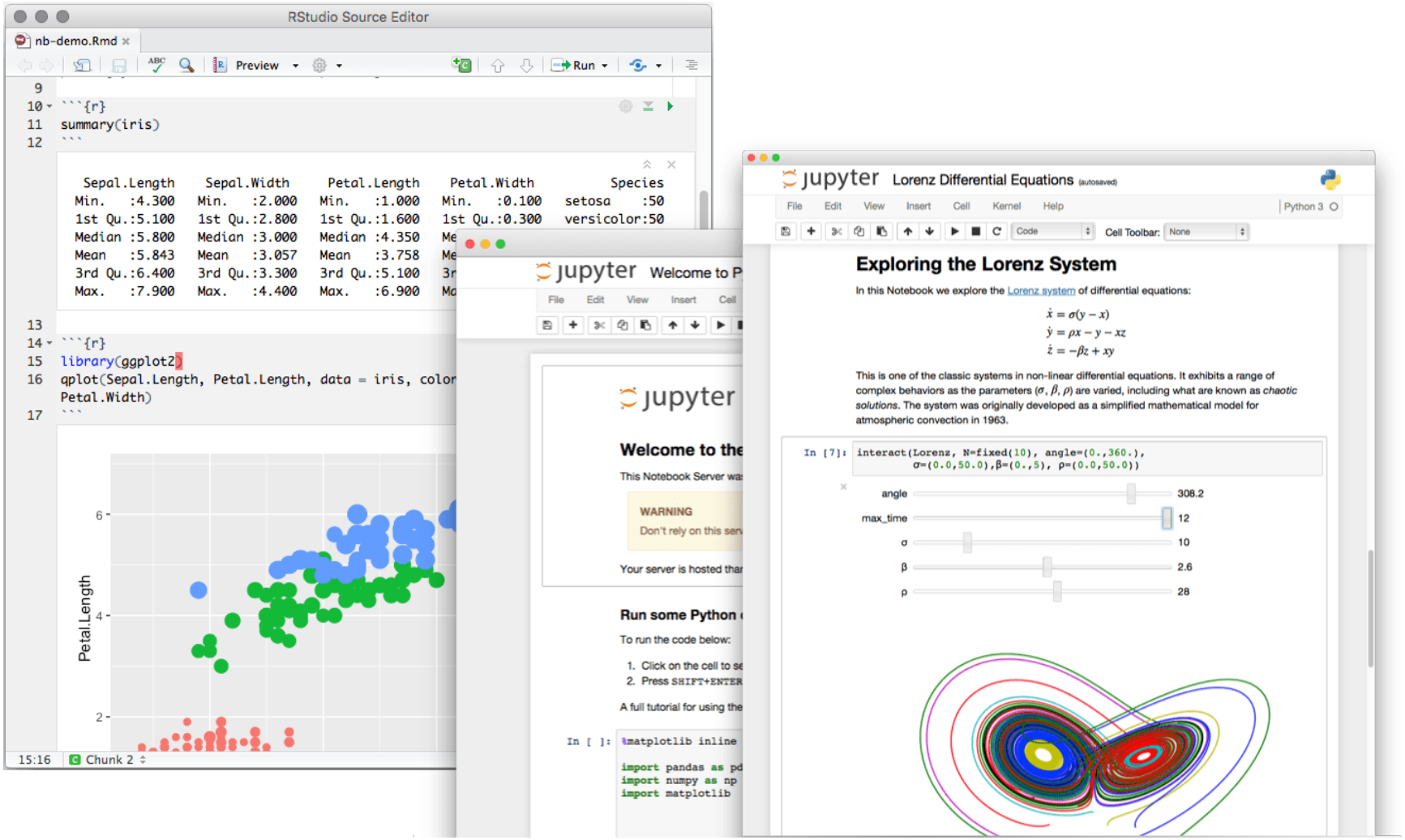
Going with the flow

DR. IAN SUDBERY
MOLECULAR BIOLOGY AND BIOTECHNOLOGY
I.SUDBERY@SHEFFIELD.AC.UK
@IanSudbery
www.github.com/sudlab

Using workflow managers to co-ordinate multistep analysis pipelines across multiple compute nodes in a reproducible manner.

Traditional HPC jobs are single monolithic programs using multi-node parallelism

Today many researchers use notebooks on clusters to do interactive/interpretive analysis of datasets



Research computing spectrum

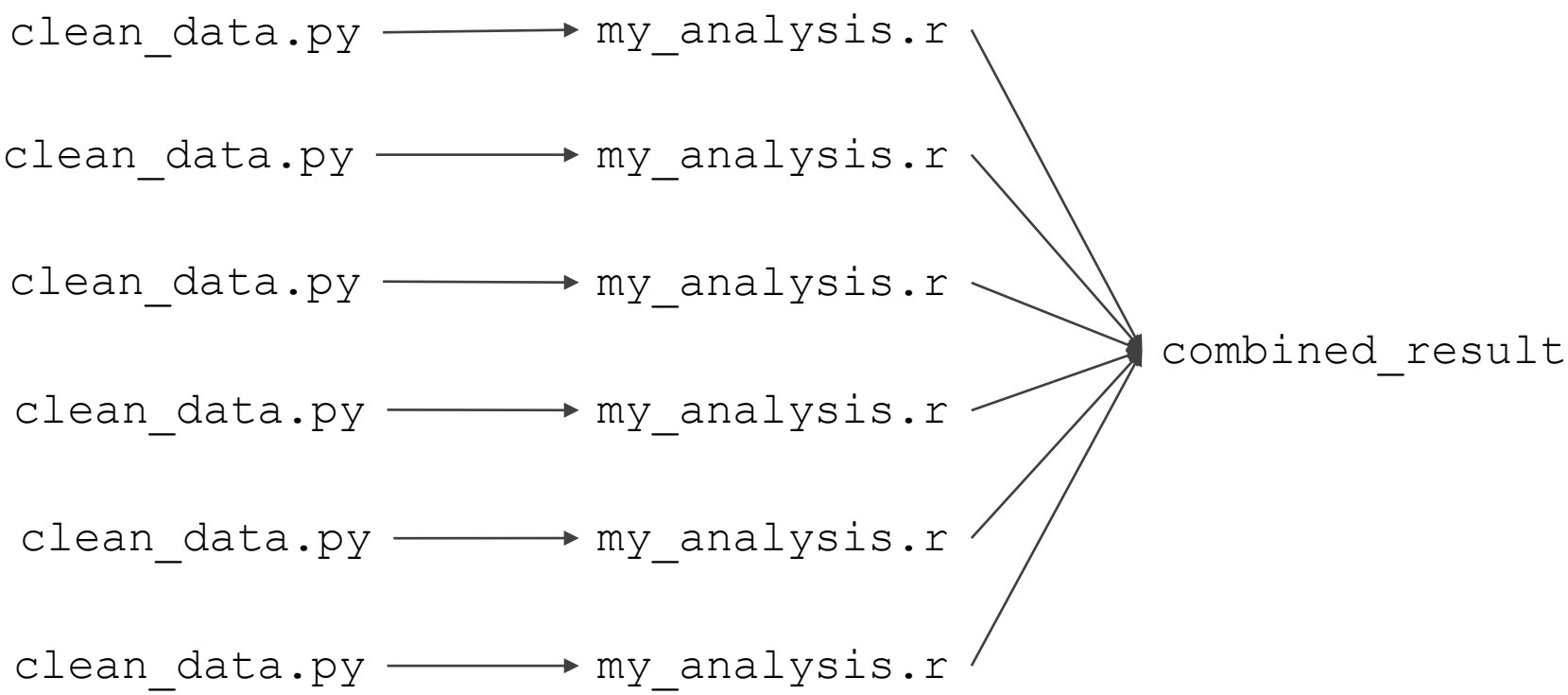
Single, large, long running, multimode jobs

?

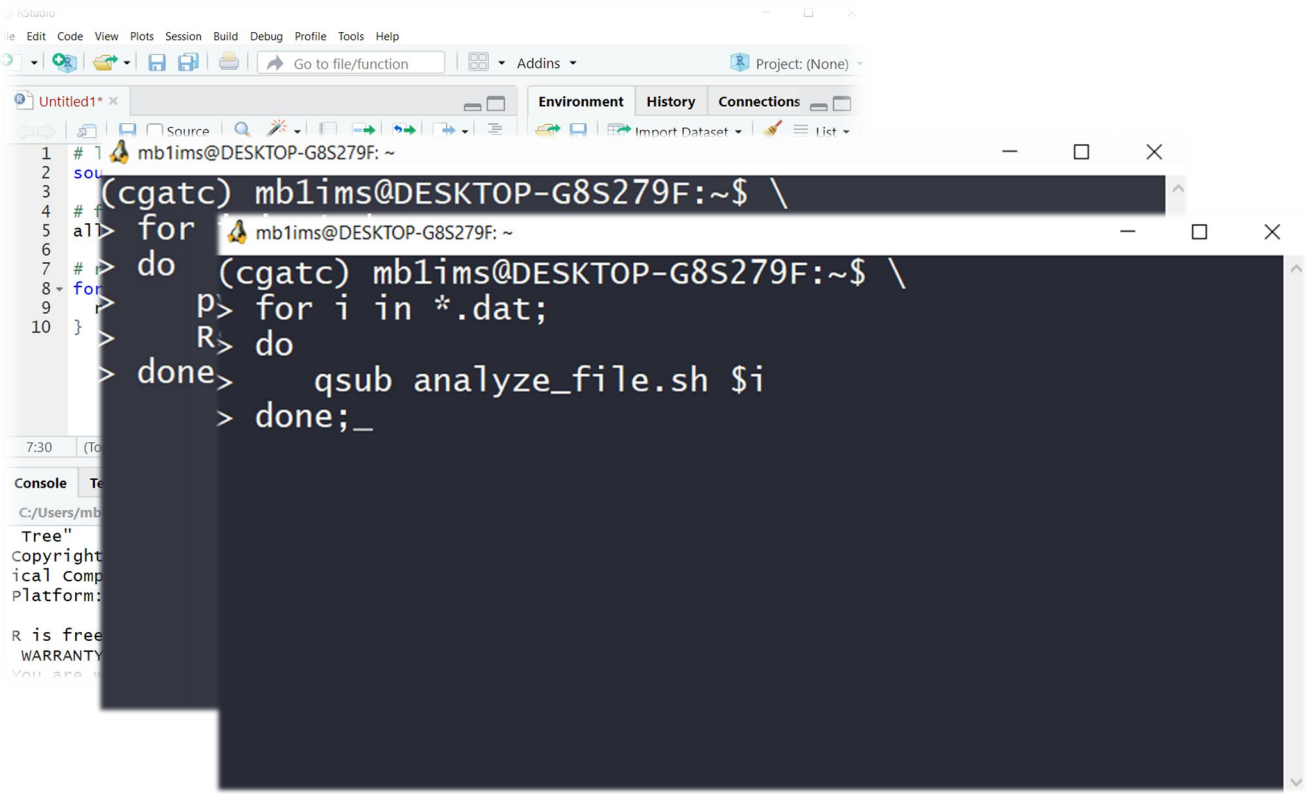
Single core, quick running, interpretive analysis

e.g. a climate model

Regression analysis of a (quite) big dataset


clean_data.py
clean_data.py
clean_data.py
clean_data.py
clean_data.py
clean_data.py
my_analysis.r
my_analysis.r
my_analysis.r
my_analysis.r
my_analysis.r
my_analysis.r
combined_result

Carrying out multi-step analyses by hand



Reproducibility

```
results = code(data)
```

- Typing at a terminal is BAD NEWS for reproducibility
- Notebooks (for low intensity work)
- Containers
- Neither very easily work with multi-node parallelism

1. Easy/Automatic
2. Reproducible
3. Generalizable/Scalable

Workflow manager

- Specify dependencies between tasks
- Check if which dependencies need updating
- Only run tasks that need updating
- Do all this unsupervised.

Modern workflow managers

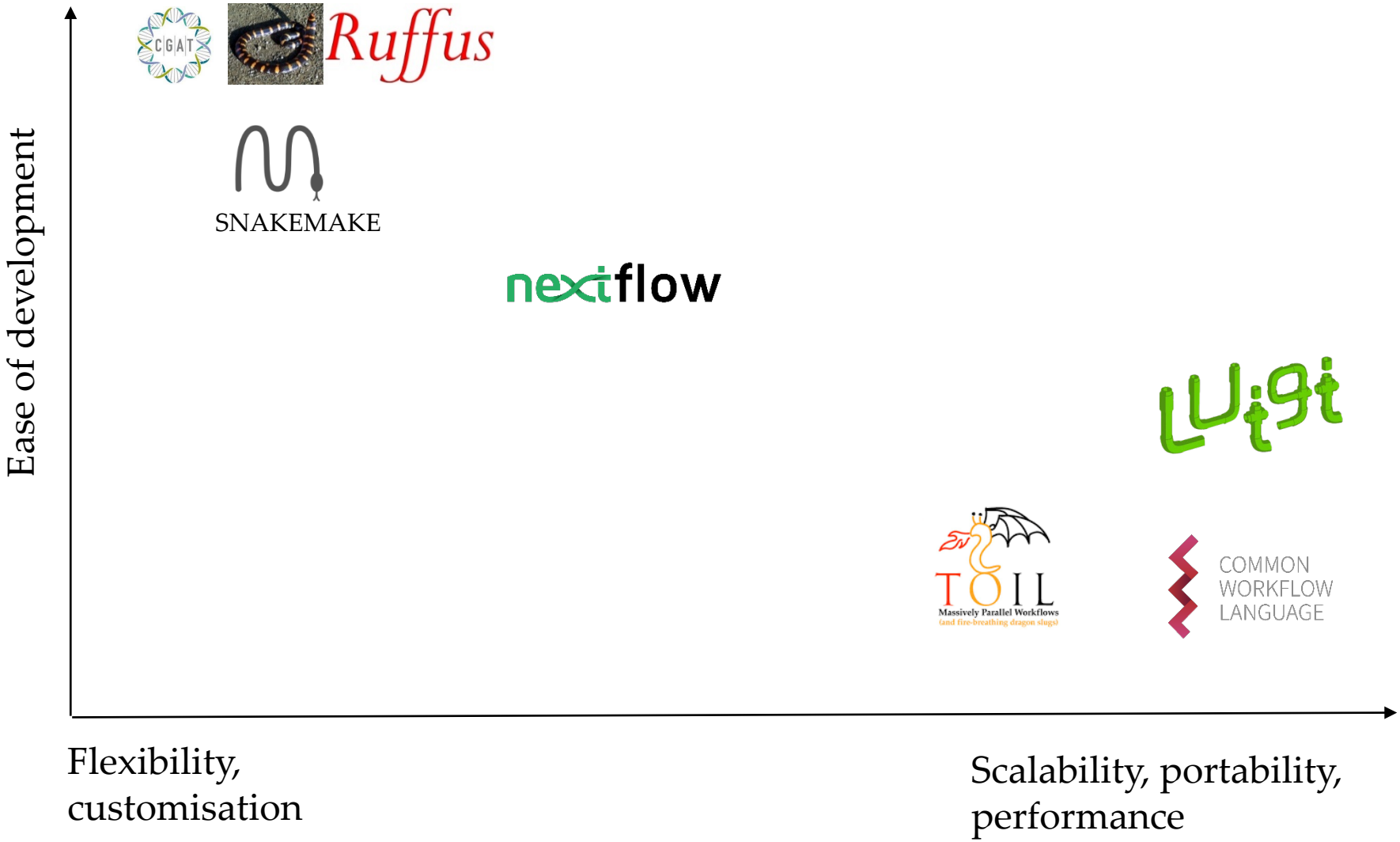
- Either DSLs, configuration based or library
- Allow more complex forms of dependency
- Automatically submit each job to the cluster
- Monitor for successful completion and automatically submit next job
- Parameterizable
- Extensive logging

Modern Workflow managers

May also provide:

- `conda`/`singularity`/`docker` integration
- Use cloud compute and/or storage as well as local cluster
- Allow (distributed) execution of arbitrary code as well as shell scripts
- Helper functions for common analysis tasks

Some modern WFM

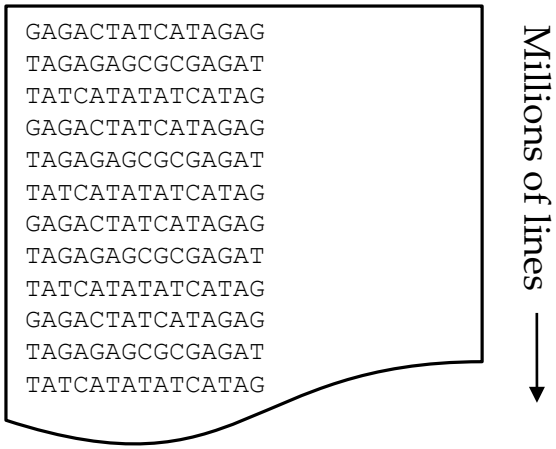


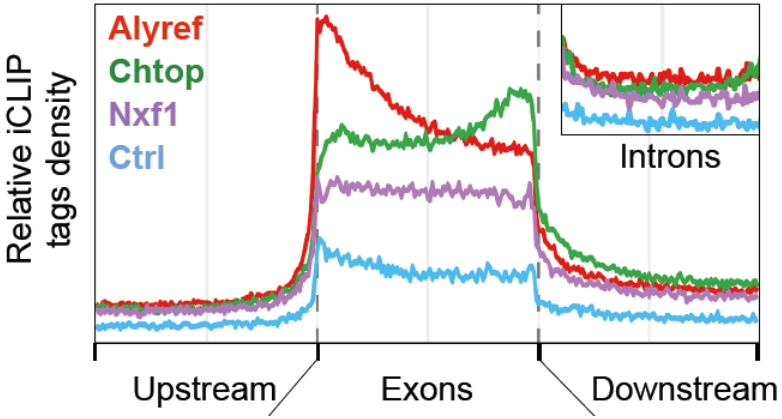
	CGAT	SNAKEMAKE	nextflow
Language	Python	DSL	DSL
Dependency Paradigm	Explicit	Implicit (pull)	Implicit (push)
Rich dependency graphs	Yes	Partial	Yes
Conda integration	Yes	Yes	Yes
Singularity/docker	Coming soon	Yes	Yes
Arbitrary code	Python	Python	Any interpreted
Cloud Execution	No	Kubernates	Amazon Batch
Cloud storage	Google/S3	Many	Many
Functions for common analysis	Yes	No	No

Demonstration

It should take less time, effort and thought to it the right way than to do it the wrong way

Gene profiles

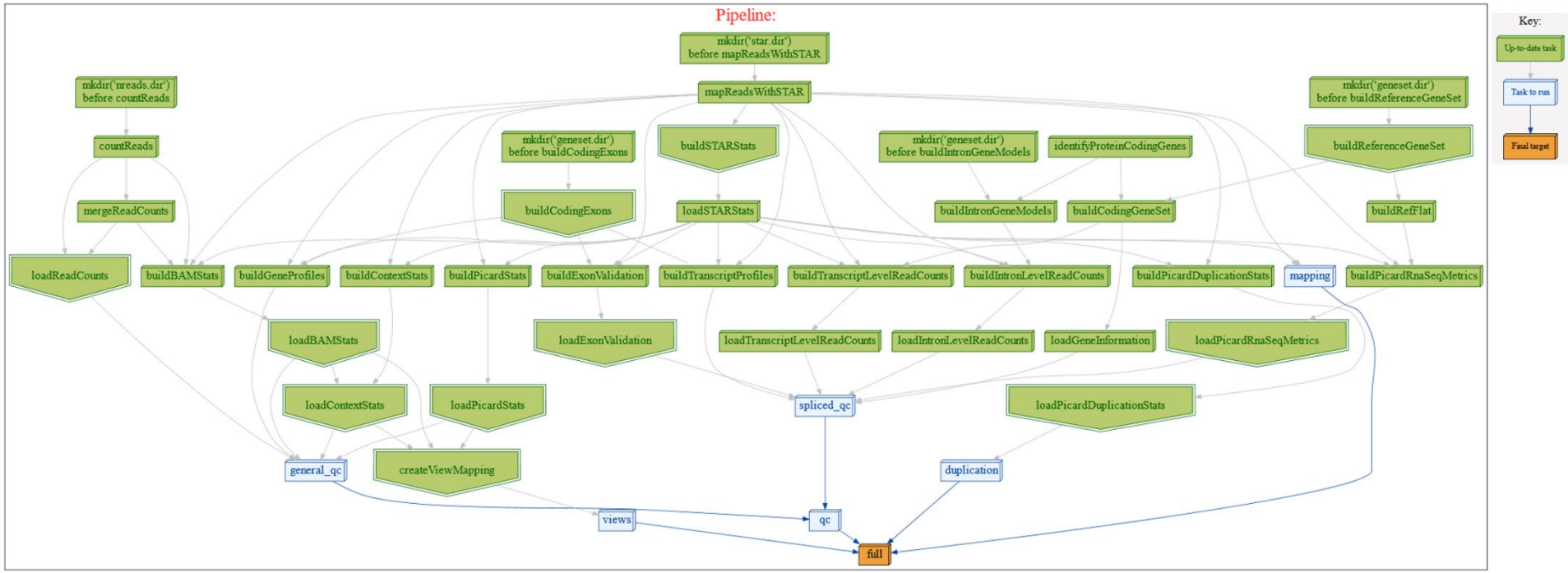




Ruffus dependency types

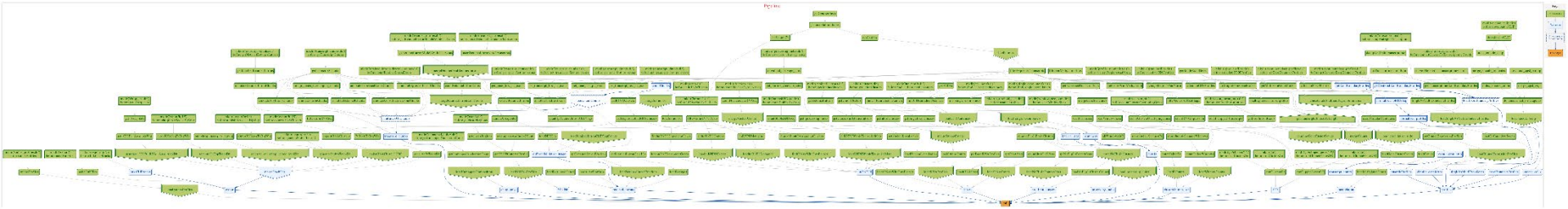
Originate	None to one
Transform	One to One
Split	One to many
Merge	Many to one
Collate	Many to fewer
Subdivide	Many to more
Follows	Dependency without common files
Files	Arbitrary relationship
Permutations Product Combinations Combinations_with_replacement	Combinatorics

Pipelines can get quite complex…



`pipeline_mapping`

Really very complicated!



Summary

- Automated farming and monitoring of pipelines of jobs to the cluster
- Create fully logged and reproducible workflows
- Generalizable and scalable

- Should be easer than writing a SGE submission script and faster than running in an interactive session

- Install with
  `conda install -c bioconda -c conda-forge cgatcore`

Acknowledgements

Sudbery Lab for Computational Genomics @ TUOS

Dr. Cristina Alexandru-Crivac
Jaime Alvarez-Benayas
Justin Coyne
Magdelena Dabrowska
Sumeet Deshmurkh
Jacob Parker
Ivaylo Yonchev

MRC Computational Genomics Analysis and Training/Tools

Dr. Adam Cribs
Sebastian Luna-Valero
Dr. Charlotte George
Dr. Antonio Berlanga-Taylor
Dr. Stephen Sansom
Dr. Tom Smith
Dr. Nicholas Ilott
Dr. Jethro Johnson
Jakub Scaber
Dr. Katherine Brown
Dr. David Sims
Dr. Andreas Heger

Dr. Leo Goodstat (Ruffus)

https://cgaticore.readthedocs.io

Cribbs AP, *et al.* *F1000Research* 2019, 8:377

https://snakemake.readthedocs.io

Köster, J and Rahmann, S. Bioinformatics 2012, 28:2520

https://nextflow.io

P. Di Tommaso, et al. Nature Biotechnology 2017 35, 316