Portals for Interactive Steering of HPC Workflows

Robert Settlage[1(✉)], Srijith Rajamohan[1], Kevin Lahmers[2], Alan Chalker[3], Eric Franz[3], Steve Gallo[4], and David Hudak[3]

[1] Advanced Research Computing, Virginia Tech, Blacksburg, VA 24060, USA
rsettlag@vt.edu
[2] Virginia Maryland College of Veterinary Medicine, Blacksburg, VA 24060, USA
[3] Ohio Supercomputer Center, Columbus, OH, USA
[4] University of Buffalo, Buffalo, NY, USA

Abstract. High performance computing workloads often benefit from human in the loop interactions. Steps in complex pipelines ranging from quality control to parameter adjustments are critical to the successful and efficient completion of modern problems. We give several example workflows in bioinformatics and deep learning where computing decisions are made throughout the processing pipelines ultimately changing the course of the compute. We also show how users can interact with the pipeline using Open OnDemand plus XDMoD or Plot.ly.

Keywords: HPC · OnDemand · XDMoD · Steering · Workflow · Deep learning · Bioinformatics

1 Introduction

The need and scale of computing requirements continue to grow. Daily, we are collecting zettabytes of data which requires computing to transform it into knowledge [1] and actionable insights. As with the data, the associated data analysis pipelines and simulations have grown in size and computational complexity often needing large cluster based computing approaches such as those enabled by high performance computing (HPC) clusters. Traditionally, high performance computing (HPC) workloads have consisted of a series of static shell scripts that are run via the command line. Changes to the pipeline, i.e. scripts, are manual, error prone, and not usually intuitive. For interactive workflows, command line intervention is not desirable and has limited adoption within mainstream HPC computing pipelines.

As an endpoint, we are looking to create user friendly, intuitive and resilient tools for use of HPC resources to handle arbitrary and potentially complex computational pipelines. As the pipelines grow in size and complexity, the computational workflows may need branching or other decisions made mid-stream.

Supported by National Science Foundation grant 1835725.

G. Juckeland and S. Chandrasekaran (Eds.): HUST 2019/SE-HER 2019/WIHPC 2019,
CCIS 1190, pp. 179–189, 2020.
https://doi.org/10.1007/978-3-030-44728-1_11

These decisions could be as simple as determining where the next step should run to get to result faster or as complex as clustering for data classification. Bioinformatics and Deep Learning are two compute heavy domains where computing pipelines with potentially complex workflows could benefit from some human intervention at run time.

The Big Bang in analytical and discovery biology could be viewed as the rise of -Omics data acquisition technologies. Today it is possible, in a single hour, using modern DNA sequencing technology to sequence the genome of hundreds of bacteria simultaneously, or using mass spectrometry, collect full metabolite profiles from humans, or using protein array technology, collect the entire human phospoproteome profile. Analyzing and making sense of all this data is a massive computational challenge. By using a combination of web forms and GUI apps in Open OnDemand complete with job and cluster status statistics provided by XDMoD, we are endeavoring to make our clusters more accessible and efficiently utilized. Here we highlight two workflows we are designing to chaperone the data from raw to result with the aim of making the workflows computationally efficient and easy to use.

Open OnDemand [2] and XDMoD [3] are open source projects with overarching goals of improving the accessibility and usage of HPC clusters. Historically, HPC cluster access has been limited to command line access via ssh. Compute jobs require creation of shell scripts and interaction with schedulers that are often unfamiliar to most new users. Combined, these normal HPC modes of operation have created barriers to use of the computational power contained in the clusters. Open OnDemand provides a rich set of browser based tools and apps to facilitate use of HPC clusters through more familiar interfaces. For instance, OnDemand has a browser based files app that allows users to interact with the HPC file system through a graphical interface. In addition to the native apps included in the standard OnDemand installation, Open OnDemand allows creation of custom user apps. XDMoD focuses more on the performance and utilization of HPC resources. XDMoD gives administrators and users tools to gauge how well the system and their jobs are running. Open OnDemand and XDMoD, combined, give users a unique set of interfaces and tools to access and utilize HPC clusters.

In traditional supervised Deep Learning, subject matter experts are required to provide accurately labeled data for training and evaluation of the model. However, this is usually either intractable or extremely labor-intensive whereas it is far easier to procure labelled data with labels that are only partially accurate. This type of weak supervision is referred to as inaccurate supervision [4]. Airflow [5] was used to orchestrate this decision-making, i.e. learning and post-processing, workflow. Airflow is built on the notion of tasks as Directed Acyclic Graphs (DAGs) which can be scheduled. Tasks can have dependencies and be restarted as needed thereby providing a repeatable pipeline for the interactive Visual Analytic framework based on Deep Learning presented here.

2 HPC Workflows

2.1 DNA Monitoring

As we move to personalized medicine and genomics, we will see more edge devices for data collection with data streams being sent to HPC clusters for efficient and timely data analysis. As an example, minION sequencers are being used in field monitoring of live stock health. In this scenario, the sequencer is brought to the farm, sequencing is performed locally, data is streamed to an HPC cluster, a perhaps complex and computationally intensive compute job is performed and near real time updates are provided via web portals to the veterinarian on-site.

In a simplistic case, this is a monitoring problem. During an outbreak, there could be additional data processing steps necessary. Here, we are only concerned with monitoring where ease of use by lab personnel and time to results is the primary interest. Using a combination of Open OnDemand and XDMoD, we are creating an interactive portal for choosing where (cluster and queue) to submit processing jobs based on cluster utilization and job performance statistics, see Fig. 1. The workflow includes DNA basecalling through the minION sequencer software (Guppy), bacterial DNA assignment and counting (Centrifuge [6]) and results viewing through Pavian [7]. Currently the portal includes a job submission form, job progress app and link to Pavian results viewer. As the portal matures, additional optional possibilities for data processing will be exposed and enabled as options in data review. The optional possibilities could include submitting additional processing jobs, pushing the data to archive, pushing the results to a database, etc. (Fig. 2).

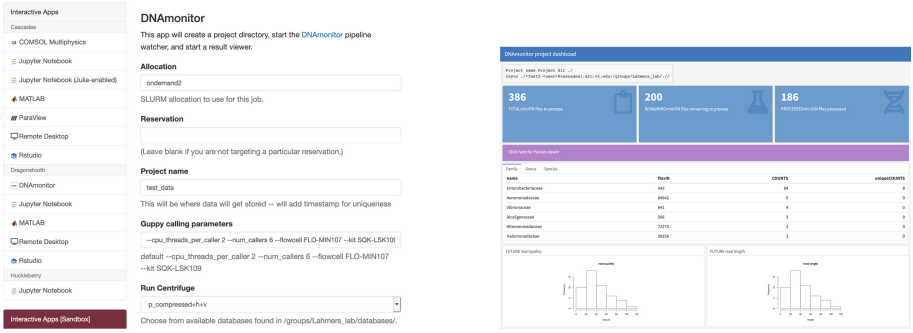

Fig. 1. DNAmonitor main page and job progress viewer with Pavian results link.

Open OnDemand and XDMoD App. Open OnDemand [2], through familiar web-based access to HPC clusters, reduces barrier to use of HPC resources and has also been shown to reduce the time to science. In fact, the median time from initial login to first job submission for all new OSC clients in 2017 using OnDemand was 10 times faster than those using traditional access methods.

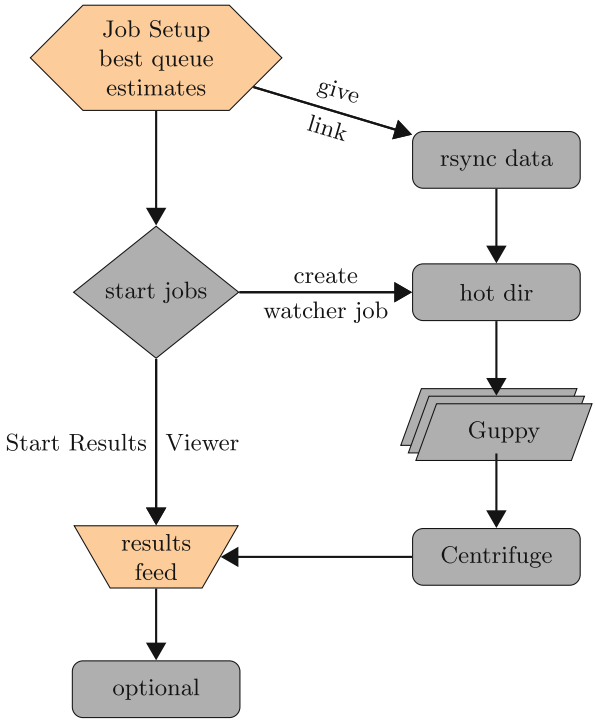

Fig. 2. Basic minION Guppy-Centrifuge DNA monitoring workflow. User interaction is indicated in orange. (Color figure online)

OnDemand greatly simplifies access to HPC resources, freeing disciplinary scientists from having to worry about the operating environment and instead focus on their research. Here, we will use OnDemand to assist users in pipeline setup and GUI access for data viewing.

Extensibility is a key component of the Open OnDemand App architecture by allowing for creation of *custom* applications. Apps can be developed to surface common GUI programs such as R, Shiny, Jupyter Notebooks etc. Additionally, apps could be developed as web forms as common job submission templates. Here we are looking to use both modes. First, we have created an app for job submission. The app takes in metadata related to the project (name, date, etc.), approximate sequencing run time, flowcell type, and gives users informed choices for where to run the analysis based on current HPC usage. Ideally, we will use both user based job statistics for jobs of similar size for the various queues the user has access to and system benchmarks for the codes used. This information will come from XDMoD [3]. Second, we are using OnDemand to surface a GUI (R Shiny) for viewing results as they are made available by the compute pipeline. This app updates as new results are available and allows users to influence future behavior of the full pipeline.

2.2 Basic NGS QC

We will start with a very basic workflow representing a common quality control (QC) step performed on Next Generation Sequencing (NGS) data as a starting point for more complete analysis including RNASeq discussed in the next section. In this case, the workflow consists of steps for data input, splitting the data for processing (more useful in later steps), raw data metric collection, and review of the collected metrics. The user interacts with the HPC clusters to define the inputs, desired cleaning steps and finally to look at the metrics related to the QC as show in Fig. 3. The form used to start the workflow includes the ability to specify which HPC queue will run the analysis as in the DNA monitoring workflow.

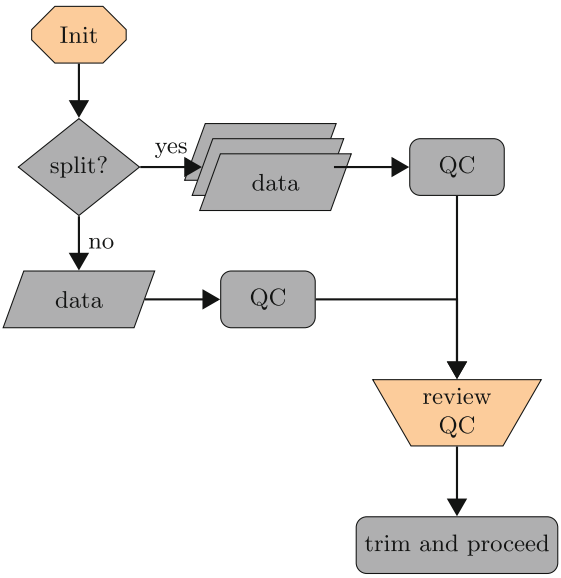

Fig. 3. Basic NGS QC workflow. User interaction is indicated in orange. (Color figure online)

2.3 RNASeq

Our RNASeq pipeline is really an extension of the NGS QC workflow. User intervention is enabled between major steps and includes options for failing samples and alternative RNASeq specific methods. Additional details and examples will be given in the session.

2.4 Weakly Supervised Deep Neural Network

The goal of this work [8] is to assess the feasibility of a weakly supervised Deep Neural Network (wsDNN) to produce projections for determining political affiliations. In a way, this can be seen as a type of target-based sentiment analysis

also known as Aspect-Based Sentiment Analysis (ABSA) [9]. This is a form of 'inaccurate weak supervision' due to the presence of errors in the labels of the data used for training.

The data was downloaded periodically using the job scheduler RQ [10] for three months. The downloader was written using the Python library Tweepy [11] and around 2.6 million tweets were downloaded over this period. The tweet information was stored in a MongoDB [12] database with a Metabase [13] interface for database visualization and querying. This is the data acquisition stage. Once this step has been completed, the data is preprocessed and cleaned using a combination of PySpark and Spacy [14] to get a cleaner corpus for training purposes.

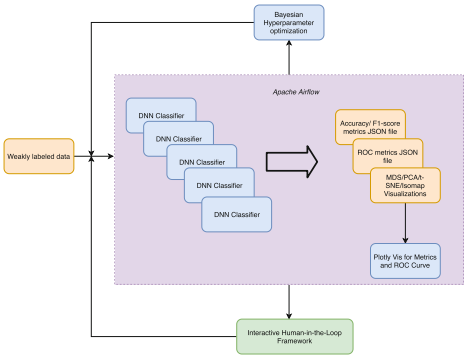

Fig. 4. Weakly supervised DNN workflow. User interaction is in green where reclassification of data points is enabled. (Color figure online)

The data, obtained after preprocessing, goes through a training and feedback loop as shown in the Fig. 4. A DNN based on static and contextual embeddings coupled with Attention mechanisms is trained on the corpus and is evaluated to produce metrics such as F1-Scores, Accuracy, ROC curves along with dimension-reduced projections as outputs. Visualizations are produced for the model metrics using the Plot.ly [15] framework from the several model runs. All of this is automated using the Airflow [5] tool.

The dimension-reduced projections are inspected by a human to assess model performance as described in [8]. Also, hyperparameter optimization is performed on the various configurations, as shown in Fig. 4, to identify the best-performing model using the Comet.ml [16] tool. The combination of the Human-in-the-loop process and the hyperparameter optimization process serves to iteratively refine the model.

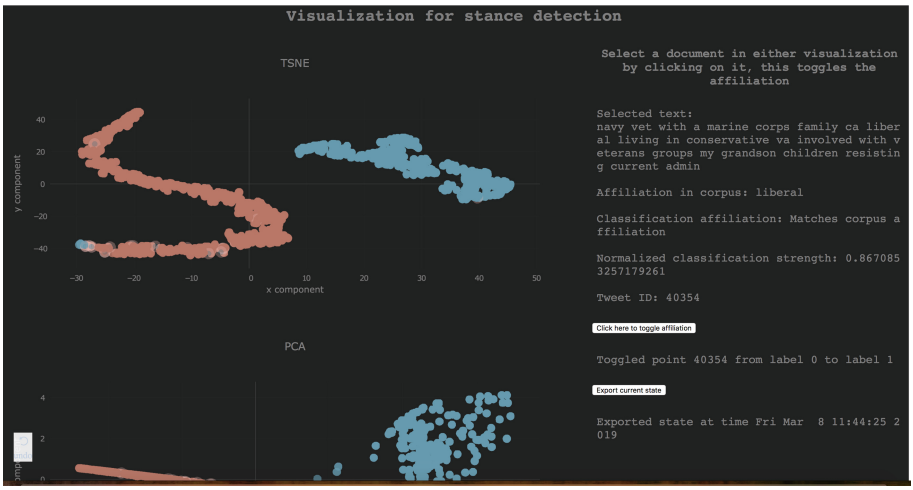

Fig. 5. Framework for stance detection

Interactive and Exploratory Web-Based Application. The interactive web-based application (Fig. 5) that is created can be divided into two halves: the left half presents the projections and the right half presents the information associated with an entity selection. To interact with the visualization the user can single click on an entity in the visualization to select it, or the user can use any of the pan/zoom/select tools to explore the visualizations. The top projection is the result of the application of t-SNE to the output of the penultimate layer and the bottom projection is the MDS projection of the same output with euclidean distance functions.

Model Interpretability. With the prevalence and success of the predictive power of DNNs, it has also faced criticisms over how the results were generated. This has accelerated efforts to provide a solution to this concern, which is informally referred to as Interpretable AI. While the application shown in Fig. 5 allows us to assess stance with a measure of uncertainty, how this determination was made is not transparent to the user.

An attention layer takes as input a 'context' and 'query' and computes the similarity of the query vector to each vector in the context matrix. In self-attention, the context and query are the same and one computes the similarity of each word in the sequence to every other word in this sequence to form an attention weight matrix. The attention weights can be used to visualize the relevance of each individual word in a sentence with respect to its classification. An example of this is illustrated in Fig. 6. The emphasized words as indicated by the darker boxes have a larger contribution to the classification outcome, thereby informing the user what words are relevant from the network's perspective.

Fig. 6. Illustration of Attention weights for model interpretability

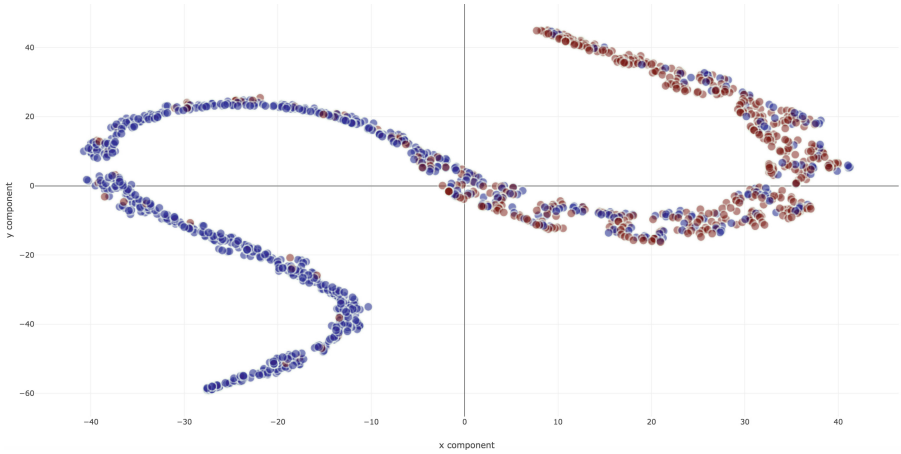

Fig. 7. Structure of t-SNE projections from the penultimate layer

Evaluation of Visualization Methods. Projections from the penultimate layer are dimension-reduced using PCA, MDS, Isomap and t-SNE to evaluate the suitability of these methods for representing the results of the networks and assessing political affiliation. MDS (Fig. 8), as a result of the nature of the projection allows quantification of stance as a function of 'distance' along the direction of the projection. t-SNE (Fig. 7) tended to reduce crowding. However, Isomap (Fig. 9) can be seen as a better technique for capturing non-linear relationships in the high-dimensional data.

Hyperparameter Optimization. Hyperparameter optimization was performed using the feature provided by the Comet.ml tool. This framework allows, through the use of APIs, the logging of model metrics and the optimization of the required hyperparameters over a desired metric such as accuracy or loss. Figure 10 shows the web interface for Comet.ml where one can inspect each pass

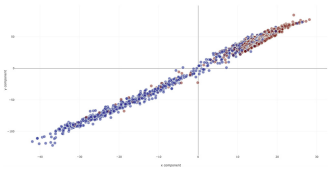

Fig. 8. Structure of MDS projections from the penultimate layer

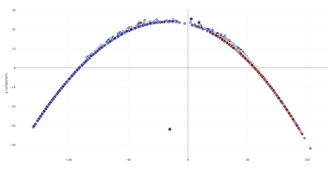

Fig. 9. Structure of Isomap projections from the penultimate layer

(0)	Status	Visible	Name	Tags	Server ...	File na...	Duration	HIDDE...	DROPO...	LINEA...	Loss
			75ba3c1...	three	8/6/19 06:2...	attention_p...	00:01:09	75	0.21775777...	55	0.02242653...
			65d2790...	three	8/6/19 06:2...	attention_p...	00:01:02	10	0.32994308...	54	0.01715301...
			17b9e7e...	three	8/6/19 06:2...	attention_p...	00:01:08	102	0.50401669...	12	0.01860596...
			53fc961...	three	8/6/19 06:2...	attention_p...	00:01:03	48	0.96513551...	41	0.33428800...
			dfd1928...	three	8/6/19 06:2...	attention_p...	00:01:14	117	0.62109994...	35	0.02168722...
			df47395...	three	8/6/19 06:2...	attention_p...	00:01:09	32	0.29615986...	36	0.01531454...
			b5658b0...	three	8/6/19 06:3...	attention_p...	00:01:09	55	0.19085956...	48	0.00567017...
			8c68c45...	three	8/6/19 06:3...	attention_p...	00:01:05	24	0.24194129...	36	0.01355262...
			cdde3c7...	three	8/6/19 06:3...	attention_p...	00:01:07	82	0.16072706...	53	0.00628004...
			ad0941e...	three	8/6/19 06:3...	attention_p...	00:01:03	17	0.14312883...	40	0.00561240...
			24c8abb5f	three	8/6/19 06:3...	attention_p...	00:01:06	71	0.44517077...	42	0.01454407...
			549298e...	three	8/6/19 06:3...	attention_p...	00:01:11	73	0.32904065...	18	0.01799950...
			2d54721...	three	8/6/19 06:3...	attention_p...	00:01:10	110	0.48485643...	33	0.02082771...
			e141ae5...	three	8/6/19 06:3...	attention_p...	00:01:06	77	0.51328984...	57	0.01738352...
			346d1ee...	three	8/6/19 06:4...	attention_p...	00:01:05	62	0.14503871...	19	0.00693153...
			6aca03e...	three	8/6/19 06:4...	attention_p...	00:01:04	35	0.09961353...	15	0.00363940...

Fig. 10. Hyperparameter optimization with Comet.ml

of an optimization run. Figure 11 demonstrates visual filtering of the various passes with an interactive parallelogram chart that can be built using the tool. This allows the user to identify optimal hyperparameter configurations for the corpus.

Workflow. The purpose of this work was to determine how well the Bidirectional LSTM (BiLSTM) networks with various static embeddings perform compared to the same networks with pretrained contextual embeddings. In this work 'Elmo' [17] was used for the contextual embedding. For static embeddings, the 100-dimensional Glove embeddings were chosen as a tradeoff between expressiveness and availability of compute and memory resources. Along with the Glove

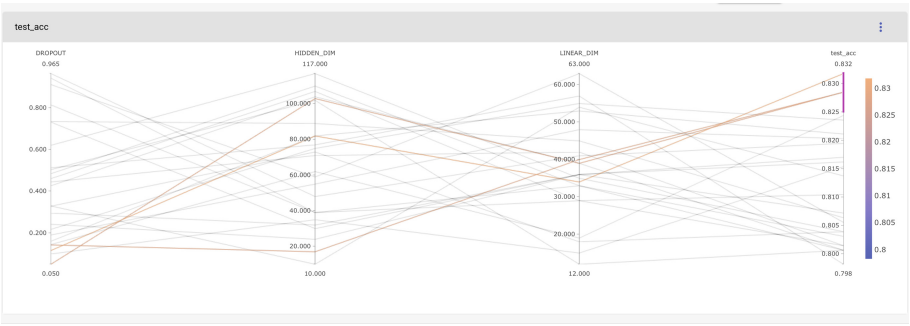

Fig. 11. Parallelogram with filters to identify optimal hyperparameter configurations

embeddings, the 'Glove.twitter.100d' embeddings and 'Charngram.100d' embeddings were also evaluated.

In order to evaluate these models, Airflow was used to automate the workflow associated with gathering metrics. Training and validation accuracy, precision, recall and F1-scores on a larger set of weakly-supervised data is noted along with the same metrics on the fully-supervised smaller test data. As a result of the class imbalance in our training data, it is critical to evaluate performance using all of the metrics above. ROC metrics are also recorded per configuration for each run so that an ROC curve can be generated. The raw metric files are processed by a script that generates the metric plots, i.e. box plots of accuracy, precision, recall and F1-scores, to compare model performance. The data files required to generate the visualizations are written out as pandas dataframes. These files are then read by the framework shown in Fig. 5 for interactive exploration. The users have the ability to interact with the documents in the corpus through the application. This allows them to correct the labels through inspection if it is deemed necessary, and export the corrected corpus back so that it can be fed back into the DNN for iterative refinement.

3 Conclusion and Future Work

In summary, case studies were presented that illustrated the use of HPC in knowledge mining. The availability of such resources expedited the iterative feedback loop necessary for the use-cases presented above. The work on the portals presented here lowered the barrier for access to HPC resources as well as increasing research productivity thereby effectively reducing the mean time to discovery and opening up HPC resource availability and use to more fields of science.

References

1. Reinsel, D., Grantz, J., Rydning, J.: The Digitization of the World - From Edge to Core. IDC White Paper - #US44413318 (2018)

2. Hudak, D., et al.: Open OnDemand: a web-based client portal for HPC centers. J. Open Source Softw. **3**(25), 622 (2018)
3. Palmer, J.T., et al.: Open XDMoD: a tool for the comprehensive management of high-performance computing resources. Comput. Sci. Eng. **17**(4), 52–62 (2015)
4. Zhou, Z.H.: A brief introduction to weakly supervised learning. Nat. Sci. Rev. **5**(1), 44–53 (2017)
5. Apache Airflow Documentation: Apache Airflow Documentation - Airflow Documentation. https://airflow.apache.org/. Accessed 13 Sept 2019
6. Kim, D., Song, L., Breitwieser, F.P., Salzberg, S.L.: Centrifuge: rapid and sensitive classification of metagenomic sequences. Genome Res. **26**(12), 1721–1729 (2016). Epub 2016 Oct 17. PubMed PMID: 27852649; PubMed Central PMCID: PMC5131823
7. Breitwieser, F.P., Salzberg, S.L.: Pavian: interactive analysis of metagenomics data for microbiome studies and pathogen identification. Bioinformatics (2019). https://doi.org/10.1093/bioinformatics/btz715. pii: btz715. [Epub ahead of print] PubMed PMID: 31553437
8. Rajamohan, S., Romanella, A., Ramesh, A.: A weakly-supervised attention-based visualization tool for assessing political affiliation. arXiv:1908.02282 [cs.CL] (2019)
9. Pontiki, M., et al.: SemEval-2016 task 5: aspect based sentiment analysis. In: Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval 2016) (2016)
10. Python-rq.org: RQ: Simple job queues for Python (2019). http://python-rq.org/. Accessed 13 Sept 2019
11. Tweepy: tweepy/tweepy, GitHub, 04 September 2019. https://github.com/tweepy/tweepy. Accessed 13 Sept 2019
12. The most popular database for modern apps, MongoDB. https://www.mongodb.com/. Accessed 13 Sept 2019
13. Metabase is the easy, open source way for everyone in your company to ask questions and learn from data. Metabase. https://metabase.com/. Accessed 13 Sept 2019
14. Explosion: explosion/spaCy, GitHub. https://github.com/explosion/spaCy. Accessed 13 Sept 2019
15. Modern Analytic Apps for the Enterprise, Plotly. https://plot.ly/. Accessed 13 Sept 2019
16. Comet.ml: Comet.ml - Supercharging Machine Learning. https://www.comet.ml/. Accessed 13 Sept 2019
17. Peters, M.E., et al.: Deep contextualized word representations. arXiv preprint arXiv:1802.05365 (2018)