DeNovoCNN: A deep learning approach to *de novo* variant calling in next generation sequencing data

Gelana Khazeeva[1,*], Karolis Sablauskas[2,*], Bart van der Sanden[3], Wouter Steyaert[1], Michael Kwint[1], Dmitrijs Rots[3], Max Hinne[4], Marcel van Gerven[4], Helger Yntema[3], Lisenka Vissers[3], Christian Gilissen[1,#]

1. Department of Human Genetics, Radboud Institute for Molecular Life Sciences, Radboud University Medical Center, Geert Grooteplein 10, 6525 GA Nijmegen, The Netherlands
2. Institute of Clinical Medicine, Faculty of Medicine, Vilnius University, Vilnius, Lithuania
3. Department of Human Genetics, Donders Centre for Neuroscience, Radboud University Medical Center, Geert Grooteplein 10, 6525 GA Nijmegen, The Netherlands
4. Donders Institute for Brain, Cognition and Behaviour, Radboud University Nijmegen

* these authors contributed equally

[#] Correspondence to: christian.gilissen@radboudumc.nl

ABSTRACT

De novo mutations (DNMs) are an important cause of genetic disorders. The accurate identification of DNMs from sequencing data is therefore fundamental to rare disease research and diagnostics. Unfortunately, identifying reliable DNMs remains a major challenge due to sequence errors, uneven coverage, and mapping artifacts. Here, we developed a deep convolutional neural network (CNN) DNM caller (DeNovoCNN), that encodes alignment of sequence reads for a trio as 160×164 resolution images. DeNovoCNN was trained on DNMs of whole exome sequencing (WES) of 2003 trios achieving on average 99.2% recall and 93.8% precision. We find that DeNovoCNN has increased recall/sensitivity and precision compared to existing *de novo* calling approaches (GATK, DeNovoGear, Samtools) based on the Genome in a Bottle reference dataset. Sanger validations of DNMs called in both exome and genome datasets confirm that DeNovoCNN outperforms existing methods. Most importantly, we show that DeNovoCNN is robust against different exome sequencing and analyses approaches, thereby allowing it to be applied on other datasets. DeNovoCNN is freely available and can be run on existing alignment (BAM/CRAM) and variant calling (VCF) files from WES and WGS without a need for variant recalling.

INTRODUCTION

Many developmental disorders, such as intellectual disability (1), autism spectrum disorder (2) and multiple congenital anomalies (3) are known to be caused by *de novo* mutations (DNMs) (4,5). The reliable identification of DNMs is, therefore, of paramount importance both for genetic testing as well as research studies. Because of the genetic heterogeneity that exists for disorders where DNMs play a major role, the identification of DNMs is typically performed based on whole exome (WES) or whole genome sequencing (WGS) data. In principle DNMs can be easily identified by selecting variants in the proband that are not present in either of the parents. In practice, however, this process is complicated by sequencing artifacts, mapping artifacts, differences in sequence coverage and mosaicism. Moreover, the genome of an average individual has 40 - 80 DNMs of which on average 1.45 occur in the coding regions (6), making DNMs considerably rarer than errors associated with sequencing technology. Practically this means that sensitivity and specificity of DNM detection is usually balanced by selecting appropriate quality score cutoffs.

Several different methods have been developed to identify DNMs in next-generation sequencing (NGS) data. With methods such as DeepTrio (7) and the Genome Analysis Toolkit (GATK) (8) *de novo* calling is achieved straightforwardly by performing multi-sample variant calling and subsequent selection of variants based on genotypes corresponding to *de novo* mutations. The downside of these approaches is that DNM calling is dependent on the variant calling, which therefore always needs to be performed with the same method. For existing dataset this potentially results in a high computational and storage overhead. Other tools, such as DeNovoGear (9) and TrioDeNovo (10) are able to call DNMs based on existing variant calls by modeling the probability of mutation transfer using mutation rate priors. All of these approaches provide high sensitivity, but the specificity is usually lower due to the amount of noise in NGS data, resulting in a high number of false positive calls (11). Subsequent filtering of DNMs based on quality criteria is, therefore, typically required.

Deep learning, a field of machine learning, has recently seen a growth in popularity amongst applications in genomics (12). Deep learning approaches have been able to achieve improvements in many genomics applications by converting genomic data into an image-like representation and employing convolutional neural networks (CNNs) (e.g. tumor type classification using RNA-Seq data (13) and germline variant calling (7)). Here we developed DeNovoCNN, a deep-learning model that encodes trio NGS data as images and uses a suite of CNNs to detect *de novo* mutations in next generation sequencing data.

MATHERIAL AND METHODS

Training and validation dataset

A cohort of 2,418 child-parent trios was used for building the training and validation datasets. All of the individuals were initially referred to the Radboudumc Department of Human Genetics with an indication of unexplained intellectual disability, for whom trio WES was performed as described before (5). Briefly, all samples were sequenced on Illumina HiSeq 2000/4000 instruments using Agilent SureSelect v4 or v5 exome enrichment kits, respectively. Initially *de novo* calling was performed using our in-house method based on Samtools. Subsequently, all calls were filtered based on quality metrics (5). The DNM calling yielded 4,183 DNM calls in 2,019 trios out of 2,418 trios, which were used to construct the training and validation datasets (Supplementary Figure 1).

Snapshots of all of the potential DNM calls were generated using the Integrative Genomics Viewer (IGV) (14) for visual inspection, and each variant was evaluated by assigning it to one of the three classes: DNM, IV (inherited variant) or UN (unknown) for cases where it was not feasible to make the confident decision on visual inspection alone. UN variants were removed from the dataset. The resulting dataset of 2,003 trios with 3,681 DNMs and 394 IVs were randomly separated into training and validation sets using a 80/20% split (Supplementary Figure 1).

Despite the large exome dataset, the total number of DNMs and IVs for training was relatively low. Therefore, we supplemented the DNM dataset by performing DNM calling using the in-house caller on 2 artificial trios where the child was unrelated to the parents. These 2 trios were constructed by randomly sampling 2 parent pairs, followed by the random choice of a child from the 1,602 trios training dataset (Supplementary Figure 1). This resulted in an additional 1,005 DNMs that were added to the training dataset. For IVs we randomly chose 42 trios from the training and validation cohort and selected 300 IVs from each trio resulting in a total of 9,900 IVs and 2,700 IV for the training and validation sets respectively (Supplementary Figure 1).

One of the challenges of DNM detection is to distinguish false positives in difficult genomic regions, so we developed a way to add such examples to the training and validation set. First, we took the current training and validation datasets to build a first-generation algorithm for DNMs calling. Second, we randomly selected 29 and 6 trios from training and validation datasets respectively and applied first-generation models to get candidate DNM calls on these trios. Finally, we manually curated all calls in IGV, selecting only those calls that were either clearly inherited from the parents or occurred in difficult regions where a lot of sequencing mistakes and artefacts were visible. This yielded 318 and 54 IVs for the training and validation datasets respectively and provided a better representation of the locations where our algorithm made the most mistakes in the previous step. All IVs and DNMs were further assigned into three categories: insertions,

deletions and single-nucleotide substitutions for training of the three different models. (Supplementary Table 1).

DeNovoCNN

Model architecture

We aimed to replicate the visual inspection process of possible DNMs performed by human experts using software such as IGV. By converting NGS data into RGB images *de novo* variant calling could be approached as a computer vision classification task with two classes: DNMs and IVs. The state-of-the-art approach for vision classification tasks is the convolutional neural network (CNN), a variation of which we chose for our purposes. The model architecture consisted of 9 2D convolutional layers with 128 filters, 3x3 kernels, ReLU activation and the same padding in each layer. After every third convolutional layer we applied batch normalization and added a Squeeze-and-Excitation block (15). Global max and average pooling were applied before the output layer (Supplementary Figure 2). The architecture was developed using Python with the Keras v.2.2.2 library (16) and TensorFlow v.1.10.0 backend (17). Using this architecture we constructed three separate models, for insertions, deletions and substitutions because of their specific visual patterns.

Image generation

Variants in *de novo* and control datasets were converted into images prior to being fed to the convolutional neural network. All variants of interest were converted into 160×164 RGB images (Supplementary Figure 3). Image generation was based on read pileup data in the location of the variant capturing 20 nucleotides before and after the candidate DNM. Read pileup data from individual trio members for the same variant position was extracted using the Pysam v.0.8.4 library (18).

Each row in the image encodes a read base sequence. Image columns were structured in a recurring pattern of 4 pixels per position, corresponding to A, C, T and G bases respectively. Thus, the image width of 164 pixels represents a sequence of 41 (164 / 4) bases with the variant starting at the center position (20 using 0 indexing). The pixel of the tuple (A, C, T, G) was filled with a value in the resulting image in case it corresponded to the base in the particular genomic position of the read, whereas the rest were filled with zeros (Figure 1). Pixel intensities have a maximum value of 255, adjusted by mapping and base quality scores with higher quality corresponding to higher pixel intensity. Each column represents the sequencing depth which was limited to 160 reads for computational performance. Red, green and blue color channels represent different individuals of the trio, corresponding to child, father and mother respectively.

Training the model

Due to the large dataset size, the substitution network was trained first using random weight initialization for 45 epochs, while insertion and deletion networks were trained for 45 epochs each, using weights from

the trained substitution network as the starting point. Batch size was set to 32 and Adadelta optimizer with default Keras parameters was used for minimization of binary cross-entropy loss in all models. The initial learning rate was set to 0.1 with a stepwise decay of 0.5 every 10 epochs. The output of the network is a vector containing probabilities for a variant being a DNM and IV. The area under the curve (AUC), overall accuracy, specificity, sensitivity and F1 score were calculated on the validation sets.

Data augmentation was applied during training the networks for substitutions, deletions and insertions. The standard augmentations included random brightness adjustment by a factor between [0.3, 1.] and horizontal flip. To increase the stability of the model, random shuffling of the reads and nucleotide labels swapping were implemented. Additionally, we observed that some specific cases of DNMs were underrepresented in our dataset, namely, DNMs in low-coverage regions and multi-nucleotide substitutions. We simulated reduced coverage by discarding a random number of reads from the pileup and enriched the dataset for multi-nucleotide substitutions by generating adjacent substitutions using the substitution dataset on-the-fly. In order to obtain the most consistent results, we eventually combined both parental channels into one using a pixelwise maximum operation, before passing the images to the neural network.

DNM prediction

After the training DeNovoCNN was used for DNM prediction on new data. The input consists of one VCF and one BAM file per sample (three per trio), as well as paths to the trained model weights for substitutions, insertions and deletions. Inference consists of two steps. The first step is dependent on the initial variant calling used for generating VCF files: using the trio VCF files, the inherited variants are discarded. This is achieved using the `bcftools isec -C child_vcf father_vcf mother_vcf` (18) command and results in around 10-fold reduction of the number of genomic locations for evaluation, which usually ends up <10,000 variants for WES (depending on capture kit size) and ~100,000 for WGS. The second step iterates through the intersected list and classifies each variant as DNM or non-DNM. The variant is considered to be *de novo* if the probability of DNM class returned by DeNovoCNN is higher or equal 0.5. During the inference, we discarded low coverage regions by using a minimum coverage requirement of 7 sequencing reads for the genomic positions in all individuals of a trio. Variants on chromosome Y were not considered.

Model validation

In order to validate the performance of the proposed deep learning model, we used datasets from different types of sequencing (exomes and genomes) as well as different types of sequencing and enrichment platforms (Supplementary Table 2, Supplementary Methods).

DeNovoCNN was compared to other algorithms, such as GATK PossibleDeNovo (8), DeNovoGear (9) and our in-house *de novo* detection algorithm.

For the validation purposes the following algorithms were run on our datasets.

- DeNovoCNN takes as an input trio VCF files and BAM/CRAM files. See DNM prediction section for the details.

- DeNovoGear version 1.1.1 (9) was run according to the specifications. DeNovoGear takes in PED and BCF files as an input. The BCF file was generated using following command:
  ```
  samtools mpileup -gDf reference.fa child.bam father.bam mother.bam
  ```

- GATK (gatk4-4.1.2.0 and gatk4-4.1.8.1) was run on BAM/CRAM or gVCF files according to the best practices for germline short variant discovery (SNPs + indels) (8).

- Our in-house method: this is an in-house developed method that generates a list of *de novo* candidates based on the VCF files of the trio (1). For WES remaining candidates were then filtered out based on gnomAD allele frequency <1.0%. For WGS data gnomAD allele frequency <0.1% and xAtlas quality score >15 filters were used. Subsequently this method performs Samtools pileups which are used to select the most likely DNM candidates based on a set of hard-filters.

RESULTS

We used a convolutional neural network (CNN) with squeeze-and-excitation blocks (15) as the architecture for DeNovoCNN (Figure 1, Supplementary Figure 2), and trained three separate models for substitutions, insertion and deletion variants. Our primary training dataset was based on DNMs and inherited variants (IVs) from a cohort of 2,418 child-parent trios that was supplemented with simulated data in order to optimize training (5), (Supplementary Figure 1, Material and Methods). All variants were converted into 160×164 RGB images (Supplementary Figure 3). The dataset was split into an 80% training (3,956 DNMs, 10,531 IVs) and 20% validation (730 DNMs, 2,835 IVs; Supplementary Table 1) dataset. DeNovoCNN generates a probability of a variant being *de novo*, and, therefore, we used a threshold of ≥0.5 to select *de novo* calls. After training DeNovoCNN achieved a high average validation sensitivity/recall rate of 97.53%, 100%, 100% and precision of 97.53%, 90.91% and 92.86% for substitutions, insertions and deletions, respectively (Supplementary Figure 4, Supplementary Figure 5, Supplementary Table 3).

Comparison on GIAB WGS dataset

In order to compare our method to other *de novo* detection methods on an independent dataset we used Illumina WGS data of an Ashkenazim Trio (NA24385; NA24149; NA24143) from the Genome in a Bottle (GIAB) consortium (19). This trio was sequenced using various different technologies in order to create a dataset of 1,342 high-quality cell-line and germline DNMs. We compared DeNovoCNN DNMs to DNMs from our in-house methods based on Samtools (18), GATK (8), GATK filtered for high confidence DNM calls (GATK_HC), and DeNovoGear filtered using a ≥0.5 and ≥0.9 probability thresholds (DeNovoGear-0.5, DeNovoGear-0.9) (9) (Figure 2A, Supplementary Table 4, Supplementary Figure 6). DeNovoCNN outperformed other algorithms with precision rate of 95.18%. Our in-house tool showed the precision of 80.28%, while GATK_HC and DeNovoGear-0.9 performance were 54.5% and 32.99% respectively. Our in-house has the highest sensitivity/recall rate of 89.2%, whereas DeNovoCNN showed a slightly lower sensitivity/recall rate of 88.38%. GATK and DeNovoGear-0.5 performance were 87.41% and 79.28% respectively.

We observed a relatively low recall for insertions and deletions of all algorithms which led us to investigate possible problems with GIAB high quality DNM dataset. Therefore, we performed manual cleaning of likely false positive *de novo* insertion and deletion calls in the GIAB gold standard dataset based on visual inspection (Supplementary Methods). We then repeated our comparison on this manually curated GIAB dataset and now found that DeNovoCNN outperformed all other methods on sensitivity/recall as well as precision and other metrics (Figure 2B, Supplementary Table 5).

Comparison on 20 WES trios

The comparison on the GIAB dataset highlighted the difficulties in obtaining high-quality validation datasets for DNMs. Therefore, we also compared DeNovoCNN performance with GATK_HC, DeNovoGear and our in-house tool on a dataset of 20 randomly selected in-house WES trios that were not previously used for training (Figure 3A). This allowed us to validate the called DNMs experimentally. For all of the DNM calls by the different methods we performed a visual selection to discard obvious false positive variants. The remaining 48 variants were validated by Sanger/IonTorrent sequencing of which 26 were confirmed as DNMs, and 3 were either inherited or false positives. For 19 variants it was not possible to perform the validation due to difficulties with designing the primers (6 variants) and depletion of the DNA sample (13 variants) (Supplementary Figure 7). Based on these validations, DeNovoCNN was able to correctly identify all confirmed DNMs (sensitivity/recall: 100%, next best DeNovoGear-0.5, In-house: 76.92%), showing slightly lower performance on other metrics than our in-house Samtools-based method (Supplementary Table 6).

Results on the multi-platform WES trios dataset

Next, we wanted to verify that DeNovoCNN is robust across different capture and sequencing approaches. We used an exome dataset of the Solve-RD consortium (20) that contains 313 trios sequenced across 13 different capture/sequencing combinations (Supplementary Methods; Supplementary Table 7). We measured the robustness of our method by considering the number of called DNMs per sample and compared this to the number of high quality DNM calls from GATK (GATK_HC) (Figure 3B, Supplementary Figure 8). In addition, we expected that the number of calls is within the same range regardless of the sequencing platform that was used. The median number of DeNovoCNN calls is 3.0, the 5th and 95th percentiles are 1.0 and 5.0 whereas the overall distribution lies between 0 and 15 calls. This result is consistent with what we observe for GATK calling (median number of calls is 3.5, the 5th and 95th percentiles are 1.65 and 7.95). This confirms that our method was not overfitted to the specific capture kit and sequencing instrument of the training dataset.

Results on 5 WGS trios

In order to confirm that DeNovoCNN also performs well on whole genome sequencing data, we used 5 in-house WGS trios. We applied DeNovoCNN to get DNMs for the trios that resulted in 1,022 calls. We compared our results only to those of GATK, as this is currently the most commonly used method for variant calling for WGS. We overlapped DeNovoCNN and GATK high quality calls dividing them into 3 groups: 469 called by both tools, 553 called only by DeNovoCNN and 3,010 called only by GATK (Figure 4). We randomly selected variants from each group, proportional to the group size, (7 for DeNovoCNN-only, 31 for GATK-only and 25 for both) and performed Sanger/IonTorrent validations (Supplementary Table 8). Out of the 25 variants that were called by both methods, 23 were confirmed as real *de novo*. For the 31 variants only called by GATK we were able to perform validations for 21 variants and all of them were false positives.

For the 7 variants only called by DeNovoCNN validation only succeeded for one, which turned out to be a real *de novo* variant. It was not possible to perform validations for 17 variants due to difficulty of designing primers in highly repetitive regions. Overall, DeNovoCNN identified slightly more DNMs than GATK but with an increased specificity.

DISCUSSION

Deep-learning methods provide a powerful new tool for genetics research. However, their application comes with certain challenges. The biggest challenge of DeNovoCNN is common for most deep learning models and is related to the fact the model generalizes well to the data that has been used in training. It can therefore be sensitive to subtle differences in the data that then lead to unexpected results. For our model we exclusively used data from a single center for training, albeit from two different capture kits and sequencing instruments. Despite the fact that we have used data augmentation techniques to prevent overfitting, and our results indicate that our model appears to be agnostic to the type of sequencing experiment (WES or WGS) and sequencing platform, we cannot exclude that some residual bias from the training data exists. We observed only one such bias that relates to the fact that all BAM files in our training dataset underwent Base Quality Score Recalibration (BQSR) (8). While not evaluated in detail, we noted that using non-BQSR BAM files generally results in a lower number of potential DNM calls.

While our model was trained on BAM files, CRAM is becoming a new data standard (21). Therefore, we also compared DeNovoCNN results from BAM files and corresponding CRAM files with quality-score binning. We find that the correlation between the prediction on BAM and CRAM files is very high (Supplementary Figure 9) and therefore expect DeNovoCNN to work equally well on CRAM files.

Recently, DeepTrio (7), an extension of DeepVariant (22) was developed. In our study we did not compare our results to those of DeepTrio because it is fundamentally a variant calling algorithm, where in order to identify DNMs, the genotypes of the variants are combined to generate a list of candidate DNMs. Currently most groups use GATK as their default variant calling algorithm and, therefore, we only include GATK in our comparison. Conversely, DeNovoCNN is not depended on the variant calling method itself, which will make it easier to use on existing datasets and incorporate in existing data analysis pipelines. Application of DeNovoCNN on variants called by a completely different variant caller (xAtlas) in 5 trio WGS resulted in similar numbers and types of DNM calls as for GATK based calls.

Although DeNovoCNN shows overall good performance compared to other methods, we have seen some possibilities for future improvements. We noticed that the performance on indels is lower than the performance on substitutions. This could be explained by the fact that the amount of the indels events in training dataset is much lower than substitutions events. In addition, DNMs were mostly validated using visual inspection. Because indel *de novo* events are more difficult to distinguish manually from sequencing

artefacts, this could have led to a poorer training dataset specifically for these events. In general, we remark that for future improvements of DNM detection, it will be essential to have sizeable, curated training datasets.

Other possible improvements could lie in the model itself. The training of deep learning models is computationally intensive, which is why we chose a relatively simple CNN architecture and implemented some hard cutoffs. Although we did not observe any clear negative effects on model performance, more complex architectures and more context information for variants could potentially improve the performance further.

AVAILABILITY

DeNovoCNN is an open source software available in the GitHub repository (https://github.com/Genome-Bioinformatics-RadboudUMC/DeNovoCNN).
The training dataset for DeNovoCNN is available in the GitHub repository (https://github.com/Genome-Bioinformatics-RadboudUMC/DeNovoCNN_training_dataset).

ACKNOWLEDGEMENT

The aims of this study contribute to the Solve-RD project (to CG and LELMV) which has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 779257.

FUNDING

This project was partly funded by The Netherlands Organisation for Scientific Research (917-17-353) to CG.

CONFLICT OF INTEREST

The authors declare that there is no conflict of interest.

REFERENCES

1. Lelieveld, S.H., Reijnders, M.R., Pfundt, R., Yntema, H.G., Kamsteeg, E.J., de Vries, P., de Vries, B.B., Willemsen, M.H., Kleefstra, T., Lohner, K. et al. (2016) Meta-analysis of 2,104 trios provides support for 10 new genes for intellectual disability. Nat Neurosci, 19, 1194-1196.
2. Satterstrom, F.K., Kosmicki, J.A., Wang, J., Breen, M.S., De Rubeis, S., An, J.Y., Peng, M., Collins, R., Grove, J., Klei, L. et al. (2020) Large-Scale Exome Sequencing Study Implicates Both Developmental and Functional Changes in the Neurobiology of Autism. Cell, 180, 568-584 e523.

3. Homsy, J., Zaidi, S., Shen, Y., Ware, J.S., Samocha, K.E., Karczewski, K.J., DePalma, S.R., McKean, D., Wakimoto, H., Gorham, J. et al. (2015) De novo mutations in congenital heart disease with neurodevelopmental and other congenital anomalies. Science, 350, 1262-1266.

4. Vissers, L.E., Gilissen, C. and Veltman, J.A. (2016) Genetic studies in intellectual disability and related disorders. Nat Rev Genet, 17, 9-18.

5. Kaplanis, J., Samocha, K.E., Wiel, L., Zhang, Z., Arvai, K.J., Eberhardt, R.Y., Gallone, G., Lelieveld, S.H., Martin, H.C., McRae, J.F. et al. (2020) Evidence for 28 genetic disorders discovered by combining healthcare and research data. Nature, 586, 757-762.

6. Goldmann, J.M., Veltman, J.A. and Gilissen, C. (2019) De Novo Mutations Reflect Development and Aging of the Human Germline. Trends Genet, 35, 828-839.

7. Kolesnikov, A., Goel, S., Nattestad, M., Yun, T., Baid, G., Yang, H., McLean, C.Y., Chang, P.-C. and Carroll, A. (2021) DeepTrio: Variant Calling in Families Using Deep Learning. bioRxiv.

8. DePristo, M.A., Banks, E., Poplin, R., Garimella, K.V., Maguire, J.R., Hartl, C., Philippakis, A.A., del Angel, G., Rivas, M.A., Hanna, M. et al. (2011) A framework for variation discovery and genotyping using next-generation DNA sequencing data. Nat Genet, 43, 491-498.

9. Ramu, A., Noordam, M.J., Schwartz, R.S., Wuster, A., Hurles, M.E., Cartwright, R.A. and Conrad, D.F. (2013) DeNovoGear: de novo indel and point mutation discovery and phasing. Nat Methods, 10, 985-987.

10. Wei, Q., Zhan, X., Zhong, X., Liu, Y., Han, Y., Chen, W. and Li, B. (2015) A Bayesian framework for de novo mutation calling in parents-offspring trios. Bioinformatics, 31, 1375-1381.

11. Liang, Y., He, L., Zhao, Y., Hao, Y., Zhou, Y., Li, M., Li, C., Pu, X. and Wen, Z. (2019) Comparative Analysis for the Performance of Variant Calling Pipelines on Detecting the de novo Mutations in Humans. Front Pharmacol, 10, 358.

12. Eraslan, G., Avsec, Z., Gagneur, J. and Theis, F.J. (2019) Deep learning: new computational modelling techniques for genomics. Nat Rev Genet, 20, 389-403.

13. Lyu, B. and Haque, A. (2018), Proceedings of the 2018 ACM international conference on bioinformatics, computational biology, and health informatics, pp. 89-96.

14. Robinson, J.T., Thorvaldsdottir, H., Winckler, W., Guttman, M., Lander, E.S., Getz, G. and Mesirov, J.P. (2011) Integrative genomics viewer. Nat Biotechnol, 29, 24-26.

15. Hu, J., Shen, L., Albanie, S., Sun, G. and Wu, E. (2020) Squeeze-and-Excitation Networks. IEEE Trans Pattern Anal Mach Intell, 42, 2011-2023.

16. Chollet, F. and others. (2015). http://keras.io

17. Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M. et al. (2016), Proceedings of the 12th USENIX conference on Operating Systems Design and Implementation. USENIX Association, Savannah, GA, USA, pp. 265–283.

18. Li, H., Handsaker, B., Wysoker, A., Fennell, T., Ruan, J., Homer, N., Marth, G., Abecasis, G., Durbin, R. and Genome Project Data Processing, S. (2009) The Sequence Alignment/Map format and SAMtools. Bioinformatics, 25, 2078-2079.

19. Zook, J.M., McDaniel, J., Olson, N.D., Wagner, J., Parikh, H., Heaton, H., Irvine, S.A., Trigg, L., Truty, R., McLean, C.Y. et al. (2019) An open resource for accurately benchmarking small variant and reference calls. Nat Biotechnol, 37, 561-566.

20. Zurek, B., Ellwanger, K., Vissers, L., Schule, R., Synofzik, M., Topf, A., de Voer, R.M., Laurie, S., Matalonga, L., Gilissen, C. et al. (2021) Solve-RD: systematic pan-European data sharing and collaborative analysis to solve rare diseases. Eur J Hum Genet.

21. Lelieveld, S.H., Veltman, J.A. and Gilissen, C. (2016) Novel bioinformatic developments for exome sequencing. Hum Genet, 135, 603-614.

22. Poplin, R., Chang, P.C., Alexander, D., Schwartz, S., Colthurst, T., Ku, A., Newburger, D., Dijamco, J., Nguyen, N., Afshar, P.T. et al. (2018) A universal SNP and small-indel variant caller using deep neural networks. Nat Biotechnol, 36, 983-987.

FIGURES

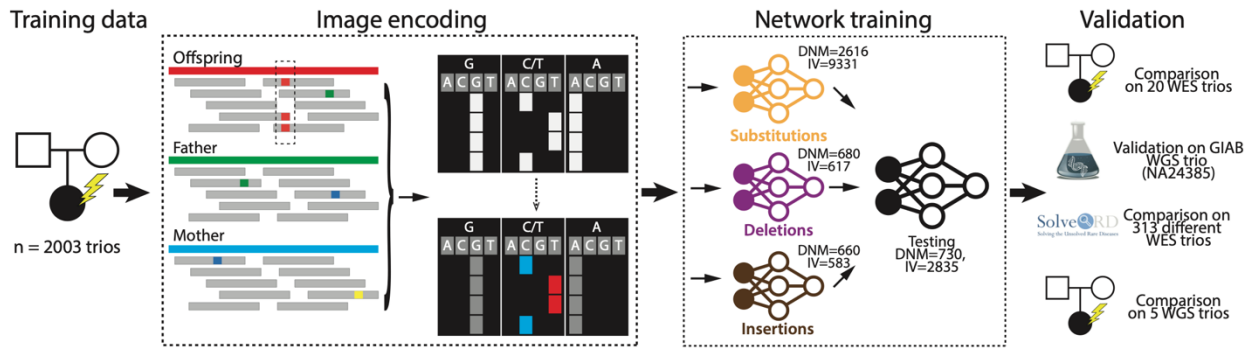
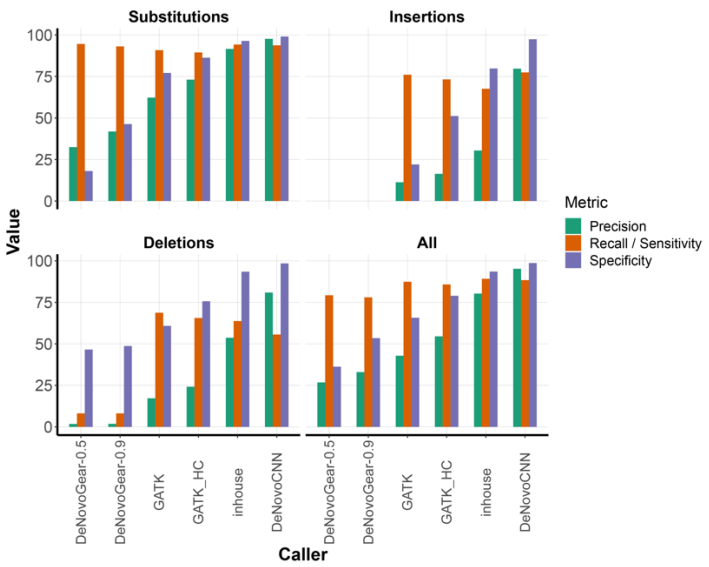


Figure 1. Overview of the method depicting the different steps starting with (from left to right), the training data, encoding of sequence data as images, training of the different deep-learning models and validation of the final model. The first step consisted of the construction of training and validation datasets using *de novo* and inherited or false *de novo* variants from 2003 trios. As is shown in the “Image encoding” section of the figure all variants were transformed to RGB images, where each color channel corresponds to the child, father and mother data respectively. Every row encodes separate read, every nucleotide in particular position is encoded as 4 consecutive columns for A, C, T, G. 80% of these images were then used to train 3 convolutional neural networks (for substitutions, deletions and insertions variants). The remaining 20% of the data (730 DNM variants, 2835 IV variants) was used as validation dataset. The resulting models were thoroughly validated using various datasets: the external gold standard GIAB WGS trio, the in-house 20 WES dataset, the in-house 5 WGS dataset, and the multi-platform 313 WES trios from SolveRD project.

Figure 2. The comparison results on the GIAB reference trio. Each subfigure contains 4 subgraphs of the different tools' performance for substitutions, insertions, deletions, and all variants. All tools for the comparison are on the horizontal axis: DeNovoGear with threshold of 0.5, DeNovoGear with threshold of 0.9, GATK, GATK high quality calls only, our in-house tool, and DeNovoCNN. The green bars show precision, orange bars show recall (sensitivity), violet bars show specificity.

a) Precision and recall of different *de novo* calling methods for the GIAB reference genome and gold standard DNM call.



b) Same as in a) but using the manually curated set of DNMs (Material and Methods).

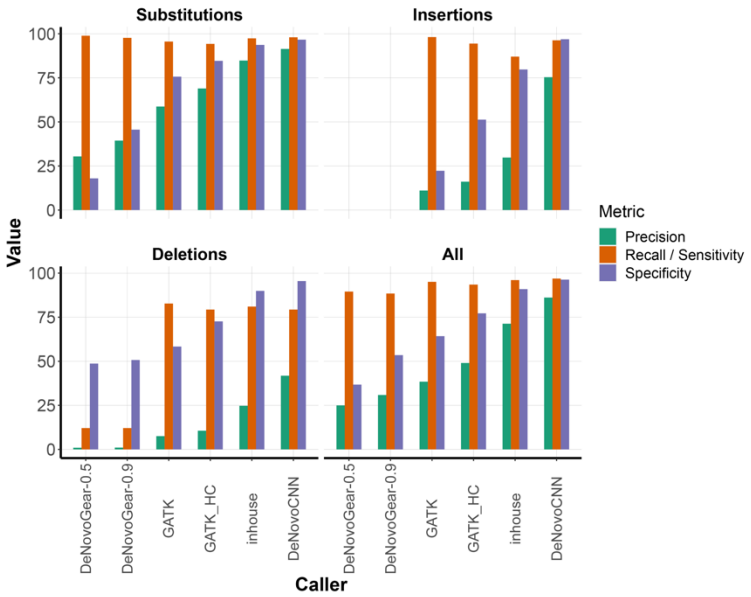
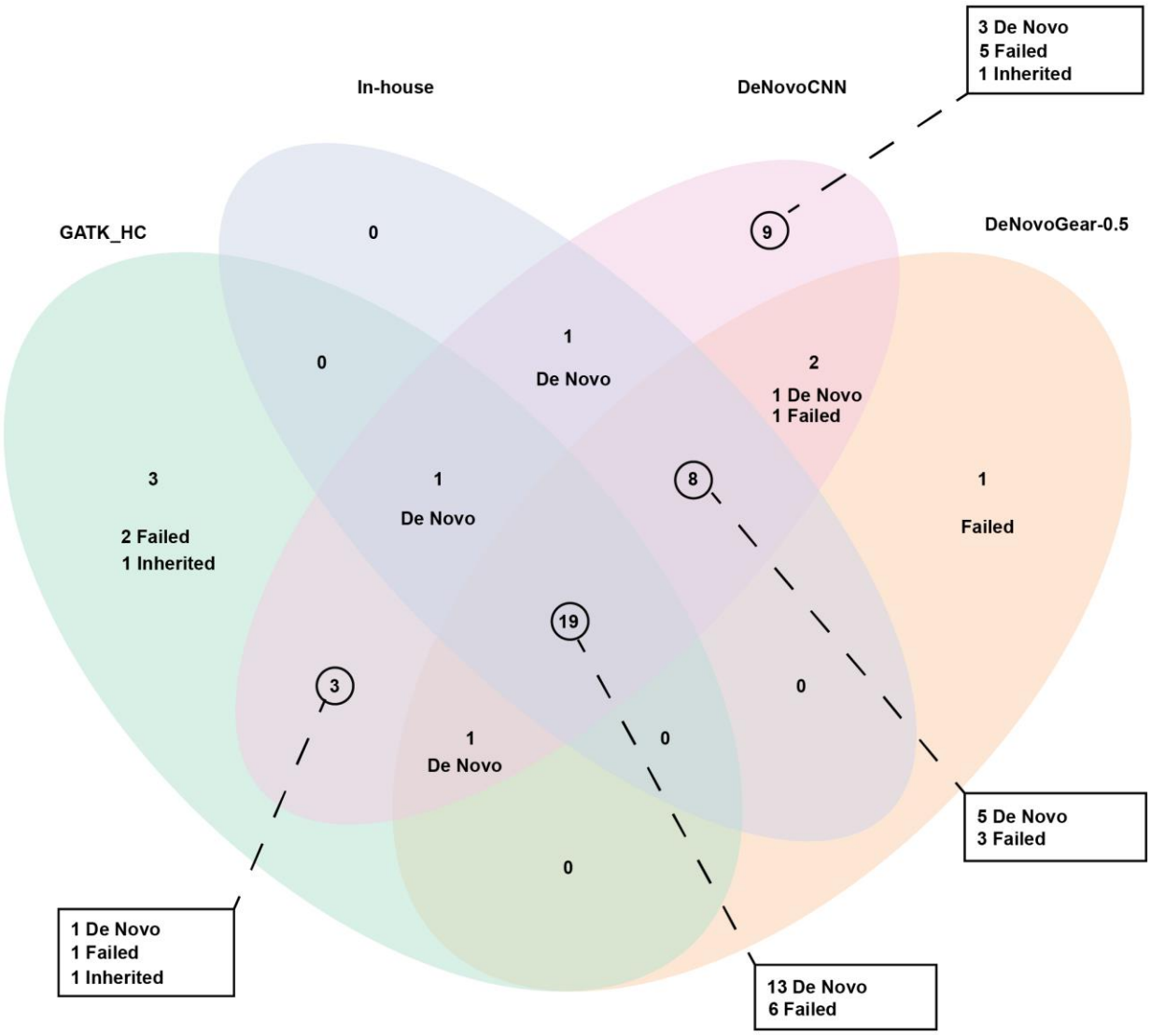
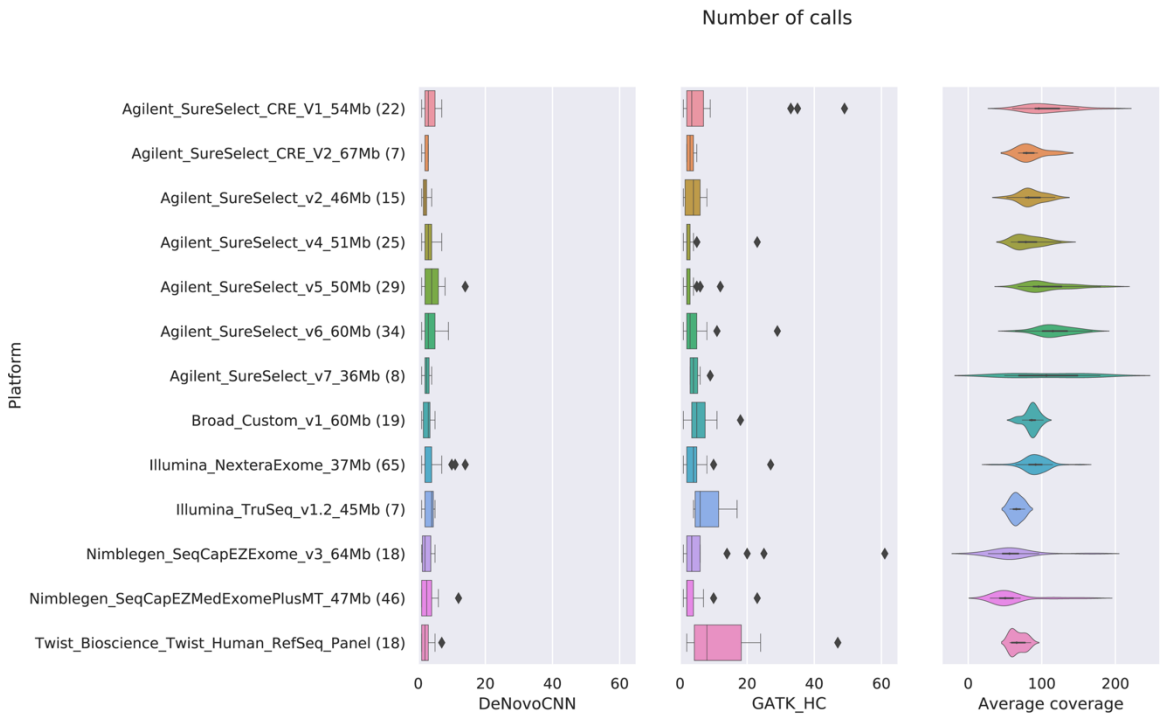


Figure 3. **Validation and comparison of the DeNovoCNN method on exome data.**

a) Results on 20 in-house WES trios of which mutations were validated by Sanger sequencing. The graph contains only variants that were marked as potentially *de novo* after manual validation in IGV, and then sent for Sanger validation (see Materials and Methods). The green oval contains all potentially *de novo* variants for GATK tool with high quality filter, the light blue oval indicate potential *de novo* variants according to the in-house tool, the magenta oval contains potentially *de novo* variants according to the DeNovoCNN tool, and finally light brown oval shows potentially *de novo* variants based on the DeNovoGear tool. Each intersection of the circles contains the number of potentially *de novo* variants that were found in the intersection of the corresponding tools' calls. For each number, the results of the Sanger validation of these variants shown.



b) The DeNovoCNN and GATK results on samples from different exome platforms. For each platform the numbers of samples used is indicated in brackets. The first two graphs show the distribution of the number of calls (on the horizontal axis) per platform using boxplots for DeNovoCNN and GATK with high quality filter respectively. The last graph shows the distribution of the average coverage per platform using violin plots. Each color corresponds to a particular platform.



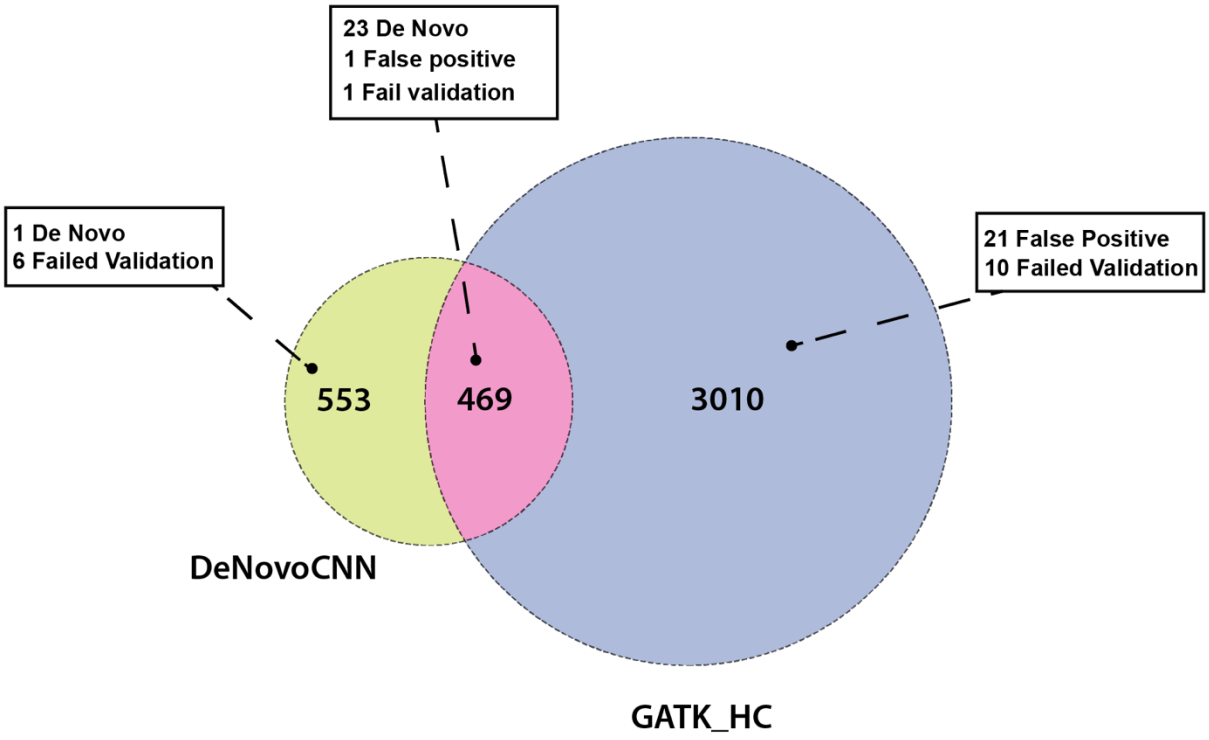


Figure 4. Venn diagram of results for the comparison of *de novo* mutation detection in WGS samples. Circle sizes are scaled to the number of identified variants. In red: variants called only by DeNovoCNN, in orange: variants called by both DeNovoCNN and high quality GATK, in green: variants called by only GATK with high quality filter.