Aggregation Tool for Genomic Concepts (ATGC): A deep learning framework for sparse genomic measures and its application to tumor mutational burden

Jordan Anaya[1], John-William Sidhom[2,3,4], Craig A. Cummings[5], Alexander S. Baras[1,2,4], and the AACR Project GENIE Consortium[6]

[1]**Department of Pathology, Johns Hopkins University School of Medicine, Baltimore, MD, USA**
[2]**The Sidney Kimmel Comprehensive Cancer Center, Johns Hopkins University School of Medicine, Baltimore, MD, USA**
[3]**Department of Biomedical Engineering, Johns Hopkins University School of Medicine, Baltimore, MD, USA**
[4]**Bloomberg Kimmel Institute for Cancer Immunotherapy, Sidney Kimmel Comprehensive Cancer Center, Johns Hopkins University School of Medicine, Baltimore, MD, USA**
[5]**Department of Oncology Biomarker Development, Genentech, South San Francisco, CA**
[6]**American Association for Cancer Research, Philadelphia, PA**

Corresponding author:
Alexander S. Baras[1,2,4]

Email address: baras@jhmi.edu

ABSTRACT

Deep learning has the ability to extract meaningful features from data given enough training examples. Large scale genomic data are well suited for this class of machine learning algorithms; however, for many of these data the labels are at the level of the sample instead of at the level of the individual genomic measures. To leverage the power of deep learning for these types of data we turn to a multiple instance learning framework, and present an easily extensible tool built with TensorFlow and Keras. We show how this tool can be applied to somatic variants (featurizing genomic position and sequence context), and accurately classify samples according to whether they contain a specific variant (hotspot or tumor suppressor) or whether they contain a type of variant (microsatellite instability). We then apply our model to the calibration of tumor mutational burden (TMB), an increasingly important metric in the field of immunotherapy, across a variety of commonly used gene panels. Regardless of the panel, we observed improvements in regression to the gold standard whole exome derived value for this metric, with additional performance benefits as more data were provided to the model (such as noncoding variants from panel assays). Our results suggest this framework could lead to improvements in a range of tasks where the sample level metric is determined by the aggregation of a set of genomic measures, such as somatic mutations that we focused on in this study.

INTRODUCTION

It is often said the whole is greater than the sum of its parts. While this may be true, the whole can often still be identified by the sum of its parts. Cancers can be classified by their mRNA or miRNA levels, protein expression, genomic methylation, chromosomal copy number, or somatic mutations (Hoadley et al., 2018; Jiao et al., 2020). Despite this, it remains a challenge to use genomic scale data in traditional machine learning tasks due to the curse of dimensionality—the high dimensional space makes the data sparse and the models prone to overfitting (Altman and Krzywinski, 2018). Current approaches to this problem include manually reducing the dimensionality through feature selection, or reducing the dimensions in an unsupervised manner, often with variational autoencoders. However, it is not always clear how to select features or otherwise reduce the data for the learning task, and dimensionality reduction carries the risk of losing important information.

Of the various genomic measures, perhaps the most common yet most complex is somatic mutations. With 3 billion base pairs and alterations which can range from single nucleotide polymorphisms (SNPs) to large insertions or deletions

(InDels), there are essentially a limitless number of unique mutations. Somatic mutation data differs from the more familiar germline GWAS mutation data in that tumors don't share an evolutionary history with each other and as a result share very few variants. While with germline mutations it's possible to build a model by selecting a set with a given allele frequency cutoff (Romero et al., 2016), with somatic mutation data this would result in throwing away most of the data only leaving a few hotspots (Chang et al., 2018). This has restricted deep learning attempts to the level of the gene rather than the mutation (Kim et al., 2018; Palazzo et al., 2019; Peng et al., 2020), and most work with mutational data has involved simply summarizing it by counting the total number of mutations, total mutational burden (TMB), or by defining mutational signatures based on the type of SNP and neighboring five prime and three prime nucleotides (Chalmers et al., 2017; Alexandrov et al., 2019).

While focusing on which genes are mutated, how many mutations there are, or what types of mutations are present are valid approaches, ideally a single model would be able to encode all of this information, and do so in a manner which allows it to learn features on its own. For example, maybe it's not the total number of mutations that matter, but the number of a given type. Maybe it's not the gene that matters, but the position within a gene. Maybe it's not the neighboring 5'/3' nucleotides that matter, but sequences further away. Deep learning excels at tasks like these where it's unclear what features are important and sufficient data exists for the model to detect patterns.

RESULTS

Model Formulation

Our strategy for applying deep learning to sparse genomic data is to reformulate the problem in the context of multiple instance learning (MIL), wherein each sample is a collection of instances/genomic measures (Dietterich et al., 1997). Applied to somatic mutations each tumor sample would be a bag, and each mutation would be an instance. Because each sample has a different number of mutations, many of which are likely unique, we need a permutation-invariant operation to summarize the contents of the bag (tumor), which in the conventional implementation of MIL is a weighted sum/average, the weights of which can be interpreted as an attention mechanism. We first featurize individual mutations based on genomic position and sequence context (Figure 1A, B) and then subsequently apply an MIL framework to that feaurized mutation data to aggregate the signal over the set of mutations present in a given sample (Figure 1C).

More generally, we present a model we refer to as Aggregation Tool for Genomic Concepts (ATGC) which allows a user to take any concepts (both instance and sample level) and aggregate with attention mechanisms for use in any downstream sample level regression or classification task (Figure 1C). The attention is a recursive layer of multiplication (used in a weighted sum/average) that serves to boost signal through the aggregation and increase the explainability of the model by showing the user which instances were given more weight in the aggregation. Of note, deep learning approaches that lend themselves to a more explainable model will be critical in translating these to clinically meaningful tools (Ching et al., 2018). Below, we will focus on featurizing and aggregating mutation data represented as somatic variants, but we expect this model to be useful for other data types as well.

Variant Level Supervised Learning

To confirm that our sequence and position concepts (Figure 1A, B) are accurate representations of a variant and allow the model to learn features we first performed some "positive control" experiments. For the data we used the public MAF provided by The Cancer Genome Atlas (TCGA) MC3 working group (Ellrott et al., 2018). We extracted the reference and alternative sequences directly from the MAF, padding SNPs and InDels and clipping InDels where necessary. Five prime and three prime sequences were pulled from the human genome (GRCh37) using the provided start positions. For each sequence element (ref/alt & 5/3 prime) in the concept we used six nucleotides (more or less nucleotides can be used, keeping in mind however that longer sequences will be prone to overfitting).

When defining a SNP with the pyrimidine as the reference sequence there are only six types of SNPs (T>C, T>G, T>A, C>T, C>G, C>A), and when considering the neighboring five prime and three prime nucleotides there are $6 \times 4 \times 4 = 96$ contexts. It is these 96 contexts which have been used to define mutational signatures (Alexandrov et al., 2013, 2019, 2020). We labeled the variants in the MAF with these 96 contexts if the variant was a SNP, and gave a separate label for

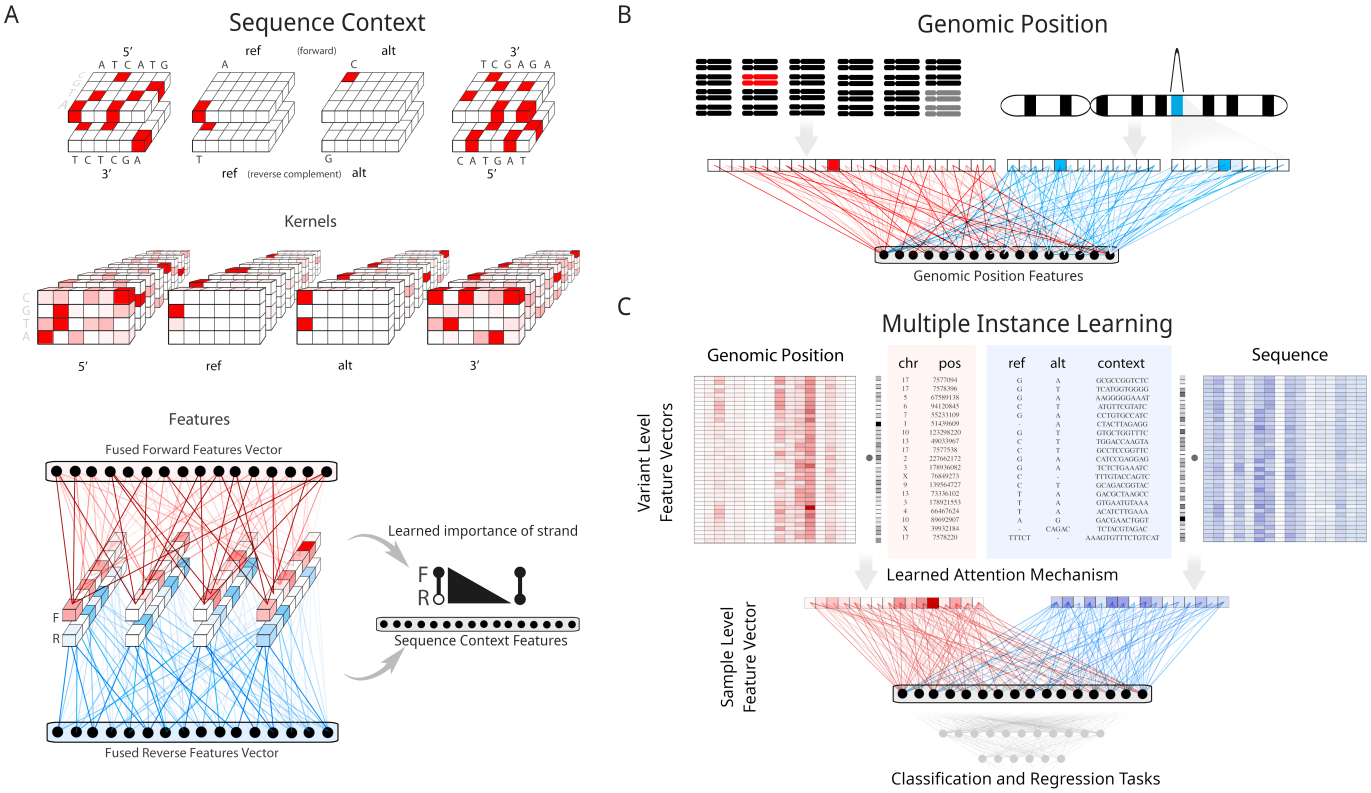

Figure 1. Featurization of mutation data and subsequent application of multiple instance learning. *Sequence and genomic position concepts.* (A) 5', 3', ref, and alt sequences are featurized independently and then fused together in a strand informed manner. Nucleotides are onehot encoded with padded nucleotides given a zero vector. The reverse complement sequences are stacked with the forward sequences such that the representation of a variant is equivalent in both directions. After a single convolution layer the forward features are fused separately from the reverse features, and then these two feature vectors are collapsed with a weighted strand aware aggregation. (B) Chromosome number is onehot encoded along with the position within the chromosome. The fractional position within an intra-chromosomal bin is then sent through two custom activation functions and then all 3 feature vectors are fused. *Sample aggregation* (C) Generally, in mutation profiling a sample is composed of a set of mutations, and a given mutation is defined in some standardized format (MAF/VCF). Each mutation is featurized according to (A) and (B) above and with MIL a weight/attention for each feature vector is developed. These weights are then used to construct a weighted sum (or average) for these feature vectors resulting in a sample level feature vector. This sample level feature vector can subsequently be employed for a variety of classification or regression tasks.

other variant types, resulting in a total of 97 classes. Figure 2A shows the results of the 97 class classification problem. Our concept achieved near perfect accuracy, giving us confidence that our sequence concept is a valid representation of a variant.

While learning the 96 contexts showed that our sequence concept could handle variants coming in forward or reverse, it still remained to be seen if the concept could keep track of which features it extracted came in forward or reverse, which is necessary when strand context matters. To test this aspect of the model we decided to see if the concept could learn variant type as provided by the MAF, specifically the consequence/variant classification of: frameshift deletion, frameshift insertion, inframe deletion, inframe insertion, missense, nonsense, silent, splice site, noncoding (5' UTR, 3' UTR, intron). Learning missense vs. nonsense vs. silent classes will require the concept to learn the 64 codons in all 6 reading frames, and forward and reverse sequences are no longer equivalent in the eyes of the classification task.

Obviously this problem requires giving the model the reading frame, but we first decided to see how well it can do without it (Figure 2C). As expected the model was able to learn the InDels and splice sites since those classes don't require reading frame, and without any identifying features the model was unable to distinguish noncoding mutations from

the other classes. When providing the reading frame concept in the form of strand and CDS position modulo 3 (and a zero vector when the variant did not fall in a CDS), the sequence concept was now able to correctly classify missense vs. nonsense vs. silent mutations (Figure 2D), indicating that our weighted max reduce of the forward and reverse sequences does allow the model to learn which strand a feature was on. The concept's relatively poor performance on what should be easy categories (inframe InDels) may be due to the limited training examples for those classes or the fact that we clipped InDels at 6 nucleotides and thus the concept is blind to InDels larger than that.

For the position concept we are primarily interested to see if our binning of the fractional position along each chromosome provides enough resolution to distinguish genes from one another. As a test we performed another multiclass classification task where we labeled variants according to some genes of interest and put all other variants in an other category. Figure 2B shows that our position concept was able to classify variants based on the gene they occurred in with high precision. In this example the negative class (other) comprises 99.77% of the data, and it may not be a coincidence that the two most frequent genes in this example (TP53, PTEN) had the best recall, suggesting there may a lower limit of detection for this concept (around .008% in this case). Of note, MSH2 and MSH6 are near each other on chromosome 2 and yet the concept was still able to distinguish them.

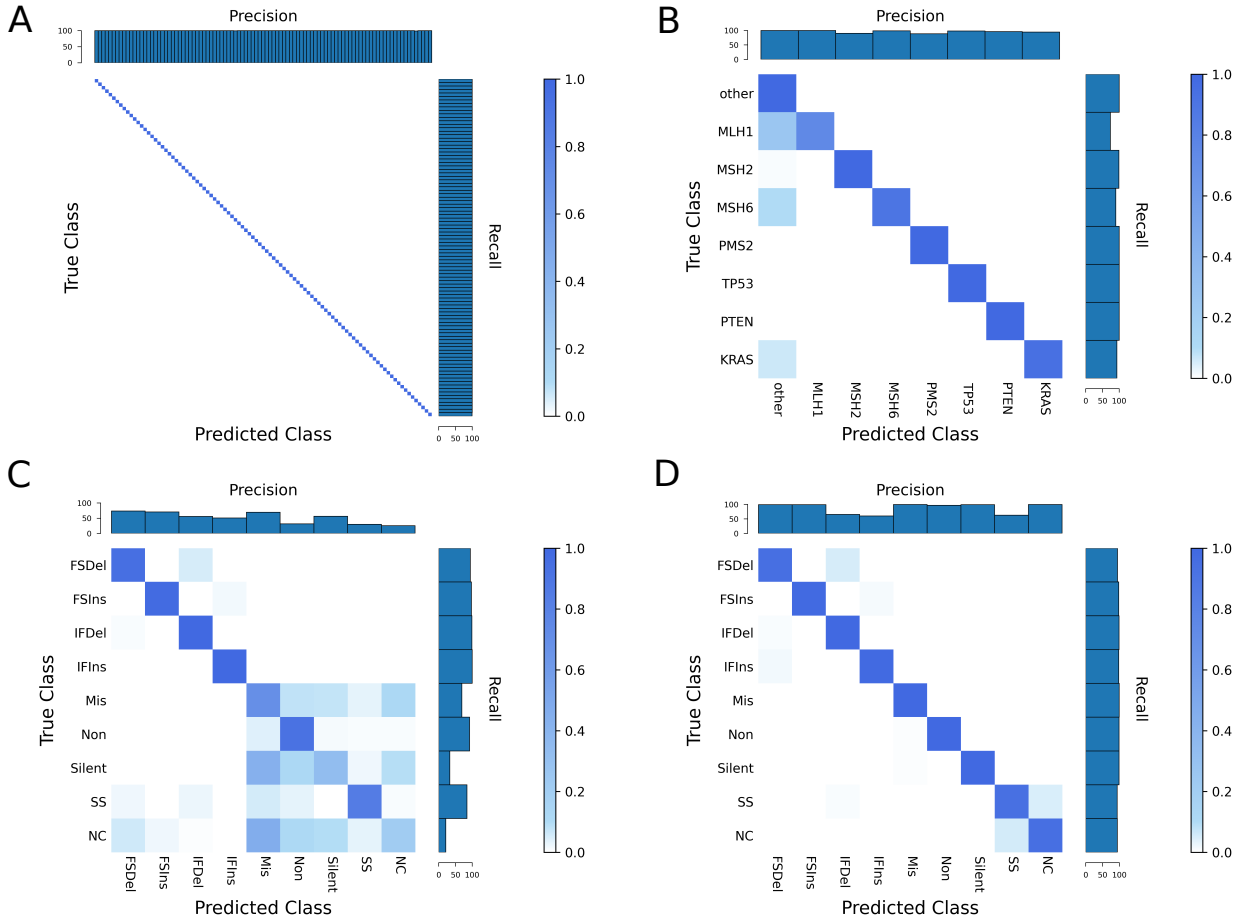

Figure 2. Characterization of performance at the mutation (instance) level. (A) 97 way classification (96 nucleotide contexts and an other/out group category) in which the model is able to accurately learn the conventional trinucleotide mutational contexts. (B) Expanding beyond simple trinucleotide context, the model also accurately can learn a genomic position/gene label. (C & D) Extending further, we show that the model can learn variant consequence encoding as defined by variant effect predictor (VEP) without and obviously better with reading frame information, in (C) and (D) respectively. *FSDel=Frameshift deletion, FSIns=Frameshift insertion, IFDel=InFrame Deletion, IFIns=InFrame Insertion, Mis=Missense, Non=Nonsense, SS=Splice Site, NC=noncoding. All four plots show row normalized confusion matrices.*

Sample Level Supervised Learning—Application of Multiple Instance Learning

To test the ability of signal to pass through our weighted sum and see if our attention mechanism is working as intended we checked to see if our model could identify samples containing either mutational hotspots, or tumor suppressors, and at the levels that these would be present in actual data. Given that a tumor suppressor is characterized by mutations throughout the gene, there is not a specific sequence for the model to find. As a result, we will rely on our position concept to classify samples according to whether they have a mutation in a specific tumor suppressor. And given that hotspots are characterized by changes in a specific amino acid (which corresponds to a specific nucleotide sequence) we will use our sequence concept to identify these samples. It would be possible to use our position concept for both tasks, but mutations near a hotspot would be picked up by the position concept and be false positives.

For the tumor suppressor task we chose to label samples based on whether or not they had a mutation in the PTEN gene. The model's recall (99.89%) was higher than its precision (95.33%), indicating that there may be mutations near the PTEN gene which the model cannot differentiate. Looking at the output of the attention layer (Figure 3A) we see that most variants were given a weight close to 0 while a few were given higher attentions. This is what we would expect—samples contain upwards of thousands of mutations but only a single variant is responsible for the label of the sample so the model needs to ignore all the other variants. Looking at the variants which received the most attention most were indeed mutations in PTEN, but a few were in neighboring genes, which is consistent with the precision metric (giving attention to these variants could result in false positives).

The PTEN mutations fall in two different bins in our position concept, it can take some time for the model to pick up both bins during training, and some other genes also fall in the bins. Clearly our position concept is not ideal for labeling variants according to the gene they fall in, so why wouldn't we just give each gene its own bin? The answer is that we would like our position concept to be gene agnostic because definitions of genes differ, and we would prefer the model to define its own idea of what a gene is. Consider how many different gene definitions are out there—when considering noncoding genes upwards of 100K genes would need to be encoded, presenting computational challenges given the number of weights those encodings would need. And when working with non-exomic data the majority of mutations will not even occur in a gene. With that said, if a user believes a task is driven by mutations in specific genes, it would make sense to onehot those genes.

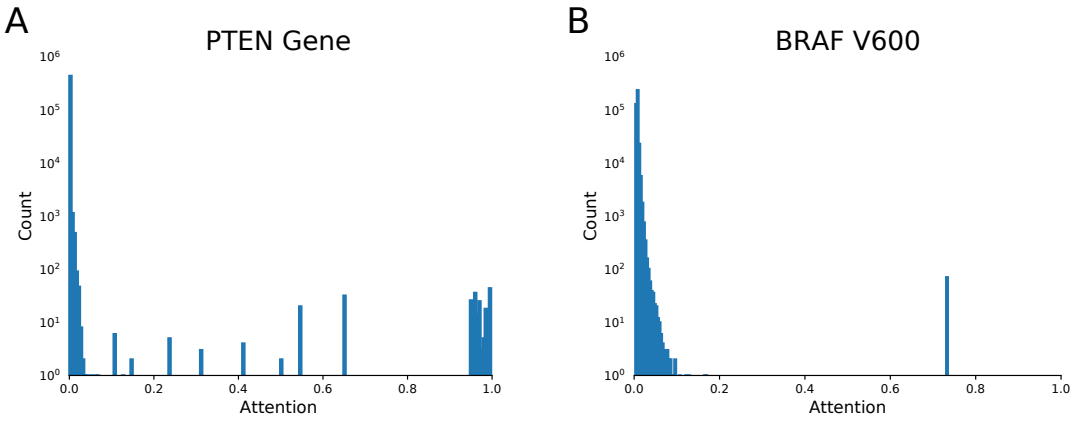

Figure 3. Attention levels highlight the ability of the MIL framework to identify the salient variants. A histogram of the attention layer output values is shown for the tumor suppressor (PTEN) and hotspot (BRAF V600) classification tasks. Only a few variants show high attention values and these map to the salient features, mutations in PTEN gene and BRAF V600 mutations (A and B, respectively).

A hotspot is defined by an amino acid change and not a nucleotide change, so it is possible for a hotspot to be characterized by several different nucleotide sequences. To keep things simple we labeled samples by whether they had a mutation at a specific nucleotide in the BRAF gene, corresponding to the BRAF V600 hotspot. The model was able to achieve both high precision (99.65%) and recall (100.0%). As seen in (Figure 3B) the model again downweighted most variants and only

gave attention to a specific variant, the BRAF V600 mutation. Because all the BRAF V600 variants had the exact same sequence they were all given the exact same attention.

Real World Example: Microsatellite Instability

Our instance and sample level controls gave us confidence that we are able to featurize variants and that the signal could pass through a sample aggregation, but we were still unsure whether variant featurization would succeed in more complex sample level tasks. Our sample level tasks up to this point essentially asked the model to find a needle in a haystack, where the needle was either a variant with a specific sequence or a specific location. However, a more relevant task would be to distinguish two different haystacks by the types and amount of hay they contain. This would correspond to being able to learn a class of tumor by the types and amounts of the variants it contains.

One example where the type and amounts of variants and not a specific variant determines the label of the sample is microsatellite instability. Characterized by mutations in the mismatch repair proteins (MLH1, MSH2, MSH6, PMS2), these tumors accumulate InDels at microsatellites due to polymerase splippage. Given that microsatellites are characterized by short repetitive sequences, and the InDels are multiples of these repeats, we would hope that a deep learning model which featurizes variants would be able to learn to recognize these types of tumors.

Current bioinformatic approaches to predicting MSI status rely on manually featurizing variants. For example, the MANTIS tool uses a BAM file to calculate the average difference between lengths of the reference and alternative alleles at select loci (Kautto et al., 2017), and MSIpred calculates a 22 feature vector using information from a MAF file and a file that defines coordinates of simple repeat sequences (Wang and Liang, 2018). Both of these approaches are valid and produce accurate predictions, but they rely on knowing the nature of the problem. If it was unclear what properties of a variant were important for determining MSI status then these manual featurizations would not be possible.

Because mutations characteristic of MSI occur at many different repeat sites, and because the repeats have a distinctive sequence, we chose to use our sequence concept for this problem. For the data we opted to use the controlled TCGA MC3 MAF rather than the public MAF since the public MAF excludes variants in introns and most simple repeats fall in introns. The TCGA has ground truth labels as defined by PCR assay for some tumor types, and we were able to obtain labels for UCEC (494), STAD (437), COAD (365), READ (126), ESCA (87), and UCS (56) tumor samples. For the sequence concept we went out to 20 nt for each component (5', 3', ref, and alt) to give the model enough space to learn about microsatellites.

We believe MANTIS and MSIpred are considered state of the art when it comes to MSI classification performance, and as can be seen in Figure 4A our model (ATGC) performs comparably. Given the performance of our model we were excited to see what was driving the performance. To classify whether a variant falls in a repeat region or not MSIpred relies on a table generated by UCSC (simpleRepeat.txt). Using this file we labeled variants according to whether they fell in these regions, and as seen in Figure 4B variants which occurred in a simple repeat region were much more likely to receive high attention than variants that did not. It is important to emphasize that all we provided the model were the 5', 3', ref, and alt sequences of the variants as defined in the MAF file, and it was able to learn through the sample labels to pay more attention to variants in simple repeats.

While our model and MSIpred output a probability ranging from 0 to 1 with most predictions at the extremes, MANTIS provides a score ranging from 0 to 2, with 0.4 used as the cutoff to separate MSI stable vs. unstable tumors. We were curious if our model and MANTIS were wrong for the same samples, and if when our model was unsure MANTIS was also unsure. In general when our model was very confident that a sample was MSI high the MANTIS value was also high and when our model was very confident a sample was MSI low the MANTIS value was also low Figure 4C. There are a few cases where the sample is labeled MSI high by the PCR, but both MANTIS and our model are confident the sample is MSI low, perhaps indicating the PCR label may not be 100% specific (or alternatively indicates an issue with the BAM files for these samples). There are also cases where MANTIS predicts the correct label but our model doesn't, or our model predicts the correct label but MANTIS does not, suggesting that the best MSI predictor may be one that incorporates predictions from multiple models.

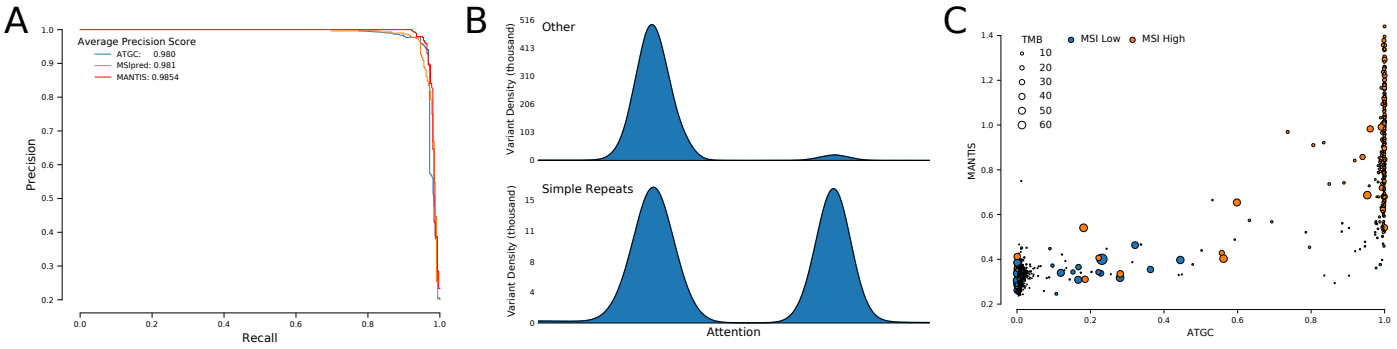

Figure 4. Application of ATGC MIL framework accurately predicts MSI status by learning which are the salient variants (A) Average precision recall curves over 9 stratified K fold splits for MSIpred and ATGC, precision recall curve for MANTIS for samples which had a MANTIS score. (B) Probability density plots of the output of the attention layer stratified by whether a variant was considered to be in a simple repeat from the UCSC genome annotations. (C) MANTIS scores plotted against ATGC output probability showing high concordance of ATGC model output to MANTIS scores. Samples are color coded by the PCR based MSI status label, and the size of each sample corresponds to its total mutational burden (in thousands).

We did not build our model with the intent of learning how to classify tumors according to MSI status, and it's possible our sequence concept is not ideal for this purpose. We also did not do an extensive parameter search, e.g. what length of sequence to use, or add other concepts into the model such as position, so it is entirely possible that the performance on this dataset could be improved upon. We simply wanted to see if a deep learning model could classify samples based on biologically meaningful features extracted from variants, and with performance comparable to manual attempts. While in the case of MSI we knew that there were relevant variant features to extract and what they were, this is not always the case, and we expect deep learning methods to be required for these problems.

Potential Application: Tumor Mutational Burden

One metric involving somatic variants where it's unclear if or what variant features may be involved is tumor mutational burden (TMB). Although defined simply as the number of mutations in a tumor, TMB can be more narrowly defined as the mutations that occur in the exome, the mutations which occur in coding sequences, mutations which occur in genes which are expressed, nonsynonymous mutations, or even a very specific type of mutation such as frameshift InDels. The reasons for potentially being interested in a subset of mutations rather than every mutation is the more neopeptides a tumor has the more likely it is to be recognized by the immune system, and certain mutations contribute more to the generation of neopeptides than others. TMB has generally been shown to correlate with response to immunotherapy (Goodman et al., 2017; Samstein et al., 2019), and the FDA has recently approved a tumor agnostic indication for immunotherapy (FDA, 2020) based on TMB along with a previous approval for immunotherapy for tumors displaying microsatellite instability (Marcus et al., 2019).

The gold standard for assessing nonsynonymous mutation burden would be from whole exome sequencing (WES); however, this is not a routine part of clinical care due to its costs and tissue sample requirements. Instead, TMB is often estimated with panel-based targeted sequencing (as was the case for the recent FDA approval for the "TMB-High" indication). These panels only cover a fraction of the exome, often around 1-5 Mb of ∼33 Mbs, and to complicate matters there are many different panels used across academic and commercial laboratories, each with potentially different sample processing protocols, sequencing technologies, and variant calling pipelines (Fancello et al., 2019). While we cannot explore the effects of all these variables, we can explore the effect of downsampling to a specific set of genes while assuming the other variables are being held constant by artificially downsampling TCGA's exome data.

To explore the ability of our model to estimate exomic TMB given panel data, we returned to the public TCGA MC3 MAF. We defined TMB as nonsynonymous mutations per megabase, used log transformed values, and to prevent outliers from affecting the model excluded samples with a TMB greater than 64. Applying this filter along with some quality control filters (see methods) resulted in a total of 9,942 samples. For restricting the TCGA data to panel coordinates we

relied on the AACR GENIE 7.0 definitions (AACR Project GENIE Consortium and others, 2017).

Panel	Input	Model	MSE	MAE	R-squared
MSK-IMPACT341	Nonsyn	Counting	0.3668	0.4992	0.5155
	Nonsyn	ATGC no Concepts	0.1496	0.3091	0.7232
	CDS	ATGC no Concepts	0.1255	0.2824	0.7767
	CDS	ATGC with Position	0.1048	0.2503	0.8202
	CDS	ATGC with Sequence	0.1074	0.2556	0.8138
MSK-IMPACT468	Nonsyn	Counting	0.2802	0.4325	0.6101
	Nonsyn	ATGC no Concepts	0.1290	0.2870	0.7685
	CDS	ATGC no Concepts	0.1064	0.2598	0.8156
	CDS	ATGC with Position	0.0903	0.2327	0.8468
	CDS	ATGC with Sequence	0.0912	0.2360	0.8446
MDA-409-V1	Nonsyn	Counting	0.2453	0.4062	0.6703
	Nonsyn	ATGC no Concepts	0.1056	0.2584	0.8155
	CDS	ATGC no Concepts	0.0858	0.2325	0.8557
	CDS	ATGC with Position	0.0763	0.2169	0.8725
	CDS	ATGC with Sequence	0.0768	0.2176	0.8709
VICC-01-R2	Nonsyn	Counting	0.2307	0.3889	0.6899
	Nonsyn	ATGC no Concepts	0.1306	0.2849	0.7675
	CDS	ATGC no Concepts	0.1064	0.2576	0.8156
	CDS	ATGC with Position	0.0986	0.2456	0.8334
	CDS	ATGC with Sequence	0.0993	0.2474	0.8291
DFCI-ONCOPANEL-3	Nonsyn	Counting	0.2488	0.4056	0.6365
	Nonsyn	ATGC no Concepts	0.1391	0.2975	0.7420
	CDS	ATGC no Concepts	0.1142	0.2679	0.8007
	CDS	ATGC with Position	0.0968	0.2417	0.8341
	CDS	ATGC with Sequence	0.0963	0.2415	0.8361

Table 1. TMB regression metrics for TCGA MC3 samples

Panel	Input	Model	MSE	MAE	R-squared
DFCI-ONCOPANEL3	Nonsyn	Counting	0.2637	0.4154	0.4949
	CDS	ATGC no Concepts	0.1026	0.2506	0.6598
		ATGC with Position	0.0955	0.2423	0.6817
		ATGC with Sequence	0.1054	0.2557	0.6489
	All	ATGC no Concepts	0.0714	0.1968	0.7937
		ATGC with Position	0.0517	0.1767	0.8400
		ATGC with Sequence	0.0580	0.1881	0.8166
VICC-01-R2	Nonsyn	Counting	0.2563	0.4088	0.5538
	CDS	ATGC no Concepts	0.1059	0.2544	0.6370
		ATGC with Position	0.0988	0.2452	0.6606
		ATGC with Sequence	0.1023	0.2527	0.6212
	All	ATGC no Concepts	0.0869	0.2210	0.7342
		ATGC with Position	0.0672	0.2026	0.7869
		ATGC with Sequence	0.0809	0.2194	0.7214

Table 2. TMB regression metrics for PCAWG ICGC samples

The standard method for estimating nonsynonymous TMB from panel data is to simply count the number of nonsynonymous mutations seen in the panel and then divide by the panel coding sequence size to get to mutations per Mb (Samstein

et al., 2019). To compare the performance of our model against this standard we restricted the TCGA data to the footprint of several panels and ran our model with no concepts, our position concept, or our sequence concept based on inputs of all CDS variants as compared to restricting to nonsynonymous variants contained in these panel, see Table 1. ATGC with no concepts (effectively a nonlinear regression with all variants having equal weight) is a major improvement over the naive counting approach in terms of predicting the whole exome derived metric. Additionally, we observed that the inclusion of all CDS variants over just nonsynonymous variants improved performance. When featurizing variants with either our position or sequence concept we see further improvements in the fit as defined by either mean squared error, mean absolute error, or R-squared. We applied random permutation testing in this context by shuffling variants with respect to which sample they came from while keeping the total number of variants of a given sample the same as in the observed data. The performance benefit of variant featurization was abrogated with the application of this random permutation testing. (Supplemental Table 1).

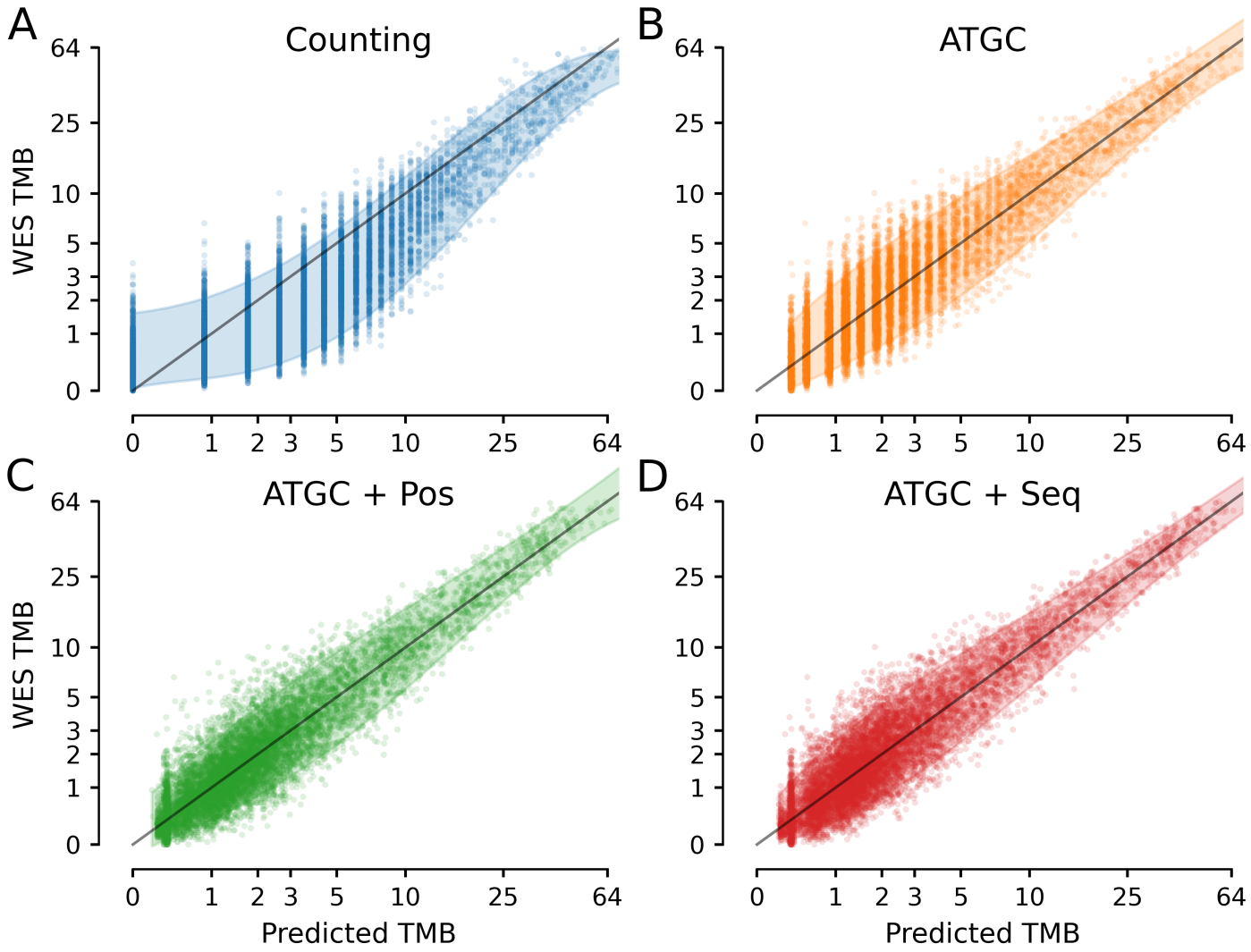

Figure 5. Comparison of panel to exomic nonsynonymous TMB regression shows improved regression characteristic with the application of ATGC and variant featurization as compared to naive counting based approaches. Predicted vs whole exome sequencing derived TMB based on the TCGA MC3 data using the footprint of the MSK-IMPACT468 panel. Shaded area represents the upper and lower quantile estimates for 90% of the data (i.e. quantile regression). The ATGC model predictions are from 8 different K folds. (A) Counting—count of nonsynonymous mutations divided by size of coding sequence covered by panel. (B) ATGC model applied with no featurization concepts, i.e. all variants have equal weight of 1. (C & D) ATGC with position or sequence concepts allowing variants to be differentially weighted.

Based on Table 1, it was evident that the inclusion of additional variants (outside of those used to derive the whole exome level metric) resulted in improved performance. The TCGA MC3 data is WES derived and therefore the vast majority of variants reported in these data are contained within exons. In contrast, panels often include non-exomic regions in their design (Supplemental Figure 1). For example, the DFCI-ONCOPANEL-3 is reported to have more intronic sequence than exonic sequence, and the VICC-01-R2 is approximately half non-CDS sequences. As noted above, models trained on TCGA data were limited to input variants that fell in the CDS with a regression target of whole exome derived nonsynonymous mutation rate. To further explore whether the inclusion of additional variants (for example intronic variants) would improved model performance we turned to whole genome data available from The Pan-Cancer Analysis of Whole Genomes (PCAWG). As with the TCGA MC3, we applied the footprint of the panels to the PCAWG whole genome data and set the regression target to the nonsynonymous mutation rate in PCAWG data restricted to the exome. As can be seen in Table 2 the inclusion of non-exomic variants helped all models except the naive counting based model (which doesn't change); however, the additional variants appeared to help the models with variant featurization more than the nonlinear model.

To further explore the difference in fit between the different models we looked at the predicted TMBs vs actual TMBs. As seen in Figure 5, the counting method initially severely underestimates TMB since it has to predict 0 any time there are no variants the panel, and then classically overestimates TMB. The nonlinear regression is able to make the residuals more symmetrical, but the model must predict the same value for a given number of panel variants (illustrated in the spaces between columns of predictions). With variant featurization the model is now able to give each variant a different weight, so two samples with the same number of panel variants can have different exome predictions. This is illustrated in both the position model and sequence model plots by the absence of the stepwise predictions.

DISCUSSION

The power of our model lies in its flexibility. With an appropriate loss function the model can be pointed to any sample level metric. Here we demonstrated its application to microsatellite instability classification and TMB calibration, but the model could have just as easily been used to classify samples according to tumor type, or regress survival. And although we regressed against nonsynonymous mutations for TMB, we could have calibrated any definition of TMB with no changes to the input or model architecture.

In these experiments we utilized TCGA MC3 data and AACR GENIE panel footprints to generate estimates of what these panels would be able to measure from the TCGA MC3 samples, an approach that has been previously described (Endris et al., 2019; Buchhalter et al., 2019; Budczies et al., 2019; Merino et al., 2020). While not entirely novel, this study represents the first attempt to our knowledge in which variant featurization is used to reweight variants and calculate exomic TMB along with MSI. Panel data typically overestimates TMB due to the panels covering commonly mutated genes in cancer, so this could be one feature that could inform TMB prediction. In fact, some protocols call for removing common mutations when calculating exomic TMB from panel data (Chalmers et al., 2017). While a strong rationale, the question then becomes which variants to count and which ones to ignore, and whether this is a binary vs continuous concept. And does the list change depending the panel? On the cancer type? Ideally TMB calibration would not require manually filtering of the data for each use case with potentially competing filtering protocols across clinics. Another potential reason for panel data overestimating TMB is the higher depth of sequencing, which may allow the panels to call variants with lower variant allele frequencies, but this is a bias that's difficult to explore without matched panel/exome data.

When thinking about how variant featurization may be improving the regression metrics it seems likely that the model is trying to identify types of variants which are overrepresented or underrepresented in the panel, potentially in a tumor specific manner. The model is trying to learn to predict the number of nonsynonymous mutations in the exome given a subsampling of the total mutations as represented in a given panel. Arguably, for most tumors this is primarily determined by the background mutation rate of the tumor. Indeed, calculating the background mutation rate has been proposed as a method for estimating TMB from panels (Yao et al., 2020). However, calculating the background mutation rate requires incorporating gene expression and replication timing data, which will likely not be available during model application, and thus would have to be inferred from another data source. This additional variant level data could easily be incorporated

into our model, and if they are in fact associated with the background mutation rate undoubtedly will improve the model's performance, but may reduce the model's ability to generalize to other datasets. As a surrogate for this, we found that the inclusion of additional variants outside of the exomic definition resulted in an improvement in the ability to estimate exomic TMB, likely due to a better characterization of the background mutational rate.

It is important to note that when training a deep learning model one must ensure the training data is reasonably representative of the potential test data. One issue with using the TCGA MC3 data is that it only includes exomic sequences, while panels usually include noncoding sequence coverage (generally for the purposes of detecting fusions). Interestingly, we observed that the inclusion of variants detected in panels outside of the strict definition for the whole exome derived metric improved model performance, as has been suggested previously (Buchhalter et al., 2019). Another consideration is that TCGA data is matched tumor-normal data, which thereby allows for direct filtering of germline mutations. In contrast, many panel-based methods do not include a matched normal sample of the patient and instead filter the identified variants against large population scale exome data sets (gnomad.broadinstitute.org, for example) for common / known germline variants; however, it is known that this approach will allow for a small number of so-called "private variants" to pass through these types of filtering approaches and incorrectly be considered somatic alterations. As a result, it would be inappropriate to use a model trained on somatic mutation data from TCGA for these types of panels. One potential consideration is to reincorporate germline mutations into the TCGA data and then filter based on public germline databases (as mentioned above) prior to subsampling based on the panel genomic footprint.

Furthermore, when applying a model trained on TCGA MC3 MAF data you are working under the assumption that the data processing which led to the MAF file was similar to the processing of the data you are applying the model to. It is for this reason that it is not necessarily the case that our model (or MSIpred for that matter) should be selected over MANTIS for MSI predictions despite the fact that our model may outperform MANTIS on TCGA data given enough concepts and hyperparameter adjustments. MANTIS operates directly on the BAM files, and as a result is less biased by the data processing pipelines in TCGA data.

Similarly to additional variant level information, it is also possible to include sample level information. Tumor type, age, sex, grade, smoking status, etc., could all easily be added to the model and likely improve performance. However, it is up to a researcher to determine if they believe the samples in the training set (in this case TCGA data) will generalize to their data. For example, a researcher who has panel data for lung cancer samples would likely either want to only train on the TCGA lung cancer data, or train on all the data with cancer type included. However, this application of the model would be assuming that the lung cancer samples in the TCGA are representative of the researcher's lung cancer samples. Ideally, laboratories would generate a set of whole exome and panel level somatic mutation results on a set of common samples and then train our models in that context (potentially also leveraging TCGA data as well) to arrive at the most generalizable and applicable model for their panel and specific aggregate measure they are trying to estimate.

Exomic TMB is associated with response to immunotherapy, which is why attempts are being made to estimate exomic TMB as accurately as possible from panel data. However, if you have the ability to extract features from variants which can be used to predict any sample level metric, why would you use those features to predict exomic TMB? A significant consideration is to directly label the samples with respect to response to therapy and then attempt to train a model to make that distinction from panel level data. While we believe this approach is more powerful, there does not exist an equivalent dataset to the TCGA with patients that received immunotherapy and containing whole exome data; and thus we would not be able to artificially generate panels for regression. As a result, any models would have to be trained on specific clinical trial datasets and would be specific to those panels.

METHODS

Data and Code

All code for processing data, running models, and generating figures is available here. Code is written in Python 3 and TensorFlow 2. All intersections were performed with PyRanges (Stovner and Sætrom, 2020). We leveraged NVIDIA V100s with 32GB of RAM for much of the computation, however most of the computations here could be reasonably

performed on CPU as well within the same coding framework. All data used is publicly available, albeit the controlled MC3 MAF is restricted access.

Model

We used a combination of TensorFlow 2 and tf.keras for implementing ATGC. In Keras the first dimension of the inputs must match the first dimension of the outputs. Given that our inputs are instances, and our outputs are samples, the current implementation places a dummy dimension in front of all inputs and outputs. Doing this precludes the use of many built-in features of Keras, requiring us to write our own generator class for training, and our own loss classes.

The current available loss functions are CrossEntropy, CrossEntropyfromlogits, MeanSquaredError, and QuantileLoss. For all embeddings we take the convention of having the first index (0) correspond to a zero/null vector. The model is modular in the sense that instance and sample featurization are separate "encoder" models from the top MIL model. This allows users to write their own classes for featurizing a concept of interest, and allows for mixing and matching various concepts. Given the overhead of downloading and processing multiple files to repeat the analyses in the paper, if a user wants to quickly start using the model we have provided a script which will generate simulated data, and a script which will perform a classification task on this data.

Instance Level Featurization

We define a genomic measure as the data type under consideration, and a genomic concept as a user-defined model of some aspect of the genomic measure. Multiple genomic concepts can be defined for a given measure. For example, a variant is a genomic measure as defined in a mutation annotation format (MAF) file and it can have a sequence concept, a position concept, an allele frequency concept, reading frame concept, etc. The key to the model learning biologically relevant features from the genomic measure is providing the model biologically accurate representations of the genomic measure in the concepts, and in a format that is conducive to generating strong gradients. For example, it's possible to simply provide the model chromosome and start position as floats, but this would not be ideal. The scale of start position is orders of magnitude larger than the chromosome value, each chromosome is a different length, and the chromosome identifier is not a continuous value.

It is important to ask "what is the goal of this concept?". For position our goal is for the model to possibly learn what gene the variant is in, and maybe what part of a gene it is in. As a result, we onehot the chromosome and then use the fractional position of the start position along the length of the chromosome to assign a bin to the start position (Figure 1B). With 100 bins and 24 chromosomes, this gives the model 2,400 unique combinations. In the event that more than one gene falls in a bin combination, or a specific location within a gene is important, we also provide the model the fractional position of the start position in its assigned bin. If our only goal was to give the model gene information we could have simply onehotted the 20k genes—highlighting the fact that a concept is truly a user's interpretation of how the problem should be framed.

The fractional position in the bin is sent through custom adaptive activations: adaptive sigmoid unit (ASU) and adaptive rectifying unit (ARU), see supplemental methods. If you are only interested in variants in the CDS, it may not make sense to use a coordinate system that spans 3 billion base pairs when only $\sim$30 million of those base pairs are being used. For increased resolution it is possible to remap the genomic coordinates to another coordinate system, potentially one with a different number of chromosomes. This decreases the chance that more than one gene ends up in the same bin combination, but increases the chance a gene is split across bins.

With our sequence concept we would like the model to learn the type of mutation, i.e. SNP vs. InDel, along with the sequences upstream and downstream of the mutation. Therefore our concept of a variant includes the five prime nucleotides, three prime nucleotides, reference sequence, and alternative sequence. To allow the model to be able to handle both SNPs and InDels we pad the reference and alternative sequences, and clip InDels that are larger than a user-defined window. The 4 DNA nucleotides are onehotted, with padded sequence encoded as a vector of all zeros.

Our sequence concept allows for any length of sequence, with the restriction that the length is even. This is due to how we handle large insertions and deletions. When insertions or deletions are larger than the requested size we define the insertion or deletion as the sequences at the edges, with an equal amount of sequence on each side.

We use a single convolutional layer to featurize each sequence component individually, and then concatenate the resulting features. Because of the symmetry of DNA and the fact that it is impossible to know which strand a mutation occurred on, there are two equivalent representations for any given variant. This further increases the sparsity of the already sparse mutation data, so to combat this we give the model variants in both the forward and reverse directions. To collapse this extra dimension of data we perform a max reduce, but to give the model the ability to remember strand information should that information be important, we perform a weighted max reduce (Figure 1A).

Sample Level Aggregation

To increase the signal through the aggregation and allow us to see which instances the model views as important we perform a weighted sum wherein the latent layer for each concept is fed into a singled-headed attention layer, and this layer multiplies back onto the latent layer before a sum. Allowing each concept to have its own attention increases the explainability of the model. For example, if the latent layers for the sequence and position concepts were fused before creating an attention layer, then when an instance receives high attention it would be unclear if that was due to the sequence concept or the position concept. At the same time, it's possible that features from multiple concepts need to interact. For example, maybe it's important that a specific gene is mutated, but also important that it's a frameshift mutation. Thus, in addition to giving attention and summing the features of individual concepts, a user may want fused concept features to get a weighted sum.

TCGA MC3 MAF Processing

Any variant with either FILTER = "PASS" or "wga" were included. We considered nonsynonymous mutations to be any mutation with the labels Missense Mutation, Nonsense Mutation, Frame Shift Del, Frame Shift Ins, In Frame Del, In Frame Ins, Nonstop Mutation. To ensure that a TCGA sample's capture kit covered the exome we downloaded the metadata of the read groups with the GDC API, see supplemental methdos.

TCGA MC3 Controlled MAF Processing

Processing was similar to the public MAF, except we allowed FILTER to equal "PASS", "NonExonic", "bitgt", "wga", or "broad_PoN_v2" to allow as many repeat variants as possible.

ICGC MAF Processing

Processing was similar to the TCGA MC3 MAF, however the MAF contained multiple samples per donor. To restrict the data to one sample per donor we dropped the duplicate rows with PANDAS.

MSI Analyses

PCR labels for TCGA samples were obtained from a combination of cBioPortal, TCGAbiolinks, and individual publications. If the label for a 5 marker call and 7 marker call were in disagreement, the 7 marker call was given preference. Both MSS and MSI-L were considered to be MSI low.

The source code for MSIpred was altered to allow it to run on Python 3, and the model was changed to output probabilities. All repeats from simpleRepeat.txt were used, but otherwise the model was run as recommended.

ACKNOWLEDGMENTS

The results here are in whole or part based upon data generated by the TCGA Research Network.

COMPETING INTERESTS

CC is an employee of Genentech, a member of the Roche Group, and holds Roche stock.

FUNDING

This research was supported by the American Association for Cancer Research (AACR) Project GENIE Consortium, Genentech, Inc., Mark Foundation for Cancer Research (19-035-ASP), and the philanthropy of Susan Wojcicki and Dennis Troper in support of Computational Pathology at Johns Hopkins.

REFERENCES

AACR Project GENIE Consortium and others (2017). AACR Project GENIE: powering precision medicine through an international consortium. *Cancer Discovery*, 7(8):818–831.

Alexandrov, L. B., Kim, J., Haradhvala, N. J., Huang, M. N., Ng, A. W., Wu, Y., Boot, A., Covington, K. R., Gordenin, D. A., Bergstrom, E. N., et al. (2019). The repertoire of mutational signatures in human cancer. *BioRxiv*, page 322859.

Alexandrov, L. B., Kim, J., Haradhvala, N. J., Huang, M. N., Ng, A. W. T., Wu, Y., Boot, A., Covington, K. R., Gordenin, D. A., Bergstrom, E. N., et al. (2020). The repertoire of mutational signatures in human cancer. *Nature*, 578(7793):94–101.

Alexandrov, L. B., Nik-Zainal, S., Wedge, D. C., Aparicio, S. A., Behjati, S., Biankin, A. V., Bignell, G. R., Bolli, N., Borg, A., Børresen-Dale, A.-L., et al. (2013). Signatures of mutational processes in human cancer. *Nature*, 500(7463):415–421.

Altman, N. S. and Krzywinski, M. (2018). The curse (s) of dimensionality this-month. *Nature Methods*, 15(6):399–400.

Buchhalter, I., Rempel, E., Endris, V., Allgäuer, M., Neumann, O., Volckmar, A.-L., Kirchner, M., Leichsenring, J., Lier, A., von Winterfeld, M., et al. (2019). Size matters: Dissecting key parameters for panel-based tumor mutational burden analysis. *International Journal of Cancer*, 144(4):848–858.

Budczies, J., Allgäuer, M., Litchfield, K., Rempel, E., Christopoulos, P., Kazdal, D., Endris, V., Thomas, M., Fröhling, S., Peters, S., et al. (2019). Optimizing panel-based tumor mutational burden (TMB) measurement. *Annals of Oncology*, 30(9):1496–1506.

Chalmers, Z. R., Connelly, C. F., Fabrizio, D., Gay, L., Ali, S. M., Ennis, R., Schrock, A., Campbell, B., Shlien, A., Chmielecki, J., et al. (2017). Analysis of 100,000 human cancer genomes reveals the landscape of tumor mutational burden. *Genome Medicine*, 9(1):34.

Chang, M. T., Bhattarai, T. S., Schram, A. M., Bielski, C. M., Donoghue, M. T., Jonsson, P., Chakravarty, D., Phillips, S., Kandoth, C., Penson, A., et al. (2018). Accelerating discovery of functional mutant alleles in cancer. *Cancer Discovery*, 8(2):174–183.

Ching, T., Himmelstein, D. S., Beaulieu-Jones, B. K., Kalinin, A. A., Do, B. T., Way, G. P., Ferrero, E., Agapow, P.-M., Zietz, M., Hoffman, M. M., et al. (2018). Opportunities and obstacles for deep learning in biology and medicine. *Journal of The Royal Society Interface*, 15(141):20170387.

Dietterich, T. G., Lathrop, R. H., and Lozano-Pérez, T. (1997). Solving the multiple instance problem with axis-parallel rectangles. *Artificial Intelligence*, 89(1-2):31–71.

Ellrott, K., Bailey, M. H., Saksena, G., Covington, K. R., Kandoth, C., Stewart, C., Hess, J., Ma, S., Chiotti, K. E., McLellan, M., et al. (2018). Scalable open science approach for mutation calling of tumor exomes using multiple genomic pipelines. *Cell Systems*, 6(3):271–281.

Endris, V., Buchhalter, I., Allgäuer, M., Rempel, E., Lier, A., Volckmar, A.-L., Kirchner, M., von Winterfeld, M., Leichsenring, J., Neumann, O., et al. (2019). Measurement of tumor mutational burden (TMB) in routine molecular diagnostics: in silico and real-life analysis of three larger gene panels. *International Journal of Cancer*, 144(9):2303–2312.

Fancello, L., Gandini, S., Pelicci, P. G., and Mazzarella, L. (2019). Tumor mutational burden quantification from targeted gene panels: major advancements and challenges. *Journal for Immunotherapy of Cancer*, 7(1):183.

FDA (2020). FDA approves pembrolizumab for adults and children with tmb-h solid tumors. https://www.fda.gov/drugs/drug-approvals-and-databases/fda-approves-pembrolizumab-adults-and-children-tmb-h-solid-tumors.

Goodman, A. M., Kato, S., Bazhenova, L., Patel, S. P., Frampton, G. M., Miller, V., Stephens, P. J., Daniels, G. A., and Kurzrock, R. (2017). Tumor mutational burden as an independent predictor of response to immunotherapy in diverse cancers. *Molecular Cancer Therapeutics*, 16(11):2598–2608.

Hoadley, K. A., Yau, C., Hinoue, T., Wolf, D. M., Lazar, A. J., Drill, E., Shen, R., Taylor, A. M., Cherniack, A. D., Thorsson, V., et al. (2018). Cell-of-origin patterns dominate the molecular classification of 10,000 tumors from 33 types of cancer. *Cell*, 173(2):291–304.

Jiao, W., Atwal, G., Polak, P., Karlic, R., Cuppen, E., Al-Shahrour, F., Bailey, P. J., Biankin, A. V., Boutros, P. C., Campbell, P. J., Chang, D. K., Cooke, S. L., Deshpande, V., Faltas, B. M., Faquin, W. C., Garraway, L., Getz, G., Grimmond, S. M., Haider, S., Hoadley, K. A., Kaiser, V. B., Kato, M., Kübler, K., Lazar, A. J., Li, C. H., Louis, D. N., Margolin, A., Martin,

S., Nahal-Bose, H. K., Nielsen, G. P., Nik-Zainal, S., Omberg, L., P'ng, C., Perry, M. D., Rheinbay, E., Rubin, M. A., Semple, C. A., Sgroi, D. C., Shibata, T., Siebert, R., Smith, J., Stein, L. D., Stobbe, M. D., Sun, R. X., Thai, K., Wright, D. W., Wu, C.-L., Yuan, K., Zhang, J., Danyi, A., de Ridder, J., van Herpen, C., Lolkema, M. P., Steeghs, N., Morris, Q., Group, P. T. S., Working, C. T., and PCAWG Consortium (2020). A deep learning system accurately classifies primary and metastatic cancers using passenger mutation patterns. *Nature Communications*, 11(1):728.

Kautto, E. A., Bonneville, R., Miya, J., Yu, L., Krook, M. A., Reeser, J. W., and Roychowdhury, S. (2017). Performance evaluation for rapid detection of pan-cancer microsatellite instability with MANTIS. *Oncotarget*, 8(5):7452.

Kim, S., Lee, H., Kim, K., and Kang, J. (2018). Mut2Vec: distributed representation of cancerous mutations. *BMC Medical Genomics*, 11(2):33.

Marcus, L., Lemery, S. J., Keegan, P., and Pazdur, R. (2019). FDA approval summary: pembrolizumab for the treatment of microsatellite instability-high solid tumors. *Clinical Cancer Research*, 25(13):3753–3758.

Merino, D. M., McShane, L. M., Fabrizio, D., Funari, V., Chen, S.-J., White, J. R., Wenz, P., Baden, J., Barrett, J. C., Chaudhary, R., Chen, L., Chen, W. S., Cheng, J.-H., Cyanam, D., Dickey, J. S., Gupta, V., Hellmann, M., Helman, E., Li, Y., Maas, J., Papin, A., Patidar, R., Quinn, K. J., Rizvi, N., Tae, H., Ward, C., Xie, M., Zehir, A., Zhao, C., Dietel, M., Stenzinger, A., Stewart, M., Allen, J., and TMB Harmonization Consortium (2020). Establishing guidelines to harmonize tumor mutational burden (TMB): in silico assessment of variation in TMB quantification across diagnostic platforms: phase I of the Friends of Cancer Research TMB Harmonization Project. *Journal for Immunotherapy of Cancer*, 8(1).

Palazzo, M., Beauseroy, P., and Yankilevich, P. (2019). A pan-cancer somatic mutation embedding using autoencoders. *BMC Bioinformatics*, 20(1):655.

Peng, J., Zou, D., Gong, W., Kang, S., and Han, L. (2020). Deep neural network classification based on somatic mutations potentially predicts clinical benefit of immune checkpoint blockade in lung adenocarcinoma. *OncoImmunology*, 9(1):1734156.

Romero, A., Carrier, P. L., Erraqabi, A., Sylvain, T., Auvolat, A., Dejoie, E., Legault, M.-A., Dubé, M.-P., Hussin, J. G., and Bengio, Y. (2016). Diet networks: thin parameters for fat genomics. *arXiv preprint arXiv:1611.09340*.

Samstein, R. M., Lee, C.-H., Shoushtari, A. N., Hellmann, M. D., Shen, R., Janjigian, Y. Y., Barron, D. A., Zehir, A., Jordan, E. J., Omuro, A., et al. (2019). Tumor mutational load predicts survival after immunotherapy across multiple cancer types. *Nature Genetics*, 51(2):202–206.

Stovner, E. B. and Sætrom, P. (2020). PyRanges: efficient comparison of genomic intervals in Python. *Bioinformatics*, 36(3):918–919.

Wang, C. and Liang, C. (2018). MSIpred: a python package for tumor microsatellite instability classification from tumor mutation annotation data using a support vector machine. *Scientific Reports*, 8(1):1–10.

Yao, L., Fu, Y., Mohiyuddin, M., and Lam, H. Y. (2020). ectmb: a robust method to estimate and classify tumor mutational burden. *Scientific Reports*, 10(1):1–10.