

**Statistics, Probability, and Algorithms in  
Bioinformatics (SPAß)  
[DRAFT - Do not distribute]**

Edited by Sven Rahmann

Lecture Notes  
Winter Semester 2003/2004

Fakultät für Mathematik und Informatik  
Freie Universität Berlin



# Contents

<b>1</b>	<b>Probability Distributions and Random Variables</b>	<b>5</b>
1.1	Probability Spaces . . . . .	5
1.2	Conditional Probabilities . . . . .	6
1.3	Random Variables . . . . .	7
1.4	Moments . . . . .	8
<b>2</b>	<b>Computer Representation of Probabilities and Distributions</b>	<b>9</b>
2.1	Parametric distributions . . . . .	10
2.2	Non-parametric distributions . . . . .	11
2.3	Mixture distributions . . . . .	12
2.4	The IEEE Floating Point Standard . . . . .	12
<b>3</b>	<b>Important Parametric Distributions</b>	<b>15</b>
3.1	Discrete Distributions . . . . .	15
<b>4</b>	<b>Generation of Random Numbers</b>	<b>19</b>
4.1	Pseudo-random numbers . . . . .	19
4.2	Uniform Random Number . . . . .	19
4.3	Non-Uniform Random Variates . . . . .	22
<b>5</b>	<b>Principle of Accept-Reject Methods</b>	<b>23</b>
5.1	Introduction . . . . .	23
5.2	Truncated normal distribution . . . . .	25
5.3	Gamma r.v. with non-integer shape parameter . . . . .	25
<b>6</b>	<b>Tests for Random Number Generation</b>	<b>27</b>
6.1	Chi-square test . . . . .	27
<b>7</b>	<b>Understanding Alignment Statistics</b>	<b>29</b>
7.1	Introduction . . . . .	29
7.2	Distribution of Optimal Local Alignment Scores . . . . .	29
<b>8</b>	<b>Understanding Alignment Statistics [2]</b>	<b>33</b>
8.1	Distribution of the Optimal Alignment Score . . . . .	33

<b>9</b>	<b>Monte Carlo Markov Chain Methods (MCMC)</b>	<b>37</b>
9.1	Introduction . . . . .	37
9.2	The Metropolis Algorithm (1953) . . . . .	40
9.3	The Hastings Algorithm (1970) . . . . .	40
9.4	The Tierney Algorithm (1974) . . . . .	41

# Introduction

As the title indicates, this lecture series is about *statistics, probability, algorithms, bioinformatics*. Additionally, everyone taking this class should also have *fun* learning something new.

We will emphasize the interplay between algorithms and probabilistic techniques, and especially their applications in bioinformatics. This is *not* a lecture about randomized algorithms. Some specific problems that we will encounter in this lecture are the following ones, in no particular order.

- Probability distributions must somehow be represented in the computer. This can be done in many different ways, and we will consider a variety of them.
- One way of representing a distribution is as a mixture of other (simpler) distributions. We will look into methods for determining the components of a mixture and the mixture coefficients.
- To find the probability of (composite) events, we usually have to sum several probabilities or integrate the density function. Especially for multivariate distributions, this turns out to be a non-trivial task.
- Many issues arise from the finite precision of floating point numbers, leading to roundoff errors or to the impossibility to represent very small probabilities. We will look at ways to avoid these problems.
- For simulations, we need to generate random numbers from different probability distributions; most notably from the uniform distribution. We will look at state-of-the-art pseudo-random number generators (PRNGs) and their implementations. Sometimes random number from different distributions are required, and we will look at several methods to generate non-uniform random numbers. While this can be relatively simple (e.g., for the exponential distribution), often complex procedures are involved (e.g., for the general gamma distribution).
- We also want to make sure that the PRNG we are using is of good quality; so we will discuss some of the empirical quality tests that a PRNG should be subjected to before its first use in “real” applications.
- In bioinformatics (but also in other fields), we often face the situation that we use an algorithm that solves a specific problem by computing an optimal or near-optimal solution, together with a “score” or quality value. We need to assess whether the output score value is significantly high to distinguish the result from “random noise”.

Often these significance estimations are quite complicated in their own right, and new algorithms have to be devised to make them feasible.

- ...

One goal of these lectures is to equip students with a solid foundation in algorithmic solutions to problems from probability theory and statistics, and provide them with a toolbox of simple but useful methods.

The first draft of these notes was prepared by the following students:

- Chapter 1 Jaroslav Latischev, Benjamin Rich
- Chapter 2 Benjamin Rich
- Chapter 3 Sven Mielordt
- Chapter 4 Wasinee Rungarityotin

# Chapter 1

## Probability Distributions and Random Variables

We start by summarizing some basic notions from probability theory which should be known by everyone interested in following these lectures. For a refresher, we recommend the textbooks by ? and ?; the latter is in German.

### 1.1 Probability Spaces

In the 1930s, Kolmogorov laid an axiomatic foundation for today's probability terminology; we are not going to be very rigorous about this and will just recall the most important points.

- All possible realizations of a random event form the *sample space* (Zustandsraum)  $\Omega$ . It can be
  - finite ( $\{1, \dots, n\}$ ; resulting number when throwing dice),
  - countable ( $\mathbb{N}, \mathbb{Z}$ ; waiting time for the first “6” when throwing dice),
  - uncountable ( $[a, b], \mathbb{R}$ ; “random number” in  $[0, 1]$ , weight of a newborn baby)

By definition, a set  $C$  is countable if there exists an injective map  $C \rightarrow \mathbb{N}$ . All of  $\mathbb{N}, \mathbb{Z}, \mathbb{Z}^n, \mathbb{Q}, \mathbb{Q}^n$  are countable.  $\mathbb{R}$  is uncountable.

- An *event* (Ereignis)  $A \subset \Omega$  is simply a subset of the sample space. Example: When throwing a 6-sided die, the event to roll an “odd number” is  $A = \{1, 3, 5\}$ .
- Especially in uncountable sample spaces not every event can be made “measurable” without the risk of paradoxa. On  $\mathbb{R}$ , we restrict ourselves to open intervals and the sets that can be derived from it using the operations allowed in a  $\sigma$ -algebra. This is a system of events  $\mathcal{A} \subset \{A | A \subset \Omega\}$  (also written as  $\mathcal{A} \subset 2^\Omega$ ; here  $2^\Omega$  denotes the power set of  $\Omega$ ) which satisfies
  1.  $\Omega \in \mathcal{A}$
  2.  $A \in \mathcal{A}$  implies  $A^c \in \mathcal{A}$  (the superscript  $c$  denotes complement)
  3. If  $A_1, A_2, \dots \in \mathcal{A}$ , then the countable union  $\bigcup_{i=1}^{\infty} A_i$  is also in  $\mathcal{A}$

$\mathcal{A} = \{\emptyset, \Omega\}$  is the trivial  $\sigma$ -algebra.  $\mathcal{A} = 2^\Omega$  is the  $\sigma$ -algebra of all possible events; as noted, it can be too big to define a meaningful measure on all events.

- A *probability measure* or *probability distribution* (Wahrscheinlichkeitsmaß oder -verteilung) on  $(\Omega, \mathcal{A})$  is a function  $\mathbb{P} : \mathcal{A} \rightarrow [0, 1]$  that satisfies

1.  $\mathbb{P}(\Omega) = 1$
2.  $\mathbb{P}(A) \in [0, 1]$  for all  $A \in \mathcal{A}$ .
3.  $\sigma$ -additivity (countable additivity):  $\mathbb{P}(\bigcup_{i=1}^{\infty} A_i) = \sum_{i=1}^{\infty} \mathbb{P}(A_i)$  if the  $A_i$  are pairwise disjoint ( $A_i \cap A_j = \emptyset$  for  $i \neq j$ ). Note that this implies finite additivity; we can always take  $A_i = \emptyset$  for infinitely many  $A_i$ .

$\Omega$  is the *certain event*,  $\emptyset$  the *impossible event*. An event  $A$  with  $\mathbb{P}(A) = 1$  happens *almost surely* (a.s.; fast sicher, f.s.).

- The triple  $(\Omega, \mathcal{A}, \mathbb{P})$  is called a *probability space*.

Two typical examples:

1. We take  $\Omega = \{1, 2, 3, 4, 5, 6\}$ ,  $\mathcal{A} = 2^\Omega$ , and define  $\mathbb{P}$  by  $\mathbb{P}(A) = |A|/6$ , where  $|A|$  denotes the cardinality of  $A$ , i.e., the number of elements in  $A$ . It is easy to verify that  $\mathbb{P}$  is  $\sigma$ -additive in this case.
2. We take  $\Omega = [0, 1]$ , and we would like to define a “uniform” probability measure on  $\Omega$ , i.e.,  $\mathbb{P}([a, b]) = b - a$  for  $0 \leq a < b \leq 1$ . So we take  $\mathcal{A}$  as the smallest  $\sigma$ -algebra generated by these intervals (this is the *Borel*  $\sigma$ -algebra on  $[0, 1]$ ; it is equal to the  $\sigma$ -algebra generated by all open sets and also contains all countable unions of intervals).

The second example is interesting because of the following “computational paradoxon”, which illustrates well an important point of these lectures: Let  $Q$  denote the rational numbers in  $[0, 1]$ . There are only countably many of them (exercise). Let  $q_1, q_2, \dots$  be an enumeration of them. Thus  $\mathbb{P}(Q) = \sum_i \mathbb{P}(\{q_i\}) = 0$ , because  $\mathbb{P}(\{x\}) = 0$  for every point  $x$  and by  $\sigma$ -additivity. Therefore the event  $[0, 1] \setminus Q$  occurs a.s.; in other words, when we draw a random number, it is a.s. not rational. However, if we draw a random number with a computer, it is always rational due to the finite precision of floating point numbers. This should serve as a warning that there is often a large gap between a theory and its computer implementation!

## 1.2 Conditional Probabilities

For an event  $B$  with  $\mathbb{P}(B) > 0$  we define the *conditional probability* of  $A$ , given that  $B$  occurs as

$$\mathbb{P}(A|B) := \frac{\mathbb{P}(A \cap B)}{\mathbb{P}(B)} = \frac{\mathbb{P}(A, B)}{\mathbb{P}(B)}.$$

In the context of probabilities of events, the comma notation is simply a shorthand for set intersection.

Let  $B \cup B^c$  be a partition of  $\Omega$ , such that both  $\mathbb{P}(B) > 0$  and  $\mathbb{P}(B^c) > 0$ . Then we have

$$\begin{aligned}\mathbb{P}(A) &= \mathbb{P}(A, B) + \mathbb{P}(A, B^c) \\ &= \mathbb{P}(A|B) \cdot \mathbb{P}(B) + \mathbb{P}(A|B^c) \cdot (1 - \mathbb{P}(B)),\end{aligned}$$

because  $\mathbb{P}(B^c) = 1 - \mathbb{P}(B)$ .

If  $\mathbb{P}(A) > 0$  and  $\mathbb{P}(B) > 0$ , we have  $\mathbb{P}(A|B) \cdot \mathbb{P}(B) = \mathbb{P}(A, B) = \mathbb{P}(B, A) = \mathbb{P}(B|A) \cdot \mathbb{P}(A)$ . From this we derive *Bayes' Rule*

$$\begin{aligned}\mathbb{P}(B|A) &= \mathbb{P}(A|B) \cdot \mathbb{P}(B) / \mathbb{P}(A) \\ &= \frac{\mathbb{P}(A|B) \cdot \mathbb{P}(B)}{\mathbb{P}(A|B) \cdot \mathbb{P}(B) + \mathbb{P}(A|B^c) \cdot (1 - \mathbb{P}(B))}.\end{aligned}$$

Two events  $A$  and  $B$  are *independent* if and only if  $\mathbb{P}(A|B) = \mathbb{P}(A)$  and  $\mathbb{P}(B|A) = \mathbb{P}(B)$ , i.e., if conditioning on one event does not change the probability of the other event. Both conditions can be summarized as  $\mathbb{P}(A, B) = \mathbb{P}(A) \cdot \mathbb{P}(B)$ .

In general,  $n$  events  $A_1, \dots, A_n$  are independent if  $\mathbb{P}(A_{i_1}, \dots, A_{i_m}) = \prod_{k=1}^m \mathbb{P}(A_{i_k})$  for all subsets  $\{i_1, \dots, i_m\} \subset \{1, \dots, n\}$ .

## 1.3 Random Variables

**Definition 1.1 (Random variable).** Let  $(\Omega, \mathcal{A}, \mathbb{P})$  be a probability space. A *random variable* (r.v.) is a function  $X : \Omega \rightarrow \mathbb{R}^d$ .

We will often assume that the dimension  $d = 1$ . Statements about  $X$  correspond to events in  $\Omega$ .

**Example 1.1 (Sum of 2 dice).** The sample space  $\Omega$  is given by  $\Omega = \{(a, b) | 1 \leq a, b \leq 6\}$ . Consider the r.v.  $X$  defined as  $X : \Omega \rightarrow \{1, 2, \dots, 12\}$ ,  $(a, b) \mapsto a + b$ . The statement “ $X = 4$ ” is really the event  $A := \{(a, b) \in \Omega | a + b = 4\} = \{(1, 3), (2, 2), (3, 1)\}$ .

A random variable defines a new probability measure on its range in terms of the measure in its domain: The distribution of  $X$  is defined in terms  $\mathbb{P}$  in  $(\Omega, \mathcal{A}, \mathbb{P})$ . In general,

$$\mathbb{P}(X \in B) := \mathbb{P}(X^{-1}(B)) = \mathbb{P} \circ X^{-1}(B).$$

If  $B$  consists of a single element  $b$ , we omit the set brackets, i.e., we write  $\mathbb{P}(X = b)$  instead of  $\mathbb{P}(X = \{b\})$ . Note that  $X$  must be *measurable*, i.e., for every  $B$  of interest,  $X^{-1}(B)$  must be in the original  $\sigma$ -algebra  $\mathcal{A}$ .

Notational remark: Frequently after a r.v. has been defined, we do no longer care about the underlying  $\Omega$ ; our interest lies only in  $X$  and its distribution or *law*  $\mathcal{L}(X)$ . To express that  $X$  has a standard Normal distribution, we either write  $\mathcal{L}(X) = \mathcal{N}(0, 1)$  or  $X \sim \mathcal{N}(0, 1)$ .

Two r.v.s  $X$  and  $Y$  defined on the same probability space are *independent* if the events  $X \in B$ ,  $Y \in C$  are independent for all choices of  $B$  and  $C$ . A family  $(X_i)$  of r.v.s is independent if for every finite subset  $(X_{i_1}, \dots, X_{i_m})$  the  $m$  events  $X_{i_1} \in B_1, \dots, X_{i_m} \in B_m$  are independent for every choice of  $B_1, \dots, B_m$ .

## 1.4 Moments

To be written.

Mean

Variance

Skewness

Kurtosis

## Chapter 2

# Computer Representation of Probabilities and Distributions

Our main interest lies in the representation of probability distributions in the computer. We will examine various ways to represent distributions, and we will also take a close look at the IEEE floating point standard, which defines the (finite) precision which is all we have to store “real” numbers in the computer.

We defined a probability measure as a function  $\mathbb{P} : \mathcal{A} \rightarrow [0, 1]$ , that means, for every event  $A \in \mathcal{A}$ , we must be able to compute  $\mathbb{P}(A)$ . Because of  $\sigma$ -additivity, it would be wasteful to store the probability of all events. For finite distributions, it would suffice to store  $\mathbb{P}(\{a\})$  for singletons  $\{a\}$  and then add singleton probabilities to obtain probabilities for composite events  $A = \{a_1, a_2, \dots, a_k\}$ .

Let us approach the question of how a distribution can be represented systematically. For now, we will restrict ourselves mainly to univariate (one-dimensional) distributions. The following functions are associated with a probability measure  $\mathbb{P}$ , and it is generally useful to have them available “at your fingertips”.

### Definition 2.1 (pmf/ff, cdf, pdf, cf).

- The *probability mass function* (pmf), sometimes *frequency function* (ff) of a distribution  $\mathbb{P}$  is defined as  $p : \Omega \rightarrow [0, 1]$ ,  $p(x) := \mathbb{P}(\{x\})$ . It is mainly of interest for discrete (finite or countable distributions).
- The *cumulative distribution function* (cdf) is defined for totally ordered sample spaces  $\Omega$  (especially for  $\Omega \subset \mathbb{R}$ ) as  $F : \Omega \rightarrow [0, 1]$ ,  $F(x) := \mathbb{P}(B_x)$ , where  $B_x := \{y \mid y \leq x\}$  denotes the set of elements “below”  $x$ . This function is equally useful for discrete and for continuous distributions. Assuming that  $\Omega \subset \mathbb{R}$ , we can define  $F$  on the whole real line, and it follows that  $\lim_{x \rightarrow -\infty} F(x) = 0$  and  $\lim_{x \rightarrow \infty} F(x) = 1$ , and  $F$  is monotone increasing.
- The *probability density function* (pdf) is defined as the rate of change of the cdf, i.e.,  $f : \Omega \rightarrow \mathbb{R}$ ,  $f(x) := \frac{d}{dx} F(y)|_{y=x}$ . We have  $f(x) \geq 0$  and  $\int_{\mathbb{R}} f(x) dx = 1$ . The pdf can only be defined where the cdf is differentiable and hence is useful only for continuous distributions. Note that

$$\mathbb{P}(]a, b]) = \int_a^b f(x) dx = F(b) - F(a).$$

- The *quantile function* (qf) is defined as  $q : [0, 1] \rightarrow \Omega \subset \mathbb{R}$ ,  $q(c) := \inf\{x \in \mathbb{R} : F(x) \geq c\}$ . It is sometimes also called the generalized inverse of the cdf. To each cumulative probability  $c$ , it associates the point  $x \in \Omega$  where  $F(x) = c$  if that is possible; otherwise, the smallest  $x$  where  $F(x) \geq c$ . This point is called the  $c$ -quantile of the distribution.

## 2.1 Parametric distributions

Many distributions that we will frequently encounter are not “arbitrary”, but in most cases the pmf or the pdf follows a simple functional form that depends on only a few (say, 1 to 3) parameters.

Sometimes, the cdf and qf are equally simple, but in other cases, they are not available in *closed form*. A closed form is an expression that can be written using elementary function (like sin, cos, log, ...). But what exactly is an elementary function? For example, if we *define* a certain cdf to be an elementary function, and then we trivially have a closed-form representation for it. Perhaps one of the defining criteria for elementary functions is that there is an efficient way to compute them numerically. Note that how best to compute a function numerically isn't at all obvious. If we were to look at the source code that is used to compute elementary functions in numerical libraries (e.g. the GNU Scientific Library), we likely wouldn't be able to recognize which function was being computed or how.

Optimally, we would have a direct implementation of each function; if this is not possible, we have to use numerical methods. A parametric distribution is represented by providing implementations of the above functions, assigning them a *type*, and then storing the type number and the values of the parameters.

As an example, consider the well-known Normal or Gaussian distribution, which is often used as a model for measurement error. Its parameters are the mean  $\mu$  and the variance  $\sigma^2$ . It has no meaningful pmf, because the probability of each singleton is zero. Its pdf is usually written as  $\varphi_{\mu, \sigma^2}$ :

$$\varphi_{\mu, \sigma^2}(x) = \frac{1}{\sqrt{2\pi} \cdot \sigma} \cdot \exp\left(\frac{-(x - \mu)^2}{2\sigma^2}\right) \quad (-\infty < x < \infty).$$

The corresponding cdf is called  $\Phi_{\mu, \sigma^2}$ , but there is no “simple closed form” for representing

$$\Phi_{\mu, \sigma^2}(x) = \int_{-\infty}^x \varphi_{\mu, \sigma^2}(x) dx.$$

Its values might be obtained by numerical integration of  $\varphi$ , or by different numerical methods. In practice one might use a combination of tabulated values and linear interpolation. The same could be done to implement the quantile function  $\Phi_{\mu, \sigma^2}^{-1}$ .

## 2.2 Non-parametric distributions

Observed data usually has its own distribution that doesn't exactly fit any of the usual parametric distributions. Typically, we either have a sequence of observations  $(X_i)_{i=1,\dots,n}$  (finite) or  $(X_i)_{i \geq 1}$  (countably infinite), where the same value may occur repeatedly. Or we may already have the observations in "histogram" form as pairs  $(X_j, C_j)$ , where  $X_j < X_{j+1}$  and  $C_j$  is the number of occurrences of  $X_j$ .

Example: The two sides of a coin are labelled 0 and 1, and we want to describe the observation  $(0, 1, 0, 0, 0, 1, 1)$  with a probability distribution. Sorting and counting the observations, this is  $\{(0, 4), (1, 3)\}$ .

Let us introduce a useful notation.

**Definition 2.2 ( $\{l:\varepsilon:u\}$  and  $\{a..b\}$ ).** Let  $\varepsilon > 0$  and let  $l < u$  be two real numbers such that  $(u-l) \in \varepsilon\mathbb{Z}$ , i.e.,  $(u-l)$  is an integer multiple of  $\varepsilon$  (frequently it will be the case that both  $l$  and  $u$  are integer multiples of  $\varepsilon$ ). Then  $\{l:\varepsilon:u\}$  denotes the set  $\{l, l+\varepsilon, l+2\varepsilon, \dots, u-\varepsilon, u\} = \{l + j\varepsilon \mid j = 0, \dots, (u-l)/\varepsilon\}$ ; it consists of  $1 + (u-l)/\varepsilon$  elements.

We also define  $\{a..b\} := \{a:1:b\} = \{a, a+1, \dots, b\}$ , assuming that  $b-a$  is an integer (often, both  $a$  and  $b$  will be integers).

Depending on the size of the sample space, different ways of representation suggest themselves.

- The number of different samples is finite and small: If the observations come from a set  $\{l:\varepsilon:u\}$ , we only need to store  $l, \varepsilon, u$ , the total number  $n$  of observations, and the corresponding *array* or *list* of counts  $(C_j)_{j=0,\dots,(u-l)/\varepsilon}$ , where  $C_j$  counts the number of observations equal to  $l + j\varepsilon$ .

If the observations have an irregular structure, it is better to use a *hash* or *associative array* to store pairs  $(X_j, C_j)$ , where  $X_j$  is the *key* and  $C_j$  is the *value*.

The pmf is now readily obtained. The cdf can be pre-computed by additionally storing cumulative sums in an additional vector or hash; the qf can then be computed by a binary search over the cdf.

- The number of different samples is finite but large, or potentially countably infinite: The best solution is to represent the central (most important) part of the distribution as a table or hash and use a simple parametric form for the tails. This may not be exact, but is a good trade-off between storage space and accuracy. See below for mixture distributions.
- We are trying to represent a continuous non-parametric distribution. In order to store an infinite amount of information in a finite amount of memory, we have to *discretize* the distribution.

We shall work with the following way to discretize a continuous distribution  $\mathbb{P}$  on  $\mathbb{R}$ .

**Definition 2.3 (( $l, \varepsilon, u$ )-discretization).** Let  $\Omega' := \{l : \varepsilon : u\} \cup \{<, >\}$ . The ( $l, \varepsilon, u$ )-discretization of  $\mathbb{P}$  with cdf  $F$  is a distribution  $\mathbb{P}'$  on  $\Omega'$  defined as follows (see also Figure 2.1).

$$\begin{aligned}\mathbb{P}'(\{<\}) &:= \mathbb{P}(-\infty, l - \varepsilon/2] = F(l - \varepsilon/2), \\ \mathbb{P}'(\{x\}) &:= \mathbb{P}(]x - \varepsilon/2, x + \varepsilon/2]) = F(x + \varepsilon/2) - F(x - \varepsilon/2) \quad (x \in \{l : \varepsilon : u\}), \\ \mathbb{P}'(\{>\}) &:= \mathbb{P}(]u + \varepsilon/2, +\infty[) = 1 - F(u + \varepsilon/2).\end{aligned}$$

To be done.

**Figure 2.1:** The ( $l, \varepsilon, u$ )-discretization of  $\mathbb{P}$

## 2.3 Mixture distributions

Often distributions which do not seem to have a simple parametric form can be represented as a *mixture* of such distributions or as a mixture of a short non-parametric distribution and a parametric approximation in the tails.

To be rewritten:

Def. Mixture

Let  $W$  be a prob. dist on  $N$ , and  $P_i$  ( $i$  in  $N$ ) be prob. dist. on  $\Omega$ . Then  $P$  defined by  $P(E) := \sum W(i) * P_i(E)$  is also a pd on  $\Omega$ , the mixture of  $P_i$  with weights  $W_i$ .

Let  $W$  be a prob. dist with density  $w$  on  $R \geq 0$

Let and  $P_x$  ( $x \geq 0$ ) be prob dist. on  $\Omega$ .

The  $P$  defined by  $P(E) := \int_{R \geq 0} w(x) * P_x(E) dx$  is a pd on  $\Omega$ , the mixture of  $(P_x)$  with weight density  $w$ .

**Example 2.1 (Intron lengths in *Drosophila Melanogaster*).** Consider the distribution representing intron length in *D. Melanogaster*. If we model gene finding with HMMs and we model “Intron” as a single state, this results in the intron length having a geometric distribution, which may not be realistic. Maybe we can represent the distribution not exactly, but accurately as a mixture of simple distribution. This would give a more efficient representation than a big table (see exercises).

## 2.4 The IEEE Floating Point Standard

[To be slightly rewritten for typesetting and copyright reasons.]

The following way of representing floating point numbers follows the ANSI/IEEE Standard 754-1985.

### 2.4.1 Single Precision

The IEEE single precision floating point standard representation requires a 32 bit word, which may be represented as numbered from 0 to 31, right to left. The leftmost bit is the sign bit, S, the next eight bits are the exponent bits, 'E', and the final 23 bits are the fraction 'F':

```
S EEEEEEEE FFFFFFFFFFFFFFFFFFFFFFFF
31 30      23 22                        0   bit index
```

The value V represented by the word may be determined as follows:

- If E=255 and F is nonzero, then V=NaN ("Not a number")
- If E=255 and F is zero and S is 1, then V=-Infinity
- If E=255 and F is zero and S is 0, then V=Infinity
- If  $0 < E < 255$  then  $V = (-1)^S * 2^{E-127} * (1.F)$  where "1.F" is intended to represent the binary number created by prefixing F with an implicit leading 1 and a binary point.
- If E=0 and F is nonzero, then  $V = (-1)^S * 2^{-126} * (0.F)$  These are "unnormalized" or "denormal" values. They have less precision than normalized numbers (loss of significance in F)
- If E=0 and F is zero and S is 1, then V=-0.
- If E=0 and F is zero and S is 0, then V=0.

In particular,

```
0 00000000 000000000000000000000000 = 0
1 00000000 000000000000000000000000 = -0

0 11111111 000000000000000000000000 = Infinity
1 11111111 000000000000000000000000 = -Infinity

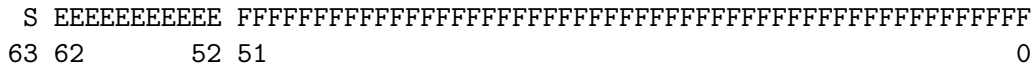
0 11111111 000001000000000000000000 = NaN
1 11111111 001000100010010101010101 = NaN

0 10000000 000000000000000000000000 = +1 * 2**(128-127) * 1.0 = 2
0 10000001 101000000000000000000000 = +1 * 2**(129-127) * 1.101 = 6.5
1 10000001 101000000000000000000000 = -1 * 2**(129-127) * 1.101 = -6.5

0 00000001 000000000000000000000000 = +1 * 2**(1-127) * 1.0 = 2**(-126)
0 00000000 100000000000000000000000 = +1 * 2**(-126) * 0.1 = 2**(-127)
0 00000000 000000000000000000000001 = +1 * 2**(-126) *
                                          0.000000000000000000000001 =
                                          2**(-149) (Smallest positive value)
```

### 2.4.2 Double Precision

The IEEE double precision floating point standard representation requires a 64 bit word



The value V represented by the word may be determined as follows:

- If E=2047 and F is nonzero, then V=NaN ("Not a number")
- If E=2047 and F is zero and S is 1, then V=-Infinity
- If E=2047 and F is zero and S is 0, then V=Infinity
- If  $0 < E < 2047$  then  $V = (-1)^S * 2^{(E-1023)} * (1.F)$  where "1.F" is intended to represent the binary number created by prefixing F with an implicit leading 1 and a binary point.
- If E=0 and F is nonzero, then  $V = (-1)^S * 2^{(-1022)} * (0.F)$  These are "unnormalized" values.
- If E=0 and F is zero and S is 1, then V=-0.
- If E=0 and F is zero and S is 0, then V=0.

### 2.4.3 Notes

From this representation, we can determine

- maximal single/double
- minimal single/double
- Epsilon

Addition, Subtraction: Re-normalization, Possible loss of accuracy.

A floating point number represents in fact a whole range of numbers.

Interval arithmetic.

# Chapter 3

## Important Parametric Distributions

We introduce some frequently occurring parametric distributions and present typical situations where they occur.

### 3.1 Discrete Distributions

Recall that a discrete distribution (on a totally ordered set) is defined by its probability mass function (pmf), also called frequency function (ff), denoted by  $p$ ,  $p(x) = \mathbb{P}(X = x)$ , or its cumulative distribution function (cdf), denoted by  $F$ ,  $F(x) = \mathbb{P}(X \leq x)$ .

#### 3.1.1 Uniform Distribution

We have a finite subset  $S \subset \mathbb{R}$ . We have

$$p(x) = \frac{1}{|S|}, \forall x \in S$$

#### 3.1.2 Binomial

Suppose we do  $n$  independent experiments, each with a probability of success  $p$ . Let  $X$  be the number of successes. Then

$$\mathbb{P}(X = k) = \binom{n}{k} p^k (1-p)^{n-k},$$

where  $\binom{n}{k} = \frac{n!}{k!(n-k)!} = \frac{n(n-1)\dots(n-k+1)}{1 \cdot 2 \dots k}$ .

Special case: If  $n = 1$ , then  $X$  is called a Bernoulli r.v. These arise frequently as *indicator variables*:

$$I_A : \Omega \rightarrow 0, 1$$
$$I_A(\omega) = \begin{cases} 1 & \text{if } \omega \in A \\ 0 & \text{if } \omega \notin A \end{cases}$$

**Remark.** The sum of  $n$  Bernoulli random variables has a binomial distribution. If you define  $X = X_1 + \dots + X_n$ , where each  $X_i$  is Bernoulli( $p$ ), then  $X$  is Binomial( $n, p$ ).

**Definition 3.1.** A random variable  $X$  with

$$\mathbb{P}(X = k) = \binom{n}{k} p^k (1-p)^{n-k}$$

is called *binomial* with parameters  $n, p$ .

Short notation: Binomial( $n, p$ ), or  $\mathcal{B}(n, p)$ , or Bin( $n, p$ ).

### Geometric

Again, we have independent experiments with probability of success  $p$ . We wait for the first success. Let  $X$  be the number of the first successful experiment. Note that  $X \in 1..∞$ . Then

$$\mathbb{P}(X = k) = p(1-p)^{k-1} = p \cdot q^{k-1}$$

where  $q = 1 - p$ . We can also compute the cdf

$$F(k) = \sum_{j=1}^k \mathbb{P}(X = j) = 1 - q^k$$

since we have a geometric sum. An interesting property of the geometric distribution is that it is *memoryless*, in the sense that

$$\mathbb{P}(X > t + d | X > t) = \mathbb{P}(X > d)$$

**Proof.**

$$\mathbb{P}(X > t + d | X > t) = \frac{q^{t+d}}{q^t} = q^d = 1 - F(d) = \mathbb{P}(X > d)$$

□

You can generalize the geometric distribution by waiting for the  $r$ -th success. Then

$$\mathbb{P}(X = k) = \binom{k-1}{r-1} p^r \cdot q^{k-r}$$

We will call this the *Discrete Gamma*( $r, p$ )

A slight variation of the geometric distribution is to let  $X$  be the number of failures until the first success ( $X \in 0..∞$ ). Then  $\mathbb{P}(X = k) = p \cdot q^k$ . We call this the *Geometric*( $p$ ).

This can be generalized similarly by letting  $X$  be the total number of failures before the  $r$ -th success (*without* the  $r-1$  successes) ( $X \in 0..∞$ ). We then say the  $X$  has a *Negative Binomial* or *Pascal* or *Polya* distribution with parameters  $r, p$ . In this case

$$\mathbb{P}(X = k) = \binom{k+r-1}{r-1} p^r \cdot q^k = \binom{-r}{k} (-1)^k p^r \cdot q^k$$

### Hypergeometric distribution

You have an urn containing  $n$  balls,  $r$  of which are black. You draw  $m$  balls without replacement. Let  $X$  be the number of black ball drawn. Then

$$\mathbb{P}(X = k) = \frac{\binom{m}{k} \binom{n-m}{r-k}}{\binom{n}{m}}$$



# Chapter 4

## Generation of Random Numbers

### 4.1 Pseudo-random numbers

The question of what a random number is “philosophical” in nature. The term *random number* implies a sequence of numbers altogether generated by random process and not a single number alone as intuition may lead. Often in scientific computation, we would like to have the randomness that is reproducible. That is we want to be able to repeat the same random experiment. Thus, most of the time and through out this class, we will use the term *random* to mean *pseudo-random* which readily implies reproducible randomness. In this lecture, we will first examine the definition of pseudo-randomness and look at the typical transformation.

**Definition 4.1 (Pseudo-random number generator).** A *Pseudo-random number generator* consists of:

1. A set of states  $\mathbb{S}$ :  $\mathbb{S} = \{0, 1, \dots, M\}^m$ , where  $M$  is the largest native integer and  $m$  is a small integer. On a 32-bit machine,  $M = 2^{32} - 1$  and a 64-bit machine,  $M = 2^{64} - 1$ .
2. State transformation  $D$  which is a function that maps one state to the next.  $D$  is defined as  $D : \mathbb{S} \rightarrow \mathbb{S}$ .
3. Output transformation  $U : \mathbb{S} \rightarrow [0, 1[$ .
4. A random seed indicating a starting state.

**Definition 4.2 (Period of the random number generator).** Definition based on the output space  $U$ . A period  $T$  is the smallest integer  $T > 0$  such that  $U_k = U_{k+T}$  for all sufficiently large  $k$ . We can also define a period based on the state space as well. With this notion, we can say that A period is the smallest integer  $T > 0$  such that  $D^T = I$ .

**Proof.** Left as the exercise. The period cannot be larger than  $M + 1$ . □

### 4.2 Uniform Random Number

We will consider three principles for designing the uniform random number generations: congruential RNG, multiply-with-carry RNG and shift-register RNG.

### 4.2.1 Congruential RNG

**Definition 4.3 (Congruential RNG).** Congruential RNG is a pseudo-random number generator defined over an integer state space. Let the state space be  $\mathbb{S} = \{0, 1, \dots, M\}$  defined over a set of integers only. Define the transformation  $D : \mathbb{S} \rightarrow \mathbb{S}$ ,

$$D(X) = (a \cdot X + b) \bmod (M + 1)$$

and the output map  $U(X) = \frac{X}{M+1}, U(X) \in [0, 1[$ .

**Example 4.1.**  $D(X) = (5X + 7) \pmod{10}$ .

$X_t$	$D(X)$	$X_{t+1}$
0	$5 \cdot 0 + 7$	7
5	$5 \cdot 5 + 7$	2
6	$5 \cdot 6 + 7$	7
7	$5 \cdot 7 + 7$	2
9	$5 \cdot 9 + 7$	2

This is obviously not a great generator because it cycles between 7 and 2.

#### Remarks on the implementation of Congruential-RNG.

- Because  $D(X)$  only contains integer arithmetics, the *mod* operation is implicit on a modern machine. That is you do not need to perform mod operation in the implementation.
- What are the good choices of  $a, b$ ?: Some heuristics say that both (especially  $a$ ) should be large prime-numbers. They should not be even numbers, because the range of  $D(X)$  will not cover the whole state space.

### 4.2.2 Multiply-with-carry RNG

**Definition 4.4 (Multiply-with-carry RNG.).** A state is a pair of machine words. The domain is  $\mathbb{S} = \{0, 1, \dots, M\}^2, M = 2^k - 1$ . Define a state as  $(Z, C)$ . The state transformation  $(Z_t, C_t)$  to  $(Z_{t+1}, C_{t+1})$  is

$$\begin{aligned} Z_{t+1} &= a \cdot Z_t + C_t \\ C_{t+1} &= \lfloor a \cdot Z_{t+1} \rfloor / (M + 1), \end{aligned}$$

where  $a$  is a constant and  $C_{t+1}$  is the overflow. Let  $K$  be the width of one machine word in bits, the output function can be written as  $U : (Z, C) \rightarrow Z / (M + 1)$ .

It is important that the constant  $a$  is large so that the carry term  $C$  also covers the domain of  $Z$ . The maximal period is  $a \cdot (M + 1) - 1$ .

### 4.2.3 Shift-Register RNG

**Definition 4.5 (Shift-Register RNG).** This RNG define a state  $X$  as  $k$ -bit words.

$$X = b_1b_2b_3\dots b_k$$

The domain of the state space is therefore  $\mathbb{S} = \{0, 1, \dots, 2^k - 1\}$ .  $D(X)$  is a series of bitwise-XOR operations on  $X$ . See example 4.2 for a design.

**How to evaluate the quality of RNG?** Look at the joint distribution of a few consecutive states. For example, we can examine a pair of consecutive states  $(X_t, X_{t+1})$  or long-range correlation by looking at longer time step such as  $(X_t, X_{t+5})$ . One heuristic is to make a correlation plot between a few consecutive states. An RNG is good enough, if one see very few structures in the plot of the joint distribution at all ranges of time steps. More formal testing can be done as well, see chapter 5.

**Example 4.2 (Uniform RNG: Keep It Simple and Stupid).** This idea was introduced by Marsaglia, 1993 and based on combining all of the above principles. State space consists of 4 machine words  $(X, Y, Z, C) \in \{0, 1, \dots, 2^k - 1\}^4$ . Initial seeds are

$$X_0 = 123456789, Y_0 = 362436, Z_0 = 521288629, C_0 = 7654321$$

Define the transformation for each of the word as follows:

$$X_{t+1} = a \cdot X_t + b$$

Suggest  $a = 69069, b = 12345$  or  $a \equiv (3 \pmod{8})$  or  $(5 \pmod{8})$  and  $b$  can be any large odd constant.

$$y^\wedge = (Y_t \ll 13);$$

$$y^\wedge = (y \gg 17);$$

$$Y_{t+1}^\wedge = (y \ll 5);$$

Note that 161 other triplets besides  $(13, 17, 5)$  also work well.  $(Z, C)$  is a pair of machine words in multiply-with-carry RNG with  $a = 69069$ .

The output of KISS-RNG is

$$U = (X + Y + Z) \pmod{2^k}.$$

This RNG works because the period is so long that sequences of random numbers are independent. For  $k = 32$ , the period length is larger than  $2^{124} \approx 10^{37}$ .

### 4.3 Non-Uniform Random Variates

We want to generate a set of samples that obeys a given probability distribution  $f(X)$  and having the CDF,  $F(X)$ . The idea is to use the inverse of the CDF,  $F^{-1}(U)$  to generate the set of samples  $X$ . This method is usually called the *inverse transformation* method.

Given a sequence  $U$  of uniform r.v. between  $[0, 1[$ , a sequence of samples  $X$  such that  $F(X) = U$  is the set of samples which obeys the cumulative distribution  $F(X)$ . Note that when one does not have a continuous  $F(X)$ , one defines the *generalized* inverse  $F^{-1}$  by

$$F^{-1}(X) = \min\{y | F(y) \geq X\}.$$

**Lemma 4.1.** *Assume that  $F$  is a monotonically increasing function and its range is uniformly distributed on  $[0, 1[$ . If  $X$  is an r.v. such that  $P(X \leq X^*) = U^* = F(X^*)$ , then the CDF of  $X$  is the function  $F$ .*

**Proof.** We must show that the CDF of the samples produced by this procedure is exactly  $F(X)$ .

$$\begin{aligned} P(X \leq X^*) &= P(F^{-1}(U^*) \leq X^*) \\ &= P(U^* \leq F(X^*)) \text{ , (1)} \\ &= F(X^*) \text{ , (2)} \end{aligned}$$

(1) uses the fact that  $F$  is monotonically increasing function. (2) uses the fact that  $0 \leq F(X) \leq 1$  and  $U^* \sim \text{Uniform}(0, 1)$ .

□

**Example 4.3.** Generate samples from the Exponential distribution.

The density  $f(x) = \lambda e^{-\lambda x}$ ,  $x \geq 0$  and the CDF  $F(x) = 1 - e^{-\lambda x}$ ,  $x \geq 0$ , obtained by direct integration of  $f(x)$ .  $F^{-1}$  is obtained by:

$$\begin{aligned} U &= 1 - e^{-\lambda x} \\ e^{-\lambda x} &= 1 - U = U^* \\ x &= -\ln(U^*)/\lambda, U^* \sim \text{Uniform}(0, 1) \end{aligned}$$

In summary,

1. Set  $U^* \sim \text{Uniform}(0, 1)$ .
2. Set  $x = -\ln(U^*)/\lambda$ .

# Chapter 5

## Principle of Accept-Reject Methods

### 5.1 Introduction

The principle of accept-reject methods is introduced in general and furthermore shown for a part of the normal distribution and the gamma distribution.

Primary goal of using accept-reject methods is to obtain a realization of a random variable  $X$  with pdf  $f$ , where  $f$  is known but difficult to sample.

#### Idea:

Find a pdf  $g$  with  $0 \leq f \leq Mg$  and sample  $y$  from  $g$ .

#### Algorithm 5.1.

1. Generate  $y$  from  $g$ .
2. Generate  $U$  uniform  $[0,1]$ .
3. if  $U \leq \frac{f(y)}{M \cdot g(y)}$  then  $X \leftarrow y$  (Accept...). Stop!  
else goto 1 (Reject...).

**Claim:**  $X$  has pdf  $f$

#### Proof.

In each iteration of step 1,  $Y = y$  is proposed with density  $g(y)$ .  
Conditional on that choice,  $Y$  is accepted with probability  $\frac{f(y)}{M \cdot g(y)} \leq 1$ .

Hence  $X = y$  is returned with probability  $g(y) \cdot \frac{f(y)}{M \cdot g(y)} = \frac{f(y)}{M}$

The overall probability of success is  $\int_y \frac{f(y)}{M} dy = \frac{1}{M}$ , with a number of trials until acceptance being geometric( $\frac{1}{M}$ ),  $\mathbb{E} = M$ .

Thus, the conditional probability to return  $X = y$  if we return is  $\frac{f(y)}{M} \div \frac{1}{M} = f(y)$ .

□

#### Refinements:

If the evaluation of  $f(x)$  is complicated, e.g.  $f(y) = \prod_{i=1}^N f_i(y)$  with  $N$  being large, the evaluation takes  $O(N)$  time.

→ Find a simple function  $l$  s.th.  $0 \leq l(y) \leq f(x)$ .

**Algorithm 5.2.**

1. Generate  $y$  from  $g$ .
2. Generate  $U$  uniform  $[0,1]$ .
3. if  $U \leq \frac{l(y)}{M \cdot g(y)}$  return  $X = y$   
 elsif  $U \leq \frac{f(y)}{M \cdot g(y)}$  then  $X \leftarrow y$   
 else goto 1

**5.2 Truncated normal distribution**

Imagine we are interested in sampling the right tail of the normal distribution, let's say sampling from  $f(x) \propto C \cdot e^{-\frac{x^2}{2}} \cdot \mathbb{I}\{x \geq t\}$  for some  $t \gg 0$ .

**Idea:**

- Use a translated exponential density with  $g_\lambda(x) = \lambda \cdot e^{-\lambda(x-t)} \cdot \mathbb{I}\{x \geq t\}$
- Optimize  $\lambda$  to get the best bound  $M_\lambda$  s.th.  $M_\lambda = \max_{x \geq t} \frac{f(x)}{g_\lambda(x)} = \max_{x \geq t} \frac{C \cdot e^{-\frac{x^2}{2}}}{\lambda \cdot e^{-\lambda(x-t)}}$
- Minimize  $M_\lambda$  in  $\lambda$  (first calculate  $M$  as a function of  $\lambda$ , then optimize it)

**5.3 Gamma r.v. with non-integer shape parameter**

We know how to sample from a  $\text{Gamma}(n, \lambda)$  with  $n \in \mathbb{N}$ .  
 Now we like to sample  $\text{Gamma}(a, 1)$  with  $a \in \mathbb{R}$ ,  $a > 1$ .

**Idea:** Use Accept-Reject with  $g: \text{Gamma}(n, 1)$  where  $n = \lfloor a \rfloor$ .

**But:** This will not work because the ratio  $\frac{f(x)}{g(x)}$  is unbounded for  $x \rightarrow \infty$ !

**Better:** Use  $\text{Gamma}(n, \lambda)$  with  $n = \lfloor a \rfloor$  and  $\lambda < 1$ . Then compute  $M_\lambda$  and minimize it.

The result is then  $\lambda = \frac{n}{a} = \frac{\lfloor a \rfloor}{a}$ .

**Finally:** Sample from  $(a, 1)$  with  $a < 1$

1. Generate  $Y \sim \text{Gamma}(a+1, 1)$
2. Generate  $U$  uniform  $[0,1]$ , and set  $Z \leftarrow U^{\frac{1}{a}} = \sqrt[a]{U}$ .
3. Set  $X = Y \cdot Z$

Then  $X \sim \text{Gamma}(a, 1)$ .

**Proof.**

Remember the density  $f(y)$  of  $Y: f(y) \propto y^a \cdot c^a$ , then  $Z = U^{\frac{1}{a}}$  has the density  $g(z) = a \cdot z^{a-1}$  for  $0 < z < 1$ .

When  $X = Y \cdot Z$ , then the cdf of  $X$  is  $\mathbb{P}(X \leq x) = \int_y f(y) \cdot G(\frac{x}{y}) dy$ , where  $G(\frac{x}{y})$  is the cdf of  $Z$  at  $\frac{x}{y}$ .

By derivation with respect to  $x$ , we get the pdf:

$$h(x) = \int_y f(y) \cdot g(\frac{x}{y}) \cdot \frac{1}{y} dy = c \cdot \int_{y=x}^{\infty} y^a \cdot e^{-y} \cdot (\frac{x}{y})^{a-1} \cdot \frac{1}{y} dy = c \cdot x^{a-1} \cdot \int_{y=x}^{\infty} e^{-y} dy = c \cdot x^{a-1} \cdot e^{-x}$$

being the pdf of a Gamma(a,1) r.v..

In general, scale parameter  $\lambda > 0$  and take  $X \propto \text{Gamma}(a,1)$ . Then  $X \leftarrow \frac{X}{\lambda}$ .

□

# Chapter 6

## Tests for Random Number Generation

This chapter introduces several tests for random number generators in order to test how well they work. The idea is to name a property of an "imitated" iid. r.v. and test it.

- Name a property:
  - Define "null hypothesis"  $H_0$ .
  - Define "alternative"  $K$  (the "rest").
- Test it:
  - Observe generated data.
  - Measure "difference" between observations and "expected data"  $H_0$ .
  - The difference measure is called the "test statistic".
  - If there's a large difference  $\rightarrow$  Reject  $H_0$ .

$\rightarrow$  How large must the difference be to reject  $H_0$ ?  $\rightarrow$  What are typical values (distribution) of the test statistic when  $H_0$  is true?

### 6.1 Chi-square test

Let's start with an example: Partition  $[0, 1]^3$  into  $10 \cdot 10 \cdot 10$  cells.

Then generate  $3 \cdot 100,000$  uniform random numbers  $\in [0,1]$  and consider non-overlapping triples  $(U_1, U_2, U_3), (U_4, U_5, U_6), \dots$  and count, how many triples fall into each cell.

$\rightarrow$  counts  $C_1, \dots, C_{1000}, \sum c_i = 100,000$

$H_0 : C \sim \text{Multinomial}(100,000, (\frac{1}{1000}, \frac{1}{1000}, \frac{1}{1000}, \dots))$

$K : C \sim \text{some other distribution}, C \sim \text{Multinomial}(100,000, p)$  with  $p \neq (\frac{1}{1000}, \frac{1}{1000}, \frac{1}{1000}, \dots)$

Thus, consider the likelihood ratio  $\Lambda = \frac{\mathbb{P}_{H_0}(Data)}{\mathbb{P}_K(Data)} = \frac{\max_{h \in H_0} \mathbb{P}_h(Data)}{\max_{k \in K} \mathbb{P}_k(Data)}$ , then  $-2 \log \Lambda$  is often used as the test statistic.

In our example above,  $\Lambda = \frac{\binom{100000}{c_1, c_2, \dots, c_{1000}} \cdot q_1^{c_1} \cdot q_2^{c_2} \cdots q_{1000}^{c_{1000}}}{\binom{100000}{c_1, c_2, \dots, c_{1000}} \cdot p_1^{c_1} \cdot p_2^{c_2} \cdots p_{1000}^{c_{1000}}}$  where  $p$  is the maximum likelihood estimate of the cell-parameters  $p_i = \frac{c_i}{100,000} = \frac{c_i}{N}$ .

$$\Rightarrow T = -2 \log \Lambda = 2 \cdot \sum_i c_i \cdot \log \frac{p_i}{q_i} = 2 \cdot \sum_i c_i \cdot \log \frac{c_i}{N \cdot q_i} = 2 \cdot N \cdot \sum_i p_i \log \frac{p_i}{q_i} \geq 0$$

(if  $f = 0 \Leftrightarrow p = q$ ).

**Remark.** For  $N \rightarrow \infty$ , the distribution  $\mathcal{L}_{H_0}(T) = \chi_{m-1}^2$  with  $m = 1000$  (number of classes).

# Chapter 7

## Understanding Alignment Statistics

### 7.1 Introduction

**Remark.** We will only consider gapless alignments in this lecture.

**Definition 7.1 (Optimal Local Alignment Score).**

Denote by  $M_n$  the optimal local alignment score of two sequences of length  $n$ .

Formally:

Sequence  $A : A_1, A_2, \dots, A_n$

Sequence  $B : B_1, B_2, \dots, B_n$

$\mathcal{A}$ : alphabet of  $A$  and  $B$

Scoring Matrix  $F : \mathcal{A} \times \mathcal{A} \mapsto \mathbb{R}$  (*symmetric, non-trivial*)

Score of local alignment starting at  $A_i$  and  $B_j$  with length  $\Delta$ :

$S_{i,j,\Delta} := \sum_{k=1}^{\Delta} F(A_{i+k}, B_{j+k})$  with  $\Delta \geq 0$  and  $0 \leq i, j \leq n - \Delta$

$S_{\Delta} := S_{0,0,\Delta}$

Then

$$M_n = \max_{\substack{\Delta \geq 0 \\ 0 \leq i, j \leq n - \Delta}} S_{i,j,\Delta}$$

### 7.2 Distribution of Optimal Local Alignment Scores

#### 7.2.1 I.I.D. Sequence Model

All letters are drawn independently from the same background distribution.

$A_1, A_2, \dots, A_n, B_1, B_2, \dots, B_n$  are i.i.d. with Distribution  $\mu$  on  $\mathcal{A}$ .

**Reasonable Requirements.** Let  $a, b$  be random letters and  $F$  be the scoring matrix, then

1.  $\mathbb{E}_{\mu \times \mu}[F(a, b)] < 0$
2.  $\Pr_{\mu \times \mu}(F(a, b) > 0) > 0$

Denote this set of requirements by (H).

## 7.2.2 Distribution of $M_n$

### Principal Result

**Theorem 7.1.** *Assume (H) holds, then:*

$$\lim_{n \rightarrow \infty} \Pr \left( M_n - \frac{2 \cdot \log(n)}{\Theta^*} \leq x \right) = e^{-K^* \cdot e^{-\Theta^* x}}$$

where  $\Theta^*$  is the unique solution of

$$\phi(\Theta) := \log \left( \mathbb{E}_{\mu \times \mu} \left[ e^{\Theta \cdot F(a,b)} \right] \right) = 0$$

$\phi(\Theta)$  is also known as the “log moment generating function” of  $F(a, b)$ .

$K^*$  depends only on  $F$ , but is difficult to calculate. Since it is less important than  $\Theta^*$ , we will treat  $K^*$  as a constant.

### Understanding Theorem 7.1: Step 1

**Remark.** For large  $x$  and  $n$ , usually p-values for alignment statistics yield

$$\Pr(M_n > x) \approx K^* \cdot n^2 \cdot e^{-\Theta^* x}$$

Here

$$\begin{aligned} \Pr(M_n > x) &= 1 - \Pr(M_n \leq x) \\ &= 1 - \Pr \left( M_n - \frac{2 \cdot \log(n)}{\Theta^n} \leq x \frac{2 \cdot \log(n)}{\Theta^n} \right) \\ &\approx 1 - e^{-K^* \cdot e^{-\Theta^* \cdot (x - \frac{2 \cdot \log(n)}{\Theta^n})}} \\ &= 1 - e^{-K^* \cdot n^2 \cdot e^{-\Theta^* x}} \\ &\approx K^* \cdot n^2 \cdot e^{-\Theta^* x} \end{aligned}$$

So theorem 7.1 is in line with common alignment statistics.

Let random variables  $X_1, X_2, \dots$  be i.i.d.  $\text{Exp}(\lambda)$  with  $\Pr(X_i > x) = e^{-\lambda x}$ .

Define  $U_n := \max_{1 \leq i \leq n} X_i$ , then

$$\begin{aligned} \Pr(U_n < x) &= \Pr(\{X_1 < x\} \cap \{X_2 < x\} \cap \dots \cap \{X_n < x\}) \\ &= (1 - e^{-\lambda x}) \cdot (1 - e^{-\lambda x}) \cdot \dots \cdot (1 - e^{-\lambda x}) \\ &= (1 - e^{-\lambda x})^n \\ &= \left(1 - \frac{n \cdot e^{-\lambda x}}{n}\right)^n \end{aligned}$$

**Remark.** From Analysis we know:

$$\lim_{n \rightarrow \infty} \left(1 - \frac{x}{n}\right)^n = e^{-x}$$

Therefore

$$\Pr\left(U_n - \frac{\log(n)}{\lambda} \leq y\right) = \Pr\left(U_n \leq y + \frac{\log(n)}{\lambda}\right) = \left(1 - \frac{e^{-\lambda y}}{n}\right)^n \rightarrow e^{-e^{-\lambda y}}$$

with  $y = x - \frac{\log(n)}{\lambda}$ .

The function  $e^{-e^{-\lambda y}}$  is an extreme-value distribution of type II (a.k.a. Gumbel distribution).

**Conclusion.** Heuristically  $M_n$  can be interpreted as the maximum of  $n^2$  i.i.d. random variables, where each r.v. has a tail that decays like  $K^* \cdot e^{-\Theta^* x}$ .

### Understanding Theorem 7.1: Step 2

What are the  $n^2$  i.i.d. random variables?

Recall

$$S_{i,j,\Delta} = \sum_{k=1}^{\Delta} F(A_{i+k}, B_{j+k})$$

**Definition 7.2.**

$$T_{i,j} := \max_{\Delta \geq 0} S_{i,j,\Delta}$$

$$T := \max_{\Delta \geq 0} S_{0,0,\Delta}$$

$T_{i,j}$  is the maximal score of all local alignments starting at  $A_{i+1}$  and  $B_{j+1}$ .

Then (neglecting edge effects)

$$M_n \approx \max_{0 \leq i,j \leq n} T_{i,j}$$

Now we have to show that

$$\Pr(T > x) \approx C \cdot e^{-\Theta^* x} \quad \text{for } x \rightarrow \infty$$

Basic idea:

$$\Pr(T > x) = \Pr\left(\max_{\Delta \geq 0} S_{\Delta} > x\right) \approx \max_{\Delta \geq 0} \Pr(S_{\Delta} > x) \approx C \cdot e^{-\Theta^* x}$$

# Chapter 8

## Understanding Alignment Statistics [2]

*Could be merged with part one in the final script.*

### 8.1 Distribution of the Optimal Alignment Score

Recap from previous section:

In order to validate the approximation for the distribution of  $M_n$ , the optimal alignment score of two sequences  $A$  and  $B$ , which is given by the maximum over all alignments starting at position  $A_i$  and  $B_j$ ,  $T_{i,j}$ :

$$M_n \approx \max_{0 \leq i, j \leq n} T_{i,j}$$

we have to show that the tail of the distribution of each  $T_{i,j}$  decays with

$$\Pr(T > x) \approx C \cdot e^{-\theta^* x}.$$

where  $\theta^*$  is the unique positive zero of the log moment generating function  $\phi$ .

In order to do so we first rewrite the distribution in question using asymptotical similarity:

$$\Pr(T > x) = \Pr\left(\max_{\Delta \geq 0} S_{\Delta} > x\right) \approx \max_{\Delta \geq 0} \Pr(S_{\Delta} > x)$$

The " $\approx$ " denotes logarithmic equivalence as defined in problem 37, exercise 11.

Now define  $\mathbb{E}[X, A]$  for some r.v.  $X$  and some condition  $A$  over values of  $X$  as the expectation of all  $x$  which fulfill  $A$ ,

$$\mathbb{E}[X, A] = \sum_{x \in X} x I_A \Pr(X = x) = \sum_{x \in A} x \Pr(X = x)$$

where  $I_A$  is the indicator function of condition  $A$ .

We can now write

$$\Pr(S_{\Delta} > x) = \mathbb{E}[1, S_A > x]$$

which can be written as

$$\mathbb{E}\left[\frac{e^{\theta S_{\Delta} - \Delta\phi(\theta)}}{e^{\theta S_{\Delta} - \Delta\phi(\theta)}}, S_{\Delta} > x\right]$$

Before we can exploit this last transformation we need to learn more about the distribution of the  $S_{\Delta}$ . In order to do so we introduce the concept of *exponential tilting*:

**Definition 8.1.** (Exponential tilting)

Given a probability measure  $Pr$  we define  $Pr_{\theta}$  as

$$Pr_{\theta}(X = x) = e^{\theta X - \phi(\theta)} Pr(X = x)$$

We call  $Pr_{\theta}$  exponentially tilted with parameter  $\theta \in \mathbb{R}$ .

Note that obviously  $Pr_0 = Pr$

Exponential tilting enables us to shift the expectation value of  $\mathbb{E}[X]$  of a r.v  $X$  to some  $x$ , depending on the choice for  $\theta$ . We can use this to construct a  $Pr_{\theta}(S_{\Delta} > x)$  with  $\mathbb{E}_{\theta}[S_{\Delta}] = x$ . The parameter  $\theta_{\Delta, x}$  which delivers this result then provides information about the unknown distribution of  $Pr(S_{\Delta} > x)$ .

Recall that  $S_{\Delta}$  is defined as the sum of  $\Delta$  symbol scores  $S_{\Delta} = \sum_{i=1}^{\Delta} F(A_i, B_i)$ . That being the case we look at the probability of a single pair of symbols

$$p_0(a, b) = Pr_0( (A, B) = (a, b) ) = \mu(a)\mu(b)$$

by tilting we obtain

$$p_{\theta}(a, b) = Pr_{\theta}( (A, B) = (a, b) ) = C_{\theta} e^{\theta F(a, b)} p_0(a, b)$$

The next step is to solve for  $C_{\theta}$

$$1 = \sum_{a, b} p_{\theta}(a, b) = C_{\theta} \sum_{a, b} e^{\theta F(a, b)} p_0(a, b) = C_{\theta} \mathbb{E}_0[e^{\theta F(a, b)}]$$

using the definition of  $\phi$  this is  $C_{\theta} e^{\phi(\theta)}$  and hence

$$C_{\theta} = e^{-\phi(\theta)}$$

So for a pair of symbols  $a$  and  $b$  exponential tilting can be done with

$$Pr_{\theta}(a, b) = e^{\theta F(a, b) - \phi(\theta)} p_0(a, b) \tag{8.1}$$

This gives us the tilted probability of a single pair of symbols which enables us to specify the tilted probability of a pair of sequences as well.

As we know from before (exercise 10, problem 35)  $\mathbb{E}_\theta[ F(A, B) ] = \phi'(\theta)$  holds. Together with the definition of  $S_\Delta$  this gives us  $\mathbb{E}_\theta[S_\Delta] = \Delta \phi'(\theta)$

This means that in order to obtain  $\mathbb{E}_\theta[S_\Delta] = x$  we have to chose  $\theta_{\Delta,x}$  such that  $\phi'(\theta) = \frac{x}{\Delta}$ . In the following let  $\theta = \theta_{\Delta,x}$ .

We now come back to our result for the tail distribution of  $S_\Delta$ :

$$\begin{aligned} & \mathbb{E}_0\left[\frac{e^{\theta S_\Delta - \Delta\phi(\theta)}}{e^{\theta S_\Delta - \Delta\phi(\theta)}}, S_\Delta > x\right] \\ &= e^{\Delta\phi(\theta)} \mathbb{E}_0[e^{-\theta S_\Delta} e^{\theta S_\Delta - \Delta\phi(\theta)}, S_\Delta > x] \\ &= e^{\Delta\phi(\theta)} \mathbb{E}_\theta[e^{-\theta S_\Delta}, S_\Delta > x] \\ &= e^{\Delta\phi(\theta) - \theta x} \mathbb{E}_\theta[e^{-\theta(S_\Delta - x)}, S_\Delta > x] \end{aligned}$$

Now it can be shown that the second term in the last transformation is bounded by 1 and by construction not exponentially small. This means that we can approximate  $Pr_0(S_\Delta > x)$  using only the first term.

$$Pr_0(S_\Delta > x) \approx e^{\Delta\phi(\theta) - \theta x} = e^{-x \left(\theta - \frac{\phi(\theta)}{\phi'(\theta)}\right)}$$

The last transformation follows from  $\phi'(\theta) = \frac{x}{\Delta}$  for  $\theta = \theta_{\Delta,x}$ . This result for the tail distribution of  $S_\Delta$  enables us to specify the tail distribution of  $T$ .

$$\begin{aligned} Pr(T > x) &= Pr(\max_{\Delta \geq 0} S_\Delta > x) \approx \max_{\Delta \geq 0} Pr(S_\Delta > x) \\ &\approx \max_{\Delta > 0} e^{-x \left(\theta - \frac{\phi(\theta)}{\phi'(\theta)}\right)} \\ &= e^{-x \min_{\Delta \geq 0} \left(\theta - \frac{\phi(\theta)}{\phi'(\theta)}\right)} \end{aligned}$$

$$\approx e^{-x \min_{\theta: \phi'(\theta) > 0} (\theta - \frac{\phi(\theta)}{\phi'(\theta)})} = e^{-x\theta^*}$$

Where  $\theta^*$  is the unique positive zero of  $\phi$ . The correctness of this last step has been proven in problem 36, exercise 10.

So  $T$  decays with a type two extreme value distribution. Since  $M_n$  is defined as the maximum over  $n^2 T_{ij}$ , the tail distribution of  $Pr(M_n > x)$  is given by  $K^* \cdot n^2 \cdot e^{-\theta^* x}$  for some constant  $K^*$ . This is the result of theorem 2.1 we were interested in.

# Chapter 9

## Monte Carlo Markov Chain Methods (MCMC)

### 9.1 Introduction

Monte Carlo Markov Chain (MCMC) methods are a group of methods that sample from a random walk on a Markov chain. In an MCMC algorithm, the Markov chain is constructed in such a manner, that the unique stationary distribution (see definition 9.3) is the posterior distribution that we want to sample from.

Here are some problems that can be solved by using MCMC methods:

- Sampling of random graphs
- Decoding of Microarray probe signals
- Sampling random points in a subset of  $\mathbb{Z}^d$

We will have a closer look on the last example.

**Example 9.1.** Let us assume the lattice  $\mathbb{Z}^2$  and a subset  $\mathbb{X}$  that includes holes, meaning the points within the holes are not included in  $\mathbb{X}$  (see figure 9.1). We define a membership function  $m_{\mathbb{X}}(x) \in \{1, 0\}$  giving information about whether a point  $x$  is part of  $\mathbb{X}$  or not. Now we want to draw a point  $x \in \mathbb{X}$  randomly (uniformly distributed).

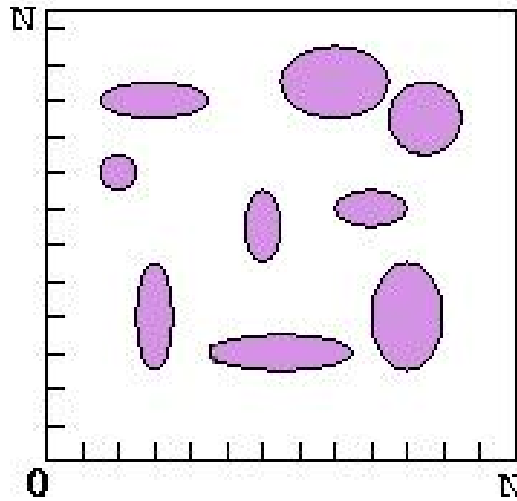
1. The first idea is to preprocess all points from  $\{0, \dots, N\}^2$  by testing their membership. A member is stored in a list. The total number of members is

$$|\mathbb{X}| = \sum_x m_{\mathbb{X}}(x).$$

To select a random point, we draw  $\mathcal{U}$  uniformly from  $\{1, \dots, |\mathbb{X}|\}$ . Finally we count through the list of members until the  $\mathcal{U}^{th}$  member is found. This method certainly is not very elaborate and costs preprocessing as well as counting time.

2. The second approach uses Accept-Reject:

```
repeat:  
  draw x from {0, ... , N}x{0, ... , N}  
  as long as m(x) = 0
```



**Figure 9.1:** A subset  $\mathbb{X}$  of  $\mathbb{Z}^2: \{0, \dots, N\}^2$  with the ellipses representing holes.

If the holes are “small” (do not cover a lot of points), this method will work fine.

3. The last approach is an MCMC method which is in fact a bit too complicated for our needs but we use it here to illustrate the concept of MCMC.

We construct a Markov process that allows us to make a random walk on  $\mathbb{X}$  by the following algorithm:

```

start in x
repeat:
  pick a random direction:
    stay with probability 1-p
    go North with probability p/4
    go West with probability p/4
    go South with probability p/4
    go East with probability p/4

  move into this direction if the new point is in X
  otherwise stay

```

Thus in each step we *propose* an immediate neighbor<sup>1</sup> of the current point and *accept* it if it is a member of  $\mathbb{X}$ . The starting point  $x$  can be any point in  $\mathbb{X}$ , we choose  $x = (0, 0)$ .

With MCMC methods two questions always arise:

1. Will the relative frequencies of the points that are drawn actually converge against the desired distribution (in this case the uniform distribution)?

<sup>1</sup>The neighborhood can be chosen in a different way as well. We define it this way for reasons of simplicity.

2. What is the mixing rate, or in other words how long does it take our random walk to be independent from the starting state?

**Lemma 9.1.** *Suppose we have a Markov chain on the state space  $\mathbb{X}$  represented by the transition matrix  $P$  that consists of the transition probabilities  $P_{ij} = \mathbb{P}(X_{t+1} = j | X_t = i)$ . With  $X_t$  we denote the state at time  $t$ .*

*If the Markov chain is **irreducible** and **aperiodic**, there exists a unique distribution  $\pi$  on  $\mathbb{X}$  that is **stationary**.*

*Additionally for any start state  $i$ , the transition probability to any state  $j$  within  $t$  time steps is independent of  $i$ :*

$$(P^t)_{ij} \rightarrow \pi_j, \text{ for } t \rightarrow \infty$$

There are some notions left to define.

**Definition 9.1 (Irreducibility).** Let  $\{X_t\}_{t \in \mathbb{N}}$  denote a Markov chain over a state space  $\mathbb{X}$  with transition matrix  $P$ .

1. A state  $i \in \mathbb{X}$  **has access to** another state  $j \in \mathbb{X}$  if

$$(P^t)_{ij} > 0$$

for some  $t \in \mathbb{N}$ .

2. Two states  $i$  and  $j$  **communicate**, if state  $i$  has access to  $j$  and  $j$  to  $i$ .
3.  $\{X_t\}_{t \in \mathbb{N}}$  is called an **irreducible** Markov chain, if all pairs of states communicate.

**Definition 9.2 (Aperiodicity).** The period of a state  $i$  is the greatest common divisor of all  $t$  for which  $(P^t)_{ii} > 0$  holds. If the period is 1, meaning that at any time step  $t$  the probability of being in state  $i$  is nonzero if we have already visited  $i$  before,  $i$  is considered to be **aperiodic**.

Essentially the demand for irreducibility and aperiodicity provides that  $(P^t)_{ij} > 0$  for all  $i, j$  and some  $t$ .

**Definition 9.3 (Stationarity).** A distribution  $\pi$  is called a **stationary** or **invariant** distribution of the Markov chain  $\{X_t\}_{t \in \mathbb{N}}$  with transition matrix  $P$ , if it satisfies

$$\pi \cdot P = \pi.$$

Any distribution  $\pi$  satisfying the **detailed balance** or **reversibility** condition,

$$\pi_i \cdot P_{ij} = \pi_j \cdot P_{ji}$$

is a stationary distribution.

The speed of convergence of  $(P^t)_{ij}$  against  $\pi_j$  depends on the largest eigen value different from 1, denoted by  $\lambda_2$ , in the following way:

$$|(P^t)_{ij} - \pi_j| \leq c_i \lambda_2^t$$

with  $c_i$  denoting some constant. We can see that if  $\lambda_2$  is very close to 1,  $(P^t)_{ij}$  will converge more slowly. Usually  $\lambda_2$  is very hard to estimate, especially if  $P$  is unknown. Therefore the so called burn-in time, the number of time steps before the first sample is taken into account (in order to provide independence from the starting state), has to be chosen carefully.

Now let us have a closer look on some MCMC methods.

## 9.2 The Metropolis Algorithm (1953)

The problem is to sample from a certain distribution  $\pi$ . We start with some state  $i \in \mathbb{X}$ . The next state  $j$  is proposed with probability  $q_{ij}$  (which can be 0 for some  $j$ ). We assume that the matrix of proposal probabilities  $q$  is symmetric, thus  $q_{ij} = q_{ji}$  holds for all states  $i$  and  $j$ . The proposal is accepted with probability

$$\alpha_{ij} := \min \left\{ 1, \frac{\pi_j}{\pi_i} \right\}.$$

In each time step, the next state is proposed and accepted accordingly.

The claim is now that the Markov chain we have constructed in this way has the stationary distribution  $\pi$ .

**Proof.** We will show that the detailed balance condition holds for  $\pi$ . For  $i = j$  the equality is trivial, thus we assume  $i$  and  $j$  to be different. Then

$$\begin{aligned} \pi_i \cdot P_{ij} &= \pi_i \cdot q_{ij} \cdot \alpha_{ij} \\ &= \begin{cases} \pi_i \cdot q_{ij} \cdot 1 & \text{if } \alpha_{ij} = 1 \\ \pi_i \cdot q_{ij} \cdot \frac{\pi_j}{\pi_i} & \text{if } \alpha_{ij} \leq 1 \end{cases} \\ &= \begin{cases} \pi_i \cdot q_{ij} \cdot 1 & \text{if } \alpha_{ji} \leq 1 \\ \pi_j \cdot q_{ij} & \text{if } \alpha_{ji} = 1 \end{cases} \\ &= \pi_j \cdot q_{ij} \cdot \alpha_{ji} \\ &= \pi_j \cdot q_{ji} \cdot \alpha_{ji} \\ &= \pi_j \cdot P_{ji} \end{aligned}$$

□

## 9.3 The Hastings Algorithm (1970)

The difference between the Hastings and the Metropolis algorithms lies within the assumption of the proposal probabilities being symmetric. Assume everything as before (except the

proposal probabilities), then the proposal is accepted with probability

$$\alpha_{ij} := \min \left\{ 1, \frac{\pi_j \cdot q_{ji}}{q_{ij} \cdot \pi_i} \right\}.$$

Analogously to section 9.2 it can be shown that the Markov chain we have constructed has the stationary distribution  $\pi$ .

## 9.4 The Tierney Algorithm (1974)

In this section we introduce the so called *independence sampler* constructed by the Tierney algorithm. Now as in 9.3, the proposal probabilities  $q_{ij}$  do not have to be symmetric, and additionally they are independent of the proposed state  $j$ , thus  $q_{ij} = q_i \quad \forall j$ .

The proposed state is then accepted with probability

$$\begin{aligned} \alpha_{ij} &:= \min \left\{ 1, \frac{\pi_j \cdot q_i}{q_j \cdot \pi_i} \right\} \\ &= \min \left\{ 1, \frac{w_j}{w_i} \right\} \end{aligned}$$

with the weights  $w_z := \frac{\pi_z}{q_z}$  for all  $z \in \mathbb{X}$ .

Again, the proof that  $\pi$  is the stationary distribution of the underlying Markov chain is analogous to section 9.2.

The independence sampler only works well, if  $q$  is a good approximation of  $\pi$ .