An Abstracting Transformation for Amino Acid Polymorphism

Anthony M. Castaldo, PhD
Research Assistant Professor
University of Texas, San Antonio
CS-TR-2012-002

February 21, 2012

1 Introduction and Terminology

We use the following standard table of Nucleic Acid Codes:

Nucleic Acid Code	**Meaning**
A	Adenosine
C	Cytosine
G	Guanine
T	Thymidine
R	A G
Y	C T
K	G T (not needed in this paper)
M	A C (not needed in this paper)
S	C G (not needed in this paper)
W	A T (not needed in this paper)
B	C G T (not A)
D	A G T (not C)
H	A C T (not G)
V	A C G (not T)
N	A C G T (any)

We use the following standard table of translating three nucleotide sequences (a codon) into amino acids:

Amino Acid (or signal)	**Codons**
A (Alanine)	GCT, GCC, GCA, GCG
C (Cysteine)	TGT, TGC
D (Aspartic Acid)	GAT, GAC
E (Glutamic Acid)	GAA, GAG
F (Phenylalanine)	TTT, TTC
G (Glycine)	GGT, GGC, GGA, GGG
H (Histidine)	CAT, CAC
I (Isoleucine)	ATT, ATC, ATA
K (Lysine)	AAA, AAG
L (Leucine)	TTA, TTG, CTT, CTC, CTA, CTG
M (Methionine) (START)	ATG
N (Asparagine)	AAT, AAC
P (Proline)	CCT, CCC, CCA, CCG
Q (Glutamine)	CAA, CAG
R (Arginine)	CGT, CGC, CGA, CGG, AGA, AGG
S (Serine)	TCT, TCC, TCA, TCG, AGT, AGC
T (Threonine)	ACT, ACC, ACA, ACG
V (Valine)	GTT, GTC, GTA, GTG
W (Tryptophan)	TGG
Y (Tyrosine)	TAT, TAC
STOP	TAA, TGA, TAG

At least within exons coding for proteins, we believe what is important is the sequence of amino acids produced, and because amino acids average about three possible codings, very different looking DNA sequences, S_1 and S_2, can code for precisely the same sequence of amino acids.

This paper describes an abstracting transformation which replaces each letter in the DNA sequence with a single letter mask that can be matched instead, such that if S_1 and S_2 both code for the same sequence of amino acids their mask sequence will be identical.

We begin with two examples. The first five-nucleotide sequence is ATGCA, and we will generalize the idea from there. We choose five because we want two nucleotides before the center letter and two afterward; this represents all three possible reading frames for the center letter (a reading frame

is where in the sequence the groups of three begin; if we do not know our reading frame, then the first 'A' can be the first of three, the second of three, or the third of three nucleotides in a codon: Three possible reading frames).

In frame 1, the center G is the third letter of codon ATG, Methionine, and there is no alternative coding.

In frame 2, the center G is the second letter of the codon TGC, Cysteine, and the alternative is TGT: But that does not change the center letter.

In frame 3, the center G is the first letter of the codon GCA, Alanine, which has alternatives GCT, GCC, GCA, and GCG: But all begin with 'G' so the first letter is not changed.

Thus, considering all possible reading frames and all possible polymorphisms of coding, the center letter of ATGCA must be G. It is invariant.

In our second example we will see variance; consider the five-nucleotide sequence CACTG:

```
CAC     (H: CAC, CAT)
 ACT    (T: ACT, ACC, ACA, ACG)
  CTG   (L: TTA, TTG, CTT, CTC, CTA, CTG)
```

So the possible codings are as follows. Note only the center letter is of concern to us, the other letters are in lower case:

```
caC..   // center belongs to H in reading frame 1.
caT..
.aCt.   // center belongs to T in reading frame 2.
.aCc.
.aCa.
.aCg.
..Tta   // center belongs to L in reading frame 3.
..Ttg
..Ctt
..Ctc
..Cta
..Ctg
```

Looking at just the third letter, the only alternatives are T and C, which means the center letter can NOT be A or G and code for the same sequence of amino acids in all three reading frames.

One can look at this mask as a 4-bit truth table; for a given five-nucleotide sequence, the center letter has up to 4 possible values, ACGT, which we can code as '1' for valid alternatives and '0' for invalid alternatives. Thus if GA are the only options, the code is '1010'. If CT are the only valid alternatives, '0101'.

Based on that simplistic analysis, there can be at most 15 possible mask codes, because '0000' is impossible. There are $4^5 = 1024$ possible five nucleotide sequences. We can examine all of them, and determine for the center letter the mask for that five nucleotide sequence. We did that with a simple computer program, subsequently sorting the values, we find that only eight masks are necessary.

The table below details them, along with how many of the possible 1024 five-nucleotide sequences actually employ the mask for the center letter. These masks are re-sorted from their numeric value in the program, which is arbitrary, to show clarity in the types of ambiguity allowed:

Mask LTR	**Counts**	**Percentage**	**Description**	**Standard Code**
`ACGT`	636	62.109%	Any	N
`.CGT`	12	1.172%	not A	B
`A.GT`	10	0.977%	not C	D
`AC.T`	68	6.641%	not G	H
`ACG.`	42	4.102%	not T	V
`A.G.`	96	9.375%	A or G	R
`.C.T`	132	12.891%	C or T	Y
`..G.`	28	2.734%	Definite G	G

The "Standard Code" is the standard nucleic acid code, which already provides for such masking in other contexts. For example, between the reading frame and amino acid polymorphism, 'N' means the nucleotide could be any of the four letters. Of course it is not just the reading frame that matters, polymorphism alone means that eight of the acids are effectively specified by their first two letters alone: Alanine (GCN), Glycine (GGN), Leucine (CTN), Proline (CCN), Arginine (CGN), Serine (TCN), Threonine (ACN), and Valine (GTN).

2 Transformation to a Mask Sequence

We can slide a five base pair window across a much larger sequence of N nucleotides, centered on positions $3 \ldots N-2$, to create a new sequence of masks that is $N-4$ elements long. We call this the "Mask Sequence."

There is one big advantage to this new mask sequence: By construction of the replacement algorithm, any two DNA sequences that produce the same amino acids in the same order will have the same mask sequence, regardless of which reading frame applies and regardless of what codon polymorphisms are used. Thus without *knowing* the reading frame, we are still not fooled into rejecting a match over polymorphism. If we *knew* the reading frame, we could just encode the amino acids directly and gain that same functionality; but without knowing it we can abstract out the reading frame with these masks.

The coding is obviously not a lossless coding; and a mask-sequence match does not ensure a letter-by-letter match. However, what a mask sequence **mismatch** means is that the original letter sequences are definitely NOT functionally identical in all reading frames; in at least one they would not produce the same sequence of amino acids.

Mask sequence matching can run much faster with a simple character-comparison algorithm. The mask sequence and letter sequence are the same length, so if we find a match between mask strings, we can revert to a previous and more detailed method on the original letter-strings, if that is important.

3 Utility and Speculations

The utility is in abstracting away the confounding factors of comparison of two DNA strands. The strength of the transformation is it allows us to search for matching strings without having to make allowances for valid amino-acid polymorphisms, and without picking a reading frame to do that. Thus it can allow us to find larger sequences of matching *amino acids,* not just similar strings of letters that, in exons, may not be functionally equivalent at all.

Those matching sequences could be evolutionarily conserved protein commonalities between species or molecules; the remains of earlier structures or molecules that have since diverged into multiple roles.

Because the masks are a function of the code used by ribosomes to select amino acids and construct proteins, they do not necessarily have any utility outside of the exons that make up the genes that will be input to the ribosomes. On the other hand, we have examined an NCBI database of 42,433 RNA sequences, 3971 of them were non-coding areas. We found no difference in distribution of these mask codes in the two areas, but we did find the 'N' code was under-represented. Here is the result of the scan of 1.2 billion nucleotides:

Code	**Expected %**	**Actual %**	**Ratio**
N (Any)	62.1094%	57.7729%	0.9306
B (not A)	1.1719%	1.6097%	1.3691
D (not C)	0.9766%	1.0791%	1.1046
H (not G)	6.6406%	5.0518%	0.7608
V (not T)	4.1016%	4.7919%	1.1692
R (A or G)	9.3750%	12.0384%	1.2821
Y (C or T)	12.8906%	14.2288%	1.1031
G (Definite)	2.7344%	3.4275%	1.2535

The ratio between expectation with a uniform distribution and what we find in actual biological sequences shows a distinct leaning toward the masks (B,R,G) at the expense of the masks (N,H). That may not indicate function, but perhaps some of these masks, because they are rare combinations that can be recognized locally (i.e with just five nucleotides), have been evolutionarily recruited as flags or signposts on the genome, either individually or in combinations.

So what we would like is to convert letter strings to mask strings, and search for **mask motifs**, to see what we find, and if we can link them to interesting regions like promoters, splice sites, or gene-start points. Alternatively, it would be interesting to convert existing motifs, promoters, splice sites we already know are interesting to mask sequences, in order to look for commonalities or clues to see if any mask sequences or combinations have any power as identifying motifs.

It seems spliceosomes in particular have to do their work without any reading frame. Perhaps because mask motifs would also be independent of any reading frame they can reveal something new: Mask motif flags that guide pre-splice manipulation.

Using masks instead of letters allows a further manipulation that can reduce a simple match-search time about 33%. We have 57.77% of masks are 'N'; about 16% of those are singletons (not adjacent to another 'N'). We can compress the mask string by using new codes for multiple-N, e.g. '1'='NN', '2'='NNN', ... '9'=10× 'N'. '992' means 10+10+3=23 'N' in a row. This scheme preserves exact comparison, and reduces string length (and search time) about 33%. it is lossless; so any matching techniques that work on strings (dynamic programming, indexing, graphing, etc) will work on the compressed strings, with the exception that length is not preserved (e.g. 'R9Y' is 12 nucleotides long).

4 The Code Table

This table shows the 1024 possible combinations discussed. The first two letters (of 5) are row labels, Letters 3 & 4 are column labels, letter 5 is ACGT in the 4 letter block they specify. For example, CA-AG is the 5th row, 9th letter, = 'VDVD'. So CAAGA has mask 'V', CAAGC has mask 'D'.

```
   AA  AC  AG  AT  CA  CC  CG  CT  GA  GC  GG  GT  TA  TC  TG  TT
AA RRRRRRRRVDVDRRRRYYYYYYYYYHHHHYYYYRRRRVVVVRRRRVVVVYYYYHHHHYYYYYYYY
AC NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
AG NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
AT NNNNHHHHNNNNHHHHNNNNNNNNNNNNNNNNRRRRGGGGGGGGGGGGHHHHHHHHHHHHHHHH
CA RRRRRRRRVDVDRRRRYYYYYYYYYHHHHYYYYRRRRVVVVRRRRVVVVYYYYHHHHYYYYYYYY
CC NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
CG NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
CT NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
GA RRRRRRRRVDVDRRRRYYYYYYYYYHHHHYYYYRRRRVVVVRRRRVVVVYYYYHHHHYYYYYYYY
GC NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
GG NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
GT NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
TA RRRRRRRRVDVDRRRRYYYYYYYYYHHHHYYYYRRRRVVVVRRRRVVVVYYYYHHHHYYYYYYYY
TC NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
TG RRRRRRRRVDVDRRRRYYYYYYYYYHHHHYYYYGGGGGGGGGGGGGGGGYYYYHHHHYYYYYYYY
TT NNNNNNNNNNNNNNNNBBBBBBBBNNNNBBBBNNNNNNNNNNNNNNNNYYYYHHHHYYYYYYYY
```

Nine rows are all 'N', eight of them begin with two letter codes that

specify a single amino acid, discussed earlier: AC, CC, CG, CT, GC, GG, GT, and TC.

The ninth such row is AG. It specifies either Arginine (AGA, AGG) or Serine (AGC, AGT), each of which have a two-letter polymorphism (CGN and TCN respectively) that allows the third letter to be anything ('N').

5 Future Work, To Be Determined

This abstraction transformation was invented by the author after some exposure to bioinformatics in 2005. The data is from NCBI, we reran some programs with more recent data when this paper was updated in 2012.

This abstracting transformation is something of a "solution in search of an application." Further exploration in 2005 required access to "special" sequence data unavailable at the time, such as known splice points or promoter regions. The method is being documented now as a Technical Report to allow outside collaboration that may provide such data.