THE ALGORITHM USED TO PRODUCE "SWINE FLU HEMAGGLUTININ"

STEPHAN ZIELINSKI

SOFTWARE: Perl code: 164 lines (including embedded sequence data); a God-awful mess typical of code and algorithm being developed in parallel. Produces .abc files; .abc files imported into Harmony Assistant X 9.2.2.a.

(Description of ABC notation):
http://en.wikipedia.org/wiki/Abc_notation

ALGORITHM:

Design decisions:

Key: A minor, because (A) it fit the mood I had in mind to evoke, and (B) with A minor, I didn't have to worry about sharps and flats.

Tempo: 120 BPM, the Harmony Assistant default. One beat per amino acid, because it seemed like as reasonable a place to start as any.

Time signature: 3/4 time. Because I wanted to make sure I retained the option to speed it up (and maybe switch to C major) to yield a polka. (Having heard the final result, I strongly doubt actually making those changes would yield a *good* polka. The dance music closest to ambient is trance; waltz/polka requires both more structure and a less random approach to syncopation.)

Mapping of amino acids to instruments and notes:

(This is what I was working from):
http://en.wikipedia.org/wiki/List_of_standard_amino_acids

Non-hydrophobic & neither aromatic nor aliphatic: controls piano. These nine amino acids sorted by van der Waals volume, high to low, and set up such that so bulky amino acids produce low notes and small ones produce high ones. (By analogy to basic acoustics: relatively big things produce slow resonances and thus lower notes.)

Symbol	Amino acid	van der Waals volume	Piano note (in ABC notation)
R	Arginine	148	C
K	Lysine	135	D
Q	Glutamine	114	E
E	Glutamic acid	109	F
N	Asparagine	96	G
T	Threonine	93	A
D	Aspartic acid	91	B
C	Cysteine	86	c
S	Serine	73	d

Aromatic: percussion.

Symbol	Amino acid	Percussive instrument
Y	Tyrosine	Snare
H	Histidine	Tambourine
W	Tryptophan	Bongo
F	Phenylalanine	Tom

Hydrophobic, but not aliphatic: controls pipe organ. Same sort of thing as the piano, a function of van der Waals volume:

Symbol	Amino acid	van der Waals volume	Organ note (in ABC notation)
M	Methionine	124	C,,
P	Proline	90	E,,
A	Alanine	67	A,,
G	Glycine	48	C,

Aliphatic: controls the bell-like droning synthesizer. (In Harmony Assistant: "Synth Pad -> Fantasia". It was going to be a contrabass, but it turned out I didn't like the way that sounded.) Here, though, the mapping is arbitrary rather than based on van der Waals volume-- both because isoleucine and leucine are isomers, and because I didn't understand the .abc notation as well as I thought I did, which led to a bug, which led to the code that eventually spat out something listenable using this mapping:

Symbol	Amino acid	van der Waals volume	Synthesizer note (in ABC notation)
I	Isoleucine	124	A,,,
V	Valine	105	C,,,
L	Leucine	124	E,,,

I don't have a good answer to "Specific notes aside, why THAT acid to instrument mapping, as opposed to any other?" I mean, to me, it's *obvious* that the answer to

> Which of these is a better analogy?
>
> (A) aromatic : aliphatic :: contrabass : percussion
> (B) aromatic : aliphatic :: percussion : contrabass

is (B). But to paraphrase one of my favorite lines from the movie *Hot Shots*: I don't have a flipping clue what I just meant, guys-- not a flipping clue.

Each voice has a simple state machine; it's fed the amino acid sequence as input, and runs along it, emitting one beat's worth of music for each acid. (Uh, not that I built it on top of a state machine library or anything. It's just a loop, incrementing beats and measures appropriately, saving data in variables as need be. But *computationally*, its acting like state machines.)

Piano: degenerate (stateless) state machine. If one of its nine amino acids is up, play the associated note; otherwise, rest.

Percussion:
  Internal state:
    For each drum sound: "I am the one that got played last"
  If this is the first beat of a (three beat) measure:
    If "I am the one that got played last" is set for anything:
      Play that one again
  If one of the four percussion amino acids is up:
    Play the associated percussion
    Set the "I am the one that got played last" for that drum sound
    Clear the other "I am the one that got played last" ones
  (Failing either of those, rest.)
(This means there are times there's more than one percussive sound happening at the beginning of a measure.)

Pipe organ:
  Internal state:
    $tone_organ, initialized to "z" (which means "rest" in ABC notation)
  If this is the first beat of a measure:

```
if (!(($measure + 4) % 8))) {
        $tone_organ = "z";
}
```

  If one of the pipe organ's four amino acids is up:
    Set $tone_organ to that note
  Play whatever $tone_organ is set to

Synthesizer:
  Same as the pipe organ, but the stop-playing condition is

```
if (!($measure % 8)) {
    $tone_contrabass = "z";
}
```

Post-processing:
  For the organ and synthesizer:
    For measures with three of the same note:
      Rewrite from three quarter notes to one dotted half note

Import each track into Harmony Assistant X, manually fiddle with voice assignments and mixer until it sounds okay. (As conceived, the piano was an oboe and the synth was a contrabass. Didn't sound good to me.)

(end of description of algorithm)

WHY I FIGURE IT WORKS (from a music theory standpoint):

At any given point in time, odds are the instruments are mostly playing notes from the set of A, C, and E. As it turns out, if you grab N musicians (regardless of instrument) and tell everyone to ignore what everyone else is playing and improvise something using only (what on a piano are) A, C, and E, you're going to get *something* that nevertheless is identifiably in the key of A minor-- made entirely of the triadic A minor chord and its close kin (such as the inverted A minor chord.) Combine that with another track where the improviser (still in isolation) is improvising using only notes from the A minor scale-- A B C D E F G A, no sharps or flats-- and while there's a greater variety of chords, odds are a listener will still say, "That's a mess-- but it's a mess in A minor."

This-- and I mean this literally-- is how wind chimes work. A wind chime's input is (depending on circumstances) either random, or a stochastic (quite possibly fractal) sequence. The high randomness doesn't matter-- the actual sound produced is in the same key as deliberately composed music.

Indeed, the whole state thing came later. Early versions of the code just had the instruments play when their acid came up, and that was it... and it sounded like wind chimes. Worse, actually-- wind chimes play chords. It was so boring, I had to do something to it so I could *stand* to listen to the whole thing in my attempt to see if I could hear any divisions that might correspond to (say) different functional groups of the protein. (Incidentally, trying to improvise a short explanation of this on the telephone to a stranger is what led me to emit the concise but nonsensical quote, "The only way I could look at that and make sense of it, since I'm not a biochemist, was, well, I could try translating it into another very organized form of information that people do, and then see if I could hear anything. And the answer, of course, turned out to be no.")

I did two state-related things things to increase the piece's musicality: made sure it had a more regular percussive rhythm, and stretched out the duration of the organ and synthesizer's notes.

Percussion was easy: the basic waltz rhythm is

```
ONE two three ONE two three
```

so as long as there's something hammering on the first beat in each measure, the piece gains a repetitious rhythmic structure consistent with canonical Western music. The actual solution-- play the last drum sound again-- is neither elegant, nor would it help bring out any details of the protein's structure. (Better would have been to hope that helical periodicity of a protein's alpha structure had major implications, and derive a rhythm track from something synced with it-- but (A) what hope I had of that was mighty faint, and (B) that periodicity does not map trivially onto a simple 3/4, 4/4, or 5/4 scheme. Each group of eighteen amino acids produces a helix with five complete turns, since the angle between each successive acid is about 100 degrees-- so it's 3.6 amino acids per turn of the helix, which is, like, ick. Besides, a protein is a three dimensional structure, with the alpha helix only being *a* level of detail. Even absent other proteins deliberately folding the new protein as it comes off the ribosome, all manner of bends and kinks appear; biochemistry is messy.)

Having the organ and synthesizer keep playing what they were playing before until they reached a particular point dictated by position *on the score* (rather than being dictated by the inherent structure of the protein) also has no chemical justification whatsoever-- but it does result in the sort of this-bar-I-go-next-bar-you-go-I-go-you-go dialogue characteristic of (much) human music. Since only one state machine is getting kicked at any particular point in time, a stretch of sequence with a lot of associated-with-the-piano amino acids will have a complicated piano part, but the organ (and synth) will either be silent, or just be providing depth by droning on with whatever note they sounded last. (Under those circumstances, listeners tend to follow the moving voice, and might even characterize the other voices as "quiet"-- even if they're playing just as loudly as they were when they played a new note every beat.) Conversely, a stretch of sequence with a lot of associated-with-the-organ acids will have a silent piano, but a complicated organ part. That the organ and synth tend to shut up at the end points of four-measure bars is strictly imitative of human music-- every now and then, voices will step back and let the other parts carry things along for a while.