Semantic workflows for benchmark challenges: Enhancing comparability, reusability and reproducibility[*]

Arunima Srivastava
Computer Science and Engineering, The Ohio State University, 2015 Neil Ave Columbus, OH 43210
Email: srivatava.1@osu.edu

Ravali Adusumilli, Hunter Boyce
Canary Center for Cancer Early Detection, Stanford University, 3155 Porter Dr., Palo Alto, CA, 94305
Email: ravali@stanford.edu, hboyce@stanford.edu

Daniel Garijo, Varun Ratnakar, Rajiv Mayani
Information Sciences Institute, University of Southern California, Marina del Rey, Los Angeles ,CA 90292
Email: dgarijo@isi.edu, varunr@isi.edu, mayani@isi.edu

Thomas Yu
Sage Bionetworks, 2901 Third Ave., Suite 330, Seattle WA 98121
Email: thomas.yu@sagebionetworks.org

Raghu Machiraju
Computer Science and Engineering, The Ohio State University, 2015 Neil Ave Columbus, OH 43210
Email: machiraju.1@osu.edu

Yolanda Gil
Information Sciences Institute, University of Southern California, Marina del Rey, Los Angeles ,CA 90292
Email: gil@isi.edu

Parag Mallick[#]
Canary Center for Cancer Early Detection, Stanford University, 3155 Porter Dr., Palo Alto, CA, 94305
Email: paragm@stanford.edu

Benchmark challenges, such as the Critical Assessment of Structure Prediction (CASP) and Dialogue for Reverse Engineering Assessments and Methods (DREAM) have been instrumental in driving the development of bioinformatics methods. Typically, challenges are posted, and then competitors perform a prediction based upon blinded test data. Challengers then submit their answers to a central server where they are scored. Recent efforts to automate these challenges have been enabled by systems in which challengers submit Docker containers, a unit of software that packages up code and all of its dependencies, to be run on the cloud. Despite their incredible value for providing an unbiased test-bed for the bioinformatics community, there remain opportunities to

[*] The work is partially supported by DARPA Deep Purple Program through a DOI contract #D17AC00006, by DARPA SIMPLEX program award W911NF-15-1-0555, and by NIH award 1R01GM11709701.

[#] Corresponding Author

further enhance the potential impact of benchmark challenges. Specifically, current approaches only evaluate end-to-end performance; it is nearly impossible to directly compare methodologies or parameters. Furthermore, the scientific community cannot easily reuse challengers' approaches, due to lack of specifics, ambiguity in tools and parameters as well as problems in sharing and maintenance. Lastly, the intuition behind why particular steps are used is not captured, as the proposed workflows are not explicitly defined, making it cumbersome to understand the flow and utilization of data. Here we introduce an approach to overcome these limitations based upon the WINGS semantic workflow system. Specifically, WINGS enables researchers to submit complete semantic workflows as challenge submissions. By submitting entries as workflows, it then becomes possible to compare not just the results and performance of a challenger, but also the methodology employed. This is particularly important when dozens of challenge entries may use nearly identical tools, but with only subtle changes in parameters (and radical differences in results). WINGS uses a component driven workflow design and offers intelligent parameter and data selection by reasoning about data characteristics. This proves to be especially critical in bioinformatics workflows where using default or incorrect parameter values is prone to drastically altering results. Different challenge entries may be readily compared through the use of abstract workflows, which also facilitate reuse. WINGS is housed on a cloud based setup, which stores data, dependencies and workflows for easy sharing and utility. It also has the ability to scale workflow executions using distributed computing through the Pegasus workflow execution system. We demonstrate the application of this architecture to the DREAM proteogenomic challenge.

Keywords: Workflows; Semantic Workflows; DREAM Challenges; Proteogenomics; Benchmarking; Big Data

1. Introduction

The volume of experimental data being generated in the field of experimental biology is growing at a rapid pace in both size and variety[1,2]. With the advent of increasingly diverse data types, many of which are high throughput, the bioinformatics community is introducing sophisticated computational approaches for data analysis[3,4].

To compare different approaches, community-wide competitive benchmark challenges have gained popularity as an unbiased method to better understand the variety of pipelines proposed by different groups. Popular challenges include the Dialogue for Reverse Engineering Assessments and Methods (DREAM)[5], Critical Assessment of Structure Prediction (CASP) protein structure prediction[6] and The Association of Biomolecular Resource Facilities' (ABRF) Proteome Informatics Research Group's (iPRG) detection and prediction challenges[7]. These challenges give competitors the opportunity to test (in a blind and unbiased manner) their approach against others in the field, and have been instrumental in advancing diverse areas from protein structure prediction[8] to variant calling[9] to analysis of pathology data[10].

Unfortunately, evaluations in these competitions have traditionally been limited to metrics that evaluate solely based on scores. Comparisons of the methods that gave rise to those results are often left to manual interpretation. When the difference between a winner and an extremely poor performer may come down to a handful of parameters in otherwise identical workflows, the lack of transparency in methods is a huge missed opportunity for the bioinformatics community. In addition, winning methods are rarely shared with the broader community, as it is cumbersome to make winning methods accessible beyond the competition framework. Thus, while these challenges provide a forum for bioinformatics researchers to independently evaluate the performance of their approaches against others, the current execution environment for challenges does not facilitate deep comparison and sharing of approaches.

Consequently, there is a critical need to reconsider the infrastructure used for executing benchmark challenges. Here we examine the potential benefits of conducting benchmark challenges within a semantic workflow environment. Workflow environments, such as Galaxy[11] and GenePattern[12], would enable a challenge to examine not just the final results, but also all the steps of a method. This could include all dependencies, relevant data, and workflow components. By having challengers enter their submissions as workflows, which are executed on challenge data in the cloud, it becomes possible to more deeply perform a meta-analysis of the entries. In addition, submissions could be easily reused and shared by members of the broader scientific community.

This work describes our effort to date using the WINGS[13] semantic workflow system to submit entries to the DREAM proteogenomic challenge. While WINGS is an established (ready-to-download for server) workflow system[14], employing it as a submission and storing protocol for data analysis challenges is a novel use of this framework. In addition to the advantages typical of workflow systems, WINGS has additional features due to its use of semantic representations and reasoning about workflow steps and data. WINGS uses semantic annotations of data characteristics and step requirements in order to facilitate the selection of appropriate input parameter values based on metadata. WINGS additionally supports the creation of an abstract workflow component for a class of tools that perform a similar task, which greatly facilitates the comparison of different challenge entries. Finally, WINGS uses the W3C PROV standard[15] to record the complete provenance of the workflow execution details that led to a final result, including what tools and versions were used, how algorithm parameters were set, and the overall method. Key features of the execution environment of WINGS include: (a) a framework for recording all runtime dependencies of multi-step workflows, where each step is a self–contained component facilitated by employing Docker[16] images. Docker offers a virtual platform for building, sharing and running application within self-sufficient “containers” which allow encapsulation and storage of WINGS workflows. This includes the tools and data underlying each step (facilitating benchmarking), (b) a dynamic cloud based environment to house these workflows, complete with all runtime dependencies and data (facilitating reproducibility), and (c) a scalable execution environment (combination of WINGS and the

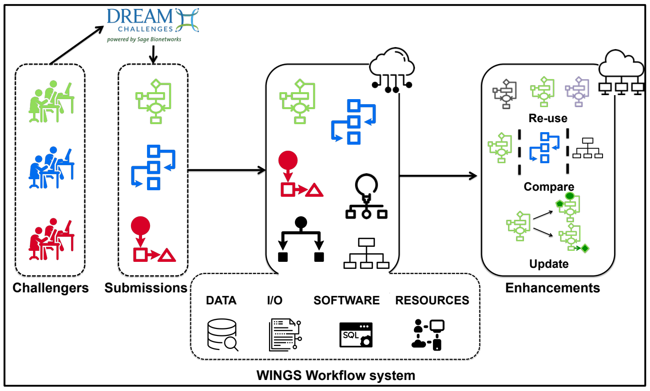
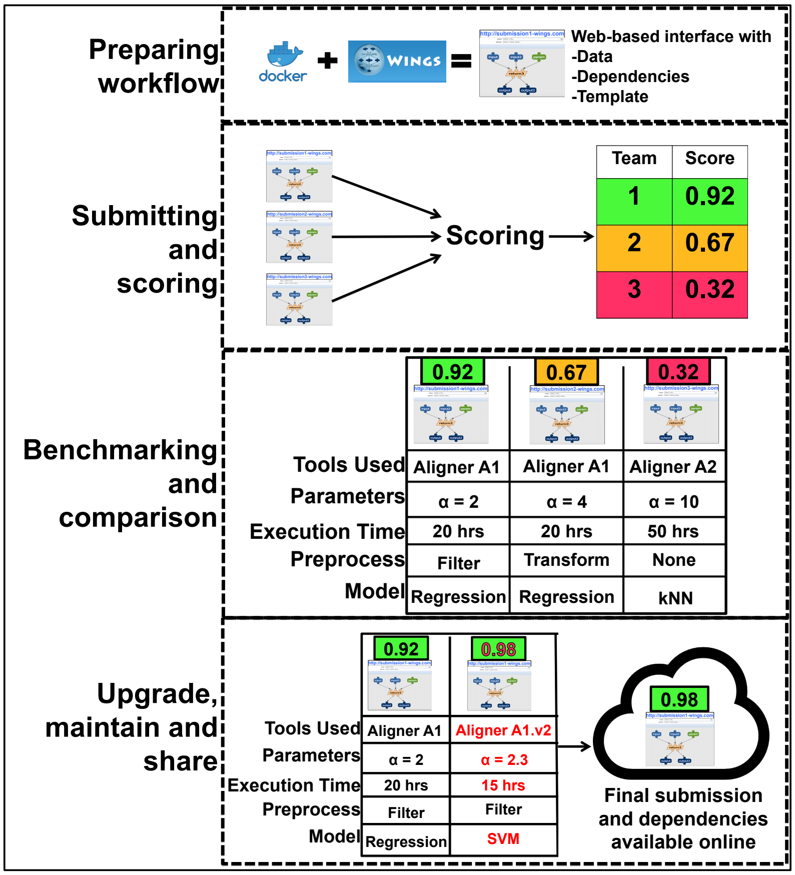


Fig. 1. Schematic for WINGS workflows in the context of data modeling and analysis competitions e.g. DREAM challenges. Building semantic workflows on the WINGS architecture enables widespread use of algorithms and methods, and enables storage and maintenance of data and workflows for use with high-throughput experiments.

Pegasus workflow management system[17] for distributed computing to reduce computational cost) to run workflows multiple times with new parameters or data (facilitating reusability).

Figure 1 shows a schematic of the use of WINGS for DREAM challenges. Integrating WINGS in current bioinformatics benchmarking challenges will support the reuse of the best performing solutions. Furthermore, it will expedite comparison between multiple different solutions, which potentially use similar constructs and tools, but differ in parameterizations that lead to significant result changes. This concludes to a better understanding of the underlying reasons that lead to a successful solution. Lastly, the extensive provenance records of all submitted solutions will greatly facilitate widespread use and adoption.

We discuss the WINGS design and the specifics of the workflow and environment construction in the sections below. Further, as proof of concept, we employ WINGS workflows to construct a full-scale pipeline for the NCI-CPTAC DREAM proteogenomic (protein prediction) challenge[18] that exhibits the main features of WINGS for reusability of workflows, reproducibility of results, and benchmarking of how results are impacted by subtle workflow variations. Lastly, we build multiple variations of the protein prediction workflow, altering different steps to illustrate how WINGS facilitates comparisons of different implementations of the workflow.



Team	Score
1	0.92
2	0.67
3	0.32

	0.92	0.67	0.32
Tools Used	Aligner A1	Aligner A1	Aligner A2
Parameters	α = 2	α = 4	α = 10
Execution Time	20 hrs	20 hrs	50 hrs
Preprocess	Filter	Transform	None
Model	Regression	Regression	kNN

	0.92	0.98
Tools Used	Aligner A1	Aligner A1.v2
Parameters	α = 2	α = 2.3
Execution Time	20 hrs	15 hrs
Preprocess	Filter	Filter
Model	Regression	SVM

Fig. 2. Using WINGS in each phase of benchmarking challenges to facilitate benchmarking, reproducibility, and reusability.

2. Methods and Materials

The WINGS workflow system can be readily integrated with the existing work cycle of a benchmark challenge such as the DREAM challenges. **Figure 2** describes the typical phases of a benchmark challenge and how a system like WINGS could fit the process. Each section below defines these phases and how the integration of WINGS can facilitate benchmarking, reproducibility, and reusability.

2.1. *Preparing and submitting workflows in WINGS for benchmark challenges*

The architecture and setup of WINGS (described in detail in the **supplementary materials**) facilitate easy usability and efficient sharing. A WINGS image, encapsulated by a Docker[16] container embedded with possible dependencies and software tools that may be needed by challengers to implement workflow steps, is built and made available at the onset of the challenge (**Figure 2**). New tools and software, as required by the codebase of each submission, can then be additionally included by the user within the WINGS framework where the submission pipeline is built.

WINGS facilitates the effective combination of utilities, scripts and tools based on different languages together under the umbrella of one single workflow, while allowing the user to see the high level view of the workflow steps in terms of the functions included within the workflow. **Figure 3** showcases the different components of a WINGS workflow. The main constructs involved are (1) *Components,* which encapsulate executable code described in terms of input data, parameters and outputs, each with unique datatypes and other semantic constraints (2) *Abstract components*, which can execute one of several codes with the same general functionality (e.g. an abstract component for normalization could be implemented by different normalization techniques, all employed on the same input, but resulting in different normalized data), (3) *Input parameters*, which may be string, integer, float, boolean or date values, (4) *Input files,* with metadata describing their type and contents, and (5) *Intermediate and final data*, which is output obtained from a component's execution that can be used as input to another component for further analysis.

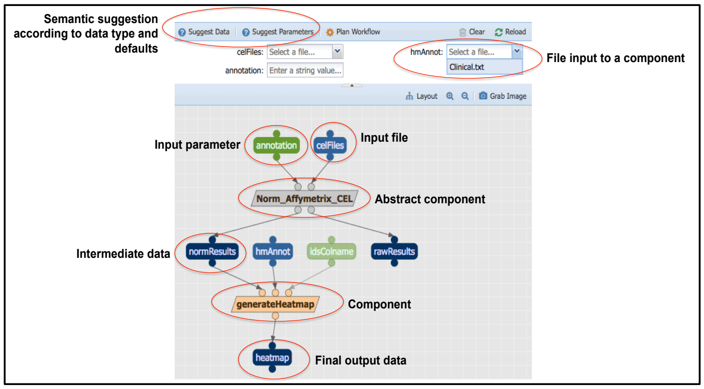


Fig. 3. Multiple components are connected in WINGS to design a workflow, as is typical of workflow systems. WINGS has unique features supported by semantic representations and reasoning: (a) automated suggestions of datasets and parameter values that are compatible with the current design of the workflow, (b) the possibility of defining abstract components that can be implemented by different tools.

Construction of a workflow in WINGS involves: (1) Creating data types and uploading raw input data, (2) Creating individual components for each distinct step in the workflow and supplying the code and scripts to generate outputs from inputs, (3) Connecting the components to reflect the flow of data from one to another. Additionally, the user can specify semantic metadata and validation rules to datasets, components, and workflows, which are used by WINGS to reason about the workflow and suggest data or parameters as well as to validate those provided by the user. The details of building a workflow in WINGS, using standard RNA-Seq processing as an example, are included in the **supplementary materials**.

We used WINGS for the NCI-CPTAC DREAM proteogenomic challenge. We created a workflow for predicting protein levels from transcriptomics data, which includes the processing of transcriptomics data from raw sequencing reads to a normalized gene-expression matrix used for protein level prediction.

2.2. *Benchmarking, comparison, upgrade and sharing of workflows*

Benchmarking challenges, such as the DREAM challenges, have historically evaluated the performance of each challenger's submission and reported on the top performing approaches. With the integration of WINGS, all submitted entries would be described as WINGS workflows. Each step of the workflows would be encapsulated in self-contained modules. Thus, each submitted workflow and their steps, can be benchmarked and compared amongst one another. WINGS abstract components would prove especially useful for comparisons as a challenger's workflow component will house the execution machinery for their specific approach while maintaining the same input and output as the components designed by their peers. Additionally, benchmarking and comparison facilitates iteratively fine-tuning a bioinformatics workflow, as it allows for easy comparisons of different input parameters, files and software modules. A record of executed workflows, with the associated meta-data as maintained in WINGS, helps identify and correct errors as well as optimize a workflow.

We use the protein prediction pipeline template provided to DREAM proteogenomic challenge participants and construct 6 variations on the same workflow (using abstract components), enabling benchmarking and comparative analysis.

Different variations of the workflow are initially compared on the basis of the same performance metric used to evaluate the results of the DREAM proteogenomics challenge. This is a correctness score, which is the aggregated Pearson's correlation of predicted protein levels to actual protein levels across samples. To further our understanding of the comparison between workflow variations, we compare three scales of data amongst each workflow execution: aligned reads, quantified transcriptomics expression, and final protein level prediction. This allows us to understand the factors culminating in the resulting correctness score. Aligned reads are compared by read coverage areas of the resulting BAM files (comparison employs deeptools module "multibamsummary"[19]), quantified expression and predicted protein levels are compared by assessing sample and gene-wise Spearman correlation of transcript/protein levels. WINGS facilitates this step-by-step comparison by allowing intermediate outputs to act as input to components performing individualized comparison. Executing non-WINGS challenge entries to store and compare intermediate output is potentially cumbersome and prone to errors as we would need: (a) access to the complete pipeline of each participant, (b) detailed annotations within the subsequent code explaining each step of the pipeline, and (c) computational power and storage to execute multiple workflows and store each intermediate and final output.

Upon completion of a challenge, the best performing solutions can easily be maintained and upgraded within the confines of the WINGS system. Any tools and data utilized can be swapped for latest versions. Additionally, utilizing the capabilities of containers ensures that the latest workflow and its ecosystem (dependencies and tools) can be encapsulated and shared with the community. The reusability of a workflow is not hampered by missing configurations, by lack of expertise to setup the computational environment, or by the absence of comprehensive descriptions of the pipeline itself.

3. Results

3.1. *WINGS workflow construction for the DREAM proteogenomic challenge*

As proof of concept for incorporating WINGS into a benchmark challenge, we built a workflow that performed protein level prediction from processed and normalized transcriptomics (RNA-Seq) data, mimicking the requirements of sub-challenge 2 of the NCI-CPTAC DREAM proteogenomic challenge 2018. Our workflow included the generation of a canonical transcriptomic expression matrix from raw reads allowing us to examine how sensitive the predictions were to changes at many phases of the workflow. Below we describe (**Figure 4**), (1) The entire workflow for protein level prediction from transcriptomics data and (2) The data and data types required to be uploaded and constructed in WINGS to facilitate workflow execution.

3.1.1. *The protein prediction workflow*

As our workflow aims to gauge protein levels for a set of samples from raw and unprocessed transcriptomics (RNA-Seq) data, it is divided to three distinct sections. (1) Alignment of raw read output from the sequencer, (2) Quantification and normalization per sample of aligned reads and lastly (3) Prediction of protein levels from processed and normalized transcriptomics data (**Figure 4**).

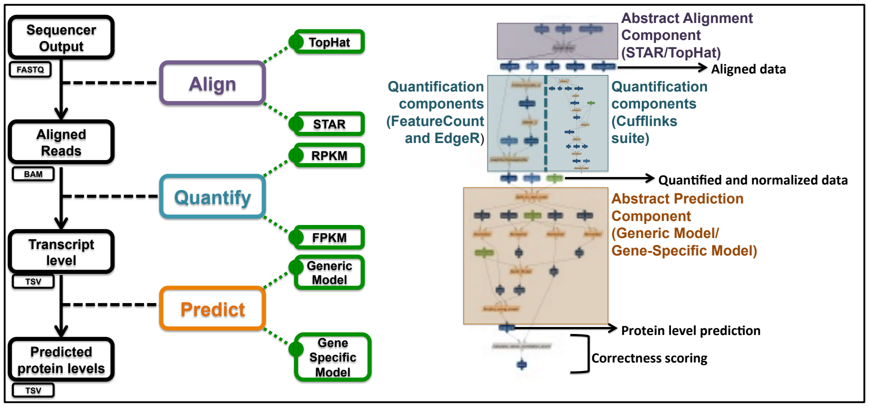

Sequencer Output
FASTQ
Aligned Reads
BAM
Transcript level
TSV
Predicted protein levels
TSV
Align
Quantify
Predict
TopHat
STAR
RPKM
FPKM
Generic Model
Gene Specific Model
Abstract Alignment Component (STAR/TopHat)
Aligned data
Quantification components (FeatureCount and EdgeR)
Quantification components (Cufflinks suite)
Quantified and normalized data
Abstract Prediction Component (Generic Model/ Gene-Specific Model)
Protein level prediction
Correctness scoring

Fig. 4. The protein prediction workflow as implemented in WINGS. The black boxes show the workflow schematic in terms of input, intermediate and output files. Alignment (purple), quantification (blue) and prediction (orange) are the three main sections of the workflow. The green boxes represent the changes to tools and parameters that result in variation of this predictive pipeline, and subsequently different outputs. On the left is the WINGS wire diagram of the complete workflow, with annotations marking the three main steps.

3.1.2. *The data and data type categorization for a workflow*

Input, output and intermediate files that are produced by the workflow dictate data types within WINGS (**Figure 4**). For the protein prediction workflow, the input files – RNA-sequencer output (FASTQ format), the output files – protein level matrix (TSV format) and the intermediate files - aligned reads (amongst others) (BAM format) guide the different data types to be constructed by the user apropos to the workflow.

The data utilized for protein prediction is The Cancer Genome Atlas/Clinical Proteomic Tumor Analysis Consortium (TCGA/CPTAC)-Colorectal Cancer datasets[20,21], which is one of the foundational proteogenomics datasets published by the National Cancer Institute (NCI). The data consists of transcriptomics and proteomics for 89 patient samples that are processed, analyzed and well characterized by multiple published experiments[22]. The raw data is available from both TCGA and CPTAC, and the processed data was extracted from supplementary material of associated publications. The data is housed within the WINGS image, hosted on an Amazon Web Server (**supplementary material**), contained within the workflow ecosystem, along with all the tools and scripts needed by the pipeline.

3.2. *Workflow variations for predicting protein levels*

We select 3 specific changes to the protein prediction workflow, spanning the three levels of input data processing and compared the final result. We aimed to make changes at each level of data dimensionality to assess the impact on the final protein prediction. The changes are made to (1) Alignment tools, (2) Transcript level quantification method and (3) Protein level prediction method as is summarized in **Figure 4**.

Alignment Tools (STAR[23] versus TopHat[24]) – We utilize the two widely adopted alignment tools for comparison. STAR is a fast, reliable reads aligner which requires a large amount of computing power but claims to address most shortcomings of other RNA-Seq aligners. TopHat is a traditional splice read mapper for RNA-Seq, which uses the ultra high-throughput short read aligner Bowtie to perform read alignment followed by identification of splice junctions.

Transcript level quantification method (FPKM versus RPKM) –The two most popular methods to quantify transcripts level expression are Fragments Per Kilobase of transcript per Million mapped reads (FPKM) and Reads Per Kilobase of transcript, per Million mapped reads (RPKM). Both normalize according to gene length, RPKM utilizes reads whereas FPKM estimates abundance based on fragments observed in a paired end experiment. We utilize the cufflinks suite[3] (cufflinks, cuffmerge, cuffquant and cuffnorm) to assess the FPKM quantification and featureCounts[25] with the EdgeR[26] R package to obtain the RPKM quantification.

Prediction method (Generic-Linear versus Gene-Specific) – The winners of the DREAM proteogenomic challenge employed multiple different models and one of the superior results was obtained by employing a Gene-Specific modeling technique for prediction[27]. Within our workflow, we aim to emulate their technique by building a unique linear model for each of the proteins to be predicted (Gene-Specific) and compare it against a one-fits-all linear model (Generic-Linear) that uses the entirety of the training data irrespective of gene and site specificity.

3.3. *Benchmarking and correctness of protein prediction across workflow variations*

As detailed above, a total of 6 different variations of the protein prediction workflow were executed using WINGS. Workflow variations included changes to the 3 distinct sections of the protein prediction workflow, namely alignment, quantification and prediction. **Table 1** summarizes the correctness (of prediction) score of the final result obtained from each variant of the workflow. We also note the approximate time (automatically recorded for each WINGS workflow execution) taken for each workflow completion. We observe the differences in quality of results based on the changes in different steps and dimensions of the prediction workflow. Namely, the largest change in resulting quality emanated from the different models used for prediction. The gene-specific model outperformed the generic linear model in all configurations. The alignment and quantification presented some minute changes in the final result quality but large differences in computational resource utilization, as the execution time was vastly different between STAR and TopHat usage, as well as evaluation of RPKM and FPKM.

Table 1. Pearson correlation based correctness score, and time taken for execution of each workflow configuration for protein level prediction of 89 samples and ~3000 proteins

Alignment	Quantification	Predictive Model	Correctness Score	Time Taken
STAR	FPKM	Linear	0.2161	~29 hrs
STAR	RPKM	Linear	0.2155	~20 hrs
STAR	FPKM	Gene-Specific	0.9064	~29 hrs
STAR	RPKM	Gene-Specific	**0.9124**	~20 hrs
TopHat	RPKM	Linear	0.2053	~103 hrs
TopHat	RPKM	Gene-Specific	0.9080	~103 hrs

3.4. *Comparison of workflow variations for predicting protein level*

Since intermediate output at each level is readily available in the WINGS provenance records, we explore each of the workflow variations at 3 different scales. Namely, we compare the aligned reads, the transcript quantitation and finally the predicted protein levels. **Figure 5** shows the WINGS workflow and the corresponding output for comparing aligned reads (BAM files). The component uses the utilities described in the section above to calculate the

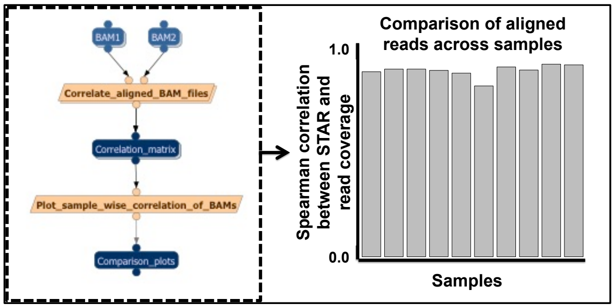


Fig. 5. Correlation between TopHat and STAR aligned reads across 10 samples (right) from the protein prediction workflow in WINGS (right).

correlation between read coverage for aligned reads obtained from both TopHat and STAR. **Figure 6** presents the component performing comparison of transcript quantification utilizing both FPKM and RPKM methodologies. The output visualizes a comprehensive comparison of both quantifications, by assessing the number of genes identified, gene and sample wise correlation and dynamic ranges of the gene-level expression.

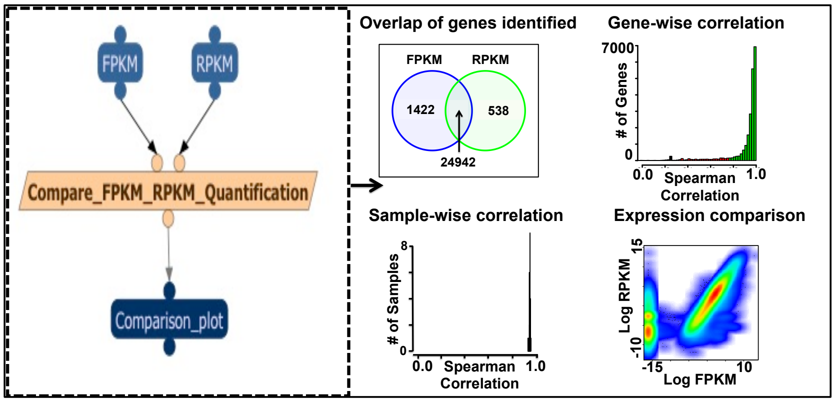


Fig. 6. Comparison between FPKM and RPKM transcript quantification obtained from the protein prediction workflow and the corresponding WINGS component utilized. Includes (Top Left) Overlap of genes identified using both the quantification methods, (Top Right) Gene-Wise expression correlation, (Bottom Left) Sample-wise expression correlation and (Bottom Right) Scatterplot of the entire quantification from both methods.

Lastly, **Figure 7** compares the final protein level prediction for two different models (Gene-Specific and Linear), as described in the section above. We show the component performing as well as visualizing the comparative analysis. Results include distribution comparison of predictions from both models and present correlation and dynamic ranges for both sets of predicted protein abundance. Changes to each step of a sequential workflow propagate downstream to alter the culminating output. The detailed analysis possible within the confines of WINGS allows us to fully understand the impact of each step's process on the final result of the protein prediction workflow. Further, since all intermediate data is accessible for each execution, data analysis and exploration can be performed in parallel at each step, including quality metrics, sanity checks and identifying critical data attributes characterizing inner workings of the pipeline. WINGS components performing analysis and exploration could be appended to the main workflow where they access intermediate data and provide immediate context to the workflow execution.

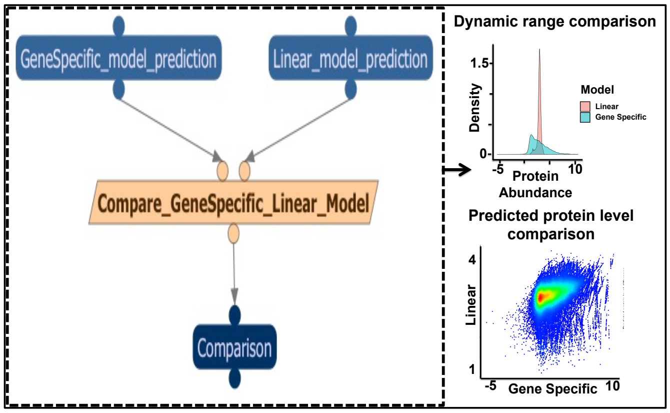


Fig. 7. Comparison between Gene-Specific and Linear modeling results obtained from the protein prediction workflow and the corresponding WINGS component utilized. Includes (Top) distribution comparison between the predicted protein levels from using each model for the 27 test samples (Bottom) Scatterplot comparison of each predicted protein level by both models for the 27 test samples.

4. Discussion and Conclusion

Our work presents the WINGS workflow infrastructure as an easy to use, effective and efficient platform for storing, maintaining and executing solutions submitted to analytical and modeling challenges. WINGS not only allows for standardization of submissions and effective reuse of workflows, it also allows for intuitive comparison between workflows as well as potential for changes and upgrades to ensure widespread adoption and rigorous reproducibility. As a proof of concept, we developed a protein prediction workflow using WINGS, akin to the DREAM proteogenomic challenge, which uses raw RNA-sequencing data as input, processing and modeling it to generate prediction for protein levels. WINGS houses the input data, performs benchmarking with different tools, techniques and models to identify the most effective configuration for protein prediction. In addition, for each variation of the workflow, we are able to identify and isolate critical changes in data across different steps as well as explore the nuances of the predictive model. Our experiments show the vast capability of WINGS and its usefulness to future bioinformatics analysis and modeling challenges. Additionally, incorporation of the WINGS paradigm in the context of data modeling and analytical challenges sheds light on a broader question of why a solution performs better than another. Constructing workflows with WINGS allows for researchers to use the most innovative methods by easily reusing the best performing approaches available for any given research question.

Supplementary material available at: https://github.com/arunima2/Supplementary_PSB_2019

References

1. Marx V. Biology: The big challenges of big data. *Nature*. 2013. doi:10.1038/498255a.

2. Stephens ZD, Lee SY, Faghri F, et al. Big data: Astronomical or genomical? *PLoS Biol*. 2015. doi:10.1371/journal.pbio.1002195.

3. Trapnell C, Roberts A, Goff L, et al. Differential gene and transcript expression analysis of RNA-seq experiments with TopHat and Cufflinks. *Nat Protoc*. 2012. doi:10.1038/nprot.2012.016.

4. Causey JL, Ashby C, Walker K, et al. DNAp: A Pipeline for DNA-seq Data Analysis. *Sci Rep*. 2018;8(1):6793. doi:10.1038/s41598-018-25022-6.

5. DREAM Challenges. http://dreamchallenges.org/.

6. Protein Structure Prediction Center. http://predictioncenter.org/.

7. Proteome Informatics Research Group (iPRG). https://abrf.org/research-group/proteome-informatics-research-group-iprg.

8. Moult J, Fidelis K, Kryshtafovych A, Schwede T, Tramontano A. Critical assessment of methods of protein structure prediction (CASP)-Round XII. *Proteins Struct Funct Bioinforma*. 2018;86:7-15. doi:10.1002/prot.25415.

9. Ewing AD, Houlahan KE, Hu Y, et al. Combining tumor genome simulation with crowdsourcing to benchmark somatic single-nucleotide-variant detection. *Nat Methods*. 2015. doi:10.1038/nmeth.3407.

10. Araújo T, Aresta G, Castro E, et al. Classification of breast cancer histology images using Convolutional Neural Networks. Sapino A, ed. *PLoS One*. 2017;12(6):e0177544. doi:10.1371/journal.pone.0177544.

11. Afgan E, Baker D, Batut B, et al. The Galaxy platform for accessible, reproducible and collaborative biomedical analyses: 2018 update. *Nucleic Acids Res*. 2018;46(W1):W537-W544. doi:10.1093/nar/gky379.

12. Reich M, Liefeld T, Gould J, Lerner J, Tamayo P, Mesirov JP. GenePattern 2.0. *Nat Genet*. 2006;38(5):500-501. doi:10.1038/ng0506-500.

13. Gil Y, Ratnakar V, Kim J, et al. Wings: Intelligent Workflow-Based Design of Computational Experiments. *IEEE Intell Syst*. 2011;26(1).

14. Zheng CL, Ratnakar V, Gil Y, McWeeney SK. Use of semantic workflows to enhance transparency and reproducibility in clinical omics. *Genome Med*. 2015. doi:10.1186/s13073-015-0202-y.

15. Missier P, Belhajjame K, Cheney J. The W3C PROV family of specifications for modelling provenance metadata. *Proc 16th Int Conf Extending Database Technol - EDBT '13*. 2013. doi:10.1145/2452376.2452478.

16. Merkel D. Docker: lightweight Linux containers for consistent development and deployment. *Linux J*. 2014. doi:10.1097/01.NND.0000320699.47006.a3.

17. Gil Y, Ratnakar V, Deelman E, Mehta G, Kim J. Wings for Pegasus: Creating Large-Scale Scientific Applications Using Semantic Representations of Computational Workflows. *Proc Twenty-Second Natl Conf Artif Intell*. 2007:1767-1774.

18. NCI-CPTAC DREAM Proteogenomics Challenge. https://www.synapse.org/#!Synapse:syn8228304/wiki/413428.

19. Ramírez F, Dündar F, Diehl S, Grüning BA, Manke T. DeepTools: A flexible platform for exploring deep-sequencing data. *Nucleic Acids Res*. 2014. doi:10.1093/nar/gku365.

20. Tomczak K, Czerwińska P, Wiznerowicz M. The Cancer Genome Atlas (TCGA): An immeasurable source of knowledge. *Wspolczesna Onkol*. 2015;1A:A68-A77. doi:10.5114/wo.2014.47136.

21. Whiteaker JR, Halusa GN, Hoofnagle AN, et al. CPTAC Assay Portal: a repository of targeted proteomic assays. *Nat Methods*. 2014;11(7):703-704. doi:10.1038/nmeth.3002.

22. Zhang B, Wang J, Wang X, et al. Proteogenomic characterization of human colon and rectal cancer. *Nature*. 2014. doi:10.1038/nature13438.

23. Dobin A, Davis CA, Schlesinger F, et al. STAR: Ultrafast universal RNA-seq aligner. *Bioinformatics*. 2013. doi:10.1093/bioinformatics/bts635.

24. Kim D, Pertea G, Trapnell C, Pimentel H, Kelley R, Salzberg SL. TopHat2: Accurate alignment of transcriptomes in the presence of insertions, deletions and gene fusions. *Genome Biol*. 2013. doi:10.1186/gb-2013-14-4-r36.

25. Liao Y, Smyth GK, Shi W. FeatureCounts: An efficient general purpose program for assigning sequence reads to genomic features. *Bioinformatics*. 2014. doi:10.1093/bioinformatics/btt656.

26. Robinson MD, McCarthy DJ, Smyth GK. edgeR: a Bioconductor package for differential expression analysis of digital gene expression data. *Bioinformatics*. 2010. doi:10.1093/bioinformatics/btp616.

27. Li H, Guan Y. Guanlab's solution to the 2018 NCI-CPTAC DREAM Proteogenomics Challenge. https://www.synapse.org/#!Synapse:syn11522015/wiki/496744.