# Workflow for Rapid Metagenome Analysis

Gunnar Schulze

Potsdam University, Potsdam, D-14482, Germany
gschulze@uni-potsdam.de

**Abstract.** Analyses of metagenomes in life sciences present new opportunities as well as challenges to the scientific community and call for advanced computational methods and workflows. The large amount of data collected from samples via next-generation sequencing (NGS) technologies render manual approaches to sequence comparison and annotation unsuitable. Rather, fast and efficient computational pipelines are needed to provide comprehensive statistics and summaries and enable the researcher to choose appropriate tools for more specific analyses. The workflow presented here builds upon previous pipelines designed for automated clustering and annotation of raw sequence reads obtained from next-generation sequencing technologies such as 454 and Illumina. Employing specialized algorithms, the sequence reads are processed at three different levels. First, raw reads are clustered at high similarity cutoff to yield clusters which can be exported as multifasta files for further analyses. Independently, open reading frames (ORFs) are predicted from raw reads and clustered at two strictness levels to yield sets of non-redundant sequences and ORF families. Furthermore, single ORFs are annotated by performing searches against the Pfam database.

**Keywords:** bioinformatics, metagenome,cd-Hit-algorithm, clustering, protein family, annotation.

## 1 Introduction: Workflow Scenario

Metagenomics in life sciences provides insights into whole ecosystems and has facilitated the understanding of biological processes, organismal interactions and genetics of various biomes throughout the world. The scientific progress in this field has been significantly enhanced by the advent of next-generation sequencing technologies which provide researchers with ever-increasing amounts of sequencing data. However, to make full use of these opportunities, new approaches of (sequence) data analysis have to be employed. Typical metagenomic datasets consist of large collections of raw sequence reads as outputted by NGS-technologies like 454-sequencing or Illumina, organized in multiple sequence files in FASTA format and come along with associated metadata (e.g. sampling location, environmental conditions, DNA isolation protocols and the type of sequencing technology used). The specific characteristics of this type of sequence data (short sequence lengths and large amounts of sequences) rendered it incompatible to prior approaches which where suitable for datasets obtained by

A.-L. Lamprecht et al. (Eds.): Process Design for Natural Scientists, CCIS 500, pp. 88–100, 2014.
DOI: 10.1007/978-3-662-45006-2_7 © Springer-Verlag Berlin, Heidelberg 2014

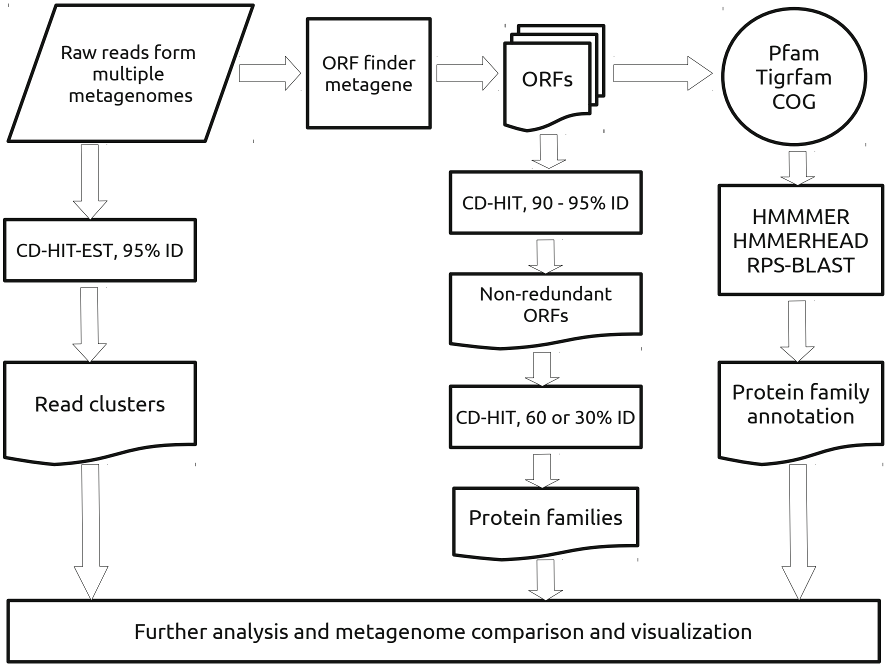
Raw reads form multiple metagenomes
ORF finder metagene
ORFs
Pfam Tigrfam COG
CD-HIT-EST, 95% ID
Read clusters
CD-HIT, 90 - 95% ID
Non-redundant ORFs
CD-HIT, 60 or 30% ID
Protein families
HMMMER HMMERHEAD RPS-BLAST
Protein family annotation
Further analysis and metagenome comparison and visualization

**Fig. 1.** Overview of the RAMMCAP (rapid analysis of multiple metagenomes with a clustering annotation pipeline) workflow following [12]

Sanger sequencing. Apart from specialized algorithms and data structures for sequence alignment and genome assembly, especially for metagenome data the computational pipelines for functional analyses are of great importance since they allow for structural insights into the genetic composition of biomes and possible implications for ecology of particular habitats.

The workflow presented here offers both a first insight in such functionalities (if present) and a basis for further, more detailed analyses. It takes a single multiple sequence FASTA file as input and provides the user with a structured dataset comprising sequence clusters at different levels of similarity and functional annotation of a subset of sequences by Pfam. The general structure of the workflow is very similar to the one depicted in figure 1.

The actual workflow consists of three major parts which are largely independent and may thus be processed in parallel. Starting from a single FASTA file containing (possibly thousands of) raw reads, the input sequences are first clustered to obtain groups of highly similar sequences. Clustering of the raw reads is a powerful tool and simplifies downstream processing since it can be used to reduce redundancies in the dataset which is of course not possible by manual interpretation of the data if the size of the dataset is large. Furthermore, by selecting an appropriate similarity (ID%) threshold the user gains a first overview of the dataset and the overall similarity of the sequences. A set of representatives of sequence clusters can be obtained and used as input to further analyses and additionally, clusters of certain size can be exported as multiple sequence FASTA files themselves and analyzed separately. An important feature of the jABC workflow framework can be exploited here. The possibility of going one step backwards and repeating the clustering at different levels of similarity and observe different outcomes without having to repeat the entire pipeline is certainly a strength of this approach. The user gains the ability to interactively proceed the workflow and may decide to redo certain analyses after adjusting parameters and then go on with the results rather than having to wait until the entire workflow has finished. By providing a suitable graphical user interface the workflow management framework can turn this somewhat tedious task in an interactive process while other tasks may already be tackled in the background.

In the second part of the workflow, open reading frames (ORFs) are predicted from raw sequencing reads independently of the initial clustering. Since ORFs indicate the potential presence of a gene in the raw reads this step is fundamental to subsequent analyses of potential gene families and functions (annotation steps). The predicted ORFs are already a valuable result an can be stored e.g. to be merged with results from other datasets later on. ORFs are further clustered successively in two more steps to yield first a non-redundant (nr) set of the initial ORFs by choosing the clustering cutoff at high similarity. In the second step, these non-redundant ORFs are clustered at a conservative cutoff to obtain so called families of ORFs. Following this approach the outcome of the clustering implicitly attains a hierarchical structure. Both the non-redundant set and the family set of ORFs can be useful for further analyses. Although the parameters for clustering may depend on the way the sequence data was obtained (see [12]), the idea is again to give the user more flexibility here and allow for iterative clustering to decide on the most appropriate clustering scheme.

Finally, the third part of the workflow carries out the protein family annotation of predicted ORFs. The Pfam [22] database provides comprehensive libraries of protein families and can be used to potentially assign ORFs to proteins/domains which in turn yields insights into the genetic content of samples. This can be considered a first step into the characterization of the biome. Additionally, more specialized databases like Tigrfam and COG can be employed to address e.g. phylogenetic issues, too.

## 2 Service Analysis

A central task in the workflow is the clustering of sequences both at the DNA level (raw reads) as well as at the level of ORFs (amino acid sequences). Due to the large amount of input sequences, a fast and memory-efficient clustering algorithm should be used. The cd-hit-suite which was published by Li *et al.* [13] in 2006 provides the user with a collection of algorithms to perform such clustering both on the DNA and on the protein sequence level and offers some basic tools to extract clustering statistics and further process the sequence data according to clustering information. The cd-hit algorithms are for example available at the corresponding webserver [1] and also part of the full RAMMCAP workflow which is accessible by scripts from the WebMGA server [27] but can also be downloaded at `http://www.bioinformatics.org/downloads/index.php/cd-hit/`. The suite can be installed locally and executed using the ExecuteCommand-SIB in jABC.

Apart from clustering, two other essential services need to be included into the workflow. Firstly, an algorithm to detect open reading frames from raw input reads is preliminary to any protein-level analysis and functional annotation. There are various programs available to call ORFs from multifasta files and e.g the WebMGA server uses `orf_finder`, `metagene` and `fraggene_scan` as tools for this task. Alternatively, the `getorf` program as part of the european molecular biology open software suite `(EMBOSS)` is already available as a SIB and can be used. Second, the annotation of the detected ORFs regarding potential memberships in gene families can be performed by using different web services. In this case, only Pfam is used to annotate single ORFs, while in theory more specialized databases could be included (see conclusion section) to match the more specific demands of the user. Nevertheless, the annotation of ORFs by Pfam yields important first insights and provides the basis for further analyses. Some of the web services provided by Pfam are already available as SIBs and in this case, the `SequenceSearch-SIB` can be employed to search the Pfam database for domain hits. To create single sequences from a FASTA file containing multiple ORFs yet another tool has to be employed. The EMBOSS provides large collections of file editing tools, e.g. the `seqretsplit` program, which is used here to split the multi FASTA file as outputted by the getorf program into single FASTA files which are in turn used as input to the SequenceSearch SIB. Since the seqretsplit program is not yet included into the collection of EBI SIBs present in jABC, a local `EMBOSS` installation is required additionally to call the program (see Conclusions). The most recent version of the `EMBOSS` can be downloaded at sourceforge.net [2].

Apart from these preliminaries (local installations of `EMBOSS` and the cd-Hit-suite) no further configurations have to be set by the user. For installation details please refer to the documentation which is available in the download versions of cd-hit and `EMBOSS` respectively. Currently the workflow also requires some basic Linux command-line tools which should be easily replaceable by specialized SIBs in the future thus making the workflow independent of the operating system.

## 3 Workflow Realization

Following the general structure of the RAMMCAP workflow as described in the introduction, the SIBs and external services are employed at three different stages of the workflow. Although these stages can be largely parallelized, the Annotation and clustering of ORFs depend on the previous prediction step. The three stages (raw reads clustering, ORF prediction and clustering, Pfam annotation) are linked by two fork SIBs which indicate tasks that can be run

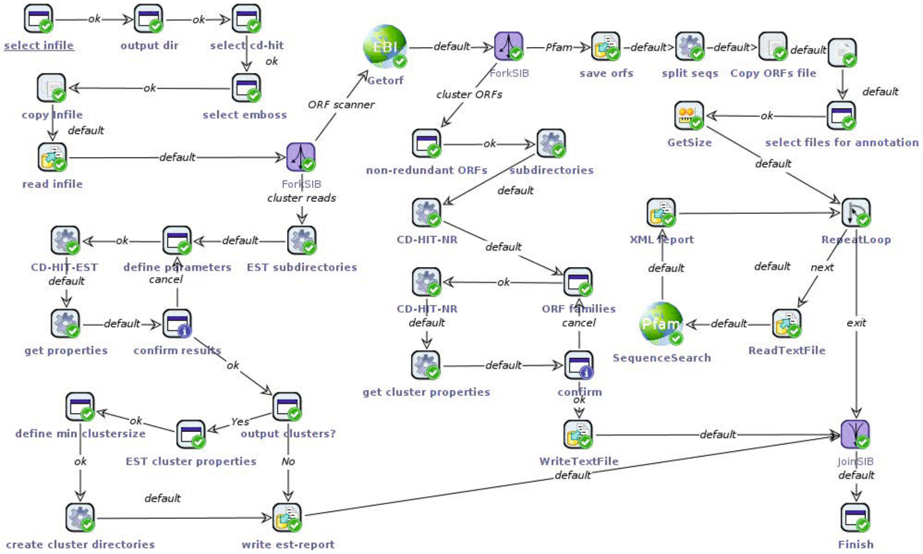

**Fig. 2.** Overview of the workflow with fork and join SIBs marked by triangles and inverted triangles, respectively.The different stages of the workflow and important intermediary steps are indicated by boxes and descriptions

in parallel 2. To make use of the specialized services and SIBS described in the previous section, additional `Common-SIBs` have to be employed and interactions with the user are required at some critical steps.

### 3.1 Basic Readin and Setup

At the beginning of the workflow the user is asked to set up some basic parameters, e.g. select an appropriate input file which should be a file in FASTA format containing multiple sequences. The `ShowFileChooser-SIB` allows for browsing the file system and selecting such files in a simple way. Analogously, the user is allowed to select or create an output directory which serves as working directory throughout the entire workflow. Two additional selections are necessary to specify the paths to cd-hit-suite and EMBOSS executables. Note that both suites keep all scripts in a single directory which allows for a single selection to gain access to all programs provided. After this general setup, the `CopyFile` SIB is used to copy the input file to the new working directory and rename it for easier access.

### 3.2 EST Clustering

After the basic readin the first `fork-SIB` is reached and the input is processed by the `getorf-SIB` to yield ORFs while raw reads are clustered via the `cd-hit-est`

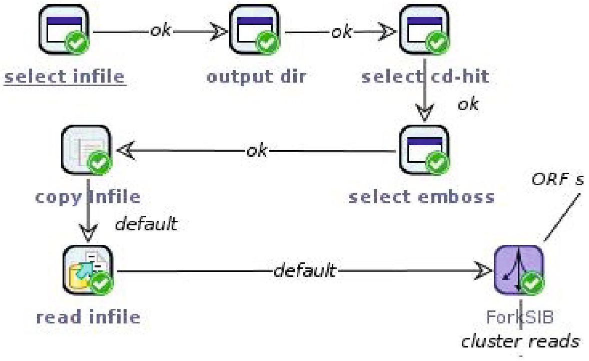

**Fig. 3. Basic readin:** Selection of the input data and general setups at the beginning of the workflow

algorithm 4. Before the actual clustering takes place, subdirectories for cd-hit-est output are created inside the intitial output directory by calling the Linux command-line tool `mkdir` via the `ExecuteCommand`-SIB. For the actual clustering, two important parameters which have to be specified by the user are a similarity cutoff (%ID) and the corresponding wordsize which is a parameter similar to the one used in `Blast` searches and in this case influences the running time and memory consumption of the `cd-hit-est` algorithm. The user can provide these two parameters by modifying initial values which represent snippets of the command-line parameters via the `ShowInputDialog-SIB`. In the next step the `ExecuteCommand-SIB` is again used to call the cd-hit-est program with the modified parameters and the output is written to the subdirectories created before. The cd-hit-suite provides scripts to extract additional information, e.g. an overview of the size of clusters and contained sequences. The `plot_len1.pl` script is called by another `ExecuteCommand-SIB` and outputs a tabular overview of the clustersizes which can be important for later analyses and is saved into a context variable. The `ShowConfirmDialog-SIB` allows the user to check if clustering was successful. If clustering was not successful or the outcome is not suitable for further analyses, the researcher can decide to refine the (possibly to strict) clustering parameters and repeat the analysis. Otherwise, the user might want to prepare some multifasta files as representatives of the larger clusters which can again be done by employing another tool from the cd-hit-suite. The user is first asked if such additional output is wanted and may then (based on the clustering information collected beforehand) decide on the minimal size of a cluster to be written into a multiple sequence FASTA file in an additional subdirectory. The SIBs required for this task are (in order) `ShowBranchingDialog`, `ShowTextDialog`, `ShowInputDialog` and `ExecuteCommand`. Finally, regardless of these choices, a report summarizing the clustering results is written into a simple text file in the EST-subdirectory.

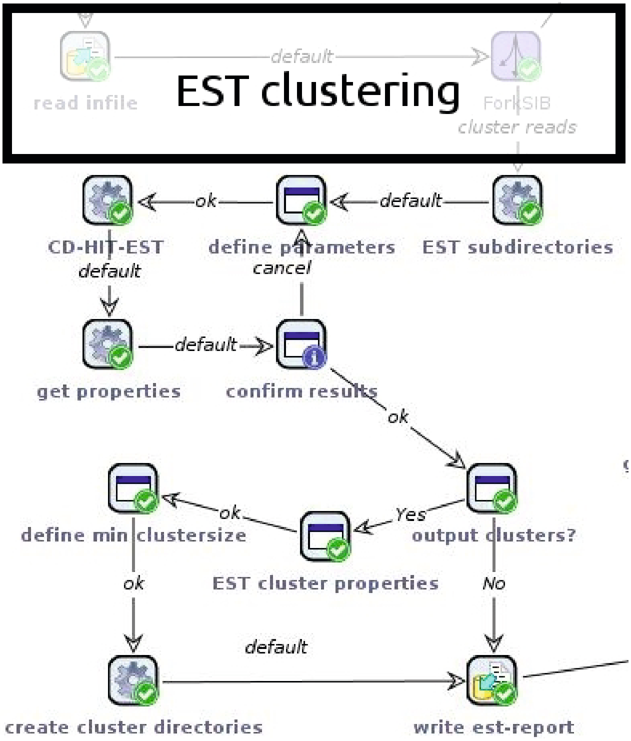

**Fig. 4.** The raw reads clustering stage of the workflow

### 3.3 ORF Prediction and Clustering

Following the other branch of the first `fork-SIB`, the raw reads (multiple sequences in a single FASTA file) are used as input to the `getorf` program to predict open reading frames. The output is a again a single FASTA file containing the predicted ORFs. Note that in this step nucleic acid sequences are converted into amino acid (protein) sequences automatically which is the also required for Pfam searches in the third stage of the workflow. Another `fork-SIB` is employed to yield two new threads for ORF clustering and Pfam annotation based on the `getorf` output.

The clustering of ORFs (see Fig. 6) is performed in two steps. First, the set of predicted ORFs is restricted to a set of non-redundant (nr) ORFs which are then further clustered into so called ORF families. To obtain a non-redundant set of ORFs the user is asked to specify a similarity threshold e.g. 95% ID at which ORFs will be considered to be redundant. After creating subdirectories for non-redundant and family sets of ORFs, this parameter is presented to the `cd-hit-algorithm` which is called by the `ExecuteCommand-SIB` analogously to the algorithm used for raw reads clustering. The resulting non-redundant set of ORFs serves as input to another clustering at lower similarity to yield family sets

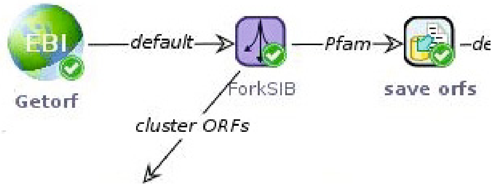

**Fig. 5.** Intermediate step: ORF prediction using the `getorf`-SIB

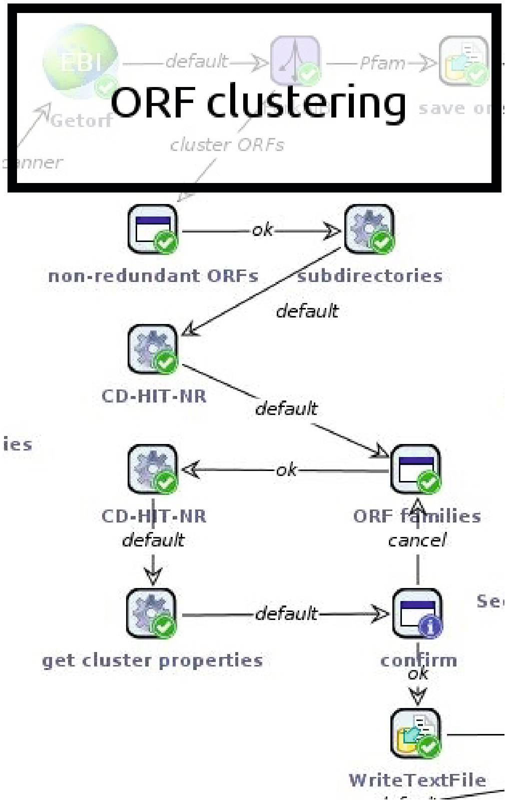

**Fig. 6.** ORF clustering using the cd-hit algorithm sequentially at two different similarity cutoffs

of ORFs. Here again the user is asked to enter appropriate cutoff and wordsize parameters and has the opportunity to redo the clustering to adjust parameters if necessary. Eventually, the ORF family clustering output is saved into a separate directory for further use.

### 3.4 Pfam Annotation

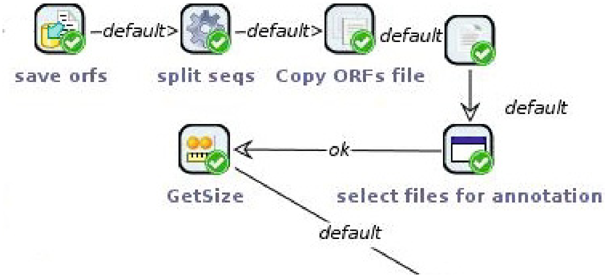

**Fig. 7.** Intermediary step: Dividing the Multi-FASTA file containing predicted ORFs into single sequence files via `seqretsplit`. Single sequence files can be used as input to `SequenceSearch`

The third stage of the workflow realizes the annotation process of ORFs which where predicted by `getorf`. First, the output of the `getorf` program which was saved in a context variable is now written into a text file in a separate directory called "singleseqs" which is used as working directory in the following. Since the `SequenceSearch-SIB` requires single sequences as input, the Multi-FASTA file has to be split into separate files containing single sequences (ORFs). This can be done efficiently by the `seqretsplit` script which is called by the `ExecuteCommand-SIB`. The initial Multi-FASTA file is moved from the "singleseqs" directory leaving only files containing single sequences which can be read in by the `SequenceSearch-SIB`. The user is now asked to select some of the single sequence files which should be queried via the `ShowFileChooser`-SIB and the size of the overall input to `SequenceSearch` is determined by the `getSize-SIB`. A repeat loop (`RepeatLoop-SIB`) is used to iterate over all query-sequences and to receive and write XML-reports into an annotation subdirectory. After finishing all annotation queries, the three threads representing the different stages of the workflow are joined and the successful completion of all tasks is stated.

## 4 Conclusion

The workflow presented here can be used to rapidly prepare raw sequence data for further analysis and allows the researcher to possibly gain first insights into

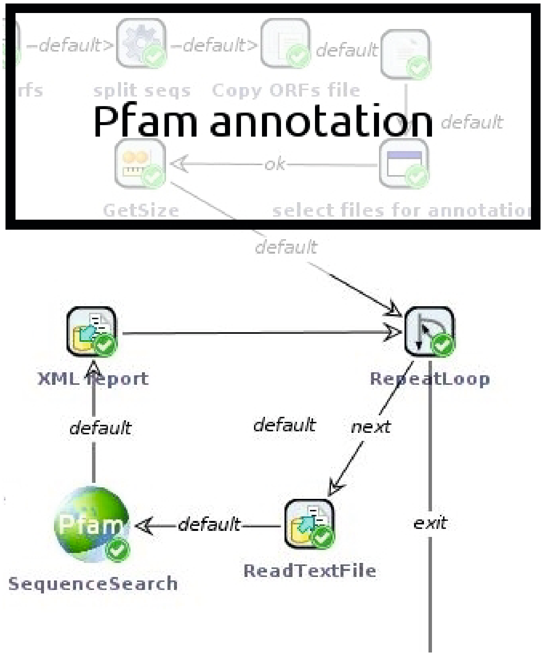

**Fig. 8.** Pfam annotation using the `SequenceSearch-SIB` and single query sequences as input

the contents of a particular sample. From a single Multi-FASTA file, clusters of sequences at different levels of similarity can be generated by a flexible procedure allowing for input-specific choice of optimal parameters. While the use of single web services and command-line tools may lead to tedious manual interconnection steps and all-in-one scripts lack flexibility, the jABC workflow framework allows for automation and flexibility when needed. Furthermore, independent tasks (threads) can be run in parallel which saves unnecessary waiting time. Although the workflow currently depends on some Linux specific command-line tools and some software has to be installed beforehand, these issues could be resolved in future versions, e.g. the `seqretsplit` script could be added to the existing collection of EBI-SIBS. Alternatively, the `jETI` toolserver could be employed to provide access to a large collection of EBI-SIBs without the need for single SIB implementations. This approach would also allow and simplify additional customization of the workflow if needed. The annotation using `SequenceSearch` currently only works for single sequences and requires the splitting of sequences and separate XML-reports are generated for every single ORF which is certainly overkill, since many ORFs might have no hit in the Pfam database. Additionally, representatives from large ORF family clusters could be extracted and queried against Pfam to restrict the search to sequences which are likely to be matched in the database.

Including the possibility of a Multi-FASTA input to SequenceSearch as is also possible at the Pfam web server might lessen these problems and also allow for a more comprehensive overview. Alternatively, a procedure to automatically extract node information from many XML files could provide the user with a single report which summarizes the results of the potentially hundreds of queries performed. Another improvement could include a setup interface at the beginning of the workflow which summarizes the various user inputs in a single request to restrict the user interactions to the potential adjustment of clustering parameters later on and thus speed up the overall analysis.

The workflow critically depends on the clustering algorithms taken from the cd-hit-suite which are highly specialized to process large inputs of short sequence reads. However, other clustering methods might be used alternatively and a generalization of the input-output structure could increase independence from one particular clustering approach which in turn offers additional opportunities for further analyses. One strength of the workflow is the extensibility at almost every stage. For example, the ORF annotation might be extended to include other, more specialized databases like Tigrfam and COG (as is done in the RAMMCAP workflow) while still allowing for customization to match the researchers preferences. In addition, the clustering could be extended and the user might define his own clustering protocol, e.g. following a hierarchical structure as is outlined in [12].

In its current state the workflow presented here can be considered both a preprocessing tool as well as a template for the rapid analysis of metagenomic data. The workflow might be extended itself or be included in other related workflows and future versions might include additional services which allow for a more specialized and customizable processing of the input data.

This article is part of a larger evaluation [6], which aimed at illustrating the power of simplicity-oriented development [18] by validating the claim that process modeling can indeed be handed over to the domain experts by providing them with a graphical modeling framework [25] that covers low-level details in a service-oriented fashion [20], integrates high-level modeling in the overall development process in a way that user-level models become directly executable [19,16], and supports ad-hoc adaptations and evolution [15,17].

The project described in this article can be characterized as follows:

- Scientific domain: bioinformatics
- Number of models: 1
- Number of hierarchy levels: 1
- Total number of SIBs: 40
- SIB libraries used (cf. [11]): common-sibs (38), ebi-sibs (1), pfam-sibs (1)
- Service technologies used: SOAP web services, REST web services

The bioinformatics part of this volume contains five other articles on workflow applications in this domain [23,3,14,24,26]. Further examples of workflow projects with he bioinformatics-specific incarnation of the jABC framework, called Bio-jETI [8], have been described, for example, in [7,9,4]. As shown in [8,10,5], bioinformatics is also a suitable field for the application of semantics-based (semi-) automatic workflow composition techniques (as provided by, e.g., [21]) to support the workflow design process.

## References

1. http://weizhong-lab.ucsd.edu/cd-hit/
2. ftp://emboss.open-bio.org/pub/EMBOSS/
3. Blaese, L.: Data Mining for Unidentified Protein Sequences. In: Lamprecht, A.-L., Margaria, T. (eds.) Process Design for Natural Scientists. CCIS, vol. 500, pp. 73–87. Springer, Heidelberg (2014)
4. Ebert, B.E., Lamprecht, A.-L., Steffen, B., Blank, L.M.: Flux-P: Automating Metabolic Flux Analysis. Metabolites 2(4), 872–890 (2012)
5. Lamprecht, A.-L.: User-Level Workflow Design. LNCS, vol. 8311. Springer, Heidelberg (2013)
6. Lamprecht, A.-L., Margaria, T.: Scientific Workflows and XMDD. In: Lamprecht, A.-L., Margaria, T. (eds.) Process Design for Natural Scientists. CCIS, vol. 500, pp. 1–13. Springer, Heidelberg (2014)
7. Lamprecht, A.-L., Margaria, T., Steffen, B.: Seven Variations of an Alignment Workflow - An Illustration of Agile Process Design and Management in Bio-jETI. In: Măndoiu, I., Wang, S.-L., Zelikovsky, A. (eds.) ISBRA 2008. LNCS (LNBI), vol. 4983, pp. 445–456. Springer, Heidelberg (2008)
8. Lamprecht, A.-L., Margaria, T., Steffen, B.: Bio-jETI: A framework for semantics-based service composition. BMC Bioinformatics 10(suppl. 10), S8 (2009)
9. Lamprecht, A.-L., Margaria, T., Steffen, B., Sczyrba, A., Hartmeier, S., Giegerich, R.: GeneFisher-P: variations of GeneFisher as processes in Bio-jETI. BMC Bioinformatics 9 (suppl. 4), S13 (2008)
10. Lamprecht, A.-L., Naujokat, S., Margaria, T., Steffen, B.: Semantics-based composition of EMBOSS services. Journal of Biomedical Semantics 2(suppl. 1), S5 (2011)
11. Lamprecht, A.-L., Wickert, A.: The Course's SIB Libraries. In: Lamprecht, A.-L., Margaria, T. (eds.) Process Design for Natural Scientists. CCIS, vol. 500, pp. 30–44. Springer, Heidelberg (2014)
12. Li, W.: Analysis and comparison of very large metagenomes with fast clustering and functional annotation. BMC Bioinformatics (2009)
13. Li, W., Godzik, A.: Cd-hit: A fast program for clustering and comparing large sets of protein or nucleotide sequences. Bioinformatics 22(13), 1658–1659 (2006)
14. Lis, M.: Constructing a Phylogenetic Tree. In: Lamprecht, A.-L., Margaria, T. (eds.) Process Design for Natural Scientists. CCIS, vol. 500, pp. 101–109. Springer, Heidelberg (2014)
15. Margaria, T., Steffen, B.: Agile IT: Thinking in User-Centric Models. In: Margaria, T., Steffen, B. (eds.) ISoLA 2008. CCIS, vol. 17, pp. 490–502. Springer, Heidelberg (2009)
16. Margaria, T., Steffen, B.: Business Process Modelling in the jABC: The One-Thing-Approach. In: Cardoso, J., van der Aalst, W. (eds.) Handbook of Research on Business Process Modeling. IGI Global (2009)

17. Margaria, T., Steffen, B.: Continuous Model-Driven Engineering. IEEE Computer 42(10), 106–109 (2009)
18. Margaria, T., Steffen, B.: Simplicity as a Driver for Agile Innovation. Computer 43(6), 90–92 (2010)
19. Margaria, T., Steffen, B.: Service-Orientation: Conquering Complexity with XMDD. In: Hinchey, M., Coyle, L. (eds.) Conquering Complexity, pp. 217–236. Springer, London (2012)
20. Margaria, T., Steffen, B., Reitenspieß, M.: Service-Oriented Design: The Roots. In: Benatallah, B., Casati, F., Traverso, P. (eds.) ICSOC 2005. LNCS, vol. 3826, pp. 450–464. Springer, Heidelberg (2005)
21. Naujokat, S., Lamprecht, A.-L., Steffen, B.: Loose Programming with PROPHETS. In: de Lara, J., Zisman, A. (eds.) Fundamental Approaches to Software Engineering. LNCS, vol. 7212, pp. 94–98. Springer, Heidelberg (2012)
22. Punta, M., Coggill, P., Eberhardt, R., Mistry, J., Tate, J., Boursnell, C., Pang, N., Forslund, K., Ceric, G., Clements, J., Heger, A., Holm, L., Sonnhammer, E., Eddy, S., Bateman, A., Finn, R.: The pfam protein families database. Nucleic Acids Research (2012)
23. Reso, J.: Protein Classification Workflow. In: Lamprecht, A.-L., Margaria, T. (eds.) Process Design for Natural Scientists. CCIS, vol. 500, pp. 65–72. Springer, Heidelberg (2014)
24. Schütt, C.: Identification of Differentially Expressed Genes. In: Lamprecht, A.-L., Margaria, T. (eds.) Process Design for Natural Scientists. CCIS, vol. 500, pp. 127–139. Springer, Heidelberg (2014)
25. Steffen, B., Margaria, T., Nagel, R., Jörges, S., Kubczak, C.: Model-Driven Development with the jABC. In: Bin, E., Ziv, A., Ur, S. (eds.) HVC 2006. LNCS, vol. 4383, pp. 92–108. Springer, Heidelberg (2007)
26. Vierheller, J.: Exploratory Data Analysis. In: Lamprecht, A.-L., Margaria, T. (eds.) Process Design for Natural Scientists. CCIS, vol. 500, pp. 110–126. Springer, Heidelberg (2014)
27. Wu, S., Zhu, Z., Fu, L., Niu, B., Li, W.: Webmga: A customizable web server for fast metagenomic sequence analysis (2011)