User Manual: Platform for Network Meta-Analysis based on Survival data

August 23, 2024

Title Platform for **N**etwork **M**eta-**A**nalysis based on **S**urvival data (NMAs)

Date 2024-8-23

Author Taihang Shao; Mingye Zhao; Fenghao Shi; Mingjun Rui and Wenxi Tang.

Bugs Report and Consultation travis_shao@outlook.com

Description This user-friendly application can be used to conduct network meta-analysis based on reconstructed survival data.

Constructed Based on R (4.3.1)

R Packages Needed shinydashboard, magrittr, shiny, writexl, readxl, DT, heemod, flexsurv, shinycssloaders, dplyr, ggplot2, shinydashboard, shinydashboardPlus, slickR, rhandsontable, shinyWidgets, shinyhelper, shinytitle, clipr, rmarkdown, shinyjs, shinyalert, markdown, markdownInput, shinymanager, rclipboard, tidyverse, gridExtra, survHE, discSurv, survminer, doBy, netmeta, survival, R2jags, ggmcmc, lme4, broom, metafor, R2WinBUGS, MatrixModels, grid

Content

1、Introduction

1.1 Opening this APP

This Shiny APP (PSA) is constructed based on R (4.3.1) and have been deployed on the **shinyapps.io**. The URL of this shiny app is: https://psurvivala.shinyapps.io/NMAs/. Please note that 900 seconds of inactivity before a browser connection to a worker process is idle and the APP will be closed.

1.2 Usage and potential users of this APP

Nowadays, decision making for anti-cancer drugs usually requires a lifetime projection including survival benefit and cost. Meanwhile, healthcare decision-making also requires comparisons of all relevant competing interventions. The individual patient data (IPD) must be available from a clinical study. However, mature IPD are quite often unobtainable. In addition, the realization of some flexible methods needs users to have a certain level of programming skills. This APP can be used to conduct network meta-analysis (NMA) based on survival data. In this APP, we provided several methods including Aggregated data (AD)-based NMA and IPD-based NMA. If users have survival data with covariates, this APP was not recommended, but methods used in this APP can still be applied. Many other methods can be considered including meta-regression and treatment-by-covariates interactions.

Potential users for this APP including investigators who have a good grasp of the principles of survival analysis and NMA, and understand the strengths and weaknesses of these methods, but do not have the time, statistical background or programming expertise to conduct these methods themselves. Potential users can also be researchers who are majored in health economics but do not have a deep understanding of evidence-based medicine or clinicians who do not have a background in programming.

1.3 How to use this APP

This User Manual will introduce the properties of this APP and discusses the situations where it may or may not be suitable. Figures will be provided to help users to learn how to use this APP quickly. Besides, description of the methods used as well as their basic codes and functions will be provided in summary here. If users are not satisfied with the results, they may use provided codes and description to generate results in R themselves. Please note that all data of examples used in this APP can be obtained from published articles. This APP do not make any statement about the appropriateness of particular therapies which are used in examples.

2、Example Data Used in This APP

2.1 Data used in the Data transform and IPD-based NMA

Data used in the Data transform and IPD-based NMA are the same. This data is the reconstructed IPD obtained from Freeman S.C. et.al.'s research[1]. And the original source is a NMA of therapies for previously untreated advanced BRAF-mutated melanoma[2]. The NMA identified 23 eligible articles reporting on thirteen phase II and phase III randomized controlled trials. The melanoma network consists of 3913 overall survival events from 6378 patients. In addition, the network includes 13 trials and 13 treatments: dacarbazine, tremelimumab, ipilimumab, dabrafenib, vemurafenib, nivolumab, pembrolizumab, ipilimumab + dacarbazine, dabrafenib + trametinib, vemurafenib + cobimetinib, nivolumab + ipilimumab, selumetinib + dacarbazine and ipilimumab + sargramostin.

2.2 Data used in the Network plot

Data used in the Network plot is obtained from a published article[3]. OS data which included nine trials (10 treatments) was chosen. 10 treatments include chemotherapy, camrelizumab + chemotherapy, nivolumab + ipilimumab, nivolumab + ipilimumab + chemotherapy, atezolizumab + chemotherapy, bevacizumab + chemotherapy, atezolizumab + bevacizumab + chemotherapy, pembrolizumab + chemotherapy, sintilimab + chemotherapy, and nivolumab + bevacizumab + chemotherapy.

2.3 Data used in the AD-based NMA

Data used in the AD-based NMA is the aggregated data obtained from Anna Wiksten et.al[4]. The original source is a study published by Jansen[5]. The example consists of survival data from 7 studies and 4 treatments (docetaxel, pemetrexed, gefitinib, and best supportive care). Any question on how to get the aggregated data please refer to Jansen's study where has a detailed description.

3、Network Meta-Analysis of Survival data (NMAs)

3.1 Introduction of NMAs

This tool (NMAs) can be used to conduct network meta-analysis (NMA) based on reconstructed survival data. In this APP, we provided several methods including Aggregated data (AD)-based NMA and Individual patient data (IPD)-based NMA. Please note that not all existing NMA methods are included in this app, so please be careful when using this APP. Methods considered in this APP include Fractional polynomials (FP), Piecewise exponential model (PWE), Parametric survival model, Cox proportional hazard (PH) Model and Generalized Gamma Model. Some useful function like data transform from IPD to Aggregated data, Network plot and PH assumption tests. Some other flexible methods like Restricted mean survival time (RMST) and Royston-Parmar models (RP) are not included in this APP and may be available in the future.

Typically, like survival extrapolation, users can compare the performance of different models through the GOF statistics. In this tool, the GOF including AIC and Deviance information criterion (DIC) are provided. AIC has already been described in section 3.1. DIC is described by the following formula[5,6]:

$$DIC = \bar{D} + pD$$

$$pD = \bar{D} - \hat{D}$$

$\bar{D}$ is the posterior mean residual deviance, pD is the 'effective number of parameters' and $\hat{D}$ is the deviance evaluated at the posterior mean of the model parameters. Please note that sometimes AIC and DIC may not be applicable when comparing different types of models. For AD-based methods, hazard plots and survival plots can be drawn through which users can conduct the visual inspection. For IPD-based methods, only treatment effects are provided. But users can draw hazard plots and survival plots themselves based on the provided results. Anyway, the final model choice should be based on the users' final usage and several sensitivity analyses.

A standardized process is important for users. Currently, researchers have attempted to provide standardized processes for conducting the NMA[7,8]. Similarly, we will provide a brief guideline here. More detailed information can be found in elsewhere. Firstly, users should conduct the PH assumption tests. Typically, only no evidence of a violation of the PH assumption in any of the trials in the network means PH models can be considered. Secondly, evaluate fit of survival models for each trial and NMA models for the complete evidence base. Finally, users should choose the most suitable model and perform sensitivity analyses using. They should also assess whether models align with external constraints during the whole NMA process.

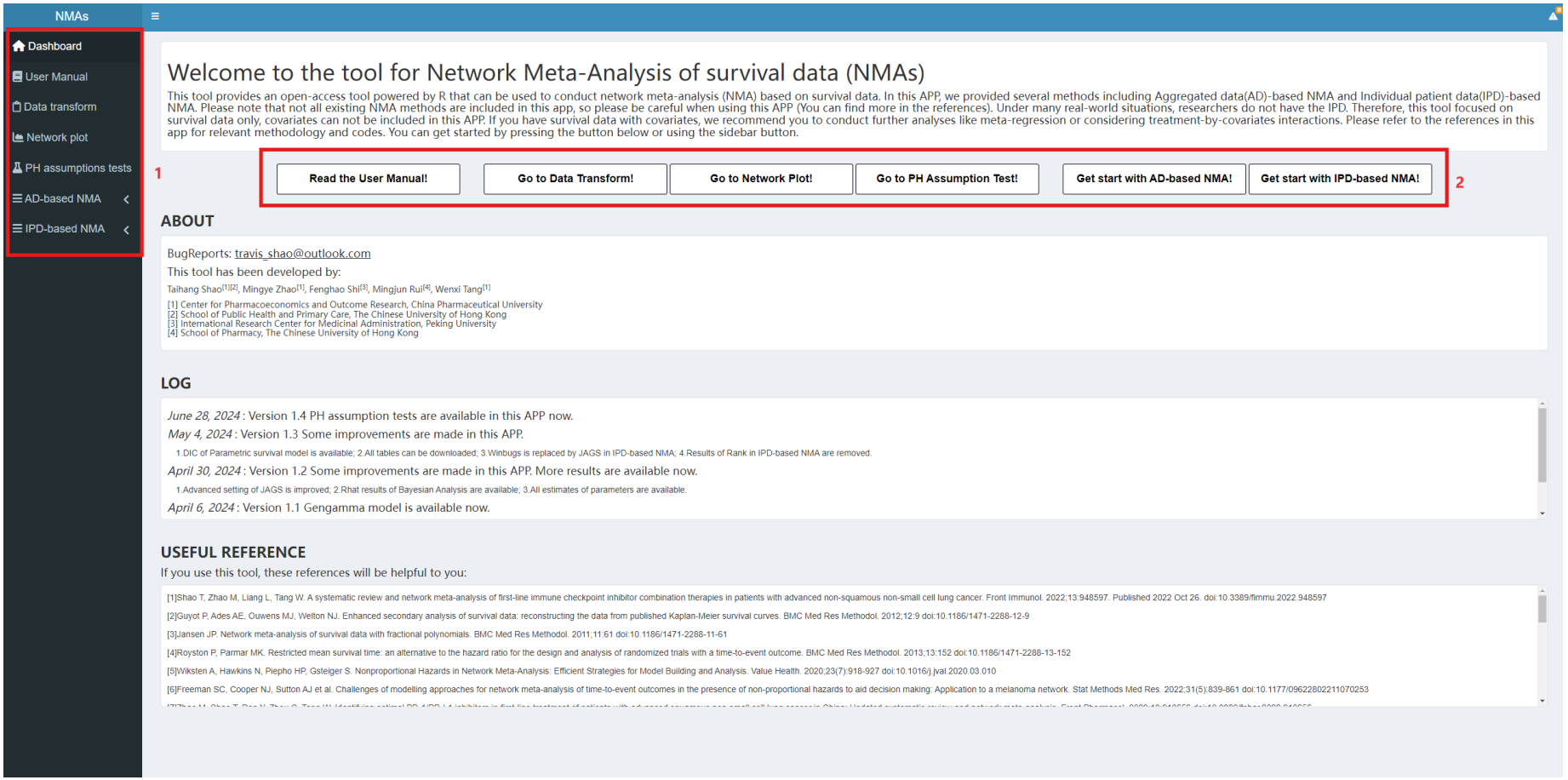

Figure 3-1 Dashboard of NMAs

3.2 Get started with NMAs

NMAs can be opened through the URL (https://psurvivala.shinyapps.io/NMAs/) directly. According to Figure 3-1, highlight position 1 is the sidebar of this tool, through which users can switch to different functions easily. Through the button showed in the highlight position 2, users can quickly move to the user manual, three small functions, and the data import section of AD-based NMA or IPD-based NMA. Please note that users should have some basic knowledge of NMA based on survival data if they want to use this tool. Thus, we provided references for some useful articles in the section "Useful Reference".

3.3 Data transform

This APP provide the function "Data transform", which can allow users to transform the IPD data into aggregated data. According to Figure 3-2, through the highlight position 1, users can upload their IPD or example data (in XLSX format). The imputed IPD data will be shown in highlight position 4, and the included treatment and reference number will be shown in highlight position 5. Examples of these two tables can be found in Table 3-1 and 3-2. In Table 3-1, **patid** is the ID of a patient in a trial; **time** is the survival time of a patient (in months); **event** refers to whether the patient is survival or dead; **arm** refers to the arm of the trial. In Table 3-2, the order of code arrangement is based on the order in which the target first appears. Typically, we use the code 1 for the reference treatment. Notably, the example IPD data used here is the same as the one used in the IPD-based NMA.

Table 3-1 Example IPD data

patid	time	event	arm	study	treatment
1	1.517747858	1	1	BREAK-3	DTIC
2	2.888616891	1	1	BREAK-3	DTIC
3	3.035495716	1	1	BREAK-3	DTIC
.....					
43	22.73584906	0	2	Robert 2013	SEL+DTIC
44	23.44339623	0	2	Robert 2013	SEL+DTIC
45	24.79245283	0	2	Robert 2013	SEL+DTIC

Table 3-2 Included treatment and reference number

Treatment_name	Treatment_code	Study_name	Study_code
DTIC	1	BREAK-3	1
.....			

Then users can use two input boxes which are shown in highlight position 2 to customize the generated aggregated data based on their own data. To be specific, **"The max timepoint"** refers to the longest start timepoint in the aggregated data. (In the melanoma network, (1) we want to set the start timepoint sequence as "0,3,6,9,12,15,18,~"; (2) and the longest follow-up time is 73 months; thus, we input 72 here). **"Step of the timepoint"** refers to the step of the start timepoint sequence. (Similarly, the sequence of "0,3,6,9,12,15,18,~", the step is 3 here). "The start timepoint" means the start time of

each time interval. (In the melanoma network, 0 is the start time of the time interval of 0-3,…,72 is the start time of the time interval of 72 to the end). After clicking the button shown in highlight position 3, users can get the aggregated data which is presented in the highlight position 6. An example can be found in Table 3-3. The obtained aggregated data can be downloaded through the button under the presented table. In Table 3-3, **studyn** is the reference number of specific study; **trtn** is the reference number of specific treatment; **time** refers the time of one specific observation (in months); **timeDelta** refers to the duration of time between two consecutive observations; **nevents** represents number of events; **natrisk** refers to number at risk. Please note that the data showed in Table 3-3 is sorted by time, a clearer example of aggregated data which is sorted by study and treatment will be shown in 3.6.1.

Table 3-3 Example aggregated data

studyn	trtn	time	timeDelta	nevents	natrisk	study	treatment
1	2	3	3	4	187	BREAK-3	DB
7	2	3	3	10	211	COMBI-d	DB
7	8	3	3	4	211	COMBI-d	DB+TR
.....							
12	12	71.92200557	2.92200557	0	6	Robert 2011	IPI+DTIC
4	6	69.41572489	0.41572489	0	8	CheckMate 067	NIV+IPI
12	1	73.05623259	1.05623259	0	1	Robert 2011	DTIC

The detailed codes of data transform will not be provided here. The method we used is Anova Parameterization, which is quite similar to the method of SC Freeman et.al.[1]. The main idea is to calculate the total events and censors in each given time intervals based on the imported IPD. An example code for Anova Parameterization can be found below. Reported total events and censors in RCTs are highly recommended to verify and calibrate the generated AD.

```
anova_data <- function(timepoints, timepoints2, ref.study=1, df){
     df2 <- survSplit(Surv(time, event) ~., data=df,
                    cut=timepoints, episode ="timegroup")
     df2$y <- df2$time - df2$tstart
     df2$n <- 1
     df3 <- summaryBy(y + event + n ~ timegroup + treatment + txCode + study + studyCode , FUN=c(sum,
max), data=df2)
     df3 <- subset(df3, select=-c(event.max, n.max))
     names(df3) <- c("spgrp", "treatment", "trtn", "study", "studyn", "y", "nevents", "natrisk",
"y.max")
     df3$start <- NA
     for(i in unique(df3$spgrp)){
          df3$start[df3$spgrp==i] <- timepoints2[i]
     }
     df3$time <- df3$start + df3$y.max
     return(df3)
}
```

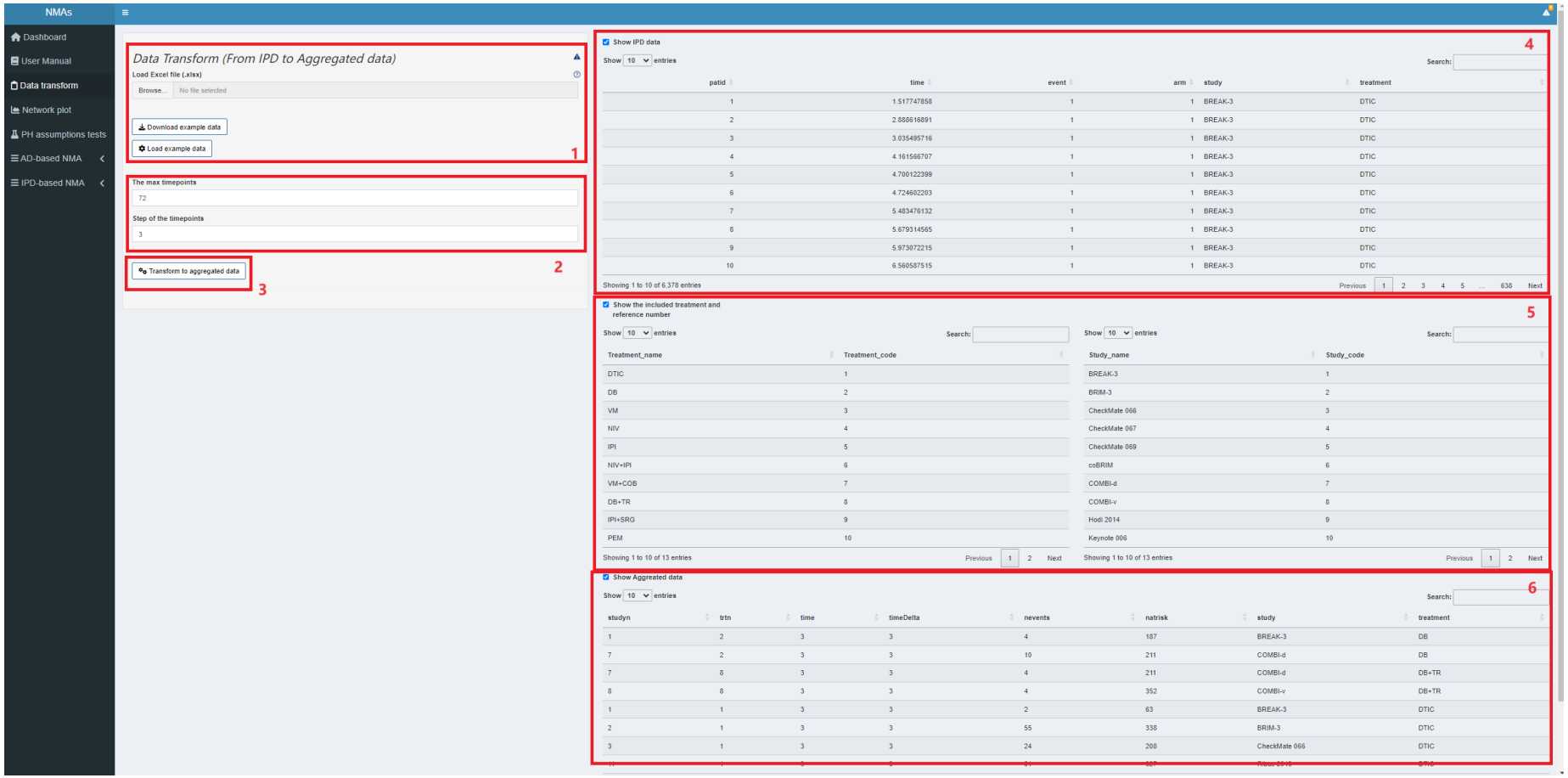

Figure 3-2 Section Data Transform

3.4 Network Plot

In section Network Plot, we allow users to draw a network plot based on their own needs. Please note that in this APP, we used the "*netgragh*" function in the "*netmeta*" package to draw a network plot. If users do not like the format of this plot, they can try other functions. For example, a simpler method like "*gemtc*" package or more difficult one like "*igraph*". An example network plot of "*netgragh*" will be shown as Figure 3-3.

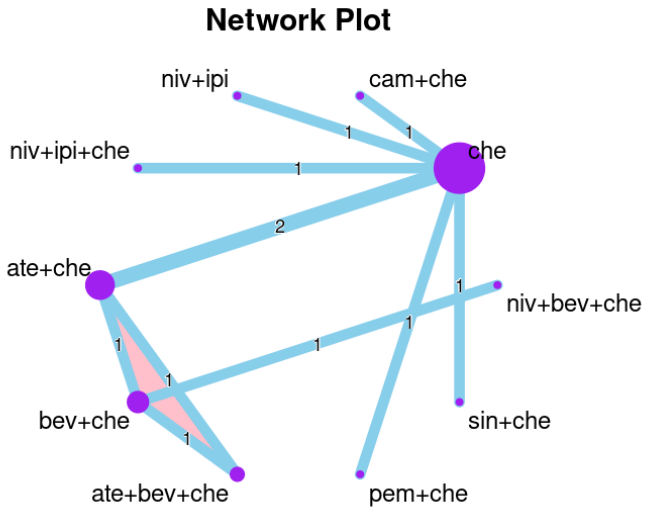

Figure 3-3 Example network plot

According to Figure 3-4, users should upload the data through the buttons in highlight position 1. The data input is shown in highlight position 3, which is similar to Table 3-4. In this table, **t1,t2** refers to the treatment code; **hr** with **lower** and **upper** refers to the hazard ratio with its upper and lower limits.

Table 3-4 Example data input of network plot

studyCode	Study	t1	t2	treat1	treat2	hr	lower	upper
1	CameL	1	2	che	cam+che	0.73	0.53	1.02
.....								
6	Impower150	6	5	bev+che	ate+che	0.84	0.71	1
6	Impower150	6	7	bev+che	ate+bev+che	0.8	0.67	0.95
6	Impower150	5	7	ate+che	ate+bev+che	0.95	0.75	1.22
.....								
9	Tasuki52	6	10	bev+che	niv+bev+che	0.85	0.63	1.14

The example data here is obtained from Shao et.al. (2022)[3]. This has been described in detail in section 2.4. This NSCLC (OS) network included 10 treatments and 6 studies. Since study 6 (Impower150) is a three-arm study, we will split it into 3 lines for input. Then, through the input boxes shown in highlight position 2, users can personalize this plot through modifying "**the title of the plot**", "**color of the line**", "**multi-arm or not**", "**color of the multi-arm area**", "**color of the points**", "**color of the interior of words**", and "**color of the edge of words**". Through pressing the "draw the plot" button, users can get and download the final plot in highlight position 4. The core codes we used to generate

the plot will be shown bellow:

```
a <- netmeta(data$lhr, data$se, treat1=data$t1, treat2=data$t2, studlab=data$Study, reference=1)
netgraph(a, labels = lab, offset = 0.02, plastic = F,
     col = ...,
     multiarm = ...,
     col.multiarm = ..., points = T,
     col.points = ..., number.of.studies = T, cex=2, cex.number.of.studies = 1.5,
     col.number.of.studies = ...,
     bg.number.of.studies = ..., cex.points = 2*c(count_trt$Freq))
title(..., adj=0.5,cex.main=2.5)
```

The summarized descriptions of function "*netgraph*" is shown in Table 3-5, a detailed one can be found through clicking the following URL: (https://www.rdocumentation.org/packages/netmeta/versions/2.8-2/topics/netgraph.netmeta).

Table 3-5 Summarized descriptions of function "netgraph"

Name	Description
a	An object of class netmeta (mandatory).
labels	An optional vector with treatment labels.
offset	Distance between edges (i.e. treatments) in graph and treatment labels for 2-D plots (value of 0.0175 corresponds to a difference of 1.75% of the range on x- and y-axis).
col	A single color (or vector of colors) for lines connecting treatments (edges) if argument plastic = FALSE. Length of the vector must be equal to the number of edges (see list element 'comparisons' in netmeta).
plastic	A logical indicating whether the appearance of the comparisons should be in '3D look' (not to be confused with argument dim).
multiarm	A logical indicating whether multi-arm studies should be marked in plot.
col.multiarm	Either a function from R package colorspace or grDevice to define colors for multi-arm studies or a character vector with colors to highlight multi-arm studies.
points	A logical indicating whether points should be printed at nodes (i.e. treatments) of the network graph.
cex.points, pch.points, col.points, bg.points	Corresponding size, type, color, and background color for points. Can be a vector with length equal to the number of treatments.
number.of.studies	A logical indicating whether number of studies should be added to network graph.
cex.number.of.studies	The magnification to be used for number of studies.
col.number.of.studies	Color for number of studies.
bg.number.of.studies	Color for shadow around number of studies.
cex	The magnification to be used for treatment labels.

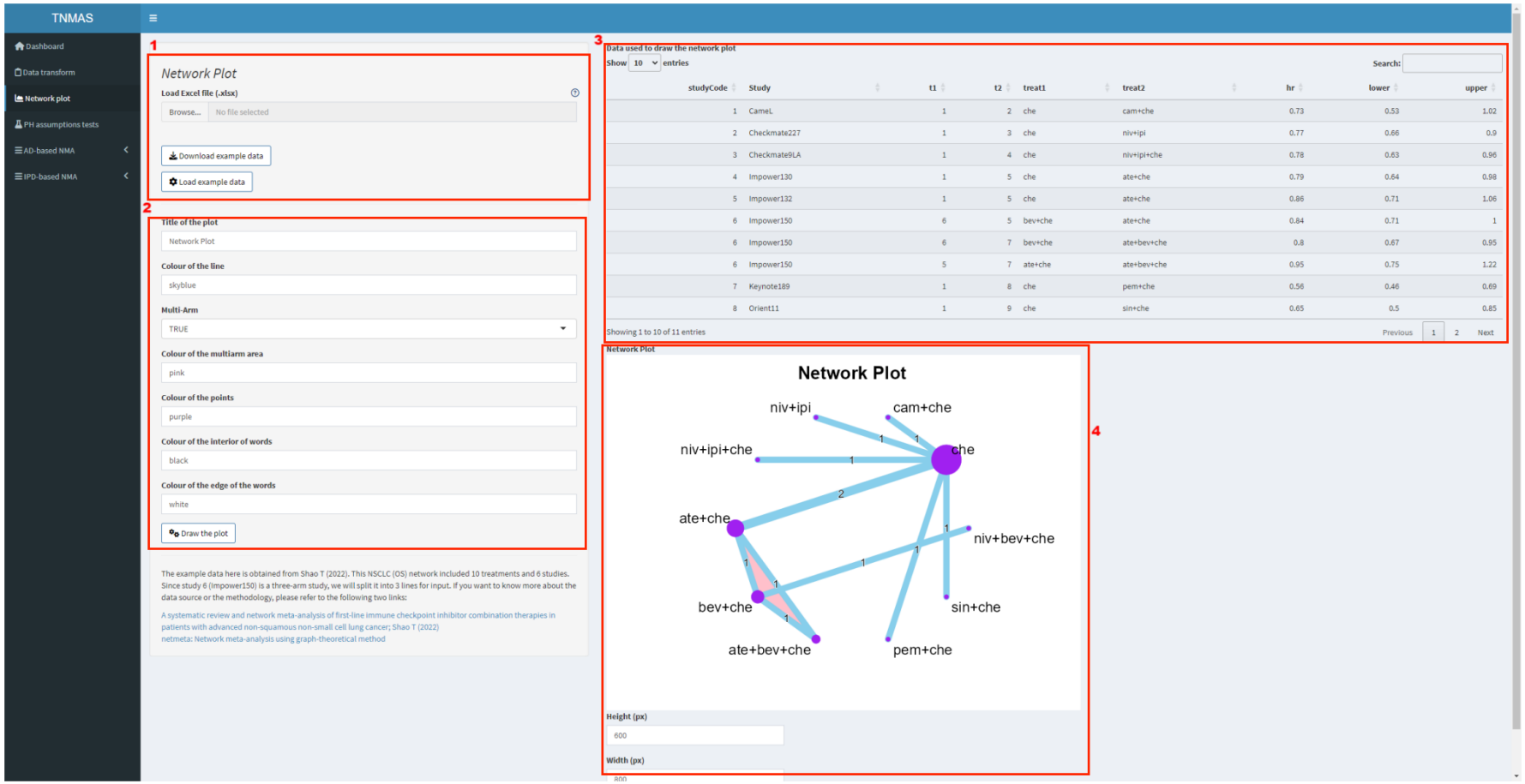

Figure 3-4 Section network plot

3.5 PH assumption tests

In section proportional hazards (PH) assumption tests, users can test whether the PH assumption is held between two arms. This APP provided three methods to test the PH assumption. They are Schoenfeld residual plot[9], Log-Log plot[1] and Grambsch-Therneau test[10]. Here is the summarized introduction for these three methods:

Schoenfeld residual plot

The Schoenfeld residual test primarily examines the proportional hazards assumption by analyzing the changes in Schoenfeld residuals over time. The basic idea is that if the proportional hazards assumption holds, the Schoenfeld residuals should be randomly distributed over time without showing any systematic trend. In this APP, user can get the P value through the Schoenfeld residual plot. Typically, if P value is smaller than 0.05, PH assumption is considered not held. An example plot can be found as Figure 3-5.

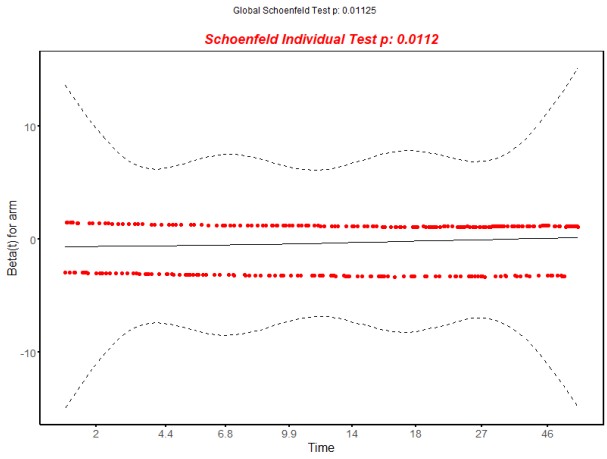

Figure 3-5 Example Schoenfeld residual plot

Grambsch-Therneau test

The Grambsch-Therneau test is a more formal statistical test based on Schoenfeld residuals. It evaluates the proportional hazards assumption by examining the correlation between Schoenfeld residuals and the ranks of time. In this APP, user can get the P value through the Grambsch-Therneau test. Typically, if P value is smaller than 0.05, PH assumption is considered not held. An example results table can be found as Table 3-6. In Table 3-6, **chisq** refers to the chi-square; **df** refers to the degree of freedom; **p** means the p value.

Table 3-6 Example Grambsch-Therneau test results

	chisq	df	p
arm	6.43	1	0.011
GLOBAL	6.43	1	0.011

Log-Log plot

If the PH assumption holds, the log-cumulative hazard curves for different groups should be parallel. Through visual inspection, if two lines are not parallel, PH assumption is considered not held. An example plot can be found as Figure 3-6.

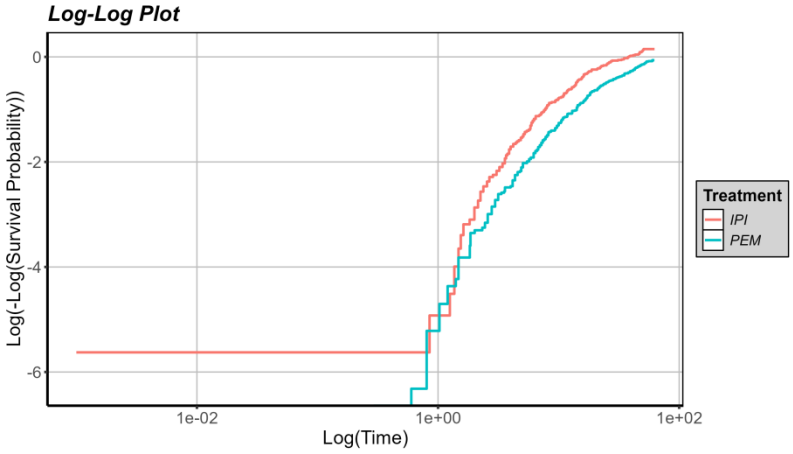

Figure 3-6 Example Log-Log plot

Usually, PH assumption uncertain in > 1 trials in the network, users should consider NMA that do not rely on PH assumption; No evidence of a violation of the PH assumption in any of the trials in the network, users can consider PH models.

According to Figure 3-7, users can upload their own data or example data through the buttons in highlight position 1. The example data here is obtained from the trial Keynote 006 of the melanoma network. A full description of melanoma network can be found in section 2.3. The example data is shown as Table 3-7, which is the same as the table in highlight position 3. Through the selectinput box and the button in highlight position 2, users can get and download the PH assumption tests results in highlight position 4.

Table 3-7 Example data for PH test

patid	time	event	arm	study	treatment
1	0.001	1	1	Keynote 006	IPI
2	0.850111857	1	1	Keynote 006	IPI
3	1.252796421	1	1	Keynote 006	IPI
.....					
554	62.28187919	0	2	Keynote 006	PEM
555	62.28187919	0	2	Keynote 006	PEM
556	62.28187919	0	2	Keynote 006	PEM

Here we provide the core codes for PH assumption tests through package "*survival*" and "*survminer*". Please note that the plots in this APP are redrawn through "*ggplot2*". Through the following codes, users can only get the raw plots.

```
fit <- coxph(formula = Surv(time, event) ~ arm, data = data)
res1 <- cox.zph(fit)
ggcoxzph(res1)
s1 <- Surv(data$time, data$event)
plot(survfit(s1 ~ data$txCode), col=c("blue", "red"), fun="cloglog", xlab="Log(time)",
ylab="log(-log(survival probability))")
print(res1)
```

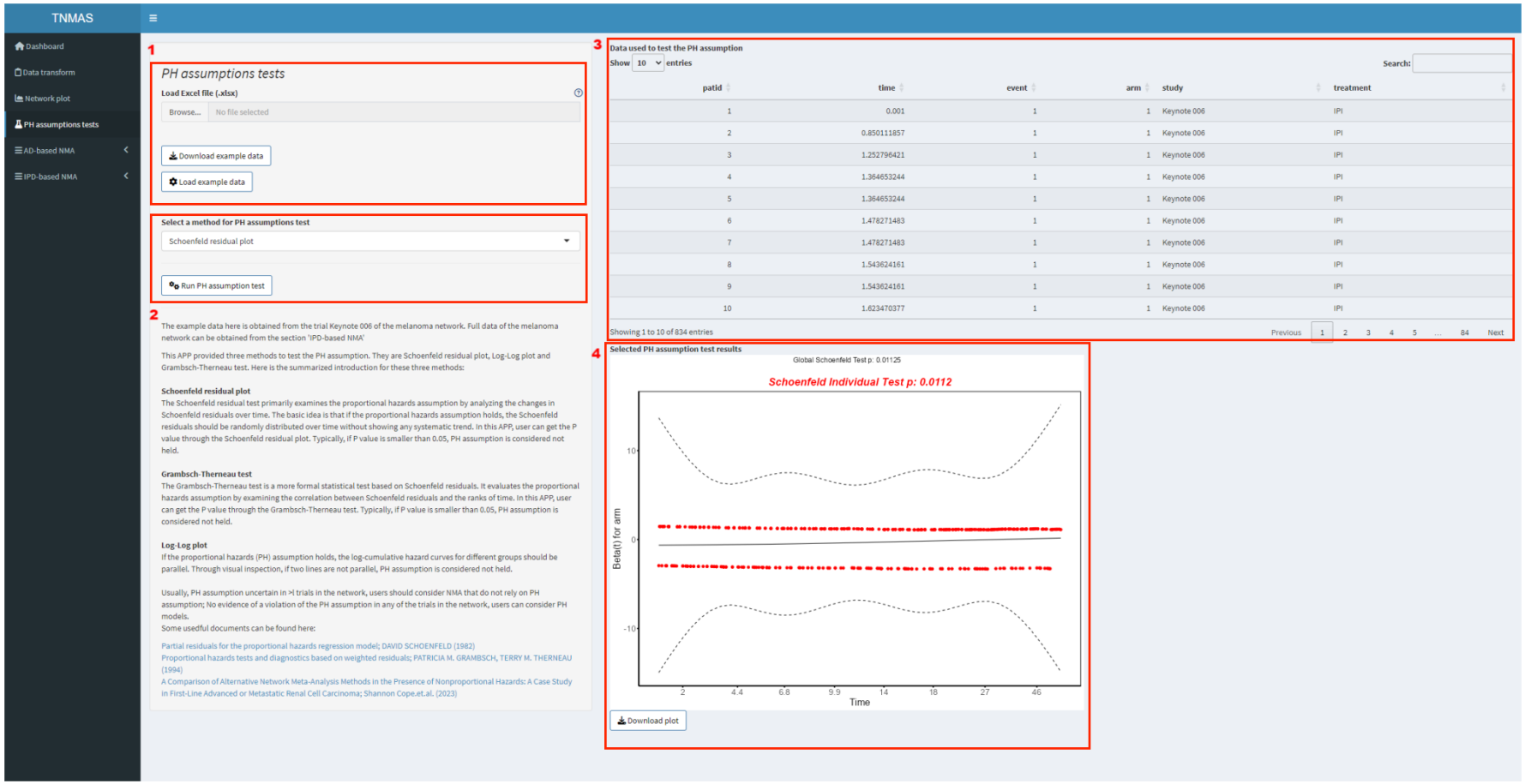

Figure 3-7 Section PH assumption tests

3.6 AD-based NMA

3.6.1 Import data

In section Import data, users can either use the example data to experience this tool, or upload their own data. According to Figure 3-8, users can finish these procedures in highlight position 1. Through the buttons in highlight position 2, users can move to different AD-based NMA quickly after importing the data. The data import can be found in the tables in highlight position 3. A example data will be presented as Table 3-8. **time** refers the time of one specific observation (in months); **timeDelta** refers to the duration of time between two consecutive observations; **nevents** represents number of events; **natrisk** refers to number at risk.

Table 3-8 Example data input of AD-based NMA

time	timeDelta	nevents	natrisk	study	treatment
2	2	7	81	Lee2010	Gefitinib
4	2	5	74	Lee2010	Gefitinib
6	2	7	69	Lee2010	Gefitinib
.....					
14	2	55	221	Shepherd2000	Docetaxel
16	2	41	166	Shepherd2000	Docetaxel
18	2	30	125	Shepherd2000	Docetaxel

The example aggregated data here is obtained from Jansen JP (2011)[5]. This NSCLC network included 4 treatments and 7 studies. A detailed description can be found in section 2.5. Treatment 1 is the reference treatment. Please note that the reference treatment should be set as treatment 1 (code 1) in NMAs.

We provided a useful note before running the model here: Some models may not be suitable for your data. (The fitted or extrapolated survival curve exhibits strange shapes.) For example, Piecewise exponential models have a key assumption that the treatment effects are proportional within a time interval. Thus, in some case, PWE do not perform well. In addition, PWE may not perform well when extrapolating survival curves beyond the observed data. In order to find the most suitable, users should try different models with different parameters input, and use methods like statistical indicators or visual inspection to select models. Besides, users should refer to the section 4.1 and follow some guidelines when selecting models.

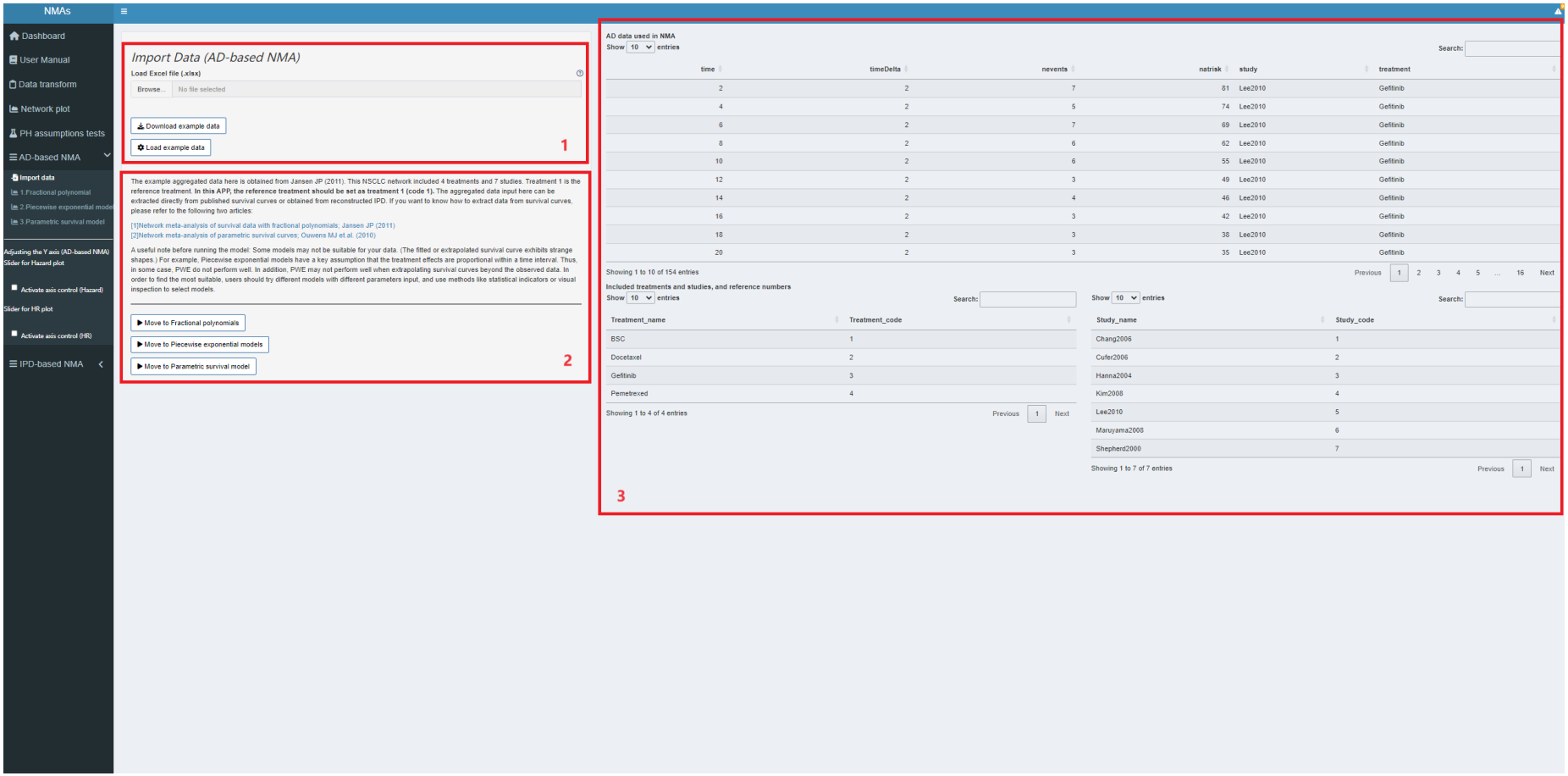

Figure 3-8 Section Import Data in AD-based NMA

3.6.2 Fractional polynomials

The detailed methodology of Fractional polynomials (FP) has been described in section3.5.1 and will not be discussed here. In this FP NMA section, we use the framework of Wiksten et al.(2020)[4]. We propose a two-step process for fitting FP models. In the first step, an ANOVA-like parameterisation is used to express and fit the models as GLM with time-varying covariates in a frequentist framework. The fit of the models in terms of the AIC is compared. The model with the lowest AIC can be selected to fit in the Bayesian setting (or Frequentist setting) in the second step. Please note that the core methodology of frequentist analysis in step one and two are the same.

For fixed effect model, we provide the parameter estimates (**treatment effects, d and mu**). Hazard plot and survival plot are also drawn based on these parameters. Using the difference in Beta and the Beta of reference treatment, users can calculate other Beta values. Then, hazard over time for each of the interventions can be calculated through the function with Beta. In addition, through **d** (**trtf** in Frequentist setting), Hazard Ratio between selected treatments can be calculated. This process can also be realized easily through EXCEL. For Example: For reference treatment a and intervention b, we construct a FP1 model with power = -2. We get d_{0ab}, d_{1ab}, $Beta_{0a}$, $Beta_{1a}$, $Beta_{0b}$, $Beta_{1b}$.
Note:

$$Beta_{0b}=Beta_{0a}+d_{0ab};\ Beta_{1b}=Beta_{1a}+d_{1ab}$$

Thus,

$$Log(HR_{ab}(t))=d_{0ab}+d_{1ab}*t^{-2},$$

$$Log(Hazard_a(t))=Beta_{0a}+Beta_{1a}*t^{-2},$$

$$Log(Hazard_b(t))=Beta_{0b}+Beta_{1b}*t^{-2}$$

$$\begin{pmatrix}\beta_{0k}\\ \beta_{1k}\end{pmatrix}=\begin{cases}\begin{pmatrix}\mu_0\\ \mu_1\end{pmatrix} & if\ k=a\\ \begin{pmatrix}\mu_0\\ \mu_1\end{pmatrix}+\begin{pmatrix}d_0\\ d_1\end{pmatrix} & if\ k=b\end{cases}$$

Note:$Beta_{0a}$, $Beta_{0b}$ are usually calculated as average from study specific estimates(μ_0, μ_1, μ_2) of the reference treatment[4,5]. One alternative is to fit and extrapolate the selected reference treatment with FP model to calculate the Beta. However, biases may be introduced through this method.

For random effect model, we provide the parameter estimates only. Plots can be drawn through the same methods. In addition, we only consider the model with a heterogeneity parameter for d_0 in random effect model. We do not consider all heterogeneity parameters (d_0, d_1, d_2 in FP2) since we believe that the methodology still needs further development[5]. (Bayesian analysis using the random-effects model would require careful specification of the prior for between-study heterogeneity. Also, maximum likelihood estimation can be problematic at this time, in which the number of parameters increases with the number of studies.) An example of Bugs codes for random effect FP models with all heterogeneity parameters can be found in Jansen JP (2011)[5]. In Jansen's study, two kinds of random effect models showed similar fit results.

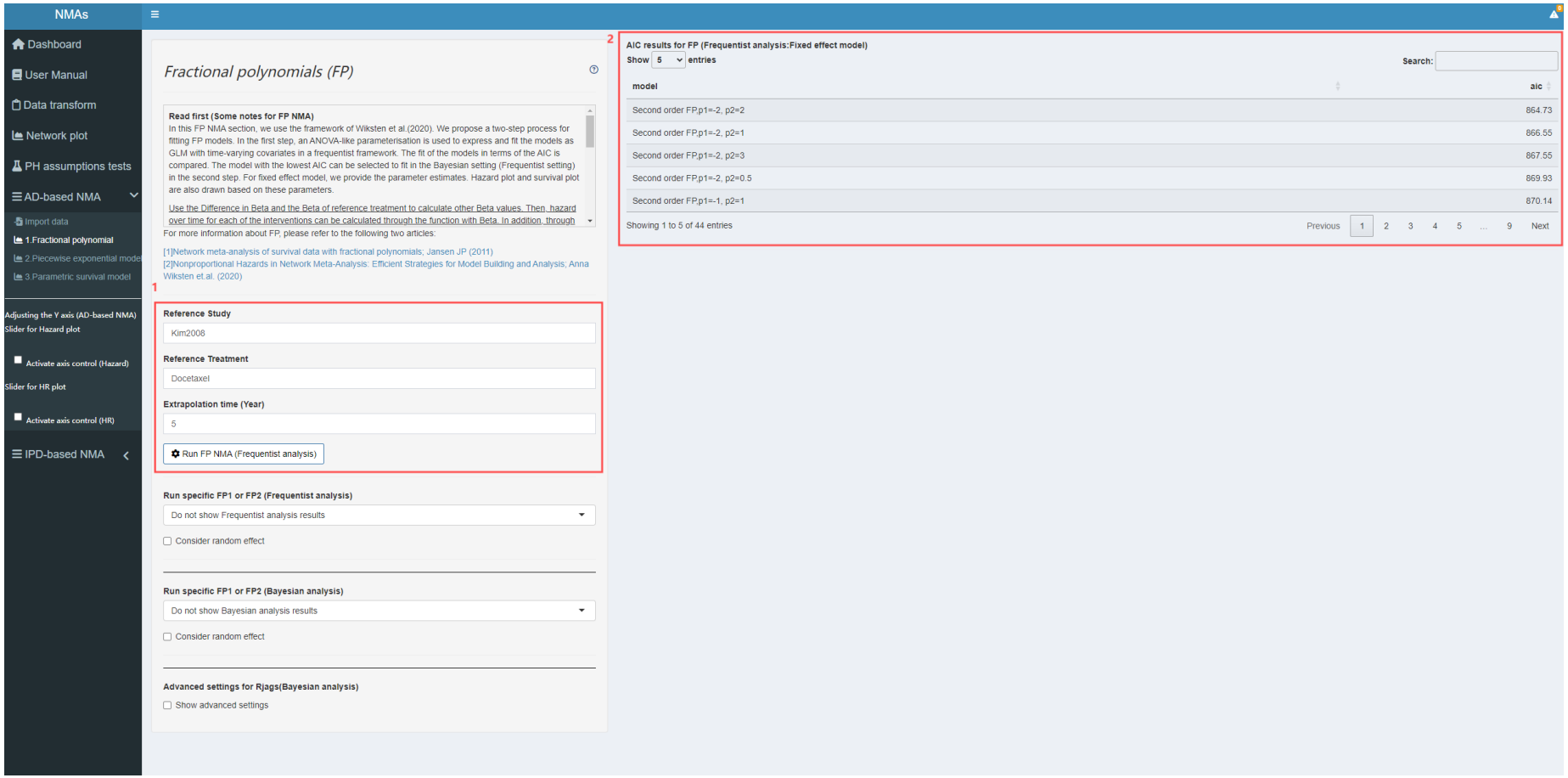

Figure 3-9 Section FP 1 (AD-based NMA)

According to the Figure 3-9, through the input box in highlight position, users can input their **Reference Study**, **Reference Treatment** and **Extrapolation Time**. Then, after pressing the "Run FP NMA (Frequentist analysis)", users can get the results of step one ("frequentist framework"), which are presented in highlight position 2.

After selecting the best FP1 and FP2 models, users can move to the "Run specific FP1 or FP2 (Frequentist analysis)" which is shown in highlight position 1 in Figure 3-10. Hazard plots, survival plots, HR plots and coefficient tables can be found in highlight position 2. Y-axis for hazard plots and HR plots can be found in highlight position 3. Please note that the sliders here can control all plots in this APP. Users can click on the checkbox "consider random effect" to run the random effect model. However, users should be aware that they should re-run the NMA model after they click on the checkbox.

In addition, after step one, users can also move to the "Run specific FP1 or FP2 (Bayesian analysis)", which is shown in highlight position 1 in Figure 3-11. Most procedures are similar to Frequentist analysis, and results can be found in highlight position 3. In highlight position 2, users can modify the settings for JAGS to run the Bayesian analysis. In this tool, "R2jags" was used. A detailed description of this package is available on (https://cran.r-project.org/web/packages/R2jags/R2jags.pdf). A brief description can be found in Table 3-9.

Table 3-9 Brief introduction of R2jags

Name	Description
data	(1) a vector or list of the names of the data objects used by the model, (2) a (named) list of the data objects themselves, or (3) the name of a "dump" format file containing the data objects, which must end in ".txt", see example below for details.
inits	a list with n.chains elements; each element of the list is itself a list of starting values for the BUGS model, or a function creating (possibly random) initial values. If inits is NULL, JAGS will generate initial values for parameters.
parameters.to.save	character vector of the names of the parameters to save which should be monitored.
model.file	file containing the model written in BUGS code. Alternatively, as in R2WinBUGS, model.file can be an R function that contains a BUGS model that is written to a temporary model file (see tempfile) using write.model
n.chains	number of Markov chains (default: 3)
n.iter	number of total iterations per chain (including burn in; default: 2000)
n.burnin	length of burn in, i.e. number of iterations to discard at the beginning. Default is n.iter/2, that is, discarding the first half of the simulations. If n.burnin is 0, jags() will run 100 iterations for adaption.
n.thin	thinning rate. Must be a positive integer. Set n.thin > 1 to save memory and computation time if n.iter is large. Default is max(1, floor(n.chains * (n.iter-n.burnin) / 1000)) which will only thin if there are at least 2000 simulations.

For Frequentist analysis results, this APP will provide estimated value, standard error, and upper and lower limits of treatment effects. For Bayesian analysis results, this APP will provide median and confidence interval for parameters **mu** (the parameters Beta of the "baseline" treatment) and **d** (the difference in $Beta_0$ and $Beta_1$ of the log hazard for treatment b relative to a). To understand these, users can refer to the formula mentioned earlier. This APP will also report the indicator of "**Rhat**",

which is used as convergence diagnostics for Markov Chains. "**Rhat**" compares the between- and within-chain estimates for model parameters and other univariate quantities of interest[11]. If chains have not mixed well (ie, the between- and within-chain estimates don't agree), R-hat is larger than 1. It is recommended to run at least four chains by default and only use the sample if R-hat is less than 1.05.

Here, we provide the codes for users to run the fixed effect FP NMA model[4]. Notably, we take the example of FP 2 with power -2 and 2. And only core codes are provided here:

```
## Codes for preparation
# set reference study and treatment
ref.study <- ...
ref.trt   <- ...
# import the data
data <- c(...)
studies <- c(...)
treatments <- c(...)
## Codes for frequentist analysis
# Model formula
models <- list("Second order FP,p1=-2, p2=2" = list(g1=function(x){x^-2}, g2=function(x){x^2},
f1=function(x){x^-2}, f2=function(x){x^2}))
# Fit the model
fit.KM.NMA<-function(bf){
     data.new=data
     data.new$g0=1
     data.new$f0=1
     data.new$g1=bf[[1]](data.new$time)
     data.new$g2=bf[[2]](data.new$time)
     data.new$f1=bf[[3]](data.new$time)
     data.new$f2=bf[[4]](data.new$time)
     f=cbind(nevents,natrisk-nevents)~trtf*f0+studyf*g0+trtf*f1+trtf*f2+studyf*g1+studyf*g2
     glm(f,family=binomial(link=cloglog),data=data.new,offset = log(timeDelta))
}
fits=lapply(models,fit.KM.NMA)

## Codes for preparing data for JAGS
d_arms <- data %>%
  group_by(studyn, trtn) %>%
  slice(1) %>%
  group_by(studyn) %>%
  dplyr::mutate(arm = 1:n(), n_arms = max(arm)) %>%
  select(studyf, trtf, studyn, trtn, arm, n_arms)
d_arms
```

```
d_std <- d_arms %>%
  group_by(studyn) %>%
  select(studyn, n_arms) %>%
  slice(1)
d_std

dat <- km %>%
  left_join(d_arms, by = c("studyf", "trtf", "studyn", "trtn"))

d_trts <- dat %>%
  mutate(studyn.arm = interaction(studyn, arm)) %>%
  filter(!duplicated(studyn.arm)) %>%
  select(studyn, arm, trtn) %>%
  arrange(studyn, arm) %>%
  tidyr::spread(key = arm, trtn, drop = FALSE)
d_trts

Nobs <- nrow(dat)
dat_jg <- list(Nobs = Nobs, Ns = nrow(d_std), Na = d_std$n_arms, r = dat$nevents, n = dat$natrisk,
time = dat$time, dt = dat$timeDelta, s = dat$studyn, a = dat$arm, t = as.matrix(select(ungroup(d_trts),
-studyn)), Ntrt = max(select(ungroup(d_trts), -studyn), na.rm = TRUE))

## Codes for Bayesian analysis
model.pars <- list(P1 = -2, P2=1)
set.seed(...)
fit <- jags(model.file = "...txt",
     data = c(dat_jg,
          list(prior.mean = rep(0, 3)),
          list(prior.prec = diag(rep(0.0001, 3))),
          model.pars),
     parameters = c("d", "mu", "Beta"),
     n.chains = 3, n.iter = 200000, n.burnin = 50000, n.thin = 3)
```

Please note that codes for random effect model are similar to fixed effect model. Thus, we will not provide them here. JAGS codes for running fixed effect FP2 Bayesian analysis will be provided in the Appendix.

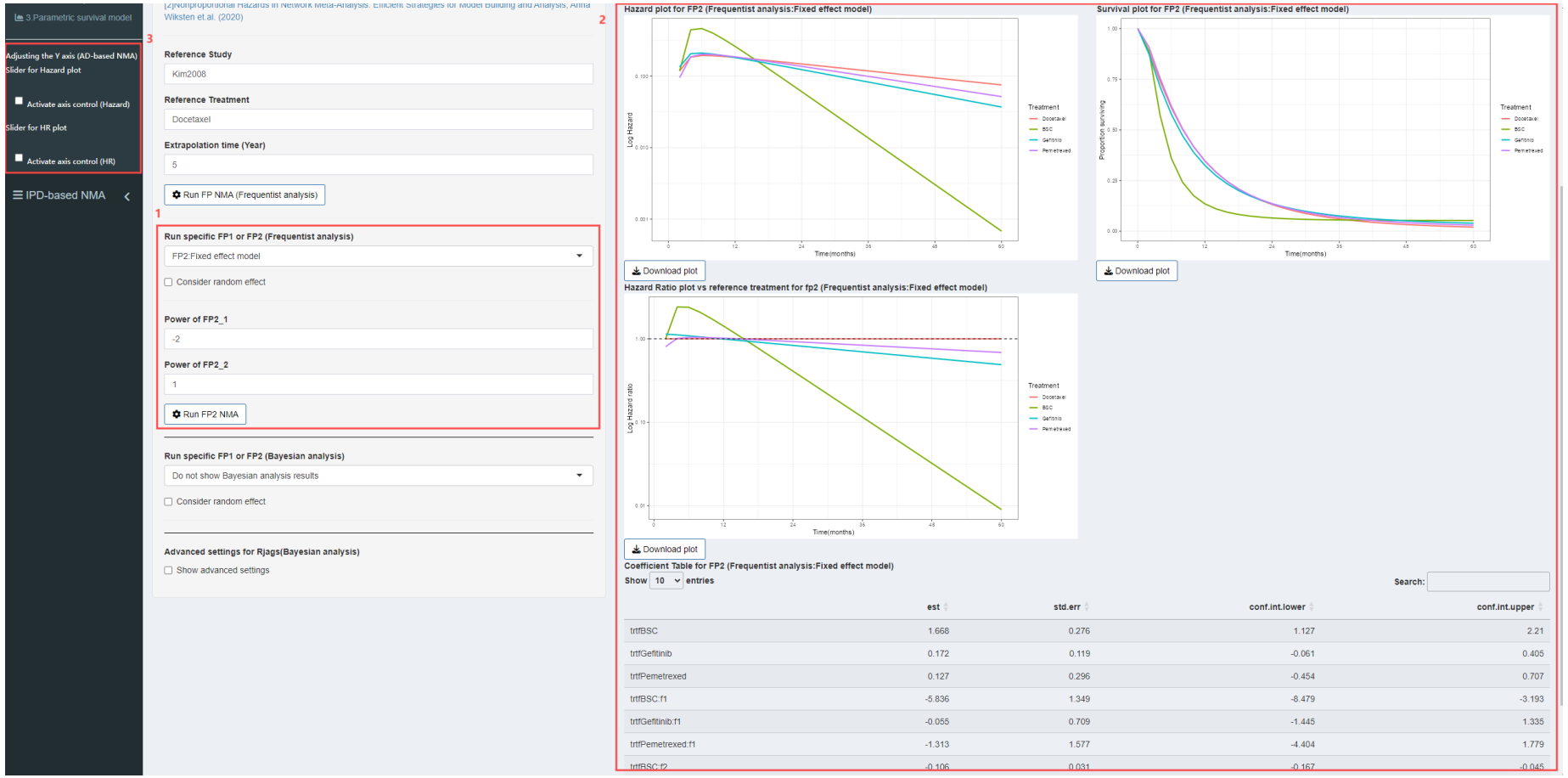

Figure 3-10 Section FP 2 (AD-based NMA)

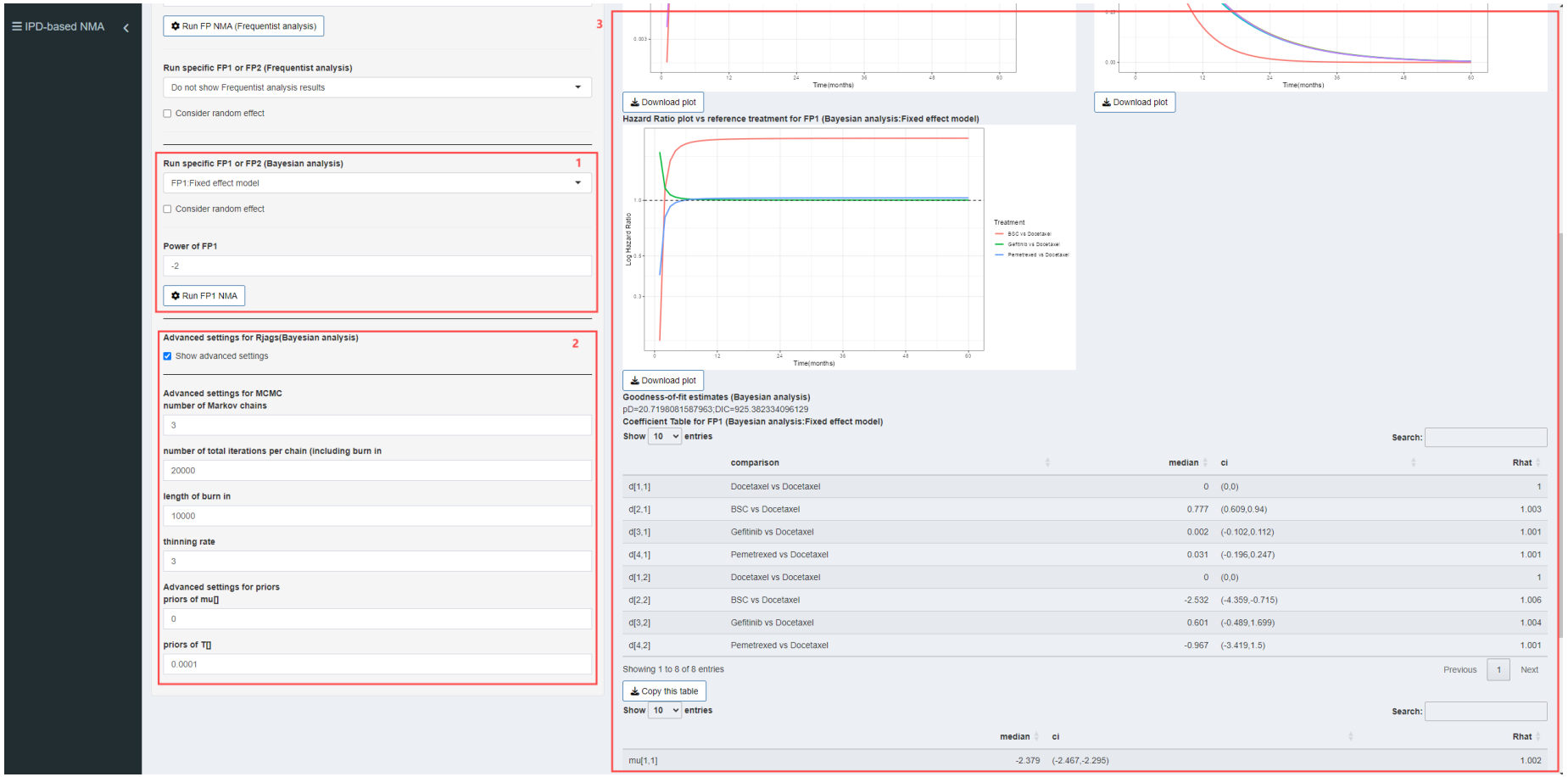

Figure 3-11 Section FP 3 (AD-based NMA)

3.6.3 Piecewise Exponential Model

The framework of Piecewise Exponential Model (PWE) model is similar to that of FP. Users can use two steps to build a PWE model. A PWE function can also be written in the form of ANOVA-like parameterisation, so that it can be fitted in the GLM framework[4,12]. Currently, only fixed effect models are available for PWE NMA. The choice of where to place cut points and how many cut points could result in many models being fitted before the best model can be selected. We recommend users to use Frequentist analysis to select suitable models before moving to Bayesian analysis.

Using the difference in Beta and the Beta of reference treatment, users can calculate other Beta values. Then, hazard over time for each of the interventions can be calculated through the function with Beta. In addition, through **d** (**trtf** in Frequentist setting), Hazard Ratio between selected treatments can be calculated. This process can also be realized easily through EXCEL. For Example: For reference treatment a and intervention b, we construct a PWE model with time point = 2. We get d_{0ab}, d_{1ab}, $Beta_{0a}$, $Beta_{1a}$, $Beta_{0b}$, $Beta_{1b}$.
Note:

$$Beta_{0b}=Beta_{0a}+d_{0ab};\ Beta_{1b}=Beta_{1a}+d_{1ab}-\ Beta_{0b}$$

Thus,

$$Log(HR_{ab}(t))=d_{0ab}(0<=t<2);\ d_{1ab}(t>=2),$$

$$Log(Hazard_a(t))=Beta_{0a}(0<=t<2);\ Beta_{1a}(t>=2),$$

$$Log(Hazard_b(t))=Beta_{0b}(0<=t<2);\ Beta_{1b}(t>=2).$$

Note:$Beta_{0a}$, $Beta_{0b}$ are usually calculated as average from study specific estimates (μ_0, μ_1, μ_2) of the reference treatment.

According to the Figure 3-12, through the input box in highlight position 1, users can input their **Reference Study**, **Reference Treatment** and **Extrapolation Time**. Then, users have to select to run which PWE model (Frequentist analysis) and input the Cutpoint of PWE in highlight position 2. After pressing the "Run PWE NMA", users can get the hazard plots, survival plots, HR plots and coefficient tables in highlight position 3. According to the Figure 3-13, users can select the PWE model (Bayesian analysis) and input the Cutpoints of PWE in highlight position 1. Hazard plots, survival plots, and coefficient tables are shown in highlight position 3. Similar to FP NMA, users can modify the settings for JAGS to run the Bayesian analysis in highlight position 2.

Here, we provide the codes for users to run the fixed effect PWE NMA model[4]. Notably, we take the example of PWE model with two cutpoints 2 and 12. The codes for data preparation and running Frequentist analysis and Bayesian analysis in PWE is quite similar to that in FP, thus we will not elaborate on it again. Users only need to make simple modifications. We only provide the codes for model formula in PWE here. JAGS codes for PWE with two cutpoints can be found in Appendix.

```
models <- list("PWE2" = list(g1=function(x){as.numeric(x > 2 & x <= 12)}, g2=function(x){as.numeric(x > 12)}, f1=function(x){as.numeric(x > 2 & x <= 12)}, f2=function(x){as.numeric(x > 12)}))
```

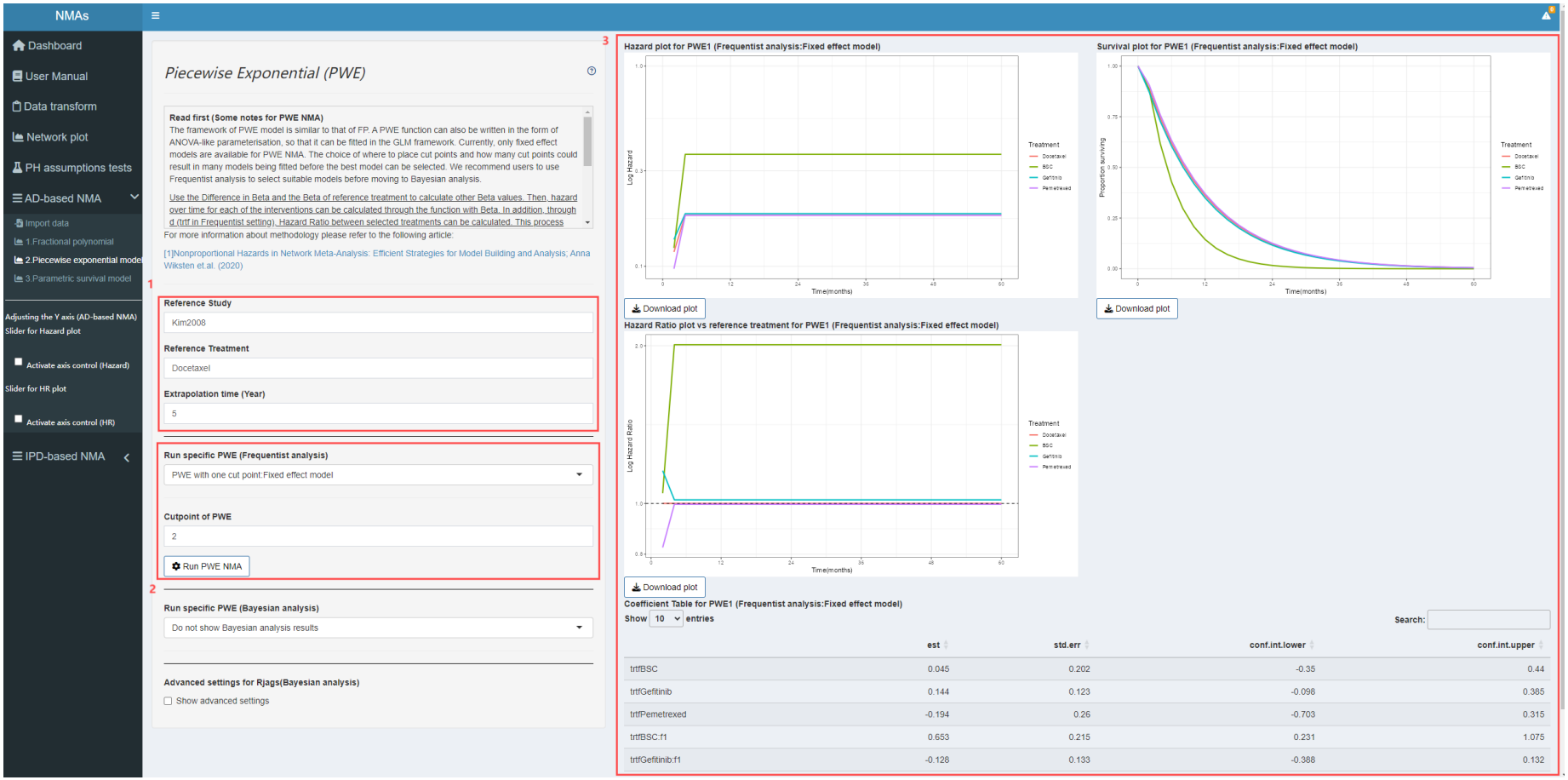

Figure 3-12 Section PWE 1 (AD-based NMA)

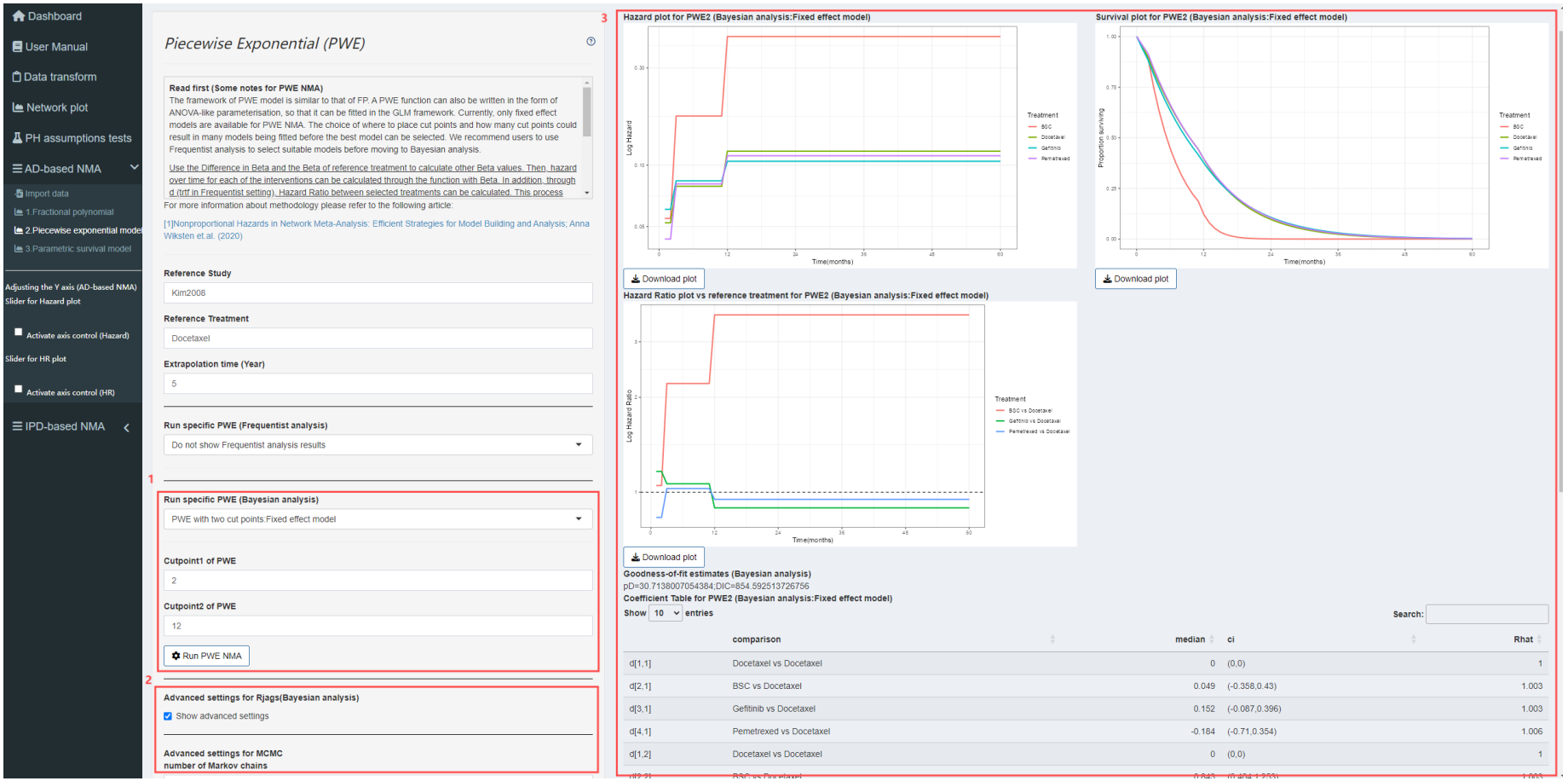

Figure 3-13 Section PWE 2 (AD-based NMA)

3.6.4 Parametric survival model

In the part of Parametric survival model (PSM), we only include Bayesian analysis (Fixed effect model and random effect model). Models including Weibull, Gompertz, log-logistic, log-normal are considered[8]. Use the Difference in Beta and the Beta of reference treatment to calculate other Beta values. Then, hazard over time for each of the interventions can be calculated through the function with Beta. In addition, through d, Hazard Ratio between selected treatments can be calculated. This process can also be realized easily through EXCEL. For Example: For reference treatment a and intervention b, we construct a PSM Weibull model. We get d_{0ab}, d_{1ab}, $Beta_{0a}$, $Beta_{1a}$, $Beta_{0b}$, $Beta_{1b}$.
Note:

$$Beta_{0b}=Beta_{0a}+d_{0ab};\ Beta_{1b}=Beta_{1a}+d_{1ab}$$

Thus,

$$Log(HR_{ab}(t))=d_{0ab}+d_{1ab}*log(t),$$

$$Log(Hazard_a(t))=Beta_{0a}+Beta_{1a}*log(t),$$

$$Log(Hazard_b(t))=Beta_{0b}+Beta_{1b}*log(t).$$

Note:$Beta_{0a}$, $Beta_{0b}$ are usually calculated as average from study specific estimates(μ_0, μ_1, μ_2) of the reference treatment. We provide the formulas for four distributions here:

Weibull:

$$f(t) = a + b * log(t)$$

Gompertz:

$$f(t) = a + b * t$$

Log-Logistic:

$$f(t) = log(\frac{\frac{e^b}{e^a} * (\frac{t}{e^a})^{e^b-1}}{1 + (\frac{t}{e^a})^{e^b}})$$

Log-Normal:

$$f(t) = \frac{(2\pi)^{-0.5} * e^{-\frac{(\frac{log(t)-a}{e^b})^2}{2}}}{e^b * t * pnorm(-\frac{log(t)-a}{e^b})}$$

Please note that FP1 with power=0 is equal to Weibull, while FP1 with power=1 is equal to Gompertz. For formula forms which is easier to understand, please refer to the description in the APP. For more information about these formulas, please refer to other studies[13-15]. According to the Figure 3-14, through the input box in highlight position 1, users can input their **Reference Study**, **Reference Treatment** and **Extrapolation Time**. Then, users can select to run which distribution (Bayesian analysis) and which model in highlight position 2. After pressing the "Run the model", users can get the hazard plots, survival plots, HR plots and coefficient tables in highlight position 4. Similar to FP and PWE NMA, users can modify the settings for JAGS to run the Bayesian analysis in highlight position 3.

Codes for PSM are quite similar to those for FP and PWE, thus we do not provide them here. JAGS codes for fixed effect PSM can be found in Appendix.

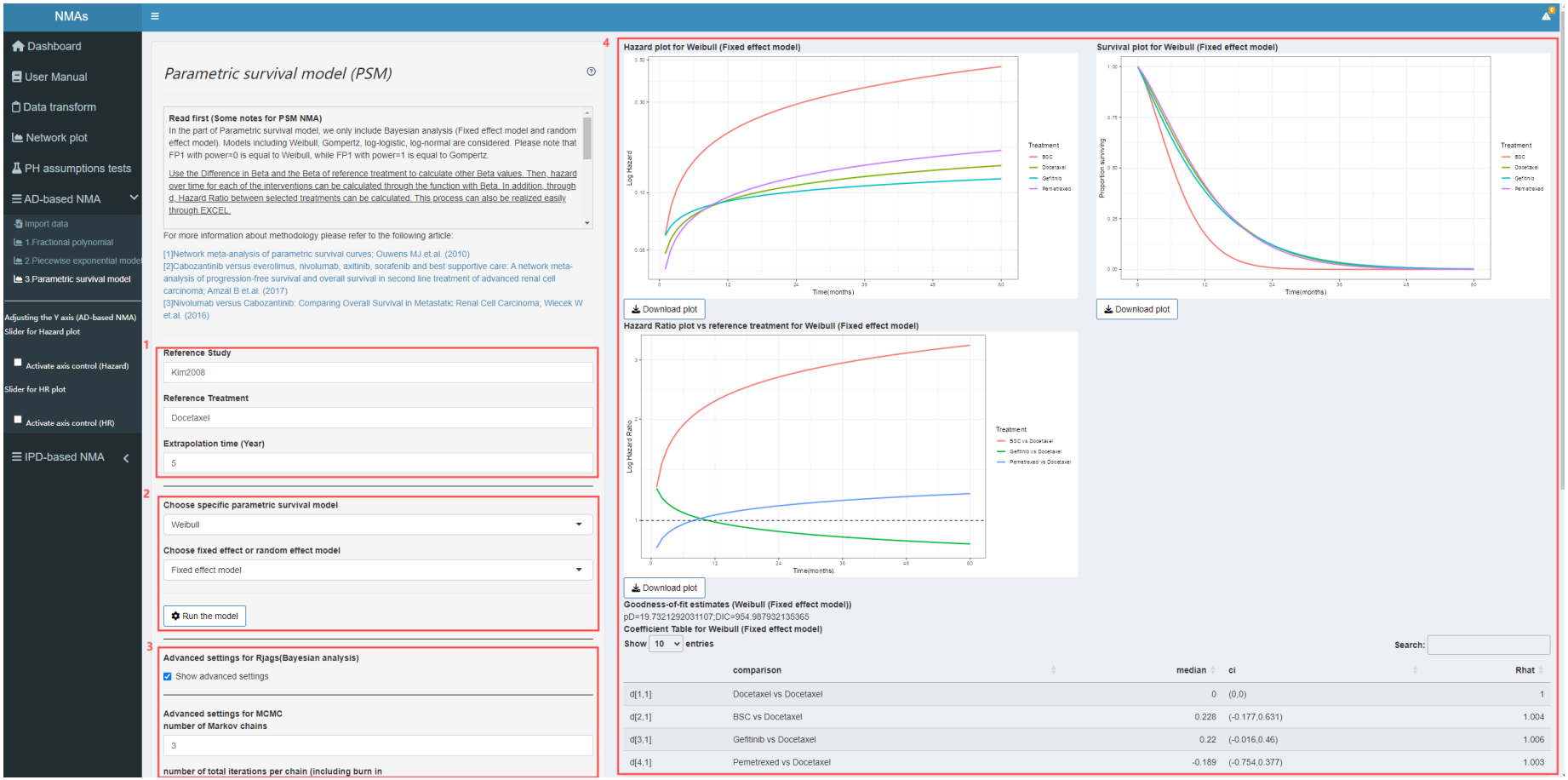

Figure 3-14 Section PSM (AD-based NMA)

3.7 IPD-based NMA

3.7.1 Import Data

The section Import data in IPD-based NMA is quite similar to that in AD-based NMA. Besides, the input example data is the melanoma network. Format of the input data is almost the same as that in the PH assumption test section. Detailed information of the melanoma network can be found in 2.3. Therefore, we do not provide further description here.

3.7.2 COX PH model

In the part of COX PH model, we only include Bayesian analysis (Fixed effect model). Treatment 1 (Code 1) is considered as the reference treatment. Changing the reference treatment can be realized through modifying the data.

The Cox PH model was fitted using a two-stage approach according to the method of Freeman SC et.al. (2022)[1]. In the first stage, a Cox PH model was fitted individually to each trial to obtain an estimate of the log HR for the treatment effect and its corresponding standard error[16].

$$h_{j,ab}(t) = h_{0j,ab}(t)\, exp(\, \alpha_{j,ab} x_{ij})$$

In this formula, $h_{j,ab}(t)$ is the hazard function for treatment *b* compared to the baseline treatment *a* in trial; *j*, $h_{0j,ab}(t)$ is the baseline hazard function for trial *j*; x_{ij} is the treatment indicator variable for patient *i* from trial *j*, taking the value 0 if patient *i* receives the baseline treatment *a* and the value 1 if patient *i* receives treatment *b*; and $\alpha_{j,ab}$ refers to the HR for a patient receiving treatment b compared to the baseline treatment *a* in trial *j*. In the second stage, the treatment effect estimate was synthesised through a standard fixed effect NMA model.

This APP will provide the results of HR (compared with the reference treatment). When HR is obtained, the user can calculate the cumulative hazard (*H(t)*) for reference treatment (written as **cumhaz_ref**). Then, multiplying HR with the **cumhaz_ref**, the cumulative hazard for each treatment (written as **cumhaz**) can be obtained. Finally, through formula '**exp(-cumhaz)**', the survival rate (*S(t)*) can be calculated.

$$HR_{\mathrm{ab},\,t_0 \to t_1} = \frac{\Delta H(t)_{b,\,t_0 \to t_1}}{\Delta H(t)_{a,\,t_0 \to t_1}}$$

$$H(t) = -\ln S\,(t)$$

Please note that in the first formula, t_0 to t_1 approaches zero infinitely. Typically, HR value is enough for common NMA. We provide the calculation of survival rate here in case some users need to conduct further research like cost-effectiveness analysis. The operation of COX PH model is quite simple. According to Figure 3-15, users only need to press the button in highlight position 1 to get the results of treatment effects in highlight position 3. Users can also modify the settings for JAGS to run the Bayesian analysis in highlight position 2.

To realize COX PH model, users have to get the hazard ratio for each trial. Detailed information of the methodology to obtain the data can be found elsewhere. Thus, we do not give any description here. In summary, this stage can be implemented using the "*coxph*" function from the "*survival*" package. After getting the data, users can use JAGS to run the fixed effect model to get the HRs. The codes to

run JAGS is similar to the AD-based NMA, and we will not discuss them here. JAGS code to run the COX PH NMA can be found in the Appendix. An example of data input of COX PH NMA can be found in Table 3-10.

Table 3-10 Example of data input of COX PH NMA

Study	Treatment1	Treatment2	HR	SE	Log HR	Variance[a]
BREAK-3	1	4	0.824	0.176	-0.193	NA
BRIM-3	1	6	0.799	0.087	-0.225	NA
CheckMate 066	1	9	0.463	0.125	-0.771	NA
CheckMate 069	8	10	0.754	0.271	-0.282	NA
COBRIM	6	7	0.687	0.127	-0.376	NA
COMBI-d	4	5	0.759	0.125	-0.276	NA
COMBI-v	5	6	1.404	0.135	0.339	NA
Hodi 2014	8	12	0.669	0.199	-0.402	NA
Keynote 006	8	11	0.730	0.095	-0.314	NA
Ribas 2013	1	2	0.878	0.088	-0.130	NA
Robert 2011	1	3	0.730	0.098	-0.315	NA
Robert 2013	1	13	0.828	0.247	-0.188	NA
CheckMate 067	8	9	0.665	0.102	-0.408	7.08E-09
CheckMate 067	8	10	0.545	0.107	-0.607	7.08E-09

a: variance of baseline treatment

Note: this example data is generated based on the melanoma network.

Users should be aware that this data is used for COX PH NMA, which can not be putted into the JAGS directly. An example of JAGS input can be found in section 4.6.2.

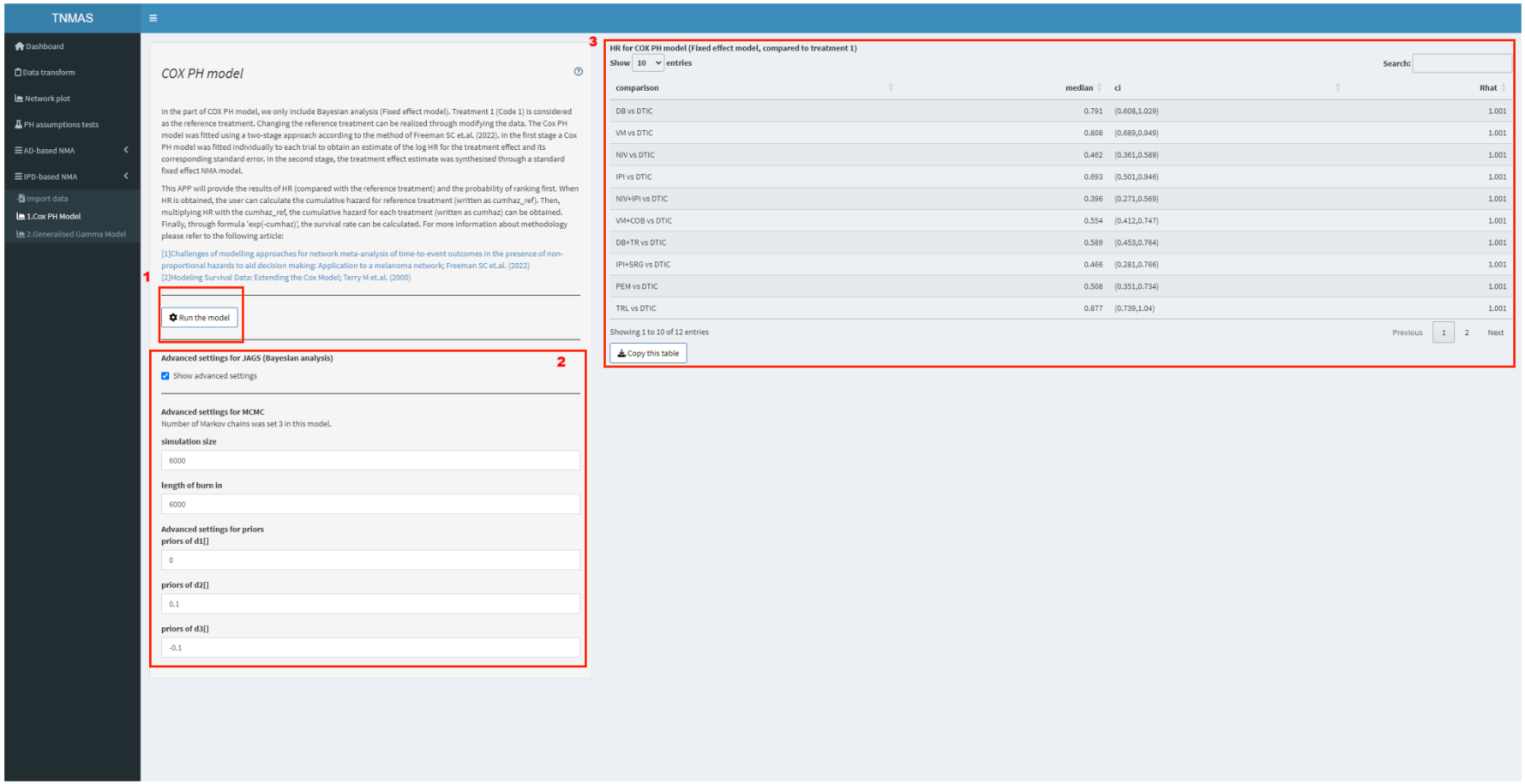

Figure 3-15 Section COX PH model (IPD-based NMA)

3.7.3 Generalised Gamma Model

Similar to COX PH Model, in the part of Generalised Gamma Model, we only include Bayesian analysis (Fixed effect model). Treatment 1 (Code 1) is considered as the reference treatment. Changing the reference treatment can be realized through modifying the data. The Generalised Gamma Model was fitted using a two-stage process according to the method of Freeman SC et.al. (2022)[1]. In the first stage, each trial was analyzed separately using the Generalised Gamma Model to obtain estimates of the log hazard ratio for the treatment effect and the corresponding standard error. Initially, each trial was independently evaluated by employing the Generalised Gamma Model. This approach facilitated the derivation of log hazard ratio estimates pertinent to the treatment effect, alongside the corresponding standard errors. Formula of Generalised Gamma Model (Q ≠ 0) can be found in section 3.4.1 and we provide it again here.

$$PDF = f(x|\mu,\sigma,Q) = \frac{|Q|(Q^{-2})^{Q^{-2}}}{\sigma t\Gamma(Q^{-2})} exp[\,Q^{-2}(Qw - e^{Qw})]$$

In this formula, $\gamma \sim G\text{amma}(Q^{-2},1)$, $\omega = log(\,Q^2\gamma)/Q$, $x = exp(\,\mu + \sigma\omega)$, *t* is survival time, μ is the location parameter, σ is the scale parameter and *Q* is the shape parameter[17]. Please note that in this model, the treatment effect was dependent on the location parameter only. In addition, in this model,

$$log(\,t_{ij}) = x_{ij}\alpha$$

x_{ij} is the treatment indicator variable for patient *i* from trial *j*, taking the value 0 if patient i receives the baseline treatment *a* and the value 1 if patient i receives treatment *b*. μ is the regression coefficient representing the treatment effect for treatment b compared to the baseline treatment a. In the second stage, the treatment effect estimate of the baseline treatment compared to treatment *i* in trial *j* was synthesised and its variability was esitmated within a standard fixed effect NMA model.

This APP will provide the results of Treatment Effects (compared with the reference treatment). The user can use "*flexsurv*" package to calculate the coefficicents (including **mu**, **sigma** and **q**) of applying Generalised Gamma Model on the reference treatment. To be specific, **mu** refers to μ, **sigma** refers to σ, and **q** refers to Q, respectively. When treatment effects is obtained, the user can calculate the survival rate (*S(t)*) through the coefficicents and treatment effects. Formula in R can be written as :

```
S(t)n = 1 - pgengamma(Time, mu = mu + TEn, sigma = sigma, Q=q, ...)
```

Similarly, hazard (*h(t)*) can be written as :

```
h(t) = hgengamma(Time, mu = mu + TEn, sigma = sigma, Q=q)
```

According to Figure 3-16, users can press the button in highlight position 1 so that they can get the results of treatment effects in highlight position 3. Users can also modify the settings for JAGS to run the Bayesian analysis in highlight position 2. Users should be care that the results are treatment effects instead of HR. If users want to get the time-varying HR, they can use the relationship between HR, cumulative hazard function and survival function.

Like COX PH model, users have to prepare the data for running Generalised Gamma Model NMA. Detailed information of the methodology to obtain the data can be found elsewhere. Thus, we do not give any description here. After getting the data, users can use JAGS to run the fixed effect model to get the treatment effects. JAGS code to run the Generalised Gamma Model NMA can be found in the Appendix. An example of data input of Generalised Gamma Model NMA can be found in Table 3-11.

Table 3-11 Example of data input of Generalised Gamma Model NMA

Study	Treatmetn1	Treatmetn2	Beta	SE	Covariance	Multi-arm
BREAK-3	DB	DTIC	0.157	0.196	NA	0
BRIM-3	VM	DTIC	0.508	0.106	NA	0
CheckMate 066	NIV	DTIC	0.847	0.164	NA	0
CheckMate 067	NIV	IPI	0.430	0.102	0.118	1
CheckMate 067	NIV+IPI	IPI	0.557	0.115	0.118	1
CheckMate 069	NIV+IPI	IPI	0.260	0.472	NA	0
coBRIM	VM+COB	VM	0.320	0.102	NA	0
COMBI-d	DB+TR	DB	0.311	0.130	NA	0
COMBI-v	VM	DB+TR	-0.234	0.090	NA	0
Hodi 2014	IPI+SRG	IPI	0.368	0.206	NA	0
Keynote 006	PEM	IPI	0.507	0.135	NA	0
Ribas 2013	TRL	DTIC	0.046	0.086	NA	0
Robert 2011	IPI+DTIC	DTIC	0.301	0.117	NA	0
Robert 2013	SEL+DTIC	DTIC	0.156	0.181	NA	0
CheckMate 067	IPI	IPI	0.000	0.118	NA	1

Note: this example data is generated based on the melanoma network.

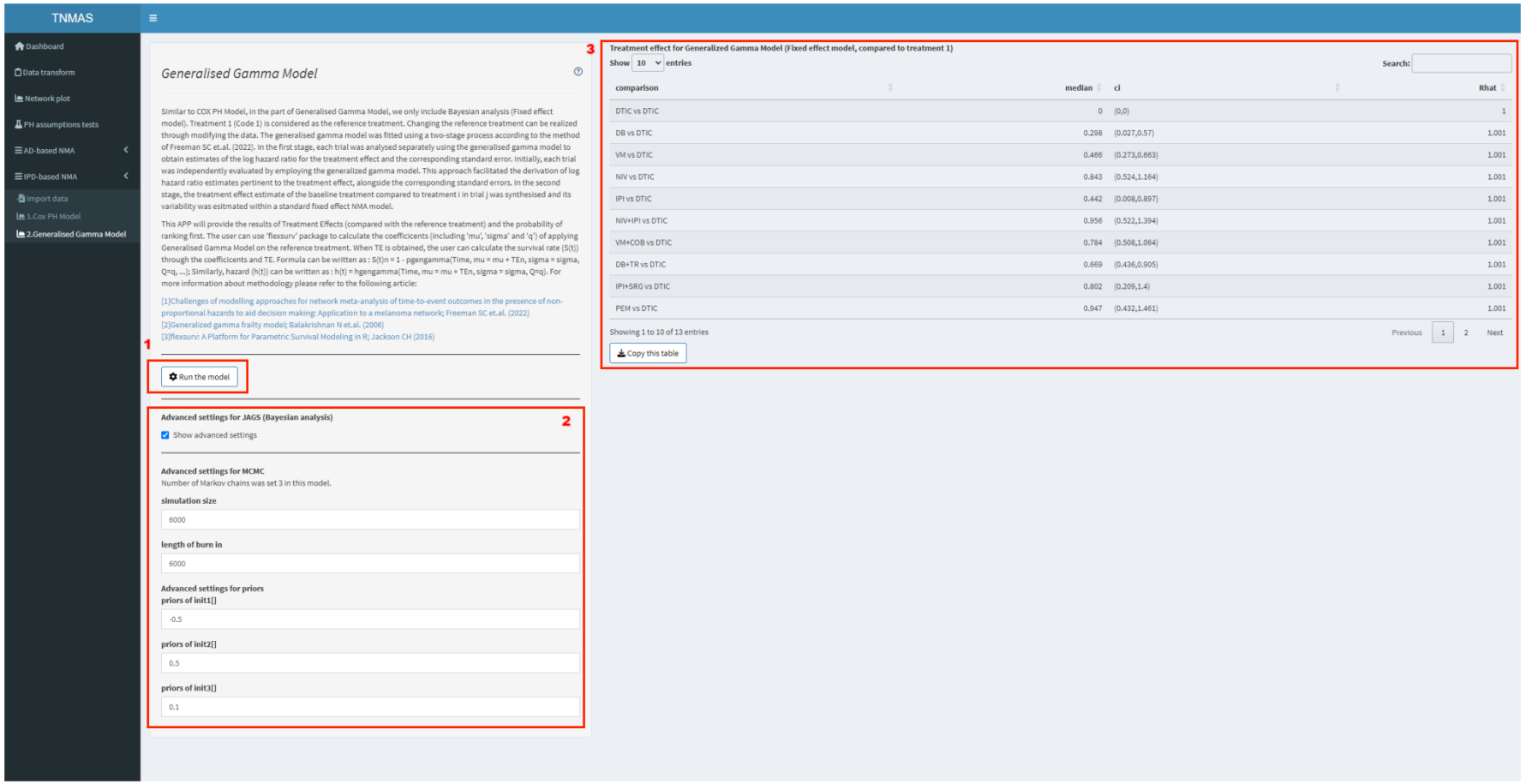

Figure 3-16 Section Generalised Gamma Model (IPD-based NMA)

4、Tips for users when using this APP

Before using NMAs, developers recommend users to read these tips. Firstly, when encountering errors, users can check the following items: (1) whether the data have been correctly inputted. (2) Check the parameters input. (3) Check the data. (4) Check the settings of JAGS. (5) There are some bugs in this APP. Bugs can be reported to the developers through email. If bugs can be found, we will be very grateful.

Secondly, users should be aware that this APP is designed to conduct NMA based on reconstructed survival data. Although users can get results without programming, they should program themselves if they have specific needs. In addition, this tool will not report any results for choosing models among different kinds of models. The reported results are limited to selecting within the same type of model. A rigorous and comprehensive step should be followed. When building NMA models, guidelines should always be paid attention to[7,8].

Finally, although users can use this APP without mastering the methodology of survival analysis and NMA, we strongly recommend users to engage in some learning to avoid selecting inappropriate models or obtaining incorrect results. Useful learning materials can be found through “Reference ” and URL in each page.

5、Reference

1. Freeman SC, Cooper NJ, Sutton AJ et al. Challenges of modelling approaches for network meta-analysis of time-to-event outcomes in the presence of non-proportional hazards to aid decision making: Application to a melanoma network. *Stat Methods Med Res*. 2022;31(5):839-861
2. Zoratti MJ, Devji T, Levine O, Thabane L, Xie F. Network meta-analysis of therapies for previously untreated advanced BRAF-mutated melanoma. *Cancer Treat Rev*. 2019;74:43-48
3. Shao T, Zhao M, Liang L, Tang W. A systematic review and network meta-analysis of first-line immune checkpoint inhibitor combination therapies in patients with advanced non-squamous non-small cell lung cancer. *Front Immunol*. 2022;13:948597
4. Wiksten A, Hawkins N, Piepho HP, Gsteiger S. Nonproportional Hazards in Network Meta-Analysis: Efficient Strategies for Model Building and Analysis. *Value Health*. 2020;23(7):918-927
5. Jansen JP. Network meta-analysis of survival data with fractional polynomials. *BMC Med Res Methodol*. 2011;11:61
6. Spiegelhalter DJ, Best NG, Carlin BP, Van Der Linde A. Bayesian measures of model complexity and fit. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*. 2002;64(4):583-639
7. The National Institute for Health and Care Excellence. CHTE2020 SOURCES AND SYNTHESIS OF EVIDENCE; UPDATE TO EVIDENCE SYNTHESIS METHODS; 2020. https://nicedsu.sites.sheffield.ac.uk/methods-development/chte2020-sources-and-synthesis-of-evidence
8. Cope S, Chan K, Campbell H et al. A Comparison of Alternative Network Meta-Analysis Methods in the Presence of Nonproportional Hazards: A Case Study in First-Line Advanced or Metastatic Renal Cell Carcinoma. *Value Health*. 2023;26(4):465-476
9. SCHOENFELD D. Partial residuals for the proportional hazards regression model. *Biometrika*. 1982;69(1):239-241
10. GRAMBSCH PM, THERNEAU TM. Proportional hazards tests and diagnostics based on weighted residuals. *Biometrika*. 1994;81(3):515-526
11. Aki V, Andrew G, Daniel S, Bob C, Paul-Christian B. Rank-Normalization, Folding, and Localization: An Improved $\widehat{R}$ for Assessing Convergence of MCMC (with Discussion). *Bayesian Anal*. 2021;16(2):667-718
12. Li Y, Gail MH, Preston DL, Graubard BI, Lubin JH. Piecewise exponential survival times and analysis of case-cohort data. *Stat Med*. 2012;31(13):1361-8
13. Amzal B, Fu S, Meng J, Lister J, Karcher H. Cabozantinib versus everolimus, nivolumab, axitinib, sorafenib and best supportive care: A network meta-analysis of progression-free survival and overall survival in second line treatment of advanced renal cell carcinoma. *PLoS One*. 2017;12(9):e0184423
14. Wiecek W, Karcher H. Nivolumab versus Cabozantinib: Comparing Overall Survival in Metastatic Renal Cell Carcinoma. *PLoS One*. 2016;11(6):e0155389
15. Ouwens MJ, Philips Z, Jansen JP. Network meta-analysis of parametric survival curves. *Res Synth Methods*. 2010;1(3-4):258-71
16. Harrell FE. Cox Proportional Hazards Regression Model. 2001:465-507
17. Cox C, Chu H, Schneider MF, Munoz A. Parametric survival analysis and taxonomy of hazard functions for the generalized gamma distribution. *Stat Med*. 2007;26(23):4352-74

6、Appendix

6.1 JAGS codes

6.1.1 JAGS codes for running FP2 NMA (fixed effect model)

```
model{
## Sampling model
for (i in 1:Nobs){
time1[i] <- (equals(P1,0) * log(time[i]) + (1-equals(P1,0)) * pow(time[i],P1)   )
time2[i] <- (  (1-equals(P2,P1)) * (    equals(P2,0) * log(time[i]) + (1-equals(P2,0)) *
pow(time[i],P2)  ) +
equals(P2,P1) * (    equals(P2,0) * log(time[i])*log(time[i])    +    (1-equals(P2,0)) *
pow(time[i],P2) * log(time[i]) ) )
}
for (i in 1:Nobs){
# likelihood: digitized KM curves, grouped into intervals [t, t+dt]
r[i] ~ dbin(p[i], n[i])
p[i] <- 1 - exp(-h[i] * dt[i])  # cumulative hazard over interval [t,t+dt] expressed as deaths
per person-month
# fractional polynomial
log(h[i]) <- Beta[s[i], a[i], 1] + Beta[s[i], a[i], 2] * time1[i] + Beta[s[i], a[i], 3] * time2[i]
}
## Arm level parameters = study effect + trt effect (consistency eq)
for (l in 1:Ns){
for (ll in 1:Na[l]){
Beta[l, ll, 1] <- mu[l, 1] + d[t[l, ll], 1] - d[t[l, 1], 1]
Beta[l, ll, 2] <- mu[l, 2] + d[t[l, ll], 2] - d[t[l, 1], 2]
Beta[l, ll, 3] <- mu[l, 3] + d[t[l, ll], 3] - d[t[l, 1], 3]
}
}
## Priors
for (j in 1:Ns){
mu[j, 1:3] ~ dmnorm(prior.mean[1:3], prior.prec[,])
}
d[1, 1] <- 0
d[1, 2] <- 0
d[1, 3] <- 0
for (k in 2:Ntrt){
d[k, 1:3] ~ dmnorm(prior.mean[1:3], prior.prec[,])
}
} # end of model
```

6.1.2 JAGS codes for running FP2 NMA (random effect model, d0)

```
model{
  for (i in 1:Nobs){
    time_transf1[i]<-(equals(P1,0)*log(time[i]) + (1-equals(P1,0))*pow(time[i],P1))
    time_transf2[i]<-((1-equals(P2,P1))*(equals(P2,0)*log(time[i])                              +
(1-equals(P2,0))*pow(time[i],P2))   +   equals(P2,P1)*(equals(P2,0)*log(time[i])*log(time[i])+
(1-equals(P2,0))*pow(time[i],P2) *log(time[i])))
    # likelihood
    r[i] ~ dbin(p[i], n[i])
    p[i] <- 1 - exp(-h[i] * dt[i])
    log(h[i])<-Beta[i,1]+ Beta[i,2]*time_transf1[i]+ Beta[i,3]* time_transf2[i]
    Beta[i,1]<-mu[s[i],1]+delta[s[i],1]*(1-equals (k[i], b[i]))
    Beta[i,2]<-mu[s[i],2]+(d[ks[s[i]],2]-d[bs[s[i]],2])*(1-equals (k[i], b[i]))
    Beta[i,3]<-mu[s[i],3]+(d[ks[s[i]],3]-d[bs[s[i]],3])*(1-equals (k[i], b[i]))
  }
  theta <- pow( sd , 2 )
  for(m in 1:Ns){
    delta[m,1]~dmnorm( md[ m, 1], theta )
    md[m,1]<-d[ks[m],1]-d[bs[m],1]
  }
  # priors
  d[1,1]<-0
  d[1,2]<-0
  d[1,3]<-0
  for(j in 2:Ntrt){
    d[j,1:3] ~ dmnorm(prior.mean[ 1 : 3 ], prior.prec[ , ])
  }
  for(k in 1:Ns){
    mu[k,1:3] ~ dmnorm(prior.mean[ 1 : 3 ], prior.prec[ , ])
  }
  sd ~ dunif(0,2)
}
```

6.1.3 JAGS codes for running PWE with two cutpoints

```
    model{
    for (i in 1:Nobs){
    # likelihood: digitized KM curves
    r[i] ~ dbin(p[i], n[i])
    p[i] <- 1 - exp(-h[i] * dt[i])
    # piecewise constant model
    log(h[i]) <- Beta[s[i], a[i], segment[i]]
    }
    for (i in 1:Ns){
    for (j in 1:Na[i]){
    Beta[i, j, 1] <- mu[i, 1] + d[t[i, j], 1] - d[t[i, 1], 1]
    for (k in 2:(Ncuts + 1)){
    Beta[i, j, k] <- mu[i, k] + d[t[i, j], k] - d[t[i, 1], k]-(mu[i, 1] + d[t[i, j], 1] - d[t[i,
1], 1])
    }
    }
    }
    ## Priors
    for (i in 1:Ns){
    for (k in 1:(Ncuts + 1)){
    mu[i, k] ~ dnorm(prior.mean, prior.prec)
    }
    }
    for (k in 1:(Ncuts + 1)){
    d[1, k] <- 0
    for (i in 2:Ntrt){
    d[i, k] ~ dnorm(prior.mean, prior.prec)
    }
    }
    } # end of model
```

6.1.4 JAGS codes for running fixed effect Weilbull

```
model{
  for (i in 1:Nobs){
 # likelihood
  r[i] ~ dbin(p[i], n[i])
  p[i] <- 1 - exp(-h[i] * dt[i])
  log(h[i])<-Beta[i,1]+ Beta[i,2]*log(time[i])
  Beta[i,1]<-mu[s[i],1]+md[s[i],1]*(1-equals (k[i], b[i]))
  Beta[i,2]<-mu[s[i],2]+md[s[i],2]*(1-equals (k[i], b[i]))
}
  for(k in 1 :Ns){
  md[k,1]<-d[ks[k],1]-d[bs[k],1]
  md[k,2]<-d[ks[k],2]-d[bs[k],2]
  }
  # priors
  d[1,1]<-0
  d[1,2]<-0
  for(j in 2 :Ntrt){
  d[j,1:2] ~ dmnorm(mean[1:2],prec2[,])
  }
  for(k in 1 :Ns){
  mu[k,1:2] ~ dmnorm(mean[1:2],prec2[,])
  }
  }
```

6.1.5 JAGS codes for running fixed effect Gompertz

```
model{
  for (i in 1:Nobs){
 # likelihood
  r[i] ~ dbin(p[i], n[i])
  p[i] <- 1 - exp(-h[i] * dt[i])
  log(h[i])<-Beta[i,1]+ Beta[i,2]*time[i]
  Beta[i,1]<-mu[s[i],1]+md[s[i],1]*(1-equals (k[i], b[i]))
  Beta[i,2]<-mu[s[i],2]+md[s[i],2]*(1-equals (k[i], b[i]))
}
  for(k in 1 :Ns){
  md[k,1]<-d[ks[k],1]-d[bs[k],1]
  md[k,2]<-d[ks[k],2]-d[bs[k],2]
  }
  # priors
  d[1,1]<-0
  d[1,2]<-0
  for(j in 2 :Ntrt){
  d[j,1:2] ~ dmnorm(mean[1:2],prec2[,])
  }
  for(k in 1 :Ns){
  mu[k,1:2] ~ dmnorm(mean[1:2],prec2[,])
  }
  }
```

6.1.6 JAGS codes for running fixed effect Log-normal

```
model{
  for (i in 1:Nobs){
 # likelihood
  r[i] ~ dbin(p[i], n[i])
  p[i]<-1-exp(-h[i]*dt[i])
  h[i]<-pow(2*3.1415926,-0.5)*exp(-pow((log(time[i])-nu[i])/exp(theta[i]),2)*0.5)             /
(exp(theta[i])*time[i]*phi(-(log(time[i])-nu[i])/exp(theta[i])))
  nu[i]<-mu[s[i],1]+md[s[i],1]*(1-equals (k[i], b[i]))
  theta[i]<-mu[s[i],2]+md[s[i],2]*(1-equals (k[i], b[i]))
}
  for(k in 1 :Ns){
  md[k,1]<-d[ks[k],1]-d[bs[k],1]
  md[k,2]<-d[ks[k],2]-d[bs[k],2]
  }
  # priors
  d[1,1]<-0
  d[1,2]<-0
  for(j in 2 :Ntrt){
  d[j,1:2] ~ dmnorm(mean[1:2],prec2[,])
  }
  for(k in 1 :Ns){
  mu[k,1:2] ~ dmnorm(mean[1:2],prec2[,])
  }
  }
```

6.1.7 JAGS codes for running fixed effect Log-logistic

```
model{
  for (i in 1:Nobs){
 # likelihood
  r[i] ~ dbin(p[i], n[i])
  p[i]<-1-exp(-h[i]*dt[i])
  h[i]<-(exp(theta[i])/exp(nu[i]))*pow(time[i]/exp(nu[i]),
exp(theta[i])-1)/(1+pow(time[i]/exp(nu[i]), exp(theta[i])))
  nu[i]<-mu[s[i],1]+md[s[i],1]*(1-equals (k[i], b[i]))
  theta[i]<-mu[s[i],2]+md[s[i],2]*(1-equals (k[i], b[i]))
}
  for(k in 1 :Ns){
  md[k,1]<-d[ks[k],1]-d[bs[k],1]
  md[k,2]<-d[ks[k],2]-d[bs[k],2]
  }
  # priors
  d[1,1]<-0
  d[1,2]<-0
  for(j in 2 :Ntrt){
  d[j,1:2] ~ dmnorm(mean[1:2],prec2[,])
  }
  for(k in 1 :Ns){
  mu[k,1:2] ~ dmnorm(mean[1:2],prec2[,])
  }
  }
```

6.1.8 JAGS codes for running Fixed effect COX PH NMA

```
model{
for(i in 1:ns2) {
    y[i,2] ~ dnorm(delta[i,2],prec[i,2])
    resdev[i] <- (y[i,2]-delta[i,2])*(y[i,2]-delta[i,2])*prec[i,2]
  }
for(i in (ns2+1):(ns2+ns3)) {
    for (k in 1:(na[i]-1)) {
        for (j in 1:(na[i]-1)) {
            Sigma[i,j,k] <- V[i]*(1-equals(j,k)) + var0[i,k+1]*equals(j,k)
          }
      }
    Omega[i,1:(na[i]-1),1:(na[i]-1)] <- inverse(Sigma[i,,])
    y[i,2:na[i]] ~ dmnorm(delta[i,2:na[i]],Omega[i,1:(na[i]-1),1:(na[i]-1)])
      for (k in 1:(na[i]-1)){
        ydiff[i,k]<- y[i,(k+1)] - delta[i,(k+1)]
        z[i,k]<- inprod(Omega[i,k,1:(na[i]-1)], ydiff[i,1:(na[i]-1)])
      }
    resdev[i]<- inprod(ydiff[i,1:(na[i]-1)], z[i,1:(na[i]-1)])
  }
for(i in 1:(ns2+ns3)){
     for (k in 2:na[i]) {
        var0[i,k] <- pow(se[i,k],2)
        prec[i,k] <- 1/var0[i,k]
        delta[i,k] <-  d[t[i,k]] - d[t[i,1]]
      }
  }
totresdev <- sum(resdev[])
d[1]<-0
for (k in 2:nt){
     d[k] ~ dnorm(0,0.0001)
}
for (k in 2:nt){
     hrd[k] <- exp(d[k])
}
}
```

6.1.9 JAGS codes for running fixed effect Generalised Gamma Model NMA

```
model {
  #Define Prior Distributions
  beta[1] <- 0
  for (tt in 2:nTx){
    beta[tt]~dnorm(0,1.0E-4)
  }
  for(ss in 1:nStudies){
    alpha[ss] ~ dnorm(0,1.0E-4)
  }
  for(ii in 1:LnObs ){
    Lmu[ii] <- alpha[Lstudy[ii]]*multi[ii] + beta[Ltx[ii]] - beta[Lbase[ii]]
    Lprec[ii] <- 1/pow(Lse[ii],2)
    Lmean[ii] ~ dnorm(Lmu[ii],Lprec[ii])
  }
  # Calculate AFT
  for (hh in 1:nTx) {
   aft[hh] <- exp(beta[hh])
  }
}
```