**Remarks about v.2 code :**

* Probably multiple functions common to calcregionAG & addregionAG that could be made as separate supporting function
* Memory ‘safety net’: because calcregionsAG and addregionsAG sometimes have to test a really high number of possible combinations, resulting in extra long vectors/data.frames and requiring a lot of memory. So I came up with a workaround to prevent data.frames exceeding lengths of 0.5e+9 (which corresponds +/- to 2Gb). If a data.frame exceeds that length, it is subdivided in smaller data.frames of max 2Gb, the analysis is ran on each data.frame at a time and results are stored in a “tmp” folder. Once the analysis is done running on all the data.frames, it imports the results in R.
* Nmodel\_saved in calcregionsAG: because the number of vertebrae and regions was so high for my models, I was trying to reduce as much as possible the number of models I was returning in the $results output of the calcregionsAG function, the way I found was only saving models with a RSS lower than 0.5 \* SD(RSS of all models). However, this implies that, when we want to calculate the weighted mean and sd of BP position, we sometimes have a very low % of models to implement in that function (less than 5% of all possible models). From my experience with my data, it’s usually not a problem because, in general only the first few models have a much better fitting (lower RSS) than all other models. But just to be sure, I still added a warning message in the calcBPvar function stating that. But maybe you have a better solution to still be able to return the results of all the models tested in calcregionsAG
* In calcregionsAG and addregionsAG, I had to use a “trick” to be able to pass Xvar and Yvar to the formula and model fitting by specifying at the begnning of the functions code: *Xvar <<- Xvar* and *Yvar <<- Yvar*. It did the try in plain R, however, after writing the vignette in Rstudio, I noticed that doing so would overwrite Xvar and Yvar in my environment. So if, in my working environment, I define a vector Xvar <- 1:10 and then, in calcregionsAG is use the parameter Xvar=5:15, then, in my working environment, Xvar will become 5:15. I guess there’s a better way of solving the issue of passing Xvar and Yvar to the formula and model fitting but I don’t know how…
* You’ll notice the gc() function is called a lot in the calcregionsAG and addregionsAG function to try to minimize the impact of the computation on the memory but I’m not sure if this is fine to leave that there for CRAN.