Available at https://jtsiskom.undip.ac.id (3 July 2020)
DOI:10.14710/jtsiskom.2020.13407

Jurnal Teknologi dan Sistem Komputer, 8(3), 2020, 234-238

Model *deep learning* untuk klasifikasi fragmen metagenom dengan spaced k-mers sebagai ekstraksi fitur

Deep learning model for metagenome fragment classification using spaced k-mers feature extraction

Nur Choiriyati[1)], Yandra Arkeman[2)], Wisnu Ananta Kusuma[1*)]

[1)]*Departemen Ilmu Komputer, Fakultas Matematika dan Ilmu Pengetahuan Alam, Institut Pertanian Bogor*
Jl. Meranti, Kampus IPB Dramaga, Babakan, Indonesia 16680

[2)]*Departemen Teknologi Industri Pertanian, Fakultas Teknologi Pertanian, Institut Pertanian Bogor*
Jl. Meranti, Kampus IPB Dramaga, Babakan, Indonesia 16680

Cara sitasi: N. Choiriyati, Y. Arkeman, and W. A. Kusuma, "Model deep learning untuk klasifikasi fragmen metagenom dengan spaced k-mers sebagai ekstraksi fitur," *Jurnal Teknologi dan Sistem Komputer*, vol. 8, no. 3, pp. 234-238, 2020. doi: 10.14710/jtsiskom.2020.13407, [Online].

Abstract - An open challenge in bioinformatics is the analysis of the sequenced metagenomes from the various environments. Several studies demonstrated bacteria classification at the genus level using k-mers as feature extraction where the highest value of k gives better accuracy but it is costly in terms of computational resources and computational time. Spaced k-mers method was used to extract the feature of the sequence using 111 1111 10001 where 1 was a match and 0 was the condition that could be a match or did not match. Currently, deep learning provides the best solutions to many problems in image recognition, speech recognition, and natural language processing. In this research, two different deep learning architectures, namely Deep Neural Network (DNN) and Convolutional Neural Network (CNN), trained to approach the taxonomic classification of metagenome data and spaced k-mers method for feature extraction. The result showed the DNN classifier reached 90.89 % and the CNN classifier reached 88.89 % accuracy at the genus level taxonomy.

Keywords *– classification; deep learning; metagenomes; spaced k-mers*

Abstrak – Tantangan dalam analisis dunia bioinformatika adalah analisis sekuens metagenom yang diambil dari berbagai lingkungan. Proses binning pada sampel metagenom dapat dilakukan dengan menghitung frekuensi kemunculan k-mers dari suatu sekuens metagenom. Ekstraksi fitur spaced k-mers dilakukan dengan membandingkan fragmen metagenom dengan substring berukuran k (k-mers), namun mengijinkan kondisi inexact matching (don't care position). Deep Learning dapat memberikan solusi terbaik untuk banyak masalah dalam pengenalan pola. Penelitian ini bertujuan untuk membandingkan kinerja dua arsitektur deep learning, yaitu DNN dan CNN, untuk klasifikasi data metagenom menggunakan spaced k-mers sebagai ekstraksi fitur. Klasifikasi dengan menggunakan deep learning memberikan hasil yang lebih baik, yaitu 90,89 % menggunakan DNN dan 88,89 % menggunakan CNN, dibandingkan dengan naïve Bayes yang menghasilkan akurasi sebesar 85,42 % pada taksonomi tingkat genus.

Kata kunci *– klasifikasi; deep learning; metagenom; spaced k-mers*

I. PENDAHULUAN

Pembacaan seluruh rantai DNA (genom) sudah biasa dilakukan dalam ilmu biologi molekular dan genetika. Pembacaan genom telah sampai ke tahap metagenom, yaitu pembacaan seluruh DNA yang diambil dari suatu ekosistem atau lingkungan, misalnya segenggam tanah atau isi perut manusia, tanpa budidaya di laboratorium atau isolasi genom individu [1]. Analisis metagenom ini telah diterapkan dalam bidang bioteknologi, ekologi, dan untuk medis [2]. Bahkan, berbagai penelitian menunjukkan potensi analisis mikroba di dalam usus untuk menangani berbagai penyakit, seperti diabetes [3] dan obesitas [4]. Sampel metagenom yang diambil dari suatu lingkungan menghasilkan fragmen yang mengandung berbagai macam mikroorganisme sehingga perlu dilakukan pengelompokan (*binning)* untuk mengetahui keragaman organisme dalam lingkungan mikroba tersebut [5].

Proses *binning* dapat dilakukan dengan dua pendekatan, yaitu pendekatan homologi dan pendekatan komposisi. Pada pendekatan homologi, dilakukan pencarian penjajaran sekuens dengan membandingkan fragmen metagenom dengan basis data sekuens dari *National Centre for Biotechnology Information* (NCBI). Hasilnya disimpulkan pada tiap level taksonomi. Metode yang menggunakan pendekatan homologi di antaranya adalah BLAST [1], [6] dan MEGAN [7].

[*)] Penulis korespondensi (Wisnu Ananta Kusuma)
Email: ananta@apps.ipb.ac.id

Submitted: 3 July 2020; Revised: 5 May 2020; Accepted: 25 May 2020; Published: 31 July 2020

Berbeda dengan pendekatan homologi, pendekatan komposisi tidak membandingkan sekuens *query* dengan sekuens referensi. Pendekatan komposisi menggunakan teknik pembelajaran mesin. Pasangan basa hasil ekstraksi fitur digunakan sebagai masukan untuk pembelajaran dengan observasi (*unsupervised learning / clusterring*) atau pembelajaran dengan contoh (*supervised learning / classification*).

Salah satu kajian dengan pendekatan komposisi melakukan teknik pembelajaran mesin dengan mengklasifikasikan fragmen metagenom menggunakan perhitungan frekuensi *k-mers* sebagai ekstraksi fitur dan *Multiclass Support Vector Machine* (SVM) sebagai *classifier* [8]. Hasil klasifikasi cukup bagus dengan panjang fragmen lebih dari 5 Kbp. Akurasi menurun dengan semakin pendeknya fragmen. Selain itu, penggunaan 5-*mers* menghasilkan 1024 fitur sehingga waktu pemrosesan menjadi lebih lama.

Kusuma dan Akiyama [9] menggunakan klasifikasi SVM dan *k-mers* frekuensi untuk mengekstraksi fitur yang melibatkan *don't care* (0) atau yang dikenal dengan frekuensi *spaced k-mers*. Frekuensi *spaced k-mers* menghasilkan 192 fitur. Secara keseluruhan, kajian ini sudah menghasilkan akurasi yang baik bahkan untuk panjang fragmen kecil 400 bp (base pair), yaitu 65,3 % untuk takson genus, 72 % untuk takson orde, 78,2 % untuk takson kelas, dan 82,1 % untuk takson filum. Pekuwali dkk. [10] melakukan penelitian mencari kombinasi fitur terbaik dari fragmen menggunakan algoritme genetika. Ektraksi fitur kajian tersebut menggunakan *spaced k-mers yang* menghasilkan pola 111 1111 10001 dengan 1 adalah *match* dan 0 adalah *don't care* sehingga menghasilkan 336 fitur. Klasifikasi menggunakan *Naive Bayesian Classifier* (NBC) dengan pola tersebut menghasilkan nilai akurasi sebesar 85,42 % pada takson genus.

Deep learning muncul kembali sebagai paradigma baru dalam pembelajaran mesin yang telah berhasil melakukan klasifikasi pola pada data yang besar (*big data*) [11]. Klasifikasi *Deep Neural Network* (DNN) juga digunakan untuk mengetahui interaksi protein-protein dengan akurasi sebesar 91,3 % [12]. *Deep Belief Network*, *Convolutional Neural Network* (CNN), dan NBC digunakan untuk klasifikasi data sekuens dengan menggunakan ekstraksi fitur 5-*mers* [13]. Dari kajian tersebut, diperoleh hasil bahwa metode CNN menghasilkan akurasi lebih baik dari metode *Deep Belief Network* dan NBC pada takson genus.

Deep learning mampu memberikan hasil akurasi yang lebih baik dibandingkan metode klasifikasi yang lain. Namun, kajian metode tersebut masih diperlukan untuk klasifikasi data metagenom. Pemilihan model juga diperlukan dengan melihat karakteristik dari data yang digunakan. Penelitian ini mengkaji penggunaan arsitektur CNN dan DNN untuk klasifikasi data metagenom. Selain itu, ekstraksi fitur spaced k-mers dengan pola 111 1111 10001 digunakan dalam kajian ini karena pola tersebut memberikan akurasi lebih baik dibandingkan k-mers biasa [13].

II. METODE PENELITIAN

Penelitian dilakukan dengan beberapa tahapan, yaitu penyiapan data, ektraksi fitur menggunakan *spaced* k-*mers*, pembuatan model DNN dan CNN, serta pengujian dan analisis.

A. Penyiapan data

Data mikroba dalam kajian ini menggunakan data penelitian dalam [9]. Data tersebut diperoleh dari situs NCBI yang diakses melalui http://www.ncbi.nlm.nih.gov. Data tersebut meliputi 19 spesies yang termasuk ke dalam 3 genus, yaitu genus *Agrobacterium*, *Bacillus*, dan *Staphylococcus*. Data yang diperoleh tersebut disimulasikan menggunakan DNA *sequencer* MetaSim [14] sehingga diperoleh sekuens DNA dalam format FASTA (*.fna).

Hasil simulasi tersebut menghasilkan 10000 fragmen data latih dengan komposisi 3000 fragmen *Agrobacterium*, 4000 fragmen *Bacillus*, dan 3000 fragmen *Staphylococcus*. Data uji yang digunakan adalah 4500 fragmen dengan 1500 fragmen *Agrobacterium*, 1500 fragmen *Bacillus*, dan 1500 fragmen *Staphylococcus*. Setiap fragmen mempunyai panjang fragmen yang tetap, yaitu 500 bp (*basepair*).

B. Ekstraksi fitur dengan spaced k-mers

Fitur dihasilkan dengan melakukan kombinasi dari 4 basa nukleotida *Adenin* (A), *Cytosine* (C), *Guanine* (G) dan *Thymine* (T) sehingga jumlah fitur kombinasi adalah 4^k dengan jumlah $k \geq 1$. Kajian ini menggunakan *spaced k-mers* dengan 1 adalah *match* dan 0 adalah *don't care* yang berarti membolehkan pasangan basa apapun mengisi bit tersebut [15]. Pola yang digunakan adalah 111 1111 10001 yang menghasilkan 336 fitur yang diperoleh dari kombinasi basa A, C, G dan T. Pola 111 membentuk fitur mulai dari AAA, AAC, AAG sampai TTT sehingga fitur yang dihasilkan pada pola ini adalah sejumlah 4^3 (64) fitur. Pola 1111 membentuk fitur mulai dari AAAA, AAAC, AAAG sampai TTTT sehingga fitur yang dihasilkan pada pola ini adalah sejumlah 4^4 (256) fitur.

Gambar 1 memberikan ilustrasi fragmen yang digunakan adalah AAAGAAACAGA. Karena pola yang digunakan adalah 111 yang berarti semua bit pada fitur harus cocok, maka untuk mengisi kolom fitur AAA dihitung berapa banyak AAA yang ada pada fragmen tersebut.

Pada pola 10001, karena 1 adalah *match* dan 0 adalah *don't care*, maka yang diperhatikan adalah bit pertama dan kelima saja seperti ditunjukkan dalam Gambar 2. Fragmen yang sama, yaitu AAAGAAACAGA, digunakan, namun dengan pola 10001. Bit 1 adalah bit yang harus cocok sedangkan 0 tidak peduli basa apa yang mengisi bit tersebut. Pola ini membentuk fitur mulai dari A***A, A***C, A***G sampai T***T sehingga fitur yang dihasilkan adalah sejumlah 4^2 (16) fitur. Setelah diperoleh fitur *spaced k-mers*, langkah selanjutnya adalah menghitung frekuensi kemunculan fitur.

C. Pembuatan model deep learning

Model *network* yang dibuat mempunyai 3 komponen lapis utama, yaitu lapis masukan, lapis tersembunyi, dan lapis keluaran. Pada lapis masukan, neuron yang digunakan adalah 336 neuron atau sebanyak fitur yang dihasilkan pada tahap ekstraksi fitur. *Backpropagation* digunakan untuk menghitung *loss function* dan memperbaiki nilai bobot yang menjadi masukan pada iterasi selanjutnya. Layer keluaran adalah jumlah genus yang telah direpresentasikan menggunakan *one-hot-vector*, yaitu [1,0,0] sebagai *Agrobacterium*, [0,1,0] sebagai *Bacillus* dan [0,0,1] sebagai *Staphylococcus*.

Simulasi arsitektur tersebut dapat dilihat pada Gambar 3 dimana X adalah masukan, Z adalah lapis tersembunyi, dan Y adalah keluaran. Jumlah neuron pada X adalah 336 yang berasal dari jumlah fitur, sedangkan Y adalah keluaran dengan jumlah neuron adalah 3. Z adalah lapis tersembunyi dimana banyaknya lapis ini dan neuron tersembunyi di tiap lapis mengikuti beberapa percobaan arsitektur. Kajian ini menggunakan arsitektur *deep learning* CNN dan DNN. CNN ini mempunyai filter (kernel) sendiri pada tiap lapis tersembunyi (*convolutional layer*).

DNN sebagai jaringan syaraf sederhana mempunyai banyak lapis tersembunyi dengan sekumpulan neuron dalam tiap lapis tersembunyinya. Tidak ada ukuran pasti untuk menentukan jumlah lapis tersembunyi dan jumlah neuron tersembunyi dalam membentuk arsitektur jaringan. Untuk itu, diperlukan metode untuk mendapatkan arsitektur yang paling optimal untuk digunakan sebagai classifier dalam pembelajaran mesin. Kajian ini menggunakan metode *try and error* dengan mengulangi beberapa variasi dari arsitektur *neural network* sampai pelatihan dan pengujian memberikan hasil yang terbaik. Aturan dalam *rule of thumb* adalah jumlah neuron tersembunyi harus berada di antara jumlah neuron pada lapis masukan dan keluaran [16]. Jumlah neuron tersembunyi 2/3 dari lapis masukan dan keluaran serta serta kurang dari 2 kali jumlah neuron pada lapis masukan.

D. Pengujian dan analisis model deep learning

Keluaran hasil pelatihan adalah model *deep learning* yang harus diuji seberapa baik model tersebut untuk memprediksi fragmen metagenom. Masukan pada tahap ini adalah data uji yang berjumlah 4500 fragmen dan keluarannya adalah hasil prediksi dari model tersebut. Hasil prediksi ini dievaluasi dengan menghitung akurasinya, yaitu seberapa banyak data yang diprediksi dengan benar (Persamaan 1).

$$\text{Akurasi} = \frac{\sum data\,uji\,benar}{\sum data\,uji} \text{ x } 100\% \qquad (1)$$

A A A G A A A C A G A

Pola 111 menghasilkan *substring*:
AAA AAG AGA GAA AAA AAC ACA CAG AGA

Fitur	AAA	AAG	…	CAA	…	CAT	…	TTT
Jumlah	2	1	…	0	…	0	…	0

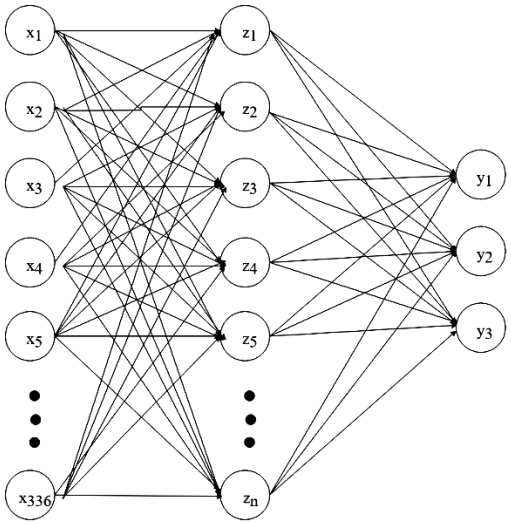
Gambar 1. Ilustrasi spaced k-mers dengan pola 111

A A A G A A A C A G A

Pola 10001 menghasilkan *substring*:
AAAGA AAGAA AGAAA GAAAC AAACA AACAG ACAGA

Fitur	A***A	A***G	…	C***T	…	T***T
Jumlah	5	1	…	0	…	0

Gambar 2. Spaced k-mers dengan pola 10001



Gambar 3. Arsitektur *simple neural network*

III. HASIL DAN PEMBAHASAN

A. Fitur spaced k-mers

Fitur *spaced k-mers* dengan pola 111 1111 10001 yang dihasilkan pada tahap ini adalah 336 fitur. Setelah dibentuk fitur, langkah selanjutnya adalah mencari frekuensi dari masing-masing fitur tersebut sehingga dari tahapan ini menghasilkan matriks komposisi yang berisi frekuensi kemunculan (Tabel 1). Tahapan ini dilakukan untuk data 10000 fragmen metagenome dan 4500 fragmen metagenome. Data 10000 fragmen selanjutnya digunakan untuk pelatihan pada model *deep learning*, sedangkan data 4500 fragmen digunakan untuk tahap pengujian model *deep learning*.

Tabel 1. Matriks komposisi fitur dengan pola 111 1111 10001 pada data latih

Fitur	AAA	…	TTT	AAAA	…	TTTT	A***A	…	T***T	Genus
Fragmen	(1)		(64)	(65)		(320)	(321)		(336)	
F1	6	…	17	0	…	8	12	…	34	*Agrobacterium*
F2	10	…	14	4	…	6	2	…	23	*Agrobacterium*
…	…	…	…	…	…	…	…	…	…	…
F10000	16	…	36	7	…	13	337	…	74	*Staphylococcus*

B. Pembuatan model deep learning

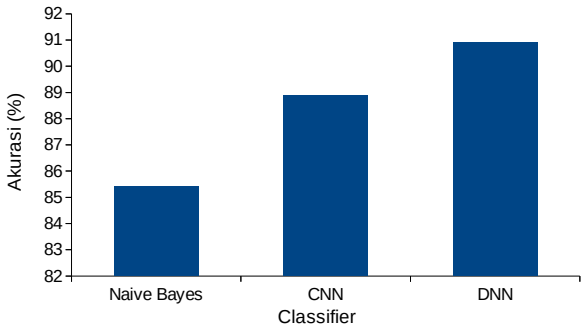
Dalam membangun arsitektur *neural network*, terdapat parameter-parameter yang digunakan, di antaranya jumlah neuron tersembunyi, jumlah lapis tersembunyi, fungsi optimasi, fungsi aktivasi, dan *batch size*. Untuk menentukan jumlah neuron dan lapis tersembunyi dilakukan beberapa percobaan *try and error* dan mengikuti aturan *rule of thumb* [16]. Pada percobaan pertama, jumlah *neuron* untuk tiap lapis adalah sama. Hasil percobaan tersebut dinyatakan dalam pada Tabel 2.

Percobaan tersebut menghasilkan akurasi yang sudah cukup bagus untuk tiap neuron pada tiap lapis tersembunyi, yaitu antara 88 % sampai 90 %. Rata-rata akurasi tertinggi diperoleh jika menggunakan 3 lapis, yaitu sebesar 90,42 %, dengan hasil tertinggi jika menggunakan 8 neuron pada tiap lapis, yaitu sebesar 90,92 %. Akurasi yang dihasilkan cukup tinggi, namun tidak terlihat perbedaan yang cukup signifikan antara banyaknya lapis dan neuron tersembunyi yang digunakan sehingga tidak perlu menggunakan banyak lapis dan neuron tersembunyi. Bahkan, jumlah lapis dan neuron tersembunyi yang terlalu banyak mengakibatkan *overfitting*. Dataset yang digunakan dalam kajian mempunyai 10000 data latih untuk mengklasifikasikan 3 genus sehingga data yang dihasilkan tidak kompleks.

C. Perbandingan classifier

Pada penelitian ini, arsitektur *classifier* DNN dibandingkan dengan arsitektur *classifier* CNN [13] dan *classifier* naïve Bayes [10]. Hasil akurasi dari klasifikasi data dinyatakan pada Gambar 4. DNN dalam kajian mempunyai akurasi tertinggi, yaitu sebesar 90,92 %, yang sebanding dengan [12], sedangkan CNN dan naïve Bayes masing-masing sebesar 88,89 % dan 85,42 %. Hasil ini juga menunjukkan bahwa CNN mempunyai akurasi lebih besar daripada naïve Bayes seperti yang dinyatakan dalam [13]. DNN ini juga mempunyai akurasi yang lebih baik daripada SVM dalam [9], yaitu sebesar 65,3 % dalam takson genus.

Lama waktu pelatihan dan pengujian tiap *classifier* dinyatakan dalam Tabel 3. Model *deep learning* memerlukan waktu pelatihan yang lebih lama karena terdapat perulangan untuk mendapatkan model yang terbaik sehingga lama pelatihan tiap *classifier* terpaut cukup jauh. Jika DNN memerlukan 128,28 detik untuk pelatihan, naïve Bayes hanya memerlukan 0,52 detik. CNN memerlukan waktu komputasi lebih lama untuk pelatihan dan pengujian, yaitu 1350,4 detik dan 1,23 detik. Hal ini disebabkan karena perhitungan pada CNN



Gambar 4. Hasil akurasi untuk tiap *classifier*

Tabel 2. Hasil akurasi tiap percobaan

Jumlah Layer Neuron	Akurasi (%) 2	3	4	5
5	88,60	90,30	89,35	89,73
6	88,73	90,50	89,21	89,59
7	88,30	90,60	89,19	89,15
8	89,34	**90,92**	89,58	89,76
9	89,05	90,37	89,12	89,14
10	89,12	90,10	89,09	89,43
15	89,12	90,00	89,48	89,36
20	88,66	90,26	89,95	89,47
30	89,40	89,50	89,49	89,78
40	90,19	89,89	89,76	90,18
50	89,70	90,36	90,14	90,00
75	90,72	90,72	90,57	90,62
100	90,17	90,46	90,37	89,25
125	90,31	90,81	90,44	89,73
150	90,82	90,57	90,24	90,53
175	90,50	90,50	90,68	90,10
200	90,77	90,89	90,44	90,44
225	90,61	90,61	90,01	90,20
250	90,66	90,85	90,67	90,76
275	90,41	90,20	90,28	90,32
300	88,60	90,30	89,35	89,73
Rata-rata	89,76	90,42	89,90	89,87

Tabel 3. Lama waktu pelatihan dan pengujian tiap *classifier*

Classifier	Pelatihan (detik)	Pengujian (detik)
DNN	128,28	0,13
CNN	1.350,4	1,23
naïve Bayes	0,52	0,18

lebih kompleks, yaitu mengekstrak fitur lokal dengan menggunakan filter yang berbeda pada tiap lapis tersembunyi.

IV. KESIMPULAN

Model *deep learning* CNN dan DNN mampu memberikan hasil akurasi yang lebih baik dibandingkan dengan naïve Bayes. Semakin kompleks model *deep learning* yang digunakan tidak menjamin akurasi yang dihasilkan lebih baik. Pada CNN, filter pada lapis konvolusi untuk fitur baru tidak membuat akurasi menjadi lebih baik dibandingkan DNN sederhana, bahkan membuat waktu komputasi menjadi lebih lama.

DAFTAR PUSTAKA

[1] H. Wu, "PCA-based linear combinations of oligonucleotide frequencies for metagenomic DNA fragment binning," in *IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology*, Sun Valley, USA, Sept. 2008, pp. 46-53. doi: 10.1109/CIBCB.2008.4675758

[2] C. Simon and R. Daniel, "Metagenomic Analyses: Past and Future Trends," *Applied and Environmental Microbiology*, vol. 4, no. 77, pp. 1153-1161, 2011. doi: 10.1128/AEM.02345-10

[3] J. Qin, "A metagenome-wide association study of gut microbiota in type 2 diabetes," *Nature*, vol. 490, pp. 55-60, 2012. doi: 10.1038/nature11450

[4] P. J. Turnbaugh, R. E. Ley, M. A. Mahowald, V. Magrini, E. R. Mardis, and J. I. Gordon, "An obesity-associated gut microbiome with increased capacity for energy harvest," *Nature*, vol. 444, pp. 1027-1031, 2006. doi: 10.1038/nature05414

[5] J.-L. Bouchot, W. L. Trimble, G. Ditzler, Y. Lan, S. Essinger, and G. Rosen, "Advances in machine learning for processing and comparison of metagenomic data," in *Computational Systems Biology*, Molecular Mechanisms to Disease: Second Edition, Elsevier, pp. 295-329, 2013. doi: 10.1016/B978-0-12-405926-9.00014-9

[6] H. Zheng and H. Wu, "A novel LDA and PCA-based hierarchical scheme for metagenomic fragment binning," in *IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology*, Nashville, USA, Apr. 2009, pp. 53-59. doi: 10.1109/CIBCB.2009.4925707

[7] D. H. Huson, A. F. Auch, J. Qi, and S. C. Schuster, "MEGAN analysis of metagenomic data," *Genome Research*, vol. 17, no. 3, pp. 377-386, 2007. doi: 10.1101/gr.5969107

[8] A. C. McHardy and I. Rigoutsos, "What's in the mix: phylogenetic classification of metagenome sequence samples," *Current Opinion in Microbiology*, vol. 10, no. 5, pp. 499-503, 2007. doi: 10.1016/j.mib.2007.08.004

[9] W. A. Kusuma and Y. Akiyama, "Metagenome fragment binning based on characterization vector," in *International Conference on Bioinformatics and Biomedical Technology*, Sanya, China, Mar. 2011, pp. 1-5.

[10] A. A. Pekuwali, W. A. Kusuma, and A. Buono, "Optimization of spaced k-mer frequency feature extraction using genetic algorithms for metagenome fragment classification," *Journal of ICT Research and Applications*, vol. 12, no. 2, pp. 123-137, 2018. doi: 10.5614/itbj.ict.res.appl.2018.12.2.2

[11] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436-444, 2015. doi: 10.1038/nature14539

[12] A. Fiannaca et al., "Deep learning models for bacteria taxonomic classification of metagenomic data," *BMC Bioinformatics*, vol. 19, no. 7, pp. 73-154, 2018. doi: 10.1186/s12859-018-2182-6

[13] P. Sunil, T. Rashmi, K. Vandana, and V. Pritish, "DeepInteract: deep neural network based protein-protein interaction prediction tool," *Current Bioinformatic*, vol. 12, no. 6, pp. 551-557, 2017. doi: 10.2174/1574893611666160815150746

[14] D. C. Richter, F. Ott, A. F. Auch, R. Schmid, and D. H. Huson, "MetaSim-A sequencing simulator for genomics and metagenomics," *PLoS ONE*, vol. 3, no. 10, pp. 417-412, 2008. doi: 10.1371/journal.pone.0003373

[15] B. M. J. Tromp and M. Li, "PatternHunter: faster and more sensitive homology search," *Bioinformatics*, vol. 18, no. 3, pp. 440-445, 2002. doi: 10.1093/bioinformatics/18.3.440

[16] S. Karsoliya, "Approximating number of hidden layer neurons in multiple hidden layer BPNN architecture," *International Journal of Engineering Trends and Technology*, vol. 3, no. 6, pp. 714-717, 2012.