

1 Training Genotype Callers with Neural 2 Networks

3 Remi Torracinta¹ and Fabien Campagne^{1,2,*}

4 ¹The HRH Prince Alwaleed Bin Talal Bin Abdulaziz Alsaud Institute for Computational
5 Biomedicine, Weill Cornell Medical College, New York, NY, United States of
6 America; Department of Physiology and Biophysics, Weill Cornell Medical College, New
7 York, NY, United States of America

8 *To whom correspondence should be addressed: fac2003@campagnelab.org

9 ABSTRACT

10 We present an open source software toolkit for training deep learning models to call genotypes in high-
11 throughput sequencing data. The software supports SAM, BAM, CRAM and Goby alignments and the
12 training of models for a variety of experimental assays and analysis protocols. We evaluate this software
13 in the Illumina Platinum whole genome datasets and find that a deep learning model trained on 80% of the
14 genome achieves a 0.986% accuracy on variants (genotype concordance) when trained with 10% of the
15 data from a genome. The software is distributed at [https://github.com/CampagneLaboratory/
16 variationanalysis](https://github.com/CampagneLaboratory/variationanalysis). The software makes it possible to train genotype calling models on consumer
17 hardware with CPUs or GPU(s). It will enable individual investigators and small laboratories to train and
18 evaluate their own models and to make open source contributions. We welcome contributions to extend
19 this early prototype or evaluate its performance on other gold standard datasets.

20 Keywords: Deep Learning, Machine Learning, Genotype Caller, High-Throughput Sequencing

21 INTRODUCTION

22 We recently presented an approach to call somatic variations with deep learning models, using no gold
23 standard variations (Torracinta et al. [2016], Campagne [2016]). In this study, we use high-throughput
24 sequencing data and available labels from gold-standard datasets to train deep-learning models that
25 can call genotypes. We built on the variationanalysis software we presented in Torracinta et al. [2016],
26 Campagne [2016] and distribute release 1.2 of the project, which now also supports training and using
27 genotyping models. In this preprint, we present the methods we used to train and evaluate genotyping
28 models, discuss how our approaches differ from the recently presented study of Pollin et al, and present
29 evaluation data on the Illumina platinum genome data.

30 The methods we present rely on deep learning models that are trained from data. These methods are
31 general and can be adapted to the characteristics of new experimental or analysis protocol by retraining
32 models with sequence data obtained with commercially available DNA samples Torracinta et al. [2016],
33 Campagne [2016], Poplin et al. [2016]. We present a brief protocol demonstrating how deep learning
34 models can be trained with the open-source variationanalysis project. To our knowledge, our study is the
35 first to offer an open-source implementation of a deep learning genotype caller and provide automated
36 protocols to train models for new assays. The methods developed for this caller support diploid as well as
37 polyploid organisms.

38 RESULTS

39 We developed a deep learning genotype caller. Briefly, the caller uses the Goby framework (Campagne et al.
40 [2013]) to observe characteristics of read alignments against a reference genome, and the variationanalysis
41 project (Torracinta et al. [2016], Campagne [2016]) to vectorize these characteristics into features and
42 labels suitable for training a feed forward neural network.

43 The caller can be trained using alignments in BAM, CRAM or Goby formats (Li et al. [2009], Fritz
44 et al. [2011], Campagne et al. [2013]) and associated labels. Labels necessary to train neural networks

45 are obtained from gold-standard datasets. Here, we used the Illumina Platinum Genomes (Eberle et al.
46 [2016]) as a source of alignments and true genotypes. In this first study, we called SNPs and disregarded
47 indels. All performance metrics are presented for SNPs.

48 We assembled a training dataset using all variants matching sites in the Platinum Genome NA12877
49 sample and 10% of other non-variant sites. This dataset was split in a training set (80% of sites),
50 validation set (10% of sites, further sub-sampled on non-variant sites) and test set (10% of sites). The
51 variationanalysis project provides tools to simplify assembling datasets (and make their production
52 consistent). See Material and Methods for a summary of the protocol and project documentation online
53 for details).

54 In this study, we mapped alignment data to 642 features for each site. Table 1 summarizes the
55 characteristics of these datasets.

Dataset	#Training Sites	#Validation Sites	#Test Sites	#Variants	#Features
PG NA12878 chr21 (dev)	923,680	4,776	114,972	11,200	642
PG NA12877 whole (test)	25,550,404	274,955	3,193,235	245,420	642

Table 1. Characteristics of Datasets

56 Initial feature mapper development was conducted on a development dataset composed exclusively
57 of chromosome 21 alignments from NA12878. This smaller dataset is independent of the final training
58 dataset since it was sequenced from a different individual. The purpose of feature mapper development is
59 to identify a mapping from alignment data to feature and label vectors that result in predictive models on
60 independent test datasets. We iteratively developed and tested about 15 mappers, identifying and fixing
61 software bugs through error analysis after each iteration. Error analysis consists in examining the types
62 of errors that the model makes (e.g., using a genome browser to visualize sites of prediction and the
63 alignment) on the test dataset. This process often suggests features that should be presented as input to
64 the network to facilitate learning. We stopped this process when performance seemed to reach a plateau
65 on the small development set, suggesting that we needed more data to train the model.

66 Table 2 shows the performance metrics obtained on the development set with a reasonably tuned
67 mapper. While these performance metrics are still far from the state of the art, they indicate that the
68 mappers do a reasonably good job of mapping alignments to vectors since a reliable model can be trained
69 with just one chromosome worth of data.

Dataset	Accuracy	Recall	Precision	F1	GC	AUC
chr21 dev	0.997	0.977	0.924	0.950	0.977	0.889

Table 2. Performance on Development Set. GC: Genotype Concordance. AUC: Area Under the
Received Operating Curve for correct variant identification. Precision and Recall: estimated over variants
only. F1: harmonic mean of Precision and Recall. Accuracy: estimated overall all genotypes.

70 The performance of deep neural networks are known to improve markedly when models are trained
71 with larger numbers of training examples. To determine the improvement that more training data would
72 bring, we trained a model with 10% of the data from the genome. Training was stopped when performance
73 measured on the validation set did not increase after 10 epochs (complete passes over the training set).
74 Table 3 shows the performance obtained when the model is trained with data from 10% of the genome
75 (excluding 90% of non-variants containing sites, but keeping all variant containing sites). Performance
76 metrics include all the gold-standard variants reported by the Platinum genome project (we did not limit
77 this analysis to variant sites that overlap confident regions, but used all variants).

78 These results show that more data can indeed markedly improve the performance of the trained
79 model. These results place the open-source caller that we present within less than 1% of genotype
80 concordance performance reported by Poplin and colleagues, while their models were training with data
81 from complete genomes. Importantly, the recall of the models is close to optimal (0.986), while precision
82 is still sub-optimal (0.936), suggesting that error analysis on the 10% dataset and future feature mapper
83 optimizations could quickly bring performance in a competitive range.

Accuracy	Recall	Precision	F1	numVariants	GC	AUC	[AUC95	AUC95]
0.994	0.986	0.936	0.961	225,941	0.986	0.908	0.904	0.911

Table 3. NA12877: Performance on the Test Set (10% of sites in the genome. The number of variants is the actual number of variants in the test set, used to estimate Precision, Recall and AUC. The AUC point estimate and 95% confidence intervals are estimated on a random sample of 50,000 of these sites.

84 MATERIALS AND METHODS

85 Alignment Data

86 Alignments were not pre-processed and were used directly after download from `ftp://ussd-ftp.illumina.com/2016-1.0/hg19/small_variants/`. Limited preprocessing was performed
87 with Goby to realign variations around indels to eliminate mis-alignment artifacts. Minimal pre-processing
88 is in contrast to Poplin et al. [2016] which used a haplotype realigner and several preprocessing steps that
89 already cleaned up the data before providing it to the model.
90

91 Source of Gold Standard Genotypes

92 We used all genotypes contained in the VCF files distributed at `ftp://ussd-ftp.illumina.com/2016-1.0/hg19/small_variants/`. In contrast to the study of Poplin et al. [2016], we did not
93 restrict the gold standard variations to regions of high-confidence because this only trains the models on
94 simpler regions of the genome and may limit its ability to discriminate variants in other regions.
95

96 Software implementation

97 Source Code

98 The source code of the software used for this study is distributed under the open-source Apache 2.0
99 license at `https://github.com/CampagneLaboratory/variationanalysis`.

100 Using trained models

101 Models trained in this study are being integrated into release 3.2+ of the Goby genotype caller (distributed
102 at `http://github.com/CampagneLaboratory/goby3`). Goby3 supports alignments in the
103 Goby, CRAM, SAM or BAM formats. A parameter is used to specify the path to a model to call
104 genotypes.

105 Training new models

106 Models for new assays can be trained by constructing a training, validation and test datasets. We
107 provide scripts to automate this activity. Detailed steps are documented with the software (see `https://github.com/CampagneLaboratory/variationanalysis`), but briefly, the datasets are
108 produced by converting alignments to .sbi files with the Goby SEQUENCE_BASE_INFORMATION output
109 format. True genotypes can be introduced in the .sbi file at this step. The .sbi file is randomized and
110 randomly split into training, validation and test sets. The training of the model uses the training and
111 validation sets for early stopping. Final model performance is estimated on the test set to verify that the
112 model generalizes. Models are saved to disk during training.
113

114 Neural Network Architecture

115 Feature Mappers

116 Feature mappers convert alignments about one sample into a fixed set of features suitable for train-
117 ing with neural networks. Regardless of the number of reads aligned at a genomic position, map-
118 pers need to produce a fixed-length output so that these outputs can be concatenated consistently
119 into a fixed-length input vector. At each genomic site, a mapper generates the number of reads sup-
120 porting each genotype (counts), the number of distinct locations in the read that support the geno-
121 type (distinct read indices). Hundreds of features are derived for each site and a complete list is
122 provided in the source code. Mappers are implemented in the variationanalysis project available at
123 GitHub `https://github.com/CampagneLaboratory/variationanalysis`. This study
124 used `org.campagnelab.dl.genotype.mappers.GenotypeMapperV13`.

125 **Model Architecture**

126 Models were developed with the DeepLearning4J (DL4J) framework (<http://deeplearning4j.org/>), ver-
127 sion 0.7.1. DL4J was selected because it is a Java framework and the models it produces can be
128 integrated with the Goby framework more easily than frameworks in other languages. Models were
129 formulated as 4 fully connected layers with RELU activation and a fully connected output layers with
130 soft-max activation. The number of output layers depends on the specific label mapper used. The
131 dense inner layers contain 5 times the number of input features. The exact model architecture used
132 is encoded in the class called `org.campagnelab.dl.genotype.learning.architecture`
133 `.graphs.CombinedWithIsVariantGenotypeAssembler` distributed in the variationanalysis
134 project Torracinta and Campagne [2016].

135 **Label Mapper**

136 We have experimented with different methods to map genotypes to label vectors. One method calls
137 alleles individually, and encodes the number of alleles. (Implemented with `org.campagnelab.dl`
138 `.genotype.mappers.NumDistinctAllelesLabelMapper` and `10 org.campagnelab.dl`
139 `.genotype.mappers.GenotypeLabelsMapper`). This method is suitable for genomes of arbi-
140 trary ploidy (e.g., plants). Another method is similar to that described in Poplin et al. [2016] and is limited
141 to diploid genomes.

142 **Early Stopping**

143 We trained models with early stopping. Briefly, performance of the model was measured on a validation
144 set and training was stopped when performance on the validation set did not increase for 10 epochs. We
145 used the harmonic mean of F1 and AUC as validation performance measure.

146 **Analysis Protocol Summary**

147 A summary of a typical analysis protocol is provided here. Running these steps requires defining some
148 environment variables and is fully explained in the software documentation. Training a model is performed
149 in two high-level steps:

- 150 • Transform an alignment (BAM, CRAM or Goby format) into an .sbi file (input for feature and label
151 mappers):

```
152 parallel-genotype-sbi.sh 10g NA12878_S1.bam
```

153 This step produces the files NA12878_S1-train.sbi and NA12878_S1-train.sbp and two pairs of
154 files, one for validation (used for early stopping) and for test set.

- 155 • Train and evaluate the model with a choice of feature mapper (the number 1 is the index of the
156 GPU to train the model on):

```
157 iterate-genotype.sh org.campagnelab.dl.genotype.\  
158 mappers.GenotypeMapperV13 1
```

159 This produces a model trained with the mapper on the training set and performance statistics on the
160 validation set (printed during training, also stored in the model directory for reference), and finally
161 runs the model on the test set, printing and storing statistics.

162 **DISCUSSION**

163 We presented a novel approach to call genotypes in high-throughput sequencing data using neural networks.
164 The approach used in this study relies on deep neural networks to call genotypes and can be trained from
165 gold-standard data. It differs from previous approaches in the following ways.

166 Training deep learning models to call genotypes is a straightforward adaption of some of the ideas that
167 we presented in Torracinta et al. [2016], Campagne [2016]. Given the existence of gold standard data for
168 genotype calls and the ability of deep neural networks to reliably estimate probabilities, the development
169 of a deep learning caller is a logical step. Our method differs from published genotype callers (e.g.,
170 GATK) which rely on carefully designed probabilistic models McKenna et al. [2010], Nielsen et al. [2011].

171 Our approach is to train probabilistic models from data. The key advantage is that new models can be
172 trained and adapted quickly to new experimental assays or data analysis protocols (e.g., combination of
173 aligners and read preprocessing techniques).

174 Poplin and colleagues have recently demonstrated that a similar idea performs extremely well across
175 a variety of datasets (Poplin et al. [2016]). In this work, these authors have leveraged the expertise of
176 Google in deep learning for images. They converted alignment data to images and trained models with
177 the Inception v2 architecture. Their work is a clear demonstration that deep learning can be used for
178 genotype calling and can result in state of the art genotype calling performance (as measured by genotype
179 concordance, and precision/recall or F1 for the identification of variants).

180 The approach of Poplin et al has several drawbacks. First, as presented, it only supports calling
181 genotypes for diploid organisms (the network predicts three states for the genotype: AA A/B or BB).
182 A universal caller should also be applicable to plants and other non diploid organisms. Our approach
183 supports and already implements label mappers that can be used with arbitrary ploidy. The computational
184 procedure described in Poplin et al. [2016] has another important drawback: its computational efficiency.
185 Converting alignments to images requires assembling a 10,000 by 300 pixel image for each site where a
186 prediction is required. The DeepVariant model therefore uses at least 3 million pixels per site. In contrast,
187 our approach uses less than a 1,000 floats to represent both features and labels. Since our approach uses
188 orders of magnitude fewer features, the training and evaluation datasets can be stored in a few tens of
189 gigabytes and model training can be conducted on a workstation with one or a few GPU cards (we trained
190 and evaluated models for this study on a workstation costing less than \$10K). Computational efficiency is
191 important to allow individual researchers to replicate results, make and evaluate method improvements,
192 and develop models for new experimental assays or analysis protocols.

193 Our methods also differ from Poplin et al. [2016] in the amount of preprocessing applied to alignment
194 data before it is provided to the neural network. We applied limited preprocessing (realignment around
195 indels), when Poplin et al used the GATK haplotype caller, which implements local reassembly. We
196 believe that many of the pre-processing/clean-up operations currently implemented by ad-hoc software
197 can be trained by back propagation given suitable training data. The fact that the test performance of the
198 models we trained with minimal preprocessing are closing in on state of the art performance suggests that
199 this hypothesis has merit.

200 Another point of difference between our study and the work of Poplin et al. [2016] is that we distribute
201 the software we developed under an open-source license. The immediate availability of the software
202 and detailed model training protocols will make it possible for other researchers to train models for new
203 platforms as well as to contribute to method development and evaluation.

204 Finally, we are looking for collaborators interested in helping develop and evaluate improved versions
205 of this genotype caller on a wide range of platforms. Training new models will be required when data
206 from new platforms becomes available and for this reason we feel that a community effort is best suited to
207 efficiently developing these technologies.

208 The software that we presented here (see also Torracinta et al. [2016], Campagne [2016]) provides a
209 test-bed infrastructure where new ideas can be tested and evaluated quickly. We hope that it will enable a
210 community of researchers to experiment with neural networks for genomic applications.

211 **ACKNOWLEDGMENTS**

212 We thank Manuele Simi for technical assistance with Maven configurations for this project.

213 **FUNDING**

214 This investigation was supported by the National Institutes of Health NIAID award 5R01AI107762 to
215 Fabien Campagne and Maureen Hanson. This investigation was also supported by the STARR cancer
216 consortium award I9-A9-084 to Samie Jaffrey, Jedd Wolchok and Fabien Campagne.

217 **REFERENCES**

218 Fabien Campagne. <http://dx.doi.org/10.1101/079087> continuation: Evaluation of adaptive somatic models
219 in a gold standard whole genome somatic dataset. *bioRxiv*, 2016. doi: 10.1101/093534. URL
220 <http://biorxiv.org/content/early/2016/12/13/093534>.

- 221 Fabien Campagne, Kevin C. Dorff, Nyasha Chambwe, James T. Robinson, and Jill P. Mesirov. Com-
222 pression of Structured High-Throughput Sequencing Data. *PLoS ONE*, 8(11):e79871, nov 2013.
223 ISSN 1932-6203. doi: 10.1371/journal.pone.0079871. URL [http://dx.plos.org/10.1371/
224 journal.pone.0079871](http://dx.plos.org/10.1371/journal.pone.0079871).
- 225 Michael A Eberle, Epameinondas Fritzilas, Peter Krusche, Morten Källberg, Benjamin L Moore,
226 Mitchell A Bekritsky, Zamin Iqbal, Han-Yu Chuang, Sean J Humphray, Aaron L Halpern, et al.
227 A reference data set of 5.4 million phased human variants validated by genetic inheritance from
228 sequencing a three-generation 17-member pedigree. *Genome Research*, 2016.
- 229 Markus Hsi-Yang Fritz, Rasko Leinonen, Guy Cochrane, and Ewan Birney. Efficient storage of high
230 throughput dna sequencing data using reference-based compression. *Genome research*, 21(5):734–740,
231 2011.
- 232 Heng Li, Bob Handsaker, Alec Wysoker, Tim Fennell, Jue Ruan, Nils Homer, Gabor Marth, Goncalo
233 Abecasis, Richard Durbin, et al. The sequence alignment/map format and samtools. *Bioinformatics*, 25
234 (16):2078–2079, 2009.
- 235 Aaron McKenna, Matthew Hanna, Eric Banks, Andrey Sivachenko, Kristian Cibulskis, Andrew Kernytzky,
236 Kiran Garimella, David Altshuler, Stacey Gabriel, Mark Daly, et al. The genome analysis toolkit: a
237 mapreduce framework for analyzing next-generation dna sequencing data. *Genome research*, 20(9):
238 1297–1303, 2010.
- 239 Rasmus Nielsen, Joshua S Paul, Anders Albrechtsen, and Yun S Song. Genotype and snp calling from
240 next-generation sequencing data. *Nature Reviews Genetics*, 12(6):443–451, 2011.
- 241 Ryan Poplin, Dan Newburger, Jojo Dijamco, Nam Nguyen, Dion Loy, Sam S. Gross, Cory Y. McLean,
242 and Mark A. DePristo. Creating a universal snp and small indel variant caller with deep neural networks.
243 *bioRxiv*, 2016. doi: 10.1101/092890. URL [http://biorxiv.org/content/early/2016/
244 12/21/092890](http://biorxiv.org/content/early/2016/12/21/092890).
- 245 Remi Torracinta and Fabien Campagne. Variationanalysis 1.0.2 software release, October 2016. URL
246 <https://doi.org/10.5281/zenodo.159203>.
- 247 Remi Torracinta, Laurent Mesnard, Susan Levine, Rita Shaknovich, Maureen Hanson, and Fabien
248 Campagne. Adaptive somatic mutations calls with deep learning and semi-simulated data. *bioRxiv*,
249 2016. doi: 10.1101/079087. URL [http://biorxiv.org/content/early/2016/10/04/
250 079087](http://biorxiv.org/content/early/2016/10/04/079087).