


NBI-Slurm: Simplified submission of Slurm jobs with energy saving mode

Andrea Telatin ^{1,2}

¹ Quadram Institute Bioscience, Norwich, NR4 7UQ, UK  ² Center for Microbial Interactions, Norwich Research Park, Norwich, NR4 7UG, UK

Date: 18 March 2026

Summary

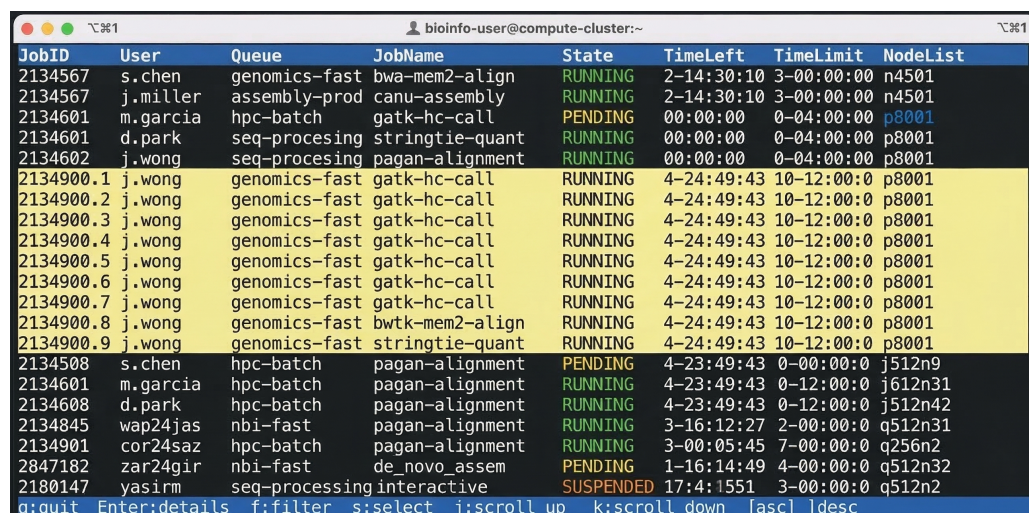
NBI-Slurm is a Perl package that provides a simplified, user-friendly interface for submitting and managing jobs on SLURM (Yoo et al., 2003) high-performance computing (HPC) clusters. It offers both a library of Perl modules for programmatic job management and a suite of command-line tools designed to reduce the cognitive overhead of SLURM's native interface. Two distinctive features of NBI-Slurm are TUI applications to view and cancel jobs, and an energy-aware scheduling mode — “eco mode” — that automatically defers flexible jobs to off-peak periods, helping research institutions reduce their computational carbon footprint without requiring users to manually plan submission times.

Statement of Need

HPC clusters are indispensable in modern research, particularly in the life sciences where large-scale sequence analyses, genome assemblies, and statistical models demand resources beyond a desktop workstation. SLURM has become the dominant workload manager in this space (Wang et al., 2020), yet its interface presents a steep learning curve. Users must learn a verbose sbatch scripting syntax, understand resource unit conventions (memory in megabytes, time in D-HH:MM:SS format), manage job dependencies manually, and repeat boilerplate directives across every submission script.

Workflow managers such as Snakemake (Mölder et al., 2021) and Nextflow (Di Tommaso et al., 2017) address this at the pipeline level by abstracting SLURM as an execution backend, but they require users to rewrite their analysis logic inside a domain-specific language. Many researchers have existing shell scripts or one-off analyses that do not warrant a full pipeline refactor. NBI-Slurm occupies a complementary niche: it wraps SLURM's interface without imposing a workflow model, making it straightforward to submit individual commands or small batches while retaining access to all SLURM features through pass-through options.

The `lsjobs` utility prints a colour-coded, human-readable table of queued jobs as a static snapshot, offering a more ergonomic alternative to the raw output of `squeue`. Its companion tool `viewjobs` provides a fully interactive terminal user interface (TUI) that allows users to browse the live job queue without leaving the terminal (Figure 1). Users can scroll through jobs with arrow or Vim keys, sort columns, inspect per-job details, toggle column visibility, and adjust column widths interactively. Individual jobs can be selected with Space and multiple selected jobs can be cancelled in bulk with a single keypress, removing the need to copy-paste job IDs into `scancel`.



| JobID | User | Queue | JobName | State | TimeLeft | Timelimit | NodeList |
|-----------|----------|----------------|-----------------|-----------|------------|-------------|----------|
| 2134567 | s.chen | genomics-fast | bwa-mem2-align | RUNNING | 2-14:30:10 | 3-00:00:00 | n4501 |
| 2134567 | j.miller | assembly-prod | canu-assembly | RUNNING | 2-14:30:10 | 3-00:00:00 | n4501 |
| 2134601 | m.garcia | hpc-batch | gatk-hc-call | PENDING | 00:00:00 | 0-04:00:00 | p8001 |
| 2134601 | d.park | seq-processing | stringtie-quant | RUNNING | 00:00:00 | 0-04:00:00 | p8001 |
| 2134602 | j.wong | seq-processing | pagan-alignment | RUNNING | 00:00:00 | 0-04:00:00 | p8001 |
| 2134900.1 | j.wong | genomics-fast | gatk-hc-call | RUNNING | 4-24:49:43 | 10-12:00:00 | p8001 |
| 2134900.2 | j.wong | genomics-fast | gatk-hc-call | RUNNING | 4-24:49:43 | 10-12:00:00 | p8001 |
| 2134900.3 | j.wong | genomics-fast | gatk-hc-call | RUNNING | 4-24:49:43 | 10-12:00:00 | p8001 |
| 2134900.4 | j.wong | genomics-fast | gatk-hc-call | RUNNING | 4-24:49:43 | 10-12:00:00 | p8001 |
| 2134900.5 | j.wong | genomics-fast | gatk-hc-call | RUNNING | 4-24:49:43 | 10-12:00:00 | p8001 |
| 2134900.6 | j.wong | genomics-fast | gatk-hc-call | RUNNING | 4-24:49:43 | 10-12:00:00 | p8001 |
| 2134900.7 | j.wong | genomics-fast | gatk-hc-call | RUNNING | 4-24:49:43 | 10-12:00:00 | p8001 |
| 2134900.8 | j.wong | genomics-fast | bwtk-mem2-align | RUNNING | 4-24:49:43 | 10-12:00:00 | p8001 |
| 2134900.9 | j.wong | genomics-fast | stringtie-quant | RUNNING | 4-24:49:43 | 10-12:00:00 | p8001 |
| 2134508 | s.chen | hpc-batch | pagan-alignment | PENDING | 4-23:49:43 | 0-00:00:00 | j512n9 |
| 2134601 | m.garcia | hpc-batch | pagan-alignment | RUNNING | 4-23:49:43 | 0-12:00:00 | j612n31 |
| 2134608 | d.park | hpc-batch | pagan-alignment | RUNNING | 4-23:49:43 | 0-12:00:00 | j512n42 |
| 2134845 | wap24jas | nbi-fast | pagan-alignment | RUNNING | 3-16:12:27 | 2-00:00:00 | q512n31 |
| 2134901 | cor24saz | hpc-batch | pagan-alignment | RUNNING | 3-00:05:45 | 7-00:00:00 | q256n2 |
| 2847182 | zar24gir | nbi-fast | de_novo_assem | PENDING | 1-16:14:49 | 4-00:00:00 | q512n32 |
| 2180147 | yasirm | seq-processing | interactive | SUSPENDED | 17:4:1551 | 3-00:00:00 | q512n2 |

q:quit Enter:details f:filter s:select j:scroll up k:scroll down [asc] |desc

Figure 1: Interactive TUI of viewjobs, showing job navigation, multi-column display, and bulk-cancel workflow. The image is AI generated from a real screenshot (Google NanoBanana)

Energy consumption in research computing is a growing concern (Lannelongue et al., 2021). Most researchers have no practical mechanism to shift flexible jobs to periods when grid electricity is cheaper or cleaner. NBI-Slurm addresses this directly with a configurable scheduling module that calculates the next available low-energy window and injects a `--begin` directive into the submission, requiring no change to the underlying command.

Availability and Installation

NBI-Slurm is distributed under the MIT licence and is available from CPAN as `NBI::Slurm`. Installation requires Perl 5.16 or later and can be performed with:

```
cpam NBI::Slurm
```

The source code is hosted at <https://github.com/quadram-institute-bioscience/NBI-Slurm> under continuous integration. Development has been active since June 2023, and the module is published to the MetaCPAN repository at <https://metacpan.org/dist/NBI-Slurm>.

Code Structure and Dependencies

The package is organised into two layers.

Perl module library (`lib/NBI/`): Five classes model the key abstractions:

- `NBI::Opts` — encapsulates SLURM resource directives (queue, threads, memory, wall-time, email, job arrays, start time). It accepts human-friendly inputs such as "8GB" or "2h30m" and converts them to SLURM's expected formats.
- `NBI::Job` — represents a job to be submitted, holding a command (or list of commands) and an `NBI::Opts` object. The `script()` method generates a complete sbatch script; `run()` submits it and returns the job identifier.
- `NBI::Queue` — queries the live SLURM queue via `squeue` and returns a list of `NBI::QueuedJob` objects, optionally filtered by user, status, name, or queue.
- `NBI::QueuedJob` — a lightweight data object representing one queued job, used by the queue-management tools.
- `NBI::EcoScheduler` — implements the energy-aware scheduling logic. Given a job's expected duration and a set of configurable windows, it finds the next period satisfying

a three-tier preference: (1) job completes within an eco window and avoids peak hours; (2) job starts in an eco window and avoids peak hours but may overrun; (3) job starts in an eco window and partially overlaps peak hours. Default windows target weekday nights (00:00–06:00) and weekend off-peak periods (00:00–07:00, 11:00–16:00), avoiding evening peaks (17:00–20:00), all of which are fully configurable with a settings file (by default `~/nbislrurm.config`).

Command-line tools (`bin/`):

| Tool | Purpose |
|-----------------------|---|
| <code>runjob</code> | Submit a command as a SLURM job with resource flags |
| <code>lsjobs</code> | List, filter, and cancel user jobs with coloured tabular output |
| <code>viewjobs</code> | Interactive terminal UI for job management |
| <code>waitjobs</code> | Block until jobs matching a pattern complete |
| <code>whojobs</code> | Show cluster utilisation grouped by user |
| <code>session</code> | Launch an interactive SLURM session |

Runtime dependencies are deliberately minimal: `Capture::Tiny` (≥ 0.40), `JSON::PP`, `Text::ASCIITable` (≥ 0.22), `Term::ANSIColor`, `Storable`, and `POSIX`—all either part of the Perl core or widely available on CPAN.

Documentation

Each module is documented with embedded POD (Plain Old Documentation), rendered on CPAN at <https://metacpan.org/dist/NBI-Slurm>. Each command-line tool provides a `--help` flag and a manual page generated from its POD. A user guide with annotated examples is maintained in the repository's `README.md`. The test suite (`t/`) covers unit behaviour of every module and integration behaviour of the command-line tools; author-facing tests (`xt/`) verify POD completeness and coverage. All tests will be able to check functions even without Slurm. To check the ability to interact with Slurm, there are optional tests that can be executed with `prove -lv xt/hpc-*.t`.

Example Applications

Submitting a parallel job. A researcher wishing to run a genome assembler with 18 cores, 64 GB RAM, and a 12-hour wall-time can write:

```
runjob -n "assembly" -c 18 -m 64 -t 12 -w ./logs/ \
  "flye --nano-raw reads.fastq --out-dir asm"
```

Processing a file list as a job array. To align 200 FASTQ files, one job per file:

```
runjob -n "align" -c 8 -m 16 --files samples.txt \
  "bwa mem ref.fa #FILE# > #FILE#.bam"
```

Energy-aware deferral. A long-running but flexible annotation job can be scheduled for the next eco window automatically. Note that by default the *eco mode* is enabled, and can be overridden with `--no-eco` or setting the `economy_mode=0` in the configuration file.

```
runjob --eco -n "annotate" -t 6 "prokka genome.fa"
```

NBI-Slurm calculates the next suitable window (e.g., the following night) and adds `--begin=2026-03-19T00:00:00` to the submission without any further user action.

Programmatic job chaining. In a Perl analysis script:

```
use NBI::Job;
use NBI::Opts;

my $opts = NBI::Opts->new(
    -queue => "long",
    -threads => 16,
    -memory => 32,
    -time => "4h");
my $job = NBI::Job->new(
    -name => "step1",
    -command => "python align.py",
    -opts => $opts);
my $id = $job->run();

my $opts2 = NBI::Opts->new(
    -queue => "short",
    -threads => 4,
    -memory => 8,
    -time => "1h");
my $job2 = NBI::Job->new(
    -name => "step2",
    -command => "python report.py --input results/",
    -opts => $opts2);
$job2->opts->dependencies([$id]);
$job2->run();
```

Acknowledgements

The author gratefully acknowledges the support of the Biotechnology and Biological Sciences Research Council (BBSRC); this research was funded by the BBSRC Institute Strategic Programme Food Microbiome and Health BB/X011054/1 and its constituent project(s) BBS/E/QU/230001B; the BBSRC Institute Strategic Programme Microbes and Food Safety BB/X011011/1 and its constituent project(s) BBS/E/QU/230002C; the BBSRC Core Capability Grant BB/CCG2260/1. This research was also supported by the infrastructure provided by the CLIMB-BIG-DATA grant MR/T030062/1. The author thanks colleagues at the Quadram Institute Bioscience for feedback and field-testing during development, and the GreenDISC working group and NBI Research Computing for support and discussions.

AI Usage Disclosure

Claude Code (Anthropic) was used during development of NBI-Slurm from version 0.10.0 onwards, assisting with code generation, refactoring, test scaffolding, and documentation drafting. It was also used to assist with drafting and editing this paper. All AI-assisted outputs were reviewed, edited, and validated by the author, who made all core design decisions and retains full responsibility for the accuracy, originality, and correctness of the submitted materials.

References

Di Tommaso, P., Chatzou, M., Floden, E. W., Barja, P. P., Palumbo, E., & Notredame, C. (2017). Nextflow enables reproducible computational workflows. *Nature Biotechnology*,

- 35(4), 316–319. <https://doi.org/10.1038/nbt.3820>
- Lannelongue, L., Grealey, J., & Inouye, M. (2021). Green algorithms: Quantifying the carbon footprint of computation. *Advanced Science*, 8(12). <https://doi.org/10.1002/adv.202100707>
- Mölder, F., Jablonski, K. P., Letcher, B., Hall, M. B., Tomkins-Tinch, C. H., Sochat, V., Forster, J., Lee, S., Twardziok, S. O., Kanitz, A., Wilm, A., Holtgrewe, M., Rahmann, S., Nahsen, S., & Köster, J. (2021). Sustainable data analysis with snakemake. *F1000Research*, 10, 33. <https://doi.org/10.12688/f1000research.29032.2>
- Wang, B., Chen, Z., & Xiao, N. (2020). A survey of system scheduling for HPC and big data. *Proceedings of the 2020 4th International Conference on High Performance Compilation, Computing and Communications*, 178–183. <https://doi.org/10.1145/3407947.3407977>
- Yoo, A. B., Jette, M. A., & Grondona, M. (2003). SLURM: Simple linux utility for resource management. *Job Scheduling Strategies for Parallel Processing*, 44–60. https://doi.org/10.1007/10968987_3