

Montreal-forced-aligner installation

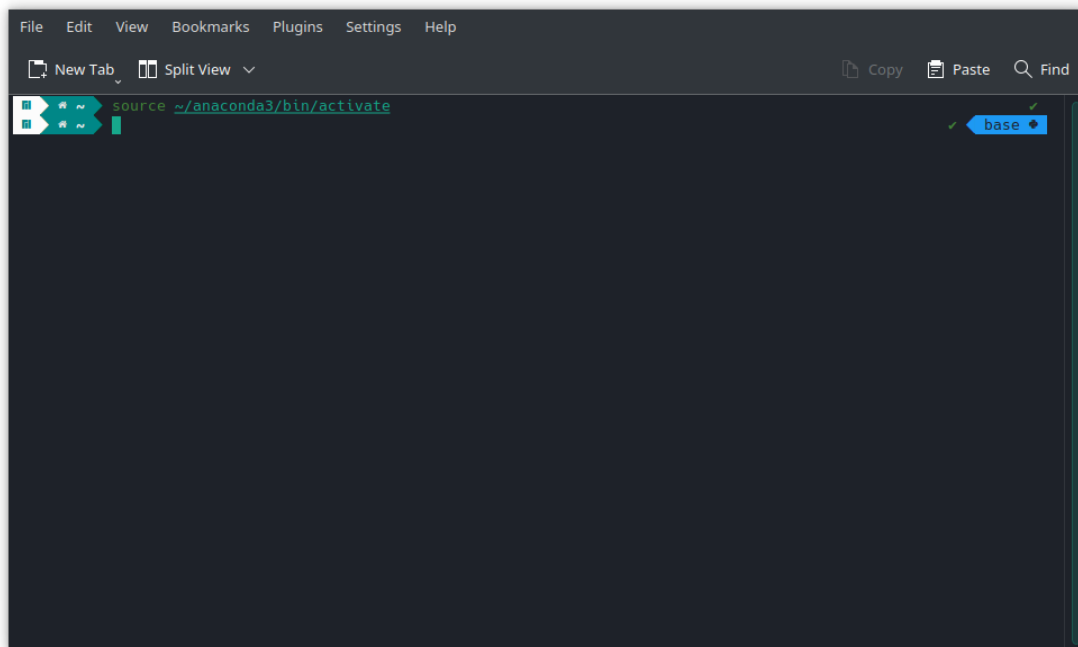
1) Prerequisite

This notebook contains the basic steps to install the Montreal-forced-aligner (MFA) on your computer. It is assumed that a working Python distribution is already on your computer. I recommend to use Anaconda that can be installed for Windows, Mac and Linux: <https://www.anaconda.com/products/distribution>

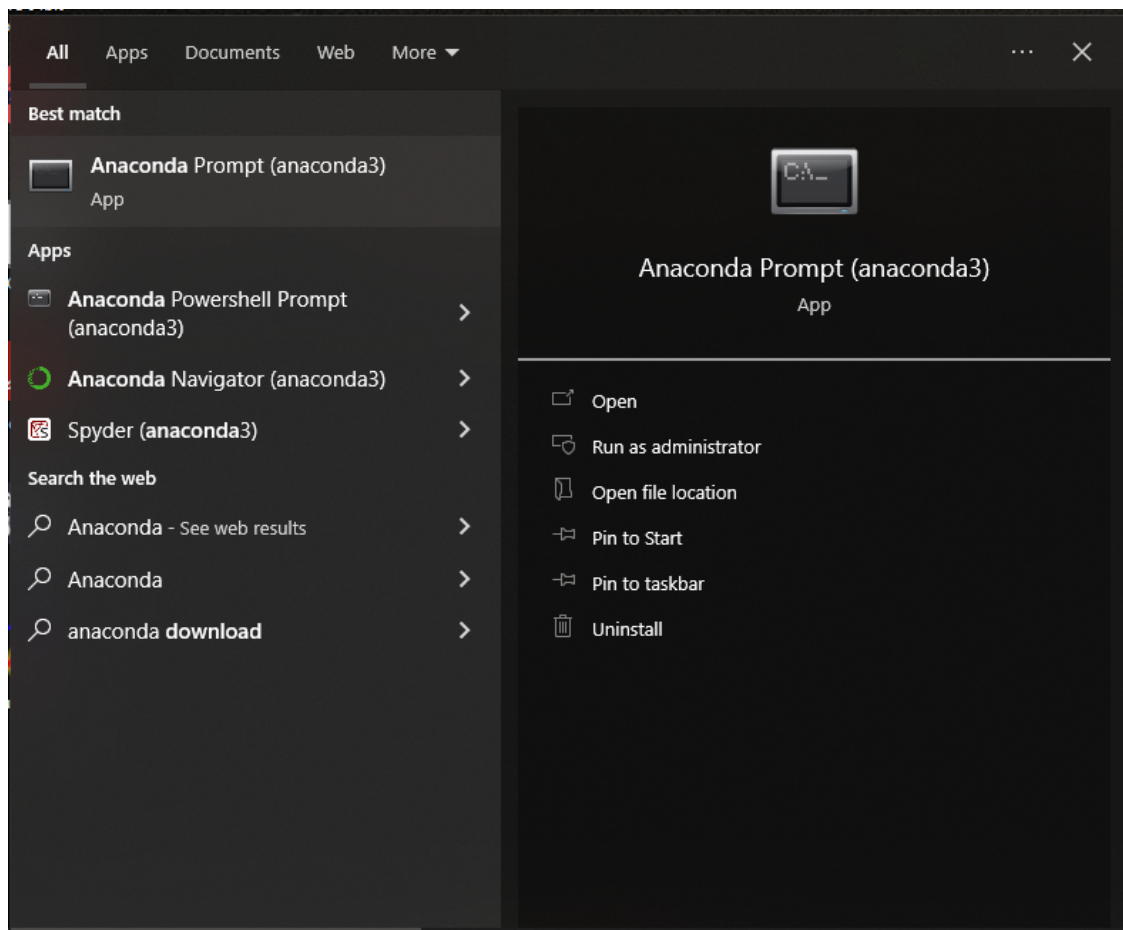
2) Open anaconda

After the successful installation, open the terminal/console/command line of your operating system.

- On Linux and Mac, you should see automatically the term “(base)” at the end or beginning of the input line. (Note: `source ~/anaconda3/bin/activate` has to be entered on Linux systems only if anaconda is not automatically initialized)



- On Windows, search for “Anaconda Prompt (anaconda3)”



and execute the App. You should see now a command line:



3) Set up an environment

A good feature of Anaconda is its easy handling of environments. (Virtual) environments have their own Python interpreter, libraries and scripts and are isolated from other environments. In case a program does not run from a specific environment due to broken packages, you can simply delete it and set up a new one with more stable packages.

If you were able to install the MFA in Session 1, but it did not work (properly), remove your environment via the following command:

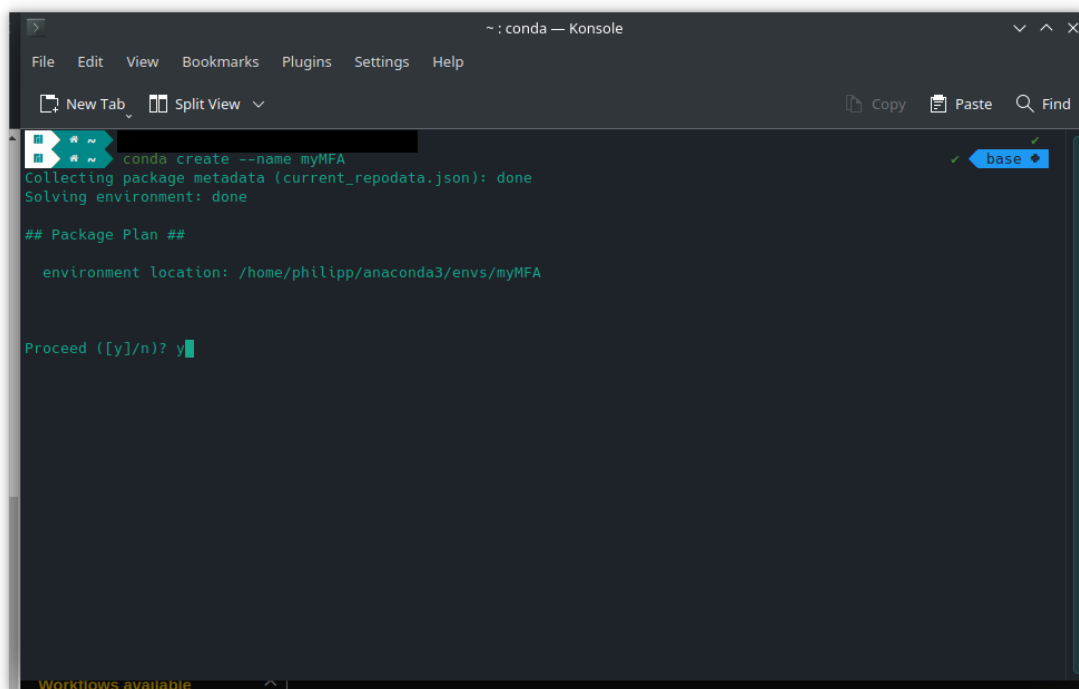
```
[ ]: conda remove --name name_of_your_environment --all
```

You can use this command whenever you want to remove an environment.

In order to create a new environment, type in (you can use any name instead of “myMFA”):

```
[ ]: conda create --name myMFA
```

You should see now the following output. Enter y.



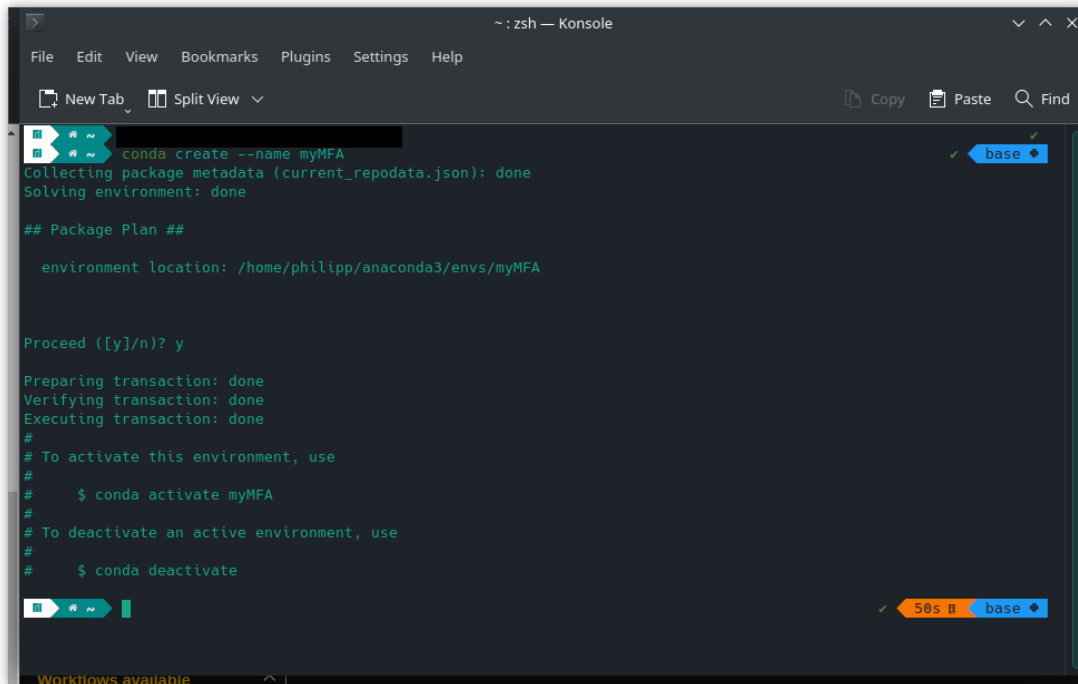
```
~ : conda — Konsole
File Edit View Bookmarks Plugins Settings Help
New Tab Split View
conda create --name myMFA
Collecting package metadata (current_repodata.json): done
Solving environment: done

## Package Plan ##

  environment location: /home/philipp/anaconda3/envs/myMFA

Proceed ([y]/n)? y
```

The environment should be installed now:



```
~: zsh — Konsole
File Edit View Bookmarks Plugins Settings Help
New Tab Split View
Copy Paste Find
conda create --name myMFA
Collecting package metadata (current_repodata.json): done
Solving environment: done

## Package Plan ##

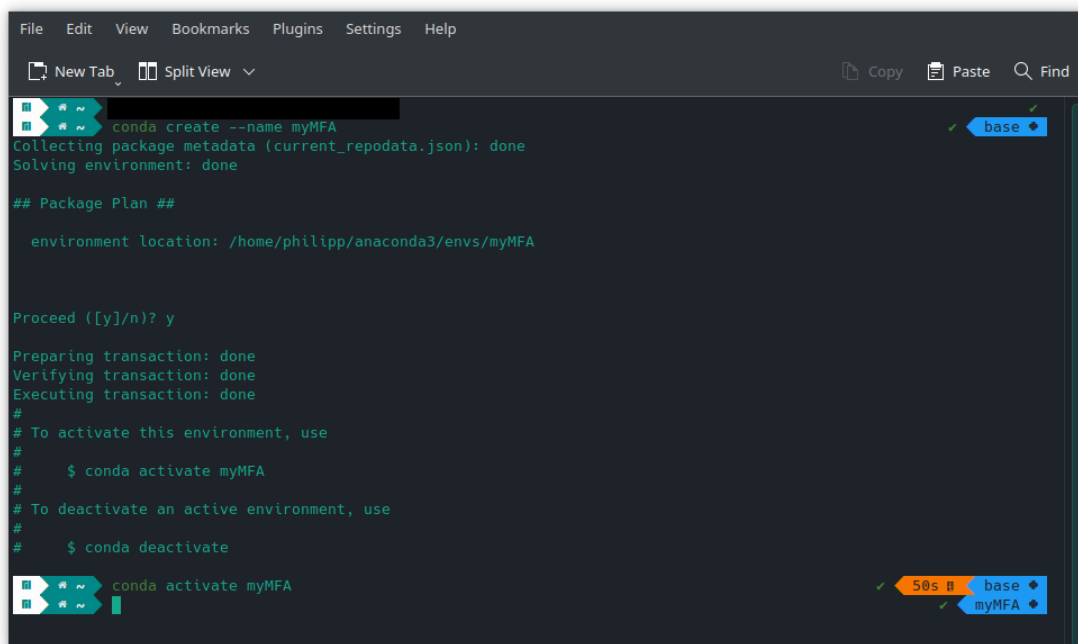
  environment location: /home/philipp/anaconda3/envs/myMFA

Proceed ([y]/n)? y
Preparing transaction: done
Verifying transaction: done
Executing transaction: done
#
# To activate this environment, use
#
#   $ conda activate myMFA
#
# To deactivate an active environment, use
#
#   $ conda deactivate
50s B base
```

The environment has to be activated by the following command:

```
[ ]: conda activate myMFA
```

You can see that the environment is activated if you see “(myMFA)” instead of “(base)”:



```
File Edit View Bookmarks Plugins Settings Help
New Tab Split View
Copy Paste Find
conda create --name myMFA
Collecting package metadata (current_repodata.json): done
Solving environment: done

## Package Plan ##

  environment location: /home/philipp/anaconda3/envs/myMFA

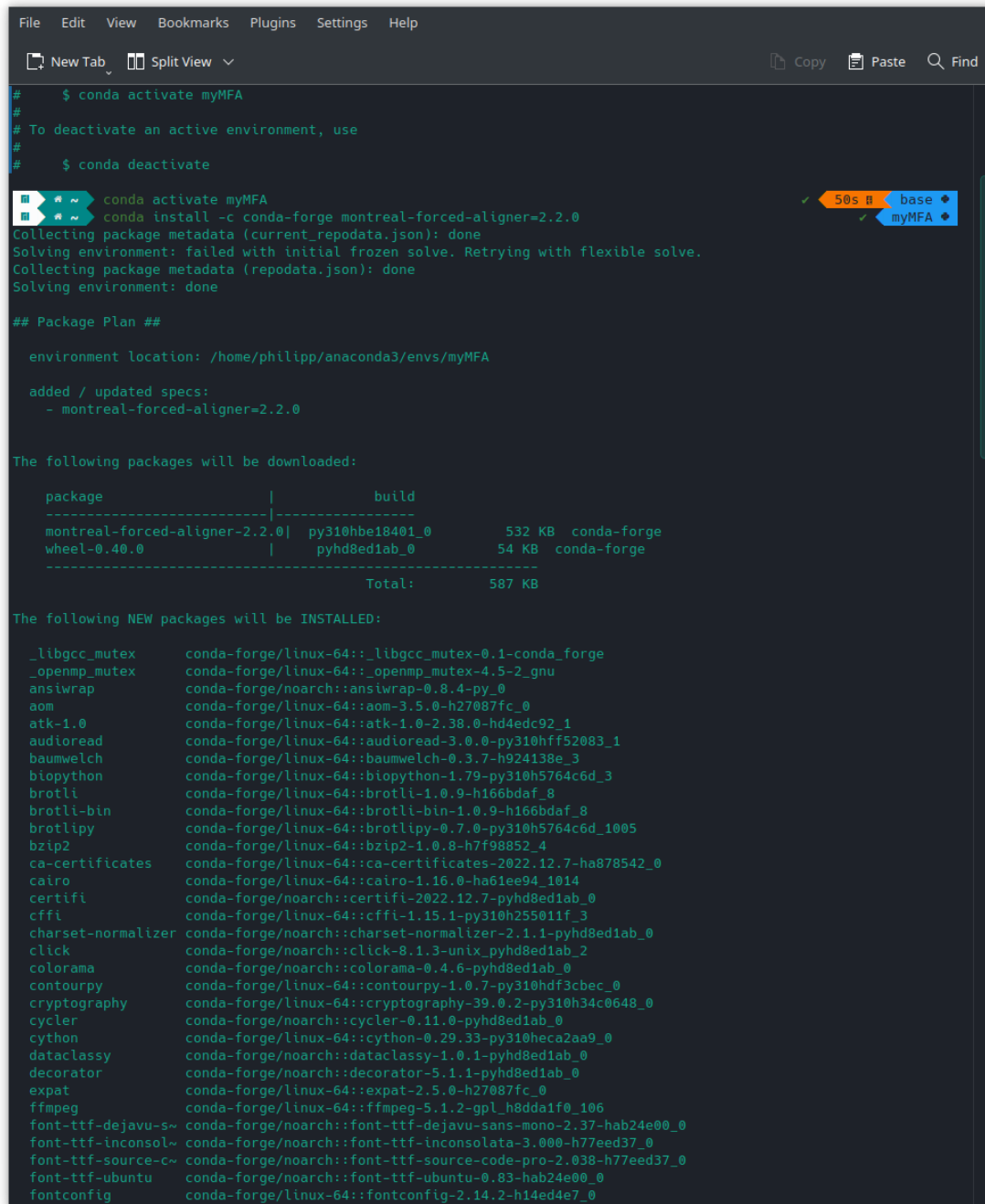
Proceed ([y]/n)? y
Preparing transaction: done
Verifying transaction: done
Executing transaction: done
#
# To activate this environment, use
#
#   $ conda activate myMFA
#
# To deactivate an active environment, use
#
#   $ conda deactivate
conda activate myMFA
50s B base
myMFA
```

4) Install the MFA

If your environment is activated, enter the following command to install the MFA:

```
[ ]: conda install -c conda-forge montreal-forced-aligner=2.2.0
```

The “=2.2.0” forces conda to install a specific version of the MFA. Although the MFA was updated recently, there seem to be certain issues of dependend packages that are installed along, so we will install the older version 2.2.0.



```
File Edit View Bookmarks Plugins Settings Help
New Tab Split View
Copy Paste Find

# $ conda activate myMFA
#
# To deactivate an active environment, use
#
# $ conda deactivate

# conda activate myMFA
# conda install -c conda-forge montreal-forced-aligner=2.2.0
Collecting package metadata (current_repodata.json): done
Solving environment: failed with initial frozen solve. Retrying with flexible solve.
Collecting package metadata (repodata.json): done
Solving environment: done

## Package Plan ##

environment location: /home/philipp/anaconda3/envs/myMFA

added / updated specs:
- montreal-forced-aligner=2.2.0

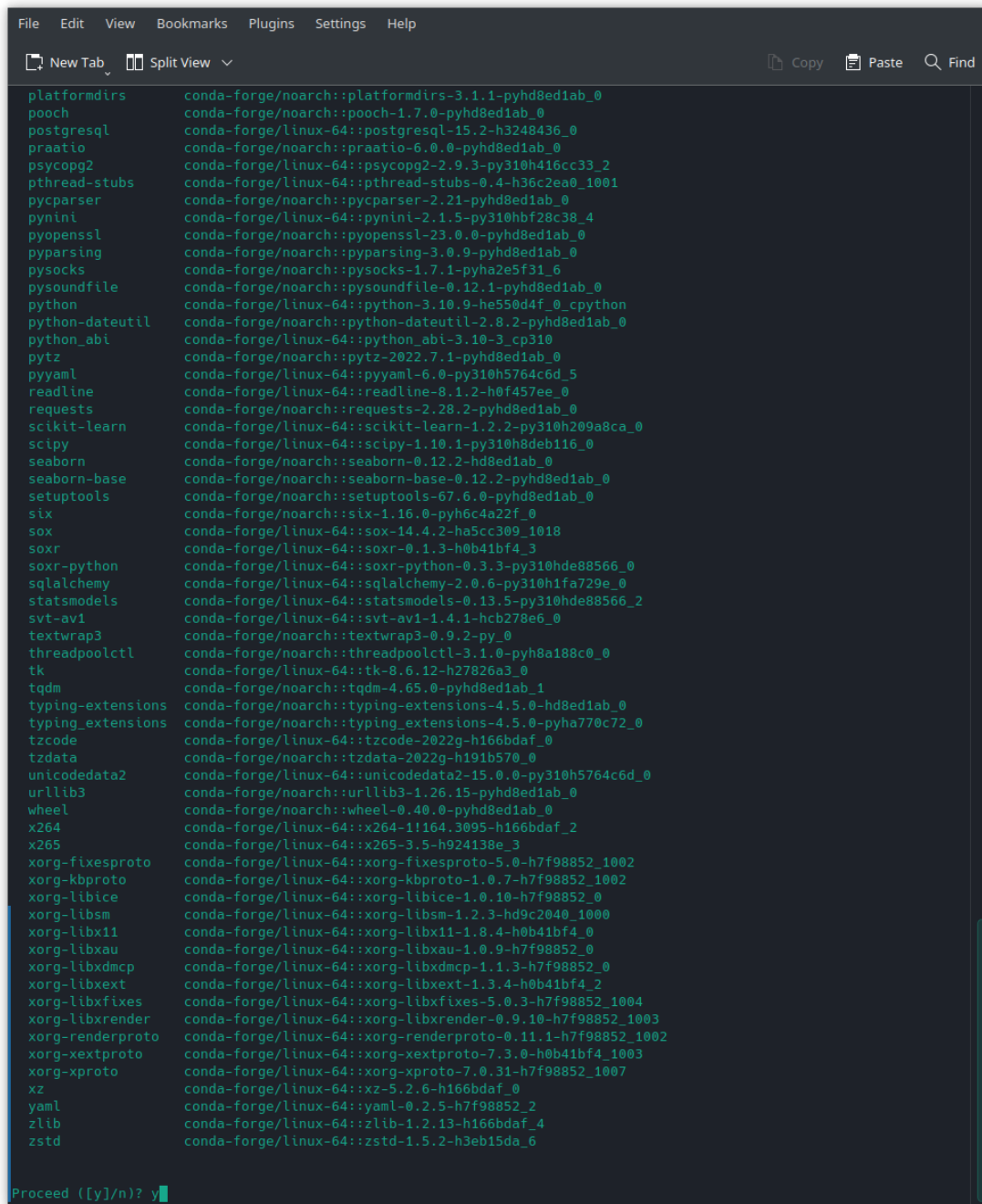
The following packages will be downloaded:

package | build
-----|-----
montreal-forced-aligner-2.2.0 | py310hbe18401_0 532 KB conda-forge
wheel-0.40.0 | pyhd8ed1ab_0 54 KB conda-forge
-----|-----
Total: 587 KB

The following NEW packages will be INSTALLED:

_libgcc_mutex conda-forge/linux-64::_libgcc_mutex-0.1-conda_forge
_openmp_mutex conda-forge/linux-64::_openmp_mutex-4.5-2_gnu
ansiwrap conda-forge/noarch::ansiwrap-0.8.4-py_0
aom conda-forge/linux-64::aom-3.5.0-h27087fc_0
atk-1.0 conda-forge/linux-64::atk-1.0-2.38.0-hd4edc92_1
audioread conda-forge/linux-64::audioread-3.0.0-py310hff52083_1
baumwelch conda-forge/linux-64::baumwelch-0.3.7-h924138e_3
biopython conda-forge/linux-64::biopython-1.79-py310h5764c6d_3
brotli conda-forge/linux-64::brotli-1.0.9-h166bdaf_8
brotli-bin conda-forge/linux-64::brotli-bin-1.0.9-h166bdaf_8
brotlipy conda-forge/linux-64::brotlipy-0.7.0-py310h5764c6d_1005
bzip2 conda-forge/linux-64::bzip2-1.0.8-h7f98852_4
ca-certificates conda-forge/linux-64::ca-certificates-2022.12.7-ha878542_0
cairo conda-forge/linux-64::cairo-1.16.0-ha61ee94_1014
certifi conda-forge/noarch::certifi-2022.12.7-pyhd8ed1ab_0
cffi conda-forge/linux-64::cffi-1.15.1-py310h255011f_3
charset-normalizer conda-forge/noarch::charset-normalizer-2.1.1-pyhd8ed1ab_0
click conda-forge/noarch::click-8.1.3-unix_pyhd8ed1ab_2
colorama conda-forge/noarch::colorama-0.4.6-pyhd8ed1ab_0
contourpy conda-forge/linux-64::contourpy-1.0.7-py310hdf3cbe_0
cryptography conda-forge/linux-64::cryptography-39.0.2-py310h34c0648_0
cycler conda-forge/noarch::cycler-0.11.0-pyhd8ed1ab_0
cython conda-forge/linux-64::cython-0.29.33-py310heca2aa9_0
dataclassy conda-forge/noarch::dataclassy-1.0.1-pyhd8ed1ab_0
decorator conda-forge/noarch::decorator-5.1.1-pyhd8ed1ab_0
expat conda-forge/linux-64::expat-2.5.0-h27087fc_0
ffmpeg conda-forge/linux-64::ffmpeg-5.1.2-gpl_h8dda1f0_106
font-ttf-dejavu-s~ conda-forge/noarch::font-ttf-dejavu-sans-mono-2.37-hab24e00_0
font-ttf-inconsol~ conda-forge/noarch::font-ttf-inconsolata-3.000-h77eed37_0
font-ttf-source-c~ conda-forge/noarch::font-ttf-source-code-pro-2.038-h77eed37_0
font-ttf-ubuntu conda-forge/noarch::font-ttf-ubuntu-0.83-hab24e00_0
fontconfig conda-forge/linux-64::fontconfig-2.14.2-h14ed4e7_0
```

It may be that you see a message like: “Solving environment: failed with initial frozen solve. Retrying with flexible solve.” Just let the program run.



```
File Edit View Bookmarks Plugins Settings Help
New Tab Split View
Copy Paste Find

platformdirs conda-forge/noarch::platformdirs-3.1.1-pyhd8ed1ab_0
pooch conda-forge/noarch::pooch-1.7.0-pyhd8ed1ab_0
postgresql conda-forge/linux-64::postgresql-15.2-h3248436_0
praatio conda-forge/noarch::praatio-6.0.0-pyhd8ed1ab_0
psycpg2 conda-forge/linux-64::psycpg2-2.9.3-py310h416cc33_2
pthread-stubs conda-forge/linux-64::pthread-stubs-0.4-h36c2ea0_1001
pyparser conda-forge/noarch::pyparser-2.21-pyhd8ed1ab_0
pynini conda-forge/linux-64::pynini-2.1.5-py310hbf28c38_4
pyopenssl conda-forge/noarch::pyopenssl-23.0.0-pyhd8ed1ab_0
pyparsing conda-forge/noarch::pyparsing-3.0.9-pyhd8ed1ab_0
pysocks conda-forge/noarch::pysocks-1.7.1-pyha2e5f31_6
pysoundfile conda-forge/noarch::pysoundfile-0.12.1-pyhd8ed1ab_0
python conda-forge/linux-64::python-3.10.9-he550d4f_0_cpython
python-dateutil conda-forge/noarch::python-dateutil-2.8.2-pyhd8ed1ab_0
python_abi conda-forge/linux-64::python_abi-3.10-3_cp310
pytz conda-forge/noarch::pytz-2022.7.1-pyhd8ed1ab_0
pyyaml conda-forge/linux-64::pyyaml-6.0-py310h5764c6d_5
readline conda-forge/linux-64::readline-8.1.2-h0f457ee_0
requests conda-forge/noarch::requests-2.28.2-pyhd8ed1ab_0
scikit-learn conda-forge/linux-64::scikit-learn-1.2.2-py310h209a8ca_0
scipy conda-forge/linux-64::scipy-1.10.1-py310h83deb116_0
seaborn conda-forge/noarch::seaborn-0.12.2-hd8ed1ab_0
seaborn-base conda-forge/noarch::seaborn-base-0.12.2-pyhd8ed1ab_0
setuptools conda-forge/noarch::setuptools-67.6.0-pyhd8ed1ab_0
six conda-forge/noarch::six-1.16.0-pyh6c4a22f_0
sox conda-forge/linux-64::sox-14.4.2-ha5cc309_1018
soxr conda-forge/linux-64::soxr-0.1.3-h0b41bf4_3
soxr-python conda-forge/linux-64::soxr-python-0.3.3-py310hde88566_0
sqlalchemy conda-forge/linux-64::sqlalchemy-2.0.6-py310h1fa729e_0
statsmodels conda-forge/linux-64::statsmodels-0.13.5-py310hde88566_2
svt-av1 conda-forge/linux-64::svt-av1-1.4.1-hcb278e6_0
textwrap3 conda-forge/noarch::textwrap3-0.9.2-py_0
threadpoolctl conda-forge/noarch::threadpoolctl-3.1.0-pyh8a188c0_0
tk conda-forge/linux-64::tk-8.6.12-h27826a3_0
tqdm conda-forge/noarch::tqdm-4.65.0-pyhd8ed1ab_1
typing-extensions conda-forge/noarch::typing-extensions-4.5.0-hd8ed1ab_0
typing_extensions conda-forge/noarch::typing_extensions-4.5.0-pyha770c72_0
tzcode conda-forge/linux-64::tzcode-2022g-h166bdaf_0
tzdata conda-forge/noarch::tzdata-2022g-h191b570_0
unicodedata2 conda-forge/linux-64::unicodedata2-15.0.0-py310h5764c6d_0
urllib3 conda-forge/noarch::urllib3-1.26.15-pyhd8ed1ab_0
wheel conda-forge/noarch::wheel-0.40.0-pyhd8ed1ab_0
x264 conda-forge/linux-64::x264-11164.3095-h166bdaf_2
x265 conda-forge/linux-64::x265-3.5-h924138e_3
xorg-fixesproto conda-forge/linux-64::xorg-fixesproto-5.0-h7f98852_1002
xorg-kbproto conda-forge/linux-64::xorg-kbproto-1.0.7-h7f98852_1002
xorg-libice conda-forge/linux-64::xorg-libice-1.0.10-h7f98852_0
xorg-libsm conda-forge/linux-64::xorg-libsm-1.2.3-hd9c2040_1000
xorg-libx11 conda-forge/linux-64::xorg-libx11-1.8.4-h0b41bf4_0
xorg-libxau conda-forge/linux-64::xorg-libxau-1.0.9-h7f98852_0
xorg-libxdmcp conda-forge/linux-64::xorg-libxdmcp-1.1.3-h7f98852_0
xorg-libxext conda-forge/linux-64::xorg-libxext-1.3.4-h0b41bf4_2
xorg-libxfixes conda-forge/linux-64::xorg-libxfixes-5.0.3-h7f98852_1004
xorg-libxrender conda-forge/linux-64::xorg-libxrender-0.9.10-h7f98852_1003
xorg-renderproto conda-forge/linux-64::xorg-renderproto-0.11.1-h7f98852_1002
xorg-xextproto conda-forge/linux-64::xorg-xextproto-7.3.0-h0b41bf4_1003
xorg-xproto conda-forge/linux-64::xorg-xproto-7.0.31-h7f98852_1007
xz conda-forge/linux-64::xz-5.2.6-h166bdaf_0
yaml conda-forge/linux-64::yaml-0.2.5-h7f98852_2
zlib conda-forge/linux-64::zlib-1.2.13-h166bdaf_4
zstd conda-forge/linux-64::zstd-1.5.2-h3eb15da_6

Proceed ([y]/n)? y
```

Proceed by entering y. You can confirm whether the MFA is installed on your computer by entering:

```
[ ]: conda list
```

A list of all packages installed in your environment is prompted. You will find the MFA there:

File	Edit	View	Bookmarks	Plugins	Settings	Help
New Tab	Split View				Copy	Paste Find
libtasn1	4.19.0	h166bdaf_0	conda-forge			
libtiff	4.5.0	h6adf6a1_2	conda-forge			
libtool	2.4.7	h27087fc_0	conda-forge			
libunistring	0.9.10	h7f98852_0	conda-forge			
libuuid	2.32.1	h7f98852_1000	conda-forge			
libva	2.17.0	h0b41bf4_0	conda-forge			
libvorbis	1.3.7	h9c3ff4c_0	conda-forge			
libvpx	1.11.0	h9c3ff4c_3	conda-forge			
libwebp	1.2.4	h1daa5a0_1	conda-forge			
libwebp-base	1.2.4	h166bdaf_0	conda-forge			
libxcb	1.13	h7f98852_1004	conda-forge			
libxml2	2.10.3	h7463322_0	conda-forge			
libzlib	1.2.13	h166bdaf_4	conda-forge			
llvmlite	0.39.1	py310h58363a5_1	conda-forge			
lmad	0.15.1b	h9c3ff4c_1	conda-forge			
matplotlib-base	3.7.1	py310he60537e_0	conda-forge			
montreal-forced-aligner	2.2.0	py310hbe18401_0	conda-forge			
mpg123	1.31.2	hcb278e6_0	conda-forge			
msgpack-python	1.0.5	py310hdf3cbec_0	conda-forge			
munkres	1.1.4	pyh9f0ad1d_0	conda-forge			
ncurses	6.3	h27087fc_1	conda-forge			
nettle	3.8.1	hc379101_1	conda-forge			
ngram	1.3.14	h924138e_2	conda-forge			
numba	0.56.4	py310ha5257ce_0	conda-forge			
numpy	1.23.5	py310h53a5b5f_0	conda-forge			
openfst	1.8.2	h924138e_2	conda-forge			
openh264	2.3.1	hcb278e6_2	conda-forge			
openjpeg	2.5.0	hfec8fc6_2	conda-forge			
openssl	3.0.8	h0b41bf4_0	conda-forge			