Aalto University
School of Electrical Engineering

Non-linear regression methods in open-source platforms for autoantibody analysis in type 1 diabetes

Rasmus Siljander

School of Electrical Engineering

Bachelor's thesis

Espoo 13.5.2020

Study coordinator	University Lecturer Markus Turunen, D.Sc.
Executive Instructor	University Lecturer Markus Turunen, D.Sc.
Assisting Instructor	Docent Heli Siljander, M.D., Ph.D.

Aalto University
School of Electrical Engineering

Aalto University, PL 1100, 00076
AALTO
www.aalto.fi
Bachelor's thesis abstract

Author: Rasmus Siljander

Title: Non-linear regression methods in open-source platforms for autoantibody analysis in type 1 diabetes

Program: Bachelor's Program in Electrical Engineering

Major: Bioinformation technology **Code:** ELEC3016

Responsible Teacher: University Lecturer Markus Turunen, D.Sc.

Instructors: Docent Heli Siljander, senior researcher, M.D., Ph.D.; D.Sc. Markus Turunen

Date: 13.5.2020 **Number of pages:** 4+31 **Language:** English

Abstract

Autoantibodies against truncated GAD (tGADA$_{96\text{-}585}$) are an example of biochemical autoantibodies that are key in the diagnostics of Type 1 Diabetes. tGADA$_{96\text{-}585}$ can be analyzed with a four-parameter logistic (4PL) regression model. Analysis is usually done with specialized software, but due to the availability of open-source (OS) platforms, it is relevant to evaluate whether OS programs can replicate the analysis. Using details from the MultiCalc^{TM} (Wallac OY, Turku, Finland) product manual, a novel Python program was built and evaluated. Model accuracy was assessed with tGADA$_{96\text{-}585}$ observations collected from different Pediatric Diabetes Research Group (University of Helsinki, Finland) studies. Differences in mean recovery error (MER; 0.7%) showed similarity between methods. The normal distributions of the relative errors were used to discover, that the likelihood of sample positivity misclassification exceeded 5% for concentrations between 4.9-5.8 RU. This accuracy was satisfactory. The lack of comprehensive algorithm instructions was identified as a source of error. Opportunities for improvement included modifications of sample size, model validation measures, and algorithm initialization. As improvements were available to a satisfactory program, it was concluded that OS programming has potential for future 4PL analysis.

Keywords four-parameter logistic regression, open-source platforms, tGADA$_{96\text{-}585}$

Foreword

I would like to thank Heli Siljander for giving me the chance to explore my interests and goals throughout this process. My gratitude for this opportunity extends to the whole Pediatric Diabetes Research Group, starting with Mikael Knip, Taina Härkönen, and the entire organization.

Table of contents

Symbols, Definitions, and abbreviations

Symbols

A	Lower asymptote of the 4PL curve
B	Slope at the inflexion point of the 4PL curve
C	Inflexion point of the 4PL curve
D	Upper asymptote of the 4PL curve
α	Transformation of 4PL parameter C
β	Transformation of 4PL parameter B
y_{obs}	Individual data measurements
y_{model}	Predicted model values for the given data
$\boldsymbol{w}$	Vector containing weights used in weighted regression
c_{obs}	Autoantibody concentration estimated by the algorithm
c_{exp}	Actual (standard solution) autoantibody concentration value
$R_{\%}$	Percentage ratio of observed and expected concentration values
c_{mult}	Autoantibody concentration calculated by MultiCalc^{TM}
c_{alg}	Autoantibody concentration calculated with Python
ϵ	(Relative) Error in algorithm calcultions
μ	Mean value of data
σ	Data standard deviation value
$CV_{\%}$	Coefficient of variation
χ^2	Chi-squared (test for population fit)
p, x, y, k	Arbitrary function variables

Definitions

Serum	Fluid component of blood containing molecules that are not related to blood clotting.

Radioimmunoassay, RIA	A method of calculating antibody concentrations, where changes in the activity of radioactively labeled antigens are monitored.
Convergence rate, CR	Measure of the approach of a sequence to a single value. Traditionally calculated using the ratio of successive sequence values.
Open-source platforms	Software platforms where source code can be openly distributed.

Abbreviations

T1D	Type 1 Diabetes
IAA	Insulin autoantibodies
GADA	Autoantibodies against glutamic acid decarboxylase
IA-2A	Autoantibodies against islet antigen-2
ZnT8A	Autoantibodies against zinc transporter 8
4PL	Four-parameter logistic model
RU	Relative units; unit of antibody concentration.
CPM	Counts per minute; radioactivity unit in RIA.
$tGADA_{96–585}$	Autoantibodies against truncated (amino acids 95-585) glutamic acid decarboxylase
PEDIA	Pediatric Diabetes Research Group, University of Helsinki, Finland
LSE	Least-squared error
WLSE	Weighted least-squared error
MER	Mean error in recovery
$CV_{\%}$	coefficient of variation
SW	Shapiro-Wilks (test for normality)
MAE	Mean absolute error
RMSE	Root mean squared error
5PL	Five-parameter logistic (model)

1. Introduction

Type 1 Diabetes (T1D) is an autoimmune disease that results in the loss of pancreatic β-cells responsible for insulin production. This leads to failure in glucose regulation and can cause severe symptoms including blindness and kidney failure (Boldison and Wong, 2018). As stated by Siljander et al. (2019), T1D patients require life-long insulin treatment. With an increasing incidence of T1D in the 21st century (Harjutsalo et al., 2013), there is a motivation for finding solutions to sample analysis that could help discover the cure to T1D.

Although the exact causes of T1D are unknown, four key biochemical autoantibodies have been found that can identify individuals susceptible to T1D (Kallionpää et al., 2019). Autoantibodies are molecules that bind to self-antigens, which are structures within the body that trigger an antibody response (Elkon and Casali, 2008). This process is called an autoimmune reaction. Autoantibodies against insulin (IAA), glutamic acid decarboxylase (GADA), islet antigen-2 (IA-2A), and zinc transporter 8 (ZnT8A) are all indicators of an autoimmune process. These autoantibodies have become instrumental in evaluating the onset and progression of T1D. (Kallionpää et al., 2019)

Autoantibodies can be measured from serum samples by a process called radioimmunoassay, where the binding of autoantibodies to radioactively labelled antigens is measured (Zaidi and Kama, 1993). As the radioactive response is relative to the number of labelled molecules in solution, autoantibody concentrations can be directly associated with their respective assay response measurement. This relationship can be modeled by methods including the four-parameter logistic model (4PL) (Dudley et al., 1985), often executed by specifically tailored software such as MultiCalc^{TM} (Perkin Elmer Life Sciences Wallac, Turku, Finland). However, autoantibody assays were mainly developed in the 1990s (Wallac Oy, 1999), and despite updates, complications between software solutions and data storage compatibilities still exist. Thus, it has become relevant to evaluate, whether open-source programming could replicate the analysis process in a more efficient manner, especially in relation to data storage and accessibility.

This paper will inspect a Python algorithm. The algorithm was initially built according to details outlined in the MultiCalc^{TM} manual (Wallac Oy, 1999), and then further refined to ensure functionality. Due to their superior accessibility and compatibility, open-source programs could help initiate a paradigm shift in autoantibody regression analysis.

2. Aims

The goal of this paper is to evaluate whether MultiCalc^{TM} software can be implemented with open-source programming. This involves inspecting four different components of the process. Firstly, sample measurement and positivity via radioimmunoassay will be briefly reviewed. Secondly, key mechanisms employed in the 4PL algorithm will be introduced, such as weighted linear and logistic regression concepts. Then, Python implementation of the MultiCalc^{TM} algorithm will be reviewed. Finally, further application possibilities will be recommended based on the Python algorithm accuracy, which will be referred to as "the goodness of fit".

The mechanisms of radioimmunoassay will only be reviewed in principle. Other topics to fall out of the scope of this investigation are the biological characteristics and practical issues related to sampling. As evaluating the mathematical model forms the core of this paper, a detailed review of radioactive processes in sample analysis is irrelevant and does not enhance our understanding of regression algorithms. Additionally, other regression methods such as the five-parameter logistic model or spline interpolation will not be discussed. This restriction to 4PL is due to the assessment of only one biochemical autoantibody, the autoantibodies against truncated GAD (tGADA$_{96–585}$). In the material, tGADA$_{96–585}$ was analyzed uniquely with 4PL methods. This factor will be more elaborately discussed in Sections 4 and 5.

3. Theoretical background

3.1. Radioimmunoassays and standard curves

As mentioned in Section 1, MultiCalc^{TM} (Wallac Oy, 1999) can be used to calculate autoantibody concentrations from assay responses. Concentrations, measured in relative units (RU), are modeled from responses, measured in counts per minute (CPM). Further exploration is presented Section 5.1. Notably, the existence of non-specific binding makes interpretation of individual response values impossible, and CPM-RU associations need to be developed for assays individually. (Wallac Oy, 1999)

Non-specific binding is the binding of proteins to unwanted molecules (Mendel and Mendel, 1985). Causes for said binding can be attributed to different factors, such as “sample container material, blocking agents, and interference between chemical antibody characteristics and the analysis systems” (Güven et al., 2014). There is variation in the level of non-specific binding between assays, meaning that knowledge about how non-specific binding has affected the 4PL curve fit is required. Standard solutions offer this knowledge.

A standard series is a collection of samples for which the concentration of the reacting substance is known. By analyzing these solutions with samples of unknown concentrations in the assay, a concentration-response regression curve can be built purely based on the responses. This curve is called the standard curve. Sample responses for unknown concentrations can then be evaluated on this curve and a specific RU value for this response can be extrapolated once the curve model is known.

3.2. Four-parameter logistic model

The four-parametric logistic (4PL) model is used to construct standard curves in the MultiCalc^{TM} software. It is a sigmoidal-shaped curve characterized (Figure 1) by four parameters, following the equation

$$y = D + \frac{A-D}{1+\left(\frac{x}{C}\right)^{B}} \tag{1}$$

where A and D are the upper and lower asymptotes of the curve. Parameter C can be interpreted as the point of inflexion, and B as the slope at that point (Mauriz et al., 2006). Alternatively, B and C can be presented as α and β, where equation (1) is equivalent to

$$y = \mathrm{D} + \frac{\mathrm{A}-\mathrm{D}}{1+e^{-\alpha-\beta \ln x}} \tag{2}$$

The lower asymptote can be interpreted as the level of non-specific binding in the assay, as it portrays assay response when the concentration approaches zero.

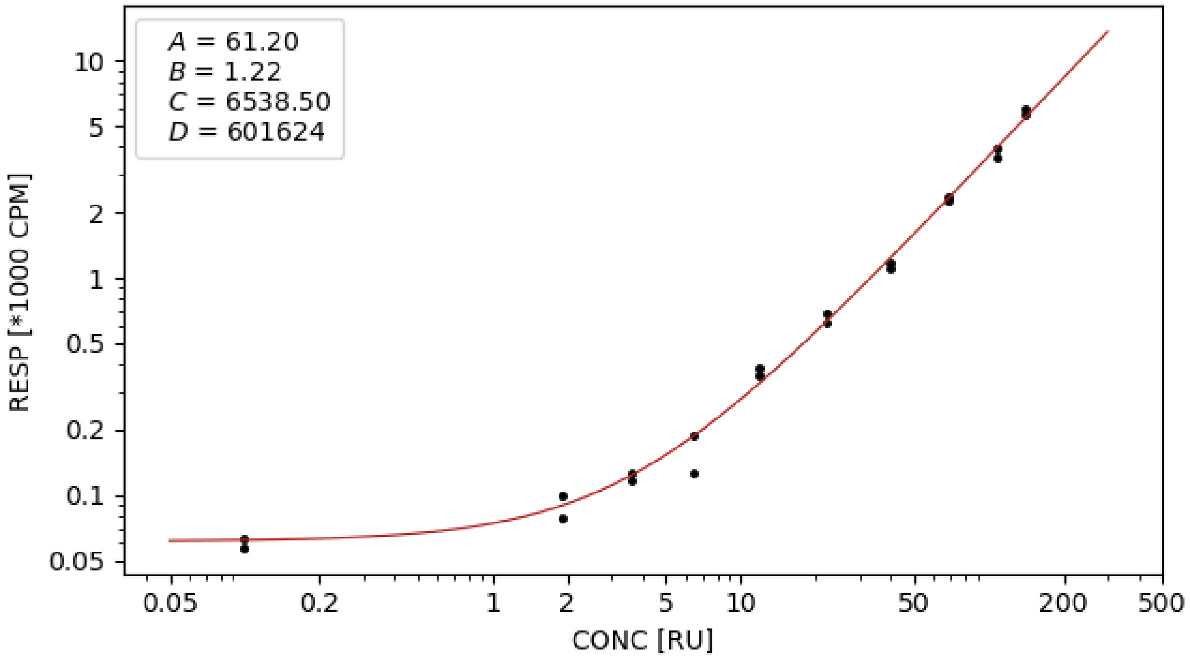


Figure 1: Example of a 4PL standard curve. Figure shows response (RESP, measured in radioactive counts per minute (CPM) multiplied by 1000) as a function of autoantibody concentration (CONC, measured in relative units (RU)). Both axes are on a logarithmic scale. Curve parameter values, fit by MultiCalc software, are presented on the legend in the top left. Noteworthy: The upper asymptote (the upper curve of the "s") is missing from this curve

In general, logistic regression calculates probabilities that continuous data falls under a certain category. The model predicts categories for unseen data. In the model, probabilities are related to each other in the following way:

$$\ln\left(\frac{P(y=Y)}{P(y=Y')}\right) = w^T x \tag{3}$$

Here ln is the natural logarithm and $P(y = Y)$ is the probability that the observation y belongs to class Y. Y' is the complement of Y, and naturally $P(y = Y') = 1 - P(y = Y)$. Similarly to how parameters a and b are applied in linear regression, operator $w^T x$ is a linear predictor and w^T is its weighting vector: $y = ax + b$ is often denoted as $y = w^T x$. Solving for $P(y = Y)$, equation (3) can be rearranged to

$$P(y = Y) = \frac{1}{1+e^{-w^T x}} \tag{4}$$

The weight vector is then chosen by maximizing the probability that condition Y is encountered with given data set x. (Jung, 2019) Similarly, equation (2) can be rearranged to

$$\frac{y-D}{A-D} = \frac{1}{1+e^{-\alpha-\beta \ln x}} \tag{5}$$

In the context of the general logistic model, the ratio $\frac{y-D}{A-D}$ describes a probability distribution characterized by parameters A and D, where probability reaches 1 when $y = A$ and 0 when $y = D$. The expression $\alpha + \beta \ln()$ is a weighting operator applied to the data to best describe the distribution.

3.3. Weighting methods for regression

When defining regression models, parameters are chosen such that the measurement errors are minimized. This can be done with least squared error (LSE) (Jung, 2019; Wasserman, 2005).

$$LSE = \sum_i [y_{obs}(i) - y_{model}(i)]^2 \tag{6}$$

where y_{obs} and y_{model} denote the observations and the model values at that measurement respectively. Minimizing LSE assumes that noise error has constant variance, also known has homoscedasticity (Wasserman, 2005; DiCiccio et al., 2017). However, in radioimmunoassay homoscedasticity is often violated, as variance is more likely a function of response magnitude. This is due to the binding kinetics of the assay as well as the noise in signal detectors that increase with response. (Gottschalk and Dunn, 2005)

This leads to issues in model fit, as LSE emphasizes minimizing errors in the end range of responses. This overfit may results in a lack of fit around responses with smaller values. This can be solved with weighting vectors $\boldsymbol{w}$ that emphasize appropriate data points. The vector acts as a normalizer, allowing for unbiased fit. (DiCiccio, et al., 2017). Weighted LSE (WLSE) hence becomes

$$WLSE = \sum_i (w_i(y_{obs}(i) - y_{model}(i))^2 \tag{7}$$

These weighting methods can be used to estimate 4PL parameters in the algorithm. Further details about the process are presented in Section 5.2.

4. Material

Data was collected from 167 previous tGADA$_{96–585}$ assay analyses run by the Pediatric Diabetes Reseacrh Group (PEDIA; University of Helsinki, Finland). These assays contained samples from several different study cohorts, comprising samples from both minors and adults. All subjects had given their consent for analysis, and data was managed confidentially. All current studies had received approval from local ethics committees and were carried out according to the principles of the Helsinki Declaration.

From each assay, three sets of data were collected: 1) response for standard solutions 2) MultiCalc^{TM} estimates for curve parameters 3) responses samples of unknown concentrations. In every assay cycle, there was up to 2 replicates of 10 different standard concentrations, amounting to a maximum of 20 observations per standards analysis. The standard concentrations used in RIA were between 0.1 and 139.9 relative units (RU). The rest of the sample spaces (76) were used to measure samples of unknown concentrations. Data from the 167 analysis runs was split into training and test sets. The training set (n=142) was used to compile the algorithm, and the test set contained 25 runs, for which both the standard responses as well as the sample responses were analyzed. Below are sample tables of all data sets.

Table 1: Sample of MultiCalc^{TM} parameter data. The letters $A, B, C,$ and D signify the 4PL curve parameters. Run ID is a number given to the assays in the order they were recorded. A and B are presented to 3 significant figures, C to 4 significant figures, and D to 5.

	Run ID					
Parameter	**8**	**9**	**10**	**11**	**12**	**13**
A	34.3	33.3	30.9	31.4	29.5	29.8
B	1.34	1.31	1.15	1.36	1.16	1.55
C	317.1	4437	4361	4371	1678	85.89
D	11681	271220	136030	308420	48224	2249.2

Table 2: Sample of standard concentration data. The first column contains the standard concentrations (RU) and the subsequent column are its respective responses (CPM) for a certain Run ID. All responses are presented as whole numbers. NA values signify "not available".

	Response (CPM) for Run ID					
Concentration (RU)	**8**	**9**	**10**	**11**	**12**	**13**
0.100	33	31	36	31	27	26
0.100	34	32	28	26	31	27
1.908	46	51	48	40	47	47
1.908	47	48	52	NA	56	27
3.626	68	52	75	NA	67	36
3.626	58	51	59	NA	70	48
6.406	106	132	107	89	101	88
6.406	98	92	101	80	99	67
11.815	194	173	182	137	190	149
11.815	175	158	181	142	194	107
21.956	277	265	300	245	273	227
21.956	236	308	333	255	223	278
40.178	754	578	744	604	649	435
40.178	757	713	644	564	700	600
68.553	1215	1140	1246	926	1236	954
68.553	1361	1179	1324	1046	1358	943
106.787	1710	1666	1872	1249	1683	1293
106.787	2215	1908	1798	1633	1886	1336
139.990	2912	3540	3506	3339	2915	3067
139.990	3908	3292	3282	3089	3769	2328

Table 3: Sample data of unknown concentrations. Columns are (CPM) responses with unknown concentrations. Each response column is compared to the standard curve constructed from standard data with a matching Run ID.

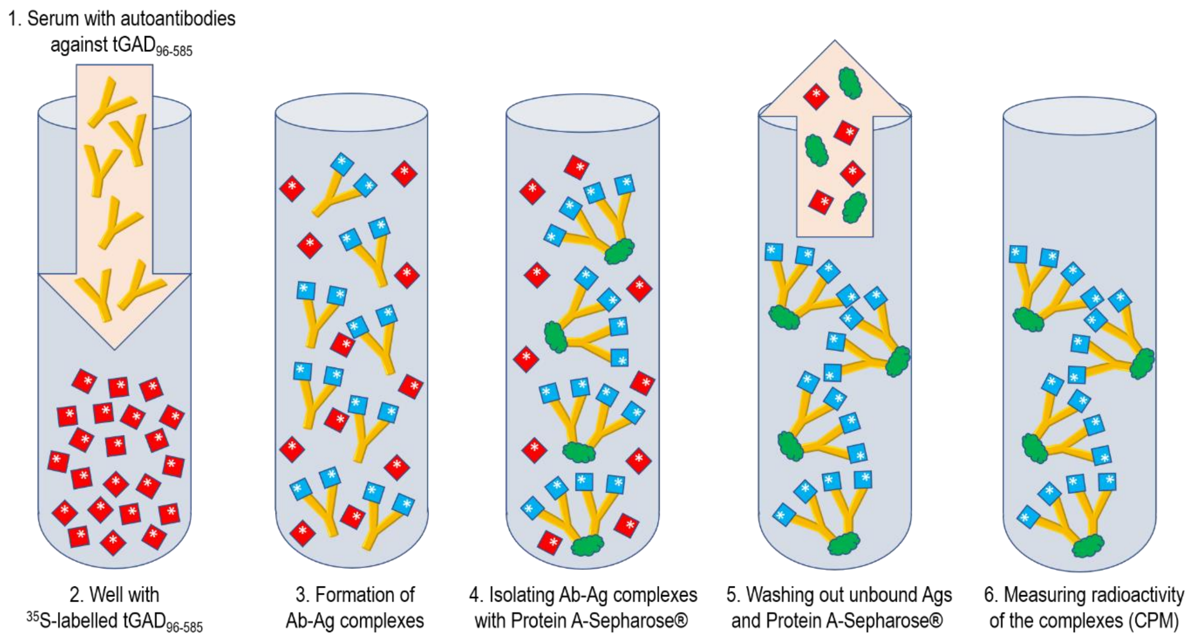
Response (CPM) for Run ID					
4	**6**	**7**	**8**	**9**	**10**
199	202	32	110	510	63
239	187	41	142	533	81
256	679	21	138	49	99
263	711	30	129	35	188
142	30	65	597	36	355
127	29	77	536	35	540
155	29	383	28	38	495
203	40	597	34	42	426
1370	44	213	41	1675	800
897	33	306	49	1388	774

5. Methods

5.1. Truncated GADA measurement and sample positivity

The data in this investigation was gathered using the PEDIA methods of $tGADA_{96–585}$ measurement. The research group based their radioimmunoassay (RIA) methodology on procedures presented by Bonifacio et al. (1995) and since modified by Savola et al. (1998). The methodology used by PEDIA differed from the work of Bonifacio et al. and Savola et al. by the used antigen, GAD, missing the first 95 amino acids of the molecule. (PEDIA, 2020)

Figure 2 describes the RIA process used to test for $tGADA_{96-585}$ in more detail. Serum containing $tGADA_{96-585}$ was incubated with ^{35}S-methionine labelled (cat no NEG709A, Perkin Elmer, USA) $tGAD_{96-585}$, its respective antigen (PEDIA, 2020). The formed antibody-antigen complexes were isolated with Protein A-Sepharose CL-4B® (Cat 17-0963, Amersham Biosciences, Uppsala, Sweden), after which unbound molecules were washed from the assay wells. The radioactivity (CPM) of the remaining complexes were measured by Microbeta Trilux 1450 or Microbeta^2 2450 (Perkin Elmer, USA). These were then passed on to MultiCalc^{TM}, which returned autoantibody concentration values modeled with 4PL regression methods (RU). Details of this mathematical processing are presented in the following section.



1. Serum with autoantibodies against $tGAD_{96-585}$
2. Well with ^{35}S-labelled $tGAD_{96-585}$
3. Formation of Ab-Ag complexes
4. Isolating Ab-Ag complexes with Protein A-Sepharose®
5. Washing out unbound Ags and Protein A-Sepharose®
6. Measuring radioactivity of the complexes (CPM)

Figure 2: **First steps of the radioimmunoassay**. ^{35}S is the radioactive label used. Ab, and Ag are the autoantibodies and its antigens, respectively. CPM is the response unit, counts per minute. Image copyright of Docent Heli Siljander, M.D., Ph.D.

To determine sample positivity, calculated sample concentrations were compared to a universal cut-off value. For tGADA$_{96\text{-}585}$, this value was 5.36 RU. Although unclear who first reported this value, it was defined on several accounts as the 99th percentile concentration of 373 non-diabetic adolescents and adults. (Hoppu et al., 2004; Savola et al., 1998). All concentrations exceeding 5.36 RU were said to be tGADA$_{96\text{-}585}$ positive. All autoantibody concentrations between the 97th (3 RU) and 99.5th (8 RU) percentile were re-assessed until autoantibody levels could be confirmed. (PEDIA, 2020; Siljander et al., 2007) This concentration range was also used as the basis for the division of results into analysis cohorts, a process further explained in Section 6.2.

5.2. The ”*2+2 regression*” algorithm

The following algorithm formed the core of the modeling methods used in both algorithms. As ”*2+2 regression*” was the mathematical procedure used by MultiCalc™ to analyze tGADA$_{96\text{-}585}$, the Python algorithm was built to follow details given in the manual as closely as possible (Wallac Oy, 1999). The algorithm is presented in detail below. Briefly, estimation was split into pairwise analysis, where two of the four parameters were evaluated at a time while keeping the other pair constant. The process repeated four steps: calculating α and β, calculating the model WLSE, calculating A and D using α and β from the previous step, and evaluation. Execution-related modifications required for the functionality of the Python program are discussed in Section 6.1, as they are more closely associated to the pre-processing and evaluation of the results data.

5.2.1 Calculating α and β

The algorithm was initiated by estimating the values of A and D and treating them as constants throughout the first phase of the process. This allowed for rearranging equation (2) in the following way:

$$logit\left(\frac{y-D}{A-D}\right) = \alpha + \beta \ln x \tag{8}$$

where the logit-operator is defined as (Cramer, 2003):

$$logit(p) = \ln \frac{p}{1-p} \tag{9}$$

for an arbitrary p. Equation (8) is a linear expression of the form $y' = ax' + b$ where $y' = logit\left(\frac{y-D}{A-D}\right)$, $a = \beta$, $x' = \ln x$, and $b = \alpha$, meaning that coefficients α and β were calculated using *weighted linear logit-log regression* (Wallac Oy, 1999). Weights for this step were defined as

$$w_i = \frac{R_i^2(1-R_i)^2}{\Delta^2 R_i} \tag{10}$$

where $R_i = \frac{y-D}{A-D}$ (Wallac Oy, 1999) and $\Delta^2 R_i$ was interpreted as the model squared error of an observed R_i value. After weighting was applied, WLSE was calculated using the equation (7) outlined in Section 3.3. This value was stored for future comparison.

5.2.2 Calculating A and D using α and β

As values for α and β have been derived in Section 5.2.1, equation (2) was interpreted as linear, where $a = A - D$, $x' = \frac{1}{1+e^{-\alpha-\beta\ln x}}$, and $b = D$. Weighted linear regression was once again justified as means to estimate A and D. Weights were calculated from "errors in the measured response" (Wallac Oy, 1999) by using the formula:

$$w_i = \frac{1}{\Delta^2 y_i} \tag{11}$$

and it was stated that $\sum_i w_i = n$ (Wallac Oy, 1999), where n is the number of observations.

5.2.3 Evaluation

Using A and D values calculated in the previous section, α and β were once more estimated (Section 5.2.1) to form a more precise estimation for true parameter values. WLSE was also calculated again. The model was then evaluated, comparing WLSE with the value calculated in Section 5.2.1. Iteration was stopped if at least one of following three conditions was met:

1. Residuals are divergent, meaning that WLSE for the current iteration was larger than WLSE for the parameters of the previous estimation for α and β.
2. 150 iterations have been completed.
3. Convergence rate (CR) < 0.02% for the first 10 iterations, CR < 0.1% for 10-50, or CR < 0.5% for the final iterations.

Here CR was defined as

$$CR = \frac{|c_{i+1}-c_i|}{c_i} * 100\% \tag{12}$$

or the percentage difference in successive WLSE calculations. This definition differed slightly from the conventional (Gradinaru et al. 2013) notation of a converge rate but was more relevant in relation to the algorithm evaluation. If none of the conditions were met, the process was reinitiated with results from the iteration as the initial estimates. If, on the other hand, iteration was stopped, final values for B and C were converted from α and β by using the formulas $B = -\beta$ and $C = e^{\frac{\alpha}{B}}$. (Wallac Oy, 1999)

5.3. Characteristic measures for model evaluation

The following section outlines the characteristic values used in the evaluation of Python model accuracy. For both the training and the test data, a single measure was defined to assess accuracy. Implementation of these measures is further described in Section 6.2. In addition, probabilities of certain diagnostic events were calculated, but since they were based on distribution types fit for test data, they are also explained in conjunction to the results.

5.3.1 Training data

The goodness of fit of the standard curve was evaluated by the average error in recovery (MER), which was defined as

$$MER = \frac{\sum_i |1 - R_{\%,i}|}{n} \tag{13}$$

In other words, MER was the average of the absolute relative deviations from the expected standard concentration. MER was based on “standards recovery” (Ederveen, 2010), which compared the concentration calculated by the model to the actual standard concentration. The ratio of these two values gave the characteristic recovery rate ($R_{\%}$):

$$R_{\%} = \frac{c_{obs}}{c_{exp}} \times 100\% \tag{14}$$

where the calculated standard values were noted by c_{obs}, and the known standard concentrations were c_{exp}. (Ederveen, 2010; Davis et al., 2000).

MER was used as a single accuracy measure, meaning that all recovery errors for all assays were evaluated in the MER calculation. The need for deviation evaluation was due to low and high recovery rates otherwise resulting in zero-sum averages. Furthermore, the mean of the absolute error was chosen for value interpretation, since mean percentage errors were easier to interpret than mean squared percentage errors.

5.3.2 Test data

The test set results were assessed based on the relative difference between MultiCalc^{TM} and Python concentrations. This was given by the error ϵ defined as

$$\epsilon_i = \frac{c_{alg,i} - c_{mult,i}}{c_{mult,i}} \tag{15}$$

where $c_{alg,i}$, $c_{mult,i}$ are the individual tGADA$_{96\text{-}585}$ concentrations calculated by each method for each response observation i. Since MultiCalc^{TM} results were a benchmark that the Python program was trying to replicate, MultiCalc^{TM} measurements were interpreted as the "actual" observations. The comparison of relative errors assumed heteroscedasticity among the responses. Since it was believed that error noise increased with response (Gottschalk and Dunn, 2005), it would be expected that relative error would stay constant throughout. As mentioned in the introduction to this sub-section, observations were then visualized for probability distribution estimations.

6. Results

Due to incomplete knowledge of certain "*2+2 regression*" algorithm procedures, modifications to the Python program had to be implemented. These included the initial estimate, outlier management, and defining weights for WLSE calculations. Solutions to these issues are presented as preliminary results before MER or ϵ data.

6.1. Algorithm modifications

6.1.1 Initial estimates

Since the algorithm was initial estimate dependent, initialization became essential for model accuracy. Moreover, results varied significantly depending on the initial estimate given (Figure 3), highlighting the role of initial estimates for accurate and consistent results.

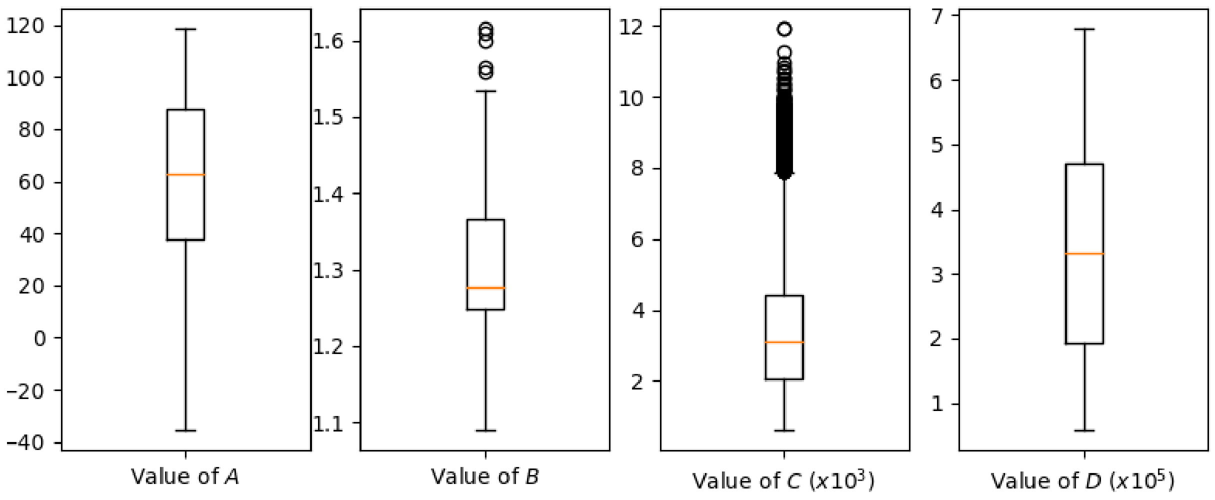


Figure 3: Initial estimate variation. Figure shows boxplots of all four curve parameters as results of 10000 separate iterations. Boxplots highlights, how given the same standards input data, the algorithm returns different values depending on the initial estimates of the algorithm.

Defining initial estimates proved challenging as instructions regarding this process were absent in the MultiCalc^{TM} manual. As a solution, the relationship between curve parameters and the standard solution data was used. It was found (Figure 4), that the first standard response was moderately ($R^2 = 0.67$) correlated with the value of A for its respective 4PL fit. Similarly, the ratio between the value of D and the response of the largest standard concentration was concentrated around 10 and 100. These values were used in conjunction to the confidence interval for A to form uniform distributions from which 1000 initial estimates were sampled and analyzed. From these, the parameters of the model with the best accuracy was chosen.

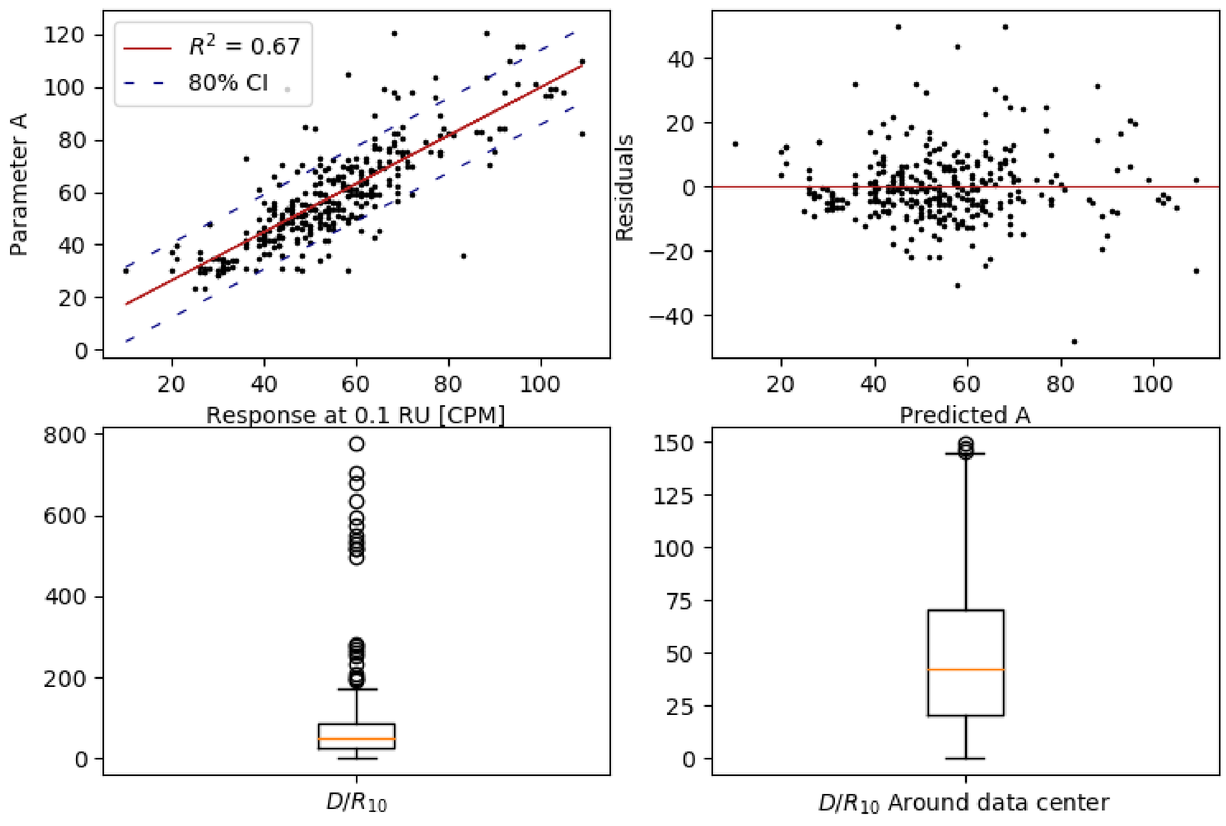


Figure 4: Definition of *A* and *D*. *Upper left:* Linear relationship between the response of the first standard, 0.1 RU (x-axis), and the value of the parameter *A* modeled for that assay by MultiCalc^{TM} (y-axis). The legend shows the Pearson correlation coefficient between the two variables. The dashed lines represent the upper and lower bounds of the 80% confidence interval of the gradient fit. *Upper right:* residual plot the predicted *A* values justifying ordinary linear regression. *Bottom*: illustrations of the ratio of the value of *D* and the response at the last standard, 139.9 RU. The right-hand side boxplot has outliers removed for better interpretation of the distribution.

6.1.2 Outliers

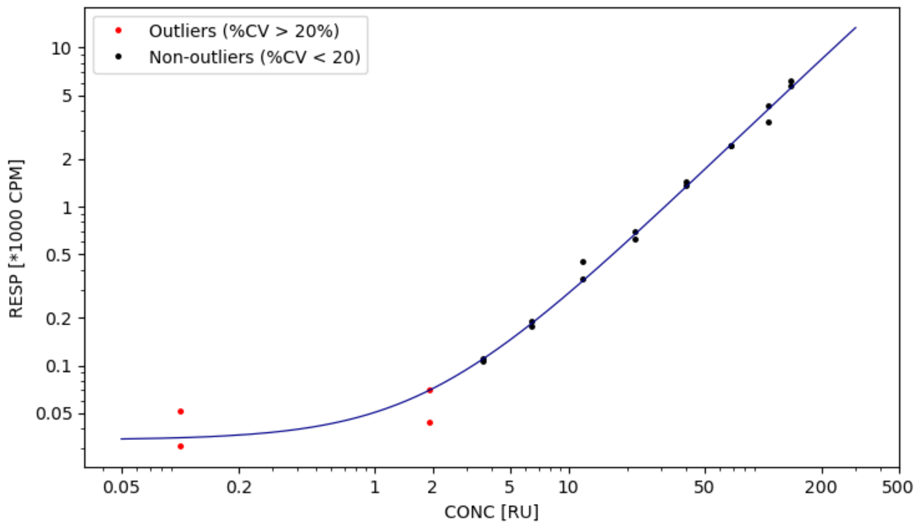


Figure 5: Graph outlier identification. Graph base is same as in Fig. 1 with CPM response divided by 1000 on the logarithmic y-axis and the RU concentration on the logarithmic x-axis. The legend in the upper left corner shows the two types of data points in the image: outlier pairs are colored in red. Non-outliers are black data points. The curve shown is MultiCalc^{TM} fit.

Due to biological noise in the assay as well as human error in sample processing, outliers in standard concentration data needed to be discarded to ensure accurate curve fit. They were defined by the coefficient of variation ($CV_{\%}$) (Ederveen, 2010)

$$CV_{\%} = \frac{\sigma}{\mu} \times 100\% \tag{16}$$

where σ is the sample standard deviation and μ is its mean. A pair of observations with $CV_{\%}$ exceeding 20% were rejected as outliers (Figure 5) (Ederveen, 2010). Although not used in model fit, the outliers were included in model validation analyses. This was since it was believed that they would not affect model validation calculations as both methods would achieve approximately same concentrations and be able to recognize outliers from the data.

6.1.3 Defining weight vectors for WLSE calculations

As defined in Section 5.2, WLSE calculations were used in the algorithm iterations to determine whether acceptable accuracy had been achieved. Without information about how to best define $\boldsymbol{w}$ for this calculation, I defined $\boldsymbol{w}$ as

$$w_i = k \frac{1}{resp_i^2} \tag{17}$$

or the inverse of the squared response multiplied with a normalizing constant

$$k = \sum_i resp_i^2 \tag{18}$$

This was an intuitive guess based on the heteroscedastic nature of the data (Section 3.3) and the form weight equation (11) (as it was also an inverse squared function).

6.2. Preliminary results

The difference in MER for the Python algorithm (23.93%) and MultiCalc^{TM} (23.26%) software was 0.7%. The similarity in values suggests that the Python algorithm successfully replicated MultiCalc^{TM}, and that much of the error in MER could be because of the biological noise of the assay. In addition, it was known that for accurate models, standard recovery rates fall between 70%-130% (Davis et al., 2000) or 80%-120% (Ederveen, 2010). As MER falls approximately within these parameters, it may be concluded that MER indicates of an accurate Python algorithm.

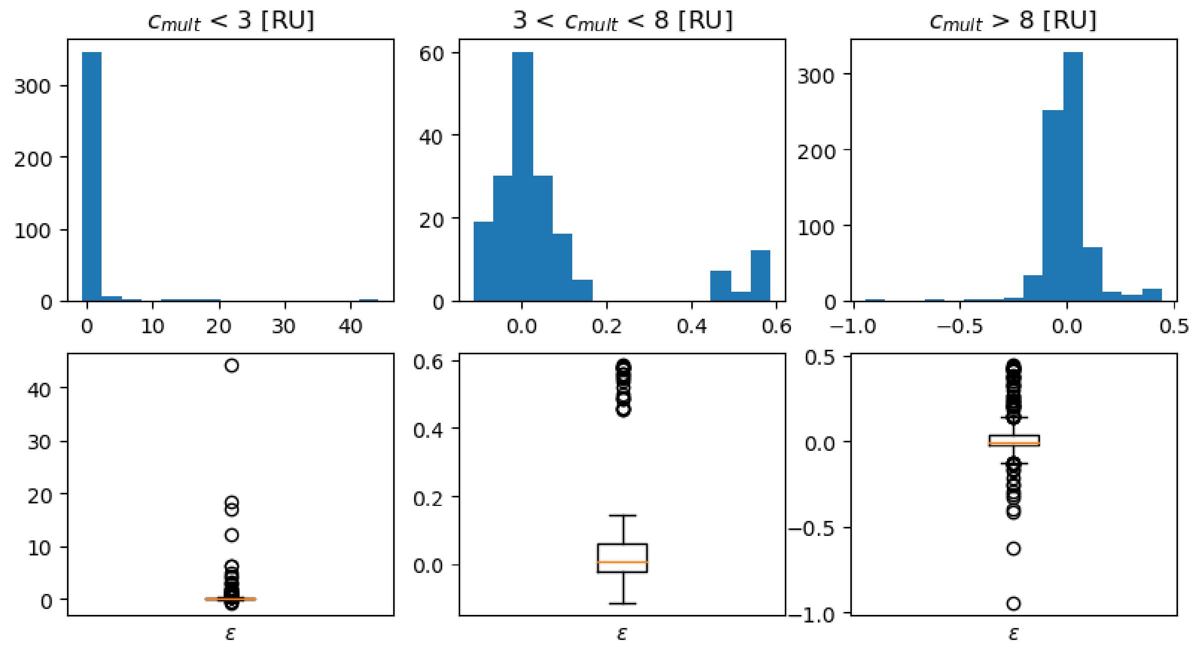


Figure 6: Unprocessed ϵ distributions for the three cohorts. The titles for the three colums distinct the respective cohorts (low, moderate, high). On the left-hand side, that severe outliers distort the low concentration distribution. Moderate concentrations are perturbed by a secondary cluster. For high concentrations there is a lagging tail, seen on the right columns.

As mentioned in Section 5.1, the ϵ was split into three cohorts based on re-analysis: low (0-3 RU), moderate (3-8 RU), and high concentration (8RU -). Preliminary visualizations (Figure 6) showed that although perturbed with noise, observations seemed to come from a normal distribution. However, outliers had to be heavily discarded (Figure 7) for the normality assumption to be justified. Outlier management allowed for hypothesis testing for normality, where the Shapiro-Wilks test was chosen as the recommended choice (Ghasemi and Zahediasl, 2012). Although this process might not have been statistically justified, data idealization was performed to get an approximate understanding of how the error behaves around the cut-off value. This approximation would then help assess, whether algorithm inaccuracy is too high for the program to be accepted. Furthermore, small sample sizes (Table 4) were used to justify that even with an ideal sample, resulting distributions would be approximations due to sample size. Distribution estimation is discussed in more detail in Section 7.1.

Table 4: Test data results. Each row shows the results values – sample size, mean, standard deviation, and normality test results – analyzed separately for each concentration cohort. The column on the right shows results for the Shapiro-Wilks normality test.

Concentration (RU)	Rejected/sample size (%)	Mean; Standard Deviation	Skewness; Kurtosis	Shapiro-Wilks test-statistic, (p-value)
0-3	79/364 (22%)	0.068; 0.091	0.342; 3.063	0.982 (1.07e-03)
3-8	29/181(16%)	0.011; 0.051	0.275; 3.044	0.973 (4.24e-03)
8-	45/729 (6%)	-0.001; 0.056	0.060; 3.369	0.972 (3.40e-10)

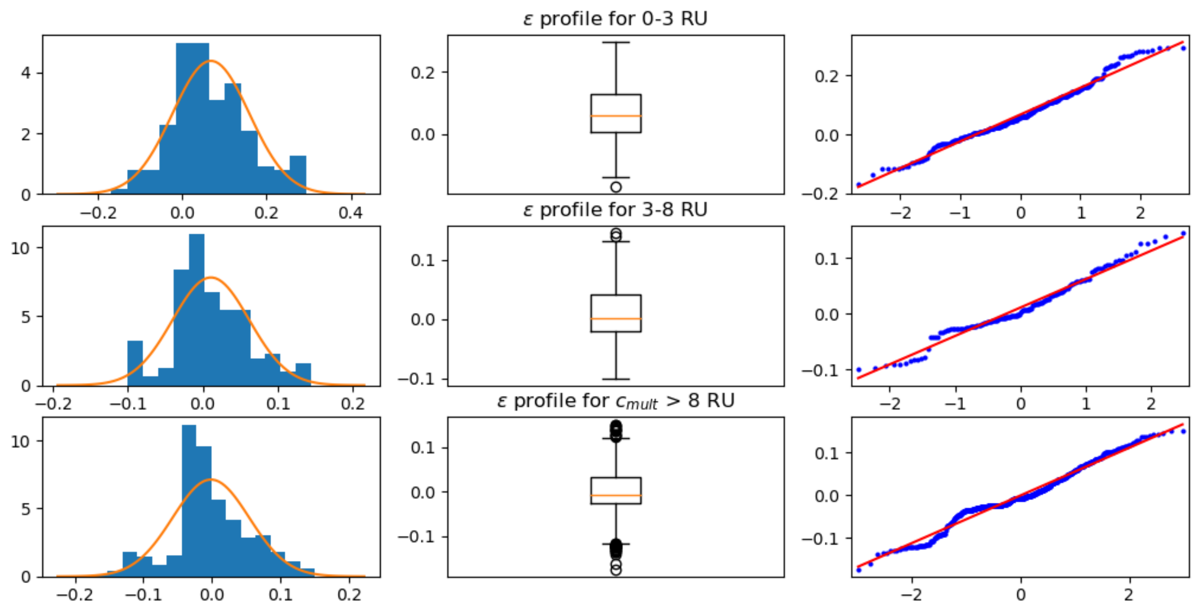


Figure 7: Error Profiles. Idealized ϵ cohorts are visualized above. The figures on the left are density histograms with fitted normal curves (parameters from Table 4). The second shows boxplots, and the Q-Q plot shows how the data follows a theoretical normal distribution. All figures show that the data roughly follows a normal distribution.

Processed test data results are summarized above, with Figure 7 showing that sample distributions are approximately normally distributed. Normality of the distributions was supported by skewness and kurtosis (Table 4), which were close to their respective reference values of zero and three (Bulmer, 1979). However, observations of normality are not supported by the SW test, which rejected normality for all concentration ranges at the 1% significance level. This rejection may have been due to small sample size approximations as previously discussed. Furthermore, as visual tests and characteristic parameters supported approximate normality, SW test results were ignored, and normality was accepted. This acceptance of normality was important in estimating the magnitude of misclassification probabilities, helping in risk assessment of algorithm implementation. Generally, all distributions had relatively similar standard deviations, suggesting that relative error was moderately constant. For the first two cohorts, the slight positivity of the mean suggested, that the Python algorithm may be skewed towards estimating slightly larger concentration values.

6.3. Probability evaluation of sample positivity classification

The normal distributions (previously defined) were utilized for calculating likelihoods for targeted diagnostic events. That is, I was interested in a probability expressed by

$$P\left(c_{alg} = x \middle| c_{mult} = y\right) \tag{19}$$

which was read as the probability that the Python algorithm calculates a concentration x given that the "actual" MultiCalc^{TM} concentration is y. The value of c_{alg} can be defined with ϵ and c_{mult}, since equation (15) can be rearranged to

$$c_{alg} = c_{mult}(\epsilon + 1) \tag{20}$$

Hence if $c_{alg} = x$, then (19) is

$$\therefore P(c_{mult}(\epsilon + 1) = x | c_{mult} = y) \tag{21}$$

and this can be rearranged to

$$P\left(\epsilon = \frac{x}{c_{mult}} - 1 \middle| c_{mult} = y\right) \tag{22}$$

In other words, Python algorithm events could be calculated from conditional ϵ distributions. The important probabilities were related to two questions: a) whether concentrations in the low and high cohorts have significant likelihood of crossing over into the reassessment range with the Python algorithm ; and b) whether the accuracy of the algorithm is high enough to ensure that the likelihood of false positivity classifications remains low (under 5%) within the retesting cohort (3-8 RU). If the a) holds true, retesting ranges can be adjusted accordingly, but if the latter b) fails, future implementation of the current Python algorithm cannot be recommended. Table 5 below illustrates the aforementioned diagnostic events in terms of (22) and the probability expressions that defined them.

Table 5: Important Diagnostic Events. The table below describes diagnostically important events in terms of the probability variables defined above.

Description	c_{mult}	c_{alg}
Low concentration observed in 3-8 RU	$y < 3$	$3 < x < 8$
High concentration observed in 3-8 RU	$y > 8$	$3 < x < 8$
False positive result	$y < 5.36$	$x > 5.36$
False negative result	$y > 5.36$	$x < 5.36$

Figure 8 evaluates the probability of the Python algorithm concentrations crossing over into the 3-8 RU threshold. Even though there was an increased likelihood (greater than 5%) of cross-over concentrations occurring for 2.46-2.99 RU and 8.01-8.82 RU, the risk of mislabeling positivity in these ranges was insignificant (of order 10^{-15} and 10^{-9}, respectively). It may be concluded that the accuracy of the model at these ranges is satisfactory enough to proceed with the re-test ranges previously set for analysis.

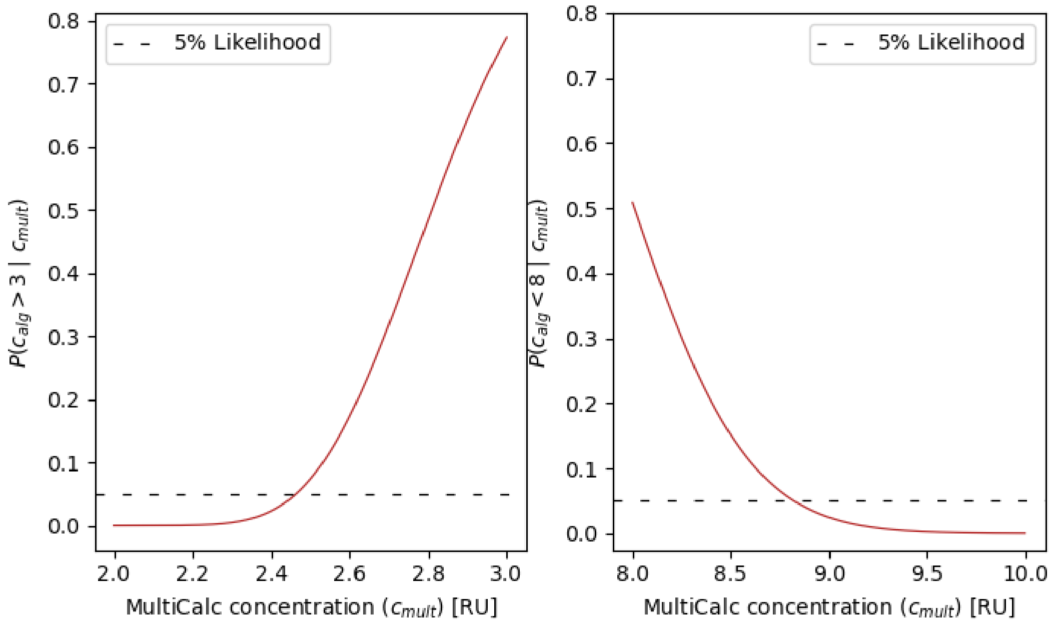


Figure 8: Likelihood of crossing over. The figures estimate the probability that the Python algorithm calculates a concentration that crosses the re-test threshold (3 or 8 RU, respectively), given a certain MultiCalc^{TM} value. The dashed line indicates heightened (over 5%) risk of this occurring. Likelihoods were calculated with cumulative probability distribution functions based on (22). The complementary probability was used to calculate values exceeding a threshold (e.g. 3 RU)

In the critical 3-8 RU range, it was found that likelihood of misdiagnosis was significant (more than 5%) between 4.90-5.78 RU (Figure 9). In this area, the likelihood of overestimation was higher due to the slight positive mean of the error distribution. Within this range, allocating additional replicates for reassessment might mitigate the effects of the error and facilitate establishing a precise estimate for the concentration. Additionally, the positive relationship between the two concentrations is highlighted in Figure 10. This suggests that the Python concentrations could be used, with confidence (unweighted correlation $R^2 = 0.96$), to adjust Python algorithm results in a way that they follow MultiCalc^{TM} results more closely.

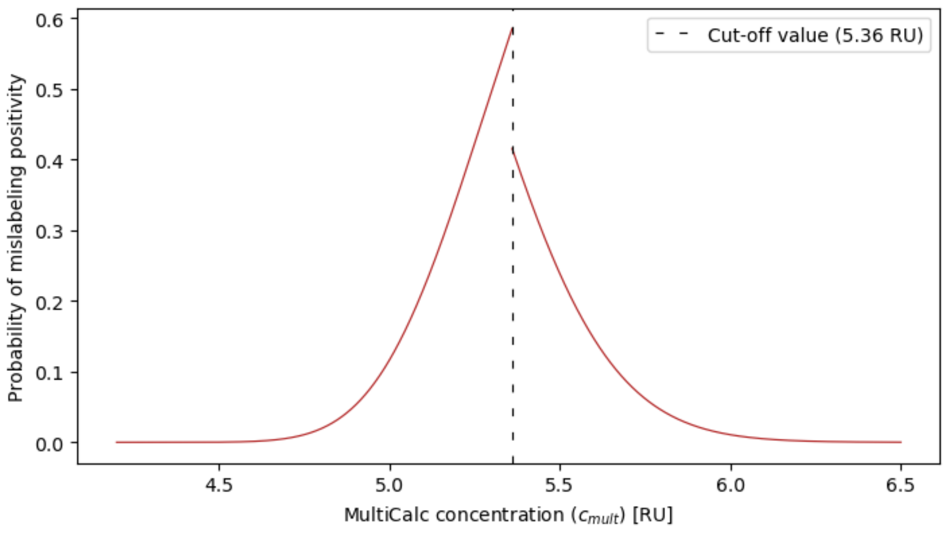


Figure 9: Miscategorization probability. Figure shows the likelihood of Python algorithm calculating mislabelled values (over 5.36 RU when not and vice versa) as a function of MultiCalc™ concentration (c_{mult}). The positivity cut-off value is visualized by the vertical dashed line.

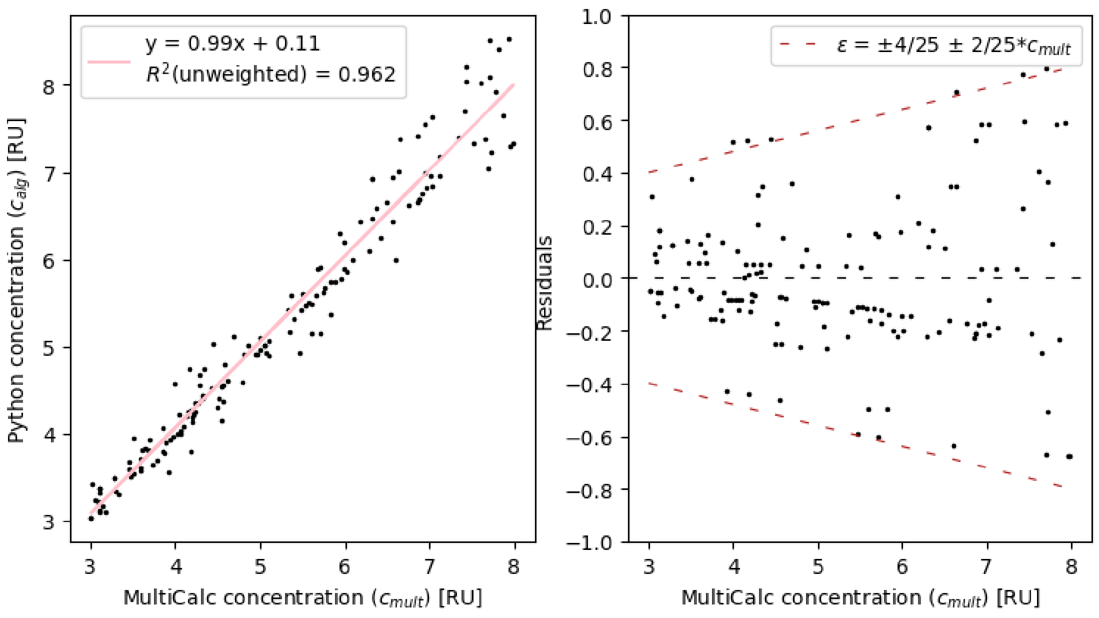


Figure 10: Linear relationship between method concentrations. Relationship between the concentrations calculated by Python (y-axis) and MultiCalc (x-axis). The linear relationship between concentrations calculated by MultiCalc™ (x-axis) and Python (y-axis) is highlighted by the linear regression line in pink. The legend for the figure on the left displays the equation of the weighted linear regression line and the unweighted correlation the upper-left corner displays the correlation coefficient. The residual model (legend on right-hand figure) was used as weights for regression calculations.

7. Discussion

Overall, the algorithm was able to replicate the MultiCalc^{TM} algorithm quite accurately and develop applicable 4PL curve parameters used for sample concentration calculations. However, several procedural aspects needed to be discussed in further detail. These topics were identified as either sources of error, explanations to sources of error, or points of improvement for algorithm functionality. This includes sample size, iteration of initial estimates, and model validation measures.

7.1. Sample size

The generous elimination of observations (Section 6.2) meant that these results need to be interpreted with criticism. The aggressive remodeling of data may have been motivated by confirmation bias, and findings could have been on data that was forced to fit a certain distribution for the sake of relevancy. This issue of fitting distributions to small sample sizes is highlighted by Cochran's sample size formula below. For samples drawn from normally distributed populations, the sample size requirements for accurate approximation of the mean and standard deviation can be calculated by (Lehmann et al., 2013)

$$n > \left(\frac{z_{\frac{\alpha}{2}}}{d}\right)^2 \sigma^2 \tag{23}$$

Where $z_{\alpha/2}$ is the z-score at confidence level α, d is the maximum allowed absolute error and σ^2 is the population variance. For instance, assuming that the sample was drawn from a distribution fitted for 3-8 RU in Section 6.2, the sample size needed to estimate relative error mean $\bar{\epsilon}$ to ±5% accuracy ($d = \pm 5\%$ of 0.011) would be at least

$$n > \left(\frac{1.96}{0.0109\ldots * 0.1}\right)^2 * 0.0511\ldots^2 \approx 8380 \tag{24}$$

This value, in comparison to the sample size in the re-assessment cohort (n = 181), highlights the approximative issues presented in Section 6.2. A larger sample of 1000, for instance, could increase confidence in distribution findings and allow for adjusting the algorithm in a way that it more appropriately follows MultiCalc^{TM} results. Furthermore, a similar sample size estimation could have been calculated beforehand, giving an understanding of how large of a sample could be needed to achieve a certain accuracy in measurements.

7.2. Iterative nature of the estimation

The initial estimate dependency of the algorithm was solved in Section 6.1.1 by iterating 1000 samples based on the relationship between standards data and parameters estimated by MultiCalc^{TM}. However, this was inefficient methodology, as it consumed significant computational power and took time to execute. Due to the high number of assay analyses completed in a laboratory, this time loss can accumulate significantly over time. To increase efficiency, the use of estimate sampling should be avoided.

Avoiding estimate sampling could be done with backtrack evaluation of the Python initial estimates. Although not substantiated with any modeling evidence, I believe associations could be formed between the initial estimate for best model fit and its standard concentration data. Since there was a linear relationship between the parameter value A and the first standard responses (Figure 4), it is likely that there exists a way of modeling a good initial estimate based on given standard data. By observing how the best initial estimate relates to input data, an initial estimate model could be developed. This model could then surpass the need for large estimate sample iteration. The model could be built by collecting the initial estimate information for every standard curve fit and using that data to model appropriate relationships. On the other hand, this again introduces the efficiency problem previously discussed: introducing a model to help estimate another model increases computational complexity, accumulating excess time loss over time.

7.3. Model Validation

7.1.1 Weights in Least Squares Calculations

As outlined in Section 6.1.3, WLSE evaluations were done with an intuitively defined weighting vector, equation (17). At the time of definition, it was unknown whether this belief of error weighting represented universal findings. However, subsequent review revealed that the literature did support this intuition, as Gottschalk and Dunn (2005) suggested the weighting vector could be related to the error variance by

$$w_i = \frac{1}{\sigma^2} \tag{25}$$

where variance could be defined by the power function

$$\sigma^2 = A(response)^B \tag{26}$$

where “A is a function of the magnitudes of the responses and the average noise level” (Gottschalk and Dunn, 2005). They define B as a constant, most commonly 1.2 – 2.0 for immunoassays. (Gottschalk and Dunn, 2005) This shows that equations (25) and (17) are similar in form: both applied the squared inverse of some form of a response function. This variance relation is supported by Davidian et al. (1988), who present a similar form of equation (26) as a common form of variance function in immunoassays. Additionally, Sadler (2008; 2015) discusses other power function forms (parabolic, three-parameter) as methods to model error variance. These alternative weighting solutions could be explored to increase model accuracy. Applying a fitted variance profile to the WLSE equation (7) could allow for more representative model validation calculations.

7.1.2 Mean Error in Recovery

While the mean absolute error (MAE) in recovery was used in the algorithm (MER), root mean squared error emerged as a more appropriate measure for model accuracy. In their paper, Chai and Draxler (2014) suggest that even though both metrics are “*widely used in model evaluations*”, they are useful for their respective error distributions. They remind, that choosing the right statistical measure is important due to the condensing of information occurring in calculating these values. While MAE is more suitable for a uniform distribution of errors, RMSE assumes a Gaussian error profile and is more heavily affected significant error observations. (Chai and Draxler, 2014) Figure 7 (Section 6.2) reflects a normal error distribution, suggesting that RMSE may be a more appropriate measure for model fit.

7.1.3 Assessment of test set data

The measure for goodness of fit of the test data, ϵ, was relative to MultiCalc^{TM} results. This meant that the quality of the error measure ϵ was dependent on the MultiCalc^{TM} standard curve fit. Additionally, due to the lack of analysis of MultiCalc^{TM} accuracy, ϵ was vulnerable to comparison of concentrations calculated with a poor fit, such as one in Figure 11. This indicated that there is a risk of determining a poor Python fit based on large ϵ, even though the standard curve fit would have been accurate.

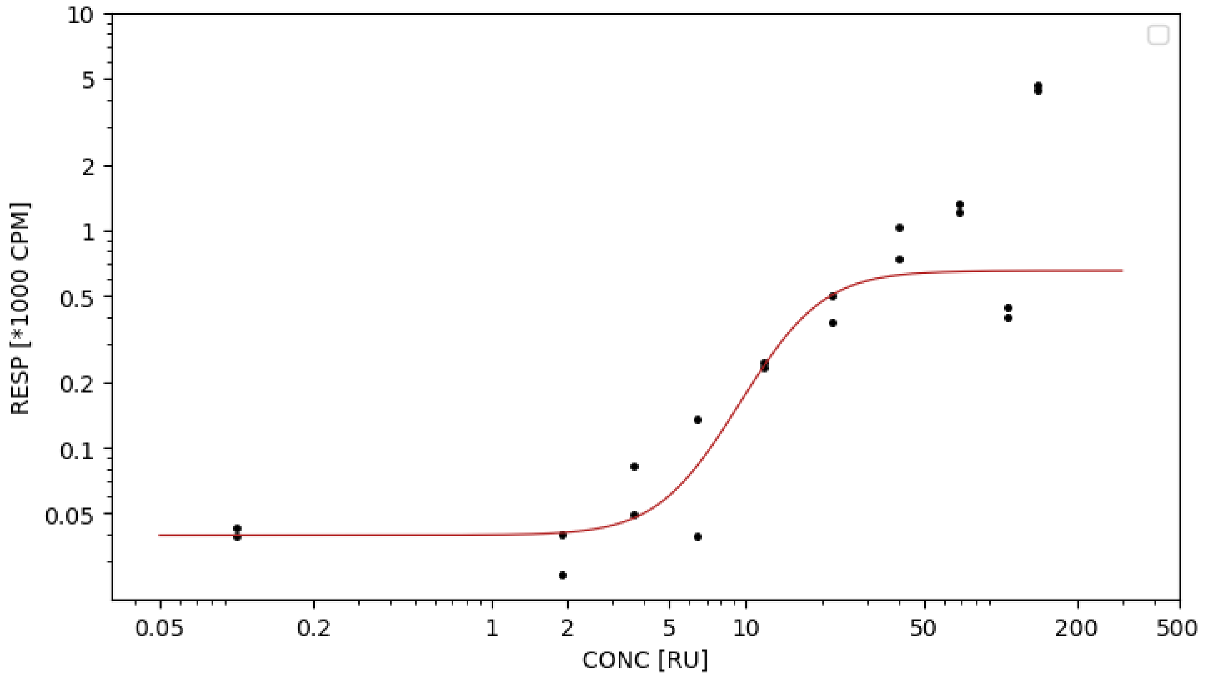


Figure 11: Example of poor MultiCalc™ fit. The regression curve has an upper tail that twists downwards approaching the outlier pair at about 700 CPM. Visual evaluation shows that the trendline should be increasing linearly (data is shaped that way), suggesting that this is a poor standard curve fit.

This uncertainty suggests that MER and ϵ were not sufficient parameters to measure goodness of fit, and supplementary evaluation is needed. Gottschalk and Dunn (2005) used a chi-squared (χ^2) test of residual variances, where residual variance was defined as the model WLSE divided by the degrees of freedom. The degrees of freedom were defined as the number of data points subtracted by the amount of curve parameters. It could be shown (Gottschalk and Dunn, 2005), that residual variance follows a χ^2 distribution, and this distribution could be used to calculate the likelihood that the model would return a worse fit under the same conditions. Applying the χ^2-test as an additional goodness of fit metric could further improve evaluations by introducing a likelihood that a better model could be fit on the assay. This could have a profound impact on identifying and discarding poor assays from evaluation. Discarding poor assays could help remove noise from ϵ observations.

However, introducing the χ^2- test may not be relevant. As mentioned in the PEDIA procedure document (2020), standard curves with more than seven dilution points (observation pairs) were accepted for MultiCalc™ processes. These results were then further assessed and accepted by supervisors. Similar inclusion criteria were not used in the Python algorithm evaluation, meaning that assays that could have been rejected were included in evaluation. Hence, while it cannot be confirmed that it was the source of noise in test data results, I believe that it is likely that the outliers and perturbed data points in the test data were a result of comparison to assays that meet rejection criteria. Since these assays should not have been considered anyway, the model validation methods used to assess the Python program were more applicable than previously discussed.

7.4. 4PL vs 5PL

Although 4PL methods can accurately fit tGADA$_{96–585}$ assay data, Gottschalk and Dunn show (2005) that model accuracy can be improved by using the five-parameter logistic (5PL) model for asymmetric data. Additionally, Cumberland et al. (2015) suggest 5PL be used as the default model for immunoassay data. According to Gottschalk and Dunn (2005), the use of 5PL may be justified, since a significant portion of immunoassay measurements have skewness associated with 4PL accuracy loss (Gottschalk and Dunn, 2005). Both papers agree, however, that for concentrations falling into the lower-end, flat part of the curve, 4PL is a better estimator.

However it is appropriate to believe that using 5PL would not significantly increase model accuracy and reliability. This is due to the nature of the data as well as the instructions of the MultiCalc^{TM} manual. For tGADA$_{96–585}$ data used in the paper, asymmetricity cannot be concluded because of the lack of the full curve shape (see Figure 1). This is supported by the estimated point of inflexion, which often fell outside of the graph range (boxplot in Figure 3 is of order 10^3). Hence, as it is not possible to determine asymmetricity, perhaps it is reasonable to assume, that the standard data is symmetric. This suggests that the use of 5PL is not relevant. Furthermore, MultiCalc^{TM} does not specify reasons for using 4PL instead of 5PL for this assay. This leads me to believe that the most replicable results are achieved by using 4PL.

8. Conclusion

Overall, the Python algorithm is an employable and accurate software, capable of replicating MultiCalc^{TM} results to a degree of accuracy. Its 0.7% deviation in MER and near-zero relative error for the test data suggest that as it stands, the program functions as required: it can accurately fit standard curves and calculate concentrations of the samples present in the assay. As with all investigations, sources of error and improvement were identified. These were initial estimate development, weighting vector definition, and model validation measures. Variance functions were proposed as a solution for weighting in model validation, as were RMSE and the χ^2-test for goodness of fit measures. The problem of developing an accurate initial estimate remains somewhat unresolved. These improvements, in combination with increasing sample size, show promise of how the algorithm can be adjusted to make it a viable replacement for the MultiCalc^{TM} software in the future.

9. References

1. Boldison, J., Wong, F.S. (2016). *Immune and pancreatic cell interactions in β type 1 diabetes*. Trends in Endocrinology & Metabolism. Vol 27:12. pp.856-867. [Cited 25.2.2020]. DOI: 10.1016/j.tem.2016.08.007.
2. Siljander, H., Honkanen, J. and Knip, M. (2019). *Microbiome and type 1 diabetes.* EBioMedicine.Vol 46. pp.512-521. [Cited 25.2.2020]. DOI: https://doi.org/10.1016/j.ebiom.2019.06.031.
3. Harjutsalo, V., Sund, R., Knip, M., Groop, P. (2013). *Incidence of Type 1 Diabetes in Finland.* JAMA. Vol 24:31. pp.427–428. [Cited 25.2.2020]. DOI: 10.1001/jama.2013.8399.
4. Elkon, K. and Casali, P. (2008). *Nature and functions of autoantibodies*. Nature Clinical Practice Rheumatology. Vol 4:9, pp.491-498. [Cited 29.4.2020]. DOI: 10.1038/ncprheum0895
5. Kallionpää, H., Somani, J. et al. (2019). *Early Detection of Peripheral Blood Cell Signature in Children Developing β-Cell Autoimmunity at a Young Age*. Diabetes. Vol 68:10. pp.2024-2034. [Cited 25.2.2020]. DOI: https://doi.org/10.2337/db19-0287
6. Zaidi, P., Kama, S. (1993). *Radioimmunoassay: Principle and Technique.* JPMA. Vol 43:12. pp.264-267. [Cited 25.2.2020]. Available: https://jpma.org.pk/PdfDownload/ 5022
7. Dudley, R.A., Edwards, P., Ekins, R.P., Finney, D.J., McKenzie, I.G.M., Raab, G.M., Rodbard, D. and Rodgers, R.P.C. (1985). *Guidelines for immunoassay data processing.* Clinical Chemistry. Vol 31:8, pp.1264-1271. [Cited 11.5.2020] DOI: https://doi.org/10.1093/clinchem/31.8.1264
8. PerkinElmer Life Science, Wallac Oy. (1999). *User Guide To Multicalc Functions*. Turku, Finland
9. Mendel, C.M., Mendel, D.B. (1985). *'Non-specific' binding. The problem, and a solution.* Biochemical Journal. Vol 228:1. pp.269. [Cited 15.3.2020]. DOI: https://doi.org/10.1042/bj2280269
10. Güven, E., Duus, K., Lydolph, M., Jørgensen, C., Laursen, I. and Houen, G. (2014). *Non-specific binding in solid phase immunoassays for autoantibodies correlates with inflammation markers.* Journal of Immunological Methods. Vol 403:1-2. pp.27. [Cited 15.3.2020]. DOI: https://doi.org/10.1016/j.jim.2013.11.014.
11. Mauriz, E., Calle, A., Lechuga, L., Quintana, J., Montoya, A., Manclús, J. (2006). *Real-time detection of chlorpyrifos at part per trillion levels in ground, surface and drinking water samples by a portable surface plasmon resonance immunosensor.* Analytica Chimica Acta, Vol 561:1-2, pp.40-47. [Cited 15.3.2020]. DOI: https://doi.org/10.1016/j.aca.2005.12.069.
12. Jung, A. (2019). *Machine Learning: Basic Principles.* pp.34-36.
13. Wasserman, L. (2005). *All Of Nonparametric Statistics*. New York, NY: Springer New York, pp.87. ISBN: 978-0387-25145-5

14. DiCiccio, C., Romano, J., Wolf, M. (2017). *Improving Weighted Least Squares Inference*. University of Zurich, Department of Economics, pp.2. [Cited 28.4.2020]. ISSN 1664-7041 (print). ISSN 1664-705X (online)
15. Gottschalk, P., Dunn, J. (2005). *The five-parameter logistic: A characterization and comparison with the four-parameter logistic*. Analytical Biochemistry, Vol 343:1, pp.54-65. [Cited 28.4.2020]. DOI: 10.1016/j.ab.2005.04.035
16. Bonifacio, E., Genovese, S., Braghi, S., Bazzigaluppi, E., Lampasona, V., Bingley, P., Rogge, L., Pastore, M., Bognetti, E., Bottazzo, G., Gale, E. and Bosi, E., (1995). *Islet autoantibody markers in IDDM: risk assessment strategies yielding high sensitivity*. Diabetologia. Vol 38, pp.816-822. [Cited 13.5.2020]. DOI: 10.1007/s001250050358
17. Savola, K., Sabbah, E., Kulmala, P., Vähäsalo, P., Ilonen, J. and Knip, M., 1998. Autoantibodies associated with Type I diabetes mellitus persist after diagnosis in children. Diabetologia. Vol 41:11, pp.1293-1297. [Cited 12.5.2020]. DOI: 10.1007/s001250051067
18. Pediatric Diabetes Research Group. (2020). *Determination of truncated GAD65 (aa 96-585) autoantibodies (tGADA) with radiobinding assay (RBA)*. University of Helsinki, Finland.
19. Siljander, H., Veijola, R., Reunanen, A., Virtanen, S., Åkerblom, H. and Knip, M., (2007). *Prediction of type 1 diabetes among siblings of affected children and in the general population*. Diabetologia, Vol 50:11. pp.2272-2275. [Cited 13.5.2020]. DOI: 10.1007/s00125-007-0799-5
20. Hoppu, S., Ronkainen, M.S., Kulmala, P., Åkerblom, M., Knip, M., Childhood Diabetes in Finland Study Group. (2004). *GAD65 antibody isotypes and epitope recognition during the prediabetic process in siblings of children with type I diabetes.* Clinical and Experimental Immunology, Vol 136:1. pp.120-128. [Cited 28.4.2020]. DOI: 10.1111/j.1365-2249.2004.02416.x
21. Cramer, J.S. (2003). *The Origins And Development Of The Logit Mode (Logit Models From Economics And Other Fields, Chapter 9)*. Cambridge, UK: Cambridge University Press, pp.10.
22. Gradirnaru, V., Hiptmair, R., Arnulf, J. (2013). *Numerical methods.* Swiss Federal Institute of Technology (ETH) Zürich, Switzerland. pp. 11.
23. Ederveen, J. (2010). *A Practical Approach to Biological Assay Validation*. Hoofdorp, Netherlands: Dutch Ministry of Housing, Spacial Planning, and the Environment.
24. Davis, D., Zhang, A., Etienne, C., Huang, I., Malit, M. (2000). *Principles of Curve Fitting for Multiplex Sandwich Immunoassays.* Hercules, California, USA: Bio-Rad Laboratories.
25. Ghasemi, A., Zahediasl, S., (2012). *Normality Tests for Statistical Analysis: A Guide for Non-Statisticians.* International Journal of Endocrinology and Metabolism, Vol. 2012:10. pp.486-489. [Cited 28.4.2020]. DOI: 10.5812/ijem.3505
26. Bulmer, M., (1979). *Principles of Statistics*. Edinburgh, UK: Dover Publications. pp. 58. ISBN: 9780486135205
27. Lehmann, N., Finger, R., Klein, T. and Calanca, P. (2013). *Sample Size Requirements for Assessing Statistical Moments of Simulated Crop Yield Distributions*. Agriculture. Vol 3:2, pp.210-220. [Cited 28.4.2020]. DOI: 10.3390/agriculture3020210

28. Davidian, M., Carroll, R., Smith, W., (1988). *Variance functions and the minimum detectable concentration in assays*. Biometrika. Vol 75:3, pp.549-556. [Cited 28.4.2020]. DOI: https://doi.org/10.1093/biomet/75.3.549
29. Sadler, W. (2008). Error models for immunoassays. Annals of Clinical Biochemistry. Vol 45:5, pp.481-485. [Cited 28.4.2020]. DOI: 10.1258/acb.2008.007230
30. Sadler, W. (2015). Using the variance function to estimate limit of blank, limit of detection and their confidence intervals. Annals of Clinical Biochemistry. Vol 53:1, pp.141-149. [Cited 28.4.2020]. DOI: https://doi.org/10.1177/0004563215575560
31. Chai, T., Draxler, R. (2014). *Root mean square error (RMSE) or mean absolute error (MAE)? – Arguments against avoiding RMSE in the literature*. Geoscientific Model Development. Vol 7:3, pp.1247-1250. [Cited 28.4.2020]. DOI: 10.5194/gmd-7-1247-2014
32. Cumberland, W., Fong, Y., Yu, X., Defawe, O., Frahm, N. and De Rosa, S. (2015). *Nonlinear Calibration Model Choice between the Four and Five-Parameter Logistic Models*. Journal of Biopharmaceutical Statistics, Vol 25:5, pp.972-983. [Cited 28.4.2020]. DOI: 10.1080/10543406.2014.920345