TGAC
The Genome Analysis Centre

RAMPART: an automated de novo assembly pipeline

Dr. Daniel Mapleson
Scientific Programmer
daniel.mapleson@tgac.ac.uk

Our location

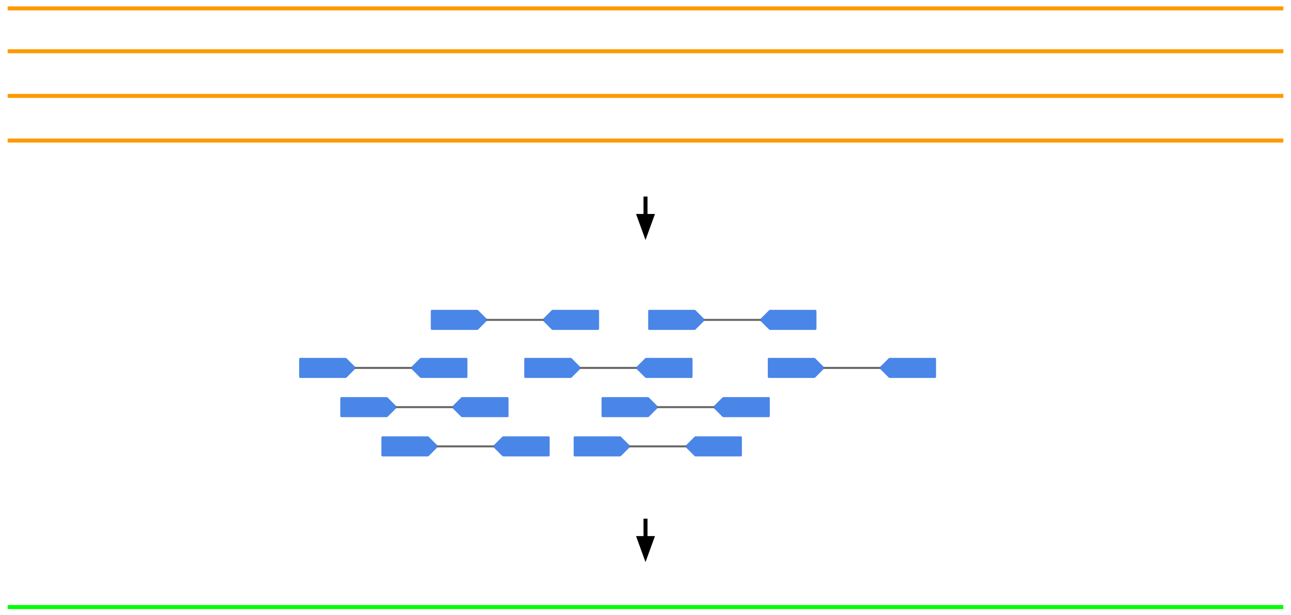
De novo Genome Assembly

(with short reads from NGS devices)

Original DNA molecules

DNA sheared into shorter fragments of equal length, these are then sequenced

Assembled sequence



"the equivalent of trying to put together a multi-million piece jigsaw puzzle without knowing what the picture on the cover of the box is" ...

De novo Genome Assembly

(with short reads from NGS devices)

... but in practice we don't just make one jigsaw we make many different jigsaws and compare them to see which one looks best.

This presents a number of questions:

- Why create many assemblies?
- What varies between assemblies?
- Which assembly is best?

Why Create Many Assemblies?

- At TGAC we sequence different types of organism. There's no one fixed pipeline that works well for all cases

- Project collaborators want the highest quality assembly we can feasibly create
    - without requiring more sequencing!

- One assembler may not work well for all genomes with all types of sequence data

- Assemblers have parameters! Which ones are best?

What Varies Between Assemblies?

- Choice of assembler
- K length
- Various pre-processing options (changing the jigsaw pieces)
    - Subsampling (Level of coverage)
    - Error correction / Quality Trimming
- Various post-processing options
    - Scaffolding (fits chunks together - creates gaps where no pieces fit)
    - Gap Closing (tries to fill the gaps)

Which Assembly is Best?

The one that matches our expectations?

- e.g. Known genome length

The most contiguous?

- Risks rewarding aggressive assemblers

Alignments

- Reads to Assembly
- Assembly to Reference

Other approaches?

RAMPART

- A configurable pipeline for generating and comparing multiple de novo genome assemblies
- Reduce the bioinformatician's workload, increase consistency and reduce errors
- Single interface to multiple assemblers
- Make efficient use of HPC Resources
- Open Source and Portable

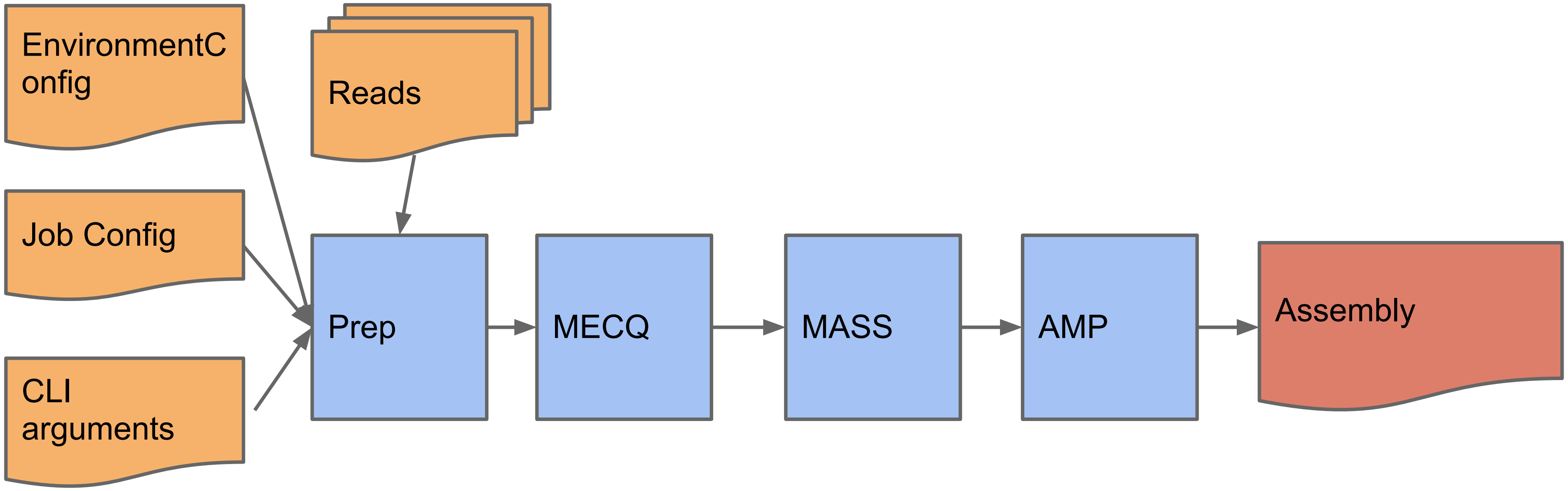
RAMPART Pipeline - Usage

Input:

- Short reads
- Job configuration file
- Environment configuration file
- CLI arguments

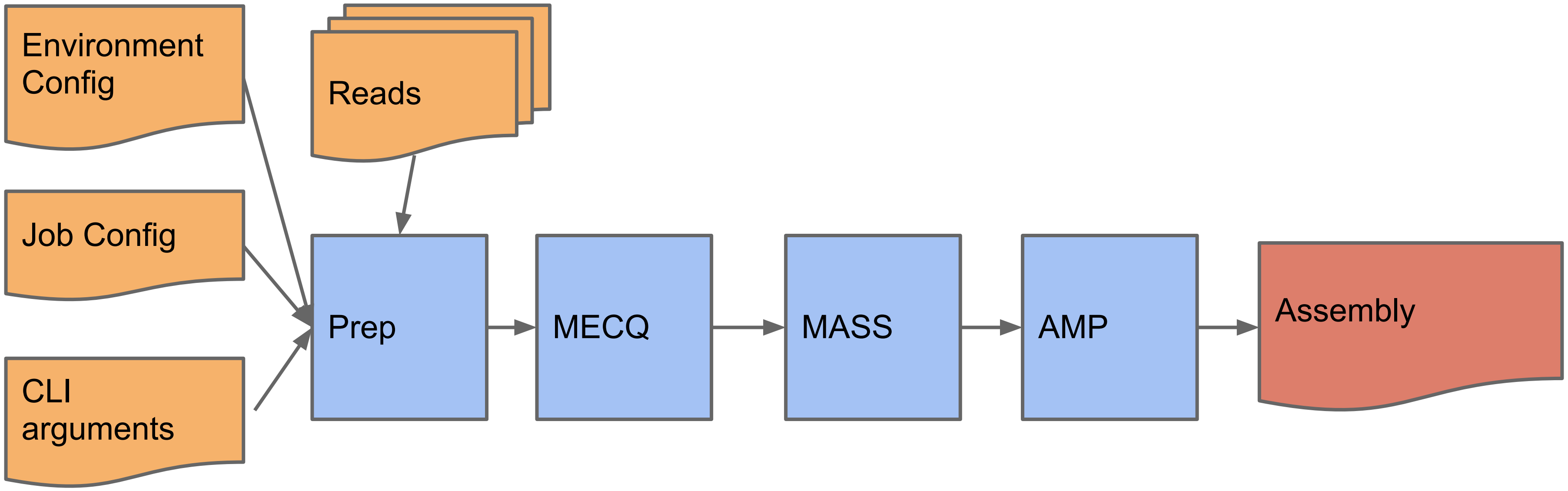
Output:

- Single optimised assembly
- All other assemblies built during the job
- Graphs
- Tables
- Logs



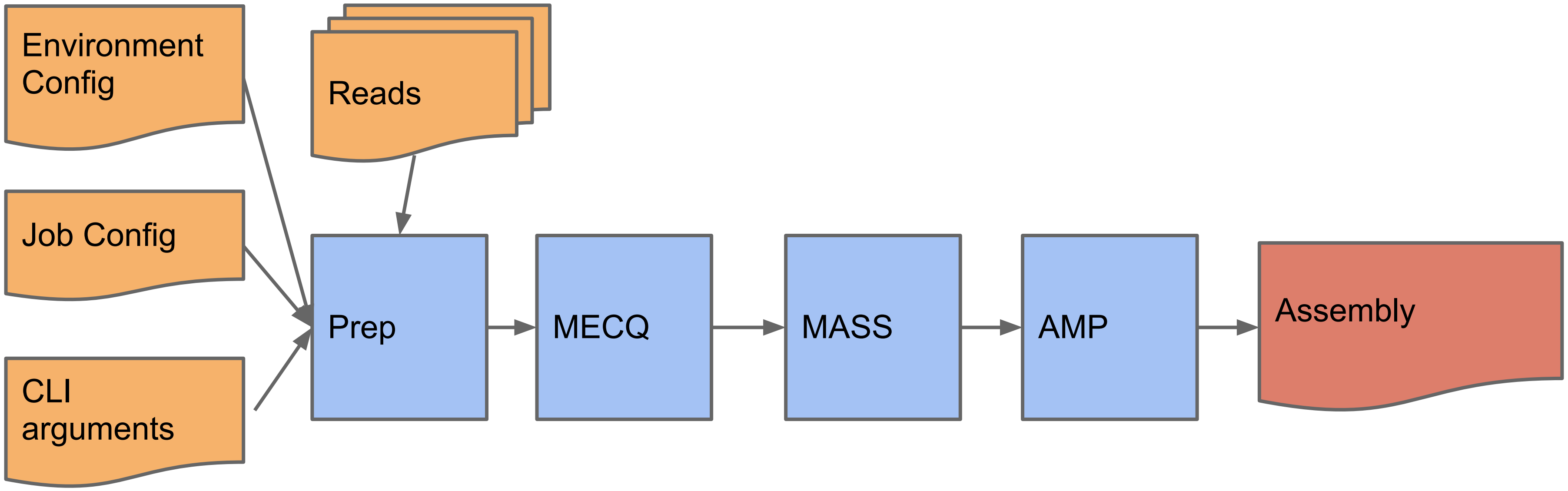
RAMPART Pipeline - Stages

- MECQ (Multiple Error Correction and Quality trimming tool)
- MASS (Multiple ASSembly tool)
- AMP (Assembly iMProver)



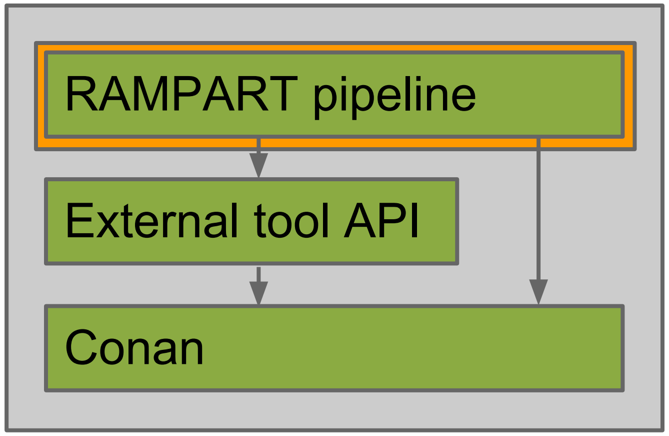
RAMPART Pipeline - Stages

- MECQ (Multiple Error Correction and Quality trimming tool)
- MASS (Multiple ASSembly tool)
- AMP (Assembly iMProver)
- **RAMPART (Robust, Automatic, Many-Parallel Assemblies Toolkit)**



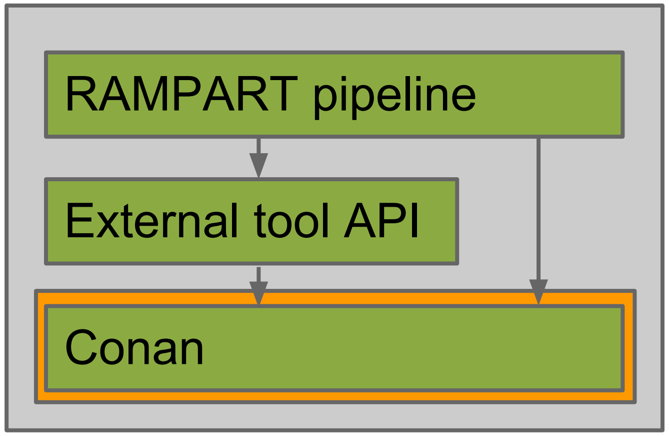
RAMPART Architecture

- Three layers in the software architecture
- Top layer is the RAMPART pipeline shown in the previous slides
- Each layer is a Java / Maven Project
- Each project and all other dependencies are compiled into a single executable jar



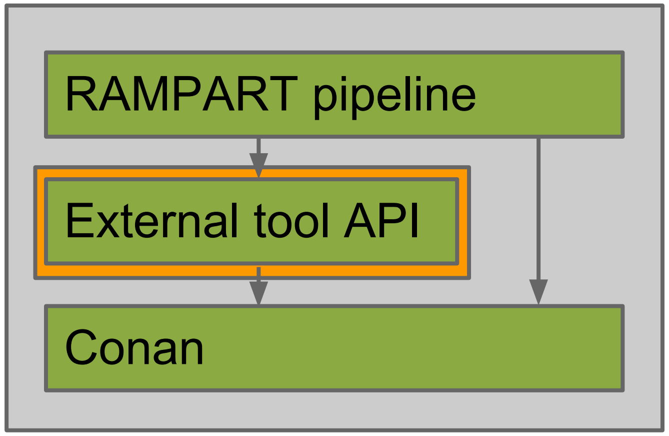
RAMPART Architecture - Conan

- *"extremely light-weight workflow management application"*
- Features:
  - Script / Tool chaining
  - LSF support
  - Multi-user
  - Web-interface
  - Task tracking
  - Task control
- TGAC extensions (ConanX)
  - "Execution Contexts" / Scheduler selection
  - Made conan portable
  - Tool loading



RAMPART Architecture - External Tool API

- All tools are wrapped as Conan Processes for running in a Conan pipeline
- Provides consistent interface to certain classes of tools
- For example, all Assemblers are assumed to have the same features:
    - Where feature control in API is missing the tool default is used
    - Where present, generic input is translated into tool specific argument
- Handles creation of tool specific configuration files automatically
    - e.g. SOAP library config



Current Version (0.4.1)

Tools:

- MECQ:
  - Quake
  - Musket
  - Sickle
- MASS:
  - Abyss
  - SOAP2
- AMP:
  - SSPACE
  - SOAP GapCloser

Features:

- Scheduling:
  - LSF
  - PBS (untested)
- Parameter optimisation:
  - K length
  - Read pre-processing
  - Auto select best assembly (optional)
- Assembly Contiguity comparison and analysis
- Single interface for customisable pipeline
- Environment agnostic

Future

Tools:

- Prep:
  - KmerGenie (or TGAC variant)
- MASS:
  - ALLPATHS-LG

Features:

- Web interface
  - User login
  - Job tracking
  - Pause / Resume jobs

Features:

- Scheduling: DRMAA Distributed Resource Management Application API — www.drmaa.org
- Enhanced comparison and validation
  - Kmer-based analysis
  - Feature Response Curves (FRCbam)
  - Contamination detection
  - Scaffolding analysis
- Read coverage optimisation
- Post-processing options:
  - Deduplication
  - Header formatting

Availability

(V1.6)

- https://github.com/TGAC/RAMPART
- https://github.com/TGAC/TgacConanProcs

- https://github.com/tburdett/Conan2.git (ConanX branch)

Current stable version is tagged as V0.4.1 - Pre compiled shaded jar available

Summary

- Not a new assembler!
  - A configurable, portable, end-to-end genome assembly pipeline
  - Automatic parameter optimisation
  - Automatic assembly comparison and validation

- Is used in TGAC for some assembly projects when we understand the data (amount, length, types) and the genome (size, heterozygosity, repeat content) and we predict the job will not demand too much computing resources

- Work in progress but:
  - Will be extended and maintained
  - Stable versions are tagged as releases in github

Acknowledgements

TGAC The Genome Analysis Centre™

Bernardo Clavijo

Robert Davey

Nizar Drou

David Swarbreck

EMBL-EBI

Tony Burdett

BBSRC bioscience for the future

Greater Norwich Development Partnership

South Norfolk COUNCIL

NORWICH City Council

Norfolk County Council

Any Questions?