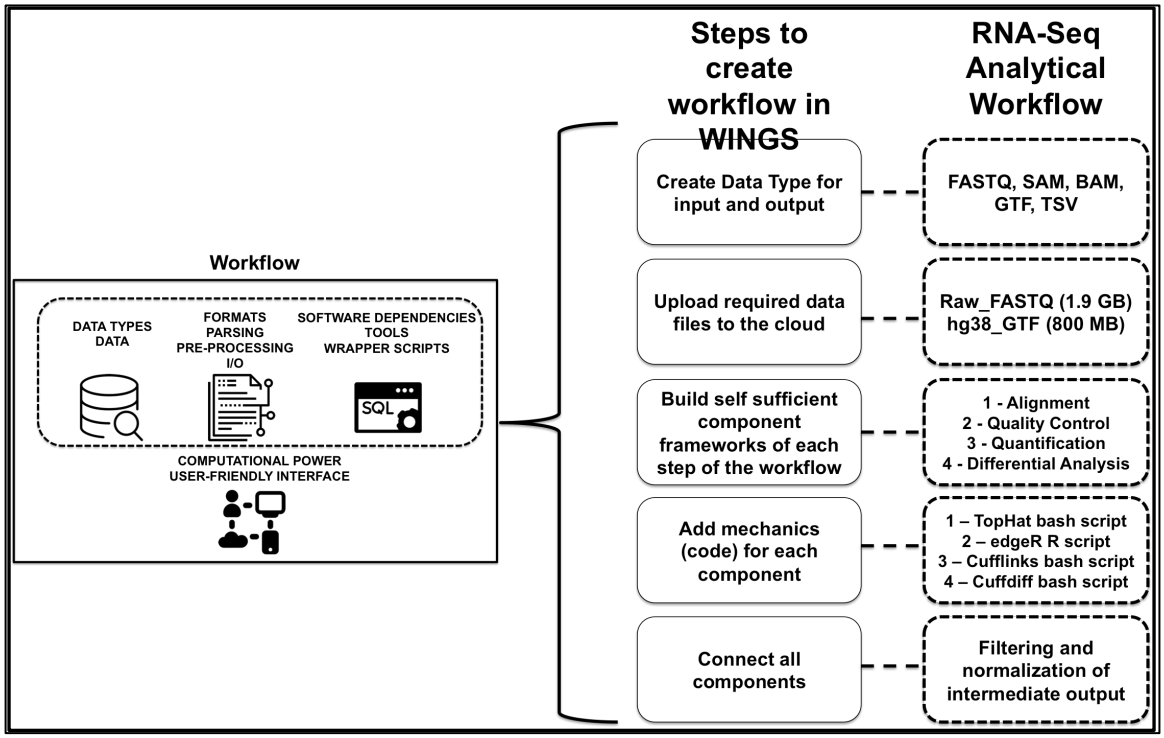
1.1 Detailed protocol for building an environment and designing workflows in WINGS



Supplementary Fig. 1. Steps to create an entire workflow in the WINGS framework, each exemplified with the steps in a real world RNA-Seq analysis protocol.

Below we navigate through a typical workflow construction protocol in WINGS, in the context of a traditional bioinformatics workflow – The processing and analysis of RNA-Seq data (**Supplementary Figure 1**)

Create data type and upload raw data

As mentioned previously, each workflow is built with components, each of which will have inputs and outputs. Every input and output of a component and subsequently a workflow is assigned a data type. This helps organize input, intermediate and output data better, and as a result, facilitates reproducibility and reusability. Thus, instantiation of each workflow begins with creating data types for all inputs, intermediate and final outputs as well as uploading the initial input data under the newly created data type. For example, in the case of RNA data analysis, data types would be created for input raw FASTQ read files, SAM and BAM intermediate outputs and GTF and TSV final outputs. Then the data to be processed will be uploaded under the FASTQ files category.

Build self-sufficient components and add utilities and scripts

Each workflow is structured with different steps, which are contained within a

component. To each component, a script will be uploaded with predetermined input, output and parameters. This script can be written in a wide array of languages but will be wrapped in a template bash script that interacts with the WINGS architecture. These scripts are contained within the workflow, and are self sufficient, thus to execute each completed component, the user only needs to specify chosen input files and parameters. Use of any popular open source databases, such as ones detailing genomic details or known mutations will exist under the WINGS framework and can be accessed by version number. The two main advantages of components that can be "black boxed" and use a central back –end data repository are that (1) alternative components performing the same function can easily be "swapped in" to account for newer tools and optimization to perform the same function and potential benchmarking tasks and (2) it enables standardization of all facets of a workflow, allowing easy comparison between input files and change in parameters. In the case of the RNA-Seq processing workflow, a few self-sufficient components to be created would be alignment, quantification and differential analysis components. These will house scripts utilizing TopHat, Cufflinks and EdgeR respectively. Alternatives to each of these tools using the same input and providing the same final output would be STAR alignment, featureCounts quantification and DESEq differential analysis. In the context of DREAM challenges and the like, existing workflows from contestants and winners can be maintained with the latest version of back-end data and the involved tools.

Connecting all components to create workflow

The penultimate step in workflow creation will be connecting all components together to produce the final output. WINGS affords a link to every input and output file as well parameters, from each component. Thus, inputs can be re-used (if multiple components need the same input) and outputs can be re-purposed as inputs to downstream components. Additionally, a component can be executed multiple times on a set of files and the resulting outputs can be merged and used and downstream input. WINGS names intermediate and final outputs with apropos names resulting in well segregated data. Specifically for a challenge scenario, entire workflows can be shared using Docker containers and reutilized without any intervention as data and scripts have been coalesced in the Docker image. In the example of the RNA-Seq analysis, components that align (input is raw FASTQ files and outputs are aligned BAM files) and components that quantify (input is sorted BAM files and output is TSV read count files) can be connected, and the intermediate output (TSV read count files) can be fed into a normalizing component along with other genomic information (genome length) that produces final expression set for the experimental samples.