diff_mean_test and sample sizes

Christoph Hafemeister

2021-03-19

Issue 98:

> Hi Christoph,
>
> absolutely phenomenal tool set, I'm really excited by the possibility to run DE testing on the output of sctransform! I was wondering if you could comment on what a sufficient minimal number of cells would ideally be for "robust" DE calling between two groups when working with the implementation of diff_mean_test()?

Good question. It will certainly depend on the fold change, so let's run some simulations to get a better idea.

First process some real data to have realistic model parameters.

```
counts <- Seurat::Read10X_h5("~/Projects/data_warehouse/raw_public_10x/Parent_NGSC3_DI_PBMC_filtered_fea
vst_out <- vst(counts, return_cell_attr = TRUE, method = "qpoisson", verbosity = 0,
    residual_type = "none")
# add the arithmetic mean, since vst uses the geometric mean
vst_out$gene_attr$amean <- rowMeans(counts[rownames(vst_out$gene_attr),
    ])
# fit mean-theta relationship; save model for later use to predict
# theta given any mean
df <- left_join(tibble::rownames_to_column(data.frame(vst_out$model_pars),
    var = "gene"), tibble::rownames_to_column(vst_out$gene_attr, var = "gene"),
    by = "gene")
vst_out$fit <- loess(formula = log10(theta) ~ log10(amean), data = df,
    control = loess.control(surface = "direct"))
```

We want to be able to simulate count data for a gene given its mean. Define a helper function to do that.

```
sim <- function(vst_out, amean, n) {
    theta <- 10^predict(vst_out$fit, newdata = log10(amean))
    return(MASS::rnegbin(n = n, mu = amean, theta = theta))
}
```

Iterate over conditions to test

```
mean1_v <- c(0.001, 0.01, 0.1, 1)
log2fc_v <- c(-10, -5, -4, -3, -2, -1, 1, 2, 3, 4, 5, 10)
n_v <- 1:20 * 10  # number of cells per group
k <- 1000  # repetitions
```

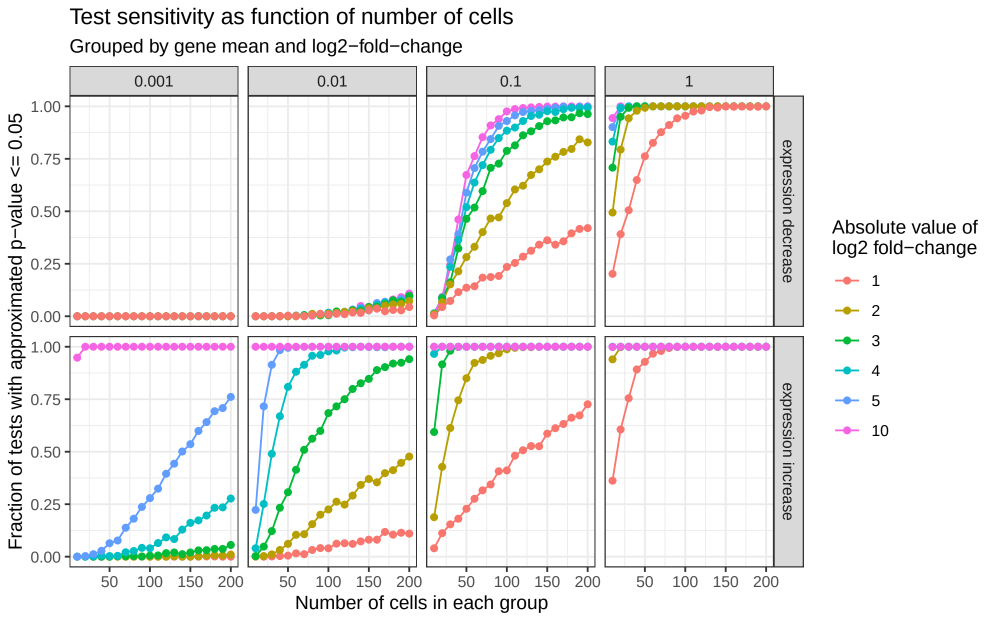
```r
genenames <- sprintf("gene_%05d", 1:k)

res_lst <- list()
for (mean1 in mean1_v) {
    for (log2fc in log2fc_v) {
        for (n in n_v) {
            # message(mean1, ' ', log2fc, ' ', n_v)
            grp <- factor(0:(2 * n - 1)%/%n)
            cellnames <- sprintf("cell_%04d", 1:(2 * n))
            dn <- list(genenames, cellnames)

            mat1 <- matrix(data = sim(vst_out, amean = mean1, n = n * k),
                nrow = k)
            mat2 <- matrix(data = sim(vst_out, amean = mean1 * (2^log2fc),
                n = n * k), nrow = k)
            mat <- as(cbind(mat1, mat2), "dgCMatrix")
            dimnames(mat) <- dn
            de_res <- diff_mean_test(y = mat, group_labels = grp, compare = c("0",
                "1"), log2FC_th = 0, mean_th = 0, cells_th = 0, verbosity = 0,
                R = 499)
            ret <- c(mean1, log2fc, n, sum(de_res$emp_pval <= 0.05, na.rm = TRUE)/k,
                sum(de_res$emp_pval <= 0.01, na.rm = TRUE)/k, sum(de_res$pval <=
                  0.05, na.rm = TRUE)/k, sum(de_res$pval <= 0.01, na.rm = TRUE)/k)
            res_lst[[length(res_lst) + 1]] <- ret
        }
    }
}
res <- do.call(rbind, res_lst)
colnames(res) <- c("mean1", "log2fc", "n", "sens_emp5", "sens_emp1", "sens5",
    "sens1")
res <- data.frame(res) %>% mutate(change = case_when(log2fc < 0 ~ "expression decrease",
    TRUE ~ "expression increase"))
```

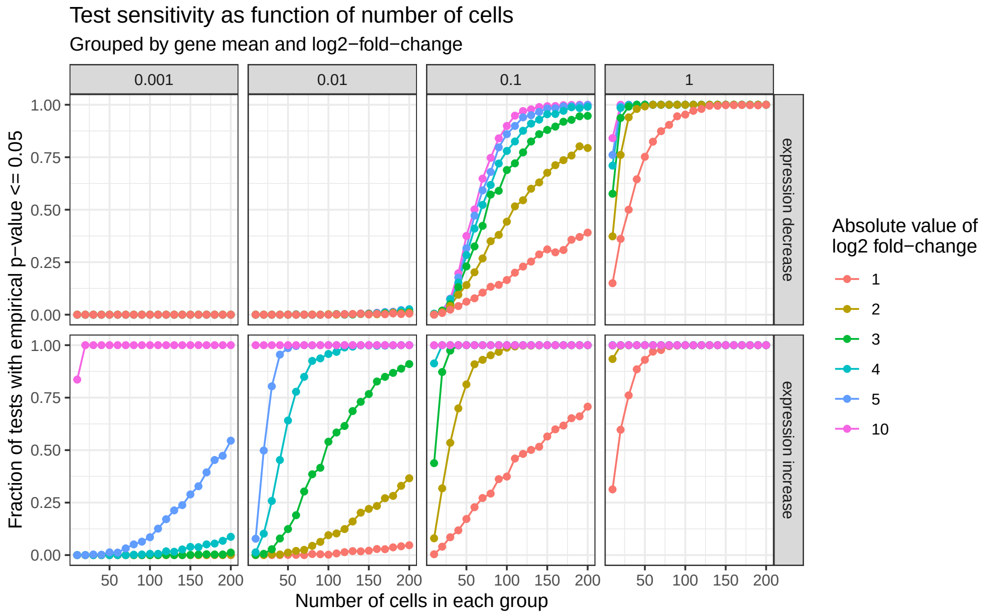
Look at DE recovery rate at approximated p-value cutoff of 0.05

```r
ggplot(res, aes(n, sens5, color = factor(abs(log2fc)))) + geom_line() +
    geom_point() + facet_grid(change ~ mean1) + ggtitle(label = "Test sensitivity as function of number
    subtitle = "Grouped by gene mean and log2-fold-change") + scale_color_discrete(name = "Absolute valu
    xlab("Number of cells in each group") + ylab("Fraction of tests with approximated p-value <= 0.05")
```



Look at DE recovery rate at empirical p-value cutoff of 0.05

```
ggplot(res, aes(n, sens_emp5, color = factor(abs(log2fc)))) + geom_line() +
    geom_point() + facet_grid(change ~ mean1) + ggtitle(label = "Test sensitivity as function of number
    subtitle = "Grouped by gene mean and log2-fold-change") + scale_color_discrete(name = "Absolute valu
    xlab("Number of cells in each group") + ylab("Fraction of tests with empirical p-value <= 0.05")
```



Print results for 100 cells and absolute log2FC of 2

```
filter(res, n == 100, abs(log2fc) == 2) %>% select(c(1:4, 6))
```

mean1	log2fc	n	sens_emp5	sens5
0.001	-2	100	0.000	0.000
0.001	2	100	0.000	0.000
0.010	-2	100	0.002	0.012
0.010	2	100	0.095	0.225
0.100	-2	100	0.443	0.539
0.100	2	100	0.988	0.988
1.000	-2	100	1.000	1.000
1.000	2	100	1.000	1.000

Session info

```
sessionInfo()
#> R version 4.0.2 (2020-06-22)
#> Platform: x86_64-apple-darwin17.0 (64-bit)
#> Running under: macOS Catalina 10.15.7
#> 
#> Matrix products: default
#> BLAS:   /Library/Frameworks/R.framework/Versions/4.0/Resources/lib/libRblas.dylib
#> LAPACK: /Library/Frameworks/R.framework/Versions/4.0/Resources/lib/libRlapack.dylib
#> 
#> locale:
#> [1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8
#> 
#> attached base packages:
#> [1] stats     graphics  grDevices utils     datasets  methods   base     
#> 
#> other attached packages:
#> [1] patchwork_1.1.0.9000   ggrepel_0.8.2          dplyr_1.0.2           
#> [4] knitr_1.30             Seurat_3.9.9.9008      sctransform_0.3.2.9005
#> [7] reshape2_1.4.4         ggplot2_3.3.2          Matrix_1.2-18         
#> 
#> loaded via a namespace (and not attached):
#>  [1] nlme_3.1-149          matrixStats_0.57.0    bit64_4.0.5          
#>  [4] RcppAnnoy_0.0.16      RColorBrewer_1.1-2    httr_1.4.2           
#>  [7] tools_4.0.2           R6_2.5.0              irlba_2.3.3          
#> [10] rpart_4.1-15          KernSmooth_2.23-17    uwot_0.1.8.9001      
#> [13] mgcv_1.8-33           lazyeval_0.2.2        colorspace_2.0-0     
#> [16] withr_2.3.0           tidyselect_1.1.0      gridExtra_2.3        
#> [19] bit_4.0.4             compiler_4.0.2        formatR_1.7          
#> [22] hdf5r_1.3.2           plotly_4.9.2.1        labeling_0.4.2       
#> [25] scales_1.1.1          lmtest_0.9-38         spatstat.data_1.4-3  
#> [28] ggridges_0.5.2        pbapply_1.4-3         spatstat_1.64-1      
#> [31] goftest_1.2-2         stringr_1.4.0         digest_0.6.27        
#> [34] spatstat.utils_1.17-0 rmarkdown_2.5         pkgconfig_2.0.3      
#> [37] htmltools_0.5.1.1     highr_0.8             fastmap_1.0.1        
#> [40] htmlwidgets_1.5.2     rlang_0.4.9           shiny_1.5.0          
#> [43] farver_2.0.3          generics_0.0.2        zoo_1.8-8            
#> [46] jsonlite_1.7.2        ica_1.0-2             magrittr_2.0.1       
#> [49] Rcpp_1.0.5            munsell_0.5.0         abind_1.4-5          
#> [52] reticulate_1.16       lifecycle_0.2.0       stringi_1.5.3        
#> [55] yaml_2.2.1            MASS_7.3-53           Rtsne_0.15           
#> [58] plyr_1.8.6            grid_4.0.2            parallel_4.0.2
#> [61] listenv_0.8.0         promises_1.1.1        crayon_1.3.4.9000
#> [64] deldir_0.1-29         miniUI_0.1.1.1        lattice_0.20-41
#> [67] cowplot_1.1.0         splines_4.0.2         tensor_1.5
#> [70] pillar_1.4.7          igraph_1.2.6          future.apply_1.6.0
#> [73] codetools_0.2-16      leiden_0.3.3          glue_1.4.2
#> [76] evaluate_0.14         data.table_1.13.2     vctrs_0.3.5
#> [79] png_0.1-7             httpuv_1.5.4          gtable_0.3.0
#> [82] RANN_2.6.1            purrr_0.3.4           polyclip_1.10-0
#> [85] tidyr_1.1.2           future_1.19.1         xfun_0.19
#> [88] rsvd_1.0.3            mime_0.9              xtable_1.8-4
#> [91] later_1.1.0.1         survival_3.2-3        viridisLite_0.3.0
#> [94] tibble_3.0.4          cluster_2.1.0         globals_0.13.1
#> [97] fitdistrplus_1.1-1    ellipsis_0.3.1        ROCR_1.0-11
```