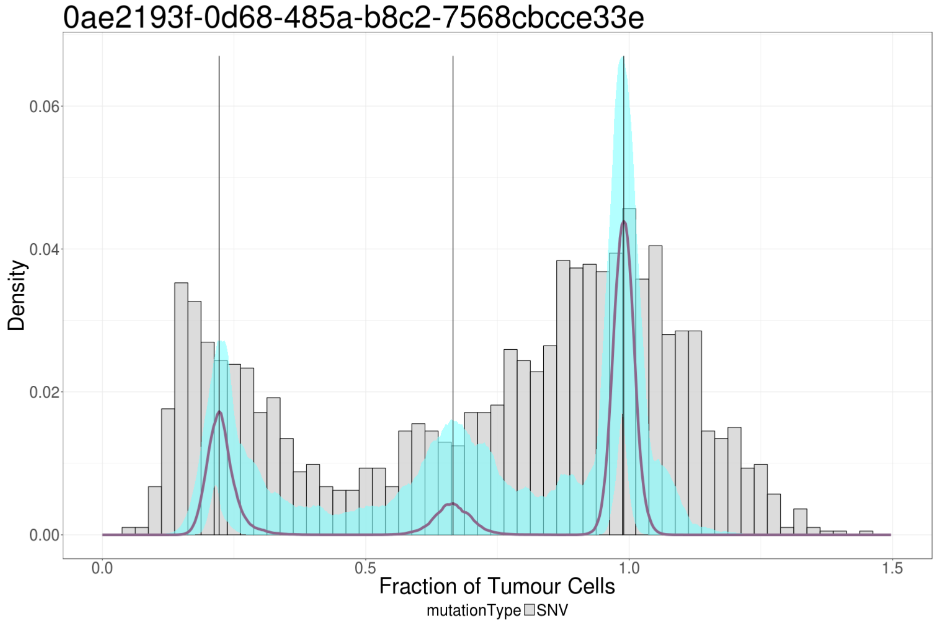


Fig. 2.7 Main output figure from a DPClust run. The grey histogram represents the input SNV data. In front of the histogram is a density line in purple with a turqoise 95% confidence interval. The density line is built up by carefully recording where each SNV is assigned throughout the MCMC iterations. The number of clusters is obtained by obtaining all peaks in the density (vertical black lines). To assign mutations to clusters, first the local minimum density between each pair of cluster locations is obtained. Mutation assignment probabilities are then obtained by going back to the MCMC iterations to record how often each mutation would have been assigned to the final clusters if those were the clusters available at that iteration. The mutation is finally assigned to the cluster with the highest number of assigments.

2.3.4 Post-processing

After completing the MCMC iterations we aim to obtain three estimates: (1) An estimate of the finite number of distributions (cell populations), K, that are present in the input data, (2) the proportion of tumour cells that each population consists of (CCF_k) and (3) likelihoods of each SNV belonging to each population. The number of cell populations K is determined by finding peaks in the posterior weight density Fig. 2.7. In each iteration j the stick-breaking procedure assigns a weight $\omega_{k,j}$ to each cluster that represents its size and the cluster has a $CCF_{k,j}$. Over many iterations weight accumulates in the CCF space, where a large amount of weight corresponds to a high likelihood of the existence of a mutation cluster. We then obtain an estimate of the number of clusters K (cell populations) by obtaining all local maxima in the weight density.

With the K clusters and their locations (CCF_k) established, SNVs can be assigned to clusters. We first establish the CCF area covered by each $k \in K$ by finding the CCF location between each pair of neighbouring clusters that corresponds to the minimum density. The minimum density on either side of a cluster represents its upper and lower CCF bound. Probabilities of a mutation belonging to a cluster are then established by accounting how often a SNV would have been assigned to each k throughout the MCMC iterations. Finally, small clusters smaller than 30 SNVs are removed.

2.3.5 Extension to multi-sample cases

Obtaining multiple samples from the same donor allows for extraction of more detailed subclonal reconstructions. These datasets can consist of multiple tumours taken from different sites (e.g. multiple primary sites, primary and metastasis), multiple samples from the same tumour or multiple samples from the same cancer that represent different time points (e.g. primary and relapse).

Multiple sampling strategies provide a series of advantages. Consider a tumour that has two subclones that each comprise 20% of tumour cells. A single sample analysis will not be able to separate the two groups of mutations as both occur in 20% of tumour cells. But if in another sample the cellular prevalence of the two subclones does vary, one can separate the two groups of mutations. In addition, having multiple samples may help resolve tree topologies. In single sample cases it is often not possible to resolve phylogeny, as more rare subclones may be placed in multiple positions in the tree. By applying the pigeonhole principle across the samples for each subclone, one can often rule out various configurations where a subclone may fit in multiple places in one sample, but not the other. Finally, with multiple sampling strategies, mutations with low allele fractions in one sample can be confirmed (or detected) in another sample where they have higher allele fractions due to higher tumour purity or higher CCF.

Approaches based on a DP can be extended into multiple dimensions (Bolli et al., 2014). The read counts across samples can be modelled as independent draws from n Binomial distributions.

$$\begin{aligned} r_{i,1} &\sim \mathrm{Bin}(r_{tot,i,1}, p_{i,1}) \\ r_{i,n} &\sim \mathrm{Bin}(r_{tot,i,n}, p_{i,n}) \end{aligned} \tag{2.25}$$

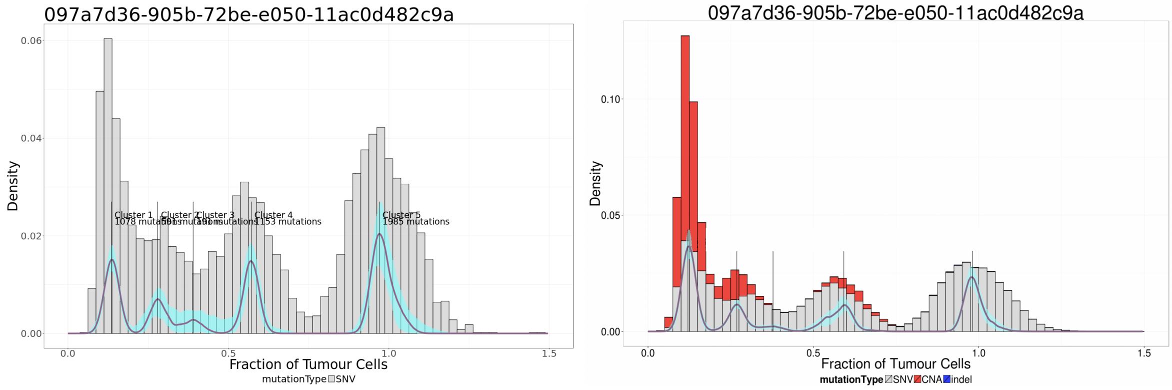
The stick-breaking procedure is performed across the samples where a cluster has a single weight (representing the number of mutations), but a separate location in each of the samples.

Posteriors are obtained across samples by calculating the total probability for each mutation for each cluster under consideration. Finally, the DP can be used to jointly perform clustering and infer phylogenetic relationships between the clusters by interleaving two stick-breaking procedures (Ghahramani et al., 2010).

Several methods for single sample analysis, including PyClone (Roth et al., 2014), SciClone (Miller et al., 2014) and CloneHD (Fischer et al., 2014), can be used to analyse multiple samples. Furthermore, automated tree inference has been implemented in PhyloSub (Jiao et al., 2014) and extended to include SNVs in copy number aberrant regions in PhyloWGS (Deshwar et al., 2015).

2.3.6 Co-clustering of indels and CNAs

Up until now CNAs have only been used to adjust the allele frequency of point mutations. CNAs can also be used to identify cellular populations. The Battenberg algorithm estimates CCF values for each subclonal alteration and it is therefore possible in principle to reconstruct the subclonal architecture through CNAs only, or jointly with SNVs. However, unlike SNVs, there are often far fewer subclonal CNAs measured, which leads to a sparser CCF space and therefore to a reconstruction with less detail. Jointly clustering SNVs and CNAs is preferred



(a) SNV only reconstruction (b) SNV, indel and CNA combined reconstruction

Fig. 2.8 Subclonal reconstruction on tumour SA6164 (also known as PD4120 and 097a7d36-905b-72be-e050-11ac0d482c9a) using (**a**) only SNVs and (**b**) SNVs, indels and CNAs. There are relatively few indels (blue bars) measured in this tumour, but those available are automatically assigned to mutation clusters. The addition of CNAs (red bars) has a more profound effect, but it does not alter the inferred subclonal architecture substantially. The CNAs provide additional support for clusters 1, 2 and 4 (counted from the left edge of the figure).

as the SNVs will anchor the cluster locations, while CNAs are then assigned to their most likely cluster.

To include CNAs in the clustering process they must be encoded with properties that the DPClust algorithm can understand. The CNA is therefore encoded as an artificial SNV, termed pseudo-SNV. But with a single pseudo-SNV representation it is not immediately clear how many reads should support the pseudo-variant and pseudo-wild-type alleles. A very high coverage could represent a large CNA event, but it would create an artificially high amount of confidence in the VAF, while low read counts do not reflect the size of the CNA events accurately. It is also not directly clear how to balance the evidence between SNVs and CNAs such that one does not dominate the other.

To resolve this issue I encode the CNAs as groups of pseudo-SNVs. First the mutation rate of the tumour is calculated using the measured SNVs. Each CNA covers a certain area of the genome and the equivalent number of mutations that a stretch of DNA would contain given the mutation rate is calculated. Each pseudo-SNV is then assigned a number of mutant and wild-type reads such that the CCF of the SNV corresponds to the CCF of the subclonal CNA.

To mimick read sampling variability the total number of reads are drawn from a Poisson distribution that takes as input the exact depth and the mutant reads are drawn from a binomial that takes the inexact depth and the exact probability of success mandated by the CCF of the CNA. By introducing read sampling variability we transform the pseudo-SNVs into an independent estimate of the CCF of the CNA. The exact total depth is set to either the median depth of all measured SNVs or, if the CNAs cannot be represented by pseudo-SNVs due to insufficient reads per chromosome copy, by 90 reads.

The Battenberg algorithm also provides a measure of confidence in the CCF of each subclonal CNA in the form of a standard deviation on the CCF obtained through bootstrapping. The tighter the standard deviation, the more confident we are in the accuracy of the CCF estimate. The binomial can be used to take this certainty into account by increasing or decreasing the number of trials undertaken. If the number of trials is lower the number of successes given the same probability of success will be more coarse, giving rise to a wider distribution. The total depth is therefore scaled down by the amount of uncertainty, which is represented by the standard deviation. As the standard deviation for the most certain cases is close to 0 we add 1 to it before scaling down the total depth. Finally, the copy number status of each pseudo-SNV is irrelevant and is set to 1 chromosome copy out of 2.

It is important to balance the evidence obtained from SNVs and CNAs such that one does not dominate the other. I have implemented the balancing using the following observation: tumours often have more SNVs than CNAs and each subclonal SNV or CNA is an independent measure of the CCF of a subpopulation of cells. With more samples the estimate of the sampled value becomes more accurate, which gives SNVs an advantage. CNAs however stretch much larger regions of the genome. The evidence is therefore balanced such that the CNAs can provide support for an (extra) cluster, but not dominate the CCF space. For this reason clonal CNAs are represented by a single pseudo-SNV and assigned to the cluster to which the pseudo-SNV is assigned afterwards. Fig. 2.8 shows an example run on the PD4120 tumour that was first described in Nik-Zainal et al. (2012a).

Co-clustering of indels is performed by including the indels as pseudo-SNVs into the input to DPClust. CCF estimates are obtained from the number of reads carrying the variant and wild-type using the procedure described for SNVs. That approach assumes the VAF estimates of the indels are recalibrated by local assembly. Due to alignment difficulties around indels the raw VAF values are often an underestimate. By assembling the local sequence and local realignment of the reads a less biased VAF estimate can be obtained that is useful for subclonal architecture inference.

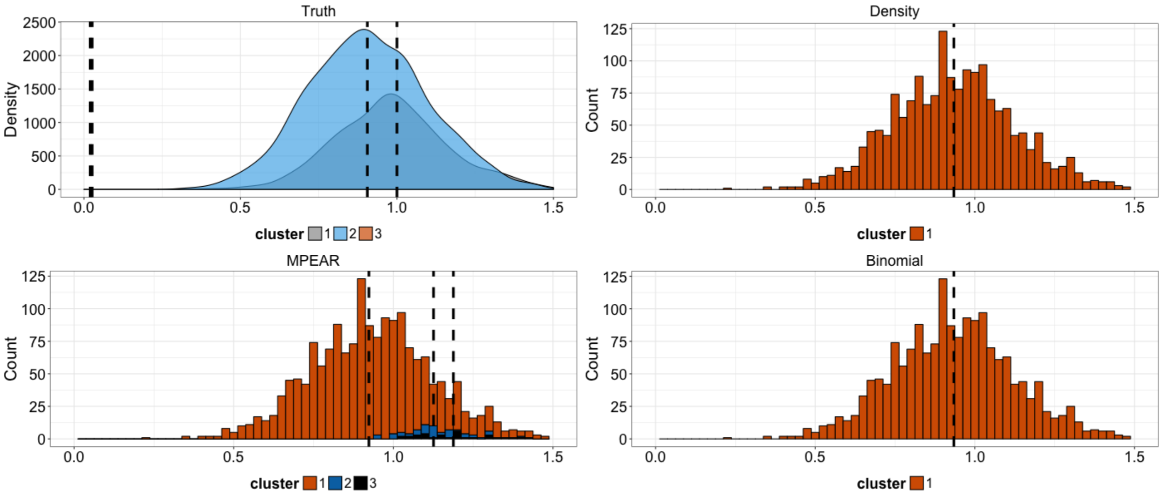


Fig. 2.9 The MPEAR cluster finding approach often finds many small mutation clusters. In this randomly generated example the truth (top left) contains three clusters: A small clone in grey (behind the blue density), a large subclone in blue and a large subclone in orange that falls below the detection limit given this tumours' purity, ploidy and coverage combination. The default density (top right) and binomial assigment (bottom right) approaches find a single cluster in between the major subclone and the clone, effectively merging the two clusters. The size of the clone and its close proximity to the subclone makes it impossible to disentangle the two clusters. MPEAR (bottom left) returns two small additional superclonal clusters in an incorrect position and therefore often requires an additional merging step, more often than the default density approach.

2.3.7 Alternative post-processing steps

In search for increased sensitivity to real clusters I have implemented alternative strategies for obtaining the number of clusters and their contents from the MCMC chain and developed additional procedures for assigning mutations to clusters. The current assignment approach is prone to find small clusters that need to be filtered from the output. It is not easy to come up with a list of criteria that capture these clusters without removing real results. The current implementation of the filtering step removes all clusters below 30 mutations. Often these clusters appear at the end of the data histogram, in the far tail of a large mutation distribution. As the MCMC chain progresses it places a cluster where the large mutation distribution belongs, but depending on its exact placement it leaves the need to explain the far tail with an extra cluster in some iterations. This process is part of the mixing required by a clustering method and it allows the chain to find evidence for extra clusters, but it has the side effect of yielding spurious small clusters. I have therefore attempted to find alternative methods for obtaining clusters that do not have this property. However, none of these new strategies yielded an improvement in performance from evaluation on real and simulated data and have therefore not been used in production.

A new method for obtaining clusters is using hierarchical clustering of mutations followed by a cut of the tree using the MPEAR (maximal posterior expected Rand index) criterion

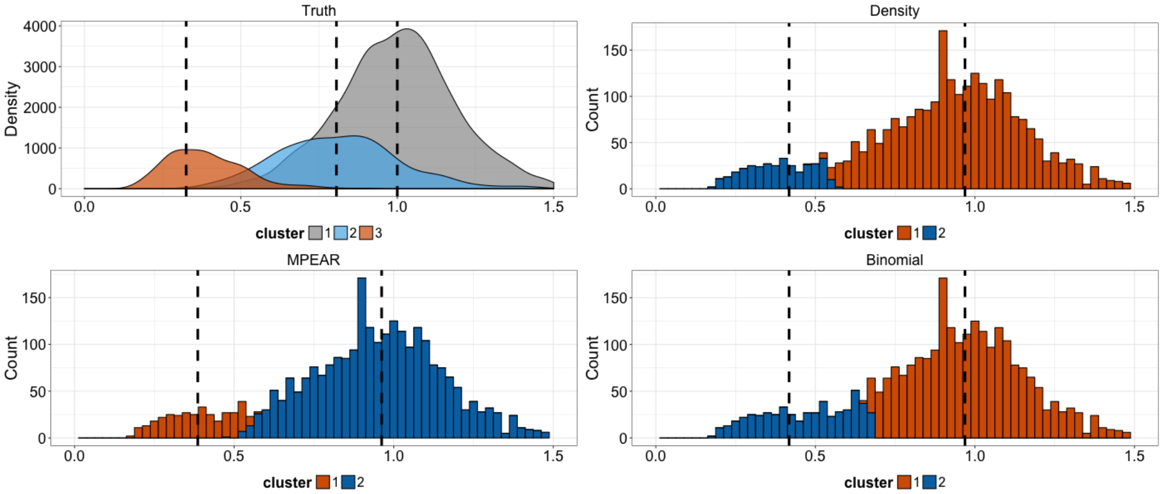


Fig. 2.10 The approach that assigns mutations using the most likely cluster based on the cluster that yields the maximum binomial probability often has the effect of assigning a mutation to its closest cluster. In this example there are three mutation clusters (top left) and all approaches find only two. Both the default density and MPEAR approaches underestimate the size of the subclone slightly (top right and bottom left), while the binomial approach estimates it to be nearly twice the actual size (bottom right).

(Fritsch and Ickstadt, 2009), also used by PyClone (Roth et al., 2014) and BitPhylogeny (Yuan et al., 2015). For this approach I first build a mutation similarity matrix through co-assignment probabilities. Each cell contains the probability that a pair of mutations belong to the same cluster. This matrix is build from the MCMC chain by counting how often the pair is assigned to the same cluster and dividing by the total number of iterations, after excluding the burn-in. After performing hierarchical clustering the MPEAR criterion is applied to k cuts of the tree, with $k < (\lceil \text{ mutations/8 } \rceil)$. The cut that yields the maximum score is chosen as the optimum solution. This approach however yields more spurious clusters, it often splits clear existing clusters found by the DPClust default approach into multiple (Fig. 2.9), and the co-assignment matrix cannot easily be constructed for large numbers of mutations.

I have also experimented with an alternative mutation assignment approach. The DPClust default approach is to calculate likelihoods of a mutation belonging to a cluster by counting how often the mutation would have been assigned to that cluster if it had been available in each MCMC iteration. That tends to yield very high probabilities of one cluster, which may not reflect the uncertainty correctly. I therefore wondered if calculating the binomial likelihood would provide a more accurate reflection:

$$\ell_{i,c} = r_{mut,i} \log \mathbf{E}(f_{i,c}) + r_{ref,i} \log(1 - \mathbf{E}(f_{ref,c})) \tag{2.26}$$

Equation 2.26 contains the total number of reads supporting the variant and reference alleles ($r_{mut,i}$ and $r_{ref,i}$) and the expected allele frequency ($\mathbf{E}(f_{ref,c})$ if the mutation belongs to cluster c, calculated using Eq. 2.16. The binomial likelihood however effectively works as assigning the mutation to its closest cluster and therefore tends to overestimate the size of small clusters (Fig. 2.10). It is also a point estimate and does not take into account the cluster size, which the default DPClust assignment approach does. The mutation assignment approach used by Gerstung et al. (2017) calculates beta-binomial probabilities with the inclusion of the cluster size and may be an interesting option in the future.

2.3.8 A downsampling strategy

Clustering a large number of mutations can take a very long time with MCMC based approaches. DPClust uses Gibbs sampling, which means it has to execute a routine for all mutations in every iteration. To improve on runtime and resource usage I have implemented a downsampling strategy that samples mutations and is capable of assigning the mutations not used for clustering afterwards. The routine performs uniform sampling of a specified number of mutations. Large clusters therefore have a higher chance of being sampled from over small clusters, keeping their relative sizes intact. For every mutation not used during clustering I find the mutation with the most similar allele frequency (referred to as its *mate*) that is clustered. By using the allele frequency the selection process is biased towards finding a mate in a similar copy number configuration. After clustering the mutation is assigned to the same cluster as its mate.

I have considered alternative strategies. Selecting copy number segments and using only the mutations in those genomic regions for clustering, but that does not leave fine-grained control over the number of sampled mutations. A biased sampling approach was also considered. It operated by first creating bins across the CCF space and then sampling equally from each bin. That approach changes the shape of the cluster distributions, which detriments the ability to correctly identify clusters. The idea was to perform the biased sampling a number of times and then combine the results from multiple MCMC runs. But preference was given to the unbiased selection due to its simplicity.

Downsampling initially started with 5,000 mutations, which affects nearly half of the tumours reported in this thesis. Later the number of sampled mutations was scaled up to 50,000 after various performance improvements had been implemented, which only affects 134 tumours reported on in this thesis.

2.4 Automated post-hoc tree building

For practical applications it is useful to have an overview of the possible trees that can be built from a given subclonal reconstruction. Nearly all data that I've worked with consists of single sample cases where the tree is difficult to derive, often multiple options are possible and multiple, disjoint, low CCF clusters cannot be disentangled. But for multi-sample cases it is informative and the tree represents the evolutionary story that links the multiple samples together.

I have therefore developed a procedure that builds all possible trees using the DPClust output, which operates regardless of the number of samples. First it classifies each pair of mutation clusters into categories that denote the possible pair-wise relationships. Then the classification is used to iterate over all possible trees, which are provided as a tree structured figure.

2.4.1 Cluster-pair classification

Clusters a and b can have the following relationships: (1) The CCF of a can be strictly greater than b, (2) it can be greater or equal than b, (3) it can be equal, (4) smaller or equal, (5) strictly smaller or (6) it can be unknown. Pairs of clusters are classified into these categories by first establishing the support for each cluster from the MCMC iterations and then sampling pairs of mutations to establish per category.

The classification procedure starts by recording a mutation preferences matrix after mutations are assigned to clusters (Fig. 2.11a and b). This matrix contains a row for each mutation and a column for each cluster and cell (i,j) contains the proportion of MCMC iterations mutation i would have been assigned to cluster j if the final clusters were available.

The approach then iterates over all cluster pairs (Fig. 2.11c). When considering clusters a and b we first sample 1000 mutations from a and b separately to create 1000 mutation pairs. The sampling is performed with replacement to reduce the effect of the different sizes of clusters a and b. Probabilities are calculated by, for each mutation pair (k,l), obtaining how often mutation k is assigned to a lower CCF than mutation l and then aggregating the counts across pairs. The same procedure holds for the greater-than and equals relationships.

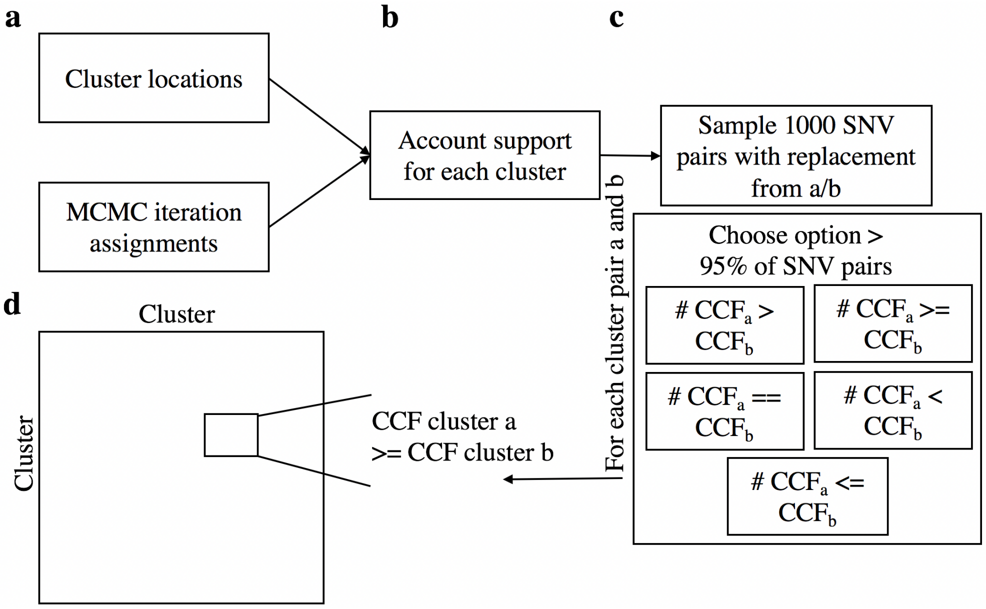

a
b
c
d
Cluster locations
MCMC iteration assignments
Account support for each cluster
Sample 1000 SNV pairs with replacement from a/b
For each cluster pair a and b
Choose option > 95% of SNV pairs
# CCF$_a$ > CCF$_b$
# CCF$_a$ >= CCF$_b$
# CCF$_a$ == CCF$_b$
# CCF$_a$ < CCF$_b$
# CCF$_a$ <= CCF$_b$
Cluster
Cluster
CCF cluster a >= CCF cluster b

Fig. 2.11 Before trees are constructed all pairs of clusters are classified into pre-defined relationships. (**a**) The procedure starts with the cluster locations and the mutation assignments during MCMC. (**b**) For each mutation it is recorded how often it would have been assigned to each cluster during the MCMC iterations if that the final cluster locations had been available, yielding a probability per cluster per mutation. (**c**) Then for each pair of clusters 1000 mutation pairs are sampled with replacement and it is counted how often the pair are assigned to the same cluster or to a different cluster, providing support for five different scenarios. (**d**) Finally, the scenario that yields support from greater than 95% of sampled SNV pairs is chosen as the final classification. If no scenario yields a 95% support the pair of clusters is classified as *unknown*.

Having obtained a probability that clusters (a,b) have a greater-than, lesser-than or equal CCF we can classify the pair into a category with a threshold at 0.95 (Fig. 2.11d). If a pair does not pass the threshold for any category, or for multiple categories, it is assigned the label unknown.

2.4.2 Tree building

The tree building process begins with creating a full inventory of all possible edges by obtaining all possible parents for each mutation cluster. The trees are then built in two phases: In the first phase all clusters that fit into a single location are placed on the tree, starting with the cluster that has the highest CCF. The pigeonhole principle is not enforced in this phase, so the phase is followed by a screening that yields an error if a the combined CCF of daughter nodes exceeds the CCF of the parent.

Then in the second phase, all clusters that fit in multiple places are considered. For each cluster, we iterate over all the possible edges involving that cluster from the inventory and over all trees obtained so far. Clusters are added to the tree in a greedy fashion on first-come first-serve basis. The pigeonhole principle is strictly enforced during this process. Some clusters may therefore not fit on the tree, which results in warnings which point to clusters that cannot coexist and warrant further investigation. If a cluster can fit in multiple places, then new trees are recorded for each configuration. This process yields a list of possible trees after all iterations are complete.

Single sample cases do not yield any warnings, because it is always possible to construct a linear tree. Multi-sample cases are more complicated however. In such cases there are two possible options to be considered: (1) The data is not clean enough and an artefact cluster is prohibiting the tree building and (2) the number of whole genome duplications is not correctly accounted for and clonal mutations have become subclonal. The output of the tree builder is useful for automated checking for violations and it will point to the clusters that are problematic.