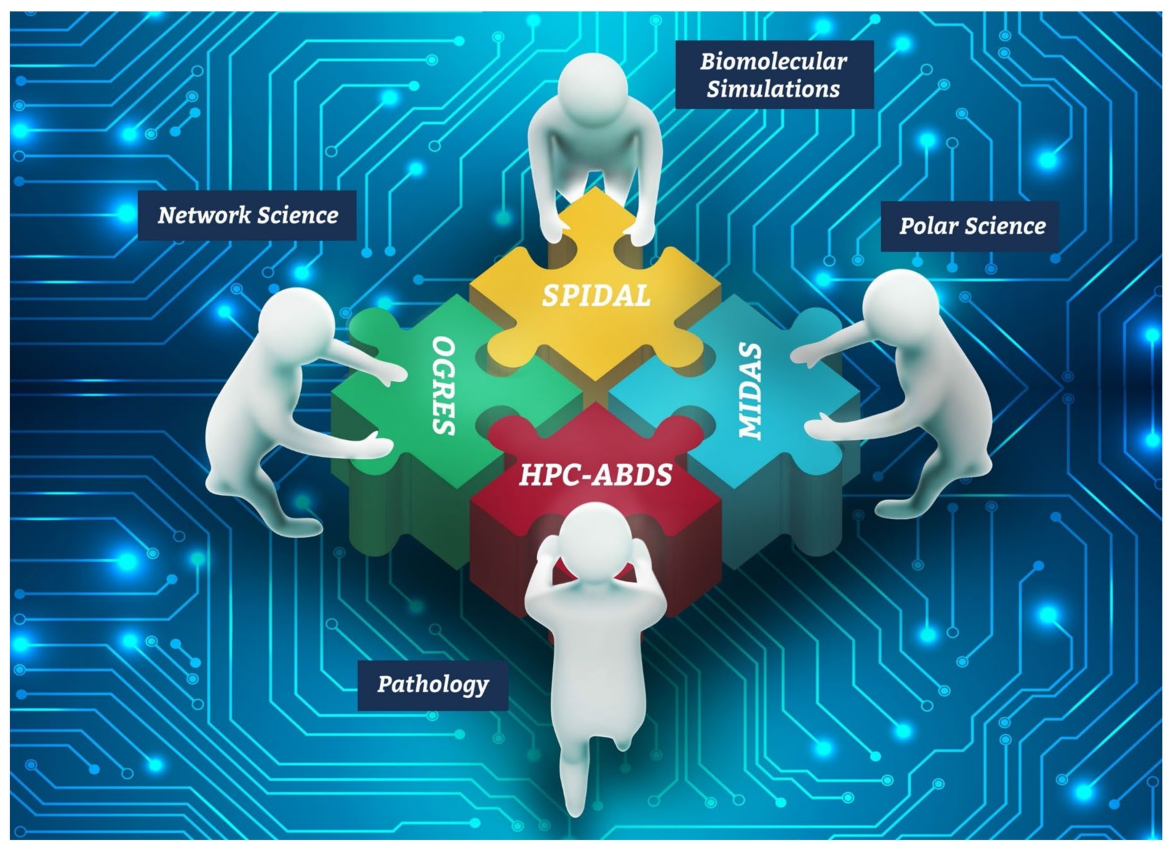
CIF21 DIBBs: Middleware and High Performance Analytics Libraries for Scalable Data Science

PI: G. Fox, Co-PIs: Madhav Marathe, Shantenu Jha, Judy Qiu, Fusheng Wang
Institutions: Arizona State, Indiana (lead), Kansas, Rutgers, Stony Brook, Virginia, and Utah.

Award #: 1443054

MLforHPC and HPCforML

- Project mainly addresses **HPCforML** enabling high performance big data
- **MLaroundHPC** broadly applicable but current **use is nonuniform** across domains
  - we need to improve Cyberinfrastructure to make MLforHPC more effective for **more users**
- Use of modest DL network to **map material/potential drug structure to properties** (generalized QSAR) with simulation and observation: Advanced Progress
- Learn **surrogates** for **large scale simulations**: initial small scale results
- Use of MLforHPC in **agent-based systems** (learn agents): Very promising but few results
- **Macroscopic** Structure as in learn complex multi-particle **potentials** scaling to N^7: many great successes
- **Learn Collective coordinates and guide ensemble** computations: dramatic progress with speedups up to 10^8
- **Microscale**; learn dynamics of small scale such as clouds, turbulence: Interesting results but much more to do
- Use of **Recurrent NN**'s to represent dynamics (**learn numerical differential operators**): Promising but only studied in small problems



NSF | Indiana University | Rutgers University | University of Virginia | Kansas University | Arizona State University | Stoney Brook | University Of Utah

Community Driven High-Performance Big Data

for bio-physical applications based on HPC, distributed systems, network science, GIS and machine/deep learning

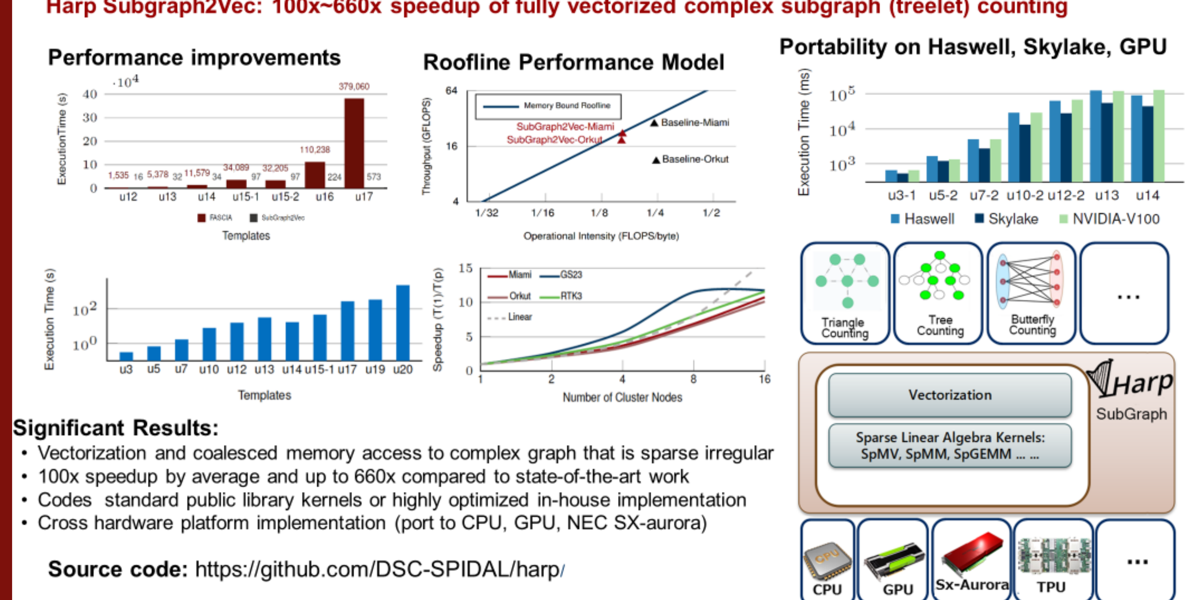
Completed Activities

- MIDAS Pilot Jobs HPC Task Management
- High Performance for Java/C++ for Machine Learning
- NIST Big Data Application Analysis: Features of data intensive Applications deriving 64 Convergence Diamonds
- HPC-ABDS: Cloud-HPC Interoperable software with performance of HPC and rich functionality of commodity Apache Stack
- Implementation of HPC and Clouds with DevOps
- Image Processing and Machine Learning Toolkit
http://hpcanalytics.org

HPC Big Data Cloud Convergence

Harp-DAAL: A HPC-Cloud convergence Data Analytics framework

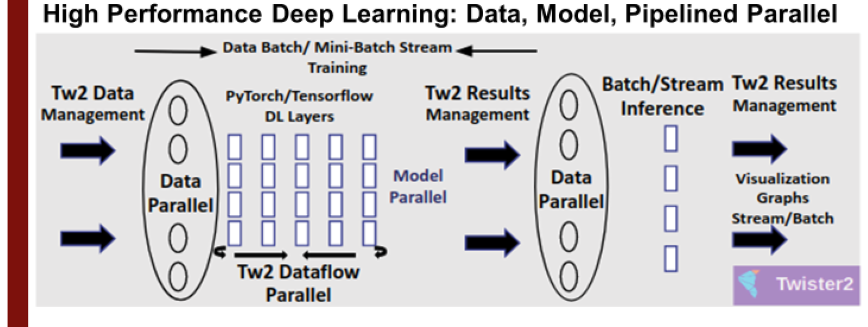
Harp Subgraph2Vec: 100x~660x speedup of fully vectorized complex subgraph (treelet) counting



Significant Results:
- Vectorization and coalesced memory access to complex graph that is sparse irregular
- 100x speedup by average and up to 660x compared to state-of-the-art work
- Codes standard public library kernels or highly optimized in-house implementation
- Cross hardware platform implementation (port to CPU, GPU, NEC SX-aurora)

Source code: https://github.com/DSC-SPIDAL/harp/

High Performance Deep Learning: Data, Model, Pipelined Parallel



Configures dataflow using Python. Dataflows can have Java, Python and CPP components connected using in-memory dataflow connections (In progress)

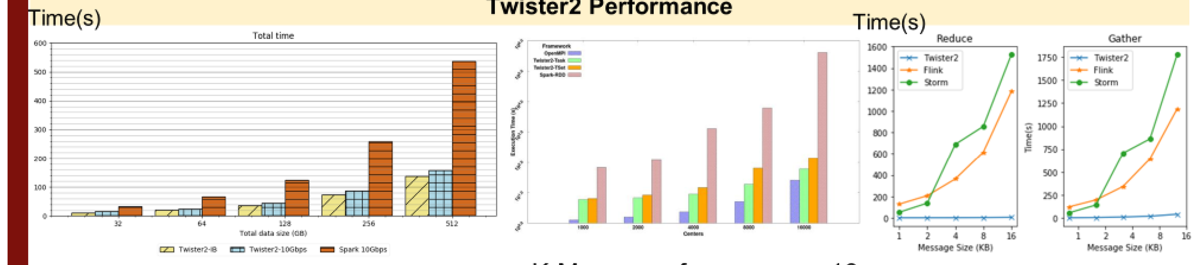
Twister2 replaces Spark and Hadoop with native high performance

Twister2 Apache Beam Runner

Apache Beam API
Twister2 Runtime

Enables high performance Beam pipelines

Twister2 Performance



Terasort performance on 16 nodes cluster with varying total data sizes

K-Means performance on 16 nodes cluster versus number of centers

Streaming micro benchmarks on 16 nodes

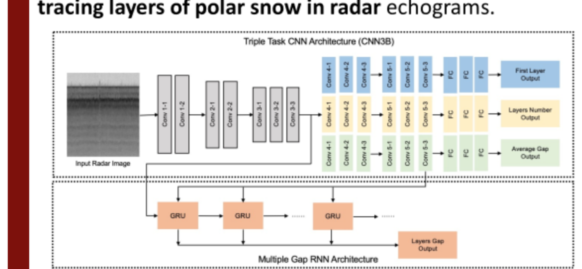
Polar Science Community

Structure of polar ice sheets from radar informatics

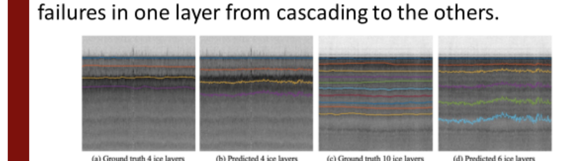
Ongoing Work

Deep learning has been successful for segmenting photos, but parsing echograms involves reasoning about weak, noisy evidence and prior information (e.g., physical constraints).

We propose a novel Convolutional Neural Network **for tracing layers of polar snow in radar** echograms.



Unlike other work, we **simultaneously** estimate the layer structure of the entire echogram, preventing recognition failures in one layer from cascading to the others.

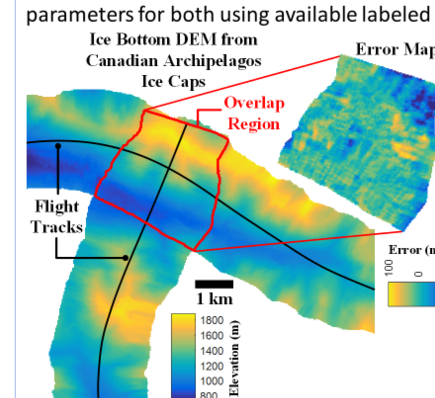


Our error is significantly lower **than baselines,** even without knowing the number of layers *a priori.*

	Mean Error (in pixels)	
	Ground Truth Numbers	Predicted Numbers
Naive CNN	29.43	-
RNN30	27.32	-
RNN256	26.20	-
CNN2B	14.93	-
CNN3B	7.91	9.27
CNN3B + RNN	7.03	8.73

Community Contributions

Our Viterbi-based layer tracker is now used routinely to process 2D radar data collected by CReSIS (50-100 TB/year), and our tree-reweighted sequential (TRW-S) surface tracker is used routinely by three groups for 3D reconstructions. Using HPC implementations of the algorithms, we have tuned parameters for both using available labeled data.



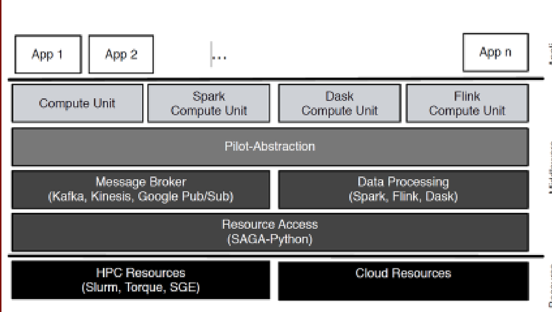
We have analyzed self consistency (flight cross overs to generate error maps) and compared performance against manually tracked layers.

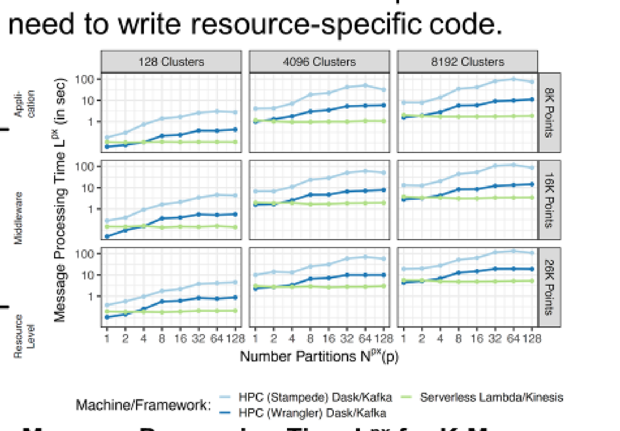
Future Community Activities
- Establish polar radar center to maintain polar radar software "toolbox" and develop ingest interfaces for more radar systems to increase access to SPIDAL tools.
- Investigate new problems: cross-track slope estimation, model order estimation, signal processing integration.
- Integrate physical model constraints into algorithms (e.g. weather models, ice flow laws, etc.)

Streaming Data Systems

Pilot-Streaming (deployed)

Pilot-Streaming provides a unified abstraction for resource management for HPC, cloud, and serverless, and allocates resource containers independent of the application workload removing the need to write resource-specific code.





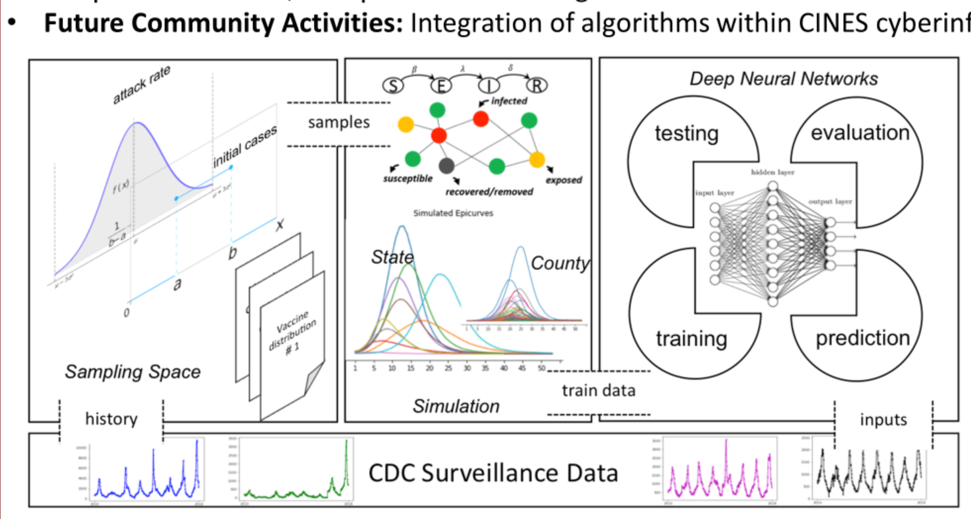
Pilot-Streaming on Serverless High-Level Architecture and Interactions: Pilot-Streaming enables the unified management of batch and streaming compute tasks on serverless infrastructure. It orchestrates the setup of Kinesis and Lambda and manages the streaming workload.

Message Processing Time L^{px} for K-Means on AWS Lambda and HPC executed via Pilot-Streaming: Broken down by # partitions, message size, and workload complexity. The processing times increase with the message size and a higher # of clusters. While for Lambda the processing times remain constant with increasing parallelism, thy increase for Dask/Kafka on HPC due to the use of shared filesystem and network resources.

Network Science Community

Network Analytics

- **Substantial Community Contributions of Project:** New advances in sequential and parallel algorithms for a broad class of network problems
  - Subgraph detection and counting, anomaly detection, dense subgraph and community detection
  - Parallel network generation: generating very large instances from many random graph models and by parallel edge switching, with different kinds of constraints
  - Applications: public health, social network analysis, scaling studies of network algorithms, sensitivity analysis, synthetic population generation
- **Ongoing Project work:** Algorithms for dense subgraphs, communities in dynamic and temporal networks, and parallel network generation
- **Future Community Activities:** Integration of algorithms within CINES cyberinfrastructure

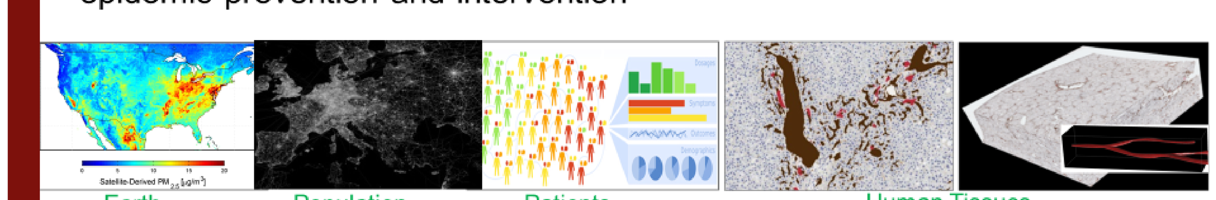


Epidemic forecasting using deep learning

Health Science Community

Spatial Big Data Query Systems

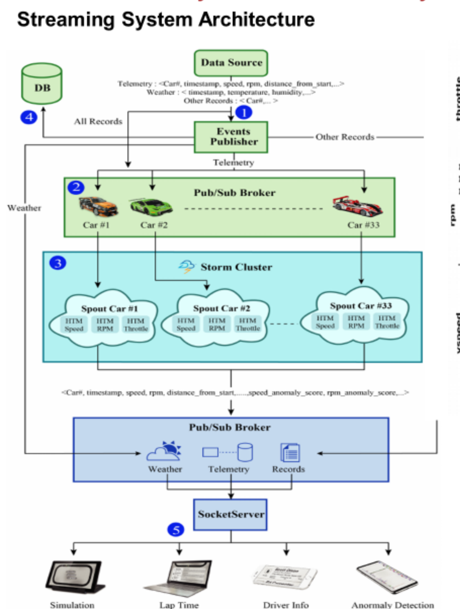
- Managing and querying big spatial data is challenged by explosion of data, multi-dimensions and the complexity of geometric computation
- Developed scalable and efficient spatial big data querying systems running on big data platforms, for both 2D and 3D
  - Hadoop-GIS, SparkGIS and iSPEED (3D)
- Integrative CPU-GPU based high performance 3D spatial queries
- The platforms can be used to support geospatial applications, big data driven public health studies, and digital pathology
- **Future:** Spatial analysis for understanding resource availability for opioid epidemic prevention and intervention



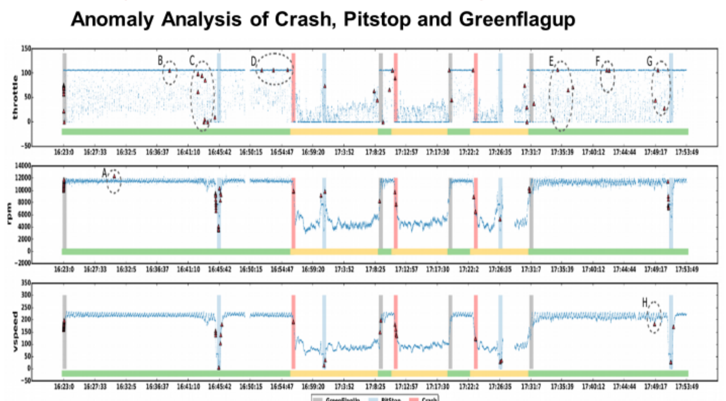
Anomaly Detection and Prediction for Streaming Data

An End-to-End Analysis in real-time for Indy500 Race Car (statistics, HTM, CNN, SSD MobileNet)

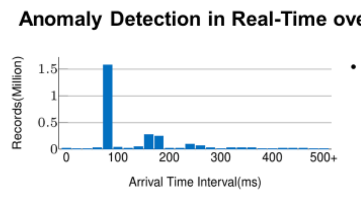
Streaming System Architecture



Anomaly Analysis of Crash, Pitstop and Greenflagup



Anomaly Detection in Real-Time over IndyCar sensor's data streams



- Data arrival rate from the telemetry is around 80~90 ms. Average latency of HTM algorithm less than 8 ms, 98% less than 20 ms and 99% less than 80 ms. The goal is to attain high accuracy, low latency and max resource utilization.

Publications
- C. Widanage, J. Li, S. Tyagi, R. Teja, B. Peng, S. Kamburugamuve, D. Baum, D. Smith, J. Qiu, and J. Koskey, "Anomaly detection over streaming data: Indy500 case study," in 2019 IEEE 12th International Conference on Cloud Computing (CLOUD), pp. 9–16, IEEE, 2019.
- S. Akkas and S. S. Maini and J, Qiu, "A fast video image detection using Tensorflow mobile networks for racing cars," in Stream Systems and Real-time Machine Learning (STREAM-ML) Workshop of IEEE Big Data Conference, IEEE, 2019.

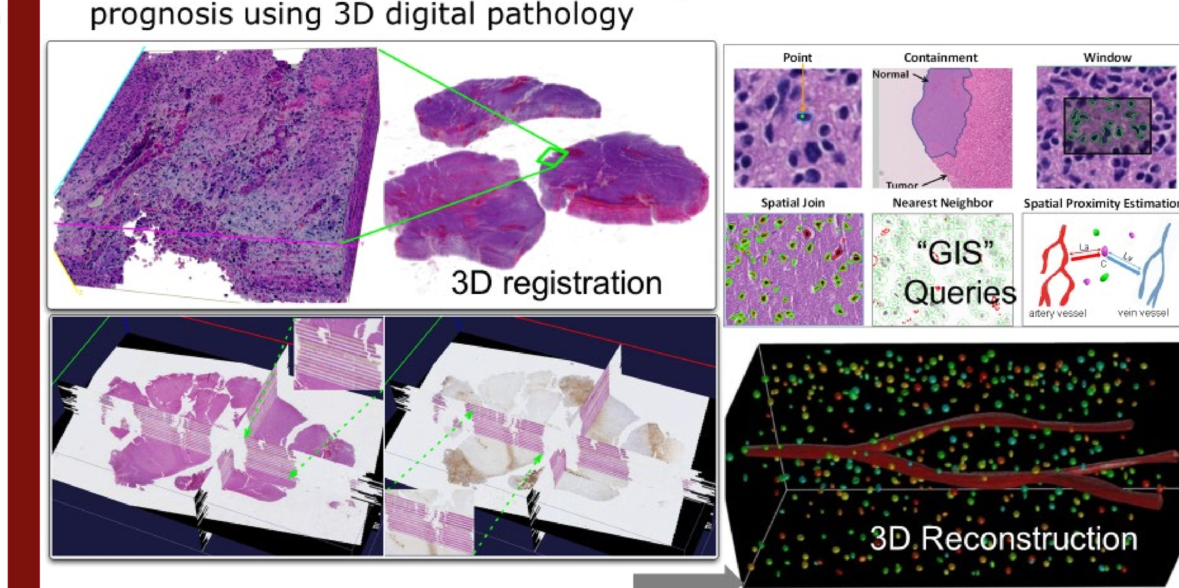
Source code: https://github.com/DSC-SPIDAL/IndyCar

Epidemic analysis and forecasting

- **Overview.** We have developed a theory/simulation guided machine learning method (TDEFSI) to forecast Influenza dynamics. Our methods provide a fundamentally new way to train and use DNNs when measured data is sparse.
  - TDEFSI produces accurate weekly high-resolution ILI forecasts from flat resolution observations. We focus on state level ILI surveillance and state (flat-resolution) or county level (high-resolution) ILI forecasts.
  - TDEFSI uses a two-branch neural network model for ILI forecasting. It combines within-season observations (observed data points of the previous weeks that characterize the ongoing epidemic) and between-season historical observations (observed data points from similar weeks of the past seasons that characterize general trends around the current week)
- **Community Impact:** Variants of TDEFSI have been used to support our ongoing participation in various forecasting challenges.
- **Current year**:
  - Extend TDEFSI and integrate in production pipelines for forecasting for Accuweather with the following extensions: (i) use it to understand the role of interventions and improved forecasts during such times, (ii) use additional high resolution data that is starting to become available to improve the forecasts and the method
  - Integrate within CINES
- Supporting the **Coronavirus outbreak** mitigation by studying questions such as: (i) risk of importation to US, (ii) possible ways it will spread in US if epidemic takes off here, (iii) possible interventions and preparedness

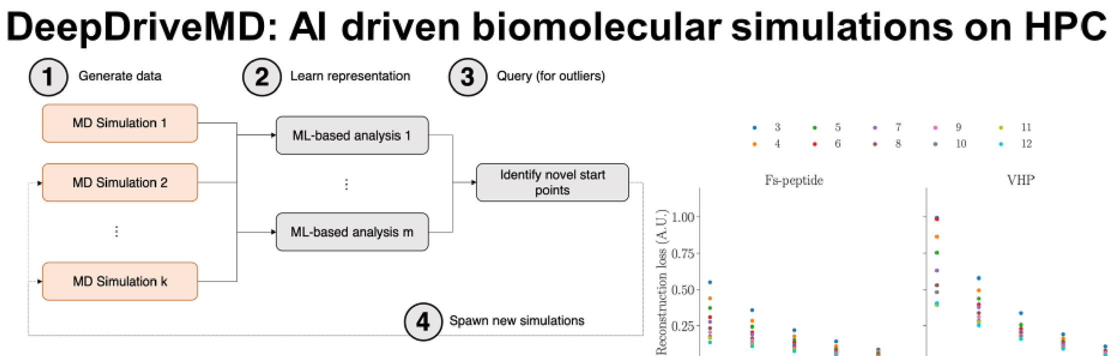
Digital pathology image analysis

- Created a suite of deep-learning based image analysis tools for level-set based image segmentation, 3D registration, reconstruction, and spatial analysis
- Future: Data analysis for understanding breast cancer treatment and prognosis using 3D digital pathology

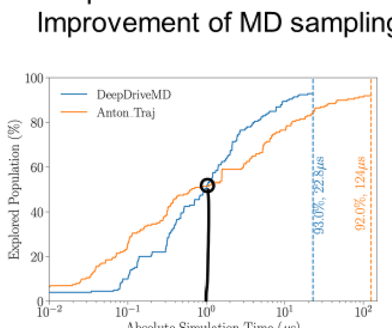


Biomolecular Science Simulation Community

DeepDriveMD: AI driven biomolecular simulations on HPC



Adaptive simulation workflow for protein folding: Improvement of MD sampling using ML (CVAE)



Future work will involve intermediate state sampling and reinforcement learning for docking scores in order to manage computational costs

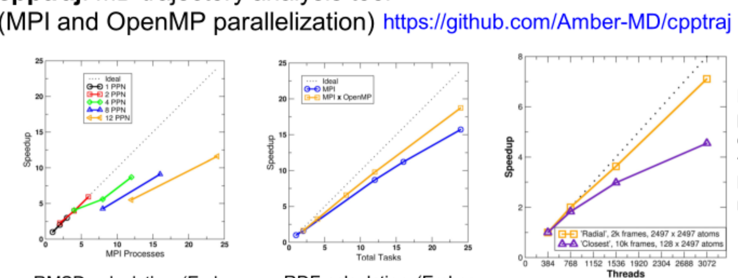
Two folding simulations (completed) show the increase in the data gathered from the number of concurrent simulations across GPUs

SUBSTANTIAL COMMUNITY CONTRIBUTIONS OF PROJECT

Released software

cpptraj: MD trajectory analysis tool (MPI and OpenMP parallelization) https://github.com/Amber-MD/cpptraj

D. R. Roe and T. E. Cheatham III. *Parallelization of CPPTRAJ enables large scale analysis of molecular dynamics trajectory data.* Journal of Computational Chemistry, 39(25):2110–2117, 2018.



RMSD calculation (Ember Cluster, U Utah)

RDF calculation (Ember Cluster, U Utah)

Neighbor calculations parallelized over ensembles and trajectories (NCSA Blue Waters XE6 nodes)

PMDA: parallel molecular dynamics analysis library (task-parallelization) https://www.mdanalysis.org/pmda/

Shujie Fan, Max Linke, Ioannis Paraskevakos, Richard J. Gowers, Michael Gecht, and Oliver Beckstein. *PMDA - Parallel Molecular Dynamics Analysis.* In, Proceedings of the 18th Python in Science Conference, pages 134 – 142, Austin, TX, 2019. SciPy. DOI: 10.25080/Majora-7ddc1dd1-013

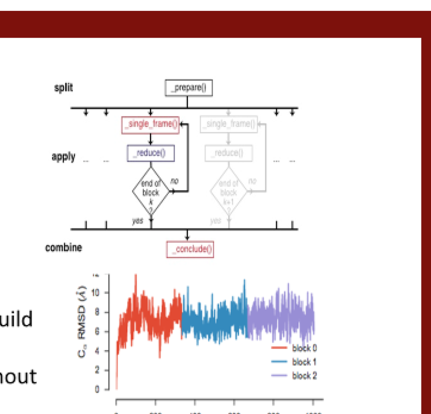
alchemlyb: general library for processing and analyzing free energy data from MD and MC simulations https://github.com/alchemistry/alchemlyb

D. Dotson, O. Beckstein, D. Wille, I. Kenney, shuail, trje3733, H. Lee, V. Lim, B. Allen, and M. S. Barhaghi. alchemistry/alchemlyb: 0.3.1, Jan. 2020. DOI: 10.5281/zenodo.3610564

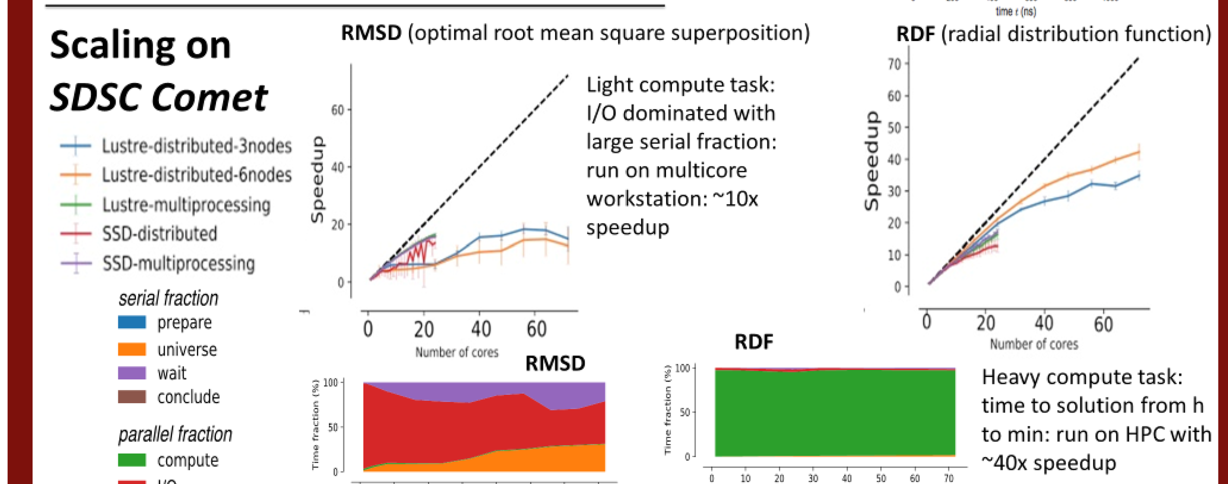
Parallel Molecular Dynamics Analysis (PMDA)

https://www.mdanalysis.org/pmda/

- Python package (release 0.3.0, GPL v2 license)
- Parallelizes widely used **MDAnalysis** Python package for MD trajectory analysis
- *split-apply-combine* trajectory analysis with *task-based parallelization*: build and executed directed acyclic graph with *Dask* (https://dask.org)
- Flexible: runs on multicore laptops/workstations to supercomputers without changes



Scaling on *SDSC Comet*



Light compute task: I/O dominated with large serial fraction: run on multicore workstation: ~10x speedup

Heavy compute task: time to solution from h to min: run on HPC with ~40x speedup