Content

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

```bash
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

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

- Extended parameter list 2

```plaintext
Name, defaultValue, hasOne_name
guest,            ,
event, wedding    ,
age_group,        , organizer
organizer,        ,
```
Extended target list

guest, age_group, organizer
Charly, child, Oscar
Cindy, child, Oscar
Abel, adult, Otto
Adam, adult, Otto
Adri, adult, Otto
Protocol template for organizer letter

```bash
#FOREACH age_group

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

```bash
#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*
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```
getFile and putFile are back-end specific, now we check if the files are present (cluster or localhost) and do srm/lfn file transfer (grid).

Input:

```
#!/bin/bash

getRemoteLocation()
{
    ARGS=${@}
    myFile=${ARGS[0]}
    remoteFile=srm://srm.grid.sara.nl/pnfs/grid.sara.nl/data/bbmri.nl/uyelas${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
}

getFile()
{
    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
}
```

Generate output:

```
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
### Alignment per lane

<table>
<thead>
<tr>
<th>Step</th>
<th>Cores</th>
<th>Ram (Gb)</th>
<th>disk</th>
<th>Average runtime (hrs)</th>
</tr>
</thead>
<tbody>
<tr>
<td>al00.fastqc</td>
<td>1</td>
<td>0.5</td>
<td>20 – 50Gb</td>
<td>6.336238616</td>
</tr>
<tr>
<td>al01.bwa_align_pair1</td>
<td>4</td>
<td>6</td>
<td></td>
<td>6.445865209</td>
</tr>
<tr>
<td>al02.bwa_align_pair2</td>
<td>4</td>
<td>6</td>
<td></td>
<td>6.445865209</td>
</tr>
<tr>
<td>al03.bwa_sampe</td>
<td>1</td>
<td>4</td>
<td></td>
<td>5.701997571</td>
</tr>
<tr>
<td>al04.sam_to.bam</td>
<td>1</td>
<td>4</td>
<td></td>
<td>1.126662113</td>
</tr>
<tr>
<td>al05.sam_sort</td>
<td>1</td>
<td>3</td>
<td></td>
<td>5.717598664</td>
</tr>
<tr>
<td>al06.picardQC</td>
<td>1</td>
<td>4</td>
<td></td>
<td>2.424761688</td>
</tr>
<tr>
<td>al07.mark_duplicates</td>
<td>1</td>
<td>4</td>
<td></td>
<td>2.364496053</td>
</tr>
<tr>
<td>al08.realign</td>
<td>1</td>
<td>10</td>
<td></td>
<td>4.11407711</td>
</tr>
<tr>
<td>al09.fixmates</td>
<td>1</td>
<td>6</td>
<td></td>
<td>2.91533394</td>
</tr>
<tr>
<td>al10.covariates_before</td>
<td>1</td>
<td>4</td>
<td></td>
<td>9.410815118</td>
</tr>
<tr>
<td>al11.recalibrate</td>
<td>1</td>
<td>4</td>
<td></td>
<td>6.387997875</td>
</tr>
<tr>
<td>al12.sam_sort</td>
<td>1</td>
<td>3</td>
<td></td>
<td>4.667501543</td>
</tr>
<tr>
<td>al13.covariates_after</td>
<td>1</td>
<td>4</td>
<td></td>
<td>10.23987346</td>
</tr>
<tr>
<td>al14.analyze_covariates</td>
<td>1</td>
<td>4</td>
<td></td>
<td>0.01349563</td>
</tr>
<tr>
<td>totals</td>
<td></td>
<td></td>
<td></td>
<td>74.3125798</td>
</tr>
</tbody>
</table>

### SNP calling per sample

<table>
<thead>
<tr>
<th>Step</th>
<th>Cores</th>
<th>Ram (Gb)</th>
<th>contamination checker</th>
<th>SNPs called</th>
</tr>
</thead>
<tbody>
<tr>
<td>vc00.merge</td>
<td>1</td>
<td>6</td>
<td>4</td>
<td>22</td>
</tr>
<tr>
<td>vc01.unified_genotyper</td>
<td>4</td>
<td>8</td>
<td>1</td>
<td>6.5</td>
</tr>
<tr>
<td>vc02.picardQC</td>
<td>1</td>
<td>4</td>
<td>4</td>
<td>9</td>
</tr>
<tr>
<td>vc03.coverage</td>
<td>4</td>
<td>10</td>
<td>1</td>
<td>6</td>
</tr>
<tr>
<td>contamination checker</td>
<td>1</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>totals</td>
<td></td>
<td></td>
<td></td>
<td>466Gb</td>
</tr>
</tbody>
</table>

### De novo assembly per sample

<table>
<thead>
<tr>
<th>Step</th>
<th>Cores</th>
<th>Mem (Mb)</th>
<th>Coverage</th>
</tr>
</thead>
<tbody>
<tr>
<td>abyss</td>
<td>6</td>
<td>140</td>
<td>100</td>
</tr>
</tbody>
</table>
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:

<table>
<thead>
<tr>
<th># Samples</th>
<th>Chip</th>
<th>Cohort</th>
</tr>
</thead>
<tbody>
<tr>
<td>24000</td>
<td>CytoSNPv2</td>
<td>LifeLines</td>
</tr>
<tr>
<td>5474</td>
<td>illumina 550k</td>
<td>Rotterdam study</td>
</tr>
<tr>
<td>416</td>
<td>affy500, illumina 550k, illumina OMNI 5M</td>
<td>Rotterdam study</td>
</tr>
<tr>
<td>3000</td>
<td>affy 250k, illumina 318k, 350k and/or 610k</td>
<td>Rotterdam Rucphen</td>
</tr>
<tr>
<td>10000</td>
<td>1 out of 5 different chips</td>
<td>Amsterdam NTR</td>
</tr>
</tbody>
</table>
Current developments

Compute Testing

Workflow: 1 of 1

id: 1
name: workflowImpute.csv

WorkflowElement:

<table>
<thead>
<tr>
<th>id</th>
<th>name</th>
<th>Workflow</th>
<th>protocol</th>
<th>PreviousSteps</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>impute2_s00</td>
<td>workflowImpute.csv</td>
<td>preparePedMapForImpute2.ftl</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>impute2_s01</td>
<td>workflowImpute.csv</td>
<td>convertPedMapToTriType.r.ftl</td>
<td>impute2_s00</td>
</tr>
<tr>
<td>3</td>
<td>impute2_s02</td>
<td>workflowImpute.csv</td>
<td>prepareStudy.ftl</td>
<td>impute2_s01</td>
</tr>
<tr>
<td>4</td>
<td>impute2_s03</td>
<td>workflowImpute.csv</td>
<td>convertPreparedStudyPedMapToGen.ftl</td>
<td>impute2_s02</td>
</tr>
<tr>
<td>5</td>
<td>impute2_s04</td>
<td>workflowImpute.csv</td>
<td>imputeWithImpute2.ftl</td>
<td>impute2_s03</td>
</tr>
<tr>
<td>6</td>
<td>impute2_s05</td>
<td>workflowImpute.csv</td>
<td>concatImpute2ResultsPerChr.ftl</td>
<td>impute2_s04</td>
</tr>
<tr>
<td>7</td>
<td>impute2_s06</td>
<td>workflowImpute.csv</td>
<td>convertConcatImpute2ResultToPedMap.ftl</td>
<td>impute2_s05</td>
</tr>
<tr>
<td>8</td>
<td>impute2_s07</td>
<td>workflowImpute.csv</td>
<td>calculateBgl2ForImpute2Results.ftl</td>
<td>impute2_s06</td>
</tr>
<tr>
<td>9</td>
<td>impute2_s08</td>
<td>workflowImpute.csv</td>
<td>convertResultPedMapToB316.ftl</td>
<td>impute2_s06</td>
</tr>
</tbody>
</table>

* = 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)

<table>
<thead>
<tr>
<th>H</th>
<th>name</th>
<th>Compu-rc-1g</th>
<th>TaskID</th>
<th>Rest.tg</th>
<th>Workflows</th>
<th>User</th>
<th>requirements</th>
<th>Status</th>
<th>Features</th>
<th>Last updated</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>input2_u0, txt1_1349165453555250000</td>
<td>BEFORE #touch SPSS_EXTERNAL.parname.par source SWORKS#tools/scratch/iptoch.js befe...</td>
<td>TASK1input2_u0, txt1__1345854535525000 touch: cannot touch /opt/gitlab/var/emr/jpname.o</td>
<td>none</td>
<td></td>
<td>input2_u0</td>
<td>bash</td>
<td></td>
<td>done</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>input2_u1, txt1_1349165453579520000</td>
<td>BEFORE #touch SPSS_EXTERNAL.parname.par source SWORKS#tools/scratch/iptoch.js befe...</td>
<td>TASK1input2_u1, txt1__1345854535795200 touch: cannot touch /opt/gitlab/var/emr/jpname.o</td>
<td>none</td>
<td></td>
<td>input2_u1</td>
<td>bash</td>
<td></td>
<td>done</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>input2_u2, txt1_1349165433786120000</td>
<td>BEFORE #touch SPSS_EXTERNAL.parname.par source SWORKS#tools/scratch/iptoch.js befe...</td>
<td>TASK1input2_u2, txt1__1345854337861200 touch: cannot touch /opt/gitlab/var/emr/jpname.o</td>
<td>none</td>
<td></td>
<td>input2_u2</td>
<td>bash</td>
<td></td>
<td>done</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>input2_u3, txt1_1349165454036890000</td>
<td>BEFORE #touch SPSS_EXTERNAL.parname.par source SWORKS#tools/scratch/iptoch.js befe...</td>
<td>TASK1input2_u3, txt1__1345854403689000 touch: cannot touch /opt/gitlab/var/emr/jpname.o</td>
<td>none</td>
<td></td>
<td>input2_u3</td>
<td>bash</td>
<td></td>
<td>done</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>input2_u4, txt1_1349165454251967000</td>
<td>BEFORE #touch SPSS_EXTERNAL.parname.par source SWORKS#tools/scratch/iptoch.js befe...</td>
<td>TASK1input2_u4, txt1__1345854425196700 touch: cannot touch /opt/gitlab/var/emr/jpname.o</td>
<td>none</td>
<td></td>
<td>input2_u4</td>
<td>bash</td>
<td></td>
<td>done</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>input2_u5, txt1_1349165454297101000</td>
<td>BEFORE #touch SPSS_EXTERNAL.parname.par source SWORKS#tools/scratch/iptoch.js befe...</td>
<td>TASK1input2_u5, txt1__1345854429710100 touch: cannot touch /opt/gitlab/var/emr/jpname.o</td>
<td>none</td>
<td></td>
<td>input2_u5</td>
<td>bash</td>
<td></td>
<td>done</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>input2_u6, txt1_1349165454352470000</td>
<td>BEFORE #touch SPSS_EXTERNAL.parname.par source SWORKS#tools/scratch/iptoch.js befe...</td>
<td>TASK1input2_u6, txt1__1345854435247000 touch: cannot touch /opt/gitlab/var/emr/jpname.o</td>
<td>none</td>
<td></td>
<td>input2_u6</td>
<td>bash</td>
<td></td>
<td>done</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>input2_u7, txt1_1349165454390598000</td>
<td>BEFORE #touch SPSS_EXTERNAL.parname.par source SWORKS#tools/scratch/iptoch.js befe...</td>
<td>TASK1input2_u7, txt1__1345854439059800 touch: cannot touch /opt/gitlab/var/emr/jpname.o</td>
<td>none</td>
<td></td>
<td>input2_u7</td>
<td>bash</td>
<td></td>
<td>done</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>input2_u8, txt1_1349165454414550000</td>
<td>BEFORE #touch SPSS_EXTERNAL.parname.par source SWORKS#tools/scratch/iptoch.js befe...</td>
<td>TASK1input2_u8, txt1__1345854441455000 touch: cannot touch /opt/gitlab/var/emr/jpname.o</td>
<td>none</td>
<td></td>
<td>input2_u8</td>
<td>bash</td>
<td></td>
<td>done</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>input2_u9, txt1_1349165454434214000</td>
<td>BEFORE #touch SPSS_EXTERNAL.parname.par source SWORKS#tools/scratch/iptoch.js befe...</td>
<td>TASK1input2_u9, txt1__1345854443421400 touch: cannot touch /opt/gitlab/var/emr/jpname.o</td>
<td>none</td>
<td></td>
<td>input2_u9</td>
<td>bash</td>
<td></td>
<td>done</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>input2_u10, txt1_1349165454455377000</td>
<td>BEFORE #touch SPSS_EXTERNAL.parname.par source SWORKS#tools/scratch/iptoch.js befe...</td>
<td>TASK1input2_u10, txt1__1345854445537700 touch: cannot touch /opt/gitlab/var/emr/jpname.o</td>
<td>none</td>
<td></td>
<td>input2_u10</td>
<td>bash</td>
<td></td>
<td>done</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>input2_u11, txt1_1349165454512522000</td>
<td>BEFORE #touch SPSS_EXTERNAL.parname.par source SWORKS#tools/scratch/iptoch.js befe...</td>
<td>TASK1input2_u11, txt1__1345854451252200 touch: cannot touch /opt/gitlab/var/emr/jpname.o</td>
<td>none</td>
<td></td>
<td>input2_u11</td>
<td>bash</td>
<td></td>
<td>done</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>input2_u12, txt1_1349165454601138000</td>
<td>BEFORE #touch SPSS_EXTERNAL.parname.par source SWORKS#tools/scratch/iptoch.js befe...</td>
<td>TASK1input2_u12, txt1__1345854460113800 touch: cannot touch /opt/gitlab/var/emr/jpname.o</td>
<td>none</td>
<td></td>
<td>input2_u12</td>
<td>bash</td>
<td></td>
<td>done</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>input2_u13, txt1_1349165454643285000</td>
<td>BEFORE #touch SPSS_EXTERNAL.parname.par source SWORKS#tools/scratch/iptoch.js befe...</td>
<td>TASK1input2_u13, txt1__1345854464328500 touch: cannot touch /opt/gitlab/var/emr/jpname.o</td>
<td>none</td>
<td></td>
<td>input2_u13</td>
<td>bash</td>
<td></td>
<td>done</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>input2_u14, txt1_1349165454722820800</td>
<td>BEFORE #touch SPSS_EXTERNAL.parname.par source SWORKS#tools/scratch/iptoch.js befe...</td>
<td>TASK1input2_u14, txt1__1345854472282080 touch: cannot touch /opt/gitlab/var/emr/jpname.o</td>
<td>none</td>
<td></td>
<td>input2_u14</td>
<td>bash</td>
<td></td>
<td>done</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>input2_u15, txt1_1349165454782379000</td>
<td>BEFORE #touch SPSS_EXTERNAL.parname.par source SWORKS#tools/scratch/iptoch.js befe...</td>
<td>TASK1input2_u15, txt1__1345854478237900 touch: cannot touch /opt/gitlab/var/emr/jpname.o</td>
<td>none</td>
<td></td>
<td>input2_u15</td>
<td>bash</td>
<td></td>
<td>done</td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>input2_u16, txt1_1349165454882313000</td>
<td>BEFORE #touch SPSS_EXTERNAL.parname.par source SWORKS#tools/scratch/iptoch.js befe...</td>
<td>TASK1input2_u16, txt1__1345854488231300 touch: cannot touch /opt/gitlab/var/emr/jpname.o</td>
<td>none</td>
<td></td>
<td>input2_u16</td>
<td>bash</td>
<td></td>
<td>done</td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>input2_u17, txt1_1349165454927916000</td>
<td>BEFORE #touch SPSS_EXTERNAL.parname.par source SWORKS#tools/scratch/iptoch.js befe...</td>
<td>TASK1input2_u17, txt1__1345854492791600 touch: cannot touch /opt/gitlab/var/emr/jpname.o</td>
<td>none</td>
<td></td>
<td>input2_u17</td>
<td>bash</td>
<td></td>
<td>done</td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>input2_u18, txt1_1349165454957661000</td>
<td>BEFORE #touch SPSS_EXTERNAL.parname.par source SWORKS#tools/scratch/iptoch.js befe...</td>
<td>TASK1input2_u18, txt1__1345854495766100 touch: cannot touch /opt/gitlab/var/emr/jpname.o</td>
<td>none</td>
<td></td>
<td>input2_u18</td>
<td>bash</td>
<td></td>
<td>done</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>input2_u19, txt1_1349165455122480000</td>
<td>BEFORE #touch SPSS_EXTERNAL.parname.par source SWORKS#tools/scratch/iptoch.js befe...</td>
<td>TASK1input2_u19, txt1__1345854512248000 touch: cannot touch /opt/gitlab/var/emr/jpname.o</td>
<td>none</td>
<td></td>
<td>input2_u19</td>
<td>bash</td>
<td></td>
<td>done</td>
<td></td>
</tr>
</tbody>
</table>
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
<table>
<thead>
<tr>
<th>GCC/compute</th>
<th>eBioGrid</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ger Strikwerda</td>
<td>David van Enckevort</td>
</tr>
<tr>
<td>Marcel Burger</td>
<td>Irene Nooren</td>
</tr>
<tr>
<td>Martijn Dijkstra</td>
<td>Jan Bot</td>
</tr>
<tr>
<td>Morris Swertz</td>
<td>Mathijs Kattenberg</td>
</tr>
<tr>
<td>Wil Bruins-Koetsier</td>
<td>Pieter Neerincx</td>
</tr>
<tr>
<td></td>
<td>Tom Visser</td>
</tr>
</tbody>
</table>

And you, our local, national and international collaborators