DEAME - Differential Expression Analysis Made Easy

Milena Kraus, Guenter Hesse, Tamara Slosarek, Marius Danner, Ajay Kesar, Akshay Bhushan, and Matthieu-P. Schapranow

Hasso Plattner Institute
Rudolf-Breitscheid-Str. 187, 14482 Potsdam, Germany
{milena.kraus,guenter.hesse,schapranow}@hpi.de
{tamara.slosarek,marius.danner,ajay.kesar}@student.hpi.uni-potsdam.de
http://www.hpi.de

Abstract. Differential gene and protein expression analysis reveals clinically significant insights that are crucial, e.g., for systems medicine approaches. However, processing of data still needs expertise of a computational biologist and existing bioinformatics tools are developed to answer only one research question at a time. As a result, current automated analysis pipelines and software platforms are not fully suited to help research-oriented clinicians answering their hypotheses arising during their clinical routine. Thus, we conducted user interviews in order to identify requirements and evaluate our research prototype of an application that i) automates the complete preprocessing of RNA sequencing data in a way that enables rapid hypothesis testing, ii) can be run by a clinician and iii) helps interpreting the data. In our contribution, we share details of our preprocessing pipeline, software architecture of our first prototype and the identified functionalities needed for rapid and clinically relevant hypothesis testing.

Keywords: Differential Expression Analysis, Explorative Analysis, Rapid Hypothesis Testing, Web Application

1 Introduction

Analysis of differential expression (DE) is the process of identifying genes or proteins that have an altered level of expression in a group of samples, which is statistically significant when compared to another. The differences in expression levels may be the result of a disease or other perturbations of the examined cells or tissues. Therefore, the identification of the differences can lead to biomarkers of a disease [11] or a transcriptomic profile that may be reversed through a new or existing treatment.

The development of next-generation sequencing (NGS) techniques have enabled the usage of RNA sequencing (RNAseq) data as primary source for DE analysis [5]. Processing of raw RNA reads includes a pipeline of quality control, read alignment and quantification, all of which require a sophisticated selection

of tools and methods [6]. Byron et al. (2016) describe examples for how analysis of RNAseq can benefit clinical practice. However, the great flexibility and resulting complexity for RNAseq have hindered its path to the clinic so far [5].

In recent years, many studies, e.g., in the context of systems medicine, included a detailed clinical examination of patients, supported by a molecular characterization via omics technologies [10]. Oftentimes these studies have an observational character and do not include the effect of an active perturbation, e.g., testing a new drug or therapy in a defined environment. Thus, effects on the molecular level, e.g., in gene expression, are the result of many *in vivo* factors. Research-oriented clinicians, i.e., physicians that work in part as a physician but also conduct research on their patients, observe these *in vivo* factors, such as gender or previous diagnoses, but only have a limited understanding and capability to interpret DE results. Contrary, computational biologists have little insights into clinical practice and thus, their research hypotheses are mainly motivated by literature. In order to find and validate a joint research hypothesis the clinician and the computational biologist must interact and communicate efficiently.

In our contribution, we share a software systems architecture as well as our first prototypical web application, which will enable clinicians and computational biologists to rapidly perform exploratory hypothesis testing based on gene and protein expressions and to interpret their results on the clinical data given.

Our contribution is structured as follows: We first describe the generic process of how differential expression analysis is performed traditionally in Section 2 and how it has been implemented in related work so far (Section 3). In Section 4 we share details on our user research, which results in specific software requirements. The developed software systems architecture and application prototype are described in Section 5.

2 The Differential Expression Analysis Process

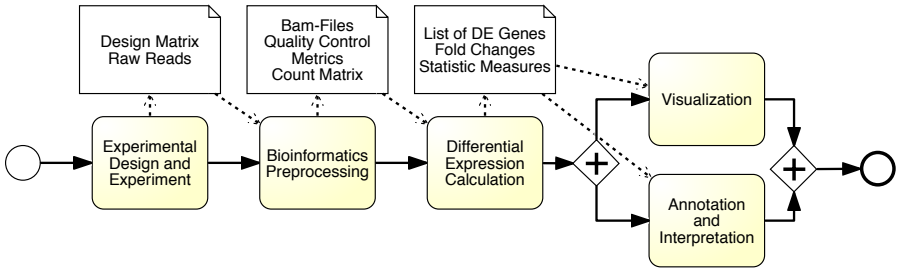

Fig. 1: Generic differential expression process steps and their results.

We provide a generic process model of all steps needed for a DE analysis emanating from an RNA sequencing experiment using Business Process Modeling Notation (BPMN) in Figure 1. The pipeline is based on Conesa et al. (2016) [6] and resembles many of the implemented pipelines described in Section 3.

Experimental Design Inherent to DE analysis are at least two groups of samples that are assumed to show differences in gene expression. These groups need to be specified before *in vitro* testing in order to plan and design the wet lab process, such as treatment with a specific chemical or drug. In contrast, the clinical context usually assumes *in vivo* experiments, e.g., biopsy analysis for a group of diseased patients as well as a healthy control group. Many of these studies are purely observational and factors that contribute to the differential gene expression are multiple and therefore not defined as clearly as in the *in vitro* setup. Confounding factors, such as batch effects or other patient specific clinical parameters, should be recorded and taken into account when analyzing DE results. As a result, the researcher needs to define a design formula which resembles the research hypothesis and is the basis of any DE experiment. The formula is then provided as input of the pipeline.

Bioinformatics Preprocessing The sequencing process results in raw reads. Raw reads go through quality control and in some cases need to be trimmed from adapter sequences prior to alignment. All reads are aligned to a reference genome or transcriptome. In the best case all genomic ranges, such as a gene, an exon or coding region, are covered by multiple reads after the alignment step. Counting tools calculate the exact quantity of reads per given genomic range.

Differential Expression Calculation DE calculation is the statistical process of finding significant expression differences of two or more groups as defined in the experimental design. In short, all counts of a genomic range in one group are compared to the counts of the same range in another group of samples. The calculation provides information about the fold change, i.e., how much more counts where found in one group when compared to the other. Additionally, p-values are given, which are adjusted for multiple testing as many data sets comprise 10-20 k genomic regions to compare.

Visualization Visualization of results is a critical part in DE analysis as raw and transformed data as well as DE results are usually high in dimension and therefore need to be displayed in a comprehensive format. Frequently used techniques are principal component analysis and clustering of data. Both give an impression of similarity between the analyzed samples. For example, plotting samples on their corresponding first and second principal component (dimension of largest variation) should result in scatters of samples grouped according to the experimental design formula. Accordingly, clustering algorithms should be able to find clusters and a dendrogram resembling the desired study groups. Clustered

heatmaps are specifically popular as they can display sample-to-sample as well as gene-to-gene relationships and the corresponding normalized and log transformed count values in a single diagram. Volcano plots depict the p-value versus expression fold change between two conditions. Differentially expressed genes or proteins are usually marked and therefore the plot gives a good overview of all results. Many more diagnostic plots are used as, e.g., depicted in a bioconductor workflow [17].

Annotation and Interpretation Annotation and interpretation of results is a critical and complex part of the analysis. Typically, more than 100 genes/proteins are found to be differentially expressed between patient groups. Regarding the most relevant expression changes, a manual search for function and involved pathways is performed. Gene Ontology (GO) annotation and Gene Set Enrichment Analysis (GSEA) help to find perturbed anatomical structures, biochemical processes or pathways in an automated manner.

3 Related Work

Gaur et al. provide an overview about automated RNAseq analysis platforms and a short description of their utility [9]. Four of the tools listed by Gaur et al. show similarities to our apporach:

The main aim of RAP [7] is to provide an RNAseq tool that does not need to be installed on the client side. The web interface provides possibility for data submission and a browsing facility for results exploration. While the overall appearance seems more user friendly than command line tools, the platform is suited for users with bioinformatics knowledge that are able to configure pipelines and interpret results. Furthermore, RAP offers a great variety of possibilities for analyzing RNAseq data and thus, no focus on DE analysis. Especially visualizations and plots are not available so far. DE genes are given as lists.

RNAminer [14] provides three different fully parameterized pipelines that work simultaneously and results are consolidated among the pipeline. However, the resulting DE genes are given as text files and any new hypothesis needs an upload of files and a manual specification of two groups of samples at the maximum.

QuickNGS [21] has many options to analyze a variety of NGS data and thus lacks visualizations and functions that are specific for RNAseq analysis. Again, results are only given in lists. Plots are limited to a static clustered heatmap and a PCA plot. Additionally, experimental design is static and as described within the publication only usable for two groups (sample and control) plus batch effects.

Wolfien et al. (2016) implemented TRAPLINE for automated analysis of RNAseq data, evaluation and annotation within the Galaxy framework [22, 3]. The TRAPLINE workflow was built to enable experimentalists to analyze data without requiring programming skills [22]. In addition to preprocessing and DE

calculation, it provides several lists of results and help or links for visualizing data. Additionally, links to annotation and interpretation tools are given.

In general, most state-of-the-art tools are designed for users with some bioinformatics knowledge that is needed to configure the pipelines and interpret the data. Moreover, some applications are built on the assumption that there is only a single experimental design or perturbation to be tested on. As a result, all programs mentioned have at least two of the following drawbacks: (i) No ad-hoc or only static visualization for DE results, (ii) a static experimental design and/or a resulting (iii) cumbersome reconfiguration for any new hypothesis to be tested. Additionally, the complete pipeline including preprocessing is repeated in every analysis of the input data, which results in redundancy when multiple hypotheses are tested on the same or a subset of samples. While the listed tools work well for interventional studies and a single hypothesis, a new approach is needed in the case of observational setups and many hypothesis.

4 Requirements Engineering

The idea and development of the web application has been discussed and evaluated iteratively within the SMART (Systems Medicine Approach for Heart Failure) consortium based on an RNAseq raw data and clinical data raised within an observational study on heart failure patients. Several iterations on mockups and prototypes were conducted within the SMART consortium, which consists of research-oriented clinicians, molecular and computational biologists.

In a literature survey, we identified relevant and state-of-the-art preprocessing tools as well as DE calculation and visualization options. In order to validate the pipeline as described in the literature, we conducted informal phone interviews with experts from different research institutes that are focused on the analysis of RNAseq data and DE analysis. We discussed all steps of the technical pipeline to determine the acceptance of tools within the user community and shortcomings of selected programs.

In the following, key findings gathered in user research and literature review are assembled to identify concrete user groups of our application.

4.1 User Groups

We identified and characterized two user groups of our application: The **Research-oriented Clinician** who is interested in (i) testing own hypotheses based on daily observations and assessed clinical parameters and (ii) interpretation of DE results in the clinical context, e.g., if results point to a disease, a potential treatment or interesting research directions. All of that should not require any programming skills. The **Computational Biologist** is primarily interested in a statistically accurate preprocessing pipeline and calculation of DE results. The execution of the pipeline should require minimum input and configuration. It should allow ad-hoc exploration and analysis of DE experiment results. Furthermore, the computational biologist would like to get publication-ready result reports.

While the computational biologist has little insights with respect to the patients studied and the resulting hypotheses, the clinician cannot perform bioinformatic processes and algorithms alone. Frequently, the clinician has no experience with omics data and therefore does not know what information can be obtained from it. Communication on interesting results and strategies on further investigations is therefore hampered. Therefore, both user groups need a platform that provides a common ground for discussion.

4.2 Software Requirements

Based on our user research observations and the shortcomings of related platforms as depicted in Section 3, we specified the following software requirements (R) of our DEAME application.

R 1 Automated Preprocessing: Only a single program execution is needed to preprocess raw RNAseq reads to count matrices.
R 2 Pipeline Configuration Options: The pipeline may be altered and configured by the computational biologist, but does not need to.
R 3 Split of Pipeline: Bioinformatics tools within the processing pipeline need to allow a split into preprocessing and experimental design/DE calculation.
R 4 State-of-the-art Tools: All bioinformatics tools need to be well-established and accepted within the scientific community.
R 5 Clinical Information: Clinical data on the samples needs to be readily accessible to setup the experimental design.
R 6 Rapid Experimental Design Creation: The translation of the clinician's hypothesis into an experimental design matrix needs to be easy and fast.
R 7 Interactive Visualization of Results: Results of DE calculation are high in dimensionality and need proper and interactive visualization.
R 8 Actionable Information on Results: Additional information on DE calculation results need to be provided within the application context, i.e., publications on regulated genes may be available.
R 9 Usability: The overall workflow should resemble the research process. The representation needs to be visually appealing but at the same time correct in content. The application provides sufficient features for the computational biologist yet comprehensible for the clinician.

5 DEAME Application

Our DEAME application is part of the systems medicine IT infrastructure (SMART IT platform) described in [12] and uses resources, such as the worker framework and the in-memory database, provided by the AnalyzeGenomes (AG) platform [18]. In Figure 2, the overall software architecture of the DEAME application as well as relevant parts of the SMART platform are modeled using Fundamental Modeling Concepts (FMC). A thorough explanation of all components will be given in this section.

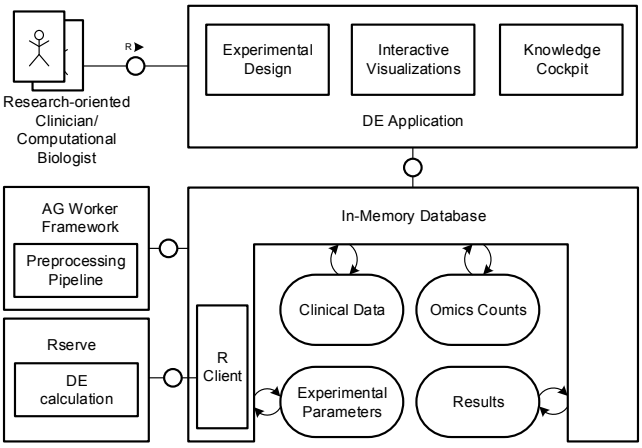

Fig. 2: Software system architecture of the DEAME application including parts provided by the SMART and AnalyzeGenomes IT infrastructure [12, 18].

5.1 Data Layer

An in-memory database contains all frequently accessed data: The patient centric star schema of the SMART platform was expanded within the experiment section (please refer to [12] for further details on the clinical data and security aspects). Tables for counts, as they are produced within the preprocessing as well as intensities from, e.g., proteomics data, are added as well as tables for experimental parameters and results of DE calculation. Furthermore, an R client is established to perform DE calculation within an Rserve instance.

5.2 Platform Layer

The platform layer contains the preprocessing pipeline, experimental setup information and DE calculation functionality. The split into preprocessing and experimental design plus DE calculation is a design decision that limited the selection of tools to be used within the pipeline when compared to the traditional setup as in Section 2. The split resembles the need given within a clinical setting, where many hypotheses may be tested and thus, the experimental design for DE calculation is not known before preprocessing of raw data. As a result, preprocessing and DE calculation are independent from each other.

Technical Preprocessing Pipeline In our architecture the preprocessing is embedded within the worker framework of AnalyzeGenomes. In Figure 3 we describe the pipeline, input and output of the individual steps and the order in which they are executed. The boxes represent applications, i.e., python wrappers around the incorporated bioinformatics tools, e.g., TopHat. Such programs could be extended and interchanged when new tools need to be introduced.

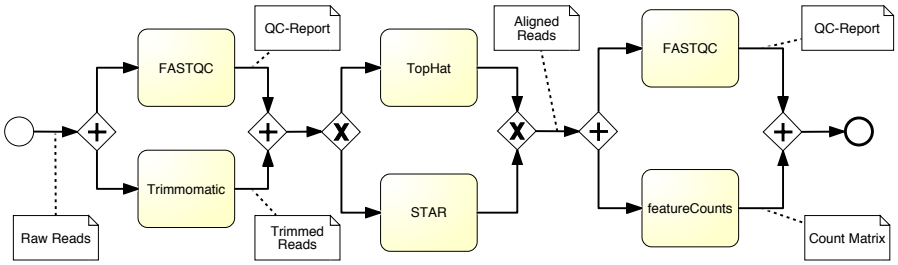

Fig. 3: Specific implementation of our RNAseq preprocessing pipeline

We identified the following tools to be suitable for our first prototype: FastQC [2] for quality control before and after trimming of reads with trimmomatic [4], Tophat [20] or STAR [8] for alignment of reads to the reference genome, and featureCounts [15] for creating count tables from alignment files. In this setup, all samples will be preprocessed only once to avoid redundancies.

DE Calculation and Design Formula Creation DE calculation as explained in Section 2 is done via DESeq2 [16] within our Rserve instance. DESeq2 is called from a stored procedure within our in-memory database and requires the raw count table as generated by our preprocessing pipeline. Furthermore, DESeq2 receives metadata on the selected patients, i.e., user selected features and the corresponding design formula, e.g., gender + age + gender:age. We reduced the number of possible designs to be a two-factorial, two-level design with an interaction term to allow for sufficiently large study groups in small data sets. `Factors` in the clinical data are of differing statistical types, i.e., they consist of numerical data, e.g., age, binary data, e.g., gender, or categorical data, e.g., race, as given in Table 1. Furthermore, categorial data can be differentiated to be exclusive, i.e., a patient can only be described by one category (blood group), or non-exclusive, i.e., a patient may be assigned to more than one instance of a category (e.g., different medications). The type of data defines how it is handled within in the experimental design procedure.

In this setting, any given `factor` needs to to be split into two `levels` as specified by the user and therefore will be reduced to a binary representation (Table 1). While `levels` are natural in the case of binary data, the `levels` of numerical and categorial data need user input. In the case of numerical data the user defines a split point x which divides the values into two groups. For exclusive categorial data, the user chooses at least one instance of the `factor` per `level` or can combine multiple instances into one level. Non-exclusive `factors` need one binary representation per instance. Thus, e.g., the instance "Beta-blocker" of the `factor` "Medications" is split into being present or absent (yes/no). A second instance of the `factor` can be used to create a second factor. `Factors`

Table 1: Description of statistical data types, `factors` and their corresponding binary representation (`levels`).

Data type	**Factor** **Full range example**	**Binary level** **Example**
Binary	Gender Male/Female	Male = all male patients Female = all female patients
Numerical	Age 0-90 years	Below_x = [0 - x) AboveAnd_x = [x - 90]
Categorial exclusive	Blood group A, B, AB, 0	Blood_1 = A, B, AB Blood_2 = 0
Categorial non-exclusive	Medication Beta-blocker, Aspirin, Thyroxin	Med_yes = Aspirin yes Med_no = Aspirin no

and `levels` are subsequently translated into the design formula as expected by DESeq2.

Interactive Visualization and Annotation Many results and intermediate results are of interest for both the clinician and computational biologist. Quality control as done by FASTQC produces an html-file for every sample which is stored and accessed for display within the application. Additionally, results from DESeq2, i.e. the list of DE genes, their test statistics and also the complete normalized and transformed count matrix, are visualized within the application. Interactive heatmaps are implemented via the clustergrammer software and its biology-specific extensions to show gene/protein names, cluster statistics and GSEA [1]. Further plots are implemented in custom D3.

5.3 Application Layer

Our application consists of three parts: (i) the experimental design panel, (ii) a visualization panel and (iii) a knowledge panel.

Experimental Design Panel The experimental design panel is the main part of the application as it enables to dynamically choose interesting clinical patient data categories to be studied in DE analysis (Figure 4). The overall goal is to split the patient population into at least two subgroups based on the patients' characteristics. For demonstration purposes, we use data from the SMART study. Patients are characterized by approx. 200 clinical variables (e.g. gender, height, blood pressure) that are grouped in categories (e.g. demographics or ECG measurements). All categories are displayed and may be expanded to show the variables. Binary variables and non-exclusive categorical data can be dragged into the design matrix directly. Continuous variables are split by the user via an interactive slider over the full range of possible values. Exclusive categorical variables may be combined within one column of the design matrix via drag-and-drop.

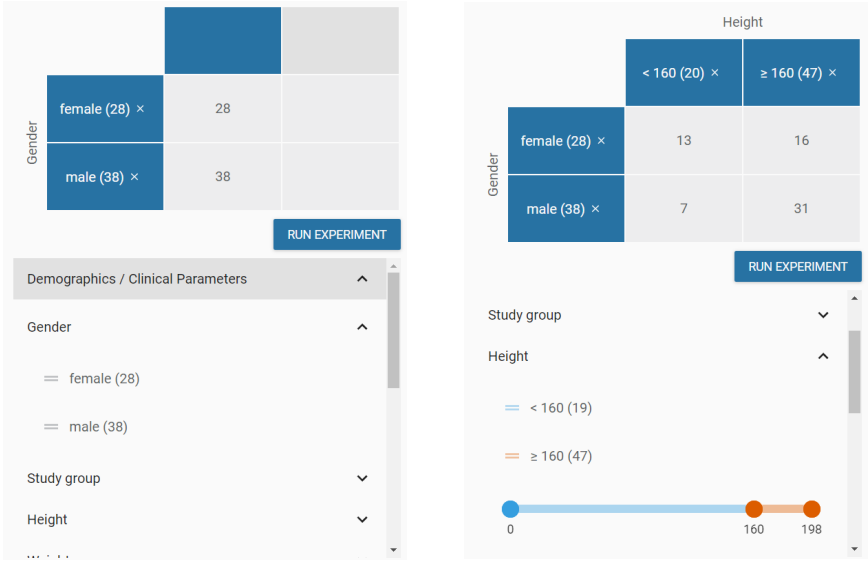

Fig. 4: Screenshot of experimental design panel.

The design matrix displays `factors` and `levels`, calculates group sizes and provides an estimation on the achieved statistical power at user interaction. After the creation of a valid design, i.e. at least three samples in every group, DeSEQ2 is triggered with the corresponding count tables generated in the preprocessing step via the 'Run Experiment'-button.

Visualization Panel Within the visualization panel the user may choose between four different tabs to choose plots on quality control of the preprocessing from FASTQC, overall characteristics of the data (e.g. PCA, sample-to-sample heatmap) and DE results (e.g. interactive heatmap, volcano plot, list of Genes/proteins and their statistics). A mouse over on genes/proteins displays a short description and clusters are clickable for statistics and for an update of the literature search. All results can be downloaded into a publication-ready report.

Knowledge Panel Especially the clinician needs additional external information on analysis results. Instead of querying for mere names of, e.g., a gene, the found relationship, e.g., effects of upregulation of a gene or the disease context, are included in the query to find actionable insights. Examples for external resources that can be leveraged are search engines such as Olelo [13] for intelligent PubMed queries or DisGeNet [19] for gene-disease associations.

6 Evaluation and Discussion

Our DEAME application is designed for users with limited to no bioinformatics knowledge while using state-of-the-art tools to meet scientific needs for accuracy (**R 4**, **R 9**). It allows easy configuration of design parameters based on the actual clinical patient information (**R 5**). Bioinformatics processing of raw RNA reads is completed automatically in the background to yield count matrices (**R 1**, **R 2**). The split of the pipeline (**R 3**) does not necessarily reduce the time to test a single hypothesis, but it avoids redundant preprocessing and thus eliminates computational overhead as soon as multiple hypotheses are tested. We bridge the gap between DE calculations and their clinical interpretation by the experimental design panel. Static design formulation as used in related work is exchanged by a more flexible handling that allows for ad-hoc adaptions (**R 6**). The interactive plots do not require additional experience or tools and display information on the found genes and proteins (**R 7**). Additionally, our knowledge panel shows literature on the found genes/proteins and includes the context of the analysis to provide actionable insights (**R 8**).

Within most of the used tools there are many options to fine-tune the analysis. We purposely do not use many of these options as they most certainly will confuse the clinician as a user. We expect the results set of regulated genes or proteins to be smaller than within a fine-tuned environment. While this is a drawback in a detailed analysis of a computational biologist, the clinicians we spoke to are interested primarily in the strong signals and are pleased with a shorter list of candidate genes/proteins. If a specific hypothesis turns out to be worth more research, the computational biologist may take over or a follow-study can be set up. Our application provides a platform for communication in DE results between the research-oriented clinician and the computational biologist. The concept of DEAME is aimed for use in observational studies, e.g. in the systems medicine context, were study design lacks a strong intervention or treatment factor to test in differential expression analysis.

7 Conclusions and Future Work

For the first time, requirements of a clinicians were included and matched with those of computational biologists in the design of an RNAseq and DE calculation platform. As a result, we planned and implemented a research prototype of an application that i) automates the complete preprocessing of RNA sequencing data in a way that enables rapid hypothesis testing, ii) can be run by a clinician and iii) helps interpreting the data. Our first working prototype will be validated in terms of specificity of the results set and the usability of the application within the SMART systems medicine consortium.

In addition to the RNAseq data, we have also started to use our framework to analyze DE proteins as calculated from shot gun proteomics. We will also extend possible design formulas to enable more complex experimental designs. Currently, our application is only usable within the SMART project, but as

soon as the data is published, we plan to provide free of cost access to the web application. Users will then be able to browse the rich SMART data or to create own projects to explore.

8 Acknowledgement

Parts of this work were generously supported by a grant of the German Federal Ministry of Education and Research (031A427B).

References

1. Clustergrammer's Documentation. http://clustergrammer.readthedocs.io/index.html
2. FASTQC Documentation. http://www.bioinformatics.bbsrc.ac.uk/projects/fastqc
3. Afgan, E., et al.: The Galaxy platform for accessible, reproducible and collaborative biomedical analyses: 2016 update. Nucl Acids Res 44 (2016)
4. Bolger, A.M., Lohse, M., Usadel, B.: Trimmomatic: a flexible trimmer for Illumina sequence data. Bioinformatics p. btu170 (2014)
5. Byron, S.A., et al.: Translating RNA sequencing into clinical diagnostics: opportunities and challenges. Nature Reviews Genetics (2016)
6. Conesa, A., et al.: A survey of best practices for RNA-seq data analysis. Genome biology 17(1), 13 (2016)
7. D'Antonio, M., et al.: RAP: RNA-Seq Analysis Pipeline, a new cloud-based NGS web application. BMC genomics 16(6), S3 (2015)
8. Dobin, A., et al.: STAR: ultrafast universal RNA-seq aligner. Bioinformatics 29(1), 15–21 (2013)
9. Gaur, P., Chaturvedi, A.: A Survey of Bioinformatics-Based Tools in RNA-Sequencing (RNA-Seq) Data Analysis. In: Translational Bioinformatics and Its Application, pp. 223–248. Springer (2017)
10. Gietzelt, M., et al.: The Use of Tools, Modelling Methods, Data Types, and Endpoints in Systems Medicine: A Survey on Projects of the German e: Med-Programme. Studies in health technology and informatics 228, 670–674 (2016)
11. Han, H., Jiang, X.: Disease biomarker query from RNA-seq data. Cancer informatics (Suppl. 1), 81 (2014)
12. Kraus, M., Schapranow, M.P.: An In-Memory Database Platform for Systems Medicine. In: Proceedings of the 9th Int'l Conf. on Bioinformatics and Computational Biology. ISCA (2017)
13. Kraus, M., et al.: Olelo: a web application for intuitive exploration of biomedical literature. Nucleic Acids Research 45(W1), W478–W483 (2017)
14. Li, J., et al.: From gigabyte to kilobyte: a bioinformatics protocol for mining large RNA-Seq transcriptomics data. PloS one 10(4), e0125000 (2015)
15. Liao, Y., Smyth, G.K., Shi, W.: FeatureCounts: An Efficient General Purpose Program for Assigning Sequence Reads to Genomic Features. Bioinformatics 30(7), 923–930 (2014)
16. Love, M., Anders, S., Huber, W.: Differential analysis of count data–the DESeq2 package. Genome Biology 15, 550 (2014)
17. Love, M.I., Anders, S., Kim, V., Huber, W.: RNA-Seq workflow: gene-level exploratory analysis and differential expression. F1000Research 4 (2015)

18. Plattner, H., Schapranow, M.P. (eds.): High-Performance In-Memory Genome Data Analysis: How In-Memory Database Technology Accelerates Personalized Medicine. Springer-Verlag (2014)
19. Queralt-Rosinach, N., Piero, J., Bravo, A., Sanz, F., Furlong, L.: DisGeNET-RDF: harnessing the innovative power of the Semantic Web to explore the genetic basis of diseases. Bioinformatics 32(14), 2236–2238 (2016)
20. Trapnell, C., Pachter, L., Salzberg, S.L.: TopHat: Discovering Splice Junctions with RNA-Seq. Bioinformatics 25(9), 1105–1111 (2009)
21. Wagle, P., Nikolić, M., Frommolt, P.: QuickNGS elevates Next-Generation Sequencing data analysis to a new level of automation. BMC genomics 16(1), 487 (2015)
22. Wolfien, M., et al.: TRAPLINE: a standardized and automated pipeline for RNA sequencing data analysis, evaluation and annotation. BMC bioinformatics 17(1), 21 (2016)