

# Food Safety Knowledge Markup Language (FSK-ML)

---

*Software Developer Guide*

*Version 3.1*

Matthias Filter (Chair)  
Miguel de Alba Aparicio  
Esther M. Sundermann  
Marcel Fuhrmann

German Federal Institute for Risk Assessment  
German Federal Institute for Risk Assessment  
German Federal Institute for Risk Assessment  
German Federal Institute for Risk Assessment

***Alumni contributors:***

Sascha Bulik  
Guido Correia Carreira  
Alexander Falenski  
Carolina Plaza-Rodriguez

German Federal Institute for Risk Assessment  
German Federal Institute for Risk Assessment  
German Federal Institute for Risk Assessment  
German Federal Institute for Risk Assessment

***Contact:***

Matthias Filter ([matthias.filter@bfr.bund.de](mailto:matthias.filter@bfr.bund.de))

## Inhalt

1.	Introduction .....	4
2.	Related Standards.....	5
2.1.	Systems Biology Markup Language (SBML) .....	5
2.2.	Open Modelling EXchange Format (OMEX) .....	5
2.3.	Predictive Modelling in Food Markup Language (PMF-ML) .....	5
2.4.	Food Safety Knowledge Markup Language (FSK-ML).....	6
2.5.	Food Safety Knowledge Simulation Experiment Description Markup Language (FSK-SED-ML) 6	
3.	Document Conventions .....	7
3.1.	Terminology .....	7
	Generic Metadata Schema .....	7
	Metadata .....	7
	Model Class .....	7
	Simulation .....	7
3.1.	Implementation Convention .....	8
3.1.1.	Type SId .....	8
3.1.2.	Script Language Specificity .....	8
3.1.3.	Referencing Script Code.....	8
3.2.	Typographical Conventions .....	9
4.	FSK-ML Specification: Single Model .....	10
4.1.	Container Structure Related Files .....	12
4.1.1.	Manifest .....	12
4.1.2.	Metadata File.....	13
4.1.3.	Version Control.....	14
4.2.	Model Related Files .....	15
4.2.1.	Model Script .....	15
4.2.2.	Model Metadata.....	15
4.2.3.	Model Related Data .....	18
4.2.4.	Third Party Packages .....	18
4.3.	Simulation Related Files .....	18

4.3.1.	Simulation Settings.....	19
4.3.2.	Simulation Results .....	19
4.3.3.	Visualization Scripts .....	19
5.	FSK-ML Specification: Joined Model.....	19
5.1.	Joining Related Files.....	21
5.1.1.	Joined Model Script.....	21
5.1.2.	Model Metadata.....	22
5.1.3.	Simulation Settings.....	22
5.2.	Container Structure Related Files .....	22
5.2.1.	Manifest .....	22
5.2.2.	Metadata File.....	22
5.2.3.	Version Control.....	22
5.3.	Single Model Related Files .....	23
6.	FSK-ML Specification: Data .....	23
6.1.	Container Structure Related Files .....	25
6.1.1.	Manifest .....	25
6.1.2.	Metadata File.....	25
6.1.3.	Version Control.....	25
6.2.	Data Related Files .....	25
6.2.1.	Data Storage .....	26
6.2.2.	Data Metadata.....	26
6.3.	Visualization .....	26
6.3.1.	Vizualization Results.....	26
6.3.2.	Visualization Script .....	27
6.3.3.	Third Party Packages .....	27
7.	Example .....	27
7.1.	An Example for a Dose-response Model .....	27
8.	Supplementary Information.....	33
9.	References.....	33

# 1. Introduction

---

Food safety risk assessments, control of food production processes as well as the development of new food products are nowadays supported by the application of mathematical modelling and data analysis techniques. Major challenges in that context are the efficient exchange and an easy reusability of knowledge (including analytical data, mathematical models, and simulation results) (Wilkinson et al., 2016). In other words knowledge should become FAIR (findability, accessibility, interoperability, and reusability) (Wilkinson et al., 2016). To ensure an efficient exchange of mathematical models and simulation results, a standardized file format is required. Filter et al. (2016) proposed for that a standardized file format called “Predictive Modelling in Food Markup Language (PMF-ML)” that is compliant to SBML. This format describes in detail how experimental data and mathematical models from the domain of food safety can be saved and encoded in a software-independent manner. A key component of PMF-ML is the option to provide metadata that is essential for an accurate and complete description of the model as well as the provisioning of parameters used for model-based simulations. Here, we present “Food Safety Knowledge Markup Language (FSK-ML)”, an advancement of PMF-ML. FSK-ML allows to define model class-specific annotation schema, i.e. different model classes from the field of food safety modelling can be annotated in a harmonized way. Also, FSK-ML allows to describe models implemented in different script-based programming languages like R, or Python. By that, FSK-ML provides a standardized information exchange format that allows to exchange software dependent models, data, and simulation results in a harmonized manner.

In summary, FSK-ML aims at harmonizing the exchange of (microbial) food safety knowledge (e.g. predictive models) including the corresponding metadata. The standard will allow creating information objects that are not only FAIR, but also ready-to-use. We would like to highlight that there are software tools that allow importing and exporting of models in this format and thus overcoming an error-prone re-implementation process. Examples for such software tools are the KNIME-based FSK-Lab (de Alba Aparicio et al., 2018) and the R-extension FSK2R (presented by Alberto Garre and co-workers on the ICPMF11<https://app.oxfordabstracts.com/events/541/program-app/submission/118286>; R is provided by R Core Team (2019)).

This guidance document is primarily designed for software developers and project managers that want to enhance their software tools with import and export functions for mathematical models, simulations, or data in the domain of food safety. It might also be of interest for those who develop new tools.

## **2. Related Standards**

---

FSK-ML is an open format and based on (1) SBML, (2) OMEX, (3) PMF-ML, and (4) SED-ML. This section provides an overview of the formats and also introduces “Food Safety Knowledge Simulation Experiment Description Markup Language (FSK-SED-ML)”.

### **2.1. Systems Biology Markup Language (SBML)**

---

The Systems Biology Markup Language (SBML) is a representation format for (biological) models that is based on XML, the eXtensible Markup Language. SBML can be used by different software tools and thus facilitates to exchange the model and also ensure that the model can be used even if the software in which the model was created in is no longer available (see Hucka et al. (2015) for details).

### **2.2. Open Modelling EXchange Format (OMEX)**

---

The Open Modelling EXchange format (OMEX), aims to support the exchange of information necessary for modelling and simulating experiments in biology. OMEX defines a OMEX-files (also called COMBINE-container) as a ZIP-file. The ZIP-file includes at least a file with a listing of the container content, the so called “Manifest”. In addition, it is possible for OMEX-files to include any kind of additional data, e.g. files that contain metadata or a mathematical model (see Bergmann et al. (2014) for details).

### **2.3. Predictive Modelling in Food Markup Language (PMF-ML)**

---

“Predictive Modelling in Food Markup Language (PMF-ML) Software Developer Guide” is a standard proposed by Filter et al. (2016). It is based on SBML and incorporates OMEX. This markup language provided for the first time a proposal how to exchange experimental data and mathematical models from the domain of predictive microbiology in a software-independent manner. Also metadata that are relevant for an accurate and complete description of the models are proposed (see Filter et al. (2016) for details).

## **2.4. Food Safety Knowledge Markup Language (FSK-ML)**

---

Food Safety Knowledge Markup Language (FSK-ML) is an open format to describe models, empirical data, and simulation results in the domain of (microbial) food safety. The markup language also describes how to encode combined models and how other model-related information (e.g. simulation results, libraries, visualization scripts etc.) can be included.

## **2.5. Food Safety Knowledge Simulation Experiment Description Markup Language (FSK-SED-ML)**

---

Food Safety Knowledge Simulation Experiment Description Markup Language (FSK-SED-ML) extends the Simulation Experiment Description Markup Language (SED-ML) to describe the simulation settings for FSK-ML compliant models. SED-ML encodes simulation descriptions on computational models of biological systems (see Bergmann, Cooper, Le Novère, Nickerson, and Waltemath (2015) for details). FSK-SED-ML is explained in detail in the Supplementary Information.

## 3. Document Conventions

---

In this section, the terminology as well as the typographical conventions for this document are presented.

### 3.1. Terminology

---

To prevent misunderstanding, relevant terms are defined in the following:

#### Generic Metadata Schema

---

The complete set of metadata concepts that allows annotating food safety models or data (definition is adapted from Haberbeck et al. (2018)).

#### Metadata

---

Data that defines and describes other data (ISO (International Organisation for Standardisation), 2004) (definition is taken from Haberbeck et al. (2018)).

#### Model Class

---

Classification of mathematical models according to their purpose, e.g. QMRA model, dose-response model, process model, consumption model, exposure model, health metric model, predictive microbial model and other empirical model. The current classification is rooted mainly on the definition of risk assessment provided by Codex Alimentarius (Codex Alimentarius Commission, 1999).

#### Simulation

---

Computer simulation is the reproduction of the behaviour of a system using a computer to simulate the outcomes of a mathematical model that represent the said system (based on Wikimedia Foundation Inc. (2019)).

Note: There is a difference between a (mathematical) model and a (model-based) simulation. A (mathematical) model is composed of the algorithms and equations used to mimic the modelled system. By contrast, a (model-based) simulation is the process of running a (mathematical) model to make a prediction on the system's behaviour (based on Wikimedia Foundation Inc. (2019)).

## 3.1. Implementation Convention

---

In this section, the implementation conventions for the markup language are defined.

### 3.1.1. Type SId

---

In SBML there is the so called SId type; FSK-ML also uses this type for certain metadata related to SBML and SED-ML. SId type restricts the characters permitted and the sequences in which those characters are allowed to appear. The definition is shown in Figure 1 (for more details see e.g. Hucka et al. (2015)).

```
letter ::= 'a'..'z','A'..'Z'
digit  ::= '0'..'9'
idChar ::= letter | digit | '_'
SId    ::= ( letter | '_' ) idChar*
```

Figure 1 Definition of the type SId (the figure is taken from Hucka et al., 2015).

### 3.1.2. Script Language Specificity

---

In FSK-ML, mathematical models and visualisation scripts can be script language-specific, i.e. these scripts can be provided e.g. in R or Python. For simplicity, examples presented in the following are always based on the R scripting language.

### 3.1.3. Referencing Script Code

---

In FSK-ML, there are two ways to refer to information that is provided as script code, e.g. a mathematical model implemented in R. First, the script code can be provided directly. Alternatively, the path to a script file can be provided (see Supplementary Information for details).



## 3.2. Typographical Conventions

---

This document uses the typographical conventions defined in the OMEX specification document (Bergmann et al., 2014). Names of objects, classes, and data types are highlighted as follows:

**Class:** Names of ordinary (concrete) classes begin with a capital letter and are printed in an upright, bold, and sans-serif typeface. In electronic document formats, the class names defined within this document are also hyperlinked to their definitions; clicking on these items will, given appropriate software, switch the view to the section in this document containing the definition of the class. In this document, class names are hyperlinked to the definition in this document only if they are changed from their definitions in the SED-ML Level 1 Version 2 specification (Bergmann et al., 2015).

**AbstractClass:** Abstract classes serve as parents of other classes. Their names begin with a capital letter and they are printed in a slanted, bold, and sans-serif typeface. In electronic document formats, the class names defined within this document are also hyperlinked to their definitions; clicking on these items will, given appropriate software, switch the view to the corresponding section in this document. Note that, for classes that are defined in SED-ML Level 1 Version 2 (Bergmann et al., 2015), the class names are not hyperlinked because they are not defined within this document.

**OtherAttributes:** Attributes of classes, data type names, literal XML, and tokens are printed in an upright typewriter typeface. Note that this convention is not meant for class names.

## 4. FSK-ML Specification: Single Model

In this chapter the core specifications of FSK-ML are presented focusing on single models. The specifications for data and the combination of multiple single models, so called joined models, are presented in Chapters 5 and 6, respectively.

The FSK-ML format describes how relevant information should be provided, i.e. what files are needed or possible to describe content relevant for food safety modelling. We distinguish the following purpose types:

- (i) file(s) containing input information, i.e. this needs to be provided by the model designer/creator,
- (e) file(s) containing information that is relevant for the execution of the model, and
- (u) file(s) containing information that supports the user, but is not relevant for the model execution.

Table 1 lists files all files that are recommended to be included to comprehensively describe a mathematical model. In order to exchange all files as one information object, a so-called FSKX-container format is proposed. An FSKX-container is a ZIP-file containing at least one file, the mandatory “manifest.xml”-file. The “manifest.xml”-file lists all files inside of the FSKX-container, i.e. it can include any number of files (see Section 4.1.1 for details). We recommend the file extension “.fskx” for the FSKX-container. Figure 2 shows an example file structure of an FSK-ML compliant model file.

**Table 1** A listing of (mandatory and recommended) files that can be included into an FSKX-container. For each file the description, example file names, information about whether or not a file is mandatory, and where to find further details are given.

	Description	Example file name with file extension that are supported	Is the file mandatory?	Purpose of the file*	Further details
Folder structure related files	List of all files in the FSKX-container	manifest.xml	yes	e	See Section 4.1.1
	Additional information about the files in the FSKX-container	metadata.rdf	yes	e	See Section 4.1.2
	Version control for software and software packages	packages.json	no	u	See Section 4.1.3

Model related files	Model script	model.r model.py model.php	yes	i	See Section 4.2.1
	Metadata about the used model	metadata.json metadata.pmf	yes	i	See Section 4.2.2
	Model annotation template	modelann.xlsx	no	u	See Section 4.2.2
	Supplements to the model, e.g. the paper where the model is published	description.pdf	no	u	
	Model related data	experiment.xlsx experiment.csv	no	i	See Section 4.2.3
	Third party packages	pack_0.10.zip pack_0.10.tgz pack_0.10.tgz-gz	no	e	See Section 4.2.4
Simulation related files	Simulation settings	sim_1.sedml	yes	i	See Section 4.3.1
	SBML-file that contains the simulation settings	model.sbml	recommended	e	See Section 4.3.1
	Simulation results	sim_1_res.r sim_1_res.py sim_1_res.csv	no	u	See Section 4.3.2
	Visualization script	visualization_1.r visualization_1.py	no	i	See Section 4.3.3
	Plots of the simulated results	plot.png, plot.bmp plot.tiff	no	u	See Section 4.3.2
Other	Read me	README.txt	no	u	

\*file contains input (i), file is relevant for the execution of the model (e), or file supports users (u)

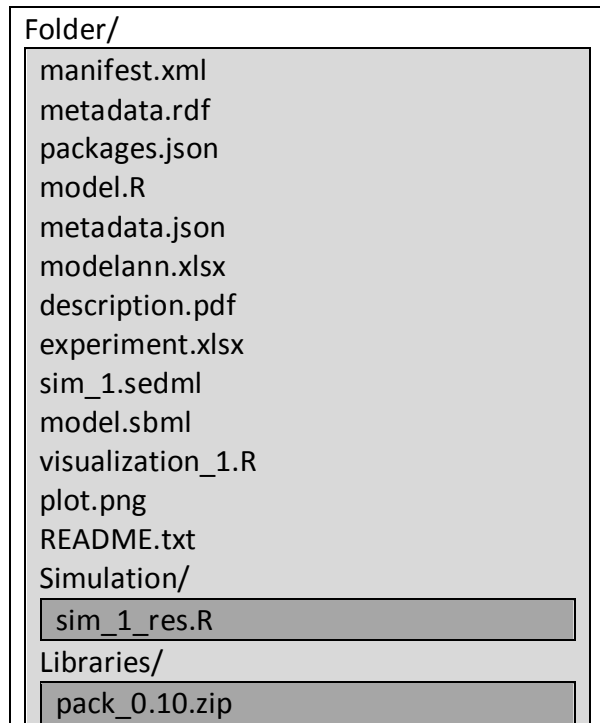


Figure 2 Example for a folder structure of an FSK-ML compliant model.

## 4.1. Container Structure Related Files

In this section the files that describe the structure and content of the FSKX-container are elucidated.

### 4.1.1. Manifest

The manifest file lists all files inside the FSKX-container (*manifest.xml* in Figure 2 and see Figure 3 for an example). That means that a valid manifest file needs to have at least one entry, declaring the container itself.

```

<?xml version="1.0" encoding="UTF-8"?>
<omexManifest
  xmlns="http://identifiers.org/combine.specifications/omex-manifest">
  <content location="."
    format="http://identifiers.org/combine.specifications/omex" />
  <content location="./manifest.xml"
    format="http://identifiers.org/combine.specifications/omex-manifest" />
  <content location="./visualization.r"
    format="http://purl.org/net/mediatypes/application/text/x-r" />
  <content location="./metaData.pmf"
    format="http://purl.org/net/mediatypes/application/application/sbml+xml" />
  <content location="./workspace.r"
    format="http://purl.org/net/mediatypes/application/text/x-r" />
  <content location="./triangle_0.10.zip"
    format="http://purl.org/net/mediatypes/application/application/zip" />

```

```

<content location="./model.r"
  format="http://purl.org/net/mediatypes/application/text/x-r" />
<content location="./param.r"
  format="http://purl.org/net/mediatypes/application/text/x-r" />
<content location="./metadata.rdf"
  format="http://identifiers.org/combine.specifications/omex-metadata" />
</omexManifest>

```

**Figure 3** Example for a manifest file in XML.

The manifest file is an XML-file and is located at the root of the container. This file contains an instantiation of the *OmexManifest* class (already defined in Bergmann et al. (2015)). For each file that is in the container there is a URI given to specify the file type. This specification is done by URIs through the Internet media types, which is previously known as MIME type (Freed & Borenstein, 1996). Table 2 shows a list of relevant file types and associated Internet media types.

**Table 2** Relevant file types and corresponding Internet media types that can be used in the manifest file.

File type	Internet media type
Zip	<a href="http://purl.org/NET/mediatypes/application/zip">http://purl.org/NET/mediatypes/application/zip</a>
Tgz	<a href="http://purl.org/NET/mediatypes/application/x-tgz">http://purl.org/NET/mediatypes/application/x-tgz</a>
Tag-gz	<a href="http://purl.org/NET/mediatypes/application/x-tar.gz">http://purl.org/NET/mediatypes/application/x-tar.gz</a>
R	<a href="http://purl.org/NET/mediatypes/application/r">http://purl.org/NET/mediatypes/application/r</a>
Python	<a href="http://purl.org/NET/mediatypes/application/python">http://purl.org/NET/mediatypes/application/python</a>
PMF	<a href="http://purl.org/NET/mediatypes/application/x-pmf">http://purl.org/NET/mediatypes/application/x-pmf</a>
SBML	<a href="http://purl.org/NET/mediatypes/application/sbml+xml">http://purl.org/NET/mediatypes/application/sbml+xml</a>
JSON	<a href="https://www.iana.org/assignments/media-types/application/json">https://www.iana.org/assignments/media-types/application/json</a>
Matlab	<a href="http://purl.org/NET/mediatypes/text/x-matlab">http://purl.org/NET/mediatypes/text/x-matlab</a>
PHP	<a href="http://purl.org/NET/mediatypes/text/x-php">http://purl.org/NET/mediatypes/text/x-php</a>
Plain text	<a href="http://purl.org/NET/mediatypes/text-xplain">http://purl.org/NET/mediatypes/text-xplain</a>
R workspace	<a href="http://purl.org/NET/mediatypes/text/x-RData">http://purl.org/NET/mediatypes/text/x-RData</a>
CSV	<a href="https://www.iana.org/assignments/media-types/text/csv">https://www.iana.org/assignments/media-types/text/csv</a>
Sedml	<a href="http://identifiers.org/combine.specifications/sed-ml">http://identifiers.org/combine.specifications/sed-ml</a>
XLSX	<a href="https://www.iana.org/assignments/media-types/application/vnd.ms-excel">https://www.iana.org/assignments/media-types/application/vnd.ms-excel</a>
BMP	<a href="https://www.iana.org/assignments/media-types/image/bmp">https://www.iana.org/assignments/media-types/image/bmp</a>
JPEG	<a href="https://www.iana.org/assignments/media-types/image/jpeg">https://www.iana.org/assignments/media-types/image/jpeg</a>
TIFF	<a href="https://www.iana.org/assignments/media-types/image/tiff">https://www.iana.org/assignments/media-types/image/tiff</a>
PNG	<a href="http://purl.org/NET/mediatypes/image/png">http://purl.org/NET/mediatypes/image/png</a>

### 4.1.2. Metadata File

The metadata file is needed to further specify files that contain script code. Specifically it is needed to distinguish the files holding the mathematical model, the visualization script, and the storage of simulation results from one another. The specification has to be done in one out of

two ways (see Section 3.1.3 for details). Note that the simulation scenarios and the corresponding parameters are included in the FSK-SED-ML-file (see Section 4.3.1 for details).

The metadata file uses the Resource Description Format (RDF; *metadata.rdf* in Figure 2; see Cyganiak et al. (2014) for details about RDF). The version of the RDF-file is described by the conformsTo-property from the Dublin Core Metadata Initiative (see Figure 4 for an example and see <https://www.dublincore.org/> for further details). The RDF-annotation involves RDF description elements that specify the type and location of the resource with Dublin Core type and source elements, respectively.

When code is provided in the programming language R, the following types are common:

- `modelScript`: filename of the model script
- `visualizationScript`: filename of the visualization script
- `workspace`: filename of the workspace with the simulation results

```
<?xml version="1.0" encoding="UTF-8"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:dcterms="http://purl.org/dc/terms/"
  xmlns:vCard="http://www.w3.org/2006/vcard/ns#">

  <rdf:Description rdf:about=".">
    <dcterms:conformsTo>2.0</dcterms:conformsTo>
  </rdf:Description>

  <rdf:Description rdf:about="/model.r">
    <dc:type>mainScript</dc:type>
  </rdf:Description>

  <rdf:Description rdf:about="/visualization.r">
    <dc:type>visualizationScript</dc:type>
  </rdf:Description>

  <rdf:Description rdf:about="/workspace.r">
    <dc:type>workspace</dc:type>
  </rdf:Description>
</rdf:RDF>
```

Figure 4 Example for a metadata file in RDF.

### 4.1.3. Version Control

The versions of the software and software packages used to run and visualize the model are listed in a JSON-file (*packages.json* in Figure 2). This file is located at the root directory of the FSKX-file. In the example presented in Figure 5, R 3 is used and the two packages “triangle” and “ggplot2” are used in Version 3.1.1 and 0.12, respectively. The purpose of this file is to inform the user, rather than interpreting it when executing the scripts.

```
{
  "language": "R 3",
  "packages": {
    "triangle": "3.1.1",
    "ggplot2": "0.12"
  }
}
```

Figure 5 Example for a JSON-file that describes the versioning in an FSKX-file.

## 4.2. Model Related Files

This section specifies the files that are related to the presented mathematical model.

### 4.2.1. Model Script

The model script is a script stored within the FSKX-container that includes the model code and packages that are required to run the model.

### 4.2.2. Model Metadata

The model metadata contains all metadata that is linked to the model (*metaData.json* in Figure 2). The model metadata follows a specific schema that depends on the model classes. Table 3 lists the currently defined metadata schema, a description of the model class, and the corresponding link that shows the model-specific template, the so called “Model Annotation Excel Template” (*modelann.xlsx* in Figure 2). This template follows a predefined set of metadata, a so called metadata schema. These schemes are provided as a community driven, online resource and thus are subject to regular improvements (see <https://goo.gl/PE4ysP> for interactive list). Note that the generic model is able to represent any model classes.

Table 3 List of currently supported model classes and the link of the “Model Annotation Excel Template” (effective November 2019).

Model class	Description	Link
Consumption model	A consumption model describes the amount of food consumed during a particular eating occasion (i.e., a serving) and/or the frequency of the consumption of these servings, or an average amount of food consumed per day. This amount may vary in time, between individuals, between the different population groups of interest and the considered exposure type (definition is	<a href="https://docs.google.com/spreadsheets/d/1R5ofJAlqAyyWN97cnBY9R9kqj10AvQOs3Gq472n6iDyc/edit#gid=1448063781">https://docs.google.com/spreadsheets/d/1R5ofJAlqAyyWN97cnBY9R9kqj10AvQOs3Gq472n6iDyc/edit#gid=1448063781</a>

	taken from Haberbeck et al. (2018)).	
Data	Symbolic representation of observable properties of the world (definition is taken from Haberbeck et al. (2018)). See Chapter 6 for details.	<a href="https://docs.google.com/spreadsheets/d/1R5ofJAlqAywN97cnBY9R9kqj10AvQOs3Gq472n6iDyc/edit#gid=1243174041">https://docs.google.com/spreadsheets/d/1R5ofJAlqAywN97cnBY9R9kqj10AvQOs3Gq472n6iDyc/edit#gid=1243174041</a>
Dose-response model	Model describing the “relationship between the magnitude of exposure (dose) to a hazard and the severity and/or frequency of associated adverse effects (response)” (Codex Alimentarius Commission, 1999; FAO/WHO, 2016) (definition is taken from Haberbeck et al. (2018)).	<a href="https://docs.google.com/spreadsheets/d/1R5ofJAlqAywN97cnBY9R9kqj10AvQOs3Gq472n6iDyc/edit#gid=2104438389">https://docs.google.com/spreadsheets/d/1R5ofJAlqAywN97cnBY9R9kqj10AvQOs3Gq472n6iDyc/edit#gid=2104438389</a>
Exposure model	A combination of the process model and the consumption model that results in the exposure assessment (definition is taken from Haberbeck et al. (2018)).	<a href="https://docs.google.com/spreadsheets/d/1R5ofJAlqAywN97cnBY9R9kqj10AvQOs3Gq472n6iDyc/edit#gid=1823080546">https://docs.google.com/spreadsheets/d/1R5ofJAlqAywN97cnBY9R9kqj10AvQOs3Gq472n6iDyc/edit#gid=1823080546</a>
Generic model	A model that represent all model classes.	<a href="https://docs.google.com/spreadsheets/d/1R5ofJAlqAywN97cnBY9R9kqj10AvQOs3Gq472n6iDyc/edit#gid=811823962">https://docs.google.com/spreadsheets/d/1R5ofJAlqAywN97cnBY9R9kqj10AvQOs3Gq472n6iDyc/edit#gid=811823962</a>
Health metrics model	Model for calculating a measure for assessing the health impact of a specific hazard in a population group: e.g. Disability-adjusted life years (DALYs) or cost per illness (definition is taken from Haberbeck et al. (2018)).	<a href="https://docs.google.com/spreadsheets/d/1R5ofJAlqAywN97cnBY9R9kqj10AvQOs3Gq472n6iDyc/edit#gid=637680402">https://docs.google.com/spreadsheets/d/1R5ofJAlqAywN97cnBY9R9kqj10AvQOs3Gq472n6iDyc/edit#gid=637680402</a>
Other empirical models	A generic model that describes a set of data in a convenient mathematical relationship without considering any underlying phenomena (definition is based on Haberbeck et al. (2018)).	<a href="https://docs.google.com/spreadsheets/d/1R5ofJAlqAywN97cnBY9R9kqj10AvQOs3Gq472n6iDyc/edit#gid=645002684">https://docs.google.com/spreadsheets/d/1R5ofJAlqAywN97cnBY9R9kqj10AvQOs3Gq472n6iDyc/edit#gid=645002684</a>
Predictive model	Models describing the microbial responses towards environmental conditions, such as storage and processing conditions and product characteristics. Traditionally, models in predictive microbiology are classified as primary and secondary (Whiting, 1993) (definition is taken from Haberbeck et al. (2018)). For further details see Haberbeck et al. (2018).	<a href="https://docs.google.com/spreadsheets/d/1R5ofJAlqAywN97cnBY9R9kqj10AvQOs3Gq472n6iDyc/edit#gid=1127734929">https://docs.google.com/spreadsheets/d/1R5ofJAlqAywN97cnBY9R9kqj10AvQOs3Gq472n6iDyc/edit#gid=1127734929</a>
Process model	Model that describes how the concentrations of the hazard change along the different steps (modules) of the food production chain (potentially from farm to	<a href="https://docs.google.com/spreadsheets/d/1R5ofJAlqAywN97cnBY9R9kqj10AvQOs3Gq472n6iDyc/edit#gid=115271917">https://docs.google.com/spreadsheets/d/1R5ofJAlqAywN97cnBY9R9kqj10AvQOs3Gq472n6iDyc/edit#gid=115271917</a>



	fork) (definition is taken from Haberbeck et al. (2018)).	
Quantitative risk assessment (QRA) model	The quantitative modelling of a scientifically based process consisting of the following steps: (i) hazard identification, (ii) hazard characterization, (iii) exposure assessment, and (iv) risk characterization (Codex Alimentarius Commission, 1999; FAO/WHO, 2016) (definition is based Haberbeck et al. (2018)).	<a href="https://docs.google.com/spreadsheets/d/1R5ofJAlqAywN97cnBY9R9kqj10AvQOs3Gq472n6iDyc/edit#gid=1679055327">https://docs.google.com/spreadsheets/d/1R5ofJAlqAywN97cnBY9R9kqj10AvQOs3Gq472n6iDyc/edit#gid=1679055327</a>
Risk characterization model	Combination of health metrics model, dose-response model and exposure model within the framework of the risk characterization (definition is taken from Haberbeck et al. (2018)).	<a href="https://docs.google.com/spreadsheets/d/1R5ofJAlqAywN97cnBY9R9kqj10AvQOs3Gq472n6iDyc/edit#gid=173959030">https://docs.google.com/spreadsheets/d/1R5ofJAlqAywN97cnBY9R9kqj10AvQOs3Gq472n6iDyc/edit#gid=173959030</a>
Toxicological reference value model	Modelling a value, that when compared with exposure, is used to estimate the likelihood and severity of an adverse effect which could occur in a given population. (based on <a href="https://docs.google.com/spreadsheets/d/1akQ8h9xvlisJrZZSy5QOQufSV17Ac_CDT2umLU-d3IM/edit#gid=0">https://docs.google.com/spreadsheets/d/1akQ8h9xvlisJrZZSy5QOQufSV17Ac_CDT2umLU-d3IM/edit#gid=0</a> ).	<a href="https://docs.google.com/spreadsheets/d/1R5ofJAlqAywN97cnBY9R9kqj10AvQOs3Gq472n6iDyc/edit#gid=1437089839">https://docs.google.com/spreadsheets/d/1R5ofJAlqAywN97cnBY9R9kqj10AvQOs3Gq472n6iDyc/edit#gid=1437089839</a>

As described in Haberbeck et al. (2018) the metadata schema is a hierarchical structure with four top level elements: (1) general information, (2) scope, (3) data background, and (4) mathematical model and data (so called “Model math/Data definition”). For details see the paper or the online version of the Generic Metadata Schema available at <https://goo.gl/PE4ysP>.

## Model Annotation Excel Template

To support the creation of models from scratch, e.g. using tools like the KNIME-based FSK-Lab (de Alba Aparicio et al., 2018), it is possible to use the “Model Annotation Excel Template” (<https://goo.gl/PE4ysP>). All relevant metadata on the model metadata could be filled into the template and then this file could be imported by tools supporting this feature.

## Controlled Vocabularies

FSK-ML recommends to use controlled vocabularies for relevant metadata (see Supplementary Information for a list or <https://docs.google.com/spreadsheets/d/1C6N4-YWX9OMmNStd2rYISUaVys-aiJGLj00cD44aVc8/edit#gid=1479548673> for the interactive list). The vocabularies are based on the terms used by existing ontologies, standards, and tools (for more information see <https://foodrisklabs.bfr.bund.de/rakip-harmonization-resources/>).

## Supported Metadata Types

To enable interoperability of provided metadata, FSK-ML requires that metadata itself must comply with certain data types, i.e. any model metadata has to adhere to one of the data types presented in Table 4. The supported data types are case-insensitive.

Data type	Code vocabulary	Description
Numeric	numeric	Real numbers
Integer	integer	Integers
Character	character	String values
Vector	vector	One-dimensional array
Matrix	matrix	Two-dimensional array

Table 4 Data types supported by FSK-ML.

### 4.2.3. Model Related Data

Data files can be included into the FSKX-file. Those data might be input for the model, data that was used for model generation (e.g. through fitting), or for model validation (*experiment.xlsx* in Figure 2).

### 4.2.4. Third Party Packages

The FSKX-file contains detailed information about the packages required for model simulation and visualization. It is advisable, but not mandatory, to also include the packages that show dependencies to the listed packages. Although the placement of the libraries in the FSKX-file is free, it is advisable to save them into a separated “Libraries”-folder (*pack\_0.10.zip* in Figure 2). Only the packages that correspond to the used system are saved, i.e. compressed tarballs for Linux etc. In the case of an R-model, the corresponding packages would be binary packages.

## 4.3. Simulation Related Files

FSK-ML not only provides an opportunity to describe script-based models, it also enables to define, store, and exchange settings for model-based simulations. All relevant settings, e.g. for model input parameters, are stored in FSK-SED-ML format (see Section 4.3.1 for details). The simulation results for each simulation scenario can be saved and visualized (see Section 4.3.2 for details about storage and Section 4.3.3 for details about the visualization). The FSK-SED-ML format is described in more detail in the Supplementary Information.

### 4.3.1. Simulation Settings

---

The FSK-SED-ML-file contains simulation settings, i.e. the parameter set (including input parameters and variables) and relevant metadata (*sim\_1.sedml* in Figure 2). This file contains at least the default parameter set and also all the parameter sets for all scenarios defined by the user. In addition to the FSK-SED-ML-files, an SBML-file contains the default parameters for user convenience (see Table 1 and *model.sbml* in Figure 2). This file is recommended, even so it is for user support only, because it ensures compatibility between single and joined models (see Chapter 5 for details about the joined models). Note that the order of parameter sets presented in the FSK-SED-ML-file is the one considered for running simulations.

### 4.3.2. Simulation Results

---

The results for each simulation can on the one hand be stored in form of figures and on the other hand in form of files. If the model is in R, simulation results can also be saved as an R-workspace (*sim\_1\_res.r* in Figure 2). Common formats in which generated plots can be saved are JPEG, PNG, and SVG-files (*plot.png* in Figure 2). Although the placement of simulation results within the FSKX-container in the file is free, it is advisable to save them in a separated “Simulation”-folder.

### 4.3.3. Visualization Scripts

---

Visualization scripts create plots from the simulation results (*visualization\_1.r* in Figure 2). Plots might be a composition of multiple sub-figures. The script has to be in the same scripting language as the corresponding model. See Section 4.3.2 for details about the storage of simulation results.

## 5. FSK-ML Specification: Joined Model

---

FSK-ML can describe joined models where the output of one model (donor model) is used as an input of a second model (receiver model).

An FSK-ML compliant joined model is presented in multiple files; this is consistent with the single model structure (see Chapter 4). Identical to the single model (Table 1), we distinguish three purposes for the files: (i) the file contains input, (e) the file contains information that is relevant for the execution of the model, and (u) the file contains information that support the user. Table 5 lists mandatory (Purpose i and e) and recommended (Purpose u) files for the

joined model. For each file, the description, example filenames, and information about whether or not a file is mandatory are presented.

The way how models are connected is specified in the SBML-file, which is located at the root directory of the FSKX-file (see Section 5.1.1). FSKX-files can contain a number of nested joined models. Figure 6 represents the folder structure of an FSK-ML compliant joined model.

**Table 5** A list of (mandatory and recommended) files that can be in the FSKX-container of a joined model. For each file the description, example file names, and information about whether or not a file is mandatory is given.

	Description	Example file name with file extension that are supported	Is the file mandatory?	Purpose of the file*	Further details
Joining related files	The SBML-file that describes the joining of the models	joined_model.sbml	yes	i	See Section 5.1.1
	Metadata about the receiver model	metadata.json metadata.pmf	yes	i	See Section 5.1.2
	Simulation settings for the joined model	sim_1.sedml	yes	i	See Section 5.1.3
Container structure related files	List of all files in the FSKX-container	manifest.xml	yes	e	See Section 5.2.1
	Additional information about the files in the FSKX-container	metadata.rdf	yes	e	See Section 5.2.2
	Version control for software and software packages	packages.json	no	u	See Section 5.2.3
Single model related files	A folder that equals the one for the single model (except for the manifest (.xml), the metadata container (.rdf), and the version control (.json) which are not included) (see Table 1 for the single model elements)	Folder/	yes	e	See Section 5.3

\*file contains input (i), file is relevant for the execution of the model (e), or file supports user (u)

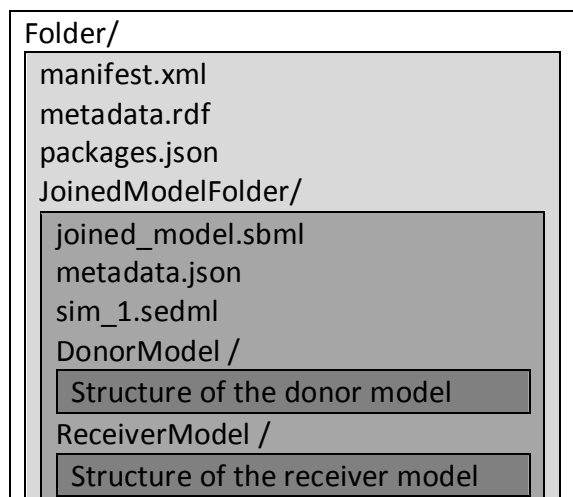


Figure 6 Example folder structure of an FSK-ML compliant model that is joined from two original models.

## 5.1. Joining Related Files

In this section, the files that describe how two models are combined are presented.

### 5.1.1. Joined Model Script

The information how to join the models is provided as SBML-file (*joined\_model.sbml* in Figure 6). In this file, the receiver and the donor model are specified (*Model1.sbml* and *Model2.sbml* in Figure 7) as well as the connection between both models (the value of *SecondModelInput* is set to *FirstModelOutput* in Figure 7).

```

<?xml version='1.0' encoding='UTF-8' standalone='no'?>
<sbml xmlns="http://www.sbml.org/sbml/level3/version1/core"
xmlns:fsk="https://foodrisklabs.bfr.bund.de/wp-content/uploads/2017/01/FSK-
ML_guidance_document_021216.pdf" comp:required="true"
xmlns:comp="http://www.sbml.org/sbml/level3/version1/comp/version1" level="3"
version="1">
  <comp:listOfExternalModelDefinitions
xmlns:comp="http://www.sbml.org/sbml/level3/version1/comp/version1">
    <comp:externalModelDefinition comp:id="Model1" comp:source="Model\Model1.sbml"/>
    <comp:externalModelDefinition comp:id="Model2" comp:source="Model\Model2.sbml"/>
  </comp:listOfExternalModelDefinitions>
  <model id="Model2">
    <comp:listOfSubmodels
xmlns:comp="http://www.sbml.org/sbml/level3/version1/comp/version1">
      <comp:submodel comp:id="submodel1" comp:modelRef="Model1"/>
      <comp:submodel comp:id="submodel2" comp:modelRef="Model2"/>
    </comp:listOfSubmodels>
    <listOfParameters>
      <parameter constant="false" id="SecondModelInput">
        <annotation>
          <fsk:command commandValue="FirstModelOutput"/>
        </annotation>
        <comp:replacedBy

```

```
xmlns:comp="http://www.sbml.org/sbml/level3/version1/comp/version1"
comp:idRef="FirstModelOutput" comp:submodelRef="submodel1"/>
  </parameter>
</listOfParameters>
</model>
</sbml>
```

Figure 7 Example for the SBML-file that describes the joining of two models.

## 5.1.2. Model Metadata

---

The metadata file contains all metadata that are linked to the mathematical models of the receiver model. As the model metadata follow a specific schema, the generic schema is used to provide all relevant metadata (*metadata.json* in Figure 6; see Section 4.2.2 for further details).

## 5.1.3. Simulation Settings

---

The FSK-SED-ML-files contains simulation settings (see Section 4.3.1 for details).

## 5.2. Container Structure Related Files

---

In this section, the files that describe the structure and content of the FSKX-container are elucidated in more detail.

### 5.2.1. Manifest

---

The manifest file lists all files inside the FSKX-container (see Section 4.1.1 for details).

### 5.2.2. Metadata File

---

The metadata file specifies the mathematical model, the storage of simulation results and their visualization (see Section 4.1.2 for details).

### 5.2.3. Version Control

---

The versions of the software and software packages used to run and visualize the model are listed in a JSON-file (see Section 4.1.3 for details).

## 5.3. Single Model Related Files

The files that describe the receiver and donor model equal the ones for the single model presented in Table 1 (except for the manifest (.xml), the metadata container (.rdf), and the version control (.json) which are not included) (see Table 1 for the single model elements and see Chapter 4 for details).

## 6. FSK-ML Specification: Data

This section specifies the storage of data and data sets in the domain of (microbial) food safety. The description focuses on data that result from model simulation, but it is assumed that it is applicable to other data, too. In order to combine all files relevant to describe data, an FSKX-container is created (see Section 4.1 for details about the FSKX-container, see Table 6 for a list of files, and see Figure 8 for an example). We distinguish three purposes for the files listed in Table 6. Mandatory files are the files that contain input and the files that contain information that is relevant for the execution of the model (Purpose i and e in Table 6, respectively). Recommended file contain information that supports the user (Purpose u in Table 6).

**Table 6** A listing of (mandatory and recommended) files that can be included into an FSKX-container. For each file the description, example file names, information about whether or not a file is mandatory, and where to find further details are given.

	Description	Example file name with file extension that are supported	Is the file mandatory?	Purpose of the file*	Further details
Folder structure related files	List of all files inside the FSKX-container	manifest.xml	yes	e	See Section 6.1.1
	Additional information about the files in the FSKX-container	metadata.rdf	yes	e	See Section 6.1.2
	Version control for software and software	packages.json	no	u	See Section 6.1.3
Data related file	The file contains the following information: 1. Data set 2. Metadata about the used data set 3. Hierarchical data	dataset.hdf5 dataset.h5	yes	i	See Section 6.2.1

Visualization related files	structure for managing multiple data sets				
	Metadata about the data	metadata.json metadata.pmf	yes	i	See Section 6.2.2
	Supplements to the data, e.g. the paper where the data is published	description.pdf	no	i	
	Visualization results	viz_1_res.r viz_1_res.py viz_1_res.csv	no	u	See Section 6.3.1
	Visualization script	visualization_1.r visualization_1.py	no	i	See Section 6.3.2
	Plots of the processed data	plot.png, plot.bmp plot.svg	no	u	See Section 6.3.2
Other	Third party packages	pack_0.10.zip pack_0.10.tgz pack_0.10.tgz-gz	no	u	See Section 6.3.3
	Read me	README.txt	no	u	

\*file contains input (i), file is relevant for understanding of the data set(s) (e), or file supports user (u)



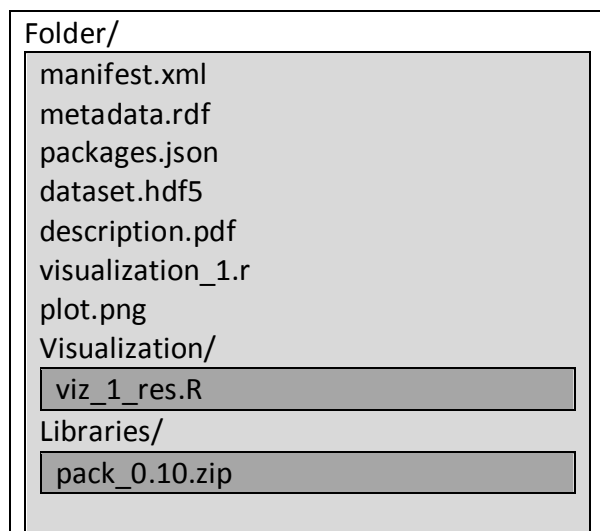


Figure 8 Example for the folder structure of FSK-ML compliant data set.

## 6.1. Container Structure Related Files

---

In this section the files that describe the structure and content of the FSKX-container are elucidated in more detail.

### 6.1.1. Manifest

---

The manifest file lists all files inside the FSKX-container (see Section 4.1.1 for details).

### 6.1.2. Metadata File

---

The RDF-metadata file identifies a visualization script if this is provided (see Section 4.1.2 for details).

### 6.1.3. Version Control

---

The versions of the software and software packages used to visualize the data are listed in a JSON-file (see Section 4.1.3 for details).

## 6.2. Data Related Files

---

In this section the file that stores the data and its attributes are presented in more detail.

### 6.2.1. Data Storage

---

Data in form of tables, multidimensional arrays, and multiple data sets can be stored in the Hierarchical Data Format (HDF) that is a file format designed to store and to organize large amounts of data (see <http://portal.hdfgroup.org/display/support> for further details about HDF5; *dataset.hdf5* in Figure 8). As it is possible to store several data sets in one file, FSK-ML allows a maximum of one HDF5 file per FSKX container.

The HDF5-format is a truly hierarchical, filesystem-like data format. The information stored in this file can be accessed independently which results in a minimal computational effort. This minimal effort facilitates an efficient file access, which is particularly important when big data are considered. The HDF5-format comprises data sets and groups. Data sets are multidimensional arrays and groups are container structures that include data sets and other groups (if multiple data sets are stored) for organizational purposes. Both, data sets and groups, can have metadata attached. The metadata represent core properties that are required for an accurate and consistent data description. Metadata is stored in the form of user-defined, named attributes and follows a predefined metadata schema, the so called DataCite Metadata Schema (see DataCite Metadata Working Group (2019) for details). In FSK-ML, the information listed in the predefined metadata schema should be included automatically. Examples for metadata are column and row names of a table (see Supplementary Information for the full metadata schema).

### 6.2.2. Data Metadata

---

The metadata file contains all metadata that are relevant to explain the data. As the metadata follow a specific schema, the data schema is used to provide all relevant metadata (*metadata.json* in Figure 8; see Section 4.2.2 for further details).

## 6.3. Visualization

---

To visualize data, the user can provide a script that specifies which data are visualized and in which way.

### 6.3.1. Visualization Results

---

The visualized data can be stored as a figure and/or in a file (*viz1\_res.r* and *plot.png* in Figure 8). See Section 4.3.2 for details.

## 6.3.2. Visualization Script

---

Visualization scripts create plots that represent data (*visualization\_1.r* in Figure 8). If necessary, these scripts can contain code to process the data, in order to be able to visualize it. See Section 4.3.3 for details.

## 6.3.3. Third Party Packages

---

The FSKX-file contains detailed information about the packages required for data visualization (see Section 4.2.4 for details).

# 7. Example

---

## 7.1. An Example for a Dose-response Model

---

The presented example describes a dose-response model that predicts the effects of food preparation by the consumer on the survival and transfer of *Campylobacter* (the model is taken from Nauta, Jacobs-Reitsma, and Havelaar (2007); the FSKX-model is available via the VRE [http://data.d4science.org/ctlg/RAKIP\\_portal/c6057928-7c1b-4d6a-9adc-d1b8859f3418](http://data.d4science.org/ctlg/RAKIP_portal/c6057928-7c1b-4d6a-9adc-d1b8859f3418) or <http://data.d4science.org/workspace-explorer-app?folderId=ekYyTitlNjBiRkg4bWRoc25aU3hwZ2Y2TzZ3bThzempaaG94SldNaDhzakJGZFA3MctwalpQOFJkK1pSVzdDQQ>). The input of the model is the predicted distribution of *Campylobacters* on chicken breast fillets purchased at retail; the prediction is based on a food chain model. The model predicts a mean probability of illness per meal.

The Supplementary Information “metadata.xlsx” describes the metadata schema of the model and Figure 9 the JSON-metadata file. For the whole example of the FSK-ML compliant model that was created using FSK-Lab (de Alba Aparicio et al., 2018) (the FSKX-model is available via the VRE [http://data.d4science.org/ctlg/RAKIP\\_portal/c6057928-7c1b-4d6a-9adc-d1b8859f3418](http://data.d4science.org/ctlg/RAKIP_portal/c6057928-7c1b-4d6a-9adc-d1b8859f3418) or <http://data.d4science.org/workspace-explorer-app?folderId=ekYyTitlNjBiRkg4bWRoc25aU3hwZ2Y2TzZ3bThzempaaG94SldNaDhzakJGZFA3MctwalpQOFJkK1pSVzdDQQ>).

```
{  
  "modelType": "genericModel",
```

```

"generalInformation": {
  "name": "Dose-response model for Campylobacter in chicken meat",
  "source": "PUBLISHED SCIENTIFIC STUDIES",
  "identifier": "DR2007Classical",
  "creator": [
    {
      "givenName": "Stylianou",
      "email": "sgeo@food.dtu.dk",
      "country": "Denmark",
      "organization": "DTU"
    }
  ],
  "creationDate": [
    2018,
    3,
    19
  ],
  "rights": "Other",
  "availability": "true",
  "reference": [
    {
      "isReferenceDescription": true,
      "publicationType": "JOUR",
      "date": [
        2006,
        11,
        15
      ],
      "doi": "https://doi.org/10.1111/j.1539-6924.2006.00834.x",
      "authorList": "Nauta, M.J, Jacobs-
Reitsma, W.F., & Havelaar, A.H.",
      "title": "A Risk Assessment Model for Campylobacter in Broiler Me
at",
      "volume": "0",
      "issue": "0",
      "status": "Published",
      "website": "https://onlinelibrary.wiley.com/doi/abs/10.1111/j.153
9-6924.2006.00834.x",
      "abstract": "A quantitative microbiological risk assessment model
describes the transmission of Campylobacter through the broiler meat production
chain and at home, from entering the processing plant until consumption of a chic
ken breast fillet meal. The exposure model is linked to a dose-
response model to allow estimation of the incidence of human campylobacteriosis.
The ultimate objective of the model is to serve as a tool to assess the effects o
f interventions to reduce campylobacteriosis in the Netherlands. The model descri
bes some basic mechanistics of processing, including the nonlinear effects of cro

```

ss-

contamination between carcasses and their leaking feces. Model input is based on the output of an accompanying farm model and Dutch count data of Campylobacters on the birds' exterior and in the feces. When processing data are lacking, expert judgment is used for model parameter estimation. The model shows that to accurately assess the effects of interventions, numbers of Campylobacter have to be explicitly incorporated in the model in addition to the prevalence of contamination.

Also, as count data usually vary by several orders of magnitude, variability in numbers within and especially between flocks has to be accounted for. Flocks with high concentrations of Campylobacter in the feces that leak from the carcasses during industrial processing seem to have a dominant impact on the human incidence. The uncertainty in the final risk estimate is large, due to a large uncertainty at several stages of the chain. Among others, more quantitative count data at several stages of the production chain are needed to decrease this uncertainty. However, this uncertainty is smaller when relative risks of interventions are calculated with the model. Hence, the model can be effectively used by risk management in deciding on strategies to reduce human campylobacteriosis."

```
    }
  ],
  "language": "English",
  "software": "R",
  "languageWrittenIn": "R",
  "modelCategory": {
    "modelClass": "Process model",
    "modelSubClass": [
      "Storage at retail",
      "Storage at consumers level"
    ],
    "basicProcess": [
      "Growth, Growth"
    ]
  },
  "status": "Uncurated",
  "objective": "The objective of the model is to estimate the probability of illness after consumption of a meal with salad prepared together with fresh chicken breast fillet.",
  "description": "The model describes the dose-response model, which predicts the effects of food preparation by the consumer on the survival and transfer of Campylobacter. The input of the model is the distribution of Campylobacters on chicken breast fillets purchased at retail, as predicted in the preceding part of the food chain model. It is assumed that human exposure occurs after cross-contamination to a salad prepared at home. The end result is a mean probability of illness per meal, which could be expressed as an annual risk estimate for a population. This is the expected human incidence of campylobacteriosis as a consequence of the consumption of salad prepared with chicken breast fillet."
```

```

    },
    "scope": {
      "product": [
        {
          "name": "Poultry – chicken, geese, duck, turkey and Guinea fowl – ostrich, pigeon Meat",
          "description": "The final product is a salad prepared together with fresh chicken breast fillet",
          "unit": "g",
          "method": [
            null
          ],
          "packaging": [
            null
          ],
          "treatment": [
            null
          ]
        }
      ],
      "hazard": [
        {
          "type": "Microorganisms",
          "name": "Campylobacter jejuni",
          "unit": "CFU/g",
          "adverseEffect": "Campylobacteriosis"
        }
      ]
    },
    "dataBackground": {},
    "modelMath": {
      "parameter": [
        {
          "id": "dose",
          "classification": "INPUT",
          "name": "Doses",
          "unit": "CFU",
          "unitCategory": "",
          "dataType": "DOUBLE",
          "source": "Assumption",
          "subject": "Product",
          "value": "rep(1,100000)",
          "variabilitySubject": "Variability in doses"
        },
        {
          "id": "Pprev",

```

```

        "classification": "INPUT",
        "name": "Probability of prevalence at retail",
        "description": "Prevalence of Cabylobacter at retail",
        "unit": "[Probability]",
        "unitCategory": "Dimensionless Quantity",
        "dataType": "DOUBLE",
        "source": "Data",
        "subject": "Product",
        "value": "0.25",
        "variabilitySubject": "Point estimate"
    },
    {
        "id": "condPillinf",
        "classification": "INPUT",
        "name": "Probability of getting ill given being infected",
        "description": "Conditional probability that a consumer gets ill,
given that he/she is infected",
        "unit": "[Probability]",
        "unitCategory": "Number Content (count/mass)",
        "dataType": "DOUBLE",
        "source": "Assumption",
        "subject": "Other",
        "value": "0.33",
        "variabilitySubject": "Point estimate"
    },
    {
        "id": "alphaGamma",
        "classification": "INPUT",
        "name": "alpha parameter of the Gamma function",
        "description": "Parameter of the Gamma function in the dose-
response relationship.",
        "unit": "[]",
        "unitCategory": "Dimensionless Quantity",
        "dataType": "DOUBLE",
        "source": "Article",
        "subject": "Other",
        "distribution": "Gamma 2",
        "value": "0.145",
        "variabilitySubject": "Variability in the dose-
response relationship"
    },
    {
        "id": "betaGamma",
        "classification": "INPUT",
        "name": "beta parameter of the Gamma function",
        "description": "Parameter of the Gamma function in the dose-

```

```

response relationship.",
    "unit": "[]",
    "unitCategory": "Dimensionless Quantity",
    "dataType": "DOUBLE",
    "source": "Article",
    "subject": "Other",
    "distribution": "Gamma 2",
    "value": "7.59",
    "variabilitySubject": "Variability in the dose-
response relationship"
  },
  {
    "id": "PrevExp",
    "classification": "OUTPUT",
    "name": "Prevalence of exposure to Campylobacters",
    "unit": "[Probability]",
    "unitCategory": "Dimensionless Quantity",
    "dataType": "DOUBLE",
    "variabilitySubject": "Point estimate"
  },
  {
    "id": "Qill",
    "classification": "OUTPUT",
    "name": "Probability of illness at a dose",
    "unit": "[Probability]",
    "unitCategory": "Dimensionless Quantity",
    "dataType": "DOUBLE",
    "variabilitySubject": "Point estimate"
  }
],
"qualityMeasures": [
  {
    "sse": 0.0,
    "mse": 0.0,
    "rmse": 0.0,
    "aic": 0.0,
    "bic": 0.0
  }
]
}

```

Figure 9 JSON metadata file.



## 8. Supplementary Information

Supplementary Information associated with this article can be found at the online version (<https://foodrisklabs.bfr.bund.de/fsk-ml-food-safety-knowledge-markup-language/>).

## 9. Funding

This work was supported by the AGINFRA PLUS project funded by the European Commission's Horizon 2020 research and innovation programme under grant agreement No 731001.

## 10. References

- Bergmann, F. T., Adams, R., Moodie, S., Cooper, J., Glont, M., Golebiewski, M., . . . Le Novere, N. (2014). COMBINE archive and OMEX format: one file to share all information to reproduce a modeling project. *BMC Bioinformatics*, 15(1), 369. doi:10.1186/s12859-014-0369-z
- Bergmann, F. T., Cooper, J., Le Novere, N., Nickerson, D., & Waltemath, D. (2015). Simulation experiment description markup language (SED-ML) Level 1 Version 2. *Journal of integrative bioinformatics*, 12(2), 262. doi:10.2390/biecoll-jib-2015-262
- Codex Alimentarius Commission. (1999). Principles and guidelines for the conduct of microbiological risk assessment. Joint FAO/WHO Food Standards Programme. *CAC/GL-30, Rome*.
- Cyganik, R., Wood, D., Lanthaler, M., Klyne, G., Carroll, J. J., & McBride, B. (2014). RDF 1.1 concepts and abstract syntax. *W3C recommendation*, 25(02).
- DataCite Metadata Working Group. (2019). DataCite metadata schema documentation for the publication and citation of research data. *DataCite e.V., Version 4.3*. doi:<https://doi.org/10.14454/7xq3-zf69>
- de Alba Aparicio, M., Buschhardt, T., Swaid, A., Valentin, L., Mesa-Varona, O., Günther, T., . . . Filter, M. (2018). FSK-Lab – An open source food safety model integration tool. *Microbial Risk Analysis*, 10, 13-19. doi:<https://doi.org/10.1016/j.mran.2018.09.001>
- FAO/WHO. (2016). Codex Alimentarius commission. *Procedural Manual, 25th ed, Rome*.
- Filter, M., Heise, A., Thöns, C., Perez-Rodriguez, F., Cid García, M. Á., & de Alba Aparicio, M. (2016). Predictive modelling in food markup language (PMF-ML). doi:10.13140/RG.2.1.1086.2488
- Freed, N., & Borenstein, N. (1996). Multipurpose internet mail extensions (MIME) part two: Media types. *rfc 2046, November*.
- Haberbeck, L. U., Plaza-Rodríguez, C., Desvignes, V., Dalgaard, P., Sanaa, M., Guillier, L., . . . Filter, M. (2018). Harmonized terms, concepts and metadata for microbiological risk assessment models: The basis for knowledge integration and exchange. *Microbial Risk Analysis*, 10, 3-12. doi:10.1016/j.mran.2018.06.001

- Hucka, M., Bergmann, F. T., Hoops, S., Keating, S. M., Sahle, S., Schaff, J. C., . . . Wilkinson, D. J. (2015). The systems biology markup language (SBML): Language specification for Level 3 Version 1 core. *Journal of integrative bioinformatics*, 12(2), 266. doi:10.2390/biecoll-jib-2015-266
- ISO (International Organisation for Standardisation). (2004). Information technology – Metadata registries (MDR) Part 1: Fr ISO/IEC 11179-1.
- Nauta, M. J., Jacobs-Reitsma, W. F., & Havelaar, A. H. (2007). A Risk Assessment Model for *Campylobacter* in Broiler Meat. *Risk Analysis*, 27(4), 845-861. doi:10.1111/j.1539-6924.2006.00834.x
- R Core Team. (2019). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL: <http://www.R-project.org>.
- Whiting, R. C., Buchanan, R.L.,. (1993). A classification of models in predictive microbiology - a reply to K.R. Davey. *Food Microbiology*, 10, 175–177.
- Wikimedia Foundation Inc. (2019). Computer simulation. URL: [https://en.wikipedia.org/wiki/Computer\\_simulation](https://en.wikipedia.org/wiki/Computer_simulation).
- Wilkinson, M. D., Dumontier, M., Aalbersberg, I. J. J., Appleton, G., Axton, M., Baak, A., . . . Mons, B. (2016). The FAIR Guiding Principles for scientific data management and stewardship. *Scientific data*, 3, 160018-160018. doi:10.1038/sdata.2016.18