Molgenis Compute based pipelines

Freerk van Dijk

Genomics Coordination Center
University Medical Center Groningen
2014-05-16
freerk.van.dijk@gmail.com

umcg

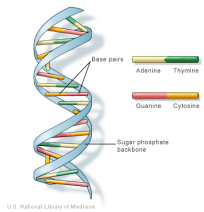
molgenis

Content

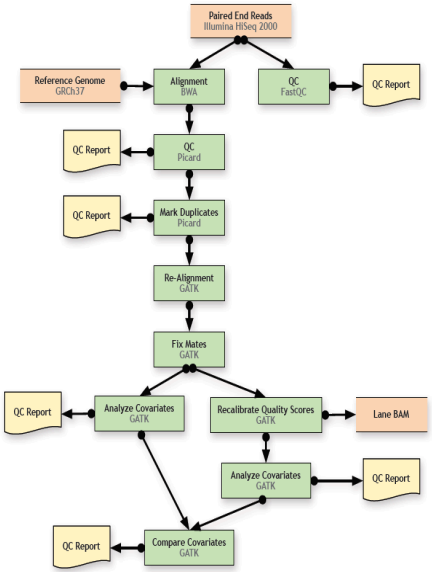
- General background
- Molgenis Compute
  - Commandline/Cluster
  - Grid
  - Cloud
- Pipelines
- LIMS
- Future work

General background

General background / example: NGS alignment workflow



HiSeq



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

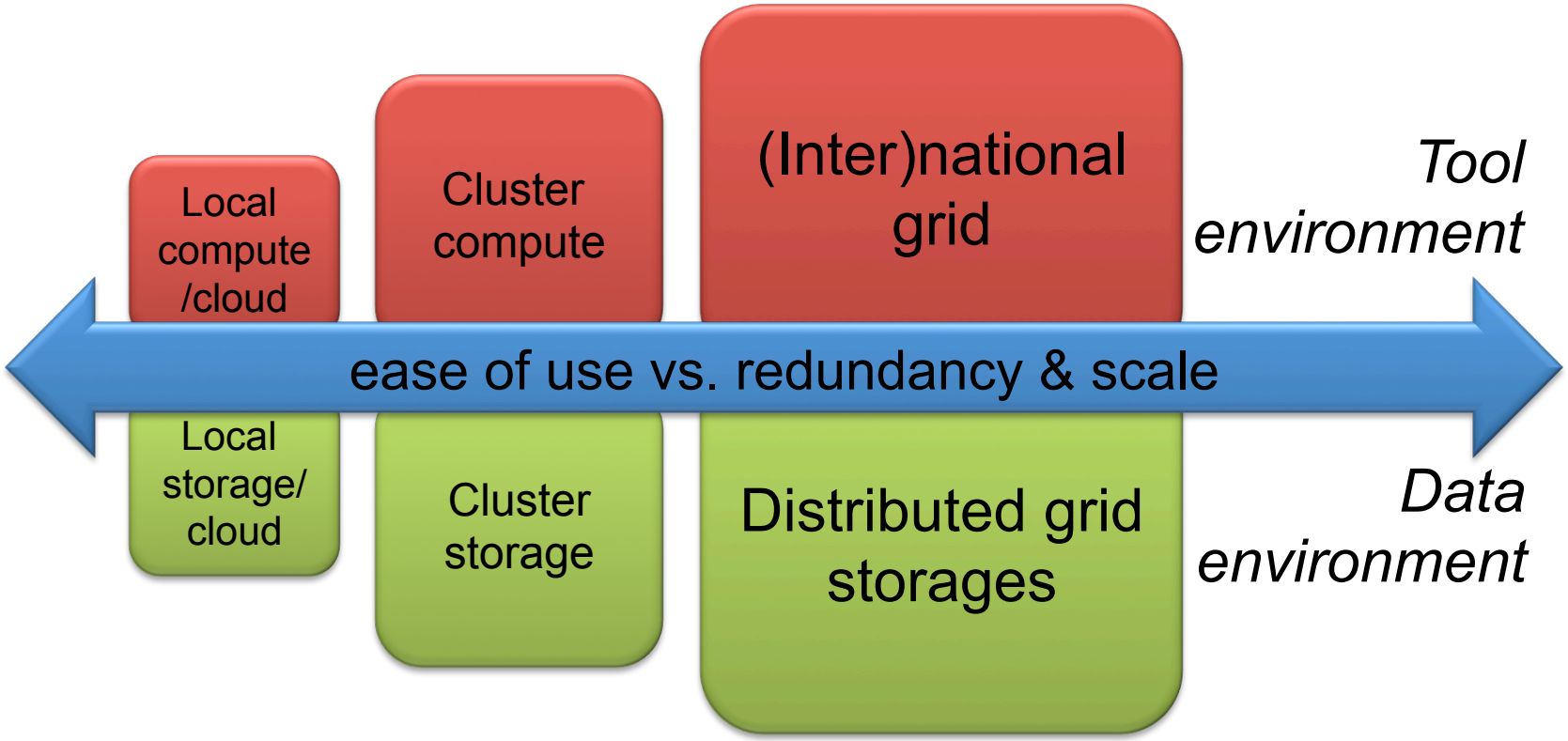
Raw data → ***alignment workflow*** → Result data

10-100 samples

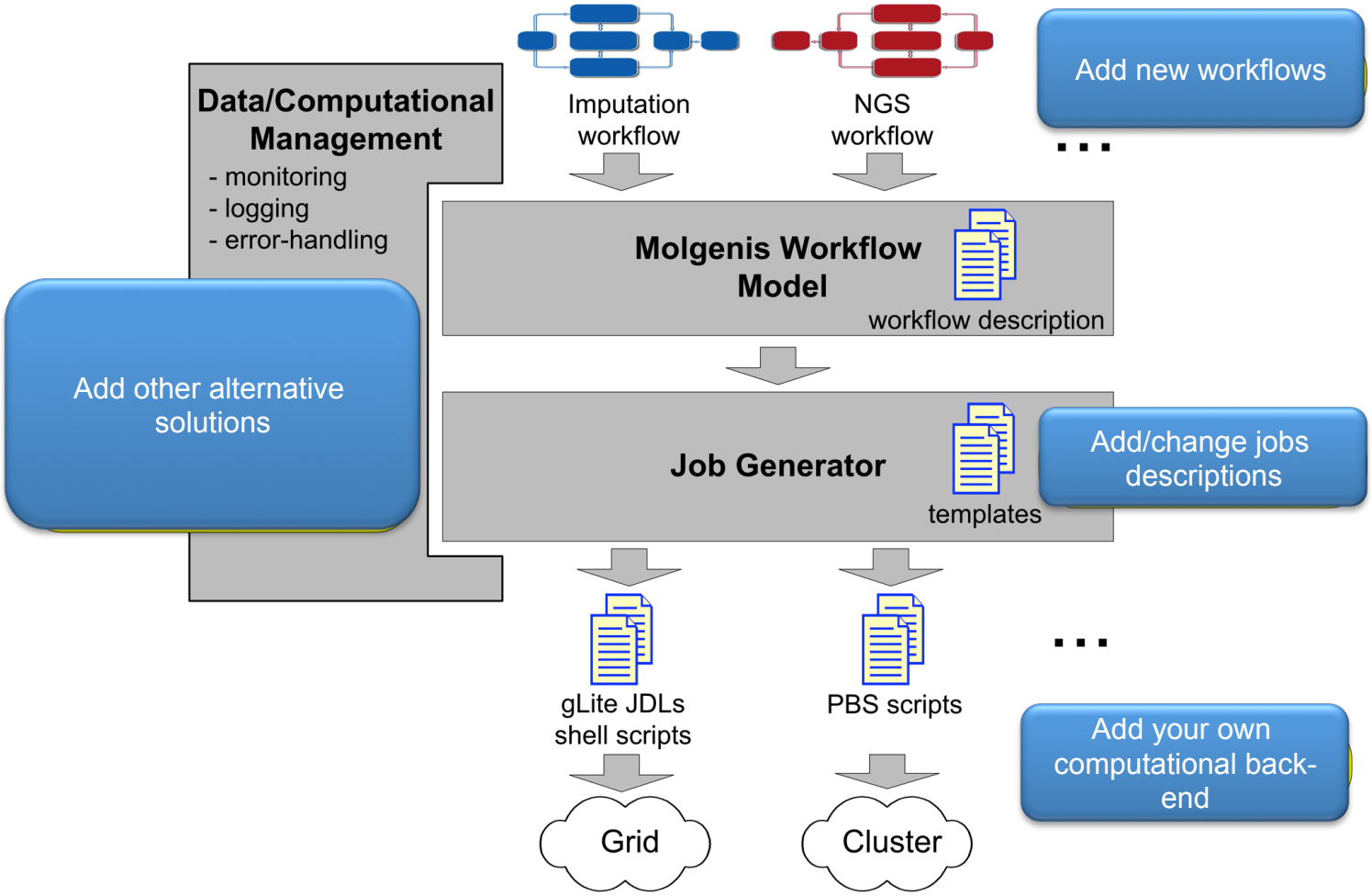
20 – 200 days

80 – 800 GB

Computational environments



Command-line generator



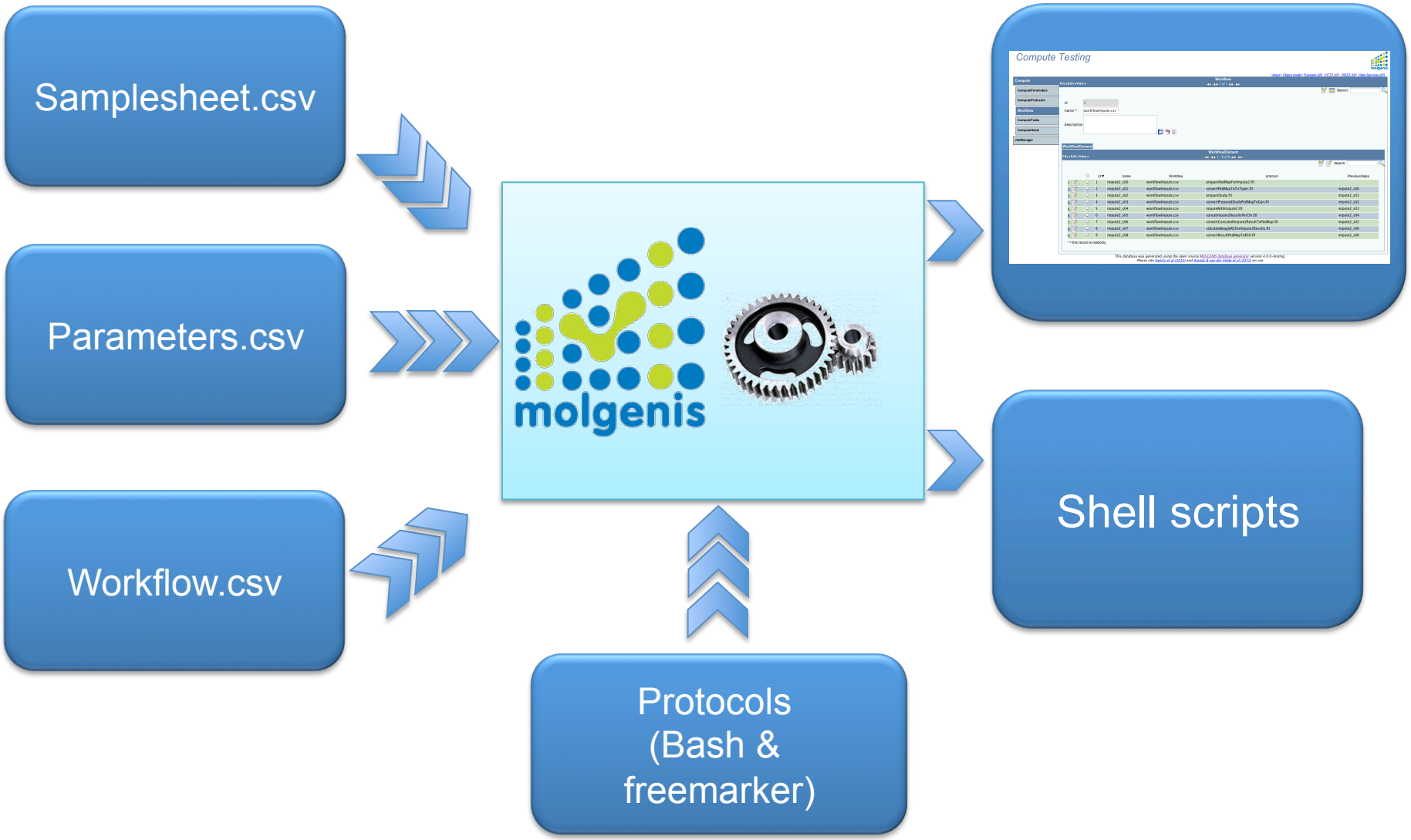
- Generates jobs (scripts) from model described in files
- Suitable for workflows (PBS cluster) and single jobs (gLite grid)

Molgenis Compute commandline

umcg

molgenis

Compute overview



Samplesheet.csv

ID	sampleID	Project	Machine	SequencingDate	Run	Flowcell	Lane	Barcode	SequencingT	Manufacture	Enrichmentkit
3	Sample3_2	Testproject2	SN163	130809	501	BD2BWCACXX	4	RPI 01 ATCACG	PE	Agilent	SureSelect_All_Exon_50MB_v5.1
4	Sample5_2	Testproject2	SN163	130809	501	BD2BWCACXX	4	RPI 04 TGACCA	PE	Agilent	SureSelect_All_Exon_50MB_v5.1
5	Sample5_2	Testproject2	SN163	130809	501	BD2BWCACXX	5	RPI 04 TGACCA	PE	Agilent	SureSelect_All_Exon_50MB_v5.1
6	Sample4_2	Testproject2	SN163	130809	501	BD2BWCACXX	4	RPI 08 ACTTGA	PE	Agilent	SureSelect_All_Exon_50MB_v5.1

Parameters.csv

Samplesheet.csv

Parameters.csv

queue	devel
mem	4
walltime	23:59:00
nodes	1
ppn	1
defaultInterprete	#!/bin/bash
stage	module load
checkStage	module list
WORKDIR	/gcc/
root	${WORKDIR}
tempDir	${WORKDIR}/groups/gaf/tmp02/tmp/
resDir	${root}/resources/
toolDir	${root}/tools/
scriptDir	${toolDir}/scripts/
gafHome	${root}/groups/gaf/
tmpDataDir	${root}/groups/gaf/tmp02/
allRawtmpDataDi	${tmpDataDir}/rawdata/
allRawNgstmpDat	${allRawtmpDataDir}/ngs/
allRunDemultiple	${tmpDataDir}/projects/${project}/${run}
runPrefix	${sequencingStartDate}_${sequencer}_${run}_${flowcell}

umcg

molgenis

Workflow.csv

step	protocol	dependencies
FastQC	protocols/FastQC.sh	
BwaAlign	protocols/BwaAlign.sh	
SamToBam	protocols/SamToBam.sh	BwaAlign
SortBam	protocols/SortBam.sh	SamToBam;inputSortBam=alignedBam;tmpSortedBam=tmpAlignedSortedBam;tmpSortedBamIdx=tmpAlignedSortedBamIdx;sc
MergeBam	protocols/MergeBam.sh	SortBam;inputMergeBam=alignedSortedBam;inputMergeBamIdx=alignedSortedBamIdx;tmpMergedBam=tmpSampleMergedB
MarkDuplica	protocols/MarkDuplicates.sh	MergeBam

Parameters.csv

Workflow.csv

Example snippet SortBam.sh:

```
#string inputSortBam
#string tmpSortedBam
#string tmpSortedBamIdx


#Run picard, sort BAM file and create index on the fly
java -jar -Xmx3g $PICARD_HOME/${sortSamJar} \
INPUT=${inputSortBam} \
OUTPUT=${tmpSortedBam} \
SORT_ORDER=coordinate \
CREATE_INDEX=true \
VALIDATION_STRINGENCY=LENIENT \
MAX_RECORDS_IN_RAM=2000000 \
TMP_DIR=${tempDir}
```

Protocol structure in more detail

```
//header
      #MOLGENIS walltime=15:00 nodes=1 cores=4 mem=6gb
      #list inputMergeBam
      #string outputMergedBam
//tool management
      module load bwa/${bwaVersion}
//data management
      getFile ${indexfile}
      getFile ${leftbarcodefqgz}
//template of the actual analysis
      bwa aln \
      ${indexfile} ${leftbarcodefqgz} \
      -t ${bwaaligncores} -f ${leftbwaout}
//data management
      putFile ${leftbwaout}
```

umcg

molgenis

Data transfer

- getFile and p
  - are back-e
  - now, we
    - check i
    - do srm/
- Input

```
getFile $
```

- Generated o

```
getFile $V
imputatio
chr20.map
```

```
#------- data transfer

getRemoteLocation()
{
    ARGS=($@)
    myFile=${ARGS[0]}
    remoteFile=srm://srm.grid.sara.nl/pnfs/grid.sara.nl/data/bbmri.nl/RP2${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 "srmrm $remoteFile"
    srmrm $remoteFile
    echo "srmcp -server_mode=passive file:///$myFile $remoteFile"
    srmcp -server_mode=passive file:///$myFile $remoteFile
        returnCode=$?

        echo "srmcopy: ${returnCode}"

        if [ $returnCode -ne 0 ]
        then
            exit 1
        fi
    else
    echo "Example usage: putData \"\$TMPDIR/datadir/myfile.txt\""
    fi
}
```

Generated back-end independent script

```
//header
#MOLGENIS walltime=15:00 nodes=1 cores=4 mem=6
//tool management
module load bwa/0.5.8c_patched
//data management
getFile $WORKDIR/resources/hg19/indices/human_g1k_v37.fa
getFile $WORKDIR/groups/gcc/projects/cardio/run01/rawdata/
121128_SN163_0484_AC1D3HACXX_L8_CAACCT_1.fq.gz
//template of the actual analysis
bwa aln \
human_g1k_v37.fa 121128_SN163_0484_AC1D3HACXX_L8_CAACCT_1.fq.gz -t 4 \
-f 121128_SN163_0484_AC1D3HACXX_L8_CAACCT_1.bwa_align.human_g1k_v37.sai
//data management
putFile $WORKDIR/groups/gcc/projects/cardio/run01/results/
121128_SN163_0484_AC1D3HACXX_L8_CAACCT_1.bwa_align.human_g1k_v37.sai
```

Best practices

- Always output to temporary files and check return code of software

```
#Get return code from last program call
returnCode=$?

echo -e "\nreturnCode MergeBam: $returnCode\n\n"

if [ $returnCode -eq 0 ]
then
    echo -e "\nMergedBam finished succesfull. Moving temp files to final.\n\n"
    mv ${tmpMergedBam} ${finalMergedBam}
else
    echo -e "\nFailed to move MergeBam results to ${finalMergedBam}\n\n"
    exit -1
fi
```

- Create md5sums

Submission in the cluster

- No dependencies

```
#
##alignment_21
#

# Skip this step if step finished already successfully
if [ -f alignment_21.sh.finished ]; then
        skip alignment_21.sh
        echo "Skipped alignment_21.sh"
else
        # Build dependency string
        dependenciesExist=false
        dependencies="-W depend=afterok"
        if ! $dependenciesExist; then
                unset dependencies
        fi

        alignment_21=$(qsub -N alignment_21 $dependencies alignment_21.sh)
        echo $alignment_21
        sleep 0
fi


#
##imputation_0
#

# Skip this step if step finished already successfully
if [ -f imputation_0.sh.finished ]; then
        skip imputation_0.sh
        echo "Skipped imputation_0.sh"
else
        # Build dependency string
        dependenciesExist=false
        dependencies="-W depend=afterok"
                if [[ -n "$alignment_0" ]]; then
                        dependenciesExist=true
                        dependencies="${dependencies}:$alignment_0"
                fi
        if ! $dependenciesExist; then
                unset dependencies
        fi

        imputation_0=$(qsub -N imputation_0 $dependencies imputation_0.sh)
        echo $imputation_0
        sleep 0
fi
```

- Dependencies

Molgenis Compute for the grid

umcg

molgenis

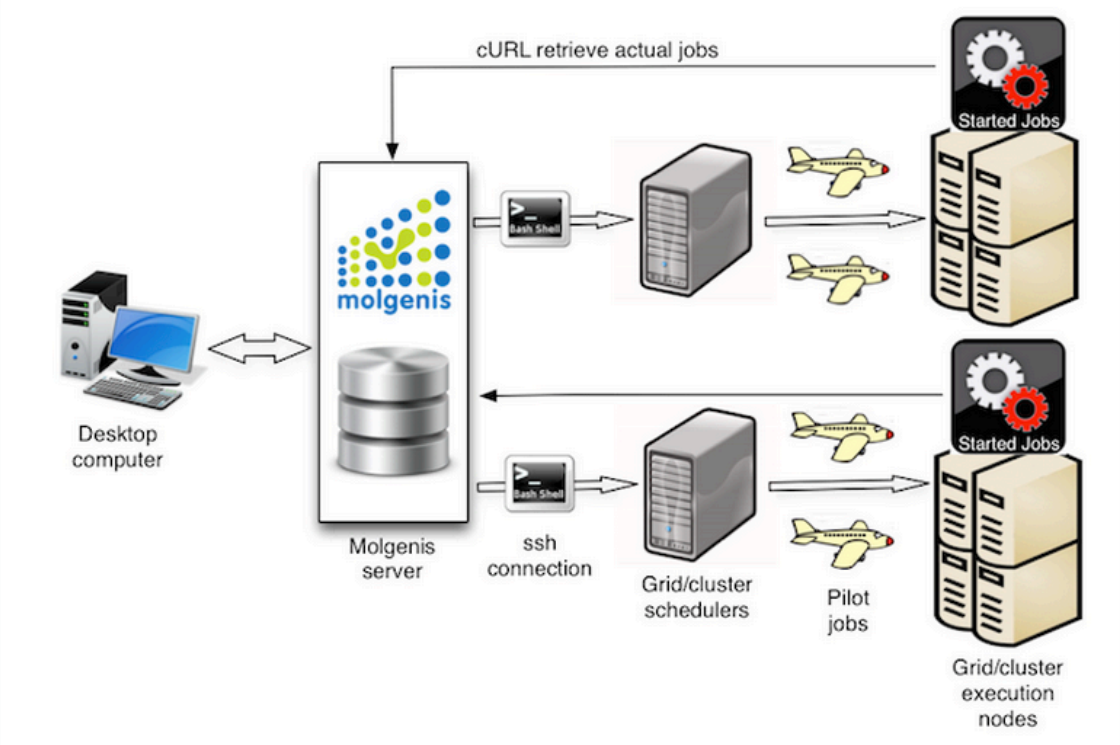
Login

molgenis

Home

Sign in

Welcome to Molgenis Compute5db Platinum Grid Edition

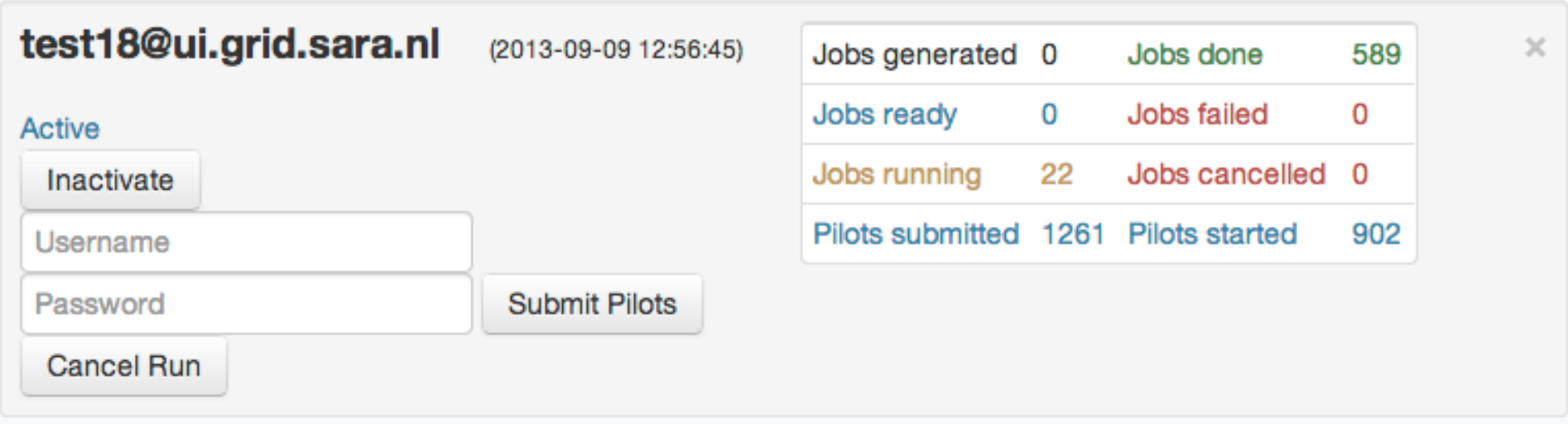


This database was generated using the open source MOLGENIS database generator version 4.0.0-testing. Please cite Swertz et al (2010) or Swertz & Jansen (2007) on use.

umcg

molgenis

Dashboard of one run in the grid



- Main features:
  - using pilot jobs (overview of pilots)
  - all running jobs can be cancelled (pilot asks server, if it is cancelled before the final data transfer)
  - all grid sites are synchronized using module system

Technical details of execution in the grid (1)

1. *Pilot* is sent to the scheduler from MOLGENIS *Server*

```
glite-wms-job-submit -d $USER … $HOME/maverick.jdl
```

2. *Pilot* asks the *Server* for *Job* todo

```
curl  … -F status=started -F backend=ui.grid.sara.nl \
http://$SERVER:8080/api/pilot > script.sh
```

3. *Pilot* starts *Job* in the computation element (CE) of the grid

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

Technical details of execution in the grid (2)

4. *Pilot* sends pulse and *Job's* update to *Server*

```
while [ 1 ] ; do
…
check_process "script.sh"
CHECK_RET=$?
if [ $CHECK_RET -eq 0 ];
then
        …
        curl … -F status=nopulse -F log_file=@inter.log …
elif
       …
        curl … -F status=pulse -F log_file=@inter.log …
```

Technical details of execution in the grid (3)

5. Job asks server, if it is cancelled before final data transfer

```
RESULT=‘curl ... -F pilotid=${pilotid} –F status=is_cancel \
http://$SERVER:8080/api/pilot‘ ...

if [ "$RESULT" = "not_cancelled" ];
then
      srmcp file:///$localFile $remoteFile
...
fi
```

MOLGENIS Compute for the cloud

umcg

molgenis

Dashboard for the grid and cloud runs

test01@ui.grid.sara.nl (2014-05-06 12:22:31)

Active

Inactivate

Username

Password

Submit Pilots

Cancel Run

Jobs generated	2	Jobs done	0
Jobs ready	0	Jobs failed	0
Jobs running	0	Jobs cancelled	0
Pilots submitted	0	Pilots started	0

test02@openstack01.gcc.rug.nl (2014-05-06 12:23:06)

Ready to run

Username

Password

Start

Jobs generated	0	Jobs done	0
Jobs ready	100	Jobs failed	0
Jobs running	0	Jobs cancelled	0

- Main features:
  - on demand virtual machine (VM) invocation, if jobs should be executed
  - Pre-installed VM's in OpenStack platform

Main steps in running in the cloud

1. Get token from the cloud **(keystone/rest)**
2. Select VM image to run analysis **(nova/rest)**
3. Start Server **(nova/rest)**
4. Associate Server with the external IP address **(nova/rest)**
5. Select and mount Disk to the Server **(nova/rest + ssh)**
6. Start analysis with **(ssh + curl)**
7. Analysis reports back **(curl)**
8. Move results from VM Disk to NFS server **(scp)**

Next plans with the cloud

- Now, we use only OpenStack platform
  - Later, add OpenNebula
- Now, we test on 10 VM's (max 64 CPU's) and 10 disk (10 * 100GB)
  - Later, scale up testing
- Now, we run only one workflow
  - Later, create more specific VM's for specific workflows
- Now, error handling is absent
  - Later, cover most possible errors in the cloud
- Now, we run on only one OpenStack server
  - Later, add more sites and exchange VM images between them, if needed

Pipelines

Supported analyses

- NGS variant calling pipeline
- RNA-seq analysis
- Liftover genome builds
- … and many more pipelines

- Out-of-the-box imputation pipeline
  - Automatic installation of tools and reference datasets
  - Liftover, phasing and imputation in 3 commands

LIMS

umcg

molgenis

LIMS

molgenis | Catalog | Data Explorer | Upload | Entities | BiobankConnect | Workflow Data Entry | Background | Entity Explorer | Admin | Account | Sign o

Choose a dataset: GAF list 2014-05-12 04:00

rch data values

a item filters

a item selection

- contact
- validationLog
- labStatusPhase
- labStatusComments
- lastUpDate
- *sequencingStartDate*
- *run*
- *flowcell*
- *lane*
- *barcodeMenu*
- *seqType*
- *prepKit*
- *capturingKit*
- *arrayFile*
- *arrayID*
- *GAF_QC_Name*
- *GAF_QC_Date*

zard

Data | Aggregates | Charts

internalSampleID	project	sequencer	sequencingStartDate	run	flowcell	lane	barcodeMenu	seqType	prepKit	capturingKit
1674	NIPTValidatie	SN163	130423	0493	BC1BFCACXX	8	RPI 10 TAGCTT	SR	NEB	None
1675	NIPTValidatie	SN163	130423	0493	BC1BFCACXX	8	RPI 11 GGCTAC	SR	NEB	None
1676	NIPTValidatie	SN163	130423	0493	BC1BFCACXX	8	RPI 12 CTTGTA	SR	NEB	None
1677	LDM_Klaas	SN163	130905	0510	BH7D9JADXX	1,2	MON 05 AAGACG	PE	Mondrian	SureSelect_All_Exon_50MB
1678	LDM_Klaas	SN163	130905	0510	BH7D9JADXX	1,2	MON 06 CCTCGG	PE	Mondrian	SureSelect_All_Exon_50MB
1679	LDM_Klaas	SN163	130905	0510	BH7D9JADXX	1,2	MON 07 GGATGT	PE	Mondrian	SureSelect_All_Exon_50MB
1680	LDM_Klaas	SN163	130905	0510	BH7D9JADXX	1,2	MON 08 TTCGCT	PE	Mondrian	SureSelect_All_Exon_50MB
1681	HaloPlex	SN163	130521	0494	AC20FTACXX	8	HP8 56 GTACGCAA	PE	Haloplex	None
1682	HaloPlex	SN163	130521	0494	AC20FTACXX	8	HP8 60 TAGGATGA	PE	Haloplex	None
1683	HaloPlex	SN163	130521	0494	AC20FTACXX	8	HP8 62 TCCGTCTA	PE	Haloplex	None
1684	HaloPlex	SN163	130521	0494	AC20FTACXX	8	HP8 64 TGAAGAGA	PE	Haloplex	None
1685	SCA_C	SN163	130710	0498	AD2AMUACXX	1	AGI 01 ATCACG	PE	Agilent	SureSelect_All_Exon_50MB
1686	SCA_C	SN163	130710	0498	AD2AMUACXX	1	AGI 08 ACTTGA	PE	Agilent	SureSelect_All_Exon_50MB
1687	SCA_C	SN163	130710	0498	AD2AMUACXX	1	AGI 10 TAGCTT	PE	Agilent	SureSelect_All_Exon_50MB
1688	SCA_C	SN163	130710	0498	AD2AMUACXX	1	AGI 11 GGCTAC	PE	Agilent	SureSelect_All_Exon_50MB
1689	SCA_C	SN163	130710	0498	AD2AMUACXX	2	AGI 02 CGATGT	PE	Agilent	SureSelect_All_Exon_50MB
1690	SCA_C	SN163	130710	0498	AD2AMUACXX	2	AGI 04 TGACCA	PE	Agilent	SureSelect_All_Exon_50MB
1691	SCA_C	SN163	130710	0498	AD2AMUACXX	2	AGI 07 CAGATC	PE	Agilent	SureSelect_All_Exon_50MB
1692	SCA_C	SN163	130710	0498	AD2AMUACXX	2	AGI 16 AAAGCA	PE	Agilent	SureSelect_All_Exon_50MB
1693	SCA_C	SN163	130809	0501	BD2BWCACXX	1	AGI 01 ATCACG	PE	Agilent	SureSelect_All_Exon_50MB

1083 data items found

« 1 ... 31 32 33 34 35 ... 55 »

Download as

This database was generated using the open source MOLGENIS database generator version 4.0.0-testing.
Please cite Swertz et al (2010) or Swertz & Jansen (2007) on use.

umcg

molgenis

Future work

Future work

- Finish implementation new NGS pipeline
- Collect and show more statistics to user in dashboard/commandline
- Add more workflows
- Unique *job/workflow* interface to run in different computational back-ends
  - Cluster (PBS, SGE, etc.)
  - Grid (gLite)
  - Cloud (OpenStack) <- Ongoing
- Interface with Galaxy (and other WMS?)
- Visual analytics of data/workflows/runs

Contributors: GCC team

- Compute development
  - George Byelas
  - Martijn Dijkstra
- Pipeline development and NGS analysis:
  - Pieter Neerincx
  - Gerben van der Vries
  - Lennart Johansson
  - Roan Kanninga

Github repositories

- Molgenis Compute (& tutorial)
  - https://github.com/molgenis/molgenis-pipelines/
- Imputation out-of-a-box
  - https://github.com/molgenis/molgenis-imputation
- Pipelines
  - https://github.com/molgenis/molgenis-pipelines/tree/master/compute5
- NGS (Under development)
  - https://github.com/molgenis/molgenis-pipelines/tree/master/compute5/NGS_alignment_SNP_calling