

Market Competition Analysis of Drug Clinical Development Using SAS

WEI CHI

PROGRAMMING MANAGER, JIANGSU HENGRUI MEDICINE CO., LTD

Disclaimer



The views and opinions expressed in the following PowerPoint slides are those of the individual presenter and should not be attributed to PhUSE, its directors, officers, employees, volunteers, members, chapters, councils, Special Interest Area Communities or affiliates, or any organization with which the presenter is employed or affiliated.

Analysis Objective

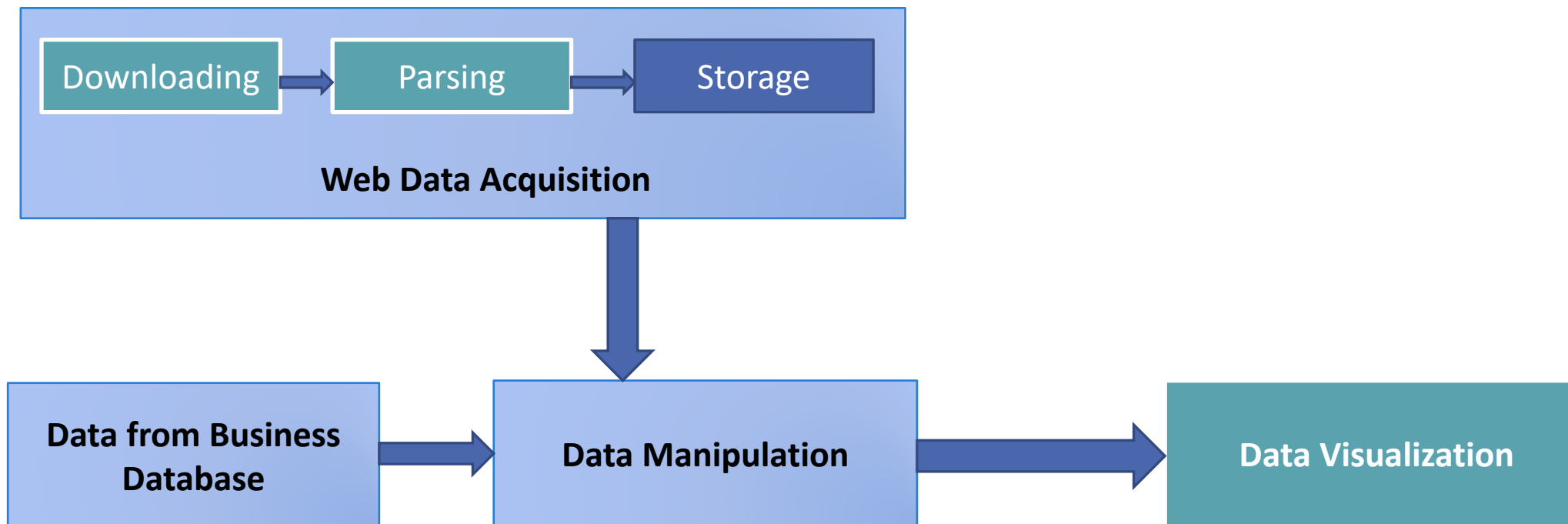
Given a target and indication, what is the competitive landscape?

Which clinical trials for this indication are in process? What their latest progress are?

In best cases, when these compound could be approved?

Given a company, how about its product pipeline?

Analysis Flow



clinicaltrials.gov

Crawler friendly. Provide links to traverse all studies. <https://clinicaltrials.gov/ct2/crawl>

There are more than 280,000 studies, which can be searched by keyword, country, etc..

Each study has an unique ID - NCTID and is displayed in a single page.

All archived studies are available to download as zipped xml files manually.

Most pages can be found in HTML or XML format.

- <https://www.clinicaltrials.gov/ct2/show/NCT03483012>
- <https://www.clinicaltrials.gov/ct2/show/NCT03483012?DisplayXML=True>
- <https://www.clinicaltrials.gov/ct2/results?cond=breast+cancer>
- <https://www.clinicaltrials.gov/ct2/results?cond=breast+cancer&DisplayXML=True>

But XML are not advised to crawl according to the website robots.txt

- <https://www.clinicaltrials.gov/robots.txt>

The robots.txt of clinicaltrials.gov



```
← → ↻ 🔒 https://clinicaltrials.gov/robots.txt

# robots.txt for http://clinicaltrials.gov
# $Revision: 76348 $
#

User-Agent: *
Disallow: /cs/                # obsolete
Disallow: /c/                 # obsolete
## Keep /ct/                  # it is obsolete, but we handle it well
Disallow: /cta/               # same info available elsewhere
Disallow: /beta/              # don't want persistent links to beta site

Disallow: /ct2/archive/       # leaving this site to archive page
Disallow: /ct2/bye/           # leaving this site to 3rd party page
Disallow: /ct2/helpdesk/      # helpdesk form
Disallow: /ct2/results/rss.xml # rss feed of current search results
Disallow: /*?displayxml=true   # don't index xml files
Disallow: /*?resultsxml=true   # don't index xml files
# Don't index archive site, don't want it found (and ranked highly) by the
# likes of Google since we really want people to come to the main site.
Disallow: /archive/           # don't index archive site

# Some docs say must be a whole number of seconds.
# Crawlers differ whether this means delay-between-calls or whether it is max-requests-in-window.
# Google ignore this value, but supposedly honored by yandex and bing.
Crawl-delay: 1

# crawlers are advised to avoid our browse hierarchy because
# it finds all studies many times, with slightly different URL's
# and term highlighting. Much better if crawlers find the studies
# using the crawler link: /ct2/crawl
```

chinadrugtrials.org.cn

There are about 6,800 of studies.

Studies can be searched by keyword. But the searching engine is less functional comparing to clinicaltrials.gov.

If the keyword is null, all studies are displayed in the returned list.

The URL for search list and single study keeps constant.

- <http://www.chinadrugtrials.org.cn/eap/clinicaltrials.searchlist>
- <http://www.chinadrugtrials.org.cn/eap/clinicaltrials.searchlistdetail>

Robots.txt does no exist on this site.

Each study has an internal hash value in the HTML source code.

Click each study in the search list to post the form with its hash value.

Check source code using Browser Web Inspector

```

▼<td align="center">
  <a href="javascript:void(0)" onclick="getDetail(this.id)" id="
  "2bdbf3ac142a4f0fb49259980dbabcdf" name="1">
    CTR20181821
  </a>
</td>

function getDetail(id){
  document.getElementById("ckm_index").value =
document.getElementById(id).getAttribute("name");
  document.getElementById("ckm_id").value = id;
  $("#searchfrm").attr("action","http://www.chinadrugtrials.org.cn/eap/clinicaltrial
s.searchlistdetail");
  document.getElementById("searchfrm").submit();
}

▼<div class="register_mainB">
  ▼<form id="searchfrm" method="post" action="http://www.chinadrugtrials.org.cn/eap/
clinicaltrials.searchlist">
    <input name="ckm_id" id="ckm_id" type="hidden" value>
    <input type="hidden" id="ckm_index" name="ckm_index" value>
    <!-- 登记号排序方式 -->
    <input type="hidden" id="sort" name="sort" value="desc">

```

SAS Web Page Downloading Toolkit

	Filename URL	Proc HTTP (since SAS 9.2)	Proc DS2 HTTP Package (since SAS 9.4M2)
Syntax	filename <i>fileref</i> URL " <i>url</i> " headers= <i>fileref</i> ; data dsn; infile <i>fileref</i> ; ...	proc http url="..." method=" <i>url</i> " out= <i>fileref</i> headerin= <i>fileref</i> headerout= <i>fileref</i> ; run;	Declare package http h(); h.createGetMethod(url); h.executeMethod(); h.getStatusCode(); h.getBodyAsString(); ...
Feature	Simple syntax Only supports GET method Not very easy to get status code	Supports all HTTP 1.1 methods(since SAS 9.4M3) Easy to get status code Supports customizing request header	OOP syntax style Supports GET, POST and HEAD method Easy to get status code Save the response as a string instead of file Supports customizing request header

Example1: Pull trial list using keyword from clinicaltrials.gov



```
%let cond = PD-1;
filename out "physical file";
filename head temp;
%do i = 1 %to &times;
  proc http url="https://www.clinicaltrials.gov/ct2/results?cond=&cond%str(&)count=10000"
    method="get" out=out headerout=head headerout_overwrite; run;
  %let StatusCode = %GetHttpStatusCode(head);
  %if &StatusCode = 200 %then %return;
  %else if ^(&StatusCode >= 500 and StatusCode < 600) %then %do;
    %put ERROR: &StatusCode; %return;
  %end;
  %else %syscall sleep(1000);
%end;
```

Example2: Pull trial list using keyword from chinadrugtrials.org.cn



Step 1 Get the total page number

```
%let keywords = pd-1;  
%let pagesize = 40;  
%let params = keywords=&keywords%str(&)pagesize=&pagesize;  
proc http url="http://www.chinadrugtrials.org.cn/eap/clinicaltrials.searchlist?%superq(params)"  
    method="get" out=out headerout=head headerout_overwrite;  
run;
```

Example2: Pull trial list using keyword from chinadrugtrials.org.cn



Step 2 Pull trial Id by each page

```
%let keywords = pd-1;
%let pagesize = 40;
%do i = 1 %to &npage;
%let params = keywords=&keywords%str(&)pagesize=&pagesize%str(&)currentpage=&i;
proc http url="http://www.chinadrugtrials.org.cn/eap/clinicaltrials.searchlist?%superq(params)"
    method="get" out=out headerout=head headerout_overwrite;
run;
...
%end;
```


Example3: Pull single trial info from chinadrugtrials.org.cn



```
%let ckm_id = eae508bb5c624eed92e9a7813ccdedf1;
```

```
%let ckm_index = 1;
```

```
%let currentpage = 1 ;
```

```
proc http
```

```
url="http://www.chinadrugtrials.org.cn/eap/clinicaltrials.searchlistdetail"
```

```
method="post"
```

```
in="ckm_id=&ckm_id%str(&)ckm_index=&ckm_index%str(&)currentpage=&currentpage"
```

```
out=resp;
```

```
run;
```

Parsing Downloaded Data using SAS

HTML	XML	JSON
PRX functions Proc DS2 PCRXFIND Package (since 9.4M5)	XML engine XMLV2 engine (since SAS 9.2) SAS XML mapper	JSON engine (since SAS 9.4M4) Proc DS2 JSON Package (since SAS 9.4M3)
<pre>declare package pcrxfind regex(); regex.parse("regexpr"); regex.match(); regex.getGroup(); regex.getMatch(); ...</pre>	<pre>libname libref xml "XML- document-path" xmlmap=<i>fileref</i>; libname libref xmlv2 "XML- document-path" xmlmap=<i>fileref</i>;</pre>	<pre>libname libref json "JSON- document-path";</pre>

Example4: Pull study start date from HTML using pcrxfind package

Actual Study Start Date ⓘ: August 15, 2017

Estimated Primary Completion Date ⓘ: May 31, 2021

Estimated Study Completion Date ⓘ: May 31, 2021

```
data out / overwrite=yes;
  declare package http h();
  declare package pcrxfind regex1();
  declare int status rc;
  declare varchar(50000) character set utf8 body;
  declare varchar(50) StartDate;
  keep StartDate;
  method init();
    regex1.parse('/Study Start Date.*?(\\w{3,} (\\d{1,2}, )?\\d{4}).*?Primary Completion Date/s');
  end;
  method run();
    set triallist;
    h.CreateGetMethod(url);
    h.ExecuteMethod();
    status = h.GetStatusCode();
    if status = 200 then do;
      h.getResponseBodyAsString(body, rc);
      if regex1.match(body) > 0 then regex1.GetGroup(StartDate,1);
    end;
  end;
enddata;
```

Encoding

The encoding of web pages from both sources are UTF-8.

When reading external files, SAS always convert them into the encoding of current session.

Option encoding= "utf-8" should be added, when the current encoding is not Unicode.

The SAS environment with single-byte encoding cannot handle this converting.

The best solution is using SAS supporting Unicode.

When parsing XML files, any encoding convert error will crash the program.

The following method could be used to remove problematic characters.

```
Data _null_;  
  infile fileref encoding="utf-8";  
  file fileref encoding= "utf-8";  
run;
```

Parallelism – Multiple Processes

Multiple processes could be used to accelerate the execution using Systask and waitfor statements.

Main program sends messages to sub programs by system macro variable SYSPARM.

Sub programs return messages to main program by macro variable specified by status=.

```
systask command "%sysget(SASROOT)\sas.exe" -noterminal -nosplash -sysparm "raw.lst1"  
-sysin "sub.sas" -log "sub1.log" -print "sub1.lst" status=rc_task1 taskname=task1;  
  
systask command "%sysget(SASROOT)\sas.exe" -noterminal -nosplash -sysparm "raw.lst2"  
-sysin "sub.sas" -log "sub2.log" -print "sub2.lst" status=rc_task2 taskname=task2;  
  
systask command "%sysget(SASROOT)\sas.exe" -noterminal -nosplash -sysparm "raw.lst3"  
-sysin "sub.sas" -log "sub3.log" -print "sub3.lst" status=rc_task3 taskname=task3;  
  
waitfor _all_ task1 task2 task3;
```

Parallelism – Multiple Threads

Multiple threads could be created to accelerate the execution using Proc DS2 thread.

Only improve performance when the process is CPU bound instead of IO bound and the data volume is large.

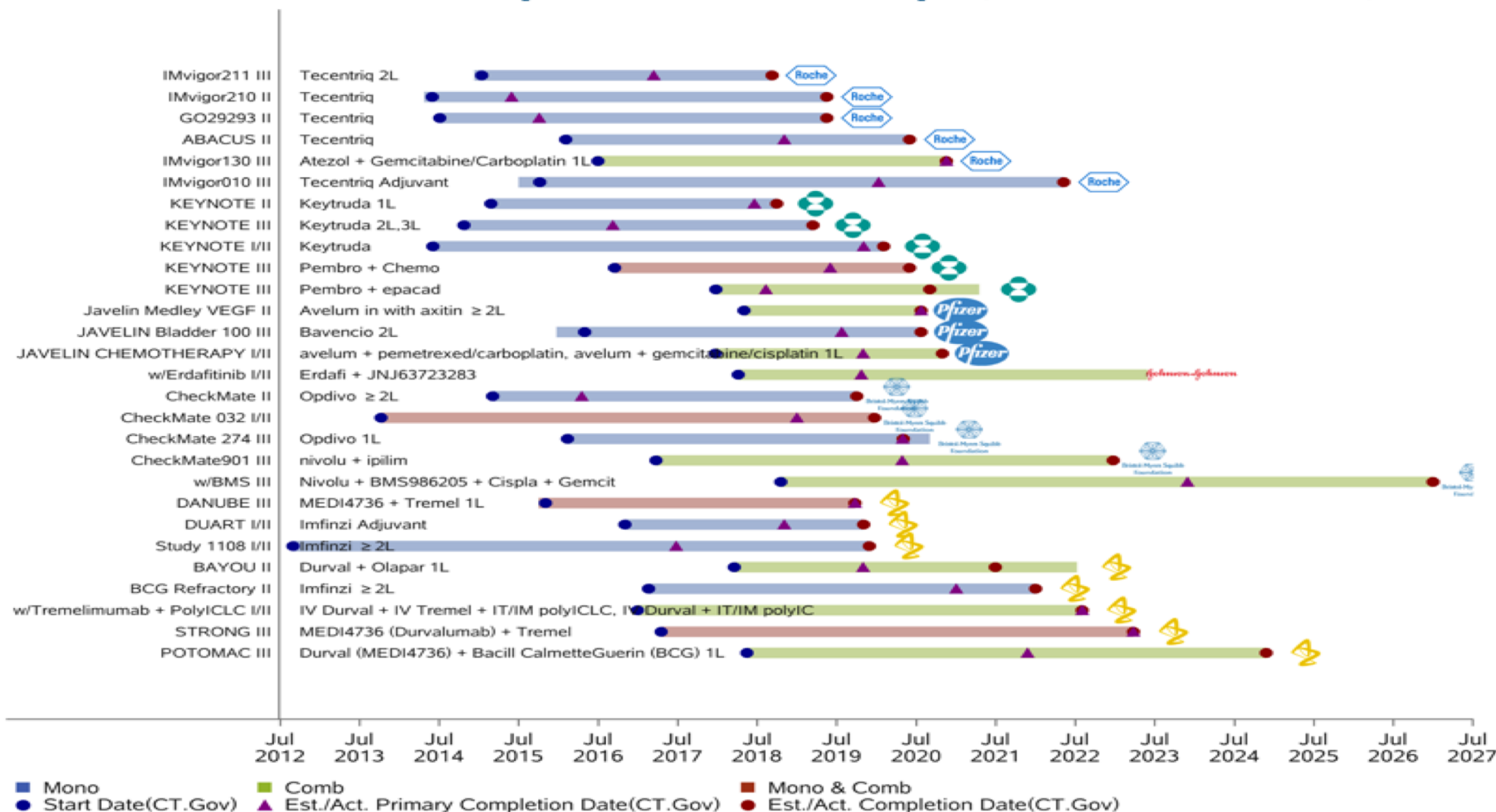
So data downloading and parsing should be split into two steps when using multiple threads.

```
proc ds2;  
  thread crawler / overwrite=yes;  
    ...  
  endthread;  
  
  data out / overwrite=yes;  
    declare thread crawler c;  
    method run();  
      set from c threads=3;  
    end;  
  enddata;  
run;  
quit;
```

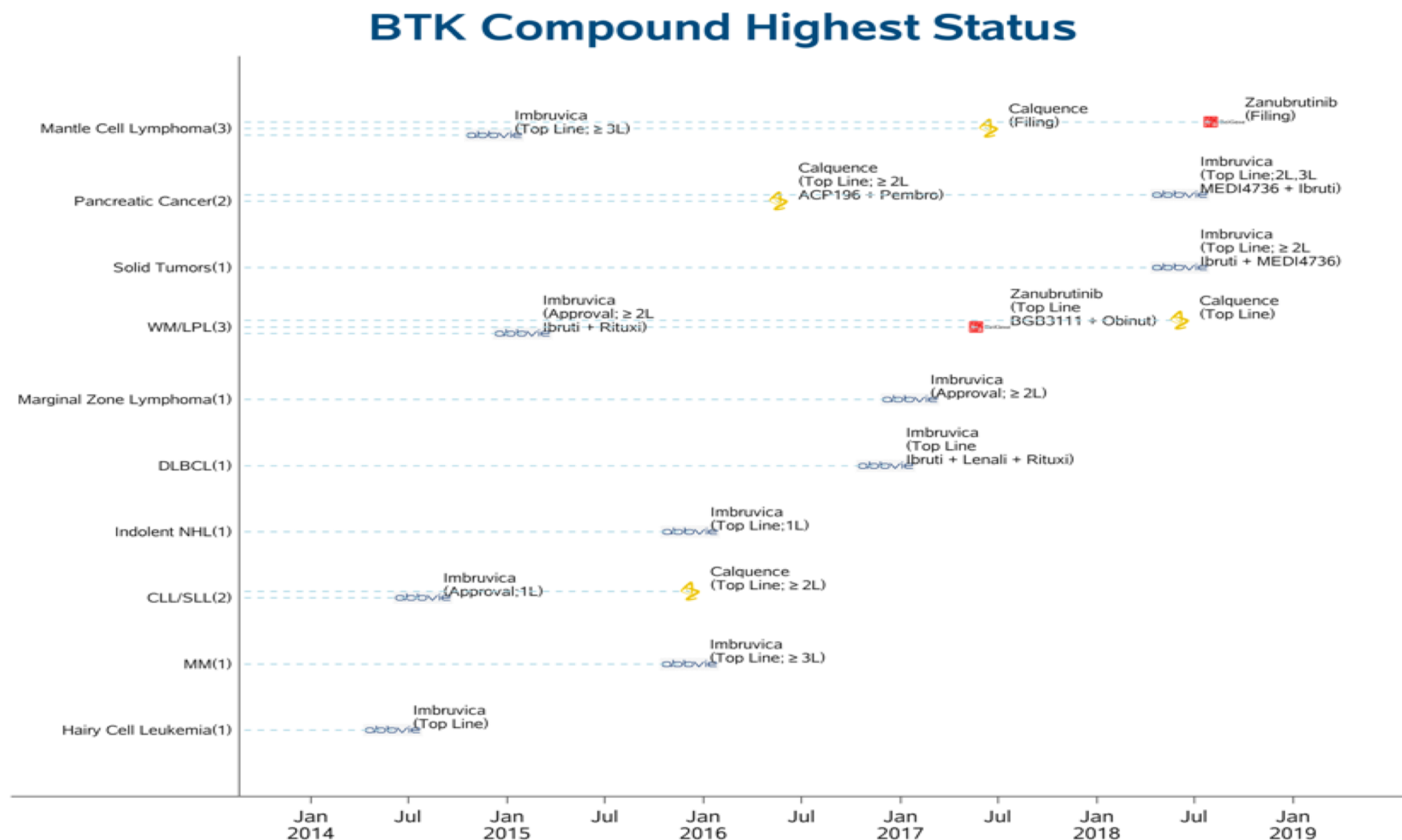
Competitive Landscape Graph by Indication



PD1 Studies Competitive Landscape(Bladder Cancer)

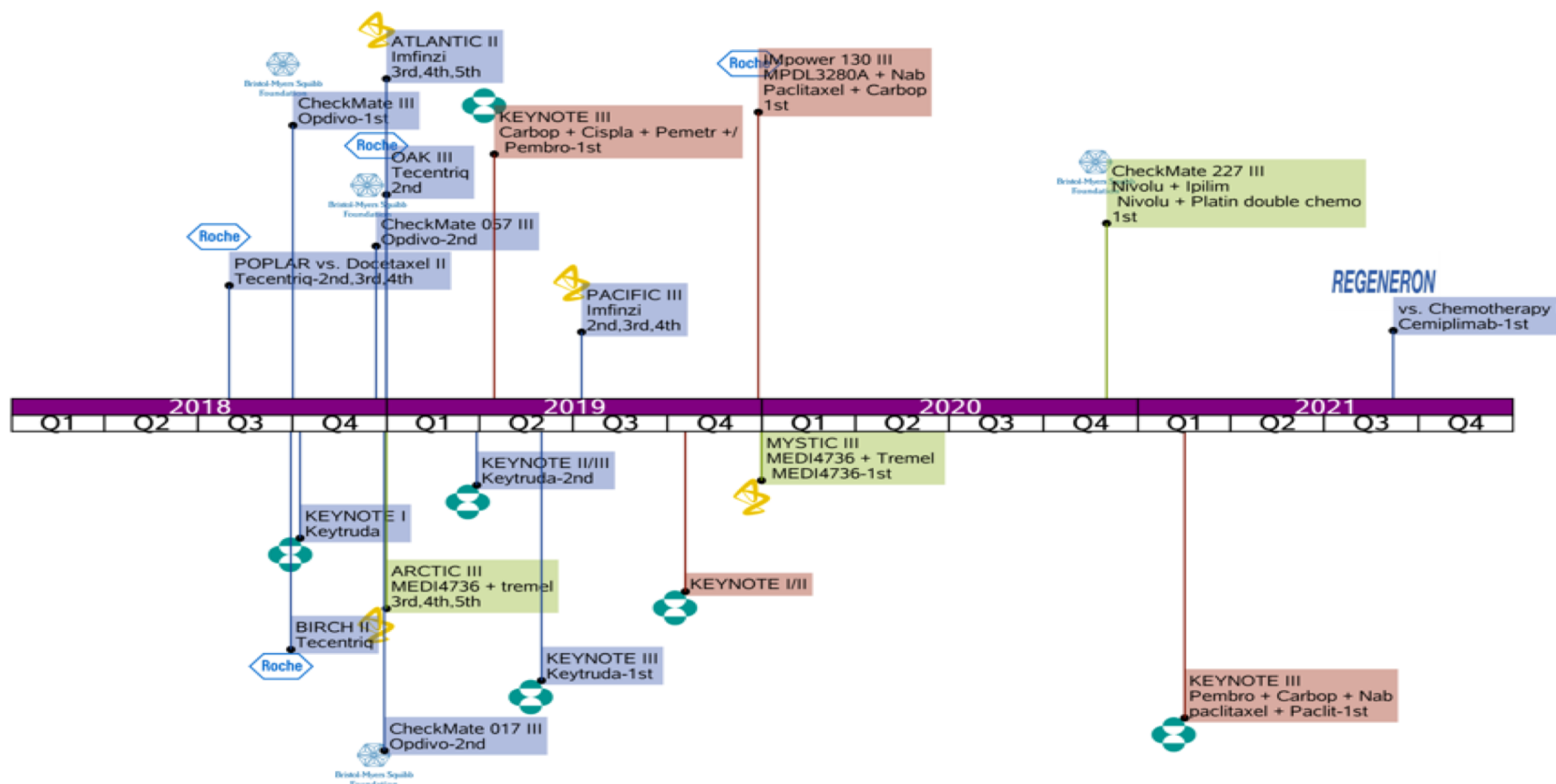


Compound Highest Status Graph



Studies Completion/Approval Projection Graph

PD1 NSCLC Best Case Approval Projections



Customized Symbol Image

Proc sgplot	GTL	SAS/GRAPH
Since SAS 9.4M1 <pre>symbolimage name=n1 image="img1"; symbolimage name=n2 image="img2"; ...</pre> scatterplot ... / group=LeadSymbol attrid=attrmapid;	Since SAS 9.4M1 <pre>symbolimage name=n1 image="img1"; symbolimage name=n2 image="img2"; ... discreteattrmap name="symbols" ; value "n1" / markerattrs=(symbol=n1); value "n2" / markerattrs=(symbol=n1); ... enddiscreteattrmap;</pre> discreteattrvar attrvar=groupmarkers var=LeadSymbol attrmap="symbols"; scatterplot ... / group=groupmarkers;	Use "image" function in annotation dataset.

Data Label Placement Algorithms

Greedy Algorithm

- ods graphics / labelplacement=greedy;

Simulation Annealing Algorithm

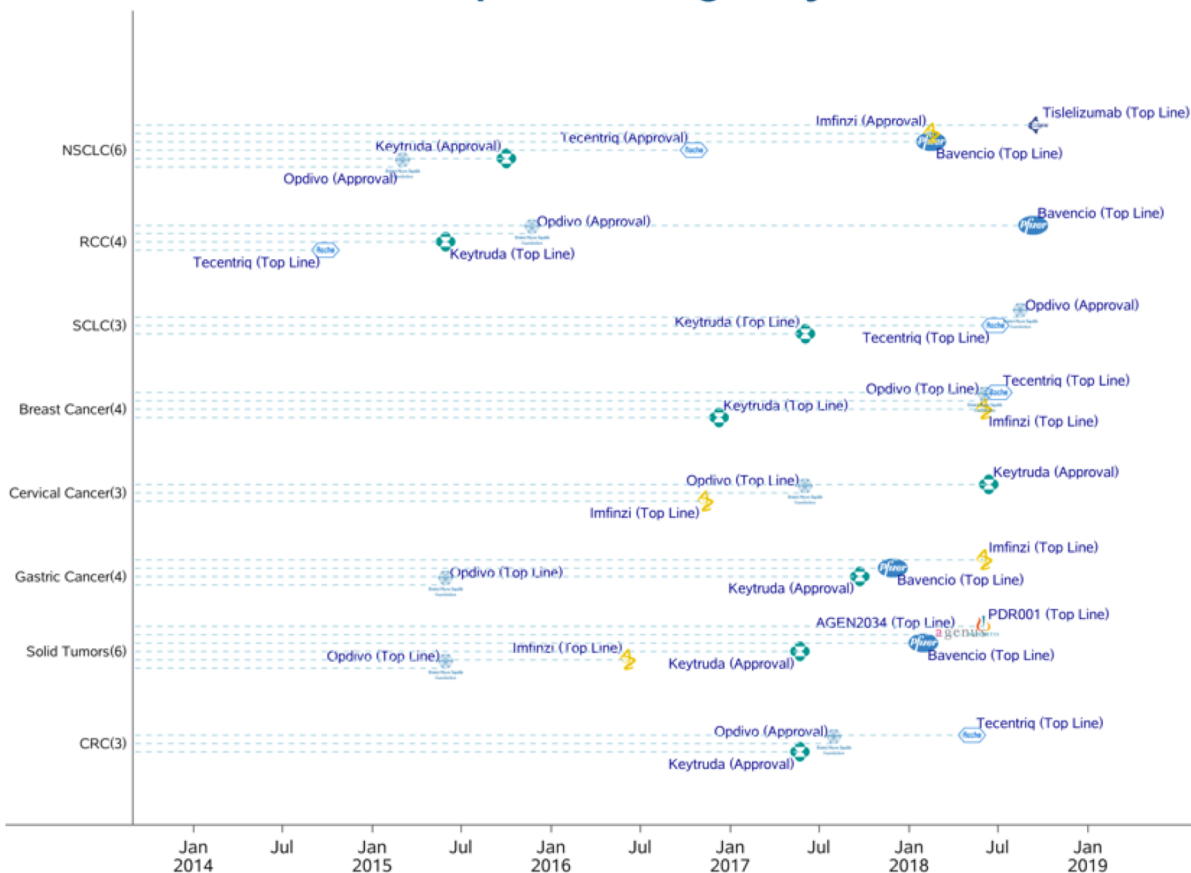
- ods graphics / labelplacement=sa;

%labelit (Warren F. Kuhfeld, Advanced ODS Graphics Examples)

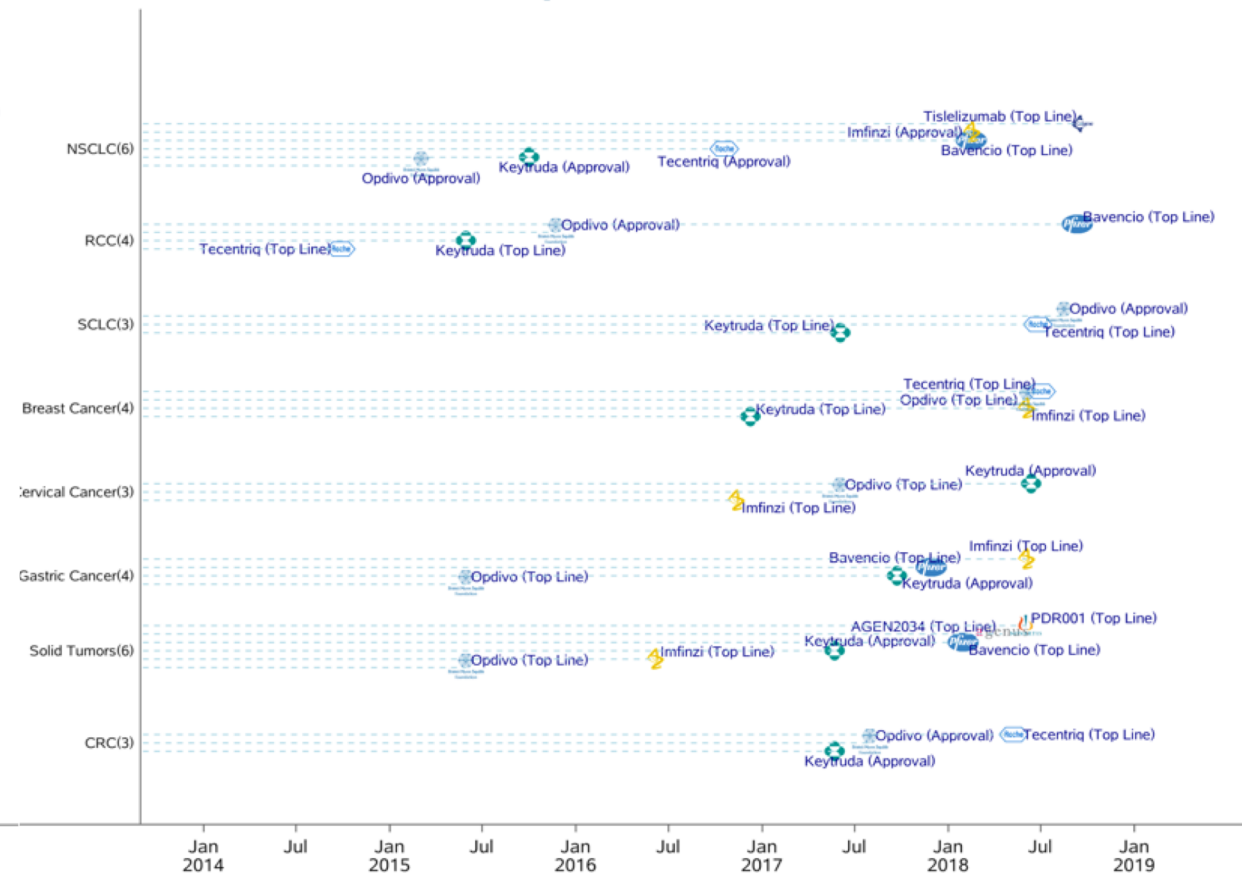
- PROC PLOT options control the range of label movement and specify preferences for label placement
- the graph is drawn by PROC SGPLOT
- the labels are placed by using SG annotation

Data Label Placement Algorithms

labelplacement=greedy



labelplacement=sa



Alternative Open-Source Tools

	Python	R	Others
Web Scraping	Requests BeautifulSoup Scrapy	rvest httpr	Jsoup (Java)
Data Manipulation	Pandas	Tidyverse (dplyr, tidyr, tibble, stringr ...)	
Data Visualization	Matplotlib	ggplot2 ggvis	D3.js (Javascript)

Examples of parsing HTML using Python and R

```
import re
import requests
from bs4 import BeautifulSoup

url = "https://www.clinicaltrials.gov/ct2/results?cond=PD1&count=10000"

r = requests.get(url, timeout = 10)
soup = BeautifulSoup(r.text, "lxml")
results = soup.find_all("a", attrs={'href':re.compile(r"/ct2/show/NCT\d{6}")})
url_list = [each['href'] for each in results]
```

```
library(tidyverse)
library(rvest)

search_list <-
  read_html("https://www.clinicaltrials.gov/ct2/results?cond=PD1&count=10000")

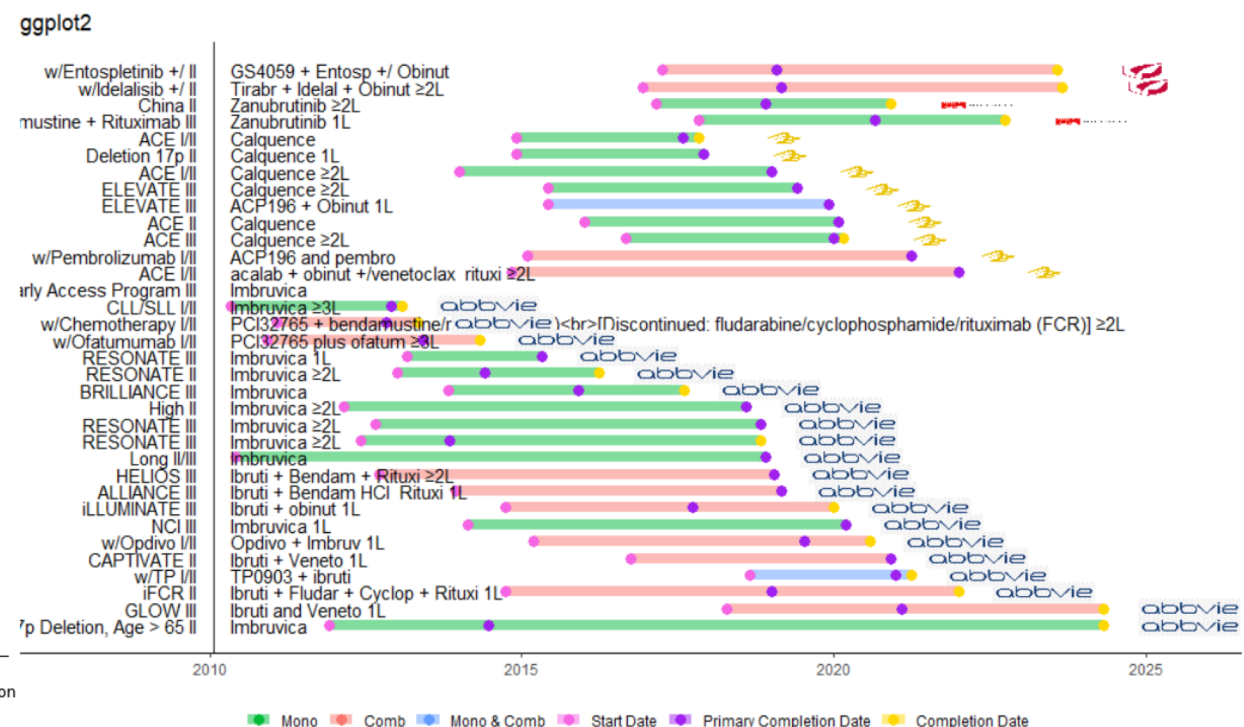
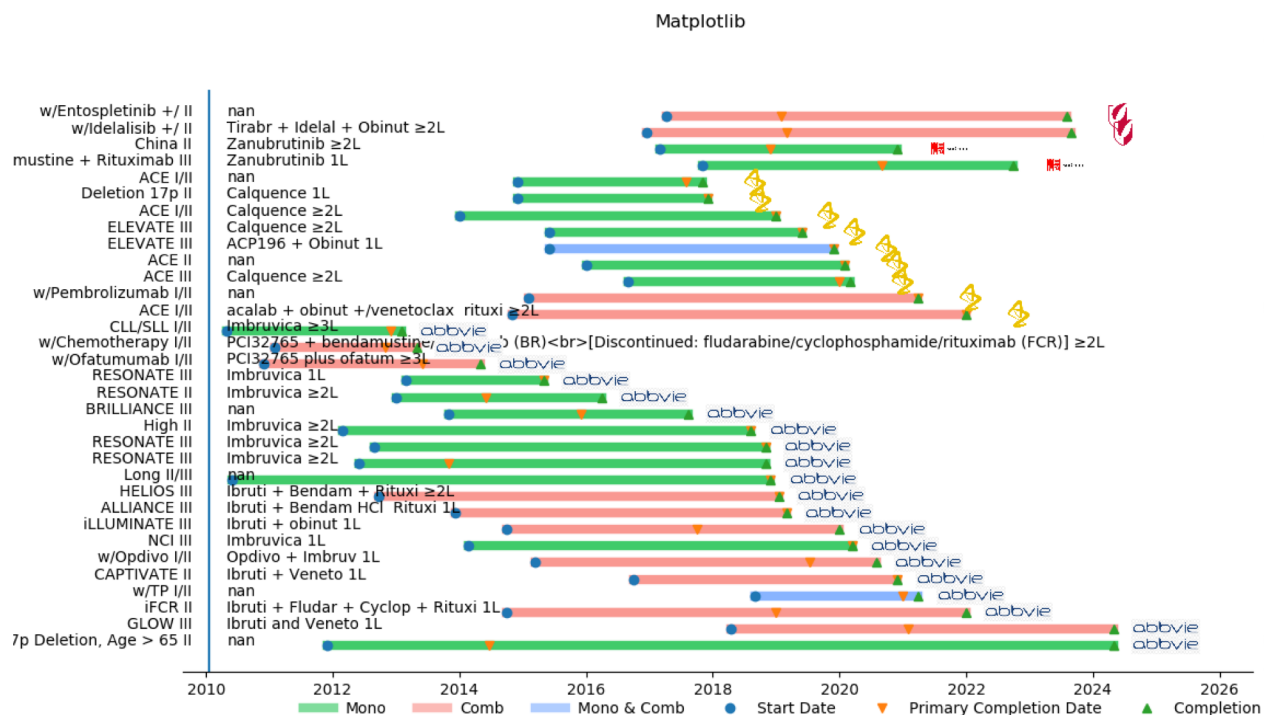
rating <- search_list %>%
  html_nodes("td a") %>%
  html_attr("href") %>%
  as_tibble() %>%
  filter(!is.na(value)) %>%
  filter(stringi::stri_startswith(value, fixed = "/ct2/show/NCT"))
```

Customized Symbol Image in Matplotlib and ggplot2

Raster images need to be read as binary objects and then are added as annotations using **AnnotationBbox** in Matplotlib and **annotation_raster** in ggplot2.

In Matplotlib multiple annotations need to be added in explicit loop.

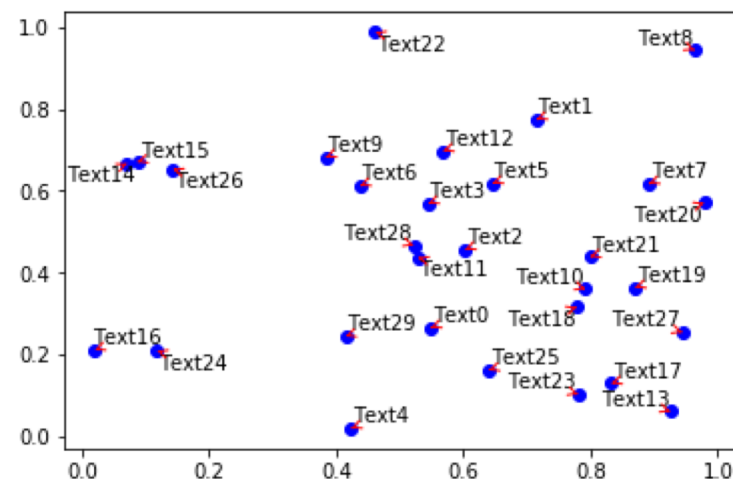
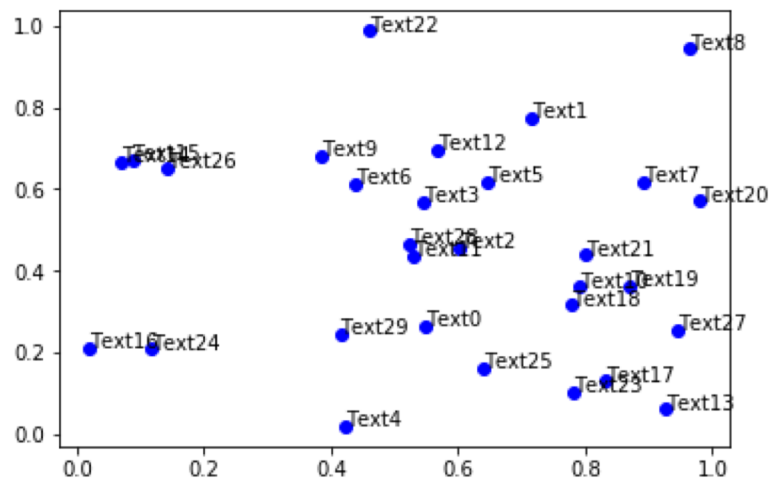
In ggplot2 multiple annotations can be added using **mapply** function.



Data Label Overlapping in Python and R

Python: adjustText

R: ggrepel



Further Information



Speaker E-mail Address: chiwei@hrglobe.cn

 *PhUSE China*
 *phuse.eu*
 *[@PhUSETwitta](https://twitter.com/PhUSETwitta)*
 *[/PhUSE](https://www.facebook.com/PhUSE)*
 *phusewiki.org*

