Welcome!

Translating workflows into Nextflow with Janis

Bioinformatics Workflows

Chain multiple tasks to perform an analysis.

Has ***inputs***, produces ***outputs***.

Can be written in essentially any language.

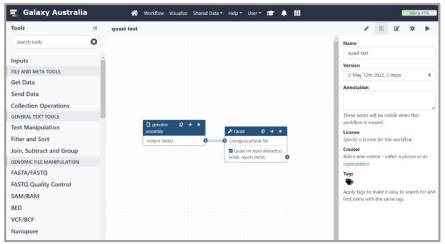
Choosing a Workflow Language

Each language has different *users & use-cases*

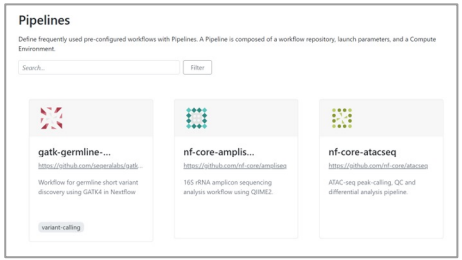
- Standard in field
- Properties / features
- Compute environments

Workflow languages are an *investment*

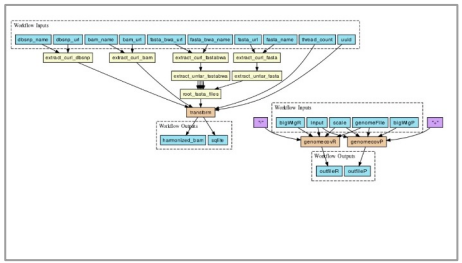
- Existing pool of ***knowledge, talent*** (researchers / staff), local legacy ***workflows***
- Ongoing support & active community?
- Learning new language takes weeks / months



Galaxy Workflow Editor



Nextflow Tower

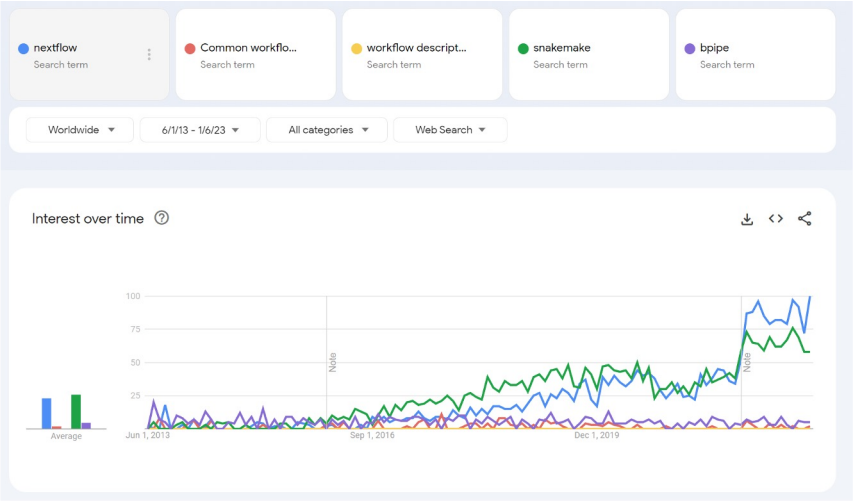


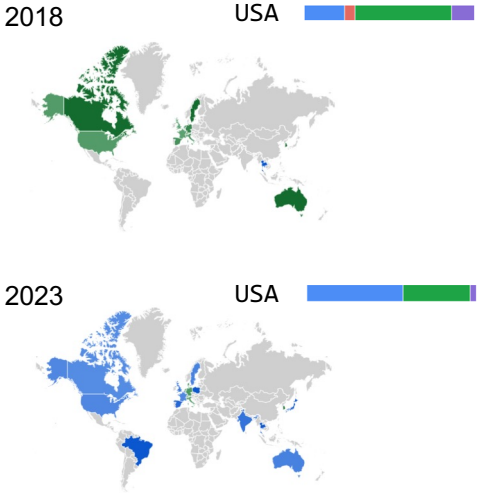
CWL viewer

Choosing a Workflow Language

Workflow languages come and go over time.

- Can we be sure our investment will still be usable in 5 years?





Choosing a Workflow Language

Workflow space *highly divided*

No ‘standard’ workflow language

- New specs continually created
- Old specs become stale
- Can’t know which spec will ‘win’

Might not even be the correct question

- There are multiple programming languages
- Which will ‘win’?

Existing Workflow systems

Michael R. Crusoe edited this page 7 days ago · 320 revisions

Permalink: https://s.apache.org/existing-workflow-systems

Cite as (update dates):

Peter Amstutz, Maxim Mikheev, Michael R. Crusoe, Nebojša Tijanić, Samuel Lampa, et al. (2022): **Existing Workflow systems.** *Common Workflow Language wiki*, GitHub. https://s.apache.org/existing-workflow-systems updated 2022-06-20, accessed 2022-06-20.

Computational Data Analysis Workflow Systems

An incomplete list

Please add new entries at the bottom.

In addition to this list, actively developed free/open-source systems should be registered at https://workflows.community/systems

See also: https://github.com/pditommaso/awesome-pipeline

1. Arvados - CWL-based distributed computing platform for data analysis on massive data sets. https://arvados.org/ https://github.com/arvados/arvados
2. Apache Taverna http://www.taverna.org.uk/ https://taverna.incubator.apache.org/
3. Galaxy http://galaxyproject.org/
4. SHIWA https://www.shiwa-workflow.eu/

...

312. SimTool/Sim2Ls: Jupyter notebook-based pipelines of Simulation Tools for the HUBzero platform lead by nanoHUB https://github.com/hubzero/simtool https://simtool.readthedocs.io/ https://doi.org/10.1371/journal.pone.0264492
313. SideIO: A Side I/O system framework for hybrid scientific workflow _(no project/source code available)_ https://doi.org/10.1016/j.jpdc.2016.07.001
314. Flyte https://flyte.org/
315. StreamFlow https://streamflow.di.unito.it/
316. Jupyter Workflow https://jupyter-workflow.di.unito.it/
317. Nnodes: a simple workflow manager for Python functions and command line tools https://github.com/icui/nnodes

Janis

Portable Pipelines Project

Janis was developed as part of the Portable Pipelines Project, a collaboration between:

- Peter MacCallum Cancer Centre
- Melbourne Bioinformatics (University of Melbourne)
- Walter and Eliza Hall Institute of Medical Research (WEHI)
- Australian BioCommons

Initial aim was to develop shared cancer bioinformatics pipelines across Parkville biomedical institutes.

But the question back then in 2018 was; **which workflow language should they be written in?** [CWL/ WDL/ snakemake/ nextflow]

Australian BioCommons

Peter Mac
Peter MacCallum Cancer Centre
Victoria Australia

WEHI
brighter together

Melbourne Bioinformatics
BIOINFORMATICS + DATA SERVICES + INFRASTRUCTURE, FOR LIFE SCIENCES TODAY

Janis

Instead of committing to a single language...

Built a framework which can ***transpile*** to CWL or WDL.

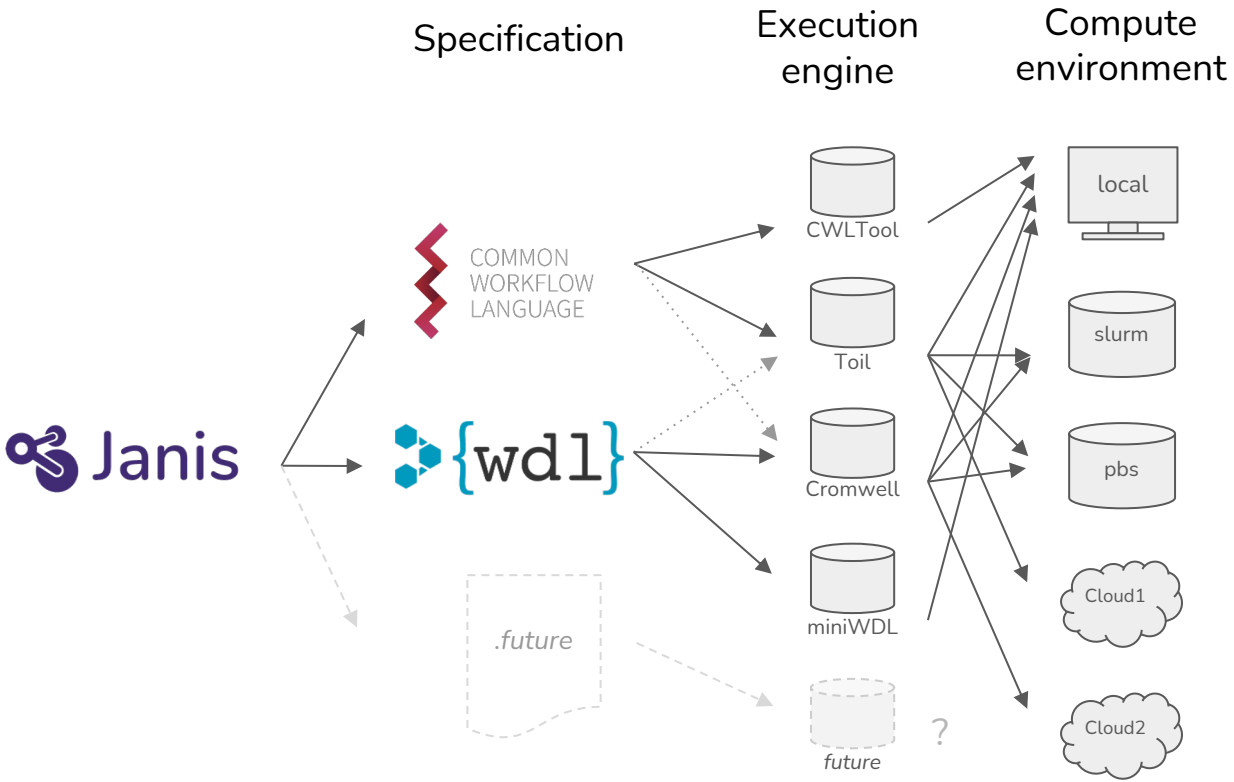
Wanted ***type-safety*** due to importance of file-types in bioinformatics

Would need to allow ***flexible execution*** across different compute environments of the respective stakeholders

If a new workflow system became popular in ***future***, could add ***transpilation*** support for the format

```python
class BcfToolsAnnotateBase(BcfToolsToolBase, ABC):

    def tool(self):
        return "bcftoolsAnnotate"

    def base_command(self):
        return ["bcftools", "annotate"]

    def inputs(self):
        return [
            ToolInput("vcf", Vcf(), position=10),
            ToolInput(
                "outputFilename",
                Filename(extension=".vcf"),
                prefix="--output",
                doc="[-o] see Common Options",
            )
        ]

    def outputs(self):
        return [ToolOutput("out", Vcf, glob=InputSelector("outputFilename"))]
```

First released in June 2019



- partial support

Janis in 2021

Fast-forward to 2021.

Had built and run more than a dozen pipelines for real-world cancer analysis @ PeterMac.

Nextflow was booming. Name of the game had changed.

CWL, Snakemake and Nextflow all developed systems for flexible execution in different environments.

Reflected on our product & realised that going forward, ***translation*** was the unique value of Janis.

→ Started work on janis-translate

Janis in 2021

Fast-forward to 2021.

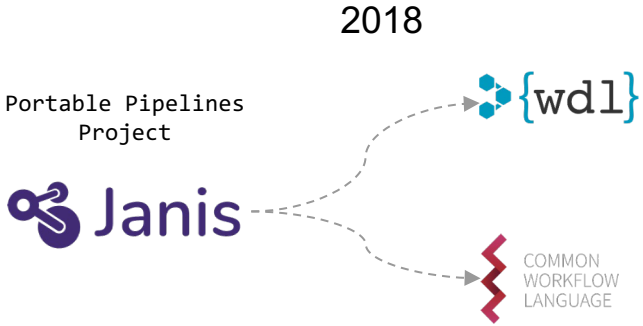
Had built and run more than a dozen pipelines for real-world cancer analysis @ PeterMac.

Nextflow was booming. Name of the game had changed.

CWL, Snakemake and Nextflow all developed systems for flexible execution in different environments.

Reflected on our product & realised that going forward, ***translation*** was the unique value of Janis.

→ Started work on janis-translate



Janis in 2021

Fast-forward to 2021.

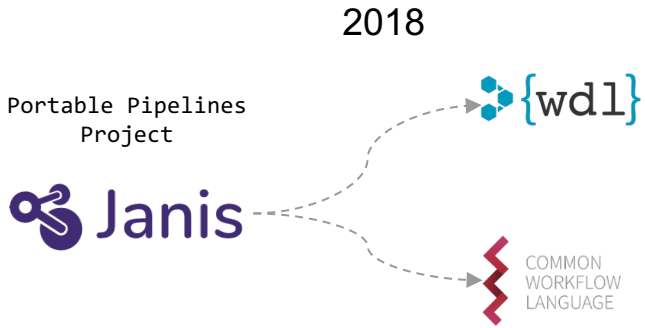
Had built and run more than a dozen pipelines for real-world cancer analysis @ PeterMac.

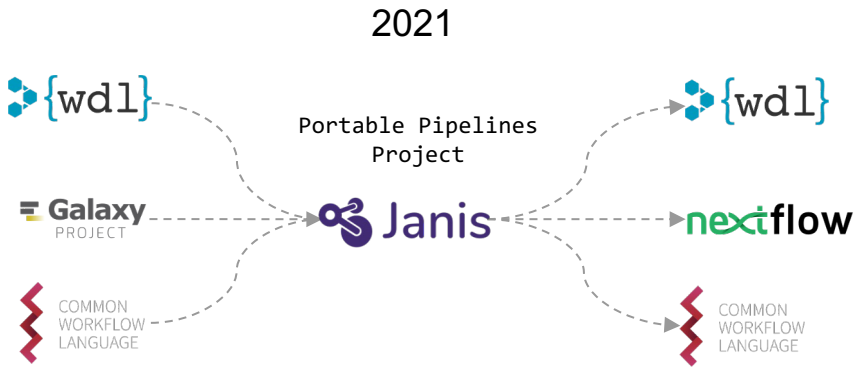
Nextflow was booming. Name of the game had changed.

CWL, Snakemake and Nextflow all developed systems for flexible execution in different environments.

Reflected on our product & realised that going forward, ***translation*** was the unique value of Janis.

→ Started work on janis-translate



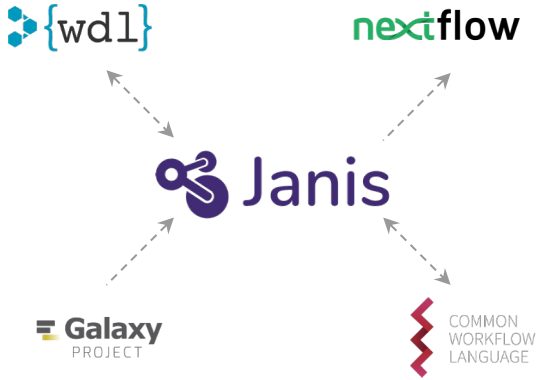


Janis Translate

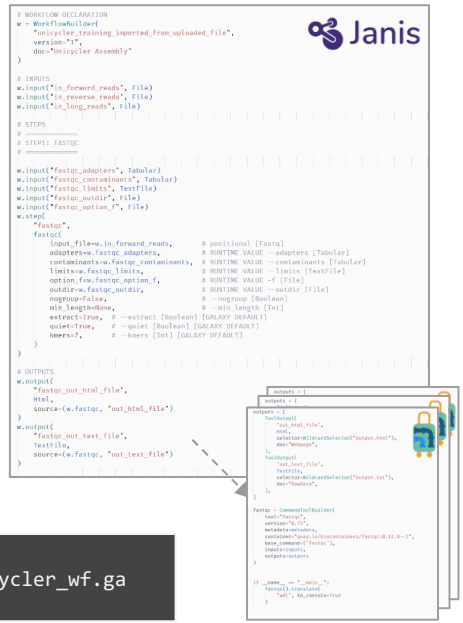
This is our best attempt at realising our goal.

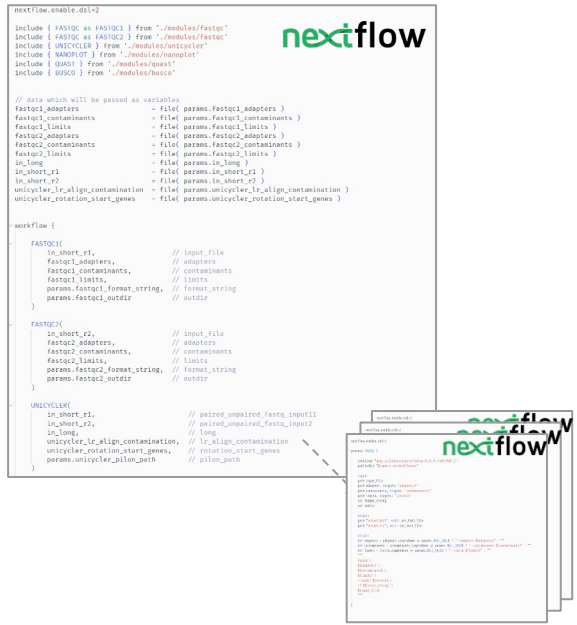
Janis-translate helps you migrate workflows / tools from one specification to another.

- Productivity tool to cut down boilerplate.
- Aims for human-readable translations.



Janis Translate



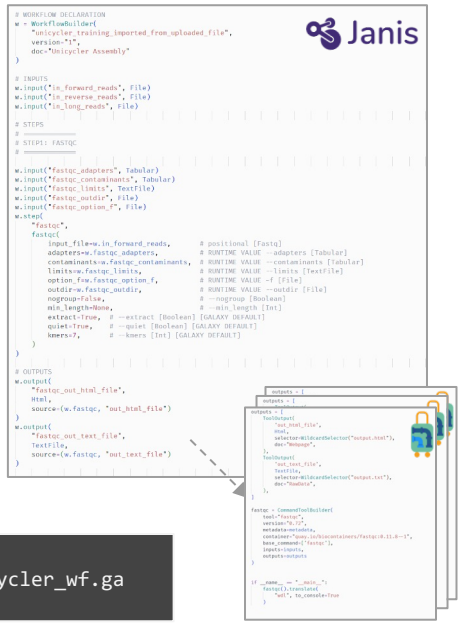


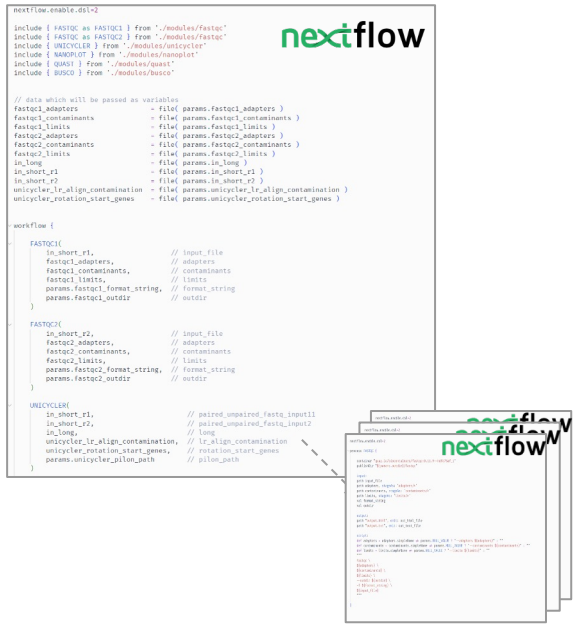

```
janis translate --from galaxy --to nextflow ./unicycler_wf.ga
```

Janis Tools (translated)

Nextflow Processes (translated)

Janis Translate






```
janis translate --from galaxy --to nextflow ./unicycler_wf.ga
```

Janis Tools (translated)

Nextflow Processes (translated)

Turns out, perfect translations are really hard!

Manual adjustments needed *(often).*

Today's Session

Hands-on experience with workflow migration using Janis Translate.

- CWL → Nextflow
- Galaxy → Nextflow

Aim is to:

- Get familiar with Janis translation output
- Run the translations with test data
- Do some proper Nextflow debugging to bring translations to a finished, runnable state

(we are here) →

Description	Est. Duration	Real time (finish)
Housekeeping	5 min	1:10pm
Intro to the session & Janis	10 min	1:20pm
CWL → NEXTFLOW TRANSLATIONS		
Intro to CWL	10 min	1:30pm
Samtools Flagstat Tool	20 min	1:50pm
GATK HaplotypeCaller Tool	20 min	2:10pm
Align Sort Markdup Workflow	20 min	2:30pm
Break	10 min	2:40pm
GALAXY → NEXTFLOW TRANSLATIONS		
Intro to Galaxy	10 min	2:50pm
Samtools Flagstat Tool	10 min	3:00pm
Limma Voom Tool	20 min	3:20pm
RNA-Seq reads to counts workflow	30 min	3:50pm
Wrap up	10 min	4:00pm
EXTENSION		
Participant workflow translations	30 min	4:30pm

Acknowledgements

Current Janis team:

- Grace Hall (Janis)
- Richard Lupat (Peter Mac)
- Daniel Park (UoM)
- Bernie Pope (UoM)
- Lisa Phippard (BioCommons)
- Evan Thomas (WEHI)

Previous team members of Portable Pipelines Project:

- *Michael Franklin (Janis)*
- *Juny Kesumadewi (Janis)*
- *Jiaan Yu (Peter Mac)*
- *Xinzhe Li (Peter Mac)*

Initial proof-of-concept

- Mohammad Bhuyan (WEHI)

Early adopters / community support

- Sebastian Hollizeck (Peter Mac)
- Tom Conway (Peter Mac)
- Kersten Bruer (DKFZ)
- Michael Crusoe (CWL)

Workshop Setup:

- Matthew Downton (NCI)
- Melissa Burke (BioCommons)
- Alex Ip (AARNET)
- Georgie Samaha (SIH)
- Audrey Stott (Pawsey)