Författare
2016-03-25 Sara Grolander, Mats Tröjbom

Kvalitetssäkring
2016-09-14 Helén Segerstedt (Tillstyrkan)
2016-09-14 Peter Larsson (Godkänd)

Guide to implementation of the SR-PSU Kd and CR parameterisation process in two Access databases

Detailed descriptions based on three examples

Contents

1 Introduction

The parameterisation of K_d and CR values in SR-PSU in done in two Microsaft Access databases. The databases were used in order to handle the vast number of data in an efficient and rational way. The objective with this document is to describe the implementation of the parameterisation process in the two Microsoft Access databases. This is done by giving three detailed examples that describe the implementation of the parameterisation process from beginning to end.
This guide follows the structure of Chapter 4 of Tröjbom et al. (2013) where the parameterisation method is described. The reader of this guide is recommended to read Chapter 4 as a complement to the descriptions of the implementation of the method given in this document. References are also made to the Appendix A,"Detailed technical description of database objects and code" which in high detail describes the technical aspects of the two Access databases.
This guide together with the technical description (Appendix A) of the database provides information and guidance to a reviewer that aims reviewing the implementation of the parameterisation in the two Access databases. This requires both knowledge of the parameterisation method described in Chapter 4 of Tröjbom et al. (2013) and general knowledge of Access databases.

1.1 Overview of the parameterisation process

The general assumptions made in the parameterisation work are described in Section 2.6 in Tröjbom et al. (2013), and the parameterisation methods are described in Chapter 4 in the same report. The parameterisation process is highly automated and is mainly based on rules and criteria that compares and selects data from different sources in order to achieve an as coherent dataset as possible. In addition, the parameterisation process contains subjective steps in the form of manual considerations and judgements. The examples in this document illustrate both of these aspects in order to give a description of how the parameterisation work was implemented in the two Access databases.
An overview of the parameterisation process is given in Section 4.1 in Tröjbom et al. (2013) and a schematic illustration is presented in Figure 4-1 in the same report. The parameterisation process can be divided in two major parts, each further divided into sub- steps, as listed below

1. Generation of the *K_d/CR data compilation* (described in Section 2). This part consists of two steps (listed below) and is implemented in a separate Access database named "SKB_chemistry_SR_PSU".
    a. Calculation of site-specific K_d and CR values (Section 2.1).
    b. Compilation of data from available literature sources (Section 2.2).

2. Evaluation and selection of data from the K_d/CR compilation and adjustments of the final parameter values (described in Section 3). This part is implemented in a separate Access database named "SKB_K_d_CR" and consists of four steps;
    a. Matching of data and ranking of literature sources (Section 3.1)
    b. Initial automated data selections based on statistical criteria (Section 3.2)
    c. Comparisons of data sources (Section 3.3)
    d. Manual evaluations and final data selection (Section 3.4).
    e.

1.2 Original files needed for reviewing the implementation of the parameterisation process in the two Access databases

It is recommended that the reviewer has access to the files and databases listed in Table 1-1.

Table 1-1.Files referred to in this document.

File name	Description
SKB_chemistry_SR_PSU.accdb	Access database for calculation of Kd and CR based on site data
SKB_Kd_CR.accdb	Access database for the Kd/CR selection
Coupling parameters to data_ver3.xlsx	Interface Excel-file where each parameter are coupled to appropriate site data.
ParamClass.xlsx	Interface Excel file where parameters are associated to specific site data and literature data sources. Sources are also ranked in this table.
FinalTable_cond_fix_PLOTB.xlsx	Excel file containing figures where Kd and CR are compared for elements across parameters.
Appendix A: Detailed technical description of database objects and code	A detailed description of the objects and codes of the two databases and instructions for updating the parameterisation chain.

1.3 Description of the three parameter examples

In total 2,139 unique K_d and CR parameters were delivered to the dose assessment model in SR-PSU. An overview of the parameterisation of these parameters is shown in Table 1-2

Table 1-2. In practice many of these parameters have been parameterised by the use of other parameters as analogues and the actual number of unique parameters is lower. In this document, the parameterisation of the three encircled parameters is described in detail:

- K_d_regoLow for Ni: The partitioning coefficient (K_d) for nickel (Ni) in the lower regolith (till). Site-specific Ni samples for deep till from Forsmark is used to parameterise this parameter.
- cR_Lake_fish for Np: The concentration ratio for neptunium (Np) for fish in freshwater lakes. As site-specific data for fish and lake water is lacking for Np, and literature data are scares, site-specific data for lanthanum (La) is utilised as an element analogue for the parameterisation of Np.
- cR_agri_veg for Cs: The concentration ratio for caesium (Cs) for vegetables on agricultural soils. Since site-specific data for caesium in vegetables is not available, literature data is utilised for this parameter.

Table 1-2. Overview of the parameterisation of the 2,139 parameters in SR_PSU (see Table 9-14 in Tröjbom et al. 2013 for a detailed explanation of the abbreviations: site data: SO, SV, SX, SM, element analogue: EA, parameter analogue: PA, element and parameter analogue: EP, literature data: LO, LV, LX. Parameters encircled in black constitute the three examples described in this document.

Parameter	Ac	Ag	Am	Ba	Ca	Cd	Cl	Cm	Co	Cs	Eu	Ho	I	Mo	Nb	Ni	Np	Pa	Pb	Pd	Po	Pu	Ra	Se	Sm	Sn	Sr	Tc	Th	U	Zr
cR_Lake_amph_NHB	EP	OV	EP	PA	LX	PA	PA	EP	PA	PA	EP	PA	PA	PA	PA	PA	EP	EP	LV	EP	PA	PA	EP	PA	PA	EP	PA	PA	PA	PA	PA
cR_Lake_bent_fish_NHB	EP	OV	EP	PA	PA	PA	PA	EP	PA	PA	EP	PA	PA	PA	PA	PA	EP	EP	OV	EP	PA	PA	EP	PA	PA	EP	PA	PA	PA	PA	PA
cR_Lake_bird_NHB	EP	OV	EP	PA	PA	PA	PA	EP	PA	PA	EP	PA	PA	PA	PA	PA	EP	EP	OV	EP	PA	LX	EP	PA	PA	EP	PA	PA	PA	PA	PA
cR_Lake_bivalve_NHB	EA	SX	EA	SX	SV	SV	SX	EA	SO	SO	SX	SX	SO	SV	SV	SX	EA	EA	SV	EA	LX	LV	EA	SO	SX	EA	SO	OV	SX	SM	SX
cR_Lake_Cray	EP	PA	EP	PA	PA	PA	PA	EP	PA	PA	PA	PA	PA	PA	PA	PA	EP	EP	PA	EP	LX	LX	LX	PA	PA	EP	PA	LX	PA	PA	PA
cR_Lake_crust_NHB	EP	PA	EP	PA	PA	PA	PA	EP	PA	PA	PA	PA	PA	PA	PA	PA	EP	EP	PA	EP	LX	LO	LO	PA	PA	EP	PA	LV	PA	PA	PA
cR_Lake_Fish	EA	LO	EA	SO	SO	LX	SO	EA	SO	SV	EA	SX	SO	SO	SV	S	EP	EA	LO	EA	LV	LV	EA	SV	SX	EA	SO	LX	OV	SM	SO
cR_Lake_Fish_NHB	EA	OV	EA	SO	SO	LV	SV	EA	SO	SV	EA	SX	SO	SO	SV	SO	EA	EA	OV	EA	LO	LO	EA	SV	SX	EA	SO	LO	LO	SM	SV
cR_Lake_gastr_NHB	EP	PA	EP	PA	PA	PA	PA	EP	PA	PA	PA	PA	PA	PA	PA	PA	EP	EP	PA	EP	LX	LX	LX	PA	PA	EP	PA	LX	PA	PA	PA
cR_Lake_ins_larvae_NHB	EP	PA	EP	PA	PA	PA	PA	EP	PA	PA	PA	PA	PA	PA	PA	PA	EP	EP	PA	EP	PA	PA	PA	PA	PA	EP	PA	LX	PA	PA	PA
cR_Lake_mammal_NHB	EP	OV	EP	PA	PA	PA	PA	EP	PA	PA	EP	PA	PA	PA	PA	PA	EP	EP	OV	EP	PA	LX	EP	PA	PA	EP	PA	PA	PA	PA	PA
cR_Lake_pel_fish_NHB	EP	OV	EP	PA	PA	PA	PA	EP	PA	PA	EP	PA	PA	PA	PA	PA	EP	EP	OV	EP	PA	PA	EP	PA	PA	EP	PA	PA	PA	PA	PA
cR_lake_pp_macro	EA	LX	EA	SX	SX	SO	SM	EA	SO	SO	SX	SM	SO	SX	SX	SO	EA	EA	SO	EA	LX	LO	EA	SO	SX	EA	SO	LV	SV	SO	SX
cR_lake_pp_micro	PA	PA	PA	PA	PA	PA	PA	EP	PA	EP	PA	EP	PA	PA	EP	PA	PA	PA	PA	PA	EP	EP	PA	EP	PA	EP	PA	PA	PA	PA	PA
cR_lake_pp_plank	EP	LX	EP	PA	PA	PA	PA	EP	PA	PA	PA	PA	PA	PA	PA	PA	EP	EP	PA	EP	LX	LX	EP	PA	PA	EP	PA	LX	PA	PA	PA
cR_Lake_pp_plank_NHB	EP	LV	EP	PA	PA	PA	PA	EP	PA	PA	PA	PA	PA	PA	PA	PA	EP	EP	PA	EP	LV	LO	EP	PA	PA	EP	PA	LV	PA	PA	PA
cR_Lake_pp_vasc_NHB	EP	LV	EP	PA	PA	PA	PA	EP	PA	PA	PA	PA	PA	PA	PA	PA	EP	EP	PA	EP	LV	OV	EP	PA	PA	EP	PA	OV	PA	PA	PA
cR_Lake_zoopl_NHB	EP	LX	EP	PA	PA	PA	PA	EP	LX	LX	PA	PA	LX	PA	PA	PA	EP	EP	LX	EP	PA	LX	PA	PA	PA	EP	EP	LX	OV	LX	PA
cR_Sea_bent_fish_NHB	EP	PA	EP	PA	PA	PA	PA	EP	PA	PA	EP	EP	PA	PA	PA	PA	EP	EP	PA	EP	PA	PA	EP	PA	PA	PA	PA	PA	PA	PA	PA
cR_Sea_bent_moll_NHB	EA	LX	EA	SX	SX	SO	SX	EA	SO	SO	SX	SX	SX	SX	SV	SV	EA	EA	SM	EA	LX	LX	LX	SV	SX	SX	EA	LX	SV	SV	SV
cR_Sea_bird_NHB	EP	LX	EP	PA	PA	PA	PA	EP	PA	PA	EP	EP	PA	PA	PA	PA	EP	EP	PA	EP	LV	LX	EP	PA	PA	PA	PA	PA	PA	LX	PA
cR_Sea_crust_NHB	EA	LX	EA	SX	SX	SV	LX	EA	SV	SV	SX	SX	LX	SX	LX	SV	EA	EA	SM	EA	LX	OV	LX	SV	SX	PA	EA	OV	SX	LX	SV
cR_Sea_Fish	EA	LX	EA	SX	SM	SV	SX	EA	SV	SO	EA	EA	SX	SX	SX	SV	EA	EA	SM	EA	LX	LO	EA	SV	SX	SX	SV	LX	SX	SV	SV
cR_Sea_Fish_NHB	EA	LO	EA	SX	SM	SO	SX	EA	SO	SO	EA	EA	SX	SX	SX	SV	EA	EA	SM	EA	LO	LO	EA	SV	SX	SX	SV	LO	SX	SV	SV
cR_Sea_mammal_NHB	EP	LX	EP	PA	PA	PA	PA	EP	PA	PA	EP	EP	PA	PA	PA	PA	EP	EP	PA	EP	LX	LX	LX	PA	PA	PA	PA	LX	PA	LX	PA
cR_Sea_pel_fish_NHB	EP	PA	EP	PA	PA	PA	PA	EP	PA	PA	EP	EP	PA	PA	PA	PA	EP	EP	PA	EP	PA	PA	EP	PA	PA	PA	PA	PA	PA	PA	PA
cR_Sea_polych_NHB	EP	LX	EP	PA	PA	LX	PA	EP	LX	LX	PA	PA	PA	PA	PA	LX	EP	EP	PA	EA	LX	LX	PA	LX	PA	PA	EP	OV	PA	PA	PA
cR_sea_pp_macro	EA	LO	EA	SX	SX	SV	SV	EA	SM	SV	SM	SM	SO	SX	SO	SO	EA	EA	SM	EA	LX	LO	EA	SV	SM	SX	SV	LO	SM	SV	SM
cR_Sea_pp_macro_NHB	EP	LO	EP	PA	PA	PA	PA	EP	PA	PA	PA	PA	PA	PA	PA	PA	EP	EP	PA	EP	LV	LV	EP	PA	PA	PA	PA	LV	PA	PA	PA
cR_sea_pp_micro	PA	PA	PA	PA	PA	PA	PA	PA	PA	EP	PA	EP	PA	PA	EP	PA	PA	PA	PA	PA	EP	EP	PA	EP	PA	EP	PA	PA	PA	PA	PA
cR_sea_pp_plank	EP	LX	EP	PA	PA	PA	PA	EP	PA	PA	PA	PA	PA	PA	PA	PA	EP	EP	PA	EP	LX	LX	EP	PA	PA	PA	PA	LX	PA	PA	PA
cR_Sea_pp_plank_NHB	EP	LO	EP	PA	PA	PA	PA	EP	PA	PA	PA	PA	PA	PA	PA	PA	EP	EP	PA	EP	LV	LV	EP	PA	PA	PA	PA	LV	PA	PA	PA
cR_Sea_pp_vasc_NHB	EP	PA	EP	PA	PA	PA	PA	EP	PA	PA	PA	PA	PA	PA	PA	PA	EP	EP	PA	EP	PA	PA	EP	PA	PA	PA	PA	PA	PA	PA	PA
cR_Sea_zoopl_NHB	EA	LX	EA	SX	SX	SX	LX	EA	OV	OV	SX	SX	SV	SX	EA	SX	EA	EA	SM	EA	LX	LX	LX	SX	SX	EA	EA	LX	OV	LX	SV
cR_agri_cereal	EA	SX	EA	SX	SV	SV	SV	EA	SO	SM	SX	SX	SO	SX	SX	SO	EA	EA	SO	EA	EA	EA	LO	SX	SX	SX	SO	EA	SV	SM	SX
cR_agri_fodder	EP	PA	EP	PA	PA	PA	PA	EP	PA	PA	PA	PA	PA	PA	PA	PA	EP	EP	PA	EP	EP	EP	PA	PA	PA	PA	PA	EP	PA	PA	PA
cR_agri_tuber	EA	OV	LO	EA	EA	EA	OV	LO	LO	LO	EA	EA	OV	OV	LV	PA	LO	EA	LO	EP	LO	LO	LO	PA	EA	EA	LO	LV	LO	LO	PA
cR_agri_veg	EA	LO	LO	EA	EA	EA	LX	LO	LO	LO	EA	EA	LX	LX	LV	PA	LO	EA	LO	EP	LV	LO	LO	PA	EA	EA	LO	EP	LO	LO	LV
cR_food_herbiv	EA	LX	EA	SX	SX	SX	SX	EA	SX	SX	SX	SX	SM	SX	SX	SX	EA	EA	SX	EA	LX	LX	SX	SX	SX	SX	SX	EA	SX	SX	SX
cR_Ter_amph_NHB	EP	LX	EP	EA	EA	LO	PA	EP	PA	LO	EP	EP	PA	PA	PA	PA	EP	EP	LO	EP	OV	OV	PA	PA	PA	PA	LO	LX	PA	PA	PA
cR_Ter_bird_egg_NHB	EP	PA	EP	EP	EP	PA	PA	EP	PA	LX	EP	EP	LX	PA	LX	PA	EP	EP	PA	EP	OV	PA	PA	PA	PA	PA	LX	LX	PA	LX	PA
cR_Ter_bird_NHB	EP	PA	EP	EP	EP	PA	PA	EP	PA	LO	EP	EP	PA	PA	OV	PA	EP	EP	LX	EP	OV	LV	LV	PA	PA	PA	LO	LV	LX	LX	PA
cR_Ter_detr_inv_NHB	OV	LX	OV	OV	OV	LO	LO	LX	LX	LO	LV	OV	LO	EA	LV	LV	EP	EA	LO	EA	LV	LX	LX	LV	OV	EA	OV	LX	LX	OV	LX
cR_Ter_fl_ins_NHB	EA	LX	LX	EA	EA	LX	PA	PA	LX	LX	LX	EA	LX	EA	LX	LX	EA	EA	LX	EA	PA	LX	LX	LX	EA	EA	LX	LX	LX	LX	LX
cR_Ter_gastr_NHB	EA	LX	LX	EA	EA	LX	LX	PA	LX	LX	LX	EA	LX	EA	LX	LX	EP	EA	LX	EA	PA	LX	LX	LX	EA	EA	LX	LX	LX	LX	LX
cR_Ter_mammal_large_NHB	EP	LX	EP	PA	PA	PA	PA	EP	PA	PA	EP	EP	PA	PA	PA	PA	EP	EP	PA	EP	LX	LO	PA	PA	PA	PA	PA	LX	PA	PA	PA
cR_Ter_mammal_small_NHB	EP	LX	EP	PA	PA	PA	PA	EP	PA	PA	EP	EP	PA	PA	PA	PA	EP	EP	PA	EP	OV	LO	PA	PA	PA	PA	PA	LX	PA	PA	PA
cR_Ter_Mush	EP	EA	EP	EA	SX	SX	PA	EP	SX	SV	PA	PA	SX	EA	PA	SX	EP	EP	SV	EA	EP	EA	EA	PA	PA	EA	SV	EP	SV	SM	PA
cR_Ter_pp	EA	PA	EA	SV	SX	SV	SO	EA	SO	SM	SX	SX	SO	SX	SX	SO	EA	EA	SO	EA	EP	EA	SO	PA	SX	SX	SO	EP	SO	SO	SX
cR_Ter_pp_grass_NHB	EP	PA	EP	PA	PA	PA	PA	EP	PA	PA	PA	PA	PA	PA	PA	PA	EP	EP	PA	EP	EP	EP	PA	PA	PA	PA	PA	EP	PA	PA	PA
cR_Ter_pp_lich_NHB	EA	SX	EA	SX	SX	SX	SX	EA	SX	SV	SX	SX	SX	SX	SX	SX	EA	EA	SV	EA	LX	EA	SV	SX	SX	SX	SV	LX	SV	SM	SX
cR_Ter_pp_NHB	EA	LV	EA	SV	SX	SO	SO	EA	SO	SM	SX	SX	SO	SX	SX	SO	EA	EA	SO	EA	EP	LO	SO	LO	SV	SX	SO	EP	SO	SV	SV
cR_Ter_pp_shrub_NHB	EP	PA	EP	PA	PA	PA	PA	EP	PA	PA	PA	PA	PA	PA	PA	PA	EP	EP	PA	EP	EP	EP	PA	PA	PA	PA	PA	EP	PA	PA	PA
cR_Ter_pp_tree_NHB	EP	PA	EP	PA	PA	PA	PA	EP	PA	PA	PA	PA	PA	PA	PA	PA	EP	EP	PA	EP	EP	EP	PA	PA	PA	PA	PA	EP	PA	PA	PA
cR_Ter_rept_NHB	EP	LX	EP	PA	PA	PA	PA	EP	PA	PA	EP	EP	PA	PA	PA	PA	EP	EP	PA	EP	OV	OV	PA	PA	PA	PA	PA	LX	PA	PA	PA
cR_Ter_soil_inv_NHB	EA	LX	LX	EA	EA	LX	LX	PA	LX	LX	LX	EA	LX	EA	LX	LX	EA	EA	LX	EA	PA	LX	LX	LX	EA	EA	LX	LX	LX	LX	LX
K_d_PM_lake	EA	SV	EA	SO	SX	SX	EA	EA	SO	SO	SX	SX	SO	SX	SX	SX	EA	EA	SX	EA	EA	EA	EA	SX	SX	EA	SV	EA	SO	SM	SX
K_d_PM_sea	EA	PA	EA	SM	SM	SM	SM	EA	SX	SX	SX	SX	SX	SM	SX	SX	EA	EA	SX	EA	EA	EA	EA	SX	SX	SX	SX	EA	SX	SX	SX
K_d_regoGL	EA	SO	EA	SX	SV	EA	SV	EA	SV	SV	SX	SV	SV	SV	SV	SV	EA	EA	SV	EA	EA	EA	SO	SO	SV	SO	SO	EA	SO	SM	SO
K_d_regoLow	EA	SO	EA	SX	SV	SM	SO	EA	SO	SO	SX	SV	SO	SO	S	SO	EA	EA	SO	EA	EA	EA	SO	SO	SV	SO	SO	EA	SO	SO	SO
K_d_regoPeat	EA	SO	EA	SX	SV	SM	SV	EA	SO	SO	SX	SV	SV	SV	SO	SO	EA	EA	SO	SV	EA	EA	SO	SM	SV	SO	SO	EA	SO	SO	SO
K_d_regoPG	EA	SO	EA	SM	SO	SO	SV	EA	SO	SV	SX	SO	SV	SV	SV	SO	EA	EA	SO	EA	EA	EA	SO	SO	SO	SM	SO	EA	SO	SO	SO
K_d_regoUp_aqu	EA	SV	EA	SO	SX	SX	SX	EA	SO	SO	SX	SX	SO	SX	SX	SX	EA	EA	SX	EA	EA	EA	EA	SX	SX	SX	SV	EA	SO	SX	SX
K_d_regoUp_drain	EA	SO	EA	SX	SV	SM	SV	EA	SO	SO	SX	SO	SO	SV	SO	SO	EA	EA	SO	SV	EA	EA	SO	SO	SV	SO	SO	EA	SO	SO	SO
K_d_regoUp_garden	EA	SO	EA	SX	SV	SO	SV	EA	SO	SV	SX	SV	SV	SO	SV	SO	EA	EA	SM	SV	EA	EA	SO	SO	SV	SO	SO	EA	SO	SO	SO
K_d_regoUp_io	EA	SO	EA	SX	SV	SO	SV	EA	SO	SV	SX	SV	SV	SO	SV	SO	EA	EA	SM	SV	EA	EA	SO	SO	SV	SO	SO	EA	SO	SO	SO
K_d_regoUp_ter	EA	SV	EA	SX	SV	SV	SV	EA	SO	SO	SX	SV	SO	SV	SO	SO	EA	EA	SO	EA	EA	EA	SO	SO	SV	SO	SO	EA	SO	SO	SO
TC_meat	EA	EA	LX	LX	LX	LX	LX	EA	LX	LX	EA	EA	LX	LX	LX	EA	EA	EA	LX	EA	EA	LX	LX	EA	EA	EA	LX	EA	LX	LX	LX
TC_milk	EA	EA	LX	LX	LX	LX	EA	EA	LX	LX	EA	EA	LX	LX	LX	LX	EA	EA	LX	EA	LX	LX	LX	LX	EA	EA	LX	EA	EA	LX	LX

2 Generation of the K_d/CR data compliation

A common database where all site-specific K_d and CR values from the Forsmark and Laxemar sites are compiled together with K_d and CR data from literature sources is created in order to make it possible to work with all available data in an efficient way. This compilation of data is called the K_d/CR compilation here. The K_d/CR compilation is created in the Access database "SKB_chemistry_SR_PSU.accdb". The methods used for calculation of site-specific K_d/CR and the literature data compiled are described in Section 4.2 in Tröjbom et al. (2013). In this chapter a detailed description of the how the process of generating the K_d/CR compilation is implemented in the Access database is given. The chapter follows the structure of Section 4.2 in Tröjbom et al. (2013) where the general method is described. It is recommended that this section is read together with this chapter.

2.1 Estimation of site-specific K_d and CR

The first step in the parameterisation process couples concentration samples of soil, water and biota from the sites (Forsmark and Laxemar) to K_d and CR parameters and calculates K_d and CR ratios from the sample pairs formed. This is achieved in five major steps:

1) The parameters are coupled to suitable categories of site specific concentration data in an excel file (Coupling parameters to data_ver3.xlsx). This Excel file serves as the interface onto the Access database where the manual coupling of parameters to data takes place. This file lists all available data categories in SKB site data[1], divided into three separate sheets dealing with Biota, Regolith and Water. The categories of data that is judged to be representative for a K_d or CR parameter is selected. The selection of data is done per category and not per individual samples. This two-step selection process via categories and not individual samples is carried out in order to facilitate the work and reduce the risk for manual errors when handling the large amounts of data.
2) Individual samples not to be included are manually excluded in the sheets Details_Biota, Details_Regolith, Details_water of the file "Coupling parameters to data_ver3.xlsx". For example replicates or erroneous data. All discarded samples are listed in Appendix E in Tröjbom et al. (2013).
3) The information in "Coupling parameters to data_ver3.xlsx" is imported to the "SKB_Chemistry _SR_PSU" Access database according to the Appendix A Section 2.1.1.
4) After import of the file "Coupling parameters to data_ver3.xlsx" to the "SKB_Chemistry _SR_PSU", all individual samples that correspond to the coupling categories are merged in two tables representing the numerator and denominator concentrations of the K_d and CR ratios to be formed. In these tables parameter names are linked to individual sample numbers (SKB sample number) representative for the specific parameter. The two macros "skapa_Valuelist_to" and "skapa_Valuelist_from" creates the two tables "ValueList_to" and "ValueList_from", where all individual samples (SKB sample number) are listed per parameter (cf. Appendix A Section 2.2.2).
5) Data pairs are combined from the tables described above and statistics are calculated per parameter forming the table "Export_K_dCR_database", which compiles all site-specific K_d and CR values into one table.

[1] The site-specific concentration data originates from the Excel-file "**SKB_Chemistry_SR_PSU.xlsx**" that is based on traceable data from the SKB SICADA database. This file has been quality assured (SKB doc 1320110, SKBdoc1327875) and is used as the original chemistry data file within the SR-PSU project.

2.1.1 RegoLow / Ni example

The RegoLow parameter is best represented with samples from the deep till in Forsmark. The regolith (sorbed) and the pore water concentrations are combined into ratios in order to estimate K_d values. In the file "Coupling parameters to data_ver3.xlsx / sheet Regolith", the regoLow parameter is assigned to the inorganic till categories from Sheppard et al. (2011) (ID_soil 88, 89, 90, 91, 92, 93), in the "Tox"-column. The corresponding pore water categories are selected in the "Water"-sheet in the "Fromx"-column (ID_water 26, 27, 28, 29).

The individual samples that correspond to the regolith (sorbed) and pore water categories are listed inTable 2-1, which is an excerpt from the "Details_Regolith" and "Details_water" sheets, respectively. In the red-coloured rows analytical replicates that are excluded are marked by an "x" in the "Excl1"-column.

In the case of RegoLow, the actually selected samples for Ni based on the information in the file "Coupling parameters to data_ver3.xlsx " are listed in Table 2-2 for the dissolved (excerpt from table "ValueList_from") and sorbed fractions (excerpt from table "ValueList_to"). The red-marked rows denote later excluded samples, which are deselected in the "Details_soil_excl" and "Details_water_excl" database tables (originating from the Coupling parameters to data_ver3.xlsx file, cf. Appendix A Section 2.1.2 and 2.2.3).

Table 2-1. Excerpts from the "Details_regolith" sheet (upper) and the "Details_water" sheet (lower) in the file "Coupling parameters to data_ver3.xlsx".

ID_soil	Excl1	Excl2	Excl3	Excl4	ExComment	ID	SR-Site	SITE	IDCODE	SKB_SAMPLE_NO	SUB_SAMPLE_NO
88						1201	FALSKT	FORSMARK	PFM007693	23480	Sample 5
89						1199	FALSKT	FORSMARK	PFM007692	23478	Sample 3
90						1200	FALSKT	FORSMARK	PFM007692	23479	Sample 4
90						1202	FALSKT	FORSMARK	PFM007693	23481	Sample 6
90	x				dubbelprov, struket	1203	FALSKT	FORSMARK	PFM007693	23482	Sample 6
91						1196	FALSKT	FORSMARK	PFM007690	23475	Sample 1
91	x				dubbelprov, struket	1197	FALSKT	FORSMARK	PFM007690	23476	Sample 1
92						1204	FALSKT	FORSMARK	PFM007694	23483	Sample 7
92	x				dubbelprov, struket	1205	FALSKT	FORSMARK	PFM007694	23484	Sample 8
93						1198	FALSKT	FORSMARK	PFM007691	23477	Sample 2

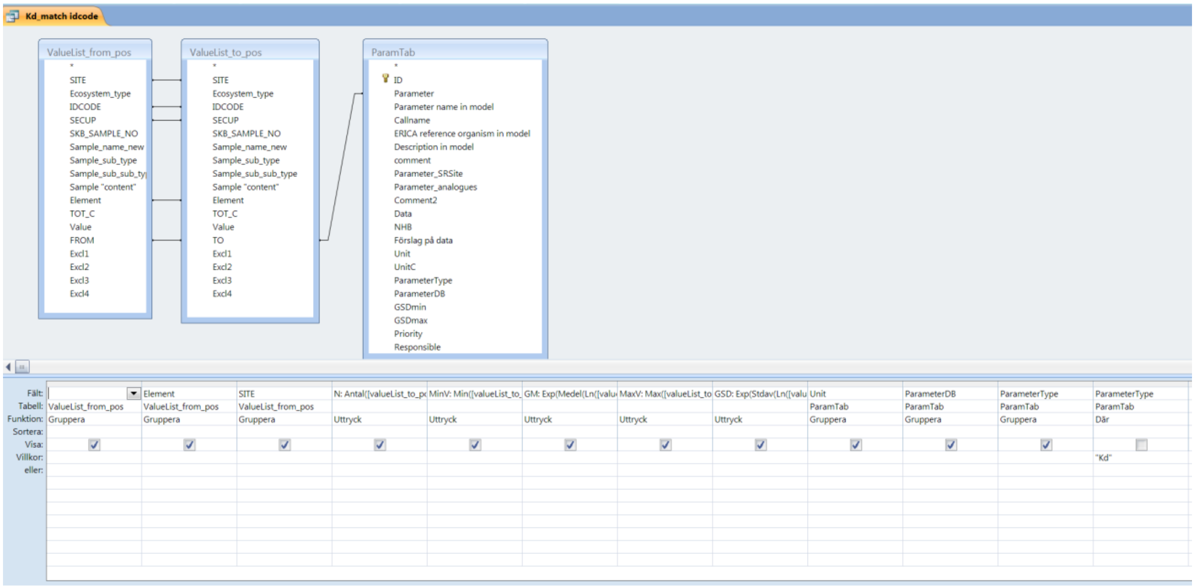
ID_wat	Excl1	Excl2	Excl3	Excl4	ExCommen	ID	SR-Site	SITE	IDCODE	SKB_SA	SUB_SAMPLE_NO
26						8551	FALSKT	FORSMAR	PFM007692	23539	Sample 3
26						8553	FALSKT	FORSMAR	PFM007693	23541	Sample 5
27						8549	FALSKT	FORSMAR	PFM007690	23537	Sample 1
27						8552	FALSKT	FORSMAR	PFM007692	23540	Sample 4
27						8554	FALSKT	FORSMAR	PFM007693	23542	Sample 6
28						8555	FALSKT	FORSMAR	PFM007694	23543	Sample 7
28	x				Dubbelprov, s	8556	FALSKT	FORSMAR	PFM007694	23544	Sample 8
29						8550	FALSKT	FORSMAR	PFM007691	23538	Sample 2

Table 2-2. The selected samples for Ni for the K_d_regoLow parameter in the tables ValueList_from (dissolved) and ValueList_to (sorbed). The value-field represents concentration in µg/l for pore water and mg/kgdw for regolith samples.

Dissolved fraction (porewater)												
SITE	Ecosystem_type	IDCODE	SECUP	SKB_SAMPLE_NO	Sample_name_new	Sample_sub_type	Sample_sub_sub_type	Sample "content"	Element	TOT_C	Value	FROM
FORSMARK	Terrester	PFM007690	1.8	23537	porewater soil_0.01	Pore water	filtered pore water	n	Ni		5.26	kD_regoLow
FORSMARK	Terrester	PFM007691	3.5	23538	porewater soil_0.03	Pore water	filtered pore water	n	Ni		4.9	kD_regoLow
FORSMARK	Terrester	PFM007692	0.5	23539	porewater soil_0	Pore water	filtered pore water	n	Ni		5.03	kD_regoLow
FORSMARK	Terrester	PFM007692	1	23540	porewater soil_0.01	Pore water	filtered pore water	n	Ni		2.55	kD_regoLow
FORSMARK	Terrester	PFM007693	0.3	23541	porewater soil_0	Pore water	filtered pore water	n	Ni		8.09	kD_regoLow
FORSMARK	Terrester	PFM007693	1	23542	porewater soil_0.01	Pore water	filtered pore water	n	Ni		4.16	kD_regoLow
FORSMARK	Terrester	PFM007694	2.5	23543	porewater soil_0.02	Pore water	filtered pore water	n	Ni		5.21	kD_regoLow
FORSMARK	Terrester	PFM007694	2.5	23544	porewater soil_0.02	Pore water	filtered pore water	n	Ni		6.31	kD_regoLow
Sorbed fraction (regolith)												
SITE	Ecosystem_type	IDCODE	SECUP	SKB_SAMPLE_NO	Sample_name_new	Sample_sub_type	Sample_sub_sub_type	Sample "content"	Element	TOT_C	Value	TO
FORSMARK	Terrester	PFM007690	1.8	23475	centrifugated soil_1.8	Inorganic	Till	n	Ni		3.259	kD_regoLow
FORSMARK	Terrester	PFM007690	1.8	23476	centrifugated soil_1.8	Inorganic	Till	n	Ni		3.153	kD_regoLow
FORSMARK	Terrester	PFM007691	3.5	23477	centrifugated soil_3.5	Inorganic	Till	n	Ni		3.082	kD_regoLow
FORSMARK	Terrester	PFM007692	0.5	23478	centrifugated soil_0.5	Inorganic	Till	n	Ni		4.983	kD_regoLow
FORSMARK	Terrester	PFM007692	1	23479	centrifugated soil_1.0	Inorganic	Till	n	Ni		4.156	kD_regoLow
FORSMARK	Terrester	PFM007693	0.3	23480	centrifugated soil_0.3	Inorganic	Till	n	Ni		5.005	kD_regoLow
FORSMARK	Terrester	PFM007693	1	23481	centrifugated soil_1.0	Inorganic	Till	n	Ni		2.901	kD_regoLow
FORSMARK	Terrester	PFM007693	1	23482	centrifugated soil_1.0	Inorganic	Till	n	Ni		3.507	kD_regoLow
FORSMARK	Terrester	PFM007694	2.5	23483	centrifugated soil_2.5	Inorganic	Till	n	Ni		3.764	kD_regoLow
FORSMARK	Terrester	PFM007694	2.5	23484	centrifugated soil_2.5	Inorganic	Till	n	Ni		3.432	kD_regoLow

Samples from the tables “ValueList_from” and “Valuelist_to” are combined into data pairs in the query “K_d_match_idcode” where the statistics are calculated based on these pairs (cf. Appendix A Section 2.2.5). In case of K_d only samples from the same geographical location (IDCODE) and the same sample depth (SECUP) are combined according to the query structure in Figure 2-1 and the corresponding SQL expression shown below. Since the analytical replicates were omitted prior to the matching the actual pairs formed therefore represent unique sampling occasions at specific locations and depth. From the query expression could be concluded that if there are more than one sample representing the same IDCODE and SECUP, data pairs will be formed for all possible permutations of the samples with identical IDCODE and SECUP combinations. In the SR-PSU parameterisation this only occurred for one sample where two different digestions methods were used to measure the sorbed fraction of the marine sediment sample PFM006045 (SKB sample no 23546 and 23548).

Figure 2-1. Screen shot showing the structure of the query "K_d_match_idcode" where data are combined into pairs and statistics are calculated. The corresponding SQL code is shown below the table.



SELECT ValueList_from_pos.FROM, ValueList_from_pos.Element, ValueList_from_pos.SITE, Count([valueList_to_pos].[value]/[valuelist_from_pos].[value]) AS N, Min([valueList_to_pos].[value]/[valuelist_from_pos].[value]) AS MinV, Exp(Avg(Log([valueList_to_pos].[value]/[valuelist_from_pos].[value]))) AS GM, Max([valueList_to_pos].[value]/[valuelist_from_pos].[value]) AS MaxV, Exp(StDev(Log([valueList_to_pos].[value]/[valuelist_from_pos].[value]))) AS GSD, ParamTab.Unit, ParamTab.ParameterDB, ParamTab.ParameterType
FROM (ValueList_from_pos INNER JOIN ValueList_to_pos ON (ValueList_from_pos.SECUP = ValueList_to_pos.SECUP) AND (ValueList_from_pos.Element = ValueList_to_pos.Element) AND (ValueList_from_pos.FROM = ValueList_to_pos.TO) AND (ValueList_from_pos.IDCODE = ValueList_to_pos.IDCODE) AND (ValueList_from_pos.SITE = ValueList_to_pos.SITE)) INNER JOIN ParamTab ON ValueList_to_pos.TO = ParamTab.Parameter
WHERE (((ParamTab.ParameterType)="K_d"))
GROUP BY ValueList_from_pos.FROM, ValueList_from_pos.Element, ValueList_from_pos.SITE, ParamTab.Unit, ParamTab.ParameterDB, ParamTab.ParameterType;

The resulting combinations in case of K_d_RegoLow for Ni are listed in

Table 2-3[2]. Seven unique sample pairs are formed for the K_d_regoLow parameter as indicated by the "Sample_No" columns. The estimates calculated from these values are identical to the final parameter values for Ni in the table "Export_K_dCR_database" (0.79, 0.79 and 1.4, respectively)[3]. Cf. Appendix A, Section 2.3.1 for description of the export file.

Table 2-3. The individual sample pairs and resulting K_d values for K_d_Regolow/Ni. From query "K_d_match_idcode_persample" in the SR_PSU_chemistry database.

[2] In the query "K_d_match_idcode_persample", the individual K_d values for Ni could be checked.
[3] In this example the geometric mean (GM) of sample pairs only from Forsmark (0.79), GM for Forsmark and Laxemar data combined (0.79) is identical due to lacking Laxemar data, and the geometrical standard deviation (GSD) is 1.4 for the same sample pairs.

Parameter	Sample_No_from	Sample_No_to	SITE	IDCODE	SECUP	Unit	Element	Kd
kD_regoLow	23539	23478	FORSMARK	PFM007692	0.5	m3/kgdw	Ni	0.990656064
kD_regoLow	23541	23480	FORSMARK	PFM007693	0.3	m3/kgdw	Ni	0.618665019
kD_regoLow	23537	23475	FORSMARK	PFM007690	1.8	m3/kgdw	Ni	0.619581749
kD_regoLow	23540	23479	FORSMARK	PFM007692	1	m3/kgdw	Ni	1.629803922
kD_regoLow	23542	23481	FORSMARK	PFM007693	1	m3/kgdw	Ni	0.697355769
kD_regoLow	23543	23483	FORSMARK	PFM007694	2.5	m3/kgdw	Ni	0.722456814
kD_regoLow	23538	23477	FORSMARK	PFM007691	3.5	m3/kgdw	Ni	0.628979592

2.1.2 cR_Lake_fish / Np example (La element analogue)

As there are no site-specific data for neptunium (Np), the cR_Lake_fish parameter is parameterised using analogues and literature data (cf. Section 2.2). According to Table 7-20 in Tröjbom et al. (2013) lanthanum (La) is utilised as element analogue for Np for cR_Lake_fish[4], and therefore the selection of site data for La is described in this section.

The cR_Lake_fish parameter is calculated from paired measurements of fish concentrations and water concentrations. In the file "Coupling parameters to data_ver3.xlsx" sheet "Biota" ten categories of site-specific fish samples are coupled to the cR_Lake_fish parameter (

[4] La is also utilised as element analogue for cR_Lake_fish_NHB, for which the selected data are identical to the cR_Lake_fish parameter. It is not clear why cR_Lake_fish_NHB was selected as parameter analogue for cR_Lake_fish in the parameterisation. This is probably a typing error. The actual CR values are not affected by selecting cR_Lake_fish _NHB as parameter analogue instead of using only La for cR_Lake_fish as element analogue directly.

Table 2-5). The corresponding lake water categories that represent filtered lake water from Forsmark and Laxemar/Simpevarp are listed in Table 2-4.

Table 2-4. The water sample categories coupled to the cR_Lake_fish parameter.

ID_water	SITE	Sample_name_new	Ecosystem_type	Sample_type	Sample_sub_type
2	FORSMARK	filtered water	Limnic	Water	Fresh water
59	SIMPEVARP	filtered water	Limnic	Water	Fresh water

Table 2-5. Excerpt from the "Coupling parameters to data_ver3.xlsx" file sheet "Biota" showing the categories coupled to the cR_Lake_fish parameter.

ID_biota	SITE	Ecosystem_type	Sample_type	Sample_sub_type	Sample_sub_sub_type	Sample "content"	Latin name	Sample preparation
3	FORSMARK	Limnic	Biota	animal	Fish	muscle	Esox lucius	dissolution of the ashed samples in HNO3/HCl/HF aliquots
4	FORSMARK	Limnic	Biota	animal	Fish	muscle	Esox lucius	total analysis
5	FORSMARK	Limnic	Biota	animal	Fish	muscle	Gymnocephalus cernuus	total analysis
99	SIMPEVARP	Limnic	Biota	animal	Fish	muscle	Perca fluviatilis	total analysis
100	SIMPEVARP	Limnic	Biota	animal	Fish	muscle	Perca fluviatilis	dissolution of the ashed samples in HNO3/HCl/HF aliquots
101	SIMPEVARP	Limnic	Biota	animal	Fish	muscle	Perca fluviatilis	dissolution of the ashed samples in HNO3/HCl/HF aliquots
6	FORSMARK	Limnic	Biota	animal	Fish	muscle	Rutilus rutilus	dissolution of the ashed samples in HNO3/HCl/HF aliquots
7	FORSMARK	Limnic	Biota	animal	Fish	muscle	Rutilus rutilus	total analysis
102	SIMPEVARP	Limnic	Biota	animal	Fish	muscle	Rutilus rutilus	total analysis
8	FORSMARK	Limnic	Biota	animal	Fish	muscle	Tinca tinca	total analysis

The selected fish categories correspond to 32 individual fish samples from Forsmark and Laxemar/Simpevarp according to Table 2-7. No exclusions were made in the "Details_Biota" sheet among these samples. Sample specific carbon contents[5] were available for all fish samples with La data. For each individual sample the carbon content was utilised to normalise the element concentration to carbon prior to the paring of samples into of CR ratios as described below. As indicated in Table 9-7 in Tröjbom et al. (2013), a large fraction of the La measurements of the fish samples falls below reporting limits (59%). This is a warning that the final CR values might be biased because several samples are omitted when the site-specific values are calculated. For other possible analogues (e.g. other REE), there is even fewer data, leaving La as the best choice.

The corresponding lake water samples constitute 3 representative samples from Forsmark and 3 from Laxemar/Simpevarp according to Table 2-6, where all data are above the reporting limits for La.

Table 2-6. The selected lake water samples from Forsmark and Laxemar/Simpevarp for La.

ID_water	SITE	IDCODE	SKB_SAMPLE_NO	SAMPLE_DATE	Ecosystem_type	Sample_type	Sample_sub_type	Sample_sub_sub_type	La (µg/l)
2	FORSMARK	PFM000117	23117	2008-04-08	Limnic	Water	Fresh water	filtered water	0.064
2	FORSMARK	PFM000107	23118	2008-04-08	Limnic	Water	Fresh water	filtered water	0.1
2	FORSMARK	PFM000074	23120	2008-04-08	Limnic	Water	Fresh water	filtered water	0.074
59	SIMPEVARP	PSM002067	23262	2008-04-21	Limnic	Water	Fresh water	filtered water	2.3
59	SIMPEVARP	PSM002066	23263	2008-04-21	Limnic	Water	Fresh water	filtered water	0.52
59	SIMPEVARP	PSM002065	23265	2008-04-21	Limnic	Water	Fresh water	filtered water	1.8

[5] In the total dataset 21 site specific biota samples were lacking sample specific carbon data, see query "Missing_Carbon" and 2.1.4.

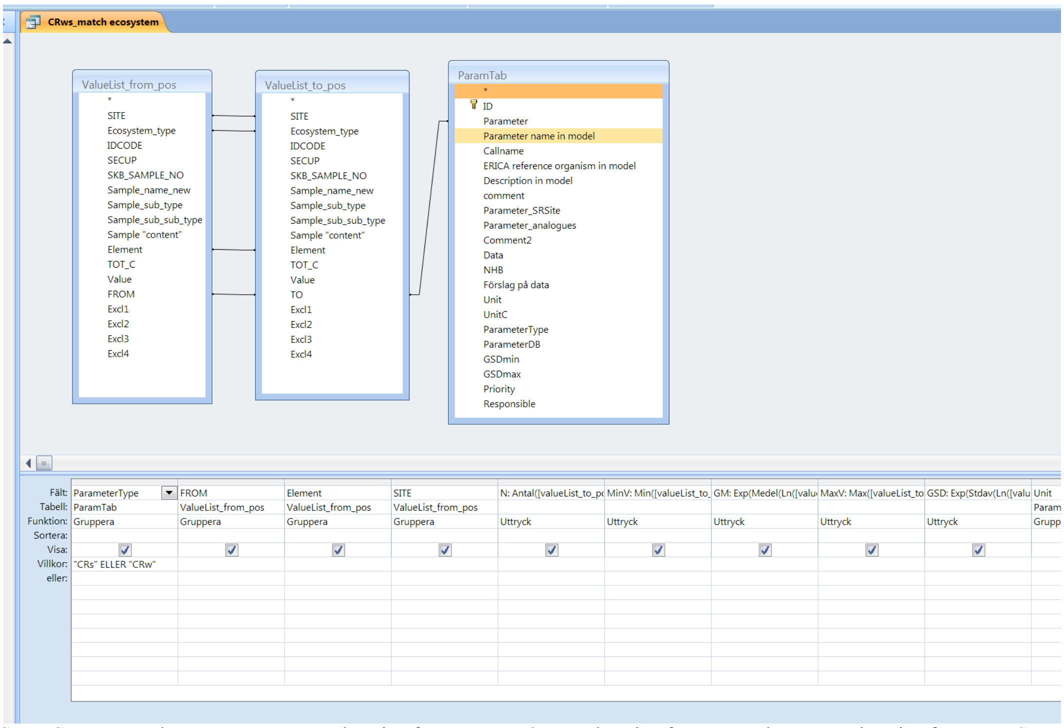
Table 2-7. The selected fish samples from Forsmark and Laxemar/Simpevarp for La. In the calculation of CR the values below reporting limits are omitted. Excerpt from the file "Coupling parameters to data_ver3.xlsx" sheet "Details_biota".

ID_biota	SITE	IDCODE	SKB_SAMPLE_NO	SUB_SAMPLE_NO	START_DATE	Sample description	Sample preparation	La (mg/kgdw)	TOT_C (mg/kgdw)
3	FORSMARK	AFM000050	23607	18	2006-12-08	Esox lucius, muscle	dissolution of the ashed samples in HNO3/HCl/HF aliquots		
4	FORSMARK	AFM000051	23601	5	2005-12-02	Esox lucius, muscle	total analysis	-0.0002	430000
4	FORSMARK	AFM000010	23602	6	2005-12-02	Esox lucius, muscle	total analysis	0.001	430000
4	FORSMARK	AFM000010	23603	6	2005-12-02	Esox lucius, muscle	total analysis		430000
4	FORSMARK	AFM000010	23604	5	2005-12-02	Esox lucius, muscle	total analysis	-0.0002	450000
4	FORSMARK	AFM000010	23605	6	2005-12-02	Esox lucius, muscle	total analysis		430000
4	FORSMARK	AFM000050	23606	5	2005-12-02	Esox lucius, muscle	total analysis	0.0004	420000
5	FORSMARK	AFM000050	23608	1	2005-12-02	Gymnocephalus cernuus, muscle	total analysis	-0.001	450000
6	FORSMARK	AFM000050	23634	17	2006-12-08	Rutilus rutilus, muscle	dissolution of the ashed samples		482000
7	FORSMARK	AFM000051	23635	1	2005-12-02	Rutilus rutilus, muscle	total analysis	-0.002	440000
7	FORSMARK	AFM000010	23637	1	2005-12-02	Rutilus rutilus, muscle	total analysis	-0.001	440000
7	FORSMARK	AFM000010	23638	1	2005-12-02	Rutilus rutilus, muscle	total analysis		440000
7	FORSMARK	AFM000010	23639	1	2005-12-02	Rutilus rutilus, muscle	total analysis		440000
8	FORSMARK	AFM000050	23668	3	2005-12-02	Tinca tinca, muscle	total analysis	-0.0002	430000
8	FORSMARK	AFM000050	23669	4	2005-12-02	Tinca tinca, muscle	total analysis	-0.0002	440000
8	FORSMARK	AFM000051	23670	2	2005-12-02	Tinca tinca, muscle	total analysis	-0.0002	450000
8	FORSMARK	AFM000051	23671	4	2005-12-02	Tinca tinca, muscle	total analysis	0.0002	430000
8	FORSMARK	AFM000051	23672	3	2005-12-02	Tinca tinca, muscle	total analysis	-0.0002	430000
8	FORSMARK	AFM000050	23673	2	2005-12-02	Tinca tinca, muscle	total analysis	-0.0002	450000
8	FORSMARK	AFM000010	23674	3	2005-12-02	Tinca tinca, muscle	total analysis	-0.0002	430000
8	FORSMARK	AFM000010	23675	4	2005-12-02	Tinca tinca, muscle	total analysis	-0.0002	460000
8	FORSMARK	AFM000010	23676	4	2005-12-02	Tinca tinca, muscle	total analysis		460000
8	FORSMARK	AFM000010	23677	4	2005-12-02	Tinca tinca, muscle	total analysis		460000
8	FORSMARK	AFM000010	23678	2	2005-12-02	Tinca tinca, muscle	total analysis	-0.0002	450000
99	SIMPEVARP	ASM000192	23822	298	2006-09-26	Perca fluviatilis, muscle	total analysis	0.0023	415400
99	SIMPEVARP	ASM000192	23826	305	2006-09-26	Perca fluviatilis, muscle	total analysis	0.0028	428700
99	SIMPEVARP	ASM000192	23832	312	2006-09-26	Perca fluviatilis, muscle	total analysis	0.0021	422900
100	SIMPEVARP	ASM000192	23834	324	2006-12-08	Perca fluviatilis, piscivorous, muscle	dissolution of the ashed samples		425000
101	SIMPEVARP	ASM000192	23836	323	2006-12-08	Perca fluviatilis, planktivorous, muscle	dissolution of the ashed samples in HNO3/HCl/HF aliquots		
102	SIMPEVARP	ASM000192	23897	297	2006-09-26	Rutilus rutilus, muscle	total analysis	0.0012	420700
102	SIMPEVARP	ASM000192	23899	304	2006-09-26	Rutilus rutilus, muscle	total analysis	0.0012	430200
102	SIMPEVARP	ASM000192	23901	311	2006-09-26	Rutilus rutilus, muscle	total analysis	0.001	425600

In case of CR all samples per ecosystem and site (Forsmark or Laxemar) are combined in all possible combinations (permutations). The rationale for this approach is that exact spatial matching is not possible for most samples as they were sampled in different campaigns and represent different locations and true sample pairs were therefore usually not available (cf. Sections 4.2.1 and 9.1.2 in Tröjbom et al. 2013).

The tables "ValueList_from" and "Valuelist_to" are generated based on the coupling information in the file "Coupling parameters to data_ver3.xlsx" (cf. Appendix A Section 2.2.2). Samples from the tables "ValueList_from" and "Valuelist_to" are combined into CR ratios in the query "CRws_match_ecosystem" where the statistics (BE, GM, GSD) are calculated based on these pairs (cf. Appendix A Section 2.2.6). Data are combined according to the query structure in Figure 2-2 and the corresponding SQL expression is shown below.

Figure 2-2. Screen shot showing the structure of the query "CRws_match_ecosystem" where data are combined into pairs and statistics are calculated. The corresponding SQL code is shown below the table.




```
SELECT ParamTab.ParameterType, ValueList_from_pos.FROM, ValueList_from_pos.Element, ValueList_from_pos.SITE,
Count([valueList_to_pos].[value]/[valuelist_from_pos].[value]) AS N, Min([valueList_to_pos].[value]/[valuelist_from_pos].[value]) AS
MinV, Exp(Avg(Log([valueList_to_pos].[value]/[valuelist_from_pos].[value]))) AS GM,
Max([valueList_to_pos].[value]/[valuelist_from_pos].[value]) AS MaxV,
Exp(StDev(Log([valueList_to_pos].[value]/[valuelist_from_pos].[value]))) AS GSD, ParamTab.Unit,
Min(([valueList_to_pos].[value]/[valuelist_to_pos].[tot_c]*1000000)/[valuelist_from_pos].[value]) AS MinC,
Exp(Avg(Log(([valueList_to_pos].[value]/[valuelist_to_pos].[tot_c]*1000000)/[valuelist_from_pos].[value]))) AS GMc,
Max(([valueList_to_pos].[value]/[valuelist_to_pos].[tot_c]*1000000)/[valuelist_from_pos].[value]) AS MaxC,
Exp(StDev(Log(([valueList_to_pos].[value]/[valuelist_to_pos].[tot_c]*1000000)/[valuelist_from_pos].[value]))) AS GSDc,
ParamTab.UnitC, ParamTab.ParameterDB
FROM ParamTab INNER JOIN (ValueList_from_pos INNER JOIN ValueList_to_pos ON (ValueList_from_pos.Ecosystem_type =
ValueList_to_pos.Ecosystem_type) AND (ValueList_from_pos.Element = ValueList_to_pos.Element) AND (ValueList_from_pos.FROM =
ValueList_to_pos.TO) AND (ValueList_from_pos.SITE = ValueList_to_pos.SITE)) ON ParamTab.Parameter = ValueList_to_pos.TO
GROUP BY ParamTab.ParameterType, ValueList_from_pos.FROM, ValueList_from_pos.Element, ValueList_from_pos.SITE,
ParamTab.Unit, ParamTab.UnitC, ParamTab.ParameterDB
HAVING (((ParamTab.ParameterType)="CRs" Or (ParamTab.ParameterType)="CRw"));
```

The individual data pairs formed are combined are listed in Table 2-8[6]. The three water samples from Forsmark (23118, 23120, 23117) were combined with three fish samples from the same site (23602, 23606, 23671). For Laxemar/Simpevarp, three water samples were combined with six fish samples. This means that the statistics of the CR was calculated based on the 9 pairs from Forsmark, and for both sites in total for 27 data pairs (3*3+3*6 permutations). The estimates calculated from these values are identical to the parameter values for La in the "Export_KdCR_database" table (0.006, 0.0002 and 2.8 m3/kgdw, respectively). The number of samples for this parameter is defined as lowest number of samples in the numerator of denominator, in this case 3.

The consequence of omitting values below reporting limits from Forsmark (59% of the fish samples, cf.

[6] From query "CRws_match_ecosystem_persample" in the SKB_chemistry_SR_PSU database.

Table 2-7) probably slightly overestimates the best estimate for cR_Lake_fish for La, which is utilised as an element analogue for Np as described in Section 2.2.

Table 2-8. The individual sample pairs and resulting CR values for cR_Lake_fish / La. From the query "CRws_match_ecosystem_persample" in the SKB_chemistry_SR_PSU database. The columns "ValueList…" show the individual samples combined (SKB sample numbers uniquely coloured). CR_dw and CR_C the ratios per dry weight and carbon, respectively.

ValueList_from_pos.SKB_SAMPLE_NO	ValueList_to_pos.SKB_SAMPLE_NO	FROM	Element	SITE	CR_dw	Unit	CR_C	UnitC
23118	23602	cR_Lake_Fish	La	FORSMARK	0.0100	m3/kgdw	0.0233	m3/kgC
23118	23606	cR_Lake_Fish	La	FORSMARK	0.0040	m3/kgdw	0.0095	m3/kgC
23118	23671	cR_Lake_Fish	La	FORSMARK	0.0020	m3/kgdw	0.0047	m3/kgC
23120	23602	cR_Lake_Fish	La	FORSMARK	0.0135	m3/kgdw	0.0314	m3/kgC
23120	23671	cR_Lake_Fish	La	FORSMARK	0.0027	m3/kgdw	0.0063	m3/kgC
23120	23606	cR_Lake_Fish	La	FORSMARK	0.0054	m3/kgdw	0.0129	m3/kgC
23117	23606	cR_Lake_Fish	La	FORSMARK	0.0063	m3/kgdw	0.0149	m3/kgC
23117	23602	cR_Lake_Fish	La	FORSMARK	0.0156	m3/kgdw	0.0363	m3/kgC
23117	23671	cR_Lake_Fish	La	FORSMARK	0.0031	m3/kgdw	0.0073	m3/kgC
23262	23826	cR_Lake_Fish	La	SIMPEVARP	0.0012	m3/kgdw	0.0028	m3/kgC
23262	23822	cR_Lake_Fish	La	SIMPEVARP	0.0010	m3/kgdw	0.0024	m3/kgC
23262	23897	cR_Lake_Fish	La	SIMPEVARP	0.0005	m3/kgdw	0.0012	m3/kgC
23262	23901	cR_Lake_Fish	La	SIMPEVARP	0.0004	m3/kgdw	0.0010	m3/kgC
23262	23899	cR_Lake_Fish	La	SIMPEVARP	0.0005	m3/kgdw	0.0012	m3/kgC
23262	23832	cR_Lake_Fish	La	SIMPEVARP	0.0009	m3/kgdw	0.0022	m3/kgC
23263	23822	cR_Lake_Fish	La	SIMPEVARP	0.0044	m3/kgdw	0.0106	m3/kgC
23263	23826	cR_Lake_Fish	La	SIMPEVARP	0.0054	m3/kgdw	0.0126	m3/kgC
23263	23832	cR_Lake_Fish	La	SIMPEVARP	0.0040	m3/kgdw	0.0095	m3/kgC
23263	23897	cR_Lake_Fish	La	SIMPEVARP	0.0023	m3/kgdw	0.0055	m3/kgC
23263	23899	cR_Lake_Fish	La	SIMPEVARP	0.0023	m3/kgdw	0.0054	m3/kgC
23263	23901	cR_Lake_Fish	La	SIMPEVARP	0.0019	m3/kgdw	0.0045	m3/kgC
23265	23822	cR_Lake_Fish	La	SIMPEVARP	0.0013	m3/kgdw	0.0031	m3/kgC
23265	23832	cR_Lake_Fish	La	SIMPEVARP	0.0012	m3/kgdw	0.0028	m3/kgC
23265	23897	cR_Lake_Fish	La	SIMPEVARP	0.0007	m3/kgdw	0.0016	m3/kgC
23265	23899	cR_Lake_Fish	La	SIMPEVARP	0.0007	m3/kgdw	0.0015	m3/kgC
23265	23901	cR_Lake_Fish	La	SIMPEVARP	0.0006	m3/kgdw	0.0013	m3/kgC
23265	23826	cR_Lake_Fish	La	SIMPEVARP	0.0016	m3/kgdw	0.0036	m3/kgC

2.1.3 Example cR_agri_veg for Cs

No site data are available for the parameterisation of cR_agri_veg for Cs, leaving this section empty, for description of available data for CR for Cs in vegetables see Section 3.1.3.

2.2 Compilation of literature data

Literature data sources included in the K_d/CR compilation are described in Section 3.2 in Tröjbom et al. (2013). The data were initially compiled in Excel-files which were imported to the database. The data reported in IAEA (2010) are only available in a published report in PDF format. This means that tables in the PDF file were converted into Excel-tables. After this conversion a manual check was done to ensure that data were correctly converted. The data from the ERICA database (Beresford et al. 2008, Hosseini et al. 2008) was exported from the ERICA tool directly into an Excel file. The Wildlife Transfer Parameter Database (ICRP 2011) data were downloaded from the site as an Excel file.

2.3 Conversion of units and statistical measures

Unit conversions of literature data are necessary prior to the parameterisation due to the fact that K_d and CR values can be reported in different units (as described in Section 2.4). K_d values can be reported in units of cubic meters or litres (m^3/kgdw or L/kgdw). CR values can be reported per dry weight, fresh weigh or normalised to carbon, resulting in CR units of kg/kg_{dw}, kg/kg_{fw} or kg/kg_C for terrestrial biota and m^3/kg_{dw}, m^3/kg_{fw} or m^3/kg_C for aquatic biota.

In the K_d/CR compilation data are stored in original units. Data are then converted into common units by the use of conversion factors derived from data in IAEA (2010) as described in Section 4.3.2 in Tröjbom et al. (2013). This conversion is done in the second Accessdatabase, SKB_K_d_CR.accdb using the table “ConversionFactors” in the (cf. Appendix A Section 3.2.1), which is generated from the IAEA-data. This table contains factors for all data categories and units.

Most data in the database are reported as geometrical mean and geometrical standard deviations. In some cases the data are reported as arithmetic means and standard deviation. In these cases the statistical measure are converted into geometrical measures as described in 4.3.2 in Tröjbom et al. (2013) and in 3.2.2 in Appendix A.

Both unit conversions and conversions of statistical measures are performed in the database query “K_dCR_convPSU” (Appendix A Section 3.2.2) prior to the selection process and generation of FinalTable (Appendix A Section 3.2.4).

The NHB-parameters (Non-Human-Biota) are delivered in a unit related to fresh weight of biota as described in Section 2.4 in Tröjbom et al. (2013) (Equation 2-5 and 2-6). These parameters are estimated per carbon or per dry weight during the data handling, and converted into fresh weight in a post-processing stage (cf. Appendix A Section 3.3.7). Site data are related to the edible parts of the animals and since the NHB-parameters applies to whole body conversion factors from Yankovich et al. (2010) are applied when possible. This is performed in a post-processing step in the “SKB_Chemistry _SR_PSU.accdb” database (cf. Appendix A Section 2.1.3 and 2.3.1).

2.4 Defining plausible parameter variation

The method for defining plausible parameter variation is described in Section 4.3 in Tröjbom et al. (2013). The GSD limits defined in Table 4-4 in Tröjbom et al. (2013) is stored in the table ParamTab (cf. Appendix A Section 3.1.3), and these GSD values are used to replace the original values according to the criteria defined in Section 4.4.2 in Tröjbom et al. (2013). The scripts for these replacements are found in the Visual Basic (VBA) module “SelectFunctions”, which is found in Appendix A Section 3.5.1 in the function “sel_gsd”.

3 Selection of data

The selection of data for each parameter/element case is done in three steps described in detail in Section 4.4 to 4.6 in Tröjbom et al. (2013). First an initial data selection is done by first matching available representative data from the K_d/CR compilation to the parameters (Section 4.4.1 in Tröjbom et al. 2013). Then, from this representative data an initial automatic data selection is done based on fixed criteria (Section 4.4.2 in Tröjbom et al. 2013). After the initial data selection a comparison among available data sources is done (Section 4.5 in Tröjbom et al. 2013). In the last step, the initial data selection and the results from the comparison of data sources are evaluated and a manual data selection is done if needed (Section 4.6 in Tröjbom et al. 2013).
In this chapter the implementation of these steps in the Access database is described in detail. Technical details of the database structure and database objects are further described in Appendix A.

3.1 Matching of data and ranking of literature sources

In this step, data from the K_d/CR compilation are assigned to each parameter in an excel file named ParamClass.xls, according to the criteria described in Section 4.4.1 in Tröjbom et al. (2013). The Excel-file ParamClass.xls list all available data categories (both site data and literature data) in the K_d/SR compilation. From this file the data categories that are assessed to be representative for each parameter are manually selected. The selection of representative data is described for each parameter in Chapters 5 to 8 in Tröjbom et al. (2013). The selected data categories are marked by writing the parameter name in the column M at the row of the selected data category as shown in the examples below (Figure 3-1). At this step representative data are also ranked according to the general rules given in Section 4.4.1 in Tröjbom et al. (2013) by denoting L1, L2, L3 and L4 for the literature sources. If data from the previous assessment SR-Site (Nordén et al. 2010) are available, these data are marked as L5. Site data from Forsmark or Laxemar are marked as FM or LX respectively. Site data derived from a combination of Forsmark and Laxemar data are marked as F&L. At this step, selections of data sources are made on parameter level and not for each element/parameter case. For some elements this might not result in the most optimal data selection (for example if a literature source defined as L3 has more samples for a specific element than the literature source defined as L1). This is re-evaluated in the final data selection process when each parameter/element is evaluated (see Section 4.3 in Tröjbom et al. 2013). Deviations from this general assumption are if needed handled during the manual step described in Section 3.4.
When the data selection is completed the file ParamClass.xls is imported into the Access database (table ParamClass) and the information is used to couple the data in the selected categories to all parameter and element cases in the database. This is described in Appendix A Section 3.1.2.

3.1.1 Example K_d_regoLow for Ni

In Figure 3-1 it is illustrated how the data selection for the parameter K_d_regoLow has been done. The categories that could be assumed to be representative for K_d_regoLow are selected. The selected data categories are marked by writing the parameter name in column M. The site-specific data are marked as LX for Laxemar data (PSUprel_LX), FM for Forsmark data (PSUprel_FM) or F&L for site data from both Forsmark and Laxemar (PSUprel_F&L). The available literature sources are ranked according to relevance, in this case IAEA (2010) data for several Media types are available, the Media type “Mineral” is assumed to best represent the till soil and therefore marked as L1, the Media type “Sand” is marked as L2, the “Clay” as L3. The data from Nordén et al. (2010) represent data used in SR-Site, this category is always selected as L5, which makes it possible to compare this data to other data sources.

In the initial matching of data the IAEA (2010) for "Clay" was marked as L2, and "sand" was marked as L3. In a later step of the parameterisation process it was noted that the data for "clay" was limited (few samples) and therefore L2 was assigned to the category "Sand" instead and L3 was assigned to data category "All soils" instead. This is also noted in the comment field to the right.

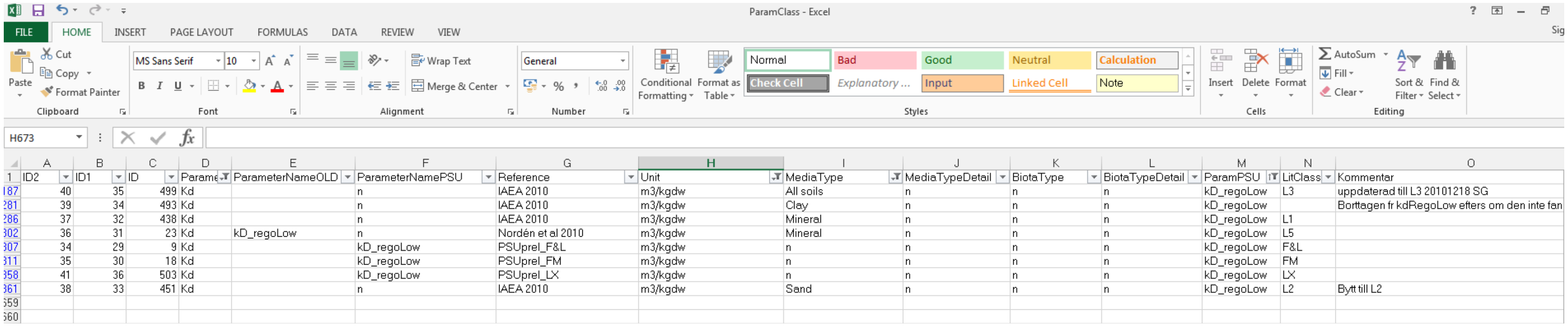
3.1.2 Example cR_lake_Fish for Np

The matching of data categories representative for the limnic fish is illustrated in Figure 3-2. The ranking of the data categories is done in column "N", IAEA (2010) data for freshwater fish, mussel tissues are selected as L1, IAEA(2010) data for whole body as L2 and ERICA data for benthic fished are selected as L3 data. As can be seen in this example, FM, LX and F&L are assigned to two rows, which is due to the fact that the site-specific data are available in two units (m^3/kgdw and m^3/kgC).

3.1.3 Example cR_agri_veg for Cs

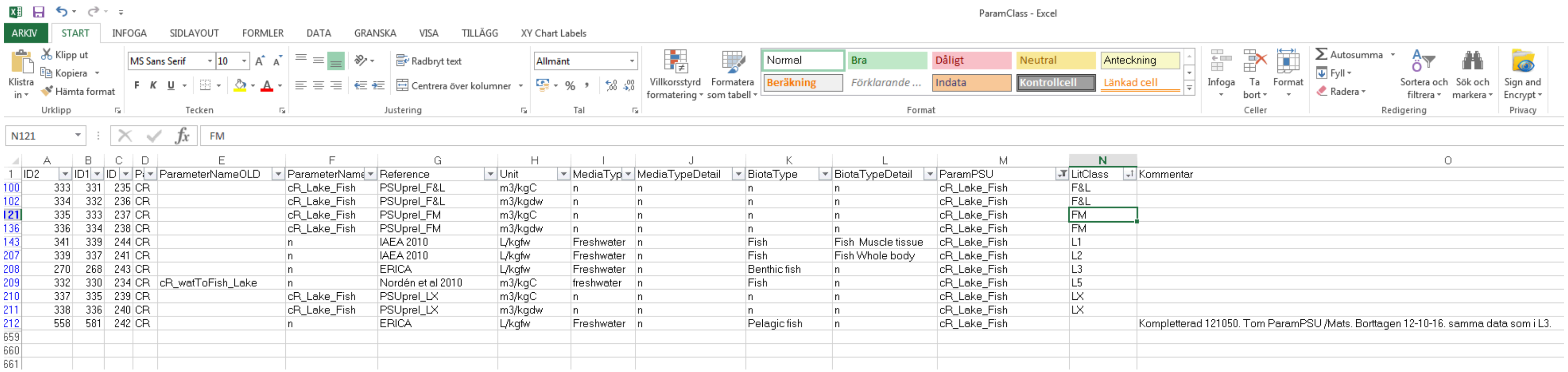
The data matching for vegetables (cR_agri_veg) is illustrated in Figure 3-3. In the case of vegetables no site-specific data are available. The filtering is done in column "K" "Biota Type" where all vegetables are selected. The IAEA (2010) data available represent several vegetable types and also different parts of the vegetables (stems, fruits, leaves) they are also divided into several Media types (all soils, mineral, organic). In this case the Media type "all soils" are selected and Leafy and non leafy vegetables are selected by marking L1, L2 and L3 in these rows (see table for details).

Figure 3-1. Data categories are assigned to each parameter using the excelfile ParamClass, in this example the data matching for K_d_regoLow parameter is illustrated. The selected data categories are also ranked (in column 'N') according to relevance according to the general rules in Section 4.4.1 in Tröjbom et al. (2013).



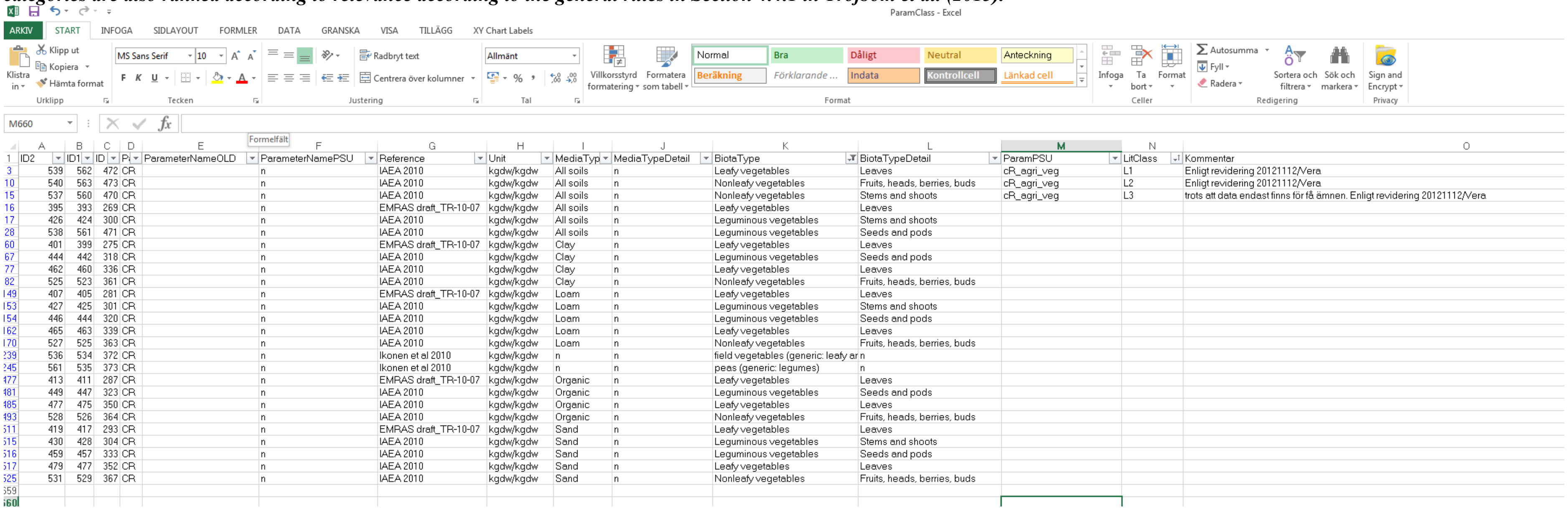
	ID2	ID1	ID	Parame	ParameterNameOLD	ParameterNamePSU	Reference	Unit	MediaType	MediaTypeDetail	BiotaType	BiotaTypeDetail	ParamPSU	LitClass	Kommentar
187	40	35	499	Kd		n	IAEA 2010	m3/kgdw	All soils	n	n	n	kD_regoLow	L3	uppdaterad till L3 20101218 SG
281	39	34	493	Kd		n	IAEA 2010	m3/kgdw	Clay	n	n	n	kD_regoLow		Borttagen fr kdRegoLow efters om den inte fan
286	37	32	438	Kd		n	IAEA 2010	m3/kgdw	Mineral	n	n	n	kD_regoLow	L1	
302	36	31	23	Kd	kD_regoLow	n	Nordén et al 2010	m3/kgdw	Mineral	n	n	n	kD_regoLow	L5	
307	34	29	9	Kd		kD_regoLow	PSUprel_F&L	m3/kgdw	n	n	n	n	kD_regoLow	F&L	
311	35	30	18	Kd		kD_regoLow	PSUprel_FM	m3/kgdw	n	n	n	n	kD_regoLow	FM	
358	41	36	503	Kd		kD_regoLow	PSUprel_LX	m3/kgdw	n	n	n	n	kD_regoLow	LX	
361	38	33	451	Kd		n	IAEA 2010	m3/kgdw	Sand	n	n	n	kD_regoLow	L2	Bytt till L2

Figure 3-2. Data categories are assigned to each parameter using the excelfile ParamClass, in this example the data matching for cR_lake_fish parameter is illustrated The selected data categories are also ranked (in column 'N') according to relevance according to the general rules in Section 4.4.1 in Tröjbom et al. (2013).



	ID2	ID1	ID	Pa	ParameterNameOLD	ParameterName	Reference	Unit	MediaTyp	MediaTypeDetail	BiotaType	BiotaTypeDetail	ParamPSU	LitClass	Kommentar
100	333	331	235	CR		cR_Lake_Fish	PSUprel_F&L	m3/kgC	n	n	n	n	cR_Lake_Fish	F&L	
102	334	332	236	CR		cR_Lake_Fish	PSUprel_F&L	m3/kgdw	n	n	n	n	cR_Lake_Fish	F&L	
121	335	333	237	CR		cR_Lake_Fish	PSUprel_FM	m3/kgC	n	n	n	n	cR_Lake_Fish	FM	
136	336	334	238	CR		cR_Lake_Fish	PSUprel_FM	m3/kgdw	n	n	n	n	cR_Lake_Fish	FM	
143	341	339	244	CR		n	IAEA 2010	L/kgfw	Freshwater	n	Fish	Fish Muscle tissue	cR_Lake_Fish	L1	
207	339	337	241	CR		n	IAEA 2010	L/kgfw	Freshwater	n	Fish	Fish Whole body	cR_Lake_Fish	L2	
208	270	268	243	CR		n	ERICA	L/kgfw	Freshwater	n	Benthic fish	n	cR_Lake_Fish	L3	
209	332	330	234	CR	cR_watToFish_Lake	n	Nordén et al 2010	m3/kgC	freshwater	n	Fish	n	cR_Lake_Fish	L5	
210	337	335	239	CR		cR_Lake_Fish	PSUprel_LX	m3/kgC	n	n	n	n	cR_Lake_Fish	LX	
211	338	336	240	CR		cR_Lake_Fish	PSUprel_LX	m3/kgdw	n	n	n	n	cR_Lake_Fish	LX	
212	558	581	242	CR		n	ERICA	L/kgfw	Freshwater	n	Pelagic fish	n	cR_Lake_Fish		Kompletterad 121050. Tom ParamPSU /Mats. Borttagen 12-10-16. samma data som i L3.

Figure 3-3. Data categories are assigned to each parameter using the excelfile ParamClass, in this example the data matching for cR_agri_veg parameter is illustrated The selected data categories are also ranked according to relevance according to the general rules in Section 4.4.1 in Tröjbom et al. (2013).



1	ID2	ID1	ID	Pa	ParameterNameOLD	ParameterNamePSU	Reference	Unit	MediaTyp	MediaTypeDetail	BiotaType	BiotaTypeDetail	ParamPSU	LitClass	Kommentar
3	539	562	472	CR		n	IAEA 2010	kgdw/kgdw	All soils	n	Leafy vegetables	Leaves	cR_agri_veg	L1	Enligt revidering 20121112/Vera
10	540	563	473	CR		n	IAEA 2010	kgdw/kgdw	All soils	n	Nonleafy vegetables	Fruits, heads, berries, buds	cR_agri_veg	L2	Enligt revidering 20121112/Vera
15	537	560	470	CR		n	IAEA 2010	kgdw/kgdw	All soils	n	Nonleafy vegetables	Stems and shoots	cR_agri_veg	L3	trots att data endast finns för få ämnen. Enligt revidering 20121112/Vera
16	395	393	269	CR		n	EMRAS draft_TR-10-07	kgdw/kgdw	All soils	n	Leafy vegetables	Leaves			
17	426	424	300	CR		n	IAEA 2010	kgdw/kgdw	All soils	n	Leguminous vegetables	Stems and shoots			
28	538	561	471	CR		n	IAEA 2010	kgdw/kgdw	All soils	n	Leguminous vegetables	Seeds and pods			
60	401	399	275	CR		n	EMRAS draft_TR-10-07	kgdw/kgdw	Clay	n	Leafy vegetables	Leaves			
67	444	442	318	CR		n	IAEA 2010	kgdw/kgdw	Clay	n	Leguminous vegetables	Seeds and pods			
77	462	460	336	CR		n	IAEA 2010	kgdw/kgdw	Clay	n	Leafy vegetables	Leaves			
82	525	523	361	CR		n	IAEA 2010	kgdw/kgdw	Clay	n	Nonleafy vegetables	Fruits, heads, berries, buds			
149	407	405	281	CR		n	EMRAS draft_TR-10-07	kgdw/kgdw	Loam	n	Leafy vegetables	Leaves			
153	427	425	301	CR		n	IAEA 2010	kgdw/kgdw	Loam	n	Leguminous vegetables	Stems and shoots			
154	446	444	320	CR		n	IAEA 2010	kgdw/kgdw	Loam	n	Leguminous vegetables	Seeds and pods			
162	465	463	339	CR		n	IAEA 2010	kgdw/kgdw	Loam	n	Leafy vegetables	Leaves			
170	527	525	363	CR		n	IAEA 2010	kgdw/kgdw	Loam	n	Nonleafy vegetables	Fruits, heads, berries, buds			
239	536	534	372	CR		n	Ikonen et al 2010	kgdw/kgdw	n	n	field vegetables (generic: leafy ar	n			
245	561	535	373	CR		n	Ikonen et al 2010	kgdw/kgdw	n	n	peas (generic: legumes)	n			
477	413	411	287	CR		n	EMRAS draft_TR-10-07	kgdw/kgdw	Organic	n	Leafy vegetables	Leaves			
481	449	447	323	CR		n	IAEA 2010	kgdw/kgdw	Organic	n	Leguminous vegetables	Seeds and pods			
485	477	475	350	CR		n	IAEA 2010	kgdw/kgdw	Organic	n	Leafy vegetables	Leaves			
493	528	526	364	CR		n	IAEA 2010	kgdw/kgdw	Organic	n	Nonleafy vegetables	Fruits, heads, berries, buds			
511	419	417	293	CR		n	EMRAS draft_TR-10-07	kgdw/kgdw	Sand	n	Leafy vegetables	Leaves			
515	430	428	304	CR		n	IAEA 2010	kgdw/kgdw	Sand	n	Leguminous vegetables	Stems and shoots			
516	459	457	333	CR		n	IAEA 2010	kgdw/kgdw	Sand	n	Leguminous vegetables	Seeds and pods			
517	479	477	352	CR		n	IAEA 2010	kgdw/kgdw	Sand	n	Leafy vegetables	Leaves			
525	531	529	367	CR		n	IAEA 2010	kgdw/kgdw	Sand	n	Nonleafy vegetables	Fruits, heads, berries, buds			
559															
560															

3.2 Initial data selection criteria

The table created after the matching of data consists of representative data sources connected to each parameter and element case (table FinalTable, cf. Appendix A Section 3.2.4). In some cases both site data (Forsmark, Laxemar, Forsmark and Laxemar data) and literature data (L1, L2, L3) are available while other parameter/element cases have no or limited data available for selection. From this table an automatic data selection is done based on basic principles and rules defined in Section 4.4.2 and illustrated in Figure 4-4 in Tröjbom et al. (2013). These rules are based on relevance of data, sample number and GSD interval. This process is done using the script presented in 3.5.1 in the Appendix A.

3.2.1 Example K_d_regoLow for Ni

K_d data for Ni in till (K_d_regoLow) are available from Forsmark and IAEA (2010) all soils (L3). Data are lacking from Laxemar and the selected data sources L1 (IAEA 2010 mineral soil) and L2 (IAEA 2010 sand). In the initial data selection the Forsmark data are assigned according to the principles described in Section 4.4.2 in Tröjbom et al. (2013). In the next step the statistical measures (minimum, maximum and GSD) for the selected Forsmark data are evaluated. This process depends on the number of samples of the selected Forsmark data, as described in Section 4.4.2 in Tröjbom et al. (2013). The number of samples for the Forsmark data is 7 which means that the data selection process follows the data selection process for cases with sample numbers between 3 and 10 (as described in Section 4.2.2 of Tröjbom et al. (2013). This means that the reported GSD is compared to the GSDmean and GSDmax values. In this case the reported GSD for the Forsmark data was 1.4 which is lower than the GSDmean (defined as 3 for the parameter K_d_regoLow) and therefore the GSDmean of 3 was assigned as the selected GSD. The minimum and maximum reported for the Forsmark data are compared to the 5^{th} and 95^{th} percentile calculated using the reported GM and the assigned GSDmean. In this case the 5^{th} and 95^{th} percentiles were selected as minimum and maximum values since these were lower/higher than the reported min/max. This is illustrated in Figure 3-4. This selection process is done using the script presented in 3.5.1 in the Appendix A.

3.2.2 Example cR_lake_Fish for Np

The initial selection of data for Np for limnic fish are presented in Figure 3-5. For Np in limnic fish data are scarce, only one sample from the ERICA database is reported, and no GSD, minimum or maximum were reported by ERICA. This data is categorised as L3, but since no other data are available this data is selected in the initial data selection process. Since the sample number is low (N=1) GSDmax is assigned and min and max are estimated based on GSDmax in order to achieve a range around the single value. The minimum value is assigned as the smallest of the reported minimum value and the 5th percentile calculated using the GSDmax, in this case the calculated 5th percentile since no minimum value was reported. The maximum value was assigned the largest value of the reported maximum value or the 95th percentile calculated using the GSDmax value, in this case the 95th percentile was assigned since no maximum value was reported. This is illustrated in Figure 3-5. This selection process is done using the script presented in 3.5.1 in the Appendix A.

3.2.3 Excample cR_agri_veg for Cs

For uptake of Cs in vegetables (cR_agri_veg) no site-specific data are available. Literature data from the assigned L1 (IAEA 2010, Non leafy vegetables) and L2 (IAEA 2010 Leafy vegetables) sources are available. According to the initial data selection criteria the highest ranked literature data source (here L1) is assigned since no site-specific data are available. The number of samples are large in this case, N=290. The reported GSD is 6 which is higher than the GSDmean for this parameter (GSDmean=4), this means that the reported GSD is selected. The lowest of the reported minimum value and the calculated 5th percentile is selected as minimum, in this case the reported minimum was the lowest and therefore selected. The highest of the reported maximum and the calculated 95th percentile are assigned, in this case the calculated 95th percentile (note that the percentiles are calculated using the selected GSD, in this case the reported GSD (6)). This is illustrated inFigure 3-6. This selection process is done using the script presented in 3.5.1 in the Appendix A.

Figure 3-4. Initial data selection for K_d_regoLow for Ni. Site data for Forsmark are available and selected according to the initial data selection principles.

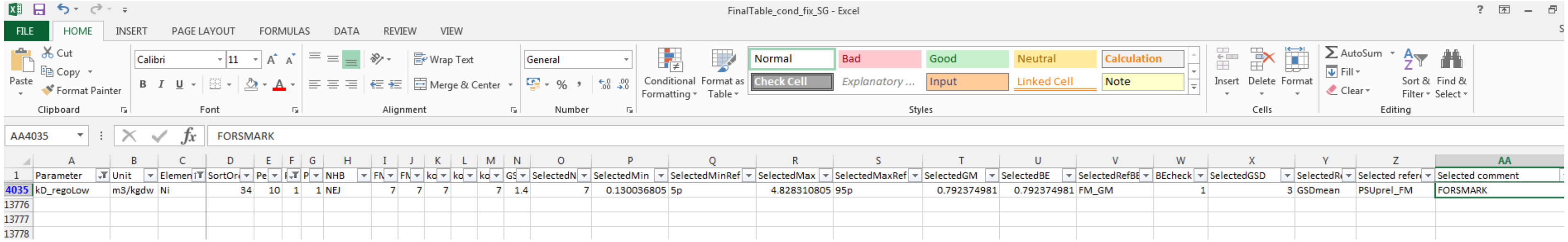


Figure 3-5. Initial data selection for cR_Lake_fish for Np. Only one literature source were available (L3=ERICA data), the sample number was only 1. These data are according to the initial data selection principles the best available data and therefore assigned.

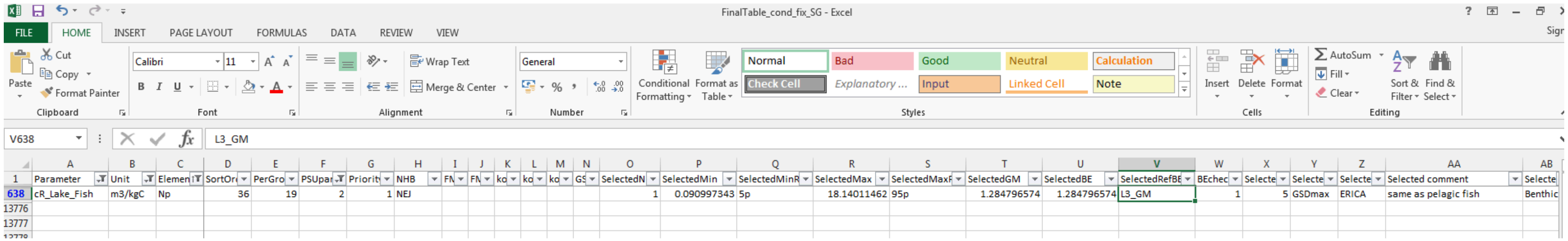
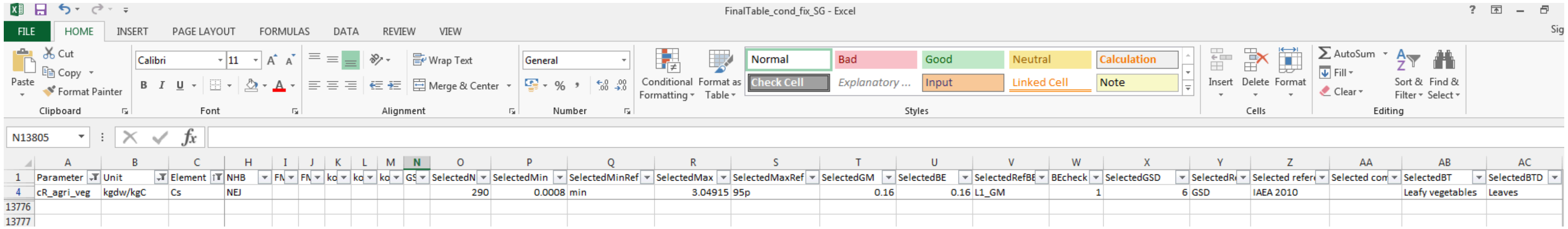


Figure 3-6. Initial data selection for cR_agri_veg for Cs. Literature data for two categories from IAEA (2010) are available the category ranked as L1 was selected.



3.3 Comparison of data sources

The data from different sources are compared using automated routines, referred to as "sense checks". These checks are described in Section 4.5 in Tröjbom et al. (2013). The code is presented in Section 3.5.1 in Appendix A (under the heading "'returnerar check av överlapp inom litteraturdata"). The sense checks are based on simple checks of data range overlap. If the ranges of the data overlap (fully or partially) the data are assumed to be reasonable. If two data sources do not overlap at all, or just overlap to a small extent, the data are indicated to be uncertain. The results of the Sense checks are returned in the database table FinalTable_cond (cf. Appendix A Section 3.2.9). The results are presented in the tables as abbreviations, which are described in Section 4.5 in Tröjbom et al. (2013). The results from these sense checks are used in the nest step of the parameterisation process, the manual evaluation and final data selection.

3.3.1 Example K_d_regoLow for Ni

In this section the results from the different sense checks for Ni K_d_regoLow are described. In this case the only sense checks that is feasible is the check of the Forsmark data compared to the L3 data source (IAEA 2010 all soils). Even though the result from other "sense checks" is not applicable all sense checks are performed automatically and the results are presented (cf. Figure 3-7).

- The first check that is done is the comparison of the selected data against the literature data range, the result is presented in the column AF, "Validation". The result is presented as S1 which means that the literature data fully encompasses the selected data (see Figure 4-5 in Tröjbom et al. 2013).
- The second check is presented in column AG, "Validation FM", in this check the Forsmark data are compared to the literature data, in this cases these two checks are identical (since the selected data are Forsmark data).
- The third check in presented in column AH, "ValidationLX", in this sense check Laxemar data are compared to literature data range. In this case this sense check is not applicable since no Laxemar data are available.
- The fourth sense check is when the combined Forsmark and Laxemar data are compared to the literature range in column AI, "ValidationF_L", in this sense check Forsmark and Laxemar data are compared to literature data range. This sense check is not applicable since no Laxemar data are presented, but since Forsmark data are available the sense check is preformed and the result is presented. In this case this check is identical to the first and second check.
- The fifth sense check is for literature data. In this sense check the overlap between different literature sources are controlled in order to see if the data found in literature are consistent. In this case only one literature data source is available and this check is not relevant. The result is presented as 1:0/0, meaning that there is 1 source available and that there is 0 out of 0 overlaps (as described in Section 4.5.4 in Tröjbom et al. 2013).

The only relevant sense check in this case with two available data sources is the check that shows that the range of Forsmark data is within the range of literature data.

3.3.2 Example cR_lake_Fish for Np

For the case Np in limnic fish, the only data available is one literature data sample from ERICA. Therefore none of the sense checks are relevant and all of the results columns are left empty (Figure 3-8). This indicates high uncertainty for this parameter, which is handled in the next step (cf. Section 3.4.2).

3.3.3 Example cR_agri_veg for Cs

The two literature data sources available for this parameter are compared and an overlap between the sources is reported in the column AJ as 2:1/1. This means that there are 2 sources available and 1 range overlap out of 1 possible overlap (Figure 3-9). No other sense check is relevant due to lack of data.

Figure 3-7 The results of the sense checks for the parameter K_d_regoLow for Ni. The results are marked as S1 in column AF, AG, AI (see text in 3.3.1 for further explanation). The sense check that is relevant in this case in the sense check of Forsmark data against the literature data. This sense check show that the Forsmark data are within the range of the literature data.

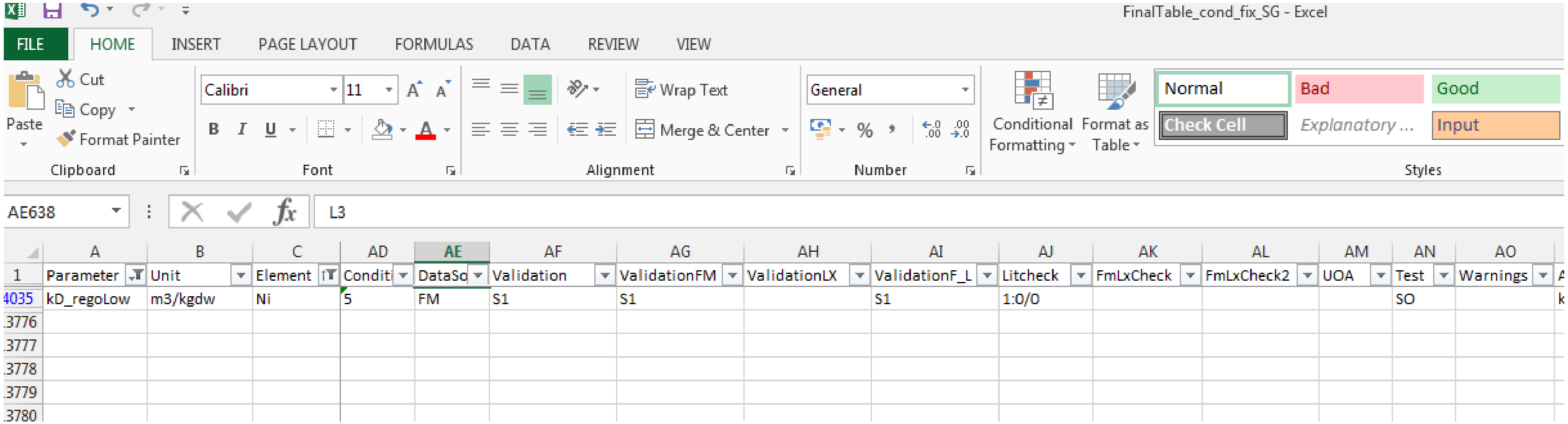


Figure 3-8. The results of the sense checks for the parameter cR_lake_fish for Np. There is only one data source available so the sense checks are not applicable for this case.

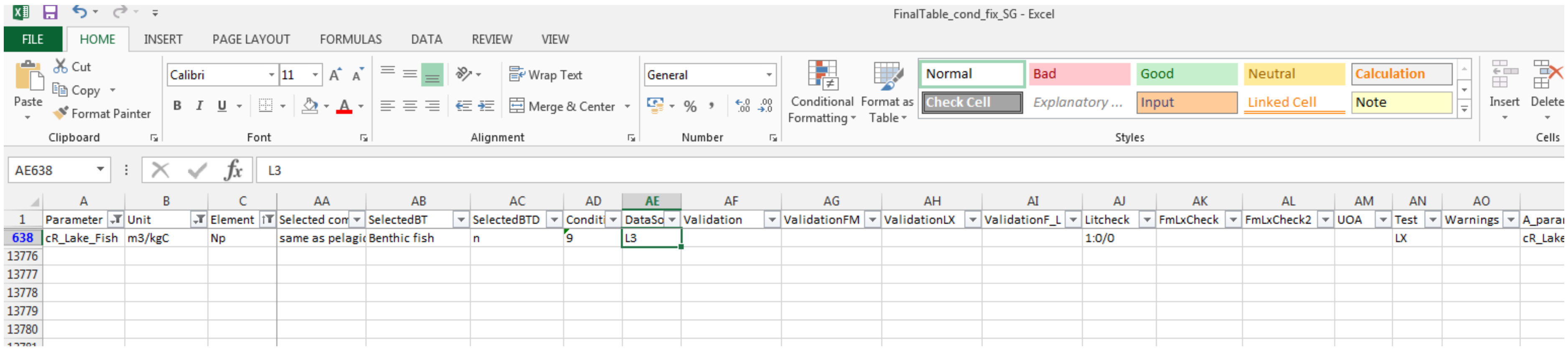
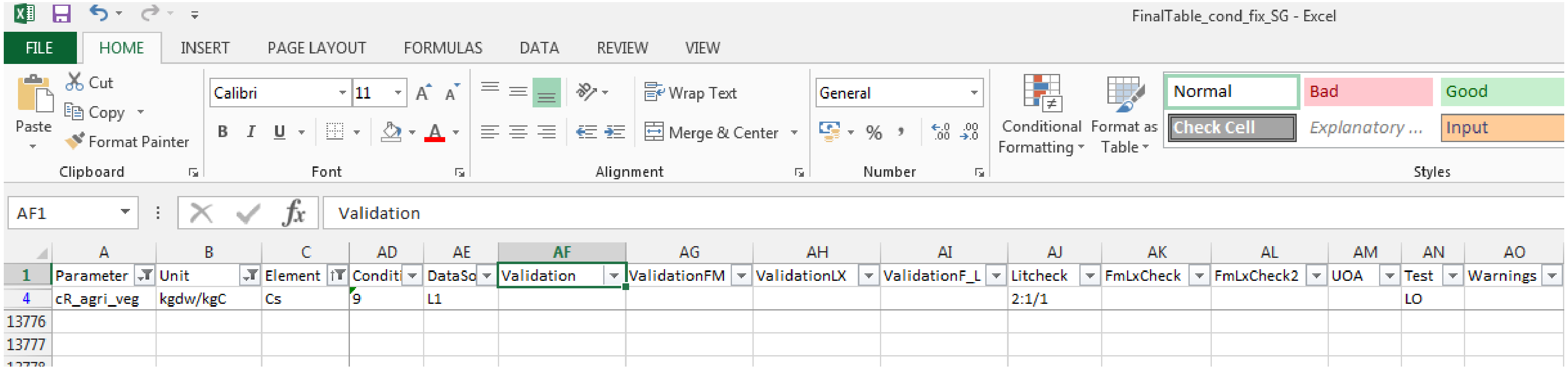


Figure 3-9. The results of the sense checks for the parameter cR_agri_veg for Cs. There is no site data available, the only sense check relevant here is the copmarision of literature data presented in "Litcheck" in column AJ which shows that there are two literature sources and that these overlap.



3.4 Final data evaluation and selection

After the initial automatic selection of data the manual data evaluation and final selection process is conducted as described in Section 4.6 in Tröjbom et al. (2013). This process is done in the interface excelfile "FinalTable_cond_fix", which is a fixed version of the FinalTable_cond table (cf. Appendix A Section 3.3.1). In this file a different data source could be selected by writing in column AP to AW. This information is later used to override the initial selection of data by importing the information back into the Access database to the table AnalogueTable (cf. Appendix A Section 3.2.6).
The final data evaluation is an iterative process, where the selected data will be re-evaluated and until the most representative data has been selected. This means that several iterations of exporting new versions of FinalTable_cond_fix, and re-import and update of AnalogueTable will be performed. All manual data selections and assumptions made when selecting a data source other than the automatic selection is documented in the FinalTable_cond_fix file to keep track on the revision history. Manual data selection could be done to override the initial automatic data selection if the sense checks show that the selected data are uncertain (large deviations from other sources) or where data are missing, scarce or uncertain. Element analogue or parameter analogue could be selected. If appropriate, a combination of parameter and element analogue could be manually selected. Analogues are selected based on the assumptions presented in Section 2.7 in Tröjbom et al. (2013).
After the manual data selection via the FinalTable_cond_fix – AnalogueTable loop, the comparisons of data sources (sense checks) are performed again with the new selected data. The results from these checks are re-evaluated and if the selected data source deviates from other available data, the data selection could be updated again in the next iteration.

To give an overview of the confidence in the selected data, a data source and quality categorisation based on the selected data is generated (cf. Table 9-13 in Tröjbom et al. 2013) and the results for all parameters and element combinations are presented in Table 9-14.
In order to assess the selected data further two types of plots were produced. 1) A plot for each parameter where selected data for all elements are plotted and sorted from the lowest to the highest value. These plots are included in the chapter of each parameter in Tröjbom et al. (2013). In this plot it can be assessed if the selected data are reasonable by comparing the data between elements. 2) The selected data for all parameters for each element are plotted. In this plot it can be evaluated in the selected data for the element are reasonable by comparing the data selected for other parameters. These plots, which were not included in the printed report, were used during the manual evaluation and are only available in the Excel file "FinalTable_cond_fix_PLOTB.xlsx" (for example Figure 3-11, Figure 3-13 and Figure 3-15).

3.4.1 Example K_d_regoLow for Ni

Forsmark data is selected for K_d_regoLow for Ni, which is the data source assumed the most representative for the parameter (according to the general assumption for parameterisation presented in Section 2.6.2 of Tröjbom et al. (2013), site-specific data should be prioritised if available). The results from the sense checks indicates that the selected data are within the range of available literature data and therefore reasonable (categorised as SO = Site data is used and overlap > 50% with literature data). No changes are made in the columns AP to AW in the FinalTable_cond_fix interface file, i.e. the automatic selection is retained. In this case a note is also given that the data for Ni presented for other soil types are not consistent and this needs to be further evaluated (cf. Figure 3-16).
In Figure 3-10 the finally selected data for K_d_regoLow are plotted. Selected data for Ni are found in lower half of the figure and next to elements that could be expected to be similar (divalent metals as Cd, Ra, Pd). In Figure 3-11 the selected data for Ni are plotted for all K_d parameters. It can be seen that K_d values for PM and aquatic sediments are higher than K_d values for other soil types. K_d value for till (K_d_regoLow) are in the same range as K_d for clay and peat soils.

The final data evaluation shows that the selected data for K_d_regoLow is coherent among available data sources and other parameters and reasonable with respect to similar elements (possible analogues).

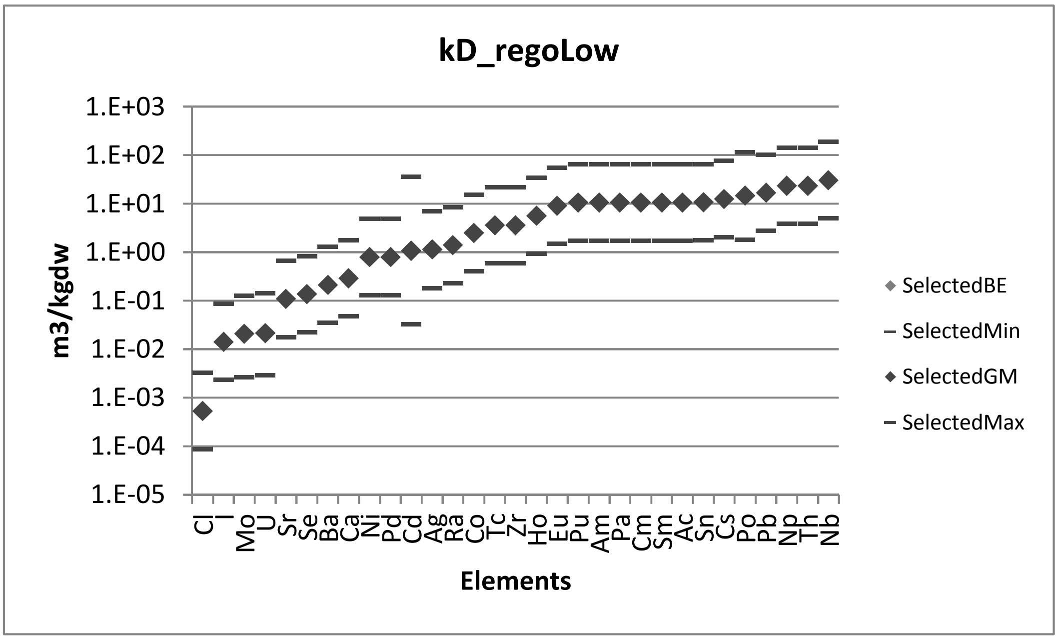


Figure 3-10. Plotted selected data for the parameter K_d_regoLow, sorted from the lowest value to the highest.

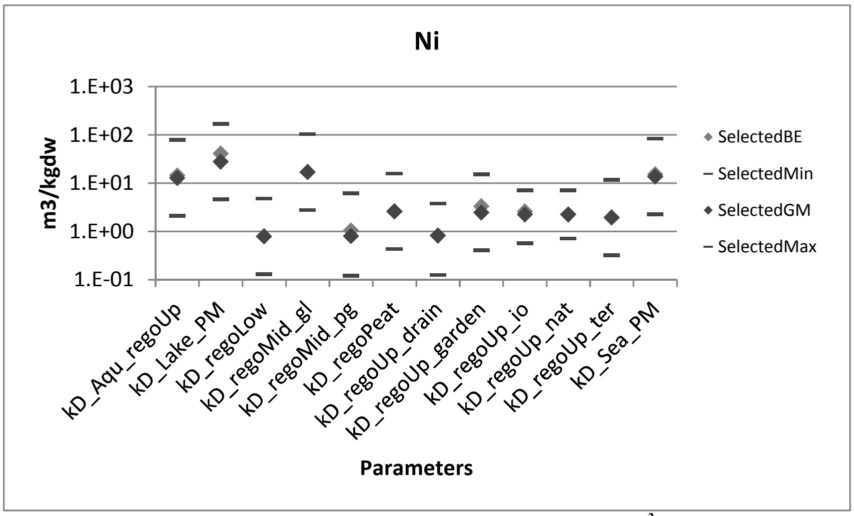


Figure 3-11. Plotted selected data for Ni for all K_d parameters (m^3/kgdw).

3.4.2 Example cR_lake_Fish for Np

For CR for Np for limnic fish data are scarce and only one sample is available in ERICA. Therefore this is not assumed to be a reasonable data selection. In this case a manual update is done via the FinalTable_cond_fix iteration. The element analogue La, is selected by writing La in the column AR "A_element" (cf. Figure 3-17). In the next iteration data for La will replace the initially selected data for Np and a note is given that the selected data is based on the element analogue La. The selection of La as a suitable element analogue for Np is discussed in Section 2.7.3 in Tröjbom et al. (2013).

Figure 3-12 shows the finally selected data for the cR_Lake_Fish parameter where La is utilised as an element analogue for Np. The data for Np is found in the lower left end, together with other actinides and lanthanides, which is assumed to be reasonable and a consequence of using La as element analogue. In Figure 3-13 where all limnic and marine CR for Np are plotted, the CR value for fish is lower than the CR value for other limnic biota (crayfish or mussels) but the CR value for marine fish is similar. These differences could be due to uncertainties in the underlying La data.

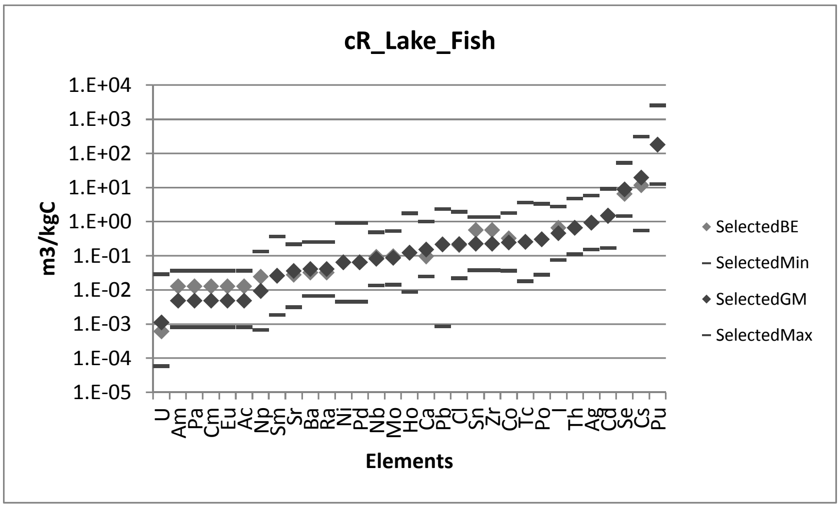


Figure 3-12. Plotted selected data for the parameter cR_Lake_Fish, sorted from the lowest value to the highest

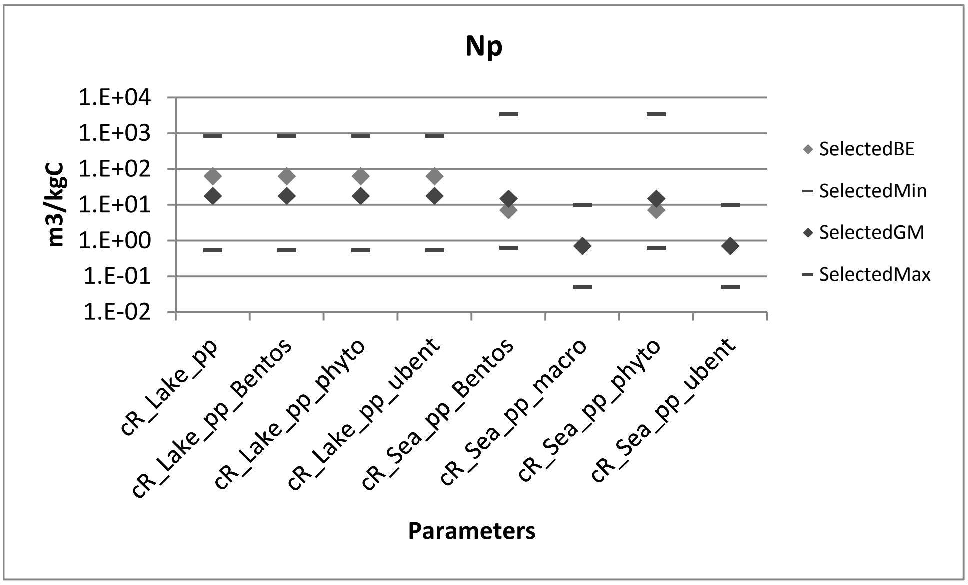


Figure 3-13. Plotted selected data for Np for all limnic and marine parameters (except NHB parameters).

3.4.3 Example cR_agri_veg for Cs

For vegetables no site data are available. The literature data are therefore assumed to be the most representative data available. Literature data ranges overlaps, and the sample number is high (>>10). This indicates that the initial selected data is reasonable and the initial data selection is not changed during the manual evaluation (no changes are performed in the columns marked green in Figure 3-18. A comment is added that motivates the data selection, which will be included in the chapter describing this parameter in Tröjbom et al. (2013).

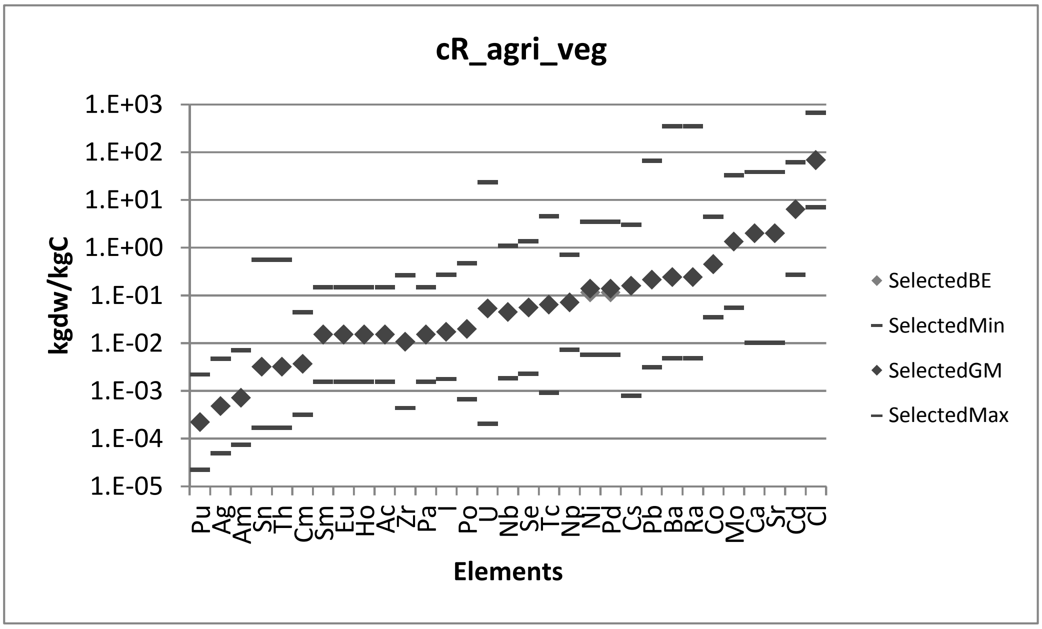


Figure 3-14. Plotted selected data for the parameter cR_agri_veg, sorted from the lowest value to the highest.

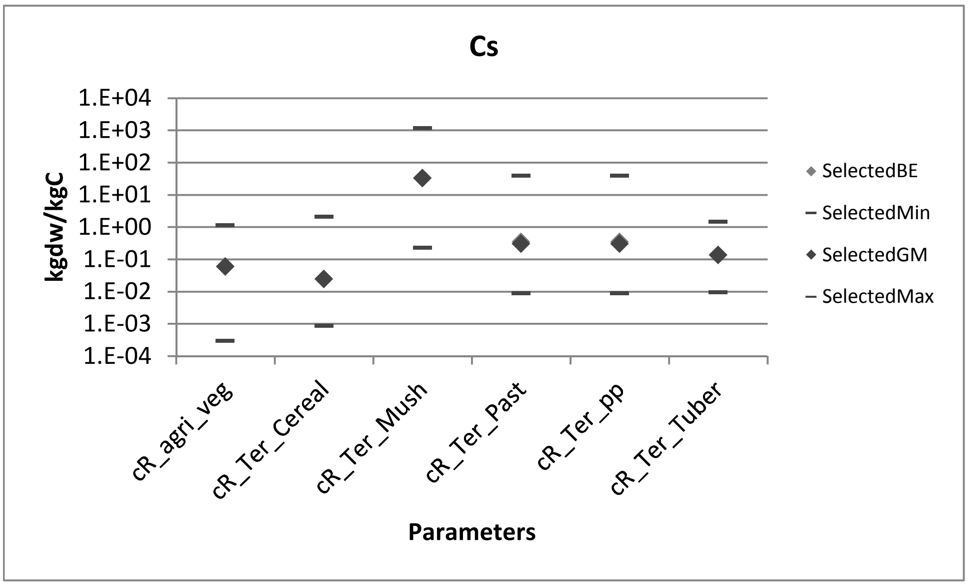


Figure 3-15. Plotted selected data for Cs for all terrestrial CR parameters (except NHB parameters).

Figure 3-16. The manual data selection for K_d_regoLow for Ni is illustrated in this figure. The column marked green could be used if an manual data selected is needed. In this case no manual update has been done.

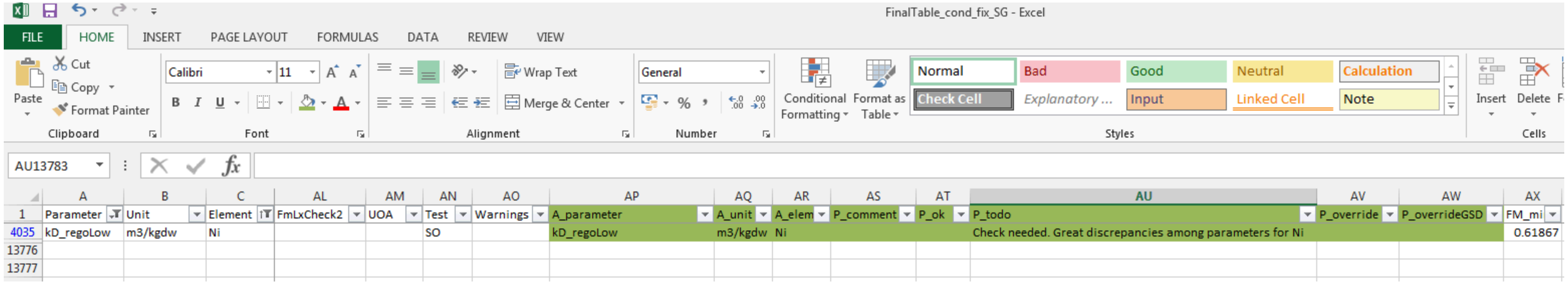


Figure 3-17. The manual data selection for cR_lake_fish for Np is illustrated in this figure. The column marked green could be used if a manual data selected is needed. In this case the available data are scares and instead an element analogue, La, is selected. This is done by writing La in column AR "A_element". This means that the initial data will be overwritten by the data for La in the next step.

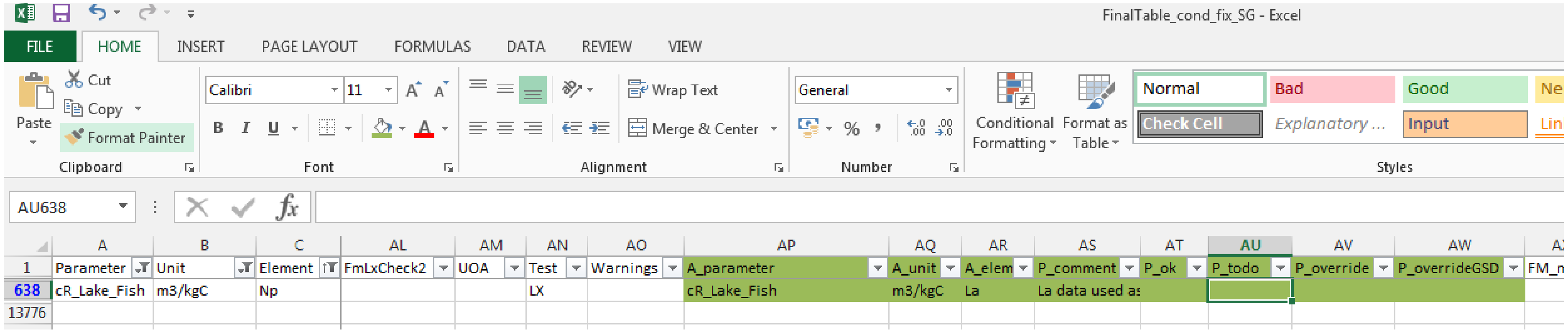
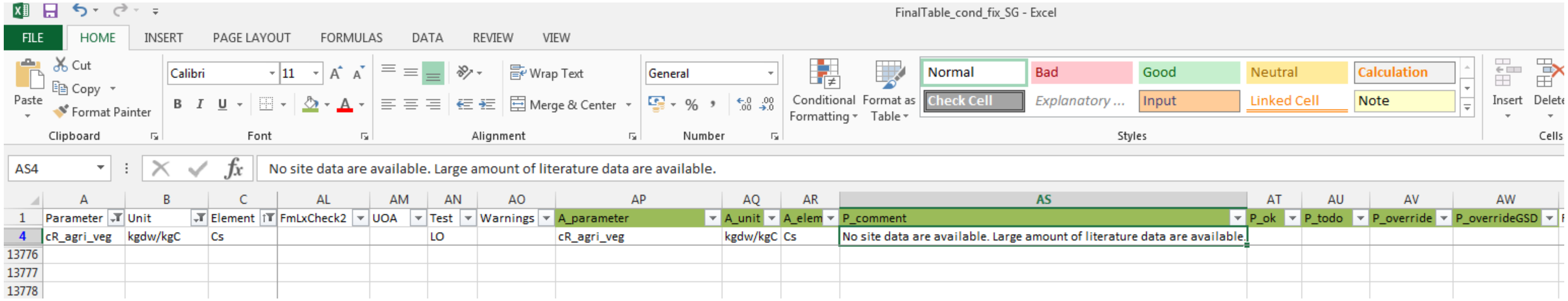


Figure 3-18. The manual data selection for cR_agri_veg for Cs is illustrated in this figure. The column marked green could be used if a manual data selected is needed. In this case the initial data selection is assumed to be the best data selection and no manual change is done. A comment is added in column AS, this column will be imported into the database.



4 References

Beresford N A, Barnett C L, Howard B J, Scott W A, Brown J E, Copplestone D, 2008. Derivation of transfer parameters for use within the ERICA Tool and the default concentration ratios for terrestrial biota. Journal of Environmental Radioactivity 99, 1393–1407.

Hosseini A, Thørring H, Brown J E, Saxén R, Ilus E, 2008. Transfer of radionuclides in aquatic ecosystems – Default concentration ratios for aquatic biota in the Erica Tool. Journal of Environmental Radioactivity 99, 1408–1429.

IAEA, 2010. Handbook of parameter values for the prediction of radionuclide transfer in terrestrial and freshwater environments. Vienna: International Atomic Energy Agency. (Technical Report Series 472)

ICRP, 2011. Wildlife Transfer Parameter Database. International Commission on Radiological Protection. Avaliable at: http://www.wildlifetransferdatabase.org [Februari 2011].

Nordén S, Avila R, de la Cruz I, Stenberg K, Grolander S, 2010. Element-specific and constant parameters used for dose calculations in SR-Site. SKB TR-10-07, Svensk Kärnbränslehantering AB.

Sheppard S, Sohlenius G, Omberg L-G, Borgiel M, Grolander S, Nordén S, 2011. Solid/liquid partition coefficients (K_d) and plant/soil concentration ratios (CR) for selected soils, tills and sediments at Forsmark. SKB R-11-24, Svensk Kärnbränslehantering AB.

Tröjbom M, Grolander S, Rensfeldt V, Nordén S, 2013. K_d and CR used for transport calculation in the biosphere in SR-PSU. SKB R-13-01, Svensk Kärnbränslehantering AB.

Yankovich T L, Beresford N A, Wood M D, Aono T, Andersson P, Barnett C L, Bennett P, Brown J E, Fesenko S, Fesenko J, Hosseini A, Howard B J, Johansen M P, Phaneuf M M, Tagami K, Takata H, Twining J R, Uchida S, 2010. Whole-body to tissue concentration ratios for use in biota dose assessments for animals. Radiation and Environmental Biophysics 49, 549–565.

Appendix A : Detailed technical description of database objects and code

Contents

1 Introduction

This Appendix contains two detailed technical descriptions of the two Access databases, i) SKB_chemistry_SR_PSU.accdb usen for utilised for estimation of site specific parameter values and compilation of literature data (Section 2), and ii) SKB_K_d_CR.accdb, were the combination datasources and selection of parameter values described in Section 3.
This Appendix lists the input files, the intermediate steps (queries and macros) and the output files of the two databases. This document should be used as a technical description of the databases when reviewing the implementation of the K_d/CR parameterisation process in the two databases. In Section 3.4 a list of all steps conducted in the databases is given, these step should be followed in order to replicate the calculation of the final parameter values.

2 Database for estimation of site specific parameter values – SR_PSU_Chemistry

The purpose with this section is to give a detailed description of the database used when estimating site-specific K_d and CR parameter values from site data. All calculations are handled in the Microsoft Access database "SR_PSU_Chemistry", with traceable links between the tables of original concentration data and the final output parameters values. In the sections below, tables with input data, intermediate result tables and output tables are described together with macros that give the final results.

2.1 Input data

This section lists the input data that have to imported into the database

2.1.1 Site specific concentration data – Regolith&Biota and Water tables

Within the SR-PSU project a compilation of quality controlled data on element concentrations in water, regolith and biota from both Forsmark (FM) and Laxemar (LX) are used. This compilation is stored in the database "SKB_Chemistry_SR_PSU" at SKB SR-PSU SVN (svn://svn.skb.se/projekt/SFR/SR-PSU/Indata/Chemistry/SKB_Chemistry_SR_PSU.accdb).

This data compilation is based on two data files originally assembled for the previous safety assessment SR-Site[7], combined with recent data from Sicada[8]. In the SR_PSU_Chemistry database regolith and biota data are stored in the "Regolith&Biota" table, whereas all water chemistry data are stored in the "Water" table. All concentration data in each table are given in the same units as described in the layout fields in the database tables.

Table 2-1. The origin of the Water and Regolith&Biota tables.

Table in this database	Original file at SR-PSU SVN	Original table
Water	SKB_Chemistry_SR_PSU.accdb	Water
Regolith&Biota	SKB_Chemistry_SR_PSU.accdb	Regolith&Biota

2.1.2 Coupling parameters to data_ver3

This input file contains tables where site specific data are compiled and coupled to specific parameters. This file is the basis for the manual selection and coupling of site data to specific parameters in the SR-PSU parameterisation. The sheets are described in the bullet list below, and the corresponding tables in the database are shown in Table 2-2 together with import scripts.

[7] Forsmark Chemistry water deposits biota 090313.xls and (Simpevarp Chemistry water deposits biota 090317.xls, in turn based on Sicada orders 09_28_2, 09_010_1, 09_010, 09_007_2, 09_001 and 08_187

[8] Sicada orders 2011_52, 2011_60 and 2011_67.

The file contains the following tables:

- Parameters: Compilation of parameters names, and descriptions of parameters. Parameter specific information is also compiled here, for example GSD (Geometric standard deviation) limits.
- Biota, Regolith and Water: Detailed categorisations of site data where each category is coupled to specific parameter names in the “to” and “from” columns. Data coupled to the same parameter according to this table, and that fulfils other selection criteria such as geographical location etc., are combined for the estimation of site specific K_d and CR ratios. The coupling at category level is performed in order to reduce the amount of manual work.
- Details_biota, details_regolith, details_water: These tables contain all available site data at sample level. The purpose with this table is to enable deselection of specific samples in the “excl” columns. “Excl1” means that the whole row is discarded, “Excl2” that REE data are discarded. The other Excl columns have not been used in the present version of the database. Any deselection in this table is motivated in the “ExComment” column. The categories listed in the Biota, Regolith and Water tables, are linked to the samples via ID_biota, ID_soil and ID_water index fields.

Table 2-2. Link between sheets in the Excel file ”Coupling Parameters To Biota_ver3” and the corresponding table in the SR_PSU_Chemistry database.

Sheet in “Coupling parameters to data_ver3”	**Table in database**	**Import script**
Parameters	ParamTab	Importera-ParamTab
Biota	Cat_biota_pclass	Importera-cat_biota_pclass
Regolith	Cat_soil_pclass	Importera-cat_soil_pclass
Water	Cat_water_pclass	Importera-cat_water_pclass
Details_biota	Details_biota_excl	Importera-Coupling parameters to data_ver3_details_biota_excl
Details_soil	Details_soil_excl	Importera-Coupling parameters to data_ver3_details_soil_excl
Details_water	Details_water_excl	Importera-Coupling parameters to data_ver3_details_water_excl

2.1.3 Conversion factors - CR_wh_t table

This table contain factors used for conversion from muscle tissue CR to whole body CR of NHB parameters. This table is found in Appendix C of the report.

2.1.4 MissingCarbon_Ex

The MissingCarbon_Ex table contain carbon content information derived from the “DMC & CC database” for the site data categories with missing carbon content information. Sample specific measurements of carbon content are used when available, which is the case for most data. If carbon content data are missing, generic values from IAEA (2010) have been assigned.

2.2 Intermediate steps – tables, queries and macros

This section describes the intermediate steps in the database consisting of dynamic queries, intermediate tables and macros.

2.2.1 Ag_start and Na_start tables

These tables are long versions of the Regolith&Biota and the Water tables, respectively. This means that each element is listed on separate rows with a link to the original table via the ID fields.
These tables are generated via macros according to Table 2-3.

Table 2-3. Macros genererating the long versions of the concentration tables.

Table in Database	Macro
Ag_start	Make_long_soild
Na_start	Make_long_water

2.2.2 ValueList_to and Valuelist_from tables

These tables contain concentration data coupled to each parameter as specified in the "Coupling Parameters To Data_ver3" file. These tables are generated via several queries included in the macros listed in Table 2-4 below.

Table 2-4. Macros generating ValueList_to and ValueList_from tables

Table in database	Macro
ValueList_to	Skapa_valueList_to
ValueList_from	Skapa_valueList_from

Within these macros, a query adds carbon content data from the table MissingCarbon_Ex, when site specific carbon content data are missing (update queries U_missingcarbon_from, and U_missingcarbon_to).

2.2.3 Details_excl table

This table compiles all Sicada sample numbers that have been specified in the Details_biota_excl, Details_soil_excl and Details_ewater_excl tables. A macro generates this table according to Table 2-5.

Table 2-5. Macro generating the Detail_excl table.

Table in database	Macro
Details_Excl	Skapa_details_excl

2.2.4 ValueList_from_pos and ValueListTo_pos queries

These tables contain concentration data coupled to each parameter as described in the section above, but with negative values excluded. Samples specified in the Details_excl tables are also discarded. These tables are generated as dynamic queries in the database as specified in Table 2-6.

Table 2-6. Questions genererating the ValueList_from_pos and ValueList_to_pos dynamic tables.

Query in database
ValueList_from_pos
ValueList_to_pos

2.2.5 K_d_match_idcode, K_d_match_idcode_Nfrom_N and K_d_match_idcode_Nto_N queries

These queries couples information in the queries ValueList_from_pos, ValueList_to_pos and ParamTab and generates statistics of K_d values based on all possible concentration pairs combined according to site, parameter, idcode and depth. Min, max, GM and GSD are calculated. The _N queries contain information about the number of unique samples in the "to" and "from" datasets based on unique SKB sample numbers. Since K_ds and CRs can be estimated using an uneven amount of source and recipient data, there will be several possible ways of reporting sample number (N) for a given K_d or CR estimation. N is reported for both the source (Nfrom) and recipient (Nto) data, and the lower of the two is generally regarded as the critical N.
The questions are related according to each other according to Table 2-7.

Table 2-7. Queries and sub-queries generating K_d statistics for concentration pairs.

Query	Sub query
K_d_match_idcode	
K_d_match_idcode_Nfrom_N	K_d_match_idcode_Nfrom
K_d_match_idcode_Nto_N	K_d_match_idcode_Nto

2.2.6 CRws_match_ecosystem, CRws_match_ecosystem_Nfrom, CRws_match_ecosystem_Nto, CRb_match_ecosystem, CRb_match_ecosystem_Nfrom and CRb_match_ecosystem_Nto queries

These queries couples information in the queries ValueList_from_pos, ValueList_to_pos and ParamTab and generates statistics of CR values based on all possible concentration pairs combined according to site and parameter. Min, max, GM and GSD are calculated. The _N queries contain information about the number of unique samples in the "to" and "from" datasets based on unique SKB sample numbers. Since K_ds and CRs can be estimated using an uneven amount of source and recipient data, there will be several possible ways of reporting sample number (N) for a given K_d or CR estimation. N is reported for both the source (Nfrom) and recipient (Nto) data, and the lowest of the two is generally regarded as the critical N.
The questions are related according to each other according to Table 2-8.

Table 2-8. Queries and sub-queries generating CR statistics for concentration pairs.

Query	Sub query	Comment
CRws_match_ecosystem		Limnic and marine
CRws_match_ecosystem_Nfrom_N	CRws_match_ecosystem_Nfrom	Limnic and marine
CRws_match_ecosystem_Nto_N	CRws_match_ecosystem_Nto	Limnic and marine
CRb_match_ecosystem		Terrestrial biota
CRb_match_ecosystem_Nfrom_N	CRws_match_ecosystem_Nfrom	Terrestrial biota
CRb_match_ecosystem_Nto_N	CRws_match_ecosystem_Nto	Terrestrial biota

2.3 Output

This section describes the output tables from the database.

2.3.1 Export_kdCR_database

The export_KdCR_database table compiles all K_d and CR data estimated from site data according to the information in the "Coupling Parameters To Data_ver3" file and assumptions built in the intermediate queries. This table is generated by several queries within a macro (see Table 2-9) and data are also post processed in order to update nomenclature and data (cf.

Table 2-10).

Table 2-9. Macro generating the Export_Kd_CR_database table.

Table in database	**Macro**
Export_KdCR_database	Skapa_Export_KdCR_database

Table 2-10. Post processing within the Skapa_Export_KdCR_database macro

Query in database	**Description**
Change_Reference_FM	Copy info to reference
Change_Reference_LX	Copy info to reference
Change_Reference_F&L	Copy info to reference
Change_Ra226_to_Ra	Change parameter Ra226 to Ra
Uppdat_NHB_factors	Update NHB_factors according to information in table CR_wh_t

2.3.2 Matched site data

This file contains detailed information of which samples have been selected per parameter according to the information in "Coupling parameters to data_ver3". The macro "Skapa Matched_categories" generates the output tables described in Table 2-11. The "Matched categories" table shows per parameter which categories of data are combined when the K_d and CR ratios are calculated. Matched_SNO shows on sample level which samples have been combined when the K_d and CR ratios are calculated.

Table 2-11. Tables included in the "Matched site data" file. These tables are genereated by the macro "Skapa_Matched_categories".

Output file	**Sheet**	**Table in database**
Matched site data_xlsx	Matched categories	Matched_categories
	Matched Sample Numbers	Matched_SNO

2.4 Summary of the steps generating site specific K_d and CR values

The following steps should be conducted in order to replicate the calculation of site specific K_d and CR values:

1. Run macros "make_long_solid" and "make_long_water" in order to generate the long concentration tables from the "Regolith&biota" and "Water" tables.

2. Import tables from "Coupling parameters to data_ver3" with saved imports according to Table 2-2.
3. Run macro "skapa_details_excl" in order to generate the Details_excl table.
4. Run macros "skapa_valuelist_from" and "skapa_valuelist_to" in order to generate the "valuelist_to" and "valuelist_from" tables.
5. Run maco "skapa_Export_KdCR_database" in order to generate the output table "Export_KdCR_database"
6. Run the macro "Skapa_Matched_Categories" to generate the tables for the "Matched Site data" file.

3 Database for combination of data sources and selection of parameter values – SKB_Kd_CR

The purpose with this section is to give a detailed description of the database used when combining K_d and CR parameter values from different data sources. All calculations are handled in the Microsoft Access database "SKB_Kd_CR", with traceable links between the input tables and the final output tables. The output tables for data deliveries and report are dynamically linked to the original data sources via the internal data structure and the input tables that contain conversion factors, matching information and manual assignments regarding analogues and manual overrides. This means that if the complete database is stored, identical results could be replicated as the database file contains all input information as well as coupling information needed. In the sections below, tables with input data, intermediate result tables and output tables are described together with macros that give the final results.

3.1 Input data

In this section, the different input data tables are described.

3.1.1 The main "K_dCR" table

In this main table all K_d and CR data are compiled in original units and formats. This table contains both site specific parameters values and all literature data merged from different sources (cf. data chapter in report).

Site specific data from the file "Export_KdCR_database", described in Section 2.3.1 are copied into the K_dCR table at update according to the procedure described below:

1. Delete previous site data with the query "Tabort_sitedata".
2. Import the new "Export_KdCR_database " table with the import script "Importera PSUprel". If the "SR_PSU_Chemistry" database is put in a new folder, the import scripts have to be updated.
3. Change name of the imported Export_KdCR_database1 to Export_KdCR_database, to replace previous version.
4. Add new site data with the query "f_KdCR_PSUprel".

3.1.2 ParamClass

Paramclass is an Excel interface file where different data sources are coupled to parameter names and literature sources are graded in different literature classes (L1 to L4) depending on relevance for the parameter (Table 3-1). The information in ParamClass is used to combine literature data with site specific data, and to grade literature data sources for the initial automated selection process.

Table 3-1. Import script and origin of the ParamClass table.

Table in database	Origin	Import script
ParamClass	ParamClass.xls, Sheet ParamClass	Importera-Paramclass

In ParamClass, all available data sources in the K_dCR table are listed. In the ParamPSU, LitClass and "Kommentarer" fields, the PSU-parameter name coupled to this information is

added together with literature class L1 to L4 for literature data, FM, LX or FMLX for site data, and comment, respectively.
If new data is added to the K_dCR table, the macro “Lägg_till_nya_paramclasses” adds new rows to the ParamClass table for further coupling to PSU parameters and classifications. This updated table might replace the external excelfile.

3.1.3 ParamTab

The ParamTab table is a compilation of parameter names and descriptions of parameters. Parameter specific information is also compiled here, for example GSD limits (Table 3-2).

Table 3-2. Import script and origin of the ParamTab table.

Table in database	Origin	Import script
ParamTab	Coupling parameters to data_ver3.xlsx, sheet “Parameters”	ParamTab

3.1.4 Elementgruppering

This table contain information about specific elements, for example relevance for SR-PSU assessment, periodic group etc. Import script is listed in Table 3-3.

Table 3-3. Import script and origin of the ”Elementgruppering” table.

Table in database	Origin	Import script
ParamTab	Gruppering av element.xlsx, sheet “Elementgruppering”	Importera-Gruppering av element

3.1.5 DataPerCategory

This table compiles generic information of dry weight and carbon content data from IAEA (2010). The origin of this table is shown in
Table 3-4. If new literature data are added, these categories must be included in the DataPerCategory table in order to achieve conversion factors (cf. Section 3.2.1).

Table 3-4. Script generating the DataPerCategory table.

Table in database	Origin	Import script
DataPerCategory	DMC & CC database.xlsx, sheet “DataPerCategory”	Importera-DMC & CC database

3.1.6 Leverans_elementspecific_mall

This table contains the table structure needed for the delivery file to Ecolego.

Table 3-5. Script generating the Leverans_elementspecific_mall table.

Table in database	Origin	Import script
Leverans_elementspecific_mall	ElementSpecific_mall.xlsx	Importera-ElementSpecific_mall

3.1.7 NHB_unitConv_coupling_parameters_to_biotainfo

This table couples NHB parameter fields “biota_type” and “biota_type_detail” to the biota description fields in the “DataPerCategory” table. This table is originally edited in Excel and imported into Access.

Table 3-6. NHB_unitConv_coupling_parameters_to_biotainfo table.

Table in database	Origin	Import script
NHB_unitConv_coupling_parameters_to_biotainfo	Not saved	Not saved

3.1.8 Leverans_unit_conversion

This table contain a conversion list for the nomenclature of units in the database and the nomenclature in the delivery files to Ecolego.

3.2 Intermediate steps – tables, queries and macros

In this section intermediate steps are described processing the input data into the output tables. For some tables, the iterative selection process makes the division in input and intermediate data not clear.

3.2.1 ConversionFactors

This table compiles conversion factors used for unit conversions of literature data. This table is generated automatically by the macro “Skapa_ny_conversionfactors” based on information in the table “DataPerCategory” (Table 3-7). If new data categories are added to the table DataPerCategory, conversion factors are added for these categories in the ConversionFactors table. If new literature data are added, these categories must be included in the DataPerCategory table in order to get conversion factors for these categories when the macro is run.

Table 3-7. The macro generating the ConversionFactorsTable from information in the DataPerCategory table.

Table in database	Based on table	Macro
ConversionFactors	DataPerCategory	Skapa_ny_conversionfactors

The conversion factors are calculated according to the equations listed in Table 3-8.

Table 3-8. Compilation of formulas for calculation of conversion factors. CC is the carbon content per dry weight (kgC/kgdw) and DMC the dry matter content (kgdw/kgfw).

	To				
From	**kgdw/kgfw**	**kgdw/kgC**	**kgdw/kgdw**		
kgdw/kgC	CC*DMC	1	CC		
kgdw/kgdw	DMC	1/CC	1		
kgdw/kgfw	1	1/(CC*DMC)	1/DMC		
	L/kgdw	**m3/kgC**	**m3/kgdw**	**m3/kgfw**	**L/kgfw**
L/kgdw	1	1/(1000*CC)	1/1000	DMC/1000	DMC
L/kgfw	1/DMC	1/(CC*DMC*1000)	1/(DMC*1000)	1/1000	1

m^3/kgC	100*CC	1	CC	CC*DMC	1000*CC*DMC
m^3/kgdw	1000	1/CC	1	DMC/1000	1000*DMC
m^3/kgfw	1000/DMC	1/(CC*DMC)	1/DMC	1	1000
kgdw/kgC	CC*DMC	1	CC		

3.2.2 KdCR_convPSU

This query, which couples information from the main K_dCR table and the ConversionFactors table, converts data from the K_dCR table into several different units based on conversion factors from the ConversionFactors table.
In the K_d/CR database table most data are given as geometric means and geometric standard deviations as defined in the distribution field. In case of arithmetic measures, the corresponding geometric estimates are recalculated dynamically according to the equations below (GM=geometric mean, GSD=geometric standard deviation, AM=arithmetic mean, SD= arithmetic standard deviation).

$$GM = \frac{\mathrm{AM}}{\sqrt{e^{\sigma^2}}} \quad , \qquad GSD = e^{\sigma} \quad , \qquad \sigma = \sqrt{ln\left(\left(\frac{SD}{AM}\right)^2 + 1\right)}$$

3.2.3 NHB_unitConv_DMC&CC_Tab

This table couples information from the “DataPerCategory” table to NHB parameters for conversions from dry weight to fresh weight data. This table is based on the table “NHB_unitConv_coupling_parameters_to_biotainfo” and the “DataPerCategory” table.

Table 3-9. NHB_unitConv_DMC&CC_Tab table.

Table in database	Table query
NHB_unitConv_DMC&CC_Tab	NHB_unitconv_DMC&CC

3.2.4 FinalTable

This main table compiles information via several linked sub-queries based on input from the KdCR_convPSU query and the ParamClass table. This table gathers per row information from different data sources as specified in the ParamClass table (per parameter x element x unit combination).

Table 3-10. The macro generating the FinalTable.

Table in database	Based on	Macro
FinalTable	KdCR_conv ParamClass	Skapa_FinalTable

In Table 3-11, the sub queries are listed that are called when the “Skapa_FinalTable” macro is run.

Table 3-11. Detailed description of the sub queries called when the "skapa_FinalTable" macro is run. Dotted cells denote additional variants according to the description of the queries shown on the row above.

Query in database	Based on table/query	Description
GM_m3/kgdw	KdCR_convPSU ParamClass	Compiles GM for the unit m3/kgdw
…	KdCR_convPSU ParamClass	Similar queries per statistical measure (N, GM, GSD, min, max, Nfrom, Nto, MT, MTD, BT, BTD, Com, Ref) for the units listed in Table 3-8.
Fin_m3/kgdw	N_m3/kgdw GM_m3/kgdw GSD_m3/kgdw min_m3/kgdw max_m3/kgdw Nfrom_m3/kgdw Nto_m3/kgdw MT_m3/kgdw MTD_m3/kgdw BT_m3/kgdw BTD_m3/kgdw Com_m3/kgdw Ref_m3/kgdw	This query compiles all statistical information for the unit m3/kgdw into a single table
…	…	Similar queries per unit
t_FinalTable_kgdw/kgdw	Fin_kgdw/kgdw	Query adding rows to FinalTable
f_FinalTable_kgC/kgC	Fin_kgC/kgC	Query adding rows to FinalTable
f_FinalTable_m3/kgC	Fin_m3/kgC	Query adding rows to FinalTable
f_FinalTable_kgdw/kgC	Fin_kgde/kgC	Query adding rows to FinalTable
f_FinalTable_m3/kgdw	Fin_m3/kgdw	Query adding rows to FinalTable
f_FinalTable_day/kgfw	Fin_day/kgfw	Query adding rows to FinalTable
f_FinalTable_day/L	Fin_day/L	Query adding rows to FinalTable

3.2.5 ElemXParamXunit

This table contain all possible element, parameter and unit combinations as specified in the tables ParamTab and ElementGruppering. The generation of this table is included in the macro generating AnalogueTable in Section 3.2.6 below.

3.2.6 AnalogueTable

AnalogueTable contain information of all manual assignments regarding assumptions of element- and parameter analogues, as well as manual overrides and comments. The macro "Uppdatera AnalogueTable" generates this table from several input files as described in Table 3-12 below. Before the macro is run the tables listed in

Table 3-13 have to be updated through import scripts. Information in AnalogueTable is updated through the iterative manual selection process via FinalTable_cond_fix files (cf. later section for a description of this file).

Table 3-12. Queries included in the "Uppdatera AnalogueTable" macro.

Query in macro	**Description**
t_ElemXParamXdw	Creates sub table structure for dry weight parameters
t_ElemXParamXC	Creates sub table structure for carbon normalised parameters
Compl_Analog_Table	Adds new parameters to AnalogueTable
AnalogueTable_uppdat_MT	Adds manual information to AnalogueTable from MT
AnalogueTable_uppdat_SN	Adds manual information to AnalogueTable from SN
AnalogueTable_uppdat_SG	Adds manual information to AnalogueTable from SG
AnalogueTable_uppdat_VR	Adds manual information to AnalogueTable from VR
AnalogueTable_copy_CtoDW	Copies all information from carbon normalised parameters (CR) to corresponding dry weigth rows.
Uppdat_overrideGSD_when_PA	When parameter analogues are used, overrideGSD is automatically set to GSDmax

Table 3-13. Tables and corresponding scripts for updating AnalogueTable

Table in database	Origin	Import script
FinalTable_cond_fix_MT	FinalTable_cond_fix_MT.xlsx	Importera-FinalTable_cond_fix_MT
FinalTable_cond_fix_SN	FinalTable_cond_fix_SN.xlsx	Importera-FinalTable_cond_fix_SN
FinalTable_cond_fix_SG	FinalTable_cond_fix_SG.xlsx	Importera-FinalTable_cond_fix_SG
FinalTable_cond_fix_VR	FinalTable_cond_fix_VR.xlsx	Importera-FinalTable_cond_fix_VR

3.2.7 FinalTable_postproc

This intermediate query adds additional statistical information to FinalTable, for example 5th and 95th percentiles. All rows from the table elemXparamXunit are combined with the rows from FinalTable.
Due to limitations in the number of possible input arguments in Access functions, alternative percentiles are calculated that can be used if GSD adjustments are made at later stage (cf. Section 3.2.9):

- **LX_5p_sel**: OOM([Lx_n]<=3;OOM([Lx_GSD]<=[GSDmax] ELLER ÄrNull([Lx_GSD]);[Lx_gm]/([GSDmax]^1.645);[Lx_gm]/([Lx_GSD]^1.645));OOM([Lx_GSD]<[GSDmean];[Lx_gm]/([GSdmean]^1.645);[Lx_gm]/([Lx_GSD]^1.645)))
- **LX_95p_sel**: OOM([Lx_n]<=3;OOM([Lx_GSD]<=[GSDmax] ELLER ÄrNull([Lx_GSD]);[Lx_gm]*([GSDmax]^1.645);[Lx_gm]*([Lx_GSD]^1.645));OOM([Lx_GSD]<[GSDmean];[Lx_gm]*([GSdmean]^1.645);[Lx_gm]*([Lx_GSD]^1.645)))

3.2.8 FinalTable_Anapost

This query combines information from FinalTable (via FinalTable_postproc) and Analogue table, which contains information from the manual assignments of analogues. This query copies information from the selected analogue, into the row of the element and parameter combination for which the analogue is assigned.
Due to the limitations in the number of possible input arguments in Access functions, 5th and 95th percentiles are adjusted if GSDmax is assigned via P_overrideGSD:

- **LX_5p_sel2:** OOM([P_overrideGSD]="GSDmax";[finaltable_postproc_1].[LX_gm]/[finaltable_postproc_1].[GSDmax]^1.645;[finaltable_postproc_1].[LX_5p_sel])
- **LX_95p_sel2:** OOM([p_overrideGSD]="GSDmax";[finaltable_postproc_1].[LX_gm]*[finaltable_postproc_1].[GSDmax]^1.645;[finaltable_postproc_1].[LX_95p_sel])

3.2.9 FinalTable_cond

This major query compiles all information from FinalTable_Anapost, and calls the Selectfunctions VBA code that make all automated parameter value selections based on the selection criteria (Cf. Section 3.5).

3.2.10 FinalTable_cond_NHB_fw

This table contain information from FinalTable_cond combined with parameter values in fresh weight units for the NHB parameters from Table_A2_NHB.

Table 3-14. NHB_unitConv_DMC&CC_Tab table.

Table in database	**Table query**
FinalTable_cond_NHB_fw	t_FinalTable_cond_NHBfw

3.3 Output data

This section describes the different output tables that comprise input to models and for tables and figures in the report. Some files are only used as support during the parameterisation work.

3.3.1 FinalTable_cond_fix

This table is a fixed version of FinalTable_cond used for updated manual assignments in Excel (Table 3-15). The updated versions of this file is imported via the scripts listed in

Table 3-13. In the current version, the data types have to be changed manually in the following rows:

- FM_Nfrom, FM_Nto, korrN_FM, korrN_LX, korrN_F_L, SelectedN are changed to Long integer
- GSDp, SelectedMin, SelectedMax, SelectedGM, SelectedBE, SelectedGSD are changed to Double

Table 3-15. Table for manual assignments during the parameterisation.

Output file	Table in database	Query generating the table
FinaTable_cond_fix.xlsx	FinalTable_cond_fix	t_FinalTable_cond_fix

3.3.2 Leveranstabell_elementspecific_Kd_CR

This table is the parameter output delivery file to the radionuclide dose model used for assessing dose to human. The output format is adapted to the import specifications of the EcoLego model.

Table 3-16. Parameter delivery table for radionuclide dose model for human biota.

Output file	Query in database
Leveranstabell_elementspecific_Kd_CR.xlsx	Leverans_elementspecific_tabell

3.3.3 Table_A1

This table is the basis for the tables in the report that compile available site and literature data per parameter and element according to Table 3-17.

Table 3-17. Compilation of availble site and literature data per element.

Output file	Table in database	Query generating the table
Table_A1.xlsx	tabell_A1	t_Tabell_A1

3.3.4 Table_A2

This table is the basis for the tables in the report that compile selected parameter values per parameter and element according to Table 3-18. Before export to Excel file, the field SelN needs to be manually changed to data type Long Integer, SelMin, SelGM, SelBE, SelMax and SelGSD need to be manually changed to data type Double.

Table 3-18. Compilation of selected parameter values per parameter and element for the human biota dose assessment.

Output file	Table in database	Query generating the table
Table_A2.xlsx	tabell_A2	t_Tabell_A2

3.3.5 FinalTable_cond_Allinfo2

This query summarises all available info per parameter and element (Table 3-19). Both selected values and detailed information of the data sources based on analogue assumptions, and information for the specific element and parameter combination without influence from these assumptions (i.e. the underlying data discarded when an analogue is assigned).

Table 3-19. Compilation of all information per element and parameter combination.

Output file	Query in database
FinalTable_cond_Allinfo2.xlsx	FinalTable_AllInfo2

3.3.6 “Figurer Parameter Jämförelser”

This table contains information used for parameter comparisons in the summary sections of the parameter descriptions. The figures are found the file “Figurer Parameter Jämförelser – Kopia.xlsx”

Table 3-20. Compilation of all information per element and parameter combination.

Output file	Query in database
Figurer Parameter Jämförelser.xlsx	Figurer Parameter Jämförelser

3.3.7 Table_A2_NHB

This table is the basis for the tables in the report that compile selected parameter values per parameter and element according to Table 3-21. Before export to Excel file, the field “GSD” need to be manually changed to data type Double.
This table is based on the parameter values calculated per dry weight in FinalTable_cond, converted in t fresh weight values by the use of generic data compiled in “NHB_unitConv_DMC&CC_Tab” (cf. Section 3.2.3).

Table 3-21. Compilation of selected parameter values per parameter and element for the non-human biota dose assessment.

Output file	Table in database	Queries generating the table
Table_A2_NHB.xlsx	tabell_A2_NHB	t_Tabell_A2_NHB_3 t_finalTable_cond_NHB_fw

3.3.8 Leverans_elementsspecific_tabell_NHB

This table is the parameter output delivery file to the radionuclide dose model used for assessing dose to non-human biota. The output format is adapted to the import specifications of the EcoLego model. This table is based on the table “FinalTable_Cond_fw”, “ParamTab” and “Leverans_unit_conversion”.

Table 3-22. Parameter delivery table for radionuclide dose model for human biota.

Output file	Query in database
Leveranstabell_elementspecific_CR_NHB.xlsx	Leverans_elementsspecific_tabell_NHB

3.3.9 SummaryFinalTable

This file summarises information based in several queries, as listed in

Table 3-23.

Table 3-23. Queries included in the SummaryFinalTable Excel file.

Output file	Sheet	Query in database
SummaryFinalTable.xlsx	OKcheck	PSUresut_crosstab_test_details
	Analogues	PSUresut_crosstab_test_analogues
	EA&PA	PSUresut_crosstab_test_EA&PA

3.3.10 EA Summarys

This file summarises information about selected element and parameter analogues based on two queries, as listed in Table 3-24.

Table 3-24. Queries included in the "EA summarys" Excel file.

Output file	Sheet	Query in database
EA Summarys.xlsx	EA matris	UsedAsEA_matris1
	Used as EA	UsedAsEA

3.3.11 FinalTable_data_plots

This table contain data for plots of selected values in the report.

Table 3-25. Table containing data for plots of selected values in the report.

Output file	Table in database	Query generating the table
FinalTable_data_to_plots.xlsx	FinalTable_data_plotsTAB	t_FinalTable_data_plots

3.4 Summary of the steps generating final parameter values

The following steps should be conducted in order to replicate the calculation of the final parameter values:
These steps are optional and only necessary if new site data have been added or if data selections have been updated:

1) Update the K_dCR table with new site data as described in Section 3.1.1.
2) Import ParamClass according to Section 3.1.2
3) Import ParamTab according to Section 3.1.3
4) Add new Parameter classes, with the macro "lägg till nya paramclasses" according to Section 3.1.2 (optitional and necessary only if new parameters have been added).
5) Create new ConversionFactors table according to Section 3.2.1

If the manual iteration has been updated, the following steps should be conducted:
Import updated FinalTable_cond_fix_NN files according to the saved imports in

1) Table 3-13.
2) Update analogueTable with the macro “Uppdatera AnalogueTable” (cf. Section 3.2.6).

These steps generates the output tables listed below:

1) Create FinalTable with the macro “skapa FinalTable” (cf. Section 3.2.4)
2) Run table query “t_FinalTable_cond_fix (cf. Section 3.3.1)
3) Run table query “t_FinalTable_data_to_plots (cf. Section 3.3.11)
4) Run table query “t_tabell_A1” (cf. Section 3.3.3).
5) Run table query “t_tabell_A2” (cf. Section 3.3.4)
6) Run table query “t_tabell_A2_NHB_3 (cf. Section 3.3.7).
7) Run t_finalTable_cond_NHB_fw (cf. Section 3.2.10).

Listing of the output files generated. See referenced sections for detailed descriptions of each file.

1) Leverans_elementsspecific_tabell_NHB (Section 3.3.8).
2) Leveranstabell_elementspecific_Kd_CR (Section 3.3.2).
3) EA summarys (Section 3.3.10).
4) Table A2_NHB (Section 3.3.7).
5) SummaryFinalTable (Section 3.3.9).
6) FinalTable_data_to_plots (Section 3.3.11).
7) Table_A1 (Section 3.3.3).
8) Table_A2 (Section 3.3.4).
9) FinalTable_cond_Allinfo2 (Section 3.3.5
10) FinalTable_cond_fix (Section 3.3.1).
11) Figurer parameter jämförelser (Section 3.3.6).
12) Matched site data (Section 2.3.2).

3.5 VBA code for data selections and evaluation

This Section lists the VBA code called from the FinalTable_Cond query.
The following functions are included in the VBA modules Selectfunctions and Classifications:

Function	Description
Min2Val	Returns the minimum value among two numbers
Max2Val	Returns the maximum value among two numbers
Max4Val	Returns the maximum value among four numbers
Max4Val	Returns the maximum value among four numbers
Sel_N	Returns the number of samples of the selected data source
Sel_Min	Returns the minimum value of the selected data source
Sel_Max	Returns the maximum value of the selected data source
Sel_GM	Returns the geometric mean of the selected data source
Sel_BE	Returns the best estimate of the selected data source
Sel_RefBE	Returns the reference of the BE of the selected data source
Sel_GSD	Returns the geometric standard deviation of the selected data source
Sel_Cond	Returns the condition number according to the selection criteria
Sel_RefMin	Returns the origin of the minimum value (min or 5-percentile)
Sel_RefMax	Returns the origin of the maximum value (max or 95-percentile)
Sel_RefGSD	Returns the origin of the GSD value (GSD, GSDmin, GSDmax)
Sel_Ref	Returns the data source selected
Sel_MT	Returns the media type of the selected data source
Sel_MTD	Returns the media type detail of the selected data source
Sel_BT	Returns the biota type of the selected data source
Sel_BTD	Returns the biota type detail of the selected data source
Sel_Com	Returns the comment from the Kd/CR database
Sel_Dsource	Returns the data source selected
L_valid	Returns result of sense check S1 to S4
L_check	Returns result of sense check among literature sources
FMLXCheck	Returns sense check of overlap between FM and LX
FMLXCheck2	Returns sense check of overlap between FM and LX
OKcheck	Returns the the data quality classification
Warnings	Returns warning if GSD>GSDmax

3.5.1 VBA module "Selectfunctions"

Option Compare Database

```
'väljer minvärdet bland två tal
Public Function Min2Val(a, b)
If IsNumeric(a) And IsNumeric(b) Then
    If CDbl(a) > CDbl(b) Then
    Min2Val = CDbl(b)
    Else: Min2Val = CDbl(a)
    End If
ElseIf IsNumeric(a) And Not IsNumeric(b) Then
    Min2Val = CDbl(a)
ElseIf IsNumeric(b) And Not IsNumeric(a) Then
    Min2Val = CDbl(b)
Else: Min2Val = Null
End If
End Function

'väljer maxvärdet bland två tal
Public Function Max2Val(a, b) As Variant
If IsNumeric(a) And IsNumeric(b) Then
    If CDbl(a) > CDbl(b) Then
    Max2Val = a
    Else: Max2Val = b
    End If
ElseIf IsNumeric(a) And Not IsNumeric(b) Then
    Max2Val = a
ElseIf IsNumeric(b) And Not IsNumeric(a) Then
    Max2Val = b
Else: Max2Val = Null
End If
End Function

'väljer minvärdet bland fyra tal
Public Function Min4Val(a, b, c, d)
Min4Val = a
If IsNumeric(a) And IsNumeric(b) And a > b Then Min4Val = b
If Not IsNumeric(a) And IsNumeric(b) Then Min4Val = b
If IsNumeric(c) And IsNumeric(Min4Val) And Min4Val > c Then Min4Val = c
If Not IsNumeric(Min4Val) And IsNumeric(c) Then Min4Val = c
If IsNumeric(d) And IsNumeric(Min4Val) And Min4Val > d Then Min4Val = d
If Not IsNumeric(Min4Val) And IsNumeric(d) Then Min4Val = d
End Function

'väljer maxvärdet bland fyra tal
Public Function Max4Val(a, b, c, d)
Max4Val = a
If IsNumeric(a) And IsNumeric(b) And a < b Then Max4Val = b
If Not IsNumeric(a) And IsNumeric(b) Then Max4Val = b
If IsNumeric(c) And IsNumeric(Max4Val) And Max4Val < c Then Max4Val = c
If Not IsNumeric(Max4Val) And IsNumeric(c) Then Max4Val = c
If IsNumeric(d) And IsNumeric(Max4Val) And Max4Val < d Then Max4Val = d
If Not IsNumeric(Max4Val) And IsNumeric(d) Then Max4Val = d
End Function
```

```
'returnerar selected N
Public Function Sel_N(korrN_F_L, F_L_GM, F_L_GSD, GSDmin, GSDmax,
GSDmean, korrN_LX, F_L_n, L1_n, L2_n, L3_n, L4_n, KorrN_FM, override,
overrideGSD) As Variant

Select Case override
  Case "F_L": Sel_N = korrN_F_L
  Case "LX": Sel_N = korrN_LX
  Case "L1": Sel_N = L1_n
  Case "L2": Sel_N = L2_n
  Case "L3": Sel_N = L3_n
  Case "L4": Sel_N = L4_n
  Case "FM": Sel_N = KorrN_FM

  Case Else

  If IsNumeric(korrN_F_L) Then

    'Condition 1'
    If ((CInt(korrN_F_L) >= 10) And (Nz(F_L_GSD) > GSDmin) And (Nz(F_L_GSD) <
GSDmax)) Then
      Sel_N = CInt(korrN_F_L)
    End If

    'Condition 2'
    If CInt(korrN_F_L) >= 10 And Nz(F_L_GSD) < GSDmin And Nz(F_L_GSD) <
GSDmax Then
      Sel_N = CInt(korrN_F_L)
    End If

    'Condition 3'
    If CInt(korrN_F_L) >= 10 And Nz(F_L_GSD) >= GSDmax Then
      Sel_N = CInt(korrN_F_L)
    End If

    'Condition 4'
    If (CInt(korrN_F_L) > 3) And (CInt(korrN_F_L) < 10) And (Nz(F_L_GSD) >
GSDmean) And (Nz(F_L_GSD) < GSDmax) Then
      Sel_N = CInt(korrN_F_L)
    End If

    'Condition 5'
    If (CInt(korrN_F_L) > 3) And (CInt(korrN_F_L) < 10) And Nz(F_L_GSD) < GSDmean
And Nz(F_L_GSD) < GSDmax Then
      Sel_N = CInt(korrN_F_L)
    End If

    'Condition 6'
    If (CInt(korrN_F_L) > 3) And (CInt(korrN_F_L) < 10) And Nz(F_L_GSD) >=
GSDmax Then
      Sel_N = CInt(korrN_F_L)
```

```
        End If

        'Condition 7'
        If (CInt(korrN_F_L) > 0) And (CInt(korrN_F_L) <= 3) And (Nz(F_L_GSD) <
GSDmax) Then
            Sel_N = CInt(korrN_F_L)
        End If

        'Condition 8'
        If (CInt(korrN_F_L) > 0) And (CInt(korrN_F_L) <= 3) And Nz(F_L_GSD) >=
GSDmax Then
            Sel_N = CInt(korrN_F_L)
        End If
    End If

    'Condition 9'
    If IsNull(korrN_F_L) Then
        If IsNumeric(L1_n) Then
            Sel_N = L1_n
        ElseIf IsNumeric(L2_n) Then
            Sel_N = L2_n
        ElseIf IsNumeric(L3_n) Then
            Sel_N = L3_n
        ElseIf IsNumeric(L4_n) Then
            Sel_N = L4_n
        Else: Sel_N = Null
        End If
    End If

End Select
End Function

'Returnerar selected min
Public Function Sel_Min(korrN_F_L, F_L_GM, F_L_GSD, GSDmin, GSDmax,
GSDmean, LX_n, F_L_n, L1_n, L2_n, L3_n, L4_n, KorrN_FM, F_L_min, LX_min,
L1_min, L2_min, L3_min, L4_min, FM_min, LX_5p, L1_5p, L2_5p, L3_5p, L4_5p,
FM_5p, override, overrideGSD) As Variant

Dim F_L_5p, F_L_5p_GSDmin, F_L_5p_GSDmean As Variant

If overrideGSD = "GSDmax" Then F_L_GSD = GSDmax

F_L_5p = F_L_GM / F_L_GSD ^ 1.645
F_L_5p_GSDmean = F_L_GM / GSDmean ^ 1.645
F_L_5p_GSDmin = F_L_GM / GSDmin ^ 1.645
F_L_5p_GSDmax = F_L_GM / GSDmax ^ 1.645

Select Case override
    Case "F_L": Sel_Min = Min2Val(F_L_min, F_L_5p)
    Case "LX": Sel_Min = Min2Val(LX_min, LX_5p)
    Case "L1": Sel_Min = Min2Val(L1_min, L1_5p)
    Case "L2": Sel_Min = Min2Val(L2_min, L2_5p)
```

```
  Case "L3": Sel_Min = Min2Val(L3_min, L3_5p)
  Case "L4": Sel_Min = Min2Val(L4_min, L4_5p)
  Case "FM": Sel_Min = Min2Val(FM_min, FM_5p)

  Case Else

  If IsNumeric(korrN_F_L) Then

      'Condition 1'
      If ((CInt(korrN_F_L) >= 10) And (Nz(F_L_GSD) > GSDmin) And (Nz(F_L_GSD) <
GSDmax)) Then
          Sel_Min = Min2Val(F_L_min, F_L_5p)
      End If

      'Condition 2'
      If CInt(korrN_F_L) >= 10 And Nz(F_L_GSD) < GSDmin And Nz(F_L_GSD) <
GSDmax Then
          Sel_Min = Min2Val(F_L_min, F_L_5p_GSDmin)
      End If

      'Condition 3'
      If CInt(korrN_F_L) >= 10 And Nz(F_L_GSD) >= GSDmax Then
          Sel_Min = Min2Val(F_L_min, F_L_5p)
      End If

      'Condition 4'
      If (CInt(korrN_F_L) > 3) And (CInt(korrN_F_L) < 10) And (Nz(F_L_GSD) >
GSDmean) And (Nz(F_L_GSD) < GSDmax) Then
          Sel_Min = Min2Val(F_L_min, F_L_5p)
      End If

      'Condition 5'
      If (CInt(korrN_F_L) > 3) And (CInt(korrN_F_L) < 10) And Nz(F_L_GSD) < GSDmean
And Nz(F_L_GSD) < GSDmax Then
          Sel_Min = Min2Val(F_L_min, F_L_5p_GSDmean)
      End If

      'Condition 6'
      If (CInt(korrN_F_L) > 3) And (CInt(korrN_F_L) < 10) And Nz(F_L_GSD) >=
GSDmax Then
          Sel_Min = Min2Val(F_L_min, F_L_5p)
      End If

      'Condition 7'
      If (CInt(korrN_F_L) > 0) And (CInt(korrN_F_L) <= 3) And (Nz(F_L_GSD) <
GSDmax) Then
          Sel_Min = Min2Val(F_L_min, F_L_5p_GSDmax)
      End If

      'Condition 8'
      If (CInt(korrN_F_L) > 0) And (CInt(korrN_F_L) <= 3) And Nz(F_L_GSD) >=
GSDmax Then
```

```
            Sel_Min = Min2Val(F_L_min, F_L_5p)
        End If
    End If

    'Condition 9'
    If IsNull(korrN_F_L) Then
        If IsNumeric(L1_n) Then
            Sel_Min = Min2Val(L1_min, L1_5p)
        ElseIf IsNumeric(L2_n) Then
            Sel_Min = Min2Val(L2_min, L2_5p)
        ElseIf IsNumeric(L3_n) Then
            Sel_Min = Min2Val(L3_min, L3_5p)
        ElseIf IsNumeric(L4_n) Then
            Sel_Min = Min2Val(L4_min, L4_5p)
        Else: Sel_Min = Null
        End If
    End If
End Select
End Function

'Returnerar selected max
Public Function Sel_Max(korrN_F_L, F_L_GM, F_L_GSD, GSDmin, GSDmax,
GSDmean, LX_n, F_L_n, L1_n, L2_n, L3_n, L4_n, KorrN_FM, F_L_max, LX_max,
L1_max, L2_max, L3_max, L4_max, FM_max, LX_95p, L1_95p, L2_95p, L3_95p, L4_95p,
FM_95p, override, overrideGSD) As Variant

Dim F_L_95p, F_L_95p_GSDmin, F_L_95p_GSDmean As Variant

If overrideGSD = "GSDmax" Then F_L_GSD = GSDmax

F_L_95p = F_L_GM * F_L_GSD ^ 1.645
F_L_95p_GSDmean = F_L_GM * GSDmean ^ 1.645
F_L_95p_GSDmin = F_L_GM * GSDmin ^ 1.645
F_L_95p_GSDmax = F_L_GM * GSDmax ^ 1.645

Select Case override
    Case "F_L": Sel_Max = Max2Val(F_L_max, F_L_95p)
    Case "LX": Sel_Max = Max2Val(LX_max, LX_95p)
    Case "L1": Sel_Max = Max2Val(L1_max, L1_95p)
    Case "L2": Sel_Max = Max2Val(L2_max, L2_95p)
    Case "L3": Sel_Max = Max2Val(L3_max, L3_95p)
    Case "L4": Sel_Max = Max2Val(L4_max, L4_95p)
    Case "FM": Sel_Max = Max2Val(FM_max, FM_95p)

    Case Else

    If IsNumeric(korrN_F_L) Then

        'Condition 1'
        If ((CInt(korrN_F_L) >= 10) And (Nz(F_L_GSD) > GSDmin) And (Nz(F_L_GSD) <
GSDmax)) Then
            Sel_Max = Max2Val(F_L_max, F_L_95p)
```

```
        End If

        'Condition 2'
        If CInt(korrN_F_L) >= 10 And Nz(F_L_GSD) < GSDmin And Nz(F_L_GSD) <
GSDmax Then
            Sel_Max = Max2Val(F_L_max, F_L_95p_GSDmin)
        End If

        'Condition 3'
        If CInt(korrN_F_L) >= 10 And Nz(F_L_GSD) >= GSDmax Then
            Sel_Max = Max2Val(F_L_max, F_L_95p)
        End If

        'Condition 4'
        If (CInt(korrN_F_L) > 3) And (CInt(korrN_F_L) < 10) And (Nz(F_L_GSD) >
GSDmean) And (Nz(F_L_GSD) < GSDmax) Then
            Sel_Max = Max2Val(F_L_max, F_L_95p)
        End If

        'Condition 5'
        If (CInt(korrN_F_L) > 3) And (CInt(korrN_F_L) < 10) And Nz(F_L_GSD) < GSDmean
And Nz(F_L_GSD) < GSDmax Then
            Sel_Max = Max2Val(F_L_max, F_L_95p_GSDmean)
        End If

        'Condition 6'
        If (CInt(korrN_F_L) > 3) And (CInt(korrN_F_L) < 10) And Nz(F_L_GSD) >=
GSDmax Then
            Sel_Max = Max2Val(F_L_max, F_L_95p)
        End If

        'Condition 7'
        If (CInt(korrN_F_L) > 0) And (CInt(korrN_F_L) <= 3) And (Nz(F_L_GSD) <
GSDmax) Then
            Sel_Max = Max2Val(F_L_max, F_L_95p_GSDmax)
        End If

        'Condition 8'
        If (CInt(korrN_F_L) > 0) And (CInt(korrN_F_L) <= 3) And Nz(F_L_GSD) >=
GSDmax Then
            Sel_Max = Max2Val(F_L_max, F_L_95p)
        End If
    End If

    'Condition 9'
    If IsNull(korrN_F_L) Then
        If IsNumeric(L1_n) Then
            Sel_Max = Max2Val(L1_max, L1_95p)
        ElseIf IsNumeric(L2_n) Then
            Sel_Max = Max2Val(L2_max, L2_95p)
        ElseIf IsNumeric(L3_n) Then
            Sel_Max = Max2Val(L3_max, L3_95p)
```

```
        ElseIf IsNumeric(L4_n) Then
            Sel_Max = Max2Val(L4_max, L4_95p)
        Else: Sel_Max = Null
        End If
    End If
End Select
End Function
```

'Returnerar selected GM
Public Function Sel_GM(korrN_F_L, F_L_GM, F_L_GSD, GSDmin, GSDmax, GSDp, LX_n, F_L_n, L1_n, L2_n, L3_n, L4_n, KorrN_FM, LX_GM, L1_GM, L2_GM, L3_GM, L4_GM, FM_GM, override, overrideGSD) As Variant

```
Select Case override
    Case "F_L": Sel_GM = F_L_GM
    Case "LX": Sel_GM = LX_GM
    Case "L1": Sel_GM = L1_GM
    Case "L2": Sel_GM = L2_GM
    Case "L3": Sel_GM = L3_GM
    Case "L4": Sel_GM = L4_GM
    Case "FM": Sel_GM = FM_GM

    Case Else

    If IsNumeric(korrN_F_L) Then

        'Condition 1'
        If ((CInt(korrN_F_L) >= 10) And (Nz(F_L_GSD) > GSDmin) And (Nz(F_L_GSD) <
GSDmax)) Then
            Sel_GM = F_L_GM
        End If

        'Condition 2'
        If CInt(korrN_F_L) >= 10 And Nz(F_L_GSD) < GSDmin And Nz(F_L_GSD) <
GSDmax Then
            Sel_GM = F_L_GM
        End If

        'Condition 3'
        If CInt(korrN_F_L) >= 10 And Nz(F_L_GSD) >= GSDmax Then
            Sel_GM = F_L_GM
        End If

        'Condition 4'
        If (CInt(korrN_F_L) > 3) And (CInt(korrN_F_L) < 10) And (Nz(F_L_GSD) >
GSDmean) And (Nz(F_L_GSD) < GSDmax) Then
            Sel_GM = F_L_GM
        End If

        'Condition 5'
        If (CInt(korrN_F_L) > 3) And (CInt(korrN_F_L) < 10) And Nz(F_L_GSD) < GSDmean
And Nz(F_L_GSD) < GSDmax Then
```

```
            Sel_GM = F_L_GM
        End If

        'Condition 6'
        If (CInt(korrN_F_L) > 3) And (CInt(korrN_F_L) < 10) And Nz(F_L_GSD) >=
GSDmax Then
            Sel_GM = F_L_GM
        End If

        'Condition 7'
        If (CInt(korrN_F_L) > 0) And (CInt(korrN_F_L) <= 3) And (Nz(F_L_GSD) <
GSDmax) Then
            Sel_GM = F_L_GM
        End If

        'Condition 8'
        If (CInt(korrN_F_L) > 0) And (CInt(korrN_F_L) <= 3) And Nz(F_L_GSD) >=
GSDmax Then
            Sel_GM = F_L_GM
        End If
    End If

    'Condition 9'
    If IsNull(korrN_F_L) Then
        If IsNumeric(L1_n) Then
            Sel_GM = L1_GM
        ElseIf IsNumeric(L2_n) Then
            Sel_GM = L2_GM
        ElseIf IsNumeric(L3_n) Then
            Sel_GM = L3_GM
        ElseIf IsNumeric(L4_n) Then
            Sel_GM = L4_GM
        Else: Sel_GM = Null
        End If
    End If
End Select
End Function

'Returnerar selected BE
Public Function Sel_BE(korrN_F_L, F_L_GM, F_L_GSD, GSDmin, GSDmax, GSDp,
LX_n, F_L_n, L1_n, L2_n, L3_n, L4_n, KorrN_FM, LX_GM, L1_GM, L2_GM, L3_GM,
L4_GM, FM_GM, override, overrideGSD) As Variant

Select Case override
    Case "F_L": Sel_BE = F_L_GM
    Case "LX": Sel_BE = LX_GM
    Case "L1": Sel_BE = L1_GM
    Case "L2": Sel_BE = L2_GM
    Case "L3": Sel_BE = L3_GM
    Case "L4": Sel_BE = L4_GM
    Case "FM": Sel_BE = FM_GM
```

```
Case Else

If IsNumeric(korrN_F_L) Then

    'Condition 1'
    If ((CInt(korrN_F_L) >= 10) And (Nz(F_L_GSD) > GSDmin) And (Nz(F_L_GSD) <
GSDmax)) Then
        If IsNumeric(FM_GM) Then Sel_BE = FM_GM Else Sel_BE = F_L_GM
    End If

    'Condition 2'
    If CInt(korrN_F_L) >= 10 And Nz(F_L_GSD) < GSDmin And Nz(F_L_GSD) <
GSDmax Then
        If IsNumeric(FM_GM) Then Sel_BE = FM_GM Else Sel_BE = F_L_GM
    End If

    'Condition 3'
    If CInt(korrN_F_L) >= 10 And Nz(F_L_GSD) >= GSDmax Then
        If IsNumeric(FM_GM) Then Sel_BE = FM_GM Else Sel_BE = F_L_GM
    End If

    'Condition 4'
    If (CInt(korrN_F_L) > 3) And (CInt(korrN_F_L) < 10) And (Nz(F_L_GSD) >
GSDmean) And (Nz(F_L_GSD) < GSDmax) Then
        If IsNumeric(FM_GM) Then Sel_BE = FM_GM Else Sel_BE = F_L_GM
    End If

    'Condition 5'
    If (CInt(korrN_F_L) > 3) And (CInt(korrN_F_L) < 10) And Nz(F_L_GSD) < GSDmean
And Nz(F_L_GSD) < GSDmax Then
        If IsNumeric(FM_GM) Then Sel_BE = FM_GM Else Sel_BE = F_L_GM
    End If

    'Condition 6'
    If (CInt(korrN_F_L) > 3) And (CInt(korrN_F_L) < 10) And Nz(F_L_GSD) >=
GSDmax Then
        If IsNumeric(FM_GM) Then Sel_BE = FM_GM Else Sel_BE = F_L_GM
    End If

    'Condition 7'
    If (CInt(korrN_F_L) > 0) And (CInt(korrN_F_L) <= 3) And (Nz(F_L_GSD) <
GSDmax) Then
        If IsNumeric(FM_GM) Then Sel_BE = FM_GM Else Sel_BE = F_L_GM
    End If

    'Condition 8'
    If (CInt(korrN_F_L) > 0) And (CInt(korrN_F_L) <= 3) And Nz(F_L_GSD) >=
GSDmax Then
        If IsNumeric(FM_GM) Then Sel_BE = FM_GM Else Sel_BE = F_L_GM
    End If
End If
```

```
   'Condition 9'
   If IsNull(korrN_F_L) Then
      If IsNumeric(L1_n) Then
         Sel_BE = L1_GM
      ElseIf IsNumeric(L2_n) Then
         Sel_BE = L2_GM
      ElseIf IsNumeric(L3_n) Then
         Sel_BE = L3_GM
      ElseIf IsNumeric(L4_n) Then
         Sel_BE = L4_GM
      Else: Sel_BE = Null
      End If
   End If
End Select
End Function

'Returnerar selected BE referens
Public Function Sel_RefBE(korrN_F_L, F_L_GM, F_L_GSD, GSDmin, GSDmax,
GSDp, LX_n, F_L_n, L1_n, L2_n, L3_n, L4_n, KorrN_FM, LX_GM, L1_GM, L2_GM,
L3_GM, L4_GM, FM_GM, override, overrideGSD) As Variant

Select Case override
   Case "F_L": Sel_RefBE = "F_L_GM"
   Case "LX": Sel_RefBE = "LX_GM"
   Case "L1": Sel_RefBE = "L1_GM"
   Case "L2": Sel_RefBE = "L2_GM"
   Case "L3": Sel_RefBE = "L3_GM"
   Case "L4": Sel_RefBE = "L4_GM"
   Case "FM": Sel_RefBE = "FM_GM"

   Case Else

   If IsNumeric(korrN_F_L) Then

      'Condition 1'
      If ((CInt(korrN_F_L) >= 10) And (Nz(F_L_GSD) > GSDmin) And (Nz(F_L_GSD) <
GSDmax)) Then
         If IsNumeric(FM_GM) Then Sel_RefBE = "FM_GM" Else Sel_RefBE = "LX_GM"
      End If

      'Condition 2'
      If CInt(korrN_F_L) >= 10 And Nz(F_L_GSD) < GSDmin And Nz(F_L_GSD) <
GSDmax Then
         If IsNumeric(FM_GM) Then Sel_RefBE = "FM_GM" Else Sel_RefBE = "LX_GM"
      End If

      'Condition 3'
      If CInt(korrN_F_L) >= 10 And Nz(F_L_GSD) >= GSDmax Then
         If IsNumeric(FM_GM) Then Sel_RefBE = "FM_GM" Else Sel_RefBE = "LX_GM"
      End If

      'Condition 4'
```

```
        If (CInt(korrN_F_L) > 3) And (CInt(korrN_F_L) < 10) And (Nz(F_L_GSD) >
GSDmean) And (Nz(F_L_GSD) < GSDmax) Then
            If IsNumeric(FM_GM) Then Sel_RefBE = "FM_GM" Else Sel_RefBE = "LX_GM"
        End If

        'Condition 5'
        If (CInt(korrN_F_L) > 3) And (CInt(korrN_F_L) < 10) And Nz(F_L_GSD) < GSDmean
And Nz(F_L_GSD) < GSDmax Then
            If IsNumeric(FM_GM) Then Sel_RefBE = "FM_GM" Else Sel_RefBE = "LX_GM"
        End If

        'Condition 6'
        If (CInt(korrN_F_L) > 3) And (CInt(korrN_F_L) < 10) And Nz(F_L_GSD) >=
GSDmax Then
            If IsNumeric(FM_GM) Then Sel_RefBE = "FM_GM" Else Sel_RefBE = "LX_GM"
        End If

        'Condition 7'
        If (CInt(korrN_F_L) > 0) And (CInt(korrN_F_L) <= 3) And (Nz(F_L_GSD) <
GSDmax) Then
           If IsNumeric(FM_GM) Then Sel_RefBE = "FM_GM" Else Sel_RefBE = "LX_GM"
           End If

        'Condition 8'
        If (CInt(korrN_F_L) > 0) And (CInt(korrN_F_L) <= 3) And Nz(F_L_GSD) >=
GSDmax Then
            If IsNumeric(FM_GM) Then Sel_RefBE = "FM_GM" Else Sel_RefBE = "LX_GM"
        End If
    End If

    'Condition 9'
    If IsNull(korrN_F_L) Then
        If IsNumeric(L1_n) Then
            Sel_RefBE = "L1_GM"
        ElseIf IsNumeric(L2_n) Then
            Sel_RefBE = "L2_GM"
        ElseIf IsNumeric(L3_n) Then
            Sel_RefBE = "L3_GM"
        ElseIf IsNumeric(L4_n) Then
            Sel_RefBE = "L4_GM"
        Else: Sel_RefBE = Null
        End If
    End If
End Select
End Function

'Returnerar selected GSD
```

Public Function sel_gsd(korrN_F_L, F_L_GM, F_L_GSD, GSDmin, GSDmax, GSDmean, LX_n, F_L_n, L1_n, L2_n, L3_n, L4_n, KorrN_FM, LX_GSD, L1_GSD, L2_GSD, L3_GSD, L4_GSD, FM_GSD, override, overrideGSD) As Variant

```
Select Case override
```

```
Case "F_L": sel_gsd = F_L_GSD
Case "LX": sel_gsd = LX_GSD
Case "L1": sel_gsd = L1_GSD
Case "L2": sel_gsd = L2_GSD
Case "L3": sel_gsd = L3_GSD
Case "L4": sel_gsd = L4_GSD
Case "FM": sel_gsd = FM_GSD

Case Else

If IsNumeric(korrN_F_L) Then

    'Condition 1'
    If ((CInt(korrN_F_L) >= 10) And (Nz(F_L_GSD) > GSDmin) And (Nz(F_L_GSD) <
GSDmax)) Then
        sel_gsd = F_L_GSD
    End If

    'Condition 2'
    If CInt(korrN_F_L) >= 10 And Nz(F_L_GSD) < GSDmin And Nz(F_L_GSD) <
GSDmax Then
        sel_gsd = GSDmin
    End If

    'Condition 3'
    If CInt(korrN_F_L) >= 10 And Nz(F_L_GSD) >= GSDmax Then
        sel_gsd = F_L_GSD
    End If

    'Condition 4'
    If (CInt(korrN_F_L) > 3) And (CInt(korrN_F_L) < 10) And (Nz(F_L_GSD) >
GSDmean) And (Nz(F_L_GSD) < GSDmax) Then
        sel_gsd = F_L_GSD
    End If

    'Condition 5'
    If (CInt(korrN_F_L) > 3) And (CInt(korrN_F_L) < 10) And Nz(F_L_GSD) < GSDmean
And Nz(F_L_GSD) < GSDmax Then
        sel_gsd = GSDmean
    End If

    'Condition 6'
    If (CInt(korrN_F_L) > 3) And (CInt(korrN_F_L) < 10) And Nz(F_L_GSD) >=
GSDmax Then
        sel_gsd = F_L_GSD
    End If

    'Condition 7'
    If (CInt(korrN_F_L) > 0) And (CInt(korrN_F_L) <= 3) And (Nz(F_L_GSD) <
GSDmax) Then
        sel_gsd = GSDmax
    End If
```

```
        'Condition 8'
        If (CInt(korrN_F_L) > 0) And (CInt(korrN_F_L) <= 3) And Nz(F_L_GSD) >=
GSDmax Then
            sel_gsd = F_L_GSD
        End If
    End If

    'Condition 9'
    If IsNull(korrN_F_L) Then
        If IsNumeric(L1_n) Then
            If CDbl(L1_n) <= 3 Then
                sel_gsd = Max2Val(GSDmax, L1_GSD)
            Else: sel_gsd = Max2Val(GSDmean, L1_GSD)
            End If
        ElseIf IsNumeric(L2_n) Then
            If CDbl(L2_n) <= 3 Then
                sel_gsd = Max2Val(GSDmax, L2_GSD)
            Else: sel_gsd = Max2Val(GSDmean, L2_GSD)
            End If
        ElseIf IsNumeric(L3_n) Then
            If CDbl(L3_n) <= 3 Then
                sel_gsd = Max2Val(GSDmax, L3_GSD)
            Else: sel_gsd = Max2Val(GSDmean, L3_GSD)
            End If
        ElseIf IsNumeric(L4_n) Then
            If CDbl(L4_n) <= 3 Then
                sel_gsd = Max2Val(GSDmax, L4_GSD)
            Else: sel_gsd = Max2Val(GSDmean, L4_GSD)
            End If
        Else: sel_gsd = Null
        End If
    End If
End Select

If overrideGSD = "GSDmax" Then sel_gsd = Max2Val(GSDmax, sel_gsd)

End Function

'returnerar condition
Public Function Sel_Cond(korrN_F_L, F_L_GM, F_L_GSD, GSDmin, GSDmax,
GSDmean, LX_n, F_L_n, L1_n, L2_n, L3_n, L4_n, KorrN_FM, override, overrideGSD) As
Variant

If overrideGSD = "GSDmax" Then F_L_GSD = GSDmax

Select Case override
    Case "F_L": Sel_Cond = 8
    Case "LX": Sel_Cond = 8
    Case "L1": Sel_Cond = 8
    Case "L2": Sel_Cond = 8
    Case "L3": Sel_Cond = 8
```

```
    Case "L4": Sel_Cond = 8
    Case "FM": Sel_Cond = 8

    Case Else

    If IsNumeric(korrN_F_L) Then

    'Condition 1'
        If ((CInt(korrN_F_L) >= 10) And (Nz(F_L_GSD) > GSDmin) And (Nz(F_L_GSD) <
GSDmax)) Then
            Sel_Cond = 1
        End If

        'Condition 2'
        If CInt(korrN_F_L) >= 10 And Nz(F_L_GSD) < GSDmin And Nz(F_L_GSD) <
GSDmax Then
            Sel_Cond = 2
        End If

        'Condition 3'
        If CInt(korrN_F_L) >= 10 And Nz(F_L_GSD) >= GSDmax Then
            Sel_Cond = 3
        End If

        'Condition 4'
        If (CInt(korrN_F_L) > 3) And (CInt(korrN_F_L) < 10) And (Nz(F_L_GSD) >
GSDmean) And (Nz(F_L_GSD) < GSDmax) Then
            Sel_Cond = 4
        End If

        'Condition 5'
        If (CInt(korrN_F_L) > 3) And (CInt(korrN_F_L) < 10) And Nz(F_L_GSD) < GSDmean
And Nz(F_L_GSD) < GSDmax Then
            Sel_Cond = 5
        End If

        'Condition 6'
        If (CInt(korrN_F_L) > 3) And (CInt(korrN_F_L) < 10) And Nz(F_L_GSD) >=
GSDmax Then
            Sel_Cond = 6
        End If

        'Condition 7'
        If (CInt(korrN_F_L) > 0) And (CInt(korrN_F_L) <= 3) And (Nz(F_L_GSD) <
GSDmax) Then
            Sel_Cond = 7
        End If

        'Condition 8'
        If (CInt(korrN_F_L) > 0) And (CInt(korrN_F_L) <= 3) And Nz(F_L_GSD) >=
GSDmax Then
            Sel_Cond = 8
```

```
        End If
    End If

    'Condition 9'
    If IsNull(korrN_F_L) Then
        If IsNumeric(L1_n) Then
            Sel_Cond = 9
        ElseIf IsNumeric(L2_n) Then
            Sel_Cond = 9
        ElseIf IsNumeric(L3_n) Then
            Sel_Cond = 9
        ElseIf IsNumeric(L4_n) Then
            Sel_Cond = 9
        Else: Sel_Cond = Null
        End If
    End If
End Select
End Function
```

'Returnerar Ref_min
Public Function Sel_RefMin(korrN_F_L, F_L_GM, F_L_GSD, GSDmin, GSDmax, GSDmean, LX_n, F_L_n, L1_n, L2_n, L3_n, L4_n, KorrN_FM, F_L_min, LX_min, L1_min, L2_min, L3_min, L4_min, FM_min, LX_5p, L1_5p, L2_5p, L3_5p, L4_5p, FM_5p, override, overrideGSD) As Variant

```
Dim F_L_5p, F_L_5p_GSDmin, F_L_5p_GSDmean, F_L_5p_GSDmax As Variant

If overrideGSD = "GSDmax" Then F_L_GSD = GSDmax

F_L_5p = F_L_GM / F_L_GSD ^ 1.645
F_L_5p_GSDmean = F_L_GM / GSDmean ^ 1.645
F_L_5p_GSDmin = F_L_GM / GSDmin ^ 1.645
F_L_5p_GSDmax = F_L_GM / GSDmax ^ 1.645

Select Case override
    Case "F_L":
        If IsNumeric(F_L_min) And IsNumeric(F_L_5p) Then
            If F_L_min < F_L_5p Then
                Sel_RefMin = "min"
            Else: Sel_RefMin = "5p"
            End If
        ElseIf IsNumeric(F_L_min) Then
            Sel_RefMin = "min"
        ElseIf IsNumeric(F_L_5p) Then
            Sel_RefMin = "5p"
        Else: Sel_RefMin = Null
        End If
    Case "LX":
        If IsNumeric(LX_min) And IsNumeric(LX_5p) Then
            If LX_min < LX_5p Then
                Sel_RefMin = "min"
            Else: Sel_RefMin = "5p"
            End If
```

```
    ElseIf IsNumeric(LX_min) Then
      Sel_RefMin = "min"
    ElseIf IsNumeric(LX_5p) Then
      Sel_RefMin = "5p"
    Else: Sel_RefMin = Null
    End If
  Case "L1":
   If IsNumeric(L1_min) And IsNumeric(L1_5p) Then
      If L1_min < L1_5p Then
        Sel_RefMin = "min"
      Else: Sel_RefMin = "5p"
      End If
    ElseIf IsNumeric(L1_min) Then
      Sel_RefMin = "min"
    ElseIf IsNumeric(L1_5p) Then
      Sel_RefMin = "5p"
    Else: Sel_RefMin = Null
    End If
  Case "L2":
   If IsNumeric(L2_min) And IsNumeric(L2_5p) Then
      If L2_min < L2_5p Then
        Sel_RefMin = "min"
      Else: Sel_RefMin = "5p"
      End If
    ElseIf IsNumeric(L2_min) Then
      Sel_RefMin = "min"
    ElseIf IsNumeric(L2_5p) Then
      Sel_RefMin = "5p"
    Else: Sel_RefMin = Null
    End If
  Case "L3":
   If IsNumeric(L3_min) And IsNumeric(L3_5p) Then
      If L3_min < L3_5p Then
        Sel_RefMin = "min"
      Else: Sel_RefMin = "5p"
      End If
    ElseIf IsNumeric(L3_min) Then
      Sel_RefMin = "min"
    ElseIf IsNumeric(L3_5p) Then
      Sel_RefMin = "5p"
    Else: Sel_RefMin = Null
    End If
  Case "L4":
   If IsNumeric(L4_min) And IsNumeric(L4_5p) Then
      If L4_min < L4_5p Then
        Sel_RefMin = "min"
      Else: Sel_RefMin = "5p"
      End If
    ElseIf IsNumeric(L4_min) Then
      Sel_RefMin = "min"
    ElseIf IsNumeric(L4_5p) Then
      Sel_RefMin = "5p"
```

```
        Else: Sel_RefMin = Null
        End If
    Case "FM":
      If IsNumeric(FM_min) And IsNumeric(FM_5p) Then
            If FM_min < FM_5p Then
                Sel_RefMin = "min"
            Else: Sel_RefMin = "5p"
            End If
        ElseIf IsNumeric(FM_min) Then
            Sel_RefMin = "min"
        ElseIf IsNumeric(FM_5p) Then
            Sel_RefMin = "5p"
        Else: Sel_RefMin = Null
        End If
    Case Else

    If IsNumeric(korrN_F_L) Then

     'Condition 1'
        If ((CInt(korrN_F_L) >= 10) And (Nz(F_L_GSD) > GSDmin) And (Nz(F_L_GSD) <
GSDmax)) Then
            If F_L_min < F_L_5p Then Sel_RefMin = "min" Else Sel_RefMin = "5p"
        End If

        'Condition 2'
        If CInt(korrN_F_L) >= 10 And Nz(F_L_GSD) < GSDmin And Nz(F_L_GSD) <
GSDmax Then
            If F_L_min < F_L_5p_GSDmin Then Sel_RefMin = "min" Else Sel_RefMin = "5p"
        End If

        'Condition 3'
        If CInt(korrN_F_L) >= 10 And Nz(F_L_GSD) >= GSDmax Then
            If F_L_min < F_L_5p Then Sel_RefMin = "min" Else Sel_RefMin = "5p"
        End If

        'Condition 4'
        If (CInt(korrN_F_L) > 3) And (CInt(korrN_F_L) < 10) And (Nz(F_L_GSD) >
GSDmean) And (Nz(F_L_GSD) < GSDmax) Then
            If F_L_min < F_L_5p Then Sel_RefMin = "min" Else Sel_RefMin = "5p"
        End If

        'Condition 5'
        If (CInt(korrN_F_L) > 3) And (CInt(korrN_F_L) < 10) And Nz(F_L_GSD) < GSDmean
And Nz(F_L_GSD) < GSDmax Then
            If F_L_min < F_L_5p_GSDmean Then Sel_RefMin = "min" Else Sel_RefMin = "5p"
        End If

        'Condition 6'
        If (CInt(korrN_F_L) > 3) And (CInt(korrN_F_L) < 10) And Nz(F_L_GSD) >=
GSDmax Then
            If F_L_min < F_L_5p Then Sel_RefMin = "min" Else Sel_RefMin = "5p"
        End If
```

```
        'Condition 7'
        If (CInt(korrN_F_L) > 0) And (CInt(korrN_F_L) <= 3) And (Nz(F_L_GSD) <
GSDmax) Then
            If F_L_min < F_L_5p_GSDmax Then Sel_RefMin = "min" Else Sel_RefMin = "5p"
        End If

        'Condition 8'
        If (CInt(korrN_F_L) > 0) And (CInt(korrN_F_L) <= 3) And Nz(F_L_GSD) >=
GSDmax Then
            If F_L_min < F_L_5p Then Sel_RefMin = "min" Else Sel_RefMin = "5p"
        End If
    End If

    'Condition 9'
    If IsNull(korrN_F_L) Then
        If IsNumeric(L1_n) Then
            If L1_min < L1_5p Then Sel_RefMin = "min" Else Sel_RefMin = "5p"
            'Sel_RefMin = Min2Val(L1_min, L1_5p)
        ElseIf IsNumeric(L2_n) Then
            If L2_min < L2_5p Then Sel_RefMin = "min" Else Sel_RefMin = "5p"
            'Sel_RefMin = Min2Val(L2_min, L2_5p)
        ElseIf IsNumeric(L3_n) Then
            If L3_min < L3_5p Then Sel_RefMin = "min" Else Sel_RefMin = "5p"
            'Sel_RefMin = Min2Val(L3_min, L3_5p)
        ElseIf IsNumeric(L4_n) Then
            If L4_min < L4_5p Then Sel_RefMin = "min" Else Sel_RefMin = "5p"
            'Sel_RefMin = Min2Val(L4_min, L4_5p)
        Else: Sel_RefMin = Null
        End If
    End If
End Select
End Function

'Returnerar Ref_max
Public Function Sel_RefMax(korrN_F_L, F_L_GM, F_L_GSD, GSDmin, GSDmax,
GSDmean, LX_n, F_L_n, L1_n, L2_n, L3_n, L4_n, KorrN_FM, F_L_max, LX_max,
L1_max, L2_max, L3_max, L4_max, FM_max, LX_95p, L1_95p, L2_95p, L3_95p, L4_95p,
FM_95p, override, overrideGSD) As Variant
Dim F_L_95p, F_L_95p_GSDp, F_L_95p_GSDmin, F_L_95p_GSDmax As Variant

If overrideGSD = "GSDmax" Then F_L_GSD = GSDmax

F_L_95p = F_L_GM * F_L_GSD ^ 1.645
F_L_95p_GSDmean = F_L_GM * GSDmean ^ 1.645
F_L_95p_GSDmin = F_L_GM * GSDmin ^ 1.645
F_L_95p_GSDmax = F_L_GM * GSDmax ^ 1.645

Select Case override
    Case "F_L":
        If IsNumeric(F_L_max) And IsNumeric(F_L_95p) Then
            If F_L_max > F_L_95p Then
```

```
            Sel_RefMax = "max"
        Else: Sel_RefMax = "95p"
        End If
    ElseIf IsNumeric(F_L_max) Then
        Sel_RefMax = "max"
    ElseIf IsNumeric(F_L_95p) Then
        Sel_RefMax = "95p"
    Else: Sel_RefMax = Null
    End If
Case "LX":
    If IsNumeric(LX_max) And IsNumeric(LX_95p) Then
        If LX_max > LX_95p Then
            Sel_RefMax = "max"
        Else: Sel_RefMax = "95p"
        End If
    ElseIf IsNumeric(LX_max) Then
        Sel_RefMax = "max"
    ElseIf IsNumeric(LX_95p) Then
        Sel_RefMax = "95p"
    Else: Sel_RefMax = Null
    End If
Case "L1":
  If IsNumeric(L1_max) And IsNumeric(L1_95p) Then
        If L1_max > L1_95p Then
            Sel_RefMax = "max"
        Else: Sel_RefMax = "95p"
        End If
    ElseIf IsNumeric(L1_max) Then
        Sel_RefMax = "max"
    ElseIf IsNumeric(L1_95p) Then
        Sel_RefMax = "95p"
    Else: Sel_RefMax = Null
    End If
Case "L2":
  If IsNumeric(L2_max) And IsNumeric(L2_95p) Then
        If L2_max > L2_95p Then
            Sel_RefMax = "max"
        Else: Sel_RefMax = "95p"
        End If
    ElseIf IsNumeric(L2_max) Then
        Sel_RefMax = "max"
    ElseIf IsNumeric(L2_95p) Then
        Sel_RefMax = "95p"
    Else: Sel_RefMax = Null
    End If
Case "L3":
  If IsNumeric(L3_max) And IsNumeric(L3_95p) Then
        If L3_max > L3_95p Then
            Sel_RefMax = "max"
        Else: Sel_RefMax = "95p"
        End If
    ElseIf IsNumeric(L3_max) Then
```

```
            Sel_RefMax = "max"
        ElseIf IsNumeric(L3_95p) Then
            Sel_RefMax = "95p"
        Else: Sel_RefMax = Null
        End If
    Case "L4":
      If IsNumeric(L4_max) And IsNumeric(L4_95p) Then
            If L4_max > L4_95p Then
                Sel_RefMax = "max"
            Else: Sel_RefMax = "95p"
            End If
        ElseIf IsNumeric(L4_max) Then
            Sel_RefMax = "max"
        ElseIf IsNumeric(L4_95p) Then
            Sel_RefMax = "95p"
        Else: Sel_RefMax = Null
        End If
    Case "FM":
      If IsNumeric(FM_max) And IsNumeric(FM_95p) Then
            If FM_max > FM_95p Then
                Sel_RefMax = "max"
            Else: Sel_RefMax = "95p"
            End If
        ElseIf IsNumeric(FM_max) Then
            Sel_RefMax = "max"
        ElseIf IsNumeric(FM_95p) Then
            Sel_RefMax = "95p"
        Else: Sel_RefMax = Null
        End If
    Case Else

    If IsNumeric(korrN_F_L) Then

        'Condition 1'
        If ((CInt(korrN_F_L) >= 10) And (Nz(F_L_GSD) > GSDmin) And (Nz(F_L_GSD) <
GSDmax)) Then
            If F_L_max > F_L_95p Then Sel_RefMax = "max" Else Sel_RefMax = "95p"
        End If

        'Condition 2'
        If CInt(korrN_F_L) >= 10 And Nz(F_L_GSD) < GSDmin And Nz(F_L_GSD) <
GSDmax Then
            If F_L_max > F_L_95p_GSDmin Then Sel_RefMax = "max" Else Sel_RefMax =
"95p"
        End If

        'Condition 3'
        If CInt(korrN_F_L) >= 10 And Nz(F_L_GSD) >= GSDmax Then
            If F_L_max > F_L_95p Then Sel_RefMax = "max" Else Sel_RefMax = "95p"
        End If

        'Condition 4'
```

```
        If (CInt(korrN_F_L) > 3) And (CInt(korrN_F_L) < 10) And (Nz(F_L_GSD) >
GSDmean) And (Nz(F_L_GSD) < GSDmax) Then
            If F_L_max > F_L_95p Then Sel_RefMax = "max" Else Sel_RefMax = "95p"
        End If

        'Condition 5'
        If (CInt(korrN_F_L) > 3) And (CInt(korrN_F_L) < 10) And Nz(F_L_GSD) < GSDmean
And Nz(F_L_GSD) < GSDmax Then
            If F_L_max > F_L_95p_GSDmean Then Sel_RefMax = "max" Else Sel_RefMax =
"95p"
        End If

        'Condition 6'
        If (CInt(korrN_F_L) > 3) And (CInt(korrN_F_L) < 10) And Nz(F_L_GSD) >=
GSDmax Then
            If F_L_max > F_L_95p Then Sel_RefMax = "max" Else Sel_RefMax = "95p"
        End If

        'Condition 7'
        If (CInt(korrN_F_L) > 0) And (CInt(korrN_F_L) <= 3) And (Nz(F_L_GSD) <
GSDmax) Then
            If F_L_max > F_L_95p_GSDmax Then Sel_RefMax = "max" Else Sel_RefMax =
"95p"
        End If

        'Condition 8'
        If (CInt(korrN_F_L) > 0) And (CInt(korrN_F_L) <= 3) And Nz(F_L_GSD) >=
GSDmax Then
            If F_L_max > F_L_95p Then Sel_RefMax = "max" Else Sel_RefMax = "95p"
        End If
    End If

    'Condition 9'
    If IsNull(korrN_F_L) Then
        If IsNumeric(L1_n) Then
            If L1_max > L1_95p Then Sel_RefMax = "max" Else Sel_RefMax = "95p"
            'Sel_RefMax = Max2Val(L1_max, L1_95p)
        ElseIf IsNumeric(L2_n) Then
            If L2_max > L2_95p Then Sel_RefMax = "max" Else Sel_RefMax = "95p"
            'Sel_RefMax = Max2Val(L2_max, L2_95p)
        ElseIf IsNumeric(L3_n) Then
            If L3_max > L3_95p Then Sel_RefMax = "max" Else Sel_RefMax = "95p"
            'Sel_RefMax = Max2Val(L3_max, L3_95p)
        ElseIf IsNumeric(L4_n) Then
            If L4_max > L4_95p Then Sel_RefMax = "max" Else Sel_RefMax = "95p"
            'Sel_RefMax = Max2Val(L4_max, L4_95p)
        Else: Sel_RefMax = Null
        End If
    End If
End Select
End Function
```

```
'Returnerar selected RefGSD
Public Function Sel_RefGSD(korrN_F_L, F_L_GM, F_L_GSD, GSDmin, GSDmax,
GSDmean, LX_n, F_L_n, L1_n, L2_n, L3_n, L4_n, KorrN_FM, LX_GSD, L1_GSD,
L2_GSD, L3_GSD, L4_GSD, FM_GSD, override, overrideGSD) As Variant
Dim sel_gsd_loc As Variant

Select Case override
   Case "F_L": Sel_RefGSD = "gsd"
   Case "LX": Sel_RefGSD = "gsd"
   Case "L1": Sel_RefGSD = "gsd"
   Case "L2": Sel_RefGSD = "gsd"
   Case "L3": Sel_RefGSD = "gsd"
   Case "L4": Sel_RefGSD = "gsd"
   Case "FM": Sel_RefGSD = "gsd"

   Case Else

   If IsNumeric(korrN_F_L) Then

      'Condition 1'
      If ((CInt(korrN_F_L) >= 10) And (Nz(F_L_GSD) > GSDmin) And (Nz(F_L_GSD) <
GSDmax)) Then
         Sel_RefGSD = "GSD"
      End If

      'Condition 2'
      If CInt(korrN_F_L) >= 10 And Nz(F_L_GSD) < GSDmin And Nz(F_L_GSD) <
GSDmax Then
         Sel_RefGSD = "GSDmin"
      End If

      'Condition 3'
      If CInt(korrN_F_L) >= 10 And Nz(F_L_GSD) >= GSDmax Then
         Sel_RefGSD = "GSD"
      End If

      'Condition 4'
      If (CInt(korrN_F_L) > 3) And (CInt(korrN_F_L) < 10) And (Nz(F_L_GSD) >
GSDmean) And (Nz(F_L_GSD) < GSDmax) Then
         Sel_RefGSD = "GSD"
      End If

      'Condition 5'
      If (CInt(korrN_F_L) > 3) And (CInt(korrN_F_L) < 10) And Nz(F_L_GSD) < GSDmean
And Nz(F_L_GSD) < GSDmax Then
         Sel_RefGSD = "GSDmean"
      End If

      'Condition 6'
      If (CInt(korrN_F_L) > 3) And (CInt(korrN_F_L) < 10) And Nz(F_L_GSD) >=
GSDmax Then
         Sel_RefGSD = "GSD"
```

```
        End If

        'Condition 7'
        If (CInt(korrN_F_L) > 0) And (CInt(korrN_F_L) <= 3) And (Nz(F_L_GSD) <
GSDmax) Then
            Sel_RefGSD = "GSDmax"
        End If

        'Condition 8'
        If (CInt(korrN_F_L) > 0) And (CInt(korrN_F_L) <= 3) And Nz(F_L_GSD) >=
GSDmax Then
            Sel_RefGSD = "GSD"
        End If
    End If

    'Condition 9'
    If IsNull(korrN_F_L) Then
        If IsNumeric(L1_n) Then
            If CDbl(L1_n) <= 3 Then
                If GSDmax < L1_GSD Then Sel_RefGSD = "GSD" Else Sel_RefGSD =
"GSDmax"
            Else: If GSDmean < L1_GSD Then Sel_RefGSD = "GSD" Else Sel_RefGSD =
"GSDmean"
            End If
        ElseIf IsNumeric(L2_n) Then
            If CDbl(L2_n) <= 3 Then
                If GSDmax < L2_GSD Then Sel_RefGSD = "GSD" Else Sel_RefGSD =
"GSDmax"
            Else: If GSDmean < L2_GSD Then Sel_RefGSD = "GSD" Else Sel_RefGSD =
"GSDmean"
            End If
        ElseIf IsNumeric(L3_n) Then
            If CDbl(L3_n) <= 3 Then
                If GSDmax < L3_GSD Then Sel_RefGSD = "GSD" Else Sel_RefGSD =
"GSDmax"
            Else: If GSDmean < L3_GSD Then Sel_RefGSD = "GSD" Else Sel_RefGSD =
"GSDmean"
            End If
        ElseIf IsNumeric(L4_n) Then
            If CDbl(L4_n) <= 3 Then
                If GSDmax < L4_GSD Then Sel_RefGSD = "GSD" Else Sel_RefGSD =
"GSDmax"
            Else: If GSDmean < L4_GSD Then Sel_RefGSD = "GSD" Else Sel_RefGSD =
"GSDmean"
            End If
        Else: Sel_RefGSD = Null
        End If
    End If
End Select

If overrideGSD = "GSDmax" Then Sel_RefGSD = "GSDmax"
```

End Function

'returnerar referens

Public Function Sel_Ref(korrN_F_L, F_L_GM, F_L_GSD, GSDmin, GSDmax, GSDmean, LX_n, L1_n, L2_n, L3_n, L4_n, KorrN_FM, F_L_ref, LX_ref, L1_ref, L2_ref, L3_ref, L4_ref, FM_ref, override, overrideGSD) As Variant

```
Select Case override
    Case "F_L": Sel_Ref = F_L_ref
    Case "LX": Sel_Ref = LX_ref
    Case "L1": Sel_Ref = L1_ref
    Case "L2": Sel_Ref = L2_ref
    Case "L3": Sel_Ref = L3_ref
    Case "L4": Sel_Ref = L4_ref
    Case "FM": Sel_Ref = FM_ref

    Case Else

    If IsNumeric(korrN_F_L) Then

     'Condition 1'
        If ((CInt(korrN_F_L) >= 10) And (Nz(F_L_GSD) > GSDmin) And (Nz(F_L_GSD) <
GSDmax)) Then
            If IsNumeric(KorrN_FM) And IsNull(LX_n) Then
                Sel_Ref = FM_ref
            ElseIf IsNull(KorrN_FM) And IsNumeric(LX_n) Then Sel_Ref = LX_ref
            Else: Sel_Ref = F_L_ref
            End If
        End If

        'Condition 2'
        If CInt(korrN_F_L) >= 10 And Nz(F_L_GSD) < GSDmin And Nz(F_L_GSD) <
GSDmax Then
            If IsNumeric(KorrN_FM) And IsNull(LX_n) Then
                Sel_Ref = FM_ref
            ElseIf IsNull(KorrN_FM) And IsNumeric(LX_n) Then Sel_Ref = LX_ref
            Else: Sel_Ref = F_L_ref
            End If
        End If

        'Condition 3'
        If CInt(korrN_F_L) >= 10 And Nz(F_L_GSD) >= GSDmax Then
            If IsNumeric(KorrN_FM) And IsNull(LX_n) Then
                Sel_Ref = FM_ref
            ElseIf IsNull(KorrN_FM) And IsNumeric(LX_n) Then Sel_Ref = LX_ref
            Else: Sel_Ref = F_L_ref
            End If
        End If

        'Condition 4'
        If (CInt(korrN_F_L) > 3) And (CInt(korrN_F_L) < 10) And (Nz(F_L_GSD) >
GSDmean) And (Nz(F_L_GSD) < GSDmax) Then
```

```
            If IsNumeric(KorrN_FM) And IsNull(LX_n) Then
                Sel_Ref = FM_ref
            ElseIf IsNull(KorrN_FM) And IsNumeric(LX_n) Then Sel_Ref = LX_ref
            Else: Sel_Ref = F_L_ref
            End If
        End If

        'Condition 5'
        If (CInt(korrN_F_L) > 3) And (CInt(korrN_F_L) < 10) And Nz(F_L_GSD) < GSDmean
And Nz(F_L_GSD) < GSDmax Then
            If IsNumeric(KorrN_FM) And IsNull(LX_n) Then
                Sel_Ref = FM_ref
            ElseIf IsNull(KorrN_FM) And IsNumeric(LX_n) Then Sel_Ref = LX_ref
            Else: Sel_Ref = F_L_ref
            End If
        End If

        'Condition 6'
        If (CInt(korrN_F_L) > 3) And (CInt(korrN_F_L) < 10) And Nz(F_L_GSD) >=
GSDmax Then
            If IsNumeric(KorrN_FM) And IsNull(LX_n) Then
                Sel_Ref = FM_ref
            ElseIf IsNull(KorrN_FM) And IsNumeric(LX_n) Then Sel_Ref = LX_ref
            Else: Sel_Ref = F_L_ref
            End If
        End If

        'Condition 7'
        If (CInt(korrN_F_L) > 0) And (CInt(korrN_F_L) <= 3) And (Nz(F_L_GSD) <
GSDmax) Then
            If IsNumeric(KorrN_FM) And IsNull(LX_n) Then
                Sel_Ref = FM_ref
            ElseIf IsNull(KorrN_FM) And IsNumeric(LX_n) Then Sel_Ref = LX_ref
            Else: Sel_Ref = F_L_ref
            End If
        End If

        'Condition 8'
        If (CInt(korrN_F_L) > 0) And (CInt(korrN_F_L) <= 3) And Nz(F_L_GSD) >=
GSDmax Then
            If IsNumeric(KorrN_FM) And IsNull(LX_n) Then
                Sel_Ref = FM_ref
            ElseIf IsNull(KorrN_FM) And IsNumeric(LX_n) Then Sel_Ref = LX_ref
            Else: Sel_Ref = F_L_ref
            End If
        End If
    End If

    'Condition 9'
    If IsNull(korrN_F_L) Then
        If IsNumeric(L1_n) Then
            Sel_Ref = L1_ref
```

```
        ElseIf IsNumeric(L2_n) Then
            Sel_Ref = L2_ref
        ElseIf IsNumeric(L3_n) Then
            Sel_Ref = L3_ref
        ElseIf IsNumeric(L4_n) Then
            Sel_Ref = L4_ref
        Else: Sel_Ref = Null
        End If
    End If
End Select
End Function

'returnerar selected MediaType
Public Function Sel_MT(korrN_F_L, F_L_GM, F_L_GSD, GSDmin, GSDmax,
GSDmean, LX_n, L1_n, L2_n, L3_n, L4_n, KorrN_FM, FM_x, LX_x, F_L_x, L1_x, L2_x,
L3_x, L4_x, override) As Variant

Select Case override
    Case "F_L": Sel_MT = F_L_x
    Case "LX": Sel_MT = LX_x
    Case "L1": Sel_MT = L1_x
    Case "L2": Sel_MT = L2_x
    Case "L3": Sel_MT = L3_x
    Case "L4": Sel_MT = L4_x
    Case "FM": Sel_MT = FM_x

    Case Else

    If IsNumeric(korrN_F_L) Then

        'Condition 1'
        If ((CInt(korrN_F_L) >= 10) And (Nz(F_L_GSD) > GSDmin) And (Nz(F_L_GSD) <
GSDmax)) Then
            Sel_MT = F_L_x
        End If

        'Condition 2'
        If CInt(korrN_F_L) >= 10 And Nz(F_L_GSD) < GSDmin And Nz(F_L_GSD) <
GSDmax Then
            Sel_MT = F_L_x
        End If

        'Condition 3'
        If CInt(korrN_F_L) >= 10 And Nz(F_L_GSD) >= GSDmax Then
            Sel_MT = F_L_x
        End If

        'Condition 4'
        If (CInt(korrN_F_L) > 3) And (CInt(korrN_F_L) < 10) And (Nz(F_L_GSD) >
GSDmean) And (Nz(F_L_GSD) < GSDmax) Then
            Sel_MT = F_L_x
        End If
```

```
        'Condition 5'
        If (CInt(korrN_F_L) > 3) And (CInt(korrN_F_L) < 10) And Nz(F_L_GSD) < GSDmean
And Nz(F_L_GSD) < GSDmax Then
            Sel_MT = F_L_x
        End If

        'Condition 6'
        If (CInt(korrN_F_L) > 3) And (CInt(korrN_F_L) < 10) And Nz(F_L_GSD) >=
GSDmax Then
            Sel_MT = F_L_x
        End If

        'Condition 7'
        If (CInt(korrN_F_L) > 0) And (CInt(korrN_F_L) <= 3) And (Nz(F_L_GSD) <
GSDmax) Then
            Sel_MT = F_L_x
        End If

        'Condition 8'
        If (CInt(korrN_F_L) > 0) And (CInt(korrN_F_L) <= 3) And Nz(F_L_GSD) >=
GSDmax Then
            Sel_MT = F_L_x
        End If
    End If

    'Condition 9'
    If IsNull(korrN_F_L) Then
        If IsNumeric(L1_n) Then
            Sel_MT = L1_x
        ElseIf IsNumeric(L2_n) Then
            Sel_MT = L2_x
        ElseIf IsNumeric(L3_n) Then
            Sel_MT = L3_x
        ElseIf IsNumeric(L4_n) Then
            Sel_MT = L4_x
        Else: Sel_MT = Null
        End If
    End If
End Select
End Function

'returnerar selected MediaTypeDetail
Public Function Sel_MTD(korrN_F_L, F_L_GM, F_L_GSD, GSDmin, GSDmax,
GSDp, LX_n, L1_n, L2_n, L3_n, L4_n, KorrN_FM, FM_x, LX_x, F_L_x, L1_x, L2_x,
L3_x, L4_x, override) As Variant

Select Case override
    Case "F_L": Sel_MTD = F_L_x
    Case "LX": Sel_MTD = LX_x
    Case "L1": Sel_MTD = L1_x
    Case "L2": Sel_MTD = L2_x
```

```
    Case "L3": Sel_MTD = L3_x
    Case "L4": Sel_MTD = L4_x
    Case "FM": Sel_MTD = FM_x

    Case Else

        'Condition 1'
        If ((CInt(korrN_F_L) >= 10) And (Nz(F_L_GSD) > GSDmin) And (Nz(F_L_GSD) <
GSDmax)) Then
            Sel_MTDD = F_L_x
        End If

        'Condition 2'
        If CInt(korrN_F_L) >= 10 And Nz(F_L_GSD) < GSDmin And Nz(F_L_GSD) <
GSDmax Then
            Sel_MTD = F_L_x
        End If

        'Condition 3'
        If CInt(korrN_F_L) >= 10 And Nz(F_L_GSD) >= GSDmax Then
            Sel_MTD = F_L_x
        End If

        'Condition 4'
        If (CInt(korrN_F_L) > 3) And (CInt(korrN_F_L) < 10) And (Nz(F_L_GSD) >
GSDmean) And (Nz(F_L_GSD) < GSDmax) Then
            Sel_MTD = F_L_x
        End If

        'Condition 5'
        If (CInt(korrN_F_L) > 3) And (CInt(korrN_F_L) < 10) And Nz(F_L_GSD) < GSDmean
And Nz(F_L_GSD) < GSDmax Then
            Sel_MTD = F_L_x
        End If

        'Condition 6'
        If (CInt(korrN_F_L) > 3) And (CInt(korrN_F_L) < 10) And Nz(F_L_GSD) >=
GSDmax Then
            Sel_MTD = F_L_x
        End If

        'Condition 7'
        If (CInt(korrN_F_L) > 0) And (CInt(korrN_F_L) <= 3) And (Nz(F_L_GSD) <
GSDmax) Then
            Sel_MTD = F_L_x
        End If

        'Condition 8'
        If (CInt(korrN_F_L) > 0) And (CInt(korrN_F_L) <= 3) And Nz(F_L_GSD) >=
GSDmax Then
            Sel_MTD = F_L_x
        End If
```

```
    End If

    'Condition 9'
    If IsNull(korrN_F_L) Then
        If IsNumeric(L1_n) Then
            Sel_MTD = L1_x
        ElseIf IsNumeric(L2_n) Then
            Sel_MTD = L2_x
        ElseIf IsNumeric(L3_n) Then
            Sel_MTD = L3_x
        ElseIf IsNumeric(L4_n) Then
            Sel_MTD = L4_x
        Else: Sel_MTD = Null
        End If
    End If
End Select
End Function

'returnerar selected BiotaType
Public Function Sel_BT(korrN_F_L, F_L_GM, F_L_GSD, GSDmin, GSDmax, GSDp,
LX_n, L1_n, L2_n, L3_n, L4_n, KorrN_FM, FM_x, LX_x, F_L_x, L1_x, L2_x, L3_x, L4_x,
override) As Variant

Select Case override
    Case "F_L": Sel_BT = F_L_x
    Case "LX": Sel_BT = LX_x
    Case "L1": Sel_BT = L1_x
    Case "L2": Sel_BT = L2_x
    Case "L3": Sel_BT = L3_x
    Case "L4": Sel_BT = L4_x
    Case "FM": Sel_BT = FM_x

    Case Else

    If IsNumeric(korrN_F_L) Then
        'Condition 1'
        If ((CInt(korrN_F_L) >= 10) And (Nz(F_L_GSD) > GSDmin) And (Nz(F_L_GSD) <
GSDmax)) Then
            Sel_BT = F_L_x
        End If

        'Condition 2'
        If CInt(korrN_F_L) >= 10 And Nz(F_L_GSD) < GSDmin And Nz(F_L_GSD) <
GSDmax Then
            Sel_BT = F_L_x
        End If

        'Condition 3'
        If CInt(korrN_F_L) >= 10 And Nz(F_L_GSD) >= GSDmax Then
            Sel_BT = F_L_x
        End If
```

```
        'Condition 4'
        If (CInt(korrN_F_L) > 3) And (CInt(korrN_F_L) < 10) And (Nz(F_L_GSD) >
GSDmean) And (Nz(F_L_GSD) < GSDmax) Then
            Sel_BT = F_L_x
        End If

        'Condition 5'
        If (CInt(korrN_F_L) > 3) And (CInt(korrN_F_L) < 10) And Nz(F_L_GSD) < GSDmean
And Nz(F_L_GSD) < GSDmax Then
            Sel_BT = F_L_x
        End If

        'Condition 6'
        If (CInt(korrN_F_L) > 3) And (CInt(korrN_F_L) < 10) And Nz(F_L_GSD) >=
GSDmax Then
            Sel_BT = F_L_x
        End If

        'Condition 7'
        If (CInt(korrN_F_L) > 0) And (CInt(korrN_F_L) <= 3) And (Nz(F_L_GSD) <
GSDmax) Then
            Sel_BT = F_L_x
        End If

        'Condition 8'
        If (CInt(korrN_F_L) > 0) And (CInt(korrN_F_L) <= 3) And Nz(F_L_GSD) >=
GSDmax Then
            Sel_BT = F_L_x
        End If
    End If

    'Condition 9'
    If IsNull(korrN_F_L) Then
        If IsNumeric(L1_n) Then
            Sel_BT = L1_x
        ElseIf IsNumeric(L2_n) Then
            Sel_BT = L2_x
        ElseIf IsNumeric(L3_n) Then
            Sel_BT = L3_x
        ElseIf IsNumeric(L4_n) Then
            Sel_BT = L4_x
        Else: Sel_BT = Null
        End If
    End If
End Select
End Function

'returnerar selected BiotaTypeDetail
```

Public Function Sel_BTD(korrN_F_L, F_L_GM, F_L_GSD, GSDmin, GSDmax, GSDmean, LX_n, L1_n, L2_n, L3_n, L4_n, KorrN_FM, FM_x, LX_x, F_L_x, L1_x, L2_x, L3_x, L4_x, override) As Variant

```
Select Case override
   Case "F_L": Sel_BTD = F_L_x
   Case "LX": Sel_BTD = LX_x
   Case "L1": Sel_BTD = L1_x
   Case "L2": Sel_BTD = L2_x
   Case "L3": Sel_BTD = L3_x
   Case "L4": Sel_BTD = L4_x
   Case "FM": Sel_BTD = FM_x

   Case Else
   If IsNumeric(korrN_F_L) Then

      'Condition 1'
      If ((CInt(korrN_F_L) >= 10) And (Nz(F_L_GSD) > GSDmin) And (Nz(F_L_GSD) <
GSDmax)) Then
         Sel_BTD = F_L_x
      End If

      'Condition 2'
      If CInt(korrN_F_L) >= 10 And Nz(F_L_GSD) < GSDmin And Nz(F_L_GSD) <
GSDmax Then
         Sel_BTD = F_L_x
      End If

      'Condition 3'
      If CInt(korrN_F_L) >= 10 And Nz(F_L_GSD) >= GSDmax Then
         Sel_BTD = F_L_x
      End If

      'Condition 4'
      If (CInt(korrN_F_L) > 3) And (CInt(korrN_F_L) < 10) And (Nz(F_L_GSD) >
GSDmean) And (Nz(F_L_GSD) < GSDmax) Then
         Sel_BTD = F_L_x
      End If

      'Condition 5'
      If (CInt(korrN_F_L) > 3) And (CInt(korrN_F_L) < 10) And Nz(F_L_GSD) < GSDmean
And Nz(F_L_GSD) < GSDmax Then
         Sel_BTD = F_L_x
      End If

      'Condition 6'
      If (CInt(korrN_F_L) > 3) And (CInt(korrN_F_L) < 10) And Nz(F_L_GSD) >=
GSDmax Then
         Sel_BTD = F_L_x
      End If

      'Condition 7'
      If (CInt(korrN_F_L) > 0) And (CInt(korrN_F_L) <= 3) And (Nz(F_L_GSD) <
GSDmax) Then
         Sel_BTD = F_L_x
      End If
```

```
        'Condition 8'
        If (CInt(korrN_F_L) > 0) And (CInt(korrN_F_L) <= 3) And Nz(F_L_GSD) >=
GSDmax Then
            Sel_BTD = F_L_x
        End If
    End If

    'Condition 9'
    If IsNull(korrN_F_L) Then
        If IsNumeric(L1_n) Then
            Sel_BTD = L1_x
        ElseIf IsNumeric(L2_n) Then
            Sel_BTD = L2_x
        ElseIf IsNumeric(L3_n) Then
            Sel_BTD = L3_x
        ElseIf IsNumeric(L4_n) Then
            Sel_BTD = L4_x
        Else: Sel_BTD = Null
        End If
    End If
End Select
End Function

'returnerar original comment från Kd/CR databas'
Public Function Sel_com(korrN_F_L, F_L_GM, F_L_GSD, GSDmin, GSDmax,
GSDmean, LX_n, L1_n, L2_n, L3_n, L4_n, KorrN_FM, F_L_com, LX_com, L1_com,
L2_com, L3_com, L4_com, FM_com, override, overrideGSD) As Variant

Select Case override
    Case "F_L": Sel_com = F_L_com
    Case "LX": Sel_com = LX_com
    Case "L1": Sel_com = L1_com
    Case "L2": Sel_com = L2_com
    Case "L3": Sel_com = L3_com
    Case "L4": Sel_com = L4_com
    Case "FM": Sel_com = FM_com

    Case Else

    If IsNumeric(korrN_F_L) Then

     'Condition 1'
        If ((CInt(korrN_F_L) >= 10) And (Nz(F_L_GSD) > GSDmin) And (Nz(F_L_GSD) <
GSDmax)) Then
            If IsNumeric(KorrN_FM) And IsNull(LX_n) Then
                Sel_com = FM_com
            ElseIf IsNull(KorrN_FM) And IsNumeric(LX_n) Then Sel_com = LX_com
            Else: Sel_com = F_L_com
            End If
        End If
```

```
    'Condition 2'
    If CInt(korrN_F_L) >= 10 And Nz(F_L_GSD) < GSDmin And Nz(F_L_GSD) <
GSDmax Then
        If IsNumeric(KorrN_FM) And IsNull(LX_n) Then
            Sel_com = FM_com
        ElseIf IsNull(KorrN_FM) And IsNumeric(LX_n) Then Sel_com = LX_com
        Else: Sel_com = F_L_com
        End If
    End If

    'Condition 3'
    If CInt(korrN_F_L) >= 10 And Nz(F_L_GSD) >= GSDmax Then
        If IsNumeric(KorrN_FM) And IsNull(LX_n) Then
            Sel_com = FM_com
        ElseIf IsNull(KorrN_FM) And IsNumeric(LX_n) Then Sel_com = LX_com
        Else: Sel_com = F_L_com
        End If
    End If

    'Condition 4'
    If (CInt(korrN_F_L) > 3) And (CInt(korrN_F_L) < 10) And (Nz(F_L_GSD) >
GSDmean) And (Nz(F_L_GSD) < GSDmax) Then
        If IsNumeric(KorrN_FM) And IsNull(LX_n) Then
            Sel_com = FM_com
        ElseIf IsNull(KorrN_FM) And IsNumeric(LX_n) Then Sel_com = LX_com
        Else: Sel_com = F_L_com
        End If
    End If

    'Condition 5'
    If (CInt(korrN_F_L) > 3) And (CInt(korrN_F_L) < 10) And Nz(F_L_GSD) < GSDmean
And Nz(F_L_GSD) < GSDmax Then
        If IsNumeric(KorrN_FM) And IsNull(LX_n) Then
            Sel_com = FM_com
        ElseIf IsNull(KorrN_FM) And IsNumeric(LX_n) Then Sel_com = LX_com
        Else: Sel_com = F_L_com
        End If
    End If

    'Condition 6'
    If (CInt(korrN_F_L) > 3) And (CInt(korrN_F_L) < 10) And Nz(F_L_GSD) >=
GSDmax Then
        If IsNumeric(KorrN_FM) And IsNull(LX_n) Then
            Sel_com = FM_com
        ElseIf IsNull(KorrN_FM) And IsNumeric(LX_n) Then Sel_com = LX_com
        Else: Sel_com = F_L_com
        End If
    End If

    'Condition 7'
    If (CInt(korrN_F_L) > 0) And (CInt(korrN_F_L) <= 3) And (Nz(F_L_GSD) <
GSDmax) Then
```

```
            If IsNumeric(KorrN_FM) And IsNull(LX_n) Then
                Sel_com = FM_com
            ElseIf IsNull(KorrN_FM) And IsNumeric(LX_n) Then Sel_com = LX_com
            Else: Sel_com = F_L_com
            End If
        End If

        'Condition 8'
        If (CInt(korrN_F_L) > 0) And (CInt(korrN_F_L) <= 3) And Nz(F_L_GSD) >=
GSDmax Then
            If IsNumeric(KorrN_FM) And IsNull(LX_n) Then
                Sel_com = FM_com
            ElseIf IsNull(KorrN_FM) And IsNumeric(LX_n) Then Sel_com = LX_com
            Else: Sel_com = F_L_com
            End If
        End If
    End If

    'Condition 9'
    If IsNull(korrN_F_L) Then
        If IsNumeric(L1_n) Then
            Sel_com = L1_com
        ElseIf IsNumeric(L2_n) Then
            Sel_com = L2_com
        ElseIf IsNumeric(L3_n) Then
            Sel_com = L3_com
        ElseIf IsNumeric(L4_n) Then
            Sel_com = L4_com
        Else: Sel_com = Null
        End If
    End If
End Select
End Function

'returnerar datasource
```

Public Function Sel_Dsource(korrN_F_L, F_L_GM, F_L_GSD, GSDmin, GSDmax,

```
GSDmean, LX_n, F_L_n, L1_n, L2_n, L3_n, L4_n, KorrN_FM, override, overrideGSD) As
Variant

Select Case override
    Case "F_L": Sel_Dsource = "FMLX"
    Case "LX": Sel_Dsource = "LX"
    Case "L1": Sel_Dsource = "L1"
    Case "L2": Sel_Dsource = "L2"
    Case "L3": Sel_Dsource = "L3"
    Case "L4": Sel_Dsource = "L4"
    Case "FM": Sel_Dsource = "FM"

    Case Else

    If IsNumeric(korrN_F_L) Then
```

```
        'Condition 1'
        If ((CInt(korrN_F_L) >= 10) And (Nz(F_L_GSD) > GSDmin) And (Nz(F_L_GSD) <
GSDmax)) Then
            Sel_Dsource = "FMLX"
        End If

        'Condition 2'
        If CInt(korrN_F_L) >= 10 And Nz(F_L_GSD) < GSDmin And Nz(F_L_GSD) <
GSDmax Then
            Sel_Dsource = "FMLX"
        End If

        'Condition 3'
        If CInt(korrN_F_L) >= 10 And Nz(F_L_GSD) >= GSDmax Then
            Sel_Dsource = "FMLX"
        End If

        'Condition 4'
        If (CInt(korrN_F_L) > 3) And (CInt(korrN_F_L) < 10) And (Nz(F_L_GSD) >
GSDmean) And (Nz(F_L_GSD) < GSDmax) Then
           Sel_Dsource = "FMLX"
        End If

        'Condition 5'
        If (CInt(korrN_F_L) > 3) And (CInt(korrN_F_L) < 10) And Nz(F_L_GSD) < GSDmean
And Nz(F_L_GSD) < GSDmax Then
            Sel_Dsource = "FMLX"
        End If

        'Condition 6'
        If (CInt(korrN_F_L) > 3) And (CInt(korrN_F_L) < 10) And Nz(F_L_GSD) >=
GSDmax Then
            Sel_Dsource = "FMLX"
        End If

        'Condition 7'
        If (CInt(korrN_F_L) > 0) And (CInt(korrN_F_L) <= 3) And (Nz(F_L_GSD) <
GSDmax) Then
            Sel_Dsource = "FMLX"
        End If

        'Condition 8'
        If (CInt(korrN_F_L) > 0) And (CInt(korrN_F_L) <= 3) And Nz(F_L_GSD) >=
GSDmax Then
            Sel_Dsource = "FMLX"
        End If
        If IsNull(KorrN_FM) Then Sel_Dsource = "LX"
        If IsNull(LX_n) Then Sel_Dsource = "FM"
    End If

    'Condition 9'
    If IsNull(korrN_F_L) Then
```

```
        If IsNumeric(L1_n) Then
            Sel_Dsource = "L1"
        ElseIf IsNumeric(L2_n) Then
            Sel_Dsource = "L2"
        ElseIf IsNumeric(L3_n) Then
            Sel_Dsource = "L3"
        ElseIf IsNumeric(L4_n) Then
            Sel_Dsource = "L4"
        Else: Sel_Dsource = Null
        End If
    End If
End Select
End Function
'returnerar SenseCheck (validering)

Public Function L_valid(LX_n, LX_GM, LX_GSD, L1_n, L1_GM, L1_GSD, L2_n,
L2_GM, L2_GSD, L3_n, L3_GM, L3_GSD, L4_n, L4_GM, L4_GSD, SelectedMin,
SelectedMax, SelectedGM, SelectedGSD, DataSource)
Dim LX_5p, LX_95p, L1_5p, L1_95p, L2_5p, L2_95p, L3_5p, L3_95p, L4_5p, L4_95p,
Lmin, Lmax, andelP

Sel_5p = SelectedGM / SelectedGSD ^ 1.645
Sel_95p = SelectedGM * SelectedGSD ^ 1.645
LX_5p = LX_GM / LX_GSD ^ 1.645
LX_95p = LX_GM * LX_GSD ^ 1.645
L1_5p = L1_GM / L1_GSD ^ 1.645
L1_95p = L1_GM * L1_GSD ^ 1.645
L2_5p = L2_GM / L2_GSD ^ 1.645
L2_95p = L2_GM * L2_GSD ^ 1.645
L3_5p = L3_GM / L3_GSD ^ 1.645
L3_95p = L3_GM * L3_GSD ^ 1.645
L4_5p = L4_GM / L4_GSD ^ 1.645
L4_95p = L4_GM * L4_GSD ^ 1.645

Lmin = Min4Val(L1_5p, L2_5p, L3_5p, L4_5p)
Lmax = Max4Val(L1_95p, L2_95p, L3_95p, L4_95p)


If DataSource = "FMLX" Or DataSource = "FM" Or DataSource = "LX" Then
    If (IsNumeric(Sel_5p) And IsNumeric(Sel_95p) And IsNumeric(Lmax) And
IsNumeric(Lmin)) Then
        If CDbl(Sel_5p) > CDbl(Lmin) And CDbl(Sel_95p) < CDbl(Lmax) Then
            L_valid = "S1"
        End If
        If CDbl(Sel_5p) < CDbl(Lmin) And CDbl(Sel_95p) > CDbl(Lmax) Then
            L_valid = "S3"
        End If
    End If
    If IsNumeric(Sel_95p) And IsNumeric(Sel_5p) And IsNumeric(Lmin) And
IsNumeric(Lmax) Then
        If CDbl(Sel_95p) > CDbl(Lmin) And CDbl(Sel_95p) < CDbl(Lmax) And CDbl(Sel_5p)
< CDbl(Lmin) And CDbl(Sel_5p) < CDbl(Lmax) Then
```

```
            andelP = 1 - (Log(CDbl(Lmin)) - Log(CDbl(Sel_5p))) / (Log(CDbl(Sel_95p)) -
Log(CDbl(Sel_5p)))
            L_valid = "S2(L" & Int(100 * andelP) & "%)"
        End If
    End If
    If IsNumeric(Sel_95p) And IsNumeric(Sel_5p) And IsNumeric(Lmax) And
IsNumeric(Lmin) Then
        If CDbl(Sel_5p) < CDbl(Lmax) And CDbl(Sel_5p) > CDbl(Lmin) And CDbl(Sel_95p)
> CDbl(Lmax) And CDbl(Sel_95p) > CDbl(Lmin) Then
            andelP = 1 - (Log(CDbl(Sel_95p)) - Log(CDbl(Lmax))) / (Log(CDbl(Sel_95p)) -
Log(CDbl(Sel_5p)))
            L_valid = "S2(U" & Int(100 * andelP) & "%)"
        End If
    End If
        If IsNumeric(Sel_5p) And IsNumeric(Lmax) Then
        If CDbl(Sel_5p) > CDbl(Lmax) Then
            L_valid = "S4"
        End If
    End If
    If IsNumeric(Sel_95p) And IsNumeric(Lmin) Then
        If CDbl(Sel_95p) < CDbl(Lmin) Then
            L_valid = "S4"
        End If
    End If
End If
End Function

'returnerar check av överlapp inom litteraturdata
Public Function L_check(L1_n, L1_GM, L1_GSD, L2_n, L2_GM, L2_GSD, L3_n,
L3_GM, L3_GSD, L4_n, L4_GM, L4_GSD)
Dim L1_5p, L1_25p, L1_75p, L1_95p, L2_5p, L2_25p, L2_75p, L2_95p, L3_5p, L3_25p,
L3_75p, L3_95p, L4_5p, L4_25p, L4_75p, L4_95p
Dim LitTot, LitTot_val, CondTot, CondTrue As Integer

L1_5p = L1_GM / L1_GSD ^ 1.645
L1_95p = L1_GM * L1_GSD ^ 1.645
L2_5p = L2_GM / L2_GSD ^ 1.645
L2_95p = L2_GM * L2_GSD ^ 1.645
L3_5p = L3_GM / L3_GSD ^ 1.645
L3_95p = L3_GM * L3_GSD ^ 1.645
L4_5p = L4_GM / L4_GSD ^ 1.645
L4_95p = L4_GM * L4_GSD ^ 1.645

LitTot = -1 * (IsNumeric(L1_n) + IsNumeric(L2_n) + IsNumeric(L3_n) + IsNumeric(L4_n))
LitTot_val = -1 * (IsNumeric(L1_5p) + IsNumeric(L2_5p) + IsNumeric(L3_5p) +
IsNumeric(L4_5p))
Select Case LitTot_val
    Case 0: CondTot = 0
    Case 1: CondTot = 0
    Case 2: CondTot = 1
    Case 3: CondTot = 3
    Case 4: CondTot = 6
```

```
    Case 5: CondTot = 10
End Select
CondTrue = 0

'L1 vs L2
If IsNumeric(L1_5p) And IsNumeric(L1_95p) And IsNumeric(L2_5p) And
IsNumeric(L2_95p) Then
    If Not ((CDbl(L1_5p) > CDbl(L2_95p)) Or (CDbl(L2_5p) > CDbl(L1_95p))) Then
CondTrue = CondTrue + 1
End If

'L1 vs L3
If IsNumeric(L1_5p) And IsNumeric(L1_95p) And IsNumeric(L3_5p) And
IsNumeric(L3_95p) Then
    If Not ((CDbl(L1_5p) > CDbl(L3_95p)) Or (CDbl(L3_5p) > CDbl(L1_95p))) Then
CondTrue = CondTrue + 1
End If

'L1 vs L4
If IsNumeric(L1_5p) And IsNumeric(L1_95p) And IsNumeric(L4_5p) And
IsNumeric(L4_95p) Then
    If Not ((CDbl(L1_5p) > CDbl(L4_95p)) Or (CDbl(L4_5p) > CDbl(L1_95p))) Then
CondTrue = CondTrue + 1
End If

'L2 vs L3
If IsNumeric(L2_5p) And IsNumeric(L2_95p) And IsNumeric(L3_5p) And
IsNumeric(L3_95p) Then
    If Not ((CDbl(L2_5p) > CDbl(L3_95p)) Or (CDbl(L3_5p) > CDbl(L2_95p))) Then
CondTrue = CondTrue + 1
End If

'L3 vs L4
If IsNumeric(L4_5p) And IsNumeric(L4_95p) And IsNumeric(L3_5p) And
IsNumeric(L3_95p) Then
    If Not ((CDbl(L4_5p) > CDbl(L3_95p)) Or (CDbl(L3_5p) > CDbl(L4_95p))) Then
CondTrue = CondTrue + 1
End If

'L2 vs L4
If IsNumeric(L4_5p) And IsNumeric(L4_95p) And IsNumeric(L2_5p) And
IsNumeric(L2_95p) Then
    If Not ((CDbl(L4_5p) > CDbl(L2_95p)) Or (CDbl(L2_5p) > CDbl(L4_95p))) Then
CondTrue = CondTrue + 1
End If

L_check = LitTot & ":" & CondTrue & "/" & CondTot
End Function

'checks overlap between Forsmark and Laxemardata
Public Function FmLxCheck(FM_GM, FM_GSD, LX_GM, LX_GSD)
Dim LX_5p, LX_95p, FM_5p, FM_95p
```

```
LX_5p = LX_GM / LX_GSD ^ 1.645
LX_95p = LX_GM * LX_GSD ^ 1.645
FM_5p = FM_GM / FM_GSD ^ 1.645
FM_95p = FM_GM * FM_GSD ^ 1.645

If (IsNumeric(FM_5p) And IsNumeric(FM_95p) And IsNumeric(LX_95p) And
IsNumeric(LX_5p)) Then
   If CDbl(FM_5p) > CDbl(LX_5p) And CDbl(FM_95p) < CDbl(LX_95p) Then
      FmLxCheck = "S1"
   End If
   If CDbl(FM_5p) < CDbl(LX_5p) And CDbl(FM_95p) > CDbl(LX_95p) Then
      FmLxCheck = "S3"
   End If
End If

If IsNumeric(FM_95p) And IsNumeric(FM_5p) And IsNumeric(LX_5p) And
IsNumeric(LX_95p) Then
   If CDbl(FM_95p) > CDbl(LX_5p) And CDbl(FM_95p) < CDbl(LX_95p) And
CDbl(FM_5p) < CDbl(LX_5p) And CDbl(FM_5p) < CDbl(LX_95p) Then
      andelP = 1 - (Log(CDbl(LX_5p)) - Log(CDbl(FM_5p))) / (Log(CDbl(FM_95p)) -
Log(CDbl(FM_5p)))
      FmLxCheck = "S2(L" & Int(100 * andelP) & "%)"
   End If
End If

If IsNumeric(FM_95p) And IsNumeric(FM_5p) And IsNumeric(LX_95p) And
IsNumeric(LX_5p) Then
   If CDbl(FM_5p) < CDbl(LX_95p) And CDbl(FM_5p) > CDbl(LX_5p) And
CDbl(FM_95p) > CDbl(LX_95p) And CDbl(FM_95p) > CDbl(LX_5p) Then
      andelP = 1 - (Log(CDbl(FM_95p)) - Log(CDbl(LX_95p))) / (Log(CDbl(FM_95p)) -
Log(CDbl(FM_5p)))
      FmLxCheck = "S2(U" & Int(100 * andelP) & "%)"
   End If
End If

If IsNumeric(FM_5p) And IsNumeric(LX_95p) Then
   If CDbl(FM_5p) > CDbl(LX_95p) Then
      FmLxCheck = "S4"
   End If
End If

If IsNumeric(FM_95p) And IsNumeric(LX_5p) Then
   If CDbl(FM_95p) < CDbl(LX_5p) Then
      FmLxCheck = "S4"
   End If
End If

End Function

'checks overlap between Forsmark and Laxemardata
```

Public Function FmLxCheck2(FM_GM, FM_GSD, LX_GM, LX_GSD)

```
Dim LX_5p, LX_95p, FM_5p, FM_95p

LX_5p = LX_GM / LX_GSD ^ 1.645
LX_95p = LX_GM * LX_GSD ^ 1.645
FM_5p = FM_GM / FM_GSD ^ 1.645
FM_95p = FM_GM * FM_GSD ^ 1.645

If (IsNumeric(FM_5p) And IsNumeric(FM_95p) And IsNumeric(LX_95p) And
IsNumeric(LX_5p)) Then
   If CDbl(FM_5p) > CDbl(LX_5p) And CDbl(FM_95p) < CDbl(LX_95p) Then
      FmLxCheck2 = "S1"
   End If
   If CDbl(FM_5p) < CDbl(LX_5p) And CDbl(FM_95p) > CDbl(LX_95p) Then
      FmLxCheck2 = "S3"
   End If
End If

If IsNumeric(FM_95p) And IsNumeric(FM_5p) And IsNumeric(LX_5p) And
IsNumeric(LX_95p) Then
   If CDbl(FM_95p) > CDbl(LX_5p) And CDbl(FM_95p) < CDbl(LX_95p) And
CDbl(FM_5p) < CDbl(LX_5p) And CDbl(FM_5p) < CDbl(LX_95p) Then
      andelP = 1 - (Log(CDbl(LX_5p)) - Log(CDbl(FM_5p))) / (Log(CDbl(FM_95p)) -
Log(CDbl(FM_5p)))
      If andelP > 0.5 Then FmLxCheck2 = "S2>50" Else FmLxCheck2 = "S2<50"
   End If
End If

If IsNumeric(FM_95p) And IsNumeric(FM_5p) And IsNumeric(LX_95p) And
IsNumeric(LX_5p) Then
   If CDbl(FM_5p) < CDbl(LX_95p) And CDbl(FM_5p) > CDbl(LX_5p) And
CDbl(FM_95p) > CDbl(LX_95p) And CDbl(FM_95p) > CDbl(LX_5p) Then
      andelP = 1 - (Log(CDbl(FM_95p)) - Log(CDbl(LX_95p))) / (Log(CDbl(FM_95p)) -
Log(CDbl(FM_5p)))
      If andelP > 0.5 Then FmLxCheck2 = "S2>50" Else FmLxCheck2 = "S2<50"
   End If
End If

If IsNumeric(FM_5p) And IsNumeric(LX_95p) Then
   If CDbl(FM_5p) > CDbl(LX_95p) Then
      FmLxCheck2 = "S4"
   End If
End If

If IsNumeric(FM_95p) And IsNumeric(LX_5p) Then
   If CDbl(FM_95p) < CDbl(LX_5p) Then
      FmLxCheck2 = "S4"
   End If
End If

End Function
```

3.5.2 VBA module "Classifications"

```
Option Compare Database

Public Function OKcheck(condition, Validation, FmLxCheck, LitCheck, UOA,
p_override, DataSource)

Select Case condition
  Case "1", "2", "4", "5", "7"
    Select Case Left(Validation, 2)
    Case "S1"
      OKcheck = "SO"
    Case "S2"
      If Val(Mid([Validation], 5, 2)) > 50 Then
        OKcheck = "SO"
      Else: OKcheck = "SV"
      End If
    Case "S3"
      OKcheck = "SV"
    Case "S4"
      OKcheck = "SV"
    Case Else
      OKcheck = "SX"
    End Select
  Case "3", "6", "8"
    OKcheck = "SM"
  Case "9"
    If DataSource = "LX" Then
      Select Case Left(Validation, 2)
      Case "S1"
        OKcheck = "TO"
      Case "S2"
        If Val(Mid([Validation], 5, 2)) > 50 Then
          OKcheck = "TO"
        Else: OKcheck = "TV"
        End If
      Case "S3"
        OKcheck = "TV"
      Case "S4"
        OKcheck = "TV"
      Case Else
        OKcheck = "TX"
      End Select
    Else
      Select Case LitCheck
        Case "0:0/0": OKcheck = "LX"
        Case "1:0/0": OKcheck = "LX"
        Case "2:1/1": OKcheck = "LO"
        Case "3:1/1": OKcheck = "LO"
        Case "3:1/2": OKcheck = "LO"
        Case "3:2/2": OKcheck = "LO"
        Case "3:2/3": OKcheck = "LO"
```

```
            Case "3:3/3": OKcheck = "LO"
            Case "4:1/1": OKcheck = "LO"
            Case "4:1/2": OKcheck = "LO"
            Case "4:2/2": OKcheck = "LO"
            Case "4:2/3": OKcheck = "LO"
            Case "4:3/3": OKcheck = "LO"
            Case "4:2/4": OKcheck = "LO"
            Case "4:3/4": OKcheck = "LO"
            Case "4:4/4": OKcheck = "LO"
            Case "4:3/6": OKcheck = "LO"
            Case "4:4/6": OKcheck = "LO"
            Case "4:5/6": OKcheck = "LO"
            Case "4:6/6": OKcheck = "LO"
            Case Else: OKcheck = "LV"
        End Select
    End If
End Select
If UOA <> "" Then
   OKcheck = UOA
End If
If p_override <> "" Then
   OKcheck = "OV"
End If
If DataSource = "L5" Then
   OKcheck = "L5"
End If

End Function
```

Public Function Warnings(condition, selected_gsd, GSDmax)

```
If IsNumeric(selected_gsd) And IsNumeric(GSDmax) Then
   If CDbl(selected_gsd) > CDbl(GSDmax) Then Warnings = ">GSDmax"
End If
End Function
```