HPC at Utrecht:
From my meeting with Ies Nijman,

1. The HPC can be utilized as a scalable model payment
2. It is a consorted server of CPUs + GPUs
3. The basic package:
    a. Interactive slot and normal submission node to the terminal
    b. Access to the Wiki Page and How to use manual with detailed instructions
    c. Supports most stat softwares like R, SAS, GenStat, Matlab, Jupyter, Py notebooks, MySQL, etc.
    d. Apps can be downloaded from interactive environment common in all users, details in pamphlet
    e. 1 share: EUR 1200
    f. 50000 CPU-hr
    g. 1tb HPC storage free, 1 tb cold storage
    h. Shared and used by anyone from the group
    i. Flexible control and lax rules
4. Extra storage
    a. 1tb HPC: EUR 280/yr
    b. 1tb cold: EUR 45/yr
5. Graphics server:
    a. GPU: 25 hrs/ share, EUR 2350
6. Trial account:
    a. Free access for one year
    b. Shared storage space; no security
    c. Redundant storage file system: once deleted, permanent
    d. Use by terminal access to submission and storage transfer nodes
    e. Access to Wiki page and How to manual
    f. Support of HPC-community in UMC, UU and other allied organizations for initial learning phase

High Performance Computing

The High Performance Computing (HPC) facility is available for all Life Science researchers at **Utrecht Science Park**. Without access to proper compute and storage infrastructures, bioinformatics analyses are impaired and life science research is seriously hampered. Currently, twenty-nine research groups are actively using the HPC cluster.
We are actively investigating new possibilities and coordinating activities aimed at providing better storage and network infrastructure. Many of these activities are done in close collaboration with the IT departments from the UBC partners as well as with research groups involved.

Technical information

The High Performance Computing facility consists of 1544 cores and 600TB of High-Performance storage. The HPC facility runs on CentOS Linux and provides a batch-wise queueing system with a few head nodes and many compute nodes for submitting and running many computational tasks in parallel.

What others say about using the HPC

More information: **HPC facility wiki**
Contact: **Ies Nijman**

Welcome to the High-Performance Computing (HPC) Facility wiki

Utrecht Bioinformatics Center

ubc.uu.nl

High-Performance Computing facility

[illegible]

Universiteit Utrecht

UMC Utrecht

Hubrecht Institute

The High-performance Computing (HPC) facility is setup to provide high-performance computing power to all life science researchers at Utrecht Science Park. Coordinated by the Utrecht/Bioinformatics Center and subsidized by Utrecht University and University Medical Center, it currently provides computational power to over twenty different research groups located within Utrecht University, UMC Utrecht, Hubrecht Institute and Princess Máxima Center for Pediatric Oncology.

(for a high-resolution version of the HPC flyer, click on the thumbnail on the left)

General information

The HPC facility consists of 1544 cores and 600TB of High-Performance storage. The HPC facility runs on CentOS Linux and provides a batch-wise queueing system with a few head nodes and many compute nodes for submitting and running many computational tasks in parallel.

Dedicated administrators maintain and develop the HPC infrastructure and provide support to end users. These positions are funded by UMC Utrecht and Utrecht University (ITS).

Participating groups. Currently, twenty-nine research groups are actively using the HPC cluster.

HPC user council. To steer future directions of the HPC infrastructure together with the participating research groups, a HPC user council has been setup. We aim to meet four times a year to discuss the usage of the HPC cluster as well as new developments.

How to get involved. Research groups can participate by funding the hardware required for their own computational needs. HPC storage capacity can be rented on a per Terabyte, per year basis. For testing purposes and/or trying out the HPC infrastructure, free trial accounts can also be arranged that have (limited) access to the HPC resources, contact us if you are interested.

Conditions and Support

The conditions that apply when using the HPC infrastructure and the level of support that we are able to provide can be found at ConditionsAndSupport.

- Conditions
- Support
- Your Privacy

Contact details

The HPC team is responsible for setting up and maintaining the HPC infrastructure, as well as for helping out with HPC related user questions. For details and contact information, go here.

First-time users

To get you started, some initial information is provided here (password required):

No permission to view HPC

How to's

A useful collection of How to's is provided here (password required):

No permission to view HPC

Software installation

The HPC infrastructure provides the basis to install any software that is needed by users. More details are provided here (password required).

- Can I install my own software?
- Software of general interest
- How to install your own software
  - C software
  - Python modules
  - Your own R version
  - R packages
  - Perl libraries
  - Making software available using GUIX
  - Making software available using LMOD
  - Making software available using SINGULARITY

Can I install my own software?

The answer is: yes, (of course) you can. There is a group-specific directory that is made available for this purpose:

```
/hpc/local/osversion/group/
```

Note that `osversion` is the current Linux version and `group` is your own group name. Here, you can install any software you like and maintain it yourself. See here for explanations how to install C, Perl or R packages.

Software of general interest

If you find a software to be of general interest to HPC users, let us know. We can provide is as an software module or install it as a system package (rpm) and update it on a regular basis.

However, if you are dependent on a specific version of a software package and don't want regular updates, we advise you to install it yourself and encourage you to take the benefits of #Making software available using LMOD.

How to install your own software

A specific directory is made available for group-specific software to be installed:

```
/hpc/local/osversion/group/
```

C software

You can download your C software of interest and unpack it in this directory. Typically, a pre-installation configuration is done by executing:

```
./configure --prefix /hpc/local/osversion/group/package
```

This will create a "Makefile" which explains to the "make" utility how the software should be compiled, and where the software will be installed. The 'prefix' is the top directory under which the whole package will be installed. You may want to include the package version number in the name of this directory; that way you can install more than one version next to each other. Use symlinks to point to the 'default' installation.

After the `configure` step, you compile the software using:

```
make
```

The Makefile will be read, building the application binaries. To install these binaries, use:

```
make install
```

That is it! You can check the user documentation of the installed software for details of how to run the application.

Python modules

use pip to install python modules.

```
module load python
module list
pip list
pip install --user <module>
```

Create your virtual software environment for a specific Python project with `Python Virtual Environment`:
`see :` http://docs.python-guide.org/en/latest/dev/virtualenvs

Your own R version

Go to http://cran-mirror.cs.uu.nl/, download the latest "R-3.0.whatever.tar.gz" file to a submit host (hpcs01/hpcs02). Copy this file to a directory that has enough space (about 300M). In this case, let's assume /tmp, but your homedir probably has enough space as well.

```
cd /tmp

wget http://cran-mirror.cs.uu.nl/src/base/R-3/R-3.0.2.tar.gz

tar -zxvf R-3.0.2.tar.gz

cd R-3.0.2
```

Now, we'll have to "configure" and "make" this:

```
./configure --prefix=/hpc/local/CentOS7/bofh/R-3.0.2

make

make install
```

Where you replace "/hpc/local/CentOS7/bofh/R-3.0.2" with some path where you have write-access. Something like "/hpc/local/CentOS7/YOURGROUP/R-3.0.2" would probably be a good choice.

Now, you would start this version of R by entering the full path:

```
/hpc/local/CentOS7/bofh/R-3.0.2/bin/R
```

Or by adding the directory /hpc/local/CentOS7/YOURGROUP/R-3.0.2/bin to your own PATH (e.g. in your $HOME/.bash_profile).

R packages

Packages can easily be installed inside R by providing a local path:

```
R

install.packages( "yourLibrary", lib = "/hpc/local/osversion/group/path" )

library( "yourLibrary", lib.loc = "/hpc/local/osversion/group/path" )
```

Alternatively, you can customize your Linux environment variables and set `R_LIBS` to `/hpc/local/osversion/group/path`. This way, you can leave out the path specification in `R`.

Check

```
module load R
R
find.package('<mypackage>')
```

Perl libraries

To install CPAN perl libraries, you first have to instruct CPAN which directory to use. You can do this by modifying your CPAN configuration, from within the CPAN shell:

```
cpan

o conf mbuildpl_arg "installdirs=site
install_base=/hpc/local/osversion/group"

o conf makepl_arg "INSTALLDIRS=site
INSTALL_BASE=/hpc/local/osversion/group"

o conf prefer_installer MB

o conf prerequisites_policy follow

o conf commit
```

After this, CPAN should install all perl libraries in the appropriate directory such that they are available on the entire cluster. Your `PERLLIB` environment variable should be set to include `/hpc/local/OSVERSION/GROUP/lib/perl5`.

In the past, the environment variable `PERL_INSTALL_ROOT` used to be used for cpan installs, but that doesn't work anymore and wreaks havoc on the install if you use the cpan configuration shown above. I.o.w., be sure not to define `PERL_INSTALL_ROOT`.

If you later just wish to install packages, you can use the `cpan -i` command from the command line. No need to startup the CPAN shell.

Making software available using `GUIX`

`Guix` is a package manager and is installed for general use.

- GNU Guix for scientists .
- GNU Guix for bioinformaticians .
- Common problems and solutions .

You can find information in the

- Wiki .

and

- Guix website in the UMC-DBG domain .

Add these lines in your ~/.bash_profile file to enable guix .

```
# guix environment

# see : https://hpcguix.op.umcutrecht.nl/getting-started


export PATH=$PATH:"/gnu/profiles/base/bin"


GX_PROFILE="$HOME/.guix-profile"

if [ -f "$GX_PROFILE/etc/profile" ]; then

  . "$GX_PROFILE/etc/profile"

fi
```

## Making software available using LMOD

#Making software available using LMOD.

## Making software available using SINGULARITY

Singularity (docker compatible but more secure) is available on the compute-nodes (n00XX).
You can download docker- or singularity- images and run your programs in singularity.
p.e. R and SAIGE module :

You can execute this in a qlogin session.

SAIGE on singularity(image)
# Download a singularty image for SAIGE:
```
singularity pull shub://singularity-hub.org/statgen/singularity-saige
```

# Login into the container
```
singularity shell ./statgen-singularity-saige-master-latest.simg
```

# Check if SAIGE is available

```
R
```

```
installed.packages()

q()

n
```

# run a R script in the container
# create 1.R

```
library("SAIGE")

print("Hello World!")
```

# You can run this in a qsub script
# Run your code by :
```
singularity exec ./saige-0.35.8.1.simg Rscript 1.R
```

SAIGE on singularity with docker image

```
singularity pull docker://wzhou88/saige:0.35.8.1

singularity shell ./saige:0.35.8.1

R

installed.packages()

q()

n
```

# run
```
singularity exec ./saige-0.35.8.1.simg Rscript 1.R
```

Utrecht Bioinformatics Center

ubc.uu.nl

High-Performance Computing facility

Participate why?
Your research project involves computationally and/or memory intensive tasks which are too big to handle on your local workstation.

Benefits of using the HPC infrastructure include: no need for system administration, speedup of computational tasks, use of shared CPU capacity and an active HPC user community.

Who can participate?
All (Life Science oriented) research groups at the University Medical Center Utrecht, Utrecht University and Hubrecht Institute

How to participate?
For testing purposes and/or trying out the HPC infrastructure, trial accounts can be arranged with (limited) access to HPC resources.
Participating research groups pay a fee based on the compute and storage resources they require.

HPC infrastructure
- 1008 cores, 10TB working memory (68 compute nodes; 12-48 cores; 128GB-1TB each)
- Linux-based Operating System (CentOS7)
- Fast and concurrent data access (10 Gb/s network, 60 Gb/s, 500TB storage system)
- Fair share usage
- Controlled (remote) access
- User-specific home directory
- Group-specific data and installation directory

Funding
The HPC facility is made possible by funding from University Medical Center Utrecht, Utrecht University, research IT program (UMC Utrecht) and ITS (UU).

Are you interested in the HPC facility and think about participating?

Do not hesitate to contact Patrick Kemmeren
p.kemmeren@umcutrecht.nl
phone: +31-(0)88-7568959
office: Stratenum 2.213

We can setup a trial account for you to explore the HPC infrastructure.
For further information also see hpcwiki.op.umcutrecht.nl.

•