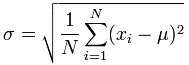
Growth change values were computed on the basis of effects of variants as predicted by SNAP (Bromberg and Rost, 2007). In the SNAP model, scores range from -100 to 100 and any positive score indicates change from wild-type functionality of the protein. As SNAP does not differentiate between negative and positive growth effects (more or less than wild-type), we made an assumption for all predictions that change is generally deleterious to the function; *i.e.* all growth predictions are below 1. To compute the growth rates and standard deviation of the prediction, all ten SNAP component network predictions were obtained for each variant. We further processed the predictions as follows:

1. For single variants, *growth* was computed on the basis of the overall SNAP score (the average of the ten components). If *score* was:
   1. *score < -40* (very neutral) 🡪 *growth = 1* (no change)
   2. *-40 < score < 0* (neutral) 🡪 *growth = 0.99+ (score/10000)* (very slightly changed growth)
   3. *0 < score < 20* (slightly non-neutral) 🡪 *growth = 1-score/100* (slight but visible change in growth)
   4. *20 < score < 40* (non-neutral) 🡪 *growth = 0.9-score/100* (change in growth)
   5. *score > 40* (very non-neutral) 🡪 *growth = max (0 or 0.7-score/100)* (significant change in growth)

Standard deviation (σ = *std*) of the growth was computed from the ten SNAP component network scores (*xi*) according to the formula:

where N = 10 and μ = overall SNAP *score*

*Std* was further pro-rated for growth as follows:

1. *score > 0* 🡪 *std = std/100* (for variants clearly predicted to alter growth)
2. *(std + score) > 0* 🡪 *std = (std+score)/100* (for variants in the possible alteration spectrum)
3. *40 < score < 0* 🡪 *std = min (0.05* **OR** *std/100)* (for variants where change in growth is unlikely but possible)
4. *score < -40* 🡪 *std = min (0.01* **OR** *std/100)* (for variants where change in growth is very unlikely)
5. For multiple variants we chose the lowest rate of *growth* for any of the individual variants as described above (and report the corresponding *std*)