Paper CT11

RECISTance is Useless

Daniel Lindenbaum, Phastar, London, United Kingdom

A NOVEL VISUALISATION OF RECIST DATA USING BUBBLES, ON A TIMELINE

Not The Hitchhiker's Guide to the Galaxy (Douglas Adams,1978) but an adventure into the graphical universe of tumour measurement data, using the RECIST cancer grading scale. With the mission of capturing the distilled meaning of such data, in an easy to interpret form, the task is explored and an initial collection of digital steps composed as a SAS® macro. A novel and informative visualisation, by means of a patient level time series depicting tumour load status, is presented. For early phase work it is possible to illustrate the tumour status across an entire study, on a single page.

RECIST IS

[1]Response Evaluation Criteria in Solid Tumours (RECIST). It is an oncology tool that consists of a set of rules and measures, by which, disease status and dynamics can be classified. Lesions are selected for study and each will fall into the category of target (maximum of 5) or non-target lesions (no limit to numbers). The sum of the diameters (of the longest axis for each lesion) for the target lesions is calculated and all relevant lesions (target and non-target) are counted. The sum of longest diameters (SLD) and the lesion counts both contribute to the assessment of disease at each visit. Possible disease statuses are complete response (CR), partial response (PR), stable disease (SD) and progressive disease (PD). There is also a status of non evaluable (NE) but this is not within the scope of this paper.

As well as “per visit” disease statuses, there are overall summaries of disease. These include: Best Overall Response (BOR), Progression-Free Survival (PFS), and Duration of Response (DOR). BOR describes the best tumour response achieved during treatment, PFS measures the time a patient survives without disease progression, and DOR indicates how long a patient experiences a response (such as tumour shrinkage or total dissappearance) before progression or other treatment interruption.

It should be noted that this is not an exhaustive implementation of the RECIST criteria and this paper is based on a simplified model (the high level principles are adhered to as the basis of derivations). The intent is to illustrate this novel method of data expression rather than a thesis on the full model. For the complete model, please see the reference section (Eisenhauera et al).

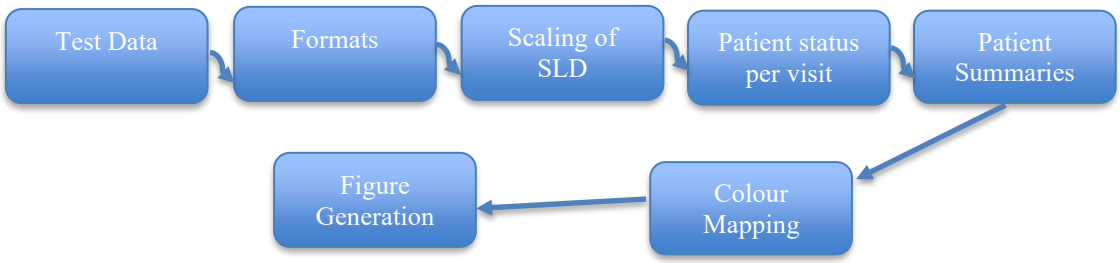
Characterisation of disease status

Status	**Type**	**Criteria**	**Abbreviation**
Complete Response	*By Visit*	*All target and non-target lesions gone (SLD=0)*	*CR*
Partial Response	*By Visit*	*No new lesions plus SLD reduced by at least 30%*	*PR*
Stable Disease	*By Visit*	*Not qualifying for CR, PR or PD*	*SD*
Progressive Disease	*By Visit*	*New lesions or SLD increased by at least 20%*	*PD*
Best Overall Response	Overall	CR > PR > SD > PD	BOR
Progression Free Survival	Overall	Elapsed time from study start to PD or Death (or censor)	PFS
Duration of Response	Overall	Elapsed time from first CR or PR to PD or Death (or censor)	DOR
[2]Objective Response Rate	Overall	Percentage of patients with CR or PR	ORR
[2]Clinical Benefit Rate	Overall	Proportion of patients who experience a positive outcome for a given period (often 6 months)	CBR

[1] RECIST version 1.1 is used as a basis for this paper.

[2] These summaries are not covered within this paper, but are potential expansions to the output figures.

Process Breakdown



Test Data

Test data are created for 30 patients (split across 3 treatment groups of 10 per group). Each patient is generated with 12 study visits. There are additional visits (13-18) that are for patient summary information. Each patient is assigned a random number of target and non-target lesions at baseline with lesion diameters initialised for all target lesions. Baseline SLD is set between 50mm and 250mm and lesion counts are set between 1 and 4 for target (TL) and non-target (NTL). Lesion counts and diameters are then adjusted at each subsequent visit. Random number generation is used as a basis for these values. All values can shift either up or down at each visit, by randomly determined proportions. Lesion counts and diameters are simulated within realistic numerical boundaries. Patient death is simulated at a 4% level per visit.

```
⇩ Baseline visit
if visit=1 then do;
  sld=round(rand("integer", 50, 250), 5); ⇦ assign baseline SLD(50 to 250 mm)
  target=1+rand("integer", 1, 3);         ⇦ assign baseline TL count(2-4 target lesions)
  ntarget=1+rand("integer", 1, 3);        ⇦ assign baseline NTL count(2-4 non-target lesions)
end;

⇩ Post-baseline visits
updown=rand("integer", 0, 1);       ⇦ add or subtract a lesion from the count at each visit
factor=rand("integer", 0, 30)/100; ⇦ a random factor to assign SLD for each visit(0-30%)
if visit gt 1 and updown=0 then sld=sld-(sld*factor);      ⇦ adjust SLD down at post-baseline
else if visit gt 1 and updown=1 then sld=sld+(sld*factor); ⇦ adjust SLD up at post-baseline
  if 1 lt visit le 12 then do;
   ntr=rand("integer", 0, 100); * non-target and SLD random % generation;
   if ntr lt 3 then do; ⇦ 3% chance of increasing NTL(count) and TL(SLD)
     ntarget2=sum(ntarget, (rand("integer", 0,1))); ⇦ add 0 or 1 to NTL(count)
     ⇩ assuming that if NTL(count) increase then TL(SLD) also increases
     i.e. general disease progression
     sld=round(sld*1.2, 5);
   end;

        ⇩  35% (70%/2) chance of reduction in non-target, target counts
        ⇩ 70% chance of reduction in SLD
        else if ntr gt 70 then do;
          ntarget=max(0, ntarget-(rand("integer", 0, 1))); ⇦ NTL 50% chance of decrease by 1
          target= max(0, target-(rand("integer", 0, 1)));  ⇦ TL 50% chance of decrease by 1
          sld=round(sld*0.9, 5); ⇦ reduce SLD by 10% - irrespective of TL/NTL reduction
        end;
```

Scaling

Scaling of the SLD values is needed, to assign the sphere sizes for the bubble plot. The level of scaling is controlled via the assignment of a value to a macro variable.

```
⇩ identify the min/max SLD values from all subjects - range for intervals
proc sql noprint;
  select min(sld), max(sld) into: minsld, :maxsld
  from recist;
quit;

⇩ factor to scale SLD bubble sizes: smaller factor=greater symbol diameter
%let scale=500;

data recist3(drop=rsld updown);
  set recist2;
  ⇨ scale the SLD for plotting purposes - based on the range of SLD values
  rsld=&maxsld - &minsld;
  factor=rsld/&scale; ⇦ determine the factor by which to scale the SLD
  sld2=round(sld/factor, 1);
run;
```

Derive patient disease status at each visit

Of the SLD values is needed, to assign the sphere sizes for the bubble plot.

```
  ⇩ if zero lesions (complete response) remaining then assign SLD to a small nominal amount
  otherwise the disease status will not be plotted
  if target=0 and ntarget=0 then do;
    sld=0;
    sld2=0.01; ⇦ nominal SLD to ensure that the visit disease status is plotted
    stat='CR'; ⇦ assign disease status
    castat=6;  ⇦ assign numeric version of disease status
  end;
  else castat=999; ⇨ set numeric disease status to the numeric code for missing values
  ⇨ assign baseline SLD values
  if visit=1 then do;
    bsld=sld;
  end;
  ⇨  examine the % changes in SLD, from baseline
  ⇨  assign partial responses - if present on study visits
  else if 1 lt visit le 12  and (dtvis=. or visit lt dtvis) then do;
    csld=(sld/bsld)*100;
    if csld le 70 and stat ne 'CR' then do;
      castat=4;
      stat='PR'; ⇦ assign disease status
    end;
  end;
  ⇩ check the % change in SLD - to determine whether any increase(>=20% and at least +5mm)
  meets criteria for PD
  if visit=2 then nsld=sld;
  else if visit le 12 and (dtf='' or visit lt dtvis) then do;
    if sld lt nsld then nsld=sld;
    if nsld ne 0 then pdsld=(sld/nsld)*100;
    if (pdsld ge 120 and (sld-nsld ge 5)) or ntchg gt 0 then do;
      castat=3;
      stat='PD';⇦ assign disease status
      ⇨ assign the PD visit number and PD status flag - to use at later visits
      if pdvis=. then pdvis=visit;
      pdstat='Y';
    end;
  end;
  ⇨ if subject is not dead and has no prior PD and status has not already been set to PR/CR
    - the visit must be - Stable Disease;
  if dtf ne 'Y' and pdstat ne 'Y' and stat ='' then do;
    stat='SD'; ⇦ assign disease status
    castat=5;
  end;
```

Derive the study level, summary statuses and values, for each patient.

BOR

```
⇨ fetch all unique disease status per subject - to determine the best overall response (BOR)
proc sql;
  create table bor as
  select distinct subject, stat
  from recist5
  where stat ne '';
quit;

proc transpose data=bor out=tbor prefix=V;
  var stat;
  by subject;
run;

⇨ assign the BOR for the study - per subject
data tbor2(keep=subject bor  allresp);
  set tbor;
  allresp=catx(',',v1,v2,v3);
  visit=14; ⇦ assign the BOR to summary visit 14
  if index(allresp,'CR')      then bor='CR'; ⇦ assign BOR as Complete Response
  else if index(allresp,'PR') then bor='PR'; ⇦ assign BOR as Partial Response
  else if index(allresp,'SD') then bor='SD'; ⇦ assign BOR as Stable Disease
  else if index(allresp,'PD') then bor='PD'; ⇦ assign BOR as Progressive Disease
run;
```

PFS

```
retain pfs;
by subject visit;
if first.subject and stat not in(' ','PD','DT')    then pfs=1; ⇦ initialise PFS time
if 1 lt visit le 12 and stat not in(' ','PD','DT') then pfs+1; ⇦ +1 for each PD/Death free
if visit=16 then do;                                             visit
  lbvx=compress(put(pfs-1,8.)); ⇦ assign the final PFS duration to summary visit 16
end;
```

DOR

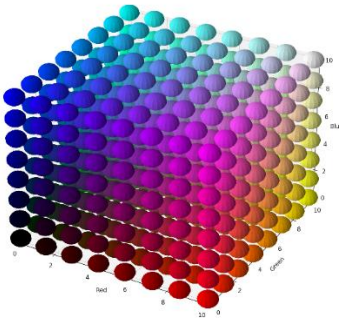
```
retain dor sddur dorstart dorstop;
⇩ assign the "DOR start flag" and "DOR start visit" for all post PR/CR visits, until PD/DT
if 1 lt visit and stat in('PR','CR') and dorflag ne 'Y' then do;
  dorstart=visit;
  dorflag='Y';
end;
⇩ if PD or DT is encountered then set the "DOR stop flag" and subtract the "DOR start visit"
from current visit
else if visit le 12 and stat in('PD','DT')  then do;
  dor=(visit-dorstart);
  dorstart=.;
  dorstop='Y';
end;
⇨ if the last study visit and response continues - set end of DOR to end of study
if visit=12 and dorstart gt . and dorstop='' then dor=(visit-dorstart)+1;
⇨  assign the DOR value to the summary visit
  if visit=15 then lbvx=put(dor, dor.); ⇦ assign the final DOR to summary visit 15
```

DOSD

```
if 2 le visit le 12 and stat in('SD','PR','CR') and sdflag ne 'Y' then do;
  sdstart=visit;
  sdflag='Y';
end;
⇨  if PD or DT is encountered then set the "SD stop flag" and subtract the "SD start visit"
from current visit;
else if visit le 12 and stat in('PD','DT')  then do;
  sddur=(visit-sdstart);
  sdstart=.;
  sdstop='Y';
end;
⇨  if the last study visit and response continues - set end of DOR to end of study
if visit=12 and sdstart gt . and sdstop='' then sddur=(visit-sdstart)+1;
⇨ assign the SDDUR value to the summary visit
if visit=17 then lbvx=put(sddur, dor.); ⇦ assign the final DOSD to summary visit 17
```

Colour Cube

Proportions of Red, Green and Blue to generate the colour gradients for SLD spheres.



RGBHEX Macro

```
⇨ macro to convert proportions of red, green and blue (out of 255 max) to a hexadecimal SAS
colour code
⇨ this uses the SAS supplied HEX2 format;
⇨ eg1 %rgbhex(rr=255, gg=0, bb=255) will give the hexadecimal colour code for a saturated
purple
⇨ eg2 %rgbhex(rr=100, gg=0, bb=100) will give the hexadecimal colour code for a less
saturated purple

%macro rgbhex(rr,gg,bb);
  %sysfunc(compress(CX%sysfunc(putn(&rr,hex2.)) ⇦ colour codes start 'CX', then red component
  %sysfunc(putn(&gg, hex2.))    ⇦ then green component
  %sysfunc(putn(&bb, hex2.))); ⇦ then blue component
%mend rgbhex;
```

COLGEN Macro

```
⇨ identify the "block size" for the entire SLD range - divided into 15 parts (15 ranges of
SLD with different colour saturation for each)
  data _null_;
    range2=(&maxsld2 - &minsld2)/15;
    call symput('block', compress(put(range2,8.)));
  run;

  %put Block Size=&block;
  %let grpspan=12;
  ⇩ initialise all colour values to zero
  %let pos1=0;
  %let pos2=0;
  %let pos3=0;
  ⇩ assign base-colour as fully saturated (255/255)
  %if       &basecol=red    %then %let pos1=255;
  %if       &basecol=green  %then %let pos2=255;
  %if       &basecol=blue   %then %let pos3=255;
  ⇩ assign secondary colour as fully saturated
  %if       &second=red     %then %let pos1=255;
  %if       &second=green   %then %let pos2=255;
  %if       &second=blue    %then %let pos3=255;
  ⇩ assign the colour that is the variable component of the 'mixture'
  %if       &varcol=red     %then %let pos1=%nrstr(&x*(&grpspan));
  %else %if &varcol=green   %then %let pos2=%nrstr(&x*(&grpspan));
  %else %if &varcol=blue    %then %let pos3=%nrstr(&x*(&grpspan));

  ⇩ generate 'x' colour shades - based on number of colour group values specified
    With increasing saturation for each iteration of x loop
  data _null;
    %do x = 1 %to 15;
      tvar="%rgbhex(&pos1,&pos2,&pos3)";
      s1="&pos1"; s2="&pos2";s3="%unquote(&pos3)";
      ⇩ assign the series of colour shades to a series of macro variables
      call symput("newcolr"!!compress(put(&x,8.)), compress(tvar));
      output;
    %end;
  run;
```

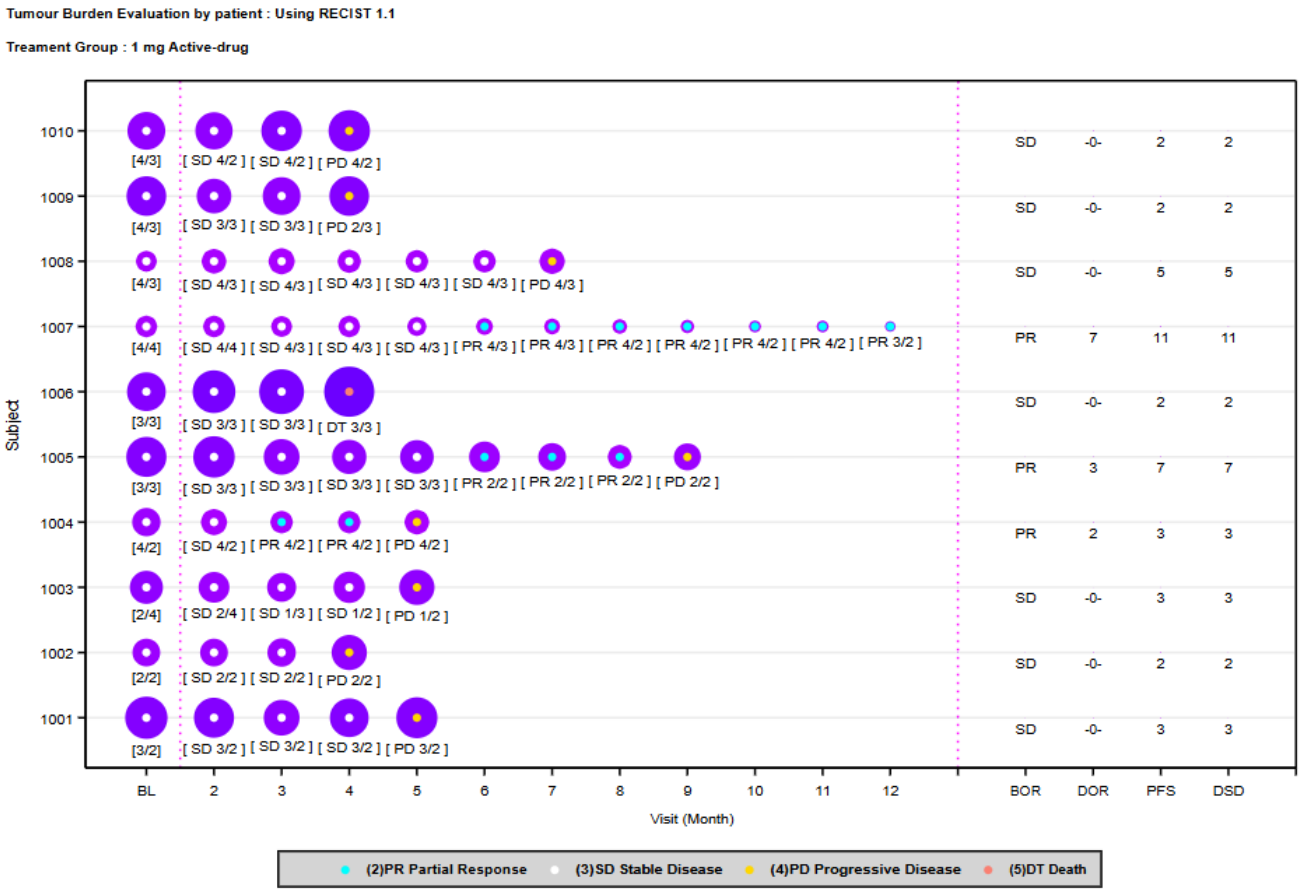
GTL to build the figure (proc template & sgrender)

```
            ⇩ associate colours with SLD ranges - for bubbles
            rangeattrmap name="recmap" ;
              range missing / rangealtcolor=white rangecolor=white;
              range 0                 - %eval((1*&block)-1)  / rangecolor=&newcolr15;
              range %eval(1*&block)   - %eval((2*&block)-1)  / rangecolor=&newcolr14;
              range %eval(2*&block)   - %eval((3*&block)-1)  / rangecolor=&newcolr13;
              range %eval(3*&block)   - %eval((4*&block)-1)  / rangecolor=&newcolr12;
              range %eval(4*&block)   - %eval((5*&block)-1)  / rangecolor=&newcolr11;
              range %eval(5*&block)   - %eval((6*&block)-1)  / rangecolor=&newcolr10;
              range %eval(6*&block)   - %eval((7*&block)-1)  / rangecolor=&newcolr9;
              range %eval(7*&block)   - %eval((8*&block)-1)  / rangecolor=&newcolr8;
              range %eval(8*&block)   - %eval((9*&block)-1)  / rangecolor=&newcolr7;
              range %eval(9*&block)   - %eval((10*&block)-1) / rangecolor=&newcolr6;
              range %eval(10*&block)  - %eval((11*&block)-1) / rangecolor=&newcolr5;
              range %eval(11*&block)  - %eval((12*&block)-1) / rangecolor=&newcolr4;
              range %eval(12*&block)  - %eval((13*&block)-1) / rangecolor=&newcolr3;
              range %eval(13*&block)  - %eval((14*&block)-1) / rangecolor=&newcolr2;
              range %eval(14*&block)  - %eval(15*&block)     / rangecolor=&newcolr1;
            endrangeattrmap ;

            ⇩ associate the colour map with the SLD2 variable
            rangeattrvar attrvar=rangevar var=sld_p attrmap="recmap" ;

            ⇩ assign colours for the disease status scatterplot (overlaid on bubbles)
            discreteattrmap name="dstatus" ;
              value " "                          / markerattrs = (color=white transparency=1);
              value "(4)PD Progressive Disease"  / markerattrs = (color=gold);
              value "(2)PR Partial Response"     / markerattrs = (color=cyan);
              value "(3)SD Stable Disease"       / markerattrs = (color=white);
              value "(1)CR Complete Response"    / markerattrs = (color=lime);
              value "(5)DT Death"                / markerattrs = (color=salmon);
            enddiscreteattrmap ;

            ⇩ associate the scatterplot colour map with the STAT variable
            discreteattrvar attrvar=rv var=stat attrmap="dstatus" ;



layout overlay/
       ⇨ define the X axis
       ⇨ define the Y axis
       ⇨ add 2 reference lines to separate baseline and summary sections
       ⇩ define the bubbleplot, based on SLD2, linked to colour map via "rangevar"
       bubbleplot x=visit y=subject size=sld_p / datalabel=status datalabelposition=bottom
       datalabelattrs=(size=5 color=black family="light")
       colorresponse=rangevar relativescale=false relativescaletype=proportional
       display=(fill) bubbleradiusmin=0 bubbleradiusmax=12 includemissinggroup=true;

       ⇩ define the scatterplot based on STAT(linked via the "rv" reference
       scatterplot x=visit y=subject / markerattrs=(size=5  symbol=circlefilled) group=rv
       groupdisplay=overlay name='rv2';
       ⇩ legend for the scatterplot (disease status)
       discretelegend 'rv2' / sortorder=ascendingformatted backgroundcolor=verylightgrey
       across=6 valueattrs=(weight=bold size=5pt family='light');
endlayout;
```

Output

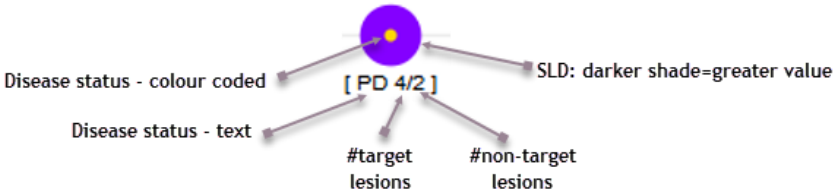
Primary output I

Macro call: `%colgen(basecol=blue, second=blue, varcol=red, treatm=1)`



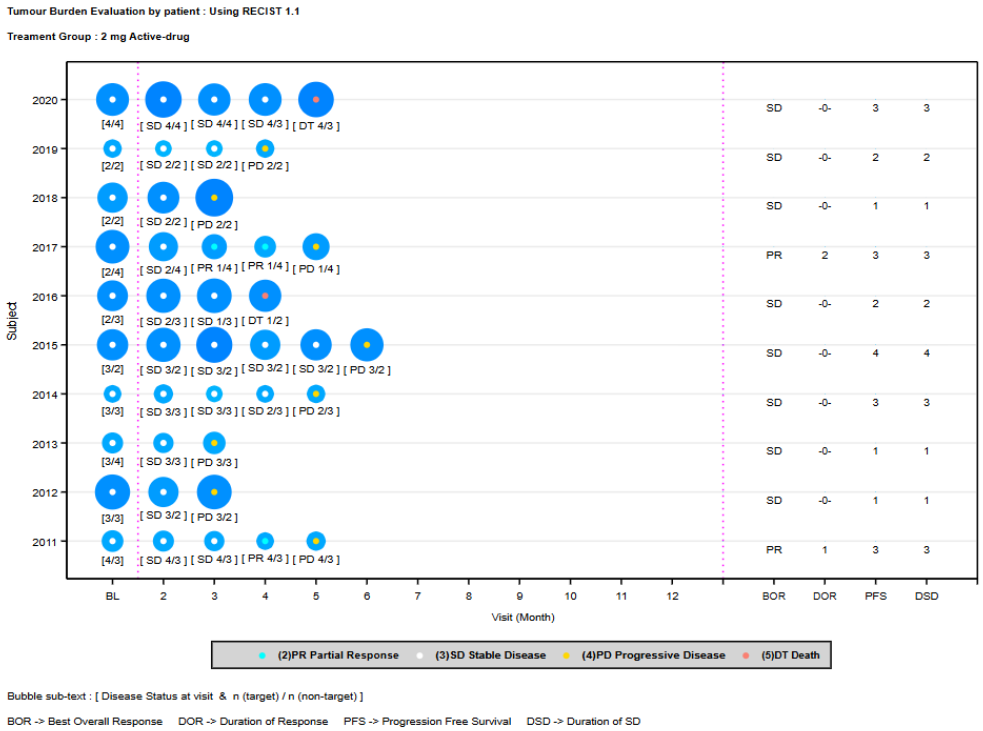
Bubble sub-text : [Disease Status at visit & n (target) / n (non-target)]

BOR -> Best Overall Response DOR -> Duration of Response PFS -> Progression Free Survival DSD -> Duration of SD



Primary output II

Macro call: **`%colgen(basecol=blue, second=blue, varcol=green, treatm=2)`**

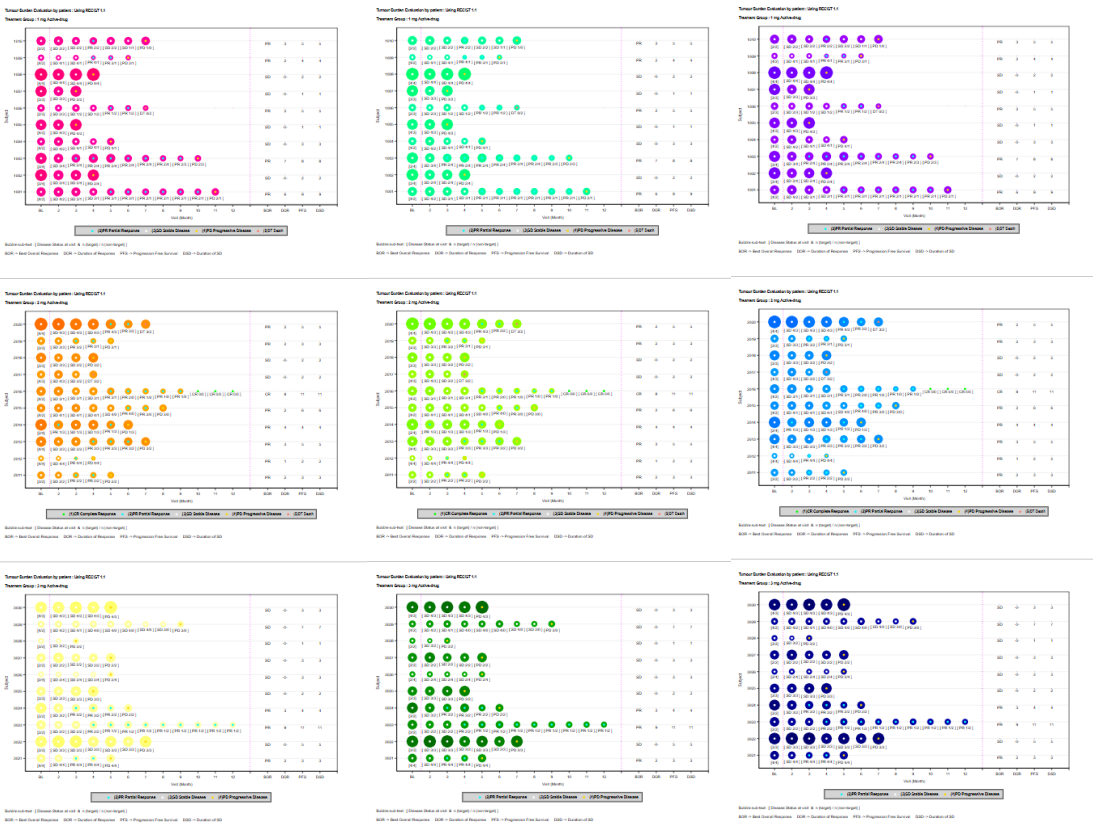


Primary output III

Macro call: **`%colgen(basecol=blue, second=blue, varcol=blue, treatm=3)`**

Illustration of a variety of colour gradients

Selection of different ‘base’ colours can be used to differentiate between groups.



Full Code

Please follow this link and download the code file:

https://drive.google.com/file/d/1omgVnZP_uxd8arqkC7YNPrBp6LP7DS5F/view?usp=sharing

Full code is also available by contacting the author via the email address below.

CONCLUSION

The RECIST model for quantifying and categorising tumour related disease is complex and generates a number of potential measures for response to treatment, or the lack thereof. These measures are usually presented in tabular format and can take some intellectual digestion to interpret. The code excerpts presented here (along with full code and macros as appendix) demonstrate a method by which a striking presentation of results can be made, along with ease of interpretation and observation of data trends.

REFERENCES

New response evaluation criteria in solid tumours: Revised RECIST guideline (version 1.1) E.A. Eisenhauera,* , P. Therasseb , J. Bogaertsc , L.H. Schwartzd , D. Sargente , R. Fordf , J. Danceyg , S. Arbuckh , S. Gwytheri , M. Mooneyg , L. Rubinsteing , L. Shankarg , L. Doddg , R. Kaplanj , D. Lacombec , J. Verweijk. EUROPEAN JOURNAL OF CANCER 45 (2009) 228 – 247.
New response evaluation criteria in solid tumours: Revised RECIST guideline (version 1.1)

SAS/GRAPH ® Colors Made Easy Max Cherny, GlaxoSmithKline, Collegeville, PA. (PHARMASUG 2009)
SAS/GRAPH ® Colors Made Easy

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Daniel Lindenbaum
Phastar
2D Bollo Lane
Chiswick
London W46LE
UK

daniel.lindenbaum@phastar.com

www.phastar.com