

Supplemental Material for

Democratizing the Rapid Screening of Protein Expression for Materials Development

Melody A. Morris, Rogério A. Bataglioli, Danielle J. Mai, Yun Jung Yang, Justin M. Paloni, Carolyn E. Mills, Zachary D. Schmitz, Erika A. Ding, Allison C. Huske, and Bradley D. Olsen*

Department of Chemical Engineering, Massachusetts Institute of Technology, 77 Massachusetts Avenue, Cambridge, Massachusetts 02139, United States

*Correspondence should be addressed to B.D.O. (bdolsen@mit.edu)

Table of Contents

A. DNA and amino acid sequences of constructs	2
B. DNA sequences of vectors	18
C. Sequencing primers	31
D. Sequence maps for vectors with mCherry	32
E. OT-2 automated expression codes	35
F. MATLAB robot analysis code (mCherry)	60
G. Assembled data for constructs	71
H. Robot-based assay supplemental data	105
I. Gantt chart of process for a single protein	106
J. Cost table for platform	109
K. Competent cell efficiencies	110
L. References	111

Supporting Information. DNA and amino acid sequences of constructs, DNA sequences of vectors, Sequencing primers, Sequence maps for vectors with mCherry, OT-2 automated expression codes, MATLAB analysis code, Assembled data for constructs, Robot-based assay supplemental data, Gantt chart of process for a single protein, Cost table for platform, Competent cell efficiencies, References

A. DNA and amino acid sequences of constructs

047A

DNA sequence:

ATGGGCAGCAGCCATCATCATCATCACAGCAGCGGCCTGGTGCCGCGCGGCAG
CCATATGGCTAGCAACCTGCCGGAAGTGAAAGATGGCACCTGCGTACCACCGTTA
TTGCGGATGGCGTGAATGGTAGCAGCGAAAAAGAGGCGCTGGTTAGCTTCGAAAAC
AGCAAGGATGGTGTGGACGTTAAAGATACCATCAACTACGAGGGCCTGGTTGCGAA
CCAGAACTATAACCCTGACCGGTACCCTGATGCACGTGAAGGCGGATGGCAGCCTGG
AGGAAATTGCGACCAAAACCACCAACGTTACCGCGGGTGAAAACGGTAACGGCACC
TGGGGTCTGGACTTCGGCAACCAGAAGCTGCAAGTTGGCGAGAAATACGTGGTTTTT
GAGAACGCGGAAAGCGTGGAATACTGATTGACACCGATAAGGACTATAACTCTAG
AGTGCCGGGTAGCGGCGTTCCGGGTAGCGGCGTGCTGGTAGCGGCGTTCCTGGTA
GCGGCGTGCCCGGTAGCGGCGTTCCTGGTAGCGGCGTGCCAGGTAGCGGCGTTCCTA
GGTAGCGGCGTGCCGGGTAGCGGCGTTCCTGGTAGCGGCGTGCCGGGTAGCGGCGT
TCCCGGTAGCGGCGTGCCAGGTAGCGGCGTTCCTGGTAGCGGCGTGCCAGGTAGCG
GCGTTCCTGGTAGCGGCGTGCCAGGTAGCGGCGTTCCTGGTAGCGGCGTGCCAGGT
AGCGGCGTTCAGGTAGCGGCGTGCCGGGTAGCGGCGTTCAGGTAGCGGCGTGCC
AGGTAGCGGCGTTCAGGTAGCGGCGTGCCAGGTAGCGGGAACCTGCCGGAAGTTA
AGGATGGTACCCTGCGCACCACCGTGATCGCGGACGGTGTGAACGGCAGCAGCGAG
AAAGAAGCGCTGGTGAGCTTCGAGAATAGCAAGGACGGTGTGGATGTTAAAGACAC
CATTAACTACGAGGGTCTGGTTGCGAACC AAAATTACACCCTGACCGGCACCCTGAT
GCATGTTAAAGCGGACGGCAGCCTGGAGGAAATCGCGACCAAGACCACCAATGTGA
CCGCGGGCGAGAATGGCAATGGTACCTGGGGTCTGGACTTTGGTAATCAGAAGCTG
CAAGTGGGCGAGAAATACGTGGTGTGTTGAGAATGCGGAAAGCGTGGAGAACCTGAT
TGATACCGATAAGGACTACAACACTAGAGTGCCGGGTAGCGGCGTTCCTGGGTAGCG
GCGTGCTGGTAGCGGCGTTCCTGGTAGCGGCGTGCCCGGTAGCGGCGTTCCTGGTA
GCGGCGTGCCAGGTAGCGGCGTTCAGGTAGCGGCGTGCCGGGTAGCGGCGTTCCT
GGTAGCGGCGTGCCGGGTAGCGGCGTTCCTGGTAGCGGCGTGCCAGGTAGCGGCGT
TCCGGGTAGCGGCGTGCCAGGTAGCGGCGTTCCTGGTAGCGGCGTGCCAGGTAGCG
GCGTTCCTGGTAGCGGCGTGCCAGGTAGCGGCGTTCAGGTAGCGGCGTGCCGGGT
AGCGGCGTTCAGGTAGCGGCGTGCCAGGTAGCGGCGTTCAGGTAGCGGCGTGCC
AGGTAGCGGGAACCTGCCGGAAGTTAAGGATGGTACCCTGCGCACCACCGTGATCG
CGGACGGTGTGAACGGCAGCAGCGAGAAAGAAGCGCTGGTGAGCTTCGAGAATAG
CAAGGACGGTGTGGATGTTAAAGACACCATTAACTACGAGGGTCTGGTTGCGAACC
AAAATTACACCCTGACCGGCACCCTGATGCATGTTAAAGCGGACGGCAGCCTGGAG
GAAATCGCGACCAAGACCACCAATGTGACCGCGGGCGAGAATGGCAATGGTACCTG
GGGTCTGGACTTTGGTAATCAGAAGCTGCAAGTGGGCGAGAAATACGTGGTGTGTTG
AGAATGCGGAAAGCGTGGAGAACCTGATTGATACCGATAAGGACTACAACACTAGT
CTCGAGGATCCGGCTGCTAACAAAGCCCGAAAGGAAGCTGAGTTGGCTGCTGCCAC
CGCTGAGCAATAA

Amino acid sequence:

[illegible]

047B

DNA sequence:

ATGGGAGGATCCCATATGGCTAGCGATACCAAGCAAGTTGTGAAGCATGAAGATAA
AAATGATAAGGCGCAGACCCTGGTTGTTGAGAAGCCGTCTAGAGTTCCGGGCAGCG
GTGTTCCGGGTAGCGGCGTTCCGGGTAGCGGCGTGCCTGGTAGCGGCGTTCTTGTA
GCGGCGTGCCCGGTAGCGGCGTTCTTGTTAGCGGCGTGCCAGGTAGCGGCGTTCCC
GGTAGCGGCGTGCCGGGTAGCGGCGTTCTTGTTAGCGGCGTGCCGGGTAGCGGCGT
TCCAGGTAGCGGCGTGCCAGGTAGCGGCGTTCCAGGTAGCGGCGTGCCAGGTAGCG
GCGTTCCAGGTAGCGGCGTGCCAGGTAGCGGCGTTCCAGGTAGCGGCGTGCCAGGT
AGCGGCGTTCCAGGTAGCGGCGTGCCAGGTAGCGGCGTTCCAGGTAGCGGCGTGCC
AGGTAGCGGCGTTCCAGGTAGCGGGGACACCAAGCAAGTGGTGAAGCACGAAGAC
AAAAATGATAAAGCGCAAACCCTGGTGGTGGAAAAGCCGACTAGAGTTCCGGGCAG
CGGTGTTCCGGGTAGCGGCGTTCCGGGTAGCGGCGTGCCTGGTAGCGGCGTTCTTG
TAGCGGCGTGCCCGGTAGCGGCGTTCTTGTTAGCGGCGTGCCAGGTAGCGGCGTTC
CCGGTAGCGGCGTGCCGGGTAGCGGCGTTCTTGTTAGCGGCGTGCCGGGTAGCGGC
GTTCCAGGTAGCGGCGTGCCAGGTAGCGGCGTTCCAGGTAGCGGCGTGCCAGGTAG
CGGCGTTCCAGGTAGCGGCGTGCCAGGTAGCGGCGTTCCAGGTAGCGGCGTGCCAG
GTAGCGGCGTTCCAGGTAGCGGCGTGCCAGGTAGCGGCGTTCCAGGTAGCGGCGTG
CCAGGTAGCGGCGTTCCAGGTAGCGGGGACACCAAGCAAGTGGTGAAGCACGAAG
ACAAAAATGATAAAGCGCAAACCCTGGTGGTGGAAAAGCCGACTAGTCTCGAGAGA
TCTCATCACCATCACCATCACTAA

Amino acid sequence:

MGGSHMASDTKQVVKHEDKNDKAQTLVVEKPSPRVPGSGVPGSGVPGSGVPGSGVPGS
GVPGSGVPGSGVPGSGVPGSGVPGSGVPGSGVPGSGVPGSGVPGSGVPGSGVPGS
SGVPGSGVPGSGVPGSGVPGSGVPGSGVPGSGVPGSGDPGSGVPGSGVPGSGVPGS
TLVVEKPTRVPGSGVPGSGVPGSGVPGSGVPGSGVPGSGVPGSGVPGSGVPGSG

Catcher

ATGGGCAGCAGCCATCATCATCATCACAGCAGCGGCCTGGTGCCGCGCGGCAG
CCATATGGCTAGCGCGATGGTTGATACCCTGAGCGGTCTGAGCAGCGAGCAAGGCC
AAAGCGGCGATATGACCATTGAGGAAGATAGCGCGACCCACATTAAGTTCAGCAAA
CGTGACGAGGATGGTAAAGAACTGGCGGGTGCGACCATGGAGCTGCGTGACAGCAG
CGGTAAAACCATCAGCACCTGGATTAGCGACGGCCAGGTTAAGGATTTCTACCTGTA
TCCGGGTAAATACACCTTTGTGGAAACCGCGGCGCCGGATGGTTATGAAGTTGCGAC
CGCGATCACCTTTACCGTGAACGAACAGGGTCAAGTGACCGTTAATGGCAAAGCGA
CCAAAGGCGACGCGCACATTGATTCTAGAGTGCCGGGTAGCGGCGTTCCGGGTAGC
GGCGTGCTGGTAGCGGCGTTTCTGGTAGCGGCGTGCCCGGTAGCGGCGTTCCCGGT
AGCGGCGTGCCAGGTAGCGGCGTTCCAGGTAGCGGCGTGCCGGGTAGCGGCGTTCC
CGGTAGCGGCGTGCCGGGTAGCGGCGTTCCCGGTAGCGGCGTGCCAGGTAGCGGCG
TTCCGGGTAGCGGCGTGCCAGGTAGCGGCGTTCTTGGTAGCGGCGTGCCAGGTAGC
GGCGTTCCCGGTAGCGGCGTGCCAGGTAGCGGCGTTCCAGGTAGCGGCGTGCCGGG
TAGCGGCGTTCCAGGTAGCGGCGTGCCAGGTAGCGGCGTTCCAGGTAGCGGCGTGC
CAGGTAGCGGTGCGATGGTTGACACCCTGAGCGGTCTGAGCAGCGAACAGGGTCAA
AGCGGCGACATGACCATCGAGGAAGATTCTGCTACCCACATTAATTTAGCAAGCG
CGACGAGGATGGCAAGGAACTGGCGGGCGCGACCATGGAAGTTCGTGACAGCAGC
GGCAAGACCATTAGCACCTGGATCAGCGATGGTCAAGTGAAAGACTTTTATCTGTAC
CCGGGCAAGTATACCTTCGTTGAAACCGCGGCGCCGGACGGCTACGAGGTGGCGAC
CGCGATTACCTTCACCGTGAACGAACAAGGCCAAGTGACCGTGAACGGCAAGGCGA
CCAAAGGTGACGCGCACATTGATACTAGAGTGCCGGGTAGCGGCGTTCCGGGTAGC
GGCGTGCTGGTAGCGGCGTTTCTGGTAGCGGCGTGCCCGGTAGCGGCGTTCCCGGT
AGCGGCGTGCCAGGTAGCGGCGTTCCAGGTAGCGGCGTGCCGGGTAGCGGCGTTCC
CGGTAGCGGCGTGCCGGGTAGCGGCGTTCCCGGTAGCGGCGTGCCAGGTAGCGGCG
TTCCGGGTAGCGGCGTGCCAGGTAGCGGCGTTCTTGGTAGCGGCGTGCCAGGTAGC
GGCGTTCCCGGTAGCGGCGTGCCAGGTAGCGGCGTTCCAGGTAGCGGCGTGCCGGG
TAGCGGCGTTCCAGGTAGCGGCGTGCCAGGTAGCGGCGTTCCAGGTAGCGGCGTGC
CAGGTAGCGGTGCGATGGTTGACACCCTGAGCGGTCTGAGCAGCGAACAGGGTCAA
AGCGGCGACATGACCATCGAGGAAGATTCTGCTACCCACATTAATTTAGCAAGCG
CGACGAGGATGGCAAGGAACTGGCGGGCGCGACCATGGAAGTTCGTGACAGCAGC
GGCAAGACCATTAGCACCTGGATCAGCGATGGTCAAGTGAAAGACTTTTATCTGTAC
CCGGGCAAGTATACCTTCGTTGAAACCGCGGCGCCGGACGGCTACGAGGTGGCGAC
CGCGATTACCTTCACCGTGAACGAACAAGGCCAAGTGACCGTGAACGGCAAGGCGA
CCAAAGGTGACGCGCACATTGATACTAGAGTGCCGGGTAGCGGCGTTCCGGGTAGC
GGCGTGCTGGTAGCGGCGTTTCTGGTAGCGGCGTGCCCGGTAGCGGCGTTCCCGGT
AGCGGCGTGCCAGGTAGCGGCGTTCCAGGTAGCGGCGTGCCGGGTAGCGGCGTTCC
CGGTAGCGGCGTGCCGGGTAGCGGCGTTCCCGGTAGCGGCGTGCCAGGTAGCGGCG
TTCCGGGTAGCGGCGTGCCAGGTAGCGGCGTTCTTGGTAGCGGCGTGCCAGGTAGC
GGCGTTCCCGGTAGCGGCGTGCCAGGTAGCGGCGTTCCAGGTAGCGGCGTGCCGGG
TAGCGGCGTTCCAGGTAGCGGCGTGCCAGGTAGCGGCGTTCCAGGTAGCGGCGTGC
CAGGTAGCGGTGCGATGGTTGACACCCTGAGCGGTCTGAGCAGCGAACAGGGTCAA
AGCGGCGACATGACCATCGAGGAAGATTCTGCTACCCACATTAATTTAGCAAGCG
CGACGAGGATGGCAAGGAACTGGCGGGCGCGACCATGGAAGTTCGTGACAGCAGC
GGCAAGACCATTAGCACCTGGATCAGCGATGGTCAAGTGAAAGACTTTTATCTGTAC
CCGGGCAAGTATACCTTCGTTGAAACCGCGGCGCCGGACGGCTACGAGGTGGCGAC
CGCGATTACCTTCACCGTGAACGAACAAGGCCAAGTGACCGTGAACGGCAAGGCGA
CCAAAGGTGACGCGCACATTGATACTAGTCTCGAGGATCCGGCTGCTAACAAAGCC
CGAAAGGAAGCTGAGTTGGCTGCTGCCACCGCTGAGCAATAA

MGSSHHHHHSSGLVPRGSHMASAMVDTLSSGLSSEQGQSGDMTIEEDSATHIKFSKRDE

DGKELAGATMELRDSSGKTISTWISDGQVKDFYLYPGKYTFVETAAPDGYEVATAITFT
VNEQQQVTVNGKATKGAHIDSRVPGSGVPGSGVPGSGVPGSGVPGSGVPGSGVPGSG
VPGSGVPGSGVPGSGVPGSGVPGSGVPGSGVPGSGVPGSGVPGSGVPGSGVPGSGV
GVPGSGVPGSGVPGSGVPGSGVPGSGVPGSGVPGSGVPGSGVPGSGVPGSGVPGSGV
IKFSKRDEDGKELAGATMELRDSSGKTISTWISDGQVKDFYLYPGKYTFVETAAPDGYE
VATAITFTVNEQQQVTVNGKATKGAHIDTRVPGSGVPGSGVPGSGVPGSGVPGSGVPGSGV
GSGVPGSGVPGSGVPGSGVPGSGVPGSGVPGSGVPGSGVPGSGVPGSGVPGSGVPGSGV
PGSGVPGSGVPGSGVPGSGVPGSGVPGSGVPGSGVPGSGVPGSGVPGSGVPGSGVPGSGV
TIEEDSATHIKFSKRDEDGKELAGATMELRDSSGKTISTWISDGQVKDFYLYPGKYTFVE
TAAPDGYEVATAITFTVNEQQQVTVNGKATKGAHIDTSLEDPAANKARKEAELAAAT
AEQ*

CC43

DNA sequence:

ATGGGCAGCAGCCATCATCATCATCACAGCAGCGGCCTGGTGCCGCGCGGCAG
CCATATGGCTAGCCGTCTGCCGGCGGGTTGGGAACAGCGTATGGATGTGAAGGGTC
GTCCGTATTTTCGTTGACCACGTTACCAAGAGCACCACCTGGGAGGACCCGCGTCCGG
AGTCTAGAGTGCCGGGTAGCGGCGTTCCGGGTAGCGGCGTGCCTGGTAGCGGCGTT
CCTGGTAGCGGCGTGCCCGGTAGCGGCGTTCCCGGTAGCGGCGTGCCAGGTAGCGG
CGTTCCAGGTAGCGGCGTGCCGGGTAGCGGCGTTCCCGGTAGCGGCGTGCCGGGT
GCGGCGTTCCCGGTAGCGGCGTGCCAGGTAGCGGCGTTCCGGGTAGCGGCGTGCCA
GGTAGCGGCGTTCCCTGGTAGCGGCGTGCCAGGTAGCGGCGTTCCCGGTAGCGGCGT
GCCAGGTAGCGGCGTTCCAGGTAGCGGCGTGCCGGGTAGCGGCGTTCCAGGTAGCG
GCGTGCCAGGTAGCGGCGTTCCAGGTAGCGGCGTGCCAGGTAGCGGTCGTCTGCCG
GCGGGCTGGGAGCAACGTATGGATGTTAAGGGTCGCCCGTATTTTGTTGACCATGTT
ACCAAGAGCACCACCTGGGAGGACCCGCGCCCGGAGACTAGAGTGCCGGGTAGCG
GCGTTCCGGGTAGCGGCGTGCTGGTAGCGGCGTTCCCTGGTAGCGGCGTGCCCGGT
GCGGCGTTCCCGGTAGCGGCGTGCCAGGTAGCGGCGTTCCAGGTAGCGGCGTGCCG
GGTAGCGGCGTTCCCGGTAGCGGCGTGCCGGGTAGCGGCGTTCCCGGTAGCGGCGT
GCCAGGTAGCGGCGTTCCGGGTAGCGGCGTGCCAGGTAGCGGCGTTCCCTGGTAGCG
GCGTGCCAGGTAGCGGCGTTCCCGGTAGCGGCGTGCCAGGTAGCGGCGTTCCAGGT
AGCGGCGTGCCGGGTAGCGGCGTTCCAGGTAGCGGCGTGCCAGGTAGCGGCGTTCC
AGGTAGCGGCGTGCCAGGTAGCGGTCGTCTGCCGGCGGGCTGGGAGCAACGTATGG
ATGTTAAGGGTCGCCCGTATTTTGTTGACCATGTTACCAAGAGCACCACCTGGGAGG
ACCCGCGCCCGGAGACTAGTCTCGAGGATCCGGCTGCTAACAAGCCCGAAAGGAA
GCTGAGTTGGCTGCTGCCACCGCTGAGCAATAA

Amino acid sequence:

MGSSHHHHHSSGLVPRGSHMASRLPAGWEQRMDVKGRPYFVDHVTKSTTWEDPRPE
SRVPGSGVPGSGVPGSGVPGSGVPGSGVPGSGVPGSGVPGSGVPGSGVPGSGV
GSGVPGSGVPGSGVPGSGVPGSGVPGSGVPGSGVPGSGVPGSGVPGSGVPGSGV
PGSGVPGSGRLPAGWEQRMDVKGRPYFVDHVTKSTTWEDPRPETRVPGSGVPGSGVPG
SGVPGSGVPGSGVPGSGVPGSGVPGSGVPGSGVPGSGVPGSGVPGSGVPGSGVPGSGV

E10

ATGGGAGGATCCCATATGTGCGCTAGCGTTCCGGGTAGCGGCGTTCCGGGTTATGGT
GTGCCGGGTTATGGTGTTCGGGCTATGGTGTTCCTGGTAGCGGCGTTCCGGGCTAT
GGCGTGCCGGGTTATGGCGTTCCGGGCTATGGTGTGCCCGGTAGCGGCGTTCCCTGGT
TACGGTGTTCGGGCTACGGTGTTCCTGGCTATGGTGTTCGCGGTAGCGGCGTTCCC
GGTTACGGTGTGCCGGGTTACGGTGTTCGCGGCTACGGTGTTCAGGTAGCGGCGTT
CCTGGATACGGCGTGCCGGGCTACGGTGTTCGGGATATGGTGTTCCTGGTAGCGGC
GTTCCAGGCTACGGTGTTCGGGTTACGGCGTGCCGGGTTACGGTACTAGTTGCCTC
GAGAAGCTTAGATCTCATCACCATCACCATCACTAA

MGGSHMCASVPGSGVPGYGVPGYGVPGYGVPGSGVPGYGVPGYGVPGYGVPGSGVPG
YGVPGYGVPGYGVPGSGVPGYGVPGYGVPGYGVPGSGVPGYGVPGYGVPGYGVPGSG
VPGYGVPGYGVPGYGTSCLEKLRSHHHHHH*

E20

ATGGGAGGATCCCATATGTGCGCTAGCGTTCCGGGTAGCGGCGTTCGGGGTTATGGT
GTGCCGGGTATGGTGTTCGGGGCTATGGTGTTCCTGGTAGCGGCGTTCGGGGCTAT
GGCGTGCCGGGTATGGCGTTCGGGGCTATGGTGTGCCCGGTAGCGGCGTTCCTGGT
TACGGTGTTCGGGGCTACGGTGTTCCTGGCTATGGTGTTCGGGTAGCGGCGTTCCT
GGTTACGGTGTGCCGGGTACGGTGTTCGGGGCTACGGTGTTCAGGTAGCGGCGTTC
CCTGGATACGGCGTGCCGGGGCTACGGTGTTCGGGGATATGGTGTTCCTGGTAGCGGC
GTTCCAGGCTACGGTGTTCGGGGTACGGCGTGCCGGGTACGGTGTTCGGGGTAGC
GGCGTTCGGGGTATGGTGTGCCGGGTATGGTGTTCGGGGCTATGGTGTTCCTGGT
AGCGGCGTTCGGGGCTATGGCGTGCCGGGTATGGCGTTCGGGGCTATGGTGTGCC
GGTAGCGGCGTTCCTGGTTACGGTGTTCGGGGCTACGGTGTTCCTGGCTATGGTGT
CCCGGTAGCGGCGTTCGGGTACGGTGTGCCGGGTACGGTGTTCGGGGCTACGGT
GTTCCAGGTAGCGGCGTTCCTGGATACGGCGTGCCGGGGCTACGGTGTTCGGGGATAT
GGTGTTCCTGGTAGCGGCGTTCAGGCTACGGTGTTCGGGGTACGGCGTGCCGGGT
TACGGTACTAGTTGCCTCGAGAAGCTTAGATCTCATCACCATCACCATCACTAA

MGGSHMCASVPGSGVPGYGVPGYGVPGYGVPGSVPGYGVPGYGVPGYGVPGSVPG
YGVPGYGVPGYGVPGSVPGYGVPGYGVPGYGVPGSVPGYGVPGYGVPGYGVPGSV
VPGYGVPGYGVPGYGTSVPGSGVPGYGVPGYGVPGYGVPGSVPGYGVPGYGVPGYG
VPGSGVPGYGVPGYGVPGYGVPGSVPGYGVPGYGVPGYGVPGSVPGYGVPGYGVP
GYGVPGSGVPGYGVPGYGVPGYGTSCLEKLRSHHHHHHH*

E40

DNA sequence:

ATGGGAGGATCCCATATGTGCGCTAGCGTTCCGGGTAGCGGCGTTCCGGGTTATGGT
GTGCCGGGTTATGGTGTTCGGGGCTATGGTGTTCCTGGTAGCGGCGTTCCGGGCTAT
GGCGTGCCGGGTTATGGCGTTCCGGGGCTATGGTGTGCCCCGGTAGCGGCGTTCCCTGGT
TACGGTGTTCGGGGCTACGGTGTTCCTGGCTATGGTGTTCCTGGTAGCGGCGTTCCC
GGTTACGGTGTGCCGGGTTACGGTGTTCCTGGCTACGGTGTTCAGGTAGCGGCGTT
CCTGGATACGGCGTGCCGGGGCTACGGTGTTCGGGGATATGGTGTTCCTGGTAGCGGC
GTTCCAGGCTACGGTGTTCGGGGTTACGGCGTGCCGGGGTTACGGTGTTCGGGGTAGC
GGCGTTCCGGGGTTATGGTGTGCCGGGTTATGGTGTTCGGGGCTATGGTGTTCCTGGT
AGCGGCGTTCCGGGGCTATGGCGTGCCGGGGTTATGGCGTTCCGGGGCTATGGTGTGCCC
GGTAGCGGCGTTCCCTGGTTACGGTGTTCGGGGCTACGGTGTTCCTGGCTATGGTGTTC
CCCGGTAGCGGCGTTCCCGGTTACGGTGTGCCGGGGTTACGGTGTTCCTGGCTACGGT
GTTCCAGGTAGCGGCGTTCCCTGGATACGGCGTGCCGGGGCTACGGTGTTCGGGGATAT
GGTGTTCCTGGTAGCGGCGTTCCAGGCTACGGTGTTCGGGGTTACGGCGTGCCGGGT
TACGGTGTTCGGGGTAGCGGCGTTCCGGGGTTATGGTGTGCCGGGGTTATGGTGTTCG
GGCTATGGTGTTCCTGGTAGCGGCGTTCCGGGGCTATGGCGTGCCGGGGTTATGGCGTT
CCGGGGCTATGGTGTGCCCGGTAGCGGCGTTCCCTGGTTACGGTGTTCGGGGCTACGGT
GTTCCCTGGCTATGGTGTTCCTGGTAGCGGCGTTCCCGGTTACGGTGTGCCGGGGTTAC
GGTGTTCCTGGCTACGGTGTTCAGGTAGCGGCGTTCCCTGGATACGGCGTGCCGGGG
TACGGTGTTCGGGGATATGGTGTTCCTGGTAGCGGCGTTCCAGGCTACGGTGTTCG
GGTTACGGCGTGCCGGGGTTACGGTGTTCGGGGTAGCGGCGTTCCGGGGTTATGGTGTG
CCGGGGTTATGGTGTTCGGGGCTATGGTGTTCCTGGTAGCGGCGTTCCGGGGCTATGGC
GTGCCGGGGTTATGGCGTTCCGGGGCTATGGTGTGCCCGGTAGCGGCGTTCCCTGGTTAC
GGTGTTCGGGGCTACGGTGTTCCTGGCTATGGTGTTCCTGGTAGCGGCGTTCCCGGT
TACGGTGTGCCGGGGTTACGGTGTTCCTGGCTACGGTGTTCAGGTAGCGGCGTTCCCT
GGATACGGCGTGCCGGGGCTACGGTGTTCGGGGATATGGTGTTCCTGGTAGCGGCGTT
CCAGGCTACGGTGTTCGGGGTTACGGCGTGCCGGGGTTACGGTACTAGTTGCCTCGAG
AAGCTTAGATCTCATCACCATCACCATCACTAA

Amino acid sequence:

MGGSHMCASVPGSGVPGYGVPGYGVPGYGVPGSGVPGYGVPGYGVPGYGVPGSGVPG
YGVPGYGVPGYGVPGSGVPGYGVPGYGVPGYGVPGSGVPGYGVPGYGVPGYGVPGSG
VPGYGVPGYGVPGYGTSPVPGSGVPGYGVPGYGVPGYGVPGSGVPGYGVPGYGVPGYGV
VPGSGVPGYGVPGYGVPGYGVPGSGVPGYGVPGYGVPGYGVPGSGVPGYGVPGYGVPG
GYGVPGSGVPGYGVPGYGVPGYGTSPVPGSGVPGYGVPGYGVPGYGVPGSGVPGYGVPG
GYGVPGYGVPGSGVPGYGVPGYGVPGYGVPGSGVPGYGVPGYGVPGYGVPGSGVPGY
GVPGYGVPGYGVPGSGVPGYGVPGYGVPGYGTSPVPGSGVPGYGVPGYGVPGYGVPGS
GVPGYGVPGYGVPGYGVPGSGVPGYGVPGYGVPGYGVPGSGVPGYGVPGYGVPGYGV
VPGSGVPGYGVPGYGVPGYGVPGSGVPGYGVPGYGTSCLEKLRRSHHHHHH*

E80

DNA sequence:

ATGGGAGGATCCCATATGTGCGCTAGCGTTCCGGGTAGCGGCGTTCCGGGTTATGGT

GTGCCGGGTTATGGTGTTCGGGGCTATGGTGTTCCTGGTAGCGGCGTTCCGGGGCTAT
GGCGTGCCGGGTTATGGCGTTCCGGGGCTATGGTGTGCCCGGTAGCGGCGTTCCCTGGT
TACGGTGTTCGGGGCTACGGTGTTCCTGGCTATGGTGTTCGGGTAGCGGCGTTCCC
GGTTACGGTGTGCCGGGTTACGGTGTTCGGGGCTACGGTGTTCAGGTAGCGGCGTT
CCTGGATACGGCGTGCCGGGGCTACGGTGTTCGGGGATATGGTGTTCCTGGTAGCGGC
GTTCCAGGCTACGGTGTTCGGGGTTACGGCGTGCCGGGGTTACGGTGTTCGGGGTAGC
GGCGTTCCGGGGTTATGGTGTGCCGGGGTTATGGTGTTCGGGGCTATGGTGTTCCTGGT
AGCGGCGTTCCGGGGCTATGGCGTGCCGGGGTTATGGCGTTCCGGGGCTATGGTGTGCCC
GGTAGCGGCGTTCCCTGGTTACGGTGTTCGGGGCTACGGTGTTCCTGGCTATGGTGTTC
CCGGGTAGCGGCGTTCCCGGTTACGGTGTGCCGGGGTTACGGTGTTCGGGGCTACGGT
GTTCCAGGTAGCGGCGTTCCCTGGATACGGCGTGCCGGGGCTACGGTGTTCGGGGATAT
GGTGTTCCTGGTAGCGGCGTTCCAGGCTACGGTGTTCGGGGTTACGGCGTGCCGGGGT
TACGGTGTTCGGGGTAGCGGCGTTCCGGGGTTATGGTGTGCCGGGGTTATGGTGTTCG
GGCTATGGTGTTCCTGGTAGCGGCGTTCCGGGGCTATGGCGTGCCGGGGTTATGGCGTT
CCGGGGCTATGGTGTGCCCGGTAGCGGCGTTCCCTGGTTACGGTGTTCGGGGCTACGGT
GTTCCCTGGCTATGGTGTTCGGGTAGCGGCGTTCCCGGTTACGGTGTGCCGGGGTTAC
GGTGTTCGGGGCTACGGTGTTCAGGTAGCGGCGTTCCCTGGATACGGCGTGCCGGGG
TACGGTGTTCGGGGATATGGTGTTCCTGGTAGCGGCGTTCCAGGCTACGGTGTTCG
GGTTACGGCGTGCCGGGGTTACGGTGTTCGGGGTAGCGGCGTTCCGGGGTTATGGTGTG
CCGGGGTTATGGTGTTCGGGGCTATGGTGTTCCTGGTAGCGGCGTTCCGGGGCTATGGC
GTGCCGGGGTTATGGCGTTCCGGGGCTATGGTGTGCCCGGTAGCGGCGTTCCCTGGTTAC
GGTGTTCGGGGCTACGGTGTTCCTGGCTATGGTGTTCGGGTAGCGGCGTTCCCGGT
TACGGTGTGCCGGGGTTACGGTGTTCGGGGCTACGGTGTTCAGGTAGCGGCGTTCCCT
GGATACGGCGTGCCGGGGCTACGGTGTTCGGGGATATGGTGTTCCTGGTAGCGGCGTT
CCAGGCTACGGTGTTCGGGGTTACGGCGTGCCGGGGTTACGGTGTTCGGGGTAGCGGC
GTTCCGGGGTTATGGTGTGCCGGGGTTATGGTGTTCGGGGCTATGGTGTTCCTGGTAGC
GGCGTTCCGGGGCTATGGCGTGCCGGGGTTATGGCGTTCCGGGGCTATGGTGTGCCCGGT
AGCGGCGTTCCCTGGTTACGGTGTTCGGGGCTACGGTGTTCCTGGCTATGGTGTTCCT
GGTAGCGGCGTTCCCGGTTACGGTGTGCCGGGGTTACGGTGTTCGGGGCTACGGTGTTC
CCAGGTAGCGGCGTTCCCTGGATACGGCGTGCCGGGGCTACGGTGTTCGGGGATATGGT
GTTCCCTGGTAGCGGCGTTCCAGGCTACGGTGTTCGGGGTTACGGCGTGCCGGGGTTAC
GGTGTTCGGGGTAGCGGCGTTCCGGGGTTATGGTGTGCCGGGGTTATGGTGTTCGGGGC
TATGGTGTTCCTGGTAGCGGCGTTCCGGGGCTATGGCGTGCCGGGGTTATGGCGTTCCG
GGCTATGGTGTGCCCGGTAGCGGCGTTCCCTGGTTACGGTGTTCGGGGCTACGGTGTTC
CCTGGCTATGGTGTTCGGGTAGCGGCGTTCCCGGTTACGGTGTGCCGGGGTTACGGT
GTTCCCGGCTACGGTGTTCAGGTAGCGGCGTTCCCTGGATACGGCGTGCCGGGGCTAC
GGTGTTCGGGGATATGGTGTTCCTGGTAGCGGCGTTCCAGGCTACGGTGTTCGGGGT
TACGGCGTGCCGGGGTTACGGTGTTCGGGGTAGCGGCGTTCCGGGGTTATGGTGTGCCG
GGTTATGGTGTTCGGGGCTATGGTGTTCCTGGTAGCGGCGTTCCGGGGCTATGGCGTG
CCGGGGTTATGGCGTTCCGGGGCTATGGTGTGCCCGGTAGCGGCGTTCCCTGGTTACGGT
GTTCCGGGGCTACGGTGTTCCTGGCTATGGTGTTCGGGTAGCGGCGTTCCCGGTTAC
GGTGTGCCGGGGTTACGGTGTTCGGGGCTACGGTGTTCAGGTAGCGGCGTTCCCTGGA
TACGGCGTGCCGGGGCTACGGTGTTCGGGGATATGGTGTTCCTGGTAGCGGCGTTCCA

GGCTACGGTGTTCCTGGGTTACGGCGTGCCGGGTTACGGTGTTCCTGGGTAGCGGCGTT
CCGGGTTATGGTGTGCCGGGTTATGGTGTTCCTGGGTAGCGGC
GTTCCGGGCTATGGCGTGCCGGGTTATGGCGTTCCGGGCTATGGTGTGCCCGGTAGC
GGCGTTCCTGGTTACGGTGTTCCTGGGCTACGGTGTTCCTGGCTATGGTGTTCCTGGTA
GCGGCGTTCCTGGTTACGGTGTGCCGGGTTACGGTGTTCCTGGGCTACGGTGTTCAG
GTAGCGGCGTTCCTGGATACGGCGTGCCGGGCTACGGTGTTCCTGGGATATGGTGTTC
CTGGTAGCGGCGTTCAGGCTACGGTGTTCCTGGGTTACGGCGTGCCGGGTTACGGTA
CTAGTTGCCTCGAGAAGCTTAGATCTCATCACCATCACCATCACTAA

Amino acid sequence:

MGGSHMCASVPGSGVPGYGVPGYGVPGYGVPGSGVPGYGVPGYGVPGYGVPGSGVPG
YGVPGYGVPGYGVPGSGVPGYGVPGYGVPGYGVPGSGVPGYGVPGYGVPGYGVPGSG
VPGYGVPGYGVPGYGTSPVPGSGVPGYGVPGYGVPGYGVPGSGVPGYGVPGYGVPGYGV
VPGSGVPGYGVPGYGVPGYGVPGSGVPGYGVPGYGVPGSGVPGYGVPGYGVPGYGVPG
GYGVPGSGVPGYGVPGYGVPGYGTSPVPGSGVPGYGVPGYGVPGYGVPGSGVPGYGVPG
GYGVPGYGVPGSGVPGYGVPGYGVPGYGVPGSGVPGYGVPGYGVPGYGVPGSGVPGY
GVPGYGVPGYGVPGSGVPGYGVPGYGVPGYGTSPVPGSGVPGYGVPGYGVPGYGVPGS
GVPGYGVPGYGVPGYGVPGSGVPGYGVPGYGVPGYGVPGSGVPGYGVPGYGVPGYGV
VPGSGVPGYGVPGYGVPGYGVPGSGVPGYGVPGYGTSPVPGSGVPGYGVPGYGVPGYGV
GYGVPGYGVPGSGVPGYGVPGYGVPGYGVPGSGVPGYGVPGYGVPGYGTSPVPGSGVP
GYGVPGYGVPGYGVPGSGVPGYGVPGYGVPGYGVPGSGVPGYGVPGYGVPGYGVPGS
GVPGYGVPGYGVPGYGVPGSGVPGYGVPGYGVPGYGVPGSGVPGYGVPGYGVPGYGT
SPVPGSGVPGYGVPGYGVPGYGVPGSGVPGYGVPGYGVPGYGVPGSGVPGYGVPGYGV
PGYGVPGSGVPGYGVPGYGVPGYGVPGSGVPGYGVPGYGVPGYGVPGSGVPGYGVPG
YGVPGYGTSPVPGSGVPGYGVPGYGVPGYGVPGSGVPGYGVPGYGVPGYGVPGSGVPG
YGVPGYGVPGYGVPGSGVPGYGVPGYGVPGYGVPGSGVPGYGVPGYGVPGYGVPGSG
VPGYGVPGYGVPGYGTSCLEKLRSHHHHHH*

hNup50

DNA sequence:

ATGGCGAAGCGTAACGCGGAAAAAGAGCTGACCGACCGTAACCTGGGACCAGGAAG
ATGAGGCGGAGGAAGTTGGCACCTTCAGCATGGCGAGCGAGGAAGTGCTGAAGAAC
CGTGCGATCAAGAAAGCGAAACGTCGTAACGTTGGTTTTGAAAGCGATACCGGTGG
CGCGTTCAAGGGTTTTAAGGGCCTGGTGGTTCCGAGCGGTGGCGGTCTGTTTCAGCGG
TTTTGGTAGCGGTGCGGGCGGTAAACCGCTGGAGGGTCTGAGCAACGGCAACAACA
TTACCAGCGCTCCGCCGTTTTCGAGCGCGAAAGCGGCGGCGGACCCGAAAGTTGCG
TTTGGTAGCCTGGCGGCGAACGGTCCGACCACCCTGGTTGACAAGGTGAGCAACCC
GAAAACCAACGGCGATAGCCAGCAACCGAGCAGCAGCGGTCTGGCGAGCAGCAAG
GCGTGCGTTGGCAACGCGTACCACAAACAGCTGGCGGCGCTGAACTGCAGCGTGCG
TGATTGGATCGTTAAGCACGTGAACACCAACCCGCTGTGCGACCTGACCCCGATCTT
CAAGGATTACGAAAAATATCTGGCGAACATTGAGCAGCAACACGGTAACAGCGGCC
GTAACAGCGAAAGCGAGAGCAACAAAGTTGCGGGCGGAAACCCAAAGCCCGAGCCT

GTTTGGCAGCACCAAGCTGCAGCAAGAAAGCACCTTCCTGTTTCACGGTAACAAAA
CCGAGGACACCCCGGATAAGAAAATGGAAGTGGCGAGCGAGAAGAAAACCGACCC
GAGCAGCCTGGGTGCGACCAGCGCGAGCTTCAACTTTGGCAAGAAAGTTGATAGCA
GCGTGCTGGGTAGCCTGAGCAGCGTTCGCTGACCGGTTTCAGCTTTAGCCCGGGCA
ACAGCAGCCTGTTCGGCAAGGACACCACCCAGAGCAAACCGGTGAGCAGCCCGTTT
CCGACCAAGCCGCTGGAAGGTCAAGCGGAGGGTGATAGCGGCGAGTGCAAAGGCG
GTGACGAGGAAGAGAACGATGAACCGCCGAAGGTGGTTGTGACCGAGGTTAAAGA
AGAGGACGCGTTCTACAGCAAGAAATGCAAGCTGTTCTATAAGAAAGATAACGAAT
TCAAGGAGAAGGGTATCGGCACCCTGCACCTGAAGCCGACCGCGAACCAGAAAACC
CAACTGCTGGTTCGTGCGGACACCAACCTGGGCAACATCCTGCTGAACGTGCTGATT
CCGCCGAACATGCCGTGCACCCGTACCGGCAAGAACAACGTGCTGATCGTTTGCGT
GCCGAACCCGCCGATTGACGAGAAGAACGCGACCATGCCGGTTACCATGCTGATCC
GTGTGAAGACCAGCGAAGACGCGGATGAGCTGCACAAAATTCTGCTGGAGAAGAAA
GATGCGACTAGTCTCGAGTAG

Amino acid sequence:

MAKRNAEKELTDRNWDQEDEAEVGTFSMASEEVLKNRAIKKAKRRNVGFESDTGGA
FKGFKGLVVPSSGGGRFSGFGSGAGGKPLEGLSNGNNITSAPPFASAKAAADPKVAFGSL
AANGPTTLVDKVSNPKTNGDSQQPSSSGLASSKACVGNAYHKQLAALNCSVRDWIVKH
VNTNPLCDLTPIFKDYEKYLANIEQQHGNSEGRNSESESNKVAAETQSPSLFGSTKLQES
TFLFHGNKTEDTPDKKMEVASEKKTDPSSLGATSASFNFGKKVDSSVLGSLSSVPLTGFS
FSPGNSSLFGKDTTQSKPVSSPFPTKPLEGQAEGDSGECKGGDEEENDEPPKVVVTEVKE
EDAFYSKKCKLFYKKDNEFKEKGIGTLHLKPTANQKTQLLVRADTNLGNILLNVLIPP
MPCTRGTGKNNVLIVCVPNPPIDEKNATMPVTMLIRVKTSSEDADELHKILLEKKDATSLE*

hNup62

DNA sequence:

ATGAGCGGCTTCAACTTTGGTGGCACCGGTGCGCCGACCGGTGGCTTCACCTTTGGT
ACCGCGAAAACCGCGACCACCACCCCGGCGACCGGTTTCAGCTTTAGCACCAGCGG
TACCGGTGGCTTCAACTTTGGTGGCGCCGTTCCAGCCGGCGACCAGCACCACCGAGCAC
CGGTCTGTTTAGCCTGGCGACCCAGACCCCGGCGACCCAAACCACCGGTTTTACCTT
TGGTACCGCGACCCTGGCGAGCGGTGGCACCGGTTTCAGCCTGGGTATTGGTGGCGAG
CAAGCTGAACCTGAGCAACACCGCGGCGACCCCGGCGATGGCGAACCCGAGCGGTT
TTGGCCTGGGTAGCAGCAACCTGACCAACGCGATTAGCAGCACCGTTACCAGCAGC
CAGGGTACCGCGCCGACCGGTTTCGTTTTTGGTCCGAGCACCACCAGCGTGGCGCCG
GCGACCACCAGCGGTGGCTTCAGCTTTACCGGTGGCAGCACCGCGCAACCGAGCGG
TTTCAACATTGGCAGCGCGGGTAACAGCGCGCAACCGACCGCGCCGGCGACCCCTGC
CGTTTACCCCGGCGACCCCGGCGGCGACCACCGCGGGTGCGACCCAGCCGGCGGGCG
CCGACCCCGACCGCGACCATACCAGCACCGGTCCGAGCCTGTTTGCGAGCATTGCG
ACCGCGCCGACCAGCAGCGCGACCACCGGTCTGAGCCTGTGCACCCCGGTGACCAC
CGCGGGTGCGCCGACCGCGGGTACCCAAGGTTTTAGCCTGAAAGCGCCGGGTGCGG
CGAGCGGTACCAGCACCACCAGCACCGCGGCGACCGCGACCGCTACCACCACC
ACCAGCAGCAGCACCACCGGTTTCGCGCTGAACCTGAAGCCGCTGGCGCCGGCGGG

CATCCCGAGCAACACCGCTGCGGCGGTTACCGCTCCGCCGGGTCCGGGTGCTGCGG
CGGGTGCGGCGGCGAGCAGCGCGATGACCTACGCGCAGCTGGAGAGCCTGATTAAC
AAATGGAGCCTGGAGCTGGAAGACCAGGAACGTCACCTTCTGCAGCAAGCGACCCA
AGTGAACGCGTGGGATCGTACCCTGATCGAGAACGGTGAAAAGATTACCAGCCTGC
ACCGTGAGGTGGAAAAGGTTAAACTGGACCAGAAGCGTCTGGACCAAGAGCTGGAT
TTCATCCTGAGCCAGCAAAAAGAGCTGGAGGACCTGCTGAGCCCGCTGGAGGAACT
GGTTAAGGAACAGAGCGGTACCATCTACCTGCAACACGCGGACGAGGAACGTGAGA
AGACCTATAAACTGGCGGAAAACATTGATGCGCAGCTGAAGCGTATGGCGCAAGAC
CTGAAAGATATCATTGAGCACCTGAACACCAGCGGTGCGCCGGCGGACACCAGCGA
TCCGCTGCAGCAAATCTGCAAAATTCTGAACGCGCACATGGACAGCCTGCAGTGGA
TCGATCAAAACAGCGCGCTGCTGCAGCGTAAGGTGGAGGAAGTTACCAAAGTGTGC
GAAGGTCGTCGTAAGGAGCAAGAACGTAGCTTCCGTATTACCTTTGACACTAGTCTC
GAGTAG

Amino acid sequence:

MSGFNFGGTGAPTGGFTFGTAKTATTTTPATGFSFSTSGTGGFNFGAPFQPATSTPSTGLFS
LATQTPATQTTGFTFGTATLASGGTGFSLGIGASKLNLNNTAATPAMANPSGFGLGSSNL
TNAISSTVTSSQGTAPTGFVFGPSTTSVAPATTSGGFSFTGGSTAQPSGFNIGSAGNSAQ
TAPATLPFTPATPAATTAGATQPAAPTPTATITSTGPSLFIASIAAPTSSATTGLSLCTPVT
TAGAPTAGTQGFSLKAPGAASGTSTTTSTAATATATTTTSSSTTGFAALNLKPLAPAGIPSN
TAAAVTAPPGPGAAAGAAASSAMTYAQLESLINKWSLELEDQERHFLQQATQVNAWD
RTLIENGEKITS LHREVEKVKLDQKRLDQELDFILSQQKELEDLLSPLEELVKEQSGTIYL
QHADEEREKTYKLAENIDAQLKRMAQDLKDIIHLNTSGAPADTSDPLQIQICKILNAHM
DSLQWIDQNSALLQRKVEEVTKVCEGRRKEQERSFRITFDTSLE*

hNup98

DNA sequence:

ATGTTCAACAAGAGCTTTGGTACCCCGTTTGGTGGCGGTACCGGCGGTTTGGCACC
ACCAGCACCTTCGGTCAGAACACCGGCTTTGGTACCACCAGCGGCGGTGCGTTTGGT
ACCAGCGCGTTTCGGTAGCAGCAACAACACCGGCGGTCTGTTTGGCAACAGCCAGAC
CAAACCGGGCGGTCTGTTTGGTACCAGCAGCTTTAGCCAGCCGGCGACCAGCACCA
GCACCGGTTTCGGTTTTGGCACCAAGCAGCACCAGTACCGCGAACACCCTGTTTCGGCACC
CGAGCACCGGTACCAGCCTGTTTCAGCAGCCAGAACAAACGCGTTTGCAGCAAAACAAA
CCGACCGGTTTCGGCAACTTTGGCACCAAGCAGCAGCAGCGGCGGTCTGTTTCGGCACC
ACCAACACCACCAGCAACCCGTTTCGGTAGCACCAGCGGCAGCCTGTTTGGTCCGAG
CAGCTTCACCGCGGCGCCGACCGGTACCACCATCAAGTTCAACCCGCCGACCGGCA
CCGACACCATGGTTAAAGCGGGTGTGAGCACCAACATCAGCACCAAGCACCAGTGC
ATTACCGCGATGAAAGAATACGAGAGCAAGAGCCTGGAGGAACTGCGTCTGGAGGA
TTATCAAGCGAACCGTAAAGGTCCGCAGAACCAAGTTGGTGGCGGCACCACCACCG
GTCTGTTTGGCAGCAGCCCGGCGACCAAGCAGCGCGACCGGTCTGTTTCAGCAGCAGC
ACCACCAACAGCGGCTTCGCGTACGGTCAGAACAAAACCGCGTTTCGGCACCAGCAC
CACCGGTTTTGGCACCAACCCGGGCGGTCTGTTTCGGTCAGCAAAACCAGCAAACCA
CCAGCCTGTTTAGCAAACCGTTTGGTCAGGCGACCACCACCAAAACACCGGCTTCA

GCTTTGGTAACACCAGCACCATTTGGTCAGCCGAGCACCAACACCATGGGGTAGCTTTG
GTGTTACCCAAGCGAGCCAGCCGGGCGGTCTGTTTGGCACCGCGACCAATACCAGC
ACCGGCACCGCGTTTGGTACCGGTACCGGTCTGTTTGGTCAAACCAACACCGGTTTT
GGTGCGGTGGGCAGCACCTGTTTGGTAACAACAAGCTGACCACCTTTGGCAGCGGT
ACCACCAGCGCGCCGAGCTTCGGCACCAACCAGCGGCGGTCTGTTTGGTTTTGGTACC
AACACCAGCGGCAACAGCATCTTTGGTAGCAAACCGGCGCCGGGTACCCTGGGTAC
CGGTCTGGGTGCGGGTTTTGGTACCGCGCTGGGTGCGGGTCAGGCGAGCCTGTTTCGG
CAACAACCAACCGAAGATTGGCGGTCCGCTGGGTACCGGTGCGTTTGGTGCGCCGG
GTTTCAACACCACCACCGCGACCTGGGTTTTTGGTGCGCCGCAGGCGCCGGTTGCGC
TGACCGACCCGAACGCGAGCGCGGCGCAGCAAGCGGTGCTGCAGCAACACATCAAC
AGCCTGACCTACAGCCCGTTTCGGTGACAGCCCGCTGTTTCGTAACCCGATGAGCGAC
CCGAAGAAGAAAGAGGAGCGTCTGAAACCGACCAACCCGGCGGGCGCAGAAGGCGC
TGACCACCCCGACCCACTATAAACTGACCCCGCGTCCGGCGACCCGTGTGCGTCCGA
AAGCGCTGCAAACCACCGGCACCGCGAAGAGCCACCTGTTTGATGGTCTGGACGAT
GACGAGCCGAGCCTGGCGAACGGTGCGTTCATGCCGAAAAAGAGCATTAAAAAGCT
GGTCTGAAAAACCTGAACAACAGCAACCTGTTTAGCCCGGTGAACCGTGACAGCG
AAAACCTGGCGAGCCCGAGCGAATACCCGGAGAACGGTGAACGTTTCAGCTTTCTG
AGCAAGCCGGTTGATGAGAACCACCAGCAAGATGGCGACGAAGATAGCCTGGTGAG
CCACTTCTATACCAACCCGATCGCGAAGCCGATTCCGCAGACCCCGGAGAGCGCGG
GTAACAAGCACAGCAACAGCAACAGCGTTGATGACACCATTGTGGCGCTGAACATG
CGTGCGGCGCTGCGTAACGGTCTGGAAGGCAGCAGCGAGGAAACCAGCTTCCACGA
CGAGAGCCTGCAAGATGACCGTGAGGAAATTGAAAACAACAGCTACCACATGCACC
CGGCGGGCATCATTCTGACCAAGGTTGGTTACTATACCATCCCGAGCATGGATGACC
TGGCGAAAATTACCAACGAGAAGGGCGAATGCATCGTGAGCGACTTTACCATTTGGT
CGTAAAGGTTACGGCAGCATCTATTTTCGAGGGTGATGTAAACCTGACCAACCTGAAC
CTGGATGACATCGTGACATTCGTCGTAAGGAAGTGGTTGTGTACCTGGATGACAAC
CAGAAACCGCCGGTTGGCGAGGGCCTGAACCGTAAGGCGGAAGTTACCCTGGACGG
TGTGTGGCCGACCGATAAAACCAGCCGTTGCCTGATCAAGAGCCCGGACCGTCTGG
CGGATATTAACCTATGAGGGCCGCTCTGGAAGCGGTGAGCCGTAAACAGGGTGCGCAA
TTCAAGGAGTACCGTCCGGAAACCGGCAGCTGGGTTTTTAAAGTGAGCCACTTCAGC
AAGTATGGTCTGCAGGACAGCGATGAGGAAGAGGAAGAGCACCCGAGCAAAACCA
GCACCAAAAAGCTGAAAACCGCGCCGCTGCCGCCGGCGAGCCAGACCACCCCGCTG
CAAATGGCGCTGAACGGTAAACCGGCGCCGCCGCCGAGGTTGAGAAGAAGGGTCA
AACTAGTCTCGAGTAG

Amino acid sequence:

MFNKSFGTPFGGGTGGFGTTSTFGQNTGFGTSSGGAFGTSAFGSSNNTGGLFGNSQTKP
GGLFGTSSFSQPATSTSTGFGFGTSTGTANTLFGTASTGTSLFSSQNNFAQNKPTGFGNF
GTSTSSGGLFGTTNTTSNPFGSTSGSLFGPSSFTAAPTGTTIKFNPPGTGDTMVKAGVSTNI
STKHQCITAMKEYESKSLEELRLEDYQANRKGPQNQVAGTTTTGLFGSSPATSSATGLF
SSSTTNSGFAYGQNKTAFGTSTTGFGTNPGLFGQQNQQTSLFSKPFQQATTTQNTGFS
FGNTSTIGQPSTNTMGSFGVTQASQPGGLFGTATNTSTGTAFGTGTGLFGQTNTGFGAV
GSTLFGNNKLTTFGSGTTSAPSFGTSSGGLFGFGTNTSGNSIFGSKPAPGTLGTGLGAGFG
TALGAGQASLFGNNQPKIGGPLGTGAFGAPGFNTTTATLFGGAPQAPVALTDPNASAAQ

QAVLQQHINSLTYSFPGDSPLFRNPMSDPKKKEERLKPTNPAAQKALTTPTHYKLTTPRPA
TRVRPKALQTTGTAKSHLFDGLDDDEPSLANGAFMPKKSIIKLVLKNLNNSNLFSPVNR
DSENLASPSEYPENGERFSFLSKPVDENHQQDGEDSLVSHFYTNPIAKPIQTPESAGNK
HSNSNSVDDTIVALNMRAALRNGLEGSSEETSFHDESLQDDREEIENNSYHMHMPAGIILT
KVGYYTIPSMDDLAKITNEKGECIVSDFTIGRKGYSIYFEGDVNLTNLNLDDIVHIRRKE
VVVYLDDNQKPPVGEGLNRKAEVTLDGWVPTDKTSRCLIKSPDRLADINYEGRLEAVS
RKQGAQFKEYRPETGSWVFKVSHFSKYGLQDSDEEEEEHPSKTSTKKLKTAPLPPASQT
TPLQMALNGKPAPPPQVEKKGQTSLE*

mCherry

DNA sequence:

ATGGGCAGCAGCCATCATCATCATCACAGCAGCGGCCTGGTGCCGCGCGGCAG
CCATATGGTGAGCAAGGGCGAGGAGGATAACATGGCCATCATCAAGGAGTTCATGC
GCTTCAAGGTGCACATGGAGGGCTCCGTGAACGGCCACGAGTTCGAGATCGAGGGC
GAGGGCGAGGGCCGCCCTACGAGGGCACCCAGACCGCCAAGCTGAAGGTGACCA
AGGGTGGCCCCCTGCCCTTCGCCTGGGACATCCTGTCCCCTCAGTTCATGTACGGCT
CCAAGGCCTACGTGAAGCACCCCGCCGACATCCCCGACTACTTGAAGCTGTCCTTCC
CCGAGGGCTTCAAGTGGGAGCGCGTGATGAACTTCGAGGACGGCGGCGTGTTGACC
GTGACCCAGGACTCCTCCCTGCAGGACGGCGAGTTCATCTACAAGGTGAAGCTGCG
CGGCACCAACTTCCCCTCCGACGGCCCCGTAATGCAGAAGAAGACCATGGGCTGGG
AGGCCTCCTCCGAGCGGATGTACCCCGAGGACGGCGCCCTGAAGGGCGAGATCAAG
CAGAGGCTGAAGCTGAAGGACGGCGGCCACTACGACGCTGAGGTCAAGACCACCTA
CAAGGCCAAGAAGCCCGTGACGCTGCCCGGCGCCTACAACGTCAACATCAAGTTGG
ACATCACCTCCCACAACGAGGACTACACCATCGTGGAACAGTACGAACGCGCCGAG
GGCCGCGACTCCACCGGCGGCATGGACGAGCTGTACAAGCTCGAGGATCCGGCTGC
TAACAAAGCCCGAAAGGAAGCTGAGTTGGCTGCTGCCACCGCTGAGCAATAA

Amino acid sequence:

MGSSHHHHHSSGLVPRGSHMVSKGEEDNMAIIKEFMRFKVHMEGSVNGHEFEIEGEG
EGRPYEGTQTAKLKVTKGGPLPFAWDILSPQFMYGSKAYVKHPADIPDYKLKLSFPEGFK
WERVMNFEDGGVVTVTQDSSLQDGEFIYKVKLRGTNFPDGPVMQKKTMGWEASSER
MYPEDGALKGEIKQRLKLKDGGHYDAEVKTTYKAKKPVQLPGAYNVNIKLDITSHNED
YTIVEQYERAEGRHSTGGMDELYKLEDPAANKARKEAELAAATAEQ*

P4

DNA sequence:

ATGGGTGCCGGCGCTGGTGCGGGCCCGGAAGGTGCAGGCGCTGGTGCGGGCCCGGA
AGGTGCCGGCGCTGGTGCGGGCCCGGAAGGTGCAGGCGCTGGTGCGGGCCCGGAAG
GTGCCGGCGCTGGTGCGGGCCCGGAAGGTGCAGGCGCTGGTGCGGGCCCGGAAGGT
GCCGGCGCTGGTGCGGGCCCGGAAGGTGCAGGCGCTGGTGCGGGCCCGGAAGGTGC
CGGCGCTGGTGCGGGCCCGGAAGGTGCAGGCGCTGGTGCGGGCCCGGAAGGTGCC
GCATGCCGACTAGCGCGCCGCAGATGCTGCGTGAAGTGCAGGAAACCAACGCGGCG
CTGCAGGACGTTCTGTGAAGTCTGCGTCAGCAGGTTAAAGAAATCACCTTCCTGAAA

AACACCGTTATGGAATCTGACGCGTCTGGTACTAGCTATCGCGATCCGATGGGTGCC
GGCGCTGGTGCGGGCCCGGAAGGTGCAGGCGCTGGTGCGGGCCCGGAAGGTGCCGG
CGCTGGTGCGGGCCCGGAAGGTGCAGGCGCTGGTGCGGGCCCGGAAGGTGCCGGCG
CTGGTGCGGGCCCGGAAGGTGCAGGCGCTGGTGCGGGCCCGGAAGGTGCCGGCGCT
GGTGCGGGCCCGGAAGGTGCAGGCGCTGGTGCGGGCCCGGAAGGTGCCGGCGCTGG
TGCGGGCCCGGAAGGTGCAGGCGCTGGTGCGGGCCCGGAAGGTGCCCGCATGCCGA
CTAGCGCGCCGAGATGCTGCGTGAACTGCAGGAAACCAACGCGGCGCTGCAGGAC
GTTCGTGAACTGCTGCGTCAGCAGGTTAAAGAAATCACCTTCCTGAAAAACACCGTT
ATGGAATCTGACGCGTCTGGTACTAGCTATCGCGATCCGATGGGTGCCGGCGCTGGT
GCGGGCCCGGAAGGTGCAGGCGCTGGTGCGGGCCCGGAAGGTGCCGGCGCTGGTG
GGGCCCGGAAGGTGCAGGCGCTGGTGCGGGCCCGGAAGGTGCCGGCGCTGGTGCGG
GCCCGGAAGGTGCAGGCGCTGGTGCGGGCCCGGAAGGTGCCGGCGCTGGTGCGGGC
CCGGAAGGTGCAGGCGCTGGTGCGGGCCCGGAAGGTGCCGGCGCTGGTGCGGGCCC
GGAAGGTGCAGGCGCTGGTGCGGGCCCGGAAGGTGCCCGCATGCCGACTAGCGCGC
CGCAGATGCTGCGTGAACTGCAGGAAACCAACGCGGCGCTGCAGGACGTTTCGTGAA
CTGCTGCGTCAGCAGGTTAAAGAAATCACCTTCCTGAAAAACACCGTTATGGAATCT
GACGCGTCTGGTACTAGCTATCGCGATCCGATGGGTGCCGGCGCTGGTGCGGGCCC
GGAAGGTGCAGGCGCTGGTGCGGGCCCGGAAGGTGCCGGCGCTGGTGCGGGCCCGG
AAGGTGCAGGCGCTGGTGCGGGCCCGGAAGGTGCCGGCGCTGGTGCGGGCCCGGAA
GGTGCAGGCGCTGGTGCGGGCCCGGAAGGTGCCGGCGCTGGTGCGGGCCCGGAAGG
TGCAGGCGCTGGTGCGGGCCCGGAAGGTGCCGGCGCTGGTGCGGGCCCGGAAGGTG
CAGGCGCTGGTGCGGGCCCGGAAGGTGCCCGCATGCCGACTAGCGCGCCGAGATG
CTGCGTGAACTGCAGGAAACCAACGCGGCGCTGCAGGACGTTTCGTGAACTGCTGCG
TCAGCAGGTTAAAGAAATCACCTTCCTGAAAAACACCGTTATGGAATCTGACGCGTC
TGGTACTAGCTATCGCGATCCGATGGGTGCCGGCGCTGGTGCGGGCCCGGAAGGTG
CAGGCGCTGGTGCGGGCCCGGAAGGTGCCGGCGCTGGTGCGGGCCCGGAAGGTGCA
GGCGCTGGTGCGGGCCCGGAAGGTGCCGGCGCTGGTGCGGGCCCGGAAGGTGCAGG
CGCTGGTGCGGGCCCGGAAGGTGCCGGCGCTGGTGCGGGCCCGGAAGGTGCAGGCG
CTGGTGCGGGCCCGGAAGGTGCCGGCGCTGGTGCGGGCCCGGAAGGTGCAGGCGCT
GGTGCGGGCCCGGAAGGTGCCCGCATGCCGACTAGTCTCGAGGATCCGGCTGCTAA
CAAAGCCCGAAAGGAAGCTGAGTTGGCTGCTGCCACCGCTGAGCAATAA

Amino acid sequence:

MGAGAGAGPEGAGAGAGPEGAGAGAGPEGAGAGAGPEGAGAGAGPEGAGAGAGAGPEG
AGAGAGPEGAGAGAGPEGAGAGAGPEGAGAGAGPEGARMPTSAPQMLRELQETNAAL
LQDVRELLRQQVKEITFLKNTVMESDASGTSYRDPMGAGAGAGPEGAGAGAGPEGAG
AGAGPEGAGAGAGPEGAGAGAGPEGAGAGAGPEGAGAGAGPEGAGAGAGPEGAGAG
AGPEGAGAGAGPEGARMPTSAPQMLRELQETNAALQDVRELLRQQVKEITFLKNTVME
SDASGTSYRDPMGAGAGAGPEGAGAGAGPEGAGAGAGPEGAGAGAGPEGAGAGAGP
EGAGAGAGPEGAGAGAGPEGAGAGAGPEGAGAGAGPEGAGAGAGPEGARMPTSAPQ
MLRELQETNAALQDVRELLRQQVKEITFLKNTVMESDASGTSYRDPMGAGAGAGPEGA
GAGAGPEGAGAGAGPEGAGAGAGPEGAGAGAGPEGAGAGAGPEGAGAGAGPEGAGA
GAGPEGAGAGAGPEGAGAGAGPEGARMPTSAPQMLRELQETNAALQDVRELLRQQVK
EITFLKNTVMESDASGTSYRDPMGAGAGAGPEGAGAGAGPEGAGAGAGPEGAGAGAG

PEGAGAGAGPEGAGAGAGPEGAGAGAGPEGAGAGAGPEGAGAGAGPEGAGAGAGPE
GARMPTSLEDPAANKARKEAELAAATAEQ*

PPxY

DNA sequence:

ATGGGCAGCAGCCATCATCATCATCACAGCAGCGGCCTGGTGCCGCGCGGCAG
CCATATGGCTAGCGAGTATCCGCCGTATCCGCCGCCCGCGTATCCGAGCGGTTCTAG
AGTGCCGGGTAGCGGTGTTCCGGGTAGCGGTGTTCCGGGTAGCGGTGTGCCGGGTAG
GCGGCGTTCCGGGTAGCGGCGTGCCTGGTAGCGGCGTTCTGGTAGCGGCGTGCCC
GGTAGCGGCGTTCCCGGTAGCGGCGTGCCAGGTAGCGGCGTTCCAGGTAGCGGCGT
GCCGGGTAGCGGCGTTCCCGGTAGCGGCGTGCCGGGTAGCGGCGTTCCCGGTAGCG
GCGTGCCAGGTAGCGGCGTTCCGGGTAGCGGCGTGCCAGGTAGCGGCGTTCTGGT
AGCGGCGTGCCAGGTAGCGGCGTTCCCGGTAGCGGCGTGCCAGGTAGCGGCGTTCC
AGGTAGCGGTGTGCCGGGTAGCGGTGTGCCGGGTAGCGGTGAATATCCGCCGTATC
CGCCGCCGCCGTATCCGAGCACTAGAGTGCCGGGTAGCGGTGTTCCGGGTAGCGGT
GTTCCGGGTAGCGGTGTGCCGGGTAGCGGCGTTCCGGGTAGCGGCGTGCCCTGGTAG
CGGCGTTCTGGTAGCGGCGTGCCCGGTAGCGGCGTTCCCGGTAGCGGCGTGCCAG
GTAGCGGCGTTCCAGGTAGCGGCGTGCCGGGTAGCGGCGTTCCCGGTAGCGGCGTG
CCGGGTAGCGGCGTTCCCGGTAGCGGCGTGCCAGGTAGCGGCGTTCCGGGTAGCGG
CGTGCCAGGTAGCGGCGTTCTGGTAGCGGCGTGCCAGGTAGCGGCGTTCCCGGTAG
GCGGCGTGCCAGGTAGCGGCGTTCCAGGTAGCGGTGTGCCGGGTAGCGGTGTGCCG
GGTAGCGGTGAATATCCGCCGTATCCGCCGCCGCCGTATCCGAGCACTAGTCTCGAG
GATCCGGCTGCTAACAAGCCCGAAAGGAAGCTGAGTTGGCTGCTGCCACCGCTGA
GCAATAA

Amino acid sequence:

MGSSHHHHHSSGLVPRGSHMASEYPPYPPPYPSGSRVPGSGVPGSGVPGSGVPGSGV
PGSGVPGSGVPGSGVPGSGVPGSGVPGSGVPGSGVPGSGVPGSGVPGSGVPGSGV
VPGSGVPGSGVPGSGVPGSGVPGSGVPGSGVPGSGVPGSGVPGSGEYPPYPPPYPSTRV
PGSGVPGSGVPGSGVPGSGVPGSGVPGSGVPGSGVPGSGVPGSGVPGSGVPGSGV
VPGSGVPGSGVPGSGVPGSGVPGSGVPGSGVPGSGVPGSGVPGSGVPGSGVPGSGV
VPGSGEYPPYPPPYPSTSLDPAANKARKEAELAAATAEQ*

Tag

DNA sequence:

ATGGGCAGCAGCCATCATCATCATCACAGCAGCGGCCTGGTGCCGCGCGGCAG
CCATATGGCTAGCGCGCACATCGTTATGGTGGACGCGTACAAGCCGACCAAGTCTA
GAGTTCCGGGCAGCGGTGTTCCGGGTAGCGGTGTTCCGGGCAGCGGTGTGCCGGGT
AGCGGCGTTCCGGGTAGCGGCGTGCTGGTAGCGGCGTTCTGGTAGCGGCGTGCC
CGGTAGCGGCGTTCCCGGTAGCGGCGTGCCAGGTAGCGGCGTTCCAGGTAGCGGCG
TGCCGGGTAGCGGCGTTCCCGGTAGCGGCGTGCCGGGTAGCGGCGTTCCCGGTAGC
GGCGTGCCAGGTAGCGGCGTTCCGGGTAGCGGCGTGCCAGGTAGCGGCGTTCTGG
TAGCGGCGTGCCAGGTAGCGGCGTTCCCGGTAGCGGCGTGCCAGGTAGCGGCGTTC

CAGGTAGCGGTGTGCCGGGTAGCGGTGTTCCGGGTAGCGGTGCGCATATCGTGATG
GTGGACGCGTACAAGCCGACCAAGACTAGAGTTCCGGGCAGCGGTGTTCCGGGTAG
CGGTGTTCCGGGCAGCGGTGTGCCGGGTAGCGGCGTTCCGGGTAGCGGCGTGCCTG
GTAGCGGCGTTTCTGGTAGCGGCGTGCCGGGTAGCGGCGTTCCCGGTAGCGGCGTG
CCAGGTAGCGGCGTTCCAGGTAGCGGCGTGCCGGGTAGCGGCGTTCCCGGTAGCGG
CGTGCCGGGTAGCGGCGTTCCCGGTAGCGGCGTGCCAGGTAGCGGCGTTCCGGGTAG
GCGGCGTGCCAGGTAGCGGCGTTTCTGGTAGCGGCGTGCCAGGTAGCGGCGTTCCC
GGTAGCGGCGTGCCAGGTAGCGGCGTTCCAGGTAGCGGTGTGCCGGGTAGCGGTGT
TCCGGGTAGCGGTGCGCATATCGTGATGGTGGACGCGTACAAGCCGACCAAGACTA
GTCTCGAGGATCCGGCTGCTAACAAAGCCCGAAAGGAAGCTGAGTTGGCTGCTGCC
ACCGCTGAGCAATAA

Amino acid sequence:

MGSSHHHHHSSGLVPRGSHMASAHIVMVDAYKPTKSRVPGSGVPGSGVPGSGVPGSG
VPGSGVPGSGVPGSGVPGSGVPGSGVPGSGVPGSGVPGSGVPGSGVPGSGVPGSG
VPGSGVPGSGVPGSGVPGSGVPGSGVPGSGVPGSGVPGSGVPGSGVPGSGVPGSGV
KTRVPGSGVPGSGVPGSGVPGSGVPGSGVPGSGVPGSGVPGSGVPGSGVPGSGVPGSGV
PGSGVPGSGVPGSGVPGSGVPGSGVPGSGVPGSGVPGSGVPGSGVPGSGVPGSGVPGSG
VPGSGVPGSGAHIVMVDAYKPTKTSLEDPAANKARKEAELAAATAEQ*

ZE

DNA sequence:

ATGGGCAGCAGCCATCATCATCATCACAGCAGCGGCCTGGTGCCGCGCGGCAG
CCATATGGCTAGCCTGGAGATTGAGGCGGCGGCGCTGGAACAAGAAAATACCGCGC
TGAAACCGAAGTGGCGGAAGTGGAGCAGGAAGTGCAACGTCTGGAGAACATCGTT
AGCCAGTACCGTACCCGTTATGGTCCGCTGTCTAGAGTGCCGGGTAGCGGCGTTCCG
GGTAGCGGCGTGCTGGTAGCGGCGTTTCTGGTAGCGGCGTGCCGGGTAGCGGCGTT
CCCGGTAGCGGCGTGCCAGGTAGCGGCGTTCCAGGTAGCGGCGTGCCGGGTAGCGG
CGTTCCCGGTAGCGGCGTGCCGGGTAGCGGCGTTCCCGGTAGCGGCGTGCCAGGTAG
GCGGCGTTCCGGGTAGCGGCGTGCCAGGTAGCGGCGTTTCTGGTAGCGGCGTGCCA
GGTAGCGGCGTTCCCGGTAGCGGCGTGCCAGGTAGCGGCGTTCCAGGTAGCGGCGT
GCCGGGTAGCGGCGTTCCAGGTAGCGGCGTGCCAGGTAGCGGCGTTCCAGGTAGCG
GCGTGCCAGGTAGCGGTCTGGAGATTGAAGCGGCGGCGCTGGAGCAAGAAAACACC
GCGCTGGAAACCGAGGTGGCGGAGCTGGAACAAGAGGTTCAACGTCTGGAAAACAT
CGTGAGCCAATACCGCACCCGCTATGGTCCGCTGACTAGAGTGCCGGGTAGCGGCG
TTCCGGGTAGCGGCGTGCTGGTAGCGGCGTTTCTGGTAGCGGCGTGCCCGGTAGCG
GCGTTCCCGGTAGCGGCGTGCCAGGTAGCGGCGTTCCAGGTAGCGGCGTGCCGGGT
AGCGGCGTTCCCGGTAGCGGCGTGCCGGGTAGCGGCGTTCCCGGTAGCGGCGTGCC
AGGTAGCGGCGTTCCGGGTAGCGGCGTGCCAGGTAGCGGCGTTTCTGGTAGCGGCG
TGCCAGGTAGCGGCGTTCCCGGTAGCGGCGTGCCAGGTAGCGGCGTTCCAGGTAGC
GGCGTGCCGGGTAGCGGCGTTCCAGGTAGCGGCGTGCCAGGTAGCGGCGTTCCAGG
TAGCGGCGTGCCAGGTAGCGGTCTGGAGATTGAAGCGGCGGCGCTGGAGCAAGAAA
ACACCGCGCTGGAAACCGAGGTGGCGGAGCTGGAACAAGAGGTTCAACGTCTGGAA

A

Amino acid sequence:

KARKEAELAAATAEQ*

ZR

DNA sequence:

CTGCTAACAAAGCCCGAAAGGAAGCTGAGTTGGCTGCTGCCACCGCTGAGCAATAA

Amino acid sequence:

[illegible]

SGVPGSGVPGSGVPGSGLEIRAAALRRRNTALRTRVAELRQQRVQRLRNEVSQYETRYGP
LTRVPGSGVPGSGVPGSGVPGSGVPGSGVPGSGVPGSGVPGSGVPGSGVPGSGVPGSGV
PGSGVPGSGVPGSGVPGSGVPGSGVPGSGVPGSGVPGSGVPGSGVPGSGVPGSGVPGSGV
VPGSGVPGSGLEIRAAALRRRNTALRTRVAELRQQRVQRLRNEVSQYETRYGPLTSLEDP
AANKARKEAELAAATAEQ*

B. DNA sequences of vectors (plasmid maps of all vectors with mCherry in Section D)

Also available at: <https://github.com/olsenlabmit/Democratizing-Protein-Expression>

Highlights indicate cloning sites:

BamHI

NdeI

XhoI

BglII

HindIII

pET-15b

CTCGAGGATCCGGCTGCTAACAAAGCCCGAAAGGAAGCTGAGTTGGCTGCTGCCAC
CGCTGAGCAATAACTAGCATAACCCCTTGGGGCCTCTAAACGGGTCTTGAGGGGTTT
TTTGCTGAAAGGAGGAAGTATATCCGGATATCCCGCAAGAGGCCCGGCAGTACCGG
CATAACCAAGCCTATGCCTACAGCATCCAGGGTGACGGTGCCGAGGATGACGATGA
GCGCATTGTTAGATTTTCATACACGGTGCCTGACTGCGTTAGCAATTTAACTGTGATA
AACTACCGCATTAAGCTTATCGATGATAAGCTGTCAAACATGAGAATTCTTGAAGA
CGAAAGGGCCTCGTGATACGCCTATTTTTATAGGTTAATGTCATGATAATAATGGTT
TCTTAGACGTCAGGTGGCACTTTTCGGGGAAATGTGCGCGGAACCCCTATTTGTTTA
TTTTTCTAAATACATTCAAATATGTATCCGCTCATGAGACAATAACCCTGATAAATG
CTTCAATAATATTGAAAAAGGAAGAGTATGAGTATTCAACATTTCCGTGTCGCCCTT
ATCCCTTTTTTTCGCGCATTTTGCCTTCCTGTTTTTGTCTACCCAGAAACGCTGGTGA
AAGTAAAAGATGCTGAAGATCAGTTGGGTGCACGAGTGGGTACATCGAACTGGAT
CTCAACAGCGGTAAGATCCTTGAGAGTTTTTCGCCCCGAAGAACGTTTTCCAATGATG
AGCACTTTTAAAGTTCTGCTATGTGGCGCGGTATTATCCCGTGTTGACGCCGGGCAA
GAGCAACTCGGTCGCCGCATACACTATTCTCAGAATGACTTGGTTGAGTACTACCA
GTCACAGAAAAGCATCTTACGGATGGCATGACAGTAAGAGAATTATGCAGTGCTGC
CATAACCATGAGTGATAAACAACACTGCGGCCAACTTACTTCTGACAACGATCGGAGGAC
CGAAGGAGCTAACCGCTTTTTTGCACAACATGGGGGATCATGTAACCTCGCCTTGATC
GTTGGGAACCGGAGCTGAATGAAGCCATACCAAACGACGAGCGTGACACCACGATG
CCTGCAGCAATGGCAACAACGTTGCGCAAACCTATTAACCTGGCGAACTACTTACTCTA
GCTTCCCGGCAACAATTAATAGACTGGATGGAGGCGGATAAAGTTGCAGGACCACT
TCTGCGCTCGGCCCTTCCGGCTGGCTGGTTTATTGCTGATAAATCTGGAGCCGGTGA
GCGTGGGTCTCGCGGTATCATTGCAGCACTGGGGCCAGATGGTAAGCCCTCCCGTAT
CGTAGTTATCTACACGACGGGGAGTCAGGCAACTATGGATGAACGAAATAGACAGA

TCGCTGAGATAGGTGCCTCACTGATTAAGCATTGGTAACTGTCAGACCAAGTTTACT
CATATATACTTTAGATTGATTTAAACTTCATTTTTAATTTAAAAGGATCTAGGTGAA
GATCCTTTTTGATAATCTCATGACCAAAATCCCTTAACGTGAGTTTTTCGTTCCACTGA
GCGTCAGACCCCGTAGAAAAGATCAAAGGATCTTCTTGAGATCCTTTTTTTCTGCGC
GTAATCTGCTGCTTGCAAACAAAAAAACCACCGCTACCAGCGGTGGTTTGTGTTGCCG
GATCAAGAGCTACCAACTCTTTTTCCGAAGGTAAGTGGCTTCAGCAGAGCGCAGATA
CCAAATACTGTCCTTCTAGTGTAGCCGTAGTTAGGCCACCACTTCAAGAACTCTGTA
GCACCGCCTACATACCTCGCTCTGCTAATCCTGTTACCAGTGGCTGCTGCCAGTGGC
GATAAGTCGTGTCTTACCGGGTTGGACTCAAGACGATAGTTACCGGATAAGGCGCA
GCGGTGCGGGCTGAACGGGGGGTTCGTGCACACAGCCCAGCTTGGAGCGAACGACCT
ACACCGAACTGAGATACCTACAGCGTGAGCTATGAGAAAGCGCCACGCTTCCCGAA
GGGAGAAAGGCGGACAGGTATCCGGTAAGCGGCAGGGTCGGAACAGGAGAGCGCA
CGAGGGAGCTTCCAGGGGGAAACGCCTGGTATCTTTATAGTCCTGTCGGGTTTCGCC
ACCTCTGACTTGAGCGTCGATTTTTGTGATGCTCGTCAGGGGGGCGGAGCCTATGGA
AAAACGCCAGCAACGCGGCCTTTTTACGGTTCCTGGCCTTTTGCTGGCCTTTTGCTCA
CATGTTCTTTTCTGCGTTATCCCCTGATTCTGTGGATAACCGTATTACCGCCTTTGAG
TGAGCTGATACCGCTCGCCGCAGCCGAACGACCGAGCGCAGCGAGTCAGTGAGCGA
GGAAGCGGAAGAGCGCCTGATGCGGTATTTTCTCCTTACGCATCTGTGCGGTATTTT
ACACCGCATATATGGTGCCTCTCAGTACAATCTGCTCTGATGCCGCATAGTTAAGC
CAGTATACACTCCGCTATCGCTACGTGACTGGGTCTATGGCTGCGCCCCGACACCCGC
CAACACCCGCTGACGCGCCCTGACGGGCTTGTCTGCTCCCGGCATCCGCTTACAGAC
AAGCTGTGACCGTCTCCGGGAGCTGCATGTGTGTCAGAGGTTTTACCGTTCATACCGA
AACGCGCGAGGCAGCTGCGGTAAAGCTCATCAGCGTGGTCGTGAAGCGATTACAG
ATGTCTGCCTGTTTCATCCGCGTCCAGCTCGTTGAGTTTCTCCAGAAGCGTTAATGTCT
GGCTTCTGATAAAGCGGGCCATGTTAAGGGCGGTTTTTTCTGTTTGGTCACTGATG
CCTCCGTGTAAGGGGGATTTCTGTTTCATGGGGGTAATGATACCGATGAAACGAGAG
AGGATGCTCACGATACGGGTACTGATGATGAACATGCCCGGTTACTGGAACGTTGT
GAGGGTAAACAACTGGCGGTATGGATGCGGCGGGACCAGAGAAAAATCACTCAGG
GTCAATGCCAGCGCTTCGTTAATACAGATGTAGGTGTTCCACAGGGTAGCCAGCAGC
ATCCTGCGATGCAGATCCGGAACATAATGGTGCAGGGCGCTGACTTCCGCGTTTCCA
GACTTTACGAAACACGGAAACCGAAGACCATTTCATGTTGTTGCTCAGGTTCGCAGAC
GTTTTGCAGCAGCAGTCGCTTCACGTTTCGCTCGCGTATCGGTGATTTCATTCTGCTAAC
CAGTAAGGCAACCCCGCCAGCCTAGCCGGGTCTCAACGACAGGAGCACGATCATG
CGCACCCGTGGCCAGGACCCAACGCTGCCCCGAGATGCGCCGCGTGCGGCTGCTGGA
GATGGCGGACGCGATGGATATGTTCTGCCAAGGGTTGGTTTGCGCATTACAGTTCT
CCGCAAGAATTGATTGGCTCCAATTCTTGGAGTGGTGAATCCGTTAGCGAGGTGCCG
CCGGCTTCCATTCAGGTTCGAGGTGGCCCCGGCTCCATGCACCGCGACGCAACGCGGG
GAGGCAGACAAGGTATAGGGCGGCGCCTACAATCCATGCCAACCCGTTCCATGTGC
TCGCCGAGGCGGCATAAATCGCCGTGACGATCAGCGGTCCAGTGATCGAAGTTAGG
CTGGTAAGAGCCGCGAGCGATCCTTGAAGCTGTCCCTGATGGTCGTCTATCTACCTGC
CTGGACAGCATGGCCTGCAACGCGGGCATCCCGATGCCGCCGGAAGCGAGAAGAAT
CATAATGGGGAAGGCCATCCAGCCTCGCGTCGCGAACGCCAGCAAGACGTAGCCCA
GCGCGTCGGCCGCCATGCCGGCGATAATGGCCTGCTTCTCGCCGAAACGTTTGGTGG

CGGGACCAGTGACGAAGGCTTGAGCGAGGGCGTGCAAGATTCCGAATACCGCAAGC
GACAGGCCGATCATCGTCGCGCTCCAGCGAAAGCGGTCTCGCCGAAAATGACCCA
GAGCGCTGCCGGCACCTGTCTACGAGTTGCATGATAAAGAAGACAGTCATAAGTG
CGGCGACGATAGTCATGCCCCGCGCCACCGGAAGGAGCTGACTGGGTTGAAGGCT
CTCAAGGGCATCGGTTCGAGATCCCGGTGCCTAATGAGTGAGCTAACTTACATTAATT
GCGTTGCGCTCACTGCCCCGCTTTCCAGTCGGGAAACCTGTCGTGCCAGCTGCATTAA
TGAATCGGCCAACGCGCGGGGAGAGGCGGTTTTCGTATTGGGCGCCAGGGTGGTTT
TTCTTTTACCAGTGAGACGGGCAACAGCTGATTGCCCTTCACCGCCTGGCCCTGAG
AGAGTTGCAGCAAGCGGTCCACGCTGGTTTGCCCCAGCAGGCGAAAATCCTGTTTGA
TGGTGGTTAACGGCGGGATATAACATGAGCTGTCTTCGGTATCGTCGTATCCCACTA
CCGAGATATCCGCACCAACGCGCAGCCCGGACTCGGTAATGGCGCGCATTGCGCCC
AGCGCCATCTGATCGTTGGCAACCAGCATCGCAGTGGAACGATGCCCTCATTCAGC
ATTTGCATGGTTTGTGAAAACCGGACATGGCACTCCAGTCGCCTTCCCGTTCCGCT
ATCGGCTGAATTTGATTGCGAGTGAGATATTTATGCCAGCCAGCCAGACGCAGACGC
GCCGAGACAGAACTTAATGGGCCCCGCTAACAGCGCGATTTGCTGGTGACCCAATGC
GACCAGATGCTCCACGCCCAGTCGCGTACCGTCTTCATGGGAGAAAATAATACTGTT
GATGGGTGTCTGGTCAGAGACATCAAGAAATAACGCCGGAACATTAGTGAGGCAG
CTTCCACAGCAATGGCATCCTGGTCATCCAGCGGATAGTTAATGATCAGCCCACTGA
CGCGTTGCGCGAGAAGATTGTGCACCGCCGCTTTACAGGCTTCGACGCCGCTTCGTT
CTACCATCGACACCACCACGCTGGCACCCAGTTGATCGGCGCGAGATTTAATCGCCC
CGACAATTTGCGACGGCGCGTGACAGGGCCAGACTGGAGGTGGCAACGCCAATCAGC
AACGACTGTTTGCCCCGCCAGTTGTTGTGCCACGCGGTTGGGAATGTAATTCAGCTCC
GCCATCGCCGCTTCCACTTTTTCCCGCGTTTTTCGCAGAAACGTGGCTGGCCTGGTTCA
CCACGCGGGAAACGGTCTGATAAGAGACACCGGCATACTCTGCGACATCGTATAAC
GTTACTGGTTTTCACATTCACCACCCTGAATTGACTCTCTTCCGGGCGCTATCATGCCA
TACCGCGAAAGGTTTTGCGCCATTCGATGGTGTCCGGGATCTCGACGCTCTCCCTTA
TGCGACTCCTGCATTAGGAAGCAGCCCAGTAGTAGGTTGAGGCCGTTGAGCACCGC
CGCCGCAAGGAATGGTGCATGCAAGGAGATGGCGCCCAACAGTCCCCCGGCCACGG
GGCCTGCCACCATACCACGCGCGAAACAAGCGCTCATGAGCCCGAAGTGCGGAGCC
CGATCTTCCCCATCGGTGATGTCGGCGATATAGGCGCCAGCAACCGCACCTGTGGCG
CCGGTGATGCCGGCCACGATGCGTCCGGCGTAGAGGATCGAGATCTCGATCCCGCG
AAATTAATACGACTCACTATAGGGGAATTGTGAGCGGATAACAATTCCCCTCTAGAA
ATAATTTTGTTTAACTTTAAGAAGGAGATATACCATGGGCAGCAGCCATCATCATCA
TCATCACAGCAGCGGCCTGGTGCCGCGCGGCAGCCATAC

pET-22b(+)

CTCGAGCACCACCACCACCACCTGAGATCCGGCTGCTAACAAAGCCCGAAAGGA
AGCTGAGTTGGCTGCTGCCACCGCTGAGCAATAACTAGCATAAACCCTTGGGGCCTC
TAAACGGGTCTTGAGGGGTTTTTTGCTGAAAGGAGGAACTATATCCGGATTGGCGAA
TGGGACGCGCCCTGTAGCGGCGCATTAAAGCGCGGCGGGTGTGGTGGTTACGCGCAG
CGTGACCGCTACACTTGCCAGCGCCCTAGCGCCCGCTCCTTTTCGCTTTCTTCCCTTCC
TTTCTCGCCACGTTTCGCCGGCTTTCCCCGTCAAGCTCTAAATCGGGGGCTCCCTTTAG
GGTTCCGATTTAGTGCTTTACGGCACCTCGACCCCAAAAACTTGATTAGGGTGATG
GTTACGTAGTGGGCCATCGCCCTGATAGACGGTTTTTCGCCCTTTGACGTTGGAGT

CCACGTTCTTTAATAGTGGACTCTTGTTCCAAACTGGAACAACACTCAACCCTATCTC
GGTCTATTCTTTTGATTTATAAGGGATTTTGCCGATTTGCGCCTATTGGTTAAAAAAT
GAGCTGATTTAACAAAAATTTAACGCGAATTTTAACAAAATATTAACGTTTACAATT
TCAGGTGGCACTTTTCGGGGAAATGTGCGCGGAACCCCTATTTGTTTATTTTCTAAA
TACATTCAAATATGTATCCGCTCATGAGACAATAACCCTGATAAATGCTTCAATAAT
ATTGAAAAAGGAAGAGTATGAGTATTCAACATTTCCGTGTCGCCCTTATTCCCTTTTT
TGCGGCATTTTGCCCTTCCTGTTTTTGCTCACCCAGAAACGCTGGTGAAAGTAAAAGA
TGCTGAAGATCAGTTGGGTGCACGAGTGGGTTACATCGAACTGGATCTCAACAGCG
GTAAGATCCTTGAGAGTTTTCGCCCCGAAGAACGTTTTCCAATGATGAGCACTTTTA
AAGTTCTGCTATGTGGCGCGGTATTATCCCGTATTGACGCCGGGCAAGAGCAACTCG
GTCGCCGCATACACTATTCTCAGAATGACTTGGTTGAGTACTCACCAGTCACAGAAA
AGCATCTTACGGATGGCATGACAGTAAGAGAATTATGCAGTGCTGCCATAACCATG
AGTGATAACACTGCGGCCAACTTACTTCTGACAACGATCGGAGGACCGAAGGAGCT
AACCGCTTTTTTGCACAACATGGGGGATCATGTAACCTCGCCTTGATCGTTGGGAACC
GGAGCTGAATGAAGCCATACCAAACGACGAGCGTGACACCACGATGCCTGCAGCAA
TGGCAACAACGTTGCGCAAACCTATTAACCTGGCGAACTACTTACTCTAGCTTCCCGGC
AACAAATTAATAGACTGGATGGAGGCGGATAAAGTTGCAGGACCACTTCTGCGCTCG
GCCCTTCCGGCTGGCTGGTTTATTGCTGATAAATCTGGAGCCGGTGAGCGTGGGTCT
CGCGGTATCATTGCAGCACTGGGGCCAGATGGTAAGCCCTCCCGTATCGTAGTTATC
TACACGACGGGGAGTCAGGCAACTATGGATGAACGAAATAGACAGATCGCTGAGAT
AGGTGCCTCACTGATTAAGCATTGGTAACCTGTCAGACCAAGTTTACTCATATATACT
TTAGATTGATTTAAAACTTCATTTTTTAATTTAAAAGGATCTAGGTGAAGATCCTTTTT
GATAATCTCATGACCAAAAATCCCTTAACGTGAGTTTTTCGTTCCACTGAGCGTCAGAC
CCCGTAGAAAAGATCAAAGGATCTTCTTGAGATCCTTTTTTTCTGCGCGTAATCTGCT
GCTTGCAAACAAAAAAACCACCGCTACCAGCGGTGGTTTGTGTTGCCGGATCAAGAG
CTACCAACTCTTTTTCCGAAGGTAACCTGGCTTCAGCAGAGCGCAGATACCAAATACT
GTCCTTCTAGTGTAGCCGTAGTTAGGCCACCACTTCAAGAACTCTGTAGCACCGCCT
ACATACCTCGCTCTGCTAATCCTGTTACCAGTGGCTGCTGCCAGTGGCGATAAGTCG
TGTCTTACCGGGTTGGACTCAAGACGATAGTTACCGGATAAGGCGCAGCGGTGCGG
CTGAACGGGGGGTTTCGTGCACACAGCCCAGCTTGGAGCGAACGACCTACACCGAAC
TGAGATACCTACAGCGTGAGCTATGAGAAAGCGCCACGCTTCCCGAAGGGAGAAAG
GCGGACAGGTATCCGGTAAGCGGCAGGGTCGGAACAGGAGAGCGCACGAGGGAGC
TTCCAGGGGGAAACGCCTGGTATCTTTATAGTCCTGTCGGGTTTTCGCCACCTCTGACT
TGAGCGTCGATTTTTGTGATGCTCGTCAGGGGGGCGGAGCCTATGGAAAAACGCCA
GCAACGCGGCCTTTTTACGGTTCCTGGCCTTTTGCTGGCCTTTTGCTCACATGTTCTTT
CCTGCGTTATCCCCTGATTCTGTGGATAACCGTATTACCGCCTTTGAGTGAGCTGATA
CCGCTCGCCGCAGCCGAACGACCGAGCGCAGCGAGTCAGTGAGCGAGGAAGCGGA
AGAGCGCCTGATGCGGTATTTTCTCCTTACGCATCTGTGCGGTATTTACACCCGCATA
TATGGTGCACCTCTCAGTACAATCTGCTCTGATGCCGCATAGTTAAGCCAGTATACAC
TCCGCTATCGCTACGTGACTGGGTCTATGGCTGCGCCCCGACACCCGCCAACACCCGC
TGACGCGCCCTGACGGGCTTGTCTGCTCCCGGCATCCGCTTACAGACAAGCTGTGAC
CGTCTCCGGGAGCTGCATGTGTCAGAGGTTTTACCGTCATCACCGAAACGCGCGAG
GCAGCTGCGGTAAAGCTCATCAGCGTGGTCGTGAAGCGATTACAGATGTCTGCCTG

TTCATCCGCGTCCAGCTCGTTGAGTTTCTCCAGAAGCGTTAATGTCTGGCTTCTGATA
AAGCGGGCCATGTTAAGGGCGGTTTTTTCCTGTTTGGTCACTGATGCCTCCGTGTAA
GGGGGATTTCTGTTCATGGGGGTAATGATACCGATGAAACGAGAGAGGATGCTCAC
GATACGGGTTACTGATGATGAACATGCCCGGTTACTGGAACGTTGTGAGGGTAAAC
AACTGGCGGTATGGATGCGGCGGGACCAGAGAAAAATCACTCAGGGTCAATGCCAG
CGCTTCGTTAATACAGATGTAGGTGTTCCACAGGGTAGCCAGCAGCATCCTGCGATG
CAGATCCGGAACATAATGGTGCAGGGCGCTGACTTCCGCGTTTCCAGACTTTACGAA
ACACGGAAACCGAAGACCATTTCATGTTGTTGCTCAGGTCGCAGACGTTTTTGCAGCAG
CAGTCGCTTCACGTTTCGCTCGCGTATCGGTGATTCTGCTAACCAGTAAGGCAA
CCCCGCCAGCCTAGCCGGGTCCTCAACGACAGGAGCACGATCATGCGCACCCGTGG
GGCCGCCATGCCGGCGATAATGGCCTGCTTCTCGCCGAAACGTTTGGTGGCGGGACC
AGTGACGAAGGCTTGAGCGAGGGCGTGCAAGATTCCGAATACCGCAAGCGACAGGC
CGATCATCGTCGCGCTCCAGCGAAAGCGGTCTCGCCGAAAATGACCCAGAGCGCT
GCCGGCACCTGTCCTACGAGTTGCATGATAAAGAAGACAGTCATAAGTGCGGCGAC
GATAGTCATGCCCCGCGCCCACCGGAAGGAGCTGACTGGGTGAAGGCTCTCAAGG
GCATCGGTTCGAGATCCCGGTGCCTAATGAGTGAGCTAACTTACATTAATTGCGTTGC
GCTCACTGCCCCGCTTTCCAGTCGGGAAACCTGTCGTGCCAGCTGCATTAATGAATCG
GCCAACGCGCGGGGAGAGGGCGGTTTTCGTATTGGGCGCCAGGGTGGTTTTTCTTTTC
ACCAGTGAGACGGGCAACAGCTGATTGCCCTTACCGCCTGGCCCTGAGAGAGTTG
CAGCAAGCGGTCCACGCTGGTTTGCCCCAGCAGGCGAAAATCCTGTTTGATGGTGGT
TAACGGCGGGATATAACATGAGCTGTCTTCGGTATCGTCGTATCCCACTACCGAGAT
ATCCGCACCAACGCGCAGCCCGGACTCGGTAAATGGCGCGCATTGCGCCCAGCGCCA
TCTGATCGTTGGCAACCAGCATCGCAGTGGGAACGATGCCCTCATTACGATTTGCA
TGGTTTGTTGAAAACCGGACATGGCACTCCAGTCGCCTTCCCGTTCCGCTATCGGCT
GAATTTGATTGCGAGTGAGATATTTATGCCAGCCAGCCAGACGCAGACGCGCCGAG
ACAGAACTTAATGGGCCCCGCTAACAGCGCGATTTGCTGGTGACCCAATGCGACCAG
ATGCTCCACGCCAGTCGCGTACCGTCTTCATGGGAGAAAATAATACTGTTGATGGG
TGTCTGGTCAGAGACATCAAGAAATAACGCCGGAACATTAGTGCAGGCAGCTTCCA
CAGCAATGGCATCCTGGTCATCCAGCGGATAGTTAATGATCAGCCCCTGACGCGTT
GCGCGAGAAGATTGTGCACCGCCGCTTTACAGGCTTCGACGCCGCTTCGTTCTACCA
TCGACACCACCACGCTGGCACCCAGTTGATCGGCGCGAGATTTAATCGCCGCGACA
ATTTGCGACGGGCGGTGCAGGGCCAGACTGGAGGTGGCAACGCCAATCAGCAACGA
CTGTTTGCCCGCCAGTTGTTGTGCCACGCGGTTGGGAATGTAATTCAGCTCCGCCAT
CGCCGCTTCCACTTTTTTCCCGCGTTTTTCGCAGAAACGTGGCTGGCCTGGTTACCCAG
CGGGAAACGGTCTGATAAGAGACACCGGCATACTCTGCGACATCGTATAACGTTAC
TGGTTTCACATTCACCACCCTGAATTGACTCTCTTCCGGGCGCTATCATGCCATACCG
CGAAAGGTTTTGCGCCATTCGATGGTGTCCGGGATCTCGACGCTCTCCCTTATGCGA
CTCCTGCATTAGGAAGCAGCCAGTAGTAGGTTGAGGCCGTTGAGCACCGCCGCCG
CAAGGAATGGTGCATGCAAGGAGATGGCGCCCCAACAGTCCCCCGGCCACGGGGCCT
GCCACCATACCCACGCCGAAACAAGCGCTCATGAGCCCGAAGTGGCGAGCCCGATC
TTCCCCATCGGTGATGTGCGCGATATAGGCGCCAGCAACCGCACCTGTGGCGCCGGT
GATGCCGGCCACGATGCGTCCGGCGTAGAGGATCGAGATCTCGATCCCGCGAAATT

AATACGACTCACTATAGGGGAATTGTGAGCGGATAACAATTCCCCTCTAGAAATAAT
TTTGTTTAACTTTAAGAAGGAGATATA **CATATG**

pQE-9

AAGCTTAATTAGCTGAGCTTGGACTCCTGTTGATAGATCCAGTAATGACCTCAGAAC
TCCATCTGGATTTGTTTCAGAACGCTCGGTTGCCGCCGGGCGTTTTTTATTGGTGAGAA
TCCAAGCTAGCTTGGCGAGATTTTCAGGAGCTAAGGAAGCTAAAATGGAGAAAAAA
ATCACTGGATATACCACCGTTGATATATCCCAATGGCATCGTAAAGAACATTTTGTAG
GCATTTCACTCAGTTGCTCAATGTACCTATAACCAGACCGTTCAGCTGGATATTACG
GCCTTTTTTAAAGACCGTAAAGAAAAATAAGCACAAGTTTTATCCGGCCTTTATTAC
ATTCTTGCCCGCCTGATGAATGCTCATCCGGAATTTTCGTATGGCAATGAAAGACGGT
GAGCTGGTGATATGGGATAGTGTTACCCCTTGTTACACCGTTTTTCCATGAGCAAAT
GAAACGTTTTTCATCGCTCTGGAGTGAATACCACGACGATTTCCGGCAGTTTCTACAC
ATATATTTCGAAGATGTGGCGTGTTACGGTGAAAACCTGGCCTATTTCCCTAAAGGG
TTTATTGAGAATATGTTTTTCGTCTCAGCCAATCCCTGGGTGAGTTTCACCAGTTTTG
ATTTAAACGTGGCCAATATGGACAACCTTCTTCGCCCCCGTTTTTCACCATGGGCAAAT
ATTATACGCAAGGCGACAAGGTGCTGATGCCGCTGGCGATTCAGGTTTCATCATGCCG
TTTGTGATGGCTTCCATGTTCGGCAGAATGCTTAATGAATTACAACAGTACTGCGATG
AGTGGCAGGGCGGGGCGTAATTTTTTTAAGGCAGTTATTGGTGCCCTTAAACGCCTG
GGGTAATGACTCTCTAGCTTGAGGCATCAAATAAAACGAAAGGCTCAGTCGAAAGA
CTGGGCCTTTTCGTTTTATCTGTTGTTTGTTCGGTGAACGCTCTCCTGAGTAGGACAAAT
CCGCCCTCTAGAGCTGCCTCGCGCGTTTTCGGTGATGACGGTGAAAACCTCTGACACA
TGCAGCTCCCGGAGACGGTCACAGCTTGTCTGTAAGCGGATGCCGGGAGCAGACAA
GCCCCGTCAGGGCGCGTCAGCGGGTGTTGGCGGGTGTCGGGGCGCAGCCATGACCCA
GTCACGTAGCGATAGCGGAGTGTATACTGGCTTAACCTATGCGGCATCAGAGCAGATT
GTACTGAGAGTGCACCATATGCGGTGTGAAATACCGCACAGATGCGTAAGGAGAAA
ATACCGCATCAGGCGCTCTTCCGCTTCCTCGCTCACTGACTCGCTGCGCTCGGTCTGTT
CGGCTGCGGCGAGCGGTATCAGCTCACTCAAAGGCGGTAATACGGTTATCCACAGA
ATCAGGGGATAACGCAGGAAAGAACATGTGAGCAAAAGGCCAGCAAAAGGCCAGG
AACCGTAAAAAGGCCGCGTTGCTGGCGTTTTTCCATAGGCTCCGCCCCCTGACGAG
CATCACAAAAATCGACGCTCAAGTCAGAGGTGGCGAAACCCGACAGGACTATAAAG
ATACCAGGCGTTTTCCCCCTGGAAGCTCCCTCGTGCGCTCTCCTGTTCCGACCCTGCCG
CTTACCGGATACCTGTCCGCCTTTCTCCCTTCGGGAAGCGTGGCGCTTTTCTCATAGCT
CACGCTGTAGGTATCTCAGTTCGGTGTAGGTCGTTTCGCTCCAAGCTGGGCTGTGTGC
ACGAACCCCCCGTTCAGCCCCGACCGCTGCGCCTTATCCGGTAACTATCGTCTTGAGT
CCAACCCGGTAAGACACGACTTATCGCCACTGGCAGCAGCCACTGGTAACAGGATT
AGCAGAGCGAGGTATGTAGGCGGTGCTACAGAGTTCTTGAAGTGGTGGCCTAACTA
CGGCTACACTAGAAGGACAGTATTTGGTATCTGCGCTCTGCTGAAGCCAGTTACCTT
CGGAAAAAGAGTTGGTAGCTCTTGATCCGGCAAACAAACCACCGCTGGTAGCGGTG
GTTTTTTTGTGTTGCAAGCAGCAGATTACGCGCAGAAAAAAAGGATCTCAAGAAGATC
CTTTGATCTTTTCTACGGGGTCTGACGCTCAGTGGAACGAAAACCTCACGTAAAGGGA
TTTTGGTCATGAGATTATCAAAAAGGATCTTCACCTAGATCCTTTTAAATTAAAAAT
GAAGTTTTTAAATCAATCTAAAGTATATATGAGTAAACTTGGTCTGACAGTTACCAAT
GCTTAATCAGTGAGGCACCTATCTCAGCGATCTGTCTATTTTCGTTTCATCCATAGTTGC

CTGACTCCCCGTCGTGTAGATAACTACGATACGGGAGGGCTTACCATCTGGCCCCAG
TGCTGCAATGATACCGCGAGACCCACGCTCACCGGCTCCAGATTTATCAGCAATAAA
CCAGCCAGCCGGAAGGGCCGAGCGCAGAAAGTGGTCCTGCAACTTTATCCGCCTCCA
TCCAGTCTATTAATTGTTGCCGGAAGCTAGAGTAAGTAGTTTCGCCAGTTAATAGTT
TGCGCAACGTTGTTGCCATTGCTACAGGCATCGTGGTGTACGCTCGTCGTTTGGTAT
GGCTTCATTCAGCTCCGGTTCCCAACGATCAAGGCGAGTTACATGATCCCCCATGTT
GTGCAAAAAAGCGGTTAGCTCCTTCGGTCCTCCGATCGTTGTCAGAAAGTAAGTTGGC
CGCAGTGTTATCACTCATGGTTATGGCAGCACTGCATAATTCTCTTACTGTCATGCCA
TCCGTAAGATGCTTTTTCTGTGACTGGTGAGTACTCAACCAAGTCATTCTGAGAATAG
TGTATGCGGCGACCGAGTTGCTCTTGCCCGGCGTCAATACGGGATAATACCGCGCCA
CATAGCAGAACTTTAAAAGTGCTCATCATTGGAAAACGTTCTTCGGGGCGAAAACTC
TCAAGGATCTTACCGCTGTTGAGATCCAGTTCGATGTAACCCACTCGTGCACCCAAC
TGATCTTCAGCATCTTTTACTTTTACCAGCGTTTCTGGGTGAGCAAAAACAGGAAGG
CAAAATGCCGCAAAAAAGGGAATAAGGGCGACACGGAAATGTTGAATACTCATACT
CTTCCTTTTTCAATATTATTGAAGCATTTATCAGGGTTATTGTCTCATGAGCGGATAC
ATATTTGAATGTATTTAGAAAAATAAACAAATAGGGGTTCGCGCACATTTCCCCGA
AAAGTGCCACCTGACGTCTAAGAAACCATTATTATCATGACATTAACCTATAAAAAT
AGGCGTATCACGAGGCCCTTTCGTCTTCACCTCGAGAAATCATAAAAAATTTATTTG
CTTTGTGAGCGGATAACAATTATAATAGATTCAATTGTGAGCGGATAACAATTTTAC
ACAGAATTCATTAAAGAGGAGAAATTAATATGAGAGGATCGCATCACCATCACCA
TCACGGATCC

pQE-60

CTCGAG AAGCTT AGATCT CATCACCATCACCATCACTAAGCTTAATTAGCTGAGCTT
GGACTCCTGTTGATAGATCCAGTAATGACCTCAGAACTCCATCTGGATTTGTTTCTCAGA
ACGCTCGGTTGCCGCCGGGCGTTTTTTATTGGTGAGAAATCCAAGCTAGCTTGGCGAG
ATTTTCAGGAGCTAAGGAAGCTAAAATGGAGAAAAAAATCACTGGATATAACCACCG
TTGATATATCCCAATGGCATCGTAAAGAACATTTTGGAGGCATTTTCAGTCAGTTGCTC
AATGTACCTATAACCAGACCGTTTCAGCTGGATATTACGGCCTTTTTTAAAGACCGTAA
AGAAAAATAAGCACAAAGTTTATCCGGCCTTTATTACATTCTTGCCCGCCTGATGA
ATGCTCATCCGGAATTTTCGTATGGCAATGAAAGACGGTGAGCTGGTGATATGGGAT
AGTGTTTACCCCTTGTTACACCGTTTTTCCATGAGCAAACTGAAACGTTTTTCATCGCTCT
GGAGTGAATACCACGACGATTTCCGGCAGTTTCTACACATATATTTCGCAAGATGTGG
CGTGTTACGGTGAAAACCTGGCCTATTTCCCTAAAGGGTTTATTGAGAATATGTTTTT
CGTCTCAGCCAATCCCTGGGTGAGTTTACCAGTTTTGATTTAAACGTGGCCAATAT
GGACAACCTTCTTCGCCCCCGTTTTTACCATGCATGGGCAAATATTATACGCAAGGCG
ACAAGGTGCTGATGCCGCTGGCGATTTCAGGTTTCATCATGCCGTCTGTGATGGCTTCC
ATGTCGGCAGAATGCTTAATGAATTACAACAGTACTGCGATGAGTGGCAGGGCGGG
GCGTAATTTTTTTAAGGCAGTTATTGGTGCCCTTAAACGCCTGGGGTAATGACTCTCT
AGCTTGAGGCATCAAATAAAACGAAAGGCTCAGTCGAAAGACTGGGCCTTTTCGTTTT
ATCTGTTGTTTGTTCGGTGAACGCTCTCCTGAGTAGGACAAATCCGCCGCTCTAGAGC
TGCCTCGCGCGTTTTCGGTGATGACGGTGAAAACCTCTGACACATGCAGCTCCCGGAG
ACGGTCACAGCTTGTCTGTAAGCGGATGCCGGGAGCAGACAAGCCCGTCAGGGCGC
GTCAGCGGGTGTGGCGGGTGTTCGGGGCGCAGCCATGACCCAGTCACGTAGCGATA

GCGGAGTGTATACTGGCTTAACTATGCGGCATCAGAGCAGATTGTACTGAGAGTGC
ACCATATGCGGTGTGAAATACCGCACAGATGCGTAAGGAGAAAATACCGCATCAGG
CGCTCTTCCGCTTCCTCGCTCACTGACTCGCTGCGCTCGGTCGTTCCGGCTGCGGCGAG
CGGTATCAGCTCACTCAAAGGCGGTAATACGGTTATCCACAGAATCAGGGGATAAC
GCAGGAAAGAACATGTGAGCAAAAGGCCAGCAAAAGGCCAGGAACCGTAAAAAGG
CCGCGTTGCTGGCGTTTTTCCATAGGCTCCGCCCCCTGACGAGCATCACAAAAATC
GACGCTCAAGTCAGAGGTGGCGAAACCCGACAGGACTATAAAGATAACCAGGCGTTT
CCCCCTGGAAGCTCCCTCGTGCGCTCTCCTGTTCCGACCCTGCCGCTTACCGGATACC
TGTCCGCTTTTCTCCCTTCGGGAAGCGTGCGCTTTTCTCATAGCTCACGCTGTAGGTA
TCTCAGTTCGGTGTAGGTCGTTTCGCTCCAAGCTGGGCTGTGTGCACGAACCCCCCGT
TCAGCCCGACCGCTGCGCCTTATCCGGTAACTATCGTCTTGAGTCCAACCCGGTAAG
ACACGACTTATCGCCACTGGCAGCAGCCACTGGTAACAGGATTAGCAGAGCGAGGT
ATGTAGGCGGTGCTACAGAGTTCTTGAAGTGGTGGCCTAACTACGGCTACACTAGAA
GGACAGTATTTGGTATCTGCGCTCTGCTGAAGCCAGTTACCTTCGGAAAAAGAGTTG
GTAGCTCTTGATCCGGCAAACAAACCCGCTGGTAGCGGTGGTTTTTTTTGTTTGCA
AGCAGCAGATTACGCGCAGAAAAAAGGATCTCAAGAAGATCCTTTGATCTTTTCTA
CGGGGTCTGACGCTCAGTGGAACGAAAACCTCACGTTAAGGGATTTTGGTCATGAGA
TTATCAAAAAGGATCTTCACCTAGATCCTTTTAAATTAATAAATGAAGTTTAAATCA
ATCTAAAGTATATATGAGTAACTTGGTCTGACAGTTACCAATGCTTAATCAGTGAG
GCACCTATCTCAGCGATCTGTCTATTTTCGTTTCATCCATAGTTGCCTGACTCCCCGTCG
TGTAAGATAACTACGATACGGGAGGGGCTTACCATCTGGCCCCAGTGCTGCAATGATAC
CGCGAGACCCACGCTCACCGGCTCCAGATTTATCAGCAATAAACCAGCCAGCCGGA
AGGGCCGAGCGCAGAAGTGGTCCTGCAACTTTATCCGCCTCCATCCAGTCTATTAAT
TGTTGCCGGGAAGCTAGAGTAAGTAGTTCGCCAGTTAATAGTTTGCGCAACGTTGTT
GCCATTGCTACAGGCATCGTGGTGTACGCTCGTCGTTTGGTATGGCTTCATTACGCT
CCGGTTCCCAACGATCAAGGCGAGTTACATGATCCCCCATGTTGTGCAAAAAAGCG
GTTAGCTCCTTCGGTCCTCCGATCGTTGTCAGAAGTAAGTTGGCCGCAGTGTTATCA
CTCATGGTTATGGCAGCACTGCATAATTCTCTTACTGTCATGCCATCCGTAAGATGCT
TTTCTGTGACTGGTGAGTACTCAACCAAGTCATTCTGAGAATAGTGTATGCGGCGAC
CGAGTTGCTCTTGCCCGGCGTCAATACGGGATAATACCGCGCCACATAGCAGAACTT
TAAAAGTGCTCATCATTGGAAAACGTTCTTCGGGGCGAAAACCTCTCAAGGATCTTAC
CGCTGTTGAGATCCAGTTCGATGTAACCCACTCGTGCACCCAACCTGATCTTCAGCAT
CTTTTACTTTACCCAGCGTTTCTGGGTGAGCAAAAACAGGAAGGCAAAATGCCGCAA
AAAAGGGAATAAGGGCGACACGGAAATGTTGAATACTCATACTCTTCCTTTTTCAAT
ATTATTGAAGCATTATCAGGGTTATTGTCTCATGAGCGGATACATATTTGAATGTAT
TTAGAAAAATAAACAAATAGGGGTTCGCGCACATTTCCCCGAAAAGTGCCACCTG
ACGTCTAAGAAACCATTATTATCATGACATTAACCTATAAAAATAGGCGTATCACGA
GGCCCTTTCGTCTTCACCTCGAGAAATCATAAAAAATTTATTTGCTTTGTGAGCGGAT
AACAATTATAATAGATTCAATTGTGAGCGGATAACAATTTACACAGAATTCATTAA
AGAGGAGAAATTAACCATGGGA

GGATCCCATATG

pGEX-4T(1H)

CTCGAGAGATCTCACCACCACCATCACCATGCGGCCGCATCGTGACTGACTGACGAT
CTGCCTCGCGCGTTTCGGTGATGACGGTGAAAACCTCTGACACATGCAGCTCCCGGA

GACGGTCACAGCTTGTCTGTAAGCGGATGCCGGGAGCAGACAAGCCCGTCAGGGCG
CGTCAGCGGGTGTGGCGGGTGTGCGGGGCGCAGCCATGACCCAGTCACGTAGCGAT
AGCGGAGTGTATAATTCTTGAAGACGAAAGGGCCTCGTGATACGCCTATTTTTATAG
GTTAATGTCATGATAATAATGGTTTCTTAGACGTCAGGTGGCACTTTTCGGGGAAAT
GTGCGCGGAACCCCTATTTGTTTATTTTTCTAAATACATTCAAATATGTATCCGCTCA
TGAGACAATAACCCTGATAAATGCTTCAATAATATTGAAAAAGGAAGAGTATGAGT
ATTCAACATTTCCGTGTCGCCCTTATTCCTTTTTTGCGGCATTTTGCCTTCCTGTTTT
TGCTCACCCAGAAACGCTGGTGAAAGTAAAAGATGCTGAAGATCAGTTGGGTGCAC
GAGTGGGTACATCGAACTGGATCTCAACAGCGGTAAGATCCTTGAGAGTTTTCGCC
CCGAAGAACGTTTTCCAATGATGAGCACTTTTAAAGTTCTGCTATGTGGCGCGGTAT
TATCCCGTGTTGACGCCGGGCAAGAGCAACTCGGTGCGCCGCATACACTATTCTCAGA
ATGACTTGGTTGAGTACTCACCAGTCACAGAAAAGCATCTTACGGATGGCATGACA
GTAAGAGAATTATGCAGTGCTGCCATAACCATGAGTGATAAACTGCGGCCAACTT
ACTTCTGACAACGATCGGAGGACCGAAGGAGCTAACCGCTTTTTTGCACAACATGG
GGGATCATGTAACCTCGCCTTGATCGTTGGGAACCGGAGCTGAATGAAGCCATACCA
AACGACGAGCGTGACACCACGATGCCTGCAGCAATGGCAACAACGTTGCGCAAACCT
ATTAACCTGGCGAACTACTTACTCTAGCTTCCCGGCAACAATTAATAGACTGGATGGA
GGCGGATAAAGTTGCAGGACCACTTCTGCGCTCGGCCCTTCCGGCTGGCTGGTTTAT
TGCTGATAAATCTGGAGCCGGTGAGCGTGGGTCTCGCGGTATCATTGCAGCACTGGG
GCCAGATGGTAAGCCCTCCCGTATCGTAGTTATCTACACGACGGGGAGTCAGGCAA
CTATGGATGAACGAAATAGACAGATCGCTGAGATAGGTGCCTCACTGATTAAGCAT
TGGTAACCTGTCAGACCAAGTTTACTCATATATACTTTAGATTGATTTAAAACCTTCATT
TTTAATTTAAAAGGATCTAGGTGAAGATCCTTTTTTGATAATCTCATGACCAAAATCC
CTTAACGTGAGTTTTTCGTTCCACTGAGCGTCAGACCCCGTAGAAAAGATCAAAGGAT
CTTCTTGAGATCCTTTTTTTCTGCGCGTAATCTGCTGCTTGCAAACAAAAAACACC
GCTACCAGCGGTGGTTTGTGGCCGGATCAAGAGCTACCAACTCTTTTTCCGAAGGT
AACTGGCTTCAGCAGAGCGCAGATACCAAATACTGTCCTTCTAGTGTAGCCGTAGTT
AGGCCACCACTTCAAGAACTCTGTAGCACCGCCTACATACCTCGCTCTGCTAATCCT
GTTACCAGTGGCTGCTGCCAGTGGCGATAAGTCGTGTCTTACCGGGTTGGACTCAAG
ACGATAGTTACCGGATAAAGGCGCAGCGGTGCGGCTGAACGGGGGGTTTCGTGCACAC
AGCCCAGCTTGGAGCGAACGACCTACACCGAACTGAGATACCTACAGCGTGAGCTA
TGAGAAAGCGCCACGCTTCCCGAAGGGAGAAAGGCGGACAGGTATCCGGTAAGCG
GCAGGGTCGGAACAGGAGAGCGCACGAGGGAGCTTCCAGGGGGAAACGCCTGGTA
TCTTTATAGTCCTGTCGGGTTTCGCCACCTCTGACTTGAGCGTCGATTTTTGTGATGC
TCGTCAGGGGGGCGGAGCCTATGGAAAAACGCCAGCAACGCGGCCTTTTTACGGTT
CCTGGCCTTTTGCTGGCCTTTTGCTCACATGTTCTTTCCTGCGTTATCCCTGATTCTG
TGGATAACCGTATTACCGCCTTTGAGTGAGCTGATACCGCTCGCCGCAGCCGAACGA
CCGAGCGCAGCGAGTCAGTGAGCGAGGAAGCGGAAGAGCGCCTGATGCGGTATTTT
CTCCTTACGCATCTGTGCGGTATTTACACCCGCATAAATCCGACACCATCGAATGG
TGCAAAACCTTTGCGGGTATGGCATGATAGCGCCCGGAAGAGAGTCAATTCAGGGT
GGTGAATGTGAAACAGTAACGTTATACGATGTGCGCAGAGTATGCCGGTGTCTCTTA
TCAGACCGTTTCCCGCGTGGTGAACCAGGCCAGCCACGTTTCTGCGAAAACGCGGG
AAAAAGTGGAAGCGGCGATGGCGGAGCTGAATTACATTCCCAACCGCGTGGCACAA

CAACTGGCGGGCAAACAGTCGTTGCTGATTGGCGTTGCCACCTCCAGTCTGGCCCTG
CACGCGCCGTCGCAAATTGTCGCGGCGATTAAATCTCGCGCCGATCAACTGGGTGCC
AGCGTGGTGGTGTGCGATGGTAGAACGAAGCGGCGTCGAAGCCTGTAAAGCGGCGGT
GCACAATCTTCTCGCGCAACGCGTCAGTGGGCTGATCATTAATATCCGCTGGATGA
CCAGGATGCCATTGCTGTGGAAGCTGCCTGCACTAATGTTCCGGCGTTATTTCTTGAT
GTCTCTGACCAGACACCCATCAACAGTATTATTTTCTCCCATGAAGACGGTACGCGA
CTGGGCGTGGAGCATCTGGTCGCATTGGGTACCCAGCAAATCGCGCTGTTAGCGGGC
CCATTAAGTTCTGTCTCGGCGCGTCTGCGTCTGGCTGGCTGGCATAAATATCTCACTC
GCAATCAAATTCAGCCGATAGCGGAACGGGAAGGCGACTGGAGTGCCATGTCCGGT
TTTCAACAAACCATGCAAATGCTGAATGAGGGCATCGTTCCCACTGCGATGCTGGTT
GCCAACGATCAGATGGCGCTGGGCGCAATGCGCGCCATTACCGAGTCCGGGCTGCG
CGTTGGTGCGGATATCTCGGTAGTGGGATACGACGATACCGAAGACAGCTCATGTTA
TATCCCGCCGTTAACCACCATCAAACAGGATTTTCGCCTGCTGGGGCAAACCAGCGT
GGACCGCTTGCTGCAACTCTCTCAGGGCCAGGCGGTGAAGGGCAATCAGCTGTTGC
CCGTCTCACTGGTGAAAAGAAAAACCACCCTGGCGCCCAATACGCAAACCGCCTCT
CCCCGCGCGTTGGCCGATTCATTAATGCAGCTGGCACGACAGGTTTCCCGACTGGAA
AGCGGGCAGTGAGCGCAACGCAATTAATGTGAGTTAGCTCACTCATTAGGCACCCC
AGGCTTTACACTTTATGCTTCCGGCTCGTATGTTGTGTGGAATTGTGAGCGGATAAC
AATTTACACAGGAAACAGCTATGACCATGATTACGGATTCCTGGCCGTCGTTTTA
CAACGTCGTGACTGGGAAAACCCTGGCGTTACCCAATTAATCGCCTTGCAGCACAT
CCCCCTTTTCGCCAGCTGGCGTAATAGCGAAGAGGCCCGCACCGATCGCCCTTCCCAA
CAGTTGCGCAGCCTGAATGGCGAATGGCGCTTTGCCTGGTTTCCGGCACCAGAAGCG
GTGCCGGAAGCTGGCTGGAGTGCGATCTTCCTGAGGCCGATACTGTCGTCGTCCCC
TCAAACCTGGCAGATGCACGGTTACGATGCGCCCATCTACACCAACGTAACCTATCCC
ATTACGGTCAATCCGCCGTTTGTTCACGGAGAATCCGACGGGTTGTTACTCGCTC
ACATTTAATGTTGATGAAAGCTGGCTACAGGAAGGCCAGACGCGAATTATTTTTGAT
GGCGTTGGAATTACGTTATCGACTGCACGGTGCACCAATGCTTCTGGCGTCAGGCAG
CCATCGGAAGCTGTGGTATGGCTGTGCAGGTCGTAAATCACTGCATAATTCGTGTGCG
CTCAAGGCGCACTCCCGTTCTGGATAATGTTTTTTGCGCCGACATCATAACGGTTCTG
GCAAATATTCTGAAATGAGCTGTTGACAATTAATCATCGGCTCGTATAATGTGTGGA
ATTGTGAGCGGATAACAATTTACACAGGAAACAGTATTCATGTCCCCTATACTAGG
TTATTGAAAATTAAGGGCCTTGTGCAACCCACTCGACTTCTTTTGAATATCTTGA
AGAAAAATATGAAGAGCATTTGTATGAGCGCGATGAAGGTGATAAATGGCGAAACA
AAAAGTTTGAATTGGGTTTGGAGTTTCCCAATCTTCCTTATTATATTGATGGTGATGT
TAAATTAACACAGTCTATGGCCATCATAACGTTATATAGCTGACAAGCACAAACATGTT
GGGTGGTTGTCCAAAAGAGCGTGACAGAGATTTCAATGCTTGAAGGAGCGGTTTTGG
ATATTAGATACGGTGTTCGAGAATTGCATATAGTAAAGACTTTGAAACTCTCAAAG
TTGATTTTCTTAGCAAGCTACCTGAAATGCTGAAAATGTTTGAAGATCGTTTATGTC
ATAAAACATATTTAAATGGTGATCATGTAACCCATCCTGACTTCATGTTGTATGACG
CTCTTGATGTTGTTTTATACATGGACCCAATGTGCCTGGATGCGTTCCCAAAATTAGT
TTGTTTTAAAAACGTATTGAAGCTATCCACAAATTGATAAGTACTTGAAATCCAG
CAAGTATATAGCATGGCCTTTGCAGGGCTGGCAAGCCACGTTTGGTGGTGGCGACCA
TCCTCCAAAATCGGATCTGGTTCCGCGTGGATCCCATATG

pET-SUMO

CTCGAGAGACAAGCTTAGGTATTTATTCGGCGCAAAGTGCGTCGGGTGATGCTGCCA
ACTTAGTCGAGCACCACCACCACCACCCTGAGATCCGGCTGCTAACAAAGCCCGA
AAGGAAGCTGAGTTGGCTGCTGCCACCGCTGAGCAATAACTAGCATAACCCCTTGG
GGCCTCTAAACGGGTCTTGAGGGGTTTTTTGCTGAAAGGAGGAACCTATATCCGGATT
GGCGAATGGGACGCGCCCTGTAGCGGCGCATTAAAGCGCGGGCGGGTGTGGTGGTTAC
GCGCAGCGTGACCGCTACACTTGCCAGCGCCCTAGCGCCCGCTCCTTTTCGCTTTCTTC
CCTTCCTTTCTCGCCACGTTTCGCCGGCTTTCCCCGTCAAGCTCTAAATCGGGGGGCTCC
CTTTAGGGTTCCGATTTAGTGCTTTACGGCACCTCGACCCCAAAAACTTGATTAGG
GTGATGGTTCACGTAGTGGGCCATCGCCCTGATAGACGGTTTTTTCGCCCTTTGACGTT
GGAGTCCACGTTCTTTAATAGTGGACTCTTGTTCCAACTGGAACAACACTCAACCC
TATCTCGGTCTATTCTTTTGATTTATAAGGGATTTTGCCGATTTTCGGCCTATTGGTTA
AAAAATGAGCTGATTTAACAAAAATTTAACGCGAATTTTAACAAAAATATTAACGCTT
ACAATTTAGGTGGCACTTTTCGGGGAAATGTGCGCGGAACCCCTATTTGTTTATTTTT
CTAAATACATTCAAATATGTATCCGCTCATGAATTAATTCTTAGAAAACTCATCGA
GCATCAAATGAACTGCAATTTATTCATATCAGGATTATCAATACCATATTTTTGAA
AAAGCCGTTTCTGTAATGAAGGAGAAACTCACCGAGGCAGTTCCATAGGATGGCA
AGATCCTGGTATCGGTCTGCGATTCCGACTCGTCCAACATCAATACAACCTATTAAT
TTCCCCTCGTCAAAAATAAGGTTATCAAGTGAGAAATCACCATGAGTGACGACTGA
ATCCGGTGAGAATGGCAAAAGTTTATGCATTTCTTTCCAGACTTGTTCAACAGGCCA
GCCATTACGCTCGTCATCAAAATCACTCGCATCAACCAAACCGTTATTCATTCGTGA
TTGCGCCTGAGCGAGACGAAATACGCGATCGCTGTAAAGGACAATTACAAACAG
GAATCGAATGCAACCGGCGCAGGAACACTGCCAGCGCATCAACAATATTTTCACCT
GAATCAGGATATTCTTCTAATACCTGGAATGCTGTTTTCCCGGGGATCGCAGTGGTG
AGTAACCATGCATCATCAGGAGTACGGATAAAATGCTTGATGGTCGGAAGAGGCAT
AAATTCCGTCAGCCAGTTTAGTCTGACCATCTCATCTGTAACATCATTGGCAACGCT
ACCTTTGCCATGTTTCAGAAACAACCTCTGGCGCATCGGGCTTCCCATAACAATCGATA
GATTGTCGCACCTGATTGCCCGACATTATCGCGAGCCCATTTATACCCATATAAATC
AGCATCCATGTTGGAATTTAATCGCGGCCTAGAGCAAGACGTTTCCCGTTGAATATG
GCTCATAACACCCCTTGTTACTGTTTATGTAAGCAGACAGTTTTATTGTTTCATGAC
CAAAATCCCTTAACGTGAGTTTTTCGTTCCACTGAGCGTCAGACCCCGTAGAAAAGAT
CAAAGGATCTTCTTGAGATCCTTTTTTTCTGCGCGTAATCTGCTGCTTGCAAACAAAA
AAACCACCGCTACCAGCGGTGGTTTTGTTTGCCGGATCAAGAGCTACCAACTCTTTTT
CCGAAGGTAACCTGGCTTCAGCAGAGCGCAGATACCAAATACTGTCCTTCTAGTGTAG
CCGTAGTTAGGCCACCACTTCAAGAACTCTGTAGCACCGCCTACATACCTCGCTCTG
CTAATCCTGTTACCAGTGGCTGCTGCCAGTGGCGATAAGTCGTGTCTTACCGGGTTG
GACTCAAGACGATAGTTACCGGATAAGGCGCAGCGGTTCGGGCTGAACGGGGGGTTG
GTGCACACAGCCCAGCTTGAGCGAACGACCTACACCGAACTGAGATACCTACAGC
GTGAGCTATGAGAAAGCGCCACGCTTCCCGAAGGGAGAAAGGCGGACAGGTATCCG
GTAAGCGGCAGGGTCGGAACAGGAGAGCGCACGAGGGAGCTTCCAGGGGGAAACG
CCTGGTATCTTTATAGTCCTGTTCGGGTTTCGCCACCTCTGACTTGAGCGTCGATTTTT
GTGATGCTCGTCAGGGGGGCGGAGCCTATGGAAAAACGCCAGCAACGCGGCCTTTT
TACGGTTCCTGGCCTTTTGCTGGCCTTTTGCTCACATGTTCTTTCCTGCGTTATCCCCT

GATTCTGTGGATAACCGTATTACCGCCTTTGAGTGAGCTGATACCGCTCGCCGCAGC
CGAACGACCGAGCGCAGCGAGTCAGTGAGCGAGGAAGCGGAAGAGCGCCTGATGC
GGTATTTTCTCCTTACGCATCTGTGCGGTATTTACACCGCAATGGTGCACCTCTCAGT
ACAATCTGCTCTGATGCCGCATAGTTAAGCCAGTATACTCCGCTATCGCTACGTG
ACTGGGTCATGGCTGCGCCCCGACACCCGCCAACACCCGCTGACGCGCCCTGACGG
GCTTGTCTGCTCCCGGCATCCGCTTACAGACAAGCTGTGACCGTCTCCGGGAGCTGC
ATGTGTCAGAGGTTTTACCGTCATCACCGAAACGCGCGAGGGCAGCTGCGGTAAAG
CTCATCAGCGTGGTTCGTGAAGCGATTACAGATGTCTGCCTGTTTCATCCGCGTCCAG
CTCGTTGAGTTTCTCCAGAAGCGTTAATGTCTGGCTTCTGATAAAGCGGGCCATGTT
AAGGGCGGTTTTTTCCTGTTTGGTCACTGATGCCTCCGTGTAAGGGGGATTCTGTTC
ATGGGGGTAATGATACCGATGAAACGAGAGAGGATGCTCACGATACGGGTACTGA
TGATGAACATGCCCGGTTACTGGAACGTTGTGAGGGTAAACAACCTGGCGGTATGGA
TGCGGCGGGACCAGAGAAAAATCACTCAGGGTCAATGCCAGCGCTTCGTTAATACA
GATGTAGGTGTTCCACAGGGTAGCCAGCAGCATCCTGCGATGCAGATCCGGAACAT
AATGGTGCAGGGCGCTGACTTCCGCGTTTCCAGACTTTACGAAACACGGAAACCGA
AGACCATTCATGTTGTTGCTCAGGTCGCAGACGTTTTGCAGCAGCAGTCGCTTCAG
TTCGCTCGCGTATCGGTGATTCAATTCTGCTAACCAGTAAGGCAACCCCGCCAGCCTA
GCCGGGTCCTCAACGACAGGAGCACGATCATGCGCACCCGTGGGGCCGCCATGCCG
GCGATAATGGCCTGCTTCTCGCCGAAACGTTTGGTGGCGGGACCAGTGACGAAGGC
TTGAGCGAGGGCGTGCAAGATTCCGAATACCGCAAGCGACAGGCCGATCATCGTCG
CGCTCCAGCGAAAGCGGTCTCTCGCCGAAAATGACCCAGAGCGCTGCCGGCACCTGT
CCTACGAGTTGCATGATAAAGAAGACAGTCATAAGTGCGGCGACGATAGTCATGCC
CCGCGCCCACCGGAAGGAGCTGACTGGGTTGAAGGCTCTCAAGGGCATCGGTGAG
ATCCCGGTGCCTAATGAGTGAGCTAACTTACATTAATTGCGTTGCGCTCACTGCCCC
CTTTCCAGTCGGGAAACCTGTCGTGCCAGCTGCATTAATGAATCGGCCAACGCGCGG
GGAGAGGCGGTTTTCGTATTGGGCGCCAGGGTGGTTTTTCTTTTACCAGTGAGACG
GGCAACAGCTGATTGCCCTTACCGCCTGGCCCTGAGAGAGTTGCAGCAAGCGGTC
CACGCTGGTTTGGCCCAGCAGGCGAAAATCCTGTTTGATGGTGGTTAACGGCGGGAT
ATAACATGAGCTGTCTTCGGTATCGTCGTATCCCACTACCGAGATATCCGCACCAAC
GCGCAGCCCGGACTCGGTAATGGCGCGCATTGCGCCCAGCGCCATCTGATCGTTGGC
AACCAGCATCGCAGTGGGAACGATGCCCTCATTACGATTTGCATGGTTTGTGAAA
ACCGGACATGGCACTCCAGTCGCCTTCCCGTTCCGCTATCGGCTGAATTTGATTGCG
AGTGAGATATTTATGCCAGCCAGCCAGACGACGCGCCGAGACAGAACTTAATG
GGCCCGCTAACAGCGCGATTTGCTGGTGACCCAATGCGACCAGATGCTCCACGCCC
AGTCGCGTACCGTCTTCATGGGAGAAAATAATACTGTTGATGGGTGTCTGGTCAGAG
ACATCAAGAAATAACGCCGGAACATTAGTGACAGGCAGCTTCCACAGCAATGGCATC
CTGGTCATCCAGCGGATAGTTAATGATCAGCCCACTGACGCGTTGCGCGAGAAGATT
GTGCACCGCCGCTTTACAGGCTTCGACGCCGCTTCGTTCTACCATCGACACCACCAC
GCTGGCACCCAGTTGATCGGCGCGAGATTTAATCGCCGCGACAATTTGCGACGGCG
CGTGCAGGGCCAGACTGGAGGTGGCAACGCCAATCAGCAACGACTGTTTGGCCGCC
AGTTGTTGTGCCACGCGGTTGGGAATGTAATTCAGCTCCGCCATCGCCGCTTCCACT
TTTTCCCGCGTTTTTCGCAGAAACGTGGCTGGCCTGGTTCACCACGCGGGAAACGGTC
TGATAAGAGACACCGGCATACTCTGCGACATCGTATAACGTTACTGGTTTCACATTC

ACCACCCTGAATTGACTCTCTTCCGGGCGCTATCATGCCATACCGCGAAAGGTTTTG
CGCCATTCGATGGTGTCCGGGATCTCGACGCTCTCCCTTATGCGACTCCTGCATTAG
GAAGCAGCCCAGTAGTAGGTTGAGGCCGTTGAGCACCGCCGCGCAAGGAATGGTG
CATGCAAGGAGATGGCGCCCAACAGTCCCCCGGCCACGGGGCCTGCCACCATACCC
ACGCCGAAACAAGCGCTCATGAGCCCGAAGTGGCGAGCCCGATCTTCCCCATCGGT
GATGTTCGGCGATATAGGCGCCAGCAACCGCACCTGTGGCGCCGGTGATGCCGGCCA
CGATGCGTCCGGCGTAGAGGATCGAGATCTCGATCCCGCGAAATTAATACGACTCA
CTATAGGGGAATTGTGAGCGGATAACAATTCCCCTCTAGAAATAATTTTGTTTAACT
TTAAGAAGGAGATATACATATGGGCAGCAGCCATCATCATCATCACGGCAGCG
GCCTGGTGCCGCGCGGCAGCGCTAGCATGTTCGGACTCAGAAGTCAATCAAGAAGCT
AAGCCAGAGGTCAAGCCAGAAGTCAAGCCTGAGACTCACATCAATTTAAAGGTGTC
CGATGGATCTTCAGAGATCTTCTTCAAGATCAAAAAGACCACTCCTTTAAGAAGGCT
GATGGAAGCGTTCGCTAAAAGACAGGGTAAGGAAATGGACTCCTTAAGATTCTTGT
ACGACGGTATTAGAATTCAAGCTGATCAGACCCCTGAAGATTTGGACATGGAGGAT
AACGATATTATTGAGGCTCACAGAGAACAGATTGGTGGT

CATATG

pUC-19

GAGATACCTACAGCGTGAGCTATGAGAAAGCGCCACGCTTCCCGAAGGGAGAAAGG
CGGACAGGTATCCGGTAAGCGGCAGGGTCGGAACAGGAGAGCGCACGAGGGAGCT
TCCAGGGGGGAAACGCCTGGTATCTTTATAGTCCTGTCGGGTTTCGCCACCTCTGACTT
GAGCGTCGATTTTTGTGATGCTCGTCAGGGGGGCGGAGCCTATGGAAAAACGCCAG
CAACGCGGCCTTTTTACGGTTCCTGGCCTTTTGCTGGCCTTTTGCTCACATGTTCTTTC
CTGCGTTATCCCCTGATTCTGTGGATAACCGTATTACCGCCTTTGAGTGAGCTGATAC
CGCTCGCCGCAGCCGAACGACCGAGCGCAGCGAGTCAGTGAGCGAGGAAGCGGAA
GAGCGCCCAATACGCAAACCGCCTCTCCCCGCGCGTGGCCGATTCATTAATGCAGC
TGGCACGACAGGTTTCCCGACTGGAAAGCGGGCAGTGAGCGCAACGCAATTAATGT
GAGTTAGCTCACTCATTAGGCACCCAGGCTTTACACTTTATGCTTCCGGCTCGTATG
TTGTGTGGAATTGTGAGCGGATAACAATTTACACAGGAAACAGCTATGACCATGAT
TACGCCAAGCTTGCATGCCTGCAGGTCGACTCTAGAGGATCCCCGGGTACCGAGCTC
GAATTCAGTGGCCGTCGTTTTACAACGTCGTGACTGGGAAAACCTGGCGTTACCCA
ACTTAATCGCCTTGCAGCACATCCCCCTTTCGCCAGCTGGCGTAATAGCGAAGAGGC
CCGCACCGATCGCCCTTCCCAACAGTTGCGCAGCCTGAATGGCGAATGGCGCCTGAT
GCGGTATTTTCTCCTTACGCATCTGTGCGGTATTTACACCCGCATATGGTGCACCTCTC
AGTACAATCTGCTCTGATGCCGCATAGTTAAGCCAGCCCCGACACCCGCCAACACCC
GCTGACGCGCCCTGACGGGCTTGTCTGCTCCCGGCATCCGCTTACAGACAAGCTGTG
ACCGTCTCCGGGAGCTGCATGTGTCAGAGGTTTTACCGTCATACCGAAACGCGCG
AGACGAAAGGGCCTCGTGATACGCCTATTTTTATAGGTTAATGTCATGATAATAATG
GTTTCTTAGACGTCAGGTGGCACTTTTCGGGGAAATGTGCGCGGAACCCCTATTTGT
TTATTTTTCTAAATACATTCAAATATGTATCCGCTCATGAGACAATAACCCTGATAA
ATGCTTCAATAATATTGAAAAAGGAAGAGTATGAGTATTCAACATTTCCGTGTCGCC
CTTATTCCTTTTTTTCGCGCATTTTGCCTTCCTGTTTTTGTCTACCCAGAAACGCTGGT
GAAAGTAAAAGATGCTGAAGATCAGTTGGGTGCACGAGTGGGTACATCGAACTGG
ATCTCAACAGCGGTAAGATCCTTGAGAGTTTTCGCCCCGAAGAACGTTTTCCAATGA
TGAGCACTTTTAAAGTTCTGCTATGTGGCGCGGTATTATCCCGTATTGACGCCGGGC

AAGAGCAACTCGGTCGCCGCATACACTATTCTCAGAATGACTTGGTTGAGTACTCAC
CAGTCACAGAAAAGCATCTTACGGATGGCATGACAGTAAGAGAATTATGCAGTGCT
GCCATAACCATGAGTGATAACACTGCGGCCAACTTACTTCTGACAACGATCGGAGG
ACCGAAGGAGCTAACCGCTTTTTTGCACAACATGGGGGATCATGTAACCTCGCCTTGA
TCGTTGGGAACCGGAGCTGAATGAAGCCATACCAAACGACGAGCGTGACACCACGA
TGCCTGTAGCAATGGCAACAACGTTGCGCAAACCTATTAACCTGGCGAACTACTTACTC
TAGCTTCCCGGCAACAATTAATAGACTGGATGGAGGCGGATAAAGTTGCAGGACCA
CTTCTGCGCTCGGCCCTTCCGGCTGGCTGGTTTATTGCTGATAAATCTGGAGCCGGTG
AGCGTGGGTCTCGCGGTATCATTGCAGCACTGGGGCCAGATGGTAAGCCCTCCCGTA
TCGTAGTTATCTACACGACGGGGAGTCAGGCAACTATGGATGAACGAAATAGACAG
ATCGCTGAGATAGGTGCCTCACTGATTAAGCATTGGTAACCTGTCAGACCAAGTTTAC
TCATATATACTTTAGATTGATTTAAAACCTTCATTTTAAATTTAAAAGGATCTAGGTGA
AGATCCTTTTTTGATAATCTCATGACCAAAATCCCTTAACGTGAGTTTTTCGTTCCACTG
AGCGTCAGACCCCGTAGAAAAGATCAAAGGATCTTCTTGAGATCCTTTTTTTCTGCG
CGTAATCTGCTGCTTGCAAACAAAAAACCCACCGCTACCAGCGGTGGTTTGTGGCC
GGATCAAGAGCTACCAACTCTTTTTCCGAAGGTAACCTGGCTTCAGCAGAGCGCAGAT
ACCAAATACTGTTCTTCTAGTGTAGCCGTAGTTAGGCCACCACTTCAAGAACTCTGT
AGCACCGCCTACATACCTCGCTCTGCTAATCCTGTTACCAGTGGCTGCTGCCAGTGG
CGATAAGTCGTGTCTTACCGGGTTGGACTCAAGACGATAGTTACCGGATAAGGCGC
AGCGGTCGGGCTGAACGGGGGGTTCGTGCACACAGCCCAGCTTGGAGCGAACGACC
TACACCGAACT

C. Sequencing primers

Table S1. Primers used to sequence-verify each gene.

Vector	Forward primer	Reverse primer
pET-15b	T7 (GENEWIZ) TAATACGACTCACTATAGGG	T7-Term (GENEWIZ) GCTAGTTATTGCTCAGCGG
pET-22b(+)	T7 (GENEWIZ)	T7-Term (GENEWIZ)
pQE-9	pQE Type III/IV CGGATAACAATTTACACAG	pQE Reverse GTTCTGAGGTCATTACTGG
pQE-60	pQE Type III/IV	pQE Reverse
pGEX-4T-1(H)	5GEX (GENEWIZ) GGGCTGGCAAGCCACGTTTGGTG	3GEX (GENEWIZ) CCGGGAGCTGCATGTGTCAGAGG
pET-SUMO	T7 (GENEWIZ)	T7-Term (GENEWIZ)

D. Sequence maps for vectors with mCherry

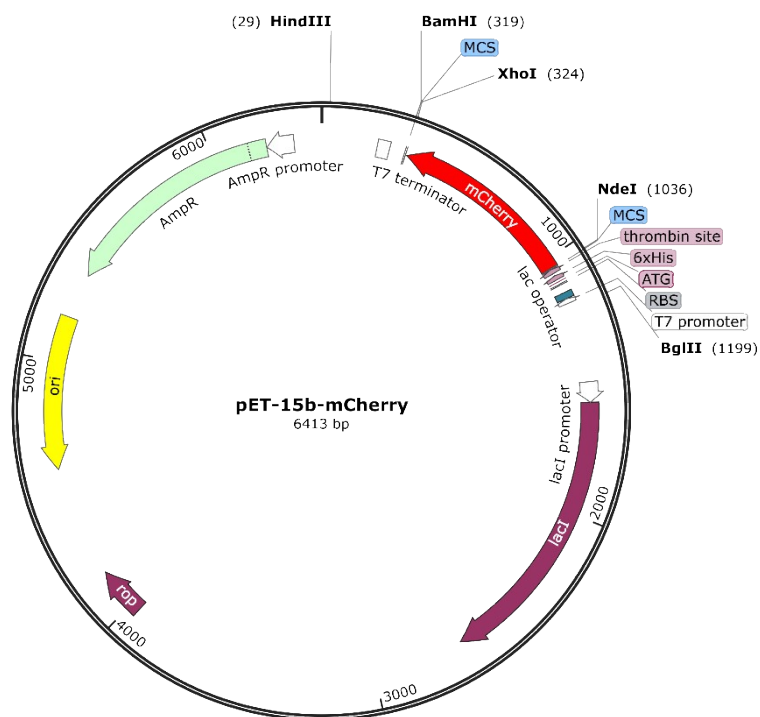


Figure S1. pET-15b with mCherry plasmid map

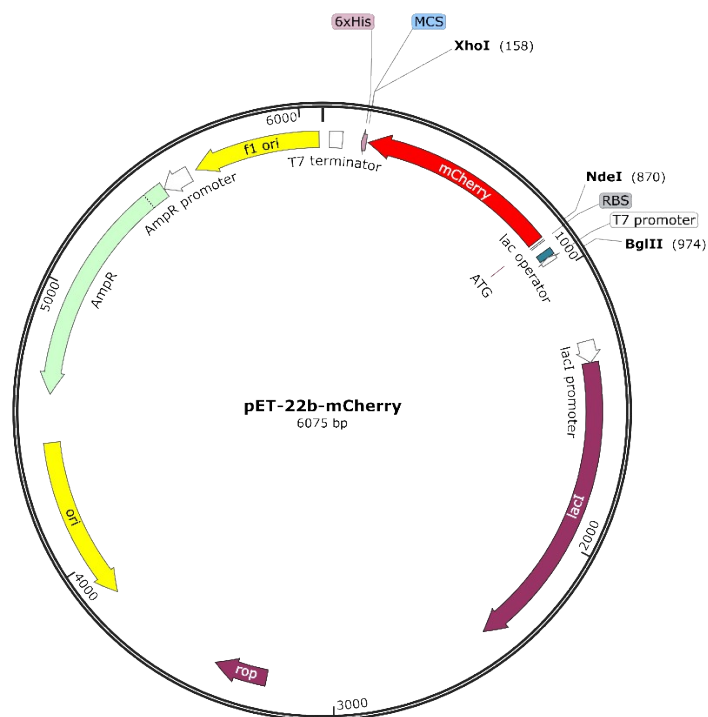


Figure S2. pET-22b(+) with mCherry plasmid map

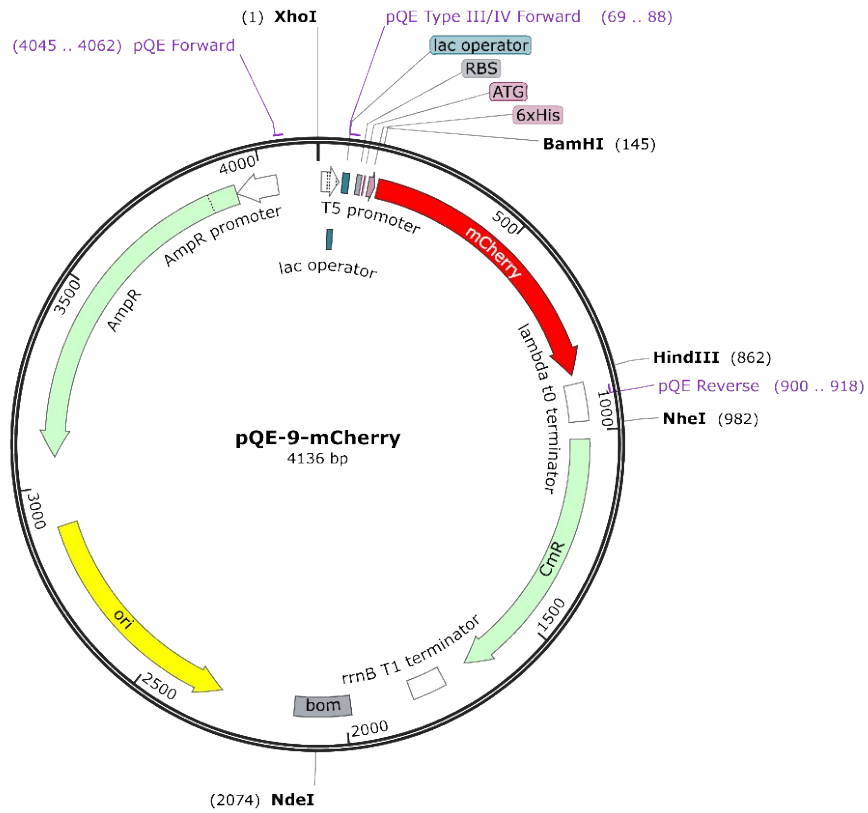


Figure S3. pQE-9 with mCherry plasmid map

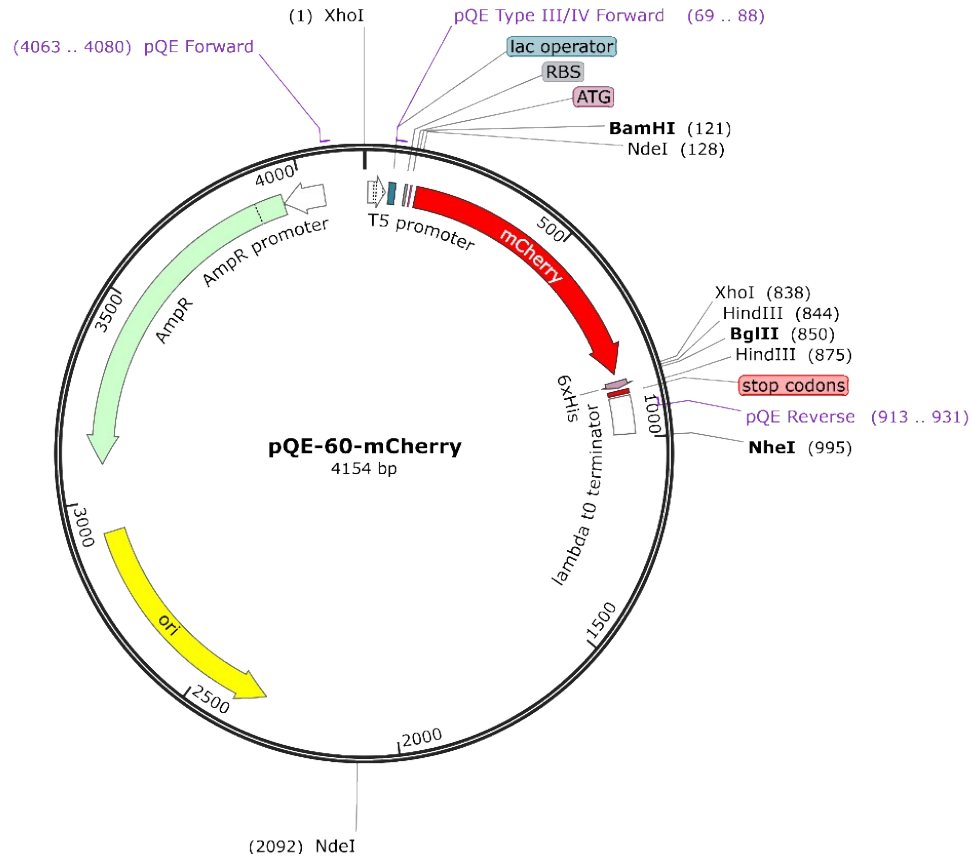


Figure S4. pQE-60 with mCherry plasmid map

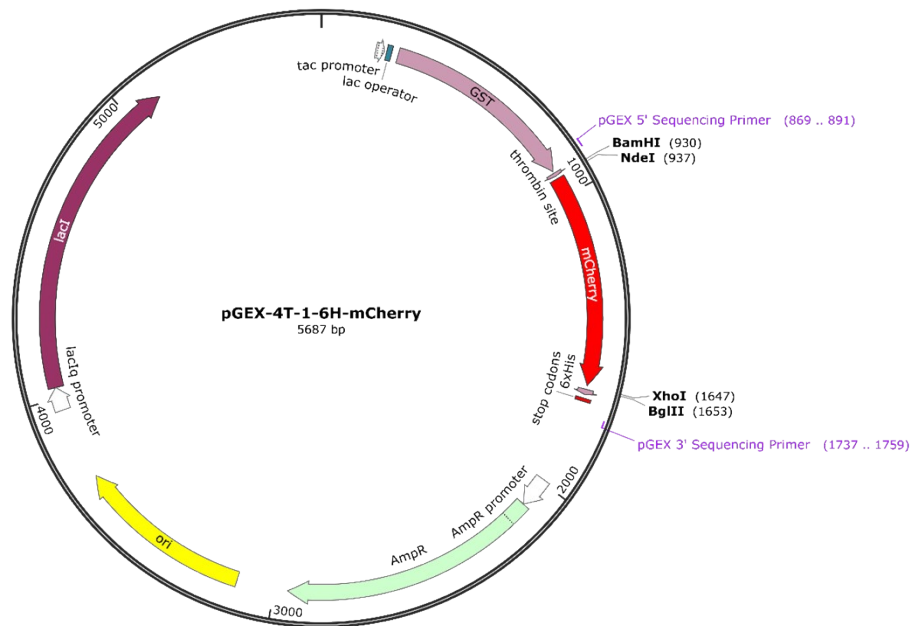


Figure S5. pGEX4T-(1H) with mCherry plasmid map

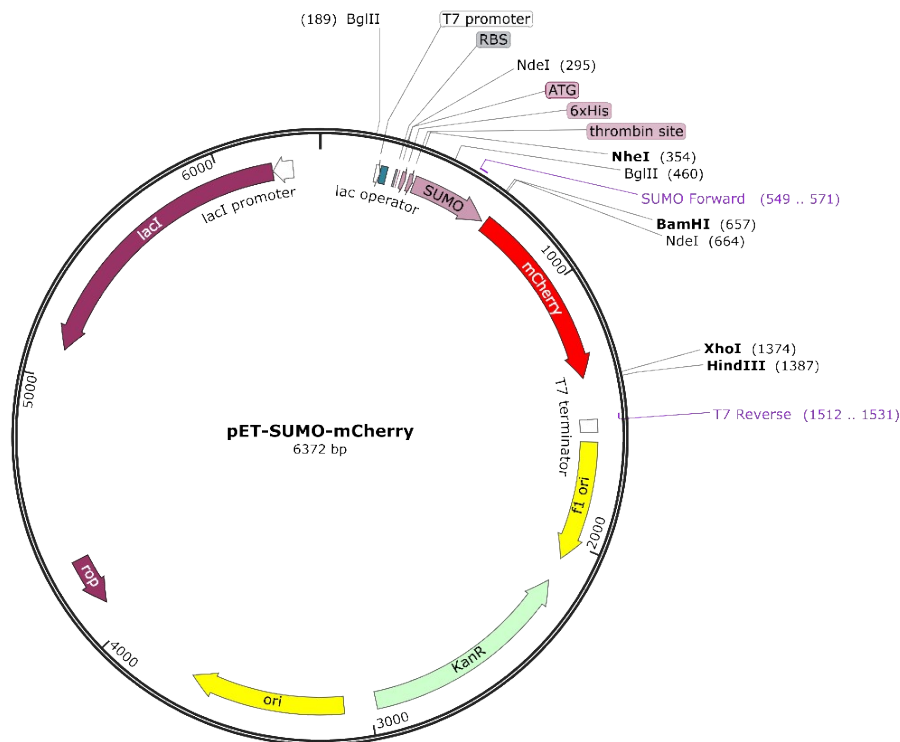


Figure S6. pET SUMO with mCherry plasmid map

E. OT-2 automated expression codes

Also available at: <https://github.com/olsenlabmit/Democratizing-Protein-Expression>

Protocol 1a: 20× antibiotic plate preparation

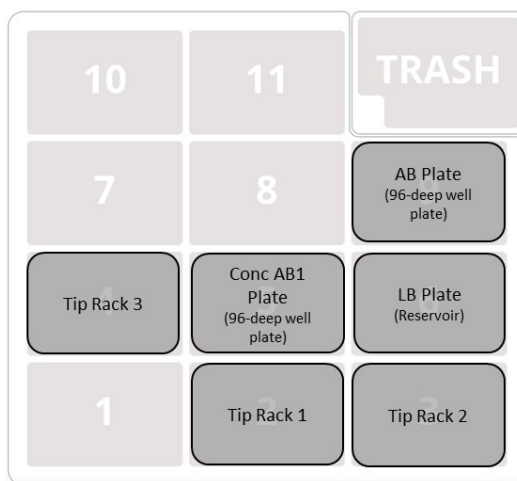


Figure S7. Protocol 1a plate set up.

####

#Create a 20x concentrated AB plate diluted in LB media. Plate configuration needs to match the final plate expression plate layout in terms of AB requirements

```

"""
#225 mL LB Media - Sterile
#25 mL 500xamp in LB Sterile - (15 mL in channel 1 - Conc_AB1)
#25 mL 500xChl in EtOH Sterile - (15 mL in channel 3 - Conc_AB1)
#25 mL 500xKan in LB Sterile - (15 mL in channel 5 - Conc_AB1)
#Tips: 2 tip boxes

from opentrons import protocol_api

metadata = {'apiLevel': '2.0',
            'protocolName': '20x AB plate prep',
            'author': 'Melody Morris',
            'description': 'Use a 500x AB stock solution to create a 20x AB
plate in the appropriate final layout'}

def run(protocol: protocol_api.ProtocolContext):
    pass

#Containers
    Conc_AB1 =
protocol.load_labware('usascientific_96_wellplate_2.4ml_deep', location='5',
label='Conc_AB1') #96 well plate
    LB_Plate =
protocol.load_labware('usascientific_96_wellplate_2.4ml_deep', location='6',
label='LB_Plate')
    AB_Plate =
protocol.load_labware('usascientific_96_wellplate_2.4ml_deep', location='9',
label='AB_Plate') #20xAB in LBPlate

#Tip racks and Pipettes
    Tip_Rack1=protocol.load_labware('opentrons_96_tiprack_300ul',location='
2', label='Tip_Rack1')
    Tip_Rack2=protocol.load_labware('opentrons_96_tiprack_300ul',location='
3', label='Tip_Rack2')
    Tip_Rack3=protocol.load_labware('opentrons_96_tiprack_300ul',location='
4', label='Tip_Rack3')

    p300m = protocol.load_instrument('p300_multi_gen2', 'left', [Tip_Rack1,
Tip_Rack2]) #multi channel pipette
    p300s = protocol.load_instrument('p300_single_gen2', 'right',
[Tip_Rack3]) #single channel pipette

#Constants
    AB_vol = 60 #uL volume of 500xConc AB
    LB_vol = 1440 #uL
    drop_targets = [3,9] #number of tip changes: 4
    control_targets = ['A12', 'B12', 'C12', 'D12', 'E12', 'G12']

# LB Transfer
    p300m.pick_up_tip()
    for i in range(0, 6):
        if i in drop_targets:
            p300m.drop_tip()
            p300m.pick_up_tip()
        p300m.transfer(LB_vol, LB_Plate.columns(0), AB_Plate.columns(i),
new_tip='never', blow_out = False, touch_tip = False)

```

```

p300m.drop_tip()

p300m.pick_up_tip()
for i in range(6, 12):
    if i in drop_targets:
        p300m.drop_tip()
        p300m.pick_up_tip()
        p300m.transfer(LB_vol, LB_Plate.columns(0), AB_Plate.columns(i),
new_tip='never', blow_out = False, touch_tip = False)
        p300m.drop_tip()

#500xAB transfer
# Transfer 500xAB to the AB_Plate - Amp and Kan
p300m.pick_up_tip()
for i in range(0,12):
    if i in drop_targets:
        p300m.drop_tip()
        p300m.pick_up_tip()
        p300m.transfer(AB_vol, Conc_AB1.columns(0), AB_Plate.columns(i),
new_tip='never', mix_after=(5,100), blow_out = False, touch_tip = False)
        p300m.drop_tip()

# Transfer 500xAB to the AB_Plate - Chlor
p300m.pick_up_tip()
p300m.transfer(AB_vol, Conc_AB1.columns(2), AB_Plate.columns(10),
new_tip='never', mix_after=(5,100), blow_out = False, touch_tip = False)
p300m.drop_tip()
# Transfer 500xAB to the AB_Plate - Kan
for i in control_targets:
    p300s.pick_up_tip()
    p300s.transfer(AB_vol, Conc_AB1.wells('F1'), AB_Plate.wells(i),
new_tip='never', mix_after=(5,100), blow_out = False, touch_tip = False)
    p300s.drop_tip()

#After use, seal the plate and store it in the 4 C fridge

```

1b1: Selection and growth plate preparation (1 selection plate, 3 growth plates)

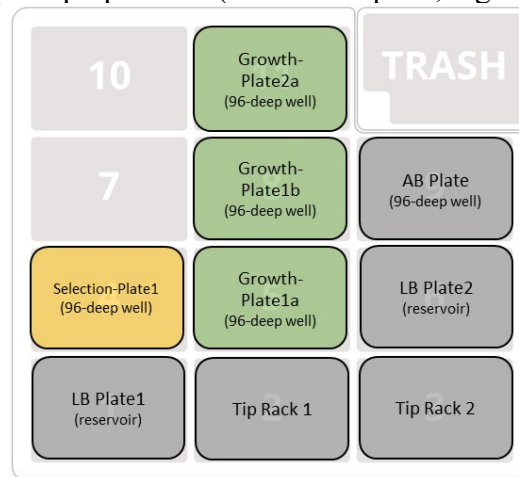


Figure S8. Protocol 1b1 plate set up.

""

```

Selection and Growth Plates Prep
"""
#500 mL LB Media - Sterile
#Filled AB Plate
#Tips: 2 tip boxes

from opentrons import protocol_api

metadata = {'apiLevel': '2.0',
            'protocolName': 'Selection and Growth Plate Prep',
            'author': 'Melody Morris',
            'description': 'Create 1 Selection and 3 Growth Plates from the
20x AB plate'}

def run(protocol: protocol_api.ProtocolContext):
    pass

#Containers
    LB_Plate1 =
protocol.load_labware('usascientific_96_wellplate_2.4ml_deep', location='1',
label='LB_Plate1') #LB Media Plate on a single channel container (No AB
Added)
    LB_Plate2 =
protocol.load_labware('usascientific_96_wellplate_2.4ml_deep', location='6',
label='LB_Plate2') #LB Media Plate on a single channel container (No AB
Added)
    AB_Plate=protocol.load_labware('usascientific_96_wellplate_2.4ml_deep',
location='9', label='AB_Plate') #20xAB Plate

Selection_Plate1=protocol.load_labware('usascientific_96_wellplate_2.4ml_deep
', location='4', label='Selection_Plate1')

Growth_Plate1=protocol.load_labware('usascientific_96_wellplate_2.4ml_deep',
location='5', label='Growth_Plate1')

Growth_Plate2=protocol.load_labware('usascientific_96_wellplate_2.4ml_deep',
location='8', label='Growth_Plate2')

Growth_Plate3=protocol.load_labware('usascientific_96_wellplate_2.4ml_deep',
location='11', label='Growth_Plate3')

#Tipracks and Pippetes

Tip_Rack1=protocol.load_labware('opentrons_96_tiprack_300ul',location='2',
label='Tip_Rack1') #Tip Rack 1

Tip_Rack2=protocol.load_labware('opentrons_96_tiprack_300ul',location='3',
label='Tip_Rack2') #Tip Rack 2

    p300m = protocol.load_instrument('p300_multi_gen2', 'left', [Tip_Rack1,
Tip_Rack2]) #multi channel pipette

#Constants
    LB_vol=[0, 570, 855, 855, 855] #uL
    AB_vol=[0, 30, 45, 45, 45] #uL 20xAB

```

```

#Plate List
Plate_List = [0, Selection_Plate1, Growth_Plate1, Growth_Plate2,
Growth_Plate3]
# i - column
# j - plate
#Master Plates and Expn Plates setup
#LB Transfer
    for i in range(0, 6):
        p300m.pick_up_tip()
        for j in range(1, 5):
            p300m.transfer(LB_vol[j], LB_Plate1.columns(0),
Plate_List[j].columns(i), new_tip='never', blow_out = False, touch_tip =
False)
        p300m.drop_tip()

    for i in range(6, 12):
        p300m.pick_up_tip()
        for j in range(1, 5):
            p300m.transfer(LB_vol[j], LB_Plate2.columns(0),
Plate_List[j].columns(i), new_tip='never', blow_out = False, touch_tip =
False)
        p300m.drop_tip()
#AB Transfer
    for i in range(0, 12):
        p300m.pick_up_tip()
        for j in range(1, 5):
            p300m.transfer(AB_vol[j], AB_Plate.columns(i),
Plate_List[j].columns(i), new_tip='never', blow_out = True, touch_tip =
False)
        p300m.drop_tip()
#Seal and store in 4 C fridge.  Incubate both the Selection and the Growth
Plates 37 C before use.

```

1b2: Selection and growth plate preparation (2 selection plate, 3 growth plates)

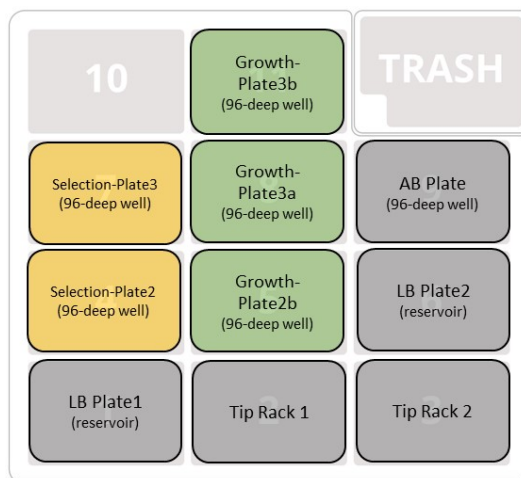


Figure S9. Protocol 1b2 plate set up.

```

"""
Selection and Growth Plates Prep
"""

```

```

#500 mL LB Media - Sterile
#Filled AB Plate
#Tips: 2 tip boxes

from opentrons import protocol_api

metadata = {'apiLevel': '2.0',
            'protocolName': 'Selection and Growth Plate Prep',
            'author': 'Melody Morris',
            'description': 'Create 2 Selection and 3 Growth Plates from the
20x AB plate'}

def run(protocol: protocol_api.ProtocolContext):
    pass

#Containers
    LB_Plate1 =
protocol.load_labware('usascientific_96_wellplate_2.4ml_deep', location='1',
label='LB_Plate1') #LB Media Plate on a single channel container (No AB
Added)
    LB_Plate2 =
protocol.load_labware('usascientific_96_wellplate_2.4ml_deep', location='6',
label='LB_Plate2') #LB Media Plate on a single channel container (No AB
Added)
    AB_Plate=protocol.load_labware('usascientific_96_wellplate_2.4ml_deep',
location='9', label='AB_Plate') #20xAB Plate

Selection_Plate1=protocol.load_labware('usascientific_96_wellplate_2.4ml_deep
', location='4', label='Selection_Plate1')

Selection_Plate2=protocol.load_labware('usascientific_96_wellplate_2.4ml_deep
', location='7', label='Selection_Plate2')

Growth_Plate1=protocol.load_labware('usascientific_96_wellplate_2.4ml_deep',
location='5', label='Growth_Plate1')

Growth_Plate2=protocol.load_labware('usascientific_96_wellplate_2.4ml_deep',
location='8', label='Growth_Plate2')

Growth_Plate3=protocol.load_labware('usascientific_96_wellplate_2.4ml_deep',
location='11', label='Growth_Plate3')

#Tipracks and Pipettes

Tip_Rack1=protocol.load_labware('opentrons_96_tiprack_300ul',location='2',
label='Tip_Rack1') #Tip Rack 1

Tip_Rack2=protocol.load_labware('opentrons_96_tiprack_300ul',location='3',
label='Tip_Rack2') #Tip Rack 2

    p300m = protocol.load_instrument('p300_multi_gen2', 'left', [Tip_Rack1,
Tip_Rack2]) #multi channel pipette

#Constants
    LB_vol=[0, 570, 570, 855, 855, 855] #uL

```

```

AB_vol=[0, 30, 30, 45, 45, 45] #uL 20xAB

#Plate List
Plate_List = [0, Selection_Plate1, Selection_Plate2, Growth_Plate1,
Growth_Plate2, Growth_Plate3]
# i - column
# j - plate
#Master Plates and Expn Plates setup
#LB Transfer
for i in range(0, 6):
    p300m.pick_up_tip()
    for j in range(1, 6):
        p300m.transfer(LB_vol[j], LB_Plate1.columns(0),
Plate_List[j].columns(i), new_tip='never', blow_out = False, touch_tip =
False)
    p300m.drop_tip()

for i in range(6, 12):
    p300m.pick_up_tip()
    for j in range(1, 6):
        p300m.transfer(LB_vol[j], LB_Plate2.columns(0),
Plate_List[j].columns(i), new_tip='never', blow_out = False, touch_tip =
False)
    p300m.drop_tip()

#AB Transfer
for i in range(0, 12):
    p300m.pick_up_tip()
    for j in range(1, 6):
        p300m.transfer(AB_vol[j], AB_Plate.columns(i),
Plate_List[j].columns(i), new_tip='never', blow_out = False, touch_tip =
False)
    p300m.drop_tip()
#Seal and store in 4 C fridge.  Incubate both the Selection and the Growth
Plates 37 C before use.

```

2: Transformation

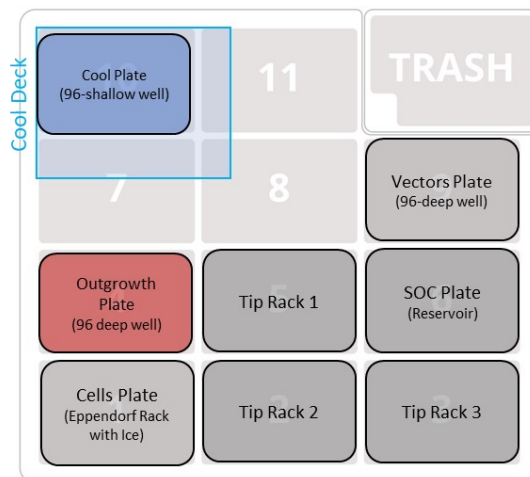


Figure S10. Protocol 2 plate set up.

```

"""
Transformation Protocol - Full Plate
"""

#Warm up the plate on top of the cold block for 30 min before starting
#Setup up the cell-vector containers 30 min before starting the experiment
#Cell volume: (45 uL x 8 ) x 1.2 = 432 uL
#Vector sln volume: (45 uL x 11) x 1.25 = 650 uL
#SOC: 50 mL
#Tips: 3 tipboxes

#Setting up vector deep well plate:
#A1 - pET15b-mCherry - 5 ng/uL in CaCl2 - 650 uL
#B1 - pET22b-mCherry - 5 ng/uL in CaCl2 - 650 uL
#C1 - pQE9-mCherry - 5 ng/uL in CaCl2 - 650 uL
#D1 - pQE60-mCherry - 5 ng/uL in CaCl2 - 650 uL
#E1 - pGEX-4T-H-mCherry - 5 ng/uL in CaCl2 - 650 uL
#F1 - pET-SUMO-mCherry - 5 ng/uL in CaCl2 - 650 uL
#G1 - pUC19 - 5 ng/uL in CaCl2 - 650 uL
#H1 - Water - 650 uL

#A2/B2 - pET15b-mCherry - 5 ng/uL in CaCl2 - 200 uL
#C2/D2 - pET15b-P4 - 5 ng/uL in CaCl2 - 200 uL

#Setting up Cell Eppendorf Plate:
#A1 - BL21          A2 - C41(DE)          A3 - T7 EXPRESS LYSY          A4 - WATER
#B1 - BL21(DE)      B2 - C43(DE)          B3 - T7 EXPRESS LYSY/Iq
#C1 - BL21*(DE)     C2- NiCO(DE)          C3 - ROSETTA 2
#D1 - Tuner(DE)     D2 - T7 Express       D3 - SG

from opentrons import protocol_api

metadata = {'apiLevel': '2.0',
            'protocolName': 'Transformation',
            'author': 'Melody Morris',
            'description': 'Transforming the DNA and competent cells'}

def run(protocol: protocol_api.ProtocolContext):
    pass

#Containers

    Outgrowth_Plate =
protocol.load_labware('usascientific_96_wellplate_2.4ml_deep',location='4',la
bel='Outgrowth_Plate') #Set to 45 C
    Cool_Plate =
protocol.load_labware('corning96incoldplate_96_wellplate_360ul',location='10'
,label='Cool_Plate') #Set to 4 C
    Cells_Plate =
protocol.load_labware('opentrons_24_tuberack_eppendorf_2ml_safelock_snapcap',
location='1', label='Cells_Plate')
    Vectors_Plate =
protocol.load_labware('usascientific_96_wellplate_2.4ml_deep', location='9',
label='Vectors_Plate')
    SOC_Plate =
protocol.load_labware('usascientific_96_wellplate_2.4ml_deep', location='6',
label='SOC_Plate')

```

```

#Tip racks and Pipettes

Tip_Rack1=protocol.load_labware('opentrons_96_tiprack_300ul',location='5',
label='Tip_Rack1')

Tip_Rack2=protocol.load_labware('opentrons_96_tiprack_300ul',location='2',
label='Tip_Rack2')

Tip_Rack3=protocol.load_labware('opentrons_96_tiprack_300ul',location='3',
label='Tip_Rack3')

    p300m = protocol.load_instrument('p300_multi_gen2', 'left', [Tip_Rack1,
Tip_Rack2, Tip_Rack3]) #multi channel pipette
    p300s = protocol.load_instrument('p300_single_gen2', 'right',
[Tip_Rack3]) #single channel pipette

#Constants
    plasmid_vol = 45 # volume of plasmid added to cool plate
    cell_vol = 45 # volume of cells added to cool plate
    soc_vol = 200 # volume of SOC added to heat plate
    cell_plasmid_vol = 50 # volume of cell/plasmid mixture transferred from
cool plate to heat plate

#Cells
    p=[]
    p.append(Cells_Plate.wells('A1')) #Cell 1 - BL21
    p.append(Cells_Plate.wells('B1')) #Cell 2 - BL21 (DE)
    p.append(Cells_Plate.wells('C1')) #Cell 3 - BL21-STAR (DE)
    p.append(Cells_Plate.wells('D1')) #Cell 4 - TUNER (DE)
    p.append(Cells_Plate.wells('A2')) #Cell 5 - C41 (DE)
    p.append(Cells_Plate.wells('B2')) #Cell 6 - C43 (DE)
    p.append(Cells_Plate.wells('C2')) #Cell 7 - NICO21 (DE)
    p.append(Cells_Plate.wells('D2')) #Cell 8 - T7 EXPRESS
    p.append(Cells_Plate.wells('A3')) #Cell 9 - T7 EXPRESS LYSY
    p.append(Cells_Plate.wells('B3')) #Cell 10 - T7 EXPRESS LYSY/Iq
    p.append(Cells_Plate.wells('C3')) #Cell 11 - ROSETTA 2
    p.append(Cells_Plate.wells('D3')) #Cell 12 - SG13
    p.append(Cells_Plate.wells('A4')) #Cell 13 - WATER

# Transfer competent cells to cold plate with single channel for the 11 first
cols
    p300s.pick_up_tip()
    p300s.distribute(cell_vol, p[0], Cool_Plate.columns(0), disposal_vol=5,
new_tip='never', blow_out= True, touch_tip = False)
    p300s.drop_tip()

    p300s.pick_up_tip()
    p300s.distribute(cell_vol, p[1], Cool_Plate.columns(1), disposal_vol=5,
new_tip='never', blow_out= True, touch_tip = False)
    p300s.drop_tip()

    p300s.pick_up_tip()
    p300s.distribute(cell_vol, p[2], Cool_Plate.columns(2), disposal_vol=5,
new_tip='never', blow_out= True, touch_tip = False)
    p300s.drop_tip()

```

```

    p300s.pick_up_tip()
    p300s.distribute(cell_vol, p[3], Cool_Plate.columns(3), disposal_vol=5,
new_tip='never', blow_out= True, touch_tip = False)
    p300s.drop_tip()

    p300s.pick_up_tip()
    p300s.distribute(cell_vol, p[4], Cool_Plate.columns(4), disposal_vol=5,
new_tip='never', blow_out= True, touch_tip = False)
    p300s.drop_tip()

    p300s.pick_up_tip()
    p300s.distribute(cell_vol, p[5], Cool_Plate.columns(5), disposal_vol=5,
new_tip='never', blow_out= True, touch_tip = False)
    p300s.drop_tip()

    p300s.pick_up_tip()
    p300s.distribute(cell_vol, p[6], Cool_Plate.columns(6), disposal_vol=5,
new_tip='never', blow_out= True, touch_tip = False)
    p300s.drop_tip()

    p300s.pick_up_tip()
    p300s.distribute(cell_vol, p[7], Cool_Plate.columns(7), disposal_vol=5,
new_tip='never', blow_out= True, touch_tip = False)
    p300s.drop_tip()

    p300s.pick_up_tip()
    p300s.distribute(cell_vol, p[8], Cool_Plate.columns(8), disposal_vol=5,
new_tip='never', blow_out= True, touch_tip = False)
    p300s.drop_tip()

    p300s.pick_up_tip()
    p300s.distribute(cell_vol, p[9], Cool_Plate.columns(9), disposal_vol=5,
new_tip='never', blow_out= True, touch_tip = False)
    p300s.drop_tip()

    p300s.pick_up_tip()
    p300s.distribute(cell_vol, p[10], Cool_Plate.columns(10), disposal_vol=5,
new_tip='never', blow_out= True, touch_tip = False)
    p300s.drop_tip()

##### HARD CODING THE CONTROLS #####
#     positive_targets = ['A12', 'B12', 'C12', 'D12'] #SG13 cells addition
#     negative_targets = ['E12', 'F12', 'G12', 'H12'] #Water

# Transfer competent cells to cold plate with single channel - SG13 -
POSITIVE CONTROL
    p300s.pick_up_tip()
    p300s.distribute(cell_vol, p[11],
[Cool_Plate.wells_by_name()[positive_targets] for positive_targets in ['A12',
'B12', 'C12', 'D12']], disposal_vol=5, new_tip='never', blow_out = True,
touch_tip = False)
    p300s.drop_tip()

# Transfer competent cells to cold plate with single channel - water -
NEGATIVE CONTROL
    p300s.pick_up_tip()

```

```

    p300s.distribute(cell_vol, p[12],
[Cool_Plate.wells_by_name()[negative_targets] for negative_targets in ['E12',
'F12', 'G12', 'H12']], disposal_vol=5, new_tip='never', blow_out = True,
touch_tip = False)
    p300s.drop_tip()

# Transfer DNA to cold plate with single channel and mix FOR THE 11 COLS
    for i in range(0,11):
        p300m.pick_up_tip()
        p300m.transfer(plasmid_vol, Vectors_Plate.columns(0),
Cool_Plate.columns(i), new_tip='never', blow_out = True, touch_tip = False)
        p300m.drop_tip()

# Transfer DNA to cold plate with single channel and mix POSITIVE AND
NEGATIVE CONTROL
    p300s.pick_up_tip()
    p300s.transfer(plasmid_vol, Vectors_Plate.wells('A2'),
Cool_Plate.wells('A12'), new_tip='never', blow_out = True, touch_tip = False)
    p300s.drop_tip()

    p300s.pick_up_tip()
    p300s.transfer(plasmid_vol, Vectors_Plate.wells('A2'),
Cool_Plate.wells('B12'), new_tip='never', blow_out = True, touch_tip = False)
    p300s.drop_tip()

    p300s.pick_up_tip()
    p300s.transfer(plasmid_vol, Vectors_Plate.wells('B2'),
Cool_Plate.wells('C12'), new_tip='never', blow_out = True, touch_tip = False)
    p300s.drop_tip()

    p300s.pick_up_tip()
    p300s.transfer(plasmid_vol, Vectors_Plate.wells('B2'),
Cool_Plate.wells('D12'), new_tip='never', blow_out = True, touch_tip = False)
    p300s.drop_tip()

    p300s.pick_up_tip()
    p300s.transfer(plasmid_vol, Vectors_Plate.wells('C2'),
Cool_Plate.wells('E12'), new_tip='never', blow_out = True, touch_tip = False)
    p300s.drop_tip()

    p300s.pick_up_tip()
    p300s.transfer(plasmid_vol, Vectors_Plate.wells('C2'),
Cool_Plate.wells('F12'), new_tip='never', blow_out = True, touch_tip = False)
    p300s.drop_tip()

    p300s.pick_up_tip()
    p300s.transfer(plasmid_vol, Vectors_Plate.wells('C2'),
Cool_Plate.wells('G12'), new_tip='never', blow_out = True, touch_tip = False)
    p300s.drop_tip()

    p300s.pick_up_tip()
    p300s.transfer(plasmid_vol, Vectors_Plate.wells('C2'),
Cool_Plate.wells('H12'), new_tip='never', blow_out = True, touch_tip = False)
    p300s.drop_tip()

# Transfer SOC to the Outgrowth Plate - no Shaking
    p300m.pick_up_tip(Tip_Rack3['A9'])

```

```

    for i in range(0,6):
        p300m.transfer(soc_vol, SOC_Plate.wells('A1'),
Outgrowth_Plate.columns(i), new_tip='never', blow_out = False, touch_tip =
False)
        p300m.drop_tip()

        p300m.pick_up_tip(Tip_Rack3['A10'])
        for i in range(6,12):
            p300m.transfer(soc_vol, SOC_Plate.wells('A1'),
Outgrowth_Plate.columns(i), new_tip='never', blow_out = False, touch_tip =
False)
            p300m.drop_tip()

        protocol.delay(seconds=1200)

# Transfer competent cells + DNA to outgrowth plate with multichannel
    for i in range(0,12):
        p300m.pick_up_tip()
        p300m.transfer(cell_plasmid_vol, Cool_Plate.columns(i),
Outgrowth_Plate.columns(i), new_tip='never', blow_out = True, touch_tip =
False)
        p300m.drop_tip()

#Incubate the Outgrowth Plate in the Plate shaker at 37 C and 300 rpm for 1 h

```

3: Seeding the 3 selection plates

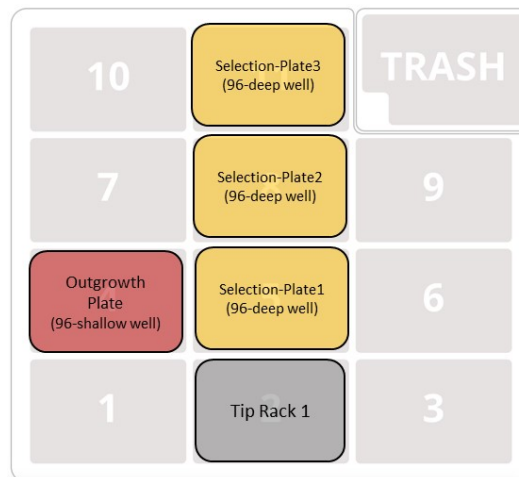


Figure S11. Protocol 3 plate set up.

```

"""
Seeding the Selection Plates
"""
#Warm up the incubator to 37 C
#tips: 1 tip box
#Outgrowth plate
from opentrons import protocol_api

metadata = {'apiLevel': '2.0',
            'protocolName': 'Seeding Selection Plates',
            'author': 'Melody Morris',

```

```

        'description': 'Seeding 3 selection plates from the
transformation plate'}

def run(protocol: protocol_api.ProtocolContext):
    pass

#Containers

Outgrowth_Plate=protocol.load_labware('usascientific_96_wellplate_2.4ml_deep'
, location='4', label='Outgrowth_Plate')

Selection_Plate1=protocol.load_labware('usascientific_96_wellplate_2.4ml_deep'
, location='5', label='Selection_Plate1')

Selection_Plate2=protocol.load_labware('usascientific_96_wellplate_2.4ml_deep'
, location='8', label='Selection_Plate2')

Selection_Plate3=protocol.load_labware('usascientific_96_wellplate_2.4ml_deep'
, location='11', label='Selection_Plate3')

#Tipracks and Pipettes

Tip_Rack1=protocol.load_labware('opentrons_96_tiprack_300ul',location='2',
label='Tip_Rack1')

    p300m = protocol.load_instrument('p300_multi_gen2', 'left',[Tip_Rack1])
#multi-channel pipette

#Constants
    Inoc_vol=60

    Plate_List = [0, Selection_Plate1, Selection_Plate2, Selection_Plate3]

#j - plates
#i - columns
#Cell Seeding
    for i in range(0, 12):
        p300m.pick_up_tip()
        for j in range(1, 4):
            p300m.transfer(Inoc_vol, Outgrowth_Plate.columns(i),
Plate_List[j].columns(i), new_tip='never', blow_out = False, touch_tip =
False)
            p300m.drop_tip()

#Seal and Incubate all the Selection Plates at 37C and 300 rpm for 20 h

```

4a: Seeding growth plates (repeat 3 times total)

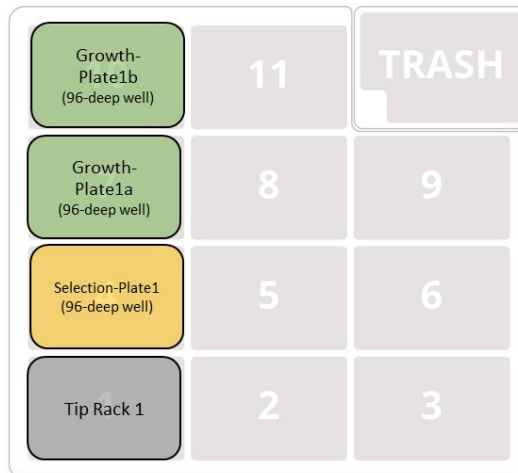


Figure S12. Protocol 4a plate set up.

```

"""
Master Plate Seeding
"""
#Warm up the incubator at 37 C for 30 min prior starting the experiment
#tips: 1 full rack per Selection Plate
#Transformed cells: 30 uL/well

from opentrons import protocol_api

metadata = {'apiLevel': '2.0',
            'protocolName': 'Seeding Growth Plates',
            'author': 'Melody Morris',
            'description': 'Seeding 2 growth plates from the selection
plates'}

def run(protocol: protocol_api.ProtocolContext):
    pass

#Containers

Selection_Plate1=protocol.load_labware('usascientific_96_wellplate_2.4ml_deep',
location='4', label='Selection_Plate1')

Growth_Plate1a=protocol.load_labware('usascientific_96_wellplate_2.4ml_deep',
location='7', label='Growth_Plate1a')

Growth_Plate1b=protocol.load_labware('usascientific_96_wellplate_2.4ml_deep',
location='10', label='Growth_Plate1b')

#Tipracks and Pipettes

Tip_Rack1=protocol.load_labware('opentrons_96_tiprack_300ul',location='1',
label='Tip_Rack1') #Tip Rack 1

    p300m = protocol.load_instrument('p300_multi_gen2', 'left',[Tip_Rack1])
#multi channel pipette

#Constants

```

```

Inoc_vol=30

Plate_List = [0, Selection_Plate1, Growth_Plate1a, Growth_Plate1b]

#j - plates
#i - columns
#Cell Seeding from the Selection Plate 1
for i in range(0,12):
    p300m.pick_up_tip()
    for j in range(1, 3):
        p300m.transfer(Inoc_vol, Plate_List[1].columns(i),
Plate_List[j+1].columns(i), new_tip='never', blow_out = False, touch_tip =
False)
        p300m.drop_tip()

#Seal and Incubate all the Growth Plates at 37C and 300 rpm for 2.5 h. Keep
the Selection Plates for OD Measurements.

```

4b: Selection Plate Transfer for OD Measurements

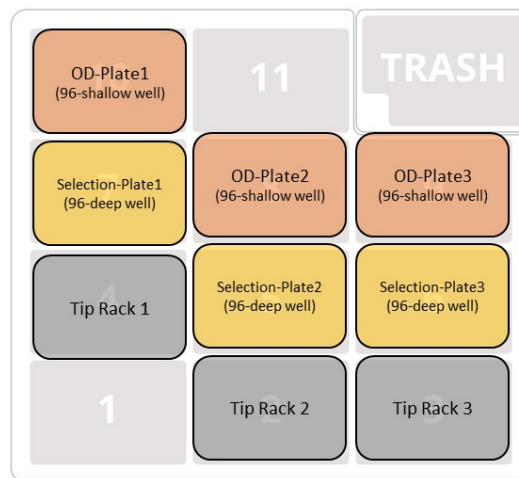


Figure S13. Protocol 4b plate set up.

```

"""
Transfer into Shallow Well Plate for OD Measurements- Selection Plates
"""
#Tips required: 3 tip racks
#3 96-Shallow well plates
from opentrons import protocol_api

metadata = {'apiLevel': '2.0',
            'protocolName': 'Selection Plate OD transfer',
            'author': 'Melody Morris',
            'description': 'Move 200 uL to 96 well plates for OD600
measurements'}

def run(protocol: protocol_api.ProtocolContext):
    pass

#Containers

```

```

    OD_Plate1=protocol.load_labware('corning_96_wellplate_360ul_flat',
location='10', label='OD_Plate1')
    OD_Plate2=protocol.load_labware('corning_96_wellplate_360ul_flat',
location='8', label='OD_Plate2')
    OD_Plate3=protocol.load_labware('corning_96_wellplate_360ul_flat',
location='9', label='OD_Plate3')

Selection_Plate1=protocol.load_labware('usascientific_96_wellplate_2.4ml_deep
', location='7', label='Selection_Plate1')

Selection_Plate2=protocol.load_labware('usascientific_96_wellplate_2.4ml_deep
', location='5', label='Selection_Plate2')

Selection_Plate3=protocol.load_labware('usascientific_96_wellplate_2.4ml_deep
', location='6', label='Selection_Plate3')

#Tipracks and Pipettes

Tip_Rack1=protocol.load_labware('opentrons_96_tiprack_300ul',location='4',
label='Tip_Rack1')

Tip_Rack2=protocol.load_labware('opentrons_96_tiprack_300ul',location='2',
label='Tip_Rack2')

Tip_Rack3=protocol.load_labware('opentrons_96_tiprack_300ul',location='3',
label='Tip_Rack3')

    p300m = protocol.load_instrument('p300_multi_gen2', 'left', [Tip_Rack1,
Tip_Rack2, Tip_Rack3]) #multi channel pipette

#constants
    OD_vol=200 #uL

#Plate List
    Plate_List = [0, Selection_Plate1, Selection_Plate2, Selection_Plate3,
OD_Plate1, OD_Plate2, OD_Plate3]

#Transfer Culture Media from the Expression Plate to the OD_Plates
    for j in range(1, 4):
        for i in range(0,12):
            p300m.pick_up_tip()
            p300m.transfer(OD_vol, Plate_List[j].columns(i),
Plate_List[j+3].columns(i), new_tip='never', blow_out = False, touch_tip =
False)
            p300m.drop_tip()

# Use plate reader or robot to obtain OD600

```

5a: Growth Plate Transfer for OD Measurements (repeat 2x total)

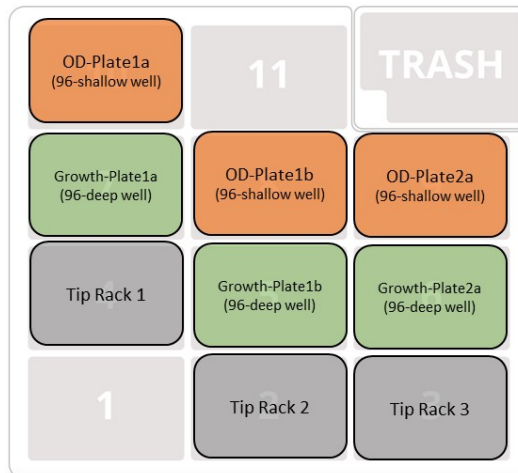


Figure S14. Protocol 5a plate set up.

```

"""
Transfer into Shallow Well Plate for OD Measurements- Growth Plates
"""
#Tips required: 3 tip racks
#3 96-Shallow well plates
from opentrons import protocol_api

metadata = {'apiLevel': '2.0',
            'protocolName': 'Growth Plate OD transfer',
            'author': 'Melody Morris',
            'description': 'Move 200 uL to 96 well plates for OD600
measurements'}

def run(protocol: protocol_api.ProtocolContext):
    pass

#Containers
    OD_Plate1=protocol.load_labware('corning_96_wellplate_360ul_flat',
location='10', label='OD_Plate1')
    OD_Plate2=protocol.load_labware('corning_96_wellplate_360ul_flat',
location='8', label='OD_Plate2')
    OD_Plate3=protocol.load_labware('corning_96_wellplate_360ul_flat',
location='9', label='OD_Plate3')

Selection_Plate1=protocol.load_labware('usascientific_96_wellplate_2.4ml_deep
', location='7', label='Selection_Plate1')

Selection_Plate2=protocol.load_labware('usascientific_96_wellplate_2.4ml_deep
', location='5', label='Selection_Plate2')

Selection_Plate3=protocol.load_labware('usascientific_96_wellplate_2.4ml_deep
', location='6', label='Selection_Plate3')

#Tipracks and Pipettes

Tip_Rack1=protocol.load_labware('opentrons_96_tiprack_300ul',location='4',
label='Tip_Rack1')

```

```

Tip_Rack2=protocol.load_labware('opentrons_96_tiprack_300ul',location='2',
label='Tip_Rack2')

Tip_Rack3=protocol.load_labware('opentrons_96_tiprack_300ul',location='3',
label='Tip_Rack3')

p300m = protocol.load_instrument('p300_multi_gen2', 'left', [Tip_Rack1,
Tip_Rack2, Tip_Rack3]) #multi channel pipette

#constants
OD_vol=200 #uL

#Plate List
Plate_List = [0, Selection_Plate1, Selection_Plate2, Selection_Plate3,
OD_Plate1, OD_Plate2, OD_Plate3]

#Transfer Culture Media from the Expression Plate to the OD_Plates
for j in range(1, 4):
    for i in range(0,12):
        p300m.pick_up_tip()
        p300m.transfer(OD_vol, Plate_List[j].columns(i),
Plate_List[j+3].columns(i), new_tip='never', blow_out = False, touch_tip =
False)
        p300m.drop_tip()

# Use plate reader or robot to obtain OD600

```

5b: Expression Plate induction (repeat 2x total)

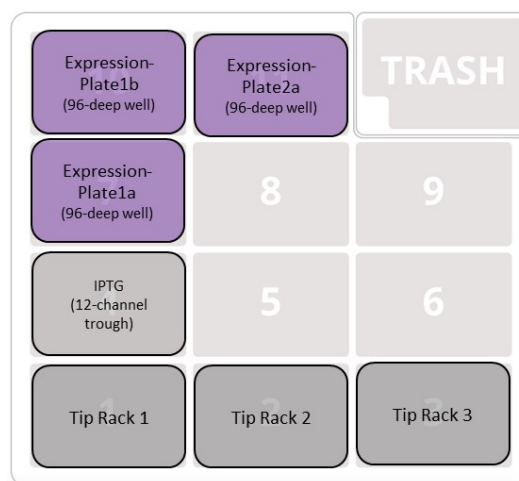


Figure S15. Protocol 5b plate set up.

```

"""
Expression Plate Induction
"""
#tips: 3 full racks
#IPTG vol: 40 mL IPTG 20 mM in LB
from opentrons import protocol_api

metadata = {'apiLevel': '2.0',
            'protocolName': 'Expression Plate Induction',

```

```

        'author': 'Melody Morris',
        'description': 'Induce with 1 uM IPTG in each well'}

def run(protocol: protocol_api.ProtocolContext):
    pass

#Containers
    IPTG_Plate =
protocol.load_labware('usascientific_96_wellplate_2.4ml_deep', location='4',
label='IPTG_Plate') #IPTG in LB media to diluted to reach the final
concentration of 1.0 uM in each well

Expression_Plate1=protocol.load_labware('usascientific_96_wellplate_2.4ml_dee
p', location='7', label='Expression_Plate1')

Expression_Plate2=protocol.load_labware('usascientific_96_wellplate_2.4ml_dee
p', location='10', label='Expression_Plate2')

Expression_Plate3=protocol.load_labware('usascientific_96_wellplate_2.4ml_dee
p', location='11', label='Expression_Plate3')

#Tipracks and Pipettes

Tip_Rack1=protocol.load_labware('opentrons_96_tiprack_300ul',location='1',
label='Tip_Rack1') #Tip Rack 1

Tip_Rack2=protocol.load_labware('opentrons_96_tiprack_300ul',location='2',
label='Tip_Rack2')

Tip_Rack3=protocol.load_labware('opentrons_96_tiprack_300ul',location='3',
label='Tip_Rack3')

    p300m = protocol.load_instrument('p300_multi_gen2', 'left', [Tip_Rack1,
Tip_Rack2, Tip_Rack3]) #multi channel pipette

#Constants
    IPTG_vol=47.36 #IPTG 20 mM in LB

#Plate List
    Plate_List = [IPTG_Plate, Expression_Plate1, Expression_Plate2,
Expression_Plate3]

#j - plates
#i - columns
#IPTG Addition
    for j in range(1, 4):
        for i in range(0,12):
            p300m.pick_up_tip()
            p300m.transfer(IPTG_vol, Plate_List[0].columns(0),
Plate_List[j].columns(i), new_tip='never', blow_out = False, touch_tip =
False)
            p300m.drop_tip()

#Seal and Incubate all the Expression Plates at 25 C and 300 rpm for 20 h

```

5c: Expression Plate Transfer for OD Measurements (repeat 2x total)

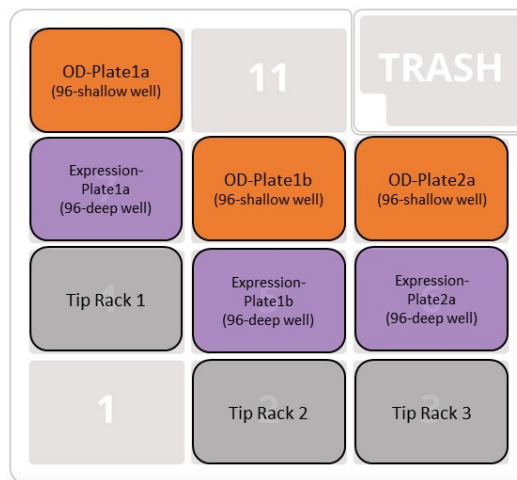


Figure S16. Protocol 5c plate set up.

```

"""
Transfer into Shallow Well Plate for OD Measurements- Expression Plates
"""
#Tips required: 3 tip racks
#3 96-Shallow well plates
from opentrons import protocol_api

metadata = {'apiLevel': '2.0',
            'protocolName': 'Selection Plate OD transfer',
            'author': 'Melody Morris',
            'description': 'Move 200 uL to 96 well plates for OD600
measurements'}

def run(protocol: protocol_api.ProtocolContext):
    pass

#Containers
    OD_Plate1=protocol.load_labware('corning_96_wellplate_360ul_flat',
location='10', label='OD_Plate1')
    OD_Plate2=protocol.load_labware('corning_96_wellplate_360ul_flat',
location='8', label='OD_Plate2')
    OD_Plate3=protocol.load_labware('corning_96_wellplate_360ul_flat',
location='9', label='OD_Plate3')

Selection_Plate1=protocol.load_labware('usascientific_96_wellplate_2.4ml_deep
', location='7', label='Selection_Plate1')

Selection_Plate2=protocol.load_labware('usascientific_96_wellplate_2.4ml_deep
', location='5', label='Selection_Plate2')

Selection_Plate3=protocol.load_labware('usascientific_96_wellplate_2.4ml_deep
', location='6', label='Selection_Plate3')

#Tipracks and Pipettes

```

```

Tip_Rack1=protocol.load_labware('opentrons_96_tiprack_300ul',location='4',
label='Tip_Rack1')

Tip_Rack2=protocol.load_labware('opentrons_96_tiprack_300ul',location='2',
label='Tip_Rack2')

Tip_Rack3=protocol.load_labware('opentrons_96_tiprack_300ul',location='3',
label='Tip_Rack3')

    p300m = protocol.load_instrument('p300_multi_gen2', 'left', [Tip_Rack1,
Tip_Rack2, Tip_Rack3]) #multi channel pipette

#constants
    OD_vol=200 #uL

#Plate List
    Plate_List = [0, Selection_Plate1, Selection_Plate2, Selection_Plate3,
OD_Plate1, OD_Plate2, OD_Plate3]

#Transfer Culture Media from the Expression Plate to the OD_Plates
    for j in range(1, 4):
        for i in range(0,12):
            p300m.pick_up_tip()
            p300m.transfer(OD_vol, Plate_List[j].columns(i),
Plate_List[j+3].columns(i), new_tip='never', blow_out = False, touch_tip =
False)
            p300m.drop_tip()

# Use plate reader or robot to obtain OD600

```

6a: Bradford analysis

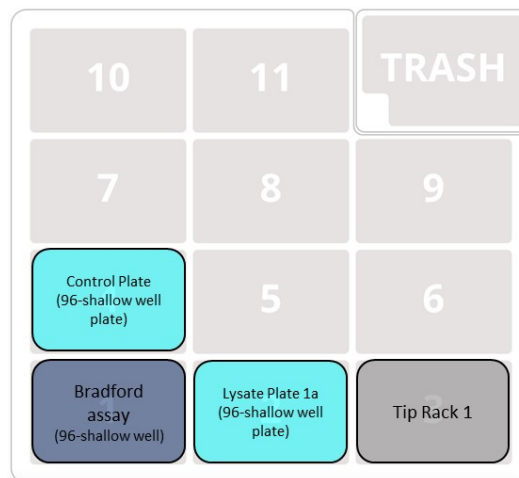


Figure S17. Protocol 6a plate set up.

```

"""
Preparing the Bradford Plate
"""
#Tips: 1 tip rack
#200 uL of Bradford Reagent in each shallow well

```

```

from opentrons import protocol_api

metadata = {'apiLevel': '2.0',
            'protocolName': 'Bradford protocol',
            'author': 'Melody Morris',
            'description': 'Add protein to Bradford assay'}

def run(protocol: protocol_api.ProtocolContext):
    pass

#Containers

    Lysate_Plate=protocol.load_labware('corning_96_wellplate_360ul_flat',
location='2', label='Lysate_Plate')

Bradford_Assay=protocol.load_labware('corning_96_wellplate_360ul_flat',locati
on='1', label='Bradford_Assay')

Controls_Plate=protocol.load_labware('corning_96_wellplate_360ul_flat',locati
on='4', label='Control_Plate')

#Tip Racks and Pipettes

Tip_Rack1=protocol.load_labware('opentrons_96_tiprack_300ul',location='3',
label='Tip_Rack1') #Tip Rack

    p300m = protocol.load_instrument('p300_multi_gen2', 'left', [Tip_Rack1])
#multi channel pipette
    p300s = protocol.load_instrument('p300_single_gen2', 'right',
[Tip_Rack1]) #single channel pipette
#constants
    Assay_vol = 20 #mg/mL
#positive_controls = [Lysate_Plate.well('A12'), Lysate_Plate.well('B12'),
Lysate_Plate.well('C12'), Lysate_Plate.well('D12')]
#calibration_curve = [Controls_Plate.well('A1'), Controls_Plate.well('B1'),
Controls_Plate.well('C1'), Controls_Plate.well('D1')]
#Supplies
#Brad_Reagent = Supply_Plate.wells('A3')

#Transfer protein assay to the Bradford Assay Plate
#start=time.time()
    for j in range(11):
        p300m.transfer(Assay_vol, Lysate_Plate.columns(j),
Bradford_Assay.columns(j), new_tip='always', mix_before=(5, 50),
mix_after=(5,50), blow_out = True, touch_tip = False)

        p300s.transfer(Assay_vol, Lysate_Plate.well('A12').bottom(2),
Bradford_Assay.well('A12').bottom(2), new_tip='always', mix_before=(5, 50),
mix_after=(5,50), blow_out = True, touch_tip = False)
        p300s.transfer(Assay_vol, Lysate_Plate.well('B12').bottom(2),
Bradford_Assay.well('B12').bottom(2), new_tip='always', mix_before=(5, 50),
mix_after=(5,50), blow_out = True, touch_tip = False)
        p300s.transfer(Assay_vol, Lysate_Plate.well('C12').bottom(2),
Bradford_Assay.well('C12').bottom(2), new_tip='always', mix_before=(5, 50),
mix_after=(5,50), blow_out = True, touch_tip = False)

```

```

    p300s.transfer(Assay_vol, Lysate_Plate.well('D12').bottom(2),
Bradford_Assay.well('D12').bottom(2), new_tip='always', mix_before=(5, 50),
mix_after=(5,50), blow_out = True, touch_tip = False)
    p300s.transfer(Assay_vol, Controls_Plate.well('A1').bottom(2),
Bradford_Assay.well('E12').bottom(2), new_tip='always', mix_before=(5, 50),
mix_after=(5,50), blow_out = True, touch_tip = False)
    p300s.transfer(Assay_vol, Controls_Plate.well('B1').bottom(2),
Bradford_Assay.well('F12').bottom(2), new_tip='always', mix_before=(5, 50),
mix_after=(5,50), blow_out = True, touch_tip = False)
    p300s.transfer(Assay_vol, Controls_Plate.well('C1').bottom(2),
Bradford_Assay.well('G12').bottom(2), new_tip='always', mix_before=(5, 50),
mix_after=(5,50), blow_out = True, touch_tip = False)
    p300s.transfer(Assay_vol, Controls_Plate.well('D1').bottom(2),
Bradford_Assay.well('H12').bottom(2), new_tip='always', mix_before=(5, 50),
mix_after=(5,50), blow_out = True, touch_tip = False)

    protocol.delay(seconds=180)

##### When the robot moves, move the Bradford Assay plate for OT1 or plate
reader for imaging
##### Measure the plate at the robot and in the plate reader (595 nm)

```

6b1: Transfer from shallow well plate to PCR plate for dot blot (repeat 2x total)

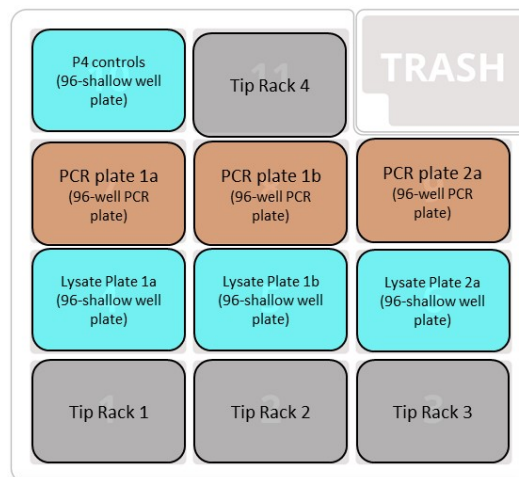


Figure S18. Protocol 6b1 plate set up.

```

"""
Preparing the Dot Blot PCR plate
"""
#Tips: 3 tip racks
from opentrons import protocol_api

metadata = {'apiLevel': '2.0',
            'protocolName': 'Dot blot denaturing and dotting',
            'author': 'Melody Morris',
            'description': 'Transfer lysate to PCR plate'}

def run(protocol: protocol_api.ProtocolContext):
    pass

```

```

#Containers

    Lysate_Plate1=protocol.load_labware('corning_96_wellplate_360ul_flat',
location='4', label='Lysate_Plate1')
    Lysate_Plate2=protocol.load_labware('corning_96_wellplate_360ul_flat',
location='5', label='Lysate_Plate2')
    Lysate_Plate3=protocol.load_labware('corning_96_wellplate_360ul_flat',
location='6', label='Lysate_Plate3')

PCR_Plate1=protocol.load_labware('pcrplateindeepwell_96_wellplate_125ul',loca
tion='7', label='PCR_Plate1')

PCR_Plate2=protocol.load_labware('pcrplateindeepwell_96_wellplate_125ul',loca
tion='8', label='PCR_Plate2')

PCR_Plate3=protocol.load_labware('pcrplateindeepwell_96_wellplate_125ul',loca
tion='9', label='PCR_Plate3')

Controls_Plate=protocol.load_labware('corning_96_wellplate_360ul_flat',locati
on='10', label='Control_Plate')

#Tip Racks and Pippetes
    Tip_Rack1=protocol.load_labware('opentrons_96_tiprack_20ul',location='1',
label='Tip_Rack1') #Tip Rack
    Tip_Rack2=protocol.load_labware('opentrons_96_tiprack_20ul',location='2',
label='Tip_Rack2') #Tip Rack
    Tip_Rack3=protocol.load_labware('opentrons_96_tiprack_20ul',location='3',
label='Tip_Rack3')

Tip_Rack4=protocol.load_labware('opentrons_96_tiprack_20ul',location='11',
label='Tip_Rack4')

    p20m = protocol.load_instrument('p20_multi_gen2', 'left', [Tip_Rack1,
Tip_Rack2, Tip_Rack3, Tip_Rack4]) #multi channel pipette

#constants
    Transfer_vol = 10 #uL

    for j in range(12):
        p20m.transfer(Transfer_vol, Lysate_Plate1.columns(j),
PCR_Plate1.columns(j), new_tip='always', blow_out = False, touch_tip = False)
        p20m.transfer(Transfer_vol, Controls_Plate.columns(0),
PCR_Plate1.columns(11), new_tip='always', blow_out = False, touch_tip =
False)

    for j in range(12):
        p20m.transfer(Transfer_vol, Lysate_Plate2.columns(j),
PCR_Plate2.columns(j), new_tip='always', blow_out = False, touch_tip = False)
        p20m.transfer(Transfer_vol, Controls_Plate.columns(0),
PCR_Plate2.columns(11), new_tip='always', blow_out = False, touch_tip =
False)

    for j in range(12):
        p20m.transfer(Transfer_vol, Lysate_Plate3.columns(j),
PCR_Plate3.columns(j), new_tip='always', blow_out = False, touch_tip = False)

```

```
p20m.transfer(Transfer_vol, Controls_Plate.columns(0),
PCR_Plate3.columns(11), new_tip='always', blow_out = False, touch_tip =
False)
```

Seal PCR plates and store in 4 C fridge (or continue); heat to 95 C for 5 min, cool to 4 C for at least 10 min before continuing to 6b2

6b2: Transfer from PCR plate to dot blot nitrocellulose membrane for dot blot (repeat 2x total)

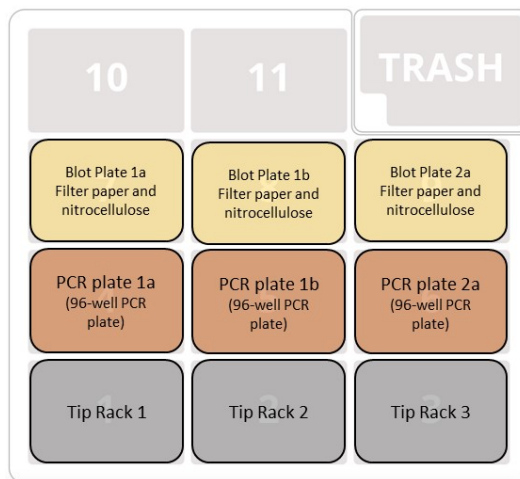


Figure S19. Protocol 6b2 plate set up.

```
"""
Transfer between PCR plate to dot blot membrane

"""
#Tips: 3 tip racks
from opentrons import protocol_api

metadata = {'apiLevel': '2.0',
            'protocolName': 'Dot blot denaturing and dotting',
            'author': 'Melody Morris',
            'description': 'Transfer lysate from PCR plate to blotting
plate'}

def run(protocol: protocol_api.ProtocolContext):
    pass

#Containers

PCR_Plate1=protocol.load_labware('pcrplateindeepwell_96_wellplate_125ul',location='4', label='PCR_Plate1')

PCR_Plate2=protocol.load_labware('pcrplateindeepwell_96_wellplate_125ul',location='5', label='PCR_Plate2')

PCR_Plate3=protocol.load_labware('pcrplateindeepwell_96_wellplate_125ul',location='6', label='PCR_Plate3')
```

```

Dot_Plate1=protocol.load_labware('dotblotbase_96_wellplate_10ul',location='7'
, label='Dot_Plate1')

Dot_Plate2=protocol.load_labware('dotblotbase_96_wellplate_10ul',location='8'
, label='Dot_Plate2')

Dot_Plate3=protocol.load_labware('dotblotbase_96_wellplate_10ul',location='9'
, label='Dot_Plate3')

#Tip Racks and Pipettes
Tip_Rack1=protocol.load_labware('opentrons_96_tiprack_20ul',location='1',
label='Tip_Rack1') #Tip Rack
Tip_Rack2=protocol.load_labware('opentrons_96_tiprack_20ul',location='2',
label='Tip_Rack2') #Tip Rack
Tip_Rack3=protocol.load_labware('opentrons_96_tiprack_20ul',location='3',
label='Tip_Rack3')

p20m = protocol.load_instrument('p20_multi_gen2', 'left', [Tip_Rack1,
Tip_Rack2, Tip_Rack3]) #multi channel pipette

#constants

Assay_vol = 3 #uL

for j in range(12):
    p20m.transfer(Assay_vol, PCR_Plate1.columns(j),
Dot_Plate1.columns(j), new_tip='always', blow_out = False, touch_tip = False)

    for j in range(12):
        p20m.transfer(Assay_vol, PCR_Plate2.columns(j),
Dot_Plate2.columns(j), new_tip='always', blow_out = False, touch_tip = False)

        for j in range(12):
            p20m.transfer(Assay_vol, PCR_Plate3.columns(j),
Dot_Plate3.columns(j), new_tip='always', blow_out = False, touch_tip = False)

##### Allow dots to fully dry before continuing

```

F. MATLAB robot analysis code (mCherry)

Also available at: <https://github.com/olsenlabmit/Democratizing-Protein-Expression>

```

% Script to assemble each proteins's data, plot results and create a
% database entry for future machine learning applications
clear
clc
close all
tic

%Input plate design (change as necessary)

ProteinofInterest = 'mCherry';

```

```

Vectors = ["pET15b", "pET22b", "pQE9", "pQE60", "pGEX4T-1H", "pETSUMO",
"pUC19", "No vector"];
Cells = ["BL21", "BL21(DE3)", "BL21*(DE3)", "Tuner(DE3)", "C41(DE3)",
"C43(DE3)", "NiCo21(DE3)", "T7 Express", "T7 Express LysY", "T7 Express
LysY/Iq", "Rosetta 2(DE3)", "Controls"];

% Assemble all growth data (change file names as necessary)
filename1 = '210226_mCherry_Selection+GrowthPlates.xlsx'; %move file to local
matlab folder for data assembly
filename2 = '210317_mCherryBradford.xlsx'; %move file to local matlab folder
for data assembly
filename3 = 'mCherryDotBlotAnalysis.xlsx'; %move file to local matlab folder
for data assembly
xlsrange = 'B25:M32';
xlsrange2 = 'C2:N9';
[SelectionPlate1] = xlsread(filename1, 'SelectionPlate1', xlsrange);
[SelectionPlate2] = xlsread(filename1, 'SelectionPlate2', xlsrange);
[SelectionPlate3] = xlsread(filename1, 'SelectionPlate3', xlsrange);
[GrowthPlate1A] = xlsread(filename1, 'GrowthPlate1A', xlsrange);
[GrowthPlate1B] = xlsread(filename1, 'GrowthPlate1B', xlsrange);
[GrowthPlate2A] = xlsread(filename1, 'GrowthPlate2A', xlsrange);
[GrowthPlate2B] = xlsread(filename1, 'GrowthPlate2B', xlsrange);
[GrowthPlate3A] = xlsread(filename1, 'GrowthPlate3A', xlsrange);
[GrowthPlate3B] = xlsread(filename1, 'GrowthPlate3B', xlsrange);
[ExpressionPlate1A] = xlsread(filename1, 'ExpressionPlate1A', xlsrange);
[ExpressionPlate1B] = xlsread(filename1, 'ExpressionPlate1B', xlsrange);
[ExpressionPlate2A] = xlsread(filename1, 'ExpressionPlate2A', xlsrange);
[ExpressionPlate2B] = xlsread(filename1, 'ExpressionPlate2B', xlsrange);
[ExpressionPlate3A] = xlsread(filename1, 'ExpressionPlate3A', xlsrange);
[ExpressionPlate3B] = xlsread(filename1, 'ExpressionPlate3B', xlsrange);

ODSelectionPlate1 = 2.87769784172662*(SelectionPlate1-min(SelectionPlate1));
ODSelectionPlate2 = 2.87769784172662*(SelectionPlate2-min(SelectionPlate2));
ODSelectionPlate3 = 2.87769784172662*(SelectionPlate3-min(SelectionPlate3));

ODGrowthPlate1A = 2.87769784172662*(GrowthPlate1A-min(GrowthPlate1A));
ODGrowthPlate1B = 2.87769784172662*(GrowthPlate1B-min(GrowthPlate1B));
ODGrowthPlate2A = 2.87769784172662*(GrowthPlate2A-min(GrowthPlate2A));
ODGrowthPlate2B = 2.87769784172662*(GrowthPlate2B-min(GrowthPlate2B));
ODGrowthPlate3A = 2.87769784172662*(GrowthPlate3A-min(GrowthPlate3A));
ODGrowthPlate3B = 2.87769784172662*(GrowthPlate3B-min(GrowthPlate3B));

ODExpressionPlate1A = 2.87769784172662*(ExpressionPlate1A-
min(ExpressionPlate1A));
ODExpressionPlate1B = 2.87769784172662*(ExpressionPlate1B-
min(ExpressionPlate1B));
ODExpressionPlate2A = 2.87769784172662*(ExpressionPlate2A-
min(ExpressionPlate2A));
ODExpressionPlate2B = 2.87769784172662*(ExpressionPlate2B-
min(ExpressionPlate2B));
ODExpressionPlate3A = 2.87769784172662*(ExpressionPlate3A-
min(ExpressionPlate3A));
ODExpressionPlate3B = 2.87769784172662*(ExpressionPlate3B-
min(ExpressionPlate3B));

% Assemble Bradford assay data

```

```

[Bradford1A] = xlsread(filename2, 'Plate1A', xlsrange);
[Bradford1B] = xlsread(filename2, 'Plate1B', xlsrange);
[Bradford2A] = xlsread(filename2, 'Plate2A', xlsrange);
[Bradford2B] = xlsread(filename2, 'Plate2B', xlsrange);
[Bradford3A] = xlsread(filename2, 'Plate3A', xlsrange);
[Bradford3B] = xlsread(filename2, 'Plate3B', xlsrange);
concBradford1A = 4.5625*Bradford1A -1.8434*ones(8,12);
concBradford1B = 4.5625*Bradford1B -1.8434*ones(8,12);
concBradford2A = 4.5625*Bradford2A -1.8434*ones(8,12);
concBradford2B = 4.5625*Bradford2B -1.8434*ones(8,12);
concBradford3A = 4.5625*Bradford3A -1.8434*ones(8,12);
concBradford3B = 4.5625*Bradford3B -1.8434*ones(8,12);

% Assemble dot blot assay data

[DotBlot1A] = xlsread(filename3, 'Plate1A', xlsrange2);
[DotBlot1B] = xlsread(filename3, 'Plate1B', xlsrange2);
[DotBlot2A] = xlsread(filename3, 'Plate2A', xlsrange2);
[DotBlot2B] = xlsread(filename3, 'Plate2B', xlsrange2);
[DotBlot3A] = xlsread(filename3, 'Plate3A', xlsrange2);
[DotBlot3B] = xlsread(filename3, 'Plate3B', xlsrange2);
concDotBlot1A= DotBlot1A*fitlm(DotBlot1A(6:8,12), [.25; .0625; 0], 'Intercept',
false).Coefficients^1;
concDotBlot1B= DotBlot1B*fitlm(DotBlot1B(6:8,12), [.25; .0625; 0], 'Intercept',
false).Coefficients{1,1};
concDotBlot2A= DotBlot2A*fitlm(DotBlot2A(6:8,12), [.25; .0625; 0], 'Intercept',
false).Coefficients{1,1};
concDotBlot2B= DotBlot2B*fitlm(DotBlot2B(6:8,12), [.25; .0625; 0], 'Intercept',
false).Coefficients{1,1};
concDotBlot3A= DotBlot3A*fitlm(DotBlot3A(6:8,12), [.25; .0625; 0], 'Intercept',
false).Coefficients{1,1};
concDotBlot3B= DotBlot3B*fitlm(DotBlot3B(6:8,12), [.25; .0625; 0], 'Intercept',
false).Coefficients{1,1};

% Compiling all data for each condition in matrix form

[CompiledData] = zeros(96,44);

for i = 1:8;
    for j = 1:12;
        k = 12*(i-1)+j;
        CompiledData(k,1)=i;
        CompiledData(k,2)=j;
        CompiledData(k,3)=ODSelectionPlate1(i,j);
        CompiledData(k,4)=ODSelectionPlate2(i,j);
        CompiledData(k,5)=ODSelectionPlate3(i,j);

        ODSelectionPlate=[ODSelectionPlate1(i,j),ODSelectionPlate2(i,j),ODSelectionPlate3(i,j)];
        N_ODSelection = sum(ODSelectionPlate > 0.05);
        CompiledData(k,6)=N_ODSelection;
        if N_ODSelection<1
            CompiledData(k,7) = mean(ODSelectionPlate);
            CompiledData(k,8) = std(ODSelectionPlate)/sqrt(6);
        else
            x = (ODSelectionPlate>0.05).*ODSelectionPlate;
            x(x==0)=[];

```

```

        CompiledData(k,7) = mean(x);
        CompiledData(k,8) = std(x)/sqrt(N_ODSelection);
    end
    CompiledData(k,9)=ODGrowthPlate1A(i,j);
    CompiledData(k,10)=ODGrowthPlate1B(i,j);
    CompiledData(k,11)=ODGrowthPlate2A(i,j);
    CompiledData(k,12)=ODGrowthPlate2B(i,j);
    CompiledData(k,13)=ODGrowthPlate3A(i,j);
    CompiledData(k,14)=ODGrowthPlate3B(i,j);

    ODGrowthPlate=[ODGrowthPlate1A(i,j),ODGrowthPlate1B(i,j),ODGrowthPlate2A(i,j),
    ,ODGrowthPlate2B(i,j), ODGrowthPlate3A(i,j), ODGrowthPlate3B(i,j)];
    N_ODGrowthPlate = sum(ODGrowthPlate > 0.05);
    CompiledData(k,15)=N_ODGrowthPlate;
    if N_ODGrowthPlate<1
        CompiledData(k,16) = mean(ODGrowthPlate);
        CompiledData(k,17) = std(ODGrowthPlate)/sqrt(6);
    else
        x = (ODGrowthPlate>0.05).*ODGrowthPlate;
        x(x==0)=[];
        CompiledData(k,16) = mean(x);
        CompiledData(k,17) = std(x)/sqrt(N_ODGrowthPlate);
    end
    CompiledData(k,18)=ODExpressionPlate1A(i,j);
    CompiledData(k,19)=ODExpressionPlate1B(i,j);
    CompiledData(k,20)=ODExpressionPlate2A(i,j);
    CompiledData(k,21)=ODExpressionPlate2B(i,j);
    CompiledData(k,22)=ODExpressionPlate3A(i,j);
    CompiledData(k,23)=ODExpressionPlate3B(i,j);

    ODExpressionPlate=[ODExpressionPlate1A(i,j),ODExpressionPlate1B(i,j),ODExpressionPlate2A(i,j),
    ODExpressionPlate2B(i,j), ODExpressionPlate3A(i,j),
    ODExpressionPlate3B(i,j)];
    N_ODExpressionPlate = sum(ODExpressionPlate > 0.05);
    CompiledData(k,24)=N_ODExpressionPlate;
    if N_ODExpressionPlate<1
        CompiledData(k,25) = mean(ODExpressionPlate);
        CompiledData(k,26) = std(ODExpressionPlate)/sqrt(6);
    else
        x = (ODExpressionPlate>0.05).*ODExpressionPlate;
        x(x==0)=[];
        CompiledData(k,25) = mean(x);
        CompiledData(k,26) = std(x)/sqrt(N_ODExpressionPlate);
    end
    CompiledData(k,27)=concBradford1A(i,j);
    CompiledData(k,28)=concBradford1B(i,j);
    CompiledData(k,29)=concBradford2A(i,j);
    CompiledData(k,30)=concBradford2B(i,j);
    CompiledData(k,31)=concBradford3A(i,j);
    CompiledData(k,32)=concBradford3B(i,j);

    concBradford=[concBradford1A(i,j),concBradford1B(i,j),concBradford2A(i,j),concBradford2B(i,j),
    concBradford3A(i,j), concBradford3B(i,j)];
    N_concBradford = sum(CompiledData(k,9:14)>.05);
    CompiledData(k,33)=N_concBradford;
    if N_concBradford<1
        CompiledData(k,34) = mean(concBradford);

```

```

        CompiledData(k,35) = std(concBradford)/sqrt(6);
    else
        x = (CompiledData(k,9:14)>.05).*concBradford;
        x(x==0)=[];
        CompiledData(k,34) = mean(x);
        CompiledData(k,35) = std(x)/sqrt(N_concBradford);
    end
    CompiledData(k,36)=concDotBlot1A(i,j);
    CompiledData(k,37)=concDotBlot1B(i,j);
    CompiledData(k,38)=concDotBlot2A(i,j);
    CompiledData(k,39)=concDotBlot2B(i,j);
    CompiledData(k,40)=concDotBlot3A(i,j);
    CompiledData(k,41)=concDotBlot3B(i,j);

concDotBlot=[concDotBlot1A(i,j),concDotBlot1B(i,j),concDotBlot2A(i,j),concDot
Blot2B(i,j), concDotBlot3A(i,j), concDotBlot3B(i,j)];
DotBlotLim=0.0;
N_concDotBlot = sum(CompiledData(k,9:14)>.05);
CompiledData(k,42)=N_concDotBlot;
if N_concDotBlot<1
    CompiledData(k,43) = mean(concDotBlot);
    CompiledData(k,44) = std(concDotBlot)/sqrt(6);
else
    x = (CompiledData(k,9:14)>.05).*concDotBlot;
    x(x==0)=[];
    CompiledData(k,43) = mean(x);
    CompiledData(k,44) = std(x)/sqrt(N_concDotBlot);
end
end
end
CompiledDataNoControls=CompiledData;
CompiledDataNoControls(CompiledDataNoControls(:,2)==12,:)=[];
CompiledDataNoControls(CompiledDataNoControls(:,1)==8,:)=[];

%Final OD vs Bradford sorted by vector type
figure
for n=1:7
    hold on
    if CompiledDataNoControls(11*n-10,1)==1
        plot(CompiledDataNoControls(11*n-
10:11*n,25),CompiledDataNoControls(11*n-10:11*n,34),'Color', '#e41a1c',
'Linestyle', 'none', 'Marker', 'o','DisplayName', Vectors(1))
    elseif CompiledDataNoControls(11*n-10,1)==2
        plot(CompiledDataNoControls(11*n-
10:11*n,25),CompiledDataNoControls(11*n-10:11*n,34),'Color','#377eb8',
'Linestyle', 'none', 'Marker', '*', 'DisplayName', Vectors(2))
    elseif CompiledDataNoControls(11*n-10,1)==3
        plot(CompiledDataNoControls(11*n-
10:11*n,25),CompiledDataNoControls(11*n-10:11*n,34),'Color', '#4daf4a',
'Linestyle', 'none', 'Marker', 's', 'DisplayName', Vectors(3))
    elseif CompiledDataNoControls(11*n-10,1)==4
        plot(CompiledDataNoControls(11*n-
10:11*n,25),CompiledDataNoControls(11*n-10:11*n,34),'Color', '#984ea3',
'Linestyle', 'none', 'Marker', 'd', 'DisplayName', Vectors(4))
    elseif CompiledDataNoControls(11*n-10,1)==5

```

```

        plot(CompiledDataNoControls(11*n-
10:11*n,25),CompiledDataNoControls(11*n-10:11*n,34),'Color', '#ff7f00',
'Linestyle', 'none', 'Marker', '^', 'DisplayName', Vectors(5))
        elseif CompiledDataNoControls(11*n-10,1)==6
            plot(CompiledDataNoControls(11*n-
10:11*n,25),CompiledDataNoControls(11*n-10:11*n,34),'Color', '#a65628',
'Linestyle', 'none', 'Marker', 'p', 'DisplayName', Vectors(6))
        elseif CompiledDataNoControls(11*n-10,1)==7
            plot(CompiledDataNoControls(11*n-
10:11*n,25),CompiledDataNoControls(11*n-10:11*n,34),'Color',
'#f781bf','Linestyle', 'none', 'Marker', 'h', 'DisplayName', Vectors(7))
        end
    end
    erbar=errorbar(CompiledDataNoControls(:,25),CompiledDataNoControls(:,34),Comp
iledDataNoControls(:,35), CompiledDataNoControls(:,35),
CompiledDataNoControls(:,26), CompiledDataNoControls(:,26),'LineStyle',
'none', 'Marker', '.', 'Color', '[0.5 0.5 0.5]');
    erbar.Annotation.LegendInformation.IconDisplayStyle = 'off';
    xlabel('Final OD600')
    ylabel('Total protein concentration (mg/mL)')
    ylim([0 5])
    title('Final OD600 vs protein concentration sorted by vector type')
    legend show
    legend('Location', 'westoutside')
    axis square
    hold off

%Final OD vs Bradford sorted by cell type
figure
for n=1:11
    hold on
    j=[n n+11 n+22 n+33 n+44 n+55 n+66];
    if CompiledDataNoControls(j(1),2)==1

plot(CompiledDataNoControls(j,25),CompiledDataNoControls(j,34),'Color',
'#a6cee3', 'Linestyle', 'none', 'Marker', 'o', 'DisplayName', Cells(1))
        elseif CompiledDataNoControls(j(1),2)==2

plot(CompiledDataNoControls(j,25),CompiledDataNoControls(j,34),'Color','#1f78
b4', 'Linestyle', 'none', 'Marker', '+', 'DisplayName', Cells(2))
        elseif CompiledDataNoControls(j(1),2)==3

plot(CompiledDataNoControls(j,25),CompiledDataNoControls(j,34),'Color',
'#b2df8a', 'Linestyle', 'none', 'Marker', '*', 'DisplayName', Cells(3))
        elseif CompiledDataNoControls(j(1),2)==4

plot(CompiledDataNoControls(j,25),CompiledDataNoControls(j,34),'Color',
'#33a02c', 'Linestyle', 'none', 'Marker', 'x', 'DisplayName', Cells(4))
        elseif CompiledDataNoControls(j(1),2)==5

plot(CompiledDataNoControls(j,25),CompiledDataNoControls(j,34),'Color',
'#fb9a99', 'Linestyle', 'none', 'Marker', 's', 'DisplayName', Cells(5))
        elseif CompiledDataNoControls(j(1),2)==6

plot(CompiledDataNoControls(j,25),CompiledDataNoControls(j,34),'Color',
'#e31a1c', 'Linestyle', 'none', 'Marker', 'd', 'DisplayName', Cells(6))
        elseif CompiledDataNoControls(j(1),2)==7

```

```

plot(CompiledDataNoControls(j,25),CompiledDataNoControls(j,34),'Color',
'#fdbf6f','Linestyle','none','Marker','^','DisplayName',Cells(7))
elseif CompiledDataNoControls(j(1),2)==8

plot(CompiledDataNoControls(j,25),CompiledDataNoControls(j,34),'Color',
'#ff7f00','Linestyle','none','Marker','>','DisplayName',Cells(8))
elseif CompiledDataNoControls(j(1),2)==9

plot(CompiledDataNoControls(j,25),CompiledDataNoControls(j,34),'Color',
'#cab2d6','Linestyle','none','Marker','<','DisplayName',Cells(9))
elseif CompiledDataNoControls(j(1),2)==10

plot(CompiledDataNoControls(j,25),CompiledDataNoControls(j,34),'Color',
'#6a3d9a','Linestyle','none','Marker','p','DisplayName',Cells(10))
elseif CompiledDataNoControls(j(1),2)==11

plot(CompiledDataNoControls(j,25),CompiledDataNoControls(j,34),'Color',
'#b15928','Linestyle','none','Marker','h','DisplayName',Cells(11))
end
end
erbar=errorbar(CompiledDataNoControls(:,25),CompiledDataNoControls(:,34),Comp
iledDataNoControls(:,35), CompiledDataNoControls(:,35),
CompiledDataNoControls(:,26), CompiledDataNoControls(:,26),'LineStyle',
'none','Marker','.', 'Color', '[0.5 0.5 0.5]');
erbar.Annotation.LegendInformation.IconDisplayStyle = 'off';
xlabel('Final OD600')
ylabel('Total protein concentration (mg/mL)')
ylim([0 5])
title('Final OD600 vs protein concentration sorted by cell type')
legend show
legend('Location', 'westoutside')
axis square
hold off

%Final OD vs Dot Blot sorted by vector type
figure
for n=1:7
    hold on
    if CompiledDataNoControls(11*n-10,1)==1
        plot(CompiledDataNoControls(11*n-
10:11*n,25),CompiledDataNoControls(11*n-10:11*n,43),'Color', '#e41a1c',
'Linestyle','none','Marker','o','DisplayName', Vectors(1))
    elseif CompiledDataNoControls(11*n-10,1)==2
        plot(CompiledDataNoControls(11*n-
10:11*n,25),CompiledDataNoControls(11*n-10:11*n,43),'Color', '#377eb8',
'Linestyle','none','Marker','*','DisplayName', Vectors(2))
    elseif CompiledDataNoControls(11*n-10,1)==3
        plot(CompiledDataNoControls(11*n-
10:11*n,25),CompiledDataNoControls(11*n-10:11*n,43),'Color', '#4daf4a',
'Linestyle','none','Marker','s','DisplayName', Vectors(3))
    elseif CompiledDataNoControls(11*n-10,1)==4
        plot(CompiledDataNoControls(11*n-
10:11*n,25),CompiledDataNoControls(11*n-10:11*n,43),'Color', '#984ea3',
'Linestyle','none','Marker','d','DisplayName', Vectors(4))
    elseif CompiledDataNoControls(11*n-10,1)==5

```

```

        plot(CompiledDataNoControls(11*n-
10:11*n,25),CompiledDataNoControls(11*n-10:11*n,43),'Color', '#ff7f00',
'Linestyle', 'none', 'Marker', '^', 'DisplayName', Vectors(5))
        elseif CompiledDataNoControls(11*n-10,1)==6
            plot(CompiledDataNoControls(11*n-
10:11*n,25),CompiledDataNoControls(11*n-10:11*n,43),'Color', '#a65628',
'Linestyle', 'none', 'Marker', 'p', 'DisplayName', Vectors(6))
            elseif CompiledDataNoControls(11*n-10,1)==7
                plot(CompiledDataNoControls(11*n-
10:11*n,25),CompiledDataNoControls(11*n-10:11*n,43),'Color',
'#f781bf','Linestyle', 'none', 'Marker', 'h', 'DisplayName', Vectors(7))
            end
        end
    end
    erbar=errorbar(CompiledDataNoControls(:,25),CompiledDataNoControls(:,43),Comp
iledDataNoControls(:,44), CompiledDataNoControls(:,44),
CompiledDataNoControls(:,26), CompiledDataNoControls(:,26),'LineStyle',
'none', 'Marker', '.', 'Color', '[0.5 0.5 0.5]');
    erbar.Annotation.LegendInformation.IconDisplayStyle = 'off';
    xlabel('Final OD600')
    ylabel('Concentration of protein of interest (mg/mL)')
    ylim([0,round(max(CompiledDataNoControls(:,43))+.15,2)])
    title('Final OD600 vs concentration of protein of interest sorted by vector
type')
    legend show
    legend('Location', 'westoutside')
    axis square
    hold off

%Final OD vs Dot blot sorted by cell type
figure
for n=1:11
    hold on
    j=[n n+11 n+22 n+33 n+44 n+55 n+66];
    if CompiledDataNoControls(j(1),2)==1

plot(CompiledDataNoControls(j,25),CompiledDataNoControls(j,43),'Color',
'#a6cee3', 'Linestyle', 'none', 'Marker', 'o', 'DisplayName', Cells(1))
        elseif CompiledDataNoControls(j(1),2)==2

plot(CompiledDataNoControls(j,25),CompiledDataNoControls(j,43),'Color','#1f78
b4', 'Linestyle', 'none', 'Marker', '+', 'DisplayName', Cells(2))
        elseif CompiledDataNoControls(j(1),2)==3

plot(CompiledDataNoControls(j,25),CompiledDataNoControls(j,43),'Color',
'#b2df8a', 'Linestyle', 'none', 'Marker', '*', 'DisplayName', Cells(3))
        elseif CompiledDataNoControls(j(1),2)==4

plot(CompiledDataNoControls(j,25),CompiledDataNoControls(j,43),'Color',
'#33a02c', 'Linestyle', 'none', 'Marker', 'x', 'DisplayName', Cells(4))
        elseif CompiledDataNoControls(j(1),2)==5

plot(CompiledDataNoControls(j,25),CompiledDataNoControls(j,43),'Color',
'#fb9a99', 'Linestyle', 'none', 'Marker', 's', 'DisplayName', Cells(5))
        elseif CompiledDataNoControls(j(1),2)==6

plot(CompiledDataNoControls(j,25),CompiledDataNoControls(j,43),'Color',
'#e31a1c', 'Linestyle', 'none', 'Marker', 'd', 'DisplayName', Cells(6))

```

```

elseif CompiledDataNoControls(j(1),2)==7

plot(CompiledDataNoControls(j,25),CompiledDataNoControls(j,43),'Color',
'#fdbf6f','Linestyle','none','Marker','^','DisplayName',Cells(7))
elseif CompiledDataNoControls(j(1),2)==8

plot(CompiledDataNoControls(j,25),CompiledDataNoControls(j,43),'Color',
'#ff7f00','Linestyle','none','Marker','>','DisplayName',Cells(8))
elseif CompiledDataNoControls(j(1),2)==9

plot(CompiledDataNoControls(j,25),CompiledDataNoControls(j,43),'Color',
'#cab2d6','Linestyle','none','Marker','<','DisplayName',Cells(9))
elseif CompiledDataNoControls(j(1),2)==10

plot(CompiledDataNoControls(j,25),CompiledDataNoControls(j,43),'Color',
'#6a3d9a','Linestyle','none','Marker','p','DisplayName',Cells(10))
elseif CompiledDataNoControls(j(1),2)==11

plot(CompiledDataNoControls(j,25),CompiledDataNoControls(j,43),'Color',
'#b15928','Linestyle','none','Marker','h','DisplayName',Cells(11))
end
end
erbar=errorbar(CompiledDataNoControls(:,25),CompiledDataNoControls(:,43),Comp
iledDataNoControls(:,44),CompiledDataNoControls(:,44),
CompiledDataNoControls(:,26),CompiledDataNoControls(:,26),'LineStyle',
'none','Marker','.', 'Color', '[0.5 0.5 0.5]');
erbar.Annotation.LegendInformation.IconDisplayStyle = 'off';
xlabel('Final OD600')
ylabel('Concentration of protein of interest(mg/mL)')
ylim([0,round(max(CompiledDataNoControls(:,43))+.15,2)])
title('Final OD600 vs concentration of protein of interest sorted by cell
type')
legend show
legend('Location','westoutside')
axis square
hold off

varnames={'Row number','Column number','ODSelectionPlate1',
'ODSelectionPlate2',...
'ODSelectionPlate3','Number of Plates Transformed','Average Selection
OD',...
'Std Dev Selection Plates','ODGrowthPlate1A','ODGrowthPlate1B',
'ODGrowthPlate2A',...
'ODGrowthPlate2B','ODGrowthPlate3A','ODGrowthPlate3B','Number of
Transformed Growth Plates',...
'Average Growth Plate OD','Std Dev Growth Plates',
'ODExpressionPlate1A',...
'ODExpressionPlate1B','ODExpressionPlate2A','ODExpressionPlate2B',
'ODExpressionPlate3A',...
'ODExpressionPlate3B','Number of ExpressionPlates Transformed','Average
Expression Plate OD',...
'Std Dev Expression Plate','BradfordPlate1A','BradfordPlate1B',
'BradfordPlate2A',...
'BradfordPlate2B','BradfordPlate3A','BradfordPlate3B','Number
Bradford','Average Bradford',...
'Std Dev Bradford','DotBlotPlate1A','DotBlotPlate1B','DotBlotPlate2A',
'DotBlotPlate2B',...

```

```

        'DotBlotPlate3A', 'DotBlotPlate3B', 'Number DotBlotPlates', 'Average
DotBlot', 'Std Dev DotBlot'};
DatabaseEntry=array2table(CompiledData, 'VariableNames', varnames);
VectorsType = DatabaseEntry.(1);
CellsType = DatabaseEntry.(2);
FirstColumn=table('Size', [96 3], 'VariableTypes',
{'string', 'string', 'string'}, 'VariableNames', {'Protein of Interest',
'Vector', 'Cell'});
for i=1:96
    FirstColumn(i,1)={ProteinofInterest};
    FirstColumn(i,2) = {Vectors(VectorsType(i))};
    FirstColumn(i,3) = {Cells(CellsType(i))};
end

DatabaseEntry=[FirstColumn DatabaseEntry];

exportfilename= strcat(ProteinofInterest, 'Assembled.xlsx');

writetable(DatabaseEntry, exportfilename)

CompiledDataNoPuc19NoControls = CompiledDataNoControls;
CompiledDataNoPuc19NoControls(CompiledDataNoPuc19NoControls(:,1)==7,:)=[];
heatmapdotblot = reshape(CompiledDataNoPuc19NoControls(:,43), [11 6]).';
%heatmapselectivity = reshape(CompiledDataNoPuc19NoControls(:,45), [11 6]).';
heatmapfilename = strcat(ProteinofInterest, 'Heatmap.xlsx');
writematrix(heatmapdotblot, heatmapfilename);
figure
h=heatmap(Cells(1:11), Vectors(1:6), heatmapdotblot, 'Colormap', cool,
'CellLabelColor', 'none', 'ColorLimits', [0
max(CompiledDataNoPuc19NoControls(:,43))]);
h.XLabel = 'Cell strains';
h.YLabel='Vectors';
h.Title = ProteinofInterest;
h.FontSize=30;
h.FontName = 'Calibri';
%writematrix(heatmapselectivity, heatmapfilename, 'Sheet', 2);

figure
imageSizeX = 4950;
imageSizeY = 2700;
[columnsInImage rowsInImage] = meshgrid(1:imageSizeX, 1:imageSizeY);
DotBlotImage=zeros(2700,4950);
for i=1:11
    centerX = 25+200+(i-1)*450;
    for j=1:6
        centerY=225+(j-1)*450;
        radius=sqrt(heatmapdotblot(j,i)/max(max(heatmapdotblot)))*200;
        circlePixels = (rowsInImage - centerY).^2 ...
            + (columnsInImage - centerX).^2 <= radius.^2;

        DotBlotImage=DotBlotImage+(256+63.5*log10(heatmapdotblot(j,i)/.961))*circlePi
xels;
    end
end
image(DotBlotImage);
load('MyColormap.mat', 'mymap')

```

```

colormap(mymap)
daspect([1 1 1])

% Cluster testing
ClusterCompiledData=[CompiledDataNoControls(:,25),CompiledDataNoControls(:,43
)];
Z = linkage(ClusterCompiledData, 'complete');
Tvector=cluster(Z, 'maxclust', 6);
Tcell=cluster(Z, 'maxclust', 11);
Rand_vector =
rand_index(Tvector,CompiledDataNoPuc19NoControls(:,1), 'adjusted');
Rand_cell =
rand_index(Tvector,CompiledDataNoPuc19NoControls(:,1), 'adjusted');
figure
gscatter(CompiledDataNoPuc19NoControls(:,25),CompiledDataNoPuc19NoControls(:,
43),Tvector)
figure
gscatter(CompiledDataNoPuc19NoControls(:,25),CompiledDataNoPuc19NoControls(:,
43),Tcell)

```

G. Assembled data for constructs

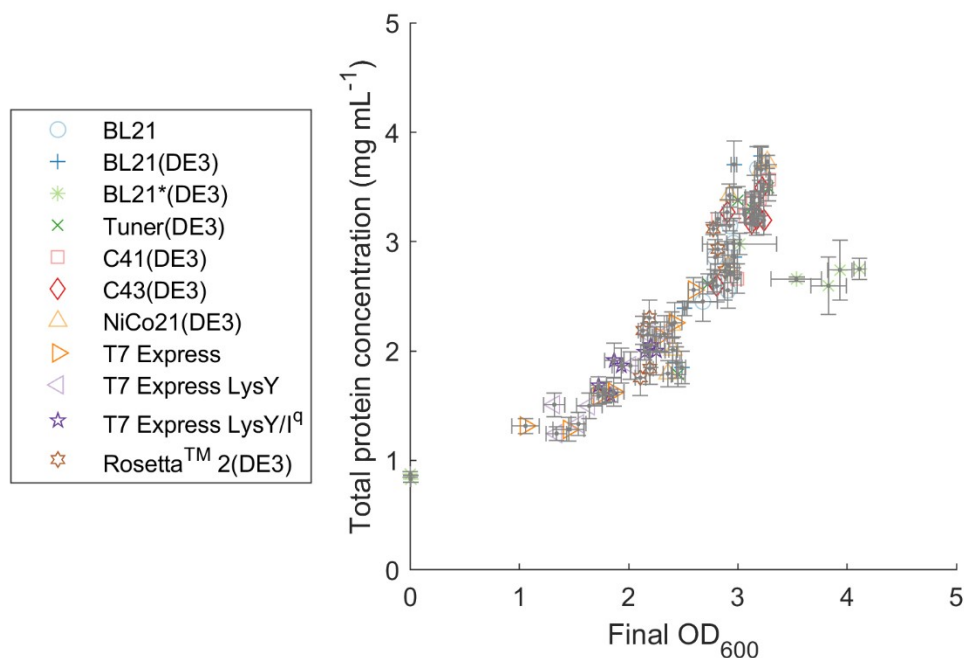


Figure S20. Final OD₆₀₀ vs. total protein concentration for 047A determined by Bradford assay, sorted by strain. Error bars reflect the standard error across successfully transformed samples.

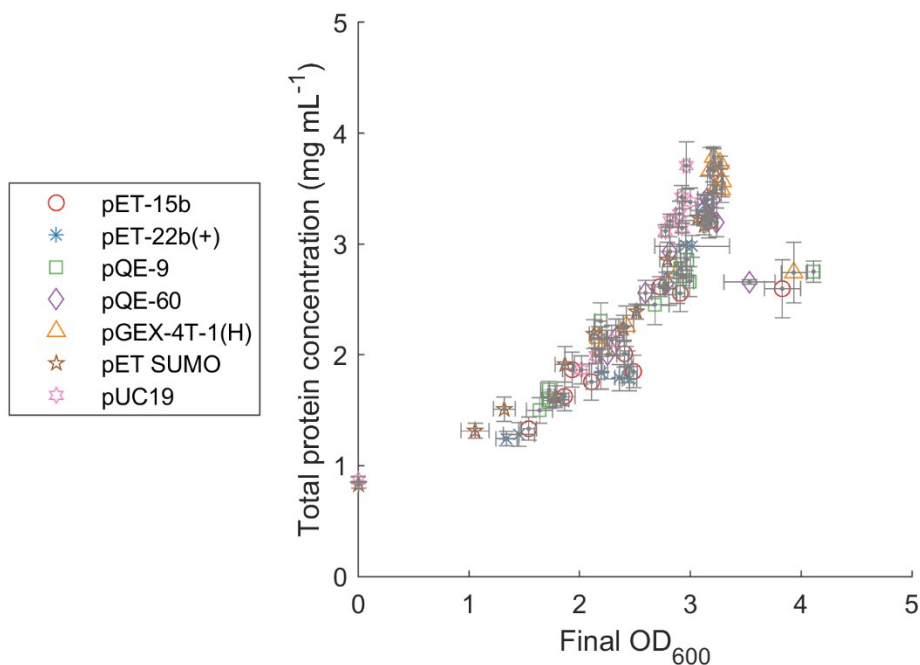


Figure S21. Final OD₆₀₀ vs. total protein concentration for 047A determined by Bradford assay, sorted by vector. Error bars reflect the standard error across successfully transformed samples.

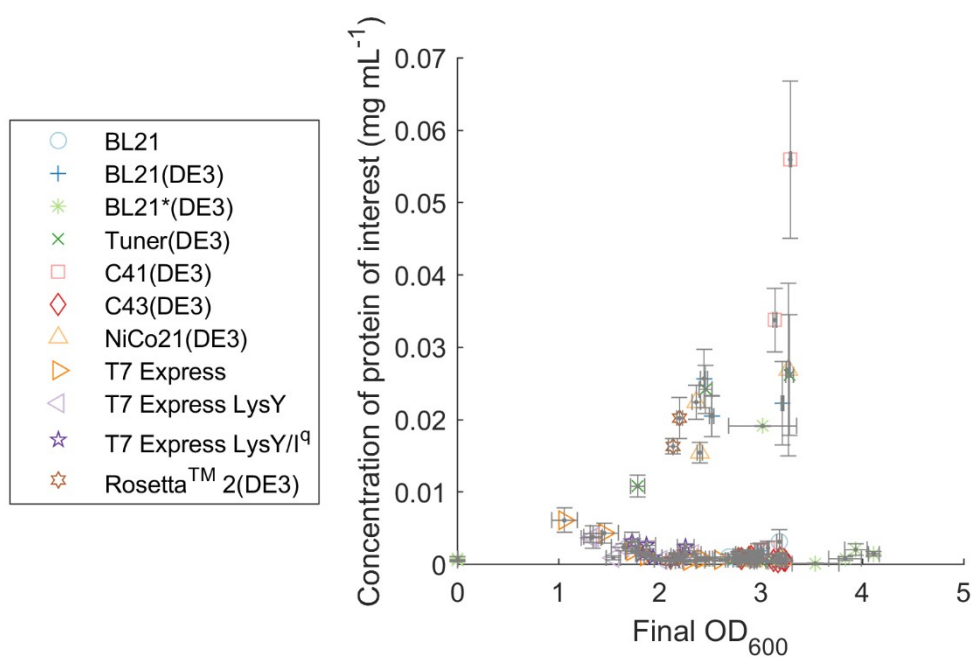


Figure S22. Final OD₆₀₀ vs. concentration of 047A, determined by dot blot assay, sorted by strain. Error bars reflect the standard error across successfully transformed samples.

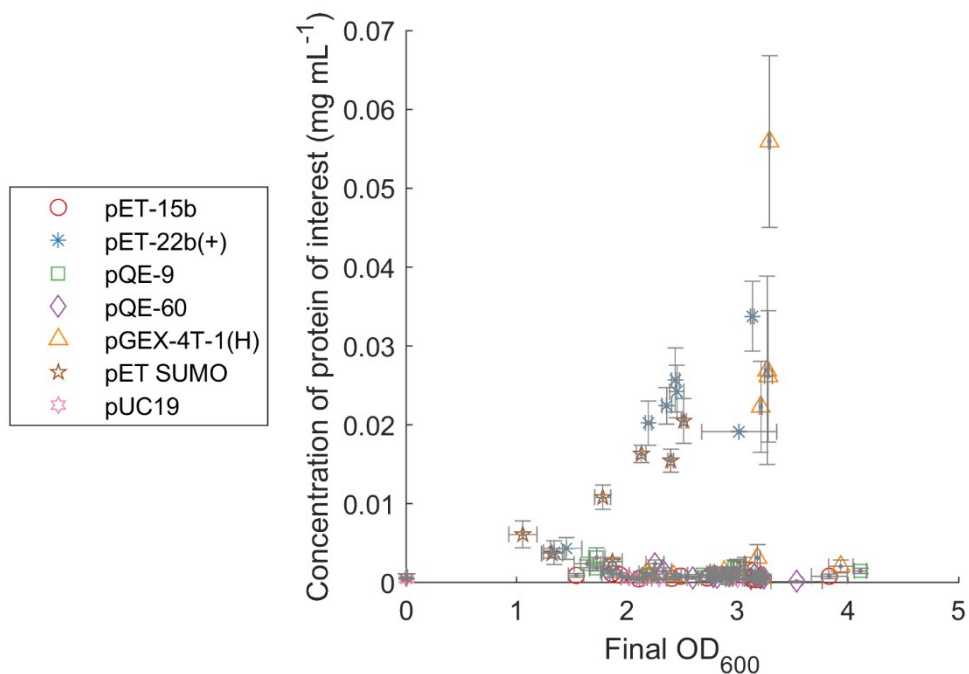


Figure S23. Final OD₆₀₀ vs. concentration of 047A, determined by dot blot assay, sorted by vector. Error bars reflect the standard error across successfully transformed samples.

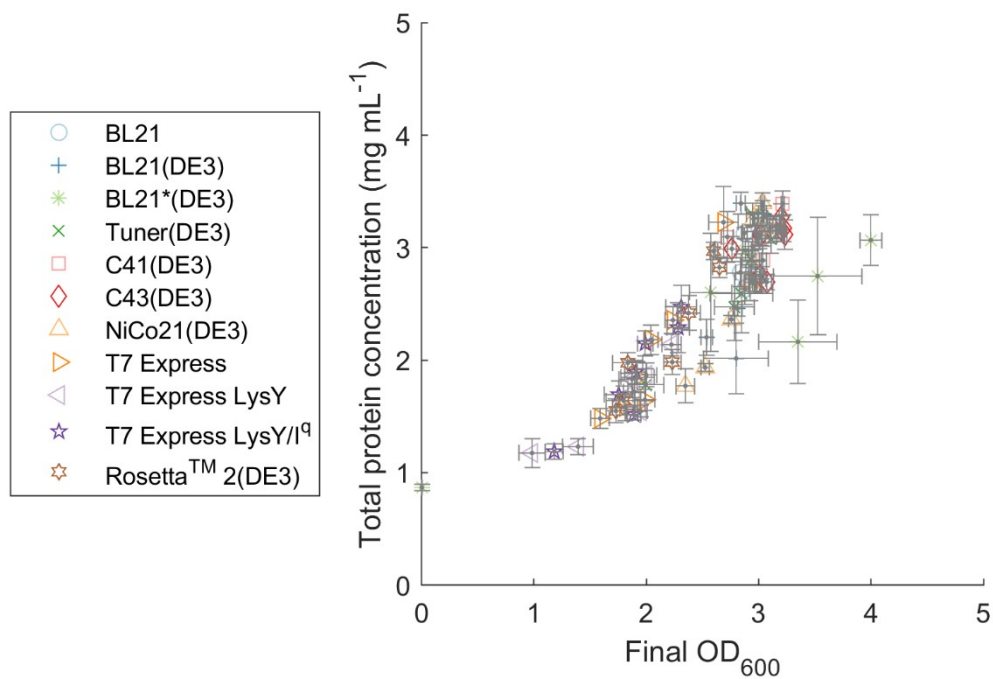


Figure S24. Final OD₆₀₀ vs. total protein concentration for 047B determined by Bradford assay, sorted by strain. Error bars reflect the standard error across successfully transformed samples.

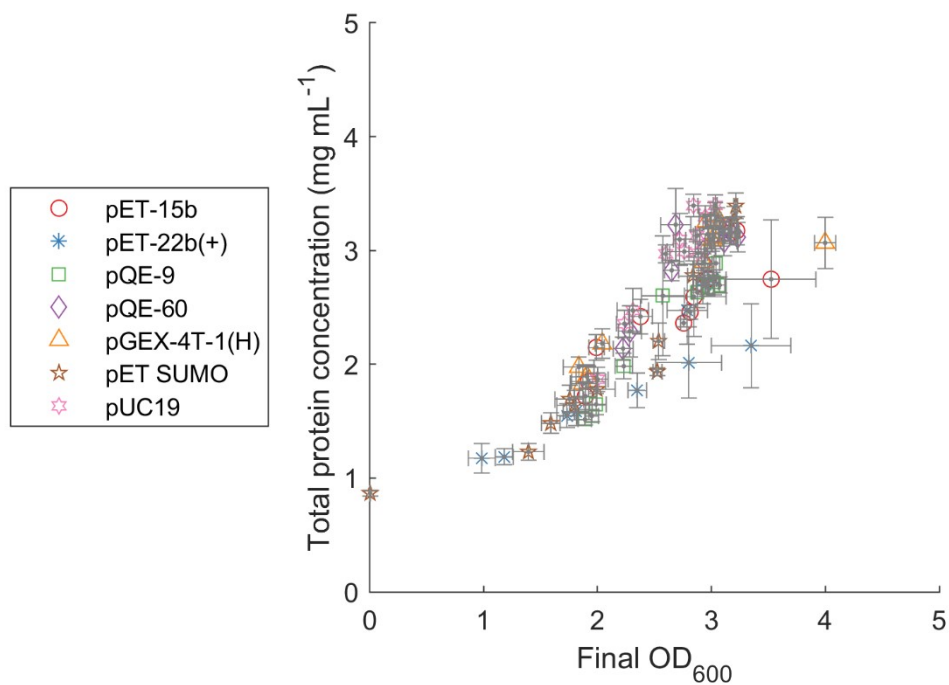


Figure S25. Final OD₆₀₀ vs. total protein concentration for 047B determined by Bradford assay, sorted by vector. Error bars reflect the standard error across successfully transformed samples.

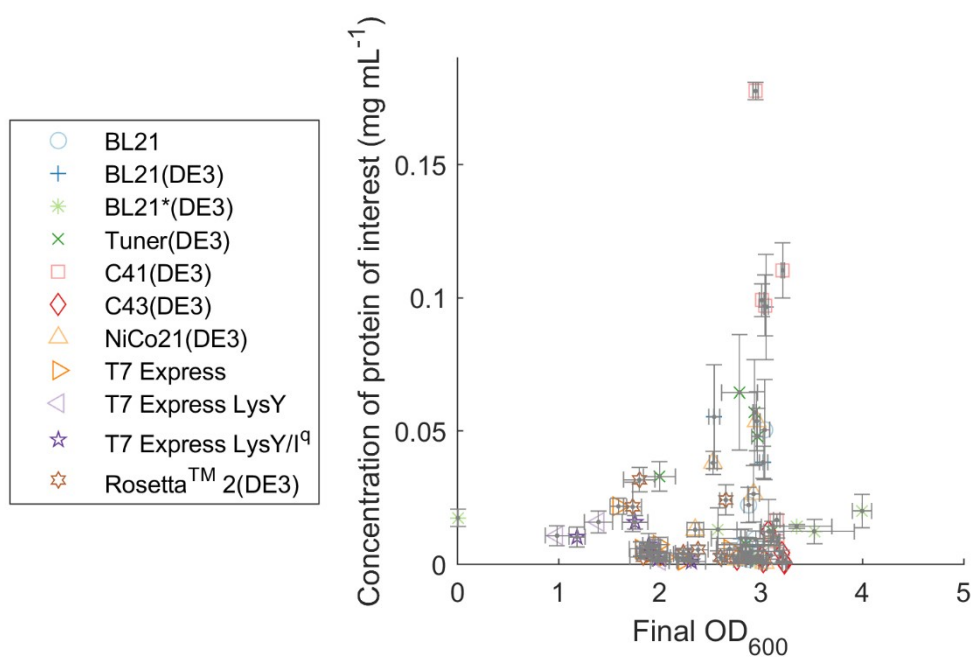


Figure S26. Final OD₆₀₀ vs. concentration of 047B, determined by dot blot assay, sorted by strain. Error bars reflect the standard error across successfully transformed samples.

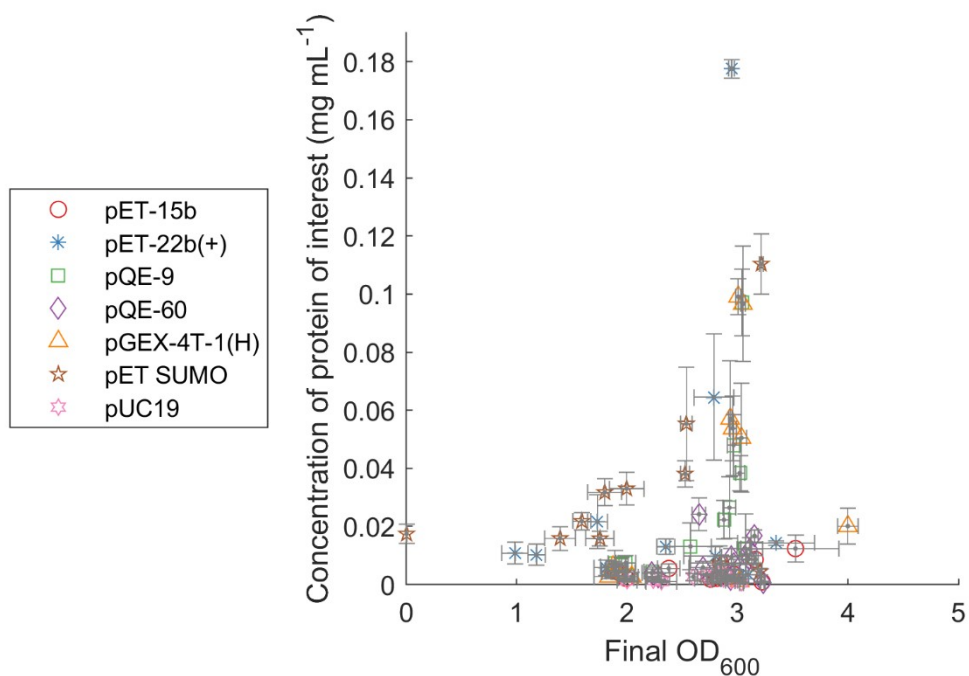


Figure S27. Final OD₆₀₀ vs. concentration of 047B, determined by dot blot assay, sorted by vector. Error bars reflect the standard error across successfully transformed samples.

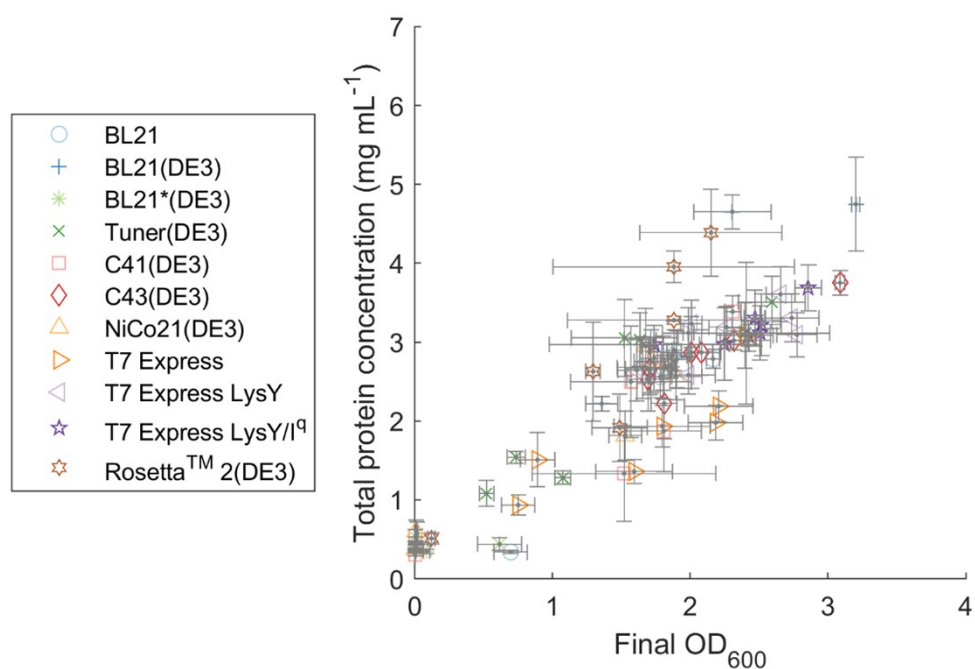


Figure S28. Final OD₆₀₀ vs. total protein concentration for Catcher determined by Bradford assay, sorted by strain. Error bars reflect the standard error across successfully transformed samples.

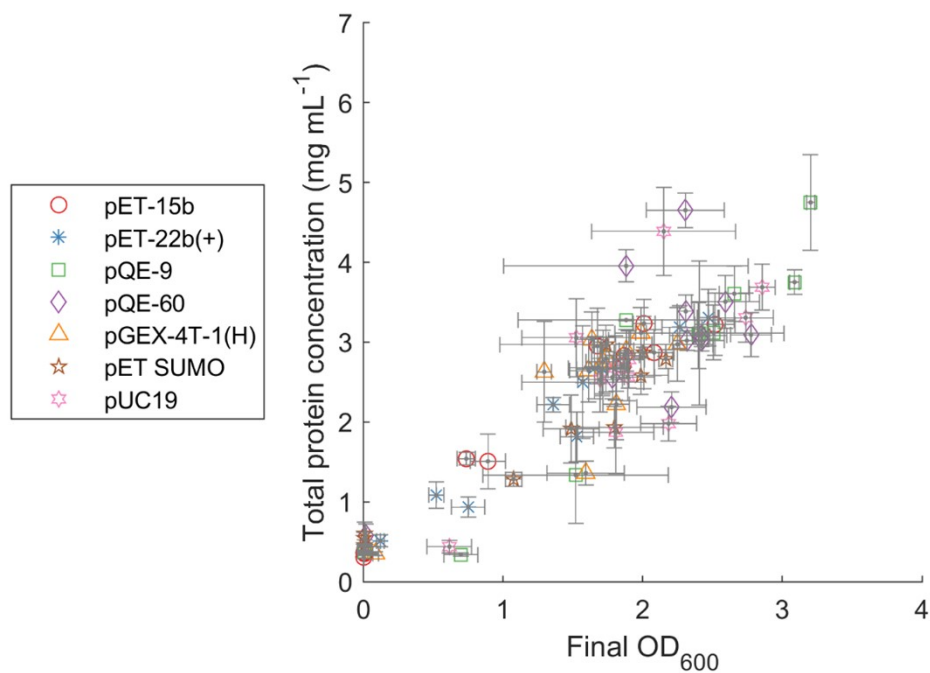


Figure S29. Final OD₆₀₀ vs. total protein concentration for Catcher determined by Bradford assay, sorted by vector. Error bars reflect the standard error across successfully transformed samples.

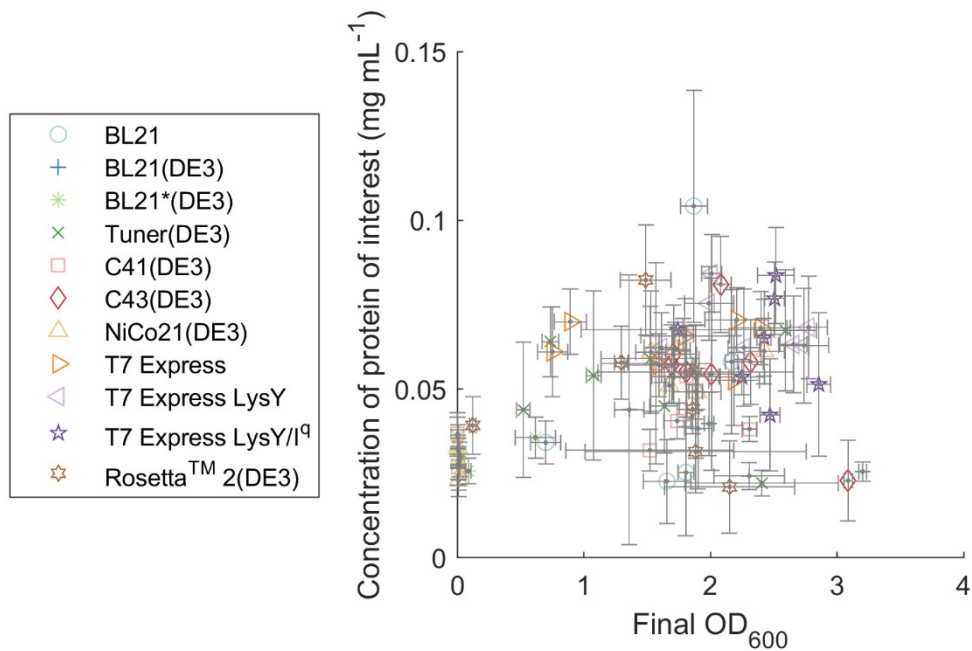


Figure S30. Final OD₆₀₀ vs. concentration of Catcher, determined by dot blot assay, sorted by strain. Error bars reflect the standard error across successfully transformed samples.

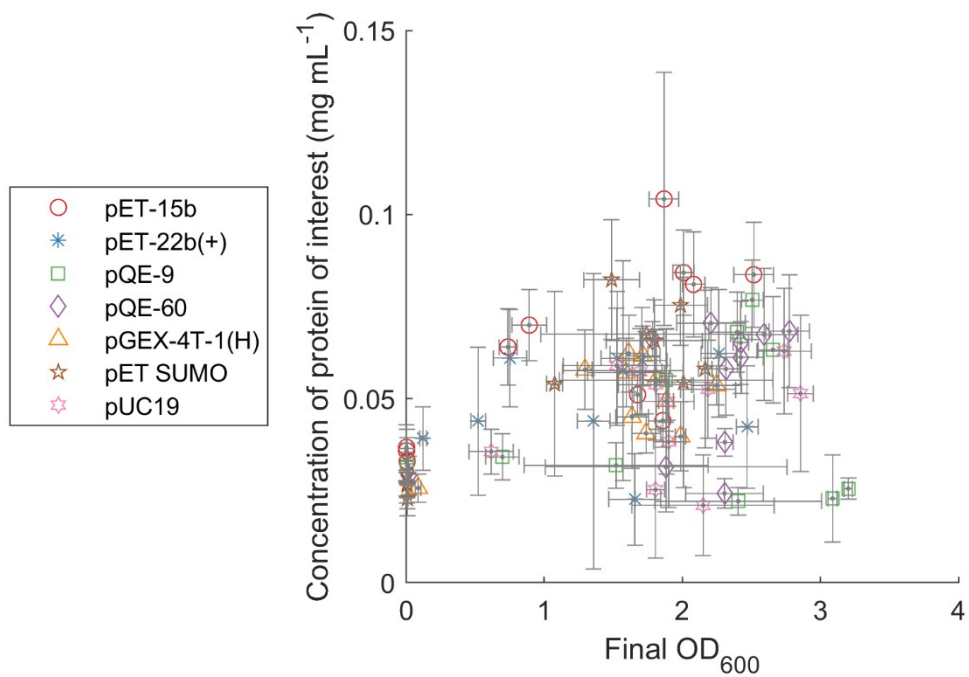


Figure S31. Final OD₆₀₀ vs. concentration of Catcher, determined by dot blot assay, sorted by vector. Error bars reflect the standard error across successfully transformed samples.

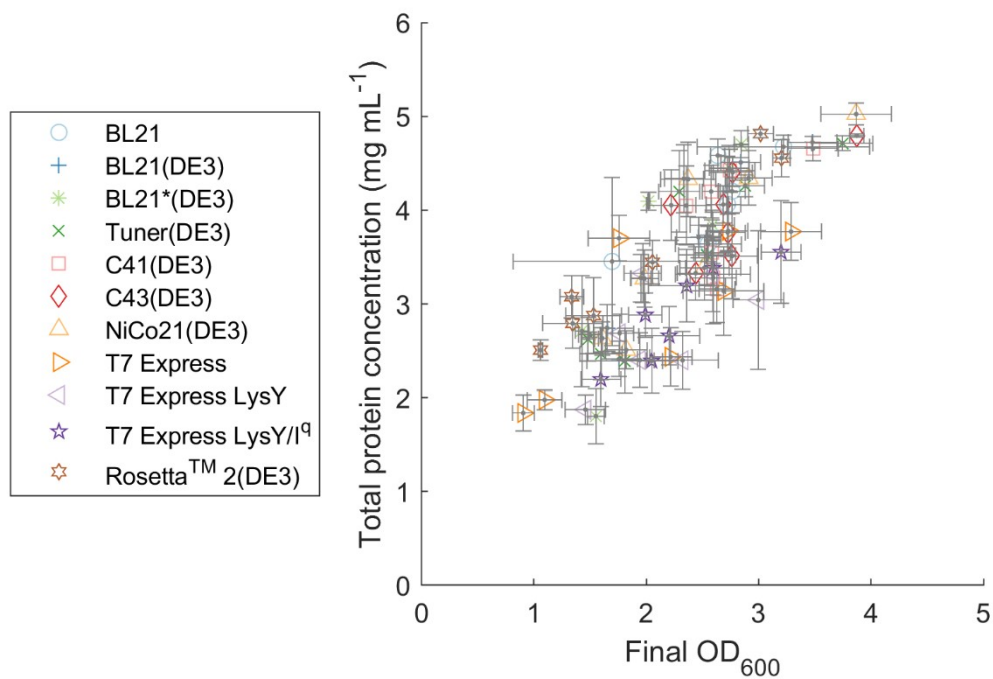


Figure S32. Final OD₆₀₀ vs. total protein concentration for CC43 determined by Bradford assay, sorted by strain. Error bars reflect the standard error across successfully transformed samples.

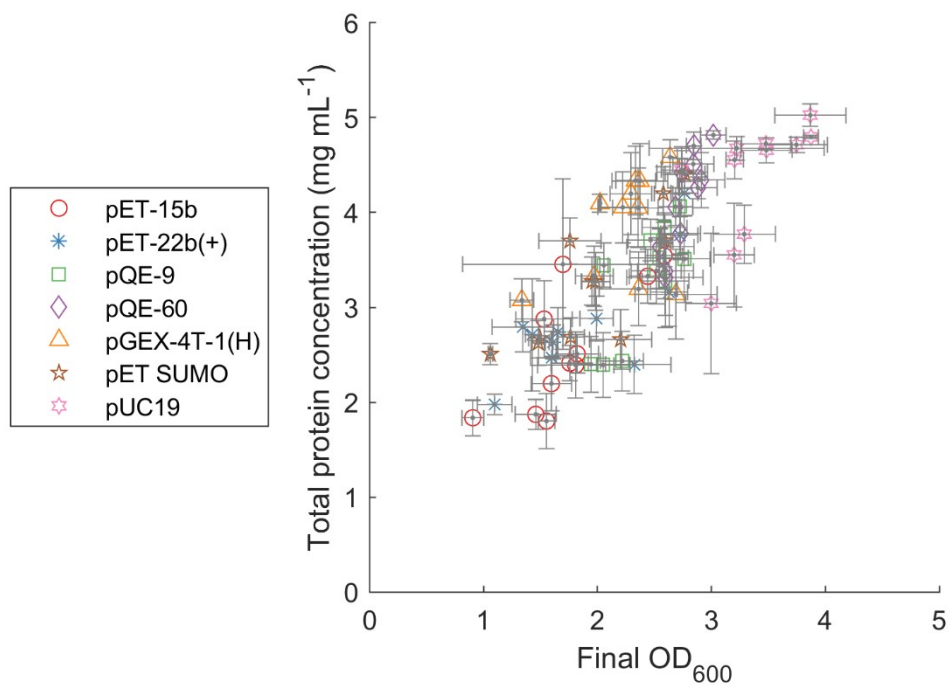


Figure S33. Final OD₆₀₀ vs. total protein concentration for CC43 determined by Bradford assay, sorted by vector. Error bars reflect the standard error across successfully transformed samples.

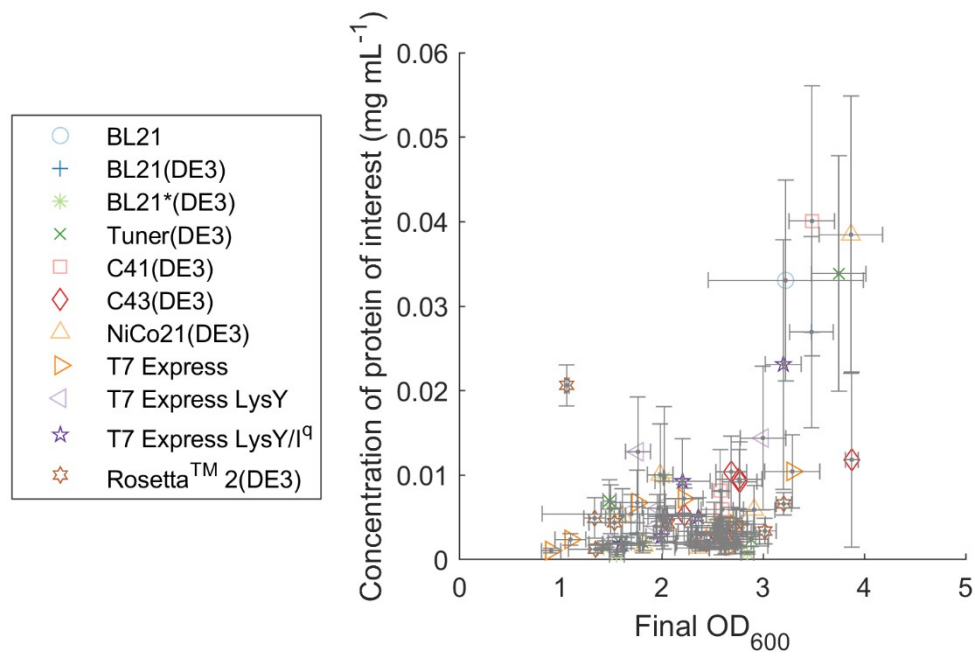


Figure S34. Final OD₆₀₀ vs. concentration of CC43, determined by dot blot assay, sorted by strain. Error bars reflect the standard error across successfully transformed samples.

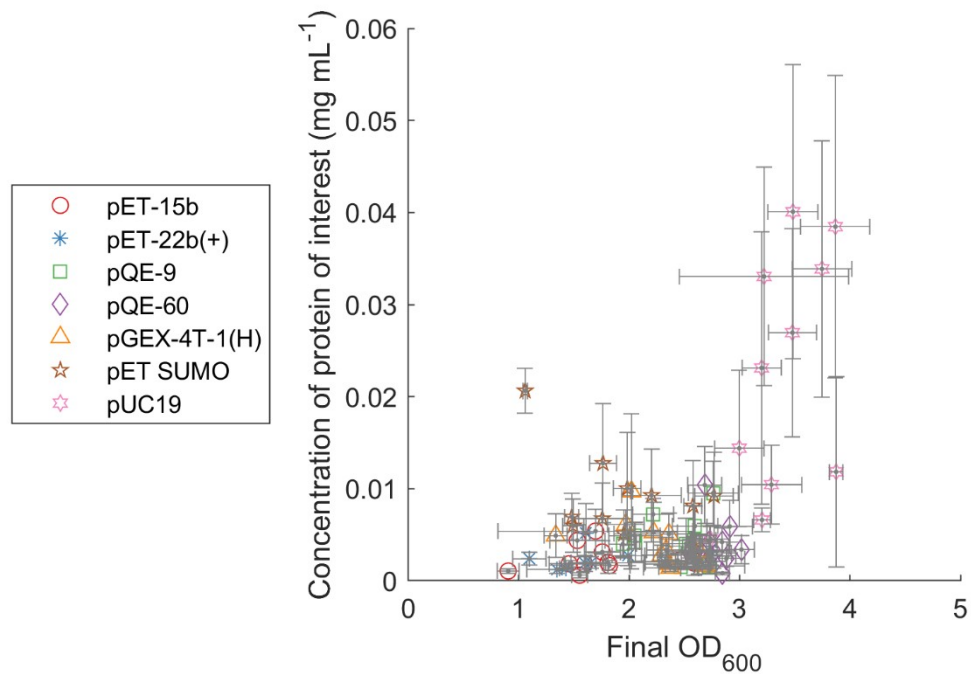


Figure S35. Final OD₆₀₀ vs. concentration of CC43, determined by dot blot assay, sorted by vector. Error bars reflect the standard error across successfully transformed samples.

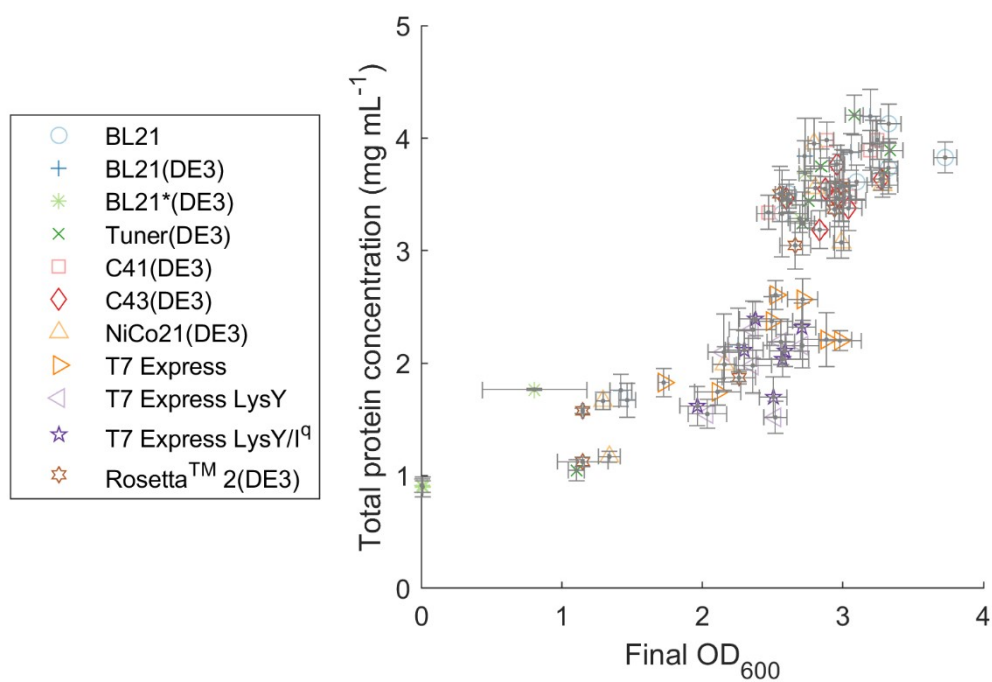


Figure S36. Final OD₆₀₀ vs. total protein concentration for E10 determined by Bradford assay, sorted by strain. Error bars reflect the standard error across successfully transformed samples.

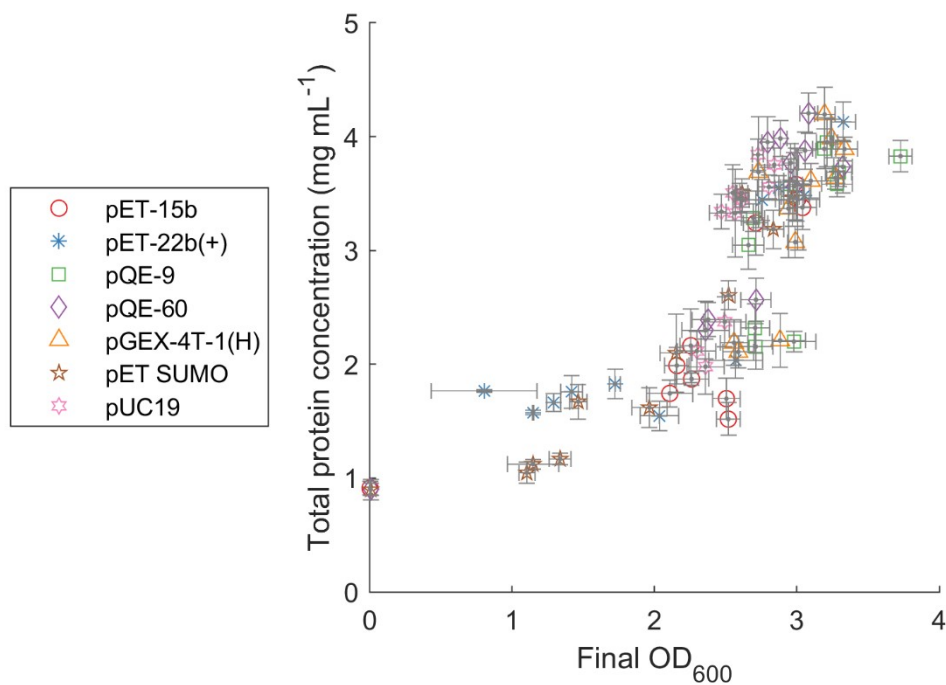


Figure S37. Final OD₆₀₀ vs. total protein concentration for E10 determined by Bradford assay, sorted by vector. Error bars reflect the standard error across successfully transformed samples.

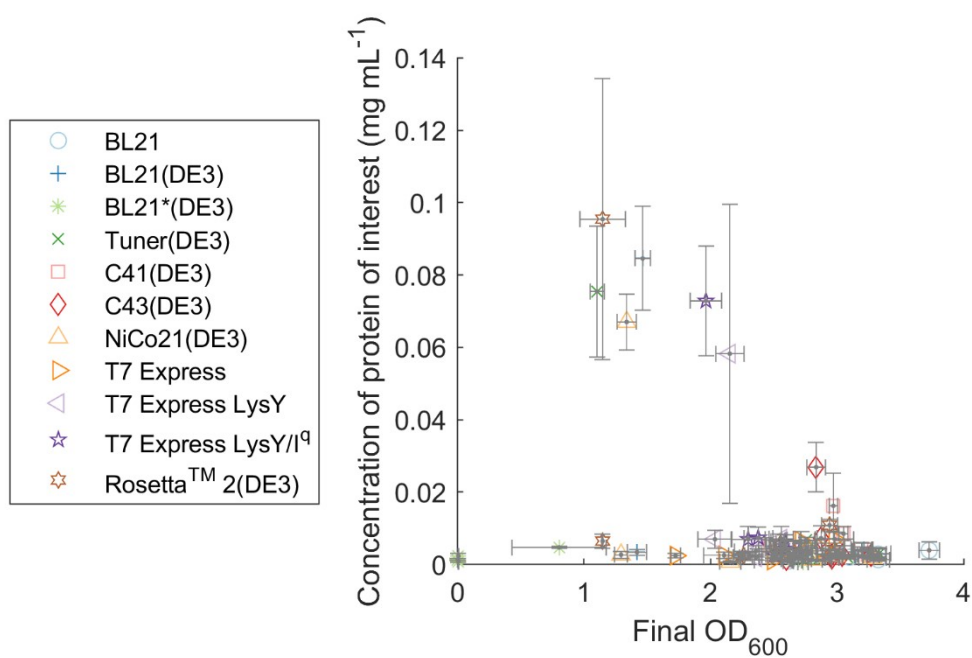


Figure S38. Final OD₆₀₀ vs. concentration of E10, determined by dot blot assay, sorted by strain. Error bars reflect the standard error across successfully transformed samples.

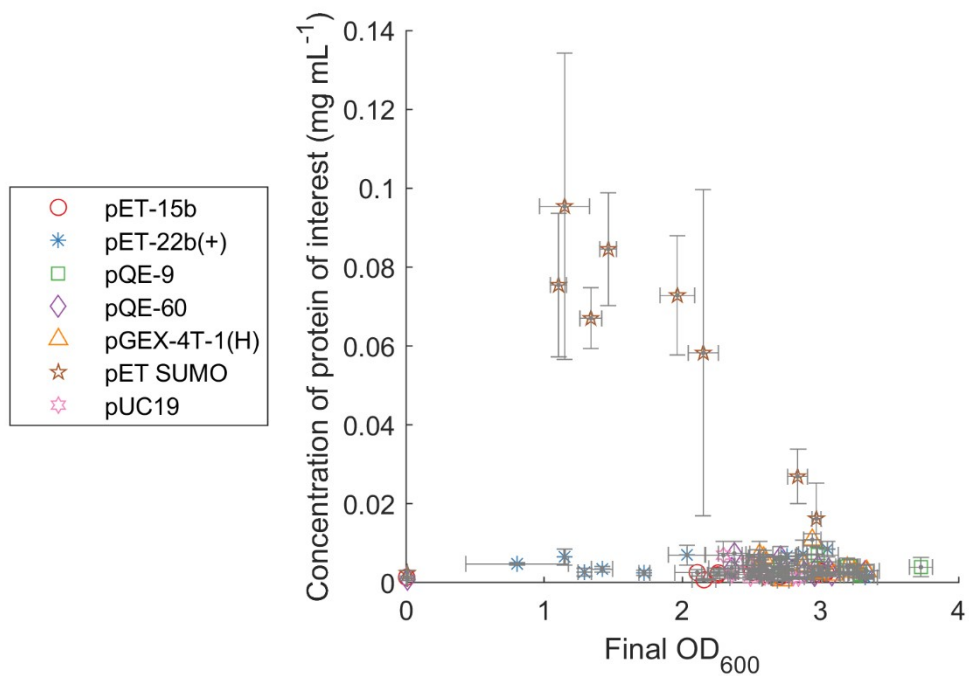


Figure S39. Final OD₆₀₀ vs. concentration of E10, determined by dot blot assay, sorted by vector. Error bars reflect the standard error across successfully transformed samples.

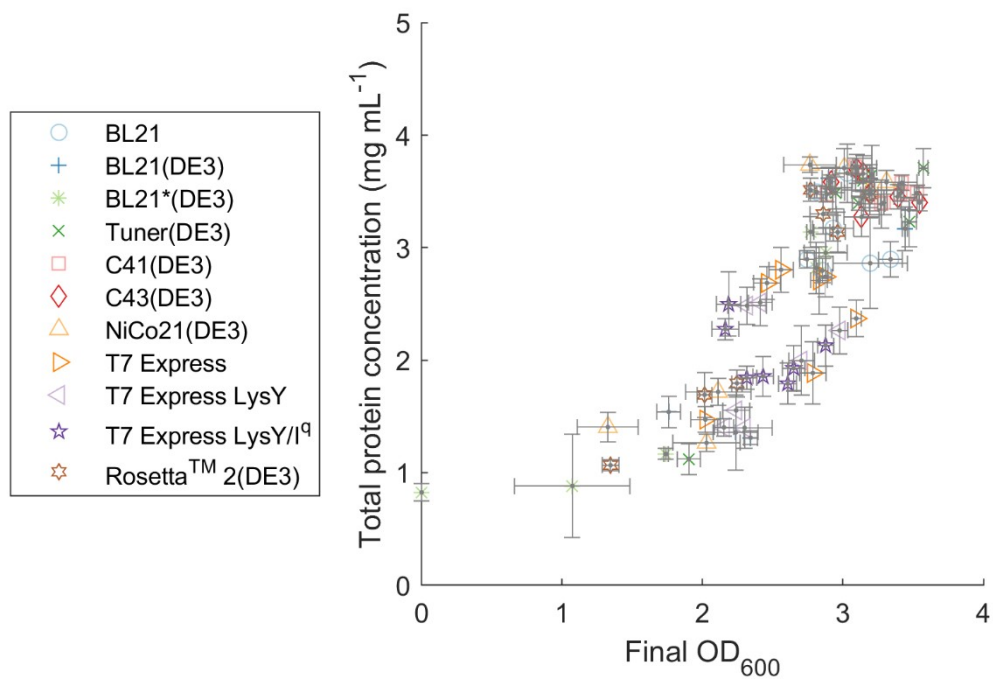


Figure S40. Final OD₆₀₀ vs. total protein concentration for E20 determined by Bradford assay, sorted by strain. Error bars reflect the standard error across successfully transformed samples.

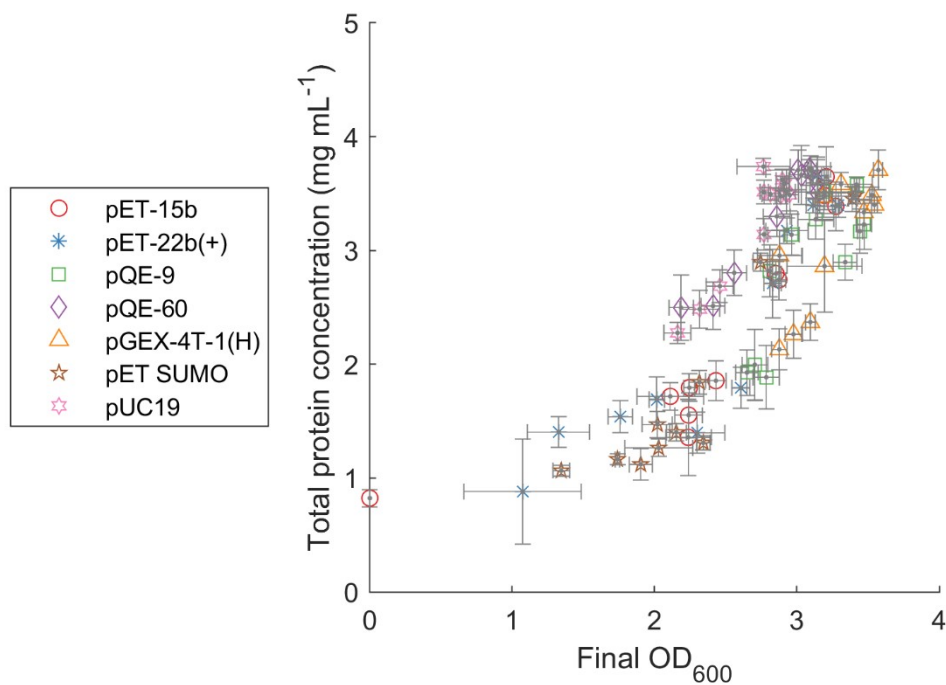


Figure S41. Final OD₆₀₀ vs. total protein concentration for E20 determined by Bradford assay, sorted by vector. Error bars reflect the standard error across successfully transformed samples.

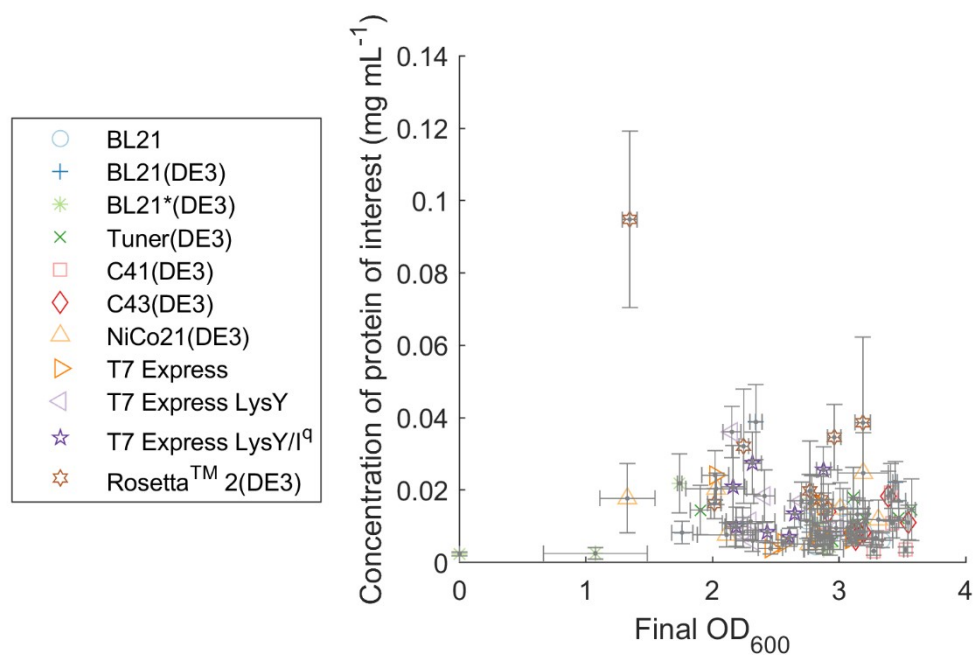


Figure S42. Final OD₆₀₀ vs. concentration of E20, determined by dot blot assay, sorted by strain. Error bars reflect the standard error across successfully transformed samples.

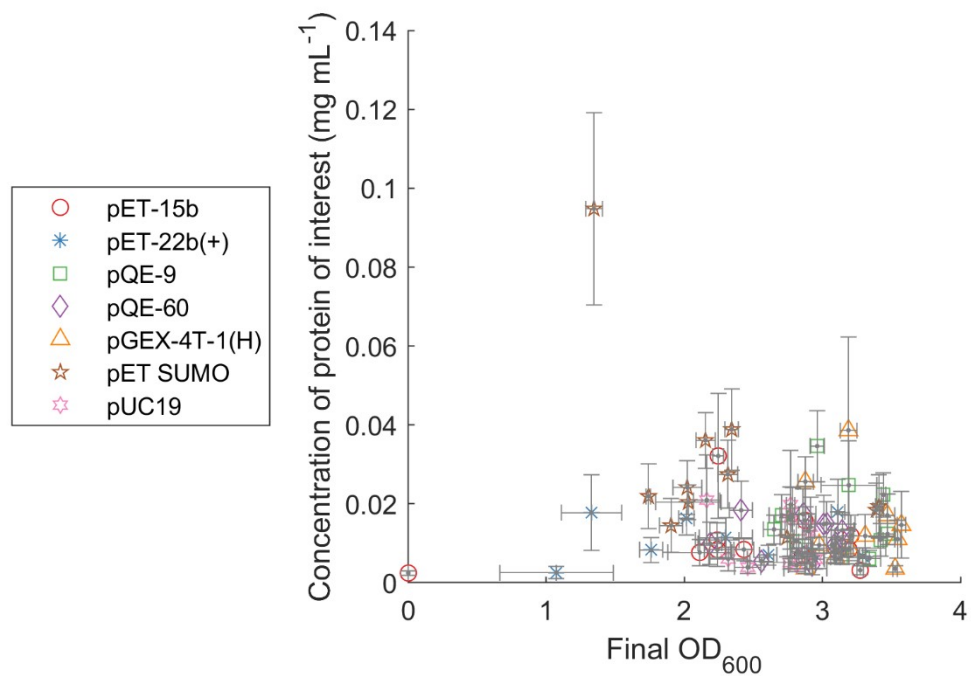


Figure S43. Final OD₆₀₀ vs. concentration of E20, determined by dot blot assay, sorted by vector. Error bars reflect the standard error across successfully transformed samples.

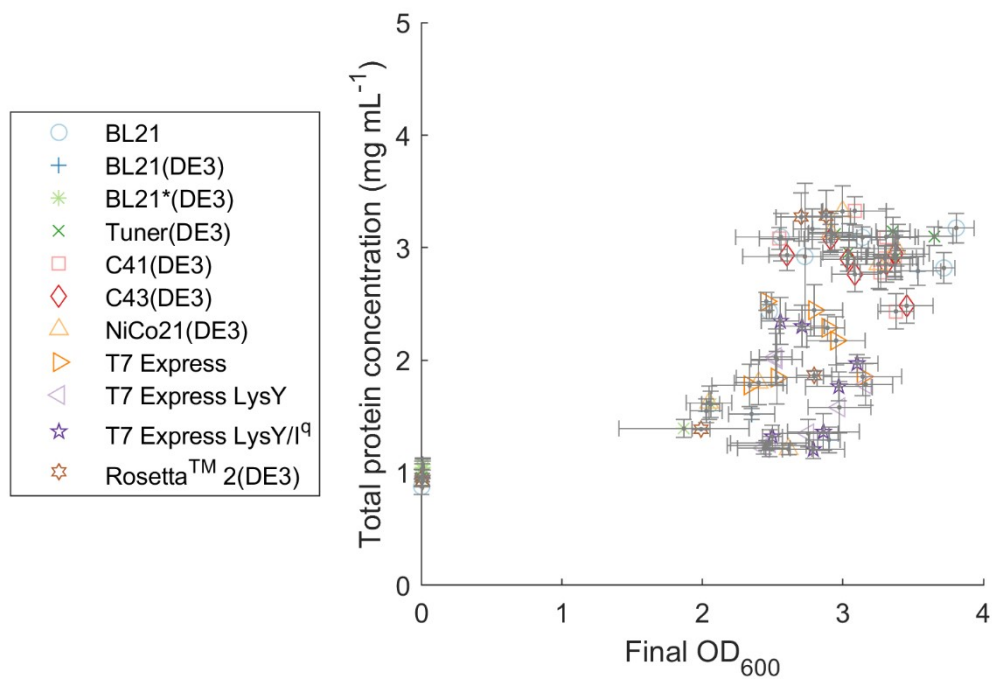


Figure S44. Final OD₆₀₀ vs. total protein concentration for E40 determined by Bradford assay, sorted by strain. Error bars reflect the standard error across successfully transformed samples.

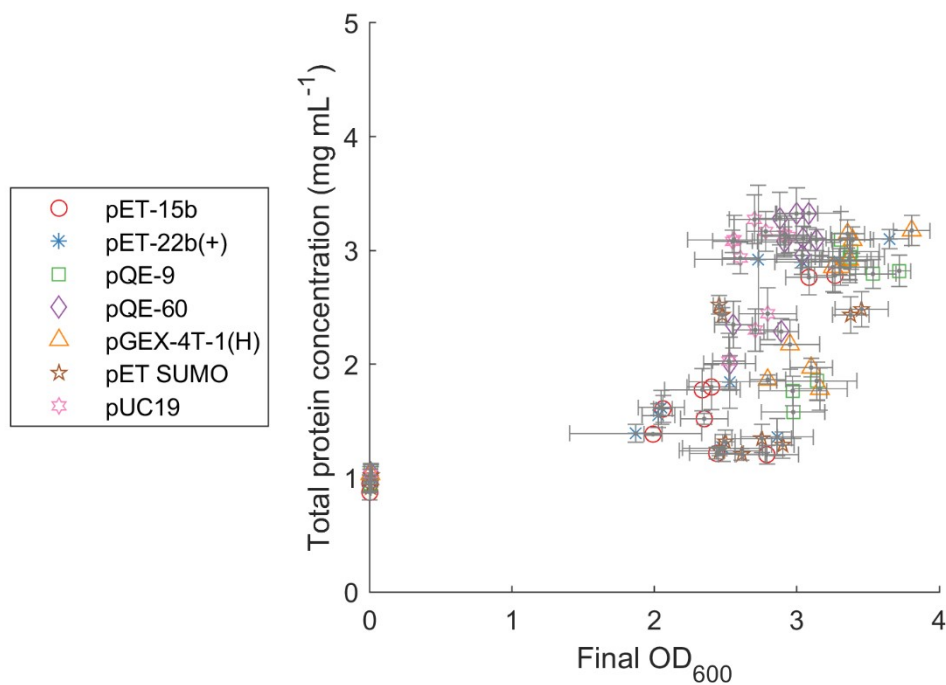


Figure S45. Final OD₆₀₀ vs. total protein concentration for E40 determined by Bradford assay, sorted by vector. Error bars reflect the standard error across successfully transformed samples.

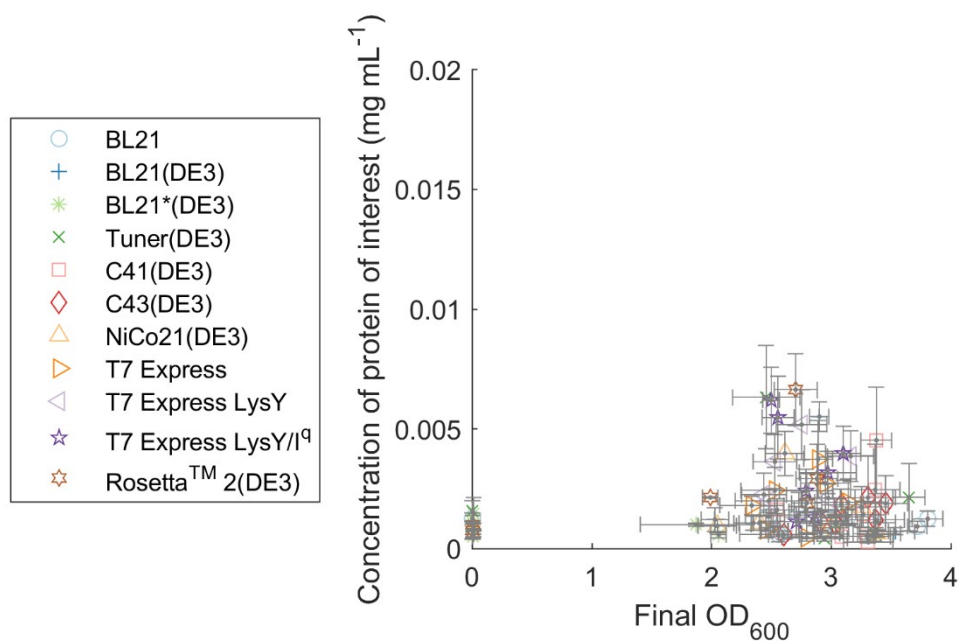


Figure S46. Final OD₆₀₀ vs. concentration of E40, determined by dot blot assay, sorted by strain. Error bars reflect the standard error across successfully transformed samples.

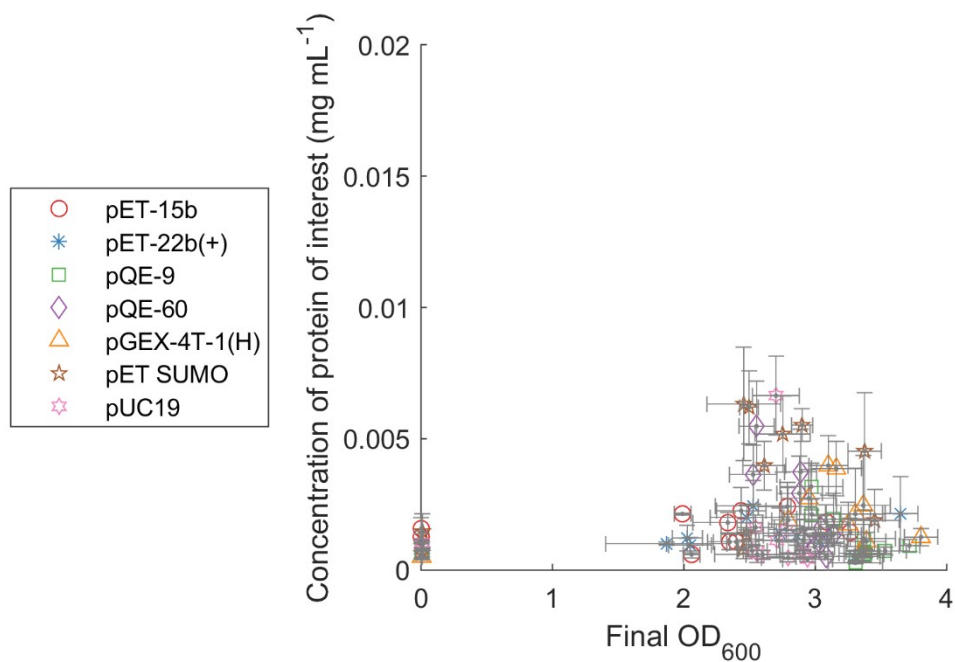


Figure S47. Final OD₆₀₀ vs. concentration of E40, determined by dot blot assay, sorted by vector. Error bars reflect the standard error across successfully transformed samples.

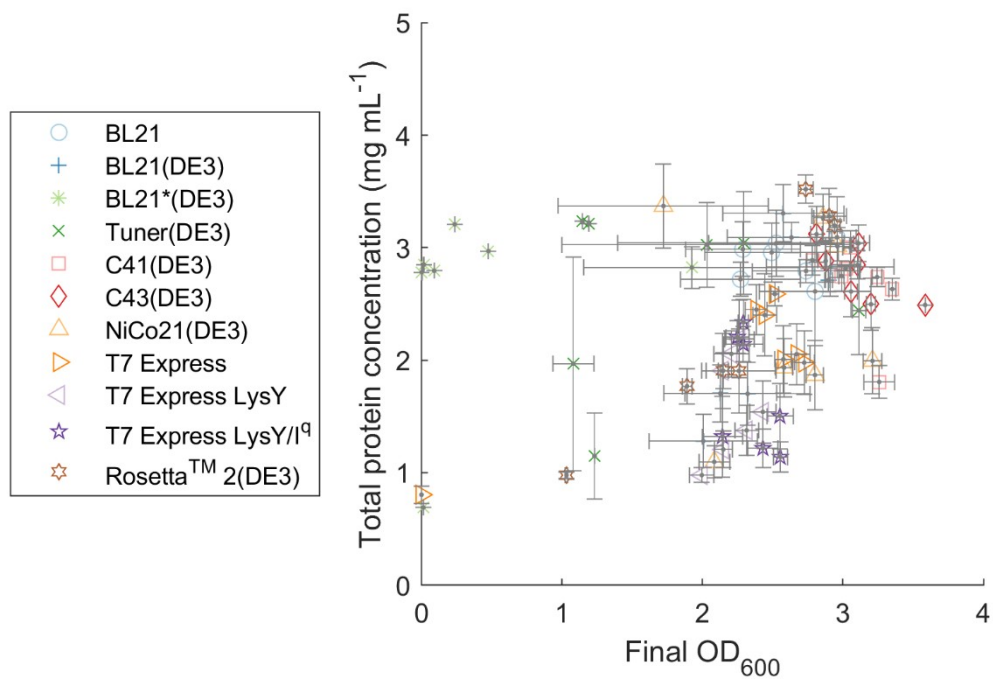


Figure S48. Final OD₆₀₀ vs. total protein concentration for E80 determined by Bradford assay, sorted by strain. Error bars reflect the standard error across successfully transformed samples.

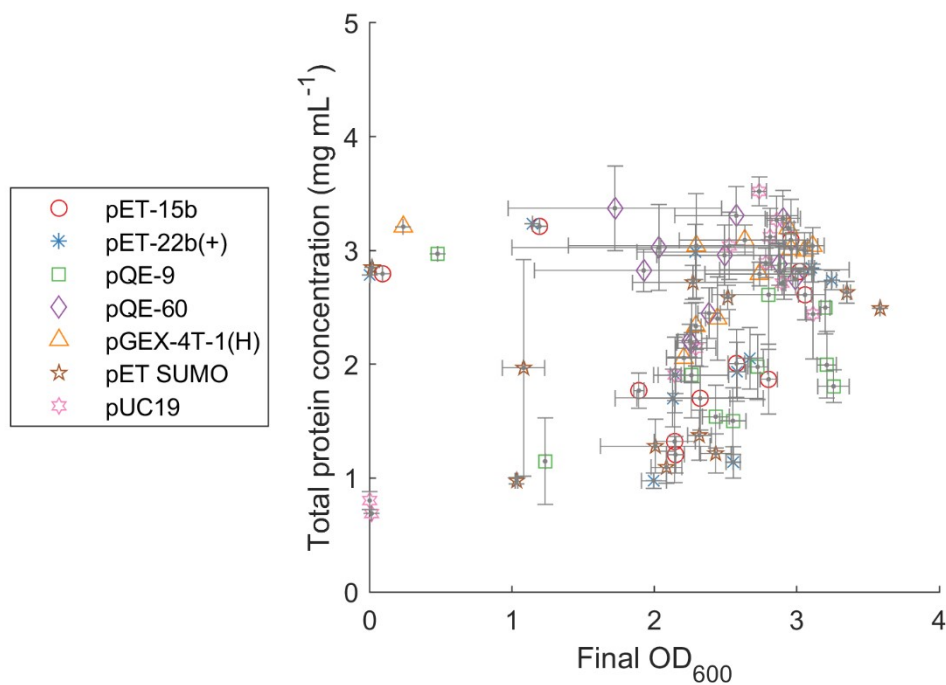


Figure S49. Final OD₆₀₀ vs. total protein concentration for E80 determined by Bradford assay, sorted by vector. Error bars reflect the standard error across successfully transformed samples.

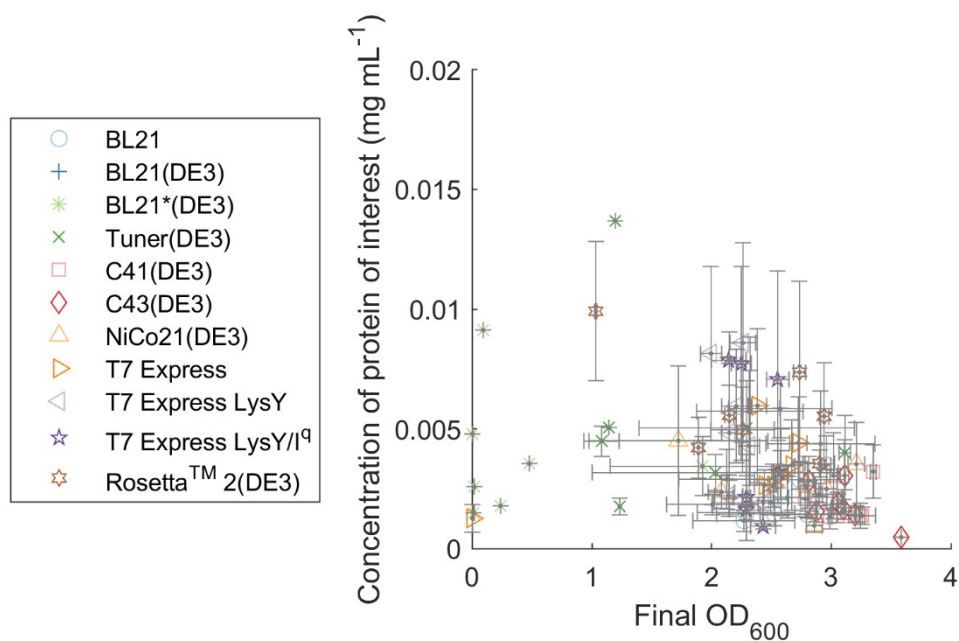


Figure S50. Final OD₆₀₀ vs. concentration of E80, determined by dot blot assay, sorted by strain. Error bars reflect the standard error across successfully transformed samples.

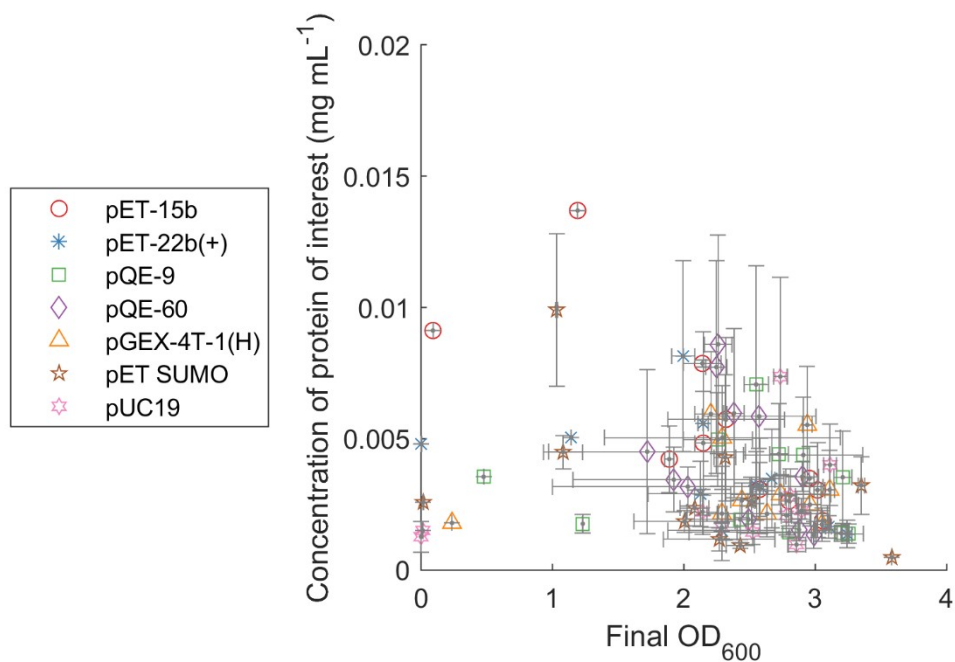


Figure S51. Final OD₆₀₀ vs. concentration of E80, determined by dot blot assay, sorted by vector. Error bars reflect the standard error across successfully transformed samples.

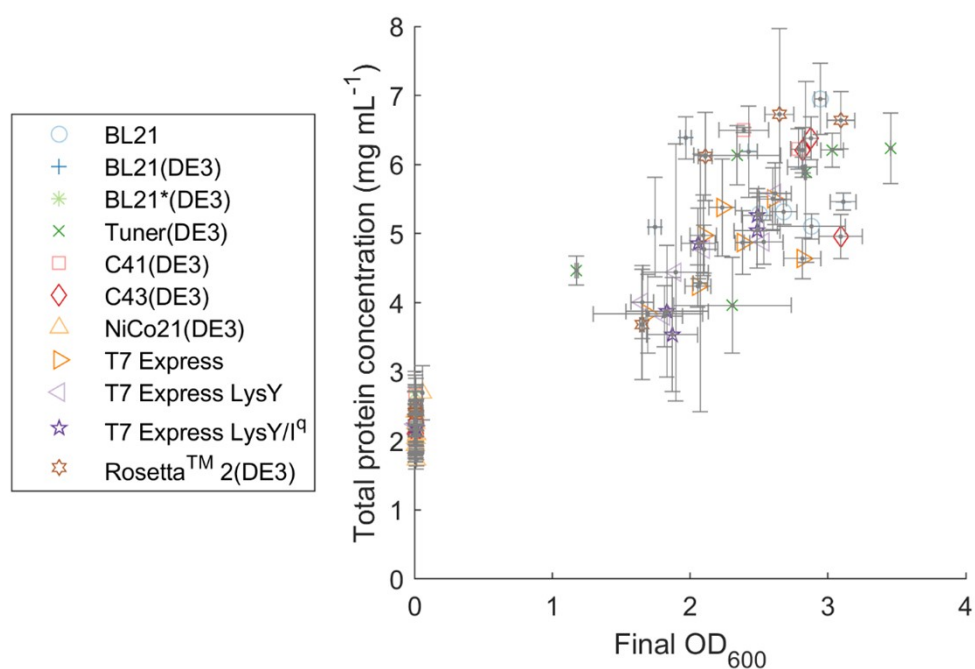


Figure S52. Final OD₆₀₀ vs. total protein concentration for hNup50 determined by Bradford assay, sorted by strain. Error bars reflect the standard error across successfully transformed samples.

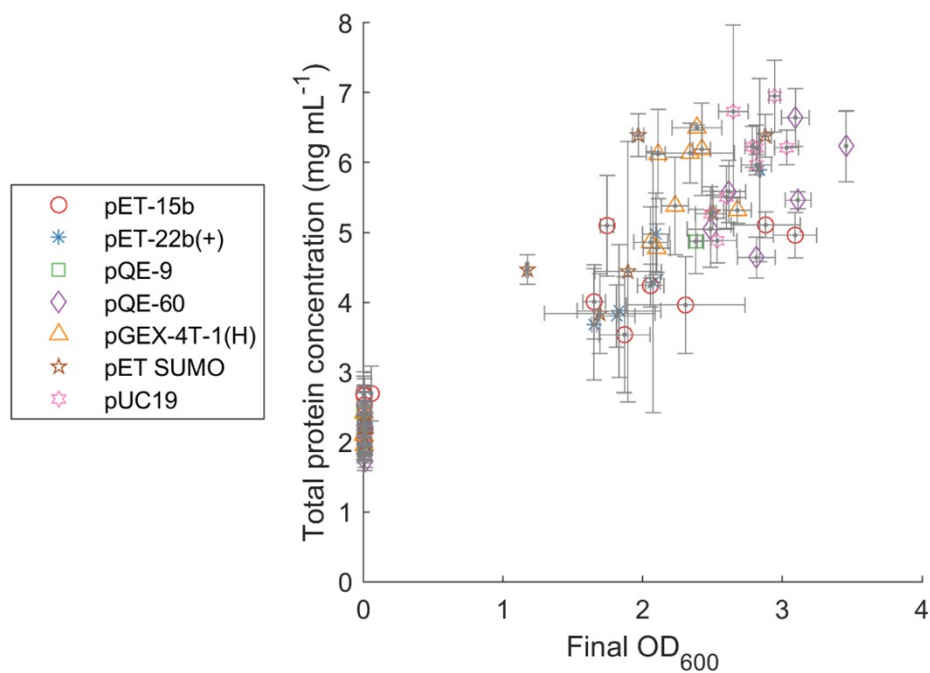


Figure S53. Final OD₆₀₀ vs. total protein concentration for hNup50 determined by Bradford assay, sorted by vector. Error bars reflect the standard error across successfully transformed samples.

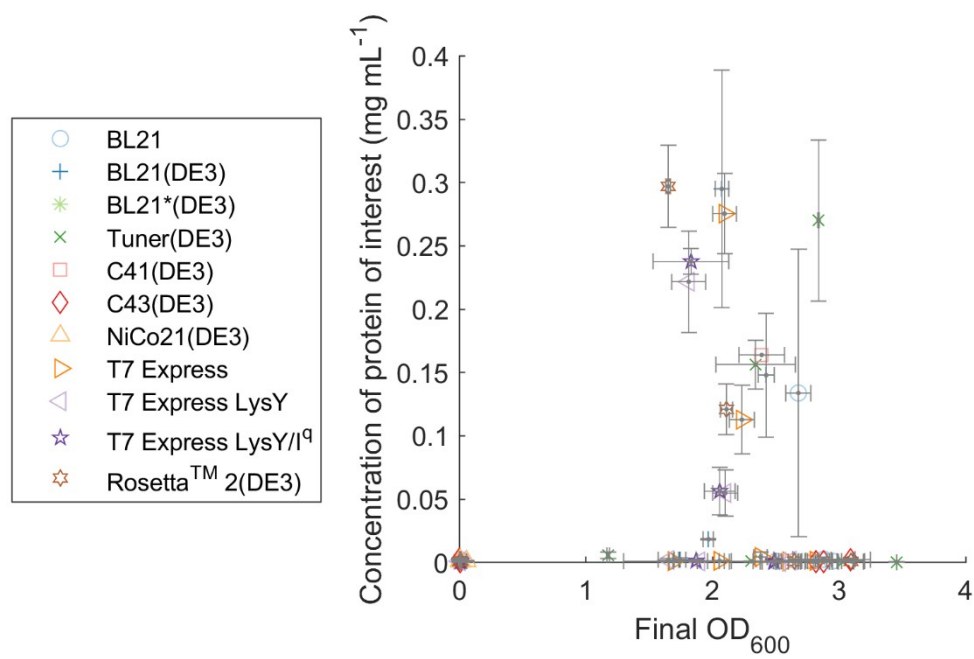


Figure S54. Final OD₆₀₀ vs. concentration of hNup50, determined by dot blot assay, sorted by strain. Error bars reflect the standard error across successfully transformed samples.

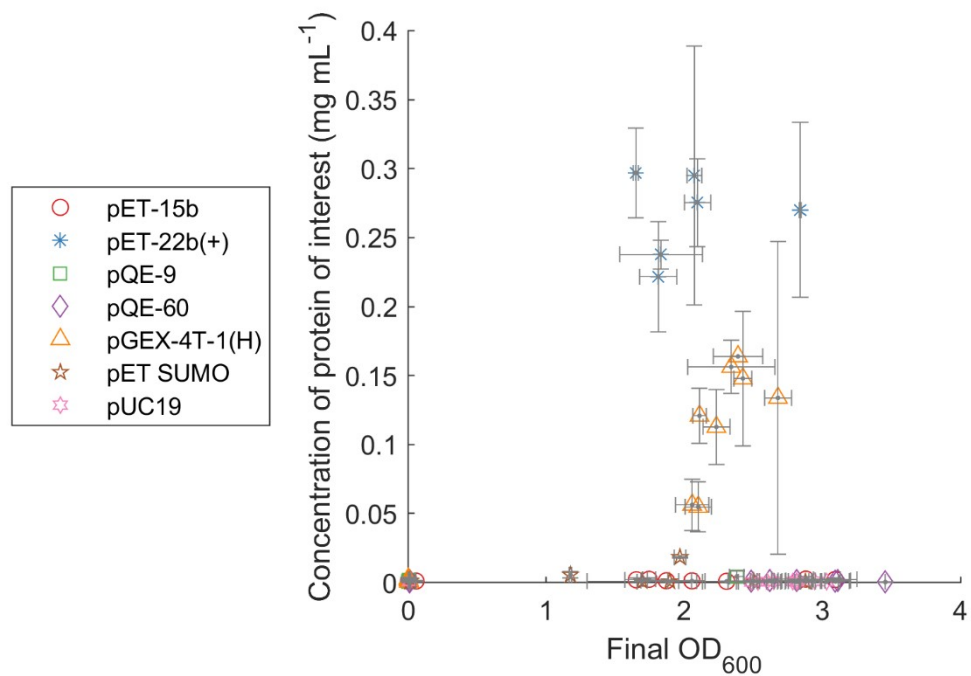


Figure S55. Final OD₆₀₀ vs. concentration of hNup50, determined by dot blot assay, sorted by vector. Error bars reflect the standard error across successfully transformed samples.

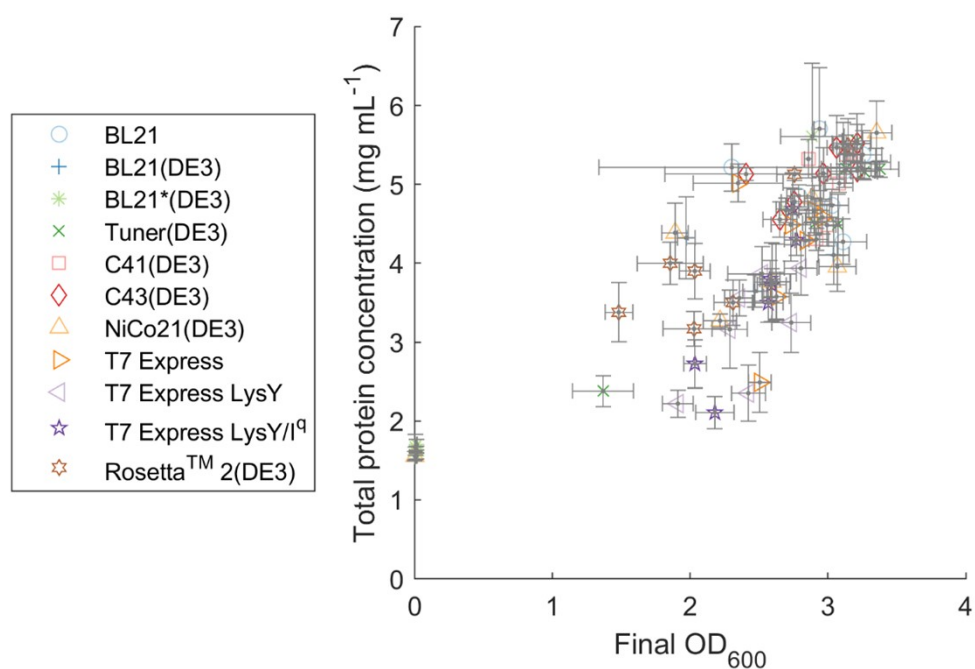


Figure S56. Final OD₆₀₀ vs. total protein concentration for hNup62 determined by Bradford assay, sorted by strain. Error bars reflect the standard error across successfully transformed samples.

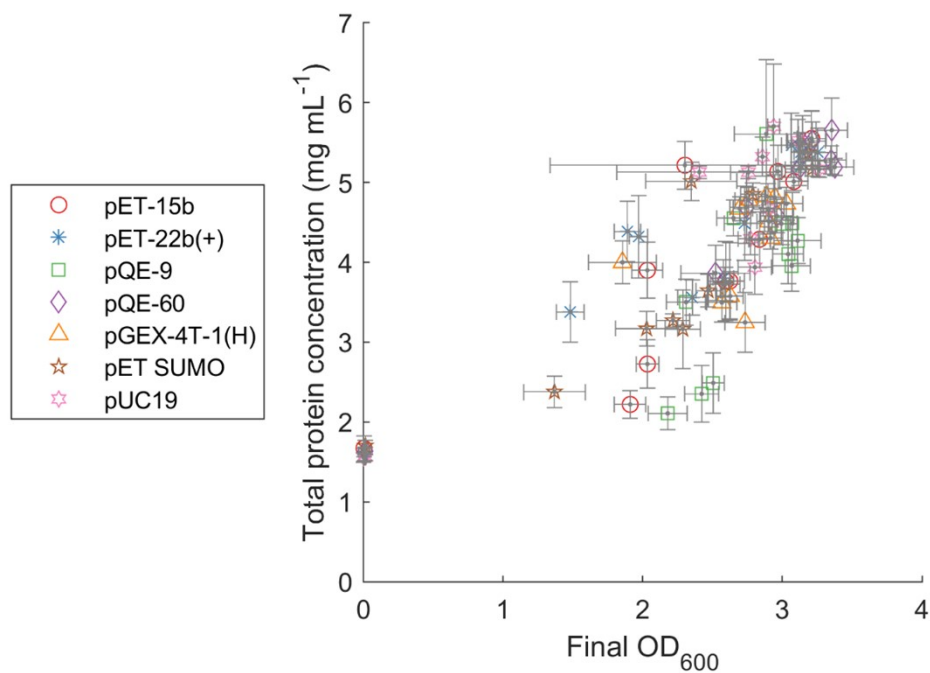


Figure S57. Final OD₆₀₀ vs. total protein concentration for hNup62 determined by Bradford assay, sorted by vector. Error bars reflect the standard error across successfully transformed samples.

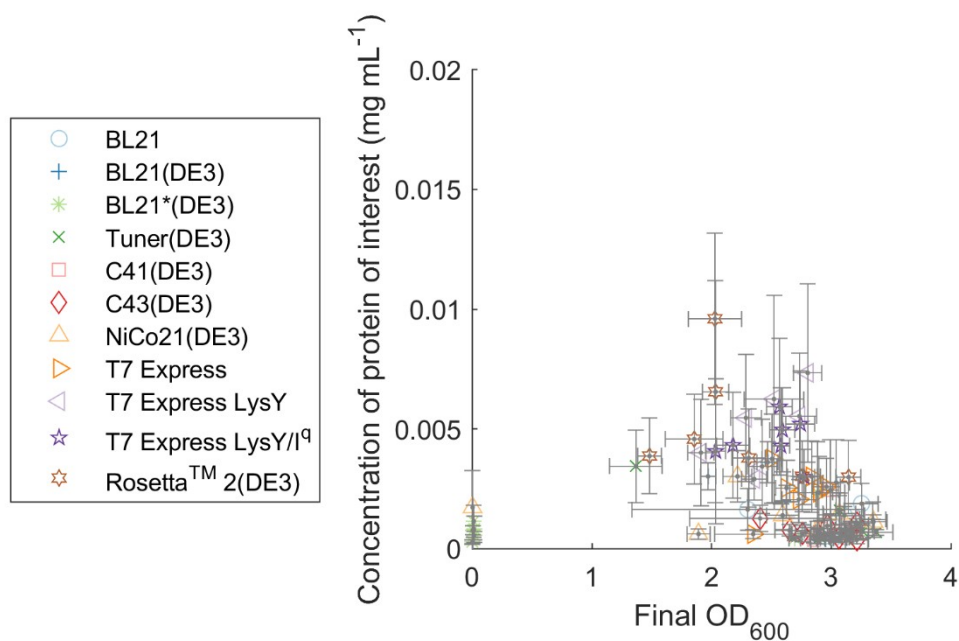


Figure S58. Final OD₆₀₀ vs. concentration of hNup62, determined by dot blot assay, sorted by strain. Error bars reflect the standard error across successfully transformed samples.

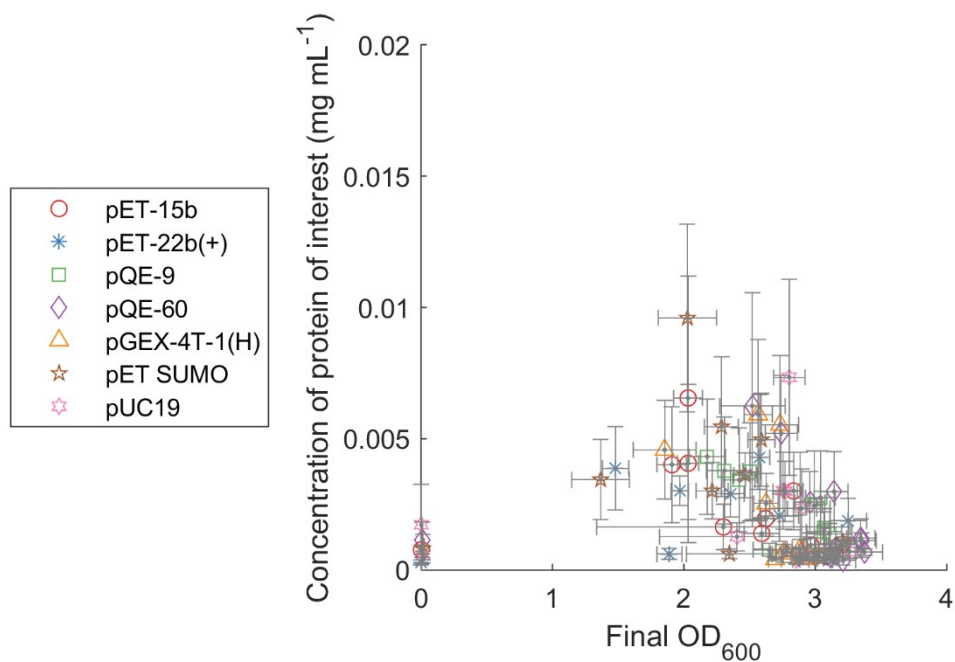


Figure S59. Final OD₆₀₀ vs. concentration of hNup62, determined by dot blot assay, sorted by vector. Error bars reflect the standard error across successfully transformed samples.

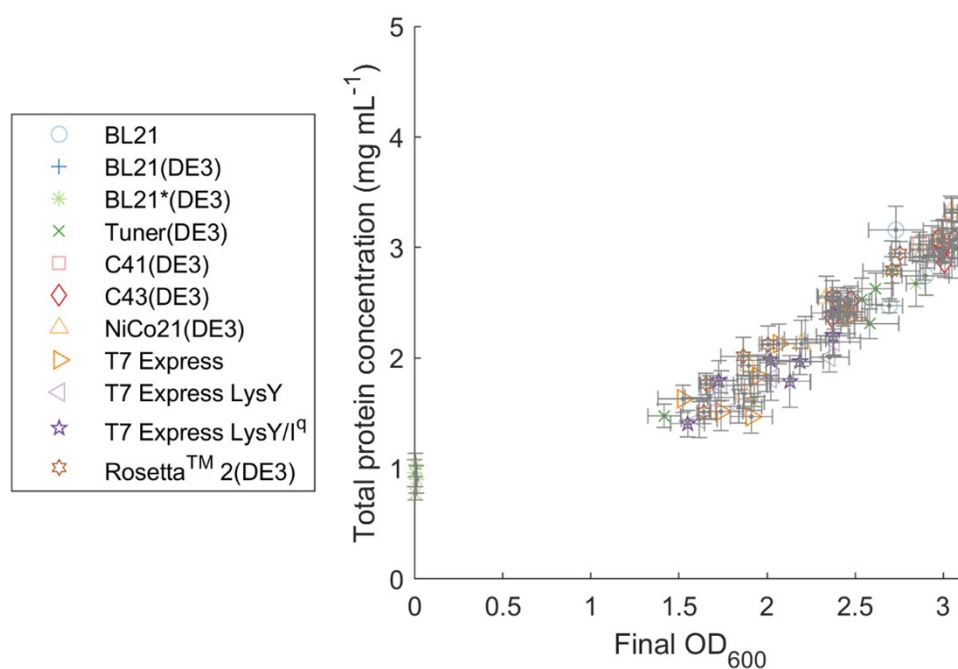


Figure S60. Final OD₆₀₀ vs. total protein concentration for hNup98 determined by Bradford assay, sorted by strain. Error bars reflect the standard error across successfully transformed samples.

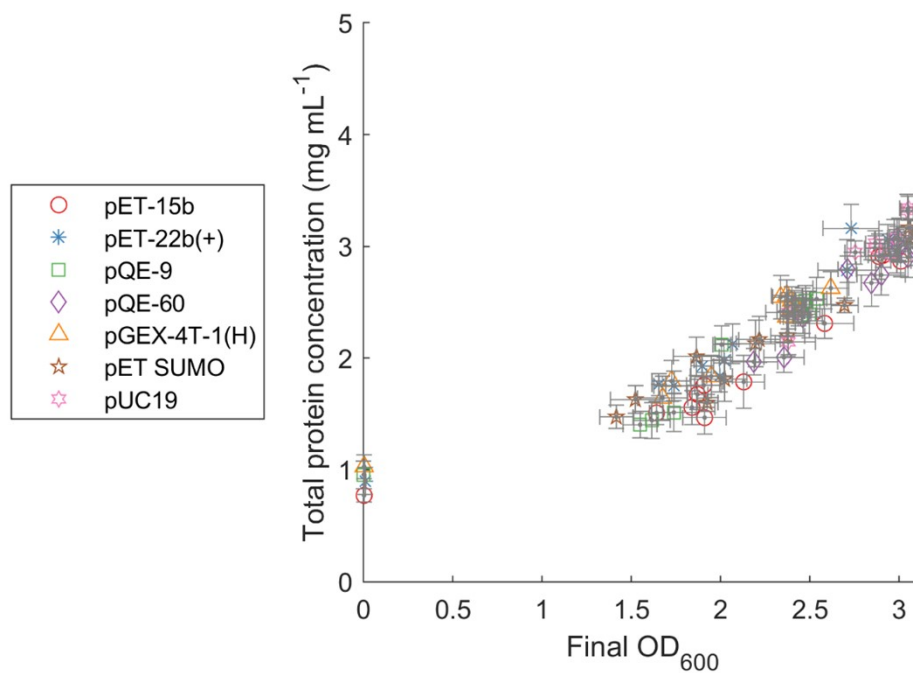


Figure S61. Final OD₆₀₀ vs. total protein concentration for hNup98 determined by Bradford assay, sorted by vector. Error bars reflect the standard error across successfully transformed samples.

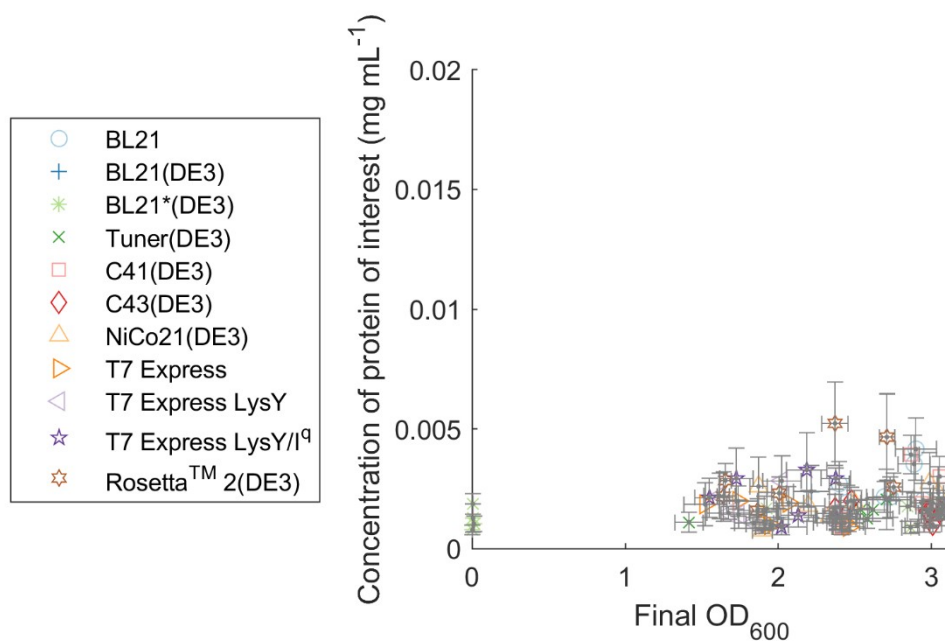


Figure S62. Final OD₆₀₀ vs. concentration of hNup98, determined by dot blot assay, sorted by strain. Error bars reflect the standard error across successfully transformed samples.

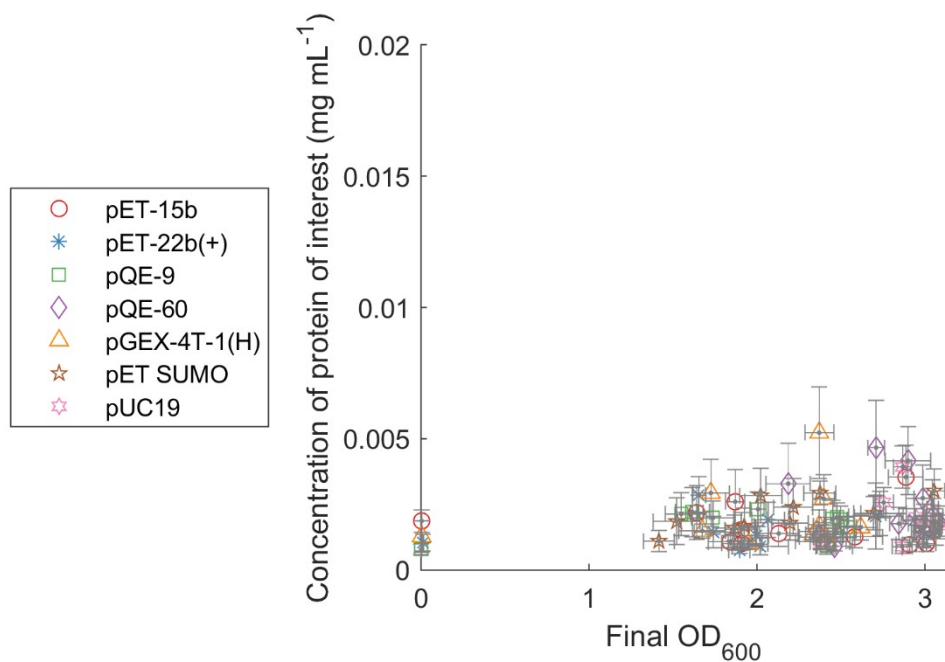


Figure S63. Final OD₆₀₀ vs. concentration of hNup98, determined by dot blot assay, sorted by vector. Error bars reflect the standard error across successfully transformed samples.

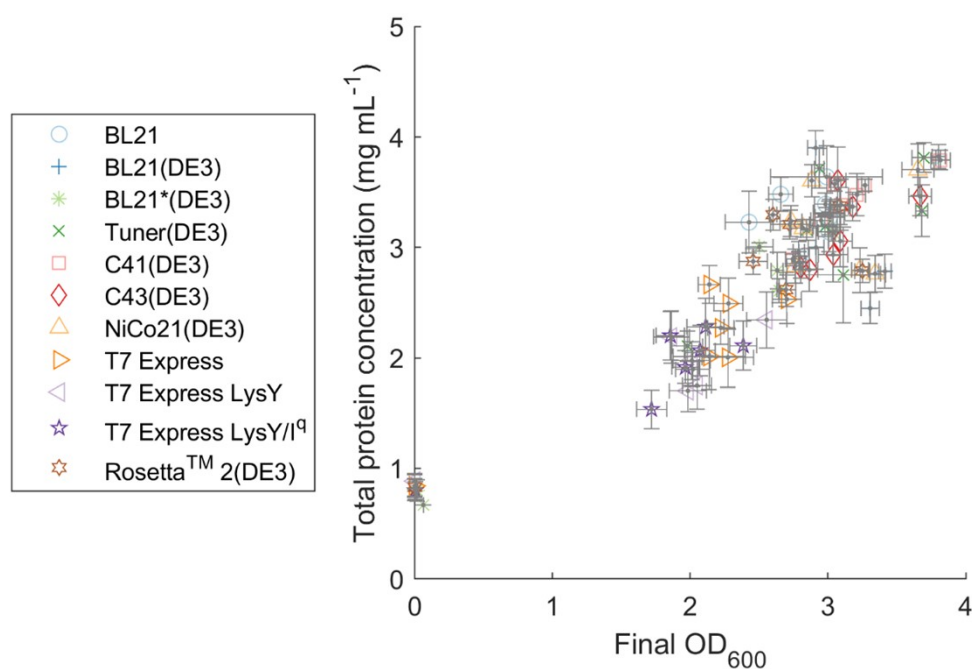


Figure S64. Final OD₆₀₀ vs. total protein concentration for mCherry determined by Bradford assay, sorted by strain. Error bars reflect the standard error across successfully transformed samples.

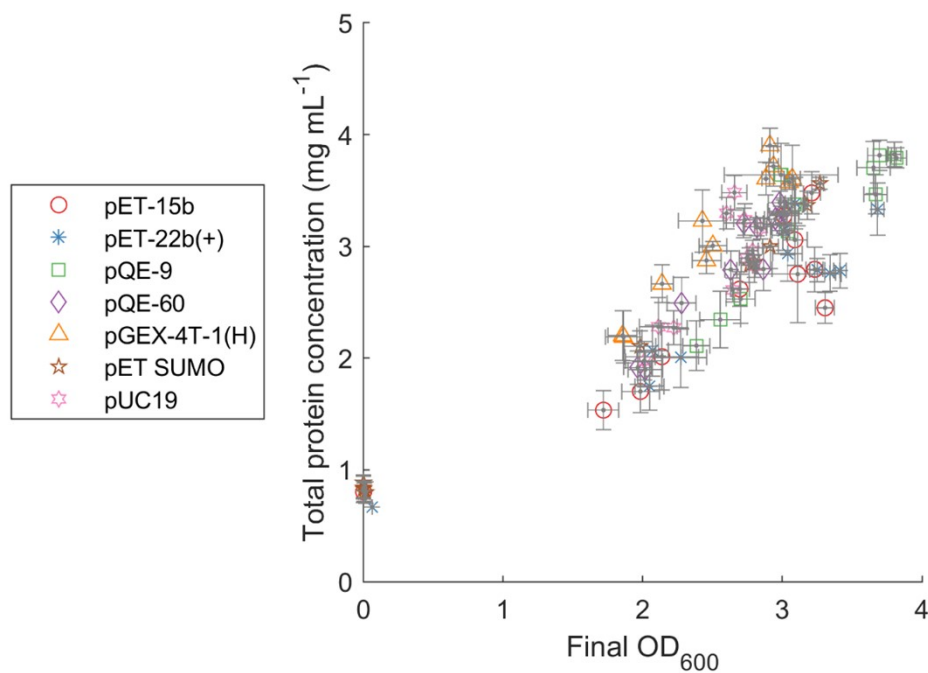


Figure S65. Final OD₆₀₀ vs. total protein concentration for mCherry determined by Bradford assay, sorted by vector. Error bars reflect the standard error across successfully transformed samples.

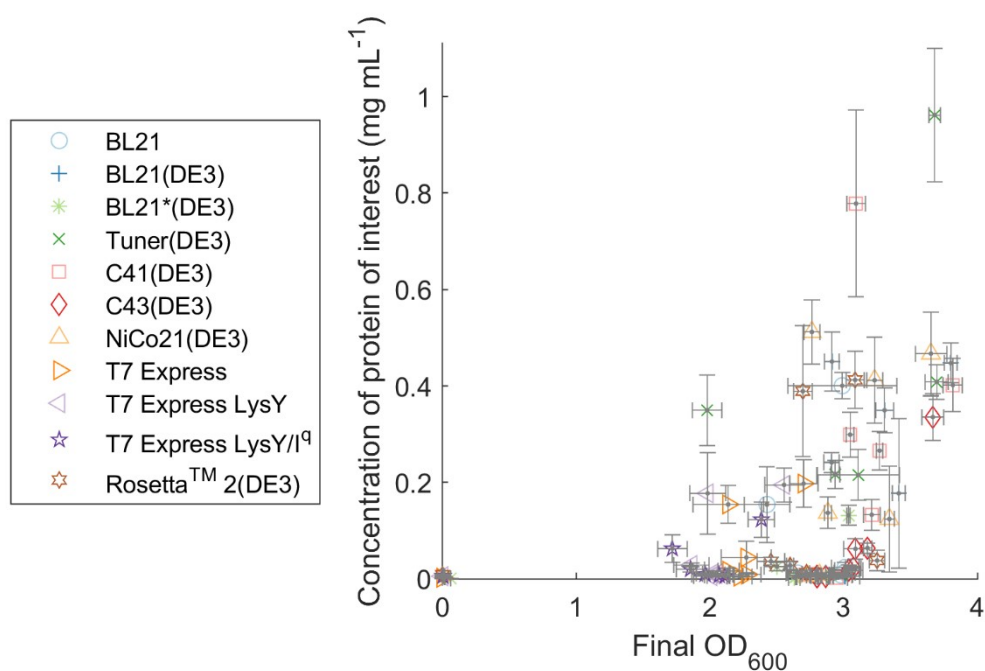


Figure S66. Final OD₆₀₀ vs. concentration of mCherry, determined by dot blot assay, sorted by strain. Error bars reflect the standard error across successfully transformed samples.

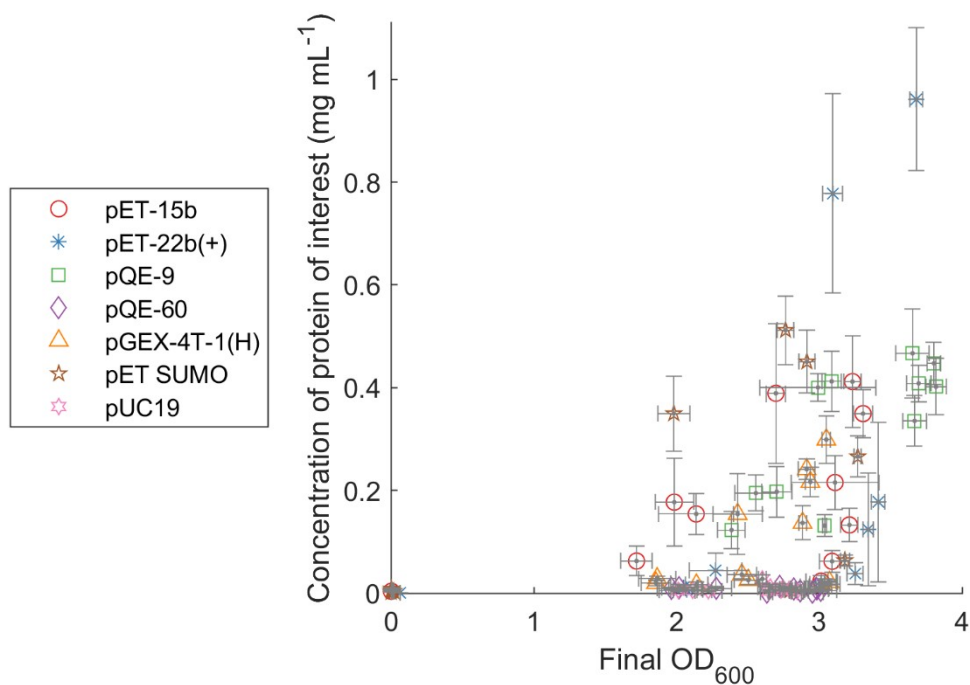


Figure S67. Final OD₆₀₀ vs. concentration of mCherry, determined by dot blot assay, sorted by vector. Error bars reflect the standard error across successfully transformed samples.

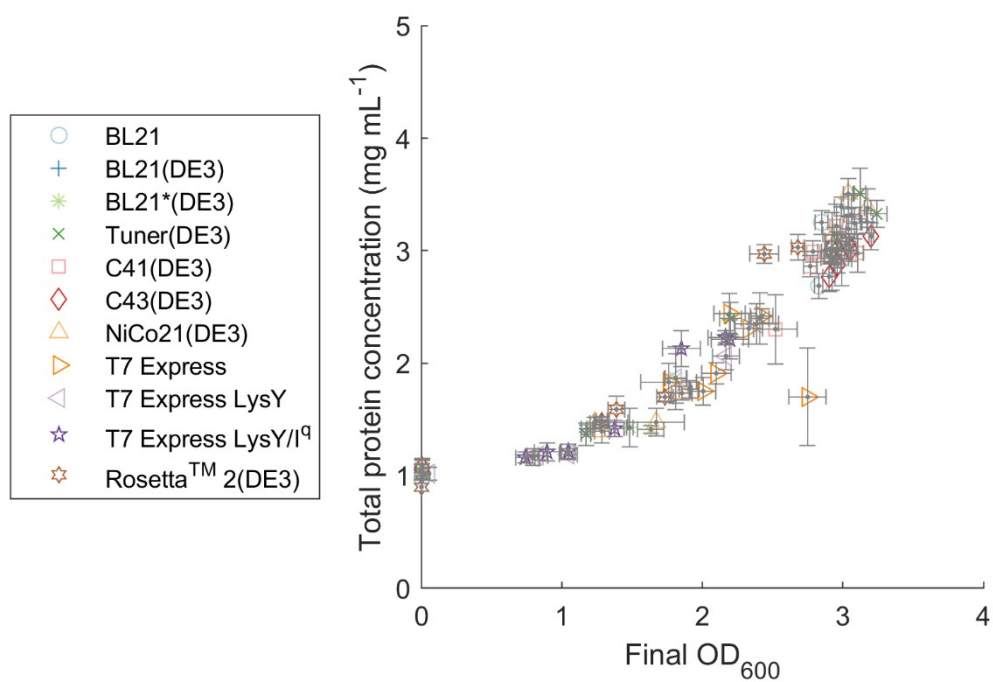


Figure S68. Final OD₆₀₀ vs. total protein concentration for P4 determined by Bradford assay, sorted by strain. Error bars reflect the standard error across successfully transformed samples.

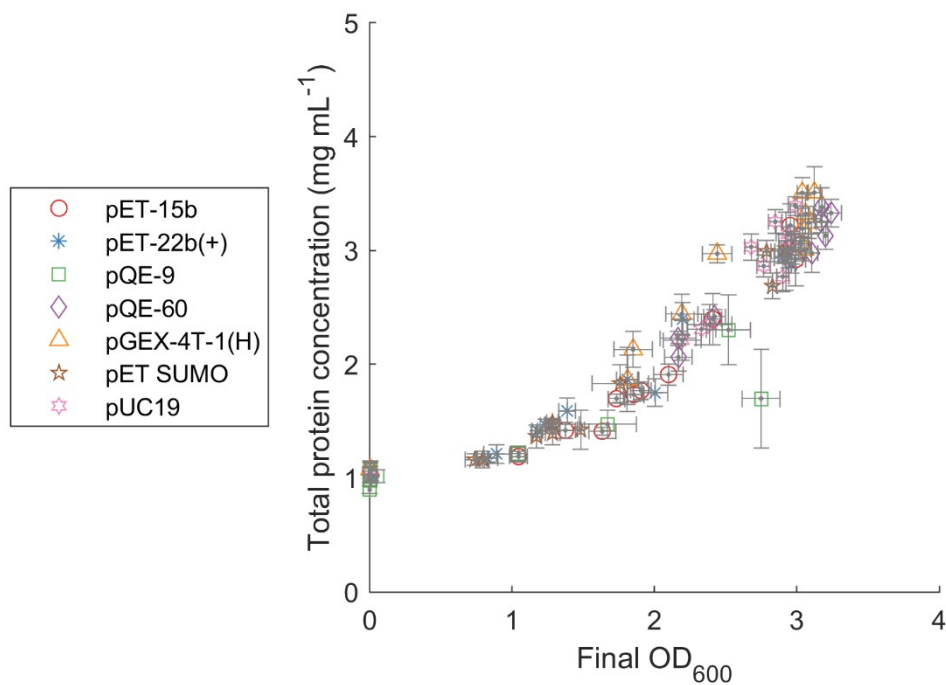


Figure S69. Final OD₆₀₀ vs. total protein concentration for P4 determined by Bradford assay, sorted by vector. Error bars reflect the standard error across successfully transformed samples.

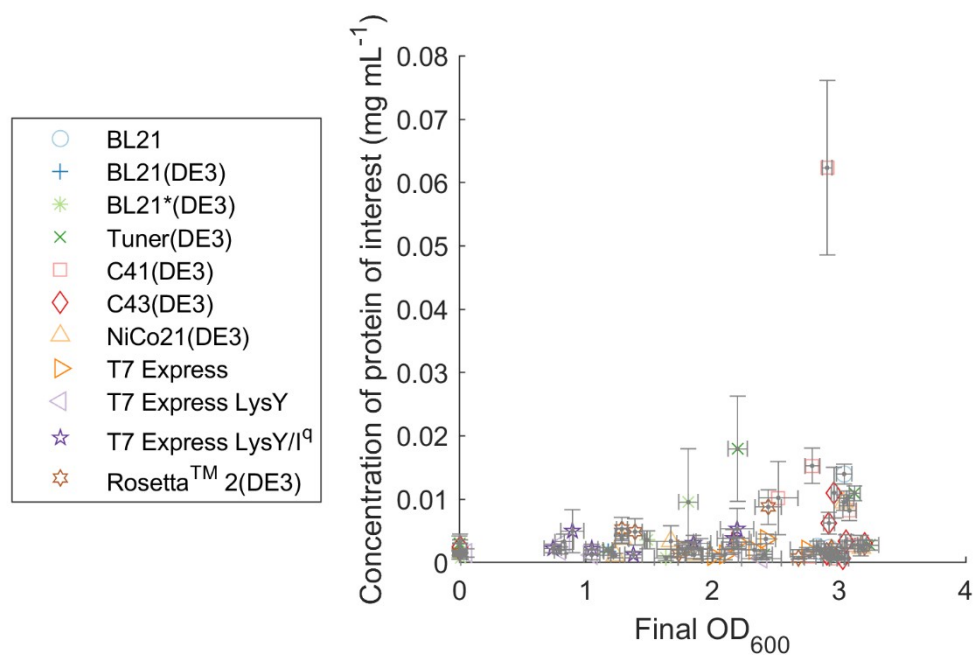


Figure S70. Final OD₆₀₀ vs. concentration of P4, determined by dot blot assay, sorted by strain. Error bars reflect the standard error across successfully transformed samples.

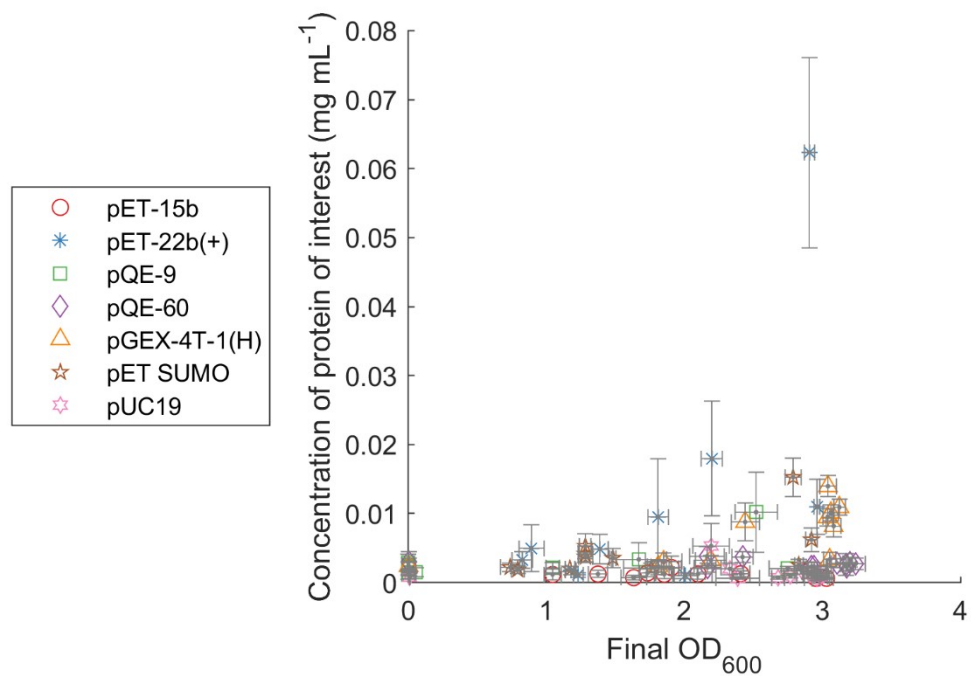


Figure S71. Final OD₆₀₀ vs. concentration of P4, determined by dot blot assay, sorted by vector. Error bars reflect the standard error across successfully transformed samples.

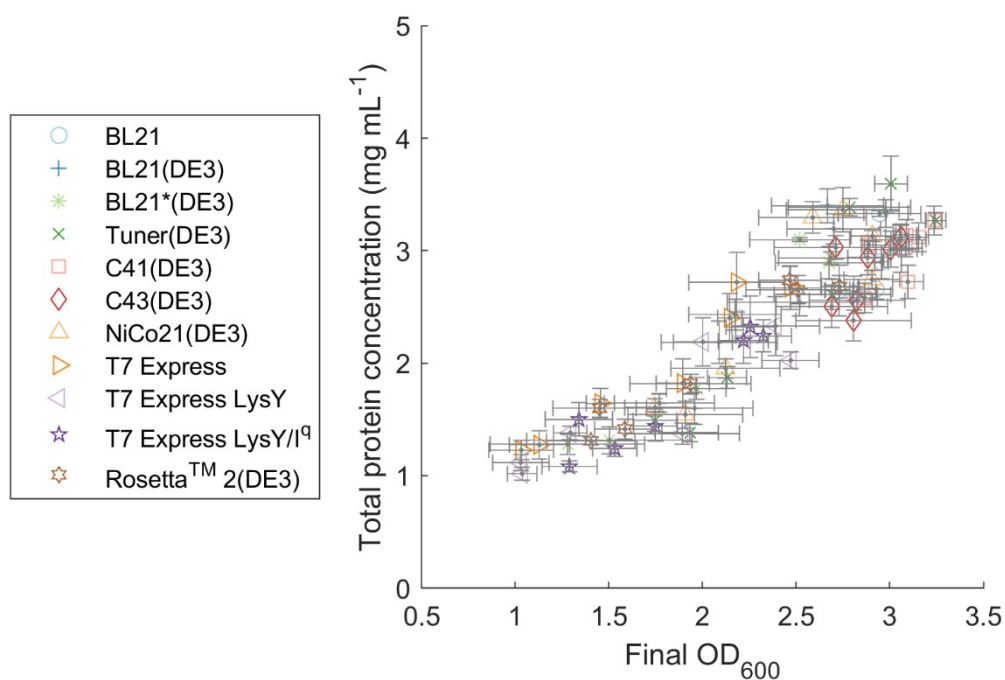


Figure S72. Final OD₆₀₀ vs. total protein concentration for PPxY determined by Bradford assay, sorted by strain. Error bars reflect the standard error across successfully transformed samples.

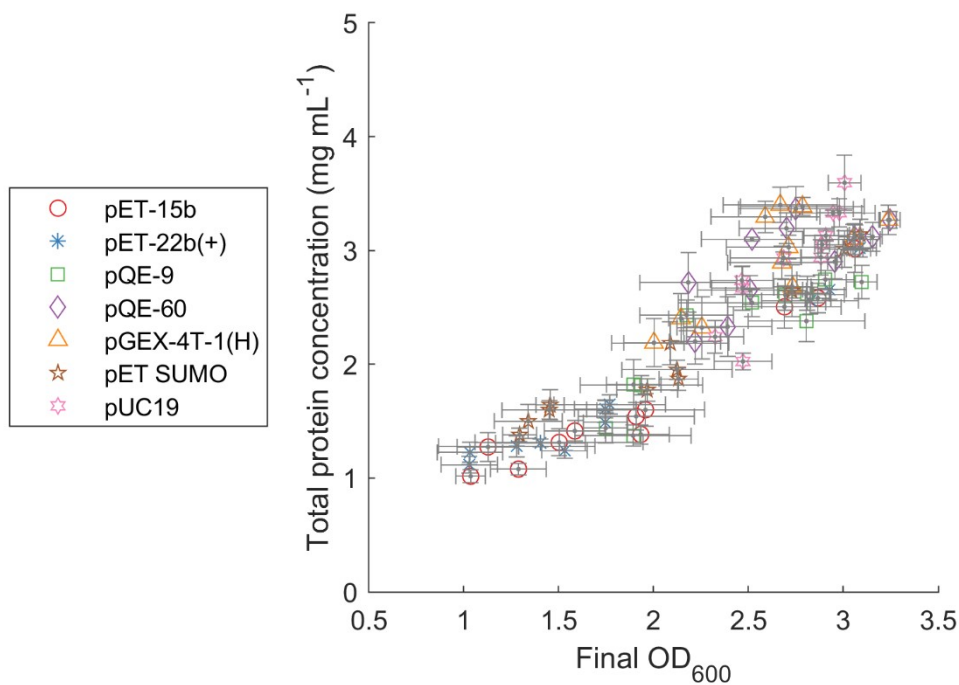


Figure S73. Final OD₆₀₀ vs. total protein concentration for PPxY determined by Bradford assay, sorted by vector. Error bars reflect the standard error across successfully transformed samples.

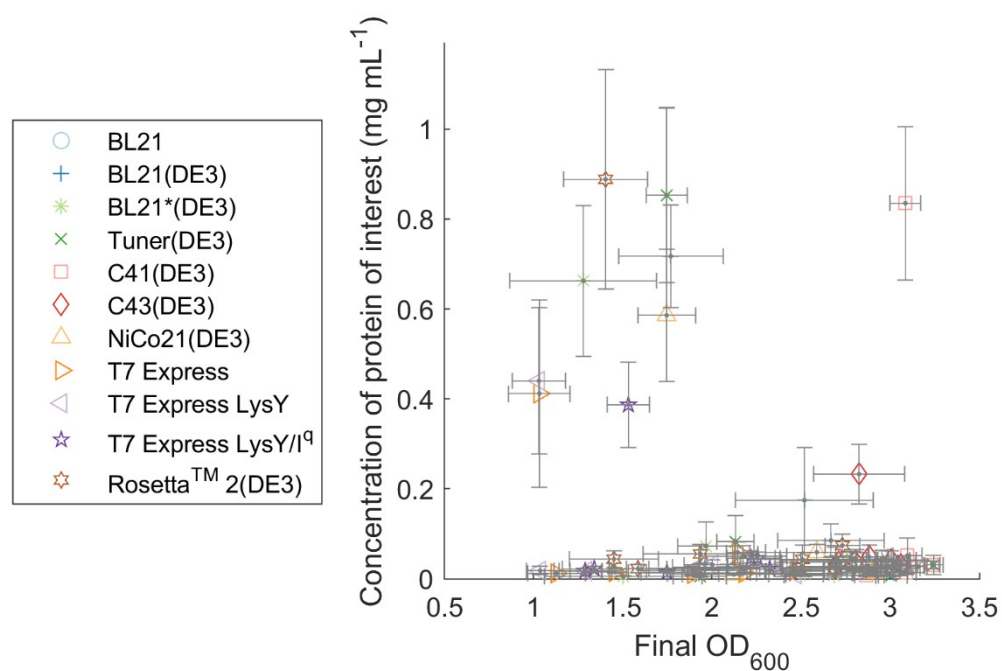


Figure S74. Final OD₆₀₀ vs. concentration of PPxY, determined by dot blot assay, sorted by strain. Error bars reflect the standard error across successfully transformed samples.

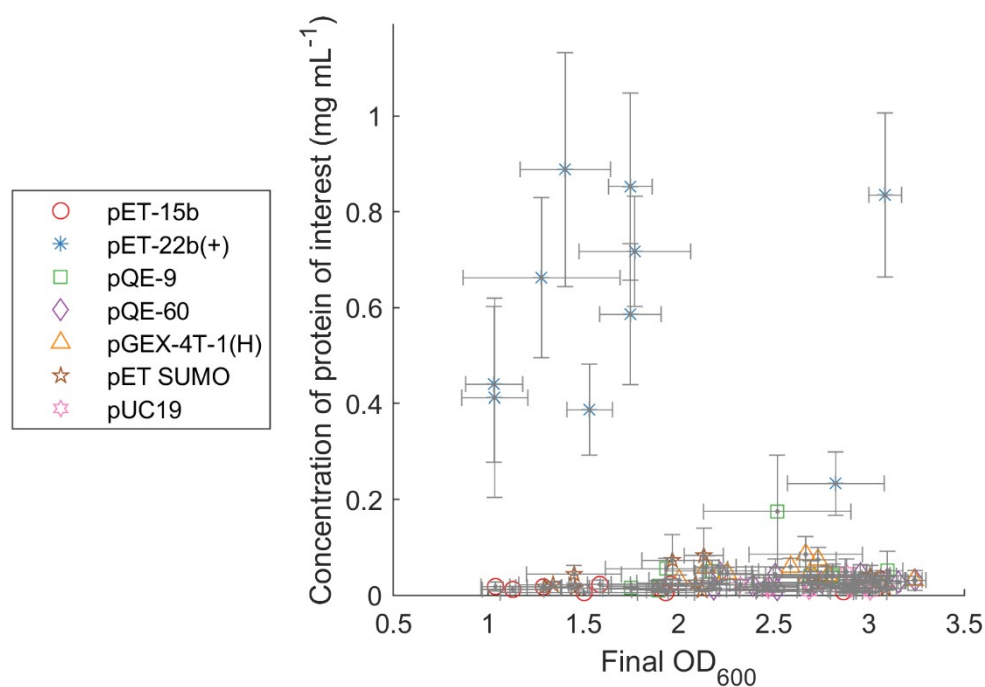


Figure S75. Final OD₆₀₀ vs. concentration of PPxY, determined by dot blot assay, sorted by vector. Error bars reflect the standard error across successfully transformed samples.

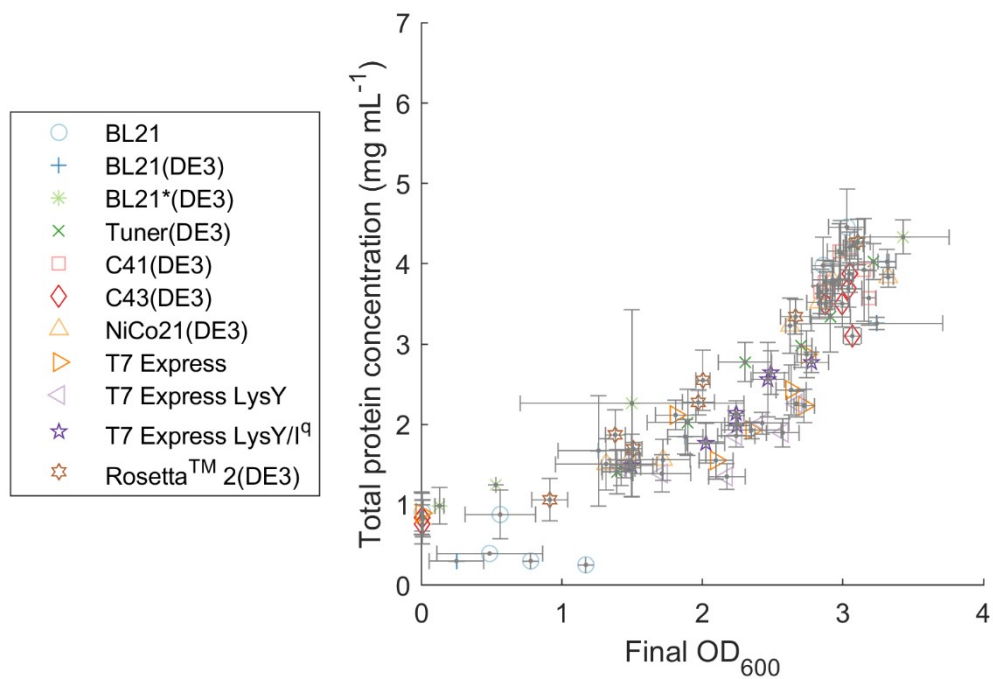


Figure S76. Final OD₆₀₀ vs. total protein concentration for Tag determined by Bradford assay, sorted by strain. Error bars reflect the standard error across successfully transformed samples.

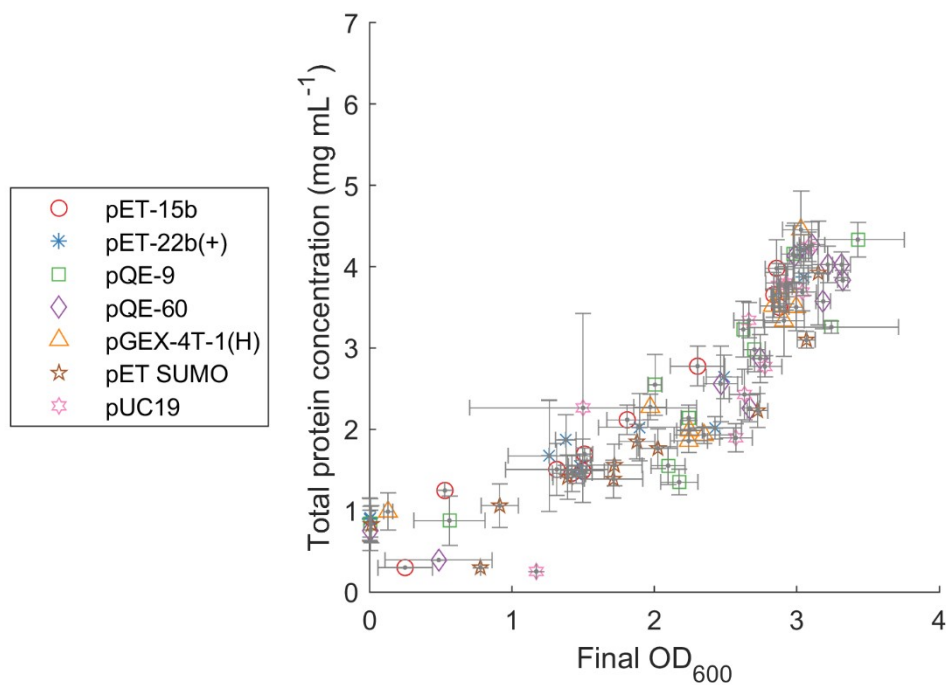


Figure S77. Final OD₆₀₀ vs. total protein concentration for Tag determined by Bradford assay, sorted by vector. Error bars reflect the standard error across successfully transformed samples.

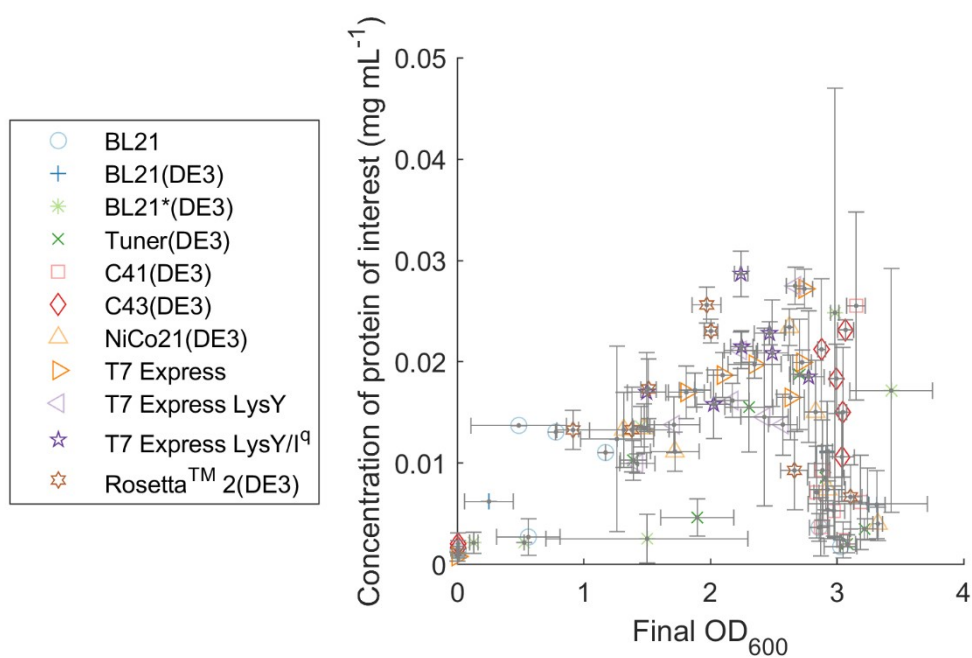


Figure S78. Final OD₆₀₀ vs. concentration of Tag, determined by dot blot assay, sorted by strain. Error bars reflect the standard error across successfully transformed samples.

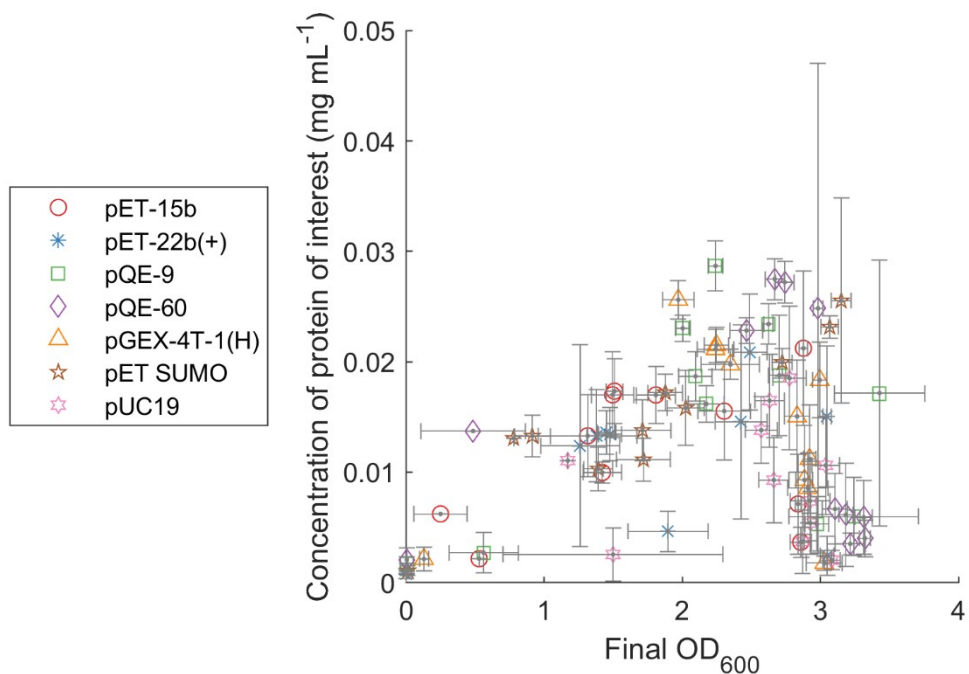


Figure S79. Final OD₆₀₀ vs. concentration of Tag, determined by dot blot assay, sorted by vector. Error bars reflect the standard error across successfully transformed samples.

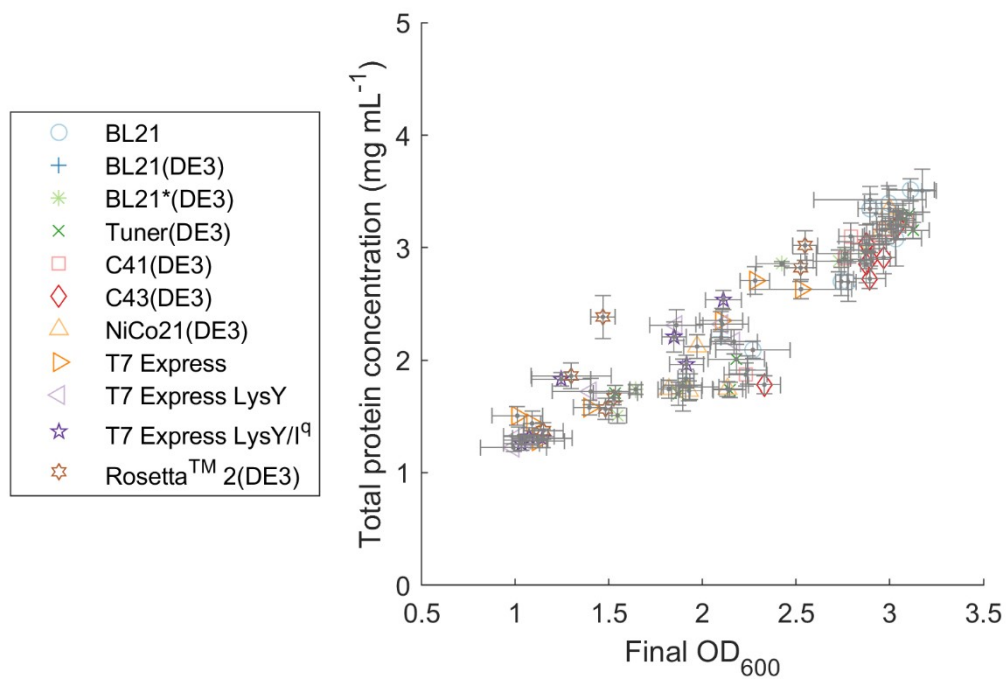


Figure S80. Final OD₆₀₀ vs. total protein concentration for ZE determined by Bradford assay, sorted by strain. Error bars reflect the standard error across successfully transformed samples.

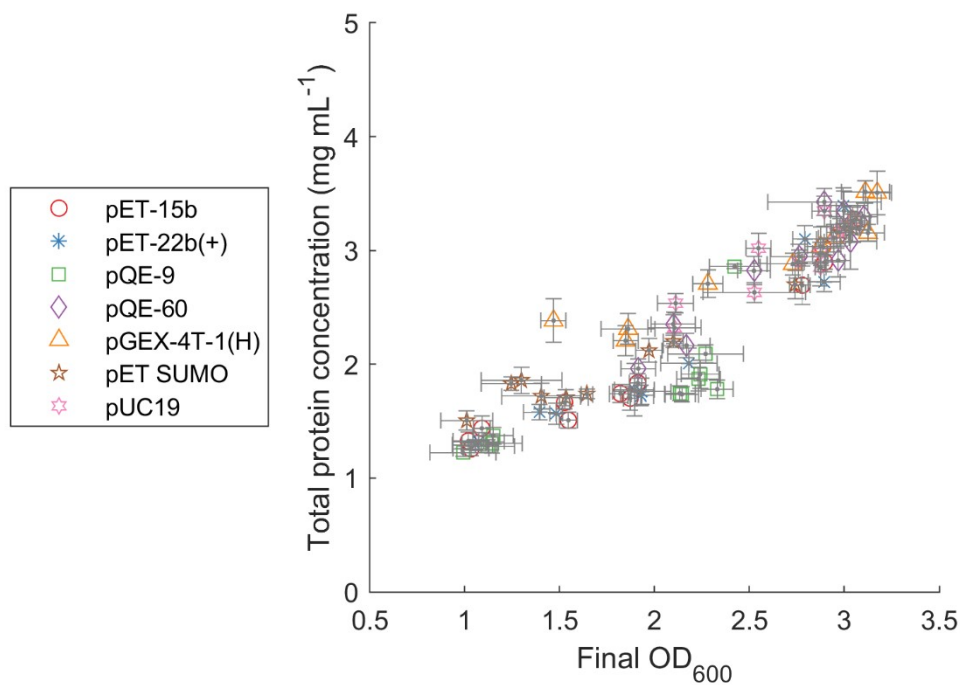


Figure S81. Final OD₆₀₀ vs. total protein concentration for ZE determined by Bradford assay, sorted by vector. Error bars reflect the standard error across successfully transformed samples.

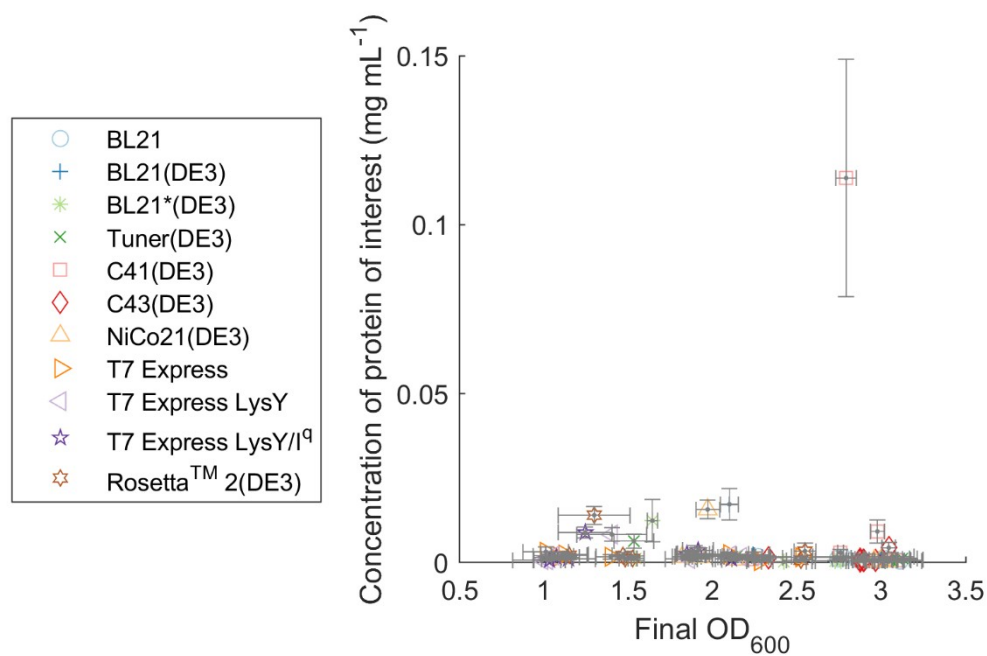


Figure S82. Final OD₆₀₀ vs. concentration of ZE, determined by dot blot assay, sorted by strain. Error bars reflect the standard error across successfully transformed samples.

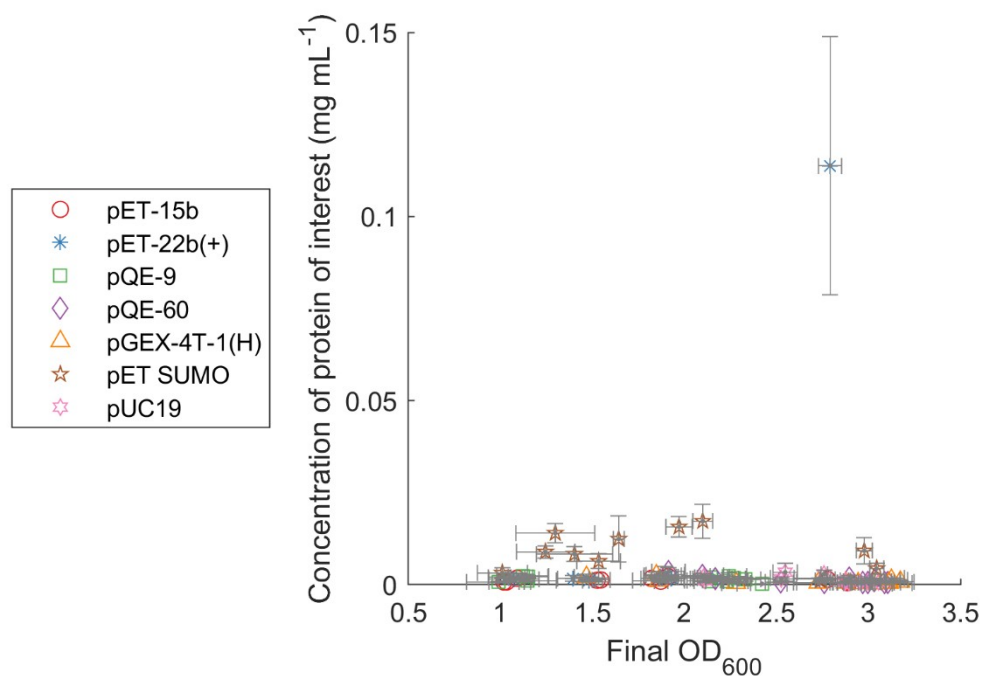


Figure S83. Final OD₆₀₀ vs. concentration of ZE, determined by dot blot assay, sorted by vector. Error bars reflect the standard error across successfully transformed samples.

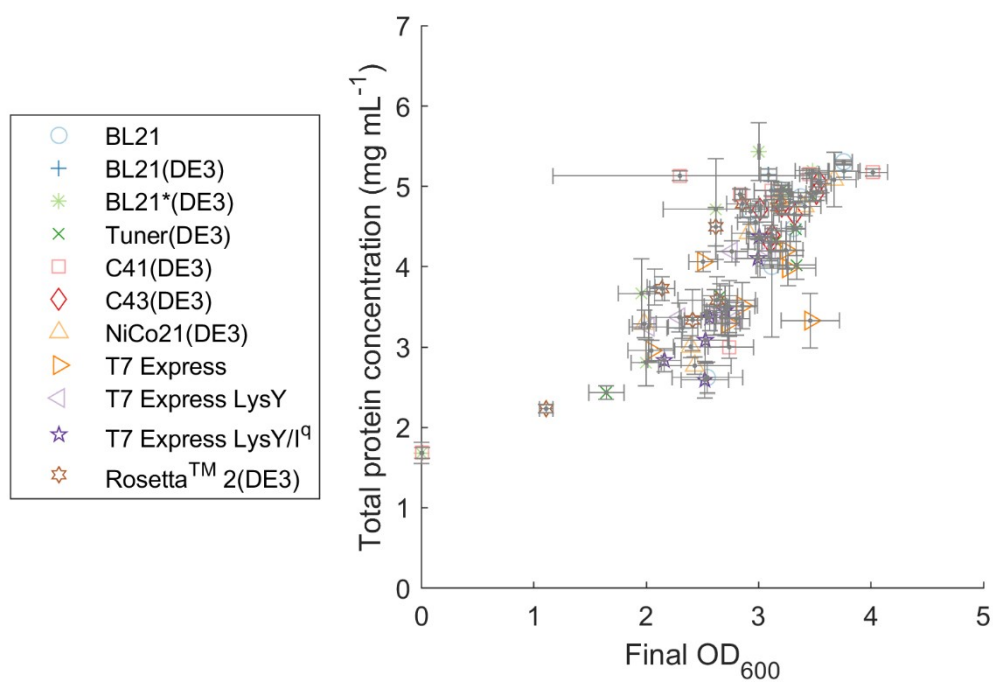


Figure S84. Final OD₆₀₀ vs. total protein concentration for ZR determined by Bradford assay, sorted by strain. Error bars reflect the standard error across successfully transformed samples.

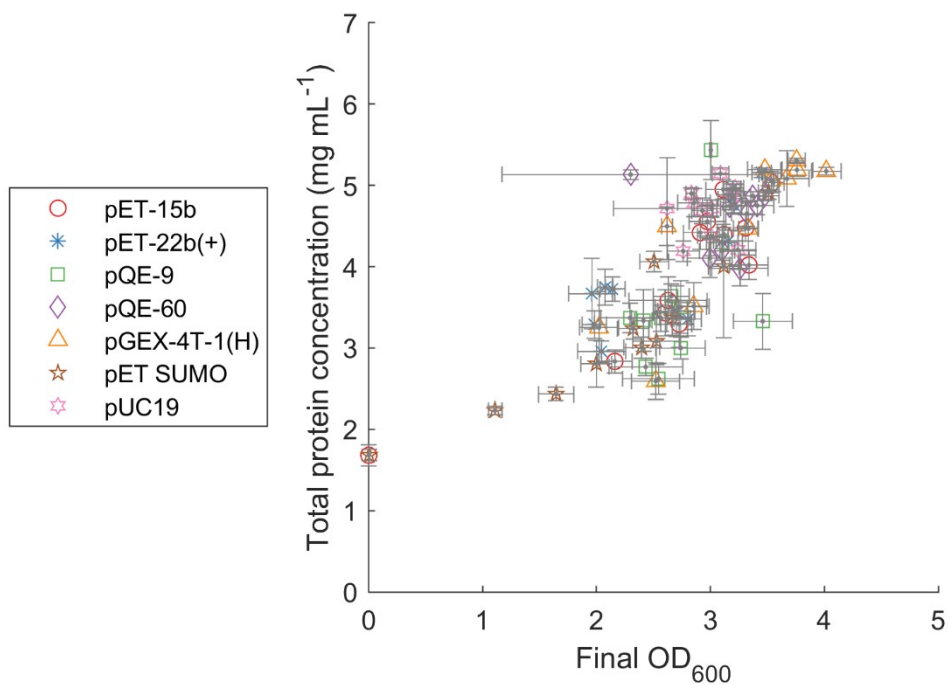


Figure S85. Final OD₆₀₀ vs. total protein concentration for ZR determined by Bradford assay, sorted by vector. Error bars reflect the standard error across successfully transformed samples.

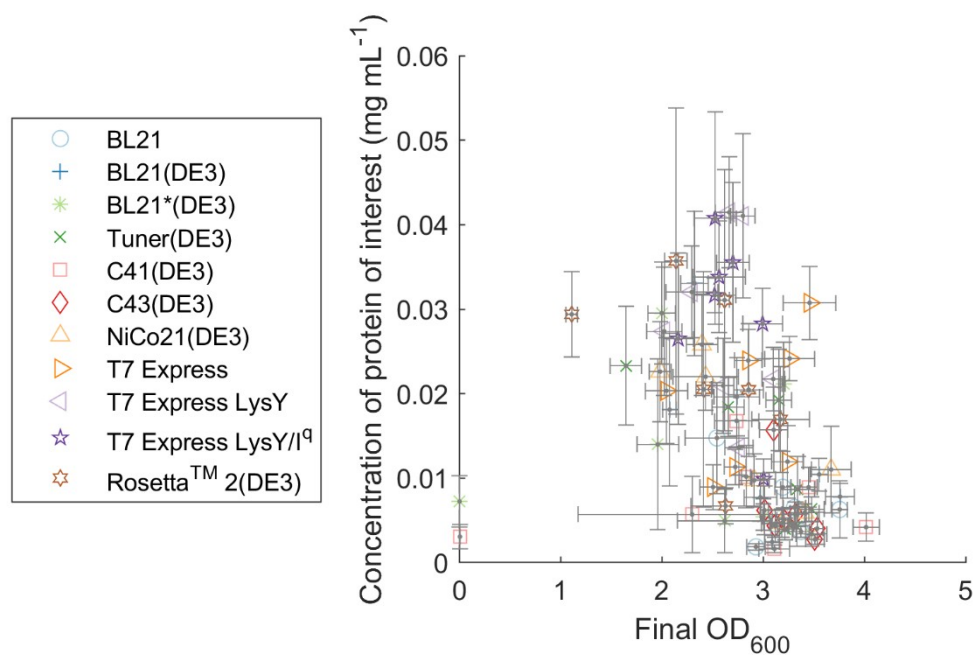


Figure S86. Final OD₆₀₀ vs. concentration of ZR, determined by dot blot assay, sorted by strain. Error bars reflect the standard error across successfully transformed samples.

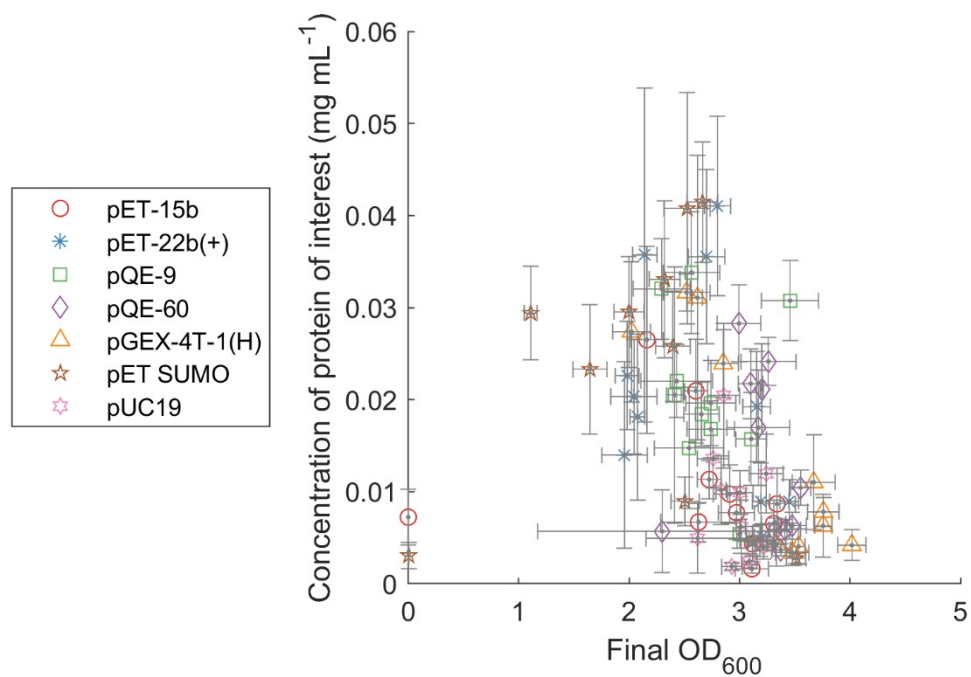


Figure S87. Final OD₆₀₀ vs. concentration of ZR, determined by dot blot assay, sorted by vector. Error bars reflect the standard error across successfully transformed samples.

H. Robot-based assay supplemental data

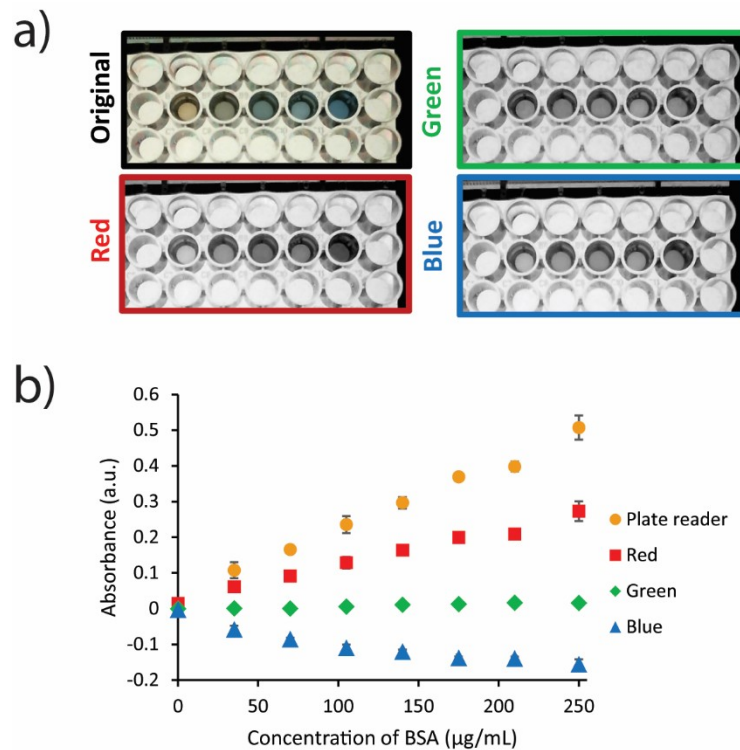


Figure S88. a) Photograph of Bradford assay wells with variable protein content, split into the red, green, and blue channels; b) Bradford assay absorbance as a function of bovine serum albumin (BSA) concentration measured by plate reader at 595 nm (yellow) and of the red, green, and blue channels.

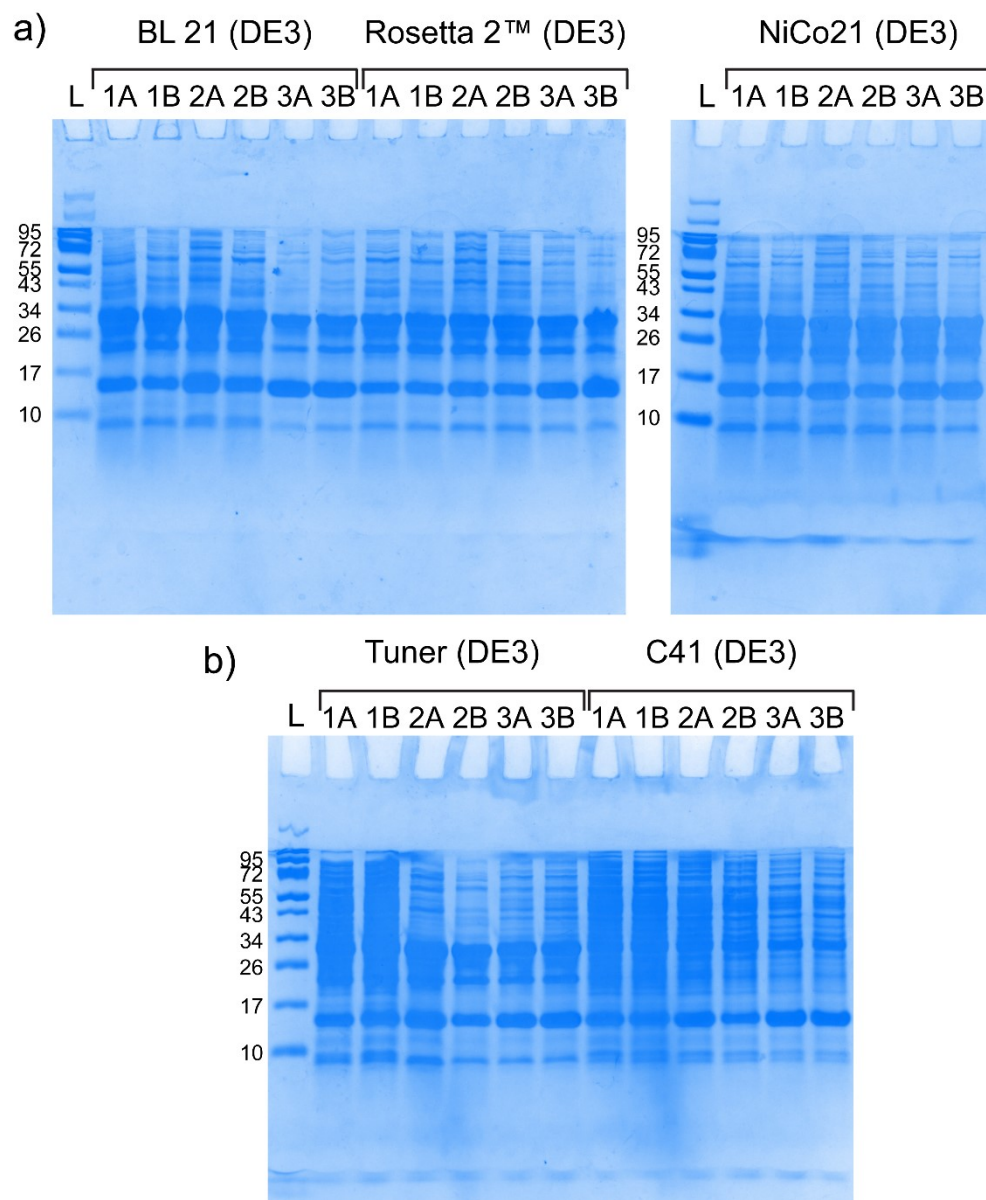


Figure S89. Denaturing SDS-PAGE gel of mCherry in pET22b vector in a) BL 21 (DE3), Rosetta 2™ (DE3), and NiCo21 (DE3), corresponding to the data points in Figure 5d circled in red, and b) Tuner (DE3) and C41 (DE3), corresponding to the data points in Figure 5d circled in green.

I. Gantt chart of process for a single protein

Table S2. Gantt Chart for Complete Automated Process

Table S2: Gantt Chart for Complete Automated Process			Day 1													
Step	Task	Duration	12 PM	1 PM	2 PM	3 PM	4 PM	5 PM	6 PM	7 PM	8 PM	9 PM	10 PM	11 PM		
Experiment Setup			Total: 02:00													
Robot, Container, and Reagent Setup																
Transformation			Total: 02:30													
Selection and outgrowth			Total: 20:30													
Transformation plate into Selection plates and outgrowth																
Expression Plate			Total: 24:30													
Transfer Selection plates into Growth plates			00:30													
Cell Growth			02:30													
OD Measurements			00:30													
Liquid Handling for Induction			00:30													
Protein Expression			20:00													
OD Measurements			00:30													
Cell Lysis			Total: 22:30													
Spin Down Cells			00:30													
Remove Supernatant			00:30													
Freeze Cells (1st time)			18:00													
Thaw Cells + Add Lysis Buffer			02:00													
Incubation			01:00													
Transfer to Shallow Well Plate			00:30													
Clear Supernatant			Total: 23:30													
Freeze Cells (2nd time)			02:00													
Thaw Cells			02:00													
Freeze Cell Overnight (3rd time)			16:30													
Thaw Cells			02:00													
Centrifuge Cell Debris			00:30													
Transfer Supernatant			00:30													
Assays - Total: 05:00			Total: 30:30													
Bradford Assay			03:00													
Dot Blot Assay			27:30													
			Protocol 1a, 1b1, and 1b2													
			Protocol 2													
			Protocol 13													

Robot=grey
Incubator=green
Centrifuge=red
Freezer=purple
Benchtop=blue

[illegible]

Step	Task	Duration	Day 3																							
			12 AM	1 AM	2 AM	3 AM	4 AM	5 AM	6 AM	7 AM	8 AM	9 AM	10 AM	11 AM	12 PM	1 PM	2 PM	3 PM	4 PM	5 PM	6 PM	7 PM	8 PM	9 PM	10 PM	11 PM
Experiment Setup		Total: 02:00																								
	Robot, Container, and Reagent Setup																									
Transformation		Total: 02:30																								
	Transformation																									
Selection and outgrowth		Total 20:30																								
	Transformation plate into Selection plates and outgrowth																									
Expression Plate		Total: 24:30																								
	Transfer Selection plates into Growth plates	00:30																								
	Cell Growth	02:30																								
	OD Measurements	00:30																								
	Liquid Handling for Induction	00:30																								
	Protein Expression	20:00																								
	OD Measurements	00:30																								
Cell Lysis		Total: 22:30																								
	Spin Down Cells	00:30																								
	Remove Supernatant	00:30																								
	Freeze Cells (1st time)	18:00																								
	Thaw Cells + Add Lysis Buffer	02:00																								
	Incubation	01:00																								
	Transfer to Shallow Well Plate	00:30																								
Clear Supernatant		Total: 23:30																								
	Freeze Cells (2nd time)	02:00																								
	Thaw Cells	02:00																								
	Freeze Cell Overnight (3rd time)	16:30																								
	Thaw Cells	02:00																								
	Centrifuge Cell Debris	00:30																								
	Transfer Supernatant	00:30																								
Assays - Total: 05:00		Total 30:30																								
	Bradford Assay	03:00																								
	Dot Blot Assay	27:30																								

Robot=grey
Incubator=green
Centrifuge=red
Freezer=purple
Benchtop=blue

Step	Task	Duration	Day 4																							
			12 AM	1 AM	2 AM	3 AM	4 AM	5 AM	6 AM	7 AM	8 AM	9 AM	10 AM	11 AM	12 PM	1 PM	2 PM	3 PM	4 PM	5 PM	6 PM	7 PM	8 PM	9 PM	10 PM	11 PM
Experiment Setup		Total: 02:00																								
	Robot, Container, and Reagent Setup																									
Transformation		Total: 02:30																								
	Transformation																									
Selection and outgrowth		Total 20:30																								
	Transformation plate into Selection plates and outgrowth																									
Expression Plate		Total: 24:30																								
	Transfer Selection plates into Growth plates	00:30																								
	Cell Growth	02:30																								
	OD Measurements	00:30																								
	Liquid Handling for Induction	00:30																								
	Protein Expression	20:00																								
	OD Measurements	00:30																								
Cell Lysis		Total: 22:30																								
	Spin Down Cells	00:30																								
	Remove Supernatant	00:30																								
	Freeze Cells (1st time)	18:00																								
	Thaw Cells + Add Lysis Buffer	02:00																								
	Incubation	01:00																								
	Transfer to Shallow Well Plate	00:30																								
Clear Supernatant		Total: 23:30																								
	Freeze Cells (2nd time)	02:00																								
	Thaw Cells	02:00																								
	Freeze Cell Overnight (3rd time)	16:30																								
	Thaw Cells	02:00																								
	Centrifuge Cell Debris	00:30																								
	Transfer Supernatant	00:30																								
Assays - Total: 05:00		Total 30:30																								
	Bradford Assay	03:00																								
	Dot Blot Assay	27:30																								

Robot=grey
Incubator=green
Centrifuge=red
Freezer=purple
Benchtop=blue

Step	Task	Duration	Day 5																							
			12 AM	1 AM	2 AM	3 AM	4 AM	5 AM	6 AM	7 AM	8 AM	9 AM	10 AM	11 AM	12 PM	1 PM	2 PM	3 PM	4 PM	5 PM	6 PM	7 PM	8 PM	9 PM	10 PM	11 PM
Experiment Setup		Total: 02:00																								
	Robot, Container, and Reagent Setup																									
Transformation		Total: 02:30																								
	Transformation																									
Selection and outgrowth		Total 20:30																								
	Transformation plate into Selection plates and outgrowth																									
Expression Plate		Total: 24:30																								
	Transfer Selection plates into Growth plates	00:30																								
	Cell Growth	02:30																								
	OD Measurements	00:30																								
	Liquid Handling for Induction	00:30																								
	Protein Expression	20:00																								
	OD Measurements	00:30																								
Cell Lysis		Total: 22:30																								
	Spin Down Cells	00:30																								
	Remove Supernatant	00:30																								
	Freeze Cells (1st time)	18:00																								
	Thaw Cells + Add Lysis Buffer	02:00																								
	Incubation	01:00																								
	Transfer to Shallow Well Plate	00:30																								
Clear Supernatant		Total: 23:30																								
	Freeze Cells (2nd time)	02:00																								
	Thaw Cells	02:00																								
	Freeze Cell Overnight (3rd time)	16:30																								
	Thaw Cells	02:00																								
	Centrifuge Cell Debris	00:30																								
	Transfer Supernatant	00:30																								
Assays - Total: 05:00		Total 30:30																								
	Bradford Assay	03:00																								
	Dot Blot Assay	27:30																								

Step	Task	Duration	Day 6																							
			12 AM	1 AM	2 AM	3 AM	4 AM	5 AM	6 AM	7 AM	8 AM	9 AM	10 AM	11 AM	12 PM	1 PM	2 PM	3 PM	4 PM	5 PM	6 PM					
Experiment Setup		Total: 02:00																								
	Robot, Container, and Reagent Setup																									
Transformation		Total: 02:30																								
	Transformation																									
Selection and outgrowth		Total 20:30																								
	Transformation plate into Selection plates and outgrowth																									
Expression Plate		Total: 24:30																								
	Transfer Selection plates into Growth plates	00:30																								
	Cell Growth	02:30																								
	OD Measurements	00:30																								
	Liquid Handling for Induction	00:30																								
	Protein Expression	20:00																								
	OD Measurements	00:30																								
Cell Lysis		Total: 22:30																								
	Spin Down Cells	00:30																								
	Remove Supernatant	00:30																								
	Freeze Cells (1st time)	18:00																								
	Thaw Cells + Add Lysis Buffer	02:00																								
	Incubation	01:00																								
	Transfer to Shallow Well Plate	00:30																								
Clear Supernatant		Total: 23:30																								
	Freeze Cells (2nd time)	02:00																								
	Thaw Cells	02:00																								
	Freeze Cell Overnight (3rd time)	16:30																								
	Thaw Cells	02:00																								
	Centrifuge Cell Debris	00:30																								
	Transfer Supernatant	00:30																								
Assays - Total: 05:00		Total 30:30																								
	Bradford Assay	03:00																								
	Dot Blot Assay	27:30																								

J. Cost table for platform

Table S3. Capital, consumables, supplies, and time costs.

Capital investments (prices as of May 2022 unless otherwise noted)		
Equipment	Price (USD)	Source
Opentrons OT-2 (includes 300 uL single- and multi-channel pipette)	\$ 7,700.00	Opentrons
Opentrons 20 uL multichannel pipette	\$ 1,800.00	Opentrons
Opentrons OT-1 (no pipettes, not currently available)	\$ 3,000.00	Opentrons
Peltier stage (with OT-1)	-	Opentrons
Seed 300K Pixel USB 2.0 Mini Webcam	\$ 8.90	DigiKey
Total capital investment	\$ 12,508.90	
Consumables (per protein)		
47 boxes of 300 uL pipette tips	\$ 155.10	Opentrons
12 boxes of 20 uL pipette tips	\$ 39.60	Opentrons
13 deep well plates (reusable)	\$ 66.33	USA Scientific
12 AeraSeal Sealing Films	\$ 10.26	VWR
5 reservoirs (reusable)	\$ 15.68	Thomas Scientific (NEST)
1 12 column reservoir (reusable)	\$ 2.67	USA Scientific
864 Roll & Grow Spherical Glass Plating Beads (reusable)	\$ 37.51	VWR
34 shallow well plates	\$ 97.08	VWR
6 PCR plates	\$ 12.70	VWR
0.45 um nitrocellulose membrane roll (562.5 cm ²)	\$ 14.70	Bio-Rad
6 Bio-Dot filter papers	\$ 4.08	Bio-Rad
Total consumables	\$ 455.72	
Reagents		
Ampicillin, kanamycin, and chloramphenicol	\$ 1.39	GoldBio
12 competent cell prepared by Zymo Mix & Go!	\$ 29.41	Zymo Research
1 L LB	\$ 4.17	FisherScientific and VWR
100 mL SOC	\$ 2.00	FisherScientific and VWR
120 mL MENT buffer with 0.5 mg mL lysozyme and 0.1 mg mL DNase I	\$ 9.14	FisherScientific and VWR
40 mL 20 mM IPTG	\$ 1.01	GoldBio
120 mL Bradford reagent	\$ 14.89	Bio-Rad
1 L TBS	\$ 0.34	VWR
1 L TTTBS	\$ 0.40	VWR
500 mL development solution	\$ 2.72	VWR
60 mL blocking solution	\$ 0.15	VWR
60 mL primary 4xHis AB solution (can be reused at least 5 times)	\$ 53.84	Qiagen
60 mL secondary AB solution (can be reused at least 5 times)	\$ 5.15	Millipore Sigma

1.2 mL NBT solution	\$ 17.64	Millipore Sigma
0.6 mL BCIP solution	\$ 12.36	Millipore Sigma
Total reagent cost	\$ 154.60	
Total cost per protein	\$ 610.32	
Person-hours		
Cloning	20	
Protein expression		
Day 1	3	
Day 2	2	
Day 3	1.5	
Day 4	1	
Day 5	4	
Day 6	8	
Sum person-hours	39.5	

K. Competent cell efficiencies

Table S4. Competent cell efficiencies (calculated using heat shock method in triplicate with pUC19 at .05 ng mL⁻¹ on ampicillin plates.

Cell strain	Efficiency (cfu µg ⁻¹)
BL21	2.8 x 10 ⁶
BL21 (DE3)	2.0 x 10 ⁶
BL21* (DE3)	1.2 x 10 ⁶
Tuner (DE3)	1.3 x 10 ⁷
C41 (DE3)	1.2 x 10 ⁶
C43 (DE3)	3.2 x 10 ⁶
NiCo21 (DE3)	1.6 x 10 ⁶
T7 Express	3.5 x 10 ⁷
T7 Express <i>lysY</i>	1.8 x 10 ⁷
T7 Express <i>lysY/I^q</i>	1.6 x 10 ⁷
Rosetta™ 2 (DE3)	4.8 x 10 ⁶
SG13009 with PREP4 plasmid	9.4 x 10 ⁶

L. References

1. B. M. Seifried, J. Cao and B. D. Olsen, *Bioconjugate Chemistry*, 2018, **29**, 1876-1884.