Libpll sequential benchmarks

November 29, 2016

1 Benchmark description

The following benchmarks compare several libpll implementations with different modes. They measure the execution time of a full likelihood computation on a fixed tree. To avoid measuring the initialization part, we repeat several times `pll_update_partials` and `pll_compute_edge_likelihood` on the same partitions and tree.

- xflouris means that the implentation used is this one: https://github.com/xflouris/libpll.
- bmorel means that the implementation used is this one: https://github.com/BenoitMorel/libpll. It supports sites repeats, and the data structure used is a bit different from xflouris (even without sites repeats): CLVs are not assumed to be sorted by sites, and an additional lookup table is used to access them in most of the core functions.
- bmorel2 is a (temporary) hacked version of bmorel where I do not use the new lookup structure for the scalers, because I think it slows down the execution, and the speed up with the sites repeats is not great.
- default mode means that the option `PLL_ATTRIB_PATTERN_TIP` and `PLL_ATTRIB_SITES_REPEATS` are unset.
- tip pattern means that the option `PLL_ATTRIB_PATTERN_TIP` is set.
- sites repeats means that the option `PLL_ATTRIB_SITES_REPEATS` is set.
- M is the size of the buffer allocated to compute the sites repeats class identifiers. When it increases, more nodes can benefit from sites repeats.

2 Summary of the results

Without sites repeats, bmorel's implementation with the additional lookup table can be from 0% to 10% slower than xflouris' one, especially with SSE and AVX architectures. bmorel2's hacked implementation (so without using the lookup for the scalers) is better, but can also be slower than xflouris'one.

In all versions, the tip pattern mode is around 1.5 faster than the default mode.

The sites repeats speed up is between 2 and 5 times faster than the tip pattern mode. It depends on :

- the dataset
- the architecture : it performs a bit better with CPU than with AVX and SSE
- the size of the sites repeats matrix buffer (M) : it performs better when the size increases

3 Benchmark

CPU architecture, 500 iterations

	seq59	seq128	seq404
xflouris default mode	8620.94 ms	116432 ms	150826 ms
bmorel default mode	8998.48 ms	119832 ms	150664 ms
xflouris tip pattern	5686.23 ms	74702.6 ms	94993.4 ms
bmorel tip pattern	5897.14 ms	80567.7 ms	101541 ms
bmorel sites repeats M=1000	3954.25 ms	46762.8 ms	65467.1 ms
bmorel sites repeats M=10000	3362.97 ms	36206.5 ms	45640.1 ms
bmorel sites repeats M=1000000	2446.75 ms	26647.3 ms	24581.6 ms
bmorel sites repeats no update sr M=1000000	2050.9 ms	20612.4 ms	17142 ms

SSE architecture, 500 iterations

	seq59	seq128	seq404
xflouris default mode	3668.12 ms	51162.5 ms	59829.2 ms
bmorel default mode	3958.71 ms	54808.7 ms	65261.9 ms
xflouris tip pattern	2510.35 ms	35259.7 ms	41151.4 ms
bmorel tip pattern	2809.44 ms	38872.6 ms	44230.5 ms
bmorel sites repeats M=1000	2050.93 ms	23216.8 ms	30559.5 ms
bmorel sites repeats M=10000	1761.89 ms	20055.1 ms	23288.7 ms
bmorel sites repeats M=1000000	1600.45 ms	16375.4 ms	14797.8 ms
bmorel sites repeats no update sr M=1000000	913.486 ms	9619.04 ms	7461.7 ms

AVX architecture, 500 iterations

	seq59	seq128	seq404
xflouris default mode	3254.32 ms	48570 ms	56181.3 ms
bmorel default mode	3618.91 ms	53191.7 ms	64534.5 ms
xflouris tip pattern	2623.61 ms	37361.2 ms	41372.5 ms
bmorel tip pattern	2557.23 ms	38809.8 ms	41619.1 ms
bmorel sites repeats M=1000	3129.43 ms	28521.6 ms	29359.7 ms
bmorel sites repeats M=10000	1723.58 ms	19300.9 ms	22789.5 ms
bmorel sites repeats M=1000000	1444.81 ms	15162.5 ms	14183.8 ms
bmorel sites repeats no update sr M=1000000	819.969 ms	9186.13 ms	6602.27 ms