```
TITLE Belmabrouk et al. 2011 LN
:hh.mod is changed based on Belmanbrouk et al. 2011
:Na, K, leak
:Manduca sexta Local interneuron
UNITS {
    (mA) = (milliamp)
    (mV) = (millivolt)
    (S) = (siemens)
    (molar) = (1/liter)
    (mM) = (millimolar)
}
? interface
NEURON {
    SUFFIX MsLN
    USEION na READ ena WRITE ina
    USEION k READ ek WRITE ik

    NONSPECIFIC_CURRENT il
    RANGE gnabar, gkbar, gl, el, gna, gk
    :By park
    RANGE il
    RANGE m, n, h
    :RANGE minf, ninf, hinf, mtau, ntau, htau
    RANGE ma,mb,na,nb,ha,hb, vv
    GLOBAL minf, hinf, ninf, mtau, htau, ntau
    THREADSAFE : assigned GLOBALs will be per thread
}
PARAMETER {
    gnabar = 0.190 (S/cm2)    <0,1e9>
    gkbar = 0.060 (S/cm2)    <0,1e9>
    gl = .0001 (S/cm2)    <0,1e9>

    el = -67 (mV)

}
STATE {
    m h n

}
ASSIGNED {
    v (mV)
    celsius (degC)
    ena (mV)
    ek (mV)
    gna (S/cm2)
    gk (S/cm2)
    ina (mA/cm2)
    ik (mA/cm2)
    il (mA/cm2)

    minf hinf ninf
    mtau (ms) htau (ms) ntau (ms)

    ma
    mb
    na
    nb
    ha
    hb
    vv
}
? currents
BREAKPOINT {
    SOLVE states METHOD cnexp
    gna = gnabar*m*m*m*h
    ina = gna*(v - ena)
    :ina = 0
    gk = gkbar*n*n*n*n
    ik = gk*(v - ek)
    :ik = 0
    il = gl*(v - el)
    :il = 0
}
INITIAL {
    rates(v)
    m = minf
    h = hinf
    n = ninf
}
? states
DERIVATIVE states {
    rates(v)
    m' = (minf-m)/mtau
    h' = (hinf-h)/htau
    n' = (ninf-n)/ntau

}
:LOCAL q10
? rates
PROCEDURE rates(v(mV)) { :Computes rate and other constants at current v.
    :Call once from HOC to initialize inf at resting v.
    LOCAL alpha, beta, sum, q10
    TABLE minf, mtau, hinf, htau, ninf, ntau DEPEND celsius FROM -100
    TO 100 WITH 200
UNITSOFF
    q10 = 3^((celsius - 6.3)/10)
    :"m" sodium activation system
    alpha = 0.32 * vtrap((v+54),4)
    beta = 0.28 * vtrap2((v+27),5)
    sum = alpha + beta
    mtau = 1/sum
    minf = alpha/sum

    ma = alpha
    mb = beta

    :"n" potassium activation system
    alpha = 0.032 * vtrap((v+52),5)
    beta = 0.5 * exp(-(v+57)/40)
    sum = alpha + beta
    ntau = 1/sum
    ninf = alpha/sum

    na = alpha
    nb = beta

    :"h" sodium inactivation system
    alpha = 0.128*exp(-(v+50)/18)
    beta = 4/(1+exp(-(v+27)/5))
    sum = alpha + beta
    htau = 1/sum
    hinf = alpha/sum

    ha = alpha
    hb = beta

    vv = v

}

FUNCTION vtrap(x,y) { :Traps for 0 in denominator of rate eqns.
    if (fabs(x/y) < 1e-6) {
        vtrap = y/(1 - x/y/2)
    }else{
        vtrap = x/(1-exp(-x/y))
    }
}
FUNCTION vtrap2(x,y) { :Traps for 0 in denominator of rate eqns.
    if (fabs(x/y) < 1e-6) {
        vtrap2 = y/(1 + x/y/2)
    }else{
        vtrap2 = x/(exp(x/y) - 1)
```

```
143         }
144     }
145     UNITSON
end
```