Honglin Wang

honglin.wang@uconn.edu • (860) 771-8889 • harry-wang12.github.io • linkedin.com/in/honglin-harry-wang/

OVERVIEW

5+ years of experience scientist in Bioinformatics, Deep Learning and Computer Vision; Have published 9 papers in peer-reviewed conferences and journals; Proficient in analysis massive bio-related datasets (Text, cell image, next-generation sequence) using various models.

EDUCATION

UNIVERSITY OF CONNECTICUT, ***School of Engineering*** Storrs, CT
Ph. D. in Computer Science and Engineering *Feb 2024*
Awards: Predoctoral prize for research excellence; Doctoral Dissertation Fellowship

UNIVERSITY OF CONNECTICUT*, School of Engineering* Storrs, CT
M. Sc. in Computer Science and Engineering *Jan 2019*
Master Thesis: A Method to Score Pathways Using Heuristic Rules

RECENT RESEARCH PROJECTS

A semi-supervised learning framework for cell type classification using graph attention neural networks *Aug 2023*

- Captured high-resolution images of mouse articular cartilage stained using MERFISH techniques. Employed image processing methods with Python-OpenCV including reducing noise, thresholding, and segmenting to detect and isolate cells and RNA stain.
- Derived graphs for each cell and its neighboring cell to create training data. Collaborated with colleagues from the biology department to assign node labels, resulting in a dataset comprising 3472 graphs categorized into 5 classes. Each node in the graph contains 11 features.
- Developed a graph-based classification model using PyTorch, comprising 2 graph attention layers and 2 fully connected layers. Each graph attention layer integrates 3 heads for enhanced feature extraction. Trained and tuned the model using a maximum of 10% labeled datasets, simulating scenarios where biologists have limited labeled samples.
- Conducted a comparative analysis of classification performances across different models, including artificial neural networks and graph convolutional networks. Our model demonstrated the highest accuracy of 75.34% compared to other models. Visualized the model's accuracy and loss during training iterations using TensorBoard and Seaborn.

A chat robot with retrieval-augmented generation using LangChain *Jan 2023*

- Developed a web crawler using Python to gather 30,000+ bio-related scientific publications from PubMed. Extracted key information such as title, publisher, published date, abstract, and main content from both the website and PDF.
- Processed and stored over 85,000 text embeddings obtained from the collected data into the Milvus vector DB using the HuggingFace Instructor embedding model.
- Created a Q&A conversation chain for retrieving information from the vector DB, enhancing the prompt's background with maximum marginal relevance and employing map reduce methods.Fine-tuned the pretrained flan-t5 model using LoRA method with self-collected bio related scientific text questions and achieve 0. 43 ROUGE score.
- Constructed a chat robot by integrating the conversation chain with fine-tuned model using LangChain and Gradio. Increased the effectiveness of the chat bot in answering benchmark questions from 30% to 78% and visualized the performance with weights& biases.

A CNN based framework for predicting gene relationships and generating gene network *Aug 2022*

- Extracted and processed gene expression matrix tables from a self-built MySQL database. Concurrently collected gene relationships from NIH and KEGG databases using a web crawler. Conducted preprocessing of the gene expression matrix involving outlier imputation and normalization with Scanpy.
- Constructed training image datasets by segmenting gene values into histograms comprising 16 bins. Generated 3D co-expression images from various combinations of three gene histograms, categorizing each image into one of three classes based on established gene relationships.
- Developed a 3D images classifier utilizing PyTorch, integrating 2 layers of 3-D CNN and 2 fully connected layers. The model achieves 80.41% accuracy on the test dataset.
- Applied the trained model to two distinct datasets and analyzed the outcomes using hypothesis testing, revealing a statistically significant difference between the two results (P-value=0.021) upon comparison.
- Utilized the model's predictions to construct gene networks by employing a self-developed algorithm and leveraged Dash for visualizing the model's performance metrics and the networks.

WORK EXPERIENCE

UConn Computer Science and Engineering Department Storrs, CT

Research Assistant *Jan 2019 – Present*

- Achieved a 95% level of automation in streamlining the end-to-end process of preprocessing, analyzing, and visualizing Next Generation Sequencing data.
- Designed and implemented two databases (SQL and NoSQL) utilizing MySQL and MongoDB, respectively, to efficiently store and manage over 200 publicly available bio-datasets. This initiative resulted in a significant 30% enhancement in data collection efficiency for subsequent research projects.
- Deployed and maintain a based pipeline to automatically receiving, preprocessing, segmenting and sending out the high-resolution cell images by deploying python program on AWS. The accuracy of segmentation reaches 93%, resulting ~50% efficiency improvement for the downstream analysis.
- Collaborated with experts from various university departments, acquiring diverse domain knowledge and contributing to the analysis of research outcomes, resulting in the development of over 10 research manuscripts, with 9 papers successfully published in respected academic journals or conference proceedings.

Biogen Cambridge, MA

Co-op, Advanced Analytics, Digital Health *Jun 2023 – Aug 2023*

- Designed and implemented an end-to-end pipeline employing Python, R, and SQL for the streamlined acquisition and preprocessing of patient clinical trials data sourced from company databases. Achieved a 20% efficiency improvement in data handling, ensuring enhanced quality and consistency for subsequent analyses.
- Implemented a dynamic-programming algorithm to effectively stratify patients, further optimizing its performance through hyperparameter optimization. The results of applying this algorithm to over 900 multiple sclerosis patients providing robust evidence supporting the effectiveness of the treatment approach.
- Developed an interactive website utilizing R-Shiny and Dash frameworks to dynamically select and execute hypothesis testing based on stratification results. Created comprehensive visualizations depicting both the stratification outcomes and hypothesis testing results. Successfully deployed this interactive platform within the company's network infrastructure using Amazon Web Services (AWS). The website garnered a substantial traffic of over 2000 visits until the completion of the internship.

UConn Information Technology Services Storrs, CT

Web developer / Software Engineer *May 2019 – Jun 2023*

- Led the front-end development team in designing and implementing a responsive web application using HTML, JQuery and CSS, resulting in a 30% increase in user engagement.
- Successfully implemented more than 5 end-to-end websites specifically designed for visualizing and operationalizing database-stored data.
- Implemented performance optimization strategies, including lazy loading, caching, and image compression, resulting in a 50% reduction in page load times.
- Developed and maintained comprehensive documentation for projects, ensuring clear understanding and seamless knowledge transfer among team members.

TECHNICAL CAPABILITIES

Programming Languages: Python (PyTorch, SciPy, Scikit-learn, OpenCV-Python, PYG), R, Matlab, SQL, PHP, JavaScript, Java.

Data Tools: PowerBI, Tableau, Shiny, AWS, PySpark, Hadoop, Github, Apache, GCP, Hugging Face, Spark, MySql, MongoDB.

Data science methods: LLM, Deep learning, Clustering, Computer Vision, Nature language processing, Hypothesis testing.

Bioinformatics packages: GOstat, GSEA, SPIA, KEGG, Seurat, Scanpy, BLAST, Tophat, Cufflinks, SALMON, STAR, SAMTools, DAVID, UCSC Genome Browser.