

```

1  TITLE Belmabrouk et al. 2011 LN
2  :hh.mod is changed based on Belmanbrouk et al. 2011
3  :Na, K, leak
4  :Manduca sexta Local interneuron
5  UNITS {
6      (mA) = (milliamp)
7      (mV) = (millivolt)
8      (S) = (siemens)
9      (molar) = (1/liter)
10     (mM) = (millimolar)
11 }
12 ? interface
13 NEURON {
14     SUFFIX MsLN
15     USEION na READ ena WRITE ina
16     USEION k READ ek WRITE ik
17
18     NONSPECIFIC_CURRENT il
19     RANGE gnabar, gkbar, gl, el, gna, gk
20     :By park
21     RANGE il
22     RANGE m, n, h
23     :RANGE minf, ninf, hinf, mtau, ntau, htau
24     RANGE ma,mb,na,nb,ha,hb, vv
25     GLOBAL minf, hinf, ninf, mtau, htau, ntau
26     THREADSAFE : assigned GLOBALs will be per thread
27 }
28 PARAMETER {
29     gnabar = 0.190 (S/cm2) <0,1e9>
30     gkbar = 0.060 (S/cm2) <0,1e9>
31     gl = .0001 (S/cm2) <0,1e9>
32
33     el = -67 (mV)
34 }
35
36 STATE {
37     m h n
38 }
39
40 ASSIGNED {
41     v (mV)
42     celsius (degC)
43     ena (mV)
44     ek (mV)
45     gna (S/cm2)
46     gk (S/cm2)
47     ina (mA/cm2)
48     ik (mA/cm2)
49     il (mA/cm2)
50
51     minf hinf ninf
52     mtau (ms) htau (ms) ntau (ms)
53
54     ma
55     mb
56     na
57     nb
58     ha
59     hb
60     vv
61 }
62 ? currents
63 BREAKPOINT {
64     SOLVE states METHOD cnexp
65     gna = gnabar*m*m*m*h
66     ina = gna*(v - ena)
67     :ina = 0
68     gk = gkbar*n*n*n
69     ik = gk*(v - ek)
70     :ik = 0
71     il = gl*(v - el)

```

```

72     :il = 0
73 }
74 INITIAL {
75     rates(v)
76     m = minf
77     h = hinf
78     n = ninf
79 }
80 ? states
81 DERIVATIVE states {
82     rates(v)
83     m' = (minf-m)/mtau
84     h' = (hinf-h)/htau
85     n' = (ninf-n)/ntau
86 }
87
88 :LOCAL q10
89 ? rates
90 PROCEDURE rates(v(mV)) { :Computes rate and other constants at current v.
91     :Call once from HOC to initialize inf at resting v.
92     LOCAL alpha, beta, sum, q10
93     TABLE minf, mtau, hinf, htau, ninf, ntau DEPEND celsius FROM -100
94     TO 100 WITH 200
95     UNITSOFF
96     q10 = 3^((celsius - 6.3)/10)
97     : "m" sodium activation system
98     alpha = 0.32 * vtrap((v+54),4)
99     beta = 0.28 * vtrap2((v+27),5)
100    sum = alpha + beta
101    mtau = 1/sum
102    minf = alpha/sum
103
104    ma = alpha
105    mb = beta
106
107    : "n" potassium activation system
108    alpha = 0.032 * vtrap((v+52),5)
109    beta = 0.5 * exp(-(v+57)/40)
110    sum = alpha + beta
111    ntau = 1/sum
112    ninf = alpha/sum
113
114    na = alpha
115    nb = beta
116
117    : "h" sodium inactivation system
118    alpha = 0.128*exp(-(v+50)/18)
119    beta = 4/(1+exp(-(v+27)/5))
120    sum = alpha + beta
121    htau = 1/sum
122    hinf = alpha/sum
123
124    ha = alpha
125    hb = beta
126
127    vv = v
128 }
129
130
131 FUNCTION vtrap(x,y) { :Traps for 0 in denominator of rate eqns.
132     if (fabs(x/y) < 1e-6) {
133         vtrap = y/(1 - x/y/2)
134     }else{
135         vtrap = x/(1-exp(-x/y))
136     }
137 }
138 FUNCTION vtrap2(x,y) { :Traps for 0 in denominator of rate eqns.
139     if (fabs(x/y) < 1e-6) {
140         vtrap2 = y/(1 + x/y/2)
141     }else{
142         vtrap2 = x/(exp(x/y) - 1)

```

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