ComparativeMarkerSelection Documentation

Module name: ComparativeMarkerSelection
Description: Computes significance values for features using several metrics, including FDR(BH), Q Value,, FWER, Feature-Specific P-Value, and Bonferroni.
Author: Joshua Gould, Gad Getz, Stefano Monti (Broad Institute)
gp-help@broad.mit.edu

The ComparativeMarkerSelection module includes several approaches to determine the features that are most closely correlated with a class template and the significance of that correlation. If the input class template has more than two classes, than a one-versus-all comparison is performed for each class. Note that the p values obtained from the one-versus-all comparison are not fully corrected for multiple hypothesis testing. The module outputs a file containing the following columns:

1. ***Rank*** - The rank of the feature within the dataset based on the value of the test statistic. If a two-sided p value is computed, the rank is with respect to the absolute value of the statistic.
2. ***Feature*** - The feature name.
3. ***Score*** - The value of the test statistic.
4. ***Feature P*** - The feature-specific p value based on permutation testing.
5. ***FWER*** (Family Wise Error Rate) - the probability of at least one null hypothesis/feature having a score better than or equal to the observed one. This measure is not feature-specific.
6. ***FDR (BH)*** - An estimate of the false discovery rate by the Benjamini and Hochberg procedure (3). The FDR is the expected proportion of erroneous rejections among all rejections.
7. ***Bonferroni*** - The value of the Bonferroni correction applied to the feature specific p value.
8. ***Q Value*** - An estimate of the FDR using the procedure developed by Storey and Tibshirani (4).
9. **maxT** - The adjusted *p*-values for the maxT multiple testing procedure described in (5), which provides strong control of the FWER.

The results from the ComparativeMarkerSelection algorithm can be viewed with the ComparativeMarkerSelectionViewer.

Booster

The idea behind the "Booster" is stop performing permutations (Monte Carlo sampling) when the p-value can be estimated with high enough accuracy.

The algorithm

(1) Set a bank of permutations, $B \leftarrow N*p$ where N is the number of features (hypotheses) and p is the average number of permutations per feature one is willing to run. This is the number of permutations performed when the Booster is not applied.

(2) Generate a list of active features $F \leftarrow \{1,\ldots,N\}$
(3) Initialize the vector of number of performed iterations (n_i) and extreme cases encountered (k_i) to 0: $n_i \leftarrow 0$, $k_i \leftarrow 0\ \forall\ i=1,\ldots,N$.
(4) Loop over the active features in F until F is empty or B=0
    a. Permute the labels and calculate a score, s_i.
    b. Compare s_i to the observed score S_i. If $s_i \geq S_i$ then increase k_i. $k_i \leftarrow k_i+1$
    c. $n_i \leftarrow n_i+1$
    d. If $2 \cdot 1.96 \cdot \sqrt{\frac{n_i - k_i + 1}{(n_i + 3)(k_i + 1)}} < \theta$ remove feature i from F (see Appendix for the derivation of this stopping criterion). In case one performs a two-sided test remove i from F if the above criterion holds when replacing k_i with $\min(k_i, n_i-k_i)$.
    e. $B \leftarrow B$-1

Note that one could use a different stopping criterion which stops whenever one is confident enough that the p-value is insignificant (above some threshold). We did not use this approach since the large p-values are used to estimate the false discovery rate.

Smoothing

Following the Monte Carlo sampling, one needs to estimate the p-value for each hypothesis, $\hat{p}_i$. Taking a Bayesian approach, p_i has a Beta(k_i+1, n_i - k_i +1) distribution, assuming a uniform prior. Two common estimators are: (i) The mode of the distribution, which coincides with the maximum likelihood estimator (since we are took a uniform prior), $\hat{p}_i = k_i / n_i$.(ii) The expectation of the distribution which yields the Laplace smoothed estimator, $\hat{p}_i = \frac{k_i + 1}{n_i + 2}$ [P. de Laplace, marquis de, *Essai philosophique sur les probabilit´es* (Courcier, Paris, 1814), trans. by F. Truscott and F. Emory, *A philosophical essay on probabilities* (Dover, New York, 1951)].

Confidence Interval

One can use the distribution of p_i to calculate confidence intervals. We take the maximal probability approach which identifies the interval with most probable values for p_i whose probabilities amount to 95% (this is also the shortest one that covers 95%). Formally, we search for p_i^L and p_i^H such that $f(p_i^L) = f(p_i^H)$ and $F(p_i^H) - F(p_i^L) = 0.95$, where f and F are the Beta(k_i+1, n_i - k_i +1) density and cumulative distribution functions respectively. In a two-sided case, one should use 2*min(k_i, n_i-k_i) instead of k_i.

Appendix

The stopping criterion is derived using a Gaussian approximation of the Beta distribution, i.e. we use the standard deviation of the Beta distribution, σ, and take 1.96σ from each side of the mean for the 95% interval. The accuracy

we want to achieve is such that $\frac{p_i^H - p_i^L}{\hat{p}_i} < \theta$ using the mean (μ) and standard deviation of the Beta distribution

we get, $\frac{2 \cdot 1.96 \cdot \sigma}{\mu} = \frac{2 \cdot 1.96 \cdot \sqrt{\frac{(k_i+1)(n_i-k_i+1)}{(n_i+2)(n_i+3)}}}{\frac{k_i+1}{n_i+2}} = 2 \cdot 1.96 \cdot \sqrt{\frac{n_i-k_i+1}{(k_i+1)(n_i+3)}} < \theta$

Since the LHS of the stopping criterion is decreasing with k_i one is confident that once the inequality holds it will hold for any greater k_i.

Exact p-value calculation

In case the number of samples, S, is small enough one can exhaustively perform all possible $\binom{S}{S_1}$ permutations

where S_1 is the number of samples in one of the two classes.

In this case $p_i = k_i / n_i$ is the exact p-value and there is no need for a confidence interval.

Words of caution

In several places in these derivations there is a hidden assumption that the probability to encounters the same value twice is very low and is taken to be 0. If this is far from the truth one should take it into account. For example, the other extreme tail (used in a two-sided test) does not necessarily have the same p-value.

Parameters:

Name	Description
input filename	The input file - .res, .gct, .odf type=Dataset
cls filename	The class file - .cls
confound variable cls filename	The class file containing the confounding variable - .cls
test direction	The test to perform (up-regulated for class 0, up-regulated for class 1, two-sided)
test statistic	The statistic to use
min std	The minimum standard deviation if test statistic is T-Test (min std)
number of permutations	The number of permutations to perform
complete	Whether to perform all possible permutations
balanced	Whether to perform balanced permutations
random seed	The seed of the random number generator
smooth p values	Whether to smooth p-values
significance booster	Whether to attempt to increase the p-value confidence for significant features
theta	Value for removing features when using significance booster
output file	The name of the output file

Return Value:

An odf file of type ComparativeMarkerSelection

References:

1. Golub, T.R., et al., *Molecular Classification of Cancer: Class Discovery and Class Prediction by Gene Expression.* Science, 1999. **286**(5439): p. 531-537.
2. Slonim, D.K., et al., *Class Prediction and Discovery Using Gene Expression Data*, in *RECOMB 2000: The Fourth Annual International Conference on Research in Computational Molecular Biology*. 2000: Tokyo, Japan. p. 263-272.
3. Benjamini, Y. and Y. Hochberg, *Controlling the False Discovery Rate: A Practical and Powerful Approach to Multiple Testing.* Journal of the Royal Statistical Society. Series B (Methodological), 1995. **57**(1): p. 289-300.
4. Storey, J.D. and R. Tibshirani, *Statistical significance for genomewide studies.* PNAS, 2003. **100**(16): p. 9440-9445.
5. Westfall, P.H. and S.S. Young, *Resampling-Based Multiple Testing: Examples and Methods for P-Value Adjustment*. Wiley Series in Probability and Statistics. 1993, New York: Wiley.

Platform dependencies:

Task type:	Gene List Selection
CPU type:	any
OS:	any
Java JVM level:	1.4
Language:	Java, R