2016_CD4Case.R

jeanette

Tue Sep 13 14:06:18 2016

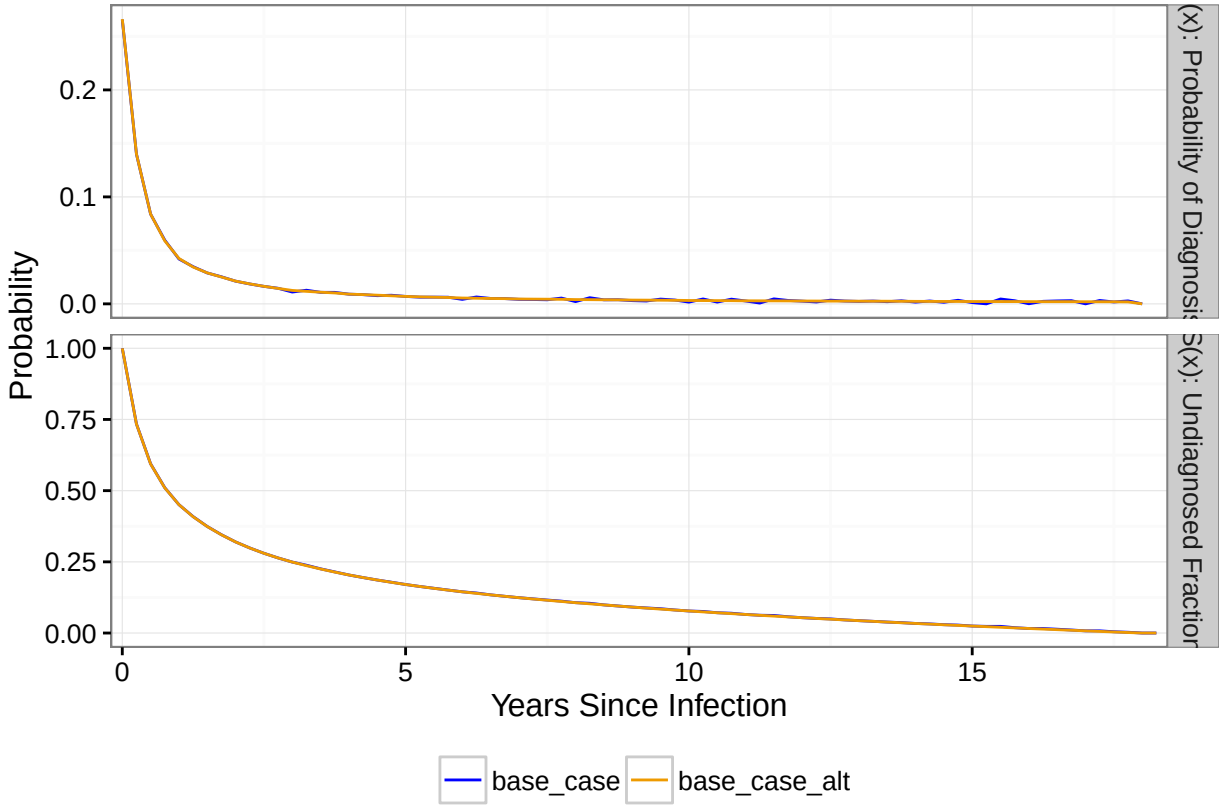
```
standalone <- FALSE

if (standalone) {
    rm(list=ls())
    # TEMPORARY: SOURCE FUNCTIONS
    source('/Users/jeanette/Dropbox/School/PhD/HIV_WA/HIVBackCalc/R/internal_fxns.R')

    # Change year min and max
    year_min <- 2005
    year_max <- 2014

    # Load libraries, data and data-cleaning file
    # Eventually this function should return the cleaned dataset,
    # but data-cleaning has the name hardcoded as msm and I'm not
    # going to generalize that right now
    setup_hivbackcalc(workd='/Users/jeanette/Dropbox/School/PhD/HIV_WA',
                      datafile='data/wa_backcalc_data_201602.csv',
                      source_these='analysis_WA/format_data.R',
                      packagefile='HIVBackCalc/R/internal_fxns.R')

    library(xtable)
    library(gridExtra)
    library(plyr)
    library(reshape2)
    library(ggplot2)
}

######################################################################
# Proof of equivalence between Base Case and Base Case Alt (Continuous)
```

```
bcVbcalt <- estimateTID(dataf$infPeriod,
                        intLength=0.25,
                        cases=c('base_case','base_case_alt'))
Sx_tab <- summary(bcVbcalt, times=c(0,0.25,0.5, 1,5,18), intLength=0.25)[,c(1,3,5)]
colnames(Sx_tab) <- c('Time', 'Original Base Case', 'Alternate Base Case')
print(xtable(Sx_tab,
             caption='Base Case TIDs using different computational approaches',
             label='tab:Sx_bcAlt',
             digits=3),
      caption.placement='top',
      table.placement='ht',
      size='small',
      include.rownames=FALSE)
```

```
## % latex table generated in R 3.3.0 by xtable 1.8-2 package
## % Tue Sep 13 14:06:18 2016
## \begin{table}[ht]
```

```
## \centering
## \caption{Base Case TIDs using different computational approaches}
## \label{tab:Sx_bcAlt}
## \begingroup\small
## \begin{tabular}{rrr}
##   \hline
## Time & Original Base Case & Alternate Base Case \\
##   \hline
## 0.000 & 0.734 & 0.734 \\
##   0.250 & 0.594 & 0.594 \\
##   0.500 & 0.510 & 0.510 \\
##   1.000 & 0.409 & 0.408 \\
##   5.000 & 0.164 & 0.164 \\
##   18.000 & 0.000 & 0.000 \\
##    \hline
## \end{tabular}
## \endgroup
## \end{table}
```

```
plot(bcVbcalt, intLength=0.25)
```



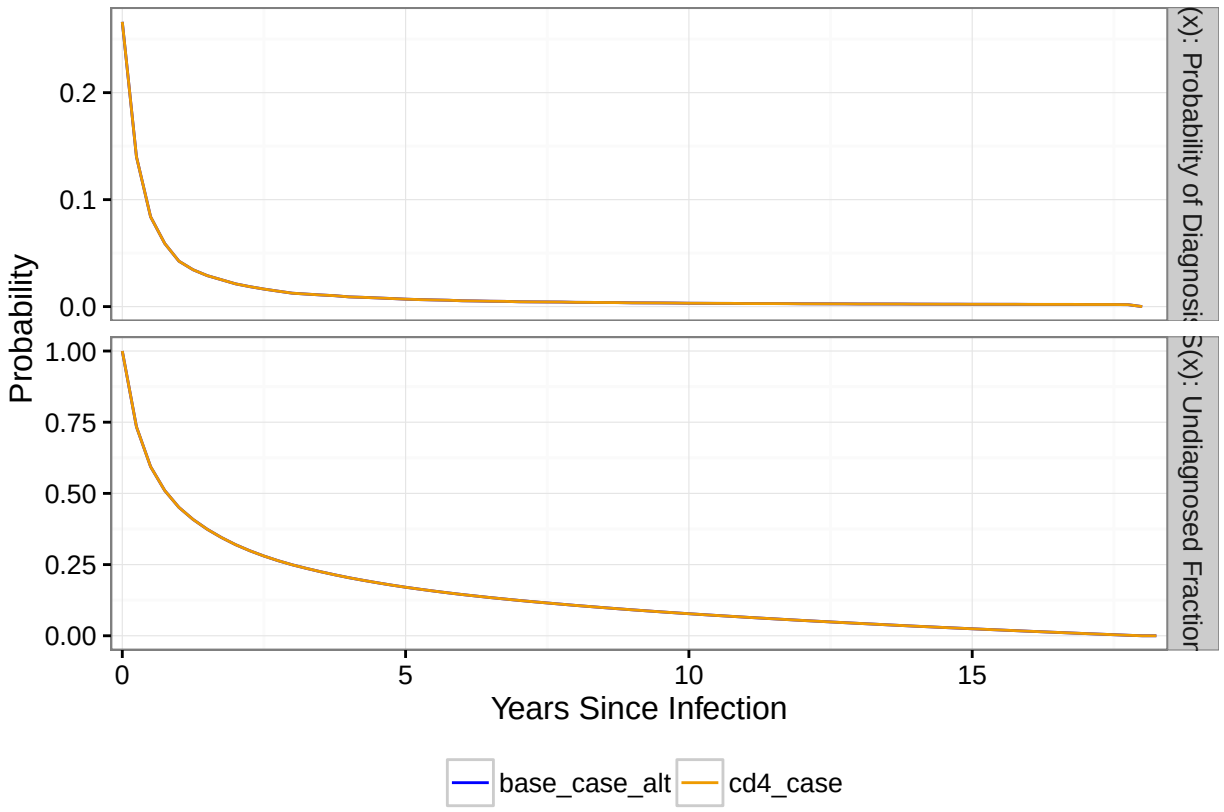

```
######################################################################
# Proof of equivalence between Fake CD4 Case and Base Case Continuous
```

```
cd4fake <- estimateTID(dataf$infPeriod,
                       intLength=0.25,
                       cases=c('base_case_alt','cd4_case'),
```

```
                      medWindows=dataf$infPeriod/2,
                      infPeriodOrig=dataf$infPeriod)
Sx_tab <- summary(cd4fake, times=c(0,0.25,0.5, 1,5,18),
                  intLength=0.25)[,c(1,3,5)]
colnames(Sx_tab) <- c('Time', 'Alternative Base Case', 'Fake CD4 Case')
print(xtable(Sx_tab,
             caption='Base Case versus Fake CD4 Case TIDs',
             label='tab:Sx_cd4fake',
             digits=3),
      caption.placement='top',
      table.placement='ht',
      size='small',
      include.rownames=FALSE)
```

```
## % latex table generated in R 3.3.0 by xtable 1.8-2 package
## % Tue Sep 13 14:06:20 2016
## \begin{table}[ht]
## \centering
## \caption{Base Case versus Fake CD4 Case TIDs}
## \label{tab:Sx_cd4fake}
## \begingroup\small
## \begin{tabular}{rrr}
##   \hline
## Time & Alternative Base Case & Fake CD4 Case \\
##   \hline
## 0.000 & 0.734 & 0.734 \\
##   0.250 & 0.594 & 0.594 \\
##   0.500 & 0.510 & 0.510 \\
##   1.000 & 0.408 & 0.408 \\
##   5.000 & 0.164 & 0.164 \\
##   18.000 & 0.000 & 0.000 \\
##    \hline
## \end{tabular}
## \endgroup
## \end{table}
```

```
plot(cd4fake, intLength=0.25)
```




```
#################################################################
# Setting up real CD4-based medians
```

```
# Define our literature-based median times to infection by CD4 bin
(cd4meds <- data.frame(cd4lower=c(500,350,200),
                       cd4upper=c(2000, 500, 350),
                       medWindow=c(1.5, 4, 8))
)
```

```
##   cd4lower cd4upper medWindow
## 1      500     2000       1.5
## 2      350      500       4.0
## 3      200      350       8.0
```

```
#**********************
# Define who should get a CD4-based median
cd4breaks <- c(0,200,350,500,2000)
windowbreaks <- c(0,3,8,16,18)

dataf <- within(dataf, {
            # Non-missing testing history
            hasTestHist <- !is.na(everHadNegTest)
            # CD4 measured within 30d
            cd4within30 <- hasTestHist & !is.na(cd4_days) & cd4_days<=30 &
                !is.na(firstcd4cnt)
            # Categories
            cd4cat=cut(firstcd4cnt, breaks=cd4breaks,
```

```
                  include.lowest=TRUE, right=FALSE)
          })
with(dataf, table(hasTestHist))
```

```
## hasTestHist
## FALSE  TRUE
##  2132  3016
```

```
with(dataf, table(cd4within30))
```

```
## cd4within30
## FALSE  TRUE
##  2970  2178
```

```
#*********************
# Assign medians

# Start with 1/2 of infPeriod, which is just the Base Case.
# Update to CD4-based median if indicated by infPeriod (infection window)
# Define our literature-based median times to infection by CD4 bin
cd4meds <- data.frame(cd4lower=c(500,350,200),
                      cd4upper=c(2000, 500, 350),
                      medWindow=c(1.5, 4, 8))

#*********************
# Assign medians

# Start with 1/2 of infPeriod, which is just the Base Case.
# Update to CD4-based median if indicated by infPeriod (infection window)
dataf <- transform(dataf, medWindows=infPeriod/2, impacted=0)

for (i in 1:nrow(cd4meds)) {
  dataf <- transform(dataf, temp=cd4within30 &
                                  firstcd4cnt>=cd4meds[i,'cd4lower'] &
                                  firstcd4cnt<cd4meds[i, 'cd4upper'] &
                                  infPeriod>=2*cd4meds[i, 'medWindow'])
    dataf <- transform(dataf, impacted=ifelse(temp==1,1,impacted))
    dataf <- within(dataf, {
                      medWindows[hasTestHist & cd4within30 &
                                  firstcd4cnt>=cd4meds[i,'cd4lower'] &
                                  firstcd4cnt<cd4meds[i, 'cd4upper'] &
                                  infPeriod>=2*cd4meds[i, 'medWindow']] <-
                                  cd4meds[i,'medWindow']
              })
}

# Was expecting 296 cases impacted; need to find the 6
with(dataf, sum(medWindows!=infPeriod/2, na.rm=TRUE))
```

```
## [1] 290
```

```r
# Ok
with(dataf, table(mode2, impacted))
```

```
##          impacted
## mode2        0    1
##   MSM     3232  171
##   non-MSM 1620  125
```

```r
with(dataf, table(mode2, impacted)/rowSums(table(mode2,impacted)))
```

```
##          impacted
## mode2              0          1
##   MSM     0.94975022 0.05024978
##   non-MSM 0.92836676 0.07163324
```

```r
# Now look among the 3016 with testing history
with(subset(dataf,!is.na(everHadNegTest)), table(mode2, impacted))
```

```
##          impacted
## mode2        0    1
##   MSM     2098  171
##   non-MSM  622  125
```

```r
with(subset(dataf,!is.na(everHadNegTest)), table(mode2, impacted)/rowSums(table(mode2,impacted)))
```

```
##          impacted
## mode2             0         1
##   MSM     0.9246364 0.0753636
##   non-MSM 0.8326640 0.1673360
```

```r
# Show old and new median windows AMONG the 3016 contributing to testing histories
ddply(subset(dataf,!is.na(everHadNegTest)), .(mode2,cd4cat), summarise,
      N_impacted=sum(impacted),
      avgOldMedian=round(mean(infPeriod/2, na.rm=TRUE),1),
      avgNewMedian=round(mean(medWindows, na.rm=TRUE),1),
      Difference=avgOldMedian-avgNewMedian)
```

```
##      mode2      cd4cat N_impacted avgOldMedian avgNewMedian Difference
## 1      MSM     [0,200)          0          3.9          3.9        0.0
## 2      MSM   [200,350)         24          1.9          1.9        0.0
## 3      MSM   [350,500)         35          1.4          1.2        0.2
## 4      MSM [500,2e+03]        112          1.3          0.8        0.5
## 5      MSM        <NA>          0          1.0          1.0        0.0
## 6  non-MSM     [0,200)          0          6.0          6.0        0.0
## 7  non-MSM   [200,350)         34          4.4          4.2        0.2
## 8  non-MSM   [350,500)         31          3.9          2.8        1.1
## 9  non-MSM [500,2e+03]         60          3.0          1.8        1.2
## 10 non-MSM        <NA>          0          2.8          2.8        0.0
```

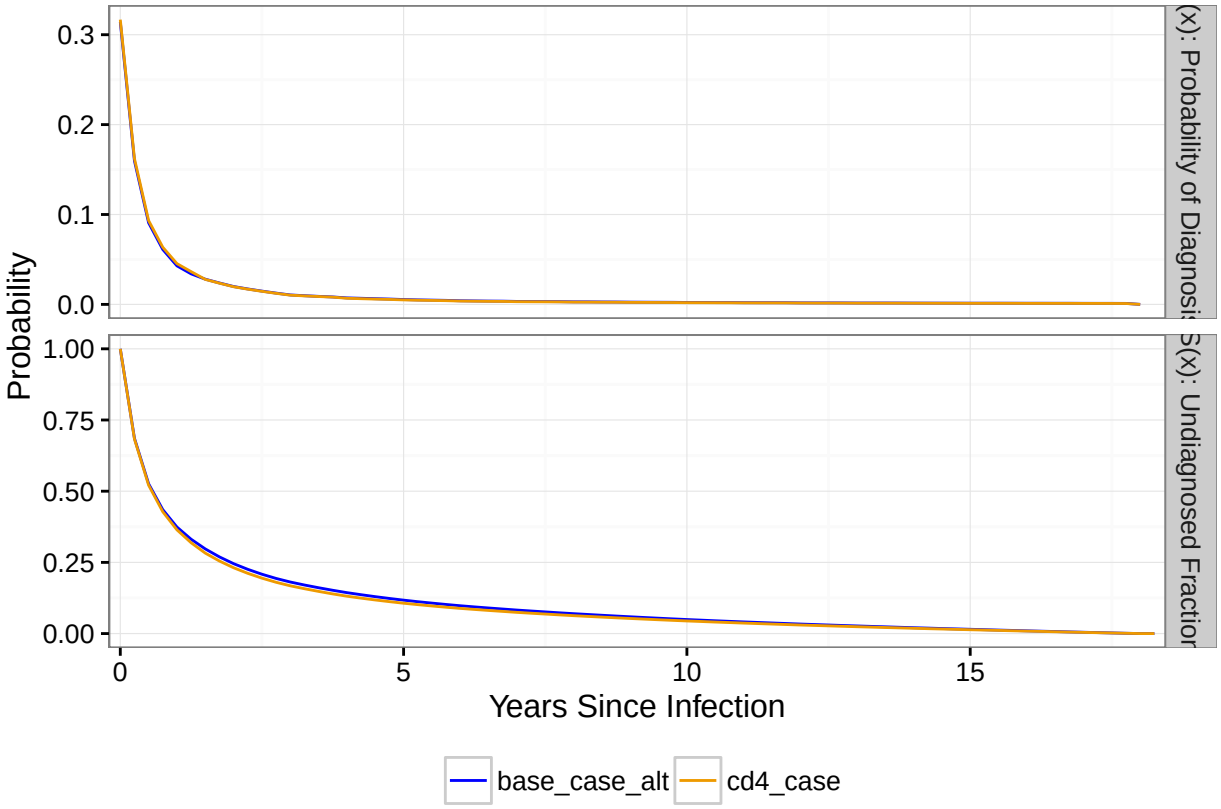
```
################################################################
# Estimate TIDs
```

```
cd4real <- estimateTID(dataf$infPeriod,
                       intLength=0.25,
                       cases=c('base_case_alt','cd4_case'),
                       medWindows=dataf$medWindows,
                       infPeriodOrig=dataf$infPeriod)
cd4real.MSM <- estimateTID(subset(dataf, mode2=='MSM')$infPeriod,
                       intLength=0.25,
                       cases=c('base_case_alt','cd4_case'),
                       medWindows=subset(dataf, mode2=='MSM')$medWindows,
                       infPeriodOrig=subset(dataf, mode2=='MSM')$infPeriod)
cd4real.nonMSM <- estimateTID(subset(dataf, mode2!='MSM')$infPeriod,
                       intLength=0.25,
                       cases=c('base_case_alt','cd4_case'),
                       medWindows=subset(dataf, mode2!='MSM')$medWindows,
                       infPeriodOrig=subset(dataf, mode2!='MSM')$infPeriod)

Sx_tab <- summary(cd4real, times=c(0,0.25,0.5, 1,5,18),
                  intLength=0.25)[,c(1,3,5)]
Sx_tab2 <- summary(cd4real.MSM, times=c(0,0.25,0.5, 1,5,18),
                  intLength=0.25)[,c(1,3,5)]
Sx_tab3 <- summary(cd4real.nonMSM, times=c(0,0.25,0.5, 1,5,18),
                  intLength=0.25)[,c(1,3,5)]
Sx_tab <- data.frame(Pop=c('All', rep('', nrow(Sx_tab)-1),
                           'MSM', rep('', nrow(Sx_tab)-1),
                           'non-MSM', rep('', nrow(Sx_tab)-1)),
                     rbind(Sx_tab, Sx_tab2, Sx_tab3))
colnames(Sx_tab) <- c('Population', 'Time',
                      'Alternative Base Case', 'CD4 Case')
print(xtable(Sx_tab,
             caption='Base Case versus CD4 Case TIDs',
             label='tab:cd4real_tab',
             digits=3),
      caption.placement='top',
      table.placement='ht',
      size='small',
      include.rownames=FALSE)
```

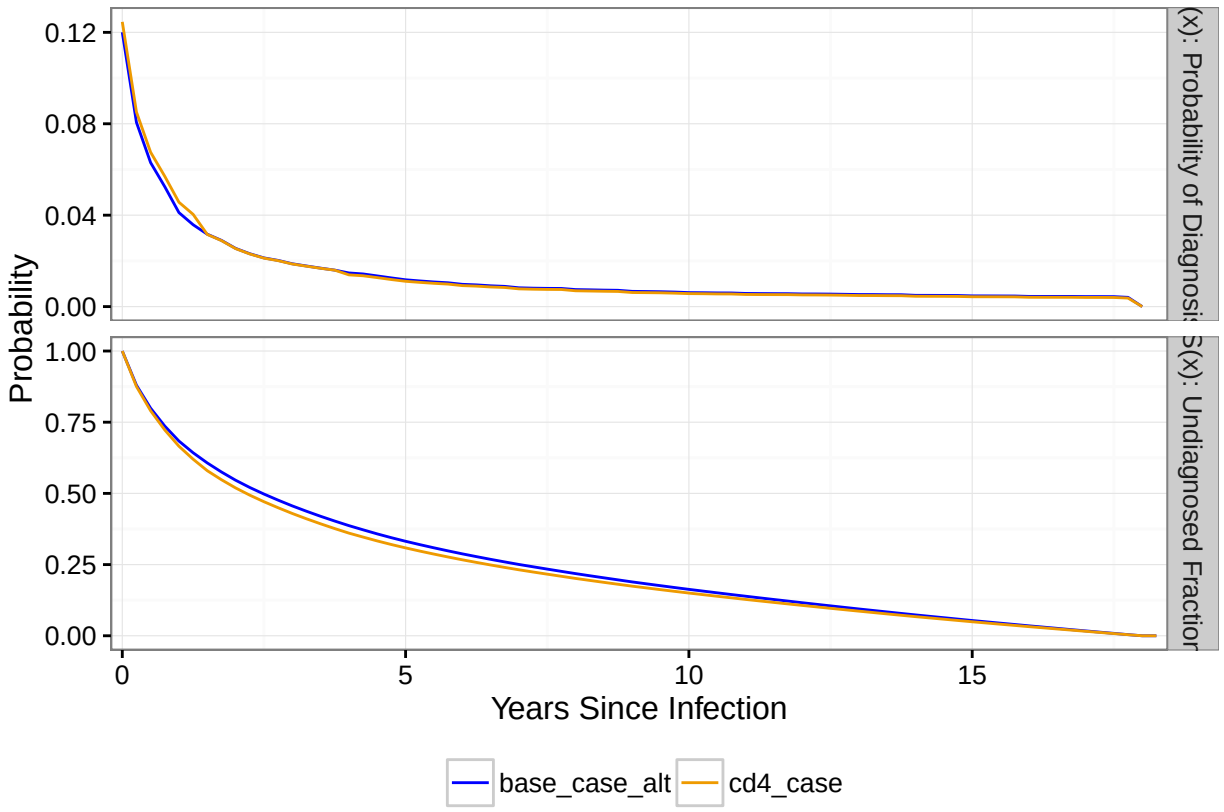
```
## % latex table generated in R 3.3.0 by xtable 1.8-2 package
## % Tue Sep 13 14:06:21 2016
## \begin{table}[ht]
## \centering
## \caption{Base Case versus CD4 Case TIDs}
## \label{tab:cd4real_tab}
## \begingroup\small
## \begin{tabular}{lrrr}
##   \hline
## Population & Time & Alternative Base Case & CD4 Case \\
##   \hline
## All & 0.000 & 0.734 & 0.731 \\
##    & 0.250 & 0.594 & 0.588 \\
```

```
##     & 0.500 & 0.510 & 0.501 \\
##     & 1.000 & 0.408 & 0.394 \\
##     & 5.000 & 0.164 & 0.150 \\
##     & 18.000 & 0.000 & 0.000 \\
##   MSM & 0.000 & 0.686 & 0.683 \\
##     & 0.250 & 0.526 & 0.521 \\
##     & 0.500 & 0.435 & 0.428 \\
##     & 1.000 & 0.331 & 0.319 \\
##     & 5.000 & 0.112 & 0.102 \\
##     & 18.000 & 0.000 & 0.000 \\
##   non-MSM & 0.000 & 0.880 & 0.875 \\
##     & 0.250 & 0.799 & 0.790 \\
##     & 0.500 & 0.736 & 0.723 \\
##     & 1.000 & 0.643 & 0.620 \\
##     & 5.000 & 0.320 & 0.297 \\
##     & 18.000 & 0.000 & 0.000 \\
##     \hline
## \end{tabular}
## \endgroup
## \end{table}
```

```
plot(cd4real.MSM, 0.25)
```




```
plot(cd4real.nonMSM, 0.25)
```




```
#################################################################
# Investigate probability reassigned and impact on  TIDs
```

```
# Compute BC probability assigned within the median window:
# just 1/infPeriod times the medWindow. Then compare that to
# 0.5, which is how much the CD4 Case assigs within the median window

dataf <- within(dataf, {
                  medProbBC=(medWindows)*(1/infPeriod)
                  probReassigned=0.5-medProbBC
      })

summary(dataf$probReassigned)
```

```
##     Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##  0.0000  0.0000  0.0000  0.0218  0.0000  0.4166    2132
```

```
ddply(subset(dataf,!is.na(everHadNegTest)), .(mode2), summarise,
      totalReassigned=sum(probReassigned, na.rm=TRUE),
      propReassigned=sum(probReassigned)/length(probReassigned))
```

```
##     mode2 totalReassigned propReassigned
## 1     MSM        39.32961     0.01733345
## 2 non-MSM        26.55218     0.03554508
```

```
# Look separately among impacted cases
ddply(subset(dataf,!is.na(everHadNegTest) & impacted==1), .(mode2), summarise,
      totalReassigned=sum(probReassigned),
      propReassigned=sum(probReassigned)/length(probReassigned))
```

```
##     mode2 totalReassigned propReassigned
## 1     MSM        39.32961      0.2299977
## 2 non-MSM        26.55218      0.2124174
```

```
timeStep <- 0.25
yearTimes <- seq(0,18,by=timeStep)

# Get full TID curves
msmSx <- summary(cd4real.MSM, times=yearTimes, intLength=0.25)
msmSx$mode='MSM'
nonmsmSx <- summary(cd4real.nonMSM, times=yearTimes, intLength=0.25)
nonmsmSx$mode='non-MSM'
FullSx <- rbind(msmSx, nonmsmSx)

# Multiply to get discrete AUC
means <- ddply(FullSx, .(mode), summarise,
      bc_auc=0.25*sum(`base_case_alt S(x)`),
      cd4_auc=0.25*sum(`cd4_case S(x)`))
means <- transform(means, ratio=cd4_auc/bc_auc, diff=bc_auc-cd4_auc)
print(means, digits=2)
```

```
##      mode bc_auc cd4_auc ratio diff
## 1     MSM    1.8     1.7  0.94 0.11
## 2 non-MSM    4.4     4.1  0.94 0.25
```

```
#####################################################################
# Prepare for estimation
```

```
# Read in true prevalence
  trueprev_data = read.csv(file.path(workd,'data/Reported_prevalence_2010-2014.csv'),
                           na.string="",
                           stringsAsFactor=FALSE,
                           check.names=FALSE)


#############################################################
# Estimate undiagnosed cases
```

```
  these_cases <- c('base_case_alt', 'cd4_case')
  names(these_cases) <- c('Base Case', 'CD4 Case')
  subgroups <- runSubgroups(dataf,
                            subvar='mode2',
                            intLength=1,
                            cases=these_cases,
                            medWindowsVar='medWindows',
                            prev=trueprev_data,
                            save=file.path(workd, 'analysis_WA/results/2016_trueprev_CD4Case.csv'))
```

```
##
## SUBGROUP:  MSM
##
## Estimating case base_case_alt ...
##
## Estimating case cd4_case ...
## Is it here???
## Is this too late???
##
## SUBGROUP:  non-MSM
##
## Estimating case base_case_alt ...
##
## Estimating case cd4_case ...
## Is it here???
## Is this too late???
```

```
# Function to extract desired comparative results
  compareUndx <- function(x, subgroups, name='') {
      totRes <- subgroups[[x]]$results

      # Summary of summaries
      sumtable <- subset(totRes$resultsSummary[order(totRes$resultsSummary$Estimate),],
                         Estimate=='Undiagnosed Cases')
      sumtable2 <- subset(totRes$resultsSummaryYear[order(totRes$resultsSummaryYear$Estimate),],
                          Estimate=='Undiagnosed Cases')
      sumtable$Year <- '2005-2014'
      sumtable <- rbind(sumtable, sumtable2)[,c('Year', colnames(sumtable)[-ncol(sumtable)])]
      colnames(sumtable)[which(colnames(sumtable)=='Diagnoses/Case')] <- 'Case'
      m <- subset(melt(sumtable), variable=='Mean', select=-Estimate)
      mwide <- dcast(m, Year~Case, value.var='value')
      mwide$Difference <- mwide[,2]-mwide[,3]
      mwide <- transform(mwide, `Percent Change`=round(100*Difference/`Base Case`),
                         check.names=FALSE)
      return(data.frame(Group=name,mwide,check.names=FALSE))
  }

# Combined comparison of undiagnosed estimates
compareAll <- rbind(compareUndx('Total-stratified', subgroups, 'Total'),
                    compareUndx('MSM', subgroups, 'MSM'),
                    compareUndx('non-MSM', subgroups, 'non-MSM'))
```

```
## Using Year, Case, Estimate as id variables
```

```
## Using Year, Subgroup, Case, Estimate as id variables
## Using Year, Subgroup, Case, Estimate as id variables
```

```
# Compare undiagnosed fractions
compareFrac  <- function(x, subgroups, name='') {
    totTP <- subgroups[[x]]$trueprev
    totTP <- rename(totTP, c('Diagnoses/Case'='Case'))
    totTP <- subset(totTP, Estimate==unique(Estimate)[2] |
                    Estimate==unique(Estimate)[4],
```

```
                select=c('Year', 'Case', 'Estimate', 'Mean'))
    mwide <- dcast(totTP, Year+Estimate~Case, value.var='Mean')
    mwide$Difference <- mwide[,3]-mwide[,4]
    mwide <- transform(mwide, `Percent Change`=round(100*Difference/`Base Case`),
                       check.names=FALSE)
    return(data.frame(Group=name,mwide,check.names=FALSE))
}

compareAll <- rbind(compareFrac('Total-stratified', subgroups, 'Total'),
                    compareFrac('MSM', subgroups, 'MSM'),
                    compareFrac('non-MSM', subgroups, 'non-MSM'))
```