1. Move course data to Taito work directory and Puhti scratch directory

1) Open WinSCP and connect to taito.csc.fi
2) Open the local directory to the left side.
3) On the right side, move to your work directory: `/wrk/training0[XX]` or `/wrk/[your_username]`
4) Move the exercises.zip file to Taito by dragging it from the left panel to the one on the right.
5) Open Putty, connect to taito.csc.fi, username: training0[XX]. When typing password no characters will show, but just keep typing.
6) Change your directory to the same: `cd /wrk/training0[XX]`
7) Unzip the file: `unzip exercises.zip`
8) Give all .R, .py and .sh -files executing rights, with selecting them all and right-clicking: Properties > Permissions -> Owner -> check the X (execute). Go through all folders.

➢ How many folders are in the training materials?

2. Download data from Internet

You may choose yourself also some other data sources, will not use these files later.

1) In Putty, Download a single file with wget:

```
mkdir data_download_test
cd data_download_test
wget http://wwwd3.ymparisto.fi/d3/gis_data/spesific/valuma.zip .
```

- Please notice the dot in the end of third row, it means that save the file to the current location, you may give there also some other folder.

2) Download a folder from FTP service with wget:

```
wget -r --no-parent -nd
ftp.funet.fi/pub/sci/geo/geodata/mml/hallintorajat_milj_tk/2017/ .
```

- -r download all files in this folder recursively
- --no-parent does not download files above the given folder
- -nd does not make similar folders to Taito

3) Download a lot of data from rsync service with rsync (actually just 1 file in this case):

```
rsync -a
rsync://rsync.nic.funet.fi/ftp/pub/sci/geo/geodata/mml/orto/etrs-tm35fin/mara_vv_25000_50/2016/N61/10m/1/ .
```

- wget always downloads everything, with rsync you can choose files based on name or type.

➢ What was the download speed?
➢ In which folder did you save the downloaded files?

1. Storing your data in Allas

From Puhti $SCRATCH-directory files are automatically removed after 90 days, so make sure to keep a copy of your important files in Allas.

1) Configure a-tools for Allas for your project and user name

```
module load allas
allas-conf
```

Give your password and the select the project by giving its number (1).

2) Upload a directory with a-put:

```
cd .. (so that you are back to exercise base directory)
a-put data_download_test
```

This will zst-zip the folder and move it to Allas, directory similar to Puhti. See the exact bucket name from command output.

3) See what you have now in Allas:

```
a-list
```

This lists the buckets of your project

```
a-list the_name_of_your_bucket
```

This lists the contents of a specific bucket

4) Download the files back from Allas:

```
mkdir ../download_data_copy
cd ../download_data_copy
a-get the_name_of_your_bucket/data_download_test.zst
```

This will download and unzip your data folder to the new folder

2. R example, calculating contours with RSAGA

We'll calculate contours based on a GeoTIFF image with RSAGA. Think that you have an R script that is ready on your laptop, and now you would like to run that in Taito.

First we will move your script to Taito and edit it to the new environment. Normally you will to edit at least the paths to files and folders. Then we will run the script interactively from Taito-shell. And finally we do the same/similar calculation as a batch job, also as array job and parallel job.

This example script's main tasks are:

- Create folders.
- Change the format of .tif files to SAGA format, because RSAGA accepts only own files.
- The file names of the .tif files to be processed are given as list inside the mapsheet.txt file.
- Calculate the contours and save them in Shape format

Run the script in Taito-shell

1) Log in to Taito-shell, using NoMachine client.
2) Open RStudio and QGIS from right mouse menu: Applications -> Taito-shell -> GIS
    - Note that rspatial-env / geo-env modules are loaded automatically.
3) We are using DEM data NLS 10m DEM already available in Taito's GIS data folder: /wrk/project_ogiir-csc/mml/dem10m/etrs-tm35fin-n2000/ So we do not need to download the input data. In QGIS, open one or a few DEM files to see the data. Or you can open /wrk/project_ogiir-csc/mml/dem10m_vrt/etrs-tm35fin-n2000/whole_finland_hierarchical.vrt to see all Finland.
4) In RStudio, open the script: ../R/01_serial/Contours_simple.R
5) Use the R console to check that needed R libraries are available in Taito. Which libraries are used in this script? Check whether those libraries are available in RStudio, with (change to correct packages): require(rgdal)

6) In RStudio, change the file paths in the R script to fit your Taito environment. Save your changes.
   `/wrk/training0[XX]/R/01_serial`
7) Run your script from RStudio.
8) You should see new files in your work directory, check with ls –l in Putty or click the refresh button in WinSCP:
   - 3 countour shapefiles in 2_shape folder.
   - temporary SAGA format files are saved to 1_grid directory 1_grid
9) Check your results with QGIS

Simple batch job script in Taito.

For scripts that take longer and need more cores or memory than what is available in Taito-shell, you have to use Taito's batch system for requesting the resources and running your script. Also if you want to do parallel processing with snow you have to run your code through the batch job system.

Now we will run the previous script as batch job.

1) In RStudio, open batch_job_serial.sh, edit the path to your R script and save the file. In batch job file, note that you:
   - load the rspatial-env module before starting the R script
   - Reserve 1 core, 5 minutes time and 1GB memory.
2) In Putty move to your work directory. Submit the job to Taito:
   `sbatch batch_job_serial.sh`
   - You should get similar files to the interactive run, and additionally
   - err.txt and out.txt – command-line outputs and error messages. If your job fails, look for reasons from both of these files.
3) See the status of your job
   `$ squeue -u your_username`
4) See how much resources were used, see the ID from output of the sbatch command:
   `seff ID_of_your_job`
   `sacct -j <SLURM_JOBID> -o elapsed,TotalCPU,reqmem,maxrss,AllocCPUS`

- elapsed – time used by the job
- TotalCPU – time used by all cores together
- reqmem – amount of requested memory
- maxrss – maximum resident set size of all tasks in job.
- AllocCPUS – how many CPUs were used

➢ Did you reserve a good amount of memory?

Array job

The idea is that this script will be run by one process for every given input file as opposed to running a for loop within the script. The R script takes the file to be calculated as a parameter, which is defined inside the batch job file. The mapsheets.txt file includes the file names for the files that should be processed.

1) Open the mapsheets.txt. How many files there is?
2) In RStudio, open both files from 02_array folder.
3) Check the file paths in both scripts.
4) In batch job file, how many cores, time and memory you reserve? (No changes needed.)
5) Submit the job to Taito
6) Check with `seff` and `sacct` how much time and resources you used?

➢ Check with sacct how much memory and time was consumed?

Parallel job with snow

Next we use snow package for parallel processing.

1) In RStudio, open both files from 03_parallel folder.
2) In RStudio, check the file paths in both scripts.
3) In batch job file, how many cores, time and memory you reserve? (No changes needed.)
    - Note, the R script is run with `srun RMPISNOW --no-save -f contours.R` command. Because we have to use the RMPISNOW command we can't run the code from Rstudio.
4) Submit the job to Taito
5) Check with `seff` and `sacct` how much time and resources you used?

3. Python example, calculate depressions and high points from DEM with rasterio

This exercise is similar to the R one, we do the same analysis in 4 different way: interactively in Taito-shell, as serial batch job, array job and parallel job in Taito. The task in exercise is to locate depressions and high points from an elevation model. The way to do this is by applying a focal mean to the elevation model and then calculating the difference between the smoothed raster (where each cell contains a mean of surrounding area) and the actual elevation model. This results in a raster where cells that are higher than surrounding cells get a positive value and cells below their neighbors get a negative value.

One advantage of multiprocessing approach over array jobs is that if we wanted to for example combine our resulting files and then do some further processing with the combined file we could easily do this. Also with multiprocessing approach, it would be easy to split one large file into smaller chunks and then process those in parallel rather than operating with multiple files.

Basic idea behind the script is to:

- Read elevation model as a *numpy* array with *rasterio*
- Create a suitable kernel that results in mean of surrounding area being computed
- Apply sliding mean to your elevation array using your kernel
- Subtract the resulting array from array that contains the original elevation model
- Save output with *rasterio*

Interactive job

Think that you have a Python script that is ready on your laptop, and now you would like to run that in Taito. In this exercise, the script is already provided and your task is to just to slightly modify it and run it in Taito.

1) Open Spyder in NoMachine, from right mouse menu: Applications -> Taito-shell -> GIS
   - Note that geoconda modules are loaded automatically.
2) Check that needed Python libraries are available in Taito, with running the import commands in the beginning of script. Which libraries are used in this script?
3) In Spyder, change the file paths in the Python script.
4) Run the script as whole or in parts from Spyder.
5) You should see new files in your work directory, check with ls –l in Putty or click the refresh button in WinSCP:
   - 3 new tif-files in 01_serial folder.
   - error.txt and output.txt – command-line outputs and error messages. If your job fails, look for reasons from both of these files.
6) Check your results with QGIS

Serial job

1) The used mapsheets.txt is the same as for R-script, simply listing the processed files.
2) In Spyder, check the batch job script, especially the paths.
   - Note, that you use the geoconda module
   - Note, if you want to run Python examples in Puhti,
     - add to batch job files: `#SBATCH --account=project_2002044`
     - use mapsheets_puhti.txt file instead of the mapsheets.txt
3) Submit the job to Taito
4) Check with `seff` and `sacct` how much time and resources you used?

Array job

Run the same script as array job. The idea is that the batch job script gives each array job one raster file to be calculated. In Python script the file names are read from arguments.

1) In Spyder, open both files from 02_array folder.
2) Check the file paths in both scripts.
3) In batch job file, how many cores, time and memory did you reserve? (No changes needed.)
4) Submit the job to Taito
5) Check with `seff` and `sacct` how much time and resources you used?

Parallel job with multiprocessing

Use the serial job script as a skeleton for parallel job script. We use multiprocessing package.

1) In Spyder, open both files from 03_parallel folder.
2) Check the file paths in both scripts.
3) In batch job file, how many cores, time and memory you reserve? (No changes needed.)

4) Submit the job to Taito
5) Check with `seff` and `sacct` how much time and resources you used?

4. Change coordinate system of many files with GDAL.

GDAL/OGR provide many useful tools. In this exercise, we will reproject the coordinate system of multiple files in a folder, and add overviews to the same files. We do not use R nor Python, but a simple Linux bash file.

1) Open the gdal.sh file with gedit/RStudio.
2) Fix the paths and save the file.
3) As there is not many files in the given folder and running the script takes just some seconds, we can run it in Taito-shell.
4) Make sure that you can use GDAL tools, ie. have appropriate modules loaded (you should have from previous exercise). Easiest option is to execute `gdalinfo` without any parameters, it should provide help how to use it.
5) Check the original file with gdalinfo. What is the coordinate system? Are the files tiled? Do they have overviews?

```
gdalinfo /appl/data/geo/mml/dem10m/etrs-tm35fin-n2000/W3/W33/W3333.tif
```

6) Change the permissions of gdal.sh, so that it can be executed: `chmod 700 gdal.sh`
7) Run the script: `./gdal.sh`

➢ Check the result file with gdalinfo. What is the coordinate system? Are the files tiled? Do they have overviews?

If you would have more files you should not use Taito-shell for such work, rather write also a SLURM batch job script and use the same script in Taito.

5. Save your results, disconnect

Move the results back to your local machine

1. Go back to WinSCP
2. Refresh your file listing on right side.
3. Drag the output-files from right to left.

Close all tools

1. In Putty: `exit`
2. WinScp you can close from top right corner X.
3. NoMachine you can also close from the top right corner X, but it will ask you if you want to disconnect (= keep your session running at Taito-shell) or Terminate (= close everything).

Links to important support pages

General

- User accounts: https://docs.csc.fi/#accounts/
- Trainings, inc materials of past events: https://www.csc.fi/web/training
- Geocomputing: https://research.csc.fi/geocomputing
- Virtual rasters: https://research.csc.fi/virtual_rasters
- Linux: https://research.csc.fi/csc-guide-linux-basics-for-csc
    - Unix cheat sheet: https://research.csc.fi/csc-guide-appendixes

Puhti and Taito

- Taito user guide: https://research.csc.fi/taito-user-guide
- Puhti user guide: https://docs.csc.fi/#computing/overview/
- NoMachine:
    - Taito: https://research.csc.fi/csc-guide-connecting-the-servers-of-csc#1.3.3
- Directories and data storage:
    - Taito: https://research.csc.fi/csc-guide-directories-and-data-storage-at-csc
    - Puhti: https://docs.csc.fi/#computing/disk/
- Batch jobs:
    - Taito: https://research.csc.fi/taito-batch-jobs
    - Puhti: https://docs.csc.fi/#computing/running/creating-job-scripts/
- Software (and modules):
    - Taito: https://research.csc.fi/software -> Geosciences
    - Puhti: https://docs.csc.fi/#apps/ -> Geosciences
- Taito GIS data: https://research.csc.fi/gis_data_in_taito
- Code examples for geocomputing: https://github.com/csc-training/geocomputing