BIOCONDA

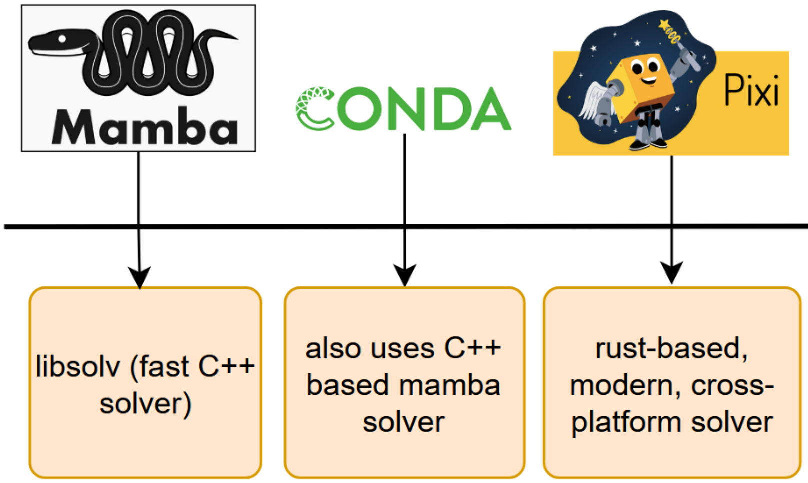
samtools
pysam
htslib
bcftools
bedtools
blast
bowtie2
star

conda-forge

numpy
scipy
pandas
requests
matplotlib
pytorch
tensorflow
python

channels/repositories

Locations where tools like conda, mamba, or pixi search for packages to install in environments



Mamba

libsolv (fast C++ solver)

CONDA

also uses C++ based mamba solver

Pixi

rust-based, modern, cross-platform solver

tools

programs that create, manage, and use environments. They decide what to do (install, activate, export). They also include "solvers".

solvers

The “engine” inside the tool that resolves dependencies and decides which versions of packages can work together.

MINIFORGE

ANACONDA

MINICONDA

assemblies/installers

Bundles that ship the tools and a starting environment. Different flavors, same core idea.

samtools.yml

samtools
htslib
bwa-mem

simple-env

fastqc
multiqc
quast

environments

Isolated workspaces containing specific versions of Python and tools

samtools.yml

```
name: samtools-env
channels:
  - conda-forge
  - bioconda
  - defaults
dependencies:
  - python=3.11
  - samtools
  - bwa-mem2
```

spec files

A YAML file describing an environment. You can share & reproduce an environment exactly, down to the version of all dependencies.

Using conda on clusters

```
module load conda
conda info  # verify path/version
mamba info # mamba is included
```

Version pinning

```
conda install numpy=1.11 # fuzzy match (1.11.*)
conda install numpy==1.11 # exact match
conda install "numpy>=1.11,<2" # range match
```

Change default environment location

```
# add these lines to `~/.condarc`
# create dir: /depot/projectspace/user/conda
envs_dirs:
  - /depot/projectspace/user/conda/envs
  - ~/.conda/envs      # fallback
pkgs_dirs:
  - /depot/projectspace/user/conda/pkgs
  - ~/.conda/pkgs      # fallback
```

Installing and managing packages

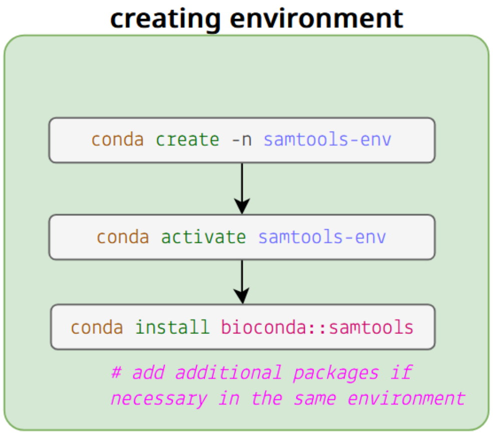
```
conda create -n aligners-env # create an env
conda activate aligners-env # activate env
conda install bioconda::bwa-mem2 # from specific channel
conda update hisat2  # update a package
conda remove hisat2  # remove package
```

Environment management

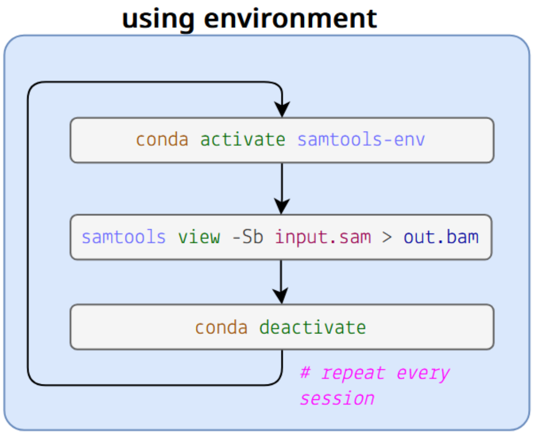
```
conda create -n rnaseq python=3.11 # create new env with specific python v
conda activate rnaseq # activate env
conda deactivate # deactivate env
conda create --clone rnaseq -n rnaseq-env # clone an env
conda env remove -n rnaseq  # remove an env
```

Inspecting environments

```
conda env export --from-history > env.yml # save requested packages
conda env create -f env.yml # recreate env
conda list --explicit > spec.txt # lock exact versions
conda create --file spec.txt # recreate exact env
```



```
conda create -n samtools-env
conda activate samtools-env
conda install bioconda::samtools
# add additional packages if
necessary in the same environment
```



```
conda activate samtools-env
samtools view -Sb input.sam > out.bam
conda deactivate
# repeat every
session
```