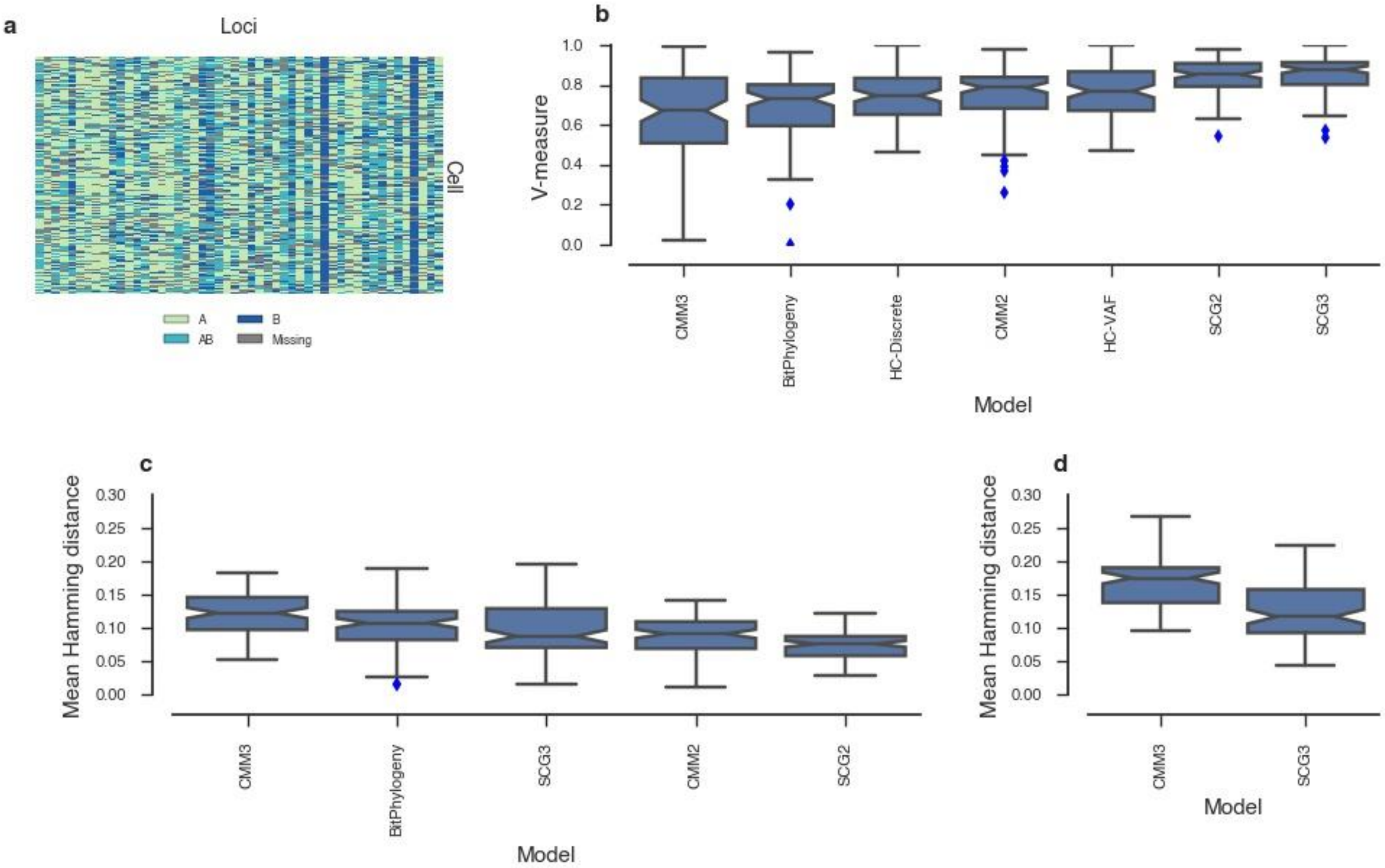
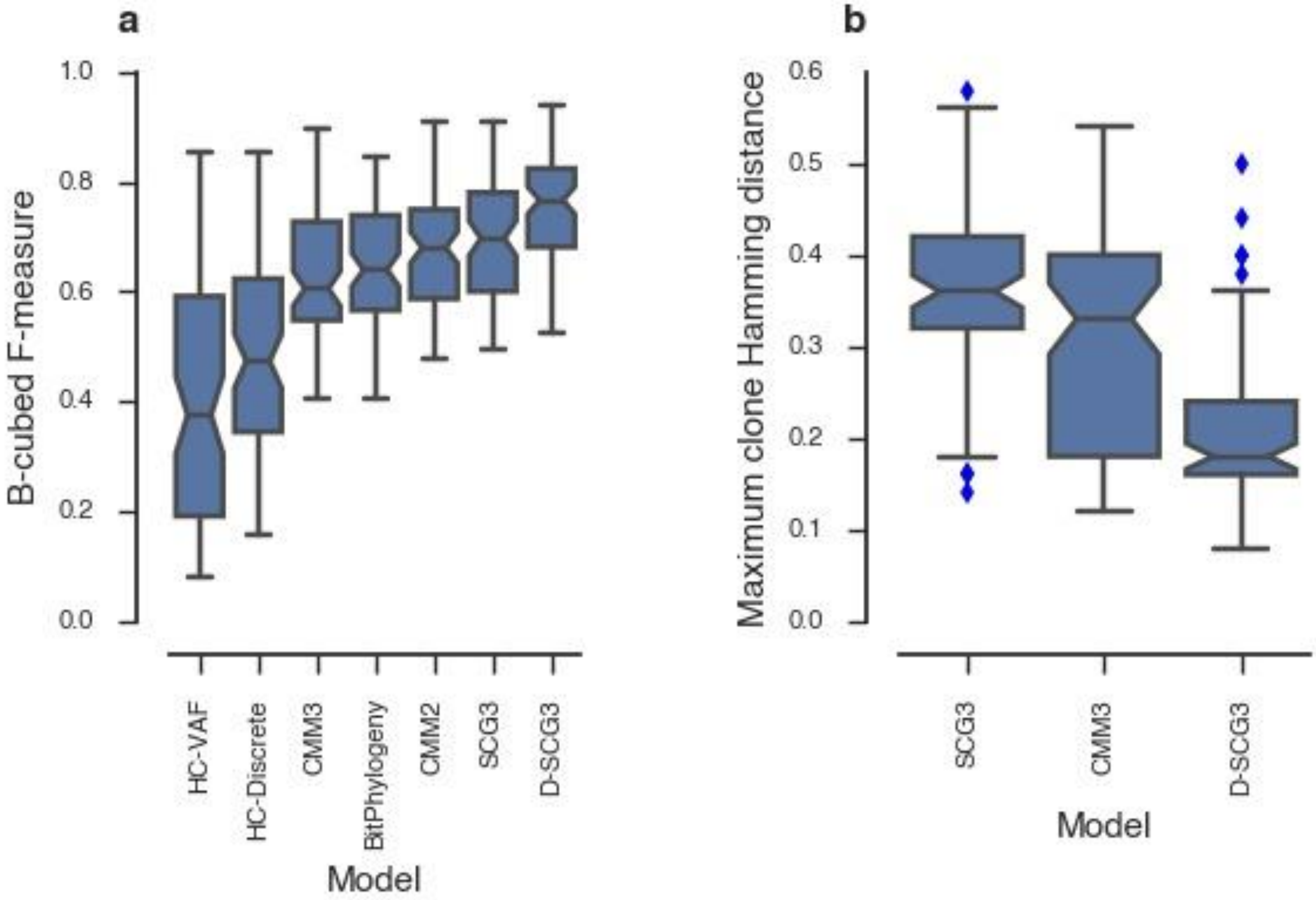


Supplementary Figure 1

Performance comparison using 90 synthetic data without doublets.

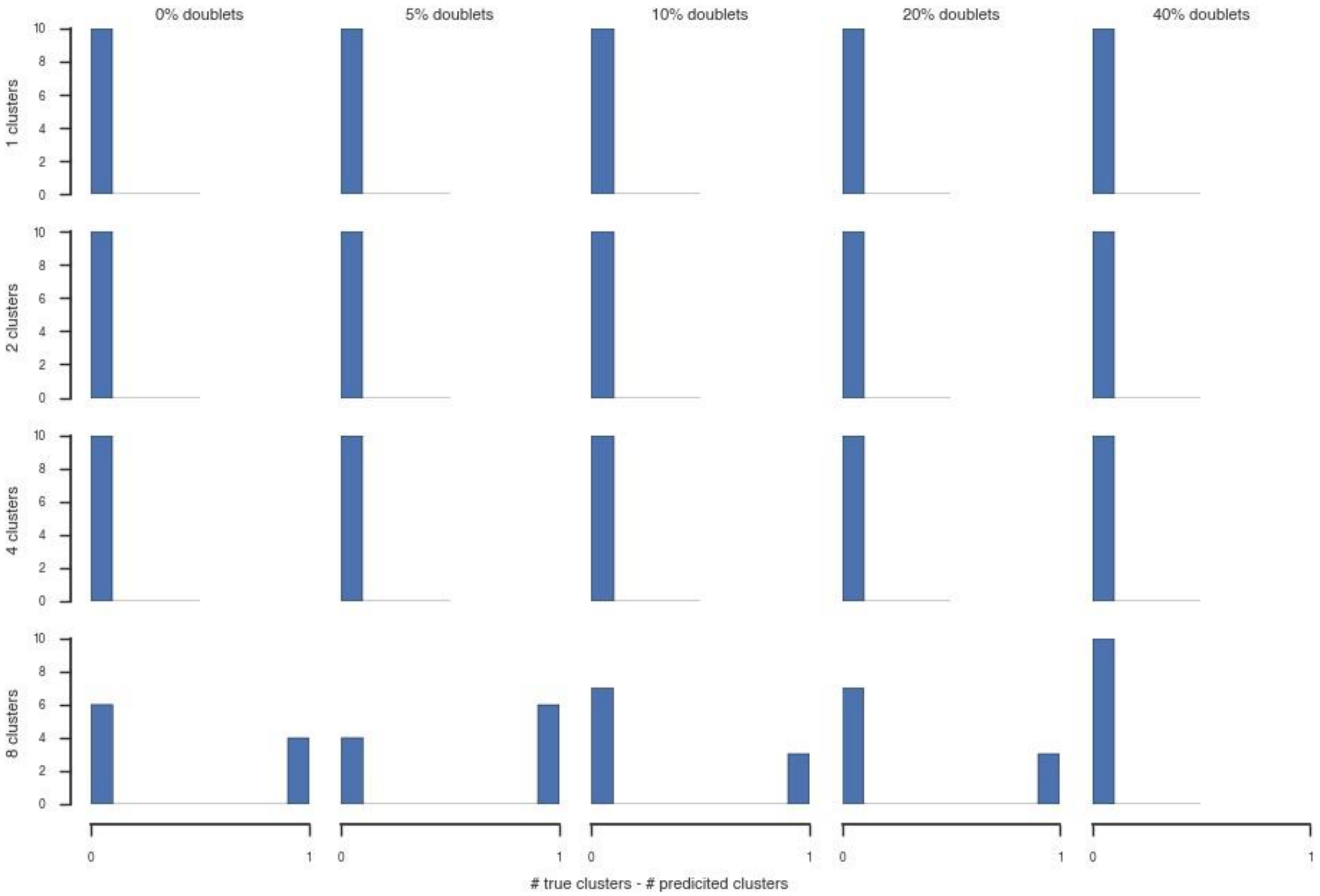
(**a**) Example synthetic data used for benchmarking. (**b**) V-measure metric used to assess clustering performance (higher is better). The mean Hamming distance between predicted genotypes for each cell and their true genotypes in the (c) two-state and (d) three-state representations respectively (lower is better).



Supplementary Figure 2

Performance comparison using 80 synthetic data with doublets.

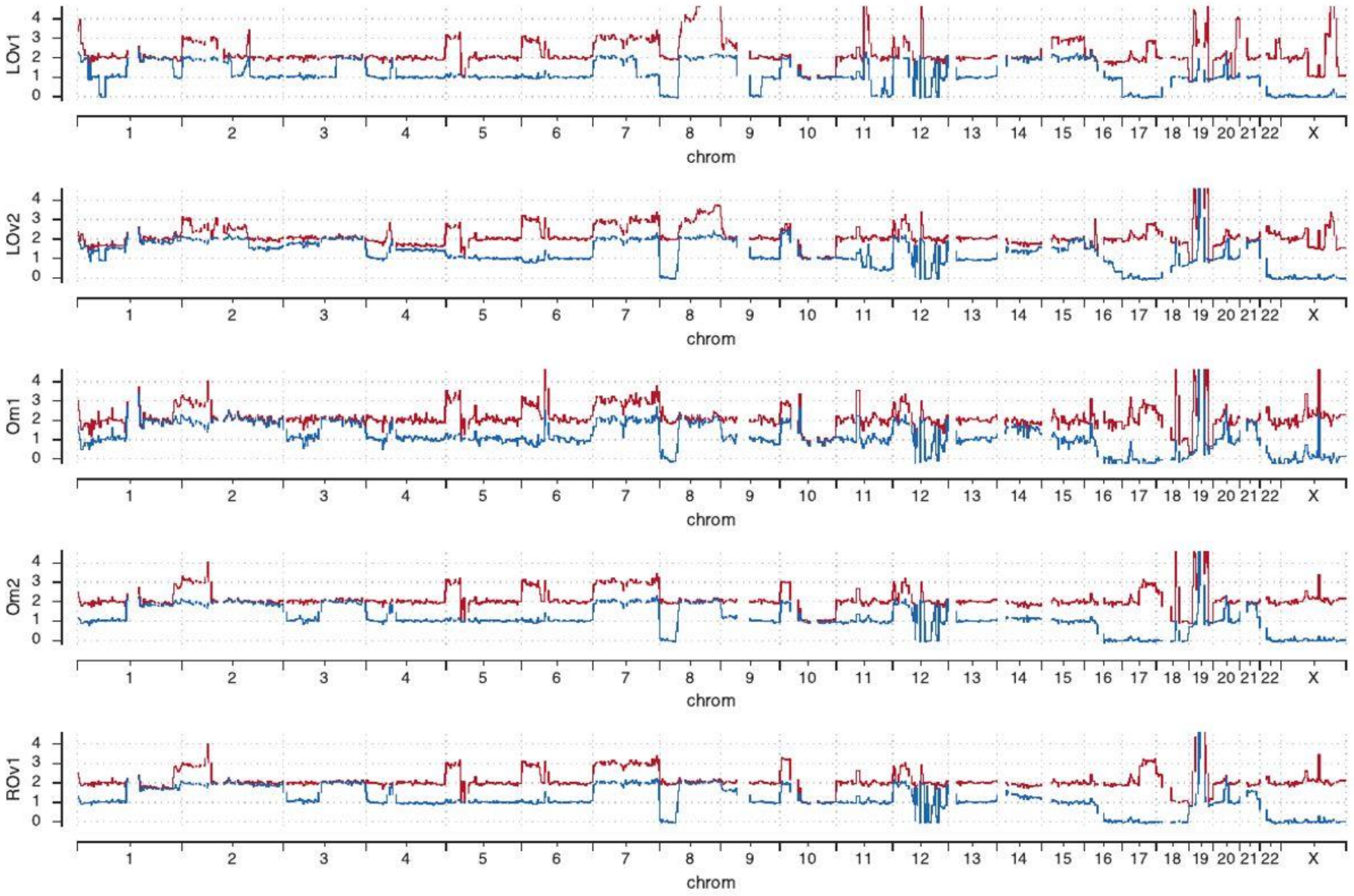
(**a**) F-measure of the B- cubed metric to assess feature allocation performance (higher is better). (**b**) Clone accuracy assessed by the maximum Hamming distance of a predicted clonal genotype to its nearest true clonal genotype in 3 state representation (lower is better).



Supplementary Figure 3

Difference between the number of true clusters and number of clusters predicted by the D-SCG3 model.

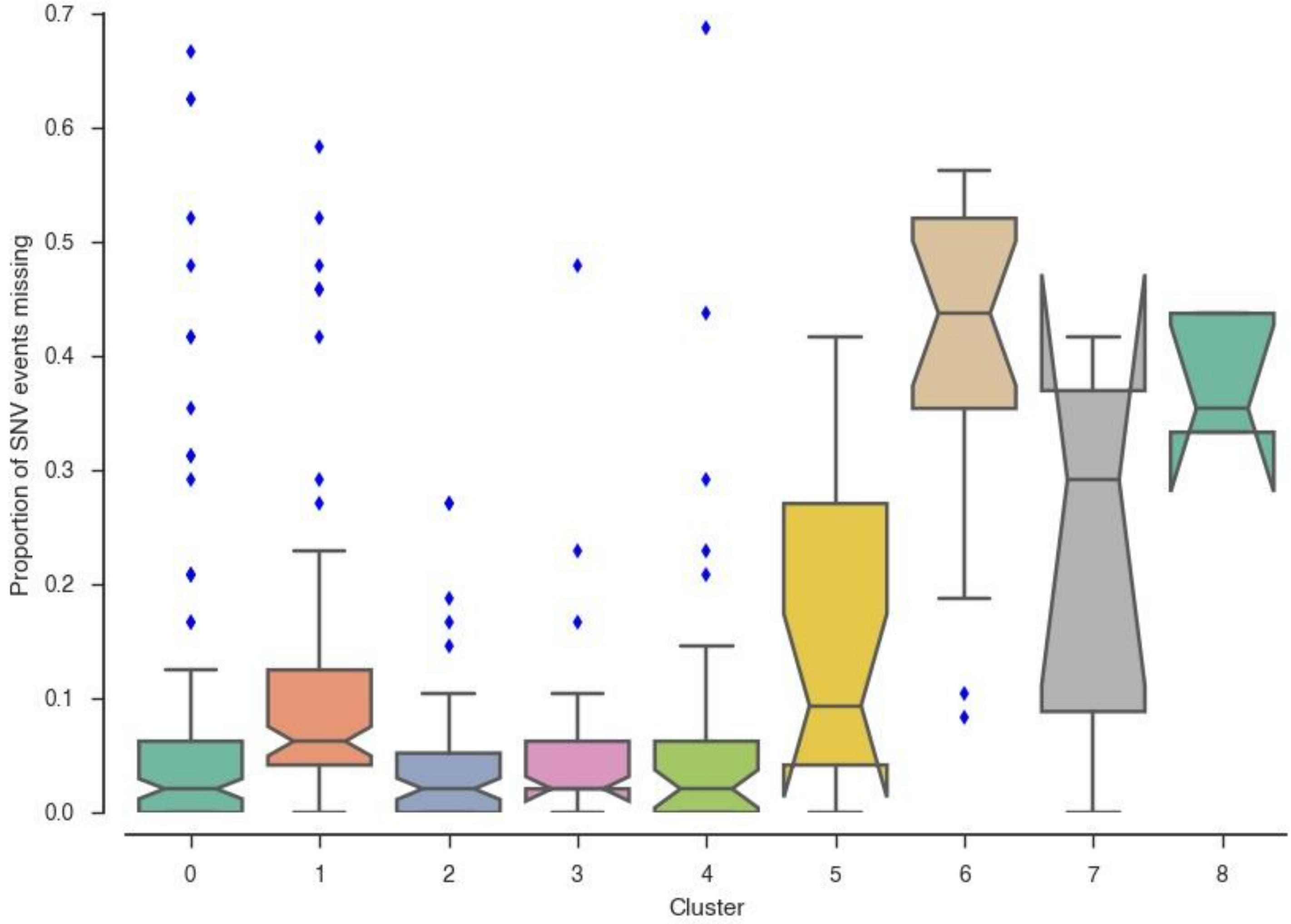
Data was simulated from the D-SCG3 model with 100 data points with 10 replicate datasets per parameter setting. We simulated data across a range of doublet probabilities and number of clusters.



Supplementary Figure 4

Copy number profile for the high grade serous ovarian cancer dataset.

Red lines indicate major copy number and blue lines indicate minor copy number. Note that this tumour likely underwent a genome doubling early in its evolutionary history.



Supplementary Figure 5

Missing data in high grade serous ovarian cancer dataset.

Proportion of missing values per cell for SNV events in the high grade serous ovarian cancer data set. Cells are grouped by cluster.

Supplementary Material for Roth et al., Clonal genotype and population structure inference from single-cell tumor sequencing

Contents

Nature Methods: doi:10.1038/nmeth.3867

1 Supplementary Results

1.1 Synthetic benchmarks

We performed benchmarking using 90 synthetic datasets generated from a range of parameters (**Supplementary Fig. 1a,** see **Supplementary Table 1** for parameter settings and the methods for the simulation method), evaluating cell clustering and clonal genotype prediction methods. We also systematically tested the impact of modelling genotypes of SNVs using three states relative to previous approaches which only consider two states. In the three state regime, we distinguished between the following states: AB (presence of both alleles) and presence (respectively, absence) of the variant allele, denoted B (respectively, A). In contrast, the two-state approach only considers the presence or absence of the B allele. We compared the Categorical mixture model with two (CMM2) and three states (CMM3) as exemplars of models which use a continuous hidden state space for genotypes. Note that the CMM2 model is the same as the BMM[8], but we use the current nomenclature to clarify the comparison. As exemplars of models using a discrete state space for the genotypes, we apply the SCG model using three states with a position specific error rate (SCG3), and the same model restricted to binary data (SCG2) (see the methods for a description of the models). We include two hierarchical clustering approaches as exemplars of methods which are not based on an explicit model: one based on the Euclidean distance of variant allelic frequencies (HC-VAF), and one based on Manhattan distance of the two-state discrete data (HC-Discrete). We also include the previously published BitPhylogeny method[17].

To determine if models significantly differed in performance we applied a two stage procedure[6]. First, we applied the Friedman test to determine if there was a significant difference in model performance across any of the models. We next applied the Nemenyi test to all pairs of models to determine which models showed significantly different performance from each other (p-value < 0.001). All statements of significance in the following paragraphs are with respect to this procedure and the results are summarised in Supplementary Tables **Supplementary Tables 2–4**.

The SCG model significantly outperformed all methods using either zygosity state-space data representation in terms of clustering performance as measured by the V-measure metric (**Supplementary Fig. 1b**, **Supplementary Tables 2** and **5**). The SCG3 model performed better than the SCG2 model, but the performance difference was not significant. The CMM2 model significantly outperformed the CMM3 model, suggesting the inclusion of additional states is not beneficial for the CMM models.

To investigate the accuracy of the predicted genotypes we computed the mean Hamming distance between the predicted genotype of a cell and its true genotype. We did this using both the two-state representation of the genotype, indicating the presence or absence of the variant allele, or the three-state representation of the genotypes. We exclude the hierarchical clustering methods from this analysis as these methods only attempt to cluster data, not predict genotypes. When comparing the SCG3 and CMM3 methods to the two-state representation of genotypes, we mapped the {AB, B} states to the ’present’ state. We could not compare models which only predict two-state genotypes to the true three-state genotypes, so we only include the CMM3 and SCG3 models in this analysis. The most accurate methods to predict clonal genotypes in the two-state representation were the SCG2 and CMM2 methods (**Supplementary Fig. 1c**, **Supplementary Tables 3** and **6**). The SCG2 method slightly outperformed the CMM2 method, but the difference was not significant. For the three-state representation, the SCG3 model significantly outperformed the CMM3 model (**Supplementary Fig. 1d**, **Supplementary Tables 4** and **6**). Taken together these results suggest the SCG model is more accurate than other approaches with respect to both clustering and inference of clonal genotypes.

1.2 Modelling doublets reduces the rate of false positive clonal population prediction

To explore the effect doublets have on clustering performance, and to determine whether our proposed model can detect and correct for them, we simulated 80 datasets with between 5% and 40% of doublet cells and either 100 or 1000 cells (**Supplementary Table 1**).

Benchmarking performance in the presence of doublets was complicated by the fact we needed to assign each data point to one or two clonal genotypes. This is a feature allocation problem, thus we use the B-cube metric[1] in place of the V-measure metric to assess performance. The D-SCG3 method significantly outperformed all approaches except for the SCG3 method (**Supplementary Fig. 2a**, **Supplementary Tables 7** and **8**). We note that the D-SCG3 method outperformed the SCG3 method, but the difference was not statistically significant ($p = 0.007$). The genotype associated with doublet clusters will be a combination of other clusters, and thus not representative of any true genotype. To assess the error this may cause in clonal genotype reconstruction, we computed the maximum Hamming distance between all predicted clonal genotype and the nearest true clonal genotype in the three state representation. The D-SCG3 significantly outperforms the both the CMM3 and SCG3 model with respect to this metric (**Supplementary Fig. 2b**, **Supplementary Tables 9** and **10**).

Our results suggest that the doublet aware D-SCG3 method can correct for doublets and significantly improve the quality of reconstructed genotypes.

1.3 Selecting the number of clones

To test if the solution found by the variational inference algorithm was selecting the correct number of clones we simulated 200 datasets from the D-SCG3 model. We varied the proportion of doublets in the datasets from 0% to 40% and the number of clusters from 1 to 8, with 10 replicates for each parameter setting. We simulated 100 data points with 50 events for each dataset. The results are summarised in **Supplementary Fig. 3**. The inference procedure correctly infers the number of clusters when the true number is 1, 2 or 4. When we increase the number of simulated clusters to 8 the procedure underestimates the number of clusters by 1 in some cases. This is due the presence of small clusters. The proportion of doublets does not seem to negatively effect our ability to estimate the number of clusters. Surprisingly the D-SCG3 model actually performed better when more doublets were present for the simulations with 8 clusters. One explanation maybe that more cells are effectively being sampled with doublets (as doublet data points contribute two cells), so that small clusters become less of an issue.

2 Supplementary Discussion

2.1 Limitations and extensions

The SCG model does not attempt to infer the copy number of loci. Given that we focus on targeted sequencing of point mutations, inference of copy number would be difficult. This is due to the fact that targeted sequencing creates uneven depths of coverage based on primer efficiency in the case of PCR, or hybridisation efficiency in the case of capture experiments. Thus, inference of total copy number from depth of coverage is extremely challenging. Whole genome sequencing of single cells will provide a better experimental design to infer total copy number, as more uniform coverage across the genome can be realised.

The SCG model does not explicitly model copy number. We instead account for copy number variation at loci implicitly, by allowing the error rate to vary in a position specific way. This allows the model to adapt to skew dropout rates at heterozygous loci with differing numbers of copies of each parental allele. Though we do not explore it in this work, we could potentially specify loci specific priors for the error rate by leveraging information from bulk sequencing experiments of the same samples. We note that despite these shortcomings, we were able to apply the method successfully to a HGSOC example which had highly variable copy number. In particular, the method was able to identify that the clonal mutation in TP53 was homozygous which is supported by bulk copy number analysis (**Supplementary Fig. 4**).

The inference method we use is only guaranteed to find a local minima of the variational objective function. This necessitates the need for multiple restarts, with no clear method to determine an appropriate number of restarts. We choose to use as many restarts as our computational budget would allow and provide guidance on selecting the number of restarts in the **Supplementary Note**. Recent research has suggested that using a more structured factorisation of the variational distribution, which breaks fewer dependencies, may lead to improved convergence to the global optimum[11]. This would be a worthwhile avenue of research that could potentially reduce the computational overhead of the method, and lead to better performance. In addition, stochastic variational inference[11 12] is scalable to very large datasets. This would open up the possibility of applying the SCG model to analyse thousands or even millions of cells.

Our method does not address the problem of inferring clonal phylogenies from single cell data. Our tentative conclusion is that incorporating phylogenetic constraints during clustering may be of benefit, but the additional computational burden currently leads to worse performance. Our results show that the phylogenetically naive CMM2 model, significantly outperforms the phylogenetically aware BitPhylogeny model. Thus, using similar data emission distributions and adding phylogenetic constraints does not improve performance. One explanation for this, given the simulation data is compatible with the BitPhylogeny model, is that the Monte Carlo Markov chain sampler used by BitPhylogeny has not been run for a sufficient number of iterations. We did not pursue this further as BitPhylogeny was run for significantly longer than any other method. The high computational cost of running the BitPhylogeny method highlights a key issue with incorporating phylogenetic constraints. Namely, inference for phylogenetic models is significantly more computationally expensive than mixture models. The additional model complexity required to incorporate phylogenetic constraints may also limit our ability to consider other useful extensions, such as doublet modelling. In addition, phylogenetic modelling would only be useful if the evolutionary model is correct. For example, the BitPhylogeny model fails to account for the possibility a mutation may be lost in descendant clones. We suggest that a more fruitful approach to the phylogenetic problem may be to separate the clustering and genotype inference task, from the inference of the phylogeny. The inferred clonal genotypes can be readily used as inputs to classical phylogenetic algorithms. This would allow researchers to leverage the large body of existing software and explore several possible evolutionary models.

2.2 Model variants

There are several variants of the SCG model depending on what features are used. For example, we consider both the SCG2 and SCG3 model when analysing SNV data. The SCG3 model allows for a richer representation of the data leading to more refined genotype predictions. Thus, we expect users would prefer to use the SCG3 model over the SCG2 model. Similarly, we contrast the doublet naive SCG3 and doublet aware D-SCG3 models. If users are certain that no doublets are present in the dataset, then the SCG3 model should be preferred. The SCG3 model is significantly faster and more memory efficient having complexity $O(NKM)$ in time and space, where N is the total number of cells, K is the number of clusters, and M the number of loci. The D-SCG3 approach has complexity $O(NK^2M)$ in both time and space, which can be significantly slower for large K. In addition, the space over which the variational inference algorithm must optimize for the D-SCG3 is likely to be more complex than for the SCG3 model, requiring more random restarts. However, we favour the use of the D-SCG3 model as most datasets we have examined contain doublets. Both variants of the models run in a few minutes for datasets of 100s of cells and ~100 loci on a modern laptop. Random restarts are required, which can scale the analysis time to several hours, depending on the number of restarts. However, restarts can be performed in parallel, reducing actual running time to minutes depending on computational resources.

3 Supplementary Note 1

3.1 Single Cell Genotyper Model

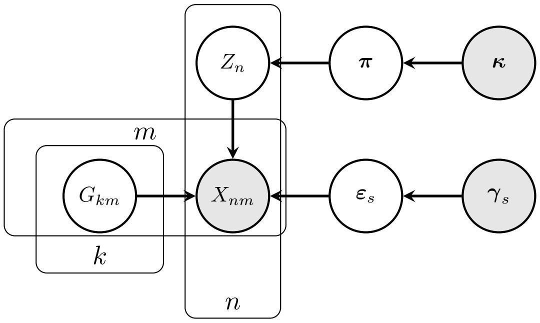
We derive the single cell genotyper (SCG) model in the following section. To ease the exposition we derive the model in steps, starting with the simplest model, progressively extending until the complete model is described.

3.1.1 Model description

The goal of the SCG model is to infer the clonal population structure and clonal genotypes given M loci measured for N cells, represented by a matrix $\boldsymbol{X}$ with dimensions N rows and M columns. In **Supplementary Note Fig. 1** we show the probabilistic graphical model and its distributional assumptions. Let $\mathcal{S}$ be the set of possible genotype values for a given locus, for example, $\mathcal{S} = \{\text{A, AB, B}\}$ for SNV data. We assume that data observed for a locus is noisy, such that given that the true genotype state of loci is $s \in \mathcal{S}$, the probability of observing a value $t \in \mathcal{S}$ is given by $\varepsilon_{st} \in [0,1]$. Intuitively, ε_{ss} represents the probability of measuring the genotype correctly, and $\sum_{t \neq s} \varepsilon_{st}$ represents the probability of making an error.

To share statistical strength among cells, we assume a set of $K \ll N$ clonal populations from which we draw the N sampled cells. A clonal population is a group of cells with identical genotype and distinct clonal populations differ in genotype at at least one of the M measured loci. We use a mixture model[15], with the components of the mixture representing clonal populations, and the component-specific parameters representing clonal genotypes. Formally, let $Z_n = k \in \{1, \ldots, K\}$ indicates the clonal population of cell n. The genotype of clone k at locus m is given by G_{km}, where any possible value from $\mathcal{S}$ is equally likely *a priori*. To maintain a finite mixture model, we assume that K is known. However, we will discuss how to infer K in section 3.1.8.

There are several model hyper-parameters that need to be set. $\boldsymbol{\kappa}$ controls the distribution over clone prevalence, or in mixture model terminology the component mix-weights. We take all components of the vector $\boldsymbol{\kappa}$ to be equal, resulting in a symmetric Dirichlet distribution. We further set this value to 1 for all analyses performed, but leave this as a configurable parameter in the software. The hyper-parameter γ_{st} specifies our prior belief about how likely we are to observe state t when the true genotype state is s. This hyper-parameter should be set using domain knowledge, so as to create an informative prior on ε_{st} (this avoids issues related to the partial identifiability of the model). For example, we use the values in **Supplementary Note Table 1** for $\boldsymbol{\gamma}$ when applying the SCG model to analyze SNV data in the synthetic benchmark data. These values place a high prior probability that the observed state matches the true genotype state when only one allele is present in the genotype. When the genotype has both alleles, we assume that the observed state is uniformly drawn from the observable states. This corresponds to prior knowledge that allelic



$$
\begin{aligned}
\boldsymbol{\pi}|\boldsymbol{\kappa} &\sim \text{Dirichlet}(\boldsymbol{\pi}|\boldsymbol{\kappa})\\
Z_n|\boldsymbol{\pi} &\sim \text{Categorical}(Z_n|\boldsymbol{\pi})\\
G_{km} &\sim \text{Uniform}(G_{km}|\mathcal{S})\\
\boldsymbol{\varepsilon}_s|\boldsymbol{\gamma}_s &\sim \text{Dirichlet}(\boldsymbol{\varepsilon}_s|\boldsymbol{\gamma}_s)\\
X_{nm}|\boldsymbol{\varepsilon},\boldsymbol{G},Z_n = k &\sim \text{Categorical}(X_{nm}|\boldsymbol{\varepsilon}_{G_{km}})
\end{aligned}
$$

Supplementary Note Figure 1: Probabilistic graphical model representing the basic SCG model. Shaded nodes represent observed values or fixed values, while a posterior distribution over the values of the un-shaded nodes is approximated using a Variational Bayesian (VB) method.

s\t	A	AB	B
A	9	0.5	0.5
AB	1	1	1
B	0.5	0.5	9

Supplementary Note Table 1: Example parameter settings for the $\boldsymbol{\gamma}$ parameter when using SNV data. The rows correspond to hidden (genotype) states and the columns correspond to observed states. Each row thus represents a setting of $\boldsymbol{\gamma}_s$ for $s \in \{A, AB, B\}$. Values are pseudo-counts in the Dirichlet distribution for $\boldsymbol{\varepsilon}_s$.

dropout only affects heterozygous loci.

3.1.2 Extension for position specific error rates

As discussed in the previous section, ε_{st} represents the probability of observing state t given the genotype has state s. In the basic model this parameter is shared across all loci. We can relax this and allow for locus specific error rates, ε_{mst}, where we now have an additional index for locus m. For SNVs this may be beneficial if the copy number exhibits substantial variability. For example, the probability of allelic dropout for the mutational genotype AB is expected to be higher than for $AABB$ since the latter genotype has twice as many copies of each allele. With more copies of each allele the chance of failing to amplify one will decrease. This has been noted in previous studies, which have selected cells in the G2/M cell cycle stage (where cells have duplicated the genome before division) to reduce dropout[16]. Similarly, the chance of dropping an allele may increase if the allele relatively less common than the other allele in the mutational genotype. We may expect a larger proportion of measurements of the B observed state for the genotype $ABBB$ than the A observed state. We note the prior on ε_{mst} could be trivially made to be position dependent, using, for example, information about coincident copy number variation to inform the prior.

3.1.3 Extension to multiple samples

If data points are measured from multiple related tissue samples it may be beneficial to include this information in the model[2,5,7,9,13]. Related samples will likely contain clones from the same phylogeny, for example spatially separated tumour masses from a patient; temporally acquired biopsies from a patient; cell line or xenograft samples from multiple passages. In this extension, we assume the input data comes from I related tissue samples and we sequence N_i cells from sample i. Let the proportion of cells from clonal population k in sample i be given by π_{ik}. The modification to the basic model is then given by

$$\begin{aligned}
\boldsymbol{\pi}_i|\boldsymbol{\kappa} &\sim \text{Dirichlet}(\boldsymbol{\pi}_i|\boldsymbol{\kappa}) \\
Z_{in}|\boldsymbol{\pi} &\sim \text{Categorical}(Z_{in}|\boldsymbol{\pi}) \\
G_{km} &\sim \text{Uniform}(G_{km}|\mathcal{S}) \\
\boldsymbol{\varepsilon}_s|\boldsymbol{\gamma}_s &\sim \text{Dirichlet}(\boldsymbol{\varepsilon}_s|\boldsymbol{\gamma}_s) \\
X_{inm}|\boldsymbol{\varepsilon},\boldsymbol{G},Z_{in}=k &\sim \text{Categorical}(X_{inm}|\boldsymbol{\varepsilon}_{G_{km}})
\end{aligned}$$

The posterior distribution for $\boldsymbol{\pi}_i$ gives a measure of the prevalence of the clones in each sample. This posterior will be corrected for doublets and uncertainty in the assignment of data points to clusters. In addition we can use the posterior distribution of $\boldsymbol{\pi}_i$ to quantify uncertainty in the prevalence estimates.

3.1.4 Extension to model doublets

We can extend the model to handle the case where some data points result from measuring two cells. We assume that measuring two cells is a rare event, and measuring more than two cells is extremely rare. Thus we will only focus on the extension to two cells, or doublets, with higher numbers of cells assumed to be measured sufficiently infrequently to be of negligible impact. The basic logic developed to handle two cells could be extended to handle more cells, but the number of combinations of clusters grows exponentially with the maximum number of cells that could be measured by a data point.

To model multiple cell measurements, we need to define the expected genotype state when two cells are measured together. To that end we need to define a binary operation, $\oplus$, that describes how we combine genotype states. In

Supplementary Note Table 2 we show how to defined $\oplus$ for SNV data. With this definition, when we combine two cells with same genotype which is homozygous, the combined state is homozygous for that allele. All other combinations yield a heterozygous state. For presence/absence data such as a binary representation of SNVs or rearrangement breakpoints, we can use a *logical or* to define $\oplus$.

We introduce the variable Y_n which is 0 when data point n is a single cell measurement and 1 when it is a doublet measurement. We let the probability of sampling a doublet be given by δ. We redefine the variable $Z_n \in \{1,\ldots,K\}$ from the previous sections as Z_n^1 to indicate the clone of origin if data point n is a single cell measurement. We introduce a new variable $Z_n^2 \in \{(k_1,k_2)|k_1,k_2 \in \{1,\ldots,K\}\}$ to indicate the clone of origin for each of the two cells if data point n is a doublet measurement. Define $\boldsymbol{\pi} \otimes \boldsymbol{\pi} = (\pi_1 \cdot \pi_1, \pi_1 \cdot \pi_2, \ldots, \pi_K \cdot \pi_K) \in \mathbb{R}^{K^2}$ to be the vector of probabilities of all doublet combinations. The extended model is then defined as

$$\begin{aligned}
\delta|\alpha,\beta &\sim \mathrm{Beta}(\delta|\alpha,\beta) \\
Y_n|\delta &\sim \mathrm{Bernoulli}(Y_n|\delta) \\
\boldsymbol{\pi}|\boldsymbol{\kappa} &\sim \mathrm{Dirichlet}(\boldsymbol{\pi}|\boldsymbol{\kappa}) \\
Z_n^1|\boldsymbol{\pi} &\sim \mathrm{Categorical}\left(Z_n^1|\boldsymbol{\pi}\right) \\
Z_n^2|\boldsymbol{\pi} &\sim \mathrm{Categorical}\left(Z_n^2|\boldsymbol{\pi}\otimes\boldsymbol{\pi}\right) \\
G_{km} &\sim \mathrm{Uniform}(G_{km}|\mathcal{S}) \\
\boldsymbol{\varepsilon}_s|\boldsymbol{\gamma}_s &\sim \mathrm{Dirichlet}(\boldsymbol{\varepsilon}_s|\boldsymbol{\gamma}_s) \\
g_{nm}|\boldsymbol{G},Y_n,Z_n^1 = k, Z_n^2 = (k_1,k_2) &= \begin{cases} G_{km} & Y_n = 0 \\ G_{k_1 m} \oplus G_{k_2 m} & Y_n = 1 \end{cases} \\
X_{nm}|\boldsymbol{\varepsilon},g_{nm} &\sim \mathrm{Categorical}(X_{nm}|\boldsymbol{\varepsilon}_{g_{nm}})
\end{aligned}$$

As we expect doublets to be rare, we use the prior on δ to encourage this prior information. For all runs reported in this work we set $\alpha = 1$ and $\beta = 99$.

In general, modelling doublets is not expected to affect performance if no doublets are present. In such a scenario, the inferred value of δ is expected to be close to 0, inducing a high probability for the variables Y_n's to be equal to zero, i.e. selecting the single cell explanation of the data. However, convergence issues during inference may cause worse performance. In addition, considerable computational complexity is added by considering doublets. The memory required to store cluster posteriors for the doublet naive model is $O(NMK)$, whereas the doublet model requires $O(NMK^2)$. We leave the option to use the doublet naive model as an option in the software. The user should select this model if they feel no doublets are likely in their data.

$\oplus$	A	AB	B
A	A	AB	AB
AB	AB	AB	AB
B	AB	AB	B

Supplementary Note Table 2: Definition of the binary operator $\oplus$ used to combine SNV genotypes of two cells.

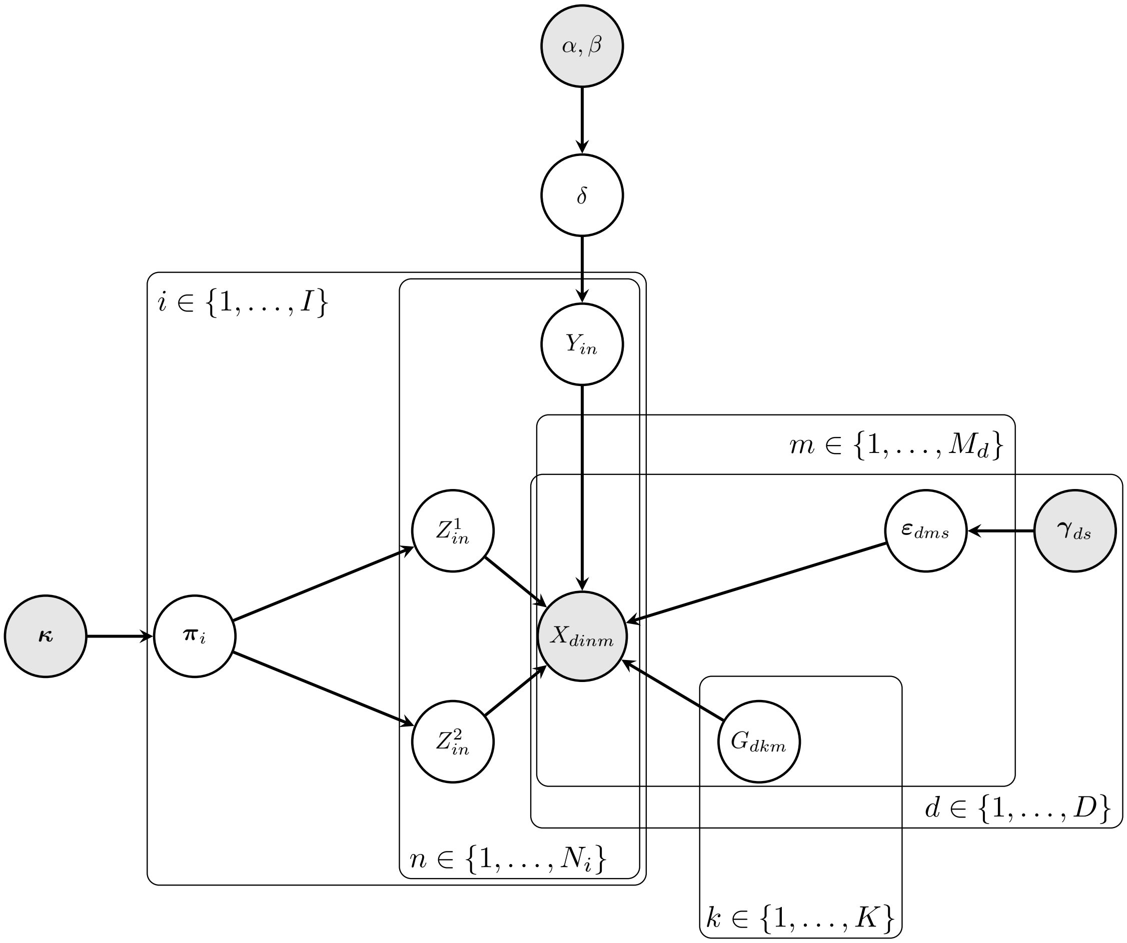
3.1.5 Extension to multiple data types

We can extend the basic model to handle multiple data types measured concurrently on the same data points. Assume that we measure D separate data types, with M_d events for data type d. Further define $\mathcal{S}_d$ be the set of states for data type d. We define the probability of observing state $t \in \mathcal{S}_d$ given genotype state $s \in \mathcal{S}_d$ as ε_{dst} with corresponding prior parameter γ_{dst}. Let the genotype of the k^{th} clone at event m for data type d be G_{dkm}. The data is now represented by a three dimensional ragged array, $\boldsymbol{X}$, with dimensions (D,N,M_d). With this notation the model can be extended as follows

$$
\begin{aligned}
\boldsymbol{\pi}|\boldsymbol{\kappa} &\sim \text{Dirichlet}(\boldsymbol{\pi}|\boldsymbol{\kappa}) \\
Z_n|\boldsymbol{\pi} &\sim \text{Categorical}(Z_n|\boldsymbol{\pi}) \\
G_{dkm} &\sim \text{Uniform}(G_{dkm}|\mathcal{S}_d) \\
\boldsymbol{\varepsilon}_{ds}|\boldsymbol{\gamma}_{ds} &\sim \text{Dirichlet}(\boldsymbol{\varepsilon}_{ds}|\boldsymbol{\gamma}_{ds}) \\
X_{dnm}|\boldsymbol{\varepsilon},\boldsymbol{G},Z_n = k &\sim \text{Categorical}(X_{dnm}|\boldsymbol{\varepsilon}_{dG_{dkm}})
\end{aligned}
$$

3.1.6 Full featured model

All of the extensions discussed previously can be combined into a single full featured model presented in **Supplementary Note Fig. 2**. A full list of model variables is provided in **Supplementary Note Table 4** with a description of the associated indices in **Supplementary Note Table 3**.



$$
\begin{aligned}
\delta|\alpha,\beta &\sim \text{Beta}\left(\delta|\alpha,\beta\right)\\
Y_{in}|\delta &\sim \text{Bernoulli}\left(Y_{in}|\delta\right)\\
\boldsymbol{\pi}_i|\boldsymbol{\kappa} &\sim \text{Dirichlet}\left(\boldsymbol{\pi}_i|\boldsymbol{\kappa}\right)\\
Z^1_{in}|\boldsymbol{\pi}_i &\sim \text{Categorical}\left(Z^1_n|\boldsymbol{\pi}_i\right)\\
Z^2_{in}|\boldsymbol{\pi}_i &\sim \text{Categorical}\left(Z^2_{in}|\boldsymbol{\pi}_i \otimes \boldsymbol{\pi}_i\right)\\
G_{dkm} &\sim \text{Uniform}(G_{dkm}|\mathcal{S}_d)\\
\boldsymbol{\varepsilon}_{dms}|\boldsymbol{\gamma}_{ds} &\sim \text{Dirichlet}\left(\boldsymbol{\varepsilon}_{dms}|\boldsymbol{\gamma}_{ds}\right)\\
g_{dnm}|\boldsymbol{G},Y_{in},Z^1_{in}=k,Z^2_{in}=(k_1,k_2) &= \begin{cases} G_{dkm} & Y_{in}=0 \\ G_{dk_1m} \oplus G_{dk_2m} & Y_{in}=1 \end{cases}\\
X_{idnm}|\boldsymbol{\varepsilon},g_{dnm} &\sim \text{Categorical}(X_{dinm}|\boldsymbol{\varepsilon}_{dg_{dnm}})
\end{aligned}
$$

Supplementary Note Figure 2: Probabilistic graphical model representing the SCG model. Shaded nodes represent observed values or fixed values, while the values of un-shaded nodes are learned using VB.

Index	Range	Description
i	$\{1,\ldots,I\}$	Index of tissue sample.
n	$\{1,\ldots,N_i\}$	Data point index. Depends on tissue sample.
d	$\{1,\ldots,D\}$	Data type index.
m	$\{1,\ldots,M_d\}$	Locus index. Depends on data type.
k	$\{1,\ldots,K\}$	Clone (cluster) index.
s	$\mathcal{S}_d$	Genotype state index. Depends on data type.
t	$\mathcal{T}_d$	Observed state index. Depends on data type.

Supplementary Note Table 3: Indices used in equations.

Variable	Range	Description	Comments
G_{kdm}	$\{1,\ldots,\lvert\mathcal{S}_d\rvert\}$	State of clone k at locus m for data type d.	
X_{idnm}	$\{1,\ldots,\lvert\mathcal{T}_d\rvert\}$	Observed state of data point n from sample i at locus m for data type d.	
ε_{dst}	$[0,1]$	Probability of observing state t given that the hidden state is s for data type d.	$\sum_{t=1}^{T_d} \varepsilon_{dst} = 1$
δ	$[0,1]$	Probability a data point is a doublet.	
π_{ik}	$[0,1]$	Proportion of cells from clone k in sample i.	$\sum_{k=1}^{K} \pi_{ik} = 1$
Y_{in}	$\{0,1\}$	Variable indicating if data point n in sample i is a doublet.	
Z_{in}^1	$\{1,\ldots,K\}$	Variable indicating which clone data point n from sample i belongs to if not a doublet.	
Z_{in}^2	$\{(k_1,k_2)\lvert k_1,k_2 \in \{1,\ldots,K\}\}$	Variable indicating which clones contribute cells to observation i in sample n if it is a doublet.	
γ_{dst}	$(0,\infty)$	Dirichlet prior for ε_{dst}.	
κ_k	$(0,\infty)$	Dirichlet prior π_{ik}.	
α,β	$(0,\infty)$	Beta prior for δ.	

Supplementary Note Table 4: Model variables.

3.1.7 Inference

We use a mean field variational method to infer the model parameters[3]. Variational inference seeks to approximate the posterior distribution $p(\theta|X)$ by a simpler distribution $q(\theta)$. Specifically we seek to minimize the Kullback–Leibler divergence between $q(\theta)$ and $p(\theta|X)$ given by

$$\begin{aligned} D_{\text{KL}}(q(\theta)|p(\theta|X)) &= \int q(\theta)\log\left(\frac{q(\theta)}{p(\theta|X)}\right)d\theta \\ &= \int q(\theta)\log\left(\frac{q(\theta)}{p(X,\theta)}\right)d\theta + \log p(X) \\ &= D_{\text{KL}}(q(\theta)|p(X,\theta)) + \log p(X) \end{aligned}$$

Since $\log p(X)$ does not depend on q, we can minimize $D_{\text{KL}}(q(\theta)|p(\theta|X))$ with respect to q by minimizing $D_{\text{KL}}(q(\theta)|p(X,\theta))$. The latter term is simpler to work with as it does not depend on the normalization constant of the posterior, which is assumed intractable to compute. Note that

$$\begin{aligned} D_{\text{KL}}(q(\theta)|p(\theta|X)) &\geq 0 \\ \Rightarrow \log p(X) &\geq -D_{\text{KL}}(q(\theta)|p(X,\theta)) \\ &= \mathbb{E}_q[\log p(X,\theta) - \log q(\theta)] \end{aligned}$$

hence $-D_{\text{KL}}(q(\theta)|p(X,\theta))$ provides a lower bound on the *evidence* of the model. In practice we maximize the evidence lower bound (ELBO), $\mathbb{E}_q[\log p(X,\theta) - \log q(\theta)]$, rather than minimize $D_{\text{KL}}(q(\theta)|p(X,\theta))$.

The standard mean field approximation assumes that q may be written as a product of distributions which depend on only a single model variable.

$$q(\theta) = \prod_i q_i(\theta_i)$$

The assumption that each q_i depends on a a single model variable can be relaxed so that each q_i depends on a subset of variables, provided the subsets are disjoint.

It is a standard result[3] that the optimal value of $q_i(\theta_i)$ can be obtained by computing

$$q_i(\theta_i) \propto \exp\left(\mathbb{E}_{\prod_{j\neq i} q_j(\theta_j)}\left[\log p(X,\theta)\right]\right)$$

To optimize the full distribution, q, we iteratively update each q_i conditioned on the previous value of $\{q_j\}_{j \neq i}$. This procedure is guaranteed to decrease $D_{\mathrm{KL}}(q(\theta)|p(X,\theta))$ which can be monitored to assess convergence.

For the SCG model we assume the following factorization.

$$\begin{aligned}
q(\boldsymbol{G},\boldsymbol{Y},\boldsymbol{Z}^{\mathbf{1}},\boldsymbol{Z}^{2},\boldsymbol{\pi},\boldsymbol{\varepsilon}) &= q(\boldsymbol{G})q(\boldsymbol{Y},\boldsymbol{Z}^{\mathbf{1}},\boldsymbol{Z}^{2})q(\boldsymbol{\pi})q(\boldsymbol{\varepsilon}) \\
&= \left[\prod_{k=1}^{K}\prod_{d=1}^{D}\prod_{m=1}^{M} q(G_{kdm})\right]\left[\prod_{i=1}^{I}\prod_{n=1}^{N_i} q(Y_{in},Z^1_{in},Z^2_{in})\right] \times \\
&\quad \left[\prod_{d=1}^{D}\prod_{s=1}^{S_d} q(\boldsymbol{\varepsilon}_{ds})\right]\left[\prod_{i=1}^{I} q(\boldsymbol{\pi_i})\right]
\end{aligned}$$

The first line follows from the mean field approximation, while the second line follows from conditional independence structure of the model. It should be noted that we do not break the dependencies between $Y_{in}, Z^1_{in}, Z^2_{in}$ since the appear together in $q(Y_{in}, Z^1_{in}, Z^2_{in})$.

To express the updates for the optimization procedure more concisely we introduce the following variables

$$\begin{aligned}
\xi_{ink} &= \mathbb{E}[\mathbb{I}(Y_{in} = 0, Z^1_{in} = k)] \\
\zeta_{ink\ell} &= \mathbb{E}[\mathbb{I}(Y_{in} = 1, Z^2_{in} = (k,\ell))] \\
\eta_{dkms} &= \mathbb{E}[\mathbb{I}(G_{dkm} = s)] \\
x_{dinmt} &= \mathbb{I}(X_{dinm} = t) \\
\mu_{dmst} &= \mathbb{E}[\log \varepsilon_{dmst}] \\
\nu_{ik} &= \mathbb{E}[\log \pi_{ik}]
\end{aligned}$$

The expectations are taken with respect to the variational distribution q, excluding contributions from terms inside the expectation.

Conveniently q_i is either a Dirichlet or Categorical distribution for all parameters. Thus the expectations defined in the previous equations can be computed analytically. Below we give the updated parameter values for q when q is a Dirichlet. When q is a Categorical distributions, we give the formulas to compute all values in the range of the associated random variable up to the normalization constant.

$$
\begin{aligned}
\bar{\gamma}_{dmst} &= \gamma_{dmst} + \\
& \sum_{i=1}^{I}\sum_{n=1}^{N_i}\sum_{k=1}^{K} \xi_{ink}\eta_{dkms}x_{dinmt} + \\
& \sum_{i=1}^{I}\sum_{n=1}^{N_i}\sum_{k=1}^{K}\sum_{\ell\neq k}^{K}\sum_{u,v\in\mathcal{S}_d:u\oplus v=s} \zeta_{ink\ell}\eta_{dkmu}\eta_{d\ell mv}x_{dinmt} + \\
& \sum_{i=1}^{I}\sum_{n=1}^{N_i}\sum_{k=1}^{K}\sum_{u\in\mathcal{S}_d:u\oplus u=s} \zeta_{inkk}\eta_{dkmu}x_{dinmt} \\
q(\boldsymbol{\varepsilon}_{ds}) &\sim \mathrm{Dirichlet}(\bar{\boldsymbol{\gamma}}_{ds})
\end{aligned}
$$

$$
\begin{aligned}
\log q(G_{dkm}=s) &\propto \log\left(\frac{1}{|\mathcal{S}_d|}\right) + \\
& \sum_{i=1}^{I}\sum_{n=1}^{N_i}\sum_{t\in\mathcal{S}_d} \xi_{ink}\mu_{dst}x_{dinmt} + \\
& 2\sum_{i=1}^{I}\sum_{n=1}^{N_i}\sum_{\ell\neq k}^{K}\sum_{w\in\mathcal{S}_d}\sum_{u:s\oplus u=w} \zeta_{ink\ell}\eta_{d\ell mu}\mu_{dwt}x_{dinmt} + \\
& \sum_{u\in\mathcal{S}_d:s\oplus s=u} \zeta_{inkk}\mu_{dut}x_{dinmt}
\end{aligned}
$$

$$
\begin{aligned}
\log q(Y_n=0, Z_n^1=k) &\propto \mathbb{E}[\log(1-\delta)] + \nu_{ik} + \\
& \sum_{d=1}^{D}\sum_{m=1}^{M_d}\sum_{s\in\mathcal{S}_d}\sum_{t\in\mathcal{S}_d} \eta_{dkms}\mu_{dst}x_{dinmt} \\
\log q(Y_n=1, Z_n^2=(k,\ell)) &\propto \mathbb{E}[\log\delta] + \nu_{ik} + \nu_{i\ell} + \\
& \sum_{d=1}^{D}\sum_{m=1}^{M_d}\sum_{s\in\mathcal{S}_d}\sum_{t\in\mathcal{S}_d}\sum_{u,v:u\oplus v=s} \eta_{dkmu}\eta_{d\ell mv}\mu_{dst}x_{dinmt} \\
\log q(Y_n=1, Z_n^2=(k,k)) &\propto \mathbb{E}[\log\delta] + 2\nu_{ik} + \\
& \sum_{d=1}^{D}\sum_{m=1}^{M_d}\sum_{s\in\mathcal{S}_d}\sum_{t\in\mathcal{S}_d}\sum_{u:u\oplus u=s} \eta_{dkmu}\mu_{dst}x_{dinmt}
\end{aligned}
$$

$$
\begin{aligned}
\bar{\alpha} &= \alpha + \sum_{i=1}^{I}\sum_{n=1}^{N_i}\sum_{k=1}^{K}\sum_{\ell=1}^{K} \zeta_{ink\ell} \\
\bar{\beta} &= \beta + \sum_{i=1}^{I}\sum_{n=1}^{N_i}\sum_{k=1}^{K} \xi_{ink} \\
q(\delta) &\sim \mathrm{Beta}(\bar{\alpha},\bar{\beta})
\end{aligned}
$$

$$
\begin{aligned}
\bar{\kappa}_{ik} &= \kappa_k + \sum_{n=1}^{N_i} \xi_{ink} + 2\sum_{n=1}^{N_i}\sum_{\ell \neq k} \zeta_{ink\ell} + \sum_{n=1}^{N_i} \zeta_{inkk} \\
q(\boldsymbol{\pi}_i) &\sim \text{Dirichlet}(\bar{\boldsymbol{\kappa}}_{\boldsymbol{i}})
\end{aligned}
$$

When understanding and interpreting the update equations it is useful to note that most decouple into four parts:

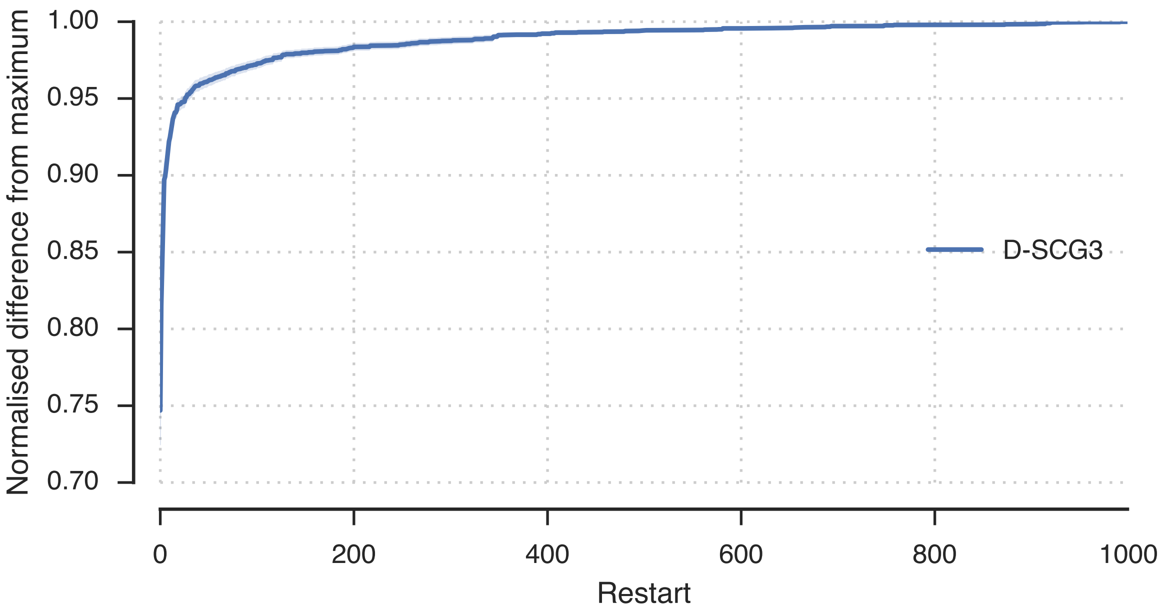
1. A prior term
2. A term for single observations
3. A term for doublets from different clonal populations
4. A term for doublets from same clonal population

To assess convergence we compute the ELBO after updating all model variables. If the value does not change by a pre-determined threshold, we determine that optimization has converged and return the current variational parameters. The inference procedure outlined above is only guaranteed to converge to a local optima. To attempt to find the global optima we perform multiple random restarts, 100 for the synthetic datasets without doublets and 1000 for the real datasets and synthetic datasets with doublets. This significantly increases the computational cost of the method, but the restarts can be performed in parallel to reduce actual run time.

Determining a sufficient number of restarts is an open problem. We suggest a simple heuristic to provide some guidance. For a pilot dataset representative of the users typical datasets, we suggest performing an extremely large number of runs. Next, compute the minimum ELBO across all runs (m) and the maximum ELBO across all runs (M). Then plot $(\hat{x}_i - m)/(M - m)$ where $\hat{x}_i = \max\{x_j : j \leq i\}$ and x_j is the j^{th} run performed assuming an arbitrary ordering of random restarts (not sorted by lower bound). The distance between one and the curve represents how far away the restart to that point is from the optimum ELBO. The goal is to identify a reasonable value such that the curve is close to one. In **Supplementary Note Fig. 3** we show this analysis on the doublet synthetic datasets. The results suggest that around 500-800 restarts would be a reasonable number to use. A dramatic gain achieved in the first 100 restarts, with little change observed after approximately 800 restarts.

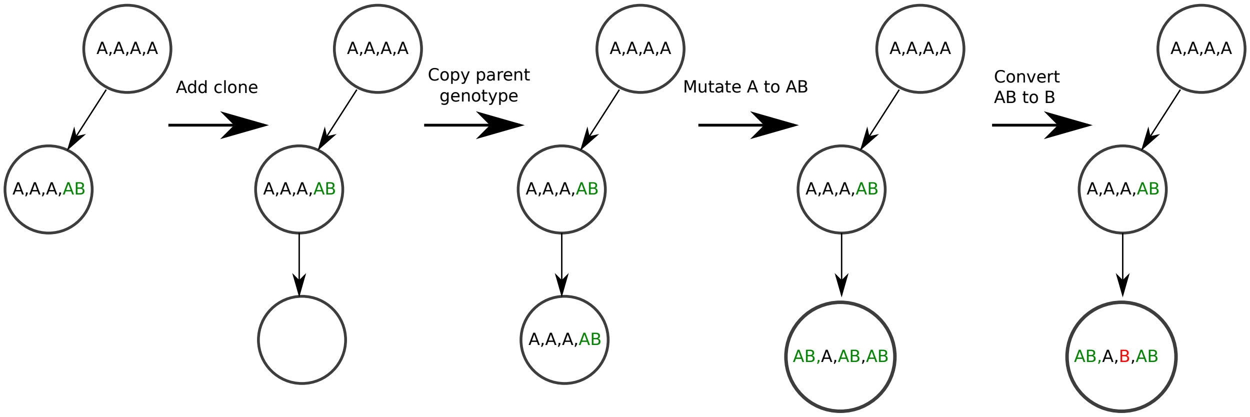
3.1.8 Model selection

A key benefit of using variational Bayesian inference is that model selection is relatively simple[4]. Since we attempt to approximate the posterior density, we obtain many of the benefits of full Bayesian inference. In particular there is an intrinsic penalty for fitting overly complex models. In the context of mixture models, this means that the model will tend to use fewer than the K possible clusters. In practice we initialize K to a much larger value than we expect, 40



Supplementary Note Figure 3: Convergence plot for doublet simulation data using the D-SCG3 method.

for all experiments reported. We then rely on the intrinsic model selection to pick the true number of clusters $K^* \ll K$. To be precise, we set K^* to the number of clusters which have at least one data point assigned to them. Data points can be assigned to clusters by choosing the most probable cluster, which for singlets is $\max_k q(Y_n = 0, Z_{in} = k)$.



Supplementary Note Figure 4: Example of one step of the clone phylogeny simulation procedure.

Nature Methods: doi:10.1038/nmeth.3867

Symbol	Definition
M	Number of loci to simulate.
N	Number of data points to simulate.
m	Number of loci converted from A to AB between parent and child clone.
λ	Average proportion of loci converted from A to AB between parent and child clones.
ω	Probability an locus in the AB state is converted to the B state.
$\boldsymbol{\pi}$	Vector of proportion of clones in cell population.
μ	Probability an locus is set to be missing.

Supplementary Note Table 5: Definition of parameters used for simulation.

4 Supplementary Note 2

4.1 Simulating synthetic datasets

In order to generate realistic ground truth dataset for benchmarking purposes we employed a hybrid simulation strategy. In the first step we simulate a clone phylogeny and clonal genotypes *in-silico*. In the second step we simulate allelic count data by sampling from the empirical distribution of variant allele frequencies (VAFs) for a set of known diploid heterozygous positions. Finally, we discretize the data and set a proportion of values as missing. We give a detailed description of the simulation procedure below with the relevant parameters defined in **Supplementary Note Table 5**. For brevity we use the term clone interchangeably with clonal population in the following description.

We provide a schematic of the following procedure in **Supplementary Note Fig. 4**. We generate the clone phylogeny by first specifying the number of loci, M, we wish to simulate. We initialize the root node in the phylogeny with the genotype that has the A state for all M loci. We then generate new clones until all M loci have been converted to the AB or B state in at least one clone. When we add a new clone, we choose a parent clone from the set of existing clones uniformly. We then copy the genotype of the parent clone to the new clone. We next mutate a Poisson distributed number of sites, m, to the AB state, where the parameter for the Poisson is λM. As a result each clone differs on average from its parent by a proportion λ of the loci. We enforce the infinite site assumptions so that only sites in the A state which have not been mutated elsewhere on the tree can be mutated. In the corner case where the number of loci to be mutated exceeds the number of remaining loci which have not been set to the AB or B state somewhere on the phylogeny, we set m to mutate the remaining loci. We next set each locus in the AB genotype state to the state B with probability ω to simulate loss of heterozygosity events. Note, we do not allow back mutations from the AB or B to A genotype at any stage, so that this procedure is guaranteed to end provided $\lambda > 0$. At the end of the procedure we return the phylogeny, defined as the graph of parent child relationships. We also return the set of clonal genotypes.

To accurately simulate allelic dropout we generated a real dataset by sequencing a panel of SNVs predicted to be heterozygous in the 184-hTert cell line. This cell line is largely diploid with amplification of chromosome 20 known to occur in later passages. We excluded positions on chromosome 20 so that all loci are expected to be in diploid regions. We sequenced a panel of 48 SNVs in 88 cells using a single-plex PCR amplification of each loci. We included bulk genomic DNA (gDNA) as a positive control for the PCR primers. We only included 32 loci with a variant allelic frequency (VAF) between 0.4 and 0.6 in the gDNA and that had coverage in more than 50% of cells. We took the distribution of VAFs at the target sites as the empirical distribution for AB state. For the A state, we took the empirical distribution to be the average background VAF computed over 30 bases in either direction of the target heterozygous position, excluding the target position. We set the empirical distribution for the B state to be one minus the values in the A state. We thus had three matrices of 88x32 allele frequencies, one for each state. We associated each of the M synthetic loci with one of the 32 real loci for the next step.

We randomly sampled the distribution, $\boldsymbol{\pi}$, of clonal prevalence from a symmetric Dirichlet distribution with parameter 1. We then proceeded to sample count data for N data points. We randomly decided if a data point was a doublet with probability δ. If the data point was not a doublet, we sampled a clone of origin from a Categorical distribution with parameter $\boldsymbol{\pi}$. If the data point was a doublet, we sampled two clones in the same way, and then combined their genotypes using the binary operator discussed in section 3.1.4. For doublets, we used the combined genotype for the remainder of the procedure. We generated count data based on the genotype of the associated clone. Specifically, we looked up the state of the associated clone genotype for the locus. We then used the relevant column from the matrix of real VAFs for the given state. For each locus we simulated a depth of coverage, d, from a Poisson distribution with mean parameter 1000. We then simulated the number of variant reads from a Binomial with the parameter given from the empirical distribution, and depth d. Finally, we applied the Binomial exact test to the data using the average variant allelic frequency of the state A empirical distribution as the null rate for each allele. The final step was to randomly assign loci as missing with probability μ. If an locus was set as missing, the depth of coverage and variant allele count was set to 0. In addition the p-values for the presence of both the A and B allele was set to 1 for missing loci. To discretize the data, we used a p-value threshold of 10^{-6} to determine if each allele was present.

5 Supplementary Note 3

5.1 Alternative methods

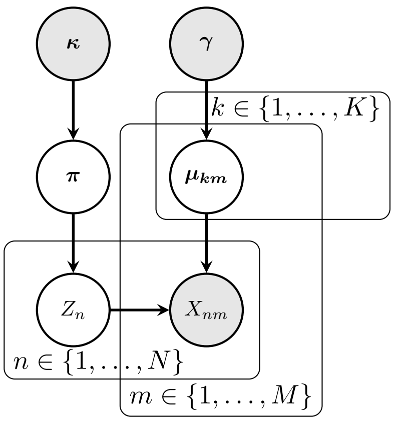
Hierarchical clustering We apply hierarchical clustering to the rows of the $N \times M$ matrix of variant allele frequencies. To construct the distance matrix we use the Euclidean (L^2) metric applied to the variant allele frequencies (HC-VAF) or the Manhattan (L^1) metric applied to the discrete data input matrix (HC-Discrete). We use the average linkage method during the agglomerative clustering stage. In order to generate a flat clustering we use *dynamicTreeCut*[14] method to automatically cut the dendrogram generated by hierarchical clustering. Hierarchical clustering provides no estimate of the clonal genotype associated with each cluster. Thus we exclude it from comparisons of genotype prediction accuracy.

We used the `dist` and `hclust` methods provided in the R (3.1.3) software package to compute distance matrices and perform agglomerative clustering. We used the `dynamicTreeCut` (1.6.2) package downloaded from CRAN repository to perform flat clustering.

Categorical mixture model The Bernoulli mixture model (BMM) is a well known approach for clustering binary valued data[3]. It has previously been applied in the single cell field to cluster cells and infer clonal genotypes[8]. The previous work was restricted to using a binary representation of SNVs, corresponding to the presence or absence of the B allele. It is straightforward to extend the BMM to a Categorical mixture model (CMM) to handle more general observation data with multiple discrete states. This is done by replacing the Bernoulli observation distribution with a Categorical distribution and the Beta prior distribution on cluster means with a Dirichlet distribution. A full description of this model is given in **Supplementary Note Fig. 5**. We perform inference for this model using the same mean field variational method as for the SCG model, with the same model selection strategy as for the SCG model. This model is implemented in the SCG package along with the code for VB inference. We use the same number of restarts for this model as the SCG model in all comparisons. We used an uninformative prior for $\boldsymbol{\mu}_{km}$, that is we set the value of $\boldsymbol{\gamma}$ to one for each component.

BitPhylogeny BitPhylogeny attempts to cluster cells, infer clonal genotypes and infer the clone phylogeny jointly[17]. The model uses a tree structured stick breaking (TSSB) prior over partitions of the data points[10]. This prior allows for the number of clusters to be inferred automatically while imposing a hierarchical structure among cluster parameters. For each event m and cluster k the authors the define the parameter $\theta_{km} \in (-\infty,\infty)$ which evolves according to a continuous time Markov process along the tree. The observed data is modelled as Bernoulli variable with parameter $\sigma(\theta_{km}) = \frac{1}{1+\exp(-\theta_{km})}$.

We use the software provided by the authors with minor modifications to reduce memory and disk space usage `https://bitbucket.org/aroth85/bitphylogeny`. We used version 0.3.2 of the software from this repository. All runs where performed by running the MCMC chain for 50,000 iterations collecting samples every 5th iteration after a burnin of 30,000 samples. These were the settings used by the software authors in[17].



$$\begin{aligned}
\boldsymbol{\pi}|\boldsymbol{\kappa} &\sim \text{Dirichlet}(\boldsymbol{\pi}|\boldsymbol{\kappa}) \\
Z_n|\boldsymbol{\pi} &\sim \text{Categorical}(Z_n|\boldsymbol{\pi}) \\
\boldsymbol{\mu}_{km}|\boldsymbol{\gamma} &\sim \text{Dirichlet}(\boldsymbol{\mu}_{km}|\boldsymbol{\gamma}) \\
X_{nm}|\boldsymbol{\mu}, Z_n = k &\sim \text{Categorical}(X_{nm}|\boldsymbol{\mu}_{km})
\end{aligned}$$

Supplementary Note Figure 5: Probabilistic graphical model representing the Categorical mixture model. Shaded nodes represent observed values or fixed values, while the values of un-shaded nodes are learned using VB.

6 Supplementary Tables

Supplementary Table 1:
Parameters used to generate synthetic data sets.
Columns:

- run_id - Identifier of synthetic experiment. Defines simulation parameters.
- doublet_prob - Probability a data point is a doublet in synthetic data set.
- loh_prob - Probability an AB event mutates to a B event per clonal generation.
- missing_prob - Probability an event is set to missing.
- mutation_prob - Probability an event mutates from A to AB per clonal generation.
- num_cells - Number of cells simulated.
- num_loci - Number of loci simulated.
- num_replicates - Number of replicates performed with given parameter settings.

Supplementary Table 2:
P-values from Nemenyi test comparing clustering accuracy using V-measure metric.
Columns:
• model_1 - First model used in Nemenyi post-hoc pairwise test.
• model_2 - Second model used in Nemenyi post-hoc pairwise test.
• nemenyi_p_value - P-value of post-hoc pairwise Nemenyi test.
• v_measure_mean_diff - Mean difference of V-measure scores. Value is model_1 - model_2.

Supplementary Table 3:
P-values from Nemenyi test comparing performance of genotype prediction using mean Hamming distance in two-state representation.
Columns:
• model_1 - First model used in Nemenyi post-hoc pairwise test.
• model_2 - Second model used in Nemenyi post-hoc pairwise test.
• nemenyi_p_value - P-value of post-hoc pairwise Nemenyi test.
• hamming_binary_snv_mean_diff - Difference of mean Hamming distance of predicted genotypes for cell to true genotype. Computed in two state representation. Value is model_1 - model_2.

Supplementary Table 4:
P-values from Nemenyi test comparing performance of genotype prediction using mean Hamming distance in three-state representation.
Columns:
• model_1 - First model used in Nemenyi post-hoc pairwise test.
• model_2 - Second model used in Nemenyi post-hoc pairwise test.
• nemenyi_p_value - P-value of post-hoc pairwise Nemenyi test.
• hamming_snv_mean_diff - Difference of mean Hamming distance of predicted genotypes for cell to true genotype. Computed in three state representation. Value is model_1 - model_2.

Supplementary Table 5:
Clustering performance of methods on synthetic data sets without doublets.
Columns:
• run_id - Identifier of synthetic experiment. Defines simulation parameters.
• run_replicate_id - Identifier of data set for a given synthetic experimental run.
• model - Method used to analyze data.
• completeness - Completeness score.
• homogeneity - Homogeneity score.
• v_measure - V-measure score. Harmonic mean of homogeneity and completeness scores.

Supplementary Table 6:
Genotyping performance of methods on synthetic data sets without doublets.
Columns:
• run_id - Identifier of synthetic experiment. Defines simulation parameters.
• run_replicate_id - Identifier of data set for a given synthetic experimental run.
• model - Method used to analyze data.
• hamming_binary_snv - Mean Hamming distance of predicted genotypes for cell to true genotype. Computed in two state representation.
• hamming_snv - Mean Hamming distance of predicted genotypes for cell to true genotype. Computed in three state representation.

Supplementary Table 7:
P-values from Nemenyi test comparing feature allocation accuracy using B-cubed metric.
Columns:

- model_1 - First model used in Nemenyi post-hoc pairwise test.
- model_2 - Second model used in Nemenyi post-hoc pairwise test.
- nemenyi_p_value - P-value of post-hoc pairwise Nemenyi test.
- f_measure_mean_diff - Mean difference of B-cubed F-measure. Value is model_1 - model_2.

Supplementary Table 8:
Feature allocation performance of methods on data sets with doublets.
Columns:

- run_id - Identifier of synthetic experiment. Defines simulation parameters.
- run_replicate_id - Identifier of data set for a given synthetic experimental run.
- model - Method used to analyze data.
- f_measure - F-measure of the B-cubed metric. Harmonic mean of B-cubed precision and recall scores.
- precision - B-cubed precision score.
- recall - B-cube recall score.

Supplementary Table 9:
P-values from Nemenyi test comparing maximum Hamming distance to nearest clone.
Columns:

- model_1 - First model used in Nemenyi post-hoc pairwise test.
- model_2 - Second model used in Nemenyi post-hoc pairwise test.
- nemenyi_p_value - P-value of post-hoc pairwise Nemenyi test.
- max_clone_hamming_mean_diff - Difference of maximum Hamming distance between predicted clonal genotypes and nearest true genotype. Value is model_1 - model_2.

Supplementary Table 10:
Accuracy of predicted clonal genotypes of methods on data sets with doublets.
Columns:

- run_id - Identifier of synthetic experiment. Defines simulation parameters.
- run_replicate_id - Identifier of data set for a given synthetic experimental run.
- model - Method used to analyze data.
- max_clone_hamming - Maximum Hamming distance between predicted clonal genotypes and nearest true genotype.

Supplementary Table 11:
Input data for CMM and SCG models for the childhood leukemia data set.
Columns:

- cell_id - Identifier of cell.
- Other columns - Observed genotype state of loci. 0 - A, 1 - AB, 2 - B, no value - missing

Supplementary Table 12:
Cluster assignments predicted by CMM3 model for the childhood leukemia data set.
Columns:

- cell_id - Identifier of cell.
- cluster - Identifier of predicted cluster.

Supplementary Table 13:
Cluster assignments predicted by SCG3 model for the childhood leukemia data set.
Columns:

- cell_id - Identifier of cell.
- cluster - Identifier of predicted cluster.

Supplementary Table 14:
Cluster assignments predicted by D-SCG3 model for the childhood leukemia data set.
Columns:

- cell_id - Identifier of cell.
- cluster - Identifier of predicted cluster.

Supplementary Table 15:
Predicted genotypes from CMM3 model of clusters with cells assigned for the childhood leukemia data set.
Columns:

- cluster - Identifier of predicted cluster.
- Other columns - Observed genotype state of loci. 0 - A, 1 - AB, 2 - B, no value - missing

Supplementary Table 16:
Predicted genotypes from SCG3 model of clusters with cells assigned for the childhood leukemia data set.
Columns:

- cluster - Identifier of predicted cluster.
- Other columns - Observed genotype state of loci. 0 - A, 1 - AB, 2 - B, no value - missing

Supplementary Table 17:
Predicted genotypes from D-SCG3 model of clusters with cells assigned for the childhood leukemia data set.
Columns:

- cluster - Identifier of predicted cluster.
- Other columns - Observed genotype state of loci. 0 - A, 1 - AB, 2 - B, no value - missing

Supplementary Table 18:
Input data for D-SCG3 model for the HGSOC data set.
Columns:

- cell_id - Identifier of cell.
- Other columns - Observed genotype state of loci. Loci names have a prefix indicating event type. States for SNVs 0 - A, 1 - AB, 2 - B, no value - missing. States for breakpoints 0 - Absent, 1 - Present.

Supplementary Table 19:
Cluster assignments for the HGSOC data set using D-SCG3 model.
Columns:

- cell_id - Identifier of cell.
- cluster - Identifier of predicted cluster.

Supplementary Table 20:
Predicted genotypes of clusters with cells assigned for the HGSOC data set using D-SCG3 model.
Columns:

- cluster - Identifier of predicted cluster.
- Other columns - Observed genotype state of loci. Loci names have a prefix indicating event type. States for SNVs 0 - A, 1 - AB, 2 - B, no value - missing. States for breakpoints 0 - Absent, 1 - Present.

Supplementary Table 21:
Predicted clone prevalences for the HGSOC data set using D-SCG3 model.
Columns:

- sample - Name of sample cells originated from.
- cluster - Identifier of predicted cluster.
- mean_posterior_prevalence - Mean value of approximate posterior for clonal prevalence.
- standard_deviation_of_posterior_prevalence - Standard deviation of approximate posterior for clonal prevalence.

References

1. Enrique Amigó, Julio Gonzalo, Javier Artiles, and Felisa Verdejo. A comparison of extrinsic clustering evaluation metrics based on formal constraints. *Information retrieval*, 12(4):461–486, 2009.

2. Ali Bashashati, Gavin Ha, Alicia Tone, Jiarui Ding, Leah M Prentice, Andrew Roth, Jamie Rosner, Karey Shumansky, Steve Kalloger, Janine Senz, et al. Distinct evolutionary trajectories of primary high-grade serous ovarian cancers revealed through spatial mutational profiling. *The Journal of pathology*, 231(1):21–34, 2013.

3. Christopher M Bishop et al. *Pattern recognition and machine learning*, volume 4. springer New York, 2006.

4. Adrian Corduneanu and Christopher M Bishop. Variational Bayesian model selection for mixture distributions. In *Artificial intelligence and Statistics*, volume 2001, pages 27–34. Morgan Kaufmann Waltham, MA, 2001.

5. Elza C de Bruin, Nicholas McGranahan, Richard Mitter, Max Salm, David C Wedge, Lucy Yates, Mariam Jamal-Hanjani, Seema Shafi, Nirupa Murugaesu, Andrew J Rowan, et al. Spatial and temporal diversity in genomic instability processes defines lung cancer evolution. *Science*, 346(6206):251–256, 2014.

6. Janez Demšar. Statistical comparisons of classifiers over multiple data sets. *The Journal of Machine Learning Research*, 7:1–30, 2006.

7. Peter Eirew, Adi Steif, Jaswinder Khattra, Gavin Ha, Damian Yap, Hossein Farahani, Karen Gelmon, Stephen Chia, Colin Mar, Adrian Wan, et al. Dynamics of genomic clones in breast cancer patient xenografts at single-cell resolution. *Nature*, 2014.

8. Charles Gawad, Winston Koh, and Stephen R Quake. Dissecting the clonal origins of childhood acute lymphoblastic leukemia by single-cell genomics. *Proceedings of the National Academy of Sciences*, 111(50):17947–17952, 2014.

9. Marco Gerlinger, Andrew J Rowan, Stuart Horswell, James Larkin, David Endesfelder, Eva Gronroos, Pierre Martinez, Nicholas Matthews, Aengus Stewart, Patrick Tarpey, et al. Intratumor heterogeneity and branched evolution revealed by multiregion sequencing. *New England Journal of Medicine*, 366(10):883–892, 2012.

10. Zoubin Ghahramani, Michael I Jordan, and Ryan P Adams. Tree-structured stick breaking for hierarchical data. In *Advances in Neural Information Processing Systems*, pages 19–27, 2010.

11. Matthew D Hoffman and David M Blei. Structured stochastic variational inference. *arXiv preprint arXiv:1404.4114*, 2014.

12. Matthew D Hoffman, David M Blei, Chong Wang, and John Paisley. Stochastic variational inference. *The Journal of Machine Learning Research*, 14(1):1303–1347, 2013.

13. Matthew KH Hong, Geoff Macintyre, David C Wedge, Peter Van Loo, Keval Patel, Sebastian Lunke, Ludmil B Alexandrov, Clare Sloggett, Marek Cmero, Francesco Marass, et al. Tracking the origins and drivers of subclonal metastatic expansion in prostate cancer. *Nature communications*, 6, 2015.

14. Peter Langfelder, Bin Zhang, and Steve Horvath. Defining clusters from a hierarchical cluster tree: the Dynamic Tree Cut package for R. *Bioinformatics*, 24(5):719–720, 2008.

15. Geoffrey McLachlan and David Peel. *Finite mixture models*. John Wiley & Sons, 2004.

16. Yong Wang, Jill Waters, Marco L Leung, Anna Unruh, Whijae Roh, Xiuqing Shi, Ken Chen, Paul Scheet, Selina Vattathil, Han Liang, et al. Clonal evolution in breast cancer revealed by single nucleus genome sequencing. *Nature*, 512(7513):155–160, 2014.

17. Ke Yuan, Thomas Sakoparnig, Florian Markowetz, and Niko Beerenwinkel. BitPhylogeny: a probabilistic framework for reconstructing intra-tumor phylogenies. *Genome biology*, 16(1):36, 2015.