BMC Bioinformatics

RESEARCH

Open Access

# Optimizing sequence data analysis using convolution neural network for the prediction of CNV bait positions

Zoltán Maróti[1*†], Peter Juma Ochieng[2,3,4*†], József Dombi[3,4], Miklós Krész[5,6,7] and Tibor Kalmár[1*]

†Zoltán Maróti and Peter Juma Ochieng these two author contributed equally to this work.

*Correspondence: maroti.zoltan@med.u-szeged.hu; juma@inf.u-szeged.hu; kalmar.tibor@med.u-szeged.hu

[1] Albert Szent-Györgyi Health Centre, University of Szeged, Korányi fasor 14-15, Szeged H-6725, Csongrád-Csanád, Hungary
[2] Interdisciplinary Research Development and Innovation Center of Excellence, Institute of Informatics, University of Szeged, Árpád tér 2, Szeged H-6720, Csongrád-Csanád, Hungary
[3] HUN-REN SZTE Research Group on Artificial Intelligence, University of Szeged, Árpád tér 2, Szeged H-6720, Csongrád-Csanád, Hungary
[4] Institute of Informatics, University of Szeged, Árpád tér 2, Szeged H-6720, Csongrád-Csanád, Hungary
[5] InnoRenew CoE, Livade 6a, Izola SI-6310, Slovenia
[6] Andrej Marušic Institute, University of Primorska, Muzejski trg 2, Koper 6000, Slovenia
[7] Department of Applied Informatics, University of Szeged, Boldogasszony sgt. 6, Szeged H-6725, Hungary

## Abstract

***Background*:** Accurate prediction of copy number variations (CNVs) from targeted capture next-generation sequencing (NGS) data relies on effective normalization of read coverage profiles. The normalization process is particularly challenging due to hidden systemic biases such as GC bias, which can significantly affect the sensitivity and specificity of CNV detection. In many cases, the kit manifests provide only the genome coordinates of the targeted regions, and the exact bait design of the oligo capture baits is not available. Although the on-target regions significantly overlap with the bait design, a lack of adequate information allows less accurate normalization of the coverage data. In this study, we propose a novel approach that utilizes a 1D convolution neural network (CNN) model to predict the positions of capture baits in complex whole-exome sequencing (WES) kits. By accurately identifying the exact positions of bait coordinates, our model enables precise normalization of GC bias across target regions, thereby allowing better CNV data normalization.

***Results*:** We evaluated the optimal hyperparameters, model architecture, and complexity to predict the likely positions of the oligo capture baits. Our analysis shows that the CNN models outperform the Dense NN for bait predictions. Batch normalization is the most important parameter for the stable training of CNN models. Our results indicate that the spatiality of the data plays an important role in the prediction performance. We have shown that combined input data, including experimental coverage, on-target information, and sequence data, are critical for bait prediction. Furthermore, comparison with the on-target information indicated that the CNN models performed better in predicting bait positions that exhibited a high degree of overlap (>90%) with the true bait positions.

***Results*:** This study highlights the potential of utilizing CNN-based approaches to optimize coverage data analysis and improve copy number data normalization. Subsequent CNV detection based on these predicted coordinates facilitates more accurate measurement of coverage profiles and better normalization for GC bias. As a result, this approach could reduce systemic bias and improve the sensitivity and specificity of CNV detection in genomic studies.

**Keywords:** Targeted capture, Oligo capture baits, Copy number variation, Machine learning

## Background

Copy number variations (CNVs) are structural genomic alterations that involve the deletion and/or duplication of DNA segments. CNVs have been implicated in various human diseases, including cancer and rare monogenic diseases. Accurate identification and characterization of CNVs are crucial for understanding the genetic basis of these diseases and developing effective diagnostic and therapeutic strategies[1]. Advances in high-throughput sequencing technologies, such as next-generation sequencing (NGS), have revolutionized the field of genomics and provided a wealth of data for studying CNVs. NGS platforms generate massive amounts of sequencing data, which contain valuable information about genomic variations. The most widely used Illumina short-read sequencing technology is suitable for single nucleotide variant (SNV) and small insertion/deletion detection. However, the analysis of sequence data for CNV detection and characterization presents significant challenges due to the complexity and scale of the data [2, 3].

Although whole-genome sequencing (WGS) is becoming more common, a large portion of genetic diagnoses are based on whole-exome sequencing (WES) [4]. WES is a targeted enrichment method in which only the coding regions of genes are sequenced. The majority of WES kits are hybridization-based, where hundreds of thousands of oligo baits are used to capture and enrich the targeted regions of the genome. These oligo capture baits are short (typically 80–120 base pairs) DNA or RNA fragments with nucleotide sequences specific to the targeted genomic region [5].

During the hybridization process, the DNA fragments of the sample are annealed with single-stranded oligo baits to form double-stranded DNA. Since the nucleotides forming the double-stranded chain bind with weak secondary hydrogen bonds, hybridization is a dynamic thermodynamic equilibrium under given conditions (temperature, ionic concentration). Next, the unbound and partially matching DNA fragments are removed, and the targeted DNA fragments complementing the oligo baits are captured and enriched [6]. The captured reads aligned to the reference genome show a distinct coverage pattern that reflects the bait design and the hybridization (and other wet-lab) conditions of the individual samples (Fig. 1).

CNV analysis relies on statistical models and algorithms that are designed to detect variations in read coverage across the genome [7–10]. In the case of WES-targeted enrichment, the hybridization of DNA fragments to thousands of oligo baits with unique sequence compositions introduces imbalances in the coverage data because the conditions of hybridization influence the capture efficiency of different baits [5]. Thus, minor changes in the temperature and buffer concentrations of reagents used in the library preparation protocol will lead to sample- and batch-specific hybridization bias between cohorts.

This bias is often referred to as GC bias, where G and C represent the two nucleotides (G = guanine, C = cytosine) that bind with three hydrogen bonds, leading to stronger (thermodynamically more stable) bonds between sequences with high GC content compared with AT-rich (A = adenine, T= thymine) sequences that bind with two hydrogen bonds. Consequently, changes in chemical or physical conditions that shift the thermodynamic balance during the hybridization process will also imbalance the hybridization efficiency and the proportion of captured DNA based on the GC

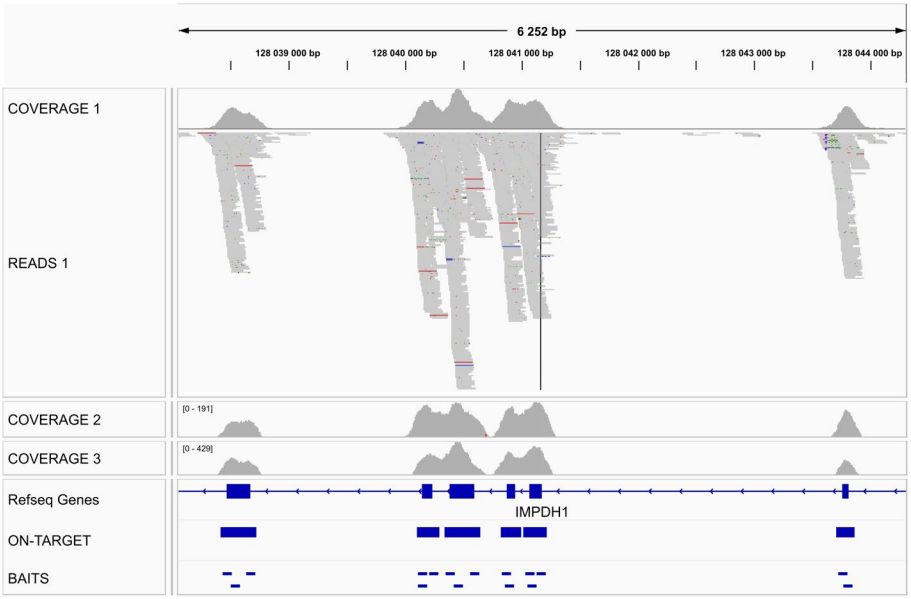


**Fig. 1** Visualization of NGS sequences and coverage profiles via the Integrative Genome Viewer. The tracks COVERAGE 1–3 display the coverage profiles of three different samples. The READ 1 track shows the actual NGS sequences aligned to the reference genome of sample 1, which is the basis of coverage profile 1. The RefSeq Genes track contains the exonic regions of a randomly selected gene (IMPDH1), while the ON-TARGET track shows the targeted region coordinates provided in the kit manifest. The BAITS track shows the actual bait design of this particular WES kit

content, leading to GC bias [11]. Due to the sheer number of baits in a WES capture kit, the bait coordinates are designed by algorithms and sensible rules based on the GC content and the sequence context within and around each target region. Since the aim is to achieve uniform, reproducible target capture, the applied rules try to minimize the factors (including GC bias) leading to too low or too high capture efficiency around the target regions. Accordingly, it is also advantageous to design baits with similar GC ratios, as hybridization of all baits is performed in a single reaction. Similarly, placing more baits around poorly enriched areas and avoiding repetitive areas based on the reference genome context can lead to better bait design that produces more uniform sequence coverage of the target regions [12, 13].

Since many of these decisions are not vendor but rather sequence context and hybridization specific, we hypothesize that the knowledge of target regions (which also defines the targeted sequence context) and the experimental coverage data in a cohort of samples can be used to pinpoint the most likely bait positions in any WES design. We propose a novel approach that harnesses the power of convolutional neural networks (CNNs) to optimize sequence data analysis for CNV bait position prediction. CNNs are a class of deep learning models that have demonstrated remarkable success in various computer vision and natural language processing tasks [14]. By leveraging the hierarchical and local feature extraction capabilities of CNNs, our objective is to capture the intricate patterns and relationships within the coverage profile and sequence context data that are indicative of bait positions and provide a robust and reliable tool to predict the most likely bait design. If the predicted bait positions

largely overlap with the exact bait coordinates, better CNV normalization could be achieved even when the bait design is not publicly available.

To test the feasibility of our approach, we used publicly available WES sequence data based on a WES kit where the exact bait design (the ground truth for training) was provided. As this dataset is publicly available, our methodology can be easily reproduced and tested without the privacy concerns of actual clinical WES data. In our manuscript, we provide a comprehensive description of the tested network architectures, the data preprocessing techniques, and the training procedures. We also compared the different machine learning models and the effects of various optimizations to determine the ML models with the best accuracy and to balance the training speed and CPU requirements. In addition,

Furthermore, we discuss the potential applications and implications of our method in the field of genomics and personalized medicine. CNV mutations contribute approximatively 5–10% of the mutation spectrum while large majority of CNV mutations are not routinely detected from WES NGS sequences. The main problem is the inadequate GC bias normalization of read counts at the target regions due to the lack of information of the actual oligo bait design. The use of approximate GC normalization based on the on-target coordinates leads to lower sensitivity and specificity, especially for detecting smaller CNVs. Through the application of deep learning techniques, our approach represents a significant step toward improving the GC bias normalization by accurately predicting the bait positions for the majority of WES kits that do not include the bait coordinates in their design manifests. As WES is still the most widely used method in clinical genetic diagnosis, our results could improve the sensitivity and specificity of CNV detection with such kits. This in turn could facilitate diagnostic and research on copy number variations and contribute to advancing our understanding of the genetic basis of complex disorders.

## Methodology

In this section, first we present a systematic workflow for the proposed model designed to predict the positions of CNV baits from whole exome sequencing (WES) data. The model follows a four-step process (Fig. 2). The first step involves the acquisition of the dataset, where the on target data, WES sequence data, human reference sequence data, and Truth bait coordinates data are gathered. The second step focuses on data preprocessing to prepare the data for subsequent analysis. In the third step, the data are partitioned into training, evaluation, and validation datasets for different models. Finally, the fourth step is the training and evaluation of the model to predict the position of the CNV bait.

### Used datasets

In accordance with national regulations in Hungary ("a humángenetikai adatok védelméről, a humángenetikai vizsgálatok és kutatások, valamint a biobankok működésének szabályairól"/Act XXI of 2008 on the protection of human genetic data, the rules of human genetic testing and research, and the operation of biobanks), genetic data that could enable personal identification such as whole exome sequencing (WES) data can only be uploaded to restricted, request-only closed repositories, even with

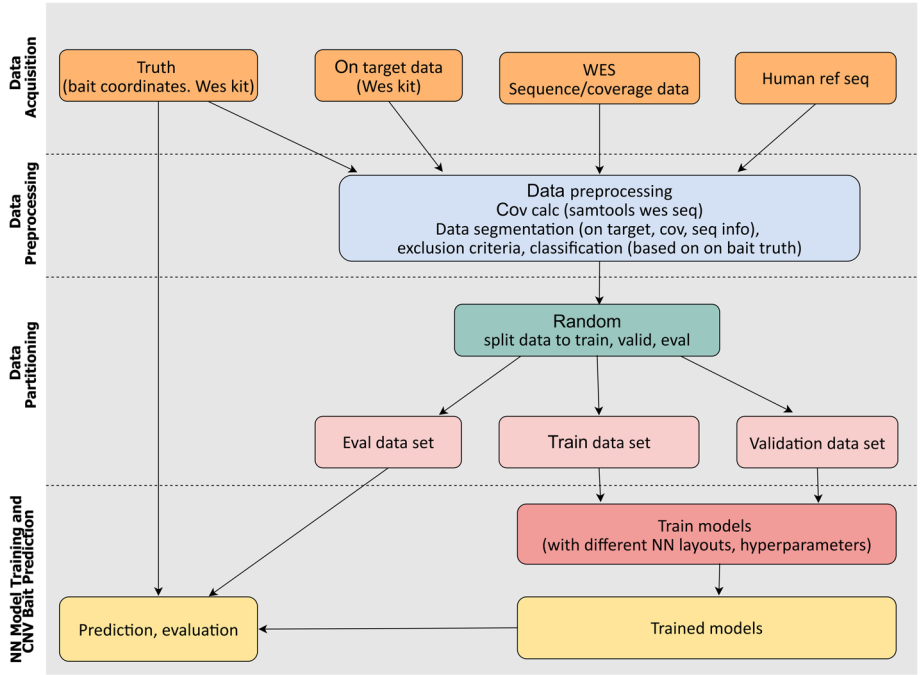


**Fig. 2** Overview of NN workflow for prediction of CNV bait positions

patient consent. Consequently, our analysis was conducted using a dataset that is publicly available after registration in a closed public repository hosted by BaseSpace, a platform provided by Illumina. Specifically, we utilized publicly available demo sequence data generated by the Illumina Nextera WES kit, accessible on BaseSpace (https://basespace.illumina.com/projects/206115910/about). This WES kit includes the bait design (the exact genomic coordinates of each bait) that we used as the ground truth for training the machine learning models. This dataset contains the sequence data of the NA12878 sample from the 1 KG phase III sample collection in 96 replicates that were sequenced on the NovaSeq Illumina sequencing platform using a 2x150 base pair S2 flow cell. We downloaded the raw FASTQ files and aligned the reads to the GRCh37 human reference genome using the BWA-MEM algorithm for paired-end alignment [15]. We used the PICARD tool MarkDuplicate to mark PCR and optical duplicate reads.

### Data preprocessing

We downloaded the target region and bait coordinate data files from the Illumina Nextera WES kit manifest files from the Illumina web site (https://support.illumina.com/downloads/nextera-dna-exome-product-files.html). We extended the target region coordinates with 2000 base pairs with bedtools [16] using the "slop -b 2000" option and merged the resulting genome regions with "bedtools merge" to create non-overlapping continuous genomic regions that includes all of the target regions with a sufficiently large genomic context. All together, this resulted in 99188 chunks of non-overlapping genome regions [https://doi.org/10.5281/zenodo.11102581: GRCh37_coordinates]. Using the genome coordinates of these genomic regions we used "samtools depth" with the "-aa -b regions.bed" options to generate the coverage

data for each genome position of the regions of interest [17]. The public data set was already normalized (60 M PE reads per sample corresponding to $\sim 72\times$ mean target coverage). Accordingly we did not have to normalize the observed absolute genome coverages across the samples.

From the used GRCh37 reference data we looked up the nucleotide sequence for each position of the regions of interest and coded them numerically (1 - A, 2 - T, 3 - C, 4 - G, 0 - N). For the same position, we also coded the target region information of the kit manifest as 1 or 0 denoting whether a given genome position is targeted (1) or not (0) in the Illumina Nextera WES kit manifest. Lastly, we also created our truth for the same genome position, coded as 1 or 0 denoting whether the given genome position is covered by a bait (1) or not (0).

Although some machine learning models can accommodate variable-length input sequences, our CNN model architecture requires that all training examples have a uniform data shape. However, since our dataset consists of 99,188 genomic chunks that vary in length, we implemented the following strategy to segment these genomic regions into equal-sized genomic windows (fixed-length data segments):

- For each replicate sample and each genomic chunk, we used a random offset "random (window size)" to avoid bias due to placing the experimental high-read coverage regions or bait features at specific positions (i.e., centers) of the data windows.
- Using this offset, we split the consecutive genomic positions of the genomic chunk into window size segments; if the last segment was smaller than the window size, we excluded this segment.
- Since we extended all genomic chunks with 2000 base pairs on both sides, a large portion of the genomic positions in the resulting genomic chunks were not target regions, had no experimental read coverage, or did not intersect the actual bait coordinates and could not be used for training; consequently, we excluded all resulting windows (data segments) that

  - had a length less than the bait length (80 base pairs) of overlap with bait positions
  - or had less than 40 mean read coverage at the covered bait positions

We hypothesized that the available genomic context around the peak data (sequence information, additional target regions, and additional peaks in the coverage near the analyzed region) can influence the accuracy of the prediction. Furthermore, larger data windows with more data and memory requirements are also expected to increase the training time of the machine learning model. Thus, to investigate the benefits and trade-offs, we used 500- and 1000-base pair (bp) length window sizes for segmenting our data.

In theory, any particular data window can include between 80 and the window size number of bait positions (as we excluded all data segments that had less than 80 bait position overlaps, i.e., less than one single bait overlapping the segment). However, in our experimental data, the number of cases in which we had exactly one bait or

more baits in the training window was not random, as exonic regions in the genome (the target regions) have cluster like distribution, and the lengths of the target regions are widely different. Hence, many regions are only covered by a single bait, but a significant portion of target regions are targeted by more overlapping or sparsely distributed baits. Consequently, the numbers of training examples for these different cases are not equal. Furthermore, the difficulty of predicting a single nonoverlapping bait or the best combination of sparse, potentially overlapping baits is radically different. To avoid overtraining for the most frequent and easiest bait per train window example situation, we classified our training examples by the following criteria:

- exactly 80 bait positions (1 bait)
- 81–160 bait positions (1–2 baits)
- 161–240 bait positions (2–3 baits)
- 41–320 bait positions (3–4 baits)
- 321-or more bait positions (5 or more baits)

Using the above classification criteria, we calculated the count of each class of the training window from the 1 M training examples and defined the reverse weight of each class of the example by the following equation

$$\hat{w}(c_i) = \frac{\min(C)}{c_i}, \tag{1}$$

where $\hat{w}$ is the reverse weight of the bait count in the $i^{th}$ class, min $(C)$ is the minimum of the example bait count of the five different classes and $i^{th}$ is the example count of the $i^{th}$ class.

Since training is based on calculating the global loss and accuracy of the whole training dataset this would favor the model to generalize on the most frequent and easiest training example while performing worse on the least frequent and much harder cases when a combination of sparse potentially overlapping baits are designed to capture a larger difficult target region. With the applied reverse weighting based on the frequency of the different complexity cases, we tried to counterbalance the model training to achieve better predictions for the hard cases without significantly sacrificing the global performance.

#### Data partitioning for model training and evaluation

Splitting the data into equal-sized windows and applying the described exclusion criteria resulted in ≈ 18.1 M data segments (examples) for the 500 base pairs and ≈15.9M data segments for the 1000 base pair length window size. For training, validation, and evaluation of the proposed CNN models, we split the 500-bp and 1000-bp example data sets into training, validation, and evaluation subsets [https://doi.org/10.5281/zenodo.11102581: train_data].

To avoid bias due to learning specific sequence contexts during training, we ensured that the nucleotide sequences in the target regions did not contain an excessive number of homologous sequences (see **Additional File 6:** Nucleotide Sequence Homology Test). Subsequently, we partitioned the data so that the training, validation, and evaluation

datasets comprised data from different random genome chunks. This approach guaranteed that each dataset was derived from unique genomic regions of the human reference genome.

We included slightly more than 1 million examples for both the training and the validation data sets, while the evaluation data set included all remaining examples. We trained all NN models for 100 epochs [https://doi.org/10.5281/zenodo.11102581: saved_models]. During training, we limited the train and validation data to the same 1-1 M examples to make a runtime assessment of training and the CNN model data setups comparable. We calculated the mean loss and accuracy of the train/validation at the end of each epochs and saved the best model for each NN model [https://doi.org/10.5281/zenodo.11102581: saved_models]. After model training, the best mean loss models were evaluated for all tested NN models using the same 1 M random examples from the evaluation data set. Due to size constraints, we deposited all raw metrics files generated during the evaluation of NN models at a Zenodo deposit [https://doi.org/10.5281/zenodo.11102581: eval_output].

#### Proposed CNN model architecture

To predict the positions of the CNV bait, our proposed 1D CNN model uses the targeted genomic region, sequence information, and experimental coverage data as input. The model uses a series of convolutional layers, batch normalization, and reverse weighting to handle the varying number of training examples of different difficulty levels, based on the overlap potential of baits within each data segment. Additionally, we incorporate dropout regularization, max pooling, and dense layers to train the model and make predictions across a specified window length. While the proposed CNN architecture includes flattening and pooling layers these can be optionally excluded without significantly impacting the model's predictive performance (Additional file 6: The proposed CNN model architecture for prediction of bait position.). In the initial convolutional layer, our model applies 50 filters, each of size 60, with a stride of 40. This layer uses ReLU activation and causal padding, which preserves the temporal sequence order. Following each convolutional layer, batch normalization is applied to standardize the activations, thereby improving training stability and speed. Additional convolutional layers follow, each configured with unique filter sizes and counts. Each layer is paired with batch normalization and dropout regularization. These layers include: Conv1D (80, 20): applies 80 filters of size 20, with causal padding and ReLU activation. Conv1D (30, 10): applies 30 filters of size 10, with causal padding and ReLU activation. Conv1D (40, 5): applies 40 filters of size 5, with causal padding and ReLU activation. Conv1D (4, 2): applies 4 filters of size 2, with causal padding and ReLU activation. Following the convolutional layers, we add a dense layer with the same number of neurons as the window size, activated by a sigmoid function. The output layer of the CNN model provides sigmoid-activated probabilities, which represent the likelihood of bait presence within each position. The bait prediction score is calculated as the product of the sigmoid activation value and the probability, enabling biological significance assessment of predicted bait locations within a genomic window. The final output of the dense layer is reshaped to fit the desired format for bait position predictions in the input sequence data. To classify the presence or absence of bait within a genomic window, we use cross-entropy as the

loss function, which helps balance sensitivity and specificity, thus reducing bias during training [18]. We used causal padding, which does not alter the one-to-one correspondence between the input and output layouts. Given that accurate bait prediction likely requires the surrounding genomic and coverage context, we also assessed prediction performance within data segments based on relative position. This approach allows us to account for the impact of flanking regions on prediction accuracy.

## Results

### Comparing the model training performance

Our first experiment compared the training performance of the proposed CNN model to that of the dense layer models in terms of the training and validation accuracy and evaluated the effect of the relevant hyperparameter optimizations. In our case, two models consisted of dense layer models, and the other eight models were 1D CNNs with different parameterizations, as shown in Table 1.

As described in the data partitioning section of the Methods, we used 500 and 1000 base pair lengths of data segments for model training. We trained all the models for 100 epochs with 1 M training and 1 M validation examples. The basic metrics (accuracy, loss) calculated for the training and validation data at the end of each epoch during training can provide important insight for the models. Large differences between the loss/accuracy of the training and validation datasets can indicate that training on the training dataset has less generalization power on the overall data for the model. Thus, we also calculated the mean loss and accuracy metrics of the training and validation data. We summarized the changes in the most important metrics (loss and accuracy) during training (Additional file 2; Figs. 3 and 4). Large fluctuations in the loss, accuracy and AUC metrics during training indicate poor training convergence of a model for a specific task. In the case of model three, our results indicated that in this task, batch normalization is essential for CONV1D models (Additional file 1: Training history of model 3; 1D CNN without batch normalization). According to our results, dense models with or without batch normalization (models 2 and 4) had very similar performances; however, the accuracy and loss were worse than those of any of the CONV1D models (models 1 and 5–10) with batch normalization. To assess the complexity of the models, we

**Table 1** Summary of the tested 10 models, including the neural network architecture, input data, and the applied optimizations

| Model | Model Architecture | Hyper-parameter optimizations | Input data |
|---|---|---|---|
| 1 | Conv1D | batch normalization, causal padding, reverse weight | COV, SEQ, ONTARGET |
| 2 | Dense | batch normalization, reverse weight | COV, SEQ, ONTARGET |
| 3 | Conv1D | causal padding, reverse weight | COV, SEQ, ONTARGET |
| 4 | Dense | reverse weight | COV, SEQ, ONTARGET |
| 5 | Conv1D | batch normalization, causal padding | COV, SEQ, ONTARGET |
| 6 | Conv1D | batch normalization, reverse weight, valid padding | COV, SEQ, ONTARGET |
| 7 | Conv1D | batch normalization, reverse weight, same padding | COV, SEQ, ONTARGET |
| 8 | Conv1D | batch normalization, causal padding, reverse weight | COV, SEQ |
| 9 | Conv1D | batch normalization, causal padding, reverse weight | COV, ONTARGET |
| 10 | Conv1D | batch normalization, causal padding, reverse weight | SEQ, ONTARGET |

**Table 2** The best accuracy, loss and means of training and validation dataset of the analyzed models in 100 epochs and the mean time required for training one epoch

| MODEL | WINDOW | train loss | val loss | mean loss | train acc | val acc | mean acc | mean time (seconds) |
|---|---|---|---|---|---|---|---|---|
| 1 | 500 | 0,0182 | 0,0316 | 0,0275 | 0,9613 | 0,9272 | 0,9415 | 1861,8 |
| 2 | 500 | 0,0561 | 0,0558 | 0,0559 | 0,8565 | 0,8559 | 0,8562 | 8508,1 |
| 3 | 500 | 0,0447 | 0,0410 | 0,0445 | 0,8916 | 0,8996 | 0,8921 | 1451,6 |
| 4 | 500 | 0,0563 | 0,0558 | 0,0561 | 0,8557 | 0,8559 | 0,8558 | 4877,8 |
| 5 | 500 | 0,0569 | 0,0927 | 0,0805 | 0,9764 | 0,9601 | 0,9667 | 1845,3 |
| 6 | 500 | 0,0212 | 0,0322 | 0,0292 | 0,9544 | 0,9245 | 0,9359 | 1536,4 |
| 7 | 500 | 0,0196 | 0,0313 | 0,0280 | 0,9581 | 0,9276 | 0,9396 | 1821,1 |
| 8 | 500 | 0,0372 | 0,0497 | 0,0468 | 0,9189 | 0,8866 | 0,8975 | 1811,3 |
| 9 | 500 | 0,0301 | 0,0332 | 0,0317 | 0,9304 | 0,9221 | 0,9260 | 1824,1 |
| 10 | 500 | 0,0174 | 0,0482 | 0,0402 | 0,9628 | 0,8760 | 0,9143 | 1836,2 |
| 1 | 1000 | 0,0161 | 0,0302 | 0,0255 | 0,9802 | 0,9611 | 0,9691 | 3813,1 |
| 2 | 1000 | 0,0615 | 0,0628 | 0,0622 | 0,9056 | 0,9034 | 0,9045 | 19675,8 |
| 3 | 1000 | 0,0364 | 0,0363 | 0,0363 | 0,9508 | 0,9508 | 0,9508 | 3059,8 |
| 4 | 1000 | 0,0618 | 0,0628 | 0,0623 | 0,9051 | 0,9034 | 0,9043 | 9757,1 |
| 5 | 1000 | 0,0306 | 0,0570 | 0,0477 | 0,9875 | 0,9759 | 0,9812 | 3798,9 |
| 6 | 1000 | 0,0172 | 0,0312 | 0,0265 | 0,9788 | 0,9585 | 0,9681 | 3361,9 |
| 7 | 1000 | 0,0160 | 0,0303 | 0,0253 | 0,9803 | 0,9603 | 0,9696 | 3727,4 |
| 8 | 1000 | 0,0372 | 0,0523 | 0,0476 | 0,9532 | 0,9346 | 0,9418 | 3793,2 |
| 9 | 1000 | 0,0278 | 0,0321 | 0,0299 | 0,9633 | 0,9572 | 0,9602 | 3770,2 |
| 10 | 1000 | 0,0174 | 0,0552 | 0,0432 | 0,9787 | 0,9183 | 0,9484 | 3839,7 |

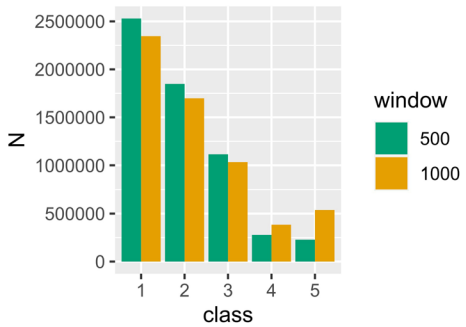


**Fig. 3** The distribution of different classes of 1 M train examples of the 500 and 1000 base pairs data sets. The classes (1–5) represent the bait count in the data segment (exactly 1, 1–2, 2–3, 3–4, 4 or more)

also calculated the mean CPU runtime of the epochs required for training. A summary of these parameters for the 10 tested models is provided in Table 2.

Our analysis shows that CONV1D models (models 1, 3, and 5–10) not only offer better accuracy and loss based on the 1 M training/validation data used but also require less CPU run time for training than do Dense models (models 2 and 4).

#### Evaluation and dissection of the models performance

During NN training, we calculated the mean loss (train and validation dataset) at the end of each epoch, and based on this value, we saved the "best loss" models. Using these models, we predicted 1 M random data examples from the evaluation subset of our data

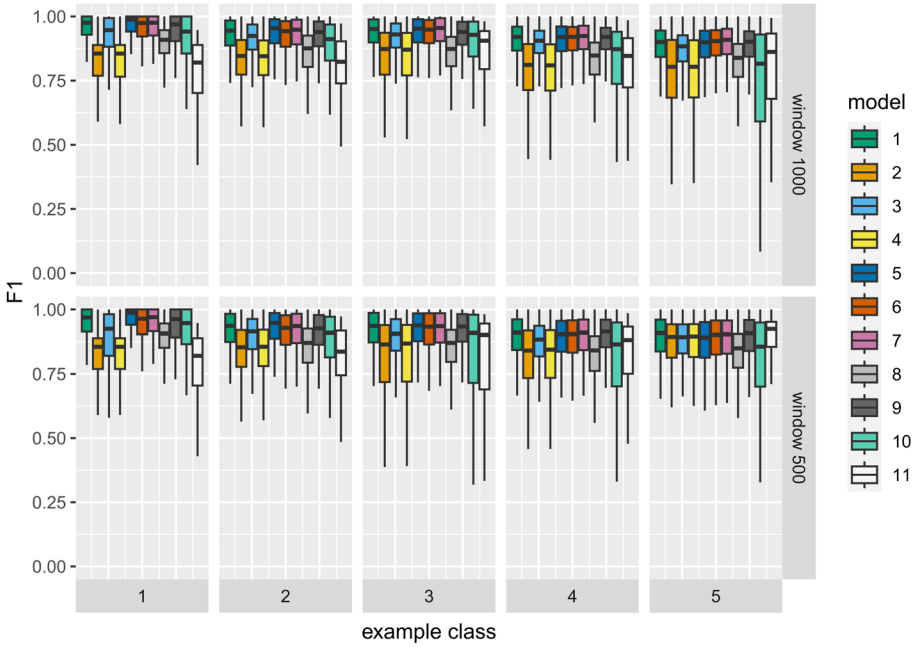


**Fig. 4** F1 score distribution of 1 M predictions of the different classes (based on the number of included baits) of train examples for window 1000 and 500 data segments

and compared the predictions with the ground truth (bait positions provided by the kit manufacturer) to calculate the most informative metrics. As described in the Methods section, we used an uneven random number of different complexity (based on the number of baits in the data segment) training examples (Fig. 3).

For each window size of input data we predicted the probability of a bait in a window size of spatial data points. Accordingly, we calculated the metrics (TP, TN, FP, FN, accuracy, precision, sensitivity, specificity, F1 score, MCC) for the 1 M individual window size of bait predictions ([https://doi.org/10.5281/zenodo.11102581: eval_output]). As stated earlier, in many cases only the on-target region information is provided in the capture kits; thus, CNV detection/normalization is based on these genome coordinates instead of the exact capture bait coordinates. Accordingly, as a baseline for this scenario (model 11), we used the positions of the on-target region as "prediction" and calculated the metrics for the same 1 M examples. We visualized the distribution of these metrics in each prediction difficulty class (single bait or more, potentially overlapping sparse baits in the data window) for the different models (Additional file 2: Distribution of the accuracy, precision, sensitivity, specificity, MCC metrics for the 5 train classes based on the 1 M individual predictions of the evaluation data set). The F1 and MCC scores differentiate the models best as they incorporate the true- and misclassification for both true positives and true negatives. In Fig. 4, we present the F1 score distribution of the 10 evaluated NN model predictions (and the baseline) in the different prediction difficulty classes.

Our results show that all NN models (models 1–10) resulted in better predictions than did the baseline model (model 11), especially for the most numerous easy (classes 1–3) bait examples. In the case of very dense bait coverage, model 11 seems to be artificially

better, especially for class 5 (with 4 or more 80 base pair length baits in the data segment) examples in the 500 base pair long data window, since in these cases, the majority of the data segment positions are also true bait positions. In the larger 1000-base pair data window case, there are fewer such extreme examples; consequently, the F1 score decreases for Model 11.

The dense models with or without batch normalization (models 2 and 4) perform much worse than the CNN models. Not surprisingly, the MCC metrics of the examined models were nearly identical (Additional file 2: Distribution of the accuracy, precision, sensitivity, specificity, and MCC metrics for the 5 training classes based on the 1 M individual predictions of the evaluation dataset). Without the batch normalization option, the Conv1D model 3 performance was unstable during training (Additional file 1: Training history of Model 3; 1D CNN without batch normalization), and the evaluation also revealed worse performance for this model (Fig. 4).

We noticed that the sensitivity of bait prediction was good even for Dense NNs; however, the specificity and precision differed greatly (Additional file 2: Distribution of the accuracy, precision, sensitivity, specificity, and MCC metrics for the 5 training classes based on the 1 M individual predictions of the evaluation dataset). Seemingly, the reverse weighting of uneven numbers of different difficulty examples helps only slightly in the case of the 500-base pair window examples for class 5 (4 or more baits in the data segment). In this window, the number of examples with very high bait numbers was considerably lower than that in single bait cases (Fig. 3). However, in the case of 1000 base pair data windows, the nonreverse weighted train option (Model 5) has a slight advantage, even for highly difficult cases. The padding options have minimal effects; causal and same padding (models 1 and 7) seem to be the best options, while the option of valid padding (model 6), where the data are not padded and only valid input points are used, is slightly worse. Our results also showed that all three input data sources (experimental coverage, on-target information and sequence data) of the input were required for the best prediction. Excluding the nucleotide sequence information in the data window (model 9) resulted in the smallest negative effect, most notably leading to slightly worse F1 and MCC predictions in the most numerous “easy” single bait situations. This likely means that when multiple near-equal solutions exist, the sequence context can help to pinpoint the most suitable position. The lack of on-target information (information on which region should be targeted by the baits) and especially the experimental coverage information (indicating which positions were covered by more reads) resulted in much worse performance and a much greater variance in the metric distribution, underlining the importance of this information. In the convolution models, the beginning and the end of the data segments partially lack flanking spatial information, as we have no or limited spatial information on one side. To evaluate the effect of the surrounding spatial context, we calculated and plotted the means of the metrics in the predicted data windows ((Fig. 5), Additional file 3: Position metric plots of the evaluated models for 1000 and 500 train sizes).

The accuracy and precision were less influenced; however, the specificity, sensitivity, F1, and MCC scores were much lower on both sides of the data segments, indicating the importance of the surrounding spatial context for bait position predictions. Consequently, we have better predictions at the inner portion of the data segment, while

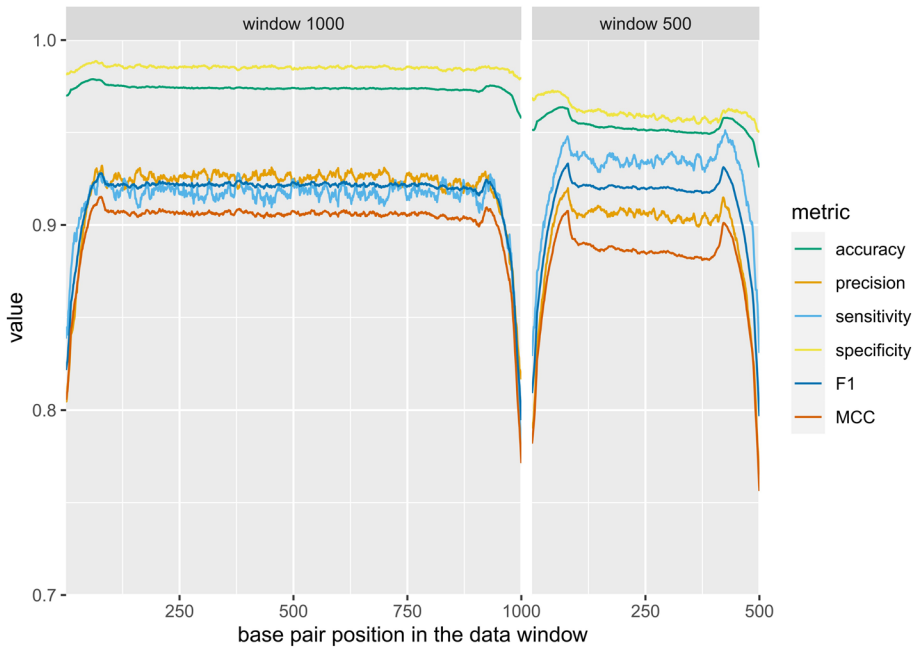


**Fig. 5** The mean of the evaluation metrics based on the position in the spatial data segment

we have worse predictions at the flank of the data segments. The same effect can be observed in the performance of all other models (Additional file 3: Position metric plots of the evaluated models for 1000 and 500 train sizes). Our results suggest that regardless of the data segment size, approximately one bait length (80 base pairs) of flanking context is required at both sides of our prediction position for the best results. Furthermore, our results also show that a larger genomic window with sufficient data context not only increases the relative ratio of good/bad prediction positions in a data segment but also improves the prediction power of the model at the inner positions of the data segment.

#### Visual analysis of the bait predictions

For all models, we predicted mixed (class 1–5) 1 M train examples. For each individual, prediction we calculated the TP, TN, FP, and FN compared to known bait positions (truth). Based on these values we calculated the individual accuracy, precision, sensitivity, specificity, F1, and MCC metrics in the case of all models and data windows. We ranked the 1 M predictions by the F1 score ([https://doi.org/10.5281/zenodo.11102581: eval_output]) and visualized a few best case ($99^{th}$ percentile), median case (median), and the worst case ($1^{st}$ percentile) examples for both window sizes (Figs. 6 and 7; Additional file 4: example plots of all models representative for the top 99, median and worst 1 percentile predictions based on the F1 score rankings of 1 M predictions).

Out of 1 M predictions model 1 resulted in 240704 perfect predictions (F1 = 1.0) in case of window size 500 and 225597 exact predictions in case of window size 1000 (Additional file 5: Metric evaluation of 1 M individual predictions for model 1 and 2). While even at the 50th percentile of examples (median) the F1 scores were $\approx 0.9473$ for Windows 500 and $\approx 0.9965$ for Windows 1000. In the worst case scenario, the F1 scores dropped to $\approx 0.6611$ and $\approx 0.6942$ at Windows 500 and 1000 respectively. However still

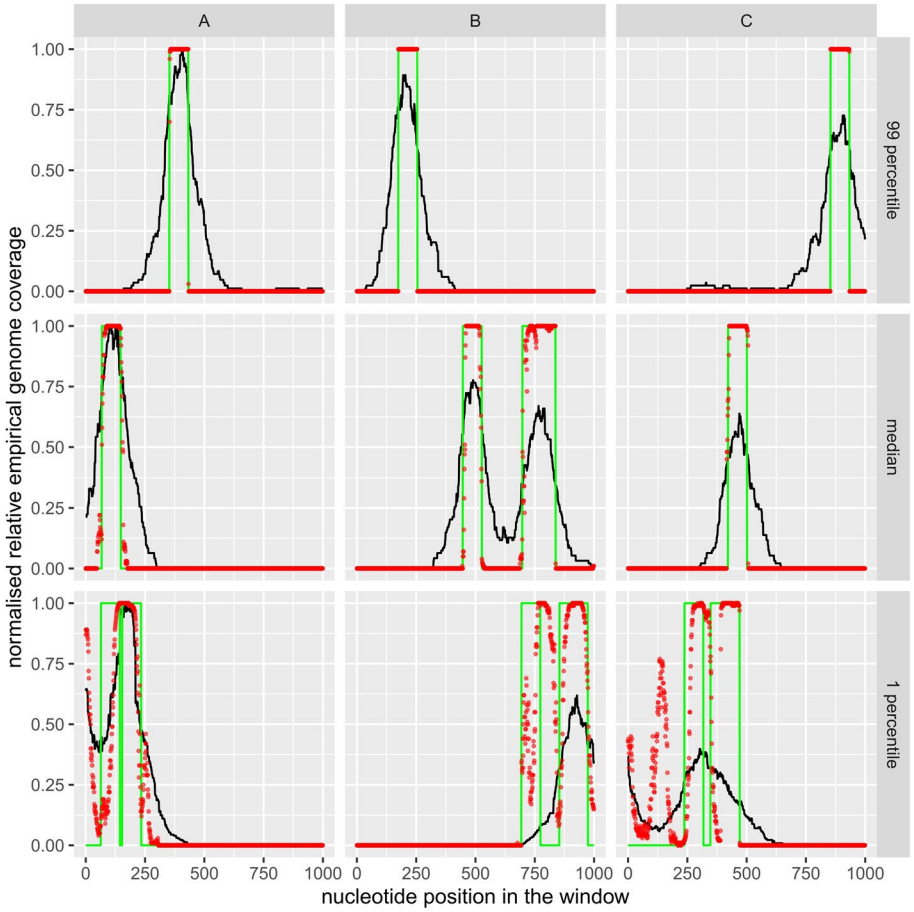


**Fig. 6** Visualization of predicted examples of model 1 based on the ranking of the F1 scores of 1 M individual predictions using the 1000 base pair data segment. Subpanels $99^{th}$, median, and $1^{st}$ percentile are 3 different (A, B, C) prediction examples with their appropriate F1 score rank. The black lines show the relative coverage in the data segment, the red dots are the probability of bait positions, the green line represents the bait positions (value 1 for bait position, value 0 for no bait position, the lines are used to better visualize the start and end of bait regions in the data segment)

in the worst-case scenarios, the predicted bait positions are mainly overlapping and/or flanking the true bait positions (Figs. 6, 7, bottom panes). In the case of the Dense models only 421 (window 500) and 592 (window 1000) were predicted exactly (F1 = 1.0) score and overall they also show much worse predictions (Additional file 5: Metric evaluation of 1 M individual predictions for model 1 and 2).

## Discussion

In targeted capture kits the oligo baits are used to capture and enrich the target DNA for NGS sequencing. The resulting sparse coverage data can be used to identify CNV variants, however due to random variations, batch effect of hidden systemic biases reflecting the wet-lab conditions of capture it is still an unsolved problem [7–9, 19–21]. One of the major systemic effects is the GC bias that influences the hybridization efficiency of baits due to the differences in their GC composition and the hidden wet-lab factors

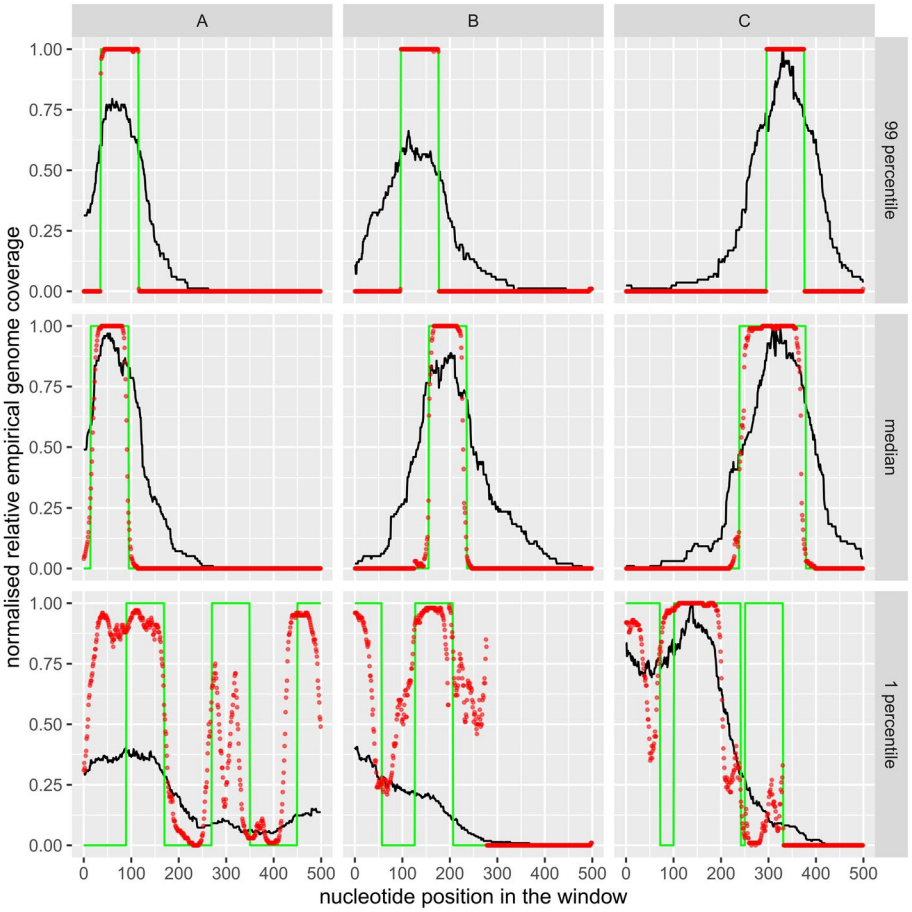


**Fig. 7** Visualization of predicted examples of model 1 based on the ranking of the F1 scores of 1 M individual predictions with the 500 base pair data segment. Subpanels $99^{th}$, median, and $1^{st}$ percentile are 3 different (A, B, C) prediction examples with their appropriate F1 score rank. The black lines show the relative coverage in the data segment, the red dots are the probability of bait positions, the green line represents the bait positions (value 1 for bait position, value 0 for no bait position, the lines are used to better visualize the start and end of bait regions in the data segment)

(temperature, buffer concentrations, and pipetting error) that may vary between individual and batch of sequenced samples.

Although the recommended wet-lab protocol aims to standardize the conditions throughout the often very long and complex library preparation steps, even minor changes can alter the hybridization efficiency of different oligo baits. For example, placing samples in the middle or side of the PCR machine could lead to temperature differences during hybridization while pipetting errors in the protocol can lead to minimal changes in buffer concentrations that could also lead to hidden differences between the samples. As these conditions influence the capture efficiency of baits based on their GC content, the resulting sample-specific differences lead to systemic bias in the read distribution between batches of samples and also between individual samples in WES data [22].

Accordingly, to allow proper CNV detection from WES data the underlying GC bias has to be mitigated. When the exact genomic coordinates of the oligo baits are known

the GC content of the bait can be determined. Using this information the systemic bias caused by the GC content of the baits can be reversed leading to better normalization of WES data. In theory, GC bias can be statistically normalized, in case we know the GC ratio of the capture oligo baits, however in many cases only the on-target information is provided in the capture kit manifests. To lower the overall sequencing cost, all WES vendors employ sophisticated algorithms to come up with a bait design that results in uniform capture coverage to eliminate the need of over-sequencing. To protect their intellectual property, many vendors do not share the bait design in the kit manifest and only the intended targeted region coordinates are included. Due to the large overlap of the targeted regions and the actual bait coordinates, in most cases the target regions can be used to normalize for GC bias. However, in some challenging genomic regions (with repetitive DNA elements, extreme GC ratios of the reference sequence, or conserved motif regions) the bait design needs to avoid certain genomic contexts. Thus, for such regions the GC content estimation based on the targeted region could be wrong as the overlap is smaller with the actual bait design in these regions. Consequently, knowledge of exact (or most probable) bait coordinates compared to using the target region coordinates could improve GC bias normalization even for these challenging regions. This often missing information is key to improving the normalization of GC bias, one of the largest hidden factors of normalization in hybridization-based target enrichment methods.

There are approximately 200K target regions in a typical WES kit that consist primarily of the coding regions of the exons of genes. The median exon size of the human genome is 120 base pairs, and approximately 70% of all exons are less than 200 base pairs in length [23]. However, due to the clustered, uneven distribution of exons, the optimal number and positions of baits to capture the whole targeted region with similar capture efficiency (and coverage) are not trivial [11]. Furthermore, extreme GC ratios, repetitive genomic contexts, and regions containing a high percentage of evolutionarily conserved genomic motifs have to be avoided in capture kit design. Accordingly, the spatiality of the genome context of the targeted and flanking regions also influences the selection of the optimal bait positions.

Because of the sheer number of baits and the complexity of the task in our work, we applied ML to predict the most likely bait positions of a capture kit. We evaluated the model performance based on the selected network architecture and hyperparameter optimization and provided input data to prove the robustness of our proposed CNN-1D model compared to the classical dense layer model with comparable N-bait counts. Our second approach to model performance evaluation was based on data window size adjustment, as the spatial context was expected to influence the predictions of the models.

In our data preparation, we split our data into equal (window size) segments. We randomized the offset of segmentation for each genomic region and sample to avoid bias from placing our predicted feature(s) in a nonrandom way inside the data segments. Since the majority of targeted (exons) regions are small (median exon size = 120 base pairs) but their position is not evenly dispersed in the human genome but rather clustered, in our data segments, we had a random number of oligo capture baits. According to our data, the majority of the data segments contained 1 or 2 oligo baits, while

fewer data segments contained more sparsely placed, potentially overlapping oligo baits. Approximately only 1/10th of the data segments contained 4 or more baits compared to a single bait. Furthermore, we expected that the difficulty of predicting the position of a single continuous bait region versus the potentially overlapping sparse positions of more baits could be different. Therefore, we also tested the feasibility of a reverse weighting approach. We classified the training examples into 5 classes based on the number of baits in the data segment (exactly 1, 1–2, 2–3, 3–4, 4 or more).

Our results show that 1D CNN with optimal parameters can be used to accurately predict the most likely bait design of complex target enrichment kits such as whole exome sequencing (WES) kits. In this task, the classic Dense NNs with a comparable number of trainable parameters perform worse and are also much more CPU intensive (Table 2, Fig. 4). Evaluation of our model shows that indeed each NN could predict a single or 1–2 baits (class 1–2) in the data segments with better accuracy and most of the bad predictions were from the more complicated class 3–5 data segments (Fig. 4, Additional file 2: Distribution of the accuracy, precision, sensitivity, specificity, MCC metrics for the 5 train classes based on the 1 M individual predictions of evaluation data set, ([https://doi.org/10.5281/zenodo.11102581](https://doi.org/10.5281/zenodo.11102581): eval_output]). Our results suggest that in case the train examples have large differences in their relative abundance in the train data set then reverse weighting for the harder, less abundant examples may improve the prediction accuracy. However, our results suggest that in the case of larger data windows (1000 base pairs or more) where a larger portion of the data has sufficient spatial context, reverse weighting may not be required.

Our results show that in case we provide the not normalized absolute genome coverage count data in our input the “batch normalization” option is required for stable CNN training. We also propose that batch normalization is also useful to apply the model on data from different sources as the absolute coverage in the data can vary between individual samples, used capture kits or laboratories. Interestingly this option has no effect on the Dense NNs. Other hyperparameters of CNNs, like the padding (valid, same or causal) play less role however causal and same options (padding of data) results in better accuracy than the “valid” option that uses only a portion of the data segment that is valid without padding.

Our results show that in case we provide the not normalized absolute genome coverage count data in our input the “batch normalization” option is required for stable CNN training. We also propose that batch normalization is also useful to apply the model on data from different sources as the absolute coverage in the data can vary between individual samples, used capture kits or laboratories. Interestingly this option has no effect on the Dense NNs. Other hyperparameters of CNNs, like the padding (valid, same or causal) plays less role however causal and same options (padding of data) results in better accuracy than the “valid” option that uses only portion of the data segment that is valid without padding. We have to note however, that our trained models are only valid on WES sequence data that has approximately $\sim 72\times$ mean target coverage used to train the models. Thus, coverage data of samples largely deviating from this mean coverage should be scaled accordingly. According to our results, the experimental coverage data, the target region information and also the sequence information in the included genome context improves the prediction power.

Our evaluation shows that the flanking genomic context is a major factor of prediction accuracy (Fig. 5, Additional file 3: Position metric plots of the evaluated models for 1000 and 500 train sizes) in the data segment. Based on this observation, for prediction it is recommended to place the on-target regions in the middle of the data segment as we observed markedly worse accuracy at the beginning and end of the data segment ($\approx$ 80 base pair = one bait length of the start and end of the data segment).

While the "on-target" regions largely overlap with the bait design positions, in the case of the complex regions considerable part of the target regions are not bait positions (Figs. 6, 7; Additional file 5: example plots of all models representative for the top 99, median, and worst 1 percentile predictions based on the F1 score rankings of 1 M predictions). Based on our experiments, even from 1 M train example, we can reach very high accuracy, specificity (0.97—0.98) and high (> 0.9) F1, MCC scores at 1000 base pair data segments indicating that in general, the predicted baits are >90% overlapping with the truth.

Our results also show that the used CNN has still generalization power providing more train examples, and likely more genomic context (with larger data segments) could also improve the prediction. Compared to the scenario when the bait positions are not included in the kit manifest and the "on-target" regions are used as the "predicted" bait positions, the CNN models provide much higher accuracy. This was also reflected by the most informative F1 and MCC scores of the same 1 M individual evaluation examples. For example, our proposed Conv1D model with a 1000 base pair long data window could predict the exact bait positions (F1 score = 1.0) in nearly the top 25 percentile of predictions while the scenario of on-target region used as the predicted bait positions resulted in no perfect match (with the highest F1 score of 0.994 and 9 FP positions in the data segment).

## Conclusion

Normalization of the read coverage profiles of complex WES kits is crucial for improving the sensitivity and specificity of CNV detection and thus could contribute to advancing our diagnostic capability for complex genetic disorders. One of the largest systemic biases is caused by the so called GC bias due to hidden differences in the wet-lab conditions between samples. While the "on-target" regions (that are provided for capture kits) largely overlap with the positions of the oligo capture bait coordinates, enabling normalization of the GC bias in many target regions, the exact bait positions could allow better GC bias normalization of the remaining more challenging regions. The predicted bait positions of our proposed 1D convolution model overlap >90% with the true bait positions. Consequently, downstream CNV detection based on these predicted coordinates to measure the coverage profile and normalize for GC bias could improve the normalization of CNV data, ultimately leading to better CNV prediction from targeted capture NGS data.

**Abbreviations**

CNVs: Copy number variations
NGS: Next-generation sequencing
CNN: Convolution neural network
WGS: Whole genome sequencing
WES: Whole-exome sequencing

SNV: Single nucleotide variant
TP: True positive
TN: True negative
FP: False positive
FN: False negative
ReLU: Rectified linear unit

## Supplementary Information

The online version contains supplementary material available at https://doi.org/10.1186/s12859-024-06006-y.

**Acknowledgements**
Tibor Kalmár and Zoltán Maróti acknowledge support from the National Laboratory of Biotechnology through Hungary from the National Research, Development, and Innovation Fund Office-NKFIH Fund No. NL-2022-00008. Peter Juma Ochieng and József Dombi acknowledge the support provided by the Ministry of Culture and Innovation of Hungary from the National Research, Development and Innovation Fund, financed under the TKP2021-NVA funding scheme, Project no TKP2021-NVA-09. The research was partially supported by the BioLOG project: Miklós Krész is grateful for the support of the National Center of Science (NCN) through grant DEC-2020/39/I/HS4/03533, the Slovenian Research and Innovation Agency (ARIS) through grant N1-0223 and the Austrian Science Fund (FWF) through grant I 5443-N. Miklós Krész has been also supported by the research program CogniCom (0013103) at the University of Primorska.

**Author contributions**
Conceptualization P.J.O., Z.M., T.K. and M.K. Data preprocessing, code for model training/evaluation Z.M. Formal analysis, model hyper parametrization P.J.O. and Z.M. Interpretation of the results Z.M., P.J.O., T.K., K.M. and J.D. Writing initial draft P.J.O. and Z.M. All authors took part in revising the results and contributed to the final manuscript. P.J.O. and Z.M. contributed equally to this study.

**Funding**
This work was supported by the University of Szeged Open Access Fund (Grant number: 6991).

**Availability of data and materials**
This study used the publicly available demo sequence data created by the Illumina Nextera WES kit available on BaseSpace (https://basespace.illumina.com/projects/206115910/about). Furthermore, the coverage profiles used in our research are only aggregate, derivative data of the original sequence data that is not representative of the genomic data of an individual and are influenced by the hidden factors of wet-lab preparation. The files containing the on-target, bait design and the chunk genome coordinates, preprocessed train, validation, evaluation data sets, the codes used in this study, the saved best loss/accuracy models, and all raw output of the evaluation are available at Zenodo [https://doi.org/10.5281/zenodo.11102581].

## Declarations

**Ethics approval and consent to participate**
Ethical approval and consent is not applicable to our research.

**Consent for publication**
Not applicable.

**Competing interests**
The authors declare that they have no competing interest.

Received: 3 May 2024 Accepted: 5 December 2024
Published online: 24 December 2024

**References**

1. McCarroll SA, Altshuler DM. Copy-number variation and association studies of human disease. Nature Genetics. 2007;39(Suppl 7):37–42.
2. Ewans LJ, Minoche AE, Schofield D, Shrestha R, Puttick C, Zhu Y, Drew A, Gayevskiy V, Elakis G, Walsh C. Whole exome and genome sequencing in mendelian disorders: a diagnostic and health economic analysis. Europ J Human Genet. 2022;30(10):1121–31.
3. Mazzarotto F, Olivotto I, Walsh R. Advantages and perils of clinical whole-exome and whole-genome sequencing in cardiomyopathy. Cardiovascul Drugs Therap. 2020;34(2):241–53.
4. Chung CC, Leung GK, Mak CC, Fung JL, Lee M, Pei SL, Mullin H, Hui VC, Chan JC, Chau JF, et al. Rapid whole-exome sequencing facilitates precision medicine in paediatric rare disease patients and reduces healthcare costs. The Lancet Regional Health–Western Pacific 2020;1.
5. Gabrielaite M, Torp MH, Rasmussen MS, Andreu-Sánchez S, Vieira FG, Pedersen CB, Kinalis S, Madsen MB, Kodama M, Demircan GS. A comparison of tools for copy-number variation detection in germline whole exome and whole genome sequencing data. Cancers. 2021;13(24):6283.
6. Pal A. Protocols in Advanced Genomics and allied techniques. Berlin: Springer; 2022.

7. Rajagopalan R, Murrell JR, Luo M, Conlin LK. A highly sensitive and specific workflow for detecting rare copy-number variants from exome sequencing data. Genome Med. 2020;12:1–11.
8. Välipakka S, Savarese M, Sagath L, Arumilli M, Giugliano T, Udd B, Hackman P. Improving copy number variant detection from sequencing data with a combination of programs and a predictive model. J Molecul Diagnost. 2020;22(1):40–9.
9. Barcelona-Cabeza R, Sanseverino W, Aiese Cigliano R. isocnv: in silico optimization of copy number variant detection from targeted or exome sequencing data. BMC Bioinformat. 2021;22:1–13.
10. Zhou J, Zhang M, Li X, Wang Z, Pan D, Shi Y. Performance comparison of four types of target enrichment baits for exome dna sequencing. Hereditas. 2021;158:1–12.
11. Tewhey R, Nakano M, Wang X, Pabón-Peña C, Novak B, Giuffre A, Lin E, Happe S, Roberts DN, LeProust EM. Enrichment of sequencing targets from the human genome by solution hybridization. Genome Biol. 2009;10:1–13.
12. Jiménez-Mena B, Flávio H, Henriques R, Manuzzi A, Ramos M, Meldrup D, Edson J, Pálsson S, Ásta Ólafsdóttir G, Ovenden JR. Fishing for dna? designing baits for population genetics in target enrichment experiments: Guidelines, considerations and the new tool superbaits. Molecul Ecol Resour. 2022;22(5):2105–19.
13. Barbitoff YA, Polev DE, Glotov AS, Serebryakova EA, Shcherbakova IV, Kiselev AM, Kostareva AA, Glotov OS, Predeus AV. Systematic dissection of biases in whole-exome and whole-genome sequencing reveals major determinants of coding sequence coverage. Scient Rep. 2020;10(1):2057.
14. Chai J, Zeng H, Li A, Ngai EW. Deep learning in computer vision: a critical review of emerging techniques and application scenarios. Mach Learn Appl. 2021;6: 100134.
15. Li H. Aligning sequence reads, clone sequences and assembly contigs with bwa-mem. arXiv preprint arXiv:1303.3997 2013.
16. Quinlan AR, Hall IM. Bedtools: a flexible suite of utilities for comparing genomic features. Bioinformatics. 2010;26(6):841–2.
17. Danecek P, Bonfield JK, Liddle J, Marshall J, Ohan V, Pollard MO, Whitwham A, Keane T, McCarthy SA, Davies RM. Twelve years of samtools and bcftools. Gigascience. 2021;10(2):008.
18. Li L, Doroslovački M, Loew MH. Approximating the gradient of cross-entropy loss function. IEEE Access. 2020;8:111626–35.
19. Nyerki E, Kalmár T, Schütz O, Lima RM, Neparáczki E, Török T, Maróti Z. correctkin: an optimized method to infer relatedness up to the 4th degree from low-coverage ancient human genomes. Genome Biol. 2023;24(1):38.
20. Rao J, Peng L, Liang X, Jiang H, Geng C, Zhao X, Liu X, Fan G, Chen F, Mu F. Performance of copy number variants detection based on whole-genome sequencing by dnbseq platforms. BMC Bioinformat. 2020;21:1–14.
21. Tilemis F-N, Marinakis NM, Veltra D, Svingou M, Kekou K, Mitrakos A, Tzetis M, Kosma K, Makrythanasis P, Traeger-Synodinos J. Germline cnv detection through whole-exome sequencing (wes) data analysis enhances resolution of rare genetic diseases. Genes. 2023;14(7):1490.
22. Uchiyama Y, Yamaguchi D, Iwama K, Miyatake S, Hamanaka K, Tsuchida N, Aoi H, Azuma Y, Itai T, Saida K. Efficient detection of copy-number variations using exome data: Batch-and sex-based analyses. Human Mutat. 2021;42(1):50–65.
23. Mokry M, Feitsma H, Nijman IJ, Bruijn E, Zaag PJ, Guryev V, Cuppen E. Accurate snp and mutation detection by targeted custom microarray-based genomic enrichment of short-fragment sequencing libraries. Nucl Acid Res. 2010;38(10):116–116.


## Publisher's Note