

Towards visual analytics of bio-workflows

George Byelas

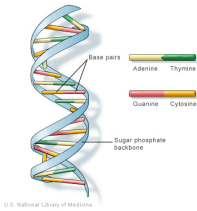
University Medical Centre Groningen, the Netherlands

NBIC-2013, Lunteren, Apr. 16th, 2013

Content

- Bio-workflows
- State of the art workflow visualizations
- How workflow visualization can be improved

Example: NGS alignment workflow



HiSeq



Raw data

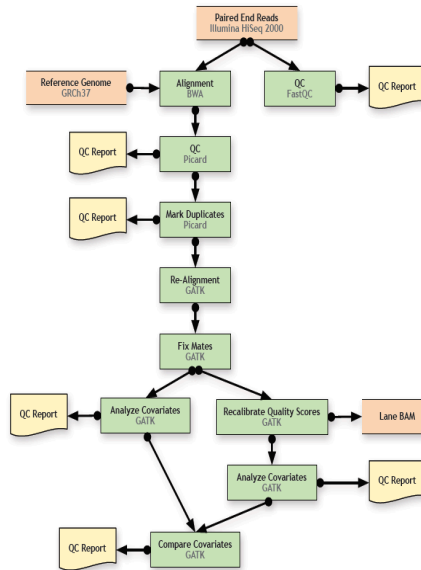
10-100 samples

**alignment
workflow**

20 – 200 days

Result data

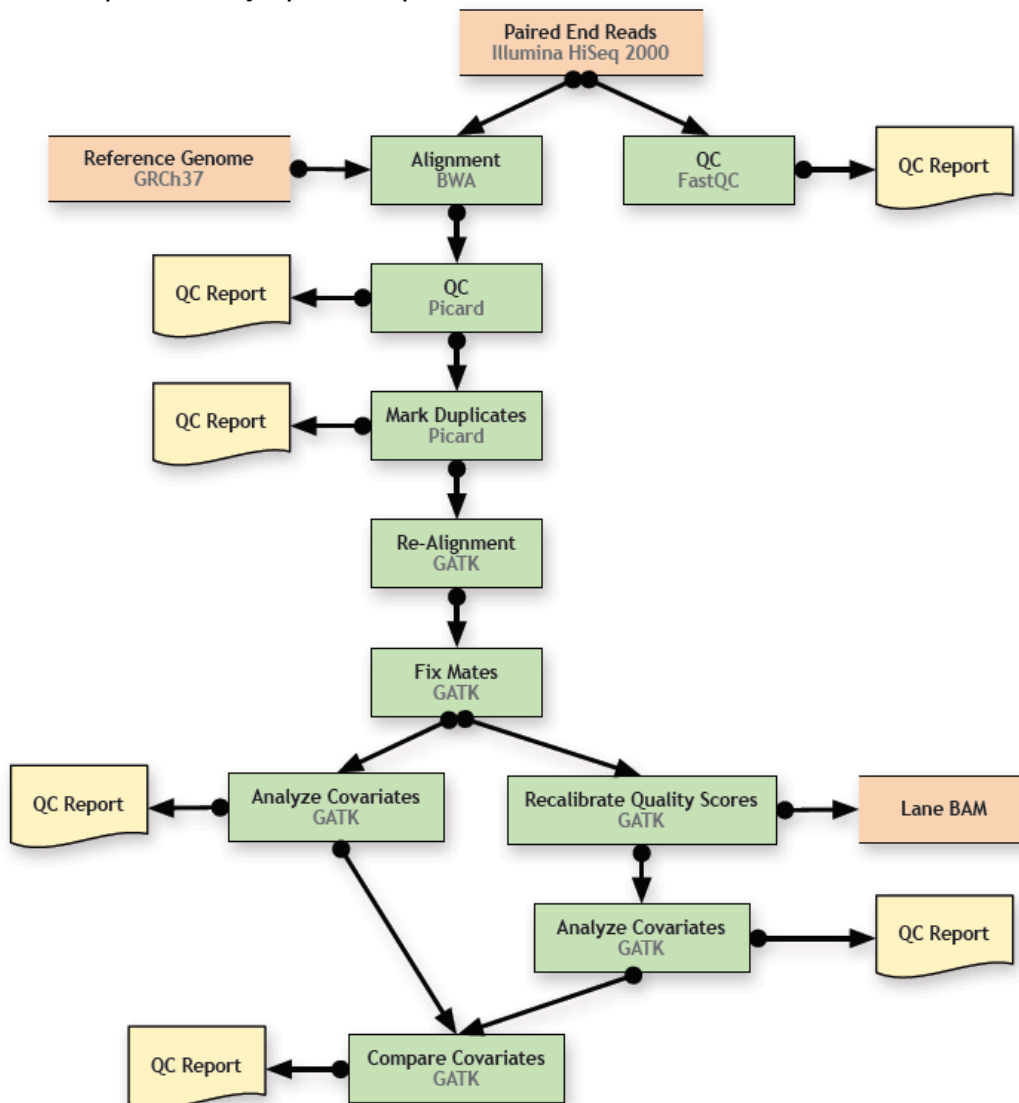
80 – 800 GB



Per Project:
1. Aligned reads
2. QC-reports
3. SNP lists

Alignment & SNP calling workflow

31 steps, ≥ 2 days per sample



• Input

- Analysis protocols
- Sample DNA data
- Reference DNA data

• Analysis

- Scripts are generated and executed

• Output

- Aligned DNA and QC reports

An analysis job (script) generated from a protocol

```
#!/bin/bash
#PBS -q test
#PBS -l nodes=1:ppn=4
#PBS -l walltime=08:00:00
#PBS -l mem=6gb
#PBS -e $GCC/test_compute/projects/batch4/intermediate/test1/err/err_test1_BwaElement1A102a_FC81D90ABXX_L7.err
#PBS -o $GCC/test_compute/projects/batch4/intermediate/test1/out/out_test1_BwaElement1A102a_FC81D90ABXX_L7.out

mkdir -p $GCC/test_compute/projects/batch4/intermediate/test1/err
mkdir -p $GCC/test_compute/projects/batch4/intermediate/test1/out
printf "test1_BwaElement1A102a_FC81D90ABXX_L7_started " >>$GCC/test_compute/projects/batch4/intermediate/test1/log_test1.txt
date "+DATE: %m/%d/%y%tTIME: %H:%M:%S" >>$GCC/test_compute/projects/batch4/intermediate/test1/log_test1.txt
date "+start time: %m/%d/%y%t %H:%M:%S" >>$GCC/test_compute/projects/batch4/intermediate/test1/
test1_BwaElement1A102a_FC81D90ABXX_L7.txt
echo running on node: `hostname` >>$GCC/test_compute/projects/batch4/intermediate/test1/
test1_BwaElement1A102a_FC81D90ABXX_L7.txt
```

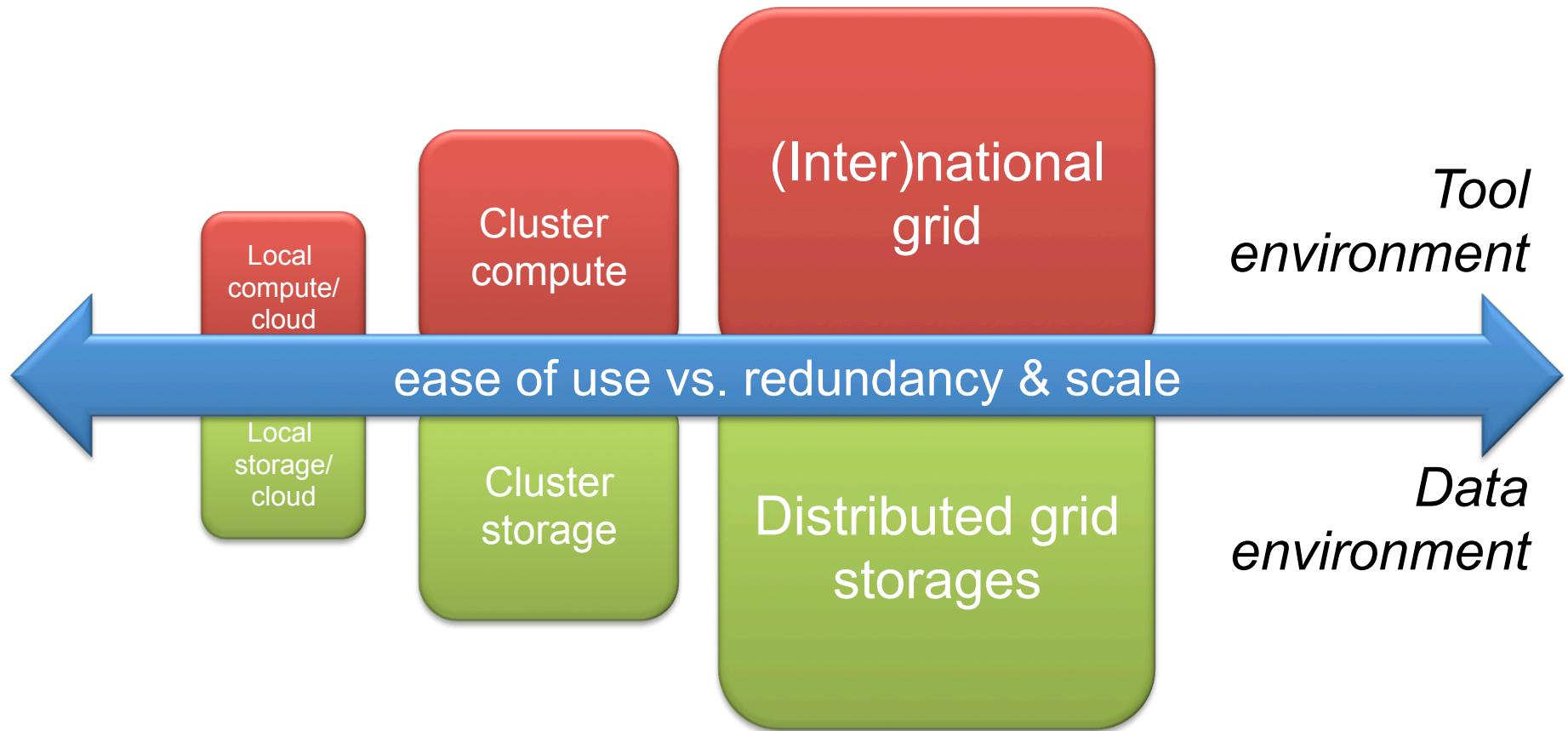
backend specific

```
/target/gpfs2/gcc/tools//bwa-0.5.8c_patched/bwa aln \
/target/gpfs2/gcc/resources/hg19/indices/human_g1k_v37.fa \
$GCC/test_compute/projects/batch4/rawdata/110121_I288_FC81D90ABXX_L7_HUMrutRGADIAAPE_1.fq.gz \
-t 4 \
-f $GCC/test_compute/projects/batch4/intermediate/A102a_110121_I288_FC81D90ABXX_L7_HUMrutRGADIAAPE_1.fq.gz.sai
```

analysis specific

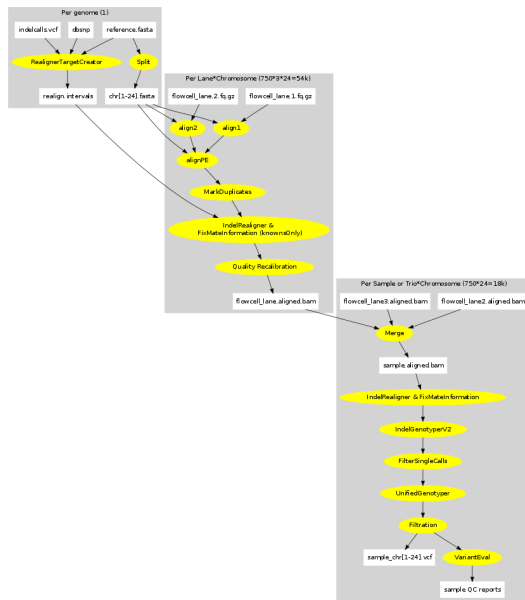
```
printf "test1_BwaElement1A102a_FC81D90ABXX_L7_finished " >>$GCC/test_compute/projects/batch4/intermediate/test1/log_test1.txt
date "+finish time: %m/%d/%y%t %H:%M:%S" >>$GCC/test_compute/projects/batch4/intermediate/test1/
test1_BwaElement1A102a_FC81D90ABXX_L7.txt
date "+DATE: %m/%d/%y%tTIME: %H:%M:%S" >>$GCC/test_compute/projects/batch4/intermediate/test1/log_test1.txt
```

Computational environments



Bio-workflow complexity

- Many analysis steps
 - Many analysis jobs
 - Different analysis tools and their dependencies
- Large various data involved
- Heterogeneous resources



Number of analysis jobs to show

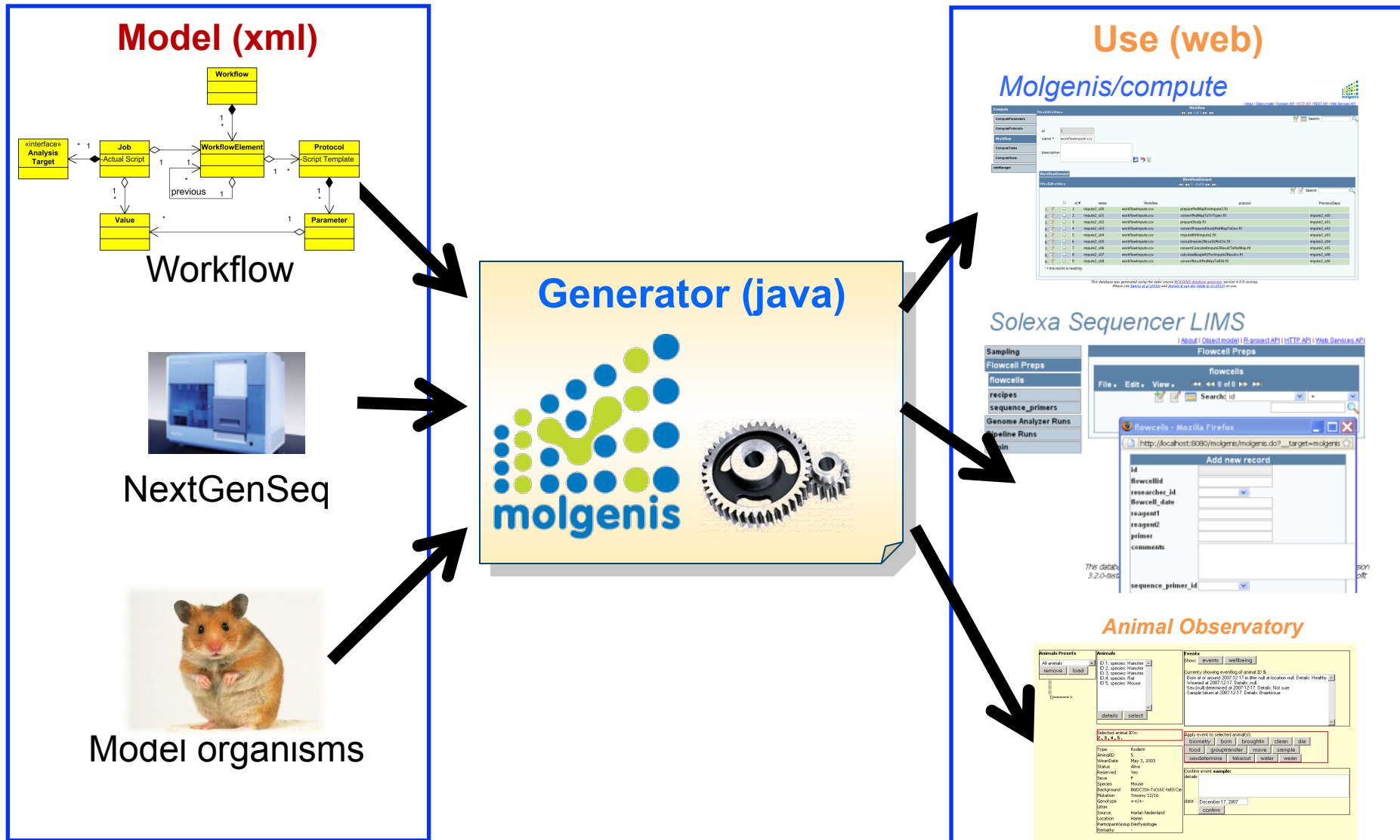
- Imputation step in GoNL

$$\text{number_jobs} = \sum_{chr=1..22} \frac{\text{Lenght}_{chr}}{5_megabase} * \frac{\text{number_of_samples}}{500}$$

$$10000 = 500 * \frac{10000}{500}$$

State of the art workflow visualizations

MOLGENIS software toolkit



Workflow design view in the generated Molgenis web-UI

Compute Testing



Compute Workflow

File Edit View

id: 1
name *: workflowImpute.csv
description:

WorkflowElement

File Edit View

	id	name	Workflow	protocol	PreviousSteps
1.	1	impute2_s00	workflowImpute.csv	preparePedMapForImpute2.ftl	
2.	2	impute2_s01	workflowImpute.csv	convertPedMapToTriTyper.ftl	impute2_s00
3.	3	impute2_s02	workflowImpute.csv	prepareStudy.ftl	impute2_s01
4.	4	impute2_s03	workflowImpute.csv	convertPreparedStudyPedMapToGen.ftl	impute2_s02
5.	5	impute2_s04	workflowImpute.csv	imputeWithImpute2.ftl	impute2_s03
6.	6	impute2_s05	workflowImpute.csv	concatImpute2ResultsPerChr.ftl	impute2_s04
7.	7	impute2_s06	workflowImpute.csv	convertConcatatedImpute2ResultToPedMap.ftl	impute2_s05
8.	8	impute2_s07	workflowImpute.csv	calculateBragleR2ForImpute2Results.ftl	impute2_s06
9.	9	impute2_s08	workflowImpute.csv	convertResultPedMapToB36.ftl	impute2_s06

* = this record is readonly.

workflow
step

analysis
protocol

previous
steps

Failed jobs overview

Search results where: **StatusCode = failed**

Interpreter	PrevSteps	requirements	StatusCode
bash	1		failed

chr: 4 from: 185000001 to: 190000001

Running on node: v33-45.gina.sara.nl

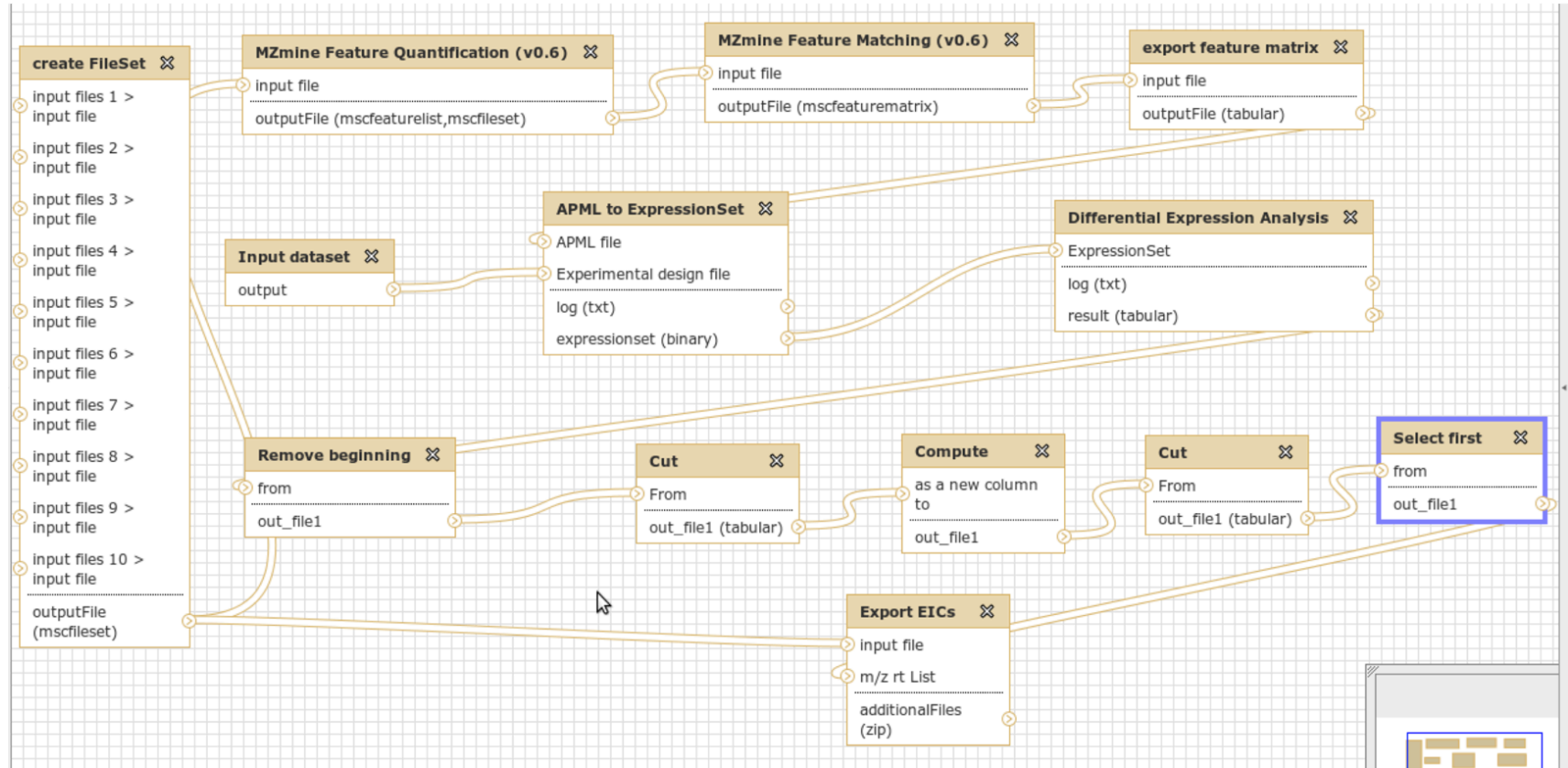
Error: terminate called after throwing an instance of 'std::bad_alloc'
what(): St9bad_alloc

How much memory:
virtual memory (kbytes, -v) 4194304

id	ComputeTask	ComputeWorker	StatusCode	StatusTime
4895	imputation_run01_6664676713199039	ulimit	failed	09:31:23
4927	imputation_run01_6664676713199039	ulimit	failed	21:31:50
4940	imputation_run01_6664676713199039	ulimit	failed	07:32:04
6740	imputation_run01_6664677232470394	ulimit	failed	11:33:11
6951	imputation_run01_6664677232470394	ulimit	failed	14:43:57
7134	imputation_run01_6664677232470394	ulimit	failed	07:21:26

TaskHistory
1 - 10 of 16

Galaxy design view



Galaxy run-time view

Running workflow "galaxy101"

Step 1: Input dataset

Exons

1: UCSC Main on Human genome (genome)

Step 2: Input dataset

Features

2: UCSC Main on Human genome (genome)

Step 3: Join

Join

Output dataset 'output' from step 3

with Output dataset 'output' from step 3

with min overlap 1

with min overlap 1

with min overlap 1

with min overlap 1

Return

Only records that are in both datasets

Action:

Hide this dataset

Step 4: Group

Select data

Output dataset 'output' from step 3

Group by column

4 (value not yet validated)

Ignore case while grouping?

False

✓ Successfully ran workflow "galaxy101" and added to the queue.

- 1: UCSC Main on Human: knownGene (genome)
- 2: UCSC Main on Human: rmsk (genome)
- 3: Join on data 2 and data 1
- 4: Group on data 3
- 5: Sort on data 4
- 6: Select first on data 5
- 7: top 5 exons

History



Unnamed history

7: top 5 exons

6: Select first on data 5

5: Sort on data 4

4: Group on data 3

3: Join on data 2 and data 1

2: UCSC Main on Human: rmsk (genome)

1: UCSC Main on Human: knownGene (genome)

History

Options



Galaxy 101

7: Compare two Queries on data 6 and data 1

5 regions, format: bed, database: hg19

Info: join (GNU coreutils) 8.5

Copyright (C) 2010 Free Software

Foundation, Inc.

License GPLv3+: GNU GPL version 3 or later

<<http://gnu.org/licenses/gpl.html>>.

This is free software: you are free to change and redistribute it.

There is NO WARRANTY, to the extent permitted by law.



| display at UCSC [main](#) | view in [GeneTrack](#) | display at Ensembl [Current](#)

1. Chrom	2. Start	3. End	4. Name
chr22	18834444	18835833	uc002zoc.2_cd
chr22	20456381	20461301	uc002zsd.3_cd
chr22	21738147	21743067	uc002zuz.3_cd
chr22	46652457	46659219	uc003bhh.2_cd
chr22	21480536	21481925	uc010gsw.1_cd

6: Select first on data 5

5: Sort on data 4

4: Group on data 3

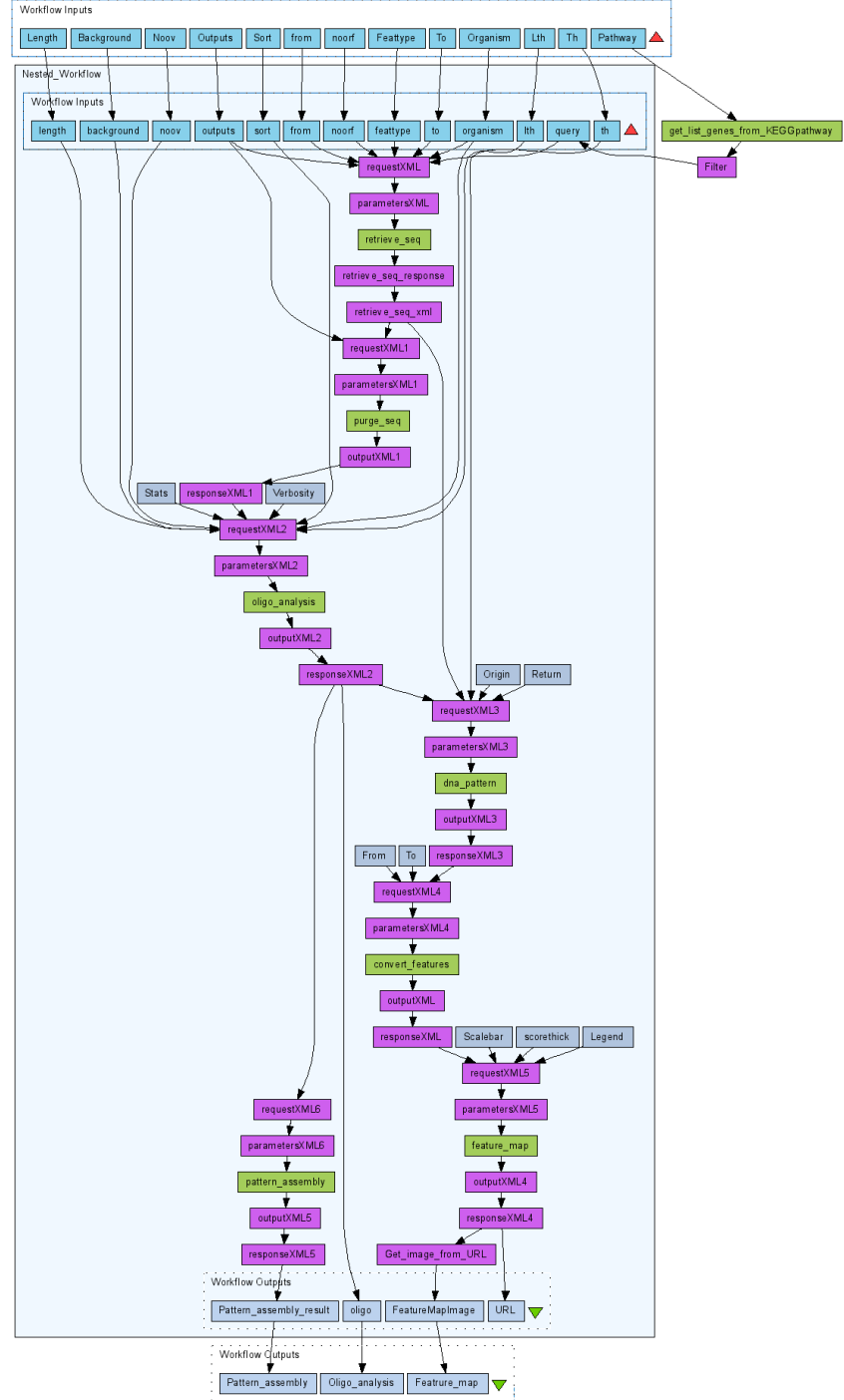
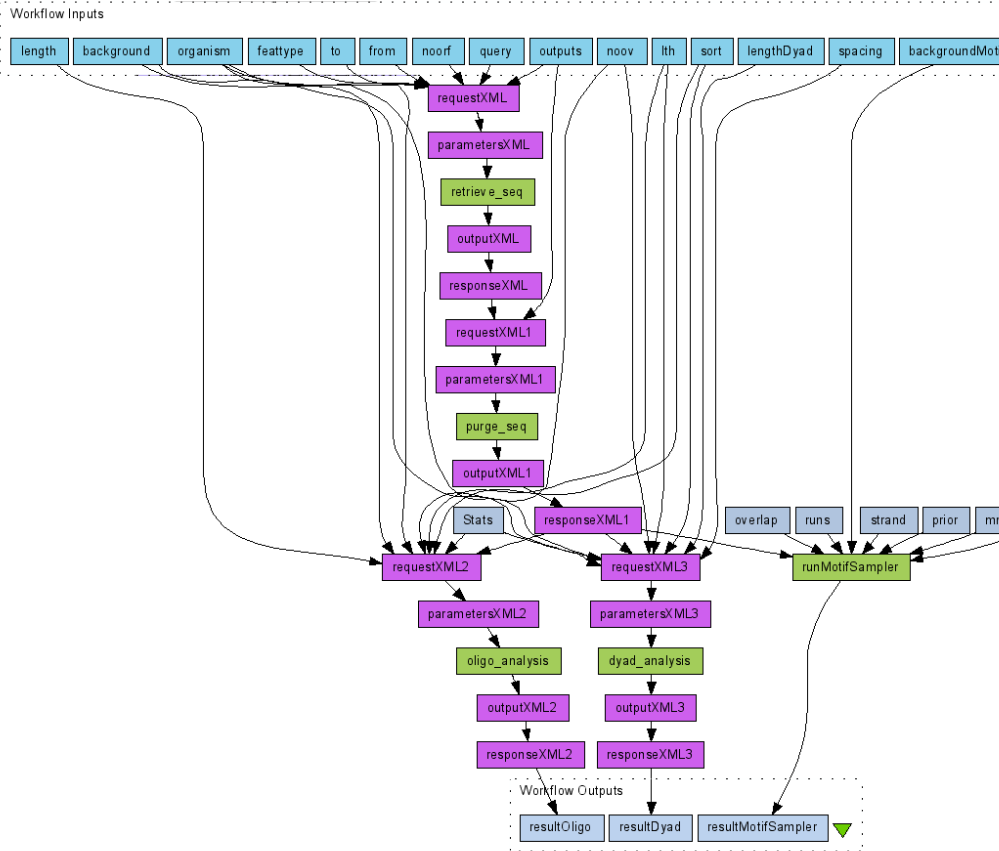
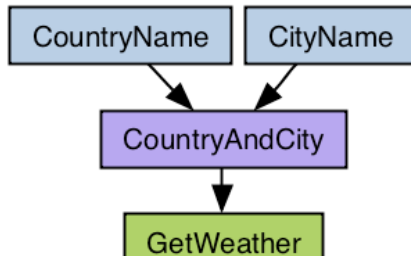
3: Join on data 2 and data 1

2: SNPs

1: Exons



Taverna

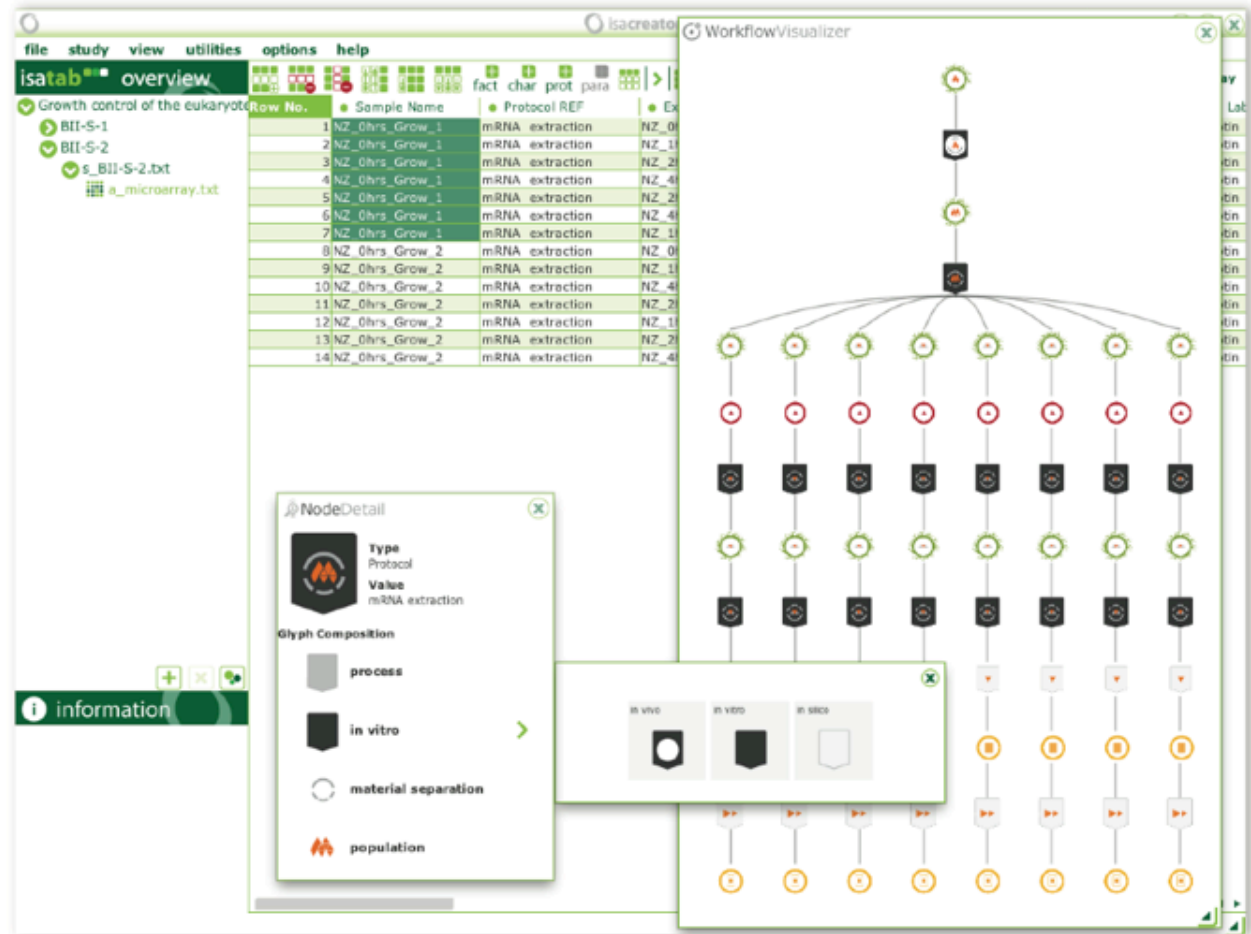


Glyph-based visualization of bio-workflows (E. Maguire *et al.*)

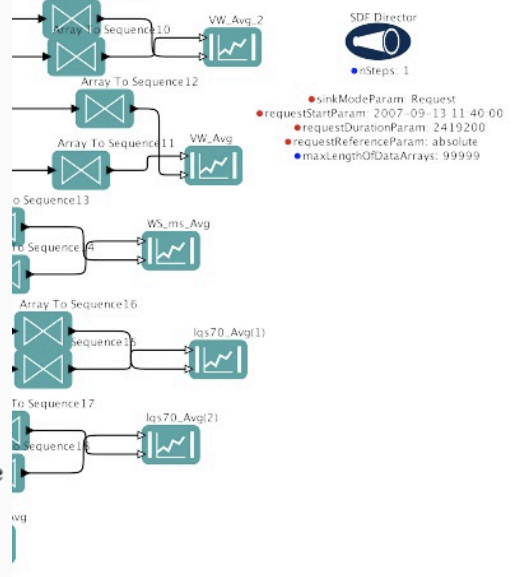
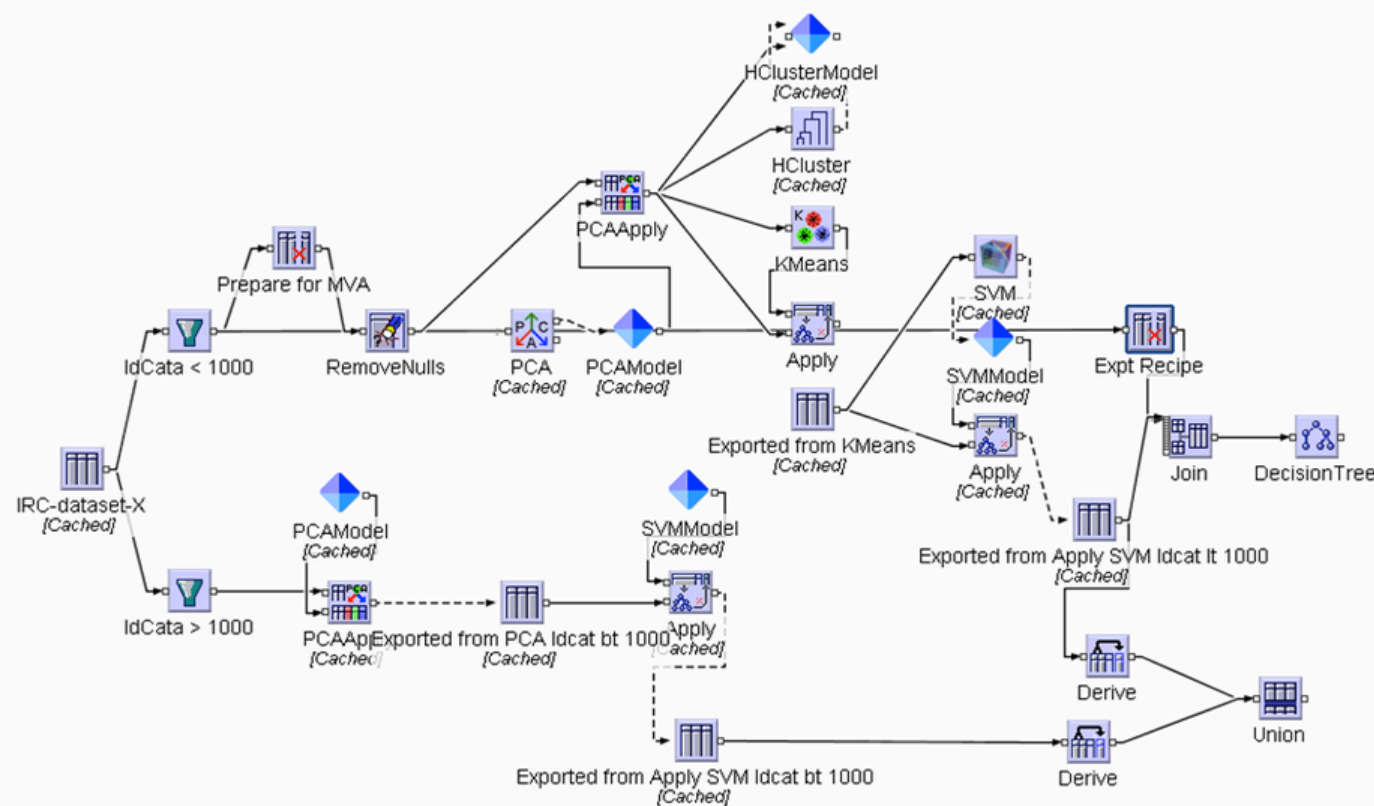
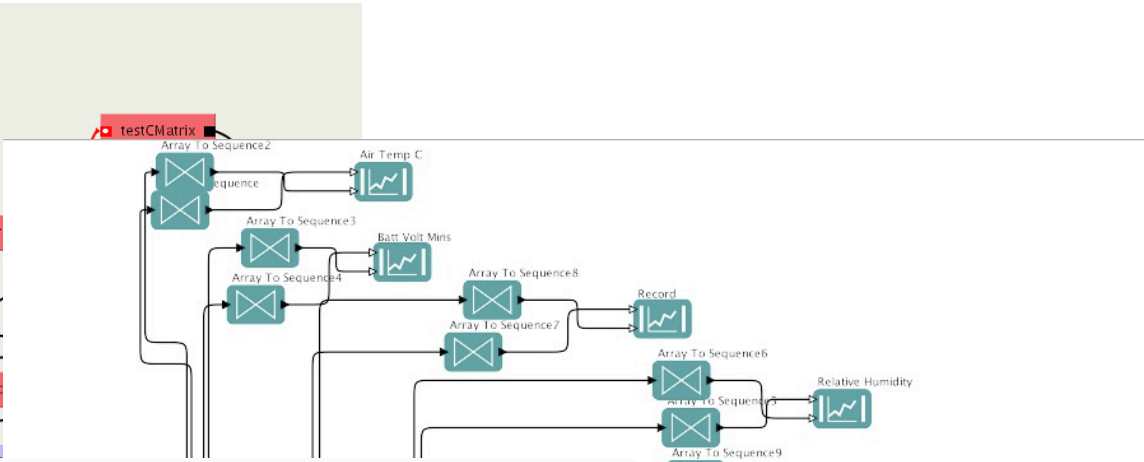
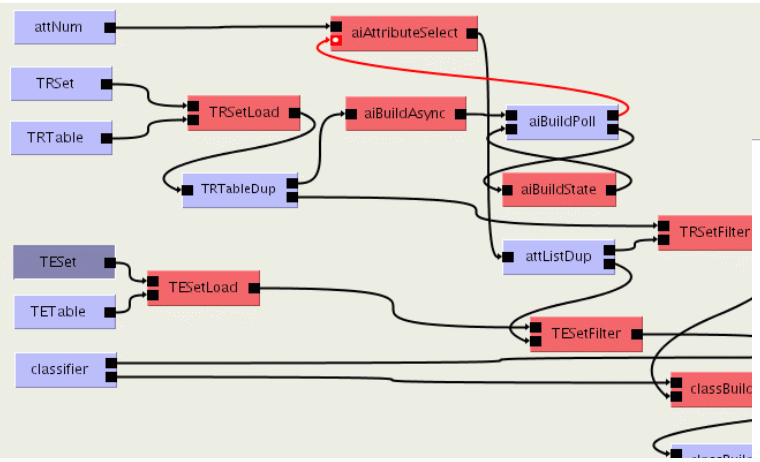
Classification

	design option 1	design option 2	design option 3	design option 4	design option 5	design option 6	design option 7
S0 Inputs and Outputs							
Process							
S7 Biological							
Device							
Chemical							
Data							
S6 In Vitro							
In Vivo							
In Silico							
S3 Data Collection							
Data Processing							
Data Analysis							
S2 Material perturbation							
Material separation							
Material amplification							
Material combination							
Material collection							
S5 Molecule							
Cellular Part							
Cell							
Tissue							
Organ							
Organism							
Population							
S4 Material induced perturbation							
Behaviourally induced perturbation							
Physically induced perturbation							

Visualization



Triana/Kepler/InforSense



SDF Director
 • nsteps: 1
 • sinkModeParam: Request
 • requestStartParam: 2007-09-13 11:40:00
 • requestDurationParam: 2419200
 • requestReferenceParam: absolute
 • maxLengthOfDataArrays: 99999

How workflow visualization can be improved (techniques taken from software visualization)

Why workflow visualization is complex

Step	Name	Command/Script	Workflow	Workflow/Task/Step	Workflow/Task/Step
1	WorkflowTask_1	WorkflowTask_1	WorkflowTask_1	WorkflowTask_1	WorkflowTask_1
2	WorkflowTask_2	WorkflowTask_2	WorkflowTask_2	WorkflowTask_2	WorkflowTask_2
3	WorkflowTask_3	WorkflowTask_3	WorkflowTask_3	WorkflowTask_3	WorkflowTask_3
4	WorkflowTask_4	WorkflowTask_4	WorkflowTask_4	WorkflowTask_4	WorkflowTask_4
5	WorkflowTask_5	WorkflowTask_5	WorkflowTask_5	WorkflowTask_5	WorkflowTask_5
6	WorkflowTask_6	WorkflowTask_6	WorkflowTask_6	WorkflowTask_6	WorkflowTask_6
7	WorkflowTask_7	WorkflowTask_7	WorkflowTask_7	WorkflowTask_7	WorkflowTask_7
8	WorkflowTask_8	WorkflowTask_8	WorkflowTask_8	WorkflowTask_8	WorkflowTask_8
9	WorkflowTask_9	WorkflowTask_9	WorkflowTask_9	WorkflowTask_9	WorkflowTask_9
10	WorkflowTask_10	WorkflowTask_10	WorkflowTask_10	WorkflowTask_10	WorkflowTask_10
11	WorkflowTask_11	WorkflowTask_11	WorkflowTask_11	WorkflowTask_11	WorkflowTask_11
12	WorkflowTask_12	WorkflowTask_12	WorkflowTask_12	WorkflowTask_12	WorkflowTask_12
13	WorkflowTask_13	WorkflowTask_13	WorkflowTask_13	WorkflowTask_13	WorkflowTask_13
14	WorkflowTask_14	WorkflowTask_14	WorkflowTask_14	WorkflowTask_14	WorkflowTask_14
15	WorkflowTask_15	WorkflowTask_15	WorkflowTask_15	WorkflowTask_15	WorkflowTask_15
16	WorkflowTask_16	WorkflowTask_16	WorkflowTask_16	WorkflowTask_16	WorkflowTask_16
17	WorkflowTask_17	WorkflowTask_17	WorkflowTask_17	WorkflowTask_17	WorkflowTask_17
18	WorkflowTask_18	WorkflowTask_18	WorkflowTask_18	WorkflowTask_18	WorkflowTask_18
19	WorkflowTask_19	WorkflowTask_19	WorkflowTask_19	WorkflowTask_19	WorkflowTask_19
20	WorkflowTask_20	WorkflowTask_20	WorkflowTask_20	WorkflowTask_20	WorkflowTask_20



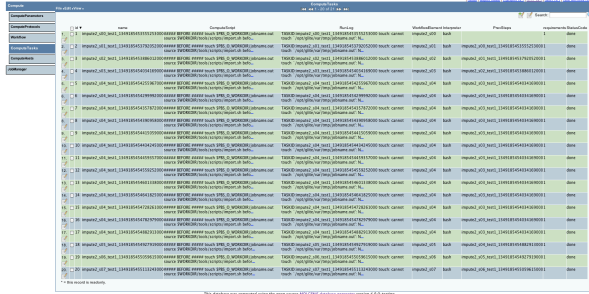
- Workflow complexity leads to
 - Large graphs/tables/logs
 - Lots of attributes
 - Workflow changes and refinements

Requirements for visual analytics

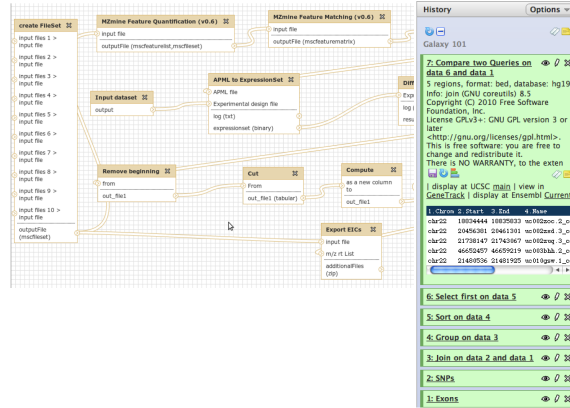
- Different views
 - Design view: structural overview and zoom in
 - Run-time execution view: history and zoom in
 - Behavior view: how parameters influence results data/quality
 - Evolution view: what elements were introduced/removed

Approach

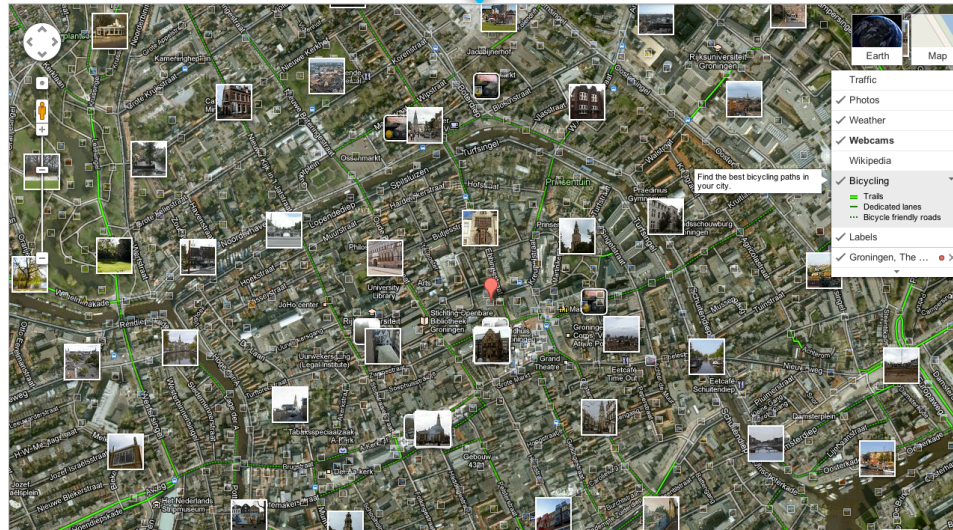
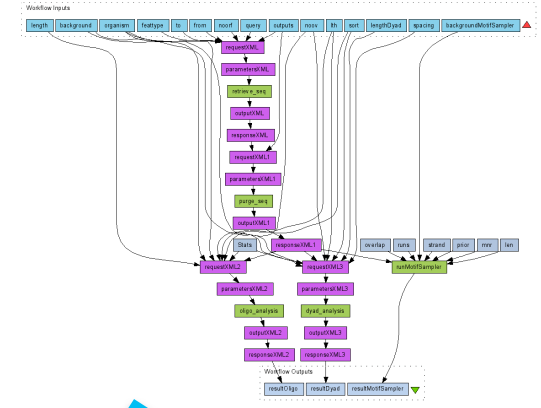
Molgenis



Galaxy



Taverna

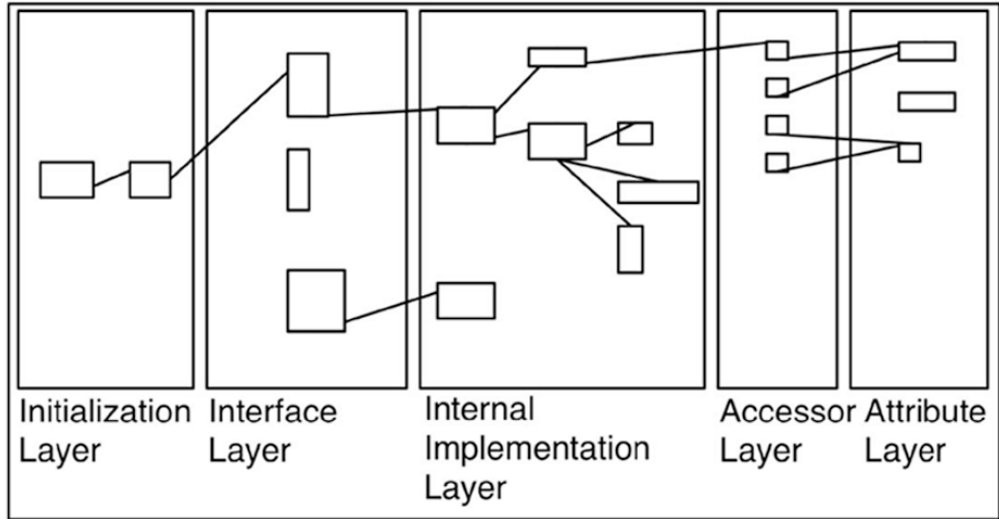
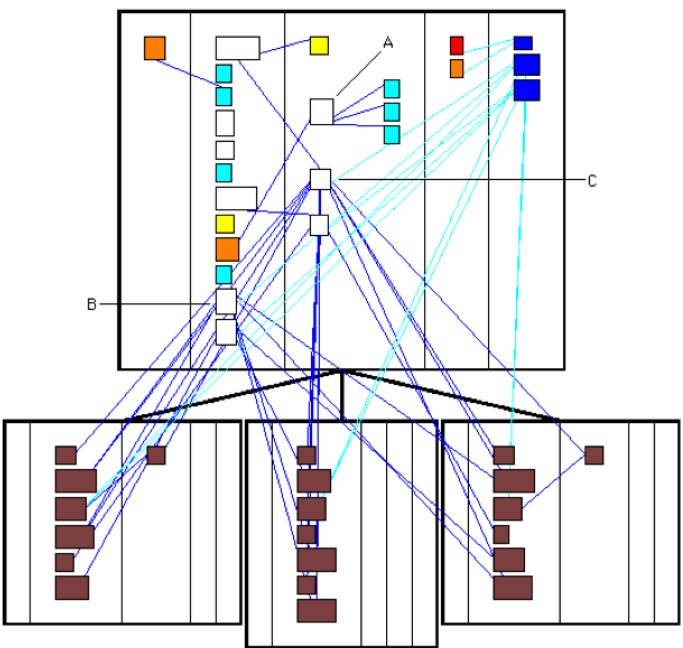
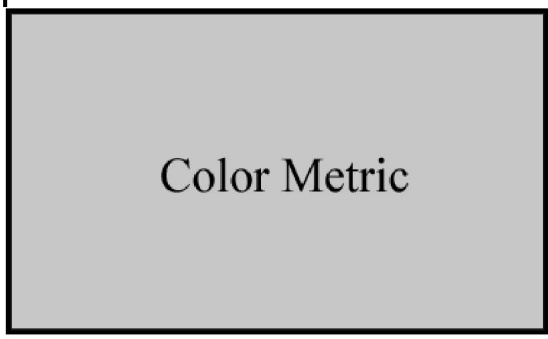


Starting point

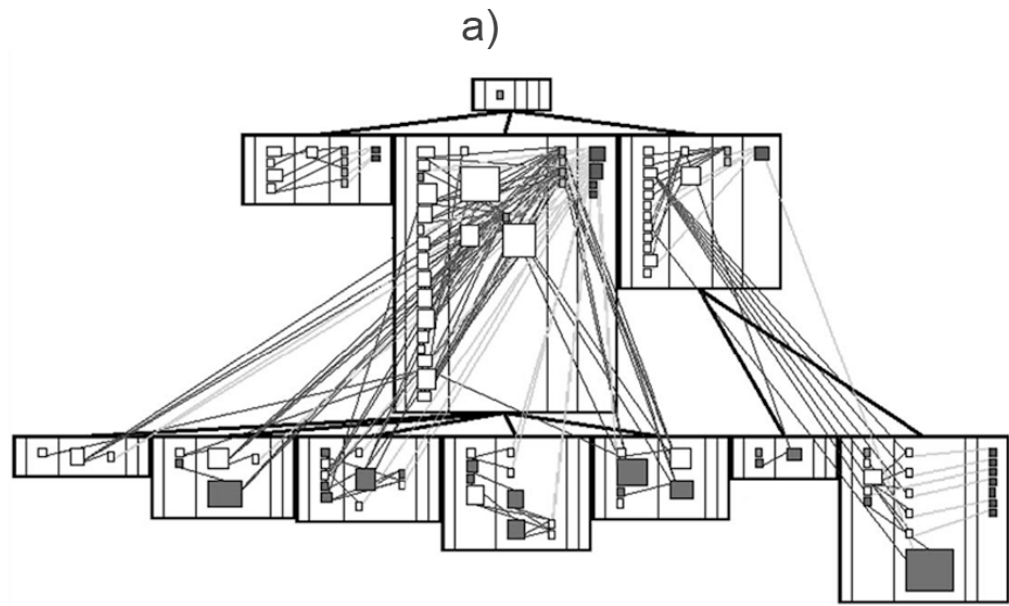
- Adding quality metrics to the workflow structure
 - # of parallel executions
 - sizes of input/output data and quality scores
 - analysis execution time
 - tools and their dependencies
 - resources required (*i.e.* CPU, RAM)

Adding quality metrics to the structural representation

Position Metrics (X,Y)



Invocation Sequence →

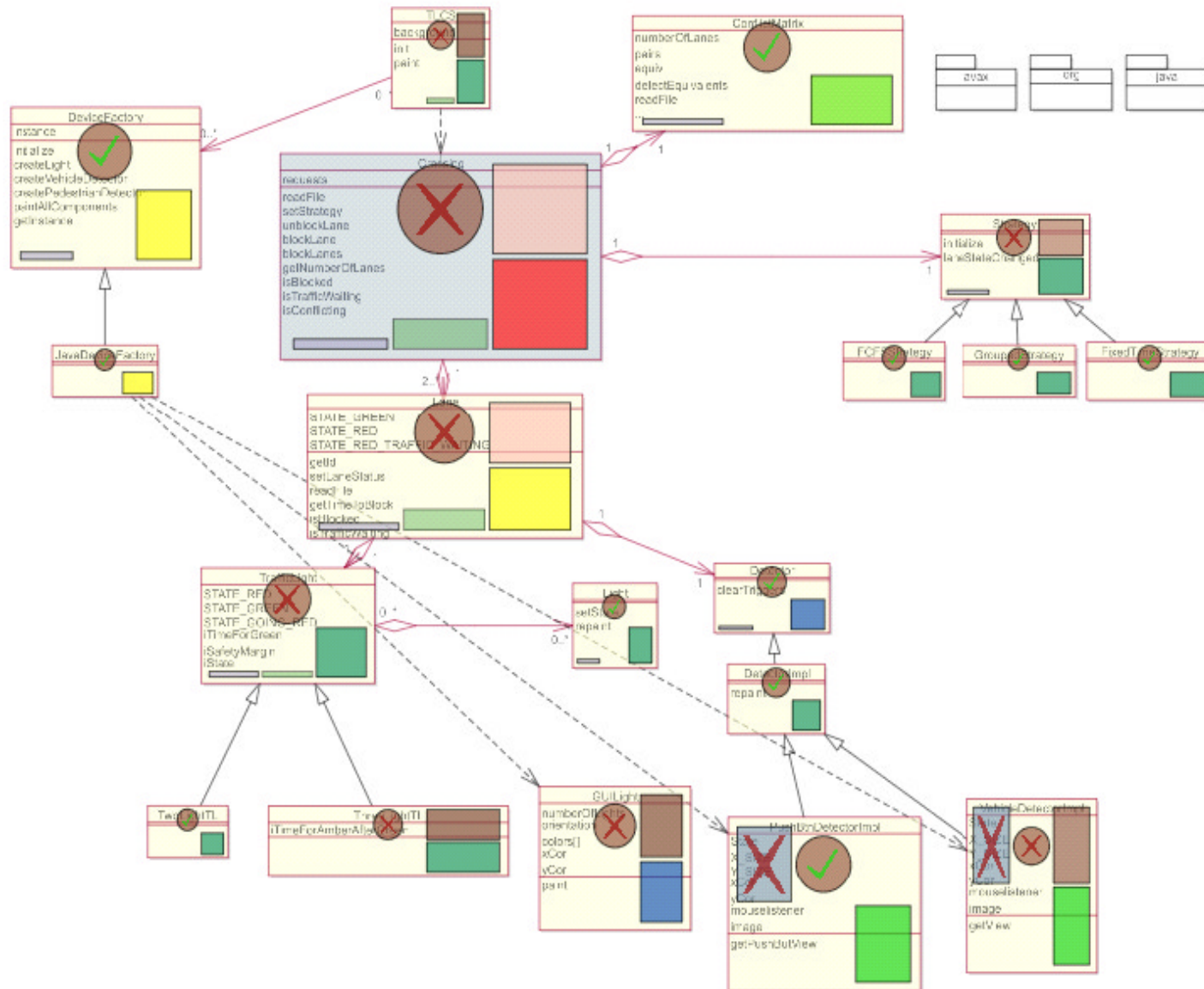


b)

Lanza et al.

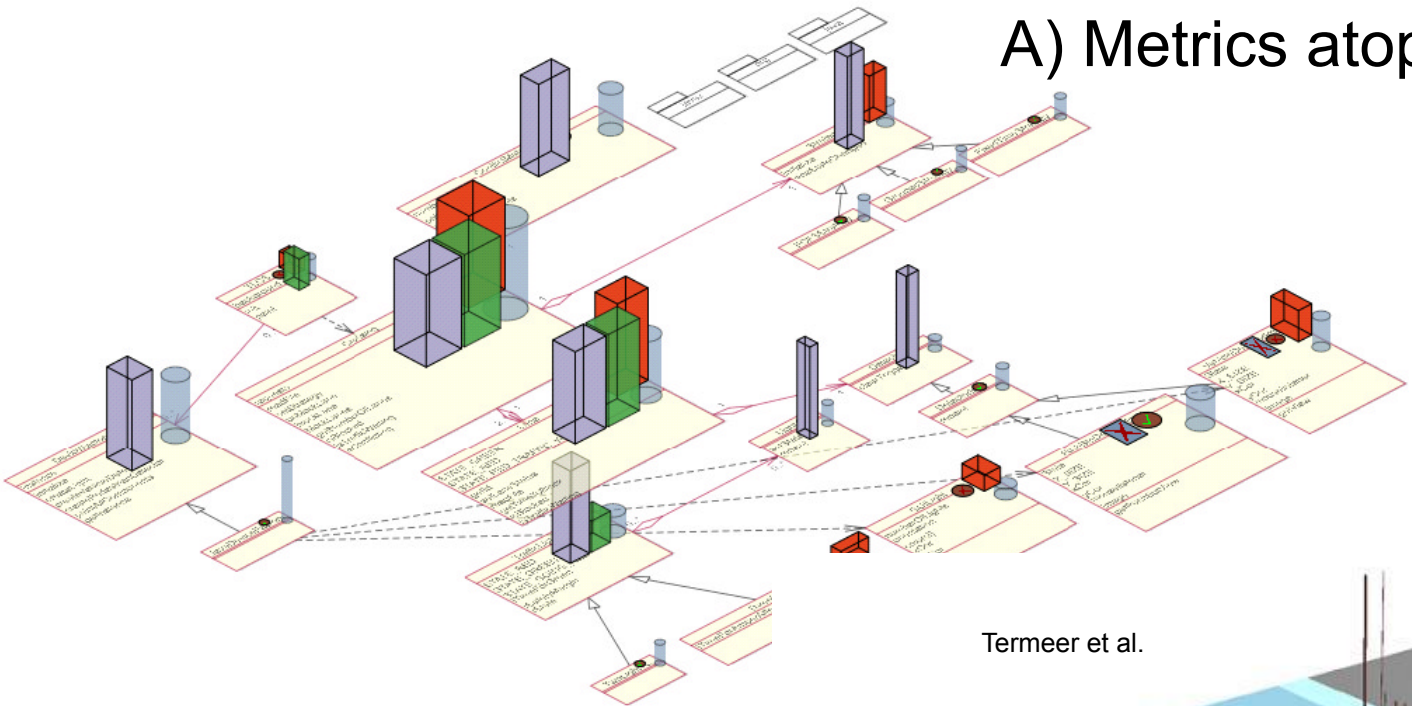


Adding quality metrics atop of a given diagram



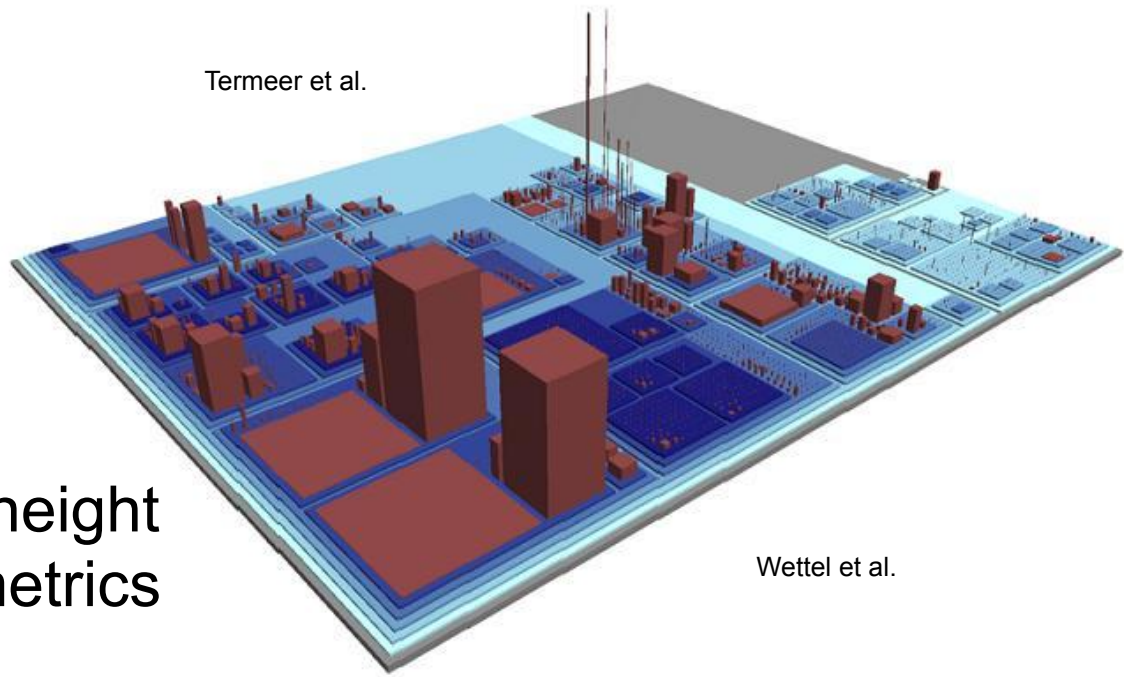
Adding metrics in 3D

A) Metrics atop of the diagram



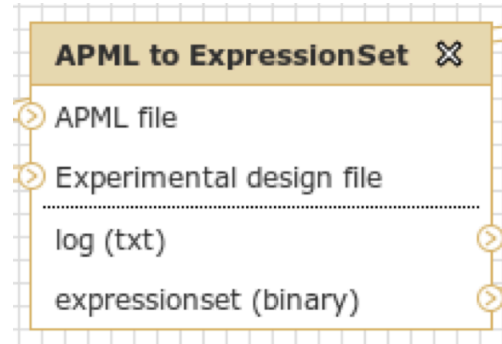
Termeer et al.

B) Using sizes and height to show metrics

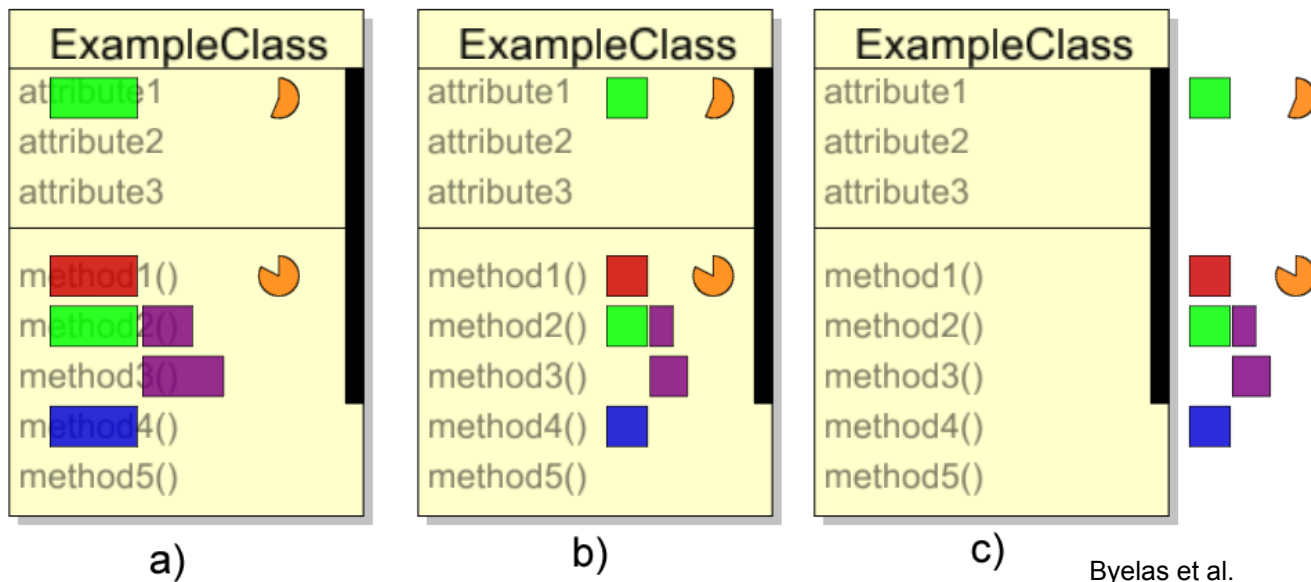


Wettel et al.

Adding metrics on the level of entity attributes

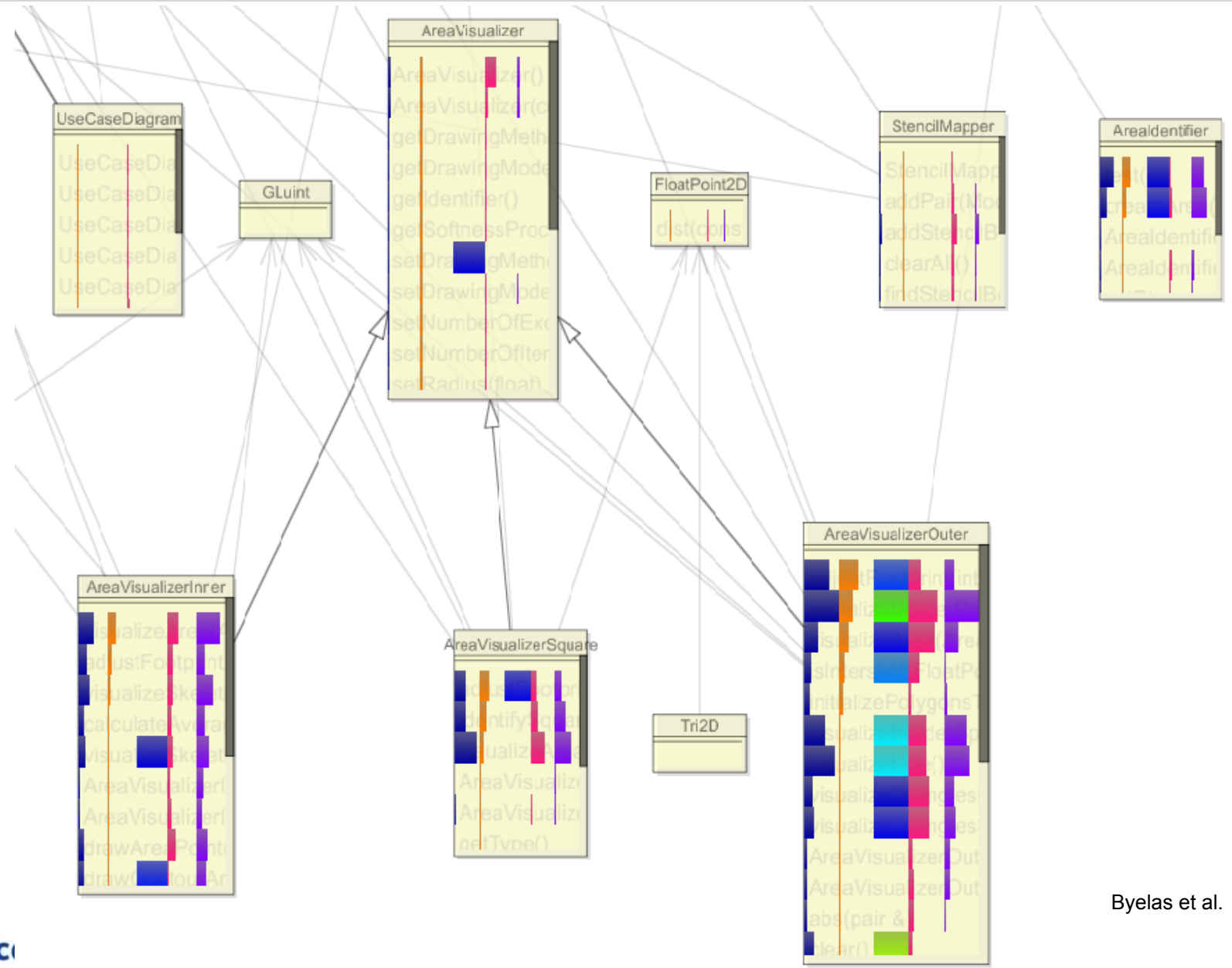


A) Galaxy workflow element



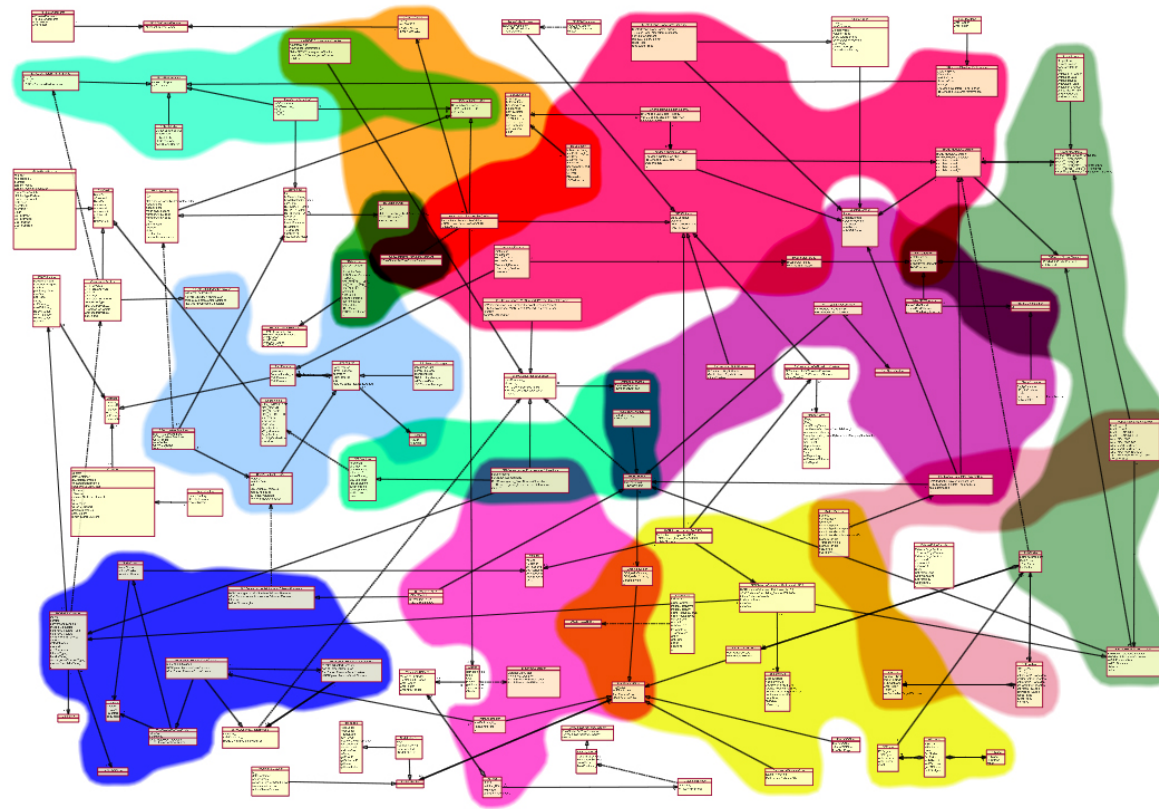
B) Adding metrics for UML class attributes

UML class diagram with 5 attribute level metrics



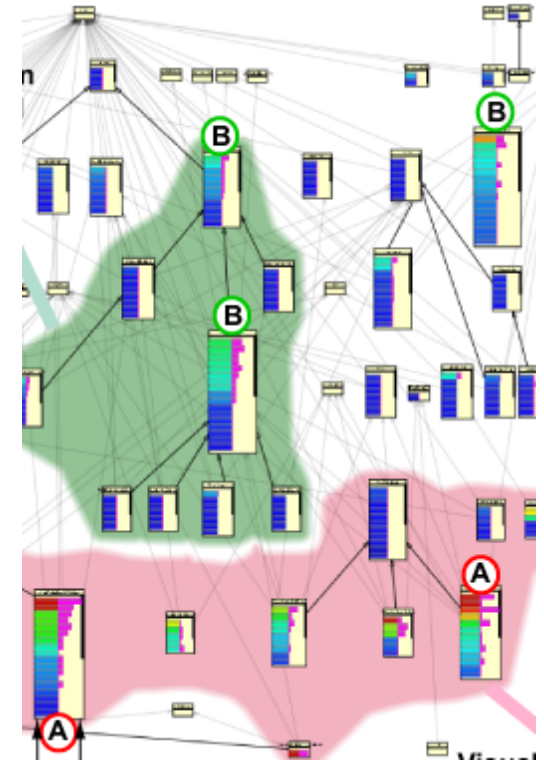
Byelas et al.

Metrics on different levels of detail



Byelas et al.

A) Showing workflow areas

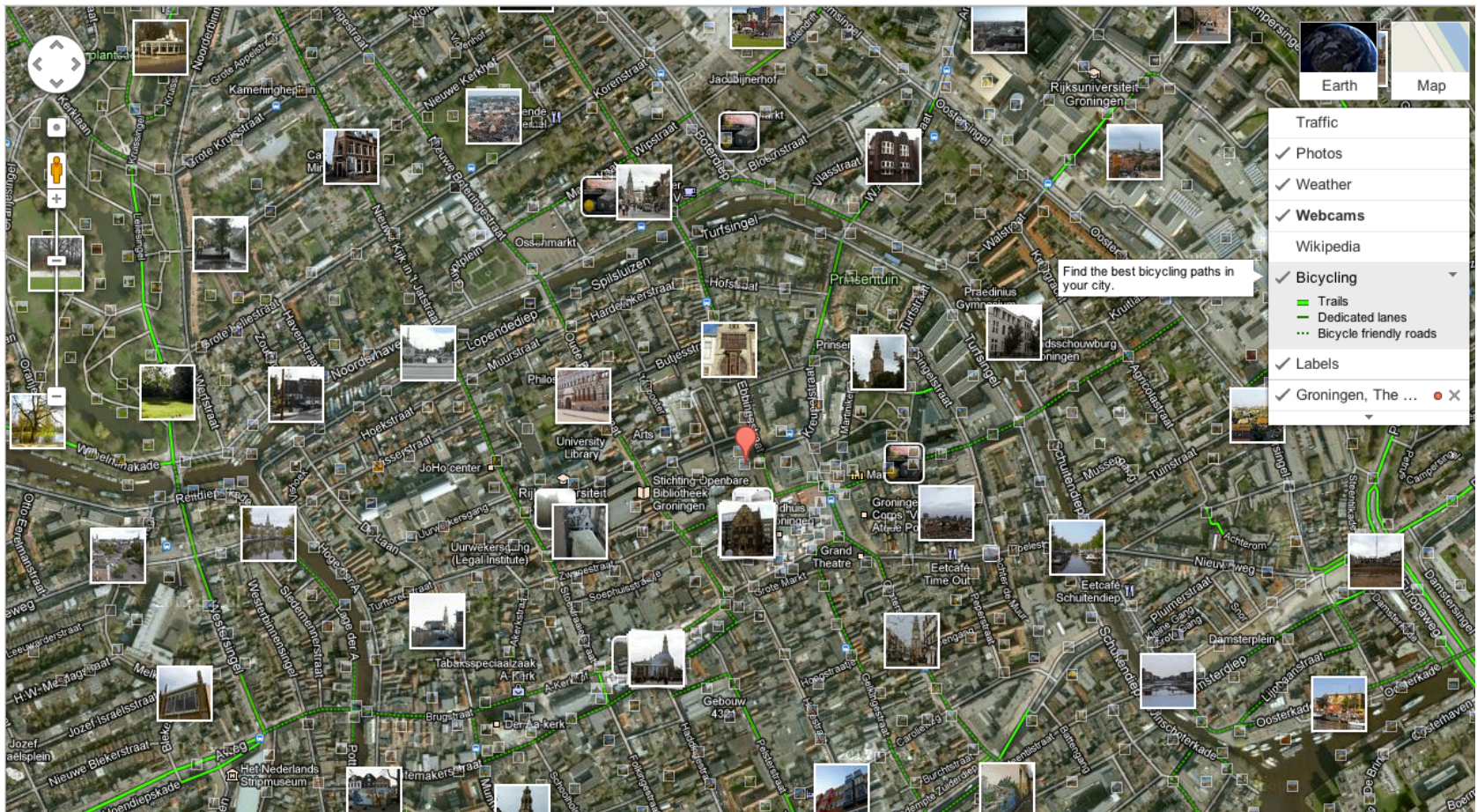


Byelas et al.

B) Metrics on different levels

Conclusion

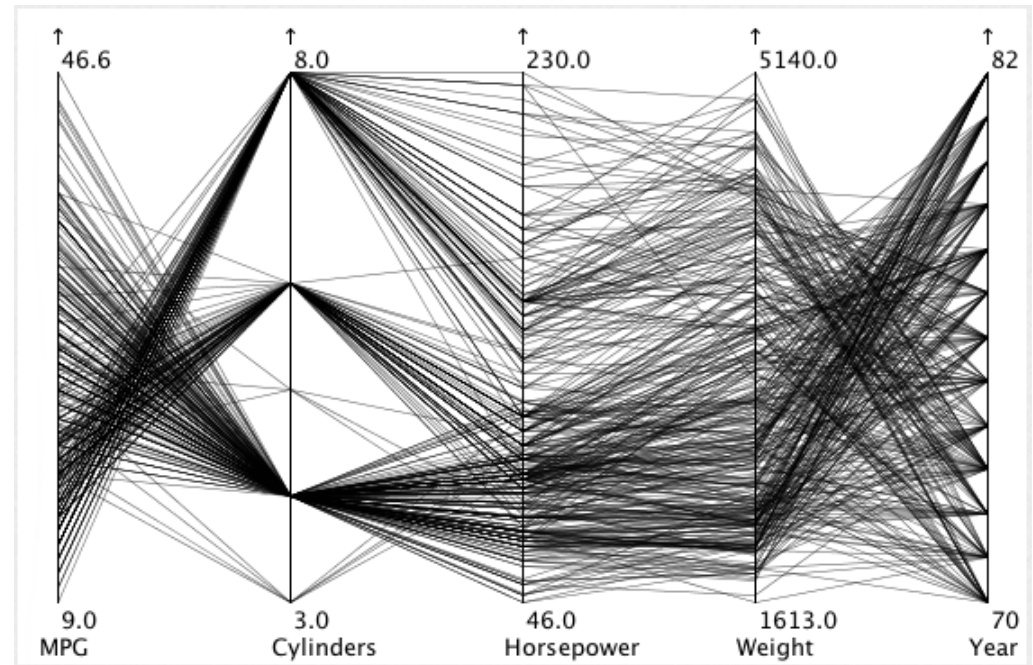
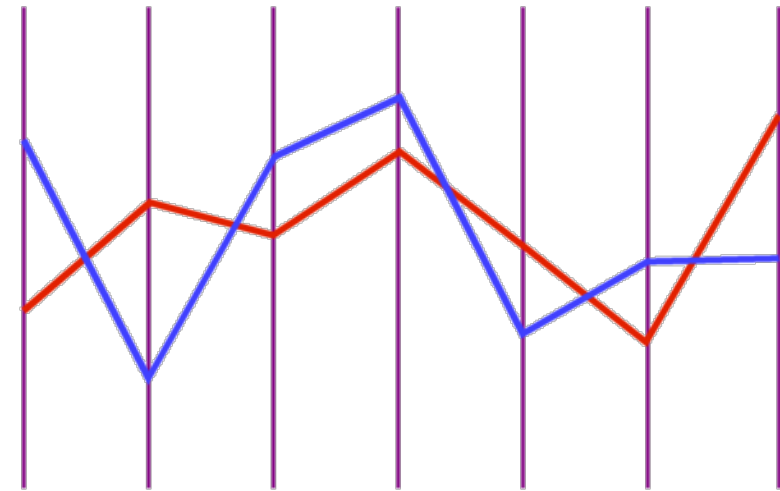
Multi-scale visualization for workflow structure



Google Maps

- Can a workflow be shown in the same way?
- Easy to hide/highlight different aspects

Workflow behavior visualization



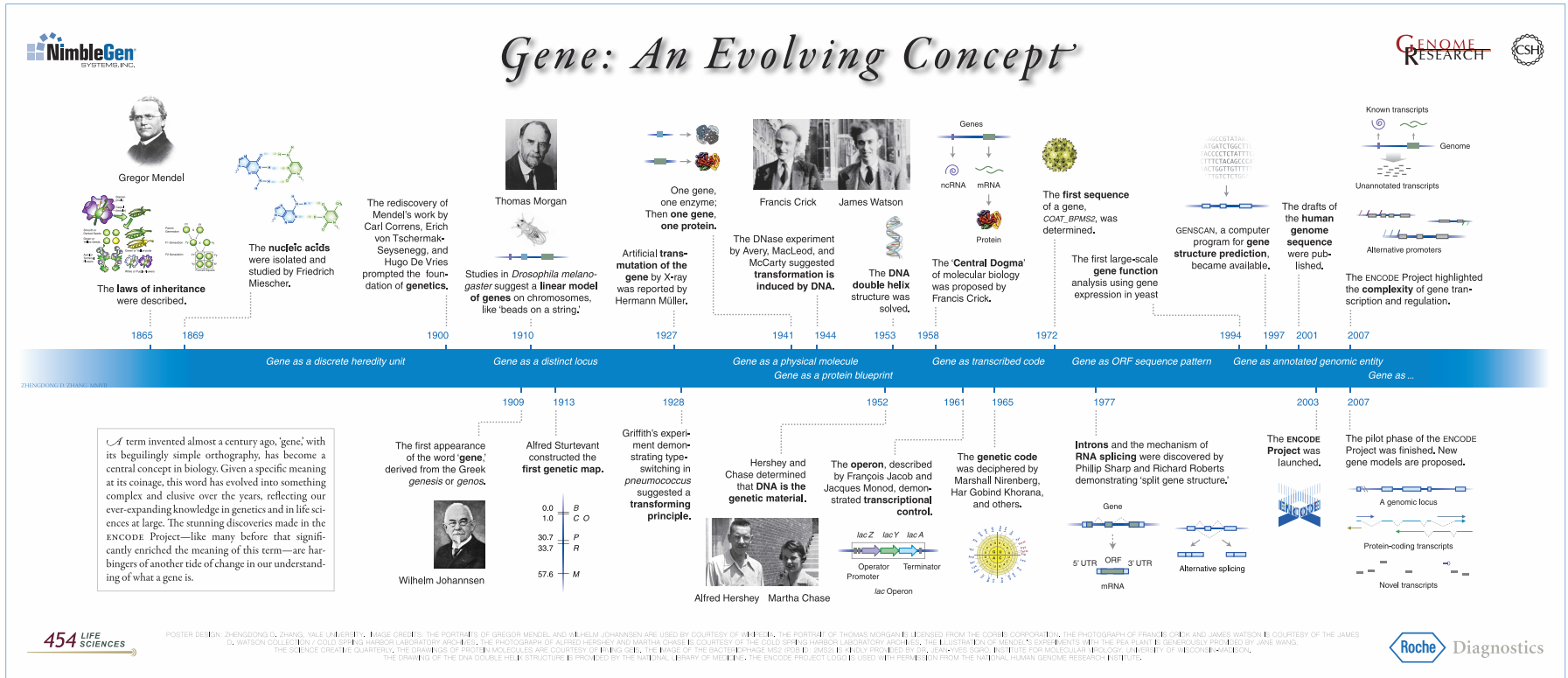
<http://eagereyes.org/>

Can we use parallel coordinates to show how

- analysis parameters
- execution settings

influence analysis results?

Workflow evolution visualization



- Can we use timelines to show how a workflow has been changed?
- What components were introduced/removed and when/why?

Questions?

<http://www.molgenis.org>

<http://www.molgenis.org/wiki/ComputeStart>

h.v.byelas@gmail.com m.a.swertz@gmail.com

H. Byelas and M. Swertz, “Visualization of bioinformatics

[molgenis_apps / doc / compute / 02_compute_imputation.md](#) 



mswertz 2 hours ago re-added imputation docs

1 contributor



file | 304 lines (226 sloc) | 16.546 kb

Imputation pipeline

This manual explains how one can do imputation. The analysis is efficient, it's needed to decide the

- *Preparing the reference:* here the reference is prepared
- *Preparing and QCing the study data:* all data is QCed, chunks the study data in a user specified number of SNPs. This extensive chunking takes 10 hours per chunk of 2000 SNPs and
- *Phasing:* phases the data using MaCH
- *Imputation:* consists of imputing the phased data

[molgenis_apps / doc / compute / 03_compute_ngs.md](#) 



mswertz an hour ago minor fixes to docs; added README.md

1 contributor



file | 193 lines (131 sloc) | 10.77 kb

Edit

Next-generation sequencing pipeline

Next-generation sequencing methods produce a growing volume of data, leading to increasing difficulties in analysis. This manual describes how one can simplify, parallelize and distribute such analysis across high performance computing resources using a standardized pipeline and the [Molgenis Compute](#) framework.

The pipeline is comprised of best-practice open-source software packages used in multiple institutions leading to high performance. The four main parts of the pipeline are:

- *Alignment:* here alignment is performed using Burrows-Wheeler Aligner [BWA](#). The produced [SAM](#) file is converted to [BAM](#)