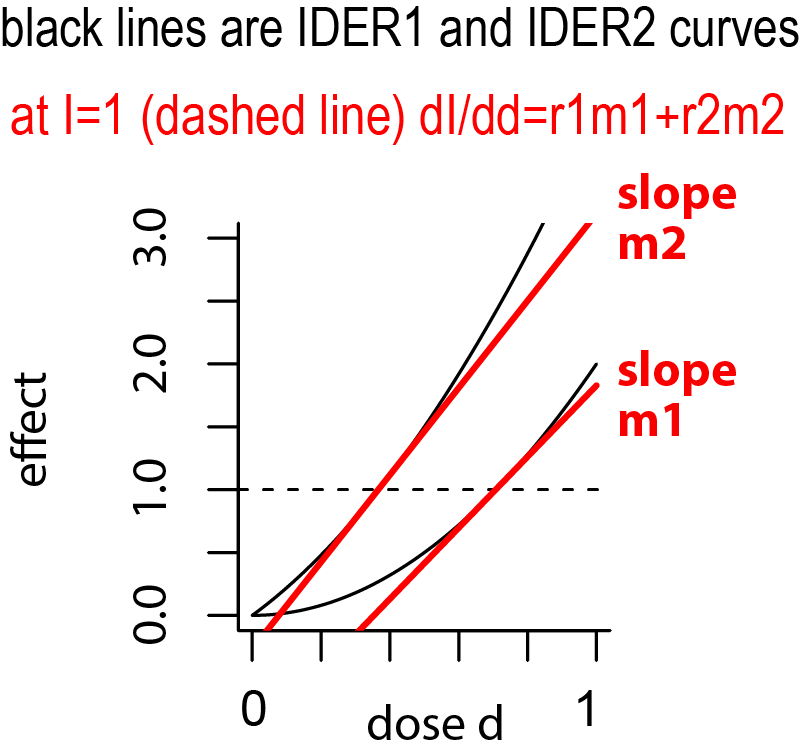
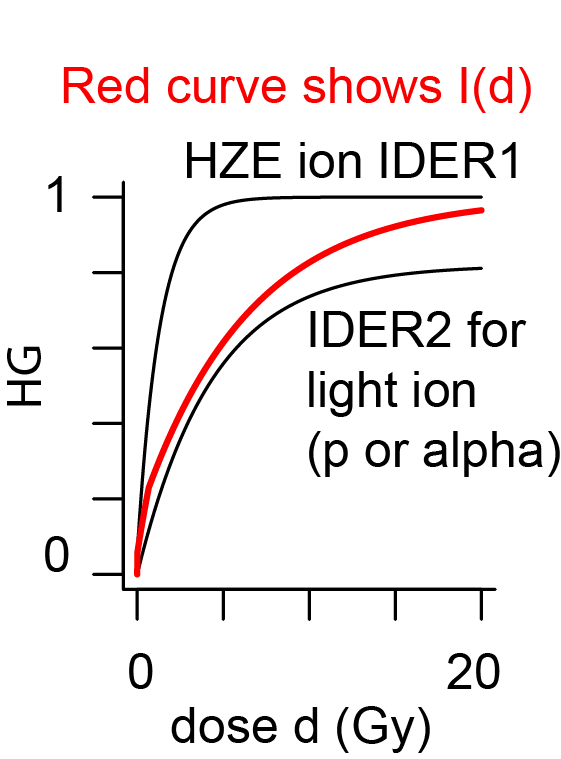
1. Mark questions and report on Cucinotta 2017. Edward questions on acronyms and terminology (see below). Edward describes GitHub; Sachs question on comment. The following item intentionally written using a lot of acronyms and specialized terminology that you both have heard and should now become fluent in. Please try to review the terms and be sure to interrupt my presentation if there is anything whose meaning is obscure.

2. Sachs report on incremental effect additivity default MIXDER for a mixture that (a) has very well behaved component IDERs or (b) has, like our HG ones, component IDERs that do not approach infinity as dose approaches infinity.

(a) In the figure a baseline no-synergy antagonism incremental effect additivity MIXDER I(d) for an (r1, r2=1-r1) mixture of ion 1 with ion 2 (e.g. 50%-50%) is calculated as follows. Assume IDERs E1 and E2 are known, are monotonically increasing, and have no upper limit as dose increases indefinitely. See the black curves. Starting at I(0)=0 we integrate dI/dd=F(I), where the slope F(I) is determined as follows. when I has already reached some level, e.g. I=1.000 as shown, find the dose each ion would have to have to reach that level if the ion is acting on its own (intersections of the dashed and solid black curves). Because each IDER is monotonic. At those points find the slopes dE1/dd1 and dE2/dd2 of the IDER tangent line. Then ion 2 contributes r2m2 to F(I) and ion 1 adds on r1m1. By the existence and uniqueness theorem for 1 non linear ODE with a given initial value there is one and only one MIXDER I(d). In programming the main simplification is that uniroot can always find the dose at the intersection points; programmed properly it should never complain that it can’t find an intersection point.

(b) However the HG data we are concerned with has a different behavior for moderate to large doses. And, reflecting this difference the IDER do not increase indefinitely, the intersection points shown above may not exist, uniroot’s abortive efforts to find the intersection points will cause fatal errors, and then I(d) can only be competed for small mixture doses. This problem can be overcome (next page) but requires considerable extra machinery, mathematical and programming wise.

The figure shows computations for a 50-50 mixture of a high LET Fe beam and a low LET 1H1 or 2He4 beam. The IDER for the Fe beam is the IDER we have developed and actually plan to use because it is a big improvement over the previous models. It is calibrated with our actual data. It never gets bigger than 1 because the endpoint “fraction of mice that have at least one HG tumor” cannot be greater than 1. That is different from our previous figure but not in itself a problem because of course we are not interested in mixture effects bigger than 1 either.

The low LET light ion IDER is a toy IDER calibrated with our actual data and using it requires care. For some reason the light GCR ion data strongly indicate an upper limit of about 0.6-0.8 for this endpoint. Our toy IDER for this data has a maximum of 0.8 which it never exceeds. So once the HZE ion drives I(d) above 0.8 uniroot can not find the now non-existent intersection point and I(d) becomes undefined, even though we are interested in effects between .8 and 1. That kind of problem occurs often and causes trouble.

One solution is to argue as follows. As we approach *I*=0.8 from below ion 2 is already doing very little. At 0.78 uniroot can find the intersection point and we can calculate the slope m2 there to add 0.5m2 to dI/dd.But m2 will obviously be very small since the curve is so flat there (whereas at I=0.8 slope m1 is still very large). So we simply extend our definition to say that at I=0.8 or more ion 2 does nothing at all; all further changes to I(d) come from ion 1, which eventually drives I(d) up to 1, but slowly because ion2 is only acting at half strength.

So one of the two main steps remaining for our whole project is to program this idea carefully using actual instead of toy IDER for the low LET contributions. Care is required. If we let ion 2 stop at I=0.78 that is a bit too soon. If we try to go to I=0.79999 uniroot may miss by mistake and think there is no solution. But that is just a technical programming issue. In principle we have a solution and only need to implement it.

This solution we will use is not the best solution. If ion 2 hates effects above 0.8 why doesn’t it put up a fight when ion 1 drives I(d) above 0.8 and force a compromise upper limit between 0.8 and 1? There is a way to do that but we will not use it for our first paper.