

# Food Safety Knowledge Markup Language (FSK-ML)

---

*Software Developer Guide*

*Version 2.0*

Matthias Filter (Chair)	Federal Institute for Risk Assessment, Germany
Carolina Plaza-Rodriguez	Federal Institute for Risk Assessment, Germany
Miguel de Alba Aparicio	Federal Institute for Risk Assessment, Germany

***Alumni contributors:***

Sascha Bulik	Federal Institute for Risk Assessment, Germany
Guido Correia Carreira	Federal Institute for Risk Assessment, Germany
Alexander Falenski	Federal Institute for Risk Assessment, Germany

***Contact:***

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

## List of contents

1.	Introduction.....	6
1.1.	Objectives.....	7
1.2.	Document conventions .....	7
2.	Related standards.....	9
2.1.	SBML.....	9
2.2.	SED-ML .....	9
2.3.	PMF-ML .....	9
2.4.	OMEX.....	9
3.	FSK-ML Specifications.....	10
3.1.	FSK Terminology .....	10
3.1.1.	Model .....	10
3.1.2.	Simulation.....	10
3.2.	The FSKX format .....	10
3.3.	Supported data types .....	11
3.4.	Files related to models.....	12
3.4.1.	Model script.....	12
3.4.2.	(Binary) Model libraries.....	13
3.4.3.	Model metadata.....	13
3.5.	Simulation files .....	15
3.5.1.	FSK-SED-ML files.....	16
3.5.2.	Visualization scripts.....	16
3.6.	Joined models.....	16
3.7.	Package versions .....	17
3.8.	Resources .....	17
3.9.	Other .....	18
3.9.1.	manifest.xml.....	18
3.9.2.	The Archive metadata .....	19
3.9.3.	Type SId .....	20
4.	FSK-SED-ML .....	21

4.1.	Language references .....	21
4.2.	SourceScript.....	21
4.3.	Model .....	22
4.4.	Simulation.....	22
4.4.1.	FskSimulationType.....	23
4.5.	Task.....	24
4.6.	DataGenerator.....	24
4.7.	Output .....	25
5.	Example models .....	27
5.1.	Dose response example model .....	27

## List of tables

Table 1 Supported data types .....	12
Table 2 Supported model classes .....	12
Table 3 List of metadata associated with controlled vocabularies lists and their sources .....	15
Table 4 FSK model classes .....	15
Table 5 Formats used in the manifest .....	19
Table 6 Predefined language references .....	21

## List of figures

Figure 1 Structure of the FSKX file.....	11
Figure 2 Example variables.....	12
Figure 3 Example model script .....	12
Figure 4 Structure of model encoded in a standardized way within the FSK-ML file .....	14
Figure 5 Example visualization script .....	16
Figure 6 Example joined model .....	17
Figure 7 Example package versions.....	17
Figure 8 Example manifest .....	18
Figure 9 Example archive metadata .....	20
Figure 10 Definition of the type SId .....	20
Figure 11 Definition of the SourceScript class .....	21
Figure 12 Example SEDML model.....	22
Figure 13 Definition of the FskSimulationType class .....	23
Figure 14 Example of the FskSimulation class.....	24
Figure 15 Example of the Task class .....	24
Figure 16 Example of the Output class with referenced and embedded script examples .....	25
Figure 17 Example SEDML document .....	26

# 1. Introduction

---

Food safety risk assessments, control of food production processes as well as the development of new food products are nowadays supported by application of mathematical modelling and data analysis techniques. This creates an increasing demand for resources facilitating the efficient, transparent and quality proven exchange of relevant information, e.g. analytical data, mathematical models, simulation setting as well as simulated data. For example, new parameterized microbial models are frequently made publicly available only in written mode via scientific publications. However, in order to apply these models to a given practical decision support question (e.g. on the growth/no-growth of a microorganism in a specific food matrix under given processing conditions) the interested end-user would have to re-implement the model based on information provided in a publication. Here it would be more efficient if those who create parameterized models could provide their model additionally as a file complying with a standardized file format that is also capable of transferring all relevant meta-information. Such a file could e.g. be provided as a supplement to the publication and could be read-in by the end user's software tools (thus overcoming an error-prone re-implementation process).

The required standardized file format has been proposed in the "Predictive Modelling in Food Markup Language (PMF-ML) Software Developer Guide". This document describes in detail how experimental data and mathematical models from the domain of predictive microbial modeling (and beyond) can be saved and encoded in a **software independent manner**. Here we further extend the PMF-ML format in order to enable in addition the exchange of knowledge / information that is **embedded in specific script-based programming languages** (e.g. "R", Matlab, Python). I.e. the FSK-ML guidance document primarily aims at harmonizing the exchange of food safety knowledge (e.g. predictive models) including the associated meta-information where this knowledge is only available in a **software-dependent** format.

The FSK-ML format therefore relaxes and adapts certain specifications of the PMF-ML format while at the same time maintaining the highest possible synergies between both formats. This will also help to make sure that food safety models encoded in a software-independent manner

(using PMF-ML) can easily be interpreted by FSK-ML import and export software functions in the future.

## 1.1. Objectives

---

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 food safety models, simulations or food safety data or for those who want to develop new tools.

The document describes the structure and requirements of FSK-ML including the structure and requirements of a newly defined file format which is based on the Open Modeling EXchange guidelines (OMEX). The document also describes information needed to make encoded script-based models executable. Future versions of the document will also describe how to encode simulation settings and settings for data visualization.

## 1.2. Document conventions

---

This document uses the conventions defined in the SED-ML specification document (Bergmann, Cooper, Le Novère, Nickerson, & Waltermath, 2015).

UML 1.0 (Unified Modelling Language), (OMG, 2009), notation is used in this document to define the constructs provided by this package. The following colours are used to provide more meaningful diagrams.

- **Black.** Items coloured black in the UML diagrams are components taken unchanged from their definition in the SED-ML Level 1 Version 2 specification document.
- **Green.** Items coloured green are components that exist in SED-ML Level 1 Version 2, but are extended by this package.
- **Blue.** Items coloured blue are new components introduced in this package specification. They have no equivalent in the SED-ML Level 1 Version 2 specification.

The following typographical conventions distinguish the names of objects and data types from other entities.

**AbstractClass:** Abstract classes are never instantiated directly, but rather serve as parents of other classes. Their names begin with a capital letter and they are printed in a slanted, bold, 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 that class. (However, for classes that are unchanged from their definitions in SED-ML Level 1 Version 2, the class names are not hyperlinked because they are not defined within this document.)

**Class:** Names of ordinary (concrete) classes begin with a capital letter and are printed in an upright, bold, sans-serif typeface. In electronic document formats, the class names are also hyperlinked to their definitions in this specification document. (However, as in the previous case, class names are not hyperlinked if they are for classes that are unchanged from their definitions in the SED-ML Level 1 Version 2 specification.)

`Something`, `otherThing`: Attributes of classes, data type names, literal XML, and tokens other than SED-ML class names, are printed in an upright typewriter typeface.



## 2. Related standards

---

### 2.1. SBML

---

See “Predictive Modelling in Food Markup Language (PMF-ML) Software Developer Guide”.

### 2.2. SED-ML

---

See “Predictive Modelling in Food Markup Language (PMF-ML) Software Developer Guide”.

### 2.3. PMF-ML

---

See “Predictive Modelling in Food Markup Language (PMF-ML) Software Developer Guide”.

### 2.4. OMEX

---

Open Modelling EXchange format, OMEX (Bergmann, et al., 2014), aims to support the exchange of information necessary for modelling and simulating experiments in biology. OMEX defines a COMBINE Archive or OMEX files as a ZIP container containing at least a file with a listing of content in the archive (manifest file), optional metadata and the files describing e.g. food safety models. The use of COMBINE standard file formats for the encoding of models is encouraged, although several Internet media types are also supported.

## 3. FSK-ML Specifications

---

### 3.1. FSK Terminology

---

#### 3.1.1. Model

A model is a mathematical description of a system consisting of state variables and their dependency on each other and external conditions (e.g. pH, temperature, time). The mathematical representation of these dependencies contain fixed parameter values specific for this model and variable values, also called “independent parameters” or “input parameters” here, that can be set to different values to attain model prediction for not measured combinations of these entities. It is to note that a model is normally only valid on a given range for these variable values.

#### 3.1.2. Simulation

Model-based predictions generated from different parameterizations of the same model are called simulations. I.e. generating model-based predictions from a set of definite values for all model input parameters will be called a simulation. It should be noted, that a simulation might create different sets of definite values for input parameter as FSK-ML support script-based simulation settings.

### 3.2. The FSKX format

---

In order to support the exchange of sets of related files a FSKX container file is created. A FSKX container file is a ZIP file containing an arbitrary (at least one) number of files. We recommend the file extension “.fskx”. The FSKX container has to comply with the Open Modeling EXchange format (OMEX) specifications set out in (Bergmann, Adams et al. 2014). It allows to exchange the following files:

- a manifest file (“manifest.xml”) - mandatory; providing a listing of all files inside the FSKX container
- a metadata file (“metadata.rdf”) providing additional information on the archive and its content

- files related to the model(s):
  - Data used for model generation / validation (e.g. “experiment\_1.numl”, “experiment\_2.xls” )
  - Model metadata (e.g. “metaData\_1.pmf”) providing all relevant metadata on referenced model script(s) / executable software (black-box) model(s)
  - Model script(s) (e.g. “model\_1.r”) / executable software (black-box) model(s) (in the future also “model\_1.exe”/ pmfx files (e.g. “model\_1.pmfx”)
  - Third party (including binary) libraries required to run the model script(s) / executable software (black-box) model(s) (e.g. “triangle\_0.10.zip”)
- file(s) related to simulation(s) / prediction and simulation / prediction result(s):
  - Simulation settings (e.g. “Simulation\_1.sed”)
  - Simulation results (e.g. “Simulation\_1\_res.r”)
  - Visualization script (e.g. “Simulation\_1\_plot.r”)
- Other file(s)
  - Model reference description (e.g. “Paper\_model\_1.pdf”)

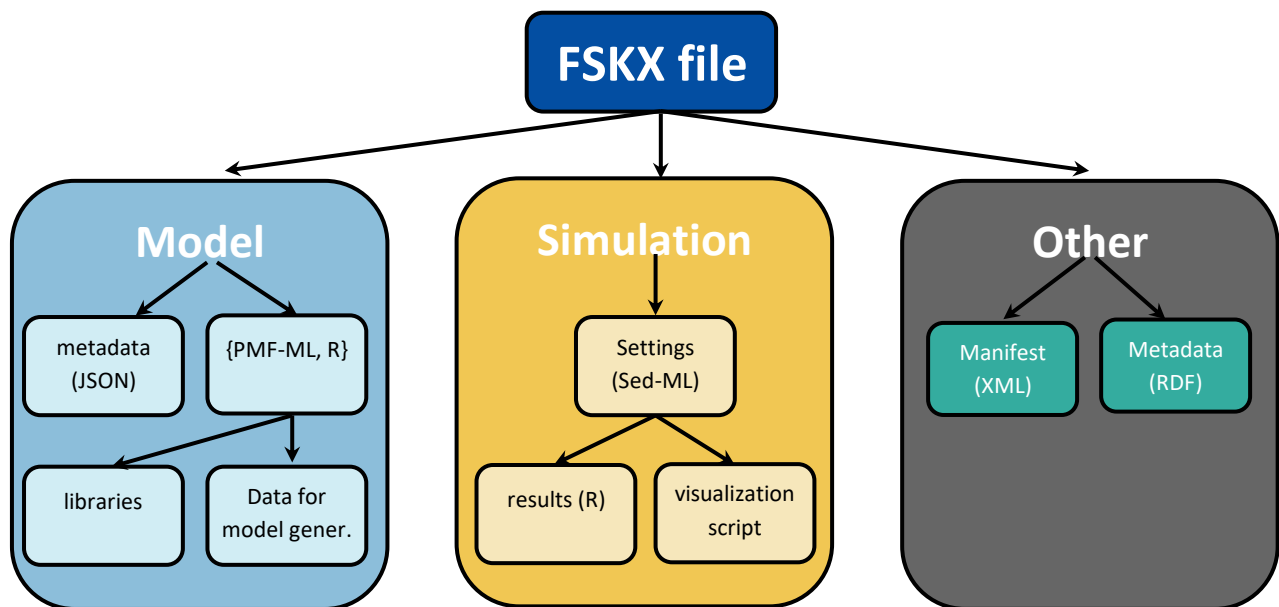


Figure 1 Structure of the FSKX file

### 3.3. Supported data types

Metadata values on model input /output parameters and simulation results of the specified models must adhere to the supported data types defined here:

- Numeric: Real numbers.
- Integer
- Character: String values

- Vector: one-dimensional array
  - Matrix: two-dimensional array
- Code vocabulary for the supported data types:

Numeric	numeric
Integer	integer
Character	character
Vector	vector
Matrix	matrix

**Table 1 Supported data types**

The supported data types are case-insensitive.

```
# Examples of scalar data types
pi = 3.14
year = 2016
name = "Miguel, Lord of Westeros"

# Vectors
cookies_per_meeting <- c(10, 5, 20, 15, 10)
cookies_per_meeting <- c(10, "fünf", 20, "fünfzehn", 10)
cookies_per_meeting <- c("zehn", "fünf", "zwanzig", "fünfzehn", "zehn")

# Matrices
menus = matrix(c("Hänschen", "Reis", "Fishfilet", "Geschnezeltes", "Spinach", "Boulette"),
               nrow = 2, ncol = 3, byrow = TRUE,
               dimnames = list(c("Menu 1", "Menu 2"), c("Montag", "Dienstag", "Mittwoch")))
```

**Figure 2 Example variables**

## 3.4. Files related to models

---

### 3.4.1. Model script

---

The model script is a script stored within the FSKX archive that calculates the output of the model. Each model script has to be declared in the manifest.xml, annotated by a dedicated metadata file (.pmf) and can be referenced by any simulation settings file (.sed)

```
#####
test_model_1 = function(a=1, b=2, c=3){
  res = 100*(rbeta(a, shape1 = b + 1, shape2 = c - b + 1))
  return(res)
}
test_model_1()
```

**Figure 3 Example model script**

### 3.4.2. (Binary) Model libraries

---

For an autonomous run the model needs to be provided such that required libraries are provided together with the model. In the case of an R model the corresponding libraries would be binary libraries for the system the model was created on.

While the placement of the libraries in the archive is free, they can be stored anywhere in the FSKX file; it is advisable to save them into a separated “lib” folder within the archive.

### 3.4.3. Model metadata

---

FSK model classes characterize the particular models that can currently be described by FSK-ML. This classification is partly based on the definition made by CODEX *Alimentarius* on the Risk Assessment (Table 4). Some of the FSK model classes are subdivided into sub-classes, which are included as controlled vocabularies lists. Apart from this FSK Model Classes, FSK-ML allows to annotate models under the label “Generic model” which allows more flexibility in providing information on the model. The model metadata follows a unique schema associated to the model class.

- **Generic model.**
- **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)”.
- **Predictive model**
- **Health metrics model**
- **Exposure model.** Combination of the process model and the consumption model that results in the exposure assessment.
- **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 fork).
- **Toxicological reference value model**
- **QRA model**
- **Risk characterization model.** Combination of health metrics model, dose-response model and exposure model within the framework of the risk characterization.
- **Other empirical models**
- **Data model**
- **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.

Each of the mentioned schemas has a specific subset of metadata. All the schemas have four common elements: general information, scope, data background and model math. These elements may differ across schemas.

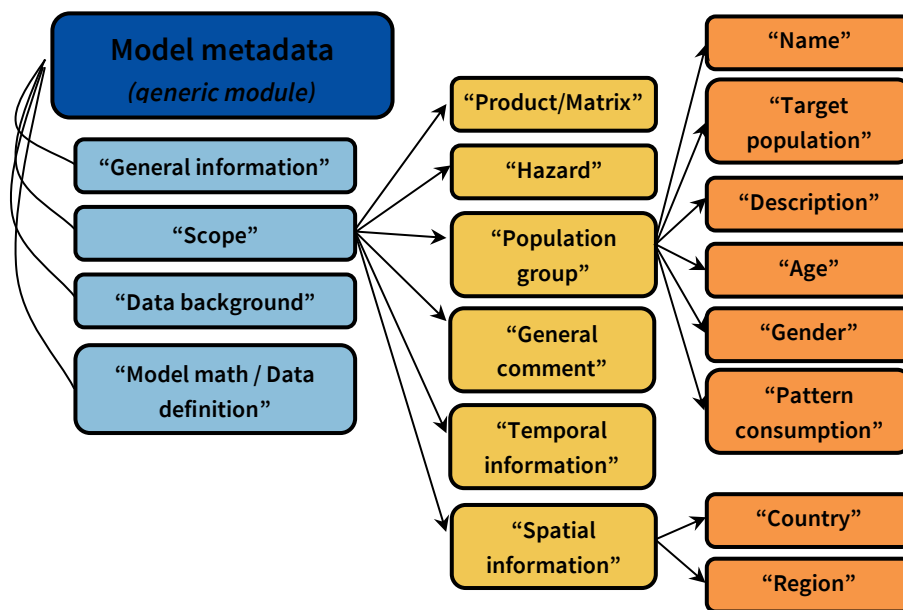


Figure 4 Structure of model encoded in a standardized way within the FSK-ML file

In addition, FSK-ML provides a list of controlled vocabularies for some specific metadata (Table 3) and <https://docs.google.com/spreadsheets/d/1C6N4-YWX9OMmNStd2rYISUaVys-aiJGLj00cD44aVc8/edit#gid=1479548673>, based on the terms used by other sources like ontologies, standards and tools (SSD-CODE, FOODON, MIME, PMM-Lab, OpenFSMR, Bibliographic Ontology Specification, etc.)

Source	controlled vocabulary_ <i>Terms and concepts</i>
Rights	controlled vocabulary
Format	controlled vocabulary_ <i>Internet Media Types [MIME]</i>
Publication Type	controlled vocabulary_ <i>Bibliographic Ontology Specification</i>
Publication Status	controlled vocabulary_ <i>Bibliographic Ontology Specification</i>
Software	controlled vocabulary
Language	controlled vocabulary_ <i>SSD-code</i>
Language written in	controlled vocabulary
Model Class	controlled vocabulary
Model Sub-Class	class-specific controlled vocabularies incl. OTHER ( <i>OpenFSMR</i> )
Basic process	sub-sub-class-specific controlled vocabularies incl. OTHER
Status	controlled vocabulary
Product-matrix name	controlled vocabulary_ <i>SSD-code</i>

<b>Product-matrix unit</b>	<i>controlled vocabulary_SSD-code_PMM-Lab</i>
<b>Method of production</b>	<i>controlled vocabulary_SSD-code</i>
<b>Packaging</b>	<i>controlled vocabulary_SSD-code</i>
<b>Product treatment</b>	<i>controlled vocabulary_SSD-code</i>
<b>Country of origin</b>	<i>controlled vocabulary_SSD-code</i>
<b>Area of origin</b>	<i>controlled vocabulary_SSD-code</i>
<b>Fisheries area</b>	<i>controlled vocabulary_SSD-code</i>
<b>Hazard type</b>	<i>controlled vocabulary_SSD-code</i>
<b>Hazard name</b>	<i>controlled vocabulary_SSD-code</i>
<b>Hazard unit</b>	<i>controlled vocabulary_SSD-code_PMM-Lab</i>
<b>Hazard ind-sum</b>	<i>controlled vocabulary_SSD-code</i>
<b>Population name</b>	<i>controlled vocabulary_FOODON</i>
<b>Laboratory country</b>	<i>controlled vocabulary_SSD-code</i>
<b>Region</b>	<i>controlled vocabulary_SSD-code</i>
<b>Country</b>	<i>controlled vocabulary_SSD-code</i>
<b>Study Assay Technology Type</b>	<i>controlled vocabulary_SSD-code</i>
<b>Accreditation procedure for the assay technology</b>	<i>controlled vocabulary_SSD-code</i>
<b>Sampling strategy</b>	<i>controlled vocabulary_SSD-code</i>
<b>Type of sampling program</b>	<i>controlled vocabulary_SSD-code</i>
<b>Sampling method</b>	<i>controlled vocabulary_SSD-code</i>
<b>Lot size unit</b>	<i>controlled vocabulary_SSD-code_PMM-Lab</i>
<b>Sampling point</b>	<i>controlled vocabulary_SSD-code</i>
<b>Methodological tool to collect data</b>	<i>controlled vocabulary</i>
<b>Type of records</b>	<i>controlled vocabulary</i>
<b>Food descriptors</b>	<i>controlled vocabulary_SSD-code</i>
<b>Laboratory accreditation</b>	<i>controlled vocabulary_SSD-code</i>
<b>Parameter classification</b>	<i>controlled vocabulary</i>
<b>Parameter type</b>	<i>controlled vocabulary</i>
<b>Parameter unit</b>	<i>controlled vocabulary_SSD-code_PMM-Lab</i>
<b>Parameter unit category</b>	<i>controlled vocabulary_PMM-Lab</i>
<b>Parameter data type</b>	<i>controlled vocabulary</i>
<b>Parameter source</b>	<i>controlled vocabulary</i>
<b>Parameter subject</b>	<i>controlled vocabulary</i>
<b>Parameter distribution</b>	<i>controlled vocabulary_probONTO</i>
<b>Model equation class - distribution</b>	<i>controlled vocabulary_iRISK</i>
<b>Fitting procedure</b>	<i>controlled vocabulary</i>
<b>type of exposure</b>	<i>controlled vocabulary</i>
<b>Simulation algorithm</b>	<i>controlled vocabulary</i>

Table 2 List of metadata associated with controlled vocabularies lists and their sources<sup>1</sup>

### 3.5. Simulation files

FSK-ML provides an opportunity to define, store and exchange simulation scenarios. All relevant settings (including the reference to the used model files) are stored in XML files using the FSK-SED-ML format (see section 4). Also visualizations to be generated on the basis of the simulation results can be referenced / defined in the FSK-SED-ML file.

<sup>1</sup> As of February 2018, available online at <https://docs.google.com/spreadsheets/d/1C6N4-YWX9OMmNStd2rYISUaVys-aiJGLj00cD44aVc8/edit#gid=1479548673>

The results of simulations are store in R workspace files.

### 3.5.1. FSK-SED-ML files

The FSK-SED-ML files are XML files (.sed) that describe specific simulations to be carried out with model(s). For this it has to describe:

- The parameters values declared as inputs and constants for the simulation or the script defining the generation of input and constant parameters for the simulation. In addition, it has to be defined how the simulation is carried out itself (including the definition how the model script is executed with each simulation input parameter set)
- Metadata on the type of simulation: deterministic, statistic or probabilistic

FSK-SED-ML is described in more detail in section 4 FSK-SED-ML.

### 3.5.2. Visualization scripts

Visualization scripts can be included and might contain a number of commands in a scripting language corresponding to the one used for the model. It can be used to create plots or charts using from the results generated in a simulation. Visualization scripts might also be referenced from within FSK-SED-ML files.

```
hist(test_model_1(200, 20, 100), breaks=50, main="Headline", xlab="Text", col="32")
```

Figure 5 Example visualization script

## 3.6. Joined models

FSK-ML can describe joined models where the output of one model is used as an input of a second model. FSKX files may content a number of nested joined models where every model is contained in a subdirectory name after its name.

In the root directory of the FSKX file an SBML file defines the joining of the models contained.



```
<?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="SimpleModel2"
comp:source="SimpleModel2\SimpleModel2\SimpleModel2.sbml"/>
    <comp:externalModelDefinition comp:id="SimpleModel2"
comp:source="SimpleModel2\SimpleModel2\SimpleModel2.sbml"/>
  </comp:listOfExternalModelDefinitions>
  <model id="SimpleModel2">
    <comp:listOfSubmodels
xmlns:comp="http://www.sbml.org/sbml/level3/version1/comp/version1">
      <comp:submodel comp:id="submodel1" comp:modelRef="SimpleModel2"/>
      <comp:submodel comp:id="submodel2" comp:modelRef="SimpleModel2"/>
    </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 6 Example joined model

## 3.7. Package versions

---

The versions of the software packages used in a model are listed in a JSON file named *packages.json* that is located at the root directory of the FSKX file.

```

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

```

Figure 7 Example package versions

## 3.8. Resources

---

FSKX files may contain resources necessary for model execution or generated after. The allowed resources are:

- Plain text files
- R workspaces (RData)
- CSV files. Mostly used to load data in a model.
- JPEG, BMP, PNG and TIFF files. Mostly used to store plots generated by a model.

## 3.9. Other

---

### 3.9.1. manifest.xml

---

The FSK container is a COMBINE archive and the presence of this file is a requirement of the COMBINE archive.

*One file must be present at the root of any COMBINE archive, named manifest.xml. This file contains an instantiation of the OmexManifest class. It contains a number of Content children, one of which represents the COMBINE archive itself. A valid manifest needs to have at least one entry, declaring the archive itself, but may contain as many entries as needed. All the files present in the archive must be listed in the manifest. The only optional entry describes the manifest.xml file. Indeed the presence of manifest.xml is mandatory and its declaration is not necessary for parsing.*

```
<?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 8** Example manifest

The Content class represents a file in the COMBINE archive. The format attribute in a content element describes the file type. It may take either a URI corresponding to Identifiers.org for COMBINE standards such as SBML or SedML or an Internet media type (Freed, 1996), previously known as “MIME type”. FSK makes use of extra file types through the following Internet media types.

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>

---

Table 3 Formats used in the manifest

## 3.9.2. The Archive metadata

A reduced model definition is allowed by providing additional annotation in the RDF format. In the FSKX archive the use of Internet media types as formats allows the detection of different types of files within the archive file such as scripts and binary libraries. However, files might share the same media types, as e.g. for R scripts and R workspace files. Therefore, the use of Internet media types is not enough.

The RDF annotation of the archive also describes the version using the conformsTo property from the Dublin Core Metadata Initiative.

```
<?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=".">
    <dc:type>mainScript</dc:type>
    <dc:source>model.r</dc:source>
  </rdf:Description>
```

```

<rdf:Description rdf:about=". ">
  <dc:type>paramScript</dc:type>
  <dc:source>param.r</dc:source>
</rdf:Description>

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

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

```

**Figure 9** Example archive metadata

The Archive metadata in FSK addresses this issue, providing a lightweight annotation using the Resource Description Format, RDF (Cyganiak, Wood, Lanthaler, Klyne, Carroll, & and McBride, 2014). This annotation involves RDF description elements that describe the type and location of the resource with DC type and source elements respectively. The accepted types are:

- `mainScript`: filename of the model script
- `paramScript`: filename of the parameters script
- `visualizationScript`: filename of the visualization script
- `workspace`: filename of the workspace with the results of the simulation

### 3.9.3. Type SId

FSK-ML uses the type SId from SBML for a number of identifiers. The SId type is derived from the basic XML type string, but with restrictions about the characters permitted and the sequences in which those characters may appear.

```

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

```

**Figure 10** Definition of the type SId

## 4. FSK-SED-ML

FSK-SED-ML extends SED-ML to describe the simulation settings for FSK-ML models. FSK-SED-ML manages this by introducing new classes and annotating extra metadata on existing SED-ML components using annotations. FSK-SED-ML files contain references to the used models, their parameterizations, simulation setups, output methods for the storage of results and methods used for visualisation of results. Unlike SED-ML FSK-SED-ML does not require the methods to be in an environment independent Markup Language format but contains the capacity to reference to scripts in supported languages. The main structure of the SED-ML format is maintained (`<listOfModels><listOfSimulations><listOfTasks><listOfDataGenerators><listOfOutputs>`), but each node of these lists might refer to one or more [SourceScript](#).

### 4.1. Language references

FSK-SED-ML defines a list of URIs for scripting languages to be used in the SED-ML components.

Language	MIME type
R	<a href="https://iana.org/assignments/mediatypes/text/x-r">https://iana.org/assignments/mediatypes/text/x-r</a>
Python	<a href="https://iana.org/assignments/mediatypes/text/x-python">https://iana.org/assignments/mediatypes/text/x-python</a>
Matlab	<a href="https://iana.org/assignments/mediatypes/text/x-matlab">https://iana.org/assignments/mediatypes/text/x-matlab</a>

Table 4 Predefined language references

### 4.2. SourceScript

[SourceScript](#) is a new class (Figure 11) that allows to reference external scripts in supported languages, and contains the path to the script and the scripting language.

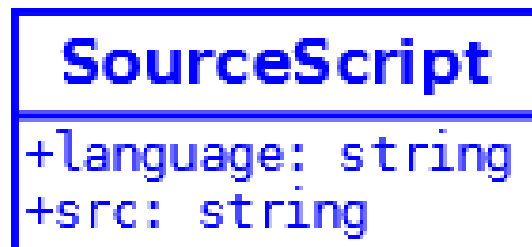


Figure 11 Definition of the SourceScript class

language

The required language attribute of data type string is used to specify the scripting language of the code embedded within a [SourceScript](#) element or referenced through the src attribute. The allowed languages are listed in Table 4.

src

The optional src attribute of data type anyURI is used to indicate the path of a local script file. The path is resolved via the local file system or as a relative link. If src is not provided the script needs to be provided within **SourceScript** as free text. When src is provided any free text in **SourceScript** is ignored.

When the src attribute is specified in a simulation script, the Simulation element should not include script code within the <sourceScript> element.

## 4.3. Model

---

References the model files used in simulations. However, the execution of models in default or specific simulation settings is defined in the Task nodes. Therefore, each FSKX file contains at least one FSK-SED-ML file providing the settings for executing each model with the default parameters.

language

The language attribute is mandatory in FSK-SED-ML. It takes a value from Table 4.

source

The source attribute is mandatory in FSK-SED-ML.

```
<listOfModels>
  <model id="model1" name="initialize_parents_flocks"
    language="https://iana.org/assignments/mediatypes/text/x-r"
    src="./model.r">
  </model>
</listOfModels>
```

Figure 12 Example SEDML model

Support for algebraic SBML based models is also provided. Supported classes of SBML are **Species** for target values, **Parameter** for dependent and constant parameters and **AssignmentRule** for implementation of the equations for target values and dependent parameters. Boundaries for the parameters can be declared by usage of the **Constraint** class.

When referencing to an existing SBML-file the SED-ML **Change** class can be used to make modifications.

## 4.4. Simulation

---

The **Simulation** class defines the settings under which models should be executed. In the simplest case this defines just new input parameters for execution of a given model, e.g. replacing default values given in the parameters script file. Each simulation scenario defined

in a **Task** instance has to refer to simulation settings defined in a **Simulation**. Nevertheless, it is possible that simulation settings defined as a script can already contain the script command that calls a specific model. In general, a Task is the place that specifies which simulation setting is combined how with which model.

In SED-ML **Simulation** act as a container for defining the simulation experiments. Script-based models use a SteadyState simulation whose algorithm has an empty KISAO id ("" ) since no algorithm is required.

Simulations are further annotated with two annotations: **FskSimulationType** and **SourceScript**.

### 4.4.1. FskSimulationType

**FskSimulationType** is a new class Figure 13 that allows providing metadata on the kind of simulation that will be performed. The following terms are proposed: deterministic, statistic and probabilistic.



Figure 13 Definition of the FskSimulationType class

- Deterministic simulations operate on definite values for all model input parameters
- Statistic simulations create descriptive analysis of observational data. These simulations are meant to describe and analyse data.
- Probabilistic simulations use probabilistic methods like Monte Carlo simulations to create model input parameters.

type

The required type attribute of data type string is used to specify the kind of simulation that will be performed. Allowed values are deterministic, statistic and probabilistic.

```
<listOfSimulations>
  <steadyState id="Simulation1">
    <algorithm kisaoID="" />
    <annotation>
      <fskSimulationType type="deterministic" />
      <sourceScript id="simulation1"
        language="https://iana.org/assignments/mediatypes/text/x-r">
        source("param.r")
        Npos <- 20
        source("model.r")
        result1 <- result

        Npos <- 30
        source("model.r")
        result2 <- result
      </sourceScript>
    </annotation>
  </steadyState>
</listOfSimulations>
```

```

        Npos <- 40
        source("model.r")
        result3 <- result
    </sourcescript>
</annotation>
</steadyState>
</listOfSimulations>

```

**Figure 14 Example of the FskSimulation class**

Simulations for SBML models are defined here and have to be interpreted and executed by the respective FSK-Node. Supported simulations will be: ‘Time courses’ (primary and primary-secondary) and ‘Parameter scans’ (all). Additional simulation until boundary condition is met or violated will be added.

Of the predefined SED-ML classes the **UniformTimeCourse** is used and expanded by annotations.

## 4.5. Task

The **Task** class defines each simulation scenario through the combination of Models with Simulation setting considered in the current FSK-SED-ML file. It further defines the “execution order” for each simulation scenario. If no additional [SourceScript](#) is defined the referenced **Simulation** will be executed.

```

<listOfTasks>
  <task id="Task1" name="FlocksSimu" modelReference="model1"
    simulationReference="Simulation1">
  </task>
  <task id="Task2" name="Model_Default" modelReference="model1"
    simulationReference="Simulation_Default">
    <annotation>
      <sourcescript id="task"
        language="https://iana.org/assignments/mediatypes/text/x-r">
        source("param.r")
        source("model.r")
      </sourcescript>
    </annotation>
  </task>
</listOfTasks>

```

**Figure 15 Example of the Task class**

## 4.6. DataGenerator

The **DataGenerator** class defines which values from which tasks should be considered for output. The post-processing of values may also be defined in order to bring values in appropriate form for later output. It is possible to specify or reference scripts here as well.



## 4.7. Output

The **Output** class defines how the values specified in the **Task** or **DataGenerator** are plotted.

```
<listOfOutputs>
  <plot2D id="plot1">
    <annotation>
      <sourcescript id="script1"
        language="https://iana.org/assignments/mediatypes/text/x-python"
        src="visualization-script.py" />
    </annotation>
  </plot2D>
  <plot2D id="plot2">
    <annotation>
      <sourcescript id="script2"
        language="https://iana.org/assignments/mediatypes/text/x-r" >
        hist(result, breaks=50, main="PREVALENCE OF PARENTS FLOCKS", xlab="Prevalence",
          col="32")
      </sourcescript>
    </annotation>
  </plot2D>
</listOfOutputs>
```

Figure 16 Example of the Output class with referenced and embedded script examples

```
<?xml version="1.0" encoding="utf-8"?>
<!-- Written by Sascha Bulik,
      Guido Correia Carreira
      Miguel de-Alba-Aparicio,
      Carolina Plaza-Rodríguez,
      Matthias Filter
-->
<sedFSKML xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://sed-ml.org/sed-ml-L1-V2.xsd"
  xmlns="http://sed-ml.org/sed-ml/level1/version2"
  level="1" version="3">

  <listOfDataDescriptions>
    <dataDescription id="dataDescription1" name="model script"
      source="./model.r" format="https://iana.org/assignments/mediatypes/text/x-r">
      <listOfDataSources>
        <dataSource id="prevalenceSource" indexSet="Prevalence" />
      </listOfDataSources>
    </dataDescription>
  </listOfDataDescriptions>

  <listOfDataGenerators>
    <dataGenerator id="dgPrevalence">
      <listOfVariables>
        <variable id="Prevalence" modelReference="model1" target="#prevalenceSource" />
      </listOfVariables>
      <math xmlns="___">
        <math:ci>Prevalence</math:ci>
      </math>
    </dataGenerator>
  </listOfDataGenerators>

  <listOfModels>
    <model id="model1" name="initialize_parents_flocks">
      <annotation>
        <sourcescript id="model"
          language="https://iana.org/assignments/mediatypes/text/x-r">
```

```

        result <- 100 * rbeta(n.iter, shape1=Npos+1,
                             shape2=Ntotal-Npos+1)
    </sourceScript>
</annotation>
</model>
</listOfModels>

<listOfSimulations>
  <steadyState id="Simulation1">
    <algorithm kisaoID="" />
    <annotation>
      <fskSimulationType type="deterministic" />
      <sourceScript id="simulation1"
        language="https://iana.org/assignments/mediatypes/text/x-r">
        n.iter <- 200
        Npos <- 30
        Ntotal <- 100
      </sourceScript>
    </annotation>
  </steadyState>
</listOfSimulations>
<listOfTasks>
  <task id="Task1" name="FlocksSimu" modelReference="model1"
    simulationReference="Simulation1">
  </task>
</listOfTasks>

<listOfOutputs>
  <plot2D id="plot1">
    <annotation>
      <sourceScript id="visualization1"
        language="https://iana.org/assignments/mediatypes/text/x-r">
        hist(result, breaks=50,
          main="PREVALENCE AMONG FLOCKS IN PARENT GENERATION",
          xlab="Prevalence", col="32")
      </sourceScript>
    </annotation>
  </plot2D>
</listOfOutputs>
</sedFSKML>

```

**Figure 17** Example SEDML document

## 5. Example models

### 5.1. Dose response example model

This example defines a parameterized dose-response model which gives the probability of infection after ingestion of PRRS virus by pigs for a given dose.

$$P(\text{Infect}|Dose) = 1 - \left(1 + \frac{Dose}{beta}\right)^{-alpha}$$

Here  $alpha=0.3$  and  $beta=14400$  ( $\log_{10}$  TCID<sub>50</sub>) are PRRS and pig specific parameters. The equation represents a beta-poisson response function where the alpha describes the slope of the response curve and the value of beta shifts the curve along the abscissa (Dose axis).

This model has one variable and two parameters with the following default values:

- Input:

Variables	Dose = 4 $\log_{10}$ TCID <sub>50</sub>
Parameter	Alpha = 0.3
	Beta = 14400 $\log_{10}$ TCID <sub>50</sub>

- Output

Variables	P_Infect_Dose
-----------	---------------

Example metadata:

General Information	Model Name		Dose Response Model for Porcine Reproductive And Respiratory Syndrome Virus
	Source		PUBLISHED SCIENTIFIC STUDIES: Dose-response data and models
	Identifier		<i>Dose-Response_Brookes_Pork_001</i>
	Creator(s)	vCard 4.0 standard	V.J. Brookes; Charles Sturt University – Wagga Wagga; vbrookes@csu.edu.au
	Date	Creation date	10.29.2013
	Rights	Rights	Copyright © 2014 Elsevier B.V.
	Availability		Public
	URL		URI / URL of openFSMR or other model repository
	References	Is_reference_description?	Yes
		Publication type	Journal
		Publication date	2014 Mar 1
		PubMed ID	24502944
		Publication DOI	<a href="https://doi.org/10.1016/j.prevetmed.2014.01.016">https://doi.org/10.1016/j.prevetmed.2014.01.016</a>
		Publication Author List	Brookes VJ, Hernández-Jover M, Holyoake P, Ward MP.
		Publication Title	Import risk assessment incorporating a dose-response model: introduction of highly pathogenic porcine reproductive and respiratory syndrome into Australia via illegally imported raw pork.
		Publication Abstract	Highly pathogenic porcine reproductive and respiratory syndrome (PRRS) has spread through parts of south-east Asia, posing a risk to Australia. The objective of this study was to assess the probability of infection of a feral or domestic pig in Australia with highly pathogenic PRRS following ingestion of illegally imported raw

			pork. A conservative scenario was considered in which 500 g of raw pork was imported from the Philippines into Australia without being detected by border security, then discarded from a household and potentially accessed by a pig. Monte Carlo simulation of a two-dimensional, stochastic model was used to estimate the probability of entry and exposure, and the probability of infection was assessed by incorporating a virus-decay and mechanistic dose-response model. Results indicated that the probability of infection of a feral pig after ingestion of raw meat was higher than the probability of infection of a domestic pig. Sensitivity analysis was used to assess the influence of input parameters on model output probability estimates, and extension of the virus-decay and dose-response model was used to explore the impact of different temperatures and time from slaughter to ingestion of the meat, different weights of meat, and the level of viraemia at slaughter on the infectivity of meat. Parameters with the highest influence on the model output were the level of viraemia of a pig prior to slaughter and the probability of access by a feral pig to food-waste discarded on property surrounding a household. Extension of the decay and dose-response model showed that small pieces of meat (10 g) from a highly pathogenic PRRS viraemic pig could contain enough virus to have a high probability of infection of a pig, and that routes to Australia by sea or air from all highly pathogenic PRRS virus endemic countries were of interest dependent on the temperature of the raw meat during transport. This study highlighted the importance of mitigation strategies such as disposal of food-waste from international traffic as quarantine waste, and the need for further research into the probability of access to food-waste on properties by feral pigs.
		Publication Journal / Vol / Issue, etc.	Prev. Vet. Med. 113, 565-579
		Publication Status	Accepted
		Publication website	<a href="http://www.sciencedirect.com/science/article/pii/S0167587714000178?via%3Dihub">http://www.sciencedirect.com/science/article/pii/S0167587714000178?via%3Dihub</a>
		Comment	0:1
	Language		English
	Software		R
	Programming language		R/knime
	Model category	Model Class	Dose-response model
		Model Sub-Class	Other
		Basic process	Infection with PRRS by ingestion of 0.5 kg of Pork (raw)
	Status		Uncurated
	Description		This is a dose-response model which describes the probability that pigs get infected with PRRSV. The model is based on a beta-poisson response function. It is considered to hold for a PRRSV dose from 0 to 8 log <sub>10</sub> TCID <sub>50</sub> /ml in residual blood serum in raw pork.
Scope	Product / matrix	Product/matrix name	Swine Meat
		Product/matrix description	Pork raw meat
		Product/matrix unit	Kg
		Country of origin	Philippines
	Hazard	Hazard type	Microorganisms
		Hazard name	Porcine Reproductive and Respiratory Syndrome Virus (PRRS)
		Hazard description	PRRS is caused by an RNA virus that has a high mutation rate, allowing rapid gain of viral fitness when freely transmitted in large host populations.
		Hazard unit	log <sub>10</sub> TCID <sub>50</sub> /ml
		Adverse effect	In 2006, a genetically distinct strain now known as "highly pathogenic PRRS virus" emerged in China, and caused 50–100% morbidity and 20–100% mortality in affected herds
		Source of contamination	Transmission of PRRS virus is possible via a variety of routes including orally by eating infected raw pork.
	Population Group	Population name	Feral or domestic pig
		Country	Australia
Model math / Data definition	Parameter / Factor / Input / Output / "Data column"	Parameter ID	1 2 3 4
		Parameter classification	1-Output 2-Input-hazard 3-Constant 4-Constant
		Parameter name	1- PInfectDose 2-Dose 3-Alpha 4-Beta
		Parameter description	1. PInfectDose is the probability that a pig gets infected given that it ingests a certain dose of PRRS virus. 2. Concentration of PRRS virus in residual blood in pork when ingested 3. Alpha = Parameter that describes the slope of the beta-poisson dose response function 4. Beta = Parameter which shifts the beta-poisson dose response function
		Parameter type	1.Number 2.Number 3.Number 4.Number

		Parameter unit	1. [] (no name) 2. log10 TCID50 3. [] (no name) 4. log10 TCID50
		Parameter unit category	1. Dimensionless Quantity 2. Number 3. Dimensionless Quantity 2. Number
		Parameter data type	1. Double 2. Double 3. Double 4. Double
		Parameter source	3. Hermann et al. (2005) 4. Hermann et al. (2005) <a href="http://www.sciencedirect.com/science/article/pii/S0167587714000178?via%3Dihub-bib0065">http://www.sciencedirect.com/science/article/pii/S0167587714000178?via%3Dihub-bib0065</a>
		Parameter subject	0:1
		Parameter distribution	0:1
		Parameter value	2. 4 log10 TCID50 3. 0.3 4. 14400 log10 TCID50
	Model equation	Model equation name	Beta-poisson response function
		Model equation / Script	$P(\text{Infect}   \text{Dose}) = 1 - \left(1 + \frac{\text{Dose}}{\text{beta}}\right)^{-\text{alpha}}$
		Level of contamination after left-censored data treatment	4 log10 TCID50
		type of exposure	Acute
		Scenario	An individual in the Philippines acquires 500 g of raw pork that may be infected with highly pathogenic PRRS virus, and transports the meat by air from Manila to Darwin, Australia. On arrival at a private household, all the meat is discarded and may be eaten by a domestic or feral pig.

Table 12 Example metadata

Corresponding JSON encoded metadata to Table 8.

```
{
  "generalInformation": {
    "name": "Dose Response Model for Porcine Reproductive And Respiratory Syndrome Virus",
    "source": "PUBLICED SCIENTIFIC STUDIES: Dose-response data and models",
    "identifier": "Dose-Response_Brookes_Pork_001",
    "creators": [
      {
        "creator": "[\"vcard\",[[\"version\",{}],\"text\", \"4.0\"],[\"prodid\",{}],\"text\", \"ez-vcard 0.10.2\"],[\"nickname\",{}],\"text\", \"V.J. Brookes\"],[\"email\",{}],\"text\", \"vbrookes@csu.edu.au\"]]"
      }
    ],
    "creationDate": 61372681200000,
    "modificationDate": [],
    "rights": "Copyright Â© 2014 Elsevier B.V.",
    "isAvailable": true,
    "url": null,
    "format": null,
    "reference": [
      {
        "reference": "TY - JOUR\r\nAB - Highly pathogenic porcine reproductive and respiratory syndrome (PRRS) has spread through parts of south-east Asia, posing a risk to Australia. The objective of this study was to assess the probability of infection of a feral or domestic pig in Australia with highly pathogenic PRRS following ingestion of illegally imported raw pork. A conservative scenario was considered in which 500 g of raw pork was imported from the Philippines into Australia without being detected by border security, then discarded from a household and potentially accessed by a pig. Monte Carlo simulation of a two-dimensional, stochastic model was used to estimate the probability of entry and exposure, and the probability of infection was assessed by incorporating a virus-decay and mechanistic dose-response model. Results indicated that the probability of infection of a feral pig after ingestion of raw meat was higher than the probability of infection of a domestic pig. Sensitivity analysis was used to assess the influence of input parameters on model output probability estimates, and extension of the virus-decay and dose-response model was used to explore the impact of different temperatures and time from slaughter to ingestion of the meat, different weights of meat, and the level of viraemia at slaughter on the infectivity of meat. Parameters with the highest influence on the model output were the level of viraemia of a pig prior to slaughter and the probability of access by a feral pig to food-waste discarded on

```

property surrounding a household. Extension of the decay and dose-response model showed that small pieces of meat (10 g) from a highly pathogenic PRRS viraemic pig could contain enough virus to have a high probability of infection of a pig, and that routes to Australia by sea or air from all highly pathogenic PRRS virus endemic countries were of interest dependent on the temperature of the raw meat during transport. This study highlighted the importance of mitigation strategies such as disposal of food-waste from international traffic as quarantine waste, and the need for further research into the probability of access to food-waste on properties by feral pigs. \r\nAU - Brookes VJ\r\nAU - HernÃ¡ndez-Jover M\r\nAU - Holyoake P\r\nAU - Ward MP\r\nDA - 2014-03-01\r\nDO - <https://doi.org/10.1016/j.prevetmed.2014.01.016>\r\nEP - 579\r\nLK - <http://www.sciencedirect.com/science/article/pii/S0167587714000178?via%3Dihub>\r\nSP - 565\r\nT2 - Prev. Vet. Med.\r\nTI - Import risk assessment incorporating a dose-response model: introduction of highly pathogenic porcine reproductive and respiratory syndrome into Australia via illegally imported raw pork.\r\nVL - 113\r\nVER - \r\n\r\n"

```

    },
    "language": "English",
    "software": "R",
    "languageWrittenIn": "R/KNIME",
    "modelCategory": {
      "modelClass": "Dose-response model",
      "modelSubClass": [
        "Other"
      ],
      "modelClassComment": null,
      "modelSubSubClass": [],
      "basicProcess": [
        "Infection with PRRS by ingestion of 0.5 kg of Pork (raw)"
      ]
    },
    "status": "Uncurated",
    "objective": null,
    "description": "This is a dose-response model which describes the probability that pigs get infected with PRRSV. The model is based on a beta-poisson response function. It is considered to hold for a PRRSV dose from 0 to 8 Log10 TCID50/mL in residual blood serum in raw pork."
  },
  "scope": {
    "product": {
      "environmentName": "Swine Meat",
      "environmentDescription": "Pork raw meat",
      "environmentUnit": "Kg",
      "productionMethod": [],
      "packaging": [],
      "productTreatment": [],
      "originCountry": "Philippines",
      "originArea": null,
      "fisheriesArea": null,
      "productionDate": null,
      "expirationDate": null
    },
    "hazard": {
      "hazardType": "Microorganisms",
      "hazardName": "Porcine Reproductive and Respiratory Syndrome Virus (PRRS)",
      "hazardDescription": "PRRS is caused by an RNA virus that has a high mutation rate, allowing rapid gain of viral fitness when freely transmitted in large host populations.",
      "hazardUnit": null,
      "adverseEffect": "In 2006, a genetically distinct strain now known as Å¢â,~Å“highly pathogenic PRRS virusÅ¢â,~i% emerged in China, and caused 50Å¢â,~â€œ100% morbidity and 20Å¢â,~â€œ100% mortality in affected herds",
      "origin": "Transmission of PRRS virus is possible via a variety of routes including orally by eating infected raw pork.",
      "benchmarkDose": null,
      "maximumResidueLimit": null,
      "noObservedAdverse": null,
      "lowestObservedAdverse": null,
      "acceptableOperator": null,
      "acuteReferenceDose": null,
      "acceptableDailyIntake": null,
      "hazardIndSum": null,
      "laboratoryName": null,
      "laboratoryCountry": null,

```

```

    "detectionLimit": null,
    "quantificationLimit": null,
    "leftCensoredData": null,
    "rangeOfContamination": null
  },
  "populationGroup": {
    "populationName": "Feral or domestic pig",
    "targetPopulation": null,
    "populationSpan": [],
    "populationDescription": [],
    "populationAge": [],
    "populationGender": null,
    "bmi": [],
    "specialDietGroups": [],
    "patternConsumption": [],
    "region": [],
    "country": [
      "Australia"
    ],
    "populationRiskFactor": [],
    "season": []
  },
  "generalComment": null,
  "temporalInformation": null,
  "region": [],
  "country": []
},
"dataBackground": null,
"modelMath": {
  "parameter": [
    {
      "id": "1",
      "classification": "output",
      "name": "PInfectDose",
      "description": "PInfectDose is the probability that a pig gets infected given that it ingests a certain dose of PRRS virus.",
      "unit": "[] (no name)",
      "unitCategory": "Dimensionless Quantity",
      "dataType": "Double",
      "source": null,
      "subject": null,
      "distribution": null,
      "value": null,
      "reference": null,
      "variabilitySubject": null,
      "modelApplicability": [],
      "error": null
    },
    {
      "id": "2",
      "classification": "input",
      "name": "Dose",
      "description": "Concentration of PRRS virus in residual blood in pork when ingested",
      "unit": "Log10 TCID50",
      "unitCategory": "Number",
      "dataType": "Double",
      "source": null,
      "subject": null,
      "distribution": null,
      "value": "4 Log10 TCID50",
      "reference": null,
      "variabilitySubject": null,
      "modelApplicability": [],
      "error": null
    },
    {
      "id": "3",
      "classification": "constant",
      "name": "Alpha",
      "description": "Alpha = Parameter that describes the slope of the beta-poisson dose response"
    }
  ]
}

```

```

function",
  "unit": "[ (no name)",
  "unitCategory": "Dimensionless Quantity",
  "dataType": "Double",
  "source": "Hermann et al. (2005)",
  "subject": null,
  "distribution": null,
  "value": "0.3",
  "reference": null,
  "variabilitySubject": null,
  "modelApplicability": [],
  "error": null
},
{
  "id": "4",
  "classification": "constant",
  "name": "Beta",
  "description": "Beta = Parameter which shifts the beta-poisson dose response function",
  "unit": "Log10 TCID50",
  "unitCategory": "Number",
  "dataType": "Double",
  "source": "Hermann et al. (2005)",
  "subject": null,
  "distribution": null,
  "value": "14400 Log10 TCID50",
  "reference": null,
  "variabilitySubject": null,
  "modelApplicability": [],
  "error": null
}
],
"sse": null,
"mse": null,
"rmse": null,
"rSquared": null,
"aic": null,
"bic": null,
"modelEquation": {
  "equationName": "Beta-poisson response function",
  "equationClass": null,
  "equationReference": [],
  "equation": "PInfectDose = 1 - (1 + Dose/Beta) ^ (-Alpha)"
},
"fittingProcedure": null,
"exposure": null,
"event": []
},
}

```



## References

- Bergmann, F., Adams, R., Moodie, S., Cooper, J., Glont, M., Golebiewski, M., et al. (2014). *COMBINE archive and OMEX format: one file to share all information to reproduce a modeling project*. (B. Bioinformatics, Ed.) Retrieved from <http://www.ncbi.nlm.nih.gov/pubmed/25494900>
- Bergmann, F., Cooper, J., Le Novère, N., Nickerson, D., & Waltermath, D. (2015). Simulation Experiment Description Markup Language (SED-ML) Level 1 Version 2. *Journal of Integrative Bioinformatics*, 262.
- Bergmann, F., Rodriguez, N., & Le Novère, N. (2015). *COMBINE Archive Specification Version 1*. Retrieved from <http://www.ncbi.nlm.nih.gov/pubmed/26528559>
- Cyganiak, R., Wood, D., Lanthaler, M., Klyne, G., Carroll, J. J., & McBride, B. (2014). *RDF 1.1 concepts and abstract syntax*. Retrieved from <https://www.w3.org/TR/rdf11-concepts/>
- Freed, N. a. (1996). *Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types*. Retrieved from [www.iana.org/assignments/media-types/media-types.xhtml](http://www.iana.org/assignments/media-types/media-types.xhtml)
- Hucka, M., Bergmann, F. T., Hoops, S., Keatin, S., Sahle, S., Schaff, J., et al. (2010). *The Systems Biology Markup Language (SBML): Language Specification for Level 3 Version 1 Core*. Retrieved from <http://identifiers.org/combine.specifications/sbml.level-3.version-1.core.release-1>
- O' Dada, J. (2011). *Numerical Markup Language*. Retrieved from <http://code.google.com>
- OMG. (2009, February). *UML 2.2 Superstructure and Infrastructure*. Retrieved from <http://www.omg.org/spec/uml/2.2>
- Waltermath, D., Adams, R., Beard, D., Bergmann, F., Bhalla, U., Britten, R., et al. (2011). *Minimum Information About a Simulation Experiment (MIASE)*. Retrieved from <http://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.1001122>
- Haberbeck, L., Plaza-Rodríguez, C., Desvignes, et al. (2018). Harmonized Terms, concepts and metadata for microbiological risk assessment models: The basis for knowledge integration and exchange. *Microbial Risk Analysis*. Volume 10. Pages 3-12.