Guide for updating the Crohn’s Disease Treatment Sequence Cost-Effectiveness Model

Version 1.0
January 2025
Dr Matthijs Versteegh & Dr Simone Huygens

Table of Contents

Guide for updating the Crohn’s Disease Treatment Sequence Cost-Effectiveness Model......1

1. Introduction......2
- Purpose of the manual......2
- Overview of existing model structure......2
- Prerequisites and required software/tools......2
- Loading the model into R......2

2. Adding a new treatment to the model......4
2.1 Required Data Elements......4

3. Updating the Network Meta-Analysis......6
3.1 Trial Eligibility Assessment......6
3.2 Data Extraction Protocol......6
3.3 Network Update Process......7
3.4 Output Generation......10

4. Cost-Effectiveness Model Updates......12
4.1 Model Structure Modifications......12
4.2 Parameter Updates......13
4.3 Uncertainty Analysis Updates......17
4.4 Generate deterministic output......17

5. Quality Control Procedures......18
5.1 Technical Validation......18

6. Documentation Requirements......18
6.1 Technical Documentation......18

7. Version Control and Archiving......19
- File naming conventions......19
- Backup procedures......19

1. Introduction

Please read the two scientific manuscripts of the model before starting your update of the model.

Versteegh M, Idema DL, Huygens S, Jenniskens K, Pierik M, Römkens T, van Schaik F, Wahab P, Huis LF, Kusters M, van der Braak K. Comparative efficacy of all available pharmaceutical therapies for moderate to severe Crohn's disease: a systematic review and network meta-analysis. Gastro Hep Advances. 2025 Aug 12:100764.

Versteegh MM, Huygens SA, Bermejo I, Grimm S, Pierik M, Römkens T, van Schaik F, Wahab P. Cost–Utility Analysis of Treatment Sequences for Moderate-to-Severe Crohn's Disease. PharmacoEconomics. 2025 Aug 23:1-3.

- Purpose of the manual

This document serves as support for updates of the Crohn's disease treatment sequence model. The manual contains instructions on updating effectiveness evidence (in the network meta-analysis), updating model code for a new comparator, updating costs and updating effects.

- Collaborators

The Crohn's disease treatment sequence model was developed in 2022-2024 in a collaboration with Zorginstituut Nederland, Cochrane Netherlands (for the systematic literature review), researchers from Maastricht University Medical Center and four appointed gastroenterologists of the 'Nederlandse Vereniging voor Maag, Darm en Leverziekten' (NVMDL). Please find an overview of the collaborators in the table below.

Organisation	Name of involved staff
Zorginstituut Nederland	Matthijs Versteegh (temporary staff) Simone Huygens (temporary staff)
Cochrane Nederland (Julius)	Anneke Damen Kevin Jenniskens Lotty Hooft
NVMDL	Tessa Römkens Fiona van Schaik Marieke Pierik Peter Wahab
MUMC+	Inigo Bermejo Sabine Grimm

- Prerequisites and required software/tools

Updating the model code requires access to and experience with R and R-studio.

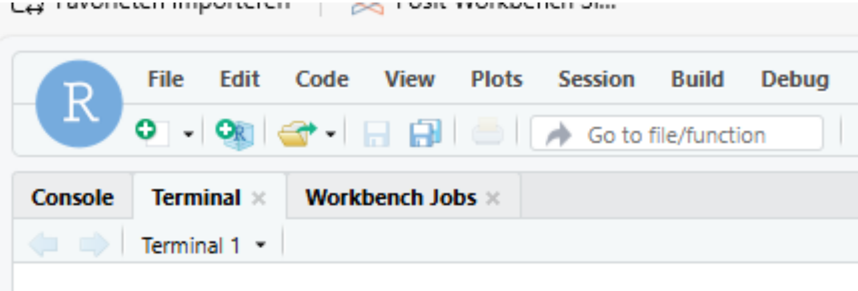
- Loading the model into R

The model is located in the following github repository:
https://github.com/matthijsversteegh/Crohn_public_model.git
If you want to load the model into the posit workbench follow these steps:

1) Open R studio / Open Posit Workbench and Open +New Session in RStudio Pro
2) File -> new project
3) Select ‘version control’
4) Select ‘Git’
5) For repository URL copy the github link above to the repository
6) You can now access the repository. If you want to commit changes to the repository you can use the same username and Personal Access Token as password, which is specific to this repository.

If the above does not work (depends on R version) try the following:

1) Open the terminal



2) Type: git clone https://github.com/matthijsversteegh/Crohn_public_model.git

2. Adding a new treatment to the model

Adding a new treatment to the model consists of the following steps:

1) updating the NMA
2) saving the NMA results for the deterministic and probabilistic analysis
3) updating the cost-effectiveness model code

All steps shall be discussed in this guidance document. If you run into problems please contact dr Matthijs Versteegh (matthijs@huygensandversteegh.com).

2.1 Required Data Elements

- General code structure

The model code is structured as an 'R-project'. The main code for the microsimulation and the NMA are provided in an R-markdown file. Functions are generally saved in a separate R-script and input data in csv files with comma separated values. All files can be accessed by opening the R-project file 'Crohn.Rproj'.

- Clinical efficacy and safety data

Clinical efficacy and safety data in the model is based on a NMA, based on an systematic literature search (SLR) with a search up to October 2023. The NMA is described in [reference]. The NMA code is part of the R-project and can be found in NMA Crohn.Rmd. How to adapt this data is explained in the next chapter of this manual.

- Cost data

The cost data has different sources by type which are described below. Costs are expressed in three monthly cycles in XXX Euros, differentiated between induction cycles and maintenance cycles.

- Drug acquisition costs

Drug acquisition costs are sourced from medicijnkosten.nl for list prices and an analysis of Vektis data for declared prices. They are imported from .csv file in section 3 of the model. To add a new treatment, an additional row needs to be added to this csv file.

Treatment duration and scheduling, and associated 3-monthly costs, are based on 'Farmacotherapeutisch kompas'.

- Health state costs

Health state costs are taken from the COIN study. They do not need to be updated with anything other than the consumer price index when a new treatment is added to the model. The data from the COIN study originates from 2011.[ref] The costs used in our model are based on additional (unpublished) analyses on this dataset. To update the costs to the current year, the equations below should be multiplied with the consumer price index of the current year relative to 2023 or the consumer price index of 2023 relative to 2015 (1.2606) can be replaced with the consumer price index of the current year relative to 2015 (depending on the available consumer price index data).

```
# Correct for consumer price index: 2023 relative to 2015 = 1.2606, 2011 relative to 2015 = 1.0667

c_mrc_2023_REM                      <- c_mrc_2011_REM*1.0667*1.2606
c_mrc_2023_ACT                      <- c_mrc_2011_ACT*1.0667*1.2606
```

- Quality of life data

Quality of life data are sourced from the IB-Dream study. They do not need to be updated when a new treatment is added to the model.

3. Updating the Network Meta-Analysis

The NMA r-code is available at request please reach out to dr. Matthijs Versteegh

The network meta-analysis produces relative risks for the following subgroups based on induction or maintenance treatment and whether patients already had a biological (exposed) or not (naïve):

- induction remission naive (chunk 1.3.1)
- induction remission exposed (chunk 1.3.2)
- maintenance remission naive (chunk 1.4.1)
- maintenance remission exposed (chunk 1.4.2)

To run the NMA, first the set-up chunks need to be run (0.1, 1.1). Subsequently, the subgroup chunks can be run line by line.

3.1 Trial Eligibility Assessment

- Inclusion/exclusion criteria review

Please refer to the original scientific manuscript.[ref]The main criteria are:

- Moderate/severe Crohn (CDAI>220)
- Defined subgroups (biologic naive or exposed, when no subgroups are available use 60% cut-off see manuscript)

- Outcome measure alignment

The model uses the Crohn Disease Activity Index (CDAI) to define health states remission and active disease. Sometimes studies report disease activity with the Harvey-Bradshaw Indicator (HBI). Then this can be mapped onto CDAI.
Remission < CDAI 150
If outcome is measured with HBI, translate HBI to CDAI, use following transformations:
# CDAI = 60 + HBI*22.5
# HBI = CDAI/22.5 - 60/22.5

- Study quality assessment

Use GRADE & Risk of bias as described in the study manuscript.[ref]

3.2 Data Extraction Protocol

- Required data points

Data should be extracted following the file that is used to read in the trial data in the NMA Crohn.Rmd script:

```
NMA_data_crohn <- read.csv(here::here("Input",
"NMA_input_file_update14022024.csv"), sep = ";" , header = T) #update with
subgroups for Loftus2023 (upadacitinib subgroups)
```

To include a new trial, add a line for each treatment in the study in the last row of the .csv file. Comparators from the same study receive the same study.id. The procedure is as following:

1) Run section 1.1 up to (and including) the read.csv section above.
2) Add new rows in the `NMA_data_crohn` object, up to columns 'event rate' (make sure the last columns for which you do not have data are provided with an 'NA' value).
    a. `NMA_data_crohn[nrow(NMA_data_crohn)+1,] <- c(#write all column values here)`
    b. Store the result in a new .csv using the following code
    ```
    write.csv(NMA_data_crohn, file = here::here("Input",
    "NMA_input_file_update[insertdate].csv"))
    ```
    c. Replace the old input file with the new input file and rerun section 1.1. Make sure to commit and push changes to the repository

3.3 Network Update Process

- Network geometry update

Check if the network is still connected for each of the subgroups. This requires running the following sections (first run preparation sections 0.1 and 1.1):

Chunk section 1.3.1 NMA code for induction remission naïve (ind_REM_nai).

```
#Exclude studies not connected from the network for High Risk of Bias
subanalysis
if(RoB == T){
df_ind_REM_nai <- df_ind_REM_nai[!(df_ind_REM_nai$study.ID == 112), ]
#Martin 1990
df_ind_REM_nai <- df_ind_REM_nai[!(df_ind_REM_nai$study.ID == 127), ]
#Sandborn 1999
df_ind_REM_nai <- df_ind_REM_nai[!(df_ind_REM_nai$study.ID == 68), ]
#Ardizzone 2003

df_ind_REM_nai <- df_ind_REM_nai[!(df_ind_REM_nai$study.ID == 150), ]
#Inconsistent but not RoB
  }

#Total number of studies included for induction response, count
length(unique(df_ind_REM_nai$study.ID))

id    <- df_ind_REM_nai$study.ID
r     <- df_ind_REM_nai$Nevent
n     <- df_ind_REM_nai$N
tr    <- df_ind_REM_nai$int.name

# Transform arm to contrast level data
pair      <- pairwise(treat = tr, event = r, n, studlab = id, sm="RR")
```

If the 'pairwise' function does not work because the network is not connected, an error will occur. You can manually check which studies are not connected by running the code below (after removing '#'):

```
#Check network connection
#nc <- netconnection(pair)
#test_netw <- data.frame(studlab = nc$studlab, subnet = nc$subnet)
#test_netw
```

Results can be generated by running the following commands:

```
# NMA model (check with I2 and Q statistic if fixed-effects or random
effects model is needed)
NMA_ind_REM_nai <- netmeta(pair, sm="RR", ref="Placebo", warn = TRUE,
details.chkmultiarm = TRUE)

forest(NMA_ind_REM_nai, ref = Reference, sortvar = "SUCRA",
drop.reference.group = TRUE, label.left = "Favours comparator",
label.right = "Favours intervention")
```

- Saving results

The updated NMA results need to be saved so they can serve as input for the cost-effectiveness model, for both the base case and the PSA. However, for use in the cost-effectiveness model, the comparator needs to match the comparator used for background transitions probabilities. The table below shows the appriopriate comparator:

Treatment and subgroup	Comparator
Induction naive line 1	Placebo
Induction naive line 2	Placebo
Induction exposed	Placebo
Maintenance naive line 1	Azathioprine 2.5 mg/kg
Maintenance naive line 2	Infliximab 5mg/kg
Maintenance exposed	Ustekinumab 6mg/kg

To do so, run the appropriate comparators in the following section of the code in the maintenance chucks (example below for naive only, but please also run for exposed). Note: *THIS OVERWRITES PREVIOUS NMA OUTPUT. If you want to test it first, save it under a different name.*

```
# Store data with AZA as comparator
v_TE_store <- forest(NMA_mai_REMRES_nai, ref = "Azathioprine 2-2.5 mg/kg")

df_TE_mai_REMRES_nai           <- data.frame(v_TE_store$treat,
exp(v_TE_store$TE), v_TE_store$seTE)
colnames(df_TE_mai_REMRES_nai) <- (c("Treatment", "RR", "seRR"))

write.csv(df_TE_mai_REMRES_nai, here::here("Output",
"df_TE_mai_REMRES_nai_AZA.csv"))

# Store data with IFX as comparator (for line 2 biologic naive patients)
v_TE_store <- forest(NMA_mai_REMRES_nai, ref = "Infliximab 5 mg/kg")
df_TE_mai_REMRES_nai           <- data.frame(v_TE_store$treat,
exp(v_TE_store$TE), v_TE_store$seTE)
```

```
colnames(df_TE_mai_REMRES_nai) <- (c("Treatment", "RR", "seRR"))

write.csv(df_TE_mai_REMRES_nai, here::here("Output",
"df_TE_mai_REMRES_nai_IFX.csv"))
```

The same applies to making new objects for the PSA (example below for naive only, but please also run for exposed). Note: *THIS OVERWRITES PREVIOUS NMA OUTPUT. If you want to test it first, save it under a different name.*

```
#Prepare object for PSA for line 1
v_NMA_mai_REMRES_nai <- forest(NMA_mai_REMRES_nai, ref = "Azathioprine 2-
2.5 mg/kg")
v_NMA_mai_REMRES_nai <- v_NMA_mai_REMRES_nai[v_NMA_mai_REMRES_nai$treat !=
"Azathioprine 2-2.5 mg/kg",]
v_NMA_mai_REMRES_nai <- v_NMA_mai_REMRES_nai$TE

m_mai_REMRES_nai      <- NMA_mai_REMRES_nai$Cov.random
ref_mai_REMRES_nai    <- NMA_mai_REMRES_nai$reference.group
m_mai_REMRES_nai      <-
m_mai_REMRES_nai[grep(ref_mai_REMRES_nai,colnames(m_mai_REMRES_nai)),grep(
ref_mai_REMRES_nai,colnames(m_mai_REMRES_nai))]
save(m_mai_REMRES_nai,v_NMA_mai_REMRES_nai, file =  here::here("Output",
"NMA_PSA_mai_REMRES_nai.RData"))

#Prepare object for PSA for line 2
v_NMA_mai_REMRES_nai <- forest(NMA_mai_REMRES_nai, ref = "Infliximab 5
mg/kg")
v_NMA_mai_REMRES_nai <- v_NMA_mai_REMRES_nai[v_NMA_mai_REMRES_nai$treat !=
"Infliximab 5 mg/kg",]
v_NMA_mai_REMRES_nai_L2 <- v_NMA_mai_REMRES_nai$TE

m_mai_REMRES_nai      <- NMA_mai_REMRES_nai$Cov.random
ref_mai_REMRES_nai    <- NMA_mai_REMRES_nai$reference.group
m_mai_REMRES_nai_L2      <-
m_mai_REMRES_nai[grep(ref_mai_REMRES_nai,colnames(m_mai_REMRES_nai)),grep(
ref_mai_REMRES_nai,colnames(m_mai_REMRES_nai))]
save(m_mai_REMRES_nai_L2,v_NMA_mai_REMRES_nai_L2, file =
here::here("Output", "NMA_PSA_mai_REMRES_nai_L2.RData"))
```

- Adverse events

The NMA for adverse events also requires an update of the .csv file (chunk 1.1.1.):

```
NMA_AE_data_crohn <- read.csv(here::here("Input",
"NMA_AE_input_file.csv"), sep = ",")
```

After its update, chunks 1.5.1 and 1.5.2 can be run.

Do not forget to store the data with the code below for both the deterministic and probabilistic setting in those sections: Note: *THIS OVERWRITES PREVIOUS NMA OUTPUT. If you want to test it first, save it under a different name.*

```
# Store data
v_TE_store              <- forest(NMA_AE_nai, ref = "Placebo")
df_NMA_AE_nai           <- data.frame(v_TE_store$treat,
exp(v_TE_store$TE), v_TE_store$seTE)
colnames(df_NMA_AE_nai) <- (c("Treatment", "RR", "seRR"))
write.csv(df_NMA_AE_nai, here::here("Output", "df_NMA_AE_nai.csv"))

#Prepare object for PSA
v_NMA_AE_nai            <- forest(NMA_AE_nai, ref = "Placebo")
v_NMA_AE_nai            <- v_NMA_AE_nai[v_NMA_AE_nai$treat != "Placebo",]
v_NMA_AE_nai            <- v_NMA_AE_nai$TE

m_NMA_AE_nai            <- NMA_AE_nai$Cov.random
ref_NMA_AE_nai          <- NMA_AE_nai$reference.group
m_NMA_AE_nai            <-
m_NMA_AE_nai[grep(ref_NMA_AE_nai,colnames(m_NMA_AE_nai)),grep(ref_NMA_AE_n
ai,colnames(m_NMA_AE_nai))]
save(m_NMA_AE_nai,v_NMA_AE_nai, file =  here::here("Output",
"NMA_PSA_AE_nai.RData"))
```

3.4 Output Generation

Run the forest plot and league tables with the comparator of interest with the following code (example for induction of remission in exposed patients, but similar for other subgroups):

- Forest plots

```
# Forest plot sorted on P score (descending order)
forest(NMA_ind_REM_exp, ref = Reference, sortvar = "SUCRA",
drop.reference.group = TRUE, label.left = "Favours comparator",
label.right = "Favours intervention")
```

- League tables

```
league_ind_REM_exp <- netleague(NMA_ind_REM_exp, digits = 2, common = F,
seq = netrank(NMA_ind_REM_exp, method = "SUCRA"))
```

- Inconsistency plots

```
plot(netsplit(NMA_ind_REM_exp))
```

- Trial based background probabilities: meta-analysis update

The background probability of induction and adverse events are based on a meta-analysis of the trials included in the NMA. Now that the trials in the NMA have changed, the meta-analysis has to be reconducted. These can be found in section 1.5.2 of the NMA code. The following code provides the relevant output after section 1.1 has been updated:

```
meta_REM_nai <- metaprop(df_ind_REM_nai$Nevent[df_ind_REM_nai$int.name ==
"Placebo"],
                         df_ind_REM_nai$N[df_ind_REM_nai$int.name ==
"Placebo"],
                         df_ind_REM_nai$studyname[df_ind_REM_nai$int.name
== "Placebo"],
                         method = "Inverse")

meta_REM_exp <- metaprop(df_ind_REM_exp$Nevent[df_ind_REM_exp$int.name ==
"Placebo"],
                         df_ind_REM_exp$N[df_ind_REM_exp$int.name ==
"Placebo"],
                         df_ind_REM_exp$studyname[df_ind_REM_exp$int.name
== "Placebo"],
                         method = "Inverse")

meta_disc_nai <-
metaprop(df_AE_data_crohn_nai$Nevent[df_AE_data_crohn_nai$int.desc ==
"Placebo"],

df_AE_data_crohn_nai$N[df_AE_data_crohn_nai$int.desc == "Placebo"],

df_AE_data_crohn_nai$studyname[df_AE_data_crohn_nai$int.desc ==
"Placebo"],
                         method = "Inverse")

meta_disc_exp <-
metaprop(df_AE_data_crohn_exp$Nevent[df_AE_data_crohn_exp$int.desc ==
"Placebo"],

df_AE_data_crohn_exp$N[df_AE_data_crohn_exp$int.desc == "Placebo"],

df_AE_data_crohn_exp$studyname[df_AE_data_crohn_exp$int.desc ==
"Placebo"],
                         method = "Inverse")
```

4. Cost-Effectiveness Model Updates

4.1 Model Structure Modifications

- New treatment arm addition

The following sections need to be updated to allow a new treatment to be included in the model:

1) Add a line with induction duration (cl_wk by default) in section 0.3

```
# Induction periods in weeks (time when you would start maintenance or new
treatment if no response, not time of evaluation of response/remission in
trial)
t_ind_IFX5                      <- cl_wk
t_ind_IFX10                     <- cl_wk
t_ind_IFX_AZA                   <- cl_wk
t_ind_ADA40                     <- cl_wk
t_ind_ADA80                     <- cl_wk
t_ind_ADA_AZA                   <- cl_wk
t_ind_VED                       <- cl_wk
t_ind_UST                       <- cl_wk
t_ind_UPA                       <- cl_wk
t_ind_AZA                       <- cl_wk
t_ind_RIS                       <- cl_wk
t_ind_MTX                       <- cl_wk
t_ind_[newtreatment]            <- cl_wk
```

2) Add the new object by adding t_ind_[treatment abbreviation] after t_ind_MTX in section 0.7.2.

```
clusterExport(cl, c('MicroSim', 'f_create_df_X', 'f_v_M_init', 'n_i',
'rtruncnorm', 'v_n', 'n_t',  'Costs', 'Effs', 'Probs',
                    'cl_wk', 't_ind_IFX5','t_ind_IFX10', 't_ind_IFX_AZA',
't_ind_ADA40','t_ind_ADA80','t_ind_ADA_AZA','t_ind_VED','t_ind_UST','t_ind
_UPA',
                    't_ind_AZA','t_ind_RIS','t_ind_MTX', 'f_extract_prob',
'df_tp_FL1_AZA', 'df_tp_FL2_IFX','df_tp_FL2_UST','df_tp_sur_other',
                    'df_tp_sur_active', 'n_states', 'df_mort',
"p_disc_placebo_nai", "p_disc_placebo_exp", 'v_dwc', 'v_dwe'))
```

3) And in section 0.8.2 (OWSA)

```
clusterExport(cl, c('MicroSim', 'f_create_df_X', 'f_v_M_init', 'n_i',
'rtruncnorm', 'v_n', 'n_t', 'age', 'sd_age',
'TRT1a','TRT2a','TRT3a','TRT4a', 'TRT5a', 'Costs', 'Effs', 'Probs',
                    'cl_wk', 't_ind_IFX5','t_ind_IFX10', 't_ind_IFX_AZA',
't_ind_ADA40','t_ind_ADA80','t_ind_ADA_AZA','t_ind_VED','t_ind_UST','t_ind
_UPA',
                    't_ind_AZA','t_ind_RIS','t_ind_MTX', 'f_extract_prob',
'df_tp_FL1_AZA', 'df_tp_FL2_nai','df_tp_FL2_exp','df_tp_FL2_UST',
                    'df_tp_FL2_IFX','df_tp_OS', 'df_tp_sur_other',
'df_tp_sur_active', 'n_states', 'df_mort', "p_disc_placebo_nai",
"p_disc_placebo_exp", 'v_dwc', 'v_dwe', 'baseline', 'PSA'))
```

4) And in section 0.9.3 (PSA)

```
clusterExport(cl, c('MicroSim', 'f_create_df_X', 'f_v_M_init', 'n_i',
'rtruncnorm', 'v_n', 'n_t',  'Costs', 'Effs', 'Probs',
                    'cl_wk', 't_ind_IFX5','t_ind_IFX10', 't_ind_IFX_AZA',
't_ind_ADA40','t_ind_ADA80','t_ind_ADA_AZA','t_ind_VED','t_ind_UST','t_ind
_UPA',
                    't_ind_AZA','t_ind_RIS','t_ind_MTX', 'f_extract_prob',
'n_states', 'df_mort', "p_disc_placebo_nai", "p_disc_placebo_exp",
'v_dwc', 'v_dwe', 'PSA' ))
```

5) Update the dataframe with treatment sequences by adding the new treatment in the appropriate treatment line in section 03.2.

```
# Define all possible combinations and exclude the sequences that are not
clinically plausible
df_trtseq <- expand.grid(line1 = c("AZA", "MTX", "IFX5", "ADA40",
"IFX+AZA", "UST"),
                         line2 = c("IFX5", "ADA40", "IFX+AZA", "VED",
"UST", "RIS", "UPA"),
                         line3 = c("IFX5", "ADA40",  "IFX+AZA", "VED",
"UST", "RIS", "UPA"),
                         line4 = c("VED", "UST", "RIS", "UPA"),
                         line5 = c("VED", "UST", "RIS", "UPA"),
stringsAsFactors = FALSE)
```

Note that not all sequences are clinically possible. The code in chunck 0.3.2 allows options to exclude sequences. For example, if a treatment can only be prescribed for biological exposed patients, sequences with the new drug in line 2 after an immunomodulator should be excluded.

```
df_trtseq <- df_trtseq[!(df_trtseq$line1 == "AZA" & df_trtseq$line2 ==
"MIR"),]
df_trtseq <- df_trtseq[!(df_trtseq$line1 == "MTX" & df_trtseq$line2 ==
"MIR"),]
```

- Cycle length considerations

The model runs with a 3 monthly cycle length. It is not recommended to change the cycle length.

4.2 Parameter Updates

- Efficacy parameters from NMA

This section assumes that the R-objects that are imported in section 0.2.1. have been updated and saved with the updated NMA.

The NMA data is read in section ‘0.2.1 Load NMA data’. This section requires one update after the NMA has been updated, which is to add the abbreviated name that will be used throughout the model. For example, if "Mirikizumab" is added to the NMA, it will be abbreviated to “MIR”. Do so in the following code section by adding an `ifelse` statement that matches the exact name used in the NMA and add an additional bracket) at the end:

```
# Add abbreviated names to the combined data frame to avoid repetition
all_NMA_data$curTrt <- ifelse(all_NMA_data$Treatment == "Azathioprine 2.5
mg/kg",  "AZA",
                       ifelse(all_NMA_data$Treatment == "Azathioprine 2
mg/kg",    "AZA",
                       ifelse(all_NMA_data$Treatment == "Azathioprine 2-
2.5 mg/kg", "AZA",
                       ifelse(all_NMA_data$Treatment == "Adalimumab 40 mg"
& all_NMA_data$type == "df_NMA_ind_REM_nai" ,   "ADA20", #maintenance
always 40mg
                       ifelse(all_NMA_data$Treatment == "Adalimumab 40 mg"
& all_NMA_data$type != "df_NMA_ind_REM_nai" ,   "ADA40", #maintenance
always 40mg
                       ifelse(all_NMA_data$Treatment == "Adalimumab 160
mg",         "ADA40", #because starts with 160 mg to 80 mg ends with 40 mg
(default dosing)
                       ifelse(all_NMA_data$Treatment == "Adalimumab +
Azathioprine 160 mg + 25-50 mg",                           "ADA+AZA",
                       ifelse(all_NMA_data$Treatment == "Adalimumab +
Azathioprine / 6-mercaptopurine 40 mg + 25-50 mg / 30 mg", "ADA+AZA",
                       ifelse(all_NMA_data$Treatment == "Etrolizumab 105
mg",       "ETR105",
                       ifelse(all_NMA_data$Treatment == "Etrolizumab 210
mg",       "ETR210",
                       ifelse(all_NMA_data$Treatment == "Infliximab 5
mg/kg",      "IFX5",
                       ifelse(all_NMA_data$Treatment == "Infliximab 10
mg/kg",     "IFX10",
                       ifelse(all_NMA_data$Treatment == "Infliximab +
Azathioprine 5 mg/kg + 2-2.5 mg/kg","IFX+AZA",
                       ifelse(all_NMA_data$Treatment == "Methotrexate 25
mg",       "MTX",
                       ifelse(all_NMA_data$Treatment == "Methotrexate 15
mg",       "MTX",
                       ifelse(all_NMA_data$Treatment == "Risankizumab 600
mg",      "RIS",
                       ifelse(all_NMA_data$Treatment == "Risankizumab 360
mg",      "RIS",
                       ifelse(all_NMA_data$Treatment == "Upadacitinib 45
mg",       "UPA",
                       ifelse(all_NMA_data$Treatment == "Upadacitinib 30
mg",       "UPA", #change 30 to 15 for other dosage
                       ifelse(all_NMA_data$Treatment == "Ustekinumab 6
mg/kg",     "UST",
                       ifelse(all_NMA_data$Treatment == "Ustekinumab 90
mg",        "UST",
                       ifelse(all_NMA_data$Treatment == "Vedolizumab 300
mg",       "VED",
                       ifelse(all_NMA_data$Treatment == "Vedolizumab 108
mg",       "VED", "noabbr")))))))))))))))))))))))
```

- Cost parameters

The following .csv's need to be updated with updated list prices or prices that include discounts taken from Vektis data.

```
# Costs per 12-weekly period using list prices from 2023
if(prices == "list"){
  df_c_trt                    <- read.csv(here::here("input",
"list_prices.csv" ), sep = ",", header = T)
  colnames(df_c_trt)          <- c("curTrt", "Induction", "Maintenance")
  }

# Costs per 12-weekly period using Vektis declaration data from 2023 year
if(prices == "discounted"){
  df_c_trt                    <- read.csv(here::here("input",
"declaration_prices.csv"), sep = ",", header = T)
  colnames(df_c_trt)          <- c("curTrt", "Induction", "Maintenance")
  }
```

To add cost for a new treatment:

```
df_c_trt[nrow(df_c_trt)+1, ]  <- c("MIR", 16691.13, 5563.71)
df_c_trt[, 2]  <- as.numeric(df_c_trt[, 2])
df_c_trt[, 3]  <- as.numeric(df_c_trt[, 3])
```

Alternatively you can export the .csv file (More > Export) and add the new treatment costs in Excel and then save the .csv and upload it under the same name. NB. It is possible that your computer system has a different default separator for .csv files (e.g. ; instead of ,). If this is true, this can be changed in sep= in the read.csv code.

- Utility values

No updates required.

- Background probabilities: meta analysis

The updated objects from the updated meta-analysis are:

```
meta_REM_nai
meta_REM_exp
meta_disc_nai
meta_disc_exp
```

These objects hold the background placebo probability and 95% confidence intervals of induction (naive and exposed) and discontinuation (naive and exposed). Use these to update the model input parameters (section 3, subsection 'Event probabilities', specifically the following code:

```
# Placebo probabilities for remission (p_ACTREM) based on meta-analysis
p_ACTREM_nai                    <- 0.2158 # 95% CI [0.1691; 0.2712]
p_ACTREM_exp                    <- 0.1525 # 95% CI [0.1102; 0.2073]

# Calculate beta parameters for use in PSA
beta_params_p_ACTREM_nai        <- beta_params(p_ACTREM_nai, ((0.2712 -
0.1691) / (2*1.96)))
beta_params_p_ACTREM_exp        <- beta_params(p_ACTREM_exp, ((0.2073 -
0.1102) / (2*1.96)))

# Placebo probabilities for all cause discontinuation based on meta-
analysis
p_disc_placebo_nai              <- 0.0769 # 95% CI [0.0516; 0.1130]
p_disc_placebo_exp              <- 0.0730 # 95% CI [0.0490; 0.1073]
```

Also update the PSA input in section 9.3, specifically this code section:

```
# Background probabilities from meta-analysis to achieve remission /
response
    p_ACTREM_nai = rbeta(n_sim, beta_params_p_ACTREM_nai$alpha,
beta_params_p_ACTREM_nai$beta), #sampled from meta-analysis with 16
studies
    p_ACTREM_exp = rbeta(n_sim, beta_params_p_ACTREM_exp$alpha,
beta_params_p_ACTREM_exp$beta), #sampled from meta-analysis with 14
studies

    p_disc_placebo_nai =  rtruncnorm(n_sim, a = 0, b = 1, mean =
0.01987418, sd = ((0.02963436-0.01320308)/1.96)*sqrt(21)), #sampled from
meta-analysis with 21 studies
    p_disc_placebo_exp = rtruncnorm(n_sim, a = 0, b = 1, mean =
0.01885574, sd = ((0.02810157-0.01253761)/1.96)*sqrt(12)), #sampled from
meta-analysis with 12 studies
```

- Background probabilities: survival functions

Updating the background probabilities is fairly simple from a code perspective, but conceptually possibly a major model code overhaul with many downstream consequences. Please contact the model authors when undertaking this exercise. Simple updates (i.e. the same model fit to the same patient population but with more patients) can simply be imported in section 0.2 by replacing the best_model$XXX in the code below. Note that updating the survival functions also requires updating the transition probability dataframe which is made in the (inactivated) part of the code 0.5.01.

```
# Survival objects for time to active disease based on the IBD-ZL or ICC
cohort
load(here::here("Input", "FirstLine_TTDA.RData"))                      #
Line 1, thiopurine, IBDZL, n = 184, Gamma
s_df_tp_FL1     <- best_model$Gamma
load(here::here("Input", "SecondLine_TTAD_antiTNF_naive_IFX.RData"))    #
Line 2+, IFX, biologic naive, IBDZL, n = 104, Lognormal
s_df_tp_FL2_nai <- best_model$Lognormal
load(here::here("Input", "SecondLine_TTAD_antiTNF_exp_ICC_uste.RData")) #
Line 2+ ustekinumab, biologic exposed, ICC, n = 139, Lognormal
s_df_tp_FL2_exp <- best_model$Lognormal
load(here::here("Input", "FirstLine_TTAS.RData"))                      #
active surgery, IBDZL, n = 198, Exponential, no covariates
s_df_tp_sur_AD <- best_model$Exponential
load(here::here("Input", "FirstLine_TTOS.RData"))                      #
other surgery, IBDZL, n = 178, Lognormal
s_df_tp_sur_OS <- best_model$Lognormal
```

4.3 Uncertainty Analysis Updates

- Probabilistic sensitivity analysis

If all steps above have been undertaken correctly, the PSA input is generated automatically, including the correct updated multivariate normal draws from the correlated NMA output.

4.4 Generate deterministic output

Output is created separately for step-up sequences and top-down sequences. Specific sequences can be compared (section 0.7.1 single sequence) or multiple sequences (section 0.7.2 multiple sequences).

Step up sequences

Step up sequences are sequences in which the first treatment is an ‘older’ non-biologic treatment (MTX or AZA). The procedure to obtain results is to run chunck 0.7.2 entirely for all sequences. Then, the top-down sequences are obtained from the full set of results in this code section:

```
results_step_up <- results_ce[results_ce$L1 == "AZA" | results_ce$L1 ==
"MTX",]
```

Top down sequences

Top down sequences start with a biologic therapy. Similar to above, the full set is run and the top down sequences are extracted afterwards with this code section:

```
results_top_down <- results_ce[results_ce$L1 != "AZA" & results_ce$L1 !=
"MTX",]
```

Table and plot treatment ranking

Once the deterministic output has been generated, all output can be derived. A dataframe with net health benefit is stored in `results_top_down` and `results_step_up.`

To generate a the ranking plot, an update to the function of `make_rank_plot` is required. In the folder FUNCTIONS find the script ‘functions.R’. There, find `make_rank_plot.` Update the function objects `rank_sorted` and `scale_fill_manual` by adding the new treatment name.

5. Quality Control Procedures

5.1 Technical Validation

- Code review process

It is recommended to update the code with a ‘4-eye’ principle. One person updates the code and another person checks the code update.

- Internal validation requirements

The addition of a new treatment to the NMA likely alters the relative risk estimates of the other treatments. It is therefore not possible to reproduce the previous model results when using the updated NMA results.

6. Documentation Requirements

6.1 Technical Documentation

- Code changes

Please make a log of the code changes that have been made. Ideally, start a new repository on Github with the old code as source, and the updates as new commits so that version control is possible all the way back to the original model.

- Parameter sources

Parameter sources are added to the core code with a ‘#’ behind the object. i.e.

```
age                             <- 39
# start age: mean age at diagnosis Jeuring et al. (2017) Era 2006-2011
```

7. Version Control and Archiving

- File naming conventions

Type	Starts with
Data frame	df_
Matrix	m_
Costs object	c_
Utility object	u_
Probability	p_
Rate	r_

Treatment Name Standardization

Add a treatment (example: Mirikizumab) using the same format as existing treatments
Follow the established naming convention: "[Drug Name] [Dosage] [Unit]"
Example: "Mirikizumab 300 mg"
Use abbreviated format "MIR" for consistency with other treatments

- Backup procedures

Create update log through github repository.