Chapter 6

Methods for a pan-cancer study of tumour heterogeneity

6.1 Introduction

In the previous chapter I described what can be learned about a single cancer through the application of subclonal architecture and life history methods. In this chapter I introduce methods to scale the analysis up to many tumours. These methods have been applied to tumours in the International Cancer Genome Consortium (ICGC) Pan-cancer Analysis of Whole Genomes (PCAWG) project. Results obtained from the application of these methods are described in the next chapter.

The work described in this chapter is the result of a long standing collaboration that has occupied my whole Ph.D. This chapter therefore contains that is not solely mine; however this additional work is essential to make this into a complete chapter. My main contribution is the procedure that combines copy number profiles from six different methods into a robust profile. It was also my responsibility to deliver the profiles of all PCAWG tumours. The text describing the consensus breakpoints is based on text by Jeff Wintersinger for the Dentro et al. manuscript. Jeff developed the consensus breakpoints component of the consensus copy number workflow.

I have also helped lead the development of the consensus subclonal architecture procedure that combines eleven subclonal architectures into a consensus. I was involved in calibration of the eleven callers by extensive comparisons on real and simulated data, was involved in the development of a simulation data set to validate the approaches and delivered the PCAWG-wide release of the results. The brief methods described in the section about consensus subclonal architecture procedure below however contain methods developed by Kaixian Yu, Maxime Tarabichi and Amit Deschwar, which included here to create a complete story.

The chapter covers a range of methods and contains text that will serve as a basis for methods descriptions in different manuscripts, including Dentro et al. (2017, manuscript in preparation) and Yu et al. (2017, manuscript in preparation). Fig. 6.1 is inspired by a figure made by Jeff Wintersinger for Dentro et al.

6.2 Consensus copy number

ICGC PCAWG relied on a consensus strategy for SNVs, SVs, and indels. Calls made separately by algorithms that are based on different principles were understood to be high-confidence predictions. For copy number calls, we relied on a similar consensus approach, which combined results from six individual copy number callers: ABSOLUTE (Carter et al., 2012), ACEseq (Kleinheinz et al., 2017), Battenberg (Nik-Zainal et al., 2012b), CloneHD (Fischer et al., 2014), JaBbA (manuscript in preparation) and Sclust (manuscript in preparation).

Each copy number caller uses a two-step process, first segmenting the genome into regions assumed to have a constant copy number status, then determining the clonal and subclonal copy number states of each segment. Disagreement amongst copy number callers arises primarily from two factors: differences in genome segmentation, and uncertainty concerning whether a whole-genome duplication (WGD) occurred. Thus, our consensus strategy resolved both factors for each sample, allowing us to determine a consensus copy number state for much of the genome across samples.

6.2.1 Assumptions behind different copy number callers

Copy number callers differ in their implementation choices and underlying assumptions, which contribute to differences in their output (Table 6.1). The copy number callers used in this project come in two different flavours: *Event based*, that fit copy number per segment (ABSOLUTE, Aceseq, Battenberg, and Sclust), and *state based*, that aim to explain the observed data by the least number of copy number states (cloneHD). The former group are more flexible to fit different copy number states, but in principle more sensitive to noise, while the latter group is generally more conservative as it aims to minimise the number of different copy number states.

Methods also utilise different approaches perform the fitting itself. Some callers first fit total copy number to the coverage ratio data and then break that into allele specific calls

(Sclust), others perform a grid-search across a range of purity and ploidy values to jointly fit allele frequencies of heterozygous SNPs and coverage data (ABSOLUTE, Aceseq and Battenberg) or train hidden markov models separately to each type of data (cloneHD). The order of events, and how much trust is put in the allele frequency or coverage data determines how sensitive the method is to noise.

Noise levels, however, will be different between methods due to differing processing steps. Some methods perform phasing of heterozygous SNPs to reduce noise on allele frequency data (ABSOLUTE, Aceseq and Battenberg), some count reads in 1kb bins across the genome to obtain a smoothed out coverage track (ABSOLUTE, Aceseq, cloneHD and Sclust) or use coverage at single SNP positions (Battenberg). Some methods correct coverage data to remove potential wave artifacts for GC content and replication timing (ABSOLUTE and Aceseq), just GC content (Battenberg and cloneHD) or not at all (Sclust). Noise therefore does not only affect methods differently due to the fitting choices, noise itself will be different due to processing choices.

Finally, approaches differ in how subclonal copy number is considered to transform a problem of potentially millions of subclonal copy number profiles per tumour sample into a tractable problem for which a solution can be found. To do so, assumptions are made on the number of copy number states per segment (2 or 3 for ABSOLUTE, Battenberg and Sclust) and how much the separate states can differ (1 copy for Battenberg and Sclust). For the JaBbA caller there currently is neither code nor a manuscript available currently and it is therefore omitted from this comparison.

Copy number methods implementation choices and assumptions

Name	**ABSOLUTE**	**Aceseq**	**Battenberg**	**cloneHD**	**Sclust**
Event based	X	X	X		X
State based				X	
Allele counts for heterozygous SNPs	X	X	X	X	X
Binned read counts logR	X	X		X	X
logR from SNP positions			X		
Phasing of SNPs	X	X	X		
Replication timing correction of logR	X	X			
GC content correction of logR	X	X	X	X	
Assume GC artifact same in tumour and normal				X	
Assume raw data shape*				X	
Maximises genome with clonal copy number states	X	X	X	X	X
Purity/ploidy grid search fitting	X	X	X		
Hidden Markov model fitting				X	
Step-wise fitting					X
Estimates subclonal CNA	X	X	X	X	X
Fits subclonal CNA	X		X	X	X
Number of subclonal states allowed	3		2	many	2
Max. differences between subclonal states			1		1

Table 6.1 Copy number callers differ by implementation choices and assumptions, which contribute to differences between callers. The table lists, from top to bottom, basic strategy for calling alterations, how raw data is obtained, how the raw data is adjusted, approaches to fitting a copy number profile and assumptions related to fitting subclonal copy number. * = cloneHD explicitly assumes the coverage data takes on a shape of a (overdispersed) Poisson distribution and allele frequencies the shape of a (overdispersed) binomial.

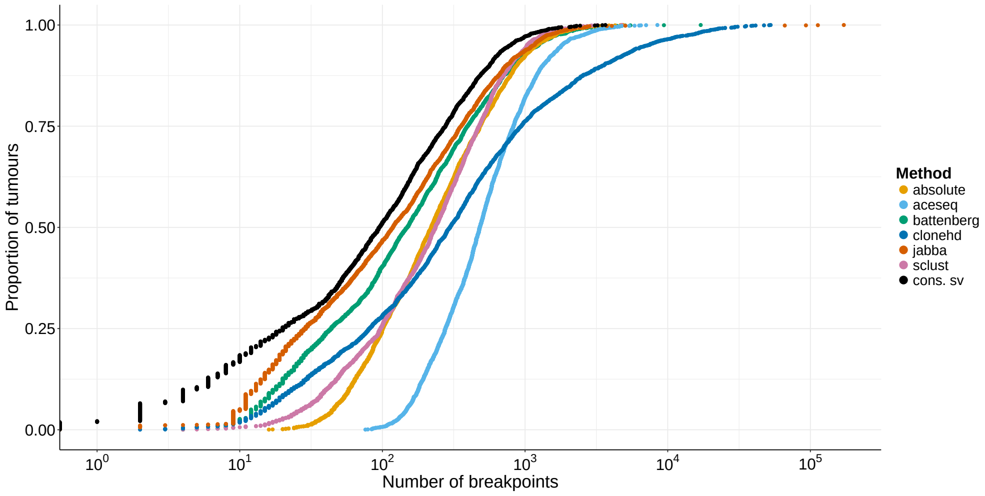


Fig. 6.1 Number of breakpoints for each of the methods used to create the consensus breakpoints (the JaBbA calls are plot for reference) and the consensus structural variants (black). cloneHD and ACEseq call more breakpoints than the other methods, hence their characterisation as *liberal* methods. This figure is inspired by one made by Jeff Wintersinger.

6.2.2 Determining consensus segment breakpoints

Copy number callers segment a sample's genome into regions assumed to have constant copy number. Each segment is bounded by a breakpoint at either end, where breakpoints correspond to a change in copy number. The collection of segments is then used to infer purity and ploidy and fit to copy number states.

We observed substantial disagreement in segmentation between the different algorithms and aimed to develop a consensus set of breakpoints, which the six callers subsequently used to call copy number.

Jeff Wintersingers consensus strategy aims to maximise "true positive" breakpoints at the potential cost of increasing "false negatives". Orthogonal evidence of copy number breakpoints from structural variants was used to quantify the "true positive" and "false negative" rate of our consensus approach. When fitting the copy number profile callers are allowed to merge adjacent segments, therefore the cost of introducing spurious breakpoints was less than that of missing breakpoints.

Copy number methods differed substantially in the number of breakpoints they defined (Fig. 6.1), with some methods calling an order-of-magnitude more breakpoints than others. Broadly speaking, these can be broken into two classes: *liberal* methods (ACEseq and cloneHD) called, on average, a great many more breakpoints than *conservative* methods (ABSOLUTE, Battenberg, JaBbA, and Sclust).

Copy number methods determine breakpoints based on data derived from the sequencing output. Methods use the BAF, logR or coverage for such purposes, sometimes in combination. These three views have their advantages and disadvantages, as was explained in the sections about Battenberg earlier in this thesis, which is a source of the observed differences in segmentation between methods Furthermore, methods differ in how BAF, logR or coverage is obtained from the sequencing data. LogR, for example, can be obtained through windows placed across the genome (overlapping or non-overlapping) or through a set of predefined single base genomic locations, while GC content can be corrected for using different approaches (Benjamini and Speed, 2012; Diskin et al., 2008).

Finally, methods can call the same segment with different breakpoints. Here the implementation matters: the exact location of a breakpoint can correspond to the edge of a window, to a known SNP or to a measured SV breakpoint. That means the called breakpoints for the same segment can be ambiguous, especially in regions with many small segments.

The algorithm that was developed for determining consensus breakpoints draws on the insight that regions between adjacent segments can be used to quantify a method's uncertainty in the exact location of the breakpoint. The segmentation released by each method consists of a set of regions defined by the genomic loci S_i and E_i, with the interval (S_i, E_i) representing a region of constant copy number. On a given chromosome, however, the region (E_{i-1}, S_i) has undefined copy number—the segmentation method inferred that CN status changed at some point within this interval, but cannot pinpoint the location.

The algorithm uses the space between segments and a fixed window size to create leeway on calls from the individual copy number methods and then looks for overlaps between methods to define consensus breakpoints. The algorithm consists of six steps, which are executed for each chromosome separately:

1. For each copy number segmentation method M, take each reported segment (S_i, E_i), and generate an interval spanning the end point of the current segment and the start point of the next, $(E_i - \delta, S_{i+1} + \delta)$. This interval indicates the belief of M that a breakpoint lies somewhere in this interval, permitting the breakpoint to move δ bases upstream or downstream beyond the reported boundaries. Here, we set δ = 50 kb, which we selected after manually comparing the breakpoints generated by a range of δ values to the underlying signal in the data. δ = 50 kb achieved a reasonable balance between false-positive consensus breakpoints (when δ was too large) and false-negative consensus breakpoints (when δ was too small).

2. Compute the intersection of intervals between the methods. Scanning from the start of the chromosome, find the first intersection I_s supported by the threshold methods T.

We defined T to be any combination of at least three of the six copy number methods, or any combination of two of the *conservative* methods (i.e., ABSOLUTE, Battenberg, JaBbA, and Sclust). This avoided calling consensus breakpoints supported by only the two *liberal* methods (ACEseq and cloneHD).

3. For a given intersection I_s: select all reported breakpoints falling within I_s. Score each breakpoint according to the size of the associated gap $G_i = \text{rank}(S_{i+1} - E_i)$, where G_i corresponds to the rank in the empirical cumulative distribution of all gaps generated by the given method. Thus, if a method assigns a large gap between two segments relative to the other segments it generates, its uncertainty in breakpoint placement is understood to be relatively large; conversely, a relatively small gap indicates high certainty. The consensus breakpoint of the intersection is then the breakpoint with the smallest G_i. In the case that two breakpoints in the intersection have the same G_i (which occurs, e.g., because both E_i and $Si+1$ fell in the intersection), arbitrarily prefer end locus to start locus and record this as the single consensus breakpoint. Otherwise, in the rare case that no input start and end loci fall in the intersection, report the upstream-most end of the intersection as the consensus breakpoint. Such cases arise when only the δ bases padding each input segment intersect, meaning that the intersection as a whole is relatively small, and that either end of the intersection can be taken as a reasonable representation of a breakpoint's position.
4. Remove all intervals that contributed to the intersection. Return to step 2. Repeat until no intersections passing the threshold remain on the chromosome.
5. Add PCAWG consensus SVs to the consensus breakpoint set. To do so, find all consensus breakpoints within 100 kb of a consensus SV. Replace the consensus BP with the consensus SV, as the SV presumably represents the same mutational event, but with greater precision concerning position. For any SVs lacking a consensus BP within 100 kb, add the SV as an additional consensus breakpoint.
6. Add breakpoints at centromeres and telomeres as necessary, as copy number status cannot be called across these boundaries. Use the chromosome lengths and centromere start and end locations reported in the hg19 human reference genome. If any centromere start or end lacks a consensus breakpoint within 1 Mb, add an additional consensus BP at that location; if a consensus breakpoint occurs within the centromere, move it to the start or end of the centromere, according to whichever point is closer. Likewise, if no breakpoint occurs within 1 Mb of a chromosome start or end position (representing telomere locations), add an additional breakpoint at the chromosome start or end.

The output of this approach is a list of breakpoints per chromosome. Each pair of adjacent breakpoints corresponds to a consensus segment, for which the six methods produced copy number calls. The next step is to combine those calls into a consensus profile.

6.2.3 Constructing consensus copy number

The consensus copy number profile should contain a call for every consensus segment, if there are enough calls. To do so I first identified 6 ways of extracting agreement between the CNA callers on a single segment (summarised in table 6.2):

(a) All methods agree on a clonal copy number call (both major and minor alleles).

(b) A single method disagrees on the copy number state of a single segment, leaving the call from this method out creates agreement.

(c) A single method disagrees on the ploidy of a sample, leaving the profile out creates agreement.

(d) The strict majority of available methods agree on clonal copy number.

(e) Complete or leave-one-out agreement is achieved by rounding subclonal copy number.

(f) Majority vote is achieved after rounding subclonal copy number.

For each sample, every segment goes through the list starting at a, until agreement is reached. On average, that obtains consensus on 90% of the genome in 86% of samples after reaching level f (Fig. 6.2). The segments that remain without a consensus call go through a second approach that is designed to find a call from a single method to be selected into the consensus profile.

To select a call I first calculate, for every CNA method, what proportion of the consensus profile it agrees with after reaching level f. This allows ranking of the methods, where an excluded profile (due to disagreement on the ploidy) is not included (see filtering below). The following additional levels were then devised:

(g) Take the call from the best method. If there is consensus for the copy number state of one of the alleles we require the best method to agree with it (see rounding below).

(h) Take the call from another method, iterating from the best to the worst performing method.

Consensus copy number levels

Star	Level	Description
3	a	Complete clonal agreement
	b	Clonal agreement of n-1 methods
	c	Clonal agreement excluding ploidy outliers
2	d	Strict majority vote on clonal copy number
	e	Complete agreement after rounding subclonal copy number
	f	Strict majority vote after rounding subclonal copy number
1	g	Best method, one allele with consensus
	h	Best method, no consensus on either allele
	i	No ploidy consensus from panel of experts

Table 6.2 Each consensus copy number segment is assigned a star quality and a confidence level. The level is based on how the consensus is obtained and can be used as a measure of the amount of confidence in the call. The star assignment is aimed to capture the quality of the copy number call in a broad scale that can be understood without details of how the consensus was established.

A special level was added to distinguish between samples where the expert panel did not reach consensus on the ploidy of a sample during a review of all the profiles and raw data for that sample. These copy number profiles were assigned copy number states through the procedure detailed above and each segment received assignment of the level corresponding to how consensus was obtained. But segments were later re-marked as level *i* to denote the extra uncertainty about the assigned copy number states.

I then devised a star rating system that denotes the amount of confidence in each of the calls. Levels *a*, *b* and *c* are the most strict and require all-but-one methods to agree at the least. These segments are therefore assigned *3 stars*. Segments for which a majority of the methods agree on either clonal or rounded clonal copy number are assigned *2 stars* (levels *d*, *e*, *f*). The remaining levels (*g*, *h*, *i*) receive *1 star* to denote the lowest confidence.

6.2.4 Rounding subclonal copy number

Subclonal copy number is reported in three different ways across the 6 methods. ABSOLUTE reports up to 3 different copy number states per segment, of which 1 is termed the ancestral state. Battenberg and Sclust report subclonal copy number as a mixture of two states, while ACEseq returns a single non-integer state (i.e. a mixture). Both cloneHD and JaBbA provided clonal calls only.

Rounded profiles were obtained in the following way:

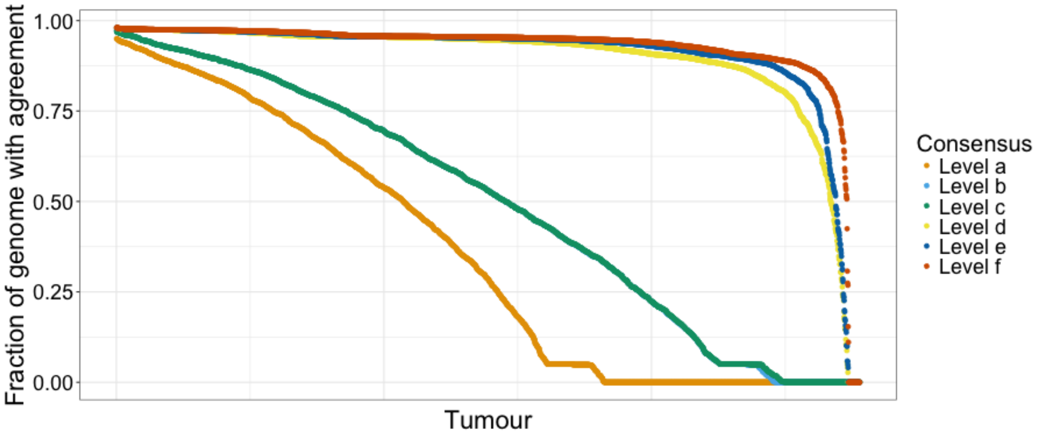


Fig. 6.2 The fraction of the genome for which a consensus can be created increases as more levels are added. Levels correspond to how consensus was obtained. In general levels a-c are the highest confidence, levels d-f medium and levels g-i low (not show in this figure). Agreement on over 90% of the genome is reached for 2406 out of 2778 samples (86%).

- ABSOLUTE: 6 ways, corresponding to rounding both alleles up and down of the ancestral state, the highest CCF state and the lowest CCF state.
- Battenberg and Sclust: 4 ways, the highest CCF state and the lowest CCF state.
- ACEseq: 4 ways, rounding both alleles up and down.

To create a consensus call for a segment I first obtain an inventory of the available copy number states across all roundings and the clonal calls from cloneHD and JaBbA are included. If there is a major/minor allele combination that satisfies the minimum number of methods criterion (either leave-one-out or majority vote) we select that state as the consensus.

If no agreement is reached I attempt to establish consensus by voting for the major and minor allele separately. An allele is accepted if it passes the minimum number of methods threshold. In some cases this leads to consensus on one of the alleles. The state of that allele is saved and fed into levels *g* and *h*, where a call is selected where one of the alleles agrees with the established consensus allele.

6.2.5 Chromosomes X and Y

Fewer methods report on X and Y chromosomes:

- X: ABSOLUTE, Battenberg (females), ACEseq, cloneHD and JaBbA
- Y: ABSOLUTE, ACEseq and JaBbA

The required number of methods to agree for the separate levels are adjusted accordingly.

6.2.6 Panel of experts review

For a range of samples the copy number callers did not unanimously agree on the ploidy. An initial computational analysis developed by Jeff Wintersinger revealed up to 361 profiles where affected. Jeff's approach was developed to automatically adjust a copy number profile in two ways: Halve the ploidy (the effect of removing a whole genome duplication), or subtracting 1 copy from each allele (the effect of removing a normal genome from the profile). A sample name was saved if an adjustment yielded a larger agreement between the methods. An additional 315 tumours did not reach over 20% agreement in a first run of the consensus approach, these tumours were also reviewed.

The samples have been put through a panel of experts review procedure. Initially to understand where the discrepancy lay between the profiles, and later to resolve the differences. The discrepancy cases can be grouped into two categories: Erroneous addition or omission of a genome doubling, or a method specific error scenario. We opted to use the manual approach after initial inspection of 100 samples because fixing the method specific scenarios would have set the process back for months, while there was only a short timeline possible to make the consensus copy number calls available PCAWG-wide.

The expert panel consisted of three core and five alternating members and sat down for four afternoon sessions. Each member prepared a figure per sample with all possibly interesting information. A central figure was used to feed the discussion contained: Copy number profiles from all methods and raw BAF, copy ratio (logR) and multiplicity values from ABSOLUTE. My personal figures contained the Battenberg profile, DPClust reconstruction, multiplicity (Fig. 6.3) and copy ratio (Fig. 6.4); an assembly of figures shown in the QC chapter of this thesis.

During the review a sample was marked as *WGD* or *no_WGD* to reflect a high or low ploidy solution the panel agreed upon. A sample was only marked on unanimous agreement amongst the panel and a maximum of roughly two minutes was maintained to discuss a sample. A sample was marked *unkown* if no agreement could be obtained within the set time. Over time, we observed that methods often show similar behaviour in particular scenarios, which made it easy to determine the disagreeing method as an outlier.

For example, Aceseq showed difficulty calling a copy number profiles with a low purity, which has since been improved. It will not only call a much higher purity, it also shifts the copy number profile up leaving a profile without losses and with empty copy number states. Figures 6.3 and 6.4 show an example, lung squamous cell carcinoma SA305293. The copy number profiles plot it showed Aceseq as calling a higher ploidy by adding an extra copy to every allele on every chromosome. The ABSOLUTE allele specific copy ratio plot clearly showed four separate states, suggesting a genome doubling. Adding a copy to every allele on chromsome 17 would leave one copy of either allele active, and it would remove all other LOH (the Battenberg QC figure shows LOH at chromosome 17p, where *TP53* resides (Fig. 6.3). Finally, the Battenberg raw data figure shows that this is a low purity tumour (separation of purple and blue lines in the bottom plot) and it confirms that, for example, chromosome 4 has the lowest coverage (top) and lowest minor allele frequency (bottom), suggesting that chromosome 4 holds the profile up. There was little discussion about this case, it is clearly whole genome doubled and should have LOH on chromosomes 4 and 17p.

Another example are cases with very heavy, wave-like, coverage artifacts where some methods experienced difficulties. Figures 6.5 and 6.6 show an example, liver cancer SA529774. cloneHD calls a large number of segments that have been fit with an additional gain, creating a much more fragmented genome as compared to the other methods. The Battenberg profile (Fig. 6.5) shows a clean fit (bottom), with a clear clonal mutation cluster (top) and mutation copy number states at integer values (middle), while the raw coverage log ratio plot (Fig. 6.6, top) confirms very noisy signal. Here the panel decided that one method gained a better fit by following the noise, whilst there is no clear evidence of additional gains, which could have lead to the calling of a whole genome doubling.

The above two examples occurred commonly enough for the panel to recognise the scenario and quickly resolve the discrepancy. In some cases however, the panel could not agree within the time limit we set for discussing a sample (which was set to 2 minutes). Figures 6.7 and 6.8 show such an example, pancreatic adenocarcinoma SA533746. ABSOLUTE and cloneHD disagree with Aceseq, Battenberg and Sclust about whether there has been a whole genome doubling. The discussion focussed on whether the allele ratio plot showed 4 distinct states and whether the subclonal segments on chromosome 4 could be fit with a clonal state when the ploidy was doubled (bottom plot Fig. 6.7). I maintain that neither of these segments would become clonal when doubling the ploidy and see no evidence of a clear mutation cluster centered around 50% of tumour cells (top and middle plots Fig. 6.7). However, within the group there was considerable doubt. We did not reach consensus within the time limits and therefore marked it as *unknown*.

Overall, the panel did not manage to agree on the WGD status of 38 cases. All segments of these genomes have been re-marked with confidence level *i* accordingly.

6.2.7 Determining consensus purity

To obtain a consensus purity I extended the calls from the 6 CNA methods with calls from a number of SNV based approaches: CliP, CTPsingle, PhyloWGS, cloneHD (on SNVs) and Ccube. Outlier calls are first removed for CNA and SNV methods separately (see filtering

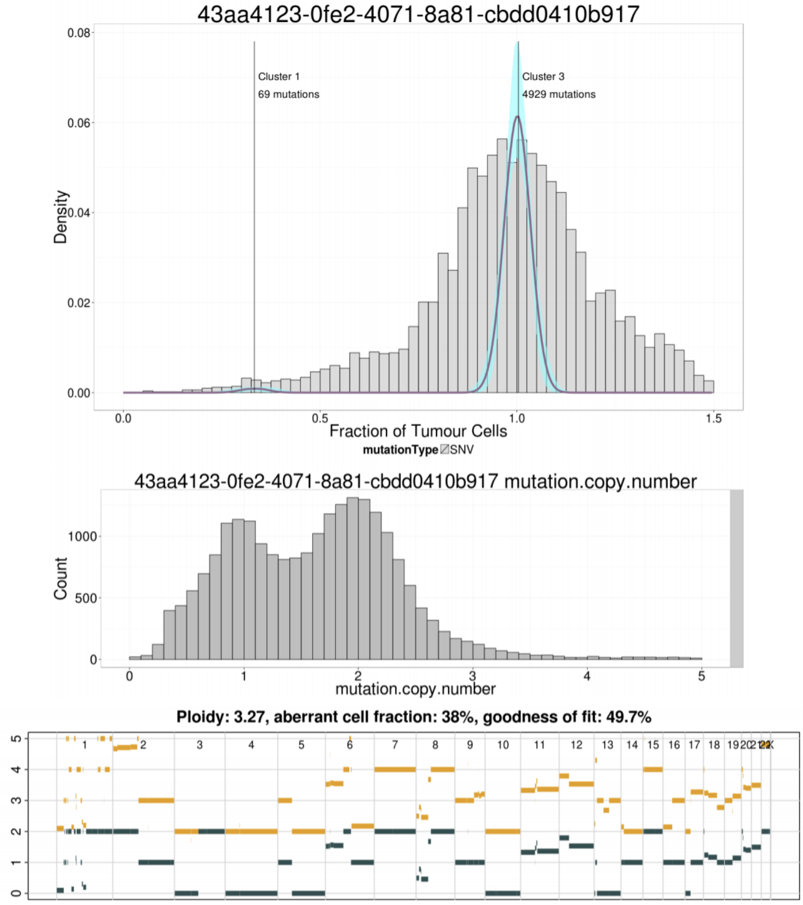


Fig. 6.3 The Battenberg QC figure for sample SA305293. It contains the mutation clustering result (top) with a histogram containing the raw CCF values of SNVs in the background and a density through where weight occurs in the foreground, SNV cluster locations are marked. The middle plot shows a histogram of mutation copy number, the raw estimate of the number of chromosomes an SNV is thought to be carried by. The bottom figure contains the Battenberg copy number profile with total copy number in orange and the minor allele in grey. This sample does not violate any of the QC metrics, and the three views of the same tumour correspond well (i.e. a clear clonal SNV cluster, SNV peaks at integer mutation copy number states 1 and 2 and a lot of allele specific copy number states at 2 that explain the ratio of SNVs on 1 and 2 chromosome copies.)

below). For each sample I establish a density over the combined data. Analogous to taking the mode we select the call that is closest to the highest peak in the density as the consensus.

There is a larger discrepancy in purity calls from CNA methods on samples with few copy number alterations. I therefore calculate the density over the calls from SNV based methods only for samples where less than 8% of the genome is altered by CNAs.

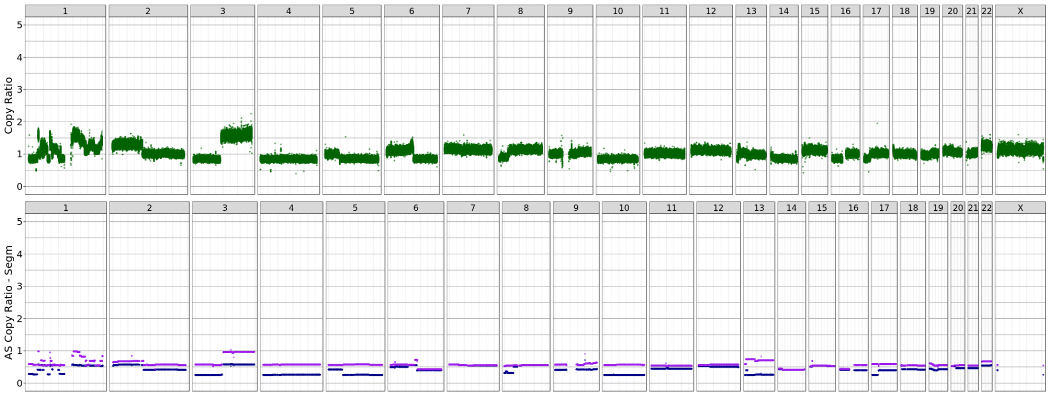


Fig. 6.4 The raw data that Battenberg produces for sample SA305293. The top figure shows the copy ratio (the R in logR), the bottom figure shows the allele specific copy ratio (R multiplied by the BAF). The top figure shows that the coverage of this sample is clean (SNP dots fall in clear straight green lines) and it shows which genomic regions correspond to the lowest relative coverage in this sample, which can be used to identify which segments are of the lowest total copy number. The bottom figure contains information on how the total copy number is relatively split into major (purple) and minor allele (blue). Focussing on chromosome 4, the figure shows it has the lowest relative coverage and the BAF of the alleles is split. That means this segment is a candidate for a 1+0 or 2+0 fit. Such reasoning helps to read a copy number profile from the raw data.

Finally, the median absolute deviation of the purity calls on a sample is calculated to capture the amount of agreement between the methods and is used as a measure of confidence.

6.2.8 Filtering

After the expert panel review of ploidy-uncertain cases, a rough reference ploidy can be obtained for almost all samples. The methods either all agreed on large portions of the genome and therefore, by extension, on the ploidy or certain ploidy calls were overruled by the expert panel.

With the accepted ploidies in hand it became possible to calculate a rough reference ploidy that serves to overrule calls from individual CNA callers. This is necessary because copy number callers can return a different ploidy in different runs and I required a way to automatically accept or reject a ploidy call. A method is allowed to deviate from the reference ploidy by a relative amount to allow for larger discrepancies on higher ploidy calls. I set the threshold at 0.25 times the reference ploidy. If a profile differed by more than this threshold, it was automatically overruled and excluded from the procedure for both consensus copy number and purity creation.

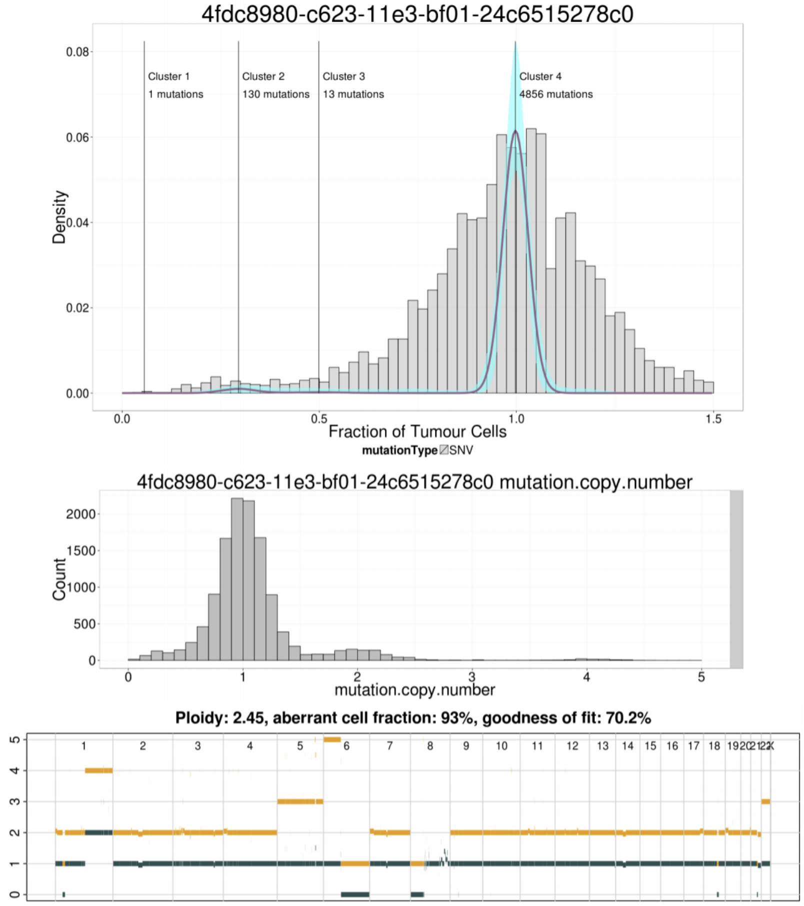


Fig. 6.5 The DPClust and Battenberg results show a clean CCF (top) and mutation copy number (middle) space which corresponds to a large clonal peak, consisting of mostly SNVs on one chromosome copy and a much smaller number of SNVs on two copies. The copy number profile (bottom) shows a relatively clean fit, with quite a few small segments that are fit with a near clonal copy number state. It shows Battenberg accounts for the noise in this sample.

Filtering on the purity calls was performed to remove outliers. A purity call was filtered out if it differed from all other non-ploidy-overruled purity calls by more than 0.2. This method was applied separately to SNV based purity values.

Finally, I also excluded calls on complex regions chromosomes 13p, 14p, 15p and 21p from some methods as they consistently appeared as losses across the entire data set.

6.3 Consensus subclonal architecture

With consensus copy number profiles established we sought to construct consensus subclonal architectures. A similar philosophy to the consensus copy number is applied: combine

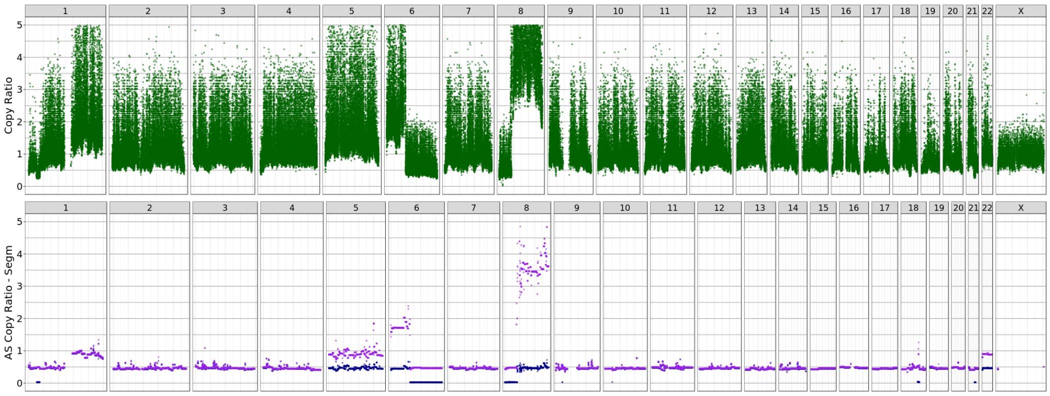


Fig. 6.6 The raw Battenberg data show what is causing problems when fitting a copy number profile for sample SA529774. The relative coverage ratio (top) is very messy and affects the allele ratio space by adding noise to the dots (bottom). This effect is most likely due to either the tumour or the normal containing strong coverage bias across the genome.

the output from multiple methods, which allows for recovery from a mistake by a single method. We have developed three orthogonal approaches that combine output from eleven individual callers (Bayclone (Sengupta et al., 2015), Ccube (manuscript in preparation), CliP (manuscript in preparation), CloneHD (Fischer et al., 2014), CTPsingle (Donmez et al., 2016), DPClust (Nik-Zainal et al., 2012b), Phylogic (Landau et al., 2013), PhyloWGS (Deshwar et al., 2015), PyClone (Roth et al., 2014), Sclust (Cun et al., 2018) and SVclone (Cmero et al., 2017)) into a consensus. We show through simulated data that the three approaches are equivalent and are consistently ranked amongst the top performing methods.

6.3.1 Three consensus approaches

Weighted Median (WeMe, by Amit Deshwar) - takes the cluster location and sizes reported by the individual methods and combines the output by minimising the earth movers distance (EMD) to the median clustering. Where the median clustering is defined as the clustering minimises its EMD to all input clusterings. To constrain the number of clusters it then performs a grid search over cluster location and size parameters to fit the median number of clusters obtained from the provided input.

Cluster ID Consensus Clustering (CICC, by Maxime Tarabichi) - uses groups of SNVs that are consistently assigned to the same cluster across methods. It first creates a vector for each mutation with as contents the cluster to which the eleven methods have assigned the mutation. Then, for each pair of vectors, a distance is calculated, resulting in a distance matrix. Hierarchical clustering is performed to cluster the mutations. The resulting tree is then cut to the median number of clusters that the eleven methods reported, rounding

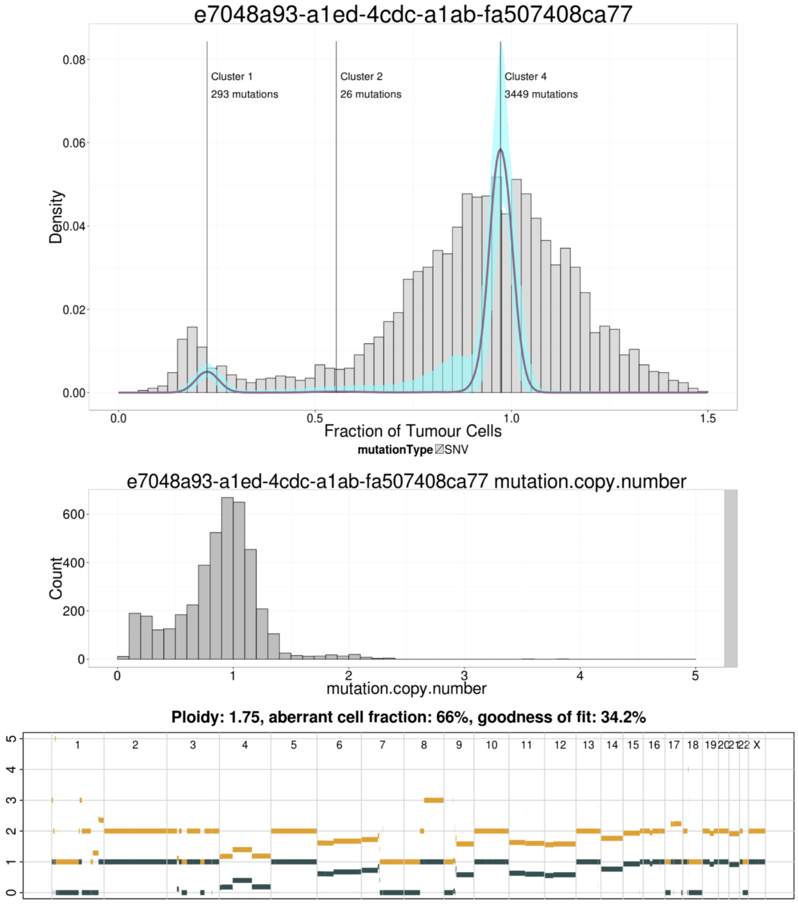


Fig. 6.7 During the review it was suggested that segments on chromosome 4 could be a separate clonal state when sample SA533746 is fit with a whole genome doubling. In that case one would expect these segments to be fit with subclonal copy number in exactly 50% of tumour cells if the profile is fit without a doubling. This figure shows that subclonal segments on chromosome 4 are not exactly within two clonal copy number states, which supports the theory that this sample has not had a whole genome duplication.

up when that number is not an integer, which results in a number of consensus clusters with their mutation assignments. Cluster locations are then determined by first calculating a consensus CCF estimate through the equations 2.14 and 2.16 in chapter 2, and then per cluster taking the median CCF of the SNVs assigned to the cluster.

Sparse Clustering for Subclonal Reconstruction (CSR, by Kaixian Yu) - starts from a mutation-to-mutation co-clustering matrix, in which cell (i, j) contains the probability that mutation i belongs to the same cluster as mutation j. The input matrix M is deconvolved using dictionary learning into a dictionary matrix D and a sparse code matrix A. A contains a sparse representation of the structure in M by its most essential components, which makes the mutations better separable. k-means clustering is then applied to A with k set to the

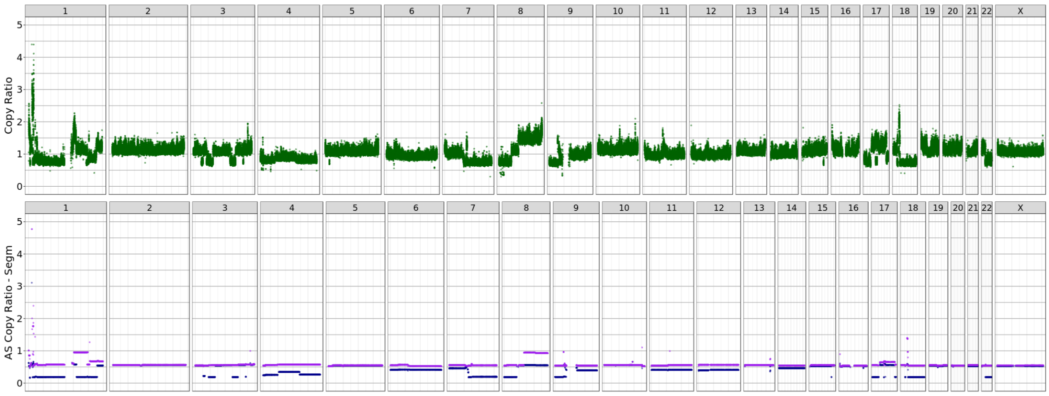


Fig. 6.8 The raw Battenberg data for sample SA533746 shows that the coverage ratio is clean (top).

median number of clusters called by the eleven methods. Cluster locations are obtained by first calculating the median cellular prevalence (CP) of each SNV across the eleven methods and then taking the average per cluster.

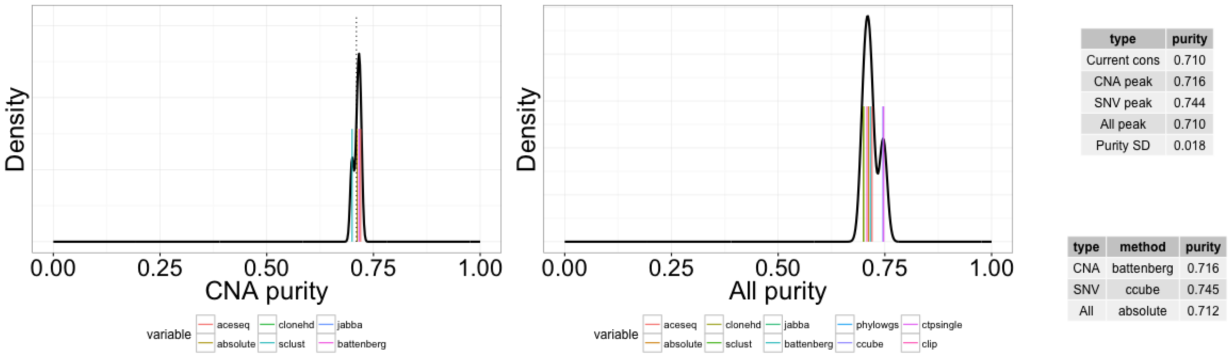
6.3.2 Performance comparison

To asses the performance of the consensus methods we compare the three consensus approaches with the eleven input methods and three methods that produce random solutions, that do not serve as input to the consensus, on the simulated data that was introduced in section 3.3 as part of the validation chapter.

In the validation chapter I also introduced three metrics that can be used to compare the results of a method to the truth or to another method: number of subclones, fraction of clonal mutations and the root-mean-squared-error (RMSE) between mutation assignments. The three metrics can be combined into a single measure by calculating the rank sum across the metrics for each method. The rank sums are normalised for whether the ranking is increasing or decreasing, resulting sum values between a best case 3/17 and worst case 3*17 as there are 17 methods.

Figure 6.10 shows the rank sums of all samples across the 17 methods when they are compared to the truth, with a black bar indicating the median rank of the results of that method and the red bar denoting the mean. A dashed line is drawn that corresponds to the lowest median rank across the methods.

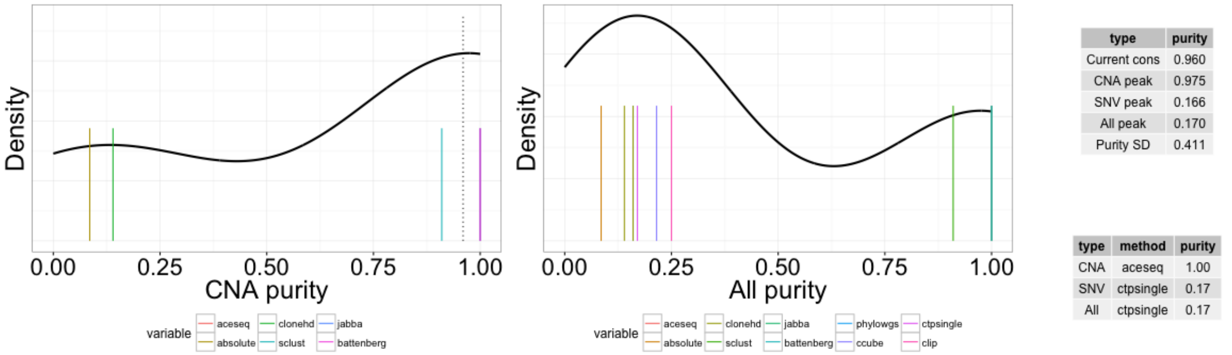
In general there are a number of methods that show a very similar performance. Phylogic, DPClust, CCube, PyClone, PhyloWGS and all three consensus methods have similar median ranks, with cloneHD and SVclone and CTPsingle not far off. The first group is followed by a second group that includes Sclust, CliP, BayClone and the informed method from



type	purity
Current cons	0.710
CNA peak	0.716
SNV peak	0.744
All peak	0.710
Purity SD	0.018

type	method	purity
CNA	battenberg	0.716
SNV	ccube	0.745
All	absolute	0.712

(a) Good agreement on purity for sample SA6251.



type	purity
Current cons	0.960
CNA peak	0.975
SNV peak	0.166
All peak	0.170
Purity SD	0.411

type	method	purity
CNA	aceseq	1.00
SNV	ctpsingle	0.17
All	ctpsingle	0.17

(b) Methods disagree on the purity for sample SA528952.

Fig. 6.9 Consensus purity establishment for two example cases. Each figure contains the purity calls from the CNA methods on the left and all methods on the right. The top table on the far right contains information about where the peaks in density are for CNA methods only, SNV methods only and all methods. The bottom contains information about which method agrees best with the consensus. The dashed line in the CNA purity figure contains the median purity from CNA methods (labelled as current consensus in the table). In scenario (**a**) there is a good agreement between the methods and a good agreement between CNA and SNV purities. This represents most tumours. I therefore opted to establish a consensus based on the overall density peak, which amounts to the mode across all CNA and SNV methods. The purity value that is closest to the peak location is then chosen as the consensus. Scenario (**b**) can occur when a copy number profile contains no clonal alterations. Not all CNA methods are capable of handling these cases, which results in disagreement and possibly an incorrect call. Inclusion of the SNV data leads shows that SNV methods agree with the lower purity value. To remove the uncertainty that CNA methods introduce in this scenario I establish consensus by evaluating the SNV methods only.

RandomClone. Finally, the third group contains the remaining two RandomClone methods that perform considerably worse on this dataset.

These results show that the three consensus approaches have a performance comparable to the best individual methods. It also shows that all eleven methods comfortably outperform a

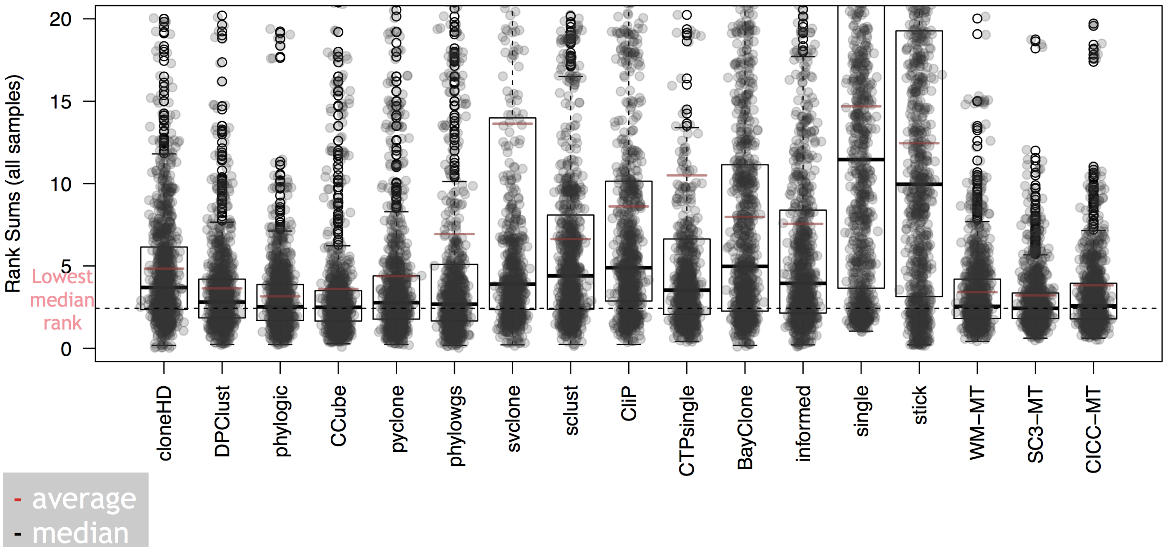


Fig. 6.10 Ranksum comparison of subclonal reconstructions across the methods. The three consensus approaches systematically perform comparable to the best individual method. There appear to be two groups of individual callers: Those that perform comparable to the consensus or are close to it, and those that perform similar to the RandomClone informed method. All methods comfortably outperform simple random approaches. These findings show that the consensus are not sensitive to an outlier solution or a poorly performing caller.

simple random approach (RandomClone stick) and assigning all mutations to a single cluster (RandomClone single). However, not all methods outperform a slightly more sophisticated random method (RandomClone informed) and these methods have been included into the consensus approaches. That the consensus methods show a comparable performance to the best methods shows that the consensus is invariant to the inclusion of a poor solution when it is constructed.

Figure 6.11 contains a pairwise comparison between the 17 methods. Each square contains a matrix with a comparison between a pair of methods (i, j) that is sorted by number of subclones (columns) and number of reads per chromosome copy (rows). A cell in the matrix is coloured blue if the *column* method has a higher ranking than the *row* method on a sample, it's coloured red if the *row* method performs better, while it's white if there is not much difference. These figures allow for exploration of performance in relation to increasing number of subclones and increasing number of reads per chromosome copy.

Similar to fig. 6.10, there appear to be three groups of methods, but their members are slightly different. Phylogic, DPClust, CCube, PyClone and PhyWGS form a block of white or lightly coloured squares in relation to each other, meaning these methods correlate very well. The second group is formed of SVclone, Sclust, CliP, CTPsingle, BayClone and the informed RandomClone method. The third group contains the final two RandomClone methods that contain nearly completely blue squares on their rows indicating they are nearly always outperformed by the callers.

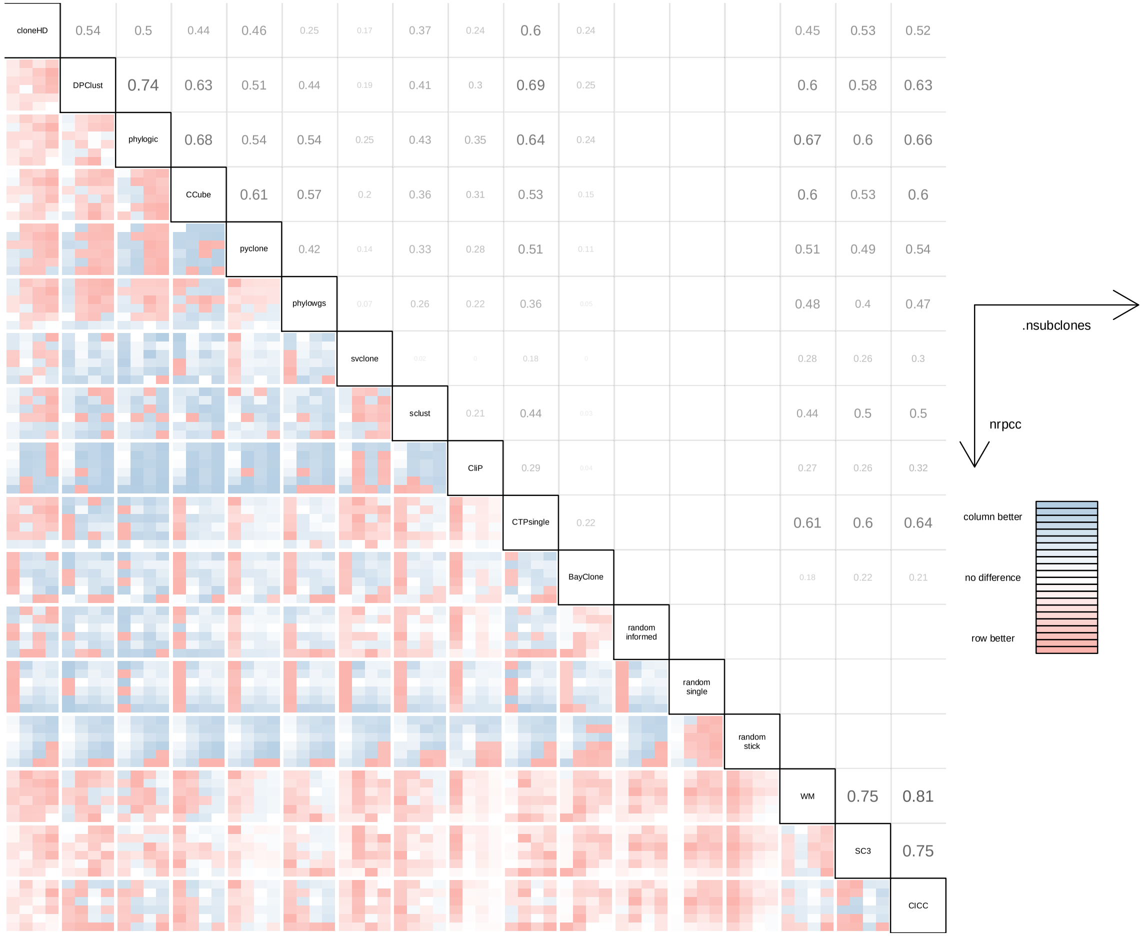


Fig. 6.11 Pairwise comparison of methods performance on the simulation data set. The comparison shows that the three consensus methods (bottom right) agree very well with each other, even though the individual methods only sporadically correlate well. The consensus methods also systematically perform better than any individual caller (including the random methods). Although in some scenarios an individual method can perform slightly better.

The consensus methods again show a very similar performance to each other and all three perform comparable to the methods in the first group. It does however appear that CICC performs better than CSR on low numbers of subclones. The squares comparing WM to CSR and CICC appear more pale, indicating that their solutions are very similar in general.

Combined, these findings show that a consensus approach is robust against an outlier solution. And because WM appears to correspond best to both other consensus methods we opted to use that for further analysis of which results are reported in the next chapter.

6.4 Purity, ploidy and sequencing coverage determine ability to detect subclones

Subclonal reconstruction depends on the ability to call subclonal SNVs in a sequenced tumour. The number of reads required to call a SNV depends on the properties of the SNV caller, and on the sequencing error rate distribution. As a rough rule of thumb, three mutant reads are typically required to detect an SNV, and mutations present in small fractions of tumour cells may be missed. The coverage at which the tumour was sequenced, the admixture of tumour and normal cells in the sequencing sample and the total amount of DNA from each tumour cell all contribute to the ability to detect clonal and subclonal mutations. The following formula combines these three factors into a power metric

$$p_s = c_s \frac{\rho}{\rho \psi_t + (1-\rho)\psi_n} \tag{6.1}$$

Here, c_s is the sequencing coverage of the tumour sample, ρ is the tumour purity, and ψ_t and ψ_n are the ploidy of the tumour and normal cells respectively (the amount of genomic material per cell, expressed in number of haploid genome copies). p_s is equivalent to the number of reads per chromosome copy and represents the expected number of reads reporting a clonal SNV. If, for example, p_s equals 10 and an SNV can be detected when there are three mutant reads, then (as an approximation) mutations present a subclone taking up 30% of tumour cells can be detected.

Figure 6.12 shows that this theoretical bound roughly corresponds to what is possible in real data. Each dot represents a tumour in the ICGC PCAWG data set. The left hand plot shows that the ability to detect subclones goes up as the number of reads per chromosome copy increases, with tumours without subclones, with 1, 2 and more subclones showing clearly visible ’bands’. The right hand plot shows the minimum CCF of the detected mutation clusters, plot against the number of reads per chromosome copy. The dashed line represents

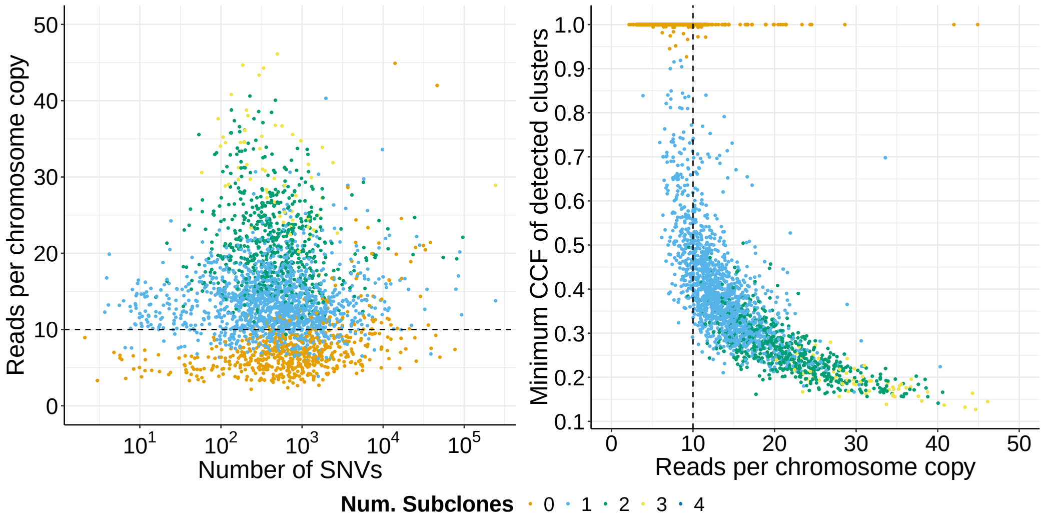


Fig. 6.12 The number of reads per chromosome copy, calculated by combining tumour purity, ploidy and sequencing coverage, determines the power to detect subclones (left). 10 reads per chromosome copy allows for detection of a subclone at 30% of tumour cells.

our theoretical bound of 10. The figure indicates that at 10 reads per chromosome copy we almost have the power to detect a subclone at 30% of tumour cells.

6.5 Correcting for the winner's curse

Figure 6.13 shows the CCF space for the same subclonal architecture and copy number profile, simulated four times with different coverage values. As coverage, and therefore the number of reads per chromosome copy, increases the light green vertical lines move closer to the black vertical lines, which indicates that the mean CCF of mutations visible in the plot moves closer to the true CCF of the clusters from which they were generated. This shifting of the weight of the clusters is caused by the winner's curse due to the clusters being represented by the mutations that by chance made it over the threshold of minimum number of supporting reads required.

As the weight of the clusters is shifted, subclonal reconstruction algorithms will also infer a shifted cluster location (if the clusters can be disentangled at all, see the top left plot in Fig 6.13). To obtain the true cluster locations and their sizes Amit Deshwar and Ignaty Leshchiner developed approaches to correct for this winner's curse effect. One approach simulates additional mutations and iteratively adjusts the cluster location depending on how much the cluster location changes. The process converges when the true cluster location has been obtained, with a corresponding size estimate from the simulated mutations. A second

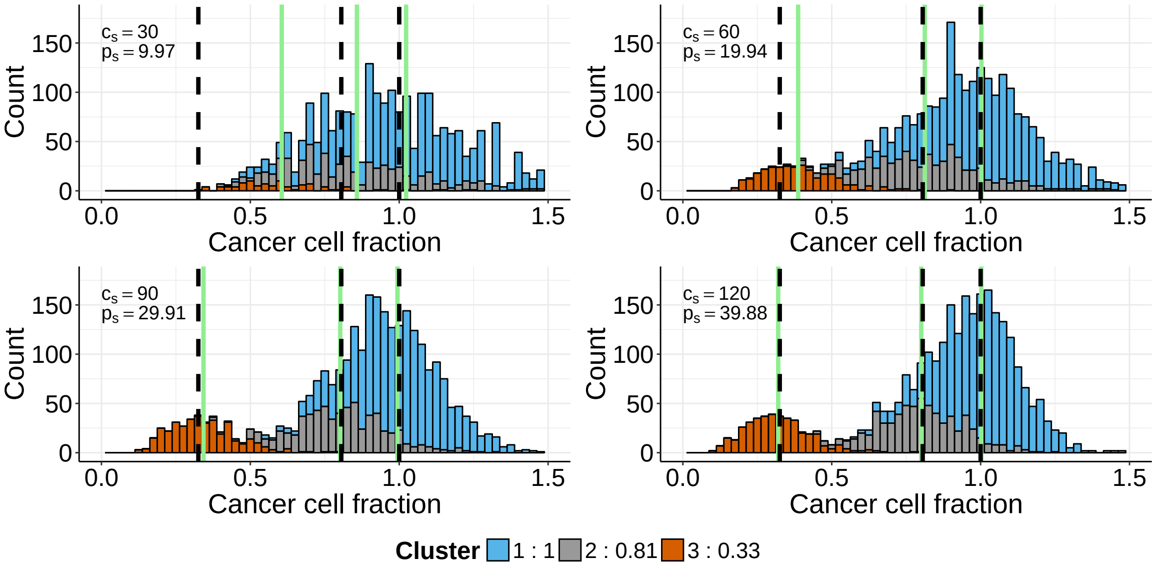


Fig. 6.13 The number of reads per chromosome copy determines the power to detect subclones. This figure shows the same tumour simulated four times with the same purity and ploidy, with a range of coverage values: 30x, 60x, 90x and 120x. The dashed black lines represent the true cluster locations, while the light green lines represent the mean CCF per cluster of mutations shown. The histogram clearly shows the effect of increasing the reads per chromosome copy: the left hand tail extends further towards 0. As the power goes up the three clusters are more fully represented, resulting in the light green bars (mean CCF of mutations present) moving towards the true cluster locations (black dashed lines). This shifting of the weight of the clusters is called the winner's curse as clusters are only represented by the mutations that by chance are supported by enough reads to be called.

approach uses moment matching to match the observed distribution to a library of available shapes and picks the shape that best corresponds to the observed CCF distribution. In the next chapter we correct the ICGC PCAWG data set for the winner's curse effect by taking the average adjustment between the two methods.