Array execution

The execution of arrays will require a previous preparation on the part of the user.

The user must structure the input data for the executions in folders or file names identified by a job number, so that it can be easily manipulated within the launch script.

Once you have the input data organized in this way, you can proceed to the execution of the jobs, launching as many tasks as needed in a submit script.

Execution of an array of 10 jobs. One node per job

```
# !/bin/bash
#SBATCH -J <job_name>
#SBATCH -p <partition>
#SBATCH -N 1
#SBATCH -N <nodes>
#SBATCH --tasks=<number>
#SBATCH --cpus-per-task=<number>
#SBATCH --constrains=<node arquitecture>  # sandy, ilk (icelake)... arquitecture
#SBATCH -t <days-HH:MM:SS>
#SBATCH --array=0-10
## %A JobId de la tarea principal (script de lanzamiento del arrayjob)
## %a Índice de la tarea dentro del arrayjob
## %j JobId de la tarea dentro del arrayjob en el gestor de colas
#SBATCH -o <array-%A-%a-.out>
#SBATCH -D .
#SBATCH --mail-type END
#SBATCH --mail-user <mail to notify>
###########################################################

module load <modules>
echo "Array job $SLURM_ARRAY_JOB_ID -> $SLURM_ARRAY_TASK_ID"
srun  program
echo "End array job $SLURM_ARRAY_JOB_ID -> $SLURM_ARRAY_TASK_ID"
```

Running an array of 100 jobs with only 10 concurrent jobs:

```
# !/bin/bash

#SBATCH -J <job_name>
#SBATCH -p <partition>
#SBATCH -N 1
#SBATCH -t <days-HH:MM:SS>
```

```
#SBATCH –array=0-100%10
## %A JobId de la tarea principal (script de lanzamiento del arrayjob)
## %a Índice de la tarea dentro del arrayjob
## %j JobId de la tarea dentro del arrayjob en el gestor de colas
#SBATCH -o <array-%A-%a-.out>
#SBATCH -D .
#SBATCH --mail-type END
#SBATCH --mail-user <mail to notify>
##################################################

module load <modules>
echo "Array job $SLURM_ARRAY_JOB_ID -> $SLURM_ARRAY_TASK_ID"
srun  program
echo "End array job $SLURM_ARRAY_JOB_ID -> $SLURM_ARRAY_TASK_ID"
```