

The “nmodl_preprocessor”

- Rewrites nmodl files to run up to 15% faster and use 15% less memory
- Same equations, written slightly differently
- Developed to work with NEURON

Written by David McDougall

Hard-code the parameters and the temperature

Input File

```
PARAMETER {  
    celsius  
    temp = 23 (degC) : original temp  
}  
  
INITIAL {  
    Q10 = 3^((celsius - temp) / 10)  
}
```

Output File

```
PARAMETER {  
    celsius  
}  
  
INITIAL {  
    assert(celsius == 37.0)  
  
    Q10 = 3^(( 37.0(degC) - 23.0(degC) ) / 10)  
}
```

Hard-code the parameters and the temperature

- They can not change at run-time
- Hard-coded values have very fast run-time performance
 - Model runs approx 5% faster

Convert assigned variables into local variables

Input File

```
ASSIGNED {  
  rate_A  
  rate_B  
}  
  
PROCEDURE rates(v) {  
  rate_A = function_A(v)  
  rate_B = function_B(v)  
}  
  
KINETIC kin {  
  rates(v)  
  ~ A <-> B (rate_A, rate_B)  
}
```

Output File

```
ASSIGNED {  
}  
  
KINETIC kin {  
  LOCAL rate_A, rate_B  
  
  rate_A = function_A(v)  
  rate_B = function_B(v)  
  
  ~ A <-> B (rate_A, rate_B)  
}
```

Convert assigned variables into local variables

- First in-line all functions and procedures
- After in-lining there will be assigned variables that can now be rewritten as local variables.
- Assigned variables are stored in RAM memory,
Local variables are stored in CPU registers

Convert assigned variables into local variables

- Only private variables can be optimized away
 - By default assigned variables are private
 - Range and Global variables are public

Optimize Q10

Input File

```
ASSIGNED {  
    Q10  
}
```

```
INITIAL {  
    Q10 = 1.23456789  
}
```

```
BREAKPOINT {  
    data = Q10 * function(v)  
}
```

Output File

```
ASSIGNED {  
}
```

```
INITIAL {  
}
```

```
BREAKPOINT {  
    data = 1.23456789 * function(v)  
}
```

Optimize Q10

- First hard-code the parameters and temperature
- Then detect temperature adjustment factors and convert them into hard-coded values

Caveats

- Don't spam Range and Global variables
- VERBATIM statements can't be analysed which may prevent these optimizations
- This program has had limited testing

Questions?

- Try it today!

https://github.com/ctrl-z-9000-times/nmodl_preprocessor