

# Brugervejledning til Danmarks Statistiks Linux-miljø hos NGC

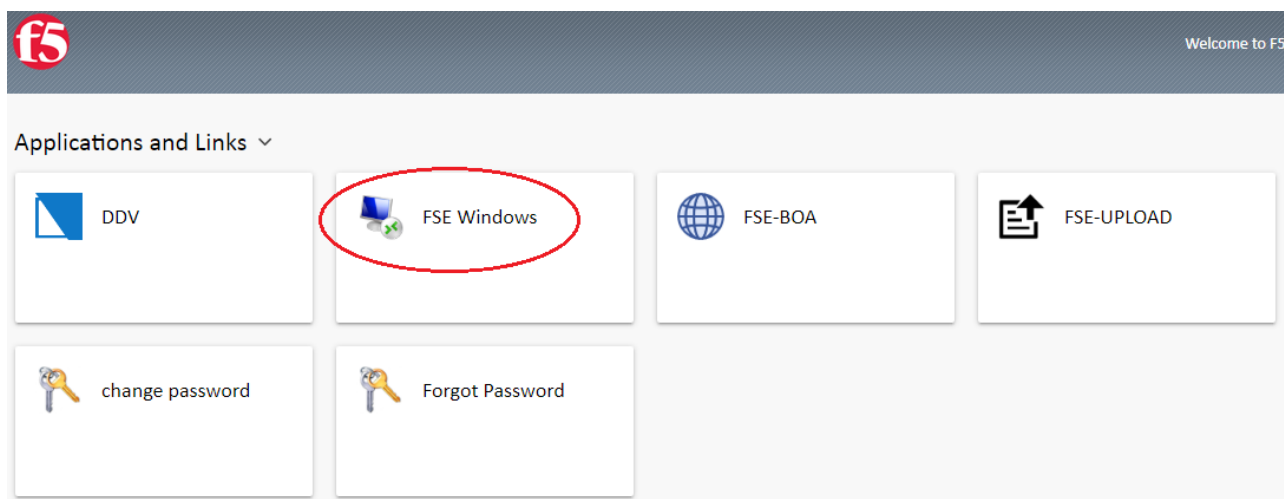
Denne brugervejledning forklarer

- Hvordan man opretter forbindelse til NGC-server og dit projekt
- Hvordan man starter terminalen
- Mappestrukturen
- Hvilke programmer, der tilbydes
- Hvordan man indlæser moduler og starter programmer

## Tilslutning til NGC Server og start af terminalen

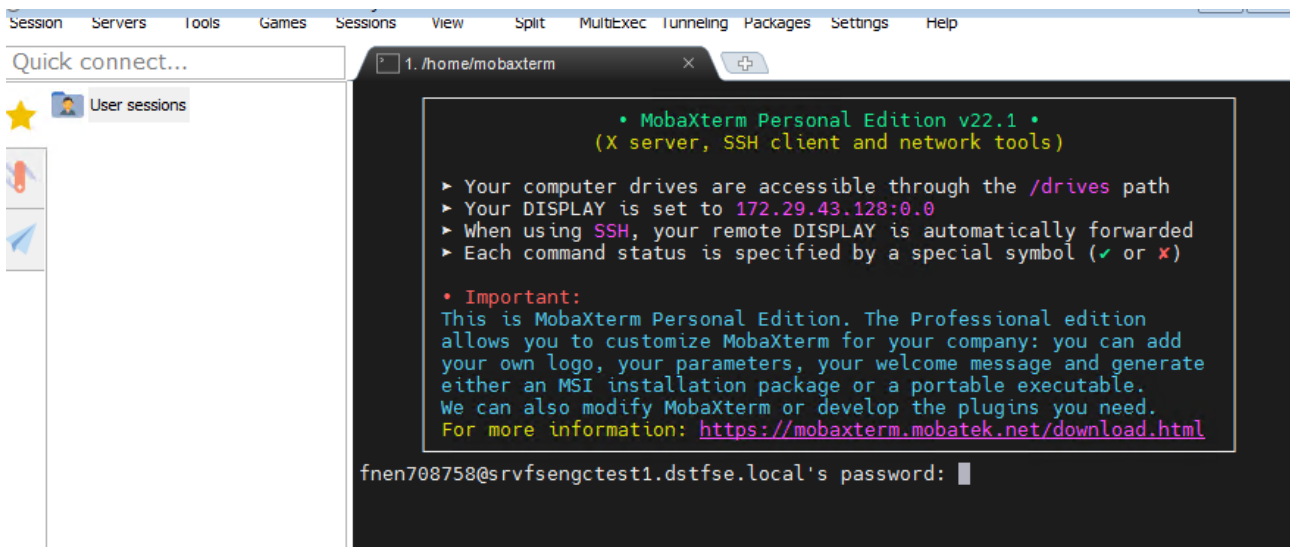
Du opretter forbindelse til din NGC-server fra FSE-windows frontend. Fra remote.dst.dk skal du vælge FSE-Windows som vist på skærmbilledet nedenfor.

Find vejledninger til at logge på Danmarks Statistiks Forskningsservere



Når du er logget på frontenden, skal du dobbeltklikke på linket til MobaXterminal. Dokumentationen for MobaXterminal er tilgængelig på <https://mobaxterm.mobatek.net/documentation.html>.

Du bliver bedt om at gå til en Linux-terminal, hvor du bliver bedt om at angive adgangskoden til dit projekt. Her skal du bruge Linux-kommandoer til at interagere med systemet.



## Mappestruktur

Du starter din session i din hjemmemappe (~), som er /brugere/[ident]+projektnummer. [ident]+projektnummer. Det vil i dette dokument henvises til som projekt-id. Mapestrukturen er den samme som andre projekter under forskningsordningen. Hvis du er bekendt med projekterne og de forskellige typer maskiner, der findes hos Danmarks Statistik, vil du kunne genkende strukturen.

Skift til databiblioteket ved at skrive `cd data`. I denne mappe finder du to mapper: rådata og arbejdsdata. Rådata indeholder alle data leveret af Danmarks Statistik (Grunddata og eksterne data). Arbejdsdata indeholder alle de data, der produceres under analyse (data til analyse, scripts, resultater osv.). Hver mappe indeholder mapper med projektnumre, og dine data vil ligge i mappen med det tilsvarende projektnummer.

```
[fnen708758@srvfsengctest1 ~]$ pwd
/users/fnen708758
[fnen708758@srvfsengctest1 ~]$ ls -l
total 2464
drwx----- 2 fnen708758 fnen708758      0 Jun 20 04:30 data
drwx----- 2 fnen708758 fnen708758 47104 Jul 12 09:51 formater
drwx----- 2 fnen708758 fnen708758 2424832 Aug 21 05:00 logging
[fnen708758@srvfsengctest1 ~]$ cd data
[fnen708758@srvfsengctest1 data]$ ls -l
total 0
drwx----- 2 fnen708758 fnen708758 0 Jul  5 10:46 rawdata
drwx----- 2 fnen708758 fnen708758 0 Jul  5 10:46 workdata
[fnen708758@srvfsengctest1 data]$
```

Hvis du har brug for at se på dine rådata for dit projekt 70xxxx, skal du skrive

```
$ cd rawdata/70xxxx
```

Og hvis du har brug for at se dine filer relateret til dine analyser, skal du skrive

```
$ cd arbejdsdata/70xxx.
```

## Adgangskodeændring

Adgangskoder kan ændres ved at bruge funktionerne på [remote.dst.dk](http://remote.dst.dk). [Se vejledningen til skift af password](#) på vores hjemmeside.

Hvis du ved login får besked om, at din adgangskode er tæt på at udløbe, kan du enten ændre den fra Windows som normalt, eller du kan ændre den fra Linux. Resultatet er det samme.

Start adgangskodeændring ved at skrive

```
$ passwd
```

Svar på prompten om eksisterende og ny adgangskode. Den nye adgangskode skal overholde de nuværende kvalitetskrav til adgangskoden.

## Tilgængelige teksteditorer

Der er flere teksteditorer, som er tilgængelige i dette Linux-miljø.

- VS-kode
- Vim
- Emacs
- Nano

Fra terminalen kan du starte hvert program ved at kalde programmets navn. Emacs er mere end en teksteditor. Det giver også mulighed for at liste filer og mapper samt kørende terminaler.

## Overførsel af analyseresultater og start af FSE-hjemsend

Programmet til overførsel af resultater kan startes med følgende kommando:

```
$ /opt/dst/hjemsend
```

Det vil derefter starte en Google Chrome-browser, som vil bede om din adgangskode.

Dette starter applikationen, som er identisk med applikationen til overførsel af resultater på Danmarks Statistiks Windows-servere. [Se vejledninger til hjemsendelse og overholdelse af datasikkerhed](#)

## Tilgængelig software

Det er muligt at bruge software både leveret af Danmarks Statistik og NGC. Vi beskriver først det sæt af software, som Danmarks Statistik har leveret. Du kan arbejde på R-sproget med enten den officielle R-software eller RStudio. Du kan også arbejde i Python med Anaconda eller Spyder.

C/C++/Fortran-kompilere er tilgængelige som til enhver almindelig Linux-distribution og kan bruges til udvikling. Du kan finde dokumentation for de to compilere for, hvordan du bruger disse værktøjer, på internettet.

## Software leveret af Danmarks Statistik

En række almindeligt tilgængelige værktøjer er blevet installeret i miljøet.

- PowerShell Core
- Anaconda Python
- Visual Studio Code
- R

## PowerShell Core

```
$ pwsh
```

PowerShell-scripts starter med følgende shebang-linje

```
$ #!/usr/bin/pwsh
```

PowerShell-scripts gøres eksekverbare med

```
$ chmod ug+x ./script
```

Derefter kan scriptet køres direkte fra prompten

```
$ ./script
```

Du kan dog ikke ændre tilstanden for et kompileret program til eksekverbar (x), hvis den er placeret på data/arbejdsdata/[projektnummer]. Du skal kopiere filen til din hjemmemappe, dvs. brugere/[projekt-id].

## Anaconda Python

Anaconda3 4.12.0/3.21.8 med Python 3.9.12

Kør disse kommandoer. Når du har kørt dem en gang, er der ikke behov for at køre dem igen.

```
$ /opt/anaconda3/bin/activate
$ /opt/anaconda3/condabin/conda init
```

Så får du besked om at logge ud igen (f.eks. via Ctrl-D og derefter R for (genopret forbindelse)). Når du er logget ind igen, kan du bruge Anaconda som normalt og køre CONDA kommandoer, f.eks.

```
$ conda info
```

/eller du kan skabe nye Anaconda-miljøer, f.eks.

```
$ conda create --name test_env python=3.10 pandas pip jupyter -y
$ conda activate test_env
$ pip install pyyaml
```

Så kan du starte f.eks. jupyter notesbog

```
$ jupyter notesbog
```

Som starter Google Chrome med din notesbog. Spyder Python editor er installeret i basismiljøet (men kan også tilføjes til egne miljøer)

```
$ conda activate base
$ spyder
```

## Brug af Anaconda-miljøer

Anvendelsen af Python vil ret hurtigt føre til, at du bliver nødt til at oprette og redigere Anaconda-miljøer. Disse miljøer er separate miljøer, hvor du kan køre forskellige versioner af Python og ikke mindst af relevante programmoduler. Miljø er beskrevet mere detaljeret hos Anaconda.

Vi anbefaler, at du opretter Anaconda-miljøer til hver af dine Python-applikationer, og at du vedligeholder og dokumenterer dine miljøer via .yaml-konfigurationsfiler. Her er et simpelt eksempel på en environment-yaml-fil:

```
name: env_number_et
dependencies:
  - Python=3.9
  - pylint
  - Pandas
  - pip
  - autopep8
  - pip:
    - MonthDelta==0.9.1
```

Her hentes moduler fra både Anacondas standarddepoter og fra det andet store modul-univers pypi (pip).

Danmarks Statistik har spejle af en række Anaconda-kanaler (inkl. standardkanalerne) og pypi.

De kanaler vi har spejle af er nedenfor, hvor CONDA smedekanalene er den vigtigste.

default channels

- <http://srvnexus1.dst.dk:8081/repository/conda/main>
- <http://srvnexus1.dst.dk:8081/repository/conda/msys2>
- <http://srvnexus1.dst.dk:8081/repository/conda/r>

non-default channels

- <http://srvnexus1.dst.dk:8081/repository/conda-forge>
- <http://srvnexus1.dst.dk:8081/repository/bioconda>
- <http://srvnexus1.dst.dk:8081/repository/plotly>
- <http://srvnexus1.dst.dk:8081/repository/pytorch>
- <http://srvnexus1.dst.dk:8081/repository/sebp>
- <http://srvnexus1.dst.dk:8081/repository/fastai>
- <http://srvnexus1.dst.dk:8081/repository/pyg>

Hvis du har brug for fx CONDA-forge, skal du tilføje en kanalsektion til .yaml-filen, som i dette eksempel, hvor sqlalchemy-modulet (findes kun i CONDAforge-kanalen) skal downloades.

```
name: env_nummer_to
channels:
  - http://srvnexus1.dst.dk:8081/repository/conda/main
  - http://srvnexus1.dst.dk:8081/repository/conda-forge
dependencies:
  - python=3.10.4
  - pylint
```

- autopep8
- psycpg2
- sqlalchemy
- sqlalchemy-codegen

Du kan også henvise til spejlede kanaler fra kommandolinjen, f.eks.

```
$ conda create --name pyt python=3.10 -y
$ conda activate pyt
$ conda install pytorch -c http://srvnexus1.dst.dk:8081/repository/pytorch -y
```

Generelt anbefaler vi, at du holder dine omgivelser snævre. Det er bedre med mere specifikke miljøer end et stort miljø beregnet til alle applikationer. Og referer kun til de kanaler, du rent faktisk har brug for.

Desuden er det en god idé at låse modulerne til bestemte versioner (såsom python=3.10.4), når ens kode når et stadie, hvor den har produktionskarakteristika. Herefter forbliver modulerne på de angivne versioner, indtil du vælger fx at oprette en ny version af miljøet til en anden applikation. Kopier yml'en, fjern versionsbindingerne og nyd alle nye funktioner.

## Anaconda-kommandoer

Her er en oversigt over de mest almindeligt anvendte Anaconda-kommandoer. Den fulde oversigt er tilgængelig fra Anaconda.

Kommando	Beskrivelse
<code>conda env update -f filename.yml</code>	create or update environment described in filename.yml
<code>conda activate env_name</code>	enable environment with env_name
<code>conda deactivate</code>	disable currently env and go back to base environment
<code>conda env remove --name env_name</code>	delete environment with the name env_name

## R

R og RStudio er installeret og et stort antal almindeligt anvendte pakker er blevet installeret. Du kan se hvilke pakker der er installeret med

```
installed.packages()
```

Du kan selv supplere de fælles pakker ved at installere til dit eget private miljø, f.eks. via

```
install.packages("Pakkenavn")
```

De fælles pakker installeres af IT. Følgende pakker er forudinstalleret

### Liste over R-pakker leveret af Danmarks Statistik

Amelia	checkmate	countrycode	data.table	DBI	dbplyr	devtools
Epi	epitools	fuzzyjoin	glmnet	glue	haven	Hmisc
httr	janitor	jsonlite	kableExtra	keras	knitr	lcmm

lme4	lubridate	magrittr	MASS	mgcv	mice	missForest
naniar	officer	patchwork	pec	pkgdown	ranger	reticulate
rio	riskRegression	rlang	rmarkdown	roxygen2	rvest	sf
spatstat	stringdist	stringi	survival	testthat	tidymodels	tidyselect
tidyverse	timereg	todor	validate	xlsx	xml2	zoo

## RStudio

RStudio kan startes ved at skrive fra prompten

```
$ rstudio
```

Så starter den velkendte gui, og du kan arbejde med scripts placeret på fx arbejdsdata.

## Visual Studio code

Du starter koden som normalt i den mappe, du vil arbejde med

```
$ code .
```

Du kan få IT i Danmarks Statistik til at downloade udvidelser (.vsix-filer) til VS Code for dig. Du skal selv installere de downloadede udvidelser, f.eks. med

```
$ code --install-extension ~/ms-vscode.PowerShell-2021.12.0.vsix
```

**Vs Code** på Linux kan halte lidt i skærmbilledet. Det kan være en fordel at prøve disse indstillinger for at reducere forsinkelser

```
{  
  "workbench.enableExperiments": falsk,  
  "telemetry.telemetryLevel": "fra",  
  "extensions.autoUpdate": falsk,  
  "python.experiments.enabled": falsk  
}
```

## LibreOffice

[LibreOffice](#) er tilgængelig på serveren og kan hjælpe dig med at opsætte dine resultater. Bare skriv

```
$ libreoffice
```

for at starte applikationen.

## HTOP

HTOP er en interaktiv procesfremviser, der viser belastning på maskinen

```
$ htop
```

Hjælp og vejledning med

```
$ man htop
```

## Git klient

Med git-klienten kan du håndtere git-repositories. Bemærk, der er ingen adgang til eksterne git remote repositories.

## Emacs editor

Start editoren:

```
$ emacs
```

Eller

```
$ emacs &
```

Hvis du vil starte emacs i baggrunden.

## NVIDIA SMI

Vis belastning af installerede GPU'er. GPU'er er kun tilgængelige på GPU-noder, ikke på Fat-nodes eller Thin-nodes.

```
$ nvidia-smi
```

## Fejlfinding

Hvis du er inaktiv for længe, skal du muligvis genautenticere. For eksempel, hvis et netværksdrev ikke er tilgængeligt. Genautentificer ved at køre kinit og svar på adgangskodeprompten.

```
$ kinit
```

## Hurtigstart med miljømoduler leveret af NGC

### Miljømoduler i Danmarks Statistik NGC Linux-miljø

NGC stiller en række moduler til rådighed. Vi vil her nævne software, som er relevant for analyse og også for størstedelens vedkommende tilgængelig på Danmarks Statistiks servere. Du kan se en komplet liste over tilgængelige moduler senere i vejledningen.

Software, som NGC stiller til rådighed, skal indlæses fra NGC's softwarelager. Disse kaldes også moduler og skal indlæses ved hver session. Nogle af disse software har afhængigheder (for eksempel R og RStudio) og kræver, at du indlæser andre moduler på forhånd ved at bruge softwaren. Se afsnittet om, hvordan du indlæser moduler for yderligere detaljer.

Installationen og konfigurationen af miljømoduler er hovedsageligt en kopi af den identiske opsætning, som NGC leverer til deres egne kunder.

Spørgsmål vedrørende konfiguration af Miljømoduler og funktionen af specifikke moduler kan rettes til IT i Danmarks Statistik, men vil i de fleste tilfælde blive videregivet til NGC.



Moduler giver en mekanisme til at indstille/frakoble alle miljøvariabler relateret til en given pakke i én UNIX-skalkommando. Moduldefinitioner er indeholdt i såkaldte modulfiler, der er placeret i flere mapper. Biblioteksstrukturen er afhængig af systemet.

For at finde aktuelt tilgængelige moduler samt indlæsning/afslæsning af moduler i dit miljø, se venligst kommandoerne nedenfor.

## Modulkommandoer

For at få en liste over alle tilgængelige moduler

```
$ module avail
```

leder efter et specifikt modul?

```
$ module avail <module name>
```

hvordan indlæser man et modul?

```
$ module load <module name/version>
```

få grundlæggende oplysninger om modulfiler

```
$ module whatis <module name>
```

få information om flere modulfiler på én gang

```
$ module whatis <module name 1> <module name 2>
```

søge efter en bestemt streng i de grundlæggende oplysninger om modulfilerne. -i-flaget er til modulparameterevaluering, der ikke er følsom over for store og små bogstaver

```
$ module apropos -i <string / module name>
```

få yderligere information med moduldisplay

```
$ module display <module name>
```

"modul display" vil også vise konflikt (moduler, der er i konflikt med dette) og prereq (moduler, der skal indlæses for at bruge dette).

liste Anaconda-moduler (dvs. Python-moduler) - liste software ikke inkluderet som moduler

```
$ module display anaconda2/<version>
```

liste Perl-moduler - liste software ikke inkluderet som moduler

```
$ module display perl/<version>
```

liste R-pakker - liste software ikke inkluderet som moduler

```
$ module display R/<version>
```

se hvilke moduler du har indlæst i miljøet

```
$ module list
```

opret et gemt standardmiljø - indlæs alle dine ønskede moduler og skriv derefter

```
$ module save
```

for at gemme et sæt ikke-standardmoduler, skal du tildele et miljønavn

```
module save <environment-name>
```

indlæse alle moduler i et standard gemt miljø

```
$ module restore
```

indlæse moduler af et specifikt gemt miljø

```
$ module restore <environment-name>
```

få en liste over dine miljøer

```
$ module savelist
```

hvordan aflæser man et modul?

```
$ module unload <module name/version>
```

skift et specifikt modul ud med et andet (især nyttigt for at skifte mellem forskellige versioner af det samme program)

```
$ module switch <old_module> <new_module>
```

aflæs alle moduler

```
$ module purge
```

administrere moduler i dit login-miljø (OBS - hold det på et absolut minimum)

tilføj modulefiler til shellens initialiseringsfil i brugerens hjemmemappe

```
$ initadd <module name>
```

gør det samme som initadd, men sætter de givne moduler foran i begyndelsen af listen

```
$ initprepend modulefile <module name>
```

fjern modulefiler fra shellens initialiseringsfiler

```
$ initrm <module name>
```

skift modulefil1 med modulefil2 i shellens initialiseringsfiler

```
$ initswitch <module name 1> <module name 2>
```

liste alle de modulefiler, der er indlæst fra shellens initialiseringsfil

```
$ initlist
```

Ryd alle modulefilerne fra shellens initialiseringsfiler

```
$ initclear
```

Ved at bruge moduloutput med andre kommandoer skriver modulkommandoer deres output til STDERR (hvor 'normale' kommandoer skriver til STDOUT). Så hvis du for eksempel bruger grep efter en modulkommando, bliver resultatet måske ikke helt, som du forventede.

```
$ module load tools
$module whatis | grep sequence
```

du skal huske at omdirigere STDERR (2>&1 nedenfor):

```
$ module whatis 2>&1 | grep sequence
```

få flere modulumuligheder

```
$ man module
```

## Liste og finde moduler

For at få vist dine aktuelt tilgængelige moduler, skriv:

```
$ module avail
```

Kommandoen vil som standard liste noget lignende dette:

```
$ module avail
----- /usr/share/Modules/modulefiles -----
dot module-git module-info modules null use.own

----- /services/tools/modulefiles -----
anaconda3/2.1.0 intel/2022.2.0.191/init_opencl/latest prottest/3.4.2 samblaster/0.1.26
anaconda3/2.2.0 intel/2022.2.0.191/inspector/2022.1.0 pytorch/1.9.0 scapp/0.1.4
anaconda3/4.0.0 intel/2022.2.0.191/inspector/latest R/3.1.2 seqwish/0.7
anaconda3/4.4.0 intel/2022.2.0.191/itac/2021.6.0 R/3.2.0 shovill/1.1.0
anaconda3/2019.10 intel/2022.2.0.191/itac/latest R/3.2.1 simple/3.0.0
anaconda3/2020.07 intel/2022.2.0.191/mpi/2021.6.0 R/3.2.5 singularity/3.7.1
anaconda3/2021.05 intel/2022.2.0.191/mpi/latest R/3.3.1 singularity/3.7.3
anaconda3/2021.11 intel/2022.2.0.191/oclfpfga/2022.1.0 R/3.4.0 singularity/3.7.4
blas/gcc/64/3.8.0 intel/2022.2.0.191/oclfpfga/latest R/3.4.3 singularity/3.8.0
cm-eigen3/3.3.7 intel/2022.2.0.191/tbb/2021.6.0 R/3.5.0 singularity/3.8.3
dotnet-core/SDK-5.0.102 intel/2022.2.0.191/tbb/latest R/3.5.3 skesa/2.4.0
gcc/10.2.0 intel/2022.2.0.191/tbb32/2021.6.0 R/3.6.1 skesa/2.4.0_saute.1.3.0
go/1.14 intel/2022.2.0.191/tbb32/latest R/3.6.1-GCC-MKL smoothxg/0.4
htslib/1.9 lapack/gcc/64/3.8.0 R/4.0.0 snpeff/5.0e
htslib/1.12 miniconda3/4.9.2 R/4.0.0-GCC-MKL somalier/0.2.13
htslib/1.13 miniconda3/4.10.3 R/4.0.3 spyder/4.0.1
htslib/1.14 miniconda3/4.12.0 R/4.0.3-GCC-MKL spyder/4.1.1
intel/2022.2.0.191/clck/2021.6.0 openblas/dynamic(default) R/4.1.0 spyder/4.1.4
intel/2022.2.0.191/clck/latest openblas/dynamic/0.2.20 R/4.1.0-GCC-MKL spyder/5.0.1
intel/2022.2.0.191/compiler-rt/2022.1.0 openmpi/gcc/64/1.10.7 R_LEGACY/3.1.2 spyder/5.1.5
intel/2022.2.0.191/compiler-rt/latest panprova/20210914 R_LEGACY/3.2.0 spyder/5.2.2
intel/2022.2.0.191/compiler-rt32/2022.1.0 patric-cli/1.032 R_LEGACY/3.2.1 sratoolkit/2.10.9
intel/2022.2.0.191/compiler-rt32/latest patric-cli/1.035 rasusa/0.4.1 sv2/1.4.3.4
intel/2022.2.0.191/compiler/2022.1.0 pcaone/0.1.6 raxml-ng/1.1.0 svanalyzer/0.36
intel/2022.2.0.191/compiler/latest perl/5.8.9 readline/8.0 tiddit/2.12.1
intel/2022.2.0.191/compiler32/2022.1.0 perl/5.20.1(default) readline/8.1 trinotate/3.2.2
intel/2022.2.0.191/compiler32/latest perl/5.20.2 root/6.24.00 truvári/2.1
intel/2022.2.0.191/debugger/2021.6.0 perl/5.24.0 rseqc/4.0.0 vcflib/1.0.2
intel/2022.2.0.191/debugger/latest perl/5.30.2 rstudio/1.4.1103 vembrane/0.6.1
intel/2022.2.0.191/dev-utilities/2021.6.0 pgadmin/5.1 rstudio/1.4.1106 vg/1.32.0
intel/2022.2.0.191/dev-utilities/latest pgagent/4.2.1 saige/0.39.0 virtualgl/2.6.5
intel/2022.2.0.191/icc/2022.1.0 pgggb/20210419 saige/0.42.1-dev20201018 vscode/1.54.2
intel/2022.2.0.191/icc/latest pindel/0.2.5b8 saige/0.44.6.5 vscode/1.61.2
intel/2022.2.0.191/icc32/2022.1.0 plass/4.687d7 saige/0.44.6.5-dev20210819 wannier90/3.1.0
intel/2022.2.0.191/icc32/latest postgresql/12.0 saige/20220508 wfmash/0.4
intel/2022.2.0.191/init_opencl/2022.1.0 ppr-meta/1.1 sambamba/0.8.0
$
```

Regelmæssige brugere bør fokusere på moduler placeret under `/services/tools/modulefiles`. Disse kan indlæses ved hjælp af kommandoen:

```
$ module load module_name
```

hvor `module_name` kan være et af ovenstående moduler.

Du kan begrænse antallet af listede moduler ved at tilføje det eller de første bogstaver i modulnavnet til kommandoen modulet tilgængelig.

```
$ module /services/tools/modulefiles avail a
-----
anaconda3/2.1.0          anaconda3/4.0.0          anaconda3/2019.10      anaconda3/2021.05
anaconda3/2.2.0  anaconda3/4.4.0  anaconda3/2020.07  anaconda3/2021.11
```

## Module whatis

For at få grundlæggende information om modulfilerne kan du bruge `module whatis modulfil`.

```
$ module whatis gcc anaconda3/2021.11
```

```
----- /services/tools/modulefiles -----
gcc/10.2.0: GCC 10.2.0 - GNU Compiler Collection
----- /services/tools/modulefiles -----
anaconda3/2021.11: Anaconda3 2021.11 - Python 3.9 distribution for scientific computing
$
```

## Modules apropos

For at søge efter en bestemt streng i de grundlæggende oplysninger om modulfilerne, kan du bruge modulet `apropos streng`.

```
$ module apropos file
```

```
----- /usr/share/Modules/modulefiles -----
use.own: adds your own modulefiles directory to MODULEPATH
----- /services/tools/modulefiles -----
samblaster/0.1.26: samblaster 0.1.26 - tool to mark duplicates and extract discordant and split reads from sam files
sratoolkit/2.10.9: SRA Toolkit 2.10.9 - convert SRA files into other formats
vcflib/1.0.2: vcflib 1.0.2 - C++ library and cmdline tools for parsing and manipulating VCF files
$
```

## Module display

Yderligere information er tilgængelig med modul display *modulfil*.

```
$ module display anaconda3/2021.11
```

```
-----  
/services/tools/modulefiles/anaconda3/2021.11:
```

```
module-whatis  {Anaconda3 2021.11 - Python 3.9 distribution for scientific computing}  
conflict      anaconda3  
conflict      python2  
conflict      python36  
prepend-path  PATH /cm/shared/apps/anaconda3/2021.11/bin  
prepend-path  MANPATH /cm/shared/apps/anaconda3/2021.11/share/man  
setenv        ANACONDA3_HOME /cm/shared/apps/anaconda3/2021.11  
setenv        ANACONDAHOME /cm/shared/apps/anaconda3/2021.11  
setenv        PYTHONVERSION 3.9  
setenv        ANACONDASITEPACKAGES /cm/shared/apps/anaconda3/2021.11/lib/python3.9/site-packages  
setenv        GTDBTK_DATA_PATH /home/databases/gtdbtk/current  
setenv        RGTDATA /home/databases/rgtdata  
setenv        GUNC_DB /home/databases/gunc
```

Warning: Anaconda provides own tools such as GCC, Perl, R, which conflict with existing environment modules - as a rule of thumb you should load anaconda as early as possible. If in doubt, always try '\$ which gcc' (or perl or R).

To list currently installed Python modules, use command 'pip list' or 'findinanaconda [-i] <string>' (from 'computer\_ome\_utils' module)'

```
-----  
Denne kommando vil også vise conflicts (moduler der er i konflikt med dette) og prereq (moduler som skal indlæses for at bruge dette).
```

## Tilføjelse og fjernelse af moduler

Brug modulindlæsning for at tilføje et eller flere moduler: *Module load modulefile*.

```
$ module load anaconda3/2021.11
```

Så er Anaconda-miljøet tilgængeligt, og du kan køre Anaconda-kommandoer og starte Python

```
$ conda info
```

```
active environment : None
user config file : /root/.condarc
populated config files :
conda version : 4.10.3
conda-build version : 3.21.5
python version : 3.9.7.final.0
virtual packages : __linux=4.18.0=0
                  __glibc=2.28=0
                  __unix=0=0
                  __archspec=1=x86_64
base environment : /cm/shared/apps/anaconda3/2021.11 (read only)
conda av data dir : /cm/shared/apps/anaconda3/2021.11/etc/conda
conda av metadata url : None
channel URLs : https://repo.anaconda.com/pkgs/main/linux-64
              https://repo.anaconda.com/pkgs/main/noarch
              https://repo.anaconda.com/pkgs/r/linux-64
              https://repo.anaconda.com/pkgs/r/noarch
package cache : /cm/shared/apps/anaconda3/2021.11/pkgs
               /root/.conda/pkgs
envs directories : /root/.conda/envs
                  /cm/shared/apps/anaconda3/2021.11/envs
platform : linux-64
user-agent : conda/4.10.3 requests/2.26.0 CPython/3.9.7 Linux/4.18
.0-372.9.1.el8.x86_64 rhel/8.6 glibc/2.28
UID:GID : 0:0
netrc file : None
offline mode : False
```

Activate the base environment

```
source activate base
```

```
(base) $
```

Vis versionen af Python installeret i basismiljøet

```
$ python --version
Python 3.9.7
```

Moduler opført med versionsoplysninger (som gcc/4.8.2) refererer til en bestemt version af et værktøj. Generelt anbefales det altid at inkludere versionen, når et værktøj indlæses; hvis du ikke angiver version, vil (standard) blive indlæst, hvis nogen er tilgængelig - ellers vil kommandoen udskrive en fejl.

```
$ module load gcc/10.2.0
$ which gcc
/cm/shared/apps/gcc/10.2.0/bin/gcc
```

Moduler vil rapportere enhver konflikt (conflict) og manglende forudsætninger (prereq)

```
$ module load R/3.5.0
Loading R/3.5.0
ERROR: R/3.5.0 cannot be loaded due to missing prereq.
HINT: the following module must be loaded first: gcc
```

```
$ module load gcc/10.2.0
$ module load R/3.5.0
$ R --version
R version 3.5.0 (2018-04-23) -- "Joy in Playing"
Copyright (C) 2018 The R Foundation for Statistical Computing
Platform: x86_64-pc-linux-gnu (64-bit)
```

R is free software and comes with ABSOLUTELY NO WARRANTY.  
You are welcome to redistribute it under the terms of the  
GNU General Public License versions 2 or 3.  
For more information about these matters see  
<http://www.gnu.org/licenses/>.

#### *Module list*

Kontroller, hvilke moduler der er indlæst

```
$ module list
Currently Loaded Modulefiles:
 1) gcc/10.2.0  2) R/3.5.0
```

#### *Module switch*

For at skifte indlæst modulfil1 med modulfil2, brug switch modulfil1 modulfil2 Hvis modulfil1 ikke er angivet, antages det at være det aktuelt indlæste modul med samme rodnavn som modulfil2.

```
$ module load anaconda3/2020.07
$ module list
Currently Loaded Modulefiles:
 1) anaconda3/2020.07
$ module switch anaconda3/2021.11
$ module list
Currently Loaded Modulefiles:
 1) anaconda3/2021.11
$ module purge
$ module list
No Modulefiles Currently Loaded.
$
```

#### *Module unload*

For at fjerne et eller flere moduler, brug modulet unload modulefile



```
$ module list
Currently Loaded Modulefiles:
  1) gcc/10.2.0   2) R/3.5.0
```

```
$ module unload R/3.5.0
$ module list
Currently Loaded Modulefiles:
  1) gcc/10.2.0
```

### *Module purge*

Hvis alle moduler skal aflæses på én gang, skal du bruge:

```
$ module purge
$ module list
No Modulefiles Currently Loaded.
```

## Håndtering af moduler i dit login-miljø

Du bør holde antallet af moduler, du indlæser i dit login-miljø, på et absolut minimum. At have for mange moduler indlæst kan potentielt forårsage problemer med forudgående krav og/eller konflikter, som påvirker eller ødelægger dit miljø.

Det er muligt permanent at tilføje, manipulere og fjerne moduler i dit login-miljø ved at bruge modulet `init...`-familien af kommandoer.

Brug tabulatorfuldførelse (*tab*-completion) til at liste underkommandoer:

```
$ module init*tab*
initadd      initclear    initlist     initprepend  initrm
$ module init
```

### *Initadd modulefile*

Tilføj modulfil(er) til shellens initialiseringsfil i brugerens hjemmemappe.

### *Initprepend modulefile\**

Gør det samme som `initadd`, men sætter de givne moduler foran i begyndelsen af listen.

### *Initrm modulefile*

Fjern modulfil(er) fra shellens initialiseringsfiler.

### *Initswitch modulefile1 modulefile2*

Skift modulfil1 med modulfil2 i shellens initialiseringsfiler.

*Initlist*: Liste over alle de modulfiler, der er indlæst fra shellens initialiseringsfil.

*Initclear*: Ryd alle modulfilerne fra shellens initialiseringsfiler.

## Liste over moduler leveret af NGC (2022-08-21)

abricate	<b>anaconda3</b>	any2fasta	bgen	blas
boost	cm-eigen3	cuda10.2	dotnet-core	emboss

<b>gcc</b>	go	htslib	intel-clck	<b>intel-compiler</b>
<b>intel-compiler32</b>	<b>intel-compiler-rt</b>	<b>intel-compiler-rt32</b>	<b>intel-debugger</b>	<b>intel-dev-utilities</b>
<b>intel-icc</b>	<b>intel-icc32</b>	<b>intel-init_openc</b>	<b>inspector</b>	<b>intel-itac</b>
<b>intel-mpi</b>	<b>intel-oclfpga</b>	<b>intel-tbb</b>	<b>intel-tbb32</b>	jasminesv
julia	kma	kraken	lapack	lftp
libffi	libmaus2	lumpy	magma	manta
megahit	melt	metasnv	metasv	miniconda3
mono	mpich	msmc	multiqc	navn
ncbi-blast	ncurses	nextflow	nextstrain	noti
nuitka	nvidia-hpc-sdk	openblas	openjdk	openmpi
oracle_jdk	panaroo	pangolin	panprova	patric-cli
pcaone	perl	<b>pgadmin</b>	pgagent	pggb
pindel	plass	<b>postgresql</b>	ppr-meta	protest
pytorch	<b>R</b>	R_LEGACY	rasusa	raxml-ng
readline	regenie	root	rseqc	<b>rstudio</b>
saige	sambamba	samblaster	scapp	seqwish
shovill	simple	singularity	skesa	smoothxg
snpeff	somalier	<b>spyder</b>	sratoolkit	sv2
svanalyzer	tiddit	trinotate	truvari	vcflib
vembrane	vg	virtualgl	<b>vscode</b>	wannier90