Copy-number analysis and inference of subclonal populations in cancer genomes using Sclust

Yupeng Cun[1,5], Tsun-Po Yang[1,2,5], Viktor Achter[3,5], Ulrich Lang[3,4] & Martin Peifer[1,2]

[1]Department of Translational Genomics, Center for Integrated Oncology Cologne–Bonn, Medical Faculty, University of Cologne, Cologne, Germany. [2]Center for Molecular Medicine Cologne (CMMC), University of Cologne, Cologne, Germany. [3]Computing Center, University of Cologne, Cologne, Germany. [4]Department of Informatics, University of Cologne, Cologne, Germany. [5]These authors contributed equally to this work. Correspondence should be addressed to M.P. (mpeifer@uni-koeln.de).

Published online 24 May 2018; doi:10.1038/nprot.2018.033

The genomes of cancer cells constantly change during pathogenesis. This evolutionary process can lead to the emergence of drug-resistant mutations in subclonal populations, which can hinder therapeutic intervention in patients. Data derived from massively parallel sequencing can be used to infer these subclonal populations using tumor-specific point mutations. The accurate determination of copy-number changes and tumor impurity is necessary to reliably infer subclonal populations by mutational clustering. This protocol describes how to use Sclust, a copy-number analysis method with a recently developed mutational clustering approach. In a series of simulations and comparisons with alternative methods, we have previously shown that Sclust accurately determines copy-number states and subclonal populations. Performance tests show that the method is computationally efficient, with copy-number analysis and mutational clustering taking <10 min. Sclust is designed such that even non-experts in computational biology or bioinformatics with basic knowledge of the Linux/Unix command-line syntax should be able to carry out analyses of subclonal populations.

INTRODUCTION

During pathogenesis, the genomes of cancer cells are constantly reshaped by a series of alterations such as nucleotide substitutions, small insertions and deletions, genomic rearrangements, and copy-number alterations. These genomic changes follow a Darwinian evolution process[1–5] and can be identified by high-throughput sequencing technologies[6]. The identification of subclonal populations (i.e., evolved populations of cancer cells that share common mutations) from genome sequence data, however, requires a detailed and accurate reconstruction of the cancer genome structure. This is essential given that observed variant allelic fractions (i.e., the fraction of mutated bases to the total coverage at the position of the mutation) of point mutations depend on local copy-number states and tumor purity. In addition, copy-number changes can be subclonal as well. By correcting for copy numbers, observed variant allelic fractions are typically transformed into cancer-cell fractions (i.e., the fraction of tumor cells carrying the mutation). To finally identify subclonal populations, the distribution of cancer-cell fractions is searched for distinct clusters that represent individual subpopulations.

Several computational methods have been developed that allow accurate determination of copy-number states from sequencing data of impure tumors[7–12]; however, only a few of these approaches can also call subclonal copy numbers[8,10,11]. The inference of subclonal copy-number changes is necessary to correctly assign cancer-cell fractions to mutations in these regions. Clonal mutations, which occur before the subclonal copy-number change, could mistakenly be classified as subclonal if subclonal copy-number states are not taken into account. The inferred allele-specific copy-number states and the estimated tumor purity can then be used to infer the cancer-cell fraction of each point mutation. To identify subclonal populations, the spectrum of computed cancer-cell fractions is searched for distinct clusters. Here, nonparametric Bayesian methods such as the Dirichlet process are widely being used[13–16]. Owing to resampling schemes, these methods are computationally intense, especially if the mutational load is high. Alternative clustering approaches have also been recently proposed[8,17–21].

This protocol describes how to use Sclust to estimate purity-corrected, allele-specific clonal and subclonal copy numbers. To estimate these values, our approach conditionally optimizes the likelihoods of genomic imbalances and read-depth ratios. Furthermore, we propose a fully nonparametric mutational clustering method by transforming the task into an inverse problem that can be solved with low computational burden by using smoothing splines. In addition, Sclust combines copy-number inference with mutational clustering in a single package. This has the advantage that the tumor purity can be directly calculated from mutation clusters if there are no or few copy-number changes present. Our method automatically switches to mutation-based purity determination if the copy-number information is insufficient. The entire method is based on our previous implementations that have been used in a series of large-scale cancer genome sequencing studies[22–25]. Detailed descriptions of all technical and mathematical aspects are given in the **Supplementary Note**.

Development of the protocol

The determination of copy-number states is a central component of the analysis of large-scale cancer genome sequencing data. We improved our method from early implementations to infer allele-specific copy numbers in whole-exome sequencing data of small-cell lung-cancer samples[23]. After these improvements, Sclust was applied to various cancer genome sequencing studies[22,24,25]. Reconstructing tumor evolution from bulk tumor-sequencing data has received some attention lately. To this end, we developed a new mutational clustering method that is less computationally intensive than other approaches and applied the method to determine the intra-tumor heterogeneity of small-cell lung cancer[22]. Sclust is used in the pan-cancer analysis of whole genomes (PCAWG) of the International Cancer Genome Consortium[26].

Comparison with alternative methods

Numerous methods have been implemented to perform copy-number analysis and mutational clustering from cancer genome sequencing data[7–21]. Of these methods, only cloneHD[8] and Sclust unify copy-number analysis and mutational clustering in a single package (**Table 1**). **Table 1** describes a comparison of the central properties of 12 alternative copy-number and mutational clustering algorithms. Further comparisons of Sclust with alternative state-of-the-art methods (by PCAWG) on ~2,800 cancer genomes are currently being carried out within the International Cancer Genome Consortium and will be published elsewhere. Moreover, Sclust is among the six methods used to construct a consensus copy-number estimate within this initiative. Our performance tests described below (**Tables 2** and **3**) showed a remarkably high computational efficiency.

Copy-number analysis. To validate our copy-number method, we compared our results from 38 whole-genome-sequenced small-cell lung-cancer samples[22] with the corresponding single-nucleotide polymorphism (SNP) array analysis based on the Absolute algorithm[27]. In general, we obtained a good concordance between Sclust and Absolute (**Fig. 1a**,**b**). For tumor purities close to one, Absolute estimates of tumor-cell content are systematically higher than those of Sclust (**Fig. 1a**). This might be because no matched normal was used in the SNP array analysis. In this setting, the algorithm must infer whether a SNP is likely to be heterozygous in the patient's normal genome. This, in particular, is difficult in loss of heterozygosity (LOH) regions if the tumor purity is close to one, such that wrongly assigned homozygous SNPs can bias purity estimates to higher values. Tumor ploidies agreed very well between the two methods (**Fig. 1b**). Only one sample showed a noticeable difference, for which Absolute assessed a diploid and Sclust a triploid tumor. Distinguishing higher-ploidy tumors from diploid tumors can in some cases be quite challenging[7].

To further assess the validity of our purity estimates, we mixed sequencing reads *in silico* from a normal genome into its matched tumor genome (small-cell lung-cancer cell line H2171; ref. 23). We mixed reads to roughly preserve the total number of reads of the tumor-cell line. As cell lines are composed of 100% tumor cells, the mixing fraction is identical to the purity and can thus be directly estimated by Sclust. By scanning a range of mixing fractions from 0.3 to 1, we obtained an almost perfect agreement between the mixing fractions and estimated purities from Sclust (**Fig. 1c**). Altogether, these results demonstrate that Sclust is able to accurately determine copy-number profiles and tumor purities.

Subclonal inference. Whole-genome sequencing of breast cancer genome PD4120a to ×188 coverage has recently been performed, followed by a considerably detailed reconstruction of its evolutionary history[15]. Two published methods have used these sequencing data to demonstrate the validity of their subclonal reconstructions: cloneHD[8] and Theta[11]. Therefore, PD4120a serves as an ideal benchmark case for Sclust.

We estimated a tumor purity of 70%, which is exactly the same as the original Battenberg estimate[15] and very close to the results of Theta[11]. The purity estimate of cloneHD[8] is 78.4%, which is slightly higher than that of the other methods. The copy-number profile computed by Sclust shows all central features as previously reported by the Battenberg method (**Fig. 2a**)[15]. We can recapitulate the deletion of chromosome 4, as well as the subclonal deletions of chromosomes 7, 13, and 22q11. By contrast, Battenberg detects a series of subclones that are almost clonal (chromosomes 6, 8, 9, 11, 12, 14, and 15), whereas Sclust assigns these regions as being clonal. In addition, Sclust shows a slightly different segmentation of chromosome 1, leading to differences in the subclonal structure as compared with that of the Battenberg method. Overall, the differences between Sclust and Battenberg are still relatively small. However, from this analysis, we cannot conclude whether Sclust is too conservative or Battenberg overcalls subclonal copy numbers.

Mutational clustering using Sclust yielded four populations (**Fig. 2b**). Together, the numbers of the clusters and their cancer-cell fractions are in agreement with the results from the Bayesian Dirichlet model[15]. Furthermore, we performed mutational clustering with PyClone[16] and PhyloWGS[13] based on copy-number calls from Sclust. To keep the computational burden in an affordable range, we sampled the 66,441 single-nucleotide somatic mutations down to a representative set of 5,000 mutations. Still, the required run time was 9 h 8 min for PyClone and 39 h 12 min for PhyloWGS. Clusters identified by PyClone agreed well with Sclust, but only the location of cluster B was shifted to a slightly lower cancer-cell fraction (as seen by peaks in the posterior distribution; **Fig. 2b**). As discussed in the following, a similar shift of cluster B was not present in the PhyloWGS analysis. As PhyloWGS determines phylogenetic trees that fit the input cancer-cell fractions, a comparison with Sclust is complicated because not all populations of the phylogenetic trees must be represented by a visual cluster. On the other side, all visual clusters should be represented in the phylogenetic trees inferred by PhyloWGS. For the different tree topologies (five, six, and seven populations/nodes), all cluster locations identified by Sclust are in perfect agreement with population sizes estimated with PhyloWGS (**Fig. 2b**). Similar to the copy-number analysis, the deviations are slightly larger for cloneHD[8], which might be due to differences in the assumptions used to reconstruct the subclonal structure. The distribution of the number of mutations assigned to the four clusters (**Fig. 2c**) is almost the same as that from the Bayesian Dirichlet model[15]. In total, copy-number analysis and mutational clustering achieved high agreement with the discussed alternative methods.

Computational performance. To assess the computational performance of Sclust, we analyzed 14 lung adenocarcinoma samples, which were sequenced on a whole-genome and exome platform[28]. Sclust (modules: *cn* and *cluster*) is carried out on a single core of an Intel Xeon X5650, 2.67-GHz processor. The results regarding run time and memory consumption of Sclust are summarized in **Tables 2** and **3**. Across all samples, the average run time of the *cn* module was 5.16 ± 0.91 s for the exomes (**Table 2**) and 94.4 ± 1.89 s for the genomes (**Table 3**). The run time of the *cluster* module was only 1.03 ± 0.24 s in the case of exome sequencing (**Table 2**) and 1.62 ± 0.27 s for the whole genomes (**Table 3**). Unlike Sclust, mutation-clustering methods based on the Bayesian Dirichlet model are slower and depend strongly on the number of mutations, owing to the massive computational burden of the Monte-Carlo Markov chain sampling. Assessing

TABLE 1 | Comparison of copy-number and subclonal architecture inference methods.

Property	ABSOLUTE, ASCAT, CNAnorm	TITAN, THetA	cloneHD	PhyloWGS, Canopy	PyClone	CITUP	Bayclone	Sclust
Performs own copy-number segmentation	N	Y	Y	N	N	N	N	Y
Uses rearrangement breakpoints in segmentation	N	N	N	N	N	N	N	Y
Calls absolute clonal copy numbers	Y	Y	Y	N	N	N	N	Y
Calls subclonal copy numbers	N	Y	Y	N	N	N	N	Y
Performs mutational clustering on single samples	N	N	Y	Y	Y	Y	Y	Y
Clusters copy-number alterations	N	N	Y	Y	N	N	N	Y
Deals with multiple samples	N	N	Y	Y	Y	Y	Y	N
Reconstructs phylogenetic trees	N	N	N	Y	N	Y	N	N
Allows subclonal copy numbers for clustering	N	N	Y	Y	Y	N	N	Y

N, no; Y, yes.

the computational burden of PyClone and PhyloWGS on the 14 whole exomes clearly shows that these resampling-based methods are orders of magnitude slower than those of Sclust (**Table 2**). Furthermore, the run times of PyClone and PhyloWGS strongly correlated with the number of mutations (PyClone: $R^2 = 0.88$, $P = 2 \times 10^{-7}$; PhyloWGS: $R^2 = 0.97$, $P = 3{\cdot}\times 10^{-11}$). By contrast, we did not find a significant correlation of the run time with the total number of mutations in either the exome- or genome-sequencing runs (exome: $R^2 = 0.061$, $P = 0.39$; genome: $R^2 = 0.053$, $P = 0.43$), and the average run time only marginally increased between whole exomes and genomes (**Table 2**). The peak memory consumption of both modules was in an acceptable range (~0.14 GB for exomes and ~8.5 GB for genomes).

Simulations. To assess the validity of our proposed mutational clustering algorithm, we simulated a series of datasets composed of a clonal and a subclonal population. To provide a realistic simulation, including copy-number changes, local coverage fluctuations, and normal contamination, we chose a real small-cell lung-cancer whole genome as background[22]. The tumor that we selected (patient number: S01563) had an estimated purity of 78.4%, a ploidy of 2, and a total mutational load of 42,494 single-nucleotide variants. Of these point mutations, we took their expected variant allele fractions and simulated new observed variant allele fractions using a binomial distribution. Furthermore, we scaled the local read depth to simulate a predefined average genome-wide coverage. Especially for small cancer-cell fractions,

TABLE 2 | Performance test of Sclust on whole-exome sequencing data for 14 lung adenocarcinoma samples[28].

Sample	SNVs	cn: time (s)	cn: mem (GB)	cluster: time (s)	cluster: mem (GB)	Time (s): PyClone	Time (s): PhyloWGS
S00488	426	5.60	0.138	0.99	0.037	403.8	1,294.6
S01302	621	5.83	0.138	0.74	0.037	1,068.5	2,112.3
S01331	237	4.08	0.137	1.08	0.037	212.0	512.6
S01341	74	5.55	0.138	0.87	0.037	189.5	226.1
S01345	381	3.88	0.137	1.46	0.037	248.0	746.4
S01346	391	5.66	0.138	0.74	0.037	270.7	1,057.7
S01356	401	4.33	0.138	0.76	0.037	376.0	1,178.5
S01381	197	6.03	0.137	0.89	0.037	140.5	378.7
S01404	172	5.84	0.138	1.04	0.037	143.1	388.2
S01405	318	3.80	0.138	1.03	0.037	227.7	703.2
S01407	188	5.71	0.137	1.47	0.037	150.4	395.0
S01467	272	6.01	0.138	1.28	0.037	295.2	585.7
S01468	482	3.93	0.138	1.09	0.037	415.2	1,180.3
S01478	462	5.98	0.137	1.05	0.037	440.3	1,209.1
Average	330.14	5.16	0.14	1.03	0.04	327.2	854.9

All samples showed a large variety in the total amount of called single-nucleotide variants (SNVs). For both central Sclust modules, *cn* and *cluster*, the total run time (time) is shown in seconds and the peak memory consumption (mem) is given in GB. Run times for PyClone and PhyloWGS are given in the last two columns.

a small number of mutated reads can occur. To mimic the sensitivity of a mutation caller, we did not accept any mutations with 2 mutated reads; we accepted only 50% with three, 80% with four, and 95% with five mutated reads. All mutations with 6 mutated reads were included in the simulations.

The results of the simulation study are shown in **Figure 3**. Here, either 70 or 30% of clonal mutations are simulated (upper or lower panels, respectively). The coverage was chosen to increase from left to right in the range of ×30, ×60, and ×90. We scaled the size of the data points proportionally to the number of mutations, which were assigned to the corresponding population. Overall, the reconstructed cancer-cell fractions agree well with their expected values, with accuracy increasing for higher-coverage data. The latter result is expected and suggests the consistency of the method, as the sampling noise decreases with increasing the coverage. The largest disagreement is found in the region where the clonal and subclonal populations merge. In this region, it is in general difficult to disentangle the two populations. Furthermore, low-coverage tumors showed systematically higher cancer-cell fractions for small subclones. This is due to the simulated sensitivity of the mutation caller, accepting predominantly read counts from the upper tail of the binomial distribution. The selection procedure thus leads to a shift toward higher cancer-cell fractions as those picked up by our method. In total, we conclude that Sclust yields valid results with limitations in discriminating highly overlapping populations and limitations in the sensitivity of the mutation caller.

Advantages and limitations of Sclust

Sclust performs copy-number analysis and mutational clustering with a particularly low computational burden. This enables users to apply the method to larger sample sets without the necessity of having access to large, high-performance computing infrastructures. By integrating copy-number analysis with mutational clustering in a single package, the determination of tumor purity directly from the mutation clusters is supported by Sclust. Furthermore, copy-number analysis with Sclust is not prone to over-segmentation.

Currently, Sclust does not support copy-number analysis without the co-sequenced matched normal. For a minority of samples, Sclust does not yield biologically correct results using default settings. This is a common problem for all in-depth

TABLE 3 | Performance test of Sclust on whole-genome sequencing data for 14 lung adenocarcinoma samples derived from the same patients as in **Table 2**.

Sample	SNVs	cn time (s)	cn mem (GB)	cluster time (s)	cluster mem (GB)
S00488	74,595	93.01	8.565	2.01	0.037
S01302	118,339	95.70	8.567	1.54	0.037
S01331	47,012	94.04	8.579	1.66	0.037
S01341	12,394	95.67	8.482	1.11	0.037
S01345	37,079	91.87	8.509	1.96	0.037
S01346	102,639	99.11	8.535	1.40	0.037
S01356	70,706	91.53	8.558	1.39	0.037
S01381	29,635	94.14	8.492	1.42	0.037
S01404	39,048	95.63	8.492	1.77	0.037
S01405	49,803	94.51	8.536	1.70	0.037
S01407	40,900	93.01	8.494	1.49	0.037
S01467	48,861	93.75	8.539	1.65	0.037
S01468	104,178	94.54	8.574	2.08	0.037
S01478	83,469	95.00	8.501	1.43	0.037
Average	61,332.71	94.39	8.530	1.62	0.037

copy-number callers and is mainly caused by the presence of whole-genome duplications. However, Sclust has a series of parameter settings to calibrate such cases. Suggestions on how to identify and calibrate these 'difficult' cases are given in the PROCEDURE section (Step 5). The current version of Sclust cannot perform multi-sample analyses or the reconstruction of phylogenetic trees (**Table 1**).

Required expertise

Non-experts in computational biology or bioinformatics should be able to carry out analyses with Sclust. However, a basic knowledge of Linux/Unix command-line syntax is required to install and run the protocol. The knowledge of scripting languages is not required but can be helpful, e.g., in converting the mutation calls into the Sclust-specific .vcf format.

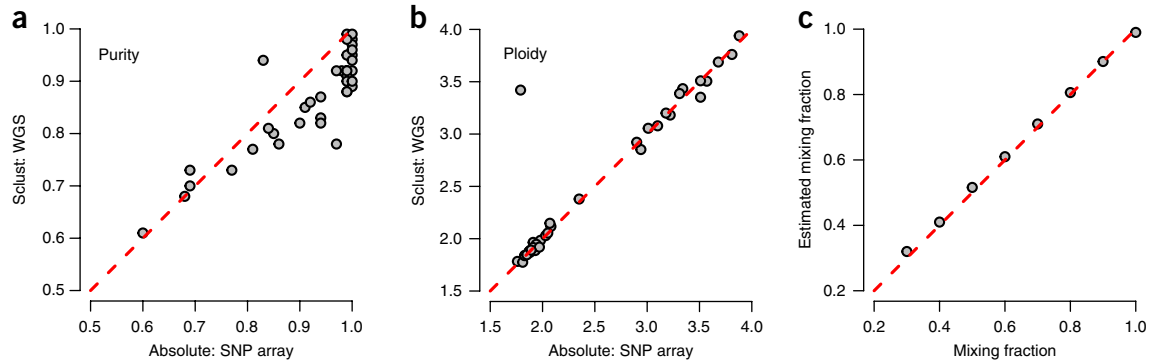

Figure 1 | Validation of the copy-number analysis of Sclust against Absolute and a mixing series of the small-cell lung-cancer cell line H2171 with its matched normal cell line. (**a**,**b**) Purity (**a**) and ploidy (**b**) estimates were compared between Absolute (SNP arrays) and Sclust (whole-genome sequencing). (**c**) *In silico* mixing experiment with the cell line H2171 with its matched normal cell line. The mixing fraction is equivalent to the tumor purity and is therefore directly estimated by Sclust.

Overview of the procedure

An overview of the various Sclust modules. Our method consists of three modules that are subsequently executed to generate the full copy-number and clonality analysis (**Fig. 4**). In the first module (*bamprocess*), all necessary input data are generated to perform copy-number analysis (Steps 1 and 2). Here, read counts over a partitioned genome are computed, for which a predefined partitioning for whole-exome and whole-genome sequencing data from human (hg19 build) and mouse (mm10 build) is provided by the program package. In addition, base counts of common SNPs are extracted from alignments to compute biallelic frequencies across the genome. Biallelic frequencies are tumor-specific allele frequencies of SNPs that are heterozygous in the matched normal. From those two datasets, clonal and subclonal allele-specific copy numbers, purity, and ploidy estimates are then computed in the *cn* module (Steps 3, 4, and 5). In the case that the amount of copy-number changes is not sufficient to reliably determine the tumor purity, the purity is directly computed from mutational clustering by iteratively shifting the cancer-cell fraction of the clonal population to one. From copy-number and purity estimates, the expected allelic fraction of each mutation is determined under the assumption of clonality. The ratio between the observed and expected allelic fractions then yields the cancer-cell fraction of each mutation. To infer the subclonal architecture of a tumor, clusters of cancer-cell fractions are determined in the *cluster* module (Step 6). These clusters then represent evolutionary subpopulations and are determined by a smoothing spline based on deconvolution of the intrinsic sampling noise from the unknown distribution of subclonal populations. Peaks in the reconstructed distribution of the subclonal populations represent the distinct mutational clusters. Each mutation is finally assigned to the most likely cluster. All technical and mathematical details are extensively discussed in the **Supplementary Note**.

Experimental design

Sclust commands. Sclust is a command-line tool that consists of three modules: *bamprocess*, *cn*, and *cluster*. These modules are controlled by parameters that are described in **Box 1**.

Input data. Our protocol requires tumor and matched normal alignment files (in .bam format), as well as a file containing all somatic point mutations (single-nucleotide changes and small insertions/deletions) in .vcf format. Note that Sclust requires special fields in the .vcf file, which can be easily generated from other .vcf files. A complete description of these extra fields is given in **Box 2**. Furthermore, Sclust allows the inclusion of rearrangement breakpoints in the segmentation of copy numbers as optional input data.

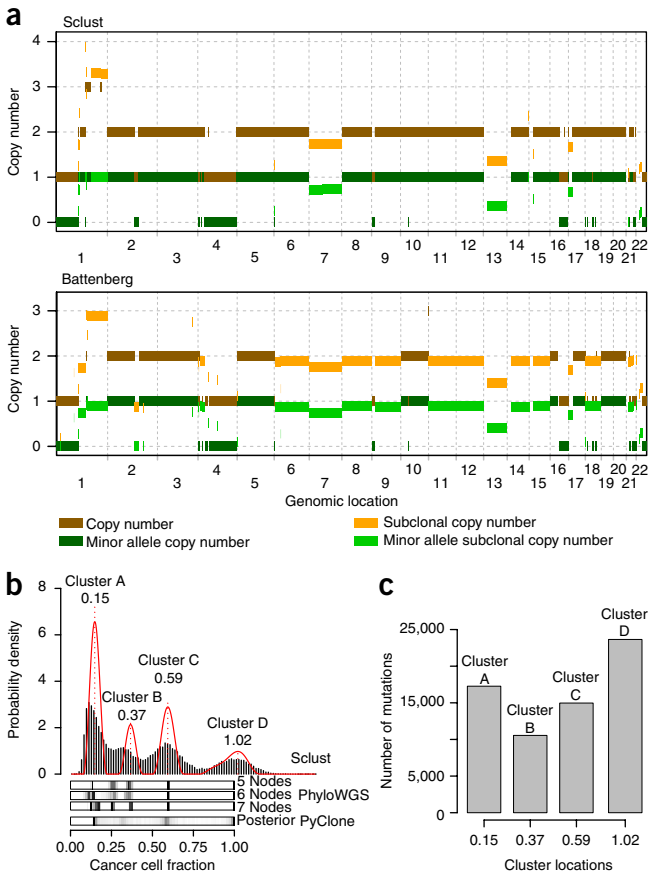

Figure 2 | Reconstruction of the subclonal structure of the breast cancer case PD4120a. (**a**, Top) The estimated copy-number states of all autosomes using Sclust and the Battenberg method are shown, where the brown and dark-green lines show the total clonal copy number and clonal minor allele copy number, respectively, for Sclust. Battenberg copy-number calls were kindly made available by the authors of the original study[15]. (Bottom) Orange and light-green lines depict the estimated subclonal copy-number states in a similar manner for the Battenberg method. (**b**) Histogram of cancer-cell fractions based on 66,441 point mutations together with results from mutational clustering by Sclust, PyClone, and PhyloWGS. (**c**) The distribution of the number of mutations assigned to the four clusters.

Conventions. Throughout the PROCEDURE, we refer to `<sample>` as the sample identifier, and chromosome names are denoted by chr1–chr22, chrX, and chrY. For the sake of simplicity, we assume that the Sclust binary is included in the PATH variable of the shell.

MATERIALS

EQUIPMENT

- Data files: tumor and matched normal data aligned to the human reference hg19 or the mouse reference mm10, and mutation calls in a .vcf format adapted to Sclust (Step 3). A list of genomic breakpoints, e.g., from rearrangement calls is optional normal.
- A standard computer system with a Linux or Mac OS X operating system. For hardware requirements, see the Equipment Setup section. The availability of a large-scale computing cluster is not essential, but it speeds up the performance of the *bamprocess* module substantially.
- The program source code, Sclust, is freely available and can be downloaded from http://www.uni-koeln.de/med-fak/sclust/Sclust.tgz

EQUIPMENT SETUP

Hardware requirements As already shown in **Tables 2** and **3**, memory requirements depend on whether a whole-exome or -genome workflow is used. Whole exome: 2 GB of memory, <50 MB of storage available for files generated by Sclust. Whole genome: 16 GB of memory, ~1.2 GB of storage available for files generated by Sclust. The Sclust program package requires 4.5 GB of storage after installation.

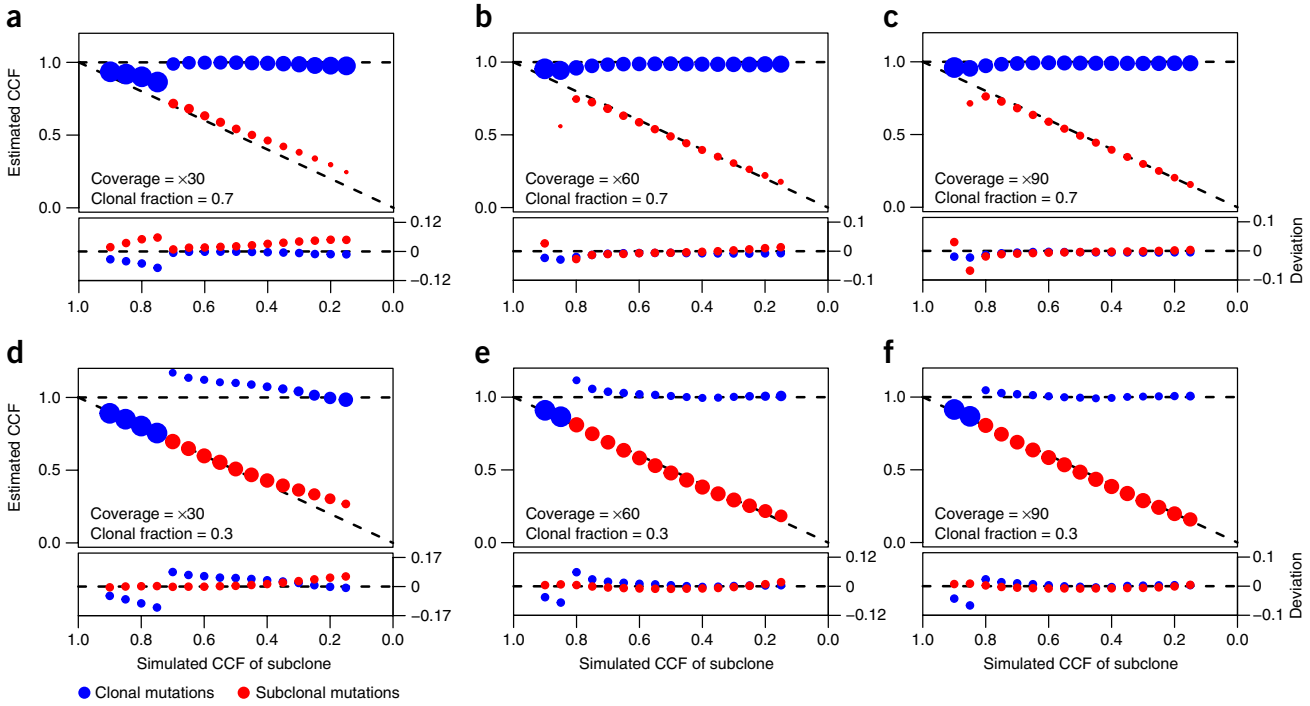

Figure 3 | Simulations of a clonal and subclonal population with different proportions of clonal mutations. (**a**–**f**) All panels show the simulated (*x* axis) versus the estimated (*y* axis) cancer-cell fractions (CCFs) of the subclones. The simulations are based on the structure of a small-cell lung-cancer sample with 42,494 point mutations. The fraction of clonal mutations (clonal fraction) is kept constant for the upper (**a**,**b**,**c**) and lower (**d**,**e**,**f**) panels but with increasing genome-wide mean coverage (×30, ×60, and ×90). Mutations that are assigned by Sclust as being clonal are shown in blue and subclonal mutations are depicted in red. The size of the data points reflects the proportion of clonal or subclonal mutations, respectively. The lower panel of each subfigure shows the deviation of the Sclust estimates from the real values used in the simulation.

Software requirements A standard C++ compiler is required to install Sclust. For Mac OS X users, Apple's xcode package (https://developer.apple.com/xcode/) contains all the relevant tools needed to build Sclust. In the case of Linux users, the installation of a standard development package is sufficient. Please make sure that a working version of libz (https://zlib.net) is installed. Sclust generates plots that are helpful to calibrate the copy-number profile if necessary. For this purpose, R (https://cran.r-project.org) is used as the plotting engine; therefore, we recommend installing a current version of R before the installation of Sclust.

Installation Download Sclust by clicking on this link: http://www.uni-koeln.de/med-fak/sclust/Sclust.tgz and expand the archive by executing `tar xvzf Sclust.tgz`. Change into the src directory with the command `cd Sclust/src` and start building the program by typing`make`. The build can be tested by executing '*make test*'.

PROCEDURE

Preprocessing of alignment files ● TIMING 15 min–15 h

▲ CRITICAL Assume that the whole-genome sequencing alignment .bam file of the tumor is denoted by <sample>_T.bam and that of the normal is <sample>_N.bam. Please make sure that both .bam files are position-sorted and indexed.

1| Extract the read ratio and SNP information of the chromosome (`<chr>`) from the .bam-files by following this command:

```
Sclust bamprocess -t <sample>_T.bam -n <sample>_N.bam -o <sample> -part 2 -build
hg19 -r <chr>
```

After completion of all chromosomes, generate temporary data files using this command:

```
<sample>_chr1_bamprocess_data.txt,...,<sample>_chrY_bamprocess_data.txt
```

Please see **Box 1** for a description of the parameters in the command.

? TROUBLESHOOTING

2| Merge temporary data files following this command:

```
Sclust bamprocess -i <sample> -o
<sample>
```

Generate a read-count file: <sample>_rcount.txt and a file containing base counts of common SNPs: <sample>_snps.txt. The formats of the output files are shown below.

<sample>_rcount.txt:

chromosome	start	end	rcount_t	rcount_n	GC
chr1	69069	70029	27	39	0.422477
chr1	861297	861417	30	32	0.628099

The first column is the chromosome of the partition and is followed by its start and end positions. The next two columns are the read counts of the tumor and matched normal having the start position within the partition. The last column is the GC content of the partition.

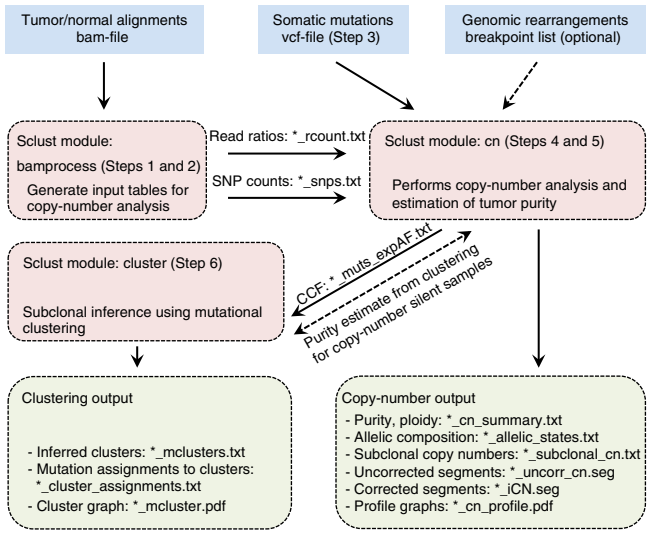

Figure 4 | Overview of the Sclust workflow. Input data are passed to the different modules of Sclust, which must be executed in a fixed order (first *bamprocess*, then *cn*, and finally *cluster*). Names of the output files generated by Sclust are shown at the bottom of the figure, and a complete description of their contents is given in the PROCEDURE section.

<sample>_snps.txt:

part_no	chromosome	position	T_A	T_C	T_G	T_T	N_A	N_C	N_G	N_T	allele_A	allele_B
2	chr1	865628	0	0	22	0	0	0	23	0	0	2
2	chr1	865682	0	0	27	0	0	0	9	0	0	2

The first column is the partition number, and is followed by the genomic position of the SNP, and then by its base counts in the tumor (in the order A, C, G, and T) and in the matched normal (in the same order). The last two columns are the two possible bases of the SNP, which are numerically encoded by A = 0, C = 1, G = 2, and T = 3.

Copy-number analysis ● TIMING <5 min

3| Convert the .vcf files to the Sclust-specific format. The header of our .vcf format is like a standard .vcf file but must contain a few extra fields. Accurate information for some of those extra fields is required, whereas a generic placeholder, e.g., '.' can be provided for other fields; please see a description of the extra fields in **Box 2**. The header lines should contain the following:

```
##INFO=<ID=DP,Number=1,Type=Integer,Description="Read Depth Tumor">
##INFO=<ID=DP_N,Number=1,Type=Integer,Description="Read Depth Normal">
##INFO=<ID=AF,Number=A,Type=Float,Description="Allelic Frequency Tumor">
##INFO=<ID=AF_N,Number=A,Type=Float,Description="Allelic Frequency Normal">
##INFO=<ID=FR,Number=1,Type=Float,Description="Forward-Reverse Score">
##INFO=<ID=TG,Number=1,Type=String,Description="Target Name (Genome Partition)">
##INFO=<ID=DB,Number=0,Type=Flag,Description="dbSNP Membership">
#CHROM      POS      ID      REF      ALT      QUAL      FILTER      INFO
```

▲ **CRITICAL STEP** Note that only mutations for which PASS is set as the filter option will be considered for mutational clustering. An example .vcf file can be found under example/data/H2171_mutations.vcf in the Sclust program package.

Box 1 | Description of parameters in Sclust

Parameters of the ***bamprocess*** module.

`-t`	Alignment file of the tumor sample
`-n`	Alignment file of the matched normal
`-o`	Name of output files (prefix): here, the same sample identifier is used for the sake of simplicty; however, a different name is also prossible
`-part`	Number of the genomic partitioning scheme: 1 for whole exome; 2 for whole genome
`-build`	Genome build: hg19 for human and mm10 for mouse
`-r`	Chromosome to extract: repeat the command above for all chromosomes: `<chr>=chr1,...,<chr>=chr22,<chr>=chrX,<chr>=chrY`

Parameters of the ***cn*** module

`-rc`	Read count data file, generated in Step 1
`-snp`	File containing base counts of common SNPs, generated in Step 1
`-vcf`	Mutation calls
`-o`	Output file name (prefix)
`-sv`	List of breakpoints, e.g., from rearrangement callers or external segmentation algorithm callers. Each line of the list should contain the chromosome name and position of the breakpoint, separated by a tab. Lines that start with '#' are ignored
`-w`	Minimal window size in kbp. Adjacent genomic windows smaller than this size are joined to reach this lower bound
`-min_r`	Minimal number of reads per partition (in the matched normal only). Similar to the previous parameter, adjacent windows are joined, if necessary, to achieve this threshold
`-min_seg`	Minimal number of genomic partitions that can form a copy-number segment
`-st`	Threshold to determine the sex of the patient by the ratio of the coverage of the Y chromosome to that of the X chromosome. If this ratio is below this threshold (specified by `-st`), the patient sex is estimated to be female
`-alpha`	This parameter determines the sensitivity of the segmentation, for which smaller values lead to a stricter segmentation. The value `-1` leads to a suppression of segmentation, e.g., if an external segmentation is provided (see `-sv` option)
`-ms`	Bandwidth of the median smoother applied to raw copy-number data to eliminate outliers. You can switch off the median smoother by setting `-ms` = 0
`-ns`	Minimal number of SNPs per segment to compute tumor purity by fitting biallelic frequencies. For whole-genome sequencing: `-ns` = 1000 and for whole-exome sequencing: `-ns` = 100 has proven to work reliably for almost all samples
`-minp`	Lower bound of tumor ploidy
`-maxp`	Upper bound of tumor ploidy. These two parameters (`-minp` and `-maxp`) are central quantities to calibrate the copy-number profile; for details see Step 5
`-minpu`	The smallest tumor purity that can be estimated using biallele frequencies. If the optimal purity hits this lower bound, the algorithm automatically switches to mutation-based purity determination
`-f2`	If this flag is set, mutation-based purity estimation is enforced

Parameters of the ***cluster*** module

`-i`	Name of input files (prefix)
`-o`	Name of output files (prefix); if this option is not set, the input name is also used as the output name
`-nbins`	This option sets the number of bins used to construct the cancer-cell fraction histogram; the default value is 100, a change is typically not necessary
`-indel`	A flag option; if this option is set, insertions and deletions (indels) are also included in the mutational clustering. Make sure that the allelic fractions are correctly computed before using this option
`-lambda`	This parameter controls the degree of smoothing to cluster the mutations. Very high values can lead to an under-calling of mutation clusters; the default value is 1×10^{-7}; a change is usually not required

4| Perform the copy-number analysis using the converted mutation call .vcf file <sample>_mutations.vcf, the read-count file <sample>_rcount.txt, and the SNP base-count file <sample>_snps.txt and using the following command. See **Box 1** for a description of the parameters in the command.

```
Sclust cn -rc <sample>_rcount.txt -snp <sample>_snps.txt -vcf <sample>_mutations.vcf
-o <sample>
```

Box 2 | Description of extra fields

DP	Coverage at the position of the mutation in the tumor (required)
DP_N	Coverage at the position of the mutation in the matched normal (required)
AF	Allelic fraction of the mutation in the tumor (required)
AF_N	Allelic fraction of the mutation in the matched normal (typically close to zero; not required; placeholder)
FR	The forward–reverse score; 0 if all reads of the mutation are facing in one direction; 1 if all forward and reverse scores are equally present (not required; placeholder)
TG	Name of the genome partition (not required; placeholder)
DB	If this flag is set, the mutation is a common SNP (mutations at SNP positions are filtered out before mutation clustering)

Generate the following files using the following command:

`<sample>_cn_summary.txt`, `<sample>_allelic_states.txt`, `<sample>_subclonal_cn.txt`, `<sample>_uncorr_cn.seg`, `<sample>_iCN.seg`, `<sample>_muts_expAF.txt`, and `<sample>_cn_profile.pdf`.

This will output the following files.

- <sample>_cn_summary.txt:

sample_name purity ploidy fraction_subclonal_cn sex_estimated status fraction_inconsistent_segs

H2171 0.99 1.92056 0.0871417 m optimum 0.00323938

The sample name is followed by estimates of purity, ploidy, the fraction of subclonal copy number throughout the genome, and the sex of the patient. The status of the copy-number analysis can take the following values: optimum (if an optimal solution was achieved), invariant (if no sufficient copy-number changes were present and the purity was calculated from the mutations), forced (if the `-f2` flag was set), and failed (if the method failed). The last value is the fraction of inconsistent subclonal segments, in the case that no unique copy-number solution was found.

- <sample>_allelic_states.txt

Sample Chromosome Start End Copy_Nr_Raw CopyNr A B LOH Theta Theta_Exp n_SNPs Is_Subclonal_CN Subclonal_P_value Is_Inconsistent_State

H2171 chr1 865532 24022908 2.11065 2 1 1 0 0.121093 0.082753 278 0 0.010588 0

H2171 chr1 24077327 93303221 2.13394 2 1 1 0 0.0862069 0.082753 375 0 0.817891 0

The first column is the sample name followed by the chromosome, and the start and end positions of the copy-number segment. Copy-number estimates of each segment are given in the next columns: (i) uncorrected copy numbers, (ii) corrected integer-valued copy numbers, (iii) major allele copy numbers, and (iv) minor allele copy numbers. If the segment has a LOH, the next column is 1 or otherwise 0. The next two columns provide the observed and model-predicted allelic imbalances, followed by the total number of SNPs, and the subclonality of the segment (0 = no;1 = yes) together with its *P* value. The last column is 1 if no unique solution for the segment was found.

- <sample>_subclonal_cn.txt:

Sample Chromosome Start End Subclonal_CN Clone1_A Clone1_B Clone1_Fraction Clone2_A Clone2_B Clone2_Fraction

H2171 chr2 41557 29404778 3.16433 2 2 0.164326 2 1 0.835674

H2171 chr3 115571276 195493637 2.44044 2 0 0.559561 2 1 0.440439

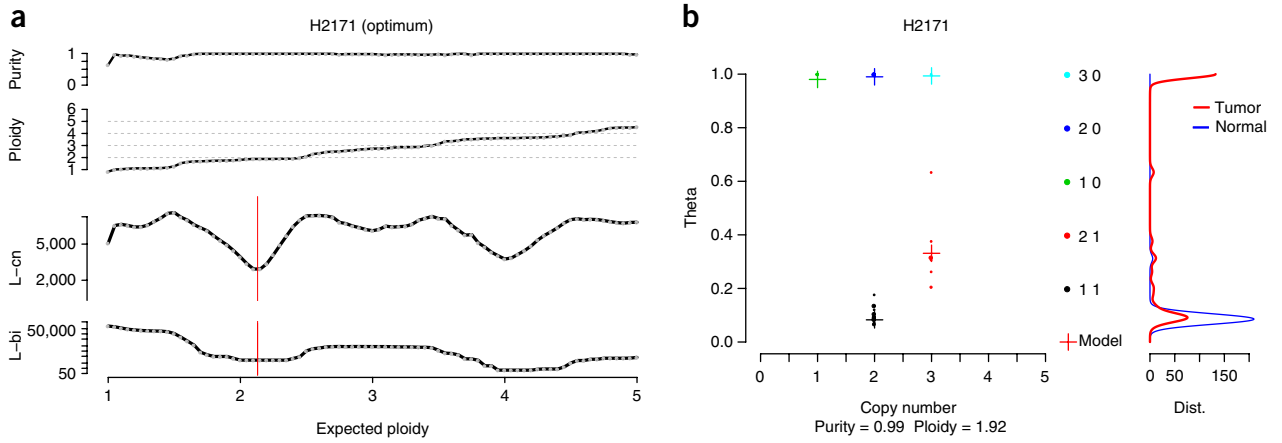

Figure 5 | Example of the first two pages of the <sample>_cn_profile.pdf file. (**a**) Purity and ploidy estimates (*y* axis) as a function of the expected ploidy (*x* axis; free parameter) are shown in the upper panel. The lower panel shows the objective functions of the read ratios (L-cn) and biallelic frequencies (L-bi) on the *y* axis. The exact structure of these objective functions is given in the **Supplementary Note**. The red bar depicts the location of the optimal result. (**b**) Observed allelic imbalances (*y* axis; theta) as a function of the estimated total copy number (*x* axis). Allelic states are depicted by different colors. Model predictions of allelic imbalances are shown as crosses. The distribution of observed allelic imbalances in the tumor (red curve) and matched normal (blue curve) are shown in the right panel.

This file contains the list of all identified subclonal copy numbers, in which the first few columns are again the sample name, chromosome, and the start and end positions of the subclonal copy-number segment. The total subclonal copy number is given in the next column, followed by major and minor copy numbers, and the clonal fraction of clone 1. The last three columns provide the same information for clone 2.

- <sample>_uncorr_cn.seg, <sample>_iCN.seg:

These files contain corrected and uncorrected copy numbers in a standard format that can, e.g., be read by the Integrated Genome Viewer (http://software.broadinstitute.org/software/igv/). The format of these files is described on the webpage given above.

- <sample>_ muts_expAF.txt

```
Mut_ID  Chr  Position  Wt  Mut  AF_obs  Coverage  AF_exp  Mut_Copies  Mut_Copies_
Raw  Is_Subclonal_CN  iCN  P_Is_Clonal

H2171_chr1:3670813_SNM  chr1  3670813  C  A  0.5625  16  0.495  1  1.13636
0  2  0.706872

H2171_chr1:6186793_SNM  chr1  6186793  C  A  0.528571  70  0.495  1  1.06782
0  2  0.71317
```

This file is provided as input for the *cluster* module (Step 5). The first column is a mutation ID, which is composed of the sample name, genomic position, and type of mutation. The following columns are the genomic position, wild type, and mutated base, as well as the observed allelic fraction of the variant. The coverage at the mutation is given in the next column, followed by the expected allelic fraction under the assumption that the mutation is clonal. The estimated and raw multiplicity (number of mutated copies) is given in the next two columns, followed by some copy-number information (copy number of the corresponding segment and its clonality). The last column is the *P* value, which indicates whether or not the mutation is clonal (i.e., testing that cancer-cell fraction = 1).

- <sample>_cn_profile.pdf

This .pdf file is a graphical representation of the results from the copy-number analysis and provides important information for calibrating the copy-number profile (Step 2). It consists of three pages, in which the first page shows the optimization process (**Fig. 5a**). For each expected ploidy (*x* axis) between 1 and 5, the actual purity and ploidy estimates are shown in the top panels. The lower panels show the objective functions based on the read ratios (L-cn) and based on biallelic frequencies (L-bi). The red bar shows the location of the optimal result, which is obtained by the global minimum of L-cn within the

scanning range defined by the parameters: `-minp` and `-maxp`. The second page of <sample>_cn_profile.pdf shows the observed allelic imbalances (denoted by theta) in dependence on the estimated total copy number (**Fig. 5b**). Allelic states are depicted by different colors. The model predictions are shown as crosses, and divergent model predictions are interpreted as subclonal copy-number changes by the algorithm. The distribution of observed allelic imbalances in the tumor and matched normal is shown in the right panel. Finally, the entire copy-number profile across the genome (total and minor allele copy numbers) is depicted on the last page of <sample>_cn_profile.pdf (see, e.g., **Fig. 2a**).

5| Calibrate the copy-number profile. To calibrate the copy-number profiles, adjust the parameters `-minp` and `-maxp` to select another local minimum of L-cn. As an example (**Fig. 5**), setting `-minp` = 3 and `-maxp` = 5 would select the right peak corresponding to a ploidy of 3.7. An experimental assessment of the overall ploidy (e.g., using a fluorescence *in situ* hybridization (FISH) analysis) can further be consulted to reliably calibrate copy-number profiles in doubtful cases.
▲ **CRITICAL STEP** Reliable results are obtained for most of the samples by using default values. However, mainly due to the presence of whole-genome duplications, the mathematically optimal solution might not be the biologically realized situation. This typically occurs in ~20–30% of cases. Unfortunately, there is no universal recipe to detect the samples that need to be calibrated. Common hints are large segments of homozygous deletions, the minimum of L-cn does not coincide with a local minimum in L-bi, the presence of superclonal mutations (cluster with a cancer-cell fraction substantially >1), and a large accumulation of mutations/subclonal copy numbers at a cancer-cell fraction of 0.5.

Mutational clustering ● TIMING <10 s
▲ **CRITICAL** Mutational clustering requires the file <sample>_muts_expAF.txt, which was generated in Step 2.
6| Perform mutational clustering using the following command:

```
Sclust cluster -i <sample>
```

This step generates the following output files:

- <sample>_mclusters.txt

Cluster_ID	CCF_Cluster	Cluster_Peak_Height	Mutations_In_Cluster
0	1.00798	14.9095	214
1	0.78729	1.11972	49

The number of each identified mutation cluster, together with its cancer-cell fraction, is given in the first two columns. Clonal mutations are typically in cluster 0. The next column is the peak height of the cluster, which correlates to the number of mutations assigned to the respective cluster (last column).

- <sample>_cluster_assignments.txt

Mut_ID	Chr	Position	Wt	Mut	CCF	Coverage	Cluster_Id	Cluster_CCF	Proba-bility	P0	P1	P2	P3	P4
H2171_chr1:3670813_SNM	chr1	3670813	C	A	1.13636	16	0	1.00798	0.800866	0.800866	0.188233	0.00998492	0.000890016	2.61627e-05
H2171_chr1:6186793_SNM	chr1	6186793	C	A	1.06782	70	0	1.00798	0.971915	0.971915	0.0280831	1.44228e-06	1.9175e-10	1.87605e-15

The first four columns contain the same information as in <sample>_muts_expAF.txt. The next column is the raw cancer-cell fraction of the mutation, followed by its coverage. The cluster number or ID, its cancer-cell fraction, and its assignment probability are given in the next few columns. Assignment probabilities to all identified clusters are provided in the remaining columns (these values sum up to 1 and can, e.g., be used to generate the co-clustering matrix).

- <sample>_mcluster.pdf

This is a graphical representation of the cancer-cell fraction histogram, together with the identified mutation clusters, that is similar to **Figure 2b**.
? TROUBLESHOOTING

? TROUBLESHOOTING

Troubleshooting advice can be found in **Table 4**.

TABLE 4 | Troubleshooting table.

Step	Problem	Possible reason	Solution
1	Error message: 'please index bam-files'	.bam files are not indexed	Make sure that the .bam files are sorted according to their genomic coordinates. If not, this can be done using SAMtools: 'samtools sort'. After sorting the .bam files, they can be indexed using 'samtools index'
6	Error message: 'QP has reached maximal number of iterations'	The quadratic programming solver did not converge	A larger smoothing parameter (-lambda option) may resolve this problem

● TIMING

Steps 1 and 2, preprocessing of alignment files: depending on your hardware, the extraction of each chromosome can be done sequentially or in parallel, e.g., on large computing clusters. In the case of whole-exome sequencing data, the sequential run can take up to 3 h but only ~15 min if run in parallel. Similarly, the sequential whole genome run takes 15 h, whereas a parallel run requires ~1.5 h
Steps 3–5, copy-number analysis: <5 min
Step 6, mutational clustering: <10 s

ANTICIPATED RESULTS

A successful completion of the protocol results in the following output files: <sample>_rcount.txt and <sample>_snps.txt (Steps 1 and 2); <sample>_cn_summary.txt, <sample>_allelic_states.txt, <sample>_subclonal_cn.txt, <sample>_uncorr_cn.seg, <sample>_iCN.seg, and <sample>_cn_profile.pdf (Steps 4 and 5); and <sample>_mclusters.txt, <sample>_cluster_assignments.txt, and <sample>_mcluster.pdf (Step 6). A detailed description of the structure of these files is given in the PROCEDURE section. To decide whether the fitted copy-number profile is correctly calibrated and of good quality, an inspection of <sample>_cn_profile.pdf is required. Information on how to interpret this file is given in Step 4, and suggestions on how to calibrate the profile are outlined in Step 5. Mutational clustering is typically robust if the copy-number profile is correctly adjusted. Exceptions are discussed in the Troubleshooting section.

Data availability

Data used in this protocol are deposited at the European Genome-phenome Archive (EGA) under accession nos. EGAS00001000925 for the small-cell lung-cancer data and EGAD00001000138 for the breast cancer case PD4120a. The lung adenocarcinoma data are deposited at dbGAP under accession code phs000488.v2.p1. Please note that due to the sensitive nature of these patient datasets, adequate approval from the data provider must be acquired to download them. The procedure for applying for data access is described in the corresponding EGA data access committee (DAC; https://www.ebi.ac.uk/ega/dacs/EGAC00001000064 for EGAS00001000925 and https://www.ebi.ac.uk/ega/dacs/EGAC00001000010 for EGAD00001000138) or dbGA (https://dbgap.ncbi.nlm.nih.gov/aa/wga.cgi?page=login) web pages.

Note: Any Supplementary Information and Source Data files are available in the online version of the paper.

ACKNOWLEDGMENTS We thank the Evolution and Heterogeneity Working Group of the PCAWG initiative for fruitful discussions; P. van Loo, D. Wedge, and S. Dentro for making the Battenberg calls for PD4120a available; and L. Maas for proofreading. The computation was performed on the DFG-funded CHEOPS Cluster of the Regional Computing Centre of Cologne. This work was supported by German Cancer Aid (Deutsche Krebshilfe, grant ID: 109679), the German Ministry of Science and Education (BMBF) as part of the e:Med program (grant nos. 01ZX1303A and 01ZX1406), and the Deutsche Forschungsgemeinschaft (CRU-286, CP2).

AUTHOR CONTRIBUTIONS Y.C. and M.P. conceived the project. Y.C., T.-P.Y., V.A., and M.P. wrote the manuscript. Y.C., T.-P.Y., V.A., and M.P. developed and optimized the algorithm. Y.C., T.-P.Y., V.A., and M.P. performed computational analysis. V.A. and U.L. provided and optimized computing and data infrastructure. All co-authors reviewed the manuscript. All authors read and approved the final manuscript.

COMPETING INTERESTS The authors declare no competing interests.

1. Beerenwinkel, N., Schwarz, R.F., Gerstung, M. & Markowetz, F. Cancer evolution: mathematical models and computational inference. *Syst. Biol.* **64**, e1–25 (2015).
2. Greaves, M. & Maley, C.C. Clonal evolution in cancer. *Nature* **481**, 306–313 (2012).
3. Nowell, P.C. The clonal evolution of tumor cell populations. *Science* **194**, 23–28 (1976).
4. Stratton, M.R., Campbell, P.J. & Futreal, P.A. The cancer genome. *Nature* **458**, 719–724 (2009).
5. Yates, L.R. & Campbell, P.J. Evolution of the cancer genome. *Nat. Rev. Genet.* **13**, 795–806 (2012).
6. Meyerson, M., Gabriel, S. & Getz, G. Advances in understanding cancer genomes through second-generation sequencing. *Nat. Rev. Genet.* **11**, 685–696 (2010).
7. Carter, S.L. *et al.* Absolute quantification of somatic DNA alterations in human cancer. *Nat. Biotechnol.* **30**, 413–421 (2012).

8. Fischer, A., Vazquez-Garcia, I., Illingworth, C.J. & Mustonen, V. High-definition reconstruction of clonal composition in cancer. *Cell Rep.* **7**, 1740–1752 (2014).
9. Gusnanto, A., Wood, H.M., Pawitan, Y., Rabbitts, P. & Berri, S. Correcting for cancer genome size and tumour cell content enables better estimation of copy number alterations from next-generation sequence data. *Bioinformatics* **28**, 40–47 (2012).
10. Ha, G. *et al.* TITAN: inference of copy number architectures in clonal cell populations from tumor whole-genome sequence data. *Genome Res.* **24**, 1881–1893 (2014).
11. Oesper, L., Mahmoody, A. & Raphael, B.J. THetA: inferring intra-tumor heterogeneity from high-throughput DNA sequencing data. *Genome Biol.* **14**, R80 (2013).
12. Van Loo, P. *et al.* Allele-specific copy number analysis of tumors. *Proc. Natl. Acad. Sci. USA* **107**, 16910–16915 (2010).
13. Deshwar, A.G. *et al.* PhyloWGS: reconstructing subclonal composition and evolution from whole-genome sequencing of tumors. *Genome Biol.* **16**, 35 (2015).
14. Jiao, W., Vembu, S., Deshwar, A.G., Stein, L. & Morris, Q. Inferring clonal evolution of tumors from single nucleotide somatic mutations. *BMC Bioinformatics* **15**, 35 (2014).
15. Nik-Zainal, S. *et al.* The life history of 21 breast cancers. *Cell* **149**, 994–1007 (2012).
16. Roth, A. *et al.* PyClone: statistical inference of clonal population structure in cancer. *Nat. Methods* **11**, 396–398 (2014).
17. Malikic, S., McPherson, A.W., Donmez, N. & Sahinalp, C.S. Clonality inference in multiple tumor samples using phylogeny. *Bioinformatics* **31**, 1349–1356 (2015).
18. Sengupta, S. *et al.* Bayclone: Bayesian nonparametric inference of tumor subclones using NGS data. *Pacific Symposium on Biocomputing* 467–478, https://www.worldscientific.com/doi/pdf/10.1142/9789814644730_0044 (2015).
19. Zare, H. *et al.* Inferring clonal composition from multiple sections of a breast cancer. *PLoS Comput. Biol.* **10**, e1003703 (2014).
20. Jiang, Y., Qiu, Y., Minn, A.J. & Zhang, N.R. Assessing intratumor heterogeneity and tracking longitudinal and spatial clonal evolutionary history by next-generation sequencing. *Proc. Natl. Acad. Sci. USA* **113**, E5528–E5537 (2016).
21. Andor, N., Harness, J.V., Muller, S., Mewes, H.W. & Petritsch, C. EXPANDS: expanding ploidy and allele frequency on nested subpopulations. *Bioinformatics* **30**, 50–60 (2014).
22. George, J. *et al.* Comprehensive genomic profiles of small cell lung cancer. *Nature* **524**, 47–53 (2015).
23. Peifer, M. *et al.* Integrative genome analyses identify key somatic driver mutations of small-cell lung cancer. *Nat. Genet.* **44**, 1104–1110 (2012).
24. Peifer, M. *et al.* Telomerase activation by genomic rearrangements in high-risk neuroblastoma. *Nature* **526**, 700–704 (2015).
25. Schramm, A. *et al.* Mutational dynamics between primary and relapse neuroblastomas. *Nat. Genet.* **47**, 872–877 (2015).
26. Gerstung, M. *et al.* The evolutionary history of 2,658 cancers. Preprint at *bioRxiv*, http://dx.doi.org/10.1101/161562 (2017).
27. Seidel, D. *et al.* A genomics-based classification of human lung tumors. *Sci. Transl. Med.* **5**, 209ra153 (2013).
28. Imielinski, M. *et al.* Mapping the hallmarks of lung adenocarcinoma with massively parallel sequencing. *Cell* **150**, 1107–1120 (2012).