



Persistent Reproducible Reporting

Nan Xiao

Genomic Data Scientist,
Seven Bridges



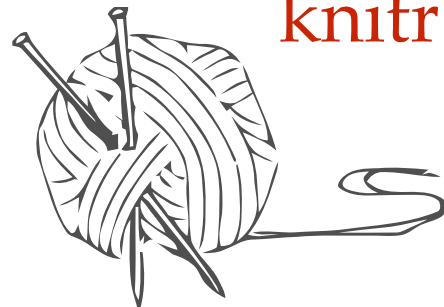
Reproducible Research



R Markdown + knitr to the rescue



+

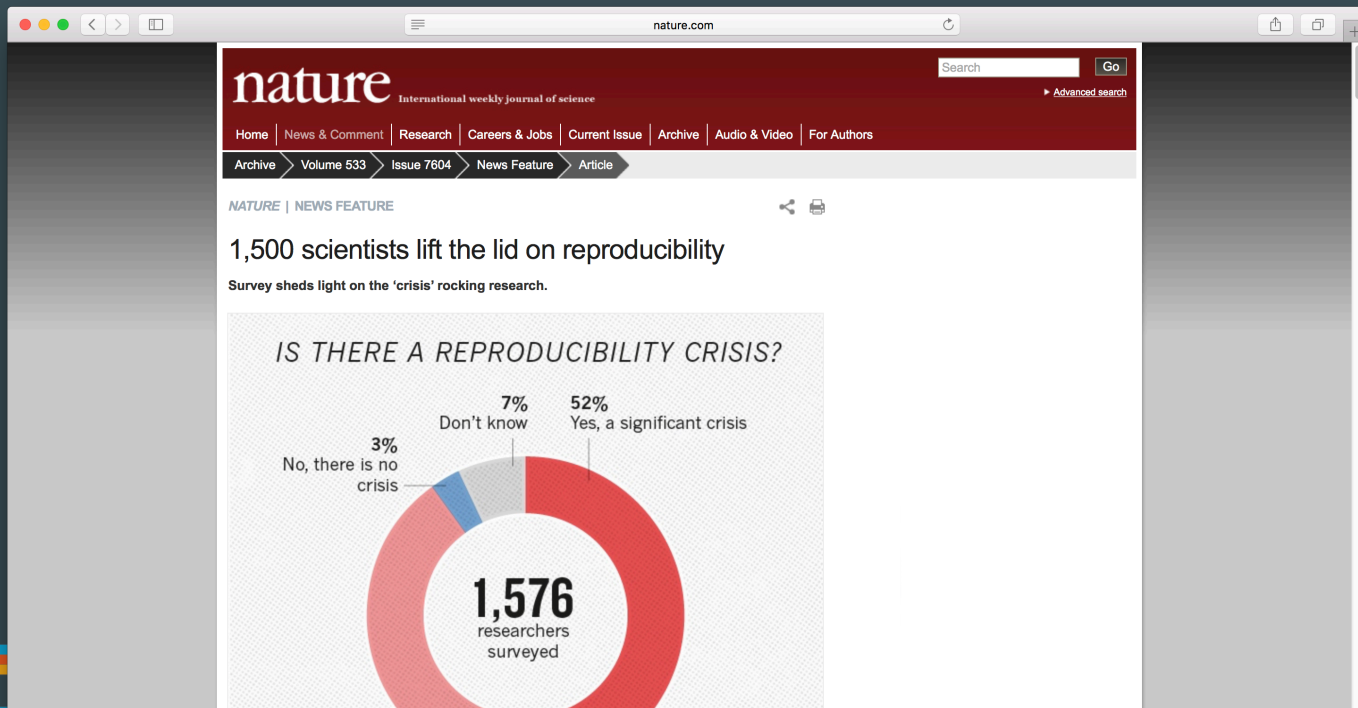


knitr



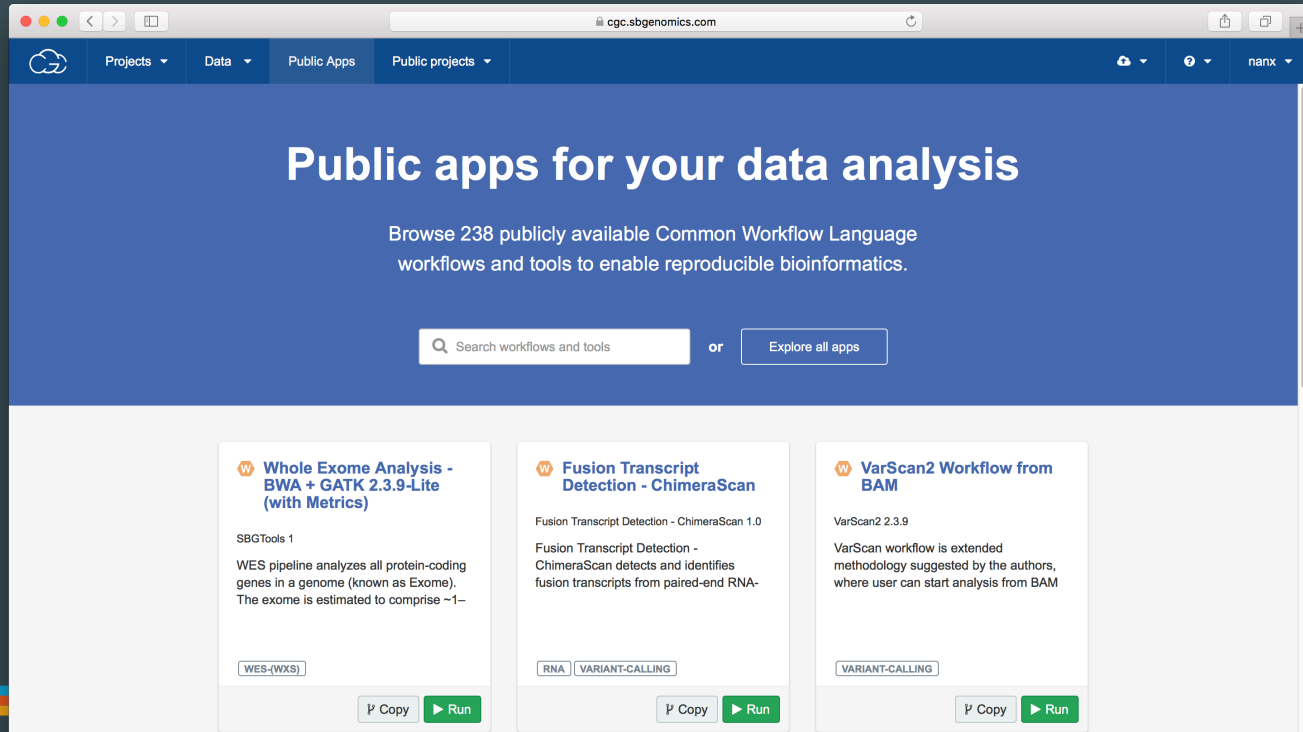
Reproducibility

... has always been a concern in both academia & industry.



Cancer Genomics Cloud

www.cancergenomicscloud.org



The screenshot shows the Cancer Genomics Cloud website. The header has a navigation bar with 'Projects', 'Data', 'Public Apps', and 'Public projects' dropdowns, and a user profile 'nanx'. The main section is titled 'Public apps for your data analysis' with a subtitle 'Browse 238 publicly available Common Workflow Language workflows and tools to enable reproducible bioinformatics.' Below this is a search bar and a button 'Explore all apps'. Three workflow cards are displayed: 'Whole Exome Analysis - BWA + GATK 2.3.9-Lite (with Metrics)', 'Fusion Transcript Detection - ChimeraScan', and 'VarScan2 Workflow from BAM'. Each card includes a description, tags, and 'Copy' and 'Run' buttons.

Public apps for your data analysis

Browse 238 publicly available Common Workflow Language workflows and tools to enable reproducible bioinformatics.

Search workflows and tools or Explore all apps

Whole Exome Analysis - BWA + GATK 2.3.9-Lite (with Metrics)

SBGTools 1

WES pipeline analyzes all protein-coding genes in a genome (known as Exome). The exome is estimated to comprise ~1–

WES-(WXS)

Copy Run

Fusion Transcript Detection - ChimeraScan

Fusion Transcript Detection - ChimeraScan 1.0

Fusion Transcript Detection - ChimeraScan detects and identifies fusion transcripts from paired-end RNA-

RNA VARIANT-CALLING

Copy Run

VarScan2 Workflow from BAM

VarScan2 2.3.9

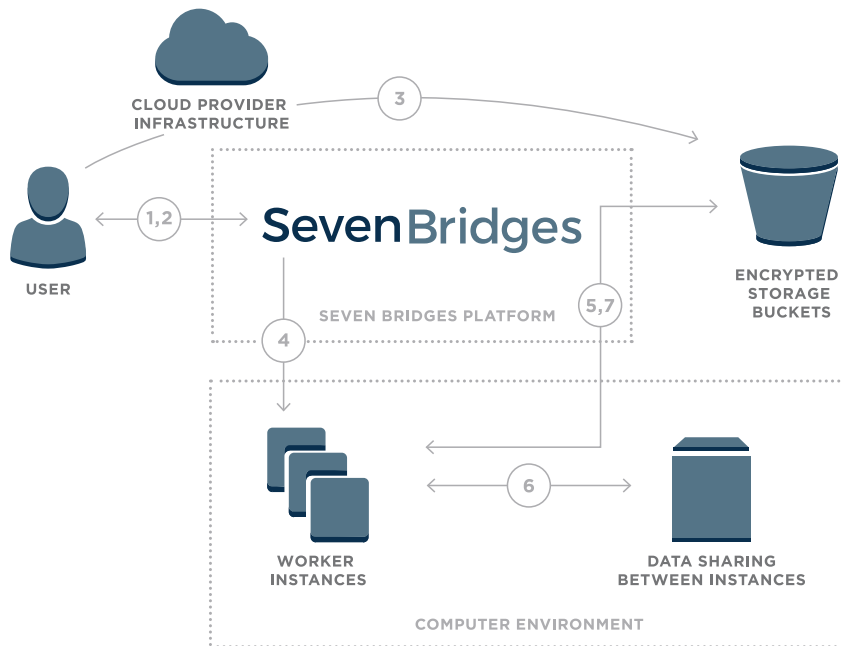
VarScan workflow is extended methodology suggested by the authors, where user can start analysis from BAM

VARIANT-CALLING

Copy Run

Hundreds of automated analysis workflows for petabyte-scale data from The Cancer Genome Atlas.

Product & Tech Innovations in CGC



COMMON
WORKFLOW
LANGUAGE

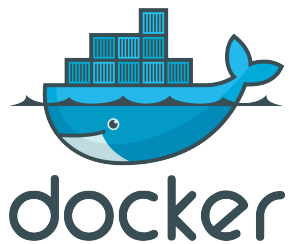


Rabix



liftr

OS-level reproducibility & persistence for reports.



+



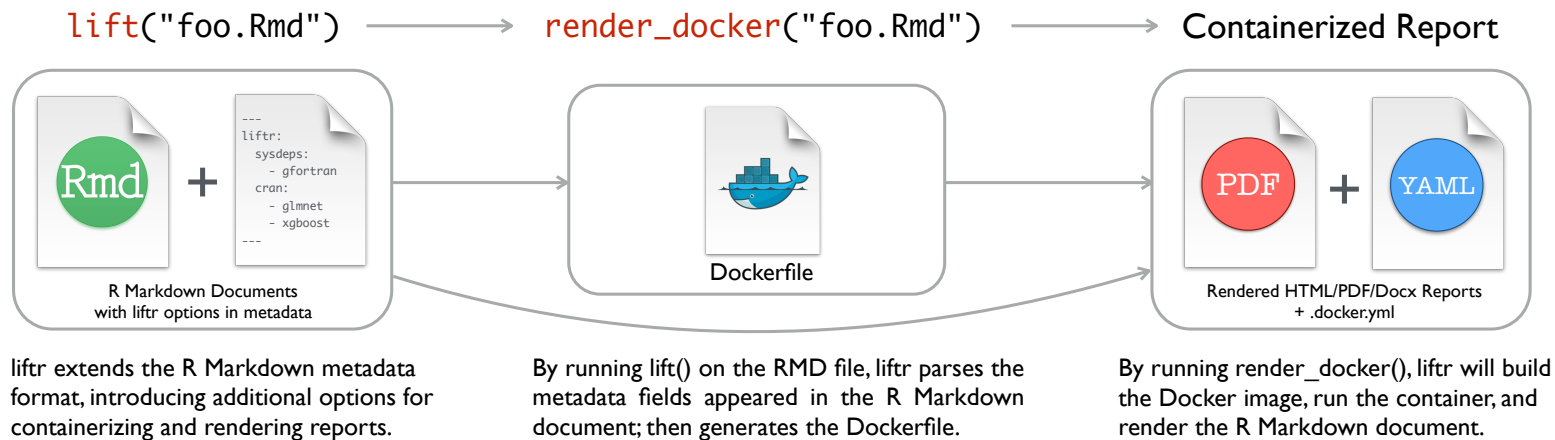
+



=



Dockerize documents as easy as 1-2-3



Dockerize documents as easy as 1-2-3

```
library("liftr")  
input = "demo.Rmd"
```

```
lift(input)           # Generate Dockerfile  
render_docker(input)  # Render report with Docker
```

```
purge_image(input)    # Clean up Docker image  
push_image(input)     # Push image to registry (devel)
```

Demo: RNA-Seq Data Analysis

Example workflow from Bioconductor

- RNA-Seq: biotechnology for measuring the expression of genes. It can help identify key genes in cancer.
- TBs of RNA-Seq data are generated. Computational tools and workflows are developed to analyze such data.
- How to ensure such reports are reproducible through time, when datasets, analysis tools are both evolving?
- Code available from: `bit.ly/liftrdemo`

Step 1

Add liftr metadata to the R Markdown document:
base image, system dependencies, package dependencies, etc.

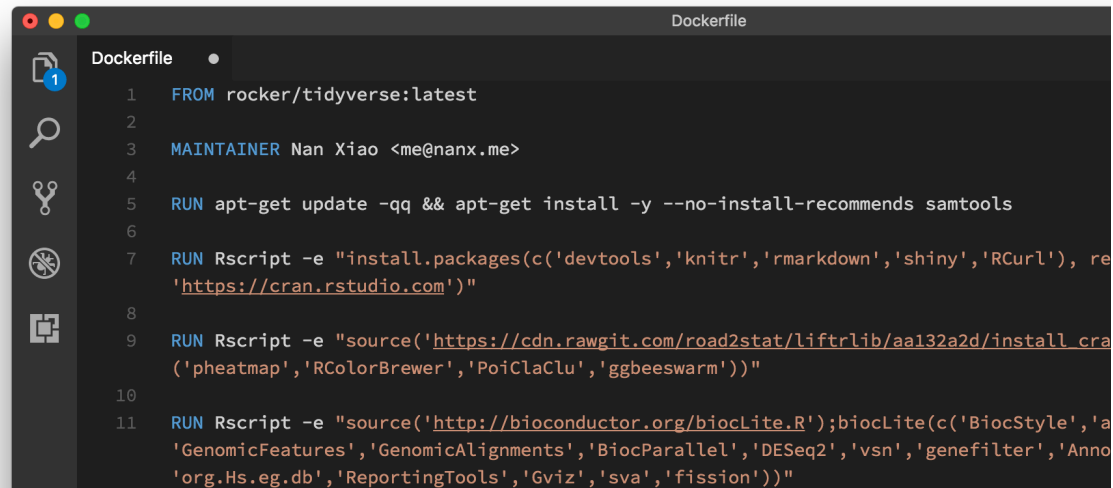
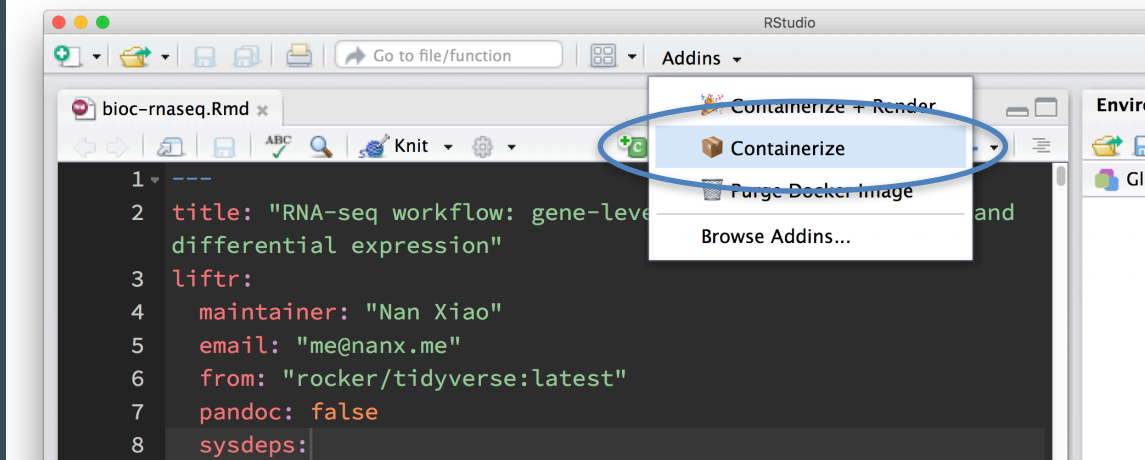
A screenshot of the RStudio application window. The title bar says 'RStudio'. The menu bar includes 'Go to file/function', 'Addins', and 'Knit'. The main editor window shows an R Markdown document titled 'bioc-rnaseq.Rmd'. The document content is as follows:

```
1 ---
2 title: "RNA-seq workflow: gene-level exploratory analysis and
3       differential expression"
4 liftr:
5   maintainer: "Nan Xiao"
6   email: "me@nanx.me"
7   from: "rocker/tidyverse:latest"
8   pandoc: false
9   sysdeps:
10     - samtools
11   cran:
12     - pheatmap
13     - RColorBrewer
14     - PoiclaClu
15     - ggbeeswarm
16   bioc:
17     - BiocStyle
18     - airway
19     - Rsamtools
20     - GenomicFeatures
21     - GenomicAlignments
```

The status bar at the bottom shows the time '44:30', the file name 'RNA-seq workflow: gene-level exploratory analysis and differential expression', and the document type 'R Markdown'. Below the editor is a 'Console' panel.

Step 2

Use `liftr::lift` to
generate Dockerfile

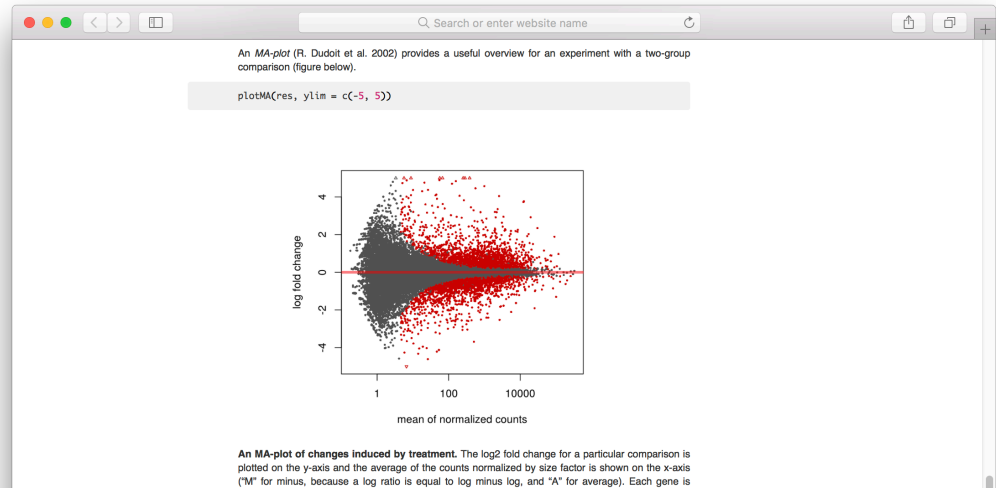
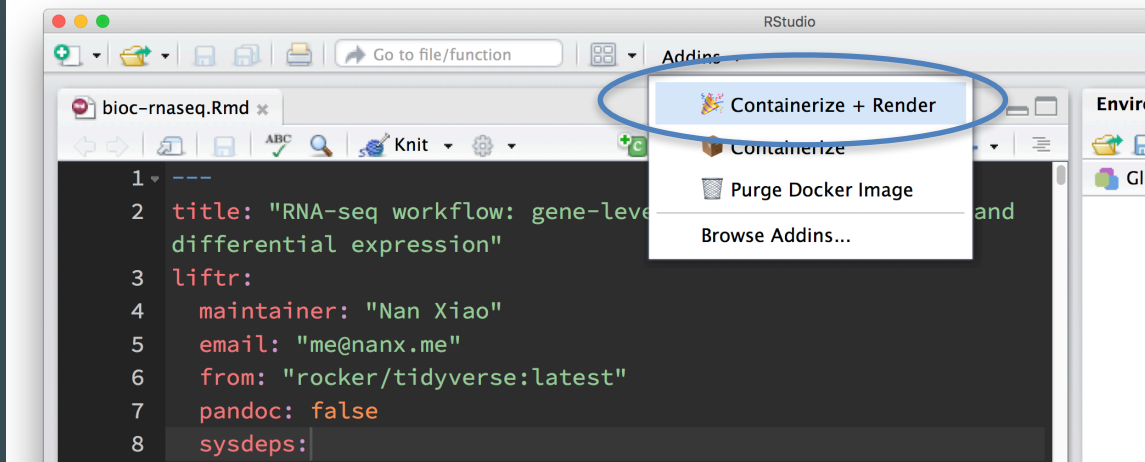


Step 3

`liftr::render_docker` will build the image, run the container, and render into PDF/HTML/Docx.

Re-compilation: cached image layers are used to improve speed.

Remove the used image, or push to registry.



Future works

- Cloud-based rendering and containerization services for dynamic documents
- Democratize reproducible report creation/sharing

Thank You!

liftr.me

@road2stat

#dockercon #liftr

