Mortality in the MITUS Model

Mortality in the MITUS model is the sum of background mortality μ_{bg} and excess mortality due to tuberculosis disease μ_{tb}.

$$\mu_{tot} = \mu_{bg} + \mu_{tb}$$

Background Mortality

The background mortality structure in the MITUS model is dependent on time (t), age group (ag), mortality group (rf), and risk group (rg). The general equation for background mortality in the model is:

$$\mu_{bg_{t,ag,rg,rf}} = \mu_{ag,t} RR_{\mu_{rg}} RR_{\mu_{rf}}$$

where the excess mortality due to a mortality or risk group is operationalized as a rate ratio, $RR_{\mu_{rf}}$ and $RR_{\mu_{rg}}$, respectively.

$RR_{\mu_{rf}}$ varies based on mortality group rf. It value is determined by the marginal distribution of the population across the mortality group and the increased risk of mortality between group 1 and 4. This relationship is defined as:

```r
RF_fact=20

RRmuRF    <- rep(NA,4);
names(RRmuRF) <- c("RF1","RF2","RF3","RF4")

RRmuRF<-exp((0:3)/3*log(RF_fact))
RRmuRF<-RRmuRF/sum(RRmuRF*mort_dist)
```

Specifically, $RR_{\mu_{rf}}$ is defined in such a way that the average weighted by the mortality distribution is equal to one:

$$\sum_{rf=1}^{4} RR_{\mu_{rf}} frc_{rf} = 1$$

Through this specification, we are able to reproduce the background mortality for each age group in the model. Current values of $RR_{\mu_{rf}}$ are 0.656049, 1.780791, 4.833810, 13.120980 for groups 1-4.

$RR_{\mu_{rg}}$ accounts for the excess mortality in the homeless population; it is parameterized with the value (3.4 [2.3,4.8]).

In addition to the equation above, the baseline mortality, $\mu_{ag,t}$, is modified by two calibration parameters: tun_{μ_t} and $tun_{\mu_{ag}}$.

tun_{μ_t} is parameterized with the value 1.0 [.85,1.15]. It scales the background mortality overtime linearly.

$tun_{\mu_{ag}}$ is parameterized with the value 0.0 [-.2,.2]. It scales the background mortality log-linearly, with an additional increasing scale by the age group $ag = [1:11]$.
The equation and application of this in R are found below:

$$\mu_{ag,t} = \mu_{base_{ag,t}} tun_{\mu_t} e^{ag * tun_{\mu_{ag}}}$$

```
mubt      <- matrix(NA,1201,11)
TunmuAg <- PV["TunmuAg"]
RRmuAg <- exp((1:11)*TunmuAg)
for(i in 1:11) {
  mubt[,i] <- SmoCurve(BgMort[,i+1])*PV["TunMubt"]/12
  mubt[,i] <- mubt[,i]*RRmuAg[i]
}
```

Limit on Background Mortality

Finally, there is a mortality ceiling in the model for the two oldest age groups (ages 85+). This is operationalized as a restriction on the product: $RR_{\mu_{rg}} RR_{\mu_{rf}}$. Currently the mortality ceiling is a discontinuous function f that takes value:

$$RR_{\mu_{rg}} RR_{\mu_{rf}} < 5, f = RR_{\mu_{rg}} RR_{\mu_{rf}}$$
$$RR_{\mu_{rg}} RR_{\mu_{rf}} > 5, f = 5$$

Tuberculosis Specific Mortality

Tuberculosis mortality rate μ_{tb} is dependent on age and disease status. μ_{tb} is assumed to be 0 for all TB health states except for active TB with ("Tx") and without ("Ac") treatment. TB mortality increases log-linearly with age. These rates have been parameterized based on the smear positive values from the old model.

```
vTMort   <- matrix(0,11,6);
 rownames(vTMort) <- c("0_4",paste(0:8*10+5,1:9*10+4,sep="_"),"95p")
 colnames(vTMort) <- c("Su","Sp","Ls","Lf","Ac","Tx")
 vTMort[,5:6] <- muIp #active disease rates default to smear positive
 RRmuTbAg <- exp(c(0,0,1:9)*TunmuTbAg)
 for(i in 1:ncol(vTMort)) {
   vTMort[,i] <- vTMort[,i] * RRmuTbAg
 }
```

The reduction in TB mortality due to treatment is defined as the scalar:

$$(1 - TxEff)^{TunTxMort}$$

$TunTxMort$ is parameterized with value 0.5 [.2,.8].

Below is the total mortality for an individual in TB treatment (assuming $RR_{\mu_{rg}} RR_{\mu_{rf}} < 5$):

```
// where TxVec[1] is Treatment Effectiveness
mubtN[t][ag]*RRmuRFN[nm]*RRmuHR[rg]+vTMortN[ag][5 ]*pow(1.0-TxVecZ[1],TunTxMort);
```

'Rebalancing' the Mortality Distribution

In order to preserve our mortality structure in the model, at each time step, the population is 'rebalanced' back to the original marginal distribution across the four mortality risk groups. This process is operationalized with a matrix of transition probabilities from/to each distinct combination of mortality risk group m and TB progression risk group p. At each time step a transition can only occur between neighboring states, i.e. a transition from m_1p_2 to m_3p_2 is not permissable. However, it should be noted that even though the marginal mortality distribution is being rebalanced, transitions across TB progression risk groups are allowed but not tracked, specifically, m_1p_2 to m_2p_3 is a non-zero transition probability.

Below is the construction of the matrix of allowable transitions in the model; all matrix values that equal one represent the possible transition from its row state to its column state. This matrix shares its form with the matrix of transition probabilities (discussed below); it is printed here in an attempt to bring clarity to the structure of the transition matrix.

```
can_go <- matrix(0,16,16)
rownames(can_go) <- colnames(can_go) <- paste0(rep(paste0("p",0:3),each=4),
                                               "_",rep(paste0("m",0:3),4))

#define can go matrix
for(i in 1:nrow(can_go)){ # i=1
  pi <- rep(0:3,each=4)[i]; mi <- rep(0:3,4)[i]
  can_go[i, rep(0:3,each=4)%in%(pi + -1:1)  & rep(0:3,4)%in%(mi + -1:1)] <- 1
}
can_go
```

```
##       p0_m0 p0_m1 p0_m2 p0_m3 p1_m0 p1_m1 p1_m2 p1_m3 p2_m0 p2_m1 p2_m2
## p0_m0     1     1     0     0     1     1     0     0     0     0     0
## p0_m1     1     1     1     0     1     1     1     0     0     0     0
## p0_m2     0     1     1     1     0     1     1     1     0     0     0
## p0_m3     0     0     1     1     0     0     1     1     0     0     0
## p1_m0     1     1     0     0     1     1     0     0     1     1     0
## p1_m1     1     1     1     0     1     1     1     0     1     1     1
## p1_m2     0     1     1     1     0     1     1     1     0     1     1
## p1_m3     0     0     1     1     0     0     1     1     0     0     1
## p2_m0     0     0     0     0     1     1     0     0     1     1     0
## p2_m1     0     0     0     0     1     1     1     0     1     1     1
## p2_m2     0     0     0     0     0     1     1     1     0     1     1
## p2_m3     0     0     0     0     0     0     1     1     0     0     1
## p3_m0     0     0     0     0     0     0     0     0     1     1     0
## p3_m1     0     0     0     0     0     0     0     0     1     1     1
## p3_m2     0     0     0     0     0     0     0     0     0     1     1
## p3_m3     0     0     0     0     0     0     0     0     0     0     1
##       p2_m3 p3_m0 p3_m1 p3_m2 p3_m3
## p0_m0     0     0     0     0     0
## p0_m1     0     0     0     0     0
## p0_m2     0     0     0     0     0
## p0_m3     0     0     0     0     0
## p1_m0     0     0     0     0     0
## p1_m1     0     0     0     0     0
## p1_m2     1     0     0     0     0
## p1_m3     1     0     0     0     0
## p2_m0     0     1     1     0     0
## p2_m1     0     1     1     1     0
## p2_m2     1     0     1     1     1
## p2_m3     1     0     0     1     1
## p3_m0     0     1     1     0     0
## p3_m1     0     1     1     1     0
## p3_m2     1     0     1     1     1
## p3_m3     1     0     0     1     1
```

This transition matrix is calculated prior to the model start for each age group. We estimate the total mortality in each age group in a single time step:

```
    for (int nm=0; nm<4; nm++){
      for (int im=0; im<4; im++){

          if ((ag<9) & ((RRmuRF[nm]*RRmuHR)<5)){
            dist_t1_v[nm+im*4]=dist_gen_v[nm+im*4]*(1-((mubtN[ag]*RRmuRF[nm]*RRmuHR*HRdist[ag])));

          } else {
            dist_t1_v[nm+im*4]=dist_gen_v[nm+im*4]*(1-((mubtN[ag]*5*HRdist[ag])));
} } } }
```

and then calculate the transitions necessary to recapture the mortality distributions.

Steps to calculating the transition matrix:

- calculate the difference between the current distribution and the initial (goal) distribution.
- construct a 16x16 matrix (of the same structure as can_go above)
- if the difference between two states is positive and the transition is allowable (as described above), the corresponding matrix element (row =from; col=to) is set to this difference
- adjust the transition rates upward based on the current distribution and then ensure they do not sum over 1.
- calculate the total fraction of each state that is transitioning outward
- set the diagonal of the transition matrix equal to the remaining non-transitioning fraction for each state: $1 - frc_{trans}$.
- update the current distribution
- loop over this process until the distribution and goal distribution are equal

This process is repeated for each age group, since mortality is dependent on age.

At each time step in the model run, these matrices of rebalancing probabilties are applied to the population. There are two adjustment factors that are calibrated in order to ensure that we reproduce the mortality distribution, particularly in more recent years. They are used to create a single vector of age group dependent adjustment factors that are applied in the calculation of the age specific transition matrices. The equation for this adjustment factor can be found below.

```
  adj_fact <- exp(PV[["adj_ag1"]]*(10:0)/11 + PV[["adj_ag11"]]*(0:10)/11)
```