TrackSigFreq: subclonal reconstructions based on mutation signatures and allele frequencies

Caitlin F Harrigan[1,2,4], Yulia Rubanova[1,2,4], Quaid Morris[1,2,3,4,5,6 †], Alina Selega[2,4]

[1]*Department of Computer Science, University of Toronto, Toronto, Canada*
[2]*Donnelly Centre for Cellular and Biomolecular Research, University of Toronto, Toronto, Canada*
[3]*Department of Molecular Genetics, University of Toronto, Toronto, Canada*
[4]*Vector Institute, Toronto, Canada*
[5]*Ontario Institute for Cancer Research, Toronto, Canada*
[6]*Memorial Sloan Kettering Cancer Centre, New York, USA (pending)*
[†]*E-mail: quaid.morris@utoronto.ca*

Mutational signatures are patterns of mutation types, many of which are linked to known mutagenic processes. Signature activity represents the proportion of mutations a signature generates. In cancer, cells may gain advantageous phenotypes through mutation accumulation, causing rapid growth of that subpopulation within the tumour. The presence of many subclones can make cancers harder to treat and have other clinical implications. Reconstructing changes in signature activities can give insight into the evolution of cells within a tumour. Recently, we introduced a new method, TrackSig, to detect changes in signature activities across time from single bulk tumour sample. By design, TrackSig is unable to identify mutation populations with different frequencies but little to no difference in signature activity. Here we present an extension of this method, TrackSigFreq, which enables trajectory reconstruction based on both observed density of mutation frequencies and changes in mutational signature activities. TrackSigFreq preserves the advantages of TrackSig, namely optimal and rapid mutation clustering through segmentation, while extending it so that it can identify distinct mutation populations that share similar signature activities.

Keywords: mutational signatures; cancer evolution; subclonal reconstruction; whole genome sequencing

1. Introduction

Mutations continuously accumulate in the genomes of our somatic cells throughout our lifetime. Driver mutations confer a selective advantage to the clonal populations that contain them; sequences of driver events precede carcinogenesis. Cancerous cells continue to acquire driver mutations, creating genetically distinct subclonal populations. Characterising this intratumour heterogeneity can shed light on a tumour's evolutionary trajectory and has important clinical implications, as different subclones may respond differently to treatment.[1]

The vast majority of the cancer genome data available are from single, bulk tumour samples; current methods struggle to use these data to reconstruct detailed evolutionary histories. Until recently, subclonal reconstruction methods have attempted to cluster mutation variant allele frequencies (VAFs) to identify and order subclonal lineages. We have recently demonstrated that in some cases, more accurate reconstructions are possible when other properties of each mutation are considered,[2] specifically, the *types* of each mutation.

Different sources of mutations, external or intrinsic to the cell, can generate distinct mutational patterns. Mutations have been classified into 96 different types based on the type of substitution and the trinucleotide context.[3] One can then define a *mutational signature* to describe a probability distribution over mutation types and the signature's *activity* to represent the proportion of mutations it generates.[4] Many mutational signatures have been linked to known mutagenic processes,[3,5,6] and thus reconstructing temporal changes in signature activities that best explain the observed mutations can help identify affected pathways, predict tumour development[7] or inform choice of treatment.[4]

Recently we introduced a new method, TrackSig,[2] to detect changes in signature activities across time using topic modeling and optimal segmentation. Notably, unlike other methods for reconstructing subclone architecture, TrackSig does not group mutations by clustering their VAFs. For those methods, accuracy depends on the sequencing depth and thus can be compromised when using single bulk sample.[8,9] Instead, TrackSig constructs a pseudo-timeline by approximately ordering mutations by their inferred prevalence in the cell population and partitions this timeline into segments with similar signature activities. Changepoints between segments indicate regions where differences in signature activities arise and often correspond to boundaries between subclones.[10] TrackSig's methodology allows it to deal with measurement noise associated with mutation VAFs, making it applicable to bulk data from a single sample, and guarantees finding an optimal placement of changepoints in signature activities. TrackSig was shown to outperform competing methods at estimating activities and identifying subclonal populations in complex scenarios such as branching evolution or violation of the infinite sites assumption.[2]

Here, we describe a new method, TrackSigFreq, which identifies subclones using both mutation type and VAF. By design, TrackSig is unable to detect changepoints between distinct subpopulations that exhibit little to no change in signature activity, but a change in VAF clustering density. TrackSigFreq extends TrackSig and incorporates information about mutation VAF density, allowing accurate identification of changepoints in such scenarios. Our extension does not rely on prior clustering of VAFs and instead modifies the likelihood function used to identify the optimal segmentation of the pseudo-timeline. We show our method's improved performance compared to original TrackSig on simulated data with varying number of subclonal populations.

2. Methods

Below we provide a brief description of TrackSig's methodology for detecting changes in mutational signature activities. Next, we outline the approach of TrackSigFreq for modeling the mutation VAF distribution and explain how we incorporate it into the segmentation algorithm. Fig. 1 gives a general overview of TrackSigFreq and illustrates its relationship to TrackSig.

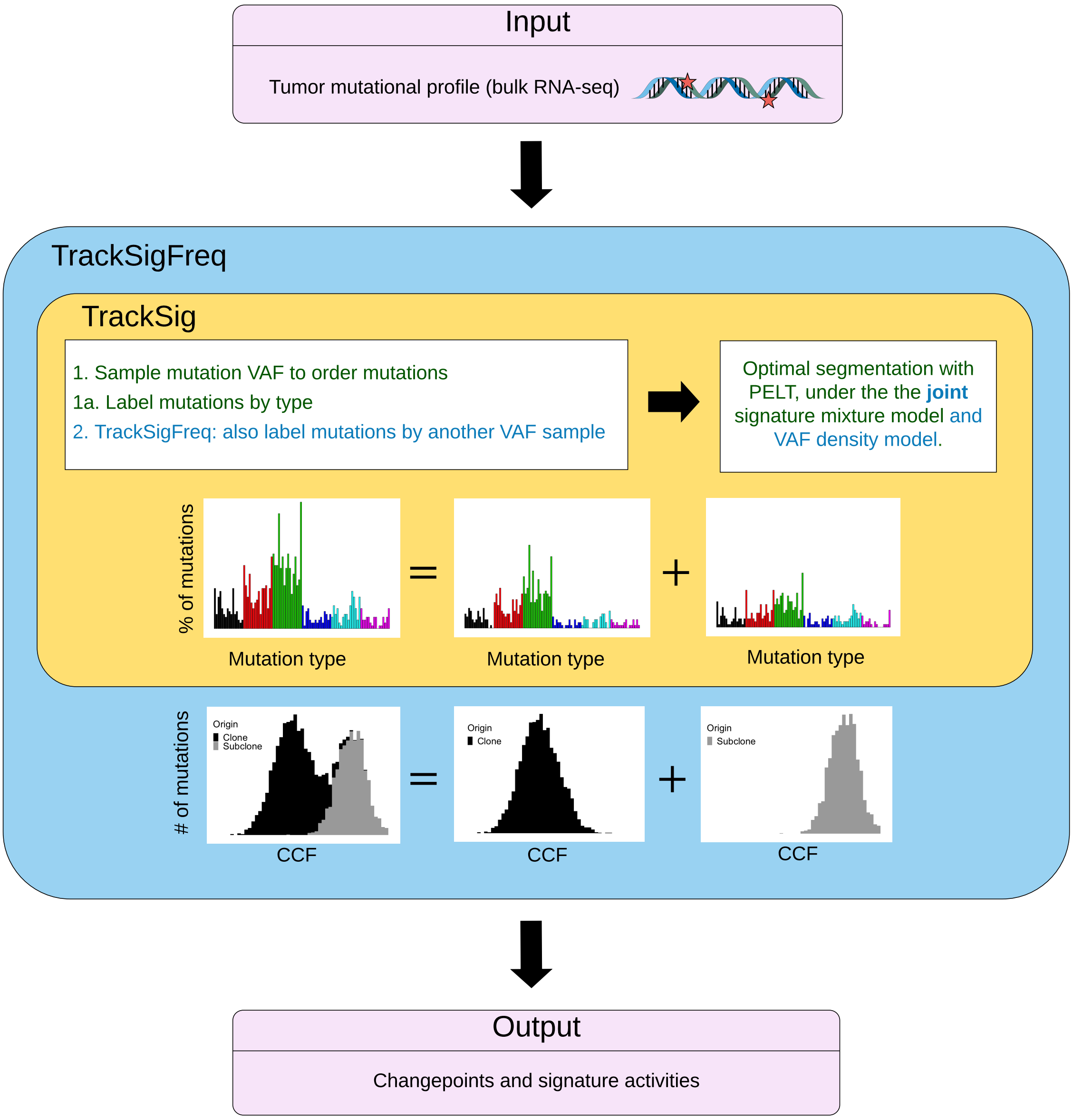
Input
Tumor mutational profile (bulk RNA-seq)
TrackSigFreq
TrackSig
1. Sample mutation VAF to order mutations
1a. Label mutations by type
2. TrackSigFreq: also label mutations by another VAF sample
Optimal segmentation with PELT, under the the joint signature mixture model and VAF density model.
% of mutations
Mutation type
Mutation type
Mutation type
# of mutations
Origin
Clone
Subclone
Origin
Clone
Origin
Subclone
CCF
CCF
CCF
Output
Changepoints and signature activities

Fig. 1. An overview of TrackSigFreq (blue box) and TrackSig (yellow box). Green text corresponds to the algorithmic steps performed by both methods, blue text outlines the extension implemented by TrackSigFreq. Both methods model the distribution of mutation types as a mixture of mutational signatures (plots in yellow box), but TrackSigFreq also models the mutation CCF distribution (plots in blue box). TrackSigFreq optimally partitions the timeline of ordered mutations using both mutational signatures and mutation VAFs, while TrackSig uses signatures only.

2.1. *TrackSig*

2.1.1. *Constructing a timeline*

TrackSig constructs an evolutionary timeline of mutation occurrence by sorting single nucleotide variants (SNVs) by their inferred population frequency (or *cancer cell fraction*, CCF), for a total of I variants. Given d_i, the total number of reads mapping to the locus containing a variant i, the number of reads containing that variant, v_i, is modeled as $v_i \sim \text{Binomial}(v_i; d_i, p_i)$, where p_i is the unobserved *variant allele frequency* of the mutation. Ideally we would like to sort mutations i and j given (v_i, d_i) and (v_j, d_j) by comparing the posterior distributions of their VAFs, p_i and p_j, or rather the CCFs implied by these VAF. The p_i posterior is easily computed via conjugacy if an uninformative Beta prior is used on p_i, i.e., $p_i \sim \text{Beta}(1,1)$:

$$p_i \sim Beta(v_i + 1, d_i - v_i + 1), \tag{1}$$

However, transforming p_i to a posterior over CCF is challenging as it depends on the copy number of the locus. Comparing the posteriors of p_i and p_j is particularly challenging if they have different copy numbers. To simplify and speed-up these computations, we generate an estimate for each p_i by sampling from this posterior, use these estimates to compute corresponding CCF_i estimates, and then order the mutations based on these. TrackSig uses a single p_i sample when the number of SNVs, I, is large, but multiple orderings can be sampled.[2]

The inferred CCF_i is computed from each sampled p_i, accounting for the copy number at the locus and sample purity (the proportion of cells in the sample that are cancerous). Specifically, we compute CCF_i by inverting this well-known relationship:

$$p_i = \frac{\rho\ m_i}{n_i}\text{CCF}_i \tag{2}$$

where ρ is the purity of the sample, m_i is the number of copies of mutant alleles per cancer cell, and n_i be the total number of copies of the locus i per cell. In many cases, $m_i = 1$ and $n_i = 2$. In the following, we will assume that $m_i = 1$, and that copy number reconstruction has been performed and n_i is provided as input.

Once the per-mutation CCFs are estimated, a pseudo-timeline is constructed by sorting these CCFs in decreasing order. The position of each SNV in this sorted list represents the pseudo-time estimate of its order of occurrence. TrackSig then partitions the pseudo-timeline into bins with constant number of SNVs (bin size of 100 mutations is chosen in Ref. 2) where each bin defines a *timepoint*. The full description of TrackSig is provided in Ref. 2.

2.1.2. *Detecting changepoints in signature activities*

TrackSig uses Pruned Exact Linear Time (PELT),[11] a dynamic programming approach to find the optimal placement of changepoints between *segments*, which are defined as regions of the timeline spanning multiple timepoints with similar mutational activities.

PELT For an ordered sequence of data, $y_{1:T} = (y_1, ..., y_T)$, and T timepoints, we seek to identify P changepoints and their positions, $\tau_{1:P} = (\tau_1, ..., \tau_P)$, splitting the data into $P + 1$ segments. PELT scores a series of ordered segments, where segment i contains datapoints $y_{(\tau_{i-1}+1):\tau_i}$. Changepoints can be identified by minimising the total segmentation score:

$$\sum_{i=1}^{P+1}[C(y_{(\tau_{i-1}+1):\tau_i})] + \beta f(P) \tag{3}$$

where C is a cost function for a segment and $\beta f(P)$ is a penalty against overfitting. PELT starts by finding partial solutions in the subsets of the timeline and uses those to recursively derive the optimal partition. Importantly, PELT prunes those changepoints that can never be the optimal changepoint in a given subproblem and by extension can not be included in the optimal solution to the total segmentation problem. This pruning allows PELT to identify the optimal changepoint in time that is subquadratic (and in some cases linear) in T.

TrackSig uses the EM algorithm[12] to fit mutational signatures to the set of mutation types in each segment being scored by PELT. The cost function of a segment is defined as $C = -2 \log \mathcal{L}_{sig}$, where $\mathcal{L}_{sig}$ is the data likelihood of a segment under the fitting mixture of predefined signatures. TrackSig minimises the Bayesian Information Criterion (BIC) by setting the PELT penalty to the number of free parameters, $f(P) = (P+1)(M-1)$. Then, minimising the objective given in Eq. 3 corresponds to finding changepoints in the pseudo-timeline that maximise the log-likelihood of observed mutation types in the found partition, while reducing the penalty associated with adding changepoints.

2.2. *TrackSigFreq*

Here we outline our extension, *TrackSigFreq*, which computes an optimal segmentation of the timeline into K segments based on changes in mutational signatures and also on VAF clustering density.

We assume that there exist K cell populations in a sample, for some unknown value of K, into which I mutations are partitioned. Each population is characterised by its CCF, ϕ_k, where $k = 1...K$ and all mutations in population k share ϕ_k. Let a mutation i belong to population k if $z_i = k$.

2.2.1. *VAF extension for TrackSig*

Extending TrackSig to incorporate VAF density requires assessing a likelihood for the (v_i, d_i) observations assigned to a segment; using this likelihood to augment the TrackSig cost function as described above; and choosing an appropriate penalty term. Then this new cost function and penalty function can be inserted directly into the TrackSig algorithm to derive optimal segmentations in terms of both VAF and mutation type.

As described in Section 2.1.1, samples of the posterior over p_i, corresponding to the observation (v_i, d_i), can be used as approximate replacements for the posteriors themselves in subsequent computations. These samples permit the use of discrete optimization algorithms, such as PELT, to find global optima. As such, our likelihood for segment k over all pairs (v_i, d_i) for which $z_i = k$ will be derived from samples $\hat{p_i}$ from $Beta(v_i + 1, d_i - v_i + 1)$. These samples $\hat{p_i}$ will be used as pseudo-observations for our VAF-based likelihood $\mathcal{L}_{VAF}$.

2.2.2. *Creating pseudo-observations*

After assigning mutations to timepoints within the pseudo-timeline using an initial round of sampling p_i values, we perform an additional round of sampling to generate estimates $\hat{p_i}$. It is important to realize that $\hat{p_i}$ is not a sample from a Beta distribution derived from the CCF of the segment, ϕ_k, but rather is a sample from our uncertainty over what the VAF is for mutation i. So, the proper way to combine these values would be to generate "pseudo-observations" of variant counts. Let $\hat{v}_i$ be a pseudo-observation for mutation i for some depth d'_i such that $\hat{v}_i = \hat{p}_i d'_i$. We can then use these pseudo-observations to estimate the posterior over p_k.

For simplicity, we set $d'_i = 1$ for all i, thus $\hat{v}_i = \hat{p}_i$. This choice of d'_i also acts to lower the bound on our certainty of $\hat{v}_i$, representing the largest possible variance on the estimate $\hat{p}_i$. Note that although $\hat{v}_i$ is not a whole number, and so does not represent a sample from the Binomial distribution, we will still use the conjugate Beta prior.

2.2.3. *Scoring a segment*

We define the cost function of a segment using the log-likelihood of pseudo-observations assigned to that segment. We compute $\log \mathcal{L}_{VAF}$ for a segment spanning mutations a to b in the TrackSigFreq timeline as:

$$\begin{aligned}
\log \mathcal{L}_{VAF} &= \log p(v_a, ..., v_b | d_a, ..., d_b, p) && (4)\\
&= \log \left(\prod_{i=a}^{b} \binom{d_i}{v_i} \int_a^b p^{\sum_{i=a}^{b} v_i} (1-p)^{\sum_{i=a}^{b} d_i - v_i} \, dp \right) && (5)\\
&= \sum_{i=a}^{b} \log \binom{d_i}{v_i} + \log \left(\mathrm{B}(\alpha, \beta)[\mathrm{IB}(b; \alpha, \beta) - \mathrm{IB}(a; \alpha, \beta)] \right), && (6)
\end{aligned}$$

where $\mathrm{IB}(x; \alpha, \beta)$ is the incomplete Beta function, $\mathrm{B}(\alpha, \beta)$ is the Beta function, and the Beta parameters are given as follows for $d_i = 1$, $v_i = \hat{v}_i$:

$$\alpha = 1 + \sum_a^b v_i \tag{7}$$

$$\beta = 1 + \sum_a^b (d_i - v_i) \tag{8}$$

Note that the sum over binomial terms cancels out when comparing segments with their composing subproblems. As such, to compare the cost of placing no changepoints in the segment between a and b with placing a changepoint at c, where $a < c < b$, we would compare the cost of the segment over $[a, b]$ with the sum of the (log-)costs of the composing subproblems over $[a, c]$ and $(c, b]$, making $\sum_{i=a}^{b} \log \binom{d_i}{v_i}$ constant with respect to the likelihood of different segmentations. We combine the VAF-based likelihood $\mathcal{L}_{VAF}$ with the TrackSig signature-based likelihood $\mathcal{L}_{sig}$ (see section 2.1.2 and Ref. 2), modifying the total cost function of a segment to be $C = -2(\mathcal{L}_{sig} + \mathcal{L}_{VAF})$. To mitigate overfitting, we modify the TrackSig BIC penalty by adding a term that scales log-linearly with the number of placed changepoints.

3. Results

To compare the performance of TrackSig with our extension TrackSigFreq, a set of simulations was generated. Three types of tumour sample were simulated: one cluster of mutations, two clusters, and three clusters. Respectively, these simulations have 0, 1, and 2 true changepoints and represent tumours with a clonal cluster only, a clonal cluster and one subclone, and a clonal cluster and two subclones. Real tumour data suggest that these patterns are relatively common among patient samples.[10] TrackSig and TrackSigFreq were run on the simulated data to recover changepoints and compared based on the percentage of simulations where the correct number of changepoints was recovered. The algorithm used to create simulated data was slightly modified from Ref. 2 to highlight a regime under which TrackSigFreq demonstrates improved performance. There are many mutational signature sets that continue to be defined,[10,13,14] and TrackSig allows user-provided signature sets to be fit to data. In particular, we make use of those derived as part of the Pancancer Analysis of Whole Genomes (PCAWG) initiative.[10]

First, we give an example of scenarios where TrackSigFreq has improved performance over TrackSig (Fig. 2). To create Fig. 2, $n = 25$ two-cluster simulations were generated. These comprised of a clonal cluster and a subclonal cluster. The signature activities and mutation CCF values in each of these 25 simulations were manually fixed to illustrate TrackSigFreq's behaviour with a clear ground truth. Every clonal cluster was simulated as having $\phi_1 = 1$ and signature activities as in Table 1. To create the subclonal cluster, ϕ_2 was picked from five evenly spaced values ranging from 0.2 to 0.85, which places the subclonal cluster at different locations along the pseudo-timeline, gradually decreasing in distance to the clonal cluster. For each possible ϕ_2, a change in signature activity, δ, was picked from five evenly spaced values ranging from 5% to 30%. This range represents the lower limit of signature change that TrackSig can reliably detect[10] and a change above the threshold where TrackSig will always detect a signature change. To get the subclonal signature activities, δ was subtracted from the clonal activity of $S3$, and added to the clonal activity of $S2.13$. This appears as signature $S3$ having a high level of activity in the clonal population and decreasing in the subclonal population, while signature $S2.13$ exhibits opposite behaviour and absorbs the proportion of activity lost by $S3$.

Table 1. Signature activities for 25 simulations in the two-cluster scenario.

Signature	Clonal cluster activity (%)	Subclonal cluster activity (%)
$S3$	60	$60 - \delta$
$S2.13$	25	$25 + \delta$
$S5$	10	10
$S1$	5	5

Fig. 2 demonstrates that TrackSigFreq can successfully identify a changepoint in scenarios where a signature change is small, such as the top row of plots, which corresponds to a signature change of 5%. Note that the changepoint location is consistent with cluster locations. TrackSig does not detect any changepoints in these scenarios. In other scenarios, both methods either locate the same changepoint (shown in black) or find changepoints that are close to each other.

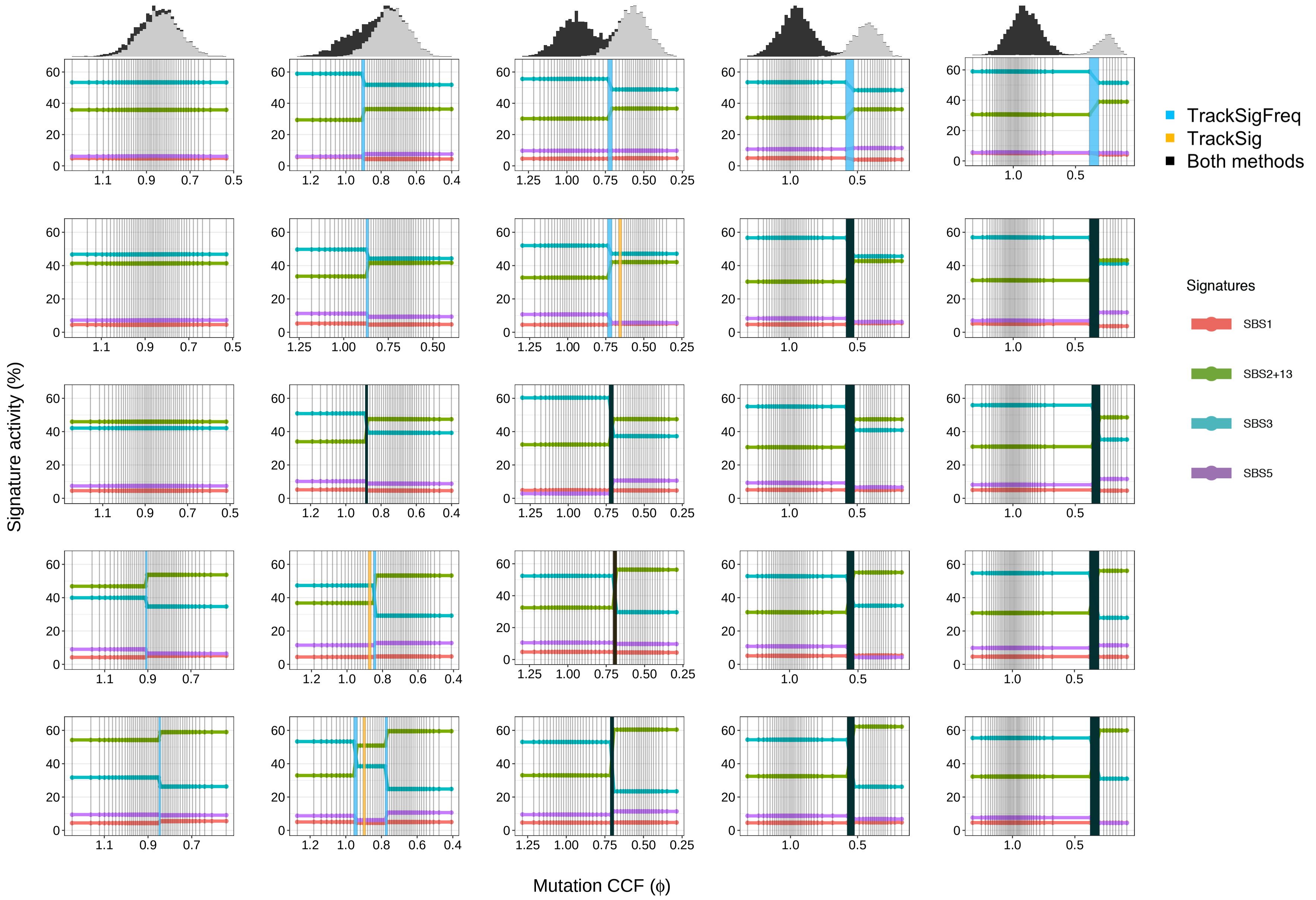

Fig. 2. *Example simulation scenarios.* Each plot shows changepoints found by TrackSig (yellow vertical line) and TrackSigFreq (blue vertical line) in a variety of two-cluster scenarios. For each plot, horizontal lines represent activity trajectories of mutational signatures (in %) as a function of decreasing CCF (x-axis). The clonal cluster (black histogram, top) always has $\phi_k = 1$. In each column (left to right), all plots show a scenario, where the subclonal cluster (grey histogram, top) has $\phi_k = \{0.85, 0.68, 0.52, 0.36, 0.2\}$, correspondingly. In each row (top to bottom), all plots show a scenario, where the signature activities changed by $\delta = \{5, 11, 18, 24, 30\}$ percent, correspondingly. Vertical grey lines indicate timepoints, the density of which increases with increasing mutation VAF density.

Next, we quantitatively compare TrackSigFreq to TrackSig and SciClone,[1] a popular method for subclonal reconstruction which uses only mutation VAF. In these simulations, SciClone is chosen to provide an upper bound on performance, because the simulations precisely match the distributional assumptions of SciClone and we are performing these simulations at high depth. We have previously reported that compared to SciClone, TrackSig has increased sensitivity at lower depths and more robustness to model misspecification error.[2] We generated $n = 1000$ simulations of three types. These simulations are similar to the example in Fig. 2, but the properties of the mutation clusters have been randomized as opposed to manually fixed. The details of these simulations are provided below.

Choice of signatures Simulations were generated with four active signatures selected as described in Ref. 2 : S1, S5 and two other signatures, denoted A1 and A2, which are uniformly sampled from the PCAWG[10] signature set. Signatures S1 and S5 were included in all simulations because these signatures are present in all real samples in PCAWG. In these simulations, we set normal copy number to two, mutant copy number to one, and purity to one. Each simulation had 5000 mutations in total and we generated $n = 1000$ simulations of each type. For every clonal cluster, signature activities were uniformly randomly sampled such that they sum to 1:

$$\begin{aligned} &S1_{activity} \in [0.03, 0.1] && A1_{activity} \in [0.4, 0.7] \\ &S5_{activity} \in [0.05, 0.15] && A2_{activity} = 1 - (S1_{activity} + S5_{activity} + A1_{activity}) \end{aligned}$$

Sampling mutation types Mutation types were sampled from a multinomial distribution of signatures in each cluster (clonal or subclonal), proportional to the number of mutations in that cluster. For each mutation i, read depth d_i was sampled according to $d_i \sim \text{Poisson}(\lambda = 100)$. The probability of a variant allele is $p_i = \frac{m_i}{n_i}\phi_k$, where ϕ_k is the CCF of the cluster k such that $z_i = k$ and m_i and n_i are the mutant and total copy number state at the locus as before. Variant counts v_i for a mutation i were sampled as $v_i \sim \text{Binomial}(d_i, p_i)$.

One-cluster simulations Only a clonal cluster is simulated, with activities as described above and $\phi_1 = 1$.

Two-cluster simulations A clonal cluster is simulated, with activities as described above and $\phi_1 = 1$. A subclonal cluster is simulated with ϕ_2 sampled from Uniform$(0.1, 0.4)$. Signature activities of the subclone are sampled on the same range as the clonal population for S1 and S5, while signature activities of A1 and A2 are sampled over a slightly larger range than above. This is to allow the change in signature exposure between clone and subclone to range between 0% and 50%.
Subclonal signature activities again must sum to 1:

$$\begin{aligned} &S1_{activity} \in [0.03, 0.1] && S5_{activity} \in [0.05, 0.15] \\ &A1_{activity} \in [0.2, 0.7] && A2_{activity} = 1 - (S1_{activity} + S5_{activity} + A1_{activity}) \end{aligned}$$

Three-cluster simulations A clonal cluster is simulated as described above with $\phi_1 = 1$. A subclonal cluster is simulated with ϕ_3 sampled from Uniform$(0.1, 0.4)$. A second subclonal cluster is simulated with ϕ_2 sampled from Uniform$(\phi_3,\ \phi_1 - \phi_3)$. This places the subclones such that $1.0 = \phi_1 > \phi_2 \geq \phi_3 \geq 0.1$. Signature activities of both subclones are sampled in the same way as in the two-cluster simulations.

Tables 2a and 2b show that both TrackSig and TrackSigFreq have similar behaviour when there is no changepoint (i.e. one-cluster simulations). TrackSigFreq showed higher sensitivity compared to TrackSig on the two- and three-cluster simulations. This can be explained by the design of the simulation-generating procedure. While the number of changepoints may be the same for a given simulation type, the nature of these changepoints can vary widely. A changepoint with a small change in signature could be missed by TrackSig, but found by TrackSigFreq if there is a large enough change in VAF density that also gives evidence of the changepoint's presence. The probability of such a changepoint being generated increases with the number of changepoints, which could explain the sharp decrease in accuracy with increasing number of changepoints seen in Table 2b, but not as strongly in Table 2a.

TrackSigFreq achieves the SciClone upper bound (Table 2c), demonstrating that neither the addition of the mutation type model, nor the sampled approximation to the VAF distribution, have a detrimental impact on TrackSigFreq's inference. In this simulation regime, TrackSigFreq attains a large improvement over TrackSig, showing that the addition of the VAF model can improve areas where we previously reported shortcomings with TrackSig.[2] We note that we expect that TrackSigFreq's performance will match TrackSig's in the regimes where SciClone performs poorly.[2]

Table 2. Simulation results for TrackSigFreq (2a, left), TrackSig (2b, middle), and SciClone with Beta-binomial mixture model (2c, right). Number of predicted changepoints versus number of true changepoints. Each cell shows the percentage of simulations which estimated certain number of changepoints (normalized within each column).

# predicted cp	*a. TrackSigFreq*			*b. TrackSig*			*c. SciClone*		
	# true changepoints			# true changepoints			# true changepoints		
	0	1	2	0	1	2	0	1	2
0	0.996	0	0	0.999	0.369	0.253	1.0	0	0
1	0.004	0.996	0.198	0.001	0.631	0.482	0	1.0	0.199
2	0	0.004	0.786	0	0	0.255	0	0	0.801
3	0	0	0.016	0	0	0.010	0	0	0

4. Discussion

Here we present TrackSigFreq, an extension of TrackSig,[2] a recent method for reconstructing evolutionary trajectories of mutational signatures in cancer. We have previously argued that no current evolutionary model explains all tumour VAF distributions.[2] TrackSig makes no parametric assumptions about CCF distributions, which can be hard to accurately reconstruct from single bulk sample data,[8] it simply searches for changepoints in the signature activity. However, by construction, TrackSig will not be able to identify subclones that do not differ in their mutational signature activities.

In contrast, TrackSigFreq is closer to other subclonal reconstruction methods[1,15,9] in that it assumes that the underlying mutation CCF distribution consists of a small number of delta functions, one in each segment. When scoring a segment, it uses our sampled approximation of the marginal likelihood, thus integrating over its uncertainty in the location of that delta function within the segment. We have shown that the incorporation of this parametric VAF model makes TrackSigFreq more sensitive to subtle changepoints than TrackSig. We propose that scoring segments using marginal likelihood rather than doing maximum likelihood estimation of CCF cluster parameters makes TrackSigFreq more robust to model misspecification errors in our parametric assumptions about VAF distributions. Thus, in TrackSigFreq, timeline segmentation is performed by jointly using mutational signatures and mutation VAFs. We demonstrate the improved performance of TrackSigFreq compared to TrackSig on simulated data in scenarios with multiple populations but modest signature changes between them.

A closely related approach, Clonesig, was recently introduced at European Conference on Computational Biology (ECCB).[16] Based on the abstract, it appears that, when published, Clonesig will use a similar implied probabilistic model as TrackSigFreq. However, Clonesig fits this model using EM,[12] which is not guaranteed to find a global optimum and has a slow convergence rate.[17] Because TrackSigFreq uses PELT,[11] it is guaranteed to find an optimal solution in subquadratic, and sometimes linear, time.

One possible extension of TrackSigFreq which we have not considered, is reweighting the contributions of individual mutations to the VAF-based likelihood, $\mathcal{L}_{VAF}$, according to their sequencing depths d_i. Currently all mutations receive equal weight.

By using optimal segmentation to reconstruct evolutionary trajectories in cancer based on both mutational signatures and clonal composition, TrackSigFreq can identify multiple populations within it even if no signature change has occurred.

Code availability The software will be available as an R package and can be currently accessed in the GitHub repository: `https://github.com/morrislab/TrackSigFreq`.

Acknowledgements QM is supported by an Associate Investigator award from Ontario Institute for Cancer Research and holds a Canada CIFAR AI chair. This research was supported by a subgrant from a Genome Canada award to C3G through Genome Quebec. This research was partially supported by a National Cancer Institutes Cancer Center Support Grant (P30 CA008748, Thompson).

References

1. C. A. Miller, B. S. White, N. D. Dees, M. Griffith, J. S. Welch, O. L. Griffith, R. Vij, M. H. Tomasson, T. A. Graubert, M. J. Walter *et al.*, Sciclone: inferring clonal architecture and tracking the spatial and temporal patterns of tumor evolution, *PLoS Computational Biology* **10** (2014).
2. Y. Rubanova, R. Shi, C. Harrigan, R. Li, J. Wintersinger, N. Sahin, A. Deshwar, Q. Morris, PCAWG-11 *et al.*, TrackSig: reconstructing evolutionary trajectories of mutations in cancer,

BioRxiv (2019).

3. L. B. Alexandrov, S. Nik-Zainal, D. C. Wedge, S. A. Aparicio, S. Behjati, A. V. Biankin, G. R. Bignell, N. Bolli, A. Borg, A.-L. Børresen-Dale *et al.*, Signatures of mutational processes in human cancer, *Nature* **500**, p. 415 (2013).
4. L. B. Alexandrov, S. Nik-Zainal, D. C. Wedge, P. J. Campbell and M. R. Stratton, Deciphering signatures of mutational processes operative in human cancer, *Cell reports* **3**, 246 (2013).
5. L. B. Alexandrov, Y. S. Ju, K. Haase, P. Van Loo, I. Martincorena, S. Nik-Zainal, Y. Totoki, A. Fujimoto, H. Nakagawa, T. Shibata *et al.*, Mutational signatures associated with tobacco smoking in human cancer, *Science* **354**, 618 (2016).
6. S. Behjati, G. Gundem, D. C. Wedge, N. D. Roberts, P. S. Tarpey, S. L. Cooke, P. Van Loo, L. B. Alexandrov, M. Ramakrishna, H. Davies *et al.*, Mutational signatures of ionizing radiation in second malignancies, *Nature Communications* **7**, p. 12605 (2016).
7. A. McPherson, A. Roth, E. Laks, T. Masud, A. Bashashati, A. W. Zhang, G. Ha, J. Biele, D. Yap, A. Wan *et al.*, Divergent modes of clonal spread and intraperitoneal mixing in high-grade serous ovarian cancer, *Nature Genetics* **48**, p. 758 (2016).
8. M. Griffith, C. A. Miller, O. L. Griffith, K. Krysiak, Z. L. Skidmore, A. Ramu, J. R. Walker, H. X. Dang, L. Trani, D. E. Larson *et al.*, Optimizing cancer genome sequencing and analysis, *Cell systems* **1**, 210 (2015).
9. A. G. Deshwar, S. Vembu, C. K. Yung, G. H. Jang, L. Stein and Q. Morris, PhyloWGS: reconstructing subclonal composition and evolution from whole-genome sequencing of tumors, *Genome Biology* **16**, p. 35 (2015).
10. S. C. Dentro, I. Leshchiner, K. Haase, M. Tarabichi, J. Wintersinger, A. G. Deshwar, K. Yu, Y. Rubanova, G. Macintyre, I. Vazquez-Garcia *et al.*, Portraits of genetic intra-tumour heterogeneity and subclonal selection across cancer types, *BioRxiv* (2018).
11. R. Killick, P. Fearnhead and I. A. Eckley, Optimal detection of changepoints with a linear computational cost, *Journal of the American Statistical Association* **107**, 1590 (2012).
12. A. P. Dempster, N. M. Laird and D. B. Rubin, Maximum likelihood from incomplete data via the EM algorithm, *Journal of the Royal Statistical Society: Series B (Methodological)* **39**, 1 (1977).
13. S. A. Forbes, D. Beare, H. Boutselakis, S. Bamford, N. Bindal, J. Tate, C. G. Cole, S. Ward, E. Dawson, L. Ponting, R. Stefancsik, B. Harsha, C. Y. Kok, M. Jia, H. Jubb, Z. Sondka, S. Thompson, T. De and P. J. Campbell, COSMIC: somatic cancer genetics at high-resolution, **45**, D777.
14. L. B. Alexandrov, J. Kim, N. J. Haradhvala, M. N. Huang, A. W. Ng, Y. Wu, A. Boot, K. R. Covington, D. A. Gordenin, E. N. Bergstrom, S. M. A. Islam, N. Lopez-Bigas, L. J. Klimczak, J. R. McPherson, S. Morganella, R. Sabarinathan, D. A. Wheeler, V. Mustonen, t. P. M. S. W. Group, G. Getz, S. G. Rozen and M. R. Stratton, The repertoire of mutational signatures in human cancer, p. 322859.
15. A. Roth, J. Khattra, D. Yap, A. Wan, E. Laks, J. Biele, G. Ha, S. Aparicio, A. Bouchard-Côté and S. P. Shah, Pyclone: statistical inference of clonal population structure in cancer, *Nature methods* **11**, p. 396 (2014).
16. J. Abecassis, F. Reyal and J.-P. Vert, Clonesig: Joint inference of intra-tumor heterogeneity and signature deconvolution in tumor bulk sequencing data, in *Proc. ISMB/ECCB*, 2019.
17. C. J. Wu *et al.*, On the convergence properties of the EM algorithm, *The Annals of statistics* **11**, 95 (1983).