Parabola Volume 62 Issue 1 (2026), Article 7

Pattern occurrence and run enumeration in random binary strings

Onur Örnek[1]

1 Introduction

Random binary strings arise in many applications, from coding theory, bio-informatics to randomised algorithms [3, 4, 7]. In this article, we discuss two fundamental questions that often arise when considering random binary strings, namely the waiting time and location until a fixed pattern appears, as well as the enumeration and distribution of runs of a given length across all strings.

In Section 2, we derive the exact expressions for the occurrence probability, conditional expectation and variance of a given pattern. We also confirm these with Monte Carlo experiments. In Section 3, we obtain a closed-form expression for the total number of runs of length i via both combinatorial and indicator-variable arguments. We also prove that the run counts satisfy the Central Limit Theorem. We conclude with remarks on potential extensions, including patterns with overlaps, non-uniform biases and higher-order runs.

Studies of pattern matching and runs in random sequences span several classic approaches. Finite automata and generating function techniques yield formulas and series expansions for waiting times and occurrence probabilities of patterns with possible overlaps [4, 7]. Markov chain embeddings lead to efficient algorithmic recurrences for run and pattern statistics. These are widely used to compute exact distributions and tail probabilities [3]. For the *longest run* problem, classical bounds and approximations go back at least to [6], while standard references such as [2] survey foundational tools.

Within this landscape, our contribution provides self-contained derivations for the expected waiting time and first-occurrence location for fixed patterns, as well as the exact enumeration and an asymptotic normality result for run counts. We also provide simple simulation codes to support our derivations.

[1]Onur Örnek is a student at Üsküdar American Academy. Email: `oornek26@my.uaa.k12.tr`.

2 Pattern occurrence in random binary strings

2.1 Occurrence probability and expected waiting time

Fix positive integers m and n with $m \le n$. Let

$$S = (S_1, S_2, \ldots, S_n)$$

be a sequence of events taking values in $\{0, 1\}$, where $S_1, S_2, \ldots, S_n$ are independent and identically distributed (iid) with

$$\Pr(S_i = 0) = \Pr(S_i = 1) = \frac{1}{2}, \quad 1 \le i \le n\,.$$

We ask about the occurrence of the pattern $E = (E_1, E_2, \ldots, E_m)$ as a contiguous block in S.

At any given starting position k, with $1 \le k \le N := n - m + 1$, the chance that

$$E = (S_k, S_{k+1}, \ldots, S_{k+m-1})$$

is $p = 2^{-m}$. If we let $q = 1 - p$, then the probability that E *never* appears in S is

$$\Pr(E \text{ does not occur}) = q^N = (1 - 2^{-m})^{n-m+1}$$

and therefore

$$\Pr(E \text{ occurs at least once}) \ = \ 1 - q^N\,.$$

For example, if we take $m = 10$ and $n = 100$, then $N = n - m + 1 = 91$, $p = 2^{-10}$ and

$$1 - q^{91} = 1 - (1 - 2^{-10})^{91} \approx 0.0851\,.$$

If we imagine generating an infinite stream of iid bits and ask when E first appears, then the waiting time X has a geometric distribution with parameter p, so the expected waiting time is $\mathbb{E}(X) = \frac{1}{p} = 2^m$ which is independent of n.

We can test this computationally with the following Python code for a binary pattern of length 10. The code generates a random 10-length binary pattern and generates randomly a binary string until the pattern is found:

```
import random

e = ""
m = 10
for i in range(m):
  e += str(random.randint(0,1))

found = []
for x in range(10000):
  s = ""
```

```
  while (1 == 1):
    s += str(random.randint(0, 1))
    index = s.find(e)
    if index == -1:
      pass
    else:
      found.append(index)
      break

print(sum(found)/len(found))
```

As an illustration, one simulation yields an empirical mean of 1023.58002 which is in close agreement with the theoretical expectation of $2^{10} = 1024$.

2.2 Conditional first-occurrence position

Let $K \in \{1, 2, \ldots, N\}$ be the position of the first occurrence of E in a random string of length n, with

$$\Pr(K = k) = q^{k-1} p, \qquad 1 \le k \le N\,.$$

Conditioning on the event that E does occur, which is equivalent to saying that $K \le N$, the conditional expected position is

$$\begin{aligned}
\mathbb{E}\big(K \mid K \le N\big) &= \sum_{k=1}^{N} k \; \Pr(K = k \mid K \le N) \\
&= \sum_{k=1}^{N} k \, \frac{\Pr(K = k)}{\Pr(K \le N)} \\
&= \frac{p}{1 - q^N} \sum_{k=1}^{N} k q^{k-1}\,. \qquad (1)
\end{aligned}$$

Similarly, one may also compute

$$\mathbb{E}\big(K^2 \mid K \le N\big) = \frac{p}{1 - q^N} \sum_{k=1}^{N} k^2 q^{k-1}$$

in order to obtain the conditional variance

$$\begin{aligned}
\mathrm{Var}(K \mid K \le N) &= \mathbb{E}(K^2 \mid K \le N) - \mathbb{E}(K \mid K \le N)^2 \\
&= \frac{p}{1 - q^N} \sum_{k=1}^{N} k^2 q^{k-1} - \left(\frac{p}{1 - q^N} \sum_{k=1}^{N} k q^{k-1} \right)^2 . \qquad (2)
\end{aligned}$$

For the computational implementation, below is the Python script used to perform these numerical evaluations with $n = 100$ and $m = 10$. The code searches for the previously generated 10-length pattern E in 10000 randomly generated 100-length strings:

```
n=100
found = []
all = []
for x in range(10000):
  s = ""
  for i in range(n):
    s += str(random.randint(0,1))

  index = s.find(e)
  all.append(index)

  if index != -1:
    found.append(index)
print("What percent of strings contained e:
    \n%", 100*len(found)/len(all))
print("Average point at which e was found:
    \n", sum(found)/len(found))
```

Monte Carlo simulations done over 10000 trials yielded an empirical mean $\overline{K}$ for the first-occurrence position and an empirical occurrence probability $\overline{p}$, where

$$\overline{K} \approx 43.652, \quad \overline{p} \approx 0.0882\,.$$

These values are close to the theoretical conditional expectation $\mathbb{E}(K \mid K \leq 91) \approx 45.326$ and the theoretical occurrence probability $1-(1-2^{-10})^{91} \approx 0.0851$, with any remaining discrepancy attributable to sampling variability in the Monte Carlo estimates. In our numerical example, $\mathrm{Var}(K \mid K \leq 91) \approx 689.7273$.

As a remark, there is a natural generalisation if the bits are replaced by symbols from an alphabet of size $\sigma \geq 2$. In this case, $p = \sigma^{-m}$, $q = 1 - \sigma^{-m}$ and $N = n - m + 1$, and formula (1) holds unchanged.

3 Enumeration of runs in binary strings

A *run* of length i in a binary string is a maximal contiguous block of i identical bits. Let $r_n(i)$ be the total number of runs of exactly i ones, summed over all 2^n binary strings of length n. More precisely,

$$r_n(i) = \sum_{\mathsf{s}\in\{0,1\}^n} \#\{\text{runs of exactly } i \text{ ones in } \mathsf{s}\}\,.$$

We wish to derive a recurrence relation for $r_n(i)$. First, we note that if $i = n$, then clearly only the all-ones string contributes one run of length n, so $r_n(n) = 1$. If $i = n-1$, then there are exactly two strings $\mathtt{011}\cdots\mathtt{1}$ and $\mathtt{111}\cdots\mathtt{10}$ with one run of ones of length $n-1$; therefore, $r_n(n-1) = 2$.

Theorem 1. *Let $n \geq 3$. For each $i \in \{1, \ldots, n-2\}$,*

$$r_n(i) = 2r_{n-1}(i) + 2^{n-i-2}\,. \tag{3}$$

Proof. Partition all 2^n binary strings of length n according to their first bit.

Case 1: The first bit is $\mathtt{0}$.
There are 2^{n-1} strings beginning with 0 which are of the form $\mathtt{s} = \mathtt{0t}$ with $\mathtt{t} \in \{\mathtt{0}, \mathtt{1}\}^{n-1}$. Then every run of exactly i ones in $\mathtt{s}$ lies entirely inside $\mathtt{t}$ and the total contribution from these strings is

$$\sum_{t \in \{\mathtt{0},\mathtt{1}\}^{n-1}} \#\{\text{runs of length } i \text{ in } \mathtt{t}\} = r_{n-1}(i)\,.$$

Case 2: The first bit is $\mathtt{1}$.
Again there are 2^{n-1} such strings with this condition. Write any such string as $\mathtt{s} = \mathtt{1u}$, with $\mathtt{u} \in \{\mathtt{0}, \mathtt{1}\}^{n-1}$. We can further divide this case into two subcases:

If a new run of exactly i ones starts at the first position, then the first i bits of $\mathtt{s}$ are all $\mathtt{1}$, immediately followed by a $\mathtt{0}$ at position $i+1$. Such strings have the form

$$\mathtt{11}\cdots\mathtt{10w}, \quad \mathtt{w} \in \{\mathtt{0}, \mathtt{1}\}^{n-i-1}$$

and there are 2^{n-i-1} choices for the tail $\mathtt{w}$.

Otherwise, no run of ones of length i starts at the first position. In that case, such a run of exactly i ones in $\mathtt{s}$ lies entirely in $\mathtt{u}$ and there are $r_{n-1}(i) - 2^{n-i-2}$ such strings in total.

From the two cases above, we have the recurrence relation

$$r_n(i) = r_{n-1}(i) + \left(2^{n-i-1} + r_{n-1}(i) - 2^{n-i-2}\right) = 2r_{n-1}(i) + 2^{n-i-2}$$

for $1 \leq i \leq n-2$. □

We now finish this section by obtaining a closed-form solution for $r_n(i)$.

Theorem 2. *For all non-negative integers $n \geq 3$ and i such that $1 \leq i \leq n$, we have that*

$$r_n(i) = \begin{cases} (n-i+3)2^{n-i-2} & \textit{if } i \leq n-1\,; \\ 1 & \textit{if } i = n. \end{cases}$$

Proof. Take an arbitrary non-negative integer $n \geq 3$ and fix $i \in \{1, 2, \ldots, n-2\}$. Set

$$b_n = \frac{r_n(i)}{2^n}$$

so we obtain a first-order recurrence relation for b_n from (3), namely

$$b_n = \frac{r_{n-1}(i)}{2^{n-1}} + 2^{-i-2} = b_{n-1} + 2^{-i-2}\,.$$

By a telescoping argument, we get that

$$b_n = b_{i+1} + \sum_{k=i+2}^{n} 2^{-i-2} = 2^{-i} + (n-i-1)2^{-i-2}\,,$$

since $r_{i+1}(i) = 2$ implies $b_{i+1} = 2/2^{i+1} = 2^{-i}$. Hence,

$$r_n(i) = 2^n b_n = 2^n\big(2^{-i} + (n-i-1)2^{-i-2}\big) = (n-i+3)2^{n-i-2}\,.$$

One can check that this formula still holds true when $i = n-1$ since $r_n(n-1) = 2$. Together with the boundary value $r_n(n) = 1$, this finishes the proof. □

4 Expected number of runs of fixed length

As an application of our enumeration formula, we now derive the expected number of runs of exactly i ones in a uniformly random binary string of length n, using a purely probabilistic argument. In particular, we recover the same closed-form expression that arose from the combinatorial enumeration.

Theorem 3. *Let $S = (S_1, S_2, \ldots, S_n)$ be drawn uniformly from $\{\mathtt{0}, \mathtt{1}\}^n$, and let*

$$X_i \;:=\; \#\big\{\textit{runs of exactly } i \textit{ consecutive ones in } S\big\}\,.$$

Then

$$\mathbb{E}(X_i) = \frac{r_n(i)}{2^n} = (n-i+3)2^{-i-2}\,.$$

Proof. Label the $n-i+1$ possible starting positions for an i-run of ones by the integer $j \in \{1, 2, \ldots, n-i+1\}$. For each such j, define the indicator function

$$I_j = \begin{cases} 1 & \text{if } S_j S_{j+1} \cdots S_{j+i-1} = \mathtt{1}^i \text{ and it is a maximal block of length } i\,; \\ 0 & \text{otherwise.} \end{cases}$$

We observe that

$$X_i = \sum_{j=1}^{n-i+1} I_j\,, \qquad \mathbb{E}(X_i) = \sum_{j=1}^{n-i+1} \mathbb{E}(I_j) = \sum_{j=1}^{n-i+1} \Pr(I_j = 1)\,.$$

We distinguish three cases for a run of ones of length i:

If $j = 1$, then this implies $S_1 S_2 \cdots S_i = \mathtt{1}^i$ and $S_{i+1} = 0$. This occurs with probability $\Pr(I_1 = 1) = 2^{-i-1}$.

If $j + i - 1 = n$, then this implies $S_{n-i+1} \cdots S_n = \mathtt{1}^i$ and $S_{n-i} = 0$. Again, this occurs with probability $\Pr(I_{n-i+1} = 1) = 2^{-i-1}$.

If $2 \le j \le n-i$, then we require $S_{j-1} = 0$, $S_j \cdots S_{j+i-1} = 1^i$, and $S_{j+i} = 0$. We have a total of $i+2$ specified bits, each equally likely, so $\Pr(I_j = 1) = 2^{-i-2}$ for each $j \in \{2, \dots, n-i\}$ and there are $(n-i-1)$ such positions.

From these cases, we see that

$$\mathbb{E}(X_i) = 2 \times 2^{-i-1} + (n-i-1)2^{-i-2} = (n-i+3)2^{-i-2} = \frac{r_n(i)}{2^n}$$

as claimed. Note that this agrees with our combinatorial formula. □

5 Asymptotic normality of run counts

We now show that for any fixed run length i, the random variable

$$X_i = \#\{\text{runs of exactly } i \text{ ones in } S\}$$

in a uniformly chosen binary string $S \in \{0,1\}^n$ satisfies the Central Limit Theorem as $n \to \infty$.

Theorem 4. *For each fixed $i \ge 1$,*

$$\frac{X_i - \mathbb{E}(X_i)}{\sqrt{\mathrm{Var}(X_i)}} \xrightarrow{d} \mathcal{N}(0,1) \quad \textit{as } n \to \infty.$$

Proof. Recall from Section 4 that $X_i = I_1 + I_2 + \cdots + I_{n-i+1}$ where

$$I_j = \begin{cases} 1 & \text{if a run of exactly } i \text{ ones begins at position } j\,; \\ 0 & \text{otherwise.} \end{cases}$$

Since the event "run of length i that starts at j" depends only on bits $S_{j-1}, S_j, \dots, S_{j+i}$, the indicator functions I_j and I_k are independent whenever $|j-k| > i+1$. Thus, $\{I_j\}$ is an $(i+1)$-dependent sequence. Each I_j has finite variance and one can check that

$$\sum_{k=-\infty}^{\infty} \mathrm{Cov}(I_1, I_{k+1}) < \infty\,.$$

By the Central Limit Theorem for m-dependent sequences [5], the normalised sum converges in distribution to the standard normal. □

6 Conclusion and potential future work

We have derived exact formulas for the occurrence probability, conditional expectation and variance of a fixed pattern in a random binary string. We have also obtained a neat closed-form for the total number of runs of a given length across all binary strings of length n. These results illustrate classical uses of geometric series and simple recurrences in probabilistic and enumerative combinatorics.

Our analysis naturally extends to alphabets of size $a \geq 2$, where pattern occurrence probabilities are adjusted by a^{-m} and overlap structures generalise via de Bruijn graphs on a symbols. For the distribution of the *maximum run length* in larger or even infinite alphabets, two directions look promising.

First, Markov chain embeddings and automata-based generating functions can still be deployed to derive exact recurrences and series approximations for run statistics, albeit with higher state dimensions. Second, models with an *infinite* alphabet connect to discovery problems (waiting times until a new distinct value appears) and to applications in genetics and machine learning (e.g., clustering/classification with unbounded label sets), suggesting potential links between occupancy schemes and run-length distributions; see [2] for classical tools that may be adapted in this setting. We leave a systematic treatment of these generalisations for future work.

Acknowledgments

The author thanks the Summer Research School 2025 for providing the opportunity to conduct this project and gratefully acknowledges the guidance and support of mentor Deyan Bozhkov throughout the research process.

References

[1] F. Balado and G. Silvestre, Runs of ones in binary strings, arXiv:2302.11532, 2023.

[2] W. Feller, *An Introduction to Probability Theory and Its Applications*, Wiley, New York, 1971.

[3] J.C. Fu and M. V. Koutras, Distribution theory of runs: A Markov chain approach, *Journal of the American Statistical Association* **89** (1994), 1050–1058.

[4] L.J. Guibas and A. M. Odlyzko, String overlaps, pattern matching, and nontransitive games, *Journal of Combinatorial Theory, Series A* **30** (1981), 183–208.

[5] I.A. Ibragimov and Y.V. Linnik, *Independent and Stationary Sequences of Random Variables*, edited by J.F.C. Kingman, Wolters-Noordhoff, Groningen, 1971.

[6] M.F. Schilling, The longest run of heads, *The College Mathematics Journal* **21** (1990), 196–207.

[7] V.T. Stefanov, S. Robin and S. Schbath, Waiting times for clumps of patterns and for structured motifs in random sequences, *Discrete Applied Mathematics* **155** (2007), 868–880.