BIO Web of Conferences **88**, 00002 (2024) https://doi.org/10.1051/bioconf/20248800002
ICIAS 2023

The Use of R Programming Language for Modelling *In vitro* Rumen Fermentation Gas Production

Hendra Herdian[1*], *Ahmad* Sofyan[1], *Awistaros Angger* Sakti[1], *Muhammad Ainsyar* Harahap[1], *Wulandari*[1], *Slamet* Widodo[1], *Harwi* Kusnadi [1], *Ayu Septi* Anggraeni [2], and *I Nyoman Guna* Darma[3]

[1] Research Center for Animal Husbandry National Research and Innovation Agency (BRIN), Cibinong Indonesia
[2] Research Center for Food Technology and Processing National Research and Innovation Agency (BRIN), Gunungkidul Indonesia
[3] IPB University, Bogor Indonesia

Abstract. The in vitro gas production technique is one of the feed evaluation methods that capture the characteristics of rumen microbial fermentation in response to the gas produced by feed ingredients. The gas production response is related and modelled to a unit of time with several parameters using non-linear regression mathematical equations. To facilitate this task, a script program in R has been developed that can perform the intended calculations. The calculation process uses nonlinear mathematical equations with intercepts. The equation shows that gas production (P) (mL) is influenced by several constants that indicate the share of gas production from the soluble fraction (FA) (mL), gas production from the potentially soluble part (FB) (mL), gas production rate (KC) (mL/h) and incubation time (T) (h). In addition to P and T, the three constants are tried to be derived by the program created. The data set is taken from several in vitro gas production studies that have been conducted previously. The Neway Software program of curve fitting calculations was used to validate the results, as the mean difference was tested using an unpaired T-test. Both programs showed the same calculation results for FA (P=0.9996), FB (P=0.9998), and KC (P=0.9962). The residual standard error (RSE) and R-square (R2) values of both programs also came with the same result they were RSE (P =0.9978), and R2 (P=0.6087) respectively. This shows that the R program script made can carry out the calculation process of making a model of the gas production equation for modeling in vitro gas production from rumen fermentation.

* Corresponding author: hendravit@yahoo.com

© The Authors, published by EDP Sciences. This is an open access article distributed under the terms of the Creative Commons Attribution License 4.0 (https://creativecommons.org/licenses/by/4.0/).

1 Introduction

The feed evaluation technique uses live livestock (in vivo), especially for ruminant livestock, apart from requiring the availability of large numbers of livestock to fulfil statistical sampling, it also involves a long maintenance process. One way to evaluate feed quickly and simply is to carry out in vitro evaluation techniques which involve the use of rumen fluid in the feed material substrate fermentation process [1]. A fairly accurate response was shown by in vitro techniques to predict in vivo results in reducing CH4 emissions due to the addition of tannin [2]. To represent observation data resulting from substrate incubation, a calculation process is needed to fit the response data into a mathematical equation that explains the conditions of the substrate incubation process in more detail and specifically [3]. In practice, the data fitting calculation process requires the help of a computer program to carry out calculations quickly and accurately. For this purpose, a program script has been created in the R programming language which can be replicated to carry out the data fitting calculation process from the substrate fermentation process by rumen fluid in vitro.

2 Materials and Methods

2.1 Model

The nonlinear equation from [4] was used as a model to characterise the gas production from rumen fermentation in vitro. The equation was described below :

$$P = a+b^{(1-e-ct)} \qquad (1)$$

With P = total gas production (mL),
a = gas distribution from soluble fraction (mL),
b = gas production from the potentially soluble part (mL),
c = gas production rate (mL/h)
e = euler's number.

2.2 Data Source and Statistical Method

Data sets were taken with kind permission from [5] and [6] which reported the supplementation of organic minerals, probiotics, inulin, and protected amino acids into ruminal feed (20 observation data) and the use of swimming crab shells in dietary feed (14 observation data) to the response of ruminal gas production in vitro respectively. Both studies used The Neway program [7] to predict the kinetic equation of gases produced from rumen fermentation in vitro. This Neway program is an external add-in program for Microsoft Excel that calculates and displays parameter values from the equation (a, b, c) along with measures of the accuracy of the prediction, i.e. Residual Standard Error (RSE), and Discriminant Coefficient (R2). To compare with the previous analysis results we use R program version 4.2.3. [8] edited on RStudio ver. [9] with minpack.lm ver. 1.2.4. [10] package for statistical calculation, for data manipulation and others used : broom ver. 1.0.5 [11], writexl ver. 1.4.2 [12], dplyr ver. 1.1.3 [13], shiny ver. 1.7.5.1 [14], readr ver. 2.1.4 [15], openxlsx ver. 4.2.5.2 [16] , shinythemes ver. 1.2.0 [17] and ggplot2 ver. 3.4.4 [18]. The mean of the results was analyzed using the T-test Two-Sample Assuming Equal Variances from Microsoft Excel ver. 2010 data analysis. Some of the parts of the R script were produced from the response of personal communication with ChatGPT version GPT-3.5 [19].

The flow of the script for the calculation process is described at Figure 1.

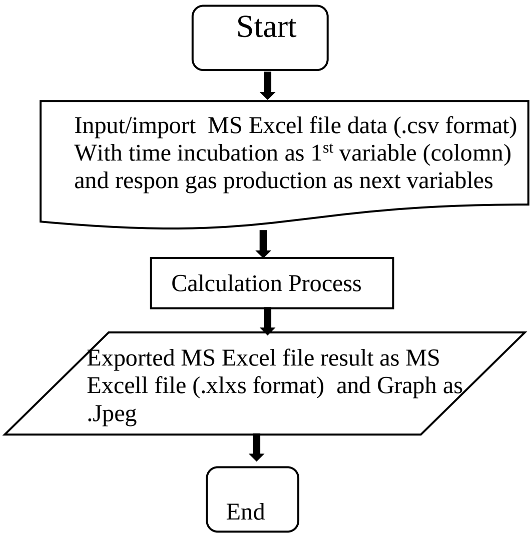


Fig. 1. The flowchart of the calculation

First, we collect the base data from MS Excel in the form of a .csv file (Comma Separated Value). The file itself should be organised in a data set constructed with the first variable/ column representing numerical data indicating the observation time points (hours), which will be used as predictive data and paired with dependent data from the 2nd column onwards, representing gas production response (mL) from each sample. The pairs of data will be used to calculate their regression characteristics using the nlsLM function [10] from the R minpack.lm package based on the nonlinear regression equation proposed [4]. The calculated constant values will be collected as a data frame and stored in a .xlsx file (downloadable). Meanwhile, for the visualisation process, the data pairs will be displayed in the form of a 2D non-linear regression graph, and this graph can be saved in a .jpeg format.

2.3 R Script

The R script to process such calculation was written as follows :

```
#====Start from here===============================================
# To Use it Just copy from Start text to end of the script, then paste it into your R or R studio (recommended) editor prior to run
# The File to import should be in .csv (comma separated value) form with the structure #1st numeric variable allocated for time
# incubation, and #2nd and toward filled as gas production response from samples observation
#---- please install the packages bellow first by remove "#" for no installed packages
#install.packages("minpack.lm")
#install.packages("shiny")
#install.packages("readr")
```

```
#install.packages("openxlsx")
#install.packages("broom")
#install.packages("writexl")
#install.packages("dplyr")
#install.packages("shinythemes")
#install.packages("ggplot2")
library(shiny)
library(readr)
library(openxlsx)
library(minpack.lm)
library(broom)
library(writexl)
library(dplyr)
library(shinythemes)
library(ggplot2)
generate_plot <- function(data, x_title, y_title, plot_title) {
 data1 <- colnames(data)[1]
 data_long <- tidyr::gather(data, key = "Variable", value = "Value", -!!as.name(data1))
 plot <- ggplot(data_long, aes(x = !!as.name(data1), y = Value, color = Variable))+
geom_blank() + geom_line(alpha = 0)+
  labs(
   x = x_title,
   y = y_title,
   title = plot_title
  ) +geom_smooth(method = loess, se = FALSE)+
  theme_minimal()
 return(plot)
}
ui <- fluidPage(
 theme = shinytheme("slate"),
 titlePanel(
  div(
   style = "font-size: 24px;", "In Vitro Gas Production Modelling (Orskov & McDonald,
1979)"
  )
 ),
 sidebarLayout(
  sidebarPanel(
   fileInput("file", "Choose File (.csv)", accept = ".csv"),
   downloadButton("downloadDataXLSX", "Download .xlsx"),
   textInput("plot_title", "Graphic title:", ""),
   textInput("x_title", "X-axis title:", ""),
   textInput("y_title", "Y-ordinate title:", ""),
   downloadButton("download_plot", "Download graphic(.jpeg)"),
   actionButton("reset_button", "Reload"),
   actionButton("quit_button", "Quit")
  ),
  mainPanel(
   h4("Observation Data", style = "text-align: center; color: yellow;"),
   tableOutput("data"),
   h4("Calculation Data", style = "text-align: center; color: orange;"),
```

```
      tableOutput("data_new"),
      plotOutput("plot")
    )
  )
)
server <- function(input, output, session) {
  data_dummy <- reactiveVal(NULL)  # Initialize data_dummy as a reactiveVal
  observeEvent(input$file, {
    req(input$file)
    data <- read.csv(input$file$datapath)
    data_dummy(data)  # Store the data in data_dummy
  })
  observeEvent(input$reset_button, {
    updateTextInput(session, "plot_title", value = "")
    updateTextInput(session, "x_title", value = "")
    updateTextInput(session, "y_title", value = "")
    data_dummy(NULL)
  })
  observeEvent(input$quit_button, {
    # Quit the application
    stopApp()
  })
  output$plot <- renderPlot({
    data <- data_dummy()  # Retrieve data from data_dummy
    if (!is.null(data)) {
      plot <- generate_plot(data, input$x_title, input$y_title, input$plot_title)
      print(plot)
    }
  })
  Data_fitcurve <- reactive({
    req(input$file)
    data <- read.csv(input$file$datapath)
    counter <- 0
    Colom <- ncol(data)
    Coefficient_table_new <- NULL
    Variable_name_WO_time <- NULL
    for (i in 2:Colom) {
      Data <- data.frame(incubate_time = data[, 1], gas = data[, i])
      model <- nlsLM(gas ~ a + b * (1 - exp(-c * incubate_time)), data = Data,
                     start = list(a =  max(Data$gas), b = 0., c = 0.1))
      summary(model)
      Coeficient_equation <- broom::tidy(model)
      Coefficient_table <- data.frame(
        a = Coeficient_equation$estimate[1],
        Std_a = Coeficient_equation$std.error[1],
        Stat_a = Coeficient_equation$statistic[1],
        P_a = Coeficient_equation$p.value[1],
        b = Coeficient_equation$estimate[2],
        Std_b = Coeficient_equation$std.error[2],
        Stat_b = Coeficient_equation$statistic[2],
        P_b = Coeficient_equation$p.value[2],
```

```
        c = Coeficient_equation$estimate[3],
        Std_c = Coeficient_equation$std.error[3],
        Stat_c = Coeficient_equation$statistic[3],
        P_c = Coeficient_equation$p.value[3],
        RSE = sigma(model), # Residual Standard Error
        R2 = 1-(sigma(model)^2/var(Data$gas))
        )
      if (counter == 0) {

Coefficient_table_new <- Coefficient_table
      } else {
        Coefficient_table_new <- rbind(Coefficient_table_new, Coefficient_table)
      }
      counter <- counter + 1
        if (is.null(Variable_name_WO_time)) {
        Variable_name_WO_time <- colnames(data)[-1]
      }
    }
    return(list(Data_fitcurve = data, Coefficient_table_new = Coefficient_table_new,
Variable_name_WO_time = Variable_name_WO_time))
  })
  output$data <- renderTable({
    Data_fitcurve()$Data_fitcurve
  })
  output$data_new <- renderTable({
    data <- cbind(Data_fitcurve()$Variable_name_WO_time,
Data_fitcurve()$Coefficient_table_new)
    colnames(data) <- c("Variable_name",
colnames(Data_fitcurve()$Coefficient_table_new))
    data
  })
  output$download_plot <- downloadHandler(
    filename = function() {
      paste("plot.jpeg", sep = "")
    },
    content = function(file) {
      data <- data_dummy()  # Retrieve data from data_dummy
      if (!is.null(data)) {
        jpeg(file, width = 800, height = 600)
        plot <- generate_plot(data, input$x_title, input$y_title, input$plot_title)
        print(plot)
        dev.off()
      }
    }
  )
  output$downloadDataXLSX <- downloadHandler(
    filename = function() {
      paste("output.xlsx", sep = "")
    },
    content = function(file) {
      if (!is.null(Data_fitcurve()$Coefficient_table_new)) {
```

```
      data <- cbind(Variable_name = Data_fitcurve()$Variable_name_WO_time,
Data_fitcurve()$Coefficient_table_new)
      output_file <- tempfile(fileext = ".xlsx")
      writexl::write_xlsx(data, output_file)
      file.copy(output_file, file)
     }
   }
  )
}
shinyApp(ui, server)
###=========================end of script=========================
```

3 Results and Discussion

This program generates web-based output that displays a data frame from a .csv file as observational data to be calculated and also displays the result data frame (Figure 2.). Visually, it also presents a graph illustrating the relationship between the observation time data and the gas production response from the samples tested (Figure 3.). The calculations data frame and the graph results could be downloaded into the desired location in the local computer directory.

The calculation results show that both software (Neway and R) are able to completely produce the requested non-linear equation calculation based on the constant values that appear (Table 1). The value of the constant "a" shows results that are not significantly different (P = 0.9996) for the two software's. The two softwares Neway and R also produce values that are not different for the constants "b" (P = 0.998) and "c" (P = 0.9962). The Neway program produces information about the effectiveness of predictions from the resulting equation in the form of Residual Standard Error (RSE) values and Coefficient of Determination (R2) values as well as the R program. The results of statistical analysis show that the RSE and R2 values for the two programs are not significantly different (P=0.9978) and (P=0.607) respectively. The R program on the other hand is able to provide information on the statistical values attached to each calculated constant. The statistical values in question are the standard deviation (Std) value and the P value (P), these two values are not displayed in the Neway program. The relationship between the graphical prediction results for all components (a,b,c, RSE, R2) between the Neway calculation results and R shows the same linear pattern (Figure 3 : 8).

The Neway program [7] is a program using a Solver program which is an add-in program to Microsoft Excel. This Add in Solver program is capable of carrying out calculation processes to achieve optimization conditions that produce the maximum or minimum value of the final goal to be achieved [20] so that this program is able to carry out calculations for curve fitting. The Neway program creates a fitting curve with a single calculation using only the [4] equation. On the other hand, the R program is an open source program that has a domain for statistical calculations and graphics [9]. The use of the R program package "minpack.lm" [10] for modelling in vitro gas production is in accordance with the use of the R program package - "drc" which can be used to calculate eight mathematical models for in vitro gas production and feed degradation in sacco [3].

Table 1. The modelling results of the gas production from rumen fermentation in vitro

Application [1)]	a [2)]			b			c			RSE [4)]	R2 [5)]
	Est [3)]	Std. [3)]	P [3)]	Est	Std.	P	Est	Std.	P		
Neway	0.4042	NA	NA	39.0262	NA	NA	0.0461	NA	NA	0.4570	0.9949
R	0.4041	0.2797	0.2456	39.0274	0.8058	0.0001	0.0461	0.0036	0.0020	0.4570	0.9936
P-2t [6)]	0.9996	NA	NA	0.9998	NA	NA	0.9962	NA	NA	0.9978	0.6087

1) Software used for fitting curve i.e. Neway [7] and R Program 4.2.3 [8]
2) Constans equation for gas production from rumen invitro fermentation [4] i.e, “a” : gas production form rapid soluble fraction (mL), “b” : gas portion from potential degraded material (mL), “c” : gas production rate (mL/h)
3) Est : Constants value from calculation, Std = Standard deviation, P = P value
4) RSE: Residual standard error
5) R2 : R-squared value
6) P-2t : two tail P value from T-test determination, NA : unrevealed value

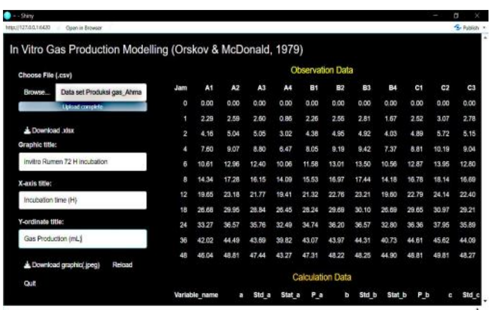


Fig. 2. Observation & calculation data display.

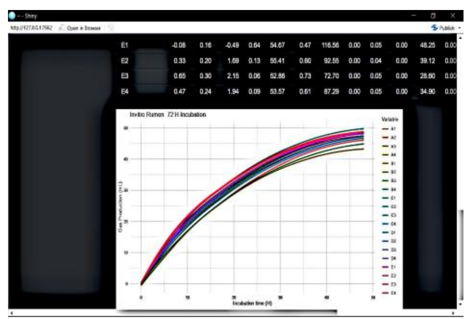
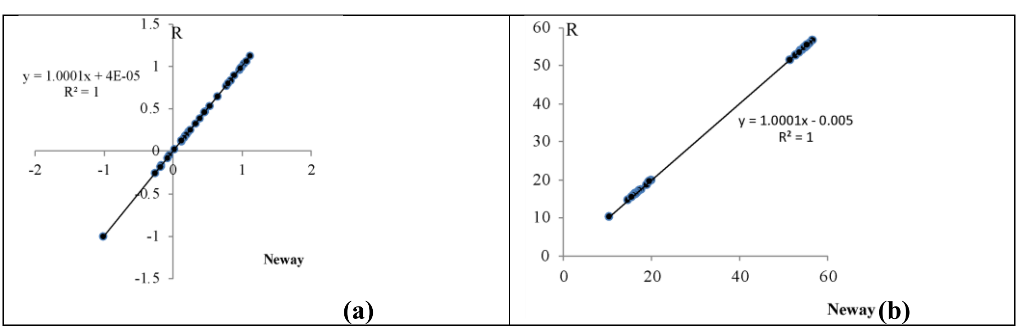


Fig. 3. Graphic visualization.



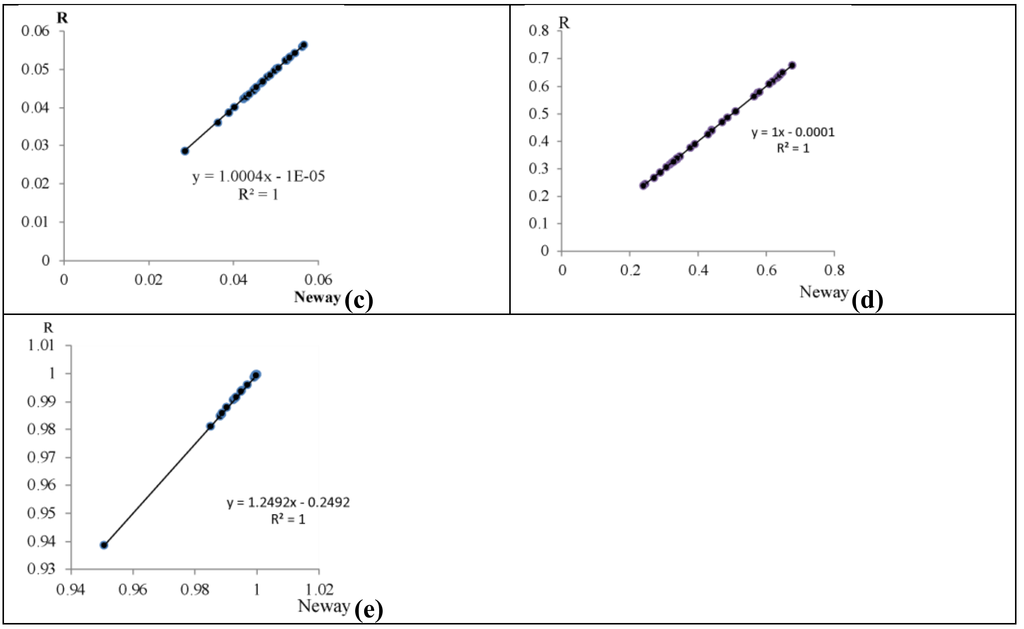


Fig. 4. (a) The "a" from Neway & R calculation, (b) The "b" from Neway & R calculation, (c) The "c" from Neway & R calculation, (d) The "RSE Neway & R calculation, and (e) The "R^2" from Neway & R calculation

4 Conclusion

The use of the R program script that has been made in this paper is able to carry out the modelling process for the process of calculating non-linear mathematical equations for gas production from rumen fluid fermentation in vitro.

Acknowledgement. This paper was funded by the Program House for Quality livestock and Seeds (Crops and Livestock) Agricultural and Food Research Organization (OR PP) National Research and Innovation Agency (BRIN) 2023.

References

1. C. Rymer, J.A. Huntington, B.A. Williams, D.I. Givens, Anim. Feed Sci. Technol. **123–124**, 9–30, (2005).
2. A. Jayanegara, F. Leiber, M. Kreuzer, Journal of Animal Physiology and Animal Nutrition, **96**(3), 365-375 (2011)
3. L.E. Sembach, H.H. Hansen, R. Dhakal, T. Eriksson, N.I. Nielsen, M.O. Nielsen, Estimates for rumen dry matter degradation of concentrates are higher, but not consistently, when evaluated based on in sacco as compared to in vitro methods, Proceedings of the 10th Nordic Feed Science Conference, (2019)
4. Ørskov E R, McDonald I, J. Agri.c Sci. **92** pp 499-3, (1979)
5. A. Sofyan, H. Herdian, A. A. Sakti, A. S. Anggraeni1, A. F. Amri, T. Kurniawan, I. N. G. Darma, A. Febrisiantosa, Evaluation of In Vitro Ruminal Fermentation Kinetics of Diet Mixed with Mineral Organic Additive Using Artificial Neural Network. In proceedings of the 4th International Conference of Animal Science and Technology (ICAST 2021) AIP Conf. Proc. 2628, 030016-1–030016-9; Indonesia, https://doi.org/10.1063/5.0143968, (2023)

6. A. S. Anggraeni, A. Sofyan, A. Jayanegara, E. B. Laconi, N. R. Kumalasari, H. Herdian, I. N. G. Darma, Impact of distinct levels swimming crab shell powder (Portunus pelagicus) supplementation on in vitro rumen fermentation properties processed with phosphoric acid as a chitosan source in proceeding of IOP Conf. Series: Earth and Environmental Science 977, Indonesia, https://iopscience.iop.org/article/10.1088/1755-1315/977/1/012124/meta ,(2022)
7. Chen X.B., Neway-Excel Microsoft Office: A Utility for Processing Data of Feed Degradability and In Vitro Gas Production. England: Rowett Research Institute, (1997)
8. R Core Team, R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL: https://www.R-project.org/, (2023)
9. Posit team, RStudio: Integrated Development Environment for R. Posit Software, PBC, Boston, MA. URL http://www.posit.co/, (2023)
10. Elzhov , Timur V., Katharine M.Mullen, Andrej-Nikolai Spiess, Ben Bolker, R Interface to the Levenberg-Marquardt Non linear Least-Squares Algorithm Found in MINPACK,Plus Support for Bounds, https://cran.r-project.org/package=minpack.lm (2023)
11. Robinson D, Hayes A, Couch S, _broom: Convert Statistical Objects into Tidy Tibbles_. R package version 1.0.5, https://CRAN.R-project.org/package=broom. (2023)
12. Ooms J., _writexl: Export Data Frames to Excel 'xlsx' Format_. R package version 1.4.2, https://CRAN.R-project.org/package=writexl, (2023)
13. Wickham H, François R, Henry L, Müller K,Vaughan D, _dplyr: A Grammar of Data Manipulation_. R package version 1.1.2, https://CRAN.R-project.org/package=dplyr, (2023)
14. Chang W, Cheng J, Allaire J, Sievert C, Schloerke B, Xie Y, Allen J, McPherson J, Dipert A, Borges B., _shiny: Web Application Framework for R_. R package version 1.7.5, https://CRAN.R-project.org/package=shiny. (2023)
15. Wickham H, Hester J, Bryan J, _readr: Read Rectangular Text Data_. R package version 2.1.4, https://CRAN.R-project.org/package=readr, (2023)
16. Schauberger P, Walker A., _openxlsx: Read, Write and Edit xlsx Files_. R package version 4.2.5.2, https://CRAN.R-project.org/package=openxlsx, (2023)
17. Chang W, _shinythemes: Themes for Shiny_. R package version 1.2.0, https://CRAN.R-project.org/package=shinythemes, (2021)
18. Wickham, H., ggplot2: Elegant Graphics for Data Analysis. Springer-Verlag New York. (2016)
19. OpenAI., ChatGPT (August 3 Version) [Large language model], https://chat.openai.com, (2023)
20. Herdian, H, Bulletin Peternakan **31,** 127-135 (2007)