Molgenis/compute

12/10/2012

UMCG, Groningen

Freerk van Dijk and George Byelas

Content

- HelloWorld example and demo
- Advanced compute features
  - Headers, software set-up, data transfer
- Library of workflows
  - NGS
  - Imputations
- Current developments and conclusion

HelloWorld

- I would like to send a wedding invitation to my friends and also messages to wedding organizers

```
Hello Abel,
We invite you for our wedding.
```

- and

```
Dear Oscar,
Please organize activities for the third group.
List of guests:
      Marina
      Claudia
```

How to automate it?

```
Hello Abel,
We invite you for our wedding.
```

- I'd like to use templates

```
Hello ${person},
We invite you for our ${event}.
```

- Template parameter

```
event = wedding
```

- Template parameter to iterate over

```
person = Abel, Adam, Adri
```

- I need to send different types of messages
  - create workflow

HelloWorld (1)

- To do it with compute, I need to specify 4 inputs:
  - **Workflow** is an ordered list of steps, e.g.
    - Write invitation
    - Contact organizers
  - A list of **protocols** - Freemarker templates
  - A list of **parameters** used in protocols
  - A list of **targets**, e.g. persons

HelloWorld (2)

- Workflow description in workflow.csv
  - consists of only one workflow element

```
name              , protocol_name   , PreviousSteps_name
GuestInvitation, GuestInvitation,
```

- Protocol listing in GuestInvitation.ftl

```
#FOREACH guest

echo "Hello ${guest},"
echo "We invite you for our ${event}."
```

HelloWorld (3)

- List of parameters in parameters.csv

```
Name , defaultValue, hasOne_name
guest,               ,
event, wedding       ,
```

- List of targets in worksheet.csv

```
guest
Charly
Cindy
Abel
Adam
Adri
```

HelloWorld (4)

- To run it from a command-line

```
sh molgenis_compute.sh \
-input=helloworld \
-id=invitation01
```

- Five scripts are generated

```
#FOREACH guest

echo "Hello Abel,"
echo "We invite you for our wedding."
```

HelloWorld (5)

- Extended workflow file (2 steps) workflow.csv

```
name,              protocol_name   , PreviousSteps_name
GuestInvitation, GuestInvitation,
OrganizerLetter, OrganizerLetter, GuestInvitation
```

- Extended target list

```
guest , age_group
Charly, child
Cindy , child
Abel  , adult
Adam  , adult
Adri  , adult
```

HelloWorld (6)

- Extended parameter list 1

```
Name, defaultValue, hasOne_name
guest,              ,
event, wedding      ,
age_group,          ,
```

- Extended parameter list 2

```
Name, defaultValue, hasOne_name
guest,              ,
event, wedding      ,
age_group,          , organizer
organizer,          ,
```

HelloWorld (7)

- Extended target list

```
guest , age_group, organizer
Charly, child    , Oscar
Cindy , child    , Oscar
Abel  , adult    , Otto
Adam  , adult    , Otto
Adri  , adult    , Otto
```

HelloWorld (8)

- Protocol template for organizer letter

```
#FOREACH age_group

echo "Dear ${organizer},"
echo "Please organize activities for the {age_group}
group."
echo "List of guests:"
<#list guest as g>
      echo "${g}"
</#list>
```

HelloWorld (8)

- Here, the guest list is folded for OrganizerLetter protocol

```
guest                , age_group, organizer
[Charly, Cindy]      , child    , Oscar
[Abel, Adam, Adri], adult,    , Otto
```

- Generated letter

```
#FOREACH age_group

echo "Dear Oscar,"
echo "Please organize activities for the child group."
echo "List of guests:"
      echo "Charly"
      echo "Cindy"
```

Demo

- Running HelloWorld on the localhost with DB and "pilot" submission
  - start molgenis to
    - generate database
    - start molgenis pilot service
  - import workflow into the database
  - generate jobs
  - execute jobs with pilot submission

  - From wiki: pilot submission is a scheduling technique, where a resource is acquired by an application so that the application can schedule work into that resource directly, rather than going through a local job scheduler.

Molgenis/compute

- Deployment and running in 5 scripts + **Tutorial**

1. Check out from Git (git clone)

```
git clone https://github.com/georgebyelas/molgenis.git
git clone https://github.com/georgebyelas/molgenis_apps.git
```

2. Build with Ant

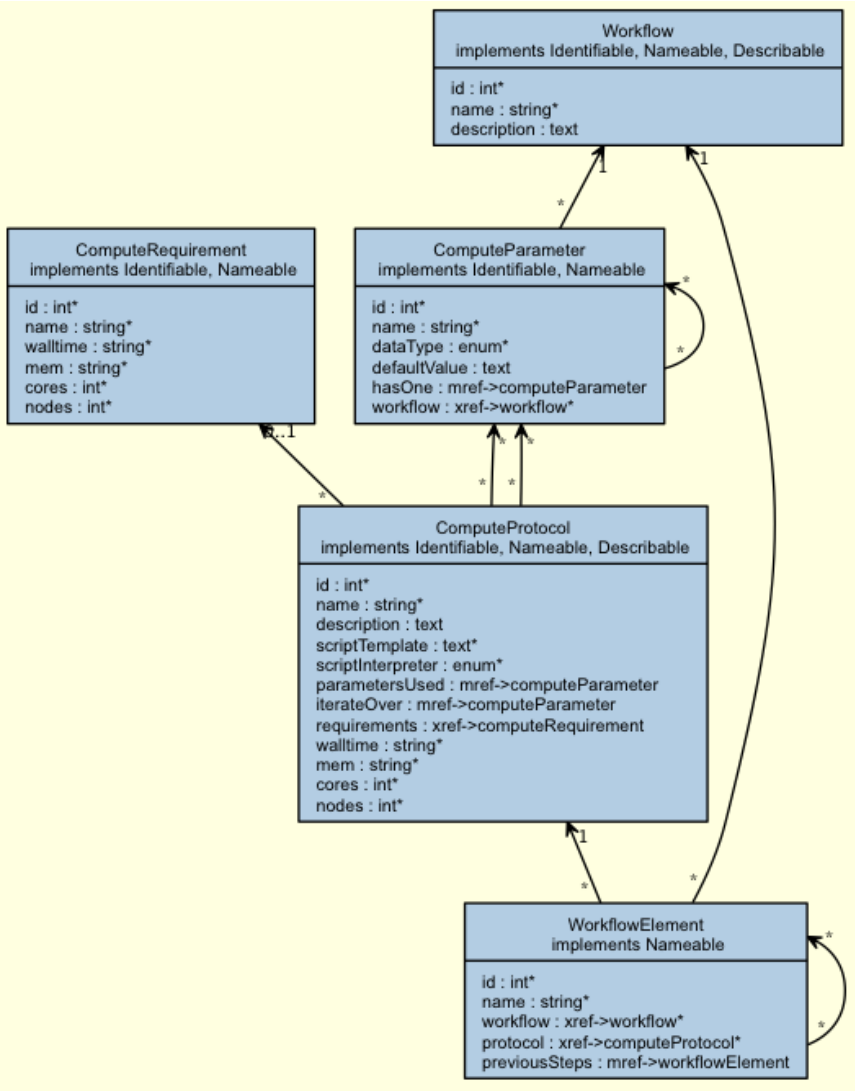
```
ant -f build_compute.xml clean-generate-compile
```

- We are planning to have releases periodically

3. Import workflow from files
   - workflow, protocols, parameters
4. Import worksheet and generate ComputeTasks
5. Run workflow with
   - pilots and DB or
   - submission script for PBS

Compute Database

- mysql
- Workflow can be
  - imported from files or
  - added through UI



Pilot listing

```
export WORKDIR=$TMPDIR
source dataTransferSRM.sh

curl  -F status=started
http://molgenis15.target.rug.nl:8080/ \
compute/api/pilot > script.sh

sh script.sh 2>&1 | tee -a log.log

curl -F status=done -F log_file=@log.log
http://molgenis15.target.rug.nl:8080/ \
compute/api/pilot
```

Protocol example

- compute aims to run computationally intense analyses on diverse and large data in heterogeneous

```
#MOLGENIS walltime=15:00:00 nodes=1 cores=4 mem=6
#FOREACH

module load bwa/${bwaVersion}

getFile ${indexfile}
getFile ${leftbarcodefqgz}

bwa aln \
${indexfile} \
${leftbarcodefqgz} \
-t ${bwaaligncores} \
-f ${leftbwaout} \

putFile ${leftbwaout}
```

header

tool management

data management

template of the actual analysis

data management

molgenis

Compute protocol detail (1)

- Molgenis header

```
#MOLGENIS walltime=hh:mm:ss nodes=n cores=c mem=m
```

- Modules available now

```
bwa/0.5.8c patched
capturing_kits/SureSelect_All_Exon_30MB_V2
capturing_kits/SureSelect_All_Exon_50MB
capturing_kits/SureSelect _All _Exon _G3362
fastqc/v0.7.0
fastqc/v0.10.1
gtool/v0.7.5_x86_64
impute/v2.2.2_x86_64_static
jdk/1.6.0_33
```

```
module load plink/1.07-x86_64
```

Compute

- getFile
  - are
  - now
    -
    -
- Input

```
getF:
```

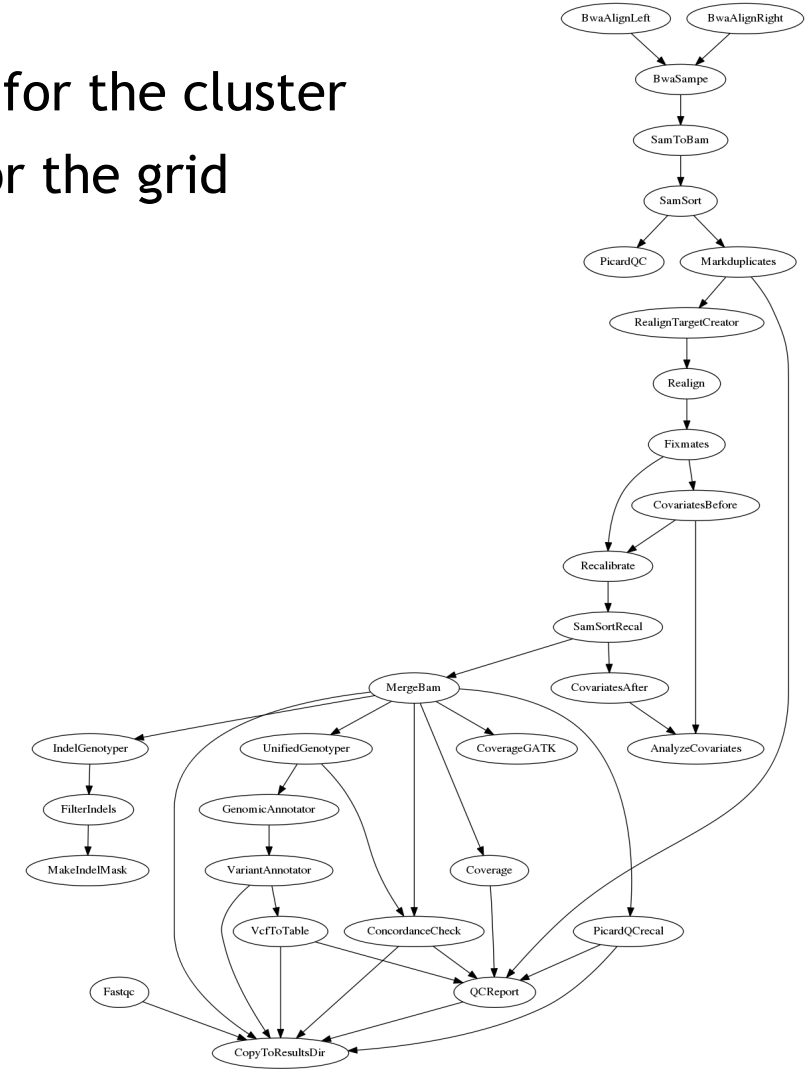
- Gener

```
getF:
imput
chr2(
```

```bash
#!/bin/bash

getRemoteLocation()
{
    ARGS=($@)
    myFile=${ARGS[0]}
    remoteFile=srm://srm.grid.sara.nl/pnfs/grid.sara.nl/data/bbmri.nl/byelas${myFile:`expr length $TMPDIR`}
    echo $remoteFile
}

getFile()
{
    ARGS=($@)
    NUMBER="${#ARGS[@]}";
    if [ "$NUMBER" -eq "1" ]
    then

    myFile=${ARGS[0]}
    remoteFile=`getRemoteLocation $myFile`

    # 1. myPath = getPath( myFile ) will strip off the file name and return the path
    mkdir -p $(dirname "$myFile")

    # 2. cp srm:.../remoteFile myFile
    echo "srmcp -server_mode=passive $remoteFile file:///$myFile"
    srmcp -server_mode=passive $remoteFile file:///$myFile
    chmod 755 $myFile

    else
    echo "Example usage: getData \"\$TMPDIR/datadir/myfile.txt\""
    fi
}

putFile()
{
    ARGS=($@)
    NUMBER="${#ARGS[@]}";
    if [ "$NUMBER" -eq "1" ]
    then
    myFile=${ARGS[0]}
    remoteFile=`getRemoteLocation $myFile`
    echo "srmcp -server_mode=passive file:///$myFile $remoteFile"
    srmcp -server_mode=passive file:///$myFile $remoteFile
    else
    echo "Example usage: getData \"\$TMPDIR/datadir/myfile.txt\""
    fi
}

export -f getRemoteLocation
export -f getFile
export -f putFile
```

Workflows

- NGS alignment
  - can have 20-30 steps – ready for the cluster
  - 6 essential steps are ready for the grid
    - FastQC
    - BwaAlign
    - BwaSampe
    - SamToBam
    - SamSort
    - PicardQC



NGS

Alignment per lane	1 sample consists of 3 lanes			
Step	**Cores**	**Ram (Gb)**	**disk**	**Average runtime (hrs)**
al00.fastqc	1	0.5	20 – 50Gb	6.336238616
al01.bwa_align_pair1	4	6		6.445865209
al02.bwa_align_pair2	4	6		6.445865209
al03.bwa_sampe	1	4		5.701997571
al04.sam_to_bam	1	4		1.126662113
al05.sam_sort	1	3		5.717598664
al06.picardQC	1	4		2.424761688
al07.mark_duplicates	1	4		2.364496053
al08.realign	1	10		4.11407711
al09.fixmates	1	6		2.91533394
al10.covariates_before	1	4		9.410815118
al11.recalibrate	1	4		6.387997875
al12.sam_sort	1	3		4.667501543
al13.covariates_after	1	4		10.23987346
al14.analyze_covariates	1	4		0.01349563
totals				74.3125798
SNP calllng per sample				
vc00.merge	1	6		22
vc01.unified_genotyper	4	8		6.5
vc02.picardQC	1	4		9
vc03.coverage	4	10		6
contamination checker	1	4		4
totals			466Gb	43.5
De novo assembly per sample				
abyss	6	140		100

Workflows (2)

- Imputation with 3 different workflows
  - impute2 – 9 steps ready for the cluster and grid
  - beagle - under development
  - minimac – under development

  - reference data creation – several steps, ready for the cluster and grid

- NGS/imputation workflows can be found at molgenis git
  - https://github.com/molgenis/molgenis_apps/tree/master/modules/compute
    - /workflows
    - /protocols

Imputation

- Impute2 analysis in parallel based on 'bins' of the Genome
  - 616 jobs per 1000 samples
  - Each job takes 50 hours, 5-8 GB of memory and generating 30 GB of data
  - One reference set for example 1000GP = 320GB
  - Data of all samples in bed bim fam (~150MB for 1000 samples and 500k snps).
- Number of samples:

# Samples	Chip	Cohort
24000	CytoSNPv2	LifeLines
5474	illumina 550k	Rotterdam study
416	affy500, illumina 550k, illumina OMNI 5M	Rotterdam study
3000	affy 250k, illumina 318k, 350k and/or 610k	Rotterdam Rucphen
10000	1 out of 5 different chips	Amsterdam NTR

Current developments

Compute Testing

molgenis

| About | Object model | R-project API | HTTP API | REST API | Web Services API

Compute

- ComputeParameters
- ComputeProtocols
- Workflow
- ComputeTasks
- ComputeHosts

JobManager

File Edit View

Workflow

1 of 1

Search:

id: 1

name *: workflowImpute.csv

description:

WorkflowElement

WorkflowElement

File Edit View

1 - 9 of 9

Search:

	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	convertConcatedImpute2ResultToPedMap.ftl	impute2_s05
8.	8	impute2_s07	workflowImpute.csv	calculateBeagleR2ForImpute2Results.ftl	impute2_s06
9.	9	impute2_s08	workflowImpute.csv	convertResultPedMapToB36.ftl	impute2_s06

* = this record is readonly.

This database was generated using the open source MOLGENIS database generator version 4.0.0-testing.
Please cite Swertz et al (2010) and Arends & van der Velde et al (2012) on use.

Current developments (compute task view)

Compute Testing

molgenis

| About | Object model | R-project API | HTTP API | REST API | Web Services API

Compute

- ComputeParameters
- ComputeProtocols
- Workflow
- ComputeTasks
- ComputeHosts

JobManager

File • Edit • View •

ComputeTasks

|◀◀ ◀◀ 1 - 20 of 21 ▶▶ ▶▶|

Search:

	id ▼	name	ComputeScript	RunLog	WorkflowElement	Interpreter	PrevSteps	requirements	StatusCode
1.	1	impute2_s00_test1_1349185453555253000	##### BEFORE ##### touch $PBS_O_WORKDIR/jobname.out source $WORKDIR/tools/scripts/import.sh befo...	TASKID:impute2_s00_test1_1349185453555253000 touch: cannot touch `/opt/glite/var/tmp/jobname.out': N...	impute2_s00	bash		1	done
2.	2	impute2_s01_test1_1349185453792052000	##### BEFORE ##### touch $PBS_O_WORKDIR/jobname.out source $WORKDIR/tools/scripts/import.sh befo...	TASKID:impute2_s01_test1_1349185453792052000 touch: cannot touch `/opt/glite/var/tmp/jobname.out': N...	impute2_s01	bash	impute2_s00_test1_1349185453555253000	1	done
3.	3	impute2_s02_test1_1349185453886012000	##### BEFORE ##### touch $PBS_O_WORKDIR/jobname.out source $WORKDIR/tools/scripts/import.sh befo...	TASKID:impute2_s02_test1_1349185453886012000 touch: cannot touch `/opt/glite/var/tmp/jobname.out': N...	impute2_s02	bash	impute2_s01_test1_1349185453792052000	1	done
4.	4	impute2_s03_test1_1349185454034169000	##### BEFORE ##### touch $PBS_O_WORKDIR/jobname.out source $WORKDIR/tools/scripts/import.sh befo...	TASKID:impute2_s03_test1_1349185454034169000 touch: cannot touch `/opt/glite/var/tmp/jobname.out': N...	impute2_s03	bash	impute2_s02_test1_1349185453886012000	1	done
5.	5	impute2_s04_test1_1349185454255967000	##### BEFORE ##### touch $PBS_O_WORKDIR/jobname.out source $WORKDIR/tools/scripts/import.sh befo...	TASKID:impute2_s04_test1_1349185454255967000 touch: cannot touch `/opt/glite/var/tmp/jobname.out': N...	impute2_s04	bash	impute2_s03_test1_1349185454034169000	1	done
6.	6	impute2_s04_test1_1349185454299992000	##### BEFORE ##### touch $PBS_O_WORKDIR/jobname.out source $WORKDIR/tools/scripts/import.sh befo...	TASKID:impute2_s04_test1_1349185454299992000 touch: cannot touch `/opt/glite/var/tmp/jobname.out': N...	impute2_s04	bash	impute2_s03_test1_1349185454034169000	1	done
7.	7	impute2_s04_test1_1349185454357872000	##### BEFORE ##### touch $PBS_O_WORKDIR/jobname.out source $WORKDIR/tools/scripts/import.sh befo...	TASKID:impute2_s04_test1_1349185454357872000 touch: cannot touch `/opt/glite/var/tmp/jobname.out': N...	impute2_s04	bash	impute2_s03_test1_1349185454034169000	1	done
8.	8	impute2_s04_test1_1349185454390958000	##### BEFORE ##### touch $PBS_O_WORKDIR/jobname.out source $WORKDIR/tools/scripts/import.sh befo...	TASKID:impute2_s04_test1_1349185454390958000 touch: cannot touch `/opt/glite/var/tmp/jobname.out': N...	impute2_s04	bash	impute2_s03_test1_1349185454034169000	1	done
9.	9	impute2_s04_test1_1349185454415059000	##### BEFORE ##### touch $PBS_O_WORKDIR/jobname.out source $WORKDIR/tools/scripts/import.sh befo...	TASKID:impute2_s04_test1_1349185454415059000 touch: cannot touch `/opt/glite/var/tmp/jobname.out': N...	impute2_s04	bash	impute2_s03_test1_1349185454034169000	1	done
10.	10	impute2_s04_test1_1349185454434245000	##### BEFORE ##### touch $PBS_O_WORKDIR/jobname.out source $WORKDIR/tools/scripts/import.sh befo...	TASKID:impute2_s04_test1_1349185454434245000 touch: cannot touch `/opt/glite/var/tmp/jobname.out': N...	impute2_s04	bash	impute2_s03_test1_1349185454034169000	1	done
11.	11	impute2_s04_test1_1349185454459357000	##### BEFORE ##### touch $PBS_O_WORKDIR/jobname.out source $WORKDIR/tools/scripts/import.sh befo...	TASKID:impute2_s04_test1_1349185454459357000 touch: cannot touch `/opt/glite/var/tmp/jobname.out': N...	impute2_s04	bash	impute2_s03_test1_1349185454034169000	1	done
12.	12	impute2_s04_test1_1349185454559252000	##### BEFORE ##### touch $PBS_O_WORKDIR/jobname.out source $WORKDIR/tools/scripts/import.sh befo...	TASKID:impute2_s04_test1_1349185454559252000 touch: cannot touch `/opt/glite/var/tmp/jobname.out': N...	impute2_s04	bash	impute2_s03_test1_1349185454034169000	1	done
13.	13	impute2_s04_test1_1349185454601338000	##### BEFORE ##### touch $PBS_O_WORKDIR/jobname.out source $WORKDIR/tools/scripts/import.sh befo...	TASKID:impute2_s04_test1_1349185454601338000 touch: cannot touch `/opt/glite/var/tmp/jobname.out': N...	impute2_s04	bash	impute2_s03_test1_1349185454034169000	1	done
14.	14	impute2_s04_test1_1349185454641825000	##### BEFORE ##### touch $PBS_O_WORKDIR/jobname.out source $WORKDIR/tools/scripts/import.sh befo...	TASKID:impute2_s04_test1_1349185454641825000 touch: cannot touch `/opt/glite/var/tmp/jobname.out': N...	impute2_s04	bash	impute2_s03_test1_1349185454034169000	1	done
15.	15	impute2_s04_test1_1349185454728261000	##### BEFORE ##### touch $PBS_O_WORKDIR/jobname.out source $WORKDIR/tools/scripts/import.sh befo...	TASKID:impute2_s04_test1_1349185454728261000 touch: cannot touch `/opt/glite/var/tmp/jobname.out': N...	impute2_s04	bash	impute2_s03_test1_1349185454034169000	1	done
16.	16	impute2_s04_test1_1349185454782979000	##### BEFORE ##### touch $PBS_O_WORKDIR/jobname.out source $WORKDIR/tools/scripts/import.sh befo...	TASKID:impute2_s04_test1_1349185454782979000 touch: cannot touch `/opt/glite/var/tmp/jobname.out': N...	impute2_s04	bash	impute2_s03_test1_1349185454034169000	1	done
17.	17	impute2_s04_test1_1349185454882913000	##### BEFORE ##### touch $PBS_O_WORKDIR/jobname.out source $WORKDIR/tools/scripts/import.sh befo...	TASKID:impute2_s04_test1_1349185454882913000 touch: cannot touch `/opt/glite/var/tmp/jobname.out': N...	impute2_s04	bash	impute2_s03_test1_1349185454034169000	1	done
18.	18	impute2_s05_test1_1349185454927919000	##### BEFORE ##### touch $PBS_O_WORKDIR/jobname.out source $WORKDIR/tools/scripts/import.sh befor...	TASKID:impute2_s05_test1_1349185454927919000 touch: cannot touch `/opt/glite/var/tmp/jobname.out': N...	impute2_s05	bash	impute2_s04_test1_1349185454882913000	1	done
19.	19	impute2_s06_test1_1349185455059615000	##### BEFORE ##### touch $PBS_O_WORKDIR/jobname.out source $WORKDIR/tools/scripts/import.sh befor...	TASKID:impute2_s06_test1_1349185455059615000 touch: cannot touch `/opt/glite/var/tmp/jobname.out': N...	impute2_s06	bash	impute2_s05_test1_1349185454927919000	1	done
20.	20	impute2_s07_test1_1349185455113243000	##### BEFORE ##### touch $PBS_O_WORKDIR/jobname.out source $WORKDIR/tools/scripts/import.sh befor...	TASKID:impute2_s07_test1_1349185455113243000 touch: cannot touch `/opt/glite/var/tmp/jobname.out': N...	impute2_s07	bash	impute2_s06_test1_1349185455059615000	1	done

* = this record is readonly.

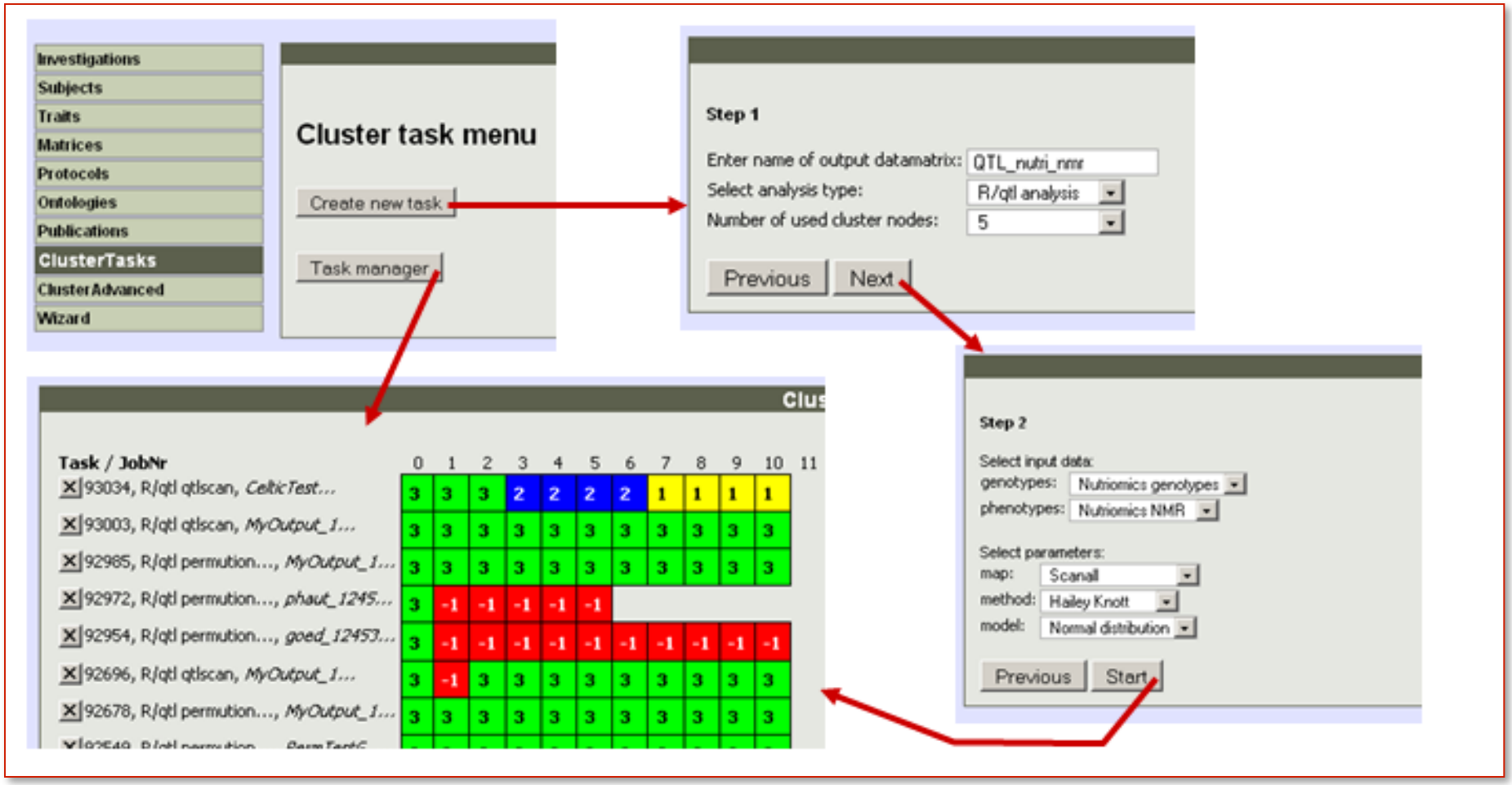
This database was generated using the open source MOLGENIS database generator version 4.0.0-testing.
Please cite Swertz et al (2010) and Arends & van der Velde et al (2012) on use.

university of groningen / genomics coordination center

umcg

molgenis

User interface to choose workflows, set parameters/ targets, and run…



Conclusion

- We do not aim to create
  - a new workflow model or
  - a new pilot submission strategy

- We do like to help non-technical people to
  - specify and
  - run complex analyses

Thank you for listening. Questions?

GCC/compute

- Ger Strikwerda
- Marcel Burger
- Martijn Dijkstra
- Morris Swertz
- Wil Bruins-Koetsier

eBioGrid

- David van Enckevort
- Irene Nooren
- Jan Bot
- Mathijs Kattenberg
- Pieter Neerincx
- Tom Visser

And you, our local, national and international collaborators

university of groningen / genomics coordination center umcg molgenis