LEIDEN UNIVERSITY MEDICAL CENTER

Automatic scheduling with `make`.

Jeroen F. J. Laros

Leiden Genome Technology Center

Department of Human Genetics

Center for Human and Clinical Genetics

Introduction

Some figures

Figure 1: Illumina HiSeq 2000

Introduction

Some figures

Figure 1: Illumina HiSeq 2000

At this moment:

- ±100 exomes per run.
- ±5 full genomes per run.
- Analysis can take up to 3 days per exome.
- Full genome sequencing is coming.

Introduction

Clusters

Figure 2: Dell M610 blade server

Using a cluster

Sun Grid Engine

A cluster needs a job scheduler.

- Open source batch-queuing system developed by Sun Microsystems.
- Now owned by Oracle, no longer free.
- Fully compatible alternatives:
  - Son of Grid.
  - Open Grid Scheduler.
  - Univa Grid Engine (commercial).

http://gridscheduler.sourceforge.net

Using a cluster

Sun Grid Engine

A cluster needs a job scheduler.

- Open source batch-queuing system developed by Sun Microsystems.
- Now owned by Oracle, no longer free.
- Fully compatible alternatives:
  - Son of Grid.
  - Open Grid Scheduler.
  - Univa Grid Engine (commercial).

http://gridscheduler.sourceforge.net

Using a cluster

Pipelines

Classical pipeline:

- Linear.
- Shell script.
    - Or with the exec() function in your favourite language.

Using a cluster

Pipelines

Classical pipeline:

- Linear.
- Shell script.
    - Or with the exec() function in your favourite language.

Drawbacks:

- Parallelisation is done manually.
- Syncing must be done manually.
- Extensive error handling.
- No save points.
- No dry runs.

Using a cluster

A different viewpoint

What if. . .

- All commands are atomic.
  - We describe input and output.
- We build a *dependency graph*.
- Trace a path in this graph to find a workflow.

Using a cluster

A different viewpoint

What if. . .

- All commands are atomic.
  - We describe input and output.
- We build a *dependency graph*.
- Trace a path in this graph to find a workflow.

This way we do not need to:

- Design a workflow.
- Figure out which parts can be run in parallel.

Theoretical solution

Petri nets

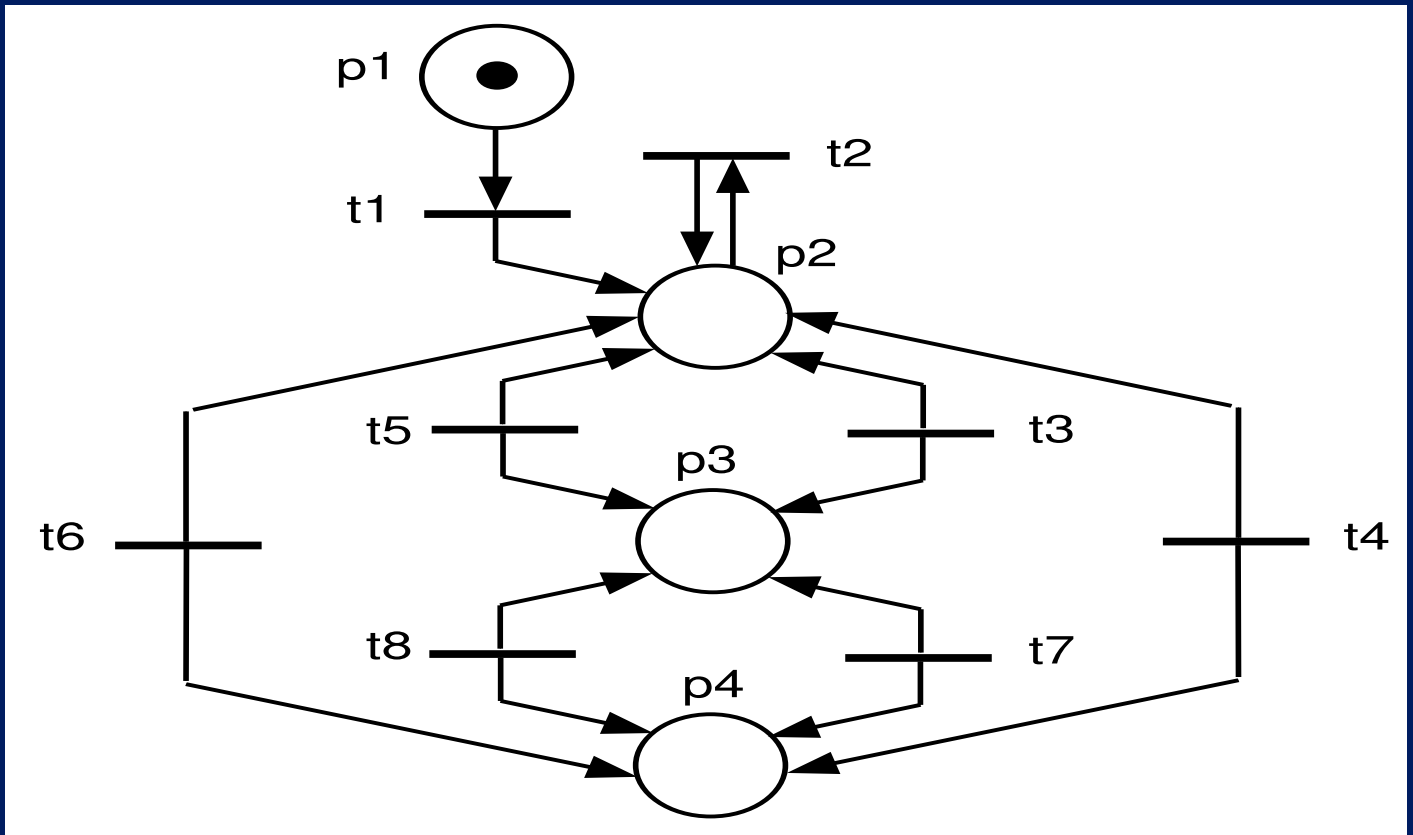

Figure 3: A model for parallelism.

Theoretical solution

Petri nets

"Mathematical modelling language for the description of distributed systems."

Background:

- Described first in 1939 by Carl Adam Petri (age 13).
- Originally intended to describe chemical processes.
- Graphical notation for stepwise processes.
- Choice, iteration, and concurrent execution.

http://en.wikipedia.org/wiki/Petri_net

Theoretical solution

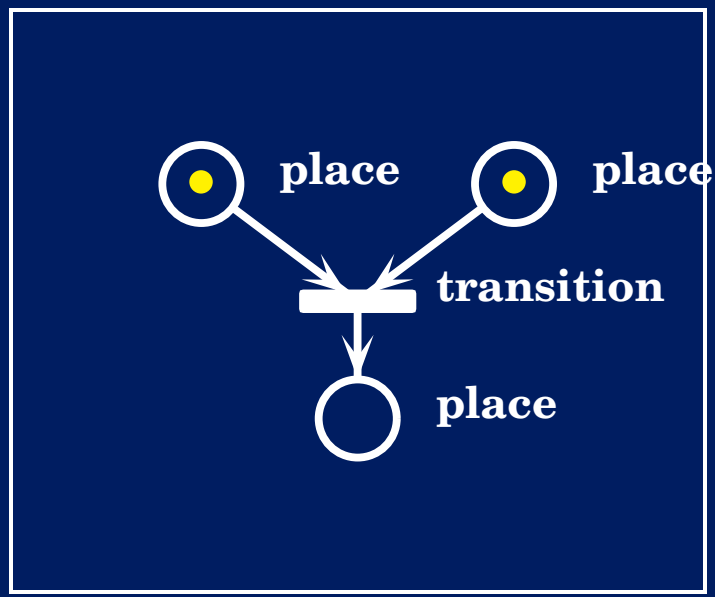

Figure 4: A simple Petri net

A *transition* has:

- A *preset*: a set of *input places*.
- A *postset*: a set of *output places*.

The output of one transition can be the input of an other.

Theoretical solution

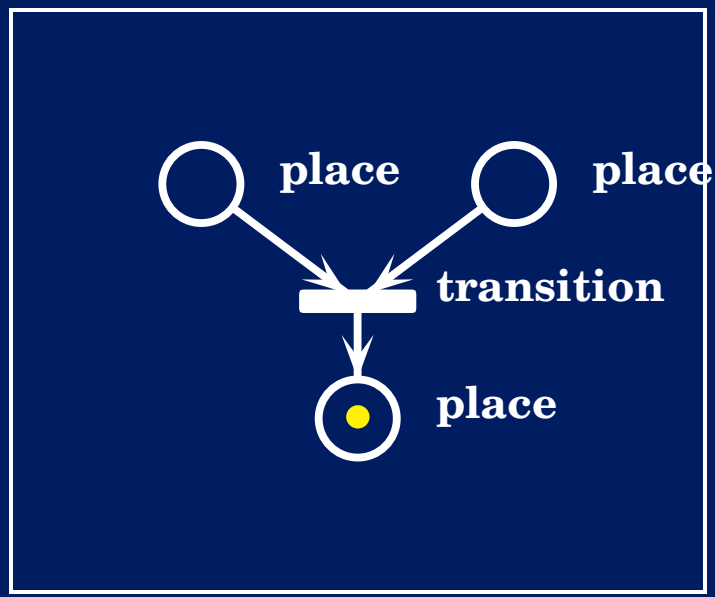

Figure 4: A simple Petri net

A *transition* has:

- A *preset*: a set of *input places*.
- A *postset*: a set of *output places*.

The output of one transition can be the input of an other.

Theoretical solution

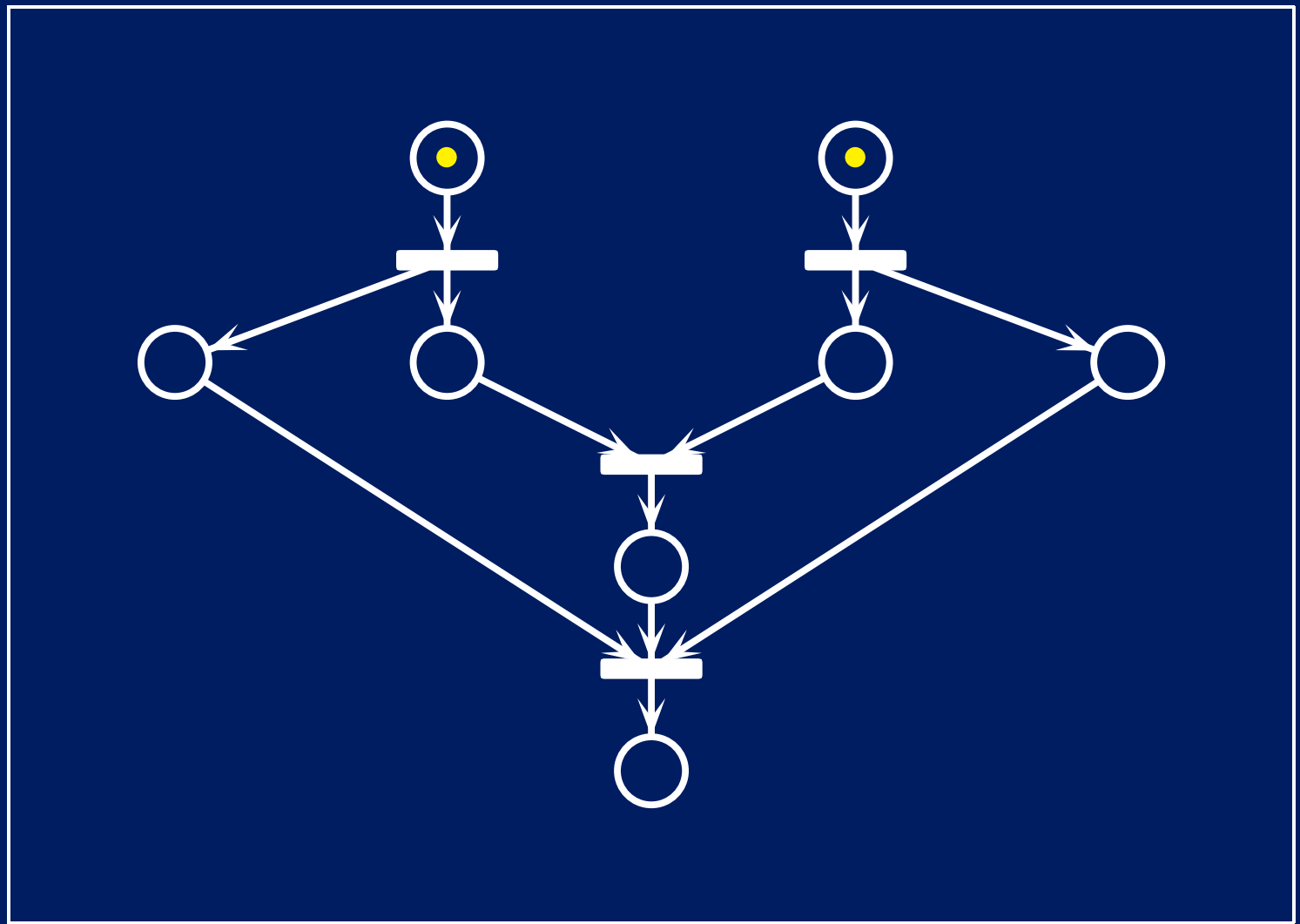

Figure 5: A more complicated Petri net

Theoretical solution

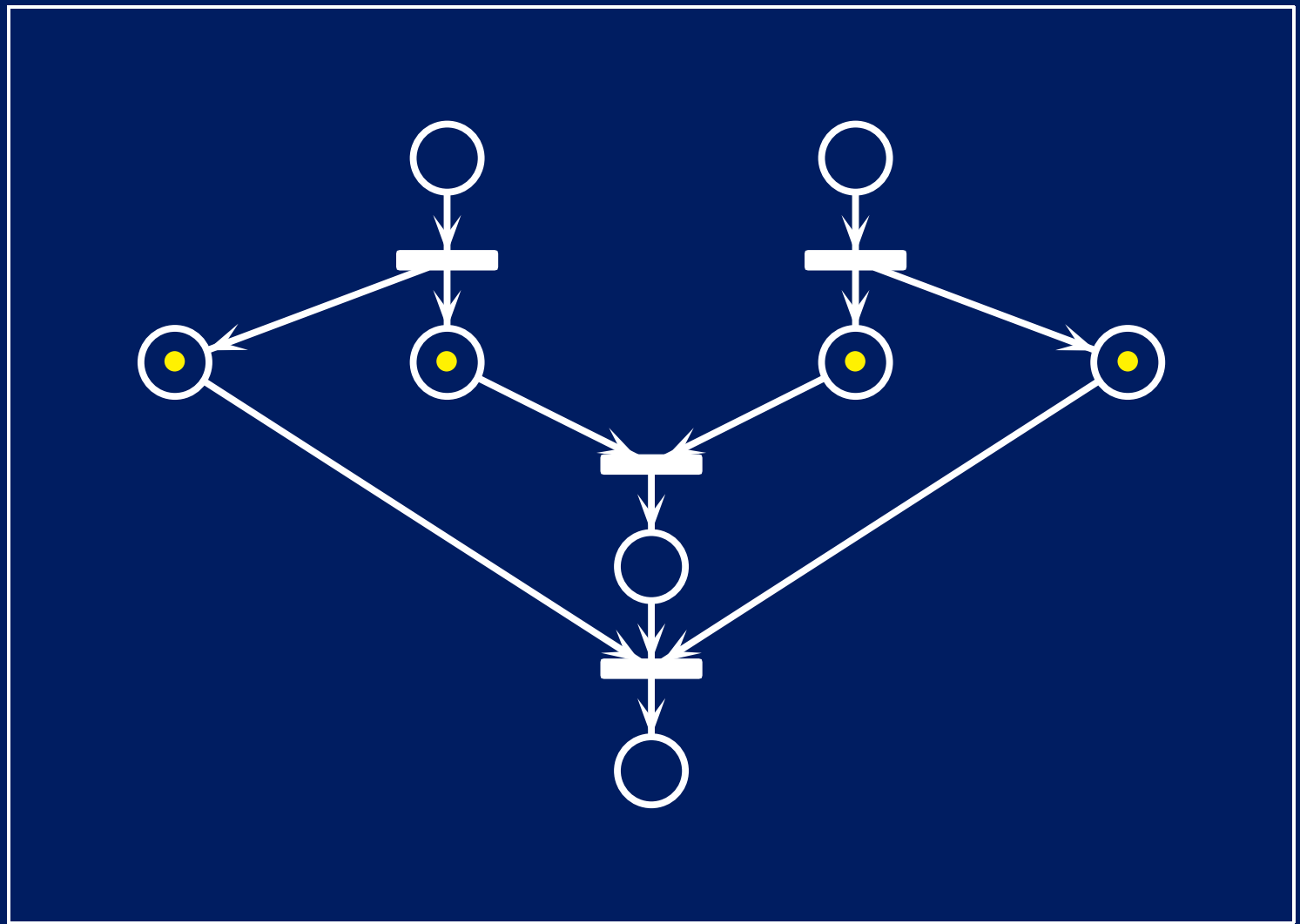

Figure 5: A more complicated Petri net

Theoretical solution

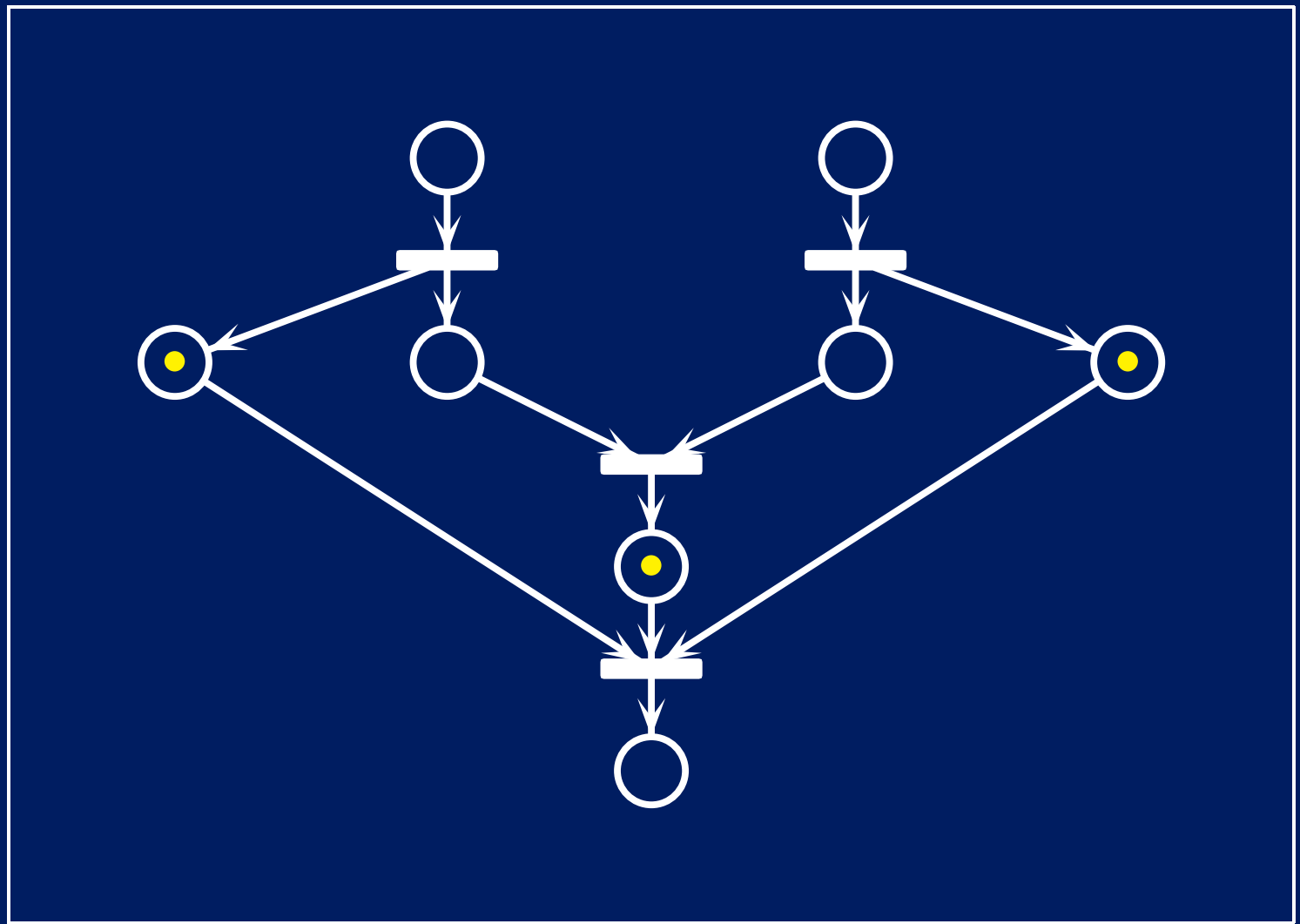

Figure 5: A more complicated Petri net

Theoretical solution

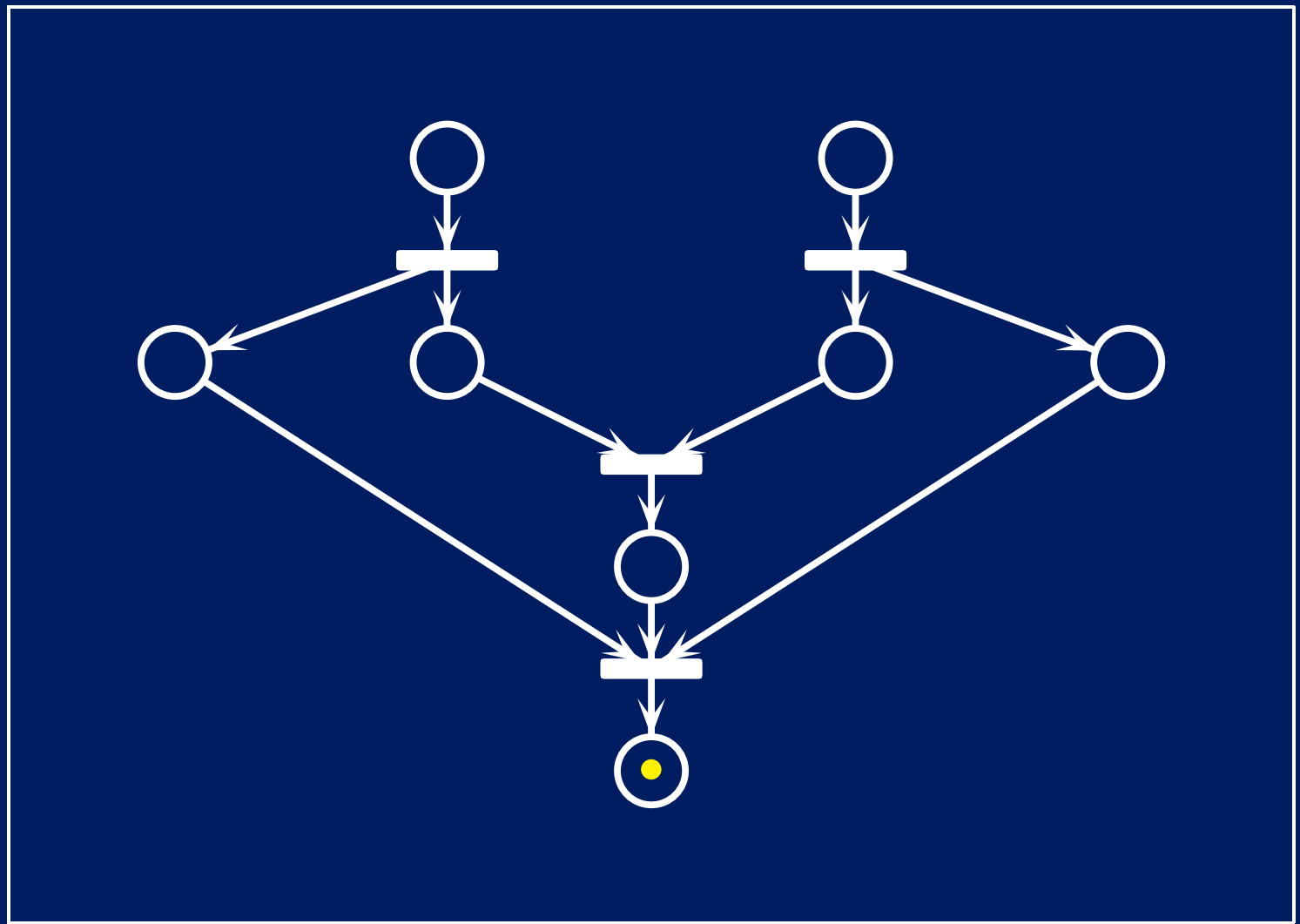

Figure 5: A more complicated Petri net

Theoretical solution

Modelling parallel processes

Key observation:

- A graph is uniquely defined by a list of nodes and their edges.

This means that you don't need to focus on the big picture.

- Each process can be modelled individually.
- Processes are linked based on the in- and outputs.

Practical solution

make

"Utility that automatically builds programs and libraries from source code by reading files called makefiles which specify how to derive the target program."

With a couple of minor alterations we can also use it for pipelines.

- Source code $\Rightarrow$ Raw data
- Program $\Rightarrow$ Result
- Building $\Rightarrow$ Analysing

http://en.wikipedia.org/wiki/Make_(software)

Practical solution

The anatomy of a Makefile

```
1 target: prerequisites
2   recepe
```

Listing 1: Makefile snippet

In the recipe, some special variables are available:

$@	Name of the target.
$<	The first prerequisite.
$^	All prerequisites.

Suffix rules

The % can be used for implicit targets, e.g., %.bam: %.sam

Practical solution

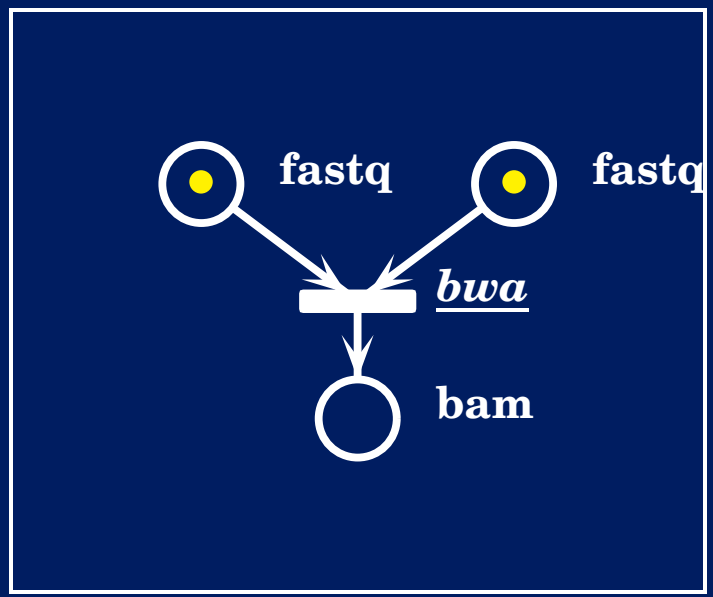

Figure 6: A simple workflow

```
1 %.bam: %_1.fq %_2.fq
2    bwa sampe reference.fa $^ > $@
```

Listing 2: Makefile snippet

Practical solution

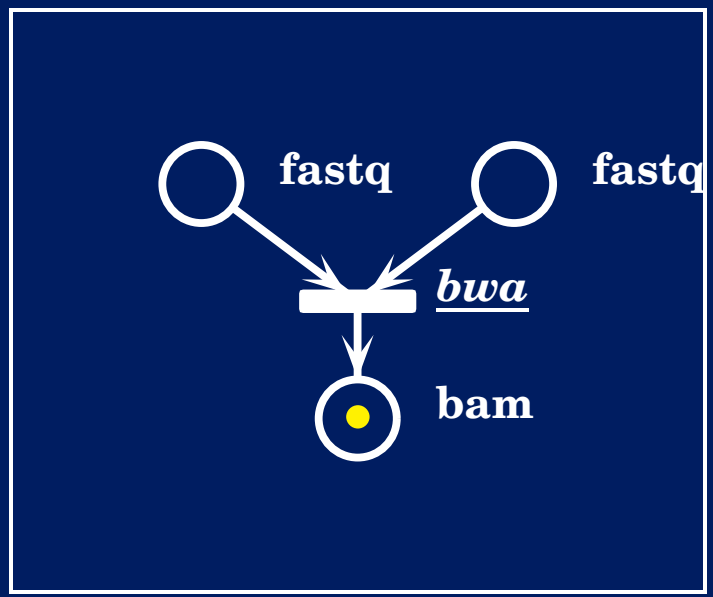

Figure 6: A simple workflow

```
1 %.bam: %_1.fq %_2.fq
2    bwa sampe reference.fa $^ > $@
```

Listing 2: Makefile snippet

Practical solution

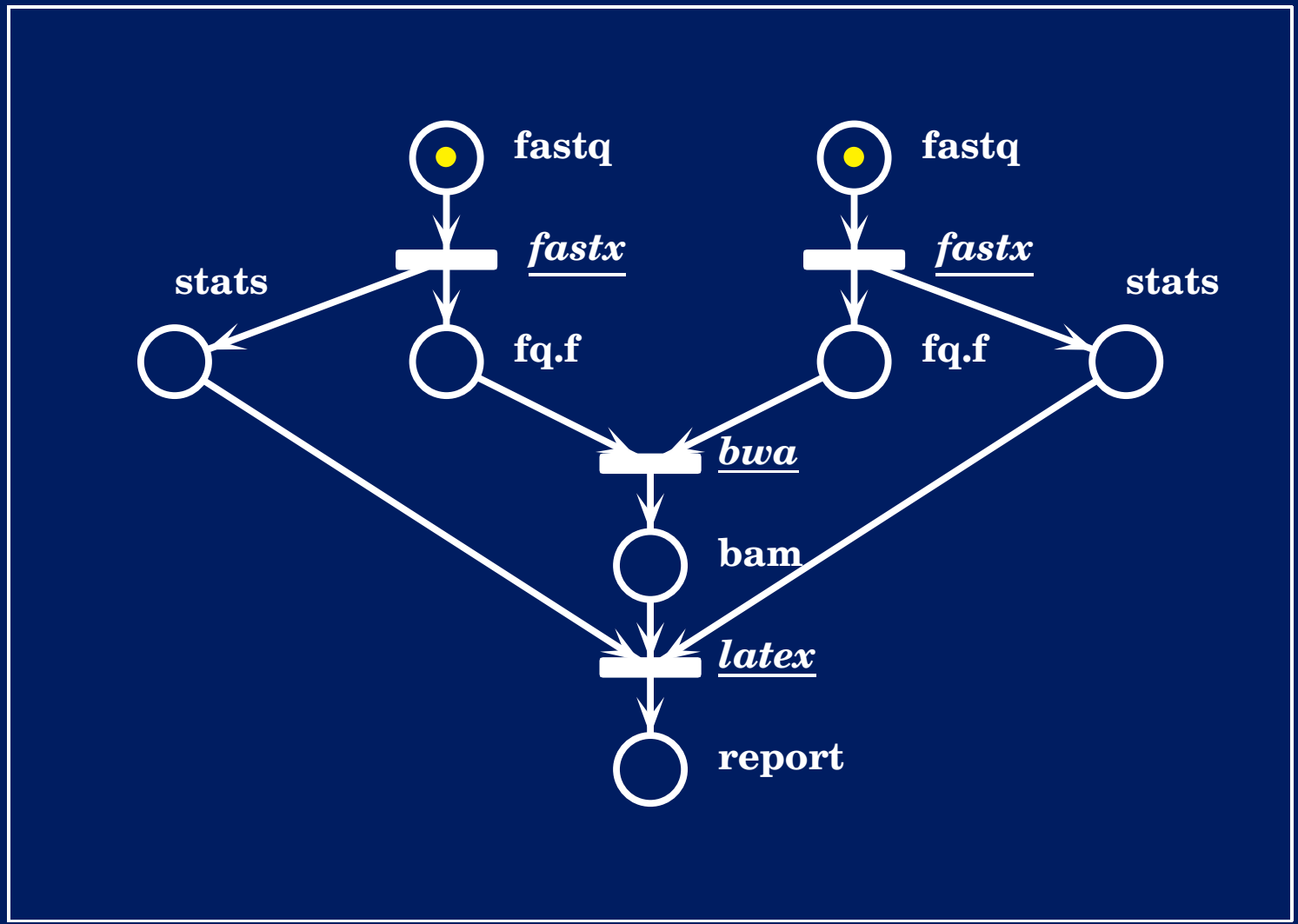

Figure 7: A parallel workflow

Practical solution

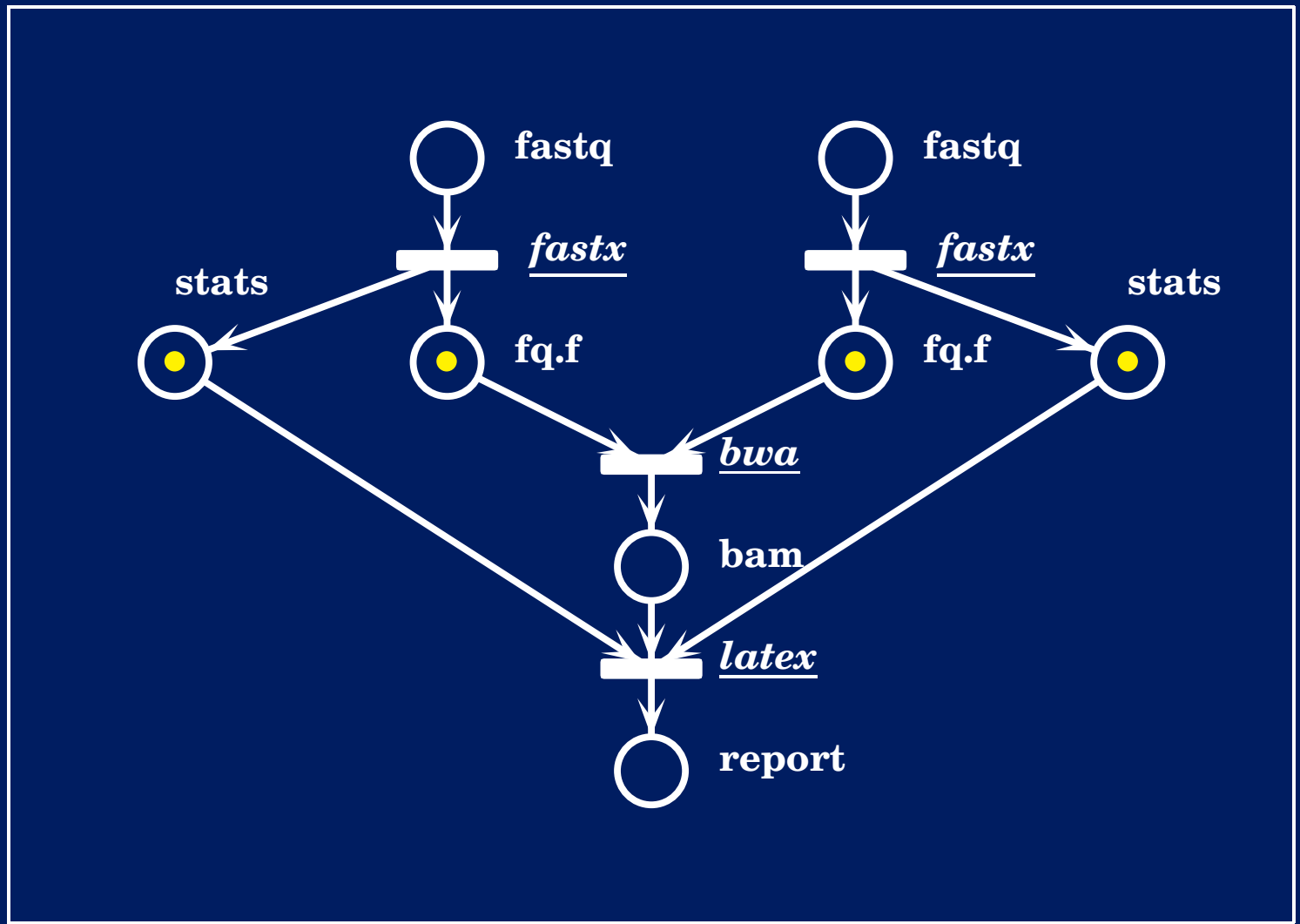

Figure 7: A parallel workflow

Practical solution

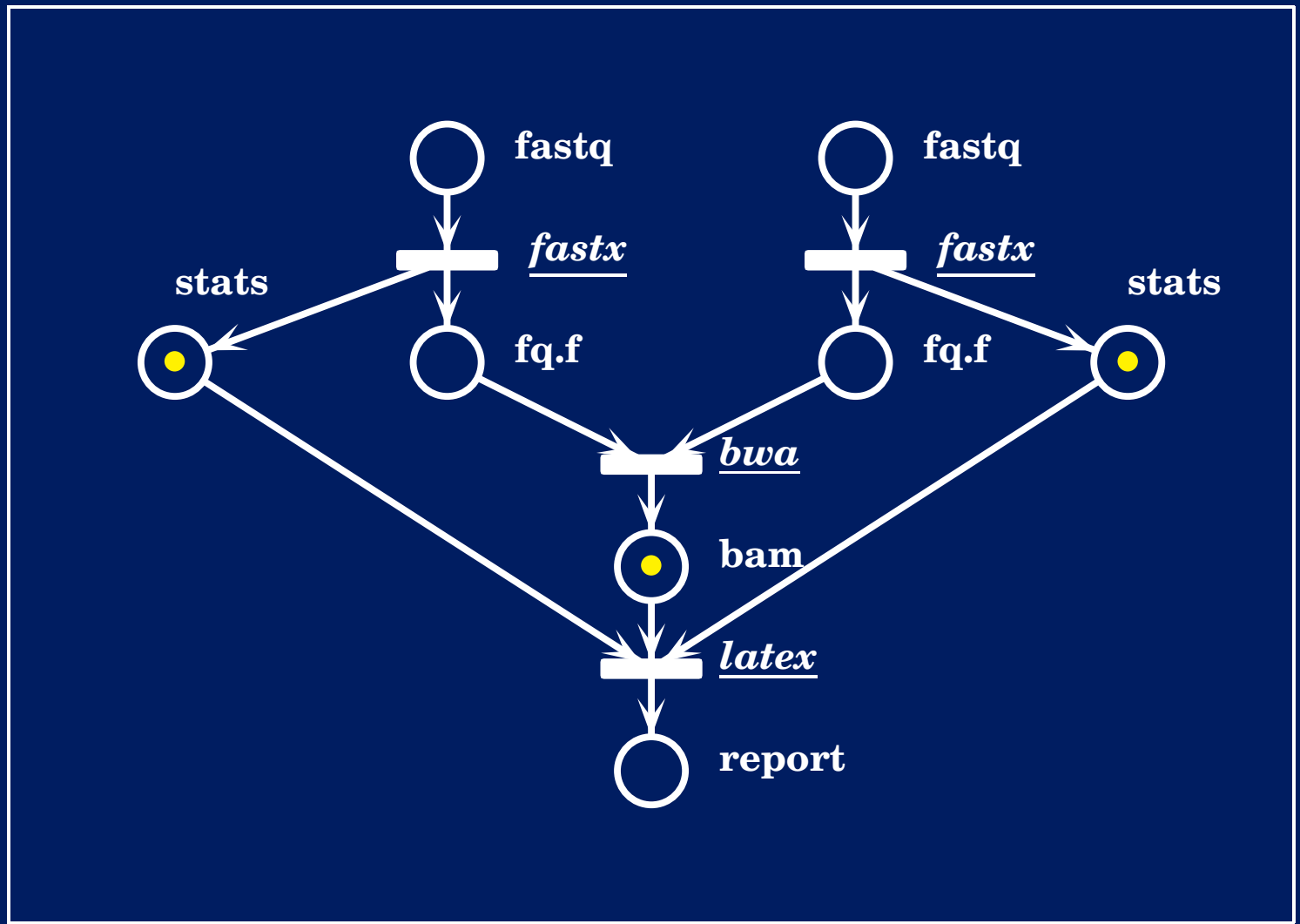

Figure 7: A parallel workflow

Practical solution

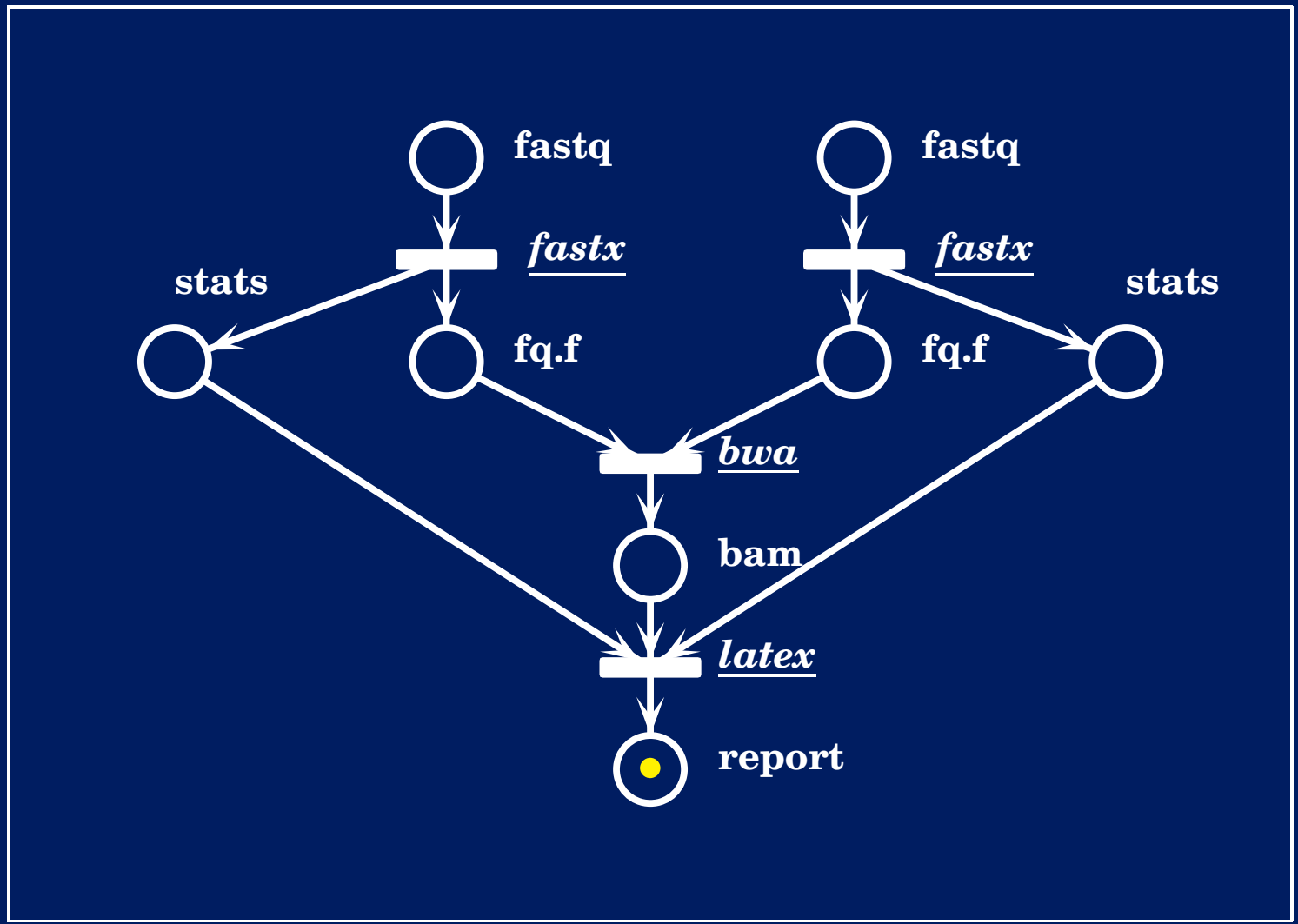

Figure 7: A parallel workflow

Practical solution

Advantages of make

Control flow:

- Implicit.
- The target is removed if the recipe returns an error.
- The build process can resume from any point.

Practical solution

Advantages of make

Control flow:

- Implicit.
- The target is removed if the recipe returns an error.
- The build process can resume from any point.

Plus:

- Portable.
  - `make`, `qmake`, `nmake`, ...
- Modular.
  - Multiple (overlapping) pipelines can be combined.
- Test without executing (`make -n`).

GAPSS3

Exome capture pipeline

First version:

- Frame written in **`bash`**.
- Custom scripts for job handling.
- Complicated to redeploy interrupted analysis.
- Modularity is limited.

https://humgenprojects.lumc.nl/trac/GAPSS3

GAPSS3

Exome capture pipeline

First version:

- Frame written in **`bash`**.
- Custom scripts for job handling.
- Complicated to redeploy interrupted analysis.
- Modularity is limited.

Current version:

- Reduced code from ±2000 to ±400 lines.
- Optimised cluster usage.
- Shorter analysis time.

https://humgenprojects.lumc.nl/trac/GAPSS3

GAPSS3

The GAPSS3 workflow

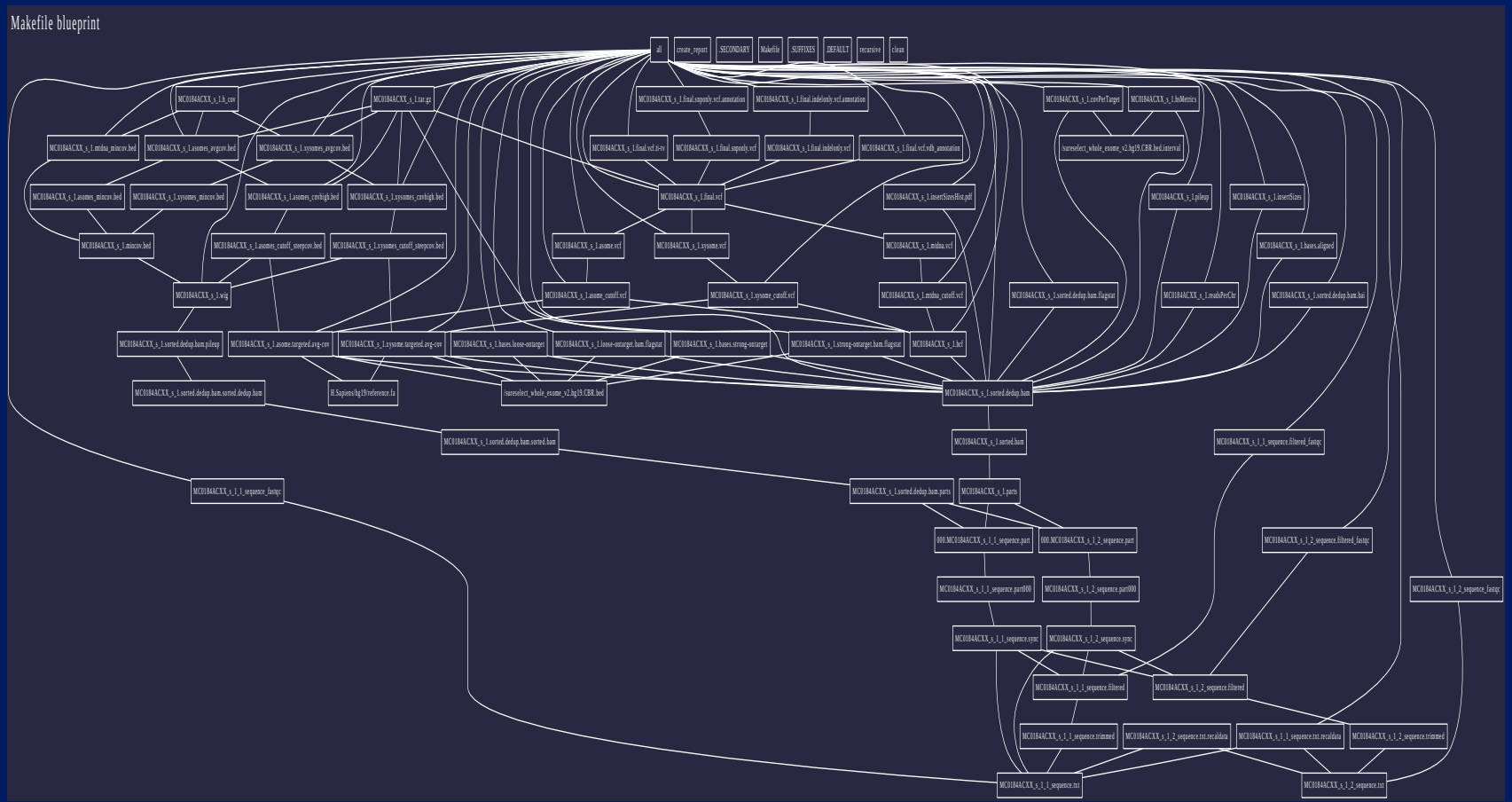

Figure 8: GAPSS3

GAPSS3

The GAPSS3 workflow

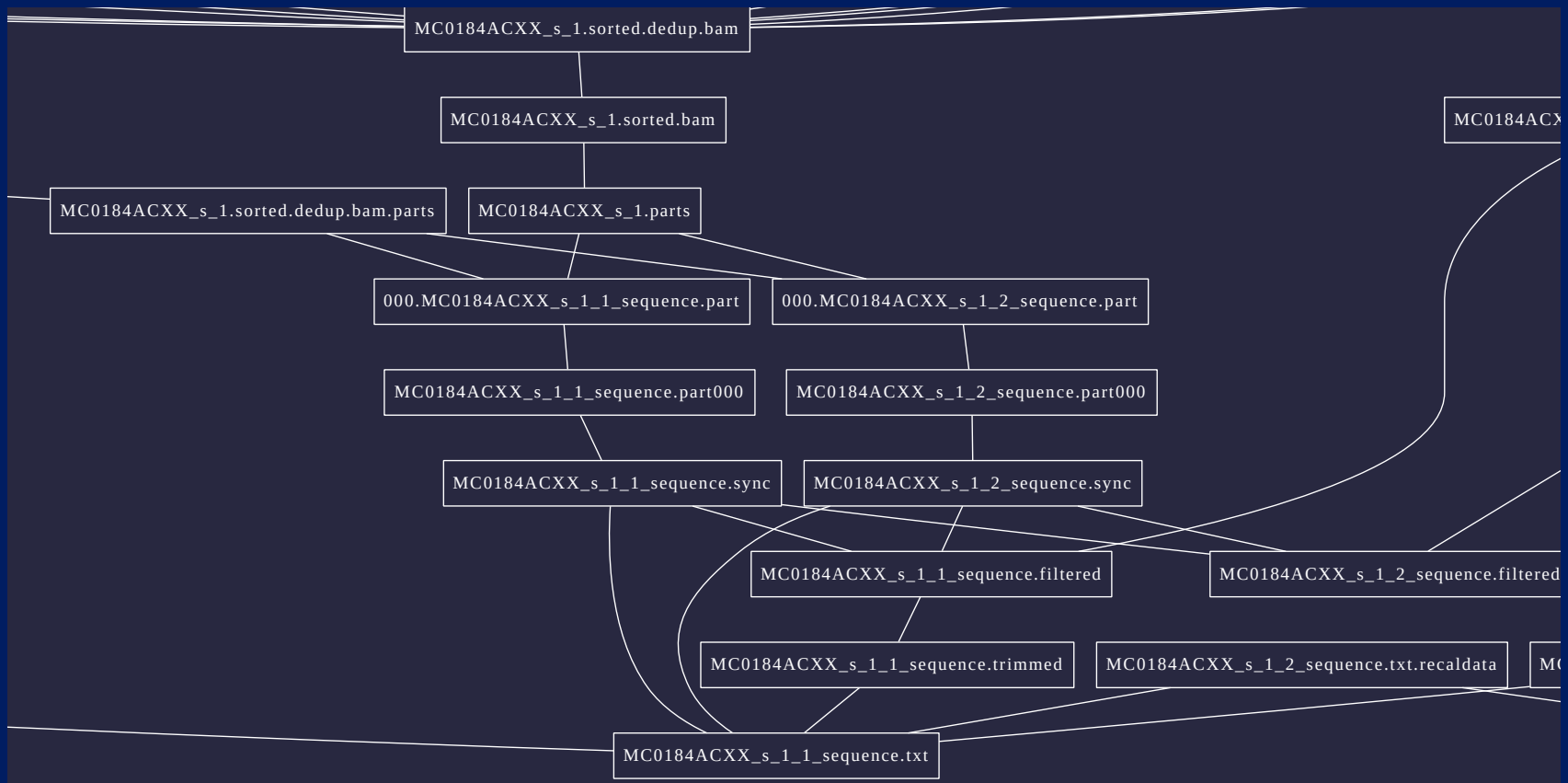

Figure 9: Part of GAPSS3

Questions?

Acknowledgements:

Michiel van Galen
Martijn Vermaat
Johan den Dunnen

https://humgenprojects.lumc.nl/trac/shark/wiki/Makefile