Seminar on collaborative software development

Instructions

Module 3 will be examined by a seminar on **13 December** that is based on the material in lecture 4 and 5. You will work in groups of two to three students and carry out a mini-project, where you together develop a small piece of software. During the seminar you will give an **oral presentation** where you present your software solution and a case study on a dataset. **Attendance is mandatory**. The presentation should be based on a **slideshow that is uploaded in Canvas** before the seminar, no later than **12 December**. Each presentation will be about 15 min long followed by 5-10 min questions. The oral presentation and slideshow will be graded together according to the criteria below. The grade for the seminar is either Pass or Fail. In order to Pass, all criteria below must be Passed.

Grading criteria	**Fail**	**Pass**
GitHub repository	The software has not been developed on GitHub, or lacks files required for an R package	The software is shared on GitHub with version history and required files
Programming practices	Good programming practices have not been considered, or it is unclear how they have been used in software design	The presentation includes explanation of how good programming practices have been incorporated into the software
Documentation	The software lacks reference and/or user documentation, or the documentation has poor quality	Clear documentation is provided for functions (reference) and users (manual)
Collaboration	The software has not been developed as a team, or GitHub collaboration tools such as pull requests have not been used	Everyone in the group has contributed with commits to the GitHub repo, as evidenced from e.g., version history and GitHub pull request discussion
Case study	The software does not work correctly, or has not been evaluated according to instructions	The software has been evaluated correctly and the results presented in a suitable way
Time	The presentation takes > 20 or < 10 min	Give the presentation within 15 +/- 5 min

In case you fail the seminar, you will have **one** opportunity to be re-examined. In this case you need to revise the presentation and/or the software, and possibly present it again at a seminar that will be scheduled later on.

Email contact for questions regarding the seminar: benjamin.ulfenborg@his.se

Software development mini-project

The task in the mini-project is to create an R package that can perform enrichment analysis on an RNA-seq dataset. The package should be based on the R packages *edgeR*, *clusterProfiler*, *enrichplot*, *org.Hs.eg.db* and *openxlsx*. It is possible to also include other dependencies as needed. One member in the group starts by creating a GitHub repo, committing a README file and adding the other group members as collaborators. Next the repo should be cloned locally and a new R package created in RStudio inside the local repo. Changes should be staged and pushed to the remote. Then all other student group members can clone the remote repo and work in parallel on the tasks listed below. Decide within the group how the work should be divided. Strive to divide the work evenly among you. In order to make use of GitHub's pull request feature (one of the grading criteria), make sure to work in separate Git branches as new features are added, and later merge these into the main branch. The R package should be able to

- Import RNA-seq count data into R and perform filtering to remove low-expressed genes. You will use the E-MTAB-2523 RNA-seq counts as a case study dataset when developing the package. This can be downloaded together with its sample table from the seminar page in Canvas. The dataset contains partially paired samples from colorectal carcinoma and health control tissue.
- Perform statistical analysis to identify differentially expressed genes between the carcinoma and normal samples. Genes should be filtered on FDR and log2 fold change.
- Perform over-representation analysis with the functions *enrichGO* and *enrichKEGG* in the *clusterProfiler* package. The key type for the genes in the dataset is SYMBOL. Note that *enrichKEGG* expects Entrez gene IDs (ENTREZID), so IDs need to be converted before analyzing KEGG pathways. The *clusterProfiler* package provides the convenient *bitr* function for this. As an alternative, you may perform gene set enrichment analysis with the gseGO and gseKEGG functions.
- Export the table with differentially expressed genes to Excel and create plots to visualize the results of the enrichment analysis. Examples of functions to use are *cneplot*, *dotplot* and *treeplot*. The last one depends on a term similarity matrix, which is calculated by calling *pairwise_termsim* on the enrichment analysis results.

Keep the **programming practices** covered in Lecture 5 in mind when developing the software. Also provide function **reference documentation** and write a short **user manual** using R Markdown explaining to users how the package should be used. When package development is finished, show that it works by using it to **analyze the case study dataset**.

Presentation layout

Below is a suggested layout for the slideshow you will prepare for your presentation. You don't have to follow this exact template, but be sure to include all information mentioned in the grading criteria. As a general guideline, you should strive to have about one slide per minute of presentation (could be less but preferably not more). Having many more will force you to rush through the slides to keep time. Since your audience has not seen the slides before, it can make the presentation hard to follow. Also avoid having too much text in the slides. Bullets are fine, but not make them several sentences long (i.e., don't fill slides with text). The slideshow is not a report, it is only there to help maintain the thread of the presentation. Bullets are best used for showing e.g., which method or parameters were used, not exactly what you are going to say.

Layout:

- Introduction
  - Group members
  - Software design, i.e., describe what functions/classes were defined and their purpose, what the functions return and in what order they should be called (could be illustrated with a flowchart)
- GitHub repository
  - Main page showing organization of files, README and URL
  - Commit history (can be viewed by going to github.com/user/repo/commits)
- Programming practices
  - What programming practices were considered and how this impacted software design, i.e., exemplify how the guidelines were implemented
- Documentation
  - Reference documentation available for functions (when using *help* function in R)
  - User manual compiled from R Markdown to HTML
- Collaboration
  - How work was divided among the group members
  - Excerpt from GitHub issues and/or pull requests made during development
- Case study
  - Summary of dataset used
  - Results from analysis, e.g., list of DEGs and enrichment result plots