Getting started with Conda and Nextflow

Week 4 - Workflow Management Training Workshop

Robert A. Petit III, PhD

Wyoming Public Health LABORATORY

Theiagen Genomics

Note! Switching things up this week

Conda and Nextflow are expansive topics

Mixing lecture with hands-on exercises

Please interrupt and ask questions

BIOCONDA®

CONDA

nextflow

nf-core

Topics for Today

Tentative Schedule

- Conda and Installation (~20 minutes)
- Nextflow and Example Runs (~30 minutes)
- Break (~10 minutes)
- Nextflow Channels & Processes (~20 minutes)
- Nextflow DSL1 vs DSL2 (~20 minutes)
- Additional Exercises and Discussion
- Wrap-up (Final ~15 minutes)

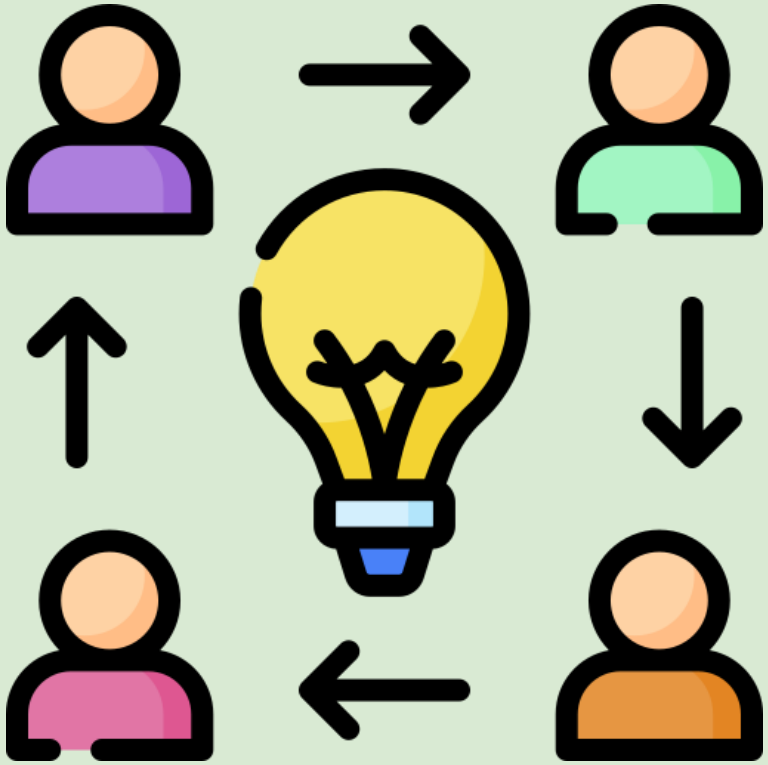
CONDA

- A package manager for installing 1000s of tools
- Available on Windows, Mac, and Linux
  - Windows has fewer available packages
- Packages are grouped into channels
  - conda-forge – 18k+ general, non-domain specific tools
  - Bioconda – 4k+ bioinformatic tools and libraries
  - Can even set up a personal channel

```
conda create -n fastq-scan fastq-scan
conda activate fastq-scan
fastq-scan -h
Usage: cat FASTQ | fastq-scan [options]
Version: 1.0.0

Optional arguments:
    -g INT   Genome size for calculating estimated
                 sequencing coverage. (Default 1)
    -p INT   ASCII offset for input quality scores,
                 can be 33 or 64. (Default 33)
    -q       Print only the QC stats, do not print
                read lengths or per-base quality scores
    -v       Print version information and exit
    -h       Show this message and exit
```

BIOCONDA® for all your bioinformatic tools



- Makes bioinformatics accessible
  - Easy installs, dependency handling
- Downstream containerization*
  - Docker – Biocontainers
  - Singularity Images - Galaxy Project
- Truly a community driven repository
  - More than 1,300 people have contributed
- Currently 4,000+ recipes are available
- Learn more at bioconda.github.io

Exercise 1: Conda

Exercise 1: Install Conda and Serotype a Shigella genome

- Head on over to GitHub: Exercise 1 - Conda

- Together we will:
  - Install Miniconda3
  - Install Mamba
  - Create a “Shigatyper” environment
  - Serotype and Shigella genome

- Wrap up the exercise

When to use and not use CONDA

- Use Conda:
  - Rapid prototyping
  - Need a quick answers
  - Personal systems

- Skip Conda:
  - Production should use containers (Docker or Singularity)
  - HPC and Cloud environments
  - Using Windows only (*e.g., not using WSL2*)

Conda “best practices”

- Keep the initial “base” environment clean
  - Mamba is an exception
- Use “conda create” for isolated environments
  - Treat these environments as disposable
- Channel priority is important
  - Prefer conda-forge before bioconda

Common Conda issues to keep in mind

- “Unable to solve environment”

- Automated docker builds can be problematic
  - If a StaPH-B Docker container is available, use it
    - Rigorous manual builds that are verified to be working
  - Curtis Kapsak is an expert for using Docker for bioinformatics

- Conda is can be fragile if misused
  - Fragility increases as “base” environment grows
  - If all else fails, just reinstall it

Questions or comments?

Introduction to Nextflow

nextflow

```
nextflow.enable.dsl=2


process sayHello {
  input:
    val cheers
  output:
    stdout

  """
  echo $cheers
  """
}

workflow {
  channel.of('Ciao','Hello','Hola') | sayHello | view
}
```

- A popular workflow manager in Bioinformatics
- Enables scalable and reproducible pipelines, with built in resumability
- Supports Conda, Docker, and Singularity
- Seamlessly move between local resources, HPC, and major cloud providers
- Regularly solicits user feedback to guide future developments

nextflow is platform independent

- Execute on a laptop of the cloud, with a simple parameter change
- Available on:
  - Bioconda, Docker, Singularity
  - Google Cloud Platform
  - Amazon Web Services
  - Microsoft Azure
  - HPC Schedulers
- Executable from:
  - Nextflow Tower

kubernetes

Google Cloud

GRID ENGINE

aws

slurm
workload manager

Microsoft Azure

IBM Spectrum LSF

nextflow tower

nf-core pushing Nextflow to the limits

- Community effort to collect curated Nextflow pipelines
  - 2400 Slack users, 1000+ GitHub contributors
- Includes 60+ hi-quality bioinformatic pipelines
  - rnaseq, mag, bactmap, many more
- nf-core/modules has 400+ DSL2* modules available
- Standardized guidelines for developers
- Thorough review process produces robust pipelines

Web platforms that support Nextflow

- Freely available web-platforms for the execution of bioinformatic pipelines
- No command-line knowledge required, allowing users to do more science
- Platforms:
  - Nextflow Tower from Seqera Labs
    - Supports workflows written in Nextflow
    - Platform agnostic and supports many providers
      - HPC, Google Cloud, Microsoft Azure, Amazon Web Services
    - Community showcase of curated pipelines
  - Terra from the Broad Institute
    - Nextflow executed inside Jupyter notebooks
    - Limited to Google Cloud Platform
  - CGC from Seven Bridges
    - Extensive API for executing workflows
    - Limited to Amazon Web Services

Exercise 2: Nextflow Introduction

Executing Nextflow Pipelines

- Use "nextflow run" to execute pipelines

```
nextflow run main.nf
```

- Can also run from GitHub repository

```
nextflow run nf-core/rnaseq
```

- "-resume" allows pipelines to be resumed

```
nextflow run nf-core/rnaseq -resume
```

- Many command-line arguments available, and additional sub-commands

Exercise 2: Install Nextflow and run a few workflows

- Head on over to GitHub: Exercise 2 – Nextflow Introduction

- Together we will:
  - Create a “Nextflow” environment
  - Execute Hello World
  - Execute Bactopia/nf-core test workflows
  - Browse Nextflow outputs

- Wrap up the exercise

A few things to keep in mind about Nextflow

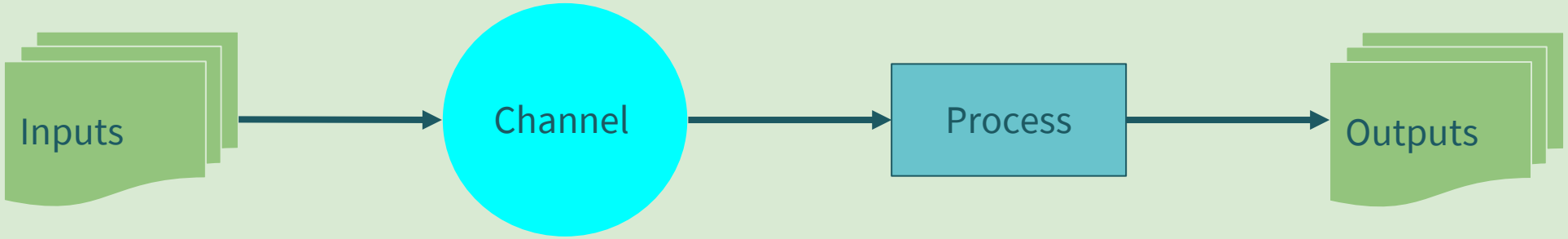
- Missing features:
  - Command line argument parser
    - nf-core has a library to handle this
  - Dry run feature
    - "stub runs" are a decent alternative
- Error messages can sometimes be hard to decipher
- By default, Nextflow uses all available resources

Questions or Comments?

Break Time ~10 minutes

Exercise 3: Nextflow Channels, Operators, and Processes

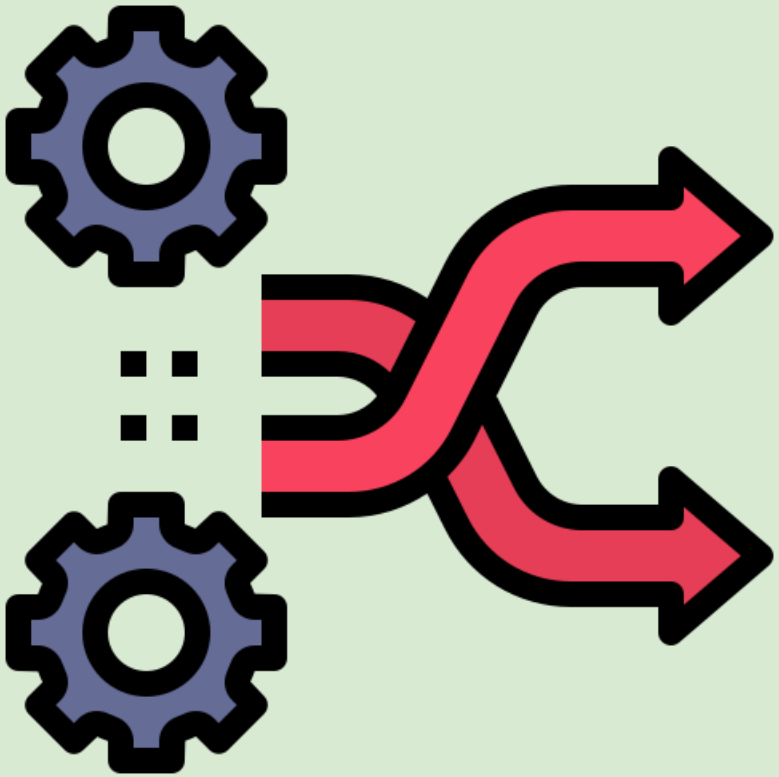
Nextflow Basic Structure



- Channel
  - Queue Channel
    - "*First-in, First-out*" (FIFO) connecting processes and operators
  - Value Channel
    - Stores a single value (*e.g., genome size*)
    - Can be a named list (*e.g., ["id": "my_sample", "genome_size": 360000]*)

- Channels can be used by Operators and Processes

Channel Operators



- Methods to connect channels, or transform values of channels
  - Filtering
  - Transforming
  - Splitting
  - Combining
  - Forking
  - Math

- More than 50 Operators available to use

Processes

- The basic unit for executing user scripts
- 30+ directives adjust optional settings
  - Can be dynamic
- Inputs and outputs are channels
  - Can be optional
- Conditional executions using ‘when’
- The script block executes user code

```
process < name > {

   [ directives ]

   input:
    < process inputs >

   output:
    < process outputs >

   when:
    < condition >

   [script|shell|exec]:
   < user script to be executed >

}
```

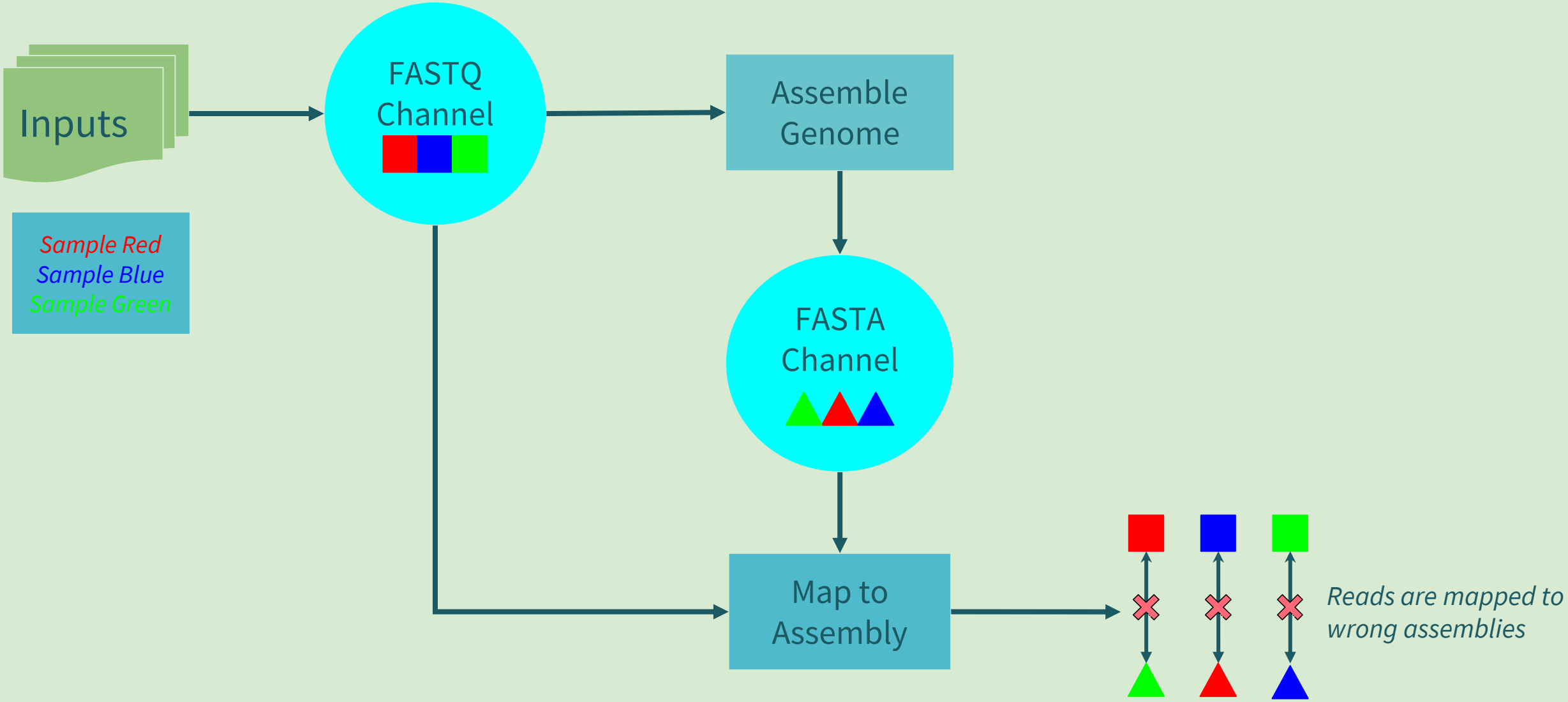
Exercise 3: Nextflow Channels and Processes

- Head on over to GitHub: Exercise 3 - Nextflow Channels and Processes

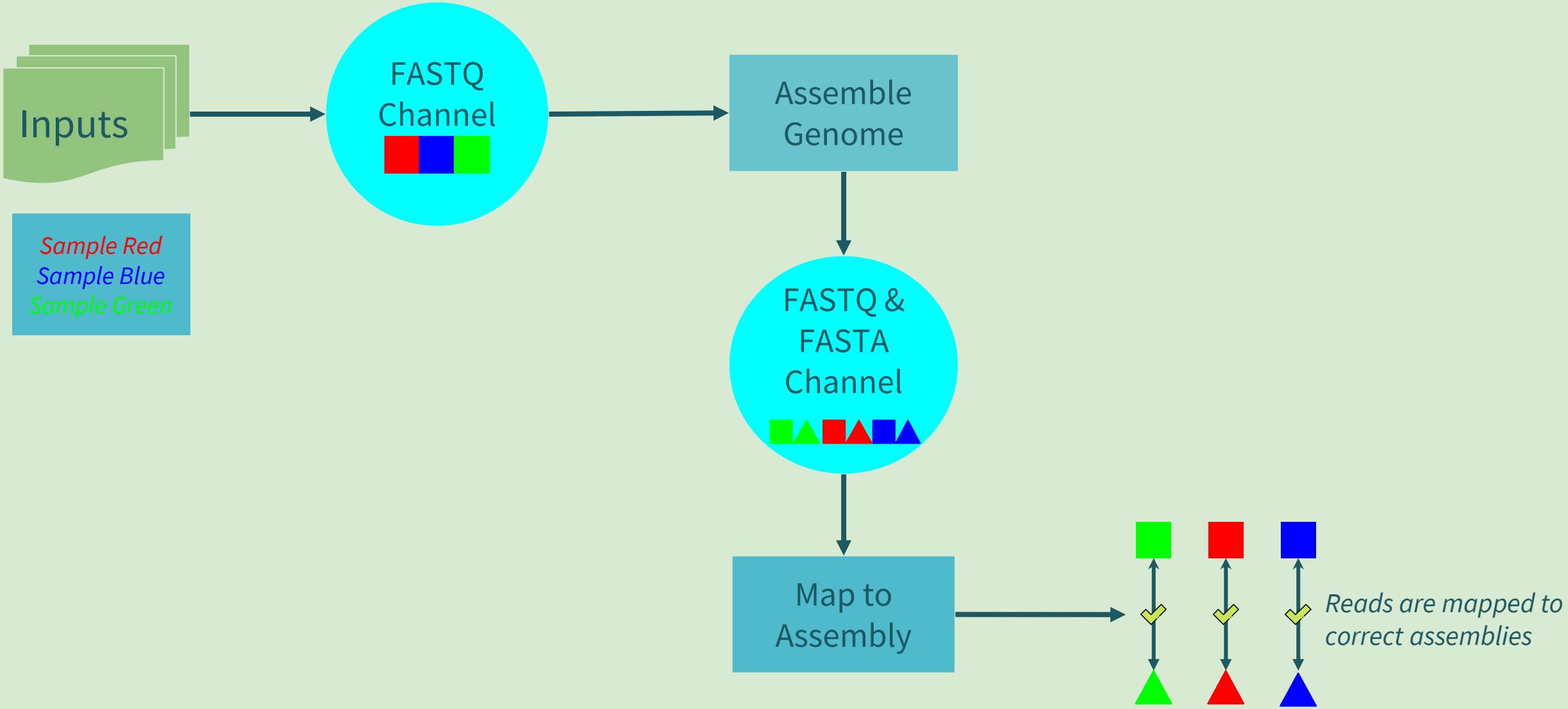
- Together we will:
  - Demonstrate Operators
  - Demonstrate FIFO nature of Channels
  - Pass Channels between Processes
  - Create Channels from Process outputs

- Wrap up the exercise

FIFO Can Cause "*Unexpected*" Results



One Solution: Carry Inputs Across Processes



Questions or Comments?

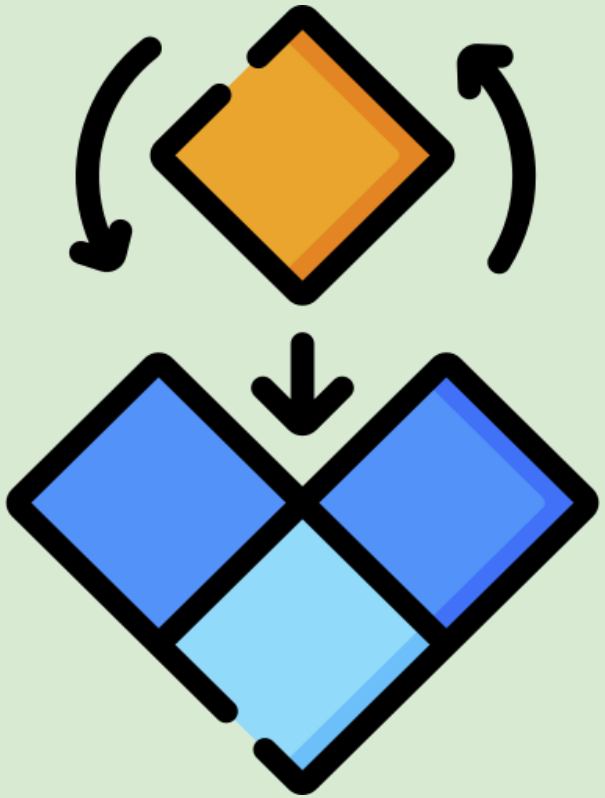
Exercise 4: Nextflow DSL1 vs DSL2

nextflow DSL1

- The original syntax for Nextflow language
- Pipelines were written in a single script
  - Increased maintenance burden as pipeline grew
- Pipelines were not modular
  - Difficult to reuse pieces from one pipeline in another
- Data channels were one-time use
  - Required channels to be duplicated

nextflow DSL2

- Major evolution in the Nextflow language
- Introduced true modularization in Nextflow workflows
  - Modules – A ***reusable*** Nextflow script with a process definition
  - Subworkflows – Multiple modules linked together
- Modules are portable and easily shared between workflows
- Data channels can be used more than once



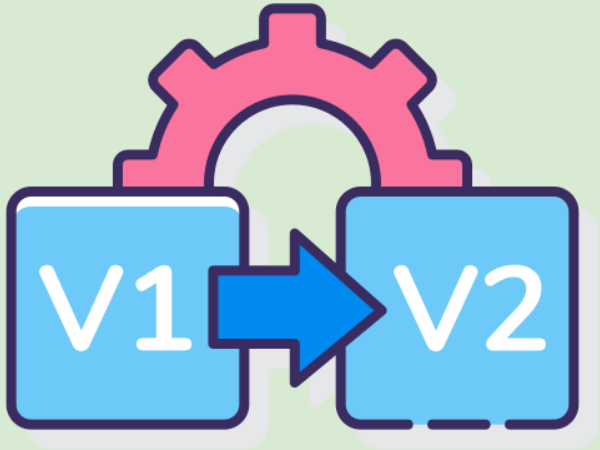
Exercise 4: Nextflow DSL1 vs DSL2

- Head on over to GitHub: Exercise 4 - Nextflow DSL1 vs DSL2

- Together we will:
  - Demonstrate config file usage
  - Walk through “scan-n-trim” DSL1 pipeline
  - Walk through “scan-n-trim” DSL2 pipeline
  - Alter command-line arguements

- Wrap up the exercise

Choose DSL2 Going Forward



- As of v22.04 DSL2 is now the default in Nextflow
  - DSL1 pipelines must tell Nextflow to use DSL1
- New pipelines should be written in DSL2
- nf-core/modules has 500+ ready to use DSL2 modules
  - Rapid prototyping and pipeline building
  - Version controlled, extensive logging, supports Conda, Docker and Singularity

Questions or Comments?

Additional Exercise Ideas

Additional Exercise Ideas

- Use “publishDir” to adjust how files are output
- Use “label” to adjust runtime requirements
- Create a “config profile” to switch between Conda and Docker
- Create a DSL2 pipeline using modules from nf-core

Additional Nextflow Resources

Nextflow and Bioconda specific support channels

- Nextflow Slack
  - General Nextflow Support
  - Nextflow devs regularly helping users
- Nf-core Slack
  - Nf-core and Nextflow devs regularly helping users
- Bioconda Gitter
  - Mostly related to submitting Bioconda recipes
- StaPH-B Slack
  - Many users from the state public health labs

Nextflow Continuing Education

- Nextflow Training Workshop
  - 10+ hour Nextflow course given by Seqera Labs
- Nextflow and nf-core
  - 10+ hour course being developed by nf-core members available from Software Carpentries
- Reproducible, scalable, and shareable analysis workflows with Nextflow
  - 10+ hour Nextflow course by Sateesh Peri, Michael Cipriano, and Matthew Hunter Seabolt

Useful Nextflow Links

- Nextflow Documentation
  - Extensive documentation of Channel operators, Process directives, executors, and many other Nextflow features
- Awesome Nextflow
  - A curated list of Nextflow pipelines
  - Presentations, tutorials, videos, etc…
- Nextflow Patterns
  - A repository of ways to do things in Nextflow

Useful nf-core Links

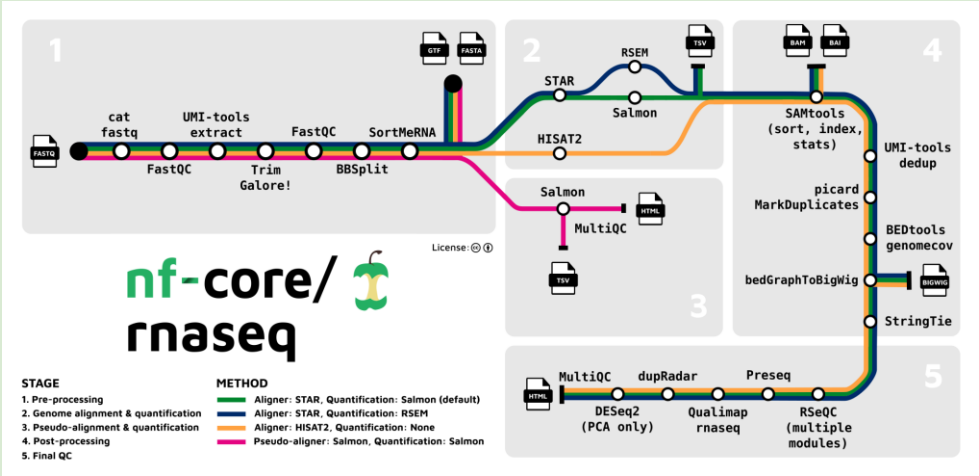
- nf-core
  - Nf-core's main website, includes documentation for each pipeline, module documentation, etc…
- nf-core bytesize Talks
  - 20-minute videos covering all things nf-core, Nextflow, and best practices
- nf-core/modules
  - More than 400 ready-to-use DSL2 modules
- nf-core/configs
  - Numerous (80+) Nextflow configs for inspiration
- nf-core/tools
  - Python package with helper tools for the nf-core community.

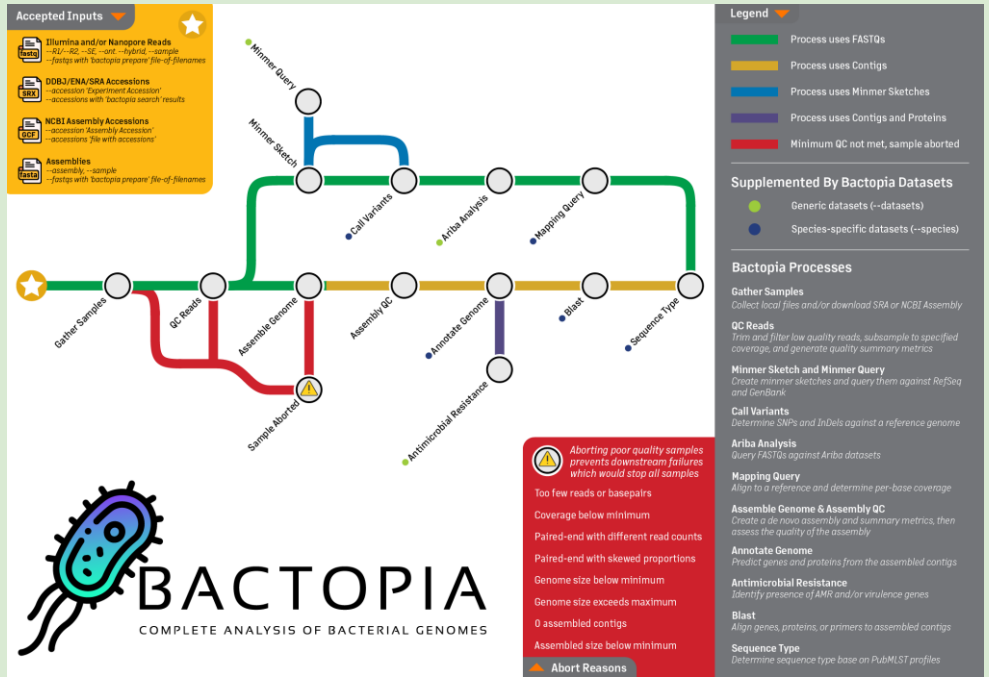
Nextflow DSL2 pipelines for inspiration

- Make use of them, or use their source code for ideas

- nf-core/rnaseq
  - RNA sequencing analysis pipeline with gene counts and extensive QC
  - Cutting-edge on Nextflow features

- Bactopia *(shameless plug)*
  - For the complete analysis of bacterial genomes
  - Illumina and Nanopore support, 130+ bioinformatic tools, 30+ additional workflows





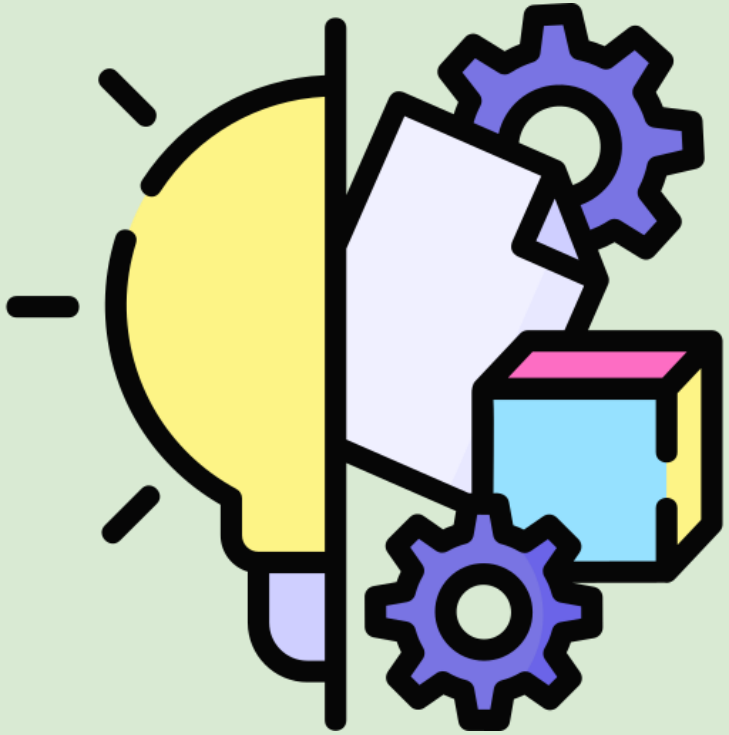
Final Wrap Up

Workflow managers make bioinformatics manageable

- Manages the execution of pipelines
  - Linking inputs/outputs of bioinformatic tools
  - Queuing jobs locally, on clusters, or the cloud
  - Logging, errors, audit trails
- Promote reproducible and reusable science
- Common workflow languages:
  - Nextflow, WDL, and Snakemake
- Pick one that works for you

The *beginnings* of a strong bioinformatic skill set

- Accessibility and Portability
  - Docker
  - Conda
- Reproducibility
  - Workflow Description Language (WDL)
  - Nextflow
- Bioinformatic Platforms
  - Terra.bio
  - Nextflow Tower



But don’t let it end here!

- *This has only been the <u>beginnings</u> of a strong bioinformatic skill set*
- Practice, practice, practice
- Make mistakes, break things, debugging error messages is a top-tier skill
- Try to figure things out, but don’t hesitate to ask questions

Questions or Comments?