CANCER/EPIGENETICS COMPUTATIONAL INFORMATICS

NGSANE

A HPC Processing Framework for Terabyte-scale Sequencing Data

Fabian A. Buske[1], Susan J. Clark[1], Denis C. Bauer[2]

[1] Cancer Epigenetics Program, Cancer Research Division, Garvan Institute of Medical Research, Kinghorn Cancer Centre, Sydney, NSW. 2010 Australia
[2] Division of Computational Informatics, CSIRO, Sydney, NSW. 2113 Australia

The initial steps in the analysis of Next Generation Sequencing (NGS) data can be automated by way of software 'pipelines'. However, individual components deprecate rapidly due to evolving technology and analysis methods, often rendering entire versions of production informatics pipelines obsolete. Constructing pipelines from Linux bash commands enables the use of hot swappable, modular components as opposed to the more rigid program-call wrapping by higher level languages, as implemented in comparable published pipelining systems. Here we present NGSANE, a Linux-based, HPC-enabled framework that minimises overhead for set-up and processing of new projects yet maintains full flexibility of custom scripting when processing raw sequence data.

HPC & parallel execution

NGSANE supports Sun Grid Engine (SGE) and Portable Batch System (PBS) job scheduling and can be operated in different modes for development and production thus enabling efficient and flexible processing of NGS data. HPC job partitioning and submission is independent from the program calls, therefore enabling new technologies (e.g. Hadoop) to be incorporated.

Data security & reusability

The framework separates project specific files from reference data, scripts, and software suites that are reusable in other projects. Access to confidential data is handled transparently via the underlying Linux permission system. The transaction between projects and framework is facilitated by a project specific configuration file that defines paths to reference data as well as the analysis tasks to perform.

Hot swapping & adaptability

NGSANE provides a unified framework (i.e. defined folder structure) for processing raw data from different experimental protocols. This allows co-investigators and reviewers to easily understand, and reproduce work using NGSANE's log and report files.

Reproducibility & checkpoint recovery

A full audit trail is generated recording performed tasks, utilised reference data, timestamps, software version as well as HPC log files, including any errors. NGSANE gracefully recovers from unsucessfully executed jobs be it due to failed commands, missing or incorrect input or under-resourced HPC jobs by cleanly restarting after the most recent successfully executed checkpoint.

Robust execution & full monitoring

In our experience, resource-intensive workflows are executed in stages with interjacent human quality control, NGSANE hence focuses on providing robust checkpointing and intuitive report generation. However, workflows can be fully automated by utilising NGSANE's control over HPC-queuing systems and by leveraging the customisable interfaces between modules when submitting multiple dependent stages at once.

Complete customisation

NGSANE's configuration file contains details about the submission system, typical resource allocations and location of third-party software. However, NGSANE's credo is that every parameter can be overwritten, hence default parameters can be adjusted in the project specific configuration file to indicate different software versions, additional resources or an altered output location.

Automated project summary creation

NGSANE generates a highlevel summary to enable informed decisions about the experimental success. This report provides an access point for new lab members or collaborators. Furthermore, the project card can be used as gold standard for software development when employing a continuous integration server.

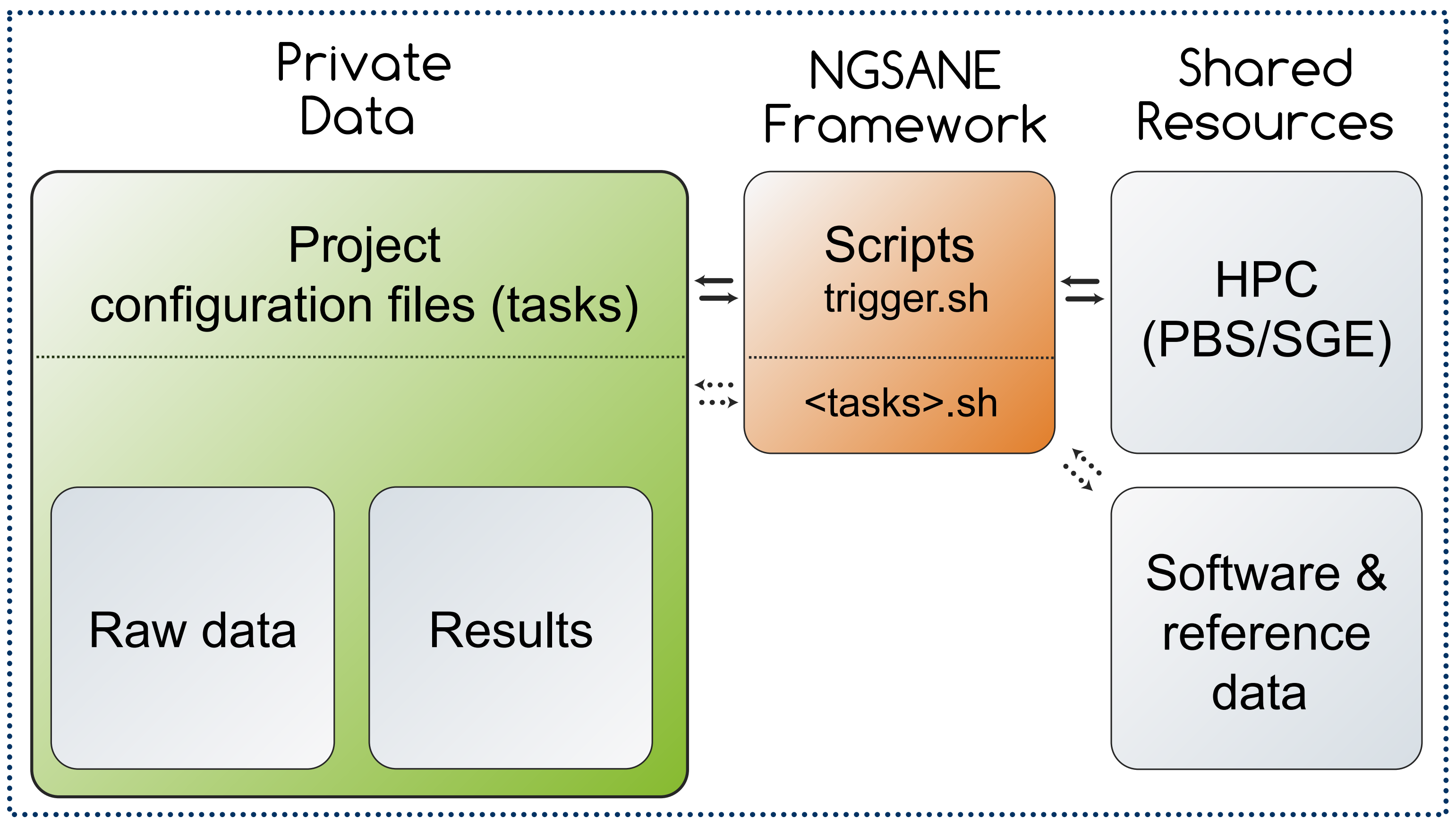
Repeated calls

Pipelines often have to be rerun on the full or a subset of the data with possibly altered parameter settings. NGSANE facilitates and documents this by allowing multiple configuration files.

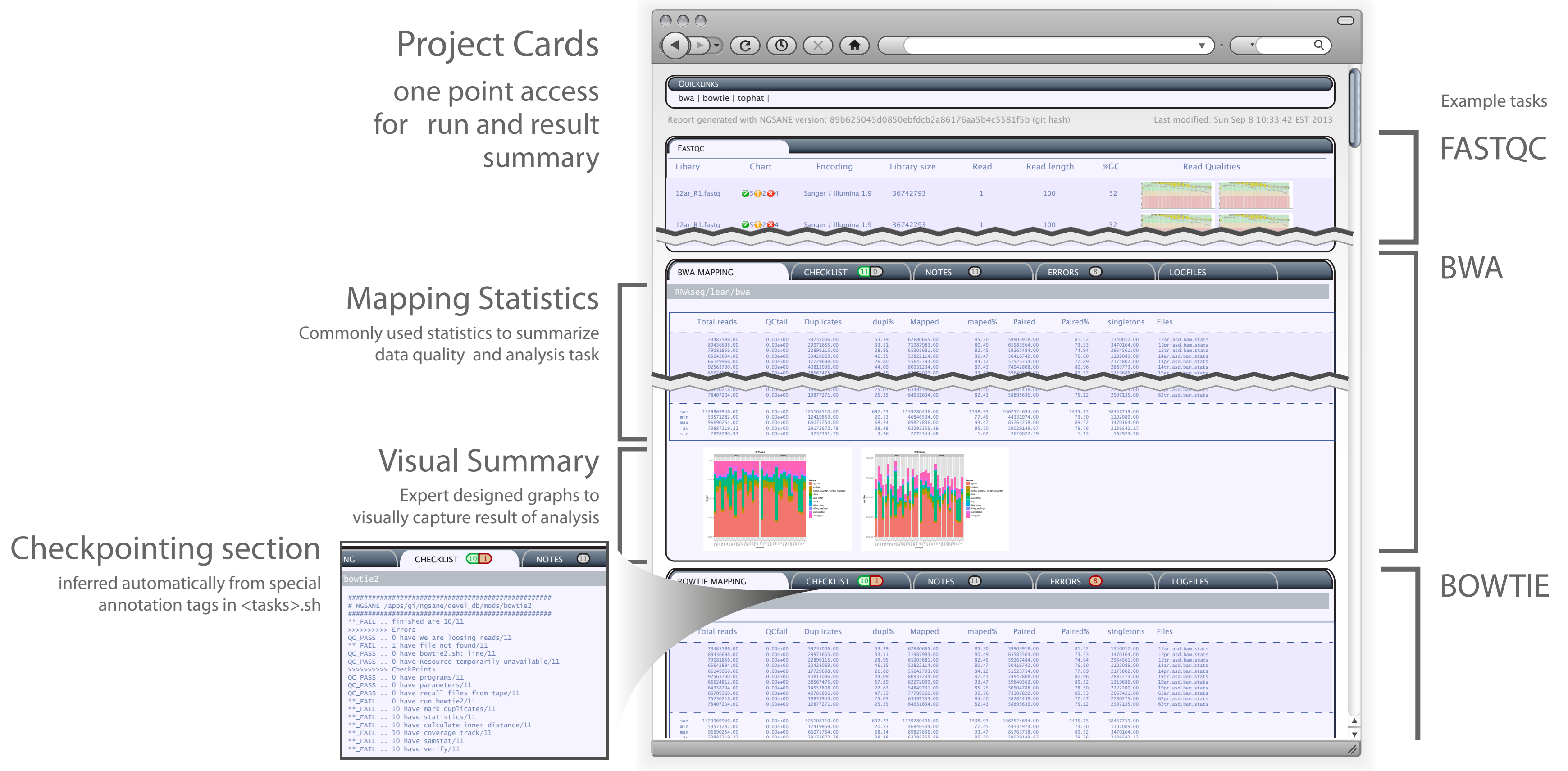
Knowledge transfer

NGSANE provides a unified framework (i.e. defined folder structure) for processing raw data from different experimental protocols. This allows co-investigators and reviewers to easily understand, and reproduce work using NGSANE's log and report files.

Concept



Example project report


Project Cards
one point access for run and result summary
Mapping Statistics
Commonly used statistics to summarize data quality and analysis task
Visual Summary
Expert designed graphs to visually capture result of analysis
Checkpointing section
inferred automatically from special annotation tags in <tasks>.sh
Example tasks
FASTQC
BWA
BOWTIE
QUICKLINKS
FASTQC
BWA MAPPING
CHECKLIST
NOTES
ERRORS
LOGFILES
BOWTIE MAPPING

bwa | bowtie | tophat |

Report generated with NGSANE version: 89b625045d0850ebfdcb2a86176aa5b4c5581f5b (git hash) Last modified: Sun Sep 8 10:33:42 EST 2013

Libary	Chart	Encoding	Library size	Read	Read length	%GC	Read Qualities
12ar_R1.fastq	5 2 4	Sanger / Illumina 1.9	36742793	1	100	52	
12ar_R1.fastq	5 2 4	Sanger / Illumina 1.9	36742793	1	100	52	

RNAseq/lean/bwa

Total reads QCfail Duplicates dupl% Mapped maped% Paired Paired% singletons Files

```
bowtie2
#########################################################
# NGSANE /apps/gi/ngsane/devel_db/mods/bowtie2
#########################################################
**_FAIL .. finished are 10/11
>>>>>>>>>> Errors
QC_PASS .. 0 have we are loosing reads/11
**_FAIL .. 1 have file not found/11
QC_PASS .. 0 have bowtie2.sh: line/11
QC_PASS .. 0 have Resource temporarily unavailable/11
>>>>>>>>>> CheckPoints
QC_PASS .. 0 have programs/11
QC_PASS .. 0 have parameters/11
QC_PASS .. 0 have recall files from tape/11
**_FAIL .. 0 have run bowtie2/11
**_FAIL .. 10 have mark duplicates/11
**_FAIL .. 10 have statistics/11
**_FAIL .. 10 have calculate inner distance/11
**_FAIL .. 10 have coverage track/11
**_FAIL .. 10 have samstat/11
**_FAIL .. 10 have verify/11
```

Currently implemented workflows include those for quality control, adapter trimming, read mapping, peak calling, motif discovery, transcript assembly, variant calling and chromatin conformation analysis. These workflows utilise publicly available published software, yet allow the end user to add their own code and create new workflows as required.

NGSANE is open source and available via GitHub at https://github.com/BauerLab/ngsane

Fabian Buske
✉ f.buske@garvan.org.au
www.garvan.org.au

Denis Bauer
✉ denis.bauer@csiro.au
www.csiro.au/CCI