Using R for Markov modelling: an introduction

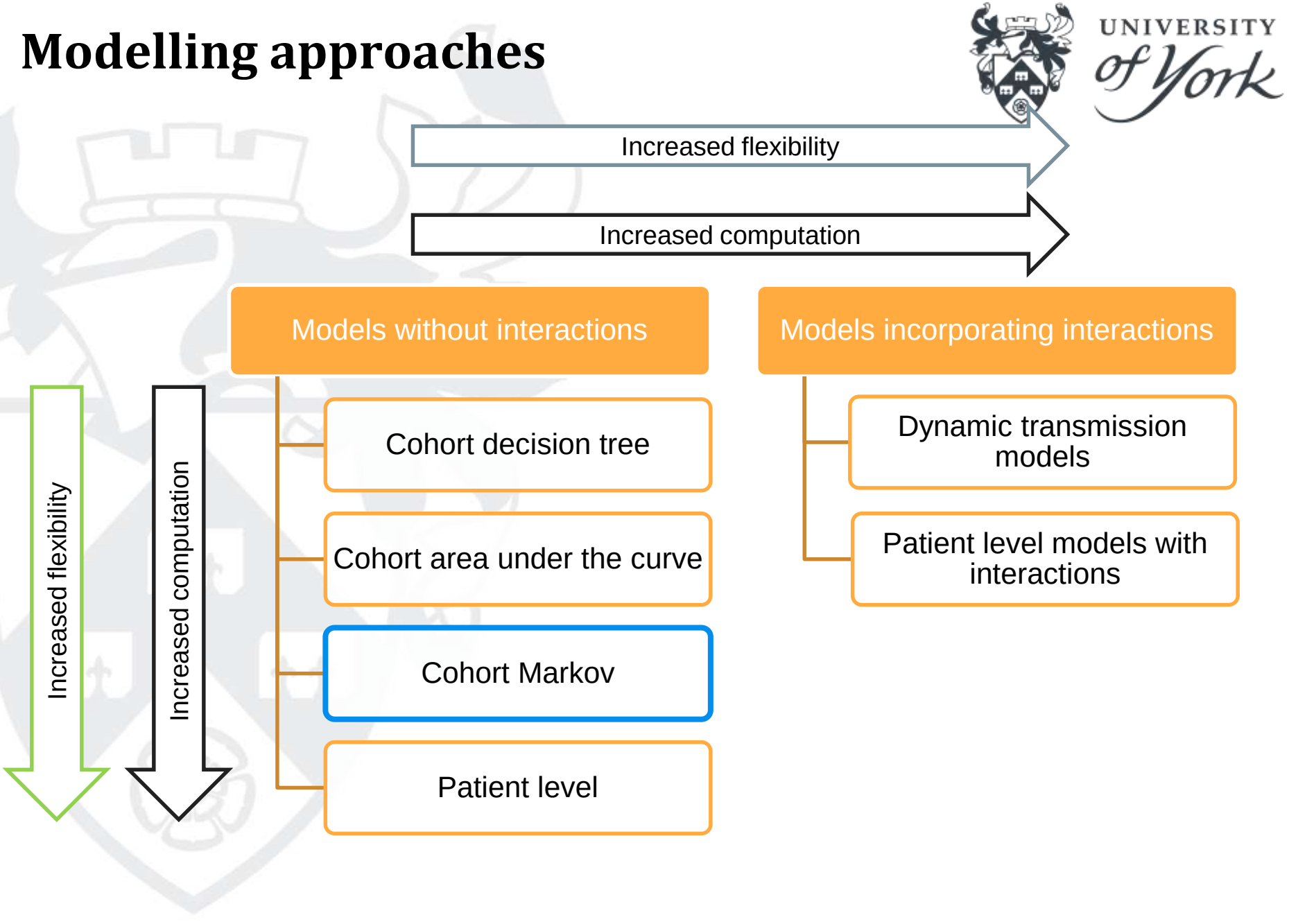
Marta Soares, Centre for Health Economics, University of York

11 July 2018

Decision modelling for cost-effectiveness

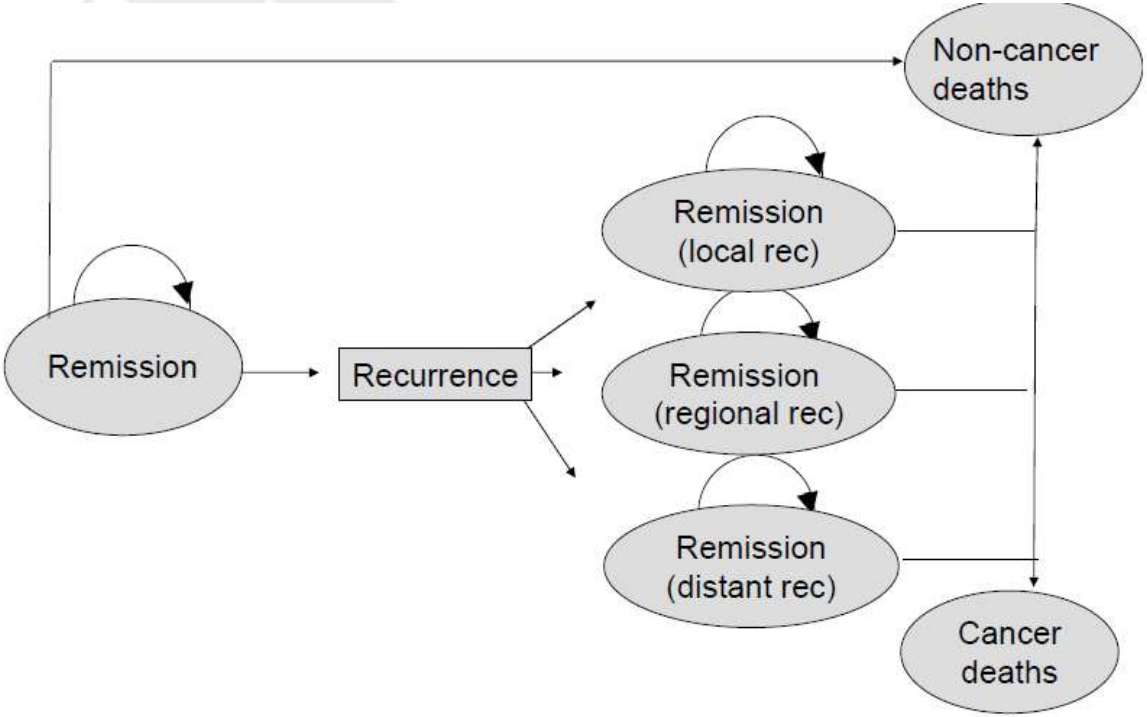
- Health systems need to make a number of decisions on health care resources: which, whom, when, where…
  - Typical example is reimbursement/access (NICE)
  - Other examples: individual funding requests (NHS England)
- Decisions informed by HTA including cost-effectiveness, the latter comparing interventions in terms of:
  - long-term effects on population health (typically measured in QALYs), and
  - overall cost implications for relevant stakeholders or individuals.
- Key advantages of decision model-based evaluations:
  - evidence from multiple sources and on multiple aspects of disease and treatment are considered together, and
  - allows principled extrapolation to the long term
  - explicit consideration of uncertainty

Modelling approaches



Cohort Markov models

- Defined as:
  - mutually exclusive and discrete number of states (absorbent state = death)
  - discrete time
  - Markov property



Cohort Markov models

- Speed of movement between the states defined based on transition probabilities

- Matrix of transition probabilities for a single cycle

From \ To	Asymptomatic	Progressive	Dead
Asymptomatic	0.666	0.167	0.167
Progressive	0	0.500	0.500
Dead	0	0	1.000

Each row sums to 1.0

- Markov models are evaluated by repeatedly applying these transition probabilities to a cohort of patients over time to generate a Markov trace

Matrix multiplication formulation

- Vector of starting states, $sm_{t0} = [x_1 \quad x_2 \quad x_3]$

- Transition probability matrix $\mathrm{TP} = \begin{bmatrix} p_{1,1} & p_{1,2} & p_{1,3} \\ p_{2,1} & p_{2,2} & p_{2,3} \\ p_{3,1} & p_{3,2} & p_{3,3} \end{bmatrix}$

- To find state membership at t1 multiply the vector and the matrix:

$$sm_{t1} = sm_{t0} \cdot \mathrm{TP} =$$
$$[(x_1 \cdot p_{1,1} + x_2 \cdot p_{2,1} + x_3 \cdot p_{3,1}) \quad (x_1 \cdot p_{1,2} + x_2 \cdot p_{2,2} + x_3 \cdot p_{3,2}) \quad (x_1 \cdot p_{1,3} + x_2 \cdot p_{2,3} + x_3 \cdot p_{3,3})]$$

- This process is applied repeatedly

$$sm_{t2} = sm_{t1} \cdot \mathrm{TP}$$
$$sm_{t3} = sm_{t2} \cdot \mathrm{TP}$$

Markov modelling in R

```
TP <- matrix(data = c(2/3,1/6,1/6,0,1/2,1/2,0,0,1), nrow = 3, byrow = TRUE)
#TP[1,2] <- 0.5 * TP[1,2]
#TP[1,3] <- 1-sum(TP[1,1:2])
costs <- c(500,200,0) # +c(100,0,0)
hrqol <- c(0.8, 0.3, 0)
cycles <- 4
```

```
> TP
          [,1]      [,2]      [,3]
[1,] 0.6666667 0.1666667 0.1666667
[2,] 0.0000000 0.5000000 0.5000000
[3,] 0.0000000 0.0000000 1.0000000
```

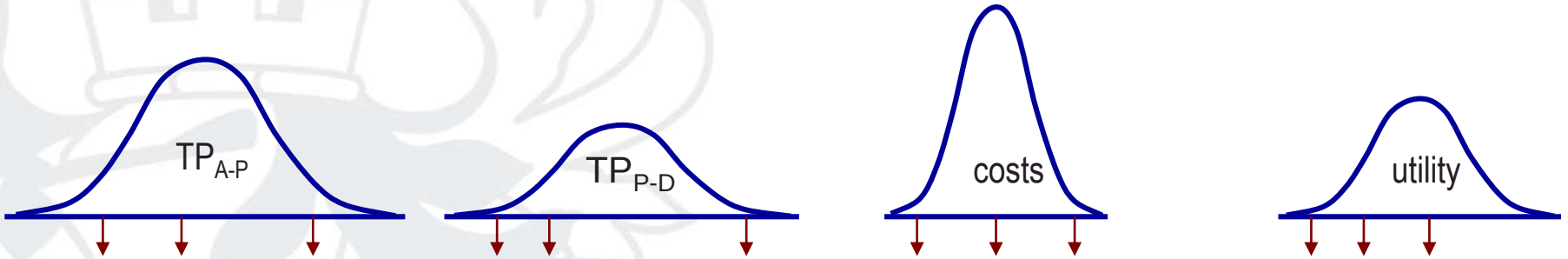
```
trace <- matrix(data = NA, nrow = cycles+1, ncol = 3)
trace[1,] <- c(1,0,0)
for (i in 1:cycles){
  trace[i+1,] <- trace[i,] %*% TP
}
```

```
> trace
          [,1]      [,2]      [,3]
[1,] 1.0000000 0.0000000 0.0000000
[2,] 0.6666667 0.1666667 0.1666667
[3,] 0.4444444 0.1944444 0.3611111
[4,] 0.2962963 0.1712963 0.5324074
[5,] 0.1975309 0.1350309 0.6674383
```

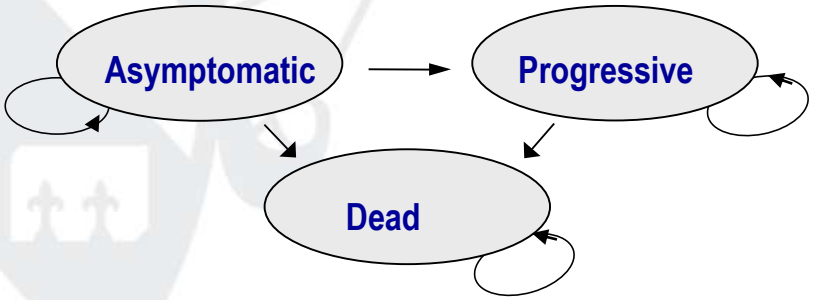
```
sum(trace[2:5,] %*% costs)
sum(trace[2:5,] %*% hrqol)
sum(trace[2:5,1:2])
```

```
> sum(trace[2:5,] %*% costs) [1] 935.9568
> sum(trace[2:5,] %*% hrqol) [1] 1.484182
> sum(trace[2:5,1:2])        [1] 2.272377
```

Decision uncertainty: Monte Carlo simulation



1.	0.31		0.4		£10,802		0.62	
2.		0.46		0.2		£5,205		0.81
3.		0.64		0.9		£7,381		0.69



	costs	effects
1	£20,400	2
2	£14,745	1.8
3	£18,364	1.1

EVPI

How things could turn out	(Net) Health			Best we could do if we knew
	Treatment A	Treatment B	Best choice	
θ1	8	12	B	12
θ2	16	8	A	16
θ3	9	14	B	14
θ4	12	10	A	12
θ5	10	16	B	16
Average	**11**	**12**		**14**

What's the best we can do now?

- Choose B
  - Expect 12 QALYs, gain 1 QALY
- But uncertain
  - Wrong decision 2/5 times (error probability = 0.4)

Could we do better?

- If we knew
  - Expect 14 QALYs

$$EVPI = E_\theta \max_j NB(j, \theta) - \max_j E_\theta NB(j, \theta) = 2 \text{ QALYs per patient}$$

Markov modelling in R
PSA

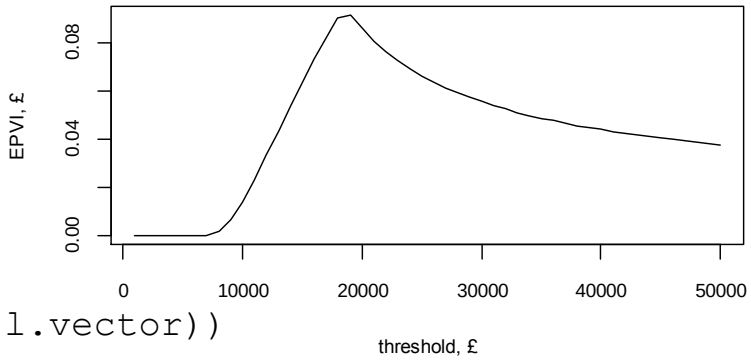
```
# enclose model in a function

model <- function(index,  treat=0, spar=mat_par, cycl=50) {
  TP <- matrix(data=0, 3,3)
  TP[1,2] <- spar[index,"p1"]*spar[index,"p2"] * (exp(logRR[index])^treat)
  TP[1,3] <- spar[index,"p1"]*(1-spar[index,"p2"])
  TP[2,3] <- spar[index,3]
  diag(TP) <- 1-apply(TP,1,sum)

  trace <- matrix(data = NA, nrow = cycl+1, ncol = 3)
  trace[1,] <- c(1,0,0)
  for (i in 1:cycl){ trace[i+1,] <- trace[i,]%*%TP }

  c(sum(trace[2:(cycl+1),] %*% (costs+c(500,0,0)^treat)),
  sum(trace[2:(cycl+1),] %*% hrqol))
}
```

Markov modelling in R
PSA

```
# define probabilistic inputs
nPSA=2000

p1     <- rbeta(2000, 20, 40)         # prob[X2>1|X1=1]
p2     <- rbeta(2000, 500,500)       # prob[X2=2|X2>1,X1=1]
tp23   <- rbeta(2000, 50,50)
logRR  <- rnorm(2000, -0.1,.3)
hrqol2 <- rbeta(2000, 2,8)

# define a matrix
mat_par     <- cbind("p1"=p1,"p2"=p2,"tp23"=tp23, "logRR"=logRR,
"hrqol2"=hrqol2)
```

Markov modelling in R
PSA, EVPI

```
# run probabilistic model using mapply

aux<- rep(c(0,1),nPSA)
r.PSA <- t(mapply(model, index=rep(2:(nPSA+1),each=2), treat=aux))
nb <- NB(r.PSA )
nb <- as.data.frame(split(nb, aux))
colMeans(nb)
sum(nb[,2]>nb[,1])/nPSA
```




```
EVPI <- vector("numeric", length=length(l.vector))
l.vector <- seq(1000,50000, by=1000)
for (k in 1:length(l.vector)) {
    nb <- NB(r.PSA, l=l.vector[k] )
    nb <- as.data.frame(split(nb, aux))
  EVPI[k] <- mean(c(nb[max.col(nb)==1,1],nb[max.col(nb)==2,2])) -
max(colMeans(nb))
}
```

Is further evidence worth collecting? (EVPI)

- Value of eliminating uncertainty in all parameters
- Maximum return to research (decision problem)
- Comparing the EVPI to the costs of research
- Comparing popEVPI across technologies, or subgroups

What type of evidence? (EVPPI)

$$EVPI_{\theta 1} = E_{\theta 1} \max_j E_{\theta 2|\theta 1} NB(j, \theta 1, \theta 2) - \max_j E_{\theta} NB(j, \theta)$$

θ
- $\theta 1$ = parameter of interest
- $\theta 2$ = other uncertainties

- Value of eliminating uncertainty in a subset of input parameters
- Useful to identify which parameters responsible for decision uncertainty
- Helps target research designs

Markov modelling in R
EVPPI (Monte Carlo)

```
fun_EVPPI <- function(p_EVPPI, nEVPPI=500){
  EVPPI_aux <- vector("list", nEVPPI); EVPPI <- matrix(NA,nrow=nEVPPI, ncol=2)
  mat_aux <- mat_par
  for (m in 1:nEVPPI) {
    mat_aux[,p_EVPPI] <- rep(mat_par[m, p_EVPPI], each=nPSA)
    re <- data.frame(t(mapply(model, index=rep(1:nPSA,each=2), treat=rep(c(0,1),nPSA),
MoreArgs=list(spar=mat_aux))))
    re <- split(re,rep(c(0,1),nPSA))
    EVPPI_aux[[m]] <- sapply(re,NB)
    EVPPI[m,] <- colMeans(EVPPI_aux[[m]]); colnames(EVPPI) <- names(re)
  }; return(list("EVPPI"=EVPPI,"param"=p_EVPPI, "data" = EVPPI_aux))
}
param.lst.EVPPI <- list(colnames(mat_par)[1:3], "logRR‟,”hrqol2”))
system.time(EVPPI <- lapply(param.lst.EVPPI,fun_EVPPI))

a <- sapply(1:length(param.lst.EVPPI), function(i){
  mean(apply(EVPPI[[i]][["EVPPI"]],1,max)) - max(apply(EVPPI[[i]][["EVPPI"]],2,mean))
})
for (i in 1:length(a)){ print(paste(paste(EVPPI[[i]][["param"]], collapse=", "),":",
a[i]))}
```

```
[1] "p1, p2, tp23 : 0.001401"
[1] "logRR : 0.086443"
[1] "hrqol2 : 0"
```

Thanks!

Marta Soares, Centre for Health Economics,
University of York