LEIDEN UNIVERSITY MEDICAL CENTER

Bioinformatics in molecular genetics

Jeroen F. J. Laros

Leiden Genome Technology Center

Department of Human Genetics

Center for Human and Clinical Genetics

Introduction

About me

The last few years:

- Master Computer Science.
- Ph.D. Mathematics and Natural Sciences.
- Post-doctoral researcher LUMC.
- Senior researcher.
    - Coördinator Bioinformatics LGTC.

Introduction

About me

The last few years:

- Master Computer Science.
- Ph.D. Mathematics and Natural Sciences.
- Post-doctoral researcher LUMC.
- Senior researcher.
    - Coördinator Bioinformatics LGTC.

Currently active in:

- Variant databases.
- Formal descriptions.
- Next Generation Sequencing.
- Metagenomics.
- Forensics.

Introduction

About me

The last few years:

- Master Computer Science.
- Ph.D. Mathematics and Natural Sciences.
- Post-doctoral researcher LUMC.
- Senior researcher.
    - Coördinator Bioinformatics LGTC.

Currently active in:

- Variant databases.
- Formal descriptions.
- Next Generation Sequencing.
- Metagenomics.
- Forensics.

Introduction

Next generation sequencing

Figure 1 : HiSeq 2000.

Figure 2 : Ion proton.

Next Generation Sequencers.

Introduction

Next generation sequencing data

```
1  @SGGPP:4:101
2  TTCGGGGGCTGGCAAATCCACTTCCGTGACACGCTACCATTCGCTGGTGGT
3  +
4  −'+4589,53330−0&07+03:54/2362−+.488587>@/25440++0(+
5  @SGGPP:4:102
6  CGGTAAACCACCCTGCTGACGGAACCCTAATGCGCCTGAAAGACAGCGTTC
7  +
8  34/−−0'+.000(.55:;:99(0(+2(22(0316;185;;0;:<<>=AA59
9  @SGGPP:4:106
10 TCGTTAACGACTTTGTTCGCCACCGCAACCGCCTGTTTCGGGTCACAGGCA
11 +
12 09875;5?<;?@A4?B:BBB<AA>CCC>C>BB0.−>=0488+3444:@5@<
13 @SGGPP:4:112
14 TTGATGAATATATTATTTCAGGGAATAATTATGACACCTTTAGAACGCATT
15 +
16 70<<@::5:<;==7;>>/79<:.:494.8(,,8:753/5@5??C>B???B7
```

Listing 1 : A FastQ file.

Introduction

Data analysis

Very diverse.

Introduction

Data analysis

Very diverse.

Align to a reference genome (resequencing):

- Variant detection.
- Phylogenetic reconstruction.

Introduction

Data analysis

Very diverse.

Align to a reference genome (resequencing):

- Variant detection.
- Phylogenetic reconstruction.

Or to multiple references:

- Antibiotic resistance testing.

Introduction

Data analysis

Very diverse.

Align to a reference genome (resequencing):

- Variant detection.
- Phylogenetic reconstruction.

Or to multiple references:

- Antibiotic resistance testing.

De novo assembly:

- First step to make a reference genome.
- Finding large rearrangements.

Introduction

Data analysis

Very diverse.

Align to a reference genome (resequencing):

- Variant detection.
- Phylogenetic reconstruction.

Or to multiple references:

- Antibiotic resistance testing.

De novo assembly:

- First step to make a reference genome.
- Finding large rearrangements.

Resequencing

Data analysis

Resequencing pipelines can roughly be divided in five steps.

Resequencing

Data analysis

Resequencing pipelines can roughly be divided in five steps.

1. Pre-alignment.
    - Quality control.
    - Data cleaning.

Resequencing

Data analysis

Resequencing pipelines can roughly be divided in five steps.

1. Pre-alignment.
    - Quality control.
    - Data cleaning.
2. Alignment.
    - Post-alignment quality control.

Resequencing

Data analysis

Resequencing pipelines can roughly be divided in five steps.

1. Pre-alignment.
    - Quality control.
    - Data cleaning.
2. Alignment.
    - Post-alignment quality control.
3. Variant calling.

Resequencing

Data analysis

Resequencing pipelines can roughly be divided in five steps.

1. Pre-alignment.
    - Quality control.
    - Data cleaning.
2. Alignment.
    - Post-alignment quality control.
3. Variant calling.
4. Filtering.
    - Post-variant calling quality control.

Resequencing

Data analysis

Resequencing pipelines can roughly be divided in five steps.

1. Pre-alignment.
    - Quality control.
    - Data cleaning.
2. Alignment.
    - Post-alignment quality control.
3. Variant calling.
4. Filtering.
    - Post-variant calling quality control.
5. Annotation.

Resequencing

Data analysis

Resequencing pipelines can roughly be divided in five steps.

1. Pre-alignment.
    - Quality control.
    - Data cleaning.
2. Alignment.
    - Post-alignment quality control.
3. Variant calling.
4. Filtering.
    - Post-variant calling quality control.
5. Annotation.

We do all of these analyses on Linux machines.

Resequencing

Linux

Figure 3 : Linux desktops.

User friendly.

- Safe.
- Low memory requirements.
- Low CPU requirements.
- Low disk requirements.
- Can be installed alongside Windows or OSX.

Resequencing

Linux

But we prefer the *command line*.

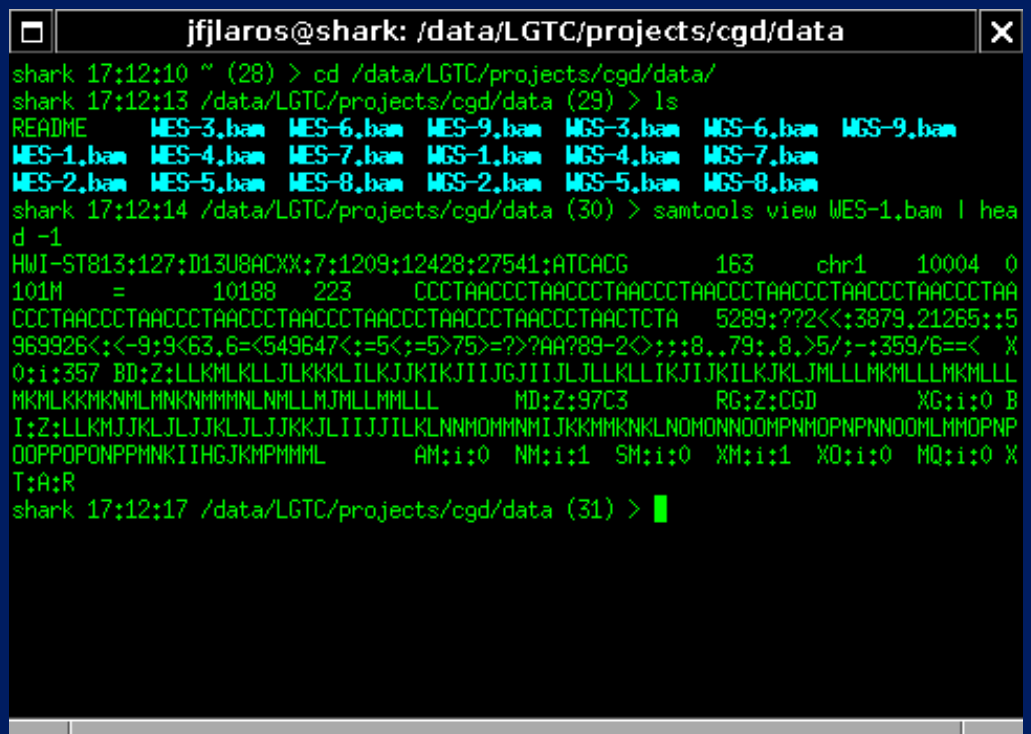

Figure 4 : Text based console.

Resequencing

Why Linux?

A popular operating system for scientific developers.

- Computer science.

Resequencing

Why Linux?

A popular operating system for scientific developers.

- Computer science.
- Physics.
- Astronomy.
- ...

Resequencing

Why Linux?

A popular operating system for scientific developers.

- Computer science.
- Physics.
- Astronomy.
- ...

It is free.

- The operating system is free.
- Almost all software is free.
    - Development software.

Resequencing

Why Linux?

A popular operating system for scientific developers.

- Computer science.
- Physics.
- Astronomy.
- ...

It is free.

- The operating system is free.
- Almost all software is free.
    - Development software.

Not only free as in costless, but also *open source*.

Resequencing

Why the command line?

```
1  bwa aln ref.fa s_1.fq > s_1.sai
2  bwa aln ref.fa s_2.fq > s_2.sai
3  bwa sampe ref.fa s_1.sai s_2.sai s_1.fq s2_fq > s.sam
4  samtools view -bt ref.fa -o s.bam s.sam
5  samtools sort s.bam s.sorted
6  samtools mpileup -f ref.fa s.sorted.bam > s.pileup
```

Listing 2 : Bash script snippet.

If the command line works, it works in a script.

- Less chance of errors.
- Less work.
- Reproducible.

Resequencing

Command line scripting

Three day course:

- Operating system basics.
- Installing and updating software.
- Connecting to other machines.
- File systems and the command line.
- Introduction to AWK.
- Regular expressions.
- Associative arrays.

M. Palmblad, R. Marissen, M. van Galen

https://humgenprojects.lumc.nl/trac/humgenprojects/wiki/scripting

Antibiotic resistance testing

The E.coli

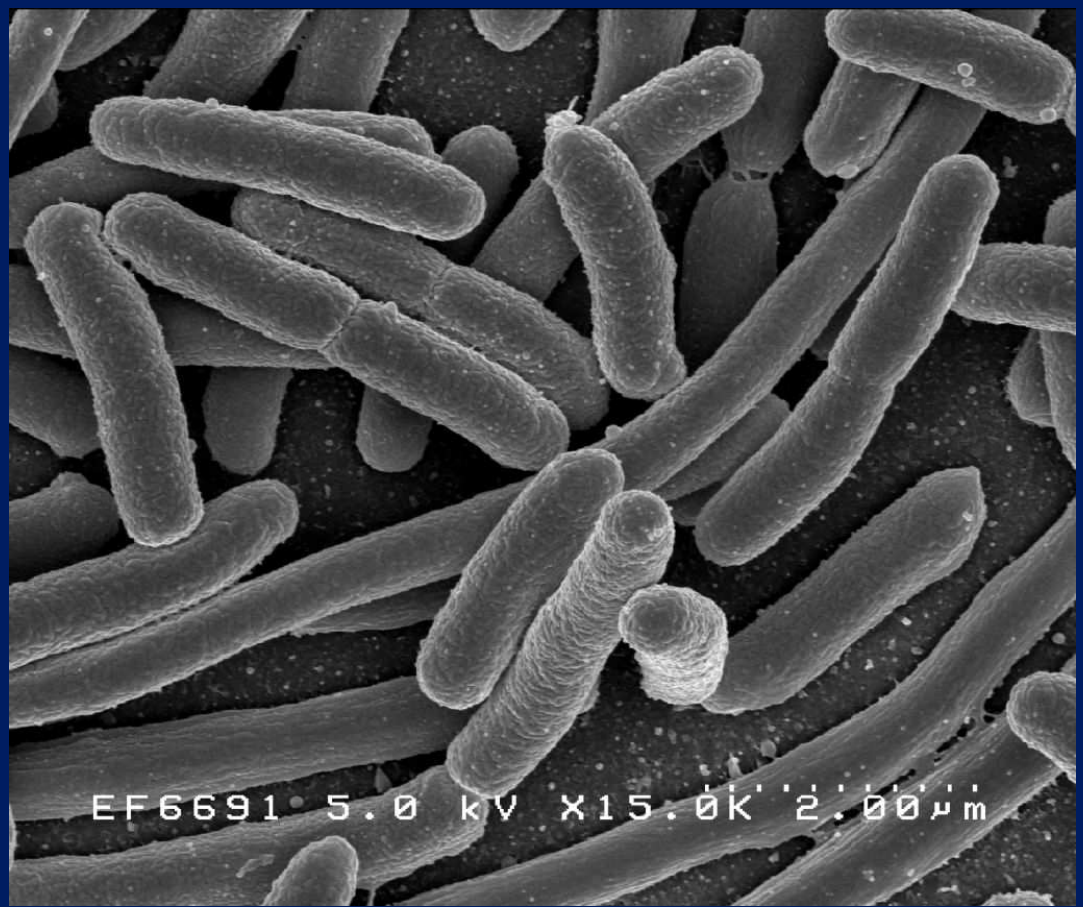

Figure 5 : Escherichia coli.

Antibiotic resistance testing

Some figures on the E. coli

Genome published in 1997.

- Genome size 4.6×10^6 basepairs.
- 4,288 genes in the assembly.
  - 86 tRNA genes.
- 2,584 operons in the assembly.
  - 7 rRNA operons.

Antibiotic resistance testing

Some figures on the E. coli

Genome published in 1997.

- Genome size 4.6×10^6 basepairs.
- 4,288 genes in the assembly.
    - 86 tRNA genes.
- 2,584 operons in the assembly.
    - 7 rRNA operons.

However, per individual strain:

- Between 4,000 and 5,500 genes.
- 16,000 genes in total (pangenome).

Antibiotic resistance testing

Some figures on the E. coli

Genome published in 1997.

- Genome size 4.6×10^6 basepairs.
- 4,288 genes in the assembly.
  - 86 tRNA genes.
- 2,584 operons in the assembly.
  - 7 rRNA operons.

However, per individual strain:

- Between 4,000 and 5,500 genes.
- 16,000 genes in total (pangenome).

Very diverse, only 20% of the genome is shared between all strains.

Antibiotic resistance testing

Plasmids

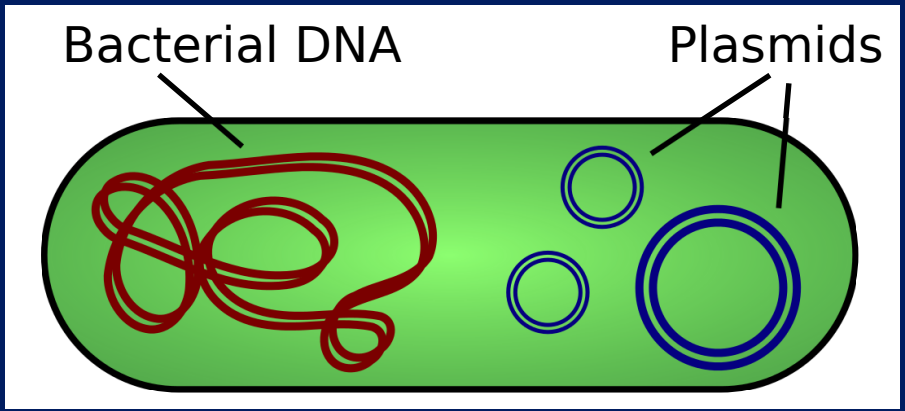

Figure 6 : Schematic overview of a cell containing plasmids.

Antibiotic resistance testing

Plasmids

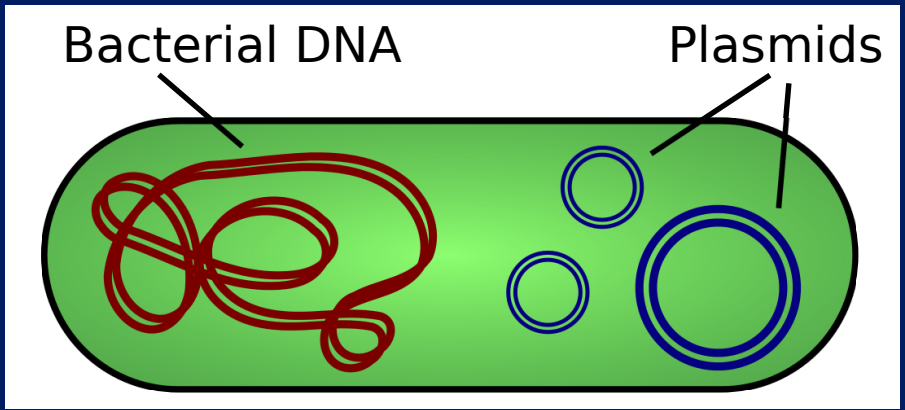

Figure 6 : Schematic overview of a cell containing plasmids.

Plasmids are small DNA molecules.

- Separate and independent from the chromosome.
- Can be transferred to other species.
- Size between 1×10^3 and 1×10^6 basepairs.
- Copy number between 1 and 1,000.
- Variable between strains and individuals.

Antibiotic resistance testing

Figure 7 : Classical antibiotic resistance test.

Antibiotic resistance testing

An alternative approach

Sequence everything.

Antibiotic resistance testing

An alternative approach

Sequence everything.

But there are some technical issues.

- Many genes look alike.
- Many plasmid share genes.

Antibiotic resistance testing

An alternative approach

Sequence everything.

But there are some technical issues.

- Many genes look alike.
- Many plasmid share genes.

So we map all our reads to all references and see which ones fit best.

This takes quite some time.

Parallel computing

Clusters

Figure 8 : Dell M610 blade server.

Parallel computing

Clusters

Massive parallel computing.

- A large number of computers working together.
- Analyse lots of samples at the same time.
- Sometimes a way to reduce memory requirements (if the problem permits it).
- Very suitable for NGS, especially alignment.

Parallel computing

Clusters

Massive parallel computing.

- A large number of computers working together.
- Analyse lots of samples at the same time.
- Sometimes a way to reduce memory requirements (if the problem permits it).
- Very suitable for NGS, especially alignment.

Cons:

- Not all problems are suitable for parallel computation.
- Programs must be adjusted to make use of a cluster.
    - Chop the problem up in parts / combine the results.

Parallel computing

Pipelines

Classical pipeline:

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

Parallel computing

Pipelines

Classical pipeline:

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

Drawbacks:

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

Parallel computing

Petri nets

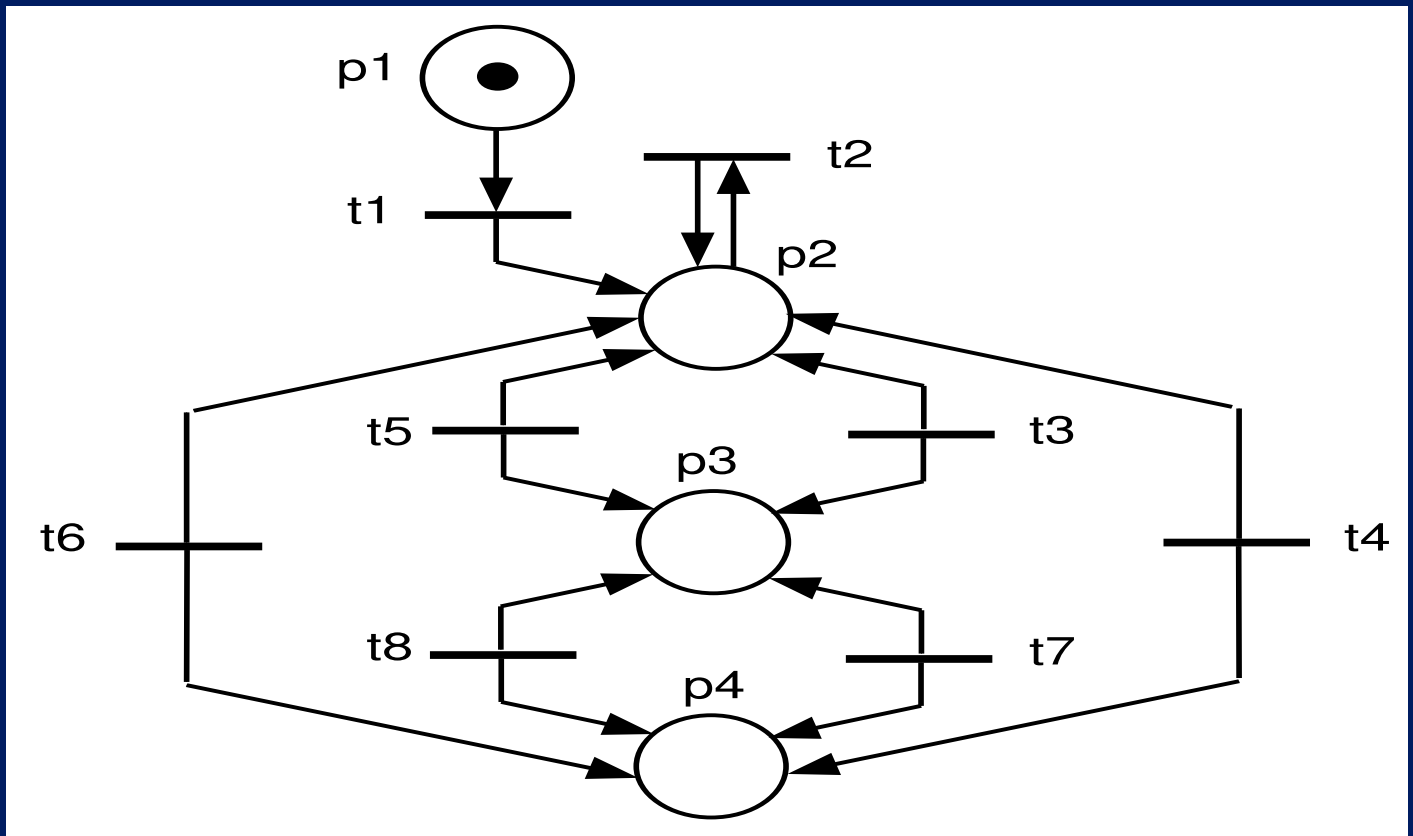

Figure 9 : A model for parallelism.

Parallel computing

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

Parallel computing

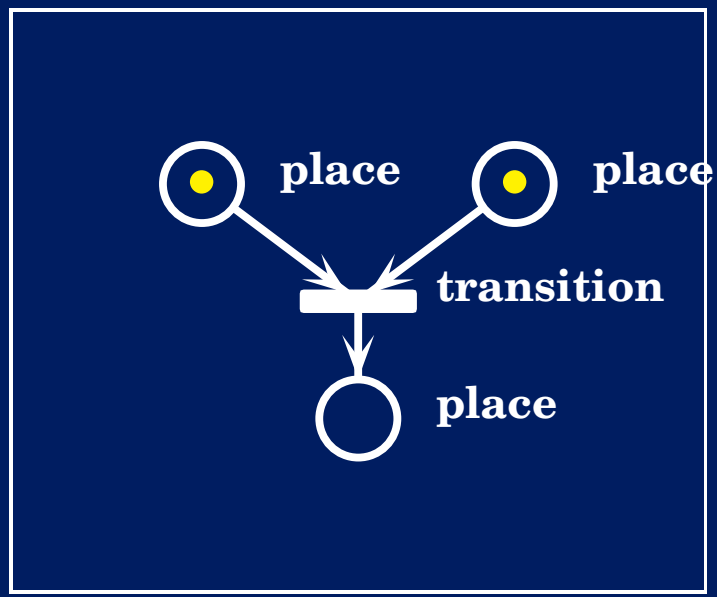

Figure 10 : A simple Petri net.

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.

Parallel computing

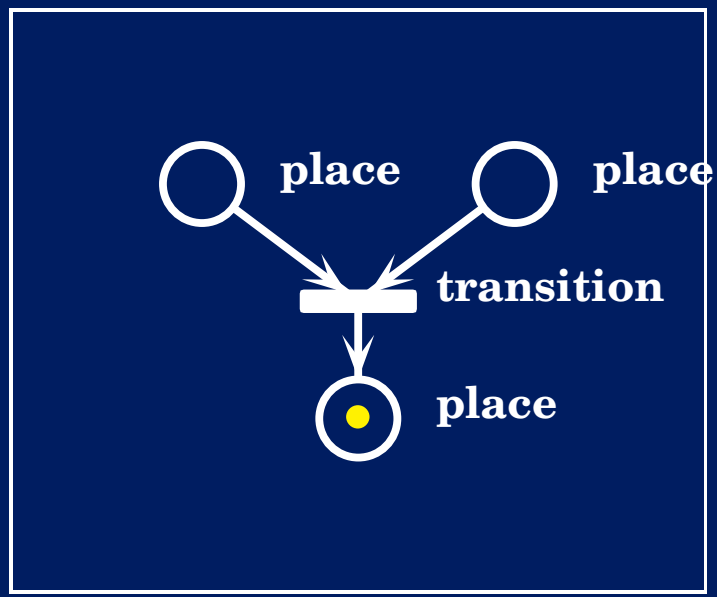

Figure 10 : A simple Petri net.

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.

Parallel computing

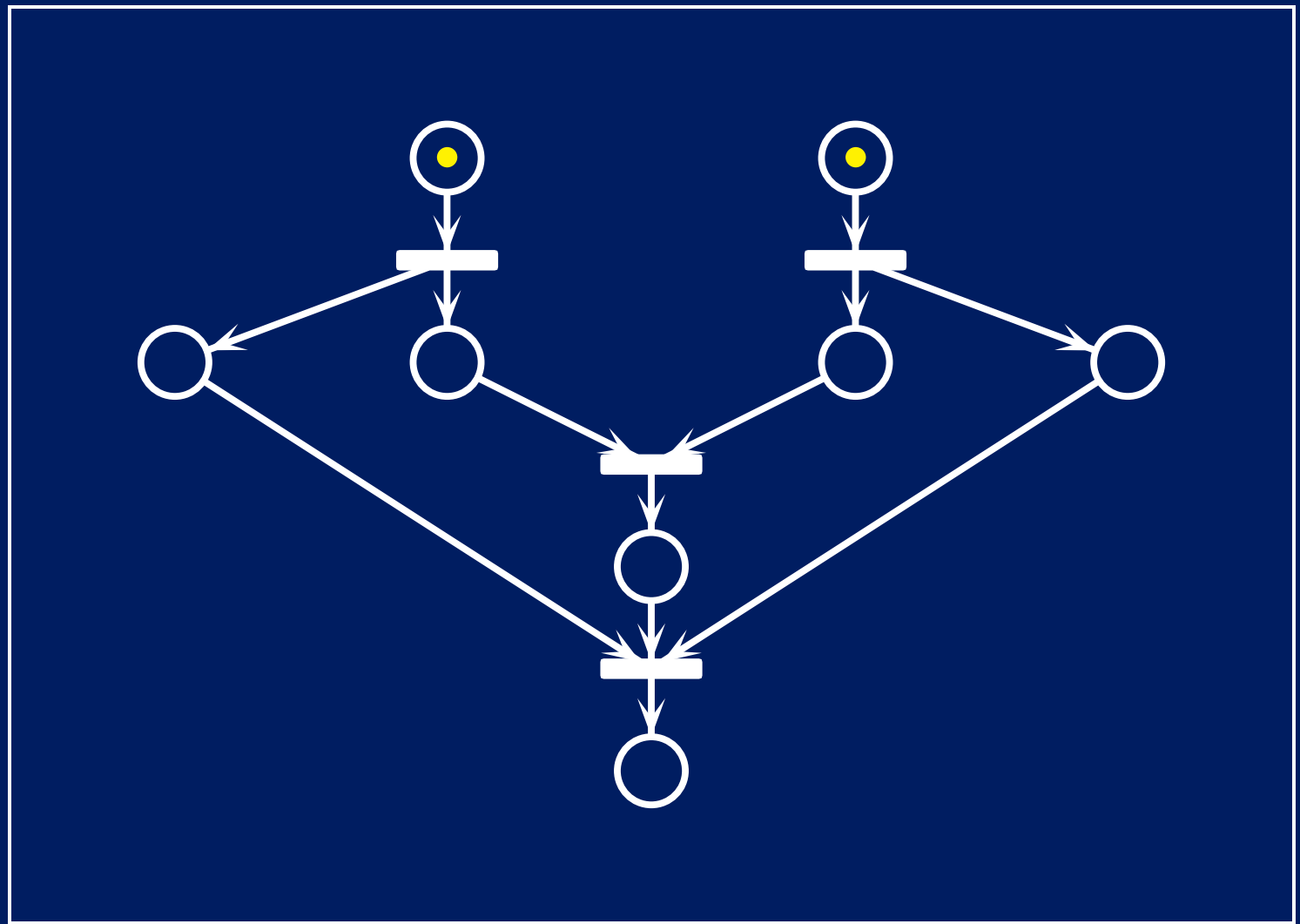

Figure 11 : A more complicated Petri net.

Parallel computing

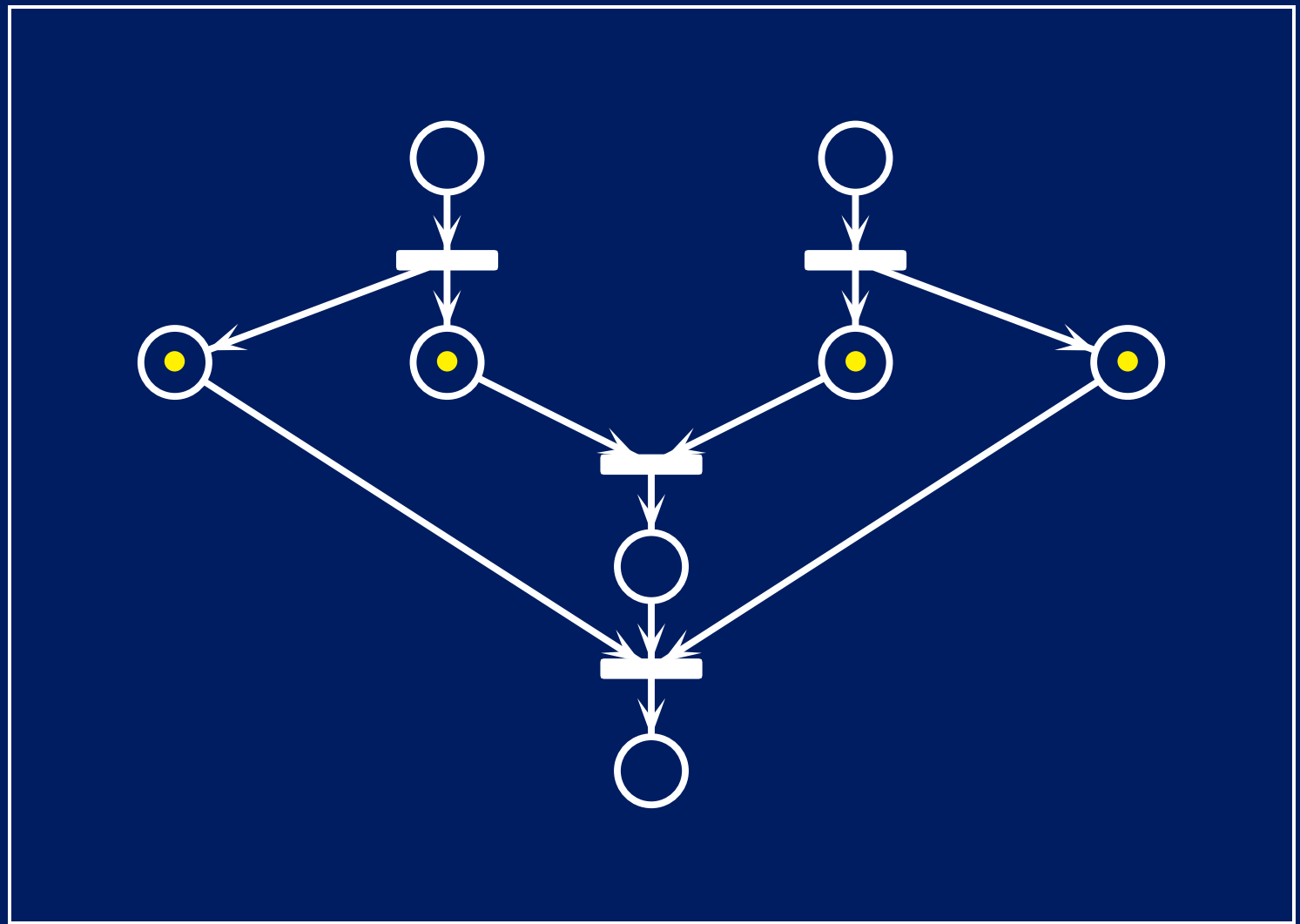

Figure 11 : A more complicated Petri net.

Parallel computing

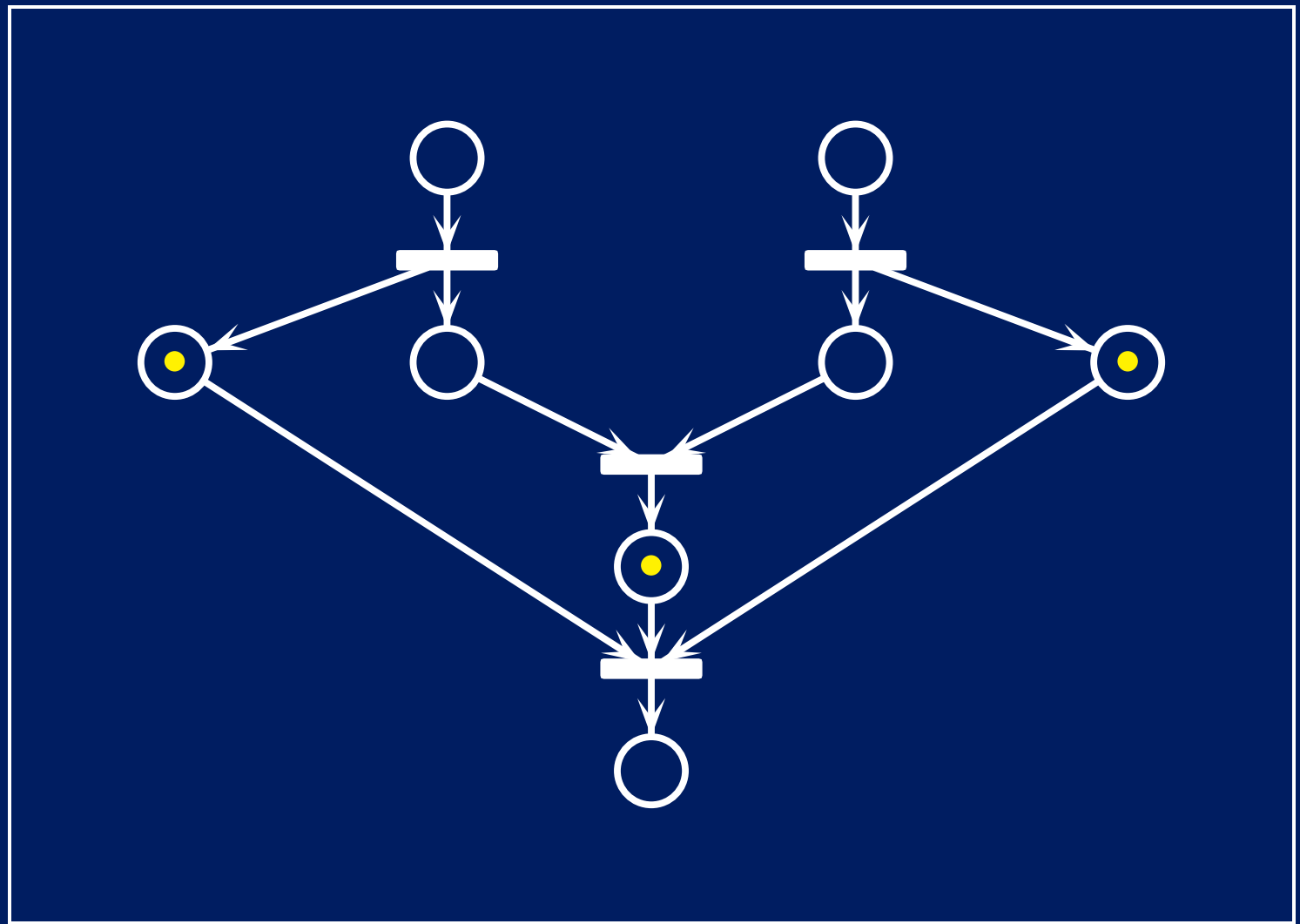

Figure 11 : A more complicated Petri net.

Parallel computing

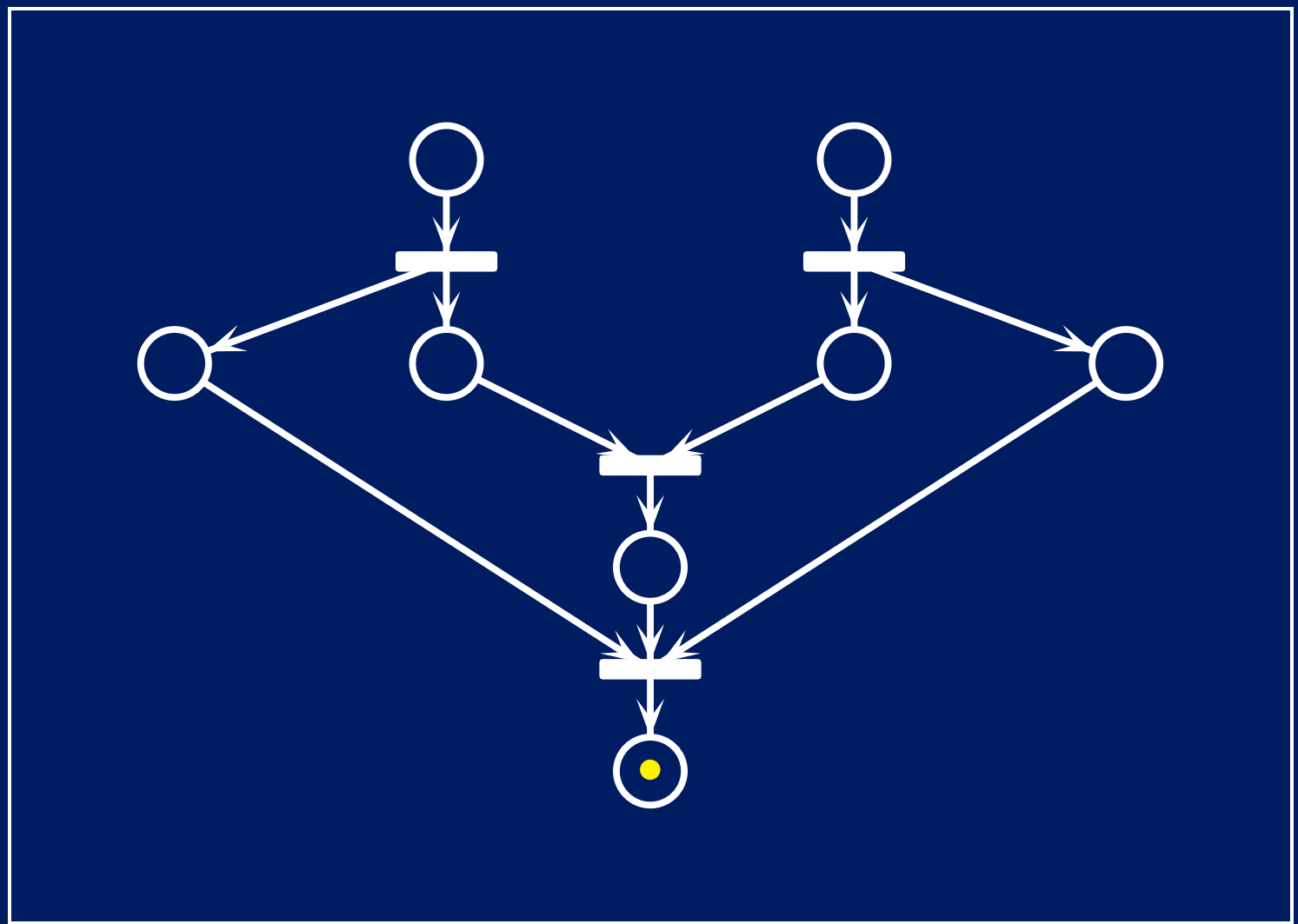

Figure 11 : A more complicated Petri net.

Parallel computing

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.

Parallel computing

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)

Parallel computing

The anatomy of a Makefile

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

Listing 3 : 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

Parallel computing

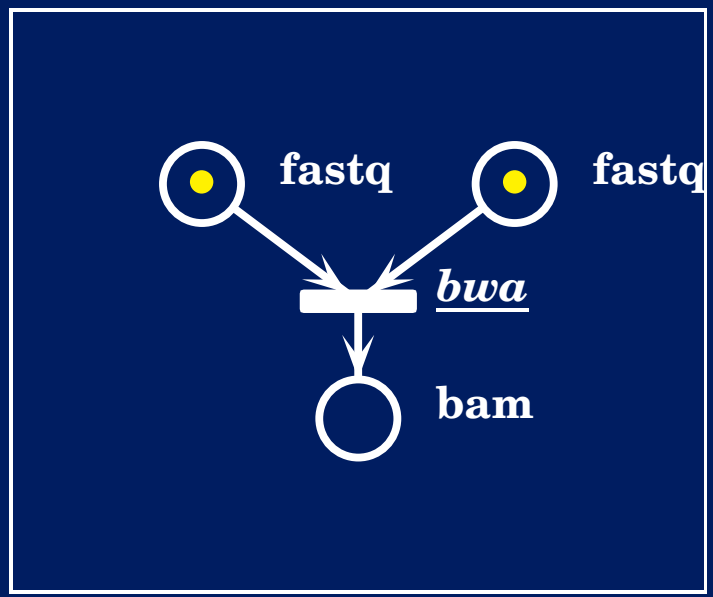

Figure 12 : A simple workflow.

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

Listing 4 : Makefile snippet.

Parallel computing

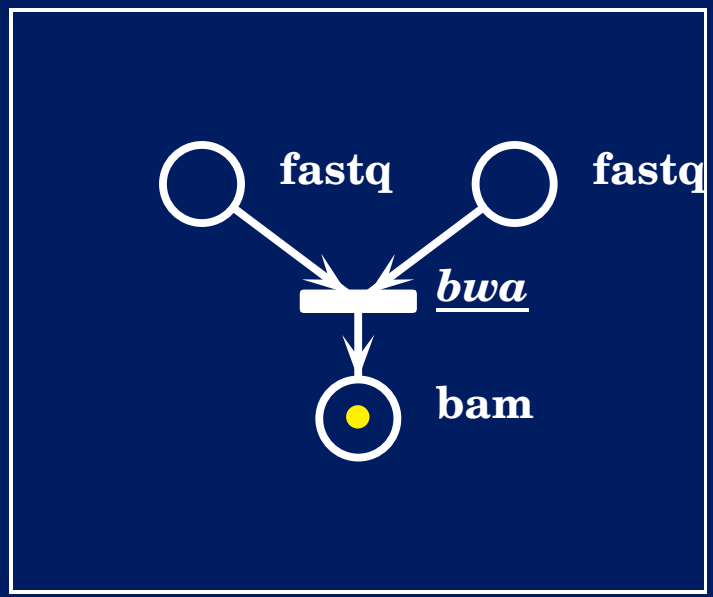

Figure 12 : A simple workflow.

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

Listing 4 : Makefile snippet.

Parallel computing

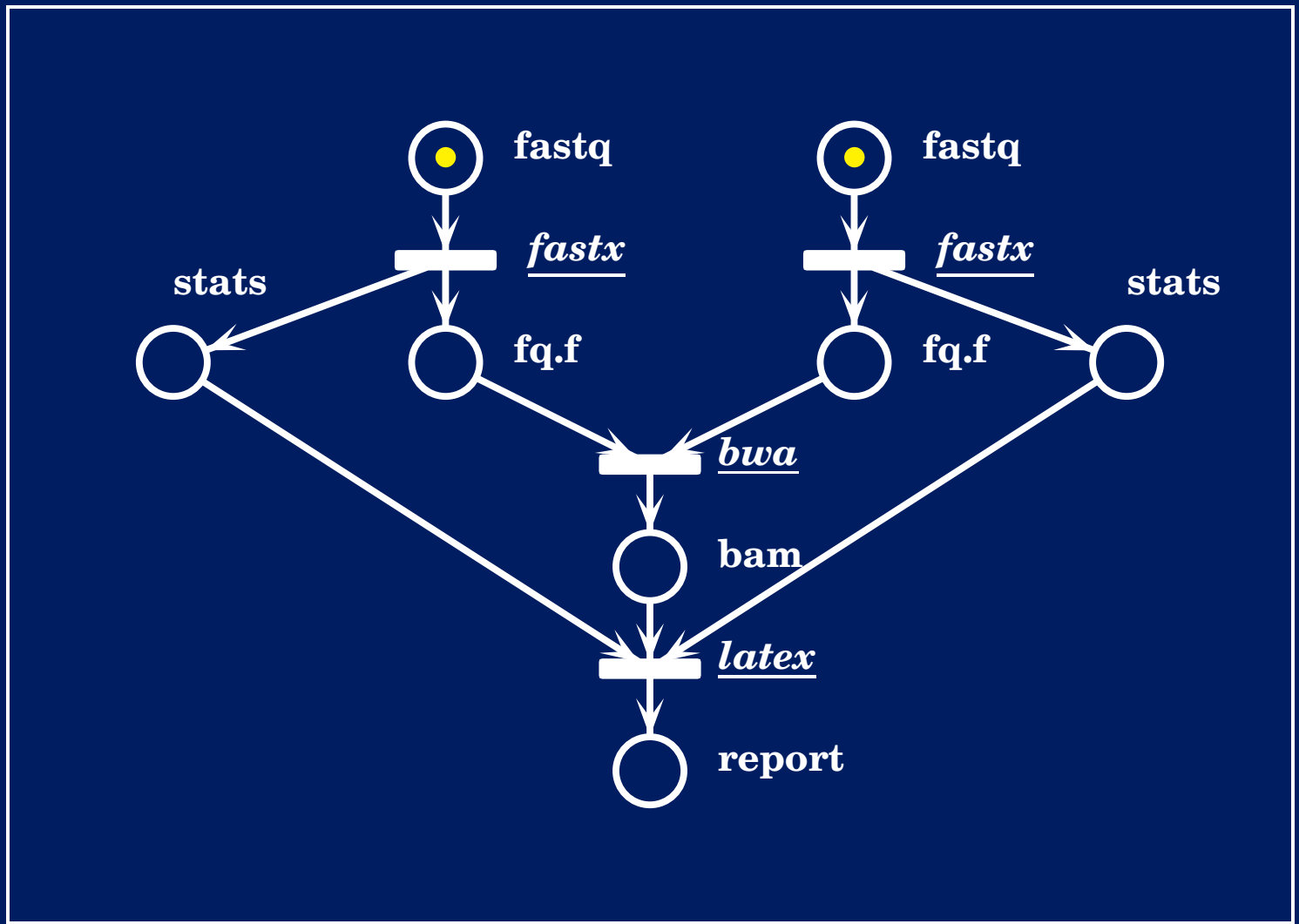

Figure 13 : A parallel workflow.

Parallel computing

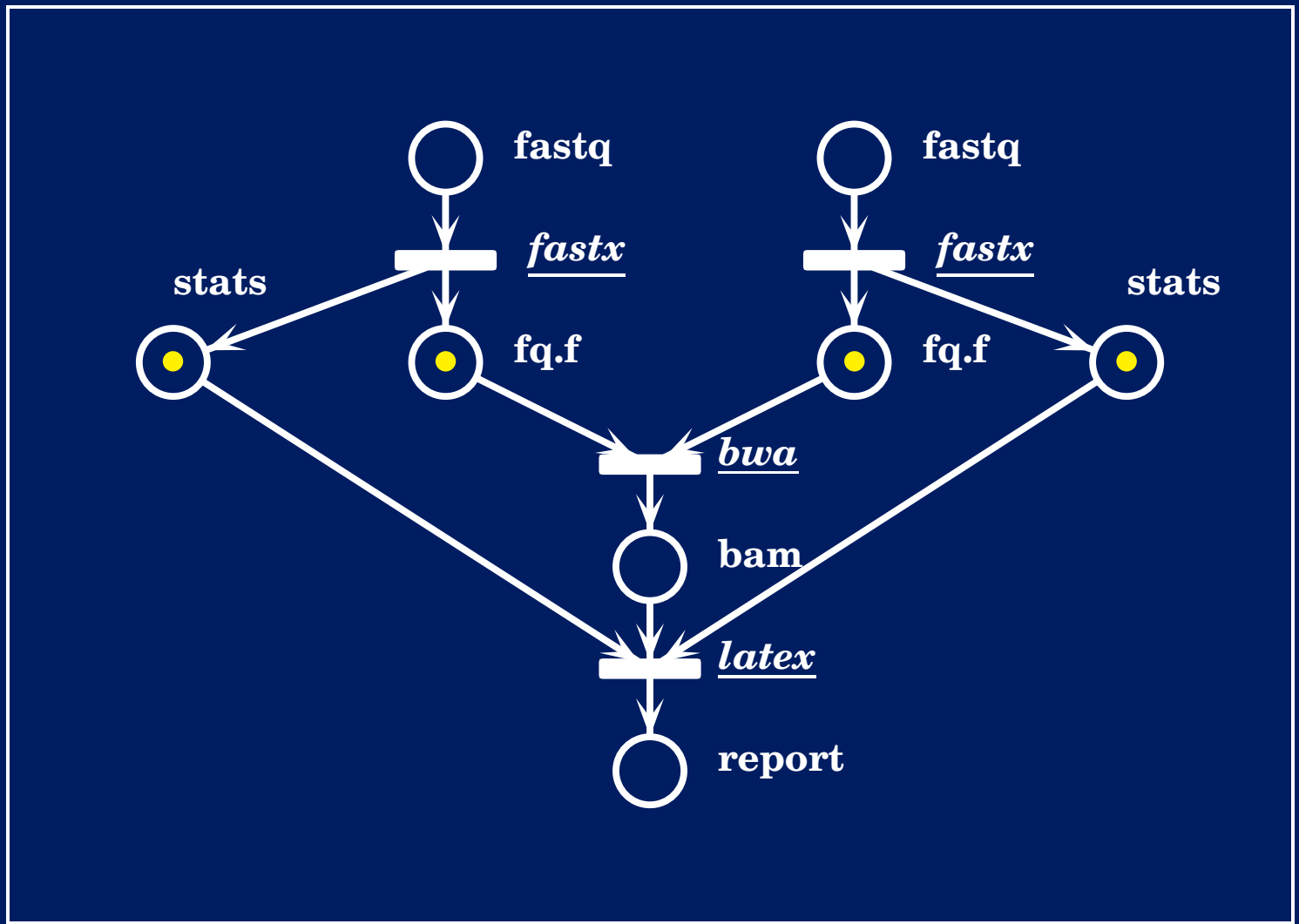

Figure 13 : A parallel workflow.

Parallel computing

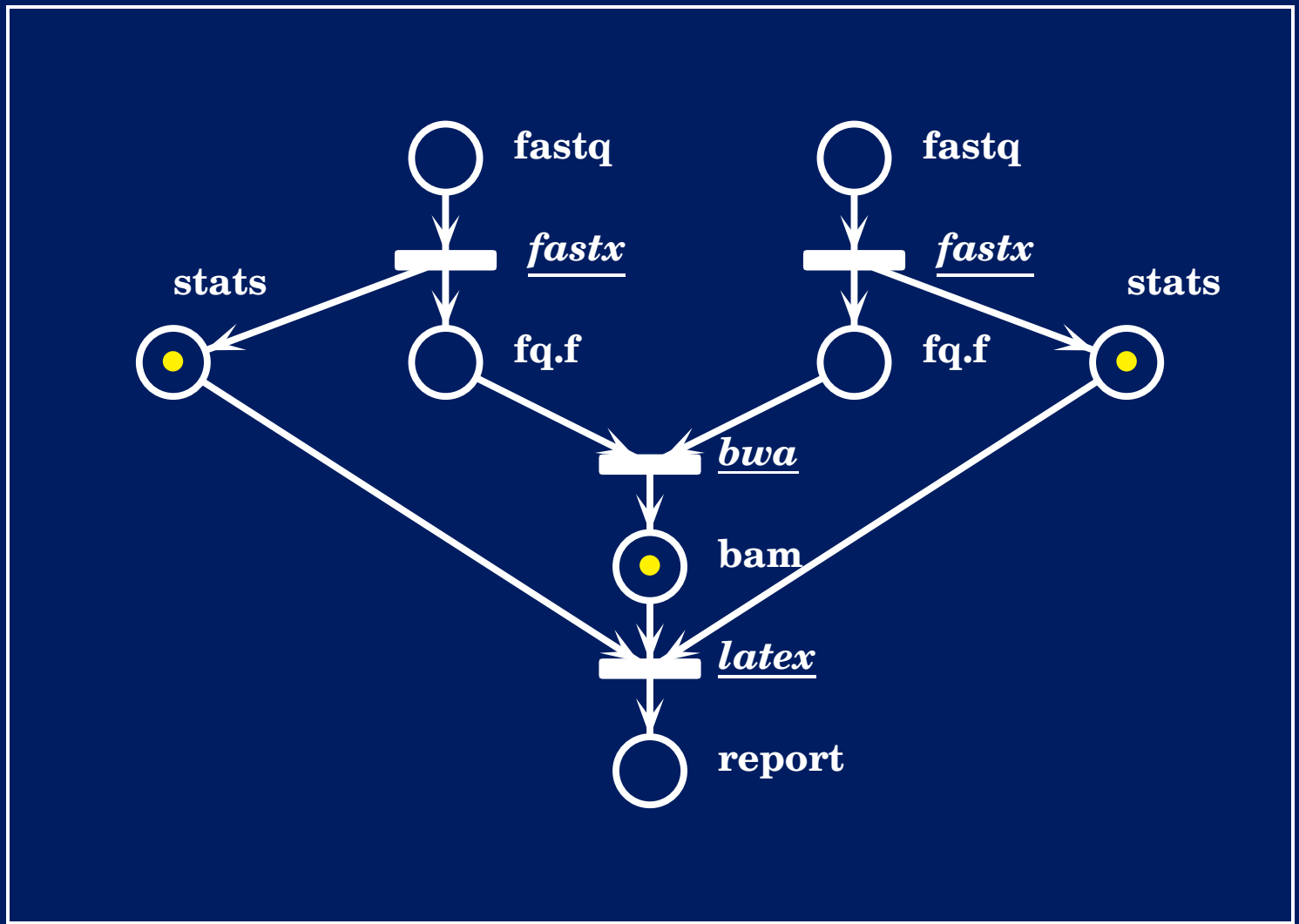

Figure 13 : A parallel workflow.

Parallel computing

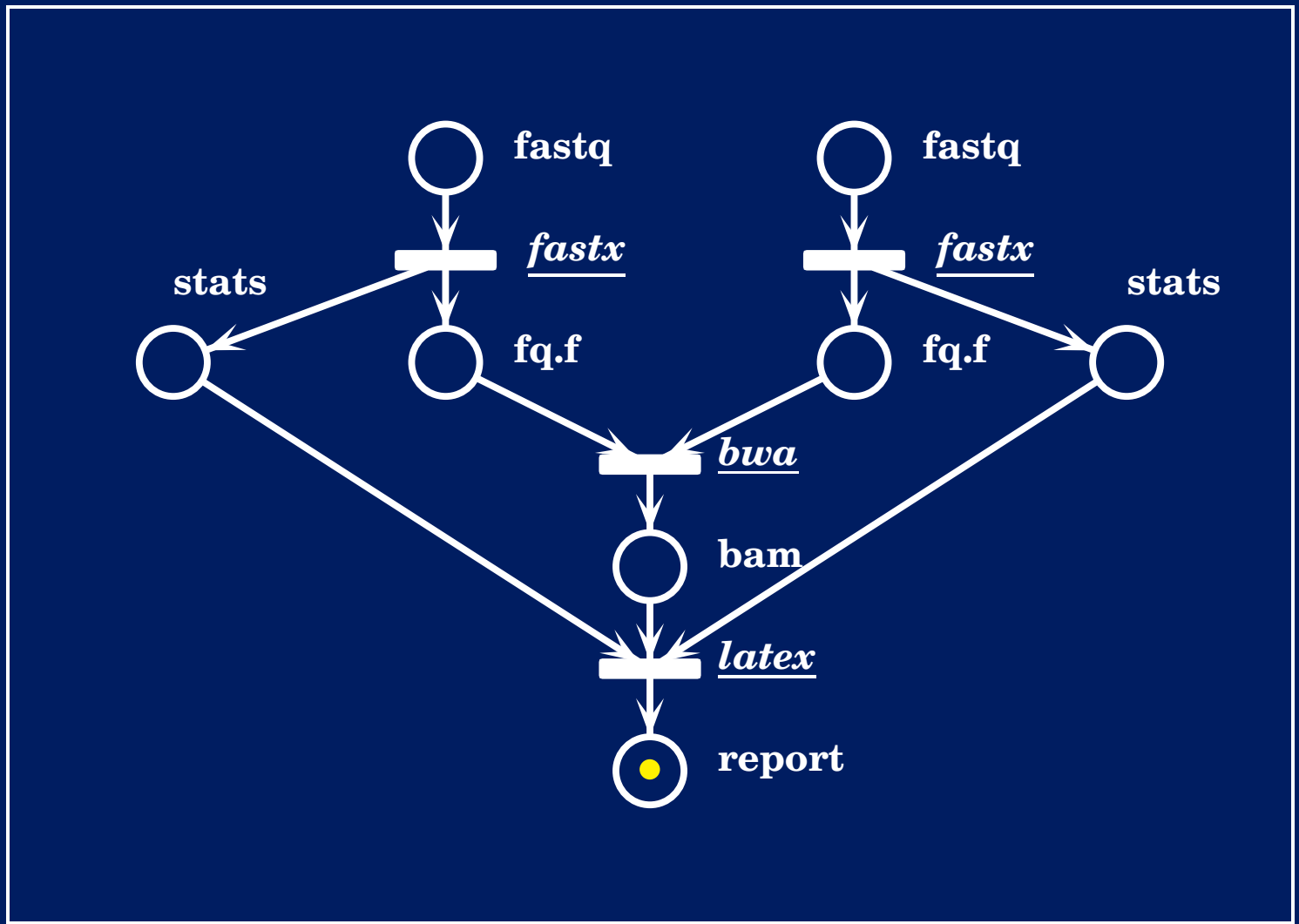

Figure 13 : A parallel workflow.

Parallel computing

Advantages of make

Control flow:

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

Parallel computing

Advantages of make

Control flow:

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

Plus:

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

Parallel computing

Example Makefile

The same pipeline as shown in Listing 2.

```
1  %.sai: %.fq
2      $(BWA) aln $(REFERENCE) $< > $@
3
4  %.sam: %_1.sai %_2.sai %_1.fq %_2.fq
5      $(BWA) sampe $(REFERENCE) $^ > $@
6
7  %.bam: %.sam
8      $(SAMTOOLS) view -bt $(REFERENCE) -o $@ $<
9
10 %.sorted.bam: %.bam
11     $(SAMTOOLS) sort $< $*.sorted
12
13 %.mpileup: %.sorted.bam
14     $(SAMTOOLS) mpileup -f $(REFERENCE) $< > $@
```

Listing 5 : Makefile snippet.

Parallel computing

Introduction course clusters

Half day course.

- Connecting to a cluster.
- Using the Sun Grid Engine.
- Do's and don'ts.
- Makefiles.

M. Villerius

Education

Education in the last year

Courses given:

- Introduction Linux and command line scripting.
- Introduction to Git.
- Next generation sequencing (NGS) data analysis.
- Python programming.
- Introduction to NGS data analysis.

Education

Education in the last year

Courses given:

- Introduction Linux and command line scripting.
- Introduction to Git.
- Next generation sequencing (NGS) data analysis.
- Python programming.
- Introduction to NGS data analysis.

Invited lectures.

- RNA-seq using Galaxy.
- Variant Calling.
- GAPSS3 using Galaxy.
- Functional annotation of metagenomes.
- Clusters and parallelising workflows.

Education

Git

Everyone in the Bioinformatics field:

- Software development.
- Project management.
- Collaboration.

Topics:

- Git Basics
- Branching
- Remotes
- Project skeleton / git annex

M. Vermaat

https://humgenprojects.lumc.nl/trac/humgenprojects/wiki/git

Education

Next Generation Sequencing data analysis

Three day course:

- Discussion of different platforms and produced data.
  - Illumina, Roche, ABI, Ion Torrent, etc.

J.M. Boer, J.T. den Dunnen, W. van IJcken, C. van Gelder

Education

Next Generation Sequencing data analysis

Three day course:

- Discussion of different platforms and produced data.
  - Illumina, Roche, ABI, Ion Torrent, etc.
- General applications.
  - Resequencing, structural variation, de novo assembly, visualisation, pipelines, etc.

J.M. Boer, J.T. den Dunnen, W. van IJcken, C. van Gelder

Education

Next Generation Sequencing data analysis

Three day course:

- Discussion of different platforms and produced data.
  - Illumina, Roche, ABI, Ion Torrent, etc.
- General applications.
  - Resequencing, structural variation, de novo assembly, visualisation, pipelines, etc.
- Specific applications.
  - QC, statistics, expression, ChIP-seq, metagenomics, etc.

J.M. Boer, J.T. den Dunnen, W. van IJcken, C. van Gelder

Education

Next Generation Sequencing data analysis

Three day course:

- Discussion of different platforms and produced data.
  - Illumina, Roche, ABI, Ion Torrent, etc.
- General applications.
  - Resequencing, structural variation, de novo assembly, visualisation, pipelines, etc.
- Specific applications.
  - QC, statistics, expression, ChIP-seq, metagenomics, etc.
- Practical sessions.
  - Galaxy, NextGENe, CLCbio.

Room for 60 people, always full.

J.M. Boer, J.T. den Dunnen, W. van IJcken, C. van Gelder

Education

NGS Introduction Course

Full day to get people acquainted with NGS data analysis.

M. van Galen

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

Education

NGS Introduction Course

Full day to get people acquainted with NGS data analysis.

- Sort version of the Linux course.
- Command line and NGS tools.
- Connecting to remote machines (clusters).
- NGS pipelines.
- Using Galaxy.

Given yearly:

- Avans Hogeschool Breda.

M. van Galen

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

Education

Programming in Python

Four day course.

- Python basics.
- Standard data structures.
- Working with NumPy arrays.
- Plotting with matplotlib.
- Object-oriented programming.
- The Biopython library.

M. Vermaat, W. Arindrato, Z. Tatum

https://humgenprojects.lumc.nl/trac/programming-course

Questions?

Acknowledgements:

Michiel van Galen
Martijn Vermaat
Wibowo Arindrato
Zuotian Tatum
Magnus Palmblad
Hailiang (Leon) Mei
Michel Villerius
Rob Marissen
Judith Boer
Wilfred van IJcken
Celia van Gelder
Johan den Dunnen