Construction techniques of Baikal microbiome research information-computational environment

E A Cherkashin and A O Shigarov

Matrosov Institute for System Dynamics and Control Theory, Siberian Branch
of Russian Academy of Sciences, Irkutsk, Russia,
Irkutsk Scientific Center, Siberian Branch of Russian Academy of Sciences, Irkutsk, Russia

E-mail: `eugeneai@icc.ru`

Abstract. A toolset and model data sources for research and developing an environment for Next Generation Sequencing data processing are considered in the paper. The environment is constructed on the base of model transformations targeted to industrial grade systems allowing domain specialists to carry on the Next Generation Sequencing research, which includes genetic data processing, visualization, and data integration. The integration allows one to get rid of restrictions imposed by an application library by its operation set and properties. The technique of the transformation is based on Model Driven Architecture principles and logical inference of the derived models and the code. The current results and the future fork are presented and discussed.

1. Introduction

In the last decade after the invention of methods for sequencing of new generation and their introduction in practice of biological systems research, a new direction of molecular genetics is formed, which is referred to as *metagenomics*. Its main object of study goes beyond the individual microscopic cultivated organisms to their communities, *microbiomes*. A total DNA (DeoxyriboNucleic Acid) is extracted from a sample, resulting in a general image of the microbiome. The method allows one to describe a significant number of new groups of organisms at all taxonomic levels. A comprehensive review of present sequencing approaches and challenges is presented in [1].

One of the types of the metagenomic studies is the *analysis of amplicons*. It is applied to investigation of the microbiota of different environments of Lake Baikal [2]. To perform the analysis, significant computational resources are required, as well as bioinformatics skills for analysis and interpretation. The Researcher composes the computational process by combining different modules of bioinformatic software, data conversion, data analysis, and visualization. To carry on the studies, the domain specialists are required to be skilled in scripting of the command shell of the operating system (Linux, Windows), running a distributed computing environment on a cluster computing system, and programming general and domain-specific languages, *e.g.*, Python and R.

The aim of this study is to develop mathematical and software support of the processes of analysis of the results of Next Generation Sequencing (NGS). We are to develop techniques and software for a visual representation of the computational process of amplicon analysis so that domain specialists would be able to compose computational pipelines, which are executed on distributed heterogeneous computing resources (clouds). The software implementation is a cloud infrastructure built on models of representation of the computational process in the form of a set of operations, structure and functions of the computing resources, and scheduling algorithms for computational resources.

In [3], the problem of cloud usage for storage and computing is stated as the basic problem of NGS since storage capacity exponential growth is slower as compared to the growth of the NGS generated data volume. The transportation between the mirror storages and the computer systems for processing data could easily cause the network capacity exhausting. The data processing in the general case of whole genome reconstruction requires terabytes of RAM and, in the case of cluster computing usage, special high-performance parallel algorithms. Two classes of users identified: *power user*, who analyses the genome, and *causal user*, who deals with the power user results, *e.g.*, integrating gene data between datasets and studies.

The domain of our IT R&D is related to processing data within natural science research activities, which are characterized with varieties of tasks, methods and multidisciplinary aims. We are observing constant increasing of the data obtained from field investigations, each new data is compared to all the data obtained in the previous years. The number of scientific problems is being increased too. Another problem to be solved is the interest of domain specialists, biologists, to be involved in data processing, guaranteeing a reasonable quality of processing. This processing pipeline should be automatic in general and be adjustable in terms of math- and bio-domains.

We propose to construct a PaaS and DaaS cloud consisting of independent network-connected SaaS-services adopted for MiSeq standard operational procedure (MiSeq SOP) used in microbiome research of Lake Baikal. PaaS will enable biologists to investigate data themselves. DaaS will allow bioinformatic specialists to work with data on demand when designing new data processing techniques. SaaS services will support the individual operations for PaaS. There is a number of SaaS software already made for NGS data processing, reference [3] contains their detailed survey.

2. Automation of MiSeq standard operational procedure

To be more concrete in the further reasoning, we briefly describe MiSeq SOP implemented with Mothur software [4]. This NGS data analysis process consists of the individual operations on genetic data, which are stored in files. To get accounted with the technique, we tried to process limnological data [5] according to the procedure presented on the Mothur website[1].

After executing the technique manually, we observed a number of issues. Parameter structure and output of Mothur commands are intricate: the researcher needs to trace file names change from a command to the following one. After each execution, Mothur commands add suffixes to the input file names. File naming depends on the input parameters, *e.g.*, the method used to process data. For example, after application `align.seq` to file named `HXH779K01.shhh.trim.good.unique.fasta` we obtain[2] `HXH779K01.shhh.trim.good.unique.align`, `HXH779K01...trim.good.unique.align.report`, and `HXH779K01...im.good.unique.flip.accnos`. Almost each operation adds new suffixes. Repeating the application of an operation to data results in repetitive suffixes. Mothur can pass correct filenames itself if the process goes directly forward, *e.g.*, in scripts. Making a mistake or returning to a previous step to refine coefficients, the researcher must trace the filenames manually.

After obtaining the results in the form of tables and charts, biologists usually repeat some filtering stages excluding additional OTUs (Operational Taxonomic Unit[3]), *e.g.*, which is similar to mitochondrial and chloroplast, but were not recognized within the MiSeq technique. Sometimes users want to replace a command with an analogous one from another package, *e.g.*, QUIIME2 or Usearch, to check the default one or take advantage of special features of the external command. In this case, data conversion must be performed.

[1] `https://www.mothur.org/wiki/MiSeq_SOP`

[2] File names are truncated in the beginning for better text layout.

[3] An abstract notion of species, used when the taxon is not determined or it is no sense in the determination.

Another possible but less frequent deviation from the technique is the involvement of the previously processed OTUs from early research, *e.g.*, samples of the previous year in the same places during ecological monitoring. In this case, the researcher needs to match the OTUs from different studies. User should write a routine for comparing OTU contents and merging group data.

Visualization is partially presented in the command set of Mothur. It is able to produce SVG vector images, but the images, in general, cannot be customized. Researchers have to use external software like R to build charts of the desired quality. Our experience shows that despite the time spent studying the chart building techniques, which is needed once, most time is spent for converting and filtering input data and refining parameters of the chart building commands.

3. Related works

Main R&D activity in the NGS domain is divided into these main directions:

- development of new efficient algorithms for data processing operations and building charts,
- organizing standardized cloud computing pipelines with HPC (High-Performance Computing) implementations of various operations,
- representing pipelines as workflows, as well as user interfaces to support interactive data processing and assessment.

At first, let us consider research devoted to the productivity of the algorithms and improving the analysis results quality. In [6], Go, C++, and Java programming languages were assessed with respect to the ease of implementation, memory consumption and overall computation performance, Go was chosen. The main requirements were addressed to big data string processing. The system is designed to store string under processing in main memory. It shares data parts between CPUs and limits input/output operations. Interesting is the fact that C++ was the slowest.

In [7], an NGS analysis pipeline was developed for the investigation of viruses' DNA contained in human body. The pipeline allowed medical engineers to focus studying on vaccine development. These findings demonstrated that the proposed NGS data analysis pipeline identified unknown viruses from the mixed clinical samples, revealed their genetic identity and variants, and characterized their genetic features in terms of the viral evolution. The process of detection is based on comparing parts of viral genome with the BLAST database and the coverage analysis.

Paper [8] deals with the implementation of a heuristic algorithm for *scaffolding*, *i.e.*, ordering, moving and orienting contigs using additional information to produce longer sequences on the next stages. In [9], a data-driven user interface and visualization are considered in the process of a clinical decision support system implementation. The user interface supplies decision-making for doctors and explanations for patients as a list of events of various kinds (medical records, NGS results over autopsy tissue, *etc.*) represented as HTML5 portlets. A comprehensive review of quality control, error detection and correction in processing NGS data is presented in [10].

The HPC techniques review we will start with an application [11] of the BOINC technology to the alignment procedure, where the Novoalign algorithm was scaled. The reference [3] reviewed the problem area, described the existing approaches and services already made, but has no mentions of the implemented cloud computing environments. Paper [12] contains an excellent review of current achievements in NGS and related areas. Authors doubt the possibility to organize a laboratory HPC-center based on cluster computing by NGS software users and suggest dealing with IaaS cloud computing. The paper has a very good review of existing commercial and open-source platforms allowing construction of pipelines of computing processes. Commercial software mostly implement predefined pipelines and inflexible, whereas open-source tends to implement either standardized pipelines or present a set of modules for individual operations and cloud service implementations, a *toolsets*. In [13], Rainbow software, a cloud implementation of NGS data processing, is considered. Rainbow is essentially a Perl script implementing map (division) for input and reduce (join) for output data, together with the distribution of the data pieces between Amazon EC2 cloud nodes. Cloud nodes perform only the alignment. The paper also has a good review of Linux virtual machine cloud distributions and bioinformatic packages. Another interesting review of cloud computing techniques is presented in [14]. Authors pay attention to the open-source cloud (Open Stack) and construction tools, like Common Workflow Language (CWL) [15] used to represent the computational process in a cloud. A comprehensive short survey can be found in [16], but at this point, it will not add essential data.

There are visual tools for genetic analysis, *e.g.*, Galaxy [17], which implements a popular approach (metaphor) of an interactive web page, where data are imported and processed with modules. Galaxy also can analyze the user script and constructs *dataflow* representation. Its primary purpose is to teach biologists to process single genome data, it can be extended to implement other NGS research procedures. This is an open-source project, under active development, and we could use it as one of the implementation platforms. Another tool is UGENE [18], it is a desktop application, open-source, written with QT5 framework, and it is also under active development. UGENE's primary function is visualizing workflows and gene data. The main criticism of the tools is presented in [19], where it is stated, that command line utilities support more functions and have higher flexibility than visual tools. Authors propose their own visual tool VisPro, connected to a cloud. The tool is built on the base of the agile approach, which assumes principal participation of the developer in the process of workflow construction, configuration and execution.

Summing up the review, we conclude that there is a good background to the construction of our infrastructure, and the techniques used in Baikal microbiome research must be adapted to this background. The technologies we spoke about before allow us to account the peculiarity of our problem, which is the requirement of greater flexibility of the computation process and domain user experience adaptation.

4. A MiSeq SOP automation approach

Limnologists perform both power and casual users activities [3], *i.e.*, processing both raw sequencing data and the result of the sequence processing, visualizing, comparing and generalizing results. HPC is usually based on two popular programming models [3]: MapReduce (Hadoop) and task programming. The first one implies that data can be split into subsets, which could be processed mostly independently. The results of the parallel processing, then, joined (reduced) to an aggregative object.

Mothur's simple filtering commands are easily run in parallel, indeed it uses CPU cores, but the computational complexity seems to be not as high as it would be reasonable to spend time on splitting and joining. The main reason for using clouds here is the accumulation of the RAM if data would not fit in the workstation memory. Some filtering is based on classification, which processes the whole gene data using subsampling. The transition of the corresponding algorithms to SaaS implies their substitution to a cluster version.

In general, to make our cloud computing architecture simpler, at the first stage of R&D we decided to use *the task queue* execution model, where computational resources execute individual tasks from a network of modules representing a variant of MiSeq SOP. The network of modules is constructed on the base of a MiSeq SOP model, where each module is a module of the Mothur package. The network is designed with Rapidminer studio as a dataflow. In figure 1 we represented the beginning of the MiSeq SOP. Integration with the cloud will require transfer data between DaaS and SaaS, store objects with metadata, and this is also accounted in our model.

5. Implementation concepts

The review of papers shows that as of to date there are two open-source projects related to automation of NGS data analysis being in active development: Galaxy and UGENE. If take Galaxy as the main data processing visualization technique, we are to make Galaxy modules adapting Mothur command, as well as to adapt Galaxy visualization techniques to the MiSeq SOP. On the other hand, we can do the same for UGENE, allowing users to work with a more responsive dynamic interface of the UGENE desktop application.

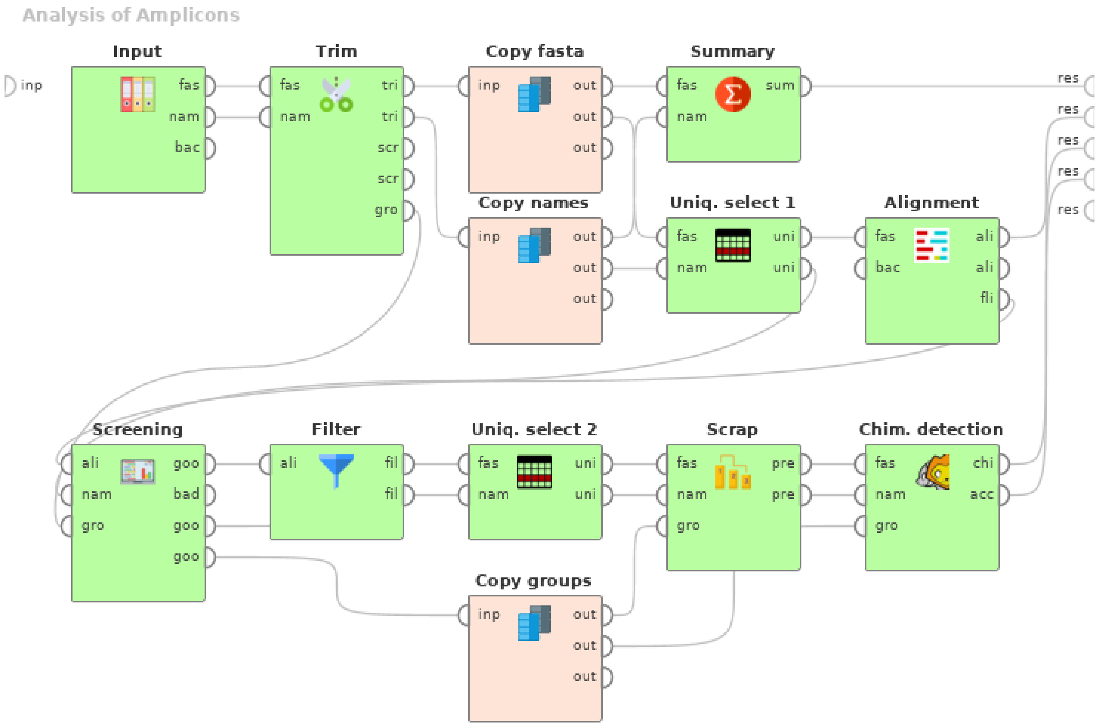

Figure 1. MiSeq first stages representation as dataflow modules [20]

In [20] and [21], we proposed and implemented a technique for dataflow representation of all Mothur commands. We use Model Driven Architecture (MDA) to generate modules for Rapidminer studio, a visual dataflow editor. According to MDA, the source code of modules is generated from the Platform Specific Model (PSM), which represents the software under development in a notation allowing the direct code generation by means of templates and other algorithmic procedures. In our case, PSM represents Java source code of the dataflow modules.

PSM is built out of the Platform Independent Model (PIM) representing the software on a more abstract level than PSM. It expresses the relations between entities, their object structures, metainformation and so on. The transition from PIM to PSM (a model transformation) is carried out by means of logical inference of the PSM properties on the base of facts representing PIM and the properties of the implementation platform, a Platform Model (PM), in our case it is Java programming language.

Some properties of PIM, *e.g.*, list of object fields, are constructed by transformation of Computationally Independent Model (CIM), an even more abstract model, which represents software as entities of the domain of Mothur commands. The transformation is also implemented as a logical inference realizing a pattern recognition. The CIM is also obtained automatically from the analysis of the C++ source code of Mothur. The analysis is implemented as a Python program scanning the sources for specific structures. Each command implementation is analyzed with a set of regular expressions matches organized in a scenario.

5.1. RDF data representation

The source model data are represented on the base of Semantic Web technologies. The model and its comprising structures are identified globally as resources. Relations between resources and literal values are expressed using standard and *ad hoc* designed ontologies. Usage of the ontologies allowed us to direct the research “along known spaces” of metadata and use the experience of the designers of ontologies, narrow the search space of the solutions.

In the cloud DaaS, we are to store files and their contents as objects with their metadata. The nowadays OMG (Object Management Group) standards describe specifications of converting relational, UML (Unified Modeling Language), SysML (System Modeling Language) metadata to RDF (Resource Description Framework) representation. So, we can store data in conventional relational or key-value databases and while retrieving supply metadata as well. In the simplest case, data and metadata can be stored in metadata storages, such as ClioPatria. While designing our cloud storage we use the JHipster Domain Language [22] and its tools to create database structures, metadata converters, formal representations of the ontologies of the stored data.

Metadata of the database stored objects describes mostly relations between a resource and its attributes. Some attributes, namely foreign keys, are the references to other resources, which are also reflected with metadata. There are rare relations between resources, which are not stored in the conventional databases. These relations reflect, *e.g.*, data provenance, additional special attributes for a particular data object. Such rare relations are infrequent and can be added in a special research investigation, so modification of the relational database structure for each of the cases has no sense. For the representation of the data, we adopted a number of standardized ontologies.

- Friend-of-a-friend (**foaf**) ontology is used for agent information: individuals, legal entities, program agents;
- Provenance (**prov**) is used for making references between documents;
- Dublin Core (**dc**) is used for published resource metadata mark up;
- DBPedia resource (**dbr**) refers external globally used classes and instance objects;
- Open annotation (**oa**) is used as a published document content representation ontology;
- The Bibliographic Ontology (**bibo**) is used for literature reference mark up.

For the representation of Mothur CIM and PIM, we developed two ontologies **mothur** and **uml**. CIM and PIM ontologies are used to represent relations between stored objects as subjects of input and output of Mothur commands.

The used approach allowed us to solve many technical problems, including providing our dataflow visualization tools with an actual set of Mothur commands, implementing an abstract engine of Mothur command properties mapping to a software environment. Within the R&D, we obtained a set of transformation scenarios expressed as object knowledge sets represented in the Logtalk [23] programming language, which is also used to generate PSM and the source code for the representation of Mothur commands for new computation and visualization environments.

5.2. Data integration: Metadata inference

To allow a casual user to take advantage of the obtained results, they are to be represented as RDF/RDFa[4] marked-up report documents (Word, Excel, PDF) and HTML5 (Hyper-Text Markup Language, version 5) web pages, *e.g.*, produced by Galaxy software. Such format allows both the user and a software agent to acquire the resulting data for their research. The markup for the documents is a part of our LOD[5]-based service providing integration with other NGS Internet resources. At present, we have not found a standardized way of the integration: there are only prototypes of annotation resources, like BioSearch [24] implemented on the base of BIO2RDF LOD technologies.

The LOD service and the desired flexibility of scientific research software require us to associate metadata to all pieces of NGS data. The metadata is stored for the main input data of MiSeq SOP and transformed with each application of commands into metadata describing command output objects. For Mothur, we construct an automatic metadata inference rules analyzing its C++ source and filename conversion algorithms. To conserve memory usage we decided to implement dynamic metadata

[4] An RDF language dialect for representing the sematic markup in web published documents.

[5] Linked Open Data, an RDF technology constraint defined by usage rules.

reconstruction when DaaS returns queried objects. For example, as thousands of sequences are organized in files, groups and OTUs, the metadata of the sequences are extended with all file/group/OUT metadata. Each sequence metadata is generated using the context of its storage, namely `fasta`–file name and relation to its group, its file provenance, *etc.*

6. Evaluation

We have been evaluating the implemented technologies on the criteria of expressiveness of the RDF source models (CIM, PIM, and PSM) representation, LogTalk programming capabilities of the transformation scenarios, and representation of the MiSeq SOP with the synthesized Rapidminer plug-in. The following results are obtained.

Semantic web technologies and knowledge graphs are universal way of describing data, basic relations between notions and model structures. The last procedure we have done was the analysis of Galaxy implementation of the Mothur's MiSeq SOP. We have spent about two days for converting it into RDF, resulting in more expressive representation of current 138 of 144 Mothur modules. The representation in the form of knowledge graphs of the source model data allows us aggregate various model data sources in one representation and select model elements with SPARQL and Prolog queries, which are interpreted in the transformation as a semantically meaningful target structures.

The Logtalk language has various structures, which allow programmer to express the transformational knowledge base with objects, providing tools for knowledge manipulation in the object-oriented way. The general scenario of the transformation is represented as a system of interconnected objects, encapsulating knowledge. Some of the objects organize facades for SPARQL and Prolog queries to the graph data, other generate target PSM structures and the source code. All the necessary structures of MDA were representable, as well as there is a number of Logtalk syntactic structures, whish are to be investigated in sense of their applicability for knowledge representation.

As we said before, as testing ground we used the source limnological data of [5]. At first stage we repeated the investigation procedure manually to grasp better understanding of the MiSeq SOP. After each refinement of the transformational knowledge base, we construct the procedure out of dataflow modules, generating scripts, which are executed and their results are compared to manually obtained ones. Some stages of Mothur MiSeq SOP contain algorithms employing subsampling, so the final results differ between runs.

A similar technique will be used in evaluation of the generated procedures in the Galaxy environment. The constructed dataflow diagram will be converted into a Galaxy notebook, executed and the results are evaluated against to the manually constructed procedure. At the last stage of the refinement, the power and casual users will be engaged to collect their opinions on the degree of convenience of the environment usage.

7. Further development

At this point, we developed means for MiSeq SOP modeling with dataflow diagrams in Rapidminer studio, execution of the models as creating scripts in the Mothur scripting language, together with MDA instrumentation support. The main problem to be solved in the future is as follows.

- Development of a converter of the models into Galaxy scenarios,
- Integrate data storage with Galaxy data import subsystem,
- Implement metadata storage and adapters for NGS data,
- Create a more sophisticated source code parser or PIM model of computation, so we could be able to infer metadata for synthesizing metadata conversion rules,
- Adopt our document authoring tools to Galaxy allowing LOD representation of results,
- Implement integration to biological/gene databases,
- Develop a set of predefined scenarios,

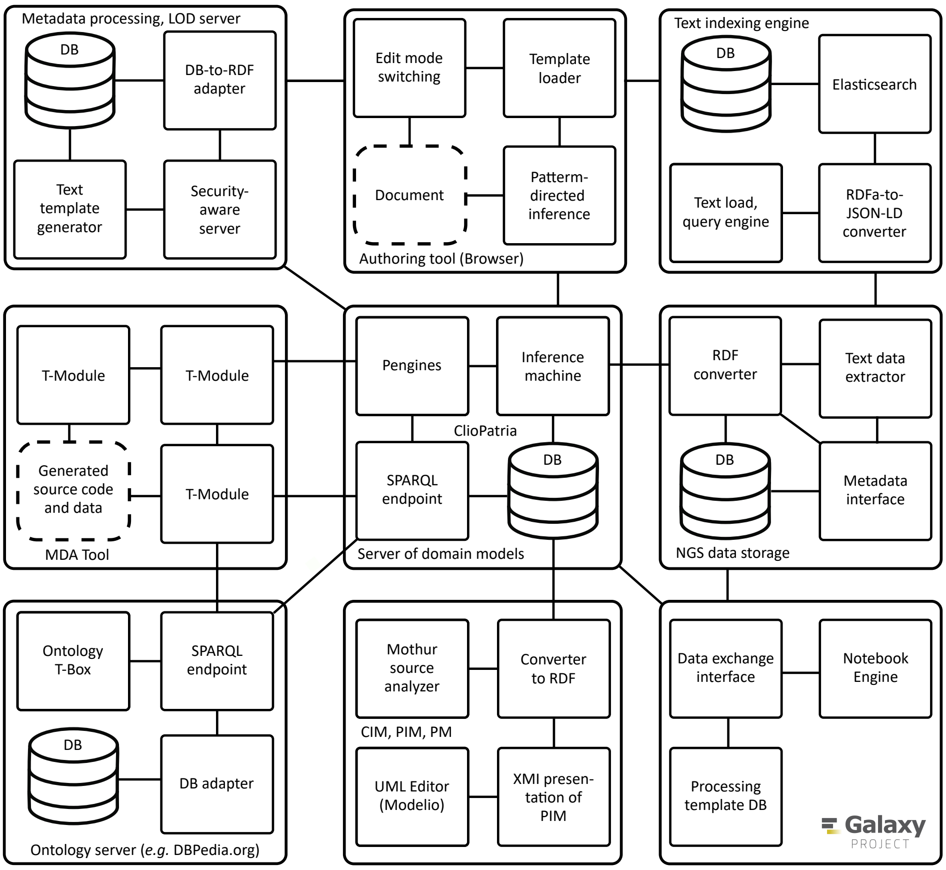

Figure 2. Architecture of information-computation environment

- Create scenario templates supporting research in monitoring.

The target modular software architecture is presented in figure 2.

Conclusions

The approach to the construction of an infrastructure for supporting Lake Baikal microbiome research based on Next Generation Sequencing is proposed. A good background of algorithms and software already constructed by various developers allows us to implement our environment utilizing adaptation of the techniques used by biologists to the background. The main contribution of the paper is as follows. We (1) analyzed the existing IT experience in the field of NGS data processing, (2) constructed a dataflow model of the technique (MiSeq implemented by Mothur package), which is to be converted to the software modules of various visual environments and cloud services. A conversion technique (3) has been developed using Model Driven Architecture (MDA), where model transformation is implemented as a logical inference system. This allows us transit from one implementation platform to another conserving gained and formalized experience.

At this stage, we limited our implementation with Mothur software specifics, but at the same time do not bother with data conversion. This will be relevant if we transit to an analogous software for the NGS data processing platform. The next stage will deal with data conversion to other NGS processing software such as open-source QIIME2 and proprietary ones like Usearch. These platforms have advantages over Mothur in data visualization and processing performance on special operations, as well as application of other methods and algorithms not included in Mothur. The NGS SaaS services [3, 12] could also be integrated into the cloud under development. Such an advantage will supply the better ground for carrying on experiments with data.

The specifics of the problems stated by Baikal microbiome research relate to construction mathematical models describing the microbial communities interaction. The models are constructed on the base of annual monitoring, data analysis and structural and parametric identification and refinement of model elements. In this case, the standard NGS procedures constructed out of operation applications must be extended with toolsets supporting user-friendly joining the previous stages of a continuous research. The infrastructure must support integral data representation and efficient query-based semantically rich access, and the proposed MDA approach allows us to quickly integrate new operations in the existing dataflow model. As we can see from our review, most realized techniques in the NGS data processing are either fixed uni-problem oriented software with *ad-hoc* infrastructures, or just a problem-oriented packages comprising sets of individual operations.

The problem set to be solved includes optimization of the utilization of cluster computing resources, planning parallel computing executions based on process structure analysis and the properties of algorithms that implement specific operations, and implementation of the control points for services.

Acknowledgment

The results related MDA transformation and its adaptation to RDF were obtained within the framework of the State Assignment of the Ministry of Education and Science of the Russian Federation for the project “Methods and technologies of cloud-based service-oriented platform for collecting, storing, and processing large volumes of multi-format interdisciplinary data and knowledge based upon the use of artificial intelligence, model-guided approach and machine learning” using facilities of the Centre of collective usage “Integrated information network of Irkutsk scientific educational complex”. The development of the infrastructure for Mothur command transformation to Rapidminer dataflow diagram is supported by the project of Irkutsk scientific center of Siberian branch of Russian Academy of sciences, grant No 4.2.

References

[1] Pereira R, Oliveira J and Sousa M 2020 Bioinformatics and computational tools for next-generation sequencing analysis in clinical genetics *J. Clin. Med.* **9** No. 1 1–30.

[2] Bashenkhaeva M V, Zakharova Yu R, Petrova D P *et al* 2015 Sub-ice microalgal and bacterial communities in freshwater lake Baikal, Russia *Environmental Microbiology* **70** No. 3 751–65.

[3] Guo X, Yu N, Li N and Pan Y 2016 Cloud computing for next-generation sequencing data analysis *Computational Methods for Next Generation Sequencing Data Analysis* ed I I Mandoiu and A Zelikovsky (John Wiley & Sons, Inc.) 3–24

[4] Kozich J J, Westcott S L, Baxter N T, Highlander S K and Schloss P D 2013 Development of a dual-index sequencing strategy and curation pipeline for analyzing amplicon sequence data on the MiSeq Illumina sequencing platform *Applied and Environmental Microbiology* **79** No. 17 5112–20

[5] Mikhailov I S, Zakharova Y R, Bukin Yu S *et al* 2019 Co-occurrence networks among bacteria and microbial eukaryotes of lake Baikal during a spring phytoplankton bloom *Microbial Ecology* **77** 96–109

[6] Costanza P, Herzeel C and Verachtert W 2019 A comparison of three programming languages for a full-fledged next generation sequencing tool *BMC Bioinformatics* **20** No. 1 (2019)301

[7] Gong Y-N, Chen G-W, Yang S-L, Lee C-J *et al* 2016 *A next-generation sequencing data analysis pipeline for detecting unknown pathogens from mixed clinical samples and revealing their genetic diversity*

[8] Gritsenko A A, Nijkamp J F, Reinders M J T and Ridder D de 2012 GRASS: a generic algorithm for scaffolding next-generation sequencing assemblies *Bioinformatics* **28** No. 11 1429–37

[9] Müller H, Reihs R, Posch A E, Kremer A, Ulrich D and Zatloukal K 2016 Data driven GUI design and visualization for a NGS based clinical decision support system *Procs. of 20th International Conference Information Visualization, 19–22 July 2016, Universidade NOVA de Lisboa, Lisbon, Portugal* 355–60

[10] Boekhorst R te, Naumenko F M, Orlova N G, Galieva E R, Spitsina A M *et al* 2016 Computational problems of analysis of short next generation sequencing reads *Vavilovskii Zhurnal Genetiki i Selektsii = Vavilov Journal of Genetics and Breeding* **20** No. 6 746–55

[11] Srimani J K, Wu P, Phan J H and Wang M D 2010 A distributed system for fast alignment of next-generation sequencing data *2010 IEEE International Conference on Bioinformatics and Biomedicine Workshops (BIBMW), Hong, Kong* 579–84

[12] Kwon T, Yoo W G, Lee W *et al* 2015 Next-generation sequencing data analysis on cloud computing *Genes Genom* **37** 489–501

[13] Zhao S, Watrous K, Zhang Ch and Zhang B 2017 Cloud computing for next-generation sequencing data analysis *Cloud Computing – Architecture and Applications* ed J Sen (IntechOpen Limited) 29–51

[14] Langmead B and Nellore A 2018 Cloud computing as a platform for genomic data analysis and collaboration *Nat. Rev. Genet.* **19** No. 4 208–19

[15] Amstutz P, Crusoe M R, Tijanic N, Chapman B *et al* 2016 *Common workflow language, v1.0*

[16] Baker Q B, Al-Rashdan W and Jararweh Y 2018 Cloud-based tools for next-generation sequencing data analysis *Procs. of Fifth International Conference on Social Networks Analysis, Management and Security (SNAMS), Valencia* 99–105

[17] Batut B, Hiltemann S, Bagnacani A, Baker D, Bhardwaj V *et al* 2018 *Community-driven data analysis training for biology cell systems*

[18] Rose R, Golosova O, Sukhomlinov D, Tiunov A and Prosperi M 2019 Flexible design of multiple metagenomics classification pipelines with UGENE *Bioinformatics* **35** No. 11 1963–5

[19] Milicchio F, Rose R, Bian J *et al* 2016 Visual programming for next-generation sequencing data analytics *BioData Mining* **9** No. 16

[20] Cherkashin E, Shigarov A, Malkov F and Morozov A 2019 An instrumental environment for metagenomic analysis *Information Technologies in the Research of Biodiversity. Springer Proceedings in Earth and Environmental Sciences* ed I Bychkov and V Voronin (Springer, Cham) 151–158 .

[21] Cherkashin E, Shigarov A and Paramonov V 2019 Representation of MDA transformation with logical objects *International Multi-Conference on Engineering, Computer and Information Sciences (SIBIRCON), Novosibirsk, Russia* 0913–8

[22] Halin A, Nuttinck A, Acher M, Devroey X, Perrouin G and Heymans P 2017 Yo variability! JHipster: a playground for web-apps analyses *Procs. of the Eleventh international workshop on variability modelling of software-intensive systems, VAMOS'17* (ACM, New York) 44–51

[23] Moura P 2009 Programming patterns for Logtalk parametric objects *Applications of Declarative Programming and Knowledge Management. Lecture Notes in Computer Science* ed A Abreu and D Seipel **6547** (Springer, Berlin, Heidelberg) 52–69

[24] Hu W, Qiu H, Huang J and Dumontier M 2017 BioSearch: a semantic search engine for Bio2RDF *Database* **2017** (2017)bax059