Andrey:   
About the plots: I don't remember exactly which code did I sent you, but is there any account for circularity? In some of the recent analyses, I've used this function at some point before making the plots to get rid of discontinuities on the edges (because -90 and 90 should be approximately the same):

prep\_for\_circ<-function(data, circ\_var,circ\_borders=c(-90,90), circ\_part = 1/6){  
  circ\_range <- max(circ\_borders)-min(circ\_borders)  
    
  data1<-copy(data[get(circ\_var)<(circ\_borders[1]+circ\_range\*circ\_part),])  
  data1[,(circ\_var):=get(circ\_var)+ circ\_range]  
    
  data2<-copy(data[get(circ\_var)>(circ\_borders[2]-circ\_range\*circ\_part),])  
  data2[,(circ\_var):=get(circ\_var)- circ\_range]  
  print(c(data1[,.N],data2[,.N], data[,.N]))  
  rbind(data,data1,data2)  
}

It simply appends a part of the data (circ\_part, 1/6 from each end by default) to the other end so that the local regression that is used to determine the curve is more precise at the ends.

So you do something like

ds\_data\_adj<-prep\_for\_circ(ds\_data, 'ctpd')

and then use adjusted data for the plots.

Lisa:   
I wrote the code for the analyses myself, so for the plots there is no account for circularity yet. Is it also necessary when ctpd is maximum -80° and minimum 80°? Because I would think that there is only a problem when the mean orientation of the previous distribution would be for example 90° and the target orientation in the prime streak would be -90°? Because then ctpd would be 180°, while it should be 0°.  Or maybe I understand it totally wrong?

Andrey:   
The main problem here is that a locally weighted regression (such as the one in geom\_smooth by default) uses neighborghood data to "smooth" the estimate at each point. So when we have an estimate at 0 it is based not only on the values at 0 but also on the other values with weights decreasing with distance between them. But then for ctpd at -90 you don't have the values < -90 to get an estimate, so you loose the precision. The further away from the boundary, the more precise you get. There are also other things like the bandwith used for smoothing that might depend on the range of values, but I'm afraid I don't know much else.