

Job Arrays

A job array runs the same script multiple times in parallel, each with a different index. Use it to process many inputs with identical resource requirements — for example, running the same analysis on 100 different samples.

Setting up an array

Enter the array specification in the **Array** field on the submission form. Examples:

Value	What it does
<code>1-100</code>	100 tasks, indexed 1 to 100.
<code>0-9</code>	10 tasks, indexed 0 to 9.
<code>1-100%10</code>	100 tasks, at most 10 running simultaneously.
<code>1,3,7,42</code>	4 tasks with specific indices.

Using the index in your script

Slurm sets the environment variable `$SLURM_ARRAY_TASK_ID` to the current task's index. Use it to select the right input for each task:

```
module load python/3.11

SAMPLES=(sample_01 sample_02 sample_03 ...)
SAMPLE=${SAMPLES[$SLURM_ARRAY_TASK_ID]}

python analyse.py --input data/${SAMPLE}.csv --output results/${SAMPLE}/
```

Or if your inputs are numbered files:

```
python analyse.py --input data/sample_${SLURM_ARRAY_TASK_ID}.csv
```

Output files

Use `%a` in the output file path to get a separate file per task. For example, setting the output field to `logs/job_%j_%a.out` produces one log file per task.

Monitoring array jobs

Array jobs appear in the Jobs page as a single entry. Each individual task is listed with its own task ID and state. A parent job shows as running until all tasks complete.

Limiting concurrency

Use the `%N` suffix (e.g. `1-500%20`) to cap how many tasks run at once. This prevents a large array from monopolising the partition and avoids hitting per-user job limits.

Created 2026-06-25 08:32:20 UTC by levk

Updated 2026-07-08 15:38:28 UTC by levk