

# Bwb tutorial: Agenda

1. Presentation
2. Demo
3. Q&A

Code: <https://github.com/BioDepot/BioDepot-workflow-builder>

Container: <https://hub.docker.com/r/biodepot/bwb>

YouTube of tutorial: [https://www.youtube.com/watch?v=r\\_03\\_UG1mBg](https://www.youtube.com/watch?v=r_03_UG1mBg)

Building containerized workflows for RNA-seq data using the BioDepot-workflow-Builder (Bwb). Hung et al. [bioRxiv 099010](https://doi.org/10.1101/099010)



# Using BioDepot-workflow-Builder (Bwb) to create and execute reproducible bioinformatics workflows

Ling-Hong Hung, [lhung@uw.edu](mailto:lhung@uw.edu)

Wes Lloyd, [wlloyd@uw.edu](mailto:wlloyd@uw.edu)

Ka Yee Yeung, [kayee@uw.edu](mailto:kayee@uw.edu)

University of Washington Tacoma



# Motivation

Transitioning from laptop/desktop/cluster based analyses to cloud based analyses

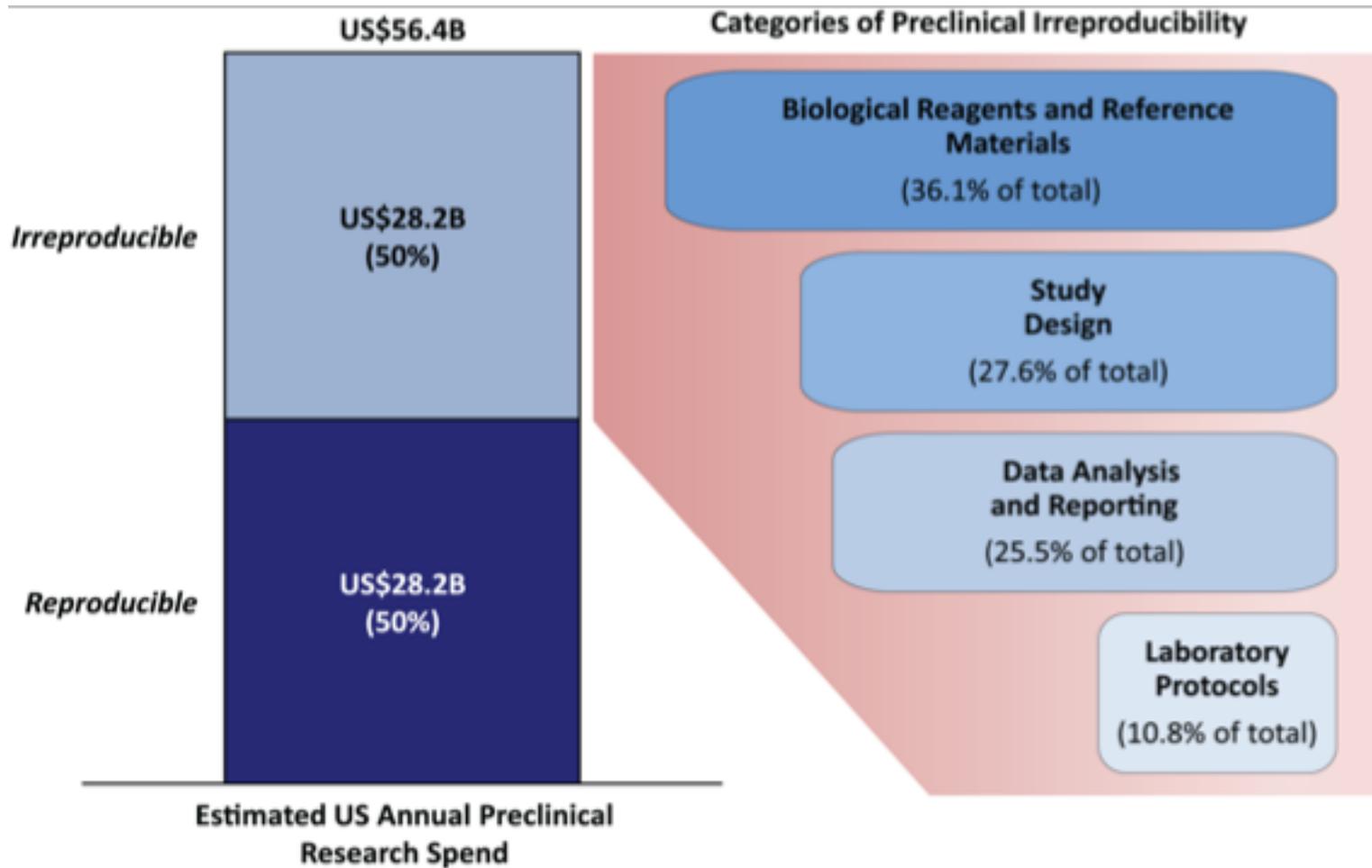
How do we facilitate sharing new workflows reproducibly?

How do we facilitate executing workflows on the cloud?

How do we get adoption and input from biologists?



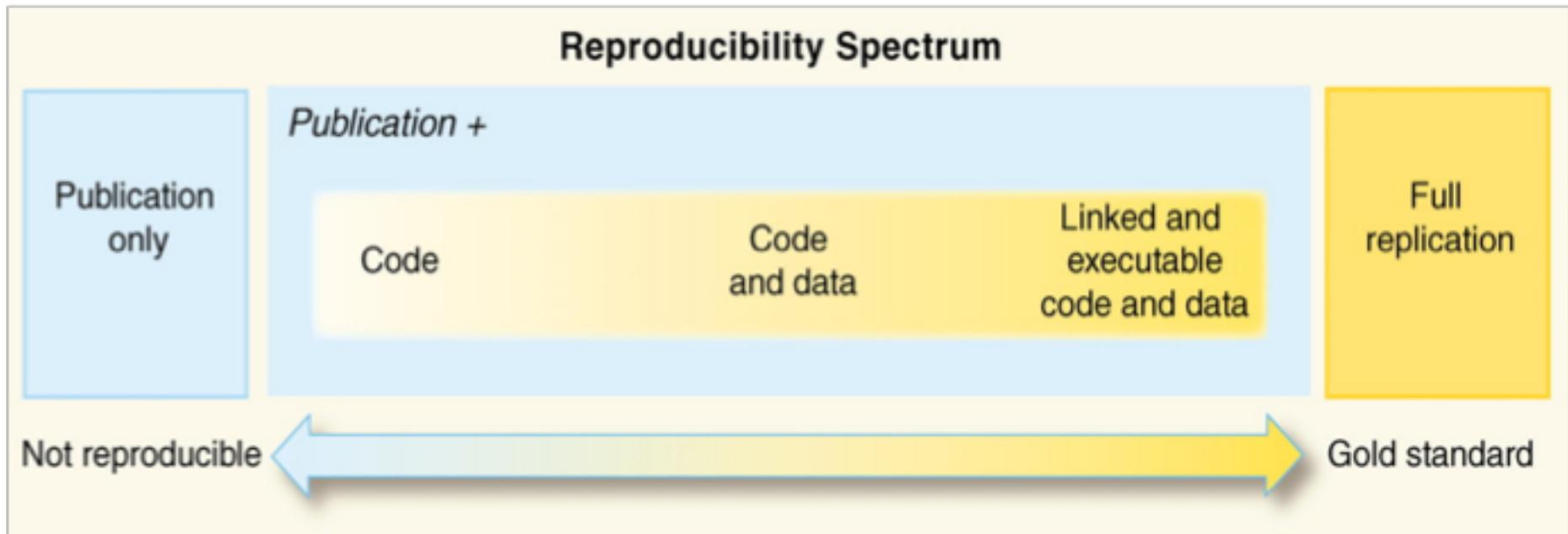
# Reproducibility in biology



Freedman et al. "The Economics of Reproducibility in Preclinical Research." PLOS Biology 2015



# The spectrum of reproducibility



Peng (2011). Reproducible research in computational science.  
*Science*, 334(6060), 1226-1227.



# Reproducibility exercise on a DREAM Challenge

Students in a 500-level bioinformatics class were asked to reproduce results from a crowd sourcing challenge in 2016-17.

Given: time series gene expression data (time 0, 24 hours) across 4 different viruses (H1N1, H3N2, RSV, Rhinovirus) in 7 studies.

Goal: build predictors to distinguish people who become contagious after exposure to flu and other respiratory viruses.



# Question: How many do you think we can reproduce?

## Challenges:

- Missing files: e.g. missing pre-processed data, need access to Google drive
- Environment setup: e.g. version conflict, memory, disk space
- Code issues: e.g. don't have software, code simply doesn't run, missing library

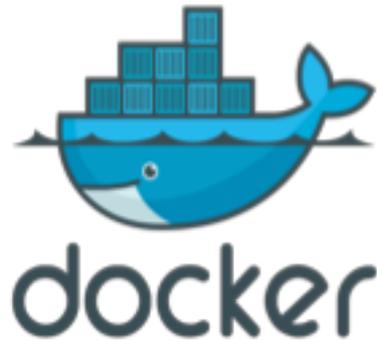


# Challenges in reproducing bioinformatics analysis results

- Parameters not given or incorrectly given
- Environment assumed i.e. directory structure
- Configuration assumed
- Different OS/software versions
- Bioinformatics analyses usually involve a pipeline consisting of many tools.
- These tools often have graphical user interfaces.

**Our solution:** We share the computing environment, not just code with Bwb.





- Docker containers wrap up a piece of software in a complete file system that contains everything it needs to run: code, runtime, system tools, system libraries – anything you can install on a server.
- It will give the same results regardless of the host environment
- Docker Hub: repository of Docker containers

<https://www.docker.com/what-docker>



# BioDepot-workflow-Builder (Bwb)

- **Drag-and-drop** form based interface
- **Modular:** Build modular bioinformatics workflows using widgets. Mix-and-match.
- **Reproducible results:** each module is encapsulated by a software container, computing environment and parameters are maintained
- **Extensible:** can add easily new widgets using a form-based builder.
- Support graphical output in modules
- Not affected by third party software upgrades.
- Easily deployed across different platforms.



# Drag-and-drop GUI

The screenshot displays a noVNC interface for a workflow named "Demo\_kallisto\_ows". The workflow diagram consists of several steps:

- Download sleuth directory (Finished)
- Download reference sequence (Finished)
- kallisto index (Finished)
- kallisto quant (Finished)
- sleuth (Finished)
- gnumeric (Running...)

Intermediate steps and dependencies include:

- Download fastq files (Finished) triggered by "Download reference sequence".
- kallisto index triggered by "Download fastq files" and "Download sleuth directory".
- kallisto quant triggered by "kallisto index".
- sleuth triggered by "kallisto quant".
- gnumeric triggered by "sleuth".

Connections are labeled with "directory → trigger" and "outputFile → inputFile".

**Quick menu**

Search for widget or select file...

- bash\_utils
- bioc\_R
- Python3
- Python2
- java8
- Perl

**Perl (from Scripting)**

Minimum perl container

Inputs:

- inputFile
- Trigger

Outputs:

- OutputDir

**ToolDock**

Perl  
Minimum perl container

**Widgets not in saved kallisto workflow**

- Perl
- jupyter\_base
- deseq2



# Widgets

- Represent modular units in a workflow, such as an executable, script or a basic operation.
- Each widget executes a Docker container
- Can be dragged to the canvas and connected to form workflows.

The screenshot displays a noVNC interface for a workflow management system. The main canvas shows a workflow titled "kallisto-sleuth workflow" with several widgets connected by arrows. The widgets include "Download sleuth directory", "Download reference sequence", "kallisto index", "kallisto quant", "sleuth", and "gnomic". A "Quick menu" is open, showing a search bar and a list of widgets: "bash\_utils", "taxi", "Python3", "Python2", "java8", and "perl". The "perl" widget is selected, and a "Perl from scripting" dialog box is open, showing the "Minimum perl container" and its inputs and outputs. The interface also features a "ToolDock" on the left with various utility icons and a "Widgets not in saved kallisto workflow" section on the right with icons for "Perl", "jupyter\_base", and "deseq2".



# Workflows

When the pipeline is to be run, the Bwb/Orange engine follows the graph, executing each widget and then propagating the output to connected widgets which are then executed, until the entire graph is traversed.

The screenshot shows the Orange3 workflow editor interface. The main workspace contains a workflow graph titled "kallisto-sleuth workflow" with the following steps:

- Download sleuth directory (Finished)
- Download reference sequence (Finished)
- Download test files (Finished)
- kallisto index (Finished)
- kallisto quant (Finished)
- sleuth (Finished)
- gnomic (Running...)

Connections between widgets are labeled with "directory = trigger" and "output\_file = inputfile". A "Quick menu" is open, showing a search for widgets and a list of available widgets including bash\_utils, bioc\_2, R, Python3, Java8, and Perl. A "ToolDock" is visible on the left side of the interface.

Below the workflow graph, there is a section titled "Widgets not in saved kallisto workflow" which lists three widgets: Perl, Jupyter\_base, and demseq2.



# Running Bwb

Bwb can be deployed on your local host or any cloud platform with Docker installed.

## 1. Download the Bwb Docker image

```
docker pull biodepot/bwb:latest
```

## 2. Start the Bwb container

```
docker run --rm -it -p 6080:6080 -v ${PWD}:/data \
-v /var/run:/var/run -v /tmp/.X11-unix:/tmp/.X11-unix biodepot/bwb
```

To access the container open up a browser window and type in localhost (or the IP of the container) followed by the port number 6080 i.e. localhost:6080



# Bwb: Implementation details

- Drag-and-drop interface based on OrangeML library (<http://orange.biolab.si>)
- OrangeML: widget is a manually written Python file
- Bwb widgets
  - stores the widget definition in 3 JSON (JavaScript Object Notation) files, 1 icon file and 1 auto-generated Python file in a directory.
  - parameters specified in widgets stored in JSON and xml files.
- Wrapped the Bwb interface in a GUiDock-VNC [Mittal et al. 2017] container.



# RNA-seq case study

## Differential expression of genes:

(count the number of transcripts made by genes under different conditions)

- align short sequence reads to genome sequenc to identify transcript (intensive C/C++ executables)
- statistical analyses to calculate which transcripts/genes are differentially expressed (R or Python libraries)
- visualization (R/Python/Jupyter

## Example

Kallisto-sleuth RNA-seq workflow



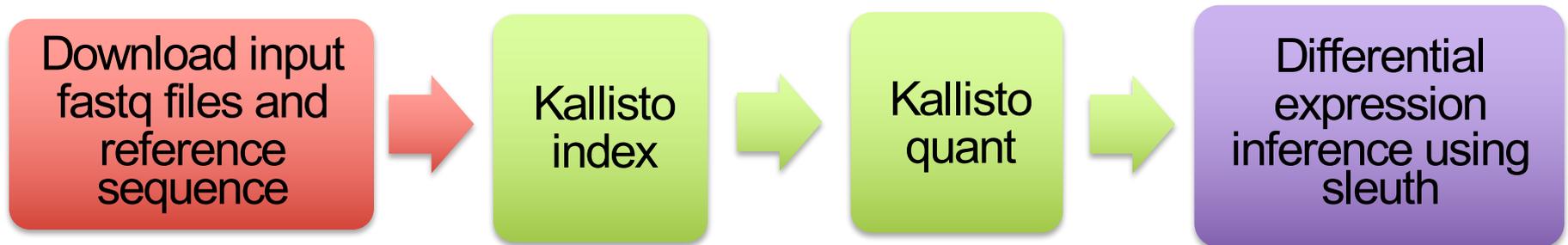
# Ex: RNA-seq workflow using Kallisto and Sleuth [Pachter Lab @ Caltech]

- Kallisto [Bray et al. 2016]
  - a program for fast RNA-Seq quantification based on pseudo-alignment.
  - Can quantify 30 million human reads in less than 3 minutes on a desktop computer using only the read sequences and a transcriptome index that itself takes less than 10 minutes to build.
  - Written in C++
- Sleuth [Pimentel et al. 2017]
  - Written in R. Shiny app with plots.
  - Dependences on other R packages
  - Recommended installation: biocLite or conda
  - Issues with the latest version of R



# Ex: RNA-seq workflow using Kallisto and Sleuth [Pachter Lab]

- Sample data: GSE37704
- RNA-seq data in which lung fibroblasts in response to loss of the developmental transcription factor HOXA1 [Trapnell et al. 2013]
- 6 samples: 3 knockdown, 3 control



The Jupyter Notebook is an open-source web application that allows you to create and share documents that contain live code, equations, visualizations and explanatory text.

**Locating alignment files**

Besides the count matrix that we will use later, the `airway[2]` package also contains eight files with a small subset of the total number of reads in the experiment. The reads were selected which aligned to a small region of chromosome 1. Here, for demonstration, we chose a subset of reads because the full alignment files are large (a few gigabytes each), and because it takes between 10-30 minutes to count the fragments for each sample. We will use these files to demonstrate how a count matrix can be constructed from BAM files. Afterwards, we will load the full count matrix corresponding to all samples and all data, which is already provided in the same package, and will continue the analysis with that full matrix.

We load the data package with the example data:

```
In [8]: # loading airway example data cell package
suppressPackageStartupMessages(library("airway"))
```

The R function `system.file` can be used to find out where on your computer the files from a package have been installed. Here we ask for the full path to the `testdata` directory, where R packages store external data, that is part of the `airway[2]` package.

```
In [10]: # specifying external directory for the bioconductor package
indir <- system.file("testdata", package="airway", mustWork=TRUE)
print(list.of.files.in.the.testdata.directory.of.the.airway.package())
list.files(indir)
```

```
[1] "List of files in the /testdata directory of the airway package:"
"0052778_series_matrix.txt" "Homo_sapiens_GPC37_75_subset.gtf" "sample_table.csv" "SraRunInfo_SRR1039511.csv" "SRR1039508_subset.bam"
"SRR1039509_subset.bam" "SRR1039512_subset.bam" "SRR1039513_subset.bam" "SRR1039516_subset.bam" "SRR1039517_subset.bam"
"SRR1039520_subset.bam" "SRR1039521_subset.bam"
```

Typically, we have a table with detailed information for each of our samples that links samples to the associated FASTQ and BAM files. For your own project, you might create such a comma-separated value (CSV) file using a text editor or spreadsheet software such as Excel. We load such a CSV file with `read.csv`:

```
In [11]: # accessing the sample table from the airway package.
csvFile <- file.path(indir, "sample_table.csv")
sampleTable <- read.csv(csvFile, row.names = 1)
sampleTable
```

	SampleName	cell	dex	alut	Run	avgLength	Experiment	Sample	BioSample
SRR1039508	GSM1275862	N61211	untc	untc	SRR1039508	126	SRX384345	SRX384346	SAMN2422569
SRR1039509	GSM1275863	N61211	tt	untc	SRR1039509	126	SRX384346	SRX384347	SAMN2422570
SRR1039512	GSM1275866	N02611	untc	untc	SRR1039512	126	SRX384348	SRX384349	SAMN2422578
SRR1039513	GSM1275867	N02611	tt	untc	SRR1039513	87	SRX384350	SRX384351	SAMN2422579
SRR1039516	GSM1275870	N08011	untc	untc	SRR1039516	120	SRX384353	SRX384354	SAMN2422582
SRR1039517	GSM1275871	N08011	tt	untc	SRR1039517	126	SRX384354	SRX384355	SAMN2422583
SRR1039520	GSM1275874	N061011	untc	untc	SRR1039520	101	SRX384357	SRX384358	SAMN2422585
SRR1039521	GSM1275875	N061011	tt	untc	SRR1039521	98	SRX384358	SRX384359	SAMN2422587



# Demo



# Take home message

- Reproducible results when containers are deployed across different hardware and software configurations.
- Dockerization makes it easy to swap out a component and not worry about dependencies. Each module can be run in a different environment.
- UI to create your own containerized widgets that run your customized code.



# Future Work

- BwB
  - Scheduler
  - Support the Common Workflow Language and YAML
- Benchmarking bioinformatics workflows on the cloud and build predictive models
- Improve the scalability of workflows



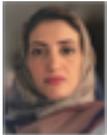
# Using Docker containers to address the reproducibility of bioinformatics analyses



GUIdock: Using Docker containers with a common graphics user interface to address the reproducibility of research. Ling-Hong Hung, Daniel Kristiyanto, Sung Bong Lee, Ka Yee Yeung. [PLOS One 2016, 11\(4\):e0152686](#).



GUIdock-VNC: using a graphical desktop sharing system to provide a browser-based interface for containerized software. Varun Mittal, Ling-Hong Hung, Jayant Keswani, Daniel Kristiyanto, Sung Bong Lee, Ka Yee Yeung. [Gigascience 2017, 6\(4\): 1-6](#).



Reproducible Bioconductor Workflows Using Browser-based Interactive Notebooks and Containers. Reem Almugbel, Ling-Hong Hung, Jiaming Hu, Abeer Almutairy, Nicole Ortogero, Yashaswi Tamta, Ka Yee Yeung. JAMIA 2018, 25(1):4-12.



Hot-starting software containers for bioinformatics analyses. Pai Zhang, Ling-Hong Hung, Wes Lloyd, Ka Yee Yeung. Gigascience 2018, 7(8): giy092.



Embedding containerized workflows inside data science notebooks enhances reproducibility. Jiaming Hu, Ling-Hong Hung, Ka Yee Yeung. [bioRxiv 309567](#).



Serverless computing provides on-demand high performance computing for biomedical research. Dimitar Kumanov, Ling-Hong Hung, Wes Lloyd, Ka Yee Yeung. [arXiv:1807.11659](#).



# UW Team



Ka Yee  
Yeung



Wes Lloyd



Ling-Hong  
Hung

Reem Almugbel  
Abeer Almutairy

Huazeng Deng  
Saranya Ravishankar Devi

Jiaming Hu  
Alyssa Ingersoll

Nicole Kauer  
Jayant Keswani

Daniel Kristiyanto  
Sung Bong Lee

Xiao Liang

Trevor Meiss  
Varun Mittal

Radhika Agumbe Sridhar  
Kuangdi Yu

Pai Zhang



Icahn School  
of Medicine at  
Mount  
Sinai

Eric Sobie  
Yuguang Xiong

# Thank you

For more information, visit  
<https://github.com/BioDepot/>

NIH grant R01GM126019



National Institute of  
General Medical Sciences

