MinCall — MinION end2end deep learning basecaller

Neven Miculinić, Marko Ratković, and Mile Sikić
{neven.miculinic, marko.ratkovic, mile.sikic}@fer.hr

Faculty of Electrical Engineering and Computing (FER), Zagreb, Croatia

Abstract. The Oxford Nanopore Technologies's MinION is the first portable DNA sequencing device. It's capable of producing long reads, over 100 kBp were reported, however, is has significantly higher error rate than other methods.
In this study, we created MinCall, an end2end basecaller model for the MinION. The model is based on deep learning and uses convolutional neural networks (CNN) in its implementation. For extra performances is uses cutting edge deep learning techniques and architectures, batch normalization and Connectionist Temporal Classification (CTC) loss. The best performing 270 layers deep model achieves state-of-the-art 90.5% match rate on E.Coli dataset using R9 pore chemistry and 1D reads.

Keywords: Basecaller, MinION, R9, CNN, CTC, Next generation sequecing

1 Introduction

In recent years, deep learning methods significantly improved the state-of-the-art in multiple domains such as computer vision, speech recognition, and natural language processing [17] [15]. In this paper, we present application of deep learning in the field of Bioinformatics for DNA basecalling problem.

Oxford Nanopore Technology's MinION nanopore sequencing platform [22] is the first portable DNA sequencing device. It's small weight, of only 90 grams, low capital cost, and long read length combined with high-throughput, real-time data analysis, and decent accuracy yield promising results in various applications. From clinical application such as monitoring infectious disease outbreaks [13] [24], characterizing structural variants in cancer [23] and even full human genome assembly [12].

Although MinION is able to produce long reads [20, 21], they have a high sequencing error rate. This has been somewhat alleviated with new R9 pore model, replacing older R7 ones. In this paper, we show that this error rate can be reduced by replacing the default base caller provided by the manufacturer with a properly trained neural network model. In the future new R9.5 chemistry and 1D$\hat{2}$ reads should supersede current models.

In the MinION device, single-stranded DNA fragments move through nanopores, which causes drops in the electric current. The electric current is measured at each pore several thousand times per second, 4000 times exactly in our dataset. The electric current depends mostly on the context of several DNA bases passing through the pore at the time of measurement. As the DNA moves through the pore, the context shifts, and the electric current changes.

The MinION device typically yields reads several thousands bases long, even couple hundred thousand bases long reads were repoted [20, 21]. However the cost in on accuracy, signicifantly lower than older, more reliable and expensive sequencing methods.

The exact error rate metric is unreliable since multiple pipeline tools could be the issue. First the sample is prepared, hopefully, uncontaminated and matching reference genome as close as possible, then sequenced using the MinION device obtaining raw data. Next, our model (or other

groups ones) come along, basecall the sequence. To evaluate error rate metric basecalled read is aligned to the reference genome using aligners with their own errors/biases, mostly commonly used BWA-MEM [18] and Graphmap [27].

2 Sequencing overview

Conceptually, the MinION sequencer works as follows. First, DNA is sheared into smaller DNA fragments and adapters are ligated to either end of the fragments. The resulting DNA fragments pass through a protein embedded in a membrane via a nanometre-sized channel, a nanopore. A single DNA strand passes through the pore. Optionally, hairpin protein adapter can merge two DNA strands, allowing both template and complement read passing through the nanopore sequentially for more accurate reads. This technique is referred as 2D reads, while we focus on 1D reads containing only template DNA and no hairpin adapter.

Electrical current runs through the nanopore and exact nucleotides context within influences the nanopore's resistance. This resistive effect is our sensor data, that is the current fluctuations as DNA passes though the pore. The nanopore is 6 nucleotides wide, and many models use this information in their approaches, while we're opted out of this technicality.

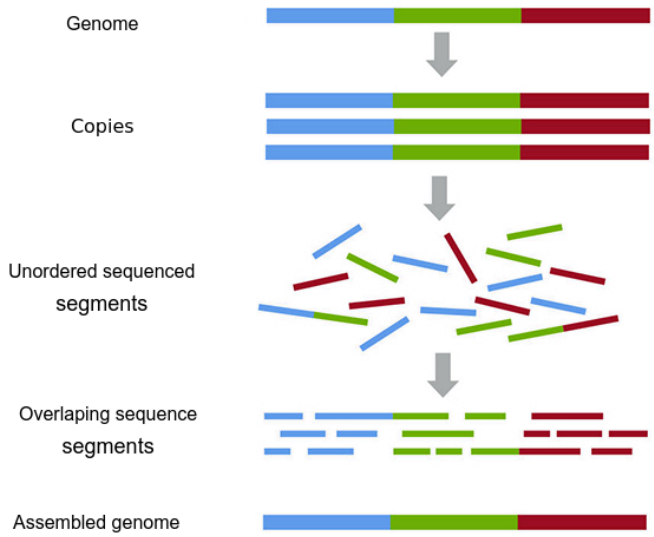

Fig. 1. Depiction of shotgun sequencing

3 Basecalling

The core of the decoding process is the basecalling step. Nowdays there's multiple basecalling options, what official and unoffical ones.

[1] Figure adapted from https://nanoporetech.com/how-it-works

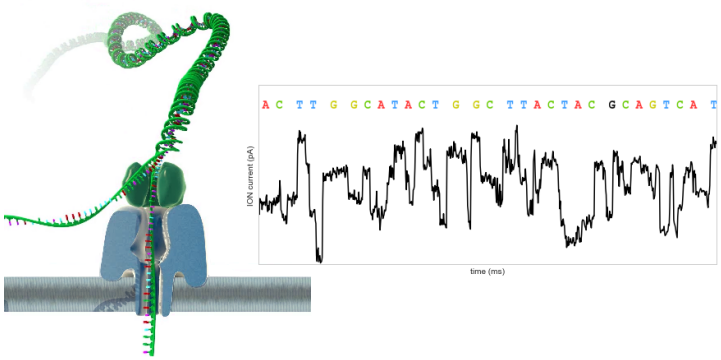

Fig. 2. DNA strain being pulled through a nanopore [1]

Earlier models were Hidden Markov model(HMM)-based where hidden state modeled DNA sequence of length 6 (6-mer) in the nanopore. Pore models were used in computing emission probabilities. [19, 25, 29, 30] and the recent open source HMM-based basecaller Nanocall [5]. Modern basecallers use RNN base models, and we opted out using CNN instead with beam search.

We compared our model on R9 chemistry with Metrichorn (HMM based approach) and Albacore (RNN based approach). For detailed basecaller overview see the appendix A.

4 Dataset

Dataset used were E.Coli K-12 strands from [20] and Lamba basecalling[2]. Both used datasets show in table 1 have been previously have passed through MinKNOW and had been basecalled by Metrichor. As 1D read analysis was the focus of this paper, only those reads were used.

Table 1. Used datasets

	Number of reads	Total bases [bp][3]	Whole genome size [bp]
E. Coli[4]	164471	1 481 687 490	4 639 675
lambda[5]	86	466 465	48 502

4.1 Data preprocessing

To help training process, the raw signal is split into smaller blocks that are used as inputs. For each Metrichor basecalled event is easy to determine the block it falls into using *start* field.

[2] Acquired from doc. dr. sc. Petra Korać i dr.sc. Paula Dobrinić

[3] Total number of bases calle by Metrichor

[4] R9 sequencing data from `http://lab.loman.net/2016/07/30/nanopore-r9-data-release/`, reference taken from `https://www.ncbi.nlm.nih.gov/nuccore/48994873`

[5] Internal dataset acquired from doc. dr.sc. Petra Korać and dr.sc. Paula Dobrinić, reference taken from `https://www.ncbi.nlm.nih.gov/nuccore/NC_001416.1`

Using this information output given by Metrichor can be determined for each block. To correct errors produced by Metrichor and possibly increase the quality of data, each read is aligned to the reference. This is done using aligner GraphMap [27] that returns the best position in the genome, hopefully, the part of the genome from which read came from. Alignment part in the genome is used as a target. Using CIGAR string returned by aligner we can correct Metrichor data and get target output for each block. This process is shown in figure 3.

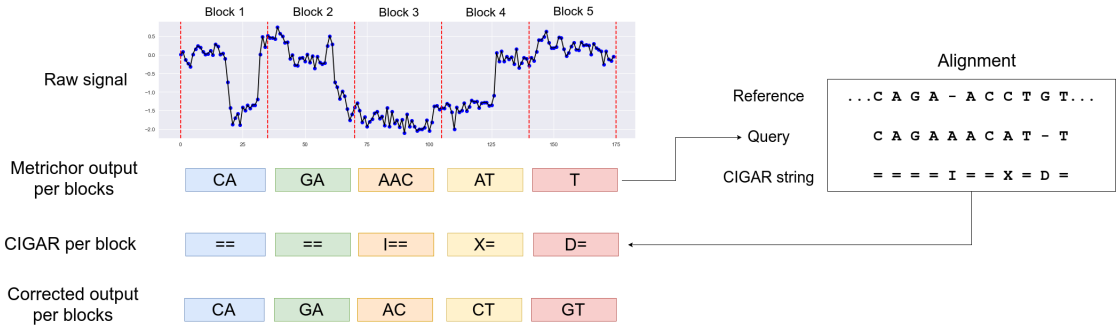

Fig. 3. Dataset preparation

To eliminate the possibility of overfitting to the known reference, the model is trained and tested on reads from different organisms. Due to limited amount of public available raw nanopore sequence data, ecoli was *divided* into two regions. Reads were split into train and test portions, depending on which region of ecoli they align. If read aligns inside first 70% of the ecoli, it is placed into train set, and if it aligns to the second portion, it is placed into test set. Reads whose alignment overlaps train and test region are not used. Important to note that ecoli genome, and genomes of the majority of other bacteria, is cyclical, so reads with alignments that wrap over edges are also discarded. Total train set consist of over 110 thousand reads. Overview of the entire learning pipeline is shown in figure 4.

Due to CTC merged nature during decoding, that is in best matched path adjacent duplicates are merged into one, we preprocess the target nucleotide sequence with surrogate nucleotides, such that each second repeated nucleotide is its surrogate. Example provided in figure 5. All raw input data were normalized to zero mean and unit variance as it yield superiour perfomance with neural networks.

5 Method

Instead of opting for the traditional path using HMM or newly adopted RNN we tried using CNN (Convolutional neural networks) [16], that is their residual version [9]. For loss, we used CTC (Connectionist temporal classification) [7] between basecalled and the target sequence. Other building blocks used are Batch normalization(BN) [11] and pooling layers. No dropout [28] were used. We used open source warp-ctc [2] GPU CTC loss implementaion and tensorflow [1].

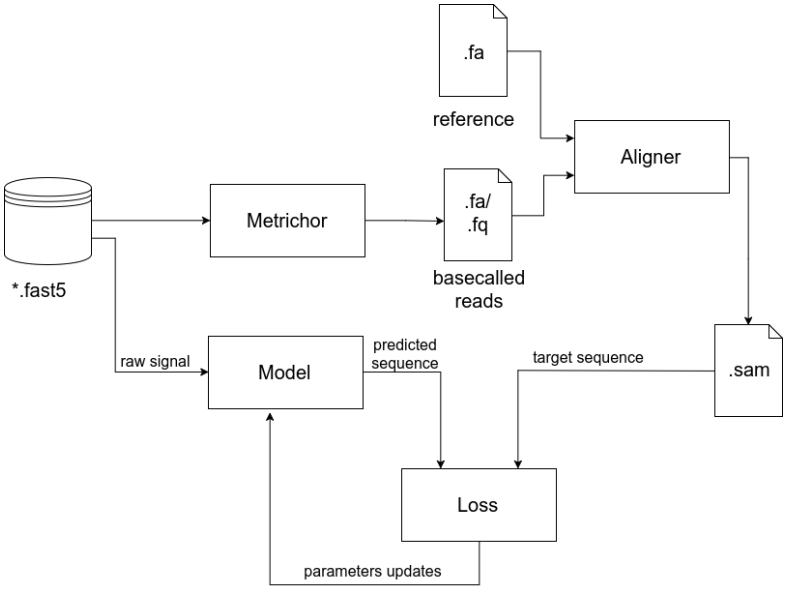

Fig. 4. Training pipeline overview

```
Target       :  A  G  A  A  A
Preprocessed:  A  G  A  A' A
```

Fig. 5. Target nucleotide sequence preprocessing

The final model is a residual neural network consisting of 72 residual blocks BN[6]-ELU[7]-CONV[8]-BN-ELU-CONV, to grand total of 2 million parameters. The used model is a variant of architecture proposed in paper [10] with the difference of ELU being used as activation instead of ReLU as it is reported [26] to speeds up the learning process and improve accuracy as the depth increase.

Each convolutional layer in this models uses 64 chanells with receptive field of 3. Because sequenced read is always shorter than the raw signal, pooling with kernel size two is used every 24 layers resulting in a reduction of dimensionality by factor 8. This is beneficial in faster learning, better generalization and increased basecalling speed.

Training the model is the minimization of previously described CTC loss. It was done using Adam [14] with default parameters, and exponentially decaying learning rate starting from 1e-3 and decay rate of 5e-6 over 100k steps[9] and minibatch size 8. To prevent gradients exploding on bad inputs, they were clipped to a range [-2, 2]. We observed no overfitting due to large dataset size.

[6] Batch normalization
[7] Exponential Linear Unit

$$\mathrm{ELU}(x) = \begin{cases} x, & \text{if } x > 0 \\ \alpha(exp(x) - 1), & \text{otherwise} \end{cases}$$

[8] 1D convolutional layer
[9] We use tf.train.exponential_decay where current learning rate, lr is $lr = \text{initial\_lr} \cdot \text{decay\_rate}^{\frac{\text{global\_step}}{\text{decay\_step}}}$

During testing we tried ReLu and PrELU [8] with no significat result difference. We also tried different channel numbers, receptive field width and various other hyperparameters during hyperparameter optimization and conclude after enough complexity, that is sufficient layers, all choices were performing similarly.

6 Results

Developed tool was compared with other available basecallers that support R9 chemistry. This includes third-party basically DeepNano and official basecallers by Oxford Nanopore (cloud-based Metrichor and Nanonet). The fact that ground truth is not known makes evaluation difficult. Different methods for evaluation were used to get clearer information about each basecaller.

A portion of the read length that aligns as correctly is called match_rate. Same goes for mismatches and insertions. Sum of all matches, mismatches, and insertions is equal to the reads length 1. For specific details see appendix B.1. Results on E.Coli test set with Graphmap aligner are shown in table 2. Evaluation with BWM-MEM and on lambda can be found in appendix B.1 under tables 3, 4 and 5. Furthermore we plot Kernel Density estimation(KDE) plots for each mentioned statistic on E.Coli dataset in figure 6.

Table 2. Alignment specifications of Ecoli R9 basecalled reads using GraphMap

	Match % (median)	Mismatch % (median)	Insertion % (median)	Deletion % (median)
DeepNano	90.254762	6.452852	**3.274420**	11.829965
Metrichor	90.560455	5.688105	3.660381	8.328271
Nanonet	90.607674	5.608912	3.652791	8.299046
MinCall	**91.408591**	**5.019141**	3.477739	**7.471608**

7 Conclusion and further work

This model used advance state-of-the-art gated residual convolutional neural network, with top models having 270 layers and over 3M parameters, yet improvements over Metrichorn baseline are marginal. As the conclusion, it might be that we've reached Bayesian error rate for R9 chemistry. Furthermore, R9.5 and 1D$\hat{2}$ reads are under development which shall yield this paper's result obsolete quite soon, yet underlying code developed could easily be adjusted and trained on new data.

Unlike Nanonet which uses custom OpenCL kernels or Albacore — a novel ONT basecaller as of May 2017 lacking GPU support, this work used world-class computational framework tensorflow with highly optimized kernels and large development community. Therefore resulting paper's effect is showcasing Residual CNN approach or pure CNN approach with CTC loss is marginally better than already established basecaller and providing code in the contemporary framework.

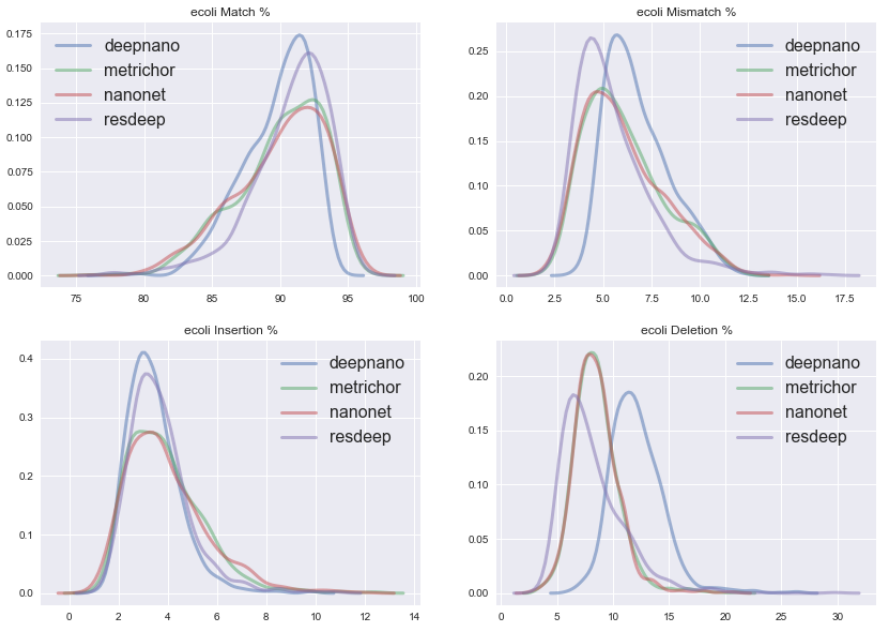

Fig. 6. KDE plot for distribution of percentage of alimnment operations for ecoli

8 Acknowledgments

First and foremost I'd like to thank my mentor izv. prof. dr. sc. Mile Sikić for setting up the problem, guiding us through its beginnings and providing helpful advice for its completion. Next Marko Ratkovic for his not only insightful conversations, but also meaningful code contributions.

Finally, I'm thanking various other people whose code, tools and advice I've used in completing this paper: Fran Jurišić, Ana Marija Selak, Ivan Sović and Martin Šošić.

Also some data were obtained in cooperation with Biologists doc. dr. sc. Petra Korać and dr.sc. Paula Dobrinić. Other were procured publicly shared from Loman labs [20]

References

1. Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G.S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., Zheng, X.: TensorFlow: Large-scale machine learning on heterogeneous systems (2015), `http://tensorflow.org/`, software available from tensorflow.org
2. Amodei, D., Anubhai, R., Battenberg, E., Case, C., Casper, J., Catanzaro, B., Chen, J., Chrzanowski, M., Coates, A., Diamos, G., Elsen, E., Engel, J., Fan, L., Fougner, C., Han, T., Hannun, A.Y., Jun, B., LeGresley, P., Lin, L., Narang, S., Ng, A.Y., Ozair, S., Prenger, R., Raiman, J., Satheesh, S., Seetapun, D., Sengupta, S., Wang, Y., Wang, Z., Wang, C., Xiao, B., Yogatama, D., Zhan, J., Zhu, Z.: Deep speech 2: End-to-end speech recognition in english and mandarin. CoRR abs/1512.02595 (2015), `http://arxiv.org/abs/1512.02595`

3. Boža, V., Brejová, B., Vinař, T.: DeepNano: Deep recurrent neural networks for base calling in MinION nanopore reads. PLOS ONE 12(6), e0178751 (jun 2017), https://doi.org/10.1371/journal.pone.0178751
4. Community, N.: Basecalling overview, https://community.nanoporetech.com/technical_documents/data-analysis/v/datd_5000_v1_reve_22aug2016/basecalling-overvi, [Accessed; 12-July-2017]
5. David, M., Dursi, L.J., Yao, D., Boutros, P.C., Simpson, J.T.: Nanocall: an open source basecaller for oxford nanopore sequencing data. Bioinformatics p. btw569 (2016)
6. David, M., Dursi, L.J., Yao, D., Boutros, P.C., Simpson, J.T.: Nanocall: An open source basecaller for oxford nanopore sequencing data. bioRxiv (2016), http://biorxiv.org/content/early/2016/03/28/046086
7. Graves, A., Fernández, S., Gomez, F., Schmidhuber, J.: Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In: Proceedings of the 23rd international conference on Machine learning. pp. 369–376. ACM (2006)
8. He, K., Zhang, X., Ren, S., Sun, J.: Delving deep into rectifiers: Surpassing human-level performance on imagenet classification (2015)
9. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 770–778 (2016)
10. He, K., Zhang, X., Ren, S., Sun, J.: Identity mappings in deep residual networks (2016)
11. Ioffe, S., Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift (2015)
12. Jain, M., Koren, S., Quick, J., Rand, A.C., Sasani, T.A., Tyson, J.R., Beggs, A.D., Dilthey, A.T., Fiddes, I.T., Malla, S., et al.: Nanopore sequencing and assembly of a human genome with ultra-long reads. bioRxiv p. 128835 (2017)
13. Judge, K., Harris, S.R., Reuter, S., Parkhill, J., Peacock, S.J.: Early insights into the potential of the oxford nanopore minion for the detection of antimicrobial resistance genes. Journal of Antimicrobial Chemotherapy 70(10), 2775–2778 (2015)
14. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization (2014)
15. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Pereira, F., Burges, C.J.C., Bottou, L., Weinberger, K.Q. (eds.) Advances in Neural Information Processing Systems 25, pp. 1097–1105. Curran Associates, Inc. (2012), https://goo.gl/UpFBv8
16. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proceedings of the IEEE 86(11), 2278–2324 (November 1998)
17. LeCun, Y., Bengio, Y.: The handbook of brain theory and neural networks. chap. Convolutional Networks for Images, Speech, and Time Series, pp. 255–258. MIT Press, Cambridge, MA, USA (1998), http://dl.acm.org/citation.cfm?id=303568.303704
18. Li, H.: Aligning sequence reads, clone sequences and assembly contigs with bwa-mem. arXiv preprint arXiv:1303.3997 (2013)
19. Loman, N.J., Quick, J., Simpson, J.T.: A complete bacterial genome assembled de novo using only nanopore sequencing data. Nature methods 12(8), 733–735 (2015)
20. Loman, N.: Nanopore R9 rapid run data release, http://lab.loman.net/2016/07/30/nanopore-r9-data-release/, [Online; posted 30-July-2016]
21. Loman, N.: Thar she blows! Ultra long read method for nanopore sequencing, http://lab.loman.net/2017/03/09/ultrareads-for-nanopore/, [Online; posted 9-March-2017]
22. Mikheyev, A.S., Tin, M.M.: A first look at the oxford nanopore minion sequencer. Molecular ecology resources 14(6), 1097–1102 (2014)
23. Norris, A.L., Workman, R.E., Fan, Y., Eshleman, J.R., Timp, W.: Nanopore sequencing detects structural variants in cancer. Cancer biology & therapy 17(3), 246–253 (2016)
24. Quick, J., Loman, N.J., Duraffour, S., Simpson, J.T., Severi, E., Cowley, L., Bore, J.A., Koundouno, R., Dudas, G., Mikhail, A., et al.: Real-time, portable genome sequencing for ebola surveillance. Nature 530(7589), 228–232 (2016)
25. Schreiber, J., Karplus, K.: Analysis of nanopore data using hidden markov models. Bioinformatics p. btv046 (2015)

26. Shah, A., Kadam, E., Shah, H., Shinde, S., Shingade, S.: Deep residual networks with exponential linear unit (2016)
27. Sović, I., Šikić, M., Wilm, A., Fenlon, S.N., Chen, S., Nagarajan, N.: Fast and sensitive mapping of nanopore sequencing reads with graphmap. Nature communications 7 (2016)
28. Srivastava, N., Hinton, G.E., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: a simple way to prevent neural networks from overfitting. Journal of Machine Learning Research 15(1), 1929–1958 (2014)
29. Szalay, T., Golovchenko, J.A.: De novo sequencing and variant calling with nanopores using poreseq. Nature biotechnology 33(10), 1087–1091 (2015)
30. Timp, W., Comer, J., Aksimentiev, A.: Dna base-calling from a nanopore using a viterbi algorithm. Biophysical journal 102(10), L37–L39 (2012)

A Basecallers

Here is currated basecaller list:

A.1 Official

Metrichor is an Oxford Nanopore company that offers cloud-based platform *EPI2ME* for analysis of nanopore data. Initially, base calling was only available by uploading data to the platform - that being the reason why this basecaller is often called Metrichor even though it is a name of the company.

With the release of R9 chemistry, this model was replaced by a more accurate recurrent neural network (RNN) implementation. Currently, Oxford Nanopore offers several RNN-based local basecaller versions under different names: Albacore, Nanonet and basecaller integrated into MinKNOW [4].

Albacore is basecaller by Oxford Nanopore Technologies ready for production and actively supported. It is available to the Nanopore Community served as a binary. The source code of Albacore was not provided and is only available through the ONT Developer Channel. Tool supports only R9.4 and future R9.5 version of the chemistry. For R9 tests in this paper we used R9.4 chemisty setting as instructed on ONT forums.

Nanonet[10] uses the same neural network that is used in Albacore but it is continually under development and does contain features such as error handling or logging needed for production use. It uses *CURRENNT* library for running neural networks. It supportes basecalling of both R9 and R9.4 chemistry versions. However in our experiments it was painfully slow, which was as expected due to it's classification as not production ready.

Scrappie[11] is another basecaller by Oxford Nanopore Technologies. Similar to Nanonet, it is the platform for ongoing development. Scrappie is reported to be the first basecaller that specifically address homopolymer base calling. It became publicly available just recently in June, 2017 and supports R9.4 and future R9.5 data.

A.2 Third-party basecallers

Nanocall [6] was the first third-party open source basecaller for nanopore data. It uses HMM approach like the original R7 Metrichor. Nanocall does not support newer chemistries after R7.3.

DeepNano [3] was the first open-source basecaller based on neural networks. It uses bidirectional recurrent neural networks implemented in Python, using the Theano library. When released, originally only supported R7 chemistry, but support for R9 and R9.4 was added recently.

[10] `https://github.com/nanoporetech/nanonet/`
[11] `https://github.com/nanoporetech/scrappie`

B Evaluation metrics

B.1 CIGAR derived metrics

A portion of the read length that aligns as correctly is called match_rate. Same goes for mismatches and insertions. Sum of all matches, mismatches, and insertions is equal to the reads length 1. Here is the specific equation list:

$$read\_len = n\_matches + n\_mismatches + n\_insertions \tag{1}$$

$$match\_rate = \frac{n\_matches}{read\_length} \tag{2}$$

$$missmatch\_rate = \frac{n\_mismatches}{read\_length} \tag{3}$$

$$insertion\_rate = \frac{n\_insertions}{read\_length} \tag{4}$$

$$match\_rate + snp\_rate + insertion\_rate = 1 \tag{5}$$

Deletion rate is defined as a total number of deletions in the alignment over the length of the aligned read.

$$deletion\_rate = \frac{n\_deletion}{read\_length} \tag{6}$$

Table 3. Alignment specifications of Ecoli R9 basecalled reads using BWA mem

	Match % (median)	Mismatch % (median)	Insertion % (median)	Deletion % (median)
DeepNano	90.254762	6.452852	3.274420	11.829965
Metrichor	90.595441	6.869543	2.531646	7.567381
Nanonet	90.988989	6.674760	**2.348552**	7.698530
MinCall	**91.470588**	**5.929204**	2.477283	**6.970362**

Table 4. Alignment specifications of Lambda basecalled reads using GraphMap

	Match % (median)	Mismatch % (median)	Insertion % (median)	Deletion % (median)
DeepNano	86.997687	9.623494	3.442490	16.052830
Metrichor	87.714988	7.835052	4.093851	**10.757491**
Nanonet	88.415611	8.178372	3.629653	11.793022
MinCall	**89.694482**	**7.238095**	**3.078796**	13.450292

Table 5. Alignment specifications of lambda R9 basecalled reads using BWA mem

	Match % (median)	Mismatch % (median)	Insertion % (median)	Deletion % (median)
DeepNano	86.625973	11.288361	2.098225	14.648308
Metrichor	87.294093	10.109186	2.376476	**9.645323**
Nanonet	87.767037	10.017598	2.354248	10.597232
MinCall	**89.049870**	**9.480883**	**1.615188**	12.962441