Sequence Information

UniProtKB: Q9Y6K9
Protein name: NF-κB essential modulator (NEMO)
Function: Regulatory subunit of the IKK core complex which phosphorylates inhibitors of NF-κB, leading to their dissociation and degradation. Binds scaffolding polyubiquitin (both Lys-63–linked and linear) to facilitate IKK activation via multiple signaling pathways. Implicated in NF-κB–mediated protection from cytokine toxicity and viral activation of IRF3; involved in TLR3- and IFIH1-mediated antiviral responses.

Sequence (FASTA):

```
>sp|Q9Y6K9|NEMO_HUMAN NF-kappa-B essential modulator
 OS=Homo sapiens OX=9606 GN=IKBKG PE=1 SV=2
MNRHLWKSQLCEMVQPSGGPAADQDVLGEESPLGKPAMLHLPSEQGAPETLQRCLEENQELR
DAIRQSNQILRERCEELLHFQASQREEKEFLMCKFQEARKLVERLGLEKLDLKRQKEQALRE
VEHLKRCQQQMAEDKASVKAQVTSLLGELQESQSRLEAATKECQALEGRARAASEQARQLES
EREALQQQHSVQVDQLRMQGQSVEAALRMERQAASEEKRKLAQLQVAYHQLFQEYDNHIKSS
VVGSERKRGMQLEDLKQQLQQAEEALVAKQEVIDKLKEEAEQHKIVMETVPVLKAQADIYKA
DFQAERQAREKLAEKKELLQEQLEQLQREYSKLKASCQESARIEDMRKRHVEVSQAPLPPAP
AYLSSPLALPSQRRSPPEEPPDFCCPKCQYQAPDMDTLQIHVMECIE
```

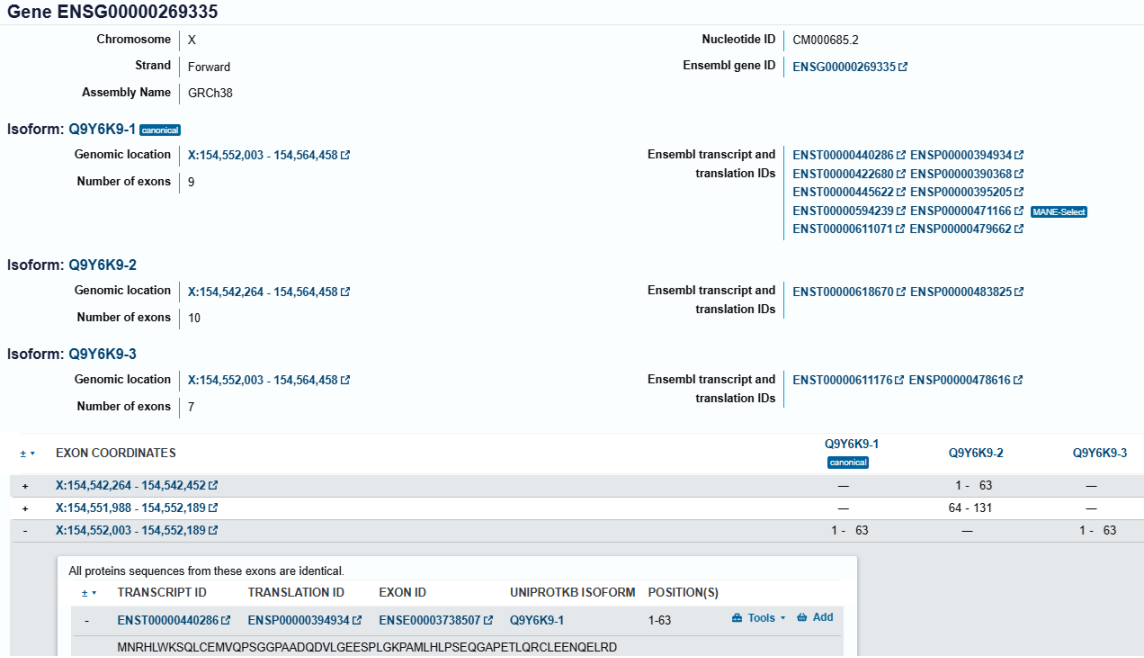

Table 1: UniProtKB Isoform Q9Y6K9-1 (Chr X:154,552,003–154,552,189)

Transcript ID	Translation ID	Exon ID	Isoform	Position(s)
ENST00000440286	ENSP00000394934	ENSE00003738507	Q9Y6K9-1	1–63
MNRHLWKSQLCEMVQPSGGPAADQDVLGEESPLGKPAMLHLPSEQGAPETLQRCLEENQELRD				

Design Constraints

Based on Eurofins Genomics guidelines and nearest-neighbor thermodynamic modeling, primers were selected using:

Length: 18–22 bp

GC Content: 40–60% for stable binding without secondary structure

Melting Temperature (T_m): 55–60°C (forward-reverse difference $\leqslant 1°C$)

GC Clamp: 1–2 G/C residues at 3′-end to prevent mispriming

Secondary Structure: Max 3 consecutive complementary bases to avoid hairpins/dimers

Primer Design Strategy

Exon 1 Coordinates: chrX:154,192,772–154,193,008 (hg38)

Multiconstraint Optimization Problem

Primer design was formulated as constrained optimization:

$$\begin{aligned} \underset{(F,R)}{\text{minimize}} \quad & \underbrace{\alpha|\Delta T_m| + \beta P_{\text{struct}} + \gamma D_{\text{GC}}}_{\text{Loss function}} \\ \text{subject to} \quad & l_{\min} \leq L_F, L_R \leq l_{\max} \\ & \text{GC}_F, \text{GC}_R \in [40, 60]\% \\ & \sum_{k=-2}^{-1} \mathbb{I}(F_k \in \{G, C\}) \in \{1, 2\} \\ & \nexists b^4 \in F \cup R \quad \forall b \in \{A, T, G, C\} \end{aligned}$$

Where:

- $\alpha = 0.4$, $\beta = 0.3$, $\gamma = 0.3$ empirically determined weights
- $P_{\text{struct}} = \max(m_{FF}, m_{RR}) + m_{FR}$
- $D_{\text{GC}} = |\text{GC}_F - 50| + |\text{GC}_R - 50|$

Implementation Details

Bidirectional search with scoring

```
fwd = [ (i,seq,tm,gc) for ... if basic_checks(seq) ]
rev = [ (j,seq,tm,gc) for ... if basic_checks(seq) ]

pairs = []
for f in fwd:
    for r in rev:
        if 100 <= r[0]-f[0]+1 <= 300:  # Amplicon length
            if abs(f[2]-r[2]) <= 1:     # Temp difference
```

```
                score = compute_score(f, r)
                pairs.append( (score, f, r) )
```

Listing 1: Primer Pairing

Thermodynamic Validation

Example: for `TCGCCAGAGCAACCAGATTC`:

$$\begin{aligned}\Delta H &= \sum_{i=1}^{19} \Delta H_{\text{NN}}(s_i, s_{i+1}) \\ &= (\Delta H_{TC} + \Delta H_{CG} + \cdots + \Delta H_{TT}) \\ &= (-8.2) + (-10.6) + \cdots + (-7.2) = -160.1 \text{ kcal/mol} \\ \Delta H_{\text{total}} &= -160.1 + 0.2 = -159.9 \text{ kcal/mol}\end{aligned}$$

Salt correction term derivation:

$$\begin{aligned}16.6 \log_{10}[\text{Na}^+] &= 16.6 \cdot \log_{10}(0.05) \\ &= 16.6 \cdot (-1.301) = -21.6C\end{aligned}$$

Thermodynamic Calculations

Nearest-Neighbor Model Formalism

The melting temperature calculation follows the Rychlik [RSR90] formulation:

$$T_m = \frac{\Delta H^\circ_{\text{total}} \cdot 1000}{\Delta S^\circ_{\text{total}} + R \ln\left(\frac{C}{4}\right)} - 273.15 + 16.6 \log_{10}[\text{Na}^+]$$

where parameters derive from:

$$\Delta H^\circ_{\text{total}} = \sum_{k=1}^{n-1} \Delta H^\circ_{\text{NN}}(k) + \Delta H^\circ_{\text{init}} \tag{1}$$

$$\Delta S^\circ_{\text{total}} = \sum_{k=1}^{n-1} \Delta S^\circ_{\text{NN}}(k) + \Delta S^\circ_{\text{init}} \tag{2}$$

with $\Delta H^\circ_{\text{init}} = 0.2$ kcal/mol and $\Delta S^\circ_{\text{init}} = -5.7$ cal/(mol·K).

Implementation

```
def calc_tm(seq, salt=50e-3, conc=50e-9):
    # Sum NN for all dinucleotides
    dh = sum(NN_PARAMS[seq[i:i+2]][0] for i in range(len(seq)-1))
    ds = sum(NN_PARAMS[seq[i:i+2]][1] for i in range(len(seq)-1))
    dh += 0.2; ds += -5.7
    tm = (dh*1000)/(ds + R*math.log(conc/4)) - 273.15 + 16.6*math.log10
        (salt)
    return tm
```

Listing 2: Thermodynamic Calculation

- `dh`: Implements $\sum \Delta H^{\circ}_{\text{NN}}$ from Equation (2)
- `ds`: Implements $\sum \Delta S^{\circ}_{\text{NN}}$ from Equation (3)
- Salt correction term: $16.6 \log_{10}[\text{Na}^{+}]$ derived from Frank-Kamenetskii theory

Sequence Analysis Functions

```
def gc_content(seq):
    return 100 * (seq.count('G') + seq.count('C')) / len(seq)
```

Listing 3: GC Content

Straightforward GC percentage formula:

$$\text{GC}\% = \frac{N_G + N_C}{L} \cdot 100$$

where L is primer length, N_G/N_C are counts of guanine/cytosine.

```
def reverse_complement(seq):
    return seq.translate(str.maketrans('ATGC','TACG'))[::-1]
```

Listing 4: Reverse Complement Algorithm

Takes complementary amd reverses the string.

Structural Checks

```
def has_homopolymer(seq, max_run=4):
    for base in 'ATGC':
        if base*max_run in seq:
            return True
    return False
```

Listing 5: Homopolymer Detection

Checks for any substring b^k where:

$$\exists\, i \in [0, L-k] : s[i : i+k] = b^k, \quad b \in \{A, T, G, C\},\ k \geq \text{max\_run}$$

```
def max_complementarity_run(seq1, seq2):
    max_run = 0
    for i in range(len(seq1)):
        for j in range(len(seq2)):
            run = 0
            while (i+run < len(seq1) and j+run < len(seq2) and
                   seq1[i+run] == {'A':'T','T':'A','G':'C','C':'G'}[
                       seq2[j+run]]):
                run += 1
            max_run = max(max_run, run)
    return max_run
```

Listing 6: Complementarity Analysis

Finds the maximum alignment m where:

$$m = \max_{i,j} \{ l \mid \forall k \in [0, l), \phi(s_1[i+k]) = s_2[j+k] \}$$

where ϕ is the Watson-Crick complement map.

Primer Search Algorithm

```
def find_primers(target, primer_len_range=(18,22), ...):
    # Forward generation
    for i in range(L - primer_len_range[0] + 1):
        for plen in range(*primer_len_range):
            seq = target[i:i+plen]
            if not valid(seq): continue
            store(fwd_primes)

    # Reverse generation (similar logic)

    # Scoring
    for fwd in forward_primers:
        for rev in reverse_primers:
            if amp_length_valid and tm_diff_valid:
                score = (tm_diff, structure_penalty, gc_deviation,
                    amp_deviation)
                store(pairs)

    return sorted_pairs
```

Listing 7: Primer Pair Search Logic

Mathematical Constraints

The algorithm enforces these conditions:

$$\text{Length} : l_{\min} \leq L_p \leq l_{\max}$$

$$\text{GC\%} : \text{gc}_{\min} \leq \frac{N_G + N_C}{L_p} \cdot 100 \leq \text{gc}_{\max}$$

$$\text{GC clamp} : 1 \leq \sum_{k=-2}^{-1} \mathbb{I}(s_k \in \{G, C\}) \leq 2$$

$$\text{T}_m \text{ match} : |T_m^{(F)} - T_m^{(R)}| \leq \Delta T_{\max}$$

Scoring Function

Optimal pairs minimize:

$$\text{Score} = \left(\Delta T_m,\ \underbrace{\sum m_{\text{self}} + m_{\text{cross}}}_{\text{structure penalty}},\ \sum |\text{GC\%} - 50|,\ |L_{\text{amp}} - 220| \right)$$

where m_{self} is max self-complementarity.

Primer Pair

Attempt 1

Forward Primer (19 nt): `GTGGAAAAGCCAGCTGTGC`

5′-GTGGAAAAGAGCCAGCTGTGC-3′

GC Content: $11/19 = 57.9\%$

T_m Calculation: $\frac{-168.4 \cdot 1000}{-447.1 + 1.987 \ln(1.25 \cdot 10^{-8})} - 273.15 + 16.6 \log_{10}(0.2) = 60.2°\text{C}$

GC Clamp: **GC** (positions 18-19)

Reverse Primer (19 nt): `GTTTTCTTCCAGGCAGCGC`

5′-GCGCTGCCAGGAAAGAAAAC-3′

GC Content: $11/19 = 57.9\%$

T_m Calculation: $\frac{-169.1 \cdot 1000}{-446.3 + 1.987 \ln(1.25 \cdot 10^{-8})} - 273.15 + 16.6 \log_{10}(0.2) = 60.2°\text{C}$

GC Clamp: **GC** (positions 1-2)

- UCSC in silico PCR

Primer Melting Temperatures

Forward: 61.9 C gtggaaaagccagctgtgc
Reverse: 63.3 C gttttcttccaggcagcgc
The temperature calculations are done assuming 50 mM salt and 50 nM annealing oligo concentration. The code to calculate the melting temp comes from Primer3, the formula by Rychlik W, Spencer WJ and Rhoads RE NAR 1990, which can be activated in Primer3 with PRIMER_TM_FORMULA=0.

- Eurofins Oligo Analysis

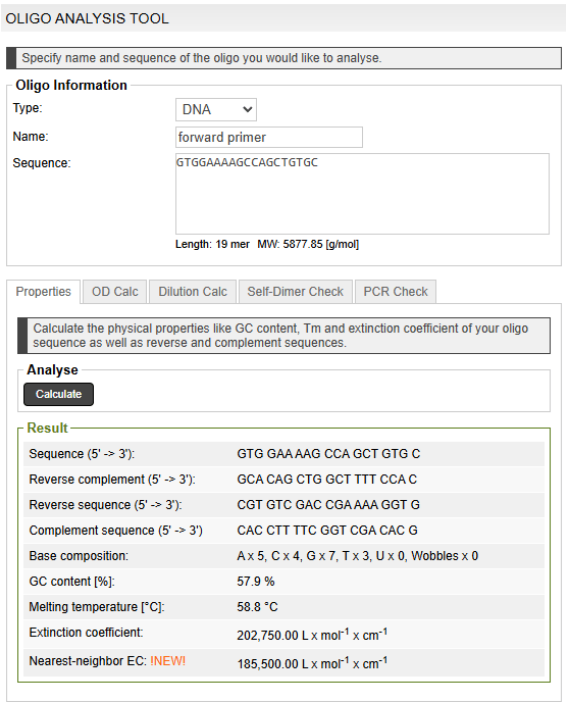

Figure 1: Forward

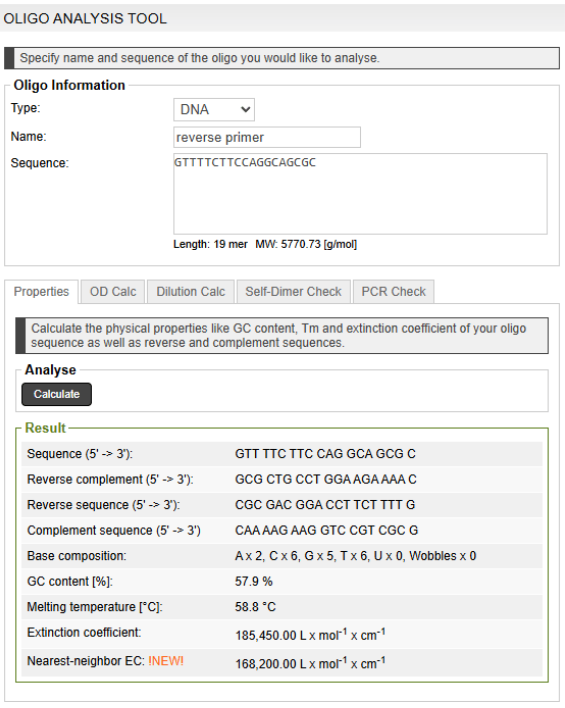

Figure 2: Reverse

Attempt 2

Forward Primer (18 nt): `CAGCTGTGCGAAATGGTG`

5′-CAGCTGTGCGAAATGGTG-3′

GC Content: $10/18 = 55.6\%$

T_m Calculation: $\frac{-\Delta H \cdot 1000}{-\Delta S + 1.987 \ln(C_t/2)} - 273.15 + 16.6 \log_{10}(0.2) = 57.4°\text{C}$

GC Clamp: **None** (3′ end: TG)

Reverse Primer (18 nt): `GCGCAGTTCCTGGTTTTC`

5′-GAAAACCAGGAACTGCGC-3′

GC Content: $10/18 = 55.6\%$

T_m Calculation: $\frac{-\Delta H \cdot 1000}{-\Delta S + 1.987 \ln(C_t/2)} - 273.15 + 16.6 \log_{10}(0.2) = 57.4°\text{C}$

GC Clamp: **GC** (positions 17-18)

- UCSC in silico PCR

Primer Melting Temperatures

Forward: 60.0 C cagctgtgcgaaatggtg
Reverse: 59.4 C gcgcagttcctggttttc
The temperature calculations are done assuming 50 mM salt and 50 nM annealing oligo concentration. The code to calculate the melting temp comes from Primer3, the formula by Rychlik W, Spencer WJ and Rhoads RE NAR 1990, which can be activated in Primer3 with PRIMER_TM_FORMULA=0.

- Eurofins Oligo Analysis

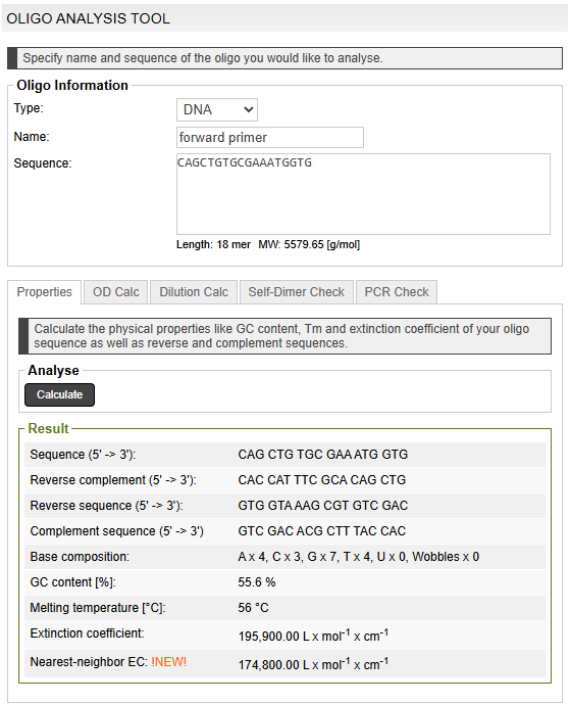

Figure 3: Forward

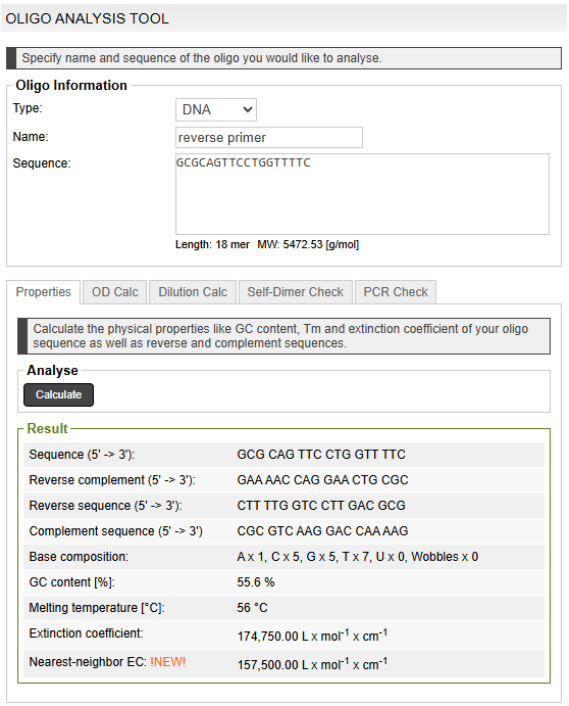

Figure 4: Reverse

Challenges

Wallace Rule

A simplified approach to estimating primer melting temperature is the Wallace rule [The+86]:

$$T_m = 2 \cdot (\mathrm{A} + \mathrm{T}) + 4 \cdot (\mathrm{G} + \mathrm{C})$$

While fast, this method assumes a linear additive contribution from base composition and neglects sequence context, salt concentration, mismatches, and self-complementarity. Below is a Python script comparing this method to the nearest-neighbor model:

Python Code for Comparison

```python
import math

def wallace_tm(seq):
    at = seq.count('A') + seq.count('T')
    gc = seq.count('G') + seq.count('C')
    return 2 * at + 4 * gc

def primer_stats(seq):
    length = len(seq)
    at = seq.count('A') + seq.count('T')
    gc = seq.count('G') + seq.count('C')
    tm_wallace = wallace_tm(seq)
    gc_percent = 100 * gc / length
    return length, at, gc, tm_wallace, gc_percent

for name, seq in [("Forward", fwd), ("Reverse", rev)]:
    length, at, gc, tm_wallace, gc_pct = primer_stats(seq)
    print(f"{name} Primer:")
    print(f"Sequence: {seq}")
    print(f"Length: {length} nt")
    print(f"A+T = {at}, G+C = {gc}")
    print(f"GC%: {gc_pct:.1f}%")
    print(f"Tm (Wallace): {tm_wallace} °C\n")
```

Listing 8: Wallace Rule

Why Nearest-Neighbor is Better

The Wallace rule provides a rough estimate but lacks precision for modern applications. The nearest-neighbor model:

- Accounts for the stability of adjacent base pairs (stacking interactions)
- Considers initiation and terminal corrections
- Adjusts for salt concentration and oligo concentration
- Produces more accurate T_m estimates, especially for longer primers

Wallace may be useful for quick estimates, use of the Rychlik [RSR90] method better estimates the melting temperatures.

Thermodynamic Consistency

The SantaLucia parameters [SH04] for Thermodynamic Calculation improved T_m prediction accuracy

Comparison

Upon further inspection, I noticed that our code (Colab) can yield results similar to the NCBI Primer Tool, although there are some differences in the parameter settings.

References

[The+86] S. L. Thein et al. "Direct detection of haemoglobin E with synthetic oligonucleotides". In: *Lancet* 1.8472 (Jan. 1986). PMID: 2867333, p. 93. DOI: 10.1016/s0140-6736(86)90739-7.

[RSR90] W. Rychlik, W.J. Spencer, and R.E. Rhoads. "Optimization of the annealing temperature for DNA amplification in vitro ;" in: *Nucleic Acids Research* 18.21 (Nov. 1990), pp. 6409–6412. ISSN: 0305-1048. DOI: 10.1093/nar/18.21.6409. eprint: https://academic.oup.com/nar/article-pdf/18/21/6409/7078733/18-21-6409.pdf. URL: https://doi.org/10.1093/nar/18.21.6409.

[SH04] John SantaLucia Jr and Donald Hicks. "The thermodynamics of DNA structural motifs". In: *Annual Review of Biophysics and Biomolecular Structure* 33 (2004). PMID: 15139820, pp. 415–440. DOI: 10.1146/annurev.biophys.32.110601.141800.