

Musical Representations of the Fibonacci String and Proteins Using Mathematica

Erik Jensen¹ and Ronald J. Rusay^{1,2}

- 1) Diablo Valley College, Pleasant Hill, California 94523
- 2) University of California, Berkeley 94720

D) Fibonacci Strings.

The Fibonacci Sequence is a list of numbers derived from a mathematical model for the population growth of rabbits published by Leonardo da Piza (Fibonacci) in the early thirteenth century. The model assumes that the system starts with one young pair of rabbits, one male and one female. The young pair matures and then produces another young pair as offspring. No rabbits die. If a young pair is represented by 0, and a mature pair by 1, then the succession of generations can be represented by the following recursive mapping: 0->1, 1->10. Starting with 0, this endomorphism can be iterated ad infinitum to compose a string which represents the growth of a population.¹

Iteration	String	Total # of pairs
0	{0}	1
1	{1}	1
2	{10}	2
3	{101}	3
4	{10110}	5
5	{10110101}	8

The sequence of Fibonacci numbers (total # of pairs) and Fibonacci-like sequences are characteristic of many natural systems. There are many examples of the same repetitive pattern which is observed in the breeding of rabbits, such as the morphology of plants and flowers: buttercups have 5 petals, lilies have 3, delphiniums have 8, marigolds have 13, asters have 21 and so forth. Other examples are also observed in nature with this numerical pattern such as a pine cone, snail shell, and nautilus shell in the proportions of their spirals.²

The Fibonacci string is aperiodic. Given that the length of one space is equal to the length of the substring (L) minus 1 when L > 1, and the length of one space = 1 when L = 1; substrings of any size repeat with spaces between them equal to {1,0,1,1,0,...} in succession. This sequence is the Fibonacci string, itself. Thus, the Fibonacci string exhibits self-similarity, a type of symmetry.¹

The following Mathematica algorithm demonstrates this property of self-similarity in the Fibonacci string:

```
A = "101101011011010110101101101011011010110110101101011011010110101010
110110101101011011010110110101101011011010110110101101011011"
a = (2Range[25]-1);
b=Drop[(2(Range[25])-1)1];
B=Flatten[StringPosition[A,"10"]
x=(1/2)(Part[B,b]-Part[B,a])
```

$$x/.{(1/2)->1}$$

This algorithm creates a list of the positions for each character in a substring of the first 25 "10" substrings and condenses it into one list.

$$\{1, 2, 4, 5, 6, 7, \dots\} \text{ or } \{y_1, z_1, y_2, z_2, y_3, z_3, \dots\} \text{ where } y = \text{character 1 of "10"} \\ \text{"1 0"}_1 \quad \text{"1 0"}_2 \quad \text{"1 0"}_3 \qquad \qquad \qquad z = \text{character 2 of "10"}$$

- The equation $[y_n - z_{(n-1)}] / 2$; $\{n \mid n = \text{an integer } > \text{ or } = 2\}$ is applied to the list of substring character positions for increasing values of n .
- If there are no spaces between "1 0"_n and $\text{"1 0"}_{(n-1)}$, then the corresponding value in the string created by step 2 is $1/2$. If there is one space between "1 0"_n and $\text{"1 0"}_{(n-1)}$, then the corresponding value generated by step two is 1.

When the transformation $(1/2) \rightarrow 0$ is performed on the previous string, the Fibonacci string re-emerges.

The Mathematica algorithm written to create music from the Fibonacci string employs the following steps:

1) Generation of a string of a particular length.

The length of the string determines the length of the song. The total number of characters must be divisible by the total number of substrings. The total number of characters can be determined by the equation:

$$F = (G^n) / (5)^{1/2}$$

1. Where F is rounded down to the nearest integer.
2. $G = ([\text{square root of } 5] - 1) / 2 = \text{Golden Mean}$
3. $n = \text{the level the string is composed to: for iteration } 0, n = 1; \text{ for iteration } 1, n=2; \text{ for iteration } 2, n=3; \text{ and so forth.}^3$

2) Division of the string into substrings of a particular length.

The choice of substring length is arbitrary. Different lengths will yield different musical characteristics. A longer substring yields more complex music since there are more possible combinations of characters and therefore more possible substrings. Comparison of songs based on substrings of different lengths reveals the organizational characteristics of the string.

3) Assignment of note duration and pitch to the substrings.

Although there are no explicit mathematical rules for assigning sound characteristics to substrings in the string, a logical ordering of the substrings based

on the patterns of their characters is the criteria by which notes of increasing frequency are assigned.

Two parallel scales have been applied. In one scale, each of the notes is played for a shorter duration and with a different tonal quality. All of the substrings that begin with 0 are apportioned to one scale, and all of the substrings that begin with 1 belong to the other. A simple example follows:

<u>SCALE 1</u>			<u>SCALE 2</u>		
<i>Substring</i>	<i>Note</i>	<i>Duration</i>	<i>Substring</i>	<i>Note</i>	<i>Duration</i>
<i>{10101}</i>	<i>C</i>	<i>.5 sec</i>	<i>{01010}</i>	<i>C</i>	<i>.75 sec</i>
<i>{10110}</i>	<i>D</i>	<i>.5 sec</i>	<i>{01011}</i>	<i>D</i>	<i>.75 sec</i>
<i>{11010}</i>	<i>E</i>	<i>.5 sec</i>	<i>{01101}</i>	<i>E</i>	<i>.75 sec</i>

There is not always the same number of substrings per group. Note that in the Fibonacci string there are never more than two 1's in succession nor more than one 0 in succession.

An additional tonal dimension is added by employing two groups of tonal qualities within one of the scales based on patterns of the characters in the substrings. This simple system creates music with rhythmic and melodic properties.

The degree to which the order and musical properties of the song describe or correspond to order within the Fibonacci string is dependent on the means by which notes and duration are assigned to substrings. Different logical orderings yield music with many different properties, and to an extent show ordered similarities. One way to determine the extent to which the music describes the order of the Fibonacci string is to analyze a randomized set of substrings through the algorithm. Any significant differences in the resultant music must be caused by the order within the Fibonacci string. The algorithm does not employ any permutations of the string. Thus, the music is based on the order of characters in the Fibonacci string. Since the string is self-similar, the music may reflect inherent symmetry.

4) Playing the sequence of notes.

The notes are played in succession with standard Mathematica commands. The music package is not necessary for this step or any other step of the program.

The following algorithm was used to create music from the Fibonacci string based on seven character substrings. The scale chosen is C minor.

```

x={1,0};
Flatten[ReplaceRepeated[0,{0->1,1->x},MaxIterations->16]]
\[Phi]=Partition[%,7]

a={0,1,0,1,0,1,0};
b={0,1,0,1,0,1,1};
c={0,1,0,1,1,0,1};
d={0,1,1,0,1,0,1};
e={0,1,1,0,1,1,0};
f={1,0,1,0,1,0,1};
g={1,0,1,0,1,1,0};
h={1,0,1,1,0,1,0};
i={1,0,1,1,0,1,1};
j={1,1,0,1,0,1,0};
k={1,1,0,1,0,1,1};
l={1,1,0,1,1,0,1};

a'=Play[Sin[2(Pi)262t]-Sin[2(Pi)263t],{t,0,.5}];
b'=Play[Sin[2(Pi)294t]-Sin[2(Pi)295t],{t,0,.5}];
c'=Play[Sin[2(Pi)311t]-Sin[2(Pi)312t],{t,0,.5}];
d'=Play[Sin[2(Pi)349t]-Sin[2(Pi)350t],{t,0,.5}];
e'=Play[Sin[2(Pi)392t]-Sin[2(Pi)393t],{t,0,.5}];
f'=Play[Abs[Sin[2(Pi)262t]],{t,0,.35}];
g'=Play[Abs[Sin[2(Pi)294t]],{t,0,.35}];
h'=Play[Abs[Sin[2(Pi)311t]],{t,0,.35}];
i'=Play[Abs[Sin[2(Pi)349t]],{t,0,.35}];
j'=Play[Sin[2(Pi)392t],{t,0,.35}];
k'=Play[Sin[2(Pi)415t],{t,0,.35}];
l'=Play[Sin[2(Pi)466t],{t,0,.35}];

Show[\[Phi]/.{a->a',b->b',c->c',d->d',e->e',f->f',g->g',h->h',i->i',j->j',k->k',l->l'}]

```

II. Music based on the sequence of amino acids in a protein.

A program, which is very similar to the one just shown, is used to create music from the primary structure of proteins. Twenty different amino acids are usually observed in a protein. A note and duration is assigned to each of the amino acids on the basis of their physical and chemical characteristics. Increasing pitch corresponds to increasing complexity of the substituted side chain in the acid. A dual scale, which is variable, is divided on the basis of the polarity of the molecules. Different assignments will produce different musical characteristics. For example, pitch can be assigned on a relative basis to the molecule's atomic weight rather than the amino acid side chain. This will produce different music. The goal is to develop sounds which capture as much as possible of the structural and chemical information of the amino acid sequence so that the macromolecule, the protein, may be recognizable by its musical signature. Each protein has a unique amino acid sequence. Similarities in primary structure will be reflected by similarities in their

corresponding songs. A sample has been produced in C minor which is based on a portion of the enzymatic protein, *lysozyme* .

III. Further Possibilities.

- Music from the Fibonacci string based on substrings of different lengths combined melodically into one song. The resultant music could reveal different characteristics of order within the string.
- Music from:
 - 1) Strings that occur in mathematical problems (e.g. Tower of Hanoi and Lucas Number).
 - 2) Strings that represent socio-economic behavior such as stock market indices and consumer prices .
- Combining related strings into a single composition:

For example, the amino acid sequence of a protein and its corresponding DNA sequence being played simultaneously. A harmonious synchronization of notes between the two strings could be produced where the DNA sequence serves as a rhythmic percussion sequence.

Web Bibliography:

- 1) <http://www.ee.surrey.ac.uk/Personal/R.Knott/Fibonacci/fibrab.html>
- 2) <http://www.ee.surrey.ac.uk/Personal/R.Knott/Fibonacci/fibnat.html>
- 3) <http://www.ee.surrey.ac.uk/Personal/R.Knott/Fibonacci/fibFormula.html>

Other related resources:

<http://ep.llnl.gov/msds/music/midi-dna.html>

<http://www.math.holycross.edu/~davids/fibonacci/fibonacci.html>

<http://ulcar.uml.edu/~iag/CS/Fibonacci.html>

<http://pass.maths.org.uk/issue3/fibonacci/index.html>