Genotyping Copy Number Alterations from single-cell RNA sequencing

Salvatore Milite, Riccardo Bergamin, Giulio Caravagna.
Dept. Mathematics and Geosciences, University of Trieste, Italy

Corresponding: (GC) gcaravagna@units.it.

Abstract. Cancers are constituted by heterogeneous populations of cells that show complex genotypes and phenotypes which we can read out by sequencing. Many attempts at deciphering the clonal process that drives these populations are focusing on single-cell technologies to resolve genetic and phenotypic intra-tumour heterogeneity. While the ideal technologies for these investigations are multi-omics assays, unfortunately these types of data are still too expensive and have limited scalability. We can resort to single-molecule assays, which are cheaper and scalable, and statistically emulate a joint assay, only if we can integrate measurements collected from independent cells of the same sample. In this work we follow this intuition and construct a new Bayesian method to genotype copy number alterations on single-cell RNA sequencing data, therefore integrating DNA and RNA measurements. Our method is unsupervised, and leverages on a segmentation of the input DNA to determine the sample subclonal composition at the copy number level, together with clone-specific phenotypes defined from RNA counts. By design our probabilistic method works without a reference RNA expression profile, and therefore can be applied in cases where this information may not be accessible. We implement and test our model on both simulated and real data, showing its ability to determine copy number associated clones and their RNA phenotypes in tumour data from 10x and Smart-Seq assays, as well as in data from the Human Cell Atlas project.

Introduction

Cancers grow from a single cell, in an evolutionary process modulated by selective forces that act upon complex combinations of cancer genotypes and phenotypes (1,2). The fuel to cancer evolution is cellular heterogeneity, both at the genotypic and phenotypic level, and much is yet to be understood regarding its effect on tumour evolution and response to therapy (3,4). Notably, the heterogeneity observed in cancer can also be produced during normal tissue development, and therefore the quest for understanding heterogeneity is a problem with implications far beyond cancer (5–7).

While the evolutionary principle of cancer growth is intuitive to conceptualise and can be replicated in-vivo (8), it is generally hard to measure cancer clonal evolution using Next Generation Sequencing technologies (3,9). Even popular single-cell sequencing assays such as 10x and Smart-Seq (10), which achieve far higher resolution than bulk counterparts, pose several challenges for these analyses (11). On top of this, the generation of genotype and phenotype measures poses challenges also in the wet-lab, with much hope put into single-cell multi-omics technologies (12) that probe multiple molecules from the same cell in parallel (eg., the DNA and RNA, Figure 1a). In principle, with multi-omics data we can explicitly model

heterogeneous genotype-level and phenotype-level cancer populations by integrating multiple measurements for each cell in a multi-dimensional matrix (i.e., a tensor). In practice, however, single-cell multi-omics assays are still too expensive, and have limited scaling capacity when we want to sequence more than some hundreds of cells. An interesting opportunity comes instead from single-molecule assays, which are getting everyday cheaper and easily scale to generate data from thousands of cells (Figure 1b). A key point is that, at least conceptually, we can attempt the statistical integration of independent assays, trying to map one dataset on top of another (Figure 1c). For this to be possible, one needs to leverage on a quantitative model for the relation between the sequenced molecules (e.g., one can attempt to predict RNA abundance from DNA copies) (13).

In this work we attempt this type of integration working with total Copy Number Alteration (CNA) profiles, and independent single-cell RNA sequencing (scRNAseq) data. The setup of this work is most similar to that of clonealign, a recently published method that assigns scRNAseq profiles to tumour clones predetermined from independent low-pass single-cell whole-genome DNA sequencing (i.e., it is supervised) (13). Our method does not require input clones, but rather a segmentation (i.e., breakpoints) of the tumour DNA obtained from independent assays, together with estimates of the total ploidy per segment. We note that this information can be generated from routine low-pass bulk DNA sequencing, which is much easier and cheaper to obtain compared to the single-cell analogous.

In this framework we formulate an unsupervised clustering problem, in which we estimate clusters of cells whose RNA profile can be explained by similar CNAs (which we infer). The calling of CNA profiles together with cell clustering in the same statistical framework is, to the best of our knowledge, a feature only available in our tool. By design, our method is also reference-free as it does not require a putative reference RNA profile of normal cells. This is different from alternative other tools (e.g., Casper and HoneyBADGER) (14,15), and can help in cases where the normal cells are difficult to obtain (e.g., in organoids models). Like clonealign, our method statistically relates the ploidy of the tumour genome (i.e., the total copies of the major and minor alleles in each segment) to cellular sub-populations associated with distinct aneuploidy profiles. After deconvolution, we can directly associate cancer clones to their transcriptomic profile, providing an explicit mapping between genotype and phenotype at the single-clone level. This is particularly important both in cancer, normal development and other diseases. In cancer, where we want to characterise how subclonal CNAs and chromosomal instability drive tumour evolution and response to therapy, a key step that points to the puzzling link between continuous chromosomal instability and pervasive somatic CNA heterogeneity (16). In pre-cancer diseases, when we want to measure how pervasive is genetic heterogeneity in cells that can be causally linked to the onset of a cancer (7).

Results

We first conceptualise our approach, focusing on DNA and RNA which we use here. If we can assume a plausible model g of the relation between some DNA features and RNA, we can attempt the statistical integration of measurements collected from assays that independently measure these molecules (13). The aim (Figure 1c) is to emulate joint assays by mapping measurements of the DNA (input y) on top of RNA measurements (e.g., input x).

Genotyping exploits intermediate representations of the inputs. From DNA, we can first compute the value $f(y)$, which in our case may represent the tumour total copy number segments (i.e., the ploidy per a certain window of the reference genome). Then, we can use $f(y)$ to predict RNA abundance at the level of segments in each single-cell; i.e., we "genotype" segments on top of RNA counts by computing $x \sim (g \circ f)(y) = g(f(y))$. For this task, following earlier works we assume that g is a linear model of the DNA ploidy; in practice we correlate the amount r of RNA transcripts that map to a segment with the number of gene copies c, using a linear relation $r \sim \alpha c$ (13). This conceptual framework is general, and can be used for other computations. For instance, when we seek to derive allele-specific expression profiles $f(y)$ are tumour mutations called from y, and we can genotype them from the RNA transcripts (17) when g uses a Binomial variable that counts alternative and reference alleles.

The CONGAS method

We have developed CONGAS, a Bayesian method to genotype CNA calls from single-cell RNAseq data, and cluster cells into subpopulations with the same CNA profile (Figure 2). The main method is based on a mixture of Poisson distributions and uses, as input, absolute counts of transcripts from single-cell RNAseq. The model requires to know, in advance, also a segmentation (Figure 2a) of the tumour genome and the ploidy of each segment (Figure 2b,c); this is $f(y)$ in Figure 1c. Assuming to have input CNA data poses a minimal burden. In cancer, the input (y) for segmentation can be generated via low-pass bulk DNA sequencing, and analysed with standard CNA calling pipelines. If this was not available and cannot be assumed straightforwardly, one can attempt to segment directly RNA counts, as we perform in one case study, and use that for CONGAS. In normal or conditions where there is a strong bias towards stable diploid populations and we might chase mostly macroscopic events, one could use pre-defined arm-level or chromosome-level segmentations (5).

With the segmentation available beforehand (assumption) we simplify the genotyping problem from a statistical point of view, because in practice we avoid segmenting noisy RNA profiles at the single-gene resolution. The relation (Figure 2d,e) between the amount of tumour DNA (i.e., the segment ploidy), and the total amount of expected RNA transcripts from genes that map onto the segment, uses a linear model for g as in (13). Input CNA segments are also used to create ad-hoc Bayesian priors based on segments ploidy; if these are not available one can assume a predefined ploidy for the analysed genome (e.g. 2 for both tumour, normal, etc.).

The CONGAS model exists in both parametric and non-parametric form as a mixture of $k \geq 1$ subclones (i.e., clusters) with different CNA profiles (Supplementary Figure S1). The model is then either a finite Dirichlet mixture with k clusters, or a Dirichlet Process with a stick-breaking construction (non-parametric). In both cases the model handles library size normalisation, and can accept optional covariates to adjust for known confounders (e.g., batch effects), as in (13). As far as we are aware, this is the first method that attempts to infer CNAs and jointly cluster input cells in a unique statistical model.

The model likelihood with the usual independence assumption among the input cells and among the input CNA segments reads as

$$p(\mathbf{Y}|\theta, \boldsymbol{\mu}, \mathbf{C}, \mathbf{Z}, \boldsymbol{\pi}) = \prod_{n=1}^{N} \prod_{i=1}^{I} p(y_{ni}|\theta, \boldsymbol{\mu}, \mathbf{C}, \mathbf{Z}, \boldsymbol{\pi})$$

Here $\mathbf{Y}$ is the $N \times I$ input data matrix of RNA counts, which describe N sequenced cells and I input segments (mapped anywhere on the genome). Counts on a segment $y_{n,i}$ are summed up by pooling all genes that map to the segment; with cumulative counts we rarely observe 0-counts segments, which allows us to avoid zero-inflated distributions often used to separate measurements from expression (18). The model uses θ_n, a Gamma-distributed latent variable which models the library size for cell n, and μ_i for the number of genes in segment i (a constant determined from data). In CONGAS $\mathbf{C}$ is the clone CNA profile for k clones, where each clone is defined by I segments and associated CNAs; the prior for $\mathbf{C}$ is a log-transform of a normal distribution, consistently with the fact that ploidies are positive values. The model allows for B covariates specified by a $B \times I$ matrix, and implements a linear model with Gaussian coefficients, equivalent to the one adopted in clonalign (13). In this formulation $\mathbf{Z}$ are the $N \times k$ latent variables that assign cells to clusters, and $\boldsymbol{\pi}$ the k-dimensional mixing proportions. Based on this modelling idea we also built alternative models that can process input data when these are already corrected for library size (e.g., in units of transcripts per millions, or read fragments per million), using in this case Gaussian likelihoods. We describe all the models in detail in the Online Methods.

CONGAS is implemented in 2 open-source packages. The CONGAS package implements all the fitting procedures in the Python probabilistic programming language Pyro (19), exploiting its backend to fit the model by stochastic variational inference, running on both CPU and GPU. An extra R package RCONGAS wraps functions for data pre- and post-processing, visualisation and model investigation around CONGAS.

Synthetic simulations

We tested CONGAS by simulating synthetic datasets directly from its generative model (all tests are available as Supplementary Data). Each simulated dataset reflected standard setups for a

10x sequencing assay with about 1000 simulated cells, in line with today's standard assays. The model scales very fast also to analyse tens of thousands of cells, thanks to GPU support native in Pyro. Overall results from our simulations showed that CONGAS can retrieve the generative model for a number of configurations of the input data (Figure 3).

The method was capable of identifying subclones from tumours that have ≤ 5 subpopulations, evolving by both linear and branched patterns of evolution (Figure 3a). In the simulated data, every cancer subpopulation was assigned a different copy number profile, reflecting somatic CNAs accruing in subclonal lineages. The performance was measured from the ratio of agreements over disagreements in cell clustering assignments (Adjusted Rand Index, ARI), and was found consistent with other information-theoretic scores usually adopted for measuring clustering performance (Supplementary Figure S2). Also, clustering assignments were stable across a number of configurations of different complexity of the simulated tumour (Figure 3b).

CONGAS could also work when we introduced overdispersion in sequencing data, a violation of its Poisson-based model that, natevely, does not support dispersion. This was obtained by sampling read counts data from a Negative Binomial model with increasing dispersion. The results showed a trend relating rand index to dispersion, with performance increasing for lower dispersion and plateauing for non-dispersed data, as expected (Figure 3c and Supplementary Figure S3).

We carried out a final important test to assess the role of the input segmentation in determining subclones. Precisely, we generated subclonal CNAs that were shorter than the input CONGAS segments, so that only a percentage of genes mapping to a segment were showing a signal in RNA data. This also provides another test-case where the assumptions of our method are violated. Performance was measured from very small subclonal CNAs (10% of genes per segment), to larger ones (up to 90% of the genes). Results showed a trend between rand index and percentage of involved genes, with good performance achieved when $\geq 40\%$ of the genes that map to a segment are associated to the subclonal CNA (Figure 3d and Supplementary Figure S4). This clearly suggests that genotyping focal CNAs that involve a handful of genes can be difficult, while wider CNAs are generally identifiable even with imperfect segmentation.

Subclonal decomposition of a triple-negative breast xenograft

We used CONGAS to analyse a triple-negative breast cancer dataset generated with 10x technology, first released as a case study for the clonalign method (13). This is an important case study because we can biologically validate our method thanks to the grand truth provided by the single-cell low-pass DNA data required by clonalign.

This dataset refers to the patient-derived xenograft SA501X2B collected from patient SA501, and has been used to determine clone-specific phenotypic properties that associate with a complex clonal architecture; notably the inferred clonal populations have been validated by

reproducing clonal dynamics over successive xenograft passages (20). From low-pass WGS analysis this sample is known to harbour three genotypically distinct clones with prevalence 82.3%, 10.8%, and 6.9% respectively (with one clone sweeping in subsequent engraftments).

To run CONGAS we used as input the DNA segmentation consistent with the largest clone identified in (13) to mimic the main bulk signal, retaining all segments with at least 10 genes. Our analysis from 504 single cells could identify two of the three expected clones (Figure 4a). The identified populations show significant differences in the counts of RNA transcripts (Supplementary Figure S5); the largest clonal population consists of $n = 380$ cells (~75% of total), and the smallest one of $n = 124$ cells (~25% of total) (Figure 4b,c). In terms of clone-specific Differential Expression (DE) analysis, we could find $n = 122$ genes that are either significantly upregulated or downregulated; Wald test over negative binomial coefficients (fitted using DESeq2) with $padj < 0.001$ (Benjamini-Hochberg method) and absolute log-fold change exceeding 0.25 to determine the genes' regulatory state (Figure 4d, Supplementary Figure S6). Note that some of these genes fall outside of the CNAs that characterize these populations, and therefore could only be marginally explained by such genetic changes. Instead, these could be explained by more complex regulatory mechanisms indirectly linked to these and other events. In this analysis library factors were also found quite variable across cells (Supplementary Figure S7).

The signals identified by CONGAS are clearly observable across multiple chromosomes, with particular strength on chromosomes 15, 16 and 18 (two-sided Poisson test, p-value $p < 0.001$, Figure 4e,f,h and compare with Figure 2d). This result is consistent with low-pass analysis originally carried out in (13), which we here use to validate our inference (Supplementary Figure S8). Concerning the differences with respect to the original analysis (Figure 4h) - i.e., the lack of identification of a third clone - is simple to explain: the DNA segment that defines this particular population contains less than 10 genes, and is therefore too small to be analysed by CONGAS. We note however that this missing cluster is poorly supported also in the original analysis, which exhibits assignment uncertainty between the second largest clones and this population (13).

Tumour and normal deconvolution in primary glioblastoma

We used CONGAS to analyse the glioblastoma Smart-Seq data released in (21). This dataset consist of $n = 430$ cells from five primary glioblastoma. In particular, we analysed one patient (MGH31) with 75 associated cells. MGH31 was the patient of choice as both in the original paper and in a successive analysis it seemed to harbor distinct subclonal populations (15). The analysis of these data is mainly challenged by i) the lack of an input CNA segmentation for CONGAS, and ii) the presence of normal healthy cells in the sequenced sample. For these reasons, this puts an extra burden on our method.

To implement this analysis we have created a simple preprocessing pipeline around CONGAS. To retrieve an input segmentation for our analysis, we have developed a simple Hidden Markov

Model to segment minor allele frequencies from the input cells, which we could successfully run on this Smart-Seq assay (Online Methods). In this way we have identified clear events of loss of heterozygosity, as well as big genome amplification involving chromosomes 7, 10, 13 and 14 (Supplementary Figure S9). We have selected these segments to run our analysis.

In a first run (Figure 5a,b), with all cells together (normal plus tumour), CONGAS identifies $k = 3$ clusters; one of them (cluster 3, $n = 10$ cells) does not show neither the LOH nor the big copy number amplification events. Interestingly, this cluster are indeed normal cells that contaminate the sample, as suggested by their comparison with a healthy reference in(15). We removed normal cells and re-run CONGAS on the remaining tumour cells, further finding $k = 3$ distinct cancer subclones (Figure 5c,d). The phylogenetic reconstruction of these clones suggest an early branching from an ancestor harbouring the amplification on chromosome 7, and the deletion of 10. Clones then branch out: one clone is sustained by a clear amplification on chromosome 5 (34% of cells) and a linear path describes the evolution of nested clones with increasing levels of aneuploidy (with the largest subclone with also 34% of cells).

The DE analysis of these few cells is inconclusive (data not shown) due to the small number of sequenced cells; nonetheless this 2-steps analysis shows how CONGAS can perform signal deconvolution in the presence of normal contamination of the input sample. This is interesting and consistent with the fact that the method can work without a reference normal expression.

Monosomy of chromosome 7 in hematopoietic precursor cells

In order to show the versatility of CONGAS we have analysed also mixtures of non-cancer cells collected within one experiment associated to the Human Cell Atlas project (5). In this case the dataset provides scRNA from hematopoietic stem and progenitor cells from the bone marrow of healthy donors and patients with bone marrow failure. We focused on one patient (patient 1) with severe aplastic anemia that eventually transformed in myelodysplastic syndrome, and for which cytogenetic analyses revealed monosomy of chromosome 7, a condition that increases the risk of developing leukaemias (5).

To analyse this data we pooled patient 1 together with one of the healthy donors (n=101 cells total, Figure 6a). This gives CONGAS both diploid cells (control, from the health patient), and cells with chromosome 7 deletion. There is not segmentation for these data, so we used full chromosomes with a diploid prior. Aneuploid cells were clearly distinguished from diploid cells by this analysis, which found $k = 2$ clusters of cells. One cluster contains diploid cells from both patients, the other cells from patient 1, which are associated with chromosome 7 monosomy. Clone-specific differential gene expression (Figure 6b) performed in the same way as the breast xenograft example reported 99 genes differentially expressed with $padj < 0.01$ and log-fold change greater than 0.25. Interestingly the top DE genes were not expressed in the aneuploid chromosome, suggesting that an integrated study of transcriptomics and copy number

alterations could eventually lead to a better understanding of how these genomic events - which have considerable dimension in megabases - can alter cellular behaviour across different pathways and functional modules throughout the whole genome.

Discussion

In this paper we present a Bayesian statistical method to genotype CNA from single-cell RNA sequencing data, which uses a powerful backend for probabilistic programming, and can run on both CPU and GPU to scale to large datasets with tens of thousands of cells (Supplementary Figure S10). CONGAS requires an input segmentation determined from other sequencing assays of the input cells (e.g, a low-pass bulk whole-genome), which nowadays is generated at a fraction of the cost of the input single-cell RNA dataset. Our method has advantages compared to approaches that require a normal reference such as RNA sequencing from a matched tissue (which often is not patient-specific) or normal cells in the samples (14,15). This means that our method finds groups of cells with different copy numbers, no matter what the reference expression is. This is a major advantage for certain experimental designs, for instance in the context of organoid models where we easily grow tumour cells, but struggle growing non-tumour cells (22), or in cell-lines based analysis.

CONGAS reconciles also tumour CNA heterogeneity from RNA by grouping input cells into putative genetic clones, using a probabilistic model for cell assignment. Compared to alternative approaches that do not attempt clone detection (e.g., CNA callers), or that separate calling from clustering, our method is completely integrated in a unique inference framework, and its results can be used to compute a clonal phylogeny with standard phylogenetic methods (23). From a CONGAS model we can also assess the phenotypic signature that characterises each clone, at least as defined at the RNA expression level. This mapping is also trivial since it comes out as a byproduct of the integration of genetic CNA events together with RNA count data.

Our analysis can be used to detect CNA-associated subclones, and measure their precise differential expression patterns, a key step to study how selective pressures shape genotypes and phenotypes evolution in distinct populations of cells. In this first work we also show - in multiple case studies - how to determine clone-specific differentially expressed genes which can only be partially explained by copy number segments, pointing to complex non-trivial regulatory mechanisms that link genotype states with expression patterns. Our method provides a solid statistical framework to approach this type of inference, which is crucial to investigate clonal dynamics in disease progression, as well as cell plasticity and patterns of drug response from the large wealth of single-cell data available nowadays.

Data Availability

The implementation of CONGAS is available into two separate packages: CONGAS (Python) and rCONGAS (R). Both packages are available at GitHub

- https://github.com/Militeee/CONGAS
- https://github.com/Militeee/rCONGAS

The analysis of real data, and scripts to generate the main figures of the paper are available in the GitHub repository https://github.com/caravagn/rCONGAS_test.

Note to the reviewers: During review these repositories are kept private. The Supplementary Data attached to this submission contains all the material currently available on the repositories.

Authors contribution

SM and GC conceptualised and derived CONGAS. SM implemented the tools and ran the synthetic tests; SM and RB gathered real data for the case studies. SM, RB and GC analysed all data, interpreted the results, and drafted the manuscript.

Competing interests

The authors declare no competing interests.

Online Methods

CONGAS

We discuss here our approach to genotype CNAs from single-cells RNA sequencing data. To be precise, while we generally refer to CONGAS as a single model, in reality the framework leverages a set of core ideas to create models with slightly different features, and that can analyse different types of data. The models are:

- (default) a Dirichlet finite mixture for RNA counts data;
- a Dirichlet finite mixture for log-normalised RNA data;
- a Dirichlet Process extension of both the above.

Also, the framework offers a simple Hidden Markov Model (HMM) to segment input single-cell data, which can be used to generate segmentations required by CONGAS if these are missing.

In next sections we discuss all models; plate notations are in Supplementary Figure S1.

Modelling the relationship between CNVs and gene expression. In order to recover distinct populations of cells that differ for the copy number of specific segments, we follow the idea of modelling the generative process of reads counting using a latent variable (13). Instead of modelling the expression of a single gene, we use the aggregated read counts over a whole genome segment. This is why our model assumes a pre-existing segmentation of the genome. This segmentation is a fundamental part of the model, as it guarantees us a convenient way of

treating the segments as independent statistical entities. As explained in the text, this extra assumption poses a minimal burden, considering the cost of generating the single-cell dataset.

The latent variable has a direct dependence on the copy number state, and it should explain the differential counts expression over a segment among clones. A simple but effective choice is to consider the segment expression to be linearly dependent on the total number of chromosomes; this is the same idea previously exploited in (13). The other factors that contribute to segment expression and that have to be taken into account are the different depth at which each cell is sequenced, and the number of genes present in a segment.

Formally, we index with $i = (1, ..., I)$ the segments, with $n = (1, .., N)$ the number of cells, and we introduce a categorical latent variable for each cell assignment $\mathbf{Z} = [z_{nk}]$ such that

$$p(\mathbf{Z}|\boldsymbol{\pi}) = \prod_{n,k} \pi_k^{z_{nk}}$$

where $\mathbf{z_n}$ is a vector that evaluates to 1 if cell n belongs to cluster k and 0 otherwise, $\boldsymbol{\pi}$ is drawn from a Dirichlet distribution as in standard finite-mixture modelling (9).

We also indicate as θ_n a Gamma distributed latent variable which models the library size and name μ_i the number of genes in a segment i; this number is a constant that depends on the input segmentation. Our CNAs are modeled as continuous LogNormal distributions; we define $\mathbf{C} = [c_{ik}]$ such that $c_{ki} \sim \text{LogNormal}(m_{ki}, v_{ki})$ so to represent the copy number value for segment i, in cluster k.

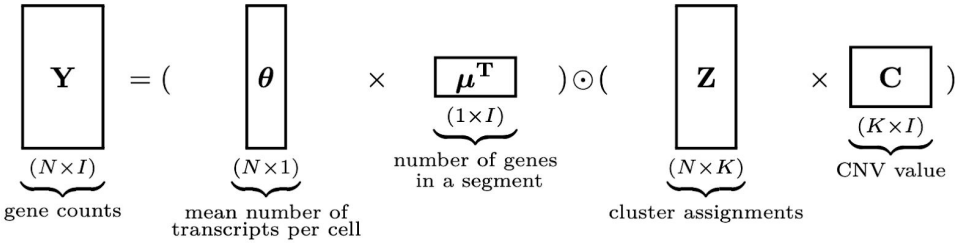
In the default read-counts based mode in CONGAS, we describe the probability of the counts $\mathbf{Y} = [y_{ni}]$ of the cell n in segment i as

$$p(y_{ni}|\theta, \boldsymbol{\mu}, \mathbf{C}, \mathbf{Z}) = \text{Pois}\left(\frac{\theta_n \cdot \mu_i \cdot \prod_{k=1}^{K} C_{ik}^{z_{nk}}}{\sum_{i=1}^{I}\prod_{k=1}^{K} C_{ik}^{z_{nk}}}\right)$$

and the full likelihood is obtained by assuming both cells and segments to be independent. This assumption is valid since we have an existing segmentation of the input genome. The model likelihood becomes

$$p(\mathbf{Y}|\theta, \boldsymbol{\mu}, \mathbf{C}, \mathbf{Z}, \boldsymbol{\pi}) = \prod_{n=1}^{N}\prod_{i=1}^{I} p(y_{ni}|\theta, \boldsymbol{\mu}, \mathbf{C}, \mathbf{Z}, \boldsymbol{\pi})$$

and a graphical representation of the model is in Supplementary Figure S1. Another way of thinking of the denominator in the formula is, given that all the effects are linear, as a matrix decomposition of the input. Note that here the denominator is omitted.



While this model does work with raw counts and accounts for both gene number and library size normalization, the CONGAS framework also supports input single-cell data that are already normalised. This helps because often the only measurements availables are in units of transcripts per million reads, or alternative normalised measures. This is also useful if one wants to perform the inference using a custom normalization method.

Therefore, in addition to the Poisson count-based model we have developed a model that works with Normal distributions. In this case we assume the input to be aggregate values (which are not anymore integers) of expression over all segments, which must be already normalized between cells and segments.

The segment expression y_{ni} value in this alternative model formulation is now expressed as a Normal distribution; this model has likelihood

$$p(\mathbf{Y}|\mathbf{\Lambda}, \mathbf{C}, \mathbf{z}) = \prod_{n=1}^{N} \prod_{i=1}^{I} \mathcal{N}(m_{ni} = \prod_{k} \mathbf{C}_{ik}^{z_{nk}}, \lambda_{ni} = \mathbf{\Lambda_i})$$

Notably here we do not have any depence over library size and on the number of genes in a segment.

Parameter estimation with stochastic variational inference. Given our models we want to learn suitable values for the parameters, that we indicate generally as Z. Using ψ to identify model hyperparameters, our goal, if we tackle the inference problem from a bayesian perspective, is to learn the posterior distribution of our parameters, namely:

$$p(\mathbf{Z} \mid \mathbf{X}, \psi) = \frac{p(\mathbf{X} \mid \mathbf{Z}, \psi) p(\mathbf{Z} \mid \psi)}{p(\mathbf{X} \mid \psi)}$$

Such distribution is generally analytically intractable and different methods for sampling or approximate inference have been developed. CONGAS is developed using the probabilistic programming language Pyro (19) and exploits stochastic variational inference (24) to get an approximation of the true posterior distribution.

Briefly, the idea behind variational inference is to find a variational distribution $q(\mathbf{Z})$, that can approximate the real posterior. In particular we do so by performing minimization using gradient descent over the negative of the evidence lower bound (ELBO)

$$\mathrm{ELBO}(q) = \mathbb{E}[\log p(\mathbf{X} \mid \mathbf{Z})] - \mathrm{KL}(q(\mathbf{Z}) \| p(\mathbf{Z}))$$

This is equivalent to minimize the Kullback-Leibler divergence between the variational and the actual posterior distribution. In this framework our latent variables are parameterized by a set of variational parameters ϕ and our goal is to learn those parameters by maximizing the ELBO. Taking the gradient from the previous equation and expliciting the dependence on the variational parameters ϕ:

$$\nabla_\phi \mathrm{ELBO} = \nabla_\phi \mathbb{E}_{q_\phi(\mathbf{Z})} \left[\log p(\mathbf{X}, \mathbf{Z}) - \log q_\phi(\mathbf{Z})\right]$$

We used the reparametrization trick to obtain a low variance Monte Carlo estimate of this gradient, and the tool can choose to calculate the gradient estimate over a minibatch of observations - this still provides an unbiased estimation of the gradient and can give a huge speedup over big datasets. We used Adam (25) as the optimizer of our choice throughout the whole paper, nevertheless the user can use any optimizer present in PyTorch (26) or define a custom one.

The tool also gives the possibility to perform MAP inference, using the same mechanism.

$$\hat{\mathbf{Z}}_{\mathrm{MAP}}(\mathbf{X}) = \underset{\mathbf{Z}}{\mathrm{argmax}}\ p(\mathbf{X} \mid \mathbf{Z}, \psi) p(\mathbf{Z} \mid \psi)$$

Note that the latent variable describing the library size dependence θ is always learned using MAP inference, while for the other variables the user can choose between a full Bayesian inference, or MAP.

Model and prior distributions. The bayesian setting gives us the opportunity to integrate some pre-existing information directly into the model. A way to guide solutions toward a meaningful direction is to assume the prior distribution of CNV values (the c_{ij} in our model; see also Figure 2b) to be centered around the ploidy values obtained from bulkDNA-seq analysis.

More in detail, the model joint distribution can be factorized as:

$$p(\mathbf{Y}, \mathbf{Z}, \mathbf{C}, \theta, \boldsymbol{\pi}) = p(\mathbf{Y} | \mathbf{Z}, \mathbf{C}, \theta, \boldsymbol{\pi}) p(\mathbf{Z}, \mathbf{C}, \theta, \boldsymbol{\pi})$$

$$= p(\mathbf{Y} | \mathbf{Z}, \mathbf{C}, \theta, \boldsymbol{\pi}) p(\mathbf{Z} | \boldsymbol{\pi}) p(\boldsymbol{\pi}) \prod_{ik} p(c_{ik}) \prod_{n} p(\theta_n)$$

In the variational framework our latent variables are approximated as variational distributions distributions $q(\mathbf{Z}, \mathbf{C}, \theta, \boldsymbol{\pi})$, supposed to be independent and factorizable. The prior distributions for our latent variables are:

- $q(c_{ik}) \sim \mathrm{LogNorm}(m_{ik}, v)$, where m_{ik} is the CNV value from bulkDNA-seq and the variance v is chosen by the user to govern how far we think the actual CNV values are from the ones inferred by bulkDNA-seq, we use a default of 0.5.
- $q(c_{theta_n}) \sim \mathrm{Gamma}(e_s, e_r)$, here θ distribution can be roughly estimated from the data; however, even large scarcely informative priors tend to work well in most of the cases. Default values are $e_s = 3$ and $e_r = 1$
- $q(\boldsymbol{\pi}) \sim \mathrm{Dirichlet}(\mathbf{r})$, the user can input is prior over the cluster distributions, by default all cluster are a priori assumed to have equal proportions (i.e. $r_k = 1/K$)

The whole model can also be visualized using plate notation (Supplementary Figure S1).

In the Gaussian mixture model, $q(c_{ik})$ prior is still distributed as a LogNormal, with the same parameter as before, while for the variance prior $\lambda_i \sim \mathrm{Uniform}(a, b)$.

For the first step of optimization $\mathbf{C}$ is initialized using k-means clustering, the library size factor is initialized as

$$\theta_{\mathbf{n}} = \frac{\mathbf{x_n}}{(\chi^T \mu)}$$

where χ is the CNV value obtained from the input bulk segmentation.

Model-selection. All these formulations assume the number of clusters K to be a constant. To pick the optimal number of clones we first select a set of candidate Ks and fit a separate model for each one, then we perform model selection. In the our packages the information criteria currently implement are:

- the Bayesian Information Criterion (BIC) $\mathrm{BIC} = k \ln(n) - 2 \ln(\widehat{L})$, where $\hat{L}$ is the likelihood of the model;
- the Akaike Information Criterion (AIC) $\mathrm{AIC} = 2k - 2 \ln(\hat{L})$;
- the Integrated Classification Likelihood (ICL), based on the BIC, which is

  $$\mathrm{ICL} = \mathrm{BIC} + \mathrm{H}(\mathbf{Z})$$

  where $\mathbf{Z}$ are latent variables and $\mathcal{H}$ their entropy

$$\mathcal{H}(X) = -\sum_{i=1}^{n} \mathrm{p}\left(x_i\right) \log \mathrm{p}\left(x_i\right) \ .$$

Where not explicitly indicated for our in silico and in vivo analysis the criterion of choice was always the BIC.

Total number of cells and segments has to be taken into account, as for some Smart-Seq runs with a small number of cells the BIC and ICL regularization might be too strong. In those cases filtering segments or using other selection criteria can help to obtain meaningful results.

Non-parametric extension. Given the importance of selecting the correct number of clusters in a finite mixture model, we also adopted a non-parametric formulation of CONGAS. This is expressed as a stick-breaking formulation of the Dirichlet Process (27).

The stick breaking model in our case can be seen as a semi-parametric way of choosing the optimal number of clusters along with their weights. Note that the Dirichlet Process is defined for an infinite number of clusters, what is commonly done in practice is to set a high number of clusters to approximate this behaviour.

The generative process in our setting has these steps (28):

- Draw $\beta_k \sim \mathrm{Beta}(1, \alpha)$
- Draw $c_{ik} \sim \mathrm{LogNormal}(m_{ki}, v_{ki})$
- The mixture weights π are obtained as

$$\pi_k(\beta_{1:T} = \beta_k \prod_{j<i}(1-\beta_j))$$

- For each $n \in 1, ..., N$ and $i \in 1, ..., I$, draw $z_n \sim \pi_k$ and $x_{ni}(c_{ik})$

Where we omitted all the variables that do not depend explicitly on the clusters K. The α is a hyperparameter that controls our prior beliefs on the number of clusters present in the data, i.e. higher α will penalize a solution with more clusters and vice versa. Learning good values for α is fundamental in a noisy setting like scRNA-seq, and different metrics have been described for such optimization problems (29). Nevertheless, throughout the paper we preferred to run different instances of the finite mixture model and then do model selection as explained above. This is due both to the difficulties in accurately optimizing α and in the increased understanding of the data one can get from having all the models fitted.

Synthetic data generation

Synthetic datasets were generated following the CONGAS model.

First, the human chromosomes from the reference hg38 were divided into segments. For the segmentation we first set an approximate minimum distance for the segments, in our case 100 Megabases and then divide the total length of each chromosome. This process gives us the maximum number of possible segments for a chromosome, let us call it m_{chr}. After that we sampled the effective number of segments i_{chr} as an integer ranging from 1 to m_{chr}, with equal probability.

Breakpoints were then calculated by first dividing each chromosome by i_{chr} and then randomizing the endpoints of each segment adding an error value $e_i \sim \mathrm{Unif}(a, -a)$, where a is equal to 10% of the segment length. The total number of genes for each segment were sampled from a negative binomial distribution with size 6 and mean equal to segment length divided by a constant, in our case $1e6$.

Independently, a tree with a number of clones K and a distance among clones of D is constructed. The distance here is just the Hamming distance between segments, i.e. the number of different segments between two clones. The tree, given the number of clones, is set up by iteratively adding each new node to it and choosing its parent randomly from the tree leaves (with equal probability). To simplify both simulation and interpretation of the results each branch of the tree has the same length so that each child diverges from its parent for the same number of CN events.

Using this information we sampled counts by using the generative model of CONGAS. For the first test we simulated trees with a number of clones ranging from 1 to 9 and distance from 1 to 4, with 50 replicates for each combination. The clone proportions were randomly sampled from a Dirichlet with concentration parameters of $\alpha_k = 1/K$.

For the second test we changed the generative model to obtain overdispersion in the input counts, which were sampled from a negative binomial with parametrization

$$\Pr(X = k) = \binom{y + \zeta - 1}{y} \left(\frac{m}{m + \zeta}\right)^y \left(\frac{\zeta}{m + \zeta}\right)$$

Where the mean is equal to m, ζ is the size and the variance is

$$\mathrm{Var}[X] = m + \frac{m^2}{\zeta}$$

Our likelihood then becomes

$$p\left(y_i^n \mid \theta, \boldsymbol{\mu}, \mathbf{C}, \boldsymbol{\beta}, \mathbf{z}\right) \sim \mathrm{NegBin}\left(m = \theta_n \cdot \mu_i \cdot C\left[z^n, i\right] \cdot \exp^{\mathbf{w}_n^{\mathrm{T}} \beta_i}, size = \zeta\right)$$

We tested size values $\{2, 4, 8, 16, 32, 50, 100, 150, 200\}$, runned 50 repetitions for each and fixed the number of clones to 2, sampled the abundance of each clone in the range $[0.7, 0.3$, and set the distance to 3.

In the third test we analyzed the case in which the input segmentation does not faithfully represent the subclonal segmentation. In the standard CONGAS model, for every gene that maps to a segment, the linear relation between segment ploidy and RNA counts affects all genes. In this test we simulated a violation of this assumption, allowing that in the same segment just a percentage γ of the total number of genes is affected by the copy number state.

For this test we evaluated five setups, where γ starts from 20%, reaches 100% (like in the first test), with steps of 20%. We again performed 50 repetitions for each value of the parameters. To focus on the role of this assumption, avoid mixing with other confounders and providing a more straightforward interpretation of the results, we here performed inference on just two segments, one of which harbouring the same copy-number among clones. Here the number of clones, abundance and distance were fixed to 2, to the range $[0.7, 0.3]$ and to 1.

Triple-negative breast xenograft analysis

Data was obtained from (13), in the form of a count matrix for which we implemented some basic preprocessing. We removed from the count matrix all those genes expressed in less than 5% of the cells, furthermore we filtered the 5% most expressed genes, as their fluctuation could influence too heavily the total segment counts. All the cells with less than 3000 expressed genes were also removed. We calculated the total number of counts in the segment as the sum of the gene counts completely overlapping with a genomic segment.

The input segmentation for CONGAS was again retrieved from (13), even if in the original papers the authors used low-pass single-cell DNA sequencing (instead of bulk, as we assume in CONGAS), and clustered cells defined by similar copy-number events. To obtain a unique input profile for CONGAS we considered the CNC profile of clone A from the original analysis. This is a good surrogate of putative values that we could obtain from a DNA sequencing assay, given that the abundance of this clone is above 80%.

To analyse these data we used a two-steps procedure We first runned the inference with default CONGAS parameters, using a learning rate of 0.01 for 800 inference steps. Posterior probabilities were computed with another 100 steps, using a learning rate of 0.05. In this first run the latent distribution over the CNA values was learned by MAP inference. In this way we set normalization factors and CNV latent variables near to good solutions, as learning the full model together is usually less stable and shows marked multimodality.

In the second step we rerun the same model forcing a full Bayesian setting, where we now can learn the full distribution of our latent variables. In this case we focused on learning mean and variances of the LogNormal densities that model CNA values, while taking fixed the other

parameters identified in the previous run. We obtained the final clustering assignments after filtering the clusters with an abundance of less than 3%.

Differential expression was performed using Seurat (30) and DESeq2 (31). Concordance between our clustering assignment and the labels of clonealign was quantified using Adjusted Rand Index (ARI), the corrected-for-chance version of the Rand index.

Glioblastoma data analysis

Input Smart-Seq scRNAseq data were obtained from (15). The original analysis does not provide an input segmentation for CONGAS. We developed a simple Hidden Markov Model (HMM) to segment exomes directly from RNA counts, and generate the missing segmentation.

HMM definition. Without assuming normalized healthy data, we can segment the genome by using allele frequencies. This approach works better with the Smart-Seq protocol than with the 10x one as it covers the whole gene instead of just the 3'/5' end. We identified heterozygous single nucleotide polymorphisms (SNPs) and calculated the ratio between counts for the major and minor alleles, deriving the minor allele frequency (MAF). In a healthy genome this is 0.5, if we disregard Binomial observational noise.

Our input data consists of mostly cancer cells, and MAFs are therefore no longer necessarily distributed around 0.5. Of course, the resolution of MAFs is limited with single-cell data, and we cannot expect to distinguish all segments breakpoints, or segment perfectly all the genome

The HMM is developed in Pyro (19), and is available in the CONGAS package. The HMM has 6 hidden states; we consider ploidy values above 5 unlikely, at least for the purpose of segmentation. Furthermore, our ability to identify different states from MAFs deteriorates with very high ploidy. Priors on hidden states are described by the following matrix

$$\begin{pmatrix} 1-5t & t & t & t & t & t \\ t & 1-5t & t & t & t & t \\ t & t & 1-5t & t & t & t \\ t & t & t & 1-5t & t & t \\ t & t & t & t & 1-5t & t \\ t & t & t & t & t & 1-5t \end{pmatrix}$$

Here $t > 0$ is the propensity to change the internal HMM state, and is the same across states. In order to perform some filtering on the data, we set a small value for t, with default 10^{-6}. The emission probabilities for MAFs are instead Beta distributions ranging in $[0, 1]$ and are parametrized to be distributed around theoretical MAF values for each ploidy state. Note that the first state identifies LOHs, which are not automatically associated to the real number of copies.

To infer the HMM parameters we use SVI as implemented in Pyro (19), with MAP inference for the transition matrix and the initial state vector. We calculate the posteriors for the state assignments. The emission probabilities, on the other hand, remain fixed and are not learned. A summary of emission probabilities is shown in Supplementary Figure 9, where also runs of this HMM with the glioblastoma Smart-Seq data are shown.

Healthy cells contamination and subclonal detection.

The data for obtaining the MAF and the TPM normalized count matrices were taken from [20].

To avoid RNA-editing the snps have been restricted to those present in the ExAC database [38] with a frequency greater than 10%. Our segmenter works with the MAF for the whole tissue, so the first thing we did was to add the reference and alternative allele values for all cells and create a pseudo-bulk. Sites with coverage less than 20 reads were discarded. Our HMM segmenter was run on those variants with a t of 1e-8 and a median filtering window of 25 sites. After a manual examination we decided to collapse all the states different from a putative LOH (i.e. those different from 1), as the variance of the MAF was too high to confidently call the other ploidy states.

We then runned CONGAS over this segmentation, and obtained an ideal number of clusters equal to 3. Among those we could clearly identify one (cluster number 3) consisting of normal cells, as it was lacking of any LOHs.

We then excluded the cells belonging to that cluster, recalculated the pseudo-bulk MAF, filtered sites with less than 30 reads and rerun the HMM segmenter. Also in this case states with a theoretical MAF value too close to each other were collapsed together, in particular 2 with 6 and 4 with 5. This new putative segmentation was given as an input for a new CONGAS run. The tool was able to identify 3 putative clones characterized by specific chromosome alterations. Differential expression was performed between normal and tumoral cells, between clone 1 and clone 3 and between clone 2 and aggregated clone 1 and 3. Differential expression was performed in the same way as in the breast xenograft example.

Hematopoietic precursors data analysis

Data for 4 healthy donors and 5 patients with bone marrow failure were available from (5). We first performed standard quality check and normalization using Seurat (30). In detail, we removed cells with a high percentage of counts coming from mitochondrial genes (cutoff > 15%) and with counts consistently lower or higher than the majority of the population (>6.8e+6 reads and <2.4e+7 reads).

Data was then normalized in CPM values (counts per million) and transformed in logarithmic scale through a $log(x+1)$ transform. As not all the patients had CNA events we selected patient 1 with monosomy at 7 and an healthy individual as control (labelled with H4). As the dataset

contained different cellular populations to remove the biases caused by marker genes highly expressed in just one population we first median filtered the gene expression counts for each chromosome using the “runmed” function in R, using a window dimension of 11.

Given the absence of a corresponding DNA-derived bulk CNA profile, and therefore lacking a segmentation associated, we had to resort to a custom segmentation, hopefully suitable to target chromosome-level aneuploidy events. We indeed segmented the genome at the level of whole chromosomes, and assumed a prior ploidy of 2 for each of them, which seems a reasonable baseline choice. After this we runned CONGAS using respectively 3.5 and 0.01 as scale and rate parameter for the library size factors prior; we had to change these values compared to other analysis as our default were optimized on 10x protocols. Clusters with less than 10 cells were discarded as outliers. Differential expression was performed with DESeq2 as for the other datasets.

References

1. Greaves M, Maley CC. Clonal evolution in cancer. Nature. 2012 Jan;481(7381):306–13.
2. Nik-Zainal S, Van Loo P, Wedge DC, Alexandrov LB, Greenman CD, Lau KW, et al. The life history of 21 breast cancers. Cell. 2012 May 25;149(5):994–1007.
3. Turajlic S, Sottoriva A, Graham T, Swanton C. Resolving genetic heterogeneity in cancer. Nat Rev Genet. 2019;20(7):404–16.
4. McGranahan N, Swanton C. Biological and therapeutic impact of intratumor heterogeneity in cancer evolution. Cancer Cell. 2015 Jan 12;27(1):15–26.
5. X Z, S G, Z W, S K, X F, Q L, et al. Single-cell RNA-seq reveals a distinct transcriptome signature of aneuploid hematopoietic cells. Blood. 2017 Oct 13;130(25):2762–73.
6. Martincorena I, Roshan A, Gerstung M, Ellis P, Van Loo P, McLaren S, et al. High burden and pervasive positive selection of somatic mutations in normal human skin. Science. 2015 May 22;348(6237):880–6.
7. Martincorena I. Somatic mutation and clonal expansions in human tissues. Genome Med. 2019 May 28;11(1):35.
8. Acar A, Nichol D, Fernandez-Mateos J, Cresswell GD, Barozzi I, Hong SP, et al. Exploiting evolutionary steering to induce collateral drug sensitivity in cancer. Nat Commun. 2020 21;11(1):1923.
9. Caravagna G, Heide T, Williams MJ, Zapata L, Nichol D, Chkhaidze K, et al. Subclonal reconstruction of tumors by using machine learning and population genetics. Nat Genet. 2020 Sep;52(9):898–907.
10. Picelli S, Faridani OR, Björklund ÅK, Winberg G, Sagasser S, Sandberg R. Full-length RNA-seq from single cells using Smart-seq2. Nat Protoc. 2014 Jan;9(1):171–81.
11. Lähnemann D, Köster J, Szczurek E, McCarthy DJ, Hicks SC, Robinson MD, et al. Eleven grand challenges in single-cell data science. Genome Biol. 2020 Feb 7;21(1):31.
12. Macaulay IC, Haerty W, Kumar P, Li YI, Hu TX, Teng MJ, et al. G&T-seq: parallel sequencing of single-cell genomes and transcriptomes. Nat Methods. 2015 Jun;12(6):519–22.
13. Campbell KR, Steif A, Laks E, Zahn H, Lai D, McPherson A, et al. clonealign: statistical integration of independent single-cell RNA and DNA sequencing data from human cancers. Genome Biol. 2019 Mar 12;20(1):54.

14. CaSpER identifies and visualizes CNV events by integrative analysis of single-cell or bulk RNA-sequencing data | Nature Communications [Internet]. [cited 2021 Jan 24]. Available from: https://www.nature.com/articles/s41467-019-13779-x
15. Fan J, Lee H-O, Lee S, Ryu D-E, Lee S, Xue C, et al. Linking transcriptional and genetic tumor heterogeneity through allele analysis of single-cell RNA-seq data. Genome Res. 2018;28(8):1217–27.
16. Watkins TBK, Lim EL, Petkovic M, Elizalde S, Birkbak NJ, Wilson GA, et al. Pervasive chromosomal instability and karyotype order in tumour evolution. Nature. 2020 Nov;587(7832):126–32.
17. 10X Genomics. 10XGenomics/vartrix: Single-Cell Genotyping Tool [Internet]. Available from: https://github.com/10xgenomics/vartrix
18. Sarkar A, Stephens M. Separating measurement and expression models clarifies confusion in single cell RNA-seq analysis. bioRxiv. 2020 Apr 18;2020.04.07.030007.
19. Bingham E, Chen JP, Jankowiak M, Obermeyer F, Pradhan N, Karaletsos T, et al. Pyro: Deep Universal Probabilistic Programming. J Mach Learn Res. 2019;20(28):1–6.
20. Eirew P, Steif A, Khattra J, Ha G, Yap D, Farahani H, et al. Dynamics of genomic clones in breast cancer patient xenografts at single-cell resolution. Nature. 2015 Feb;518(7539):422–6.
21. Patel AP, Tirosh I, Trombetta JJ, Shalek AK, Gillespie SM, Wakimoto H, et al. Single-cell RNA-seq highlights intratumoral heterogeneity in primary glioblastoma. Science. 2014 Jun 20;344(6190):1396–401.
22. Vlachogiannis G, Hedayat S, Vatsiou A, Jamin Y, Fernández-Mateos J, Khan K, et al. Patient-derived organoids model treatment response of metastatic gastrointestinal cancers. Science. 2018 Feb 23;359(6378):920–6.
23. Caravagna G, Giarratano Y, Ramazzotti D, Tomlinson I, Graham TA, Sanguinetti G, et al. Detecting repeated cancer evolution from multi-region tumor sequencing data. Nat Methods. 2018;15(9):707–14.
24. Blei DM, Kucukelbir A, McAuliffe JD. Variational Inference: A Review for Statisticians. J Am Stat Assoc. 2017 Apr 3;112(518):859–77.
25. Kingma DP, Ba J. Adam: A Method for Stochastic Optimization. ArXiv14126980 Cs [Internet]. 2017 Jan 29 [cited 2020 Dec 4]; Available from: http://arxiv.org/abs/1412.6980
26. Paszke A, Gross S, Massa F, Lerer A, Bradbury J, Chanan G, et al. PyTorch: An Imperative Style, High-Performance Deep Learning Library. Adv Neural Inf Process Syst. 2019;32:8026–37.
27. Ferguson TS. A Bayesian Analysis of Some Nonparametric Problems. Ann Stat. 1973 Mar;1(2):209–30.
28. Ishwaran H, James LF. Gibbs Sampling Methods for Stick-Breaking Priors. J Am Stat Assoc. 2001 Mar 1;96(453):161–73.
29. Blei DM, Jordan MI. Variational inference for Dirichlet process mixtures. Bayesian Anal. 2006 Mar;1(1):121–43.
30. Stuart T, Butler A, Hoffman P, Hafemeister C, Papalexi E, Mauck WM, et al. Comprehensive Integration of Single-Cell Data. Cell. 2019 Jun 13;177(7):1888-1902.e21.
31. Love MI, Huber W, Anders S. Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2. Genome Biol. 2014 Dec 5;15(12):550.

Main Text Figures

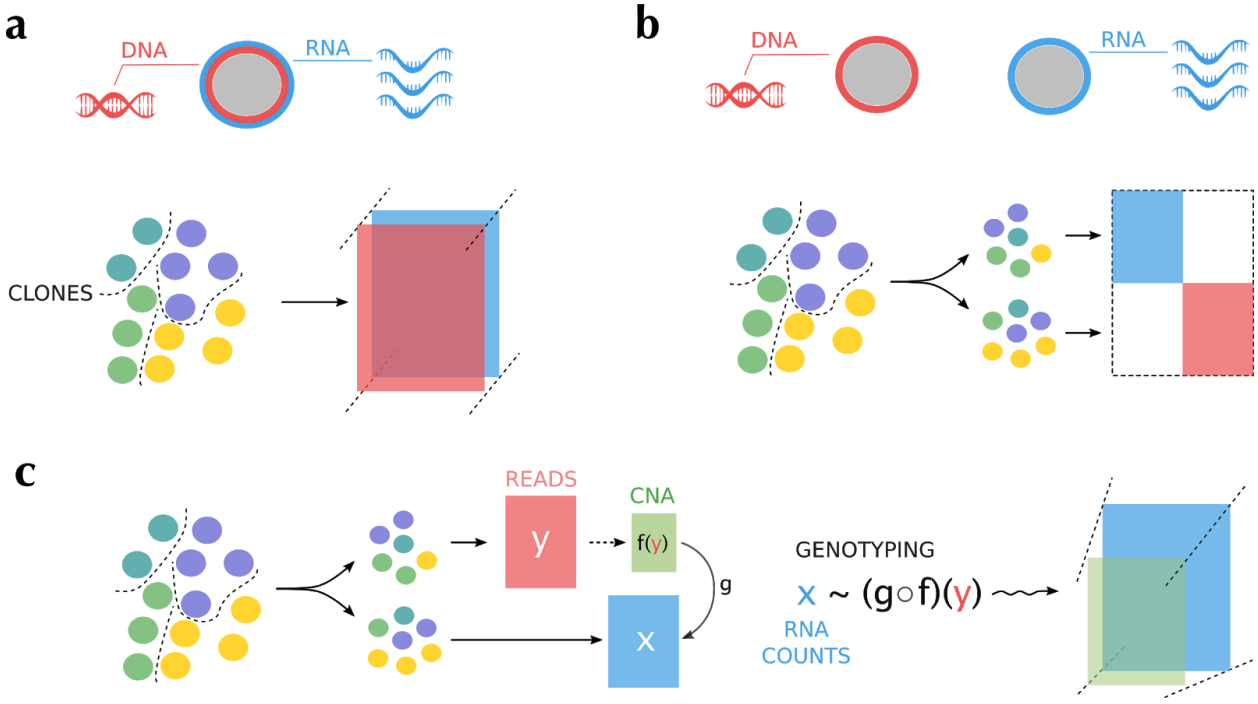


Figure 1. a. Single-cell multi-omics from a population of heterogeneous cells, phylogenetically related into 4 clones (distinct colours). In this assay we simultaneously measure, for instance, the DNA and RNA from the same cell, and use a tensor to describe the data. **b.** If we cannot implement a multi-omics assay, we can split the cancer cells into groups and generate independent measures; these data cannot be directly overlapped. **c.** The measurements obtained from split data (x and y) can be integrated, if we have a model for the relation between the sequenced molecules. Assuming DNA and RNA, for instance, we can first use DNA reads y and determine Copy Number Alterations (CNA). Notice that dimensionality of $f(y)$ can be lower than that of y, as it depends the data used in y. If y are low-pass single-cell measurements, we can call single-cell copy numbers, cluster cells and set $f(y)$ to be a matrix of copy number clones. If y are reads from bulk DNA sequencing, $f(y)$ can be a vector reporting clonal bulk copy numbers. In both cases with breakpoints and ploidy data we can use the DNA amounts to predict the expected RNA transcripts, assuming a link function g applied to $f(y)$. In this way we genotype CNAs on top of RNA, emulating a joint measurement (observed RNA x and inferred DNA, $f(y)$) for each cell.

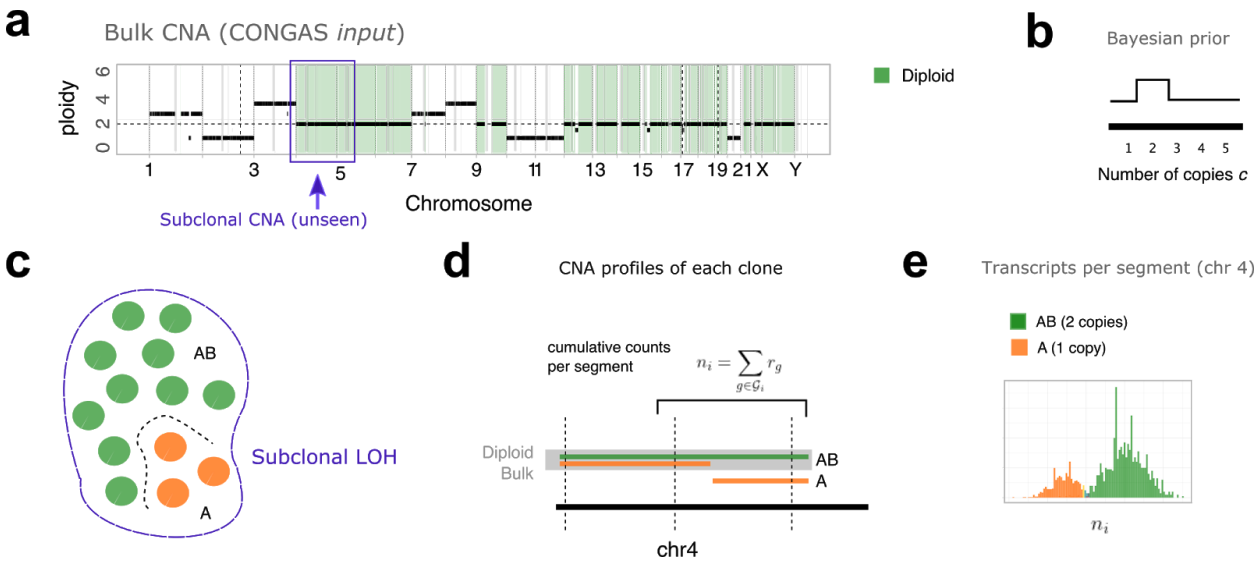


Figure 2. a. The CONGAS model for CNA Genotyping from single-cells takes as input a bulk segmentation with total copy numbers (i.e., segment ploidy(. The input segments are genotyped on top of single-cell RNA sequencing data. In this example we picture a small subpopulation of cancer cells (i.e., a subclone), harbouring a CNA around chromosome 4. **b.** CONGAS uses the total ploidy (i.e., total CNA) of each segment to build a prior for the parameter that model copy number in single cells. For chromosome 4, this would be peaked at 2 since the tumour bulk does not show evidence of the subclone. **c.** The subclone (orange) is here characterised by an LOH on chromosome 4. The genotypes are briefly referred to as AB and A, referring to the major and minor alleles. **d.** CONGAS assumes a linear model of the relation between the total number of RNA transcripts r_i in segment i, and the segment ploidy c from DNA (the sum of major and minor allele counts). These values are computed from the set of genes G_i that map to a segment. **e.** CONGAS normalises counts for library size and the number of genes that map to each segment. For this subclone, our model expects to see more transcripts in cells that are diploid (AB), compared to those that have undergone LOH (A).

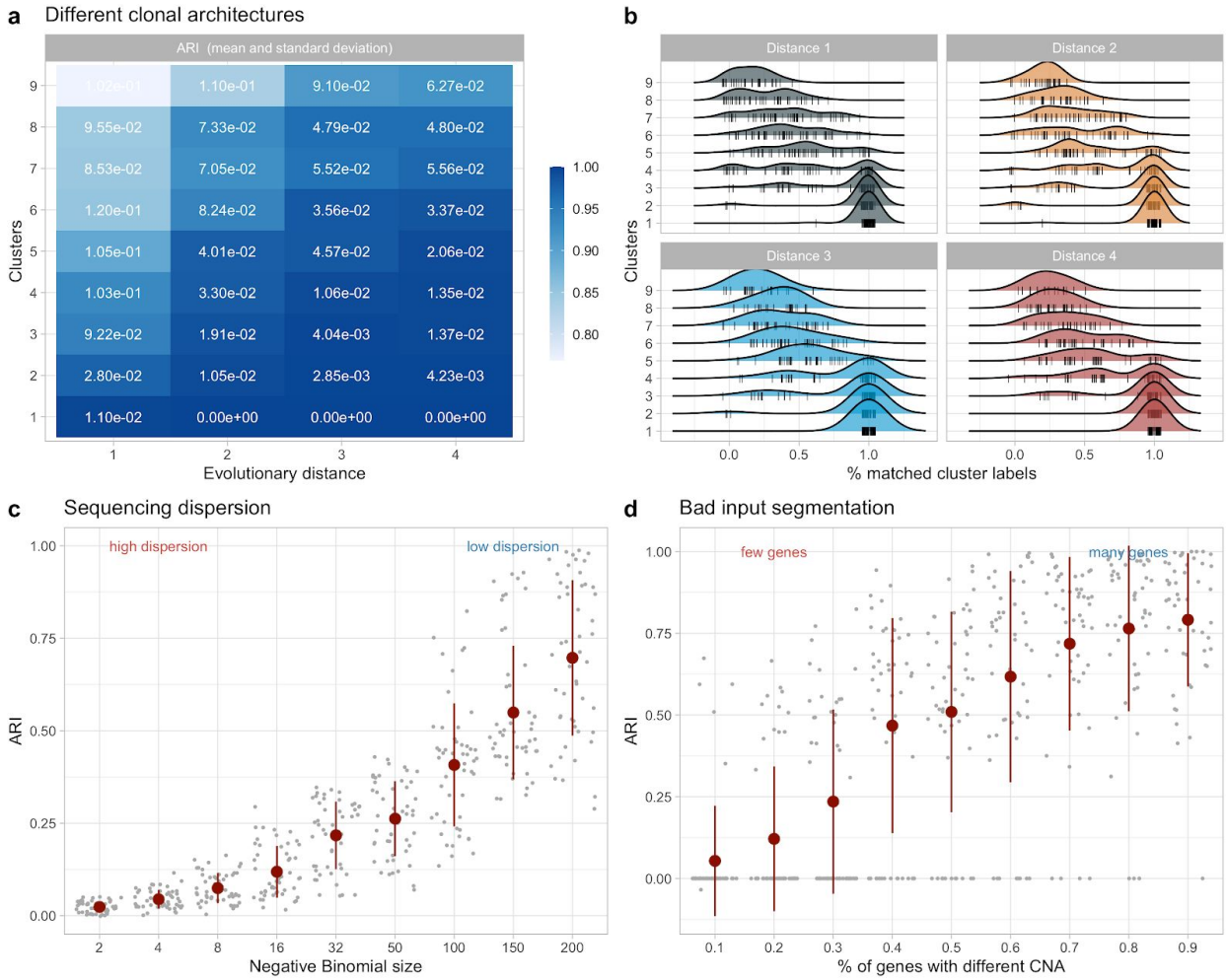


Figure 3. a CONGAS synthetic tests with different subclonal architectures, obtained sampling clone trees with variable number of nodes. The degree of tumour heterogeneity is tuned by an evolutionary distance, which counts the number of CNAs that a subclone acquires, relative to its ancestor. The bulk input profile for CONGAS is generated by considering CNA segments from the most prevalent clone. We scan models with up to 9 clones, with distance ranging from 1 to 4. The performance is measured by using the adjusted rand index (ARI) between the simulated and retrieved cell assignments. The heatmap colour reflects the mean, and the standard deviation is annotated. **b.** Smoothed density for the percentage of cluster labels matched in every simulation, split by trees of increasing distance. **c.** ARI from synthetic tests simulating sequencing overdispersion after changing the Poisson model in CONGAS with a Negative Binomial and variable dispersions. For each test a clonal architecture with 2 clones. is simulated. **d.** ARI from synthetic tests simulating input segmentations that are misleading, meaning that only a subset of genes that map to a segment is affected by the CNA, so input breakpoints from bulk do not match subclonal breakpoints. For each test a clonal architecture with 2 clones and a fixed number of 2 segments is simulated.

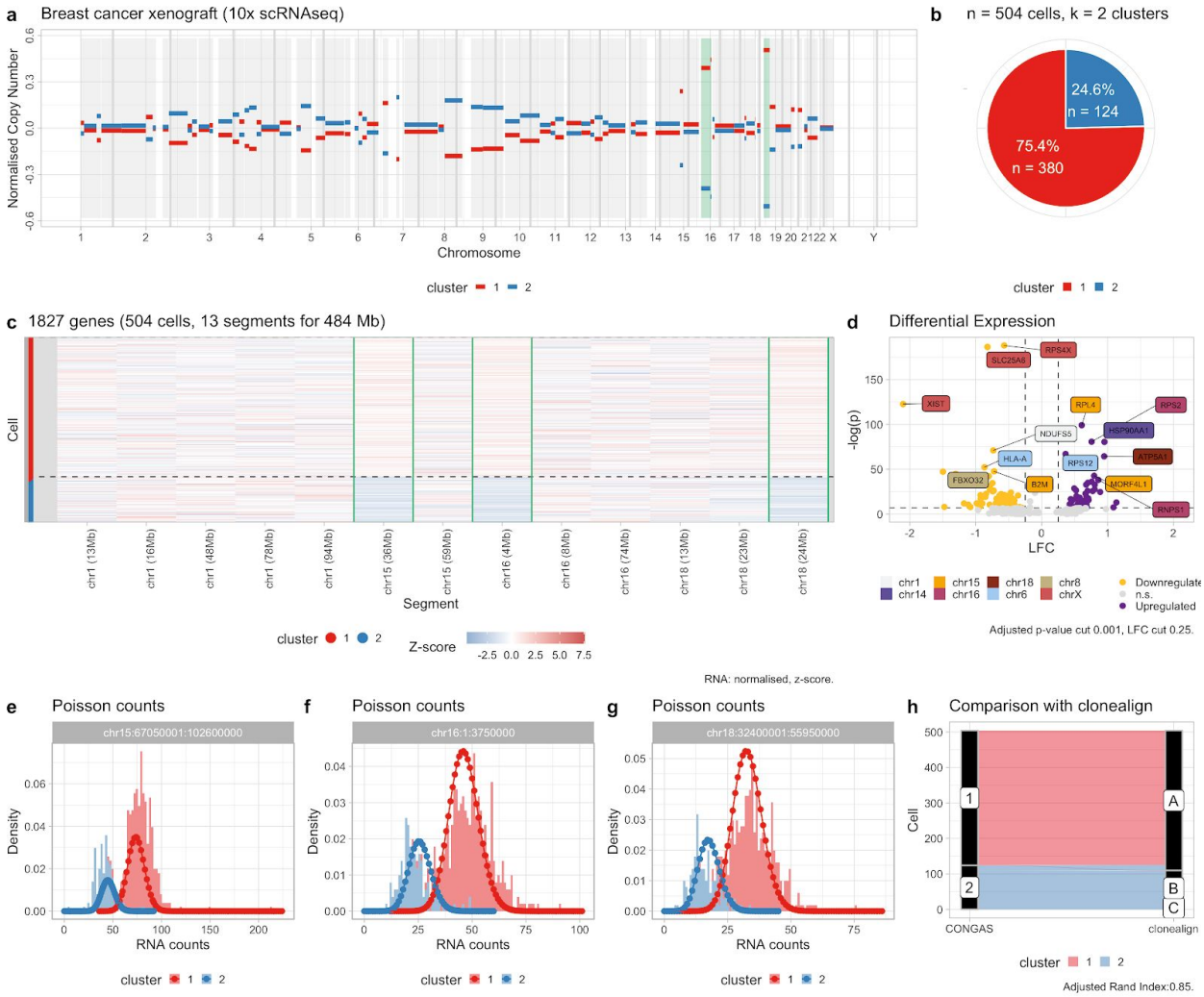


Figure 4. a. Analysis of $n = 504$ single cells from a breast xenograft, sequenced by using 10x technology in (13). CONGAS finds $k = 2$ clusters of cells, which show significant differences in the counts of RNA transcripts mapping to some segments of chromosomes 15, 16 and 18. **b.** The largest clonal population consists of $n = 380$ cells (~75% of total), the smallest one $n = 124$ cells (~25% of total). **c.** Raw RNA counts (normalised per segment, plot using the z-score) showing transcriptional counts in a subset of the tumour genome (same segments highlighted as in panel a). Chromosome 1 is included as a graphical control: according to CONGAS no significant copy number differences exist among the clones (i.e., the CNA is clonal). Also other segments from the same affected chromosomes are clonal. **d.** Genome-wide clone-specific Differential Expression analysis highlights $n = 212$ genes that are either upregulated or downregulated ($p < 0.01$ and absolute log-fold change > 0.25 to determine up-regulation); notice that some of those genes do not reside in genome portions with CNAs that characterise the populations. **e, f, g.** RNA transcripts count for the genes mapping to the segment on chromosome 15, 16 and 18, which are highlighted in panels a/b. The densities on top of the histograms are the Poisson mixtures inferred by CONGAS. **h.** Comparison between clustering assignments of CONGAS and clonealign on these data (13). The input of clonealign are the CNA profiles of three clones, obtained from low-pass single-cell whole-genome sequencing data. The tool then assigns the RNA data to each one of the input clones. The

only difference between the analyses is a very small cluster that is not found by CONGAS. All the other cells are assigned exactly the same cluster by both analyses (Adjusted Rand Index 0.85).

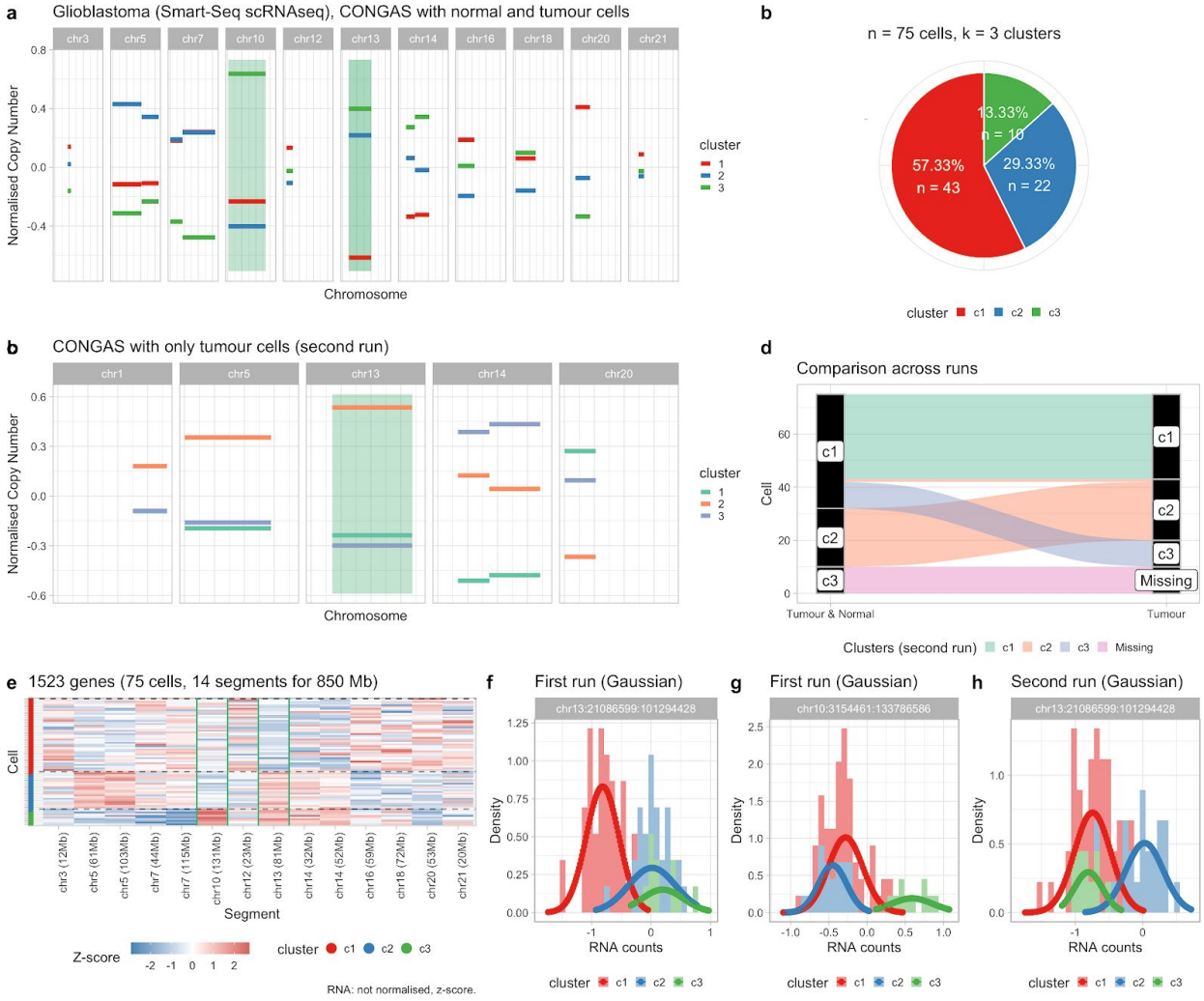


Figure 5. a. Analysis of glioblastoma Smart-Seq data for $n = 75$ single cells, a small portion of which are known to be non-cancerous (15,21). The input segmentation for CONGAS is unavailable, we retrieve it here by segmenting the minor allele frequency of heterozygous polymorphism, using an Hidden Markov Model. This panel shows a first run of CONGAS with all input cells; cluster 3 is the cluster of normal cells. **b.** Mixing proportions for the first CONGAS run with both tumour and normal cells. **c.** CONGAS run after that normals cells have been removed. **d.** Mapping between clusters obtained in the first and second run. Cluster 3 from the first run are normal cells; cluster 1 from the first run splits into two clusters in the second run. **e.** Input raw data (z-score), highlighting segments as in panel (a). **f, g.** Input data and Gaussian

density for the segments on chromosome 10 and 13, in the first run. **h.** Input data and Gaussian density for the segment on chromosome 13, in the second run.

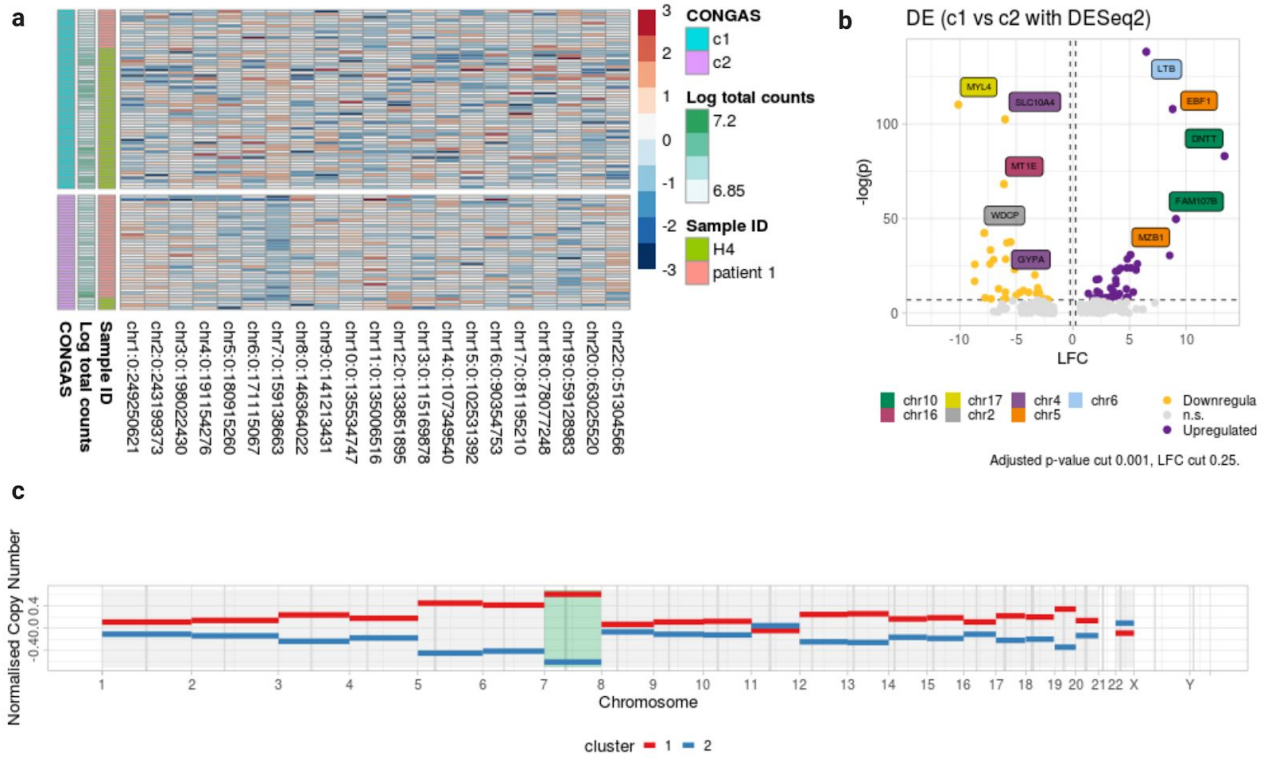
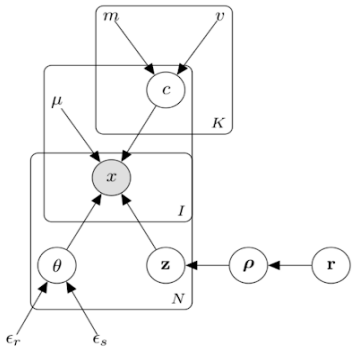


Figure 6. a. Analysis of bone marrow Smart-Seq samples for $n = 100$ cells and 2 patients, one healthy (H4) and one with bone marrow failure (patient 1). As these dataset does not have a precalculated input CNA segmentation, we have aggregated gene counts at the level of whole chromosomes and genotyped those large-scale events. This panel shows how CONGAS is clearly able to distinguish between the healthy population and the one with the disease, in this case harboring a deletion in chromosome 7 in a subset of cells of patient 1. It can also be noticed how clustering assignments are not correlated with the total number of counts, suggesting that CONGAS can correctly normalize for sequencing efficiency. **b.** Volcano plot showing the gene differentially expressed between the two clusters; note that the highlighted genes map off from chromosome 7. **c.** Genome wide visualization of the CNV profiles inferred by CONGAS highlights the strongest signal on the monosomy for chromosome 7, which is causative for bone marrow failure in patient 1.

Supplementary Figures

- Supplementary Figure S1. *CONGAS plate notation*
- Supplementary Figure S2. *Performance for different clonal architectures.*
- Supplementary Figure S3. *Performance with sequencing overdispersion.*
- Supplementary Figure S4. *Performance with miscalled segments.*
- Supplementary Figure S5. *Input raw counts with DEGs for breast xenograft.*
- Supplementary Figure S6. *Genome-wide DE for breast xenograft.*
- Supplementary Figure S7. *Fit report for breast xenograft.*
- Supplementary Figure S8. *Comparison with clonelaign for breast xenograft.*
- Supplementary Figure S9. *HMM runs on the GBM dataset.*
- Supplementary Figure S10. *Performances with CPU and GPU.*

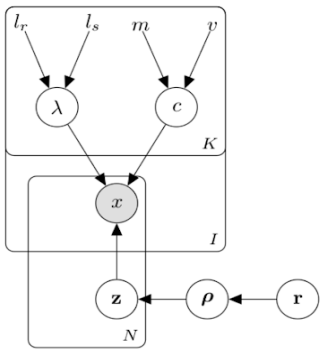
A



Distributions:

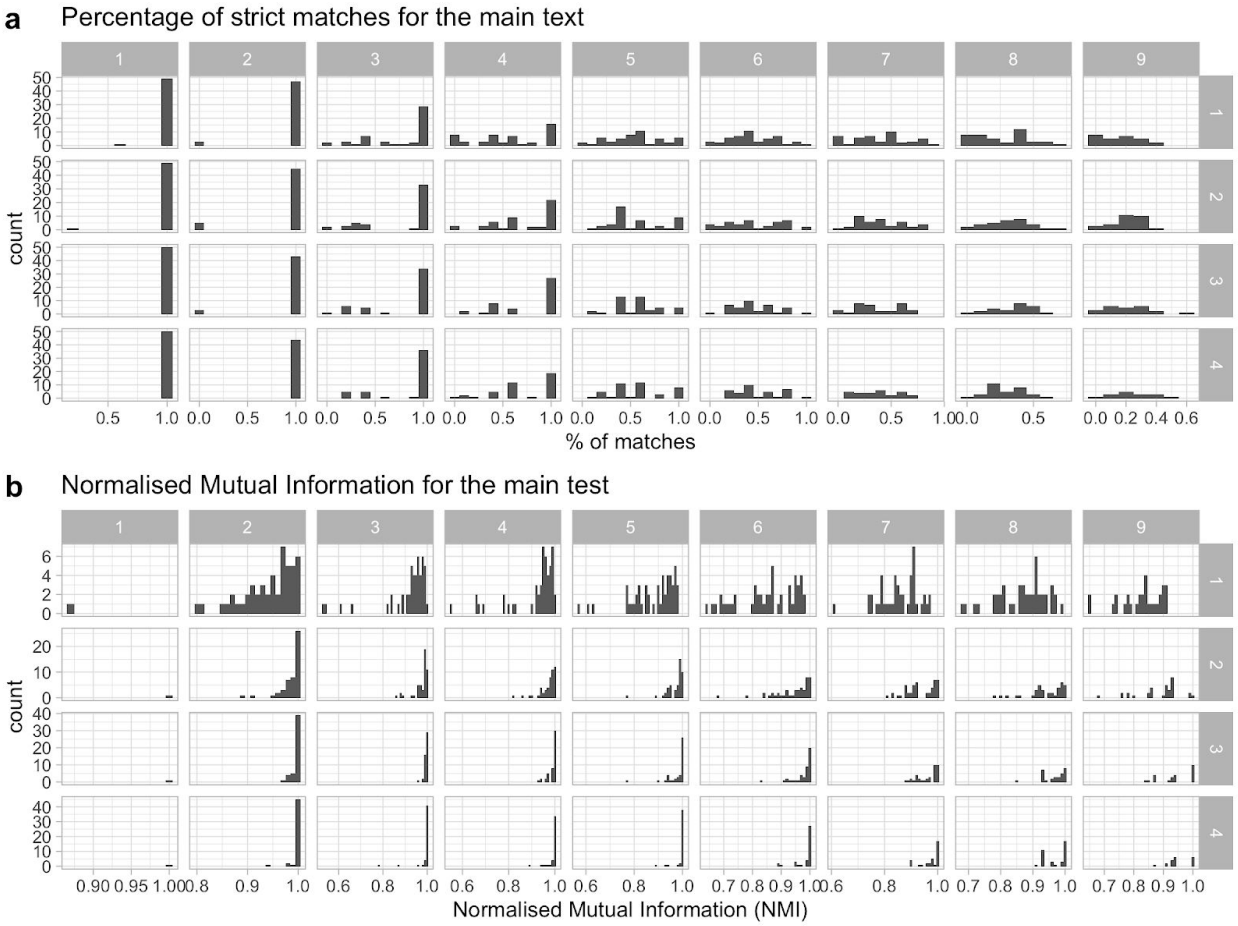
$c \sim LogNormal(m, v)$
$\theta \sim Gamma(\epsilon_r, \epsilon_s)$
$x \sim Poisson(\theta \cdot \mu \cdot c_{[z_n=1]})$
$\mathbf{z} \sim Cat(\boldsymbol{\rho})$
$\boldsymbol{\rho} \sim Dirichlet(\mathbf{r})$
Constants:
μ = number of genes in a segment
$\mathbf{r}$ = concentration vector

B

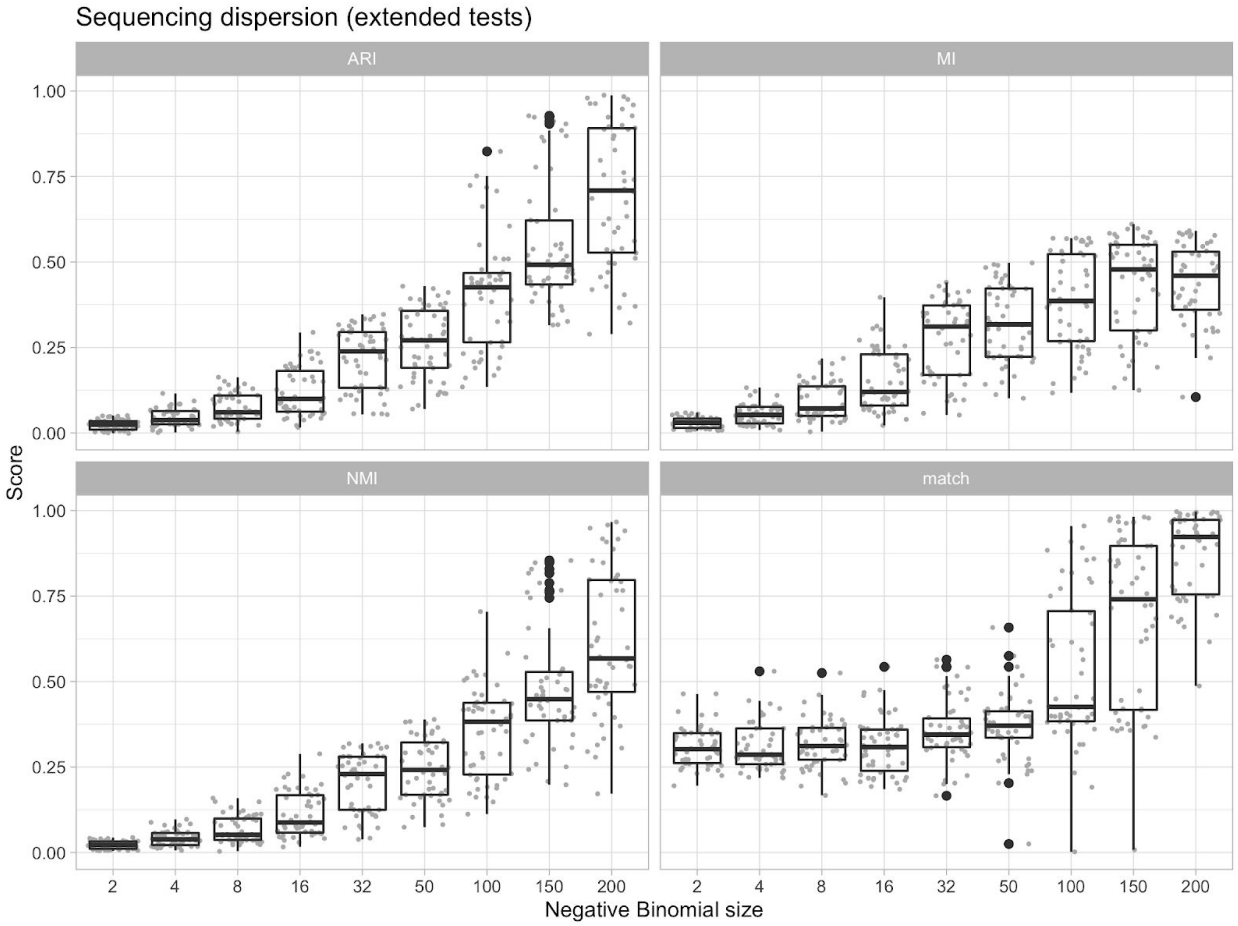


Distributions:
$c \sim Normal(m, v)$
$x \sim Normal(c_{[z_n=1]}, \lambda_{[z_n=1]})$
$\lambda \sim Uniform(l_r, l_s)$
$\mathbf{z} \sim Cat(\boldsymbol{\rho})$
$\boldsymbol{\rho} \sim Dirichlet(\mathbf{r})$
Constants:
$\mathbf{r}$ = concentration vector

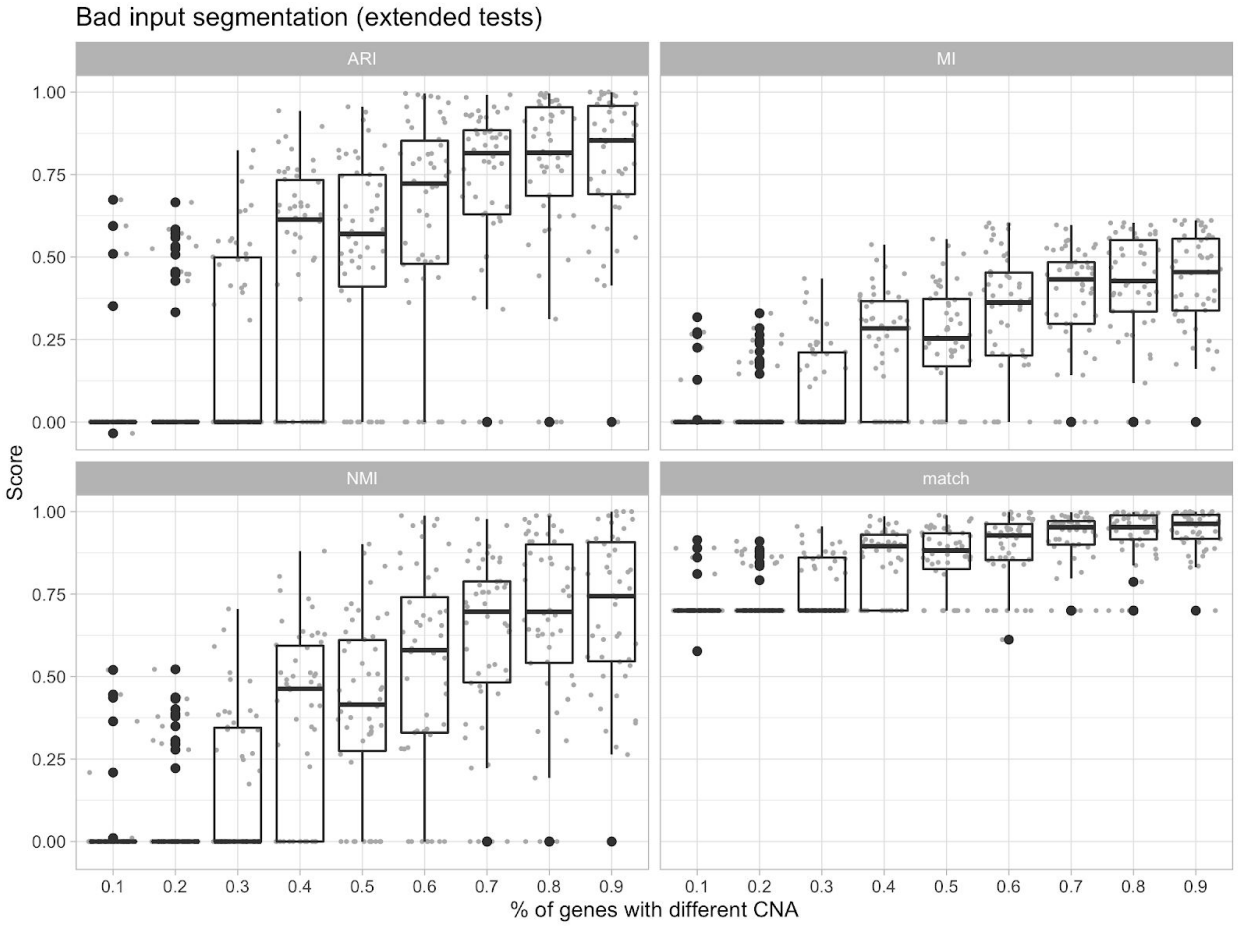
Supplementary Figure S1. CONGAS probabilistic graphical models in plate notation. **A.** CONGAS main model for counts as a finite mixture of Poissons; here N indexes the number of cells, while I and K represent respectively the total number of segments and clusters. Note that the latent variables **z** and all other variables are vectors of dimension K. **B.** CONGAS alternative model as a finite mixture of independent Gaussian distribution, which can process continuous inputs. We assume in this case that the data is not only discretized, but has already been normalized for library size and other confounding variables.



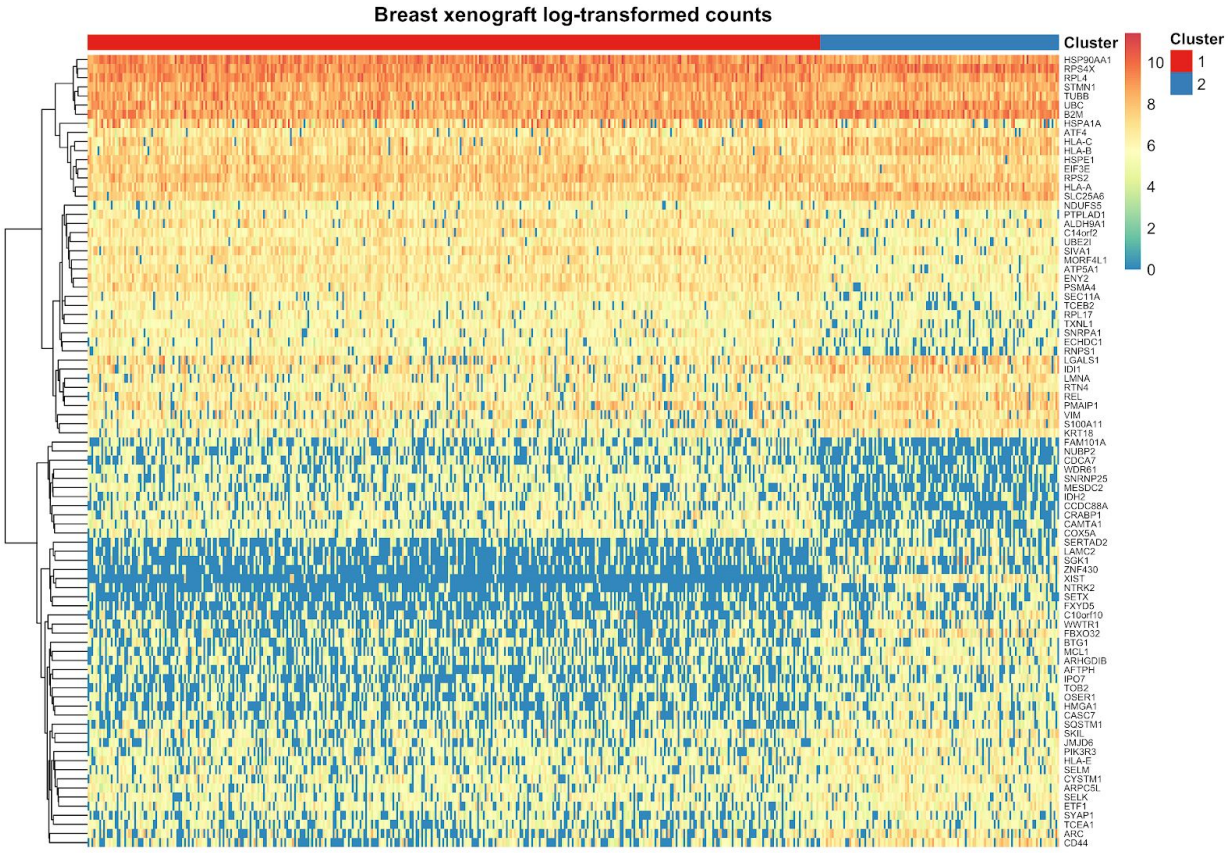
Supplementary Figure S2. a,b. Performance for the main synthetic test, where we scan different numbers of clusters assembled on trees with increasing evolutionary distance. We show the percentage of cluster labels that match in a simulation. We do this by sorting clusters by size, assigning them labels, and then assigning each cell to a cluster according to the model. We then check position by position if the clusters match - this is a strong penalty. In bottom we show NMI, the mutual information normalised in [0,1].



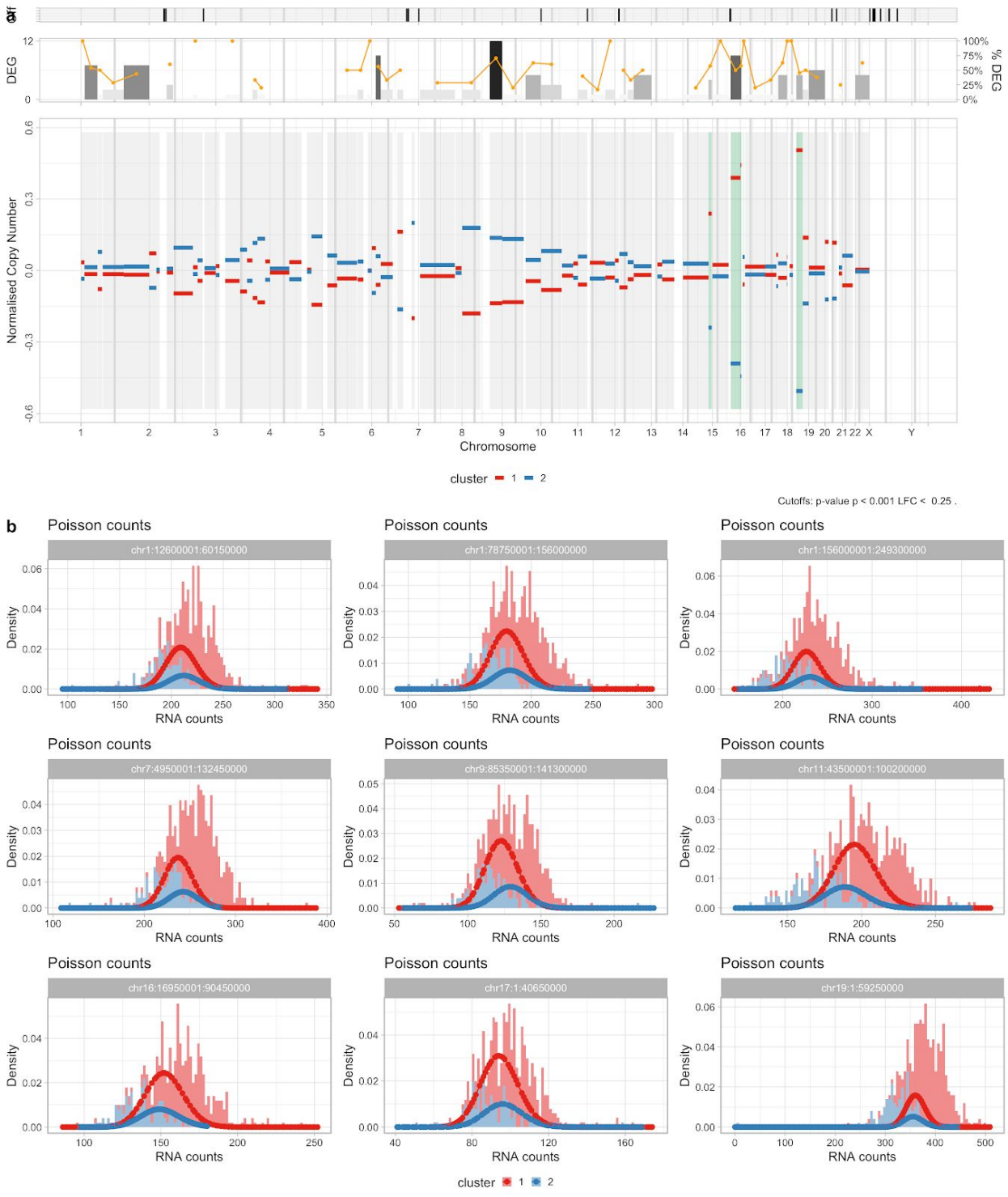
Supplementary Figure S3. Performance for synthetic tests with sequencing overdispersion. In this case we use different parameters of a Negative Binomial distribution to generate read counts (Main Text). The performance shows a clear trend; here the scores are computed between simulated and inferred clustering labels. ARI is the adjusted rand index, MI and NMI the mutual information and its normalised extension. Score match reports the proportion of cells with the exact same cluster label (simulated versus inferred).



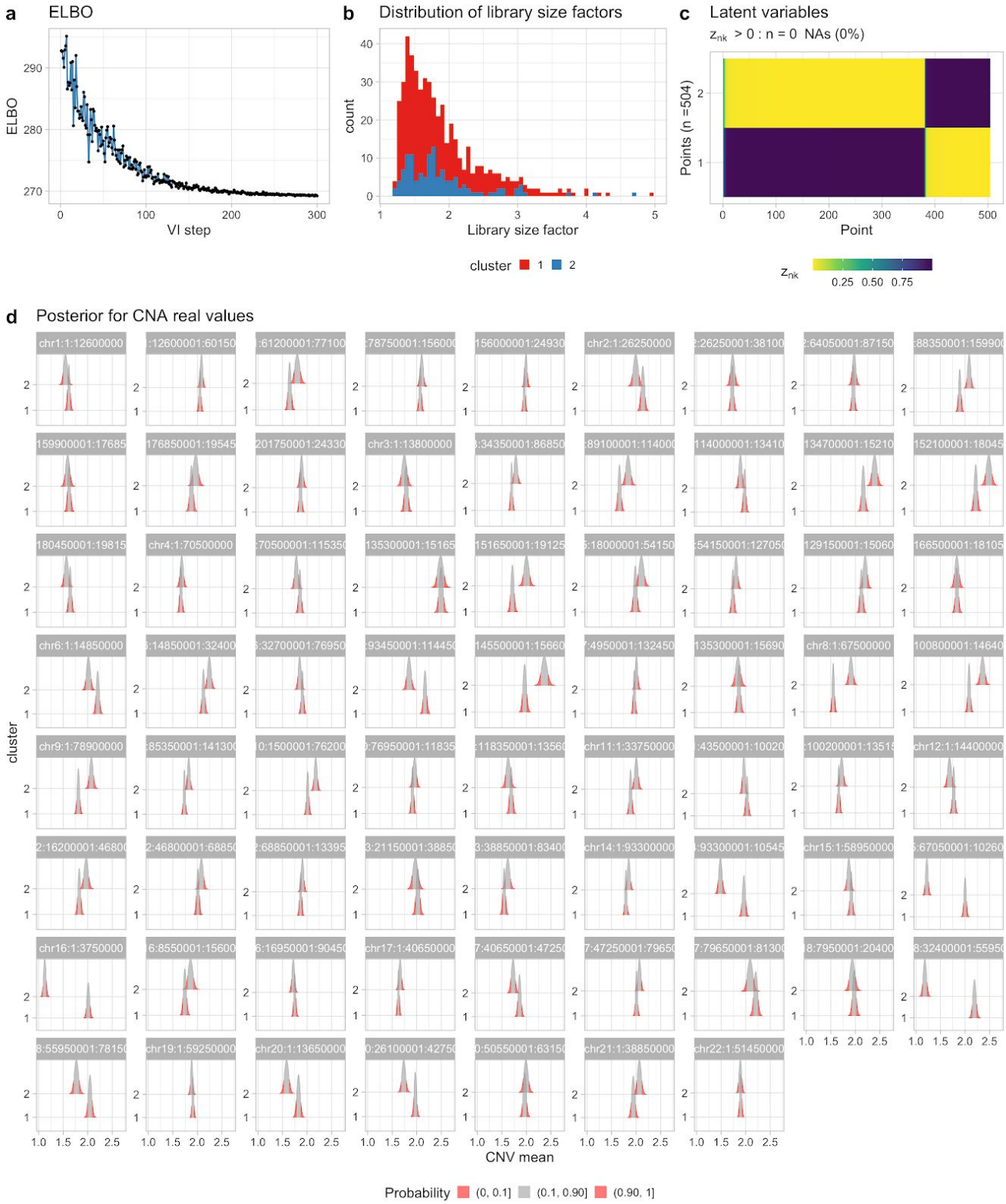
Supplementary Figure S4. Performance for synthetic tests with subclones that have CNA segments where only a portion of the mapped genes obeys the linear DNA/RNA relation. In practice, this tests for the presence of subclonal CNAs that involve segments smaller than the ones given in input to CONGAS. In this case we use different proportions of genes, ranging from 10% to 90% (see also Main Text). The performance shows a clear trend; the same scores in Supplementary Figure S2 are reported.



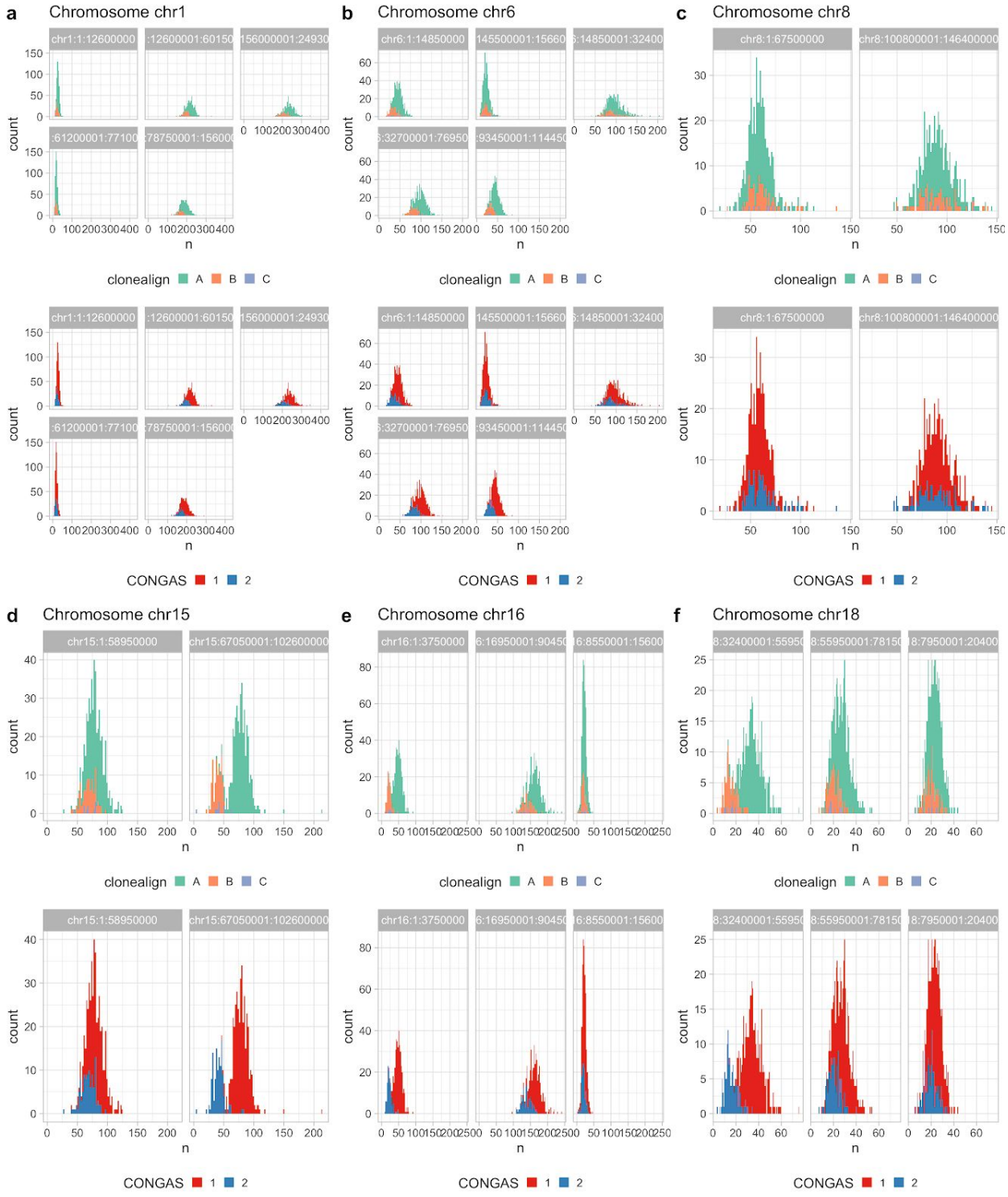
Supplementary Figure S5. Heatmap for the input raw counts of the breast cancer xenograft discussed in the Main Text. Each column is a cell, each row one of 212 differentially expressed genes; count values are log transformed. Rows are clustered by hierarchical clustering.



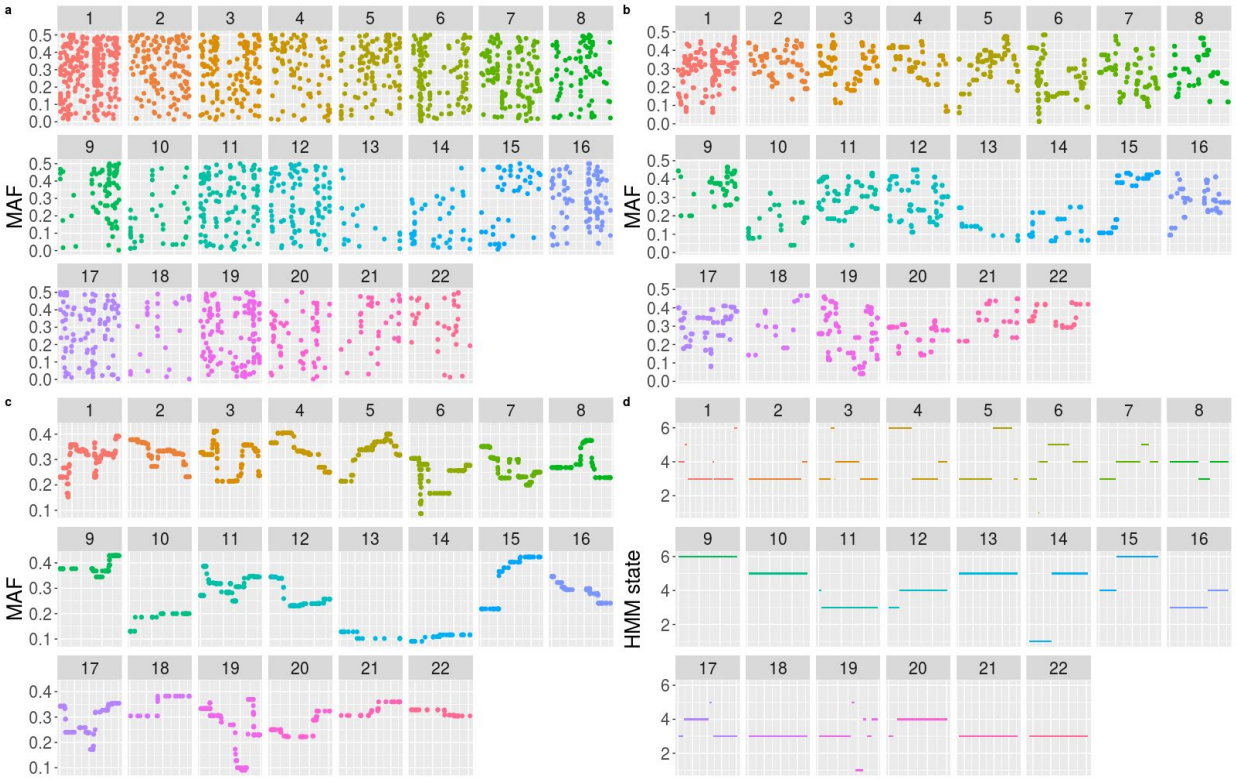
Supplementary Figure S6. a. Genome-wide DEGs for the breast cancer xenograft discussed in the Main Text. Above the genome we show the number and the percentage of DEGs per segment; the top marks are DEGs off the input CNA segments. **b.** Data density and CONGAS mixture for segments with >250 genes (the largest segments of this tumour).



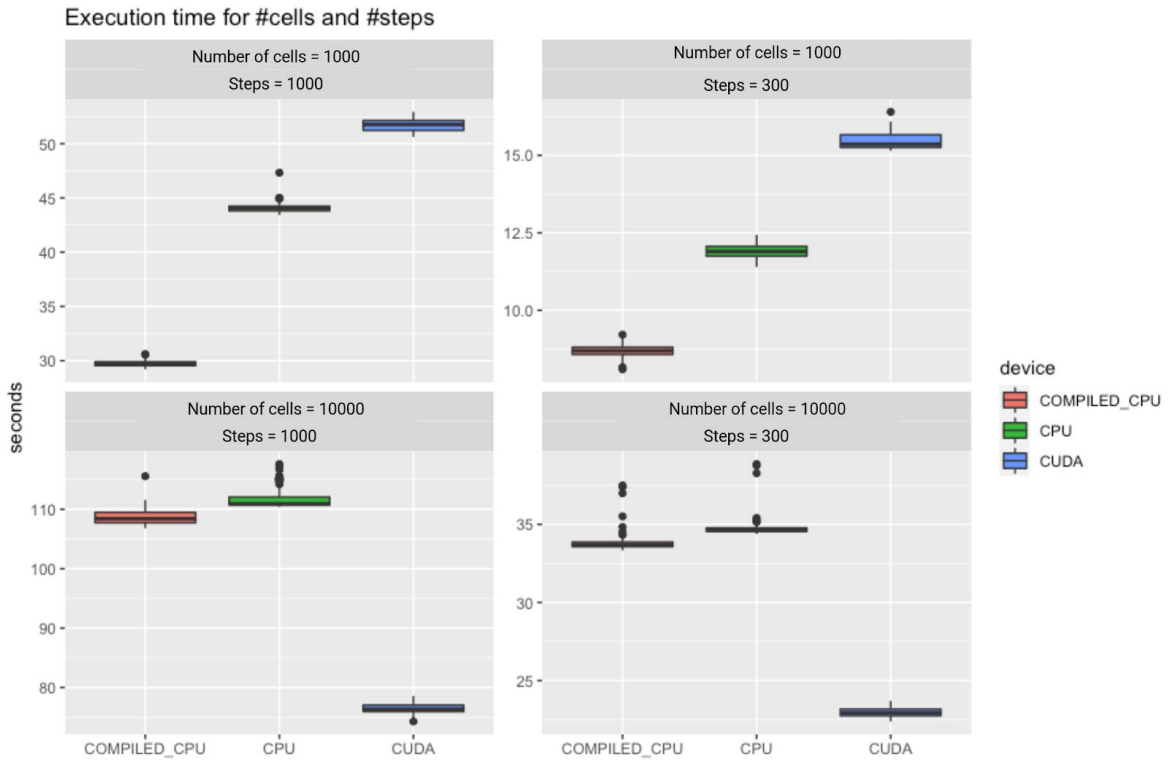
Supplementary Figure S7. a. ELBO to analyse the breast cancer xenograft discussed in the Main Text. **b.** Library size factors per clone, inferred by CONGAS. **c.** Model's latent variables show a clear separation of the clusters



Supplementary Figure S8. a-e. For every segment in a subset of the input chromosomes we plot the data coloured accordingly to the clustering assignments obtained by clonealign (top), and CONGAS (bottom). Clone C from clonealign is very difficult to identify; both other clusters match perfectly.



Supplementary Figure S9. Results of the first run for our HMM semgenter **a-c.** Plots of the raw MAF data after median filtering over chromosomes. The three plots show increasing width for the median window, in order from the upper left 1 (no filtering at all), 5, 21. These plots also highlight the intrinsic noise in the data, which makes it hard to call confidently the CNV regions **d.** Plot of the HMM states after inference provide breakpoints and segmentation values.



Supplementary Figure S10. Execution time for CONGAS runned on two simulated dataset, one with 1000 cells and the other with 10000 cells. For each dataset we timed 100 executions for respectively 300 and 1000 gradient update steps in 3 different settings: standard python interpreter, JIT compiler and GPU/CUDA. From the plot it is clear how for small cell numbers the CPU is faster, but as the number of cells starts to increase the GPU can give an effective speed up to the calculation. All calculations were performed on a machine with 2 Intel Xeon vCPU @2.2GHz, 13 GB of RAM and an NVIDIA Tesla T4.