

ARTICLE

<https://doi.org/10.1038/s41467-019-09027-x>

OPEN

Deep convolutional neural networks for accurate somatic mutation detection

Sayed Mohammad Ebrahim Sahraeian¹, Ruolin Liu¹, Bayo Lau¹, Karl Podesta², Marghoob Mohiyuddin¹ & Hugo Y.K. Lam¹

Accurate detection of somatic mutations is still a challenge in cancer analysis. Here we present NeuSomatic, the first convolutional neural network approach for somatic mutation detection, which significantly outperforms previous methods on different sequencing platforms, sequencing strategies, and tumor purities. NeuSomatic summarizes sequence alignments into small matrices and incorporates more than a hundred features to capture mutation signals effectively. It can be used universally as a stand-alone somatic mutation detection method or with an ensemble of existing methods to achieve the highest accuracy.

¹Roche Sequencing Solutions, Belmont, CA 94002, USA. ²Microsoft Azure, Dublin 18 D18 P521, Ireland. Correspondence and requests for materials should be addressed to H.Y.K.L. (email: hugo.lam@roche.com)

Somatic mutations are critical signatures in cancer genesis, progression, and treatment. Accurate detection of somatic mutations is challenging due to tumor-normal cross contamination, tumor heterogeneity, sequencing artifacts, and coverage. In general, effectively filtering false-positive calls, which are introduced by the aforementioned issues, and precisely keeping hard-to-catch true-positive calls, which may occur with low allele-frequency (AF) or occur in low-complexity regions, are crucial for an accurate somatic mutation detection algorithm.

To date, a range of tools have been developed to address the somatic mutation detection problem including MuTect2¹, MuSE², VarDict³, VarScan2⁴, Strelka2⁵, and SomaticSniper⁶. These tools employ different statistical and algorithmic approaches, which perform well in certain cancer or sample types for which they were designed; however, they are limited in generalization to a broader range of sample types and sequencing technologies and thus may exhibit suboptimal accuracy in such scenarios^{7–9}. In our earlier work, SomaticSeq¹⁰, we used an ensemble approach to maximize the sensitivity by integrating algorithmically orthogonal methods. It also used machine learning to integrate almost a hundred features to keep the precision high, leading to an accuracy improvement over all individual methods. Nevertheless, the machine-learning backbone used in SomaticSeq relies on a set of extracted features for the mutations' locations. As a result, it cannot fully capture the raw information in the genomic contexts of the somatic mutations to further distinguish true somatic mutations from background errors, limiting its performance in challenging situations, such as low-complexity regions and low tumor purity.

Here we address the limitation of generalizability and complexity of statistical modeling of tumor sequencing data by leveraging deep Convolutional Neural Networks (CNNs). CNNs have recently shown significant performance in classification problems from different domains including germline variant calling^{11–13} and skin cancer classification¹⁴. Even so, applying CNNs to the challenging problem of somatic mutation detection has not been explored. The only previous deep learning based attempt¹⁵ was to apply a six-layer fully connected neural network to a set of manually extracted features. This approach lacks the power provided by the CNN architecture, which is to learn feature representations directly from the raw data using patterns seen in local regions. Besides, due to the complexity of fully connected networks, it has less generalizability and scalability as seen in CNNs.

We introduce NeuSomatic, the first CNN-based approach for somatic mutation detection that can effectively leverage signals derived from sequence alignment, as well as from other methods to accurately identify somatic mutations. Unlike other deep learning based methods that are focused on germline variants, NeuSomatic is addressing a bigger unmet need in terms of accuracy due to the complexity of tumor samples. It can effectively capture important mutation signals directly from the raw data and consistently achieve high accuracy for different sequencing technologies, sample purities, and sequencing strategies such as whole-genome vs. target enrichment.

Results

NeuSomatic overview. The inputs to NeuSomatic's network are candidate somatic mutations identified by scanning the sequence alignments for the tumor sample as well as the matched normal sample (Fig. 1; Supplementary Figs. 1 and 2). Somatic mutations reported by other methods can also be included in this list of candidates. For each candidate locus, we construct a 3-dimensional feature matrix \mathbf{M} (of size $k \times 5 \times 32$), consisting of k channels each of size 5×32 , to capture signals from the region centered around that locus. Each channel of the matrix \mathbf{M} has five

rows representing the four nucleotide bases as well as the gapped base ('-'), and 32 columns representing the alignment columns around the candidate location.

The first three channels, respectively, are the reference, tumor-frequency, and normal-frequency channels that summarize the reference bases around the candidate locus, as well as the frequency of different bases in that region. We augment the reference sequence around the candidate locus with gaps corresponding to the insertions in the read alignments (Fig. 1a; Supplementary Figs. 1 and 2) in order to capture the insertions. Thus, each column of tumor and normal-frequency matrices represents the frequency of A/C/G/T/gap bases in the corresponding multiple sequence alignment (MSA) column of the tumor and normal samples, respectively. The remaining channels summarize other features, such as coverage, base quality, mapping quality, strand-bias, and clipping information for reads supporting different bases. If NeuSomatic is used in the ensemble mode, we also use additional channels for features reported by the individual somatic mutation detection methods. With this concise, yet comprehensive structured representation, NeuSomatic can use the necessary information in tumor, normal, and reference to differentiate difficult to catch somatic mutations with low AF from germline variants, as well as sequencing errors. This design also enables the use of convolutional filters in the CNN to capture contextual patterns in the sub-blocks of the matrix.

To compare to other CNN approaches used in genomics problems, DeepVariant¹¹ uses read pileup as the input for germline variant calling. In contrast, we use base frequency summary for each column as the input to our network. This simplifies the CNN structure, allowing a substantially more efficient implementation. For example, DeepVariant takes ~1000 CPU core-hours to call germline variants for a 30× whole-genome sample¹⁶, whereas the stand-alone version of NeuSomatic can detect somatic mutations from 30× tumor-normal pair samples in ~156 CPU core-hours, despite handling two (tumor-normal) samples instead of one and looking for candidates at lower somatic AFs than germline 50 or 100% AF. Another germline variant calling method, Clairvoyante¹², uses three channels to summarize base counts for allele counts, deletions, and insertions at the center of the window. In contrast, we summarize all these events in a single base frequency matrix using the reference augmentation approach described earlier, which can clearly represent all the insertion and deletion (INDEL) events across the window.

NeuSomatic employs a novel CNN structure that predicts the type and length of a candidate somatic mutation given the feature matrix \mathbf{M} (Fig. 1b). The proposed CNN consists of nine convolutional layers structured in four blocks with shortcut identity connections inspired by ResNet¹⁷ but with a different formation to adapt to the proposed input structure. We use two softmax classifiers and one regressor on the final layer. The first classifier identifies whether the candidate is a non-somatic call, SNV, insertion, or deletion. The second classifier predicts the length of the somatic mutation with four classes (0 indicating non-somatic, or lengths from 1, 2, or greater than 2), and the regressor predicts the location of the somatic mutation. Using the output of these classifiers we identify the set of somatic mutations. If the lengths of INDELS are predicted to be larger than 2, we perform a simple post-processing step on reads overlapping that position to resolve the INDEL sequence from the read alignment CIGAR string. This has been shown to perform well for data generated by Illumina sequencers. For higher error rate sequencing data, more complex local realignment post-processing is conducted to resolve the INDEL sequence.

Since NeuSomatic can be used in stand-alone and ensemble modes, we use NeuSomatic-S to denote the stand-alone mode, while reserving NeuSomatic to denote the ensemble mode. We

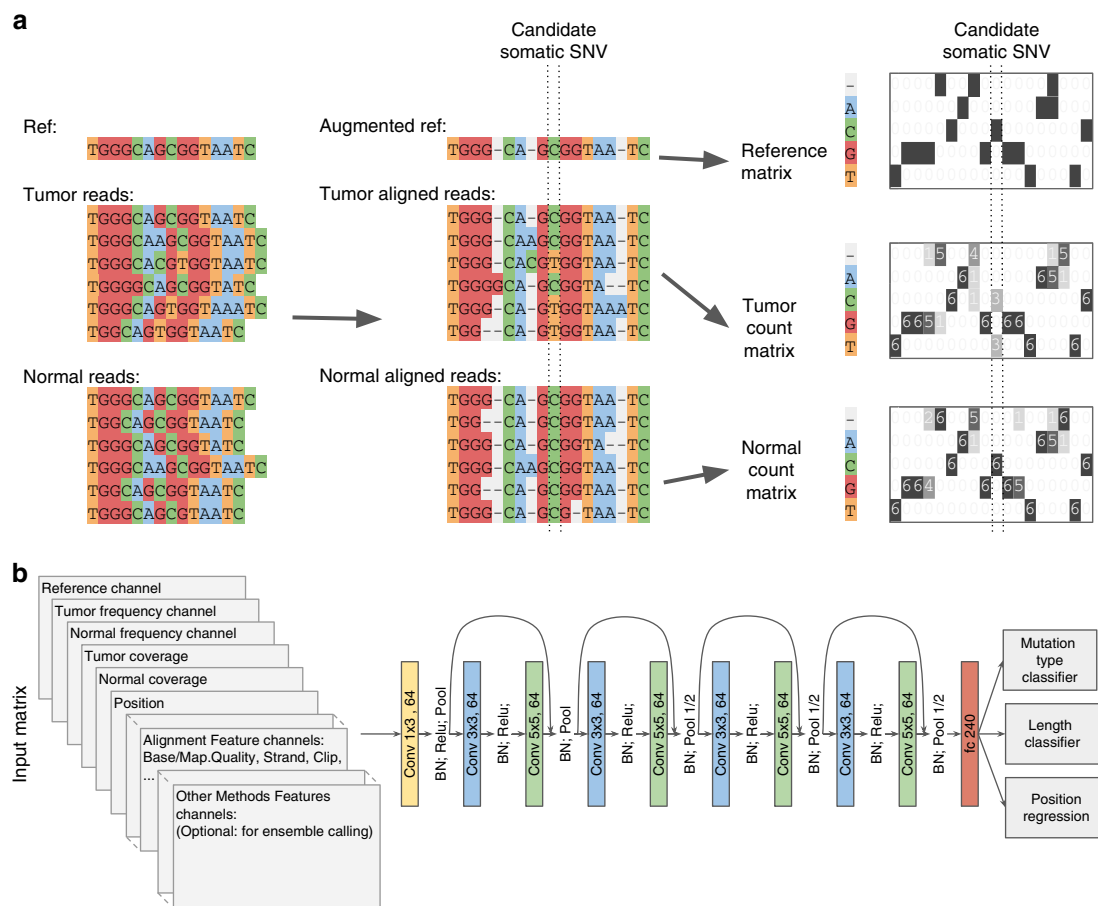


Fig. 1 NeuSomatic overview. **a** Toy example of input matrix preparation for a given candidate somatic SNV. Sequence alignment information in a window of seven bases around the candidate somatic mutation is extracted. The reference sequence is then augmented by adding gaps to account for insertions in the reads. The augmented alignment is then summarized into the reference matrix, the tumor count matrix, and the normal count matrix. The count matrices record the number of A/C/G/T and gap ('-') characters in each column of the alignment, while the reference matrix records the reference bases in each column. The count matrices are then normalized by coverage to reflect base frequencies in each column. Separate channels are reserved to record the tumor and normal coverages. **b** The input 3-dimensional matrix and the proposed NeuSomatic network architecture. The input matrix consists of reference channel, tumor and normal-frequency channels, coverage and position channels, followed by several channels summarizing the alignment features. When used in ensemble mode, NeuSomatic also includes additional channels for other individual methods features. NeuSomatic network architecture consists of nine convolutional layers structured in four blocks with shortcut identity connections. We use two softmax classifiers and one regressor on the final layer to predict the mutation type, size, and position

compared NeuSomatic and NeuSomatic-S against the state-of-the-art somatic mutation detection methods including MuTect2¹, MuSE², SomaticSniper⁶, Strelka2⁵, VarDict³, and VarScan2⁴, and against the ensemble approach, SomaticSeq¹⁰. We compared and contrasted performance using multiple synthetic and real datasets. We report below, the synthetic datasets in increasing order of somatic mutation detection difficulty considering the AF of somatic mutation in the datasets.

Comparison on the Platinum sample mixture dataset. For the first synthetic dataset, as in previous studies^{5,10} we mixed two normal Platinum Genomes¹⁸ samples, NA12877 and NA12878, at 70:30, 50:50, and 25:75 tumor purity ratios to create three tumor contamination profiles, and at 5:95 ratio to create a contaminated normal sample. We also included a test with 100% pure normal and 50% pure tumor. We used the germline variants in NA12878, which were reference calls in NA12877 as truth set for the evaluation. Both NeuSomatic-S and NeuSomatic significantly outperformed all other methods (Fig. 2 and Supplementary Table 1). NeuSomatic’s performance improvement over other approaches increased with lower, more challenging tumor

purities (25:75 mixture). In summary, NeuSomatic yielded up to 99.6 and 97.2% F1-scores for SNVs and INDELs, respectively, overall and an improvement of up to 7.2% over the best method in the lowest sample purity for this dataset. For the sample with 50% tumor purity, reducing normal purity from 100 to 95% had minor impact on NeuSomatic’s performance (<0.3%), whereas it caused ~3% decrease in SomaticSeq accuracy.

Comparison on the ICGC-TCGA DREAM challenge datasets. For the second synthetic dataset, we used the ICGC-TCGA DREAM Challenge Stage 3 and Stage 4 datasets¹⁹, which were constructed by computationally spiking mutations into a healthy genome of a paired normal sample with different AFs (See Methods). We mixed the tumor and normal samples to create five tumor/normal purity scenarios. NeuSomatic-S outperformed all other stand-alone methods for both Stage 3 and Stage 4 datasets by over 8% for SNVs and 22% for INDELs on average (Figs. 3 and 4; Supplementary Tables 2 and 3). This performance improvement increased with decreasing tumor purity. We further observed that NeuSomatic (the ensemble mode) clearly outperformed both SomaticSeq and NeuSomatic-S, even though

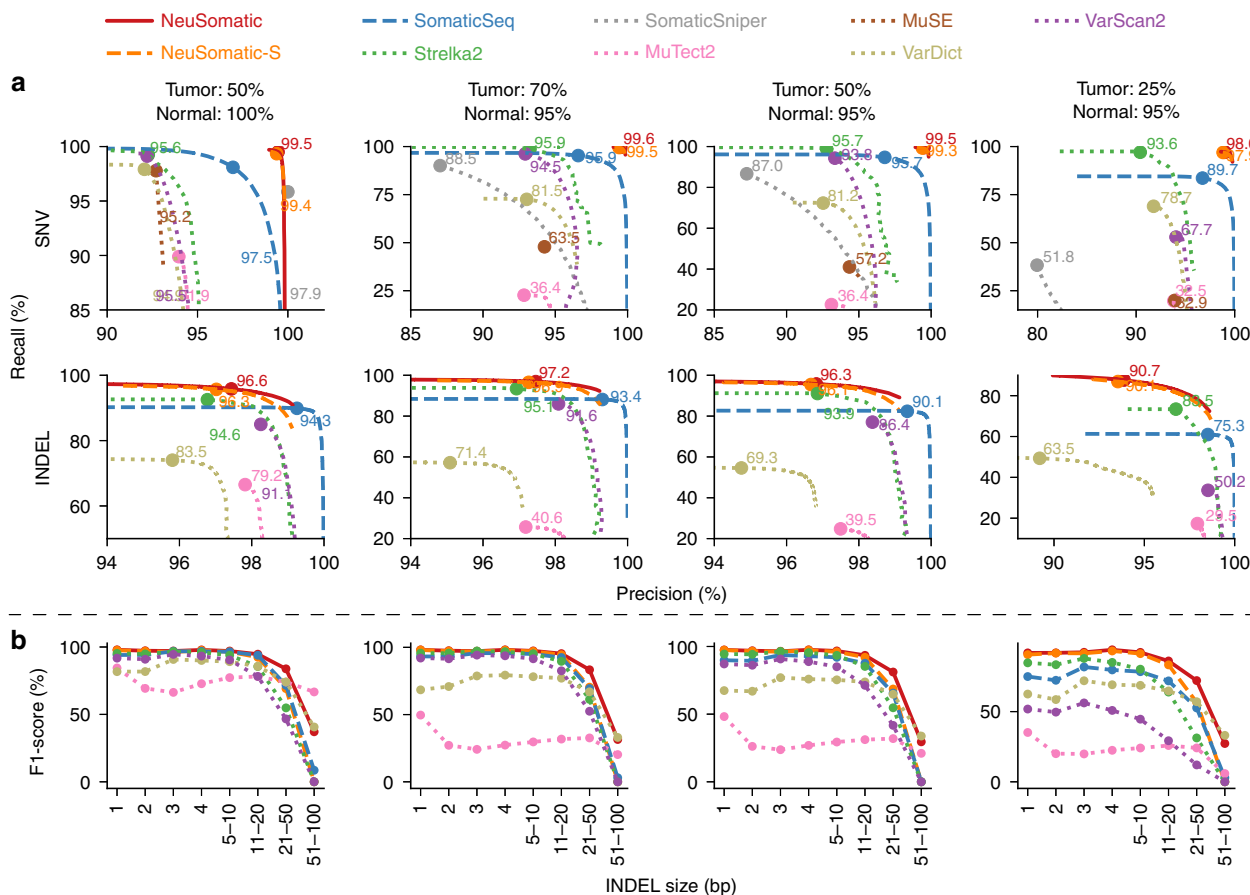


Fig. 2 Performance analysis of the Platinum two sample mixture dataset. In this dataset, four tumor and normal purity scenarios (50% T:100% N, 70% T:95% N, 50% T:95% N, and 25% T:95% N) are used. **a** Precision-recall analysis: the confidence or quality scores are used to derive the precision-recall curves. The highest F1-score achieved by each algorithm is printed on the curve and marked with a solid circle. **b** Performance analysis of INDEL accuracy (F1-score) for different INDEL sizes

NeuSomatic-S still outperformed SomaticSeq in more challenging scenarios, such as SNVs in the 25:75 mixture and INDELS in the 25:75 and 50:50 mixtures. In summary, NeuSomatic yielded up to 96.2 and 93.5% F1-scores for SNVs and INDELS, respectively, overall and an improvement of up to 34.6% over the best method in the lowest sample purity. For the sample with 50% tumor purity, reducing normal purity from 100 to 95% had minor impact on NeuSomatic's performance (~1.2% on average), whereas SomaticSeq and Strelka2 had >3% decrease in F1-score.

Comparison on the Platinum tumor spike dataset. For the third synthetic dataset, as in previous studies^{1,10}, we constructed a tumor sample by spiking reads from NA12878 into NA12877 in variant positions of NA12878 with spike in frequencies sampled from a binomial distribution with means [0.05, 0.1, 0.2, 0.3] and used an independent set of NA12877 reads as pure normal. Note that, unlike earlier strategy, which mixed samples in fixed proportions yielding somatic mutations at fixed AFs, this mixing approach generated them at varying AFs ranging from 0.025 to 0.3. NeuSomatic yielded 80.9 and 66.7% F1-scores for SNVs and INDELS, respectively, overall and an improvement of up to 4% over the best method (Supplementary Fig. 3; Supplementary Table 4). For low AF somatic mutations, the performance improvement was even higher (11% improvement for AF = 0.025 and 8% improvement for AF = 0.05) (Supplementary Fig. 3b).

Comparison on the whole-exome and targeted panels. To assess the performance of NeuSomatic on different target enrichments,

we used a whole-exome and a targeted panel dataset from the Ashkenazi Jewish trio²⁰ (Supplementary Figs. 4 and 5; Supplementary Tables 5 and 6). We trained NeuSomatic and SomaticSeq on the whole-exome dataset and applied the trained model on both the whole-exome and the panel. For whole-exome, NeuSomatic achieved up to 99.3 and 88.6% F1-scores for SNVs and INDELS, respectively. For the targeted panel, NeuSomatic and NeuSomatic-S consistently outperformed other methods with >99.2% F1-score for SNVs. Applying the model trained on whole-genome Platinum-mixture data on both target enrichment sets yielded similar performance, which confirmed the robustness of NeuSomatic (Supplementary Figs. 6 and 7). Similar to other datasets, for the sample with 50% tumor purity, reducing normal purity from 100 to 95% on whole-exome dataset could minimally reduce NeuSomatic's F1-score (~0.3% on average), whereas SomaticSeq and Strelka2 had >5% decrease in F1-score.

Comparison on the PacBio dataset. We further evaluated NeuSomatic's performance on reads with high error rates, particularly those from the long-read sequencing platforms. We used tumor-normal pair samples simulated with 20, 30, and 50% AF somatic mutations based on the raw PacBio reads (Fig. 5, Supplementary Table 7). NeuSomatic identified somatic SNVs and INDELS with F1-scores of up to 98.1 and 86.2%, respectively, which outperformed VarDict³ by up to 34.4% for SNVs and up to 53.2% for INDELS. This analysis confirms the capability of NeuSomatic in detecting somatic mutations even when the sequence reads have high error rate as in PacBio long raw reads.

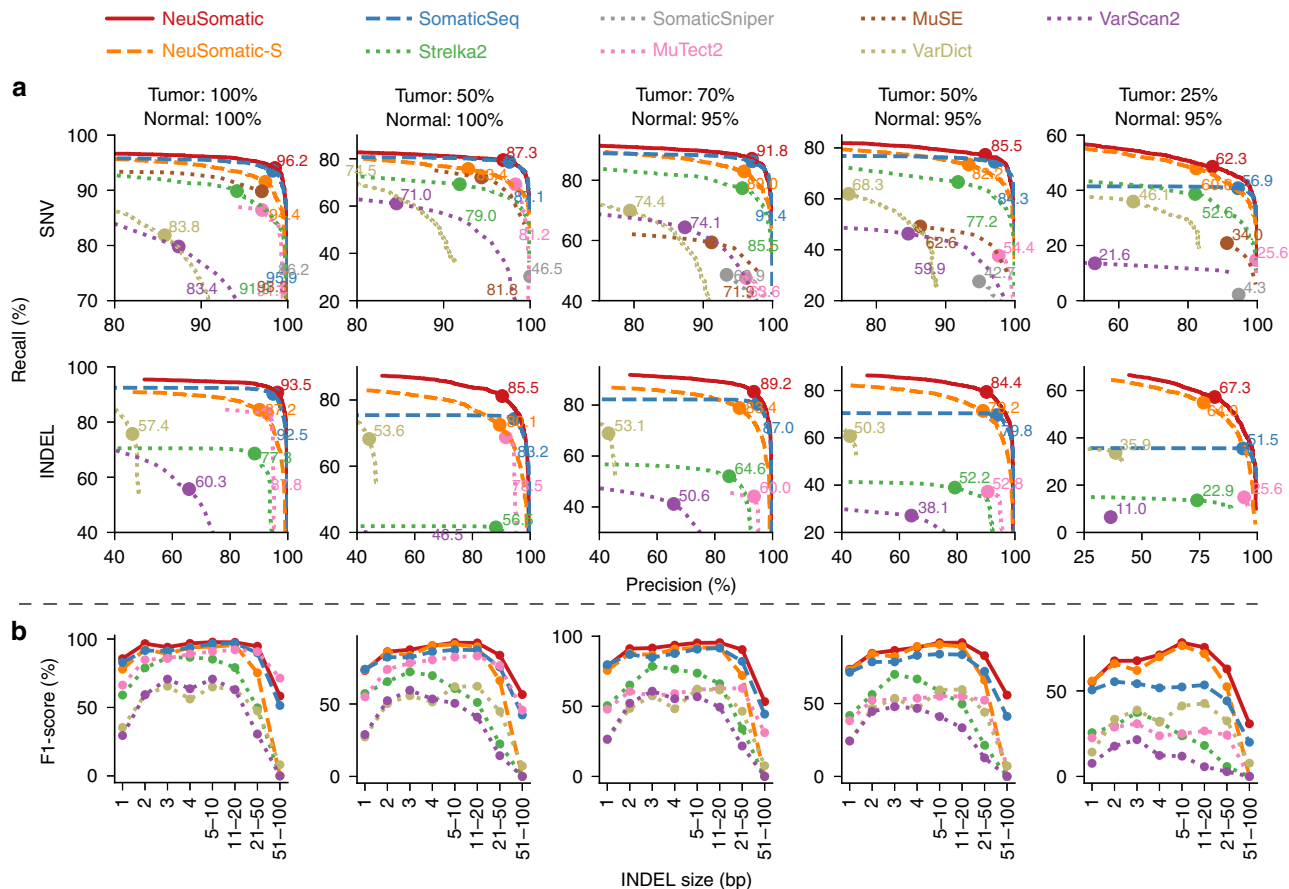


Fig. 3 Performance analysis of the DREAM Stage 3 dataset. In this dataset, five tumor and normal purity scenarios (100% T:100% N, 50% T:100% N, 70% T:95% N, 50% T:95% N, and 25% T:95% N) are used. **a** Precision-recall analysis: the confidence or quality scores are used to derive the precision-recall curves. The highest F1-score achieved by each algorithm is printed on the curve and marked with a solid circle. **b** Performance analysis of INDEL accuracy (F1-score) for different INDEL sizes

Comparison for different INDEL sizes. It is worth noting that NeuSomatic consistently outperformed other methods for various INDEL sizes in different datasets (Figs. 2b, 3b, 4b, 5b; Supplementary Figs. 3c and 8). For large (>50 bases) INDELS, since most of the short reads with somatic INDELS are soft-clipped, the INDEL information is lost in the pileup count matrices. For such cases, NeuSomatic benefited from other methods' predictions, since some of the methods like VarDict and MuTect2 used local assembly for their predictions.

INDEL type and position accuracy. For all datasets discussed, we also assessed the performance of INDEL calling by different somatic mutation detection methods using the more relaxed criterion of simply predicting the positions of the somatic INDELS correctly (and ignoring the exact INDEL sequence). Again, we observed similar superiority of NeuSomatic over other schemes indicating that the main improvements are contributed by the proposed CNN structure and not the post-processing INDEL resolution steps (Supplementary Figs. 9 and 10).

Read coverage analysis. To evaluate the impact of sequence coverage on different techniques, we downsampled the whole-exome dataset to obtain samples with sequence coverages in the range of 20x and 100x (Fig. 6, Supplementary Fig. 11). NeuSomatic consistently outperformed other techniques for different coverages. The improvement increased as the problem became more challenging for lower coverages samples. In addition, reducing the coverage from 100x to 50x had very minimal impact

(~1.5% for SNVs and ~5% for INDELS) on NeuSomatic, whereas SomaticSeq's F1-score dropped by ~20% for both SNVs and INDELS, and Strelka2's F1-score dropped by ~13% for SNVs and ~15% for INDELS. This analysis revealed both the robustness of NeuSomatic to coverage perturbations, as well as its advantage in challenging scenarios, which could be seen at lower coverages.

Training robustness. We assessed the robustness of NeuSomatic's training for specific purity by training and testing on different purities for the DREAM Challenge Stage 3 datasets. Supplementary Fig. 12 shows that performance degrades only marginally even when we trained and tested on very different tumor purities. We also observed that training using data aggregated from multiple tumor purities was as good as training on the target tumor purity (Supplementary Fig. 12). This suggests that a training set incorporating multiple tumor purities is sufficient to get a model, which is robust to tumor purity variation.

Comparison on real data. In the absence of a high-quality, comprehensive ground truth dataset for somatic mutations²¹, like the Genome-in-a-Bottle gold set for germline variants²², we would not be able to calculate F1 accuracy outside of synthetic data. Fortunately, there are existing datasets with validated somatic mutations we could take to estimate the accuracy performance of NeuSomatic on real data (See Methods for more details on how to extrapolate the F1-score on real data). We used two datasets: CLL1²³, a chronic lymphocytic leukemia patient whole-genome data with 961 validated somatic SNVs, and

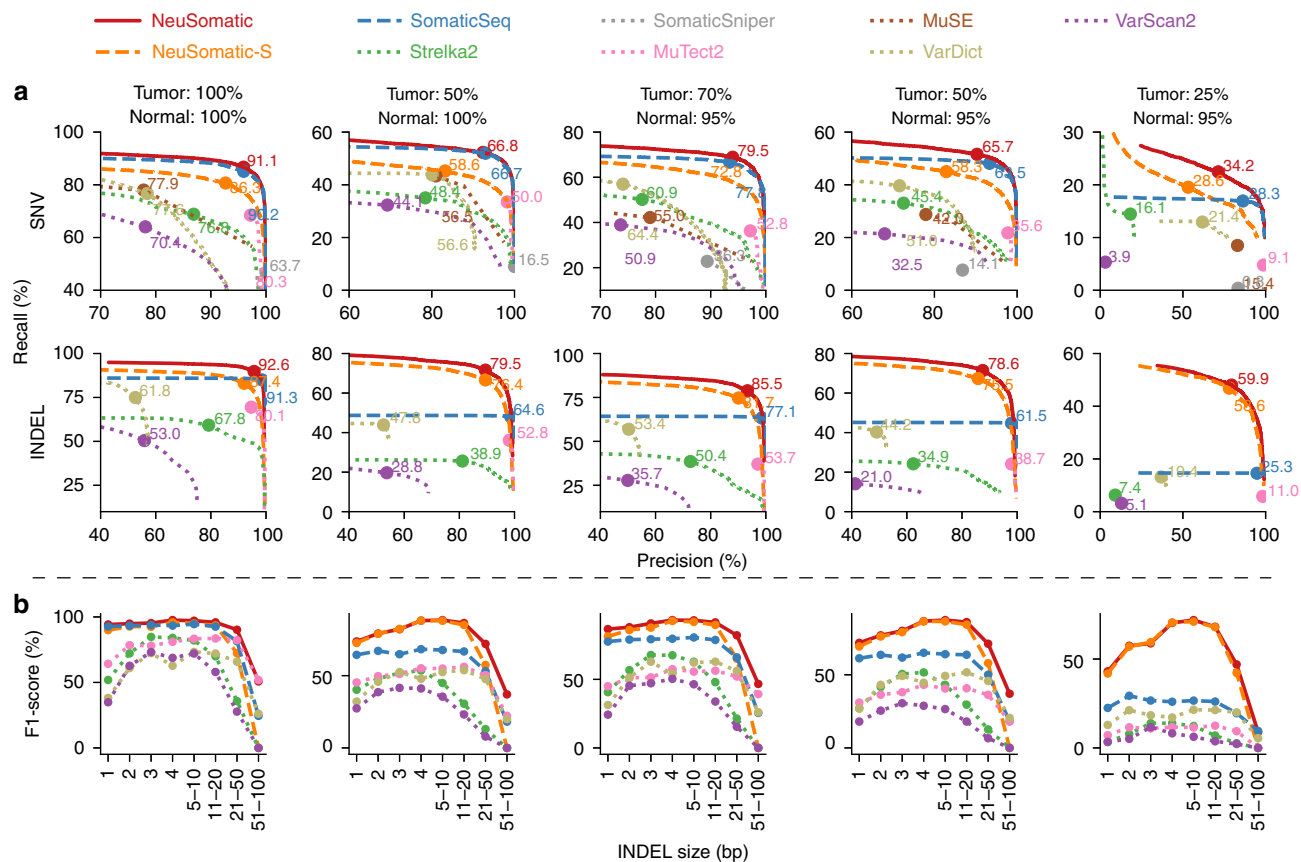


Fig. 4 Performance analysis of the DREAM Stage 4 dataset. In this dataset, five tumor and normal purity scenarios (100% T:100% N, 50% T:100% N, 70% T:95% N, 50% T:95% N, and 25% T:95% N) are used. **a** Precision-recall analysis: the confidence or quality scores are used to derive the precision-recall curves. The highest F1-score achieved by each algorithm is printed on the curve and marked with a solid circle. **b** Performance analysis of INDEL accuracy (F1-score) for different INDEL sizes

COLO-829^{24,25}, an immortal metastatic malignant melanoma cell line-derived whole-genome dataset with 454 validated somatic SNVs. To evaluate NeuSomatic on these two real WGS samples, we used models trained on the DREAM Challenge Stage 3. As shown in Supplementary Tables 8 and 9, NeuSomatic achieved the highest extrapolated F1-score of 99.7 and 93.2%, respectively, for the COLO-829 malignant melanoma sample and the CLL1 chronic lymphocytic leukemia sample. We also evaluated NeuSomatic on a TCGA^{26,27} whole-exome sequencing (WES) sample of colorectal adenocarcinoma (TCGA-AZ-6601), achieving the highest extrapolated F1-score of over 99.6% (Supplementary Table 10).

In order to demonstrate NeuSomatic's scalability and cost effectiveness on the cloud, we also processed 261 whole-exome sequenced cancer samples (Supplementary Table 11) from TCGA on the Microsoft Azure cloud platform using both the ensemble and stand-alone modes. These samples were taken across multiple cancer types including colorectal adenocarcinoma, ovarian serous adenocarcinoma, cervical squamous cell carcinoma, and endocervical adenocarcinoma. While the cloud platform enabled us to automatically spin up compute instances on demand, it took, on average, 2.42 hours and 0.72 hours for ensemble and stand-alone modes, respectively, to process each sample. Using Azure's preemptible compute instances (the standard H16 instance types were used with 16 cores each) resulted in low per sample processing costs of 0.77 USD and 0.23 USD for the ensemble and stand-alone modes, respectively. We also assessed the accuracy of NeuSomatic on these samples by comparing against the 44,270 validated SNPs across these samples, which provided us with

recall rates of 98.9 and 97.2% for ensemble and stand-alone modes, respectively. Thus, NeuSomatic not only can be used on different sequencing technologies or sequencing strategies but also can be run on a variety of compute platforms including a local HPC cluster and on an elastic cloud compute infrastructure.

Discussion

NeuSomatic is the first deep learning based framework for somatic mutation detection, which is high-performing and universal. While using the same CNN architecture, it achieves the best accuracy for varying tumor purities across multiple datasets ranging from synthetic to real, across multiple sequencing strategies ranging from whole genome to targeted as well as across multiple sequencing technologies ranging from short reads to high-error long reads. Specifically, for low tumor purities and low allelic frequencies, NeuSomatic significantly outperforms other state-of-the-art somatic mutation detection methods, thus demonstrating its capability in addressing the hard problem. NeuSomatic utilizes an efficient implementation of convolutional neural networks for solving the somatic mutation detection problem with speed and accuracy. It uses a novel summarization of tumor/normal alignment information as a set of input matrices that can effectively capture main signals in the genomic context. Training the proposed CNN architecture on these matrices enables learning feature representations directly from the raw data. The deep features, learned from observed training data, can accurately identify the important mutation signatures that can differentiate true calls from artifacts introduced by sequencing

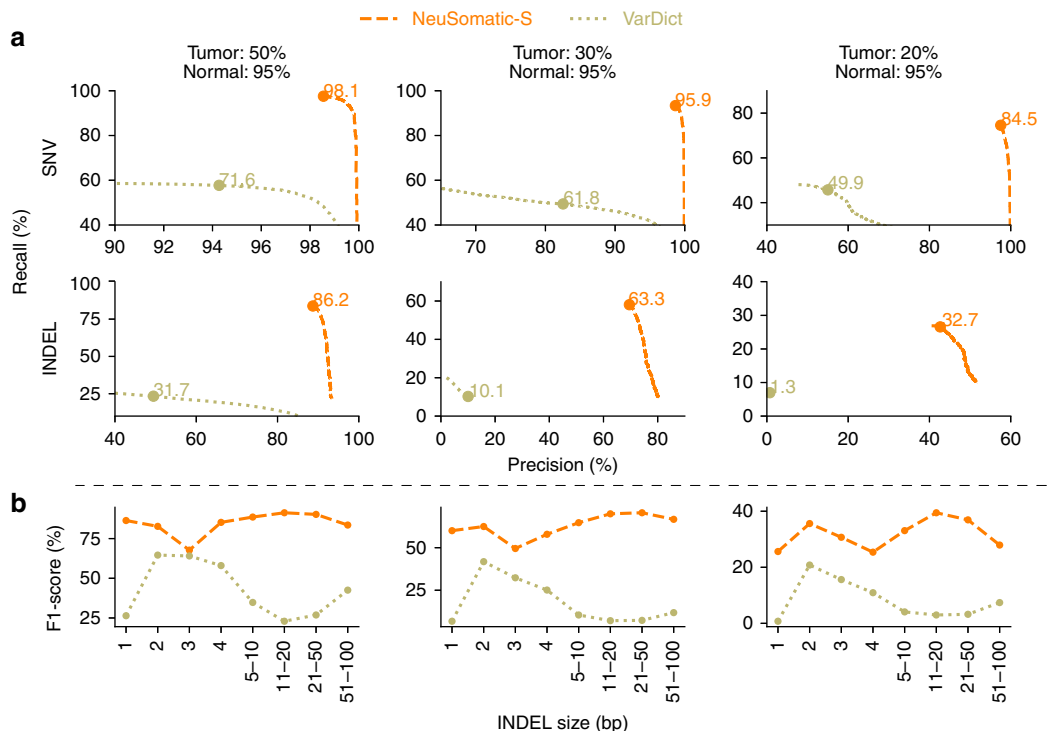


Fig. 5 Performance analysis of the PacBio dataset. In this dataset, three tumor and normal purity scenarios (70% T:95% N, 50% T:95% N, and 25% T:95% N) are used. **a** Precision-recall analysis: the confidence or quality scores are used to derive the precision-recall curves. The highest F1-score achieved by each algorithm is printed on the curve and marked with a solid circle. **b** Performance analysis of INDEL accuracy (F1-score) for different INDEL sizes

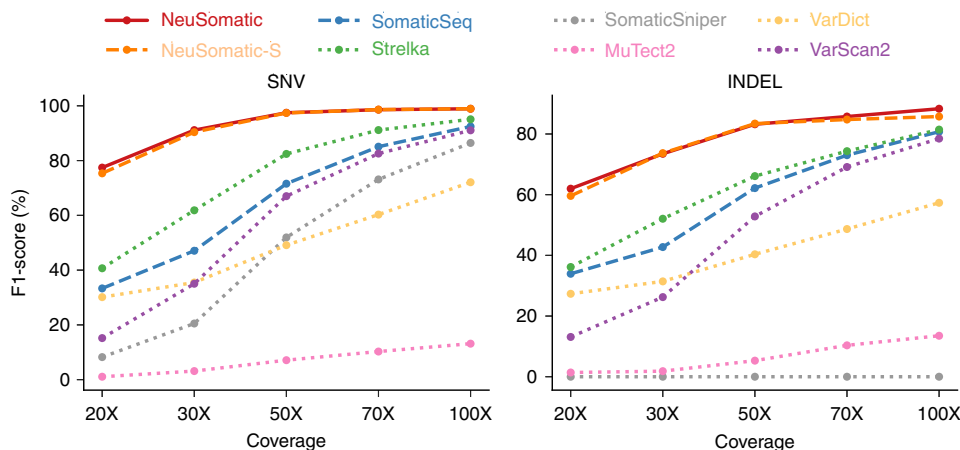


Fig. 6 Performance analysis of the sequence coverage impact on the whole-exome sample mixture dataset. In this example, tumor has 50% purity and normal has 95% purity. Y-axis illustrates the highest F1-score achieved by each algorithm for sample alignments coverages ranging from 20× to 100×

errors, cross contamination, or coverage biases. We believe NeuSomatic advances the state-of-the-art significantly by providing a very broadly applicable approach for somatic mutation detection.

Methods

ICGC-TCGA DREAM Challenge data. Stage 3 data consist of a normal sample and a tumor sample constructed by computationally spiking 7,903 SNVs and 7,604 INDELS mutations into a healthy genome of the same normal sample with three different AFs of 50, 33, and 20% to create synthetic but realistic tumoral normal pairs. Stage 4 data have similar formation, but with 16,268 SNVs and 14,194 INDELS in two subclones of 30 and 15% AF. We then constructed an impure normal by mixing 95% normal and 5% tumor reads. We also constructed four tumor mixtures by mixing tumor and normal, respectively, at 100:0, 70:30, 50:50, and 25:75 ratios. Thus, the somatic mutations across these four tumor mixture ratios have AFs ranging from 5 to 50% for Stage 3 dataset, and 3.75 to 30% for Stage 4 dataset.

Platinum synthetic tumor data. We downloaded 200× Platinum genomes samples NA12878 and NA12877 and their truth germline variants (v2017-1.0)¹⁸ to construct a virtual tumor and normal pair (ENA accession number PRJEB3246). For the normal, we downsampled NA12877 to 50×. For tumor, we constructed three 50× in silico mixture samples with 70, 50, and 25% tumor purities, by independently downsampling NA12877, respectively, at 15×, 25×, and 37.5×, and mixing each with downsampled NA12878 at 35×, 25×, and 12.5×. We use the heterozygous and homozygous variants in NA12878, which are reference calls in NA12877 and are at least five bases apart from NA12877 variants and 300 base apart from each other as the truth set for the training and evaluation steps (1,103,285 SNVs and 174,754 INDELS). Thus, depending on the zygosity of the germline variants in NA12878, somatic mutations across these three tumor mixture ratios have AFs ranging from 12.5 to 70%.

We also generated another 50× virtual tumor sample by randomly spiking reads from a downsampled (to 50× coverage) NA12878 into a downsampled (to 50× coverage) NA12877 data at heterozygous and homozygous variant locations in NA12878, which are reference calls in NA12877. For each variant, we randomly assigned the frequencies of spiked reads from a binomial distribution with means

[0.05, 0.1, 0.2, 0.3]. Thus, depending on the zygosity of the variant, the mean somatic mutations AFs ranges from 2.5 to 30%. To avoid ambiguity in the truth set, we only used variants for which the relevant paired-end reads did not overlap any other variants (316,050 SNVs and 46,978 INDELS). This generated a contaminated tumor with reads from NA12878. We also used another independent downsampled (to 50×) data for NA12877 as the pure normal.

For both experiments, FastQ files and truth germline variants were downloaded and aligned with BWA-MEM (v0.7.15)²⁸ followed by Picard MarkDuplicates (v2.10.10) (<https://broadinstitute.github.io/picard>), and GATK IndelRealigner and Base Quality Score Recalibration (v3.7)²⁹.

Real tumor-normal pair data. We used the CLL1 chronic lymphocytic leukemia dataset²³ (accession: <https://www.ebi.ac.uk/ega/datasets/EGAD00001000023>) and the COLO-829 immortal metastatic malignant melanoma cell line dataset^{24,25} (accession: <https://www.ebi.ac.uk/ega/studies/EGAS00000000052>) to assess our approach on real tumor-normal pair data with published lists of validated somatic mutations.

The COLO-829 dataset consists of 80× whole-genome sequencing tumor sample and its matched normal blood COLO-829BL sample at 60×, with 454 validated somatic SNVs. CLL1 has a whole-genome sequencing tumor sample and a matched normal, respectively, at 53× and 42× coverage, with 961 published somatic SNVs.

The TCGA-AZ-6601^{26,27} dataset is a whole-exome sequencing of a colon adenocarcinoma tumor sample and its matched normal tissue from TCGA. The tumor and normal samples were sequenced at depths of 145× and 165×, respectively. We used 952 validated SNVs in TCGA³⁰ and COSMIC³¹ databases as the ground truth somatic mutations for this sample.

For real data, we compute extrapolated precision as the percentage of predicted somatic mutations that have been called by at least two stand-alone methods, or have been reported as verified somatic mutations in at least two samples of the same cancer type in COSMIC database. We then compute extrapolated F1-score based on the harmonic mean of recall and this extrapolated precision.

Whole-exome and targeted panel data. To assess NeuSomatic on different target enrichment experiments we used whole-exome datasets from the Ashkenazi Jewish trio²⁰. We downloaded deep-sequenced (200× coverage) whole-exome alignment files for HG003 and HG004 (<ftp://ftp-trace.ncbi.nlm.nih.gov/giab/ftp/>), along with the high-confidence germline variants (Genome-in-a-Bottle release v3.3.2). We then used mixtures of random 70×, 50×, and 25× downsamples of HG004 and 30×, 50×, and 75× downsamples of HG003, to construct 70, 50, and 25% pure tumor samples, respectively. We also constructed a 95% pure normal by mixing 95× HG003 and 5× HG004 downsampled alignments. For our analysis, we used Agilent SureSelect Human All Exon V5 BED file. The ground truth somatic mutations were identified similar to the Platinum synthetic tumor data (11,720 SNVs, 878 INDELS). Depending on the zygosity of the germline variants in HG004, somatic mutations across these three tumor mixture ratios have AFs ranging from 12.5 to 70%.

For validating the performance on the target panel, we restricted the above alignment and truth data to Illumina's TruSight inherited disease panel BED file (216 SNVs, 5 INDELS). We only evaluated the performance on SNVs due to the limited number of true INDELS in the target panel region.

PacBio data. For long-reads analysis, we downloaded the high-confidence germline variants (Genome-in-a-Bottle release v3.3.2) for HG002 sample (<ftp://ftp-trace.ncbi.nlm.nih.gov/giab/ftp/>)²⁰. We built the long-reads error profile using the CHM1 dataset³² (SRA accession SRX533609). We then simulated a 100× pure normal sample using the VarSim simulation framework³³ in combination with the LongISLND in silico long-reads sequencer simulator³⁴. Using a set of random somatic mutations, we also simulated a 100× pure tumor sample with the same error profile. We used NGMLR (v0.2.6)³⁵ to align the sequences. We then mixed a 47.5× downsample of pure normal alignment and 2.5× downsample of the pure tumor alignment to form the 50× normal pair with 95% purity, and mixed 40×, 35×, and 25× independent downsamples of normal, respectively, with 10×, 15×, and 25× downsamples of pure tumor, to construct 50× tumor mixtures of 20, 30, and 50% purity. We restricted the training set to a 120 megabase region in chromosome 1 (with 39,818 truth somatic SNVs and 38,804 truth somatic INDELS), and the testing set to whole chromosome 22 (with 12,201 truth somatic SNVs and 12,185 truth somatic INDELS). Somatic mutations across the three tumor mixture ratios have AFs ranging from 20 to 50%.

Candidate mutation preparation. As the first step, we scan tumor read alignments to find candidate locations with evidence of mutations. Many of these positions have either germline variants or erroneous calls made due to the complexity of the genomic region, or sequencing artifacts. We apply a set of liberal filters on the set of candidate locations to make sure the number of such locations is reasonable. In general, for SNVs, we required AF ≥ 0.03 or more than two reads supporting the SNV and Phred scaled base quality score larger than 19 (larger than 14 for real WES dataset) as the minimum requirements. For 1-base INDELS, we required AF ≥ 0.02 or more than one read support. For INDELS larger than 1-base, we require

AF ≥ 0.03 . For the ensemble approach, we also included any somatic mutation detected by other somatic mutation detection methods as input candidate. For the PacBio dataset, we used AF ≥ 0.1 for SNVs and INDELS larger than 1-base, and AF ≥ 0.15 for 1-base INDELS.

For the DREAM Challenge dataset, we excluded variants existing in dbSNP³⁶ from the input candidates. For fair comparison, we also filtered dbSNP calls for all other somatic mutation detection tools.

Input mutation matrix. For each candidate position, we prepare a 3-dimensional matrix \mathbf{M} with k channels of size 5×32 (Fig. 1a; Supplementary Figs. 1 and 2). The five rows in each channel corresponds to four DNA bases A, C, G, and T, and the gap character ('-'). Each of the 32 columns of the matrix represents one column of the alignment.

For each candidate location, we extract the tumor and normal read alignments. As shown in Fig. 1a, we then consider the read alignments of tumor and normal sample to the reference as an MSA. To this end, we augment the reference sequence by adding gaps to the reference sequence, when there is insertion in reads. It must be noted that this process does not need any further realignment of the original read alignments of input BAM files, but only restructuring the alignments into MSA format by assigning additional columns wherever insertions has occurred. If there are multiple distinct insertions in multiple reads after a specific position, we consider them as left-aligned sequences and put them in the same set of columns (See for instance insertions of A and C bases in the ninth column of the toy example in Fig. 1a). With this read representation, we find the frequency of A/C/G/T/- characters in each column and record separate matrices for tumor and normal samples (channels C_2 and C_3 in matrix \mathbf{M}). In channel C_1 , we record the reference base (or gap) in each column. Channels C_i ($4 \leq i \leq k$) record other alignment signals in tumor and normal samples, such as coverage, base quality, mapping quality, strands, clipping information, edit distance, alignment score, and paired-end information. For instance, for the base quality channel, we have a matrix of size 5×32 for each sample, which records the average base quality of reads that have a given base (for a given row) in each column. As another instance, for the edit distance channel, we have a matrix of size 5×32 for each sample, which records the average edit distance of reads that have a given base (for a given row) in each column. One channel of matrix \mathbf{M} is devoted to specify the column where the candidate is located in. In the current implementation, we used a total of 26 channels in the stand-alone NeuSomatic-S approach.

For the ensemble extension of NeuSomatic, we also included additional channels to capture features reported by each of the six individual methods used. In this implementation, we used 93 additional channels to represent features extracted from other methods, and alignments reported by SomaticSeq. Thus, the ensemble mode of NeuSomatic had 119 input channels for each candidate matrix.

For each candidate location, we report the alignment information in a window of seven bases around the candidate. We reserve 32 columns to take into account the augmented alignment with insertions. In rare cases where we have a large insertion, 32 columns may not be enough to represent the alignment. For such cases, we truncate the insertions so that we can record at least three bases in the vicinity of the candidate.

CNN architecture. The proposed CNN (Fig. 1b) consists of nine convolutional layers structured as follows. The input matrices are fed into the first convolution layer with 64 output channels, 1×3 kernel size and Relu activation followed by a batch normalization and a max-pooling layer. The output of this layer is then fed to a set of four blocks with shortcut identity connection similar to ResNet structure. These blocks consist of a convolution layer with 3×3 kernels followed by batch normalization and a convolution layer with 5×5 kernels. Between these shortcut blocks, we use batch normalization and max-pooling layers. The output of final block is fed to a fully connected layer of size 240. The resulting feature vector is then fed to two softmax classifiers and a regressor. The first classifier is a 4-way classifier that predicts the mutation type from the four classes of non-somatic, SNV, insertion, and deletion. The second classifier predicts the length of the predicted mutation from the four categories of 0, 1, 2, and ≥ 3 . Non-somatic calls are annotated as zero size mutations, SNVs and 1-base INDELS are annotated as size 1, while 2-base and ≥ 3 size INDELS are, respectively, annotated as 2 and ≥ 3 size mutations. The regressor predicts the column of the mutations in the matrix, to assure the prediction is targeted the right position and is optimized using a smooth L1 loss function.

The CNN has less than 900 K parameters, which enables us to have a highly efficient implementation by using large batch sizes. The whole-genome training process took ~ 8 h on a machine with 8 Tesla K80 Nvidia GPU's.

CNN training. For DREAM Challenge, Platinum, and target enrichment datasets, we randomly split the genomic regions to 50% training and 50% testing sets. For the PacBio dataset, we trained NeuSomatic on a 120 megabase region on chromosome 1, and tested it on all of chromosome 22.

For each dataset, we combined the generated training input matrices from all different tumor/normal purity scenarios, and used the combined set for training the network. We then applied this unified trained model to test in each individual tumor/normal purity setting.

The DREAM Challenge dataset has 15,507 somatic mutations for Stage 3 and 30,462 somatic mutations for Stage 4. For better network training, we spiked in ~95 K more SNVs and ~95 K more INDELS with similar AF distributions to the original DREAM data into the tumor samples of Stages 3 and 4 using BAMSurgeon¹⁹.

We trained the network using a batch size of 1000 with SGD optimizer with learning rate of 0.01, and momentum of 0.9, and we multiplied the learning rate by 0.1 every 400 epochs.

Since, in general, the input candidate locations have significantly more non-somatic (reference or germline) calls than true somatic mutations, in each epoch we use all the true somatic mutations in the training set and randomly selected non-somatic candidates with twice the number of the true somatic mutations. We used a weighted softmax classification loss function, to balance for the number of candidates in each category. For DREAM Challenge data, since we added more synthetic mutations in the training set, we boosted the weight for the non-somatic category to achieve higher precision on test set.

For assessing synthetic target enrichment datasets, we used whole-exome and whole-genome data as the training set.

To test on real WGS samples CLL1 and COLO-829, we used models trained on DREAM Challenge Stage 3 for SomaticSeq and NeuSomatic. For the real WES sample TCGA-AZ-6601, we prepared a training set using data from another TCGA WES dataset, TCGA-AZ-4315³⁰. We mixed the tumor and normal alignments from this dataset and split the mixture into two equal alignments. We then used one alignment as the pure normal and spiked in ~91 K random SNVs and ~9 K random INDELS into the other alignment using BAMSurgeon to generate a synthetic tumor sample for training. We used models trained on this synthetic tumor-normal WES dataset to test NeuSomatic and SomaticSeq on the real WES dataset, TCGA-AZ-6601. For the experiment on 261 real TCGA samples, we used a similar approach to prepare a training set using 12 TCGA samples. The models trained on this synthetic dataset were used to test on the 261 TCGA samples.

Hyper-parameter tuning. For hyper-parameter tuning, we used 10% of the genome in the DREAM Challenge Stage 3 experiment and used the derived parameters in all other experiments.

We further explored different network architectures such as the pre-activation ResNet architecture with 4 to 16 ResNet blocks (including ResNet-18 and ResNet-34 architectures) (Supplementary Fig. 13a–e), as well as some variants of the proposed residual NeuSomatic architecture (Supplementary Fig. 13f–m). To evaluate these networks, we split the training data in the DREAM Stage 3 dataset into two halves and used one to train different architectures and the other to evaluate them in the stand-alone mode. Supplementary Table 12 compares these architectures in terms of accuracy, number of network parameters, memory usage, and speed. In general, all these networks can obtain relatively high accuracy compared to the conventional somatic mutation detection approaches. This observation revealed the importance of the proposed data summarization approach, which captures main signals in the genomic context of the candidates and facilitates efficient implementation of convolutional networks on the somatic mutation detection problem. The default ResNet architectures with two 3×3 convolution filters (Supplementary Fig. 13a–e) have lower average accuracy compared to those with the proposed residual blocks (Supplementary Fig. 13f–m). In addition, networks with strided convolution (Supplementary Fig. 13a–g) have larger number of network parameters and run-time requirements. In summary, although each network architecture shows advantages in some of the compared aspects, we selected the proposed NeuSomatic network architecture (Fig. 1b; Supplementary Fig. 13k) as our default network architecture as a compromise of all these factors, while other networks can easily be adapted by users given their use-cases and time/computational constraints.

Other somatic mutation detection algorithms. We used Strelka2 (v2.8.4), Mutect2 (v4.0.0.0), SomaticSniper (v1.0.5.0), MuSE (v1.0rc), VarDict (v1.5.1), VarScan2 (v2.3.7), and SomaticSeq (v2.7.0) somatic mutation detection algorithms in our analysis, with their default settings.

We used VarDict as an alternative approach to NeuSomatic on PacBio data. To enable detecting somatic mutations on high-error rate long reads, we used VarDict with “-m 10000 -Q 1 -q 5 -X 1 -c 1 -S 2 -E 3 -g 4 -k 0” parameter settings. Besides, as in NeuSomatic, we used AF ≥ 0.1 for SNVs and AF ≥ 0.15 for INDELS.

To train SomaticSeq, we also followed the same 50% train/test region splitting as used for NeuSomatic. In addition, as in NeuSomatic, for each dataset we combined the training data from all different tumor/normal purity scenarios to train the SomaticSeq SNV and INDEL classifiers. These unified classifiers were then used to predict in each individual tumor/normal purity setting.

For the precision-recall analysis, somatic mutations were sorted based on the confidence or quality scores assigned by each tool. For MuSE, we used the tier assignments as the sorting criterion. For VarDict, VarScan2, MuTect2, Strelka2, and SomaticSniper we, respectively, used SSF, SSC, TLOD, SomaticEVS, and SSC values reported in the VCF file for sorting. For SomaticSeq and NeuSomatic, we used the somatic mutation quality score in the QUAL field. NeuSomatic reports quality scores for the predicted somatic mutations based on the probability of predictions by CNN.

To analyze performance on real samples, we used the PASS somatic calls from different methods (For VarDict we restricted to calls with StrongSomatic status). For NeuSomatic, we used 0.97 as the quality score threshold for WGS and 0.6 for WES.

Computational complexity. For whole-genome data, scanning $30\times$ tumor and normal alignments to find candidates, extracting features, and preparing the input matrices can take ~3.9 h on a dual-14 core Intel Xeon CPU E5-2680 v4 2.40 GHz machine. The whole-genome training process can take ~8 h on a machine with 8 Tesla K80 Nvidia GPU's (~90 s per epoch of size 580,000). Depending on the cutoff AF on candidate somatic mutations, computing the network predictions for the candidate mutations on a $30\times$ whole-genome data can take ~35 min (with AF cutoff of 0.05, 3.9 M candidates) to ~100 min (with AF cutoff of 0.03, 11.5 M candidates) with 8 Tesla K80 Nvidia GPUs. For $125\times$ whole-exome data, the whole scanning, preparation, and computing the network predictions can take ~30 min. The end-to-end run times for predicting somatic mutations on a $125\times$ whole-exome dataset and a $30\times$ whole-genome dataset using NeuSomatic ensemble and stand-alone approaches (in CPU-only mode) were compared with other somatic mutation detection techniques in Supplementary Figs. 14 and 15.

Code availability. NeuSomatic is written in Python and C++. Its deep learning framework is implemented using PyTorch 0.3.1 to enable GPU application for training/testing. The source code is available at <https://github.com/bioinform/neusomatic> under the Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International license. The results in this paper were based on NeuSomatic v0.1.3.

Reporting summary. Further information on experimental design is available in the Nature Research Reporting Summary linked to this article.

Data availability

Sequence data for this study were collected from various sources, i.e., the European Nucleotide Archive (accession: PRJEB3246; <https://www.ebi.ac.uk/ena>), the Sequence Read Archive (accession: SRX1026041; <https://www.ncbi.nlm.nih.gov/sra>), the International Cancer Genome Consortium (project: ICGC-TCGA DREAM Mutation Calling Challenge, controlled access: <https://icgc.org/>), The Cancer Genome Atlas (accessions: TCGA-AZ-6601, TCGA-AZ-4315; controlled access: <https://gdc.cancer.gov/>), the European Genome-phenome Archive (accessions: EGAS00000000052, EGAD00001000023; controlled access: <https://www.ebi.ac.uk/ega/>), and the Genome-in-a-Bottle (accessions: HG002, HG003, HG004; <ftp://ftp-trace.ncbi.nlm.nih.gov/giab/ftp/>). Synthetic data were generated from the above datasets using the scripts at https://github.com/bioinform/neusomatic/blob/paper/etc/data_scripts.zip. All other relevant data are available upon request.

Received: 14 November 2018 Accepted: 14 February 2019

Published online: 04 March 2019

References

- Cibulskis, K. et al. Sensitive detection of somatic point mutations in impure and heterogeneous cancer samples. *Nat. Biotechnol.* **31**, 213 (2013).
- Fan, Y. et al. MuSE: accounting for tumor heterogeneity using a sample-specific error model improves sensitivity and specificity in mutation calling from sequencing data. *Genome Biol.* **17**, 178 (2016).
- Lai, Z. et al. VarDict: a novel and versatile variant caller for next-generation sequencing in cancer research. *Nucleic Acids Res.* **44**, e108–e108 (2016).
- Koboldt, D. C. et al. VarScan 2: somatic mutation and copy number alteration discovery in cancer by exome sequencing. *Genome Res.* **22**, 568–576 (2012).
- Kim, S. et al. Strelka2: fast and accurate calling of germline and somatic variants. *Nat. Methods* **15**, 591–594 (2018).
- Larson, D. E. et al. SomaticSniper: identification of somatic point mutations in whole genome sequencing data. *Bioinformatics* **28**, 311–317 (2011).
- Wang, Q. et al. Detecting somatic point mutations in cancer genome sequencing data: a comparison of mutation callers. *Genome Med.* **5**, 91 (2013).
- Alioti, T. S. et al. A comprehensive assessment of somatic mutation detection in cancer using whole-genome sequencing. *Nat. Commun.* **6**, 10001 (2015).
- Roberts, N. D. et al. A comparative analysis of algorithms for somatic SNV detection in cancer. *Bioinformatics* **29**, 2223–2230 (2013).
- Fang, L. T. et al. An ensemble approach to accurately detect somatic mutations using SomaticSeq. *Genome Biol.* **16**, 197 (2015).
- Poplin, R. et al. A universal SNP and small-indel variant caller using deep neural networks. *Nat. Biotechnol.* **36**, 983 (2018).
- Luo, R., Sedlazeck, F. J., Lam, T.-W. & Schatz, M. Clairvoyante: a multi-task convolutional deep neural network for variant calling in single molecule sequencing. <https://www.biorxiv.org/content/early/2018/04/28/310458> (2018).

13. Simple convolutional neural network for genomic variant calling with tensorflow. <https://towardsdatascience.com/simple-convolution-neural-network-for-genomic-variant-calling-with-tensorflow-c085dbc2026f> (2017).
14. Esteva, A. et al. Dermatologist-level classification of skin cancer with deep neural networks. *Nature* **542**, 115–118 (2017).
15. Torracinta, R. et al. Adaptive somatic mutations calls with deep learning and semi-simulated data. <https://www.biorxiv.org/content/early/2016/10/04/079087> (2016).
16. Running DeepVariant. <https://cloud.google.com/genomics/docs/tutorials/deepvariant> (2018).
17. He, K., Zhang, X., Ren, S. & Sun, J. Deep residual learning for image recognition. *Proc. IEEE conf. comput. vis. pattern recognit.* 770–778 (2016) http://openaccess.thecvf.com/content_cvpr_2016/html/He_Deep_Residual_Learning_CVPR_2016_paper.html.
18. Eberle, M. A. et al. A reference data set of 5.4 million phased human variants validated by genetic inheritance from sequencing a three-generation 17-member pedigree. *Genome Res.* **27**, 157–164 (2017).
19. Ewing, A. D. et al. Combining tumor genome simulation with crowdsourcing to benchmark somatic single-nucleotide-variant detection. *Nat. Methods* **12**, 623 (2015).
20. Zook, J. M. et al. Extensive sequencing of seven human genomes to characterize benchmark reference materials. *Sci. Data* **3**, 160025 (2016).
21. Xu, C. A review of somatic single nucleotide variant calling algorithms for next-generation sequencing data. *Comput. Struct. Biotechnol. J.* **16**, 15–24 (2018).
22. Zook, J. M. et al. Integrating human sequence data sets provides a resource of benchmark SNP and indel genotype calls. *Nat. Biotechnol.* **32**, 246 (2014).
23. Puente, X. S. et al. Whole-genome sequencing identifies recurrent mutations in chronic lymphocytic leukaemia. *Nature* **475**, 101 (2011).
24. Morse, H. G. & Moore, G. E. Cytogenetic homogeneity in eight independent sites in a case of malignant melanoma. *Cancer Genet. Cytogenet.* **69**, 108–112 (1993).
25. Pleasance, E. D. et al. A comprehensive catalogue of somatic mutations from a human cancer genome. *Nature* **463**, 191 (2010).
26. Network, C. G. A. et al. Comprehensive molecular characterization of human colon and rectal cancer. *Nature* **487**, 330 (2012).
27. Grasso, C. S. et al. Genetic mechanisms of immune evasion in colorectal cancer. *Cancer Discov.* **8**, 730–749 (2018).
28. Li, H. Aligning sequence reads, clone sequences and assembly contigs with BWA-MEM. <https://arxiv.org/abs/1303.3997> (2013).
29. Van der Auwera, G. A. et al. From FastQ data to high-confidence variant calls: the genome analysis toolkit best practices pipeline. *Curr. Protoc. Bioinforma.* **43**, 10–11 (2013).
30. Grossman, R. L. et al. Toward a shared vision for cancer genomic data. *N. Engl. J. Med.* **375**, 1109–1112 (2016).
31. Forbes, S. A. et al. COSMIC: somatic cancer genetics at high-resolution. *Nucleic Acids Res.* **45**, D777–D783 (2016).
32. Chaisson, M. J. et al. Resolving the complexity of the human genome using single-molecule sequencing. *Nature* **517**, 608 (2015).
33. Mu, J. C. et al. VarSim: a high-fidelity simulation and validation framework for high-throughput genome sequencing with cancer applications. *Bioinformatics* **31**, 1469–1471 (2014).
34. Lau, B. et al. LongISLND: in silico sequencing of lengthy and noisy datatypes. *Bioinformatics* **32**, 3829–3832 (2016).
35. Sedlazeck, F. J. et al. Accurate detection of complex structural variations using single-molecule sequencing. *Nat. Methods* **15**, 461–468 (2018).
36. Sherry, S. T. et al. dbSNP: the NCBI database of genetic variation. *Nucleic Acids Res.* **29**, 308–311 (2001).

Acknowledgements

We thank Li Tai Fang, Yunfei Guo, Lijing Yao, Betty Ulitsky, Jian Li, Anoop Grewal, Daniel Kogan, and Michael Braverman from Roche Sequencing Solutions for fruitful discussions and proofreading the manuscript.

Author contributions

S.M.E.S., M.M., and H.Y.K.L. conceived the study, reviewed the analysis, and wrote the manuscript. H.Y.K.L. designed the study. M.M. and H.Y.K.L. supervised the study. S.M.E.S. designed and developed the NeuSomatic algorithm and its software. R.L. and S.M.E.S. developed the algorithm for scanning the genome, preparing the augmented alignments, and extracting alignment features. S.M.E.S. performed the overall analysis. B.L., S.M.E.S., and M.M. performed the long-read analysis. S.M.E.S., M.M., and K.P. performed the cloud computation. All authors reviewed and edited the manuscript.

Additional information

Supplementary Information accompanies this paper at <https://doi.org/10.1038/s41467-019-09027-x>.

Competing interests: S.M.E.S., M.M., and H.Y.K.L. have filed a patent application on this invention. The remaining authors declare no competing interests.

Reprints and permission information is available online at <http://npg.nature.com/reprintsandpermissions/>

Journal peer review information: *Nature Communications* thanks Daniel Chow and the other anonymous reviewer(s) for their contribution to the peer review of this work.

Publisher's note: Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this license, visit <http://creativecommons.org/licenses/by/4.0/>.

© The Author(s) 2019