

# BEAGLE 3 FOR BEAST 2.7 ON EULER

cEvo group meeting, 2024-12-17

# Beagle 3 for BEAST2.7

**BEAGLE** (Bayesian Evolutionary Analysis by Sampling Trees Library) is a high-performance computational library designed to accelerate phylogenetic likelihood calculations.

*NB: Only available for genetic sequence data*

All benchmarking was done using:

- Openjdk 17.0.8.1\_1
- Beagle 3.1.2
- BEAST 2.7.4 with ReMaster, Feast and BeastLabs packages
- NVIDIA GeForce RTX 2080 GPUs and AMD EPYC CPUs

Set-up largely inspired by Remco Bouckaert analyses and CIPRES study  
(<https://www.beast2.org/2021/03/31/command-line-options.html>, bottom)

$3 \times 4 \times 4 \times (5 \times 3 + 3 \times 1) \times 4 \times 2 = \underline{1728}$  configurations tested

Taxa	Partitions	Total seq. length	CPUs	GPUs	Beagle option	Instances
20	1	500	1	1	java	true
250	3	1000	2	2	beagle_CPU	false
1000	6	3000	4	4	beagle_SSE	
	9	9000	6		beagle_GPU	
			8			

- If instances true, it is equal to number of CPU cores or assigned GPUs
- Threads are always set to the number or CPU cores or GPUs
- Whenever more than 1 GPUs are used, -beagle\_order option is specified

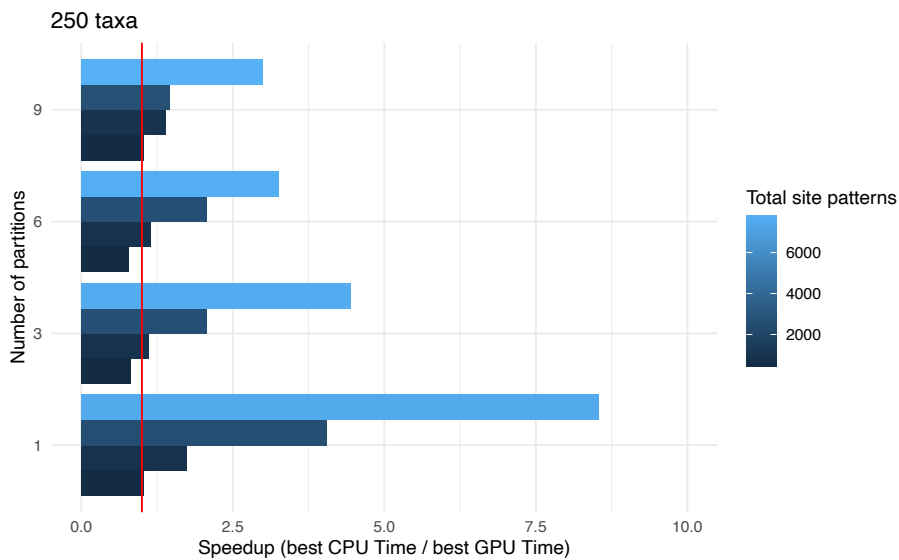
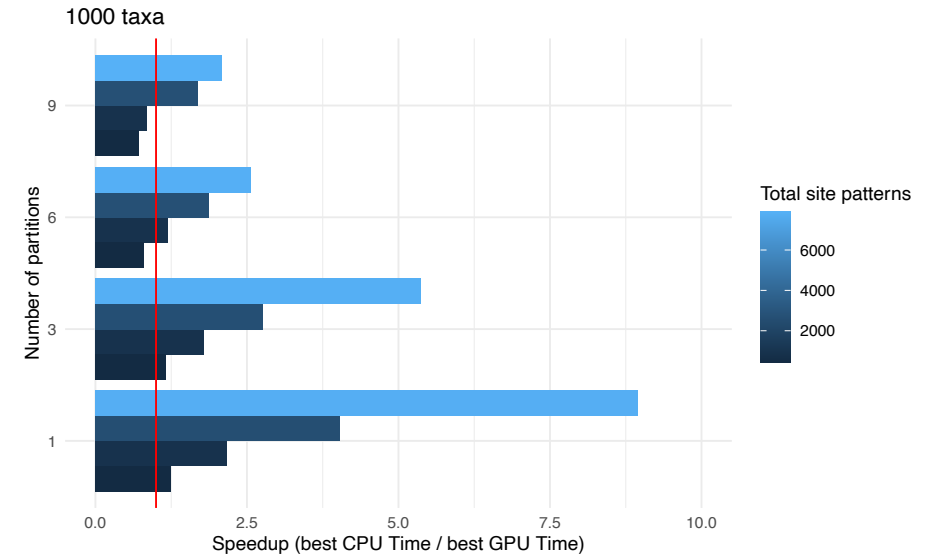
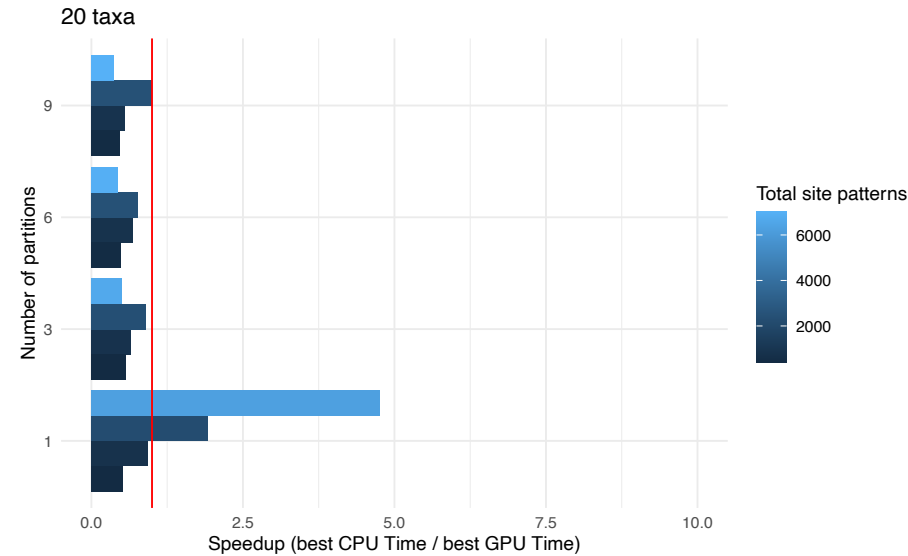
$3 \times 4 \times 4 \times (5 \times 3 + 3 \times 1) \times 4 \times 2 = \underline{1728}$  configurations tested

Taxa	Partitions	Total seq. length	CPUs	GPUs	Beagle option	Instances
20	1	500	1	1	java	true
250	3	1000	2	2	beagle_CPU	false
1000	6	3000	4	4	beagle_SSE	
	9	9000	6		beagle_GPU	
			8			

For every data setup (taxa-partitions-seq. length):

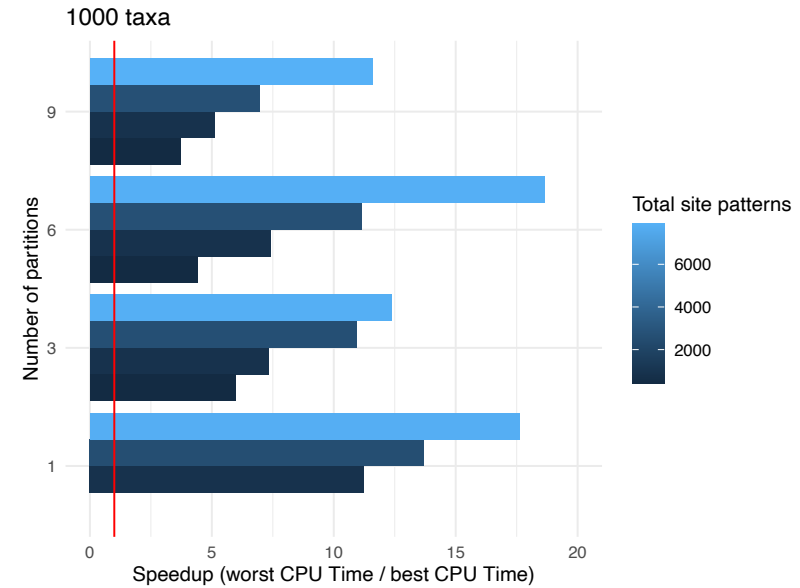
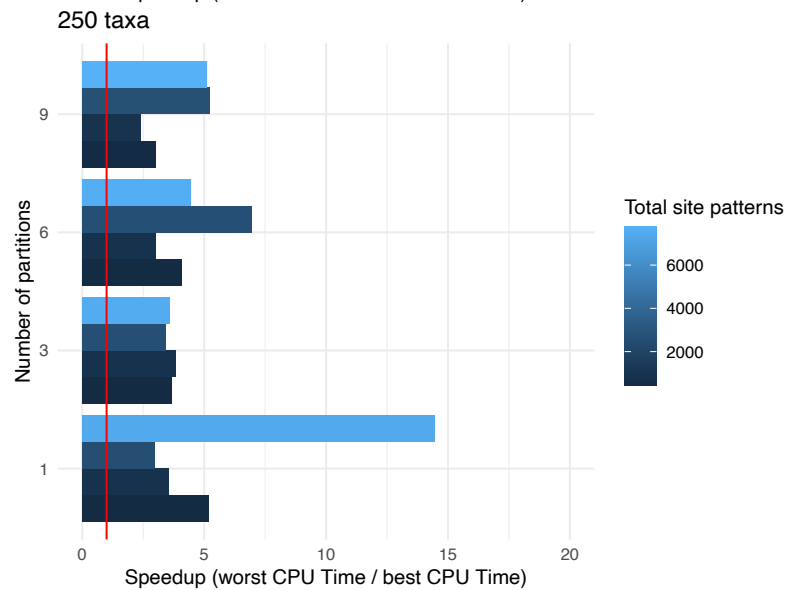
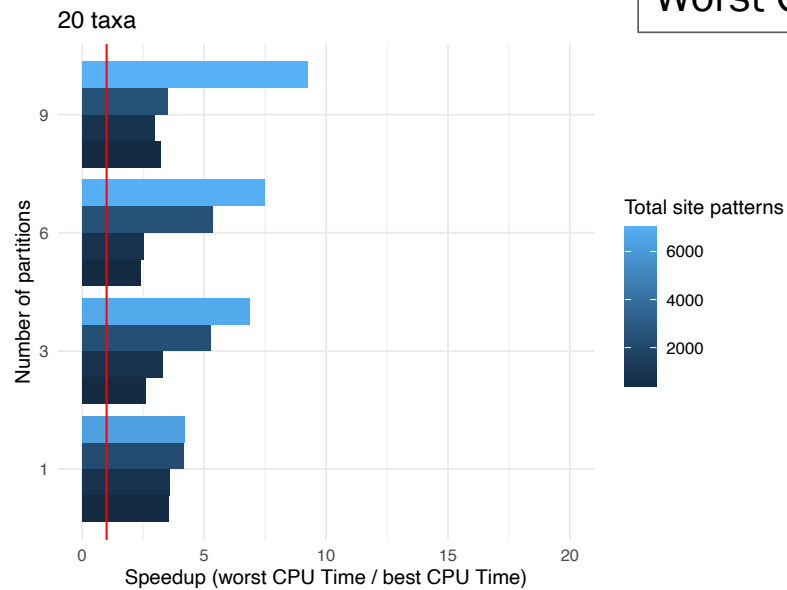
- Simulated coalescent trees and sequences at the tips
- Effective pop. size = 2, all taxa sampled at present
- HKY + Gamma, subst. rate = 1
- Inference of tree, coalescent and substitution model parameters using broad default priors.
- In case of multiple partitions, MultiPartitionTreeLikelihood (BeastLABS, pre-release) was used.

## Best CPU vs best GPU time

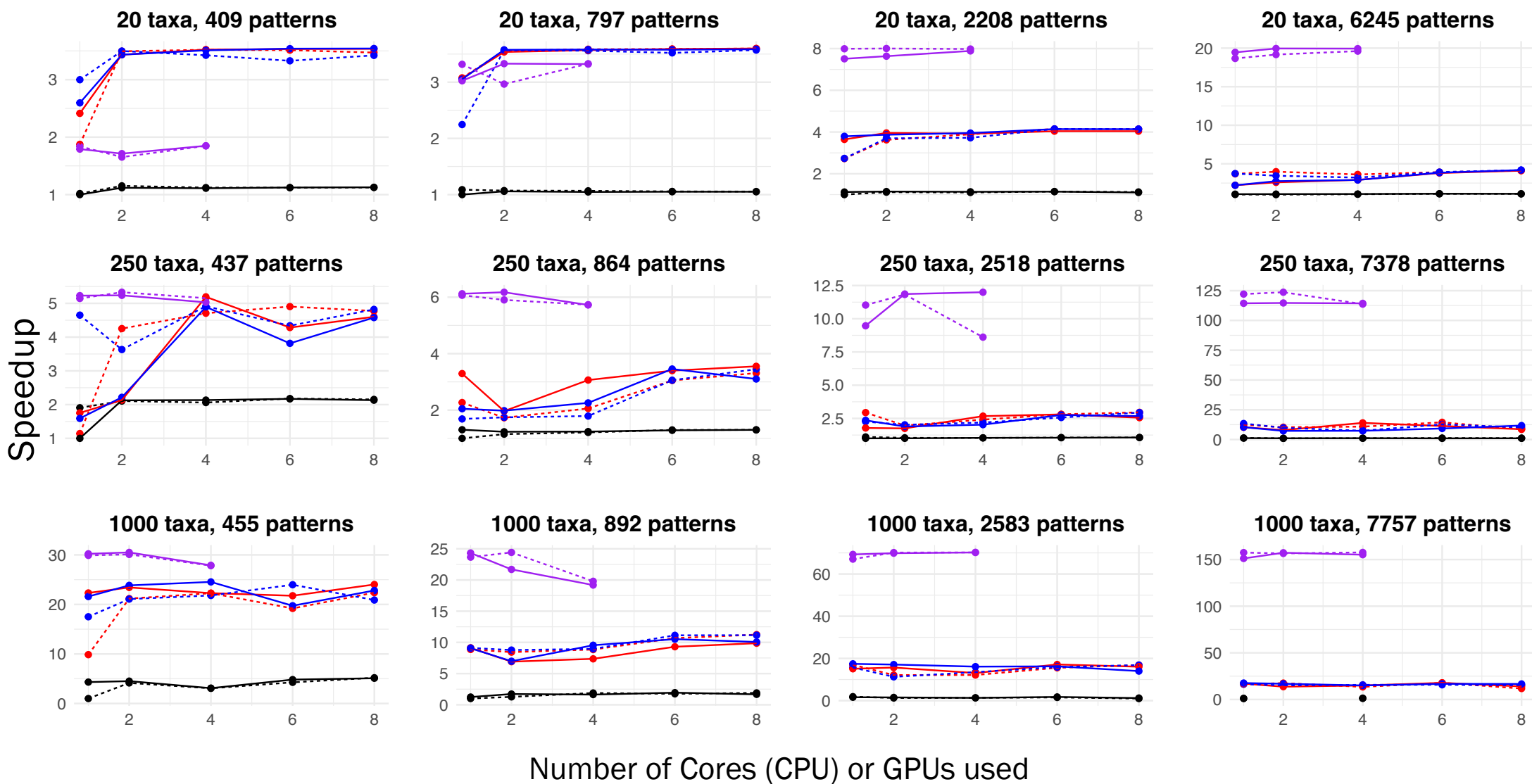


- Site patterns per partition affect speedup the most.
- Small trees would only benefit from GPU, given large (>5000) number of site patterns per partition
- For medium and large trees speedup is seen at around 500 to 1000 total site patterns and is significant at >2000 site patterns.
- This effect diminished with increasing number of partitions (if total number of site patterns does not change).

## Worst CPU vs best CPU time



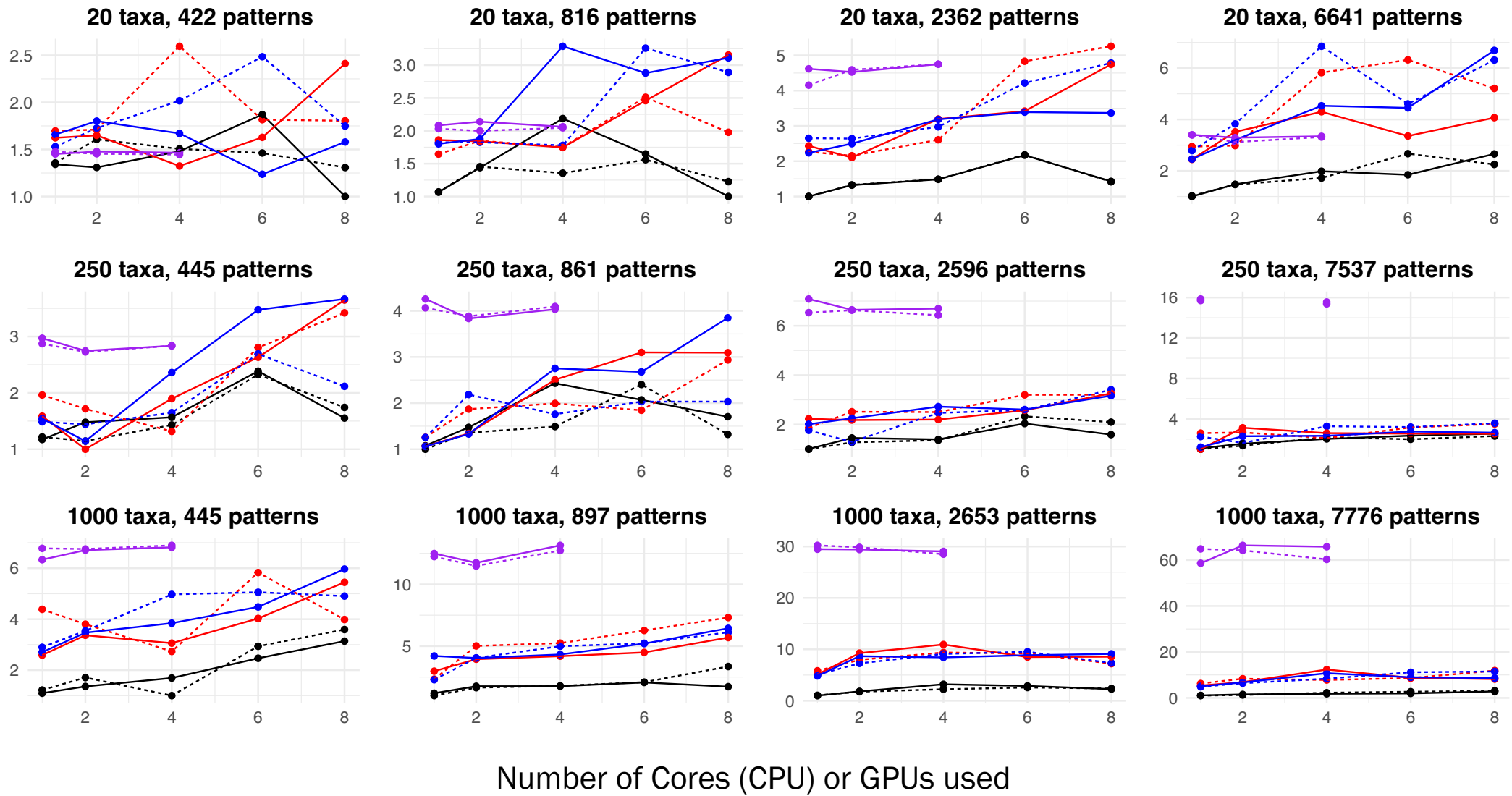
Even without GPU availability, optimising for best CPU running configuration is worthwhile in ALL cases!



1 partition

Instances — false ..... true Beagle Variant — java — beagle\_CPU — beagle\_SSE — beagle\_GPU

Speedup

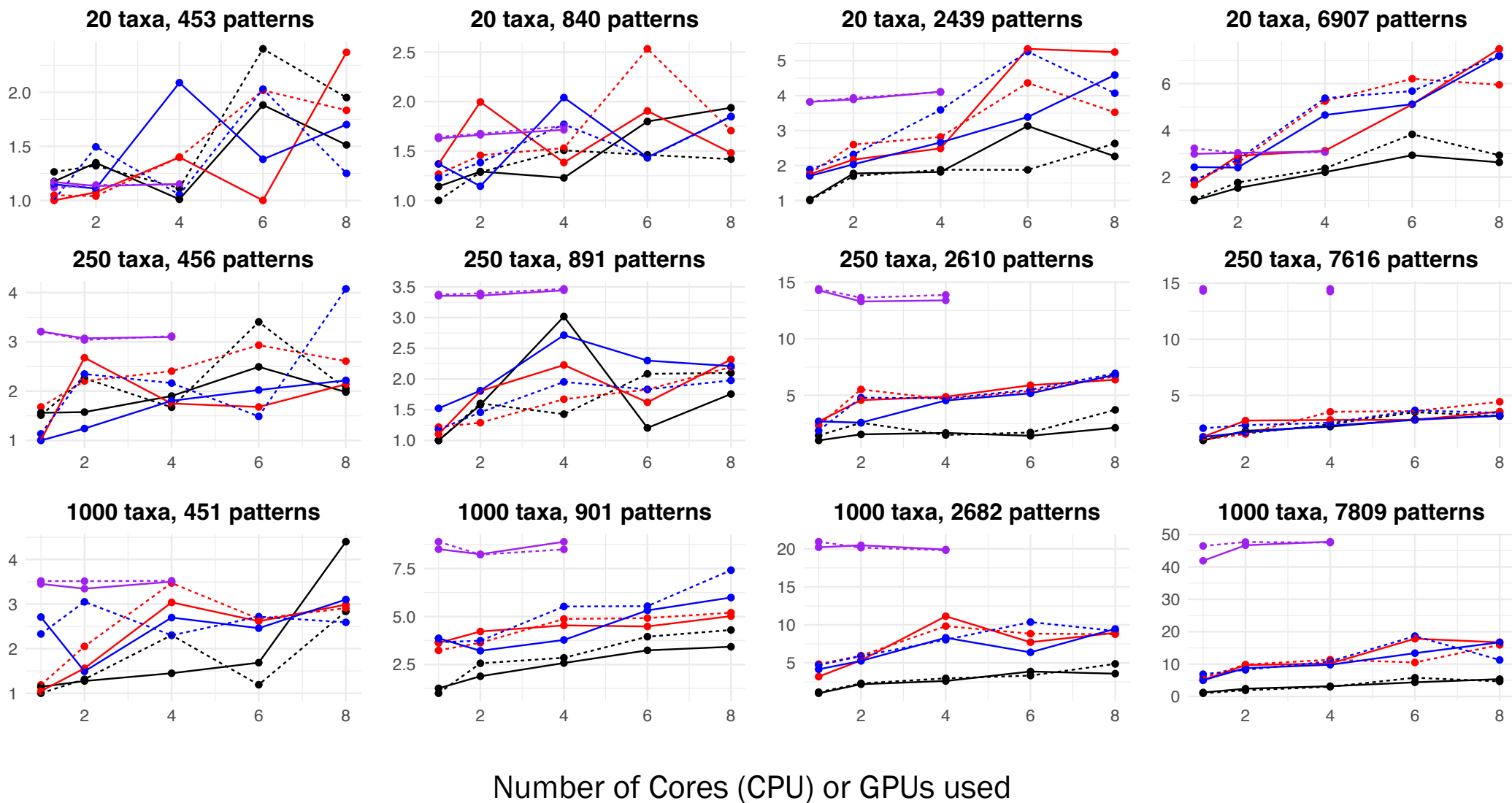


3 partitions

Instances — false .... true Beagle Variant — java — beagle\_CPU — beagle\_SSE — beagle\_GPU



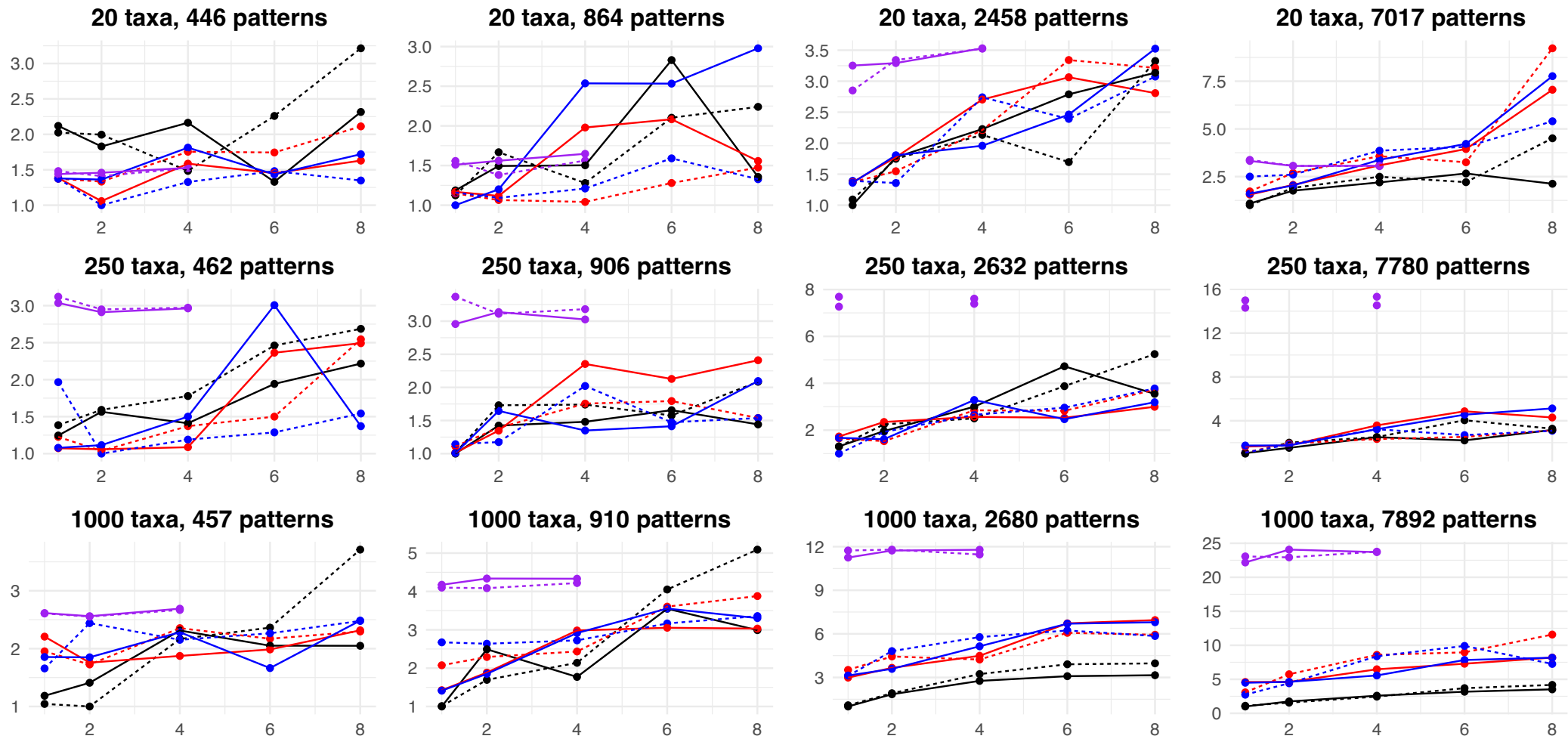
Speedup



6 partitions

Instances — false .... true Beagle Variant — java — beagle\_CPU — beagle\_SSE — beagle\_GPU

Speedup



Number of Cores (CPU) or GPUs used

9 partitions

Instances — false ···· true Beagle Variant —●— java —●— beagle\_CPU —●— beagle\_SSE —●— beagle\_GPU

# Real data

Thanks Noémie!

- Bacteria causing Pertussis
- 1248 taxa
- 1 partition
- 4701 sites
- 3388 patterns (4081486 constant sites)

# Real data

Thanks Noémie!

- Bacteria causing Pertussis
- 1248 taxa
- 1 partition
- 4701 sites
- 3388 patterns (4081486 constant sites)

	CPUs / GPUs	10 <sup>6</sup> iterations	Speedup	10 <sup>8</sup> iteration
java	1 / 0	14 h 56 min	---	62 days 5 h 39 min
beagle_SSE	1 / 0	36 min	24.7	2 days 12 h 30 min
beagle_SSE	6 / 0	18-19 min	47.7	1 days 7 h 17 min
beagle_GPU	1 / 1	17-18 min	50.6	1 day 5 h 30 min

# Benchmarking resources for your data

- Automatic Beagle 3 benchmarking: available in BEAST1 and being implemented in BEAST2.

# Benchmarking resources for your data

- Automatic Beagle 3 benchmarking: available in BEAST1 and being implemented in BEAST2.

	CPU / GPU	10 <sup>6</sup> iterations	Speedup	10 <sup>8</sup> iteration
java	1 / 0	14 h 56 min	---	62 days 5 h 39 min
beagle_SSE	1 / 0	36 min	24.7	2 days 12 h 30 min
beagle_SSE	6 / 0	18-19 min	47.7	1 days 7 h 17 min
beagle_GPU	1 / 1	17-18 min	50.6	1 day 5 h 30 min

# Benchmarking resources for your data

- Automatic Beagle 3 benchmarking: available in BEAST1 and being implemented in BEAST2.

	CPU / GPU	10 <sup>6</sup> iterations	Speedup	10 <sup>8</sup> iteration
java	1 / 0	14 h 56 min	---	62 days 5 h 39 min
beagle_SSE	1 / 0	36 min	24.7	2 days 12 h 30 min
beagle_SSE	6 / 0	18-19 min	47.7	1 days 7 h 17 min
beagle_SSE	8 / 0	30 min	30	2 days 2 h 22 min
beagle_GPU	1 / 1	17-18 min	50.6	1 day 5 h 30 min

# Benchmarking resources for your data

- Automatic Beagle 3 benchmarking: available in BEAST1 and being implemented in BEAST2.
- Benchmarking for yourself: [https://github.com/jugne/beast-beagle-benchmark/tree/main/self\\_benchmarking\\_scripts](https://github.com/jugne/beast-beagle-benchmark/tree/main/self_benchmarking_scripts)

```
template_beast_task.slurm x
1  ▶ #!/bin/bash
2  #SBATCH --job-name=my_job
3  #SBATCH --ntasks=1
4  #SBATCH --time=01:00:00
5  #SBATCH --mem-per-cpu=16
6  #SBATCH --cpus-per-task=%CPUS%
7  #SBATCH --output=job_%CPUS%.out
8
9
10 # DO NOT change the last two lines above. You may always add more options.
11 # Your job commands should be below and contain
12 # "-threads -1" OR "-threads %CPUS%" option to BEAST
13 # This script is only intended for jobs NOT using GPUs
```



# Benchmarking resources for your data

- Automatic Beagle 3 benchmarking: available in BEAST1 and being implemented in BEAST2.
- Benchmarking for yourself: [https://github.com/jugne/beast-beagle-benchmark/tree/main/self\\_benchmarking\\_scripts](https://github.com/jugne/beast-beagle-benchmark/tree/main/self_benchmarking_scripts)

```
jugne@eu-login-32:/cluster/scratch/jugne/self_benchmarking_scripts$ ./submit_jobs.sh --minCPU 1 --maxCPU 4 --increment 1
Submitting job 1 with 1 CPUs...
Submitted batch job 18480684
Submitting job 2 with 2 CPUs...
Submitted batch job 18480685
Submitting job 3 with 3 CPUs...
Submitted batch job 18480686
Submitting job 4 with 4 CPUs...
Submitted batch job 18480687
All jobs submitted.
```

# Benchmarking resources for your data

- Automatic Beagle 3 benchmarking: available in BEAST1 and being implemented in BEAST2.
- Benchmarking for yourself: [https://github.com/jugne/beast-beagle-benchmark/tree/main/self\\_benchmarking\\_scripts](https://github.com/jugne/beast-beagle-benchmark/tree/main/self_benchmarking_scripts)

```
[jugne@eu-login-32:/cluster/scratch/jugne/self_benchmarking_scripts$ ./print_results.sh  
File: ./job_1.out, CPUs: 1, Time: .32 minutes  
File: ./job_2.out, CPUs: 2, Time: .39 minutes  
File: ./job_3.out, CPUs: 3, Time: .26 minutes  
File: ./job_4.out, CPUs: 4, Time: .28 minutes
```

# Remarks

- Cipres benchmarking study specify data by number of site patterns and partitions, not number of taxa. I have seen significant influence on speedup depending also on number of taxa.
- It is always beneficial to use Beagle.
- In majority of the cases adequate number of CPU cores is enough and GPU is not needed.
- It is still hard to come up with rules of thumb for exact number of cores needed.
- In majority of the cases increasing the number of GPUs did not improve runtime significantly.