

Full length article

FSK-Lab – An open source food safety model integration tool

Miguel de Alba Aparicio¹, Tasja Buschhardt¹, Ahmad Swaid, Lars Valentin, Octavio Mesa-Varona, Taras Günther, Carolina Plaza-Rodriguez, Matthias Filter*

German Federal Institute for Risk Assessment (BfR), Department 4 – Biological Safety, Max-Dohrn-Straße 8–10, 10589 Berlin, Germany



ARTICLE INFO

Keywords:

Information exchange format
FSK-ML
Modelling
Open-source software
FoodRisk-Labs
Microbial risk assessment

ABSTRACT

In the last decades a large number of models have been developed in the quantitative microbial risk assessment (QMRA) and predictive microbiology (PM) domains. These models were generated with different scripting languages (e.g. R, MATLAB), commercial tools (e.g. @Risk) or even proprietary software tools (e.g. FSSP, FDA-iRISK). The heterogeneity in software tools used to generate models together with the lack of a harmonized model exchange format has made the (re)-use of existing models in different software tools or simulation environments very difficult. The adoption of a harmonized information exchange format called Food Safety Knowledge Markup Language (FSK-ML) would be a solution to this challenge. FSK-ML defines a framework for encoding all relevant data, metadata and model scripts in a machine-readable format. A specific feature of FSK-ML is that it allows the user to provide model scripts in different scripting languages (e.g. R, Perl, Python or MATLAB). Model metadata can be provided in accordance with the metadata schema and controlled vocabularies proposed from the Risk Assessment Knowledge Integration Platform (RAKIP) community.

In order to achieve a broad adoption of FSK-ML by the scientific community it is however of extraordinary importance to provide support in terms of easy to use software solutions. In fact, it has to be as simple as possible for the end-user to create, export, import or modify standard-compliant files. Food Safety Knowledge Lab (FSK-Lab) represents such a user-friendly software tool that can create, read (import), write (export), execute and combine FSK-ML compliant objects. All metadata needed to annotate a model can also be entered and edited through FSK-Lab. It also allows generating (export) files that comply with FSK-ML and that carry the file extension “.fskx”. This ensures that all information on a model is contained in this information exchange file and that the user does not have to write and compile FSK-ML files “by hand”.

FSK-Lab extends the open source Konstanz Information Miner (KNIME) data analytics platform (URL: www.knime.org), which is a graphical programming framework allowing users to create data analysis workflows from building blocks (so called nodes). Within FSK-Lab, each node has a specific task e.g. model creation can be performed with the “FSK Creator” node. As a KNIME extension, FSK-Lab allows the user to execute and integrate code from several programming languages, like Java, R, Python. All FSK-Lab software code is freely available under the GNU public license version 3.

1. Introduction

Quantitative microbial risk assessment (QMRA) and predictive microbiology (PM) modelling are fields of increasing relevance to food safety professionals and public authorities (Koutsoumanis et al., 2016; Membré and Guillou, 2016). In these research domains, great efforts

have been undertaken to create a rich variety of data and models (herein referred to as “knowledge”) and also software tools (resources) supporting end-users needs (Tenenhaus-Aziza and Ellouze, 2015). However, there is a gap in software solutions that enables food safety professionals and authorities to exploit knowledge from research publications in an efficient manner.

Abbreviations: FSK-Lab, Food Safety Knowledge Lab; FSK-ML, Food Safety Knowledge – Markup Language; FSSP, Food Spoilage and Safety Predictor; GUI, graphical user interface; IDE, integrated development environment; KNIME, Konstanz Information Miner; PM, predictive microbiology; PMP, Pathogen Modelling Program; QMRA, quantitative microbial risk assessment; RAKIP, Risk Assessment and Knowledge Integration Platform; SBML, Systems Biology Markup Language

* Corresponding author.

E-mail addresses: miguel.de-alba-aporicio@bfr.bund.de (M. de Alba Aparicio), tasja.buschhardt@bfr.bund.de (T. Buschhardt), ahmad.swaid@bfr.bund.de (A. Swaid), lars.valentin@bfr.bund.de (L. Valentin), octavio.mesa-varona@bfr.bund.de (O. Mesa-Varona), taras.guenther@bfr.bund.de (T. Günther), carolina.plaza-rodriguez@bfr.bund.de (C. Plaza-Rodriguez), matthias.filter@bfr.bund.de (M. Filter).

¹ These authors contributed equally.

<https://doi.org/10.1016/j.mran.2018.09.001>

Received 28 February 2018; Received in revised form 18 September 2018; Accepted 18 September 2018

Available online 19 September 2018

2352-3522/ © 2018 German Federal Institute for Risk Assessment. Published by Elsevier B.V. This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>).

The diversity of software tools and the lack of harmonized information exchange formats for existing knowledge have been accompanied by a variety of problems. One problem arises through software incompatibilities that hinder the simple “import” of available models into current modelling or simulation tools. Users are consequently forced to manually re-encode a model in the software tool that fits their needs. This process is time-consuming and error-prone and hampers the ability to combine or exchange existing knowledge. As an example, even though there is a basic import feature for simulation results of predictive models in the software tool FDA-iRisk® (URL: <https://irisk.foodrisk.org/>) it still needs data formatting skills to convert e.g. the output of prediction tools like Pathogen Modeling Program (PMP) (URL: <https://pmp.errc.ars.usda.gov/default.aspx>) or Food Spoilage and Safety Predictor (FSSP) (URL: <http://fssp.food.dtu.dk/>) into the required format. If the import and exchange format of knowledge was standardized between software tools users would be able to spend more time on actual research rather than struggling with the use of different software tools and data formats.

An additional problem is the re-use of models (and data) from peer-reviewed publications. Extracting relevant information (e.g. equations, parameters, and model metadata) from a publication can seem daunting, as model components and data are often not described with sufficient detail. Re-implementing these models into existing software tools would therefore require excessive time and effort and the process would be prone to errors. The adoption of a standardized model exchange format in peer-reviewed publications would allow straightforward comparison, re-use and extension of published models, supporting effective progress within the QMRA and PM modelling domains.

The lack of common terminologies and incompleteness of metadata is another problem that negatively affects the exploration and re-use of knowledge. Proper model annotation is a prerequisite for understanding what a model is about and for correctly interpreting model-based results (Plaza-Rodriguez et al., 2015). The establishment of a set of harmonized metadata fields helps avoiding misunderstandings and improves the quality and efficiency of knowledge exchange (Haberbeck et al., 2018). Such harmonized metadata are also of high importance for search and filter functions in model repositories.

A general strategy to overcome those problems mentioned above has been proposed by Plaza-Rodriguez et al. (2015). This strategy includes the establishment and adoption of harmonized information exchange format(s) and the creation of software tools supporting such format(s). The present work provides an update on the progress made along this strategy. Specifically it will provide a short introduction into the information exchange format called Food Safety Knowledge Markup Language (FSK-ML) and the open source software tool Food Safety Knowledge Lab (FSK-Lab) which allows modellers to create and share their QMRA or PM models in this harmonized file format.

2. Methods

2.1. FSK-ML

FSK-ML is a formal specification of an information exchange format that supports the harmonized description and the exchange of script-based models in the domains of QMRA and PM. A FSK-ML software developer guidance document (URL: <https://foodrisklabs.bfr.bund.de/fsk-ml-food-safety-knowledge-markup-language/>) has been created that explicitly specifies the structure and content of FSK-ML files. FSK-ML has been designed by adopting principles of related standards, namely the COMBINE Archive, the Simulation Experiment Description Markup Language (SED-ML) and the Systems Biology Markup Language (SBML). A COMBINE archive is a single file bundling the various documents necessary for a modelling and simulations project, and all relevant information. The archive is encoded using the Open Modeling EXchange format (OMEX) (Bergmann et al., 2014). SED-ML is an XML-based format for encoding simulation experiments. SED-ML allows to

define the models to use, the experimental tasks to run and which results to produce (Waltemath et al., 2011). SBML (URL: http://sbml.org/Main_Page) is a computer-readable XML format for representing models of biological processes. SBML is suitable for, but not limited to, models using a process description approach (Hucka et al., 2015).

One specific modification of FSK-ML is the support of models generated in various scripting languages or desktop applications. This new feature is of high relevance in the QMRA and PM modelling domains as most legacy models are only available in specific script-based programming languages (e.g. R, MATLAB and Python), stand-alone software tools (e.g. @Risk), or proprietary software tools (e.g. FSSP, FDA-iRISK). Still, the extent to which FSK-ML compliant models can be executed or even combined depends on the capabilities of each software tool that reads FSK-ML formatted files. As an example, FSK-Lab is currently able to execute, modify or combine models from the R scripting language, and will support Python models in the very near future.

FSK-ML also defines how to specify and exchange metadata on QMRA and PM domain models. Specifically, FSK-ML exploits an annotation schema with harmonized metadata concepts that were developed by the RAKIP community (Haberbeck et al., 2018). In this annotation schema, each metadata term is explicitly defined to support understanding and interpretation of model results. The metadata terms are designed in a way that allows different types of knowledge (e.g. exposure models, dose-response models, predictive models, data, etc.) to be annotated with the same annotation schema. Through use of the same annotation schema it becomes possible to provide a harmonized machine-readable annotation for different types of models within the QMRA and PM domain.

At the core, FSK-ML defines a standard for a new file format that can store all information on a model in a compressed ZIP folder and assign such files the “.fskx” file extension. This FSKX file contains e.g. a script with code describing the model, a script with code for visualizations of results, and a file with all model metadata. FSKX files can also contain additional information like a Readme file, image files, simulation result files, R workspaces, publications, etc. The user can access the files within the FSKX archive by simply un-zipping the compressed file format.

2.2. FSK-Lab

FSK-Lab is an extension to the open-source software KNIME (URL: www.knime.org) (Berthold et al., 2006) which is a platform for integrated data access, data mining, statistics, data visualization and reporting. KNIME allows users to assemble data analysis workflows from functional building blocks called “nodes”. These workflows are like flowcharts where each process (“node”) performs a specific task. Examples of KNIME workflows with FSK-Lab nodes are shown in Fig. 1. KNIME nodes can be executed alone or together with other nodes of a workflow. If nodes are to be executed together they have to be connected (like processes in a flowchart). Connections (“edges”) between nodes are used to transfer data from one node to the other. Most nodes have specific graphical elements to connect an incoming edge to (“in-port”, on left side of node) or to provide data to other nodes (“out-port”, on right side of node). By double-clicking on a node icon a graphical user interface (GUI) opens that enable the user to provide node-specific settings. The execution status of each node is represented by a traffic light symbol below the node. A yellow symbol indicates that the node settings are completed and the node is ready for execution, green indicates that the node was successfully executed (the specific task of the node was finished), and red indicates that the node settings are not completed. If a red symbol with a cross occurs below the node it indicates that the execution of the node failed and an error message will appear in the KNIME console. After the successful execution of a node, its result can be accessed and inspected via the node’s menu from the node’s out-port(s). Workflows as a whole can be saved, exported and

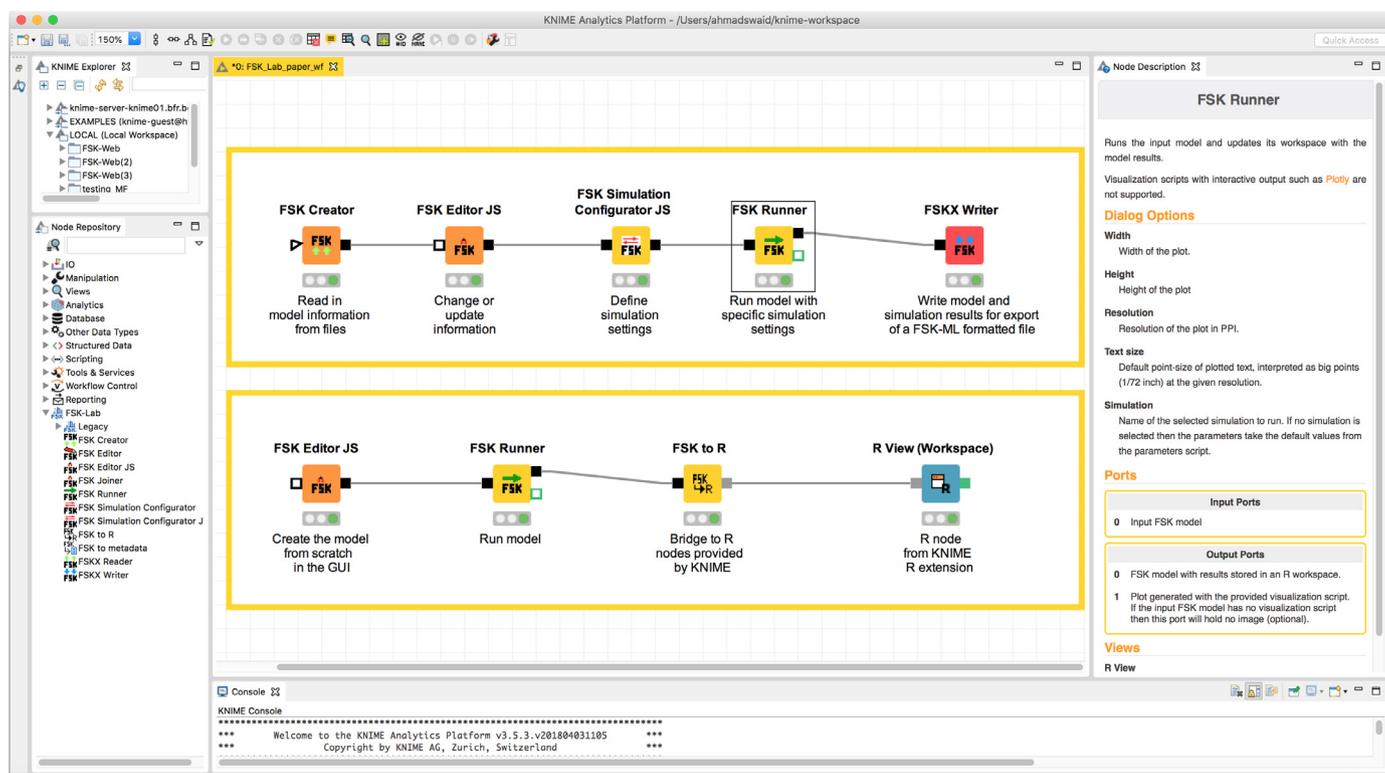


Fig. 1. Screenshot of the FSK-Lab toolkit which is integrated into the KNIME software. The customizable KNIME software graphical user interface is comprised of five panes (windows): KNIME Explorer—providing an overview on all workflows in the active workspace (upper-left); Node Repository—displaying all available nodes structured into so called libraries—including the FSK-Lab node library (lower left); Workflow Editor—an interactive workflow generation, modification and execution view, here with two workflows (center); Console—place where warnings or error message are displayed to the user (bottom); and Node Description—place where the functionalities and possible settings of each node are described (upper right).

exchanged. This helps to create transparency and quality control on performed data processing steps and supports collaboration among different users.

FSK-Lab extends KNIME through a collection of new nodes, that add specific functionality while complying with the KNIME interface definition (Berthold et al., 2006). FSK-Lab nodes were programmed in Java using the open-source Integrated Development Environment (IDE) Eclipse. The third party open-source software libraries used in FSK-Lab are listed in Supplement 1. With respect to FSK-ML, the software FSK-Lab also serves as a reference implementation, meaning it is the main software used to test (and improve) practical and technical issues arising from the FSK-ML guidelines. As FSK-Lab is made available as GNU public license version 3 (GPLv3) the software code can be used under the conditions defined by the licence by other software developers in their tools.

2.3. Example of FSK-Lab use and application

Due to its modular design and its integration into KNIME, FSK-Lab can be used for an indefinite number of data analysis tasks. On the other hand, it is possible to provide “template workflows” (designed for a very specific task), which can be used by end-users without substantial configuration efforts. As an example, we provide such a FSK-Lab “template workflow” together with a sample model file in FSK-ML format (both files are provided as supplemental material: Example_workflow.knwf and Example_QRA.fskx). The sample model is a simplified QMRA on *Escherichia coli* (*E. coli*) O157:H7 infections linked to the consumption of frozen ground beef in children under three years of age. This model was originally published as a basic QMRA example (Pouillot and Delignette-Muller, 2010). It is important to note that the model is fictitious and results should not be interpreted. The R code of the original model was published in the R package manual for

mc2d under “1.3 A basic example” (Pouillot et al., 2016). To execute the “template workflow” it is required to install KNIME with the FSK-Lab extension. Detailed FSK-Lab installation instructions are provided at (URL: <https://foodrisklabs.bfr.bund.de/fsk-lab/>). Video tutorials were created to explain the incentive behind the development of FSK-ML and FSK-Lab and to showcase the functionalities of FSK-LAB. These video tutorials can be accessed here URL: XXXX.

3. Results

FSK-Lab is a free, modular, open-source toolkit for the generation, annotation, execution and combination of script based QMRA and PM models. FSK-Lab extends the open source data analytics platform KNIME and provides a set of new software features called nodes. Through the connection and combination of nodes into a data processing workflow users can perform desired data processing, modelling or simulation tasks (Fig. 1). Thus, FSK-Lab can be characterized as a generic graphical modelling toolbox that also supports the exchange and integration of existing risk assessment models from different programming languages, as e.g. R, MATLAB, Python. Currently, FSK-Lab only supports integration of models written in R-code, but other programming languages and stand-alone software tools will be supported in the future.

A specific feature of FSK-Lab is the support (reading and writing) for the information exchange format FSK-ML. By supporting FSK-ML, FSK-Lab promotes efficient information exchange, transparency and application of good scientific practice within the food safety modelling domain (Plaza-Rodríguez et al., 2018). The provisioning of models in a harmonized file format will allow end-users in the future to retrieve models developed by others easily from e.g. community model repositories, which are websites with user-friendly access (including search and filter functionalities) to models encoded in FSK-ML. An

Table 1
Overview of currently available FSK-Lab nodes and their specific functionality.

Node	Node description/functionality
<i>FSK Creator</i>	Creates an FSK object. The user has to provide a model script (defining the model in a supported scripting language) and the model metadata in a pre-defined Excel template (in Microsoft® Office Excel format). A visualization script (defining graphical visualization of results in scripting language) can optionally be provided; Information on model input and output parameters and their default values have to be provided inside the metadata template. The given default parameter values define the <i>defaultSimulation</i> of the model.
<i>FSK Editor</i>	Allows to edit the content of an existing FSK object or to provide content for a new FSK object from scratch.
<i>FSK Joiner</i>	Combines two FSK objects graphically into a new joint model. The joint model is made available as FSK object at the Out-port.
<i>FSK Runner</i>	Execute the model given at the In-port with a user-defined simulation setting. If the model contains a visualization script the generated plot is accessible at the node's image Out-port.
<i>FSK Simulation Configurator JS</i>	Allows to create user-defined simulation settings for any FSK model. The settings of the <i>defaultSimulation</i> will not be modified and is always kept together with the user-defined simulation setting.
<i>FSK to R</i>	Extracts the R workspace out of an FSK object and places it into the node's Out-port as <i>RPortObject</i> . If the R workspace is null or non-existent (as when the model has not been run), the generated <i>RPortObject</i> would be null.
<i>FSK to metadata</i>	Extracts the metadata from an FSK object into a KNIME table.
<i>FSKX Reader</i>	Creates an FSK object from an FSK-ML compliant FSKX file.
<i>FSKX Writer</i>	Creates an FSK-ML compliant FSKX file.

example of such a model repository is the Risk Assessment and Knowledge Integration Platform (RAKIP) (URL: <https://foodrisklabs.bfr.bund.de/rakip-model-repository-web-services/>).

3.1. FSK-Lab functionalities and the modular software design

A central methodological concept of FSK-Lab is modularization. This means that FSK-Lab was designed not as a monolithic piece of software with a limited number of functionalities, but as a tool that provides the highest possible degree of flexibility in terms of what kind of data processing tasks can be accomplished.

The functionality of FSK-Lab is therefore provided through a collection of KNIME “nodes”, which can be selected from the KNIME “node repository”. Each FSK-Lab node performs a specific task and through the combination of nodes it is possible to create use-case specific data processing workflows from the same “building blocks”. Table 1 provides an overview of currently available FSK-Lab nodes and their specific functionality.

For example, it is possible to combine the FSK Runner node with any other FSK-Lab node that has an FSK object out-port, e.g. the FSK Reader, FSK Creator, FSK Editor JS or FSK Joiner. In this way a user can easily create customized data processing pipelines according to his specific needs. In the simplest case, a user might just want to import and execute an FSK-ML compliant model. For this it is sufficient to connect the FSK Reader node (that allows importing the “.fskx” file) with the FSK Runner node (that does the model execution). Other users might want to perform more complex data processing or modelling tasks (as e.g. illustrated in Fig. 1). A particular workflow might execute the following tasks: creating FSK model object from existing R-scripts and an Excel file with metadata, updating the model metadata, creating new simulation settings, executing the model and exporting the model and the simulation results into an FSK-ML compliant file (see upper workflow in Fig. 1, from left to right).

To facilitate such modular software design, each FSK-Lab node has been equipped with a node-specific GUI allowing the user to provide node-specific settings (Figs. 2a and b and 3). For example the GUI of the FSK Editor JS simplifies metadata provisioning (Fig. 2a), the FSK Simulation Configurator JS allows users to change simulation settings (Fig. 2b), and the FSK Joiner GUI can be used to join two models into a new, more complex, model (Fig. 3). Further, FSK-Lab ensures, that the result of each node's data processing task is made available in human-readable form at the node's out-port(s). This includes FSK model objects or images generated during model execution.

The modular design concept of FSK-Lab also enriches the application potential of KNIME itself, as information generated by FSK-Lab nodes can be further processed with nodes from the KNIME R Integration extension and through that, with all other KNIME or KNIME

community nodes. This functionality is illustrated in workflow b in Fig. 1, where the FSK to R node is used to transfer an R workspace from FSK-Lab to an R node from KNIME.

Finally, the modular design concept ensures that new functionalities can easily be implemented as new FSK-Lab nodes without causing problems with already existing FSK-Lab functions or workflows.

4. Discussion

FSK-Lab has been developed with the main objective to facilitate efficient exchange of models in the food safety modelling domain. To achieve this objective, FSK-Lab was established as open source software within the community driven data analytics platform KNIME. Through KNIME, FSK-Lab inherits a “software language” integration functionality, making it capable of combining models from different programming languages (e.g. from R, Python, or Java). Further, FSK-Lab integrates into the intuitive graphical programming workflow engine of KNIME supporting the re-use, easy adaptation and combination of existing models into new risk assessment models. Another innovative FSK-Lab feature in this context is the support for FSK-ML. FSK-ML is the first domain-specific information exchange format that supports legacy models generated in different programming languages or tools. On this side, FSK-Lab serves also as a reference implementation for FSK-ML and allows users to annotate complete risk assessment models or certain building blocks of larger risk assessment models (e.g. a predictive microbial model, process model, exposure model, dose-response model, health metric model, consumption model, other empirical model) in a consistent and harmonized manner. FSK-ML adopts the metadata and annotation schema proposed by the RAKIP community (Haberbeck et al., 2018). The adoption of FSK-ML as a domain-specific information exchange format by other software tools would allow efficient and transparent exchange of models within the risk assessment community. It would also promote the exchange of models via community driven online resources like the RAKIP model repository that could serve as a bridge between modelling experts and model “end-users”.

Due to its modular software design and its open source license, FSK-Lab can be improved and missing functionalities can easily be included, also by others. In the near future FSK-Lab will be further validated and new features will be made available. Given the provided application flexibility together with full transparency on all data processing steps, FSK-Lab might become a valuable tool for the food safety modelling domain in the future.

5. Conclusion

This research introduces FSK-Lab as a new open source software

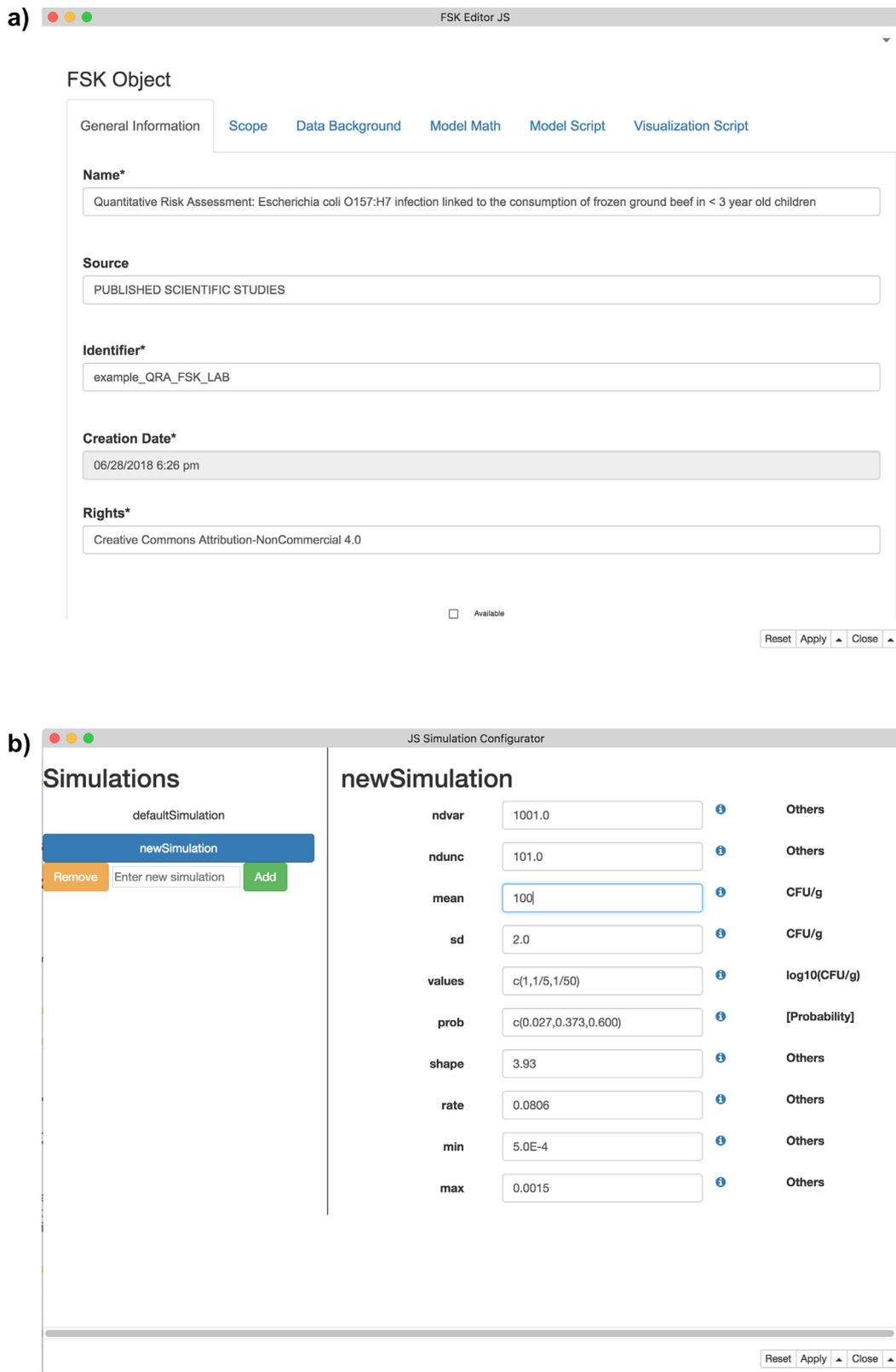


Fig. 2. (a) Screenshot of the FSK Editor JS GUI. The user can provide detailed annotation for all relevant aspects of the model including the model input and output parameters. The provided metadata fields are synchronized with the model annotation schema proposed by the RAKIP community (see also Haberbeck et al., 2018). This GUI is also equipped, with appropriate drop-down menus for available controlled vocabularies. Mandatory metadata fields are highlighted with *. (b) Screenshot of the Simulation Configurator JS GUI. Model input parameters and their default values are displayed. The user can change the default simulation settings by adding a new simulation. The i-icons provide additional information about each parameter.

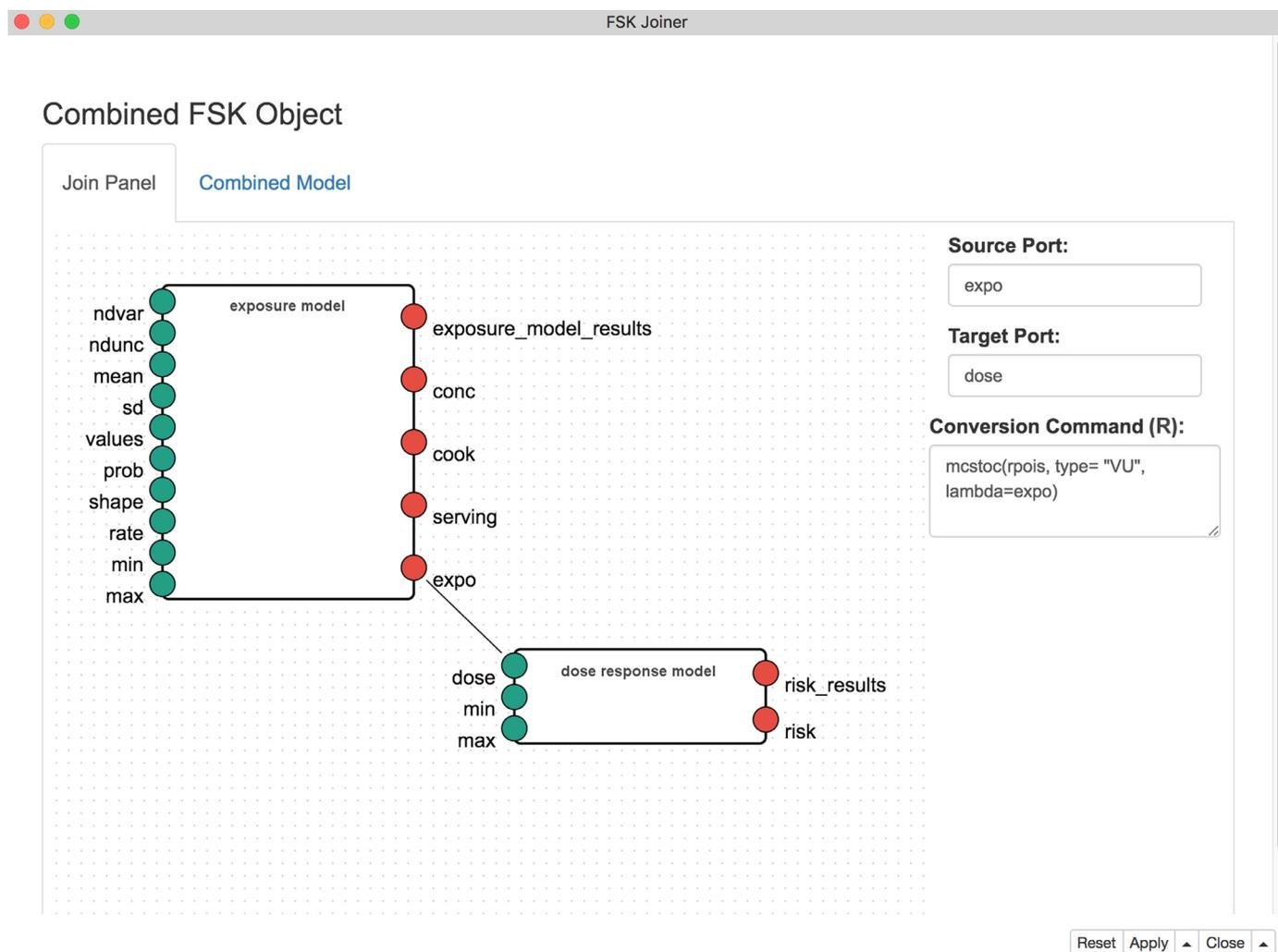


Fig. 3. Screenshot of the FSK Joiner configuration GUI. This GUI allows the graphical combination of two FSK models by creating a link between any of the model outputs of the first model to any of the inputs of the second model. Via a click on an established link it is possible to provide small scripts that are executed on the output of the first model before the values are used as input for the second model (e.g. to allow for conversion of units, averaging, grouping etc.).

resource for the food safety modelling domain. FSK-Lab is the first tool that also supports the information exchange format FSK-ML that allows exchanging models generated in different programming languages, like Java, R or Python in a harmonized way. Such a harmonized format is a pre-requisite for future import functions in any risk assessment software tool that aims at “importing” models from 3rd parties. Beside other user groups FSK-Lab specifically supports modelers in performing the unavoidable extra work linked with the adoption of the proposed FSK-ML format. As soon as there are FSK-ML supporting libraries in other tools and languages this will become even more convenient for the modeler and end-users. Due to the modular design concept of FSK-Lab this tool may become a valuable resource for the different user groups in the risk assessment domain as each user can customize it to his specific application scenario and needs.

Funding

The work presented in this paper has been partly supported by the AGINFRA PLUS project that is funded by the European Commission's Horizon 2020 research and innovation programme under grant agreement No. 731001, the EFSA-BfR Framework Partnership Agreement GP/EFSA/AMU/2016/01, by funds of the Federal Ministry of Food and Agriculture 2816502914 (BMEL) for the project “FoodAuthent” based on a decision of the Parliament of the Federal Republic of Germany via

the Federal Office for Agriculture and Food (BLE) under the innovation support programme as well as by the Federal Institute for Risk Assessment (BfR), the National Food Institute (DTU Food) from the Technical University of Denmark (DTU) and the French Agency for Food, Environmental and Occupational Health & Safety (ANSES).

The content of this paper does not reflect the official opinion of the European Union, EFSA, BMEL or BLE. Responsibility for the information and views expressed in this paper lies entirely with the author(s).

Declarations of interest

None.

Glossary

- FSK-ML (Food Safety Knowledge – Markup Language): FSK-ML is a specification of a knowledge exchange format that can be used to encode models in the food safety risk assessment domain.
- FSKX-file: File format defined within the FSK-ML specifications supporting the exchange of risk assessment models.
- KNIME: platform for integrated data access, data mining, statistics, data visualization and reporting.
- RAKIP: Risk Assessment and Knowledge Integration Platform is an initiative that aims at promoting the sharing and execution of

curated QMRA and PM models using the foundation of the proposed harmonized information exchange format (FSK-ML).

Acknowledgements

We thank Armin Weiser, Christian Thöns, Kilian Thiel for their valuable contribution to the creation of the FSK-Lab software, Sascha Bulik for his contribution to FSK-ML and the RAKIP team for their work on the RAKIP terms and concepts metadata schema.

Supplementary materials

Supplementary material associated with this article can be found, in the online version, at doi:[10.1016/j.mran.2018.09.001](https://doi.org/10.1016/j.mran.2018.09.001).

References

- Bergmann, F.T., Adams, R., Moodie, S., Cooper, J., Glont, M., Golebiewski, M., Hucka, M., Laibe, C., Miller, A.K., Nickerson, D.P., Olivier, B.G., Rodriguez, N., Sauro, H.M., Scharm, M., Soiland-Reyes, S., Waltemath, D., Yvon, F., Le Novere, N., 2014. COMBINE archive and OMEX format: one file to share all information to reproduce a modeling project. *BMC Bioinform.* 15, 369.
- Berthold, M.R., Cebon, N., Dill, F., Di Fatta, G., Gabriel, T.R., Georg, F., Meinel, T., Ohl, P., Sieb, C., Wiswedel, B., 2006. KNIME: The Konstanz Information Miner. In: 4th International Industrial Simulation Conference 2006, ISC 2006, pp. 58–61.
- Haberbeck, L.U., Rodríguez, C.P., Desvignes, V., Dalgaard, P., Sanaa, M., Guillier, L., Nauta, M., Filter, M., 2018. Harmonized terms, concepts and metadata for microbiological risk assessment models: the basis for knowledge integration and exchange. *Microb. Risk Anal.*
- Hucka, M., Bergmann, F.T., Hoops, S., Keating, S.M., Sahle, S., Schaff, J.C., Smith, L.P., Wilkinson, D.J., 2015. The Systems Biology Markup Language (SBML): Language specification for level 3 version 1 core. *J. Integr. Bioinform.* 12 (2), 266.
- Koutsoumanis, K.P., Lianou, A., Gougouli, M., 2016. Latest developments in foodborne pathogens modeling. *Curr. Opin. Food Sci.* 8, 89–98.
- Membré, J.-M., Guillou, S., 2016. Latest developments in foodborne pathogen risk assessment. *Curr. Opin. Food Sci.* 8, 120–126.
- Plaza-Rodríguez, C., Haberbeck, L.U., Desvignes, V., Dalgaard, P., Sanaa, M., Nauta, M., Filter, M., Guillier, L., 2018. Towards transparent and consistent exchange of knowledge for improved microbiological food safety. *Curr. Opin. Food Sci.* 19, 129–137.
- Plaza-Rodríguez, C., Thoens, C., Falenski, A., Weiser, A.A., Appel, B., Kaesbohrer, A., Filter, M., 2015. A strategy to establish Food Safety Model Repositories. *Int. J. Food Microbiol.* 204, 81–90.
- Pouillot, R., Delignette-Muller, M.-L., Kelly, D.L., Denis, J.-B., 2016. The mc2d package. <https://cran.r-project.org/web/packages/mc2d/vignettes/docmcEnglish.pdf>.
- Pouillot, R., Delignette-Muller, M.L., 2010. Evaluating variability and uncertainty separately in microbial quantitative risk assessment using two R packages. *Int. J. Food Microbiol.* 142 (3), 330–340.
- Tenenhaus-Aziza, F., Ellouze, M., 2015. Software for predictive microbiology and risk assessment: a description and comparison of tools presented at the ICPMF8 Software Fair. *Food Microbiol.* 45 (Pt B), 290–299.
- Waltemath, D., Adams, R., Bergmann, F.T., Hucka, M., Kolpakov, F., Miller, A.K., Moraru, I.I., Nickerson, D., Sahle, S., Snoep, J.L., Le Novere, N., 2011. Reproducible computational biology experiments with SED-ML—the Simulation Experiment Description Markup Language. *BMC Syst. Biol.* 5, 198.