

Operon Finder: a deep learning-based web server for accurate prediction of prokaryotic operons

Tejasvi Singh Tomar, Pratik Dasgupta, and Shankar Prasad Kanaujia*

Department of Biosciences and Bioengineering, Indian Institute of Technology Guwahati,
Guwahati 781039, Assam, India

*Corresponding author: spkanaujia@iitg.ac.in, spkanaujia@gmail.com

Author version of the accepted article. This version contains the content of the peer-reviewed article (received 13 October 2022, accepted 8 December 2022). The original accepted-manuscript file was unavailable, so the first author reconstructed this version from the authors' June 2022 manuscript and the changes made during peer review, with language edits (October 2026). It does not include the publisher's copyediting or typesetting. The article was published as: Tomar TS, Dasgupta P, Kanaujia SP (2023). Operon Finder: A deep learning-based web server for accurate prediction of prokaryotic operons. *Journal of Molecular Biology* 435(14), 167921. <https://doi.org/10.1016/j.jmb.2022.167921>. Please cite the published article. © 2022. This manuscript version is made available under the CC-BY-NC-ND 4.0 license <https://creativecommons.org/licenses/by-nc-nd/4.0/>.

Keywords: Operon; Genome analysis; Deep learning; Web server; Algorithm; Bioinformatics

Abstract

Operons are groups of adjacent genes that are transcribed together under the control of a common promoter. Because operons shape gene regulation and many physiological pathways, identifying them accurately is valuable. Detecting operons experimentally is slow and expensive, so experts often predict them from the genomic organization and functional relationships of genes. As the number of completely sequenced genomes grows, automated tools and web servers have become preferable to manual prediction. The most accurate current methods use deep learning to predict whether adjacent genes belong to the same operon, but using them requires programming knowledge and computational skills. Here we present Operon Finder, a web server for on-the-fly operon prediction with deep learning. It offers genome search, live filtering of predicted operons, viewing and downloading of operonic DNA sequences, export of the predictions, and links to the NCBI (National Center for Biotechnology Information) and UniProt databases. Operon Finder is available at <https://www.iitg.ac.in/spkanaujia/operonfinder.html>, and its source code is available at <https://github.com/SPKlab/Operon-Finder>.

1 Introduction

An operon is a set of contiguous genes that are co-transcribed. The genes in an operon often belong to the same regulatory pathway and depend on one another. Studies of operons have revealed many patterns of gene regulation and have helped to identify pathogenicity factors that may serve as drug targets (Bergman et al., 2007). These applications have drawn wide interest in identifying operons. Operons have traditionally been detected experimentally, which is time-consuming and costly. Gene-level data and the known functions of proteins also allow experts to predict operons by inspection, but such predictions are error-prone.

Several computational methods have been developed to automate operon prediction and improve its accuracy. Qin and co-workers developed a hidden Markov model (HMM)-based method that uses intergenic distances and the phylogenetic distribution of gene orthologs (Bergman et al., 2007). Merino and co-workers outperformed earlier methods by combining intergenic distances with STRING functional association scores in a two-layer neural network (Taboada et al., 2010). Their predictions were collected in the Prokaryotic Operon DataBase (ProOpDB; Taboada et al., 2012), and the method later became Operon-mapper, a web server that predicts operons in a user-supplied genome (Taboada et al., 2018). Building on advances in deep learning, Stevens and co-workers developed Operon Hunter, which, to the best of our knowledge, is the most accurate method for detecting operons in prokaryotes (Assaf et al., 2021). Operon Hunter draws the genomic context of each pair of adjacent genes as an image and classifies it with an 18-layer residual network (ResNet18; He et al., 2016), reaching 85% accuracy on experimentally verified operons. However, its hardware requirements, including a GPU (graphics processing unit), make it unsuitable for personal use or for a public web service. Operon Finder keeps Operon Hunter's visual representation but replaces the model and the prediction pipeline with optimized counterparts suited to a fast and accurate public web service.

2 Methods

2.1 Prediction pipeline

Operon Finder follows the method of Operon Hunter (Assaf et al., 2021; Figure 1). Users enter a genome identifier in a web form or pick one of 9,140 prokaryotic genomes from a searchable drop-down menu. For the selected genome, Operon Finder retrieves the protein-coding genes,

ordered by position, and their genomic context from the PATRIC (PathoSystems Resource Integration Center) database (Wattam et al., 2014), and the functional association scores between their encoded proteins from the STRING database, version 11.5 (Szklarczyk et al., 2021). For each gene, the PATRIC compare-region service returns the surrounding region together with the corresponding regions of related reference and representative genomes, in which genes are grouped into protein families. STRING proteins are matched to PATRIC genes through RefSeq locus tags, using the identifiers that STRING maps to its proteins by BLAST (Altschul et al., 1990) against KEGG and UniProt: KEGG identifiers, ordered locus names, and open reading frame names. Some PATRIC genomes lack RefSeq locus tags for some of their genes. To extend the analysis to these genomes, Operon Finder assigns approximate tags from these identifiers and from the tags of neighboring genes, and flags the approximate assignments in the results.

The pipeline then draws one image for each adjacent gene pair. The image shows the neighborhood of the pair in the selected genome and in the related genomes, one genome per row. Genes are drawn as arrows that point in the direction of transcription and are colored by protein family. The STRING score of the pair sets the opacity (alpha value) of the arrows. The prediction model classifies each image and returns the probability that the two genes belong to the same operon, which Operon Finder reports as a confidence score between 0 and 1.

Consecutive gene pairs whose confidence score reaches a threshold of 0.5 are merged into predicted operons. Users can lower the threshold to look for candidate operons, but predictions below 0.5 are less reliable.

2.2 Optimizations

We optimized both the data pipeline and the prediction model (Figure 1). Operon Finder parses the text files from STRING and PATRIC with regular expressions, which Python executes in compiled C code, instead of the procedural Python logic of Operon Hunter. This makes parsing about ten times faster. Parsed data are stored as JSON (JavaScript Object Notation) files, and responses from the remote databases are cached locally, so the data for a genome are processed only once and the load on the external databases is reduced. Operon Hunter computes the image pixels for one gene pair at a time. Operon Finder distributes this work across all CPU (central processing unit) cores, which speeds it up roughly by the number of cores. It also reduces the

images from 300×300 to 64×64 pixels, with no measurable loss of accuracy and less disk space.

For prediction, Operon Finder replaces the ResNet18 network of Operon Hunter with MobileNetV2 (Sandler et al., 2018), which has 70% fewer parameters and runs faster without a significant loss of accuracy. On the server, the model is compiled just in time with TorchScript instead of running in the eager mode that Operon Hunter uses, which mainly helps to catch programming errors (Paszke et al., 2019). This makes prediction 27% faster. With quantization-aware training, the model weights are stored at 16-bit instead of 32-bit precision, cutting storage and computation by half (Jacob et al., 2018; Ma et al., 2020). Together, these optimizations let Operon Finder provide on-demand predictions to multiple users.

2.3 Web server implementation

The web server is written mainly in Python and runs on a machine with a 32-core processor and 64 GB of RAM (random access memory). The interface is built with the Streamlit framework, which serves the application with the Tornado web server and renders the page with the React library. It sends results to the browser as Protocol Buffers messages over a WebSocket connection, so the page updates without reloading. Protocol Buffers messages are smaller than equivalent JSON messages (Popić et al., 2016). Several users can use the interface at the same time. Predictions for new genomes are computed in background processes, and requests to PATRIC are sent concurrently. Predictions are saved on disk as JSON files and reused when the same genome is requested again. Users can leave an email address to be notified when the predictions for a new genome are ready. The server is accessed through Cloudflare, which hides the server's IP (Internet Protocol) address and protects it from malicious traffic.

3 Results

3.1 Using Operon Finder

Users choose a genome either by searching the 9,140 supported prokaryotic genomes by organism name or by entering a PATRIC genome identifier. Predicted operons are listed in order of their genomic start positions and numbered sequentially (Figure 1). For each operon, a table lists the genes with their RefSeq locus tags, descriptions, encoded proteins, and UniProt

accessions, linked to the NCBI and UniProt databases, together with the confidence score of each gene pair and the intergenic distances. Confidence scores are colored from red (0) through yellow (0.5) to green (1). The DNA sequence spanning any range of genes in an operon can be displayed and downloaded. Operons can be filtered live by confidence threshold, number of genes, gene identifiers, and keywords in the gene descriptions. The results can be downloaded as a tab-separated values (TSV) file for local analysis.

3.2 Accuracy and speed

We evaluated Operon Finder on 254 experimentally verified operons of *Escherichia coli* str. K-12 substr. MG1655 and *Bacillus subtilis* subsp. *subtilis* str. 168. Stevens and co-workers used these operons as a test set and excluded them from training (Assaf et al., 2021). Operon Finder predicted the operon boundaries with 86% accuracy, compared with 85% for Operon Hunter. It took 0.46 s to predict a single gene pair, compared with 1.93 s for Operon Hunter, a 76% reduction in prediction time (Figure 1). Operon Finder is therefore faster than Operon Hunter without being less accurate.

To test the predictions further, we downloaded all experimentally validated operons in RegulonDB (Tierrafría et al., 2022) and kept those with a strong evidence level. We ran Operon Finder on the genome of *E. coli* K-12 and compared its predictions with these operons. Operon Finder cannot predict operons that consist of genes for RNAs other than mRNAs, so these operons were excluded. On the remaining operons, Operon Finder reached 86.9% sensitivity, 83.7% precision, 75.9% specificity, and 82.4% accuracy, computed as described by Assaf et al. (2021). The full comparison is given in the supplementary data (D1) of the published article.

4 Conclusion

Operon Finder is a public web server that provides on-demand operon predictions with a deep learning method for 9,140 prokaryotic genomes. Through the optimizations described above, including the lighter MobileNetV2 architecture, it cuts the prediction time per gene pair by 76% relative to Operon Hunter without compromising accuracy. Users can browse operons, filter them on several criteria, follow links to NCBI and UniProt, and download the predictions and the operonic DNA sequences. We hope that Operon Finder will speed up research that relies on gene annotation.

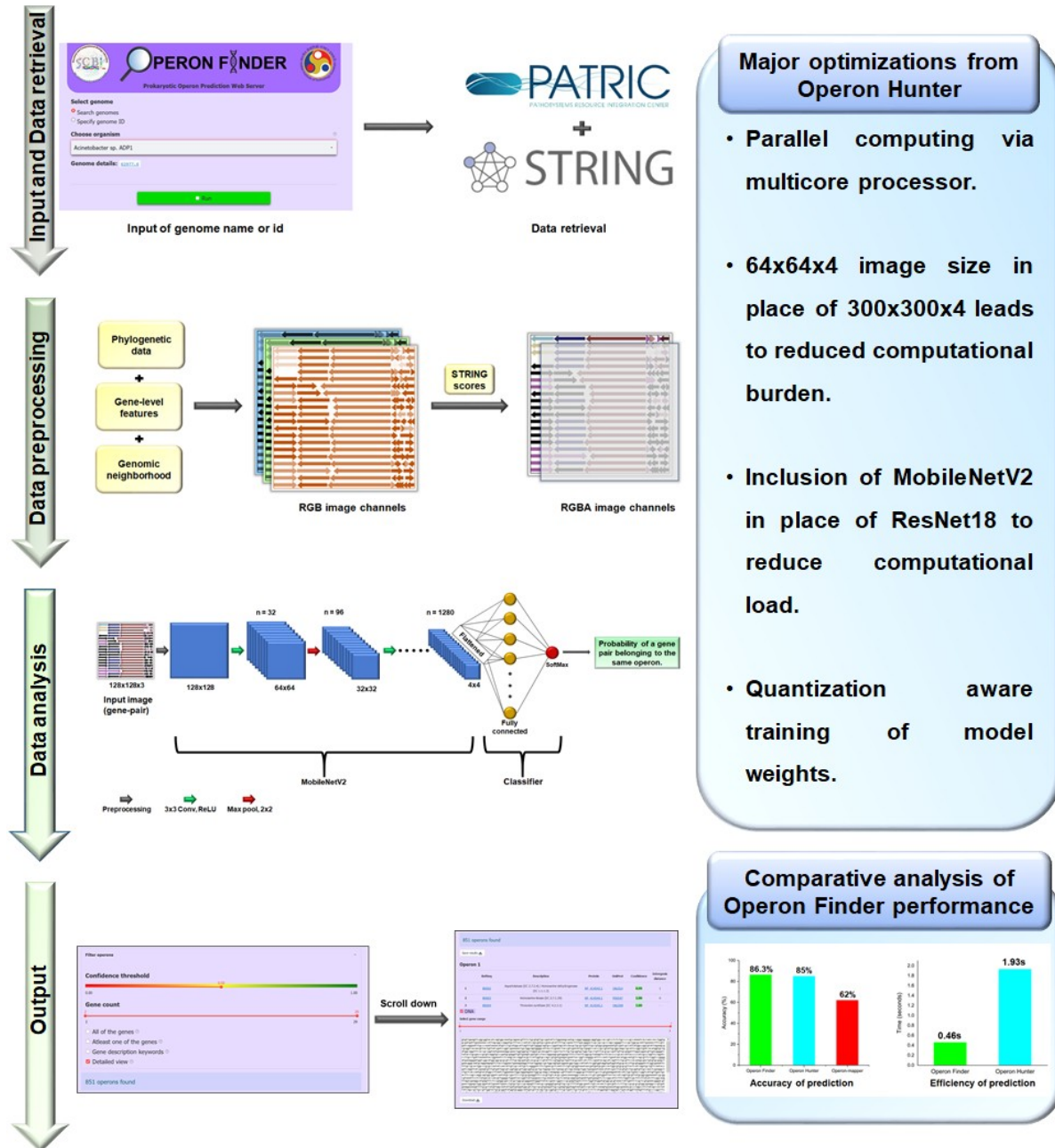


Figure 1. Overview of Operon Finder. The four stages are shown from top to bottom. Input and data retrieval: the user selects a genome, and data are retrieved from PATRIC and STRING. Data preprocessing: phylogenetic data, gene-level features, and the genomic neighborhood of each gene pair are drawn as an image in which the STRING score sets the opacity of the arrows. Data analysis: a MobileNetV2-based classifier returns the probability that the two genes belong to the same operon. Output: predicted operons are filtered and listed with gene details and their DNA sequences. The top-right box lists the main optimizations over Operon Hunter. The box below compares the accuracy of Operon Finder, Operon Hunter, and Operon-mapper, and the time Operon Finder and Operon Hunter take per gene pair.

Data availability

Web server: <https://www.iitg.ac.in/spkanaujia/operonfinder.html>

Source code: <https://github.com/SPKlab/Operon-Finder>

Acknowledgments

We thank the Indian Institute of Technology Guwahati for its facilities and our colleagues in the Structural and Computational Biology Laboratory (SCBL) for support. P.D. received a research fellowship from the Ministry of Education (MoE), Government of India.

Author contributions

T.S.T. implemented the method and built the web server. P.D. performed the data analysis and drafted the manuscript. S.P.K. conceived the project and finalized the manuscript.

Competing interests

The authors declare no competing interests.

References

- Altschul SF, Gish W, Miller W, Myers EW, Lipman DJ (1990). Basic local alignment search tool. *J. Mol. Biol.* 215(3), 403–410. [https://doi.org/10.1016/S0022-2836\(05\)80360-2](https://doi.org/10.1016/S0022-2836(05)80360-2)
- Assaf R, Xia F, Stevens R (2021). Detecting operons in bacterial genomes via visual representation learning. *Sci. Rep.* 11(1), 2124. <https://doi.org/10.1038/s41598-021-81169-9>
- Bergman NH, Passalacqua KD, Hanna PC, Qin ZS (2007). Operon prediction for sequenced bacterial genomes without experimental information. *Appl. Environ. Microbiol.* 73(3), 846–854. <https://doi.org/10.1128/AEM.01686-06>
- He K, Zhang X, Ren S, Sun J (2016). Deep residual learning for image recognition. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 770–778. <https://doi.org/10.1109/CVPR.2016.90>
- Jacob B, Kligys S, Chen B, et al. (2018). Quantization and training of neural networks for efficient integer-arithmetic-only inference. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2704–2713. <https://doi.org/10.1109/CVPR.2018.00286>
- Ma X, Yuan G, Lin S, et al. (2020). Tiny but accurate: A pruned, quantized and optimized memristor crossbar framework for ultra efficient DNN implementation. In: *25th Asia and South Pacific*

- Design Automation Conference (ASP-DAC)*, 301–306. <https://doi.org/10.1109/ASP-DAC47756.2020.9045658>
- Paszke A, Gross S, Massa F, et al. (2019). PyTorch: An imperative style, high-performance deep learning library. In: *Advances in Neural Information Processing Systems 32 (NeurIPS 2019)*. <https://arxiv.org/abs/1912.01703>
- Popić S, Pezer D, Mrazovac B, Teslić N (2016). Performance evaluation of using Protocol Buffers in the Internet of Things communication. In: *International Conference on Smart Systems and Technologies (SST)*, 261–265. <https://doi.org/10.1109/SST.2016.7765670>
- Sandler M, Howard A, Zhu M, Zhmoginov A, Chen L-C (2018). MobileNetV2: Inverted residuals and linear bottlenecks. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 4510–4520. <https://doi.org/10.1109/CVPR.2018.00474>
- Szklarczyk D, Gable AL, Nastou KC, et al. (2021). The STRING database in 2021: customizable protein–protein networks, and functional characterization of user-uploaded gene/measurement sets. *Nucleic Acids Res.* 49(D1), D605–D612. <https://doi.org/10.1093/nar/gkaa1074>
- Taboada B, Verde C, Merino E (2010). High accuracy operon prediction method based on STRING database scores. *Nucleic Acids Res.* 38(12), e130. <https://doi.org/10.1093/nar/gkq254>
- Taboada B, Ciria R, Martinez-Guerrero CE, Merino E (2012). ProOpDB: Prokaryotic Operon DataBase. *Nucleic Acids Res.* 40(D1), D627–D631. <https://doi.org/10.1093/nar/gkr1020>
- Taboada B, Estrada K, Ciria R, Merino E (2018). Operon-mapper: a web server for precise operon identification in bacterial and archaeal genomes. *Bioinformatics* 34(23), 4118–4120. <https://doi.org/10.1093/bioinformatics/bty496>
- Tierrafraía VH, Rioualen C, Salgado H, et al. (2022). RegulonDB 11.0: Comprehensive high-throughput datasets on transcriptional regulation in *Escherichia coli* K-12. *Microb. Genom.* 8(5), 000833. <https://doi.org/10.1099/mgen.0.000833>
- Wattam AR, Abraham D, Dalay O, et al. (2014). PATRIC, the bacterial bioinformatics database and analysis resource. *Nucleic Acids Res.* 42(D1), D581–D591. <https://doi.org/10.1093/nar/gkt1099>