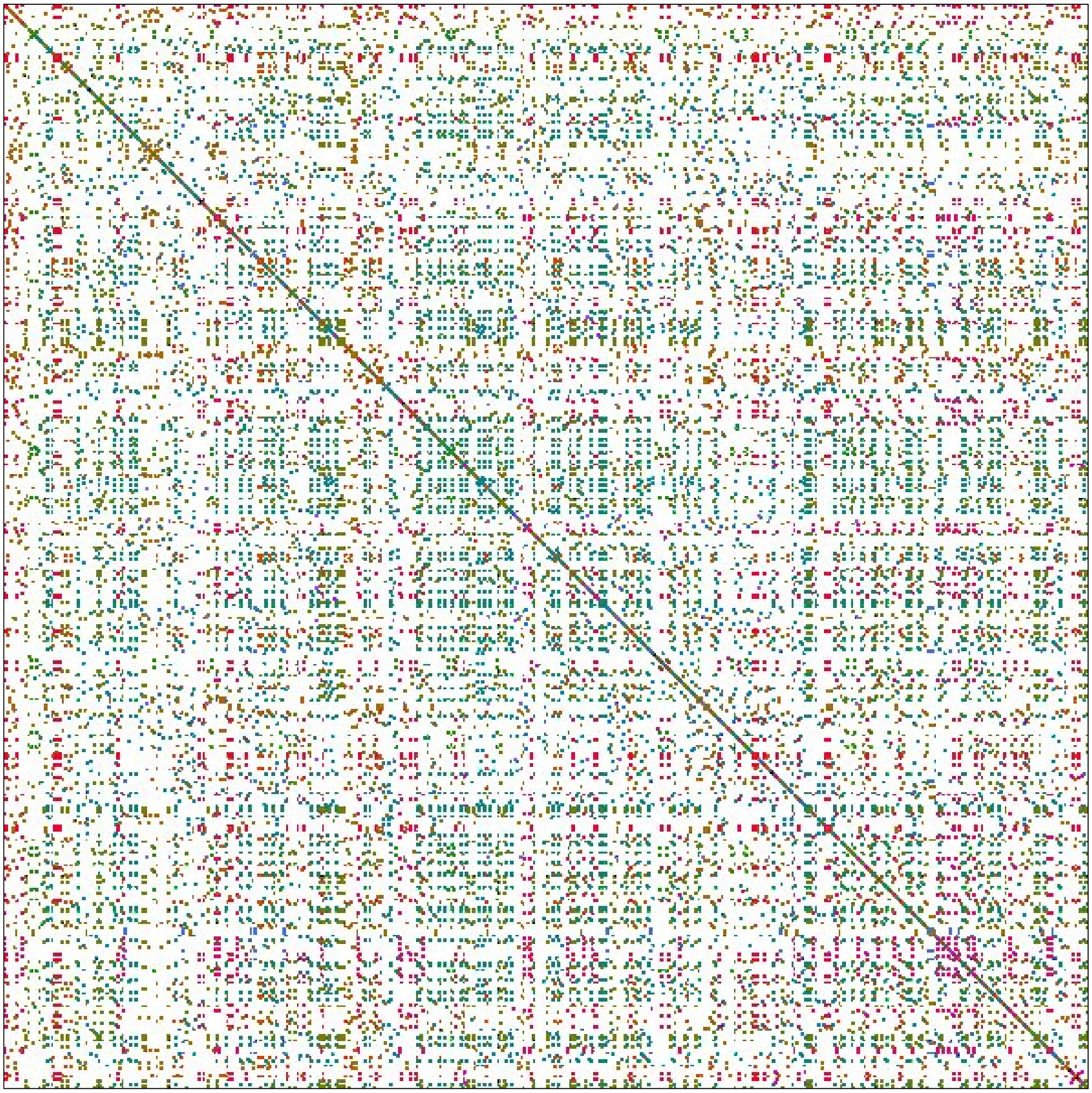
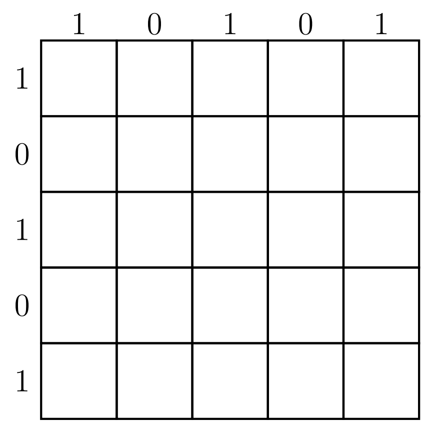
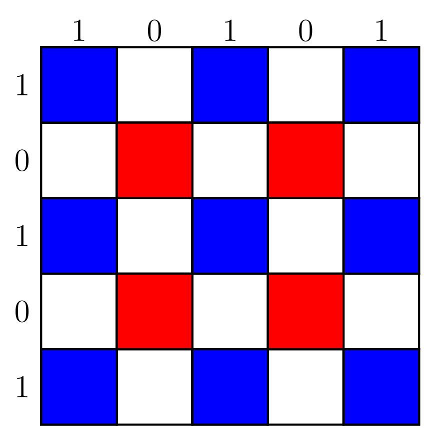


The diagram above pictorializes a self-similarity matrix applied to the phonemic transcription of Iago's speech to Roderigo in Act I of *Othello*. It provides a visual representation of repeated phonemes within the text. An explanation of the method occurs from page 2 to page 6. The explanation is readable, hopefully helpful, but ultimately skippable. Page 4 provides a small disclaimer on the method, specifically the phonemic transcription. Page 7 and 8 contain the analysis for the diagram. Page 9 provides some suggestions for future implementation.

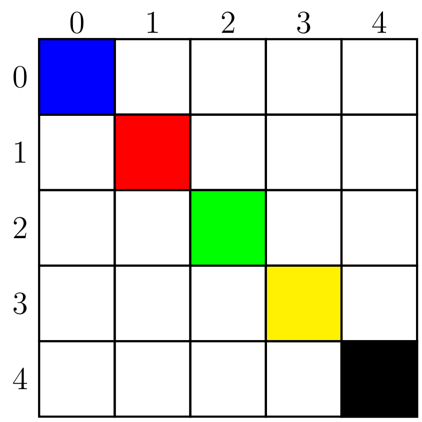
A self-similarity matrix maps recurrences within a particular string of data. For example, if the string were 10101, the self-similarity matrix would show that "1" repeats thrice, and "0," twice. So how does it do this. Well, imagine a square grid in which its length and width were determined by the number of terms within the string. Since our supposed string has 5 terms, we would have a 5x5 grid. Now, imagine that the string is laid out term-by-term outside the grid vertically and horizontally. Something like this:



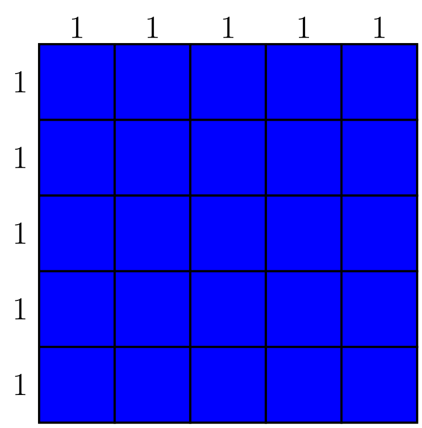
We can now give a name to every square within the grid. The top left square can be called (1,1), and the one to its right will be (1,0). The bottom right square will be (1,1). These names are not unique, and for our purposes, they do not need to be. We are interested in squares names that contain identical terms, like (0,0) and (1,1). Let us color in those squares. Let be (1,1) be blue and (0,0), red.



From the above image, let us derive a few heuristics for reading it. Notice that the diagonal will always be filled in, since each term is similar to itself. The grid is also symmetrical along the diagonal. The more the space outside the diagonal is filled-in, the more self-similar a string is. If the string contains zero self-similarity, grid will be filled-in only along the diagonal.

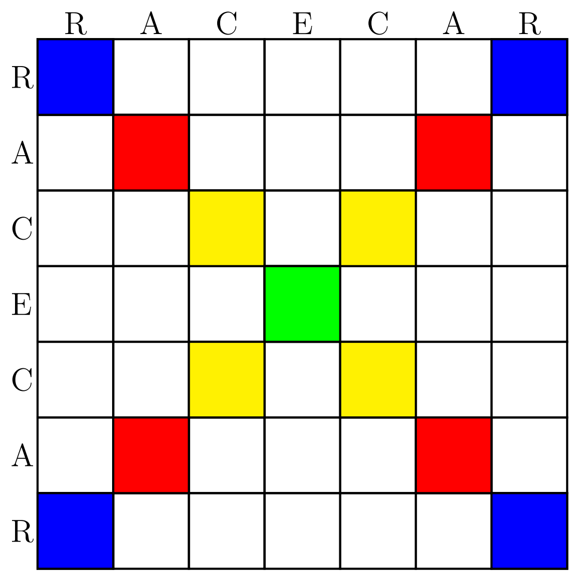


If a string is entirely self-similar, the entire grid will be filled-in.



That's the gist of the self-similarity matrix. The explanation is neither mathematically rigorous nor complete, but it will do for our purposes. Our toy examples are 5x5s but we can make the grid as large as we want. The Othello grid has about 700 terms. There are a few local patterns that emerge with longer strings, and thus larger grids, but more on that later.

So far we have generated a self-similarity matrix from a string of numbers, but we can use it to a string of anything, including words. For example, this is how the word RACECAR would look as a self-similarity matrix:

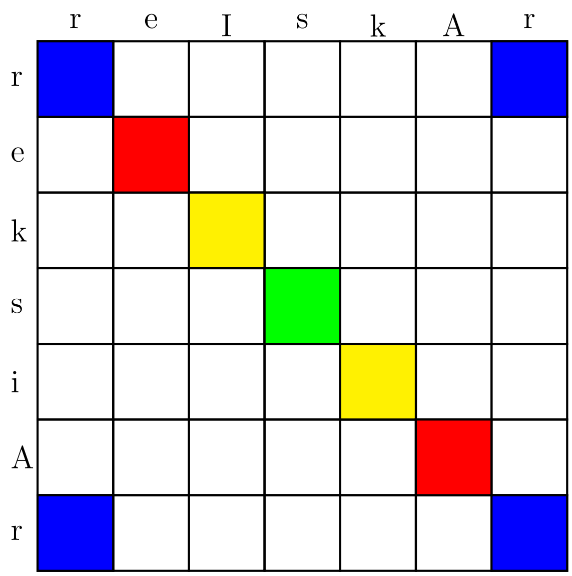


But we are interested in sound, so for this project, we are going to generate the self-similarity matrix from a phonemic transcription. And before we proceed any further, a few disclaimers are in order. This project transcribed words into the forty-four phonemes used in English. It might have been more precise to use the full-fledged IPA(International Phonetic Alphabet), but the IPA is too fine an instrument for our purposes, for it distinguishes not only how words sound but also how those sounds are produced. For example, IPA distinguishes the t-sounds of "stop" and "top" by how the tongue is placed differently when making them. I admit there is a minute difference in sound.

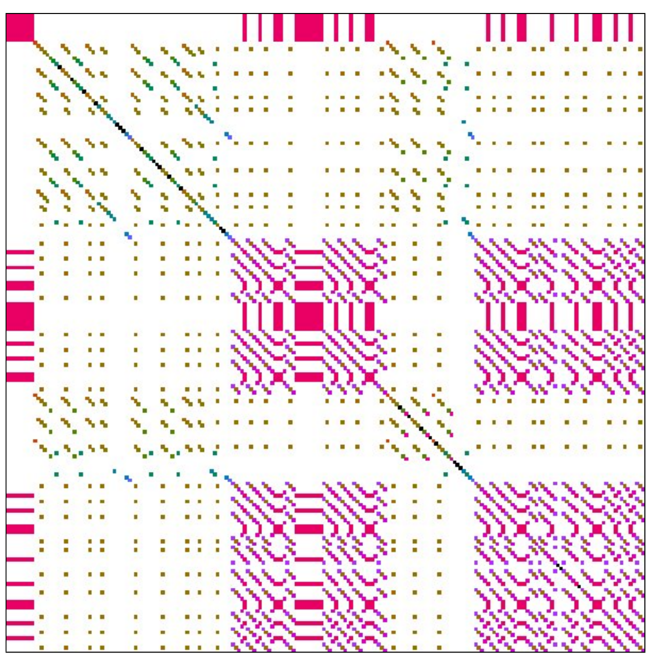
But if we want to map how many times we hear, not produce, a particular sound, such a distinction may be counterproductive to mapping our sense of repetition. The self-similarity matrix is by no means a catch-all of our hearing experience, but the IPA interferes with the aspect of hearing, repetition, that we want to capture.

In a similar vein, the stress marks have been omitted from the Othello transcription. Those are the important disclaimers. There are more. For example, spaces are also omitted, and the transcription was done by hand, so there are bound to be errors.

But let us first apply the tool to a phonemic transcription of RACECAR. We would have [reɪskaːr]. We no longer observe the palendromic pattern in the self-similarity matrix. Note that A stands for aː in the following figure. Pictures are harder to notate in phonemes.



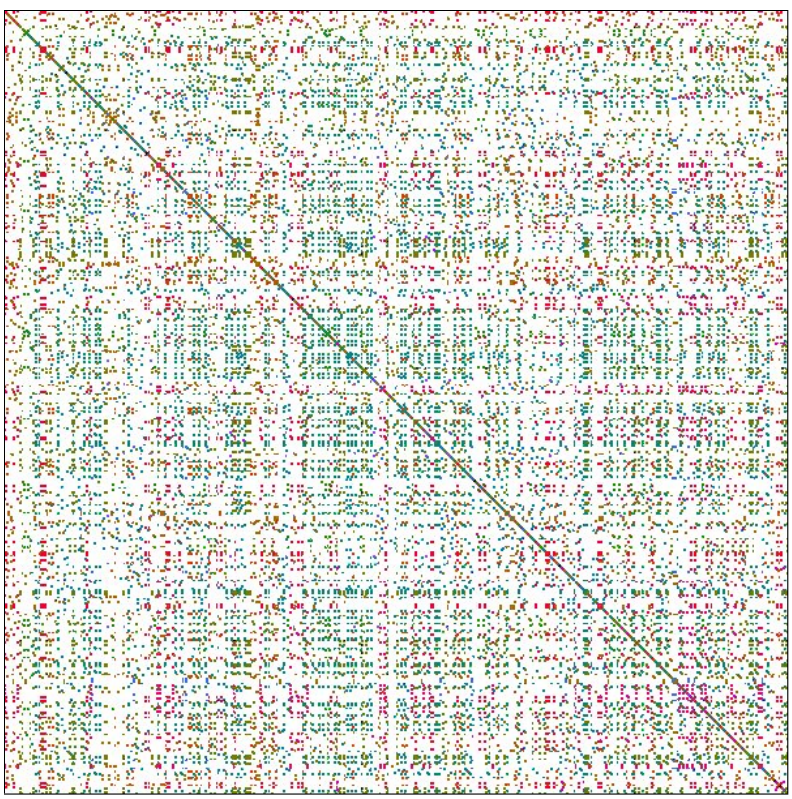
All of our examples have been short strings, single words. But what if they were much longer? Well, that would take us much longer. Thankfully, Colin Morris has developed Songsim, which measures by the word, the repetitiveness of song lyrics. Here is his rendition of “All You Need Is Love.” The full tool is available on his GitHub: http://colinmorris.github.io.



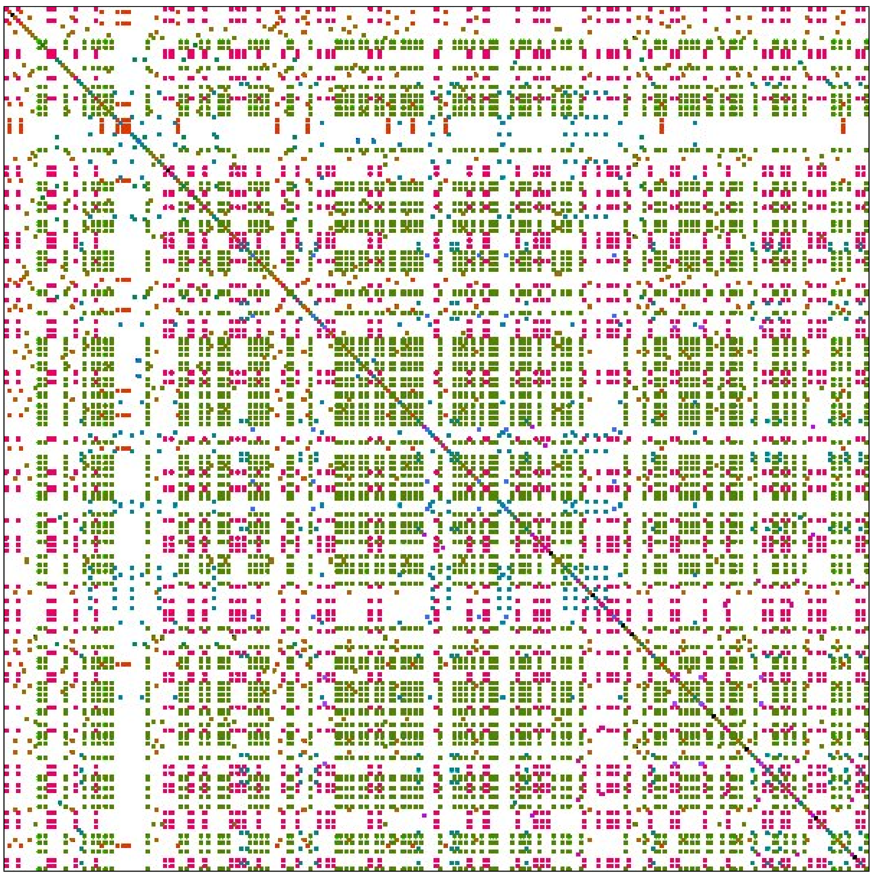
Morris developed Songsim to analyze the repetition of words and phrases, but the tool can be adapted to map the repetition of phonemes. We just apply the tool to the phonemic transcription of a text, like we did with RACECAR. And although we are interested in the repetition of phonemes and not words, there are a few observations that can be made about Moriss's work that we can carry over to our analysis of the Othello version.

We first notice the long diagonal that exists in every matrix, but we also notice some short ones. These short diagonals are strings of repetition that occur locally, which, in this case, is "all you need is love." We also notice a few solid squares. There are no gaps because these are repetitions of the same word: "love, love, love." The squares are mostly magenta and olive, corresponding to "love" and "you" respectively. The two are the most-used words in the song. There are also a few porous squares. These, though not as repetitive, include repetitive elements such as "you" or "love."

Let us now try to extend these observations—the small-diagonals, squares (solid and porous), and color prominence—to the Othello version reproduced below:

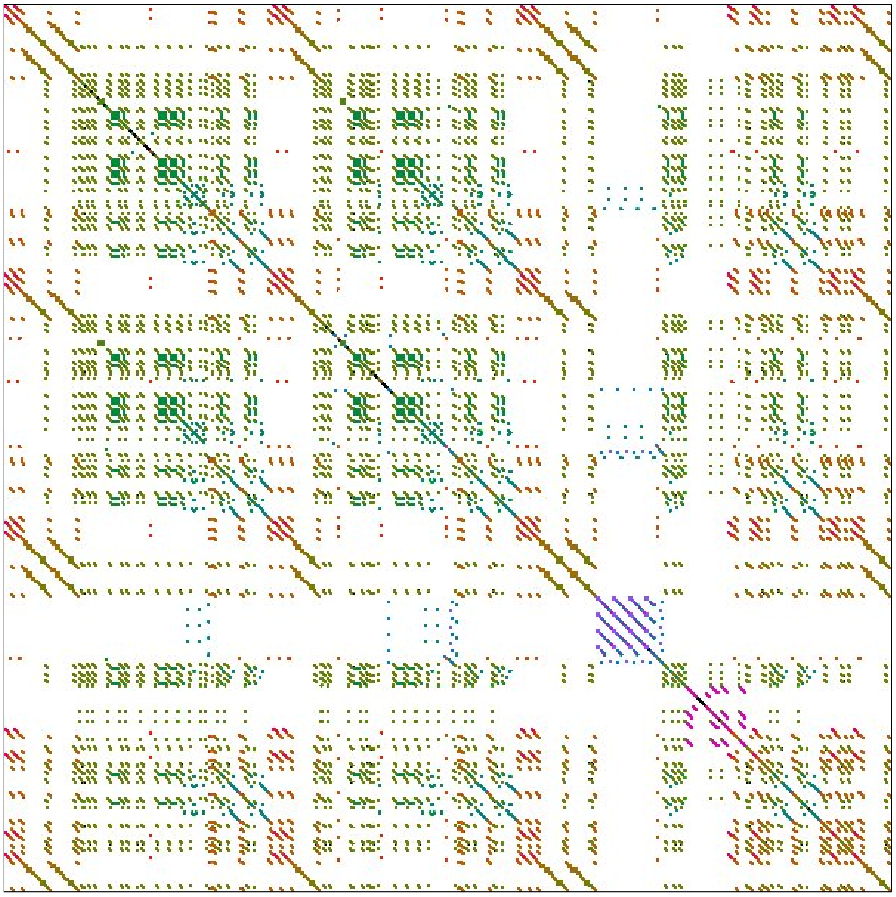


We first notice a prominence of turquoise, olive, and magenta. This is unsurprising considering that these correspond to "ə," "ŋ," and "ɪ," three frequently used phonemes. But we also notice porous turquoise squares with magenta dots filling the gutters between them. What does this mean? Well, at the expense on losing "ŋ," let us try to make the pattern more obvious by isolating the vowel sounds and generating a self-similarity matrix from the isolated string.



The pattern now looks closer to Morris's rendition of the Beatles. The repetitions seem to occur more frequently, and the squares are more closely-knit. Part of this is due to our having stripped the text of its consonants, so the pattern is a little artificial. But the pattern is not mere mathematical artifice. We see something similar to the tight square pattern in contemporary pop.

Below is Lady GaGa's "Bad Romance" done by Morris. Similar?



Now, we might care to remember that Morris is working with words whereas we are working with phonemes, so the repetition of "Rah rah ah-ah-ah" is not the same as that of "ə." Nonetheless, it might be still worth asking whether vowel repetition would contribute to a text sounding more like a song. "Bad Romance" makes constant use of "aː" with anaphoras of "I want" or words like "drama." Do vowel repetitions create a chorus-like quality?

The method was deliberately applied to Iago rather than, say, Desdemona to see whether such qualities could be identified in parts of the play that are not explicitly lyrical. In the speech, Iago tries to maintain the pentameter, at times, but the meter and stress is rather erratic. But there did seem to be a sense of order, one that I initially thought lay with the frequent "s" sounds in lines such as "But he; as loving his own pride and purposes." But the matrix locates that feeling rather with vowel repetition.

The self-similarity matrix, with all its limitations can show whether terms of a sequence repeat, when they repeat, and how frequently they repeat. The project made use of much of these properties, but it missed out on capitalizing an extra piece of information that the matrix provides: color. The colors in all of the matrices in this project were randomly assigned. But if we were to assign the colors according to how loud each phoneme is, then the matrix would double as a heat map showing which parts call for louder performances. If we could also assign colors according to how related two phonemes are, then the matrix might be more intuitive as chromatic similarity would imply phonemic similarity. We could even introduce IPA into the transcription, as the matrix would show through color that the different sounds may sound the same too us. The phonemes would all have to be tagged, but once they have been pre-processed, the tool could apply a filter and designate colors according to volume or phonemic similarity.

There is also the matter of size, noise, and signal. The Othello example stretched the tool the furthest by applying it to about 700 terms, but as we saw, this decreases the legibility of the matrix unless there is a stark and obvious pattern. We could devise a test that would determine which data to discard. It would measure how frequently a repetition occurs within a designated sub-section of the string, and if a repetitive term does not meet the standards, it would be considered insignificant. This might help parse the meaningful repetitions from the less important ones. Discarding less-meaningful data would allow us to apply the tool to a larger data set to see whether we could discern other patterns. I have a hypothesis that Iago sounds differently when he is lying and when he is telling the truth. Perhaps that might be something that could be explored with the above amendation.