Clonality inference from single tumor samples using low coverage sequence data (Supplementary Materials)

Nilgun Donmez[1,2], Salem Malikic[1,2], Alexander W. Wyatt[2,3], Martin E. Gleave[2], Colin C. Collins[2,3], and S. Cenk Sahinalp[1,2,4]

1 School of Computing Science, Simon Fraser University, Burnaby, BC, Canada
2 Vancouver Prostate Centre, Vancouver, BC, Canada
3 Dept. of Urologic Sciences, University of British Columbia, Vancouver, BC, Canada
4 School of Informatics and Computing, Indiana University, Bloomington, IN, USA

Appendix 1: mILP formulation

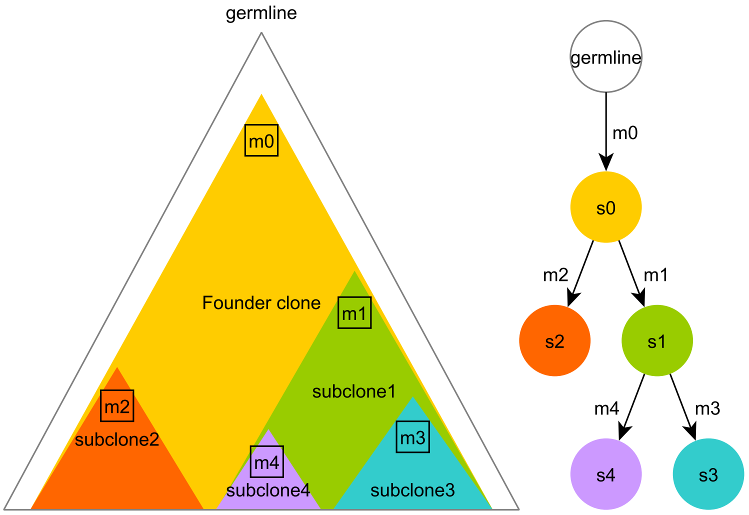

Fig. S1. Illustration of cancer progression and clonal evolution. (Left:) Cancer typically starts with an initial founder clone containing one or more driver mutations (m0). During cancer progression, additional subclones (e.g. subclone1, subclone2, etc) might emerge with newly acquired mutations (e.g. m1, m2), possibly in response to therapy or other stress factors. (Right:) The tree representation of the clonal evolution example given on the left. In our formulations, each subclone corresponds to a node in this tree. Clonal mutations are those assigned to the founder clone (s0), while subclonal mutations are those assigned to any node under the founder clone.

Figure S1 illustrates the clonal evolution of cancer and how it relates to tree topologies in our formulations. Let $T = (V, E)$ be an arbitrary tree topology with $k = |V|$ nodes and let $B(v)$ denote the set of children of node v in T. We define β_v to be the cancer cell fraction of node (i.e. subclone) v and impose the following constraints to ensure the solution admits a valid phylogeny:

$$\forall v \in V : \beta_v \geq \sum_{u \in B(v)} \beta_u \tag{1}$$

Additionally, we require that $\beta_r \leq 1.0$, where r denotes the root of T. Note that since one of the clusters obtained from the previous stage will have a CCF of 1.0 by design, β_r is guaranteed to be set to this value in the optimal solution.

Next, we define the indicator variables δ_{jv}, where $\delta_{jv} = 1$ iff subclone j is assigned to node v, and 0 otherwise. We require that each subclone is assigned to one and only one node:

$$\sum_{v \in V} \delta_{jv} = 1 \tag{2}$$

Subject to the constraints defined above, the objective of CTPsingle is to minimise the following sum:

$$\sum_j \sum_v \delta_{jv} |f_j - \beta_v| \tag{3}$$

The mixed ILP formulation as given above is expected to have few variables in single sample datasets, and can be solved using a simple iterative approach similar to the heuristic version of CITUP as described in [5].

Briefly, this is accomplished in two steps: (1) given fixed values for the variables β_v, find the optimal assignment of subclones to nodes as given by δ_{jv} that minimises equation 3; (2) given the assignment δ_{jv} of subclones to nodes, calculate values for β_v that minimise equation 3. The initial values of β_v for Step 1 are chosen randomly and these two steps are repeated in order until the decrease in objective score as given by equation 3 is less than a user-defined threshold.

While Step 1 still employs integer variables, Step 2 is a standard case of linear programming. As neither step can increase the objective score, convergence of the algorithm to at least a local optimum is guaranteed. Although convergence to a global optimum is not guaranteed for large trees, this problem can be partially alleviated by performing multiple re-starts.

Alternatively, it is possible to introduce a set of new variables x_{jv} with the additional constraints that:

$$\forall j, v : x_{jv} \geq \delta_{jv} - 1 + f_j - \beta_v \tag{4}$$

$$\forall j, v : x_{jv} \geq \delta_{jv} - 1 - f_j + \beta_v \tag{5}$$

$$\forall j, v : x_{jv} \geq 0 \tag{6}$$

In this case, the objective is modified to minimise $\sum_j \sum_v x_{jv}$. However, as we have previously shown [5], the iterative version typically works faster with comparable accuracy.

In practice, inferring phylogeny from single-sample tumors is typically an under-determined problem except for special cases where the estimated frequencies admit only one solution. As a result, rather than reporting a single unique solution, CTPsingle reports all feasible solutions with some topologies eliminated. Further elimination of tree topologies may be possible by examination of nearby germline mutations or additional information about the tumors and is left to the user.

Appendix 2: Simulation set up

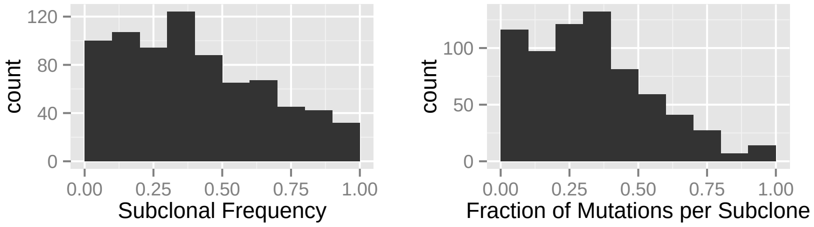

Fig. S2. The distribution of subclonal frequencies (cancer cell fractions adjusted by tumor purity) and fraction of mutations per subclone across all simulation datasets. In each case, the histograms are plotted using a bin-width of 0.1. For the fraction of mutations per subclone plot, we excluded simulation datasets that contain a single subclone.

For each set of experiment settings, we generate 50 tumor profiles. For each tumor profile, we randomly pick a number of mutations to simulate from the interval $[500, 5000]$. Next, we randomly select the number of subclones from the closed interval $[1, 4]$. Note that this value does not include the population of normal cells (i.e. healthy cell contamination in the tumor sample). Instead, tumor purity (i.e. fraction of cancerous cells in tumor sample) is uniformly chosen from the interval $[0.3, 1.0]$. While this range of subclones might sound low, we believe that it is representative of what can be reliably detected from low to medium purity tumors with low coverage sequencing data. In addition, these simulation parameters are based on our observations on the PCAWG data.

The subclonal composition for each tumor is simulated using the following procedure. First, we randomly pick a tree topology from the set of all possible rooted trees (see [5] for a method to derive such trees). Next, we simulate 'genotype proportions' using a Dirichlet distribution. This is done with the stick breaking analogy where the stick sizes are uniformly drawn from the discrete interval $[1, 10]$. The cancer cell fraction for each subclone is then computed using the genotype proportions according to the topology of the tree. Similarly, the fraction of the mutations assigned to each subclone is drawn separately from another Dirichlet distribution as described above. The distribution of subclonal frequencies (i.e. cancer cell fractions adjusted by tumor purity) and fraction of mutations per subclone across all simulation datasets is given in figure S2. As can be seen from the figure, a substantial number of simulated subclones have low frequencies. In addition, the mutations are distributed to subclone in a fairly non-uniform manner in our simulations, reflecting the challenges of real datasets.

To mimic the highly variable coverage observed in real sequencing experiments, we simulate the coverage depth of each mutational position as follows. For each position, the total number of reads is drawn from the beta-binomial distribution with parameters $N_0, \alpha = 2, \beta = 5$. This asymmetric distribution features a longer right tail with most values concentrated toward the lower end of the distribution. For experiments (1a) and (1b), we vary N_0 from 100 to 120 which results in a mean read coverage of 28∼35x per simulation. For experiments (2a) and (2b), we fix $N_0 = 10000$ which results in a mean coverage of ∼2800x.

Given the tumor purity TP for a sample, and the total read count n for a position, the variant read count y for that position is sampled from a binomial distribution with $Binom(n, p)$, where $p = 0.5 \times TP \times CF$ and CF denotes the cancer cell fraction of the subclone containing the mutation. The reference read count is simply taken to be $n - y$.

For the multi-sample experiments (1b) and (2b), we keep the tree topology and the assignment of the mutations fixed in both samples, while the tumor purity and cancer cell fractions of the subclones are simulated independently for each sample as described above. For the low coverage experiments, we also remove any mutation that has total read count less than 15 as these are likely to be rejected by a mutation caller in real sequencing experiments (although they still count towards the total number of mutations to be simulated). For multi-sample datasets, such mutations are removed from both samples as the other tools might interpret the absence of a mutation in one sample as the absence of a subclone in that sample. Note that this is an unrealistic advantage for these tools, since in a real dataset it is common to miss a mutation due to variable coverage in one sample while it is called with high confidence in another sample.

Calculation of evaluation measures and run-time settings for AncesTree, LICHeE and PyClone: To calculate the tumor purity estimated by AncesTree for a sample, we use the sum of the corresponding row of the usage matrix U reported by this tool. Similarly, for LICHeE we use the sum of the frequencies assigned to each non-germline node in a sample. The predicted number of subclones is taken to be the number of columns in matrix U for AncesTree and the number of nodes minus 1 for LICHeE (since the first node represents the germline for this tool). The subclonal frequencies for AncesTree are calculated as $U \times B$, where U and B denote the usage and clonal matrices as described in [2]. For LICHeE, the subclonal frequencies are obtained using the node frequencies and the tree topology reported. We ran AncesTree with the default parameters in all datasets. For the low coverage datasets, we ran LICHeE with the parameters 'maxVAFAbsent=minVAFPresent=0.01' and for the deep coverage datasets, we ran it with 'maxVAFAbsent=minVAFPresent=0.005'. These parameters are set according to the examples given in the LICHeE distribution. The other parameters were kept at default values. For PyClone, we generated configuration files for the simulated datasets based on the example script "pyclone_beta_binomial.yaml", which uses a beta-binomial model to perform clustering. We kept all parameters in this script the same with the exception that we increased the number of iterations to 2000. To compute the final clusters, we ran PyClone with the cluster command, however, we had to terminate these jobs after they reached a 90 hour limit on our cluster. Instead, we used an in-house script to calculate the most frequent cluster assignment configuration in the final 500 iterations. This assignment is then used to compute the number of subclones, tumor purity and cancer cell fractions. While we acknowledge that this is not ideal, we had to resort to this approach given the time limitations. The scripts that were used to generate and evaluate the simulations can be found at `https://github.com/nlgndnmz/CTPsingle`.

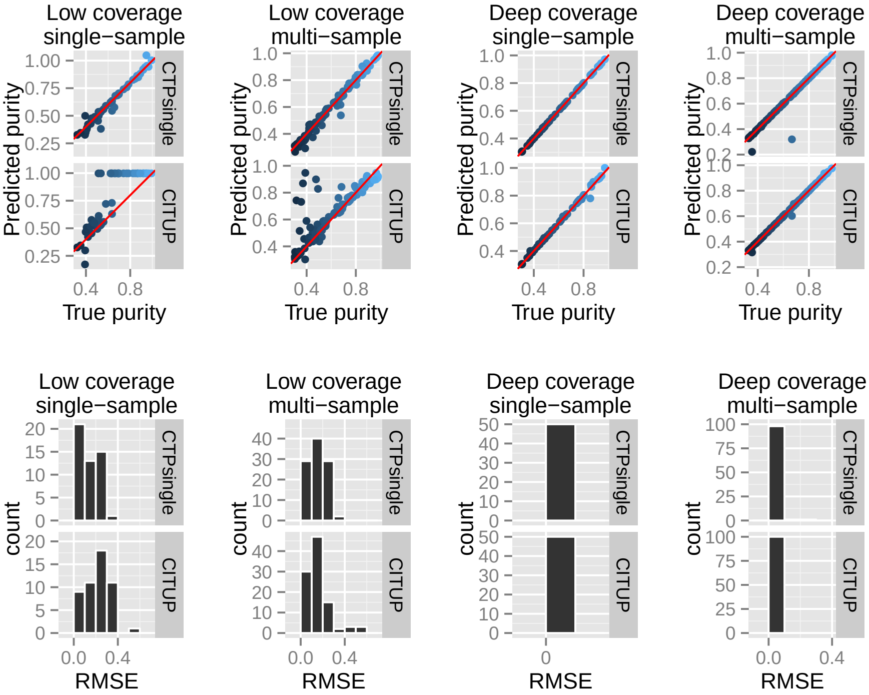

Fig. S3. Comparison of CTPsingle to CITUP on the simulation datasets. Top plots show the predicted purity versus true purity for CTPsingle and CITUP, while the bottom plots show the RMSE measure.

Appendix 3: Bioinformatic analysis of the prostate tumors

Illumina reads obtained from whole-exome sequencing are mapped to GRCh37 human reference using BWA [4]. Copy number calls are obtained from Nexus Copy Number^{TM} software from Illumina and somatic SNVs are called using MuTect[1]. For SNVs, only mutations that are marked as 'KEEP' by MuTect are included for further analysis. Mutations that reside on copy number altered regions of the genome are discarded prior to use with CTPsingle. Set of genes with non-synonymous mutations are determined using the Variant Effect Predictor tool from Ensembl[3].

Appendix 4: Effect of false positive SNVs and copy number alterations

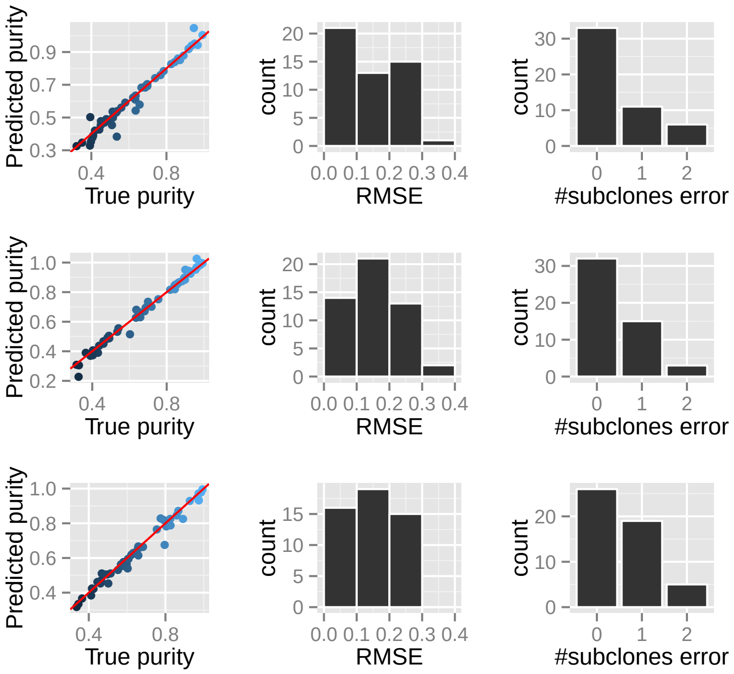

Fig. S4. Effect of false positive SNVs and copy number alterations on the performance of CTPsingle. The top row shows the performance of CTPsingle on the (1a) dataset described in the Results section. The middle row shows the performance of CTPsingle when 1% of the total SNV calls represent false positives. The bottom row shows the performance of CTPsingle when 30% of the genome is copy number altered.

Real datasets may contain a number of false positive mutation calls due to various reasons such as sequencing artefacts or mapping ambiguity. To evaluate the performance of CTPsingle in such cases, we performed an additional experiment with 50 simulations where a fraction of the called mutations represent false positives. For this experiment, we use the same parameters as in (1a), however we simulate 1% of the total mutations as false positives. For each such mutation, we randomly select a frequency from the interval $[0, 1]$ and simulate its variant read count based on this frequency.

To show that CTPsingle is able to infer clonality in tumors with moderate levels of copy number changes, we also performed 50 simulations on tumors with 30% genome aberrations. Single nucleotide mutations that fall under these aberrant regions are discarded, losing about 30% of the data points on average. The other parameters were kept the same as in experiment (1a).

Figure S4 plots the results of CTPsingle in these datasets as compared to its performance on (1a). These plots suggest that CTPsingle's performance does not deteriorate significantly in these datasets. Note that the slight improvement in the RMSE measure in the bottom row is probably due to the decreased number of total mutations usable by CTPsingle in this dataset.

0.1 Appendix 5: Power to detect subclones on low and high coverage datasets

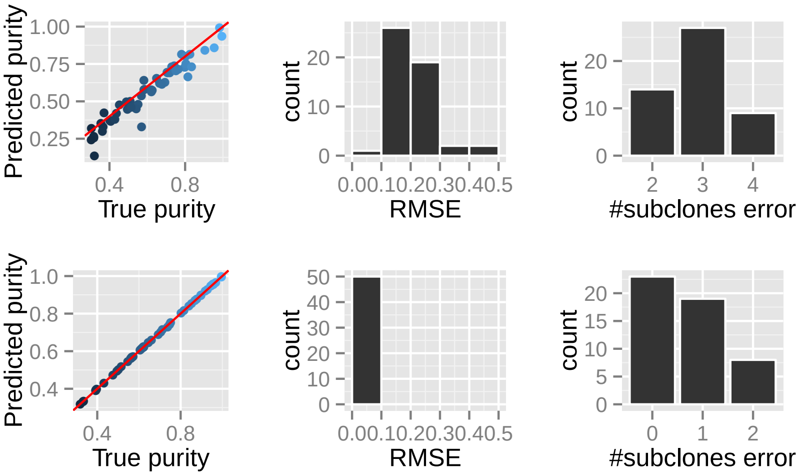

Fig. S5. Performance of CTPsingle on simulated datasets containing 5 to 6 subclones. The top plots show the performance of CTPsingle on low coverage data, while the bottom plots show the performance of CTPsingle on high coverage data. All other simulation parameters were kept as described in Results and Appendix 2.

To investigate whether CTPsingle is effective for tumors that are highly heterogeneous (i.e. contain many subclones), we also performed additional simulations with 5 to 6 subclones. For these experiments, we again generated 50 low coverage and 50 high coverage single-sample simulations. All other simulation parameters were kept the same as described in section 3.1 and Appendix 2. Figure S5 illustrates the performance of CTPsingle on these additional datasets. As can be seen from the figure, CTPsingle could not estimate the correct number of subclones when applied to low coverage data. On the other hand, it can still estimate the tumor purity with good accuracy and has reasonably low RMSE. On deep coverage data, the performance is only mildly affected by additional subclones, suggesting that CTPsingle is suitable for highly heterogeneous tumors when deep sequencing data is available.

References

1. Cibulskis, K., Lawrence, M.S., Carter, S.L., Sivachenko, A., Jaffe, D., Sougnez, C., Gabriel, S., Meyerson, M., Lander, E.S., Getz, G.: Sensitive detection of somatic point mutations in impure and heterogeneous cancer samples. Nature biotechnology 31(3), 213–219 (2013)
2. El-Kebir, M., Oesper, L., Acheson-Field, H., Raphael, B.J.: Reconstruction of clonal trees and tumor composition from multi-sample sequencing data. Bioinformatics 31(12), i62–i70 (2015)
3. Hubbard, T., Barker, D., Birney, E., Cameron, G., Chen, Y., Clark, L., Cox, T., Cuff, J., Curwen, V., Down, T., et al.: The ensembl genome database project. Nucleic acids research 30(1), 38–41 (2002)
4. Li, H., Durbin, R.: Fast and accurate long-read alignment with burrows–wheeler transform. Bioinformatics 26(5), 589–595 (2010)
5. Malikic, S., McPherson, A.W., Donmez, N., Sahinalp, C.S.: Clonality inference in multiple tumor samples using phylogeny. Bioinformatics 31(9), 1349–1356 (2015)