Supplementary program description

QCIC model operation document

The development of the QCIC model is primarily based on a system of ordinary differential equations (ODEs) and the fourth-order Runge-Kutta algorithm for numerical solving. When accessing and running the relevant programs, users need to focus on the following two key aspects:

1. **Generation of Virtual Patient Cohorts.** Users can execute parallel computations by running the "QCIC model basic operation code" to generate virtual patient cohorts. During this process, users can adjust model parameters according to research needs to simulate different treatment strategies or clinical scenarios. The generation of virtual patients is the core step of the entire model operation, and the output data will serve as the foundation for subsequent analysis.

2. **Data Usage and Processing.** We have detailed the fundamental principles of data processing and analysis in the methods section of the manuscript. Users can choose their preferred programming language (such as Python, R, or Matlab) for further development. To facilitate understanding, we provide scripts written in Matlab to demonstrate the data processing workflow and usage methods. However, users need to carefully review the relevant documentation and fully comprehend the program logic to ensure correct usage and extension of the model's functionalities.

By following the steps above, users can effectively use the QCIC model for virtual patient generation and data analysis. However, due to GitHub's limitations on repository and file size, we cannot upload extensive procedural data, which may inconvenience users trying to run the program and generate results directly. If you're interested in this work, please contact the corresponding author to obtain the relevant procedural data, enabling you to better reproduce the research findings and explore the model in greater depth.

QCIC model basic operation code

Step 1: Create a Data Storage Folder.

Create a folder named **"VP"** to store the generated data. This folder will serve as a data hub, ensuring that subsequent result analysis can seamlessly access the required data files.

Step 2: Run the "VirPat()" Function to Generate Virtual Patients.

The **"VirPat()"** function is a highly integrated function document that includes all the subfunctions required for numerical calculations. The primary function of this code is to generate virtual patient data, and its core internal parameters are as follows:

- **n**: Represents the number of virtual patients to be generated. The value of **n** depends on the sampling frequency of the tumor heterogeneity parameters. （During the debugging process, it is advised that users set "n" to 5, which entails performing 5 parallel computations. If utilizing a server with superior performance for calculations, "n" may be set to 100 to enhance computational speed.） By adjusting the “**sample**” parameter in the **"LHS_Matrix()"** function, users can generate virtual patient datasets of varying scales. （During the debugging process, it is recommended that users set "sample" to 100, which means generating 100 virtual patients.）
- **number**: Represents the number of CPU cores used for parallel computing. By specifying the number of cores, users can fully utilize the computational power of multi-core CPUs, significantly improving computational efficiency. This parameter needs to be set based on the performance of the user's local server.

Step 3: Generate a Set of Tumor Patient Heterogeneity Parameters.

(1) Use **"S=LHS_Matrix()"** to perform random sampling of tumor

heterogeneity parameters.

- **sample**: Represents the number of parameter samplings (i.e., the expected number of virtual patients to be generated). This parameter determines how many sets of parameter combinations are generated.

- **para**: Represents the total number of tumor heterogeneity parameters. Each parameter corresponds to a specific characteristic of tumor heterogeneity.

- The **"Beta_Matrix()"** function is used to perform random parameter sampling following the Beta distribution. The Beta distribution is a flexible distribution particularly well-suited for describing the diversity of tumor heterogeneity parameters.

(2) Structure of the Parameter Set. The returned matrix **"S"** contains the values of all parameters. Each row of the matrix describes the physiological characteristics of a virtual patient, and the total number of rows equals the total number of generated virtual patients. This structured data organization form enables more efficient subsequent analysis and processing.

(3) Saving the Parameter Set. Save the generated parameter matrix **"S"** in the **"VP"** folder and name it **"VP_LHS_Matrix.mat"**. This file format facilitates quick loading and use in subsequent steps.

Step 4: Perform Parallel Computation for the Parametric QCIC Model.

(1) Create a parallel computing pool with parpool (number). The number represents the quantity of parallel computing pools to be created. By specifying the number of cores, users can fully utilize the computing resources of multicore CPUs or clusters, significantly improving computational efficiency.

(2) Call the **"main()"** function. The **"main()"** function is an internal function required for parallel computation, primarily used to invoke tumor heterogeneity parameters to execute the QCIC model and the Runge-Kutta algorithm. This function significantly enhances computational efficiency by parallelizing

computing tasks.

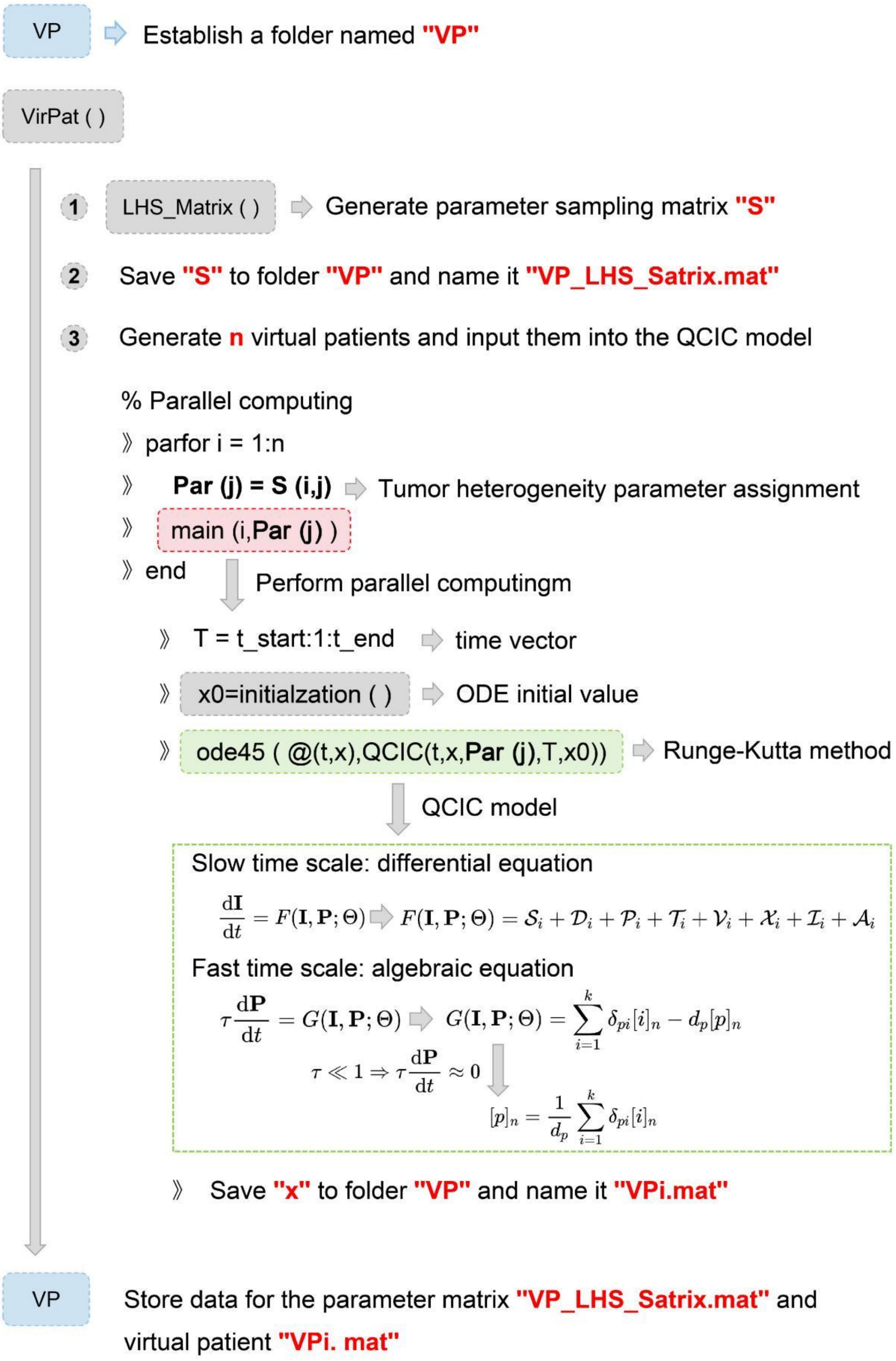
(3) Other Auxiliary Functions.

- The **"initialization()"** function is used to call the initial values for the ordinary differential equations (ODEs).
- The **"ode45"** function represents the standard Runge-Kutta algorithm, which is utilized for numerically solving ODEs. This algorithm boasts high precision and stability, making it suitable for handling complex biomedical models.
- The **"QCIC()"** function primarily serves to store the computational formulas of the model. It defines the core equations of the tumor-immune dynamics model.
- The **"parameter()"** function is used to store the parameters of the QCIC model. Various dosing regimens can be achieved by adjusting the parameters "par.k_in_Targ" and "par.k_in_Chem".

Step 5: Perform parallel computation for the parametric QCIC model.

After performing numerical computations for the ith iteration, export the data and name it **"VPi"**, saving it in the **"VP"** folder. This naming convention facilitates subsequent data management and analysis.

By following the above steps, users can generate virtual patient data, execute numerical computations for the QCIC model, and ultimately save the results in the designated folder. This process not only enhances computational efficiency but also ensures data traceability and reproducibility. We hope this detailed procedural explanation will help users better understand and utilize the code. Should you have any questions or suggestions, please feel free to contact us at any time.



VP ⇨ Establish a folder named **"VP"**

VirPat ( )

1 LHS_Matrix ( ) ⇨ Generate parameter sampling matrix **"S"**

2 Save **"S"** to folder **"VP"** and name it **"VP_LHS_Satrix.mat"**

3 Generate **n** virtual patients and input them into the QCIC model

```
% Parallel computing
》 parfor i = 1:n
》     Par (j) = S (i,j)  ⇨ Tumor heterogeneity parameter assignment
》   main (i,Par (j) )
》 end
```

Perform parallel computingm

```
》 T = t_start:1:t_end  ⇨ time vector
》 x0=initialzation ( )  ⇨ ODE initial value
》 ode45 ( @(t,x),QCIC(t,x,Par (j),T,x0))  ⇨ Runge-Kutta method
```

QCIC model

Slow time scale: differential equation

$$\frac{\mathrm{d}\mathbf{I}}{\mathrm{d}t} = F(\mathbf{I},\mathbf{P};\Theta) \Rightarrow F(\mathbf{I},\mathbf{P};\Theta) = \mathcal{S}_i + \mathcal{D}_i + \mathcal{P}_i + \mathcal{T}_i + \mathcal{V}_i + \mathcal{X}_i + \mathcal{I}_i + \mathcal{A}_i$$

Fast time scale: algebraic equation

$$\tau\frac{\mathrm{d}\mathbf{P}}{\mathrm{d}t} = G(\mathbf{I},\mathbf{P};\Theta) \Rightarrow G(\mathbf{I},\mathbf{P};\Theta) = \sum_{i=1}^{k}\delta_{pi}[i]_n - d_p[p]_n$$

$$\tau \ll 1 \Rightarrow \tau\frac{\mathrm{d}\mathbf{P}}{\mathrm{d}t} \approx 0$$

$$[p]_n = \frac{1}{d_p}\sum_{i=1}^{k}\delta_{pi}[i]_n$$

》 Save **"x"** to folder **"VP"** and name it **"VPi.mat"**

VP Store data for the parameter matrix **"VP_LHS_Satrix.mat"** and virtual patient **"VPi. mat"**

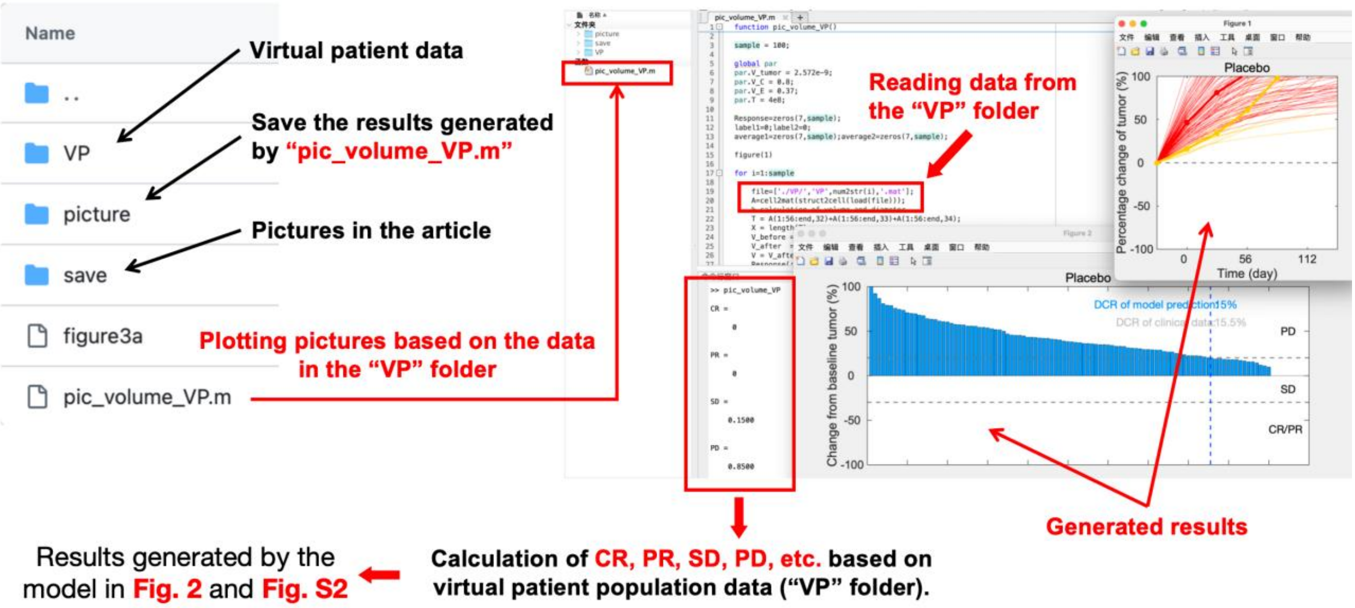
(Figure 1. Basic code calculation flowchart)

Prediction of short-term treatment efficacy

(Fig. 2, Fig. 3, and Fig. S2)

Step 1: Incorporate the Generated Virtual Patient Data.

Users can directly copy the virtual patient data (**"VP"** folder) generated through the **"QCIC model basic operation code"** into the corresponding "Fig.3a.3b/3c" folder.



Name
..
VP
picture
save
figure3a
pic_volume_VP.m
名称
文件夹
picture
save
VP
pic_volume_VP.m
pic_volume_VP.m
Figure 1
文件 编辑 查看 插入 工具 桌面 窗口 帮助
Placebo
Percentage change of tumor (%)
100
50
0
-50
-100
0
56
112
Time (day)
Figure 2
文件 编辑 查看 插入 工具 桌面 窗口 帮助
Placebo
Change from baseline tumor (%)
100
50
0
-50
-100
DCR of model prediction15%
DCR of clinical data:15.5%
PD
SD
CR/PR

Virtual patient data

Save the results generated by "pic_volume_VP.m"

Pictures in the article

Plotting pictures based on the data in the "VP" folder

Reading data from the "VP" folder

```
function pic_volume_VP()

sample = 100;

global par
par.V_tumor = 2.572e-9;
par.V_C = 0.8;
par.V_E = 0.37;
par.T = 4e8;

Response=zeros(7,sample);
label1=0;label2=0;
average1=zeros(7,sample);average2=zeros(7,sample);

figure(1)

for i=1:sample

    file=['./VP/','VP',num2str(i),'.mat'];
    A=cell2mat(struct2cell(load(file)));

    T = A(1:56:end,32)+A(1:56:end,33)+A(1:56:end,34);
    X = length
    V_before =
    V_after  =
    V = V_afte
```

```
>> pic_volume_VP

CR =

     0


PR =

     0


SD =

    0.1500


PD =

    0.8500
```

Generated results

Results generated by the model in **Fig. 2** and **Fig. S2**

Calculation of CR, PR, SD, PD, etc. based on virtual patient population data ("VP" folder).

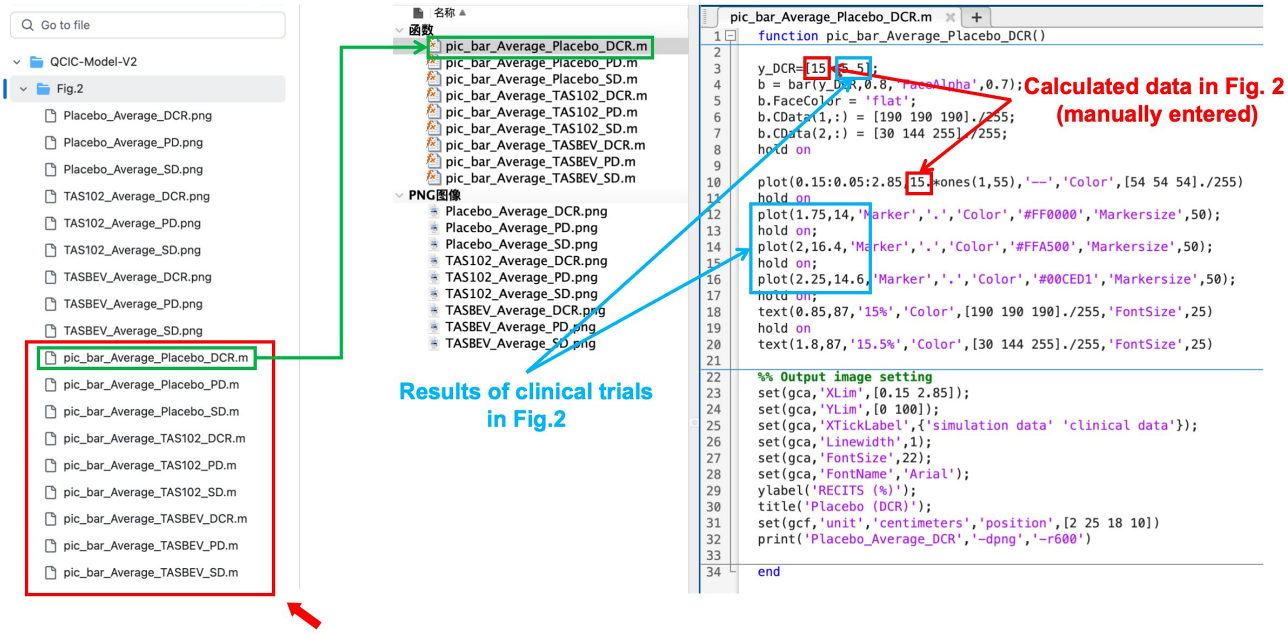
(Figure 2. Dynamic changes in disease progression in virtual patients)

Step 2: Run the "pic_volume_VP.m" File.

Run the **"pic volume VP.m"** file to calculate the TRI indicators mentioned in the manuscript and generate the corresponding "Fig.3a.3b/3c" and short-term efficacy data (CR/PR/SD/PD). Then, manually calculate the ORR and DCR using the RECIST criteria introduced in the manuscript (ORR=CR+PR, DCR=CR+PR+SD). Finally, obtain the corresponding values for CR, PR, SD, PD, ORR, and DCR.

Step 3: Create Fig.2 and Fig.S2.

Manually enter the obtained values for CR, PR, SD, PD, ORR, and DCR into the corresponding positions in Fig.2 and Fig.S2. If you plan to extend this model example to other cancer research studies, you will need to modify the corresponding clinical data.


Go to file
QCIC-Model-V2
Fig.2
Placebo_Average_DCR.png
Placebo_Average_PD.png
Placebo_Average_SD.png
TAS102_Average_DCR.png
TAS102_Average_PD.png
TAS102_Average_SD.png
TASBEV_Average_DCR.png
TASBEV_Average_PD.png
TASBEV_Average_SD.png
pic_bar_Average_Placebo_DCR.m
pic_bar_Average_Placebo_PD.m
pic_bar_Average_Placebo_SD.m
pic_bar_Average_TAS102_DCR.m
pic_bar_Average_TAS102_PD.m
pic_bar_Average_TAS102_SD.m
pic_bar_Average_TASBEV_DCR.m
pic_bar_Average_TASBEV_PD.m
pic_bar_Average_TASBEV_SD.m
名称
函数
pic_bar_Average_Placebo_DCR.m
pic_bar_Average_Placebo_PD.m
pic_bar_Average_Placebo_SD.m
pic_bar_Average_TAS102_DCR.m
pic_bar_Average_TAS102_PD.m
pic_bar_Average_TAS102_SD.m
pic_bar_Average_TASBEV_DCR.m
pic_bar_Average_TASBEV_PD.m
pic_bar_Average_TASBEV_SD.m
PNG图像
Placebo_Average_DCR.png
Placebo_Average_PD.png
Placebo_Average_SD.png
TAS102_Average_DCR.png
TAS102_Average_PD.png
TAS102_Average_SD.png
TASBEV_Average_DCR.png
TASBEV_Average_PD.png
TASBEV_Average_SD.png
pic_bar_Average_Placebo_DCR.m

```
function pic_bar_Average_Placebo_DCR()

y_DCR=[15 15.5];
b = bar(y_DCR,0.8,'FaceAlpha',0.7);
b.FaceColor = 'flat';
b.CData(1,:) = [190 190 190]./255;
b.CData(2,:) = [30 144 255]./255;
hold on

plot(0.15:0.05:2.85,15.*ones(1,55),'--','Color',[54 54 54]./255)
hold on
plot(1.75,14,'Marker','.','Color','#FF0000','Markersize',50);
hold on;
plot(2,16.4,'Marker','.','Color','#FFA500','Markersize',50);
hold on;
plot(2.25,14.6,'Marker','.','Color','#00CED1','Markersize',50);
hold on;
text(0.85,87,'15%','Color',[190 190 190]./255,'FontSize',25)
hold on
text(1.8,87,'15.5%','Color',[30 144 255]./255,'FontSize',25)

%% Output image setting
set(gca,'XLim',[0.15 2.85]);
set(gca,'YLim',[0 100]);
set(gca,'XTickLabel',{'simulation data' 'clinical data'});
set(gca,'Linewidth',1);
set(gca,'FontSize',22);
set(gca,'FontName','Arial');
ylabel('RECITS (%)');
title('Placebo (DCR)');
set(gcf,'unit','centimeters','position',[2 25 18 10])
print('Placebo_Average_DCR','-dpng','-r600')

end
```

Calculated data in Fig. 2 (manually entered)

Results of clinical trials in Fig.2

The processed model data (for details, refer to the methodology section of the manuscript and the computational results in Fig. 3) along with the experimental data.

(Figure 3. Visualization presentation of key indicators)

Explanation 1: The "Fig.2" and "Fig.S2" folders are used to plot the comparison of short-term efficacy evaluation indicators at the population level between model simulations and clinical data. The extracted clinical data are nested into the corresponding plotting program. The results of the model simulations are also placed in the plotting program in the form of data points.

Explanation 2: The "Fig.3" folder is used to illustrate the evolution of the disease dynamics of 100 baseline patients. The "Fig.3a", "Fig.3b", and "Fig.3c" folders are dedicated to presenting the computational results for the control group, the TAS-102 chemotherapy group, and the combination therapy group of TAS-102 with Bevacizumab, respectively. The "VP" folder contains data for these 100 baseline patients.

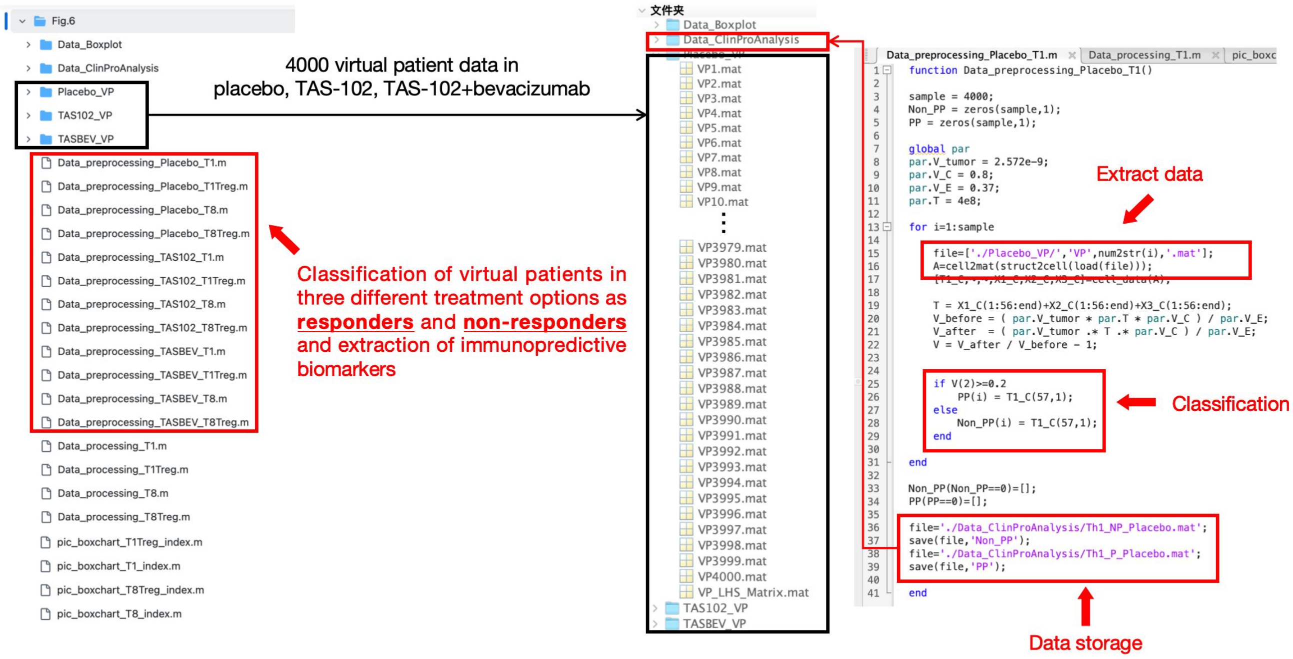
Explanation 3: Due to the data storage limitations on the GitHub platform, we are unable to upload the extensive procedural datasets. If you are interested in this section or wish to conduct further analyses, please feel free to contact the corresponding author to obtain the relevant data.

Predictive biomarkers for mCRC based on the QCIC model

(Fig.6)

Step 1: Establish Folders for Storing Virtual Patient Data.

Create three folders named **"Placebo_VP"**, **"TAS02_VP"**, and **"TASBEV_VP"** to store the virtual patient data generated under different treatment regimens. Due to the large size of the case data, it is inconvenient to upload to GitHub. You can contact the author to obtain the data.



4000 virtual patient data in placebo, TAS-102, TAS-102+bevacizumab

Classification of virtual patients in three different treatment options as **responders** and **non-responders** and extraction of immunopredictive biomarkers

```
function Data_preprocessing_Placebo_T1()

sample = 4000;
Non_PP = zeros(sample,1);
PP = zeros(sample,1);

global par
par.V_tumor = 2.572e-9;
par.V_C = 0.8;
par.V_E = 0.37;
par.T = 4e8;

for i=1:sample

    file=['./Placebo_VP/','VP',num2str(i),'.mat'];
    A=cell2mat(struct2cell(load(file)));
    [T1_C,~,~,X1_C,X2_C,X3_C]=cell_data(A);

    T = X1_C(1:56:end)+X2_C(1:56:end)+X3_C(1:56:end);
    V_before = ( par.V_tumor * par.T * par.V_C ) / par.V_E;
    V_after  = ( par.V_tumor .* T .* par.V_C ) / par.V_E;
    V = V_after / V_before - 1;


    if V(2)>=0.2
        PP(i) = T1_C(57,1);
    else
        Non_PP(i) = T1_C(57,1);
    end

end

Non_PP(Non_PP==0)=[];
PP(PP==0)=[];

file='./Data_ClinProAnalysis/Th1_NP_Placebo.mat';
save(file,'Non_PP');
file='./Data_ClinProAnalysis/Th1_P_Placebo.mat';
save(file,'PP');

end
```

Extract data

Classification

Data storage

(Figure 4. Calculation program for data extraction and processing process)

Step 2: Establish Folders for Storing Processed Intermediate Data.

Create two folders named **"Data_Boxplot"** and **"Data_ClinProAnalysis"** to store the processed intermediate data.

Step 3: Run Data Processing Programs to Obtain Categorized Data.

Run the data processing programs to obtain categorized data. For example, run the **"Data_preprocessing_Placebo_T1()"** function to call the virtual patient data from the **"Placebo_VP"** folder. Users can classify patients into responders (R) and non-responders (NR) based on tumor size and save them separately in the **"Data_ClinProAnalysis"** folder with the names **"Th1_NP_Placebo.mat"** and **"Th1_P_Placebo.mat"**.

Step 4: Extract 500 Data Points for Analysis from Categorized Data.

For example, run the **"Data_processing_T1()"** function to call the CD4+ Th1 cell data from **"Data_ClinProAnalysis"**. Then, extract data for 500 CD4+ Th1 cells from both responders (R) and non-responders (NR). Subsequently, save the results in the "Data_Boxplot" folder with the name **"T1.mat"**.

Step 5: Plot Distribution Charts.

For example, run the **"pic_boxchart_T1_index()"** function to call the **"T1.mat"** data extracted from the **"Data_Boxplot"** folder. Finally, generate the image Fig.6.

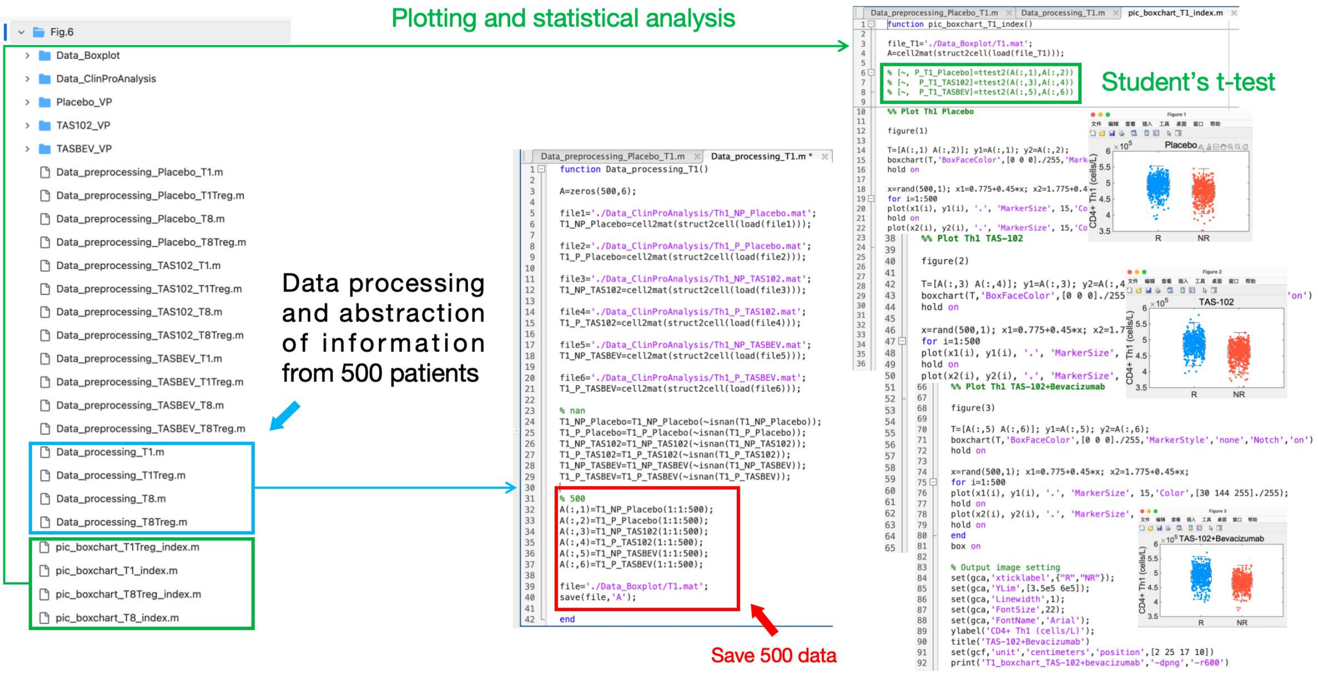


Fig.6
Data_Boxplot
Data_ClinProAnalysis
Placebo_VP
TAS102_VP
TASBEV_VP
Data_preprocessing_Placebo_T1.m
Data_preprocessing_Placebo_T1Treg.m
Data_preprocessing_Placebo_T8.m
Data_preprocessing_Placebo_T8Treg.m
Data_preprocessing_TAS102_T1.m
Data_preprocessing_TAS102_T1Treg.m
Data_preprocessing_TAS102_T8.m
Data_preprocessing_TAS102_T8Treg.m
Data_preprocessing_TASBEV_T1.m
Data_preprocessing_TASBEV_T1Treg.m
Data_preprocessing_TASBEV_T8.m
Data_preprocessing_TASBEV_T8Treg.m
Data_processing_T1.m
Data_processing_T1Treg.m
Data_processing_T8.m
Data_processing_T8Treg.m
pic_boxchart_T1Treg_index.m
pic_boxchart_T1_index.m
pic_boxchart_T8Treg_index.m
pic_boxchart_T8_index.m
Plotting and statistical analysis
Data processing and abstraction of information from 500 patients
Data_preprocessing_Placebo_T1.m
Data_processing_T1.m *
Save 500 data
Data_processing_Placebo_T1.m
Data_processing_T1.m
pic_boxchart_T1_index.m
Student's t-test
Figure 1
Placebo
Figure 2
TAS-102
Figure 3
TAS-102+Bevacizumab
文件 编辑 查看 插入 工具 桌面 窗口 帮助
$\times 10^5$
CD4+ Th1 (cells/L)
6
5.5
5
4.5
4
3.5
R
NR

```
function Data_processing_T1()

A=zeros(500,6);

file1='./Data_ClinProAnalysis/Th1_NP_Placebo.mat';
T1_NP_Placebo=cell2mat(struct2cell(load(file1)));

file2='./Data_ClinProAnalysis/Th1_P_Placebo.mat';
T1_P_Placebo=cell2mat(struct2cell(load(file2)));

file3='./Data_ClinProAnalysis/Th1_NP_TAS102.mat';
T1_NP_TAS102=cell2mat(struct2cell(load(file3)));

file4='./Data_ClinProAnalysis/Th1_P_TAS102.mat';
T1_P_TAS102=cell2mat(struct2cell(load(file4)));

file5='./Data_ClinProAnalysis/Th1_NP_TASBEV.mat';
T1_NP_TASBEV=cell2mat(struct2cell(load(file5)));

file6='./Data_ClinProAnalysis/Th1_P_TASBEV.mat';
T1_P_TASBEV=cell2mat(struct2cell(load(file6)));

% nan
T1_NP_Placebo=T1_NP_Placebo(~isnan(T1_NP_Placebo));
T1_P_Placebo=T1_P_Placebo(~isnan(T1_P_Placebo));
T1_NP_TAS102=T1_NP_TAS102(~isnan(T1_NP_TAS102));
T1_P_TAS102=T1_P_TAS102(~isnan(T1_P_TAS102));
T1_NP_TASBEV=T1_NP_TASBEV(~isnan(T1_NP_TASBEV));
T1_P_TASBEV=T1_P_TASBEV(~isnan(T1_P_TASBEV));

% 500
A(:,1)=T1_NP_Placebo(1:1:500);
A(:,2)=T1_P_Placebo(1:1:500);
A(:,3)=T1_NP_TAS102(1:1:500);
A(:,4)=T1_P_TAS102(1:1:500);
A(:,5)=T1_NP_TASBEV(1:1:500);
A(:,6)=T1_P_TASBEV(1:1:500);

file='./Data_Boxplot/T1.mat';
save(file,'A');

end
```

```
function pic_boxchart_T1_index()

file_T1='./Data_Boxplot/T1.mat';
A=cell2mat(struct2cell(load(file_T1)));

% [~, P_T1_Placebo]=ttest2(A(:,1),A(:,2))
% [~,  P_T1_TAS102]=ttest2(A(:,3),A(:,4))
% [~,  P_T1_TASBEV]=ttest2(A(:,5),A(:,6))

%% Plot Th1 Placebo

figure(1)

T=[A(:,1) A(:,2)]; y1=A(:,1); y2=A(:,2);
boxchart(T,'BoxFaceColor',[0 0 0]./255,'Mark
hold on

x=rand(500,1); x1=0.775+0.45*x; x2=1.775+0.4
for i=1:500
plot(x1(i), y1(i), '.', 'MarkerSize', 15,'Co
hold on
plot(x2(i), y2(i), '.', 'MarkerSize', 15,'Co
```

```
%% Plot Th1 TAS-102

figure(2)

T=[A(:,3) A(:,4)]; y1=A(:,3); y2=A(:,4
boxchart(T,'BoxFaceColor',[0 0 0]./255                              'on')
hold on

x=rand(500,1); x1=0.775+0.45*x; x2=1.7
for i=1:500
plot(x1(i), y1(i), '.', 'MarkerSize',
hold on
plot(x2(i), y2(i), '.', 'MarkerSize',
```

```
%% Plot Th1 TAS-102+Bevacizumab

figure(3)

T=[A(:,5) A(:,6)]; y1=A(:,5); y2=A(:,6);
boxchart(T,'BoxFaceColor',[0 0 0]./255,'MarkerStyle','none','Notch','on')
hold on

x=rand(500,1); x1=0.775+0.45*x; x2=1.775+0.45*x;
for i=1:500
plot(x1(i), y1(i), '.', 'MarkerSize', 15,'Color',[30 144 255]./255);
hold on
plot(x2(i), y2(i), '.', 'MarkerSize',
hold on
end
box on

% Output image setting
set(gca,'xticklabel',{"R","NR"});
set(gca,'YLim',[3.5e5 6e5]);
set(gca,'Linewidth',1);
set(gca,'FontSize',22);
set(gca,'FontName','Arial');
ylabel('CD4+ Th1 (cells/L)');
title('TAS-102+Bevacizumab')
set(gcf,'unit','centimeters','position',[2 25 17 10])
print('T1_boxchart_TAS-102+bevacizumab','-dpng','-r600')
```

(Figure 5. Plot the distribution of predictive biomarkers)

Explanation: The "Fig.6" folder is used to plot the distribution of different predictive biomarkers between responders and non-responders. The folders "Placebo_VP", "TAS102_VP", and "TASBEV_VP" are utilized to store data for 4,000 randomly generated patients in the control group, the TAS-102 chemotherapy group, and the TAS-102 combined with Bevacizumab group, respectively. **As mentioned before, this part of the data can be obtained from the corresponding author.** Other process programs and calculation results are only saved in their corresponding locations within the "Fig.6" folder.

Predictive biomarkers for mCRC based on the QCIC model

(Fig.7)

Step 1: Establish Folders for Data Storage.

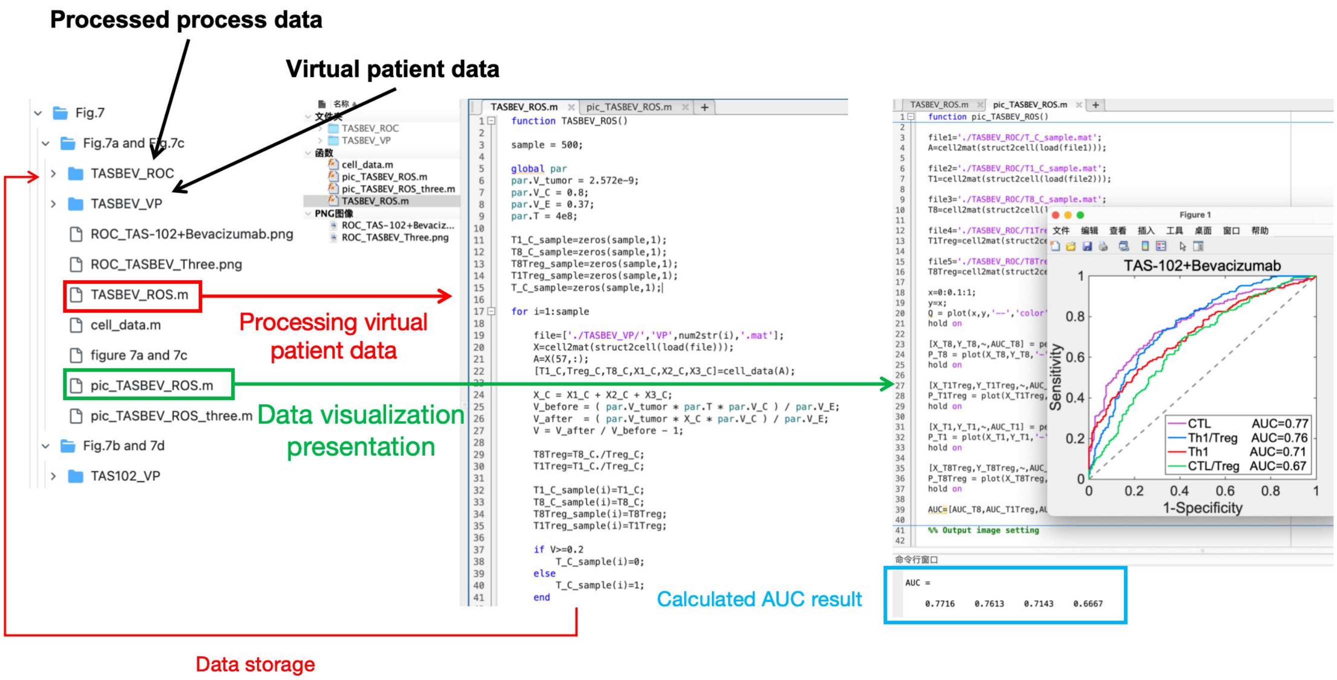
Take drawing Fig.7a and Fig.7c as examples. Users need to create folders named **"TASBEV_VP"** and **"TASBEV_ROC"** to import patient datasets for combination therapy.

Step 2: Data Processing.

For example, users can run the **"TASBEV_ROS()"** function to call and process the data under the **"TASBEV_VP"** folder. The processed data should be saved in the **"TASBEV_ROC"** folder.

Step 3: Plot ROC Curves.

Users can call the **"pic_TASBEV_ROS()"** and **"pic_TASBEV_ROS_three()"** functions to generate visual images.



```
function TASBEV_ROS()

sample = 500;

global par
par.V_tumor = 2.572e-9;
par.V_C = 0.8;
par.V_E = 0.37;
par.T = 4e8;

T1_C_sample=zeros(sample,1);
T8_C_sample=zeros(sample,1);
T8Treg_sample=zeros(sample,1);
T1Treg_sample=zeros(sample,1);
T_C_sample=zeros(sample,1);

for i=1:sample

    file=['./TASBEV_VP/','VP',num2str(i),'.mat'];
    X=cell2mat(struct2cell(load(file)));
    A=X(57,:);
    [T1_C,Treg_C,T8_C,X1_C,X2_C,X3_C]=cell_data(A);

    X_C = X1_C + X2_C + X3_C;
    V_before = ( par.V_tumor * par.T * par.V_C ) / par.V_E;
    V_after  = ( par.V_tumor * X_C * par.V_C ) / par.V_E;
    V = V_after / V_before - 1;

    T8Treg=T8_C./Treg_C;
    T1Treg=T1_C./Treg_C;

    T1_C_sample(i)=T1_C;
    T8_C_sample(i)=T8_C;
    T8Treg_sample(i)=T8Treg;
    T1Treg_sample(i)=T1Treg;

    if V>=0.2
        T_C_sample(i)=0;
    else
        T_C_sample(i)=1;
    end
```

```
function pic_TASBEV_ROS()

file1='./TASBEV_ROC/T_C_sample.mat';
A=cell2mat(struct2cell(load(file1)));

file2='./TASBEV_ROC/T1_C_sample.mat';
T1=cell2mat(struct2cell(load(file2)));

file3='./TASBEV_ROC/T8_C_sample.mat';
T8=cell2mat(struct2cell(l

file4='./TASBEV_ROC/T1Tre
T1Treg=cell2mat(struct2ce

file5='./TASBEV_ROC/T8Tre
T8Treg=cell2mat(struct2ce

x=0:0.1:1;
y=x;
Q = plot(x,y,'--','color'
hold on

[X_T8,Y_T8,~,AUC_T8] = pe
P_T8 = plot(X_T8,Y_T8,'-'
hold on

[X_T1Treg,Y_T1Treg,~,AUC_
P_T1Treg = plot(X_T1Treg,
hold on

[X_T1,Y_T1,~,AUC_T1] = pe
P_T1 = plot(X_T1,Y_T1,'-'
hold on

[X_T8Treg,Y_T8Treg,~,AUC_
P_T8Treg = plot(X_T8Treg,
hold on

AUC=[AUC_T8,AUC_T1Treg,AL

%% Output image setting
```

```
AUC =

    0.7716    0.7613    0.7143    0.6667
```

(Figure 6. Evaluate the ROC and AUC of predictive biomarkers.)

Explanation: The "Fig.7" folder is dedicated to plotting the ROC analysis results for different biomarkers. **The folders "TAS102_VP" and "TASBEV_VP" are used to store 500 valid data points extracted from the 4,000 randomly generated patients in the TAS-102 chemotherapy group and the TAS-102 combined with Bevacizumab group, respectively. Due to the large volume of data, this part of the data can be obtained from the corresponding author.**

Prediction of the overall survival

(Fig.8, Fig.4, and Fig.5)

Step 1: Conduct Basic Analysis on Generated Virtual Patient Data.

Take Fig.8 as an example. To further categorize the expression levels of predictive biomarkers for survival analysis, users need to divide the generated large-scale data (matrices) into two categories based on the expression levels of predictive biomarkers and store them in the **"ProAnalysis"** folder.

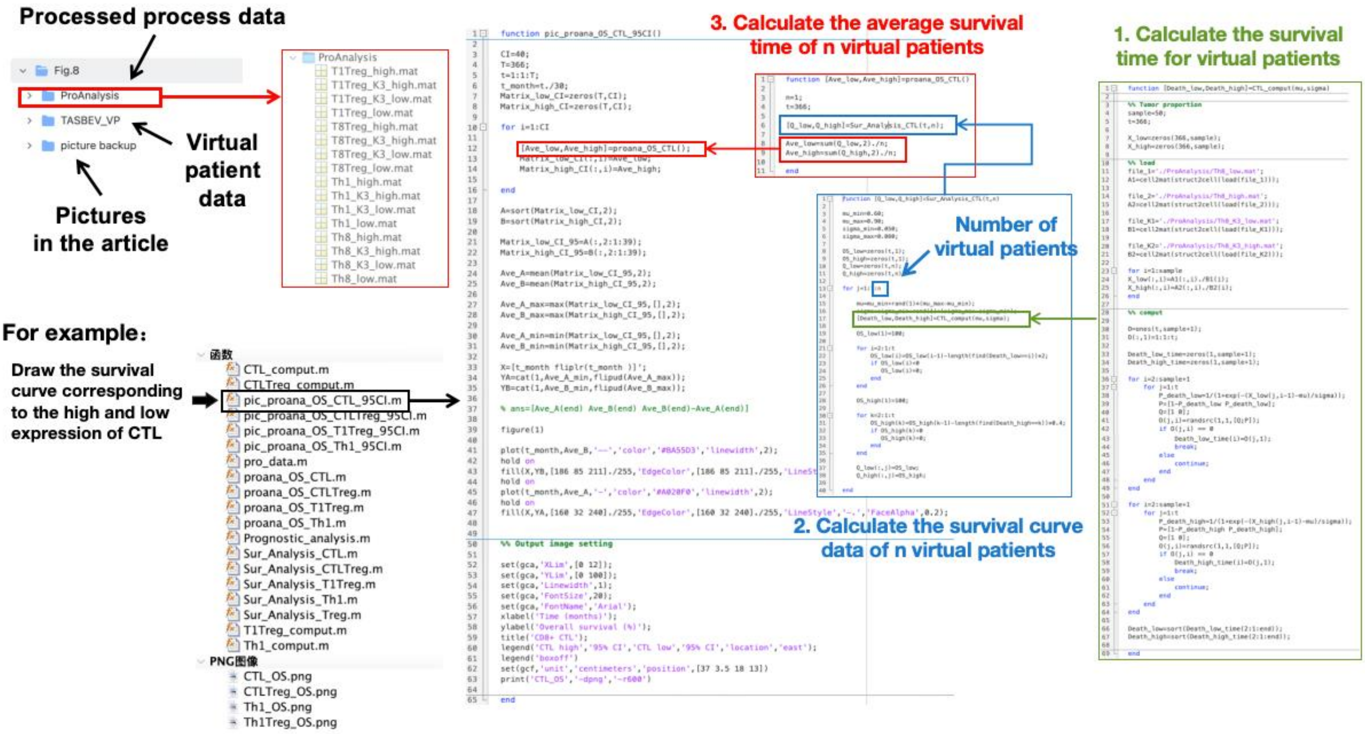
Step 2: Calculate Survival Probabilities Using the DPF Formula in the Manuscript.

Users can design their own functions based on the formula provided in the manuscript. Alternatively, users can utilize the code we have written, but they need to pay attention to the calling format and data preprocessing.

If there are difficulties in practical operation, users can contact us to obtain materials with case data. However, due to the large volume of data, we are unable to upload it one by one to GitHub.

Step 3: Perform Stochastic Calculations Based on Survival Probabilities to Obtain Survival Curves.

We have integrated all the operational details into the function for drawing images. For example, the plotting function **"pic_proana_OS_CTL_95CI()"** displays the survival curves for high and low expression of the CTL indicator. In the plotting process, functions such as **"proana_OS_CTL()"**, **"Sur_Analysis_CTL(t,n)"**, and **"CTL_comput(mu,sigma)"** need to be called. The **"CTL_comput(mu,sigma)"** function further requires the data stored in the "ProAnalysis" folder from Step One.



```
function pic_proana_OS_CTL_95CI()

CI=40;
T=366;
t=1:1:T;
t_month=t./30;
Matrix_low_CI=zeros(T,CI);
Matrix_high_CI=zeros(T,CI);

for i=1:CI

    [Ave_low,Ave_high]=proana_OS_CTL();
    Matrix_low_CI(:,i)=Ave_low;
    Matrix_high_CI(:,i)=Ave_high;

end

A=sort(Matrix_low_CI,2);
B=sort(Matrix_high_CI,2);

Matrix_low_CI_95=A(:,2:1:39);
Matrix_high_CI_95=B(:,2:1:39);

Ave_A=mean(Matrix_low_CI_95,2);
Ave_B=mean(Matrix_high_CI_95,2);

Ave_A_max=max(Matrix_low_CI_95,[],2);
Ave_B_max=max(Matrix_high_CI_95,[],2);

Ave_A_min=min(Matrix_low_CI_95,[],2);
Ave_B_min=min(Matrix_high_CI_95,[],2);

X=[t_month fliplr(t_month )]';
YA=cat(1,Ave_A_min,flipud(Ave_A_max));
YB=cat(1,Ave_B_min,flipud(Ave_B_max));

% ans=[Ave_A(end) Ave_B(end) Ave_B(end)-Ave_A(end)]

figure(1)

plot(t_month,Ave_B,'--','color','#BA55D3','linewidth',2);
hold on
fill(X,YB,[186 85 211]./255,'EdgeColor',[186 85 211]./255,'LineSt
hold on
plot(t_month,Ave_A,'-','color','#A020F0','linewidth',2);
hold on
fill(X,YA,[160 32 240]./255,'EdgeColor',[160 32 240]./255,'LineStyle','-.','FaceAlpha',0.2);


%% Output image setting

set(gca,'XLim',[0 12]);
set(gca,'YLim',[0 100]);
set(gca,'Linewidth',1);
set(gca,'FontSize',20);
set(gca,'FontName','Arial');
xlabel('Time (months)');
ylabel('Overall survival (%)');
title('CD8+ CTL');
legend('CTL high','95% CI','CTL low','95% CI','location','east');
legend('boxoff')
set(gcf,'unit','centimeters','position',[37 3.5 18 13])
print('CTL_OS','-dpng','-r600')

end
```

```
function [Ave_low,Ave_high]=proana_OS_CTL()

n=1;
t=366;

[Q_low,Q_high]=Sur_Analysis_CTL(t,n);

Ave_low=sum(Q_low,2)./n;
Ave_high=sum(Q_high,2)./n;

end
```

```
function [Q_low,Q_high]=Sur_Analysis_CTL(t,n)

mu_min=0.60;
mu_max=0.90;
sigma_min=0.050;
sigma_max=0.080;

OS_low=zeros(t,1);
OS_high=zeros(t,1);
Q_low=zeros(t,n);
Q_high=zeros(t,n);

for j=1:n

    mu=mu_min+rand(1)*(mu_max-mu_min);
    sigma=sigma_min+rand(1)*(sigma_max-sigma_min);
    [Death_low,Death_high]=CTL_comput(mu,sigma);

    OS_low(1)=100;

    for i=2:1:t
        OS_low(i)=OS_low(i-1)-length(find(Death_low==i))*2;
        if OS_low(i)<0
           OS_low(i)=0;
        end
    end

    OS_high(1)=100;

    for k=2:1:t
        OS_high(k)=OS_high(k-1)-length(find(Death_high==k))*0.4;
        if OS_high(k)<0
           OS_high(k)=0;
        end
    end

    Q_low(:,j)=OS_low;
    Q_high(:,j)=OS_high;

end
```

```
function [Death_low,Death_high]=CTL_comput(mu,sigma)

%% Tumor proportion
sample=50;
t=366;

X_low=zeros(366,sample);
X_high=zeros(366,sample);

%% load
file_1='./ProAnalysis/Th8_low.mat';
A1=cell2mat(struct2cell(load(file_1)));

file_2='./ProAnalysis/Th8_high.mat';
A2=cell2mat(struct2cell(load(file_2)));

file_K1='./ProAnalysis/Th8_K3_low.mat';
B1=cell2mat(struct2cell(load(file_K1)));

file_K2='./ProAnalysis/Th8_K3_high.mat';
B2=cell2mat(struct2cell(load(file_K2)));

for i=1:sample
X_low(:,i)=A1(:,i)./B1(i);
X_high(:,i)=A2(:,i)./B2(i);
end

%% comput

D=ones(t,sample+1);
D(:,1)=1:1:t;

Death_low_time=zeros(1,sample+1);
Death_high_time=zeros(1,sample+1);

for i=2:sample+1
    for j=1:t
        P_death_low=1/(1+exp(-(X_low(j,i-1)-mu)/sigma));
        P=[1-P_death_low P_death_low];
        Q=[1 0];
        D(j,i)=randsrc(1,1,[Q;P]);
        if D(j,i) == 0
            Death_low_time(i)=D(j,1);
            break;
        else
            continue;
        end
    end
end

for i=2:sample+1
    for j=1:t
        P_death_high=1/(1+exp(-(X_high(j,i-1)-mu)/sigma));
        P=[1-P_death_high P_death_high];
        Q=[1 0];
        D(j,i)=randsrc(1,1,[Q;P]);
        if D(j,i) == 0
            Death_high_time(i)=D(j,1);
            break;
        else
            continue;
        end
    end
end

Death_low=sort(Death_low_time(2:1:end));
Death_high=sort(Death_high_time(2:1:end));

end
```

(Figure 7. Flowchart of Virtual Patient Survival Analysis.)

Explanation 1: The calculation process for Fig.5 is similar to that of Fig.8. We will save the finally processed data in the plotting folder. The results of Fig.4 can be directly obtained from the data of Fig.5. It should be noted that Fig.5 and Fig.8 are the results of stochastic calculations, and there may be differences each time they are run.

Explanation 2: The **"Fig.4"** folder illustrates the comparison of long-term treatment outcomes (M-OS and survival frequency) between model simulations and clinical data. The clinical data extracted from clinical trials are directly saved in the corresponding folder, and the numerical data from model simulations are placed in the plotting program.

Explanation 3: The **"Fig.5"** folder is dedicated to plotting survival curves. The survival curve data extracted from clinical trials are stored in the **"Placebo_Clin"** folder. The **"Ave.mat"** file saves the corresponding model-calculated survival curve values for direct access. The folders **"Placebo_VP"**, **"TAS102_VP"**, and **"TASBEV_VP"** are used to store data for the 100 baseline patients in the control group, the TAS-102 chemotherapy group, and the TAS-102 combined with Bevacizumab group, respectively. As mentioned earlier, this part of the data can be obtained from the corresponding author.

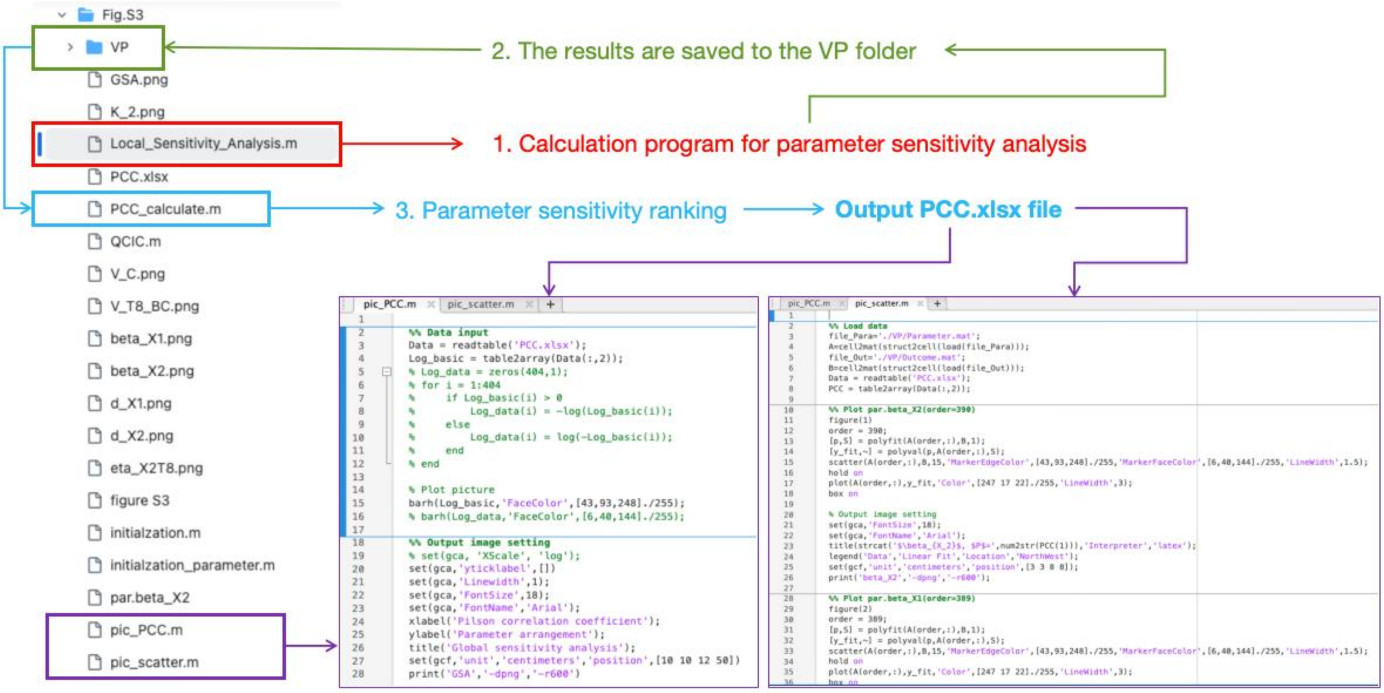
Explanation 4: The **"Fig.8"** folder is used for plotting survival analysis results based on high and low expression levels of different biomarkers. All process files and plotting programs are stored within the **"Fig.8"** folder. Similarly, due to the volume of data, the virtual patient data in **"TASBEV_VP"** can be obtained from the corresponding author.

Parameter sensitivity analysis

(Fig.S3 and Fig.S4)

Explanation 1: The "Fig.S3" folder is used for plotting the results of the global sensitivity analysis. The "VP" folder is designated for storing the results of 1,000 calculations. This part of the data can be generated by running the relevant programs, or you can contact the corresponding author to obtain the raw data.

Explanation 2: The "Fig.S4" folder is used for plotting the global sensitivity analysis results of the 21 tumor heterogeneity parameters required for generating virtual patients.



2. The results are saved to the VP folder

1. Calculation program for parameter sensitivity analysis

3. Parameter sensitivity ranking → Output PCC.xlsx file

```
%% Data input
Data = readtable('PCC.xlsx');
Log_basic = table2array(Data(:,2));
% Log_data = zeros(404,1);
% for i = 1:404
%     if Log_basic(i) > 0
%         Log_data(i) = -log(Log_basic(i));
%     else
%         Log_data(i) = log(-Log_basic(i));
%     end
% end

% Plot picture
barh(Log_basic,'FaceColor',[43,93,248]./255);
% barh(Log_data,'FaceColor',[6,40,144]./255);

%% Output image setting
% set(gca, 'XScale', 'log');
set(gca,'yticklabel',[])
set(gca,'Linewidth',1);
set(gca,'FontSize',18);
set(gca,'FontName','Arial');
xlabel('Pilson correlation coefficient');
ylabel('Parameter arrangement');
title('Global sensitivity analysis');
set(gcf,'unit','centimeters','position',[10 10 12 50])
print('GSA','-dpng','-r600')
```

```

%% Load data
file_Para='./VP/Parameter.mat';
A=cell2mat(struct2cell(load(file_Para)));
file_Out='./VP/Outcome.mat';
B=cell2mat(struct2cell(load(file_Out)));
Data = readtable('PCC.xlsx');
PCC = table2array(Data(:,2));

%% Plot par.beta_X2(order=390)
figure(1)
order = 390;
[p,S] = polyfit(A(order,:),B,1);
[y_fit,~] = polyval(p,A(order,:),S);
scatter(A(order,:),B,15,'MarkerEdgeColor',[43,93,248]./255,'MarkerFaceColor',[6,40,144]./255,'LineWidth',1.5);
hold on
plot(A(order,:),y_fit,'Color',[247 17 22]./255,'LineWidth',3);
box on

% Output image setting
set(gca,'FontSize',18);
set(gca,'FontName','Arial');
title(strcat('$\beta_{X_2}$, $P$=',num2str(PCC(1))),'Interpreter','latex');
legend('Data','Linear Fit','Location','NorthWest');
set(gcf,'unit','centimeters','position',[3 3 8 8]);
print('beta_X2','-dpng','-r600');

%% Plot par.beta_X1(order=389)
figure(2)
order = 389;
[p,S] = polyfit(A(order,:),B,1);
[y_fit,~] = polyval(p,A(order,:),S);
scatter(A(order,:),B,15,'MarkerEdgeColor',[43,93,248]./255,'MarkerFaceColor',[6,40,144]./255,'LineWidth',1.5);
hold on
plot(A(order,:),y_fit,'Color',[247 17 22]./255,'LineWidth',3);
box on
```

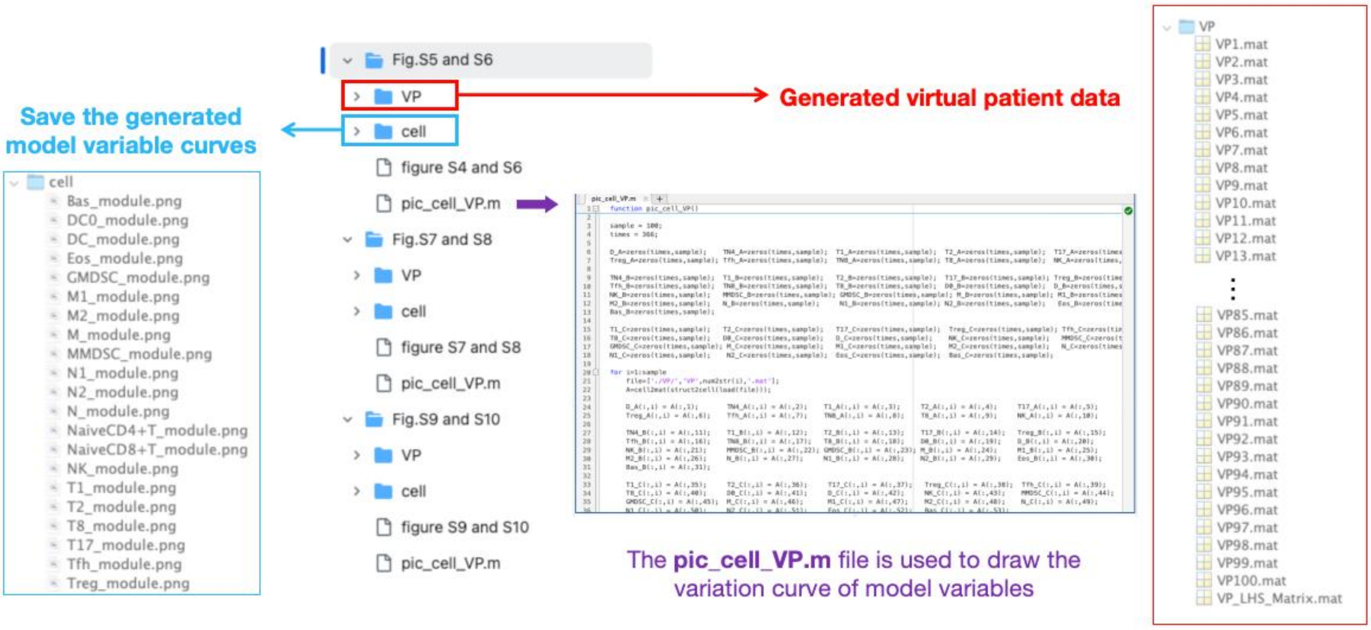
4. The pic-PCC.m file displays the sensitivity analysis results, while the pic_scatter.m file generates a random scatter plot of the eight most sensitive parameters.

(Figure 8. Explanation of the main running program for sensitivity analysis.)

Visualization of process data

(Fig.S5 ~ Fig.S10)

Explanation : The "Fig.S5" to "Fig.S10" folders are used for plotting the output results of model variables in the control group, the TAS-102 chemotherapy group, and the TAS-102 combined with Bevacizumab group. The "VP" folder contains the generated virtual patient data, which can be self-generated or obtained from the corresponding author.



Save the generated model variable curves

Generated virtual patient data

```
function pic_cell_VP()

sample = 100;
times = 366;

D_A=zeros(times,sample);    TN4_A=zeros(times,sample);  T1_A=zeros(times,sample);  T2_A=zeros(times,sample);  T17_A=zeros(times
Treg_A=zeros(times,sample); Tfh_A=zeros(times,sample);  TN8_A=zeros(times,sample); T8_A=zeros(times,sample);  NK_A=zeros(times,

TN4_B=zeros(times,sample);  T1_B=zeros(times,sample);   T2_B=zeros(times,sample);  T17_B=zeros(times,sample); Treg_B=zeros(time
Tfh_B=zeros(times,sample);  TN8_B=zeros(times,sample);  T8_B=zeros(times,sample);  D0_B=zeros(times,sample);  D_B=zeros(times,s
NK_B=zeros(times,sample);   MMDSC_B=zeros(times,sample); GMDSC_B=zeros(times,sample); M_B=zeros(times,sample); M1_B=zeros(times
M2_B=zeros(times,sample);   N_B=zeros(times,sample);     N1_B=zeros(times,sample); N2_B=zeros(times,sample);   Eos_B=zeros(time
Bas_B=zeros(times,sample);

T1_C=zeros(times,sample);   T2_C=zeros(times,sample);   T17_C=zeros(times,sample);  Treg_C=zeros(times,sample); Tfh_C=zeros(tim
T8_C=zeros(times,sample);   D0_C=zeros(times,sample);   D_C=zeros(times,sample);    NK_C=zeros(times,sample);   MMDSC_C=zeros(t
GMDSC_C=zeros(times,sample); M_C=zeros(times,sample);   M1_C=zeros(times,sample);   M2_C=zeros(times,sample);   N_C=zeros(times
N1_C=zeros(times,sample);    N2_C=zeros(times,sample);  Eos_C=zeros(times,sample);  Bas_C=zeros(times,sample);

for i=1:sample
    file=['./VP/','VP',num2str(i),'.mat'];
    A=cell2mat(struct2cell(load(file)));

    D_A(:,i) = A(:,1);       TN4_A(:,i) = A(:,2);    T1_A(:,i) = A(:,3);    T2_A(:,i) = A(:,4);     T17_A(:,i) = A(:,5);
    Treg_A(:,i) = A(:,6);    Tfh_A(:,i) = A(:,7);    TN8_A(:,i) = A(:,8);   T8_A(:,i) = A(:,9);     NK_A(:,i) = A(:,10);

    TN4_B(:,i) = A(:,11);    T1_B(:,i) = A(:,12);    T2_B(:,i) = A(:,13);   T17_B(:,i) = A(:,14);   Treg_B(:,i) = A(:,15);
    Tfh_B(:,i) = A(:,16);    TN8_B(:,i) = A(:,17);   T8_B(:,i) = A(:,18);   D0_B(:,i) = A(:,19);    D_B(:,i) = A(:,20);
    NK_B(:,i) = A(:,21);     MMDSC_B(:,i) = A(:,22); GMDSC_B(:,i) = A(:,23); M_B(:,i) = A(:,24);    M1_B(:,i) = A(:,25);
    M2_B(:,i) = A(:,26);     N_B(:,i) = A(:,27);     N1_B(:,i) = A(:,28);   N2_B(:,i) = A(:,29);    Eos_B(:,i) = A(:,30);
    Bas_B(:,i) = A(:,31);

    T1_C(:,i) = A(:,35);     T2_C(:,i) = A(:,36);    T17_C(:,i) = A(:,37);  Treg_C(:,i) = A(:,38);  Tfh_C(:,i) = A(:,39);
    T8_C(:,i) = A(:,40);     D0_C(:,i) = A(:,41);    D_C(:,i) = A(:,42);    NK_C(:,i) = A(:,43);    MMDSC_C(:,i) = A(:,44);
    GMDSC_C(:,i) = A(:,45);  M_C(:,i) = A(:,46);     M1_C(:,i) = A(:,47);   M2_C(:,i) = A(:,48);    N_C(:,i) = A(:,49);
    N1_C(:,i) = A(:,50);     N2_C(:,i) = A(:,51);    Eos_C(:,i) = A(:,52);  Bas_C(:,i) = A(:,53);
```

The **pic_cell_VP.m** file is used to draw the variation curve of model variables

(Figure 9. Flowchart for drawing the evolution curves of model variables.)