*Genome Medicine, 2015, 7:73 doi:10.1186/s13073-015-0202-y, Published: 25 July 2015.*

# Use of Semantic Workflows to Enhance Transparency and Reproducibility in Clinical Omics

**Christina L Zheng[1,2§], Varun Ratnakar [3], Yolanda Gil[3], and Shannon K McWeeney[1,2,4]**

[1]Division of Bioinformatics and Computational Biology, Department of Medical Informatics and Clinical Epidemiology, Oregon Health & Science University, Portland, Oregon, USA

[2]Knight Cancer Institute, Oregon Health & Science University, Portland, Oregon, USA

[3]Information Sciences Institute, University of Southern California, Los Angeles, California, USA

[4]Division of Biostatistics, Department of Public Health and Preventative Medicine, Oregon Health & Science University, Portland, Oregon, USA

[§]Corresponding author

Email addresses:
*zheng@ohsu.edu, varunr@isi.edu, gil@isi.edu, mcweeney@ohsu.edu*

## Abstract

### Background

Recent highly publicized cases of premature patient assignment into clinical trials, resulting from non-reproducible omics analyses, have many calling for a more thorough examination into the field of translational omics and emphasizing the critical need for transparency and reproducibility to ensure patient safety. The use of workflow platforms such as Galaxy and Taverna, have greatly enhanced the use, transparency, and reproducibility of omics analysis pipelines in the research domain and would be an invaluable tool in a clinical setting. However, the use of these workflow platforms requires deep domain expertise, which particularly within the multi-disciplinary fields of translational and clinical omics, may not always be present in a clinical setting. This lack of domain expertise may put patient safety at risk and make these workflow platforms difficult to operationalize in a clinical setting. In contrast, semantic workflows are a different class of workflow platforms where resultant workflow runs are transparent, reproducible, and semantically validated. Through the semantic enforcement of all datasets, analyses, and user-defined rules/constraints, users are guided through each workflow run, enhancing analytic validity and patient safety.

**Methods**

To evaluate the effectiveness of semantic workflows within translational and clinical omics, we have implemented a clinical omics pipeline for annotating DNA sequence variants identified through next generation sequencing, using the Workflow Instance Generation and Specialization (WINGS) semantic workflow platform.

**Results**

We found that the implementation and execution of our clinical omics pipeline in a semantic workflow helped us to meet the requirements for enhanced transparency, reproducibility, and analytical validity recommended for clinical omics. We further found that many features of the WINGS platform were particularly primed to help support the critical needs of clinical omics analyses.

**Conclusions**

This is the first implementation and execution of a clinical omics pipeline using semantic workflows. Evaluation of this implementation provides guidance for their use in both translational and clinical settings.

# Background

High throughput 'omics' technologies such as genomics, proteomics, metabolomics etc. hold great promise for precision medicine wherein a patient's personal omics data is used to inform individualized care. Recently published preclinical omics studies highlight the tremendous potential omics can have on improving patient care through assessing disease risk [1-4], averting potential adverse drug reactions [5-7], and ultimately tailoring treatment to the individual, not the disease [8-10]. The potential of having disease traits be monitored through omics data of healthy individuals [11] has also garnered much excitement.

Despite the large number of published preclinical omics studies, only a few have been successfully translated into a clinical setting [12, 13]. The primary scientific causes for this have been attributed to 1) preclinical omics studies not being adequately designed to answer the intended clinical question and 2) inadequate statistical or bioinformatics rigor [14]. The latter issue has garnered much attention with respect to both the benchmarking and quality control of omics analysis pipelines and the transparency and reproducibility of those pipelines once they are established. Efforts to benchmark the accuracy, biases, and sources of errors within omics analysis methods are critical to translational and clinical omics [15]. On the heels of the US Food and Drug Administration's (FDA) approval of the first next-generation sequencing instrument [16], their recent public workshop on Next Generation Sequencing Standards highlighted the critical need for the quality assurance of computational biology pipelines [17]. Towards these efforts, the National Institute of Standards and Technology (NIST), in conjunction with the Genome in a Bottle Consortium, recently published a set of high-confidence, genome-wide single-nucleotide polymorphism (SNP), indel and genotype calls, based on a genome

sequence that they have established as a DNA reference material and made freely available to be used as a truth table in the benchmarking of bioinformatics methods for identifying DNA variants from sequenced genomes [15]. Unfortunately efforts towards making clinical omics analysis pipelines more transparent and reproducible are still in their infancy. Even in the clinical and translational research domain, there has been a critical need for computational transparency and reproducibility [14, 18]. This is exemplified by a recent study in which over 1500 person hours were dedicated to the 'forensic omics' task of deciphering the exact data sets used and determining how the data were processed for assignment of patients to clinical trial [19].

Thus, a key challenge now is how we can increase transparency and reproducibility. This question is pertinent to clinical omics and the scientific community as a whole [20-22]. This is highlighted by the recent work of Garijo et al., whose efforts to reproduce a published computational method led them to publish a set of reproducibility guidelines for authors [23]. They recommend that authors include all pertinent data: the exact input data used, key intermediate data, output data, and any third party data (i.e., from external databases) for the analysis. They also recommend the inclusion of all software code, parameters, and configuration files necessary for the analysis. Finally they recommended including a high level flow diagram to guide users through the entire approach. Two recent reports echoed similar requirements for translational and clinical omics with the addition of key transparency requirements including the need for data provenance to help ensure data integrity and the need to enhance analytical validity to help ensure "we are doing the test correctly" [14, 18]. We have summarized the requirements across these studies into a checklist to facilitate the evaluation of transparency and reproducibility in translational and clinical omics (Table 1).

Workflow systems such as Galaxy [24] and Taverna [25] help to meet many of the requirements listed above and have greatly enhanced the use, transparency, and reproducibility of omics pipelines in the research domain [25, 26]. With these systems, exact input, key intermediate, final output, and relevant external data are all preserved. All code, computational configurations, parameters, and their provenance can be captured within these systems. These systems also provide a high level flow diagram to guide users through execution. However a key requirement is inherently missing from these systems: there is no way to include veracity checks during workflow runs to enhance analytical validity. The execution of workflows within these systems therefore requires deep domain knowledge and expertise to ensure data integrity and analytical validity. For example, it is the user's responsibility to ensure that the correct input is provided, the systems do not inherently validate the input provided, nor do they provide guidance to the user of the appropriate input needed. Particularly within multi-disciplinary fields such as translational and clinical omics where expertise from clinicians, laboratory personnel, bioinformaticists, and statisticians must be effectively integrated and navigated, expertise across all fields may not always be present in 'real time' in the clinical setting thus putting patient safety at risk and making these workflow platforms inadequate for a clinical setting.

**Table 1. Criteria Checklist for Enhanced Transparency and Reproducibility in Clinical Omics**

| |
|---|
| ▪ exact input data used for the analysis |
| ▪ key intermediate data generated from the analysis |
| ▪ third party data (i.e., data from external sources) |
| ▪ output data |
| ▪ provenance of all data used |
| ▪ all code/software used in the analysis |
| ▪ provenance of all code used |
| ▪ documentation of computing environment used |
| ▪ veracity checks to ensure analytical validity |
| ▪ high-level flow diagram describing the analysis |

We recently investigated the use of semantic workflows with the analysis of multi-omics data and found that the encapsulation of multi-step omics analysis methods within a semantic framework resulted in a transparent, reproducible, and semantically validated analysis framework [27], making semantic workflows a potential viable candidate for clinical omics. Semantic workflows are a unique and different class of workflow platforms. Similar to other workflow systems, semantic workflows manage and record the execution of complex computations, record provenance, and allow end-users to reproduce workflows. However; unique to semantic workflow systems is its ability to generate semantically validated workflow runs wherein domain expertise can be encoded within user-defined rules and constraints, and these rules and constraints are semantically enforced to help guide users through a workflow run. This guidance enhances data integrity and analytical validity throughout a workflow run, thus making semantic workflows a potential candidate for meeting the critical needs of transparency, reproducibility, and analytical validity in a clinical setting.

To evaluate the use of semantic workflows within clinical omics, we have implemented and executed the first clinical omics analysis pipeline using the Workflow Instance Generation and Specialization (WINGS) semantic workflow platform [28]. We found the WINGS platform capable of effectively meeting the checklist of requirements for enhanced transparency, reproducibility, and analytical validity recommended for translational and clinical omics defined at the beginning of this study. We further found that many features of the WINGS platform were particularly effective in supporting the critical needs of clinical omics analyses, such as the need to keep pace with frequent updates of biological life science databases, to enforce consistency/data integrity across heterogeneous biological/clinical data, to keep pace with rapid updates/development of omics software tools, and to process large omics data sets.

# Methods and Results

### Use-Case: Clinical Omics Analysis Pipeline

The clinical omics pipeline use-case, in this study, is a DNA variant annotation pipeline, provided by the Knight Diagnostic Laboratories (KDL) at Oregon Health and Science University (OHSU) for this implementation, aimed at coalescing molecular, pathogenic, and population annotation information on DNA variants identified through DNA sequencing from a patient's tumor sample. DNA sequencing was performed on the Ion Torrent Personal Genome Machine (PGM$^{TM}$) System for Next-Generation Sequencing, using the GeneTrails Solid Tumor Panel®, which delivers information on 37 genes commonly involved in solid tumors.

The omics annotation pipeline begins with a file of sequenced DNA variants from a patient's tumor sample. All identified DNA sequence variants are annotated with the following information: 1) potential effect on the resultant protein(s), 2) annotation within the Catalogue of Somatic Mutations in Cancer (COSMIC) database [29], and 3) annotation within the Single Nucleotide Polymorphism database (dbSNP) [30]. The potential molecular effect of the DNA variant on the amino acid (aa) sequence of the resultant protein(s) (e.g., non-synonymous) is analysed using the Bioconductor VariantAnnotation package [31]. Information regarding the DNA variants potential pathogenic associations with cancer and its frequency within the population is obtained through COSMIC and dbSNP, respectively. Additional manually curated information regarding the DNA variants (for example, if it is within a homo-polymer region), if available, is also incorporated. The final output of the annotation pipeline is a file coalescing all of the obtained annotation information for all identified DNA variants from the patient's tumor sample. This output is then used by clinicians to aid in determining individualized patient care.

This DNA variant annotation pipeline use-case involves a small number of annotation resources; however, even at this level, the importance of and difficulty in adhering to the requirements of transparency, reproducibility and accuracy is evident. For example, the computational code for this analysis pipeline was stored on multiple desktop machines and executed by multiple laboratory personnel. The lack of a central location for the storage and execution of the code exposed opportunities for potential errors and inconsistencies, making reproducibility very difficult. The use of multiple workstations introduced potential inconsistencies arising from the use of different versions of software or code. Potential errors or inconsistencies might have also arisen from unmet constraints such as ensuring that all genomic coordinates among the different annotation resources are of the same genomic assembly. Additionally a lack of version control and automated provenance tracking of the annotation sources further complicates the task of accuracy and reproducibility.

### The WINGS Semantic Workflow System

The WINGS workflow system [28] is a unique class of workflow platforms wherein analysis pipelines are transformed into transparent, reproducible, semantically validated workflow runs. Similarly to other workflow systems, through

the encapsulation of analysis steps into individual workflow components with predefined inputs, outputs, and parameters, WINGS tracks and records the provenance of complex computations and enables end-users to reproduce workflows. However, unique to WINGS is its ability to generate semantically validated workflow runs wherein all components and datasets are automatically checked for coherence and consistency and all user-defined rules and constraints are semantically enforced. WINGS accomplishes this through two features not found in other workflow platforms: 1) the integration of individual workflow components and their datasets, and 2) the semantic enforcement of user-defined rules and constraints. Formal descriptions and detailed algorithms for WINGS can be found in Gil et. al., 2011 [32].

The integration of individual workflow components and their datasets within WINGS is achieved through the use of individual ontologies used to define and organize all datasets and workflow components, respectively. Within the dataset ontology, categories are defined for each dataset, and within the workflow component ontology, categories are defined for each workflow component. Categories can be developed using study custom or standardized biological ontologies (e.g., EDAM [33], SeqOntology [34, 35], etc.). In this way, all datasets and workflow components are clearly defined (e.g., metadata, parameters) and organized within their individual categories. These categories can then be used to define relationships within an individual ontology such as defining one dataset as a sub-class of an existing dataset or defining one workflow component as a sub-class of an existing workflow component. These categories can also be used to define relationships across the two ontologies, such that the use of specific dataset categories can be restricted or pre-set within individual workflow components. The ability for cross-talk between the two ontologies creates an unprecedented integration between workflow components and their datasets wherein only predefined datasets are used and set throughout the workflow thus maintaining data integrity. Within other workflow platforms, such as Galaxy and Taverna, which do not have this level of integration, data integrity is at risk, as the correct usage of datasets throughout a workflow run is not automatically verified. Although Galaxy and Taverna workflow components can explicitly be defined to specify the format type (e.g., FASTA file, sam/bam format) of required datasets, no explicit inherent format type checking is performed to ensure that a dataset of the specified format type was provided by the user.

Further enhancing the ability of WINGS to create semantically validated workflow runs is that it can semantically enforce user-defined rules and constraints. In doing so, workflow developers are able to further refine relationships across and between datasets and workflow components. For example, developers can constrain all datasets within a workflow run to have a specific metadata value (for instance, specific genome assembly). Rules can also be defined to require that specific datasets be processed by specific workflow components (described further below). In essence, through the use of predefined rules and constraints, domain knowledge and expertise is embodied and disseminated with each workflow. This not only enhances the analytical accuracy and validity of each workflow run, but it also guides users through

a workflow run as error messages are displayed if any rule or constraint is violated. Optional semantically validated datasets can also be suggested upon user request.

WINGS has other functionality that is not directly related to its semantic capabilities [36]. One is the large-scale execution of workflows, which was one of the first capabilities incorporated in WINGS to support large-scale earthquake simulations [37]. Once a workflow is set up, WINGS can execute it in several alternative modes [38]. In one mode, its execution environment can be a local host, with WINGS generating scripted codes, or a distributed execution on a network of local machines. Alternatively, WINGS can generate execution-ready workflows that can be submitted to either Apache OODT [39] or the Pegasus/Condor execution engine [40] that are designed for large-scale distributed data processing in a variety of environments, such as local clusters, shared infrastructure, or cloud resources. Furthermore, based on user-defined execution requirements, WINGS can automatically generate the most appropriate and/or efficient workflows [41]. WINGS has not, however, been used to compose web services into workflows while other workflow systems such as Taverna can support it.

WINGS publishes and shares workflows using the W3C PROV-O ontology for workflow executions and its extension OPMW to represent workflow templates (http://www.w3.org/TR/prov-o/, http://www.opmw.org/model/OPMW/). OPMW is based on the W3C PROV model as well as the earlier Open Provenance Model adopted by many workflow systems [42]. OPMW supports the representations of workflows at a fine granularity with a lot of details pertaining to workflows that are not covered in more generic provenance models [43]. OPMW also allows the representation of links between a workflow template, a workflow instance created from it, and a workflow execution that resulted from an instance. Finally, OPMW also supports the representation of attribution metadata about a workflow, which some applications consume.

The WINGS workflow repository is publicly available and is part of the WEST ecosystem [44] that integrates different workflow tools with diverse functions (workflow design, validation, execution, visualization, browsing and mining) created by a variety of research groups. These tools include LONI Pipeline [45], Apache OODT and Pegasus/Condor. The workflow repository has been used to mine workflow patterns [44, 46]. WEST uses workflow representation standards and semantic technologies to enable each tool to import workflow templates and executions in the format they need. WEST is the first integrated environment where a variety of workflow systems and functions interoperate, and where workflows produced by a given tool can be used by more than one other tool. Other benefits of this approach include the interoperability among the applications in the ecosystem, the flexibility to interchange data, and facilitating the integration of content modelled in other vocabularies. Our representations are mapped to an extension of PROV for reusable plans called P-PLAN [47] as a basis to further map to processes other than workflows such as scientific experiments that use ISA [48]. Workflow repositories such as myExperiment [49] and CrowdLabs [50] can be used for sharing scientific workflows created with other systems. These workflows are reused by scientists that

seek, retrieve, and reapply them. However, these workflows are not described with any structured annotations or constraints that capture their applicability as WINGS does.

Other workflow systems used in biomedical research such as LONI Pipeline, Taverna, GenePattern [51], and Galaxy offer very useful capabilities, and include libraries of components that are widely used in the community, such as genomic analysis tools or Bioconductor services [52]. However, their workflow representations specify the software to run at each step, but do not represent constraints such as whether an algorithm is appropriate given a dataset's characteristics or how to set a software tool's parameters to get best results. The SADI framework proposes best practices for documenting services with semantic constraints, and provides a plug-in for Taverna where services can be incorporated into the workflow based on semantic constraints, but does not support constraint propagation and reasoning at the workflow level [53]. WINGS is unique in capturing such semantic constraints. Please refer to Supplemental Information for additional information on the WINGS system.

## Implementation of A Clinical Omics Workflow Using the WINGS Semantic Workflow System

The first step in implementing a WINGS semantic workflow is for a workflow developer to create all datasets, components, rules, and constraints needed for an analysis pipeline. These are then used to build the workflow template needed for workflow users to execute reproducible and semantically-validated workflow runs. Each is described in more detail below.

### Datasets and Their Metadata

Datasets consist of any input, output, or intermediate data files within an analysis pipeline. For example, within our DNA variant annotation pipeline, key datasets include 1) Patient_Called_DNA_Variant_File: the file of sequenced DNA variants from a patient's tumor, 2) COSMICSubset: the GeneTrails-specific subset of COSMIC, 3) SNPSubset: the GeneTrails-specific subset of dbSNP, and 4) Final_Annotation_of_DNA_Variants: the final annotation file of the identified DNA variants. Please refer to Table 2 for a complete list of datasets found within our pipeline. Because all datasets are defined within an ontology, WINGS is able to effectively organize and constrain the use of each dataset (Figure 1A). We note that custom or standardized ontologies (e.g., The Sequence Ontology which not only represents the DNA variants but also contains the Protein Feature Ontology to handle protein consequence [54]) can easily be used. Some datasets are defined as their own entity (e.g., GeneTrails_Genes or Patient_Called_DNA_Variant_File) while others are defined as sub-classes to other datasets (e.g., Queried_SNP_Result and SNPSubset are sub-classes of SNPData). By defining datasets as sub-classes to other datasets, common metadata can be shared among the parent and child datasets. For example, dbSNPVersionId is common metadata for SNPData, SNPSubset, and Queried_SNP_Result datasets. Metadata for each dataset can be defined, populated, updated, and viewed using the WINGS framework (Figure 1B). Metadata can also be

**Table 2. WINGS Datasets for our Clinical Omics Use-Case**

| Dataset | Description |
|---|---|
| GeneTrails_Genes | list of genes on the GeneTrails Solid Tumor Panel® |
| COSMICSubset | GeneTrails specific subset of COSMIC |
| SNPSubset | GeneTrails specific subset of dbSNP |
| Patient_Called_DNA_Variant_File | identified DNA variants from a patient's tumor sample |
| Queried_COSMIC_Result | queried COSMIC annotation specific to a Patient_Called_DNA_Variant_File |
| Queried_SNP_Result | queried dbSNP annotation specific to a Patient_Called_DNA_Variant_File |
| Transcript_File | transcripts of interest from GeneTrails_Genes |
| Predicted_Protein_Consequence | predicted consequence(s) specific to a Patient_Called_DNA_Variant_File |
| In_House_Curation_of_DNA_Variants | manually curated information on sequence characteristics of previously identified DNA variants |
| Final_Annotation_of_DNA_Variants | coalesced annotation information from the workflow specific to a Patient_Called_DNA_Variant_File |

automatically populated and propagated through-out a workflow run. For a complete list of metadata used in our workflow, please refer to the Supplemental Information.

**Workflow Components**

Workflow components define and encapsulate each step of an analysis pipeline. Similarly to datasets, all WINGS components are classified using an ontology where an individual component can either be classified as its own entity or grouped under a super-component class termed "component-type". Component-types are used to group components sharing a common base set of input/output datasets such as those encapsulating code for different versions of the same tool or different tools performing similar functions. Component-types can also be used to effectively organize and enhance the flexibility of individual components within a workflow template wherein components can be easily incorporated into existing component-types with their use semantically enforced (discussed further below).

To capitalize on the many features of component-types, each step of our clinical omics pipeline was segregated into the following component-types: 1) *CreateLocalCOSMIC*, 2) *CreateLocalSNP*, 3) *QueryLocalCOSMIC*, 4) *QueryLocalSNP*, 5) *PredictProteinConsequence*, and 6) *MergeAnnotation* (Figure 2A). *CreateLocalCOSMIC* created a dataset containing a subset of COSMIC annotation specific for genes found on the GeneTrails Solid Tumor Panel®. *CreateLocalSNP* creates a dataset containing a subset of dbSNP annotation specific

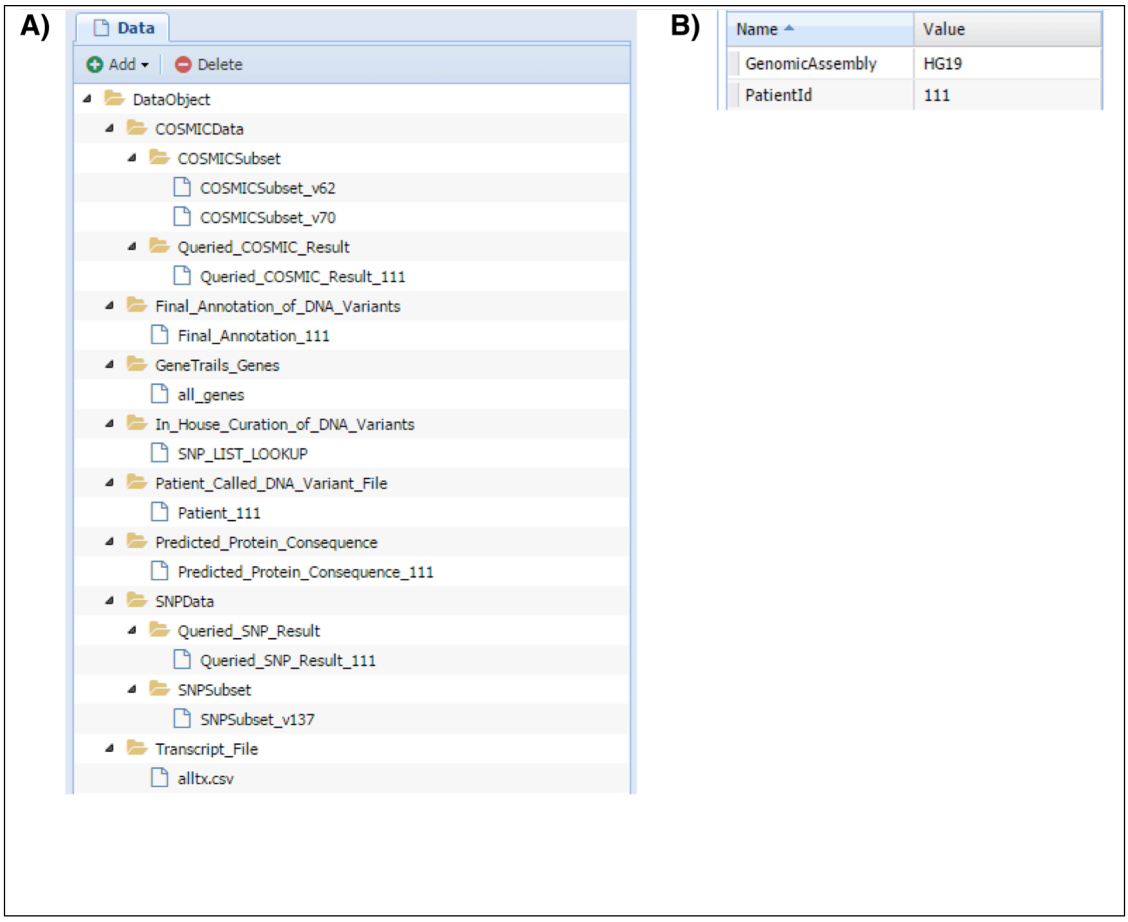


**Figure 1. WINGS Datasets Ontology for Our Clinical Omics Use-Case.** WINGS datasets - any input, output, or intermediate data files - within a workflow template are classified within an ontology. A) The ontology classifying the datasets within our WINGS omics workflow is shown. Each dataset can be defined as an individual class or defined as a sub-class of an existing dataset. Patient_Called_DNA_Variant_File is an example of an individually defined dataset class while COSMICSubset and Queried_COSMIC_Result are examples of sub-classes under the COSMICData dataset. Each dataset can be further defined with metadata. The defined metadata and its value for a Patient_Called_DNA_Variant_File are shown (B).

for genes found on the GeneTrails Solid Tumor Panel®. *QueryLocalCOSMIC* queried the COSMIC subset dataset for annotation information pertaining to a file of identified DNA variants from a patient's tumor sample. *QueryLocalSNP* queried the dbSNP subset dataset for annotation information pertaining to a file of identified DNA variants from a patient's tumor sample. *PredictProteinConsequence* predicted the potential molecular effect of the resultant amino acid changes caused by the DNA variant identified from a patient's tumor sample. *MergeAnnotation* merged all annotation information obtained from the other components, in addition to information obtained from a file of manually curated annotations that detail sequence characteristics of the identified DNA variant (for example, within a homopolymer

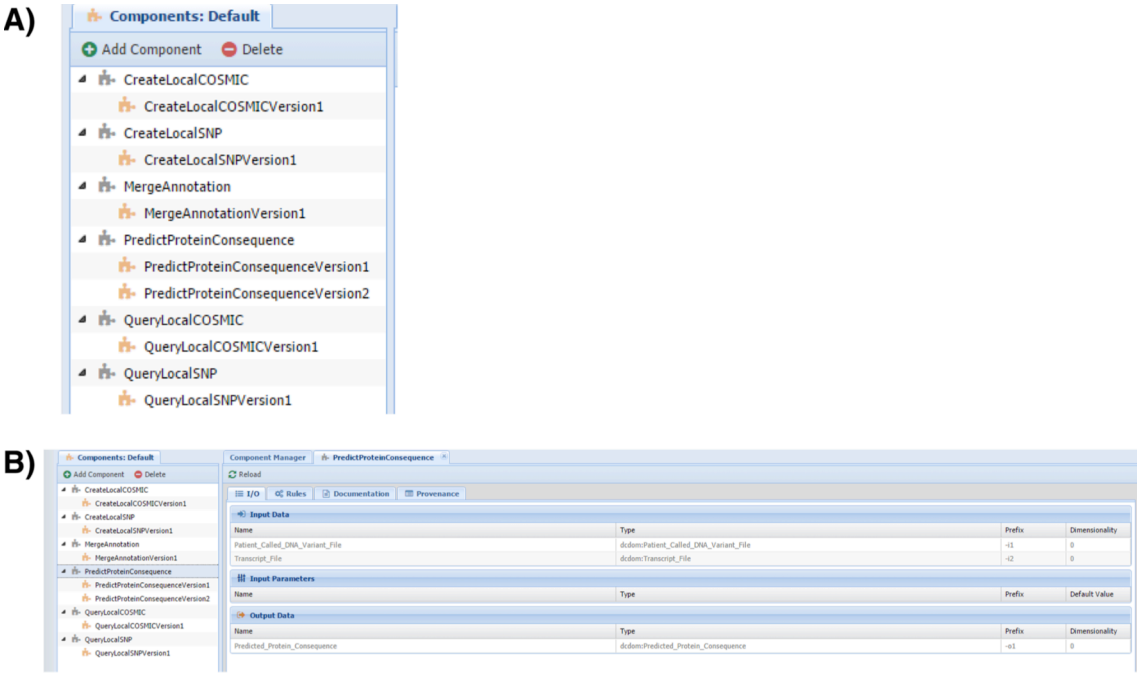


**Figure 2. WINGS Workflow Components Ontology for Our Clinical Omics Use-Case.** WINGS components are used to encapsulate individual steps of an analysis pipeline and are classified within an ontology in a workflow template. Individual components can be classified as their own component-class or as a sub-class of a component-type. Component-types are used to group components sharing a common base set of input and output datasets such as those encapsulating code for different versions of the same tool or different tools performing similar functions. Component-types can also be used to effectively organize and enhance the flexibility of individual components within a workflow template. Each step of our clinical omics analysis pipeline was encapsulated within a component-type, even if only one component is currently defined (A). Individual component-types are shown in grey while individual components are depicted in yellow. Each component is defined with the following: 1) input datasets, 2) computational code, and 3) output datasets. For example, each *PredictProteinConsequence* component was defined with the following two input datasets: 1) Patient_Called_DNA_Variant_File and 2) Transcript_File and the following output dataset: 1) Predicted_Protein_Consequence (B). The R code needed for the analysis of this step was included to complete the creation of the component.

region); it then output a final file detailing the annotation information for the identified DNA variants.

Individual components were then created for each component-type. For example, the components *PredictProteinConsequenceVersion1* and *PredictProteinConsequenceVersion2* were created under the *PredictProteinConsequence* component-type and the component *QueryLocalCOSMICVersion1* was created under the *QueryLocalCOSMIC*

**Table 3. WINGS Input/output Datasets for each Component-Type within our Clinical Omics Use-Case**

| Component-Type | Input Dataset(s) | Output Dataset(s) |
|---|---|---|
| CreateLocalCOSMIC | GeneTrails_Genes | COSMICSubset |
| CreateLocalSNP | GeneTrails_Genes | SNPSubset |
| QueryLocalCOSMIC | Patient_Called_DNA_Variant_File, COSMICSubset | Queried_COSMIC_Result |
| QueryLocalSNP | Patient_Called_DNA_Variant_File, SNPSubset | Queried_SNP_Result |
| PredictProteinConsequence | Patient_Called_DNA_Variant_File, Transcript_File | Predicted_Protein_Consequence |
| MergeAnnotation | Pateint_Called_Variant_File, Queried_COSMIC_Result, Queried_SNP_Result, Predicted_Protein_Consequence, In_House_Curation_of_DNA_Variants | Final_Annotation_of_DNA_Variants |

component-type. Each component was defined with the following: 1) input datasets, 2) computational code, and 3) output datasets. For example, each *PredictProteinConsequence* component was defined with the following two input datasets: 1) Patient_Called_DNA_Variant_File and 2) Transcript_File and the following output dataset: 1) Predicted_Protein_Consequence (Figure 2B). Thus datasets not classified as a Patient_Called_DNA_Variant_File or Transcript_File dataset would not be a valid input into the *PredictProteinConsequence* component. Similarly, any output from the *PredictProteinConsequence* component would be classified as a Predicted_Protein_Consequence dataset. The code needed for the analysis of this step was included to complete the creation of the component. This component utilizes the Bioconductor VariantAnnotation package [31] for its analysis (please refer to Clinical Omics Analysis Pipeline section for more detail) however code implementing other popular annotation methods may easily be incorporated or used in its place. Please refer to Table 3 for a complete description of all input/output datasets for each component-type.

**Semantic Rules and Constraints**

Workflow rules and constraints can be used to enforce user-defined rules/constraints needed within a workflow template to create a semantically validated workflow run such as any pre-specified requirements for input datasets, inter-dependencies between components and/or datasets, or recommended/proposed regulations. Rules and constraints currently defined within our clinical workflow include requiring that genomic coordinates across all datasets be of the same genomic assembly and ensuring the propagation of pre-defined sets of metadata (e.g., patient ID number, software versions, data set versions) throughout a workflow run. Effective metadata propagations aid in effective provenance tracking. User-defined rules and constraints have also been put in place to pre-define the use of specific components, within each of our component-types, with specific versions of datasets. For example, a rule has been defined specifying that the UseComponentVersion metadata value in the Transcript_File dataset must be equal to the ComponentVersion

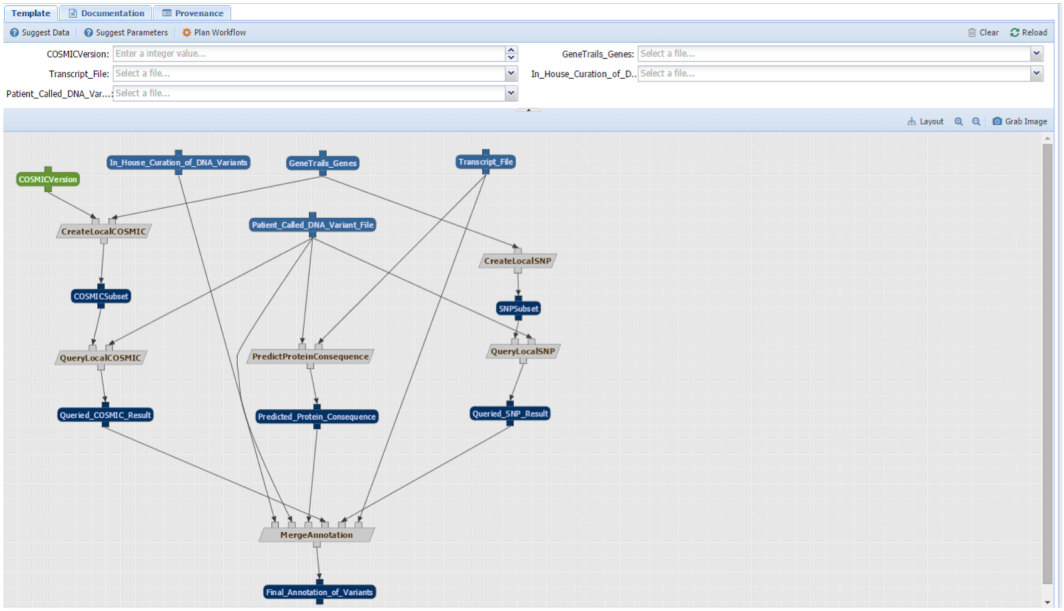


**Figure 3. WINGS Workflow Template for Our Clinical Omics Use-Case.** WINGS templates are fully connected representations of all components, datasets, and rules and constraints of an analysis pipeline needed to execute a semantically validated workflow run. A workflow template representing our clinical omics analysis pipeline is shown (A). Within our workflow template, each step is represented by its component-type (grey rectangles), however; please note that individual components can also be used to build a workflow template, sequentially connected to one another, and has all input and output datasets (blue rounded rectangles) represented. Once a workflow template is created, WINGS generates an accompanied GUI for the workflow template, thus allowing workflow users to execute workflow runs. Due the enforcement of all user-defined rules and constraints, each workflow run is semantically validated. Pre-defined rules and constraints also enables WINGS to help guide users through a workflow run by suggesting semantically validated inputs and parameters (Suggest Data and Suggest Parameters buttons). For example, due to our predefined rules and constraints, only datasets with the same genomic assembly would be suggested for this workflow template.

parameter value of the PredictProteinConsequence component used. Every component under the PredictProteinConsequence component-type has a value set for ComponentVersion, indicating its version number, and set to match the value of the UseComponentVersion metadata value a Transcript_File dataset. Thus a user is effectively choosing a specific component from a component type when choosing a specific input dataset. Similar rules have been set up for pre-defining the use of specific components within each component type. Please refer to the Supplemental Information for a full list of rules and constraints defined within our clinical omics workflow.

### Assembly of a Workflow Run

Once all datasets, components, rules and constraints are defined and created, each can be pieced together to assemble a workflow template (Figure 3). Our workflow template was assembled using only component-types; however, individual components can also be used to build a workflow template. The workflow template illustrates each step of our analysis pipeline in addition to all input and output datasets.

### Execution of a Workflow Run

Workflow users interact with WINGS in a different way from a workflow developer. Workflow users do not need to know how the workflow was developed in order to use it. Upon the creation of a workflow template, WINGS generates a GUI for workflow users to interact with and run assembled workflows (see top of Figure 3). With this GUI, users are able to choose the desired parameters and inputs for this workflow. Furthermore, through the semantic reasoning [28, 32] of pre-defined rules and constraints, the 'Suggest Parameters' and 'Suggest Data' buttons within the GUI can be used to suggest appropriate parameters and inputs, respectively, for a workflow run. This guides users effectively and accurately through a workflow run. For example, due to our pre-defined rules and constraints, upon the selection of a Patient_Called_DNA_Variant_File, WINGS would only allow the selection of additional input objects of the same genomic assembly, as specified in their individual GenomicAssembly metadata. If a user chooses an input inconsistent with the pre-defined rules and constraints, a message is displayed informing the user of the error and requiring the user to choose an alternative input. Once all parameters and inputs are provided, the workflow run can be planned and ultimately run with the 'Plan Workflow' button. As the workflow run is being executed, WINGS directs users to a user interface where the run can be monitored and, when needed, reports from code execution failures are displayed to aid in debugging workflows and the underlying code.

### Execution of Our Clinical Omics Workflow

The executed workflow plan of a successful run of our clinical omics workflow highlighting all parameters, datasets, and components used is shown in Figure 4. Particularly when component-types are used to assemble a workflow run, as in our clinical omics pipeline, this schema shows the actual components used during the execution as these may change as data inputs change. Based on the use of the same input data and versions of annotation sources, the final output from this workflow run was found to be identical (based on the use of the unix *diff* command) to the output obtained from the original analysis pipeline. Our final workflow output had the added benefits of having all run-time parameters and metadata automatically tracked and the assurance that all parameters, datasets, and components used during the analysis were consistent with all user-defined rules and constraints. Please refer to Supplemental Information for more detailed instructions on how to execute a run of our clinical omics workflow on the WINGS site.

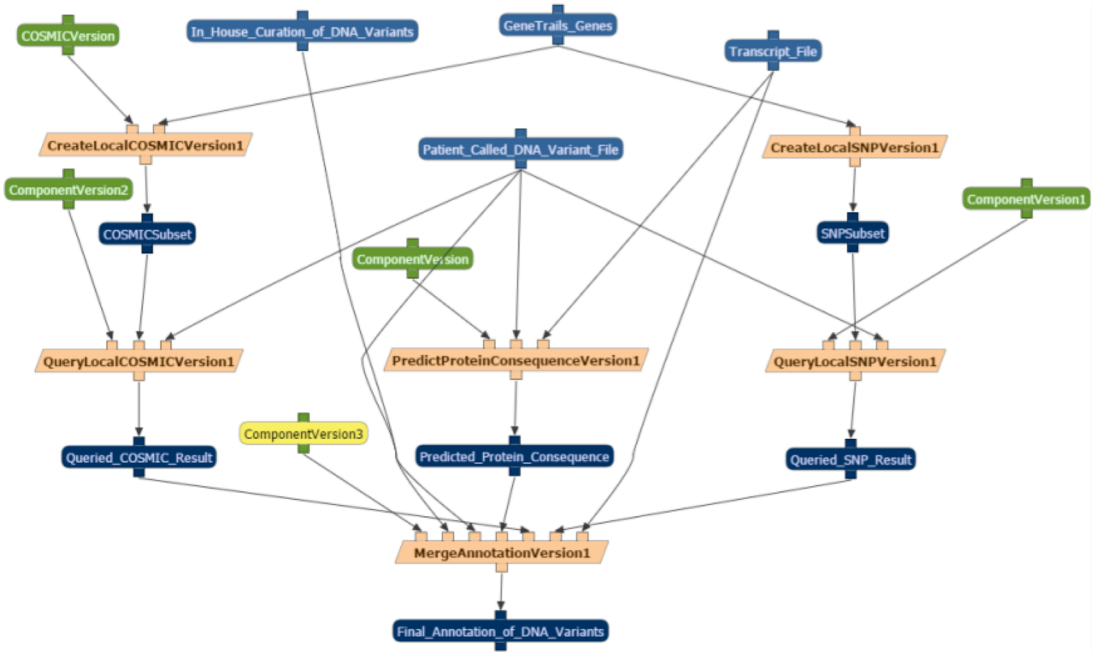


**Figure 4. Execution of Our Clinical Omics Use-Case WINGS Workflow.** Once a workflow run is executed, the details of the run are shown. Displayed is the successful execution of our clinical omics use-case WINGS workflow. All input parameters (green), input and output data objects (blue), and individual components (yellow) of the workflow run are shown. Particularly when component-types are used to define a workflow template, the details of an executed workflow run can be used to identify the exact components used for each workflow run. Based on the chosen input datasets and the user-defined rules and constraints, Version1 of each component-type was used in our executed workflow run.

## Meeting the Minimal Requirements of Transparency and Reproducibility of Clinical Omics

Based on the checklist of requirements recommended for enhanced transparency and reproducibility of translational and clinical omics defined in Table 1, our WINGS implemented clinical omics workflow met all requirements. All data, including the exact input data used, intermediate data, third party data, output data, and their provenance was captured and preserved within our implemented workflow. All code, configurations, computing environment, and their provenance were preserved along with a high level diagram illustrating all steps of the analysis. And most importantly, the user-defined rules and constraints within our workflow provided the veracity checks needed to enhance analytical validity.

# Discussion

The implementation of our clinical omics DNA variant annotation pipeline use-case within the WINGS platform is the first implementation and execution of a clinical omics pipeline in a semantic workflow. We found that the implementation of our clinical omics annotation pipeline into a semantic workflow helped us to achieve the requirements for enhanced transparency, reproducibility, and analytical accuracy recommended for translational and clinical omics. During the implementation of our clinical omics workflow, we also found many features of the WINGS system were particularly primed to support the specific needs of clinical omics analyses. These include the need to: 1) keep pace with frequent updates of biological life science databases, 2) enforce consistency and data integrity across heterogeneous biological and clinical data, 3) keep pace with rapid updates and development of omics software tools, and 4) process large omics data sets. Each is described below.

### Frequent Updates of Molecular Life Science Databases

The analysis and interpretation of omics data relies heavily on information within molecular life science databases such as those provided by the National Center for Biotechnology Information (NCBI) [55], European Molecular Biology Laboratory – European Bioinformatics Institute (EMBL-EBI) [56], and the UCSC Genome Browser [57]. Gene and transcript information supplied by NCBI's Reference Sequence (RefSeq) database [58] and EMBL-EBI Ensembl database [59] serves as the foundation of many omics studies, particularly in RNA-seq studies [60]. Databases such as dbSNP, COSMIC, and clinVAR [61] provide annotation information for DNA variants regarding their frequency within the population and potential associations with disease and clinical phenotype.

To keep pace with our growing biological knowledge, information within these databases is constantly updated. For example, RefSeq databases are updated twice a month [58], the COSMIC database is updated every 2 months [62], and new builds of dbSNP are periodically released, especially after a new genome release or after a large submission of SNPs [30]. To ensure that the most current biological knowledge is used to analyse and interpret omics data, particularly within a clinical setting, it is imperative that all provenances of the databases are effectively captured and tracked.

WINGS' ability to ***dynamically extract and propagate metadata within a component*** enhances the capture and tracking of provenance of datasets associated with frequently updated biological databases. The ability to dynamically extract metadata within a component is a new and unique feature of WINGS that helps to prevent any errors that may arise if manual intervention were needed. For example, the version of R used within each component of our clinical omics workflow is dynamically extracted at runtime and automatically propagated to the RVersionId metadata value of its output dataset. Within other workflow platforms such as Galaxy and Taverna, metadata can only be manually populated and cannot be dynamically extracted at runtime.

### Heterogeneity/Consistency of Biological Data

The analysis and interpretation of omics data also relies heavily on disparate and heterogeneous sets of biological data. For example, a typical RNA-seq analysis protocol involves two very different types of biological data: 1) the genomic sequence used for the alignment of the RNA-seq reads and 2) the annotated transcript models used for expression quantification. Within our DNA variant annotation pipeline, biological information across multiple databases is used. Thus to ensure consistency and validity across these heterogeneous data sources, it is critical that the disparate data types be consistent with one another.

The WINGS platform helps to ***ensure consistency across heterogeneous data sets through the use of its semantic technology***. For our clinical omics workflow, user-defined rules and constraints were used to ensure that all datasets were of the same genomic assembly and that specific datasets were processed using specific workflow components. Further enhancing the consistency across disparate datasets is WINGS ability to ***predefine and constrain the specific datasets allowed as input/output for each component***. Predefining and constraining the types of datasets allowed helps to maintain the integrity of the datasets used. These features to enhance data integrity and veracity are absent in other workflow platforms.

### Rapid Development of Omics Software Tools

Paralleling and at times even driven by, our growth of biological knowledge is the rapid development of new and existing omics analysis software tools. As an example, two popular short-read alignment tools, BWA [63] and Tophat [64], had a total of 7 and 3 releases, respectively, in the year 2014. For a workflow system to effectively support clinical omics, in addition to efficiently tracking the specific versions of the software used, it is critical that the integration of new or updated software tools within new or existing workflows be user-friendly and efficient.

Two features of the WINGS platform help to efficiently incorporate new tools and updates to existing tools. The first feature is WINGS' ***ability to group related components under a common component-type***: this allows components for alternative tools or updated versions of existing tools to be easily added into an existing workflow template and their use semantically enforced; related to this, the second feature is its ability to ***track the provenance of all component-types, components and workflow templates***. A timestamp and user-id is associated with the creation and update of each. Provenance for data objects is also similarly tracked.

### Processing of Large Omics Data Sets

The ability to store and process large data sets has become a mandatory part of analysing omics data, particularly as the volume and complexity of omics data continue to increase [65, 66]. WINGS' ***ability to execute workflows under a variety of modes either it be in a local host, across a network of local machines, or across large scale distributed data processing environments such as clusters or cloud services*** is an invaluable tool in processing large omics data sets.

## Conclusions

We implemented and executed a clinical omics pipeline aimed at annotating DNA variants identified through large-scale DNA sequencing using the WINGS semantic workflow system. We found the semantic workflows in WINGS capable of effectively meeting the requirements for enhanced transparency, reproducibility, and analytical validity recommended for translational and clinical omics. We further found many features of the WINGS platform particularly effective in supporting the specific needs of clinical omics analyses.

The next stage for the application of WINGS in this setting is extension to other clinical omics use cases, as well as clinical user evaluation to facilitate seamless integration in these settings. We also note that the needs for reproducibility extend beyond the clinical setting. With regard to methods development, the semantic constraints in WINGS allow for more efficient and robust dissemination of methods and workflows to the broader research community, particularly to non-expert users. The FDA's Computational Science Center has now started to receive next generation sequencing (NGS) data with regulatory submissions that must be validated and analysed, along with the corresponding methods. For FDA approval diagnostic devices, analytical validation of the device to establish performance characteristics such as analytical specificity, precision (repeatability and reproducibility), and limits of detection is essential. As such validation may require developing an algorithm or determining the threshold for clinical decisions, these steps must be captured such that the rationale and evidence for these decisions can also be evaluated. Finally, given the National Institutes of Health's initiatives to improve reproducibility, particularly in preclinical research, frameworks such as WINGS will become more and more essential to the research enterprise.

## Competing interests

The authors declare that they have no competing interests.

## Authors' contributions

CLZ implemented the workflow and drafted the manuscript. VR and YG designed/updated WINGS for use with clinical omics. SKM and YG designed the study and helped revise the manuscript. All authors read and approved the final manuscript.

## Acknowledgements

Special thanks to Drs. Christopher Corless and Carol Beadling at the OHSU Knight Diagnostics Laboratory. We gratefully acknowledge funding from the US National Science Foundation (ACI-1355475), NIH/NCATS (5UL1RR024140), NIH/NCI (5P30CA06533), and Leukemia and Lymphoma Society (7005-11).

# References


1. Saracchi E, Fermi S, Brighina L: **Emerging candidate biomarkers for Parkinson's disease: a review.** Aging Dis 2013, **5**(1):27-34.
2. Thomas L, Di Stefano AL, Ducray F: **Predictive biomarkers in adult gliomas: the present and the future.** Curr Opin Oncol 2013, **25**(6):689-694.
3. Kim Y, Kislinger T: **Novel approaches for the identification of biomarkers of aggressive prostate cancer.** Genome Med 2013, **5**(6):56.
4. Ellis MJ, Perou CM: **The genomic landscape of breast cancer as a therapeutic roadmap.** Cancer Discov 2013, **3**(1):27-34.
5. Church D, Kerr R, Domingo E, Rosmarin D, Palles C, Maskell K, Tomlinson I, Kerr D: **'Toxgnostics': an unmet need in cancer medicine.** Nat Rev Cancer 2014, **14**(6):440-445.
6. James LP: **Metabolomics: integration of a new "omics" with clinical pharmacology.** Clin Pharmacol Ther 2013, **94**(5):547-551.
7. Li H, Jia W: **Cometabolism of microbes and host: implications for drug metabolism and drug-induced toxicity.** Clin Pharmacol Ther 2013, **94**(5):574-581.
8. Lopez-Lopez E, Gutierrez-Camino A, Bilbao-Aldaiturriaga N, Pombar-Gomez M, Martin-Guerrero I, Garcia-Orad A: **Pharmacogenetics of childhood acute lymphoblastic leukemia.** Pharmacogenomics 2014, **15**(10):1383-1398.
9. Pouget JG, Muller DJ: **Pharmacogenetics of antipsychotic treatment in schizophrenia.** Methods Mol Biol 2014, **1175**:557-587.
10. Lymperopoulos A, French F: **Pharmacogenomics of heart failure.** Methods Mol Biol 2014, **1175**:245-257.
11. Chen R, Mias GI, Li-Pook-Than J, Jiang L, Lam HY, Chen R, Miriami E, Karczewski KJ, Hariharan M, Dewey FE, Cheng Y, Clark MJ, Im H, Habegger L, Balasubramanian S, O'Huallachain M, Dudley JT, Hillenmeyer S, Haraksingh R, Sharon D, Euskirchen G, Lacroute P, Bettinger K, Boyle AP, Kasowski M, Grubert F, Seki S, Garcia M, Whirl-Carrillo M, Gallardo M, Blasco MA, Greenberg PL, Snyder P, Klein TE, Altman RB, Butte AJ, Ashley EA, Gerstein M, Nadeau KC, Tang H, Snyder M: **Personal omics profiling reveals dynamic molecular and medical phenotypes.** Cell 2012, **148**(6):1293-1307.
12. Ransohoff DF: **Promises and limitations of biomarkers.** Recent Results Cancer Res 2009, **181**:55-59.
13. Ransohoff DF: **The process to discover and develop biomarkers for cancer: a work in progress.** J Natl Cancer Inst 2008, **100**(20):1419-1420.
14. Committee on the Review of Omics-Based Tests for Predicting Patient Outcomes in Clinical Trials, Board on Health Care Services, Board on Health Sciences Policy, Institute of Medicine: 2012, .

15. Zook JM, Chapman B, Wang J, Mittelman D, Hofmann O, Hide W, Salit M: **Integrating human sequence data sets provides a resource of benchmark SNP and indel genotype calls.** Nat Biotechnol 2014, **32**(3):246-251.

16. Collins FS, Hamburg MA: **First FDA authorization for next-generation sequencer.** N Engl J Med 2013, **369**(25):2369-2371.

17. FDA Public Workshop: Next Generation Sequencing Standards [http://www.fda.gov/ScienceResearch/SpecialTopics/RegulatoryScience/ucm389561.htm]

18. Baggerly KA, Coombes KR: **What information should be required to support clinical "omics" publications?** Clin Chem 2011, **57**(5):688-690.

19. Baggerly KA, Coombes KR: **DERIVING CHEMOSENSITIVITY FROM CELL LINES: FORENSIC BIOINFORMATICS AND REPRODUCIBLE RESEARCH IN HIGH-THROUGHPUT BIOLOGY.** ANNALS OF APPLIED STATISTICS} 2009}, **3}**(4}):1309-1334}.

20. Vasilevsky NA, Brush MH, Paddock H, Ponting L, Tripathy SJ, Larocca GM, Haendel MA: **On the reproducibility of science: unique identification of research resources in the biomedical literature.** PeerJ 2013, **1**:e148.

21. Begley CG, Ellis LM: **Drug development: Raise standards for preclinical cancer research.** Nature 2012, **483**(7391):531-533.

22. Anderson WP: **Reproducibility: Stamp out shabby research conduct.** Nature 2015, **519**(7542):158.

23. Garijo D, Kinnings S, Xie L, Xie L, Zhang Y, Bourne PE, Gil Y: **Quantifying reproducibility in computational biology: the case of the tuberculosis drugome.** PLoS One 2013, **8**(11):e80278.

24. Giardine B, Riemer C, Hardison RC, Burhans R, Elnitski L, Shah P, Zhang Y, Blankenberg D, Albert I, Taylor J, Miller W, Kent WJ, Nekrutenko A: **Galaxy: a platform for interactive large-scale genome analysis.** Genome Res 2005, **15**(10):1451-1455.

25. Oinn T, Addis M, Ferris J, Marvin D, Senger M, Greenwood M, Carver T, Glover K, Pocock MR, Wipat A, Li P: **Taverna: a tool for the composition and enactment of bioinformatics workflows.** Bioinformatics 2004, **20**(17):3045-3054.

26. Jagtap PD, Johnson JE, Onsongo G, Sadler FW, Murray K, Wang Y, Shenykman GM, Bandhakavi S, Smith LM, Griffin TJ: **Flexible and Accessible Workflows for Improved Proteogenomic Analysis Using the Galaxy Framework.** J Proteome Res 2014, .

27. Gil Y, McWeeney S, Mason CE: **Using Semantic Workflows to Disseminate Best Practices and Accelerate Discoveries in Multi-Omic Data Analysis.** AAAI Workshop on Expanding the Boundaries of Health Informatics using AI (HIAI) 2013, .

28. Gil Y, Ratnakar V, Kim J, Gonzalez-Calero PA, Groth P, Moody J, Deelman E: **Wings: Intelligent Workflow-Based Design of Computational Experiments.** IEEE Intelligent Systems 2011, **26**(1).

29. Forbes SA, Bindal N, Bamford S, Cole C, Kok CY, Beare D, Jia M, Shepherd R, Leung K, Menzies A, Teague JW, Campbell PJ, Stratton MR, Futreal PA: **COSMIC: mining complete cancer genomes in the Catalogue of Somatic Mutations in Cancer.** Nucleic Acids Res 2011, **39**(Database issue):D945-50.

30. Bhagwat M: **Searching NCBI's dbSNP database.** Curr Protoc Bioinformatics 2010, **Chapter 1**:Unit 1.19.

31. Obenchain V, Lawrence M, Carey V, Gogarten S, Shannon P, Morgan M: **VariantAnnotation: a Bioconductor package for exploration and annotation of genetic variants.** Bioinformatics 2014, **30**(14):2076-2078.

32. Gil Y, Gonzalez-Calero PA, Kim J, Moody J, Ratnakar V: **A Semantic Framework for Automatic Generation of Computational Workflows Using Distributed Data and Component Catalogs.** Journal of Experimental and Theoretical Artificial Intelligence 2011, .

33. Ison J, Kalas M, Jonassen I, Bolser D, Uludag M, McWilliam H, Malone J, Lopez R, Pettifer S, Rice P: **EDAM: an ontology of bioinformatics operations, types of data and identifiers, topics and formats.** Bioinformatics 2013, **29**(10):1325-1332.

34. Eilbeck K, Lewis SE, Mungall CJ, Yandell M, Stein L, Durbin R, Ashburner M: **The Sequence Ontology: a tool for the unification of genome annotations.** Genome Biol 2005, **6**(5):R44.

35. Mungall CJ, Batchelor C, Eilbeck K: **Evolution of the Sequence Ontology terms and relationships.** J Biomed Inform 2011, **44**(1):87-93.

36. Gil Y: **Intelligent Workflow Systems and Provenance-Aware Software.** In Proceedings of the Seventh International Congress on Environmental Modeling and Software, San Diego, CA 2014, .

37. Gil Y, Ratnakar V, Deelman E, Mehta G, and Ki J: **Wings for Pegasus: Creating Large-Scale Scientific Applications Using Semantic Representations of Computational Workflows .** Proceedings of the 19th Annual Conference on Innovative Applications of Artificial Intelligence (IAAI), Vancouver, British Columbia, Canada, July 22-26, 2007 .

38. Gil Y: **Mapping Semantic Workflows to Alternative Workflow Execution Engines.** Proceedings of the 7th IEEE International Conference on Semantic Computing (ICSC), 2013 .

39. Mattmann C, Crichton D, Medvidovic N, and Hughes S: **A Software Architecture-Based Framework for Highly Distributed and Data Intensive Scientific Applications .** Proceedings of the 28th International Conference on Software Engineering (ICSE06), pp. 721-730, Shanghai, China, 2006 .

40. Deelman E, Singh G, Su MH, Blythe J, Gil Y, Kesselman C, Mehta G, Vahi K, Berriman GB, Good J, Laity A, Jacob JC, and Katz DS: **Pegasus: a Framework for Mapping Complex Scientific Workflows onto Distributed Systems.** Scientific Programming Journal 2005, **13**:219-237.

41. Gil Y, Ratnakar V, Verma R, Hart A, Ramirez P, Mattmann C, Sumarlidason A, and Park SL: **Time-Bound Analytic Tasks on Large Datasets through Dynamic Configuration of Workflows. .** In Proceedings of the Eighth Workshop on Workflows in Support of Large-Scale Science (WORKS), held in conjunction with SC 2013, Denver, CO, 2013 .

42. Moreau L, Clifford B, Freire J, Futrelle J, Gil Y, Groth P, Kwasnikowska N, Miles S, Missier P, Myers J, Plale B, Simmhan Y, Stephan E, and denBussche JV: **The Open Provenance Model Core Specification (v1.1).** Future Generation Computer Systems 2011, **27**(6).

43. Garijo D, and Gil Y: **A New Approach for Publishing Workflows: Abstractions, Standards, and Linked Data .** Proceedings of the 6th Workshop on Workflows in Support of Large-Scale Science (WORKS-11), held in conjunction with SC-11, Seattle, WA, Nov. 12-18 2011 .

44. Garijo D, Gil Y, and Corcho O: **Towards Workflow Ecosystems Through Semantic and Standard Representations .** Proceedings of the Ninth Workshop on Workflows in Support of Large-Scale Science (WORKS), held in conjunction with the IEEE ACM International Conference on High-Performance Computing (SC), New Orleans, LA, 2014 .

45. Dinov I, Van Horn JD, Lozev KM, Magsipoc R, Petrosyan P, Liu Z, MacKenzie-Graham A, Eggert P, Parker DS, and Toga AW: **Efficient, Distributed and Interactive Neuroimaging Data Analysis using the LONI Pipeline.** Front. Neuroinform 2009, **3**(22):1-10.

46. Garijo D, Corcho O, Gil Y, Gutman BA, Dinov ID, Thompson P, and Toga AW: **FragFlow: Automated Fragment Detection in Scientific Workflows. .** Proceedings of the IEEE Conference on e-Science, Guarujua, Brazil, 2014 .

47. Garijo D, and Gil Y: **Augmenting PROV with Plans in P-PLAN: Scientific Processes as Linked Data Second .** International Workshop on Linked Science: Tackling Big Data (LISC), held in conjunction with the International Semantic Web Conference (ISWC), Boston, MA, 2012 .

48. Rocca-Serra P, Brandizi M, Maguire E, Sklyar N, Taylor C, Begley K, Field D, Harris S, Hide W, Hofmann O, Neumann S, Sterk P, Tong W, and Sansone SA: **ISA software suite: supporting standards-compliant experimental annotation and enabling curation at the community level .** Bioinformatics 2010, **26**(18):2354-2356.

49. De Roure D, Goble C, and Stevens R: **The Design and Realisation of the myExperiment Virtual Research Environment for Social Sharing of Workflows.** Future Generation Computer Systems 2009, **25**:561-567.

50. Mates P, Santos S, Freire J, and Silva CT: **CrowdLabs: social analysis and visualization for the sciences.** Proceeding SSDBM'11 Proceedings of the 23rd international conference on Scientific and statistical database management 2011, :555-564.

51. Reich M, Liefeld T, Gould J, Lerner J, Tamayo P, Mesirov JP: **GenePattern 2.0.** Nat Genet 2006, **38**(5):500-501.

52. Gentleman RC, Carey VJ, Bates DM, Bolstad B, Dettling M, Dudoit S, Ellis B, Gautier L, Ge Y, Gentry J, Hornik K, Hothorn T, Huber W, Iacus S, Irizarry R, Leisch F, Li C, Maechler M, Rossini AJ, Sawitzki G, Smith C, Smyth G, Tierney L, Yang JY, Zhang J: **Bioconductor: open software development for computational biology and bioinformatics.** Genome Biol 2004, **5**(10):R80.

53. Wilkinson MD, Vandervalk B, McCarthy L: **The Semantic Automated Discovery and Integration (SADI) Web service Design-Pattern, API and Reference Implementation.** J Biomed Semantics 2011, **2**(1):8-1480-2-8.

54. Reeves GA, Eilbeck K, Magrane M, O'Donovan C, Montecchi-Palazzi L, Harris MA, Orchard S, Jimenez RC, Prlic A, Hubbard TJ, Hermjakob H, Thornton JM: **The Protein Feature Ontology: a tool for the unification of protein feature annotations.** Bioinformatics 2008, **24**(23):2767-2772.

55. NCBI Resource Coordinators: **Database resources of the National Center for Biotechnology Information.** Nucleic Acids Res 2014, **42**(Database issue):D7-17.

56. Brooksbank C, Bergman MT, Apweiler R, Birney E, Thornton J: **The European Bioinformatics Institute's data resources 2014.** Nucleic Acids Res 2014, **42**(Database issue):D18-25.

57. Karolchik D, Barber GP, Casper J, Clawson H, Cline MS, Diekhans M, Dreszer TR, Fujita PA, Guruvadoo L, Haeussler M, Harte RA, Heitner S, Hinrichs AS, Learned K, Lee BT, Li CH, Raney BJ, Rhead B, Rosenbloom KR, Sloan CA, Speir ML, Zweig AS, Haussler D, Kuhn RM, Kent WJ: **The UCSC Genome Browser database: 2014 update.** Nucleic Acids Res 2014, **42**(Database issue):D764-70.

58. Pruitt KD, Tatusova T, Brown GR, Maglott DR: **NCBI Reference Sequences (RefSeq): current status, new features and genome annotation policy.** Nucleic Acids Res 2012, **40**(Database issue):D130-5.

59. Fernandez-Suarez XM, Schuster MK: **Using the ensembl genome server to browse genomic sequence data.** Curr Protoc Bioinformatics 2010, **Chapter 1**:Unit1.15.

60. Wu PY, Phan JH, Wang MD: **Assessing the impact of human genome annotation choice on RNA-seq expression estimates.** BMC Bioinformatics 2013, **14 Suppl 11**:S8-2105-14-S11-S8. Epub 2013 Nov 4.

61. Landrum MJ, Lee JM, Riley GR, Jang W, Rubinstein WS, Church DM, Maglott DR: **ClinVar: public archive of relationships among sequence variation and human phenotype.** Nucleic Acids Res 2014, **42**(Database issue):D980-5.

62. Forbes SA, Bhamra G, Bamford S, Dawson E, Kok C, Clements J, Menzies A, Teague JW, Futreal PA, Stratton MR: **The Catalogue of Somatic Mutations in Cancer (COSMIC).** Curr Protoc Hum Genet 2008, **Chapter 10**:Unit 10.11.

63. Li H, Durbin R: **Fast and accurate short read alignment with Burrows-Wheeler transform.** Bioinformatics 2009, **25**(14):1754-1760.

64. Trapnell C, Pachter L, Salzberg SL: **TopHat: discovering splice junctions with RNA-Seq.** Bioinformatics 2009, **25**(9):1105-1111.

65. Qu H, Fang X: **A brief review on the Human Encyclopedia of DNA Elements (ENCODE) project.** Genomics Proteomics Bioinformatics 2013, **11**(3):135-141.

66. Tomczak K, Czerwinska P, Wiznerowicz M: **The Cancer Genome Atlas (TCGA): an immeasurable source of knowledge.** Contemp Oncol (Pozn) 2015, **19**(1A):A68-77.

# Supplemental Materials

## Dataset Metadata

GeneTrails_Genes:

- version (int): version number
- source (string): data source (i.e., Hugo Gene Nomenclature Committee)

COSMICSubset:

- COSMICVersionId (int): version of COSMIC
- GenomicAssembly (string): genomic assembly of genomic coordinates
- UsedComponentVersion (int): version number of *CreateLocalCOSMICSubset* used
- UseComponentVersion (int): version number of *QueryLocalCOSMIC* to be used
- Versions of R and R packages used to create the data object (string)

SNPSubset:

- dbSNPVersionId (int): version of dbSNP
- GenomicAssembly (string): genomic assembly of genomic coordinates
- UsedComponentVersion (int): version number of *CreateLocalSNPSubset* used
- UseComponentVersion (int): version number of *QueryLocalSNP* to be used
- Versions of R and R packages used to create the data object (string)

Patient_Called_DNA_Variant_File:

- PatientId (int): unique identifier for patient sample
- GenomicAssembly (string): genomic assembly of genomic coordinates

Queried_COSMIC_Result:

- PatientId (int): unique identifier for patient sample

- COSMICVersionId: version COSMIC
- GenomicAssembly (string): genomic assembly of genomic coordinates
- UsedComponentVersion (int): version number of *QueryLocalCOSMIC* used
- Versions of R and R packages used to create the data object (string)

Queried_SNP_Result:

- PatientId (int): unique identifier for patient sample
- dbSNPVersionId: version of dbSNP
- GenomicAssembly (string): genomic assembly of genomic coordinates
- UsedComponentVersion (int): version number of *QueryLocalCOSMIC* used
- Versions of R and R packages used to create the data object (string)

Transcript_File:

- TranscriptSource (string): data source (i.e., NCBI RefSeq)
- TranscriptFileVersionId (int): version of transcript_file
- GenomicAssembly (string): genomic assembly of genomic coordinates
- UseComponentVersion (int): version of *PredictProteinConsequence* to be used

Predicted_Protein_Consequence:

- PatientId (int): unique identifier for patient sample
- GenomicAssembly (string): genomic assembly of genomic coordinates
- UsedComponentVersion (int): version number of *PredictProteinConsequence* used
- TranscriptFileVersionId (int): version of transcript_file used
- Versions of R and R packages used to create the data object (string)

In_House_Curation_of_DNA_Variants:

- InHouseCurationVersionId (int): version of In_House_Curation_of_DNA_Variants
- GenomicAssembly (string): genomic assembly of genomic coordinates
- UseComponent (int): version of *MergeAnnotation* to be used

Final_Annotation_of_DNA_Variants:

- PatientId (int): unique identifier for patient sample
- GenomicAssembly (string): genomic assembly of genomic coordinates
- UsedComponentVersion (int): version number of *MergeAnnotation* used
- Versions of R and R packages used to create the data object (string)

## Semantic Rules and Constraints

Rules and constraints were encapsulated within individual workflow component-types/components. Below is a list of all semantic rules and constraints.

*CreateLocalCOSMIC:* 1) Have the version of COSMIC be propagated to the COSMICVersionId metadata for the COSMICSubset data-object; 2) Obtain and

propagate the version of R and R packages onto their respective metadata values for the COSMICSubset data-object.

*CreateLocalSNP*: 1) Have the version of dbSNP be propagated to the dbSNPVersionId metadata for the SNPSubset data-object; 2) Obtain and propagate the version of R and R packages onto their respective metadata values for the SNPSubset data-object.

*QueryLocalCOSMIC*: 1) The GenomicAssembly metadata of the Local_COSMIC_subset data-object must be the same as the GenomicAssembly metadata of the Patient_Called_DNA_Variant_File data-object; 2) Have the PatientId metadata from the Patient_Called_DNA_Variant_File data-object be propagated to the PatiendId metadata for the Queried_COSMIC_Result data-object; 3) Have the GenomicAssembly metadata from the Patient_Called_DNA_Variant_File data-object be propagated to the GenomicAssemly metadata for the Queried_COSMIC_Result data-object; 4) Have the UseComponentVersion metadata for the Local_COSMIC_Subset data-object be the same as the ComponentVersion parameter of the *QueryLocalCOSMIC* component; 5) Have the ComponentVersion parameter of the *QueryLocalCOSMIC* component used be propagated to the UsedComponentVersion metadata of the Query_COSMIC_Result data-object; 6) Obtain and propagate the version of R and R packages onto their respective metadata values for the Queried_COSMIC_Result data-object.
*QueryLocalSNP*: 1) The GenomicAssembly metadata of the Local_SNP_subset data-object must be the same as the GenomicAssembly metadata of the Patient_Called_DNA_Variant_File data-object; 2) Have the PatientId metadata from the Patient_Called_DNA_Variant_File data-object be propagated to the PatiendId metadata for the Queried_SNP_Result data-object; 3) Have the GenomicAssembly metadata from the Patient_Called_DNA_Variant_File data-object be propagated to the GenomicAssemly metadata for the Queried_SNP_Result data-object; 4) Have the UseComponentVersion metadata for the Local_SNP_Subset data-object be the same as the ComponentVersion parameter of the *QueryLocalSN*P component; 5) Have the ComponentVersion parameter of the *QueryLocalSNP* component used be propagated to the UsedComponentVersion metadata of the Query_SNP_Result data-object; 6) Obtain and propagate the version of R and R packages onto their respective metadata values for the Queried_SNP_Result data-object.

*PredictProteinConsequence*: 1) The GenomicAssembly metadata of the Transcript_File data-object must be the same as the GenomicAssembly metadata of the Patient_Called_DNA_Variant_File data-object; 2) Have the PatientId metadata from the Patient_Called_DNA_Variant_File data-object be propagated to the PatiendId metadata for the Predicted_Protein_Consequence data-object; 3) Have the GenomicAssembly metadata from the Patient_Called_DNA_Variant_File data-object be propagated to the GenomicAssemly metadata for the Predicted_Protein_Consequence data-object; 4) Have the UseComponentVersion

metadata for the Transcript_File data-object be the same as the ComponentVersion parameter of the *PredictProteinConsequence* component; 5) Have the ComponentVersion parameter of the *PredictProteinConsequence* component used be propagated to the UsedComponentVersion metadata of the Predicted_Protein_Consequence data-object; 6) Obtain and propagate the version of R and R packages onto their respective metadata values for the Predicted_Protein_Consequence data-object.

*MergeAnnotation*: 1) The GenomicAssembly metadata of the In_House_Curation_of_DNA_Variants data-object must be the same as the GenomicAssembly metadata of the Patient_Called_DNA_Variant_File data-object; 2) Have the PatientId metadata from the Patient_Called_DNA_Variant_File data-object be propagated to the PatiendId metadata for the Final_Annotation_of_DNA_Variants data-object; 3) Have the GenomicAssembly metadata from the Patient_Called_DNA_Variant_File data-object be propagated to the GenomicAssemly metadata for the Final_Annotation_of_DNA_Variants data-object; 4) Have the UseComponentVersion metadata for the In_House_Curation_of_DNA_Variants data-object be the same as the ComponentVersion parameter of the *MergeAnnotation* component; 5) Have the ComponentVersion parameter of the *MergeAnnotation* component used be propagated to the UsedComponentVersion metadata of the Final_Annotation_of_DNA_Variants data-object; 6) Obtain and propagate the version of R and R packages onto their respective metadata values for the Final_Annotation_of_DNA_Variants data-object.

## Running our Clinical Omics Workflow on the WINGS Public Site

Please following the instructions below to access and run the clinical omics workflow described in this manuscript.

Use the following link to access our workflow:

http://www.wings-workflows.org/wings-portal/users/genmed/ClinicalOmics/workflows

with the following credentials:

Username: genmed
Password: genmed123

After logging onto the portal, you will be directed to the 'VariantAnnotation' workflow template which is the genomic annotation portion of the workflow described in our manuscript. For simplicity and access requirements (i.e., credentials

are needed to download COSMIC data), we have pre-run the 'CreateLocalCOSMIC' and 'CreateLocalSNP' components thus the 'COSMICSubset' and 'SNPSubset' data sets are available for use with the workflow. To execute a run of the workflow, the appropriate data sets must be provided. With the semantic enforcement of pre-defined rules and constraints, WINGS helps guide users to the use of the appropriate data sets with 1) the use the 'SuggestData' feature wherein set(s) of semantically validated data sets are provided or 2) the use of individual pull drop down menus. Error messages warn users of the use of inconsistent/incorrect data sets. Once all data sets have been chosen, the 'Plan Workflow' button can be used to detail the exact components to be used during the workflow run. This is particularly informative in cases where workflow templates are built using component types (as exemplified by our current workflow template). When a workflow is selected, the details of workflow run will be displayed. Users can then run the workflow by pressing the 'Run Selected Workflow' button. A window will then pop up directing users to the 'Access Runs' page to monitor the execution. Users can return to this page at any time by selecting the 'Access Run' tab under the 'Analysis' tab on the top of the page. The 'Access Runs' page can also be used to access and save data objects generated during a workflow run. Users can view or save any generated data objects. Saved data objects can be viewed using the 'Manage Data' tab under the 'Advanced' tab on the top of the page. To execute additional runs of this workflow template, please access the 'Run Workflows' tab under the 'Analysis' tab on the top of the page. For more detailed information on individual components and/or data sets, please access their respective tabs under the 'Advanced' tab on the top of the page.

For a more detailed tutorial on the WINGS system, please see the following URL:

http://www.wings-workflows.org/tutorial