Identification of Subclonal Drivers and Copy-Number Variants from Bulk and Single-Cell DNA Sequencing of Tumors

Ashley Mae Conard
Advisor: Dr. Benjamin Raphael
Computer Science Department and
Center for Computational and Molecular Biology Program
Brown University

May 1, 2019

Contents

1 Cancer Genomics Background for Two Projects

Cancer is a disease that can be characterized by a heterogeneous mixture of tumor cells, each with their own mutations, a concept called *intratumor heterogeneity*. One of the greatest challenges is to identify which mutations provide a fitness benefit to the tumor cells, so called *driver mutations*. Driver mutations help these tumor cells proliferate and sometimes grow to resist treatment. Most tumor sequencing studies use *bulk* DNA sequencing, resulting in a mixture of hundreds or thousands of sequenced tumor cells, to identify these driver mutations. Importantly, these driver mutations are thought to be clonal, i.e. present in all tumor cells. Only recently the existence of *subclonal driver mutations* has been shown, i.e. mutations that only affect a subset of tumor cells [32]. Not accounting for the possibility of subclonal driver mutations may compromise the efficacy of targeted cancer therapies [26, 37, 60, 67]. Furthermore, distinct somatic events in different tumor subclones can affect the same gene or pathway, suggesting constraints to tumor evolution (i.e. *convergent evolution*). In Section 2, we present a pipeline to identify subclonal driver mutations. We discover several novel subclonal driver events across 9 cancer types.

In order to truly know which mutations influence each other, we must look at individual cells, because mutations occur in each individual cell. Tumors form through cellular acquisition of fitness enhancing mutations over the lifetime of an individual. *Copy-number variants* (CNVs) are a type of mutation affecting consecutive positions in the genome, and are pervasive across many cancer types [10, 16, 78]. Such mutations can lead to gene dosage effects in oncogenes and tumor suppressor genes. Different subpopulations of cells, or *clones*, have different complements of CNVs. CNVs occur in individual cells, and thus should be identified at single-cell resolution. Single-cell DNA sequencing (scDNA) is a useful technique to identify the complement of CNVs in individual cells, thus also enabling us to infer the clones directly. In Section 3, we present a method to infer the complement of CNVs for each clone, while jointly identifying clonal composition of a tumor from scDNA data. Inferring tumor composition provides an opportunity to improve our understanding of intratumor heterogeneity, clonal expansion and metastatic dissemination [44, 71].

2 Method to Identify Subclonal Driver Events Across 9 Cancer Types

GOAL: Develop a method to identify subclonal driver events.

2.1 Introduction: Characterizing Mutation Events

When a gene acquires a mutation, that can be described as a mutation *event*. Only a subset of these events contribute to the progression of the tumor, so called *drivers* events. Mutated *driver genes* are those that confer a selective proliferation advantage for the tumor cell, and are difficult to identify amidst the large number of *passenger* mutations. In an average tumor, there are about two to eight driver mutations, and anywhere from 10 to 1000 passenger mutations [67]. It is thus, very difficult to identify the somatic mutations which are drivers, and which mutations are passengers.

We can examine the deleteriousness of a mutation in a gene to distinguish between passenger and driver events. To do so, we examine the protein product encoded by the gene in question. Functional alterations in the protein can result from activating or inactivating mutations in the genome. Vogelstein et al. 2013 [67] proposes that the majority of activating mutations form from *missense* mutations. A missense mutation is a *single nucleotide variant* (SNV), or point mutation, which alters the amino acid in the encoded protein. Activating mutations that recur at one position (i.e. locus) in the genome are called *hotspot* mutations [56]. Vogelstein defines a gene which has acquired a hotspot mutation, an *oncogene* (OG). We also adopt these definitions for activating mutations, hotspots, and OGs in this work.

Inactivating mutations cause the encoded protein to acquire a loss of function mutation. Vogelstein et al. 2013 [67] proposes that the majority of inactivating mutations form by three types of protein-truncating mutations, which can span the full length of the gene. When protein-truncating mutations occur in a gene, this gene is a *tumor suppressor gene* (TSG).

Specific types of protein-truncating mutations include *nonsense*, where one amino acid changes to a stop codon; *non-stop*, where a mutation deletes the stop codon; and *splice-site*, where there is an insertion, deletion or change in the number of nucleotides in a particular site, at which splicing occurs. These three types of mutations alter or terminate the encoded protein's function, and are hence said to be *driver mutations*. As a result, these driver mutations render the gene to be a *driver gene*.

Note the strong distinction between OGs and TSGs. OGs drive abnormal cell growth caused by mutations that either increase gene expression or cause uncontrolled activity of the oncogene-encoded proteins. TSGs control cell growth in the opposite manner, generally by inhibiting cell growth. In many tumors, TSGs become inactivated or lost, therefore eliminating negative regulators of cell growth. Hence, TSGs contribute to the abnormal proliferation of tumor cells [22].

Both OGs and TSGs describe driver genes. A single driver gene can have both driver and passenger mutations. The challenge is to distinguish which mutations could cause the gene to be considered a driver gene. For example, as Vogelstein [67] points out, the driver gene APC has driver gene mutations which truncate the encoded protein within its N-terminal 1600 amino acids and are therefore tumor suppressor mutations. APC also contains missense mutations and C-terminal 1200 amino acids found throughout the APC gene, which are passenger gene mutations [67].

Most cancer research is performed under the assumption that the majority of driver mutations are present in all tumor cells, which are known as *clonal* mutations [67]. However, the intratumor heterogeneous nature of cancer also suggests that there are mutations in only a subset of the cells that give fitness benefits to the tumor. These types of mutations are called *subclonal* mutations. Little is currently known about subclonal driver mutations, and their importance within the context of intratumor heterogeneity. Dr. Marco Gerlinger points out that intratumor heterogeneity presents significant difficulties for personalized medicine and development of biomarkers [20]. Dr. Bert Vogelstein expresses that understanding the role of subclonal driver genes is of great clinical importance [67]. These subpopulations of cells with different mutations develop and compete with one another, forming specific genetic signatures that drive tumor development [23, 51] and treatment options [21]. These subpopulations may therefore provide insights into the complexities of intratumor heterogeneity, and reasons for drug resistance.

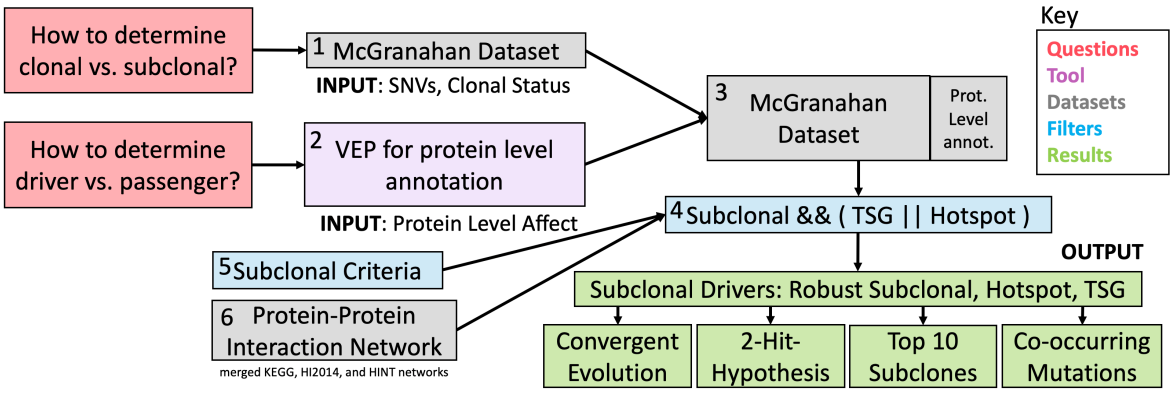


Figure 1: Pipeline illustrating the central questions and method to identify subclonal driver events.

Past research has shown evidence for subclonal driver events, such as Gerlinger et al. 2012 [20], who identify several instances of subclonal driver events within one tumor. Specifically, they show that SETD2, PTEN, and KDM5C genes experienced multiple different and spatially separated inactivating mutations within a renal carcinoma tumor, which points to instances of convergent evolution. Furthermore, McGranahan et al. 2015 [42], identifies subclonal frequencies of driver events across 9 different cancer types. They identify 32 driver genes, of which several could lead to subclonal expansions. The analysis in this paper expands on the work performed by McGranahan et al. 2015 (see Section 2.2.1).

Thus, the goal of this work is to accurately identify and characterize the driver mutation events which are subclonal. Said differently, we aim to identify mutation events occurring in a subset of tumor cells which contribute to functional changes in protein function. Mutations which occur in only a subset of the cells within a tumor have the potential to accumulate deleterious mutations, and yet go unidentified and ignored due to their small frequency within the tumor, and their previously presumed innocuous nature. These types of mutations can cause problems for patients such as drug resistance, relapse, and metastatic tumor growth [60, 37, 26]. Examining the driver mutations found in only a subset of the tumor enables us to more accurately characterize the heterogeneity within tumor cells. Moreover, this more robust characterization of tumors facilitates better personalized therapeutic treatment.

2.2 Method: Novel Method to Identify Subclonal Driver Mutation Events

We accomplish our goal to accurately identify and characterize subclonal driver mutation events by studying the deleteriousness of a mutation in a given gene, and examining the protein product encoded by the gene in question. Figure 1 illustrates our pipeline, presenting both the central questions and method used to identify subclonal driver events. First, we ask how to find the small amount of driver mutations amidst potentially thousands of passenger mutations. In tandem, we ask how to discern between the driver SNV mutations that are clonal and subclonal. These two questions prompt two kinds of input, (1) data which provides both the list of SNVs and their clonal statuses (Section 2.2.1), and (2) their encoded

protein level mutation information (Section 2.2.4). (3) By then merging these two types of data, we can resolve which SNVs cause functional changes in a cell. Using stringent criteria (4,5,6) for how to determine subclonal driver mutation events (Section 2.2.5), we are able to identify subclonal driver events, some of which are novel, and not reported by McGranahan et al. 2015 [42], but supported by the literature as being deleterious cancer mutations (see Section 2.3).

2.2.1 Published Data from 9 Cancer Types

Tumor cells can form from large regional mutations in the genome (affecting multiple genes), and from SNVs within a gene. To focus on SNVs, we use the 9 cancer type dataset from McGranahan et al. 2015 [42], which comprises of a subset of The Cancer Genome Atlas (TCGA) data from breast cancer (BRCA), urothelial bladder cancer (BLCA), lung adenocarcinoma (LUAD), lung squamous cell carcinoma (LUSC), cutaneous melanoma (SKCM), colon adenocarcinoma (COAD), glioblastoma multiforme (GBM), head and neck squamous cell carcinoma (HNSC), and clear cell kidney carcinoma (KIRC). There are 2694 patients and a total of 516,672 somatic mutations [42].

In the analysis performed by McGranahan et al. 2015 [42], they analyze the aforementioned 9 cancer types to determine subclonal frequencies of driver events, and to identify patterns of tumor evolution across the 9 cancer types. They additionally identify a few mutation events which could lead to subclonal expansions.

2.2.2 Our Method Provides Output to Characterize Subclonal Drivers.

Our analysis builds off of the annotated subset of TCGA by McGranahan et al. 2015 [42], by incorporating protein level information, known hotspot genes, and TSGs to identify driver events occurring in only a fraction of cancer cells, so called *subclonal drivers*. Our method highlighted in Figure 1 outputs identified subclonal drivers and divides them into those which are subclonal TSGs, subclonal hotspot mutations (called oncogenes at the gene level), and robustly subclonal hotspot mutations and TGS (i.e. mutations are only present in at most 80% of the tumor cells). Our method then identifies which of these subclonal drivers are instances of either convergent evolution, 2-hit-hypothesis (both gene copies must be targeted by a truncating mutation to cause tumor cells to develop), co-occurring mutations in the same cells, and the top subclones across 9 different cancer types. Additionally, all SNVs, their protein level annotation, clonal status, and any label aforementioned is output into a .csv file for the user.

2.2.3 Determining Clonal and Subclonal Events

Step (1) illustrated in Figure 1 is to determine clonal and subclonal mutations. The clonal status of genetic mutations can be determined by identifying the fraction of cancer cells which harbor the mutation in question, that is to calculate the cancer cell fraction (CCF), see Eq. 1. In the [42] dataset, the authors integrate ASCAT derived copy-number profiles (i.e. the complement of detected CNVs across the genome), δ ploidy (i.e. average genome copy-number) and ρ purity estimates (i.e. the proportion of tumor cells in the sample) with the SNVs and associated variant allele frequencies (VAF - the proportion of reads

at a locus which contain the SNV) as reported by TCGA [63]. Mutations are classified as clonal if the upper limit of the 95% confidence interval for the CCF (Eq. 1) is greater or equal to 1, and subclonal otherwise. We add a more stringent classification by deeming mutations as robustly subclonal if the upper limit of the 95% confidence interval falls below 80%, signifying that those mutations are only present in at most 80% of the tumor cells.

$$\text{CCF} = \frac{\text{VAF} \times (\rho \times \delta + (2 \times (1 - \rho)))}{\rho} \tag{1}$$

The CCF gives a measure of the proportion of cells with the mutation in question. This CCF calculation assumes that there is one mutated chromosome copy within the subset of tumor cells, and that there is the same ploidy for all tumor cells.

2.2.4 Classification of Functional Mutations

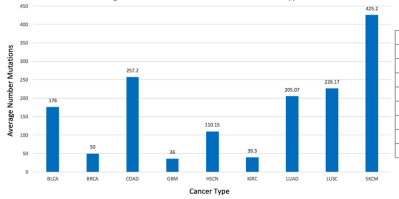
Step (2) is to obtain the protein level annotations for each SNV extracted from the TCGA dataset. For this we use the Ensembl Variant Effect Predictor (VEP), which analyzes, annotates, and prioritizes genomic variants in both coding and non-coding regions of the genome [43]. VEP provides gene and transcript-related information, including the type of mutation, the associated amino acid change, and protein sequence position for every SNV, which are all used in this analysis. Specifically for TSGs, the types of mutations we consider are missense, nonsense, nonstop, and splice-site. To identify hotspot mutations we use the protein sequence position and associated amino acid change. In step (3), protein level annotation is then added for each mutation in the dataset.

Given 2694 patients, 516,672 somatic mutations, their clonal statuses, and their encoded protein alterations, in step (4), we provide a reliable way to identify the more frequently mutated positions (so called hotspots) and genes (tumor suppressors), across all cancer types and patients. In order for a gene to be classified as a TSG, we adopt the "20/20 rule" by Vogelstein et al. 2013 on the basis of mutation patterns rather than frequencies [67]. This rule states that genes with $\geq$20% truncating (inactivating) mutations are tumor suppressor genes, and can be used to identify even the "modestly mutated driver genes" [67]. Inactivating mutations denote those that are non-stop, splice-site, or nonsense. The score of each TSG is simply the sum of all inactivating mutations, divided by all mutations for that gene across all patients (inactivating and activating) as shown in Eq. 2.

$$\text{TSG} = \frac{\text{non-stop} + \text{splice-site} + \text{nonsense}}{\text{missense} + \text{non-stop} + \text{splice-site} + \text{nonsense}} \tag{2}$$

We imposed an additional constraint that there should be at least 20 total mutations across patients for a gene to be considered. We use these two constraints to identify a list of TSGs from the McGranahan dataset, to which we then union with the list of 71 TSG

A.



B.

Cancer Type	Total # Samples	¬ Hypermutators	Percent Total
BLCA	122	130	93.85
BRCA	887	903	98.23
COAD	155	190	81.58
GBM	249	250	99.6
HNSC	284	289	98.27
KIRC	303	303	100
LUAD	222	257	86.38
LUSC	118	131	90.08
SKCM	172	241	71.37

C.

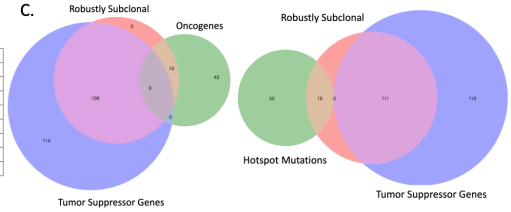


Figure 2: A. Histogram of the average number of mutations across the 9 cancer types used in this analysis. B. Table of all tumor samples with and without hypermutator samples. C. (left) Numbers of subclonal drivers at gene level, and (right) numbers of subclonal drivers at event level, meaning that oncogenes are labeled by their hotspot mutation.

provided from Vogelstein et al. 2013 [67], giving a total of 107 TSGs. Between our list of 56 identified TSGs and Vogelstein's list, there is significant overlap, with 20 TSGs in common.

In order for a gene to be classified as having a hotspot mutation, we obtained a list of 159 hotspot mutations using MutationAligner, which is an online resource that uses multiple sequence analysis to locate protein domain hotspots [18]. Hotspot mutations are identified by summing the number of missense mutations found across comparable residues of domain-containing genes. The MutationAligner database contains information on ∼500,000 missense mutations from more than 5000 patients across 22 cancer types. There are two main criteria for counting the missense mutations at each position: at least two missense mutations are found at each position in the alignment, and at least three-quarters of the aligned sequence domains have no gaps at the analyzed positions [18]. Then, those filtered positions are used for further analysis.

2.2.5 Criteria for Subclonal Driver Events

In order to accurately determine subclonal driver mutations, in step (5) we apply the following criteria as a filter to every SNV: the mutation is one of 107 TSGs or 159 hotspot mutations, the upper limit of the 95% confidence interval for the CCF is below 1, and only genes with VAFs below 0.55 are considered to have valid subclonal mutations. Furthermore, to avoid hypermutator samples (mostly found in cutaneous melanoma, see Figure 2 A), we only analyze samples with less than 400 mutations. Without hypermutator samples, we analyze 2512 out of 2693 patients, which is about 93% of the original amount of patients. See Figure 2 B for details by cancer type.

We imposed an additional criterion to the CCF calculation to identify *robustly subclonal* mutations: for each mutation in question, the upper limit of the 95% confidence interval falls below 80%, signifying that those mutations are only present in at most 80% of the tumor cells. The addition of protein level information to identify hotspot and tumor suppressor mutations permits a focused analysis of which point and truncating mutations cause the encoded protein function to change. We considered these types of mutations to be true potential driver mutations for further downstream analysis.

In step (6) we integrate a large protein-protein interaction network to scan for instances of convergent evolution across 9 cancer types between interacting genes. To determine which genes interact, we created a merged network between KEGG, HI2014, and HINT to obtain a total of 760,856 pairs of interacting genes. We used this large network to identify patients

	Number Subclones		
	Subclonal Hotspots	Subclonal Tumor Suppressors	Robustly Subclonal
Total	69	229	130
BRCA	10	29	13
BLCA	3	21	12
COAD	10	23	6
GBM	14	15	15
HNSC	12	37	27
KIRC	1	44	22
LUAD	7	33	21
LUSC	4	13	7
SKCM	8	14	7

Top 10 Subclones					
Subclonal Hotspots	# Samples	Subclonal Tumor Suppressors	# Samples	Robustly Subclonal	# Samples
PIK3CA_545	16	NBPF1	33	NBPF1	26
KRAS_12	8	APC	19	KMT2C	8
NRAS_61	7	PBRM1	15	SETD2	7
PIK3CA_542	5	KMT2C	11	PIK3CA_545	6
TP53_273	4	SETD2	10	PBRM1	5
TP53_248	4	PTEN	10	PTEN	5
IDH1_132	3	VHL	8	APC	4
SMAD4_361	2	TP53	8	TP53	4
TP53_179	2	CDH1	6	EP300	3
TP53_245	2	FAT1	6	FAT1	3

Table 1: Left: Number of subclones total and by cancer type. All robustly subclonal mutations are subsets of either subclonal hotspots or subclonal tumor suppressor genes. Right: Top 10 subclonal hotspots, subclonal tumor suppressors, and robustly subclonal hotspots and tumor suppressors.

with subclonal driver mutations that are known to interact.

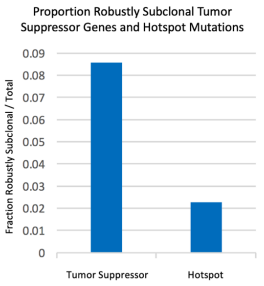
2.3 Results: Our Method Output Recapitulates and Highlights Novel Subclonal Driver Events

The goal of this work is to accurately identify subclonal driver events, which we accomplish by analyzing changes in protein function. In this Section 2.3, we summarize our results and highlight interesting events. In Section 2.3.1 we highlight novel and known instances of convergent evolution across 9 different cancer types. Of these 2512 patients, we identified 266 patients with subclonal driver mutations, based on our classification. Total across 9 cancer types there are 69 subclonal oncogenes/hotspot mutations, 229 subclonal TSG genes, and 130 robustly subclonal events. Figure 2 C presents these results at the (left) gene level and (right) event level. The left shows oncogenes, which overlap with TSGs, whereas the right disambiguates those hotspot mutations in oncogenes to remove overlap with TSGs. For example, a hotspot mutation in PIK3CA 545 and PIK3CA 542 are considered as two separate events, but by collapsing at the gene level, we say that PIK3CA is mutated (i.e. one event). In later analysis, we characterize an activating mutation using the more accurate description: hotspot mutation, rather than oncogene.

Highlighted in Table 1 (left), we identify 69 subclonal hotspot mutation events by considering only hotspot mutations, coupled with our constraints discussed in Section 2.2.5. Table 1 (right) illustrates that TP53 and PIK3CA mutations dominate the top ten hotspot mutations. We similarly identify 229 tumor suppressor mutation events, where we see the largest number of mutations in NBPF1, most of which are robustly subclonal (see Section 2.3.1 for further details). Of the 130 robustly subclonal mutations, which are a subset of subclonal hotspots or subclonal TSGs, NBPF1 has by far the largest number of robustly subclonal mutations inactivating mutations.

Considering all 9 cancer types, the overall proportion of robustly subclonal TSG mutations make up 0.086% of the total number of inactivating TSG mutations (Figure 3 A). The proportion of robust subclonal hotspot mutations make up 0.023% of the total number of activating mutations for these known hotspots (Figure 3 A). Figure 3 C and D show the proportions of robust subclonal, subclonal, and clonal mutations for each tumor each hotspot mutation, and suppressor gene that appeared in our dataset. Importantly, notice in Figure 3 B that NBPF1 is both the most frequent subclonal driver event and is most often robustly subclonal, given all mutation events for that gene.

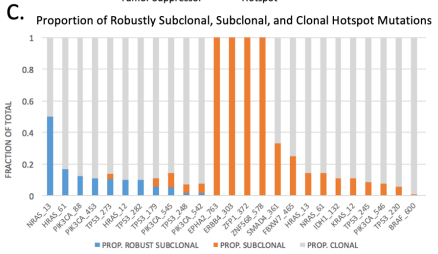
A.



B.

Subclonal Driver	Robustly Subclonal	Subclonal	Clonal	Total # Mutations	Percent Total
NBPF1	26	7	1	34	76
EP300	3	0	9	12	25
SETD2	7	3	20	30	23
KMT2C	8	3	34	45	18
PTEN	5	5	27	37	14
PBRM1	5	10	31	48	10.9
TP53_273	3	1	25	29	10.3
PIK3CA_545	6	10	97	113	5
NRAS_61	0	7	42	49	0

C.



D.

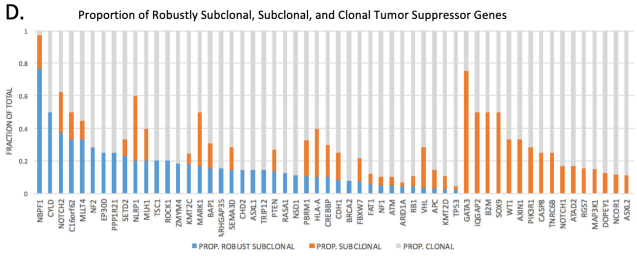


Figure 3: A. Proportion of total robustly subclonal mutations to total clonal mutations for both TSG events and hotspot mutation events. B. Table of subclonal drivers discussed in this text. C. Proportions of robustly subclonal, subclonal, and clonal hotspot mutations. D. Proportions of robustly subclonal, subclonal, and clonal tumor suppressor genes.

2.3.1 Identification of Several Novel Subclonal Events

As aforementioned, the most frequent subclonal driver TSG is NBPF1, comprising of 33 events, of which 26 are robustly subclonal. We identified only 1 clonal driver event, which means that 76.5% of the driver events were robustly subclonal. NBPF1 (Neuroblastoma BreakPoint Family, member 1) is a member of the NBPF gene family, which are predominantly located on duplicated regions of chromosome 1. NBPF genes have been found to be involved in cancer and in brain and developmental disorders [3]. Of the 33 subclonal events, all but one were splice-site events. The exception is a nonsense mutation in a LUAD patient, which has as low tumor purity of 29%. The most frequent position for this splice-site is occurs at position 16918653. There are mutations in the NBPF1 gene in 7/33 BLCA, 1/33 GBM, 11/33 HNSC, 1/33 KIRC, 7/33 LUAD, and 6/33 SKCM samples. Recent reports about this gene point to its importance in several cancer types [2]. For example, NBPF1-expressing colon cancer cells formed fewer colonies than the control cells, signifying that NBPF1 could be suppressing anchorage-independent growth. Nevertheless, little focused research has been done outside neuroblastoma, where NBPF1 exerts growth inhibitory effects by inducing a G1 cell cycle arrest [2].

We identify two co-occurring mutations in colon adenocarcinoma patient COAD-D5-6922. Specifically, COAD-D5-6922 has two nonsense subclonal driver mutations in the interacting genes APC and PIK3R1. APC contains a mutation present in 76% of the cells, and PIK3R1 contains a mutation in 72%. The purity of this sample is 74%. Both mutations are characteristically mutated in colon cancers, especially APC, where loss of this gene is known to play a major role in the molecular and histological changes of colorectal cancers [48, 68, 15] For both genes, the cancer cell fractions (CCFs) are consistent with mutations occurring in the same tumor subclone. This is a strong example of two subclonal mutations co-occurring in the same cells, contributing to the tumor phenotype.

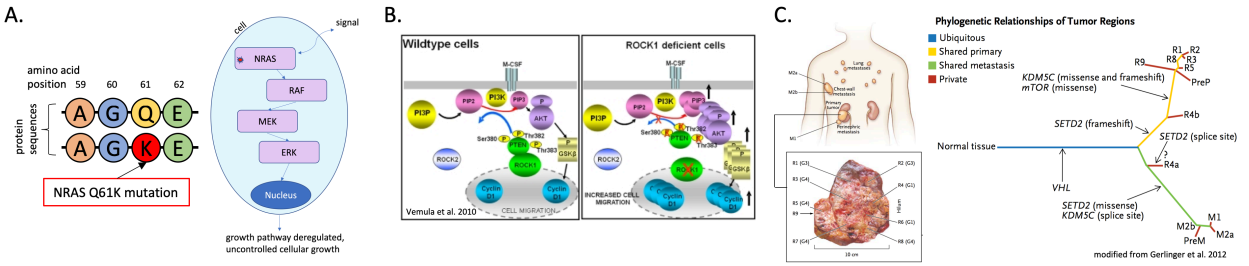


Figure 4: A. (left) NRAS Q61K mutation from Glutamine to Lysine. (right) NRAS mediated pathway modified from Vu et al. 2016 [69] B. Proposed model of ROCK1-mediated PTEN regulation. C. (left) Sites of biopsies and regions collected from nephrectomy and metastasectomy samples, where "G" indicates the tumor grade. (right) Phylogeny of acquired mutations. Branch lengths are proportional to the number of nonsynonymous mutations separating the branching points. Potential driver mutations were acquired by the indicated genes in the branch (arrows). Figure altered from Gerlinger et al. 2012 [20].

Multiple mutations in the same driver gene illustrates the idea that a gene can be targeted more than once by an activating or inactivating mutation. We identify several such events where a gene in targeted twice. Most notably, in patient SKCM-DA-A3F3 afflicted with skin cutaneous melanoma, we identify two missense mutations to the hotspot in the NRAS, in protein sequence position 61, changing a Glutamine to a Lysine (Q61K) as shown in Figure 4 A (left). At a tumor sample purity of 97%, one mutation occurs in 76% of the cancer cells, and the other occurs in 75%. For both mutations, the cancer cell fractions (CCFs) are consistent with mutations occurring in the same tumor subclone. NRAS is a member of the RAS family of proteins and when mutated, deregulates growth pathways leading to uncontrolled cell proliferation. This is highlighted in Figure 4 A (right) modified from Vu et al. 2016 [69]. Q61K mutations to NRAS codon 61 is the most frequent RAS alteration in primary sporadic melanomas with reported frequencies in up to 50% of cases [55]. Targeting this NRAS Q61K mutation in tumor patients is an area of active research [31].

2.3.2 Discovery of Novel and Known Instances of Convergent Evolution

When there are distinct somatic events in different tumor subclones that can affect the same gene or pathway, this suggests constraints to tumor evolution, so called *convergent evolution*. Convergent evolution describes the fact that, within one tumor there are multiple subpopulations which develop similar functions (e.g. by mutating the same gene) even though the most common ancestor did not have that mutation. Two good indicators for convergent evolution used here are to see more than one subclonal driver mutation in the same patient, and to identify which of these subclonal drivers interact. Here we present three instances of convergent evolution missed by McGranahan et al. at both the gene and pathway levels.

Patient HNSC-CV-6955 head and neck squamous cell carcinoma contains two nonsense subclonal driver mutations in FAT1. One occurs in 41% of the cancer cells, and the other occurs in 48%, with a tumor sample purity of 42%. The CCFs for both mutations are consistent with mutations occurring in independent tumor subclones, which then target the same gene independently. FAT1 mutations are common in HNSC and its mutational status is a strong prognostic factor in patients with HPV-negative HNSCC [29].

In glioblastoma patient GBM-28-2513 we identify two subclonal driver mutations that interact in the RhoA-ROCK-PTEN pathway. [62]. ROCK1, which has a nonsense mutation

in 52%, and PTEN, which contains a splice-site mutation in 36% of the cells. For both genes, their CCFs are consistent with mutations occurring in independent tumor subclones. Rho-A-dependent activation of PTEN likely occurs through the Rho-A-dependent kinase ROCK1. Figure 4 B (left) schematically shows that ROCK1 plays an important role to regulate levels of PIP3/AKT through receptor activation of PTEN, thus suppressing inflammatory cell migration [64]. Figure 4 B (right) shows that in ROCK1 deficient cells, ROCK1 cannot regulate PTEN phosphorylation and stability, thus causing increased cell migation. This is a strong example of convergent evolution not identified by McGranahan et al., where the RhoA-ROCK-PTEN pathway is disrupted independently by distinct subclones within the same tumor. McGranahan et al. identify another example of convergent evolution in this tumor sample between PTEN and PIK3R1. We do not identify this relationship because PIK3R1 has a missense mutation but not a hotspot mutation. Therefore, our strict definition for subclonal driver mutations does not include this event.

Lastly, our results identify two kidney renal clear cell carcinoma (KIRC-B0-5399 and KIRC-B4-5835) patients, each with two tumor-suppressor subclonal driver mutations in SETD2, which converge on loss of function. We uniquely identify two distinct mutations in SETD2 for KIRC-B4-5835, where the tumor purity is 0.55. One subclonal driver mutation occurs in 73% of cells, and the other occurs in 27%. The CCFs for both mutations suggest independent subclone formation. We and McGranahan et al. identify the two distinct mutations in SETD2 for KIRC-B0-5399, where the tumor purity is 0.59. One subclonal driver mutation occurs in 19% of the cancer cells, and the other occurs in 40%. Again, the CCFs for both mutations are consistent with mutations appearing in distinct subclones. Gerlinger et al. 2012 also identify distinct mutations which occurred in the SETD2 gene in the adipose capsule of the kidney, highlighted in a phylogeny in Figure 4 C [20]. Each instance of convergent evolution aforementioned points to possible constraints to tumor evolution, which could be exploited for therapeutic treatment in the future.

2.4 Conclusions

We designed a novel method to identify subclonal driver events from SNV mutation data, across 9 cancer types. Our results recapitulate several known driver genes, as well as identify several novel subclonal drivers. We accomplished this by examining those mutations which contribute to changes in protein function. McGranahan et al. used a list of driver genes from Lawrence et al. 2014, which may be why McGranahan et al. missed several subclonal drivers [33]. In the future, we would like to add several features to this method, including identifying subclonal copy-number variants, clonal drivers, and mutually exclusive mutation events. We would also like to draw direct comparisons between cancer types, patients, and their gender and age.

Past researchers have shown that some subclonal driver mutations are associated with negative clinical outcomes [32, 67]. Landau et al. 2013 discovers patterns of clonal evolution which provides an understanding into the adverse clinical outcomes caused by subclonal mutations. He claims that the presence of subclonal mutations increases with treatment, which indicates that previously untargetted subpopulations could continue to proliferate and diversify. Most treatment stategies are biased to detect clonal drivers occurring at high frequency within single cancer samples [67]. Landau points out that it is compulsory to

focus also on subclonal events, to be able to have effective targeted therapy. Our method presented here moves us closer to identifying these subclonal driver events which negatively influence treatment.

3 Inference Model to Identify Copy-Number Variants from Single-Cell DNA Sequencing of Tumors

GOAL: Develop a model to accurately identify copy-number variants in single-cell DNA sequencing data.

3.1 Introduction: Single-Cell DNA Copy-Number Inference

Noninherited mutations acquired over the lifetime of an individual drive tumor progression [16, 51]. *Copy-number variants* (CNVs) are a type of mutation that are pervasive across many cancer types [10, 16, 78]. A cell has a CNV in a genomic region when the number of copies of that region differs from the heterozygous diploid state, i.e. one copy inherited from the mother and one from the father. Gains in a region also amplify any point mutations residing in that region, thus magnifying their potency. Different subpopulations of cells, or *clones*, have different complements of CNVs. In contrast to the commonly used technique of *bulk* DNA sequencing, where a mixture of hundreds or thousands of cells are sequenced, single-cell sequencing DNA (scDNA) is technique which allows us to identify the complement of CNVs in individual cells, thus allowing us to infer the clones directly. Inferring tumor clones provides an opportunity to improve our understanding of intratumor heterogeneity, clonal expansion and metastatic dissemination [44, 71].

Fluctuations in *read count*, i.e. the number of reads mapping to different genomic regions, provide signal to detect CNVs. Higher read count from the diploid state indicates a gain in the number of copies of a region of a chromosome, and lower read count indicates a loss. The complement of detected CNVs across the genome form a cell's *copy-number profile* (CNP), characterized by a vector of integers from 0 to an upper-bound (generally below 10) [16, 78].

In scDNA, the DNA source material is from a single cell and is thus significantly smaller than in bulk, where many cells are sequenced at once. To arrive at measurable quantities of DNA source material, whole-genome amplification (WGA) is a necessary step in scDNA. Currently, this step leads to high amounts of experimental noise due to uneven coverage depth, false-positive errors and false negative sites without coverage depth leading to allelic dropout in which not all copies of a segment are amplified and can occur at 10%-30% of mutation sites [46], and other sequencing and amplification errors [19, 57, 8]. Consequently, these errors lead to non-uniform read counts of genomic segments that have the same number of copies [45], which makes *copy-number calling*, (i.e. identifying CNVs) in scDNA data a challenging task. This problem is particularly pronounced with whole exome sequencing (WES) data, where only the coding part ($\sim$3%) of the genome is sequenced. Furthermore, WES data suffers from inherent noise due to various biases that are introduced in target capture and sequencing phases [1]. In contrast to WES, whole genome sequencing (WGS) of single cells improves the read count uniformity and helps classify non-coding and structural variants, which may extend past the exomes [19].

There are currently few methods which identify CNPs of scDNA data [4, 17, 81, 82, 75], and while many scDNA cancer studies produce WES data [19, 27, 36, 50, 72, 76], current methods use WGS data, and do not claim to extend to WES data. Furthermore, these methods consider cells independently when inferring CNPs, even though it is known that

tumor cells are the result of a common evolutionary process [51], and as such many cells have similar CNPs [12, 16]. To leverage this fact, similar cells can be considered *jointly* to enhance the power of inferring the CNPs (i.e. clones) of a tumor. Dr. Nicholas Navin points out in Chen et al. 2016's single-cell sequencing review [8] that amplification errors are randomly distributed and can be mitigated by detection across multiple single cells. Nevertheless, no current scDNA CNP inference methods consider cells jointly.

3.2 Method: Overview of Proposed Model for Single-Cell DNA Copy-Number Profile Inference

Current approaches to solving this problem of inferring cell CNPs relies on algorithms such as hidden Markov models (HMM) and circular binary segmentation (CBS), which can normalize noisy read count data after single-cell WGA to identify regions which are over or under represented when compared to the diploid genome [41, 52]. Here, we propose a model in which we are given m cells with read counts across the genome, each divided into n *bins*, i.e. defined-length regions of the genome (typically each is about 500 to 50 kilobases (KB)) [81, 17], with the goal of determining the CNP for each single cell by *jointly* considering all cells in the cohort. More formally, given a cell, the observations are the corrected (for GC bias and other errors) read counts $\mathbf{r} = (r_1, \ldots, r_n)$ where each r_j is the corrected read count for bin j. The corresponding CNP $\mathbf{p} = (p_1, \ldots, p_n)$, where p_j denotes the integer copy-number for bin j. We have m such vectors $\mathbf{r}$, which form a corrected read count matrix $R = [r_{ij}] \in \mathbb{R}_{\geq 0}^{m \times n}$ and denotes the corrected read count for cell i in bin j. Its corresponding CNP matrix $P = [p_{ij}] \in \mathbb{N}_{\geq 0}^{m \times n}$ denotes the integer copy-number for cell i in bin j. More formally, we consider the following problem:

scDNA CNP Inference Problem: Given a training set $T = \{(R_1, P_1), \ldots, (R_{|T|}, P_{|T|})\}$ and a number of clones k, find $f : \mathbb{R}^{m \times n} \to \mathbb{N}^{m \times n}$ that minimizes the error over the training data[1], where f satisfies $f(R_t) = P_t$ and for every corrected read count matrix R, $P = f(R)$ has at most k unique rows.

Note that $m \gg k$ for samples of tumor cells, as we assume that a tumor is composed of clones (groups of cells that have the same complement of CNVs). First, beginning in Section 3.3, we present the motivations for solving this problem, and analyze results from current methods. Next, in Section 3.4, we describe a deep learning approach which infers the assignment of cells to clones, and the CNPs for those cells and clones, by considering cells jointly. Our proposed method leverages the known evolution of the tumor in which cells evolve from a common ancestor, and thus share similar CNPs [13, 16, 19].

3.3 Review of Current Methods: Robustness Issues

Current methods for scDNA copy-number inference include Ginkgo [17], AneuFinder [4], nbCNV [81], Poisson-CNV [75], and the method by Zhang et al. 2013 [82]. Importantly, these methods do not consider cells jointly when inferring the CNPs for each cell and thus

[1] explained in Section 3.4

do not infer clones. We analyze results from the currently most commonly used methods AneuFinder and Ginkgo. Note that although Ginkgo and AneuFinder report a dendrogram that relates cells, they do not use the relatedness of cells during CNP inference. There are two versions of AneuFinder and Ginkgo. AneuFinder uses both a hidden Markov model and a change-point [52, 65] approach. Ginkgo has a fixed bin length setting, which fixes the sizes of all bins as specified by the user, and a variable bin length setting in which the bin size is set by the user but can change as determined by Ginkgo.

We first compared the concordance in CNP inference between AneuFinder and Ginkgo using tumor and normal WGS scDNA data from patients with triple-negative breast cancer patients from Gao et al. 2016 [16], where they identified 1-3 predominant clones per tumor. We analyzed 3 patients (T4, T5, and T7) due to their very different CNV patterns, to assess the robustness of Ginkgo and AneuFinder. Patient T5 data comprises of 128 cells (38 normal cells and 90 tumor cells). Two populations (i.e. clones) were identified in the study of this patient, in the primary tumor [16]. Patient T4 data comprises of 54 tumor cells and 3 clones were identified. Patient T7 data comprises of 68 tumor cells, and two clones were identified. Neither T4 nor T7 has matched normal cell data. We then assess the performance of both methods using a simulated dataset ($\sim 1\times$ coverage) comprising of two cells populations, one harboring a very small CNV (either deletion or tandem duplication).

3.3.1 Low Concordance Between Ginkgo and AneuFinder

We compared the inferred cell CNPs produced by AneuFinder [4] and Ginkgo [17] from patients T4, T5, and T7 [16] at a large bin size of 250KB, the closest option in Ginkgo to the 220KB median bin size used in the analysis by Gao et al. 2016 [16]. We chose a normal cell SRR3082324 from patient T5 as AneuFinder's reference, given its low read count variance across the genome, $\sim 0.11\times$ average breadth of coverage across the genome (measure of read coverage uniformity), and overall inferred CNP with little deviation from 2 across all bins when inferring the CNP using Ginkgo.

We can see through Figure 5 B that as CNV events become more complicated (see Figure 5 C), both AneuFinder and Ginkgo have increasingly more difficulty producing commensurate results. This is very apparent for patients T4 and T7, where many of the cells have very complicated events, and For example, results for patient T7 show that Ginkgo (variable bins) and AneuFinder (change-point) achieve less than 30% similarity between their inferred CNPs on 41.4% of the cells at the large bin size of 250KB. We show AneuFinder (change-point) and Ginkgo (variable bins) as these versions produced the best results for these CNP inferences.

Ginkgo infers many small CNVs and also produces copy-number 'peaks', which drastically impact the average ploidy calculation and contribute to incorrect CNP inference (see arrow Figure 5 D. Figure 5 D illustrates several examples of cell integer CNPs, where the dotted circles in A are associated with the corresponding dotted plots in D.

The percent similarity between many of the inferred CNPs is either quite low, or quite high, which could indicate that several CNPs differ by just a constant. If this is the case, considering cells jointly would give more power to detect the true CNPs for all cells. After scaling all inferred CNPs for Ginkgo (variable bins) and AneuFinder (change-point), we determined that although the percent similarity did increase for many cells, there are still significant differences between the results of the two methods for many cells, indicating a

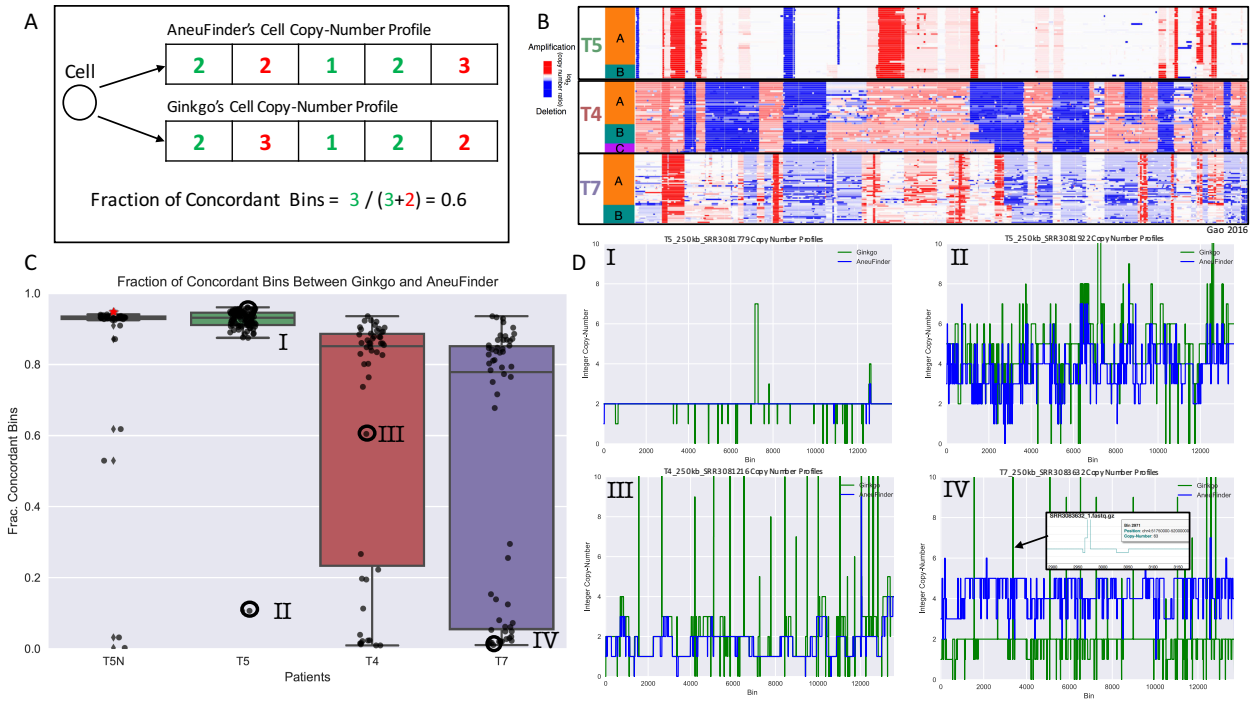


Figure 5: A. Illustration of percent similarity (percent of positions with the same copy-number called) calculation used to compare CNP results between Ginkgo and AneuFinder. B. Percent similarity between cells sequenced in patients T4, T5, T5N (normal cell sample) and T7, between Ginkgo (variable bins) and AneuFinder (change-point). The percentage of cells with a percent similarity below 30% is denoted in black. C. Log2 copy-number ratio plots as reported by Gao et al. 2016 [16]. The log2 copy-number ratio is the absolute ratio to the cell ground (i.e. the median ratio of that cell) and is *not* equivalent to the integer copy-number but is conceptually similar. Blue indicates deletion events, white indicates diploid regions, and red indicates amplification events. Clones identified (clone A, B or C) are indicated on the left. Single cells are plotted on the y-axis, and CNVs are plotted on the x-axis, ordered by genomic location. Note that over half of the tumor cells in patient T5 are mostly diploid, and are not shown in B, but are considered in A. D. Several examples of copy-number profiles which correspond to A. Ginkgo occasionally infers extremely unlikely CNVs for very small segments of the genome, which results in small 'peaks' throughout CNPs, drastically affecting average ploidy calculations.

more complicated disparity between both methods. If cells from the same clone had been considered jointly, this would enable us to infer true CNVs with higher precision.

3.3.2 Low Concordance Between AneuFinder and Ginkgo at Various Bin Sizes for Cells of Normal Clone

As we attempt to identify smaller CNVs by decreasing bin size, AneuFinder and Ginkgo deteriorate in their abilities to robustly infer CNPs. To illustrate this, we compared the inferred CNPs of 35 normal cells from patient T5 by Ginkgo and AneuFinder, against the ground truth diploid CNP. Figure 6 shows the fraction of diploid bins inferred for each cell across both versions of Ginkgo and AneuFinder at various bin sizes. As bin size decreases, the variance in read counts across the genome increases, which significantly hinders these methods from inferring diploid bins for these normal cells. Importantly, even when setting the bin size, Ginkgo's "variable" bin setting means that Ginkgo can change the binsize. For this reason, the plots may be deceptively showing fairly strong fractions of diploid bins, when

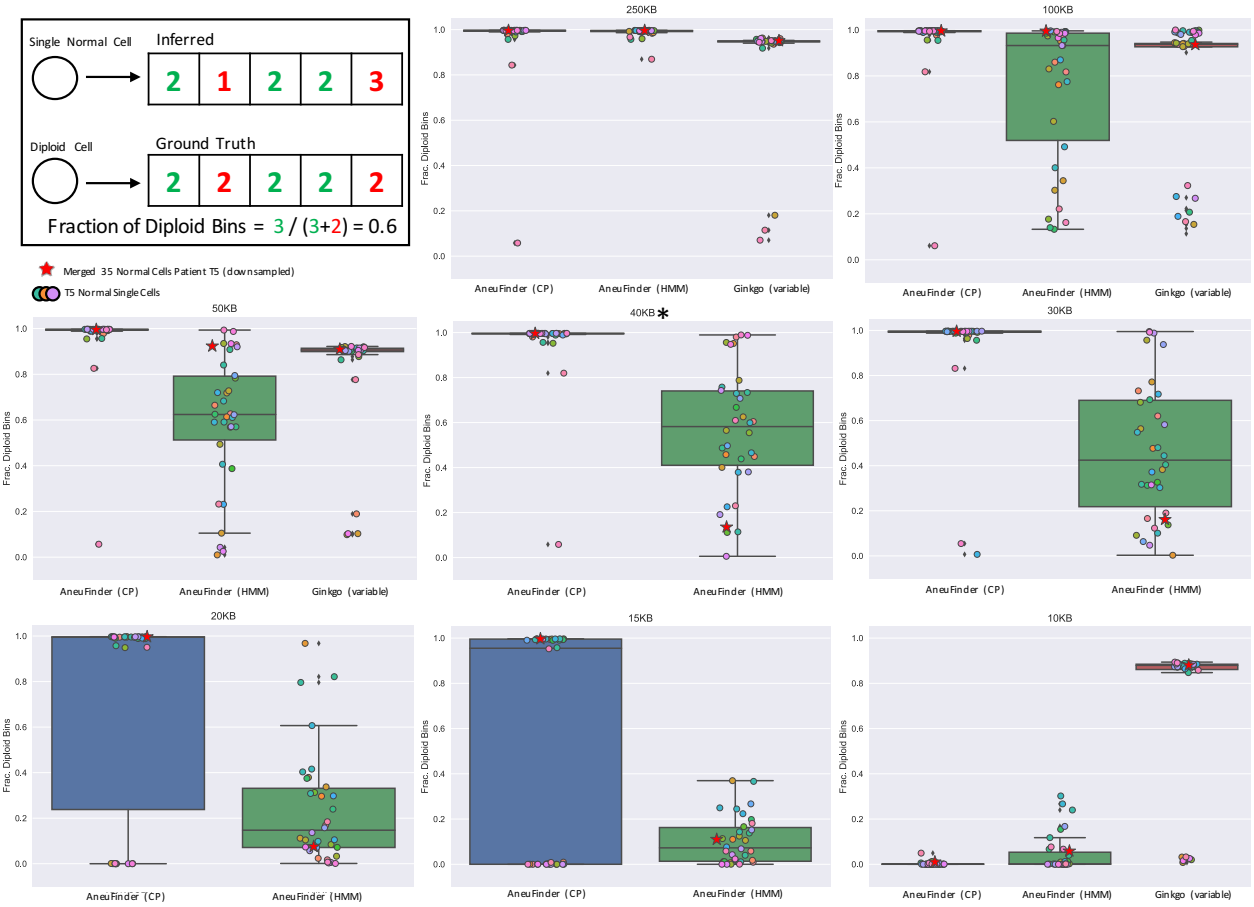


Figure 6: Each panel shows the fraction of diploid bins inferred by AneuFinder and Ginkgo methods for 35 normal cells and two merged cells at bin sizes 250KB to 10KB. There are two versions of both methods: AneuFinder change-point (ADNACOPY), AneuFinder (AHMM), Ginkgo fixed bin size (GFIXCP), and Ginkgo variable bin size (GVARCP). Each point is either one of 35 normal cells (2 outliers removed and one is AneuFinder reference), a merged down sampled cell (red star) to measure an average cell of all normal cells. Percent of cells below diploid bin fraction 0.3 are noted in black for 20KB, 15KB, and 10KB.

in fact Ginkgo is altering the bin sizes, such as for Ginkgo (variable) at 10KB. Note that we used one of the normal cells (SRR3082324) as AneuFinder's reference.

Small CNVs involving a single gene or portion of a gene are not well understood, yet instrumental in disease progression as several studies note across many conditions including cancer, immunodeficiency, neurodevelopmental, and metabolic disorders [5, 54, 61, 79, 73, 25, 28, 40]. For example, Xi et al. 2011 [73] detect and quantitative PCR validate many CNVs between 100 basepairs (BP) and 1000BP in an Acute Myeloid Leukemia patient, including one 40BP event, at 10BP bin resolution. No currently used scDNA CNV inference methods are accurately able to infer CNPs at small bin sizes.

Figure 6 illustrates how the number of diploid bins inferred by both methods decreases as the bin size decreases. Each lightly colored point is one of 35 cells. 2 outlier cells with high sequencing error were removed to provide the best dataset for AneuFinder and Ginkgo to analyze, and cell SRR3082324 was used as AneuFinder's reference. AneuFinder's HMM method begins to deteriorate rapidly in its ability to call diploid bins at 100KB. The ADNACOPY method is overall less sensitive than AHMM, and maintains high fidelity when

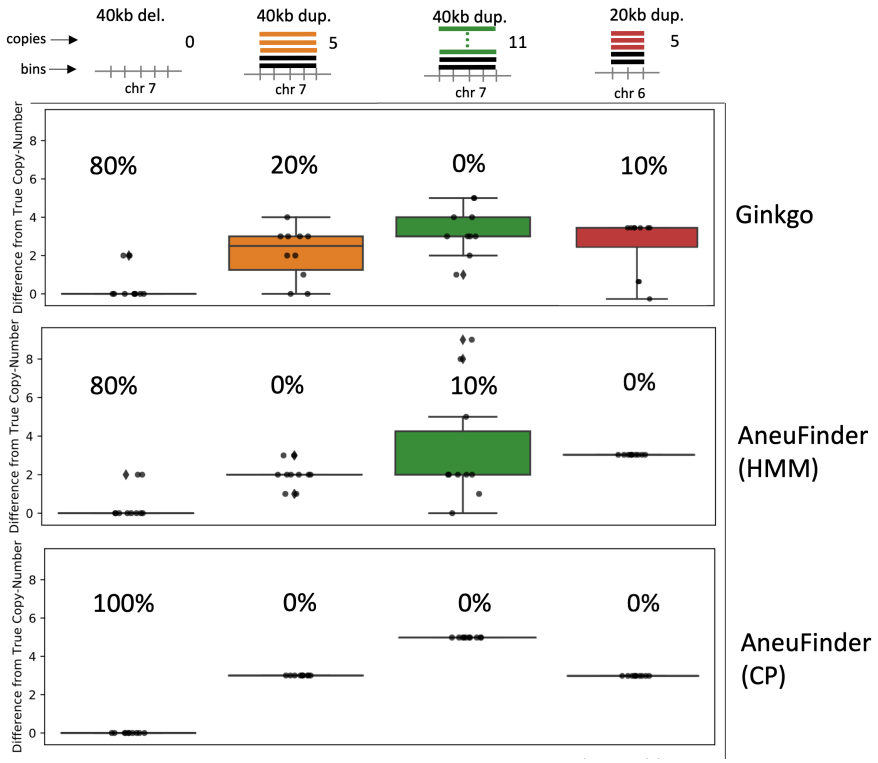


Figure 7: Simulation results show how many known small copy-number variations Ginkgo and AneuFinder are able to identify. (top) Depiction of small simulated CNV events. (bottom) Each box-plot highlights the called copy-number at the location of the copy-number variation (the search flanked 30KB on both sides). The percent above each plot illustrates how many of 10 cells were correctly identified to have the CNV.

inferring diploid bins for most cells until 20KB, when it rapidly declines. Both Ginkgo fixed and variable bin size methods begin to fail inferring diploid bins across all cells starting at a high bin size of 250KB. At 10KB bin size, while AneuFinder fails to infer diploid bins, Ginkgo fails for only a couple of cells. This could again be predominantly due to the fact that Ginkgo's "variable" setting enables it to change the bin size, although a fixed size is set. Overall, it is clear that currently used scDNA CNP inference methods are unable to robustly detect true CNVs at small bin sizes, and not even at 100KB for AneuFinder's HMM model.

3.3.3 Current Methods Miss and Incorrectly Call Small Copy-Number Changes

Ginkgo and AneuFinder are unable to accurately identify small CNV events, and return many erroneous CNV events. To evaluate this claim, we simulated several small CNV events, generated 10 cells for each, and ran Ginkgo and AneuFinder at a 10KB bin size. More specifically, we generated four aberrant genomes harboring either a 40KB deletion, 40KB tandem duplication with copy-number 5, 40KB tandem duplication with copy-number 11, or 20KB tandem duplication with copy-number 5). We simulated 10 cells from each of the these aberrant genomes. We ran AneuFinder and Ginkgo on these sets at a 10KB bin size. Figure 7 (top) illustrates which small events Ginkgo and AneuFinder were to detect. The box plots illustrate the difference from the true copy-number for each region. We considered 30KB up and down from the location where we inserted the CNV. In the ideal scenario, the difference is 0. The percentage above each plot indicates what percent of cells were correctly

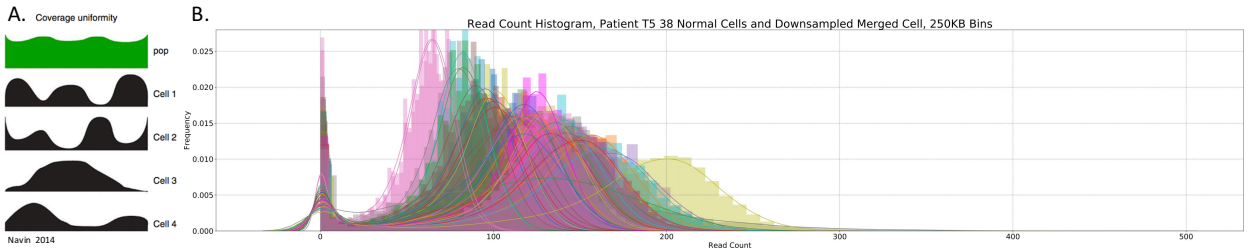


Figure 8: A. Figure from Navin et al. 2014 [45] depicting that coverage uniformity varies per cell, and is often more uniform when sequencing a population of cells (i.e. bulk sequencing). B. Histogram showing the non-uniform read count distribution for 38 normal cells from Patient T5 and a merged down sampled cell (magenta). This is a real data example of the diagram in A. See **[appendix]** for read count histogram at 10kb bin size.

classified as having that CNV. Overall, very few of the small CNV events were correctly identified by Ginkgo and AneuFinder.

3.3.4 Proposed Solution: Consider Cells Jointly to Alleviate Single-Cell DNA Errors in Individual Cells and Improve CNP Inference

Considering cells jointly helps identify true CNV events, especially in the case of small CNVs and at small bin sizes. scDNA fluctuations in coverage and sequencing errors significantly hinder accurate estimates of CNVs occurring across cells of the same clone [45, 46, 47, 8, 7, 16, 4, 17, 81]. In order to utilize read depth information across cells of the same clone, we show that (1) coverage uniformity varies from cell to cell for regions with the same copy-number, and (2) sequencing errors are uniformly distributed across the genome (thus independent between cells). To show that points (1) and (2) can be alleviated by merging information across cells of the same clone, we analyzed 38 normal cells from patient T5 from Gao et al. 2016 [16].

3.3.5 Single-Cell DNA Coverage Uniformity Varies from Cell to Cell

Towards point 1, Figure 8 A from Navin et al. 2014 [45] shows how, compared to bulk sequencing coverage (green), scDNA sequencing's WGA step results in non-uniform read coverage across the genome for regions of the same copy-number (cells 1 - 4). The histogram in Figure 8 B illustrates what is depicted in A, that scDNA WGA errors cause non-uniform read counts across the genome for these 38 normal cells. As coverage uniformity varies from cell to cell, considering cells of the same clone would enable us to more robustly determine the true clone CNP (see Section 3.3.7). One can use a merged cell to illustrate this (see magenta distribution in Figure 8 B).

3.3.6 Single-Cell DNA Errors are Uniformly Distributed and Contribute to False Positive CNVs for Current Methods

Towards point 2, as Navin suggests in [8], amplification errors are randomly distributed across the genome, and can be mitigated by considering cells of the same clone, jointly. For current methods such as Ginkgo, these randomly distributed sequencing errors are sometimes inferred as true events. Figure 9 illustrates this in a bar graph of the frequency of non-diploid

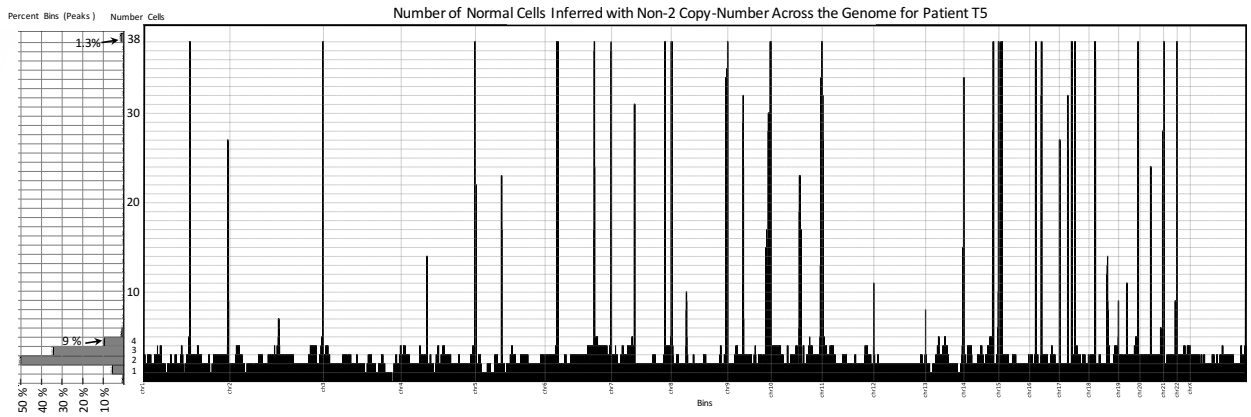


Figure 9: Bar plot of the frequency of non-2 copy-number calls across the genome for 38 normal cells from patient T5 as produced by Ginkgo variable bins (250KB bin size). Large peaks could be a result of issues including germline copy-numbers, unaddressed GC rich and centromeres.

inferred bins for the 38 normal cells (y-axis) across all genomic bins (x-axis) for Ginkgo's variable bin method. The left histogram shows the number of bins where that many cells have a non-diploid copy-number, with only 1.3% of all bins having a non-2 copy-number across all 38 cells, which could be unaddressed GC rich and centromeric regions. In most bins (49.97%), only 2 cells at a time have a non-2 copy-number. Very few non-diploid inferred bins are shared by 38 cells across the genome, indicating that sequencing error is indeed random across each cell. This story holds for Ginkgo's fixed bin method and AneuFinder's methods. As bin size decreases, current methods infer higher amounts of sequencing errors as true events (see Figures 6 and 7).

3.3.7 Integrate Information Across Cells of the Same Clone to Improve Single-Cell DNA CNP Inference

scDNA coverage variation between cells in regions of the same copy-number (Section 3.3.5) and uniformly distributed sequencing errors (Section 3.3.6) can be addressed by considering cells jointly to more accurately infer scDNA CNPs, especially at small bin sizes. Using a simple approach to consider cells jointly; pooling all reads across all cells to create a 'merged cell' enables us to remove some of the confounding sequencing errors, and determine where the changes in read depth are consistent across cells, signifying a true CNV for that clone.

Figure 6 illustrates how this technique of merging information across cells generally helps increase the fraction of diploid bins inferred by both Ginkgo and AneuFinder. In addition to 35 normal cells, there is a merged cells (red star). The red star is the *merged down sampled cell*, formed from 35 normal cells, and has been down sampled to the same coverage as the other 35 cells ($\sim$0.1$\times$).

At higher bin sizes 250KB and 100KB, the merged down sampled cell (red star) shows that by merging information across cells of the same clone, we remove the variability present in many single cells. At smaller bin sizes, we see that in most cases, the merged down sampled cell (red star) has a higher fraction of diploid bins compared to the majority of the 35 normal cells. ADNACOPY starts to break down significantly at 20KB and 15KB, and although about half of the cells have close to zero inferred diploid bins at 15KB, the merged

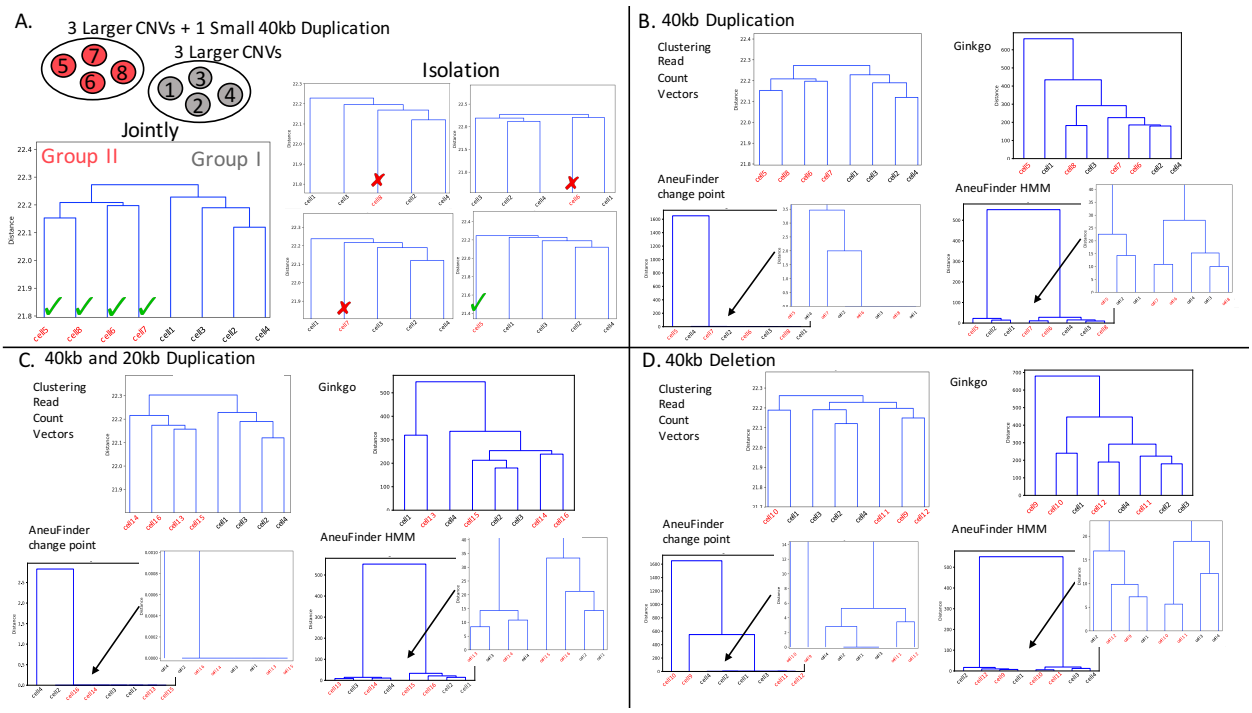


Figure 10: [Change Ginkgo histo.] A. Hierarchical clustering dendrograms (complete linkage and Euclidean distance) comparing vectors of normalized read counts between 8 cells (4 with small CNV(s)- red cells). A. (right) Alone most red cells with the 40KB duplication cluster with black cells, not harboring the small CNV. (left) When all four red cells are considered together, the small CNV is identified and the red cells cluster together. B. Hierarchical clustering dendrograms relating 8 cells using: simple clustering of noisy read count vectors, Ginkgo (variable bins) inferred CNP vectors and AneuFinder methods' CNP vectors. Red cells harbor a small 40KB duplication in chromosome 7. C. Same comparison as in B but red cells also harbor a small 20KB duplication in chromosome 6. D. Same comparison as in B but red cells harbor a small 40KB deletion in the same location of chromosome 7. Overall, simply clustering without copy-number inference and segmentation does better than both methods to identify small CNVs.

downsampled cell is still inferred as diploid.

AHMM is unable to accurately call these diploid cells beginning at 100KB. As the merged downsampled cell represents information from a collection of the other cells, it is AHMM is unable to accurately classify the merged downsampled cell as diploid in some cases, which could be due to the fact that the merged cell is a downsampled average of the 35 normal cells, which were also mostly unable to be classified as diploid. Ginkgo is consistently unable to infer all the normal cells at diploid, but does not deteriorate like AneuFinder. Further, at each bin size the fraction of diploid bins remains high for the merged downsampled cell.

In conclusion, in most cases for both Ginkgo and AneuFinder, these results show that merging information across cells of the same clone can help accurately infer the CNP of the normal clone and thus, these 35 normal cells. Importantly, especially for the cases where the merged cells do not have a high fraction of diploid bins, it is crucial to consider the conclusions one can draw from this naive method of utilizing information across cells (via pooling reads across all .bam files of 35 cells). Algorithmically, we are not interested to pool reads from similar cells together. We are instead interested to build a consensus copy-number for particular bins which present ambiguous information denoting a possible amplification or deletion.

3.3.8 Using All Cells of the Same Clone to Detect Small CNVs Outperforms Ginkgo and AneuFinder When Relating Cells

We do not have reliable read count data to determine single cell events by considering one cell at a time due to scDNA WGA sequencing errors and other sequencing biases. However, under the assumption that such errors occur uniformly at random throughout the genome and are thus independent for all cells, we can look across cells of the same clone to determine a consensus CNP for that clone, thus mitigating the errors and biases which affect individual cells. Some cells will be more correlated with others across bins, as the same fluctuations serve as a proxy to determine the underlying CNV events. This is especially useful when trying to detect small but true CNVs. So, can we identify small CNVs by looking at multiple cells from the same clone? Do these events go undetected when considering all cells separately?

Figure 10 shows several hierarchical clustering dendrograms using complete linkage clustering and Euclidean distance to relate two cell populations; black cells harbor 3 large CNVs, and red cells harbor those 3 large CNVs and either one or two small CNVs (20KB tandem duplication with copy-number 5, 40KB tandem duplication with copy-number 5, or 40KB deletion). In Figure 10 A, when considering each of the cells harboring the small CNV separately (Figure 10 A. (right)), most cells harboring the 40KB tandem duplication CNV (red cells) cluster with cells not harboring that event (black cells). However, when considering all cells harboring the small event (A. left), we can detect the 40KB duplication, which elucidates that we can identify small CNVs by looking at multiple cells from the same clone.

In some instances it is difficult to judge whether or not an event has been detected. It is more useful to consider the resulting relationships between cells, which can be illustrated through a dendrogram. Although Ginkgo and AneuFinder report a dendrogram to relate cells, they do not use the relatedness of cells during CNP inference. Figure 10 B,C, and D show how clustering read counts without segmentation or copy-number calling outperform segmentation and copy-number calling to identify the small events (outlined in Table 7).

3.4 Method: Simultaneous Clustering and Classification of Single-Cell DNA CNVs by Considering Cells Jointly

3.4.1 Single-Cell Copy-Number Identification Using Deep Learning

All cancer cells share a common evolutionary lineage, resulting in groups of cells (i.e. *clones*), with different complements of copy-number variants (CNVs). In order to understand tumor composition and progression given a cohort of single-cells, we must identify which cells are part of the same (1) clone, and (2) identify the complement of CNVs which characterize the cells in each clone, so called the copy-number profile (CNP). In this way, we obtain the *pattern* of CNV events (complement of CNVs) which uniquely characterize each clone.

For (1), this is essentially a clustering problem. Unsupervised deep learning architectures have been widely applied to cluster, to learn low-dimensional feature representations from high-dimentional unlabelled data, similar to classical principal component analysis (PCA) or factor analysis, yet it uses a non-linear model. These types of approaches include stacked autoencoders, restricted Boltzmann machines, and deep belief networks. Clustering has been heavily studied to determine appropriate distance functions and grouping algorithms yet very

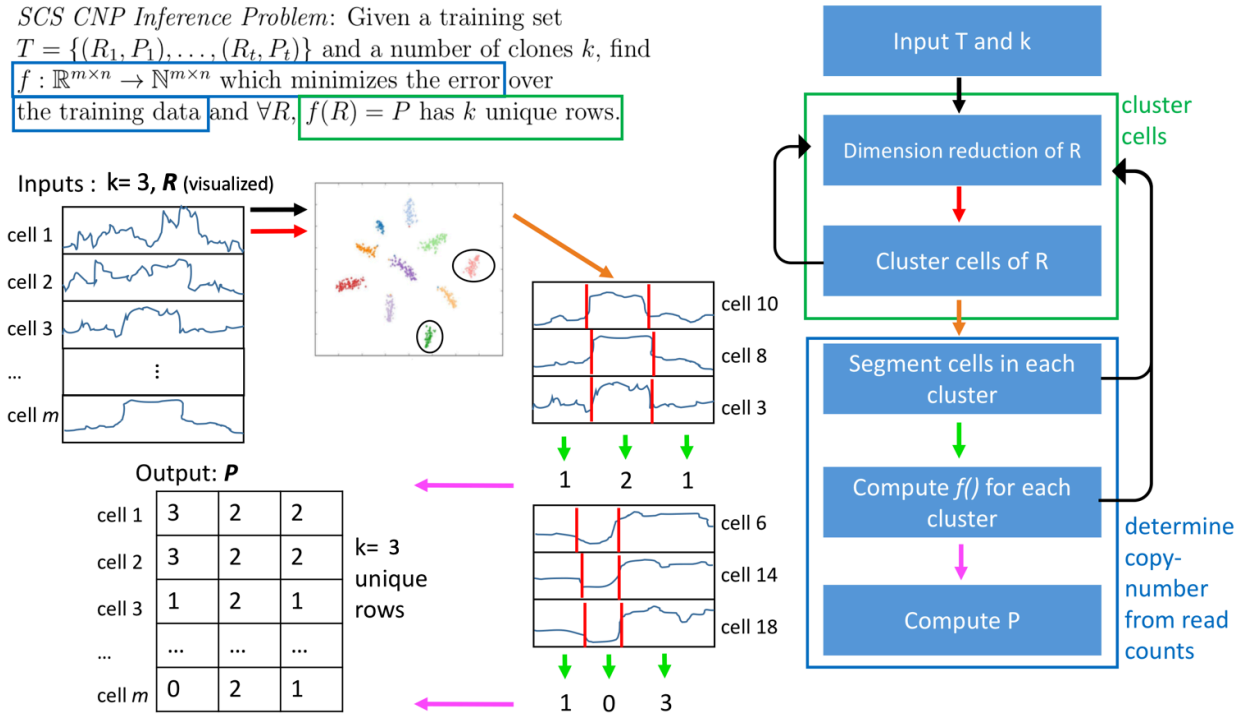


Figure 11: Approach to jointly infer cell-clone assignments, and clone copy-number profiles.

little work has focused on learning representations for clustering [74, 66, 53, 30, 14]. This task entails learning feature representation which divide the cells into groups based on their read count profiles, and then clustering once these features have been identified to separate the cells into groups (i.e. clones). Then supervised learning can be applied to infer the CNPs for all clones. In what follows, we present a model which leverages information across cells to infer the cell to clone assignments for each clone, and then infers the CNP for each clone.

3.4.2 Method Overview

Figure 11 illustrates the proposed approach to jointly infer cell to clone assignments and the clone CNP. Given m single-cells, each divided into n bins, the goal is to determining the CNP for each clone to which each single-cell belongs, and determine the cell-clone assignments themselves by jointly considering all cells in the cohort.

More formally, given a cell i, the observations are the corrected (for GC bias and other errors) read counts (CRC) $\mathbf{r} = (r_1, \ldots, r_n)$, where there are n bins. We want to generate the corresponding CNP $\mathbf{p} = (p_1, \ldots, p_n)$, where p_j denotes the integer copy-number for bin j. We have m such vectors $\mathbf{r}$, which form a CRC matrix $R = [r_{ij}] \in \mathbb{R}_{\geq 0}^{m \times n}$, where r_{ij} denotes the CRC for cell i in bin j. Each matrix R has a corresponding copy-number matrix $P = [p_{ij}] \in \mathbb{N}_{\geq 0}^{m \times n}$, where p_{ij} denotes the copy-number for cell i in bin j, and P has k unique rows. Thus, our training set is formed by $T = \{(R_1, P_1), \ldots, (R_{|T|}, P_{|T|})\}$. We can now consider the following formal problem statement:

scDNA CNP Inference Problem: Given a training set $T = \{(R_1, P_1), \ldots, (R_{|T|}, P_{|T|})\}$ and a number of clones k, find $f : \mathbb{R}^{m \times n} \to \mathbb{N}^{m \times n}$ that minimizes the error over the training data

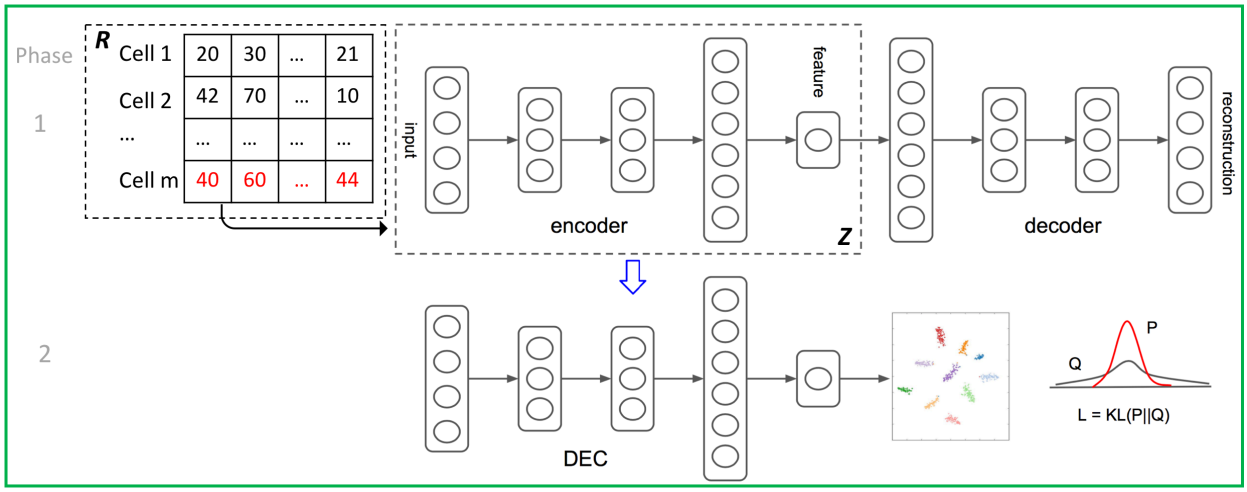


Figure 12: Diagram of Deep Embedded Clustering algorithm with R corrected read count matrix input. Modified diagram from Xie et al. 2016 [74]. Green box associates this figure with the green box in Figure 11.

where f satisfies $f(R_t) = P_t$ and for every corrected read count matrix R, $P = f(R)$ has at most k unique rows.

Note that we assume $m \gg k$ for samples of tumor cells, as we assume that a tumor is composed of clones (groups of cells that have the same complement of CNVs). In the following sections, we explain the steps of this approach, each with the title in Figure 11. Where 'REPEAT' is written indicates when a step should be repeated to inform the assignment of cells to clones, segmentation, and classification of segments to a copy-number state (as illustrated by the black back arrows in Figure 11.

3.4.3 Dimension Reduction of R, Cluster Cells of R

Input to this approach is a training data set T and a number of clones k. One training dataset T is a tuple (R_t, P_t) composed of a CRC matrix R. Now consider our first problem of clustering a set of m cells $\mathbf{r}_i \in R$ into k clusters which are each represented by a centroid μ_c, $c = 1, \ldots, k$.

Summary: We adopt the Deep Embedded Clustering (DEC) algorithm proposed by Xie et al. 2016 [74] is an unsupervised clustering method which learns a non-linear mapping $g(\theta) : R \to Z$ from the data space R, onto a lower dimensional latent feature space Z (using an autoencoder), where θ are the learned parameters, and simultaneously learns the clustering, using a deep neural network. DEC iteratively optimizes a clustering objective (Fig. 12), with the goal of simultaneously solving for soft cluster assignment of data points (i.e. cells) to a fixed number of k clusters (i.e. clones) and determining the underlying feature representation of the clusters.

Clustering is done gradually by iteratively optimizing a Kullback-Leibler (KL) divergence based clustering objective with a self-training target distribution. Stochastic gradient descent (SGD) via back propagation on the clustering objective enables us to learn the mapping $g(\theta)$, which is parameterized by a deep neural network. Thus, DEC jointly performs feature

embedding and clustering, while being robust to hyper-parameter values and allowing for imbalanced clustering data.

Method Steps: More specifically as shown in Figure 12, DEC has two phases, (1) in which DEC is initialized with a stacked autoencoder (SAE) to reduce the dimensionality of the data and learn the best representation of each cell read count sequence. After training the autoencoder, the encoder layers are used as the initial mapping between the data space R and the feature space Z.

During the parameter optimization phase (2), the data are passed through the initialized deep neural network to obtain the embedded (i.e. encoded) data points, which are then clustered using k-means clustering in the feature space Z, to obtain *initial* centroids μ_c, $c = 1, \ldots, k$. With these initial centroids, DEC improves the clustering by alternating between two steps of an unsupervised algorithm. First, soft assignments are computed between the embedded points and the cluster centroids. Second, the non-linear mapping $g(\theta)$ is updated and the cluster centroids are fine-tuned by learning from the current high confidence soft assignments using an auxiliary target distribution [74]. In this way, DEC iterates between computing an auxiliary target distribution and minimizing the KL divergence to that auxiliary distribution until a convergence criterion is satisfied.

Advantages: DEC can be viewed as an unsupervised extension of a more semisupervised self-training method, where it is possible to learn a representation which is appropriate for clustering data *without any* ground truth cluster membership labels. This is therefore ideal to uncover cell to clone assignments and representations of each cluster (i.e. clone) in the data, given that we only know the clone assignments for normal cells.

Contributions: This method provides several novel contributions which can be leveraged in this work. 1) DEC jointly optimizes the representation of the input data (i.e. the deep embedding of each cell in R), and the clusters themselves, 2) DEC iteratively refines the model via soft cluster assignment, and 3) achieves 'state-of-the-art' clustering results (in terms of accuracy and speed) [74]. Xie et al. 2016 [74] studied the performance and accuracy of their method on two very well known image datasets (MNIST [34] and STL-10 [11]) and one text dataset (Reuters [35]).

Value of Autoencoder: Importantly, as explained by Xie et al. 2017 [74], Vincent et al. 2010 [66], Hinton and Salakhutdinov et al. 2006 [24] and others, SAEs have shown to consistently produce semantically significant and well separated representations on real-world data [74]. Autoencoders naturally fit into the deep neural network architecture, and this sort of non-linear technique is useful to discover better and more compact representations of features [66]. For this reason, it is easier for this unsupervised representation which is learned by SAE to facilitate learning cluster assignment and representations with DEC. Another beautiful advantage of using an autoencoder is to be able to learn the noise present in normal cells, and incorporate this learned model when reconstructing and denoising tumor cells.

Model Assumptions: DEC makes the important assumption that the initial classifier's high confidence predictions are mostly correct. To avoid misclassification due to improper place-

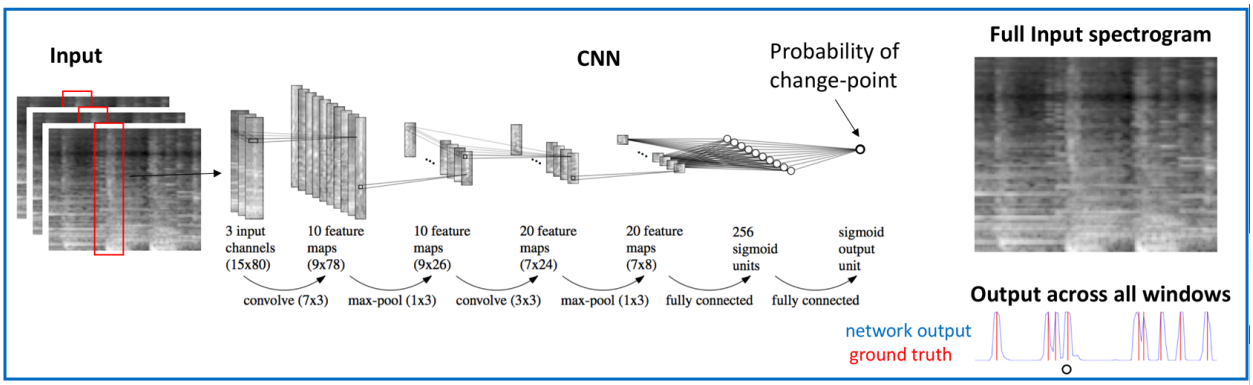


Figure 13: Depiction of convolutional neural network used to segment multiple musical signals concurrently, along with windowed input and full output depiction. Altered image from Schluter et al. 2014 [59]

ment of initial cluster centroids, Xie et al. 2016 [74] run k-means with 20 restarts to pick the result with the best objective value. Note that this model also assumes that we know the number of clusters k. In most single-cell sequencing (scDNA) cancer studies, the number of clones ranges from 3 to about 5 [44, 6, 81], thus, by fixing the number of clones k within this range, we can compare the classification performance to determine the optimal number of clones for a given dataset.

3.4.4 Segment Cells in Each Cluster

For each clone (i.e. cluster), we perform segmentation of the cells in that cluster to identify the change points common to those cells. Let $R' = [r'_{ij}] \in \mathbb{R}_{\geq 0}^{l \times n}$, denote the matrix of the subset l of cells, such that $l < m$, which belong to a clone, such that $k > 1$. Entry r'_{ij} denotes the CRC for cell i in bin j for that clone.

Onset and change point detection: Following Schluter et al. 2014 [59] and Neuner et al. 2015 [49], we can look across multiple cell CRC vectors (i.e. signals represented by R') of one or several KB bin sizes and detect swift vertical changes in those cell CRC vectors across genomic positions (i.e. change points). Detection of changes in our CRC vectors (or signals) are akin to detection of musical onsets using musical audio signals. In Figure 13, Schluter et al. 2014 [59] detect musical onsets by finding the starting points of all musically relevant events in an audio signal. As it pertains to spectral representation, onset detection is closely related to edge detection in images, and change point detection in signals [38, 59, 49, 83]. Onsets are characterized by sharp shifts in the signal over time, and highlighting such edges requires local information. This task of identifying the sharp shifts can be accomplished by convolution with a small filter kernel, which is the basic operation of a convolutional neural network (CNN). Schluter et al. 2014 [59] claims that if a randomly initialized CNN can detect edges, it should be able to learn a set of suitable filter kernels to detect onsets, which they demonstrate. In what follows, we explain their model and draw parallels to our task.

Training and output for onset detection: To train a CNN for onset detection, the input to

the CNN are spectrogram excerpts centered on the frame to classify (see Figure 13 far left). Each excerpt can be represented as a matrix $E = [e_{ij} \in \mathbb{Q}^{l \times f}]$ composed of l spectrogram frequency bands by f frames. e_{ij} describes the sound decibel at frequency band i at the time in frame j. Binary labels (0,1) distinguish onsets from non-onsets, and are the labels to predict. Figure 13 illustrates the input to the network during training, which consists of a matrix of $m = 80$ frequency bands by $f = 15$ frames by 3 channels (i.e. magnitude spectrograms, described later), and the label to predict. The initial filters across the inputs are 7 frames by 3 frequency bands, by 3 channels, and the learned filters change in the consecutive convolution and pooling layers. For each input window, the output unit computes a weighted sum of the hidden unit states (256 in Figure 13), and then applies a logistic sigmoid function, resulting in a prediction value between 0 and 1, which is interpretable as an onset probability for that window of 15 frames across 80 frequency bands.

Training and output for change point detection: For change point detection training, the input similarly consists of several matrices $E = [e_{ij} \in \mathbb{R}^{l \times f}]$ defining windows (i.e. genomic regions), each comprising of l cells by f bins, where e_{ij} describes the corrected CRC for cell i in bin j. Binary labels (0,1) distinguish change points from non-change points, and are the labels to predict for each window. The input structure from [59] would correspond directly to training data of $f = 15$ consecutive bins across all cells in a clone, where each of the 3 channels would correspond to a higher or lower KB bin size. Note that we begin with a simpler model considering only one KB bin size. Importantly, in [59], the authors say that it did not improve predictions to replace the binary targets with sharp Gaussians and turn the classification problem into a regression one, but we can try to see if this improves our results. The output for each window would similarly return a value between 0 and 1, interpretable as a change-point probability occurring in that window, considering all cells in that clone.

Desired output for change point and onset detection: The onset detection output for a full input spectrogram (see Figure 13 far right "Full Input spectrogram") consists of an output vector $\mathbf{o} = (o_1, \ldots, o_n)$ such that for each i, $0 \leq o_i \leq 1$ (see Figure 13 "Output across all windows"), where n is the length equal to the entire duration of the spectrogram of music (equivalent to the entire genome length), with prediction values o_i between $0-1$ denoting the probability/confidence of onset events over time. The desired output for detecting change points over genomic regions is similarly a vector equal to the length of the binned genome (n), with prediction values $0-1$, where values in genomic bin locations closer to 1 indicate higher probability/confidence of change point events.

Convolution explanation: The task of finding changes in the spectrogram signal over time requires filters that are wide in time, and narrow in frequency [59]. Just as with onset detection, change point detection requires high time/space resolution, but is oblivious to frequency (or height of the signal). Thus, similar to Schluter et al. 2014 [59], we can perform max pooling in the CNN over only the corrected read counts domain, just as they do for the frequency domain (see Figure 13).

Channels in onset detection: The use of several channels in [59] is comparable to extracting features across multiple color channels of an input image. In their onset detection method,

the authors train on a stack of spectrograms with a different window size in each channel (e.g. 23ms, 46ms, and 93ms [59]), while maintaining the same frame rates. By using logarithmic filter banks, they are able to reduce to the same number of frequency bands across all channels. In this way, each neuron integrates information of high frequency and temporal accuracy for its temporal location [59].

Channel equivalence for CNV change point detection: In change point detection of signals, this would be equivalent to having multiple channels where the inputs are the same cells with larger or smaller KB bin sizes. For example channel 1 would contain cell CRC vectors of 50KB bin size, channel 2 would contain 250KB, and channel 3 would contain 500KB. To begin, we choose a simpler model, and consider all cells in a clone as one channel. To handle the variability in cluster size, we set a minimum number of cells to belong to each cluster (b), and set a mini-batch size for training input where $b \times f$ randomly selected cells (i.e. signals) to be fed as input to the CNN for training. The discrepancy between each label prediction and its corresponding target is used to improve the performance of the network through the back-propagation step for several epochs. Training then continues with a new mini-batch of b randomly selected input signals from the clone, until all signals have been passed through the network, x times. x is a hyper parameter that must be determined in the calibration phase. This is just one example of how to culminate results across all cells within a clone, while using the same CNN model for all clones (i.e. clusters).

Testing for onset detection: During testing, Schluter et al. 2014 [59] aim to detect onsets in a new input signal (see Figure 13 "Full Input spectrogram" and Figure 14 "Spectrogram 1" and "Spectrogram 2"). Each full spectrogram matrix $S = [s_{ij}] \in \mathbb{Q}^{l \times n}$ is composed of l spectrogram frequency bands by n frames (milliseconds), which define the windows across the full spectrogram (see Figure 13 far left), centered on the frame to classify. S is fed into the network, and using a sliding window approach, the authors apply convolution and pooling operations to the windowed input, and obtain an onset activation function over time as output (o). This function (Figure 13 bottom right) is smoothed by convolution with a 5 frame Hamming window, and local maxima higher than a threshold are reported as onsets.

Testing for change point detection: During testing to segment cells assigned to a clone, if the number of cells $l > b$, we sample b cells from that cell-clone assignment, and compute the change point predictions across the genome for those cells, and then resample until all cells have been considered in the network, s times. The final output change point predictions are the average of all change point probabilities across all cells in the clone, and form the output change point vector o for the cells of that clone.

This output (Figure 14 "Output change points" blue lines) gives as a measure of confidence in change point events for the cells assigned to the same clone. Moreover, this output can also be used to determine how well the cells have been assigned to clones (see Figure 14), and help inform our choice of k clusters.

Figure 14 shows two spectrograms and the associated predicted change points. In grey are the equivalences to the task of determining change points for CNV events across the genome. Thus, consider the output onset (i.e. change point) probabilities for spectrogram 2 in Figure 14. Let τ denote the threshold to accept a particular change point. If this output

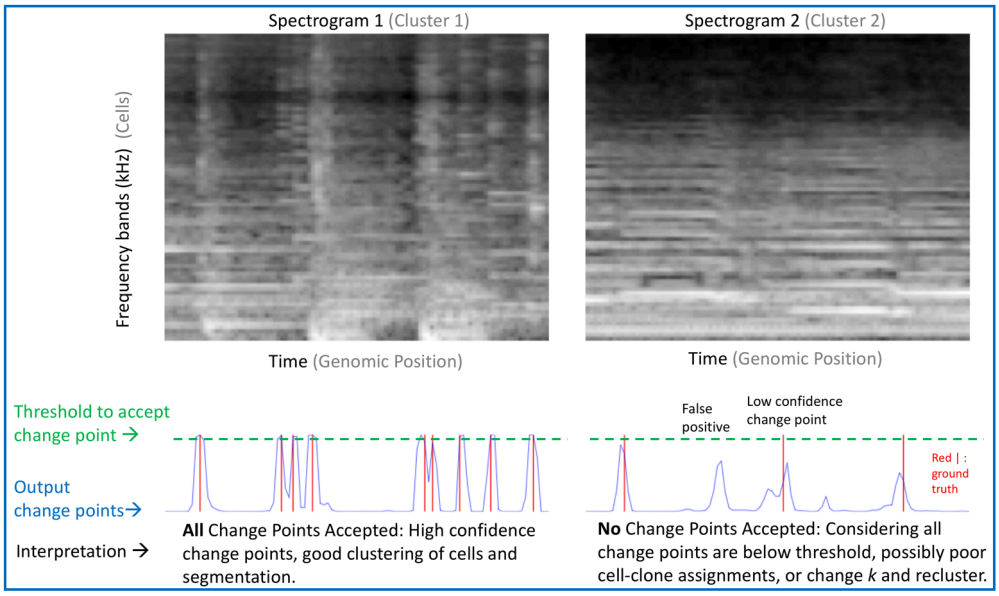


Figure 14: Output of CNN for two spectrograms. Gray figure labels indicates parallels to [59]. This figure is modified from Schluter et al. 2014 [59].

were characterizing several cells assigned to the same clone, that overall confidence measure is low (for example Figure 14 B Spectrogram 2), which could indicate poor cell-clone assignments, or that either τ or k should be changed.

REPEAT: In the event that we want to improve cell-clone assignments or change k, the algorithm returns to the dimension reduction of R and cluster cells of R phase to try and refine the clusters so to determine which cells should be reassigned to another clone. In the event that this process has been repeated and the soft cell-clone assignment does not change, we have two options. (1) It is possible that the cell(s) may have sequencing errors that are significantly overcome by considering other cells in that same clone to which it has been assigned. (2) We can alter k and recluster and measure the difference in between and within cluster distance between the clusters formed by the two choices of k. In contrast, spectrogram 1 shows several high confidence peaks, indicating likely time points of musical onsets. This would correspond to high confidence change points in copy-number across the genome. There are several extensions and modifications possible for the current model for segmentation, including:

1. It may not be enough to look across cells, so also look at bin size. Thus, add multiple channels, where inputs are the same cells with larger or smaller KB bin sizes. For example channel 1 would contain cell CRC vectors with 50KB bin size, channel 2: 250KB, and channel 3: 500KB. This would allow us to determine the best level of granularity (i.e. what bin size) for determining copy-number events. Bin size has a huge influence on the discovery and accuracy to detect true copy-number events (see Figure 6 for examples).

2. Use normal cells to train model, and not just rely on simulation data.

3. Several other approaches to segment include:

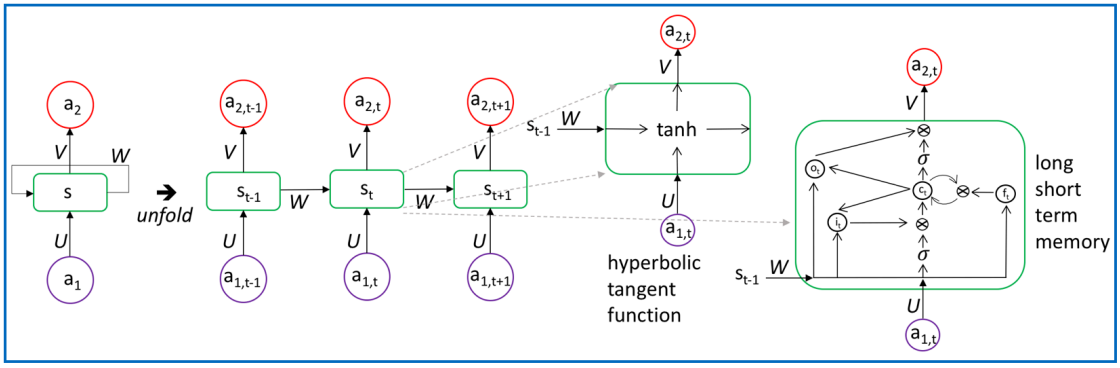


Figure 15: Illustration of a recurrent neural network with *either* long short-term memory *or* the hyperbolic tangent function. i_t is the input gate, f_t is the forget gate, and o_t is the output gate.

(a) Consider implementing multiple networks to accommodate different size inputs for the varying number of cells $l < m$.

(b) Add several types of max pooling layers at the beginning of the CNN to facilitate size reduction of various size input number of cells l, and then use the same CNN structure thereafter.

(c) Compare this approach (segmentation then assigning copy-number to each segment), to another approach without segmentation.

3.4.5 Compute $f()$ for Each Cluster

Each segment is then passed into a recurrent neural network (RNN) to predict the copy-number state of that segment. Each cell is represented by a corrected copy-number (CRC) vector, which can be thought of as an aperiodic and noisy CRC *signal* over space (i.e. genomic loci). RNN are a type of deep neural network widely used to model sequential and temporal data such as signals [9, 58, 39], with connections occurring in a directed cycle. Many types of neural networks require fixed size inputs with fixed size outputs, where the information in the hidden layers is only based on the input data as it flows in one direction, i.e. *feedforward*. RNNs use internal memory to handle *different input sequence sizes* and are thus ideal for sequential learning processes where positional memory matters.

For the input at each each time step (i.e. window or region) $a_{1,t}$, Figure 15 illustrates the network unfolding in space. For example, if the sequence we cared about was 15 bins, the network would unroll into a 15-layer neural network. s_t represents the hidden state at time step t, and is the memory of the network which is calculated based on the previous hidden state as well as the input at the current time step t.

$$s_t = f(U_{a,t} + W_{s_{t-1}}), \tag{3}$$

where U_{at} denotes input to the network for a window at time t and $W_{s_{t-1}}$ denote the weights from time $t-1$. Two examples of this function f are illustrated on the right of Figure 15. Please note that s_{-1} is required to calculate the first hidden state and is normally initialized to all zeroes. $a_{2,t}$ is the output at step t and the input for the next step. If $a_{2,t}$ were the final output layer, we would be able to use this output to predict the next region of the sequence,

or classify the sequence (to determine the copy-number classification). So, the RNN receives information from a combination of the current input data and the hidden layer at the last time step. Given that the inputs to the RNN can be of varying sizes, this allow for learning representations of the segments that are of varying lengths.

REPEAT: Across a number of segments s, if the output maximum integer copy-number classification probabilities are below a threshold δ, the algorithm returns to the dimension reduction of R and cluster cells of R phase, to try and refine the clusters so to determine which cells should be reassigned to another clone. When both the soft cell-clone assignments and the segmentation for the cells assigned to each cluster do not change, each segment for each clone is classified and denoted its final copy-number, along with the classification probability for each segment.

Normal RNNs cannot look too far back in time due to vanishing gradient problems. More explicitly, when error back propagates to the previous layers, it multiplies the gradient of the activation function, which is below 1. Thus, after so many steps, it will decay to about 0. As shown in Figure 15, LSTM (Long Short Term Memory) can be incorporated into RNNs which use the concept of gating where internal memory can be updated i_t, erased f_t or read out o_t, to avoid the vanishing gradient problem found in RNNs. As many segments could be quite long, we choose to incorporate LSTM into the RNN model.

3.4.6 Compute P

Each segment has now been classified with a copy-number, is assigned back to its corresponding cell (and clone), and forms the clone's full CNP. Each segmented and classified cell $c_i = (c_1, \ldots, c_{n'})$ where $c_j \in \mathbb{N}_{\geq 0}$ denotes a segment with an integer copy-number, and n' are the number of segments for that clone. This process is repeated for all cells of each clone. Because each clone is segmented differently, the function $h(c_i)$ splits each classified segment c_i into bins such that each clone can be compared on the same scale. These converted data form the final output matrix $P = [p_{ij}] \in \mathbb{N}_{\geq 0}^{m \times n}$ where p_{ij} denotes the copy-number for cell i in bin j, and P has k unique rows.

3.5 Conclusions

Deep learning models eliminate the need to use hand-crafted features or rules, however both supervised and semi-supervised learning tasks usually require labels. In Section 3.4 we describe an unsupervised cluster discovery approach through an iterative optimization process of 1) feature extraction, and 2) clustering, followed by supervised 3) deep learning (convolutional and recurrent) neural network models to segment and classify cells into clones, each represented by a CNP. This iterative deep learning approach has been recently adopted to address other problems. Wang et al. 2017 [70] (Figure 16 black) presents a looped deep pseudo-task optimization framework to jointly extract deep CNN features and image labels. Zeng et al. 2016 [80] (Figure 16 green) presents a deep model which automatically learns scene-specific features in static video surveillance without labels from the target scene. Their algorithm is able to jointly learn a scene-specific classifier and the distribution of the target samples. And Yan et al. 2016 [77] (Figure 16 red) proposes a model to jointly learn deep

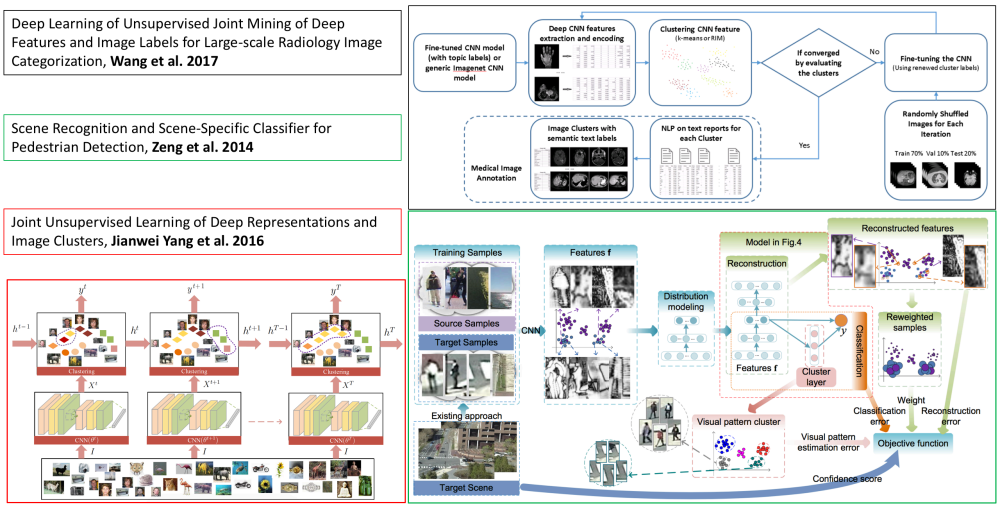


Figure 16: Three iterative methods for joint feature selection, clustering, and classification.

representations and image clusters, where image clustering is performed in the forward pass and representation learning on the backwards pass. The aforementioned three methods each employ iterative approaches where each part of their algorithm informs the other. We used these as a base for our iterative method.

As we highlight in Section 3.3, there is a need for new methods to consider cells jointly when inferring the copy-number for scDNA data, where there is little starting material. The benefits of considering cells jointly are heightened when inferring small CNVs, as illustrated in Section 3.3.3 and Section 3.3.8. As single-cell sequencing is becoming more popular, it is imperative to accurately measure copy-number changes across the genome for cancer patients.x Accurate identification of putative and actionable CNV mutations will enable clinicians to prescribe more tailored therapies, an aid researchers to understand cancer progression more broadly.

References

[1] K. C. Amarasinghe, J. Li, and S. K. Halgamuge. CoNVEX: copy number variation estimation in exome sequencing data using HMM. *BMC Bioinformatics*, 14 Suppl 2:S2, 2013.

[2] Vanessa Andries, Karl Vandepoele, Katrien Staes, Geert Berx, Pieter Bogaert, Gert Van Isterdael, Daisy Ginneberge, Eef Parthoens, Jonathan Vandenbussche, Kris Gevaert, et al. Nbpf1, a tumor suppressor candidate in neuroblastoma, exerts growth inhibitory effects by inducing a g1 cell cycle arrest. *BMC cancer*, 15(1):391, 2015.

[3] Vanessa Andries, Karl Vandepoele, and Frans Van Roy. The nbpf gene family. In *Neuroblastoma-Present and Future*. IntechOpen, 2012.

[4] B. Bakker, A. Taudt, M. E. Belderbos, D. Porubsky, D. C. Spierings, T. V. de Jong, N. Halsema, H. G. Kazemier, K. Hoekstra-Wakker, A. Bradley, E. S. de Bont, A. van den Berg, V. Guryev, P. M. Lansdorp, M. Colome-Tatche, and F. Foijer. Single-cell sequencing reveals karyotype heterogeneity in murine and human malignancies. *Genome Biol.*, 17(1):115, May 2016.

[5] Samprit Banerjee, Derek Oldridge, Maria Poptsova, Wasay M Hussain, Dimple Chakravarty, and

Francesca Demichelis. A computational framework discovers new copy number variants with functional importance. *PloS one*, 6(3):e17539, 2011.

[6] T. Baslan, J. Kendall, L. Rodgers, H. Cox, M. Riggs, A. Stepansky, J. Troge, K. Ravi, D. Esposito, B. Lakshmi, M. Wigler, N. Navin, and J. Hicks. Genome-wide copy number analysis of single cells. *Nat Protoc*, 7(6):1024–1041, Jun 2012.

[7] X. Cai, G. D. Evrony, H. S. Lehmann, P. C. Elhosary, B. K. Mehta, A. Poduri, and C. A. Walsh. Single-cell, genome-wide sequencing identifies clonal somatic copy-number variation in the human brain. *Cell Rep*, 10(4):645, Feb 2014.

[8] X. Chen, J. C. Love, N. E. Navin, L. Pachter, M. J. Stubbington, V. Svensson, J. V. Sweedler, and S. A. Teichmann. Single-cell analysis at the threshold. *Nat. Biotechnol.*, 34(11):1111–1118, Nov 2016.

[9] Keunwoo Choi, George Fazekas, Mark B. Sandler, and Kyunghyun Cho. Convolutional recurrent neural networks for music classification. *CoRR*, abs/1609.04243, 2016.

[10] G. Ciriello, M. L. Miller, B. A. Aksoy, Y. Senbabaoglu, N. Schultz, and C. Sander. Emerging landscape of oncogenic signatures across human cancers. *Nat. Genet.*, 45(10):1127–1133, Oct 2013.

[11] Adam Coates, Andrew Ng, and Honglak Lee. An analysis of single-layer networks in unsupervised feature learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 215–223, 2011.

[12] A. Davis, R. Gao, and N. Navin. Tumor Evolution: Linear, Branching, Neutral or Punctuated? *Biochim. Biophys. Acta*, Jan 2017.

[13] A. Davis and N. E. Navin. Computing tumor trees from single cells. *Genome Biol.*, 17(1):113, 2016.

[14] Nat Dilokthanakul, Pedro A. M. Mediano, Marta Garnelo, Matthew C. H. Lee, Hugh Salimbeni, Kai Arulkumaran, and Murray Shanahan. Deep unsupervised clustering with gaussian mixture variational autoencoders. *CoRR*, abs/1611.02648, 2016.

[15] R Fodde. The apc gene in colorectal cancer. *European journal of cancer*, 38(7):867–871, 2002.

[16] R. Gao, A. Davis, T. O. McDonald, E. Sei, X. Shi, Y. Wang, P. C. Tsai, A. Casasent, J. Waters, H. Zhang, F. Meric-Bernstam, F. Michor, and N. E. Navin. Punctuated copy number evolution and clonal stasis in triple-negative breast cancer. *Nat. Genet.*, Aug 2016.

[17] T. Garvin, R. Aboukhalil, J. Kendall, T. Baslan, G. S. Atwal, J. Hicks, M. Wigler, and M. C. Schatz. Interactive analysis and assessment of single-cell copy-number variations. *Nat. Methods*, 12(11):1058–1060, Nov 2015.

[18] N. P. Gauthier, E. Reznik, J. Gao, S. O. Sumer, N. Schultz, C. Sander, and M. L. Miller. MutationAligner: a resource of recurrent mutation hotspots in protein domains in cancer. *Nucleic Acids Res.*, 44(D1):D986–991, Jan 2016.

[19] C. Gawad, W. Koh, and S. R. Quake. Single-cell genome sequencing: current state of the science. *Nat. Rev. Genet.*, 17(3):175–188, Mar 2016.

[20] M. Gerlinger, A. J. Rowan, S. Horswell, J. Larkin, D. Endesfelder, E. Gronroos, P. Martinez, N. Matthews, A. Stewart, P. Tarpey, I. Varela, B. Phillimore, S. Begum, N. Q. McDonald, A. Butler, D. Jones, K. Raine, C. Latimer, C. R. Santos, M. Nohadani, A. C. Eklund, B. Spencer-Dene, G. Clark, L. Pickering, G. Stamp, M. Gore, Z. Szallasi, J. Downward, P. A. Futreal, and C. Swanton. Intratumor heterogeneity and branched evolution revealed by multiregion sequencing. *N. Engl. J. Med.*, 366(10):883–892, Mar 2012.

[21] M. Gerlinger and C. Swanton. How Darwinian models inform therapeutic failure initiated by clonal heterogeneity in cancer medicine. *Br. J. Cancer*, 103(8):1139–1143, Oct 2010.

[22] Cooper GM. The Cell: A Molecular Approach. 2nd edition. In *The Cell*. Sunderland, MA, 2000.

[23] M. Greaves and C. C. Maley. Clonal evolution in cancer. *Nature*, 481(7381):306–313, Jan 2012.

[24] Geoffrey Hinton and Ruslan Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504 – 507, 2006.

[25] Dana Hollenbeck, Crescenda L Williams, Kathryn Drazba, Maria Descartes, Bruce R Korf, S Lane Rutledge, Edward J Lose, Nathaniel H Robin, Andrew J Carroll, and Fady M Mikhail. Clinical relevance of small copy-number variants in chromosomal microarray clinical testing. *Genetics in Medicine*, 19(4):377–385, 2016.

[26] Matthew KH Hong, Geoff Macintyre, David C Wedge, Peter Van Loo, Keval Patel, Sebastian Lunke, Ludmil B Alexandrov, Clare Sloggett, Marek Cmero, Francesco Marass, et al. Tracking the origins and drivers of subclonal metastatic expansion in prostate cancer. *Nature communications*, 6:6605, 2015.

[27] Y. Hou, L. Song, P. Zhu, B. Zhang, Y. Tao, X. Xu, F. Li, K. Wu, J. Liang, D. Shao, H. Wu, X. Ye, C. Ye, R. Wu, M. Jian, Y. Chen, W. Xie, R. Zhang, L. Chen, X. Liu, X. Yao, H. Zheng, C. Yu, Q. Li, Z. Gong, M. Mao, X. Yang, L. Yang, J. Li, W. Wang, Z. Lu, N. Gu, G. Laurie, L. Bolund, K. Kristiansen, J. Wang, H. Yang, Y. Li, X. Zhang, and J. Wang. Single-cell exome sequencing and monoclonal evolution of a JAK2-negative myeloproliferative neoplasm. *Cell*, 148(5):873–885, Mar 2012.

[28] Jeffrey M Kidd, Gregory M Cooper, William F Donahue, Hillary S Hayden, Nick Sampas, Tina Graves, Nancy Hansen, Brian Teague, Can Alkan, Francesca Antonacci, et al. Mapping and sequencing of structural variation from eight human genomes. *Nature*, 453(7191):56, 2008.

[29] Ki Tae Kim, Bo-Sung Kim, and Ju Han Kim. Association between fat1 mutation and overall survival in patients with human papillomavirus–negative head and neck squamous cell carcinoma. *Head & neck*, 38(S1):E2021–E2029, 2016.

[30] Serkan Kiranyaz, Turker Ince, and Moncef Gabbouj. Real-time patient-specific ecg classification by 1-d convolutional neural networks. *IEEE Transactions on Biomedical Engineering*, 63(3):664–675, 2016.

[31] Peter Koelblinger and Reinhard Dummer. Targeted treatment of advanced nras-mutated melanoma., 2017.

[32] D. A. Landau, S. L. Carter, P. Stojanov, A. McKenna, K. Stevenson, M. S. Lawrence, C. Sougnez, C. Stewart, A. Sivachenko, L. Wang, Y. Wan, W. Zhang, S. A. Shukla, A. Vartanov, S. M. Fernandes, G. Saksena, K. Cibulskis, B. Tesar, S. Gabriel, N. Hacohen, M. Meyerson, E. S. Lander, D. Neuberg, J. R. Brown, G. Getz, and C. J. Wu. Evolution and impact of subclonal mutations in chronic lymphocytic leukemia. *Cell*, 152(4):714–726, Feb 2013.

[33] M. S. Lawrence, P. Stojanov, C. H. Mermel, J. T. Robinson, L. A. Garraway, T. R. Golub, M. Meyerson, S. B. Gabriel, E. S. Lander, and G. Getz. Discovery and saturation analysis of cancer genes across 21 tumour types. *Nature*, 505(7484):495–501, Jan 2014.

[34] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[35] David D Lewis, Yiming Yang, Tony G Rose, and Fan Li. Rcv1: A new benchmark collection for text categorization research. *Journal of machine learning research*, 5(Apr):361–397, 2004.

[36] J. G. Lohr, V. A. Adalsteinsson, K. Cibulskis, A. D. Choudhury, M. Rosenberg, P. Cruz-Gordillo, J. M. Francis, C. Z. Zhang, A. K. Shalek, R. Satija, J. J. Trombetta, D. Lu, N. Tallapragada, N. Tahirova, S. Kim, B. Blumenstiel, C. Sougnez, A. Lowe, B. Wong, D. Auclair, E. M. Van Allen, M. Nakabayashi, R. T. Lis, G. S. Lee, T. Li, M. S. Chabot, A. Ly, M. E. Taplin, T. E. Clancy, M. Loda, A. Regev, M. Meyerson, W. C. Hahn, P. W. Kantoff, T. R. Golub, G. Getz, J. S. Boehm, and J. C. Love. Whole-exome sequencing of circulating tumor cells provides a window into metastatic prostate cancer. *Nat. Biotechnol.*, 32(5):479–484, May 2014.

[37] Sydney X Lu and Omar Abdel-Wahab. Genetic drivers of vulnerability and resistance in relapsed acute lymphoblastic leukemia. *Proceedings of the National Academy of Sciences*, 113(40):11071–11073, 2016.

[38] Haobo Lyu, Hui Lu, and Lichao Mou. Learning a transferable change rule from a recurrent neural network for land cover change detection. *Remote Sensing*, 8(6):506, 2016.

[39] Danilo P Mandic, Jonathon A Chambers, et al. *Recurrent neural networks for prediction: learning algorithms, architectures and stability*. Wiley Online Library, 2001.

[40] Amy L Masson, Bente A Talseth-Palmer, Tiffany-Jane Evans, Desma M Grice, Garry N Hannan, and Rodney J Scott. Expanding the genetic basis of copy number variation in familial breast cancer. *Hereditary cancer in clinical practice*, 12(1):15, 2014.

[41] K. J. McCallum and J. P. Wang. Quantifying copy number variations using a hidden Markov model with inhomogeneous emission distributions. *Biostatistics*, 14(3):600–611, Jul 2013.

[42] N. McGranahan, F. Favero, E. C. de Bruin, N. J. Birkbak, Z. Szallasi, and C. Swanton. Clonal status of actionable driver events and the timing of mutational processes in cancer evolution. *Sci Transl Med*, 7(283):283ra54, Apr 2015.

[43] W. McLaren, L. Gil, S. E. Hunt, H. S. Riat, G. R. Ritchie, A. Thormann, P. Flicek, and F. Cunningham. The Ensembl Variant Effect Predictor. *Genome Biol.*, 17(1):122, 2016.

[44] N. Navin, J. Kendall, J. Troge, P. Andrews, L. Rodgers, J. McIndoo, K. Cook, A. Stepansky, D. Levy, D. Esposito, L. Muthuswamy, A. Krasnitz, W. R. McCombie, J. Hicks, and M. Wigler. Tumour evolution inferred by single-cell sequencing. *Nature*, 472(7341):90–94, Apr 2011.

[45] N. E. Navin. Cancer genomics: one cell at a time. *Genome Biology*, 15(452), 2014.

[46] N. E. Navin. The first five years of single-cell cancer genomics and beyond. *Genome Res.*, 25(10):1499–1507, Oct 2015.

[47] N. E. Navin and K. Chen. Genotyping tumor clones from single-cell data. *Nat. Methods*, 13(7):555–556, Jun 2016.

[48] Cancer Genome Atlas Network et al. Comprehensive molecular characterization of human colon and rectal cancer. *Nature*, 487(7407):330, 2012.

[49] Hans Neuner. Design of artificial neural networks for change-point detection. In *The 1st International Workshop on the Quality of Geodetic Observation and Monitoring Systems (QuGOMS'11)*, pages 139–144. Springer, 2015.

[50] X. Ni, M. Zhuo, Z. Su, J. Duan, Y. Gao, Z. Wang, C. Zong, H. Bai, A. R. Chapman, J. Zhao, L. Xu, T. An, Q. Ma, Y. Wang, M. Wu, Y. Sun, S. Wang, Z. Li, X. Yang, J. Yong, X. D. Su, Y. Lu, F. Bai, X. S. Xie, and J. Wang. Reproducible copy number variation patterns among single circulating tumor cells of lung cancer patients. *Proc. Natl. Acad. Sci. U.S.A.*, 110(52):21083–21088, Dec 2013.

[51] P. C. Nowell. The clonal evolution of tumor cell populations. *Science*, 194(4260):23–28, Oct 1976.

[52] A. B. Olshen, E. S. Venkatraman, R. Lucito, and M. Wigler. Circular binary segmentation for the analysis of array-based DNA copy number data. *Biostatistics*, 5(4):557–572, Oct 2004.

[53] Moein Owhadi-Kareshk and Mohammad-R Akbarzadeh-T. Representation learning by denoising autoencoders for clustering-based classification. In *Computer and Knowledge Engineering (ICCKE), 2015 5th International Conference on*, pages 228–233. IEEE, 2015.

[54] Rolph Pfundt, Marisol del Rosario, Lisenka ELM Vissers, Michael P Kwint, Irene M Janssen, Nicole de Leeuw, Helger G Yntema, Marcel R Nelen, Dorien Lugtenberg, Erik-Jan Kamsteeg, et al. Detection of clinically relevant copy-number variants by exome sequencing in a large cohort of genetic disorders. *Genetics in Medicine*, 19(6):667, 2017.

[55] Anton Platz, Suzanne Egyhazi, Ulrik Ringborg, and Johan Hansson. Human cutaneous melanoma; a review of nras and braf mutation frequencies in relation to histogenetic subclass and body site. *Molecular oncology*, 1(4):395–405, 2008.

[56] I. B. Rogozin and Y. I. Pavlov. Theoretical analysis of mutation hotspots and their DNA sequence context specificity. *Mutat. Res.*, 544(1):65–85, Sep 2003.

[57] A. Roth, A. McPherson, E. Laks, J. Biele, D. Yap, A. Wan, M. A. Smith, C. B. Nielsen, J. N. McAlpine, S. Aparicio, A. Bouchard-Cote, and S. P. Shah. Clonal genotype and population structure inference from single-cell tumor sequencing. *Nat. Methods*, 13(7):573–576, Jul 2016.

[58] Haşim Sak, Andrew Senior, and Françoise Beaufays. Long short-term memory recurrent neural network architectures for large scale acoustic modeling. In *Fifteenth Annual Conference of the International Speech Communication Association*, 2014.

[59] Jan Schluter and Sebastian Bock. Improved musical onset detection with convolutional neural networks. In *Acoustics, speech and signal processing (icassp), 2014 ieee international conference on*, pages 6979–6983. IEEE, 2014.

[60] Michael W Schmitt, Lawrence A Loeb, and Jesse J Salk. The influence of subclonal resistance mutations on targeted cancer therapy. *Nature reviews Clinical oncology*, 13(6):335, 2016.

[61] Adam Shlien and David Malkin. Copy number variations and cancer. *Genome medicine*, 1(6):62, 2009.

[62] Lars Tönges, Jan Christoph Koch, Mathias Bähr, and Paul Lingor. Rocking regeneration: Rho kinase inhibition as molecular target for neurorestoration. *Frontiers in molecular neuroscience*, 4:39, 2011.

[63] Peter Van Loo, Silje H Nordgard, Ole Christian Lingjærde, Hege G Russnes, Inga H Rye, Wei Sun, Victor J Weigman, Peter Marynen, Anders Zetterberg, Bjørn Naume, et al. Allele-specific copy number analysis of tumors. *Proceedings of the National Academy of Sciences*, 107(39):16910–16915, 2010.

[64] S. Vemula, J. Shi, P. Hanneman, L. Wei, and R. Kapur. ROCK1 functions as a suppressor of inflammatory cell migration by regulating PTEN phosphorylation and stability. *Blood*, 115(9):1785–1796, Mar 2010.

[65] E. S. Venkatraman and A. B. Olshen. A faster circular binary segmentation algorithm for the analysis of array CGH data. *Bioinformatics*, 23(6):657–663, Mar 2007.

[66] Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, and Pierre-Antoine Manzagol. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *J. Mach. Learn. Res.*, 11:3371–3408, December 2010.

[67] B. Vogelstein, N. Papadopoulos, V. E. Velculescu, S. Zhou, L. A. Diaz, and K. W. Kinzler. Cancer genome landscapes. *Science*, 339(6127):1546–1558, Mar 2013.

[68] Quirinus JM Voorham, Beatriz Carvalho, Angela J Spiertz, Bart Claes, Sandra Mongera, Nicole CT van Grieken, Heike Grabsch, Martin Kliment, Bjorn Rembacken, Mark A van de Wiel, et al. Comprehensive mutation analysis in colorectal flat adenomas. *PloS one*, 7(7):e41963, 2012.

[69] Ha Linh Vu and Andrew E Aplin. Targeting mutant nras signaling pathways in melanoma. *Pharmacological research*, 107:111–116, 2016.

[70] Xiaosong Wang, Le Lu, Hoo-Chang Shin, Lauren Kim, Mohammadhadi Bagheri, Isabella Nogues, Jianhua Yao, and Ronald M. Summers. Unsupervised joint mining of deep features and image labels for large-scale radiology image categorization and scene recognition. *CoRR*, abs/1701.06599, 2017.

[71] Y. Wang and N. E. Navin. Advances and applications of single-cell sequencing technologies. *Mol. Cell*, 58(4):598–609, May 2015.

[72] Y. Wang, J. Waters, M. L. Leung, A. Unruh, W. Roh, X. Shi, K. Chen, P. Scheet, S. Vattathil, H. Liang, A. Multani, H. Zhang, R. Zhao, F. Michor, F. Meric-Bernstam, and N. E. Navin. Clonal evolution in breast cancer revealed by single nucleus genome sequencing. *Nature*, 512(7513):155–160, Aug 2014.

[73] Ruibin Xi, Angela G Hadjipanayis, Lovelace J Luquette, Tae-Min Kim, Eunjung Lee, Jianhua Zhang, Mark D Johnson, Donna M Muzny, David A Wheeler, Richard A Gibbs, et al. Copy number variation detection in whole-genome sequencing data using the bayesian information criterion. *Proceedings of the National Academy of Sciences*, 108(46):E1128–E1136, 2011.

[74] Junyuan Xie, Ross B. Girshick, and Ali Farhadi. Unsupervised deep embedding for clustering analysis. *CoRR*, abs/1511.06335, 2015.

[75] Bo Xu, Hongmin Cai, Changsheng Zhang, Xi Yang, and Guoqiang Han. Copy number variants calling for single cell sequencing data by multi-constrained optimization. *Computational biology and chemistry*, 63:15–20, 2016.

[76] X. Xu, Y. Hou, X. Yin, L. Bao, A. Tang, L. Song, F. Li, S. Tsang, K. Wu, H. Wu, W. He, L. Zeng, M. Xing, R. Wu, H. Jiang, X. Liu, D. Cao, G. Guo, X. Hu, Y. Gui, Z. Li, W. Xie, X. Sun, M. Shi, Z. Cai, B. Wang, M. Zhong, J. Li, Z. Lu, N. Gu, X. Zhang, L. Goodman, L. Bolund, J. Wang, H. Yang, K. Kristiansen, M. Dean, Y. Li, and J. Wang. Single-cell exome sequencing reveals single-nucleotide mutation characteristics of a kidney tumor. *Cell*, 148(5):886–895, Mar 2012.

[77] Jianwei Yang, Devi Parikh, and Dhruv Batra. Joint unsupervised learning of deep representations and image clusters. *CoRR*, abs/1604.03628, 2016.

[78] Z. Yu, A. Li, and M. Wang. CloneCNA: detecting subclonal somatic copy number alterations in heterogeneous tumor samples from whole-exome sequencing data. *BMC Bioinformatics*, 17:310, 2016.

[79] Fatima Zare, Michelle Dow, Nicholas Monteleone, Abdelrahman Hosny, and Sheida Nabavi. An evaluation of copy number variation detection tools for cancer using whole exome sequencing data. *BMC bioinformatics*, 18(1):286, 2017.

[80] Xingyu Zeng, Wanli Ouyang, Meng Wang, and Xiaogang Wang. *Deep Learning of Scene-Specific Classifier for Pedestrian Detection*, pages 472–487. Springer International Publishing, Cham, 2014.

[81] C. Zhang, H. Cai, J. Huang, and Y. Song. nbCNV: a multi-constrained optimization model for discovering copy number variants in single-cell sequencing data. *BMC Bioinformatics*, 17:384, Sep 2016.

[82] C. Zhang, C. Zhang, S. Chen, X. Yin, X. Pan, G. Lin, Y. Tan, K. Tan, Z. Xu, P. Hu, X. Li, F. Chen, X. Xu, Y. Li, X. Zhang, H. Jiang, and W. Wang. A single cell level based method for copy number variation analysis by low coverage massively parallel sequencing. *PLoS ONE*, 8(1):e54236, 2013.

[83] Xinghui Zhang, Lei Xiao, and Jianshe Kang. Degradation prediction model based on a neural network with dynamic windows. *Sensors*, 15(3):6996–7015, 2015.