

**A peer-reviewed version of this preprint was published in PeerJ on 10 July 2017.**

[View the peer-reviewed version](https://peerj.com/articles/cs-120) (peerj.com/articles/cs-120), which is the preferred citable publication unless you specifically need to cite this preprint.

Swainston N, Currin A, Green L, Breitling R, Day PJ, Kell DB. 2017. CodonGenie: optimised ambiguous codon design tools. PeerJ Computer Science 3:e120 <https://doi.org/10.7717/peerj-cs.120>

# CodonGenie: optimised ambiguous codon design tools

Neil Swainston <sup>Corresp., 1</sup>, Andrew Currin <sup>1</sup>, Lucy Green <sup>1</sup>, Rainer Breitling <sup>1,2</sup>, Philip J Day <sup>3</sup>, Douglas B Kell <sup>1,2</sup>

<sup>1</sup> Manchester Centre for Synthetic Biology of Fine and Speciality Chemicals (SYNBIOCHEM), University of Manchester, Manchester, United Kingdom

<sup>2</sup> School of Chemistry, University of Manchester, Manchester, United Kingdom

<sup>3</sup> Faculty of Biology, Medicine and Health, University of Manchester, Manchester, United Kingdom

Corresponding Author: Neil Swainston

Email address: neil.swainston@manchester.ac.uk

CodonGenie, freely available from <http://codon.synbiochem.co.uk>, is a simple web application for designing ambiguous codons to support protein mutagenesis applications. Ambiguous codons are derived from specific heterogeneous nucleotide mixtures, which create sequence degeneracy when synthesised in a DNA library. In directed evolution studies, such codons are carefully selected to encode multiple amino acids. For example, the codon NTN, where the code N denotes a mixture of all four nucleotides, will encode a mixture of phenylalanine, leucine, isoleucine, methionine and valine. Given a user-defined target collection of amino acids matched to an intended host organism, CodonGenie designs and analyses all ambiguous codons that encode the required amino acids. The codons are ranked according to their efficiency in encoding the required amino acids while minimising the inclusion of additional amino acids and stop codons. Organism-specific codon usage is also considered.

# 1 **CodonGenie: optimised ambiguous codon design tools**

2 <sup>1,\*</sup>Neil Swainston, <sup>1</sup>Andrew Currin, <sup>1</sup>Lucy Green, <sup>1,2</sup>Rainer Breitling, <sup>3</sup>Philip J Day, <sup>1,2</sup>Douglas B  
3 Kell

4 <sup>1</sup>Manchester Centre for Synthetic Biology of Fine and Speciality Chemicals (SYNBIOCHEM),  
5 Manchester Institute of Biotechnology, University of Manchester, Manchester M1 7DN, United  
6 Kingdom.

7 <sup>2</sup>School of Chemistry, University of Manchester, Manchester M13 9PL, United Kingdom.

8 <sup>3</sup>Faculty of Biology, Medicine and Health, University of Manchester, Manchester M13 9PL,  
9 United Kingdom.

10 \*Corresponding author.

## 11 **Abstract**

12 CodonGenie, freely available from <http://codon.synbiochem.co.uk>, is a simple web application  
13 for designing ambiguous codons to support protein mutagenesis applications. Ambiguous codons  
14 are derived from specific heterogeneous nucleotide mixtures, which create sequence degeneracy  
15 when synthesised in a DNA library. In directed evolution studies, such codons are carefully  
16 selected to encode multiple amino acids. For example, the codon NTN, where the code N  
17 denotes a mixture of all four nucleotides, will encode a mixture of phenylalanine, leucine,  
18 isoleucine, methionine and valine. Given a user-defined target collection of amino acids matched  
19 to an intended host organism, CodonGenie designs and analyses all ambiguous codons that  
20 encode the required amino acids. The codons are ranked according to their efficiency in  
21 encoding the required amino acids while minimising the inclusion of additional amino acids and  
22 stop codons. Organism-specific codon usage is also considered.

23 **Keywords:** codon, directed evolution, mutagenesis, protein engineering, enzyme engineering,  
24 industrial biotechnology.

## 25 **Introduction**

26 Protein engineering seeks to synthesise proteins possessing particular function and structure  
27 through directed evolution approaches, and is a discipline with a long history (Jäckel, Kast &  
28 Hilvert, 2008). Traditional approaches include error-prone PCR and site-directed mutagenesis,  
29 and both approaches can produce reasonably large variant libraries that can be screened for a  
30 range of desired features; typically achieving an increased enzymatic activity over the wild type  
31 variant. These approaches have had a number of successes, but suffer from the limitations of  
32 being unable, a) to control the specific sites and nature of introduced mutations (in the case of  
33 error-prone PCR); and b) to generate much larger variant libraries – including the introduction of

34 mutations away from the active site – in the case of site-directed mutagenesis.

35 In contrast, more recently introduced synthetic biology approaches to protein engineering have  
36 allowed for the controlled and large-scale mutagenesis of wild-type proteins (Currin et al., 2015).  
37 The ability to design and assemble synthetic DNA *de novo*, introducing variant codons (those  
38 containing mixtures of nucleotides) at precisely defined positions, allows for the synthesis and  
39 expression of large and diverse combinatorial libraries, in which the position and biochemical  
40 nature of the mutations are fully controlled (Swainston et al., 2014; Currin et al., 2014; Currin et  
41 al., 2017).

42 The design of variant protein libraries typically involves a manual process in which required sites  
43 for mutation are selected, and ambiguous codons designed to introduce controlled variation in  
44 these positions. In this process, one may wish to design a codon to specify any subset of amino  
45 acids in a given position. Since each amino acid may be included in the subset or otherwise, the  
46 number of possible subsets is  $2^{20} - 1$ , i.e. there are 1,048,575 possible subsets of 20 amino acids,  
47 not all of which are uniquely designable using ambiguous codons (of which there are less than  
48  $15^3 - 4^3 = 3311$ , the exact number depending on the genetic code used by an organism).

49 Given the degeneracy of the codon table, there are often multiple ways to encode a chosen set of  
50 amino acids. The experimenter must a) decide if it is feasible to encode all desired amino acids  
51 (Mena & Daugherty, 2005); b) determine whether this creates an acceptable number of sequence  
52 combinations (depending on screening capability and throughput) (Kille et al., 2013; Lutz,  
53 2010); and c) consider the codon usage of the organism to be used (Nakamura, Gojobori &  
54 Ikemura, 2000). It therefore follows that the design of ambiguous codons is non-trivial, and as  
55 such, specialised software tools for the design of ambiguous codons have been recently released  
56 (Halweg-Edwards et al., 2016). The CodonGenie software presented here adds to this toolkit,  
57 and considers the above parameters according to the user input and ranks the variant codons with  
58 respect to the host organism to provide a quick and easy-to-use means of selecting the optimal  
59 variant codon.

## 60 **Materials & Methods**

### 61 **Algorithm**

62 The standard codon table is such that 17 of the 20 naturally occurring amino acids are encoded  
63 by codons with fixed bases in the first and second positions, with the third “wobble”-position  
64 allowing variation that accounts for the degeneracy of the DNA code. Determining optimal  
65 ambiguous codons for combinations of amino acids involves the following process, which is  
66 optimized for computational efficiency, compared to a brute-force examination of all possible  
67 ambiguous codons:

68 Align the first two positions and select the most specific ambiguous bases to encode the  
69 alignment. For example, with the combination asparagine and isoleucine (encoded by AA [CT]

70 and AT [ACT] respectively), the alignment of the first two positions is A [AT], i.e. AW.

71 All combinations of aligned wobble positions are calculated, i.e. [CA], [CC], [CT], [TA],  
72 [TC], [TT]. These are then collapsed into unique sets, in this example giving [CA], C, [CT],  
73 [TA] and T.

74 The first two and wobble position bases are combined to produce candidate ambiguous codons,  
75 which are scored as described below.

76 Three amino acids (leucine, arginine and serine) cannot be simply encoded by codons with fixed  
77 bases in the first and second positions. (For example, both CTN and TT [AG] encode leucine.)

78 For combinations including these more complex residues, the above algorithm is performed for  
79 each encoding and the results combined.

80 Note that CodonGenie returns not only the most “specific” ambiguous codons, that is, the codons  
81 that provide the fewest DNA variants whilst encoding all target amino acids. Providing results  
82 that include less specific ambiguous codons, which may also encode additional amino acids,  
83 allows the user to perform a trade-off between library size and codon specificity, depending on  
84 the experimental objective. A smaller library is generally advantageous for screening purposes,  
85 but may contain codons that are unfavoured by the target host organism.

## 86 Scoring

87 The goal of the scoring scheme is to preferentially rank the most efficient ambiguous codons.

88 That is, the ambiguous codons that encodes all of the required amino acids while minimising the  
89 encoding on non-desired amino acids.

90 The score for an ambiguous codon is therefore defined as the mean of the *value*,  $v_i$ , of each of the  
91 codons that it encodes. For codons that encode required amino acids,  $v_i$  is the ratio of the  
92 frequency of the codon  $f_i$  and the frequency of the most frequent synonymous codon  $f_j$  for the  
93 amino acid that it encodes. For codons that encode non-required amino acids,  $v_i$  is zero.

94 
$$\text{score} = \frac{1}{|C|} \sum_{i \in C} v_i \quad , \text{ where}$$

$$95 \quad v_i = \begin{cases} \frac{f_i}{\max(\{f_j : j \in S_i\})} & i \in R \\ 0 & i \notin R \end{cases}$$

96  $C = \{\text{all variants of ambiguous codon } c\}$

97  $A = \{\text{target amino acids}\}$

98  $a_i$  : amino acid encoded by codon  $i \in C$

99  $f_i$  : codon usage frequency of codon  $i \in C$

100  $S_i = \{j : a_j = a_i\}$  Set of synonymous codons of codon  $i$

101  $R = \{i \in C : a_i \in A\}$  Set of codon variants of  $c$  encoding target amino acids

102 This scoring algorithm thus achieves a principled trade-off between codon specificity, library  
103 size and codon favourability (according to the codon usage preferences of the target organism).

#### 104 **Web service access**

105 CodonGenie also offers a RESTful web service interface, supporting its integration with  
106 software pipelines. The Design method can be accessed by specifying required amino acids and  
107 required host organism (as an NCBI Taxonomy id (Federhen, 2012)) as follows:

108 <http://codon.synbiochem.co.uk/codons?aminoAcids=DE&organism=4932>

109 Similarly, the Analyse method can be accessed by specifying a variant codon and the required  
110 organism:

111 <http://codon.synbiochem.co.uk/codons?codon=NSS&organism=4932>

112 In both cases, results are returned in json format.

#### 113 **Distribution**

114 The web application is freely available from <http://codon.synbiochem.co.uk>. CodonGenie is  
115 written in Python (using the Flask framework) and HTML / Javascript (using the Bootstrap and  
116 AngularJS libraries) and is packaged as a Docker application for ease of deployment. Source  
117 code is available from <https://github.com/synbiochem/CodonGenie>.

#### 118 **Results and Discussion**

119 CodonGenie provides a simple web interface affording two functions: a) the design, and b) the  
120 analysis of ambiguous codons. Considering the Design module, the user specifies the  
121 combination of amino acids to be encoded and an organism in which the library will be  
122 expressed. The codon usage table is automatically extracted from the Codon Usage Database  
123 (Nakamura, Gojobori & Ikemura, 2000), which as of January 2017 provided support for 35,799  
124 organisms. CodonGenie then calculates suitable ambiguous codons and presents these in an  
125 interactive table (see Fig. 1).

126 The Analyse module provides the functionality of checking an existing ambiguous codon. Users  
127 specify a variant codon and required host organism, and the results returned indicate which  
128 amino acids are encoded along with their codon usage frequency.

129 The benefit of CodonGenie can be exemplified by the design of an ambiguous codon to encode  
130 non-polar amino acids phenylalanine, leucine, isoleucine, methionine and valine. A simple and

131 widely used ambiguous codon to encode this subset is NTN, which equates to 16 DNA variants.  
132 However, CodonGenie identifies that these same amino acids can be encoded by the DTK codon  
133 (where D denotes [AGT] and K denotes [GT]) using 6 variants. Selecting DTK therefore means  
134 fewer enzyme variants need to be screened to test all sequence combinations. This benefit is  
135 particularly significant when encoding multiple variant codons. For example, when using 3 DTK  
136 codons the library size is reduced from 4096 ( $16^3$ ) to 213 ( $6^3$ ) combinations.

## 137 **Conclusion**

138 CodonGenie provides two simple-to-use yet valuable tools that aid the design of variant protein  
139 libraries in mutagenesis and directed evolution studies. Through both its web and web service  
140 interfaces, CodonGenie is amenable to future integration with new and existing variant library  
141 design software tools (Swainston et al, 2014). Its modular and open-source format allows for  
142 straightforward adaptation to emerging needs in the synthetic biology community, in particular  
143 the consideration of augmented genetic codes and expanded genetic alphabets (Lajoie et al,  
144 2013; Zhang, 2017).

## 145 **References**

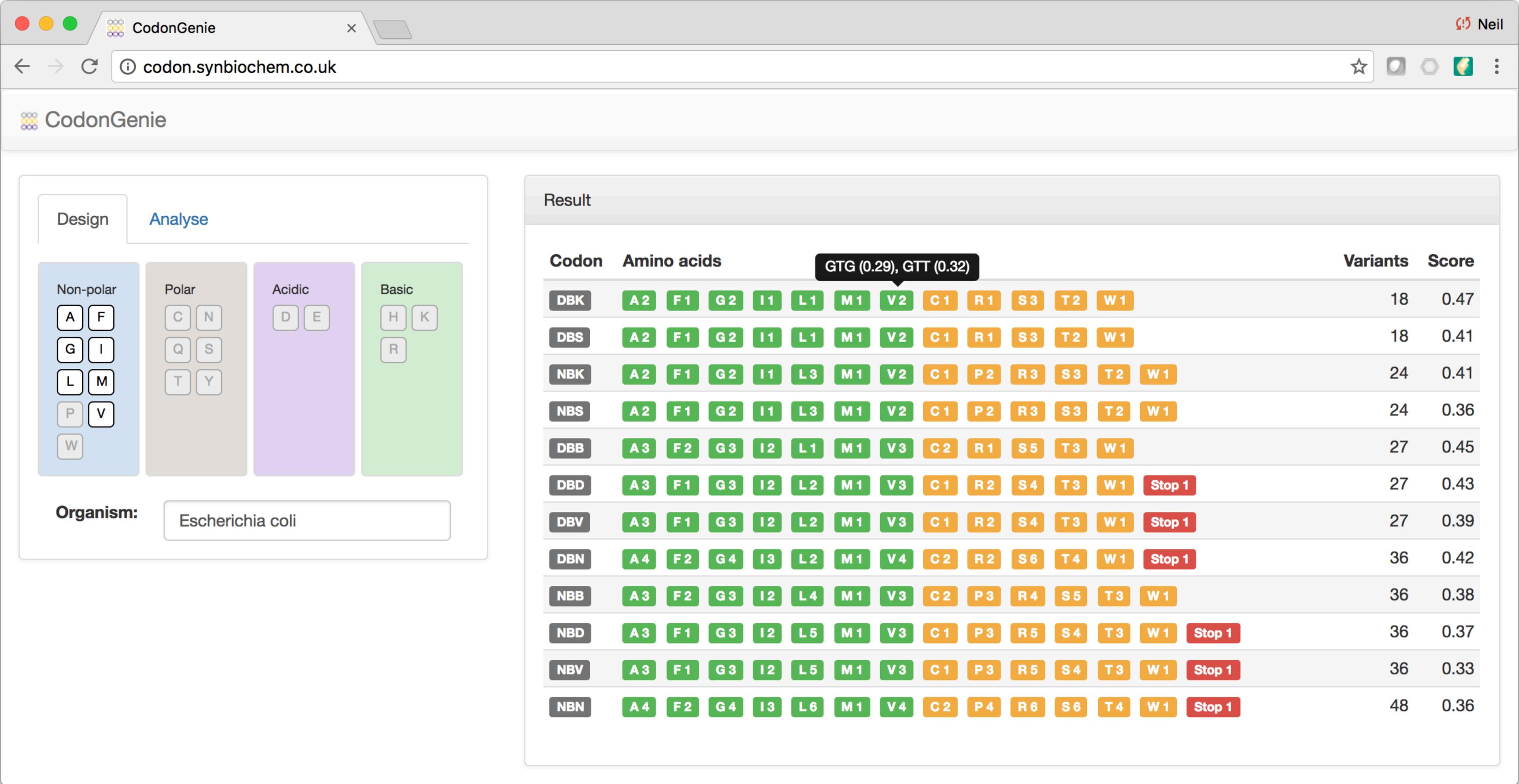
- 146 Currin, A., Swainston, N., Day, P. J., and Kell, D. B. (2014) SpeedyGenes: an improved gene  
147 synthesis method for the efficient production of error-corrected, synthetic protein libraries for  
148 directed evolution. *Protein Eng Des Sel.* 27: 273-80.
- 149 Currin, A., Swainston, N., Day, P. J., and Kell, D. B. (2015) Synthetic biology for the directed  
150 evolution of protein biocatalysts: navigating sequence space intelligently. *Chem Soc Rev.* 44:  
151 1172-239.
- 152 Currin, A., Swainston, N., Day, P. J., and Kell D. B. (2017) SpeedyGenes: Exploiting an  
153 Improved Gene Synthesis Method for the Efficient Production of Synthetic Protein Libraries for  
154 Directed Evolution. *Methods Mol Biol.* 1472: 63-78.
- 155 Federhen, S. (2012) The NCBI Taxonomy database. *Nucleic Acids Res.* 40, D136-43.
- 156 Halweg-Edwards AL, Pines G, Winkler JD, Pines A, Gill RT. (2016) A Web Interface for Codon  
157 Compression. *ACS Synth Biol.* 5: 1021-3.
- 158 Jäckel, C., Kast, P., Hilvert, D. (2008) Protein design by directed evolution. *Annu Rev Biophys.*  
159 37: 153-73.
- 160 Kille, S., Acevedo-Rocha, C.G., Parra, L.P., Zhang, Z.G., Opperman, D.J., Reetz, M.T.,  
161 Acevedo, J.P. (2013) Reducing codon redundancy and screening effort of combinatorial protein  
162 libraries created by saturation mutagenesis. *ACS Synth Biol.* 2: 83-92.
- 163 Lajoie, M.J., Rovner, A.J., Goodman, D.B., Aerni, H.R., Haimovich, A.D., Kuznetsov, G.,

- 164 Lutz, S. (2010) Beyond directed evolution--semi-rational protein engineering and design. *Curr*  
165 *Opin Biotechnol.* 21: 734-43.
- 166 Mena, M.A., Daugherty, P.S. (2005) Automated design of degenerate codon libraries. *Protein*  
167 *Eng Des Sel.* 18: 559-61.
- 168 Mercer, J.A., Wang, H.H., Carr, P.A., Mosberg, J.A., Rohland, N., Schultz, P.G., Jacobson, J.M.,  
169 Rinehart, J., Church, G.M., Isaacs, F.J. (2013) Genomically recoded organisms expand  
170 biological functions. *Science.* 342: 357-60.
- 171 Nakamura, Y., Gojobori, T., and Ikemura, T. (2000) Codon usage tabulated from the  
172 international DNA sequence databases: status for the year 2000. *Nucl Acids Res.* 28: 292.
- 173 Swainston, N., Currin, A., Day, P. J., and Kell, D. B. (2014) GeneGenie: optimized oligomer  
174 design for directed evolution. *Nucleic Acids Res.* 42: W395-400.
- 175 Zhang, Y., Lamb, B.M., Feldman, A.W., Zhou, A.X., Lavergne, T., Li, L., Romesberg, F.E.  
176 (2017) A semisynthetic organism engineered for the stable expansion of the genetic alphabet.  
177 *Proc Natl Acad Sci U S A.* DOI: 10.1073/pnas.1616443114. [Epub ahead of print].

## Figure 1 (on next page)

### CodonGenie Design interface

Users specify required amino acid combinations in the left-hand side panel. (Amino acids are grouped together in the interface in subsets of polar, non-polar, acidic and basic residues. In this example, the non-polar residues *A*, *F*, *G*, *I*, *L*, *M* and *V* have been selected.) Variant codons are listed in the Result panel, ordered by increasing number of Variants and decreasing codon Score (see Materials & Methods). The most specific codons are prioritised (e.g., the preferred codon in the above example, *DBK*, is *[AGT][CGT][GT]* and therefore encodes 18 DNA variants). Variant codons are shown in grey, with their encodings shown in green, orange and red for required amino acids, additional amino acids and stop codons, respectively. A given variant codon may encode an amino acid multiple times, and this is displayed in the output. For example, the preferred codon *DBK* encodes valine twice (with *GTG* and *GTT*), and these encodings and their organism-specific codon usage frequencies may be visualised through a tooltip.



Design

Analyse

Non-polar

A F  
G I  
L M  
P V  
W

Polar

C N  
Q S  
T Y

Acidic

D E

Basic

H K  
R

Organism:

Escherichia coli

Result

Codon	Amino acids	Variants	Score
DBK	A2 F1 G2 I1 L1 M1 V2 C1 R1 S3 T2 W1	18	0.47
DBS	A2 F1 G2 I1 L1 M1 V2 C1 R1 S3 T2 W1	18	0.41
NBK	A2 F1 G2 I1 L3 M1 V2 C1 P2 R3 S3 T2 W1	24	0.41
NBS	A2 F1 G2 I1 L3 M1 V2 C1 P2 R3 S3 T2 W1	24	0.36
DBB	A3 F2 G3 I2 L1 M1 V3 C2 R1 S5 T3 W1	27	0.45
DBD	A3 F1 G3 I2 L2 M1 V3 C1 R2 S4 T3 W1 Stop 1	27	0.43
DBV	A3 F1 G3 I2 L2 M1 V3 C1 R2 S4 T3 W1 Stop 1	27	0.39
DBN	A4 F2 G4 I3 L2 M1 V4 C2 R2 S6 T4 W1 Stop 1	36	0.42
NBB	A3 F2 G3 I2 L4 M1 V3 C2 P3 R4 S5 T3 W1	36	0.38
NBD	A3 F1 G3 I2 L5 M1 V3 C1 P3 R5 S4 T3 W1 Stop 1	36	0.37
NBV	A3 F1 G3 I2 L5 M1 V3 C1 P3 R5 S4 T3 W1 Stop 1	36	0.33
NBN	A4 F2 G4 I3 L6 M1 V4 C2 P4 R6 S6 T4 W1 Stop 1	48	0.36