

Portals for interactive steering of HPC workflows

SC19 Workshop on Interactive High Performance Computing

Robert Settlage*¹ and Srijith Rajamohan*¹

Kevin Lahmers², Alan Chalker³, Eric Franz³, Steve Gallo⁴, David Hudak³

*presenting

¹Advanced Research Computing at Virginia Tech,

²Virginia Maryland College of Veterinary Medicine,

³Ohio Supercomputer Center,

⁴Center for Computational Research, University of Buffalo

Supported by National Science Foundation grant 1835725.

- 1 Introduction**

 - Today's goals
 - iHPC vs Gateway vs HiL

- 2 Project 1: enabling non-traditional HPC users**

 - Herd health monitoring via DNA sequencing
 - Barriers to use of HPC resources
 - Open OnDemand as an enabling tool

- 3 Project 2: Visualizing Ideology using Weakly-Supervised Learning**

- 4 Thoughts and conclusions**

Show case work at Virginia Tech where HPC, human interaction and portals/gateways have converged to enable interesting science.

Project 1: enabling complex workflows with Open OnDemand

Herd health monitoring via DNA sequencing.

Project 2: iHPC workflow for Weakly-Supervised Learning

Visualizing Ideology using Weakly-Supervised Learning.

In this discussion, we are really combining three concepts: interactive computing, gateways, and human-in-loop computing.

iHPC

Combination of HPC and human interaction to influence computational workflows.

Gateway

Portal of entry to compute resources.

HiL

Computation requiring human interaction integral to computation.

Detection and genotyping of veterinary viral pathogens

Genome sequencing is more accessible, portable, and becoming the go-to for screening. Here we are enabling vets by building tools for HPC from the edge.

Porcine Reproductive and Respiratory Syndrome virus in Pigs

- Annual losses in US of \$760M
- Genotyping important for animal movement and treatment decisions

Infectious Bronchitis virus in Chickens

- Similar to PRRSv in variability and vaccination efficacy

Equine Herpesvirus-1 in Horses

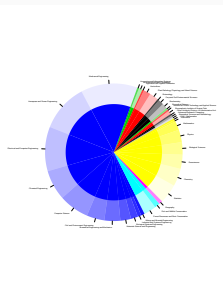
- Highly contagious
- Quarantine released based on genotype identification

High Performance Computing Barriers

Availability of hardware is not an (immediate) issue.

Access and use barriers are largely self-imposed.

- System access: ssh
- Software: no root access, modules
- Data (in/out): ftp, scp, rsync, etc
- Compute configuration, script writing: vi, emacs, etc
- Compute execution: job scheduling



Open, interactive HPC via the web.

Provides easy to use and extend, web-based access to HPC.

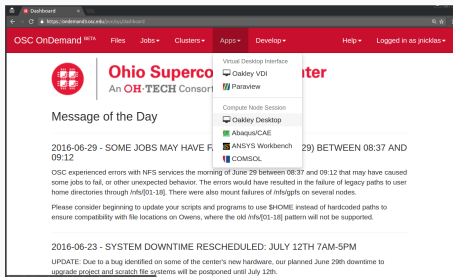
Features:

- Plugin-free web experience
- Easy file management
- Command-line shell environment
- Job Management and monitoring
- Graphical desktop environments and applications

Users come with a modern web browser and HPC credentials.

Open OnDemand provides zero-install and single sign-on solution.

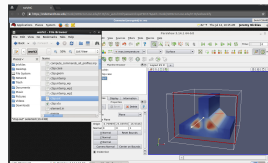
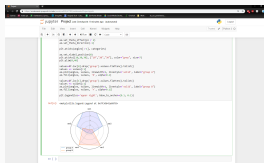
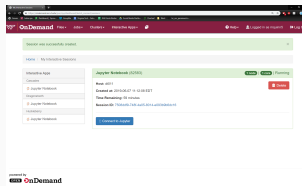
- Landing page
- Files App
- Job Composer App
- Job Monitor

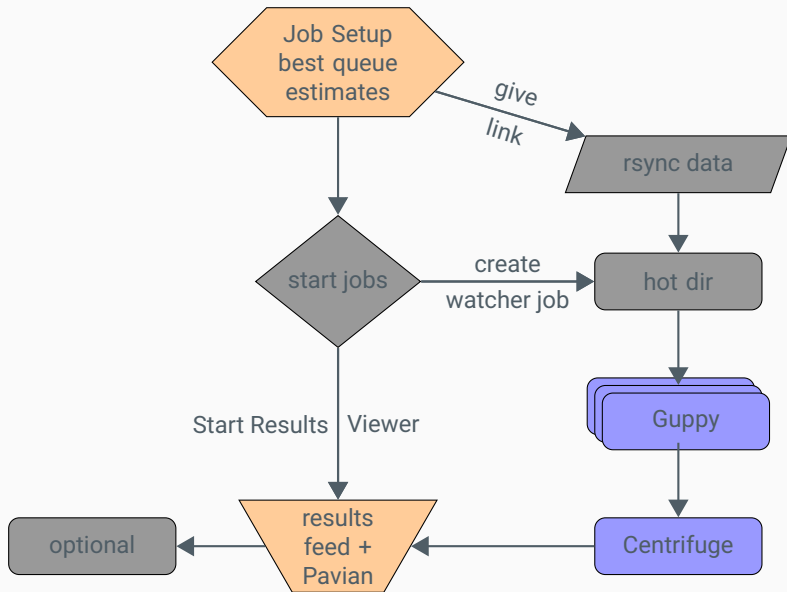


OnDemand uses a plug-in style wrapper to facilitate app development.

Users and sites can develop and share custom apps.

- Jupyter Notebooks
- Matlab
- Rstudio
- ParaView, Comsol, etc



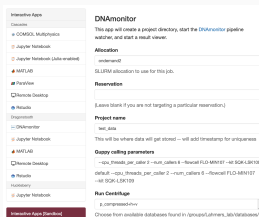


Open OnDemand Pathogen detector job form

Job form abstracts away all unessential scheduler settings.

WIP – add estimates for time to first results and project completion based on queue parameters, job characteristics and current queue jobload.

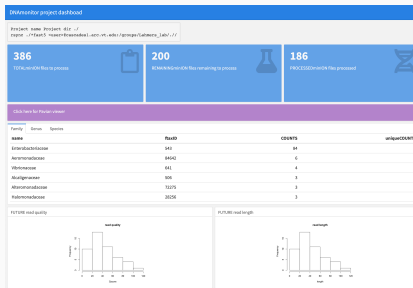
- keep form to basics
- inform choice of queues
 - GPU queue gives faster time per job
 - CPU queue may be faster to complete full project
- end result is submission of job
 - watcher for waiting for data upload
 - viewer for real time viewing of results as completed
- typical sequencing run gives 300-1000 sequencing files taking 1 min (GPU) to 10 min (CPU) to process



MinION sequencing results in many small fast5 files.

MinION -> HPC storage -> Guppy -> Centrifuge -> additional steps.

- status for project metrics
- link to Pavian viewer
- add logic for additional steps
 - blast
 - send data to db
 - kill all??
 - more ...



Open OnDemand Pavian results viewer

Pavian is a Shiny App (R).

VT-ARC enables R-Shiny through Singularity.

- Pavian results viewer
- results inform treatment
- results determine next steps

Pavian metagenomics data explorer

This page shows the summary of the classifications in the selected sample set. The cells have a bar chart that shows the relative of the cell value to other cell values in the same category, with the absolute values being a separate category from the rest.

Classification summary [Raw read numbers](#)

Show 10 rows | [CDS](#) | [Prot](#) | [Copy](#) | [Column visibility](#)

Items	Number of raw reads	Classified reads	Chordate reads	Artificial reads	Unclassified reads	Microbial reads	Bacterial reads	Viral reads	Fungal reads	Protocoon reads
PT1	12,022,204	98.4%	93.3%	5.80%	0.580%	0.014%	0.012%	0%	0.000607%	0.000225%
PT2	8,296,305	98.9%	99.2%	0.749%	0.146%	0.00067%	0.0017%	0%	0%	0%
PT3	17,600,844	98.7%	96.2%	3.43%	0.339%	0.0462%	0.0380%	0.000017%	0.000328%	0.000320%
PT4	29,033,719	98.8%	97.8%	2%	0.118%	0.00033%	0.00376%	0%	0.000013%	0.000122%
PT5	26,010,865	98.2%	96.2%	2.32%	0.730%	0.054%	0.0590%	0.0333%	0.000308%	0.000387%
PT6	27,281,708	98.9%	97.8%	1.84%	0.373%	0.0111%	0.033%	0.0000624%	0.0000403%	0.000011%
PT7	50,005,374	98.2%	70.8%	28.0%	0.773%	0.071%	0.040%	0.000380%	0.0000020%	0.000021%
PT9-11	36,931,034	98.8%	97.8%	1.84%	0.334%	0.000369%	0.000369%	0.0000000%	0.0000000%	0.0000000%
PT5-12	35,095,203	98.8%	95.7%	0.6009%	0.334%	0.000370%	0.000370%	0.0000112%	0.0000101%	0%
PT9	36,006,814	98.7%	96.8%	0.881%	0.246%	0.0102%	0.00774%	0.0000377%	0.000004%	0%
PT10	21,215,274	98.8%	98.8%	1.01%	0.331%	0.00480%	0.00337%	0.0000938%	0.000044%	0%

Showing 1 to 11 of 11 entries

LTZ Explore identifications across all samples in the Sample Comparison View.

Previous Next

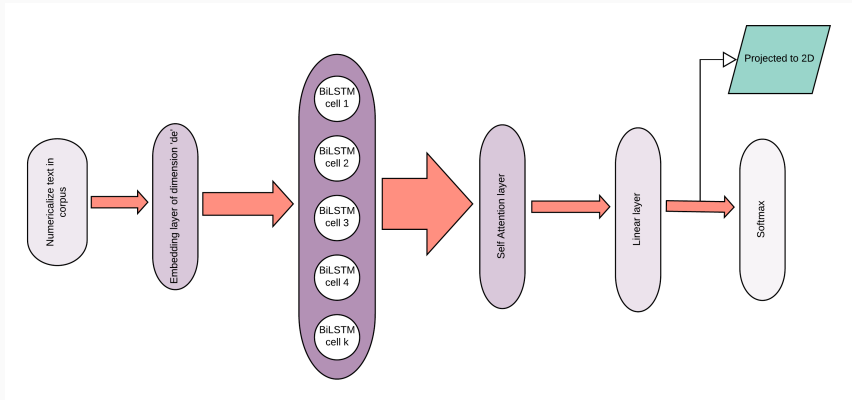
Visualizing Ideology using Weakly-Supervised Learning

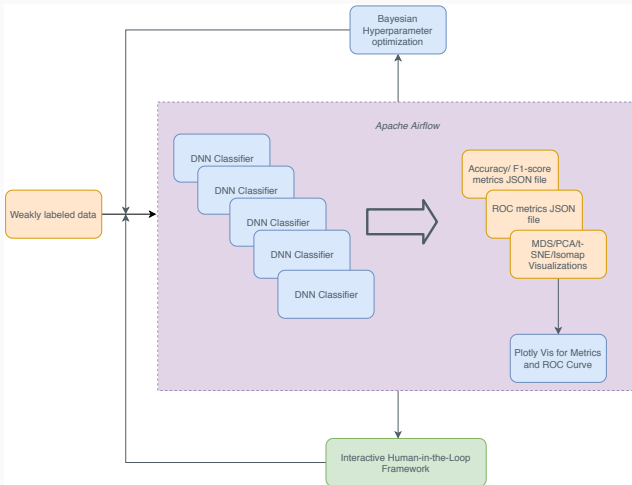
- **Natural Language Understanding**
 - Bidirectional LSTMs with static and contextual embeddings
- **Augmented Intelligence**
 - Human-in-the-loop: Results of DNNs used to create projections that assist humans in classifying documents
- **Interpretable**
 - Self-attention gives insight into the decision-making process of a DNN
- **Evaluation of visualization techniques**
 - tSNE, PCA, MDS, Isomap

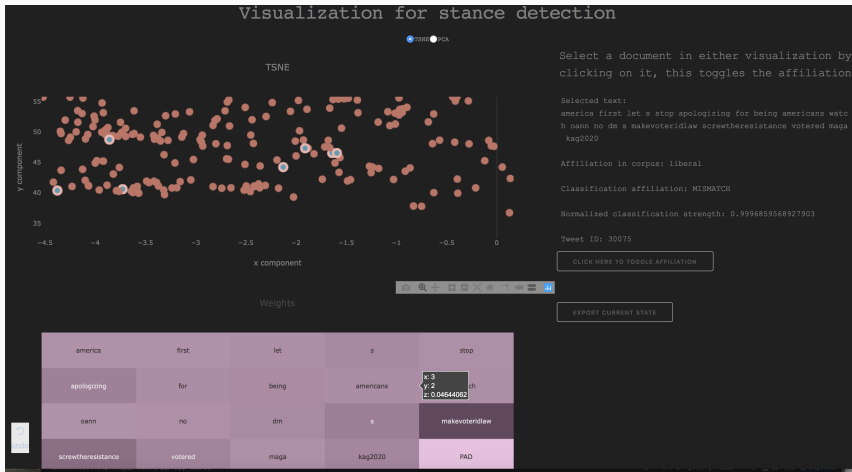
<https://arxiv.org/abs/1908.02282>

- Pytorch code that runs on GPUs
- Run on a Dell C4140 with 4 Volta GPUs with 16GB per-GPU memory
- Workflow automated with Airflow
- Hyperparameter optimization done with Comet.ml
- Group Python packages into conda environments to mitigate conflicts

- Gather social media posts related to certain political hashtags, along with user metadata
- Use PySpark/Spacy to clean and normalize data
- Create 2D projections of document affiliation using the DNN
- Interactive web application that displays the projections to help correctly label this weakly-supervised corpus
- A type of Active Learning
- Evaluate visualization/cognitive efficiencies of various visualization types







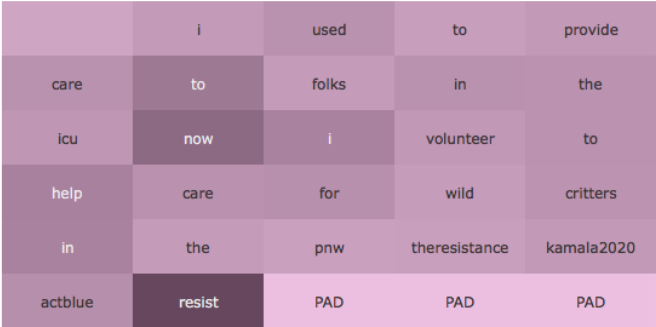
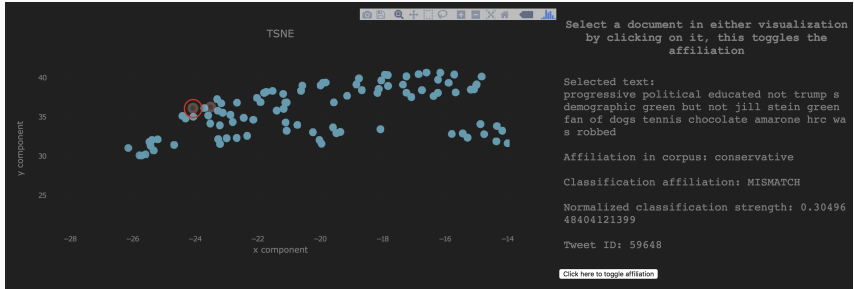
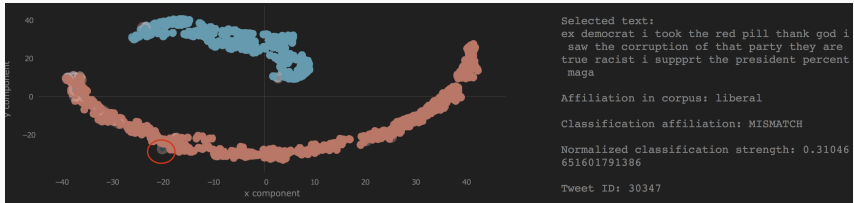


Figure: Heatmap of Attention weights

Incorrect labels



Plots

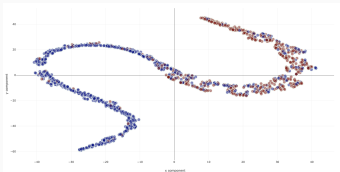


Figure: tSNE



Figure: PCA

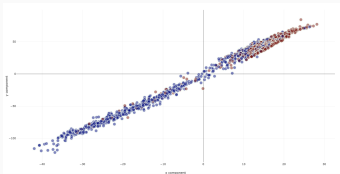


Figure: MDS

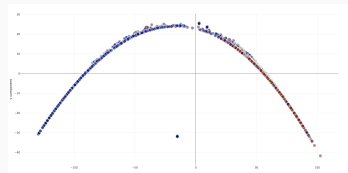
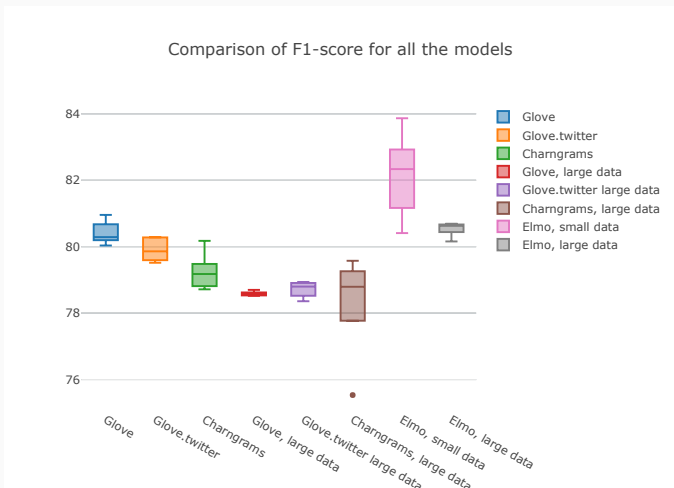


Figure: Isomap

Figure: F1 scores for different embeddings



Questions?