TAMPERE UNIVERSITY OF TECHNOLOGY

Author(s) Aho, Kaisa-Leena; Kerkelä, Erja; Yli-Harja, Olli; Roos, Christophe

Title Construction of a computational data analysis pipeline using a workflow system

Citation Aho, Kaisa-Leena; Kerkelä, Erja; Yli-Harja, Olli; Roos, Christophe 2010. Construction of a computational data analysis pipeline using a workflow system. In: Nykter, Matti; Ruusuvuori, Pekka; Carlberg, Carsten; Yli-Harja, Olli (ed.) . Proceedings of the Seventh International Workshop on Computational Systems Biology, WCSB 2010, Luxembourg, June 16-18, 2010. TICSP Series vol. 51, 4 p.

Year 2010

DOI -

Version Publisher's version

URN http://URN.fi/URN:NBN:fi:tty-201308131280

Copyright Creative Commons License, http://creativecommons.org/licenses/by/2.0/

All material supplied via TUT DPub is protected by copyright and other intellectual property rights, and duplication or sale of all or part of any of the repository collections is not permitted, except that material may be duplicated by you for your research use or educational purposes in electronic or print form. You must obtain permission for any other use. Electronic or print copies may not be offered, whether for sale or otherwise to anyone who is not an authorized user.

In Proceedings of the Seventh International Workshop on Computational Systems Biology, WCSB, 2010, Nykter, M., Ruusuvuori, P., Carlberg, C., Yli-Harja, O. eds., pp. 11-14, Luxembourg
ISBN 978-952-15-2384-7 / ISSN 1456-2774

CONSTRUCTION OF A COMPUTATIONAL DATA ANALYSIS PIPELINE USING A WORKFLOW SYSTEM

Kaisa-Leena Aho[1], Erja Kerkelä[2], Olli Yli-Harja[1] and Christophe Roos[1]

[1]Tampere University of Technology, Department of Signal Processing,
Korkeakoulunkatu 10, 33720 Tampere, Finland,
[2]University of Tampere and Tampere Univ. Hospital, Regea - Institute for Regenerative Medicine,
Biokatu 12, 33520 Tampere, Finland,
kaisa-leena.aho@tut.fi, erja.kerkela@regea.fi, olli.yli-harja@tut.fi, christophe.roos@tut.fi

ABSTRACT

We present how we used a workflow system to create a computational pipeline for storing, preprocessing and statistically analyzing gene expression microarray data together with public annotation information. The pipeline demonstrates that the used workflow system can facilitate reuse and documentation of the analysis methods which are adaptable to changing integrative data analysis approaches.

1. INTRODUCTION

Computational analysis of biological large-scale data sets often requires repeating similar analysis steps. One reason to this is that measurements are often performed more than once for a given experimental setup. It may also be necessary to perform similar analyses to the same dataset in order to see how choosing different parameters affects the results. Another reason is that measurement data of a given measurement system usually needs to go through certain analysis steps. Thereby, using continuously the same measurement system brings stability to data analysis. As long as the experimental setup remains the same, repeating the analysis process could happen on the side of wet lab researcher if the process can be enacted in a simpler way than running scripts from a command line console. On the other hand, as research progresses, the biological question and the experimental setup change and computational experts need to develop new data analysis approaches. When the measurement instruments remain the same, certain parts of the analysis can remain the same, while other parts need to be further developed. Against this background and as analysis tasks are shifting between the wet lab experimentalist and the computational specialist, there is a need for facilitating the reuse of parts of the analysis flow.

To respond to this demand, many commercial and public workflow systems have been developed [1]. They enable construction of workflows, pipelines of computational tools. For example, Taverna is an open-source platform for creating and executing computational workflows [2,3]. It enables the use and integration of separate distributed web-services, including computational tools and databases, through one interface where workflows can be constructed out of these services. Taverna is not limited to only biological analysis. It enables the use of a variety of tools for different fields. GenePattern is another platform that provides analysis tools for genomic data [4]. These tools can be used in workflow construction.

In this work, we tested the Medicel Integrator (Medicel Oy, Espoo, Finland) in workflow construction. It differs from the earlier mentioned systems in that it also includes a local data warehouse which regroups data from several web databases in one location. It enables storage of experimental and public data as well as experimental descriptions in a formal way which facilitates the computational use of the information in data analyses.

We implemented a commonly used microarray data analysis pipeline to evaluate whether such a workflow system can facilitate reuse of analysis methods. We created a modular computational workflow for storing, preprocessing and statistically analyzing Affymetrix gene expression data together with public annotation information retrieved from the data warehouse. More specifically, the implemented data analysis methodology performs normalization, a statistical test to find differentially expressed genes between two different sample groups and a statistical enrichment test to find gene ontologies that are significantly enriched or depleted among found differentially expressed genes.

The workflow can be reused for repetitive analysis of a given data set when one wishes to test different parameters in statistical tests. Alternatively, it can be reused for another data set by only replacing the data to be analyzed. For a user with programming background, it is also adaptable to changing analysis needs, supporting the reuse of established analysis methods. The used input data and parameters as well as the obtained intermediary and end results are saved within the workflow as documentation.

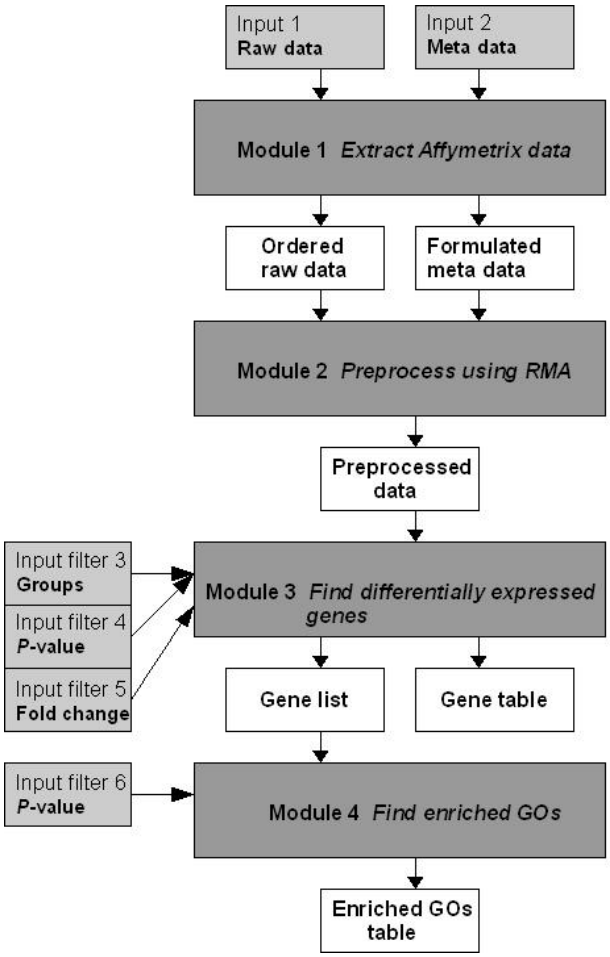

Figure1. Block diagram of the modular workflow scheme, where the user can do the most common modifications to inputs and run the workflow or its modules. The white boxes represent intermediary or end results which can be visualized and exported. Input 1 represents Affymetrix files. Input 2 is a table which includes information of each sample. Input 3 defines the sample groups. Inputs 4-6 define thresholds.

2. RESULTS

2.1. Workflow system

We used the Medicel Integrator workflow system for workflow construction. It enables construction, enaction, storage and reuse of data analysis procedures organized as tool pipelines. The platform provides access to the R programming environment and Bioconductor (www.bioconductor.org) [5] as well as a variety of IT and computational data analysis tools that can be used as building blocks in workflow construction. Its associated data warehouse integrates data from public databases, such as Ensembl, Uniprot, Intact, Reactome and Kyoto Encyclopedia of Genes and Genomes (KEGG). .

2.2. Modularity and reuse of a workflow

For simplicity, we mainly describe our workflow on the level of modules. We use the term module to refer to a set of interconnected computational tools within a workflow. Such modules have each a high-level functionality, such as reading in data, performing data preprocessing or performing analytical operations (Figure 1). The computational tools forming sub-pipelines (not shown) within each module have lower-level functionalities, such as converting data from one data format to another, performing a statistical test, or sorting rows within a table.

The modular structure of the workflow supports reuse of the analysis methods. The user can visualize the whole workflow in a modular scheme (Figure 1), where the relevant inputs, parameters and outputs of each module are exposed. From there, it is possible to execute the whole workflow or only chosen modules. It is also possible to visualize the tools within a module and execute them one at a time. The inputs of each module include data to be analyzed and parameters related to the functionality of individual tools within the modules. The outputs of the modules are intermediary and end results of the pipeline. They can be visualized and exported. Prior to reuse of the workflow, the data analyzed can be replaced and the parameters can be modified by the user. Each workflow, module, or tool can also be copied as en entity within the workflow system and pasted into another workflow for reuse as well as modified for other analytical needs.

The modular structure of the workflow also supports documentation of the analytical procedure, since the inputs, parameters and outputs of each module, i.e. intermediary and end results are saved.

2.3. Workflow description

The established workflow includes processes that are typically covered in the analysis of DNA microarray data: 1) loading and reorganizing data based on sample information, 2) data preprocessing, 3) identification of singular gene expression patterns, and 4) mapping of the observed singularities to biological context data and identification of significantly enriched or depleted annotations. An analogous analysis has been performed to study differences in stem cells between two different culture conditions [6].

2.3.1. Load and reorganize data and sample information

The first module of the workflow takes in the raw Affymetrix data files (.CEL) and a metadata table including the file names together with sample information, such as cell, tissue, organ, or treatment type. The module reorganizes the raw data files and produces a metadata file which can be read in the following analysis steps. The raw data files and the metadata table can be modified between module executions.

2.3.2. Preprocess the data

The second module of the workflow performs measurement summing and normalization. Single probe measurements in the raw Affymetrix data files given as input are converted to gene-wise values for transcript abundance using the robust multiarray average (RMA) algorithm of R [7,8]. In this example workflow we used a custom CDF file *hgu133plus2hsensgcdf* (loaded at http://brainarray.mbni.med.umich.edu) which defines how the probes are pooled to values of Ensembl genes [9]. The description of the different samples is given to this module as a second input. The output of the module is a table which includes one RMA normalized expression value for each Ensembl gene across all samples. The column headers include information relative to sample descriptions that can be used in later analysis steps when filtering the data.

2.3.3. Identify singular gene expression patterns

The third module of the workflow compares the expression levels of genes between two sample groups using the Linear Models for Microarray Data (limma) algorithm in R [10]. The statistical test is performed as defined for comparison of two groups of samples [11]. The module takes the normalized data matrix from the previous module as input. Experimental parameters available to define the sample groups to be compared, as well as the thresholds for adjusted *p*-value and fold change, are given within an R script as another input. The output is a table that includes the names of the Ensembl genes that are differentially expressed between the sample groups. For each gene, the statistics for differential expression from the limma test are presented, including the average $\log_2$ fold change between the sample groups and a *p*-value adjusted for multiple testing (using Benjamini and Hochberg's method to control the false discovery rate) [11]. Another output includes only a list of names of the differentially expressed Ensembl genes. This list serves as input for the subsequent module.

2.3.4. Identify significantly enriched and depleted annotations to GO terms

The fourth module finds GO categories that are over- or underrepresented among the group of differentially expressed genes with respect to all the genes measured on the chip. The core of the module implements a hypergeometric test. This module uses the list of Ensembl gene names as input. A threshold *p*-value for significant enrichment according to the hypergeometric test is given as another input. The output of the module is a table including a ranked list of significantly over- and underrepresented GO categories together with a set of key values, such as their *p*-values from both statistical tests and a value indicating whether the category is over- or underrepresented.

2.3.5. Demonstrating workflow reusability

We used two separate data sets to test the workflow. The first data set of 10 arrays was on adipose stem cells cultured in two different culture media [6]. The second data set of 70 arrays was on human embryonic stem cells and induced pluripotent stem cells. The latter data was a collection of public data sets retrieved from a stem cell database (Kong et al., manuscript in preparation).

For the first data set, we performed an analogous analysis as presented in [6]. The raw data from the 10 arrays were loaded into input 1. The metadata table in input 2 contained the raw data file names and for each of the files either the classification term HS (human serum) or FBS (fetal bovine serum). Two combinations of inputs 3-6 were tested (Table 1). In one case the input 5 was eliminated to allow fold changes of all sizes.

To analyze the second data set, data in input 1 were replaced by the raw data from the 70 arrays. Input 2 was modified to include the raw data file names and their classification into hIPS (human induced pluripotent stem cell) or hESC (human embryonic stem cell). The used inputs 3-6 are presented in Table 1.

Table 1. Inputs used to demonstrate reusability.

Input 1	**Input 3**	**Input 4**	**Input 5**	**Input 6**
10 arrays	HS FBS	< 0.05	> 2	< 0.05
	HS FBS	< 0.05	eliminated	< 0.05
70 arrays	hIPS hESC	< 0.01	> 1.5	< 0.05

3. DISCUSSION

We used a workflow system to construct a workflow for DNA microarray data analysis. We showed that using such a system can help structure the analysis processes for reuse and documentation purposes. Within the workflow, the analytical processes are organized into a modular cascade of computational tools, which can be executed either as a whole or separately. The whole analysis process is thus automated and explicit and can be easily tuned and reused as whole or in parts for similar tasks when new experimental data is obtained or to analyze the same data using different parameters. Only the input data and chosen parameters used in the analysis need to be replaced by the user. After executing the workflow, all the intermediary and end results of the analysis cascade are visible and exportable. The workflow is saved and provides documentation of how results have been obtained.

The re-usability value of the pipeline depends on the user's background. In the simplest case, for a non-programmer, it is replacing input parameters as in the cases we demonstrated with two data sets, or as in a work where the enrichment analysis module was modified for the needs of another analysis on yeast genes [12]. This required replacing one input of human genes within the module with a list of yeast genes. For a user with programming background, it is also possible to

modify the operation of the workflow e.g. by modifying the computational tools.

The possibility provided by the used workflow system to visualize the whole analysis process as a pipeline including the inputs, analysis tools, and outputs, is an advantage, because what happens to the data is explicit to the user. This is not the case in many microarray data analysis software, such as GeneSpring (Agilent Technologies).

The used workflow system differs from e.g. Taverna and GenePattern in that it includes a local data warehouse to which information can be stored and from which other biological data can be retrieved and integrated with the data being analyzed. Within our workflow, this characteristic was exploited in the fourth module which takes a list of Ensembl genes and retrieves their annotations to GO terms from the data warehouse to perform an enrichment analysis. An analogous enrichment analysis could be made with a few modifications for any type of category into which the genes have been classified in the data warehouse. Enrichment could for example be calculated with respect to involvement in signaling or metabolic pathways, but also other things, such as genomic location of the genes, or subcellular location or domain features of the corresponding proteins.

An essentially similar data analysis as covered by our workflow could be performed entirely in the R programming environment [7,8,10,11,13]. Many microarray data analysis methods have been implemented in R. However, using R requires programming experience and modularization of the analysis happens only within the script. Modularization raises the efficiency of re-use. It is important to make it accessible, which is not the case when it is embedded within a script, in particular for a user not familiar with R.

Chipster is an example of software developed for facilitating the use of R commands through a graphical user interface (http://chipster.sourceforge.net, The Finnish IT Center for Science). It is an open-source platform that provides computational tools for workflow construction primarily for microarray data analysis. In functionality, the tools correspond to the modules presented in this work, since they perform preprocessing and analytical tasks to microarray data. These tools can be chained and their parameters can be modified by the user. Overall, it provides a variety of ready-made analytical tools for the tasks that in Integrator were covered by a pipeline of tools of lower-level functionalities within each analytical module. In Integrator, the user was expected to build these modules before they could be used. On the other hand, Chipster provides analysis tools for a selected set of data types, whereas Integrator has broader applicability.

Executing the workflow within the tested system is in general slower than running a corresponding series of commands in R, because within the workflow, there are other tools around those that run the R codes. The time required for constructing or using a workflow or R script depends, among other, on the user's background. We consider that constructing a workflow of the presented type was rather time-consuming, but so can be writing an R script if we take into account the time needed to familiarize with the R packages. Once the workflow has been established, it is simple and efficient to enact it with new data or parameters. Similarly, one could rapidly modify an R script, but this requires programming skills. It is also easier to share a modular workflow than to share a script, because the workflow is more communicative with respect to the operations being performed. Therefore, we conclude that the workflow system tested can function as a platform for constructing modular easy-to-use data analysis methods intended for users without programming experience.

4. ACKNOWLEDGMENTS

The work was supported by Tampere Graduate School in Information Science and Engineering, the Academy of Finland (application no. 129657, Finnish Programme for Centres of Excellence in Research 2006-2011), and EU-FP6 project ESTOOLS. We thank Jukka Matilainen and Daniel Nicorici (Medicel Oy) for help.

5. REFERENCES

[1] A. Tiwari and A.K.T. Sekhar, "Workflow based framework for life-science informatics," Computational Biology and Chemistry, vol. 31, pp. 305-319, 2007.

[2] D. Hull, K. Wolstencroft, R. Stevens, et al., "Taverna: a tool for building and running workflows of services," *Nucleic Acids Research*, vol. 34, pp. W729-W732, 2006.

[3] T. Oinn, M. Greenwood, M. Addis, et al., "Taverna: lessons in creating a workflow environment for the life sciences," *Concurrency and Computation: Practice and Experience*, vol. 18, pp. 1067-1100, 2006.

[4] M. Reich, T. Liefeld, J. Gould, et al., "GenePattern2.0," *Nature Genetics, vol.* 38 no. 5, pp. 500-501, 2006.

[5] R.C. Gentleman, V.J.Carey, D.M. Bates, et al., "Bioconductor: open software development for computational biology and bioinformatics," *Genome Biology*, vol. 5, R80, Sept. 2004.

[6] B. Lindroos, K.-L. Aho, H. Kuokkanen, et al., "Differential gene expression in adipose stem cells cultured in allogeneic human serum versus fetal bovine serum," *Tissue Engineering Part A*, published electronically Apr 13th, 2010.

[7] R.A. Irizarry, B.M. Bolstad, F. Collin, et al., "Summaries of Affymetrix GeneChip probe level data," *Nucleic Acids Research*, vol. 31, e15, 2003.

[8] R.A. Irizarry, B. Hobbs, F. Collin, et al., "Exploration, normalization, and summaries of high density oligonucleotide array probe level data," *Biostatistics*, vol. 4, pp. 249-264, 2003.

[9] M. Dai, P. Wang, A.D. Boyd, et al., "Evolving gene/transcript definitions significantly alter the interpretation of GeneChip data," *Nucleic Acids Research*, vol. 33, e175, 2005.

[10] G.K. Smyth, "Linear models and empirical Bayes methods for assessing differential expression in microarray experiments," *Statistical Applications in Genetics and Molecular Biology*, vol. 3, Article3, 2004.

[11] G.K. Smyth, M. Ritchie, N. Thorne, et al., "limma: linear models for microarray data user's guide," Bioconductor package help documentation, Melbourne, 2007.

[12] T. Aho, H. Almusa, Matilainen J., et al., "Reconstruction and validation of RefRec: a global model for the yeast molecular interaction network", accepted in *PLOS ONE*.

[13] S. Falcon and R. Gentleman, "Using GOstats to test gene lists for GO term association," *Bioinformatics*, vol. 23, pp. 257-258, 2007.