# CellPhy: accurate and fast probabilistic inference of single-cell phylogenies from scDNA-seq data

*Alexey Kozlov[1#], Joao Alves[2,3,4#], Alexandros Stamatakis[1,5] and David Posada[2,3,4*]*

[1]The Exelixis Lab, Computational Molecular Evolution Group, Heidelberg Institute for Theoretical Studies, Heidelberg D-69118, Germany
[2]CINBIO, Universidade de Vigo, 36310 Vigo, Spain
[3]Department of Biochemistry, Genetics, and Immunology, Universidade de Vigo, 36310 Vigo, Spain
[4]Galicia Sur Health Research Institute (IIS Galicia Sur), SERGAS-UVIGO.
[5]Institute for Theoretical Informatics, Karlsruhe Institute of Technology, D-76128, Germany

[#]Shared first-authorship
*Corresponding author

## Abstract

We have developed a maximum likelihood framework called CellPhy for inferring phylogenetic trees from single-cell DNA sequencing (scDNA-seq) data, that can be directly applied to somatic cells and clones. CellPhy is based on a finite-site Markov nucleotide substitution model with 10 diploid states, akin to those typically used in statistical phylogenetics. It includes a dedicated error function for single cells that explicitly incorporates amplification/sequencing error and allelic dropout (ADO). Moreover, it can explicitly consider the uncertainty of the variant calling process by using genotype likelihoods as input. We implemented CellPhy in a widely used open-source phylogenetic inference package (RAxML-NG) that provides statistical confidence measurements on the estimated tree and scales particularly well on large phylogenies with hundreds or even thousands of cells. To benchmark CellPhy, we carried out 19,400 coalescent simulations of cell samples from exponentially-growing tumors for which the true phylogeny was known. We evolved single-cell diploid DNA genotypes along the simulated genealogies under different scenarios including infinite- and finite-sites nucleotide mutation models, trinucleotide mutational signatures, sequencing and amplification errors, allele dropouts, and doublet cells. Our simulations suggest that CellPhy is robust to amplification/sequencing errors and to ADO and that it outperforms the state-of-the-art methods under realistic scDNA-seq scenarios both in terms of accuracy and speed. In addition, we sequenced 24 single-cell whole genomes from a colorectal cancer, and together with three published scDNA-seq data sets, analyzed them to illustrate how CellPhy can provide more reliable biological insights than competing methods. CellPhy is freely available at https://github.com/amkozlov/cellphy.

# Introduction

The study of single cells is revolutionizing biology, unveiling unprecedented levels of genomic and phenotypic heterogeneity within otherwise seemingly homogeneous tissues [1–6]. Understanding this somatic mosaicism has applications in multiple areas of Biology due to its intrinsic connection to development, aging, and disease [7,8]. However, the analysis of single-cell genomic data is not devoid of challenges [5,9], including the development of more integrative, scalable, and biologically realistic models of somatic evolution that are able to handle the inherent noise of single-cell data –mainly amplification error and allele dropout (ADO) [10]. Understanding how somatic cells evolve is one of the main applications of single-cell technologies. In particular, the reconstruction of cell phylogenies from single-cell DNA sequencing (scDNA-seq) can help us understand the mode and tempo of cell diversification and the underlying mechanisms. Indeed, if geographical information is also available, the cell genealogy can inform us about cellular expansions and migrations, of utmost relevance for example in cancer.

In the recent past, several methods have been proposed to reconstruct phylogenetic trees from scDNA-seq data [11,12]. OncoNem [13] implements a nested-effects likelihood model to correct for the observational noise, plus a simple heuristic search that attempts to maximize the likelihood of the data across tree space. It reconstructs a tree of clones and mutations by assigning cells to the clones. OncoNem assumes an infinite-site mutation (ISM) model, and as we will see below, it is very slow and can only be used with very small datasets. SCITE [14] implements an ISM similar to the one used by OncoNem, but uses Markov Chain Monte Carlo (MCMC) to sample likelihoods/posterior probabilities. It essentially estimates a mutation tree to which it attaches the cells a posteriori. Conveniently, it can infer false positive (FP) and false negative (FN) error rates from the data. infSCITE [15] extends SCITE in order to consider cell doublets, to test the validity of the ISM and to learn the FP rate from panel sequencing data. SiFit [16] implements a Markov finite-site model of evolution and a heuristic ML tree search algorithm. It is also able to estimate FP and FN error rates, and in the reported simulations it outperforms OncoNem and SCITE in terms of speed and accuracy.

All the methods mentioned above were implemented *de novo* for the specific problem of single-cell phylogenetic reconstruction. We reasoned that organismal phylogenetics is a well-developed field with a number of extremely popular software implementations, with hundreds of thousands of users, whose performance has been streamlined for many years, and that could be leveraged to obtain more accurate somatic cell phylogenies. Therefore, we implemented a specific model for single-cell genotypes, CellPhy, into an existing, successful framework for statistical phylogenetics, RAxML-NG [17]. Our computer simulations and the analysis of several empirical datasets show that CellPhy is able to reconstruct more accurate single-cell phylogenetic trees across different biological scenarios and is substantially faster than the competing likelihood-based methods. We also evaluated a maximum parsimony-based tool TNT [18], a very simple and therefore extremely fast approach for phylogenetic reconstruction. Although TNT performed surprisingly well in error-free simulation scenarios, its accuracy quickly degraded in the presence of the typical biases observed in scDNA-seq data.

# Results

## CellPhy

We developed a probabilistic model for the phylogenetic analysis of single-cell diploid genotypes inferred from sc-DNA-seq experiments, called CellPhy. For tractability, we focused on single nucleotide variants (SNVs), which is arguably the most common type of genetic data obtained from somatic tissues nowadays. Current models of evolution for single-cells only consider the absence/presence of mutations regardless of the nucleotides involved [13,14,16]. This means that they deal with at most a ternary state space where genotypes can only have 0, 1, or 2 mutant alleles (normal, heterozygous, or homozygous mutant, respectively). Instead, here we model the changes among all possible 10 unphased genotypes $\Gamma$ = {AA, AC, AG, AT, CC, CG, CT, GG, GT, and TT}. Because somatic evolution proceeds mostly by mitosis, where both daughter cells receive the same set of chromosomes, and recombination can be safely ignored, there is a unique cell history that is recorded in exactly the same

way in the maternal and paternal chromosomes. This allows CellPhy to use unphased genotypes to infer the cell phylogeny, that is, we do not need to know whether a mutation occurred in the maternal or in the paternal chromosome. Indeed, phased genotypes would convey additional information for phylogenetic inference, but the vast majority of empirical datasets available to date are unphased due to technical limitations. Hence, in CellPhy we essentially extended the well-established finite-site continuous-time Markov model of DNA sequence evolution with four states [19] to 10 states. We assume that all SNVs evolve in the same way and independently of each other. In particular, we adopt the genotype equivalent of the classical general time-reversible (GTR) model of nucleotide substitution [20].

Importantly, single-cell genotypes can be very noisy mainly due to biases during whole-genome amplification (WGA) [10]. While previous methods rely on error models based on FP and FN rate parameters, we built an error model with two free parameters that are the ADO rate ($\delta$) and the amplification/sequencing error rate ($\varepsilon$). In comparison, the advantage of this parameterization is that it can incorporate plausible situations such as an amplification/sequencing error converting a homozygous mutant into a heterozygous genotype. Due to its low probability of occurrence, we discard the possibility of observing more than one amplification/sequencing error at a given site, but we allow for the presence of both, ADO and amplification/sequencing error, in a single genotype. In addition, instead of using a genotype error model, CellPhy can directly incorporate the phred-scaled genotype likelihoods provided by a single-cell variant caller.

We assume that the evolutionary history of the sample cells can be appropriately represented as an unrooted binary tree. Given the single-cell SNV genotypes (as a matrix in FASTA or PHYLIP format, or in a standard VCF file), and the genotype evolution and error model, CellPhy computes the likelihood of any given tree, as a product of the independent probabilities across SNVs, using the standard Felsenstein pruning algorithm [19,21]. Conveniently, CellPhy does not need to assume any particular genotypic configuration at the root, like other programs. For example, SiFit assumes that the root of the tree is homozygous for the reference allele at all sites. Instead, the CellPhy tree can be easily rooted *a posteriori* using a particular set of cells as an outgroup [see 19]. For example, if we are studying tumor cells, the outgroup could be one or more healthy cells.

We implemented CellPhy's phylogenetic model in RAxML-NG [17], a very popular maximum likelihood (ML) framework in organismal phylogenetics. Therefore, to obtain ML estimates of the model parameters (substitution rates, ADO, and amplification/sequencing errors), and of the cell tree, CellPhy leverages the optimization routines and tree search strategy of RAxML [22] and RAxML-NG [17]. The latter, for example, is known to work particularly well on large trees [23]. The fact that CellPhy exploits RAxML-NG allows it also to calculate confidence values for the tree branches using either the standard [24] or transfer [25] bootstrap (BS) techniques. Moreover, CellPhy can perform ancestral state reconstruction [26] to obtain ancestral ML genotypes and therefore is able to map mutations to the branches of the ML tree. CellPhy is freely available, together with documentation, tutorials, and example data at https://github.com/amkozlov/cellphy.

# Validation and benchmarking

### Simulation 1: infinite-site model, low number of SNVs ("target-ISM")

For simulated datasets of 40 cells and 250-1000 SNVs under an infinite-site mutation model, phylogenetic accuracy decreased rapidly for all methods with increasing levels of genotype error and/or ADO (Figures 1, S1-S2). The genotype coding strategy ("keep", "remove", and "missing") influenced accuracy only when genotype errors were present, with "missing" and "keep" being clearly better than "remove". CellPhy (which by default uses the error model EP17; see Methods) was, overall, the most accurate method, although closely followed by SiFit and infSCITE. Nonetheless, infSCITE performed worse when genotype errors were common. The parsimony-based inference tool TNT was as accurate as CellPhy, SiFit, or infSCITE in the absence of ADO and genotype error, but clearly worse otherwise. OncoNEM, which produced highly unresolved trees (i.e., with 50% to 90% polytomies), performed badly under all scenarios. Because OncoNEM is also extremely slow and cannot handle large data sets (see Computational speed section below), we did not evaluate it further.

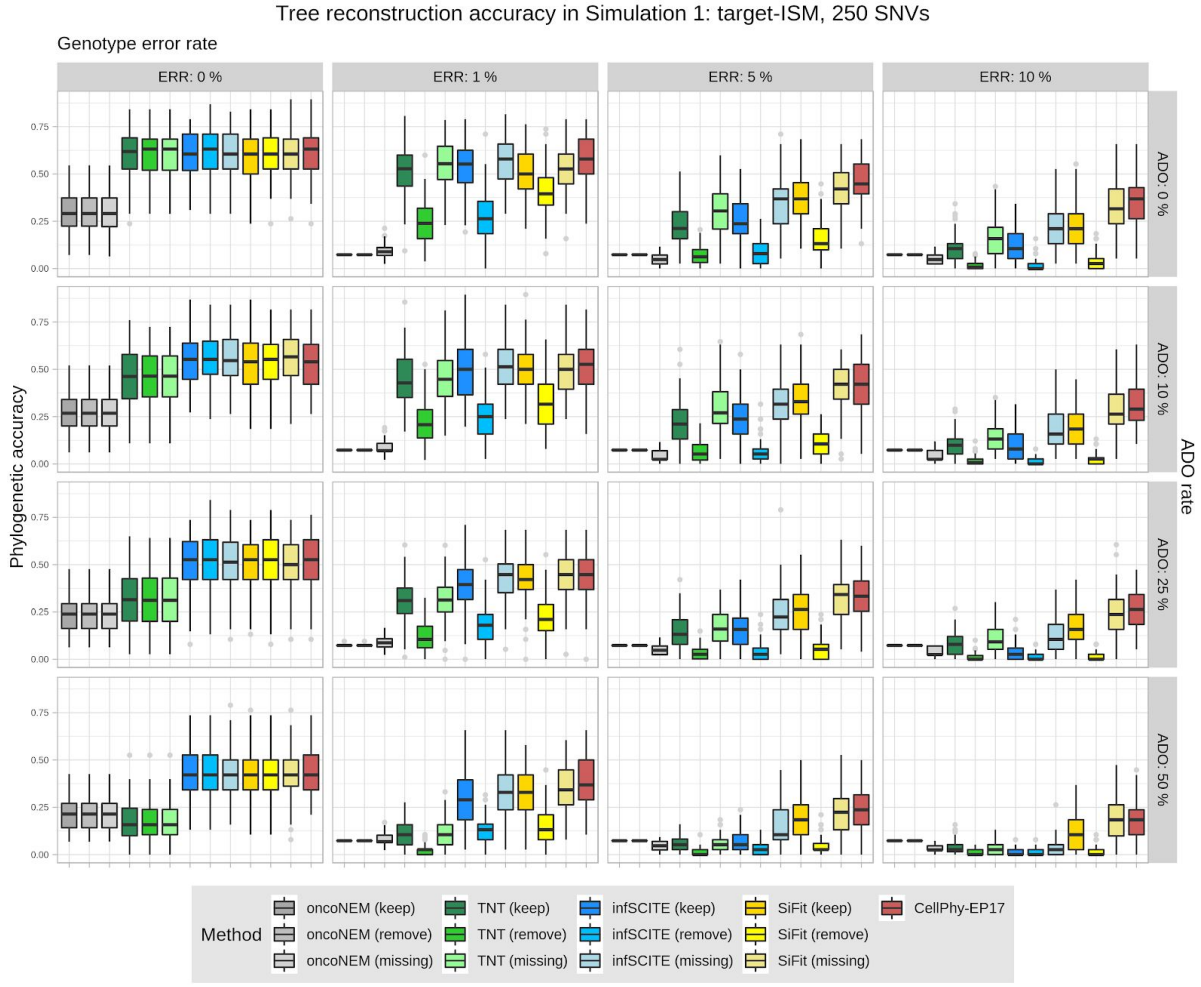


**Figure 1. Phylogenetic accuracy in Simulation 1 ("target-ISM") with 250 SNVs**. Datasets consisted of 40 cells. Accuracy was evaluated under different levels of genotype error (ERR), allele dropout (ADO), and genotype recoding strategies ("keep", "remove", "missing") as explained in the main text. Phylogenetic accuracy is defined as 1 – nRF (see Methods).

## Simulation 2: finite-site model, large number of SNVs ("WGS-FSM")

When we simulated larger data sets (100 cells, ~2,000 SNVs), in this case under a finite-site model of DNA evolution, overall phylogenetic accuracy increased for all methods. As before, all methods performed worse with higher levels of ADO and/or genotype error (Figure 2). CellPhy was again the most accurate method overall, in particular when the data contained a large number of genotype errors and ADO events. Again, the "missing" coding strategy was slightly superior to "keep", particularly with higher genotype error rates, and substantially better than "remove". Thus, for subsequent simulations, we only considered the "missing" strategy.

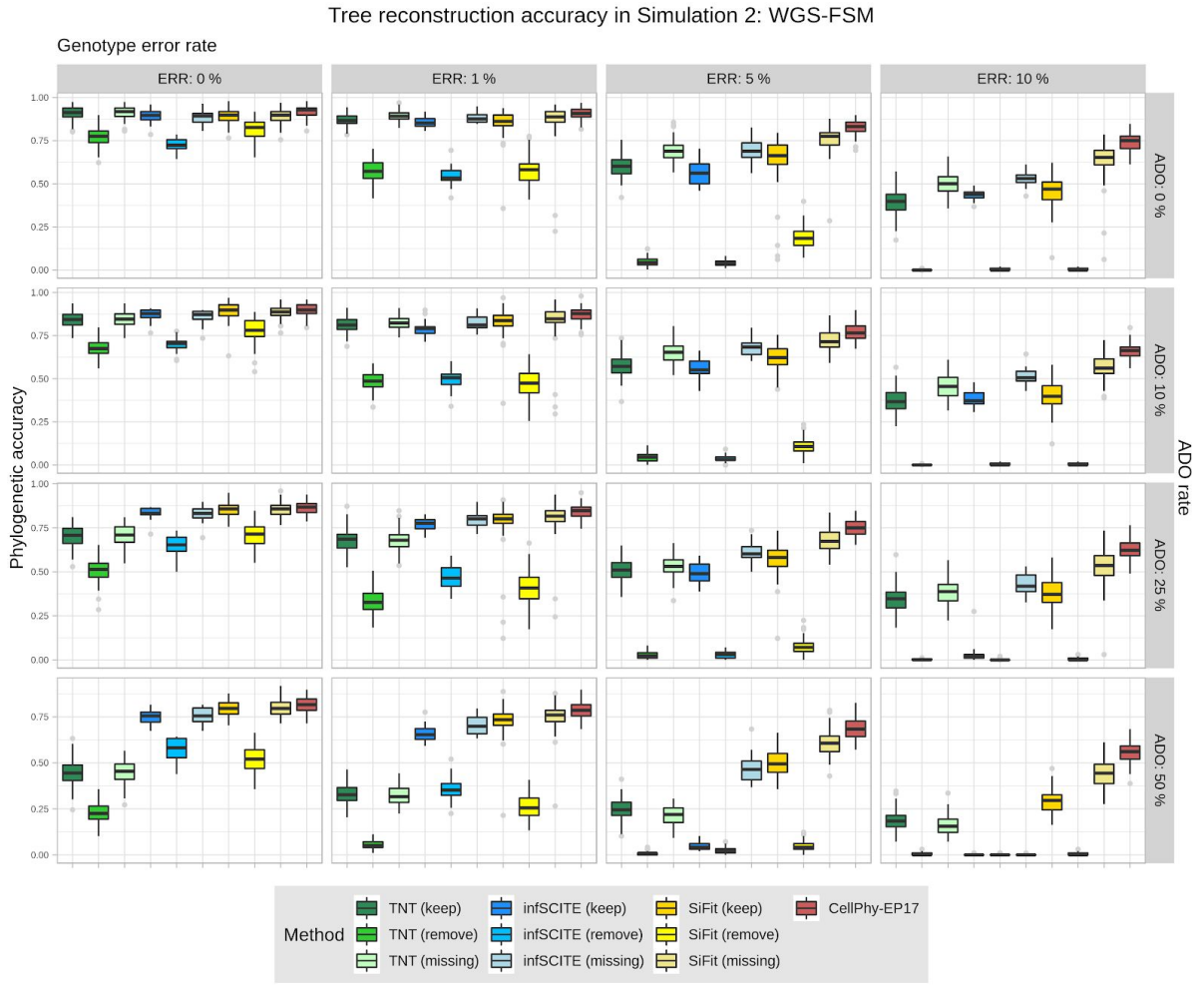


**Figure 2. Phylogenetic reconstruction accuracy in Simulation 2 ("WGS-FSM").** Datasets consisted of 100 cells and ~2000 SNVs. Accuracy was evaluated under different levels of genotype error (ERR), allele dropout (ADO), and genotype coding strategies ("keep", "remove", "missing"). Phylogenetic accuracy is defined as 1 – nRF (see Methods).

## Simulation 3: mutational signatures, large number of SNVs ("WGS-sig")

Here we considered relatively large datasets (60 cells, 1,000-4,000 SNVs), which were simulated to evolve under COSMIC trinucleotide mutational signatures 1 and 5. The trends were as before, and CellPhy consistently outperformed the competing methods, especially with increasing levels of genotype error and/or ADO (Figures S3-S4).

## Simulation 4: genotype likelihoods from NGS read counts ("NGS-like")

When we simulated read counts and the input data consisted of the inferred genotypes, the advantages of CellPhy became even more evident, in particular under its genotype likelihood model ("CellPhy-GL") and under a more realistic sequencing depth for single-cells (5x) (Figure 3). While at 30x and 100x the accuracy differences were substantially smaller, CellPhy still performed as well as or better than the competing methods (Figures S5-S6).

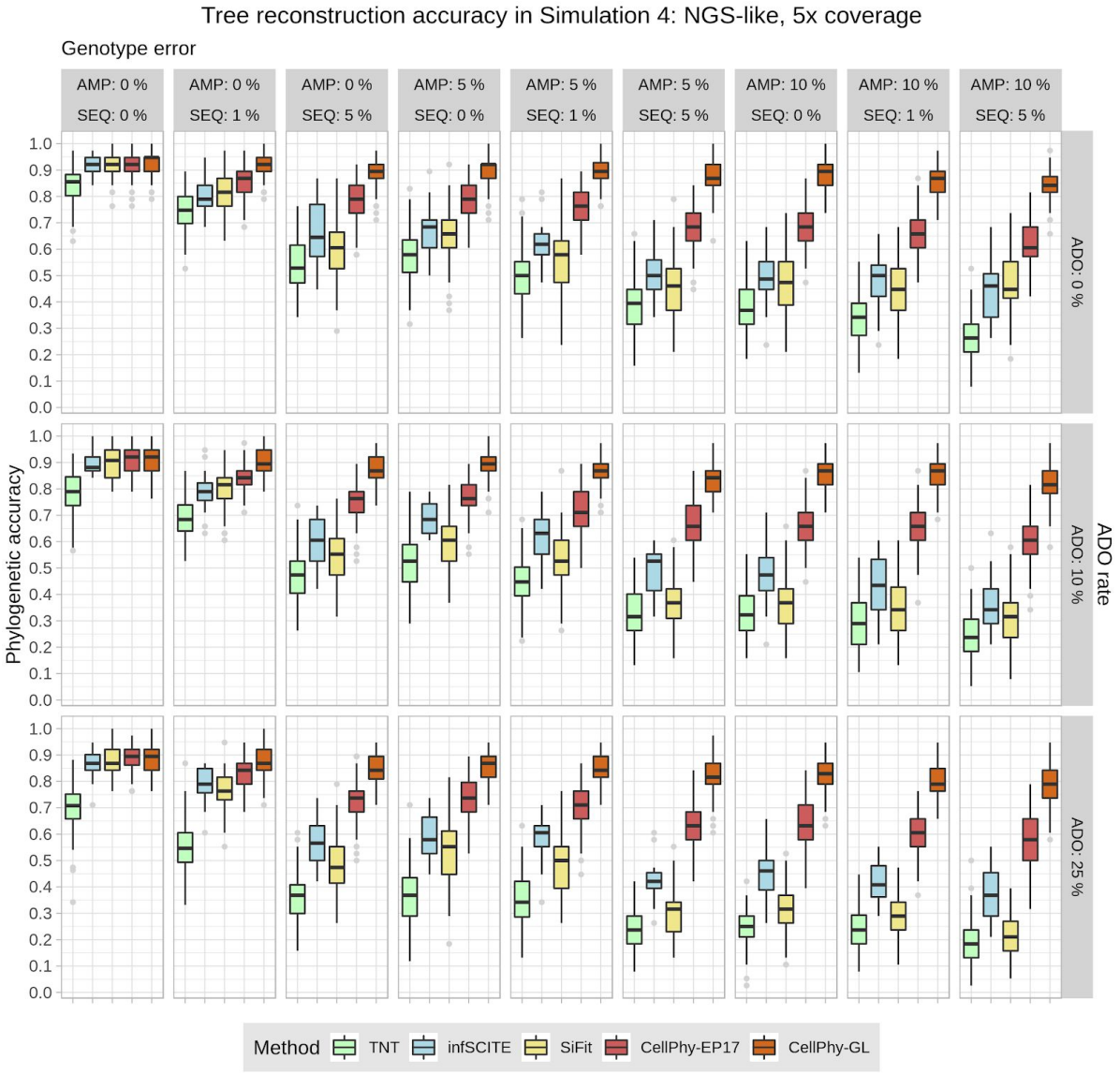


**Figure 3. Phylogenetic accuracy in Simulation 4 ("NGS-like") at 5x.** Datasets consisted of 40 cells and a ~1000-2000 SNVs. All methods use the ML genotypes except CellPhy-GL, which uses the genotype likelihoods. Phylogenetic accuracy is defined as 1 – nRF (see Methods). AMP is the amplification error rate, SEQ is the sequencing error rate, and ADO is the allele dropout rate. CellPhy was run using the genotype error model (CellPhy-EP17) and using the genotype likelihoods (CellPhy-GL).

## Simulation 5: NGS doublets

In many single-cell experiments, depending on the isolation method used, doublet cells can be relatively common [5], so we also assessed their effect. As expected, the presence of cell doublets reduced phylogenetic accuracy, but CellPhy was consistently the best method, particularly in the presence of ADO (Figure S7).

## Simulation 6: NGS for very large numbers of cells and SNVs

We also assessed phylogenetic accuracy on very large scDNA-seq datasets, with up to 1,000 cells and 50,000 SNVs, without doublets. Here we only evaluated TNT, SiFit, and CellPhy, as the infSCITE jobs were still running after one month. Here, CellPhy clearly outperformed the competing methods, with rapidly increasing accuracy as a function of the number of SNVs, benefiting further from the use of genotype likelihoods (Figure 4).

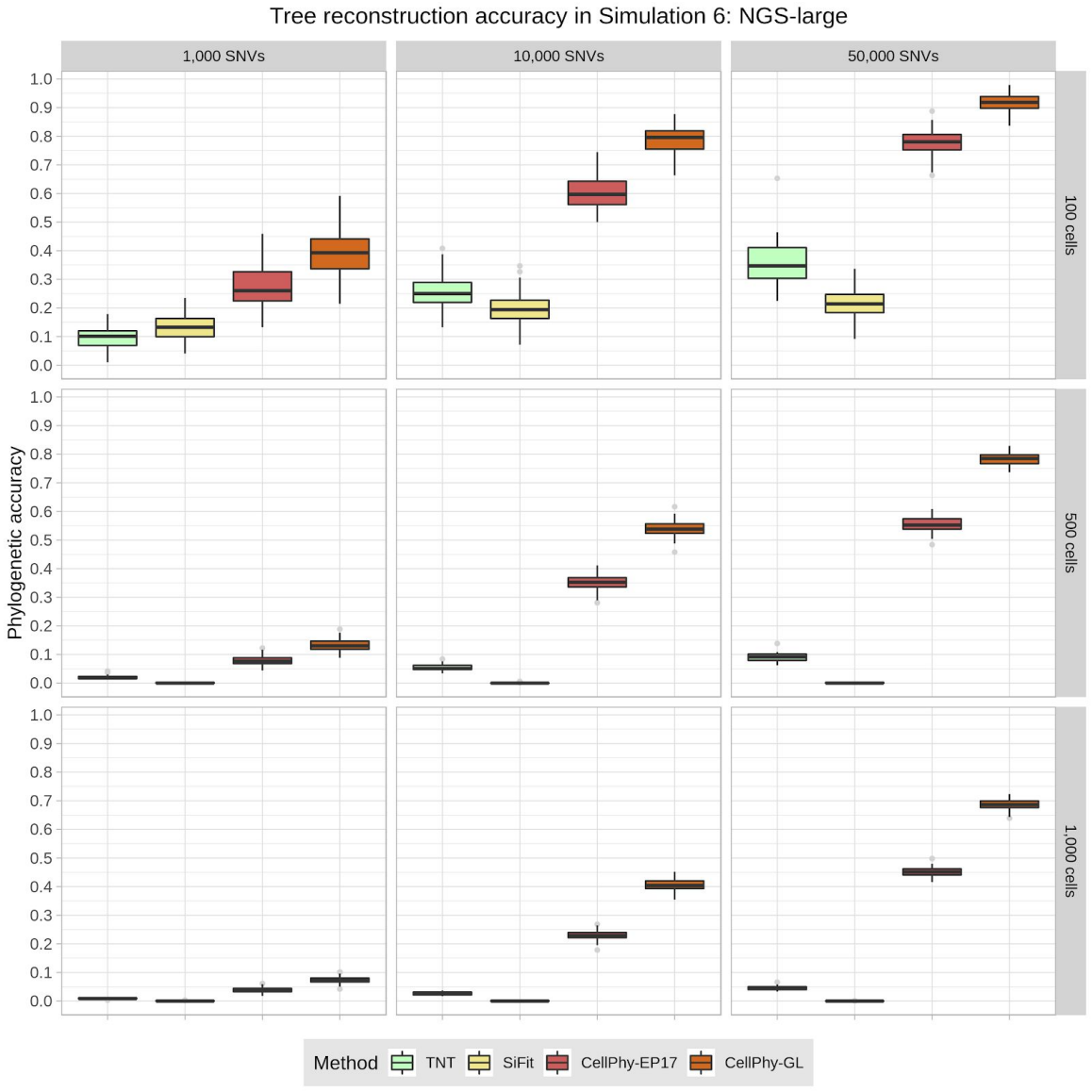


**Figure 4**. **Phylogenetic reconstruction accuracy in Simulation 6 ("NGS-large").** Mutations were introduced according to signature S1, and the sequencing depth was 5x. Read counts were simulated with 5% amplification error, 1% sequencing error, and 10% ADO. All methods use the inferred genotypes except CellPhy-GL, which uses the genotype likelihoods. Phylogenetic accuracy is defined as 1 – nRF (see Methods). For more than 1000 SNVs or more than 100 cells, only 10 replicates were run for SiFit for reasonable running times. CellPhy was run using the genotype error model (CellPhy-EP17) and using the genotype likelihoods (CellPhy-GL)

## Estimation of genotype error and ADO rates

Apart from inferring the ML tree, CellPhy is able to obtain ML estimates of the genotyping error and the ADO rate of sc-DNA-seq datasets. Across the different simulation scenarios described above, CellPhy estimated the genotyping error quite accurately (MSE: 0.00003 - 0.003), with a slight over- or underestimation when its true value was below or above 5%, respectively (Figure S8A). The ML estimates of the ADO rate were more variable and tended to underestimate the true value (MSE = 0.001 - 0.02), but were still generally accurate, particularly at higher rates (Figure S8B). As expected, in both cases, improved estimates were obtained with larger datasets comprising more SNVs.

## Computational speed

We compared the speed of the different methods by recording the running times for six simulated and two empirical data sets (Figure 5). Clearly, TNT was the fastest method by at least two orders of magnitude, which is not surprising as parsimony is much more affordable to compute than probabilistic models of evolution. After TNT, CellPhy (under both EP17 and GL models) was the second fastest method, being one to two orders of magnitude faster than SiFit, infSCITE, or OncoNem. For some of the largest data sets, including both simulated and empirical data, infSCITE and OncoNem did not finish after several days.

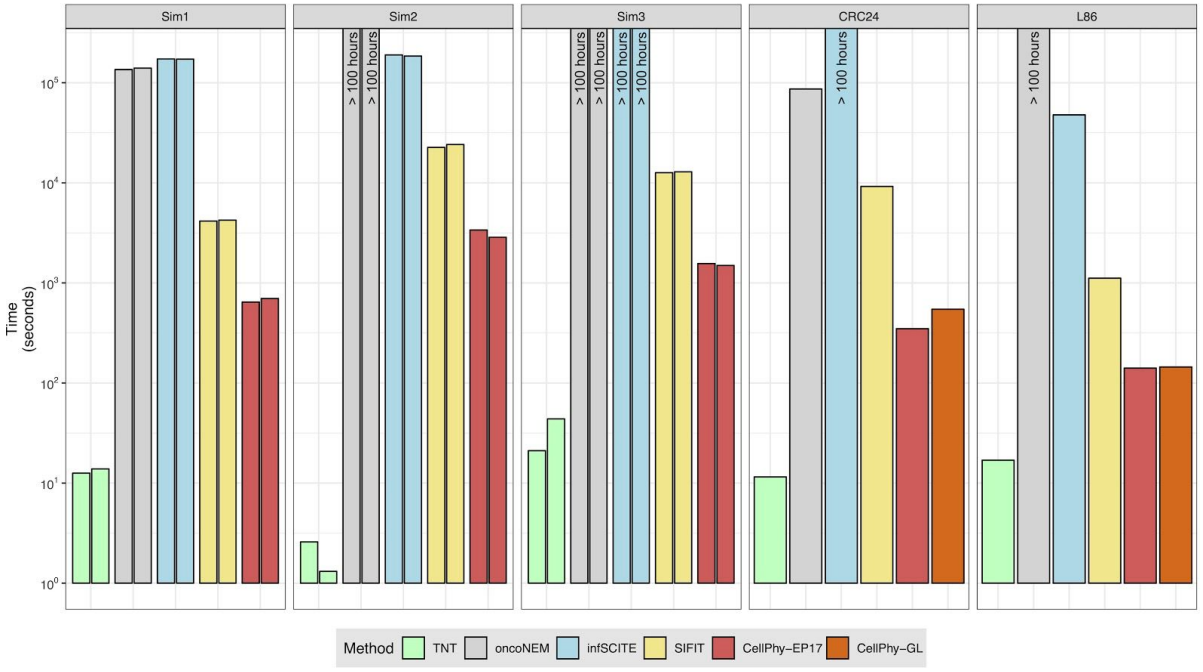


**Figure 5. Speed comparisons for simulated and real datasets.** "Sim1" corresponds to two simulated single-cell datasets with 40 cells and 4753 and 4761 SNVs. "Sim2" corresponds to two simulated datasets with 100 cells and 9935 and 9942 SNVs. "Sim3" corresponds to two simulated datasets with 60 cells and 9982 and 9985 SNVs. CRC24 and L86 correspond to two empirical datasets (see Methods). Note the logarithmic time scale on the y-axis.

# Application to single-cell data

## Phylogenetic reconstruction of a colorectal cancer

We analyzed a single-cell WGS dataset (CRC24) produced in our lab, consisting of 24 cells collected from two regions of a primary tumor in a patient with colorectal cancer (CRC). After filtering out germline polymorphisms, SNVs in non-diploid regions, and low-quality variants (see *Methods*), we identified a total of 18,099 SNVs. Some of the SNVs occurred in established CRC driver genes, such as *APC*, *BRAF*, *BRCA2*, *LRP1B*, and *MAP2K4*, among others. In the ML tree estimated by CellPhy, using the genotype likelihoods ("CellPhy-GL" model), (Figure 6A), cells tended to group according to their geographical location and phenotype, although not in a perfect fashion. Some of these ancestral relationships are well supported by the data, as reflected by several high bootstrap values, but others are not. This illustrates one of the important features of CellPhy: it is able to provide phylogenetic confidence measurements for different parts of the tree. Interestingly, non-stem cells had in general longer branch lengths than stem cells, suggesting potential differences in the evolutionary rate between these two cell types. We mapped the non-synonymous mutations onto the internal branches of the tree using a custom script (see *Methods*) and found that the vast majority are shared by all tumor cells sampled (i.e., clonal mutations), which includes variants affecting genes previously associated with CRC progression (e.g., *INPP1*, *CDC5L*, *ROR2*, *EXOSC5*) [27–30]. The tree topologies inferred by SiFit, infSCITE, and TNT (Figure S9A-C) were clearly distinct from the topology inferred by CellPhy (nRF=0.41, 0.68 and 0.86, respectively), but with a similar, albeit not identical, overall pattern regarding geography and stemness. Unfortunately, for the SiFit and infSCITE trees, the absence of branch support measures makes it difficult to properly evaluate which regions of the estimated trees can be trusted.

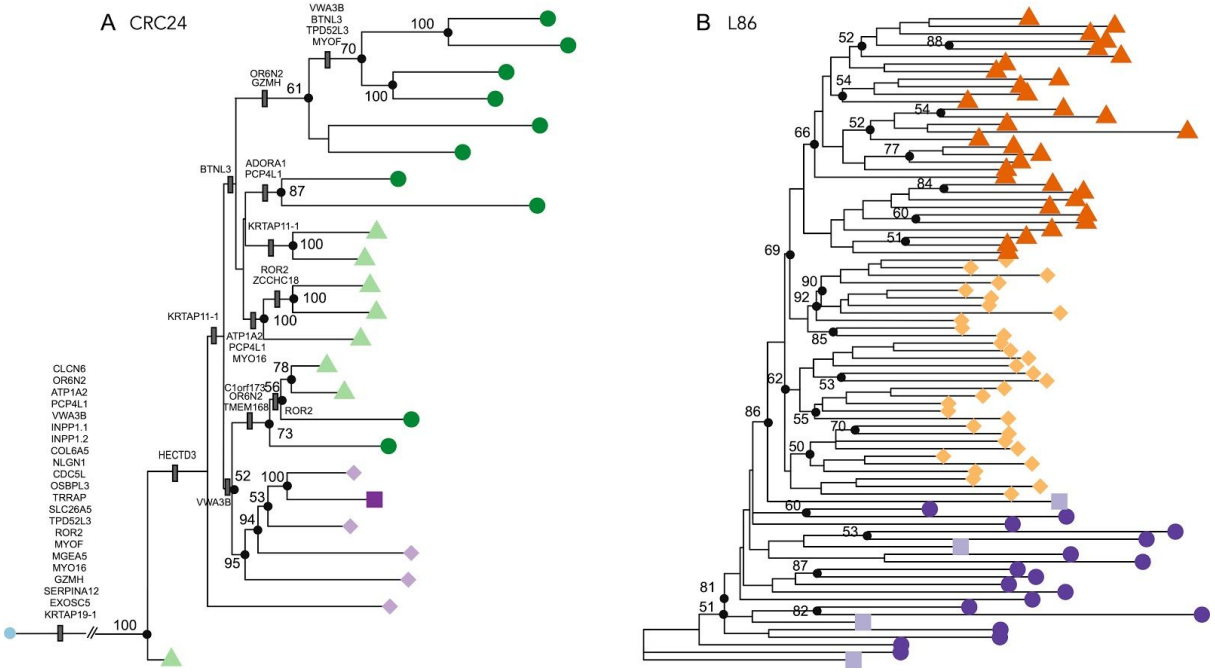


**Figure 6. CellPhy tree for the CRC24 and L86 datasets. (A)** "CellPhy-GL" CRC24 tree. Distinct shapes and colors represent cell type: healthy (blue circle); tumor-non-stem from TI region (dark green circle), tumor-stem from TI region (light green triangle), tumor-non-stem from TM region (dark purple square), tumor-stem from TM region (light purple diamond). Only bootstrap support values above 50 are shown. Non-synonymous mutations displayed on internal branches. **(B)** "CellPhy-GL" L86 tree. Distinct shapes and colors represent cell type: healthy diploid cells - from both primary and metastatic sites - (dark purple circle), healthy diploid cells missorted (light purple square), primary tumor aneuploid cells (light orange diamond), metastatic aneuploid cells (dark orange triangle). Only bootstrap values >= 50 are shown.

## Revisiting the evolutionary history of a metastatic colorectal cancer

We also explored a published dataset from a metastatic colorectal cancer patient [31]. In the original study, after performing custom targeted sequencing of 186 single-cells sampled from primary and metastatic lesions, the authors derived a cell tree using SCITE and inferred a polyclonal seeding of liver metastases (i.e., distinct populations of tumor cells migrated from the primary tumor towards the liver). However, their findings have been recently re-evaluated in two different studies [32,33]. In particular, the former applied the newly developed SiCloneFit, which relaxes the infinite-sites assumption, to the original SNV dataset and also proposed a polyclonal seeding of the metastases, while the latter performed a joint analysis between SNVs and copy-number variants (CNVs) with SCARLET to suggest that the liver metastasis was instead seeded by a single clone. As the focus of this analysis is on cancer history, to speed up computation we removed the majority of the healthy cells in the original dataset, ending up with 86 cells (L86 data set). Using SC-Caller [34], we identified 597 SNVs distributed over the portion of the genome not affected by CNVs, which included most variants detected in the original study (e.g., *APC*, *NRAS*, *MYH11*, *LINGO2*, *IL7R*, *F8*, *FUS*). In the CellPhy tree, all metastatic cells clustered together with high support, indicative of a monoclonal origin of the liver metastasis (Figure 6B). Also, we can see here two distinct, well-supported metastatic cell populations. Interestingly, while some primary (PA-54 and PA-27) and metastatic aneuploid (MA-25 and MA-26) cells were intermixed with healthy diploid cells, these correspond to cells that were mislabelled during FACS sorting, as previously noted by the authors. Furthermore, after mapping the non-synonymous mutations onto the internal branches of the CellPhy tree (Figure S10), we found that all cancer cells harbor somatic variants affecting genes that have been shown to contribute to human intestinal neoplasia (i.e., *MYH11* and *STAG1*) [35,36]. Notably, none of the competing methods was able to recover a single metastatic clade (Figure S11). Although in the SiFit tree most metastatic cells cluster together, some of them appear intermixed with the primary tumor cells, while in the infSCITE and TNT trees the tumor cells did not form a clade, a result that does not seem very realistic.

## Applicability of CellPhy to non-cancer data and to bulk clonal sequences

Finally, we used CellPhy to analyze two non-cancer WGS single-cell datasets. The first of these datasets (E15) consists of 242 somatic SNVs from 15 single neurons from a normal individual (Figure S12A) [37]. In the CellPhy tree built using genotype likelihoods (Figure S12A), different lineages seem to have very distinct evolutionary rates. However, most branches had very low bootstrap support (<50%), suggesting that more SNVs are required before reliable interpretations can be made. All the other methods recovered very distinct trees (Figure S12B-D), and in the case of TNT (infSCITE and SiFit do not assess branch support) also with very low bootstrap values.

The second dataset (LS140) consists of 140 single cell-derived human haematopoietic stem and progenitor colonies from a healthy individual [38]. In this case, because there is no amplification involved, we expect a very small genotype error and no ADO. Remarkably, CellPhy ML estimates for this data set were zero for both error and ADO parameters. The CellPhy tree shows in general very high bootstrap values (Figure S13), highlighting the quality of this dataset, which has a strong phylogenetic signal. This was somehow expected given that it includes 127,884 SNVs and lacks single-cell biases. Moreover, our analysis confirms an early, well-supported, partition of the cell colonies into two distinct groups, together with a lack of geographical structure, reinforcing the idea of a continuous redistribution of stem cell pools at the whole-body level.

# Discussion

We have developed CellPhy, a phylogenetic tool for the analysis of single-cell SNV data that is inspired by existing models and methods in statistical organismal phylogenetics. Unlike some of its competitors (OncoNEM, infSCITE), CellPhy does assume an infinite-site model of evolution. Furthermore, its evolutionary model explicitly considers the 10 possible unphased DNA genotypes and can account for their uncertainty by using genotype likelihoods as input. If uncertainty information is not available, CellPhy can estimate single-cell error and ADO rates from the genotype data matrix. Finally, CellPhy can reconstruct ancestral states, predict mutations on tree branches, and provide a statistical measure of branch support. To benchmark CellPhy, we conducted a series of computer simulations under different scenarios with varying degrees of complexity. Unlike in previous comparisons [13–16] we consider more realistic somatic genealogies by sampling sets of cells from a growing population. This results in more difficult trees, with shorter internal and longer terminal branches, that require hundreds of SNVs to be accurately estimated. Overall, CellPhy was the most accurate method, both under infinite- and finite-site mutation models, under different mutational patterns (e.g., using COSMIC trinucleotide mutational signatures), and in particular with higher levels of noise (genotype errors and ADO) in the data. Our simulations also suggest that accounting for SNV calling uncertainty is important when sequencing depth is low to moderate, which normally is the case for single-cell WGS due to the sequencing costs. With a sequencing depth of 5x, the consideration of genotype uncertainty makes CellPhy much more accurate than its competitors. Importantly, the accuracy of CellPhy does not come at the cost of speed. CellPhy is one to two orders of magnitude faster than SiFit, infSCITE, or OncoNEM. Although, as expected, parsimony-based TNT was by far the fastest method in our evaluation, this comes at the price of a largely reduced accuracy under most scenarios.

The analysis of real data suggests that CellPhy is able to provide biological insights where other methods fail to do so, with a seemingly more reliable grouping of cells according to phenotype or location. For example, in our re-analysis of the dataset from Leung *et al.* [31] CellPhy was able to recover a monoclonal metastasis – in line with a recent analysis by Satas *et al.* [33] that accounts for copy number variants–, while the competing methods implied a polyclonal seeding. Importantly, the CellPhy bootstrap analyses illustrate the importance of explicitly considering phylogenetic uncertainty, as in the absence of it one cannot really assess if different parts of a tree are equally supported by the data. Also, the analysis of cell colonies shows that CellPhy can also be used to estimate trees from clonal sequences that do not necessarily correspond to amplified single-cells.

Overall, our results suggest that CellPhy could become the method of choice for the estimation of phylogenetic trees from WGS single-cell SNV data. It is important to note that, in order to achieve high phylogenetic accuracy under realistic conditions (i.e., growing populations with long terminal branches), CellPhy requires a reasonably high number of SNVs (hundreds to tens of thousands, depending on the number of cells). However, as can be seen from our benchmark results, this limitation is common to all methods. It reflects a fundamental problem of poor signal-to-noise ratio: the faster the cell population grows, the more cells we have and the higher the single-cell error and ADO rates are, the more SNVs are needed to recover the true cell relationships. We expect that further improvements in single-cell sequencing accuracy, variant calling sensitivity, and genotype phasing will yield datasets that are better suited for phylogenetic inference. Furthermore, some common simplifying assumptions of the classical phylogenetic models (reversibility, stationarity, context-independence) could be more problematic in the context of somatic evolution. Eliminating those assumptions, albeit methodologically and computationally challenging, could potentially improve accuracy on datasets with scarce phylogenetic signal.

# Methods

## The CellPhy model

### Model of genotype evolution

We built a finite-site continuous-time Markov model of DNA sequence evolution with 10 states, in which all SNVs are independent and identically distributed (i.i.d.). This extended general time-reversible model of genotype substitution (GTR-like; [see 20] is defined by the instantaneous rate matrix Q, which contains the instantaneous transition rates $q_{i \leftrightarrow j}$ among genotypes *i* and *j*. For computational convenience, we assume a time-reversible process, in which case the Q matrix can be represented as a product of a symmetric exchangeability matrix R ($r_{i \leftrightarrow j} = r_{j \leftrightarrow i}$) and a diagonal matrix of stationary genotype frequencies $\pi$:

$$Q = R \cdot diag(\pi_i),\ i \in \Gamma \quad (1)$$

Because maternal and paternal chromosomes should evolve independently in somatic cells, we can assume that the instantaneous transition rate of nucleotide *x* to nucleotide *y* does not depend on the identity of the homologous allele (*n*). In other words, we assume $r_{nx \leftrightarrow ny} = r_{x \leftrightarrow y}$. Conveniently, these two assumptions significantly reduce the number of free parameters in the Q matrix: we only need to estimate five exchangeabilities ($r_{A \leftrightarrow C}$, $r_{A \leftrightarrow G}$, $r_{A \leftrightarrow T}$, $r_{C \leftrightarrow G}$, $r_{C \leftrightarrow T}$; *let* $r_{G \leftrightarrow T} = 1$) and nine stationary genotype frequencies ($\pi_{AA}$, $\pi_{AC}$, $\pi_{AG}$, $\pi_{AT}$, $\pi_{CC}$, $\pi_{CG}$, $\pi_{CT}$, $\pi_{GG}$, $\pi_{GT}$; $\pi_{TT} = 1 - \sum \pi_{ij}$).

$$Q = \begin{array}{c} \\ AA \\ AC \\ AG \\ AT \\ CC \\ CG \\ CT \\ GG \\ GT \\ TT \end{array} \begin{array}{c} \begin{array}{cccccccccc} AA & AC & AG & AT & CC & CG & CT & GG & GT & TT \end{array} \\ \begin{pmatrix} -q_{AA} & r_{A\leftrightarrow C}\pi_{AC} & r_{A\leftrightarrow G}\pi_{AG} & r_{A\leftrightarrow T}\pi_{AT} & 0 & 0 & 0 & 0 & 0 & 0 \\ r_{A\leftrightarrow C}\pi_{AA} & -q_{AC} & r_{C\leftrightarrow G}\pi_{AG} & r_{C\leftrightarrow T}\pi_{CT} & r_{A\leftrightarrow C}\pi_{CC} & r_{A\leftrightarrow G}\pi_{CG} & r_{A\leftrightarrow T}\pi_{CT} & 0 & 0 & 0 \\ r_{A\leftrightarrow G}\pi_{AA} & r_{C\leftrightarrow G}\pi_{AC} & -q_{AG} & r_{G\leftrightarrow T}\pi_{CT} & 0 & r_{A\leftrightarrow C}\pi_{CG} & 0 & r_{A\leftrightarrow G}\pi_{GG} & r_{A\leftrightarrow T}\pi_{GT} & 0 \\ r_{A\leftrightarrow T}\pi_{AA} & r_{C\leftrightarrow T}\pi_{AC} & r_{G\leftrightarrow T}\pi_{AG} & -q_{AT} & 0 & 0 & r_{A\leftrightarrow C}\pi_{CT} & 0 & r_{A\leftrightarrow G}\pi_{GT} & r_{A\leftrightarrow T}\pi_{TT} \\ 0 & r_{A\leftrightarrow C}\pi_{AC} & 0 & 0 & -q_{CC} & r_{C\leftrightarrow G}\pi_{CG} & r_{C\leftrightarrow T}\pi_{CT} & 0 & 0 & 0 \\ 0 & r_{A\leftrightarrow G}\pi_{AC} & r_{A\leftrightarrow C}\pi_{AG} & 0 & r_{C\leftrightarrow G}\pi_{CC} & -q_{CG} & r_{G\leftrightarrow T}\pi_{CT} & r_{C\leftrightarrow G}\pi_{GG} & r_{C\leftrightarrow T}\pi_{GT} & 0 \\ 0 & r_{A\leftrightarrow T}\pi_{AC} & 0 & r_{A\leftrightarrow C}\pi_{AT} & r_{C\leftrightarrow T}\pi_{CC} & r_{G\leftrightarrow T}\pi_{CG} & -q_{CT} & 0 & r_{C\leftrightarrow G}\pi_{GT} & r_{C\leftrightarrow T}\pi_{TT} \\ 0 & 0 & r_{A\leftrightarrow G}\pi_{AG} & 0 & 0 & r_{C\leftrightarrow G}\pi_{CG} & 0 & -q_{GG} & r_{G\leftrightarrow T}\pi_{GT} & 0 \\ 0 & 0 & r_{A\leftrightarrow T}\pi_{AG} & r_{A\leftrightarrow G}\pi_{AT} & 0 & r_{C\leftrightarrow T}\pi_{CG} & r_{C\leftrightarrow G}\pi_{CT} & r_{G\leftrightarrow T}\pi_{GG} & -q_{GT} & r_{G\leftrightarrow T}\pi_{TT} \\ 0 & 0 & 0 & r_{A\leftrightarrow T}\pi_{AT} & 0 & 0 & r_{C\leftrightarrow T}\pi_{CT} & 0 & r_{G\leftrightarrow T}\pi_{GT} & -q_{TT} \end{pmatrix} \end{array} \quad (2)$$

The probabilities of changing from a given genotype to another along a branch of length *t* are then given by [19]:

$$P(t) = e^{Qt} \quad (3)$$

In the following, we will refer to this genotype evolution model as GTGTR4, where the "4" denotes the 4-state matrix that expands into the 10-state genotype space.

### Single-cell genotype error model

To incorporate errors in the observed genotypes, arising during whole-genome amplification (WGA) and sequencing, we consider two free parameters: the ADO rate ($\delta$) and the amplification/sequencing error rate ($\varepsilon$). More specifically, $\varepsilon$ is the probability that a genotype *x* will be observed as another genotype $y \neq x$ either due to amplification error or sequencing error. At the same time, $\delta$ is the probability that the amplification of one of the two alleles has failed, and thus we observe the homozygous genotype defined by the amplified allele. For simplicity, and given that they are rare events, we only consider a single amplification/sequencing error per site, but we allow for the presence of both an amplification/sequencing error and an ADO event in a particular genotype. Under these assumptions, we compute the likelihood of a true genotype *x* for the SNV *i* and cell *j*, given the observed genotype *y,* as follows:

$$L_i^j(x) = Pr(S_i^j = y \mid x, \varepsilon, \delta) = \begin{cases} 1 & \text{if } y = \text{'-'(missing)} \\ Pr_0(x) & \text{if } x = y \\ Pr_1(x,y) & \text{if } d(x,y) = 1 \\ \varepsilon\delta/6 & \text{if } d(x,y) = 2 \wedge y \in \mathcal{H} \\ 0 & \text{otherwise} \end{cases}$$

(4)

where $\mathscr{H}$ = {AA, CC, GG, TT} is the set of all homozygous genotypes, and *d(x, y)* is the number of nucleotide differences between genotypes *x* and *y*:

$$d(x,y) = d(x_1x_2, y_1y_2) = \begin{cases} 0 & \text{if } x_1 = y_1 \wedge x_2 = y_2 \\ 2 & \text{if } x_1 \neq y_1 \wedge x_2 \neq y_2 \\ 1 & \text{otherwise} \end{cases} \quad (5)$$

$Pr_0(x)$ is the probability of observing the true genotype *x*:

$$Pr_0(x) = \begin{cases} 1 - \varepsilon + \varepsilon\delta/2 & \text{if } y \in \mathcal{H} \\ 1 - \varepsilon - \delta + \varepsilon\delta & \text{otherwise} \end{cases} \quad (6)$$

and $Pr_1(x, y)$ is the probability of observing a genotype *y* with a *d(x, y)* of 1 from the true genotype *x*:

$$Pr_1(x,y) = \begin{cases} \varepsilon(1-\delta)/3 & \text{if } x \in \mathcal{H} \\ \delta/2 + \varepsilon/6 - \varepsilon\delta/3 & \text{if } x \notin \mathcal{H} \wedge y \in \mathcal{H} \\ \varepsilon(1-\delta)/6 & \text{otherwise} \end{cases} \quad (7)$$

For missing data, we consider all possible genotypes to be equally likely, which is a standard assumption in likelihood-based phylogenetic inference. In the following, we will refer to this genotype error model as EP17.

## Phylogenetic likelihood

We consider the evolutionary history of cells as an unrooted binary tree $T = (\tau, t)$, where $\tau$ is a tree topology and *t* is a vector of branch lengths. A branch length represents the mean number of expected mutations per SNV. We define the phylogenetic likelihood of the tree *T* as the conditional probability of the observed SNV matrix *S* given the model *M* with parameters $\theta$ and the cell phylogeny by *T*:

$$L(T, M, \theta) = Pr(S \mid T, M, \theta) \quad (8)$$

We assume that genomic sites evolve independently under model *M*, and therefore the probability of observing the SNV matrix *S* is the product over the independent probabilities for all individual SNVs, $S_i$:

$$L(T, M, \theta) = \prod_{i=1}^{s} Pr(S_i \mid T, M, \theta) \quad (9)$$

where *s* is the total number of SNVs. For numerical reasons, that is, to avoid floating-point underflow, in practice we calculate the log-likelihood score:

$$log\, L(T, M, \theta) = \sum_{i=1}^{s} log\, Pr(S_i \mid T, M, \theta) \quad (10)$$

Let us place an additional imaginary node, called *virtual root*, into an arbitrary branch of the unrooted tree *T* at an arbitrary position along that branch. Without loss of generality, we can assume that this virtual root node has an index of 0, we can index the tip nodes from 1 to *c* by their respective cell numbers, and index the internal nodes from *c*+1 to 2*c*–2 (note that, an unrooted binary tree with *c* leaves has *c*–3 inner nodes, i.e., (2*c*–2) – (*c*+1) = *c*–3). Then, under the GTGTR4 model, per-SNV probabilities can be computed as follows:

$$Pr(S_i \mid T,\ GTGTR4,\ R, \pi) = \sum_{x \in \Gamma} L_i^0(x)\pi_x,\ i = 1...s \tag{11}$$

where $x$ is a genotype, $\pi_x$ is a stationary frequency of the genotype $x$, and $L_i^0$ is the vector of genotype likelihoods at the virtual root, which we compute recursively as follows:

$$L_i^v(x) = [\sum_{y \in \Gamma} P_{x,y}(t_u)L_i^u(y)] \cdot [\sum_{y \in \Gamma} P_{x,y}(t_w)L_i^w(y)],\quad \forall x \in \Gamma,\ v \notin [1...c] \tag{12}$$

where $y$ is another genotype, *u* and *w* are both children nodes of *v* in the direction from the virtual root (Figure S14) for the respective branch lengths. We initialize the genotype likelihood vectors at the tip nodes ($L_i^1$ .. $L_i^c$) depending on the input type (see Section "Input data" below) and error model. If the input is the genotype matrix *S*, and no error model is considered, these tip likelihood vectors are:

$$L_i^v(x) = \{1\ if\ S_i^v = x,\ 0\ otherwise\},\quad v \in [1...c],\ i \in [1...s] \tag{13}$$

while if we include the EP17 error model, the tip likelihoods are computed as in Eq. 4. Otherwise, if the input consists of the full genotype likelihoods (G10 field) the tip likelihoods are:

$$L_i^v(x) = 10^{G10_i^v(x)},\quad \forall x \in \Gamma,\ v \in [1...c],\ i \in [1...s] \tag{14}$$

Finally, if phred-scaled likelihoods for REF/REF, REF/ALT and ALT/ALT genotypes are provided (PL field in a VCF file), the calculation is as follows:

$$L_i^v(x) = L_i^v(x_1x_2) = \begin{cases} 10^{-0.1 \cdot PL_i^v(0)} & \text{if } x_1 = \text{REF} \wedge x_2 = \text{REF} \\ 10^{-0.1 \cdot PL_i^v(1)} & \text{if } (x_1 = \text{REF} \wedge x_2 = \text{ALT}) \vee (x_1 = \text{ALT} \wedge x_2 = \text{REF}) \\ 10^{-0.1 \cdot PL_i^v(2)} & \text{if } x_1 = \text{ALT} \wedge x_2 = \text{ALT} \\ 0 & \text{otherwise} \end{cases} \tag{15}$$

The final tree can be rooted using an outgroup [see 19]. For the sake of completeness, we included a simple explanation of standard phylogenetic likelihood calculations on DNA sequence alignments in the Supplementary Material (Figures S14-S16, Note 1).

# Implementation

## Overview

We implemented CellPhy as a pipeline which is based on a modified version of RAxML-NG [17]. In addition to the core tree search functionality of RAxML-NG, CellPhy pipeline offers features such as VCF conversion, mutation mapping and tree visualization using the *ggtree* package [39]. Furthermore, CellPhy provides reasonable defaults for most parameters (model of evolution, number of starting trees, etc.), which allows the user to run a "standard" CellPhy analysis by specifying just a single parameter, the input VCF file (or the genotype matrix). Alternatively, expert users can customize every aspect of CellPhy analysis to match their needs, as we show in the tutorial (https://github.com/amkozlov/cellphy/blob/master/doc/CellPhy-Tutorial.pdf). In the remainder of this section, we will also give implementation details on each individual step of the CellPhy pipeline. CellPhy code and documentation are freely available at: https://github.com/amkozlov/cellphy

## Input data

CellPhy accepts two types of input, a matrix of genotypes in FASTA or PHYLIP format, or a standard VCF file (https://samtools.github.io/hts-specs/VCFv4.3.pdf). When the input is a genotype matrix, genotypes are encoded as shown in Table S2. When the input is a VCF, CellPhy can be run in two distinct modes. The first mode ("CellPhy-EP17") requires a VCF with at least the GT field (that stores the genotype calls), in which case CellPhy simply extracts the genotype matrix. The second mode ("CellPhy-GL") requires a VCF with the PL field (which stores the phred-scaled genotype likelihoods) and uses instead the likelihoods of each genotype. Importantly, while commonly used variant callers for single-cell data (e.g., Monovar [40]) generate VCF files with a standard PL field, users should be aware that the PL definition may differ from its standard meaning in different callers. Indeed, SC-Caller [34], for instance, uses the PL field to not only store the likelihood of heterozygous and alternative homozygous genotypes, but also the likelihood of sequencing noise and amplification artifacts. On this basis, the PL field in VCF files stemming from SC-Caller needs to be converted to the standard PL format before CellPhy can be used (see Table S3 outlining CellPhy's compatibility with popular variant calling algorithms).

## Phylogenetic tree search

CellPhy uses the broadly used and well-tested heuristic tree search strategy of RAxML [22] and RAxML-NG [17]. CellPhy's search algorithm starts by default with 20 different trees, 10 obtained using maximum parsimony-based stepwise addition and 10 with a completely random topology. The ML tree search itself alternates between model parameter optimization and tree topology optimization phases. The key mechanism for searching tree topologies are so-called subtree pruning and regrafting (SPR) [41] moves that attempt to remove subtrees from the tree and subsequently place them into different branches to assess if the likelihood improves. Those SPR moves are applied iteratively until no SPR move can be found that further improves the likelihood of the tree. In this case CellPhy terminates and returns the best-found ML tree. For further details, please see [42] and references therein.

## Model parameter optimization

CellPhy uses the L-BFGS-B method [43] to optimize genotype substitution rates and equilibrium base frequencies. ADO rate and amplification/sequencing error rate are optimized independently using Brent's method [44]. After each iteration of Brent's algorithm, CellPhy re-computes all per-genotype likelihoods according to Eq. 4 using the new values of the ADO rate $\delta$`, and the amplification/sequencing error ε'.

## Branch support

CellPhy can compute confidence values for individual branches of the ML tree using two bootstrap (BS) techniques, the standard BS [24] and the transfer BS [25]. In the standard BS the first step consists of generating a number of BS replicates, typically 100 to 1,000, from the original dataset by sampling SNV sites with replacement. Then, an ML tree is estimated for each replicate. Finally, the support for each

branch in the ML tree is computed as the percentage of BS trees that contain that branch as outlined in the example provided in Figure S16. The standard BS only takes into account exact matches (i.e., the branch in the ML tree and in the BS trees must match exactly to be counted). In contrast, the transfer BS also takes into account inexact matches to account for the tree search uncertainty and vastness of the tree search space in phylogenetic analyses with many cells.

### Mapping mutations onto the tree

CellPhy can show predicted mutations on the branches of the inferred cell tree. To this end, it performs marginal ancestral state reconstruction [26] to obtain the ML genotype for every SNV at every inner node of the tree. At the tips of the tree, occupied by the observed cell genotypes, depending on the input, CellPhy applies Eq. 4 to compute genotype likelihoods given the observed genotype $y$ and estimated error rates ($\delta$, $\varepsilon$), or directly uses the genotype likelihoods provided in the VCF file. Then, it compares ML genotypes between two nodes connected by a branch, and if they differ, a mutation is predicted on the corresponding branch. Mutation mapping output consists of two files, a branch-labelled tree in the Newick format, and a text file with a list of predicted mutations (SNV names or positions) at each branch. We additionally provide a script (https://github.com/amkozlov/cellphy/blob/master/script/mutation-map.R) that automatically generates a plot with the mutations mapped onto the resulting phylogenetic tree, together with a tutorial that explains its use.

### Computational efficiency

RAxML-NG was developed with a particular focus on high performance and scalability to large datasets. Hence CellPhy capitalizes on numerous computational optimizations implemented therein, including highly efficient and vectorized likelihood calculation code, coarse- and fine-grained parallelization with multi-threading, checkpointing and fast transfer bootstrap computation [45].

## Benchmarking

We used computer simulations to benchmark the accuracy of CellPhy under different scenarios (Table S1), relative to the state-of-the-art methods for single-cell phylogenies OncoNem [13], infSCITE [14,15], and SiFit [16]. For these methods, the input is the matrix of observed reference/non-reference homozygous/heterozygous genotypes. In addition, we included the standard phylogenetic method TNT [18]. TNT implements a maximum parsimony (MP) approach and therefore attempts to find the tree/s that requires the least number of mutations to explain the data. TNT is very popular in MP organismal phylogenetics and has been heavily optimized for computational speed and efficiency. It is not designed for single-cell and/or NGS data and therefore assumes that the observed genotypes are error-free.

### Simulation of genealogies and genotypes

For the simulation of single-cell diploid genotypes, we used CellCoal [46]. This program can simulate the evolution of a set of cells sampled from a growing population, introducing single nucleotide variants along the coalescent genealogy under different models of DNA mutation. Furthermore, it can also introduce the typical errors of single-cell sequencing, specifically, ADO, amplification and sequencing errors, and doublets, either to the observed genotypes or directly into the read counts (Fig. S1).

We designed six distinct simulation scenarios (Simulation 1 - 6) representing different types of sc-DNA-seq datasets (Table S1), including variable numbers of cells (40-1000) and sites (1000-50000). We chose a set of scenarios and parameter values that in our opinion are representative of different situations that researchers are likely to encounter. In all cases, the cell sample was assumed to come from an exponentially growing population (growth rate equal to $1\times10^{-4}$) with a present-day effective population size of 10000. Across scenarios, we set a constant value of 0.1 for the root branch. Note that the mutations in this branch are shared by all cells (Fig. S1). We also defined an outgroup branch length (Fig. S1) of zero in all cases, so the healthy cell and the most recent common ancestor (MRCA) of the sample (a single healthy cell plus a number of tumor cells) have identical genotypes. The standard coalescent process results in an ultrametric tree, where all tips have the same distance from the MRCA

of the sample. However, we introduced rate variation across cell lineages by multiplying the branch lengths of the resulting coalescent genealogy with scaling factors sampled from a Gamma distribution with a mean of 1.0 (see Yang 1996).

Only in the first simulation scenario, we considered a fixed number of SNVs. In the remaining scenarios, the number of observed mutations resulted from the application of a mutation rate of $1\times10^{-6}$ [47], plus the different sc-DNA-seq errors. We explored different infinite- and finite-sites mutation models at the single nucleotide or trinucleotide level. Except for the ISM scenarios (Simulations 1 and 3), we introduced a variation of the mutation rate across sites using a Gamma distribution with a mean of 1.0 (as in Yang 1996).

We simulated the observed genotype matrices in two distinct ways. In the first three scenarios (Simulations 1-3) we obtained the observed genotypes by directly adding sequencing/amplification errors (i.e., changing one or both alleles) and ADO to the simulated genotype matrices. In the other three (Simulations 4-6), the generation of the observed genotypes was more complex. In this case, we first simulated read counts for each cell based on the true genotypes, considering different overdispersed sequencing depths, as well as amplification and sequencing errors. For simplicity, we assumed that maternal and paternal chromosomes were amplified with the same probability, and the number of reads was half for those genomic positions in which only one allele was amplified. In Simulation 5, we introduced so-called doublets, that is, two cells that are erroneously sequenced together, and that thus appear as a single cell in the sequencing data. For every combination of parameters, we generated 100 replicates. In total, we generated 19,400 cell samples.

**Simulation 1: infinite-site model, low number of SNVs ("target-ISM")**
We started with a simple scenario with 40 sampled cells and exactly 250, 500, or 1000 SNVs, assuming a diploid ISM model [48]. Under this model, a given site can only accumulate a single mutation along the genealogy, either in the maternal or paternal chromosome. Genotype errors and ADO were introduced at different rates (Table 1).

**Simulation 2: finite-site model, large number of SNVs ("WGS-FSM")**
In this case, we simulated a larger number of SNVs, more typical of whole-genome sequencing (WGS) experiments. The number of sampled cells was 100. The mutation model in this case was a non-reversible version of the finite-site General Time Reversible Markov model [20], that we called GTnR, assuming a set of single-nucleotide instantaneous rates extrapolated (basically we pooled the same mutation in the same rate independently of the 5' and 3' context) from the trinucleotide mutational signature 1 at COSMIC (https://cancer.sanger.ac.uk/cosmic/signatures):

$$Q_{GTnR} = \begin{bmatrix} 0 & 0.03 & 0.12 & 0.04 \\ 0.11 & 0 & 0.02 & 0.68 \\ 0.68 & 0.02 & 0 & 0.11 \\ 0.04 & 0.12 & 0.03 & 0 \end{bmatrix}$$

The overall mutation rate was set to $1\times10^{-6}$, which resulted in about 2000 *true* SNVs (see Table 1). However, since ADO events and genotype errors can introduce false negatives and false positives, the number of *observed* SNVs ranged between 1147 and 10000. The mutation rates varied across sites according to a Gamma distribution (+G) with shape parameter and mean equal to 1.0 (i.e., moderate among-site rate heterogeneity).

**Simulation 3: mutational signatures, large number of SNVs ("WGS-sig")**
This scenario is similar to the previous one but simpler, with 60 cells, and using a trinucleotide ISM model, with COSMIC signatures 1 and 5. The former is a ubiquitous signature in human cells with a predominance of C>T transitions in the NCG trinucleotide context, and related to the spontaneous deamination of 5-methylcytosine [49]. The latter is also a common age-related signature with a predominance of T>C substitutions in the ATN trinucleotide context, related to transcriptional strand bias [50].

**Simulation 4: genotype likelihoods from NGS read counts ("NGS-like")**
In this scenario, and in the next two, we simulated NGS read counts from the simulated genotypes. The number of sampled cells was 40, with 10000 sites and the same mutation model (GTnR+G) and mutation rates as in Simulation 3. We explored three sequencing depths (5x, 30x and 100x), three ADO rates (0, 0.05, 0.10), three amplification error rates (0, 0.05, 0.10), and three sequencing error rates (0, 0.01, 0.05). We assumed amplification and sequencing errors among nucleotides were equally likely. From the read counts, CellCoal can also infer the likelihood for all 10 possible (unphased) genotypes at each SNV site, in this case under a 4-template amplification model. The input for CellPhy were all 10 genotype likelihoods, while for the other programs we used the most likely genotype.

**Simulation 5: NGS doublets ("NGS-doublet")**
In this case, we intended to explore the effects of doublets in the data. Settings were very similar to those specified for Simulation 4 but, for simplicity, we fixed the sequencing depth to 5x, and explored only two amplification error rates (0, 0.05) and two sequencing error rates (0, 0.01). We tested using four doublet rates (0, 0.05, 0.10, 0.20).

**Simulation 6: NGS for large numbers of cells and SNVs ("NGS-large")**
Finally, in order to assess the scalability of the tools, we simulated scenarios with 100, 500, or 1000 cells, and with 1000, 10000, or 50000 sites. Given the mutation rate, a large number of cells, and, most importantly, the amplification and sequencing error rates, almost all sites were observed as SNVs. Settings were very similar to those specified for Simulation 5 but, for simplicity, we fixed the sequencing depth to 5x, and explored only one amplification (0.05), and one sequencing (0.01) error value.

## Settings for the phylogenetic analyses

**Coding DNA into ternary genotypes**
Our simulations produce unphased DNA genotypes, with 10 possible states. However, except for CellPhy, existing tools work with an alphabet composed of 0 (homozygous for the reference allele), 1 (heterozygous), 2 (homozygous for the alternative allele) and 3 (missing genotype). Therefore, in order to compare with these tools, we had to encode the simulated DNA genotypes into ternary genotypes (0-3). For this, we used the true reference allele, considering that, in real life, we normally know which allele is the reference. For the sake of simplicity, we did not introduce germline mutations. Importantly, our simulations do not necessarily produce bi-allelic SNVs, as in the finite-site model multiple mutations can occur at the same time, and amplification and sequencing errors can also result in new alleles being called. CellPhy does not have any limitation with respect to the number of alleles at an SNV site, but competing tools handle multi-allelic sites in different ways. We considered three ways of coding sites with more than two alleles into ternary genotypes:

- Option *keep*: transform all heterozygotes to "1", and all homozygotes for the alternative allele/s to "2". In this case, all simulated sites are kept, regardless of the number of observed alleles.
- Option *remove*: eliminate sites from the data with more than two alleles. The final genotype matrix includes only bi-allelic sites.
- Option *missing*: keep only those genotypes that contain the reference allele and/or the major (most common) alternative allele. All other genotypes (containing minor alternative alleles) are considered as missing data ("3"). The final ternary genotype matrix, therefore, includes the same number of sites as the original DNA genotype matrix.

We explored all three encoding options only in Simulations 1 and 2. In the remaining simulations, we used only the 'missing' option, as it maximized accuracy in most cases.

**TNT settings**
For all simulated and empirical datasets, we performed the TNT analyses using a binary data matrix in TNT format. For each run, we allowed 1000 trees to be retained and performed tree searches by setting *mult = replic 100*. All equally parsimonious trees were stored and additional *ttags* were used to store branch lengths and bootstrap support values.

**OncoNEM settings**
OncoNEM analyses were performed following the recommended settings in the OncoNEM vignette, but only for Simulation 1 due to its heavy computational requirements. We set the false positive rate to be the true genotype error for each scenario, and tree search was performed for 200 iterations.

**infSCITE settings**
For Simulations 1-3, we ran infSCITE using a ternary data matrix composed of 0, 1, 2 and 3, as described above, and set the false positive rate (*-fd*) to be the true genotype error for each simulated scenario. For Simulations 4-5, the false positive rate was set as the sum of the true sequencing and amplification errors rates for each scenario. For the empirical analyses, the false positive rate was set to *1e-05*. In all runs, the remaining parameters were set to default values and results were obtained after running an MCMC chain with 5 million steps, which we found to be a fair compromise between runtime and apparent MCMC convergence (the best tree score barely changed after 1M iterations).

**SiFit settings**
For all simulated and empirical datasets, we ran SiFit for 200000 iterations. The command line was:
*java -jar /SiFit.jar -m <CELLS> -n <SNVS> -r 1 -iter 200000 -df 1 -ipMat snv_hap.XXX.sf -cellNames snv_hap.XXXX.names*

**CellPhy settings**
For Simulations 1-6, we performed a heuristic tree search starting from a single parsimony-based tree, and using the GTGTR4 mutation model. The command line for runs using the ML genotype matrix was *cellphy.sh RAXML --search --msa snv_hap.XXXX.phy --model GTGTR4+FO+E --tree pars{1}*, while for using genotype likelihoods (VCF) was *cellphy.sh RAXML --search --msa vcf.XXXX --model GTGTR4+FO --tree pars{1}*. For all empirical datasets except LS140, in order to take advantage of the genotype likelihood model, we used an *in-house* bash script (sc-caller-convert.sh, distributed together with CellPhy) to convert the PL field from our SC-Caller VCFs. In short, following SC-Caller authors' suggestion, we took the highest likelihood score of the first two values in the PL field (i.e., sequencing noise, amplification artifact) as the phred-scaled genotype likelihood of the reference homozygous (0/0) genotype, and the remaining values as the likelihood for heterozygous (0/1) and alternative homozygous (1/1) genotypes, respectively. Afterwards, we ran CellPhy using the following command line *cellphy.sh RAXML --all --msa XXXX.vcf --model GTGTR4+FO --bs-metric fbp,tbe --bs-trees 100* to perform an all-in-one analysis (ML tree inference and bootstrapping based on 100 bootstrap trees). For LS140, since we only had the genotype matrix available and the dataset was generated without resorting to whole-genome amplification, we ran CellPhy without the single-cell error model using the following command line *cellphy.sh RAXML --all --msa XXXX.vcf --model GTGTR4+FO --prob-msa off --bs-metric fbp,tbe --bs-trees 100*.

## Evaluation of phylogenetic accuracy

We defined *phylogenetic accuracy* as one minus the normalized Robinson-Foulds (nRF) distance [51] between the inferred tree and the (true) simulated tree. This normalization consists in dividing the (absolute) RF distance by the total number of (internal) branches in *both* trees. Hence, the nRF distance is a convenient metric that goes from zero to one and reflects the proportion of branches (bipartitions of the data) that have been correctly inferred.

## Running time comparisons

We characterized the computational efficiency of CellPhy by comparing running times for all methods on six datasets from Simulations 1-3 (*sim1-ADO:0.50,ERR:0.10*, *sim2-ADO:0.10,ERR:0.05,* and *sim3-ADO:0.15,ERR:0.10,Signature1*) and two empirical datasets (CRC24 and L86) described below. The central processing unit (CPU) running time was defined as the real time returned by the Linux/Unix 'time' command. All analyses were run using a single core from an Intel Xeon E5-2680 v3 Haswell Processor 2.5GHz with 128 Gb of RAM.

## Analysis of empirical data

### In-house single-cell WGS data from a colorectal cancer patient (CRC24)

We obtained a fresh frozen primary tumor tissue, together with adjacent normal tissue, from a colorectal cancer patient from the Biobank of I.D.I.S.-C.H.U.S. (PT13/0010/0068), integrated into the Spanish National Biobank Network, and processed following standard operating procedures with the appropriate approval of the Ethical and Scientific Committees (CAEI Galicia 2014/015). We isolated EpCAM+ cells with a BD FACSAria III cytometer and amplified the genomes of 24 cells with Ampli1 (Silicon Biosystems). For each cell, we built whole genome sequencing libraries using the KAPA (Kapa Biosystems) library kit. Each library was then sequenced at ~6x on an Illumina Novaseq 6000 at the National Center of Genomic Analysis (CNAG-CR; https://www.cnag.crg.eu/).

### Retrieval of publicly available datasets (L86, E15 and LS140)

We also analyzed three public data sets with 86, 15 and 140 cells (referred to as L86, E15 and LS140, respectively). The L86 dataset consists of targeted sequencing data from 86 cells from a metastatic colorectal cancer patient (Leung et al. 2017) that we downloaded from the Sequence Read Archive (SRA) in FASTQ format, together with paired healthy-tumor bulk cell population samples (accession number: SRP074289). The E15 dataset consists of WGS data from 15 neurons (Evrony et al. 2015) from a healthy donor, downloaded from the SRA in FASTQ format, together with a bulk cell population from heart tissue (accession number: SRP041470). The LS140 dataset consists of 140 single cell-derived human haematopoietic stem and progenitor colonies from a healthy individual [38]. For this dataset, we directly downloaded the substitution calls from Mendeley data archive (https://data.mendeley.com/datasets/yzjw2stk7f/1).

### NGS data processing and variant calling

We aligned single-cell and bulk reads to the human reference GRCh37 using the MEM algorithm in the BWA software [52]. For all datasets, the mapped reads were then independently processed by filtering reads displaying low mapping-quality, performing local realignment around indels, and removing PCR duplicates. For the tumor bulk samples (i.e., CRC24 & L86) we obtained SNV calls using the paired-sample variant-calling approach implemented in the MuTect2 software [53]. For the E15 dataset, we ran HaplotypeCaller from the Genome Analysis Toolkit (GATK) [54] software on the bulk sample from the heart tissue to identify, and subsequently remove, all germline variants.

In parallel, we used the single-cell specific SC-caller software [34] to retrieve single-cell SNV calls. In short, for each single-cell BAM we ran SC-Caller together with the corresponding healthy bulk DNA as input under default settings. Since different amplification methods were used to generate each dataset, we defined the bias estimation interval (*--lamb*) as the average amplicon size of each amplification method - 10,000 for MDA-based protocols (L86, E15) and 2,000 for Ampli1 (CRC24). In addition, since the actual genomic targets of the L86 dataset were not available, we ran SC-caller on the entire exome and applied a series of heavy filters (see below) to remove potential off-target calls. We additionally estimated copy-number variants (CNVs) for each single-cell dataset. For the sc-WGS datasets (CRC24 and E15), we obtained CNV calls with the Ginkgo software [55] using variable-length bins of around 500 kb. As for the L86 dataset, CNVs were determined using CNVPanelizer, an algorithm specifically designed to infer copy number states from targeted sequencing data.

We filtered our raw single-cell VCFs by excluding short indels, SNVs with a flag other than “True” in the SO format field (i.e., showing weak evidence of being a true somatic mutation), and variable sites with an alternative read count < 3. We additionally excluded variable sites in which the ML genotype estimate was above 120 (Phred-scaled) as such uncertainty in the genotype call was usually associated with sites experiencing a massive disparity in the proportion of both alleles (i.e., allelic bias). Moreover, as we are primarily interested in analyzing diploid genomic regions, SNVs located within CN variable regions were additionally removed from each single-cell VCF.

For each dataset, we then merged single-cell VCFs using the bcftools software [56] and a “consensus” filter was applied to only retain sites present in at least one cell and the bulk tumor sample, or in two cells. For the E15 dataset, we limited this “consensus” filter to somatic sites observed in at least two cells, as all variants observed in the bulk sample were classified as germline. Finally, we additionally

removed positions missing (i.e., not covered by any read) in more than 50% of the cells, as well as SNVs comprising more than one alternative allele. Importantly, for the L86 dataset, off-target SNVs located outside exonic regions were additionally filtered out. For the LS140, we converted the binary genotype matrix into a VCF by transforming 0, 1 and NA values into 0/0, 0/1 and ./., respectively. Afterwards, all duplicated (non-biallelic) positions and indels were also removed.

# Data availability

Raw single-cell whole genome sequencing data from CRC24 have been deposited in the Sequence Read Archive (SRA - https://www.ncbi.nlm.nih.gov/sra) database under the accession code XXXXX. We have additionally analyzed previously published single-cell data sets [31,37]. Raw sequencing data for these sets are available from SRA database, under accession numbers SRP074289 (L86) and SRP041470 (E15). Furthermore, the genotype matrix for LS140 dataset [38] was generated from the substitution calls available at the Mendeley data archive (https://data.mendeley.com/datasets/yzjw2stk7f/1).

# Acknowledgments

We want to thank Debora Chantada, Pilar Alvariño, and Sonia Prado for their help in obtaining the CRC24 dataset, and Phylogenomics lab members for their comments.

# Funding

This work was supported by the European Research Council (ERC-617457- PHYLOCANCER awarded to D.P.) and by the Spanish Ministry of Economy and Competitiveness - MINECO (BFU2015-63774-P awarded to D.P.). D.P. receives further support from Xunta de Galicia. J.M.A. is currently supported by an AXA Research Fund Postdoctoral Fellowship. This work was also financially supported by the Klaus Tschira Foundation (A.K. and A.S.).

# Author contributions

D.P. conceived the CellPhy model and the experimental design. A.K. implemented the model within RAxML-NG and ran part of the simulations. A.S. supervised the phylogenetic implementation. J.M.A. ran part of the simulations and performed the empirical analyses. All authors contributed to manuscript writing.

# References

1. Gawad, C., Koh, W. & Quake, S. R. Single-cell genome sequencing: current state of the science. *Nat. Rev. Genet.* **17**, 175–188 (2016).
2. Tanay, A. & Regev, A. Scaling single-cell genomics from phenomenology to mechanism. *Nature* **541**, 331 (2017).
3. Macaulay, I. C. & Voet, T. Single cell genomics: advances and future perspectives. *PLoS Genet.* **10**, e1004126 (2014).
4. Baslan, T. & Hicks, J. Unravelling biology and shifting paradigms in cancer with single-cell sequencing. *Nat. Rev. Cancer* **17**, 557–569 (2017).
5. Stegle, O., Teichmann, S. A. & Marioni, J. C. Computational and analytical challenges in single-cell transcriptomics. *Nat. Rev. Genet.* **16**, 133–145 (2015).
6. Lim, B., Lin, Y. & Navin, N. Advancing Cancer Research and Medicine with Single-Cell Genomics. *Cancer Cell* **37**, 456–470 (2020).
7. Marioni, J. C. & Arendt, D. How Single-Cell Genomics Is Changing Evolutionary and Developmental Biology. *Annu. Rev. Cell Dev. Biol.* **33**, 537–553 (2017).
8. Wiedmeier, J. E., Noel, P., Lin, W., Von Hoff, D. D. & Han, H. Single-Cell Sequencing in Precision Medicine. *Precision Medicine in Cancer Therapy* 237–252 (2019) doi:10.1007/978-3-030-16391-4_9.
9. Lähnemann, D. *et al.* Eleven grand challenges in single-cell data science. *Genome Biol.* **21**, 31 (2020).

10. Navin, N. E. Cancer genomics: one cell at a time. *Genome Biol.* **15**, 452 (2014).
11. Kuipers, J., Jahn, K. & Beerenwinkel, N. Advances in understanding tumour evolution through single-cell sequencing. *Biochim. Biophys. Acta Rev. Cancer* **1867**, 127–138 (2017).
12. Zafar, H., Navin, N., Nakhleh, L. & Chen, K. Computational approaches for inferring tumor evolution from single-cell genomic data. *Current Opinion in Systems Biology* **7**, 16–25 (2018).
13. Ross, E. M. & Markowetz, F. OncoNEM: inferring tumor evolution from single-cell sequencing data. *Genome Biol.* **17**, 69 (2016).
14. Jahn, K., Kuipers, J. & Beerenwinkel, N. Tree inference for single-cell data. *Genome Biol.* **17**, 86 (2016).
15. Kuipers, J., Jahn, K., Raphael, B. J. & Beerenwinkel, N. Single-cell sequencing data reveal widespread recurrence and loss of mutational hits in the life histories of tumors. *Genome Res.* **27**, 1885–1894 (2017).
16. Zafar, H., Tzen, A., Navin, N., Chen, K. & Nakhleh, L. SiFit: inferring tumor trees from single-cell sequencing data under finite-sites models. *Genome Biol.* **18**, 178 (2017).
17. Kozlov, A. M., Darriba, D., Flouri, T., Morel, B. & Stamatakis, A. RAxML-NG: A fast, scalable, and user-friendly tool for maximum likelihood phylogenetic inference. *Bioinformatics* **35**, 4453–4455 (2019).
18. Goloboff, P. A. & Catalano, S. A. TNT version 1.5, including a full implementation of phylogenetic morphometrics. *Cladistics* **32**, 221–238 (2016).
19. Felsenstein, J. *Inferring phylogenies*. vol. 2 (Sinauer associates Sunderland, MA, 2004).
20. Tavaré, S. Some probabilistic and statistical problems in the analysis of DNA sequences. *Lectures on mathematics in the life sciences* **17**, 57–86 (1986).
21. Felsenstein, J. Evolutionary trees from DNA sequences: a maximum likelihood approach. *J. Mol. Evol.* **17**, 368–376 (1981).
22. Stamatakis, A. RAxML version 8: a tool for phylogenetic analysis and post-analysis of large phylogenies. *Bioinformatics* **30**, 1312–1313 (2014).
23. Zhou, X., Shen, X.-X., Hittinger, C. T. & Rokas, A. Evaluating Fast Maximum Likelihood-Based Phylogenetic Programs Using Empirical Phylogenomic Data Sets. *Mol. Biol. Evol.* **35**, 486–503 (2018).
24. Felsenstein, J. Confidence limits on phylogenies: an approach using the bootstrap. *Evolution* **39**, 783–791 (1985).
25. Lemoine, F. *et al.* Renewing Felsenstein's phylogenetic bootstrap in the era of big data. *Nature* **556**, 452–456 (2018).
26. Yang, Z., Kumar, S. & Nei, M. A new method of inference of ancestral nucleotide and amino acid sequences. *Genetics* **141**, 1641–1650 (1995).
27. Li, J. *et al.* CDC5L Promotes hTERT Expression and Colorectal Tumor Growth. *Cell. Physiol. Biochem.* **41**, 2475–2488 (2017).
28. Ma, S. S. Q. *et al.* ROR2 is epigenetically inactivated in the early stages of colorectal neoplasia and is associated with proliferation and migration. *BMC Cancer* **16**, 508 (2016).
29. Pan, H. *et al.* EXOSC5 as a Novel Prognostic Marker Promotes Proliferation of Colorectal Cancer via Activating the ERK and AKT Pathways. *Front. Oncol.* **9**, 643 (2019).
30. Li, S.-R., Gyselman, V. G., Lalude, O., Dorudi, S. & Bustin, S. A. Transcription of the inositol polyphosphate 1-phosphatase gene (INPP1) is upregulated in human colorectal cancer. *Molecular Carcinogenesis: Published in cooperation with the University of Texas MD Anderson Cancer Center* **27**, 322–329 (2000).
31. Leung, M. L. *et al.* Single-cell DNA sequencing reveals a late-dissemination model in metastatic colorectal cancer. *Genome Res.* **27**, 1287–1299 (2017).
32. Zafar, H., Navin, N., Chen, K. & Nakhleh, L. SiCloneFit: Bayesian inference of population structure, genotype, and phylogeny of tumor clones from single-cell genome sequencing data. *Genome Res.* **29**, 1847–1859 (2019).
33. Satas, G., Zaccaria, S., Mon, G. & Raphael, B. J. SCARLET: Single-Cell Tumor Phylogeny Inference with Copy-Number Constrained Mutation Losses. *Cell Systems* **10**, 323–332.e8 (2020).
34. Dong, X. *et al.* Accurate identification of single-nucleotide variants in whole-genome-amplified single cells. *Nature Methods* vol. 14 491–493 (2017).
35. Alhopuro, P. *et al.* Unregulated smooth-muscle myosin in human intestinal neoplasia. *Proc. Natl. Acad. Sci. U. S. A.* **105**, 5513–5518 (2008).
36. Romero-Pérez, L., Surdez, D., Brunet, E., Delattre, O. & Grünewald, T. G. P. STAG Mutations in Cancer. *Trends Cancer Res.* **5**, 506–520 (2019).
37. Evrony, G. D. *et al.* Cell lineage analysis in human brain using endogenous retroelements. *Neuron* **85**, 49–59 (2015).
38. Lee-Six, H. *et al.* Population dynamics of normal human blood inferred from somatic mutations. *Nature* **561**, 473–478 (2018).
39. Yu, G., Smith, D. K., Zhu, H., Guan, Y. & Lam, T. T. ggtree : an r package for visualization and annotation of phylogenetic trees with their covariates and other associated data. *Methods in Ecology and Evolution* vol. 8 28–36 (2017).
40. Zafar, H., Wang, Y., Nakhleh, L., Navin, N. & Chen, K. Monovar: single-nucleotide variant detection in single cells. *Nat. Methods* **13**, 505–507 (2016).
41. Robinson, D. F. Comparison of labeled trees with valency three. *J. Combin. Theory Ser. B* **11**, 105–119 (1971).
42. Kozlov, O. Models, optimizations, and tools for large-scale phylogenetic inference, handling sequence uncertainty, and taxonomic validation. (Karlsruher Institut für Technologie (KIT), 2018).
43. Fletcher, R. Practical Methods of Optimization. (2000) doi:10.1002/9781118723203.
44. Brent, R. P. An algorithm with guaranteed convergence for finding a zero of a function. *The Computer Journal*

vol. 14 422–425 (1971).
45. Lutteropp, S., Kozlov, A. M. & Stamatakis, A. A fast and memory-efficient implementation of the transfer bootstrap. *Bioinformatics* **36**, 2280–2281 (2020).
46. Posada, D. CellCoal: Coalescent Simulation of Single-Cell Sequencing Samples. *Mol. Biol. Evol.* **37**, 1535–1542 (2020).
47. Martincorena, I. & Campbell, P. J. Somatic mutation in cancer and normal cells. *Science* **349**, 1483–1489 (2015).
48. Kimura, M. The number of heterozygous nucleotide sites maintained in a finite population due to steady flux of mutations. *Genetics* **61**, 893–903 (1969).
49. Alexandrov, L. B. *et al.* Signatures of mutational processes in human cancer. *Nature* **500**, 415–421 (2013).
50. Alexandrov, L. B. *et al.* Clock-like mutational processes in human somatic cells. *Nat. Genet.* **47**, 1402–1407 (2015).
51. Robinson, D. F. & Foulds, L. R. Comparison of phylogenetic trees. *Math. Biosci.* **53**, 131–147 (1981).
52. Li, H. Aligning sequence reads, clone sequences and assembly contigs with BWA-MEM. *arXiv [q-bio.GN]* (2013).
53. Cibulskis, K. *et al.* Sensitive detection of somatic point mutations in impure and heterogeneous cancer samples. *Nature Biotechnology* vol. 31 213–219 (2013).
54. Poplin, R. *et al.* Scaling accurate genetic variant discovery to tens of thousands of samples. doi:10.1101/201178.
55. Garvin, T. *et al.* Interactive analysis and assessment of single-cell copy-number variations. *Nat. Methods* **12**, 1058–1060 (2015).
56. Li, H. A statistical framework for SNP calling, mutation discovery, association mapping and population genetical parameter estimation from sequencing data. *Bioinformatics* **27**, 2987–2993 (2011).