Randomness classification

… fun guaranteed

Vladimir Matlach
Diego Krivochen
Jiri Milicka
Lukas Zamecnik

```
1000010100000001000011100000001101100100101001101110010101100011110001000101011101 1
1011111100010001000000000011111111100010010100011010101111111100101010101110101010 1
1000000010101001100010100111000111011001010110110111110011111101101000000011011001 10
1001110110101101000011111111101101100101000001011111110000011110101100011000100110 11
1010011011100111001011000110001000001100111110001011100111001110100011010011000010 10
1000101001110111011101101111100010111001001100111101010110011111011001110000000011 100
0110000111101100010001001011111010101000101001100000000110100001001110010101000011 01
1011001110011011111110001001001000110001101110110011011010000001110111110111110111 100
0011011001011001100001110011110000010111010110011111011110001101100100100000101010 101
1101001000111100000101110110100101010110000100110110110010110111100011111101011000 01
1111010010101001110010011100100100011100111100010000110110010100111110100010101011 011
0001100001111111010100110011100000111110110010011101110110100011000100100010111111 01
0001101101010010001101100011101111110011010111000110100110011010110010100110100100 00
1010100111101100001001000100011011101101010000111100000001000000000001001111101000 01
1111000001101000110011001011011101111101011110111111111010111110111111111011100111 111
1001000111100100110101011100101000011100110010011100100101011111011101000010100000 01
0010000110000100011011111000000001000100111100110100100010000011111001101100001101 01
1111111111001101000011111010001010010111010110100010110011111111010101001000111001 01
0111101110111010101011001010100010010110101010010101101101101001011110010001110011 00
1000011101001010001111110011101001101100011001010100101000001110100000001101101000 11
1001110010101110101110100001001100110110101010110011101000111110111100100010100111 10
1010011011001010010011011111101010001010011010111010000011101000010000000100111001 11
0001111111001001000111010100100100011110111110100011000001001010011110011111100100 00
```

… what if i told you

Randomness

… now we know

knowing anything about sequences that…

should be random

– or –

look random

is important

Randomness

What is randomness?

… lack of patterns and ***unpredictability***

… each token has the same probability to appear next in seq.

… no better prediction of next tokens than guess

… truly random string = no rules, no grammar

Randomness from linguist perspective

Non random strings have rules

… rules create patterns

… natural languages are not random strings

Linguistics has non-random sequences

… we naturally want to know the kind of rules behind it

It shows up that…

*… **quantification of some sequence properties** might answer the questions*

*… beyond linguistics, **various textual sequences** exists that we really like to understand*

the DNA

ACTTGTCAGTGCATATCGTAATCGTGTACGTATTGCAAGCGTTATCGGATATGC
GATCGGTAGCTGCTATCGATCGCGGTATTATATATCGGTGACTGCGATGATGCG
CGATTATCGGATCTAGCTAGCTGATTATCGATCAGTCTGATGCATGCGGCGCGT
ATATCTAGCGTATTTTTTCGATCGGCGTATGCATGCTGCATCGATCACGCATTAG
CATGTTGCCGGATTATCGATCGATCGATGCTGCATGTATAAATCGATCGGATATT
CGCGGCTATATTAGGCTATAGCTATGAGT…

… it is said that huge portion of DNA is just a random rubbish…

Voynich manuscript

… *very* ***famous***

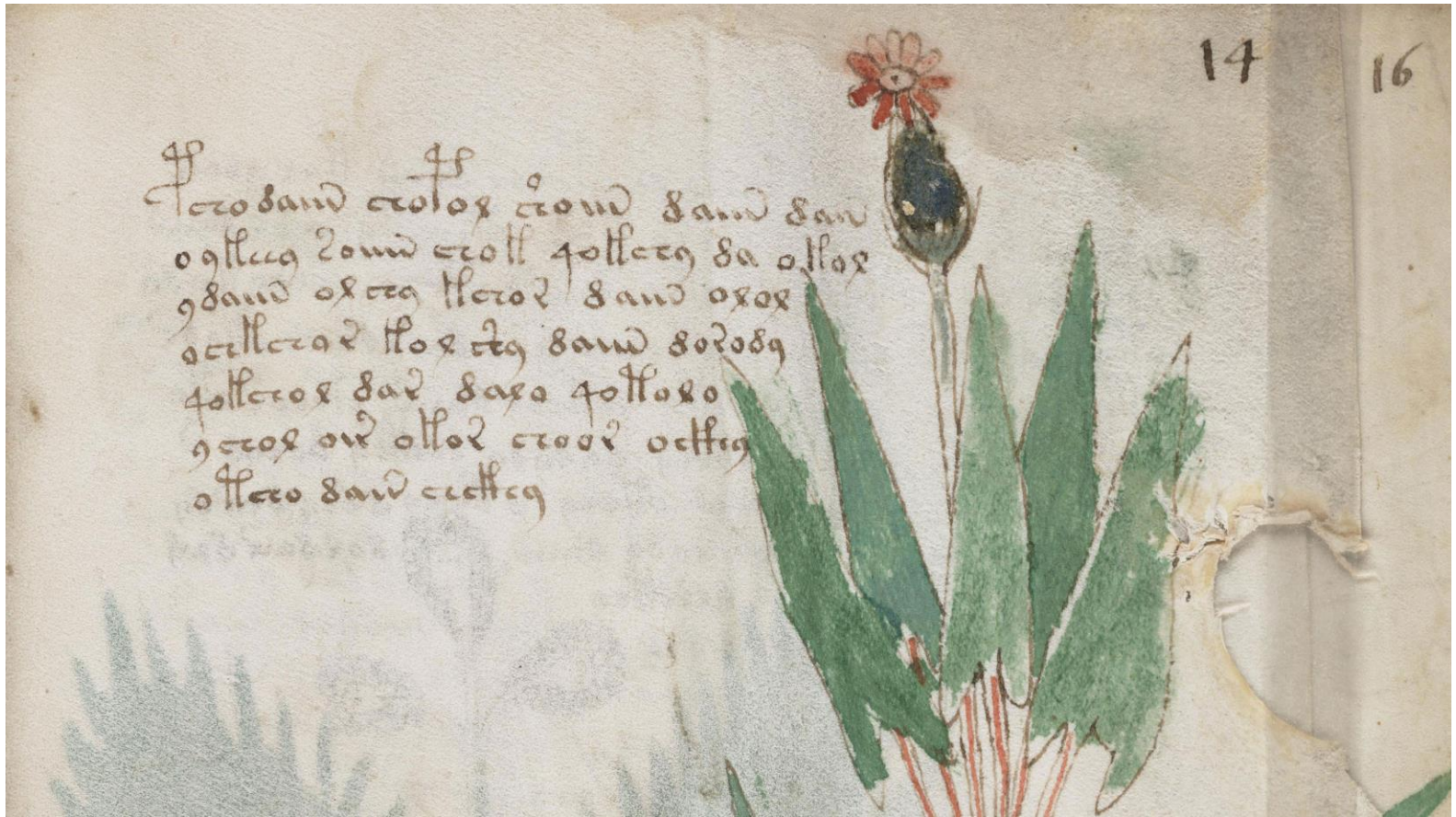
… *early* ***15th century*** *(carbon-dated)*

… ***a lot of $$$*** *from* ***Roman Emperor Rudolph II.***

… ***unknown language***

… ***might be encrypted***

… ***might be a fraud***



So we live in a world of…

unkwnown sequences

… that we really want to understand better

And also in the world where…

we pay a lot of money for the *real true random*

… b/c of banks, computer security, casinos

What we want?

We want to take this… *(click)*

We want to take this…

```
1000010100000001000011100000001101100100101001101110010101100011110001000101011110 11
1011111100010001000000000111111110001001010001101010111111110010101010111010101 01
1000000010101001100010100111000111011001010110110111110011111101101000000011011001 10
1001110110101101000011111111101101100101000001011111100000111101011000110001001101 1
1010011011100111001011000110001000001100111110001011100111001110100011010011000010 10
1000101001110111011101101111100010111001001100111101010110011111011001110000000011 100
0110000111101100010001001011111010101000101001100000000011010000100111001010100001 101
1011001110011011111110001001001000110001101110110011011010000001110111110111110111 100
```

And be able to answer

*… Are these sequences **random** & how much?*

… Encrypted?

… Monkey-typed?

… Natural Language look like?

… or some trivial repetitions?

And say:

„Yes, it is military hard encryption“

There sure exist

methods detecting *true random*

Like…

Diehard tests battery, Entropy, Chi-squared test, mean-test, Pi constant approximation with a needle etc.

But! … those methods answer only

„Is that sequence really perfect random?“

The rest of our questions are unanswered…

SO here comes our method that can answer them!

And it shows up, that it can be done…

with just the most **basic quantitative linguistics tools**…

… and programming.

All we need…

n-grams

&

type to token ratio

and programming.

How does the method work?

Method: *Simple but effective*

1. Take a **sequence**, in any language or of any type.

2. **Convert** all distinguishable symbols into **binary**

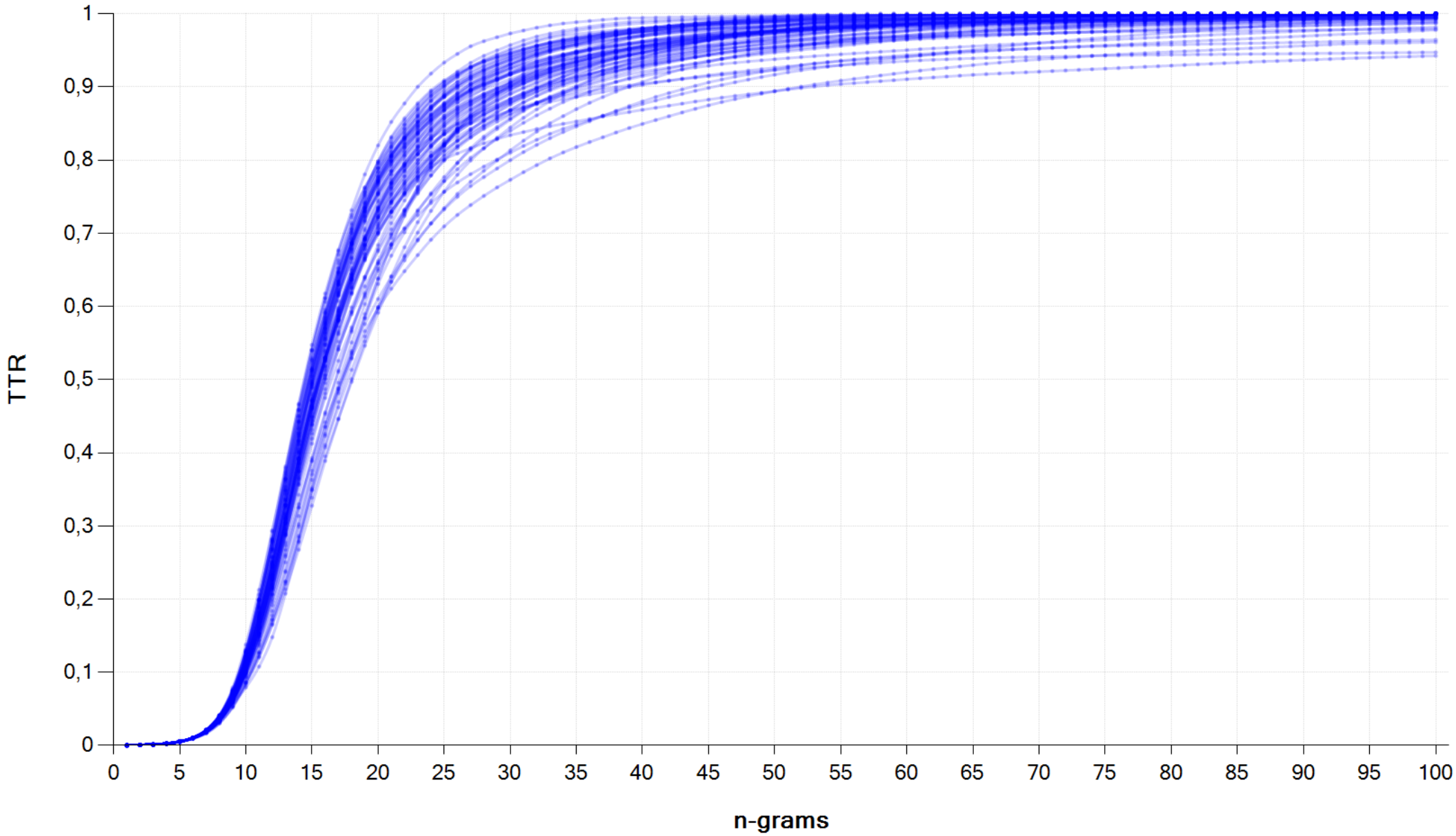
… new only binary sequence emerges

3. Take **n-grams** of the sequence:

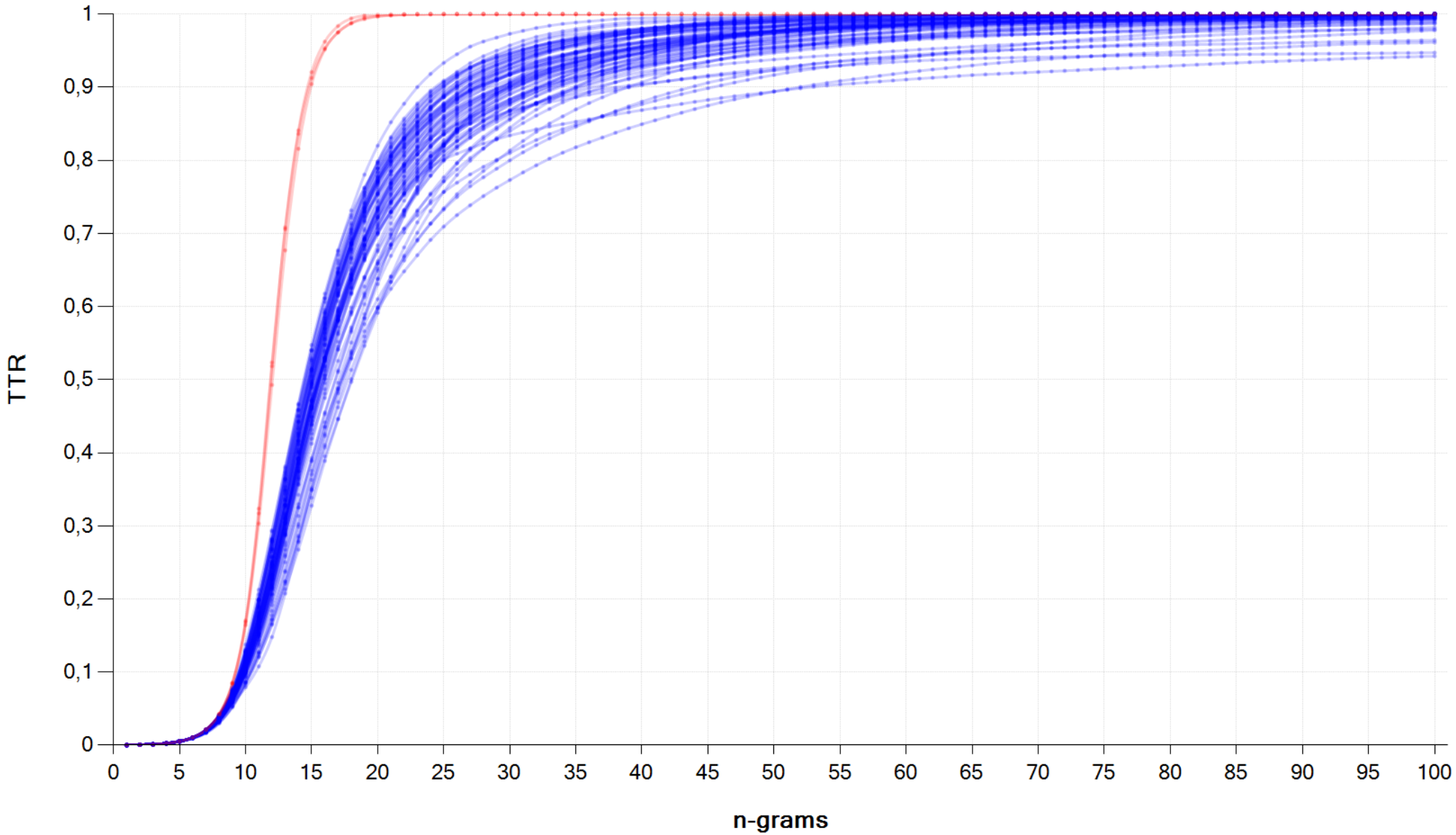
    Do for ***n***={2, k}, *where k is ½ of the sequence length:*

    Calculate Type to Token Ratio (TTR)

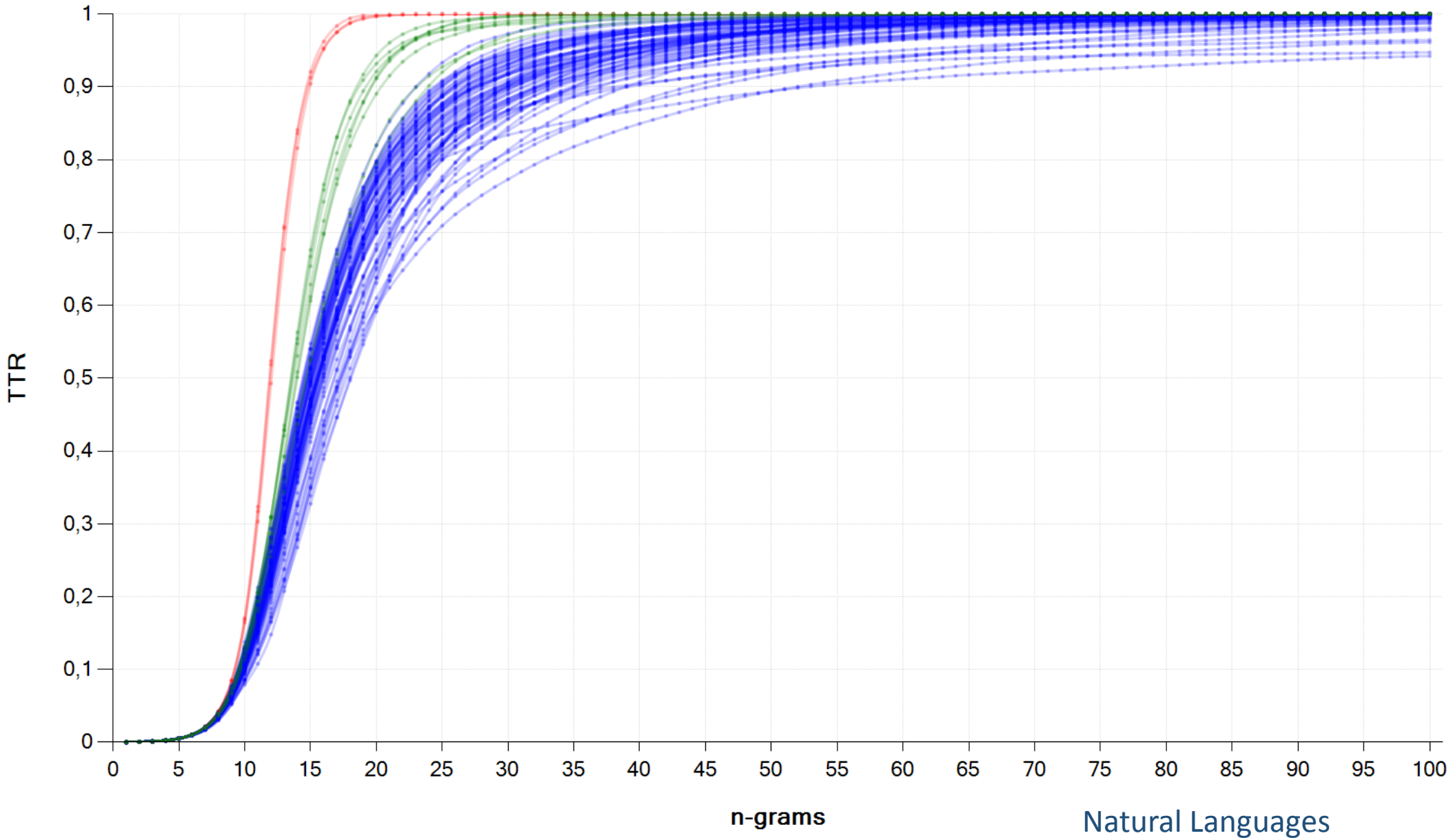
4. Plot **n-gram** size and calculated TTR values to chart



70 books, various languages



Random sequences from atmospheric noise

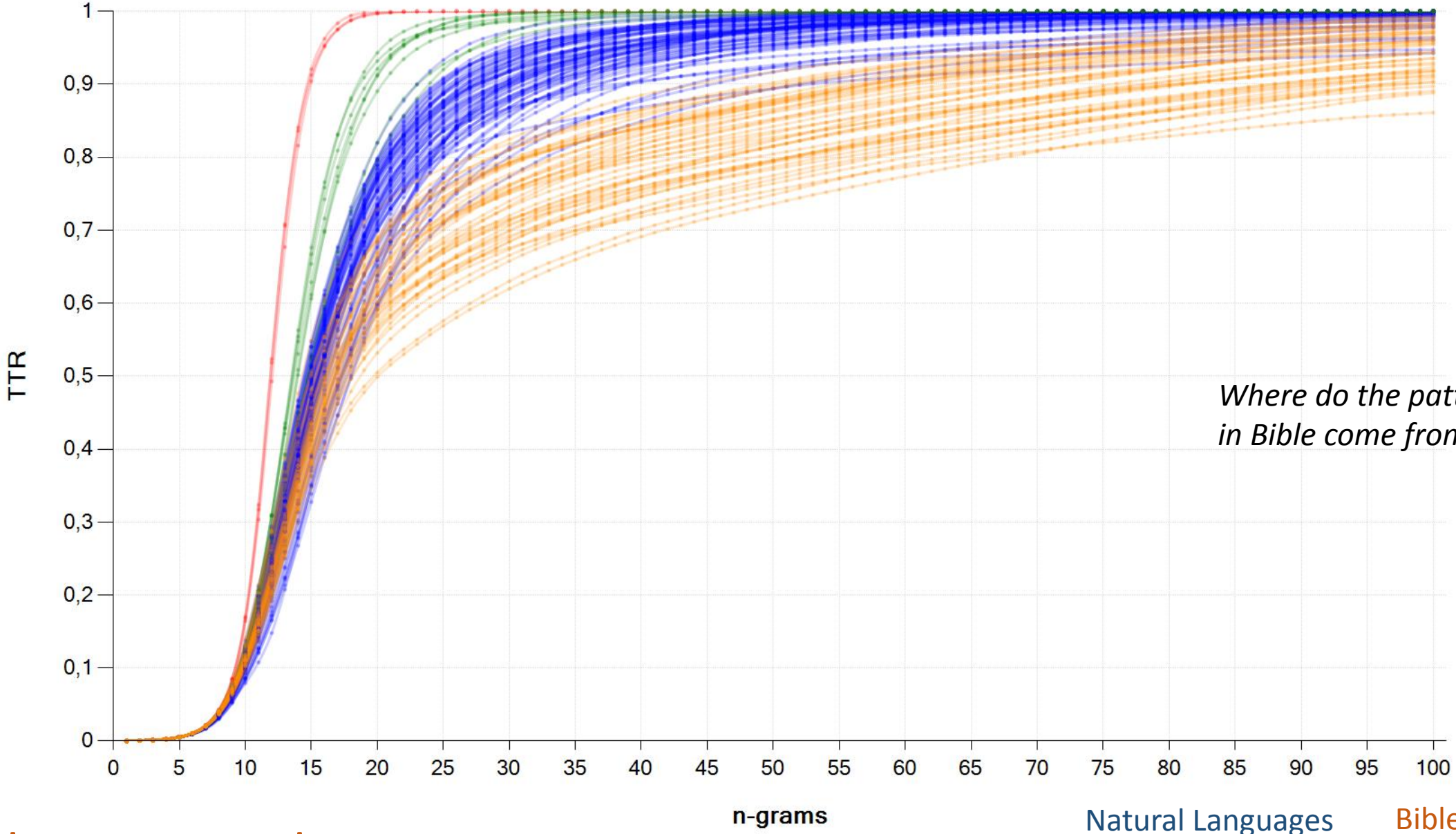


Monkey-typed texts

Natural Languages

True Random

Monkey-typed



Where do the patterns in Bible come from?

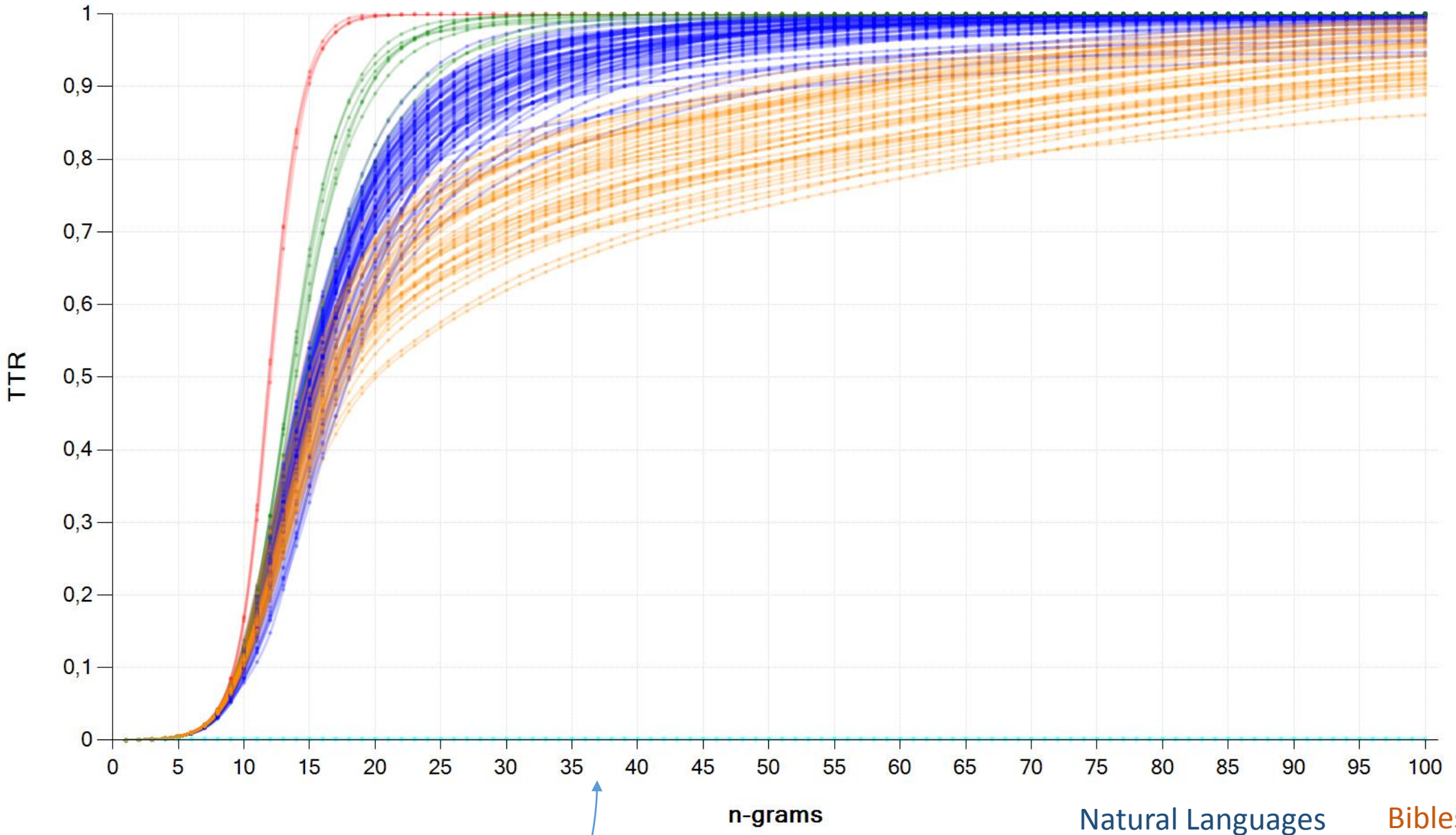
Bibles in 43 languages

Natural Languages

Bibles

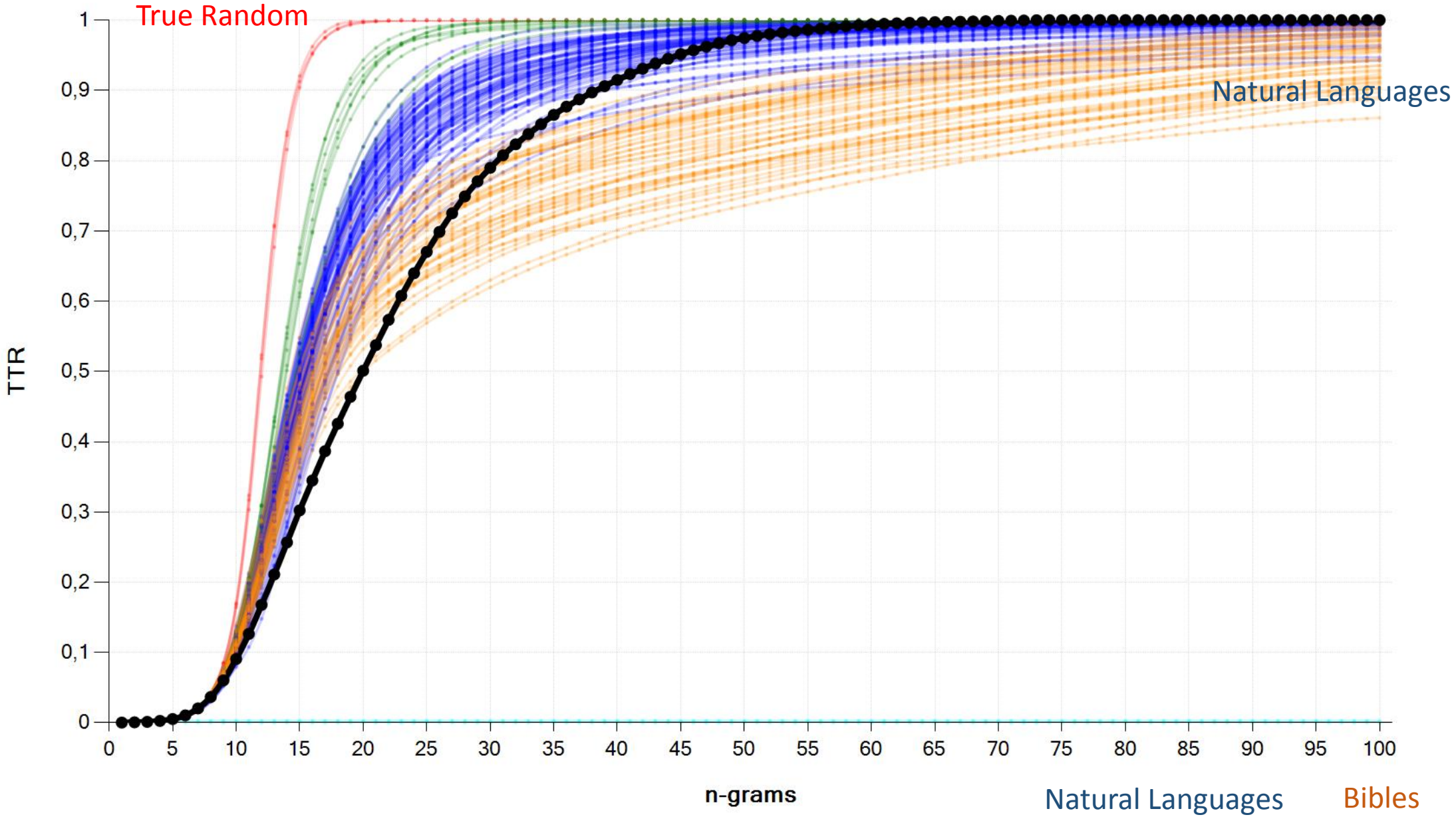
True Random

Monkey-typed



Only repetitions

Natural Languages
True Random
Monkey-typed
Bibles
Only repetitions



Voynich manuscript

Natural Languages
True Random
Monkey-typed
Bibles
Only repetitions
Voynich

That means:

We can, with some degree of probability…

determine the pattern nature of sequences.

From

… perfectly random sequences

… to encrypted data,

… monkey-typed,

… natural langage texts and

… trivial repetitions

We have also the next step method

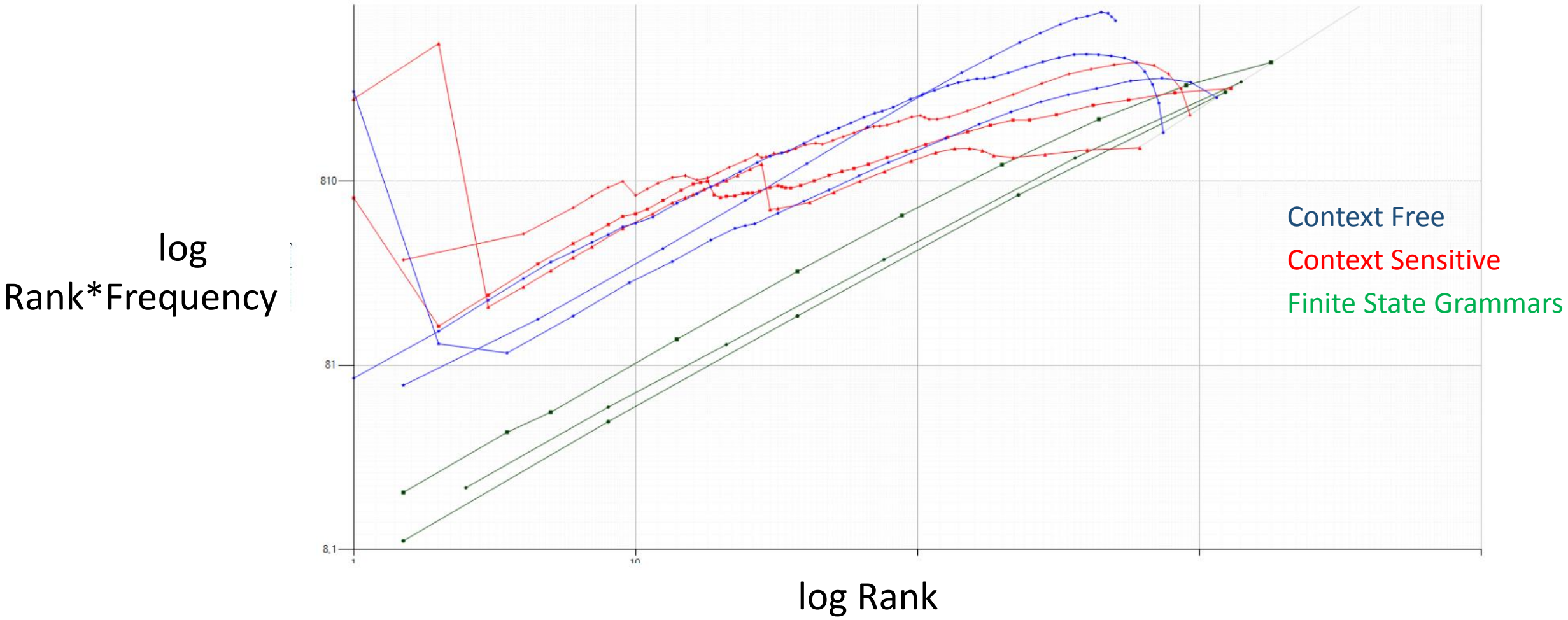
for random sequences

we can distinguish the complexity of generator used

… in terms of **Chomsky grammar hierarchy**
if the generator was like…

Final state machine
Context free machine
Context sensitive machine

… the method is based on Zipfian behavior of n-grams & needs more research



A preview: Stochastic generated sequences @ 14 n-gram

Plotted Zipfian behavior can be used for stochastic generator classification

Summary: What we have?

Possible decision tree distinguishing

True Random, Pseudo Random, Encryption,

Monkey Typing, Natural Languages , Natural Languages with added structures, Trivial repetitions

With a branch distinguishing

… complexity of pseudo-random generator

Summary: What‘s the difference with the other methods?

Other methods tend to **care only** for

randomness

&

not studying inner structure

Exceptions exist,

but no direct trend to understand „what‘s inside“

Applications

Linguistics

… analysing unknown sequences & languages

Computer security

… testing entropy sources in computer & classifying the genertor complexity

The end

Thank you for attention

Questions?