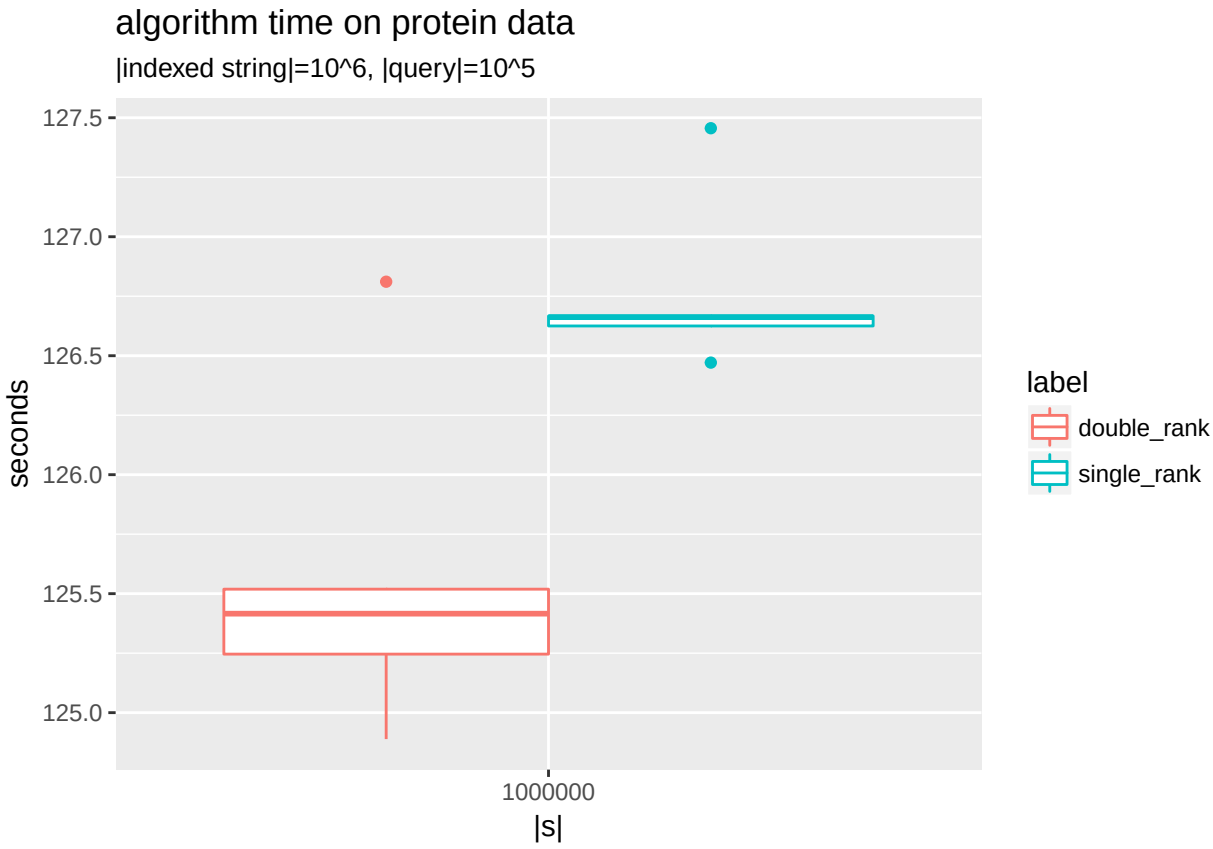
Lazy vs. non lazy

O. Denas

12/28/2016

Contents

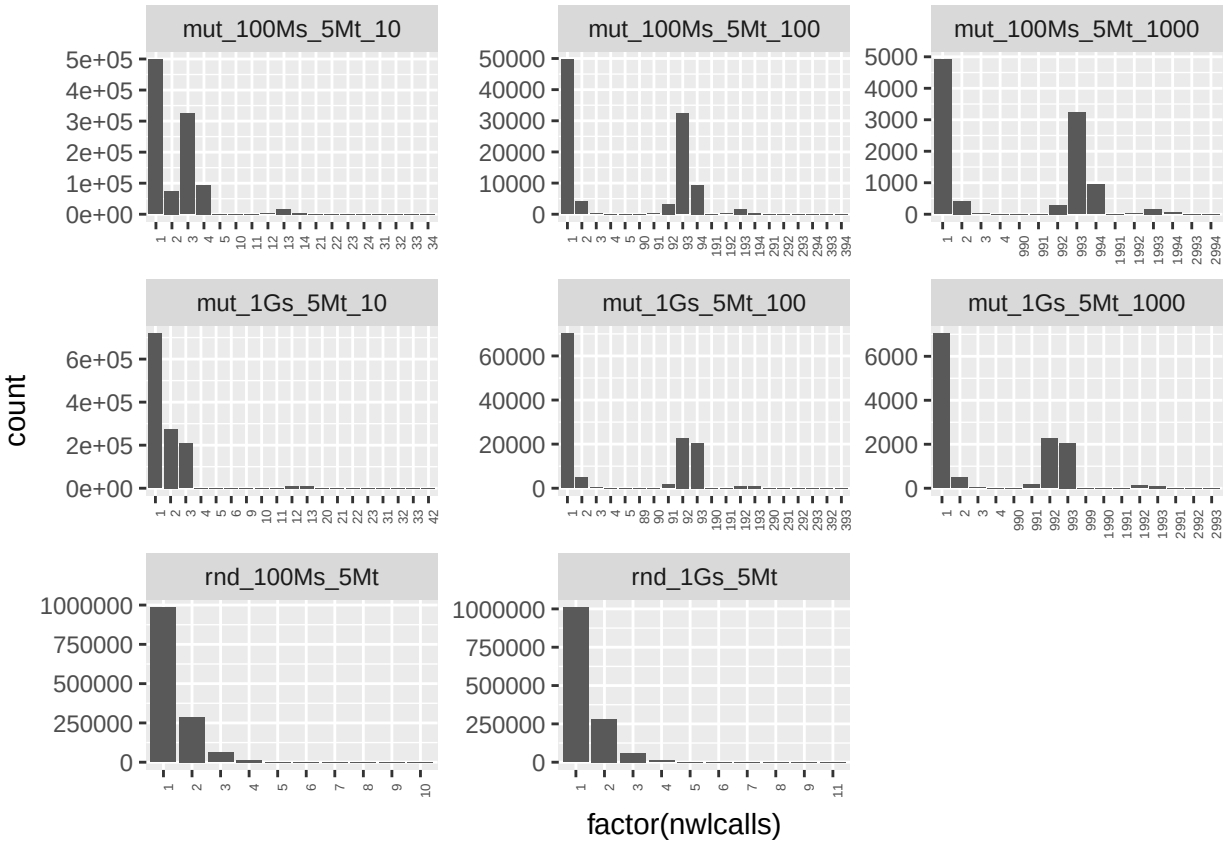
1 Double vs. single rank



2 Lazy vs non-lazy

2.1 Input properties

For various types ("mut_XMs_YMt_Z" means `s` and `t` are random identical strings of length X, and Y million respectively with mutations inserted every Z characters. "rnd_XMs_YMt" means `s` and `t` are random strings of length X, and Y million respectively) of inputs run the MS algorithm and count the number of consecutive `lazy_wl()` calls.

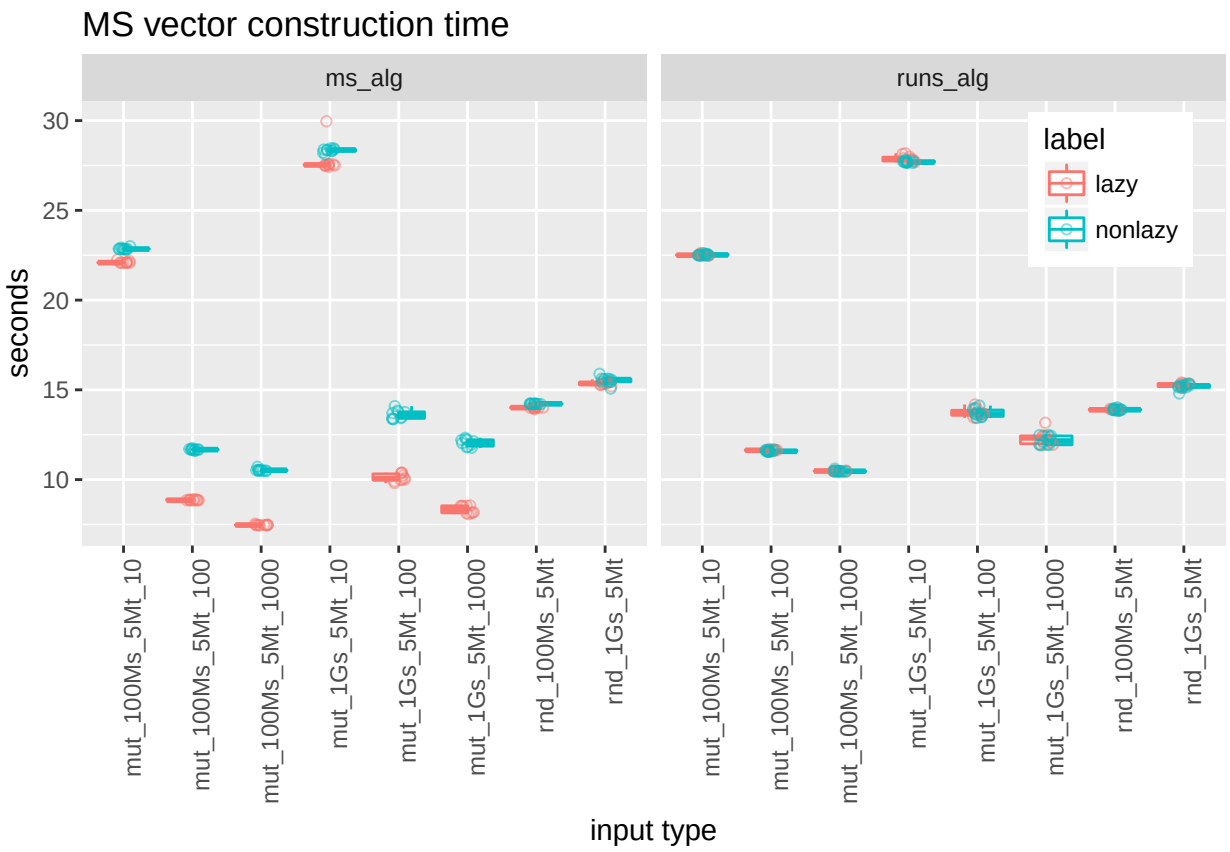


2.2 Code

The lazy and non-lazy versions differ in a couple of lines of code as follows

```
if(flags.lazy){
    for(; I.first <= I.second && h_star < ms_size; ){
        c = t[h_star];
        I = bstep_interval(st, I, c); //I.bstep(c);
        if(I.first <= I.second){
            v = st.lazy_wl(v, c);
            h_star++;
        }
    }
    if(h_star > h_star_prev) // // we must have called lazy_wl(). complete the node
        st.lazy_wl_followup(v);
} else { // non-lazy weiner links
    for(; I.first <= I.second && h_star < ms_size; ){
        c = t[h_star];
        I = bstep_interval(st, I, c); //I.bstep(c);
        if(I.first <= I.second){
            v = st.lazy_wl(v, c);
            h_star++;
        }
    }
}
```

2.3 Run time

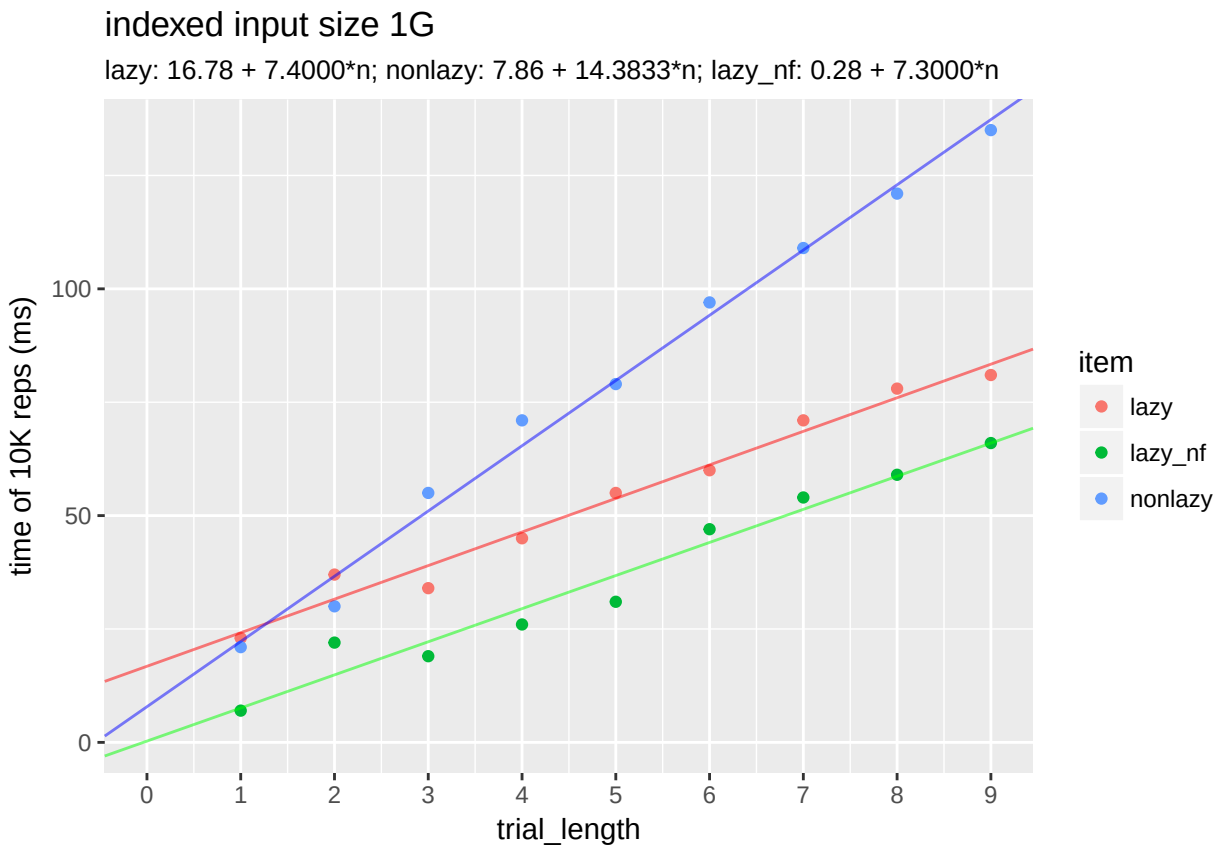

MS vector construction time
ms_alg
runs_alg
label
lazy
nonlazy
seconds
30
25
20
15
10
mut_100Ms_5Mt_10
mut_100Ms_5Mt_100
mut_100Ms_5Mt_1000
mut_1Gs_5Mt_10
mut_1Gs_5Mt_100
mut_1Gs_5Mt_1000
rnd_100Ms_5Mt
rnd_1Gs_5Mt
input type

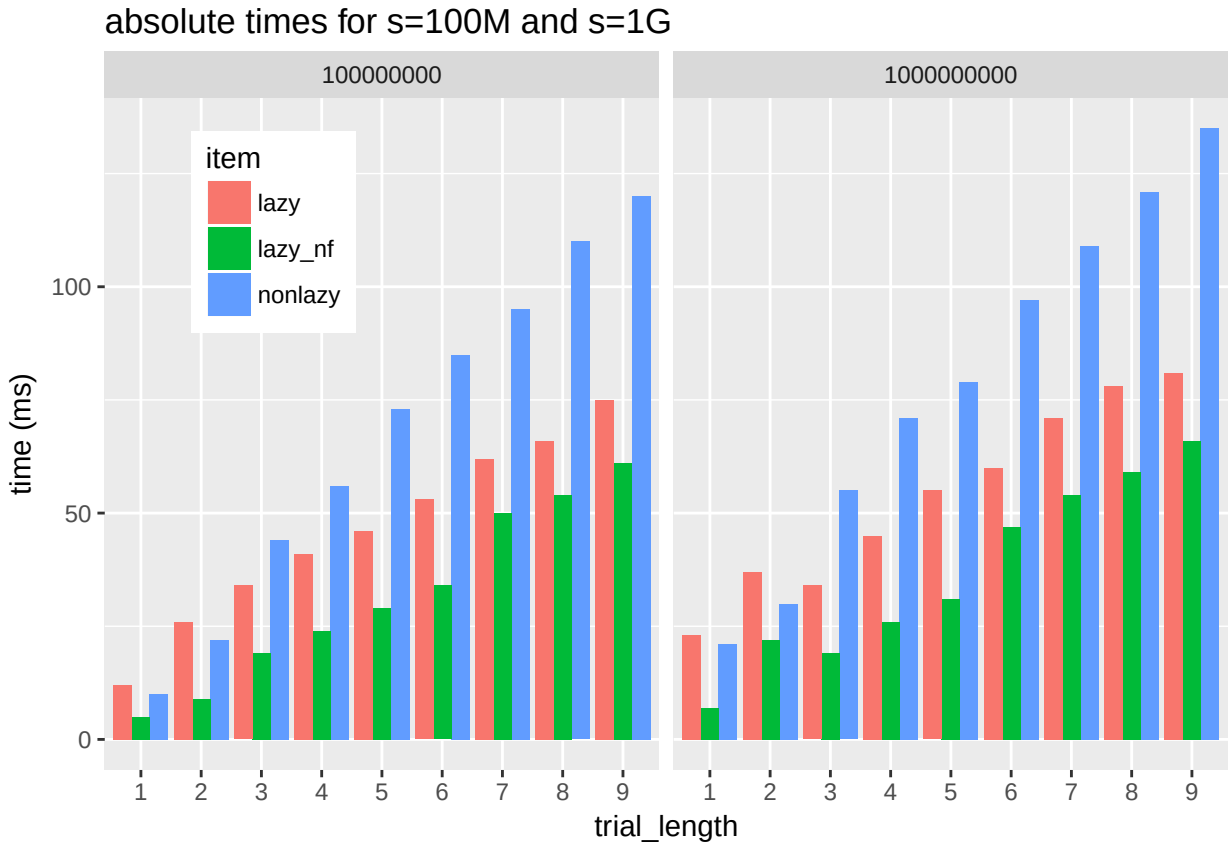
2.4 Sandbox timing

Measure the time of 10k repetitions of

- (lazy) n consecutive `lazy_wl()` calls followed by a `lazy_wl_followup()`
- (nonlazy) n consecutive `wl()` calls
- (lazy_nf) n consecutive `lazy_wl()` calls

```
// lazy
for(size_type i = 0; i < trial_length; i++)
    v = st.lazy_wl(v, s_rev[k--]);
if(h_star > h_star_prev) // // we must have called lazy_wl(). complete the node
    st.lazy_wl_followup(v);
...
// non-lazy
for(size_type i = 0; i < trial_length; i++)
    v = st.wl(v, s_rev[k--]);
...
// lazy_nf
for(size_type i = 0; i < trial_length; i++)
    v = st.lazy_wl(v, s_rev[k--]);
```





Using the linear fits above, this is the expected toal time the `wl()` or `lazy_wl()` calls should take (in ms).

```
##               b_path pred_abs_diff_sec actual_abs_diff_sec
## 1   mut_100Ms_5Mt_10              0.49               -0.75
## 2  mut_100Ms_5Mt_100             -1.40               -2.81
## 3 mut_100Ms_5Mt_1000             -1.59               -3.06
## 4     mut_1Gs_5Mt_10              0.80               -0.58
## 5    mut_1Gs_5Mt_100             -1.47               -3.51
## 6   mut_1Gs_5Mt_1000             -1.69               -3.71
## 7      rnd_100Ms_5Mt              1.06               -0.21
## 8        rnd_1Gs_5Mt              1.11               -0.17
```