Common Workflow Language (CWL)-based software pipeline for *de novo* genome assembly from long- and short-read data

Pasi K. Korhonen

Potsdam, 5th June 2019

Introduction

Short-read draft genome assemblies (< 150 bp)

- Limits the extent of post-genomic analyses

Long-read genome assemblies (< 30 kbp)

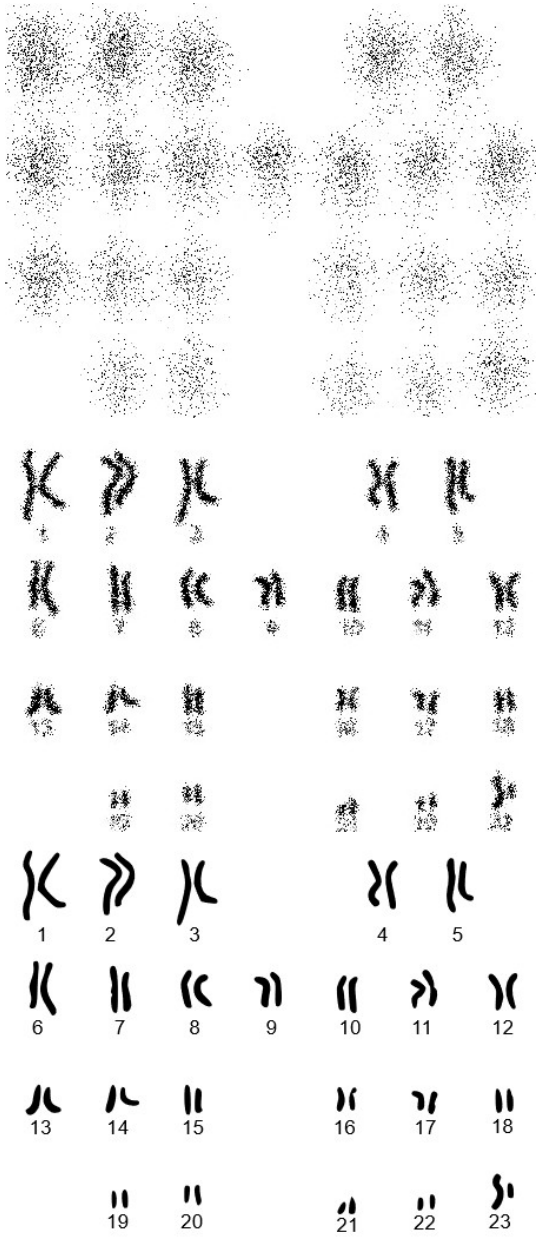
- Substantially better assemblies
- Chromosomal contiguity lacking

Scaffolding technologies

- Hi-C and BioNano
- Close to chromosomal level contiguity

AIM is to assemble complete chromosomes from data fragments called reads that represent nucleotide base pairs A/T/C/G

```
>Read1
ATTTACGTTACTTTTAAGCCCTTTGGGTTAAATGCATTTTAAGCCTTC
```



Source for images: Wikimedia Commons

Introduction

Short-read draft genome assemblies (< 150 bp)

- Limits the extent of post-genomic analyses

Long-read genome assemblies (< 30 kbp)

- Substantially better assemblies
- Chromosomal contiguity lacking

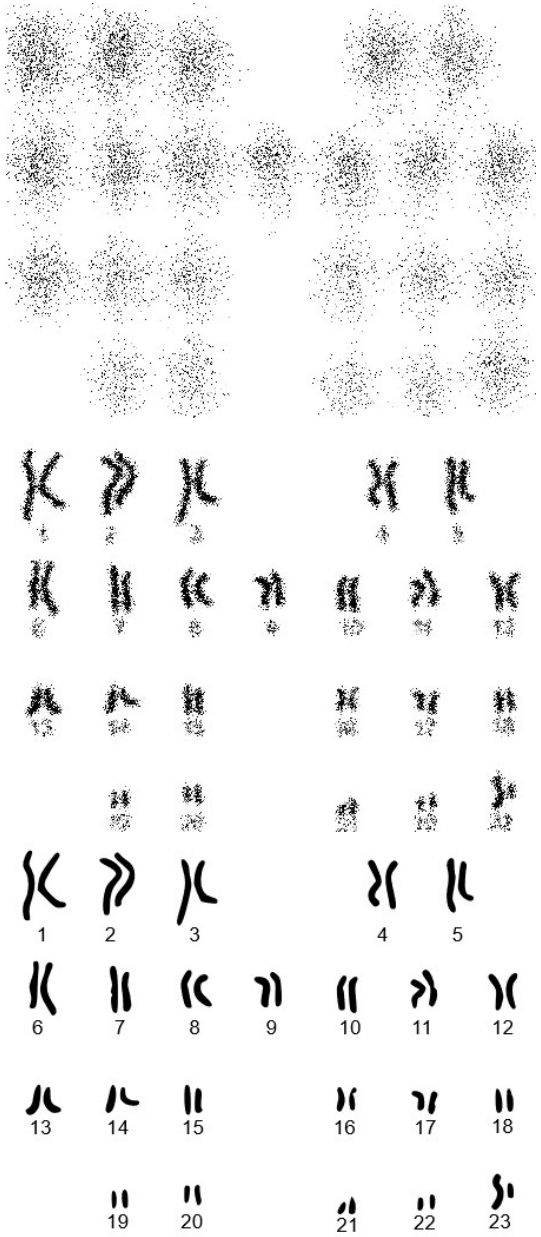
Scaffolding technologies

- Hi-C and BioNano
- Close to chromosomal level contiguity

Computational challenges in reproducibility

Assembly pipeline

Reassembled reference genomes



Source for images: Wikimedia Commons

Computational challenges in reproducibility

Dependency to environment and software dependency management

Dependency 'Hell'

- "50% of software can be successfully built or installed"

Computational environment

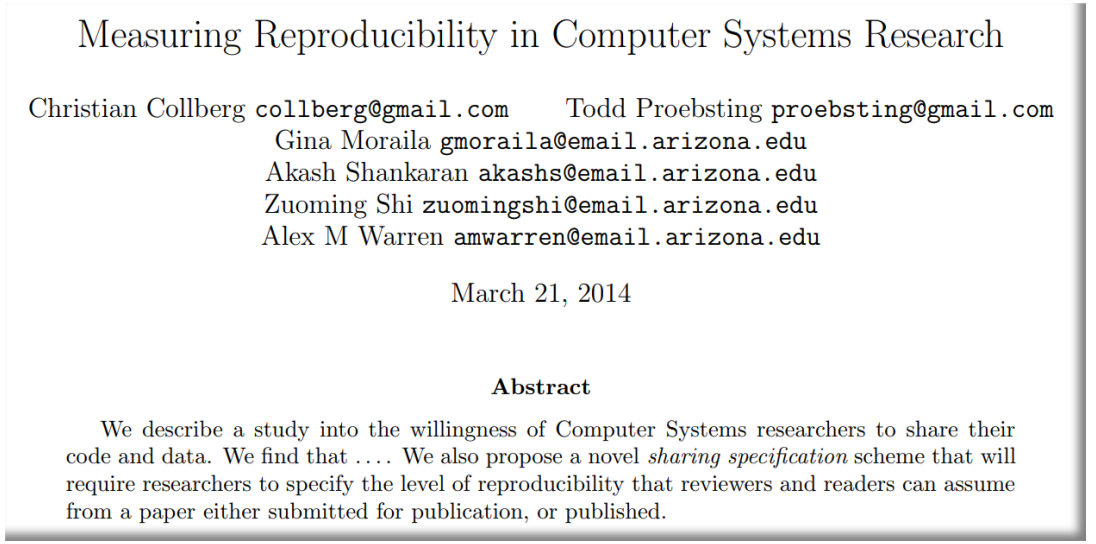
- "Installing or building SW necessary to run the code in question assumes the ability to *recreate the computational environment* of the original researchers"

Conda package management

- Resolves the dependencies among the software packages
- Easy software installation
- BioConda covers most of the software required for the assembly pipeline
- Creates a virtual environment

Docker

- Almost completely resolves the dependency to environment



Measuring Reproducibility in Computer Systems Research

Christian Collberg collberg@gmail.com Todd Proebsting proebsting@gmail.com
Gina Moraila gmoraila@email.arizona.edu
Akash Shankaran akashs@email.arizona.edu
Zuoming Shi zuomingshi@email.arizona.edu
Alex M Warren amwarren@email.arizona.edu

March 21, 2014

Abstract

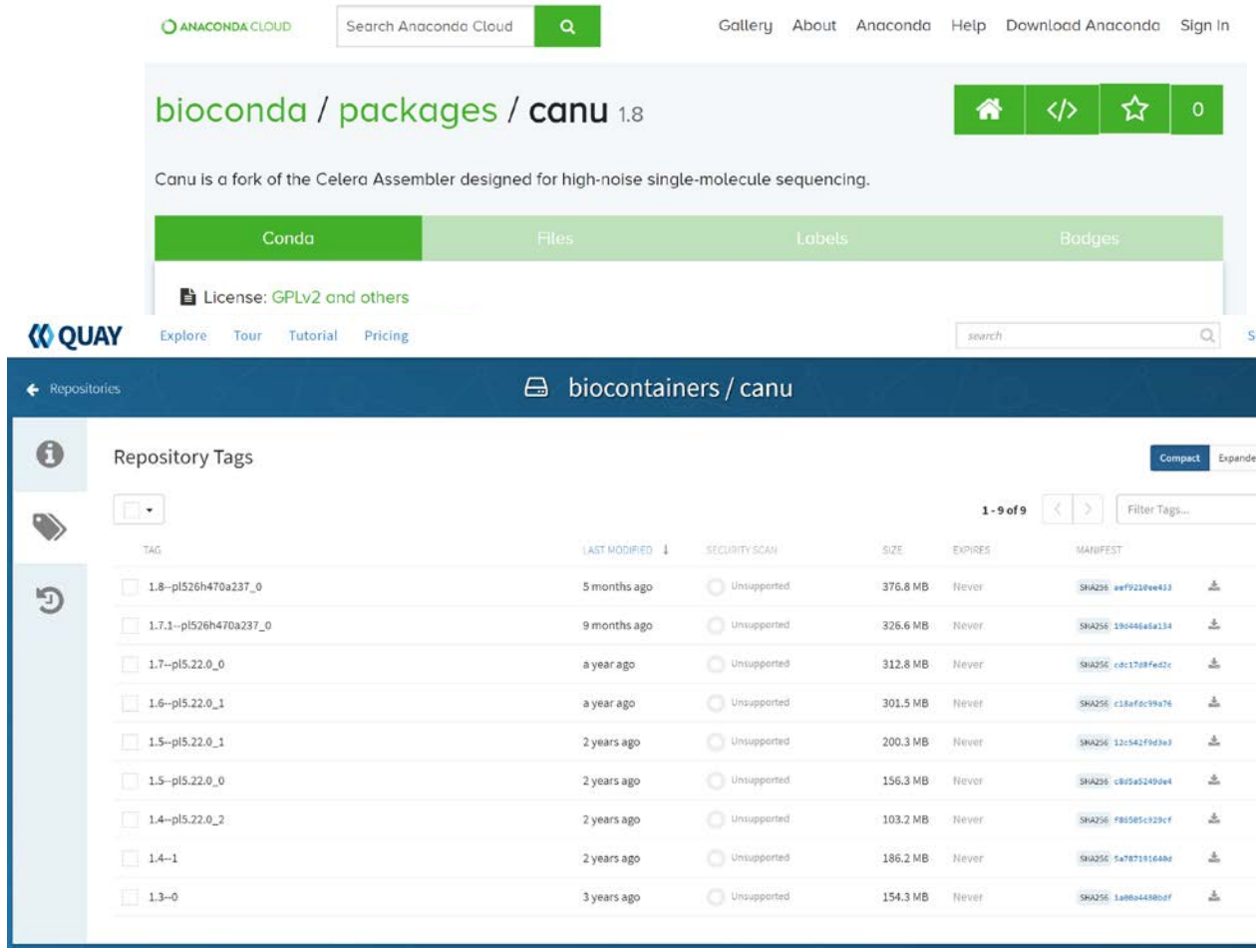
We describe a study into the willingness of Computer Systems researchers to share their code and data. We find that .... We also propose a novel *sharing specification* scheme that will require researchers to specify the level of reproducibility that reviewers and readers can assume from a paper either submitted for publication, or published.

An introduction to Docker for reproducible research

Carl Boettiger
Center for Stock Assessment Research,
110 Shaffer Rd, Santa Cruz, CA 95050, USA
cboettig(at)gmail.com

BioConda in Conda and in Quay registry

BioConda software packages



Docker images for BioConda software packages

nature methods

Correspondence | Published: 02 July 2018

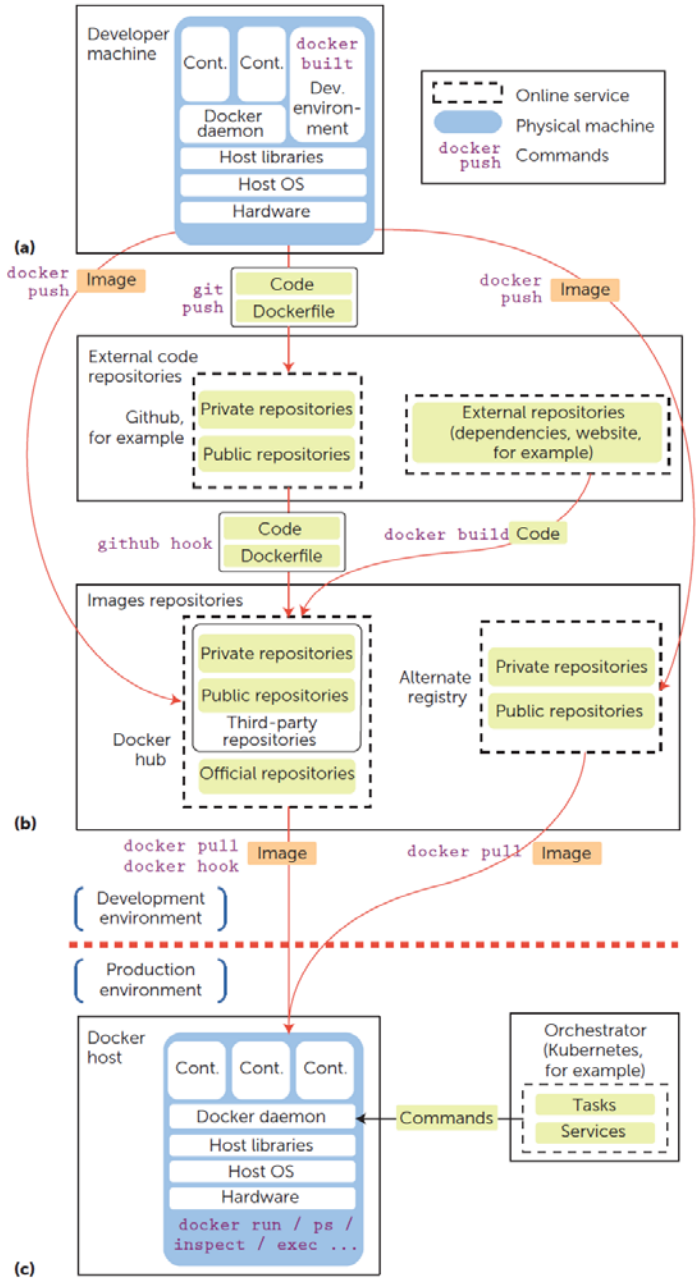
Bioconda: sustainable and comprehensive software distribution for the life sciences

Björn Grüning, Ryan Dale, Andreas Sjödin, Brad A. Chapman, Jillian Rowe, Christopher H. Tomkins-Tinch, Renan Valieris, Johannes Köster ✉ & The Bioconda Team

Nature Methods **15**, 475–476 (2018) | Download Citation ↓

Docker

- Implements a virtualised operating system
  - Containers share the Linux kernel with the host machine
- Dockerfile is used to build an image
- DockerHub/Quay can be used for distribution
- Container are executed using root rights
- **Singularity** and **udocker** execute containers in user mode



(Combe et. al, IEEE Cloud Computing, 2016)

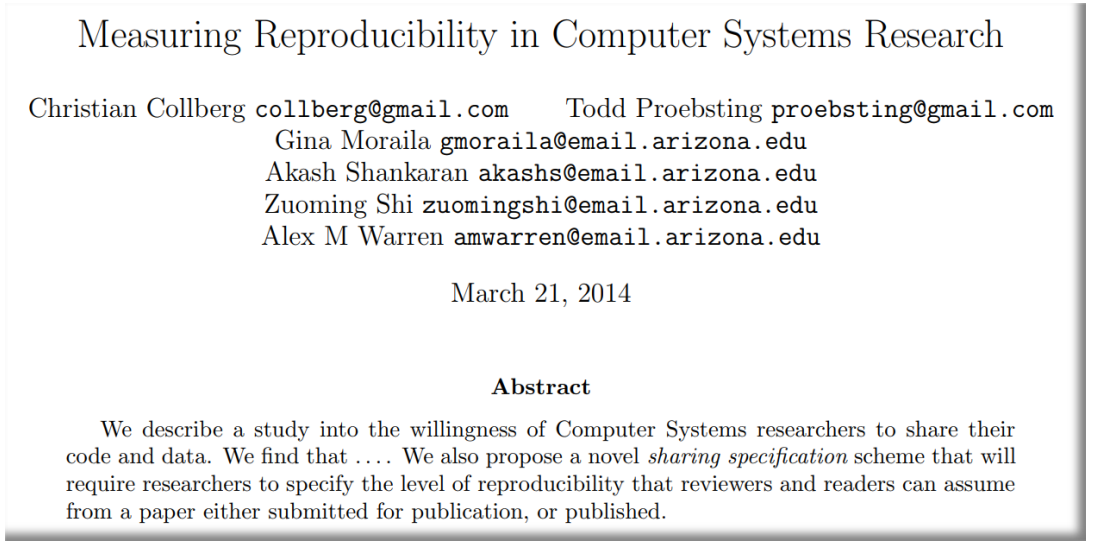
Parameter values for software

Documentation of parameter values

- “Incomplete documentation of parameters involved meant **as few as 30% of analyses** using the popular software structure could be reproduced”

Common Workflow Language (CWL)

- Parameter values have to be written into a .yml file in workflow definitions



Measuring Reproducibility in Computer Systems Research

Christian Collberg collberg@gmail.com Todd Proebsting proebsting@gmail.com
Gina Moraila gmoraila@email.arizona.edu
Akash Shankaran akashs@email.arizona.edu
Zuoming Shi zuomingshi@email.arizona.edu
Alex M Warren amwarren@email.arizona.edu

March 21, 2014

Abstract

We describe a study into the willingness of Computer Systems researchers to share their code and data. We find that .... We also propose a novel *sharing specification* scheme that will require researchers to specify the level of reproducibility that reviewers and readers can assume from a paper either submitted for publication, or published.

CWL

CWL is a specification to describe a workflow

- Command line is wrapped into a text file
- Parameters are delivered in .yml file
- Workflow definition is separated from tool wrappers
- Has multiple implementations
- Scales to different computing environments

Reference implementation

- Supports automated software installation using BioConda and Docker while workflow progresses
- Has beta support for both udocker and singularity
- Does not support parallel runs in scatter feature

Common Workflow Language, v1.0

Version 2 Fileset posted on 09.07.2016, 05:26 by Peter Amstutz, Michael R. Crusoe, Nebojša Tijanić, Brad Chapman, John Chilton, Michael Heuer, Andrey Kartashov, Dan Leehr, Hervé Ménager, Maya Nedeljkovich, Matt Scales, Stian Soiland-Reyes, Luka Stojanovic

https://www.commonwl.org

Software	Description
cwltool	Reference implementation of CWL
Arvados	Distributed computing platform for data analysis on massive data sets. Using CWL on Arvados
Toil	Toil is a workflow engine entirely written in Python.
CWL-Airflow	Package to run CWL workflows in Apache-Airflow (supported by BioWardrobe Team, CCHMC)
REANA	RE usable ANAlyses
Cromwell	Cromwell workflow engine
CWLEXEC	Apache 2.0 licensed CWL executor for IBM Spectrum LSF, supported by IBM for customers with valid contracts.

Reproducibility

Reproducibility of the results in publications

- More than 70% of researchers have **tried and failed** to reproduce another scientist's experiments, and more than half have **failed to reproduce their own** experiments

NATURE | NEWS FEATURE

1,500 scientists lift the lid on reproducibility

Survey sheds light on the 'crisis' rocking research.

Monya Baker

25 May 2016 | Corrected: 28 July 2016

CWL

- Resolves repeatability and reproducibility issues together with BioConda and Docker

GitHub

- Versioned distribution channel for software

Data distribution

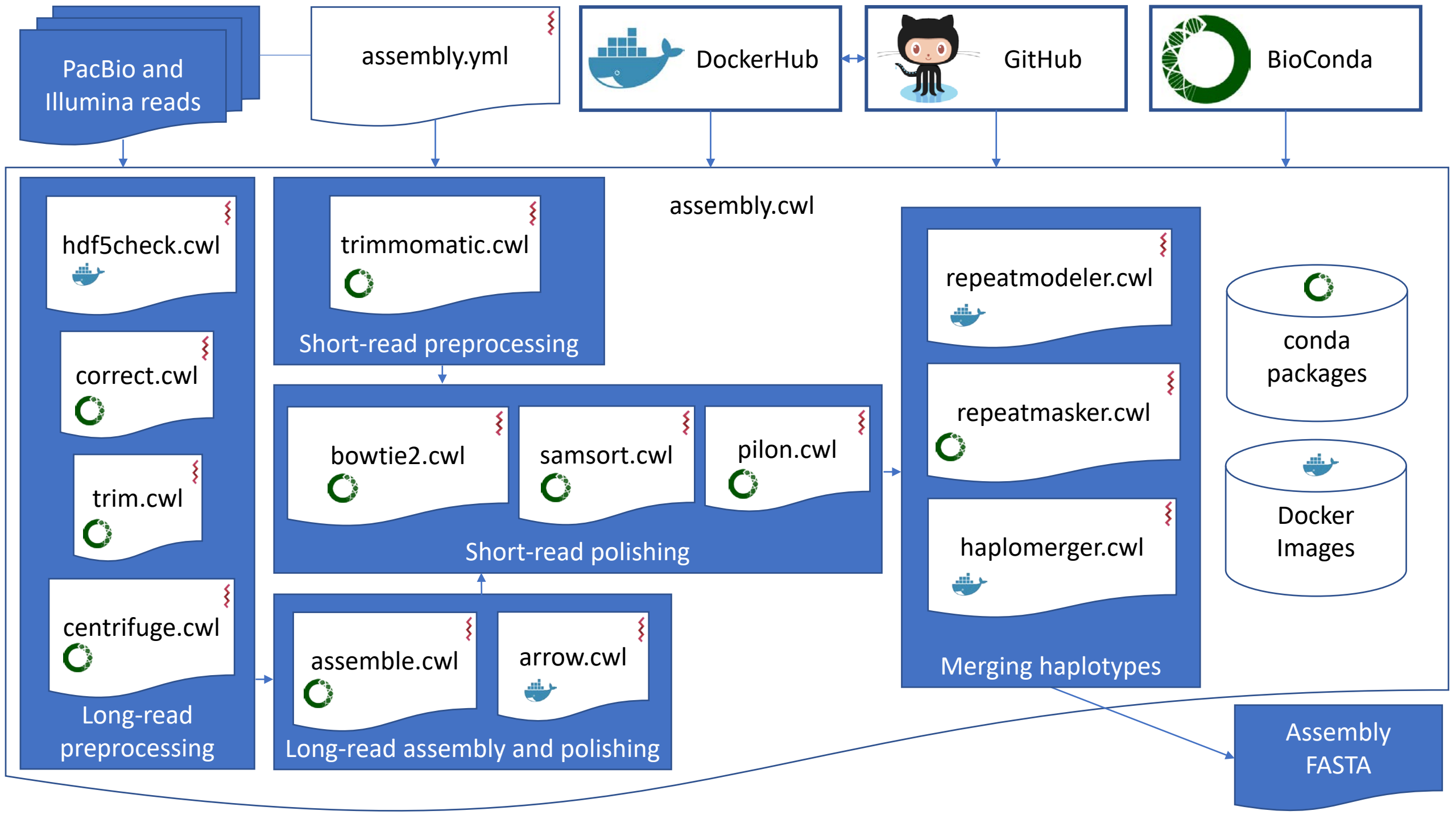
- NCBI / EBI

NCBI

EMBL-EBI

Tool	Software dependencies	Environment	Parameter values	Reproducibility	Repeatable workflow
Conda / BioConda	✓	✓	✗	✗	✓
Docker	✗	✓	✗	✓ ✗	
CWL	✗	✗	✓	✗	

Assembly pipeline


PacBio and Illumina reads
assembly.yml
DockerHub
GitHub
BioConda
assembly.cwl
hdf5check.cwl
correct.cwl
trim.cwl
centrifuge.cwl
Long-read preprocessing
trimmomatic.cwl
Short-read preprocessing
bowtie2.cwl
samsort.cwl
pilon.cwl
Short-read polishing
assemble.cwl
arrow.cwl
Long-read assembly and polishing
repeatmodeler.cwl
repeatmasker.cwl
haplomerger.cwl
Merging haplotypes
conda packages
Docker Images
Assembly FASTA

Using BioConda

- Installing **software packages** or **docker images**
  - Canu v1.6 had a dependency issue
    - Version vs. build: **v1.6 build 5**
  - Docker images elected
- Docker images run slower than direct installations from BioConda
  - Applies specifically for the program Canu
- BioConda packages do not always run correctly
  - RepeatModeler failed to predict custom repeats

nature methods

Correspondence | Published: 02 July 2018

Bioconda: sustainable and comprehensive software distribution for the life sciences

Björn Grüning, Ryan Dale, Andreas Sjödin, Brad A. Chapman, Jillian Rowe, Christopher H. Tomkins-Tinch, Renan Valieris, Johannes Köster ✉ & The Bioconda Team

Nature Methods **15**, 475–476 (2018) | Download Citation ↓

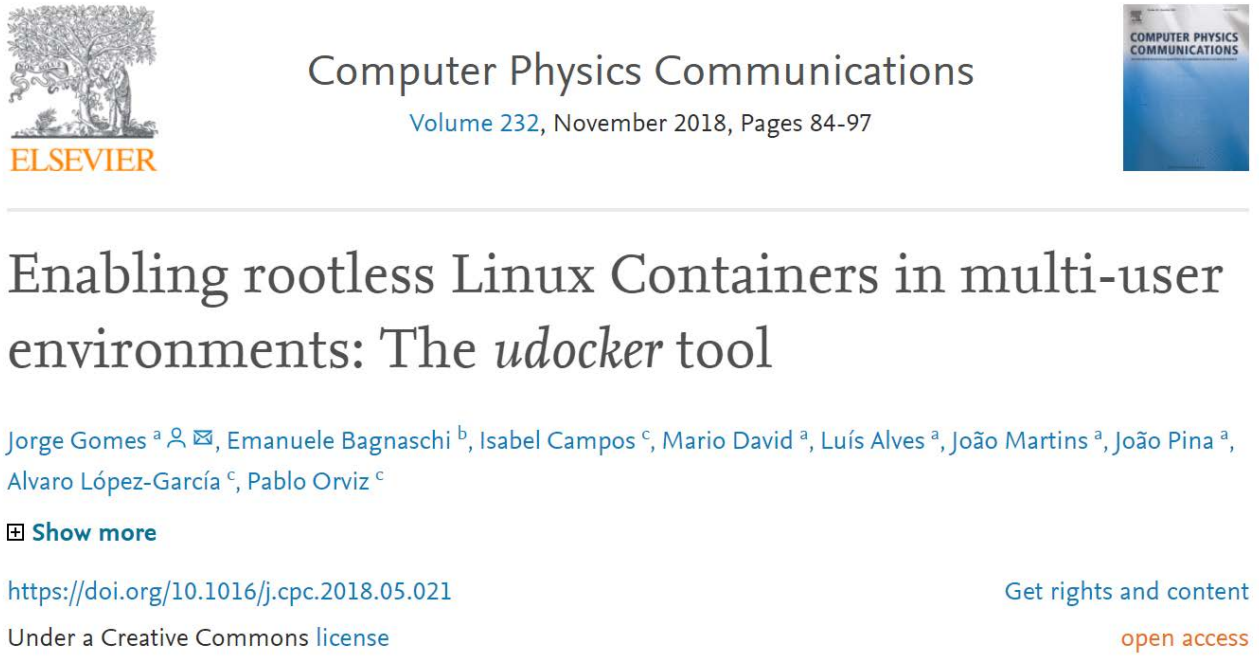
Using udocker

Pros

- Does not require root rights
- Easy to debug

Cons

- Hard to install
  - May require custom installation depending on Linux version
- Inconsistencies in behaviour in comparison to docker
  - Soft links inside the docker images may create issues



ELSEVIER

Computer Physics Communications

Volume 232, November 2018, Pages 84-97

Enabling rootless Linux Containers in multi-user environments: The *udocker* tool

Jorge Gomes [a], Emanuele Bagnaschi [b], Isabel Campos [c], Mario David [a], Luís Alves [a], João Martins [a], João Pina [a], Alvaro López-García [c], Pablo Orviz [c]

Show more

https://doi.org/10.1016/j.cpc.2018.05.021

Using CWL v1.0

Common Workflow Language, v1.0

Version 2 Fileset posted on 09.07.2016, 05:26 by Peter Amstutz, Michael R. Crusoe, Nebojša Tijanić, Brad Chapman, John Chilton, Michael Heuer, Andrey Kartashov, Dan Leehr, Hervé Ménager, Maya Nedeljkovich, Matt Scales, Stian Soiland-Reyes, Luka Stojanovic

Learning curve

No ‘if clause’ available

- One cannot create branches in workflow

CWL requires read-only access for docker but not for udocker

- May lead to discrepancies in the design of container images

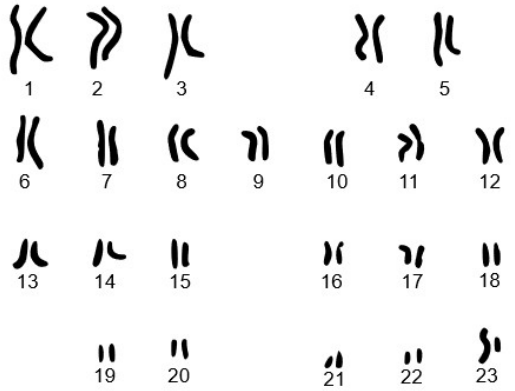
Build number cannot be defined for BioConda packages

Reassembling reference genomes

Quality of genome assembly

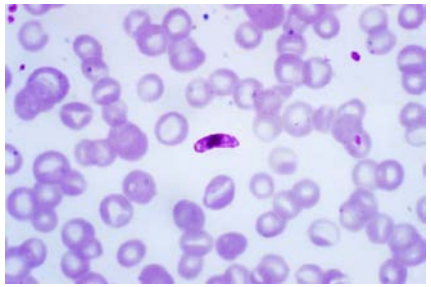
- Requirements defined by National Human Genome Research Institute (NIH) (https://www.genome.gov/10000923)
  - The **accuracy** of the assembled nucleotides is at least 99.99% (1 error in 10,000 nucleotides)
  - Decontaminated, ordered contigs (each > 30 kb) form **contiguous** chromosomes
    - Size of each gap is estimated
  - Each chromosome has at least 95% **completeness**

Human chromosome karyotype (~3 Gb)



Source for images: Wikimedia Commons

Assembly of three model organisms



P. falciparum

- 1987 in Netherlands
- Continuous in-vitro culture
- DNA extracted from haploid developmental phase (red blood cells of host)

C. elegans

- 1951 near Bristol, UK
- Propagated in cultures and distributes to multiple labs internationally

D. melanogaster

- Reference assembly from libraries in 1990, 1998, 1999

Source for images: Wikimedia Commons

Reference quality achieved?

P. falciparum

accuracy: yes

contiguity: yes

completeness: yes

	Reference	Assembly
Genome size (nt)	23,292,622	23,350,454
Sequence count	14	14
Quast genome fraction (%)	100	99.648
Quast aligned length (nt)	23,292,622	23,276,411
Number of Ns (nt)	0	0
Gap count	0	0
GC content (%)	19.34	19.33
Longest sequence (nt)	3,291,936	3,294,056
Accuracy of mismatches and indels in coding regions	100%	99.988%
Accuracy of mismatches and indels in non-coding regions	100%	99.922%

360 / 5,515 = 6.5%

C. elegans

accuracy: yes

contiguity: no

completeness: yes

	Reference	Assembly
Genome size (nt)	100,286,401	102,615,360
Sequence count	7	54
Quast genome fraction (%)	100	96.997
Quast aligned length (nt)	100,286,401	97,651,504
Number of Ns (nt)	0	0
Gap count	0	0
GC content (%)	35.44	35.44
Longest sequence (nt)	20,924,180	11,799,614
Accuracy of mismatches and indels in coding regions	100%	99.994%
Accuracy of mismatches and indels in non-coding regions	100%	99.925%

121 / 20,081 = 0.60%

D. melanogaster

accuracy: yes

contiguity: no

completeness: no

	Reference	Assembly
Genome size (nt)	137,547,960	129,695,906
Sequence count	7	61
Quast genome fraction (%)	99.644	91.514
Quast aligned length (nt)	137,057,808	126,646,721
Number of Ns (nt)	490385	0
Gap count	268	0
GC content (%)	42.08	42.17
Longest sequence (nt)	32,079,331	25,791,812
Accuracy of mismatches and indels in coding regions	100%	99.992%
Accuracy of mismatches and indels in non-coding regions	100%	99.951%

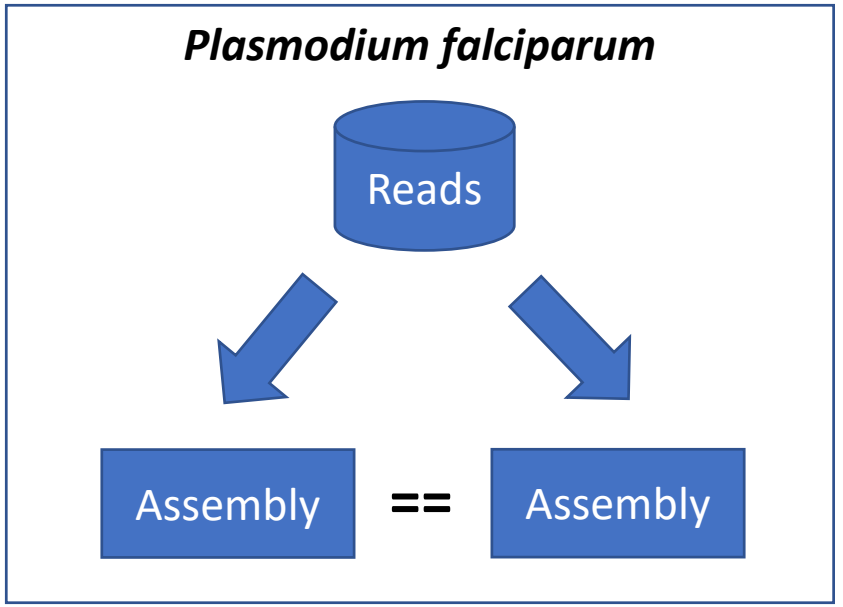
120 / 13,911 = 0.86%

mutated proteins

Conclusions

CWL together with the programs conda and docker can

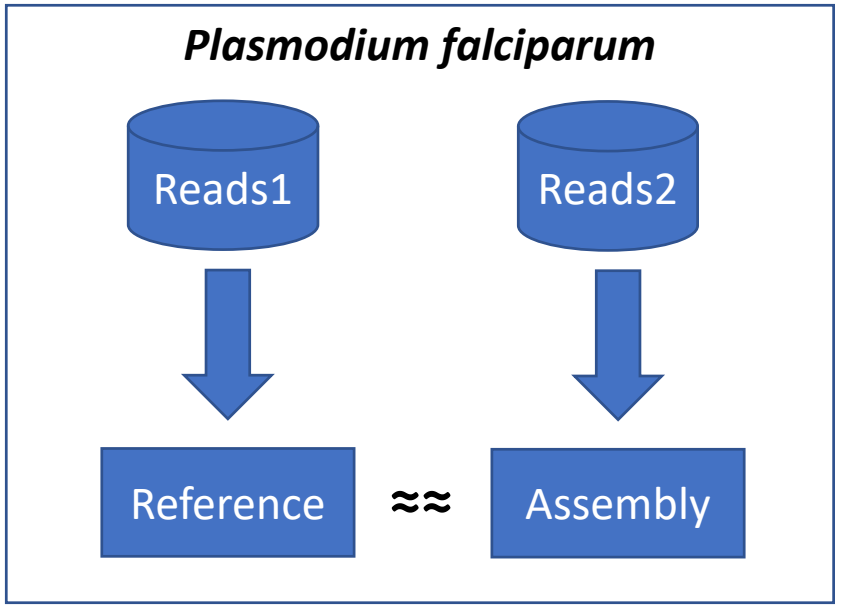
- **create a repeatable pipeline**
- reproduce the results
- create a reusable pipeline



Conclusions

CWL together with the programs conda and docker can

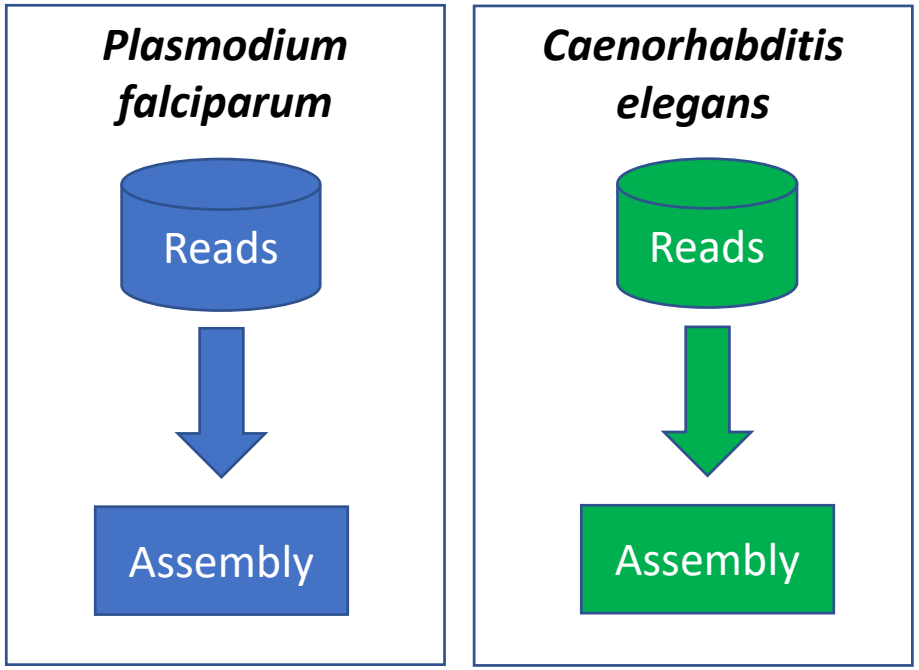
- create a repeatable pipeline
- **reproduce the results**
- create a reusable pipeline



Conclusions

CWL together with the programs conda and docker can

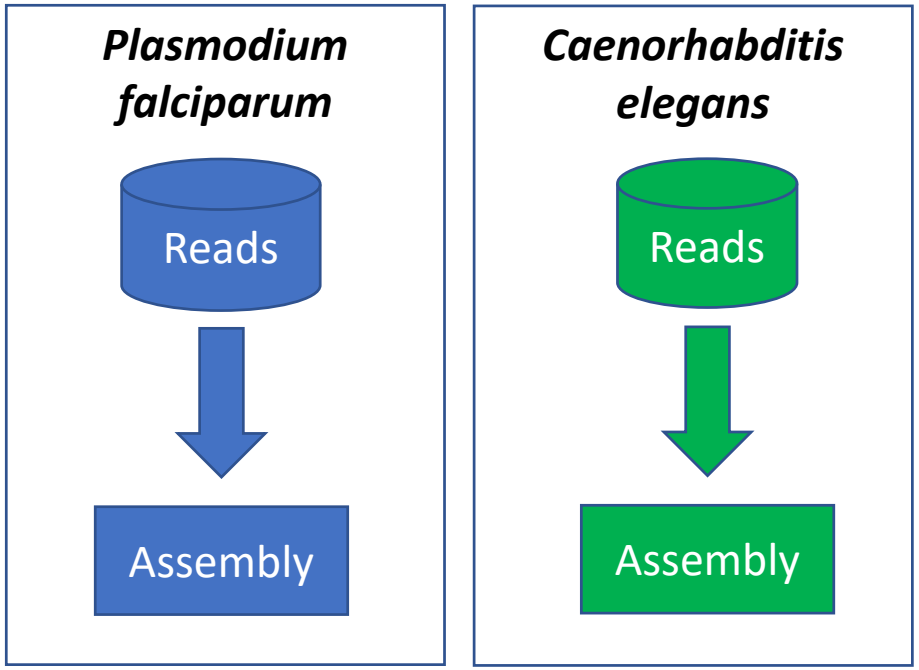
- create a repeatable pipeline
- reproduce the results
- **create a reusable pipeline**



Conclusions

CWL together with the programs conda and docker can

- create a repeatable pipeline
- reproduce the results
- create a reusable pipeline



The resulting assemblies are close to reference quality

Next steps

- Support Hi-C and BioNano scaffolding technologies
- Support Nanopore long reads
- Integrate workflow to HPC cluster
- Replace udocker with Singularity
- Use CWL to automate genome annotation

GigaScience, 0, 2019, 1–0

doi: 10.1093/gigascience/giz014
Advance Access Publication Date: 0 2019
Technical Note

TECHNICAL NOTE

Common workflow language (CWL)-based software pipeline for *de novo* genome assembly from long- and short-read data

Pasi K. Korhonen*, Ross S. Hall, Neil D. Young and Robin B. Gasser*

Department of Veterinary Biosciences, Melbourne Veterinary School, The University of Melbourne, Parkville, Victoria 3010, Australia

https://www.researchgate.net/publication/331459007_Common_Workflow_Language_CWL-based_software_pipeline_for_de_novo_genome_assembly_from_long-_and_short-read_data

https://github.com/vetscience/Assemblosis

Acknowledgements