Libpll sequential benchmarks

December 16, 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
tip pattern	6409ms	83700ms	111602ms
repeats 100000	2768ms	32855ms	30542ms
repeats 1000000	2629ms	28380ms	24982ms
repeats 1000000 no update	2084ms	21993ms	16232ms
bmorel sites repeats (ti opt) 1000000	2522ms	27592ms	22289ms
bmorel sites repeats no update (ti opt) 1000000	2065ms	21670ms	16063ms

SSE architecture, 500 iterations

	seq59	seq128	seq404
tip pattern	3589ms	47830ms	57893ms
repeats 100000	1722ms	19882ms	19532ms
repeats 1000000	1659ms	18106ms	15991ms
repeats 1000000 no update	1130ms	11895ms	9033ms
bmorel sites repeats (ti opt) 1000000	1808ms	18425ms	16707ms
bmorel sites repeats no update (ti opt) 1000000	1309ms	12478ms	10035ms

AVX architecture, 500 iterations

	seq59	seq128	seq404
tip pattern	3321ms	46104ms	55210ms
repeats 100000	1721ms	19001ms	18879ms
repeats 1000000	1602ms	17391ms	15651ms
repeats 1000000 no update	1090ms	11162ms	8448ms
bmorel sites repeats (ti opt) 1000000	1778ms	17662ms	16295ms
bmorel sites repeats no update (ti opt) 1000000	1271ms	11891ms	9685ms