Siddhartha Jain

Contact Information

Siddhartha Jain
Computer Science Department
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA 15213

Email: tmfs10@gmail.com
Homepage: https://tmfs10.github.io

Research Interests

Genomics, Protein design, Molecular dynamics, Machine learning, Optimization

Work Experience

Massachusetts Institute of Technology, Cambridge, MA
Postdoctoral Associate December 2017 – Present

Education

Carnegie Mellon University, Pittsburgh, PA
Ph.D. Computer Science September 2011 – October 2017
advised by Prof. Ziv Bar-Joseph (Feb 2013 – October 2017)

Brown University, Providence, RI
B.Sc. Mathematics-Computer Science September 2007 – May 2011

Refereed Publications

Using neural networks for reducing the dimensions of single-cell RNA-Seq data.
Chieh Lin, Siddhartha Jain, Hannah Kim, Ziv Bar-Joseph.
Nucleic Acids Research, 2017.

Transcriptome analyses identify key cellular factors associated with HIV-1 associated neuropathogenesis in infected men.
Narasimhan J. Venkatachari, Siddhartha Jain, Leah Walker, Shalmali Bilavaker-Mehla, Ansuman Chattopadhyay, Ziv Bar-Joseph, Charles Rinaldo, Ann Ragin, Eric Seaberg, Andrew Levine, James Becker, Eileen Martin, Ned Sacktor, Velpandi Ayyavoo.
AIDS Journal, 2017.

Reconstructing the temporal progression of HIV-1 immune response pathways.
Siddhartha Jain, Joel Arrais, Narasimhan J. Venkatachari, Velpandi Ayyavoo, Ziv Bar-Joseph.
Intelligent Systems for Molecular Biology, (ISMB), 2016

Temporal transcriptional response to latency reversing agents identifies specific factors regulating HIV-1 viral transcriptional switch.
Narasimhan J Venkatachari, Jennifer M Zerbato, Siddhartha Jain, Allison E Mancini, Ansuman Chattopadhyay, Nicolas Sluis-Cremer, Ziv Bar-Joseph, Velpandi Ayyavoo.
Retrovirology, 2015.

Multitask Learning of Signaling and Regulatory Networks with Application to Studying Human Response to Flu.
Siddhartha Jain, Anthony Gitter, and Ziv Bar-Joseph.

PLOS Computational Biology. 10:12, 2014 and
Society for Laboratory Automation & Screening, (SLAS), 2015

Large Neighborhood Search for the Dial-a-Ride Problem.
Siddhartha Jain and Pascal Van Hentenryck.
17th International Conference on Principles and Practices of Constraint Programming, (CP), 2011.

A General Nogood-Learning Framework for Pseudo-Boolean Multi-Valued SAT.
Siddhartha Jain, Ashish Sabharwal, and Meinolf Sellmann.
25th Conference on Artificial Intelligence, (AAAI), 2011.

A Complete Multi-Valued SAT Solver.
Siddhartha Jain, Eoin O'Mahony, and Meinolf Sellmann
16th International Conference on Principles and Practice of Constraint Programming, (CP), 2010.

Upper Bounds on the Number of Solutions of Binary Integer Programs.
Siddhartha Jain, Serdar Kadioglu, and Meinolf Sellmann.
7th International Conference on Integration of AI and OR Techniques in Constraint Programming, (CP), 2010.

Posters

Reconstructing the temporal progression of HIV-1 immune response pathways.
Siddhartha Jain, Joel Arrais, Narasimhan J. Venkatachari, Velpandi Ayyavoo, Ziv Bar-Joseph.
Probabilistic Modeling in Genomics, 2015.

Transfer learning for reconstructing dynamic signaling and regulatory networks.
Siddhartha Jain, Anthony Gitter, and Ziv Bar-Joseph.
18th Annual International Conference on Research in Computational Molecular Biology, (RECOMB), 2014.

Technical reports

Parallel Heuristics for TSP on MapReduce.
Siddhartha Jain and Matthew Mallozzi.
Brown University Tech Report

Research Summary

Modeling effects of the epigenome on gene expression [1] We're working on modeling the effects of epigenetic modifications – specifically DNA methylation and select histone modifications of gene body, promoter, and enhancer regions on gene expression. Our aim is to isolate the enhancer and promoter regions that are important to a gene's expression. Given that, and knowledge of which TF binding sites (which can be obtained via Chip-Seq/motif scan/etc), we can output TF-DNA interaction networks conditioned on the epigenome. We intend to build a regression based model for gene expression and further also explore whether deep learning techniques can be used to improve performance.

Analyzing single-cell RNA-seq data using neural networks [2] We developed a method based on neural networks (NN) for the analysis and retrieval of single cell RNA-Seq data. We tested various NN architectures, some that incorporated protein-DNA and protein-protein interaction networks into the architecture, and used these to obtain a reduced dimension representation of the single cell expression data. We showed that the NN method improves upon prior methods in both, the ability to correctly group cells in experiments not used in the training and the ability to correctly infer cell type or state by querying a database of tens of thousands of single cell profiles. Such database queries (which can be performed

[1] Ongoing project
[2] Working paper

using our web server) can enable researchers to better characterize cells when analyzing heterogeneous scRNA-Seq samples.

Reconstructing temporal progression of signaling pathways[1] A stimulant (like a virus) to a cell can interact with proteins in the cell and cause a cascade of signaling pathways that activate/repress the expression of genes downstream. The differential expression of such genes can cause further signaling cascades that cause the differential expression of another set of genes. While the gene expression at different time points can be observed, it is not possible to directly observe the signaling pathways responsible for their differential expression. We developed a tool, TimePath, to learn the progression of such signaling pathways across time based on time series gene expression data and protein-protein interaction data.

Multitask learning of signaling and regulatory networks [3] We developed a tool, MT-SDREM, to jointly learn signaling pathways and regulatory networks using time series gene expression data from multiple related conditions, TF-gene interaction data, and a protein-protein signaling network. Our tool built on an existing tool called SDREM which learnt the signaling pathways and regulatory network for just a single condition. For joint learning, we shared information the following information :- (1) we ensured that the condition-specific networks learnt were consistent – i.e. the edge directions were the same for all networks (2) If a TF was predicted to regulate more than one condition, it's prior for regulating any condition was increased. Validation with respect to RNAi screen hits, GO analysis and manual examination of the predicted signaling proteins demonstrated the advantage of joint inference.

Heuristics for Mixed Integer Programming We programmed a Mixed Integer programming solver from scratch (using existing LP solvers) and experimented with various node selection and branching heuristics with application to the Warehouse Location problem.

Large neighborhood search for Dial-a-ride problem[1] The Dial-a-ride problem involves a set of passengers that have to be picked up and dropped off at various locations within certain time windows. We developed an algorithm for this problem that was able to obtain close to optimal solutions much more quickly than the state of the art algorithms while still being able to prove if the problem was infeasible – something that most other algorithms did not do.

Clause Learning for Constraint Programs[1] Redundant clause learning is a popular technique used by SAT solvers to speed up search for a solution or proof of infeasibility for a boolean formula. We developed an algorithm to effectively learn clauses for a class of problems called multi-valued satisfiability problems which are a generalization of boolean formulas. Experimental results showed that our algorithm was several times faster than the standard technique of simply converting the multi-valued formula into its boolean equivalent.

Solution counting[1] Solution counting for integer programs with only binary variables is a #P-complete problem. The motivation for this project was that several good techniques for lower bounding the number of solutions for such problems had been developed but the only non-trivial upper bounding technique was extremely slow (in fact, in experiments with the benchmarks we were looking at, it always timed out). We developed an upper bounding technique based on dynamic programming that was able to obtain non-trivial upper bounds much faster for several problems.

Automated Failure Recovery We examined a programming language abstraction called Stabilizers which had been developed for Concurrent ML for automatically recovering from failures in program execution. We wrote a Javascript library for doing the equivalent for web applications (a use case would be if the network connection suddenly died).

Honours and Awards

Wally George Fellowship offered at Georgia Tech for Ph.D. studies
Ontario Trillium Scholarship offered at U. of Toronto for Ph.D. studies
Undergraduate Teaching and Research Assistanship Award Grant for doing Research for Summer

[3] Led to one or more publications

2010
Perry and Dr. Hilary Hoffmeister Brown Annual Fund Scholarship for years 2007-8, 2008-09, 2010-11

Teaching Experience

Carnegie Mellon University

- Graduate teaching assistant for Introduction to Machine Learning in Fall 2014
- Graduate teaching assistant for Principles of Imperative Computation in Spring 2014

Brown University

- Head Teaching Assistant for *Introduction to Computer Systems* in Fall 2009 and Fall 2010
- Teaching Assistant for *Design & Analysis of Algorithms* in Spring 2010
- Teaching Assistant for *Introduction to Computer Systems* in Fall 2008

Community Service

Reviewer for RECOMB 2017, Bioinformatics 2016, Recomb 2016, PLOS One 2014, ISMB 2015, RECOMB 2015, Bioinformatics 2015, ACM-BCB 2015, CP 2013, CPAIOR 2013

References

Ziv Bar-Joseph
Carnegie Mellon University
`zivbj@cs.cmu.edu`
Tel: 412-268-8595

Naftali Kaminski
Yale University
`naftali.kaminski@yale.edu`

Velpandi Ayyavoo
University of Pittsburgh
`velpandi@pitt.edu`
Tel: 412-624-3070