IPDfromKM: Reconstruct Individual Patient Data from Published Kaplan-Meier Survival Curves

Na Liu,[1] Yanhong Zhou,[1] J. Jack Lee*

Abstract

Background: When applying secondary analysis on published survival data, it is critical to obtain each patient’s raw data, because the individual patient data (IPD) approach has been considered as the gold standard of data analysis. However, researchers often lack access to the IPD. We aim to propose a straightforward and robust approach to help researchers to obtain IPD from published survival curves with a friendly software platform.

Results: Improving upon the existing methods, we proposed an easy-to-use, two-stage approach to reconstruct IPD from published Kaplan-Meier (K-M) curves. Stage 1 extracts raw data coordinates and Stage 2 reconstructs IPD using the proposed method. To facilitate the use of the proposed method, we develop the R package IPDfromKM and an accompanied web-based Shiny application. Both the R package and Shiny application have an “all-in-one” feature such that users can use them to extract raw data coordinates from published K-M curves, reconstruct IPD from data coordinates extracted, visualize the reconstructed IPD, assess the accuracy of the reconstruction, and perform secondary analysis on the IPD. We illustrate the use of the R package and the Shiny application with K-M curves from published studies. Extensive simulations and real world data applications demonstrate that the proposed method has high accuracy and great reliability in estimating the number of events, number of patients at risk, survival probabilities, median survival times, as well as hazard ratios.

Conclusions: IPDfromKM has great flexibility and accuracy to reconstruct IPD from published K-M curves with different shapes. We believe that the R package and the Shiny application will greatly facilitate the potential use of quality IPD data and advance the use of secondary data to make informed decision in medical research.

Keywords: individual patient data (IPD); Kaplan-Meier curve; meta-analysis; R package; survival analysis; Shiny; web-based application

[1] Equal contribution.
Background

Typical information used for meta-analysis of survival data reported from clinical trials often includes a summary for the outcomes of each arm, hazard ratios, and Kaplan-Meier (K-M) curves along with the numbers of patients at risk [1]. When applying secondary analysis on such published survival data, difficulties usually come from insufficient details in the reported data, which are often reported using aggregated summary statistics. For example, when conducting the meta-analysis on time-to-event data, it is possible that the proportional hazard ratio assumption may not hold, and alternative measures of the survival difference are needed to avoid bias [2, 3]. In this case, it is of great importance to obtain individual patient data (IPD), with which one can not only perform the standard survival analysis, but also can assess if the assumption of the original method is appropriate in order to decide whether or not alternative methods should be applied [4]. Furthermore, one can undertake additional subgroup analyses not reported in the aggregated data. For this reason, the IPD approach is considered as the gold standard in data analysis.

However, researchers conducting meta-analysis or other secondary analyses may lack access to the IPD, partly due to the confidentiality of clinical data. Therefore, a method that is able to reconstruct IPD from published K-M curves can greatly facilitate secondary analyses on survival data. Several methods have been reported in the literature. The iterative algorithm based on Kaplan-Meier (K-M) estimation method (referred to as “iKM” hereafter) proposed by Guyot et al. [5] is a classic approach among many proposed. It has been used in various secondary analyses. For example, Satagopan et al. [6] used it to reconstruct time-to-event data from a melanoma data set for evaluating different treatment benefits according to biomarker subgroups. Wei and Royston [7] developed a STATA function to apply the iKM algorithm with some adaptions and applied it to reconstruct IPD from K-M curves from multiple trials.

The iKM method, however, does have several limitations. First, external software is needed to extract data points before using the iKM algorithm. The original iKM method suggests a manual approach to pick up a sufficient number of points from K-M curves via mouse-clicks using the DigitizeIt software. This manual approach is recommended because several K-M curves on the same graph can intertwine with each other, making them hard to read out automatically.

Second, the iKM method has restricted requirements when picking up the points manually: (1) the survival probability needed to decrease monotonically as time increases, (2) the points where the numbers of patients at risk are reported must be included, and (3) users need to sort the data coordinates into time intervals determined by the time points, for which the number of patients at risk is reported. K-M curves in publication typically report the number of patients at risk at several time points under the x-axis. For example, if number of patients at risk is reported at month 3, 6, and 9, then data coordinates need to be manually organized into the time intervals: [0 − 3), [3 − 6), [6 − 9), and [9, + ∞). The first two requirements are generally hard to keep for a large amount of manual mouse-clicks and the last one is also time consuming. Thus, it is important to have a flexible function to perform these tasks automatically.

Third, there is a numeric issue in the original iKM method where the iterative procedure may result in infinite loops, which is due to a boundary set up for the
number of censored patients. Also, the number of events tends to be underestimated when there are multiple, consecutive vertical points for one step.

Finally, there is a lack of user-friendly software for clinical researchers to use this method. The published functions [5, 8, 6] for the iKM method have at least one of the following inconveniences: external digitizing software is need for data extraction; users need to manually check if data input is appropriate, or the auto process program not convenience or stable; no accuracy assessment is provided for users to directly evaluate the reconstruction results within the same software.

To overcome the aforementioned limitations, we proposed the two-stage modified-iKM approach to extract IPD, which provides an improved, accurate, user-friendly, and stable workflow to reconstruct IPD. We not only relaxed the restricted requirement for data input, but also improved the flexibility and stability of the original iKM method. More importantly, we develop an all-in-one software platform that allows clinicians or medical researchers to go from a single K-M curve image directly to reconstructed data, without using additional software to aid data extraction or without manual treatment on the data before reconstruction. The software includes an \texttt{R} package and a web-based Shiny application. Specifically, users can use either the \texttt{R} package or the Shiny application to (1) extract data coordinates from published K-M curves, (2) preprocess extracted data points, (3) run the modified-iKM algorithm to estimate the number of patients at risk, the number of events, and the number of censored outcomes for each pre-specified interval, and to reconstruct IPD, (4) provide graphs and statistical summary for users to evaluate the accuracy of the reconstruction process, and (5) conduct survival analysis based on reconstructed IPD. The \texttt{R} package is beneficial for users with some programming skills so that they can perform the reconstruction IPD within \texttt{R} or expand their future research on the package. The advantage of the Shiny application is that it has a point-and-click interface, does not require installing any software, and can be used on any machine with an internet browser. This feature is appealing for clinicians and medical researchers who are not necessarily familiar with statistical programming. Extensive simulations and real world data applications demonstrate that the proposed method has high accuracy and great reliability in the estimations of the parameters of interest, e.g., median survival, hazard ratio, and survival probability.

Our work is of great importance because it not only further supplements and strengthens the original iKM algorithm, but it also provides a user-friendly, all-in-one software available in different platforms, accommodating the needs of researchers with or without familiarity with statistical programming. The availability of the two different platforms of software can further enhance quality extraction of IPD for meta-analysis and other secondary analyses using time-to-event data.

**Implementation**

The modified-iKM algorithm for IPD reconstruction is a two-stage process, as shown in Figure 1. Stage 1 aims to extract quality data coordinates \((\text{time, survival probability})\) from K-M curves. In Stage 2, the data coordinates are preprocessed and IPD is reconstructed using the proposed iterative algorithm.
Coordinates extraction in Stage 1

Data coordinates can be extracted from K-M curves using the \( R \) function and Shiny application we develop (https://www.trialdesign.org/one-page-shell.html#IPDfromKM). There are also a number of other software options available on Windows or Mac operating systems to digitize the graphs. The commonly used software are DigitizeIt (http://www.digitizeit.de/), ScanIt (http://amsterchen.com), and Plot Digitizer (http://plotdigitizer.sourceforge.net/). Extensive applications of real world trial examples and simulated K-M curves show that data extracted using the different approaches yield comparable results during IPD reconstruction in Stage 2 (more details in Implementation and Simulation Result sections). We provide video tutorials on how to extract data coordinates using these tools in the Shiny application.

To ensure accuracy of estimation in Stage 2, it is critical to extract quality data coordinates in Stage 1. We recommend the following when extracting data points:

1. When there are multiple K-M curves in the same figure, if the curves are tangled together or there are censoring markers on the curve of interest, use Adobe Illustrator to separate lines (a video tutorial on this is included in the Shiny application) and then use the software mentioned above to extract data points.

2. Extract as many points as possible. This is attainable using our package or the software aforementioned.

3. Make sure the data points extracted are as evenly distributed as possible on the K-M curves.

4. Be sure to click on both the top and bottom points on the segment at the points where survival probability drops (forming a segment on the curve).

IPD reconstruction in Stage 2

The IPD reconstruction is carried out using the modified-iKM algorithm, which is based on the K-M estimation method [9] and improved upon the iKM algorithm [5]. We provide an overview of the algorithm in this section and delineate it in detail in Appendix 0.1. Let \( T_k \) and \( S_k \) denote the time and survival probability, respectively, at time \( k \). The data points extracted in Stage 1 is typically a \( N \times 2 \) table, with each row being \((T_k, S_k)\), for \( k = 1, 2, ..., N \), where \( N \) is the number of data points extracted. The IPD reconstruction consists of two main steps, as noted below.

1. Process the raw data.
   (a) Sort the data by \( T_k \).
   (b) Make monotonicity adjustment by first using Turkey’s fences to detect and remove unreasonable inputs, and then ensure that survival probability decreases over time.
   (c) Perform step control to improve accuracy of estimation.

2. Reconstruct IPD using an iterative algorithm adapted from the iKM method.
   (a) Estimate K-M parameters at each coordinate \( (k) \) including the number of patient at risk \( \hat{n}_k \), number of patients censored \( \hat{cens}_k \), and number of events\( \hat{d}_k \), \( k = 1, \cdots, N \). In this step, we modified the boundary setup for the number of censored observations to prevent infinite loops.
   (b) Construct IPD using the parameters estimated from Step (2a).
Steps (1b) and (1c) are great improvements in comparison to the original iKM method since they improve the flexibility of the method. The flexibility ensures that when users accidentally click on some points away from the K-M curve, the algorithm can detect such points and exclude them when reconstructing data, and that the number of events can be more accurately estimated. The modification in Step (2a) enhances the stability of the iterative algorithm. More details for the modified-iKM algorithm are provided in Appendix 0.1.

To assess the accuracy of the modified-iKM algorithm, we employ several metrics and make them easily accessible through the software we develop. First, we provide graphs to visualize the reconstructed results: (1) estimated survival probability at each read-in time point using the reconstructed IPD is compared with the corresponding read-in survival probabilities, and (2) estimated number of patients at risk is also compared with reported values, when the number of patients at risk is provided. Second, we provide several summary statistics to aid the accuracy assessment. One is the root mean square error (RMSE), which measures the difference in survival probabilities calculated using reconstructed data and original data. Additionally, the mean absolute error and the max absolute error are also provided to assess the precision of the estimation. Third, we use the Kolmogorov-Smirnov test to compare the distribution of the read-in and the estimated survival curves. A large p-value is desired, as it indicates that the discrepancy between the read-in and estimated survival curves is small.

Results

R package
The R package IPDfromKM is available from the Comprehensive R Archive Network (CRAN) at https://CRAN.R-project.org/package=IPDfromKM. The package contains several functions; the descriptions and objects returned for the functions are presented in Table 1. We provide an example for each of the functions below.

Extract data coordinates
The getpoints() function is used to extract data coordinates from published K-M curves. The function has the following arguments:
- f the K-M curves in the bitmap images (e.g., .png, .jpeg, .bmp, .tiff).
- x1: the actual label of the left-most points on x-axis.
- x2: the actual label of the right-most points on x-axis.
- y1: the actual label of the lowest point on y-axis.
- y2: the actual label of the highest point on y-axis.

For the image, the use of .png file is highly recommend, since it can shorten the processing time in R. In addition to the image itself, two x-coordinates (x1 and x2) and two y-coordinates (y1 and y2) are needed to decide the location and scale of the coordinates system. Below is an example to read in an image for data extraction.

R > points <- getpoints("filepath/filename.png",x1=0,x2=60,y1=0,y2=1)

After the file is read into R, instruction will be provided in the R console to guide the extraction of data points. Specifically, users need to click on the leftmost and rightmost points on the x-axis, and click on the lowest and the highest points on
the y-axis. Then they can collect the data coordinates by mouse-clicking on the curve. To get a desirable estimation, we suggest that users collect points such that the extracted points are as evenly distributed across the curve as possible, and that they include the points where the survival probability drops. The data points extracted will be returned as a two-column data set (e.g., \textit{points} in the example), and this data set can be used as the input of \texttt{preprocess()} function described below.

In the following text, we take the build-in data set \texttt{Radiationdata} to demonstrate the use of the package to preprocess raw data, reconstruct IPD, and conduct secondary analysis on the reconstructed IPD. The data set was extracted from published K-M curves from a two-arm randomized controlled trial \cite{10} using ScanIt. This study randomized 424 head and neck cancer patients to two treatment groups: 213 in radiotherapy group (referred to as “radio”) and 211 in the radiotherapy plus cetuximab group (referred to as “radio_plus”). The primary outcome was the duration of locoregional control. There were 145 pairs of coordinates extracted for the radio treatment, and 136 pairs for the radio_plus treatment. Coordinates for each treatment were saved as a two-column table in \texttt{Radiationdata}: the first column is for survival times, and the second column is for survival probabilities reported in percentages. \texttt{Radiationdata} also includes risk times (in months): \textit{trisk} = (0, 10, 20, 30, 40, 50, 60, 70) and the number of patients at risk at each risk time point (\textit{nrisk}).

\textbf{Process data coordinates}

To prepare data to reconstruct IPD, we first preprocess the raw data points in an appropriate format. This can be done using the \texttt{preprocess()} function. This function contains the following arguments:

- \texttt{dat}: a two-column data set with the first column being survival time, and the second the survival probability extracted from a published K-M curve.
- \texttt{trisk}: a vector containing risk time points (i.e., time points at which the number of patients at risk are reported). This often can be found under the x-axis of a K-M curve.
- \texttt{nrisk}: a vector containing the number of patients at risk. This often can be found under the x-axis of a K-M curve.
- \texttt{maxy}: the scale of survival probability. Set \texttt{maxy=100} when the probabilities are reported in percentages (e.g., 70\%). Set \texttt{maxy=1} when the probabilities are reported using decimal numbers (e.g., 0.7).

\begin{verbatim}
R > names(Radiationdata)
[1] "radio" "radioplus" "trisk" "nrisk.radio" "nrisk.radioplus"
R > Radiationdata$trisk
[1] 0 10 20 30 40 50 60 70
R > Radiationdata$nrisk.radio
[1] 213 122 80 51 30 10
R > Radiationdata$nrisk.radioplus
[1] 211 143 101 66 35 9
R > pre_radio <- preprocess(dat= Radiationdata$radio, trisk= Radiationdata$trisk, nrisk= Radiationdata$nrisk.radio, maxy=100)
R > pre_radio_plus <- preprocess(dat= Radiationdata$radioplus, trisk= Radiationdata$trisk, nrisk= Radiationdata$nrisk.radioplus, maxy=100)
\end{verbatim}
The output of the `preprocess()` function is a class object that can be used directly in the `getIPD()` function to construct IPD.

**Reconstruct IPD**

After the raw data is processed using the `preprocess()` function, we can use the `getIPD()` function to reconstruct the IPD. The function has the following arguments:

- `prep`: the class object returned from the `preprocess()` function.
- `armID`: an arbitrary label used as the group indicator for the reconstructed IPD. Typically 0 for the control group and 1 for the treatment group.
- `tot.events`: total number of events. Only available for some published curves, and the default value is NULL.

```r
R > est_radio <- getIPD(prep=pre_radio, armID=0, tot.events=NULL)
R > est_radio_plus <- getIPD(prep=pre_radio_plus, armID=1, tot.events=NULL)
```

**Accuracy assessment**

To view the accuracy assessment results, simply call the `summary()` function. Because of page limits, we show only one example below.

```r
R > summary(est_radio)
```

| interval | lower | upper | trisk | nrisk | nrisk.hat | censor.hat | event.hat |
|----------|-------|-------|-------|-------|-----------|------------|-----------|
| 1        | 1     | 42    | 0     | 213   | 213       | 8          | 83        |
| 2        | 43    | 66    | 10    | 122   | 122       | 10         | 32        |
| 3        | 67    | 88    | 20    | 80    | 80        | 19         | 10        |
| 4        | 89    | 109   | 30    | 51    | 51        | 16         | 5         |
| 5        | 110   | 129   | 40    | 30    | 30        | 17         | 3         |
| 6        | 130   | 144   | 50    | 10    | 10        | 9          | 1         |

The root-mean-square error between estimated and read-in survival probabilities is 0.004.
The mean absolute error between estimated and read-in survival probabilities is 0.003.
The max absolute error between estimated and read-in survival probabilities is 0.013.

The Kolmogorov-Smirnov test:
Test statistics D= 0.07639 p-value= 0.7948
Null hypothesis: distributions of the read-in and estimated survival probabilities are the same.

We see that our algorithm can accurately estimate the numbers at risk and provide estimates on number of events at each risk time. The small values for RMSE and the max absolute error of the estimation, along with the large p-value of the
Kolmogorov-Smirnov test shows that the reconstructed IPD is accurate. Additionally, we can use the `plot()` function to graph the survival curves from the reconstructed IPD, and compare them with those generated using original data points. The function takes the object returned by `getIPD()` directly.

```r
R > plot(est_radio)
R > plot(est_radio_plus)
```

The output using the `plot()` for the radio group is provided in Figure A.1 (Appendix 0.2), which shows three graphs: (1) K-M curve comparison with estimated versus read-in; (2) number of patients at risk using the reconstructed IPD versus reported; and (3) difference in survival probability at different time points for the reconstructed IPD data set and the read-in data set. When the interval information is not available while reconstructing the IPD, the second graph will not be shown.

### Secondary analysis

If survival analysis is of interest, we can run the `survreport()` function, which includes the following arguments:

- `ipd1`: a three-column (i.e., time, status, treatment indicator) table of IPD for treatment 1.
- `ipd2`: a three-column (i.e., time, status, treatment indicator) table of IPD for treatment 2.
- `arms`: number of treatment arms. It takes the value of either 1 or 2.
- `interval`: the time intervals for which the landmark survival probabilities are of interest. The default is at every 6 months.
- `s`: the survival probabilities for which the corresponding survival times are of interest, e.g., \( s=0.5 \) means that the median survival time is desired.

Researchers working with clinical data are often interested in the survival times at which survival probabilities are specified (e.g., median survival time at which 50% of patients survive). The example below shows the survival times for the pre-specified survival probability \( s = (0.50, 0.75, 0.95) \). This function also returns K-M curves and cumulative risk in a figure for both groups (Figure A.2 in Appendix 0.2).

```r
R > survreport(ipd1=est_radio$IPD,ipd2=est_radio_plus$IPD,
+ arms=2,interval=8,s=c(0.50,0.75,0.95))

print(report$arm1)
$survtime

| time  | 0.95LCI | 0.95UCI |
|-------|---------|---------|
| survprob = 0.95 | 2.59 | 0.779 | 3.37 |
| survprob = 0.75 | 5.73 | 4.680 | 7.55 |
| survprob = 0.5  | 14.80| 11.700| 23.10|

> print(report$arm2)
$survtime

| time  | 0.95LCI | 0.95UCI |
|-------|---------|---------|
| survprob = 0.95 | 3.1 | 2.07 | 5.17 |
| survprob = 0.75 | 8.3 | 7.52 | 10.60 |
```
survprob = 0.5 25.4 15.60 NA

Shiny Application
To facilitate the use of the modified-iKM algorithm to reconstruct IPD for a broader audience, who are not necessarily familiar with statistical programming, we develop a user-friendly Shiny application, which is freely available at https://www.trialdesign.org/one-page-shell.html#IPDfromKM. The app has a straightforward interface with three main panels: Data Extraction (used to extract data points), Reconstruct Individual Patients Data (used to reconstruct IPD), and User Guide (providing extensive tutorials). Overall, the app has the capability to complete the following tasks.

1. Extract data points from published K-M curves.
2. Process data points extracted from published K-M curves.
3. Reconstruct IPD using the modified-iKM algorithm.
4. Assess the accuracy of the reconstruction.
5. Perform survival analysis using the reconstructed IPD data.
6. Generate a concise report for the IPD reconstruction.
7. Provide an extensive user guide for understanding the method and using the Shiny application.

Typical input required to use the app includes the following.

1. An image file (.png or .jpeg) for data point extraction.
2. A .csv or .txt file containing the coordinates extracted from published K-M curves (for IPD reconstruction). File templates are available in the Shiny application.
3. Risk time (trisk).
4. Number of patients at risk (nrisk).
5. Total number of patients reported (optional when information for nrisk is provided).
6. Total number of events reported (optional, but having it will improve accuracy).

We provide video tutorials on how to extract coordinates using the Shiny application and other software and two examples of using the app to reconstruct IPD from data coordinates extracted from K-M curves. The tutorial are accessible under the User Guide panel of the Shiny application. In the following text, we show an example of using the Shiny application to reconstruct IPD for two treatment groups simultaneously.

We illustrate the use of the app using data for the radio and radio_plus treatment groups introduced in the R package examples. The data set was saved in the radio_radioplus.csv file, which is provided in the app as a template. As shown in Figure 2, to input the data, we select “Two” under Number of treatment groups and upload the data file. After that, we type in available information, including risk times, number of patients at risks, and the total number of patients for each group. When finishing data input, we simply click the Begin Calculation button and the results are shown on the right side of the app. There are four tabs for displaying the results.
As shown in Figure 2, the first tab **Reconstructed IPD** shows the individual patient data reconstructed from the data provided. **Accuracy assessment** tab (Figure A.3) shows two plots and a table. The first plot displays the comparison of K-M curves using the IPD and original data set for each treatment group, followed by a plot of the difference between survival probabilities. The table shows the summary statistics such as root mean square error and Kolmogorov-Smirnov test statistics and p-values to help assess the accuracy of the IPD reconstruction. Under the **Survival analysis** tab (Figure A.4), K-M curves and cumulative hazard functions are displayed for both treatments. For each treatment, the landmark survival probabilities, corresponding standard error, and 95% confidence interval (CI) are reported. Below the landmark probability tables, the app also provides the critical survival times given pre-specified survival probabilities. For example, we see that the median survival time for treatment 1 (radio group) is 14.9 months and for treatment 2 (radio_plus group) is 24.5 months. A concise report is also available under the **Download report** tab (Figure A.5).

**Implementation example**

We now illustrate the use of the modified-iKM method for reconstructing IPD from K-M curves published for a two-arm randomized controlled trial in non-small cell lung cancer (NSCLC) patients [11]. This trial is known as the POLAR trial, in which a total of 287 patients with previously treated, advanced or metastatic NSCLC were randomized to receive either atezolizumab or docetaxel. The aim of this trial was to assess the efficacy of the two drugs for patients with NSCLC, analyzed by PD-L1 expression levels on tumor cells and tumor-infiltrating immune cells. Here, the baseline PD-L1 expression was scored by immunohistochemistry in tumor cells (as the percentage of PD-L1 expressing tumor cells TC3 (≥50%), TC2 (≥5% and <50%), TC1 (≥1% and <5%), and TC0 (<1%)) and tumor-infiltrating immune cells (as the percentage of tumor area: IC3 (≥10%), IC2 (≥5% and <10%), IC1 (≥1% and <5%), and IC0 (<1%)). Overall survival was estimated for all patients in five groups: TC3 or IC3 patients, TC2/3 or IC2/3 patients, TC1/2/3 or IC1/2/3 patients, and TC0 or IC0, and intention to treat for both atezolizumab and docetaxel treatments. The five K-M curves in the POLAR trial are shown in Figure A.6 (Appendix 0.4). To apply our method and assess its accuracy, we extract raw data points from the K-M curves, reconstruct IPD from the points extracted, use the reconstructed IPD to calculate the median overall survival (OS) and hazard ratio, and compare them with the published results. As shown in Table 2, the median OS calculated from the reconstructed IPD using our method were almost identical to the published results. We also note that the data points extracted using different software may have some effects on the results, indicating the importance of careful extraction of data points from K-M curves. As shown, we see that the data points extracted using our function from the IPDfromKM package have competitive performance to those extracted using DigitizeIt and ScanIt.

**Simulation Result**

To further assess the accuracy of the estimations using the modified-iKM algorithm, we conducted a simulation study with six trials that each had both control and
treatment groups. The sample size for each of the 12 groups was 200. We generated the underlying survival time \( T_i, i = 1, \ldots, 200 \) using the Weibull distribution with the survival function \( S(t) = \exp(-\lambda t^\gamma) \), where \( \lambda \) and \( \gamma \) are the scale and shape parameters, respectively [12]. When \( \gamma \) is larger than 1, the survival curve has an increasing hazard; when equal to 1, the survival curve has a constant hazard; and when less than 1, the survival curve has a decreasing hazard. The scale parameter was specified such that the mean survival \( E(X) = \lambda \Gamma(1 + 1/\gamma) \) was 12 months for treatment groups, and 6 months for control groups. We assume 36 months of recruiting time, and a maximum of 24 months of follow-up. The censoring time was generated by the minimum value of the time generated from exponential distribution with parameter \( \lambda^\ast \) and the maximum follow-up time. The value of \( \lambda^\ast \) was set such that the censor rate was either 30% or 60%. K-M curves were then generated by the \texttt{survfit()} function from the \textit{survival R} package. The number of patients at risk was reported every 3 months (20 intervals), or every 10 months (6 intervals), or not reported at all (no risk information). The simulated trials represent diverse situations mimicking K-M survival curves obtained from real trials.

We used the \textit{IPDfromKM} package, DigitizeIt, and ScanIt software to extract the coordinates of the curves, then we used the \texttt{preprocess()} and \texttt{getIPD()} functions from the \textit{IPDfromKM} package to process the raw data and obtain IPD. We then compare the estimated IPD with the true IPD generated. We first examine the accuracy of the three software packages in the coordinates extraction by comparing the estimated number at risk to the true number at risk. Figure 3a shows the results for two simulated curves. As demonstrated, the modified-iKM can accurately estimate the number of patients at risk regardless of software used to extract data points, as long as a study reports the number of patients at risk, under both low and higher censoring rates. Without reported numbers at risk, estimation cannot be accurate, as previously noted in the literature [5]. Thus, reporting this information in published studies is highly recommended.

Figure 3b shows the mean absolute error of the estimation for the number of patients at risk when the reported number at risk was available (i.e., the case with 6 or 20 time intervals in the trial), where the error was determined by \( \text{error} = \text{estimated value} - \text{true value} \). The data extracted yielded comparable results for the different software. The results were more accurate with more time intervals provided. For instance, when there was information for 20 intervals, the mean absolute error was less than 2 regardless of censoring rate. When the censoring rate was large (e.g., 60%), the estimation using information from six intervals had greater error.

Next, we used the reconstructed IPD to conduct standard survival analysis to evaluate the performance of our method. Figure 4a shows the estimated median survival times and the one-year survival probability. The difference between the estimated and true median survival times was within 5% in most cases. Figure 4b shows the comparison between the true hazard ratio and estimated hazard ratios on the log scale. The hazard ratios and corresponding standard deviations are almost identical in most cases, regardless of which software was used to extract the data coordinates.
Conclusion

In this work, we introduce an R package, \textit{IPDfromKM}, and a web-based Shiny application, to reconstruct IPD from published K-M curves, based on the proposed modified-iKM algorithm. There are several improvements in the modified-iKM algorithm, in comparison to the original iKM method. First, we provided a function for users to extract data points from K-M curves directly. This enables our work to have the all-in-one feature, which streamlines the process of IPD reconstruction without requiring extra software to obtain the raw data coordinates. Our real world examples and simulated examples show that the data points extracted using the \textit{IPDfromKM} package yield competitive results to those extracted using DigitizeIt and ScanIt as long as they have similar number of points extracted.

Second, our procedure to preprocess raw data coordinates is flexible and accurate. The original iKM method does not have a function for preprocessing raw data coordinates, and users need to manually check the monotonicity assumption and sort the extracted data points. Stagopan et al.\cite{6} published a function that can automatically preprocess the raw data coordinates. However their function tends to be unstable. For example, the function simply deletes the points having a larger survival probabilities than the points right before them, which introduces additional errors into the extracted data points. It also trims data points at the tail of a K-M curve. This can be detrimental as the K-M curve often has a long horizontal tail. If data points were trimmed at the tail, it would be less likely to reliably estimate the number of censored observations occurring at the end of the trial. The \texttt{preprocess()} function from the \textit{ReconstructKM} package published on GitHub \cite{8} can preprocess the data coordinates but it also requires manual check for monotonicity and the curve endings. To overcome these limitations, we proposed a flexible and accurate algorithm to preprocess the raw data coordinates extracted from K-M curves and the algorithm is easily implemented by using the \texttt{preprocess()} function from the \textit{IPDfromKM} package or by uploading the data to the Shiny application.

Third, the modified-iKM algorithm further improves the accuracy and stability of the reconstruction process. We propose a step control procedure for the number of events at jumping steps in the K-M curves to ensure the accuracy of estimation of the number of events. We also refine the boundary set up in the iterative procedure to prevent endless loops from happening, which greatly improves the stability of the iKM method.

Finally, we provide several approaches to evaluate the accuracy of the algorithm with an easy-to-use function, with which the quality of the reconstructed IPD is easily assessed. The assessment can be easily carried out using the \textit{IPDfromKM} package or the Shiny application. Both the R package and the Shiny application have the all-in-one feature that allows clinicians or medical researchers to go from an image file to reconstructed data without the need for additional software.

Despite the strengths of the modified-iKM algorithm and its accompanying software, there are a couple of challenges or limitations worth mentioning. First, when there are tangled lines with a lot of the censoring markers, even with the help of software such as Adobe Illustrator, it is still challenging to separate out each K-M curve for digitizing. Second, while the modified-iKM algorithm can accurately estimate the survival probabilities, the number of events, and the number of patients at...
risk, the algorithm is based on a uniform censoring assumption, which may be violated in certain trials. Thus, future work can expand on this and consider extension to non-uniform censoring mechanism.

Availability and requirements
Project name: IPDfromKM R package and Shiny web application.
Project home page: [https://CRAN.R-project.org/package=IPDfromKM](https://CRAN.R-project.org/package=IPDfromKM), and [https://www.trialdesign.org/one-page-shell.html#IPDfromKM](https://www.trialdesign.org/one-page-shell.html#IPDfromKM).
Operating system(s): Platform independent.
Programming language: R.
Other requirements: none.
License: GPL-2.
Any restrictions to use by non-academics: none

Abbreviations
IPD: individual patient data. iKM: iterative algorithm based on Kaplan-Meier estimation. RMSE: normalized root mean square error. K-M: Kaplan-Meier.

Ethic approval and consent to participate
Not applicable. This research uses only data published and does not involve human subjects, human material, or human data.

Consent for publication
Not applicable.

Availability of data and material
The R code for the project can be download at [https://CRAN.R-project.org/package=IPDfromKM](https://CRAN.R-project.org/package=IPDfromKM).

Competing interests
The authors declare that they have no competing interests.

Funding
JJL’s research was supported in part by the grants CA016672 and CA221703 from the National Cancer Institute and RP150519 and RP160668 from the Cancer Prevention and Research Institute of Texas.

Author’s contributions
All authors contributed to the conception and design of the study. NL and YZ developed the R package and the web-based Shiny application. All authors participated in manuscript writing. JJL gave final approval of the manuscript submitted for publication.

Acknowledgements
The authors thank Ryan Sun for his constructive comments in the earlier version of the manuscript and Jessica Swann for her editorial assistance.

References
1. Green, S., Higgins, J.: Cochrane Handbook for Systematic Reviews of Interventions. Version (2005)
2. Earle, C.C., Wells, G.A.: An assessment of methods to combine published survival curves. Medical Decision Making 20(1), 104–111 (2000)
3. Lueza, B., Rotolo, F., Bonastre, J., Pignon, J.-P., Michiels, S.: Bias and precision of methods for estimating the difference in restricted mean survival time from an individual patient data meta-analysis. BMC medical research methodology 16(1), 37 (2016)
4. Tudur, C., Williamson, P.R., Khan, S., Best, L.Y.: The value of the aggregate data approach in meta-analysis with time-to-event outcomes. Journal of the Royal Statistical Society: Series A (Statistics in Society) 164(2), 357–370 (2001)
5. Guyot, P., Ades, A., Ouwens, M.J., Welton, N.J.: Enhanced secondary analysis of survival data: reconstructing the data from published kaplan-meier survival curves. BMJ medical research methodology 12(1), 9 (2012)
6. Satagopan, J.M., Iasonos, A., Kanik, J.G.: A reconstructed melanoma data set for evaluating differential treatment benefit according to biomarker subgroups. Data in brief 12, 667–675 (2017)
7. Wei, Y., Royston, P.: Reconstructing time-to-event data from published kaplan–meier curves. The Stata Journal 17(4), 786–802 (2017)
8. Sun, R.: R package to Reconstruct Individual-level Data From KM Plots Published in Academic Journals. GitHub (2018)
9. Kaplan, E.L., Meier, P.: Nonparametric estimation from incomplete observations. Journal of the American statistical association 53(282), 457–481 (1958)
10. Bonner, J.A., Harari, P.M., Giralt, J., Azarnia, N., Shin, D.M., Cohen, R.B., Jones, C.U., Sur, R., Raben, D., Jassim, J., et al.: Radiotherapy plus cetuximab for squamous-cell carcinoma of the head and neck. New England Journal of Medicine 354(6), 567–578 (2006)
11. Fehrenbacher, L., Spira, A., Ballinger, A., Levanski, C., et al.: Atezolizumab versus docetaxel for patients with previously treated non-small-cell lung cancer (poplar): a multicentre, open-label, phase 2 randomised controlled trial. The Lancet 387(10030), 1837–1846 (2016)
12. Kim, J.S., Yum, B.-J.: Selection between weibull and lognormal distributions: A comparative simulation study. Computational Statistics & Data Analysis 53(2), 477–485 (2008)
Figures

Figure 1: The flowchart of IPD reconstruction from published K-M curves.

Figure 2: IPD reconstruction of two treatments using the Shiny application.
(a) Estimated number of patients at risk versus reported number at risk under different censoring rates, using data extracted using IPDfromKM package, ScanIt, or DigitizeIt. Results were reported when the number of patients at risk was reported for 20 intervals, 6 intervals, or not reported at all (no risk information).

(b) Mean absolute error of the estimation for the number of patients at risk under different censoring rates, when the number of patients at risk was reported for 6 or 20 intervals. The error is determined by (estimated value − true value). Data were extracted using IPDfromKM package, ScanIt, or DigitizeIt.

Figure 3: Accuracy assessment in terms of the number of patients at risk.
(a) Error of estimates on median survival time and one-year survival probability, based on data extracted using IPDfromKM package, ScanIt, or DigitizeIt. Results are shown when the number of patients at risk are reported for 20 intervals, 6 intervals, or not reported at all (no risk information).

(b) Log hazard ratio based on data extracted using IPDfromKM package, ScanIt, or DigitizeIt. Results are reported for cases where the number of patients at risk are reported for 20 intervals and 6 intervals.

Figure 4: Survival analysis on the reconstructed IPD and the true data.
Tables

| Function   | Description                                                                 | Object returned                                                                 |
|------------|----------------------------------------------------------------------------|---------------------------------------------------------------------------------|
| getpoints  | Extract raw data coordinates from published K-M curves                     | A data frame containing the x- and y-coordinates of the K-M curve of interest.  |
| preprocess | Preprocess the read-in data coordinates                                    | A list including cleaned data ready for reconstruction and a "riskmat" table displaying the index of read-in points within each time interval. |
| getIPD     | Estimate the IPD                                                            | A list including the reconstructed IPD.                                          |
| survreport | Perform the survival analysis on reconstructed IPD.                        | K-M curve, cumulative hazard, times for targeted survival probabilities.         |
| plot       | Plot the object returned by getIPD                                          | K-M curves and number at risk for both reconstructed IPD and read-in data.       |
| summary    | Summarize objects returned by getIPD() and survreport()                    | Descriptive results for accuracy assessment and survival analysis on reconstructed IPD. |

Table 1: Overview of the user visible functions in IPDfromKM. Please consult the documentation (e.g., help("preprocess")) for function arguments and detailed return types.

| Group | Arm          | n   | Median OS | Hazard Ratio |
|-------|--------------|-----|-----------|--------------|
|       |              |     | Report    | R | D | S | Report    | R | D | S |
| 1     | Atezolizumab | 24  | 15.5      | 15.5 | 15.5 | 15.5 | 0.49 | 0.48 | 0.45 | 0.46 |
|       | Docetaxel    | 23  | 11.1      | 11.1 | 11.1 | 11.1 |       |      |      |      |
| 2     | Atezolizumab | 50  | 15.1      | 15.3 | 15.3 | 15.3 | 0.54 | 0.54 | 0.56 | 0.53 |
|       | Docetaxel    | 55  | 7.4       | 7.6  | 7.4  | 8.1  |       |      |      |      |
| 3     | Atezolizumab | 93  | 15.5      | 15.3 | 15.5 | 15.7 | 0.59 | 0.59 | 0.58 | 0.59 |
|       | Docetaxel    | 102 | 9.2       | 9.3  | 9.2  | 9.6  |       |      |      |      |
| 4     | Atezolizumab | 51  | 9.7       | 9.7  | 9.7  | 9.5  | 1.04 | 1.06 | 0.99 | 1.03 |
|       | Docetaxel    | 41  | 9.7       | 9.7  | 9.7  | 9.8  |       |      |      |      |
| 5     | Atezolizumab | 144 | 12.6      | 13.3 | 13.4 | 12.3 | 0.73 | 0.72 | 0.70 | 0.72 |
|       | Docetaxel    | 143 | 9.7       | 9.7  | 9.7  | 9.8  |       |      |      |      |

Table 2: Estimates of median overall survival (OS) and hazard ratio using the modified-iKM algorithm based on data extracted using different software (R: R package IPDfromKM; D: DigitizeIt; S: ScanIt), in comparison to published results in the POLAR trial. Group 1: TC3 or IC3; 2: TC2/3 or IC2/3; 3: TC1/2/3 or IC1/2/3; 4: TC0 or IC0; 5: All patients. The value of n refers to sample size.
Additional Files
0.1 Modified-iKM algorithm in details

The IPD reconstruction is carried out using the modified-iKM algorithm, which is based on the K-M estimation method [9] and improves upon the iKM algorithm [5]. Details regarding the iKM estimation method and the modified-iKM algorithm are provided below.

The KM estimator

The KM estimator, first proposed by Kaplan and Meier [9] is a non-parametric estimator of the survival function. It is determined by the product over the failure times of the conditional probabilities of surviving to the next failure time. Specifically, suppose there are Q distinct failure times. Let \( t_q \) denote a time where at least an event (e.g., a patient dies) is observed, \( n_q \) is the number of subjects at risk at time \( t_q \), and \( d_q \) is the number of subjects who experience the event at that time, \( q = 1, \ldots, Q \). The KM estimator \( \hat{S}_{t_q} \) is formally defined as

\[
\hat{S}_{t_q} = \prod_{j=1}^{q} \frac{1 - d_j}{n_j} = \hat{S}_{t_{q-1}} \cdot \frac{1 - d_j}{n_j}, \quad q = 1, \ldots, Q.
\]

The number of patients at risk \( n_{q+1} \) can be determined by the number of patients at risk at time \( t_q \), minus the corresponding number of patient experienced events and number of patients censored.

\[
n_{q+1} = n_q - d_q - c_q.
\]

The modified-iKM algorithm

Before initializing the iterative algorithm, we first sort extracted coordinates by time and make monotonicity adjustment on survival probabilities. In addition, to improve estimation accuracy, we propose a step control procedure while calculating the number of events at the coordinate \( k \) (denoted as \( d_k \)). Usually, there will be multiple points at the same time, and thus multiple survival probabilities are available at this time. In a K-M curve, this is reflected by a drop in the survival probability at a time point. We referred to such a time point as a jumping step. Suppose there are \( s \) consecutive points read out at the jumping step from the target K-M curve, and denote them as \( j, j + 1, \ldots, j + s - 1 \). The original iKM method requires you to have all the values for \( d_j, d_{j+1}, \ldots, d_{j+s-1} \) and add them together to get the estimated total number of events that happened at the jumping step. However, this could end up in under-estimation of the number at risk due to rounding. That is, \( d_j, d_{j+1}, \ldots, d_{j+s-1} \) can be too small to round off to zero. When too many jumping points exist, the number at risk is under-estimated. As a simple illustrative example, suppose \( s = 10 \), if all the values for \( d_j, d_{j+1}, \ldots, d_{j+s-1} \) are within \([0.1, 0.5)\). After rounding these values separately and adding them together, the estimated number at risk is zero, but in reality, it lies between 1 and 5. To avoid such a problem, we recommend controlling the steps to ensure that there are only two points remaining in each vertical segment on the K-M curve: one at the beginning and the other one at the end of the segment. The corresponding survival rates at these two points are simply \( \hat{S}_{\text{last}(k)} \) and \( \hat{S}_k \). By doing this, the number of events at this step needs to be calculated and rounded only one time, thus alleviating the under-estimation problem.

To start the iterative estimation process, we first divide the preprocessed coordinates from the K-M curve into \( I \) intervals. Denote the number of at risk at the beginning of the intervals as \( n_{\text{risk}1}, n_{\text{risk}2}, \ldots, n_{\text{risk}I} \) and denote the time at which the number of patients at risk is provided as \( t_{\text{risk}1}, t_{\text{risk}2}, \ldots, t_{\text{risk}I} \). For each interval, we denote the index of the first point as \( \text{lower}_i \) and of the last point as \( \text{upper}_i \). The iterative estimation process proceeds as follows:

1. Initialize the total number of patients censored at interval \( i \) (\( n_{\text{censor}_i} \)) using the difference between the reported number at risk at the beginning of interval \( i + 1 \) and the number at risk in this interval if no censoring occurs (\( n_{\text{risk}i+1} \) in interval \( i \)). The value of \( n_{\text{risk}i+1} \) is given by \( n_{\text{risk}i+1} = n_{\text{risk}i} + \hat{S}_{\text{lower}_i+1}/\hat{S}_{\text{lower}_i} \), rounded to the nearest integer, where \( \hat{S}_{\text{lower}_i+1}/\hat{S}_{\text{lower}_i} \) is the probability of survival at the beginning of interval \( i + 1 \) conditional on being alive at the beginning of interval \( i \). Thus we have

\[
n_{\text{censor}_i} = n_{\text{risk}i} \cdot \hat{S}_{\text{lower}_i+1}/\hat{S}_{\text{lower}_i} - n_{\text{risk}i+1}.
\]

2. Determine the number of patients censored between the extracted coordinates \( k \) and \( k + 1 \) (denoted as \( n_{\text{censor}_k} \)). Assuming a constant censoring rate, it is straightforward to determine the censoring time by distributing the number of censored patients evenly over the interval \( [T_k, T_{k+1}] \):

\[
n_{\text{censor}_k} = \sum_{m=1}^{n_{\text{censor}_k}} I(\text{te censor}_m \in [T_k, T_{k+1}]),
\]

where \( I(\text{te censor}_m \in [T_k, T_{k+1}]) \) is an indicator function, which returns 1 if the censoring time \( \text{te censor}_m \) lies within the interval \( [T_k, T_{k+1}] \).
3 Determine the number of patients at risk for the coordinate \( k + 1 \) (i.e., \( \hat{n}_{k+1} \)) as

\[
\hat{n}_{k+1} = \hat{n}_k - \hat{d}_k - \hat{ncensor}_k, \quad k = \text{lower}_i, \cdots, \text{upper}_i 
\]  

(5)

according to equation (2), where \( \hat{d}_k \) is determined by

\[
\hat{d}_k = \left( 1 - \frac{S_k}{\hat{S}_{KM_{last(k)}}} \right). 
\]

Round to the nearest integer, based on a rearrangement of equation (1). Here \( \hat{S}_{KM_{last(k)}} \), instead of \( \hat{S}_{KM_{k-1}} \), is used because there may not be an event at the extracted coordinate \( k - 1 \). Thus, \( \hat{S}_{KM_{last(k)}} \) is the survival probability of the last point before time \( T_k \).

4 Check if the estimated number at risk at the start of the next interval \( i + 1 \) (i.e., \( \hat{nrisk}_{i+1} \), which is just \( \hat{n}_{\text{upper}_i} + 1 \)) is equal to the reported value \( nrisk_{i+1} \).

- If \( \hat{nrisk}_{i+1} = nrisk_{i+1} \), move to step 5.
- Otherwise, set \( \hat{ncensor}_i := \hat{ncensor}_i + \hat{n}_{\text{upper}_i} + 1 - nrisk_{i+1} \).

5 Repeat the iteration in steps 1-4, as long as one of the following two conditions holds:

\[
(\hat{n}_{\text{upper}_i} + 1 > nrisk_{i+1}) \quad \text{and} \quad (\hat{ncensor}_i < nrisk_i - nrisk_{i+1}), 
\]  

(6)

or

\[
(\hat{n}_{\text{upper}_i} + 1 < nrisk_{i+1}) \quad \text{and} \quad (\hat{ncensor}_i > 0). 
\]  

(7)

The condition in (6) shows that the number of patients at risk for interval \( i \) is greater than the reported value, so the iteration will continue to increase the number of censored patients for this interval. On the other hand, condition (7) shows that the number of patients at risk at the end of interval \( i \) is less than the reported value, thus the iteration will continue to decrease the estimation of the number of censored patients for the interval. Note that, in condition (6), the upper bound on the number of censored patients (\( \hat{ncensor}_i \)) is added (i.e., \( \hat{ncensor}_i < nrisk_i - nrisk_{i+1} \)) to ensure that the number of censored patients for each interval lies within the range of \([0, nrisk_i - nrisk_{i+1}] \). This proper boundary condition is needed to avoid having an infinite number of iterations, which was not considered in the original iKM method.

6 Check if the iteration reaches the last interval.

- If \( i + 1 \) is not the last interval, i.e., \( i + 1 \neq I \), repeat steps 1-4.
- Otherwise, adjust the initial guess of the censored patients in the last interval as

\[
\hat{ncensor}_I = \min(\text{Average censoring rate} \times \text{length of the last interval}, \quad nrisk_i - \text{endpts} - (\text{tot.events} - \sum_{k=1}^{\text{upper}_I - 1} \hat{d}_k)), 
\]  

(8)

where \( \text{endpts} \) is the last reported number of patients at risk. The adjustment is made because the number at risk for the last interval (\( nrisk_I \)) is typically not available, and thus the equation in (3) cannot be determined directly.
0.2 Figures returned for accuracy assessment or secondary analysis

The IPDfromKM package provides `plot()` function to visualize the accuracy of IPD reconstruction (Figure A.1), in addition to formal test. It also provides `survreport()` function to conduct secondary survival analysis and returns figures for the new analysis (Figure A.2).

Figure A.1: Results of accuracy assessment results using the `plot()` function.

Figure A.2: Graphs reported by the `survreport()` function.
0.3 Additional output of using the app to reconstruct IPD for two treatment groups

Figure A.3: Accuracy assessment using the IPDfromKM Shiny application.
Figure A.4: Secondary analysis using the IPDfromKM Shiny application.
(a) Available formats for report.

(b) A snapshot of the PDF report.

Figure A.5: Download report using the IPDfromKM Shiny application.
0.4 Original K-M curves in the POLAR trial

Figure A.6 shows the K-M curves and estimation of median survival and hazard ratio for the subgroups in the POLAR trial.